



#BIKES4ERP

11/09/2020

Informatics 370 Deliverable 2 – Functional Specification

The functional specification deliverable aims to give further information about what the system should do. This document thus includes the detailed use case diagrams and their narratives for describing the steps included in each use case, a complete set of data flow diagrams and, finally, activity diagrams to depict the flow of information through the system. To support the diagrams, detailed descriptions of all inputs and outputs are provided and validated against the requirements.



Name (from left to right)	Surname	Student Number
Byron J.	Donald	17237913
Hansen	Li	17073708
Tapiwarufaro	Ndoro (Group Leader)	17210918
Cameron	King	17165904
Gareth	Phillips	17004162

Virtual Visionaries
GROUP 17

Table of Contents

Table of Contents.....	i
Table of Figures.....	iv
1. Deliverable Introduction.....	2
1.1 Introduction	2
2. Use Cases	3
2.1 Introduction	3
2.2 Use Case Diagrams	4
2.2.1 Introduction	4
2.2.2 Sub-system 1: User	5
2.2.3 Sub-system 2: Mechanic Schedule	6
2.2.4 Sub-system 3: Tracking Hardware	7
2.2.5 Sub-system 4: Student.....	8
2.2.6 Sub-system 5: Bicycle	9
2.2.7 Sub-system 6: Bicycle Part.....	10
2.2.8 Sub-system 7: Tracking.....	11
2.2.9 Sub-system 8: Job Tasks	12
2.2.10 Sub-system 9: School	13
2.2.11 Sub-system 10: Job	14
2.2.12 Sub-system 11: Reporting	15
2.2.13 Sub-system 12: Notification	16
2.2.14 Sub-system 13: Administration.....	17
2.2.15 Conclusion	18
2.3 Use Case Narratives	19
2.3.1 Introduction	19
2.3.2 Sub-system 1: User	20
2.3.3 Sub-system 2: Mechanic Schedule	45
2.3.4 Sub-system 3: Tracking Hardware	49
2.3.5 Sub-system 4: Student.....	67
2.3.6 Sub-system 5: Bicycle	89
2.3.7 Sub-system 6: Bicycle Part.....	106
2.3.8 Sub-system 7: Tracking.....	124
2.3.9 Sub-system 8: Job Tasks	128
2.3.10 Sub-system 9: School	146
2.3.11 Sub-system 10: Job	155



2.3.12	Sub-system 11: Reporting	164
2.3.13	Sub-system 12: Notification	178
2.3.14	Sub-system 13: Administration.....	182
2.3.15	Conclusion	199
3.	Process Models.....	200
3.1	Introduction	200
3.2	Context Diagram	201
3.2.1	Introduction	201
3.2.2	Diagram	202
3.2.3	Conclusion	207
3.3	Functional Decomposition Diagram.....	208
3.3.1	Introduction	208
3.3.2	Diagram	209
3.3.3	Conclusion	222
3.4	Data Flow Diagrams	223
3.4.1	Introduction	223
3.4.2	High Level Data Flow Diagrams	224
3.4.3	Mid-level Data Flow Diagrams	248
3.4.4	Primitive Level Data Flow Diagrams.....	329
3.4.5	Conclusion	410
4.	UML Modelling	411
4.1	Introduction	411
4.2	Activity Diagrams.....	412
4.2.1	Sub-system 1: User	412
4.2.2	Sub-system 2: Mechanic Schedule	423
4.2.3	Sub-system 3: Tracking Hardware	425
4.2.4	Sub-system 4: Student.....	433
4.2.5	Sub-system 5: Bicycle	443
4.2.6	Sub-system 6: Bicycle Part.....	451
4.2.7	Sub-system 7: Tracking.....	459
4.2.8	Sub-system 8: Job Tasks	461
4.2.9	Sub-system 9: School	469
4.2.10	Sub-system 10: Job	473
4.2.11	Sub-system 11: Reporting	477
4.2.12	Sub-system 12: Notification	483
4.2.13	Sub-system 13: Administration.....	485



4.3	Conclusion	493
5.	Data Model	494
5.1	Introduction	494
5.2	Logical Entity Relationship Diagram	495
5.1	Conclusion	496
6.	Interfaces and other inputs	497
6.1	Introduction	497
6.2	Interfaces and Inputs Table	498
6.3	Conclusion	540
7.	Reports and other outputs	541
7.1	Introduction	541
7.2	Reports and other outputs	542
7.1	Conclusion	595
8.	Validation	596
8.1	Introduction	596
8.2	Validate the details of the functional specification against the Requirements	597
8.3	Conclusion	609
9.	Deliverable Conclusion & Sign-off	610
9.1	Conclusion	610
9.2	Client Sign-off	611
9.3	Group Sign-off	611
10.	Complexity	612
10.1	Complexity Matrix	612

Table of Figures

Figure 1 User use case diagram	5
Figure 2 Mechanic use case diagram	6
Figure 3 Tracking use case diagram.....	7
Figure 4 Student use case diagram.....	8
Figure 5 Bicycle use case diagram	9
Figure 6 Bicycle part use case diagram.....	10
Figure 7 Tracking use case diagram.....	11
Figure 8 Job Tasks use case diagram	12
Figure 9 School use case diagram	13
Figure 10 Job use case diagram	14
Figure 11 Reporting use case diagram	15
Figure 12 Notification use case diagram	16
Figure 13 Administration use case diagram.....	17
Figure 14 Context Diagram Sub-system 1 and 2	202
Figure 15 Context Diagram Sub-system 3 and 4	203
Figure 16 Context Diagram Sub-system 5, 6 and 7	204
Figure 17 Context Diagram Sub-system 8 and 9	205
Figure 18 Context Diagram Sub-system 10, 11, 12 and 13.....	206
Figure 19 User Function Decomposition Diagram (Split).....	209
Figure 20 Mechanic Schedule Function Decomposition Diagram (Split)	210
Figure 21 Tracking Hardware Function Decomposition Diagram (Split)	211
Figure 22 Student Function Decomposition Diagram (Split).....	212
Figure 23 Bicycle Function Decomposition Diagram (Split).....	213
Figure 24 Bicycle Part Function Decomposition Diagram (Split)	214
Figure 25 Tracking Function Decomposition Diagram (Split)	215
Figure 26 Job Tasks Function Decomposition Diagram (Split).....	216
Figure 27 School Function Decomposition Diagram (Split).....	217
Figure 28 Job Function Decomposition Diagram (Split).....	218
Figure 29 Reporting Function Decomposition Diagram (Split).....	219
Figure 30 Notification Function Decomposition Diagram (Split).....	220
Figure 31 Administration Function Decomposition Diagram (Split)	221
Figure 32 User high level diagram	227
Figure 33 Mechanic Schedule high level diagram	228
Figure 34 Hardware high level diagram	230
Figure 35 Student high level diagram.....	233
Figure 36 Bicycle high level diagram	235
Figure 37 Bicycle Part high level diagram	237
Figure 38 Tracking high level diagram.....	238
Figure 39 Job Task high level	240
Figure 40 School high level	241
Figure 41 Job high level diagram	242
Figure 42 Reporting high level diagram	244
Figure 43 Notification high level diagram.....	245
Figure 44 Administration high level diagram	247
Figure 45 Login mid-level diagram	248
Figure 46 Logout mid-level diagram.....	249
Figure 47 Forgot Password mid-level diagram	250

Figure 48 Add User mid-level diagram	251
Figure 49 Search User mid-level diagram	252
Figure 50 Update User mid-level diagram	253
Figure 51 Delete User mid-level diagram	254
Figure 52 Add User Type mid-level diagram	255
Figure 53 Search User Type mid-level diagram	256
Figure 54 Update User Type mid-level diagram	257
Figure 55 Delete User Type mid-level diagram	258
Figure 56 Complete Job mid-level diagram	259
Figure 57 View Job tasks mid-level diagram	260
Figure 58 Add new antenna mid-level diagram.....	261
Figure 59 Search Antenna mid-level diagram.....	262
Figure 60 Update Antenna mid-level diagram.....	263
Figure 61 Delete Antenna mid-level diagram.....	264
Figure 62 Add new tag mid-level diagram	265
Figure 63 Search Tag mid-level diagram	266
Figure 64 Update Tag mid-level diagram	267
Figure 65 Delete Tag mid-level diagram	268
Figure 66 Add Student mid-level diagram.....	269
Figure 67 Search Student Mid-level diagram	270
Figure 68 Update Student mid-level diagram	271
Figure 69 Delete Student mid-level diagram.....	272
Figure 70 Capture Student Marks mid-level diagram.....	273
Figure 71 Search Student Marks mid-level diagram	274
Figure 72 Add Mark Type mid-level diagram	275
Figure 73 Search Mark Type mid-level diagram	276
Figure 74 Update Mark Type mid-level diagram	277
Figure 75 Delete Mark Type mid-level diagram	278
Figure 76 Add Bicycle mid-level diagram	279
Figure 77 Search Bicycle mid-level diagram	280
Figure 78 Update Bicycle mid-level diagram.....	281
Figure 79 Delete Bicycle mid-level diagram	282
Figure 80 Assign Bicycle mid-level diagram	283
Figure 81 Unassign Bicycle mid-level diagram	284
Figure 82 Assign Tag mid-level diagram	285
Figure 83 Unassign tag mid-level diagram	286
Figure 84 Add Bicycle Part mid-level diagram.....	287
Figure 85 Search Bicycle Part diagram.....	288
Figure 86 Update Bicycle Part mid-level diagram.....	289
Figure 87 Delete Bicycle Part mid-level diagram.....	290
Figure 88 Add Bicycle Section mid-level diagram	291
Figure 89 Search Bicycle Section mid-level diagram	292
Figure 90 Update Bicycle Section mid-level diagram	293
Figure 91 Delete Bicycle Section mid-level diagram	294
Figure 92 Add Log Entry mid-level diagram.....	295
Figure 93 Search Log Entry mid-level diagram.....	296
Figure 94 Add Fault Task mid-level diagram	297
Figure 95 Search Fault Task mid-level diagram	298
Figure 96 Update Fault Task mid-level diagram	299

Figure 97 Delete Fault Task mid-level diagram	300
Figure 98 Add Maintenance Task mid-level diagram	301
Figure 99 Search Maintenance Task mid-level diagram	302
Figure 100 Update Maintenance Task mid-level diagram	303
Figure 101 Delete Maintenance Task mid-level diagram	304
Figure 102 Add School mid-level diagram	305
Figure 103 Search School mid-level diagram	306
Figure 104 Update School mid-level diagram	307
Figure 105 Delete School mid-level diagram	308
Figure 106 Report Fault mid-level diagram	309
Figure 107 View Jobs mid-level diagram	310
Figure 108 Check-in Bicycle mid-level diagram	311
Figure 109 Report Maintenance mid-level diagram	312
Figure 110 Generate Maintenance Schedule mid-level diagram	313
Figure 111 Generate Maintenance Requirements report mid-level diagram	314
Figure 112 Generate Student Report mid-level diagram	315
Figure 113 Generate Bicycle History Report mid-level diagram	316
Figure 114 Generate Log Report mid-level diagram	317
Figure 115 Generate Parts Used report mid-level diagram	318
Figure 116 Send Student Did Not Arrive Notification mid-level diagram	319
Figure 117 Send Student Maintenance Notification mid-level diagram	320
Figure 118 Backup Data mid-level diagram	321
Figure 119 Restore Data mid-level diagram	322
Figure 120 View Audit Trail Log mid-level diagram	323
Figure 121 Search Audit Trail Entry mid-level diagram	324
Figure 122 Add Route mid-level diagram	325
Figure 123 Search Route mid-level diagram	326
Figure 124 Update Route mid-level diagram	327
Figure 125 Delete Route mid-level diagram	328
Figure 126 Login Primitive Diagram	329
Figure 127 Logout Primitive Diagram	330
Figure 128 Forgot Password Primitive Diagram	331
Figure 129 Add User Primitive Diagram	332
Figure 130 Search User Primitive Diagram	333
Figure 131 Update User Primitive Diagram	334
Figure 132 Delete User Primitive Diagram	335
Figure 133 Add User Type Primitive Diagram	336
Figure 134 Search User Type Primitive Diagram	337
Figure 135 Update User Type Primitive Diagram	338
Figure 136 Delete User Type Primitive Diagram	339
Figure 137 Complete Job Primitive Diagram	340
Figure 138 View Job Tasks Primitive Diagram	341
Figure 139 Add Antenna Primitive Diagram	342
Figure 140 Search Antenna Primitive Diagram	343
Figure 141 Update Antenna Primitive Diagram	344
Figure 142 Delete Antenna Primitive Diagram	345
Figure 143 Add Tag Primitive Diagram	346
Figure 144 Search Tag Primitive Diagram	347
Figure 145 Update Tag Primitive Diagram	348

Figure 146 Delete Tag Primitive Diagram.....	349
Figure 147 Add Student Primitive Diagram	350
Figure 148 Search Student Primitive Diagram	351
Figure 149 Update Student Primitive Diagram	352
Figure 150 Delete Student Primitive Diagram	353
Figure 151 Add Student Mark Primitive Diagram	354
Figure 152 Search Student Primitive Diagram	355
Figure 153 Add Mark Type Primitive Diagram.....	356
Figure 154 Search Mark Type Primitive Diagram.....	357
Figure 155 Update Mark Type Primitive Diagram.....	358
Figure 156 Delete Mark Type Primitive Diagram.....	359
Figure 157 Add Bicycle Primitive Diagram	360
Figure 158 Search Bicycle Primitive Diagram	361
Figure 159 Update Bicycle Primitive Diagram	362
Figure 160 Delete Bicycle Primitive Diagram.....	363
Figure 161 Assign Bicycle to Student Primitive Diagram.....	364
Figure 162 Unassign Bicycle from Student Primitive Diagram.....	365
Figure 163 Assign Tag to Bicycle Primitive Diagram	366
Figure 164 Unassign Tag from Bicycle Primitive Diagram	367
Figure 165 Add Bicycle Part Primitive Diagram	368
Figure 166 Search Bicycle Part Primitive Diagram	369
Figure 167 Update Bicycle Part Primitive Diagram	370
Figure 168 Delete Bicycle Part Primitive Diagram	371
Figure 169 Add Bicycle Section Primitive Diagram	372
Figure 170 Search Bicycle Section Primitive Diagram	373
Figure 171 Update Bicycle Section Primitive Diagram	374
Figure 172 Delete Bicycle Section Primitive Diagram.....	375
Figure 173 Add Log Entry Primitive Diagram	376
Figure 174 Search Log Entry Primitive Diagram	377
Figure 175 Add Fault Task Primitive Diagram	378
Figure 176 Search Fault Task Primitive Diagram.....	379
Figure 177 Update Fault Task Primitive Diagram	380
Figure 178 Delete Fault Task Primitive Diagram.....	381
Figure 179 Add Maintenance Task Primitive Diagram.....	382
Figure 180 Search Maintenance Task Primitive Diagram.....	383
Figure 181 Update Maintenance Task Primitive Diagram.....	384
Figure 182 Delete Maintenance Task Primitive Diagram.....	385
Figure 183 Add School Primitive Diagram.....	386
Figure 184 Search School Primitive Diagram.....	387
Figure 185 Update School Primitive Diagram.....	388
Figure 186 Delete School Primitive Diagram.....	389
Figure 187 Report Fault Primitive Diagram.....	390
Figure 188 View Jobs Primitive Diagram	391
Figure 189 Check-in Bicycle Primitive Diagram	392
Figure 190 Report Maintenance Primitive Diagram	393
Figure 191 Generate Maintenance Schedule Primitive Diagram.....	394
Figure 192 Generate Maintenance Requirements Report Primitive Diagram	395
Figure 193 Generate Student Report Primitive Diagram	396
Figure 194 Generate Bicycle History Report Primitive Diagram	397

Figure 195 Generate Log Report Primitive Diagram	398
Figure 196 Generate Parts Used Report Primitive Diagram	399
Figure 197 Student Did Not Arrive Notification Primitive Diagram	400
Figure 198 Send Student Maintenance Notification Primitive Diagram	401
Figure 199 Backup Data Primitive Diagram	402
Figure 200 Restore Data Primitive Diagram	403
Figure 201 View Audit Trail Primitive Diagram	404
Figure 202 Search Audit Trail Entry Primitive Diagram	405
Figure 203 Add Route Primitive Diagram	406
Figure 204 Search Route Primitive Diagram	407
Figure 205 Update Route Primitive Diagram	408
Figure 206 Delete Route Primitive Diagram	409
Figure 207 Login Activity Diagram	412
Figure 208 Logout Activity Diagram	413
Figure 209 Forgot Password	414
Figure 210 Add User Activity Diagram	415
Figure 211 Search User Activity Diagram	416
Figure 212 Update User Activity Diagram	417
Figure 213 Delete User Activity Diagram	418
Figure 214 Add User Type Activity Diagram	419
Figure 215 Search User Type Activity Diagrams	420
Figure 216 Update User Type Activity Diagram	421
Figure 217 Delete User Type Activity Diagram	422
Figure 218 Complete Job Activity Diagram	423
Figure 219 View Job Tasks Activity Diagram	424
Figure 220 Add Antenna Activity Diagram	425
Figure 221 Search Antenna Activity Diagram	426
Figure 222 Update Antenna Activity Diagram	427
Figure 223 Delete Antenna Activity Diagram	428
Figure 224 Add Tag Activity Diagram	429
Figure 225 Search Tag Activity Diagram	430
Figure 226 Update Tag Activity Diagram	431
Figure 227 Delete Tag Activity Diagram	432
Figure 228 Add Student Activity Diagram	433
Figure 229 Search Student Activity Diagram	434
Figure 230 Update Student Activity Diagram	435
Figure 231 Delete Student Activity Diagram	436
Figure 232 Capture Student Marks Activity Diagram	437
Figure 233 Search Student Marks Activity Diagram	438
Figure 234 Add Mark Type Activity Diagram	439
Figure 235 Search Mark Type Activity Diagram	440
Figure 236 Update Mark Type Activity Diagram	441
Figure 237 Delete Mark Type Activity Diagram	442
Figure 238 Add Bicycle Activity Diagram	443
Figure 239 Search Bicycle Activity Diagram	444
Figure 240 Update Bicycle Activity Diagram	445
Figure 241 Delete Bicycle Activity Diagram	446
Figure 242 Assign Bicycle to Student Activity Diagram	447
Figure 243 Unassign Bicycle from Student Activity Diagram	448

Figure 244 Assign Tag to Bicycle Activity Diagram	449
Figure 245 Unassign Tag from Bicycle Activity Diagram	450
Figure 246 Add Bicycle Part Activity Diagram	451
Figure 247 Search Bicycle Part Activity Diagram	452
Figure 248 Update Bicycle Part Activity Diagram	453
Figure 249 Delete Bicycle Part Activity Diagram	454
Figure 250 Add Bicycle Section Activity Diagram.....	455
Figure 251 Search Bicycle Section Activity Diagram.....	456
Figure 252 Update Bicycle Section Activity Diagram	457
Figure 253 Delete Bicycle Section Activity Diagram.....	458
Figure 254 Add Log Entry Activity Diagram	459
Figure 255 Search Log Entry Activity Diagram	460
Figure 256 Add Fault Task Activity Diagram.....	461
Figure 257 Search Fault Task Activity Diagram.....	462
Figure 258 Update Fault Task Activity Diagram	463
Figure 259 Delete Fault Task Activity Diagram.....	464
Figure 260 Add Maintenance Task Activity Diagram.....	465
Figure 261 Search Maintenance Task Activity Diagram.....	466
Figure 262 Update Maintenance Task Activity Diagram.....	467
Figure 263 Delete Maintenance Task Activity Diagram.....	468
Figure 264 Add School Activity Diagram	469
Figure 265 Search School Activity Diagram	470
Figure 266 Update School Activity Diagram.....	471
Figure 267 Delete School Activity Diagram	472
Figure 268 Report Fault Activity Diagram.....	473
Figure 269 View Jobs Activity Diagrams.....	474
Figure 270 Check-in Bicycle Activity Diagram	475
Figure 271 Report Maintenance Activity Diagram	476
Figure 272 Generate Maintenance Schedule Activity Diagram.....	477
Figure 273 Generate Maintenance Requirements Report Activity Diagram	478
Figure 274 Generate Student Report Activity Diagram	479
Figure 275 Generate Bicycle History Activity Diagram	480
Figure 276 Generate Log Report Activity Diagram	481
Figure 277 Generate Parts Used Report Activity Diagram	482
Figure 278 Send Student Did Not Arrive Notification Activity Diagram	483
Figure 279 Send Student Maintenance Notification Activity Diagram	484
Figure 280 Backup Data Activity Diagram	485
Figure 281 Restore Data Activity Diagram.....	486
Figure 282 View Audit Trail Activity Diagram	487
Figure 283 Search Audit Trail Entry Activity Diagram.....	488
Figure 284 Add Route Activity Diagram	489
Figure 285 Search Route Activity Diagram	490
Figure 286 Update Route Activity Diagram	491
Figure 287 Delete Route Activity Diagram	492
Figure 288 Logical Entity Relational Diagram	495



1. Deliverable Introduction

1.1 Introduction

The purpose of this deliverable is to gain insight into how we're going to design this system through the use of diagrams and illustrations. By doing it this way we can map out our ideas and processes so that for the next deliverable when we design our wire-frames we already have the backbone of the system set up and we can therefore base our wireframes off this information.

We will therefore need to ensure that our processes and flows make logical sense so that we can understand our system as best as possible but is subject to change as the system develops.



2. Use Cases

2.1 Introduction

Use cases refer to the different processes within a system. They are found within individual sub systems which in turn makes up the entire system. They can therefore be seen as the building blocks of a system when it comes to the process within a system.



2.2 Use Case Diagrams

2.2.1 Introduction

Use case diagrams are used to demonstrate the interactions between the users and the system. This will include all the processes and sub-systems of our system as well as how the users interact with them and how they will interact with each other.

Lastly Use case diagrams also shows what outputs the system generates if there are any.



2.2.2 Sub-system 1: User

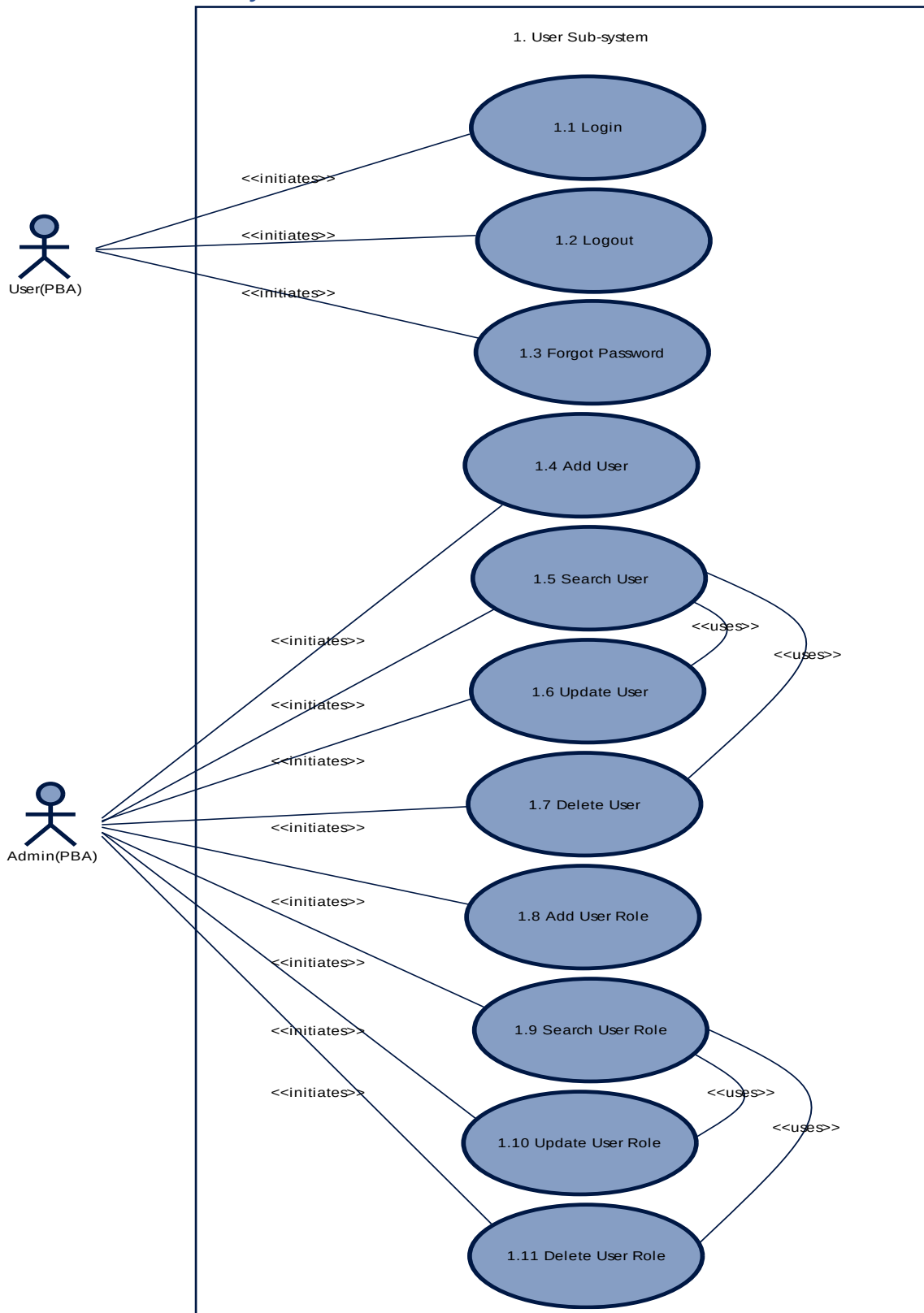


Figure 1 User use case diagram



2.2.3 Sub-system 2: Mechanic Schedule

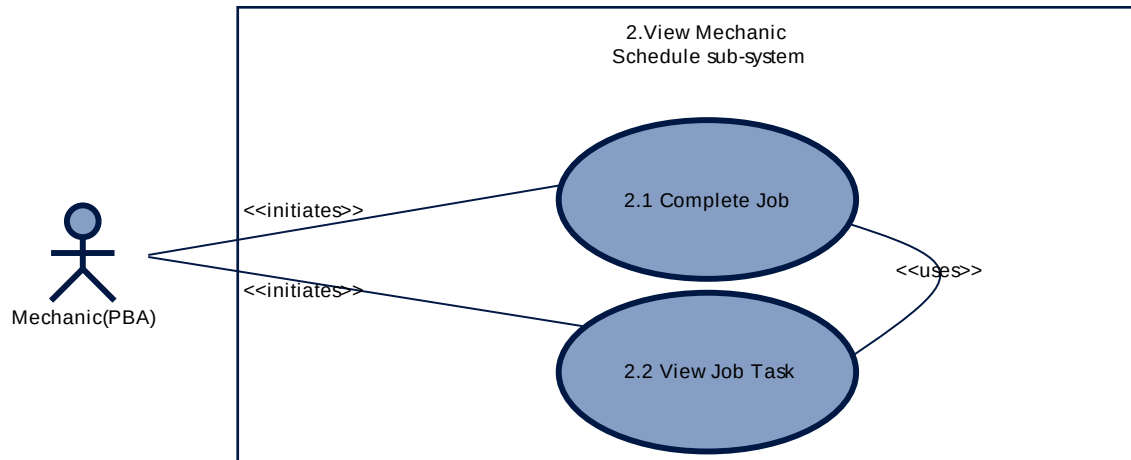


Figure 2 Mechanic use case diagram



2.2.4 Sub-system 3: Tracking Hardware

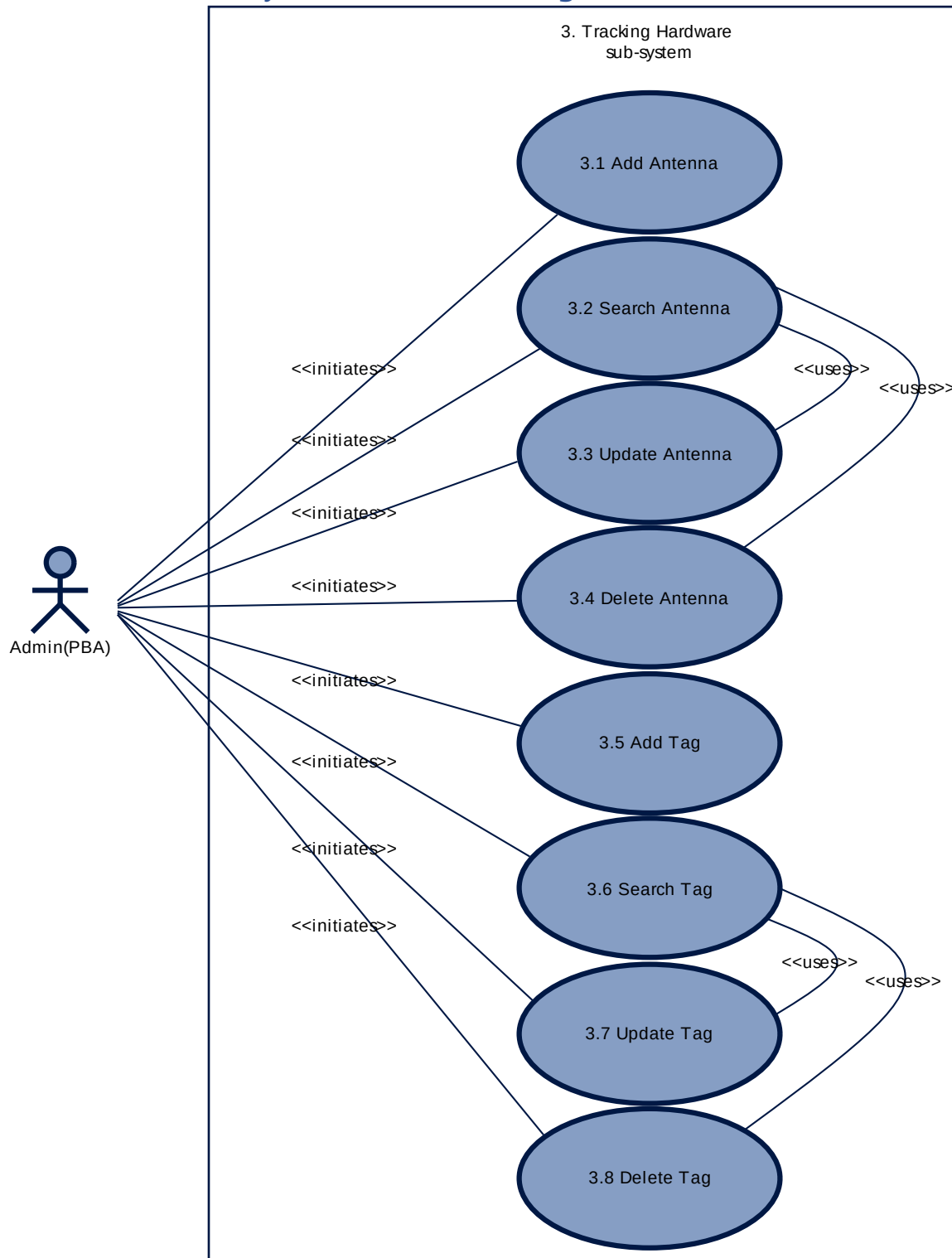


Figure 3 Tracking use case diagram



2.2.5 Sub-system 4: Student

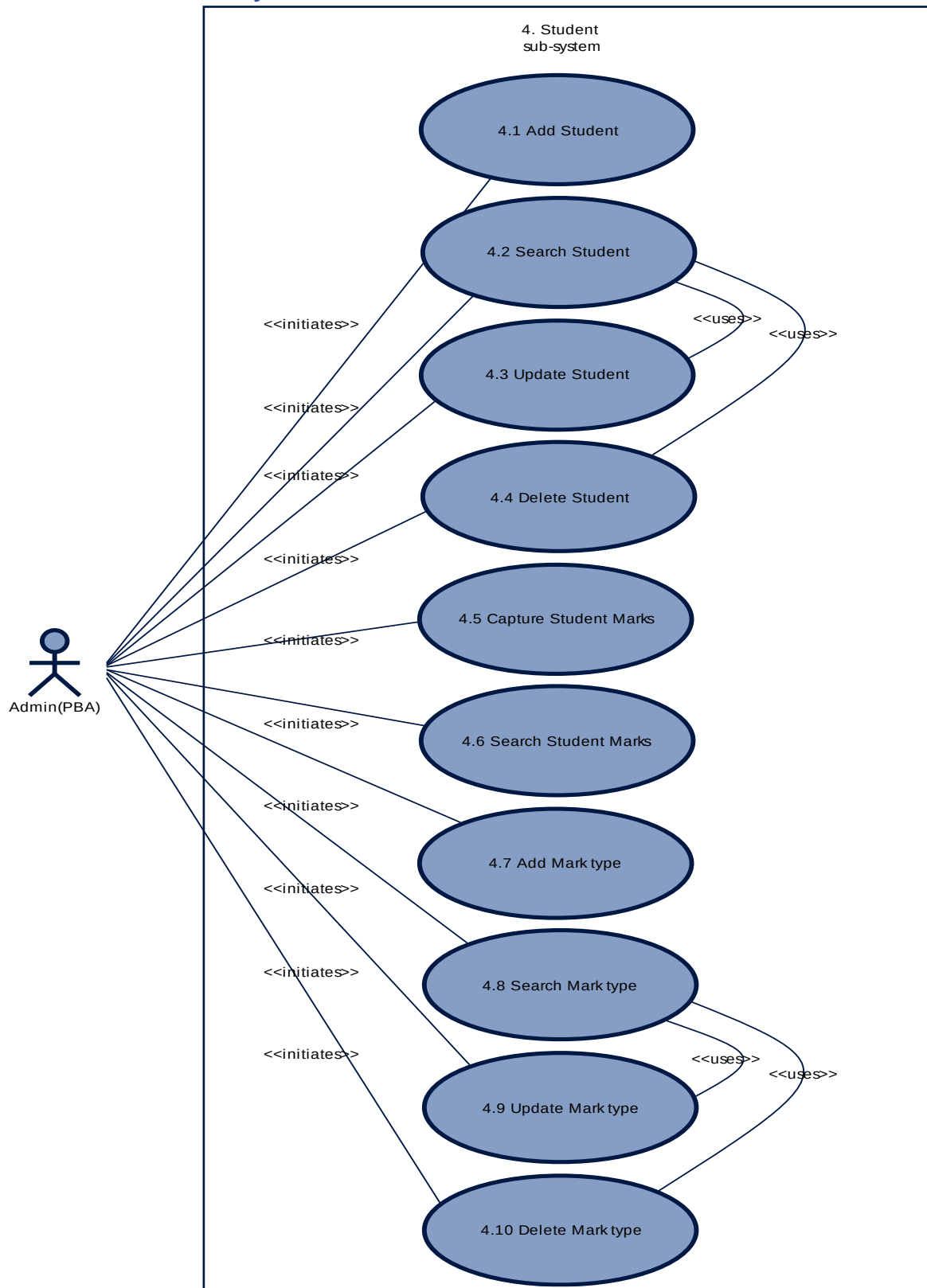


Figure 4 Student use case diagram



2.2.6 Sub-system 5: Bicycle

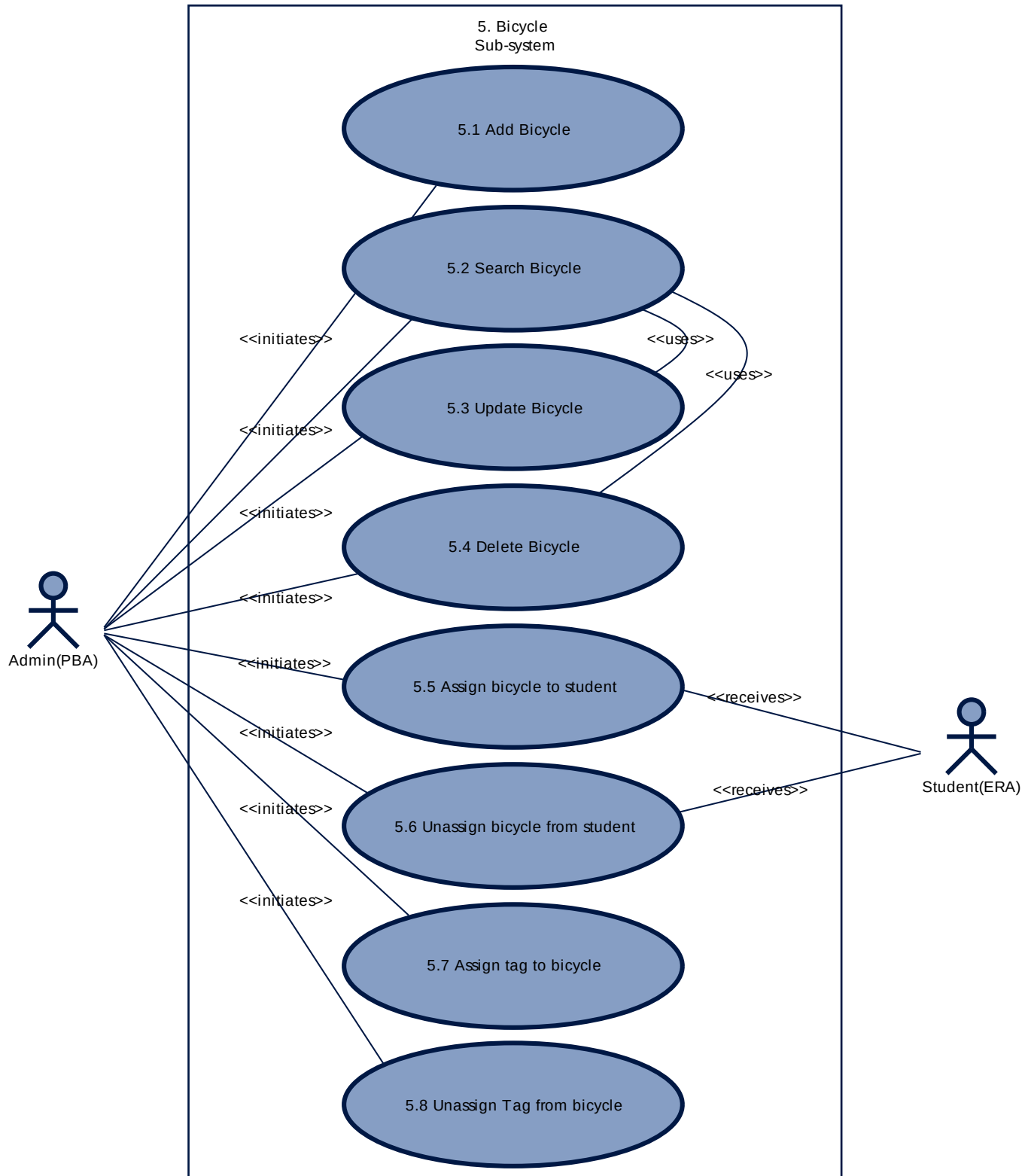


Figure 5 Bicycle use case diagram



2.2.7 Sub-system 6: Bicycle Part

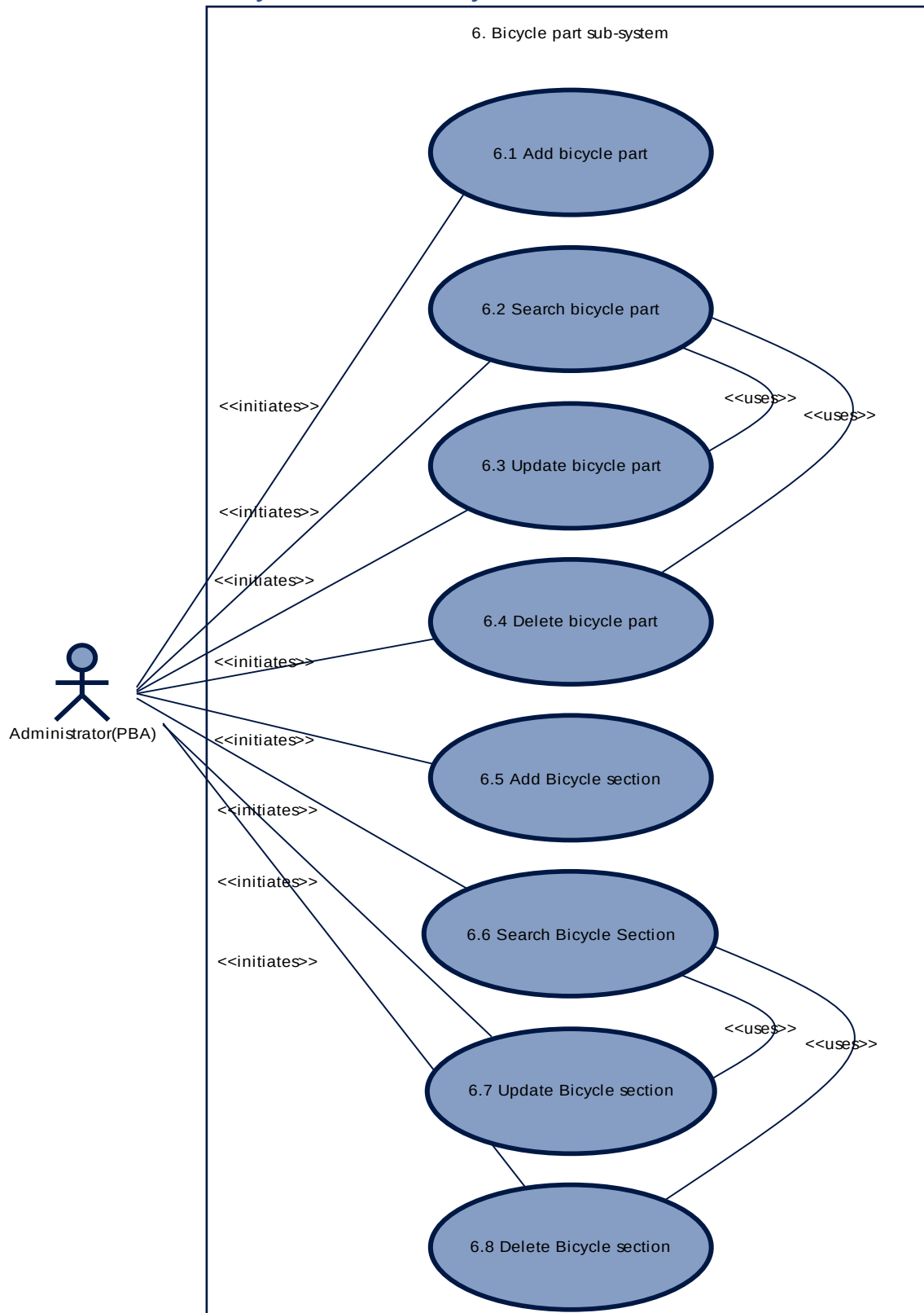


Figure 6 Bicycle part use case diagram



2.2.8 Sub-system 7: Tracking

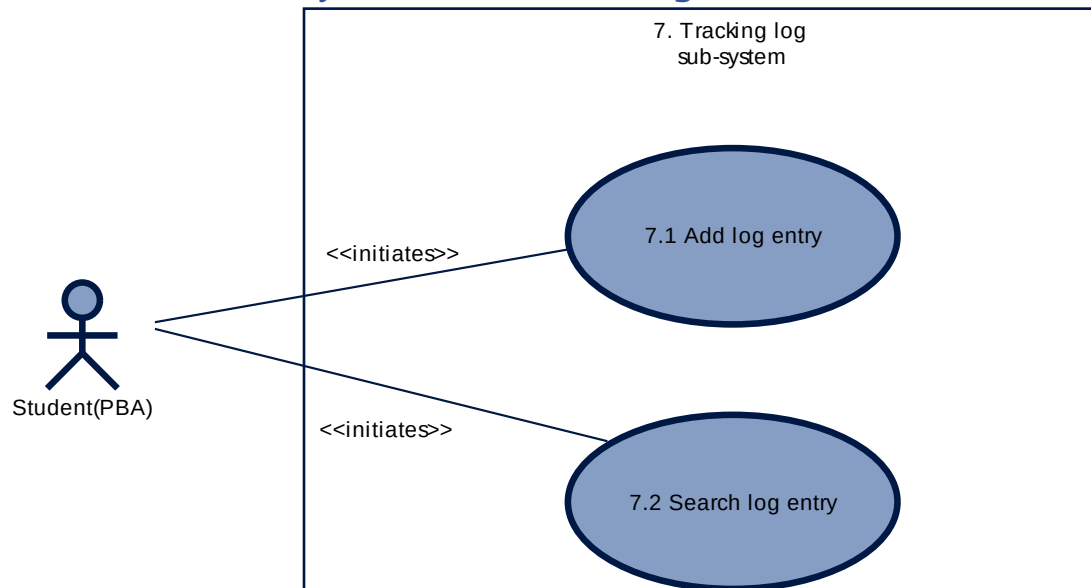


Figure 7 Tracking use case diagram



2.2.9 Sub-system 8: Job Tasks

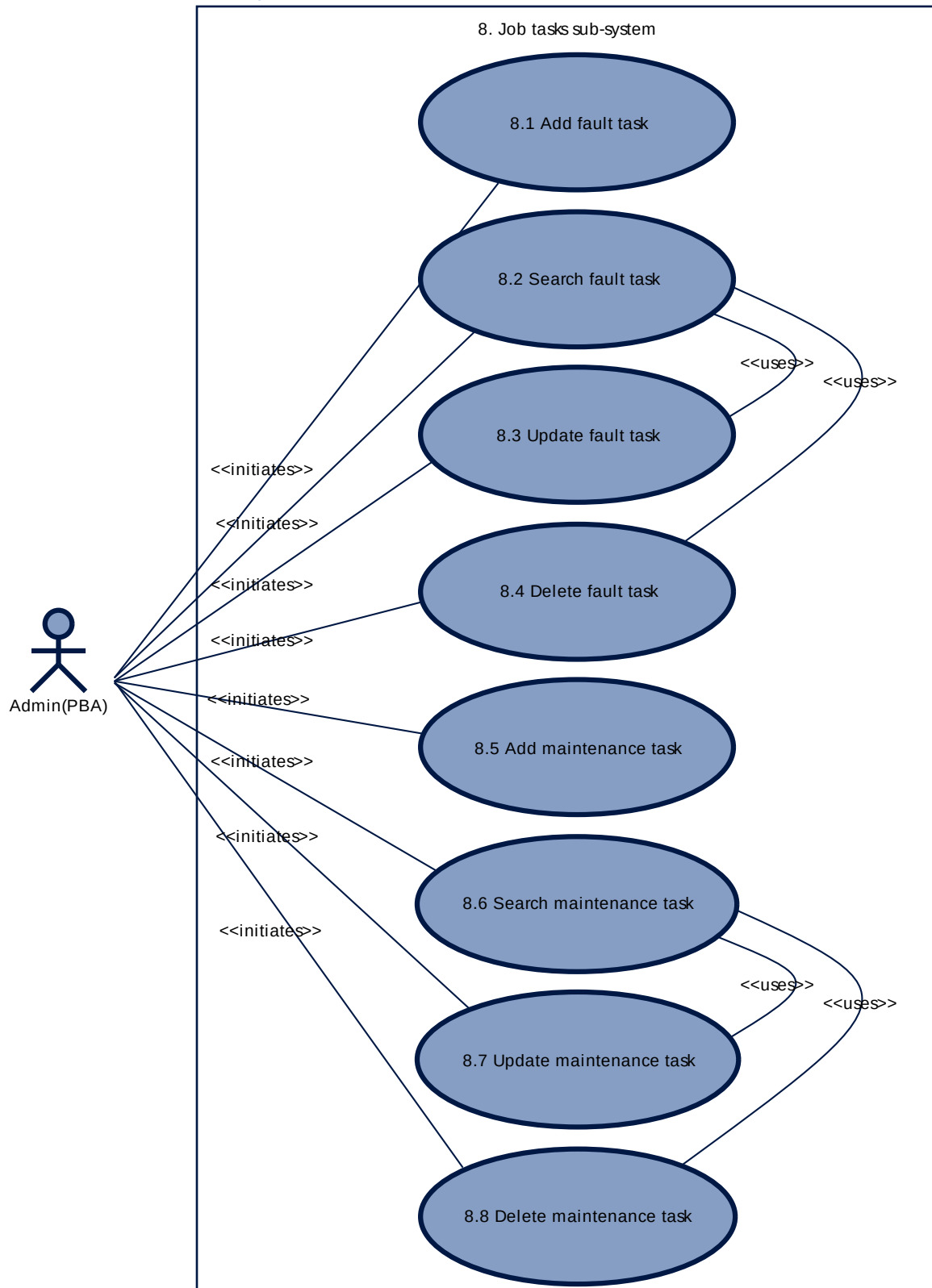


Figure 8 Job Tasks use case diagram



2.2.10 Sub-system 9: School

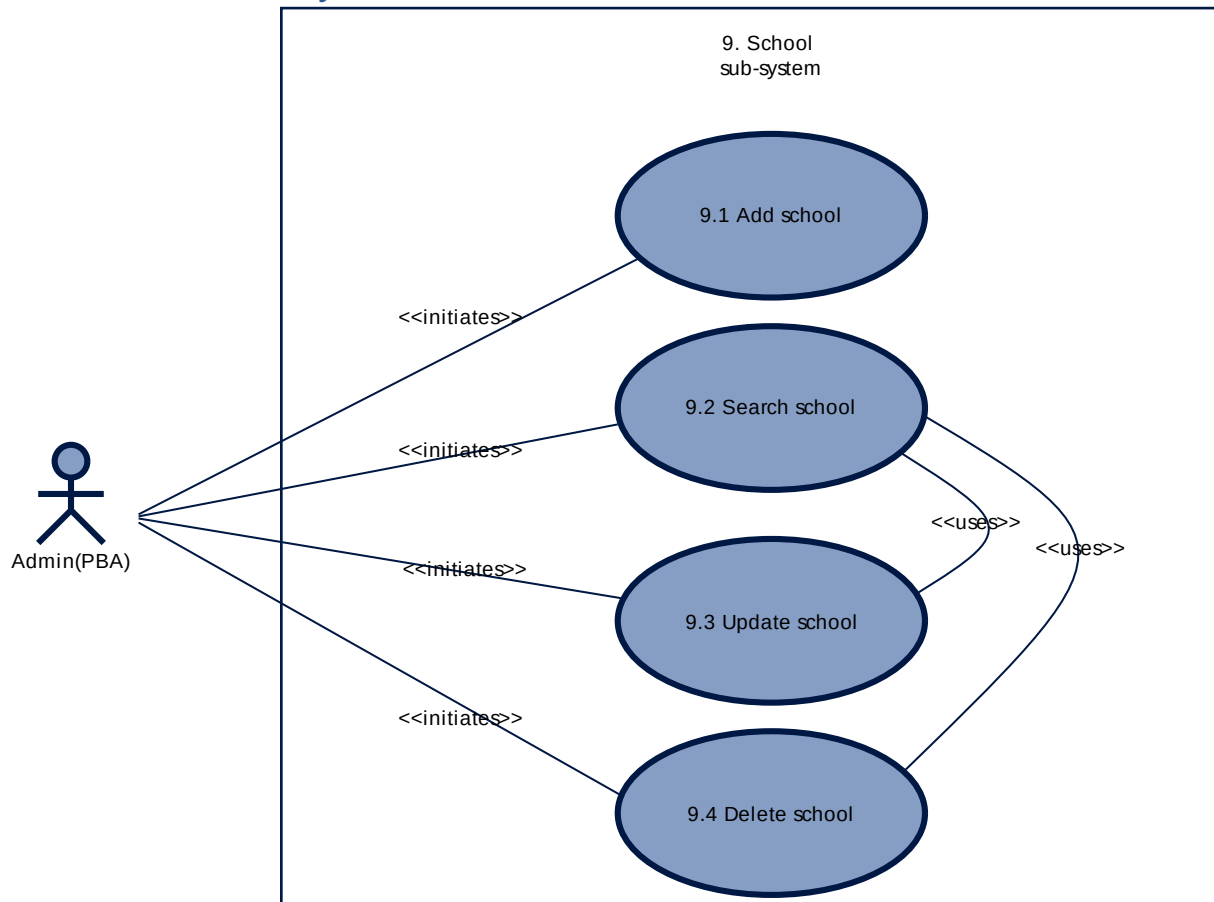


Figure 9 School use case diagram



2.2.11 Sub-system 10: Job

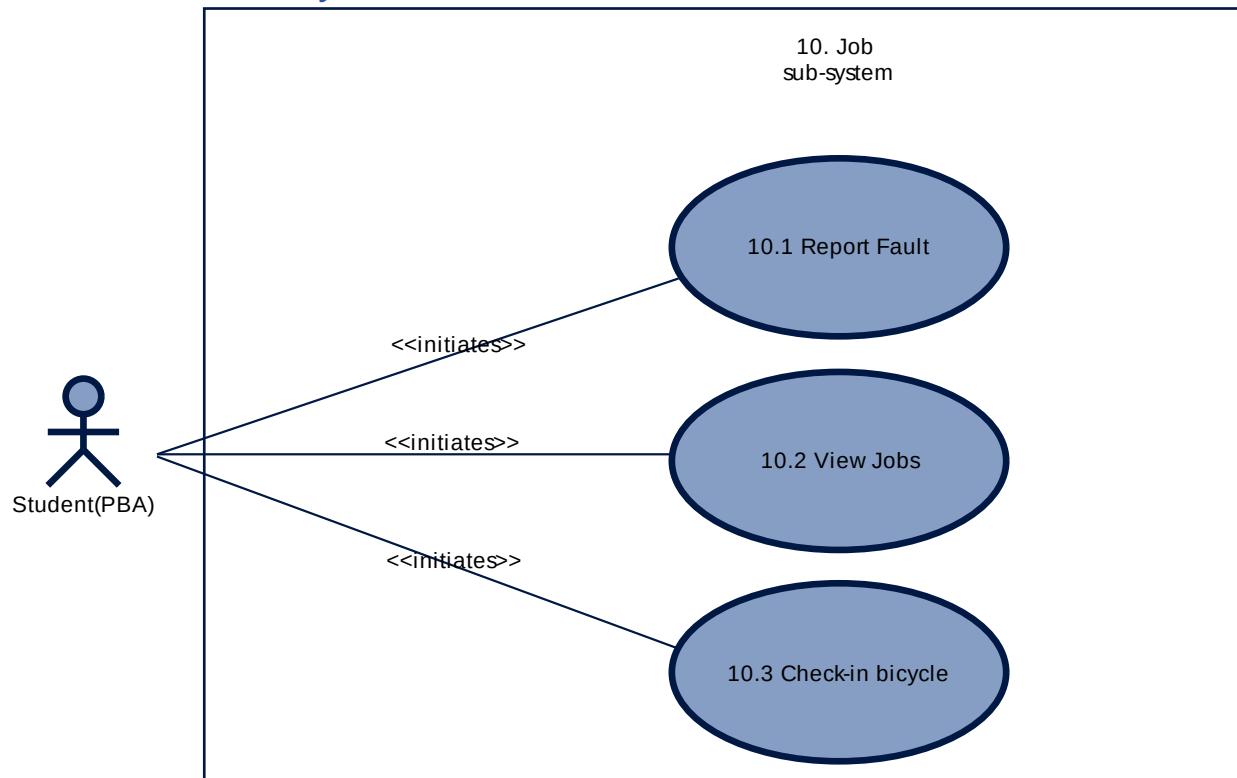


Figure 10 Job use case diagram



2.2.12 Sub-system 11: Reporting

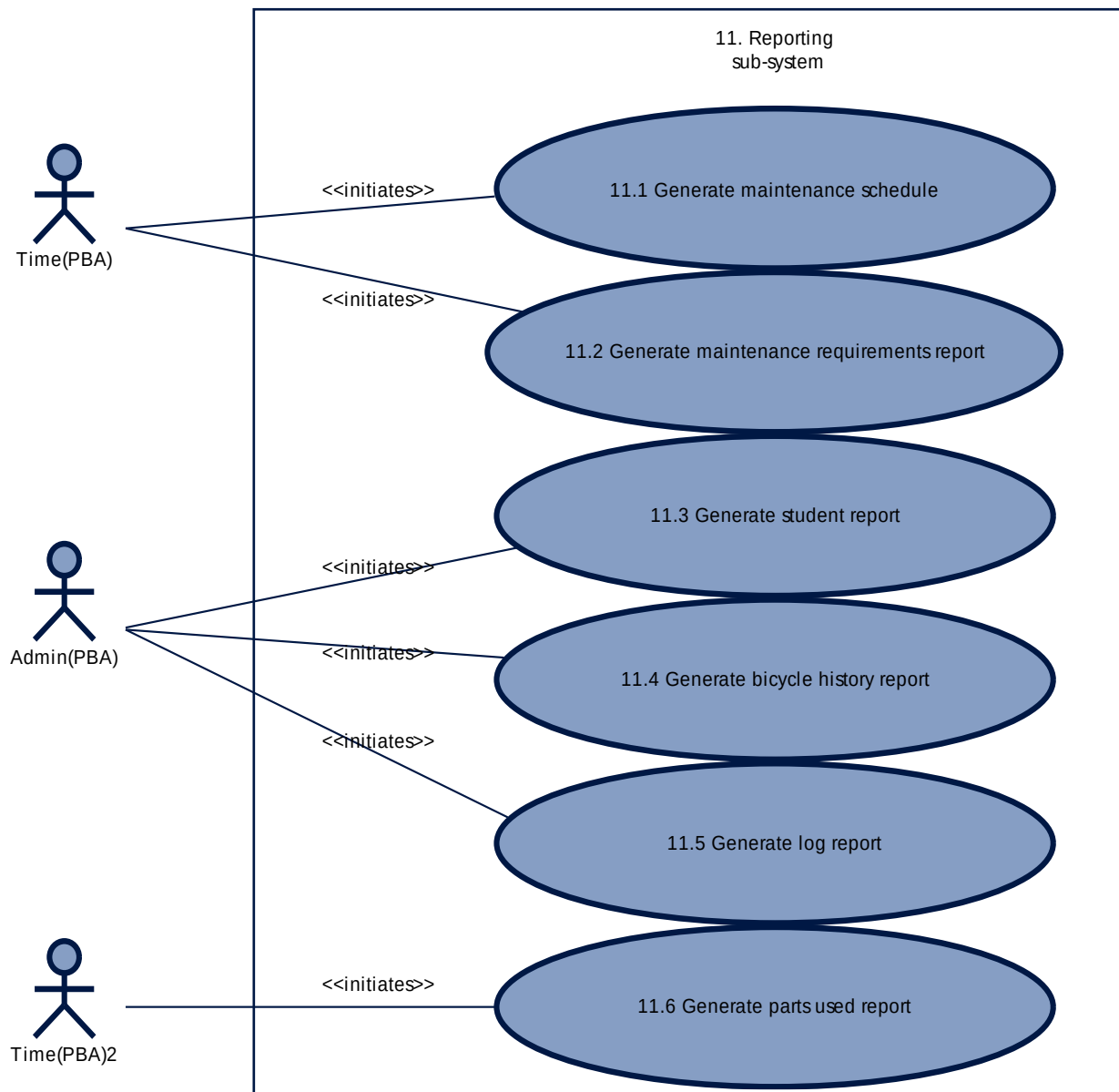


Figure 11 Reporting use case diagram



2.2.13 Sub-system 12: Notification

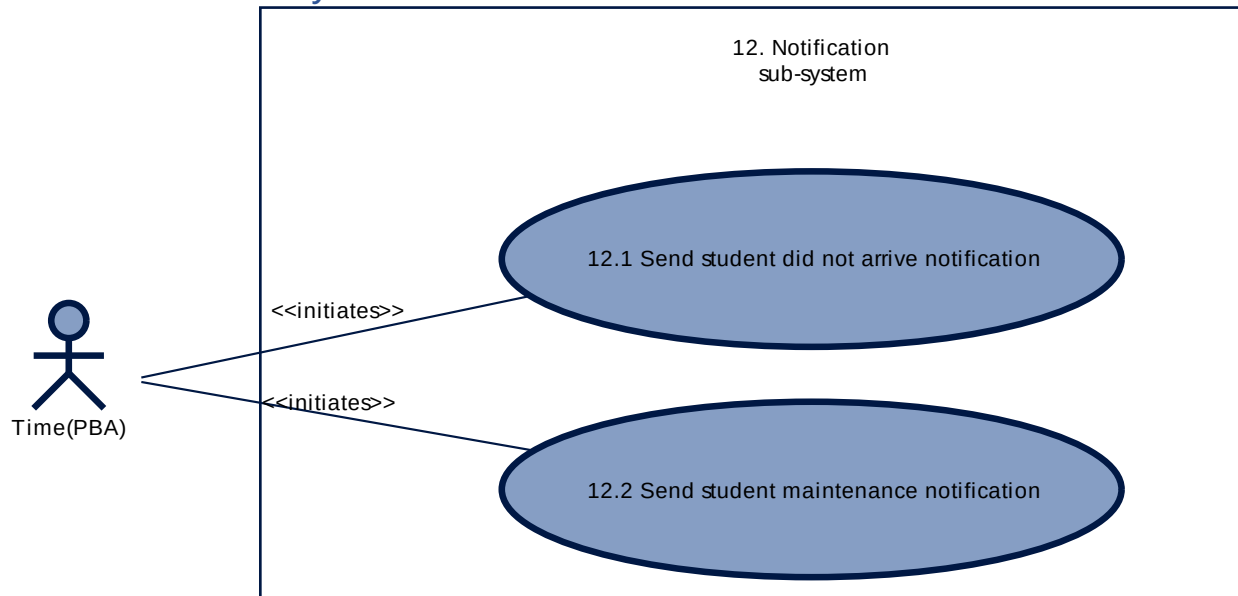


Figure 12 Notification use case diagram



2.2.14 Sub-system 13: Administration

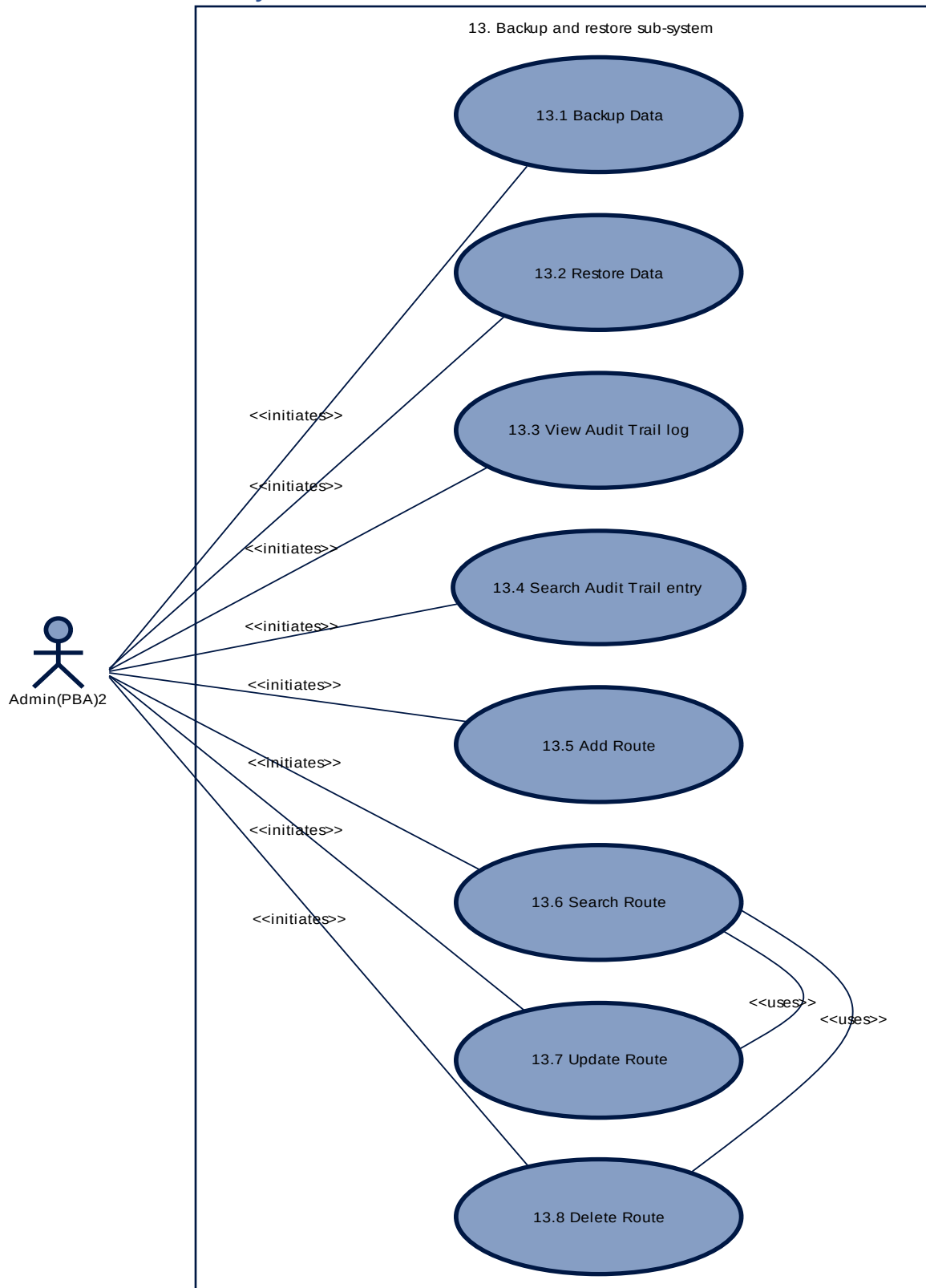


Figure 13 Administration use case diagram



Virtual Visionaries
We See What You Don't

11/09/20
#BIKES4ERP



2.2.15 Conclusion

The conclusion after looking at the use case diagrams is that we are now able to gain better insight into the multiple processes within each individual sub-system of the main system.



2.3 Use Case Narratives

2.3.1 Introduction

Use case Narratives expand on Use case diagrams by going into a lot more detail about each individual Use case. This includes the steps involved in each use case as well as how the use interacts with the process. Lastly the Use case narratives also assists in mentioning critical information such as business rules, what's changed in a system as well as triggers.



2.3.2 Sub-system 1: User

2.3.2.1 Login

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Login	Use Case Type:
Use Case ID:	1.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	User	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby a User (Administrator or Mechanic) request to log into the system. The use case is initiated when the User requests to log into the system and proceeds to input their email and password which is then validated by comparing the inputted details with the Username and Password stored in the database. The use case will conclude when the User is granted access to system functionality.	
Pre-Condition:	The User must be registered on the system.	
Trigger:	The User requests to log into the system.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The User requests to login.	Step 2: The system loads the login User details and prompts the User to enter the following details: - Username - Password
	Step 3: The User inputs their login details. [alt]	
	Step 4: The User selects to login to the system.	Step 5: The system captures the entered details and verifies the Username and Password fields have been filled. [alt]



		Step 6: The system READS the User table and validates that the inputted Username and Password are identical to the Username and Password stored in the database. [alt]
		Step 7: The system grants the User access to the systems functionality based on their User Type access level from the UserType table and loads the User home details.
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 3: The User selects the Forgot Password option; the system will Invoke Use Case 1.3 Forgot Password.	
	Alt Step 5: System detects a verification error; the required fields have not been filled. The system will notify the User of the error. Return to Step 3.	
	Alt Step 6: System detects a validation error; the Username and Password combination is invalid and does not match the Username and Password details stored in the database. The system will notify the User of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the User has been successfully logged into the system.	
Post-Condition:	• The User will be granted access to system functionality based on their access level stored in the UserType table. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only registered Users will be granted access to the system functionality. • Users must be granted different access permissions based on their User Type.	
Implementation Constraints and Specifications:	• Passwords will be hashed and compared using the SHA256 algorithm.	
Assumptions:	• None	
Open Issues:	• None	



2.3.2.2 Logout

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Logout	Use Case Type:
Use Case ID:	1.2	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	User	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby a User requests to log out of the system. The User will select the logout option and confirm their logout request. The use case concludes when the system notifies the User that they have been successfully logged out.	
Pre-Condition:	The User must be logged into the system.	
Trigger:	The User requests to logout of the system.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The User requests to the Logout option.	Step 2: The system will prompt the User to confirm the logout request.
	Step 3: The User confirms the logout request. [alt]	Step 4: The system will logout the User and display the login User details.
		Step 5: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID



Alternate Courses:	Alt Step 3: The User does not confirm the logout request, the Use Case is then terminated.
Conclusion:	The use case narrative concludes when the system notifies the User that they have been successfully logged out of the system.
Post-Condition:	• The User can no longer navigate the system or access functionality. • The AuditTrail table has been successfully updated.
Business Rules:	• The system will automatically logout a User after 10 minutes of inactivity.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.2.3 *Forgot Password*

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Forgot Password	Use Case Type:
Use Case ID:	1.3	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	User	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby a User requests to reset their password. The User will select the Forgot Password option and the system will prompt the User to enter their email address and notify them that they will be sent a reset password link. The email will instruct the User the follow the link and they will be prompted to input and confirm their new password. The Use Case will conclude when the system notifies the User that the password has been reset successfully.	
Pre-Condition:	The User must be registered on the system.	
Trigger:	The User requests to reset their password.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The User requests to reset their password.	Step 2: The system Displays the Forgot Password details and prompts the User to input their email address.
	Step 3: The User inputs their email address.	
	Step 4: The User selects to receive the reset password link.	Step 5: The system captures the inputted email address verifies that the email address field filled in and in the correct format.



	Step 6: The system READS the User details from the User table to validate that the inputted email address is registered to a User that exists on the system. [alt]
	Step 7: The system sends a Reset Password link to the Users email address.
	Step 8: The system notifies the User that the Reset Password link has been sent.
Step 9: The User selects the Reset Password link.	Step 10: The system displays the Reset Password details and prompts the User to input the following details: • New Password • Confirm New Password
Step 11: The User inputs the new password details.	
Step 12: The User selects the Reset Password details.	Step 13: The system captures the inputted password details and verifies that required fields have been filled. The system will also verify that the following inputted fields match identically as well as meet the password specification criteria: [alt] ○ New Password ○ Confirm New Password
	Step 14: The system prompts the User to confirm their password reset.
Step 15: The User confirms their password reset. [alt]	Step 16: The system UPDATES the Password attribute in the User table with the New Password inputted by the User.
	Step 17: The system notifies the User that their password has been successfully updated. [alt]
	Step 18: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID



Alternate Courses:	Alt Step 6: The inputted email address is not registered to a User on the system, the system will notify the User of the error. Return to Step 3.
	Alt Step 13: The entered passwords are not identical, or they do meet the password specification criteria. The system will notify the User of the error. Return to Step 11.
	Alt Step 15: The User does not confirm they want to continue with the Reset Password process. Terminate the use case.
	Alt Step 17: The password was not updated successfully; the system will notify the User of the error. Return to Step 11.
Conclusion:	The use case narrative concludes when the system notifies the user that their password has been successfully updated.
Post-Condition:	- The Users password has been updated in the User table. - The AuditTrail table has been successfully updated
Business Rules:	Only a User can reset their password.
Implementation Constraints and Specifications:	- Passwords meet the following specifications: - Contain more than 7 characters - Contain an uppercase letter - Contain a number - All passwords stored in the system are hashed.
Assumptions:	None
Open Issues:	None



2.3.2.4 Add User

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Add User	Use Case Type:
Use Case ID:	1.4	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new user to the system. The system will prompt the Administrator to input the required user details. The Administrator will confirm the inputted user details and the system will add the new user details to the database. The use case will conclude when the system notifies the Administrator that the user has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new user.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new user.	Step 2: The system displays the add user details and prompts the Administrator to input the required user details: • Username • Password • FirstName • Surname • Gender • Age • Email • Telephone



		User Type
	Step3: The Administrator inputs the user details.	
	Step 4: The Administrator selects to add the new user details to the system.	Step 5: The system captures and validates the following fields are filled and the inputted details are in the correct format [alt]:
		Step 6: The system READS the User details from the User table in the database and verifies that no duplicate user exists in the system by comparing the inputted Email address to the Email address of the registered user existing in the User table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the user.
	Step 8: The Administrator confirms to add the new user. [alt]	Step 10: The user is assigned a unique UserID which is set to the next consecutive number after the previous UserID in the database and creates the new user in User table.
		Step 11: The system notifies the Administrator that the user has been added successfully. [alt]
		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted user already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new user. Return to step 3.	



	Alt Step 11: The user was not added successfully, the system will notify the Administrator of the error. Return to step 3.
Conclusion:	The use case concludes when the Administrator has been notified that the user has been added successfully.
Post-Condition:	• A new user is added to the system with the relevant details. • The AuditTrail table is successfully updated. • The User table is successfully updated.
Business Rules:	• Only an Administrator can add a user to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.2.5 Search User

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search User	Use Case Type:
Use Case ID:	1.5	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a user. The system will prompt the Administrator to select the user search criteria they wish to search by and the Administrator will input the user search criteria details. The use case concludes when the system displays the searched user details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a user's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a user in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the user search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific user details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified user details from the User table in the database. [alt]
		Step 7: The system displays the following details: - User table - Username - Password - FirstName - Surname - Gender - Age - Email - Telephone - UserType table - UserType
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: - AuditTrailID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified user does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched user details are displayed.	
Post-Condition:	- The user has been searched successfully. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can search for a user.	
Implementation Constraints and Specifications:	None	
Assumptions:	The user exists in the system.	
Open Issues:	None	



2.3.2.6 Update User

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Update User	Use Case Type:
Use Case ID:	1.6	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a user. The system will prompt the Administrator to input the specified user search criteria of the user they wish to update. The system will then display the details of the specific user for the Administrator to make changes and update the user details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the user has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a user.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific user in the system.	Step 2: The system invokes Use Case 1.5 Search User.
	Step 3: The Administrator inputs the updated user details.	
	Step 4: The Administrator selects to update the user details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]:



		• Username • Password • FirstName • Surname • Gender • Age • Email • Telephone • User Type
		Step 6: The system prompts the Administrator to confirm the updated user details.
	Step 7: The Administrator confirms to update the user details. [alt]	Step 8: The system updates the specific user in the User table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the user details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The user was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the user was updated successfully.	
Post-Condition:	• The User table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can update a user.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The user exists in the system.	
Open Issues:	• None	



2.3.2.7 Delete User

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Delete User	Use Case Type:
Use Case ID:	1.7	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific user from the system. The system will prompt the Administrator to select the user search criteria of the user they wish to delete. The Administrator will confirm the deletion of the searched user and the system will delete the selected user from the database. The use case will conclude when the system has notified the Administrator that the user has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific user.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific user from the system.	Step 2: The system Invokes Use Case 1.5 Search User
	Step 3: The administrator selects to delete the user.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the user. [alt]	Step 6: The system removes the selected user from the User table and updates the table in the database.



		Step 7: The system notifies the administrator that the user has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does not confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the user has been deleted successfully.	
Post-Condition:	• The User table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a user.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The user exists in the system.	
Open Issues:	• None	



2.3.2.8 Add User Type

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Add User Type	Use Case Type:
Use Case ID:	1.8	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new user type to the system. The system will prompt the Administrator to input the required user type details. The Administrator will confirm the inputted user type details and the system will add the new user type details to the database. The use case will conclude when the system notifies the Administrator that the user type has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new user type.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new user type.	Step 2: The system displays the add user type details and prompts the Administrator to input the required user type details: - User Type - User Type Permissions
	Step3: The Administrator inputs the user type details.	
	Step 4: The Administrator selects to add the new user type details to the system.	Step 5: The system captures and validates the following fields are filled and the



		inputted details are in the correct format [alt]:
		Step 6: The system READS the User Type details from the UserType table in the database and verifies that no duplicate user type exists on the system by comparing the inputted User Type to the User Types in the UserType table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the user type.
	Step 8: The Administrator confirms to add the new user type. [alt]	Step 10: The user type is assigned a unique UserID which is set to the next consecutive number after the previous UserID in the database and creates the new user type in UserType table.
		Step 11: The system notifies the Administrator that the user type has been added successfully. [alt]
		Step 12: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted user type already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new user type. Return to step 3.	
	Alt Step 11: The user type was not added successfully, the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case concludes when the Administrator has been notified that the user type has been added successfully.	
Post-Condition:	• A new user type is added to the system with the relevant details. • The AuditTrail table is successfully updated. • The UserType table is successfully updated.	



Business Rules:	Only an Administrator can add a user type to the system.
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None



2.3.2.9 Search User Type

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search User Type	Use Case Type:
Use Case ID:	1.9	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a user type. The system will prompt the Administrator to select the user type search criteria they wish to search by and the Administrator will input the user type search criteria details. The use case concludes when the system displays the searched user type details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a user type's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a user type in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the user type search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific user type details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified user type details from the UserType table in the database. [alt]
		Step 7: The system displays the following details: - UserType table - UserTypeID - UserType - UserTypePermissions
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: - AuditTrailID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified user type does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched user type details are displayed.	
Post-Condition:	- The user type has been searched successfully. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can search for a user type.	
Implementation Constraints and Specifications:	None	
Assumptions:	The user type exists in the system.	
Open Issues:	None	



2.3.2.10 Update User Type

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Update User Type	Use Case Type:
Use Case ID:	1.10	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a user type. The system will prompt the Administrator to input the specified user type search criteria of the user type they wish to update. The system will then display the details of the specific user type for the Administrator to make changes and update the user type details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the user type has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a user type.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific user type in the system.	Step 2: The system invokes Use Case 1.9 Search User Type.
	Step 3: The Administrator inputs the updated user details.	
	Step 4: The Administrator selects to update the user details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]:



		- User Type - User Type Permissions
		Step 6: The system prompts the Administrator to confirm the updated user type details.
	Step 7: The Administrator confirms to update the user type details. [alt]	Step 8: The system updates the specific user type in the UserType table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the user type details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The user type was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the user type was updated successfully.	
Post-Condition:	- The UserType table has been updated and changes saved to the database. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can update a user type.	
Implementation Constraints and Specifications:	None	
Assumptions:	The user type exists in the system.	
Open Issues:	None	



2.3.2.11 Delete User Type

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Delete User Type	Use Case Type:
Use Case ID:	1.11	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific user type from the system. The system will prompt the Administrator to select the user type search criteria of the user type they wish to delete. The Administrator will confirm the deletion of the searched user type and the system will delete the selected user type from the database. The use case will conclude when the system has notified the Administrator that the user type has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific user type.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific user type from the system.	Step 2: The system Invokes Use Case 1.9 Search User Type
	Step 3: The administrator selects to delete the user type.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the user type. [alt]	Step 6: The system removes the selected user type from the UserType table and updates the table in the database.



		Step 7: The system notifies the administrator that the user type has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the user type has been deleted successfully.	
Post-Condition:	• The UserType table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a user type.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The user type exists in the system.	
Open Issues:	• None	



2.3.3 Sub-system 2: Mechanic Schedule

2.3.3.1 Complete Job

System Name:	VVs
Author:	Byron Donald & Tapiwarufaro Ndoro
Date:	06/05/2020
Version:	2.0

Use Case Name:	Complete Job	Use Case Type:
Use Case ID:	2.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Mechanic	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby the mechanic interacts with the system to change the status of a job to completed (Both Maintenance and Repair Jobs). This is done by retrieving details of that job and updating the status to complete once they are done attending to all the tasks for the job. The use case concludes when the system notifies the mechanic that the job was completed successfully.	
Pre-Condition:	There must be pending jobs on the system assigned to that mechanic.	
Trigger:	The mechanic initiates the system process by requesting to complete a job.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The mechanic initiates the system process by requesting to complete a job.	Step 2: The system Invokes Use Case 10.2 View Job tasks
		Step 3: The system prompts the Mechanic to select a specific job to complete.
	Step 4: The mechanic selects specific job to complete.	Step 5: The system requests if the Mechanic used the required parts.



	Step 6: The mechanic confirms the use of the required parts [alt].	Step 7: The system updates the PartUsed in the ReportFaultTask Table and saves the updated details to the database.
		Step 8: The system updates the status of the job in the ReportLine Table and saves the updated details to the database.
		Step 9: The system notifies the mechanic that the job was completed successfully.
		Step 10: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 6: The Mechanic selects parts not used, Skip Step 7 and move to Step 8.	
Conclusion:	The use case concludes when the job status is changed to assigned.	
Post-Condition:	• No duplicate jobs are assigned. • Audit Trail is successfully updated.	
Business Rules:	• The process is initiated every Friday	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The system prioritizes the oldest jobs	
Open Issues:	• None	



2.3.3.2 View Job tasks

System Name:	VVs
Author:	Byron Donald
Date:	13/04/2020
Version:	1.0

Use Case Name:	View Job Tasks	Use Case Type:
Use Case ID:	2.2	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Mechanic	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby the Mechanic requests to view all the tasks of the specific job. The Mechanic will initiate the search after which the system will provide the different search/filter criteria available for the Mechanic to choose from and fill in. Then the system will then display the relevant job tasks.	
Pre-Condition:	Job exist with tasks in it.	
Trigger:	The Mechanic requests to view job task in the system.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Mechanic requests to view the tasks in a job within the system.	Step 2: The system prompts the Mechanic to select the job search criteria and input the required details for the specified criteria.
	Step 3: The Mechanic inputs the specified criteria details.	
	Step 4: The Mechanic selects to view the specified jobs tasks details. [alt]	Step 5: The system captures and validates the search criteria details are entered in the correct format. [alt]



		Step 6: The system READS all the Report Line details that meet the criteria from the ReportLine table in the database. [alt]
		Step 7: The system displays the Report Line details: • PartFaultID • SectionPartID • FaultReportID • ReportLineStatusID
		Step 8: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • UserID • DateTime • TransactionDescription
Alternate Courses:	Alt Step 4: The Mechanic does not initiate the search. Return to Step 3.	
	Alt Step 5: System validation error, formatting error or required fields Have not been entered. Return to Step 3	
	Alt Step 6: No entries exist in the database that meet the criteria. Return to Step 3.	
Conclusion:	All relevant jobs have been displayed.	
Post-Condition:	• The jobs have been searched successfully.	
Business Rules:	• Only authorized users and functions can view jobs.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• None	
Open Issues:	• None	



2.3.4 Sub-system 3: Tracking Hardware

2.3.4.1 Add Antenna

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Add Antenna	Use Case Type:
Use Case ID:	3.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new antenna to the system. The system will prompt the Administrator to input the required antenna details. The Administrator will confirm the inputted antenna details and the system will add the new antenna details to the database. The use case will conclude when the system notifies the Administrator that the antenna has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new antenna.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new antenna.	Step 2: The system displays the add antenna details and prompts the Administrator to input the required antenna details: - Coordinates - Date installed - Antenna Status



	Step3: The Administrator inputs the antenna details.	
	Step 4: The Administrator selects to add the new antenna details to the system.	Step 5: The system captures and validates the following fields are filled and the inputted details are in the correct format [alt]:
		Step 6: The system READS the Antenna details from the Antenna table in the database and verifies that no duplicate antenna exists on the system by comparing the inputted Antenna to the Antennas in the Antenna table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the antenna.
	Step 8: The Administrator confirms to add the new antenna. [alt]	Step 10: The antenna is assigned a unique AntennaID which is set to the next consecutive number after the previous AntennaID in the database and creates the new antenna in Antenna table.
		Step 11: The system notifies the Administrator that the antenna has been added successfully. [alt]
		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted antenna already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new antenna. Return to step 3.	
	Alt Step 11: The antenna was not added successfully, the system will notify the Administrator of the error. Return to step 3.	



Conclusion:	The use case concludes when the Administrator has been notified that the antenna has been added successfully.
Post-Condition:	• A new antenna is added to the system with the relevant details. • The AuditTrail table is successfully updated. • The Antenna table is successfully updated.
Business Rules:	• Only an Administrator can add a antenna to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.4.2 Search Antenna

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Search Antenna	Use Case Type:
Use Case ID:	3.2	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for an antenna. The system will prompt the Administrator to select the antenna search criteria they wish to search by and the Administrator will input the antenna search criteria details. The use case concludes when the system displays the searched antenna details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search an antenna's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for an antenna in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the antenna search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific antenna details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified antenna details from the Antenna table in the database. [alt]
		Step 7: The system displays the following details: • Antenna table – AntennaID – Coordinates – DateInstalled • AntennaStatus table – AntennaStatus
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditTrailID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified antenna does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched antenna details are displayed.	
Post-Condition:	• The antenna has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for an antenna.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The antenna exists in the system.	
Open Issues:	• None	



2.3.4.3 Update Antenna

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Update Antenna	Use Case Type:
Use Case ID:	3.3	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a antenna. The system will prompt the Administrator to input the specified antenna search criteria of the antenna they wish to update. The system will then display the details of the specific antenna for the Administrator to make changes and update the antenna details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the antenna has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update an antenna.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific antenna in the system.	Step 2: The system invokes Use Case 3.2 Search Antenna.
	Step 3: The Administrator inputs the updated user details.	
	Step 4: The Administrator selects to update the user details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]:



		• Antenna Status • Coordinates • DateInstalled
		Step 6: The system prompts the Administrator to confirm the updated antenna details.
	Step 7: The Administrator confirms to update the antenna details. [alt]	Step 8: The system updates the specific antenna in the Antenna table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the antenna details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The antenna was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the antenna was updated successfully.	
Post-Condition:	• The Antenna table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can update an antenna.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The antenna exists in the system.	
Open Issues:	• None	



2.3.4.4 Delete Antenna

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Delete Antenna	Use Case Type:
Use Case ID:	3.4	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific from the system. The system will prompt the Administrator to select the antenna search criteria of the antenna they wish to delete. The Administrator will confirm the deletion of the searched antenna and the system will delete the selected antenna from the database. The use case will conclude when the system has notified the Administrator that the antenna has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific antenna.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific antenna from the system.	Step 2: The system Invokes Use Case 3.1 Search Antenna
	Step 3: The administrator selects to delete the antenna.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the antenna. [alt]	Step 6: The system removes the selected antenna from the Antenna table and updates the table in the database.



		Step 7: The system notifies the administrator that the antenna has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the antenna has been deleted successfully.	
Post-Condition:	• The Antenna table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete an antenna.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The antenna exists in the system.	
Open Issues:	• None	



2.3.4.5 Add Tag

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Add Tag	Use Case Type:
Use Case ID:	3.5	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new tag to the system. The system will prompt the Administrator to input the required tag details. The Administrator will confirm the inputted tag details and the system will add the new tag details to the database. The use case will conclude when the system notifies the Administrator that the tag has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new tag.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new tag.	Step 2: The system displays the add tag details and prompts the Administrator to input the required tag details: - Tag Status - Date installed
	Step3: The Administrator inputs the tag details.	
	Step 4: The Administrator selects to add the new tag details to the system.	Step 5: The system captures and validates the following fields are filled and the



		inputted details are in the correct format [alt]:
		Step 6: The system READS the Tag details from the Tag table in the database and verifies that no duplicate tag exists on the system by comparing the inputted Tag to the Tags in the Tag table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the tag.
	Step 8: The Administrator confirms to add the new tag. [alt]	Step 10: The tag is assigned a unique TagID which is set to the next consecutive number after the previous TagID in the database and creates the new tag in Tag table.
		Step 11: The system notifies the Administrator that the tag has been added successfully. [alt]
		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted tag already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new tag. Return to step 3.	
	Alt Step 11: The tag was not added successfully, the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case concludes when the Administrator has been notified that the tag has been added successfully.	
Post-Condition:	• A new tag is added to the system with the relevant details. • The AuditTrail table is successfully updated. • The Tag table is successfully updated.	



Business Rules:	• Only an Administrator can add a tag to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.4.6 Search Tag

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Search Tag	Use Case Type:
Use Case ID:	3.6	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a tag. The system will prompt the Administrator to select the tag search criteria they wish to search by, and the Administrator will input the tag search criteria details. The use case concludes when the system displays the searched tag details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a tag's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a tag in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the tag search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific tag details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified tag details from the Tag table in the database. [alt]
		Step 7: The system displays the following details: • Tag table – TagID – DateInstalled • TagStatus table – TagStatus
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditTrailID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified tag does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched tag details are displayed.	
Post-Condition:	• The tag has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for a tag.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The tag exists in the system.	
Open Issues:	• None	



2.3.4.7 Update Tag

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Update Tag	Use Case Type:
Use Case ID:	3.7	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a tag. The system will prompt the Administrator to input the specified tag search criteria of the tag they wish to update. The system will then display the details of the specific tag for the Administrator to make changes and update the tag details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the tag has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a tag.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific tag in the system.	Step 2: The system invokes Use Case 3.6 Search Tag.
	Step 3: The Administrator inputs the updated user details.	
	Step 4: The Administrator selects to update the user details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]: • Tag Status



		Date Installed
		Step 6: The system prompts the Administrator to confirm the updated tag details.
	Step 7: The Administrator confirms to update the tag details. [alt]	Step 8: The system updates the specific tag in the Tag table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the tag details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select to confirm the update. Return to Step 3.	
	Alt Step 9: The tag was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the tag was updated successfully.	
Post-Condition:	- The Tag table has been updated and changes saved to the database. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can update a tag.	
Implementation Constraints and Specifications:	None	
Assumptions:	The tag exists in the system.	
Open Issues:	None	



2.3.4.8 Delete Tag

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Delete Tag	Use Case Type:
Use Case ID:	3.8	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific from the system. The system will prompt the Administrator to select the tag search criteria of the tag they wish to delete. The Administrator will confirm the deletion of the searched tag and the system will delete the selected tag from the database. The use case will conclude when the system has notified the Administrator that the tag has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific tag.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific tag from the system.	Step 2: The system Invokes Use Case 3.6 Search Tag
	Step 3: The administrator selects to delete the tag.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the tag. [alt]	Step 6: The system removes the selected tag from the Tag table and updates the table in the database.



		Step 7: The system notifies the administrator that the tag has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the tag has been deleted successfully.	
Post-Condition:	• The Tag table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a tag.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The tag exists in the system.	
Open Issues:	• None	



2.3.5 Sub-system 4: Student

2.3.5.1 Add Student

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Add Student	Use Case Type:
Use Case ID:	4.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new student to the system. The system will prompt the Administrator to input the required student details. The Administrator will confirm the inputted student details and the system will add the new student details to the database. The use case will conclude when the system notifies the Administrator that the student has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new student.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new student.	Step 2: The system displays the add student details and prompts the Administrator to input the required user details: • School Name • Village Name • Guardian Name • Guardian Email • Guardian Phone • Student Name



		• Student Surname • Student Date of Birth • Student Gender • Phone Number
	Step3: The Administrator inputs the student details.	
	Step 4: The Administrator selects to add the new user details to the system.	Step 5: The system captures and validates the following fields are filled and the inputted details are in the correct format [alt]:
		Step 6: The system READS the Student details from the Student table in the database and verifies that no duplicate student exists on the system by comparing the inputted Student Name to StudentName of registered students existing in the Student table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the student.
	Step 8: The Administrator confirms to add the new student. [alt]	Step 10: The student is assigned a unique StudentID which is set to the next consecutive number after the previous StudentID in the database and creates the new student in Student table.
		Step 11: The system notifies the Administrator that the student has been added successfully. [alt]
Alternate Courses:		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted student already exists in the system; the system will notify the Administrator of the error. Return to step 3.	



	Alt Step 8: The Administrator does not confirm they wish to add the new student. Return to step 3.
	Alt Step 11: The student was not added successfully, the system will notify the Administrator of the error. Return to step 3.
Conclusion:	The use case concludes when the Administrator has been notified that the student has been added successfully.
Post-Condition:	• A new student is added to the system with the relevant details. • The AuditTrail table is successfully updated. • The Student table is successfully updated.
Business Rules:	• Only an Administrator can add a student to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.5.2 Search Student

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search Student	Use Case Type:
Use Case ID:	4.2	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a student. The system will prompt the Administrator to select the student search criteria they wish to search by and the Administrator will input the student search criteria details. The use case concludes when the system displays the searched student details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a student's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a student in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the student search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific student details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified student details from the Student table in the database. [alt]
		Step 7: The system displays the following details: • Student table - SchoolName - VillageName - GuardianName - StudentName - StudentSurname - StudentDOB - StudentGender - PhoneNumber • Guardian table - GuardianName - GuardianEmail - GuardianPhone
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditTrailID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified student does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched student details are displayed.	
Post-Condition:	• The student has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for a student.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The student exists in the system.	
Open Issues:	• None	



2.3.5.3 Update Student

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Update Student	Use Case Type:
Use Case ID:	4.3	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a student. The system will prompt the Administrator to input the specified student search criteria of the student they wish to update. The system will then display the details of the specific student for the Administrator to make changes and update the student details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the student has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a student.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific student in the system.	Step 2: The system invokes Use Case 4.3 Search Student.
	Step 3: The Administrator inputs the updated student details.	
	Step 4: The Administrator selects to update the student details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]: School Name



		• Village Name • Guardian Name • Guardian Email • Guardian Phone • Student Name • Student Surname • Student DOB • Student Gender • Phone Number
		Step 6: The system prompts the Administrator to confirm the updated student details.
	Step 7: The Administrator confirms to update the student details. [alt]	Step 8: The system updates the specific student in the Student table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the student details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The student was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the student was updated successfully.	
Post-Condition:	• The Student table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can update a student.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The student exists in the system.	



2.3.5.4 Delete Student

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Delete Student	Use Case Type:
Use Case ID:	4.4	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific student from the system. The system will prompt the Administrator to select the student search criteria of the student they wish to delete. The Administrator will confirm the deletion of the searched student and the system will delete the selected student from the database. The use case will conclude when the system has notified the Administrator that the student has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific student.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific student from the system.	Step 2: The system Invokes Use Case 1.5 Search Student
	Step 3: The administrator selects to delete the student.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the student. [alt]	Step 6: The system removes the selected student from the Student table and updates the table in the database.



		Step 7: The system notifies the administrator that the student has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does not confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the student has been deleted successfully.	
Post-Condition:	• The Student table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a student.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The student exists in the system.	
Open Issues:	• None	



2.3.5.5 Capture Student Marks

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Capture Student Marks	Use Case Type:
Use Case ID:	4.5	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby new Student Marks are entered into the system. The Administrator will select the Capture Student Marks option and provide the required details. The system will save the new Student Marks to the database and display a confirmation message.	
Pre-Condition:	The Administrator must be logged in.	
Trigger:	The Administrator selects the add new Student option.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The administrator requests to add new Student marks to the system.	Step 2: The system loads the Student marks Details.
	Step 3: The Administrator enters the new Student marks.	
	Step 4: The Administrator selects the add Student mark option.	Step 5: The system captures the Student marks.
		Step 6: The system verifies that the information entered is in the correct format and all the relevant fields have been filled [alt]



		Step 7: The system prompts the Administrator for confirmation to add the new Student marks.
	Step 8: The Administrator confirms to add the new Student marks. [alt]	
		Step 9: The new Student marks are saved to the StudentMark table and saved to the database.
		Step 10: Display success message to Administrator. [alt]
		Step 11: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • AuditTrailDescription
Alternate Courses:	Alt Step 6: System validation error, formatting error or required fields have not been entered. Return to Step 3.	
	Alt Step 8: The Administrator does not confirm add. Return to Step 3.	
	Alt Step 10: Student marks were not added successfully, notify Administrator.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the new Student mark has been successfully added.	
Post-Condition:	• The Student mark has been successfully added and the StudentMark table has been updated. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can add a Student marks.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• None	
Open Issues:	• None	



2.3.5.6 Search Student Marks

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search Student Marks	Use Case Type:
Use Case ID:	4.6	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search a Students' Marks. The Administrator will select the criteria they wish to search by and enter the StudentID of the Students who's marks they wish to search. The system will display the requested Students mark report.	
Pre-Condition:	The Administrator is logged in.	
Trigger:	The Administrator requests to search a Students' marks.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to search for a Student marks in the system.	Step 2: The system prompts the Administrator to select the Student criteria they wish to search by and enter the required details for specified criteria.
	Step 3: The Administrator enters the prompted details.	
	Step 4: The Administrator selects the search student marks option. [alt]	Step 5: The system captures the Student details.
		Step 6: The system validates the Students details. [alt]



		Step 7: The system READS the specified Student from the Student table and StudentMark table in the database. [alt]
		Step 8: The system displays the Student details: • Student table - StudentID - StudentName - StudentSurname • Mark table - Date - Mark
		Step 9: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • AuditTrailDescription
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Terminate Use Case.	
	Alt Step 6: System validation error, formatting error or required fields have not been entered. Return to Step 3.	
	Alt Step 7: Student does not exist in the database, notify Administrator. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched Student marks report is displayed.	
Post-Condition:	• The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for a Students' marks.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• None	
Open Issues:	• None	



2.3.5.7 Add Mark Type

System Name:	VVs
Author:	Cameron King
Date:	06/05/2020
Version:	1.0

Use Case Name:	Add Mark Type	Use Case Type:
Use Case ID:	4.7	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new Mark Type to the system. The system will prompt the Administrator to input the required Mark Type details. The Administrator will confirm the inputted Mark Type details and the system will add the new Mark Type details to the database. The use case will conclude when the system notifies the Administrator that the Mark Type has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new Mark Type.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new Mark Type.	Step 2: The system displays the add student details and prompts the Administrator to input the required user details: Mark Type
	Step3: The Administrator inputs the Mark Type details.	
	Step 4: The Administrator selects to add the new user details to the system.	Step 5: The system captures and validates the following fields are filled and the



		inputted details are in the correct format [alt]:
		Step 6: The system READS the Mark Type details from the Mark Type table in the database and verifies that no duplicate Mark Type exists on the system by comparing the inputted Mark Type to Mark Type of registered Mark Types existing in the Mark Type table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the Mark Type.
	Step 8: The Administrator confirms to add the new Mark Type. [alt]	Step 10: The student is assigned a unique MarkTypeID which is set to the next consecutive number after the previous MarkTypeID in the database and creates the new MarkType in MarkType table.
		Step 11: The system notifies the Administrator that the MarkType has been added successfully. [alt]
		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted Mark Type already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new Mark Type. Return to step 3.	
	Alt Step 11: The Mark Type was not added successfully, the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case concludes when the Administrator has been notified that the Mark Type has been added successfully.	
Post-Condition:	• A new Mark Type is added to the system with the relevant details. • The AuditTrail table is successfully updated.	



	• The MarkType table is successfully updated.
Business Rules:	• Only an Administrator can add a student to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.5.8 Search Mark Type

System Name:	VVs
Author:	Cameron King
Date:	06/05/2020
Version:	1.0

Use Case Name:	Search Mark Type	Use Case Type:
Use Case ID:	4.8	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a Mark Type. The system will prompt the Administrator to select the Mark Type search criteria they wish to search by and the Administrator will input the Mark Type search criteria details. The use case concludes when the system displays the searched Mark Type details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a Mark Type's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a Mark Type in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the Mark Type search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific Mark Type details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified Mark Type details from the Mark Type table in the database. [alt]
		Step 7: The system displays the following details: • MarkType table – MarkType
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditTrailID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified Mark Type does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched Mark Type details are displayed.	
Post-Condition:	• The Mark Type has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for a Mark Type.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Mark Type exists in the system.	
Open Issues:	• None	



2.3.5.9 Update Mark Type

System Name:	VVs
Author:	Cameron King
Date:	06/05/2020
Version:	1.0

Use Case Name:	Update Mark Type	Use Case Type:
Use Case ID:	4.9	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a Mark Type. The system will prompt the Administrator to input the specified Mark Type search criteria of the Mark Type they wish to update. The system will then display the details of the specific Mark Type for the Administrator to make changes and update the Mark Type details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the Mark Type has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a Mark Type.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific Mark Type in the system.	Step 2: The system invokes Use Case 4.8 Search Mark Type.
	Step 3: The Administrator inputs the updated Mark Type details.	
	Step 4: The Administrator selects to update the Mark Type details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]:



		Mark Type
		Step 6: The system prompts the Administrator to confirm the updated Mark Type details.
	Step 7: The Administrator confirms to update the Mark Type details. [alt]	Step 8: The system updates the specific Mark Type in the Mark Type table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the Mark Type details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The Mark Type was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the Mark Type was updated successfully.	
Post-Condition:	- The Mark Type table has been updated and changes saved to the database. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can update a Mark Type.	
Implementation Constraints and Specifications:	None	
Assumptions:	The Mark Type exists in the system.	



2.3.5.10 Delete Mark Type

System Name:	VVs
Author:	Cameron King
Date:	06/05/2020
Version:	1.0

Use Case Name:	Delete Mark Type	Use Case Type:
Use Case ID:	4.10	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific Mark Type from the system. The system will prompt the Administrator to select the Mark Type search criteria of the Mark Type they wish to delete. The Administrator will confirm the deletion of the searched Mark Type and the system will delete the selected Mark Type from the database. The use case will conclude when the system has notified the Administrator that the Mark Type has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific Mark Type.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific Mark Type from the system.	Step 2: The system Invokes Use Case 4.8 Search Mark Type
	Step 3: The administrator selects to delete the Mark Type.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the Mark Type. [alt]	Step 6: The system removes the selected Mark Type from the MarkType table and updates the table in the database.



		Step 7: The system notifies the administrator that the Mark Type has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does not confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the Mark Type has been deleted successfully.	
Post-Condition:	• The Mark Type table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a Mark Type.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Mark Type exists in the system.	
Open Issues:	• None	



2.3.6 Sub-system 5: Bicycle

2.3.6.1 Add Bicycle

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Add Bicycle	Use Case Type:
Use Case ID:	5.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new bicycle to the system. The system will prompt the Administrator to input the required bicycle details. The Administrator will confirm the inputted bicycle details and the system will add the new bicycle details to the database. The use case will conclude when the system notifies the Administrator that the bicycle has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new bicycle.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new bicycle.	Step 2: The system displays the add bicycle details and prompts the Administrator to input the required bicycle details: - Bicycle Status - Bicycle Make - Date Added
	Step3: The Administrator inputs the bicycle details.	



	Step 4: The Administrator selects to add the new bicycle details to the system.	Step 5: The system captures and validates the following fields are filled and the inputted details are in the correct format [alt]:
		Step 6: The system READS the Bicycle details from the Bicycle table in the database and verifies that no duplicate bicycle exists on the system by comparing the inputted Bicycle to the Bicycles in the Bicycle table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the bicycle.
	Step 8: The Administrator confirms to add the new bicycle. [alt]	Step 10: The bicycle is assigned a unique BicycleID which is set to the next consecutive number after the previous BicycleID in the database and creates the new bicycle in Bicycle table.
		Step 11: The system notifies the Administrator that the bicycle has been added successfully. [alt]
		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted bicycle already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new bicycle. Return to step 3.	
	Alt Step 11: The bicycle was not added successfully, the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case concludes when the Administrator has been notified that the bicycle has been added successfully.	
Post-Condition:	• A new bicycle is added to the system with the relevant details.	



	• The AuditTrail table is successfully updated. • The Bicycle table is successfully updated.
Business Rules:	• Only an Administrator can add a bicycle to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.6.2 Search Bicycle

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Search Bicycle	Use Case Type:
Use Case ID:	5.2	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a bicycle. The system will prompt the Administrator to select the bicycle search criteria they wish to search by and the Administrator will input the bicycle search criteria details. The use case concludes when the system displays the searched bicycle details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a bicycle's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a bicycle in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the bicycle search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific bicycle details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified bicycle details from the Bicycle table in the database. [alt]
		Step 7: The system displays the following details: • Bicycle table – BicycleID – DateAdded – BicycleMake • BicycleStatus table – BicycleStatus • Tag table – TagID • Student table – StudentID – StudentName – StudentSurname
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditTrailID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified bicycle does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched bicycle details are displayed.	
Post-Condition:	• The bicycle has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for a bicycle.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The bicycle exists in the system.	
Open Issues:	• None	



2.3.6.3 Update Bicycle

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Update Bicycle	Use Case Type:
Use Case ID:	5.3	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a bicycle. The system will prompt the Administrator to input the specified bicycle search criteria of the bicycle they wish to update. The system will then display the details of the specific bicycle for the Administrator to make changes and update the bicycle details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the bicycle has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a bicycle.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific bicycle in the system.	Step 2: The system invokes Use Case 5.2 Search Bicycle.
	Step 3: The Administrator inputs the updated user details.	
	Step 4: The Administrator selects to update the user details.	Step 5: The system captures and validates the required fields are filled and the details are in the correct format [alt]:



		Step 6: The system prompts the Administrator to confirm the updated bicycle details.
	Step 7: The Administrator confirms to update the bicycle details. [alt]	Step 8: The system updates the specific bicycle in the Bicycle table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the bicycle details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The bicycle was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the bicycle was updated successfully.	
Post-Condition:	• The Bicycle table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can update a bicycle.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The bicycle exists in the system.	
Open Issues:	• None	



2.3.6.4 Delete Bicycle

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Delete Bicycle	Use Case Type:
Use Case ID:	5.4	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific from the system. The system will prompt the Administrator to select the bicycle search criteria of the bicycle they wish to delete. The Administrator will confirm the deletion of the searched bicycle and the system will delete the selected bicycle from the database. The use case will conclude when the system has notified the Administrator that the bicycle has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific bicycle.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific bicycle from the system.	Step 2: The system Invokes Use Case 5.2 Search Bicycle
	Step 3: The administrator selects to delete the bicycle.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the bicycle. [alt]	Step 6: The system removes the selected bicycle from the Bicycle table and updates the table in the database.



		Step 7: The system notifies the administrator that the bicycle has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the bicycle has been deleted successfully.	
Post-Condition:	• The Bicycle table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a bicycle.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The bicycle exists in the system.	
Open Issues:	• None	



2.3.6.5 Assign a Bicycle to Student

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Assign a Bicycle to Student	Use Case Type:
Use Case ID:	5.5	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	Student (ERA)	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an administrator assigns a bicycle to a student. The system notifies the administrator once successfully assigned	
Pre-Condition:	The Administrator must be logged in.	
Trigger:	The Administrator requests to assign a bicycle to a student.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The administrator requests to Assign a bicycle to a student	Step 2: The system invokes use case 4.2 Search student and displays a list of all the students have not been assigned a bicycle.
	Step 3: The administrator selects the student they wish to assign a bicycle to. [alt]	Step 4: The system will READ the BicycleStatus details from the BicycleStatus table to retrieve a list of bicycles with a BicycleStatus to "available".
		Step 5: The system will display the available bicycle details and prompt the Administrator to select the bicycle they wish to assign to the student.
	Step 6: The Administrator selects the bicycle.	Step 7: The system prompts the Administrator to confirm the assign bicycle action.



	Step 8: The Administrator confirms the assign bicycle action. [alt]	Step 9: The system will assign the bicycle to the student and updates the following tables in the database: - Bicycle table - BicycleStatusID - StudentID - BicycleStudent table - BicycleStudentID - BicycleID - StudentID - DataAssigned - BicycleStatus table - BicycleStatus - Student table - BicycleID
		Step 10: The system notifies the administrator that the assign action was successful. [alt]
		Step 11: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 3: The Administrator cancels the assignment. Terminate use case.	
	Alt Step 8: The Administrator declines the bicycle unassign action. Terminate the Use Case.	
	Alt Step 10: The system notifies the Administrator that the assign was unsuccessful. Return to step 1.	
Conclusion:	The use case narrative concludes when the administrator is notified that the Bicycle has been successfully assigned to a student	
Post-Condition:	- The Bicycle, BicycleStudent, BicycleStatus and Student tables have been updated and changes saved to the database. - The AuditTrail table has been successfully updated.	
Business Rules:	Only an administrator can assign a bike to a student	
Implementation Constraints and Specifications:	None	
Assumptions:	- The Bicycle already exists on the system - The Student already exists on the system	
Open Issues:	None	



2.3.6.6 Unassign Bicycle from student

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Unassign Bicycle from Student	Use Case Type:
Use Case ID:	5.6	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	Student (ERA)	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an administrator Unassigning a bicycle from a student. The system notifies the administrator once successfully Unassigned	
Pre-Condition:	The Administrator must be logged in.	
Trigger:	The Administrator requests to Unassign a bicycle	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The administrator requests to Unassign a bicycle from a student	Step 2: The system invokes use case 4.2 Search student and displays a list of all the students with a bicycle assigned to them.
	Step 3: The administrator selects the Student they wish to unassign. [alt]	Step 4: The system prompts the Administrator to confirm the bicycle unassign action.
	Step 5: The Administrator confirms the bicycle unassign action. [alt]	Step 6: The system unassigns bicycle from the relevant student and updates the following tables: - Bicycle table - BicycleStatusID - StudentID - BicycleStudent table



		- BicycleStudentID - BicycleID - StudentID - DataUnassigned • BicycleStatus table - BicycleStatus • Student table - BicycleID
		Step 7: The system notifies the administrator that the Unassign was successful and the student gets a notification. [alt]
		Step 8: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 3: The administrator cancels the Unassign. Terminate use case.	
	Alt Step 7: The system notifies the administrator that the Unassign was unsuccessful. Return to step 1.	
Conclusion:	The use case narrative concludes when the administrator is notified that the Bicycle has been successfully Unassigned from a student	
Post-Condition:	• The Bicycle table and Student table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an administrator can Unassign a bike from a student.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Bicycle already exists on the system. • The student already exists on the system.	
Open Issues:	• None	



2.3.6.7 Assign Tag to bicycle

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Assign Tag to bicycle	Use Case Type:
Use Case ID:	5.7	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor		
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an administrator assigns a Tag to a Bicycle. The system then notifies the administrator once successfully assigned	
Pre-Condition:	The Administrator must be logged in.	
Trigger:	The Administrator requests to assign a Tag	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The administrator requests to assign a tag to a bicycle.	Step 2: The system invokes use case 5.2 Search bicycle and displays a list of all the bicycles that have not been assigned tag.
	Step 3: The administrator selects the bicycle they wish to assign a tag to. [alt]	Step 4: The system will READ the TagStatus details from the TagStatus table to retrieve a list of tags with a TagStatus set to "available".
		Step 5: The system will display the available tags list and prompt the Administrator to select the tag they wish to assign to the bicycle.



	Step 6: The Administrator selects the tag.	Step 7: The system prompts the Administrator to confirm the assign tag action.
	Step 8: The Administrator confirms the assign tag action. [alt]	Step 9: The system will assign the tag to the bicycle and updates the following tables in the database: - Bicycle table - TagID - StudentID - Tag table - TagStatusID - DateInstalled - StudentID - DataAssigned - TagStatus table - TagStatus
		Step 10: The system notifies the administrator that the assign action was successful. [alt]
		Step 11: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 3: The administrator cancels the assignment. Terminate the use case.	
	Alt Step 8: The Administrator declines the tag unassign action. Terminate the Use Case.	
	Alt Step 10: The system notifies the administrator that the assign was unsuccessful. Return to step 1.	
Conclusion:	The use case narrative concludes when the administrator is notified that the tag has been successfully assigned to a bicycle.	
Post-Condition:	- The Bicycle, Tag, TagStatus table have been updated and changes saved to the database. - The AuditTrail table has been successfully updated.	
Business Rules:	Only an administrator can assign a bicycle to a student.	
Implementation Constraints and Specifications:	None	
Assumptions:	- The Bicycle already exists on the system. - The Tag already exists on the system.	
Open Issues:	None	



2.3.6.8 Unassign Tag from Bicycle

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Unassign Tag from Bicycle	Use Case Type:
Use Case ID:	5.8	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an administrator Unassigning a Tag from a bicycle. The system notifies the administrator once successfully Unassigned	
Pre-Condition:	The Administrator must be logged in.	
Trigger:	The Administrator requests to Unassign a bicycle	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The administrator requests to unassign a tag from a bicycle.	Step 2: The system invokes use case 5.2 Search Bicycle and displays a list of all the bicycles without a tag assigned to them.
	Step 3: The administrator selects the Bicycle they wish to unassign. [alt]	Step 4: The system prompts the Administrator to confirm the tag unassign action.
	Step 5: The Administrator confirms the tag unassign action. [alt]	Step 6: The system unassigns tag from the relevant bicycle and updates the following tables: - Bicycle table - TagID - StudentID - Tag table



		- TagStatusID - DateInstalled - StudentID - DataAssigned • TagStatus table - TagStatus
		Step 7: The system notifies the administrator that the unassign action was successful. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 3: The administrator cancels the Unassign action. Terminate use case.	
	Alt Step 6: The system notifies the administrator that the unassign was unsuccessful. Return to step 1.	
Conclusion:	The use case narrative concludes when the administrator is notified that the Tag has been successfully unassigned from the bicycle	
Post-Condition:	• The Bicycle table and Tag table have been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an administrator can Unassign a Tag from a Bicycle	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Bicycle already exists on the system • The Tag already exists on the system	
Open Issues:	• None	



2.3.7 Sub-system 6: Bicycle Part

2.3.7.1 Add Bicycle Part

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Add Bicycle Part	Use Case Type:
Use Case ID:	6.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new Part to the system. The system will prompt the Administrator to input the required Part details. The Administrator will confirm the inputted Part details and the system will add the new Part details to the database. The use case will conclude when the system notifies the Administrator that the Part has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new Part.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new Part.	Step 2: The system displays the add Part details and prompts the Administrator to input the required Part details: - PartName - PartDescription
	Step3: The Administrator inputs the Part details.	



	Step 4: The Administrator selects to add the new Part details to the system.	Step 5: The system captures and validates the following fields are filled and the inputted details are in the correct format [alt]:
		Step 6: The system READS the Part details from the Part table in the database and verifies that no duplicate Part exists on the system by comparing the inputted PartName to PartName of registered Parts existing in the Part table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the Part.
	Step 8: The Administrator confirms to add the new Part. [alt]	Step 10: The Part is assigned a unique PartID which is set to the next consecutive number after the previous PartID in the database and creates the new Part in Part table.
		Step 11: The system notifies the Administrator that the Part has been added successfully. [alt]
		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted Part already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new Part. Return to step 3.	
	Alt Step 11: The Part was not added successfully, the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case concludes when the Administrator has been notified that the Part has been added successfully.	
Post-Condition:	• A new Part is added to the system with the relevant details.	



	• The AuditTrail table is successfully updated. • The Part table is successfully updated.
Business Rules:	• Only an Administrator can add a Part to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.7.2 Search Bicycle Part

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Search Bicycle Part	Use Case Type:
Use Case ID:	6.2	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a Part. The system will prompt the Administrator to select the Part search criteria they wish to search by, and the Administrator will input the Part search criteria details. The use case concludes when the system displays the searched Part details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a Part's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a Part in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the Bicycle Part search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific Part details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified user details from the Part table in the database. [alt]
		Step 7: The system displays the following details: • Part table – PartName – PartDescription • PartType table – PartType
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditTrailID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified Part does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched Part details are displayed.	
Post-Condition:	• The Part has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for a Part.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Part exists in the system.	
Open Issues:	• None	



2.3.7.3 Update Bicycle Part

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Update Bicycle Part	Use Case Type:
Use Case ID:	6.3	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a Part. The system will prompt the Administrator to input the specified Part search criteria of the Part they wish to update. The system will then display the details of the specific Part for the Administrator to make changes and update the Part details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the Part has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a Part.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific Part in the system.	Step 2: The system invokes Use Case 6.2 Search Part.
	Step 3: The Administrator inputs the updated Part details.	
	Step 4: The Administrator selects to update the Part details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]: • PartName



		PartDescription
		Step 6: The system prompts the Administrator to confirm the updated Part details.
	Step 7: The Administrator confirms to update the Part details. [alt]	Step 8: The system updates the specific Part in the Part table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the Part details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The Part was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the Part was updated successfully.	
Post-Condition:	- The Part table has been updated and changes saved to the database. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can update a Part.	
Implementation Constraints and Specifications:	None	
Assumptions:	The Part exists in the system.	
Open Issues:	None	



2.3.7.4 Delete Bicycle Part

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Delete Bicycle Part	Use Case Type:
Use Case ID:	6.4	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific Part from the system. The system will prompt the Administrator to select the Part search criteria of the Part they wish to delete. The Administrator will confirm the deletion of the searched Part and the system will delete the selected Part from the database. The use case will conclude when the system has notified the Administrator that the Part has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific Part.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific Part from the system.	Step 2: The system Invokes Use Case 6.2 Search Part
	Step 3: The administrator selects to delete the Part.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the Part. [alt]	Step 6: The system removes the selected Part from the Part table and updates the table in the database.



		Step 7: The system notifies the administrator that the Part has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does not confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the Part has been deleted successfully.	
Post-Condition:	• The Part table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a Part.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Part exists in the system.	
Open Issues:	• None	



2.3.7.5 Add Bicycle Section

System Name:	VVs
Author:	Gareth Phillips
Date:	09/04/2020
Version:	1.0

Use Case Name:	Add Bicycle Section	Use Case Type:
Use Case ID:	6.5	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new Section to the system. The system will prompt the Administrator to input the required Section details. The Administrator will confirm the inputted Section details and the system will add the new Section details to the database. The use case will conclude when the system notifies the Administrator that the Section has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new Section.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new Section.	Step 2: The system displays the add user details and prompts the Administrator to input the required Section details: • SectionName • SectionDescription
	Step3: The Administrator inputs the section details.	
	Step 4: The Administrator selects to add the new Section details to the system.	Step 5: The system captures and validates the following fields are filled and the



		inputted details are in the correct format [alt]:
		Step 6: The system READS the Section details from the Section table in the database and verifies that no duplicate Section exists on the system by comparing the inputted SectionName to SectionName of registered Sections existing in the Section table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the Section.
	Step 8: The Administrator confirms to add the new Section. [alt]	Step 10: The Section is assigned a unique SectionID which is set to the next consecutive number after the previous SectionID in the database and creates the new Section in Section table.
		Step 11: The system notifies the Administrator that the Section has been added successfully. [alt]
		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted Section already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new Section. Return to step 3.	
	Alt Step 11: The Section was not added successfully, the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case concludes when the Administrator has been notified that the Section has been added successfully.	
Post-Condition:	• A new Section is added to the system with the relevant details. • The AuditTrail table is successfully updated.	



	• The Section table is successfully updated.
Business Rules:	• Only an Administrator can add a Section to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.7.6 Search Bicycle Section

System Name:	VVs
Author:	Gareth Phillips
Date:	09/04/2020
Version:	1.0

Use Case Name:	Search Bicycle Section	Use Case Type:
Use Case ID:	6.6	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a Section. The system will prompt the Administrator to select the Section search criteria they wish to search by, and the Administrator will input the Section search criteria details. The use case concludes when the system displays the searched Section details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a Section's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a Section in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the Section search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific Section details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified Section details from the Section table in the database. [alt]
		Step 7: The system displays the following details: • Section table – SectionName – SectionDescription
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditTrailID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified Section does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched Section details are displayed.	
Post-Condition:	• The Section has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for a Section.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Section exists in the system.	
Open Issues:	• None	



2.3.7.7 Update Bicycle Section

System Name:	VVs
Author:	Gareth Phillips
Date:	09/04/2020
Version:	1.0

Use Case Name:	Update Bicycle Section	Use Case Type:
Use Case ID:	6.7	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a Section. The system will prompt the Administrator to input the specified Section search criteria of the Section they wish to update. The system will then display the details of the specific Section for the Administrator to make changes and update the Section details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the Section has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a Section.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific Section in the system.	Step 2: The system invokes Use Case 6.6 Search Section.
	Step 3: The Administrator inputs the updated Section details.	
	Step 4: The Administrator selects to update the Section details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]:



		• SectionName • SectionDescription
		Step 6: The system prompts the Administrator to confirm the updated Section details.
	Step 7: The Administrator confirms to update the Section details. [alt]	Step 8: The system updates the specific Section in the Section table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the Section details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The Section was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the Section was updated successfully.	
Post-Condition:	• The Section table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can update a Section.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Section exists in the system.	
Open Issues:	• None	



2.3.7.8 Delete Bicycle Section

System Name:	VVs
Author:	Gareth Phillips
Date:	09/04/2020
Version:	1.0

Use Case Name:	Delete Bicycle Section	Use Case Type:
Use Case ID:	6.8	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific Section from the system. The system will prompt the Administrator to select the Section search criteria of the Section they wish to delete. The Administrator will confirm the deletion of the searched Section and the system will delete the selected Section from the database. The use case will conclude when the system has notified the Administrator that the Section has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific Section.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific Section from the system.	Step 2: The system Invokes Use Case 6.6 Search Section
	Step 3: The administrator selects to delete the Section.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the Section. [alt]	Step 6: The system removes the selected Section from the Section table and updates the table in the database.



		Step 7: The system notifies the administrator that the Section has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does not confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the Section has been deleted successfully.	
Post-Condition:	• The Section table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a Section.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Section exists in the system.	
Open Issues:	• None	



2.3.8 Sub-system 7: Tracking

2.3.8.1 Add Log Entry

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Add Log Entry	Use Case Type:
Use Case ID:	7.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Student	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Administrator	
Description:	This use case narrative describes the event whereby a student passes through an area which is within the antenna range which in turn creates an entry in the log. The use case concludes when the log entry has been successfully added.	
Pre-Condition:	The tag is registered on the system.	
Trigger:	The tag is in range of the antenna	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The student passes through an area which is within the antenna area with their tag.	Step 2: Antenna captures the tags details: - TagID - AntennaID - DateTime
		Step 3: The system captures and saves the tag details to the Log table. [alt]
		Step 4: The system creates and saves and entry in the AuditTrail table with the following details: AuditID



		• DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 3: Tag read error, Notify Admin. Terminate the Use case.	
Conclusion:	The use case narrative concludes once a log entry has been successfully saved to the database.	
Post-Condition:	• A new log entry has been successfully added to the system. • The AuditTrail table is successfully updated. • The Log table is successfully updated.	
Business Rules:	• Tag must be in the range of the antenna.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The antenna and system are connected.	
Open Issues:	• None	



2.3.8.2 Search Log Entry

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search Log entry	Use Case Type:
Use Case ID:	7.2	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby a Administrator requests to search for a specific log entry in the system. The Administrator will select the specific criteria they wish to search by, and then enter the required details. The use case will conclude when the system has successfully displayed the specific Log entry.	
Pre-Condition:	The Administrator is logged in.	
Trigger:	Administrator wants to search for an entry.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to search for a Log entry in the system.	Step 2: The system prompts the Administrator to select the Log entry search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects the search log details. [alt]	Step 5: The system captures the Log entry search details.
		Step 6: The system validates the log entry details. [alt]



		Step 7: The system READS the Log entry from the Log table in the database. [alt]
		Step 8: The system displays the Log details: • LogID • AntennaID • TagID • DateTime
		Step 9: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • Description
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to Step 3.	
	Alt Step 6: System validation error, formatting error or required fields have not been entered. Return to Step 3.	
	Alt Step 7: Log entry does not exist in the database, notify Administrator. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched log entry details are displayed.	
Post-Condition:	• The log entry has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only a student can search for a specific log entry.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• None	
Open Issues:	• None	



2.3.9 Sub-system 8: Job Tasks

2.3.9.1 Add Fault Task

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Add Fault Task	Use Case Type:
Use Case ID:	8.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new fault task to the system. The system will prompt the Administrator to input the required fault task details. The Administrator will confirm the inputted fault task details and the system will add the new fault task details to the database. The use case will conclude when the system notifies the Administrator that the fault task has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new fault task.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new fault task to the system.	Step 2: The system displays the add fault task details and prompts the Administrator to input the required Fault task details: - Fault Description - Parts needed
	Step3: The administrator inputs the fault task details.	



	Step 4: The Administrator selects to add the new fault task details to the system.	Step 5: The system captures and validates the following fields are filled and the inputted details are in the correct format [alt]: - Fault task Description - Parts needed
		Step 6: The system READS the FaultTask details from the FaultTask table in the database and verifies that no duplicate fault task exists on the system. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the fault task.
	Step 8: The Administrator confirms to add the new fault task. [alt]	Step 10: The fault task is assigned a unique FaultTaskID which is set to the next consecutive number after the previous FaultTaskID in the database and saves the fault task details to the FaultTask table.
		Step 11: The system notifies the Administrator that the fault task has been added successfully. [alt]
		Step 12: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted fault task already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new fault task. Return to step 3.	
	Alt Step 11: The fault task was not added successfully, the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case concludes when the Administrator has been notified that the fault task has been added successfully.	



Post-Condition:	• A new fault task is added to the system with the relevant details. • The AuditTrail table is successfully updated. • The FaultTask table is successfully updated.
Business Rules:	• Only a System Administrator can add a fault task to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.9.2 Search Fault Task

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search Fault Task	Use Case Type:
Use Case ID:	8.2	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a fault task. The system will prompt the Administrator to select the fault task search criteria they wish to search by and the Administrator will input the fault task search criteria details. The use case concludes when the system displays the searched fault task details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a fault task's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search a fault task in the system.	Step 2: The system displays the Fault task search criteria details and prompts the Administrator to select the fault task search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific fault task details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified fault task details from the FaultTask table in the database. [alt]
		Step 7: The system displays the fault task details: • FaultTaskID • FaultTaskDescription • PartsNeeded
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditTrailID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified fault task does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched fault task details are displayed.	
Post-Condition:	• The fault task has been searched successfully. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can search for a fault task.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The fault task exists in the system.	
Open Issues:	• None	



2.3.9.3 Update Fault Task

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Update Fault Task	Use Case Type:
Use Case ID:	8.3	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a fault task. The system will prompt the Administrator to input the specified fault task search criteria of the fault task they wish to update. The system will then display the details of the specific fault task for the Administrator to make changes and update the fault task details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the fault task has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a fault task.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific fault task in the system.	Step 2: The system invokes Use Case 8.2 Search Fault Task.
	Step 3: The Administrator inputs the updated fault task details.	
	Step 4: The Administrator selects to update the fault task details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]:



		- Fault task Description - Parts needed
		Step 6: The system prompts the Administrator to confirm the updated fault task details.
	Step 7: The Administrator confirms to update the fault task details. [alt]	Step 8: The system updates the specific fault task in the FaultTask Table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the fault task details have been updated successfully. [alt]
		Step 10: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The fault task was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the fault task was updated successfully.	
Post-Condition:	- The FaultTask table has been updated and changes saved to the database. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can update a fault task.	
Implementation Constraints and Specifications:	None	
Assumptions:	The fault task exists in the system.	
Open Issues:	None	



2.3.9.4 Delete Fault Task

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Delete Fault Task	Use Case Type:
Use Case ID:	8.4	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific fault task from the system. The system will prompt the Administrator to select the fault task search criteria of the fault task they wish to delete. The Administrator will confirm the deletion of the searched fault task and the system will delete the selected fault task from the database. The use case will conclude when the system has notified the Administrator that the fault task has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific fault task.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific fault task from the system.	Step 2: The system Invokes Use Case 8.2 Search Fault Task
	Step 3: The administrator selects to delete the fault task.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the fault task. [alt]	Step 6: The system removes the selected fault task from the FaultTask table and updates the table in the database.



		Step 7: The system notifies the administrator that the fault task has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the fault task has been deleted successfully.	
Post-Condition:	• The FaultTask table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a fault task.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The fault task exists in the system	
Open Issues:	• None	



2.3.9.5 Add Maintenance Task

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Add Maintenance Task	Use Case Type:
Use Case ID:	8.5	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new maintenance task to the system. The system will prompt the Administrator to input the required maintenance task details. The Administrator will confirm the inputted maintenance task details and the system will add the new maintenance task details to the database. The use case will conclude when the system notifies the Administrator that the maintenance task has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new maintenance task.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new maintenance task.	Step 2: The system displays the add maintenance task details and prompts the Administrator to input the required maintenance task details: - MaintenanceTaskDescription - Frequency
	Step3: The Administrator inputs the maintenance task details.	



	Step 4: The Administrator selects to add the new maintenance task details to the system.	Step 5: The system captures and validates the following fields are filled and the inputted details are in the correct format [alt]:
		Step 6: The system READS the maintenance task details from the MaintenanceTask table in the database and verifies that no duplicate maintenance task exists on the system by comparing the inputted Maintenance task to the Maintenance tasks in the MaintenanceTask table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the.
	Step 8: The Administrator confirms to add the new maintenance task. [alt]	Step 10: The maintenance task is assigned a unique MaintenanceTaskID which is set to the next consecutive number after the previous MaintenanceTaskID in the database and creates the new maintenance task in MaintenanceTask table.
		Step 11: The system notifies the Administrator that the maintenance task has been added successfully. [alt]
		Step 12: The system creates and saves and entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted maintenance task already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new maintenance task. Return to step 3.	
	Alt Step 11: The maintenance task was not added successfully, the system will notify the Administrator of the error. Return to step 3.	



Conclusion:	The use case concludes when the Administrator has been notified that the maintenance task has been added successfully.
Post-Condition:	• A new maintenance task is added to the system with the relevant details. • The AuditTrail table is successfully updated. • The MaintenanceTask table is successfully updated.
Business Rules:	• Only an Administrator can add a maintenance task to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.9.6 Search Maintenance Task

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search Maintenance Task	Use Case Type:
Use Case ID:	8.6	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a maintenance task. The system will prompt the Administrator to select the maintenance task search criteria they wish to search by, and the Administrator will input the maintenance task search criteria details. The use case concludes when the system displays the searched maintenance task details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a maintenance task's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a maintenance task in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the maintenance task search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	



	Step 4: The Administrator selects to search for the specific maintenance task details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]
		Step 6: The system READS the specified maintenance task details from the MaintenanceTask table in the database. [alt]
		Step 7: The system displays the following details: - MaintenanceTask table - MaintenanceTaskID - MaintenaceTaskDescription - MaintenanceGroup table - MaintenanceGroupID - Frequency
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: - AuditTrailID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified maintenance task does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched maintenance task details are displayed.	
Post-Condition:	- The maintenance task has been searched successfully. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can search for a maintenance task.	
Implementation Constraints and Specifications:	None	
Assumptions:	The maintenance task exists in the system.	
Open Issues:	None	



2.3.9.7 Update Maintenance Task

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Update Maintenance Task	Use Case Type:
Use Case ID:	8.7	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a maintenance task. The system will prompt the Administrator to input the specified maintenance task search criteria of the maintenance task they wish to update. The system will then display the details of the specific maintenance task for the Administrator to make changes and update the maintenance task details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the maintenance task has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a maintenance task.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific maintenance task in the system.	Step 2: The system invokes Use Case 8.6 Search Maintenance task.
	Step 3: The Administrator inputs the updated user details.	
	Step 4: The Administrator selects to update the user details.	Step 5: The system captures and validates the required fields are filled and the details are in the correct format [alt]:



		Step 6: The system prompts the Administrator to confirm the updated maintenance task details.
	Step 7: The Administrator confirms to update the maintenance task details. [alt]	Step 8: The system updates the specific maintenance task in the MaintenanceTask table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the maintenance task details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The maintenance task was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the maintenance task was updated successfully.	
Post-Condition:	- The MaintenanceTask table has been updated and changes saved to the database. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can update a maintenance task.	
Implementation Constraints and Specifications:	None	
Assumptions:	The maintenance task exists in the system.	
Open Issues:	None	



2.3.9.8 Delete Maintenance Task

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Delete Maintenance Task	Use Case Type:
Use Case ID:	8.8	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific from the system. The system will prompt the Administrator to select the maintenance task search criteria of the maintenance task they wish to delete. The Administrator will confirm the deletion of the searched maintenance task and the system will delete the selected from the database. The use case will conclude when the system has notified the Administrator that the maintenance task has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific maintenance task.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific maintenance task from the system.	Step 2: The system Invokes Use Case 8.6 Search Maintenance Task
	Step 3: The administrator selects to delete the maintenance task.	Step 4: The system prompts the administrator for confirmation of the delete action.



	Step 5: The administrator confirms the deletion of the maintenance task. [alt]	Step 6: The system removes the selected maintenance task from the MaintenanceTask table and updates the table in the database.
		Step 7: The system notifies the administrator that the maintenance task has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the maintenance task has been deleted successfully.	
Post-Condition:	• The MaintenanceTask table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a maintenance task.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The maintenance task exists in the system.	
Open Issues:	• None	



2.3.10 Sub-system 9: School

2.3.10.1 Add School

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Add School	Use Case Type:
Use Case ID:	9.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new school to the system. The system will prompt the Administrator to input the required school details. The Administrator will confirm the inputted school details and the system will add the new school details to the database. The use case will conclude when the system notifies the Administrator that the school has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new school.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new school.	Step 2: The system displays the add school details and prompts the Administrator to input the required school details: • School Name • School Telephone • School Email • Village Name



	Step3: The Administrator inputs the school details.	
	Step 4: The Administrator selects to add the new school details to the system.	Step 5: The system captures and validates the following fields are filled and the inputted details are in the correct format [alt]:
		Step 6: The system READS the school details from the School table in the database and verifies that no duplicate school exists on the system by comparing the inputted School to the Schools in the School table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the.
	Step 8: The Administrator confirms to add the new school. [alt]	Step 10: The school is assigned a unique SchoolID which is set to the next consecutive number after the previous SchoolID in the database and creates the new school in School table.
		Step 11: The system notifies the Administrator that the school has been added successfully. [alt]
		Step 12: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted school already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new school. Return to step 3.	
	Alt Step 11: The school was not added successfully, the system will notify the Administrator of the error. Return to step 3.	



Conclusion:	The use case concludes when the Administrator has been notified that the school has been added successfully.
Post-Condition:	• A new school is added to the system with the relevant details. • The AuditTrail table is successfully updated. • The School table is successfully updated.
Business Rules:	• Only an Administrator can add a school to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.10.2 Search School

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search School	Use Case Type:
Use Case ID:	9.2	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a school. The system will prompt the Administrator to select the school search criteria they wish to search by, and the Administrator will input the school search criteria details. The use case concludes when the system displays the searched school details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a school's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a school in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the school search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific school details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified school details from the School table in the database. [alt]
		Step 7: The system displays the following details: - School table - SchoolID - SchoolName - SchoolEmail - SchoolTelephone - Village table - VillageID - VillageName
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: - AuditTrailID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified school does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched school details are displayed.	
Post-Condition:	- The school has been searched successfully. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can search for a school.	
Implementation Constraints and Specifications:	None	
Assumptions:	The school exists in the system.	
Open Issues:	None	



2.3.10.3 Update School

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Update School	Use Case Type:
Use Case ID:	9.3	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a school. The system will prompt the Administrator to input the specified school search criteria of the school they wish to update. The system will then display the details of the specific school for the Administrator to make changes and update the school details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the school has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a school.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific school in the system.	Step 2: The system invokes Use Case 9.2 Search School.
	Step 3: The Administrator inputs the updated user details.	
	Step 4: The Administrator selects to update the user details.	Step 5: The system captures and validates the required fields are filled and the details are in the correct format [alt]:



		Step 6: The system prompts the Administrator to confirm the updated details.
	Step 7: The Administrator confirms to update the school details. [alt]	Step 8: The system updates the specific school in the School table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the school details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The school was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the school was updated successfully.	
Post-Condition:	• The School table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can update a school.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The school exists in the system.	
Open Issues:	• None	



2.3.10.4 Delete School

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Delete School	Use Case Type:
Use Case ID:	9.4	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific from the system. The system will prompt the Administrator to select the school search criteria of the school they wish to delete. The Administrator will confirm the deletion of the searched school and the system will delete the selected from the database. The use case will conclude when the system has notified the Administrator that the school has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific school.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific school from the system.	Step 2: The system Invokes Use Case 9.2 Search School
	Step 3: The administrator selects to delete the school.	Step 4: The system prompts the administrator for confirmation of the delete action.
	Step 5: The administrator confirms the deletion of the school. [alt]	Step 6: The system removes the selected school from the School table and updates the table in the database.



		Step 7: The system notifies the administrator that the school has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the school has been deleted successfully.	
Post-Condition:	• The School table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a school.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The school exists in the system.	
Open Issues:	• None	



2.3.11 Sub-system 10: Job

2.3.11.1 Report Fault

System Name:	VVs
Author:	Byron Donald
Date:	13/04/2020
Version:	1.0

Use Case Name:	Report Fault	Use Case Type:
Use Case ID:	10.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Student	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby a student reporting the faults that apply to the bicycle, can create a job entry on the system. The student requests to report a fault on the system. The system lists all the registered students in the school's proximity. The student will then select their credentials and verify them before the system displays a bicycle blueprint. The student will then make use of the blueprint to identify the problems after which the system creates a job entry and notifies the student that it was reported successfully.	
Pre-Condition:	The student reports the fault at the school	
Trigger:	The student selects to report a fault.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The student selects the report fault option	Step 2: The system identifies the School where the request is made.
		Step 3: The system READS from the Student table and lists all the students that have the corresponding SchoolID



		Step 4: Display List of Students in the specific School.
	Step 5: The student selects their credentials from the list.	Step 6: The system prompts the student to confirm their selection.
	Step 7: The student verifies their credentials and selects to confirm. [alt]	Step 8: The system captures the BicycleID that is registered to the student
		Step 9: The system displays a blueprint of a bicycle that identifies the different parts of a bicycle that the student can select from.
	Step 10: The student selects the part of the bicycle where their problem is located.	Step 11: The system READS all the entries from the Task (FaultTask) table where the PartID Corresponds to the part the student selected
		Step 12: The system displays the retrieved entries for the student to choose from.
	Step 13: The Student selects the task that appears identical to their problem.	Step 14: The system captures the student's selection.
		Step 15: The system prompts the student to confirm their selection.
	Step 16: The student confirms their selection. [alt]	Step 17: The system requests if the student would like to submit the job or if they have any other faults to report.
	Step 18: The student is done and selects to submit. [alt]	Step 19: The system creates a new Job entry and assigns it a unique JobID which is set to the next consecutive number after the previous JobID. The new entry also includes the following details: - BicycleID - Date - SchoolID - Notes(nullable) - TaskID (all that are relevant)
		Step 20: The system displays a successfully submitted notification.



		Step 21: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 7: The credentials are incorrect, and the student selects to go back. Return to step 3.	
	Alt Step 16: The student selected incorrectly, selects the back option. Return to step 9.	
	Alt Step 18: The student decides to report another fault. The system saves the current selection and repeats Step 9 to 17 until they are ready to submit.	
Conclusion:	The student's faults are reported, and a new Job entry is created.	
Post-Condition:	• The job is ready to be assigned. • Audit Trail is successfully updated.	
Business Rules:	• Only registered students can report a fault • Only registered bikes can be reported	
Implementation Constraints and Specifications:	• Students can only report at school because they do not have personal devices	
Assumptions:	• The student knows how to interact with the GUI.	
Open Issues:	• None	



2.3.11.2 View Jobs

System Name:	VVs
Author:	Byron Donald
Date:	13/04/2020
Version:	1.0

Use Case Name:	View Jobs	Use Case Type:
Use Case ID:	10.2	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Mechanic	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby the Mechanic requests to view all jobs meeting certain criteria. The Mechanic will initiate the search after which the system will provide the different search/filter criteria available for the Mechanic to choose from and fill in. Then the system will then display the relevant jobs.	
Pre-Condition:	The mechanic has jobs	
Trigger:	The Mechanic requests to view jobs in the system.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Mechanic requests to view jobs in the system.	Step 2: The system provides different search/filter criteria.
	Step 3: The Mechanic inputs the criteria details.	
	Step 4: The Mechanic initiates the search. [alt]	Step 5: The system captures the criteria details.
		Step 6: The system validates the inputted details. [alt]



		Step 7: The system READS all the jobs that meet the criteria from the Jobs table in the database. [alt]
		Step 8: System invokes Use Case 2.2 View Job Tasks.
		Step 9: The system displays a list of the details for the job and the task required: • BicycleID • Date • StudentID • Notes(nullable) • ReportLineID (all that are relevant)
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The Mechanic doesn't initiate the search. Return to Step 3.	
	Alt Step 6: System validation error, formatting error or required fields Have not been entered. Return to Step 3	
	Alt Step 7: No entries exist in the database that meet the criteria. Return to Step 3.	
Conclusion:	All relevant jobs have been displayed.	
Post-Condition:	• The jobs have been searched successfully.	
Business Rules:	• Only authorized users can view jobs.	
Implementation Constraints and Specifications:	• The mechanics literacy level is a constraint as we need to design the system in such a way that a person with a low literacy level can use the system	
Assumptions:	• The mechanic will be trained to use the system	
Open Issues:	• None	



2.3.11.3 Check-In Bicycle

System Name:	VVs
Author:	Byron Donald
Date:	13/04/2020
Version:	1.0

Use Case Name:	Check-in Bicycle	Use Case Type:
Use Case ID:	10.3	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Student	
Primary System Actor	Mechanic	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby a student notifies the mechanic of the arrival of their Bicycle. The student arrives and drops off their bicycle at the mechanic. The mechanic receives the bicycle and verifies that it appears on the schedule and the student's details match. The mechanic then marks the bicycle as arrived for a job.	
Pre-Condition:	- The student has a bicycle that is registered to the student with them physically. - The mechanic is logged into the system and has their schedule open.	
Trigger:	The Student arrives with their Bicycle.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Student arrives with their Bicycle.	Step 2: The mechanic requests the name and surname of the student.
	Step 3: The Student provides their details.	Step 4: The mechanic verifies the students' details and the BicycleID on the schedule. [alt]
		Step 5: The mechanic invokes use case 10.2 View Jobs and assigns the Received status.



		Step 6: The mechanic dismisses the student.
Alternate Courses:	Alt Step 4: The details provided do not appear on the schedule. Mechanic Informs the student and dismisses them.	
Conclusion:	The mechanic has the bicycle and has updated the relevant job to Received.	
Post-Condition:	• The Bicycle is ready to be repaired.	
Business Rules:	• The bicycle must be registered to a valid student and the details much match.	
Implementation Constraints and Specifications:	• Students must physically check in their bicycles	
Assumptions:	• Students check in their bicycles at school on a specific date	
Open Issues:	• None	



2.3.11.4 Report Maintenance

System Name:	VVs
Author:	Byron Donald
Date:	13/04/2020
Version:	1.0

Use Case Name:	Report Maintenance	Use Case Type:
Use Case ID:	10.4	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Mechanic	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Administrator	
Description:	This use case narrative describes the event whereby the system automatically updates the details of the maintenance task that was completed and in turn will calculate when the next maintenance for the bike is due.	
Pre-Condition:	- There is already a job entry for each maintenance task waiting for completion. - Mechanic is logged into the system	
Trigger:	The mechanic completes a maintenance task job.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The mechanic completes a maintenance task job.	Step 2: The system captures the details of the job and maintenance task that was completed.
		Step 3: The system invokes use case 8.6 using the TaskID That was captured from the job. [alt]
		Step 4: The system captures the details of the MaintenanceTask retrieved.
		Step 5: The system uses the - Frequency - Date



		Variables from the MaintenanceTask & Job tables to calculate the next date based on the frequency and captures it.
		Step 6: The system creates a new Job entry and assigns it a unique JobID which is set to the next consecutive number after the previous JobID. The new entry also includes the following details that originate from the previous job: • BicycleID • TaskID • SchoolID the following that was calculated: • Date the following that was reset to false: • Assigned Then saves all the details in the Job table
		Step 7: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 3: The task was not found. The system notifies the Mechanic and sends all the current information before the process is terminated.	
Conclusion:	The next maintenance task job is created and stored successfully.	
Post-Condition:	• The job is ready to be assigned. • Audit Trail is successfully updated.	
Business Rules:	• Bikes need to be checked in for maintenance	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The maintenance tasks have been properly tracked since the purchase of the bicycle. • The first set of maintenance tasks are dated from purchase.	
Open Issues:	• None	



2.3.12 Sub-system 11: Reporting

2.3.12.1 Generate Maintenance Schedule

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Generate Maintenance Schedule	Use Case Type:
Use Case ID:	11.1 (PBA)	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Time	
Primary System Actor	None	
Other Participating Actors:	Mechanic	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby time triggers the generation of a maintenance schedule within certain criteria.	
Pre-Condition:	There must be pending activities for a particular mechanic to carry out.	
Trigger:	After a certain period, passes, the system is prompted to generate a maintenance schedule.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: After a certain period in time passes, the system is prompted to generate a maintenance schedule.	Step 2: The system READS the pre-set duration of the schedule (e.g. weekly) using Period from the BusinessRule Table
		Step 3: The system READS data from the following related table to populate the maintenance schedule. - School Table: - SchoolName - Student Table:



		- StudentName - Surname - Telephone • Bicycle Table: - BicycleID • Section Table: - SectionID • Part Table: - PartID • MaintenanceTask Table: - Description
		Step 4: The system generates the maintenance schedule. [alt].
	With what? What must I do? It looks fine to me lol	Step 5: The system sends the maintenance schedule report via email to the mechanic.
		Step 6: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: There was an error in generating the maintenance schedule. Notify Administrator via email. Terminate Use-Case.	
Conclusion:	The use case narrative concludes when the generated Maintenance Schedule is successfully displayed/printed.	
Post-Condition:	• The generated maintenance schedule remains fixed until the next set is generated. • The AuditTrail table is updated.	
Business Rules:	• Only time can trigger the generation of this maintenance schedule.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The mechanic will be able to view the maintenance schedule, the same as the administrator.	
Open Issues:	• None	



2.3.12.2 Generate Maintenance Requirements Report

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Generate Maintenance Requirements Report	Use Case Type:
Use Case ID:	11.2 (PBA)	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Time	
Primary System Actor	None	
Other Participating Actors:	Mechanic	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby time triggers the generation of a maintenance requirements report for the mechanic that is linked to the maintenance schedule for that particular period.	
Pre-Condition:	- There must be pending maintenance activities for a particular mechanic to carry out before generating any requirements report - A maintenance schedule must have been generated.	
Trigger:	Periodically, the system is prompted to generate a maintenance requirement report for that period's maintenance schedule.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: Periodically, the system is prompted to generate a maintenance requirement report for that period's maintenance schedule.	Step 2: The system READS the pre-set duration of the schedule (e.g. weekly) using Period from the BusinessRule Table
		Step 3: The system invokes Use Case 11.1 Generate Maintenance Schedule to extract the relevant requirements.



		Step 4: The system reads data from the following related table to populate the maintenance requirements report linked to the maintenance schedule. - School Table: - SchoolName - Section Table: - SectionID - Part Table: - PartDescription To have a detail of the parts and tools needed for the activities in that maintenance round.
		Step 5: The system generates the maintenance requirements report. [alt].
		Step 6: The system sends the maintenance requirements report via email to the mechanic.
		Step 7: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 5: There was an error in generating the maintenance requirements report. Notify Administrator via email. Terminate Use-Case.	
Conclusion:	The use case narrative concludes when the generated Maintenance Requirements Report is successfully displayed/printed.	
Post-Condition:	- The generated report remains fixed until the next report is generated. - The AuditTrail table is updated	
Business Rules:	Only time can trigger the generation of this maintenance requirements report.	
Implementation Constraints and Specifications:	None	
Assumptions:	The Mechanic will be able to view the maintenance schedule, the same as the administrator.	
Open Issues:	None	



2.3.12.3 Generate Student Report

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Generate Student Report	Use Case Type:
Use Case ID:	11.3	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator (PBA)	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the series of events that take place when a system administrator requests to generate a student report showing the progress in bicycle usage and academic progress.	
Pre-Condition:	The Administrator must be registered and logged in on the system.	
Trigger:	The Administrator selects the Generate Student Report item.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator selects the Generate Student Report item.	Step 2: The system captures the StudentID of the particular student who has been selected.
		Step 3: The system uses the StudentID to extract the following data linked to that student: - Student Table: - StudentName - StudentSurname - StudentDoB - StudentGender - School Table: - SchoolName



		• Bicycle Table: – BicycleID • Tag Table: – TagID • StudentAttendance Table: – AttendanceID – Date • StudentMark Table: – Date – Mark
		Step 4: The system generates a report for that particular student. [alt]
		Step 5: The system displays a success message and asks if the administrator would like to view the report.
	Step 6: The Administrator selects to "View". [alt]	Step 7: The system displays/prints the student report.
	Step 8: The Administrator selects to "Export". [alt]	Step 9: The system exports the report to pdf. [alt]
		Step 10: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 4: The system encounters an error in Report Generation. Display an error message to the Administrator. Return to Step 1.	
	Alt Step 6: The Administrator selects "Cancel". Terminate Use Case.	
	Alt Step 8: The Administrator selects "Cancel" on the report. Terminate Use Case.	
	Alt Step 9: Exporting error. Display an error message to the Administrator. Return to Step 7.	
Conclusion:	The use case narrative concludes when the Administrator successfully views or exports the Student Report.	
Post-Condition:	• The exported report gets saved in the chosen location by the Administrator. • The AuditTrail table is updated	
Business Rules:	• Only the Administrator must be able to generate the Student Report.	



Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.12.4 Generate Bicycle History Report

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Generate Bicycle History Report	Use Case Type:
Use Case ID:	11.4	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator (PBA)	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the series of events that take place when a system administrator requests to generate a bicycle history report showing the maintenance history, faults reported, repairs completed, as well as users.	
Pre-Condition:	The Administrator must be registered and logged in on the system.	
Trigger:	The Administrator selects the Generate Bicycle History Report item.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator selects the Generate Bicycle History Report item.	Step 2: The system captures the BicycleID of the particular bicycle which has been selected.
		Step 3: The system uses the BicycleID to extract the following data linked to that bicycle: • Student Table: – StudentID • School Table: – SchoolName



		• Tag Table: – TagID • BicycleStatus Table: – BicycleStatus • MechanicJob Table: – JobID – Date • MechanicJobStatus Table: – MechanicJobStatus • JobTask Table: – JobID • Task Table: – TaskDescription • FaultTask Table: – FaultTaskID • MaintenanceTask Table: – MaintenanceTaskID
		Step 4: The system generates a history report for that particular bicycle. [alt]
		Step 5: The system displays a success message and asks if the administrator would like to view the history report.
	Step 6: The Administrator selects to "View". [alt]	Step 7: The system displays/prints the bicycle history report.
	Step 8: The Administrator selects on "Export". [alt]	Step 9: The system exports the report to pdf. [alt]
		Step 10: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • Description
Alternate Courses:	Alt Step 4: The system encounters an error in Report Generation. Display an error message to the Administrator. Return to Step 1.	
	Alt Step 6: The Administrator selects "Cancel". Terminate Use Case.	
	Alt Step 8: The Administrator selects "Cancel" on the report. Terminate Use Case.	
	Alt Step 9: Exporting error. Display an error message to the Administrator. Return to Step 7.	



Conclusion:	The use case narrative concludes when the Administrator successfully views or exports the Bicycle History Report.
Post-Condition:	• The exported report gets saved in the chosen location by the Administrator. • The AuditTrail table is updated.
Business Rules:	• Only the Administrator must be able to generate the Bicycle History Report.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.12.5 Generate Log Report

System Name:	VVs
Author:	Tapiwarufaro Ndoro
Date:	09/04/2020
Version:	1.0

Use Case Name:	Generate Log Report	Use Case Type:
Use Case ID:	11.5	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator (PBA)	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative outlines the course of events that take place when an Administrator prompts the system to generate a Log report of the Antenna activity as per specified criteria	
Pre-Condition:	The Administrator must be registered and logged in on the system.	
Trigger:	The Administrator selects the Generate Log Report item.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator selects the Generate Log Report item.	Step 2: The system prompts the Administrator to specify the criteria for this Log report.
	Step 3: The Administrator picks the criteria from select lists.	Step 4: The system captures the specified Criteria.
		Step 5: The system uses the specified criteria to extract the following data: - Antenna Table: - AntennaID - Coordinates - AntennaStatus Table: - AntennaStatus



		- Log Table: - LogID - DateTime - Bicycle Table: - BicycleID - Student Table: - StudentID
		Step 6: The system generates a Log report for that given criteria. [alt]
		Step 7: The system displays a success message and asks if the administrator would like to view the history report.
	Step 8: The administrator selects "View". [alt]	Step 9: The system displays/prints the bicycle history report.
	Step 10: The administrator selects "Export". [alt]	Step 11: The system exports the report to pdf. [alt]
		Step 12: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 6: The system encounters an error in Report Generation which then displays an error message to the Administrator. Return to Step 3.	
	Alt Step 8: The Administrator selects "Cancel". Terminate Use Case.	
	Alt Step 10: The Administrator selects "Cancel" on the report. Terminate Use Case.	
	Alt Step 11: Exporting error. Display an error message to the Administrator. Return to Step 9.	
Conclusion:	The use case narrative concludes when the Administrator successfully views or exports the Log Report.	
Post-Condition:	- The exported report gets saved in the chosen location by the Administrator. - The AuditTrail table is updated.	
Business Rules:	Only the Administrator must be able to generate the Log Report.	
Implementation Constraints and Specifications:	None	
Assumptions:	None	
Open Issues:	None	



2.3.12.6 Generate Parts Used Report

System Name:	VVs
Author:	Hansen Li
Date:	07/05/2020
Version:	1.0

Use Case Name:	Generate Parts Used Report	Use Case Type:
Use Case ID:	11.6	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Time (PBA)	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	Mechanic & Administrator	
Description:	This use case narrative describes the event whereby time triggers the generation of a Parts Used Report which will show all the parts that were used that week for their respective jobs as well as display the names and descriptions.	
Pre-Condition:	Parts that have been used that week must have been recorded in the system	
Trigger:	After a certain period, passes, the system is prompted to generate a Parts Used Report.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: After a certain period in time passes, the system is prompted to generate a Parts Used report.	Step 2: The system READS the pre-set duration of the schedule (e.g. weekly).
		Step 3: The system READS data from the following related table to populate the maintenance schedule. - Part Table: - PartName - PartDescription



		- Section Table: - SectionDescription
		Step 4: The system generates the Parts Used Report. [alt].
		Step 5: The system displays/prints the Parts Used Report for the Mechanic.
		Step 6: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 4: There was an error in generating the Parts Used Report. Notify Administrator via email. Terminate Use-Case.	
Conclusion:	The use case narrative concludes when the generated Parts Used Report is successfully displayed/printed.	
Post-Condition:	- The Parts Used Report remains fixed until the next set is generated. - The AuditTrail table is updated.	
Business Rules:	Only time can trigger the generation of this parts used report.	
Implementation Constraints and Specifications:	None	
Assumptions:	The mechanic will be able to view the parts used report, the same as the Administrator.	
Open Issues:	None	



2.3.13 Sub-system 12: Notification

2.3.13.1 Send Student Did Not Arrive Notification

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Send student did not arrive notification	Use Case Type:
Use Case ID:	12.1	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Time (PBA)	
Primary System Actor	None	
Other Participating Actors:	Parent (ERA)	
Other Interested Stakeholders:	None	
Description:	This use case describes how a notification must be sent to the parents if a child doesn't arrive at school	
Pre-Condition:	N/A	
Trigger:	It is school time and the Child hasn't arrived at school	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The student has not arrived for school.	Step 2: The system READs data from the following table to identify if a student has not arrived by checking the DateTime. - Log Table - LogID - DateTime
		Step 3: The system READs data from the following tables to populate the notification which will be sent to the guardians. - Bicycle Table: - BicycleID



		• Student Table: - StudentName - StudentSurname • Guardian Table: - GuardianName - GuardianPhoneNumber
		Step 4: The system generates the notification
		Step 5: The system sends an SMS to the student's parents with the following details: [alt] • The time and date the SMS was sent • The name of the student
		Step 6: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The SMS was not sent successfully; the system will notify the Administrator. Return to Step 1.	
Conclusion:	The use case narrative concludes when the SMS is sent to the parents	
Post-Condition:	• The system has sent an SMS to the parents	
Business Rules:	• The SMS must only be sent out if the child hasn't arrived	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The student's parent's contact number is in the system.	
Open Issues:	• None	



2.3.13.2 Send Student Maintenance Notification

System Name:	VVs
Author:	Hansen Li
Date:	10/04/2020
Version:	1.0

Use Case Name:	Send student Maintenance Notification	Use Case Type:
Use Case ID:	12.2	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Time (PBA)	
Primary System Actor	None	
Other Participating Actors:	Student (ERA)	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the notification being sent to the student when their maintenance is due.	
Pre-Condition:	The student has booked in their bike for maintenance	
Trigger:	A week before the maintenance date has arrived	
Typical Course Of Events:	Actor Action	System Response
	Step 1: A week before the maintenance date has arrived	Step 2: The system READS data from the following tables. - Student Table: - StudentName - StudentPhoneNumber - ScheduledMaintenance Table: - CheckInDate
		Step 3: The system generates the maintenance notification
		Step 4: The system sends an SMS to the student with the following details:



		The date and time their maintenance is due. [alt]
		Step 5: The system creates and saves an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 4: The system failed to send, return to step 1.	
Conclusion:	The use case narrative concludes when the system has sent the SMS to the student	
Post-Condition:	An SMS has been sent out from the system	
Business Rules:	The SMS must only be sent out a week before	
Implementation Constraints and Specifications:	None	
Assumptions:	The student already exists on the system	
Open Issues:	None	



2.3.14 Sub-system 13: Administration

2.3.14.1 Backup Data

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Backup Data	Use Case Type:
Use Case ID:	13.1	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to back up the systems database. Following a successful backup, all the data in the system will have a restore point that will be created so that the database can be rolled back.	
Pre-Condition:	- The Administrator is logged in. - The database is in a valid state.	
Trigger:	The Administrator wants to back up the data in the database.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to back up the database. [alt]	Step 2: The system will prompt the Administrator to confirm the database backup request.
	Step 3: The Administrator confirms the system backup.	Step 4: The system makes a direct copy of the database and creates a restore point. The backup is saved in the Backup file.
		Step 5: The system notifies the Administrator that the backup was successful. [alt]



		Step 6: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 1: The backup request is initiated by Time. Continue to Step 4.	
	Alt Step 5: The system notifies the Administrator was unsuccessful. Return to Step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the database has been successfully backed up.	
Post-Condition:	• The database has been successfully backed up.	
Business Rules:	• Only an Administrator can back up the database. • The system will be automatically backed up weekly.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• None	
Open Issues:	• None	



2.3.14.2 Restore Data

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	Restore Data	Use Case Type:
Use Case ID:	13.2	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to restore system data from a backup file. The Administrator will select the Backfile and will be notified when the backup has been completed.	
Pre-Condition:	- The Administrator must be logged in. - A Backup file must exist.	
Trigger:	The Administrator requests to Restore the database data.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to Restore system data from a Backup file.	Step 2: The system will prompt the Administrator for confirmation of the system restore.
	Step 3: The Administrator confirms they want to continue with the system restoration. [alt]	Step 4: The system prompts the Administrator to select the Backup file they wish to restore from.
	Step 5: The Administrator selects the Backup file and selects the continue option. [alt]	Step 6: The system performs the system restore.



		Step 7: The system notifies the Administrator that the system restore was successful. [alt]
		Step 8: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 3: The Administrator selects the Cancel Restore option. Terminate Use Case.	
	Alt Step 5: The Administrator does not select a Backup file. Terminate Use Case.	
	Alt Step 6: The system notifies the Administrator the system restore was unsuccessful. Return to Step 5.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the system has been successfully restored.	
Post-Condition:	The system has been successfully restored or rolled-back.	
Business Rules:	• Only the Administrator can perform a system restore.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• None	
Open Issues:	• None	



2.3.14.3 View Audit Trail Log

System Name:	VVs
Author:	Cameron King
Date:	05/04/2020
Version:	1.0

Use Case Name:	View Audit Trail Log	Use Case Type:
Use Case ID:	13.3	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to view the Audit Trail log. The system will display a list of all the Audit Trail entries in chronological order.	
Pre-Condition:	The Administrator is logged in.	
Trigger:	The Administrator requests to view the Audit Trail log.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to view the Audit Trail log.	Step 2: The system READS the Audit Trail details from the AuditTrail table in the database.
		Step 3: The system displays a list of Audit Trail log entries. [alt]
		Step 4: The system creates and save an entry in the AuditTrail table with the following details: - AuditID - DateTime - TransactionDescription



		• UserID
Alternate Courses:	Alt Step 3: The system cannot display the Audit Trail log. Terminate Use Case.	
Conclusion:	The use case narrative concludes when system displays a list of the Audit Trail log entries.	
Post-Condition:	• The Audit Trail log is displayed.	
Business Rules:	• Only the administrator is allowed to view the Audit Trail. • Every transaction in the system is tracked.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• None	
Open Issues:	• None	



2.3.14.4 Search Audit Trail Entry

System Name:	VVs
Author:	Cameron King
Date:	10/04/2020
Version:	1.0

Use Case Name:	Search Audit Trail Entry	Use Case Type:
Use Case ID:	13.4	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search a specific Audit Trail entry. The Administrator will select the criteria they wish to search by, and the system will display the Audit Trail entry according to the criteria.	
Pre-Condition:	The Administrator is logged in.	
Trigger:	The Administrator requests to search a specific Audit Trail entry.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to search a specific Audit Trail entry.	Step 2: The system prompts the Administrator to enter a keyword to search for.
	Step 3: The Administrator enters the keyword.	
	Step 4: The Administrator selects to search a specific Audit Trail entry.	Step 5: The system captures the inputted Audit Trail entry search criteria.
		Step 6: The system READS the Audit Trail entry from the AuditTrail table in the database based on the keyword.



		Step 7: The system displays the Audit Trail entry. [alt]
		Step 8: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • AuditTrailDescription • UserID
Alternate Courses:	Alt Step 7: Audit Trail not found notification. Return to Step 2.	
Conclusion:	The use case narrative concludes when the system displays the Audit Trail entry.	
Post-Condition:	• Audit Trail log entry is displayed.	
Business Rules:	• Only the administrator is allowed to view Audit Trail entry.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• None	
Open Issues:	• None	



2.3.14.5 Add Route

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Add Route	Use Case Type:
Use Case ID:	13.5	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to add a new Route to the system. The system will prompt the Administrator to input the required Route details. The Administrator will confirm the inputted Route details and the system will add the new Route details to the database. The use case will conclude when the system notifies the Administrator that the Route has been added successfully.	
Pre-Condition:	The Administrator must be logged into the system.	
Trigger:	The Administrator requests to add a new Route.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to add a new Route.	Step 2: The system displays the add Route details and prompts the Administrator to input the required Route details: • AntennaID • Position
	Step3: The Administrator inputs the Route details.	
	Step 4: The Administrator selects to add the new Route details to the system.	Step 5: The system captures and validates the following fields are filled and the



		inputted details are in the correct format [alt]:
		Step 6: The system READS the Route details from the Route table in the database and verifies that no duplicate Route exists on the system by comparing the inputted RouteID to RouteID of registered Routes existing in the Route table. [alt]
		Step 7: The system prompts the Administrator to confirm the addition of the Route.
	Step 8: The Administrator confirms to add the new Route. [alt]	Step 10: The Route is assigned a unique RouteID which is set to the next consecutive number after the previous RouteID in the database and creates the new Route in Route table.
		Step 11: The system notifies the Administrator that the Route has been added successfully. [alt]
		Step 12: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The system detects a formatting error, or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The inputted Route already exists in the system; the system will notify the Administrator of the error. Return to step 3.	
	Alt Step 8: The Administrator does not confirm they wish to add the new Route. Return to step 3.	
	Alt Step 11: The Route was not added successfully, the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case concludes when the Administrator has been notified that the Route has been added successfully.	
Post-Condition:	• A new Route is added to the system with the relevant details. • The AuditTrail table is successfully updated.	



	• The Route table is successfully updated.
Business Rules:	• Only an Administrator can add a Route to the system.
Implementation Constraints and Specifications:	• None
Assumptions:	• None
Open Issues:	• None



2.3.14.6 Search Route

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Search Route	Use Case Type:
Use Case ID:	13.6	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to search for a Route. The system will prompt the Administrator to select the Route search criteria they wish to search by and the Administrator will input the Route search criteria details. The use case concludes when the system displays the searched Route details.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to search a Route's details.	
Typical Course Of Events	Actor Action	System Response
	Step 1: The Administrator requests to search for a Route in the system.	Step 2: The system displays the search criteria details prompts the Administrator to select the Route search criteria and input the required details for the specified criteria.
	Step 3: The Administrator inputs the prompted search criteria details.	
	Step 4: The Administrator selects to search for the specific Route details. [alt]	Step 5: The system captures and validates the search criteria field is filled and the details are in the correct format. [alt]



		Step 6: The system READS the specified Route details from the Route table in the database. [alt]
		Step 7: The system displays the following details: - Route table - AntennaID - Position
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: - AuditTrailID - DateTime - TransactionDescription - UserID
Alternate Courses:	Alt Step 4: The Administrator does not confirm the search. Return to step 1.	
	Alt Step 5: System incurred a validation error; required fields have not been filled or details are in the incorrect format. The system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 6: The specified Route does not exist in the database; the system will notify the Administrator of the error. Return to Step 3.	
Conclusion:	The use case narrative concludes when the searched Route details are displayed.	
Post-Condition:	- The Route has been searched successfully. - The AuditTrail table has been successfully updated.	
Business Rules:	Only the Administrator can search for a Route.	
Implementation Constraints and Specifications:	None	
Assumptions:	The Route exists in the system.	
Open Issues:	None	



2.3.14.7 Update Route

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Update Route	Use Case Type:
Use Case ID:	13.7	Business Requirements: ☐
Priority:	Medium	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to update a Route. The system will prompt the Administrator to input the specified Route search criteria of the Route they wish to update. The system will then display the details of the specific Route for the Administrator to make changes and update the Route details. The Administrator will confirm the updated details and the system will update the database. The use case will conclude when the Administrator is notified that the Route has been updated successfully.	
Pre-Condition:	The Administrator is logged into the system.	
Trigger:	The Administrator requests to update a Route.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to update a specific Route in the system.	Step 2: The system invokes Use Case 13.6 Search Route.
	Step 3: The Administrator inputs the updated Route details.	
	Step 4: The Administrator selects to update the Route details.	Step 5: The system captures and validates the following fields are filled and the details are in the correct format [alt]:



		• AntennaID • Position
		Step 6: The system prompts the Administrator to confirm the updated Route details.
	Step 7: The Administrator confirms to update the Route details. [alt]	Step 8: The system updates the specific Route in the Route table and saves the updated details to the database.
		Step 9: The system notifies the Administrator that the Route details have been updated successfully. [alt]
		Step 10: The system creates and save an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternative Courses:	Alt Step 5: The system detects a formatting error or the required fields have not been filled, the system will notify the Administrator of the error. Return to Step 3.	
	Alt Step 7: The Administrator does not select the confirm update option. Return to Step 3.	
	Alt Step 9: The Route was not updated successfully; the system will notify the Administrator of the error. Return to step 3.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the Route was updated successfully.	
Post-Condition:	• The Route table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only the Administrator can update a Route.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Route exists in the system.	
Open Issues:	• None	



2.3.14.8 Delete Route

System Name:	VVs
Author:	Gareth Phillips
Date:	05/04/2020
Version:	1.0

Use Case Name:	Delete Route	Use Case Type:
Use Case ID:	13.8	Business Requirements: ☐
Priority:	Low	System Analysis: ☒
Source:	VVs User Requirements	System Design: ☐
Primary Business Actor	Administrator	
Primary System Actor	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case narrative describes the event whereby an Administrator requests to delete a specific Route from the system. The system will prompt the Administrator to select the Route search criteria of the Route they wish to delete. The Administrator will confirm the deletion of the searched Route and the system will delete the selected Route from the database. The use case will conclude when the system has notified the Administrator that the Route has been deleted successfully.	
Pre-Condition:	The Administrator must be logged into system.	
Trigger:	The Administrator requests to delete a specific Route.	
Typical Course Of Events:	Actor Action	System Response
	Step 1: The Administrator requests to delete a specific Route from the system.	Step 2: The system Invokes Use Case 13.6 Search Route
	Step 3: The Administrator selects to delete the Route.	Step 4: The system prompts the Administrator for confirmation of the delete action.
	Step 5: The Administrator confirms the deletion of the Route. [alt]	Step 6: The system removes the selected Route from the Route table and updates the table in the database.



		Step 7: The system notifies the Administrator that the Route has been deleted successfully. [alt]
		Step 8: The system creates and saves an entry in the AuditTrail table with the following details: • AuditID • DateTime • TransactionDescription • UserID
Alternate Courses:	Alt Step 5: The Administrator does not confirm the deletion. Terminate the Use Case.	
	Alt Step 7: The deletion was not successful; the system will notify the Administrator of the error. Return to step 1.	
Conclusion:	The use case narrative concludes when the Administrator is notified that the Route has been deleted successfully.	
Post-Condition:	• The Route table has been updated and changes saved to the database. • The AuditTrail table has been successfully updated.	
Business Rules:	• Only an Administrator can delete a Route.	
Implementation Constraints and Specifications:	• None	
Assumptions:	• The Route exists in the system.	
Open Issues:	• None	



2.3.15 Conclusion

In conclusion we were able to gain a lot more insight into each of our processes by listing out each step in such a way that it is easily understandable. We also made various changes to our use case diagram after identifying some needed changes from the narratives.



3. Process Models

3.1 Introduction

Process models can vary in terms of amount of information shown largely depending on what type of diagram is used. Process models show all the inputs and outputs of a process which means that it gets all its information from the use case narratives. Process models are a useful way to identify all the individual flows of data and information in and out of the system.



3.2 Context Diagram

3.2.1 Introduction

The context diagram is the highest level of data flow diagrams and basically shows all the flows in and out of the system as a whole. Due to it having to show so much data the context diagram does not include data stores or invokes.



3.2.2 Diagram

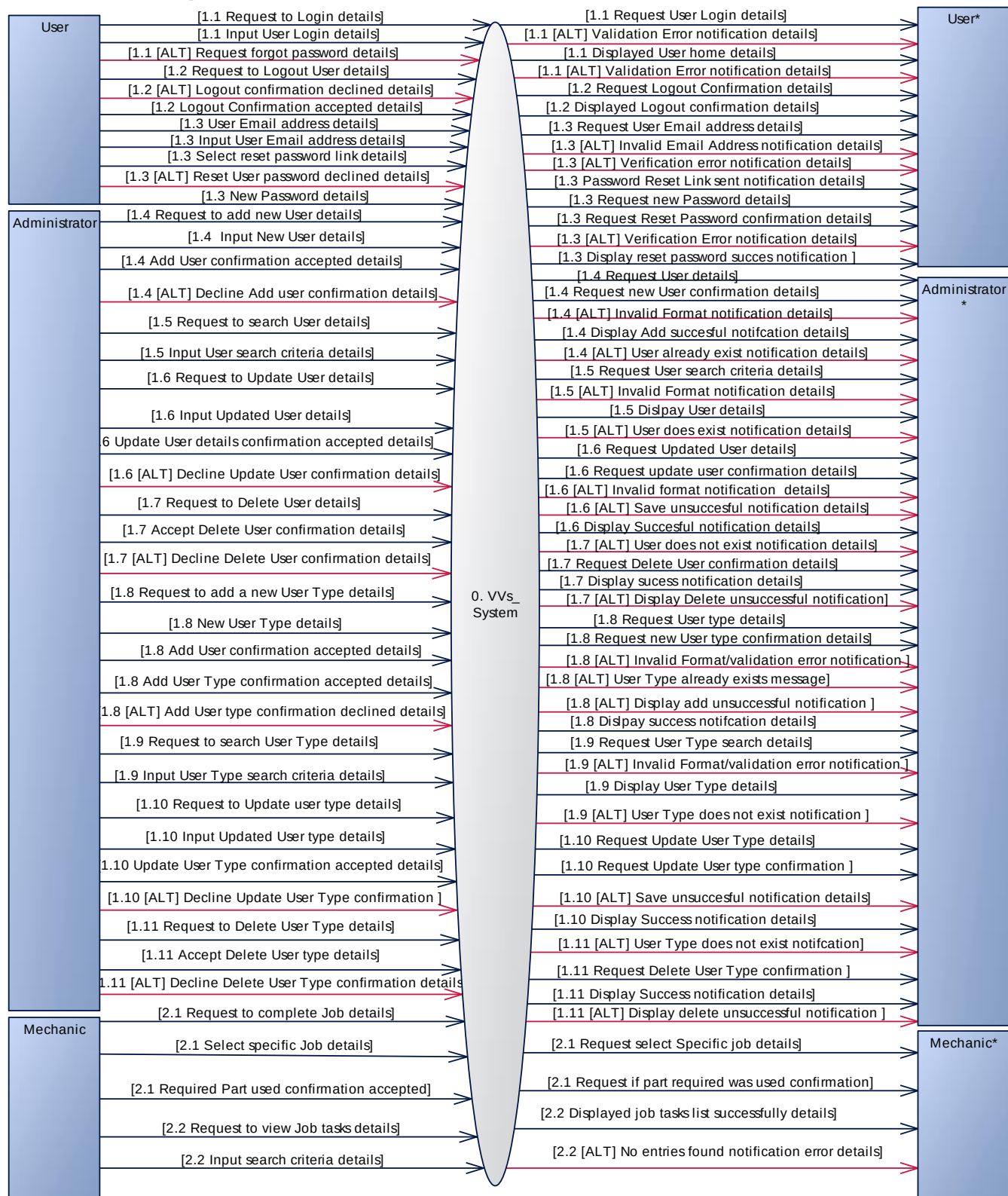


Figure 14 Context Diagram Sub-system 1 and 2

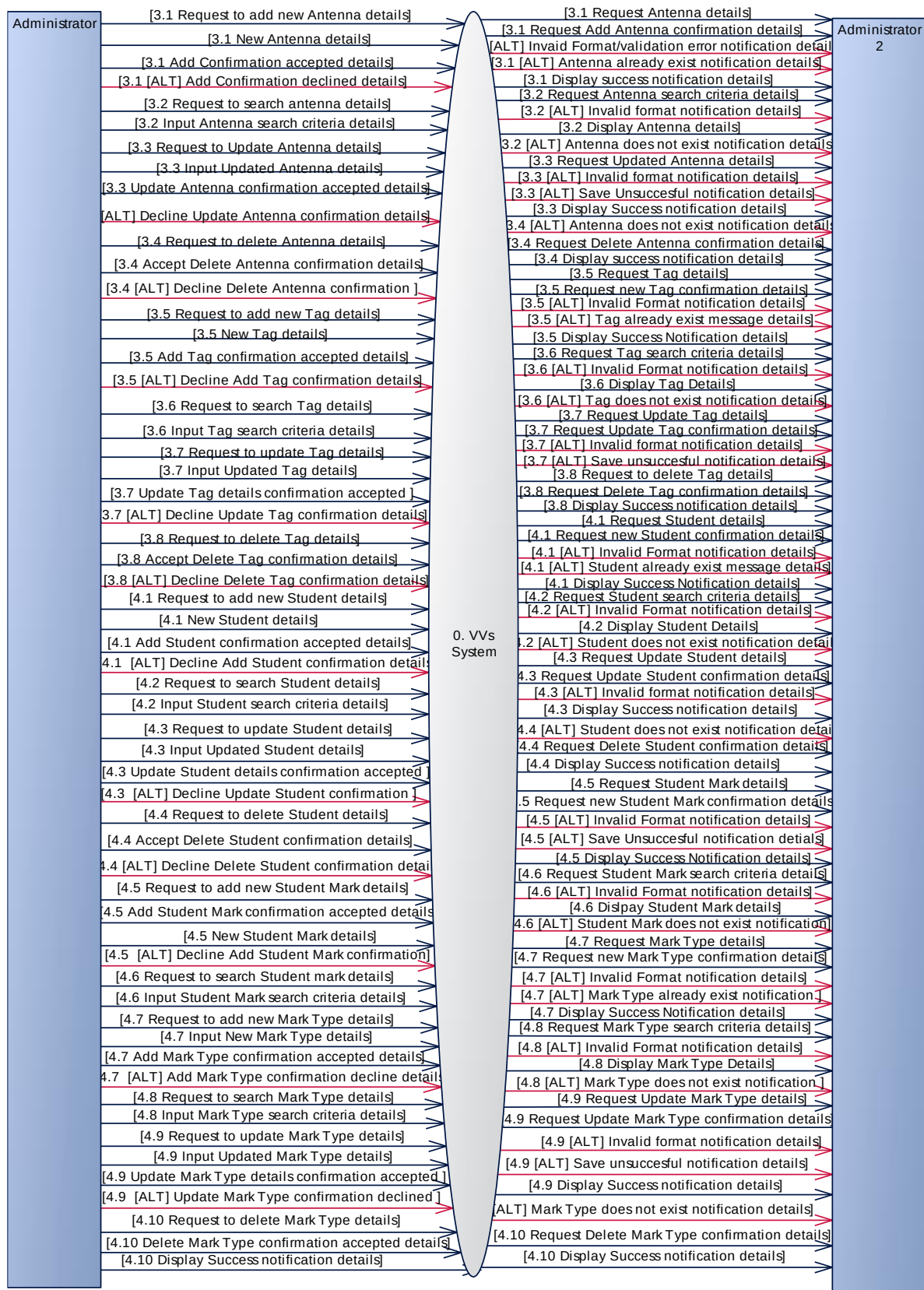


Figure 15 Context Diagram Sub-system 3 and 4

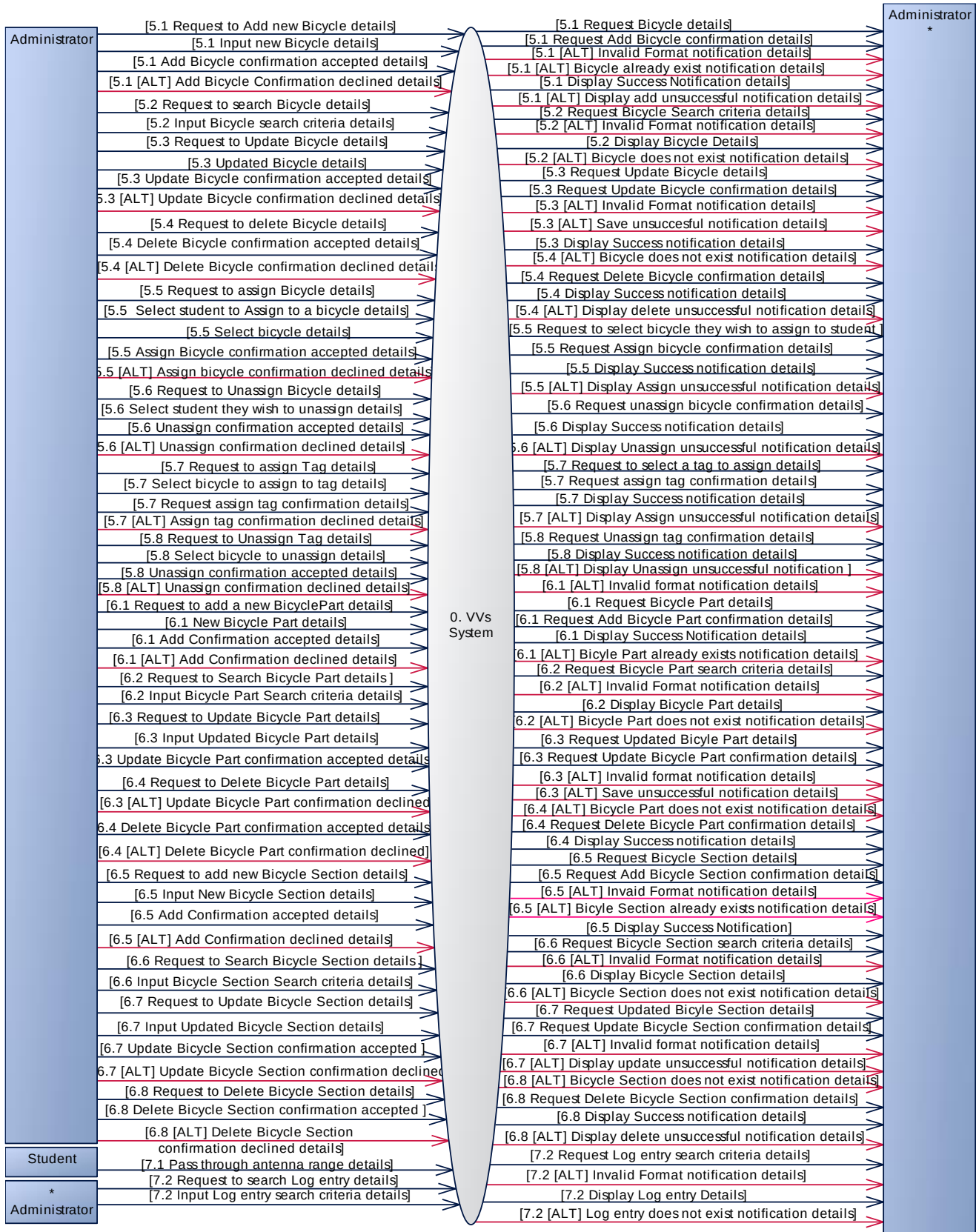


Figure 16 Context Diagram Sub-system 5, 6 and 7

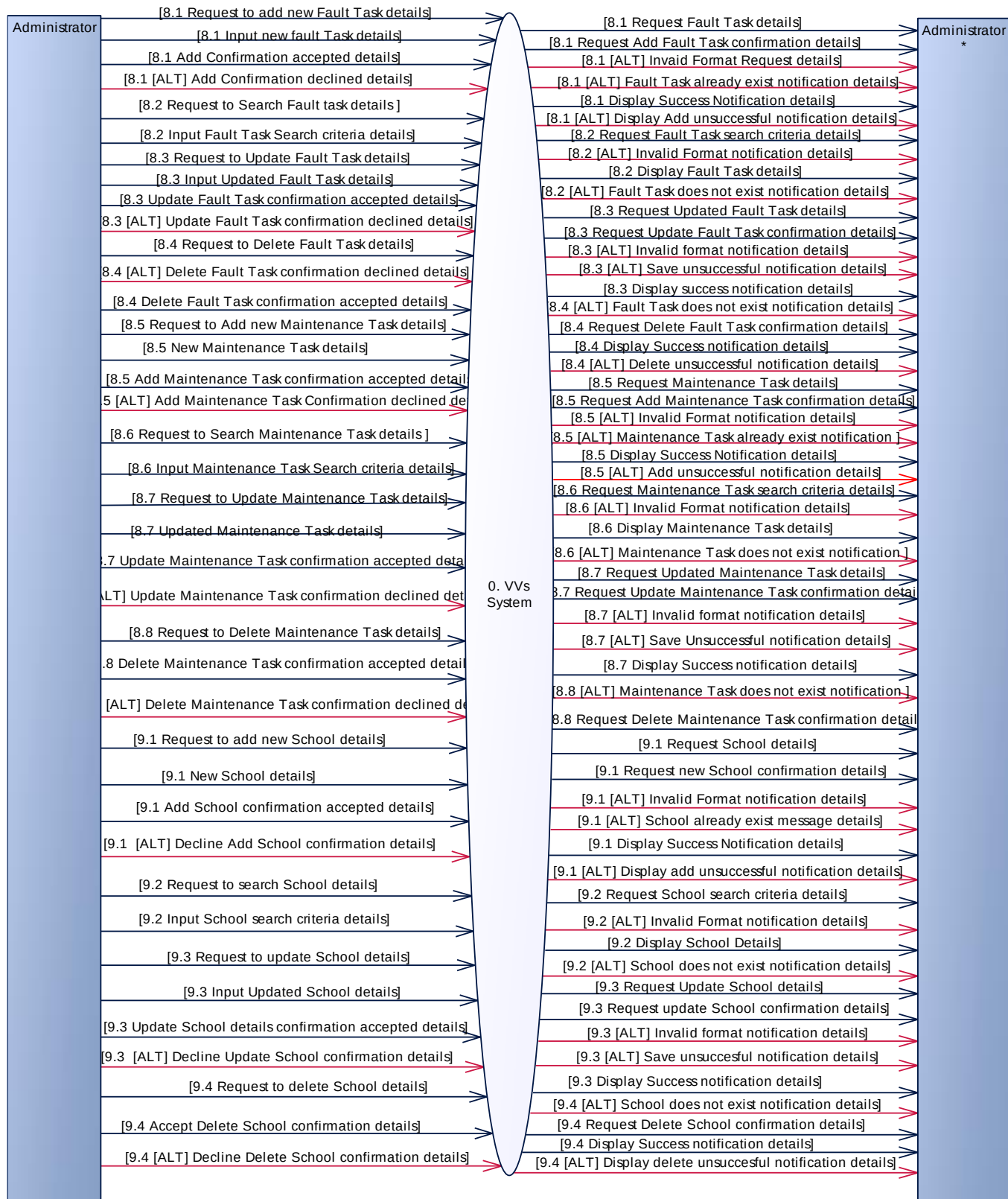


Figure 17 Context Diagram Sub-system 8 and 9

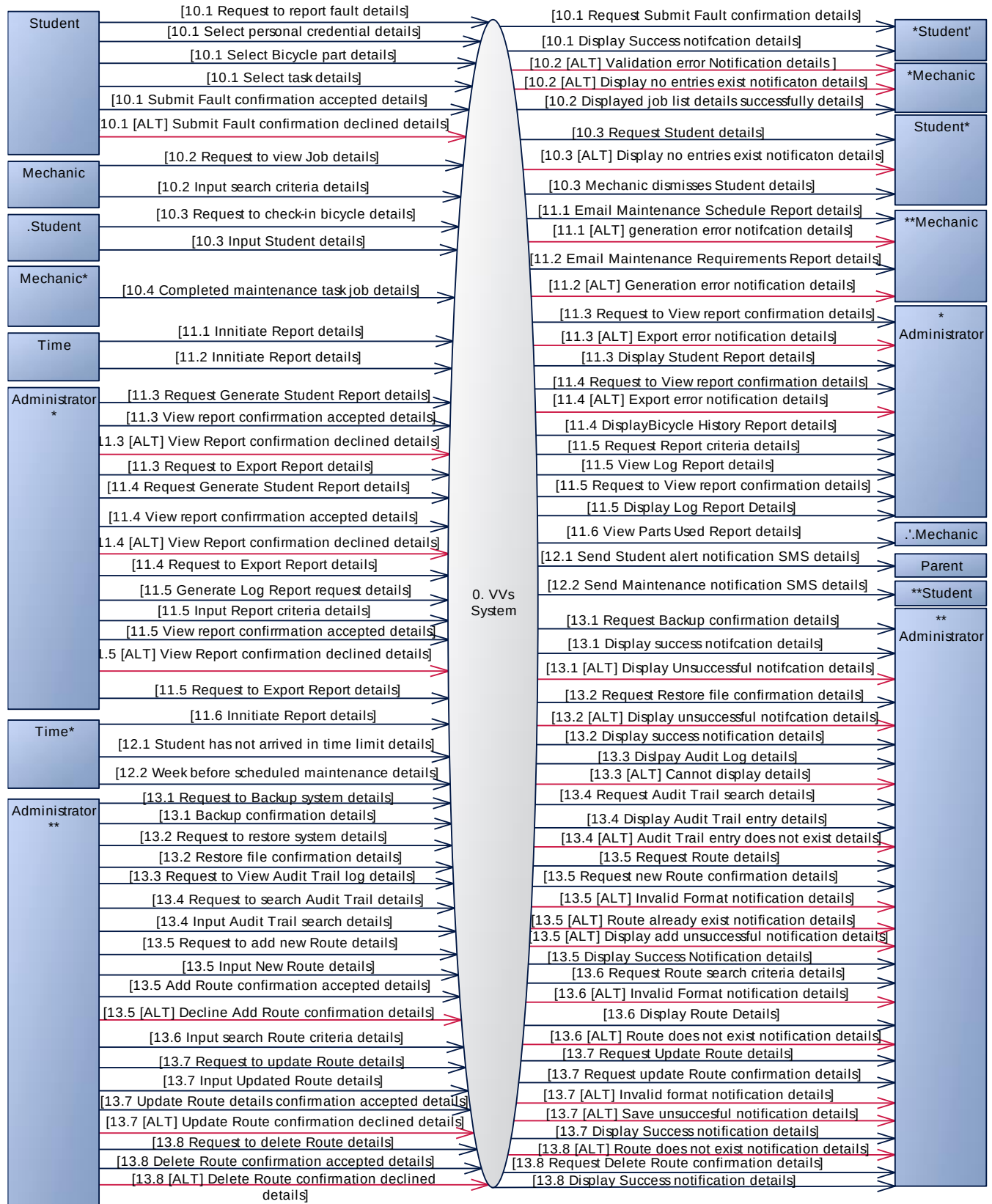


Figure 18 Context Diagram Sub-system 10, 11, 12 and 13



3.2.3 Conclusion

After looking at the context diagram we can see that although there is an overwhelming amount of information about the overall inputs and outputs of the system it does not give a great amount of detail to the individual processes.



3.3 Functional Decomposition Diagram

3.3.1 Introduction

The decomposition diagram is a useful way to see a breakdown of what each Sub-system contains as it can be seen as an expansion of the sub-system. In a decomposition diagram we can see all the way from the sub-system number to the step number of a certain process.



3.3.2 Diagram

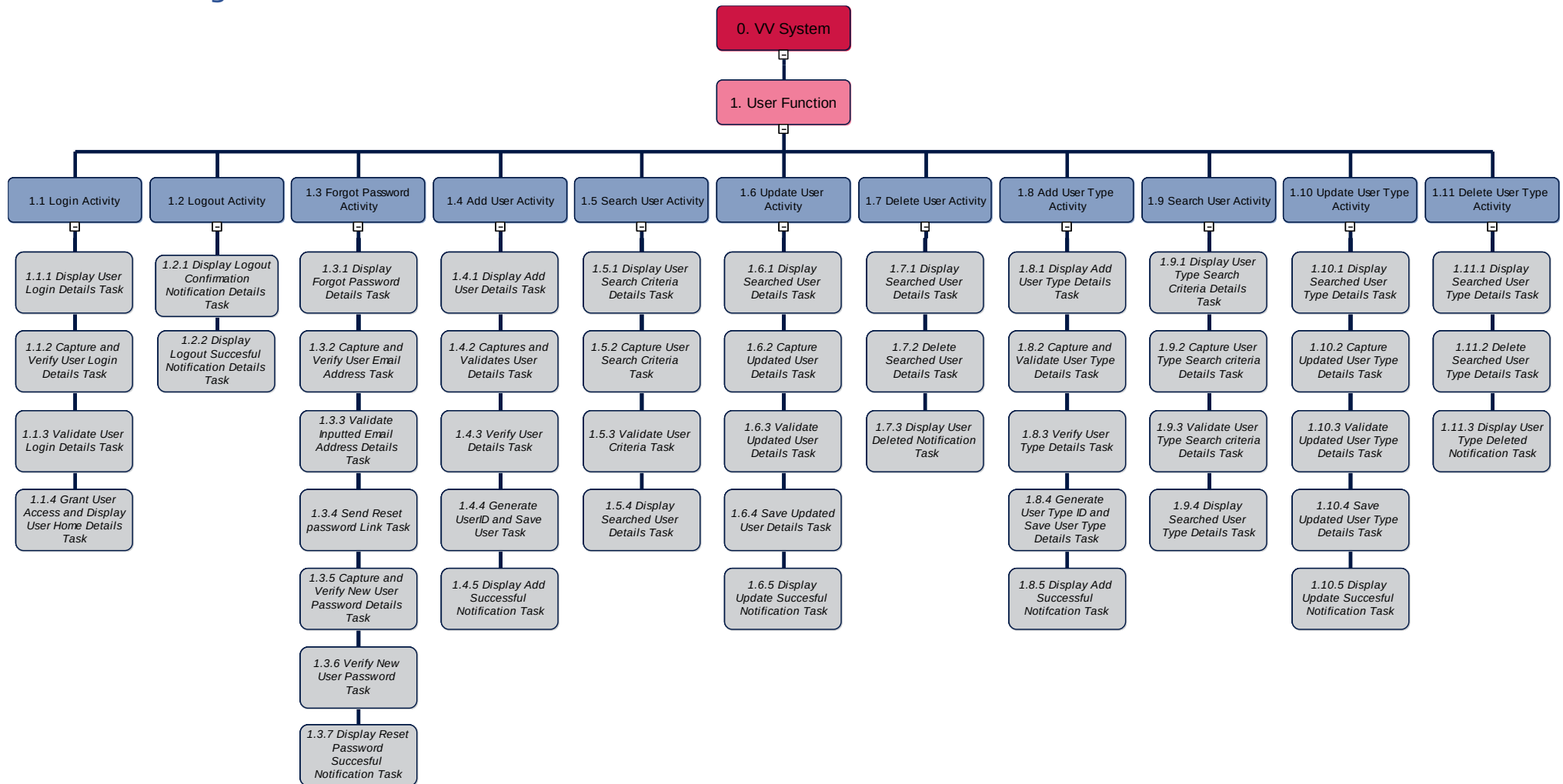


Figure 19 User Function Decomposition Diagram (Split)

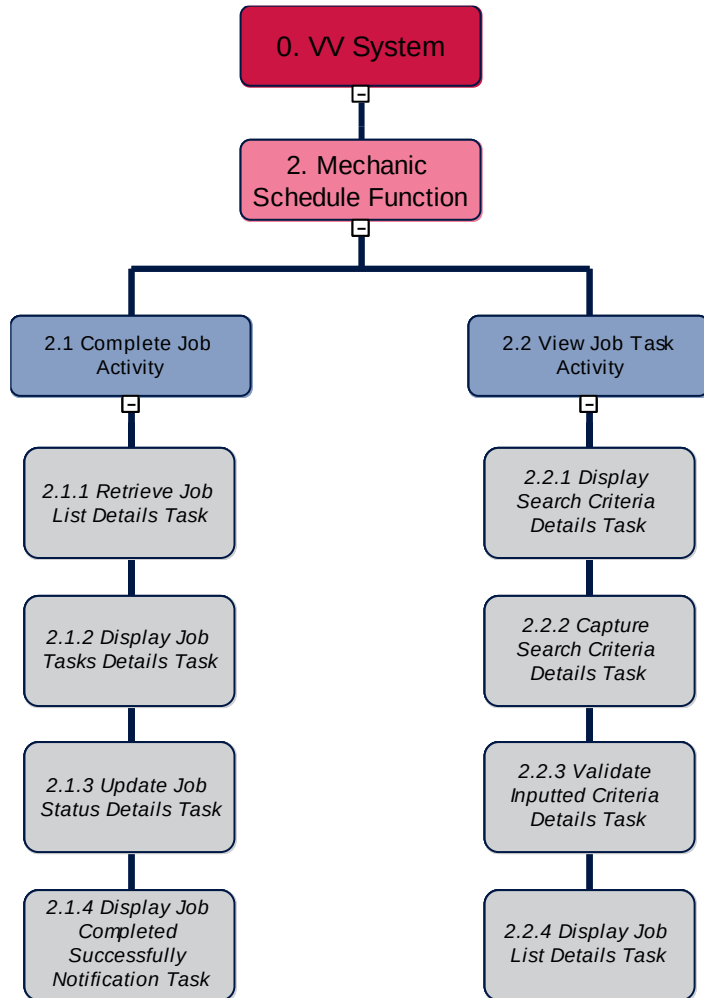


Figure 20 Mechanic Schedule Function Decomposition Diagram (Split)

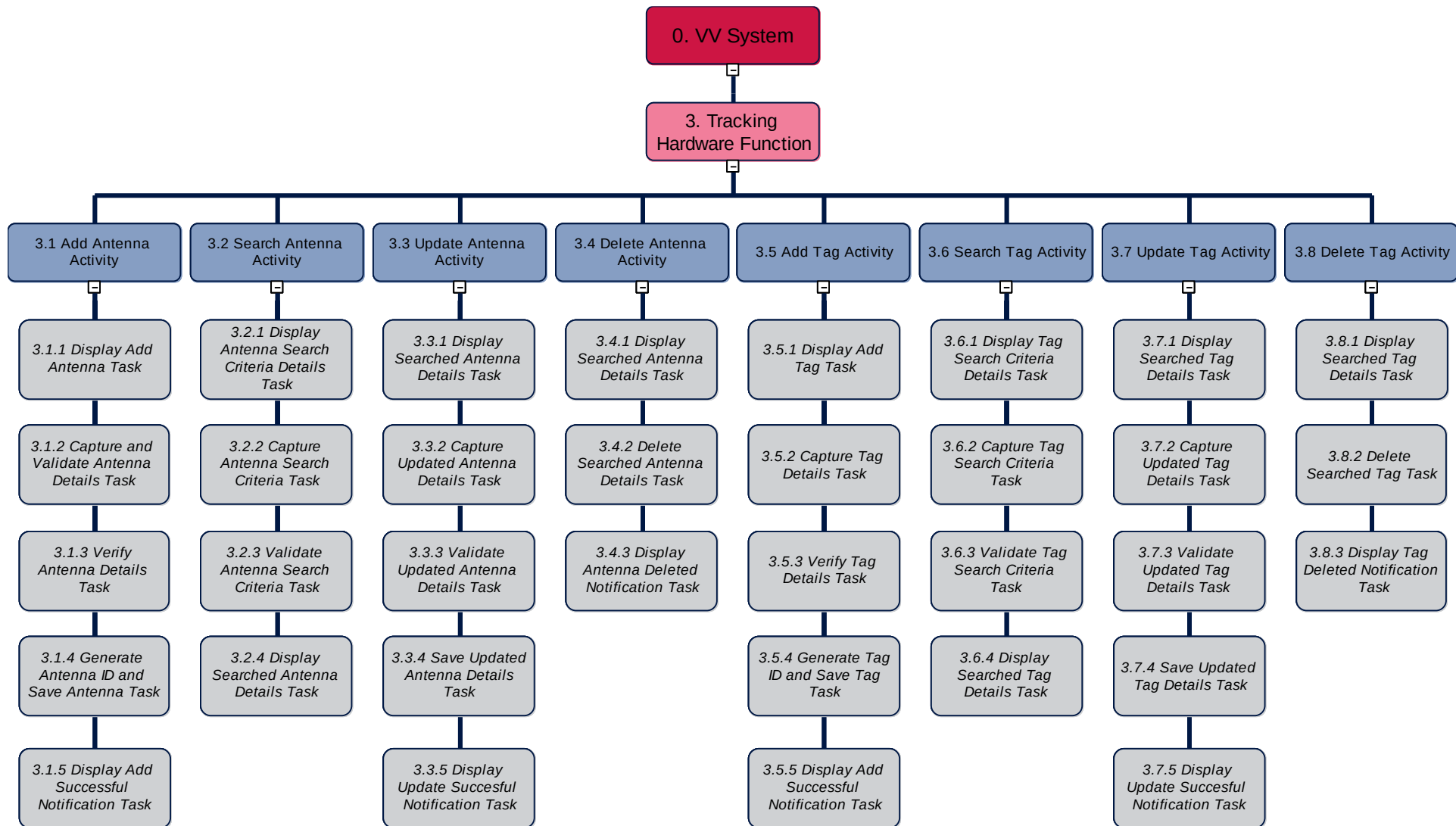


Figure 21 Tracking Hardware Function Decomposition Diagram (Split)

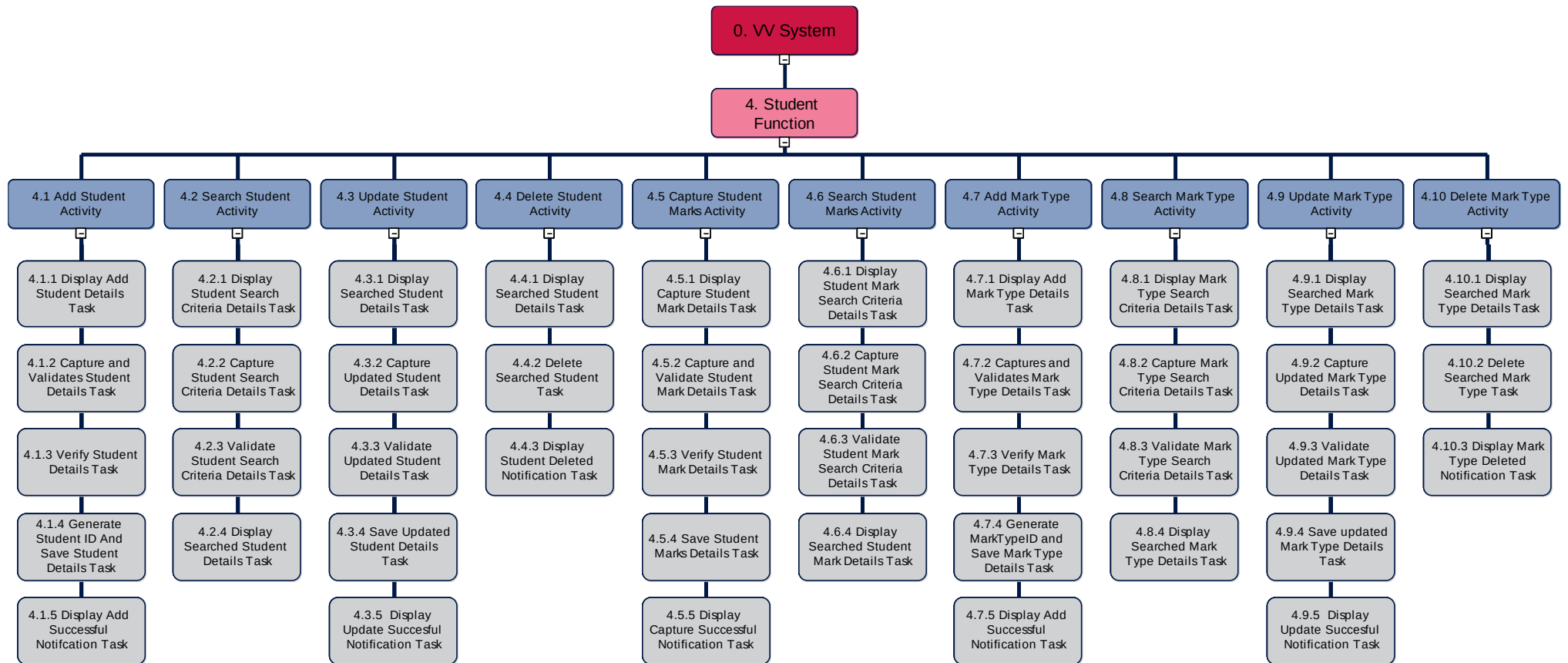


Figure 22 Student Function Decomposition Diagram (Split)

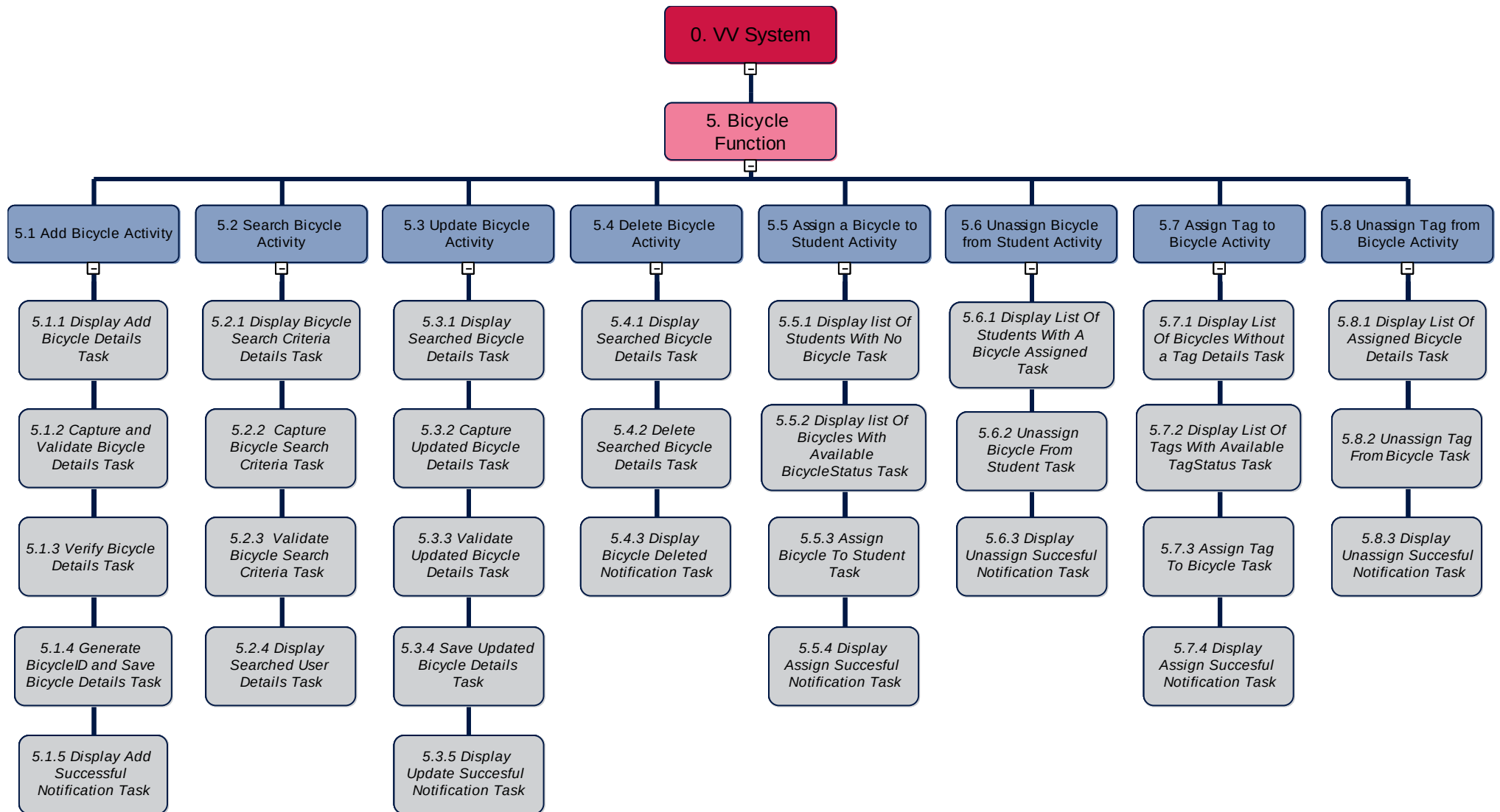


Figure 23 Bicycle Function Decomposition Diagram (Split)

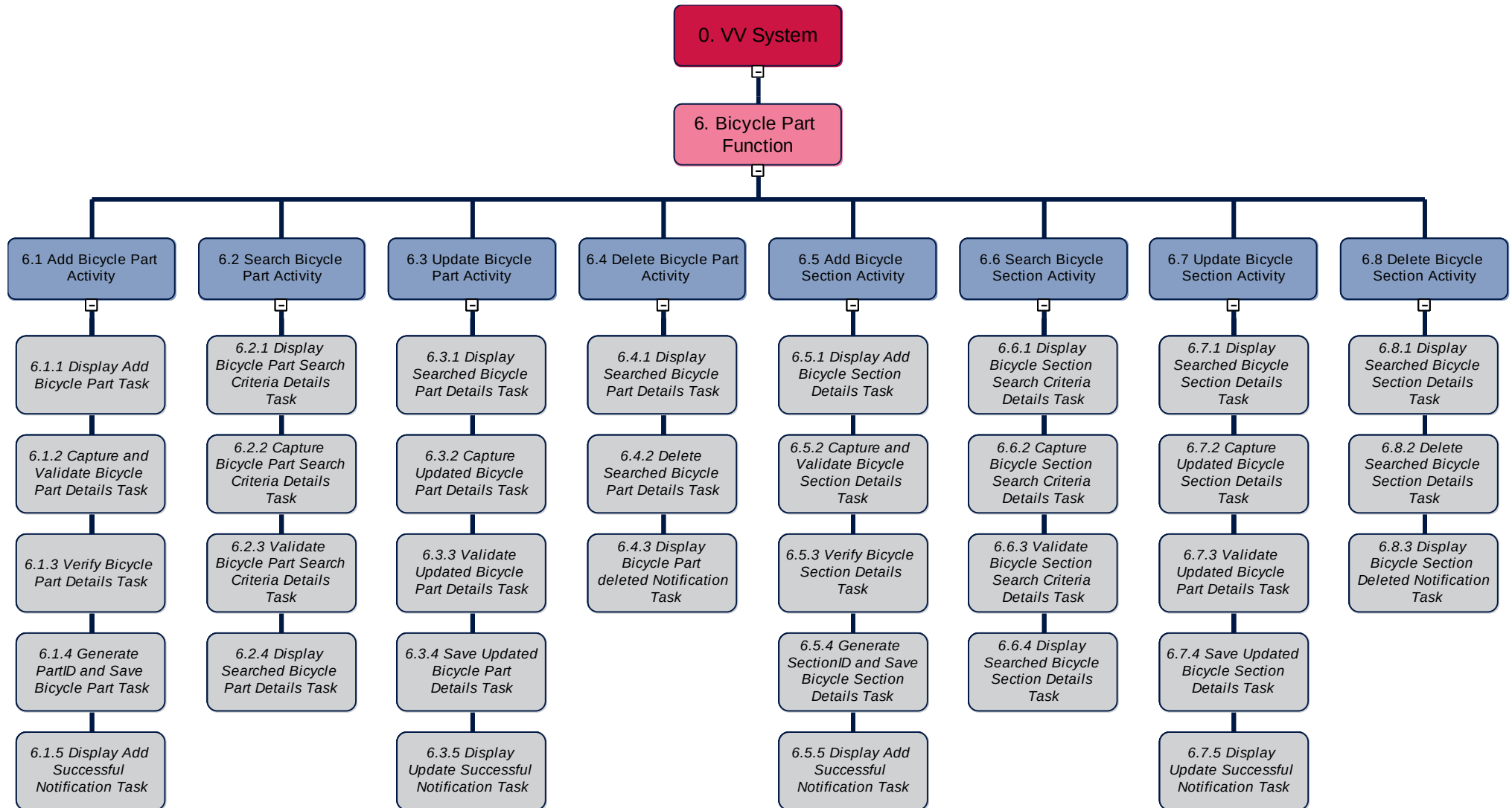


Figure 24 Bicycle Part Function Decomposition Diagram (Split)

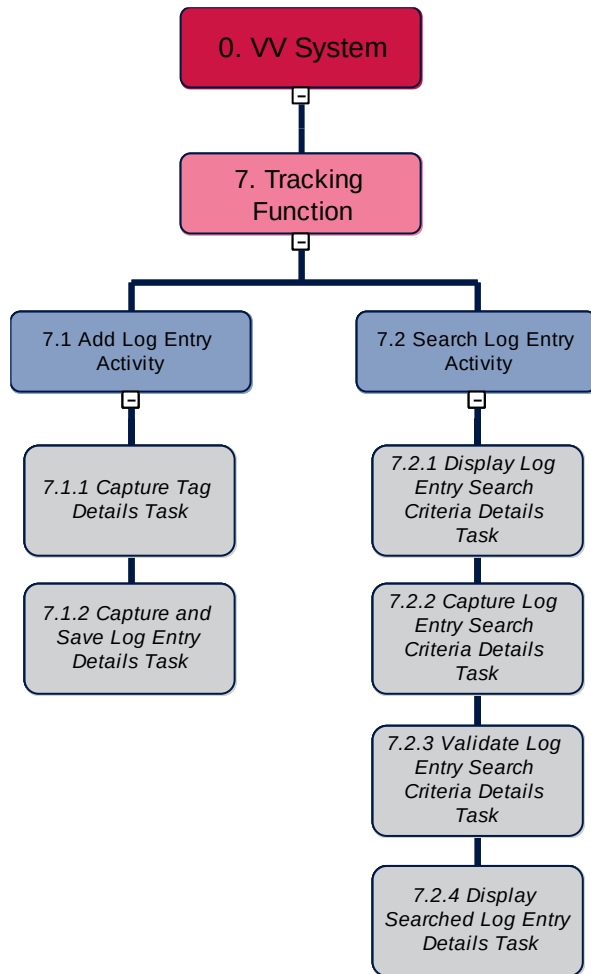


Figure 25 Tracking Function Decomposition Diagram (Split)

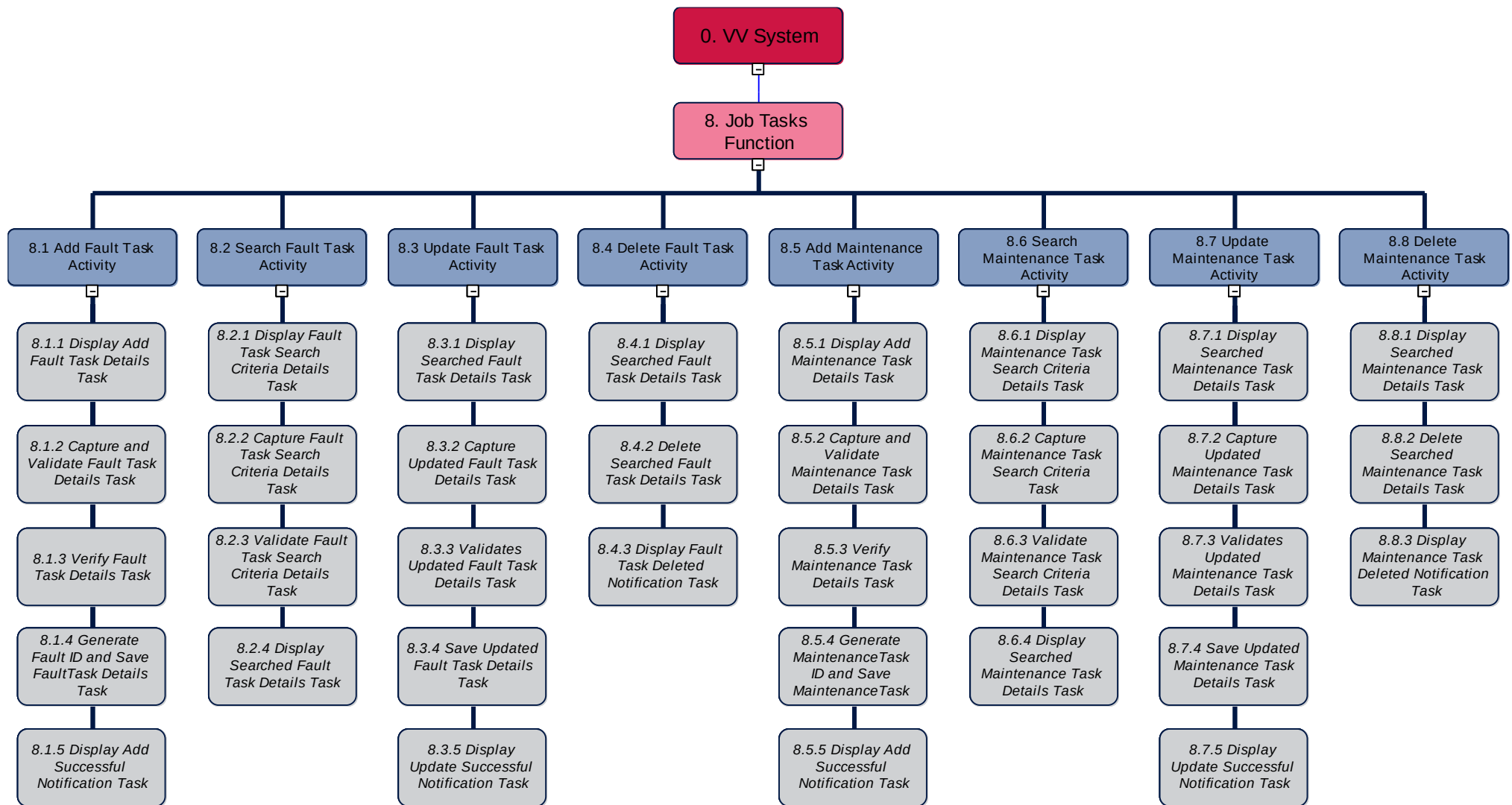


Figure 26 Job Tasks Function Decomposition Diagram (Split)

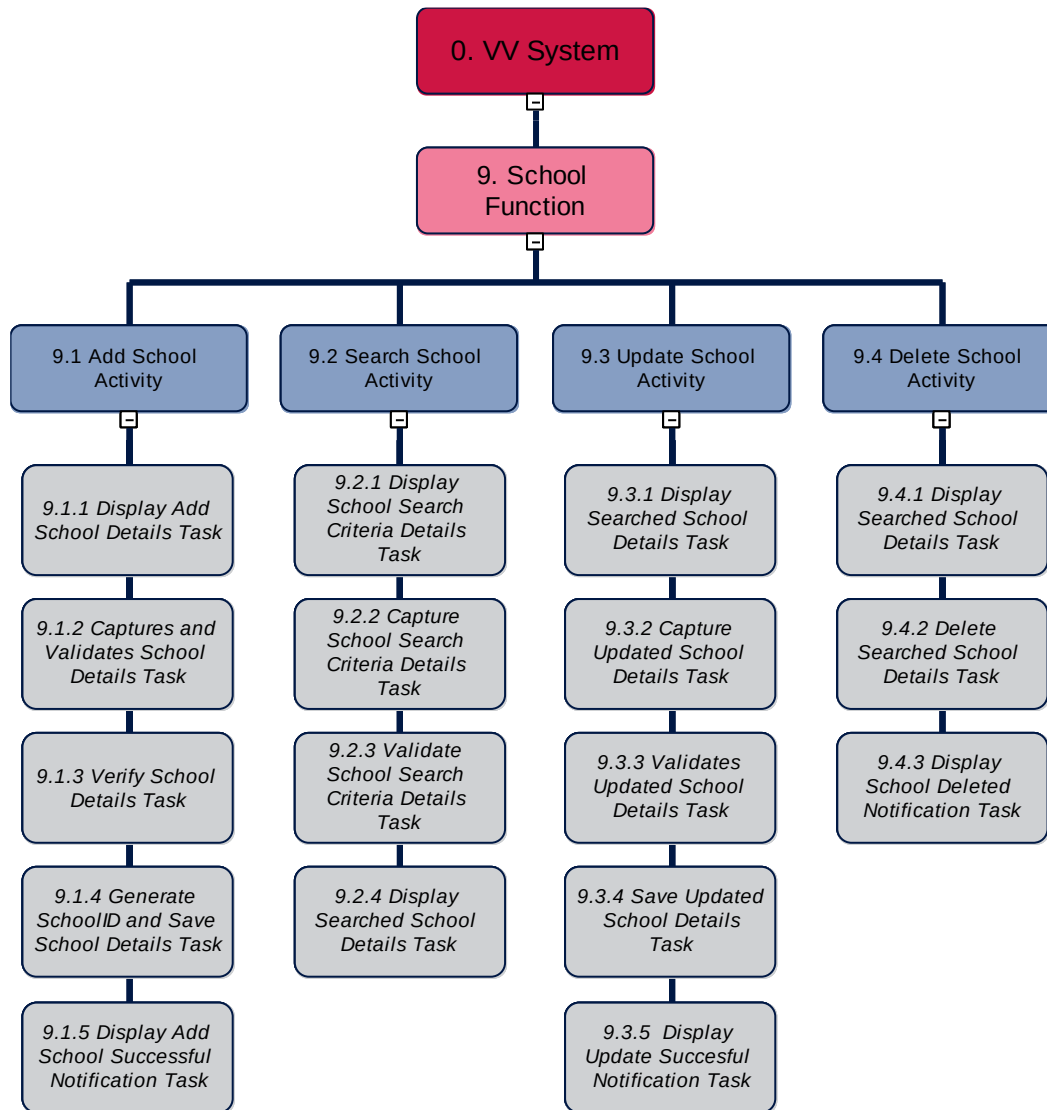


Figure 27 School Function Decomposition Diagram (Split)

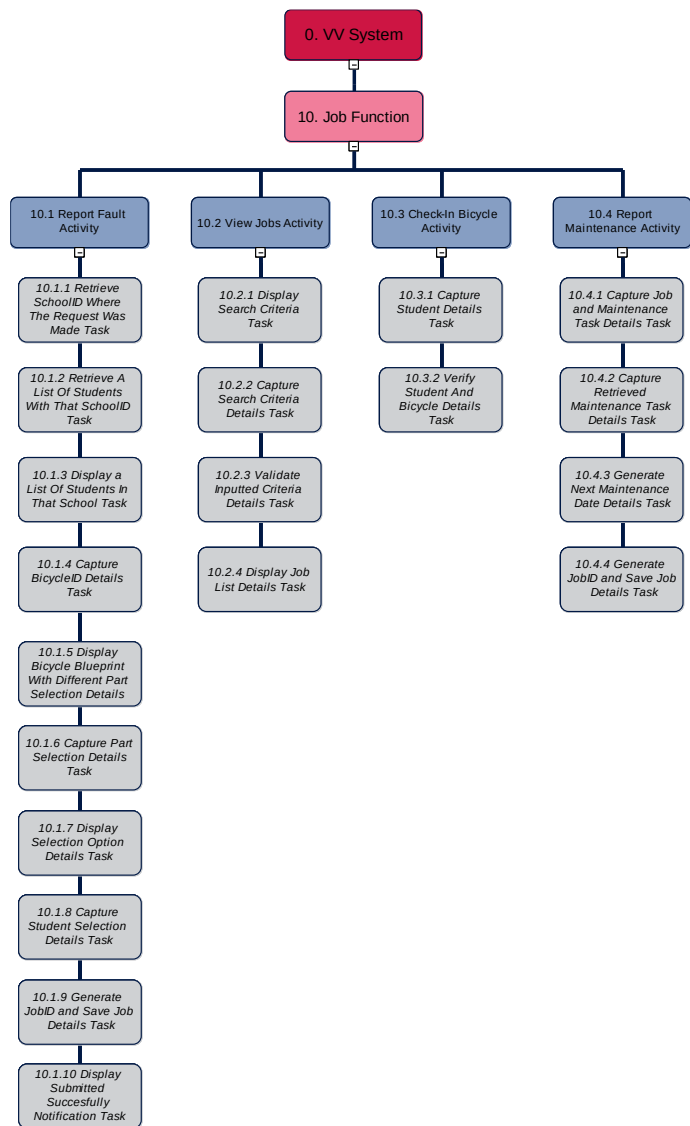


Figure 28 Job Function Decomposition Diagram (Split)



11/09/20

#BIKES4ERP

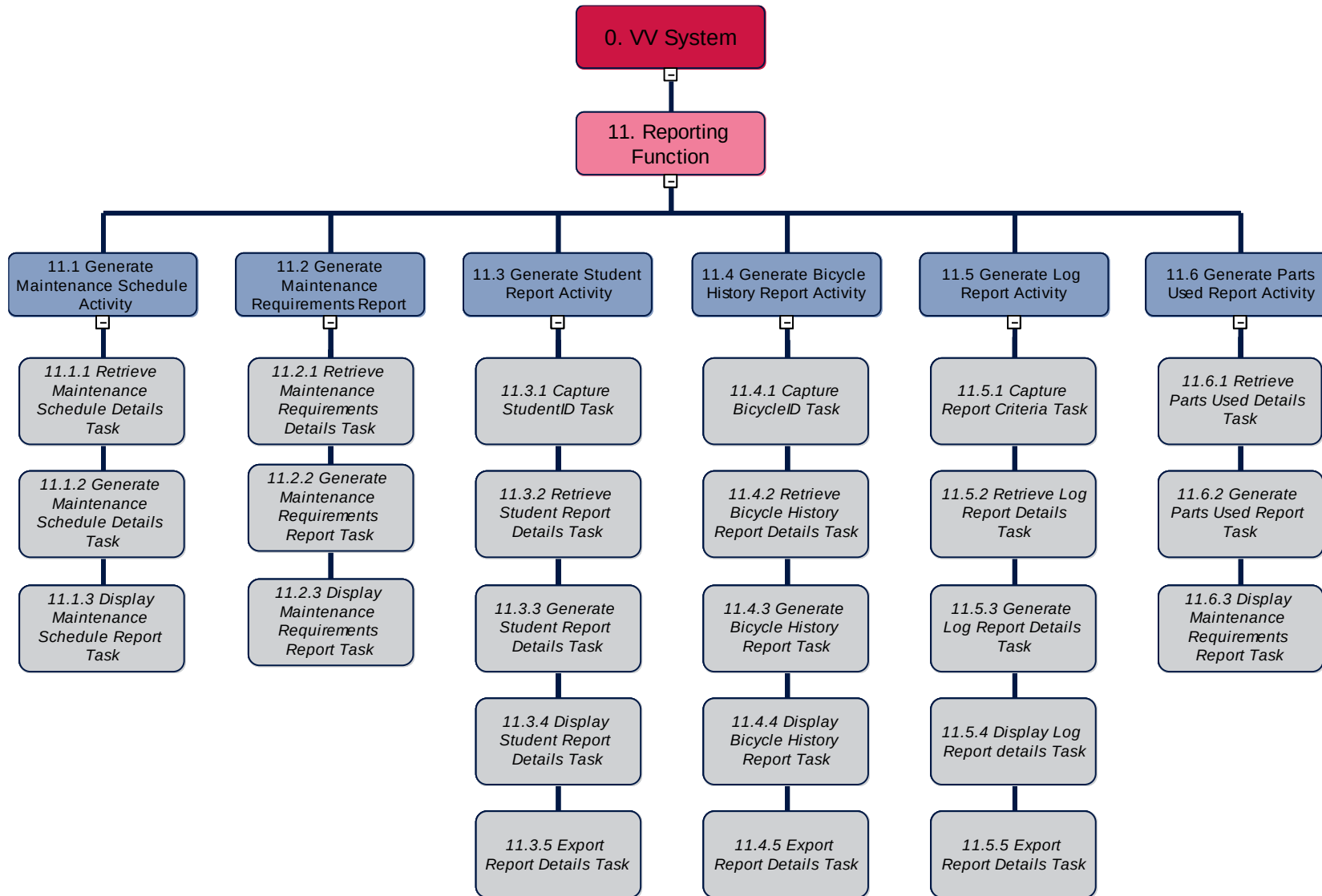


Figure 29 Reporting Function Decomposition Diagram (Split)

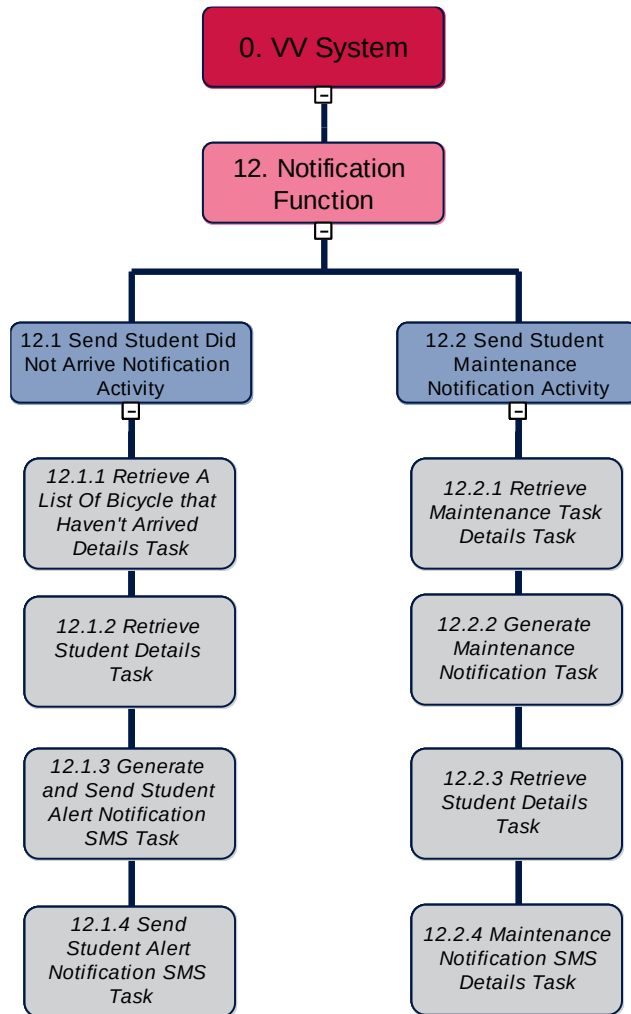


Figure 30 Notification Function Decomposition Diagram (Split)

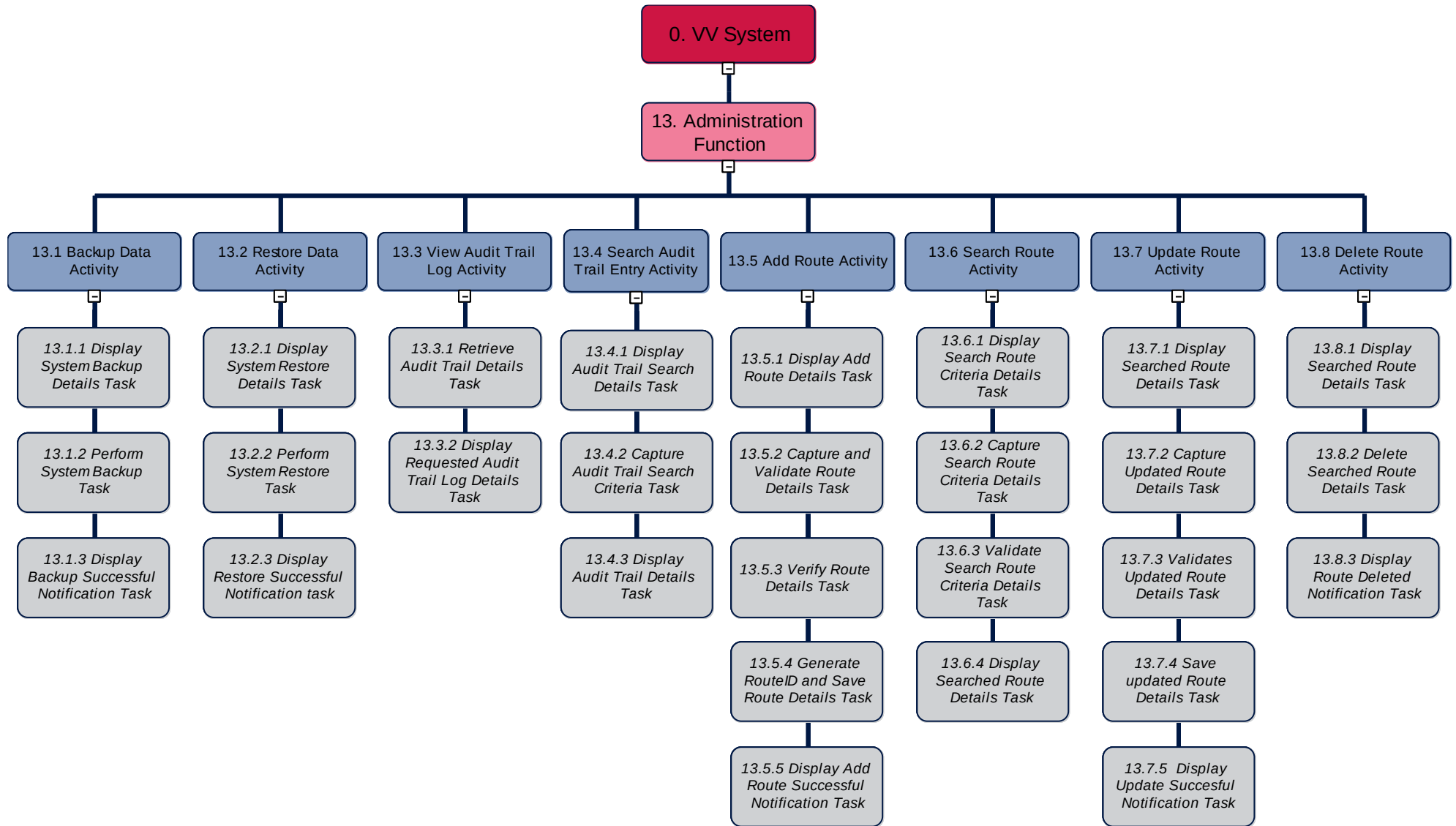


Figure 31 Administration Function Decomposition Diagram (Split)



3.3.3 Conclusion

After looking at our decomposition diagram we can see how each sub-system has its own processes which in turn have their own steps. This therefore allows us to break down a large amount of data into smaller pieces in a more understandable way.



3.4 Data Flow Diagrams

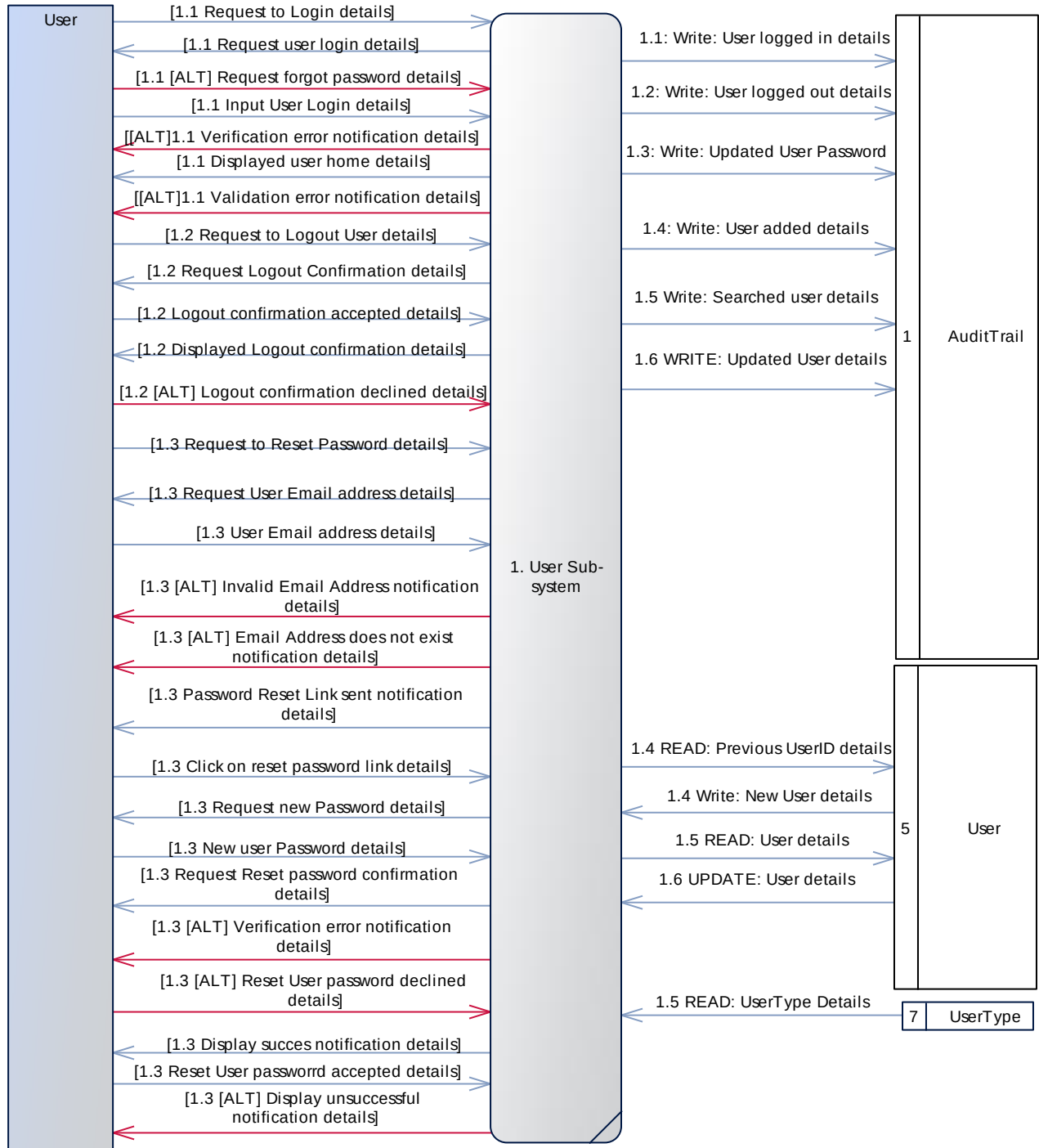
3.4.1 Introduction

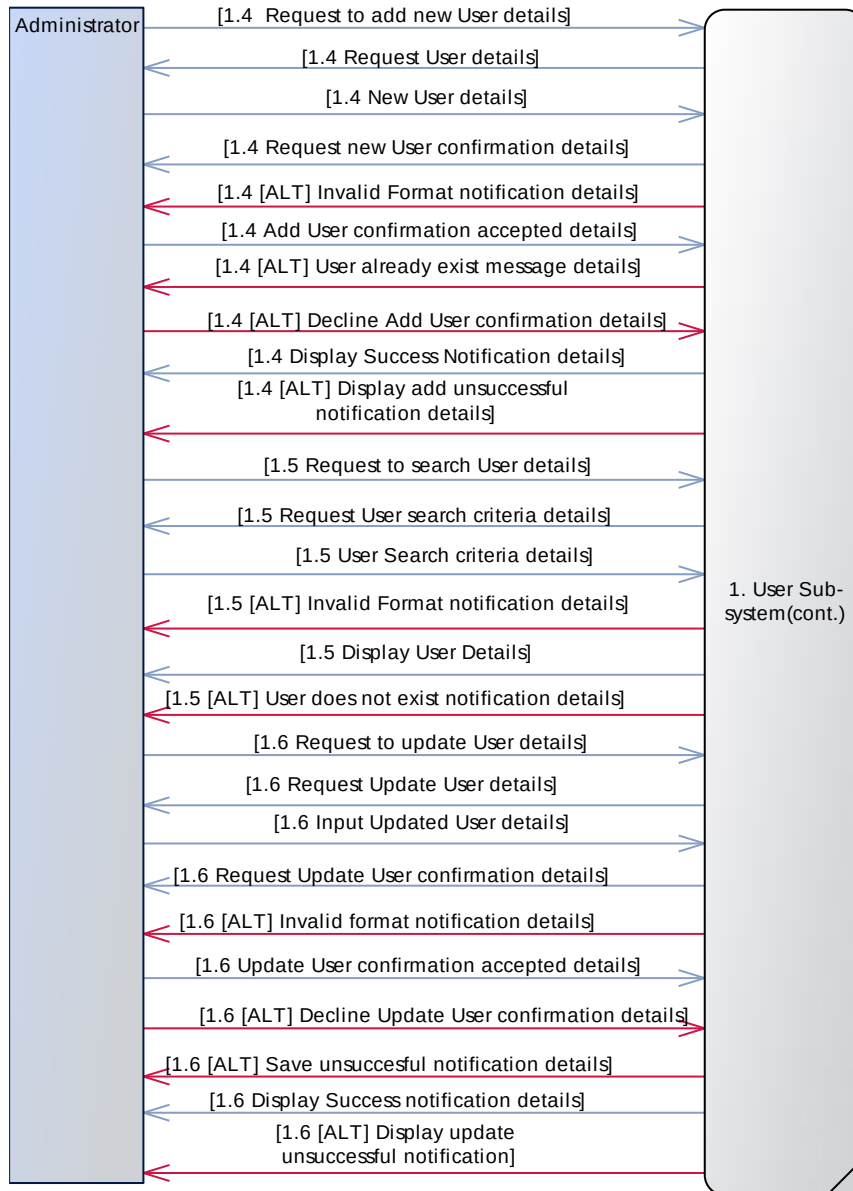
Data flow diagrams can be seen as an extension of the context diagram as they go into much more detail about each process and its flows. The high level shows all the sub system's inputs and outputs as well as the data stores which the process makes use of. The next diagram is the middle level and is very similar to the high except it shows the inputs and outputs of each process instead of each subsystem as well as showing invokes (if there are any). The last data flow diagram is the primitive level diagram and this diagram is the most in depth as it shows the steps of each process as well in a diagram format.

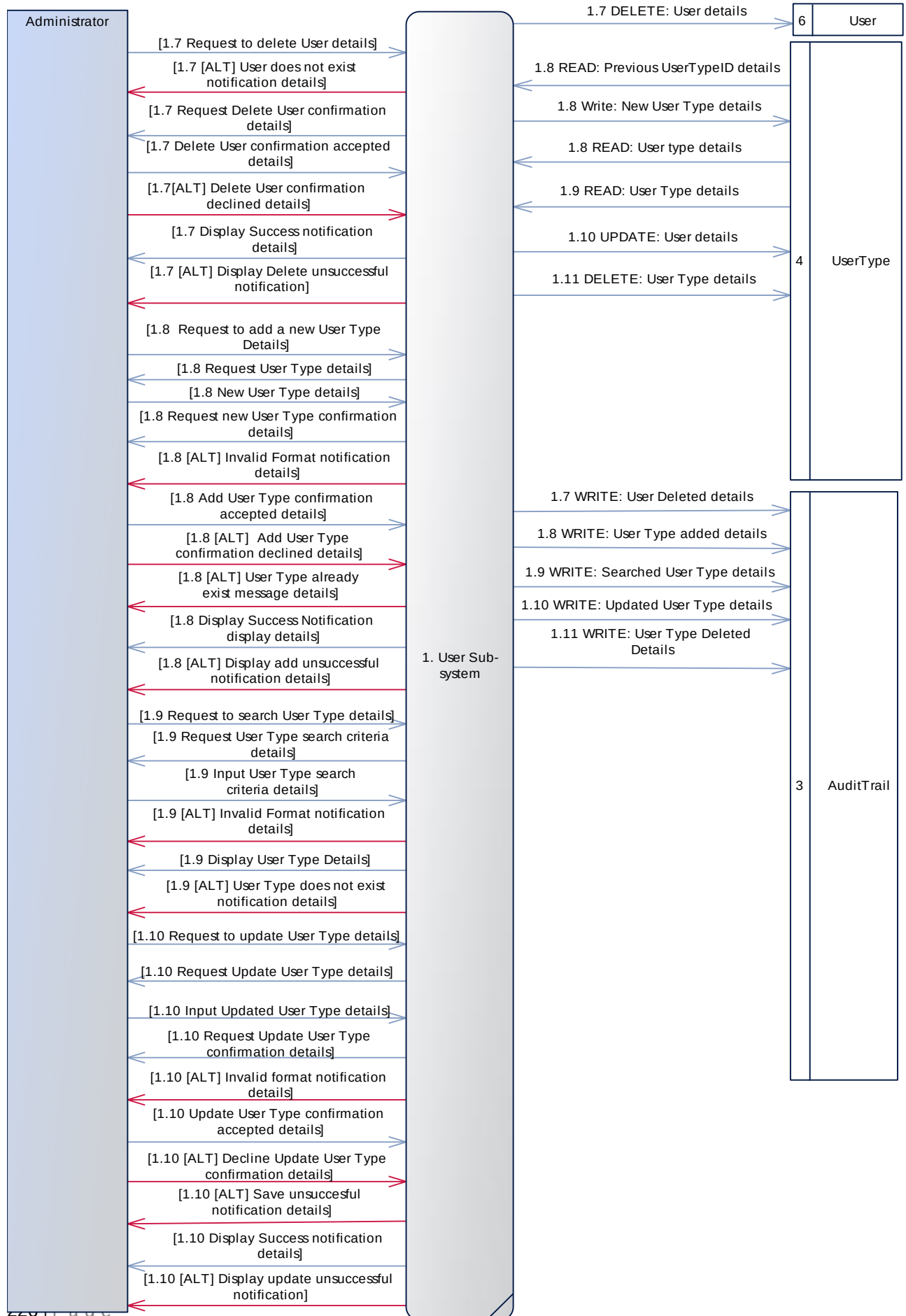


3.4.2 High Level Data Flow Diagrams

3.4.2.1 Sub-system 1: User







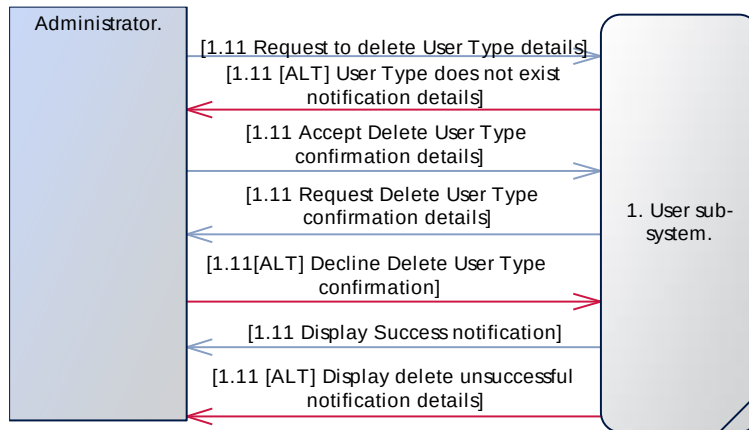


Figure 32 User high level diagram



3.4.2.2 Sub-system 2: Mechanic Schedule

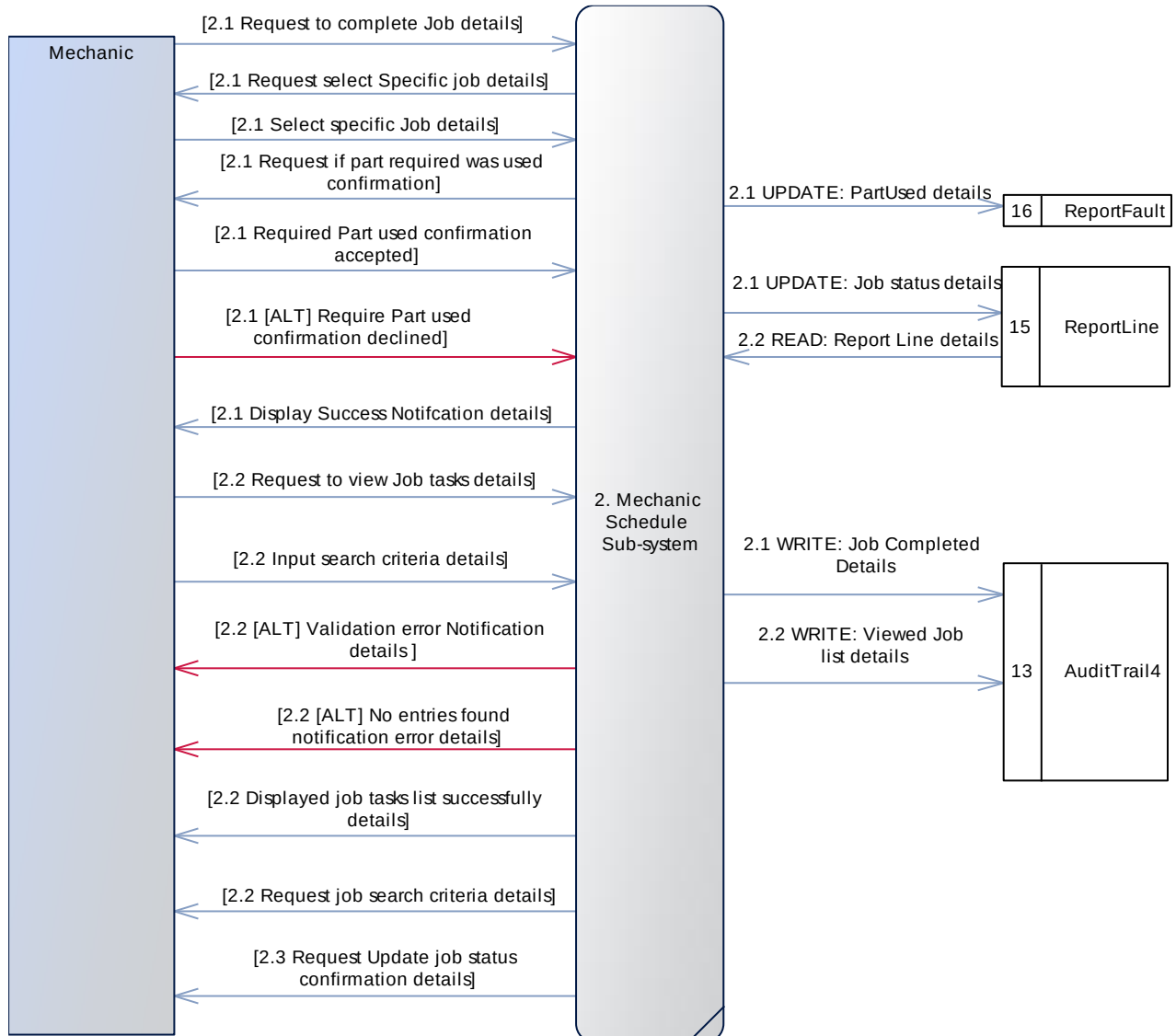
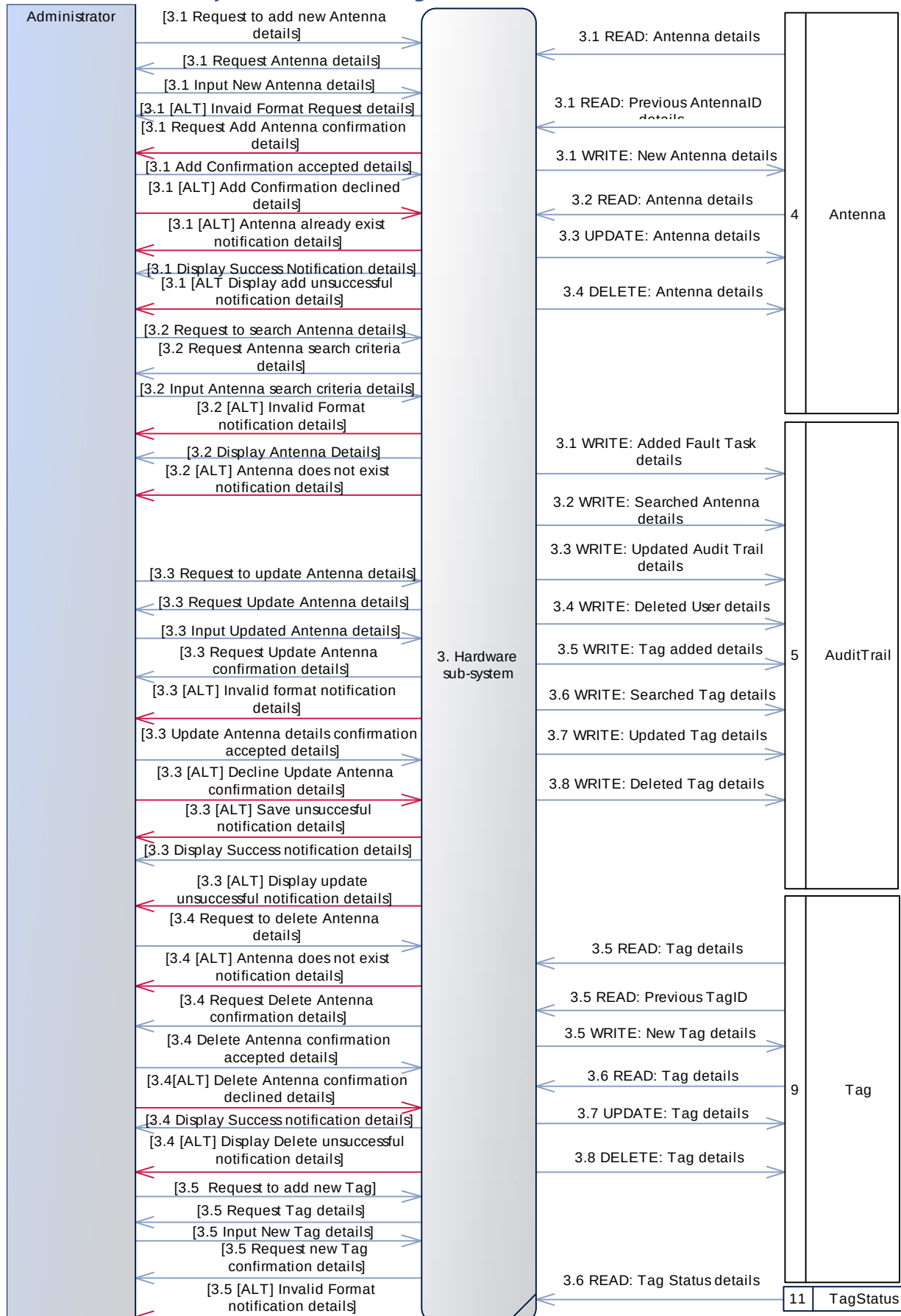


Figure 33 Mechanic Schedule high level diagram



3.4.2.3 Sub-system 3: Tracking Hardware



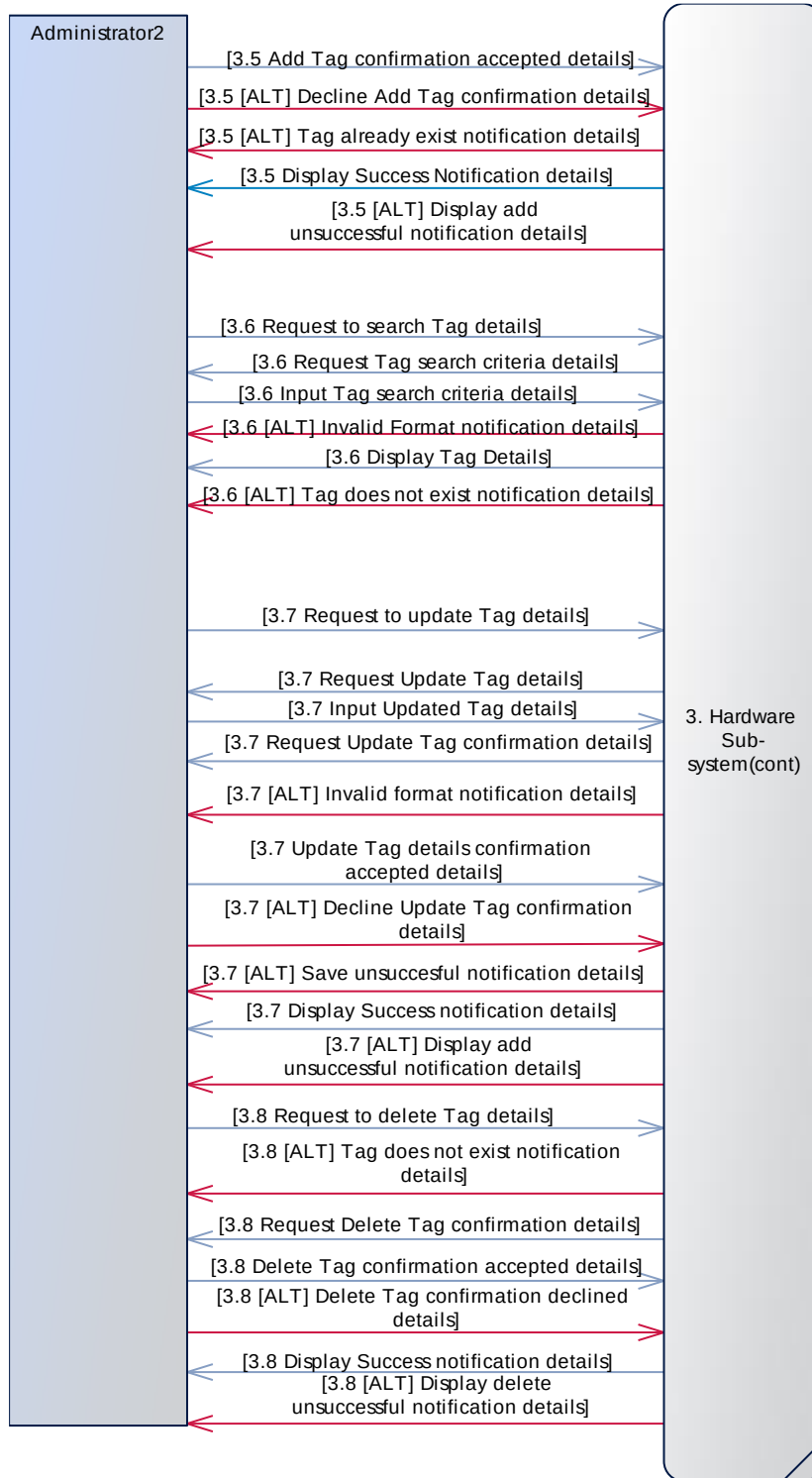
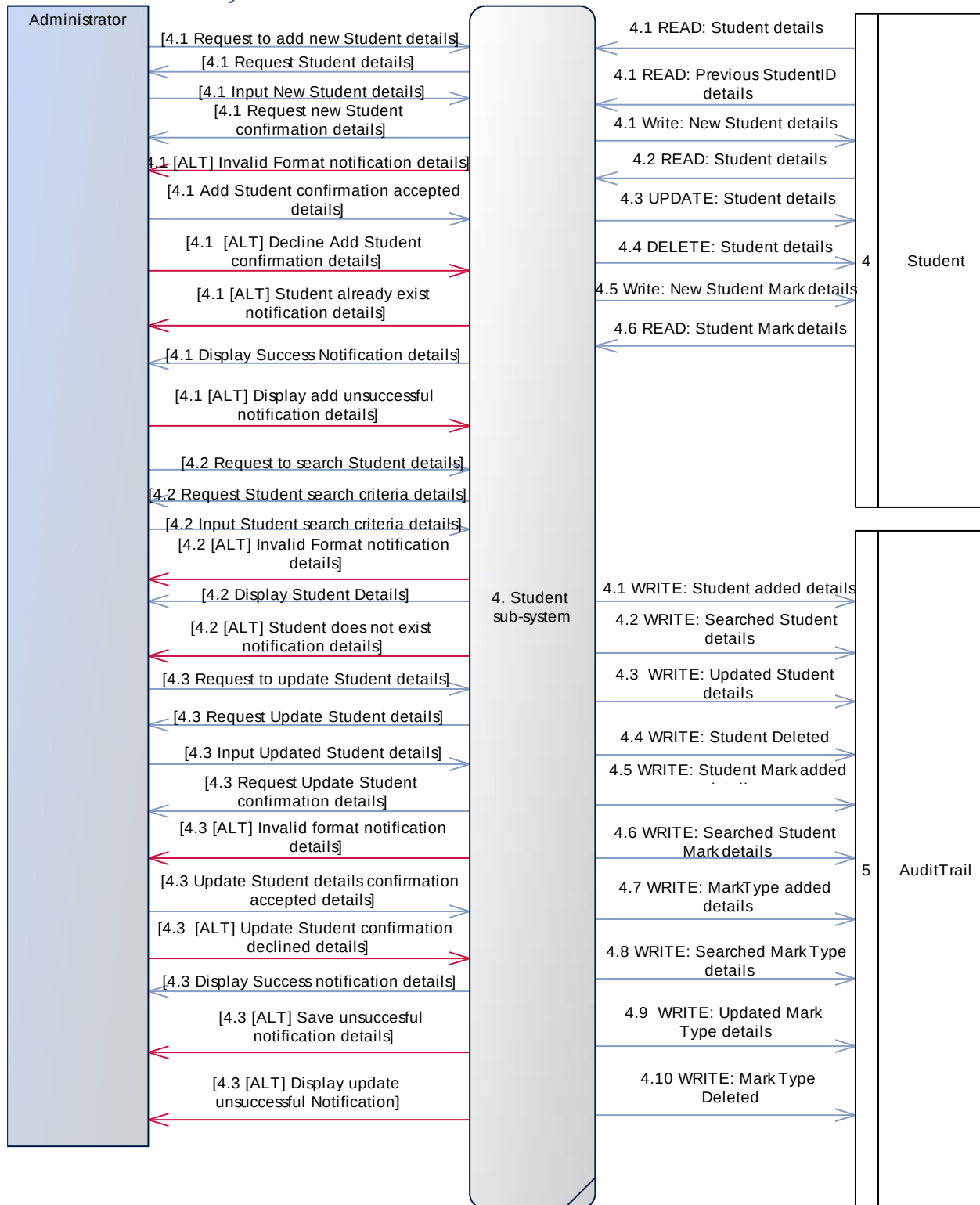
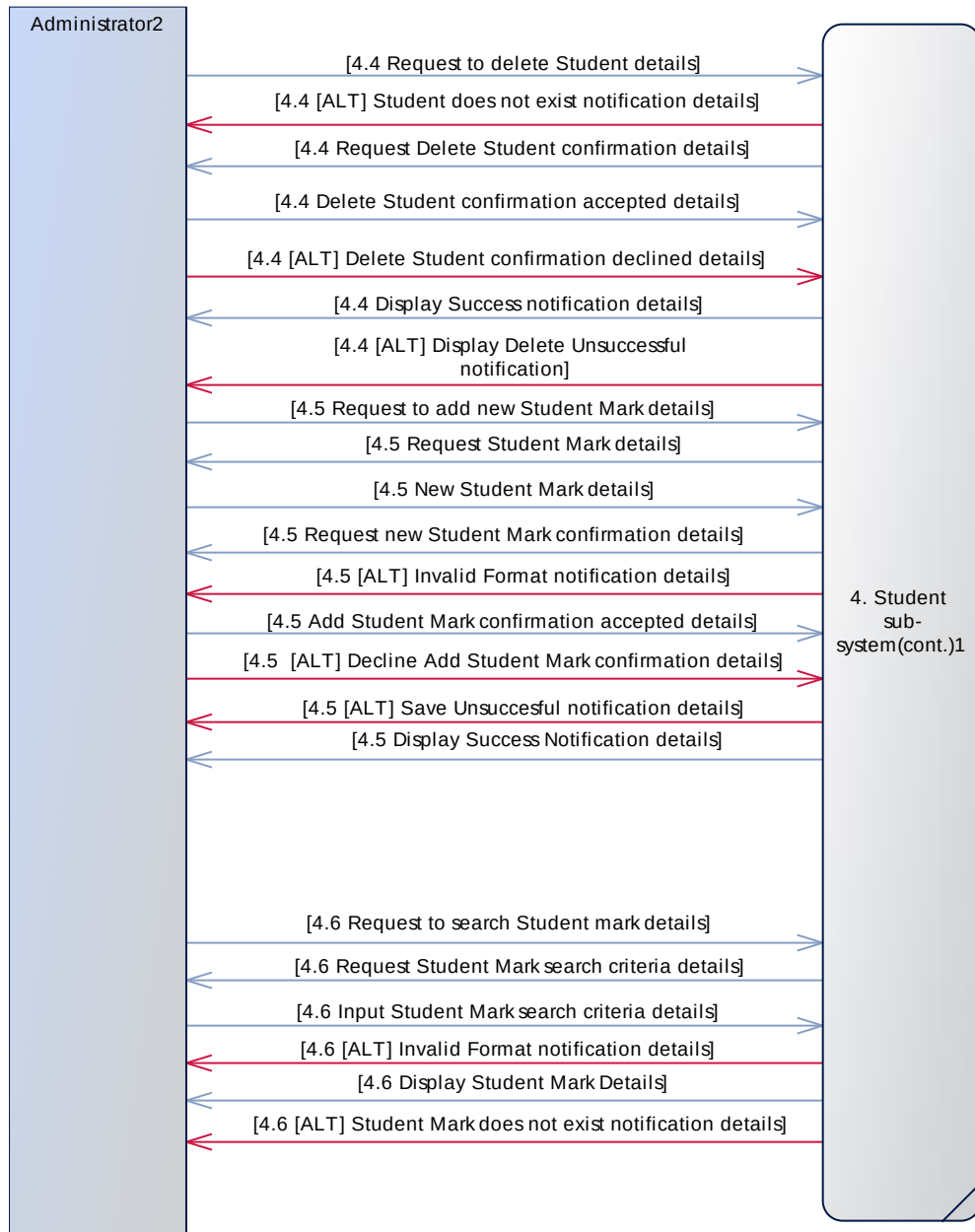


Figure 34 Hardware high level diagram



3.4.2.4 Sub-system 4: Student





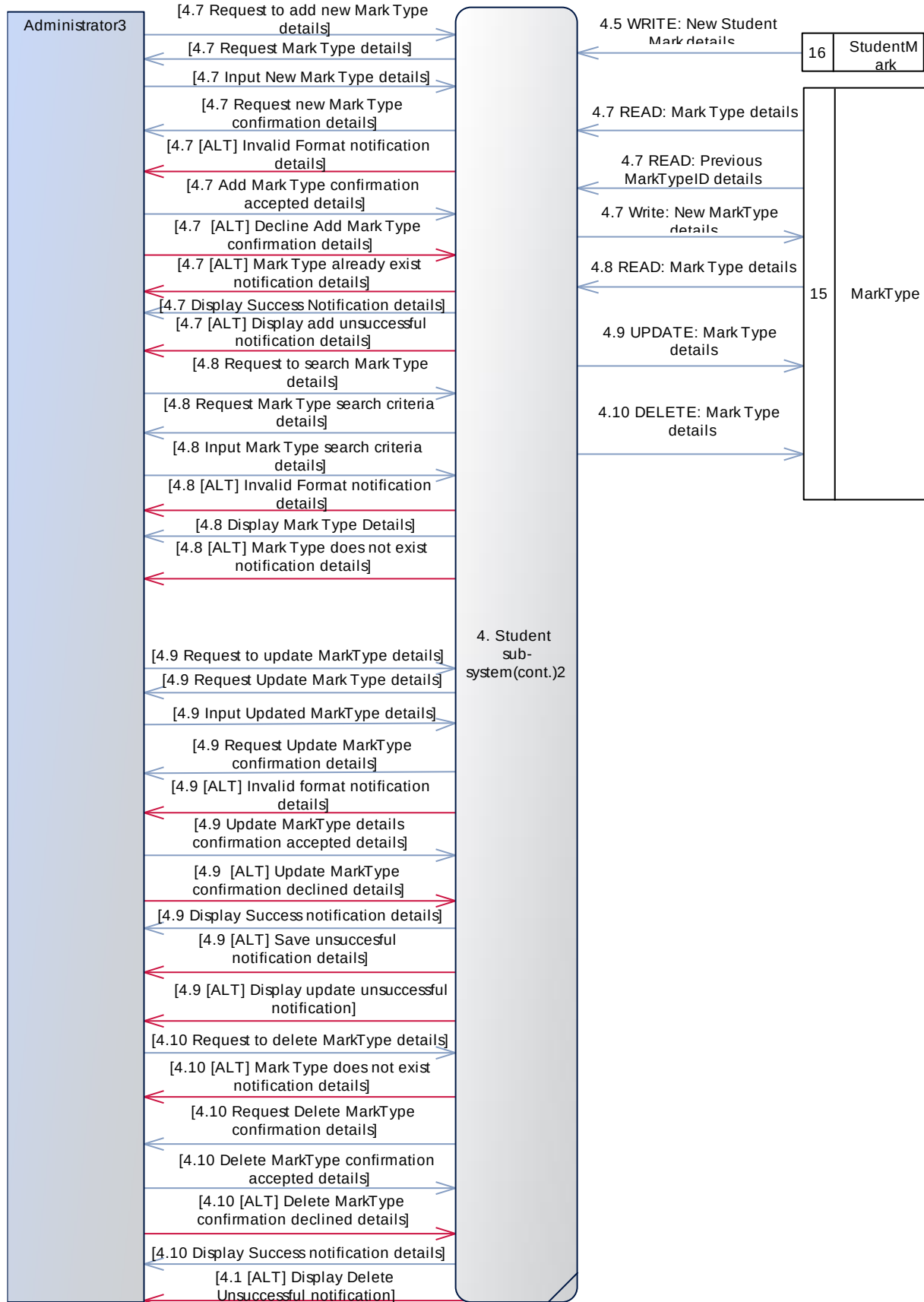
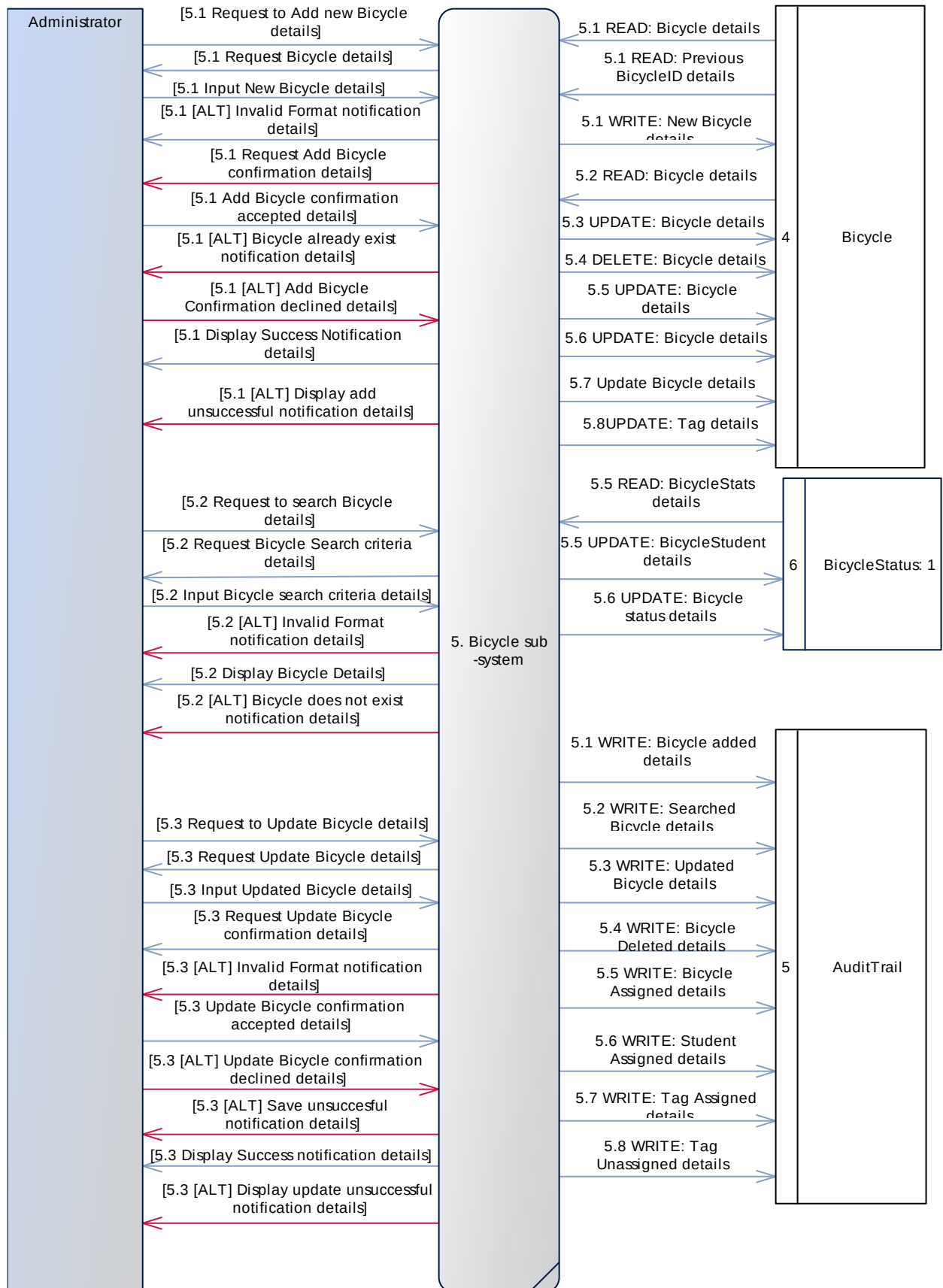


Figure 35 Student high level diagram



3.4.2.5 Sub-system 5: Bicycle



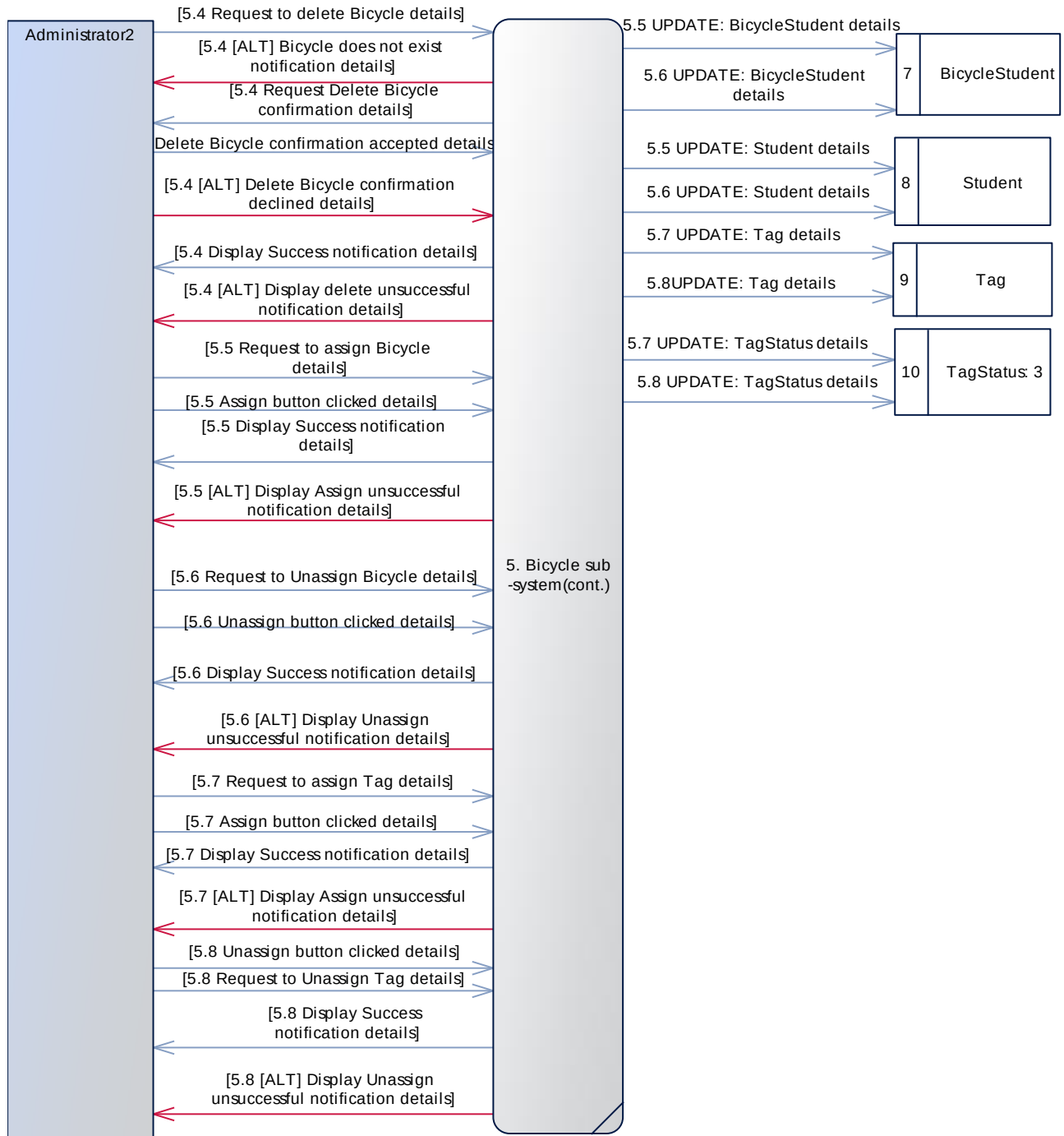
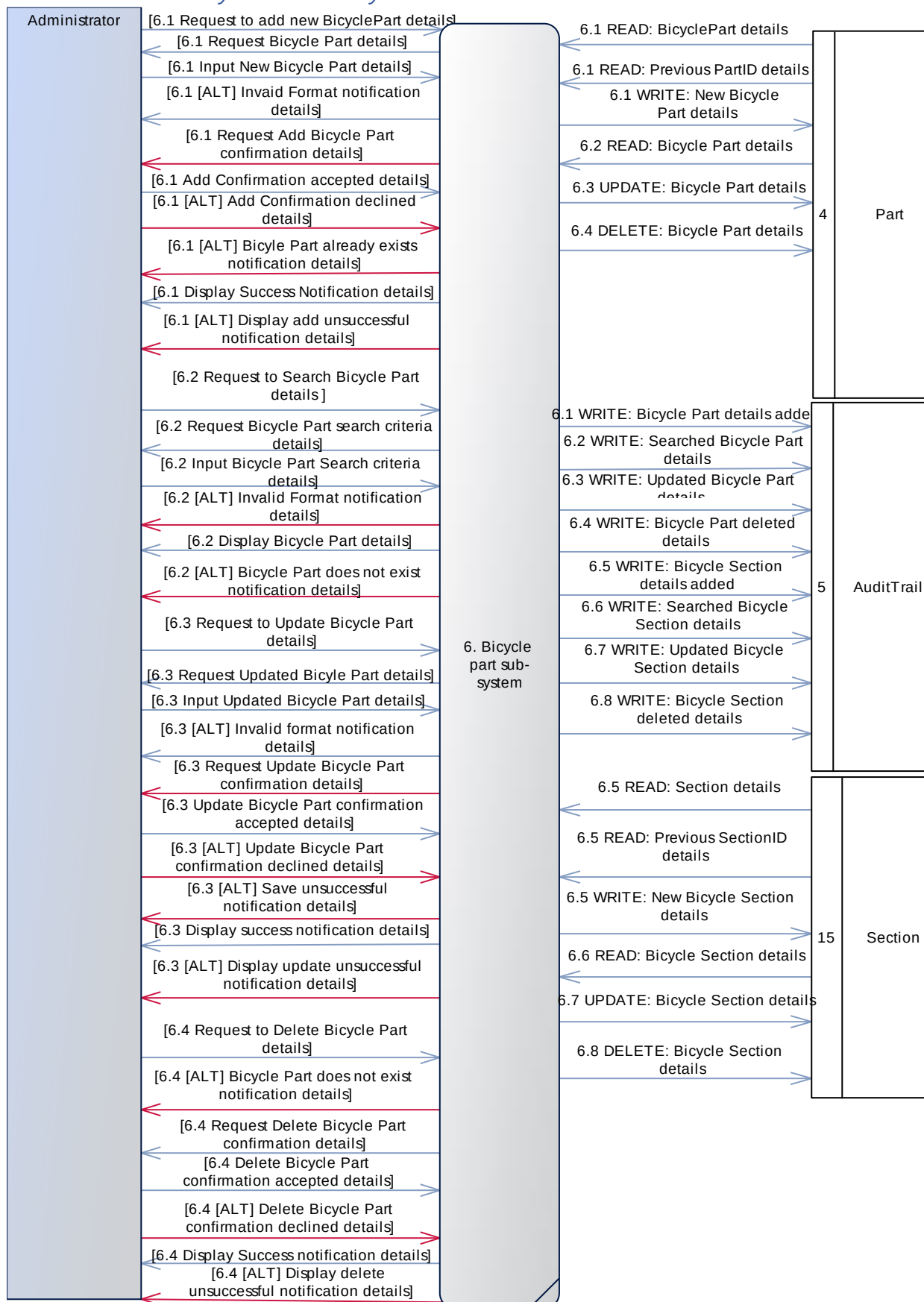


Figure 36 Bicycle high level diagram



3.4.2.6 Sub-system 6: Bicycle Part



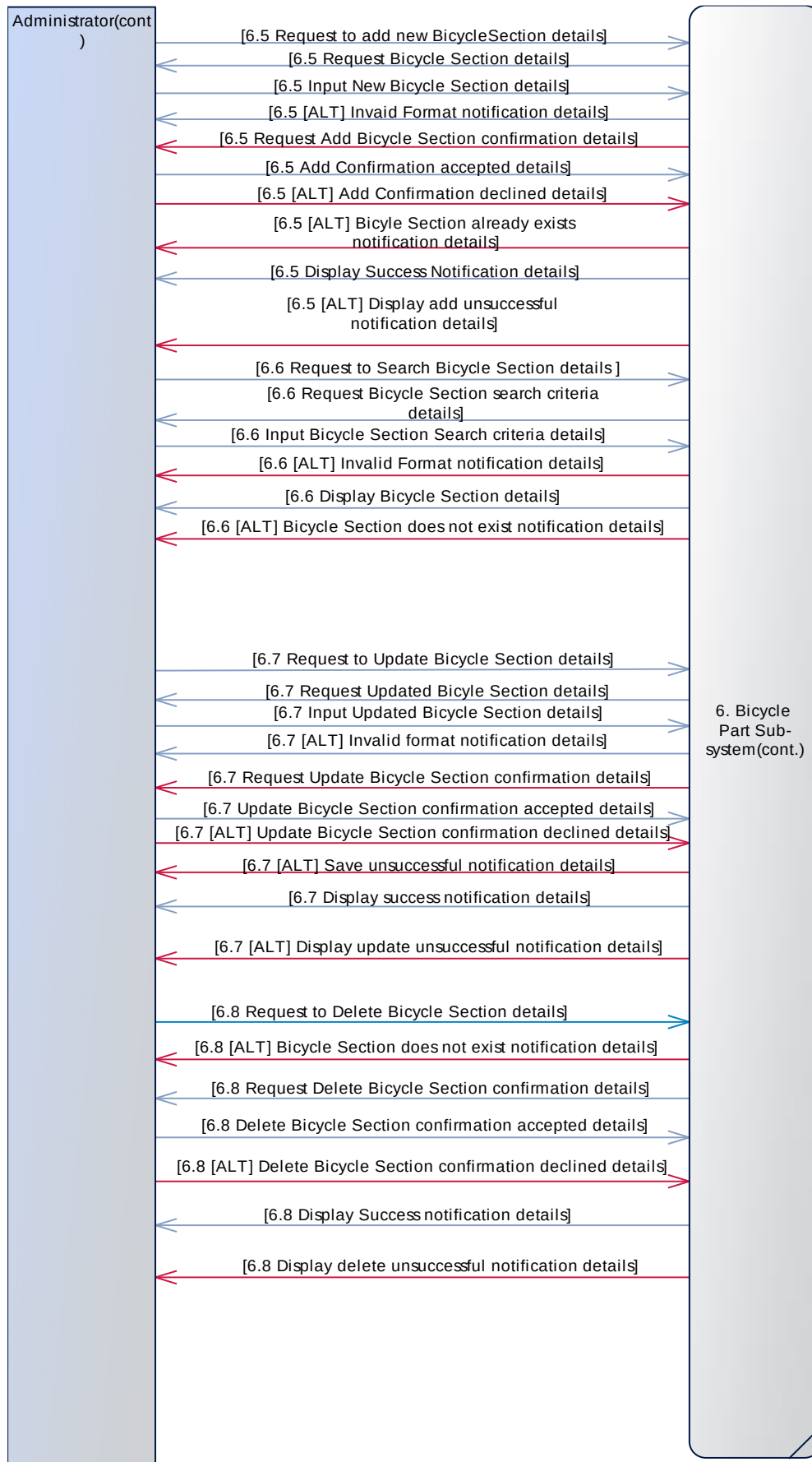


Figure 37 Bicycle Part high level diagram



3.4.2.7 Sub-system 7: Tracking

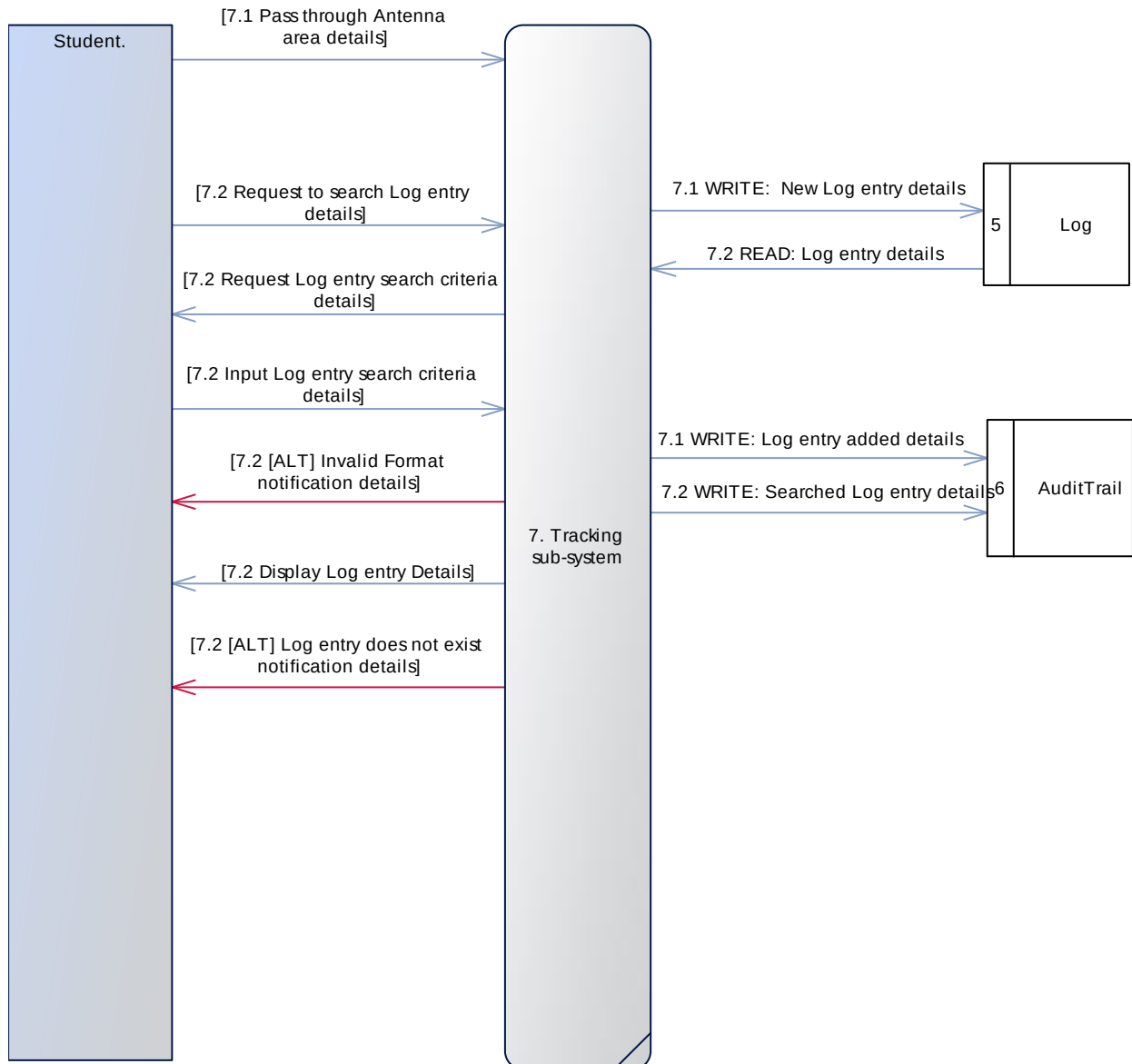
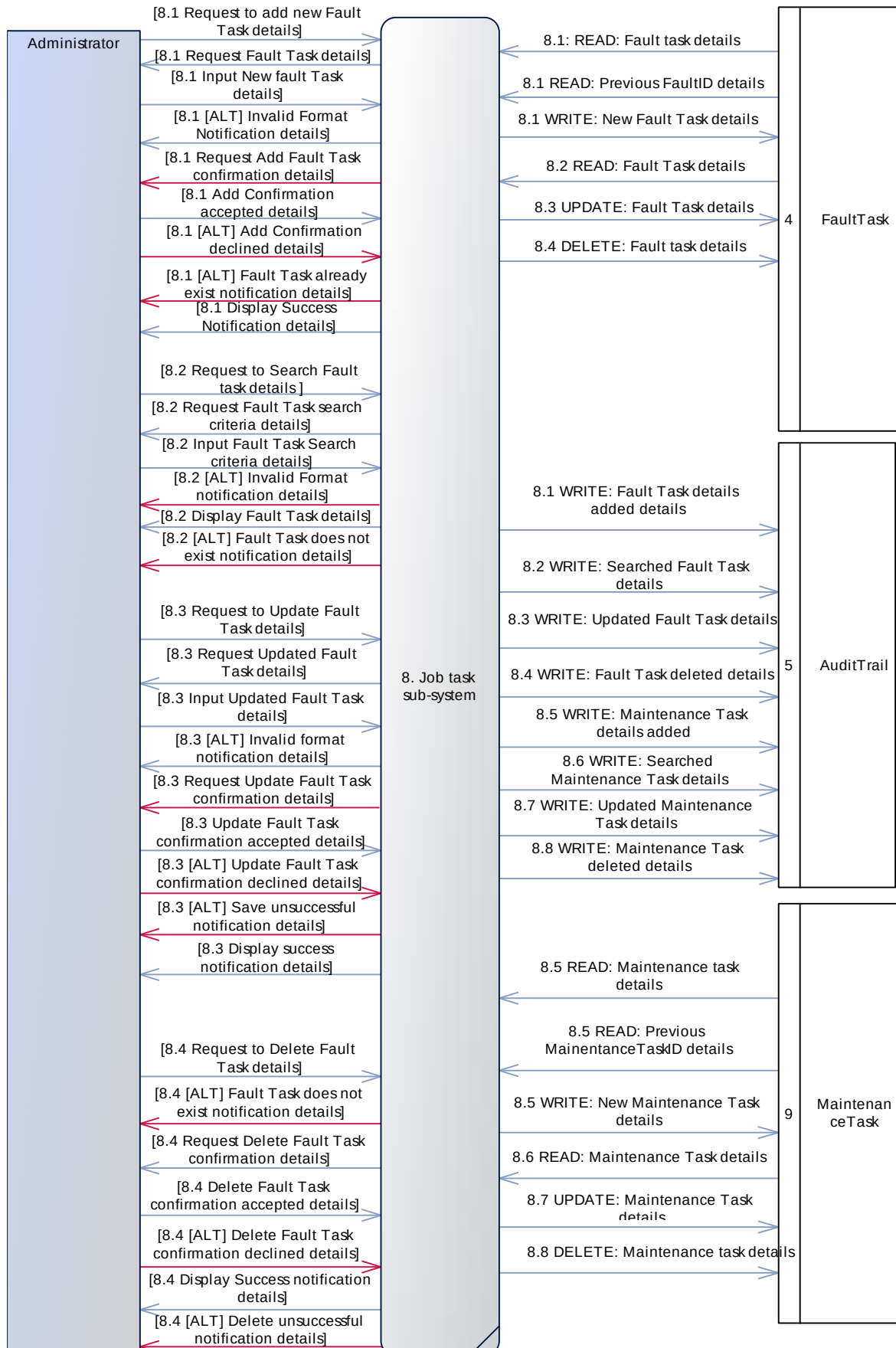


Figure 38 Tracking high level diagram



3.4.2.8 Sub-system 8: Job Tasks



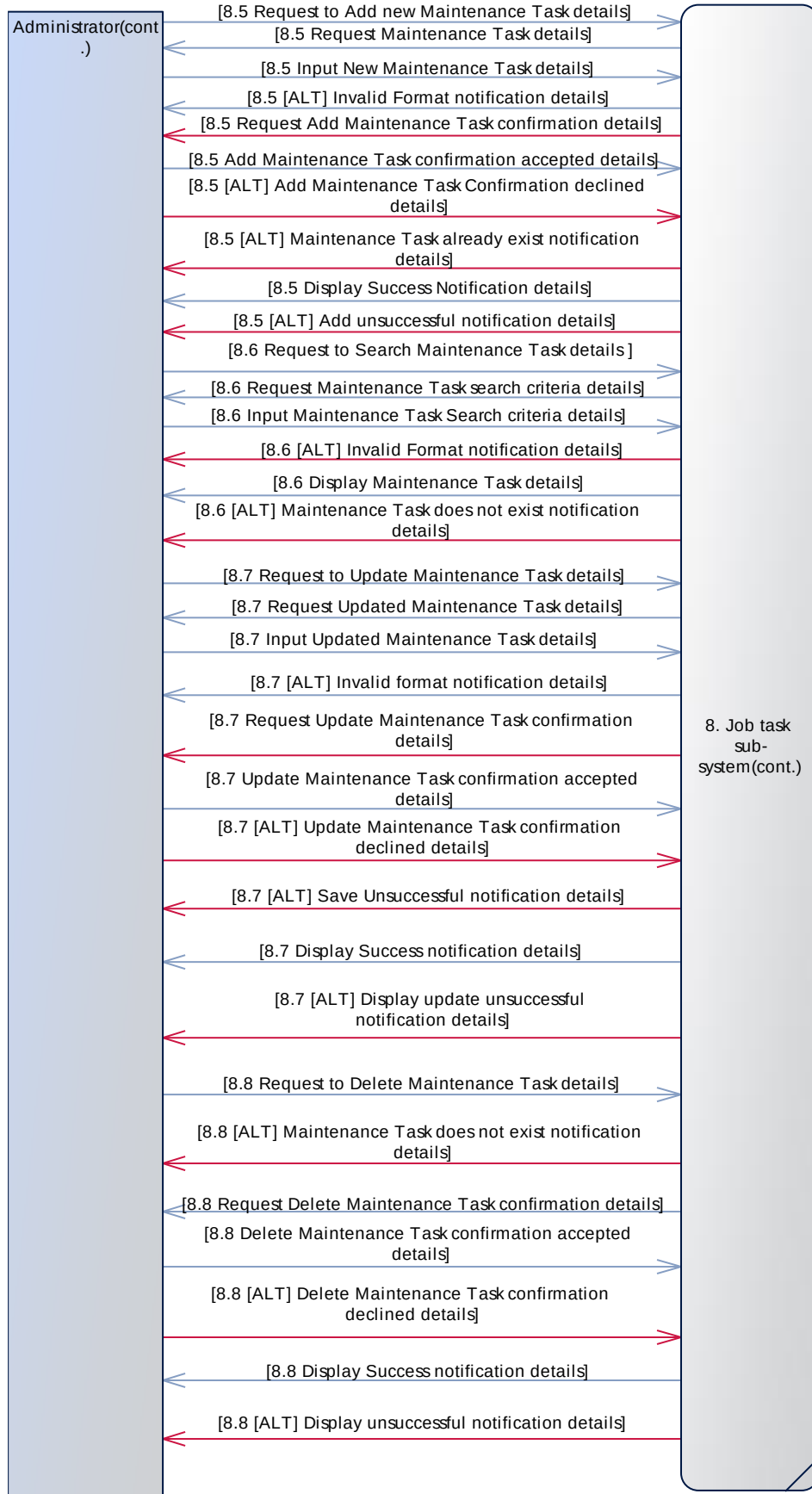


Figure 39 Job Task high level



3.4.2.9 Sub-system 9: School

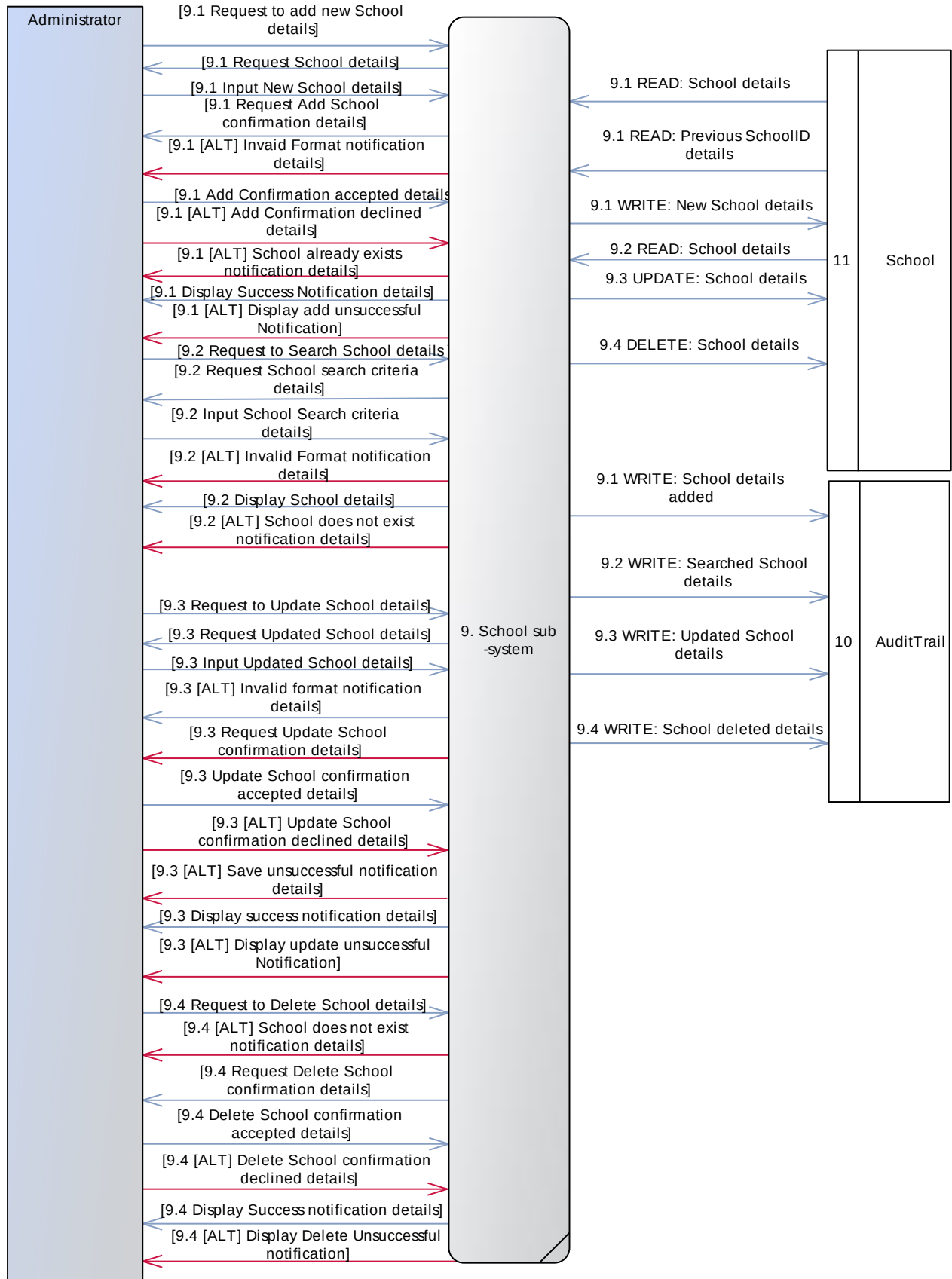


Figure 40 School high level



3.4.2.10 Sub-system 10: Job

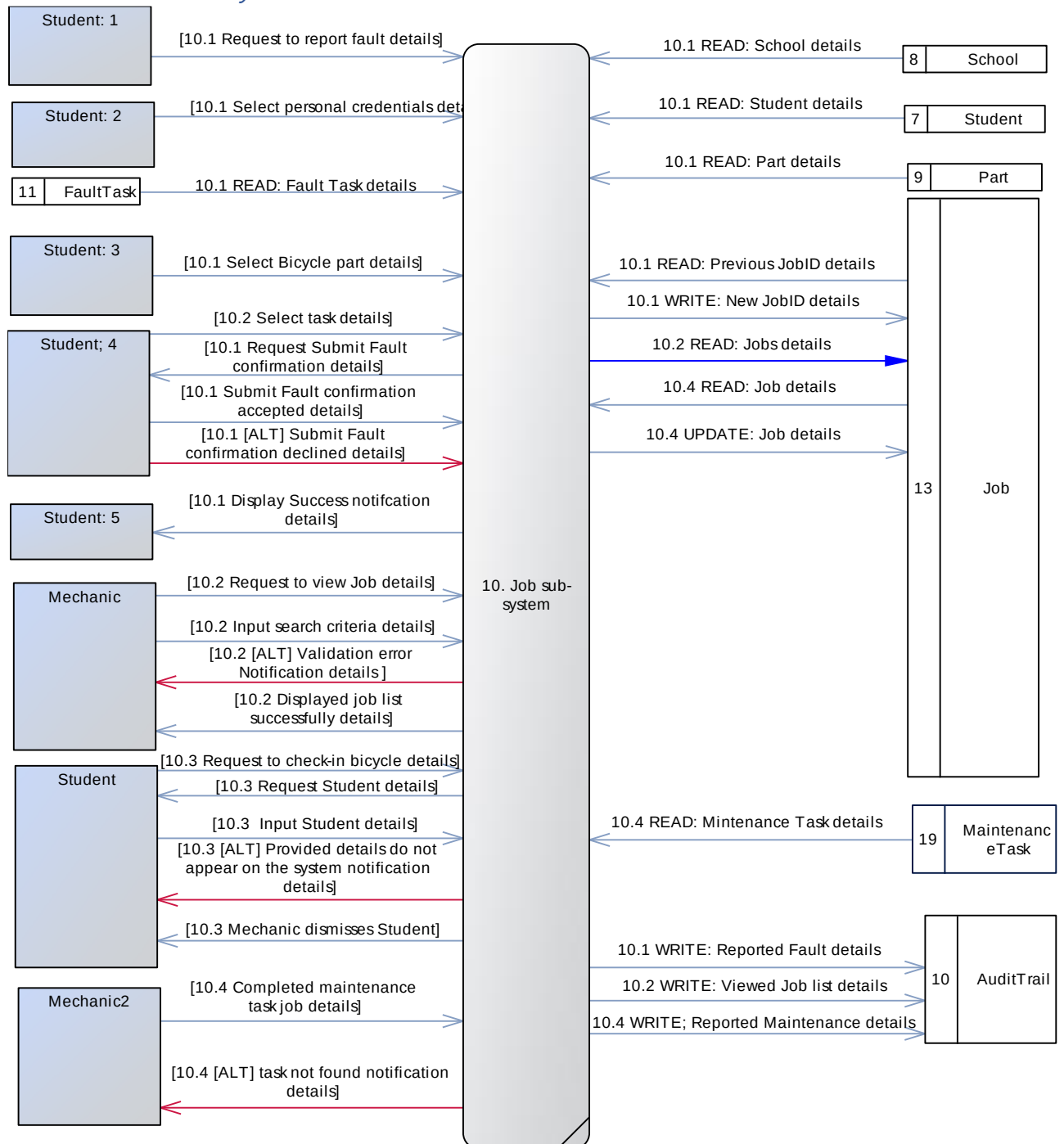
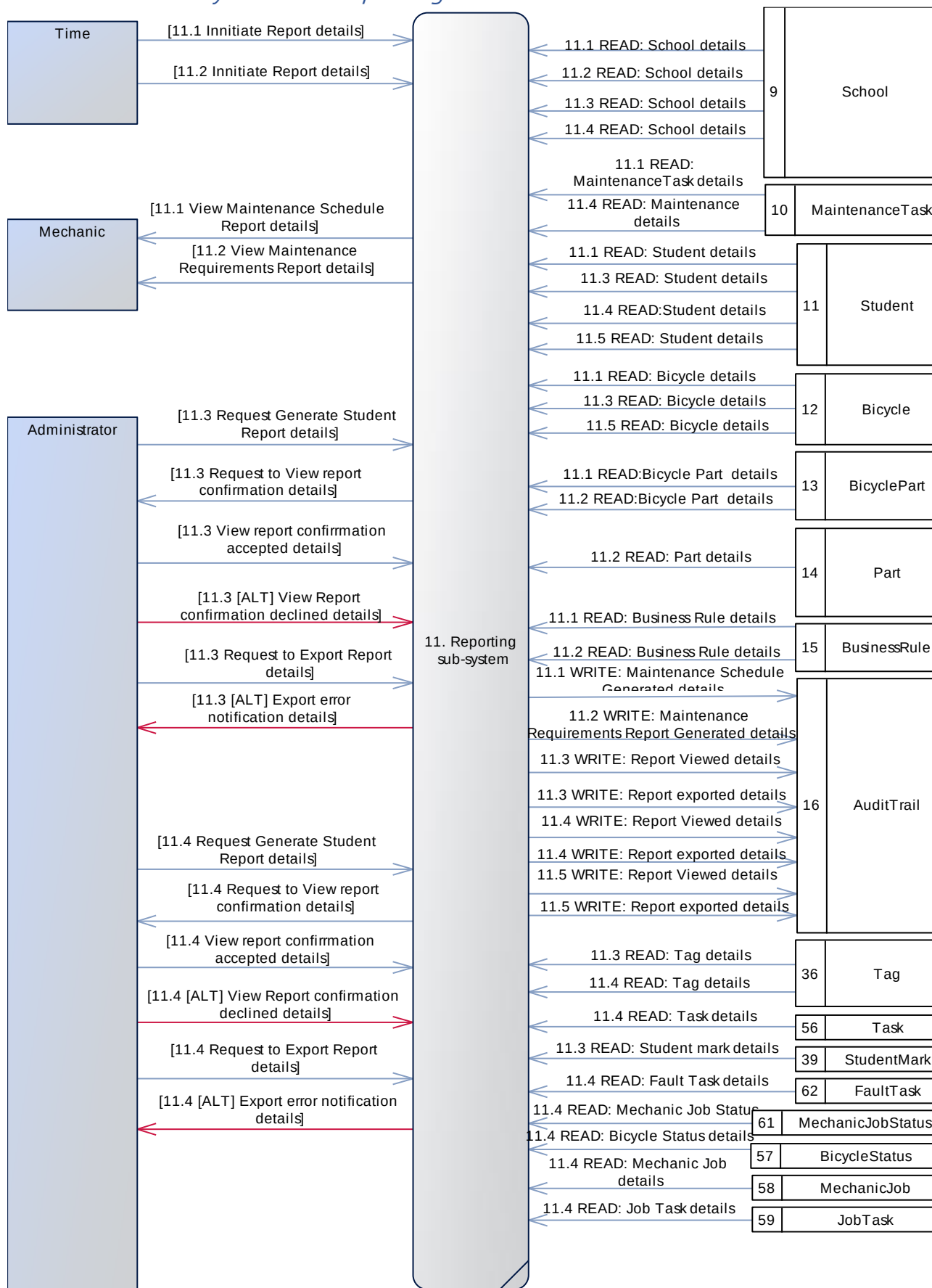


Figure 41 Job high level diagram



3.4.2.11 Sub-system 11: Reporting



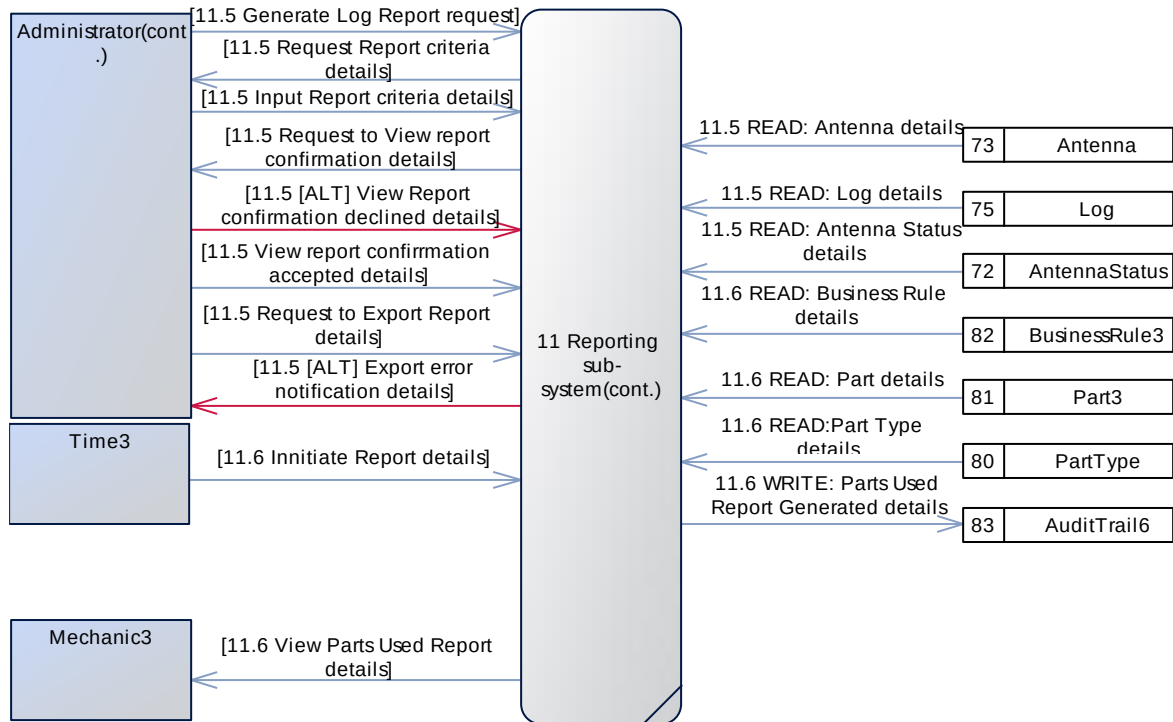


Figure 42 Reporting high level diagram



3.4.2.12 Sub-system 12: Notification

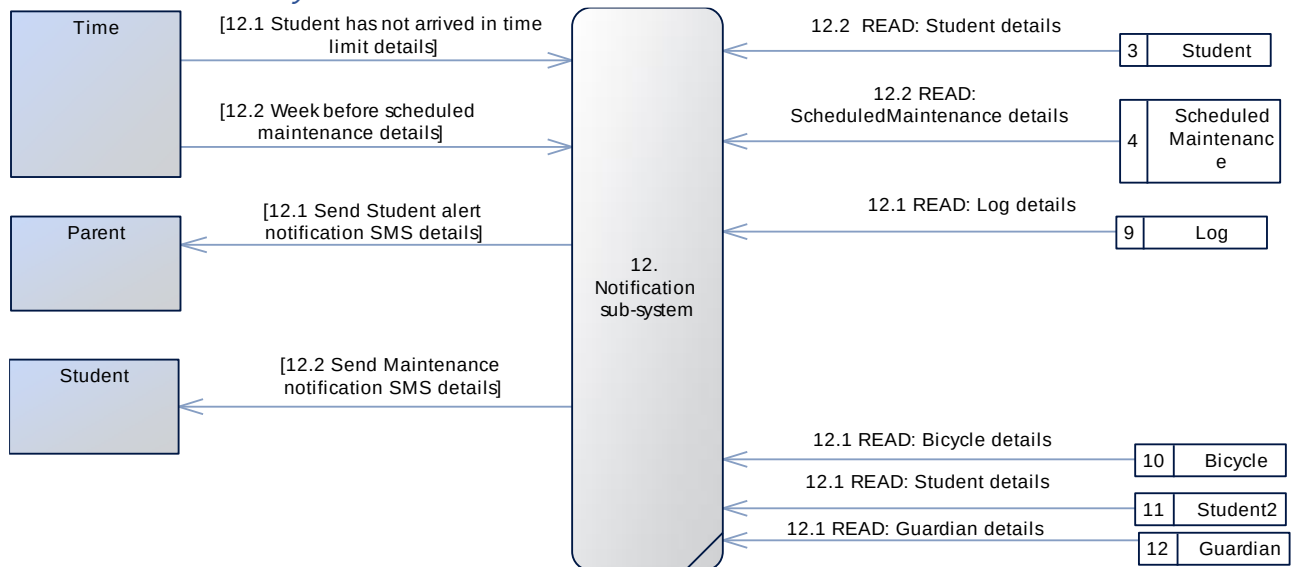
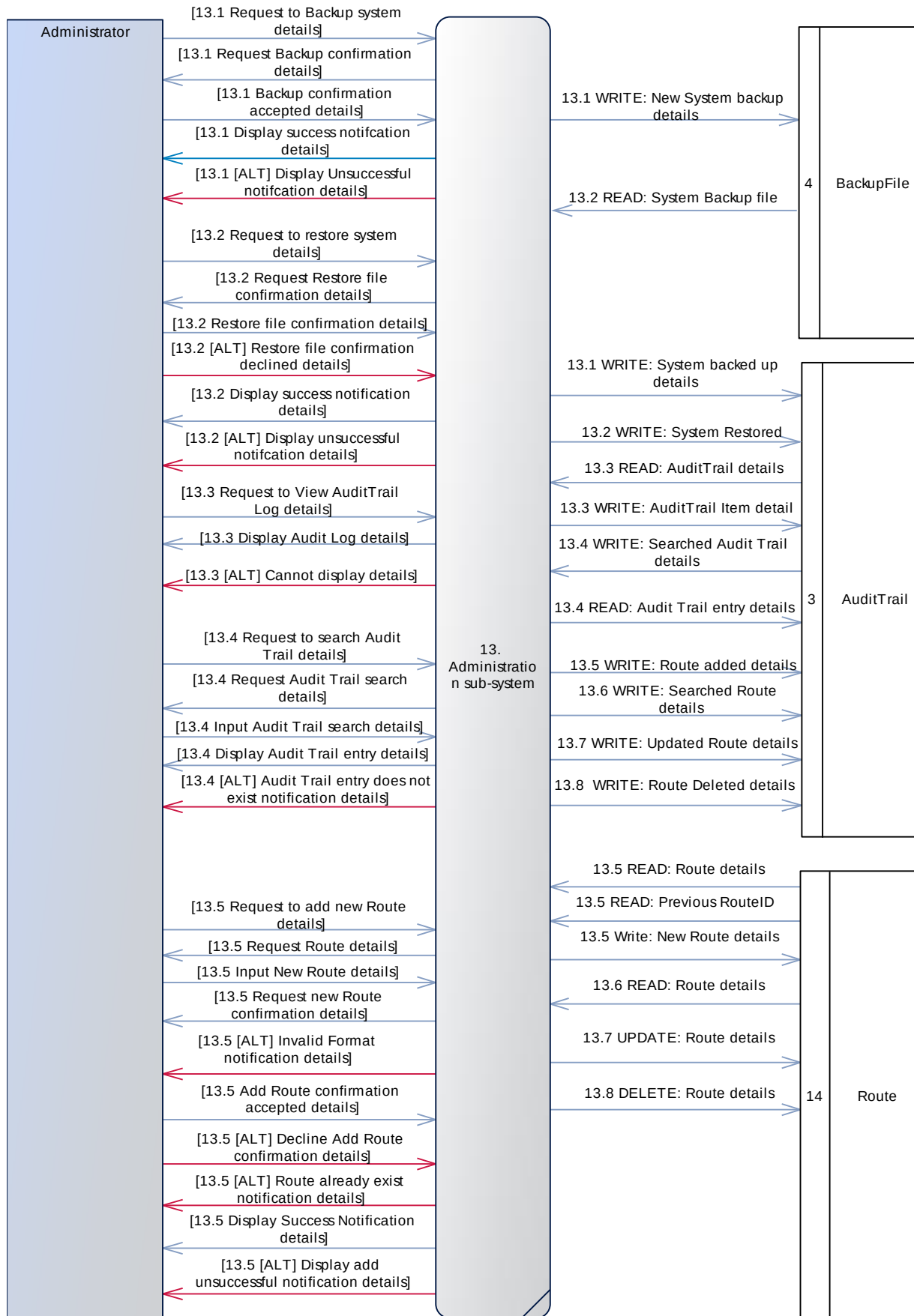


Figure 43 Notification high level diagram



3.4.2.13 Sub-system 13: Administration



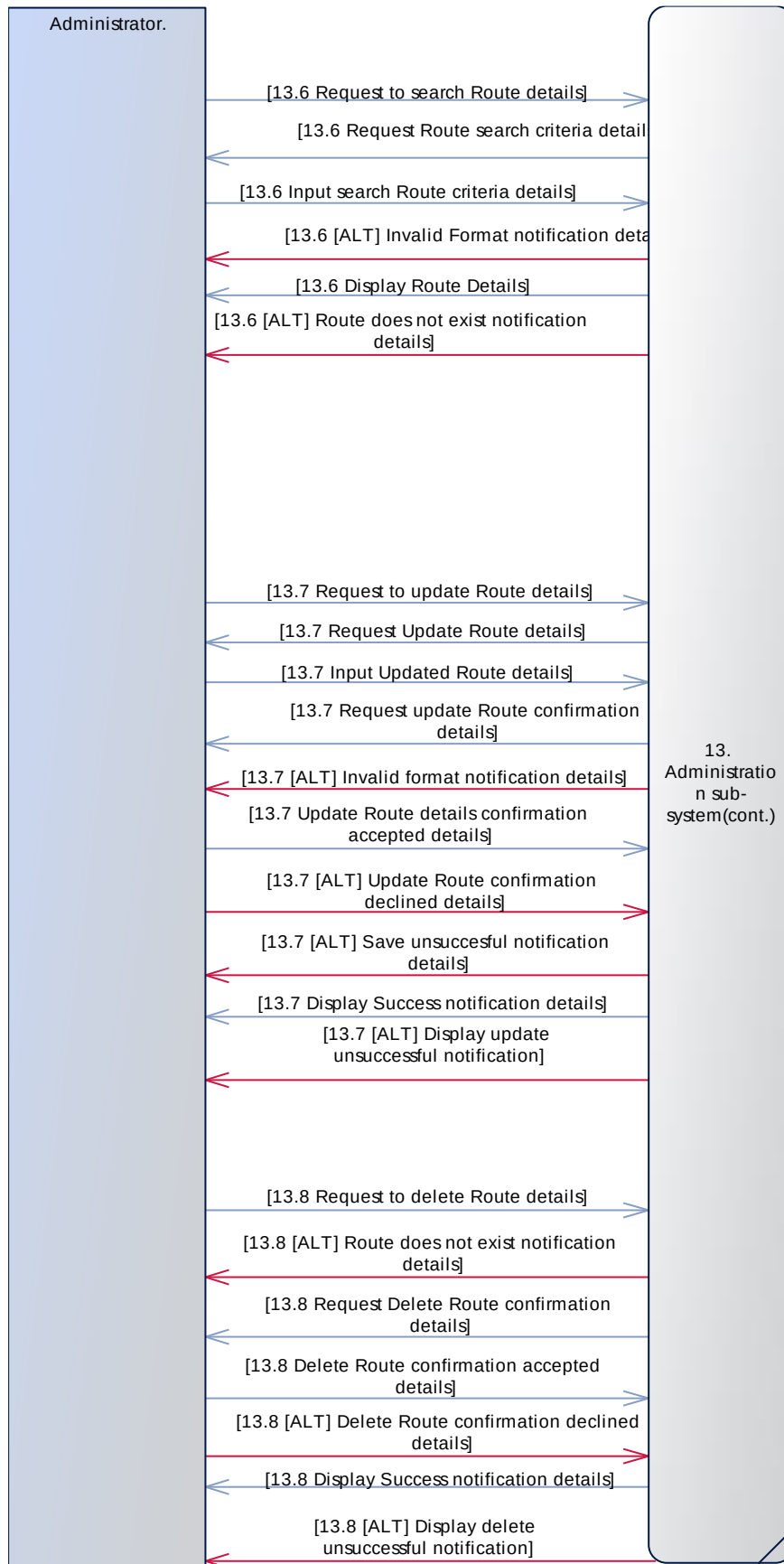


Figure 44 Administration high level diagram



3.4.3 Mid-level Data Flow Diagrams

3.4.3.1 Sub-system 1: User

3.4.3.1.1 Login

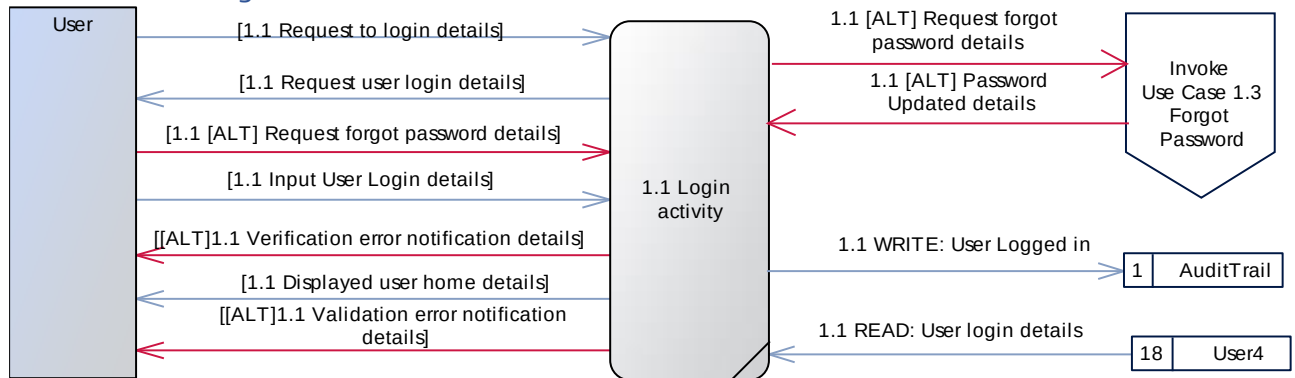


Figure 45 Login mid-level diagram



3.4.3.1.2 Logout

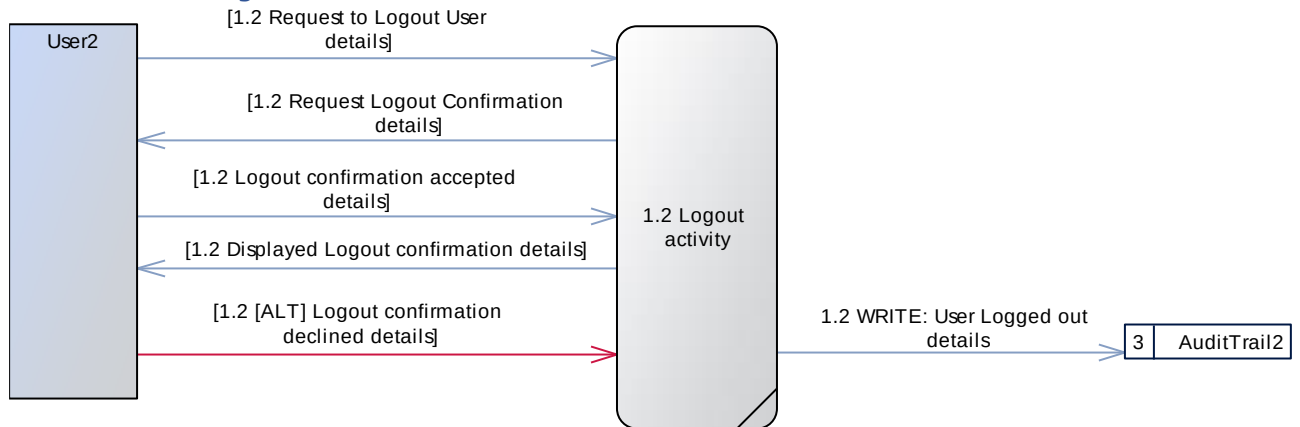


Figure 46 Logout mid-level diagram



3.4.3.1.3 Forgot Password

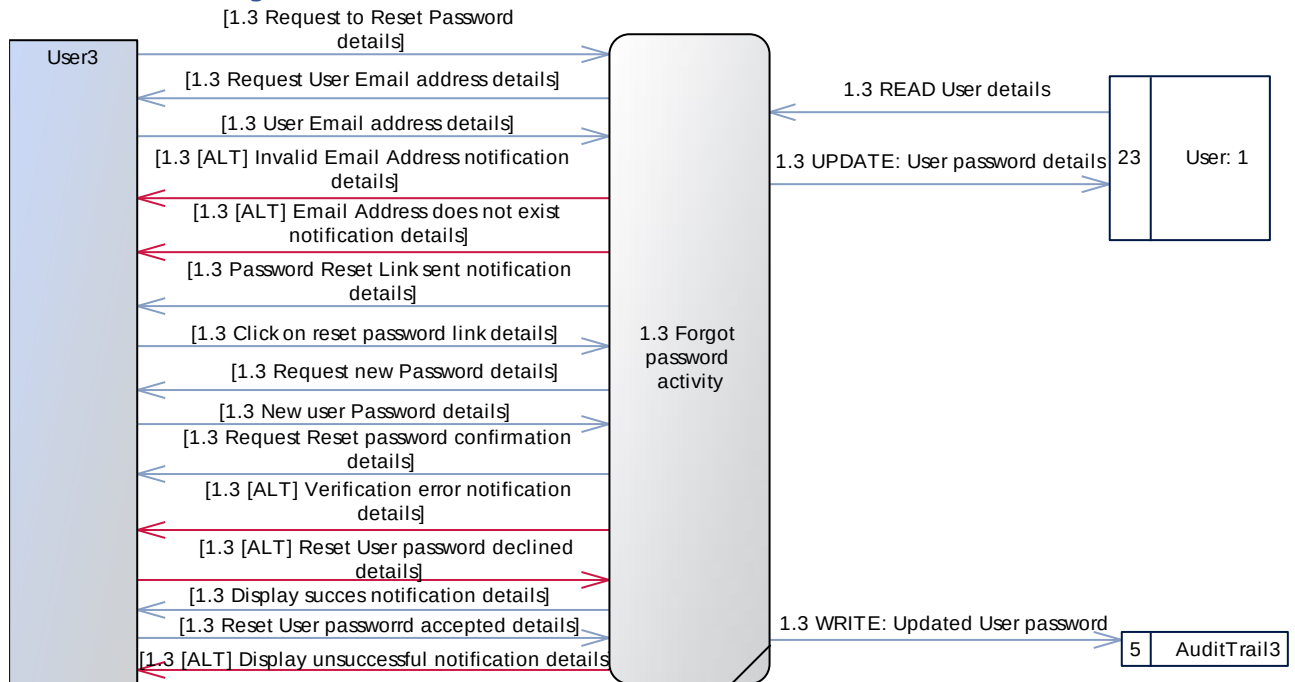


Figure 47 Forgot Password mid-level diagram



3.4.3.1.4 Add User

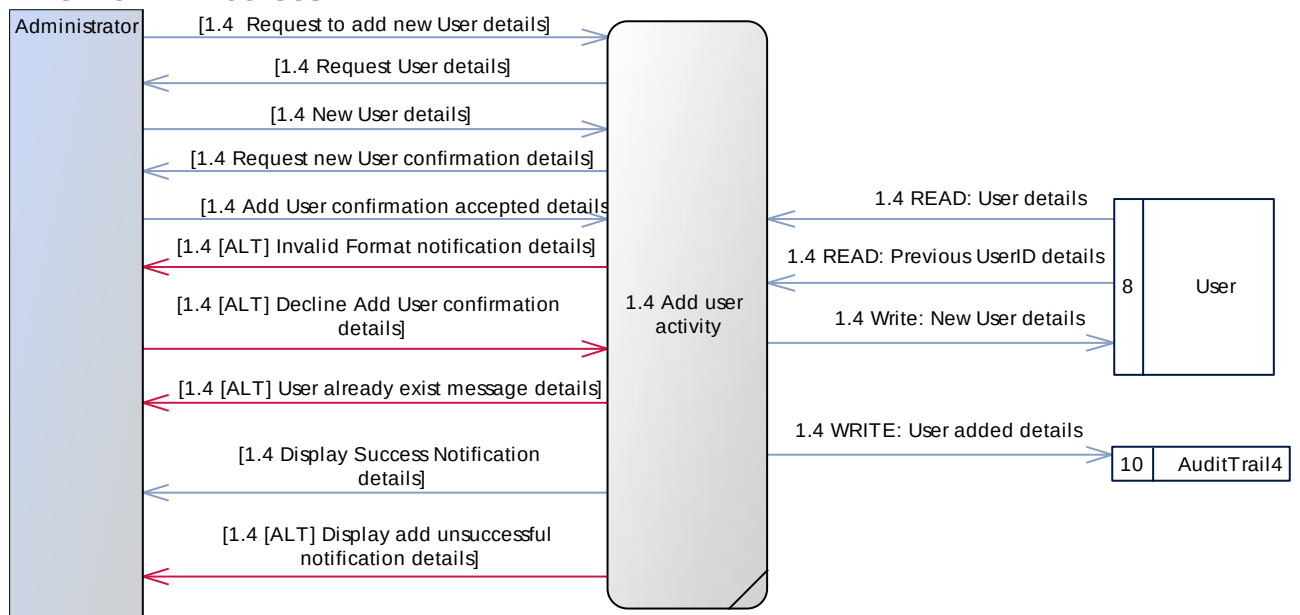


Figure 48 Add User mid-level diagram



3.4.3.1.5 Search User

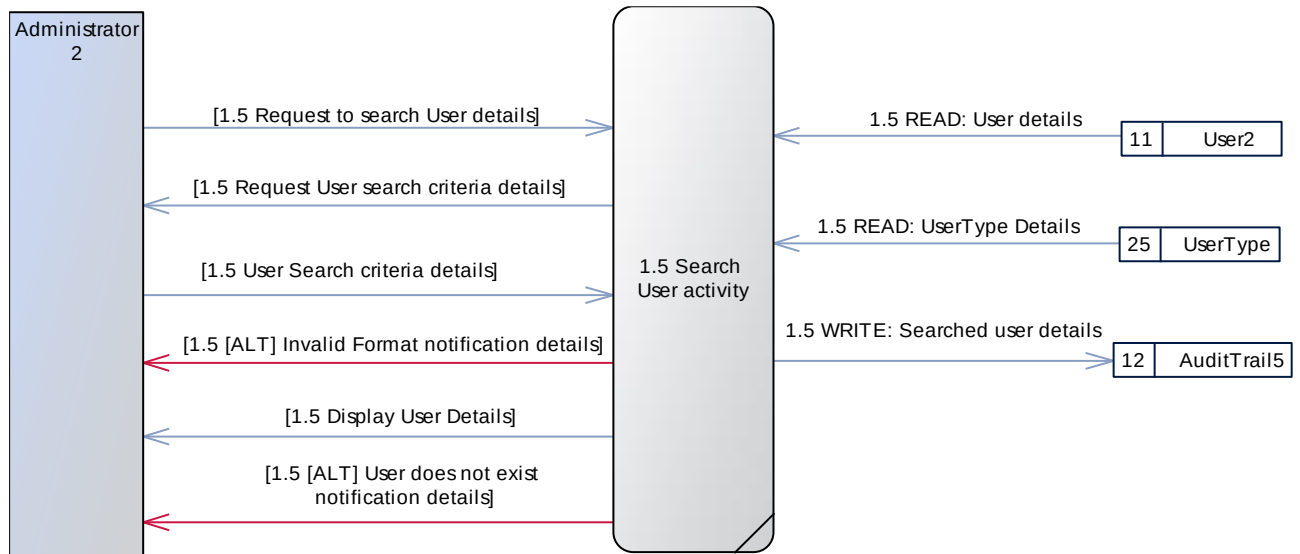


Figure 49 Search User mid-level diagram



3.4.3.1.6 Update User

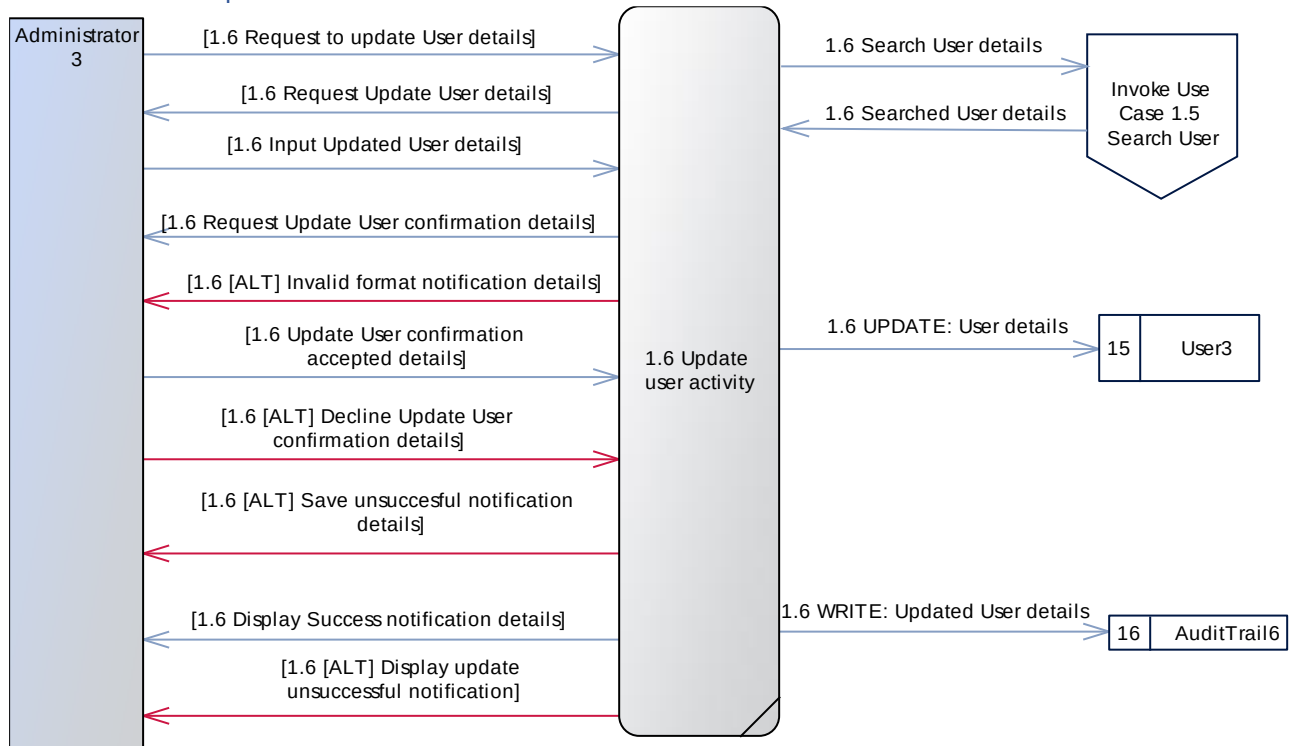


Figure 50 Update User mid-level diagram



3.4.3.1.7 Delete User

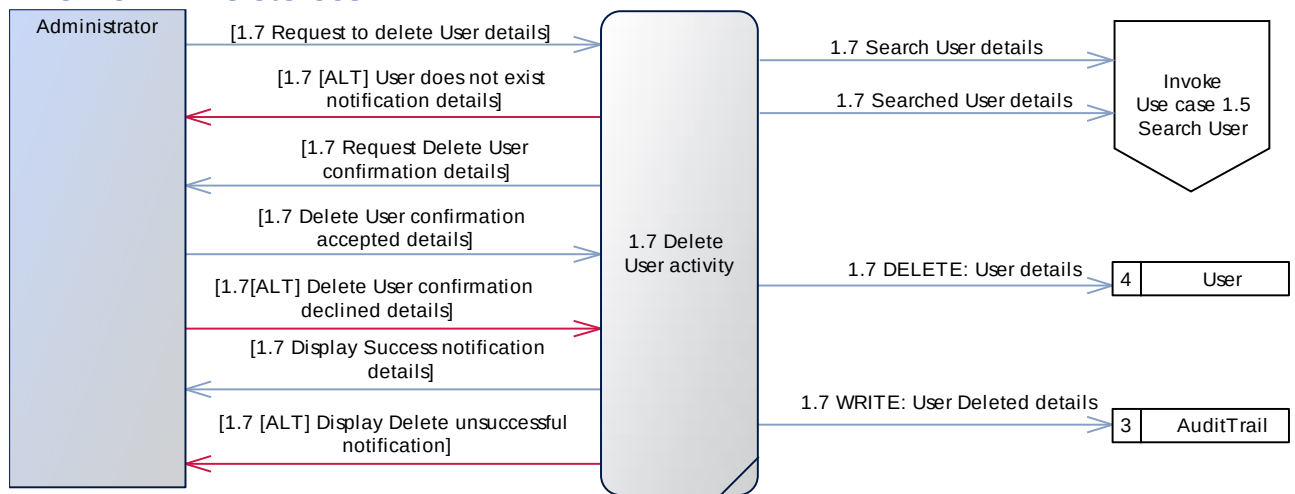


Figure 51 Delete User mid-level diagram



3.4.3.1.8 Add User Type

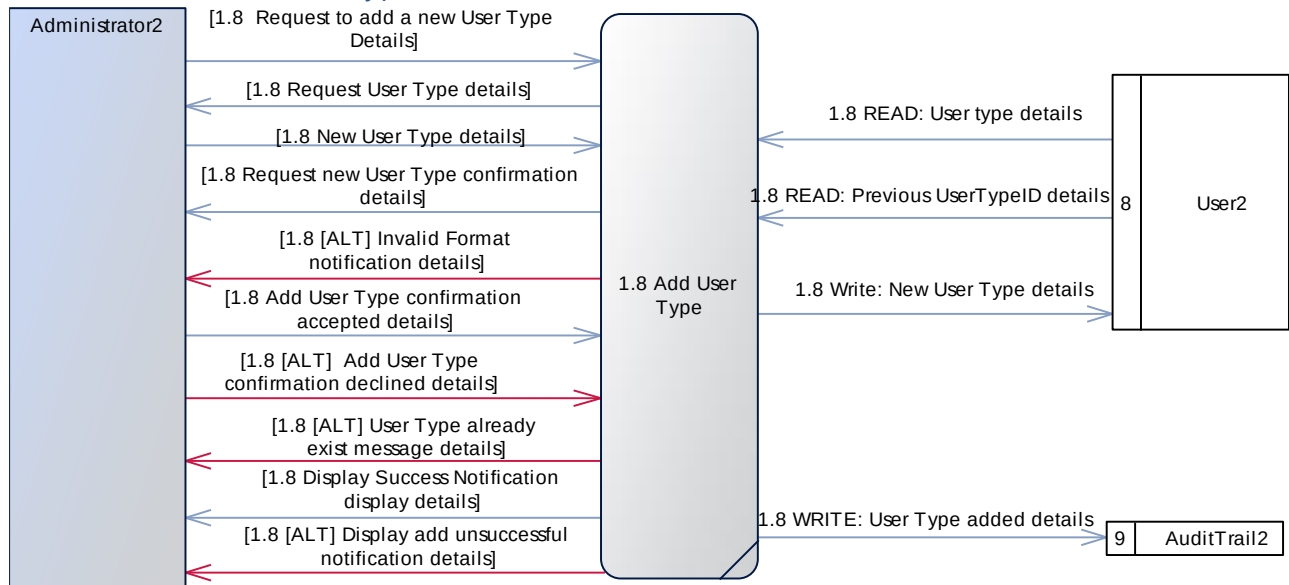


Figure 52 Add User Type mid-level diagram



3.4.3.1.9 Search User Type

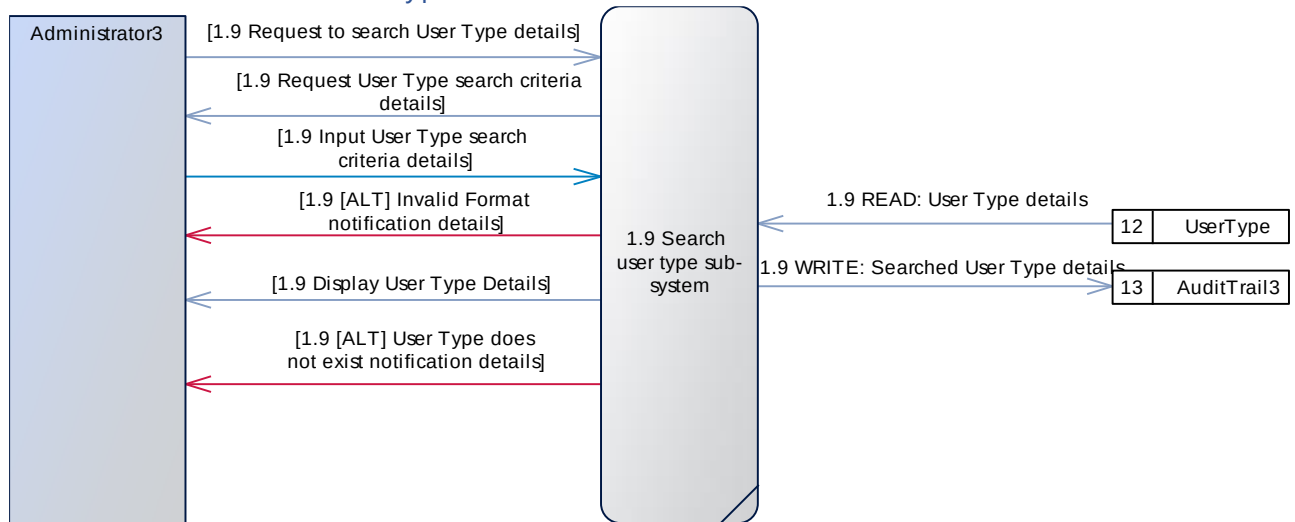


Figure 53 Search User Type mid-level diagram



3.4.3.1.10 Update User Type

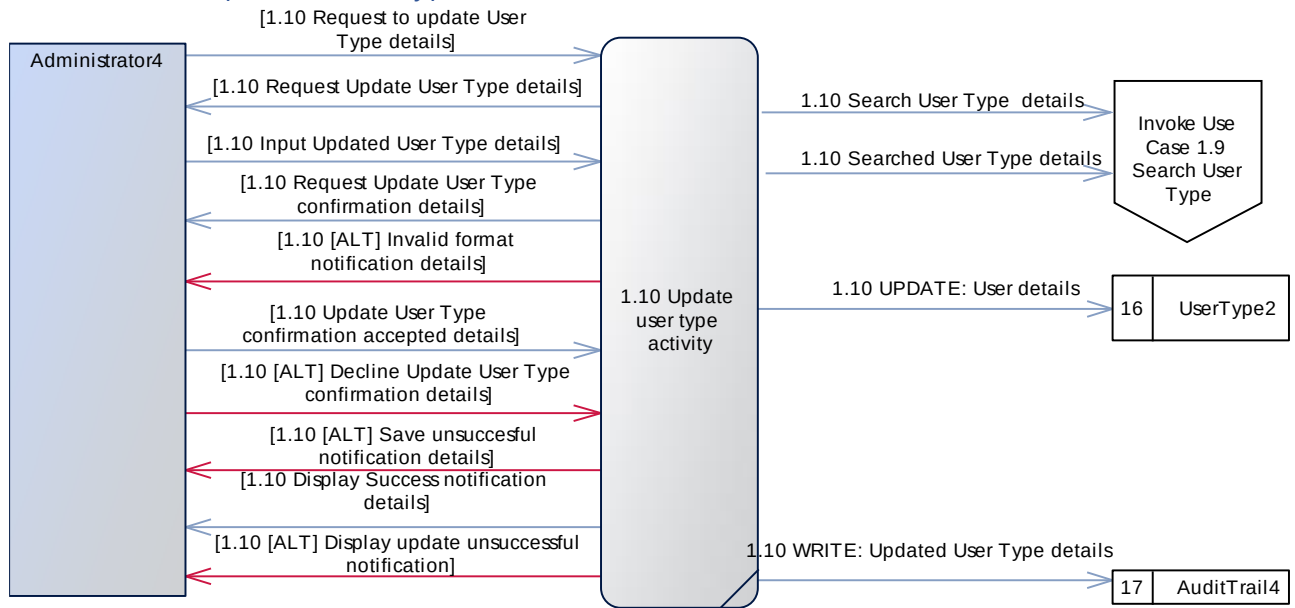


Figure 54 Update User Type mid-level diagram



3.4.3.1.11 Delete User Type

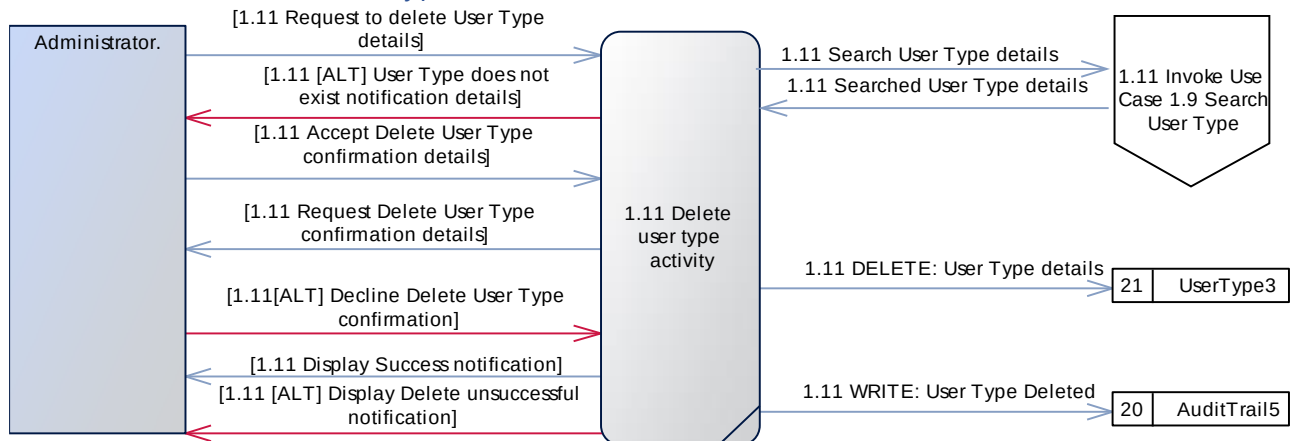


Figure 55 Delete User Type mid-level diagram



3.4.3.2 Sub-system 2: Mechanic Schedule

3.4.3.2.1 Complete Job

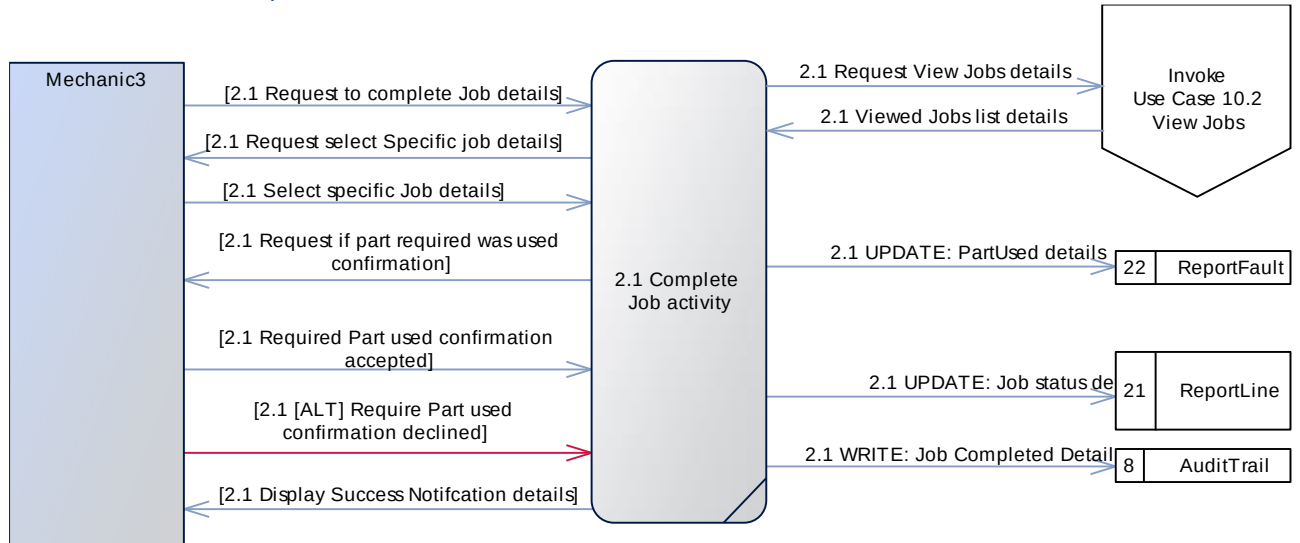


Figure 56 Complete Job mid-level diagram



3.4.3.2.2 View Job Tasks

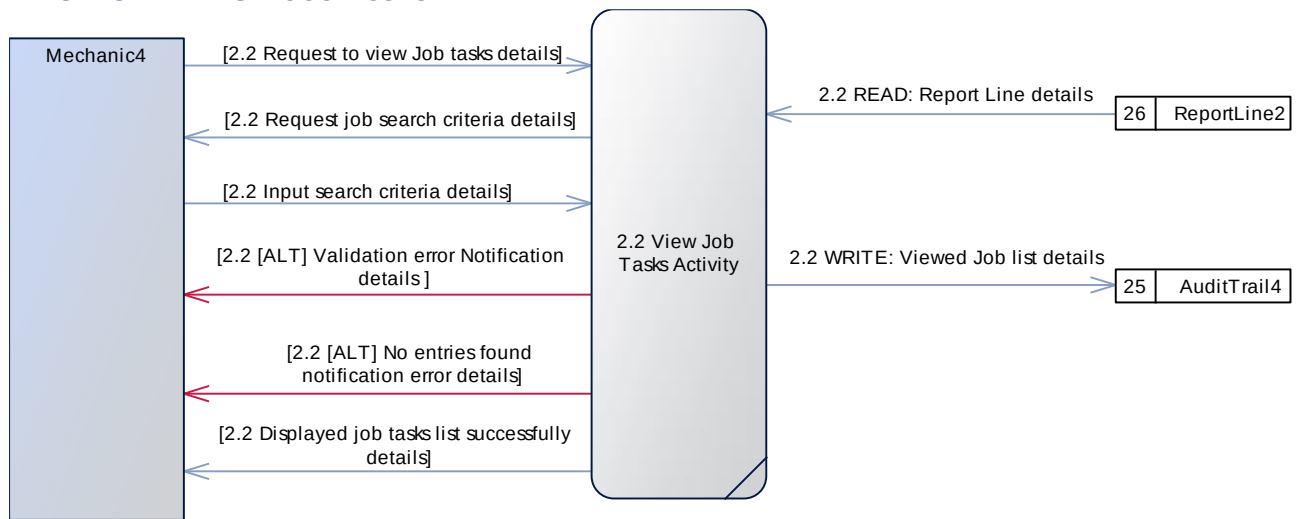


Figure 57 View Job tasks mid-level diagram



3.4.3.3 Sub-system 3: Tracking Hardware

3.4.3.3.1 Add New Antenna

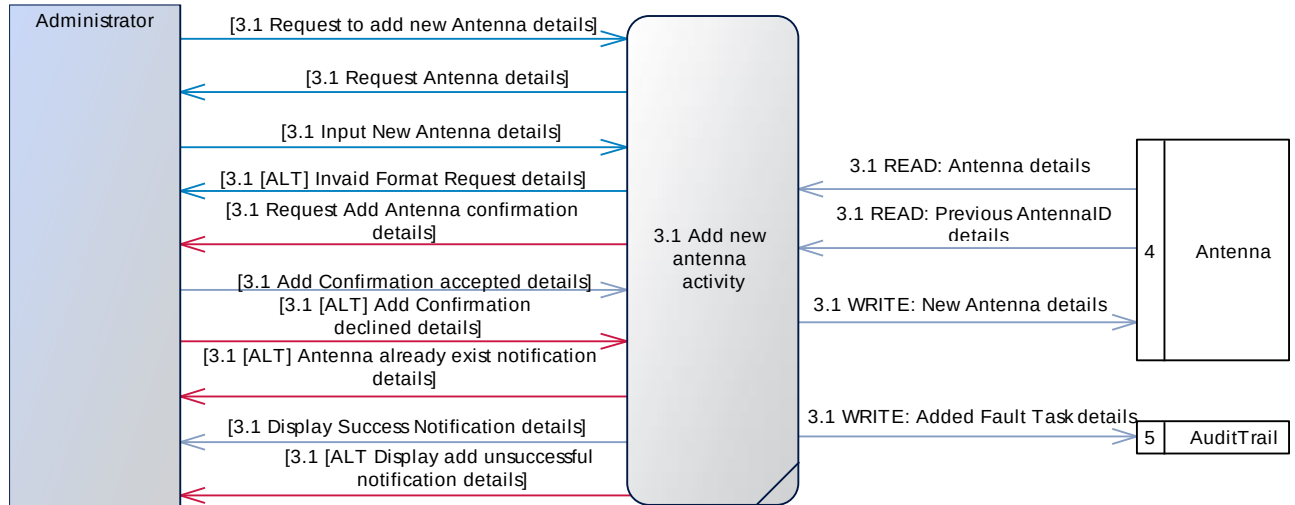


Figure 58 Add new antenna mid-level diagram



3.4.3.3.2 Search Antenna

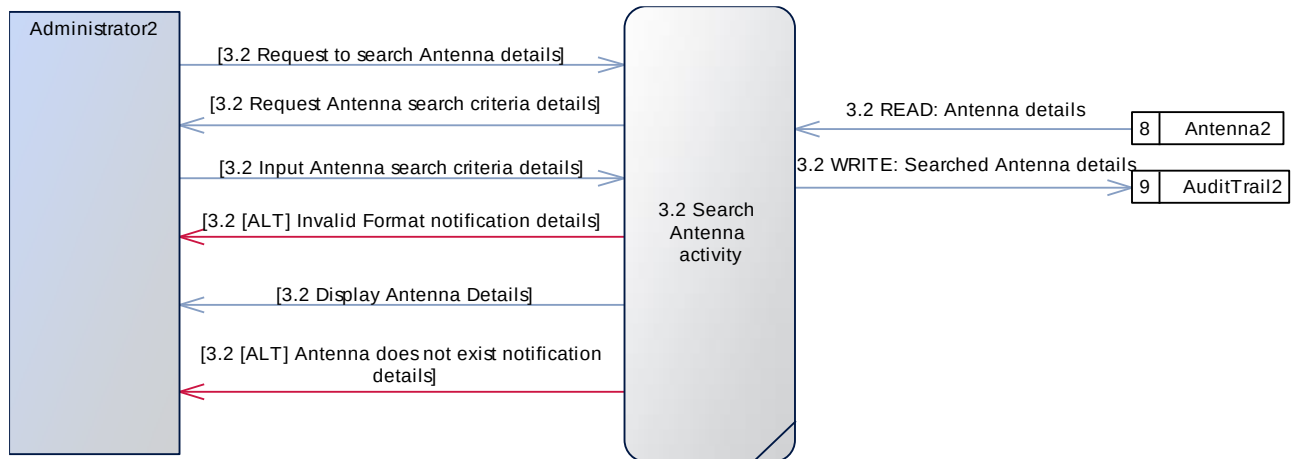


Figure 59 Search Antenna mid-level diagram



3.4.3.3.3 Update Antenna

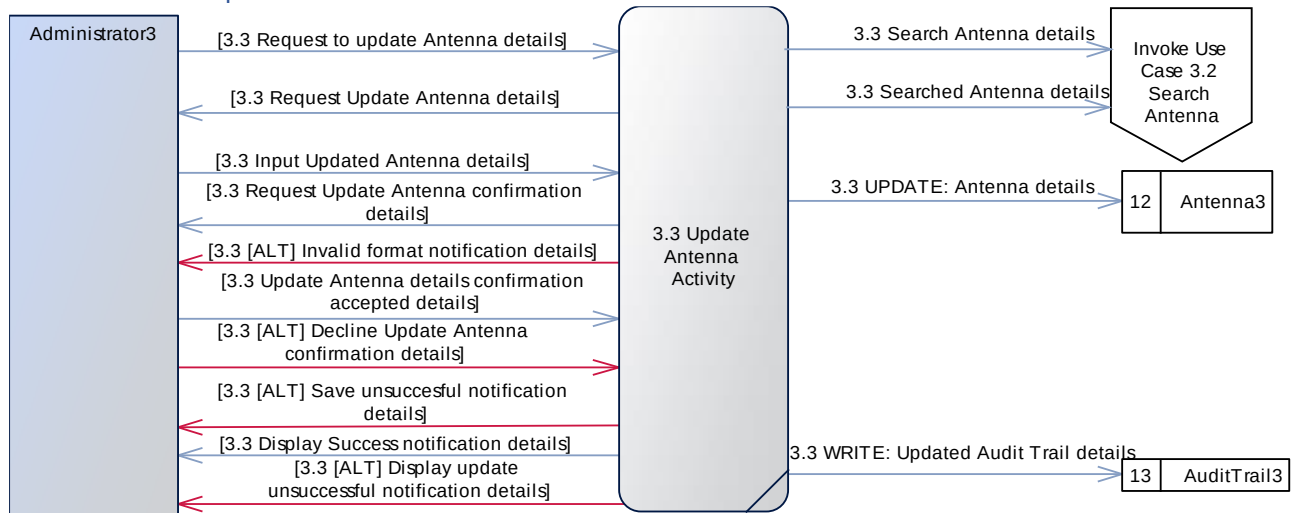


Figure 60 Update Antenna mid-level diagram



3.4.3.3.4 Delete Antenna

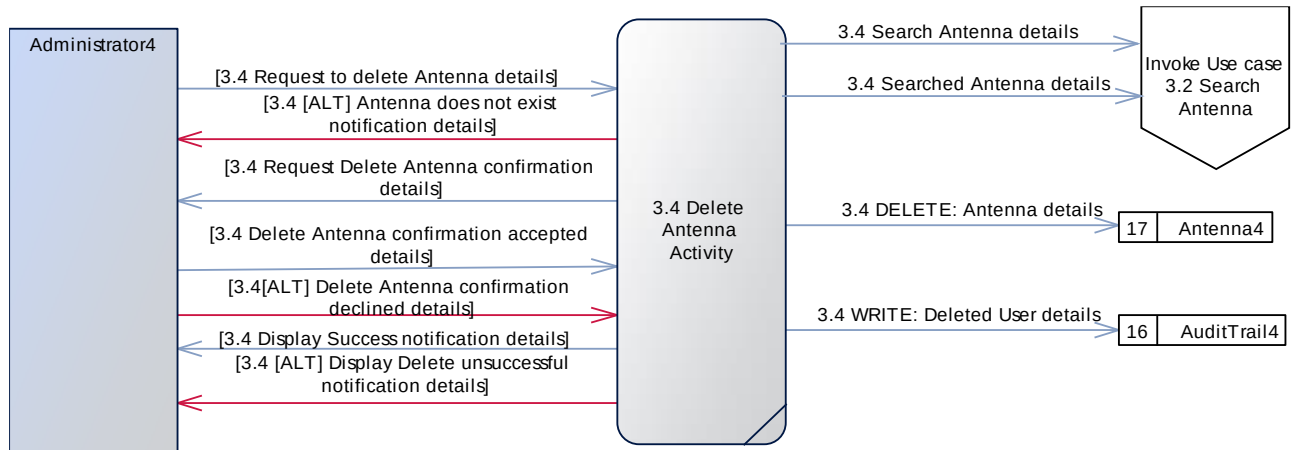


Figure 61Delete Antenna mid-level diagram



3.4.3.3.5 Add New Tag

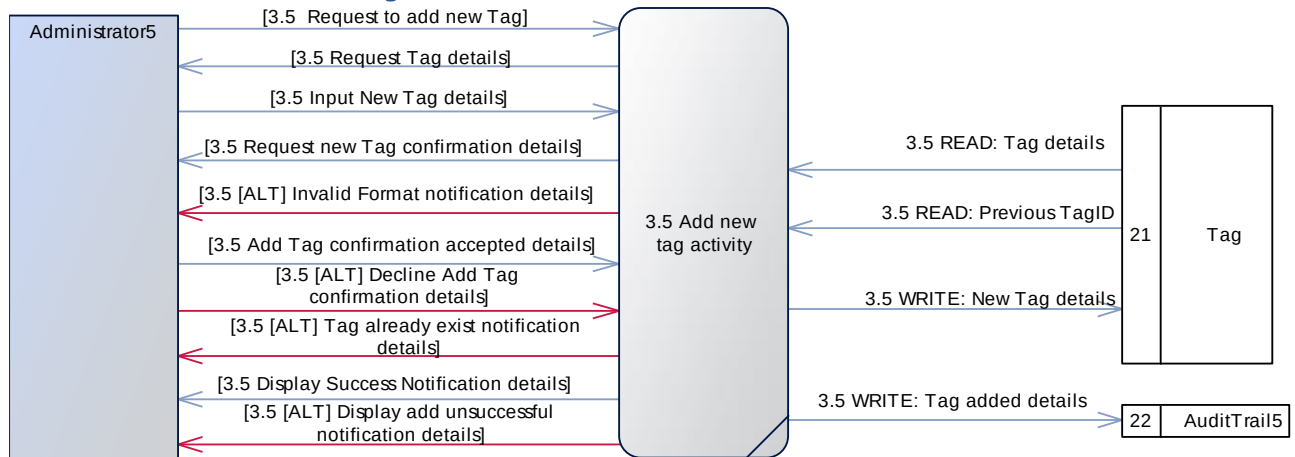


Figure 62 Add new tag mid-level diagram



3.4.3.3.6 Search Tag

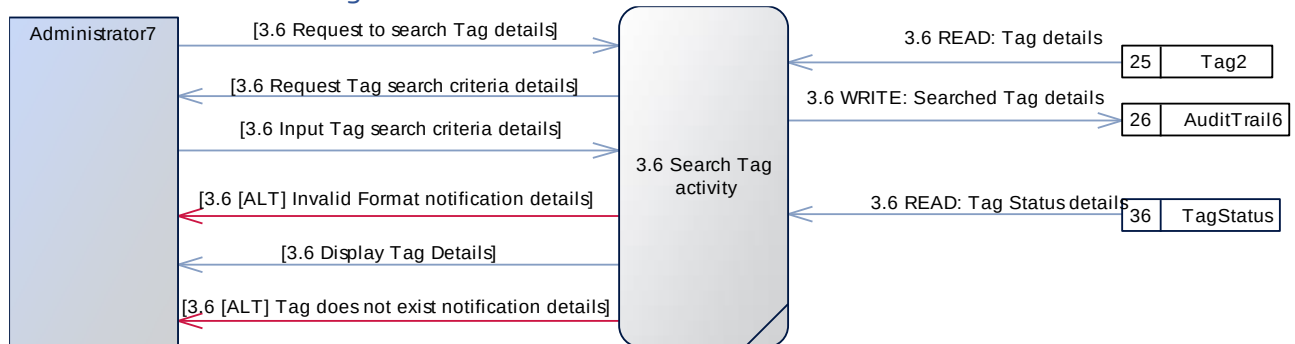


Figure 63 Search Tag mid-level diagram



3.4.3.3.7 Update Tag

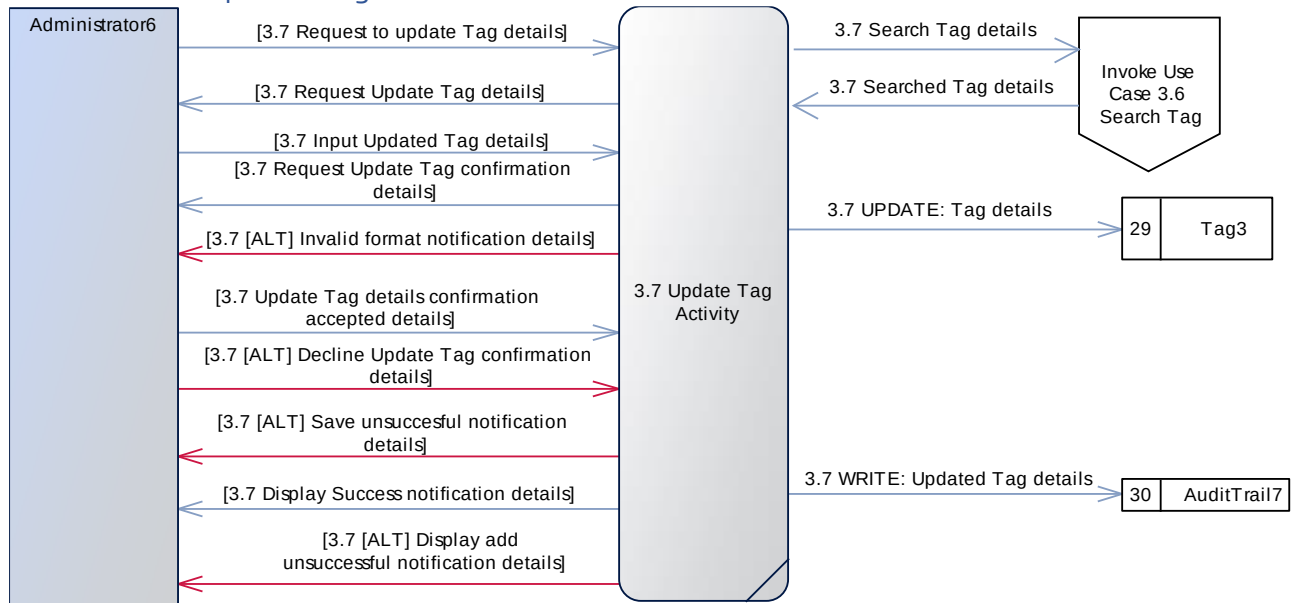


Figure 64 Update Tag mid-level diagram



3.4.3.3.8 Delete Tag

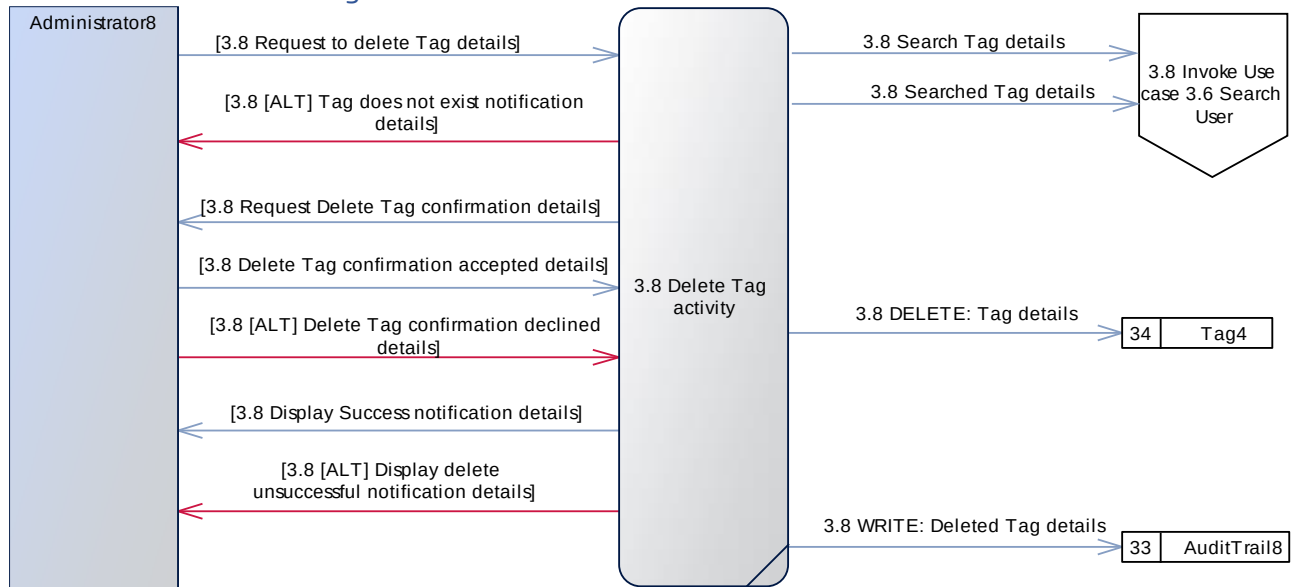


Figure 65 Delete Tag mid-level diagram



3.4.3.4 Sub-system 4: Student

3.4.3.4.1 Add Student

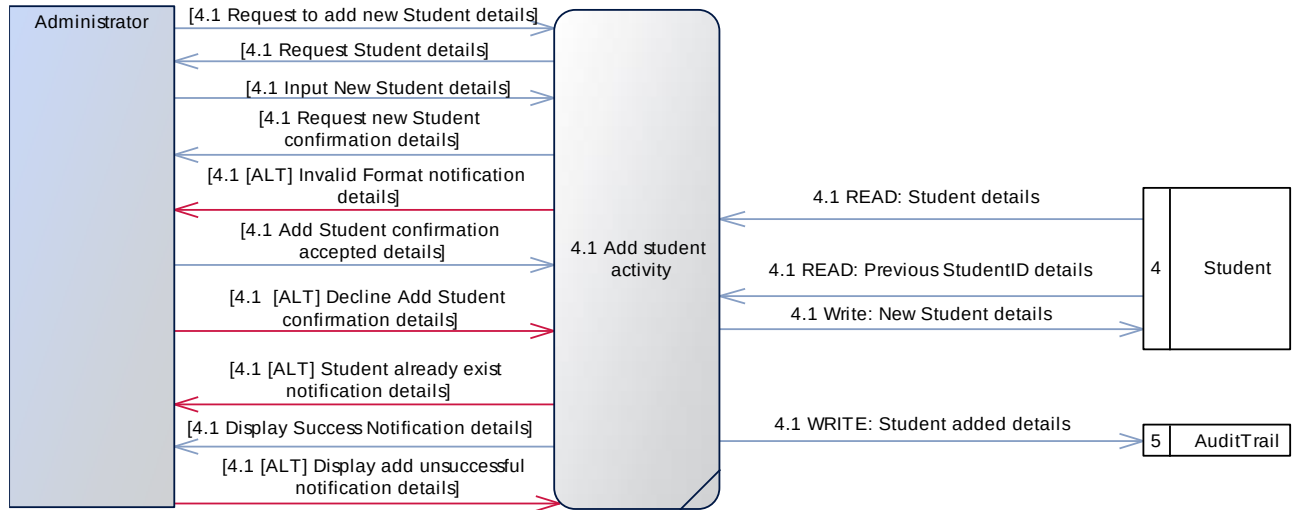


Figure 66 Add Student mid-level diagram



3.4.3.4.2 Search Student

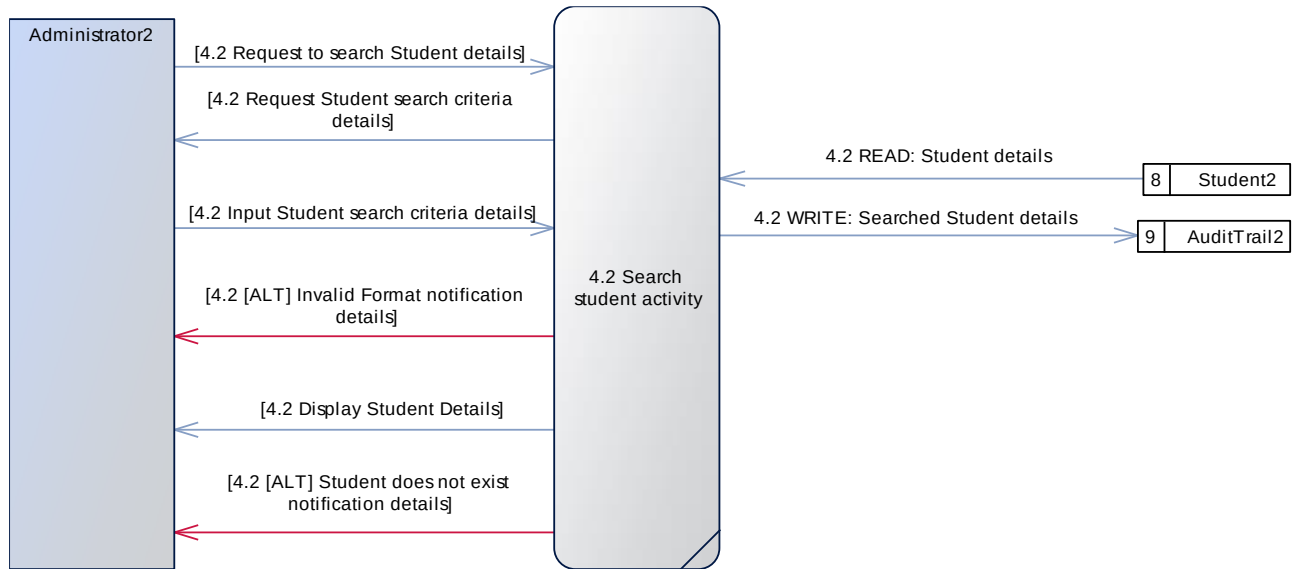


Figure 67 Search Student Mid-level diagram



3.4.3.4.3 Update Student

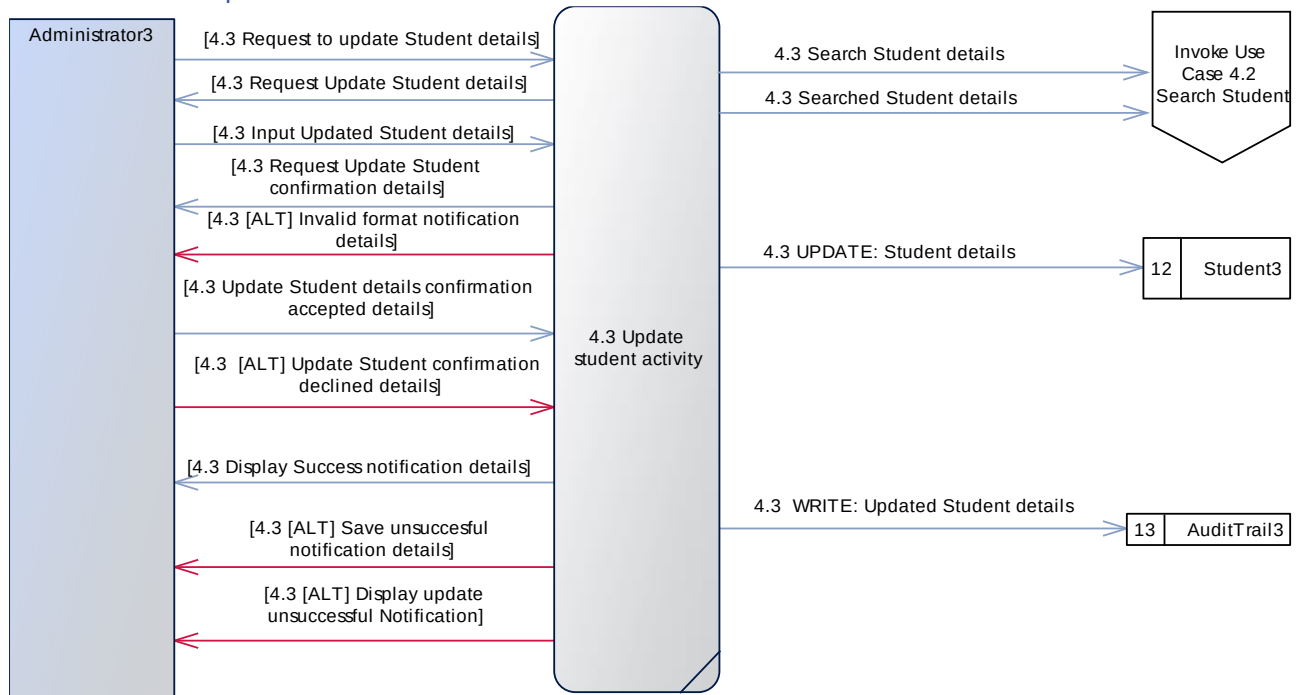


Figure 68 Update Student mid-level diagram



3.4.3.4.4 Delete Student

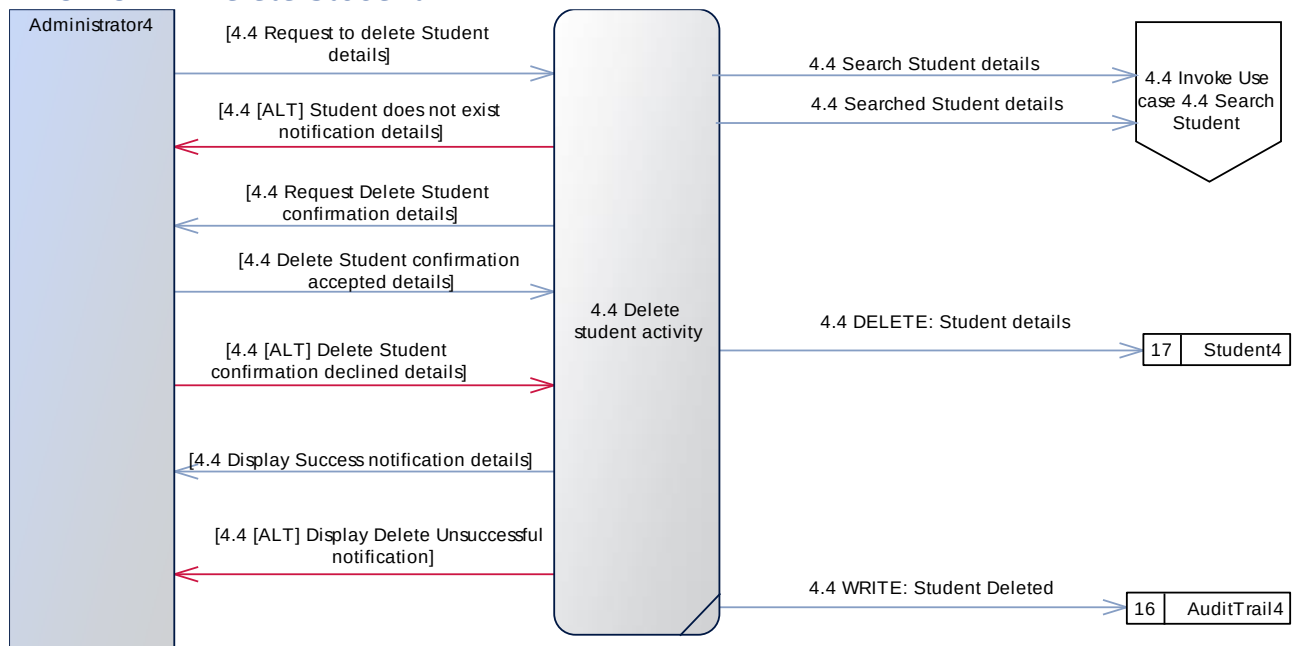


Figure 69 Delete Student mid-level diagram



3.4.3.4.5 Capture Student Marks

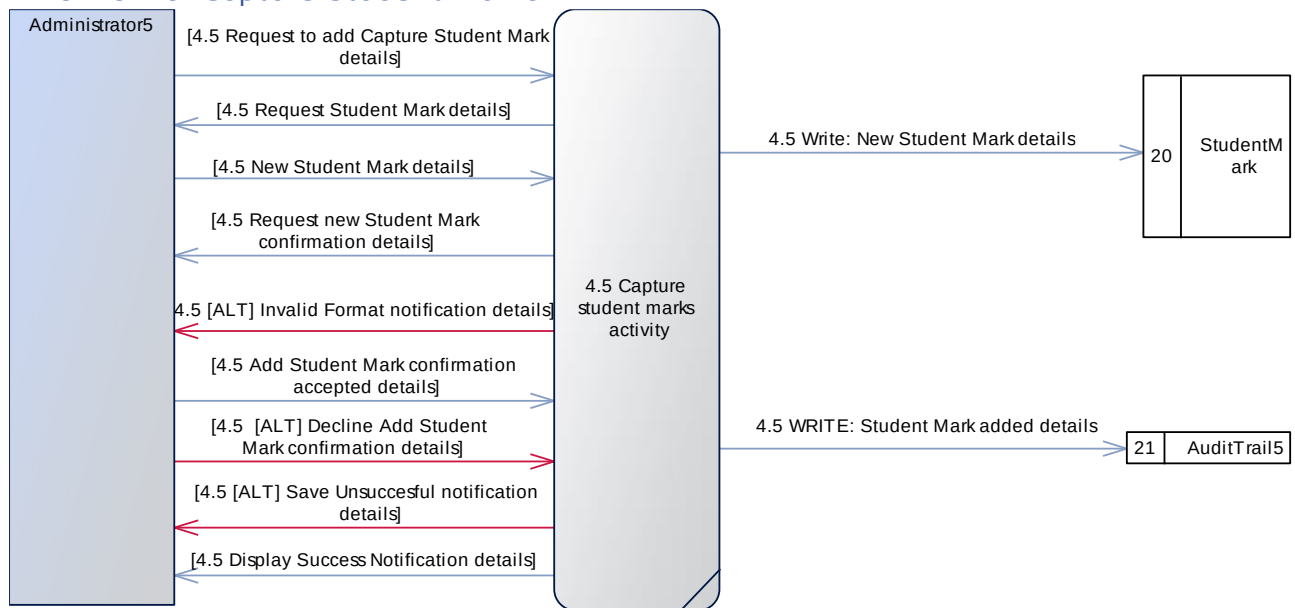


Figure 70 Capture Student Marks mid-level diagram



3.4.3.4.6 Search Student Marks

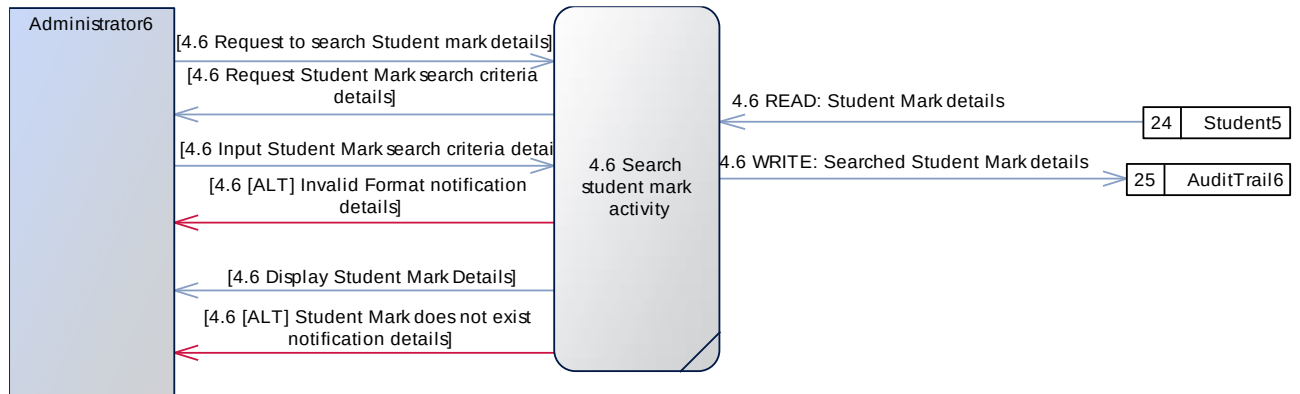


Figure 71 Search Student Marks mid-level diagram



3.4.3.4.7 Add Mark Type

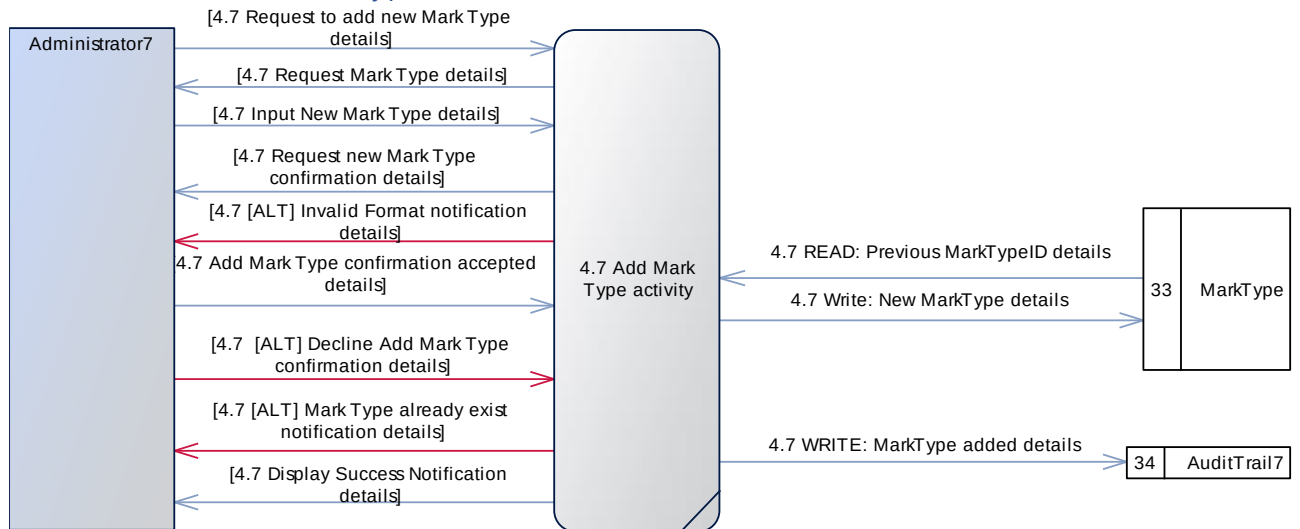


Figure 72 Add Mark Type mid-level diagram



3.4.3.4.8 Search Mark Type

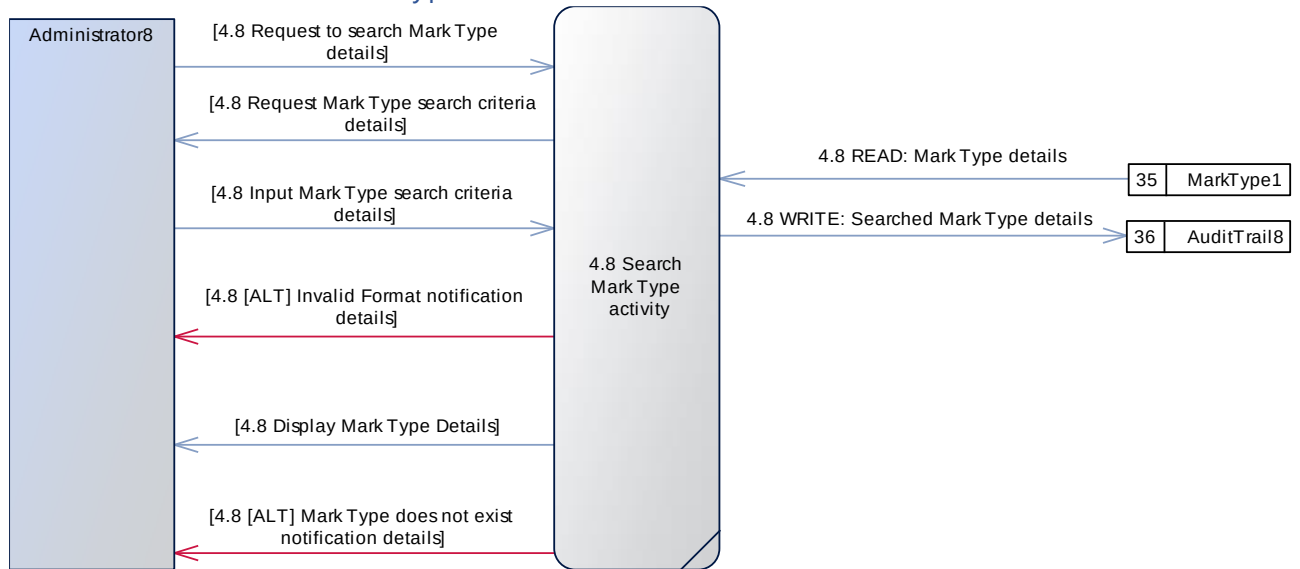


Figure 73 Search Mark Type mid-level diagram



3.4.3.4.9 Update Mark Type

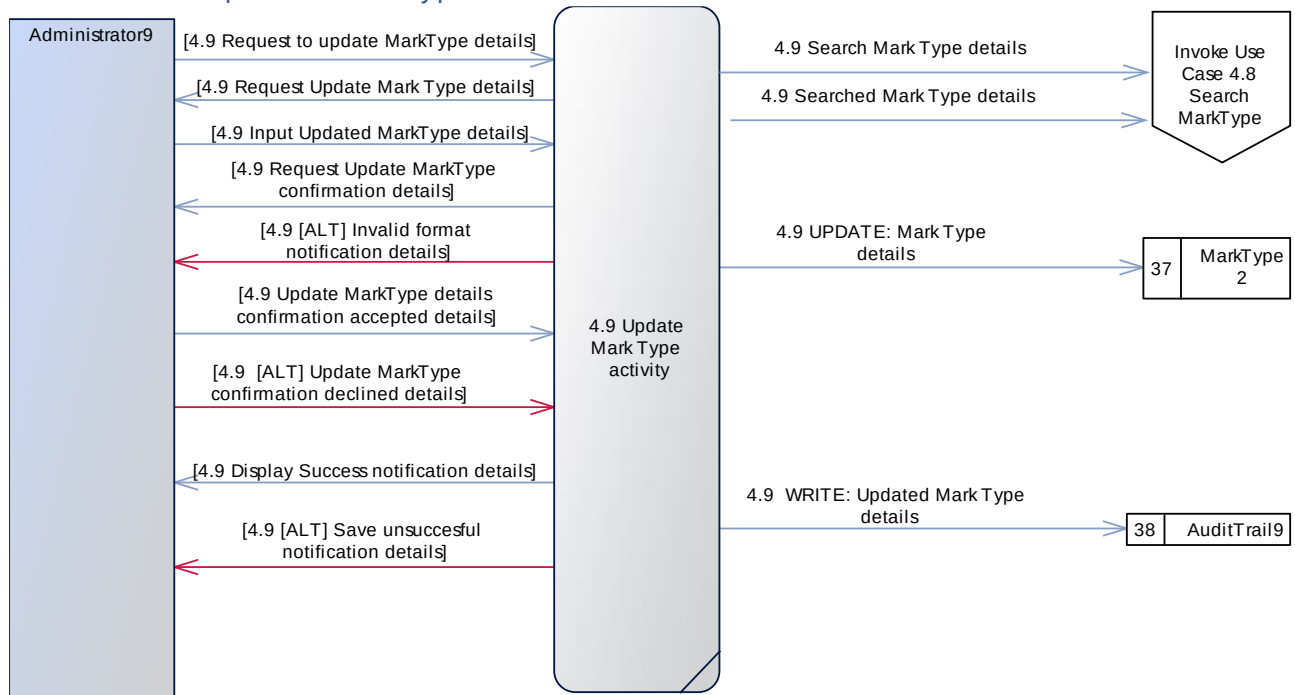


Figure 74 Update Mark Type mid-level diagram



3.4.3.4.10 Add Mark Type

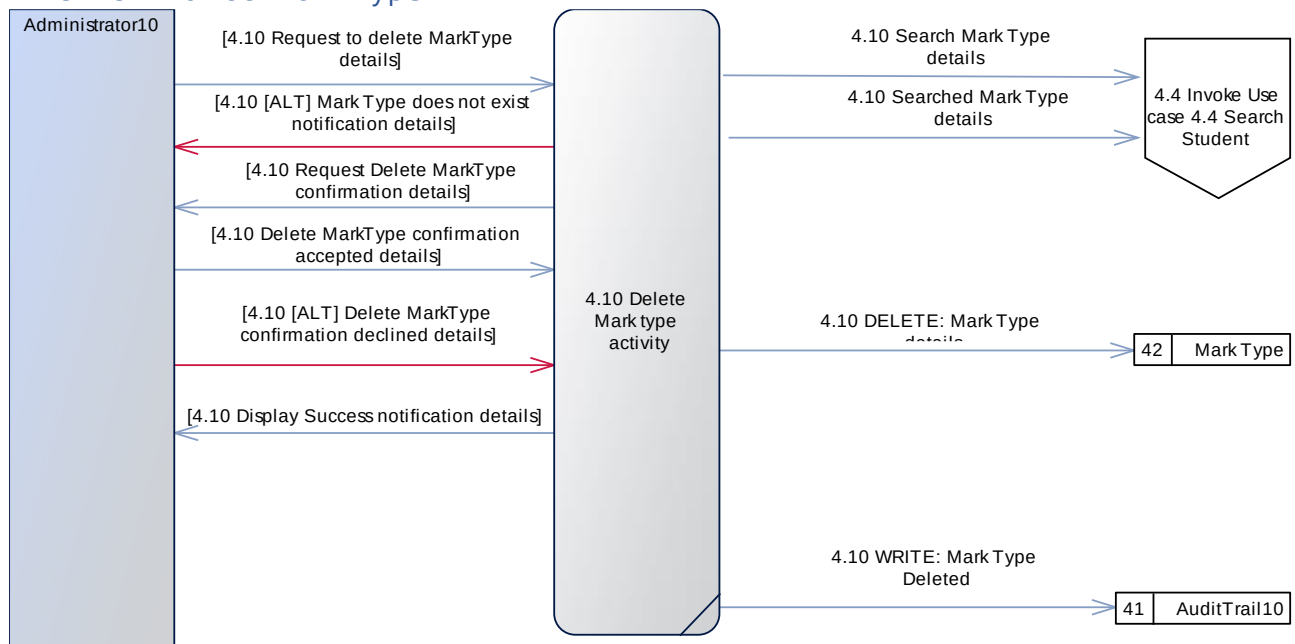


Figure 75 Delete Mark Type mid-level diagram



3.4.3.5 Sub-system 5: Bicycle

3.4.3.5.1 Add Bicycle

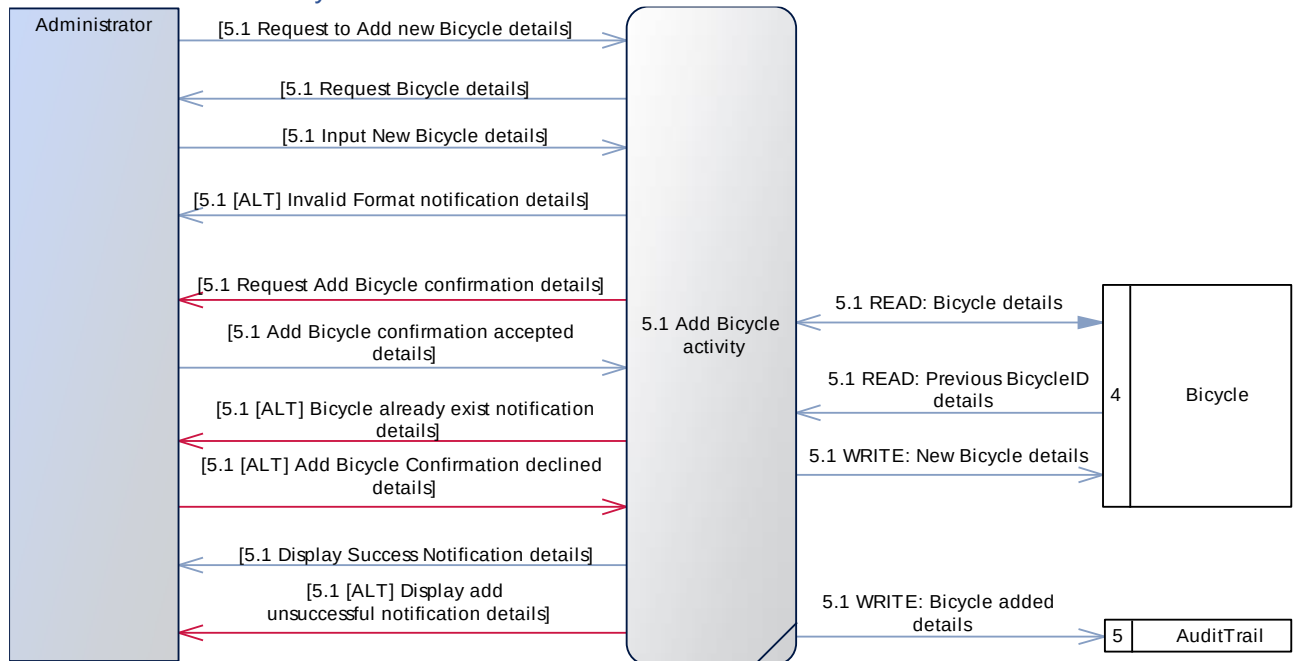


Figure 76 Add Bicycle mid-level diagram



3.4.3.5.2 Search Bicycle

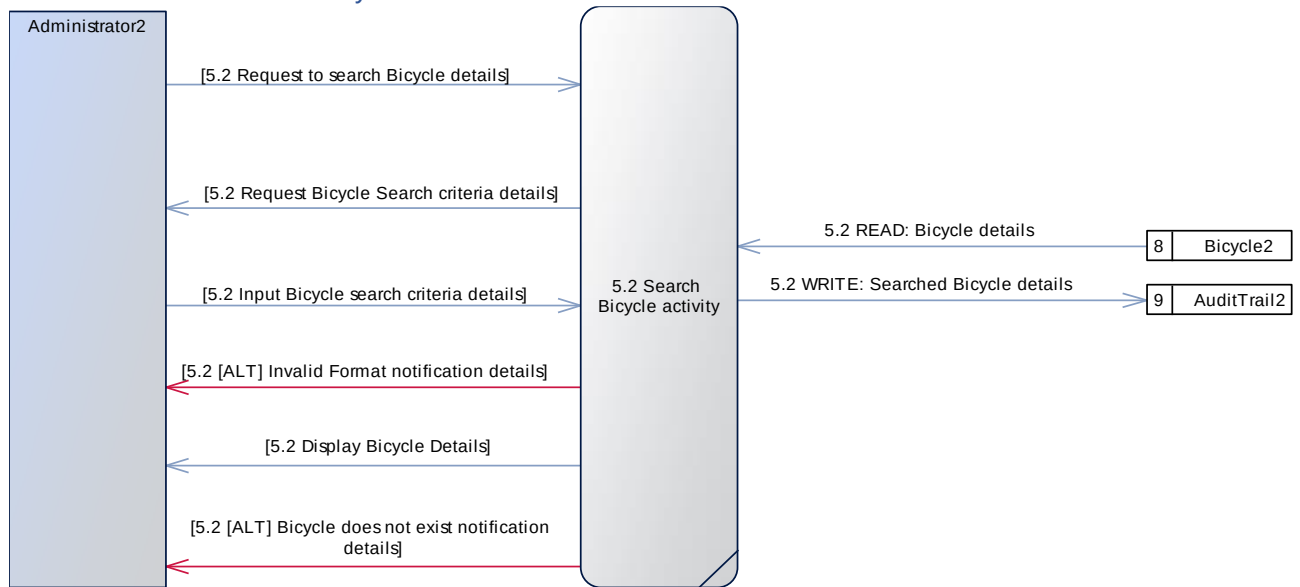


Figure 77 Search Bicycle mid-level diagram



3.4.3.5.3 Update Bicycle

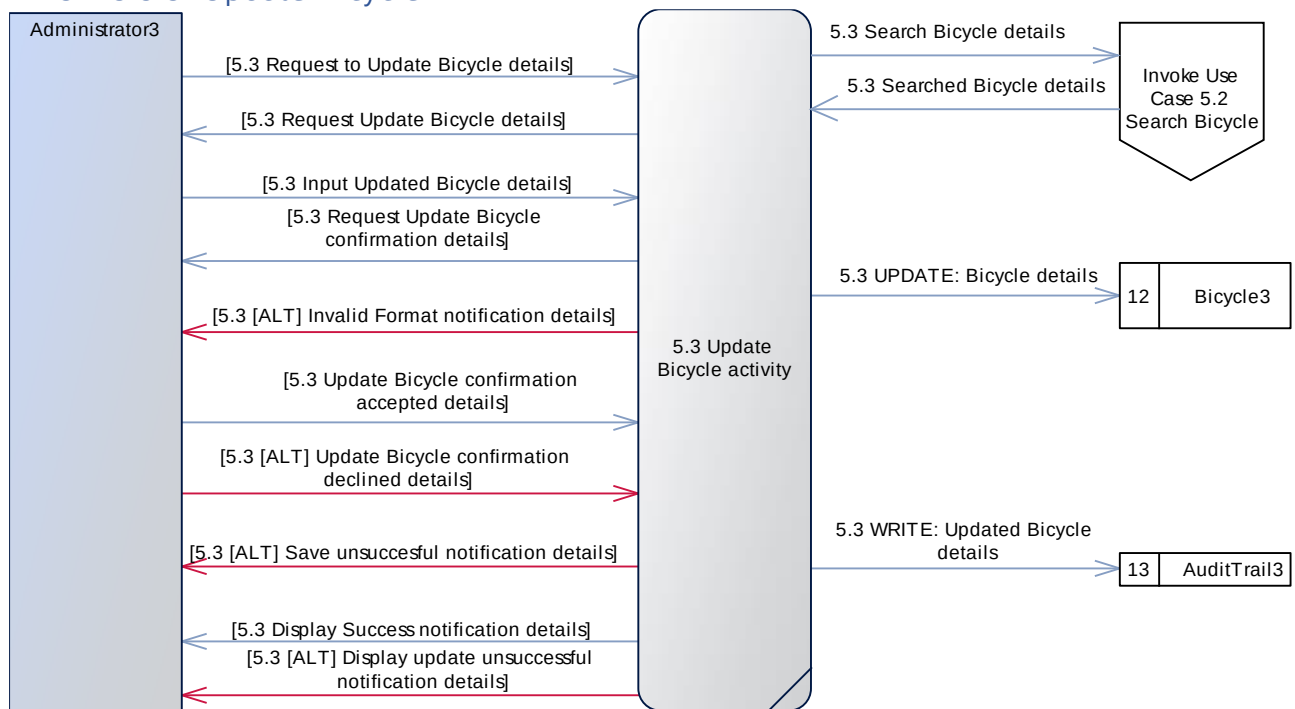


Figure 78 Update Bicycle mid-level diagram



3.4.3.5.4 Delete Bicycle

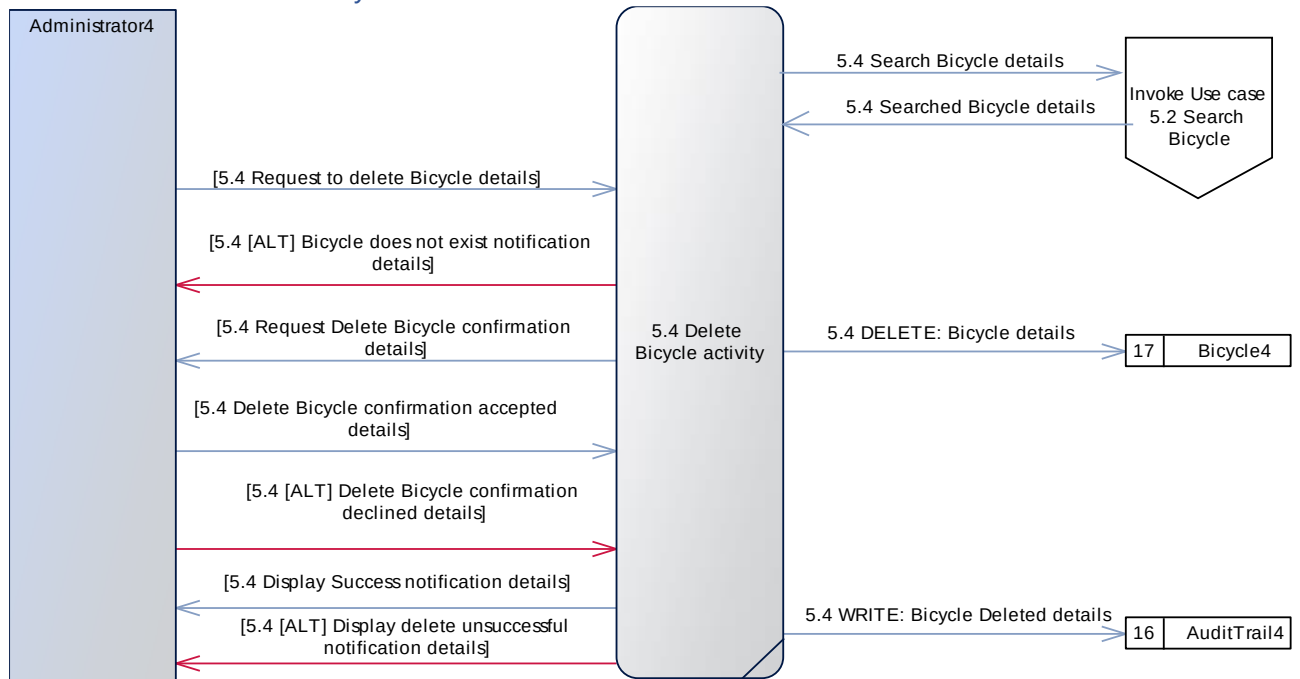


Figure 79 Delete Bicycle mid-level diagram



3.4.3.5.5 Assign Bicycle

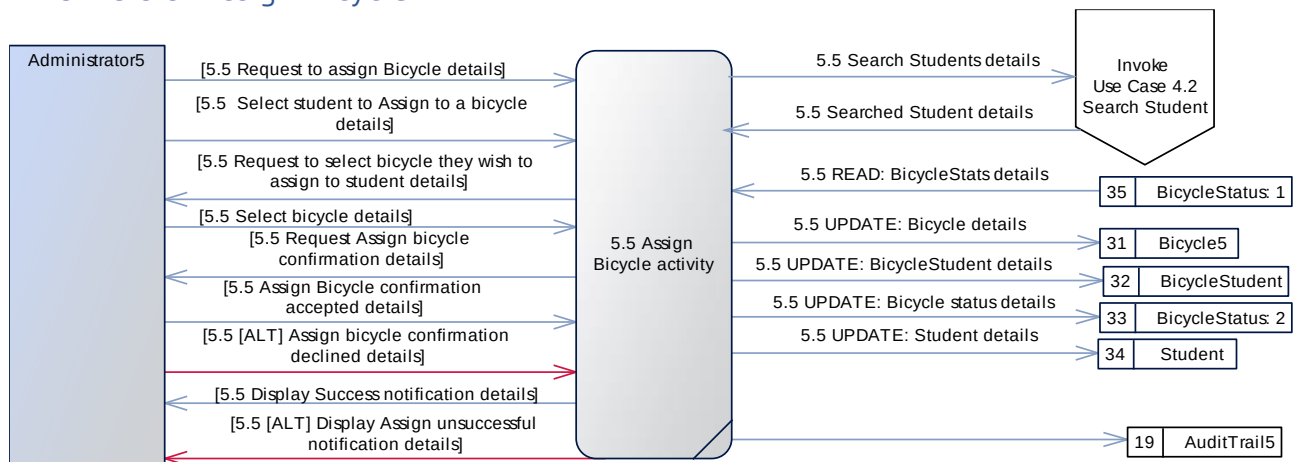


Figure 80 Assign Bicycle mid-level diagram



3.4.3.5.6 Unassign Bicycle

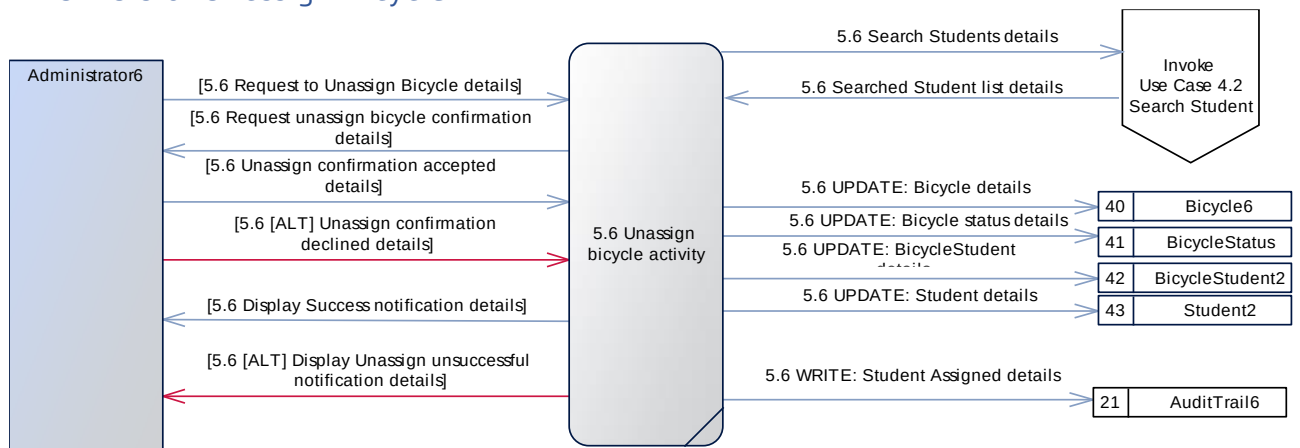


Figure 81 Unassign Bicycle mid-level diagram



3.4.3.5.7 Assign Tag to Bicycle

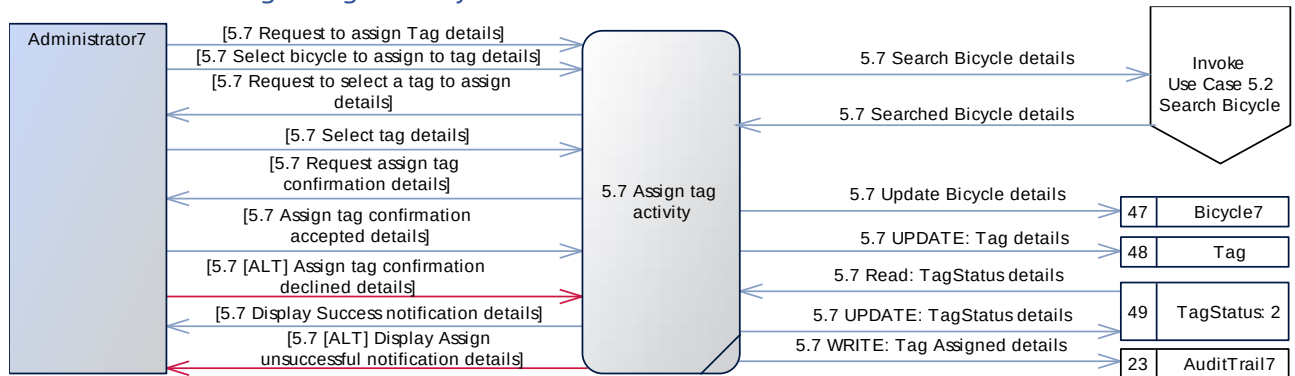


Figure 82 Assign Tag mid-level diagram



3.4.3.5.8 Unassign Tag

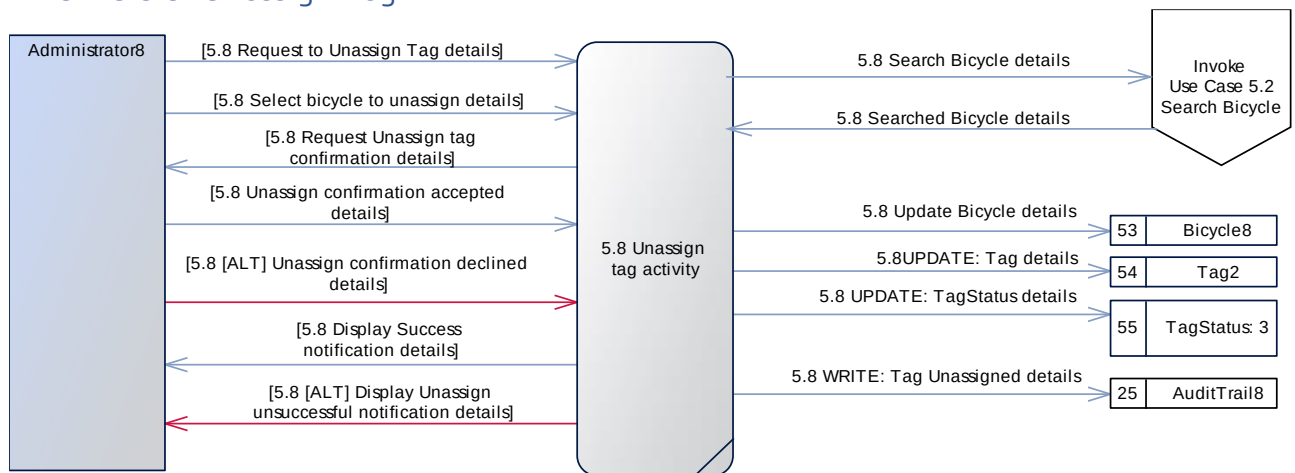


Figure 83 Unassign tag mid-level diagram



3.4.3.6 Sub-system 6: Bicycle Part

3.4.3.6.1 Add Bicycle Part

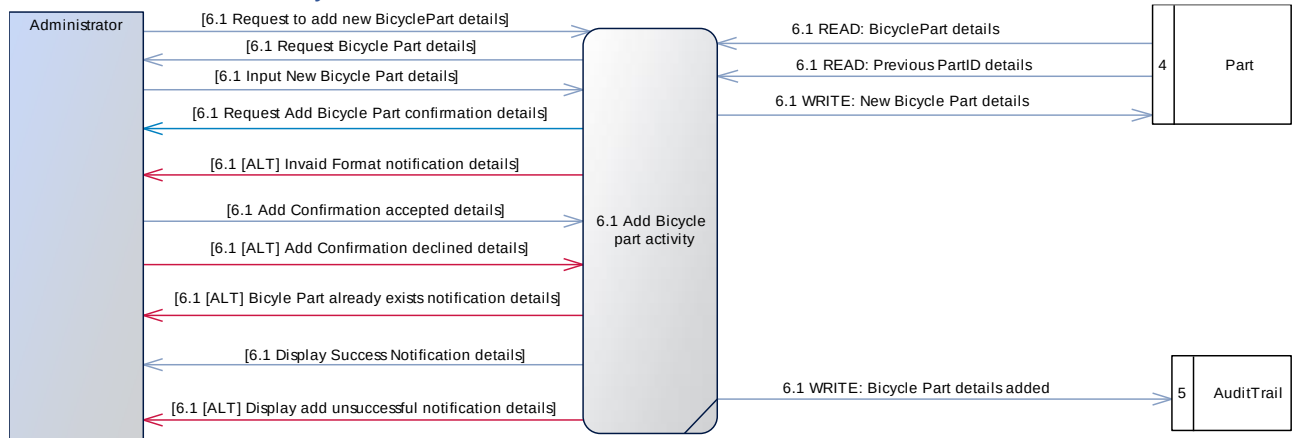


Figure 84 Add Bicycle Part mid-level diagram



3.4.3.6.2 Search Bicycle Part

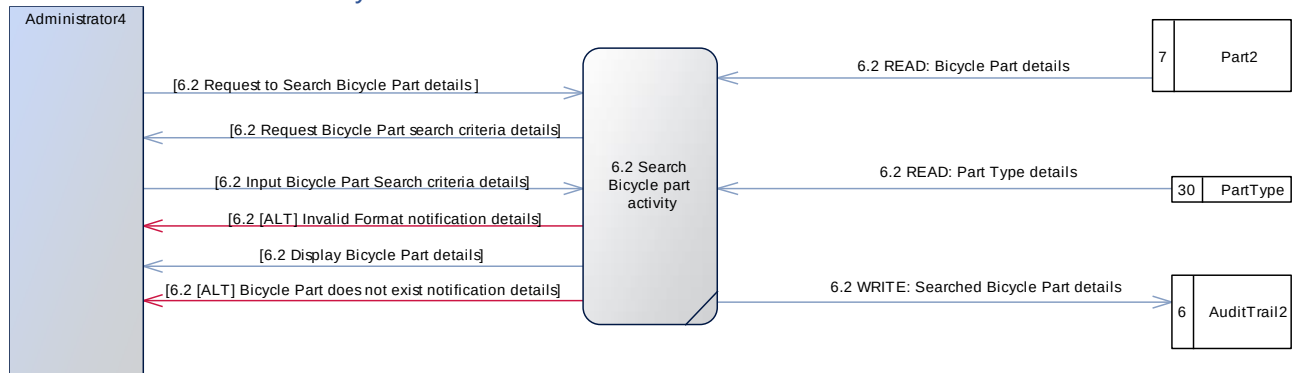


Figure 85 Search Bicycle Part diagram



3.4.3.6.3 Update Bicycle Part

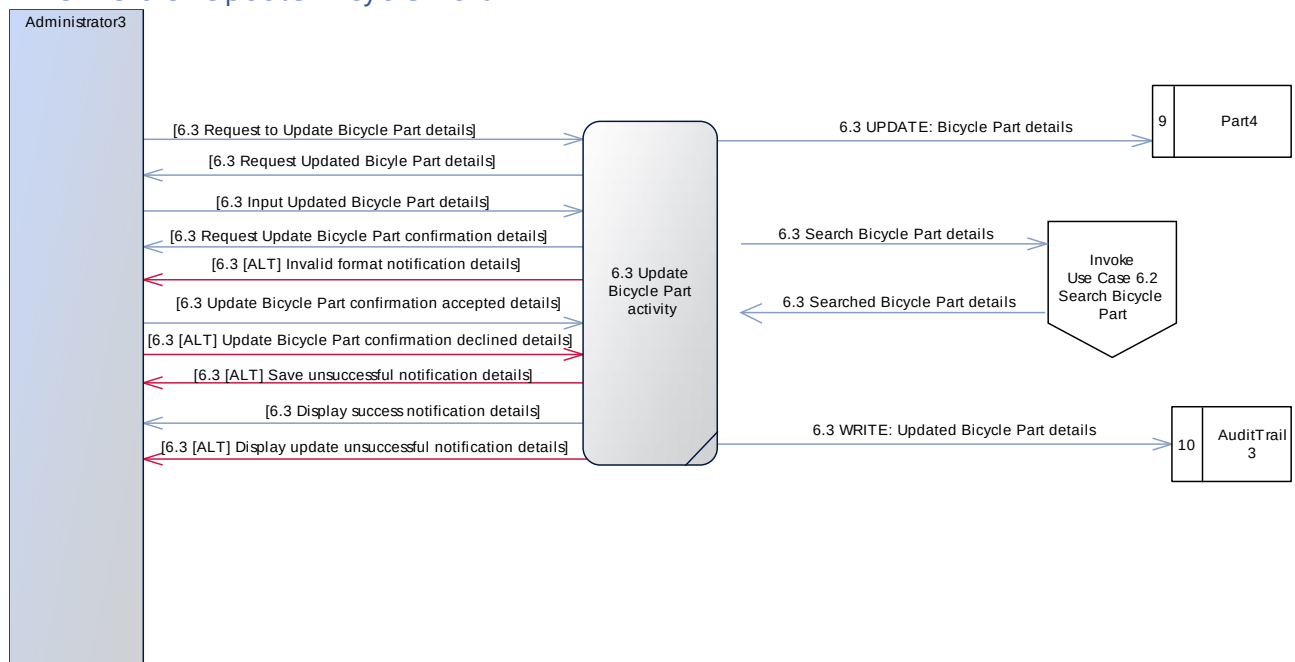


Figure 86 Update Bicycle Part mid-level diagram



3.4.3.6.4 Delete Bicycle Part

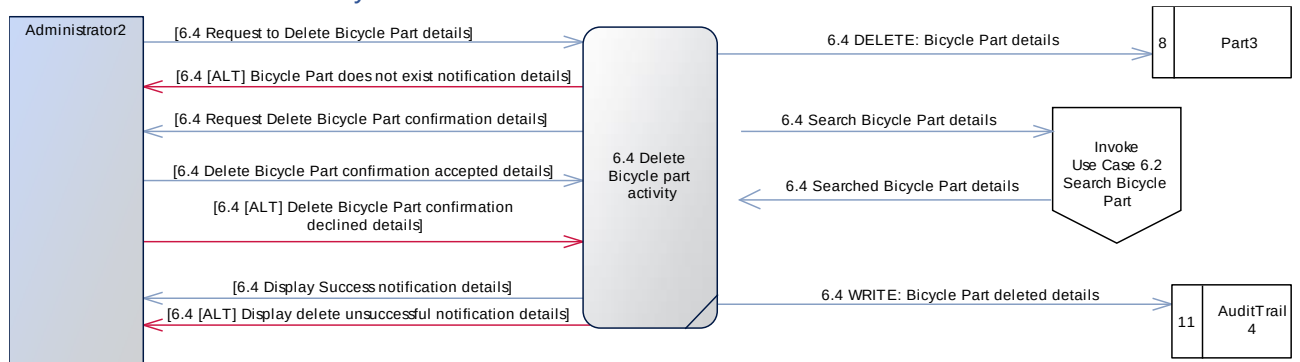


Figure 87 Delete Bicycle Part mid-level diagram



3.4.3.6.5 Add Bicycle Section

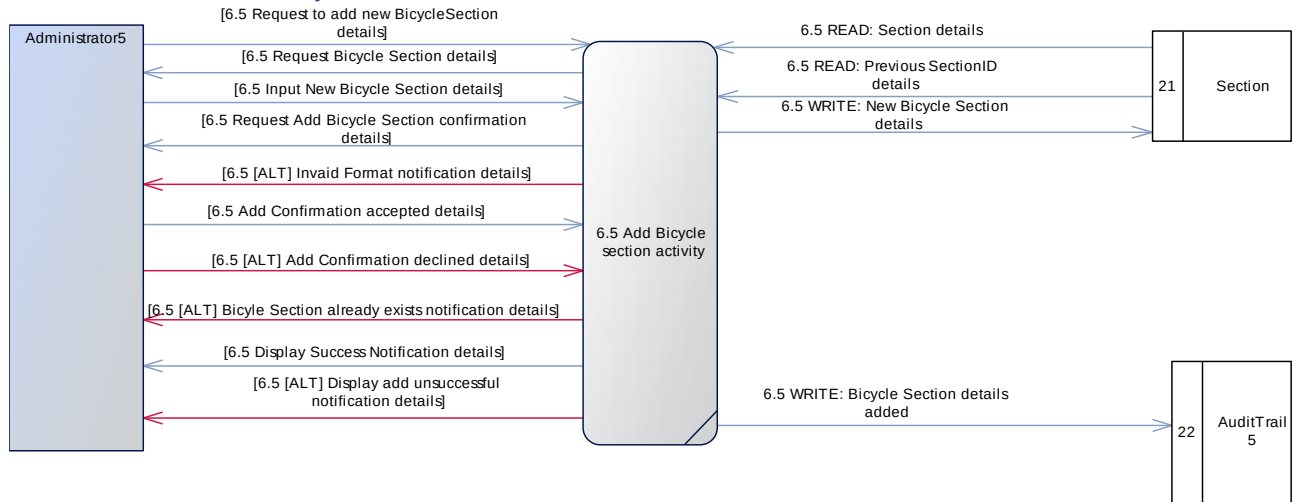


Figure 88 Add Bicycle Section mid-level diagram



3.4.3.6.6 Search Bicycle Section

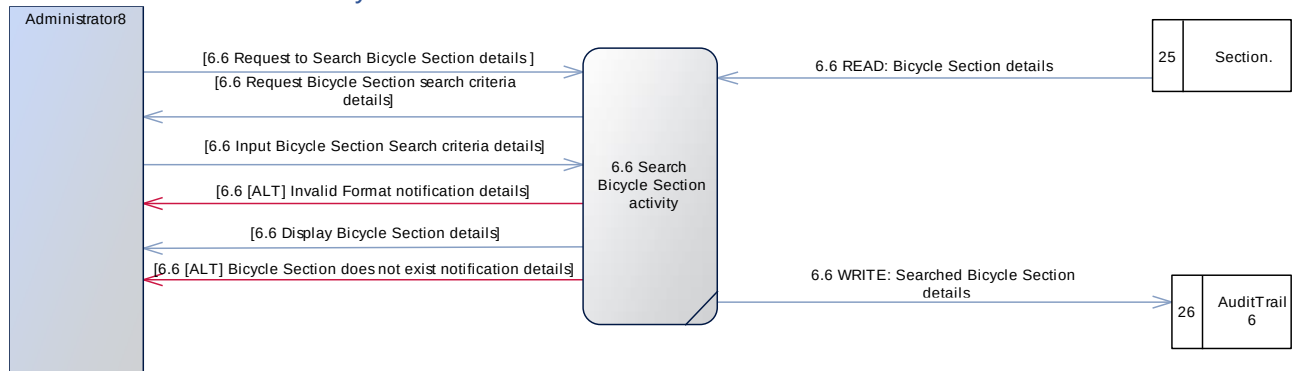


Figure 89 Search Bicycle Section mid-level diagram



3.4.3.6.7 Update Bicycle Section

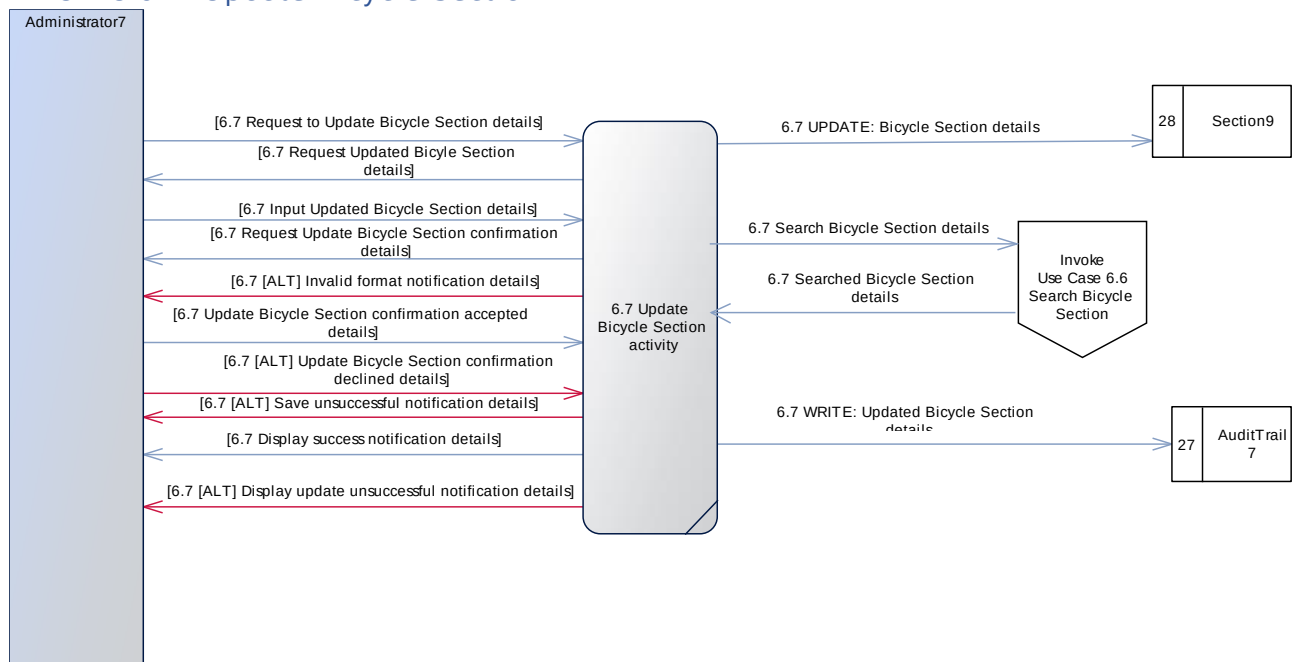


Figure 90 Update Bicycle Section mid-level diagram



3.4.3.6.8 Delete Bicycle Section

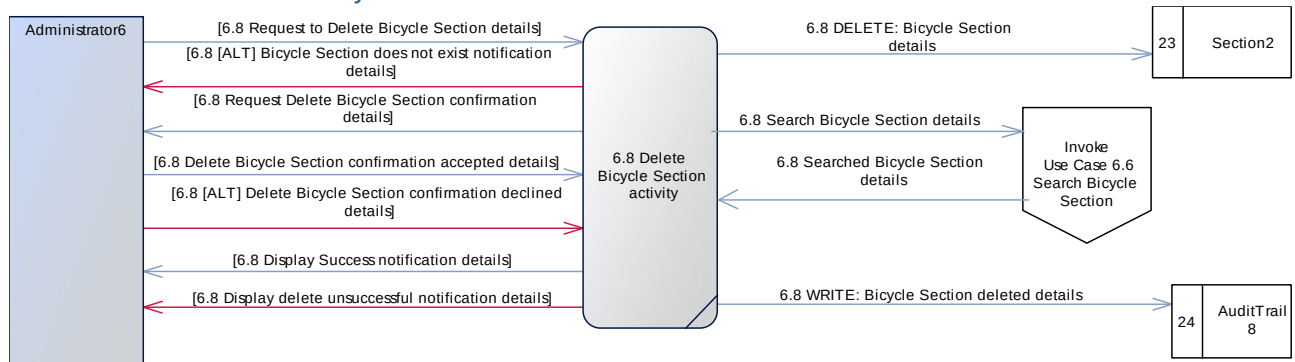


Figure 91 Delete Bicycle Section mid-level diagram

3.4.3.7 Sub-system 7: Tracking

3.4.3.7.1 Add Log Entry

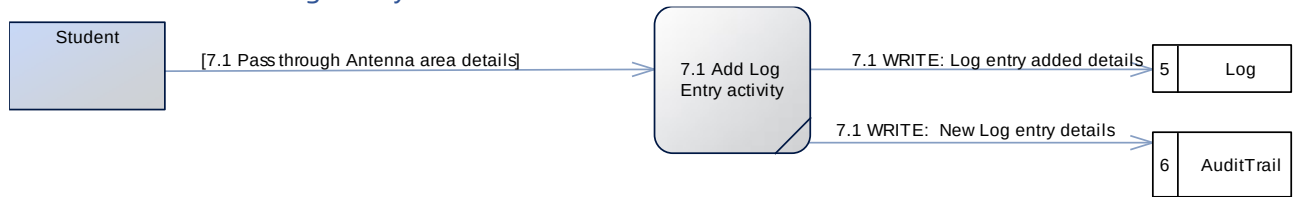


Figure 92 Add Log Entry mid-level diagram



3.4.3.7.2 Search Log Entry

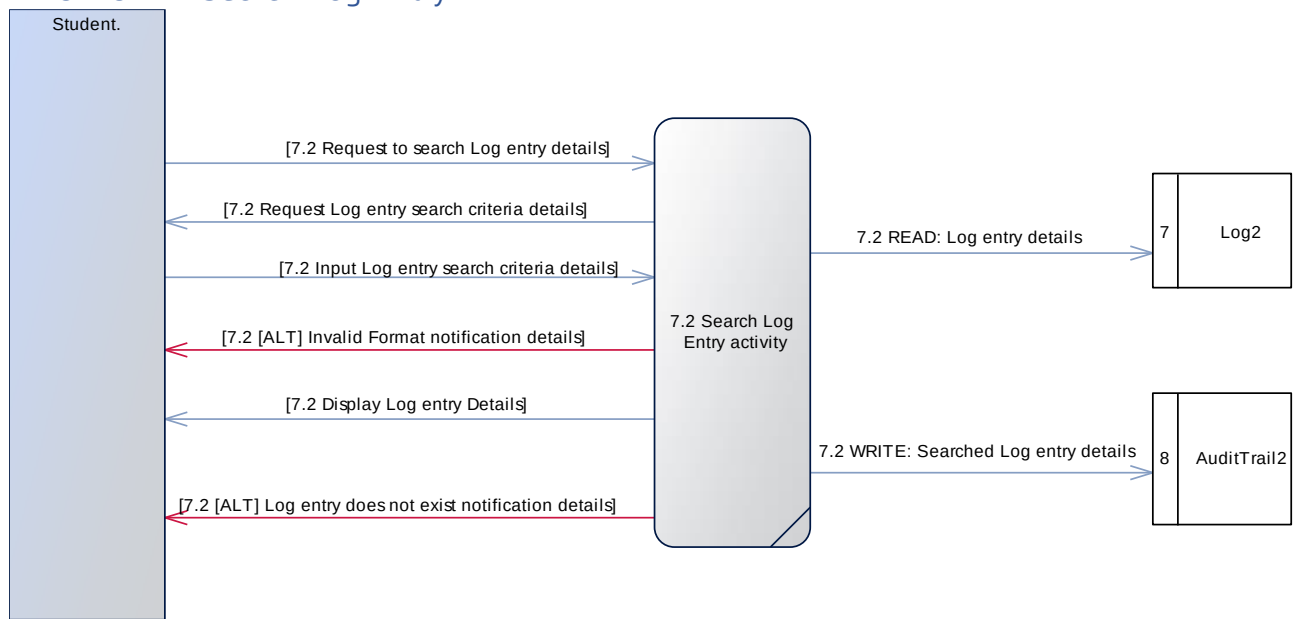


Figure 93 Search Log Entry mid-level diagram



3.4.3.8 Sub-system 8: Job Tasks

3.4.3.8.1 Add Fault Task

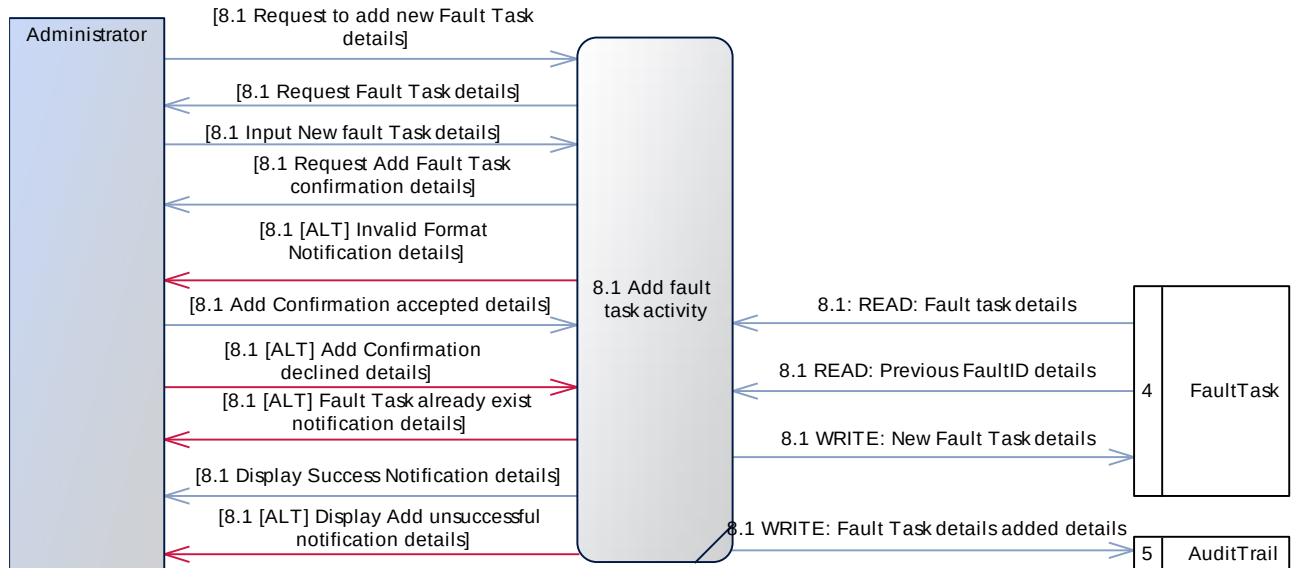


Figure 94 Add Fault Task mid-level diagram



3.4.3.8.2 Search Fault Task

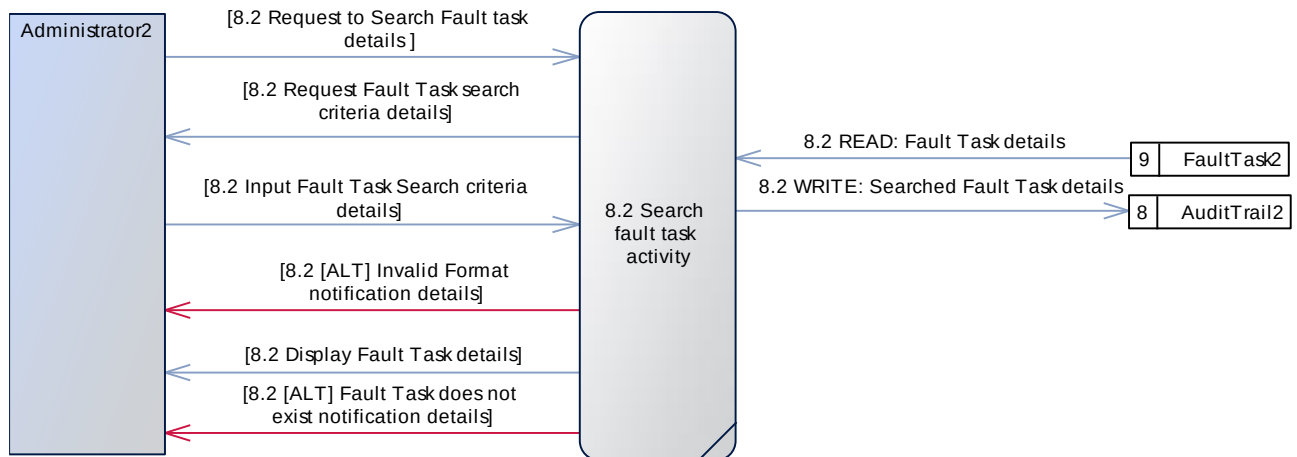


Figure 95 Search Fault Task mid-level diagram



3.4.3.8.3 Update Fault Task

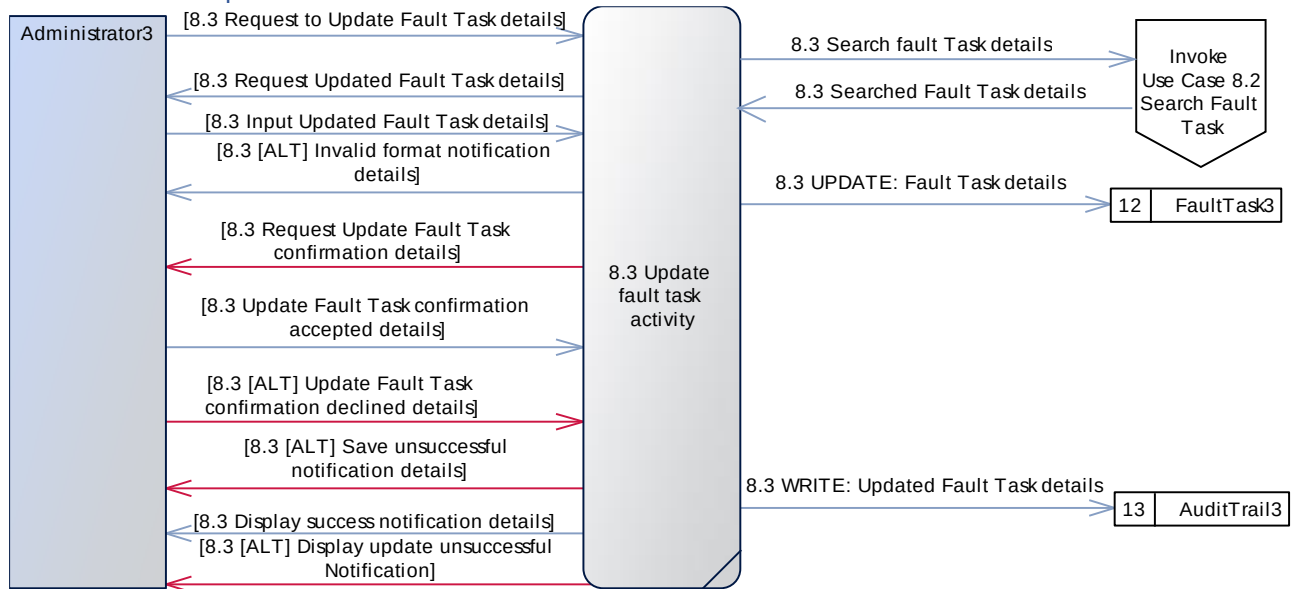


Figure 96 Update Fault Task mid-level diagram



3.4.3.8.4 Delete Fault Task

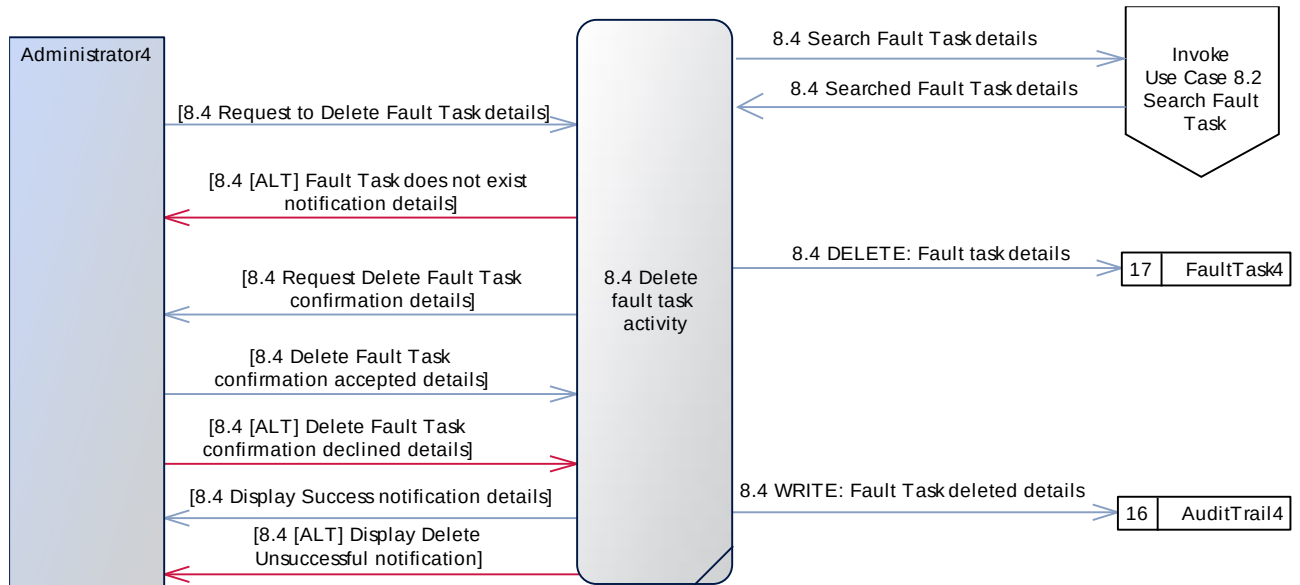


Figure 97 Delete Fault Task mid-level diagram



3.4.3.8.5 Add Maintenance Task

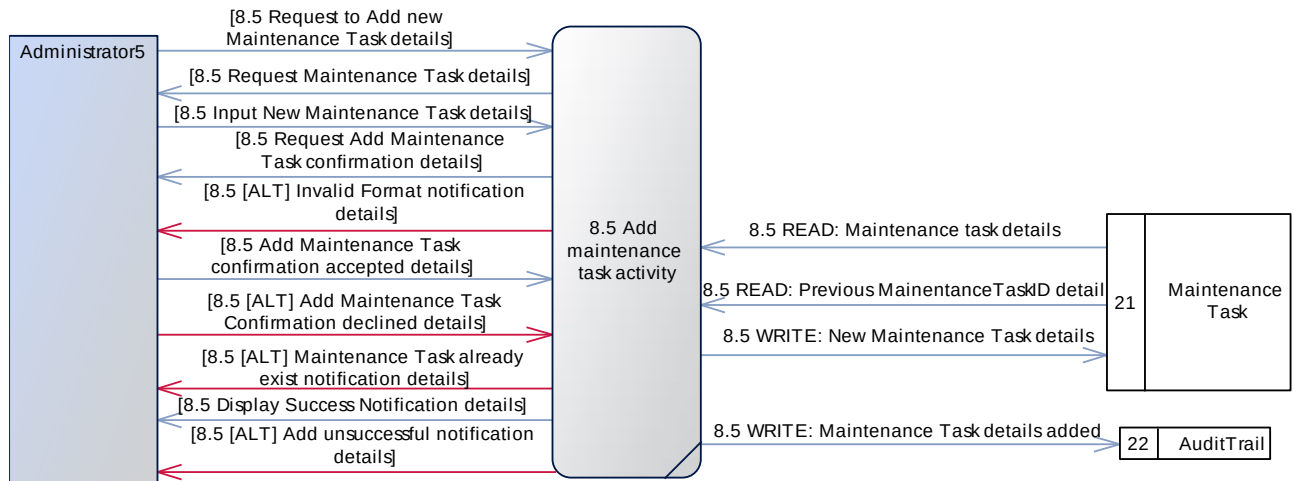


Figure 98 Add Maintenance Task mid-level diagram



3.4.3.8.6 Search Maintenance Task

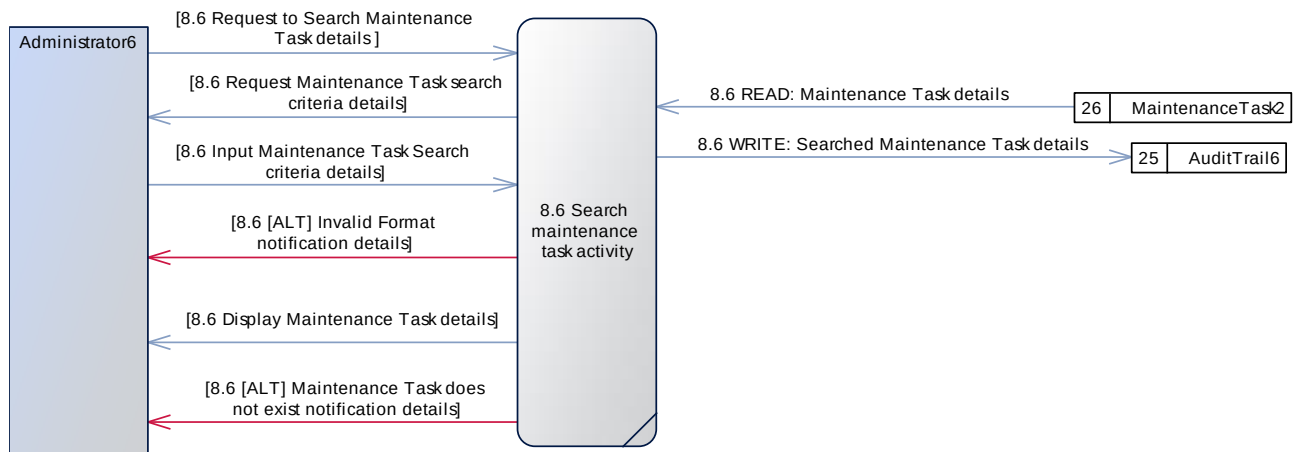


Figure 99 Search Maintenance Task mid-level diagram



3.4.3.8.7 Update Maintenance Task

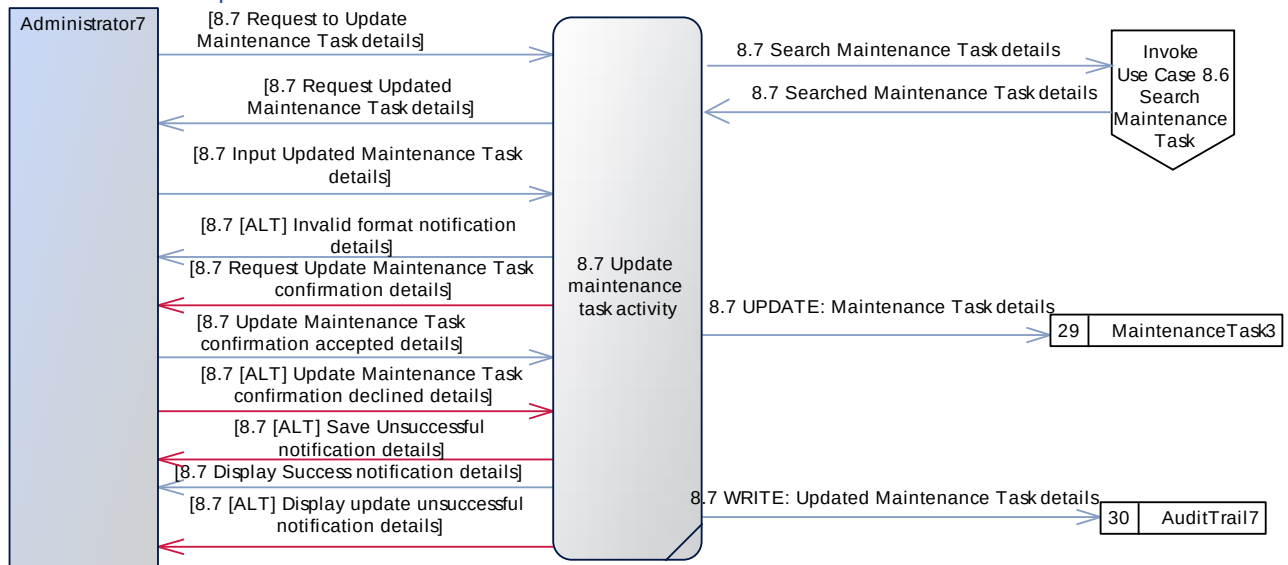


Figure 100 Update Maintenance Task mid-level diagram



3.4.3.8.8 Delete Maintenance Task

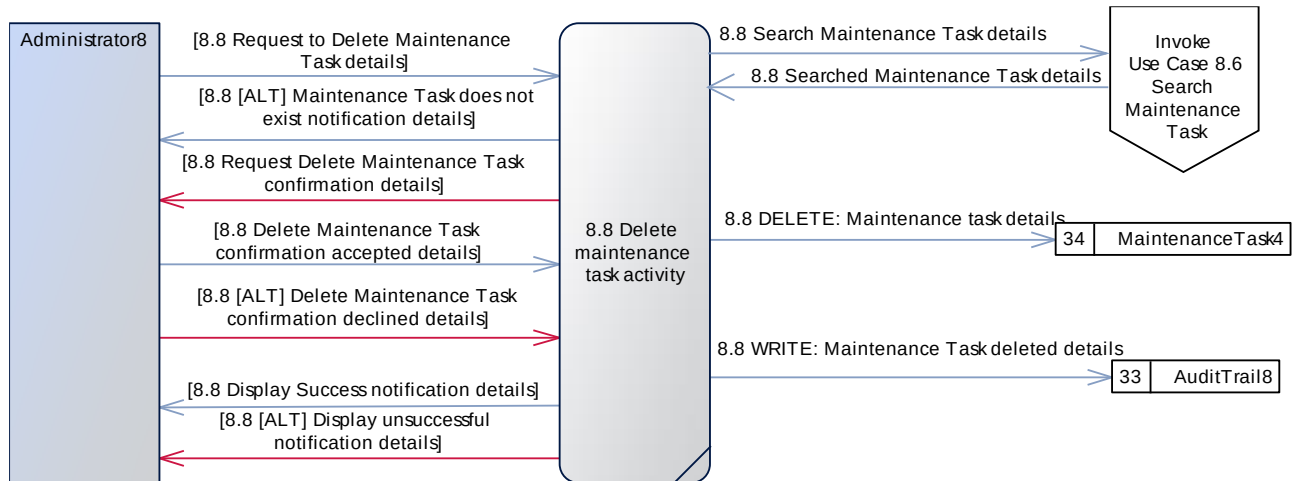


Figure 101 Delete Maintenance Task mid-level diagram



3.4.3.9 Sub-system 9: School

3.4.3.9.1 Add School

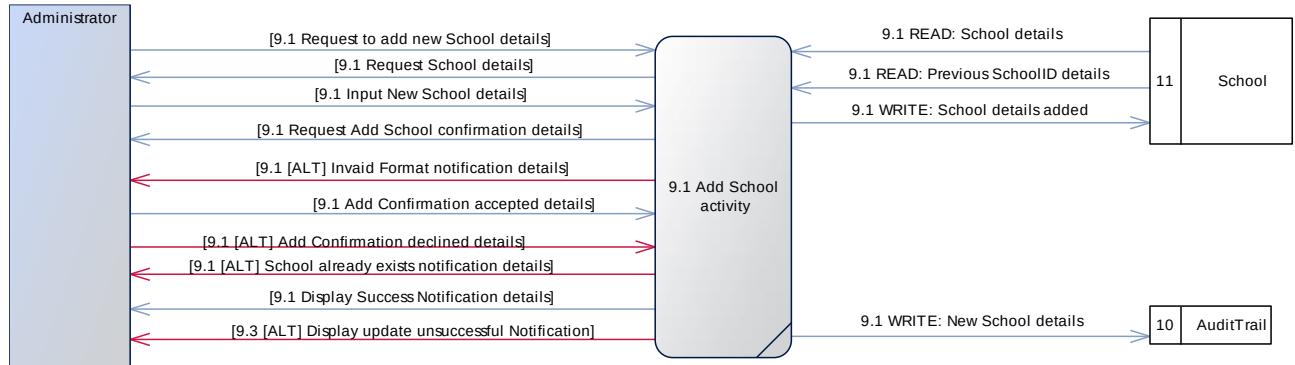


Figure 102 Add School mid-level diagram



3.4.3.9.2 Search School

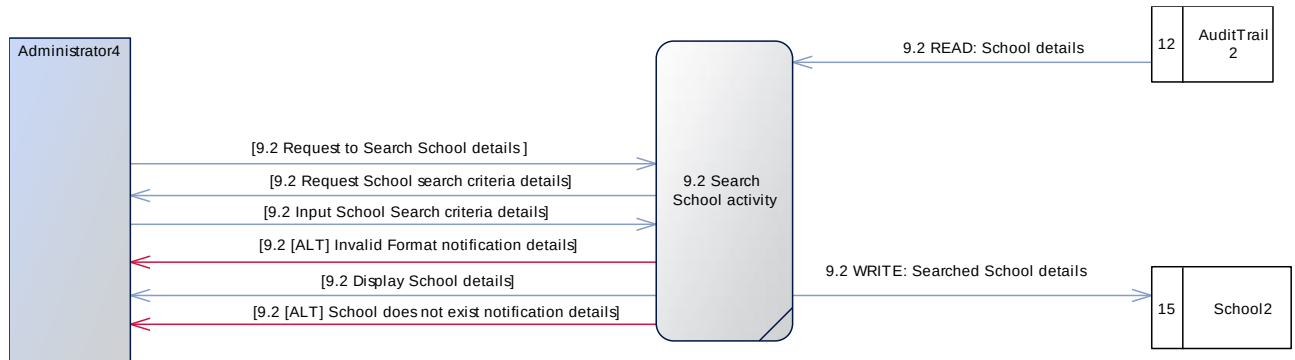


Figure 103 Search School mid-level diagram



3.4.3.9.3 Update School

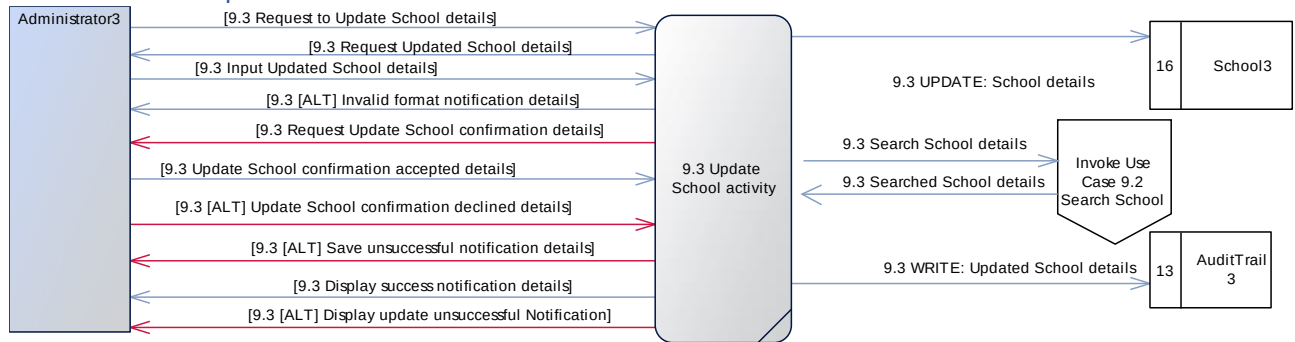


Figure 104 Update School mid-level diagram



3.4.3.9.4 Delete School

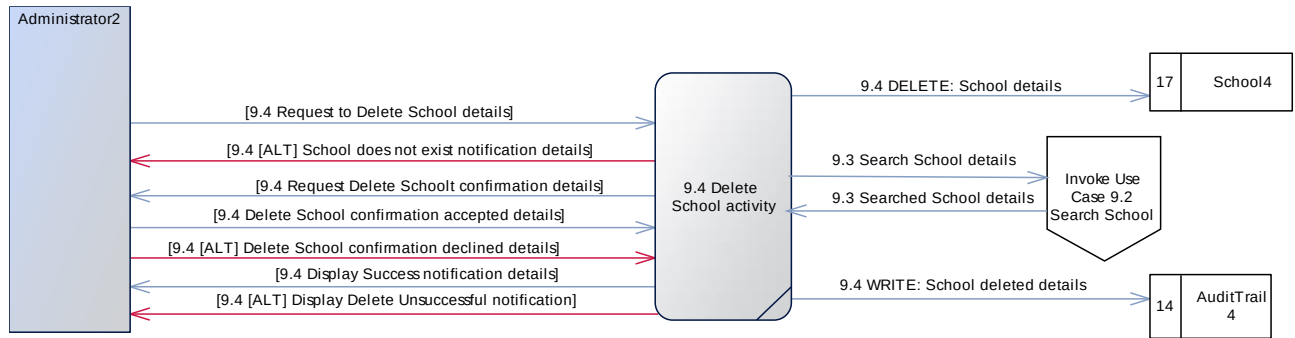


Figure 105 Delete School mid-level diagram



3.4.3.10 Sub-system 10: Job

3.4.3.10.1 Report Fault

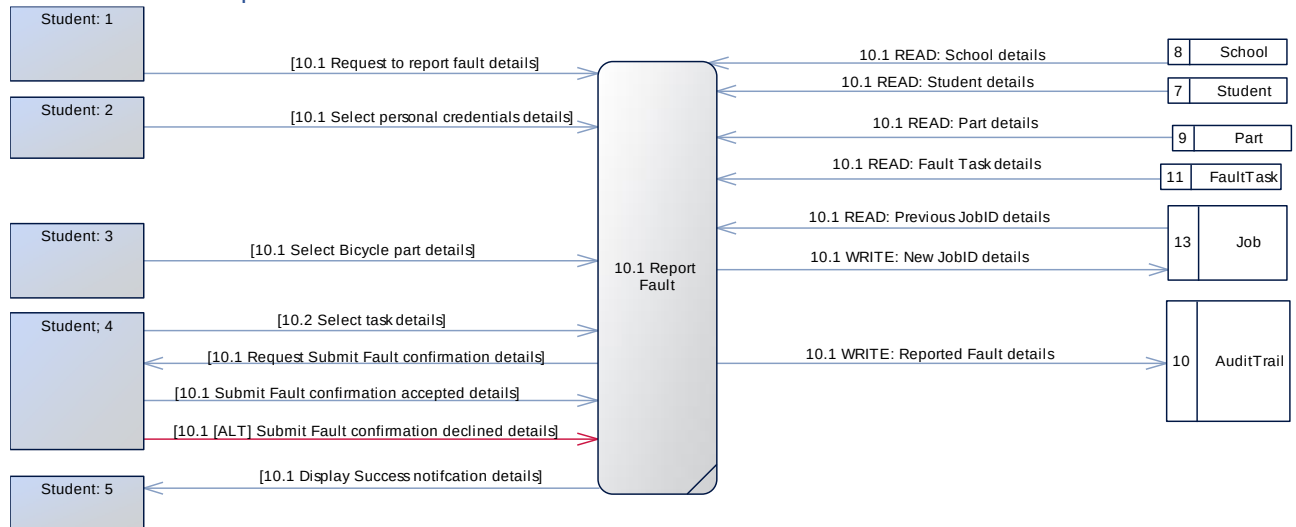


Figure 106 Report Fault mid-level diagram



3.4.3.10.2 View Jobs

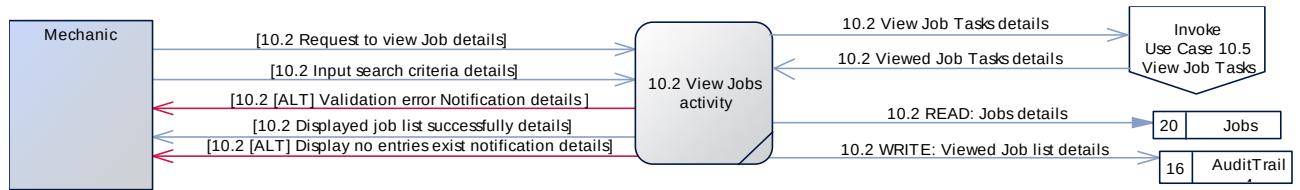


Figure 107 View Jobs mid-level diagram



3.4.3.10.3 Check-in Bicycle

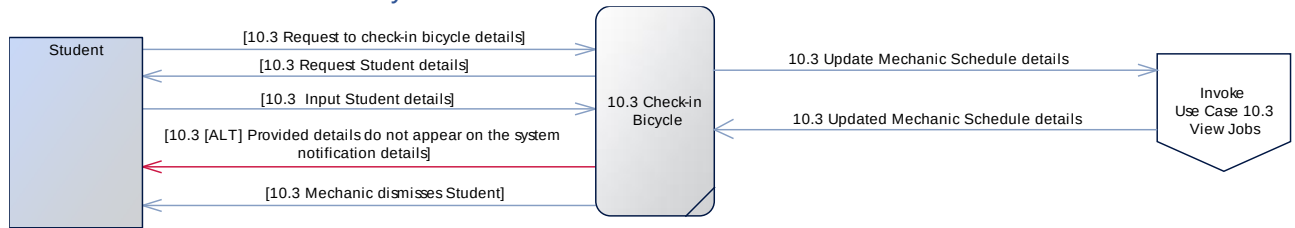


Figure 108 Check-in Bicycle mid-level diagram



3.4.3.10.4 Report Maintenance

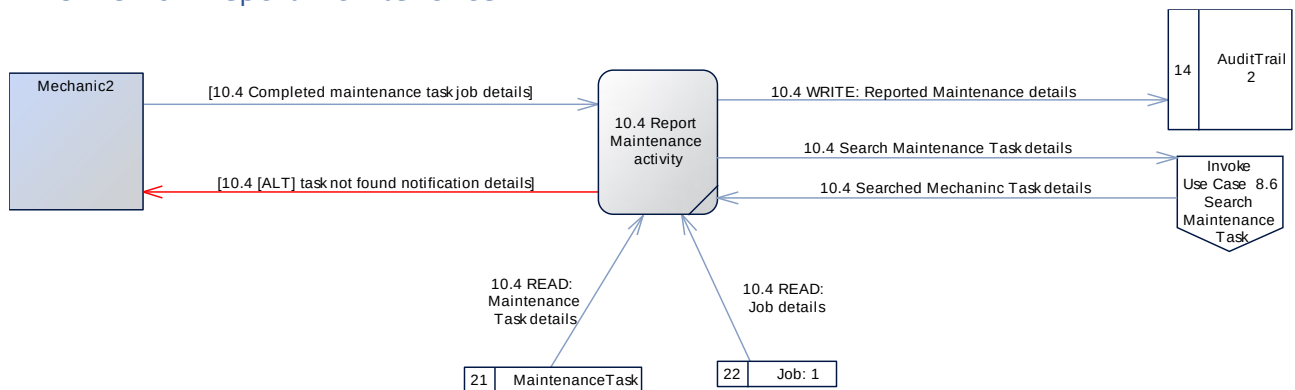


Figure 109 Report Maintenance mid-level diagram



3.4.3.11 Sub-system 11: Reporting

3.4.3.11.1 Generate Maintenance Schedule

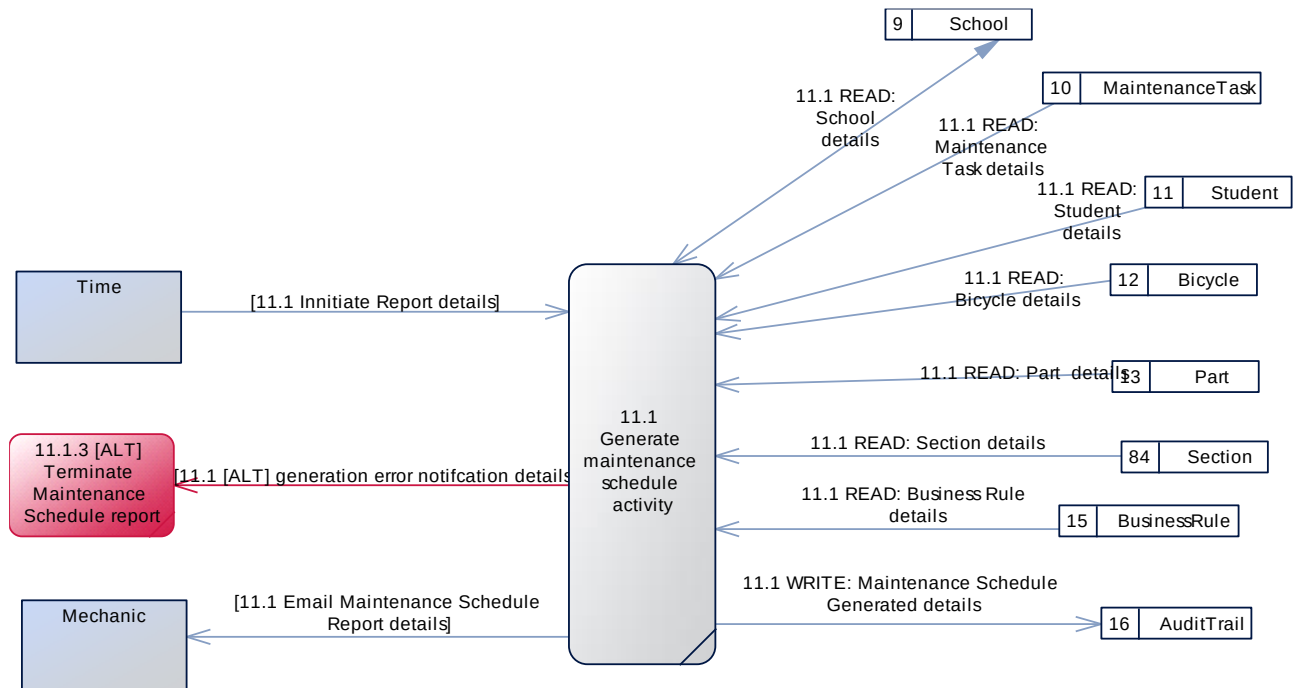


Figure 110 Generate Maintenance Schedule mid-level diagram



3.4.3.11.2 Generate Maintenance Requirements Report

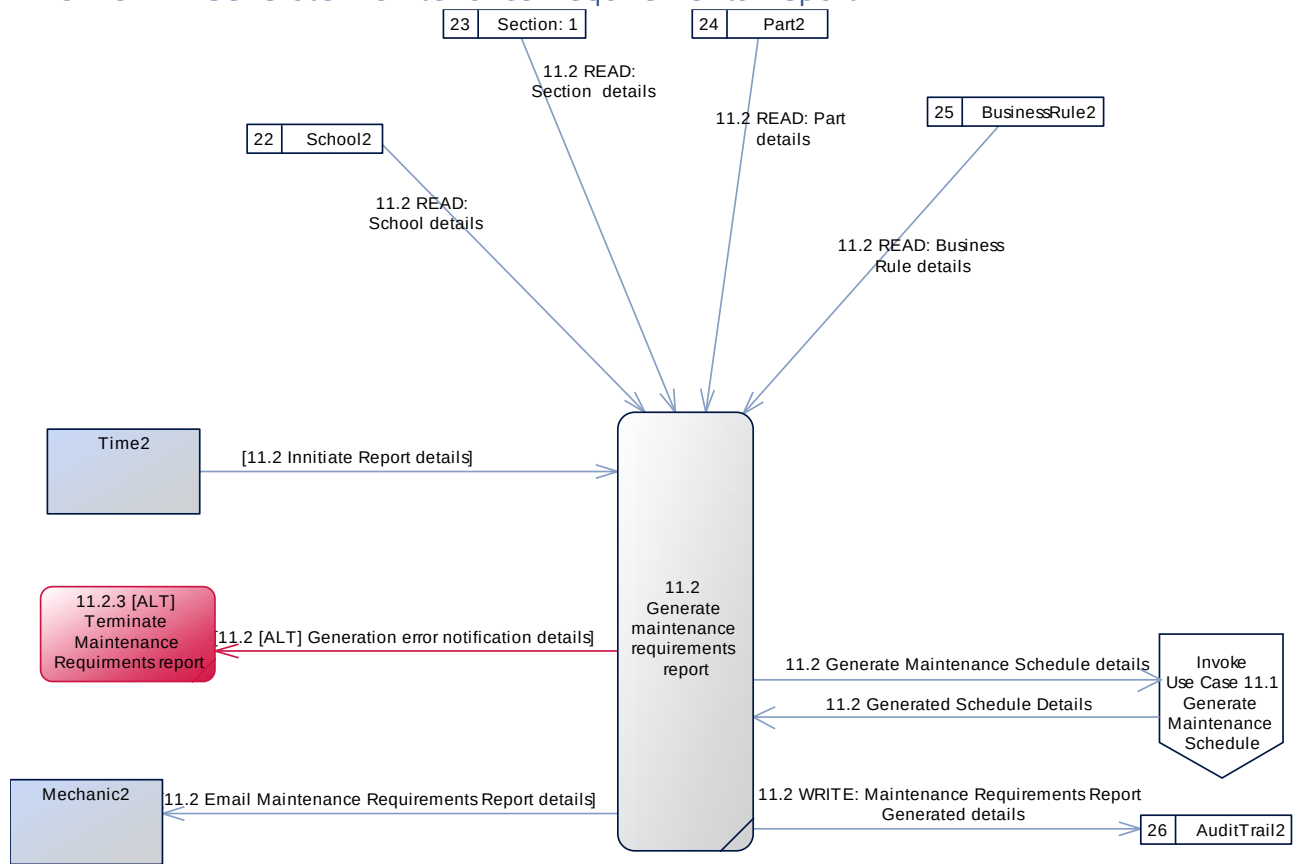


Figure 111 Generate Maintenance Requirements report mid-level diagram



3.4.3.11.3 Generate Student Report

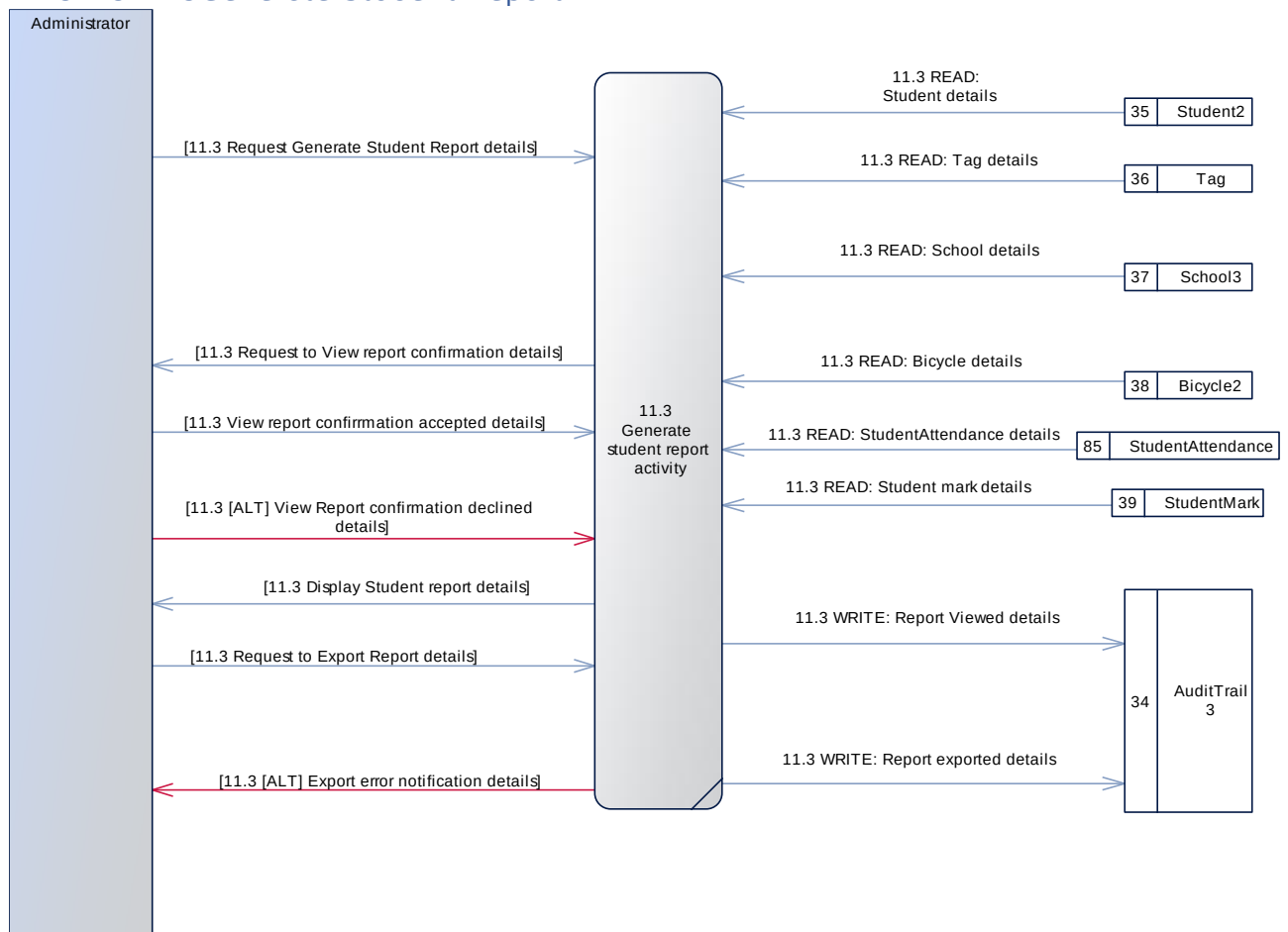


Figure 112 Generate Student Report mid-level diagram



3.4.3.11.4 Generate Bicycle History Report

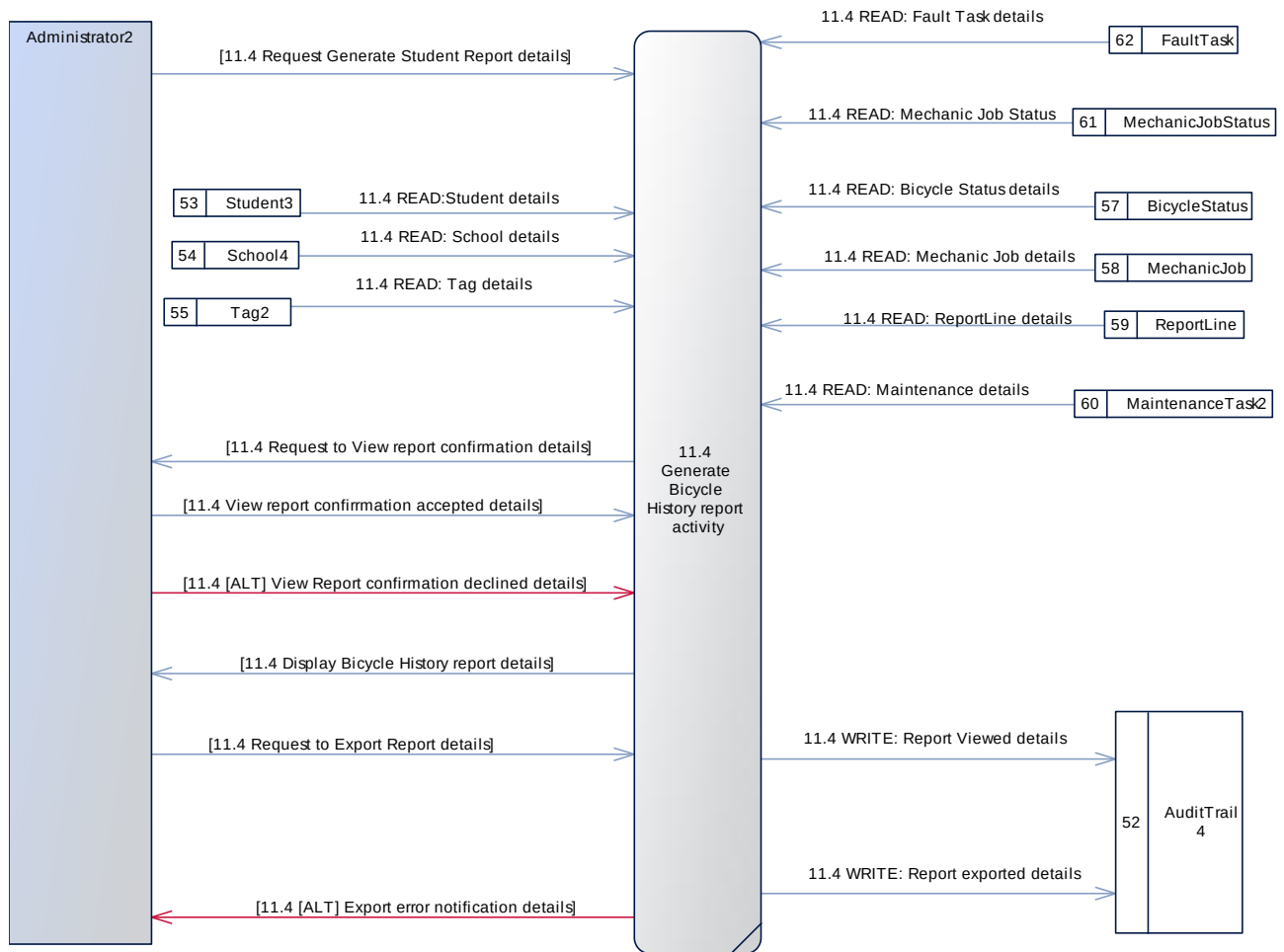


Figure 113 Generate Bicycle History Report mid-level diagram



3.4.3.11.5 Generate Log Report

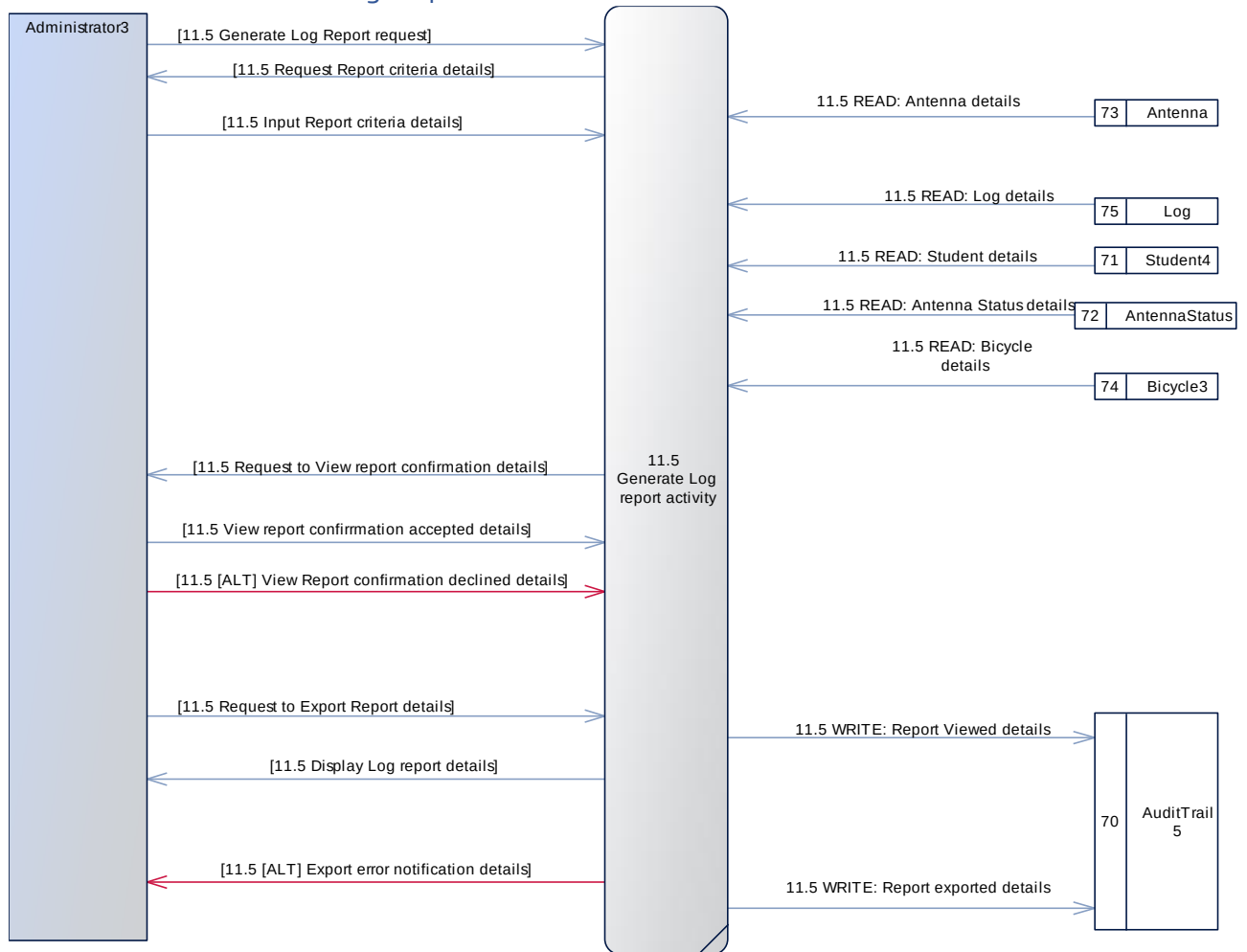


Figure 114 Generate Log Report mid-level diagram



3.4.3.11.6 Generate Parts Used

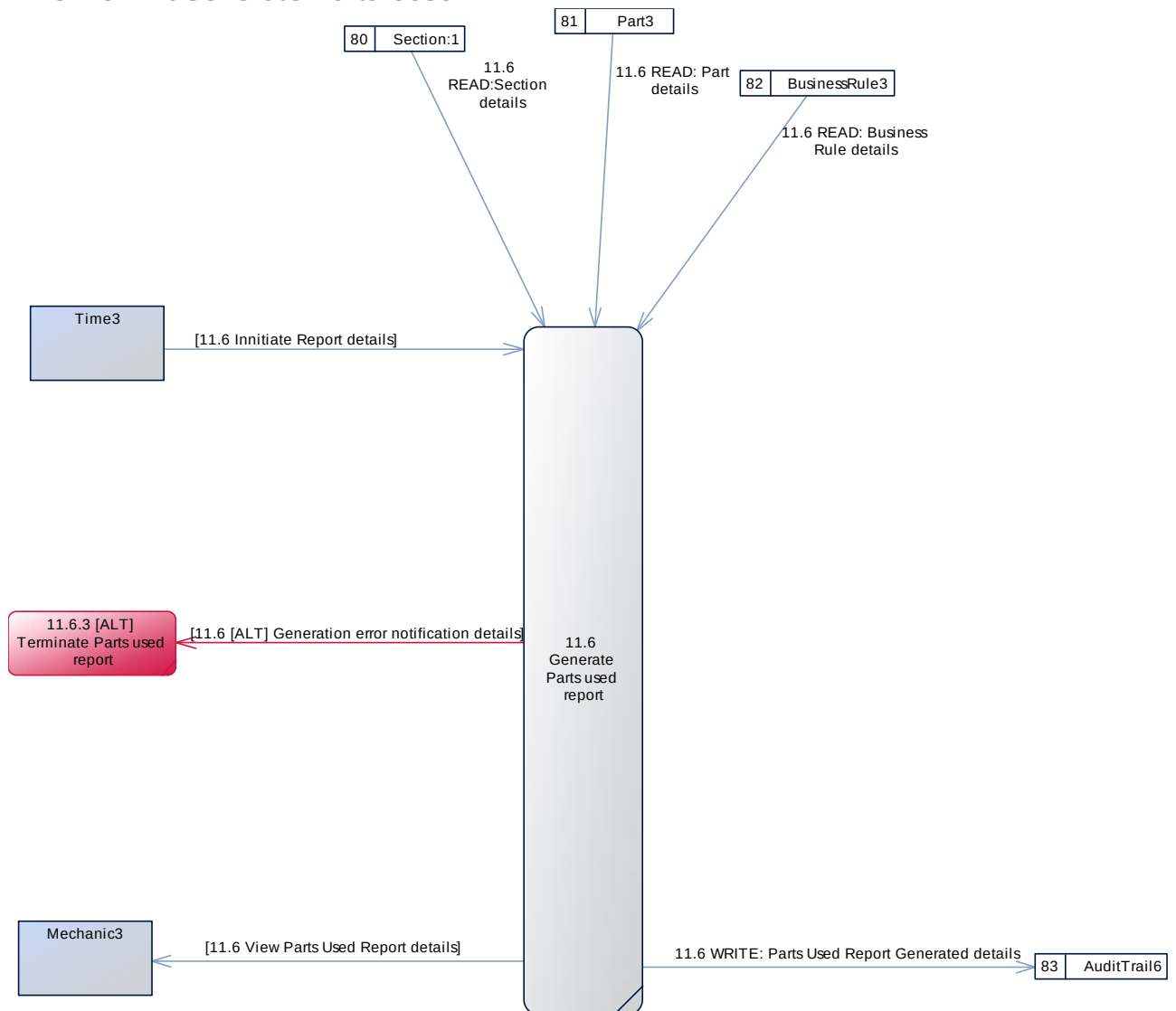


Figure 115 Generate Parts Used report mid-level diagram



3.4.3.12 Sub-system 12: Notification

3.4.3.12.1 Send Student Did Not Arrive Notification

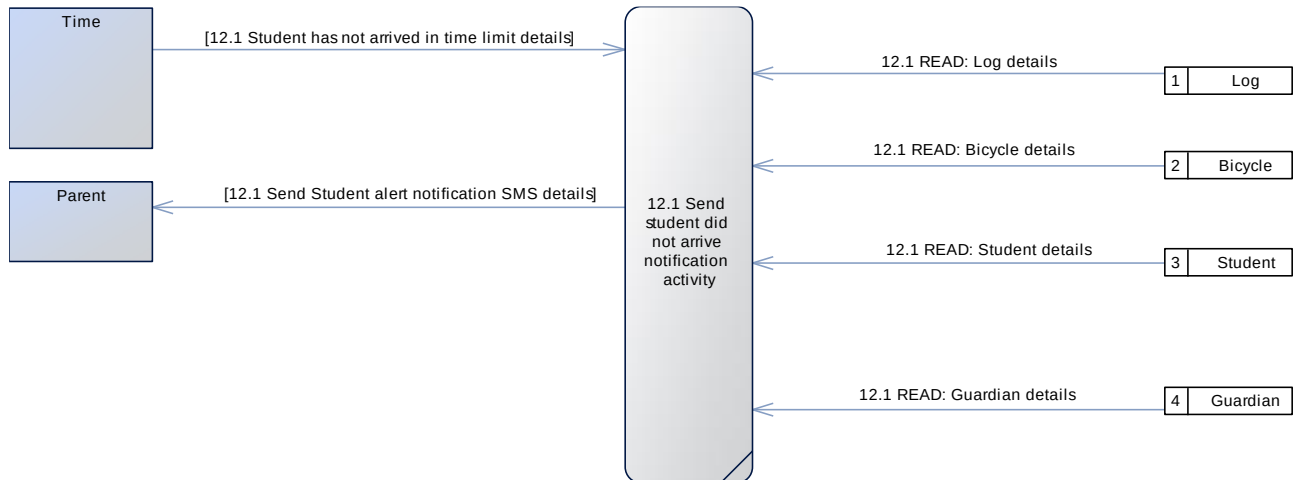


Figure 116 Send Student Did Not Arrive Notification mid-level diagram



3.4.3.12.2 Send Student Maintenance Notification

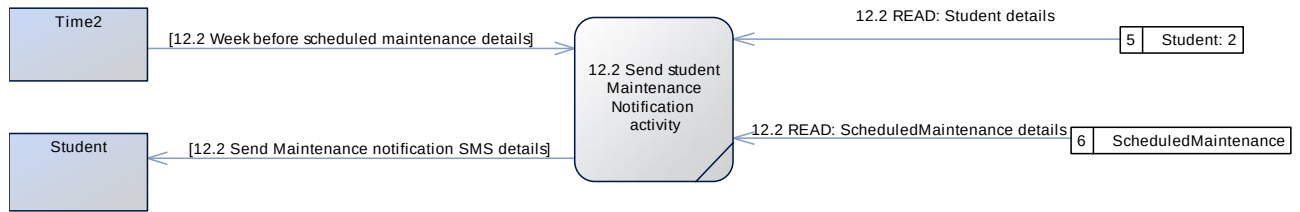


Figure 117 Send Student Maintenance Notification mid-level diagram



3.4.3.13 Sub-system 13: Administration

3.4.3.13.1 Backup Data

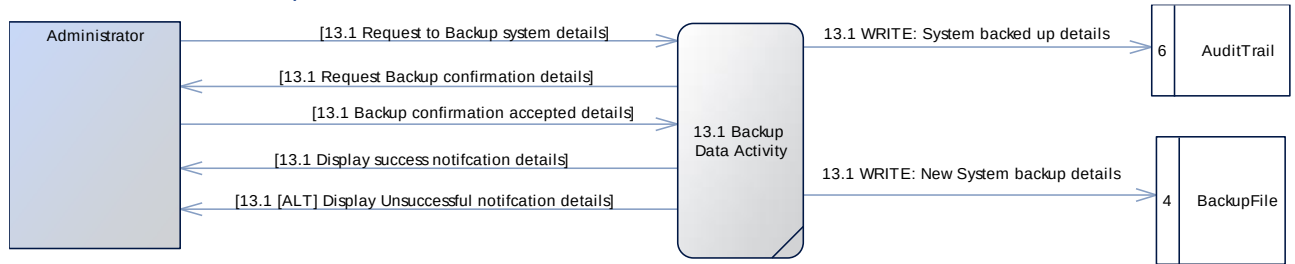


Figure 118 Backup Data mid-level diagram



3.4.3.13.2 Restore Data

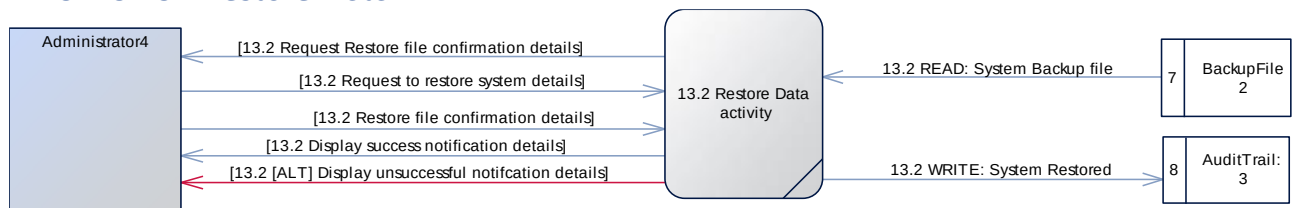


Figure 119 Restore Data mid-level diagram

3.4.3.13.3View Audit Trail

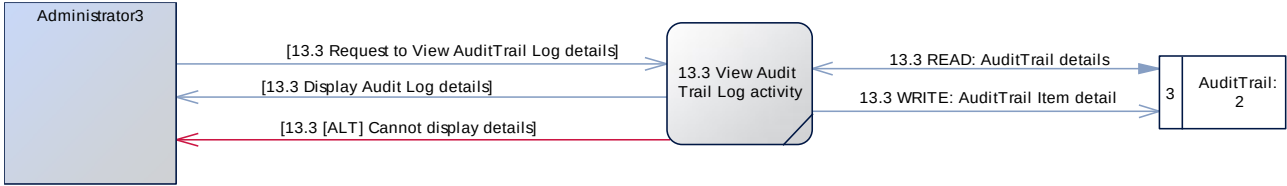


Figure 120 View Audit Trail Log mid-level diagram



3.4.3.13.4 Search Audit Trail Entry

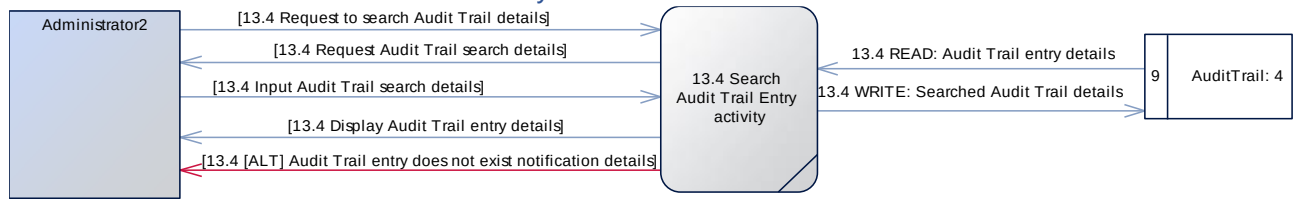


Figure 121 Search Audit Trail Entry mid-level diagram



3.4.3.13.5 Add Route

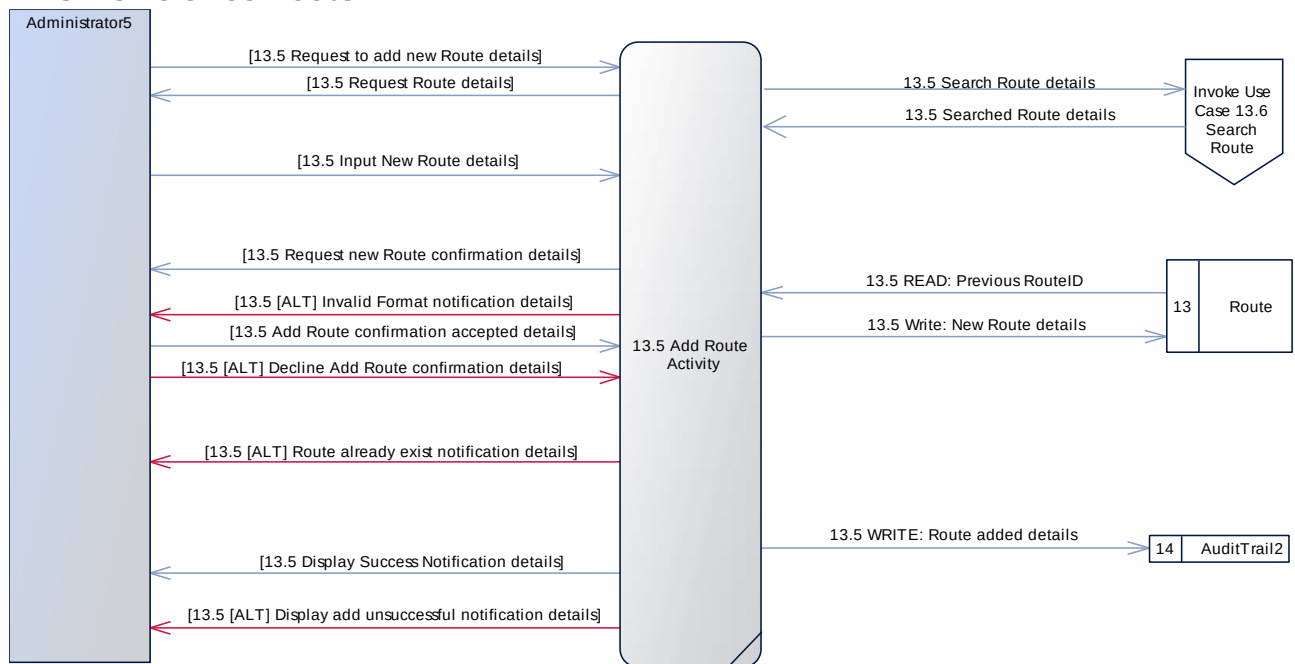


Figure 122 Add Route mid-level diagram



3.4.3.13.6 Search Route

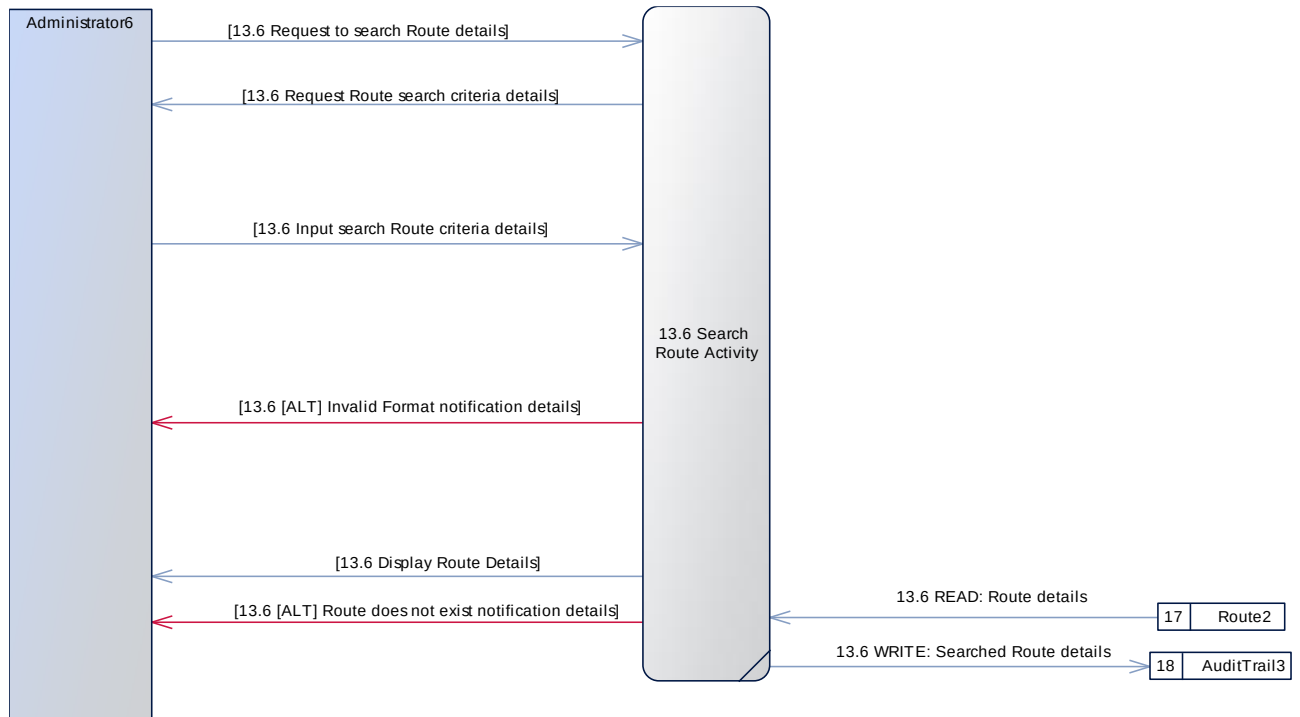


Figure 123 Search Route mid-level diagram



3.4.3.13.7 Update Route

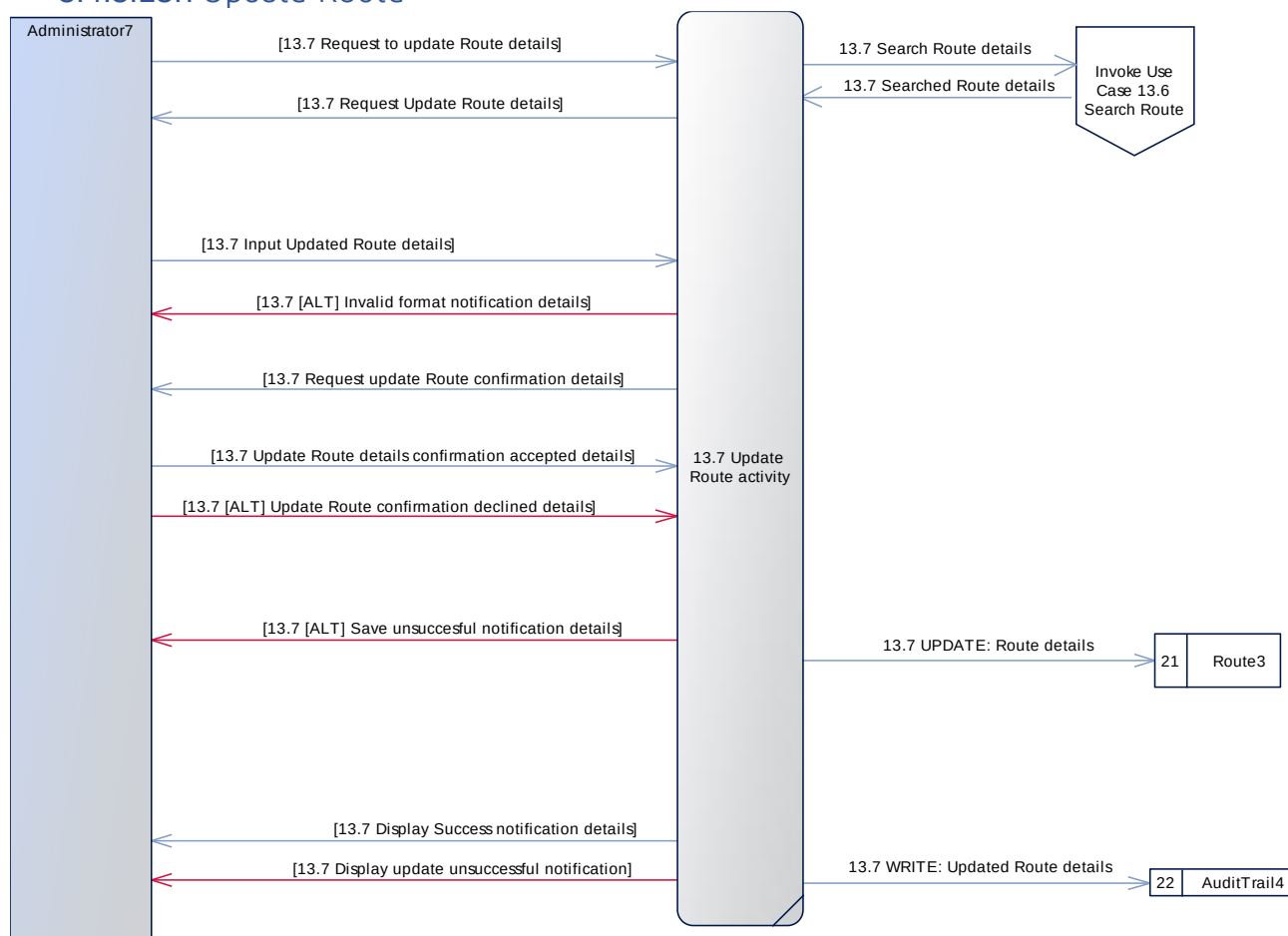


Figure 124 Update Route mid-level diagram



3.4.3.13.8 Delete Route

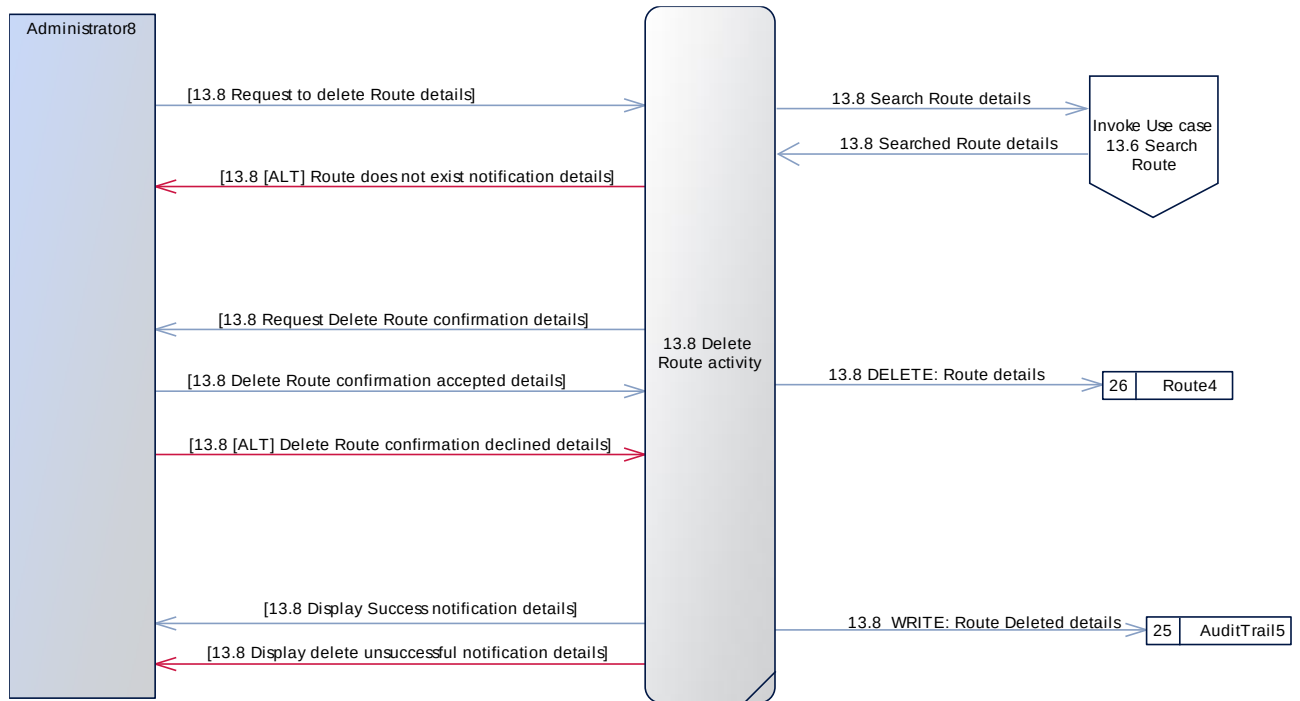


Figure 125 Delete Route mid-level diagram



3.4.4 Primitive Level Data Flow Diagrams

3.4.4.1 Sub-system 1: User

3.4.4.1.1 Login

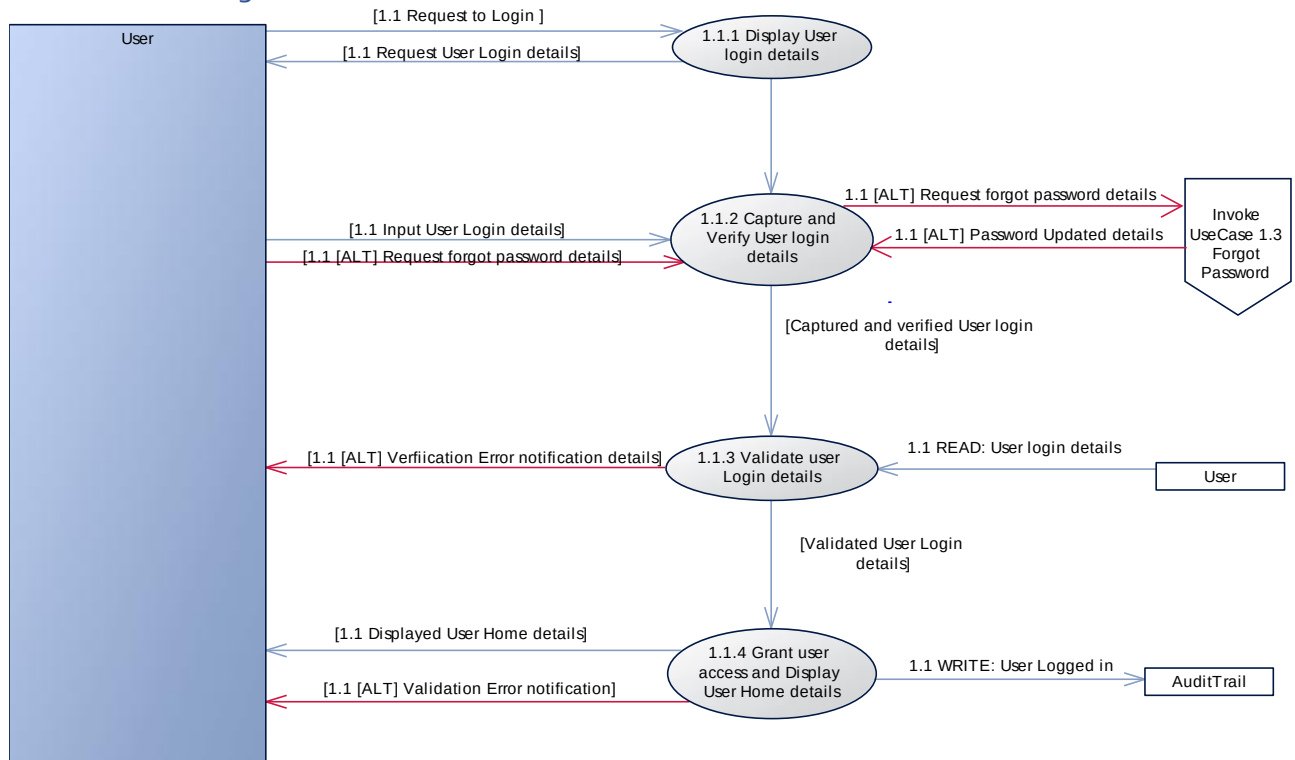


Figure 126 Login Primitive Diagram



3.4.4.1.2 Logout

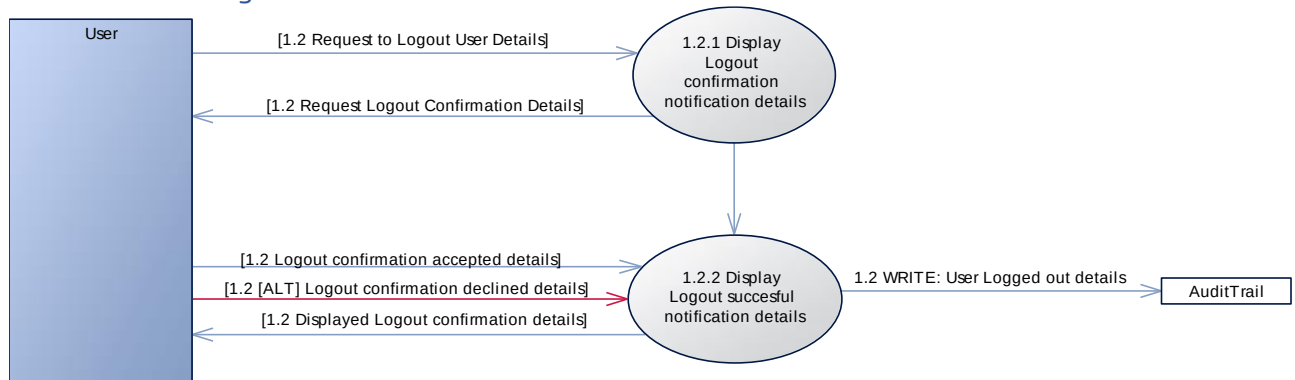


Figure 127 Logout Primitive Diagram



3.4.4.1.3 Forgot Password

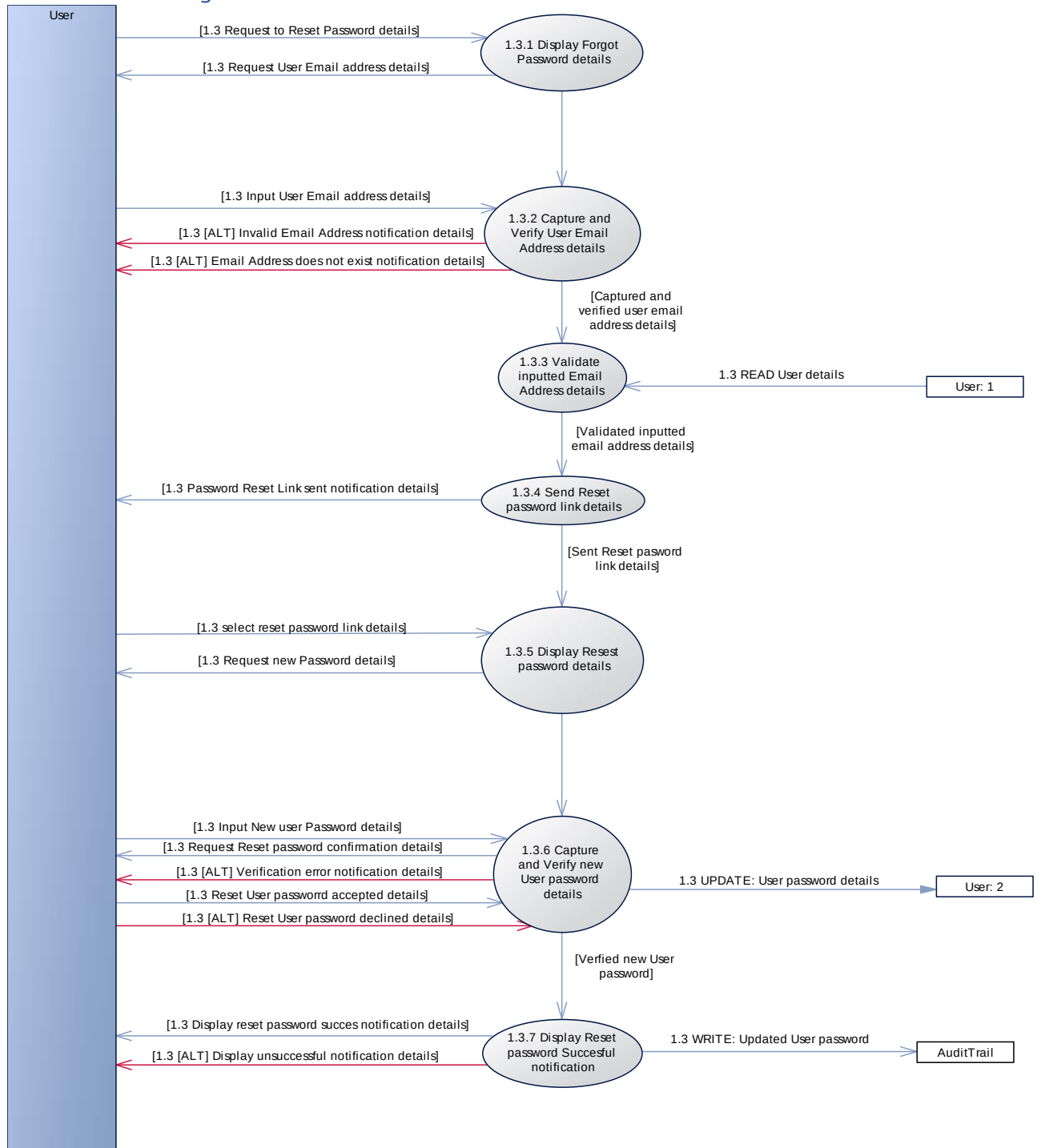


Figure 128 Forgot Password Primitive Diagram



3.4.4.1.4 Add User

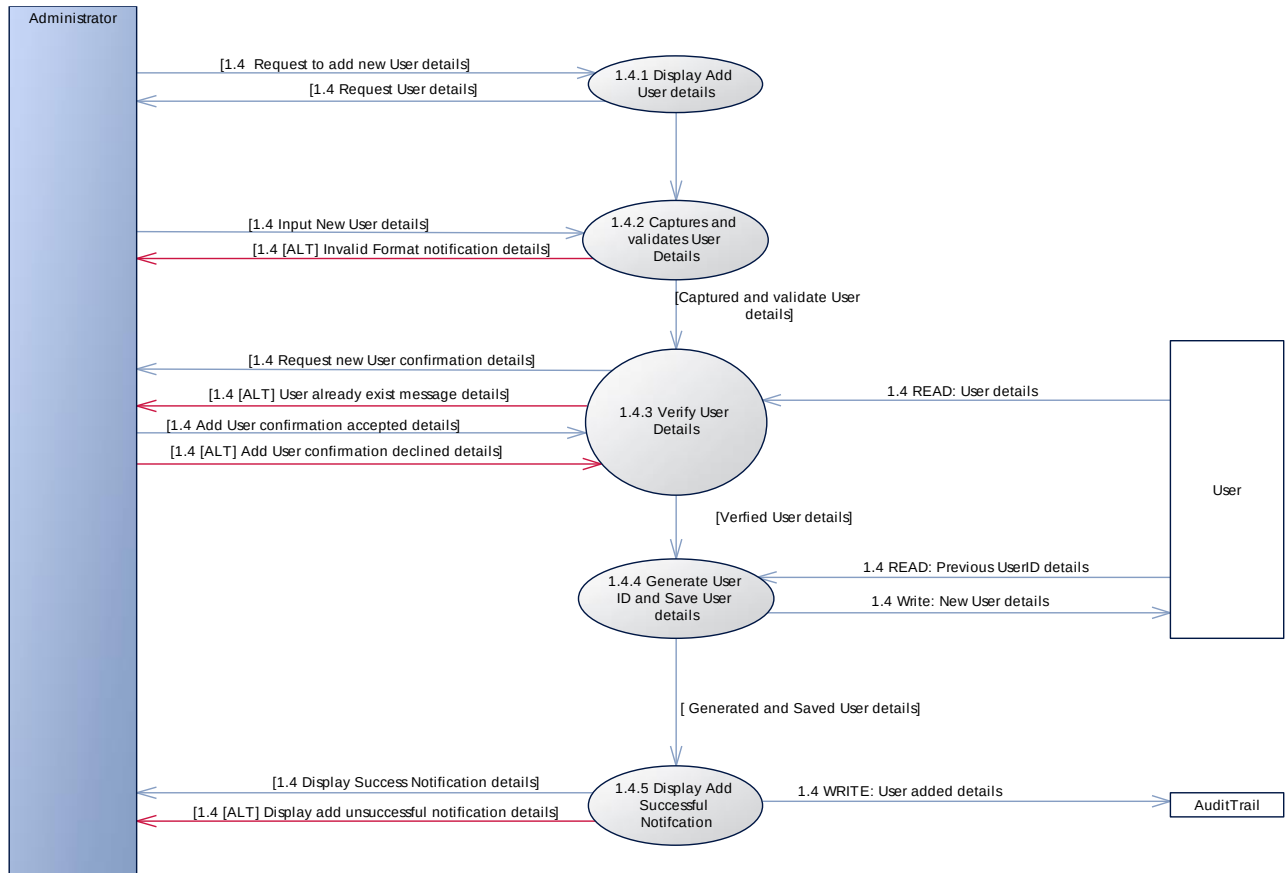


Figure 129 Add User Primitive Diagram



3.4.4.1.5 Search User

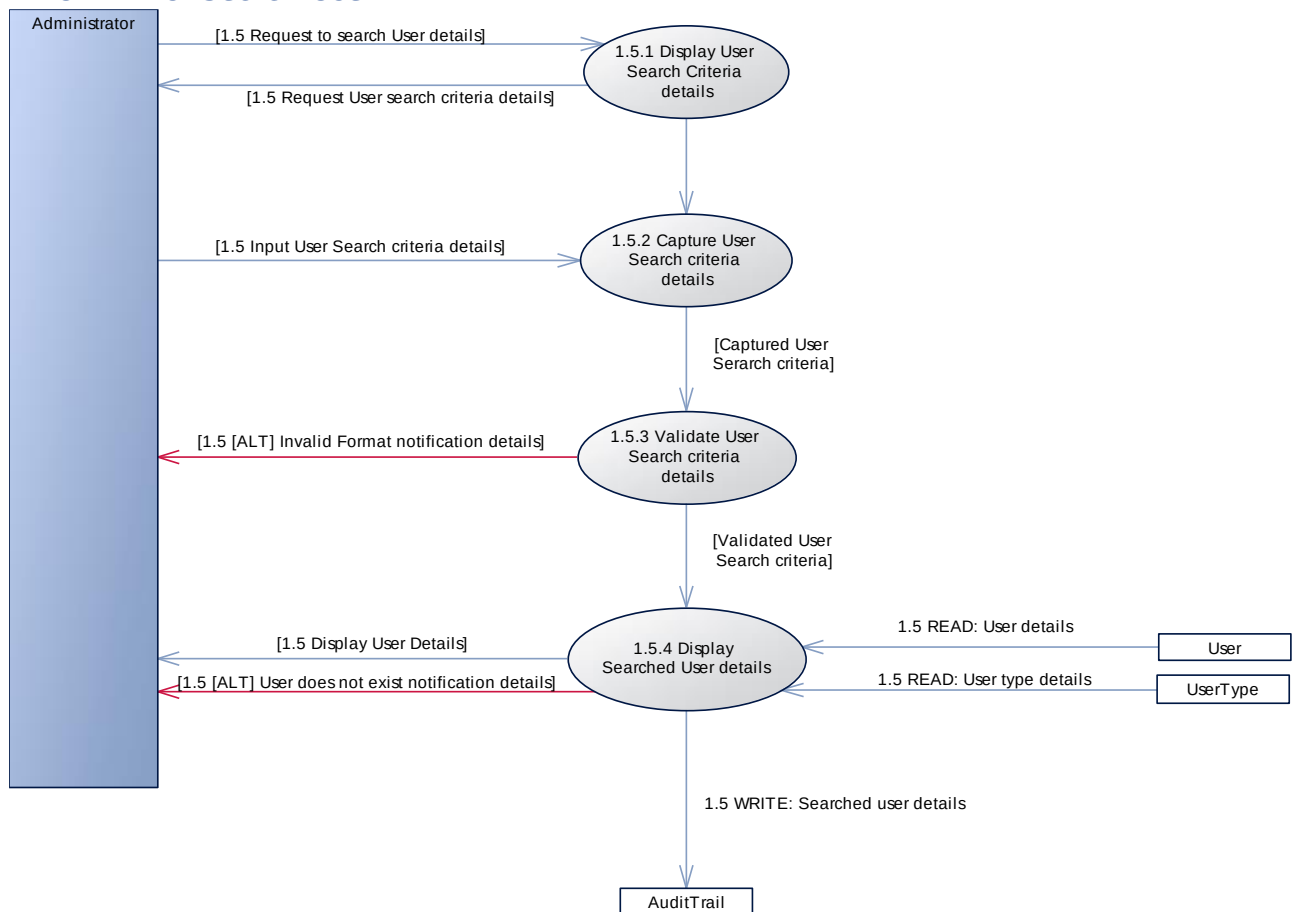


Figure 130 Search User Primitive Diagram



3.4.4.1.6 Update User

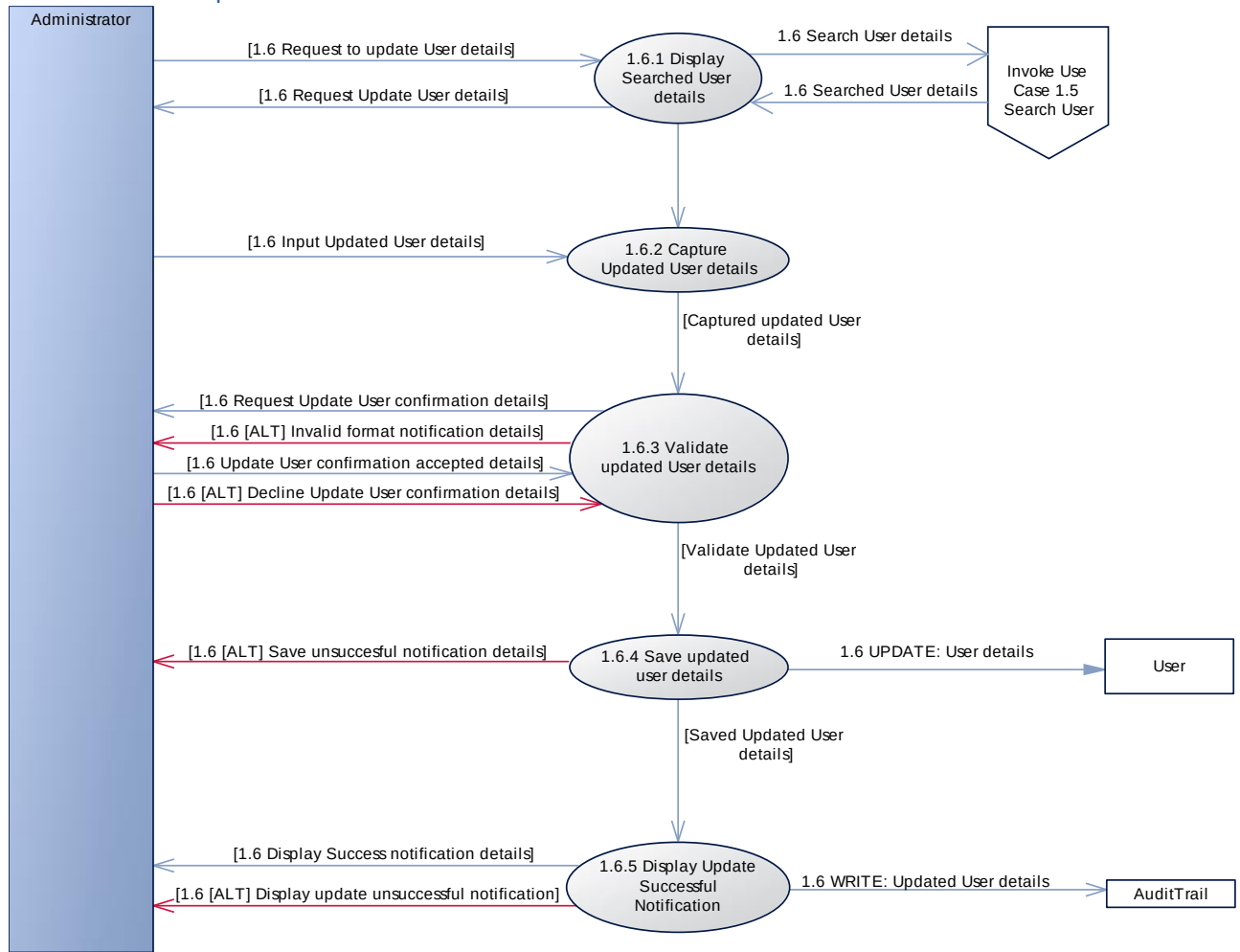


Figure 131 Update User Primitive Diagram



3.4.4.1.7 Delete User

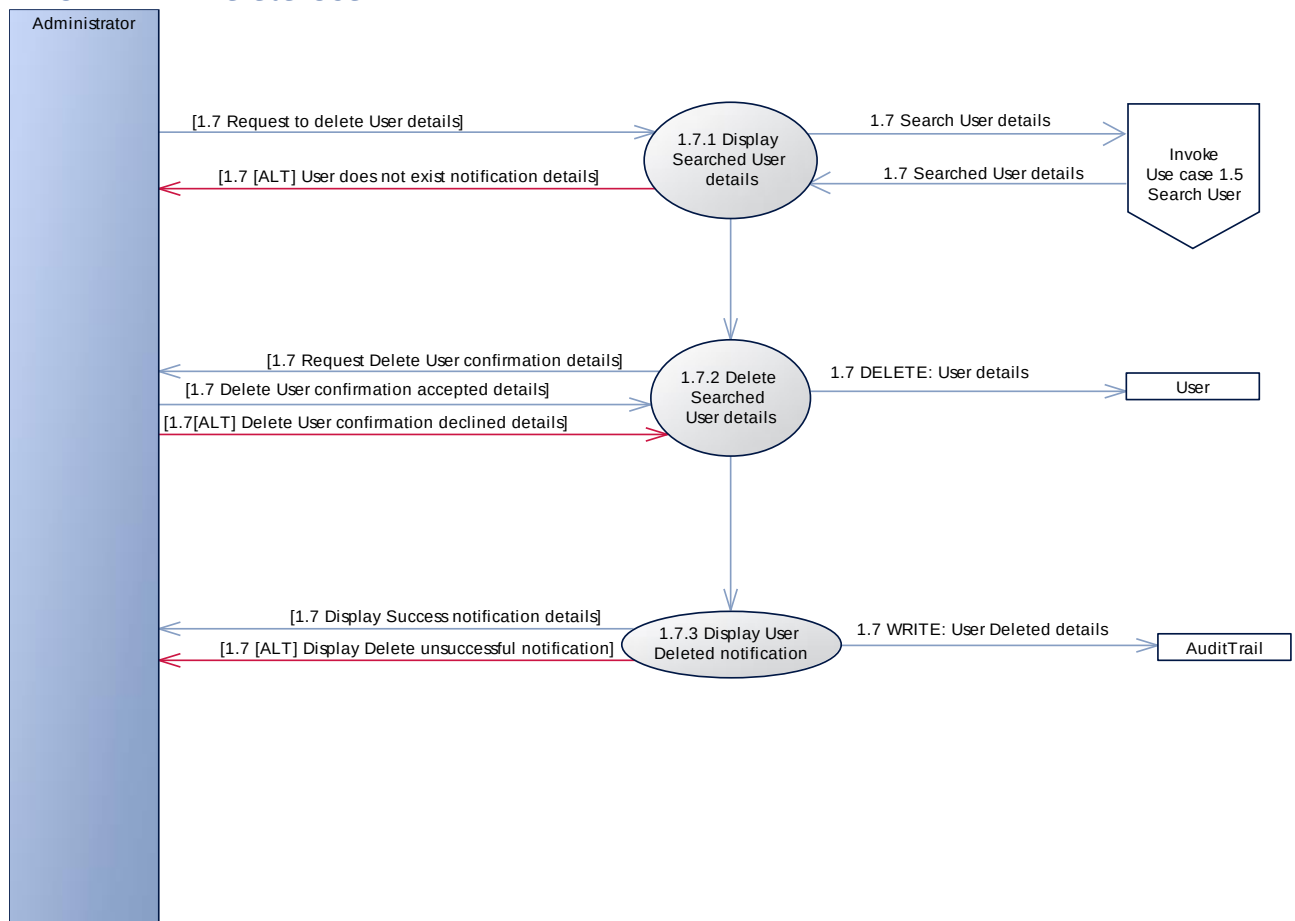


Figure 132 Delete User Primitive Diagram



3.4.4.1.8 Add User Type

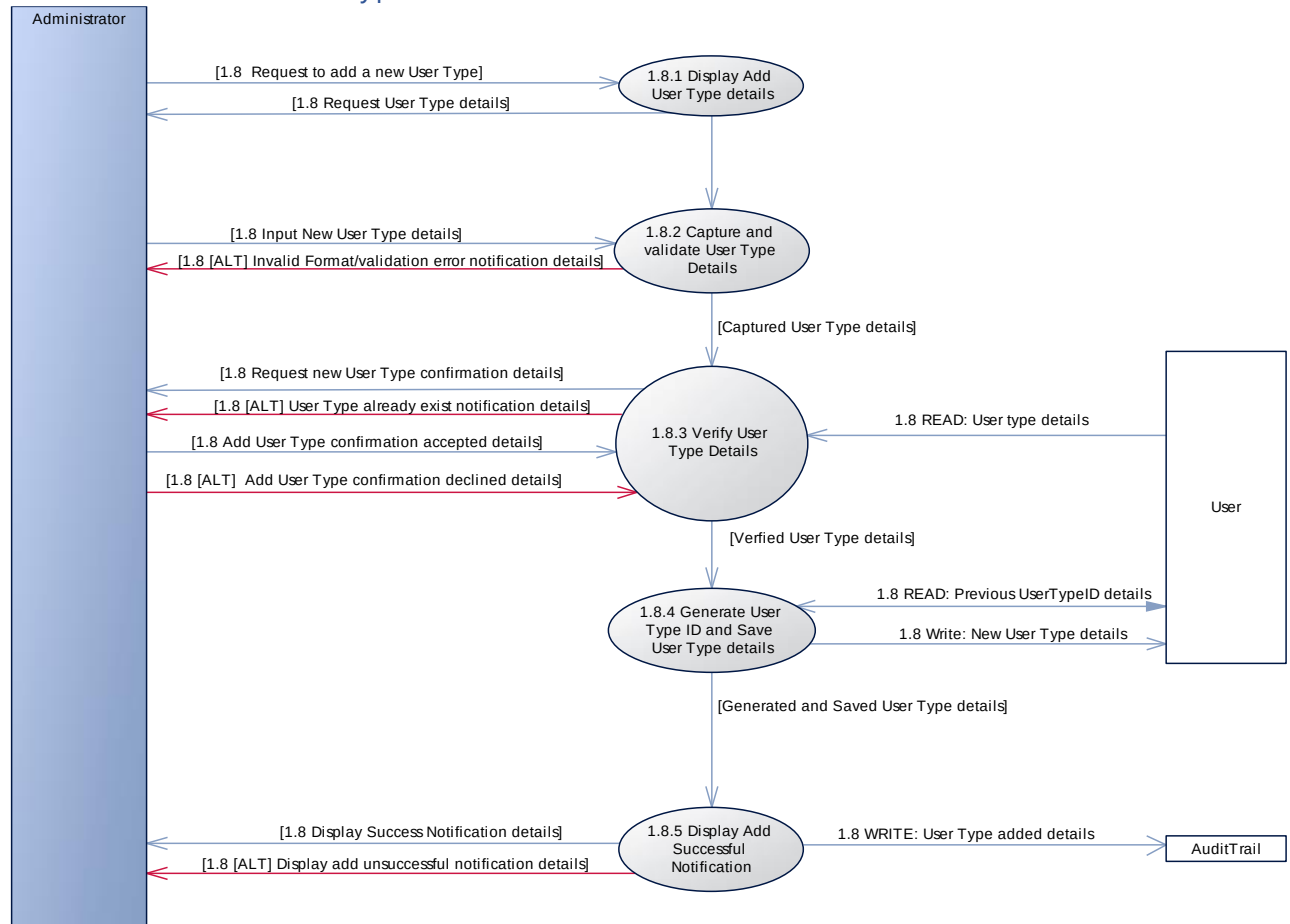


Figure 133 Add User Type Primitive Diagram



3.4.4.1.9 Search User Type

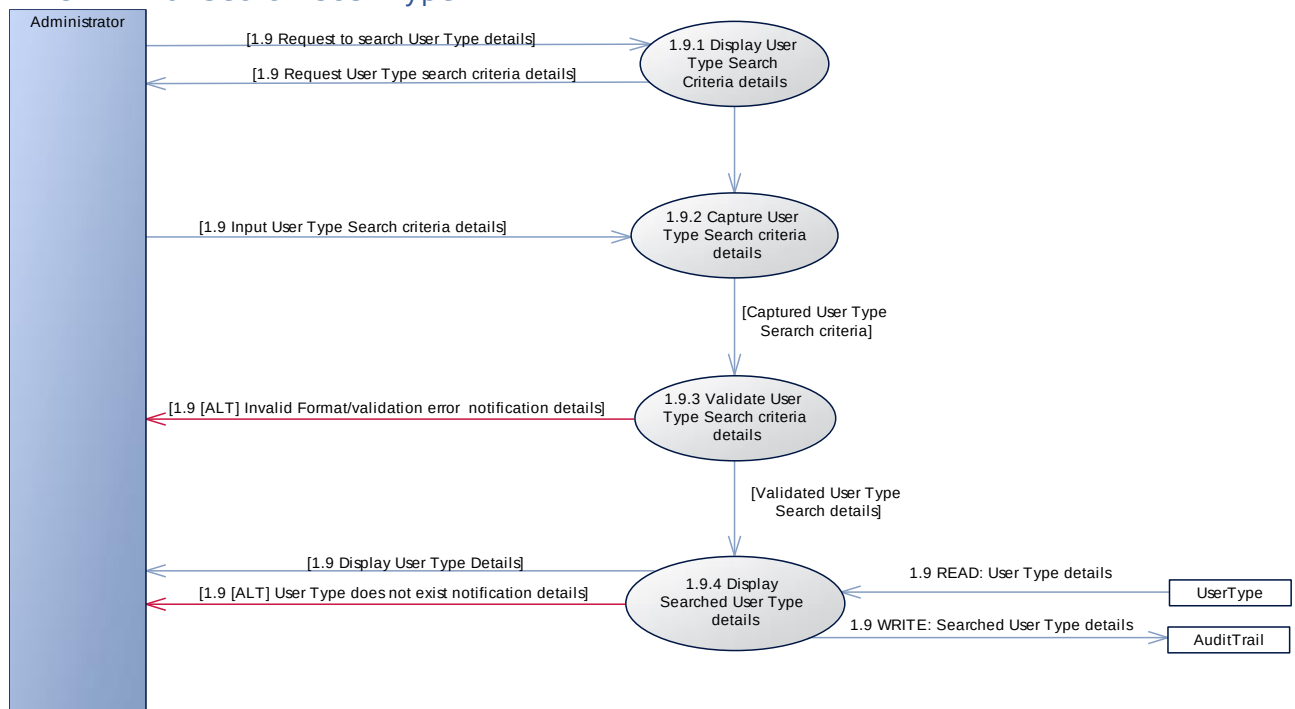


Figure 134 Search User Type Primitive Diagram



3.4.4.1.10 Update User Type

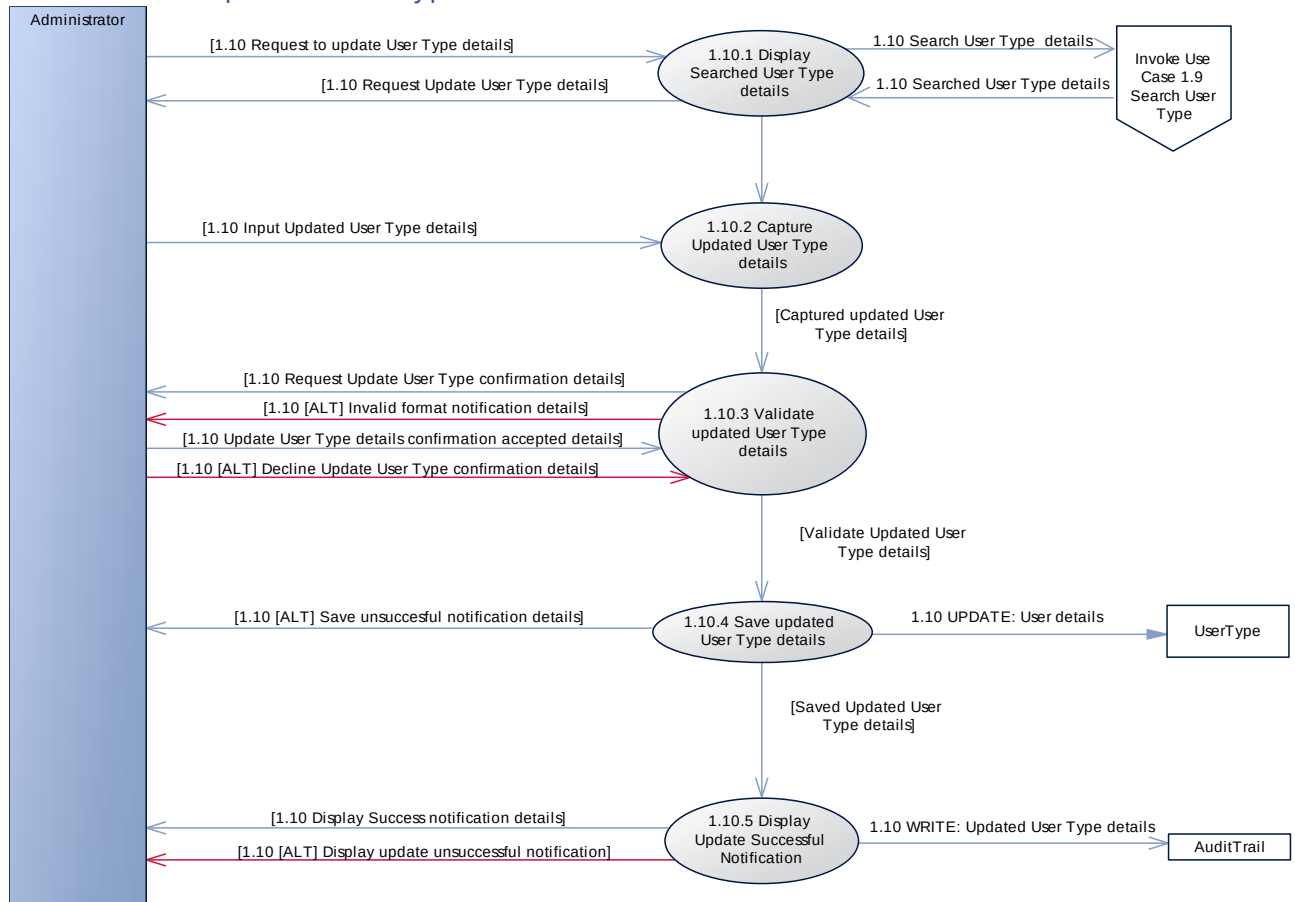


Figure 135 Update User Type Primitive Diagram



3.4.4.1.11 Delete User Type

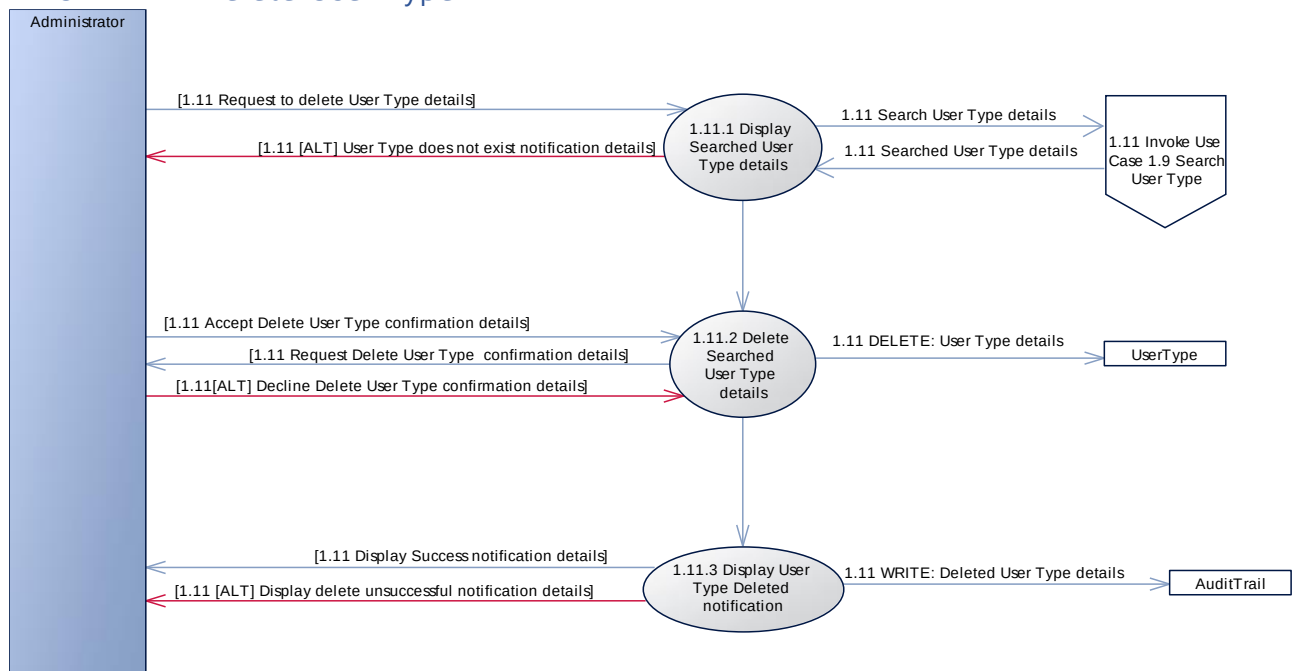


Figure 136 Delete User Type Primitive Diagram



3.4.4.2 Sub-system 2: Mechanic Schedule

3.4.4.2.1 Complete Job

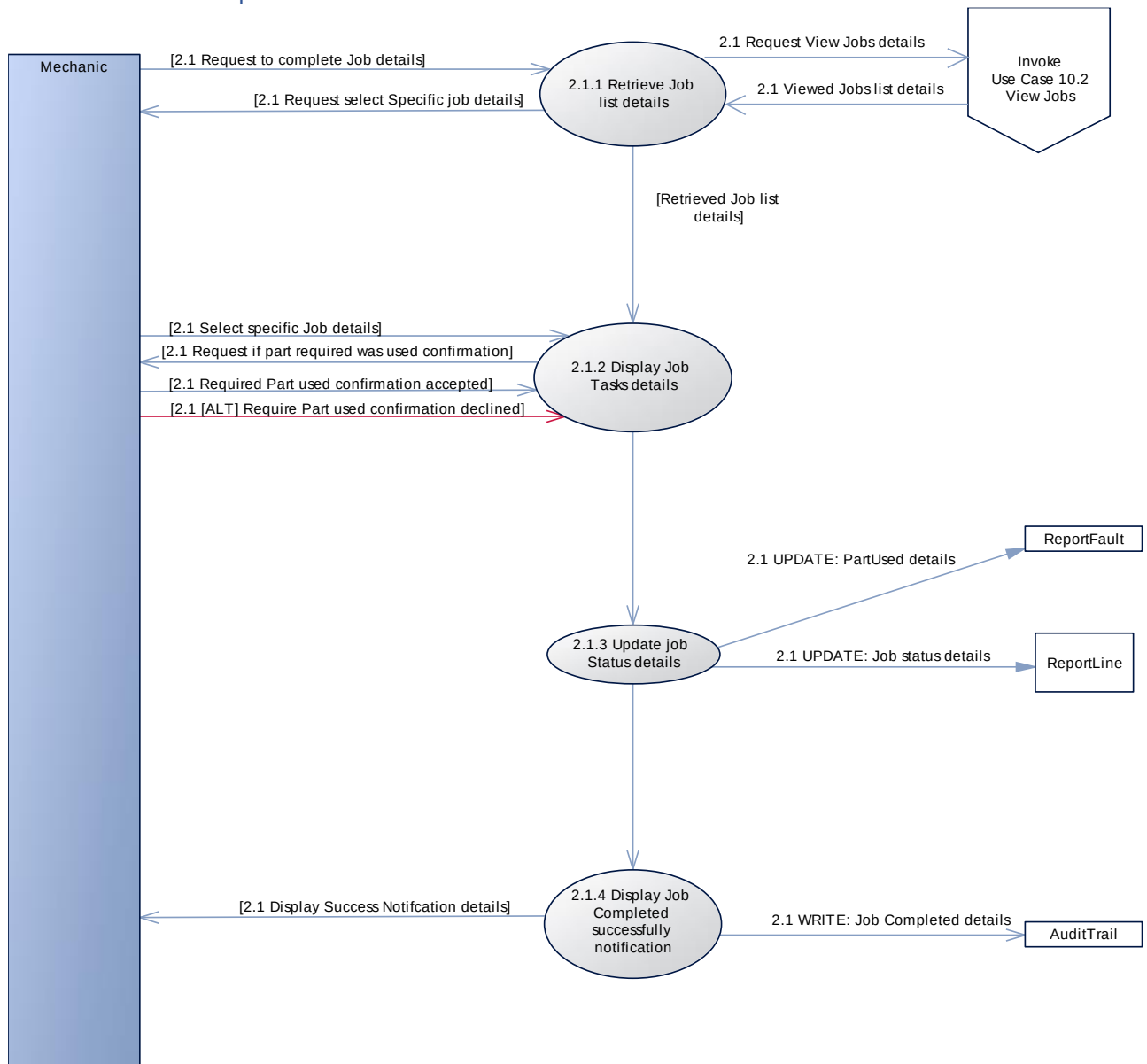


Figure 137 Complete Job Primitive Diagram



3.4.4.2.2 View Job Tasks

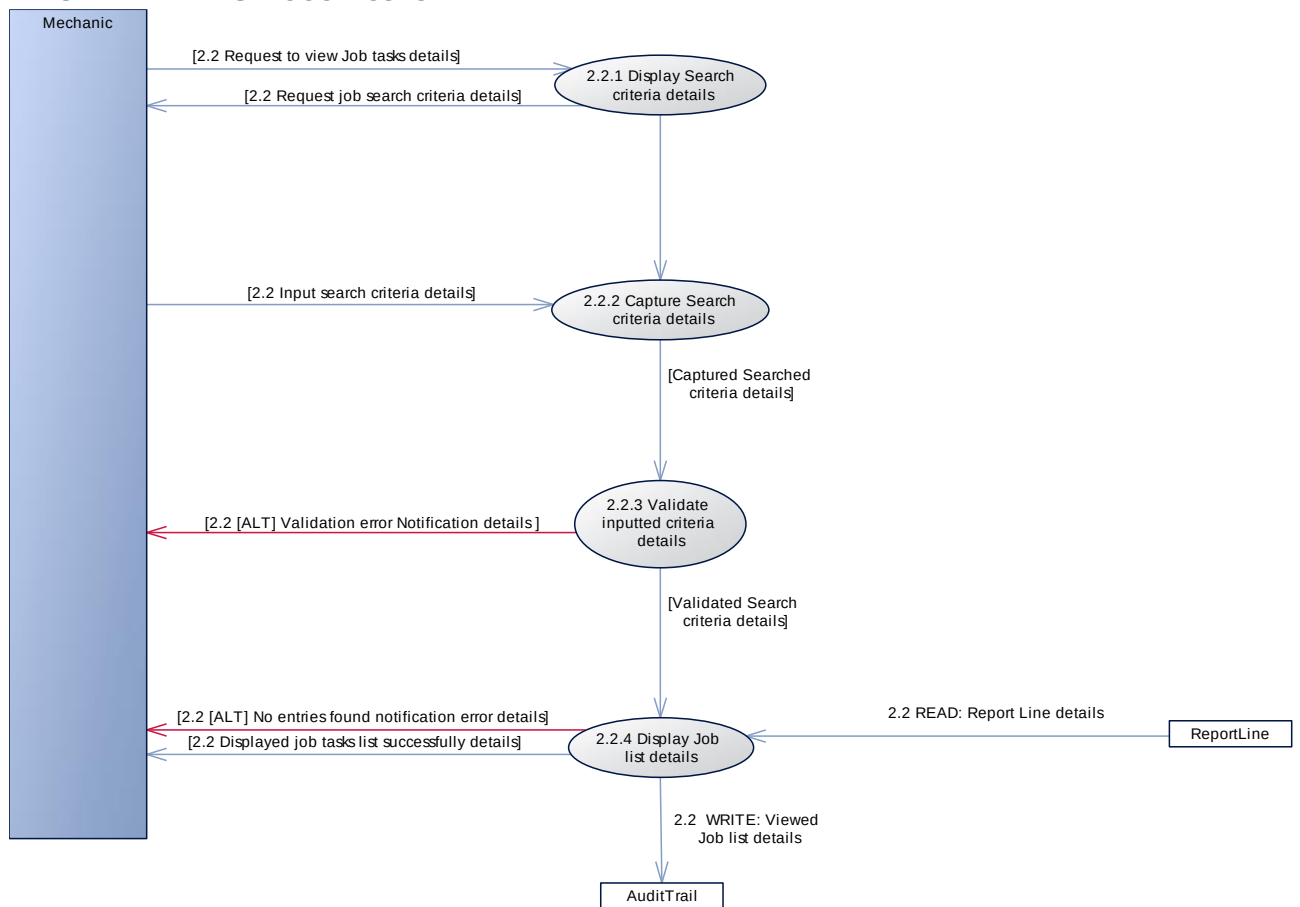


Figure 138 View Job Tasks Primitive Diagram



3.4.4.3 Sub-system 3: Tracking Hardware

3.4.4.3.1 Add Antenna

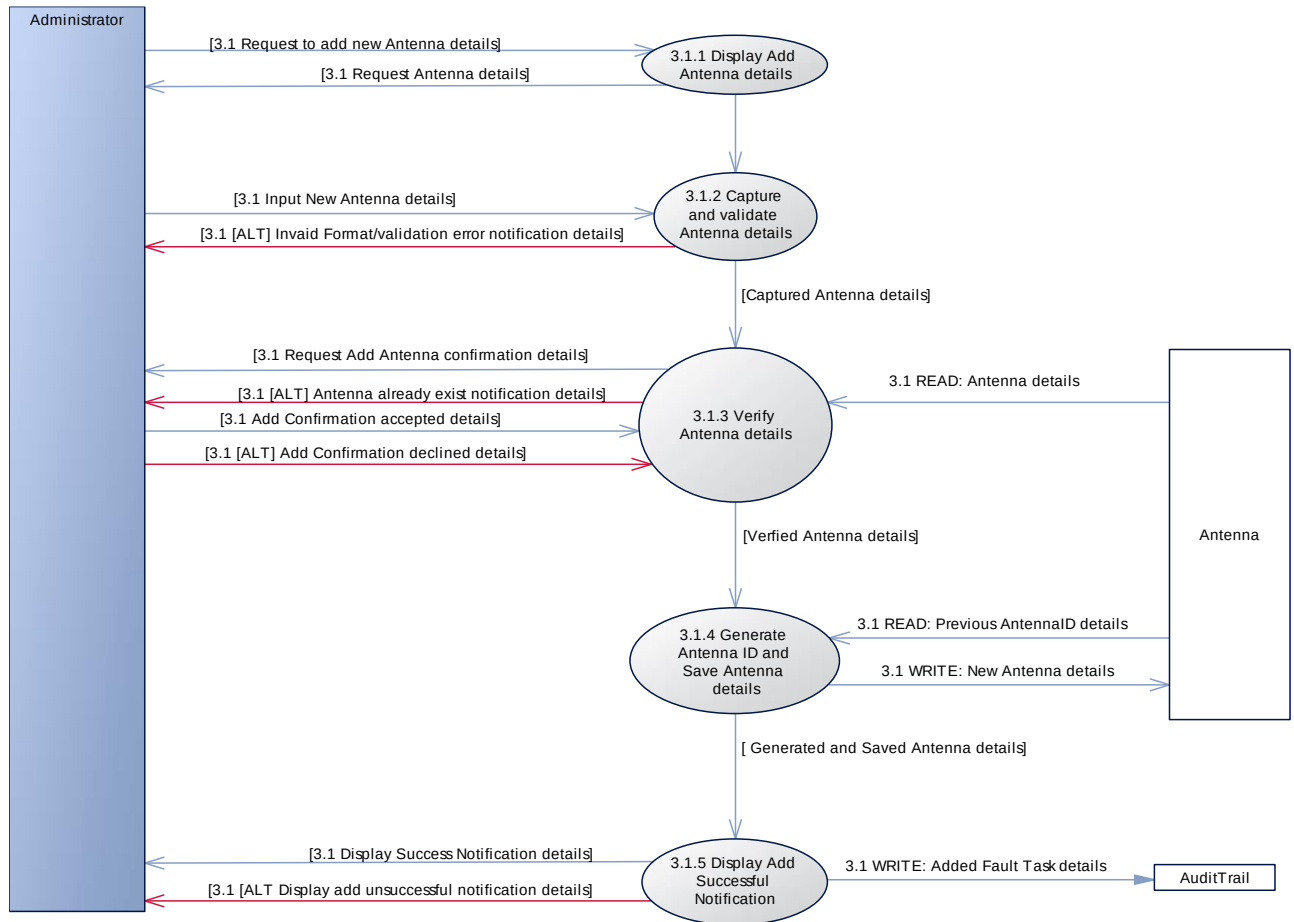


Figure 139 Add Antenna Primitive Diagram



3.4.4.3.2 Search Antenna

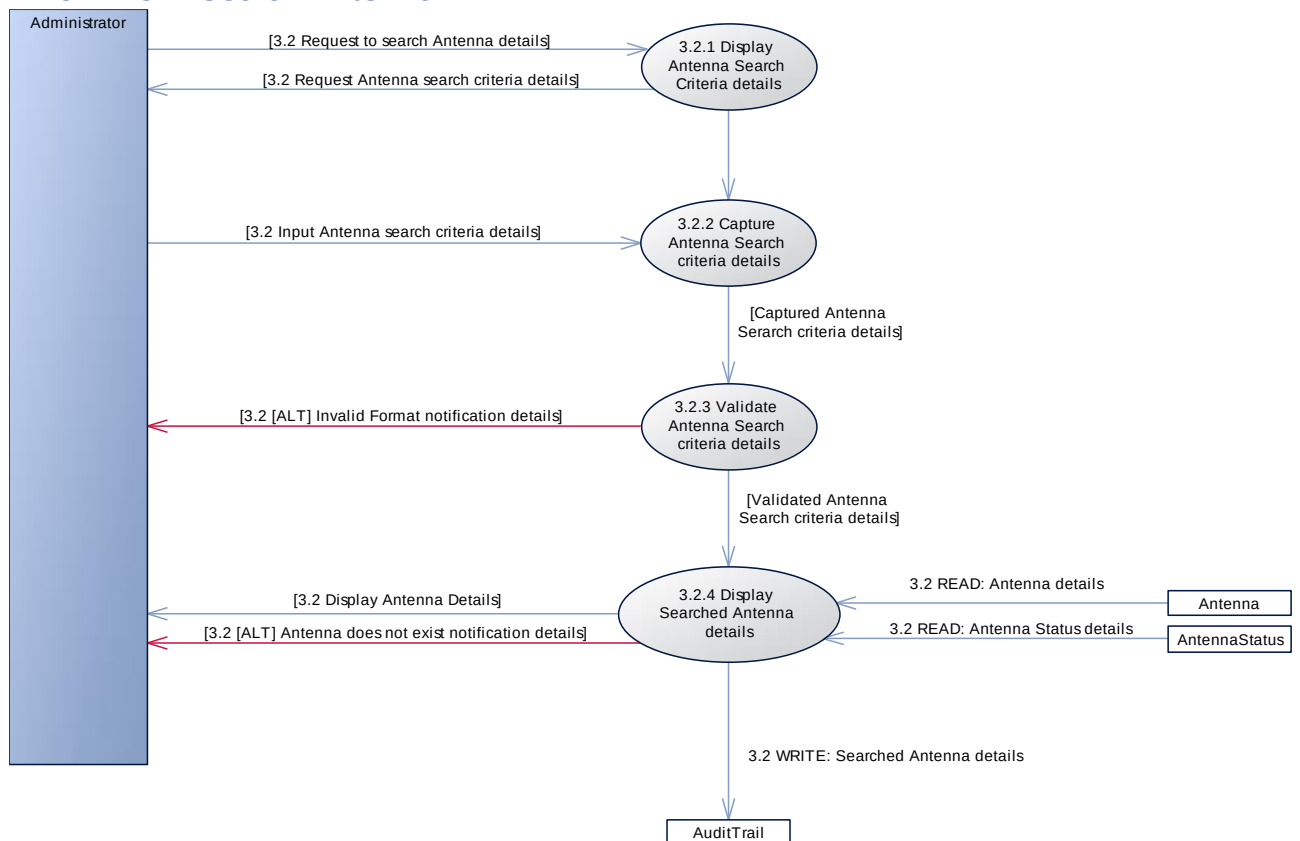


Figure 140 Search Antenna Primitive Diagram



3.4.4.3.3 Update Antenna

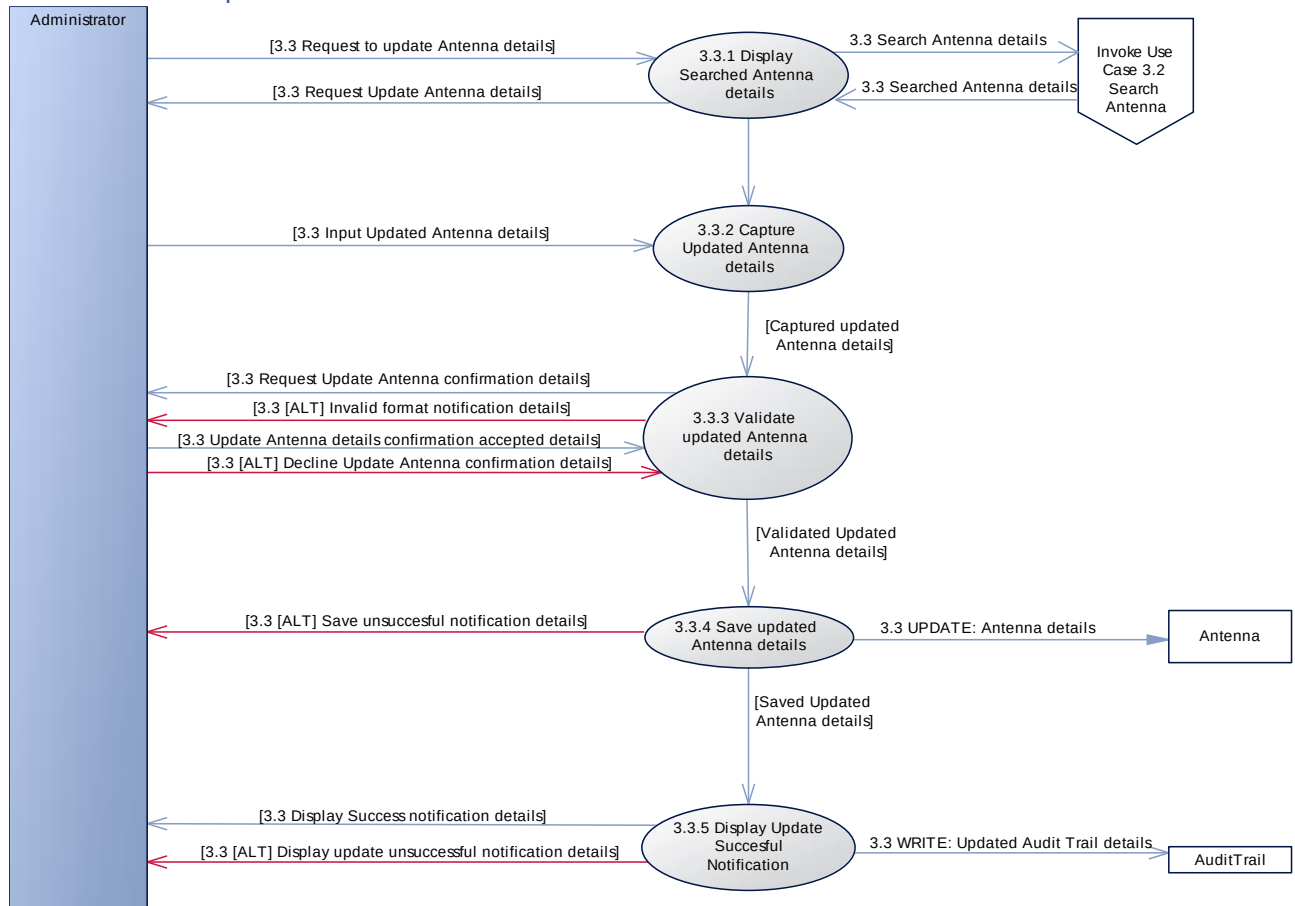


Figure 141 Update Antenna Primitive Diagram



3.4.4.3.4 Delete Antenna

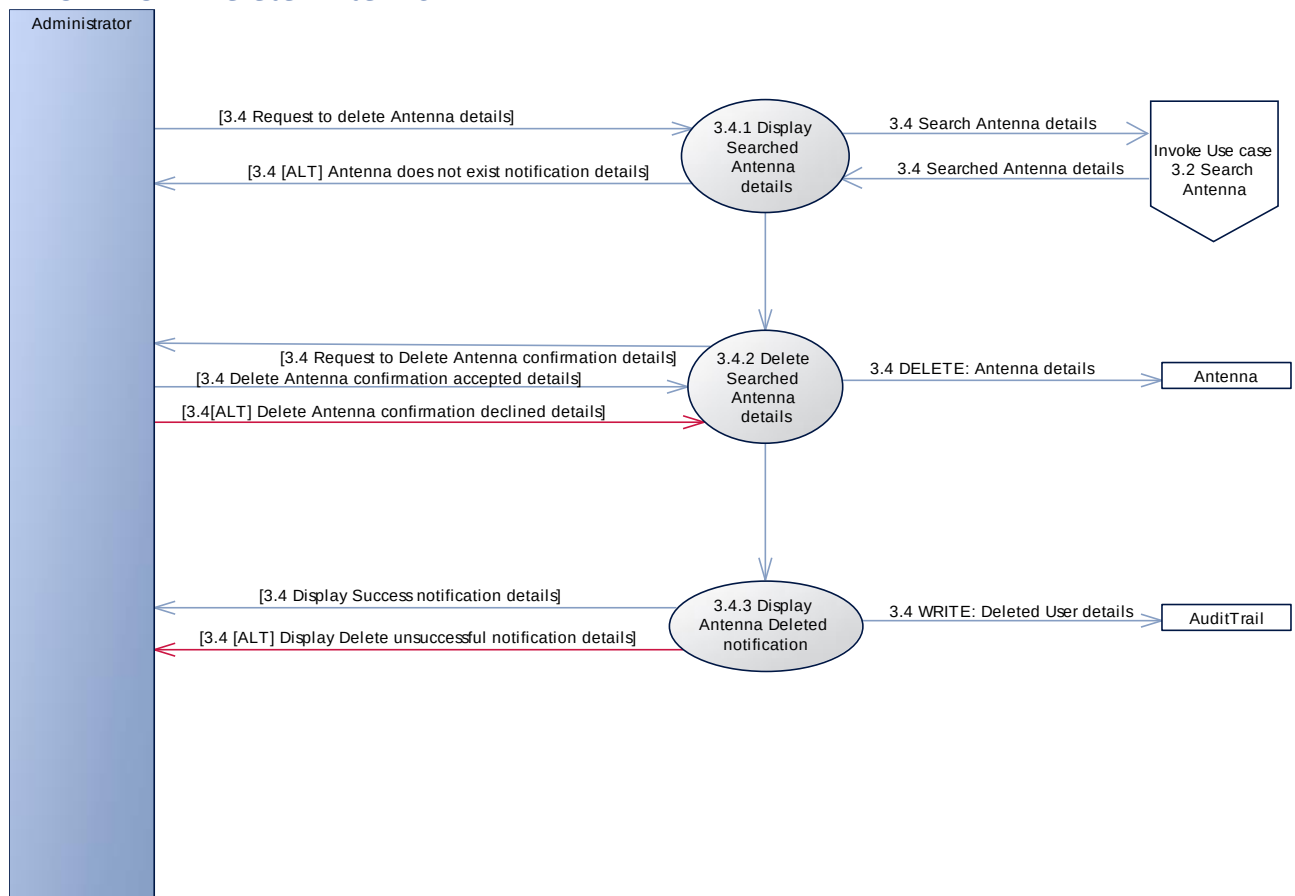


Figure 142 Delete Antenna Primitive Diagram



3.4.4.3.5 Add Tag

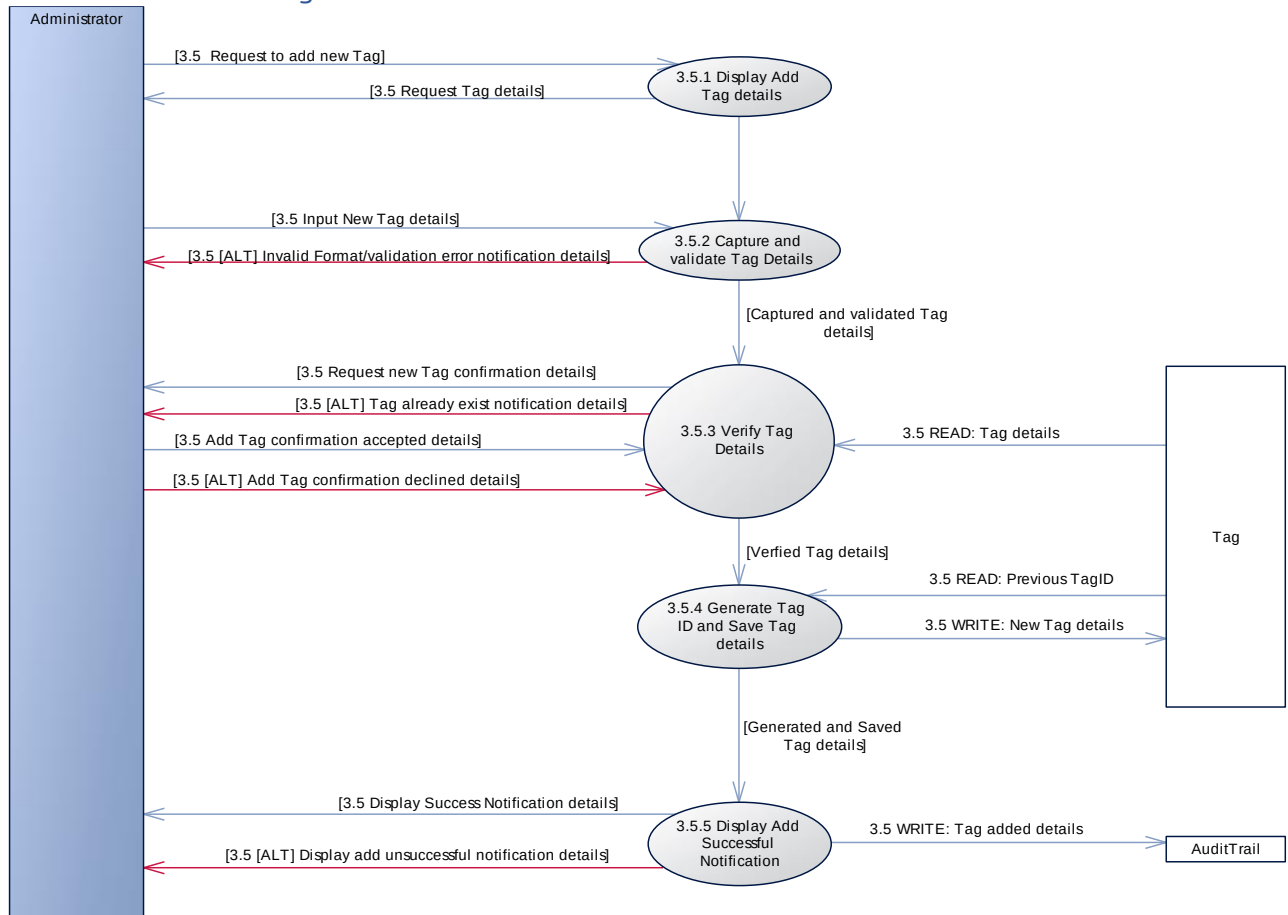


Figure 143 Add Tag Primitive Diagram



3.4.4.3.6 Search Tag

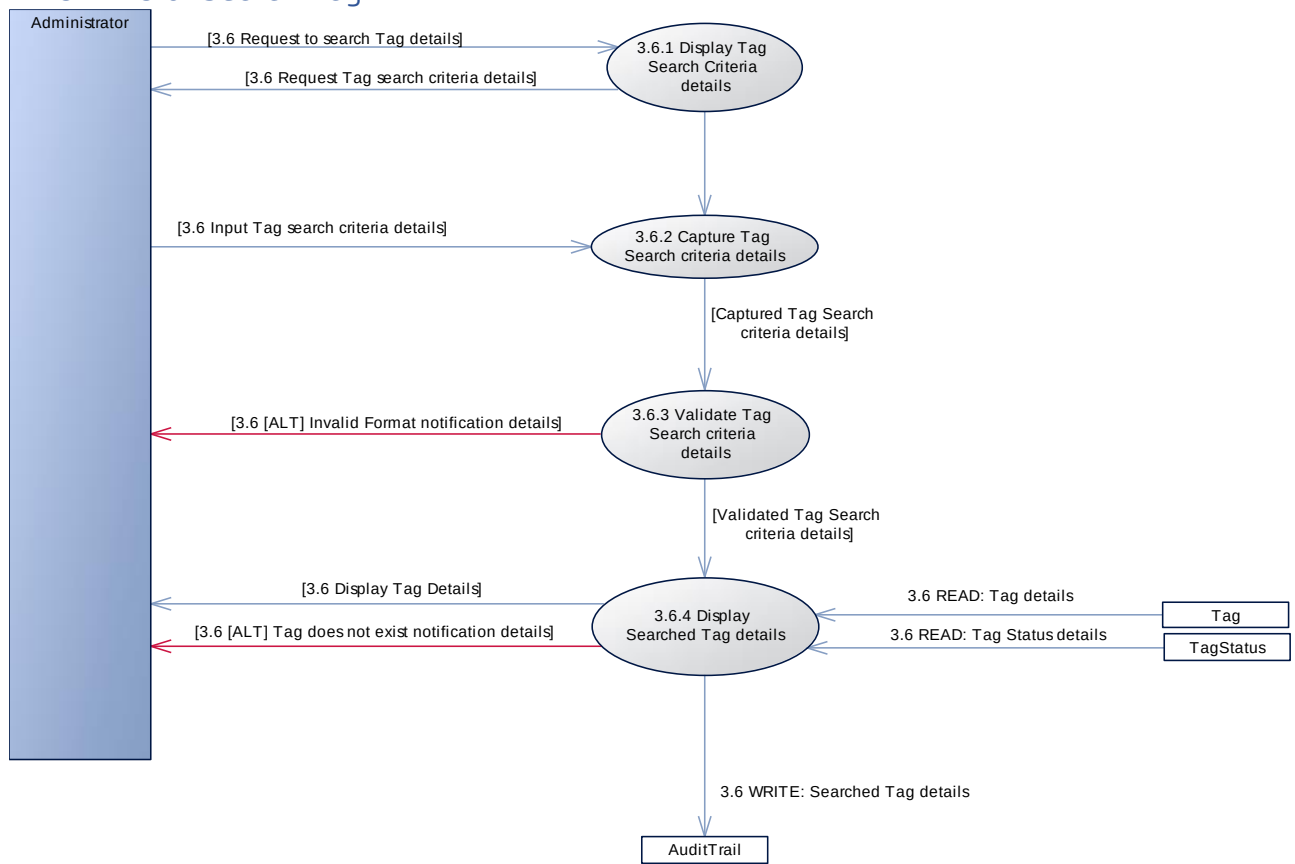


Figure 144 Search Tag Primitive Diagram



3.4.4.3.7 Update Tag

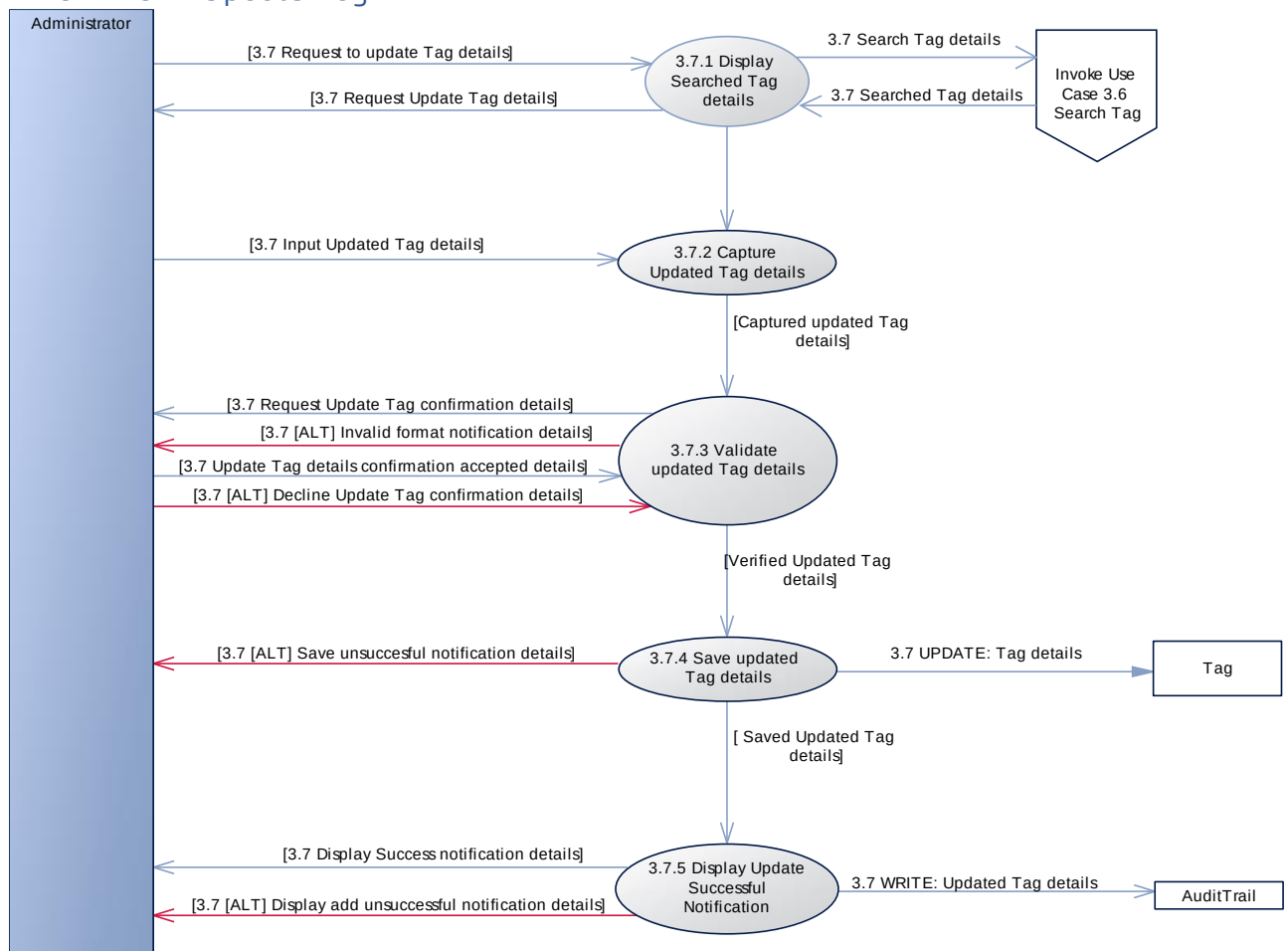


Figure 145 Update Tag Primitive Diagram



3.4.4.3.8 Delete Tag

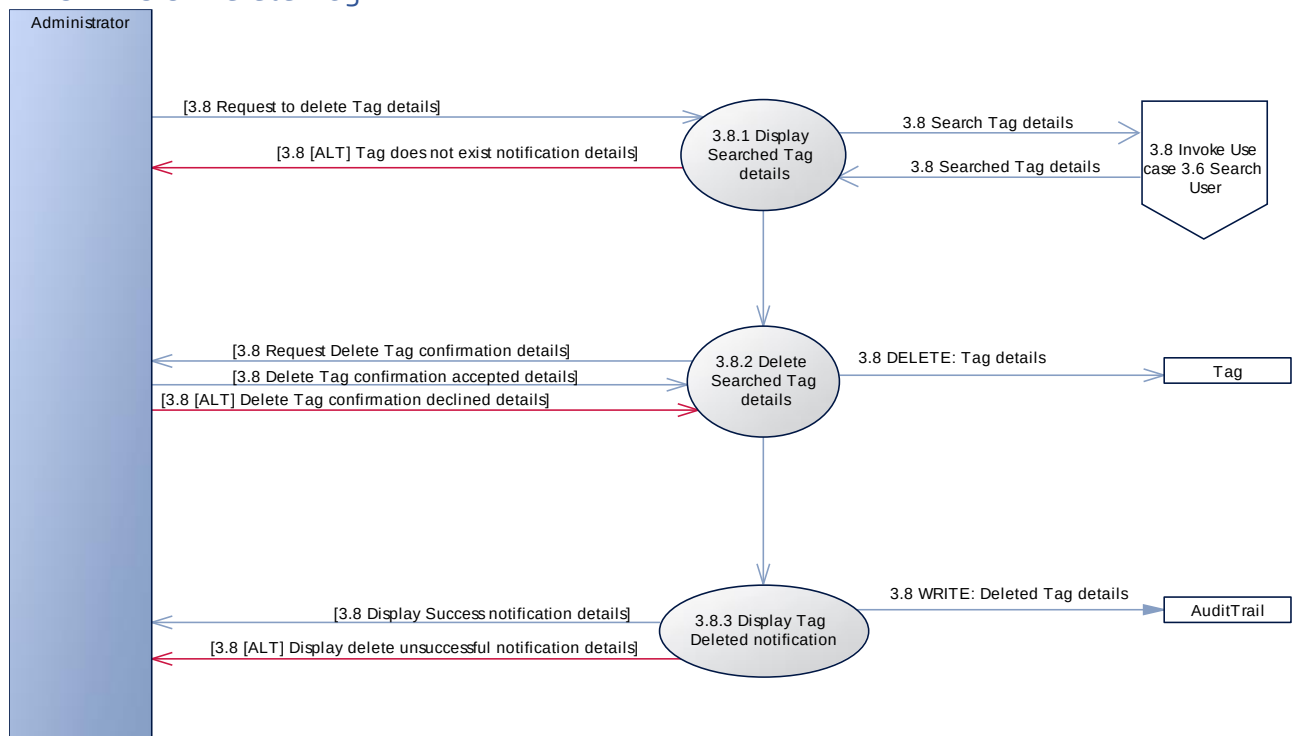


Figure 146 Delete Tag Primitive Diagram



3.4.4.4 Sub-system 4: Student

3.4.4.4.1 Add Student

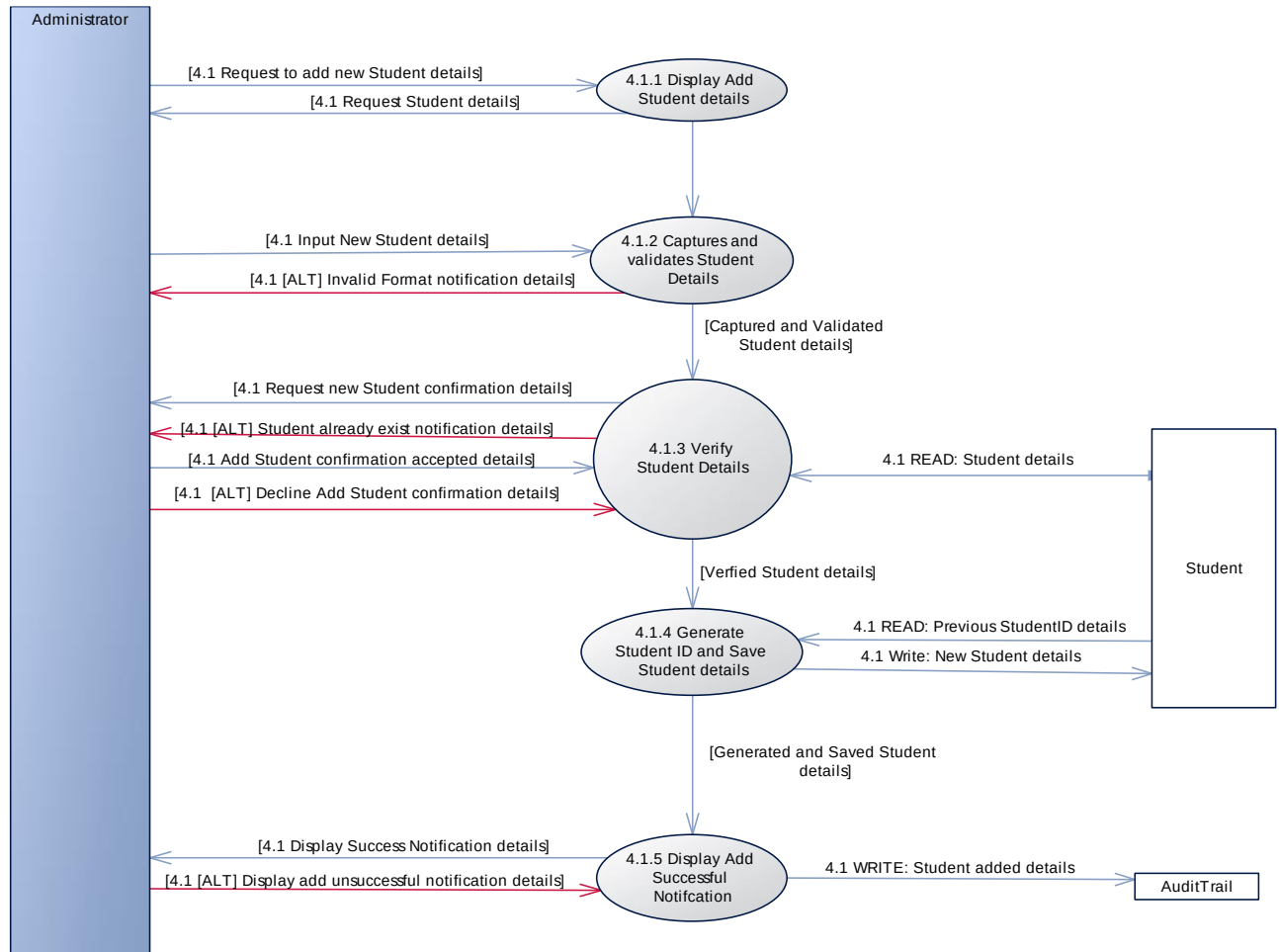


Figure 147 Add Student Primitive Diagram



3.4.4.4.2 Search Student

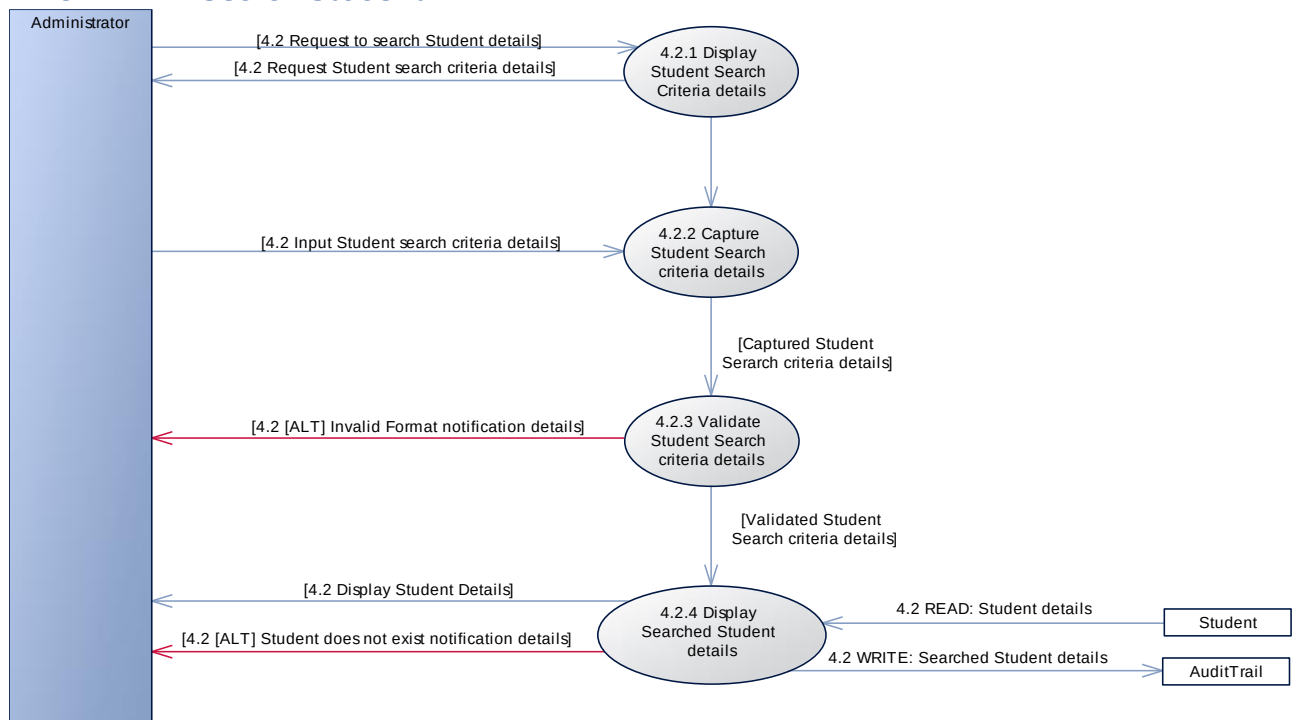


Figure 148 Search Student Primitive Diagram



3.4.4.4.3 Update Student

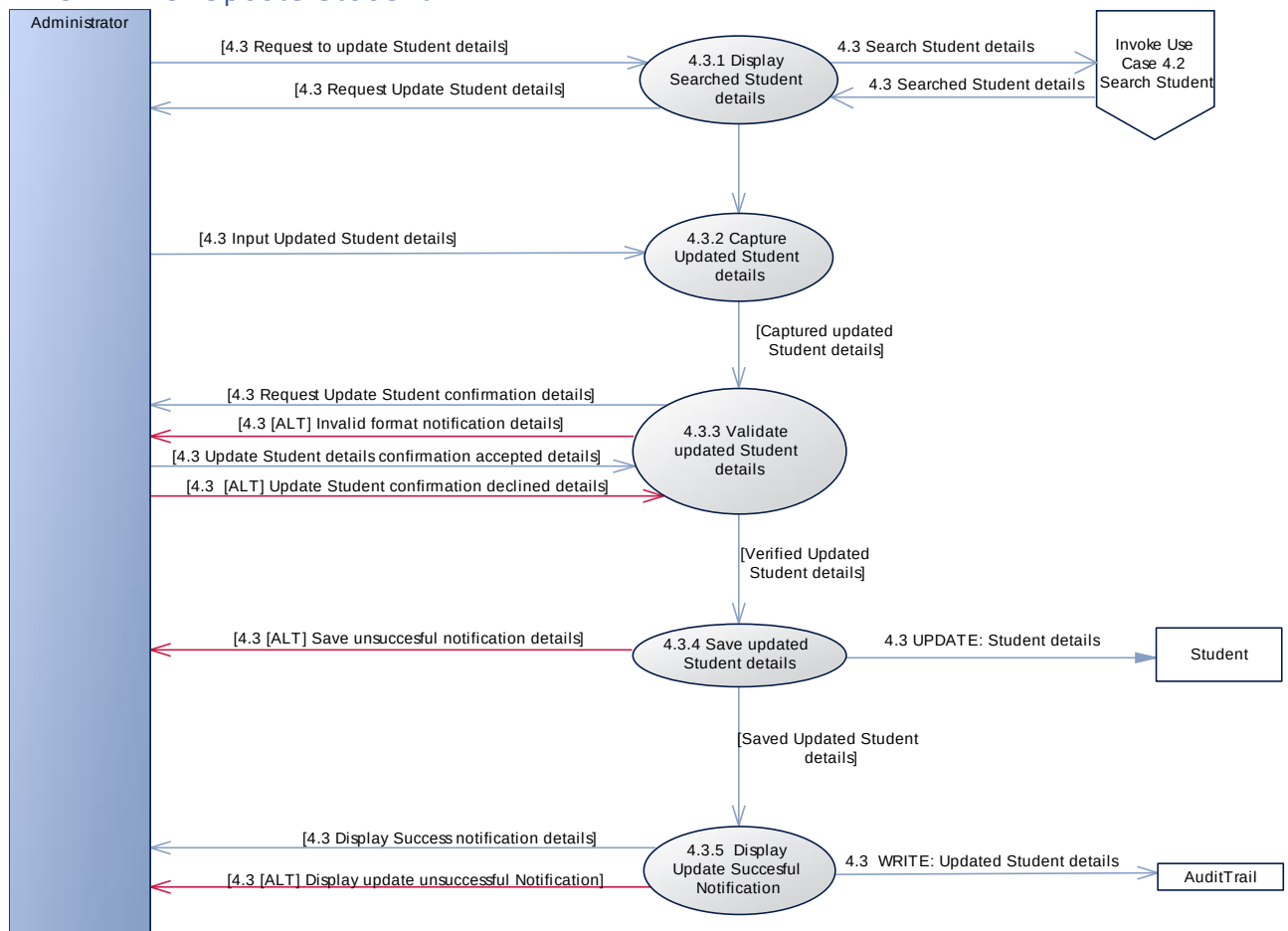


Figure 149 Update Student Primitive Diagram



3.4.4.4 Delete Student

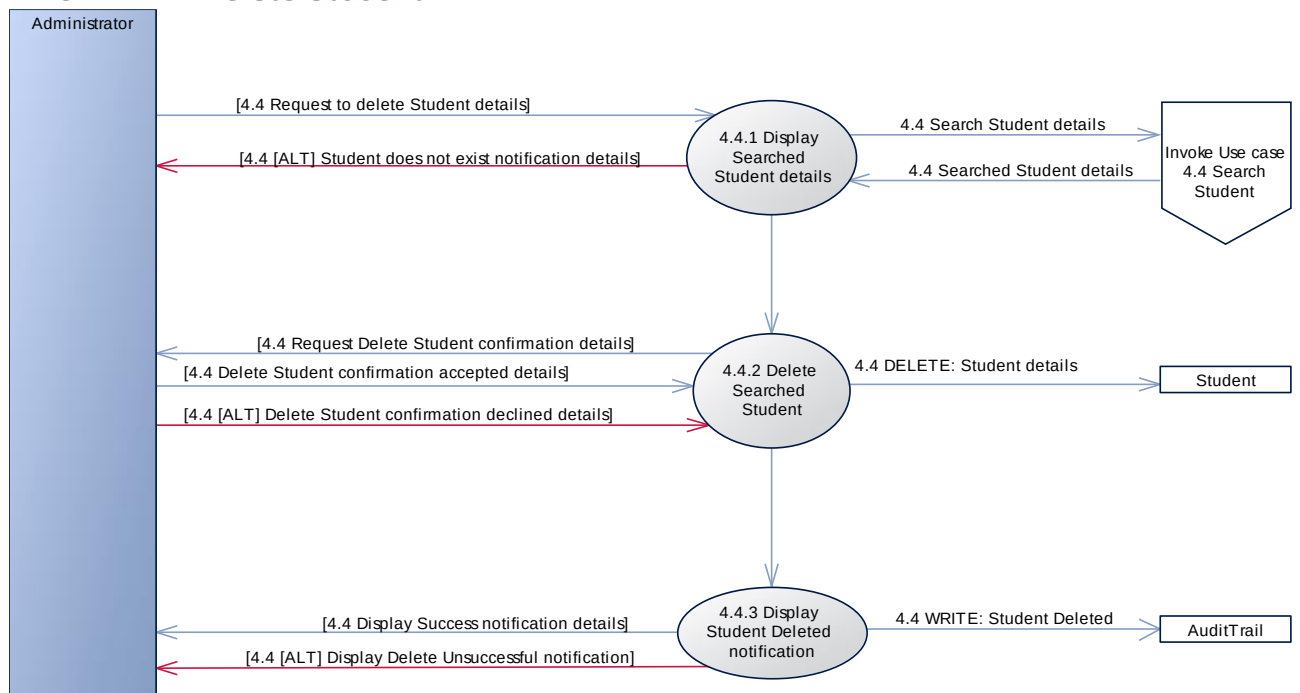


Figure 150 Delete Student Primitive Diagram



3.4.4.4.5 Capture Student Marks

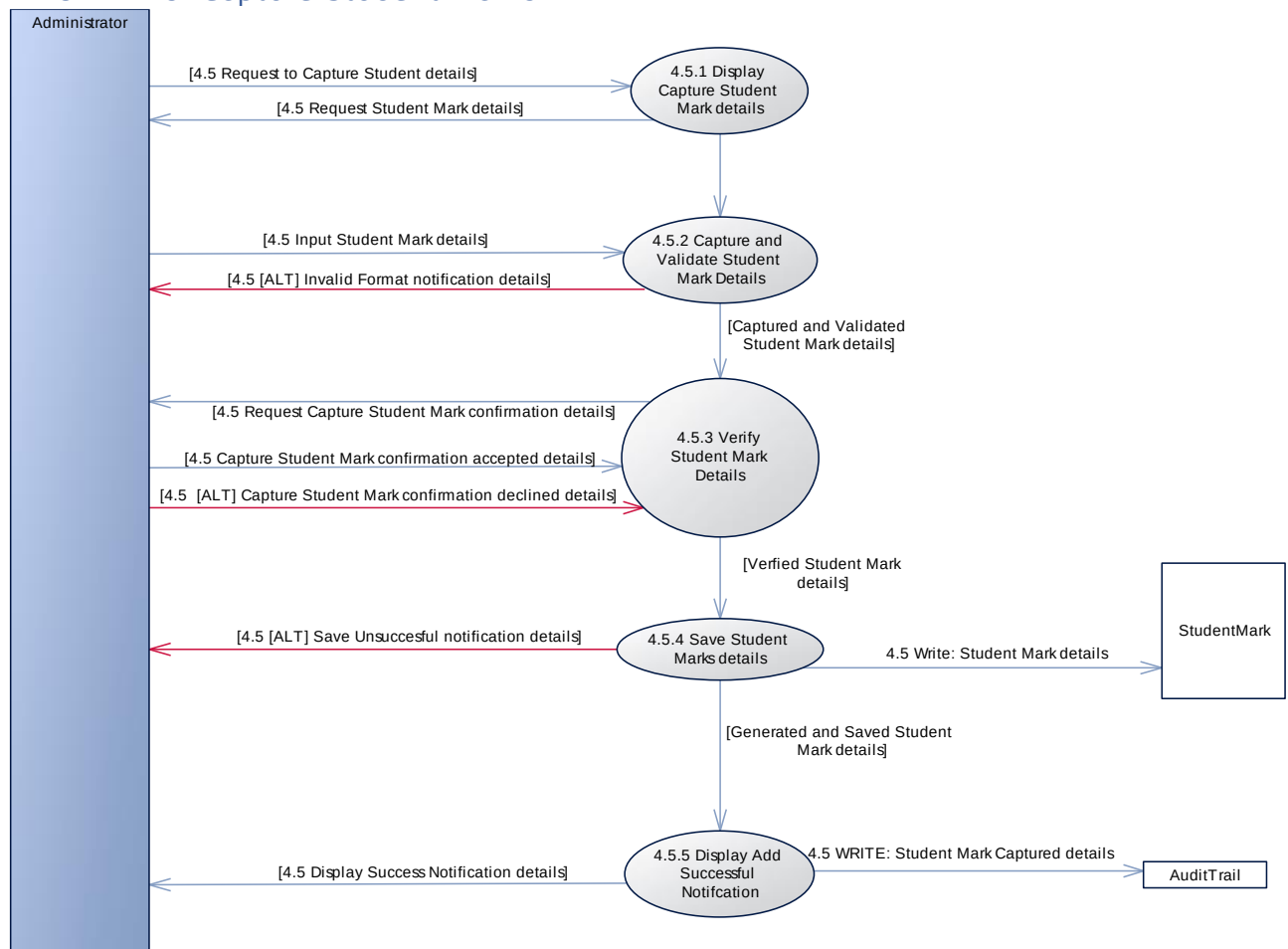


Figure 151 Add Student Mark Primitive Diagram



3.4.4.4.6 Search Student Marks

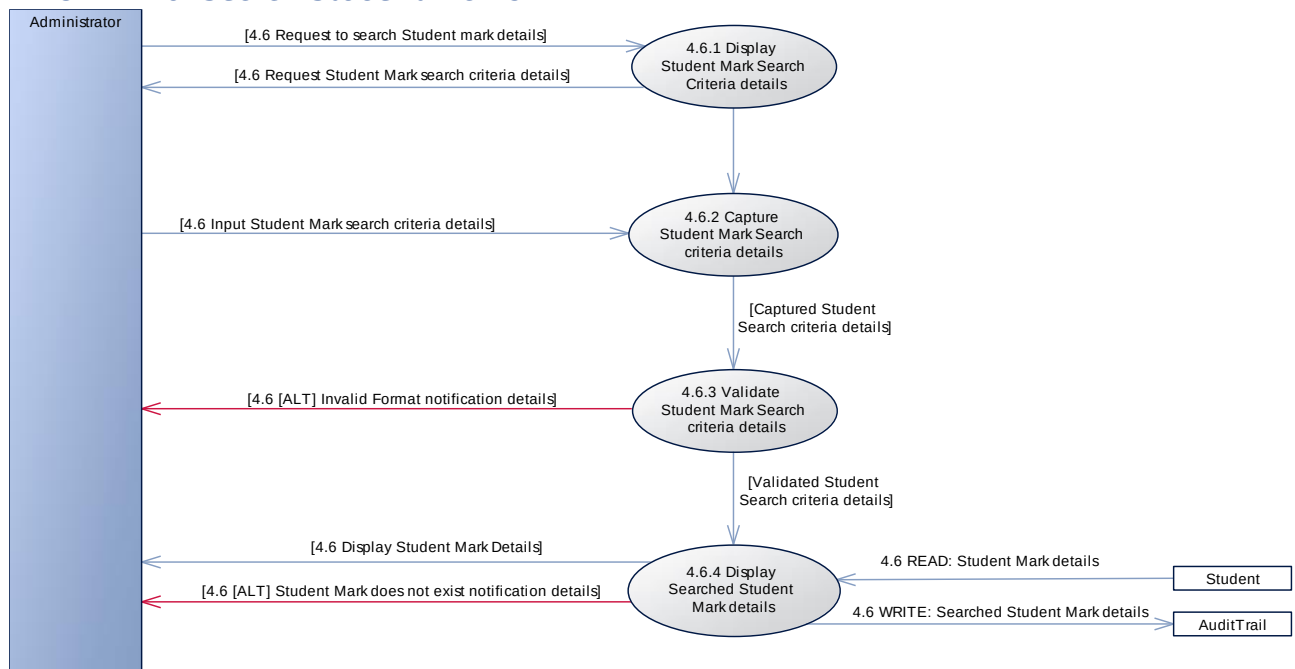


Figure 152 Search Student Primitive Diagram



3.4.4.4.7 Add Mark Type

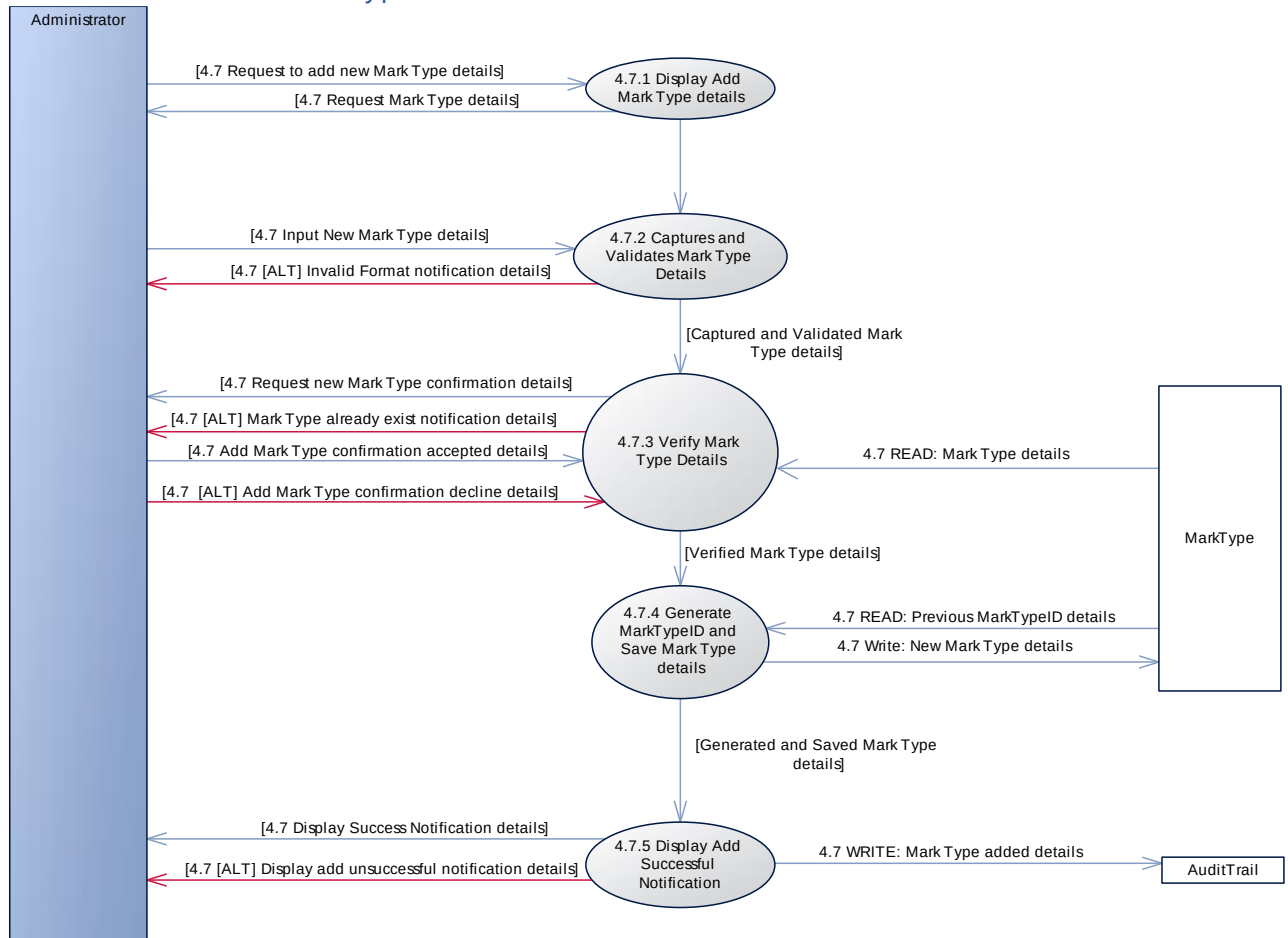


Figure 153 Add Mark Type Primitive Diagram



3.4.4.4.8 Search Mark Type

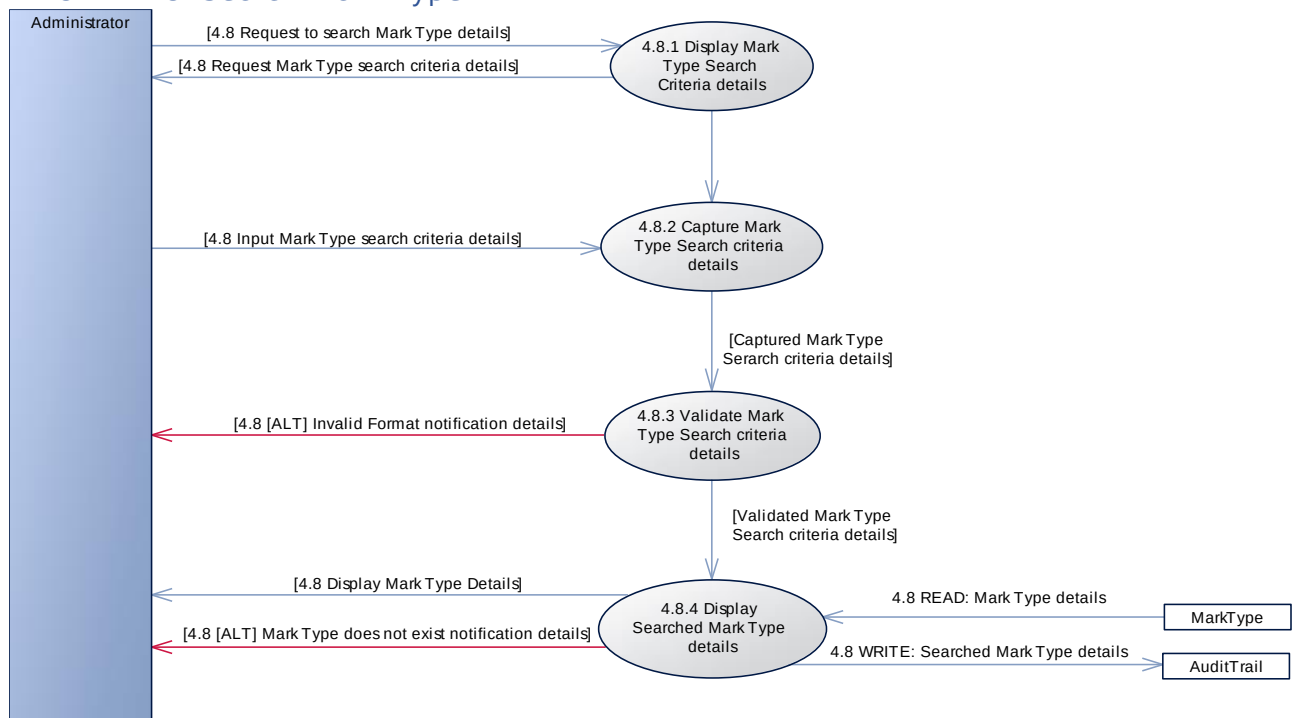


Figure 154 Search Mark Type Primitive Diagram



3.4.4.4.9 Update Mark Type

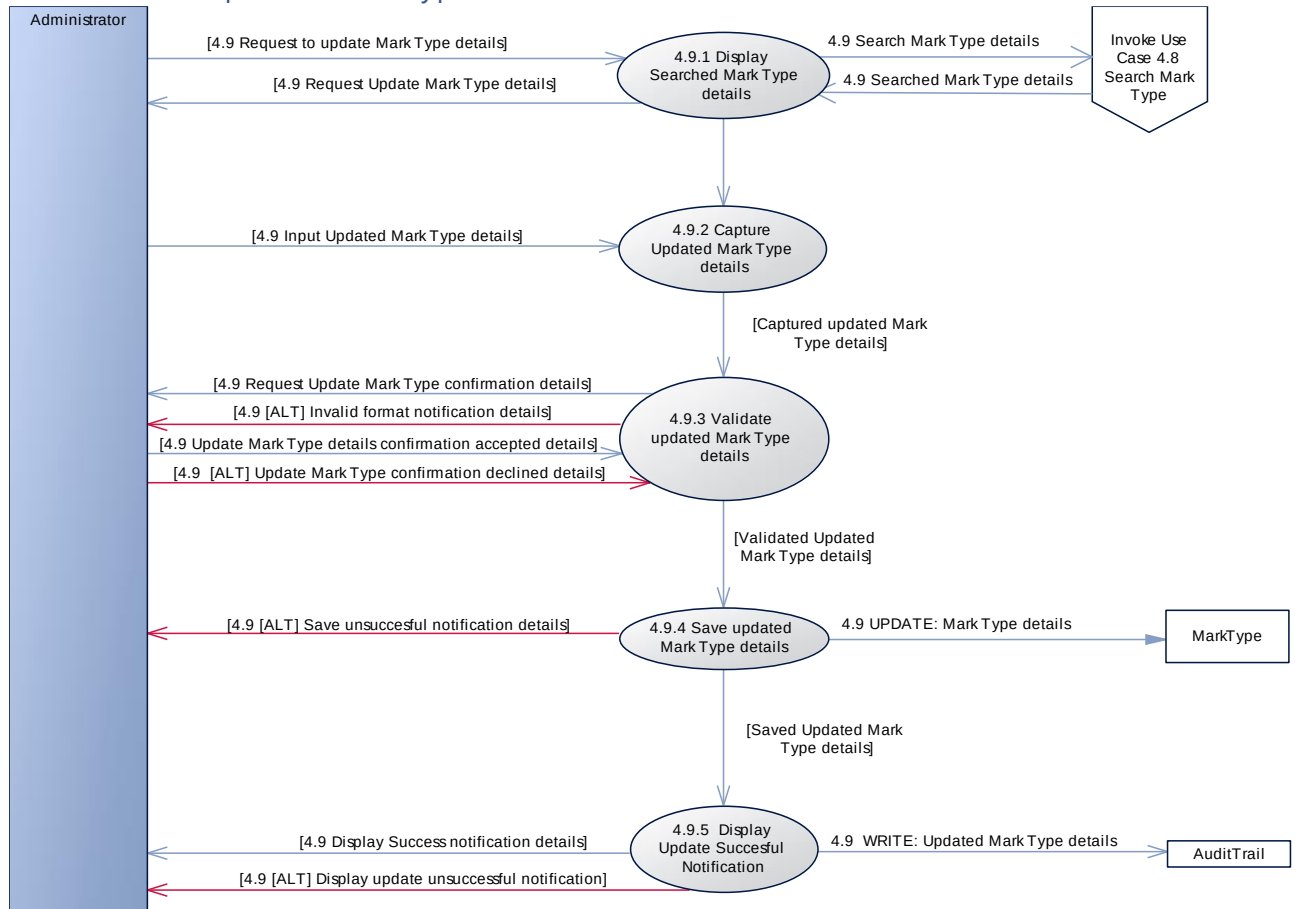


Figure 155 Update Mark Type Primitive Diagram



3.4.4.4.10 Delete Mark Type

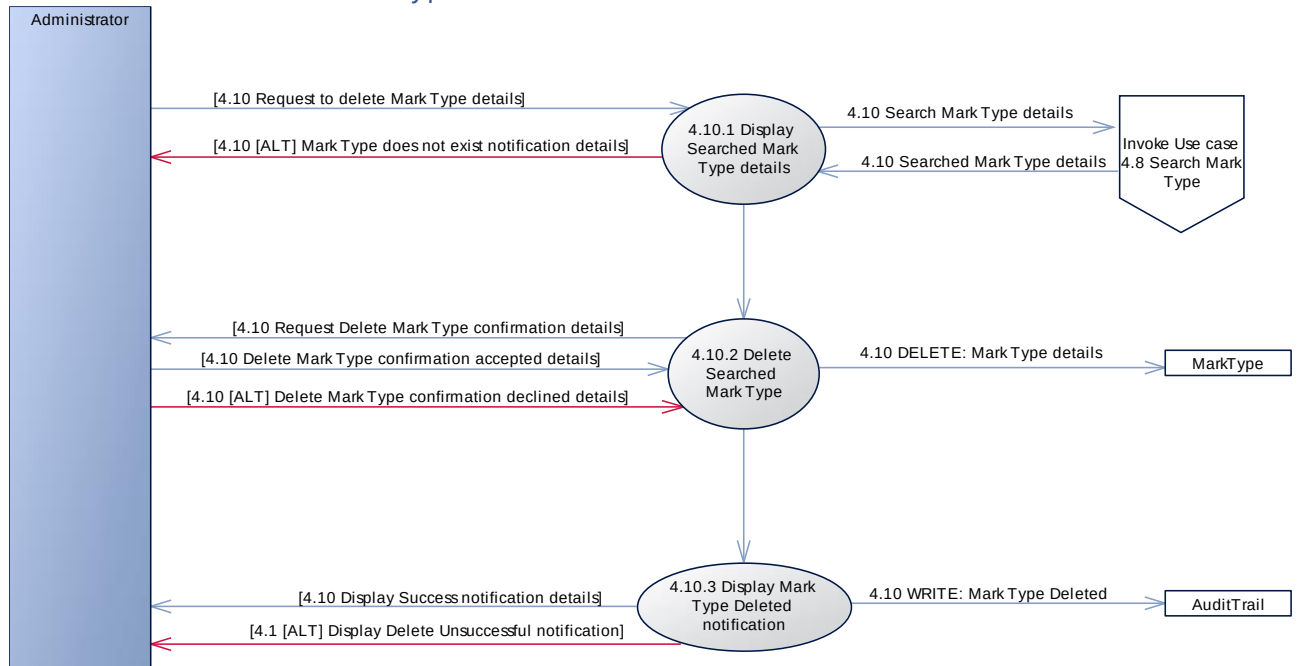


Figure 156 Delete Mark Type Primitive Diagram



3.4.4.5 Sub-system 5: Bicycle

3.4.4.5.1 Add Bicycle

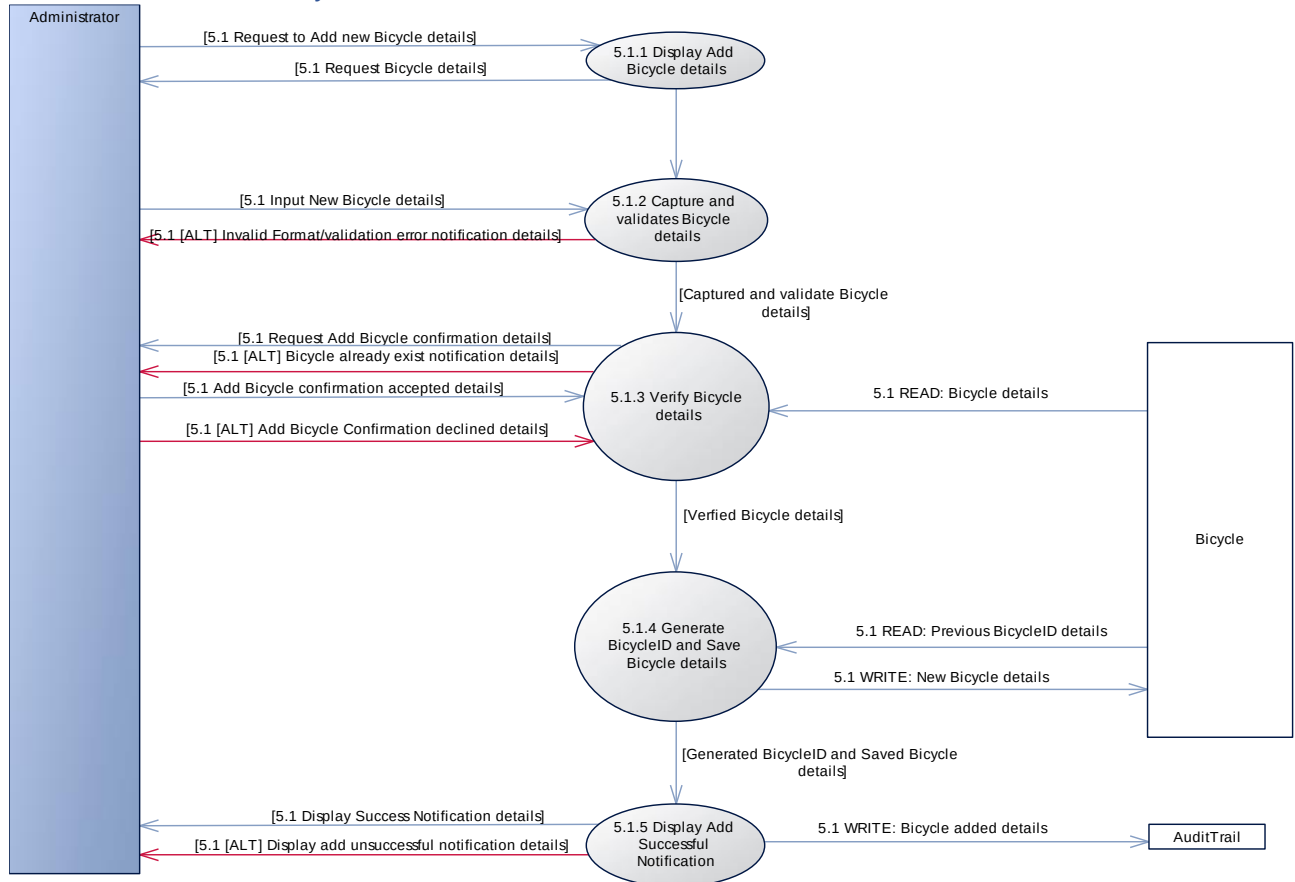


Figure 157 Add Bicycle Primitive Diagram



3.4.4.5.2 Search Bicycle

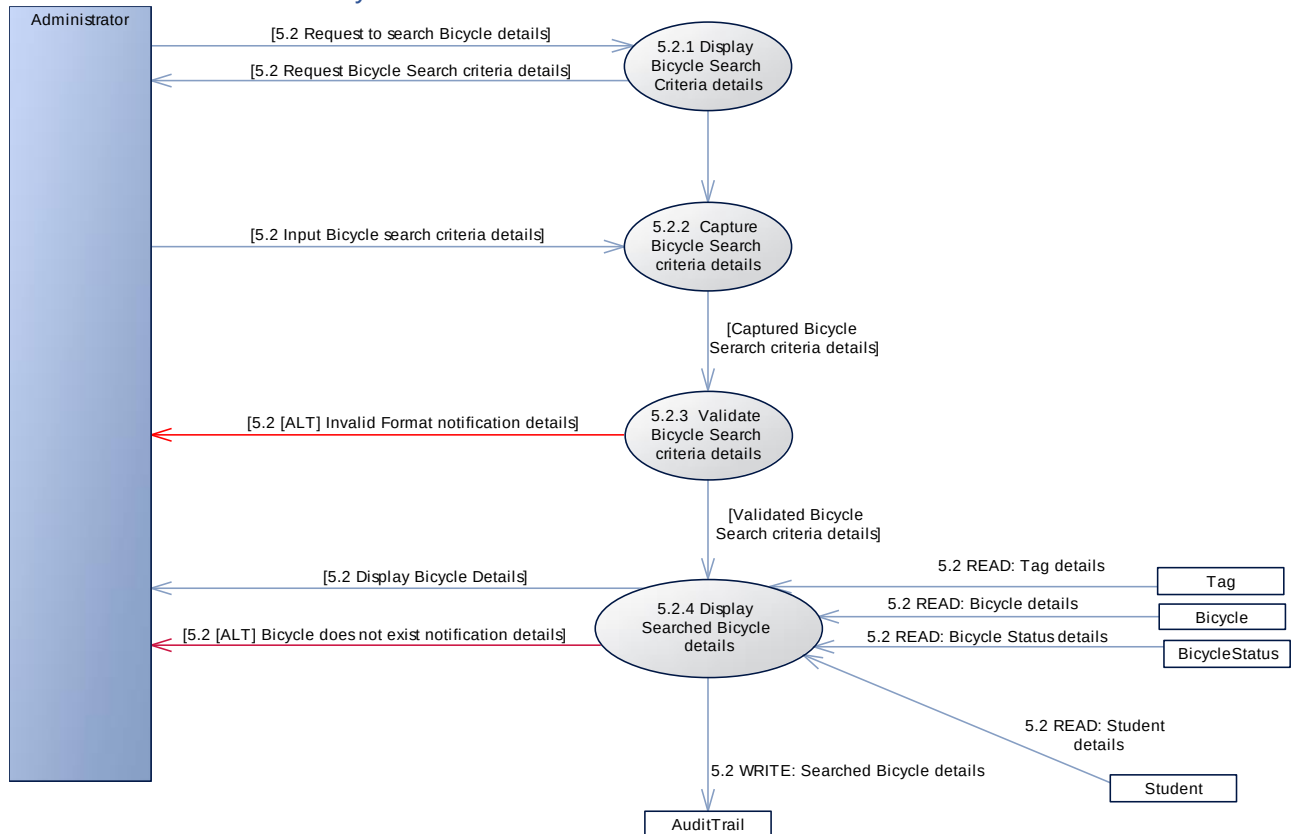


Figure 158 Search Bicycle Primitive Diagram



3.4.4.5.3 Update Bicycle

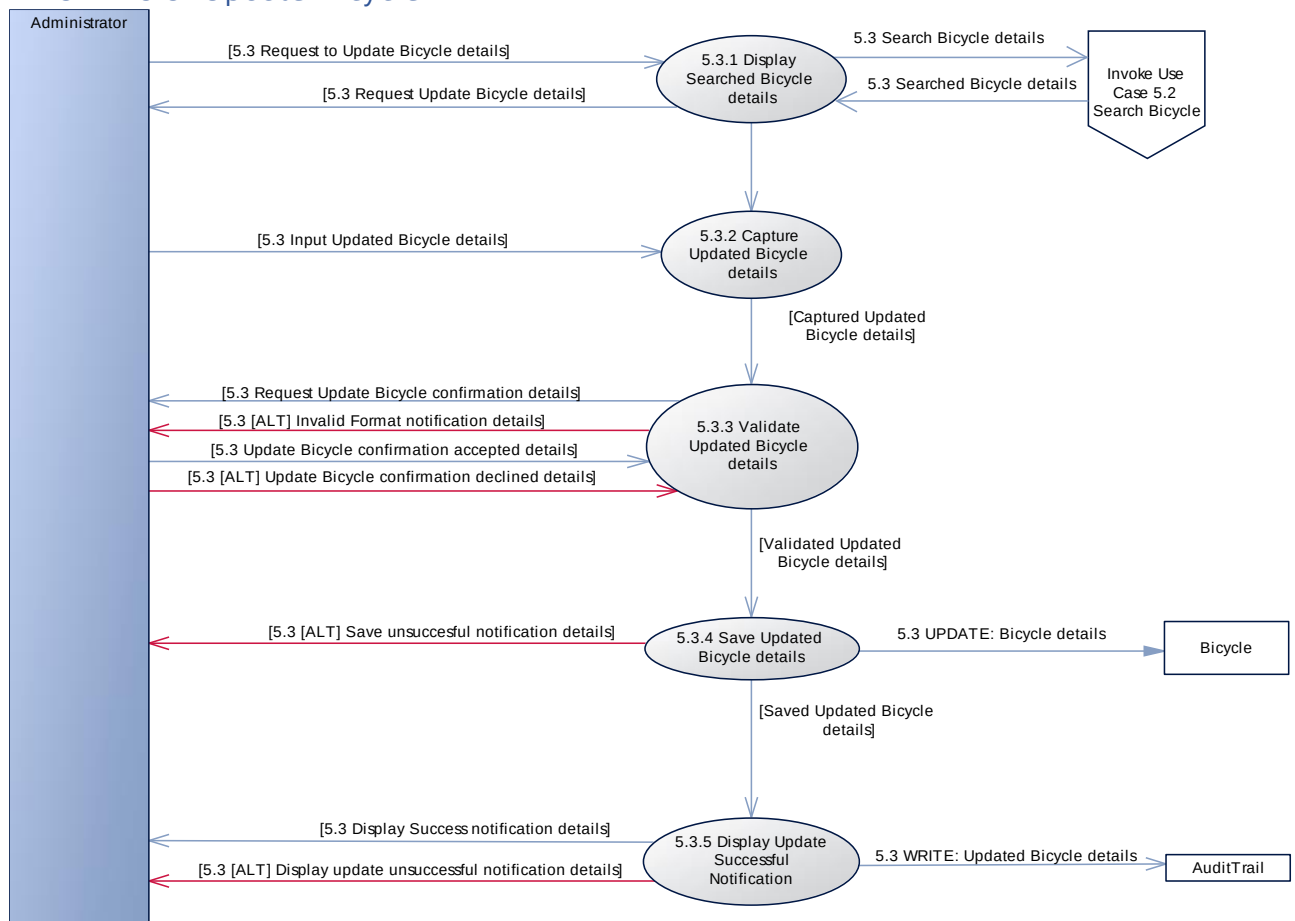


Figure 159 Update Bicycle Primitive Diagram



3.4.4.5.4 Delete Bicycle

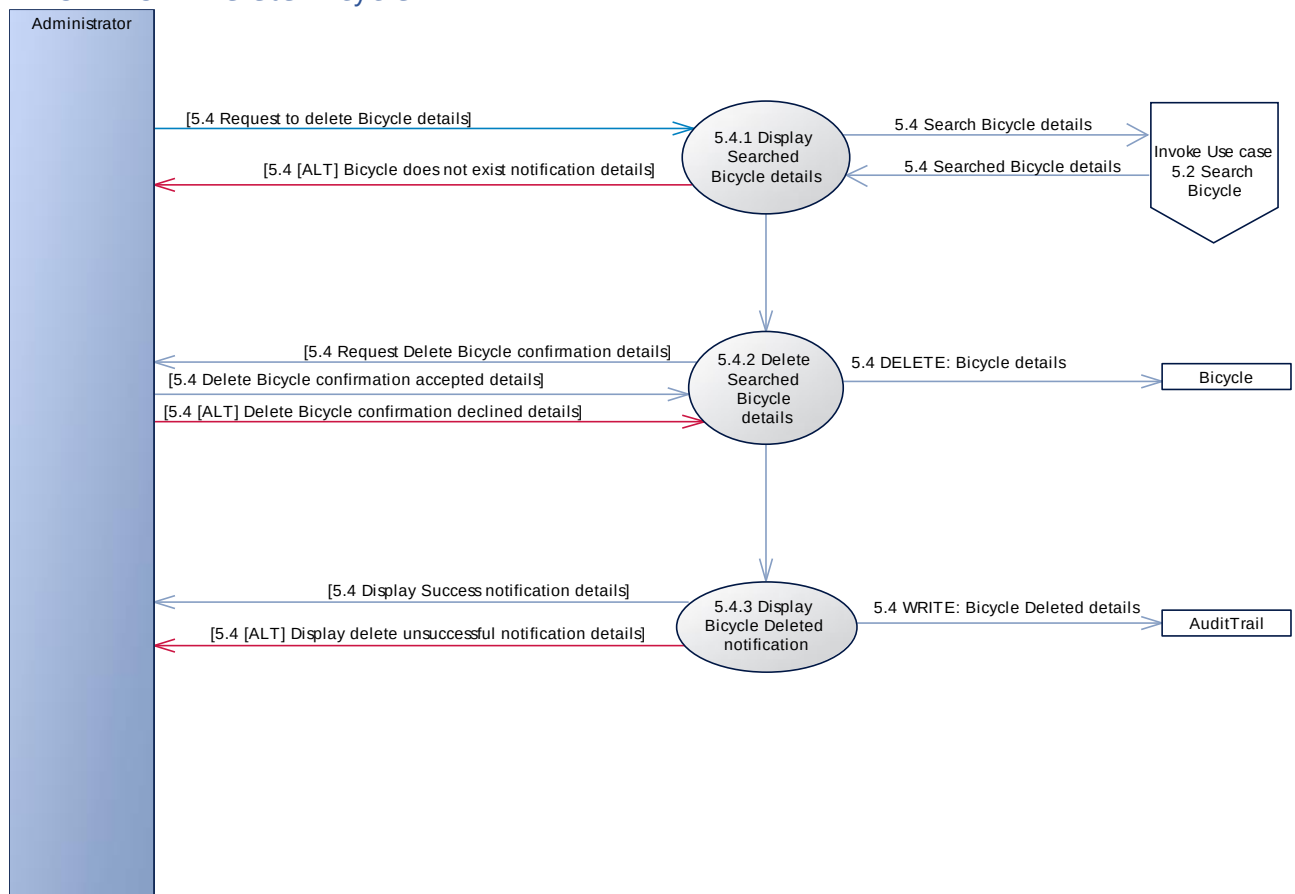


Figure 160 Delete Bicycle Primitive Diagram

3.4.4.5.5 Assign Bicycle to Student

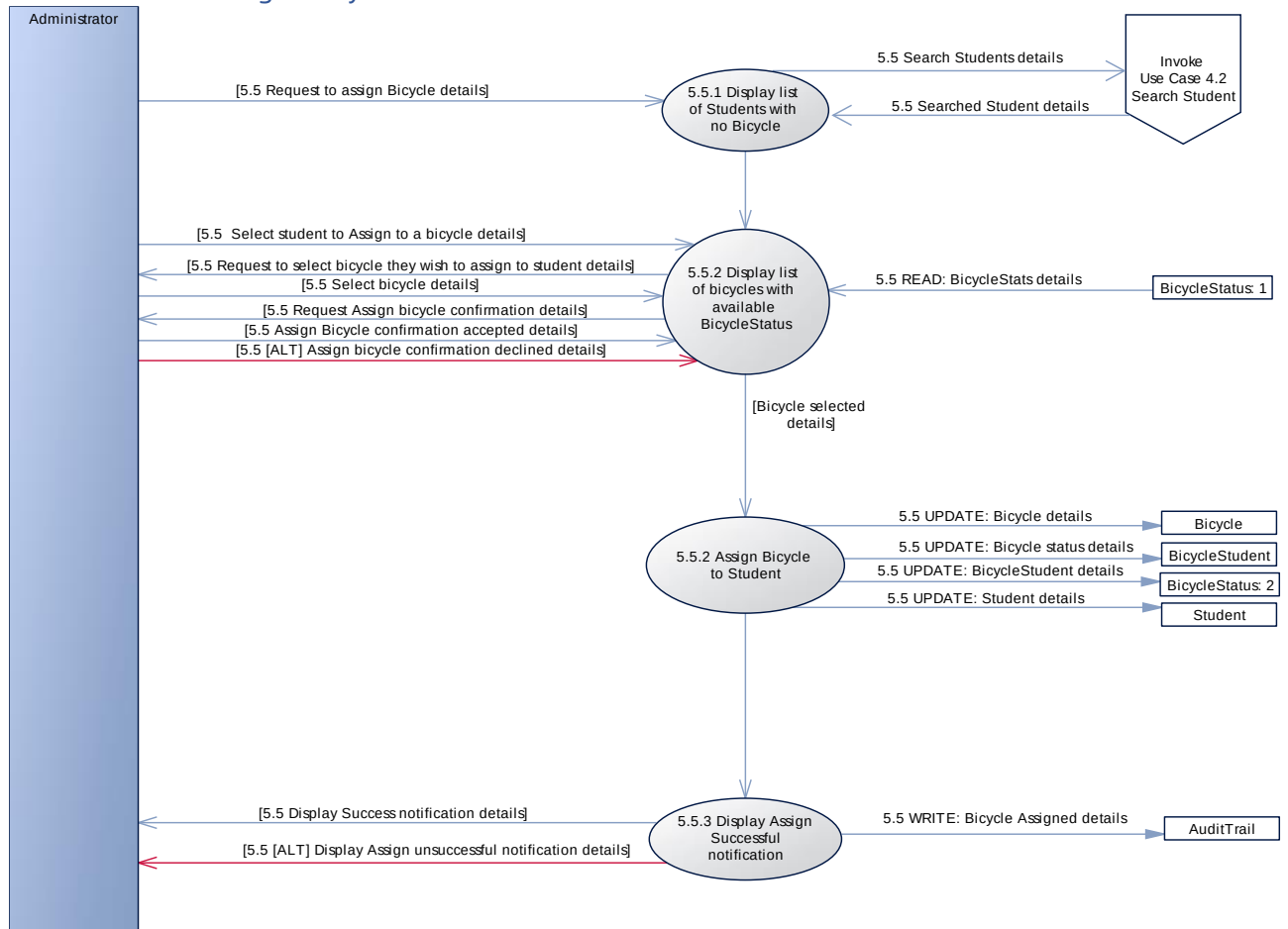


Figure 161 Assign Bicycle to Student Primitive Diagram



3.4.4.5.6 Unassign Bicycle from Student

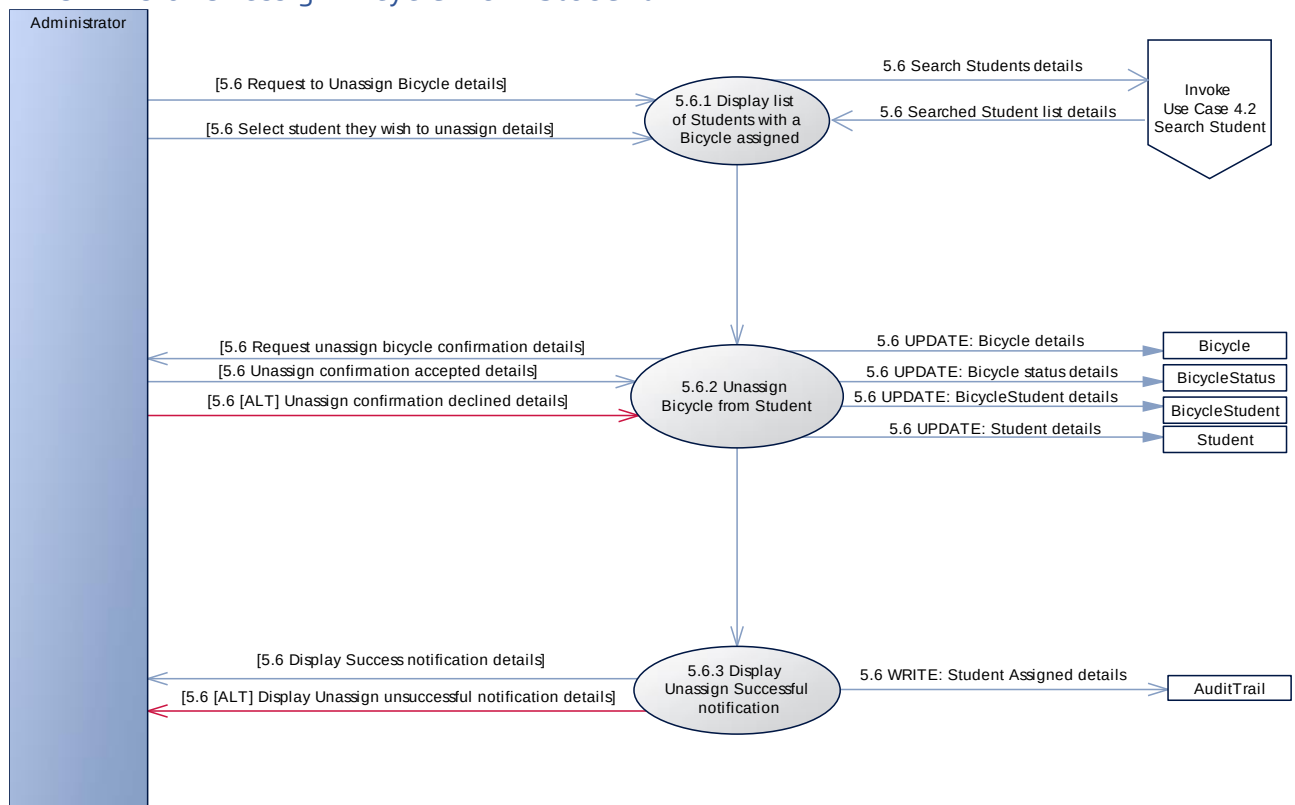


Figure 162 Unassign Bicycle from Student Primitive Diagram



3.4.4.5.7 Assign Tag to Bicycle

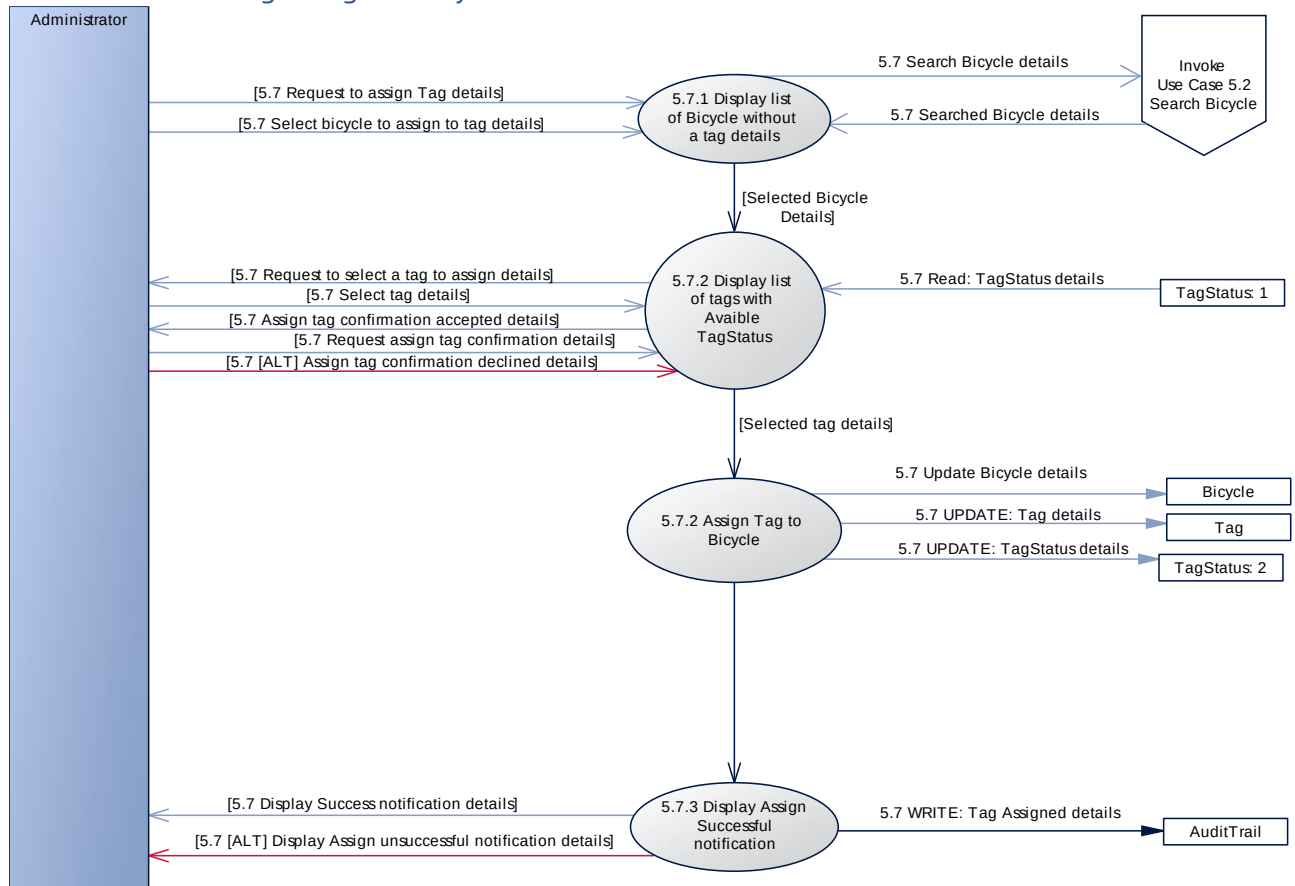


Figure 163 Assign Tag to Bicycle Primitive Diagram



3.4.4.5.8 Unassign Tag from Bicycle

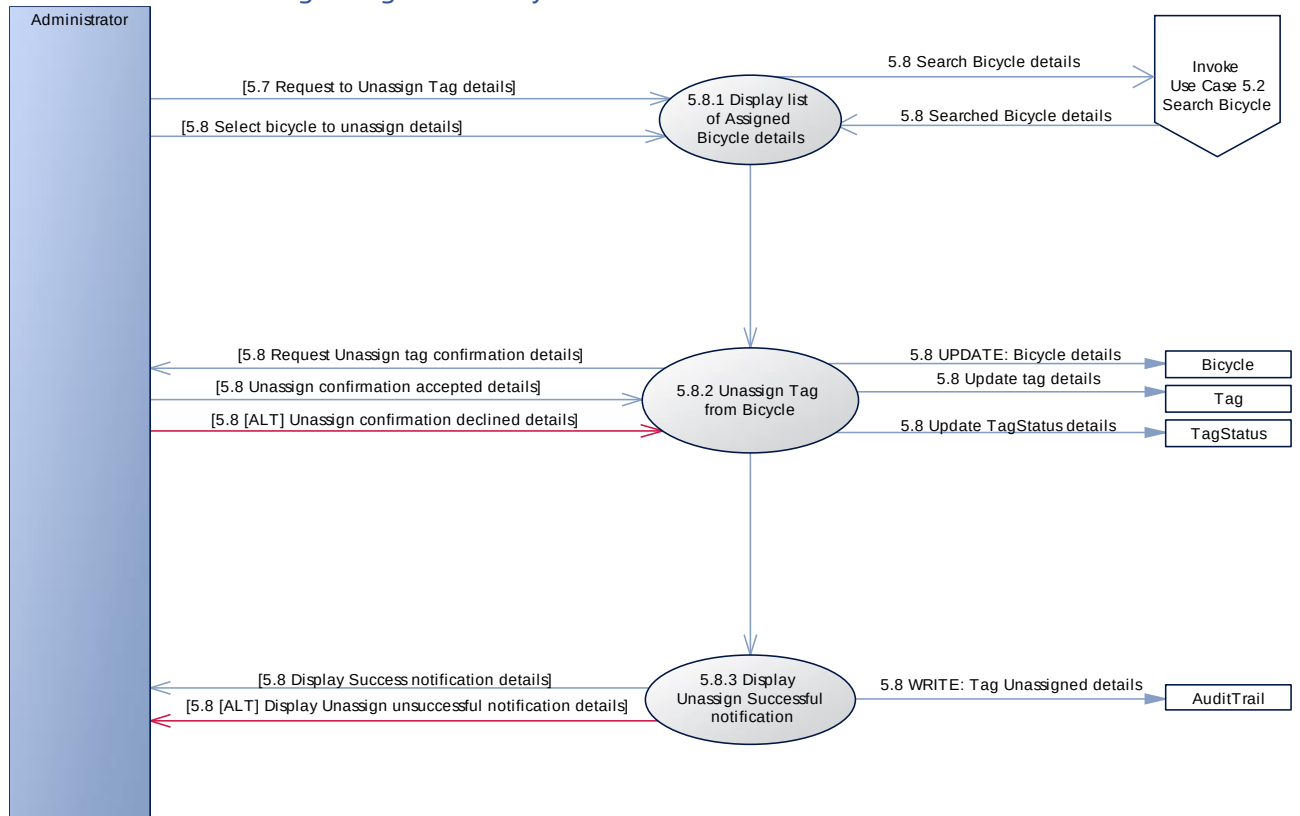


Figure 164 Unassign Tag from Bicycle Primitive Diagram



3.4.4.6 Sub-system 6: Bicycle Part

3.4.4.6.1 Add Bicycle Part

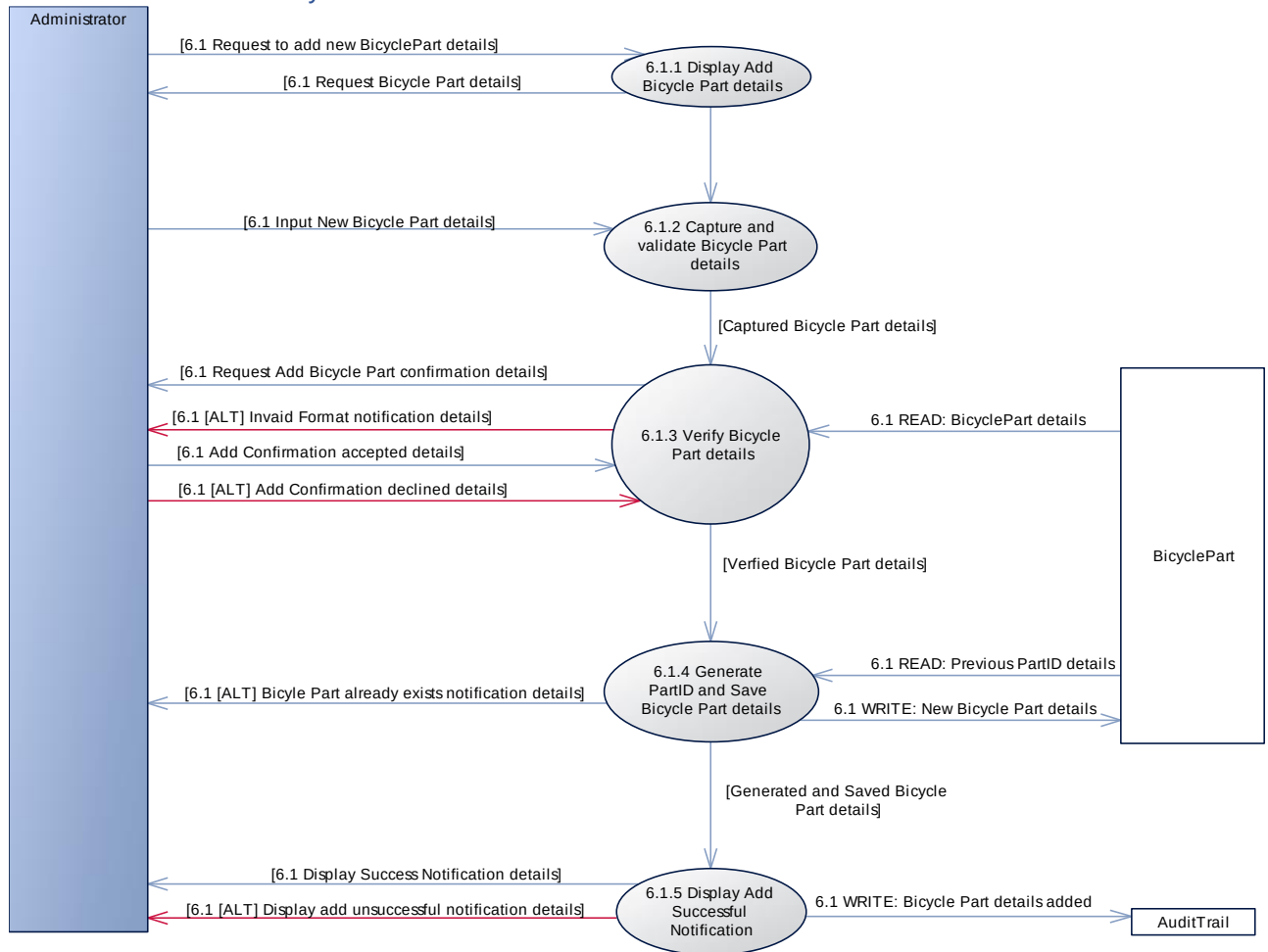


Figure 165 Add Bicycle Part Primitive Diagram



3.4.4.6.2 Search Bicycle Part

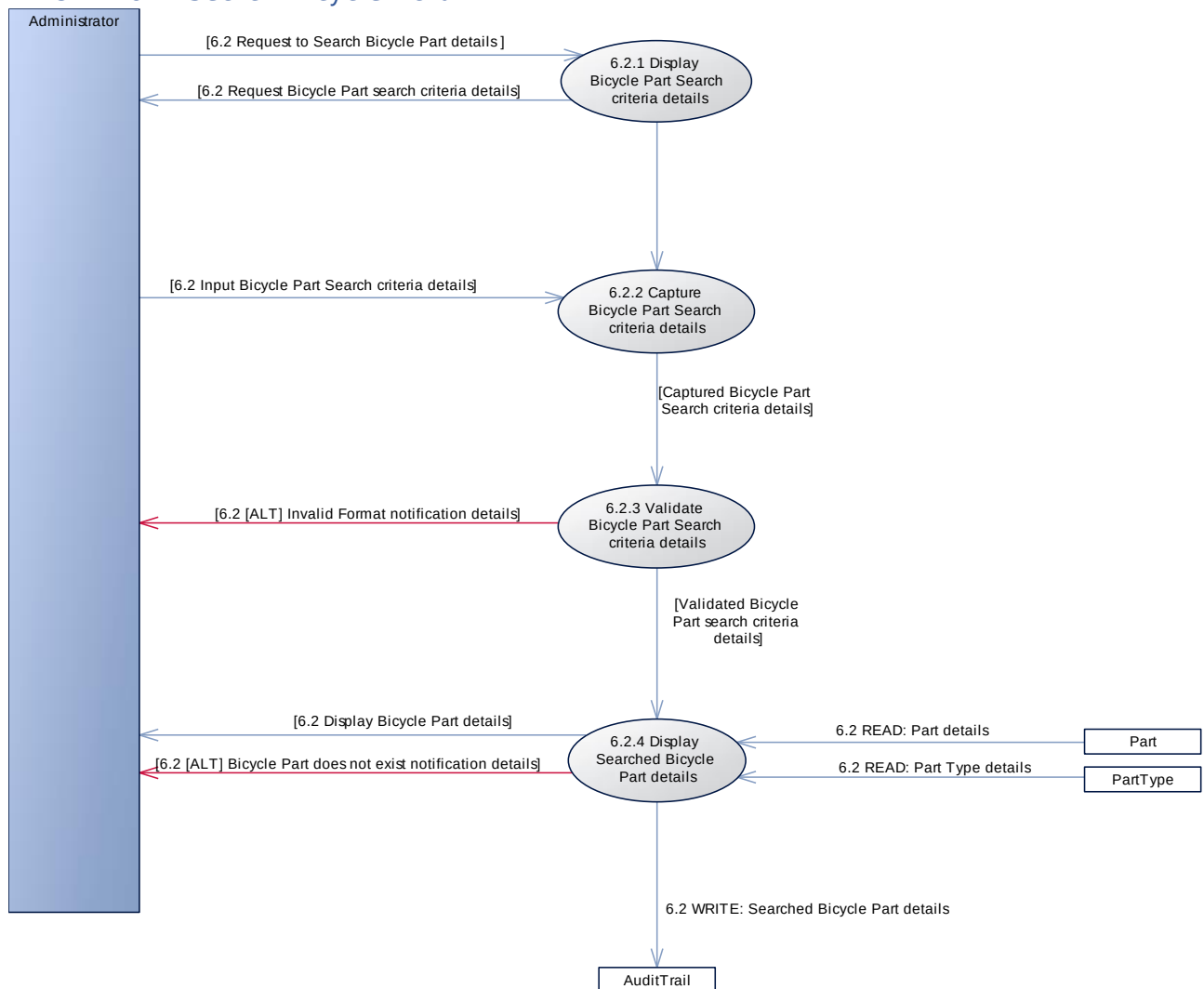


Figure 166 Search Bicycle Part Primitive Diagram



3.4.4.6.3 Update Bicycle Part

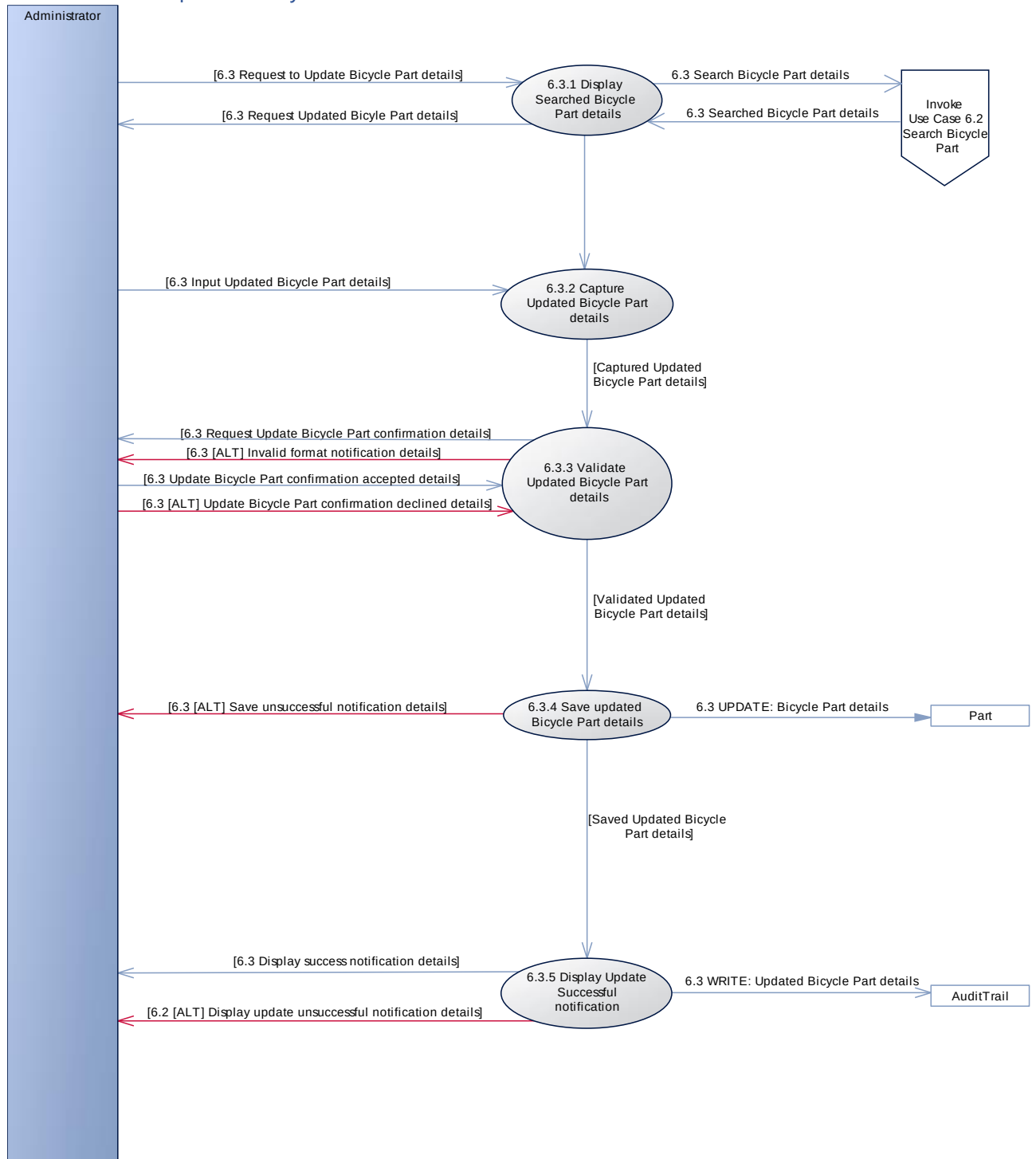


Figure 167 Update Bicycle Part Primitive Diagram



3.4.4.6.4 Delete Bicycle Part

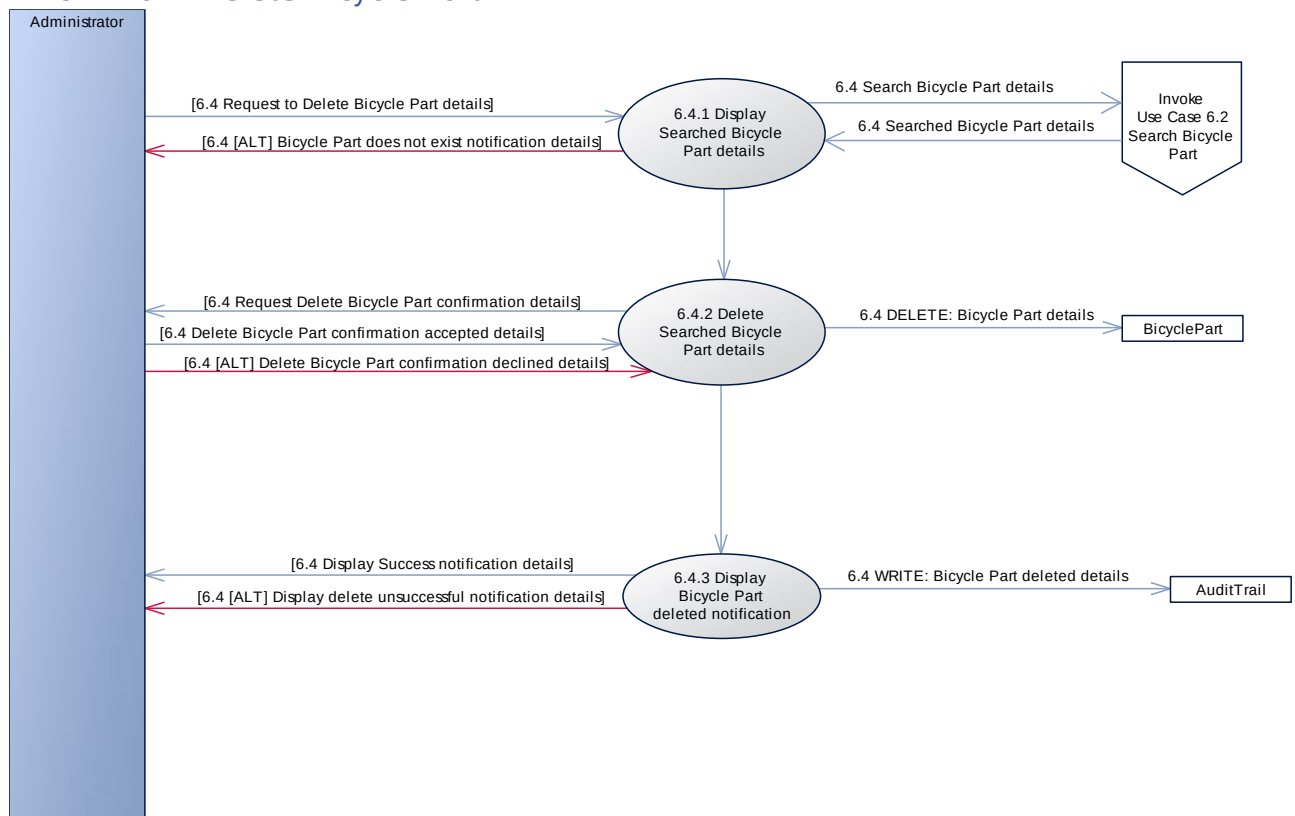


Figure 168 Delete Bicycle Part Primitive Diagram



3.4.4.6.5 Add Bicycle Section

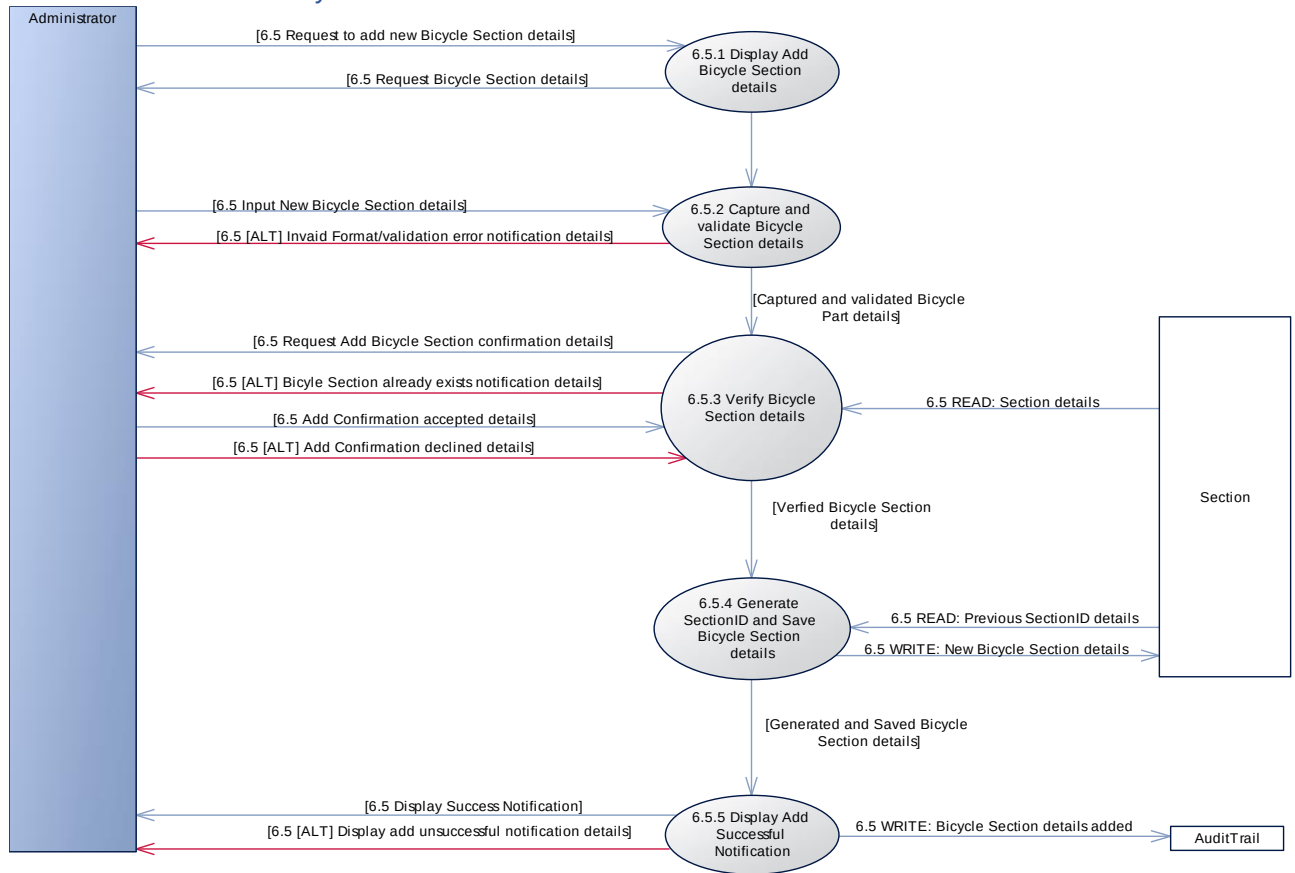


Figure 169 Add Bicycle Section Primitive Diagram



3.4.4.6.6 Search Bicycle Section

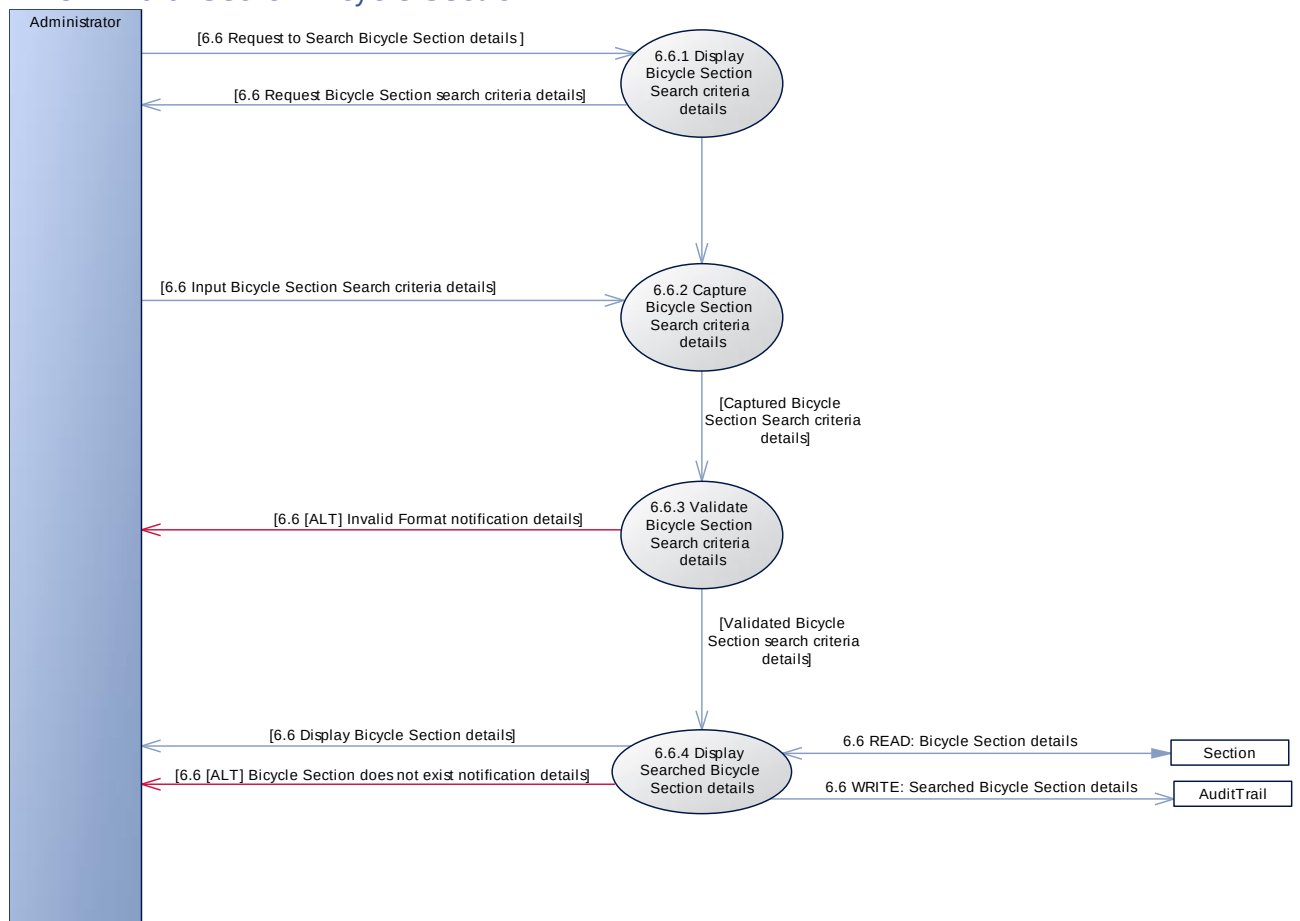


Figure 170 Search Bicycle Section Primitive Diagram



3.4.4.6.7 Update Bicycle Section

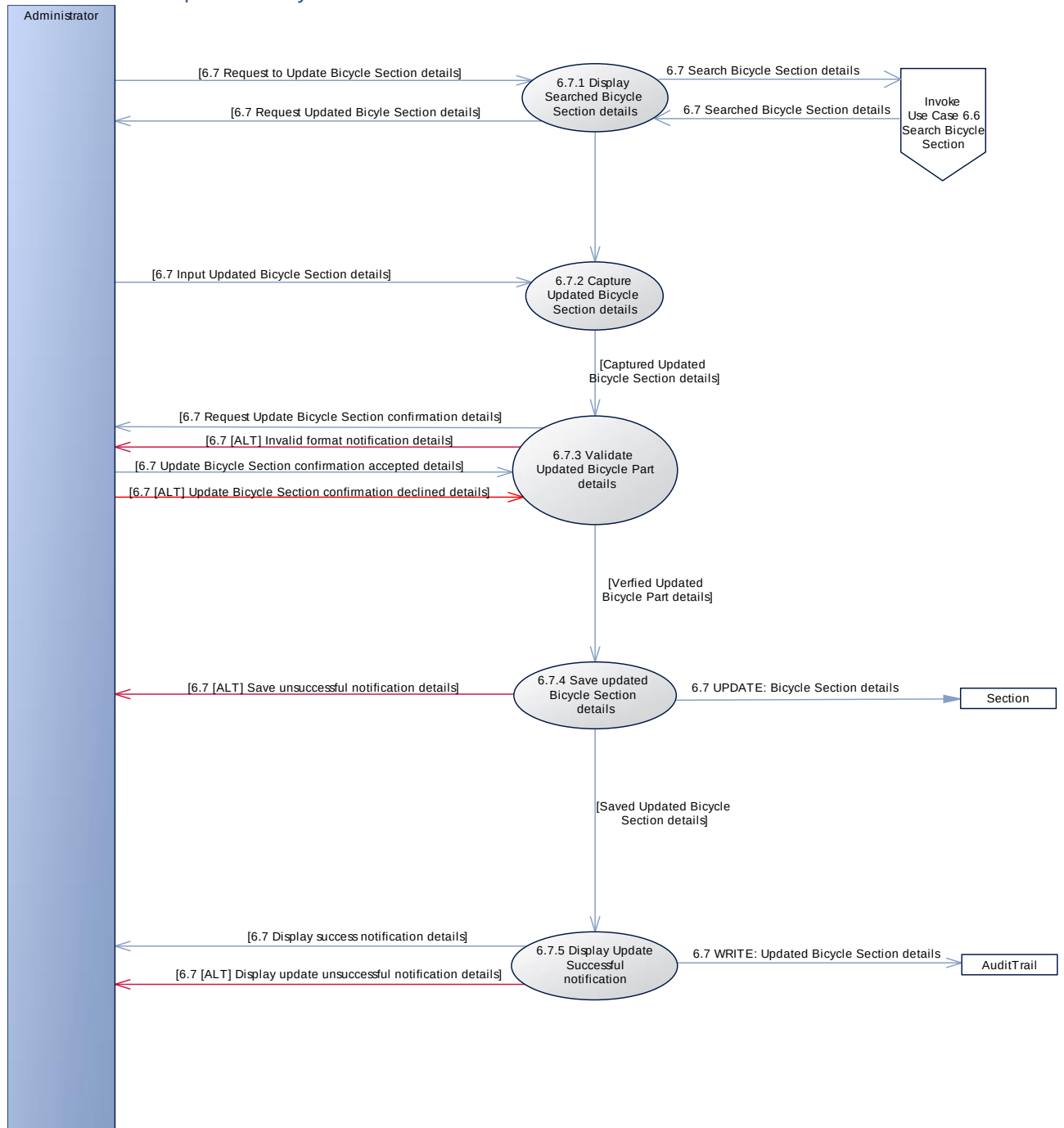


Figure 171 Update Bicycle Section Primitive Diagram



3.4.4.6.8 Delete Bicycle Section

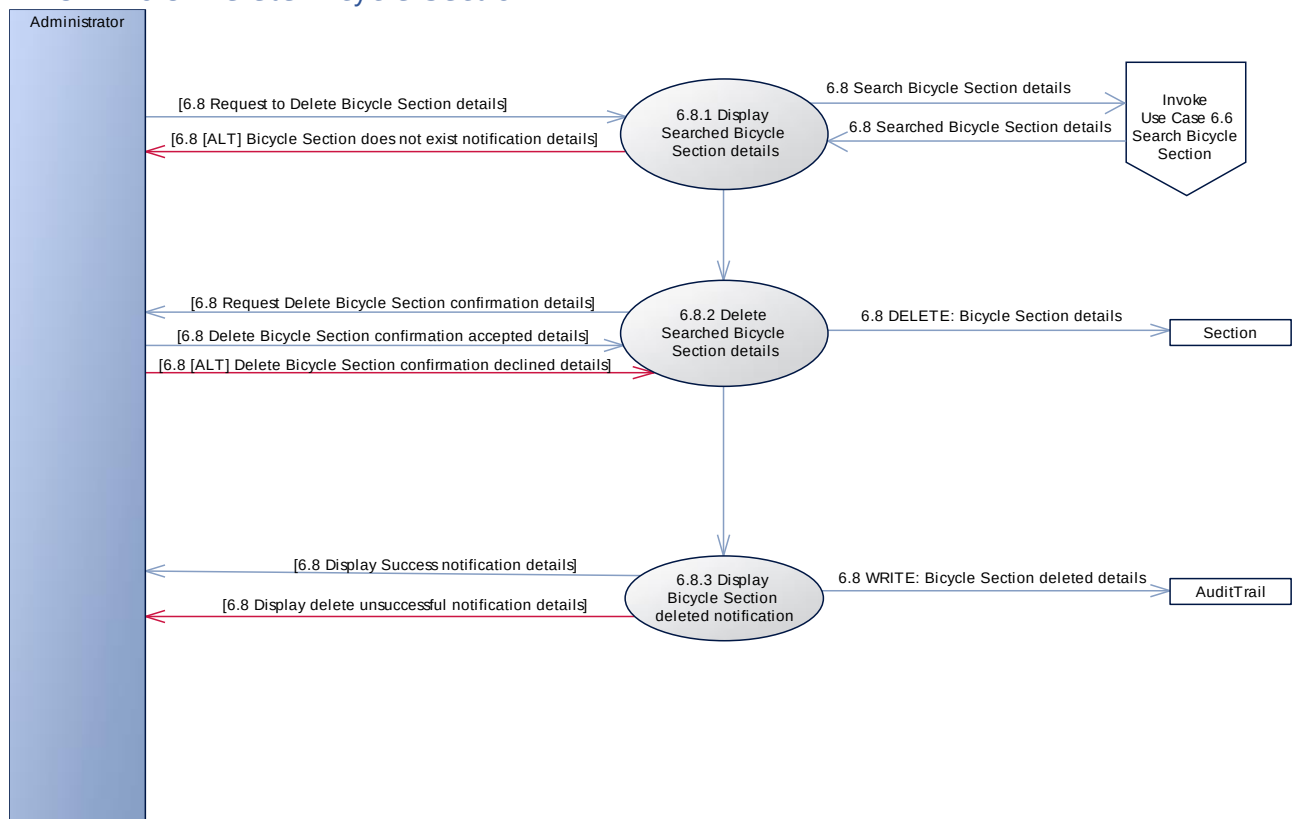


Figure 172 Delete Bicycle Section Primitive Diagram



3.4.4.7 Sub-system 7: Tracking

3.4.4.7.1 Add Log Entry

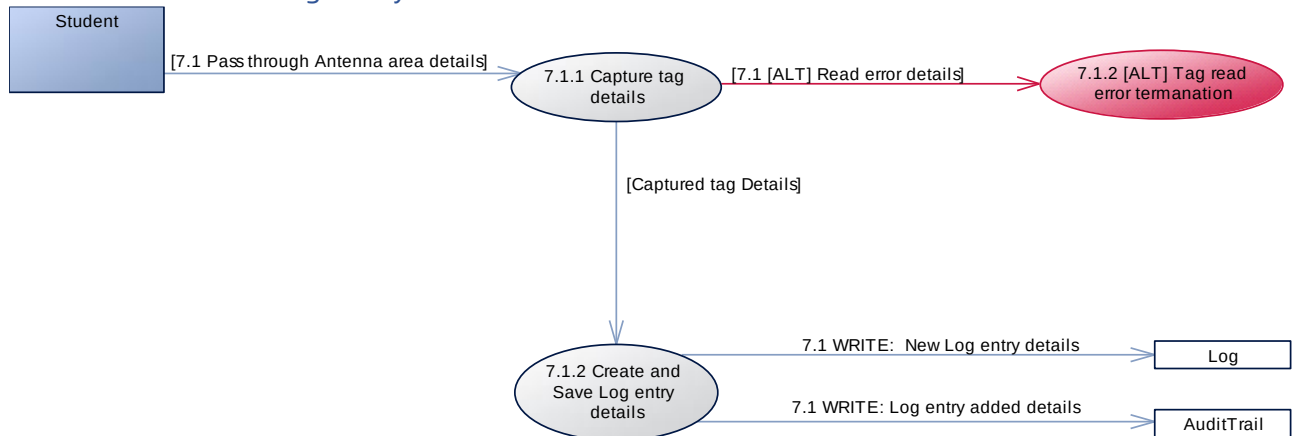


Figure 173 Add Log Entry Primitive Diagram



3.4.4.7.2 Search Log Entry

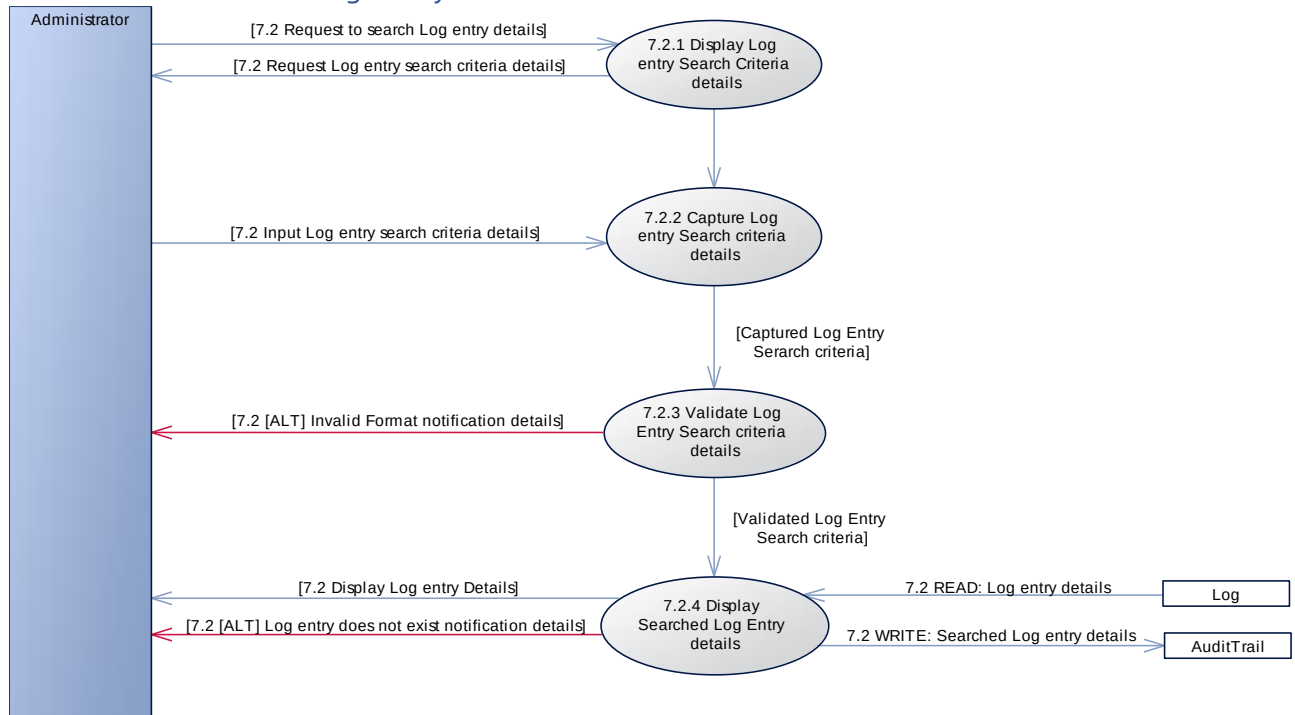


Figure 174 Search Log Entry Primitive Diagram



3.4.4.8 Sub-system 8: Job Tasks

3.4.4.8.1 Add Fault Task

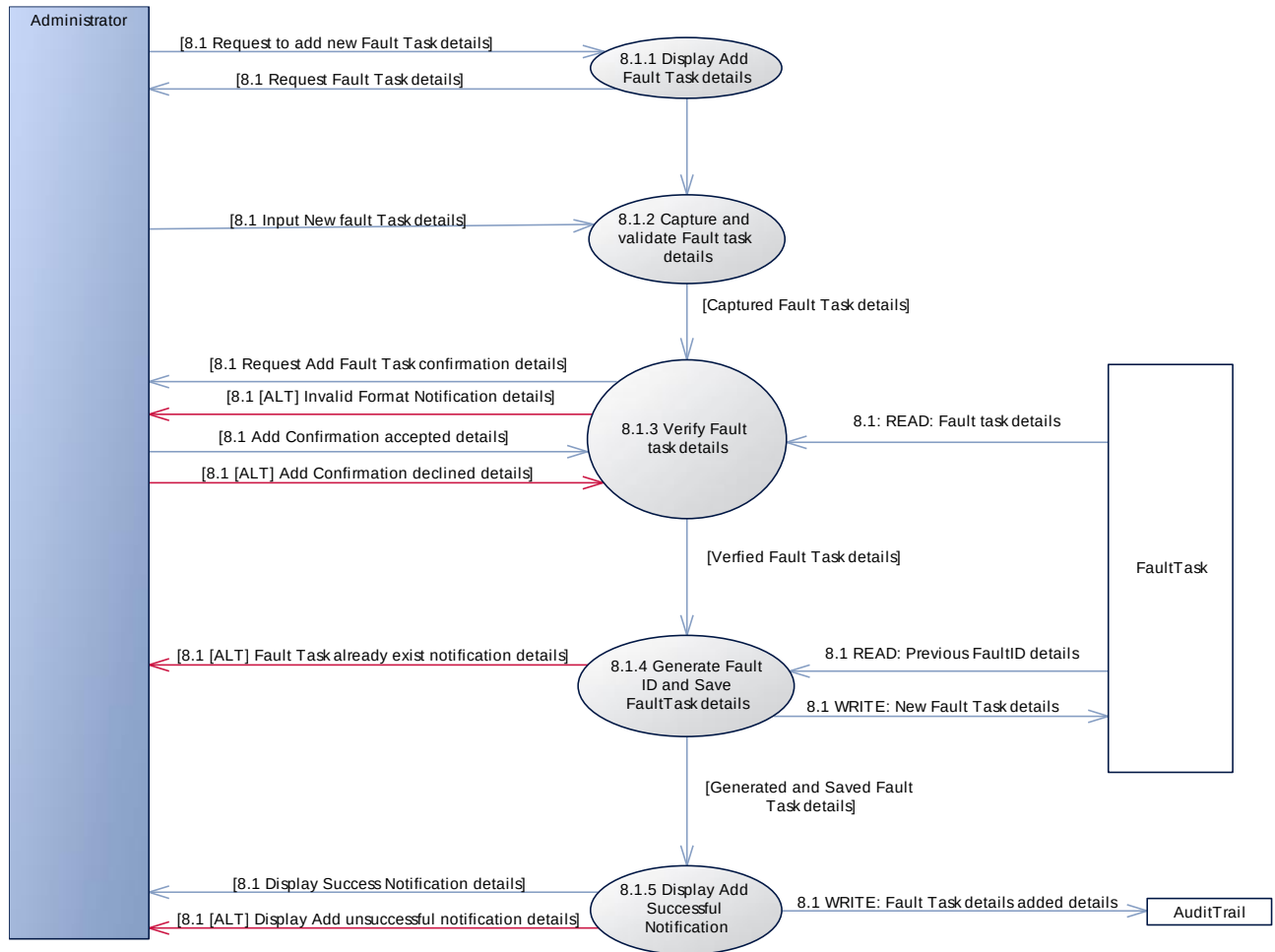


Figure 175 Add Fault Task Primitive Diagram



3.4.4.8.2 Search Fault Task

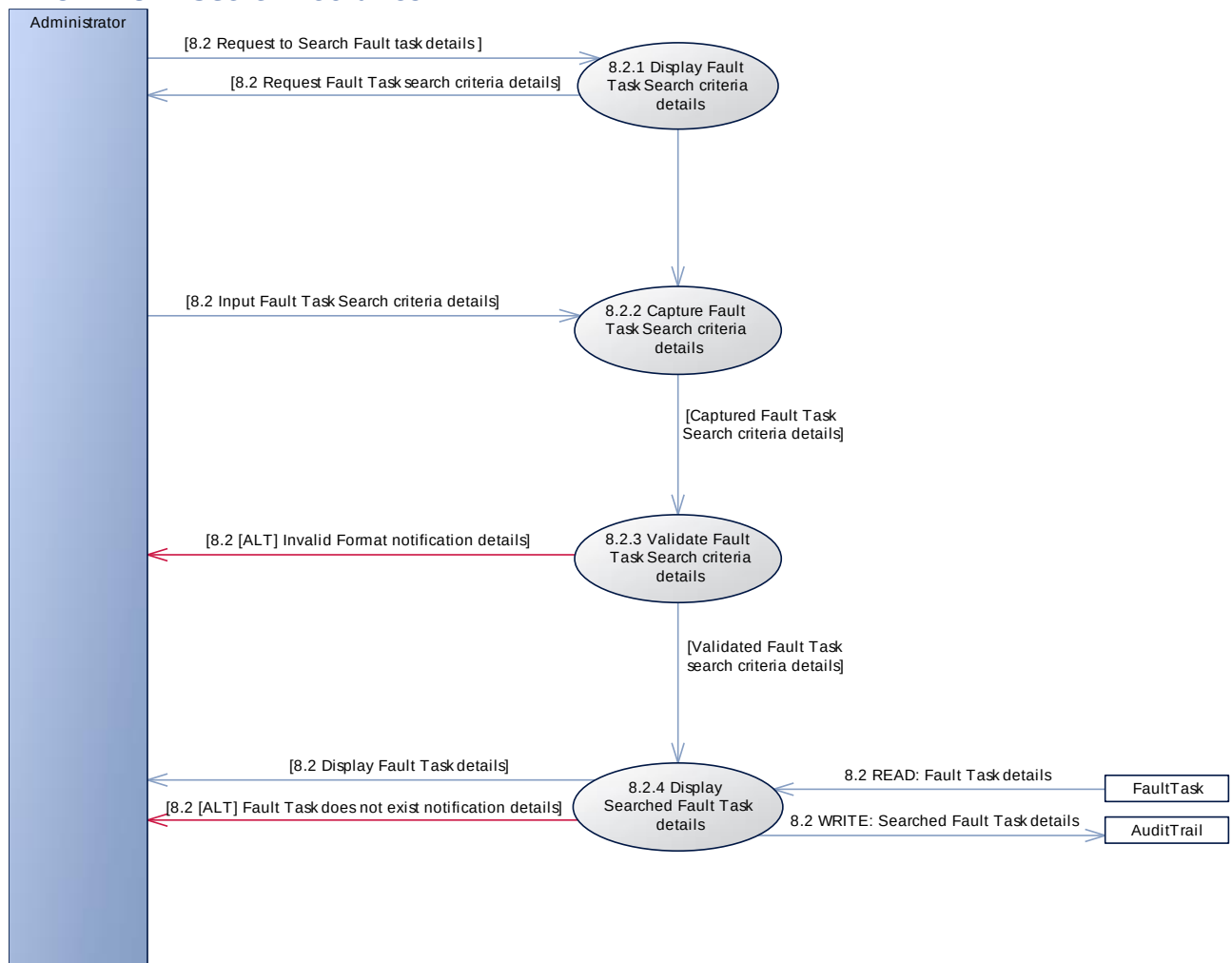


Figure 176 Search Fault Task Primitive Diagram



3.4.4.8.3 Update Fault Task

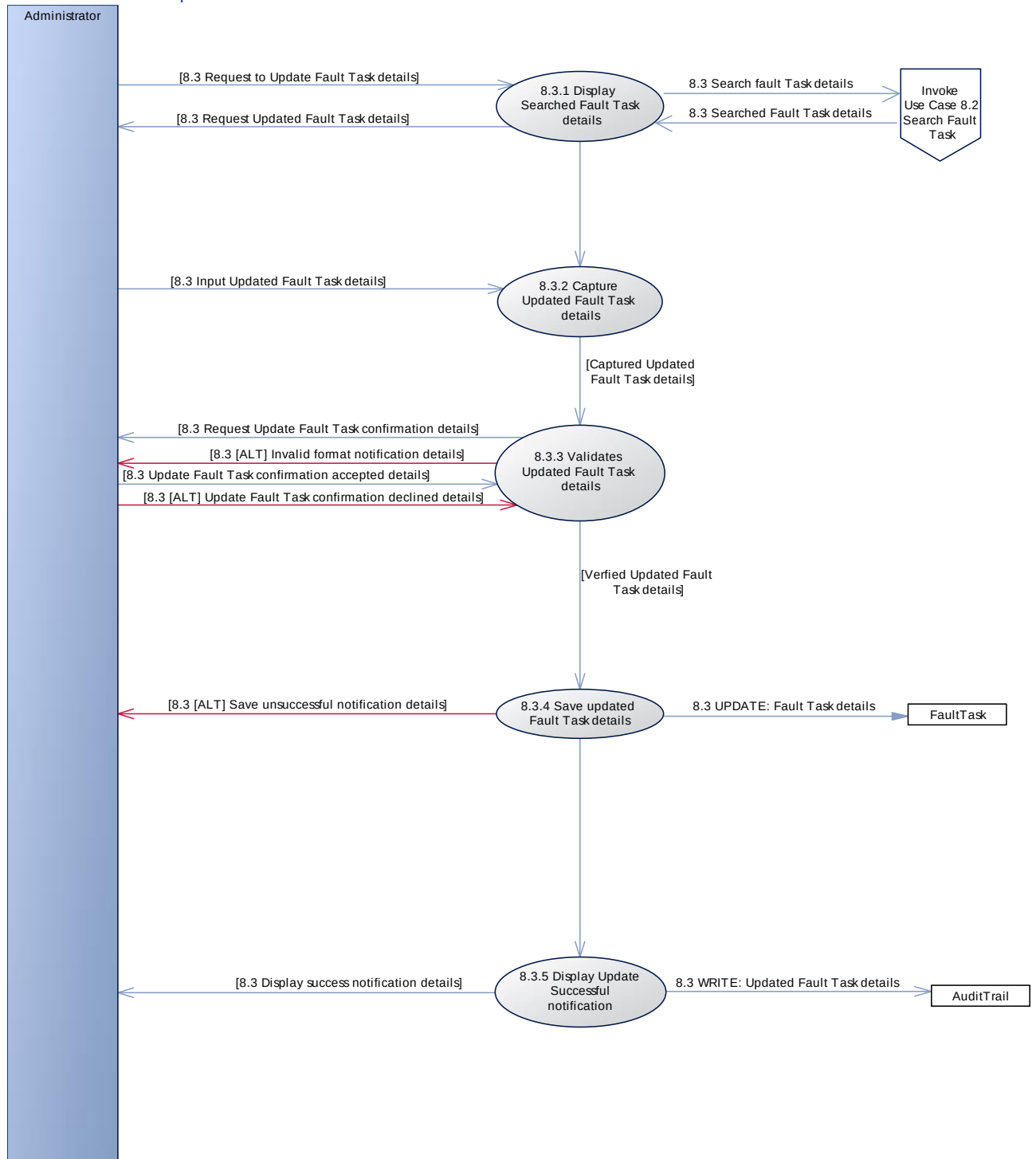


Figure 177 Update Fault Task Primitive Diagram



3.4.4.8.4 Delete Fault Task

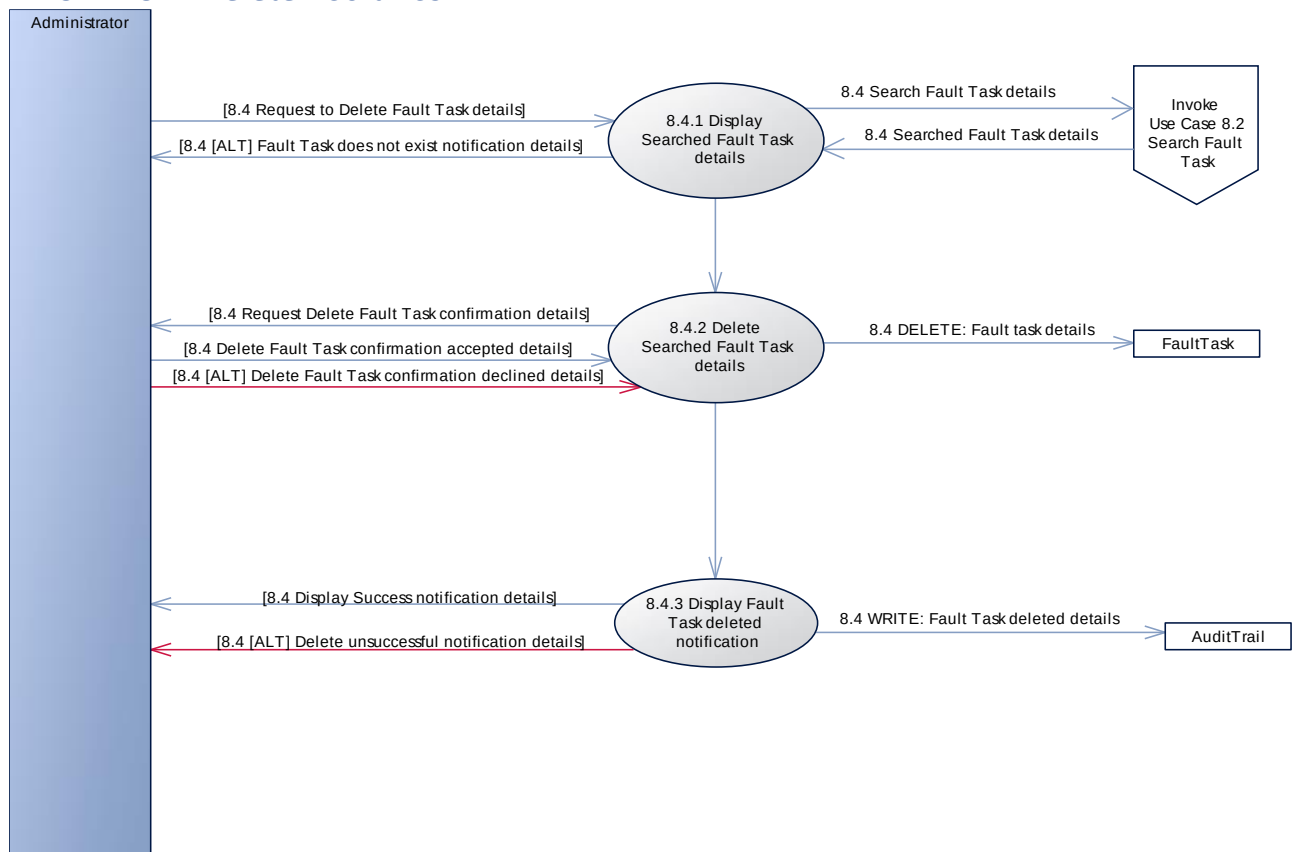


Figure 178 Delete Fault Task Primitive Diagram



3.4.4.8.5 Add Maintenance Task

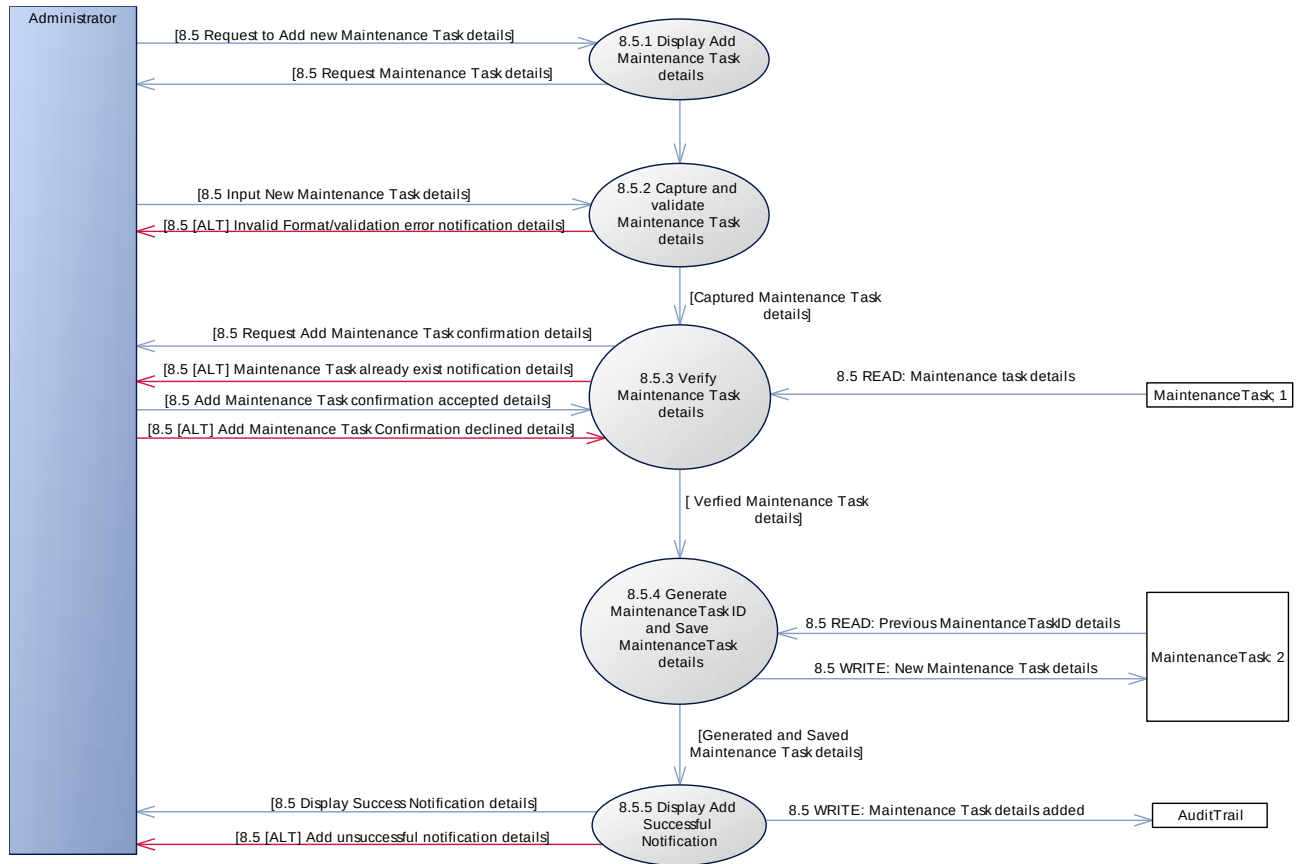


Figure 179 Add Maintenance Task Primitive Diagram



3.4.4.8.6 Search Maintenance Task

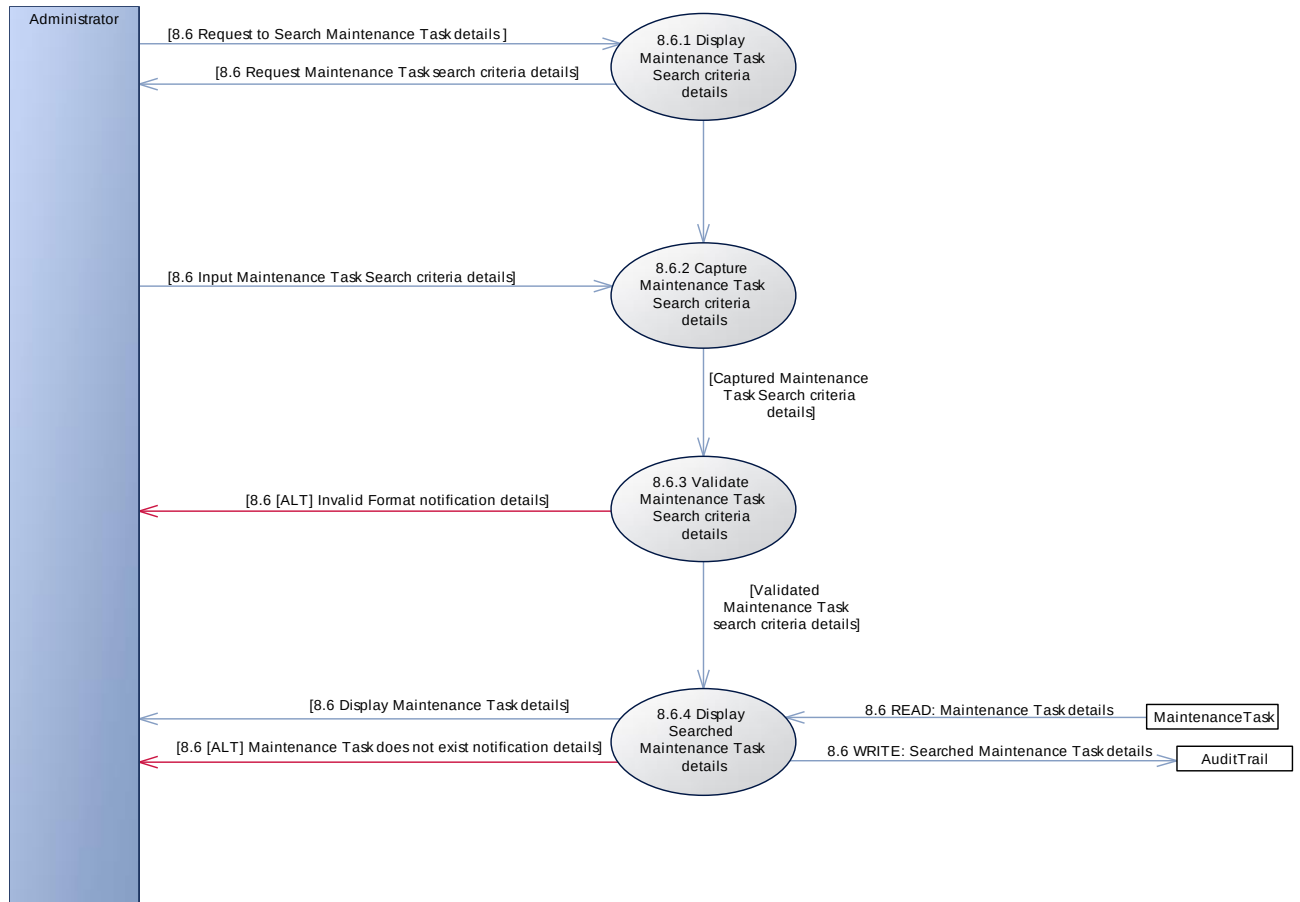


Figure 180 Search Maintenance Task Primitive Diagram



3.4.4.8.7 Update Maintenance Task

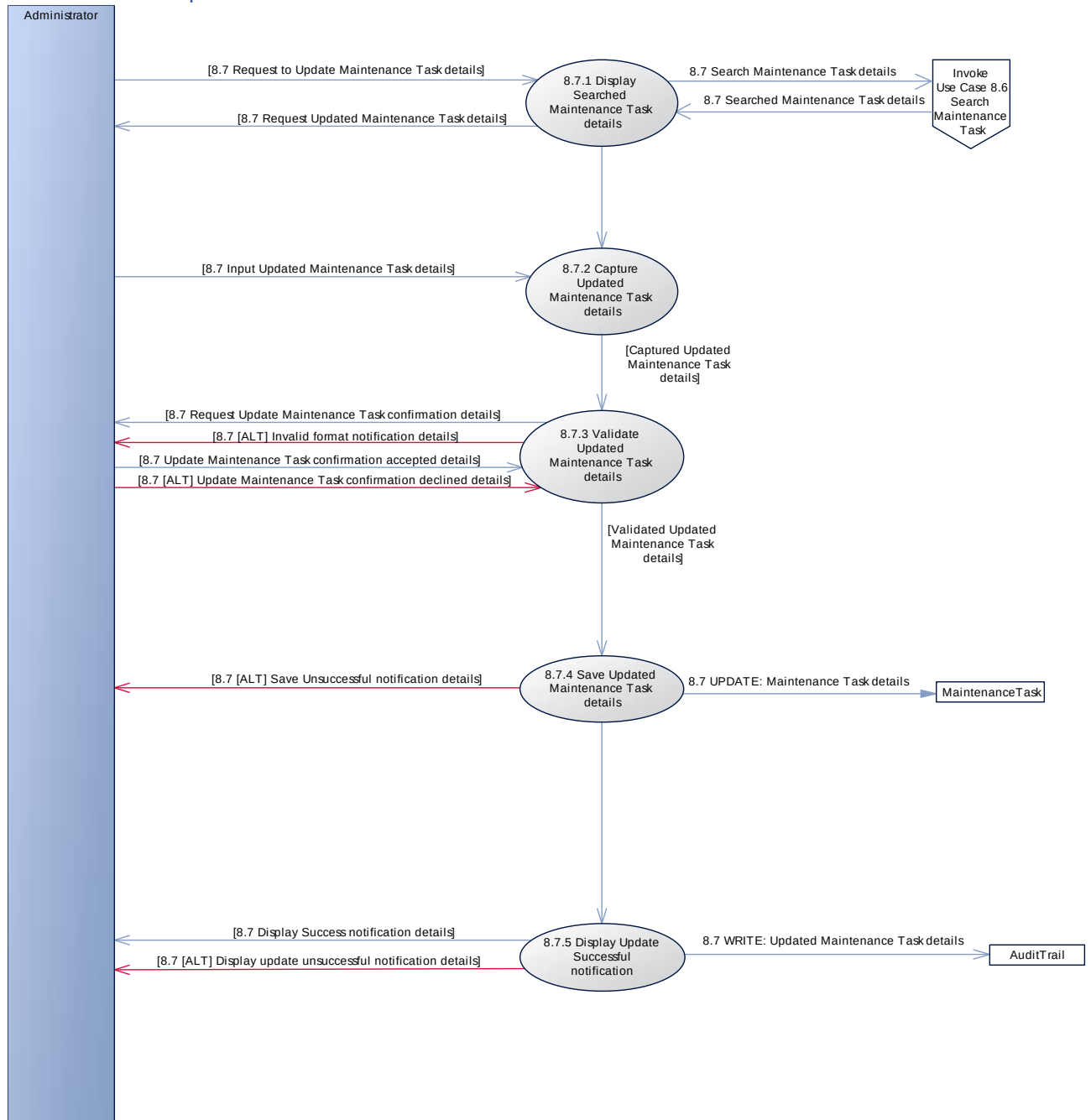


Figure 181 Update Maintenance Task Primitive Diagram



3.4.4.8.8 Delete Maintenance Task

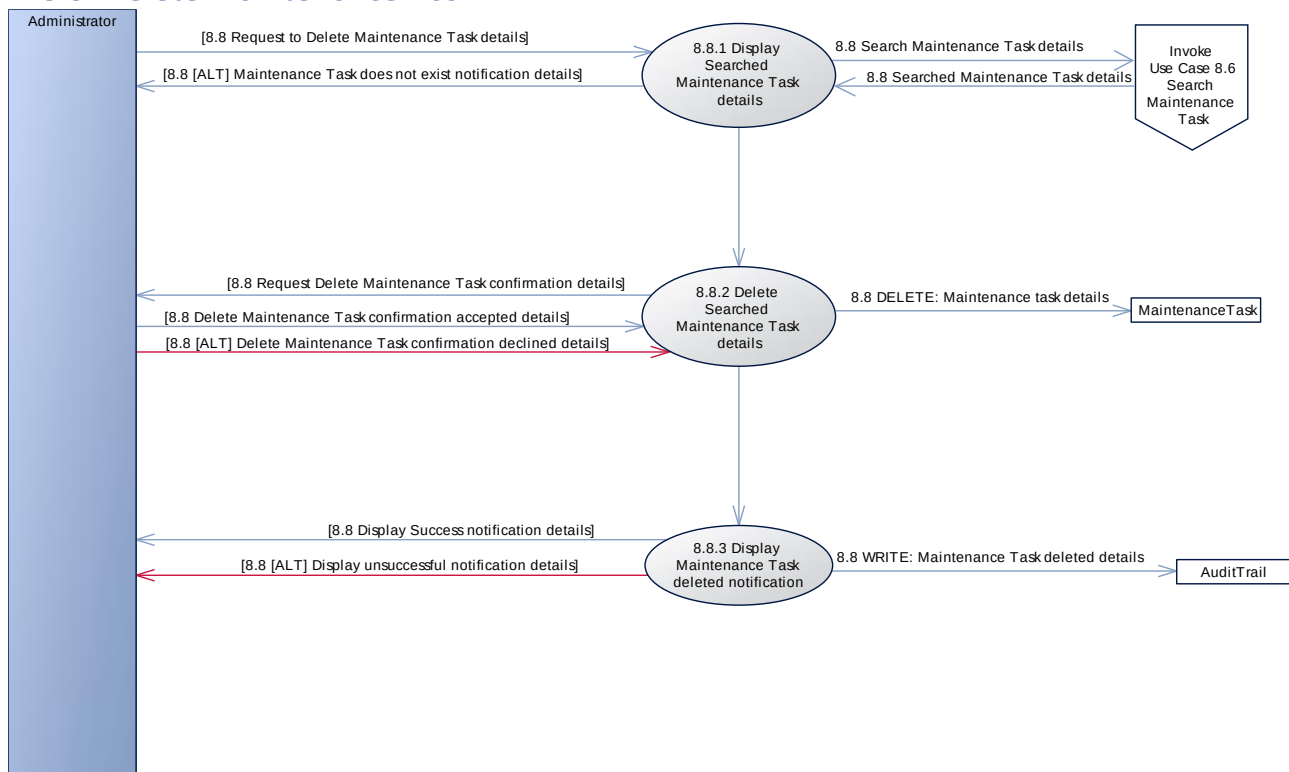


Figure 182 Delete Maintenance Task Primitive Diagram



3.4.4.9 Sub-system 9: School

3.4.4.9.1 Add School

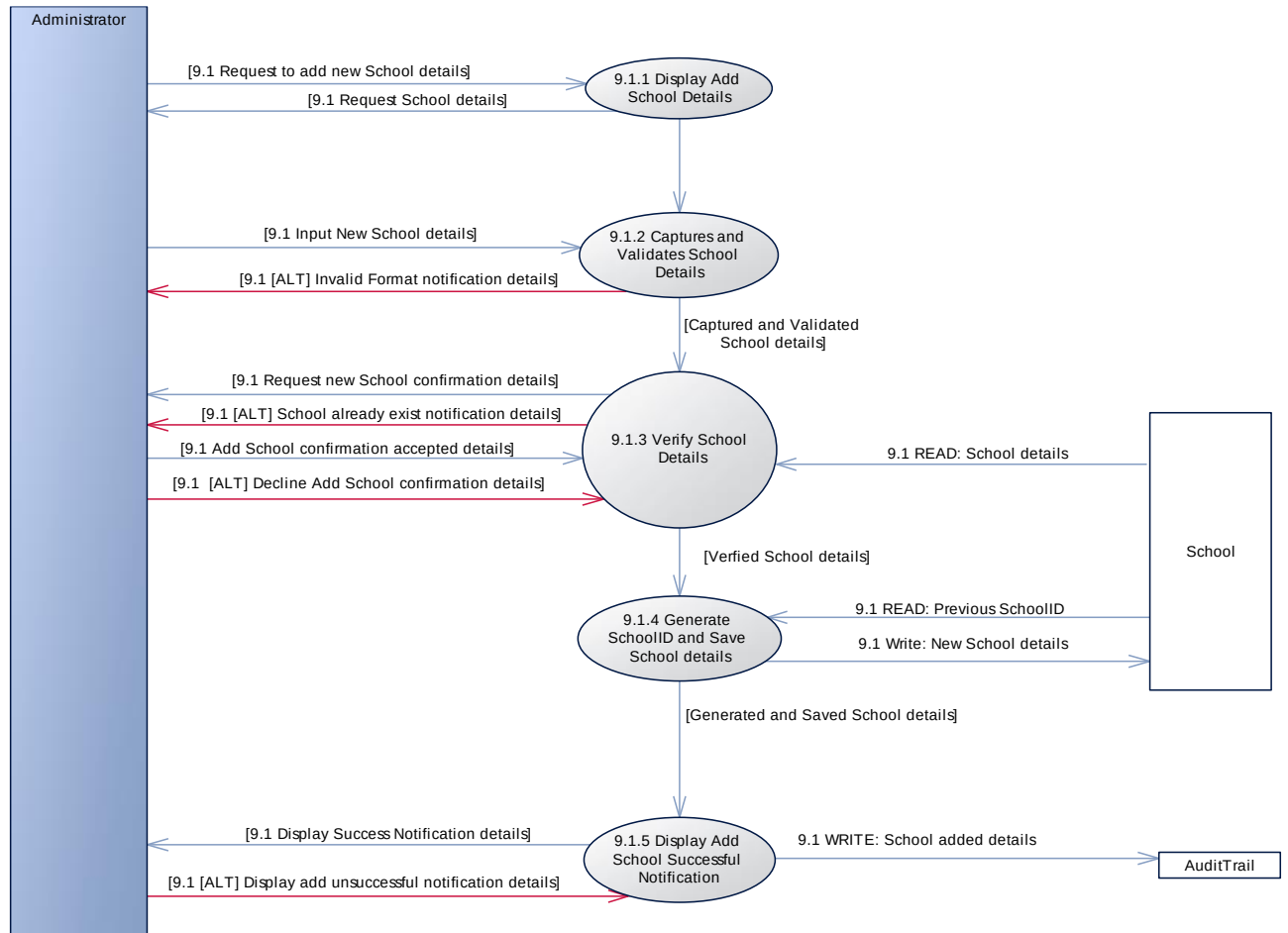


Figure 183 Add School Primitive Diagram



3.4.4.9.2 Search School

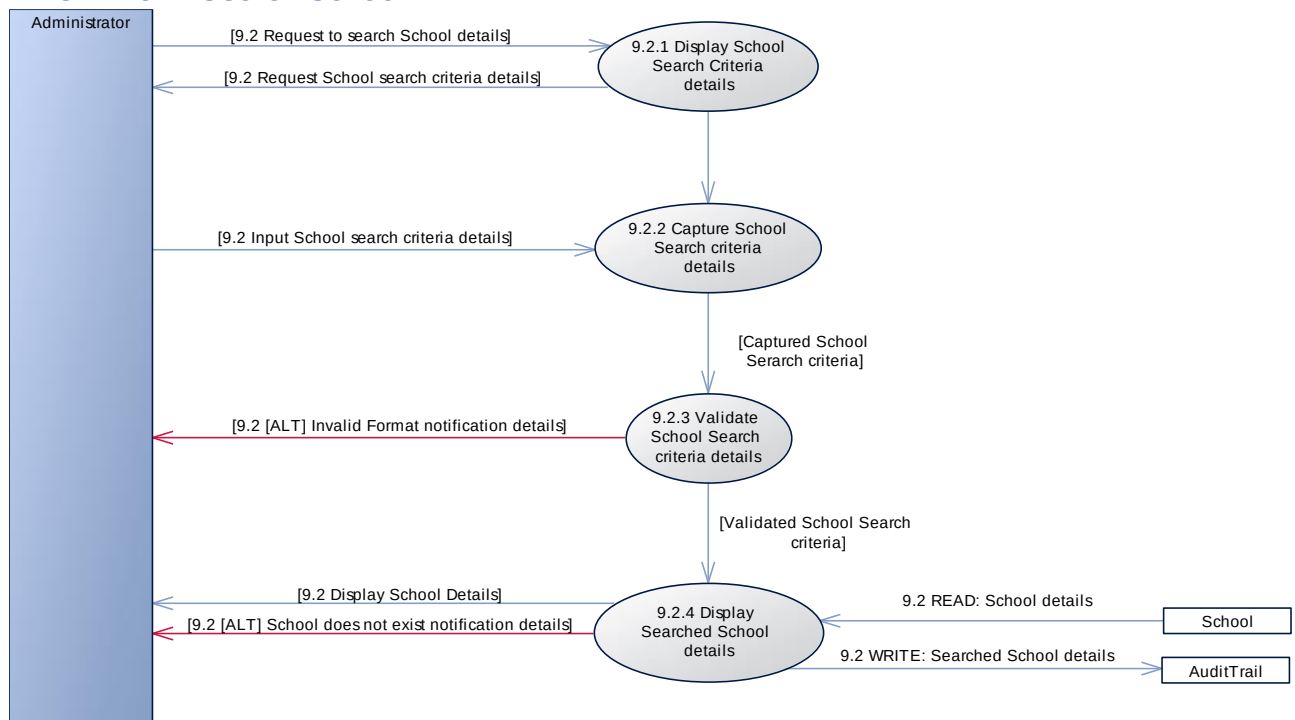


Figure 184 Search School Primitive Diagram



3.4.4.9.3 Update School

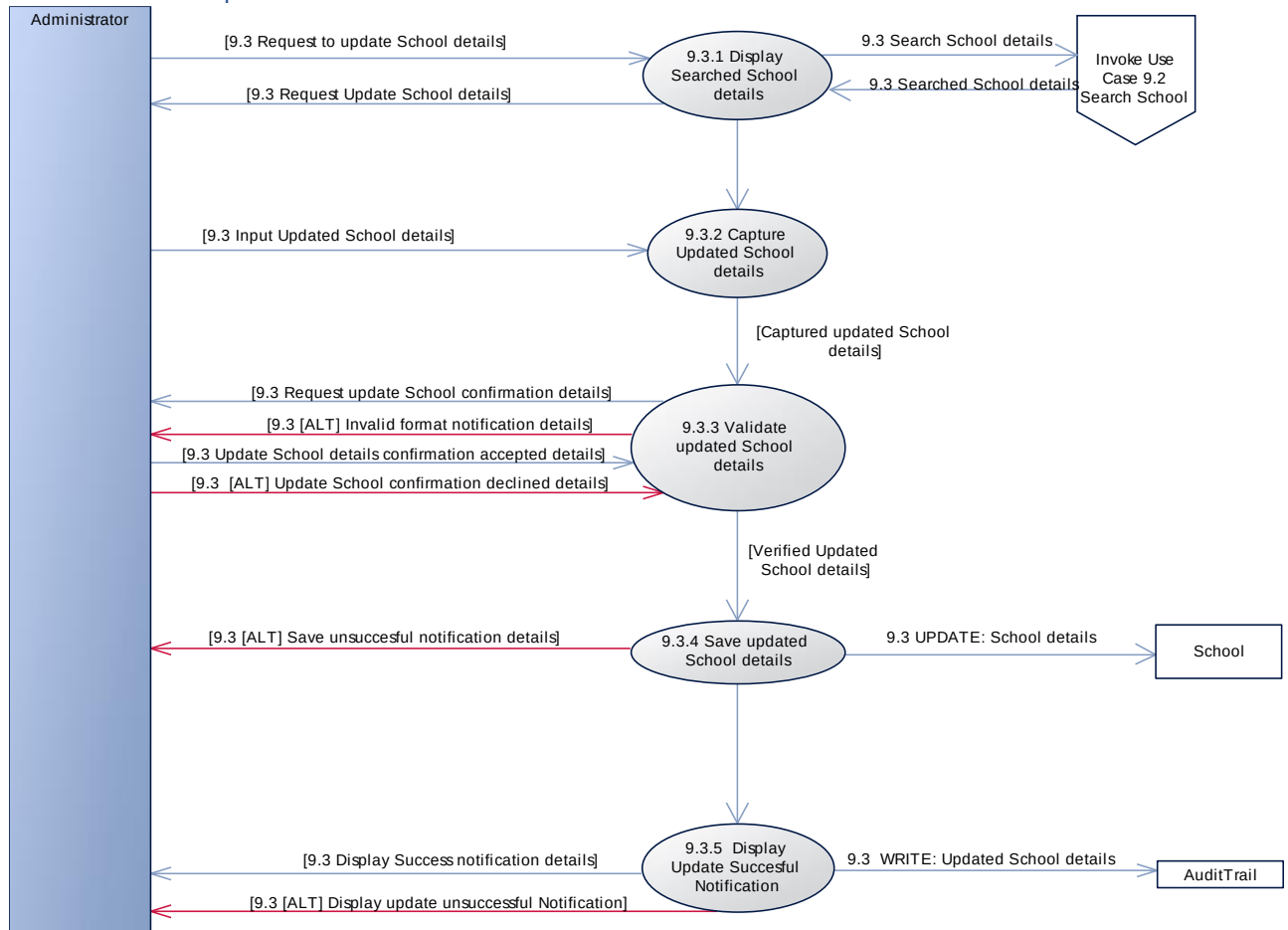


Figure 185 Update School Primitive Diagram



3.4.4.9.4 Delete School

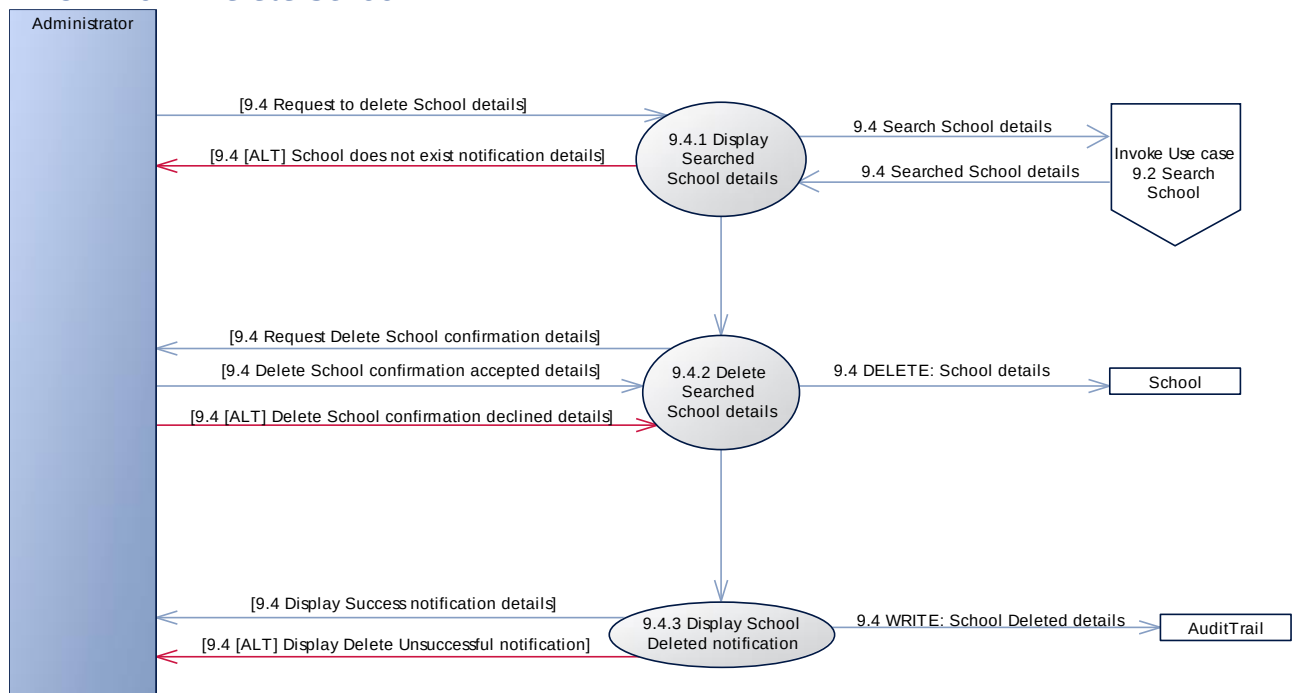


Figure 186 Delete School Primitive Diagram



3.4.4.10 Sub-system 10: Job

3.4.4.10.1 Report Fault

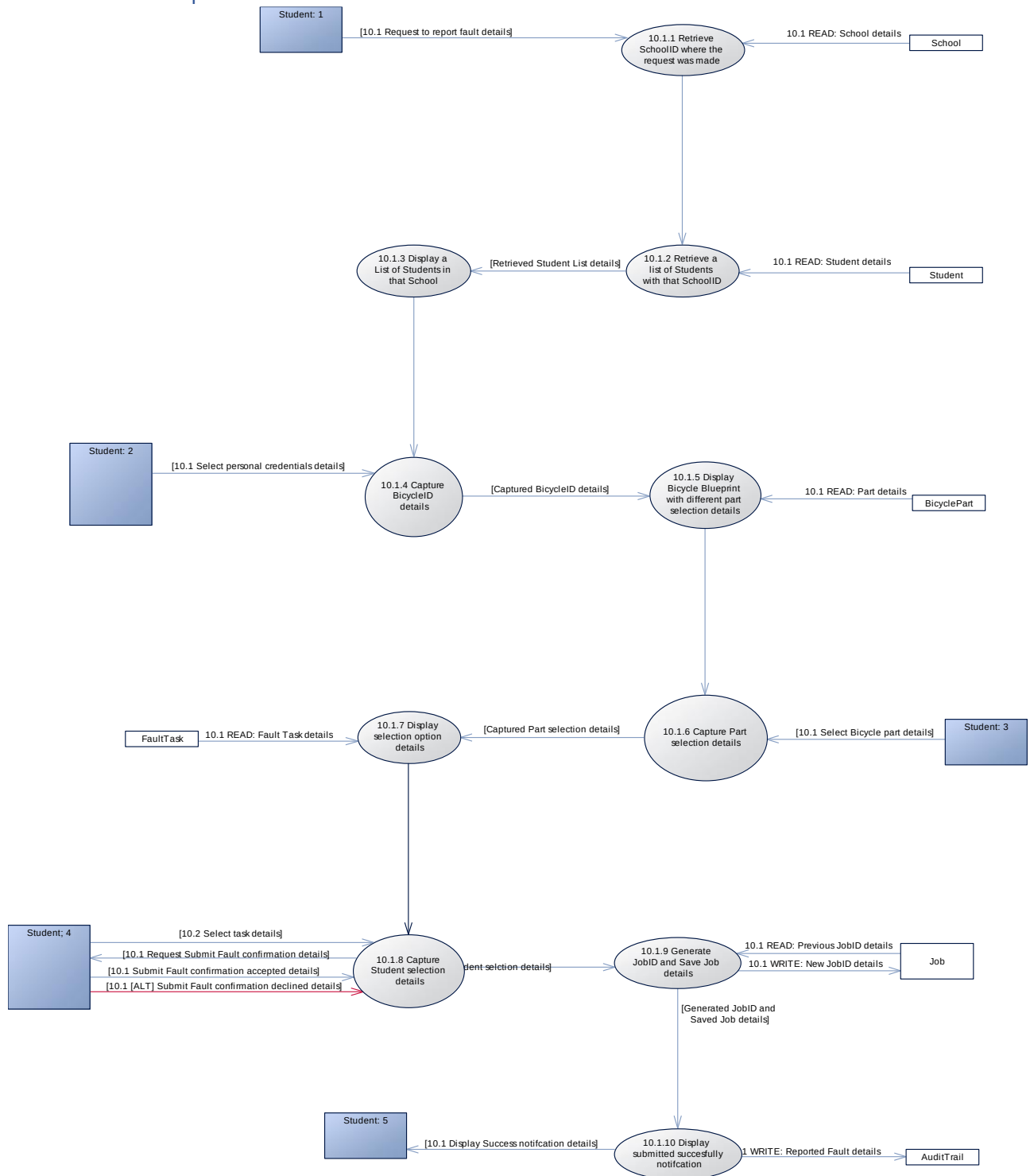


Figure 187 Report Fault Primitive Diagram



3.4.4.10.2 View Jobs

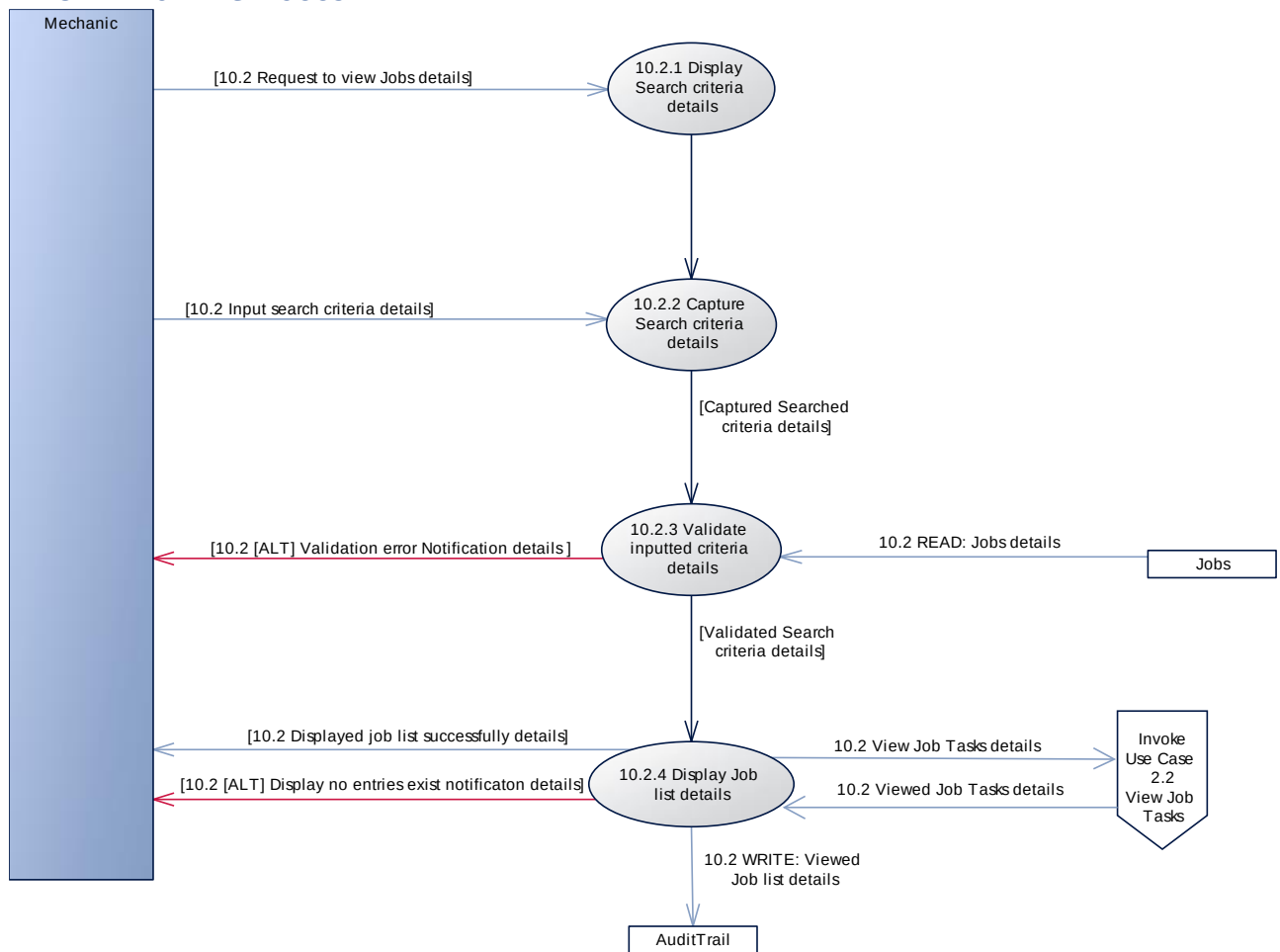


Figure 188 View Jobs Primitive Diagram



3.4.4.10.3 Check-in Bicycle

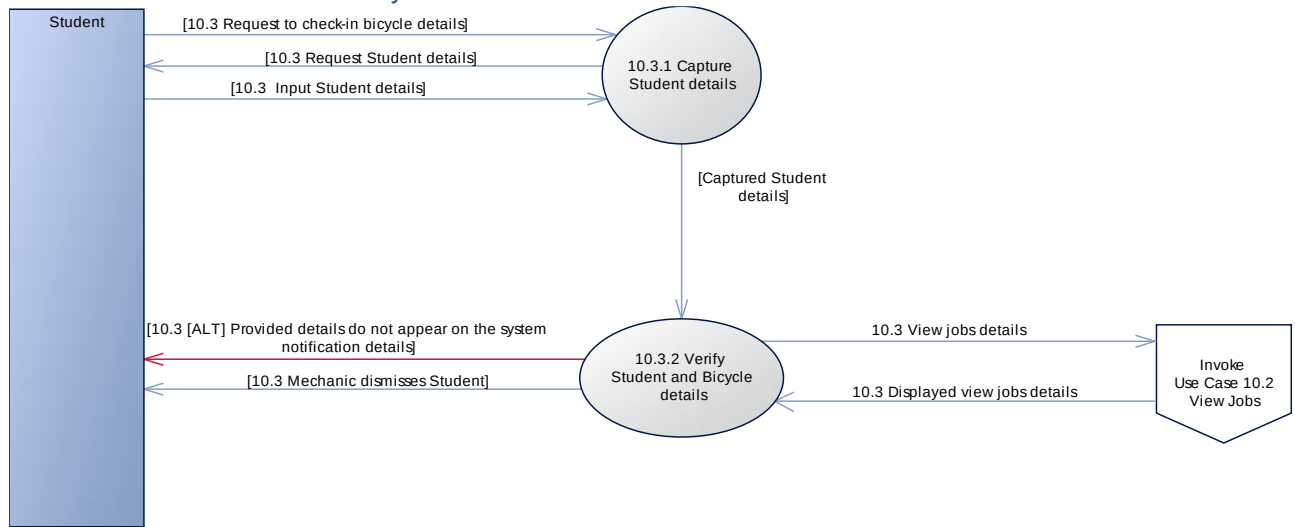


Figure 189 Check-in Bicycle Primitive Diagram



3.4.4.10.4 Report Maintenance

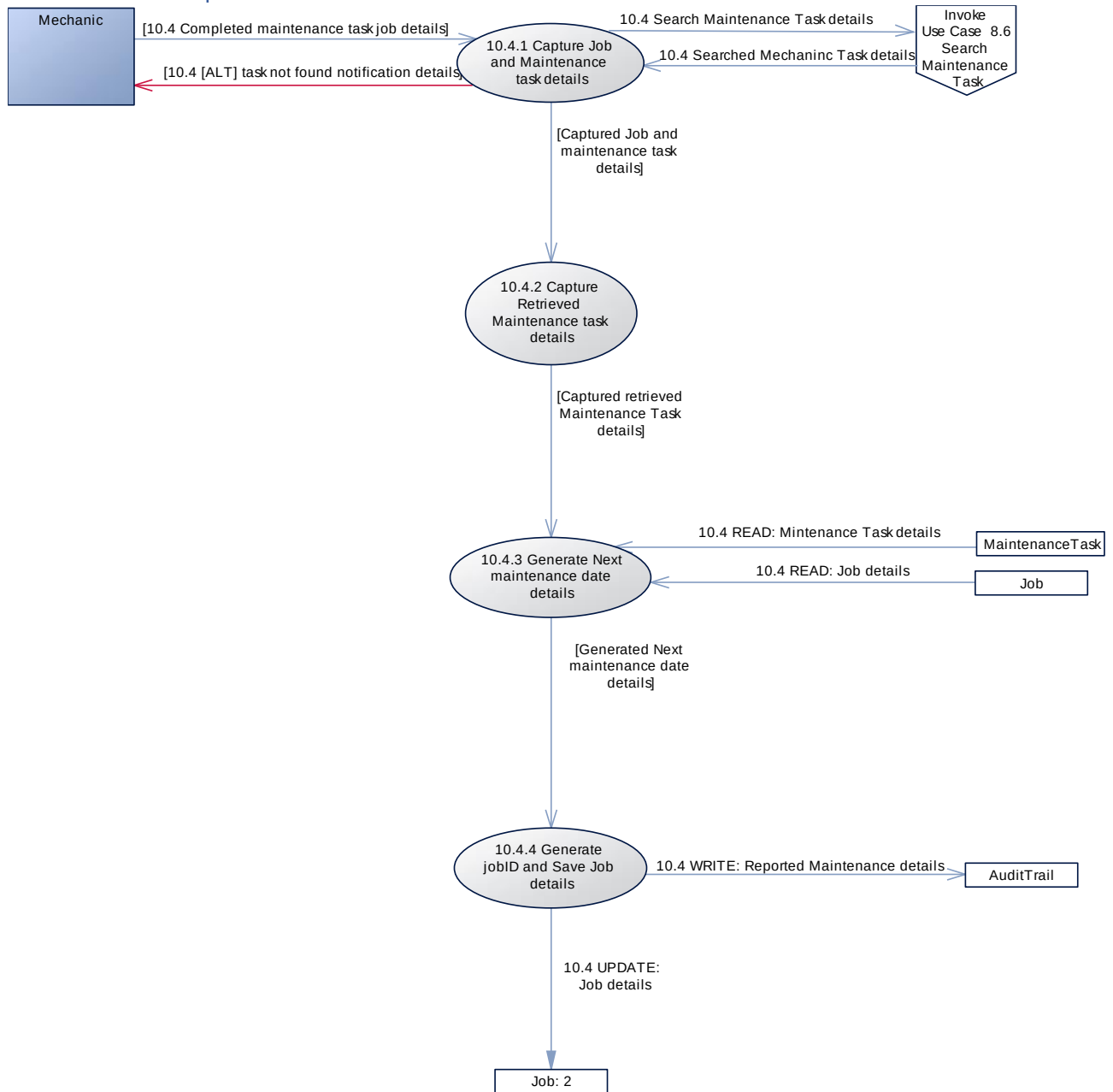


Figure 190 Report Maintenance Primitive Diagram



3.4.4.11 Sub-system 11: Reporting

3.4.4.11.1 Generate Maintenance Schedule

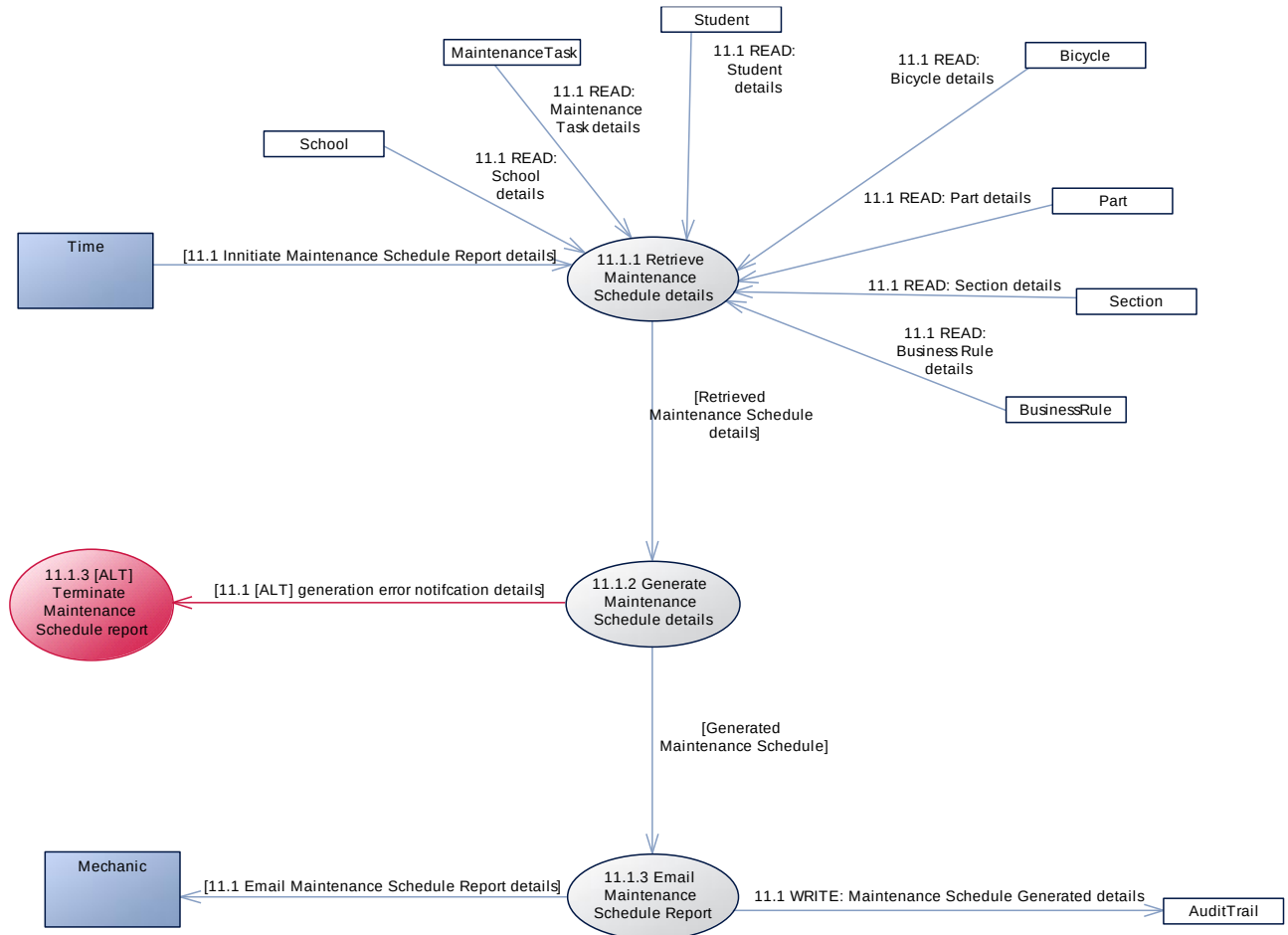


Figure 191 Generate Maintenance Schedule Primitive Diagram



3.4.4.11.2 Generate Maintenance Requirements Report

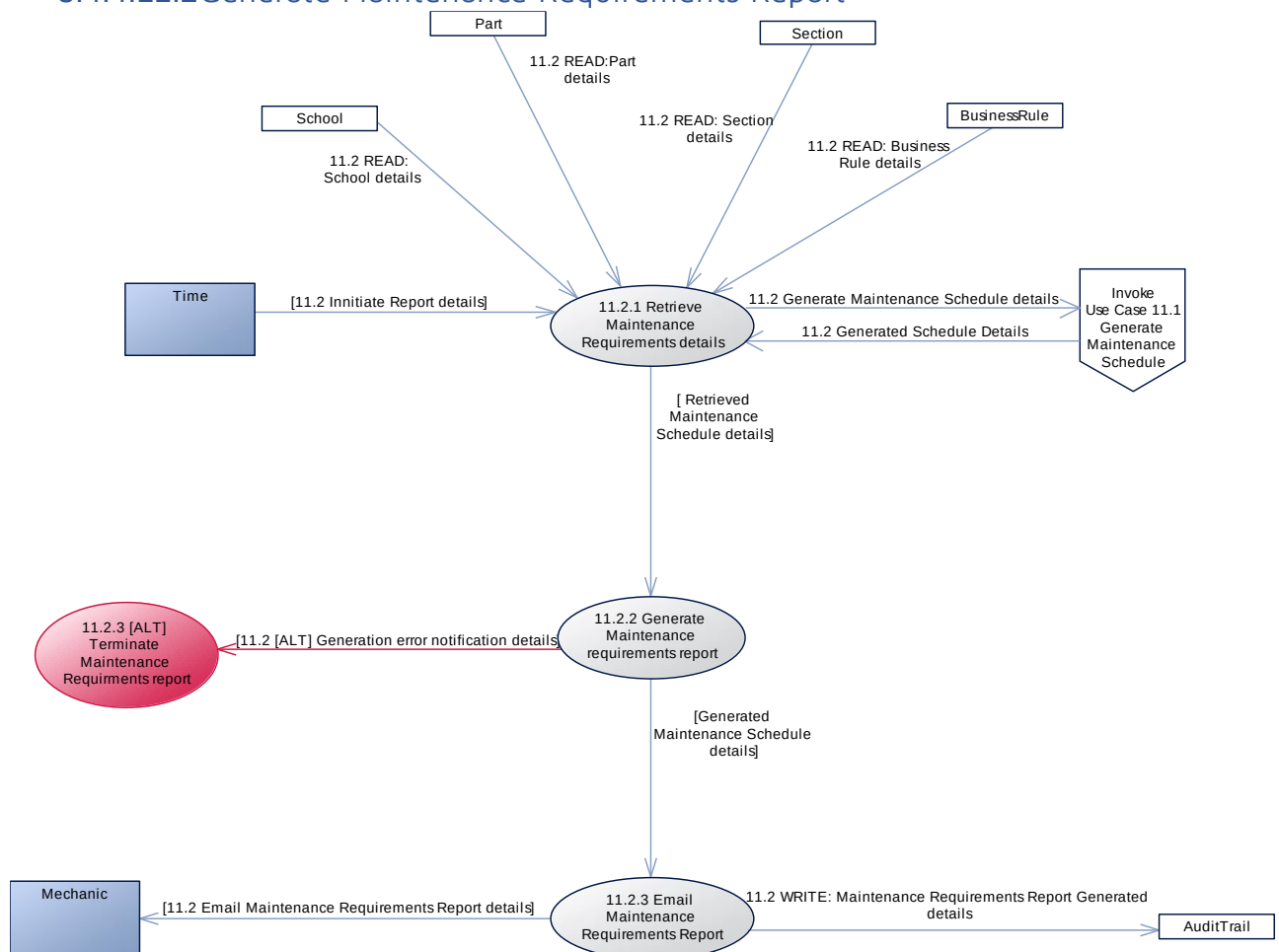


Figure 192 Generate Maintenance Requirements Report Primitive Diagram



3.4.4.11.3 Generate Student Report

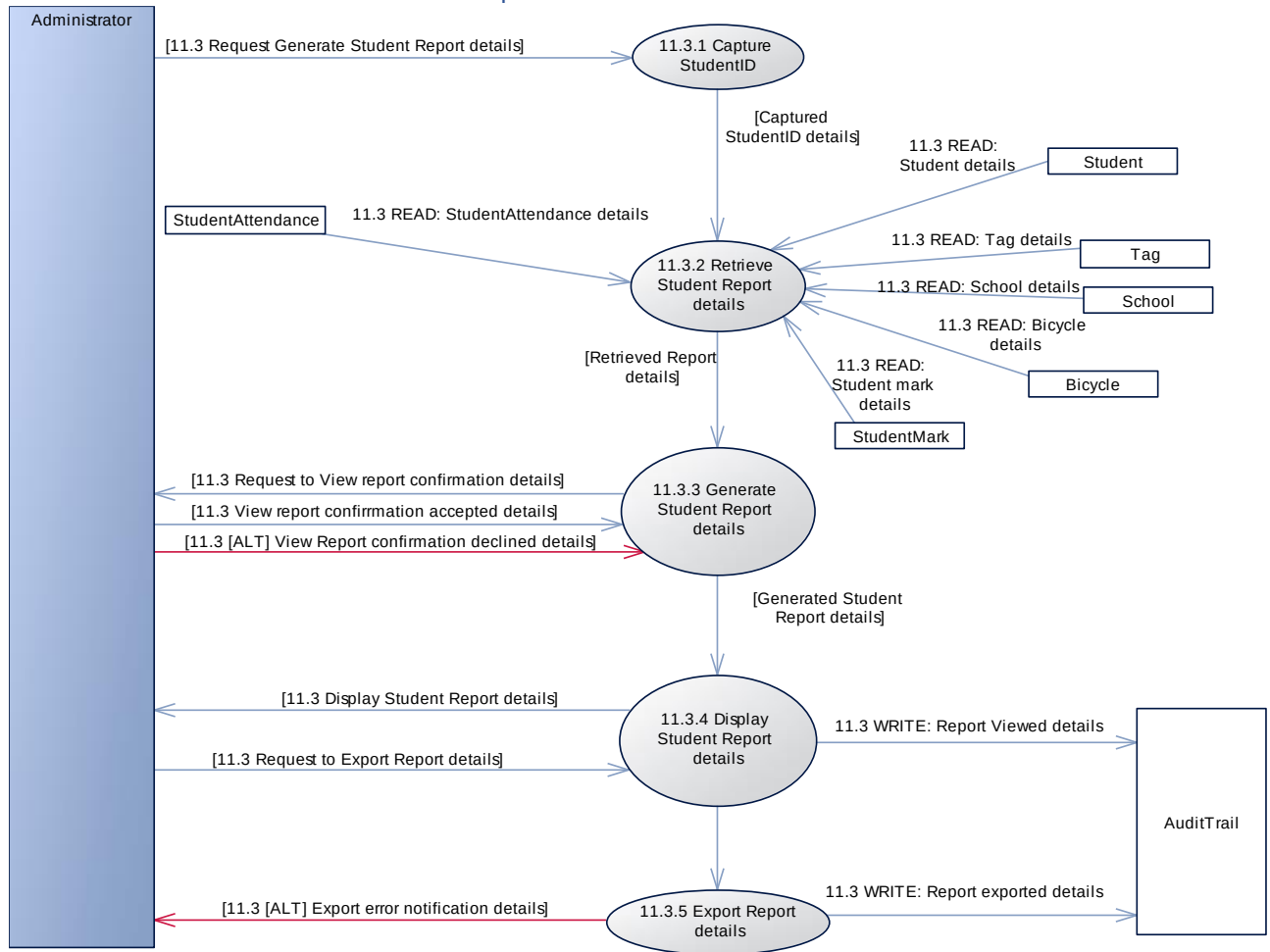


Figure 193 Generate Student Report Primitive Diagram



3.4.4.11.4 Generate Bicycle History Report

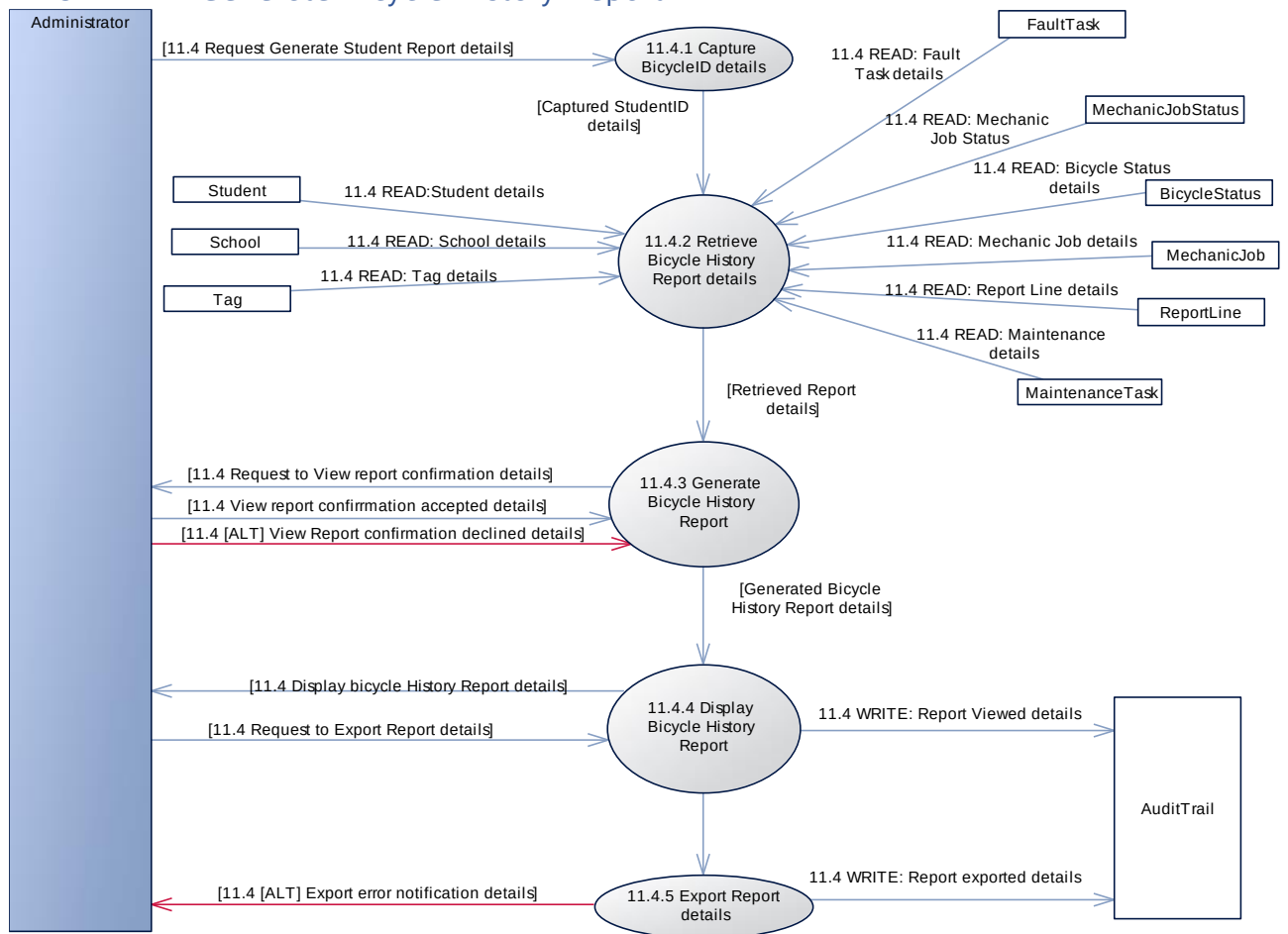


Figure 194 Generate Bicycle History Report Primitive Diagram



3.4.4.11.5 Generate Log Report

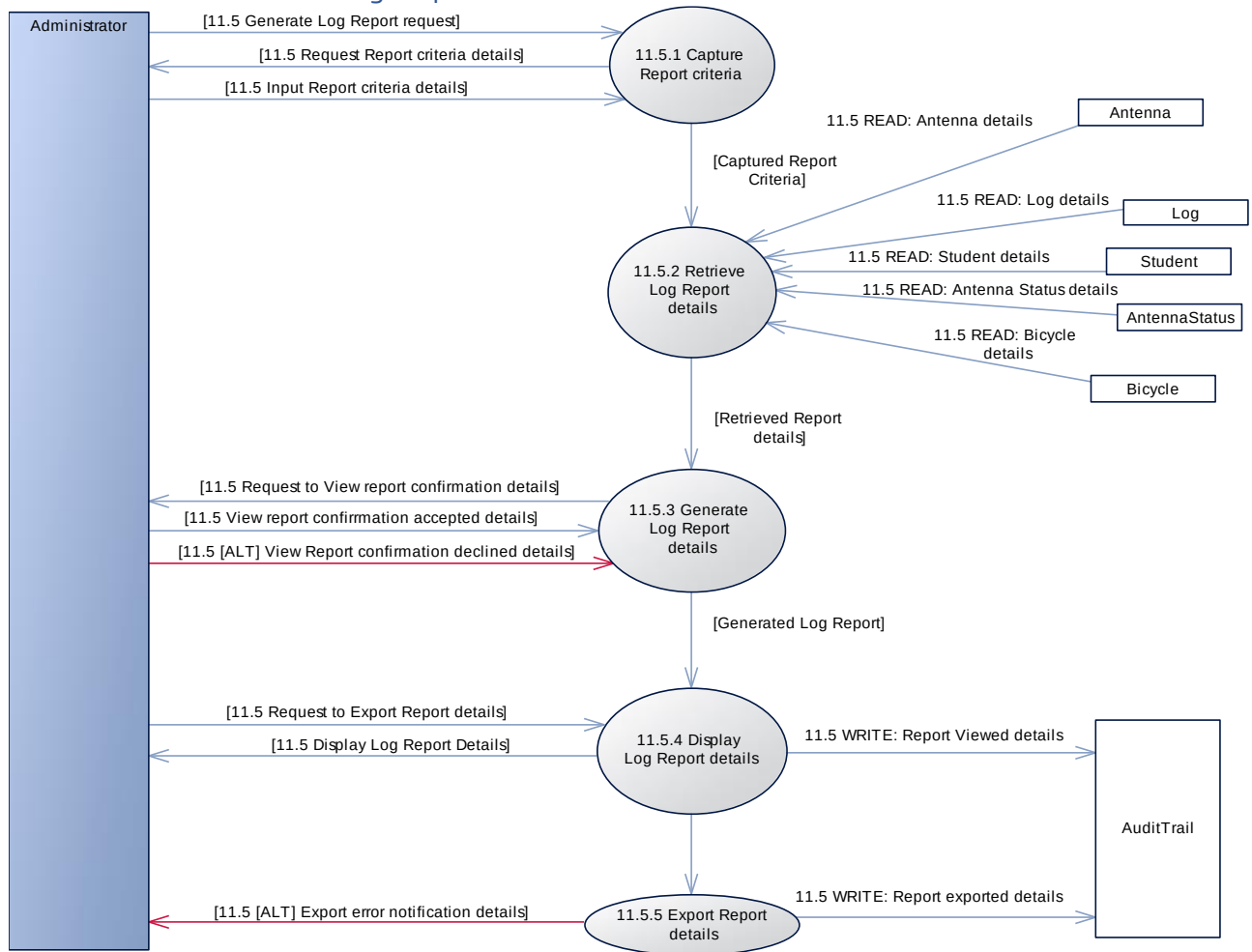


Figure 195 Generate Log Report Primitive Diagram



3.4.4.11.6 Generate Parts Used Report

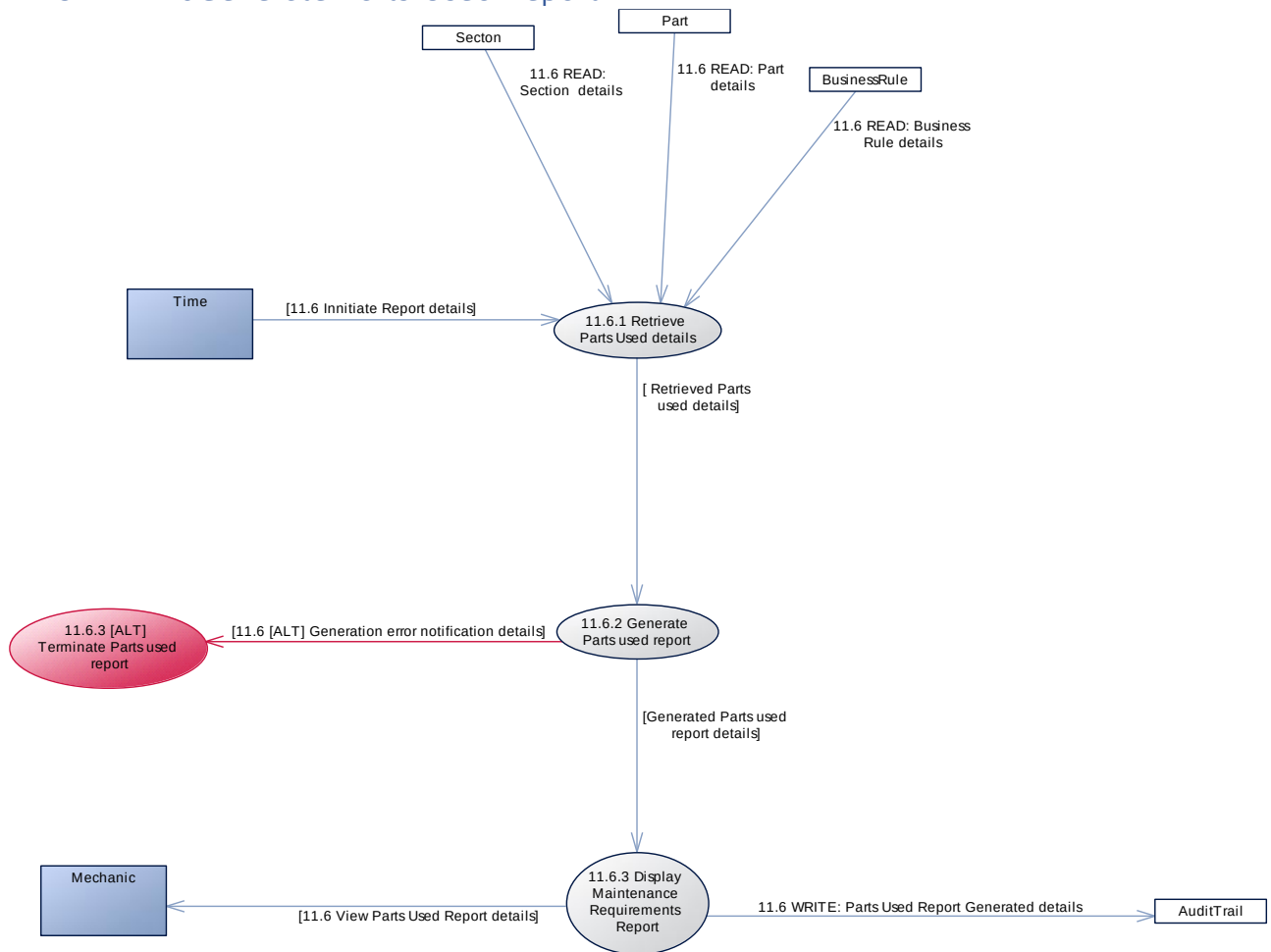


Figure 196 Generate Parts Used Report Primitive Diagram



3.4.4.12 Sub-system 12: Notification

3.4.4.12.1 Send Student Did Not Arrive Notification

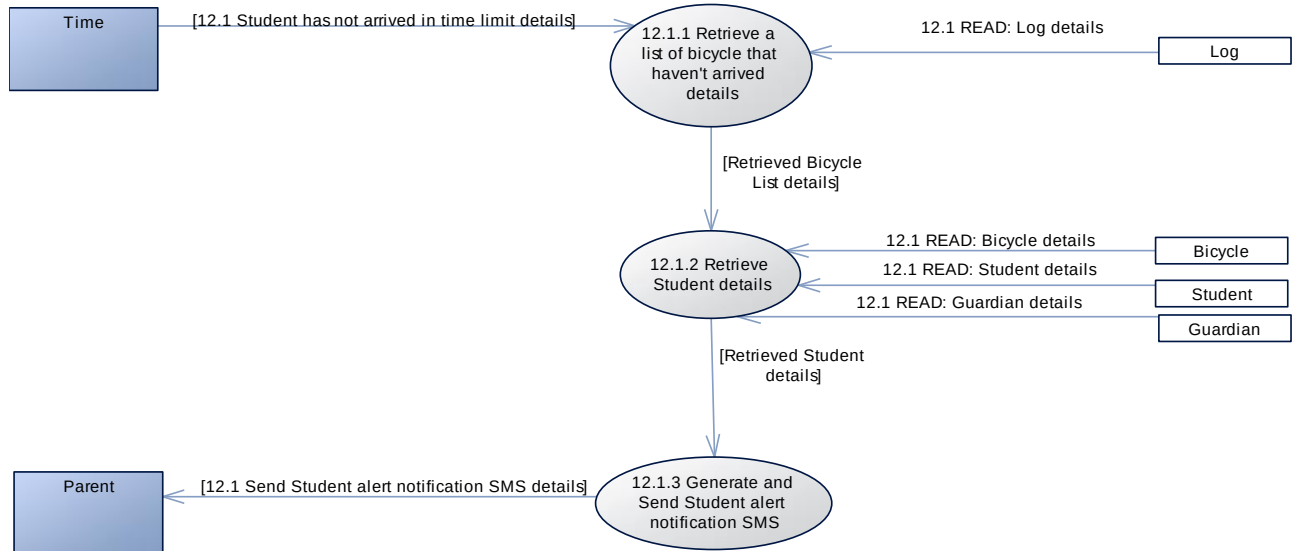


Figure 197 Student Did Not Arrive Notification Primitive Diagram



3.4.4.12.2 Send Student Maintenance Notification

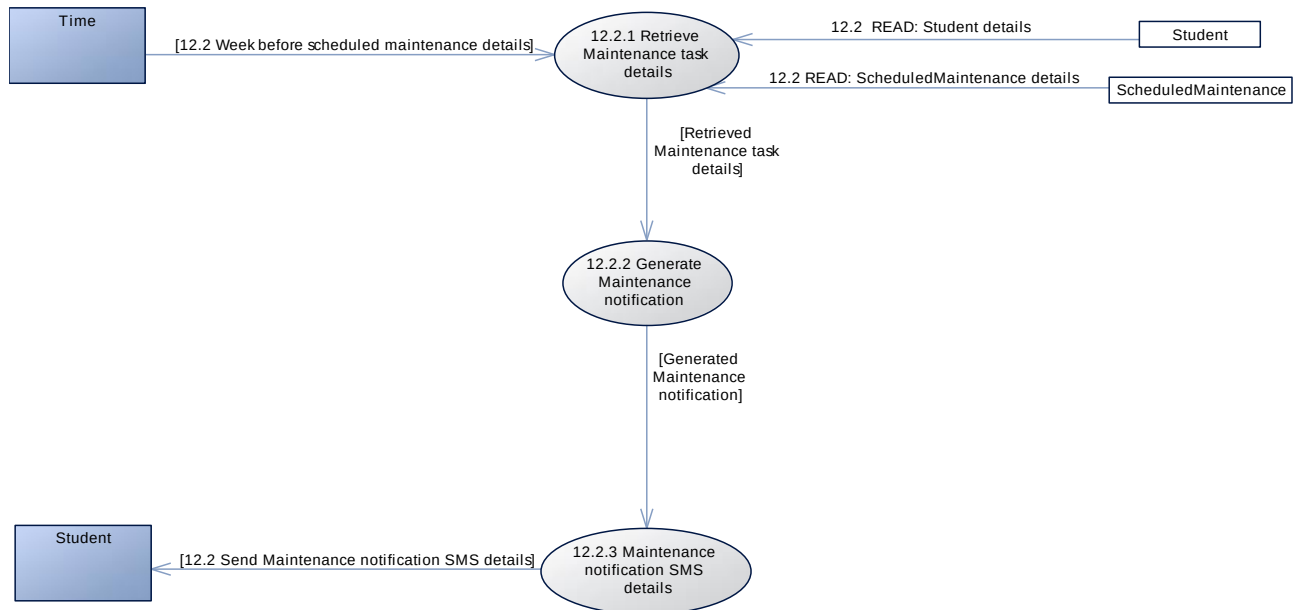


Figure 198 Send Student Maintenance Notification Primitive Diagram



3.4.4.13 Sub-system 13: Administration

3.4.4.13.1 Backup Data

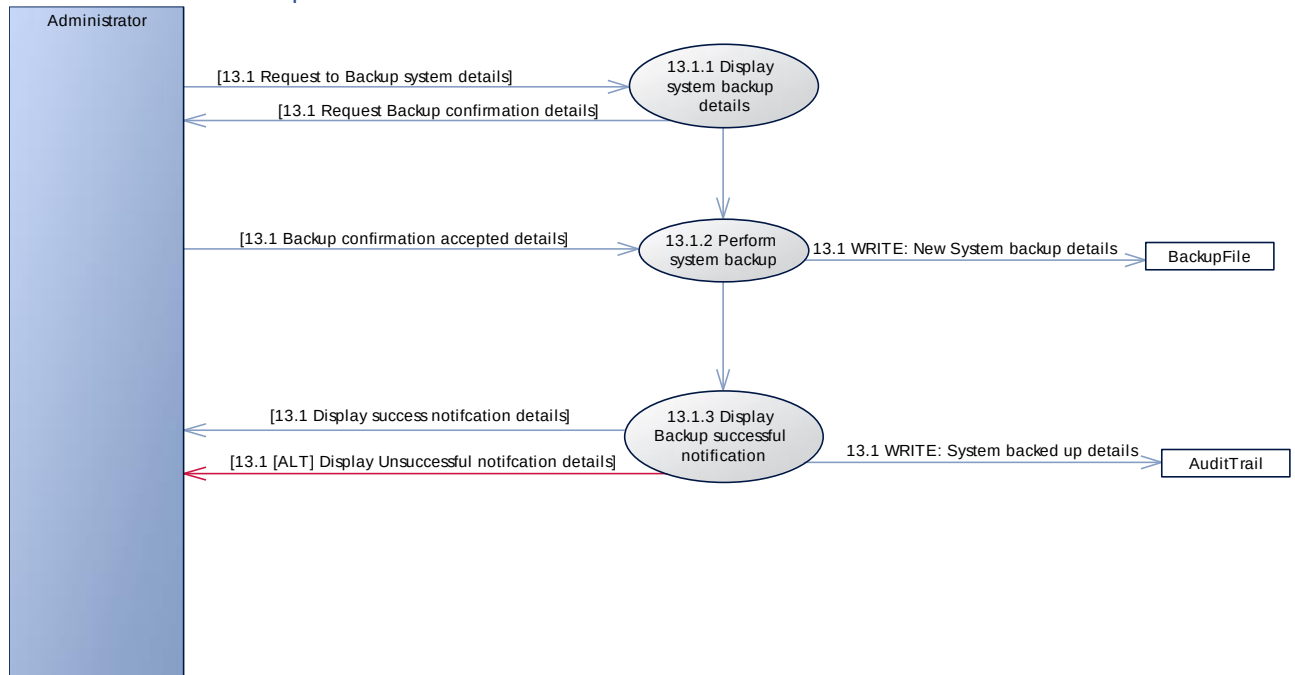


Figure 199 Backup Data Primitive Diagram



3.4.4.13.2 Restore Data

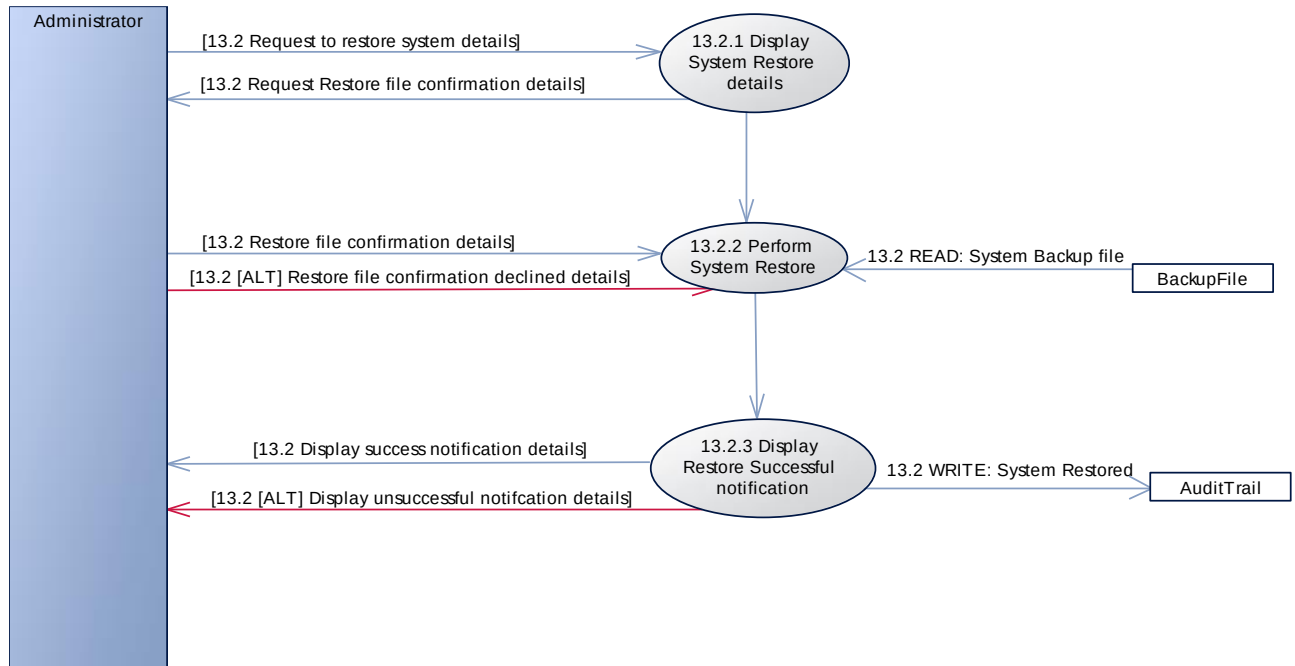


Figure 200 Restore Data Primitive Diagram



3.4.4.13.3 View Audit Trail

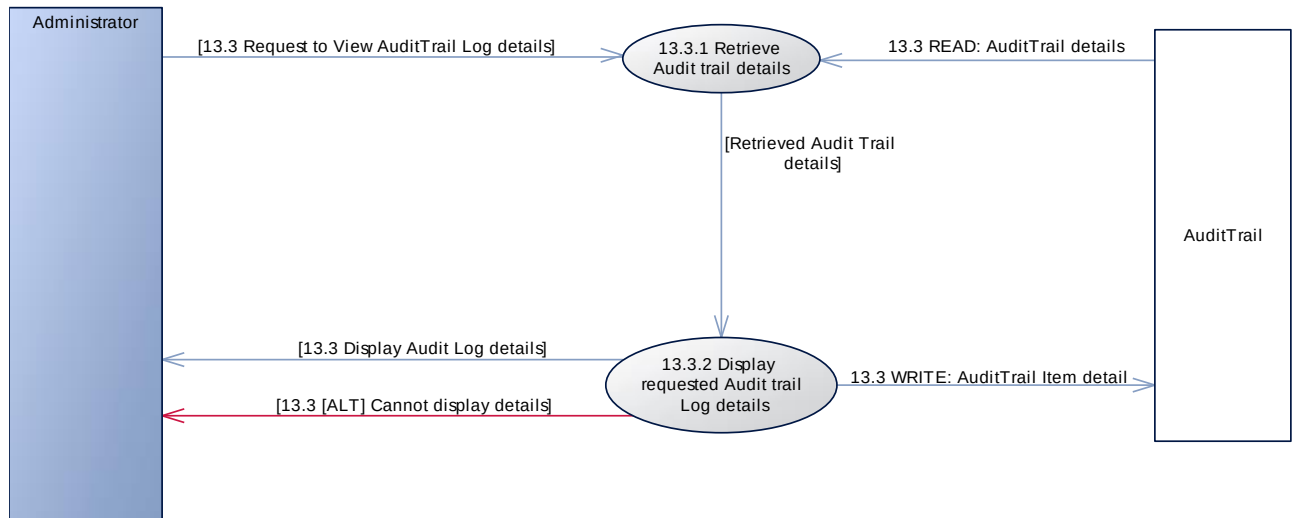


Figure 201 View Audit Trail Primitive Diagram



3.4.4.13.4 Search Audit Trail Entry

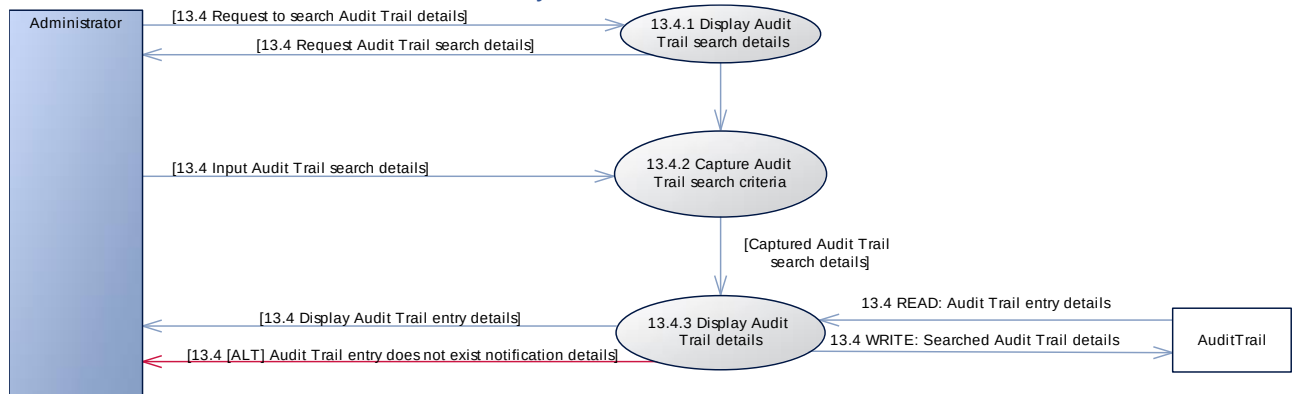


Figure 202 Search Audit Trail Entry Primitive Diagram



3.4.4.13.5 Add Route

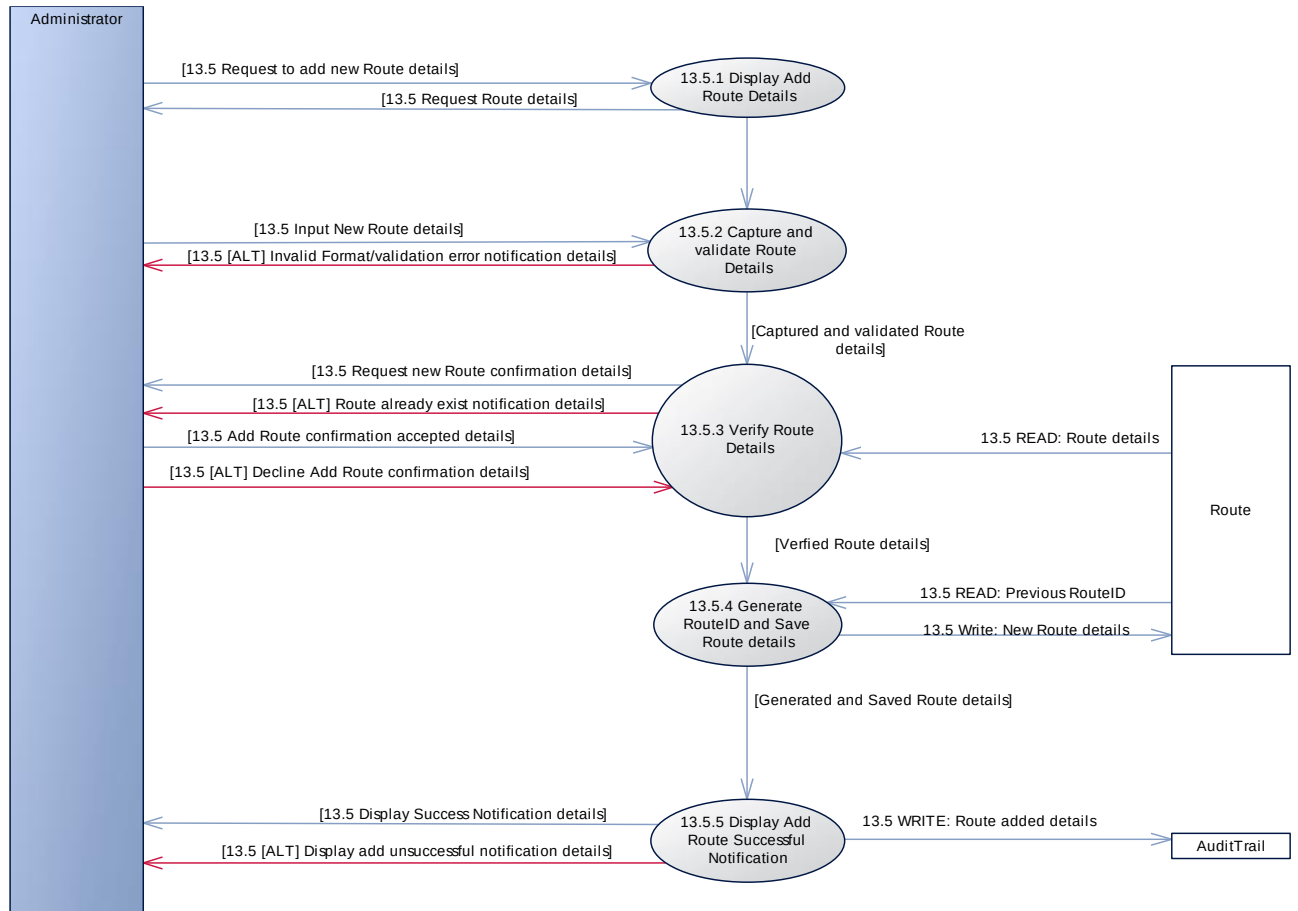


Figure 203 Add Route Primitive Diagram



3.4.4.13.6 Search Route

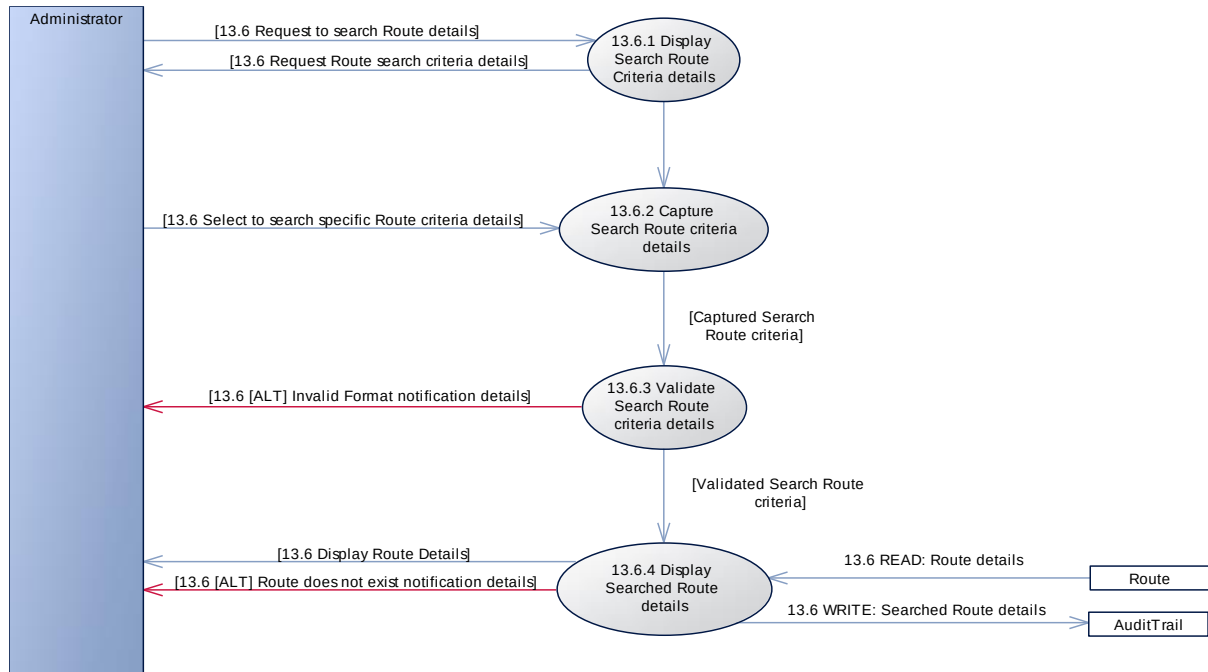


Figure 204 Search Route Primitive Diagram



3.4.4.13.7 Update Route

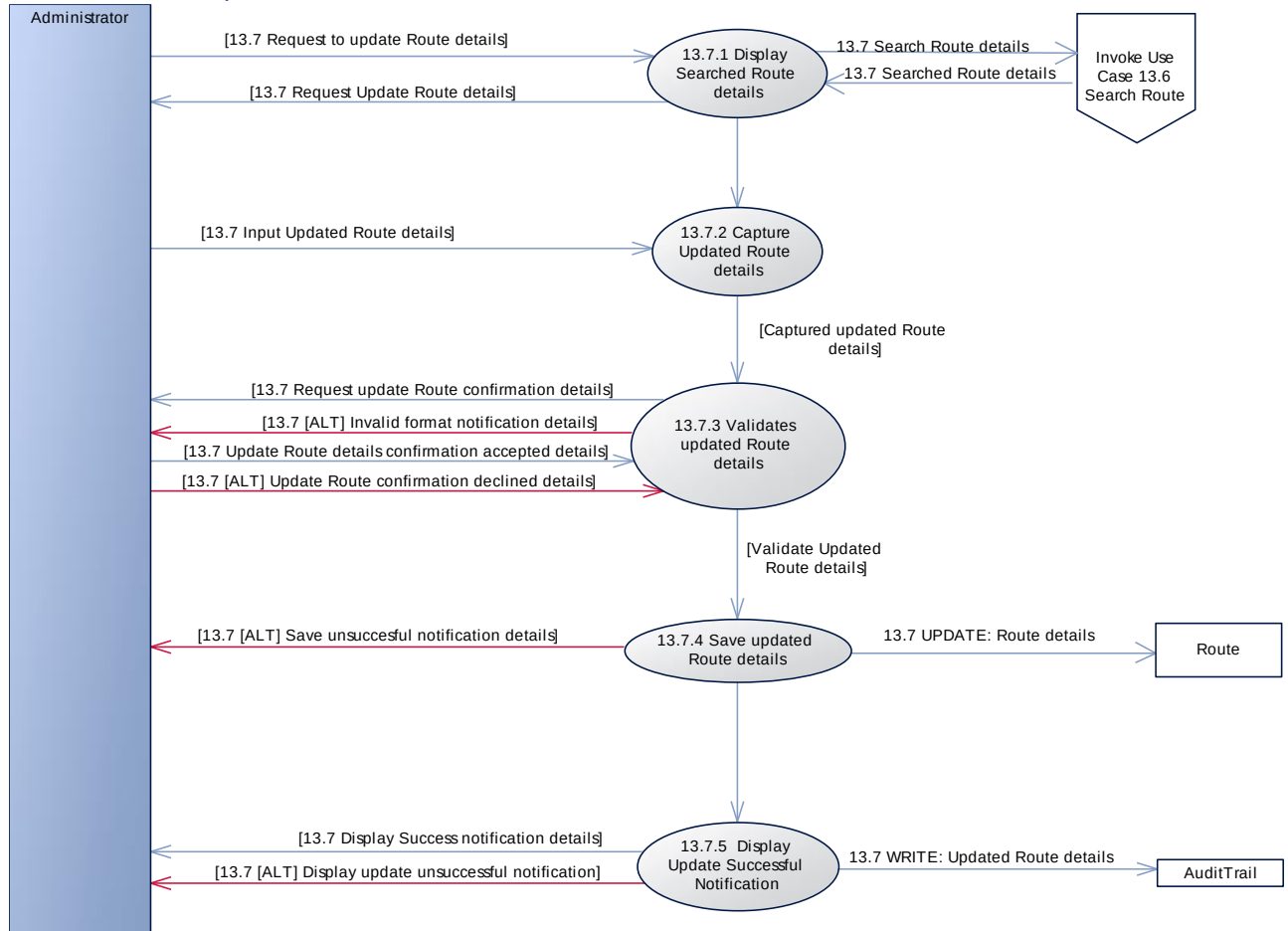


Figure 205 Update Route Primitive Diagram



3.4.4.13.8 Delete Route

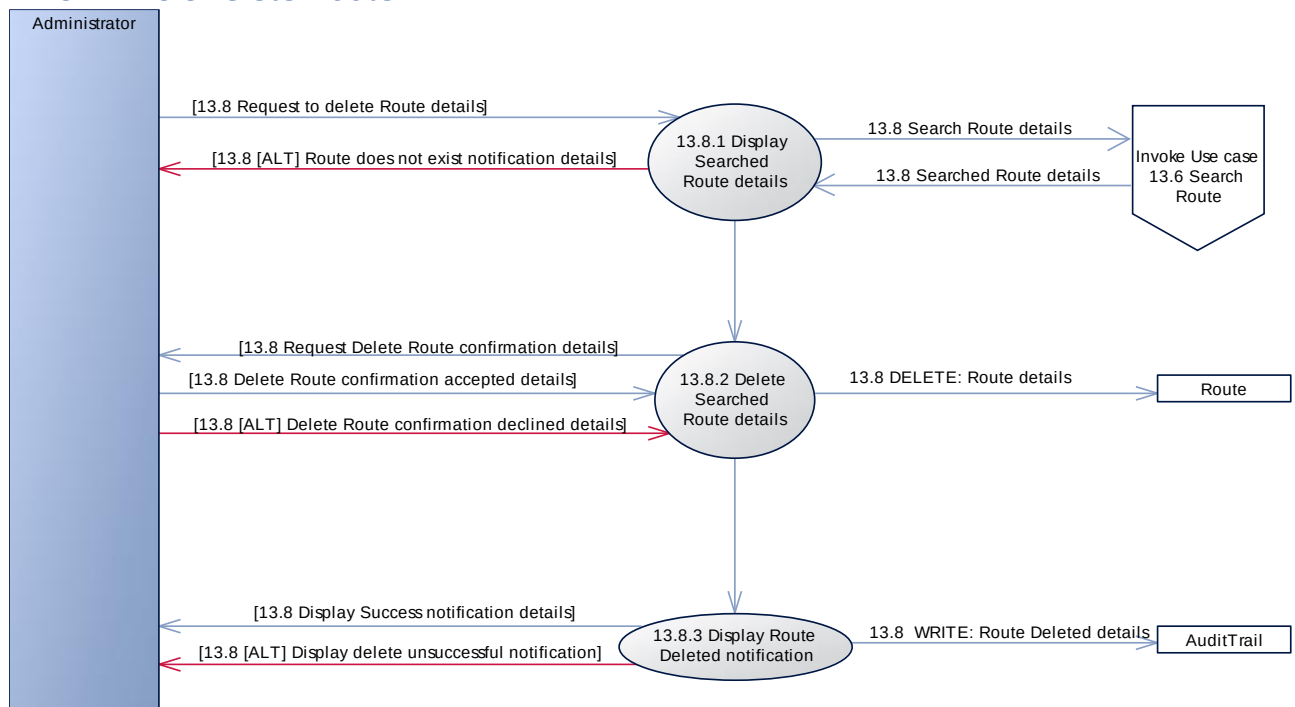


Figure 206 Delete Route Primitive Diagram



Virtual Visionaries
We See What You Don't

11/09/20
#BIKES4ERP



3.4.5 Conclusion

After reviewing all our Data Flow Diagrams, we can now fully gain a understanding to how our processes work.



4. UML Modelling

4.1 Introduction

The diagrams are essentially a way to view an entire process from start to end in the form of a diagram. It allows us to see how the user will interact with the system every step and how the system will respond to the user.



4.2 Activity Diagrams

4.2.1 Sub-system 1: User

4.2.1.1 Login

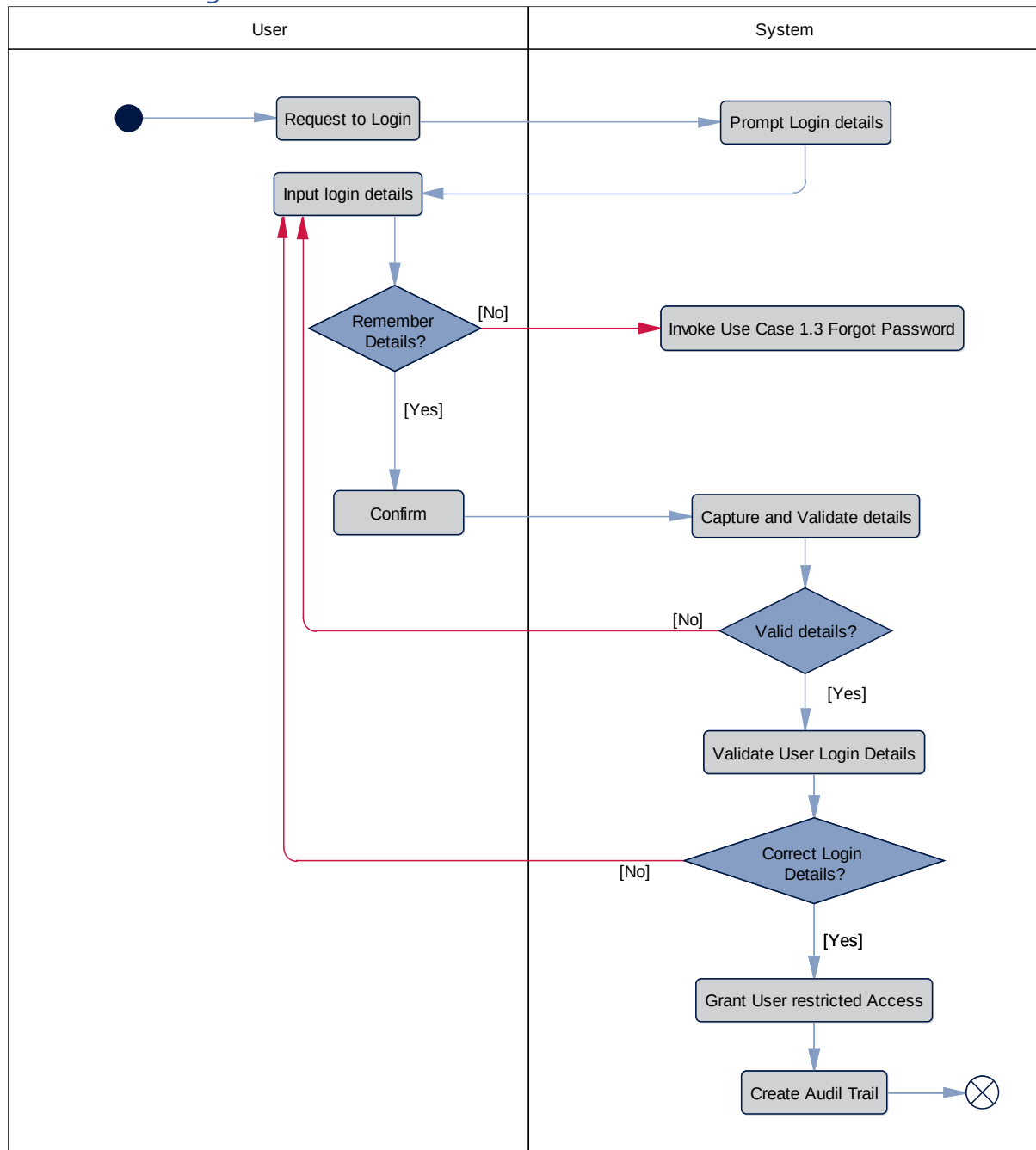


Figure 207 Login Activity Diagram



4.2.1.2 Logout

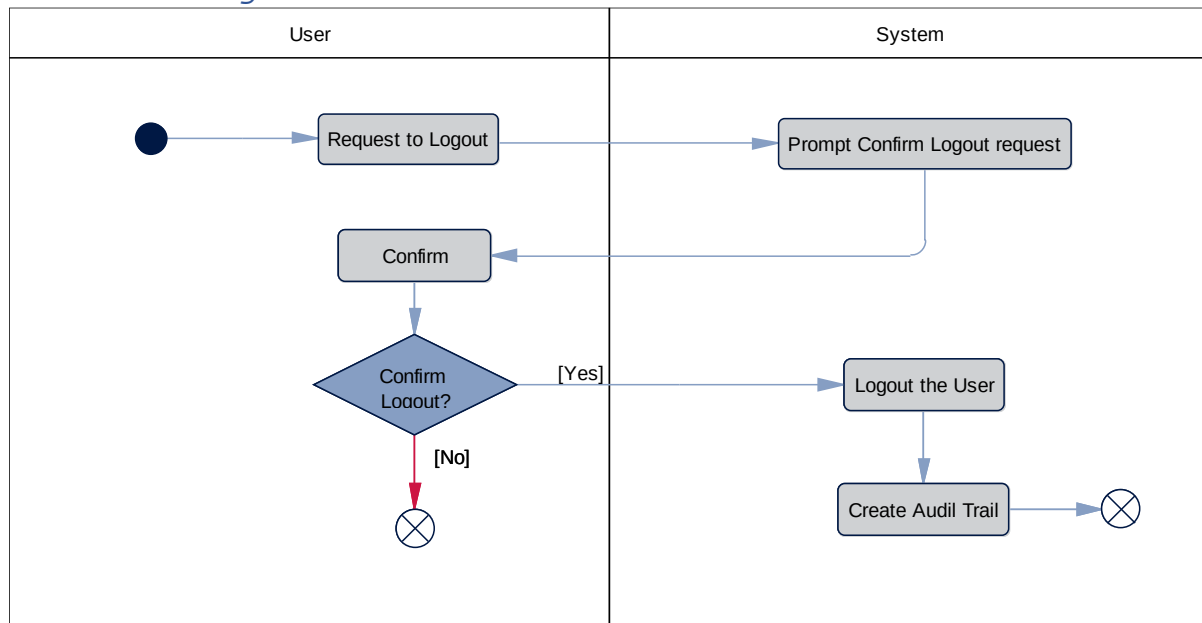


Figure 208 Logout Activity Diagram

4.2.1.3 *Forgot Password*

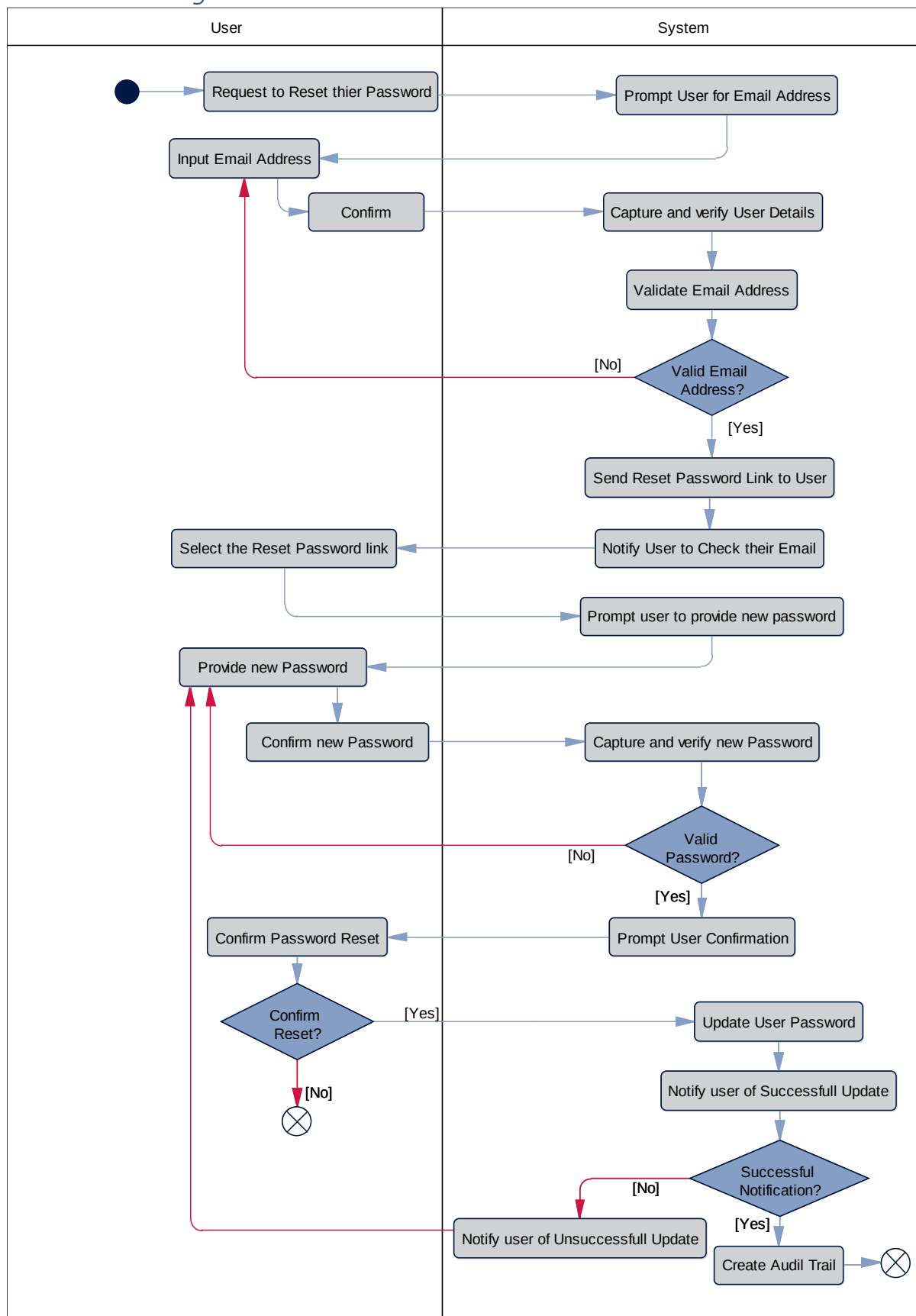


Figure 209 Forgot Password



4.2.1.4 Add User

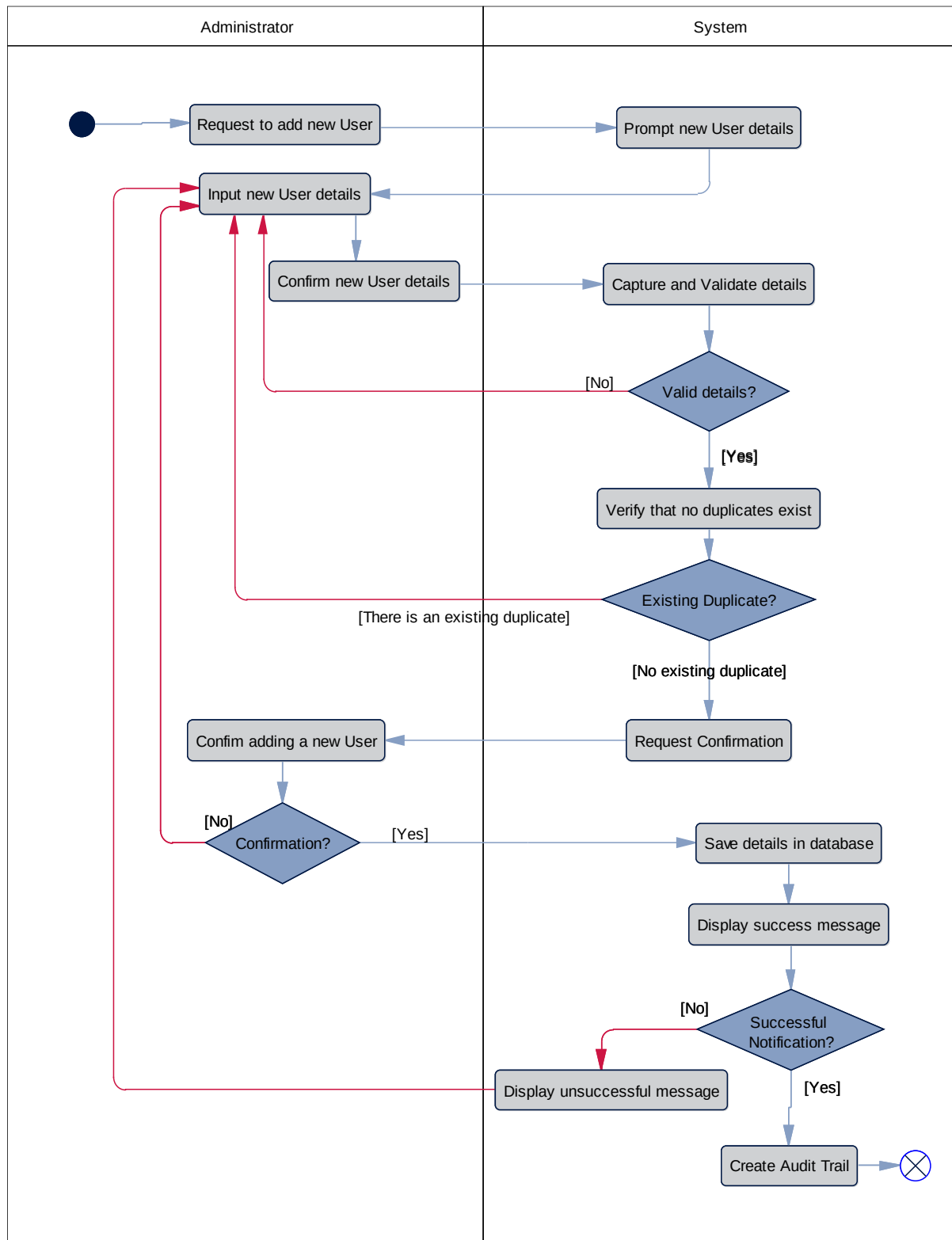


Figure 210 Add User Activity Diagram



4.2.1.5 Search User

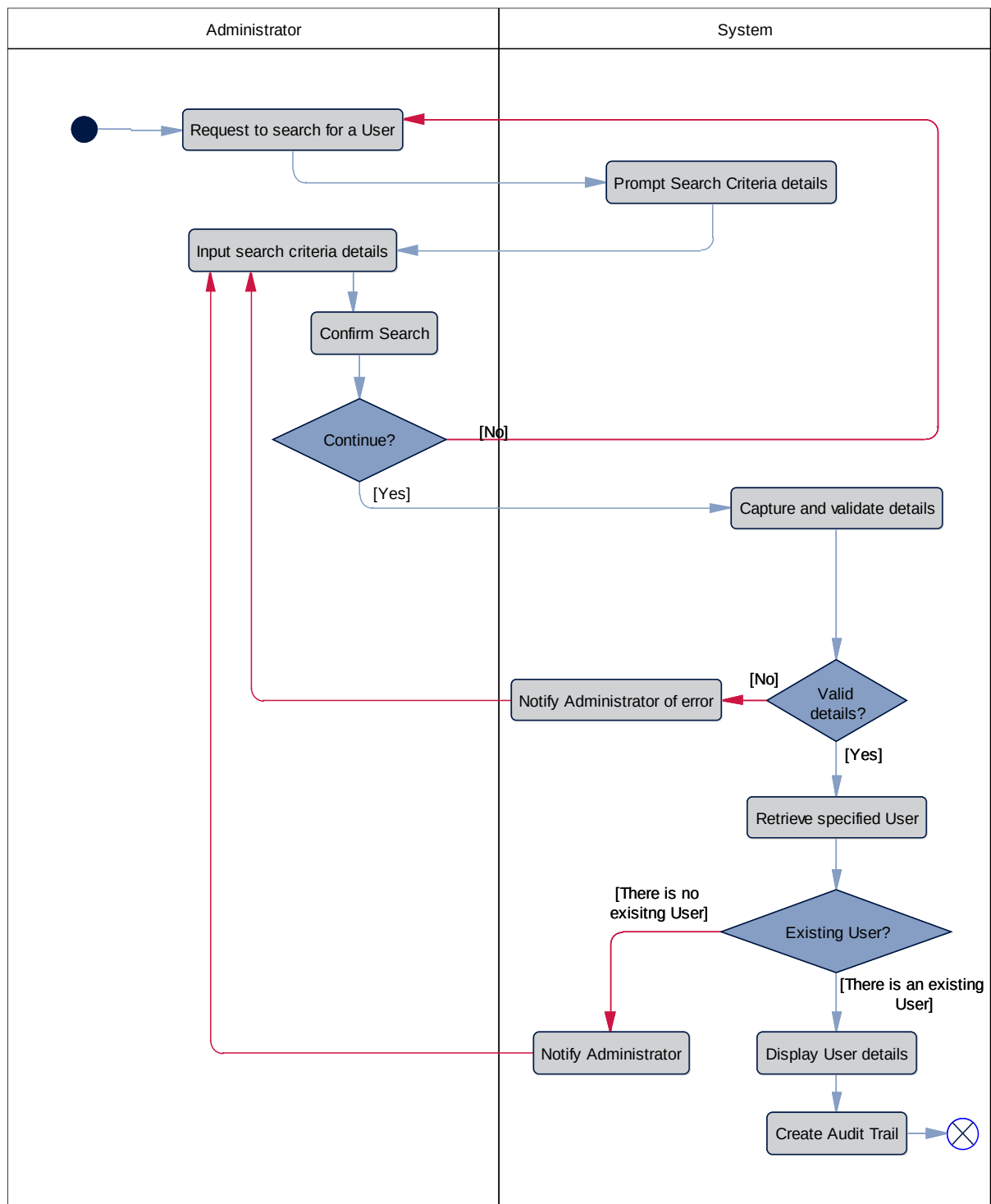


Figure 211 Search User Activity Diagram



4.2.1.6 Update User

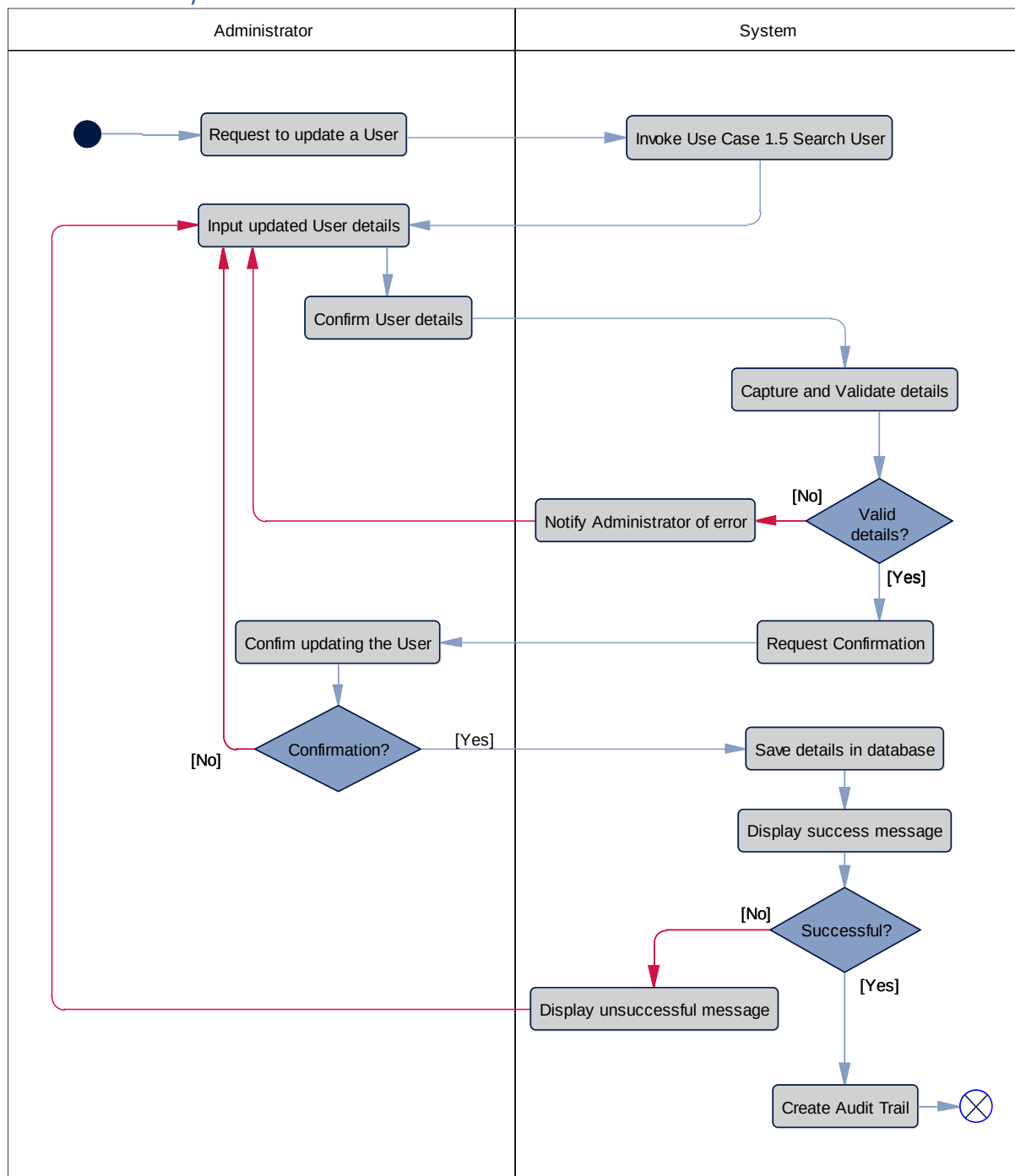


Figure 212 Update User Activity Diagram



4.2.1.7 Delete User

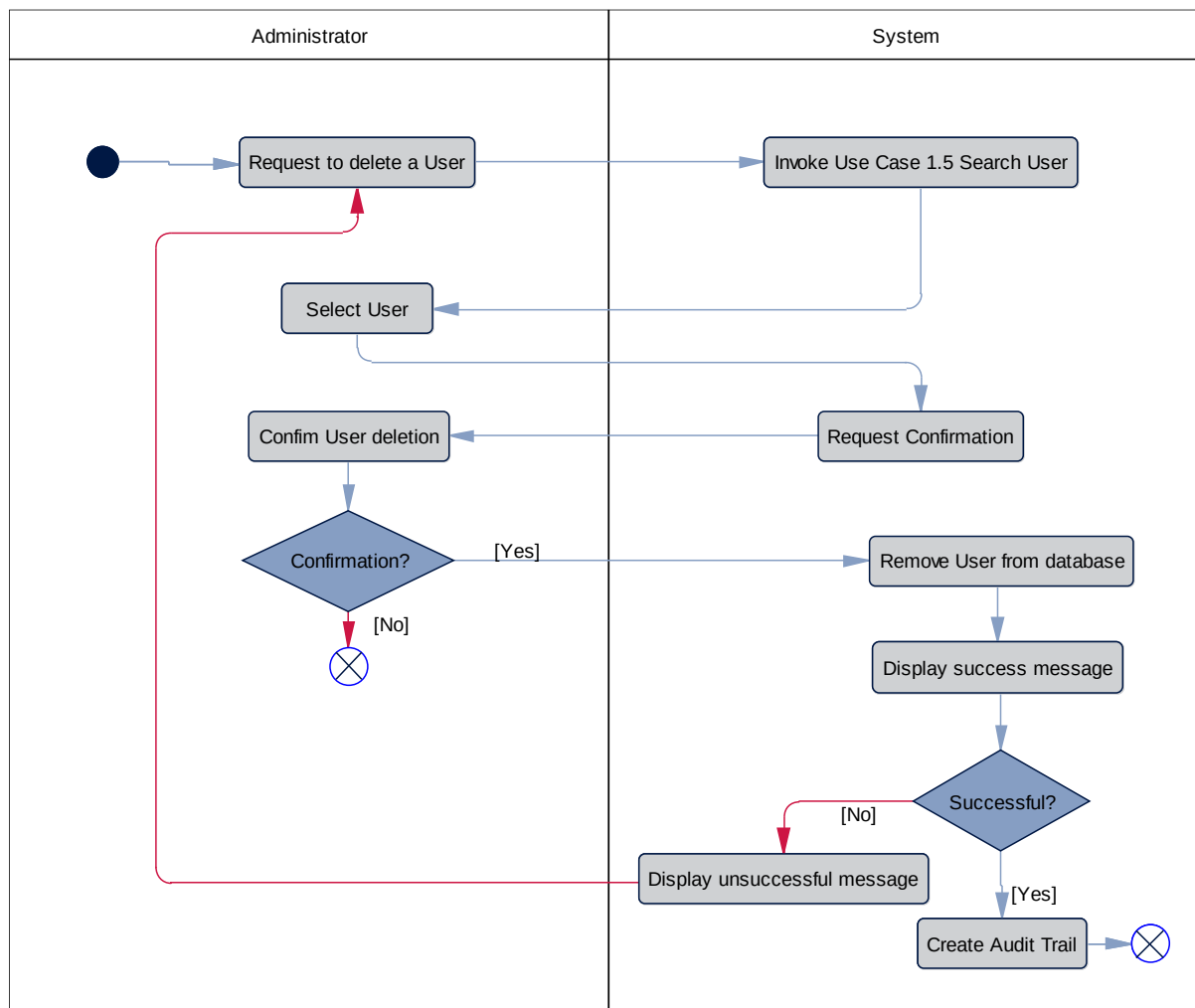


Figure 213 Delete User Activity Diagram



4.2.1.8 Add User Type

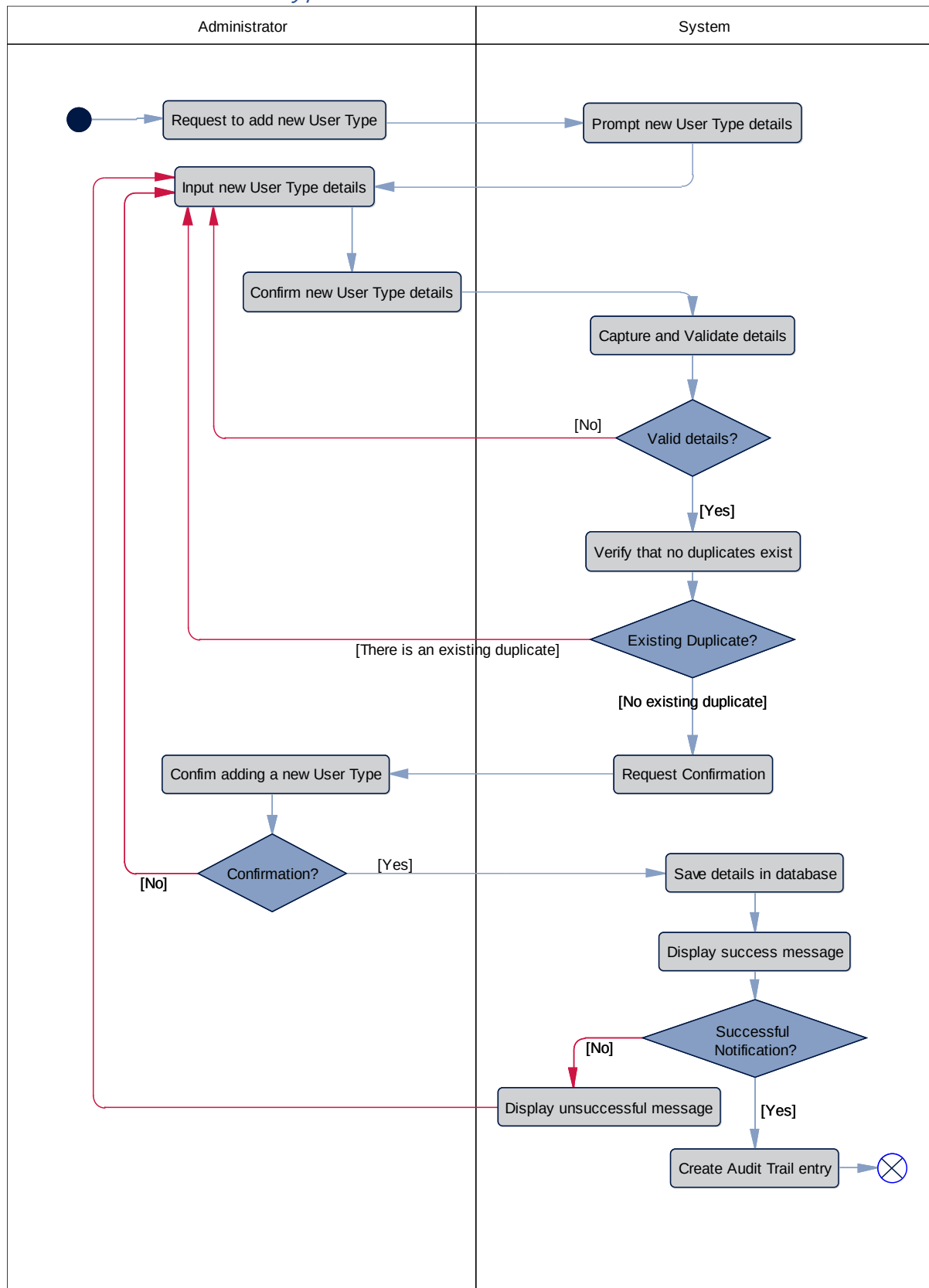


Figure 214 Add User Type Activity Diagram



4.2.1.9 Search User Type

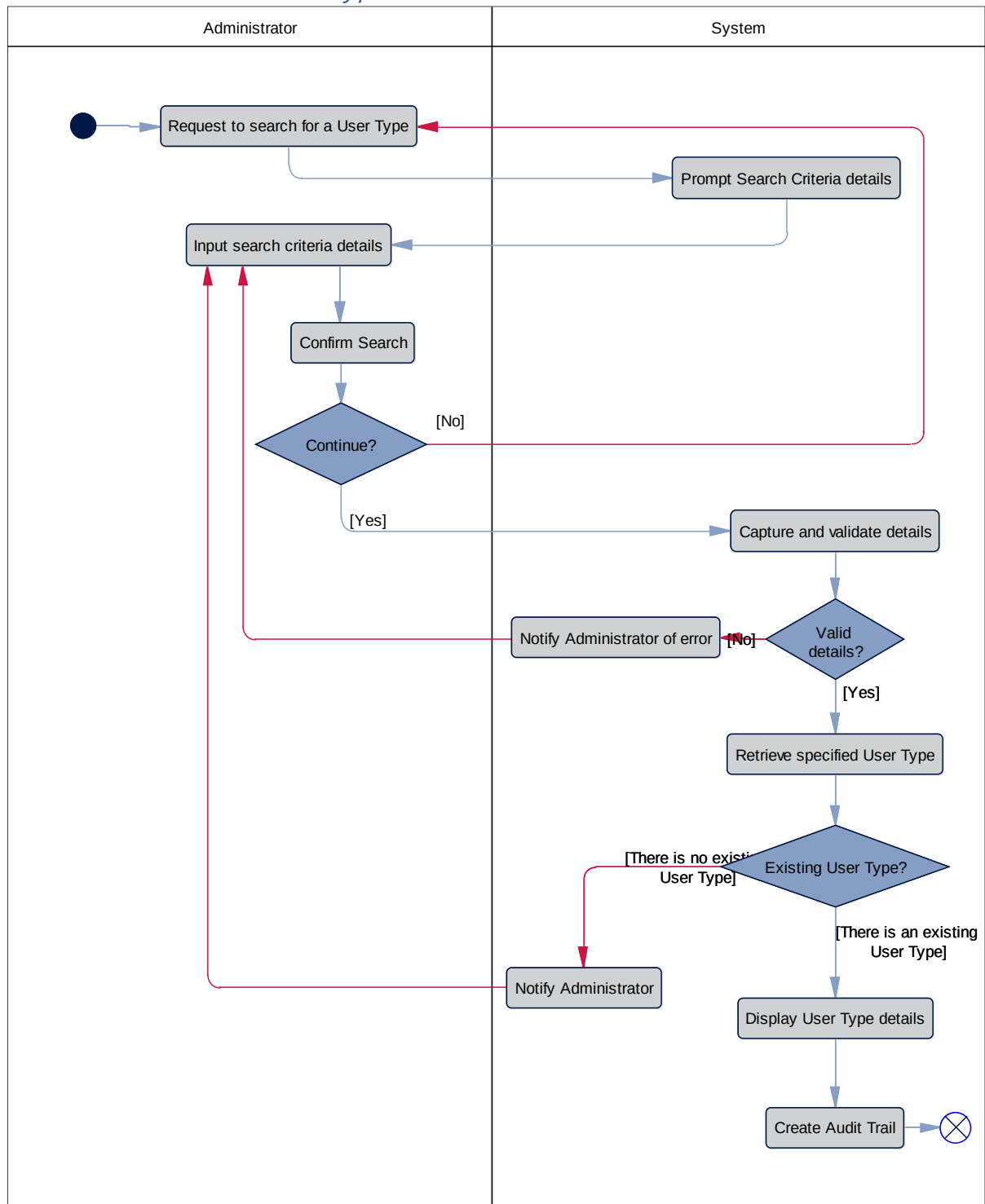


Figure 215 Search User Type Activity Diagrams



4.2.1.10 Update User Type

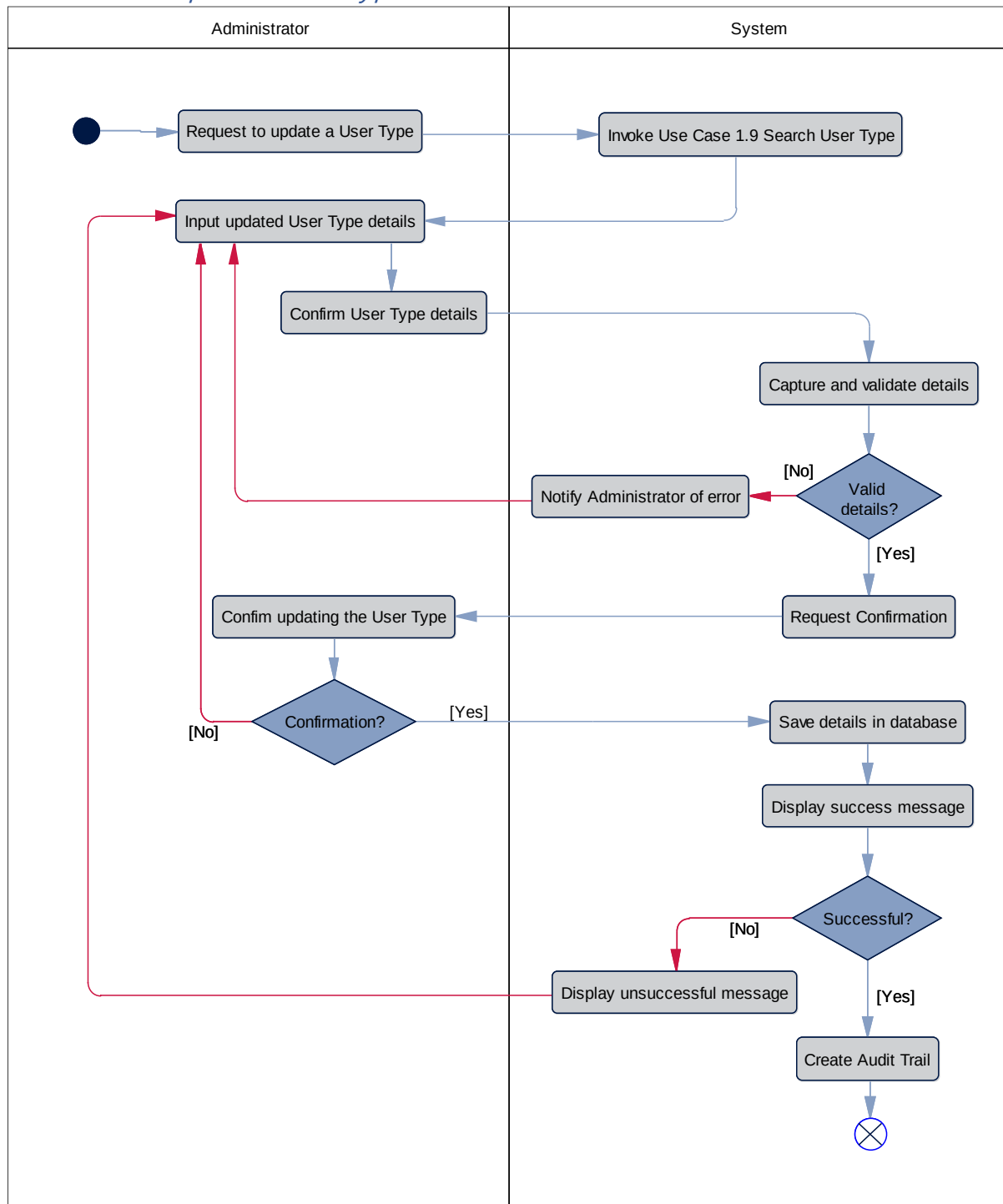


Figure 216 Update User Type Activity Diagram



4.2.1.11 Delete User Type

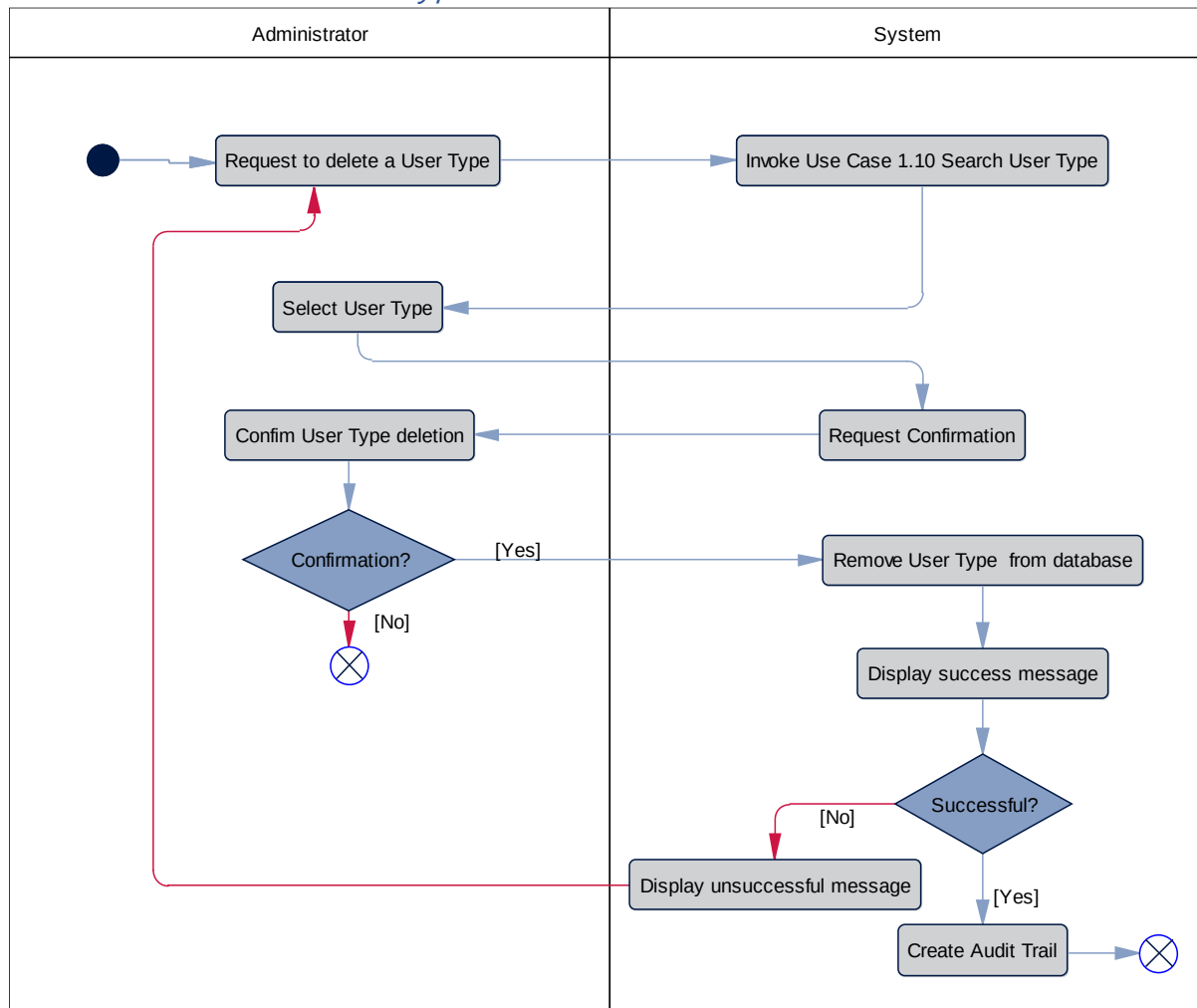


Figure 217 Delete User Type Activity Diagram



4.2.2 Sub-system 2: Mechanic Schedule

4.2.2.1 Complete Job

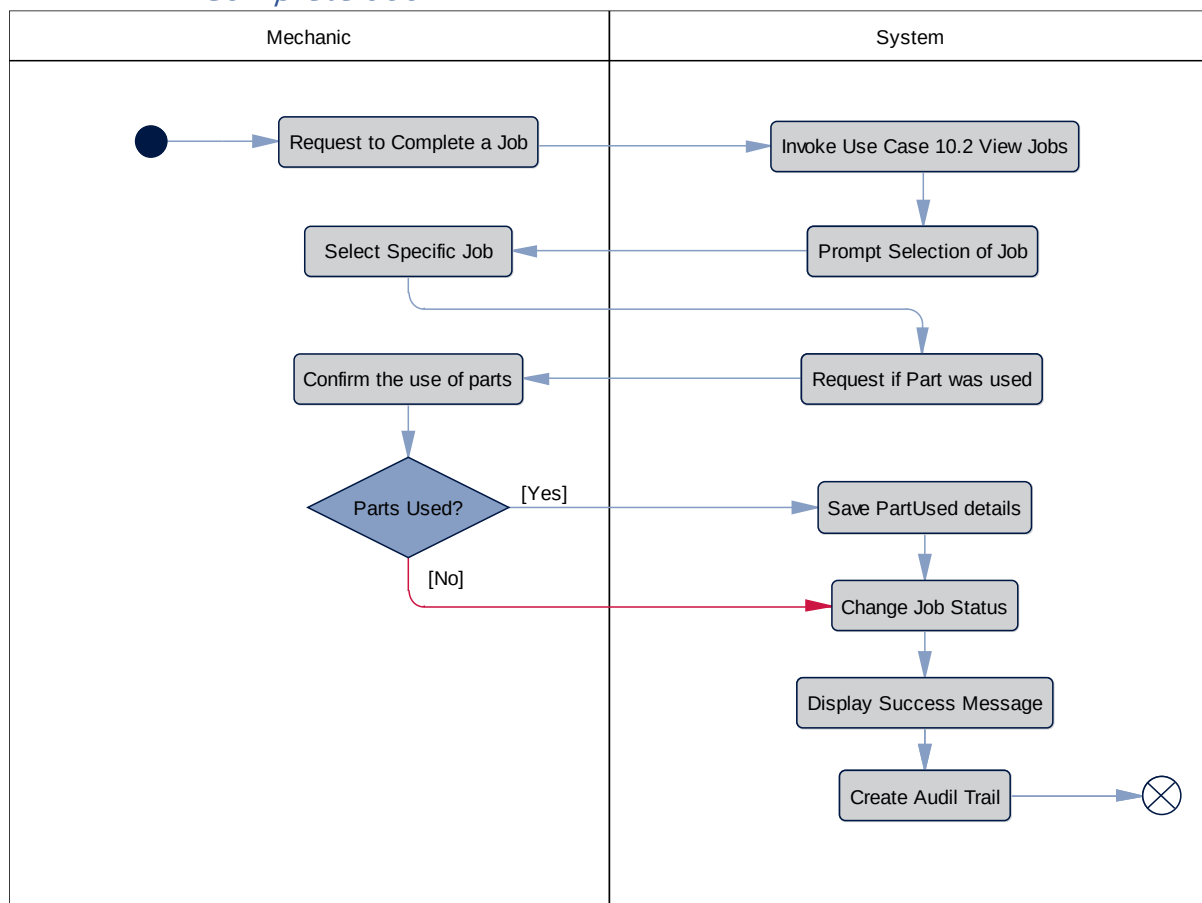


Figure 218 Complete Job Activity Diagram



4.2.2.2 View Job Tasks

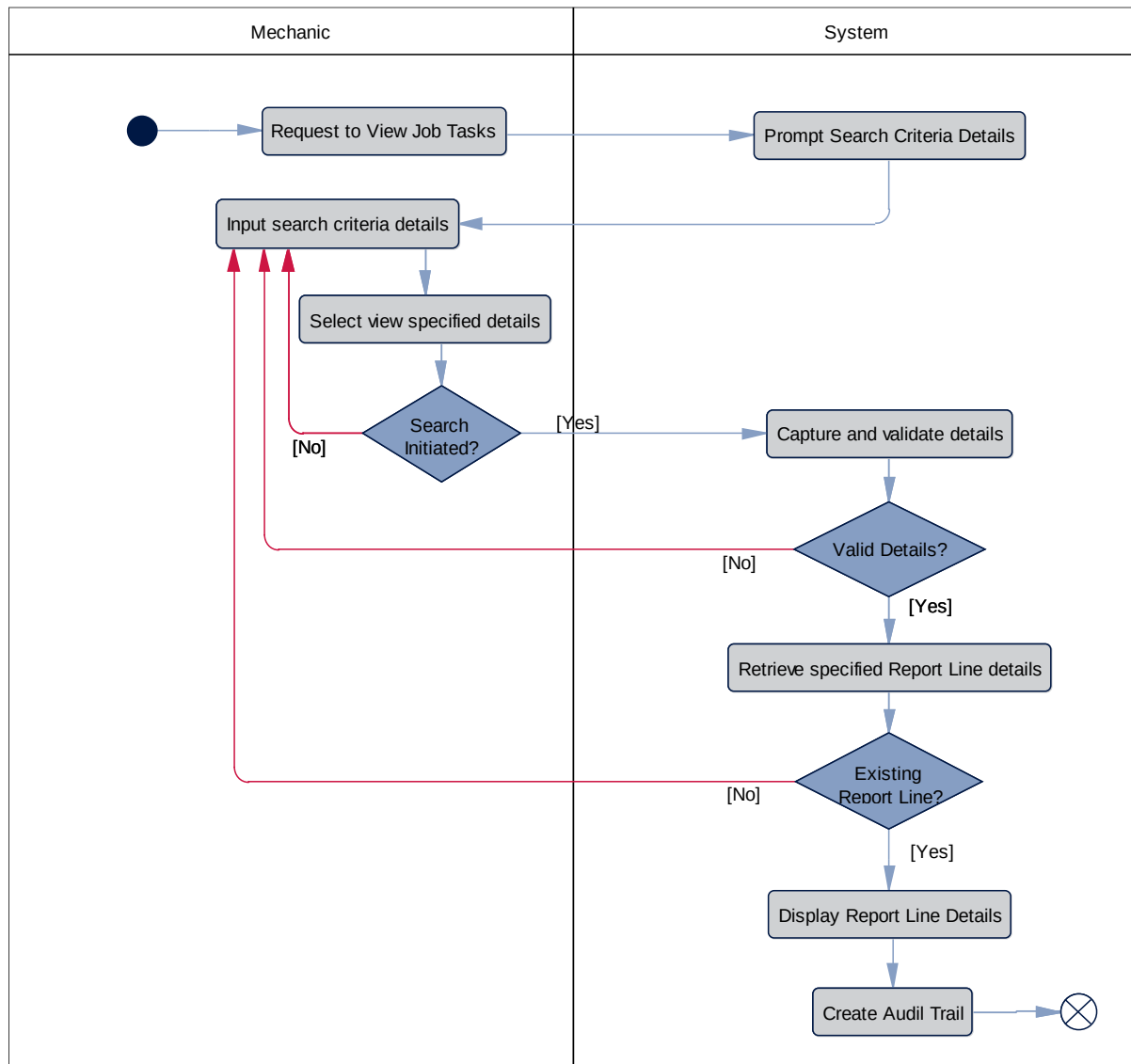


Figure 219 View Job Tasks Activity Diagram



4.2.3 Sub-system 3: Tracking Hardware

4.2.3.1 Add Antenna

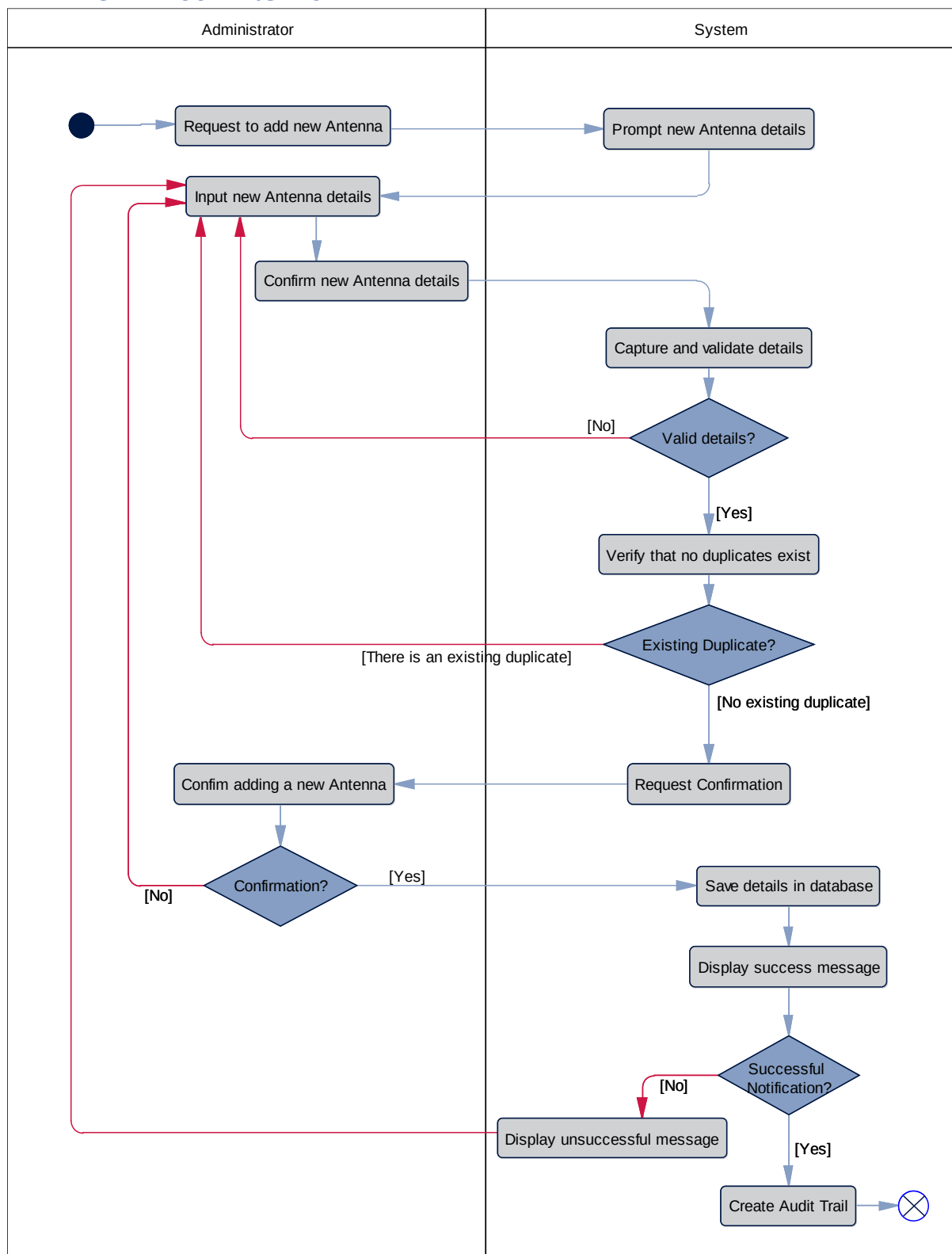


Figure 220 Add Antenna Activity Diagram



4.2.3.2 Search Antenna

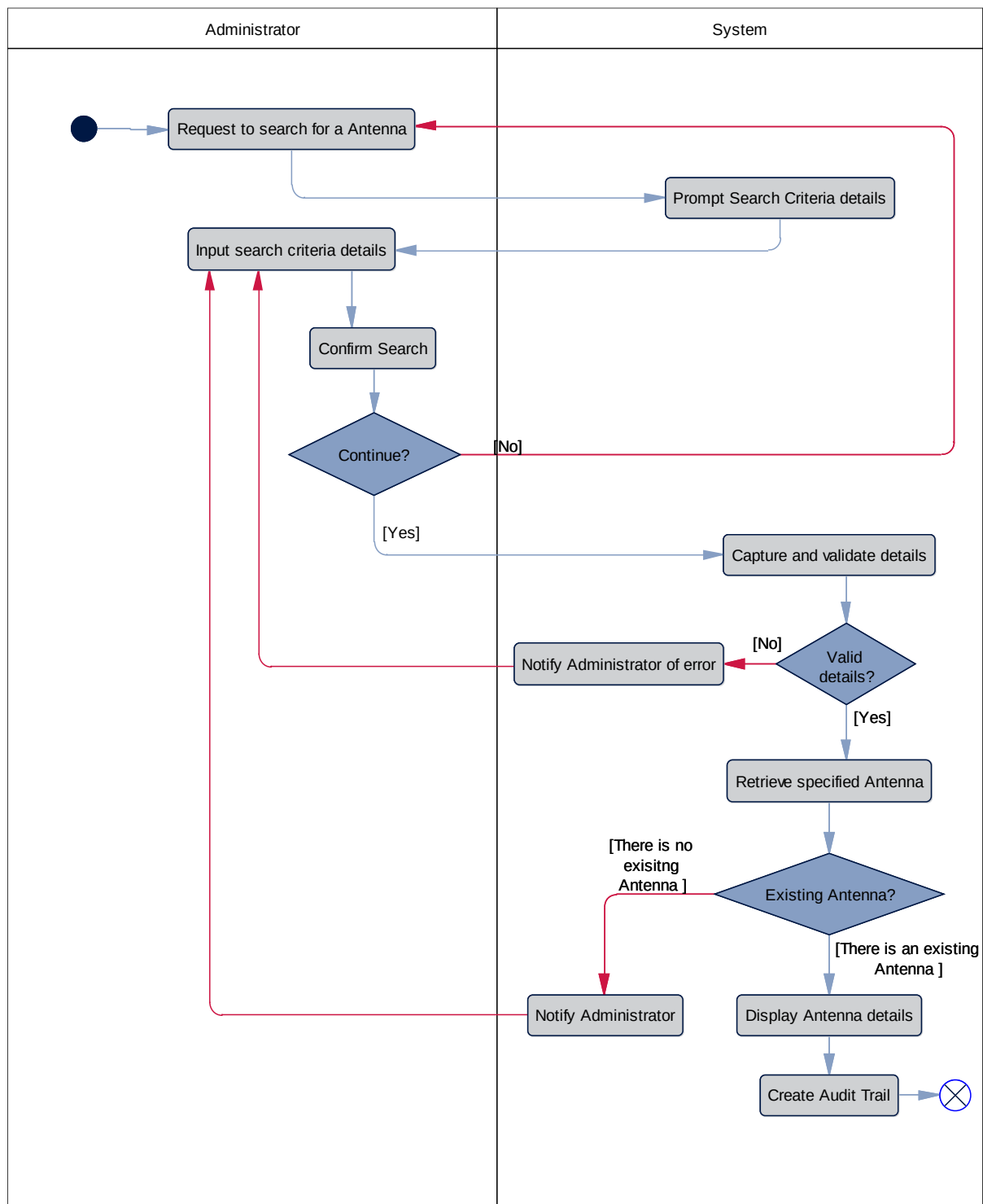


Figure 221 Search Antenna Activity Diagram



4.2.3.3 Update Antenna

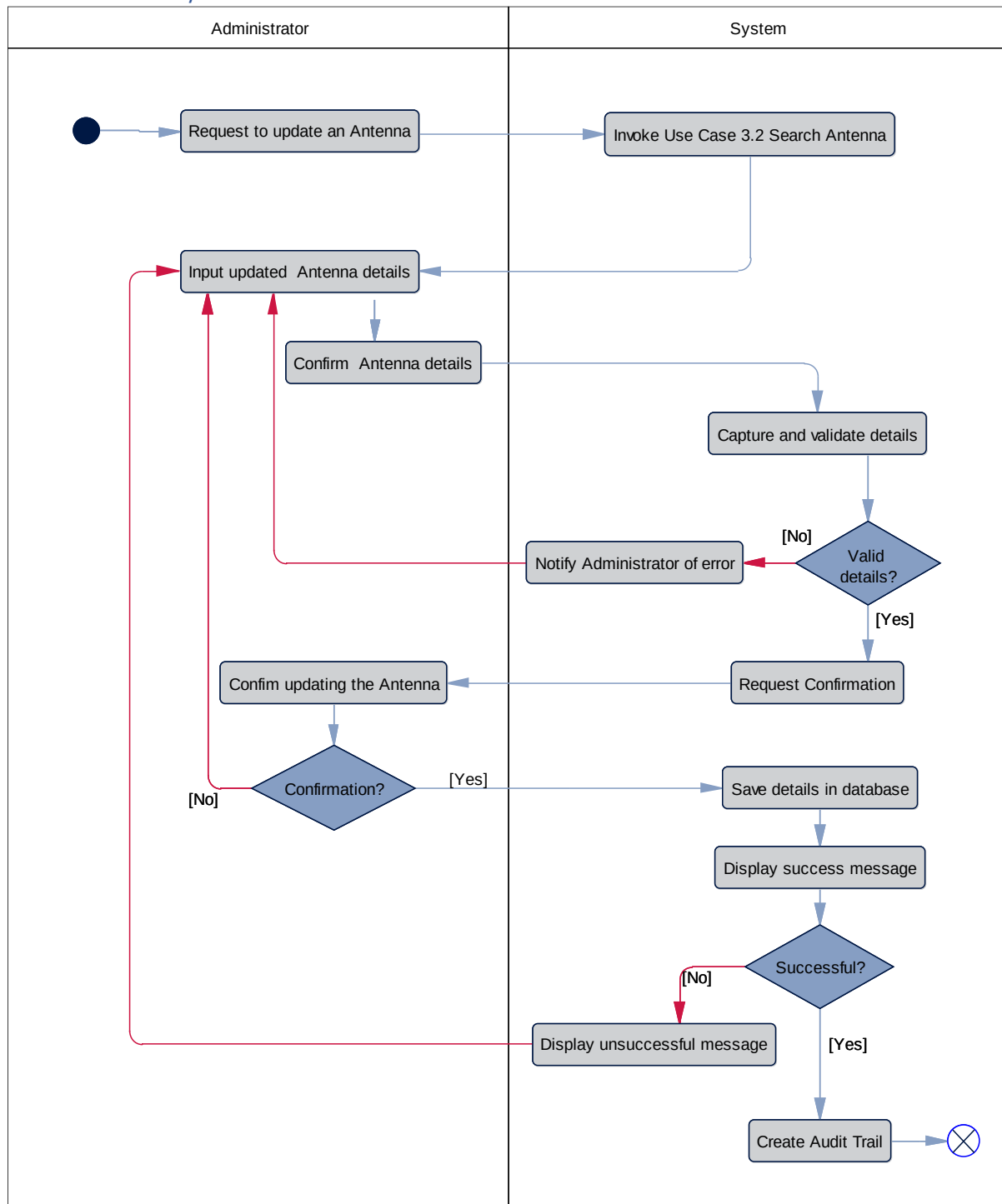


Figure 222 Update Antenna Activity Diagram



4.2.3.4 Delete Antenna

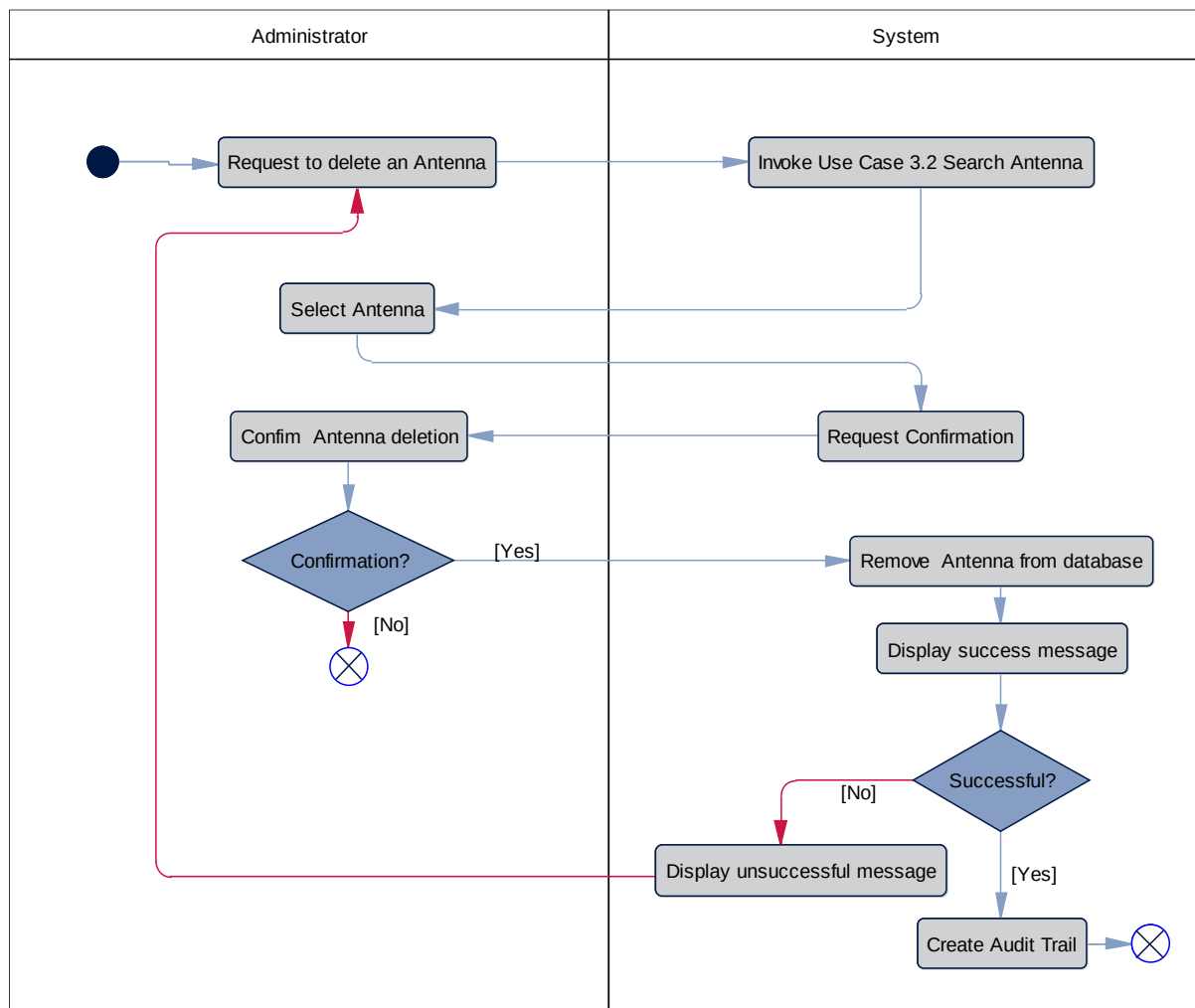


Figure 223 Delete Antenna Activity Diagram



4.2.3.5 Add Tag

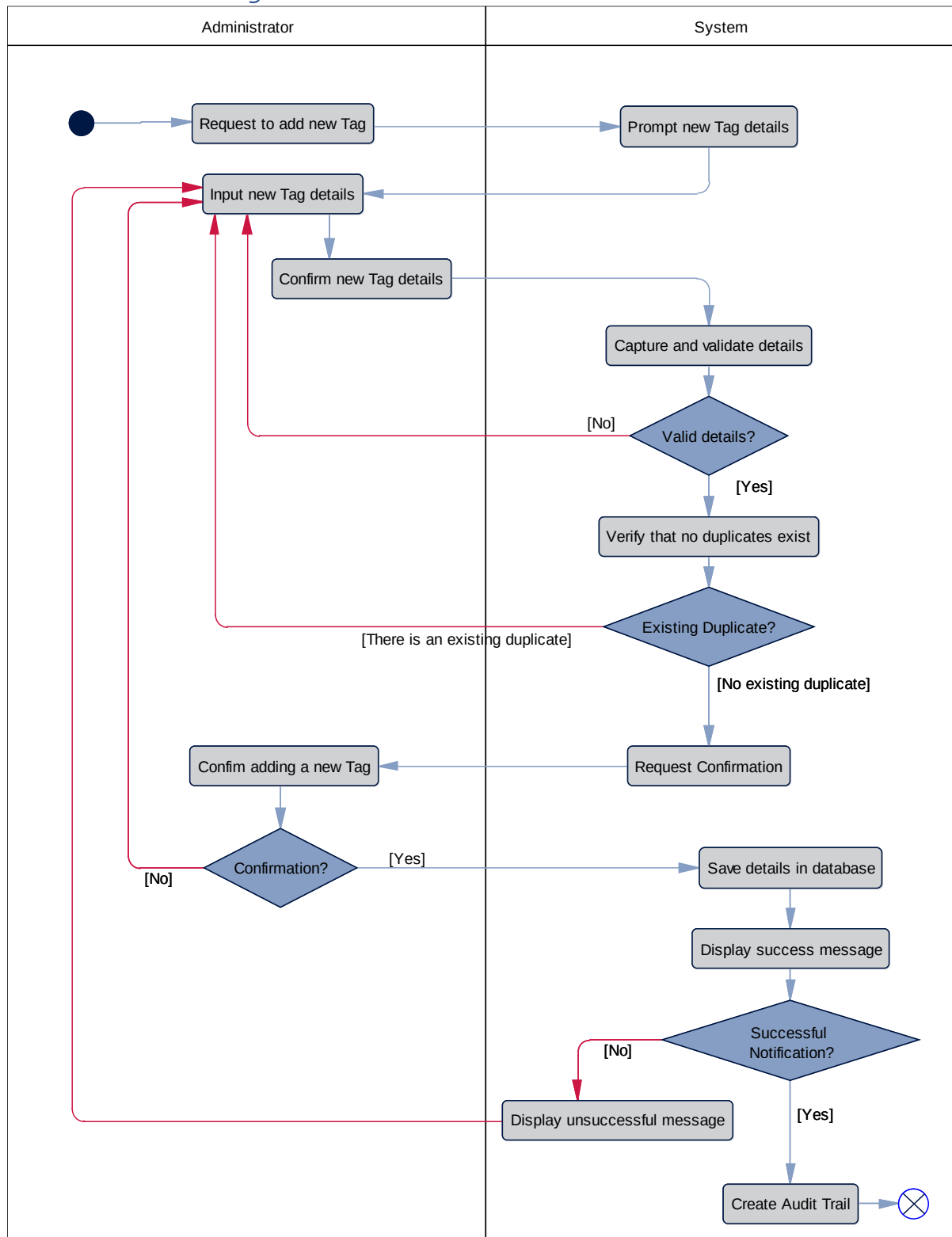


Figure 224 Add Tag Activity Diagram



4.2.3.6 Search Tag

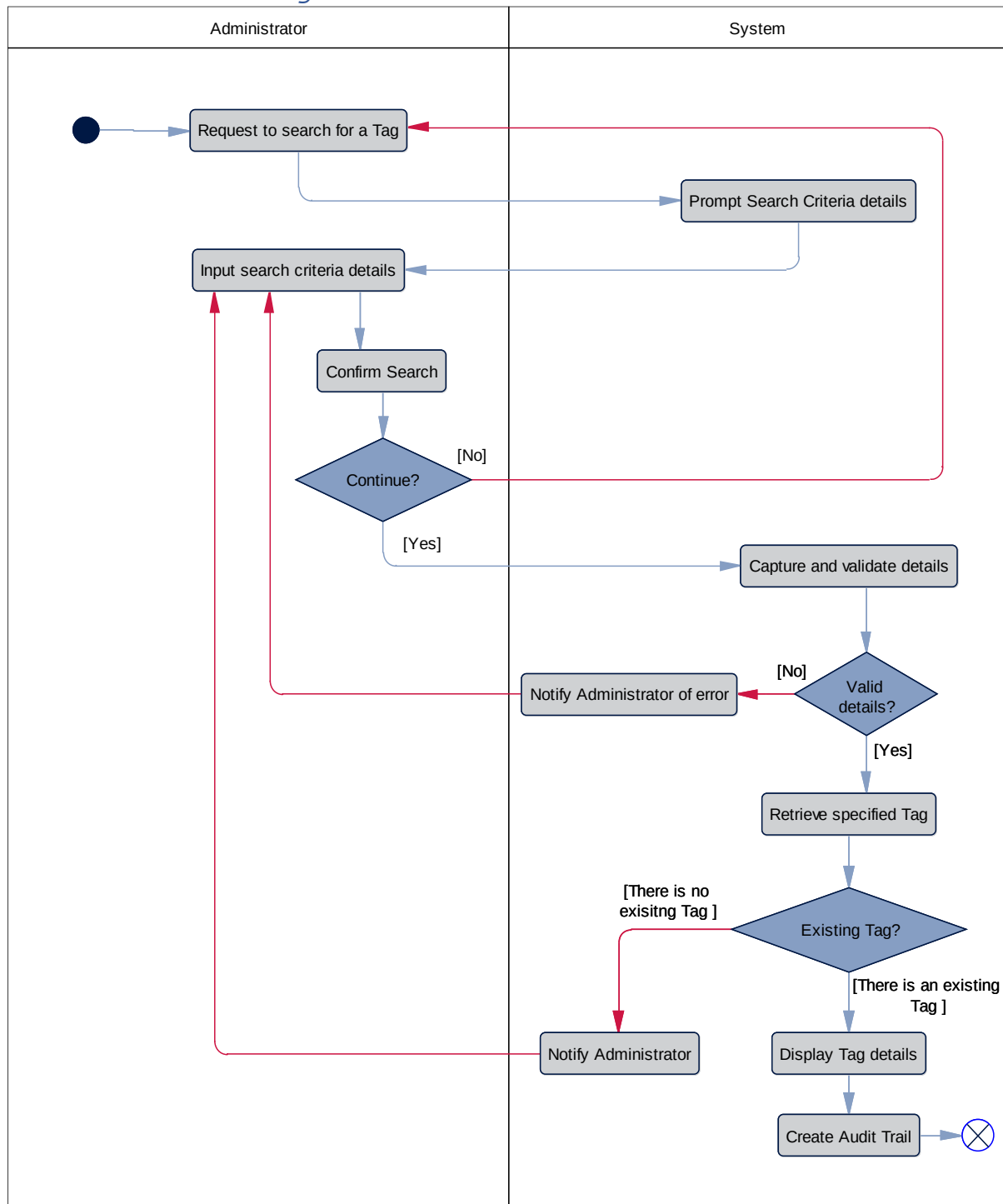


Figure 225 Search Tag Activity Diagram



4.2.3.7 Update Tag

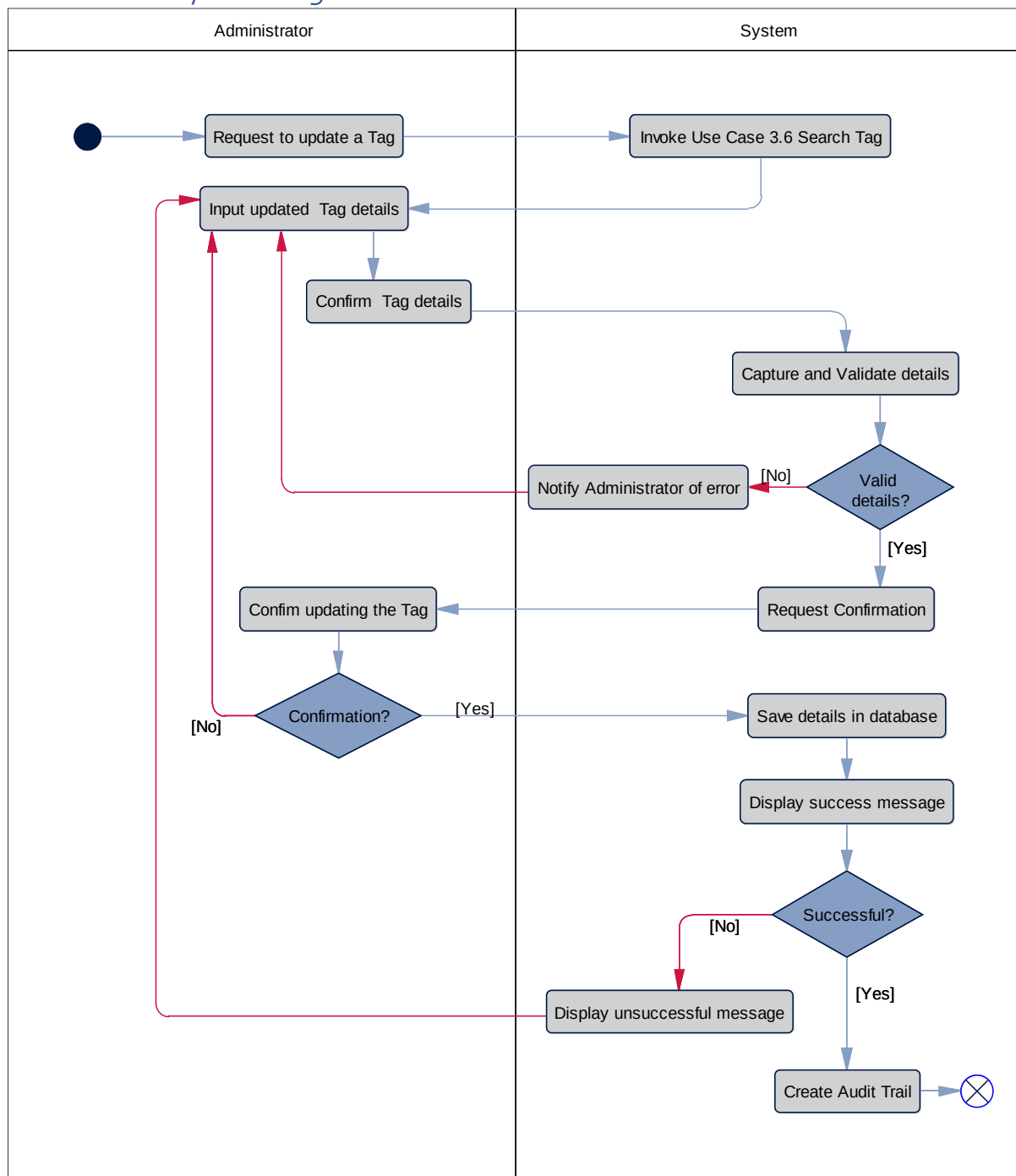


Figure 226 Update Tag Activity Diagram



4.2.3.8 Delete Tag

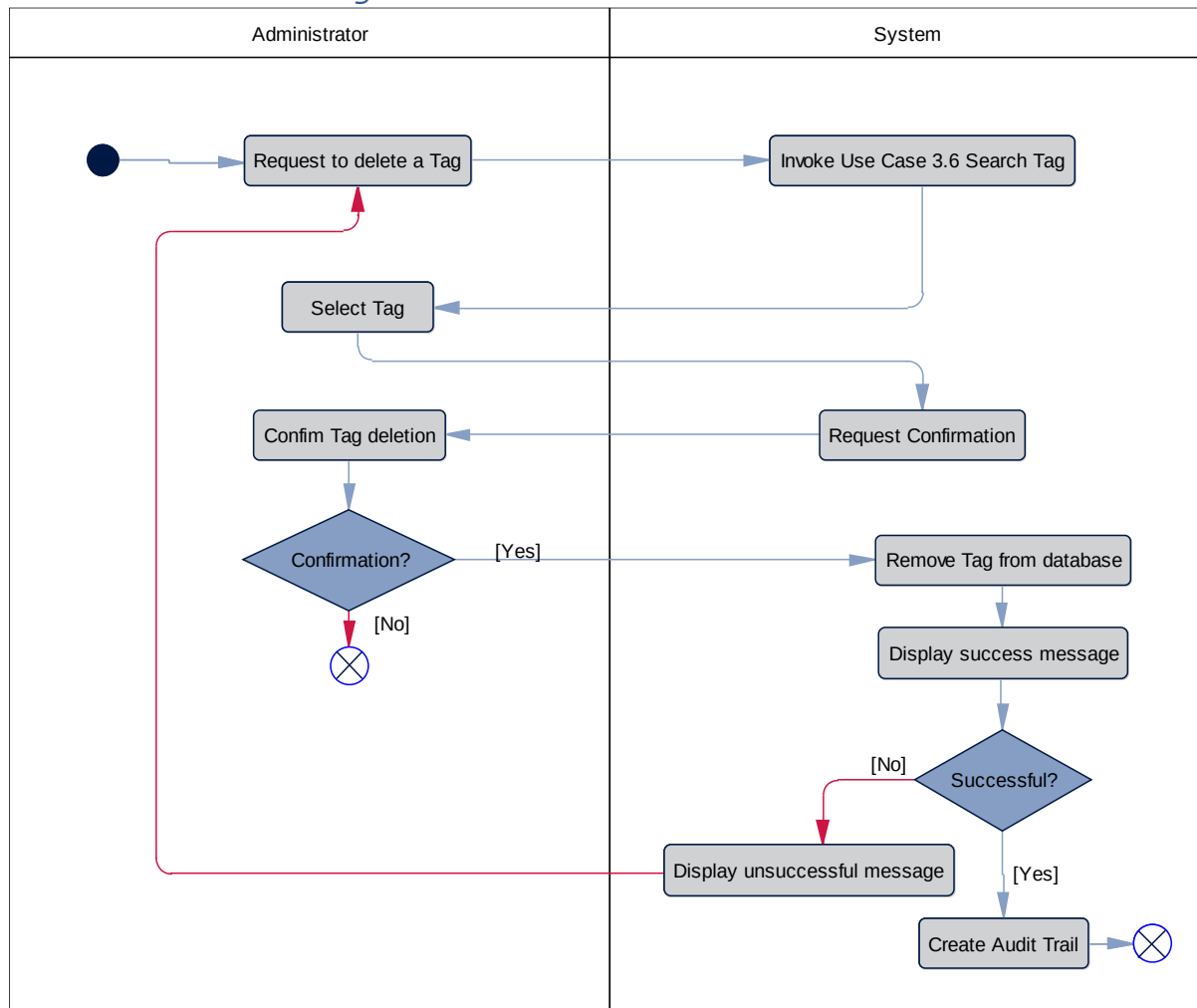


Figure 227 Delete Tag Activity Diagram



4.2.4 Sub-system 4: Student

4.2.4.1 Add Student

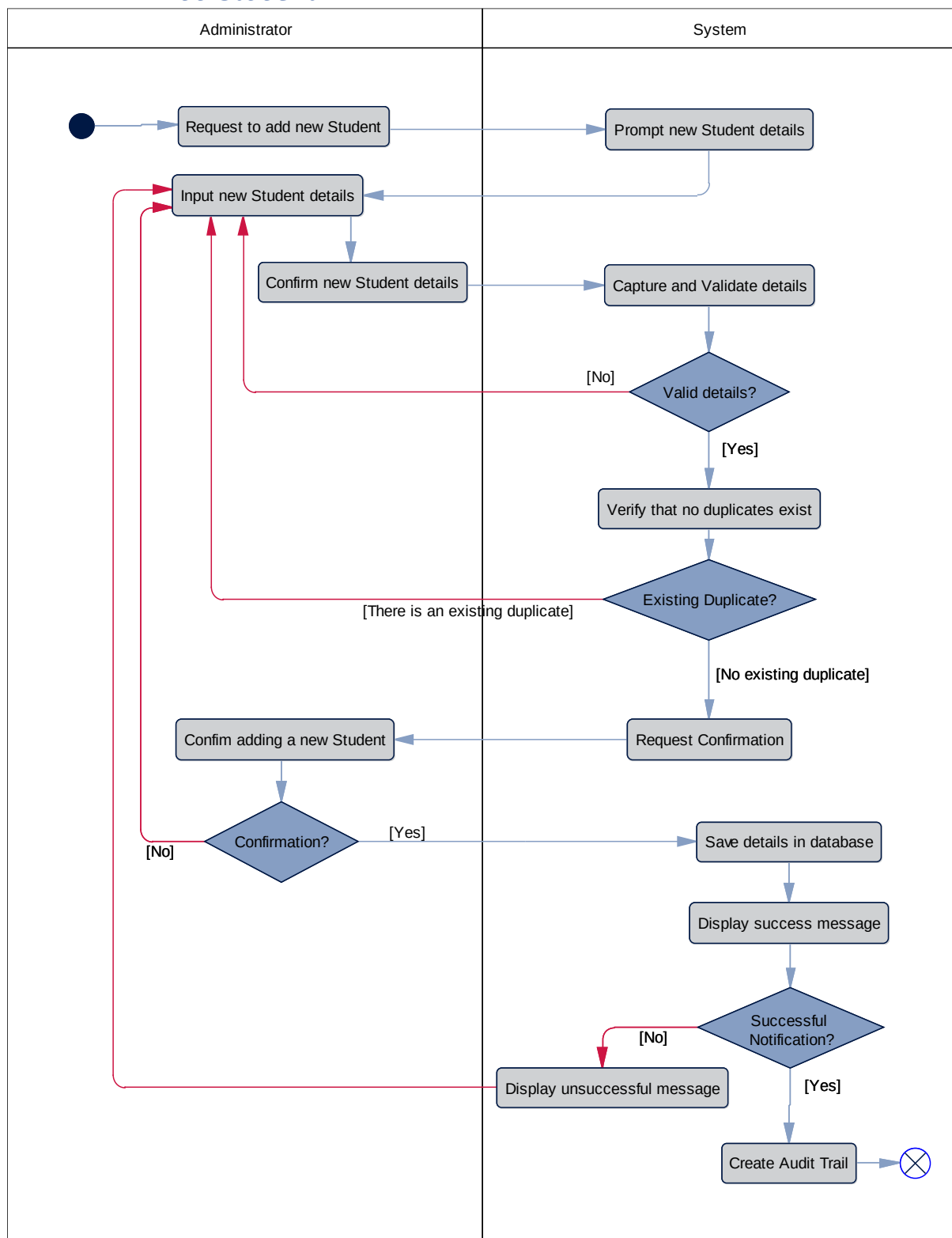


Figure 228 Add Student Activity Diagram



4.2.4.2 Search Student

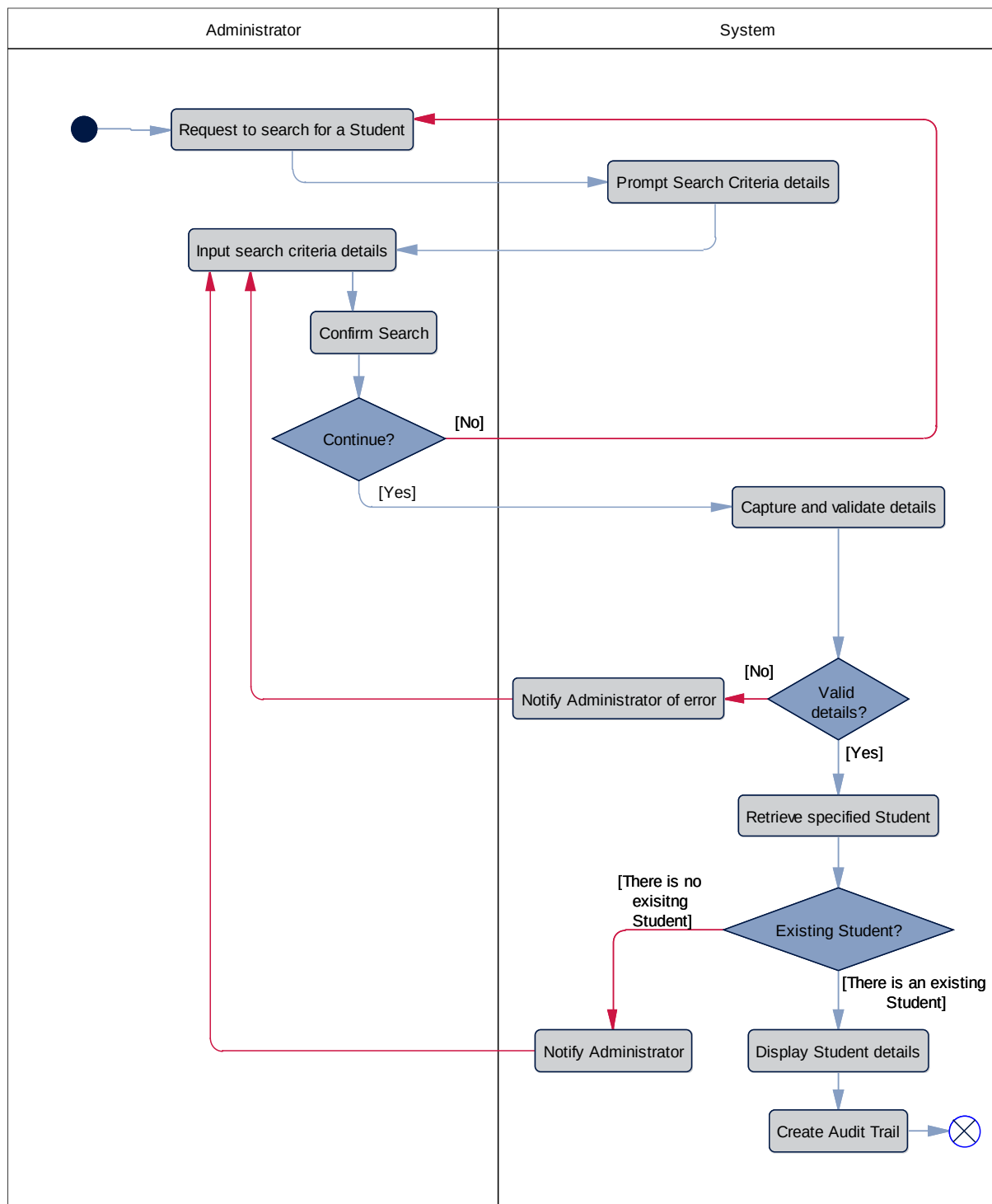


Figure 229 Search Student Activity Diagram



4.2.4.3 Update Student

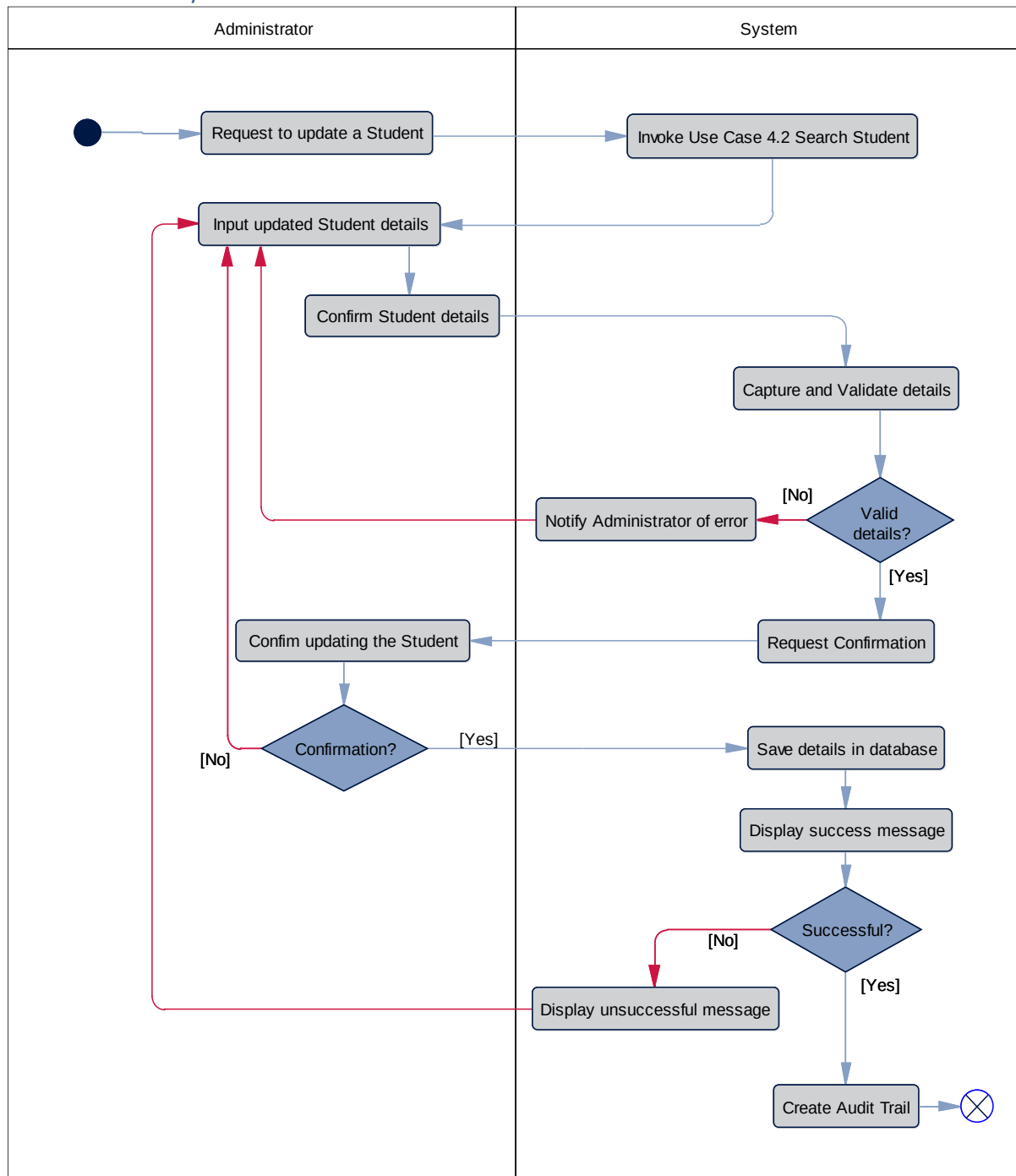


Figure 230 Update Student Activity Diagram



4.2.4.4 Delete Student

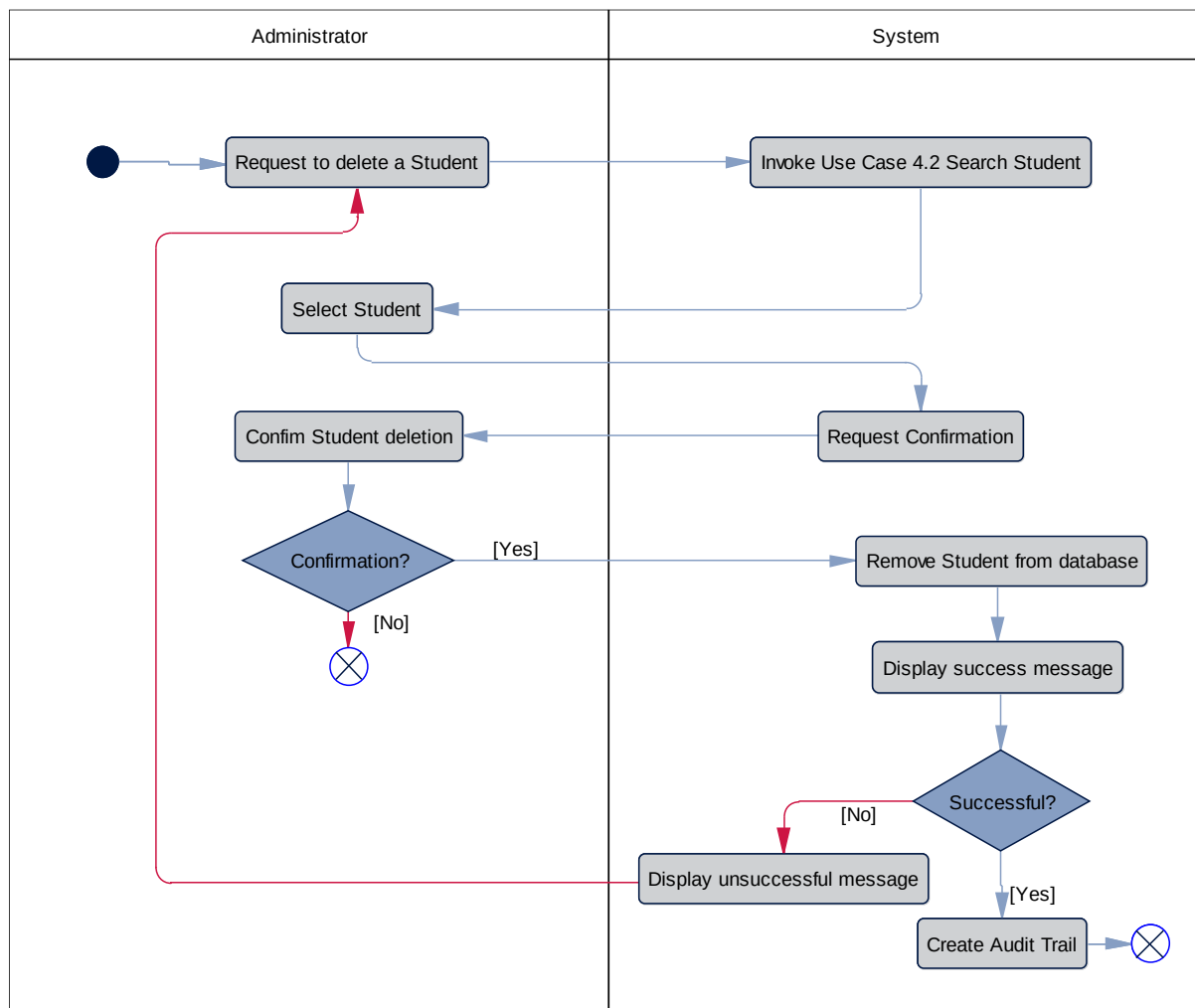


Figure 231 Delete Student Activity Diagram



4.2.4.5 Capture Student Marks

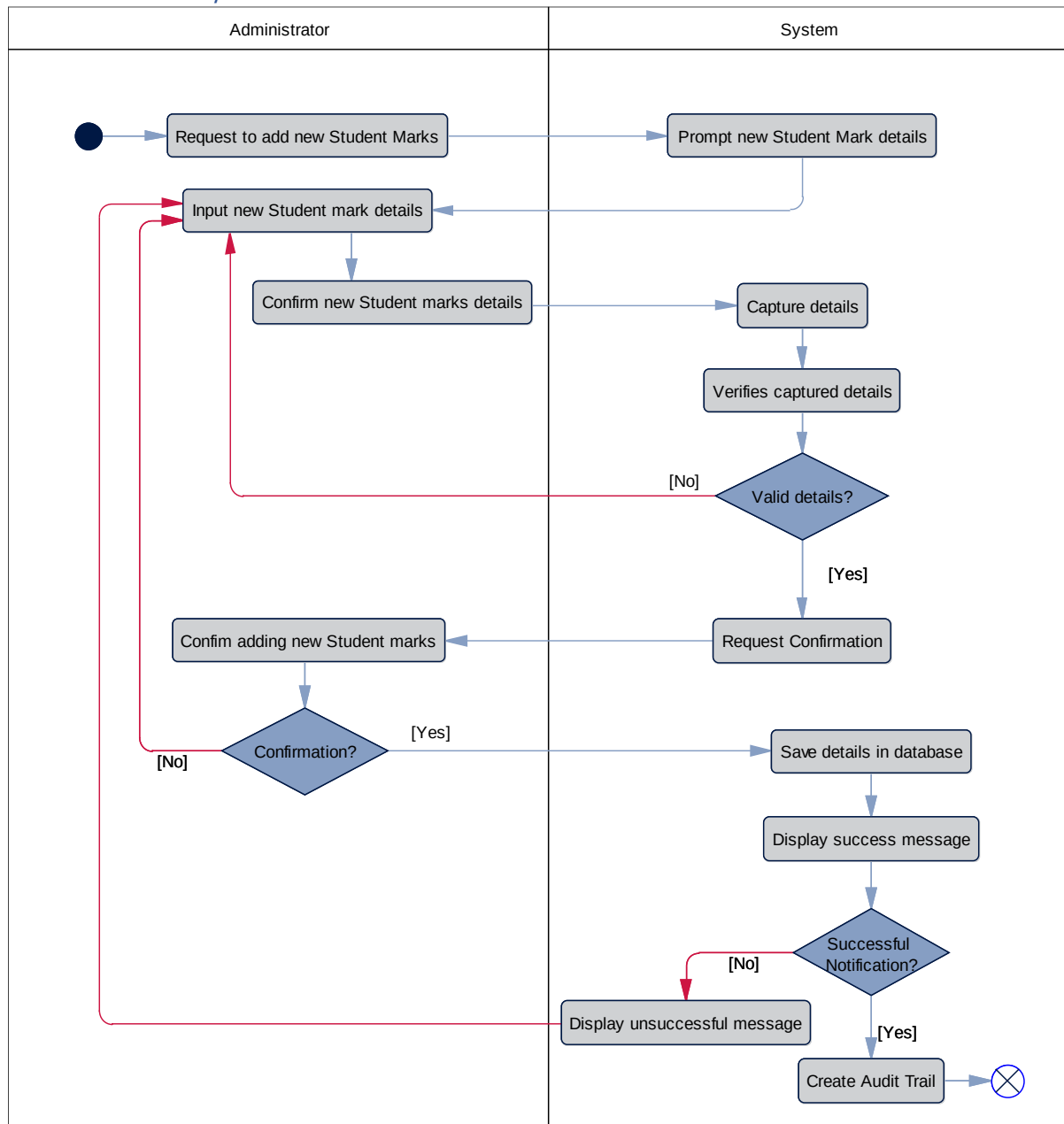


Figure 232 Capture Student Marks Activity Diagram



4.2.4.6 Search Student Marks

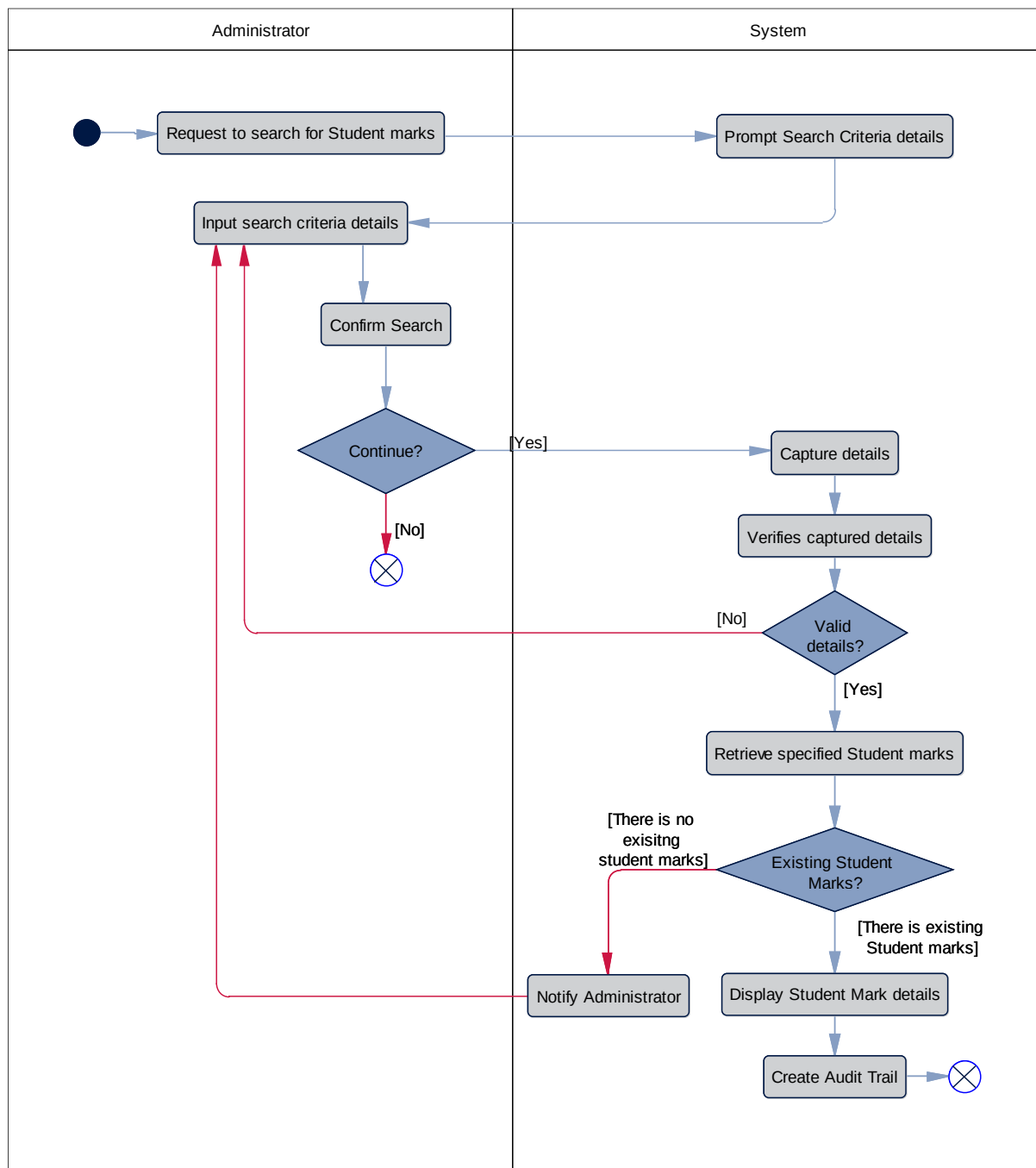


Figure 233 Search Student Marks Activity Diagram



4.2.4.7 Add Mark Type

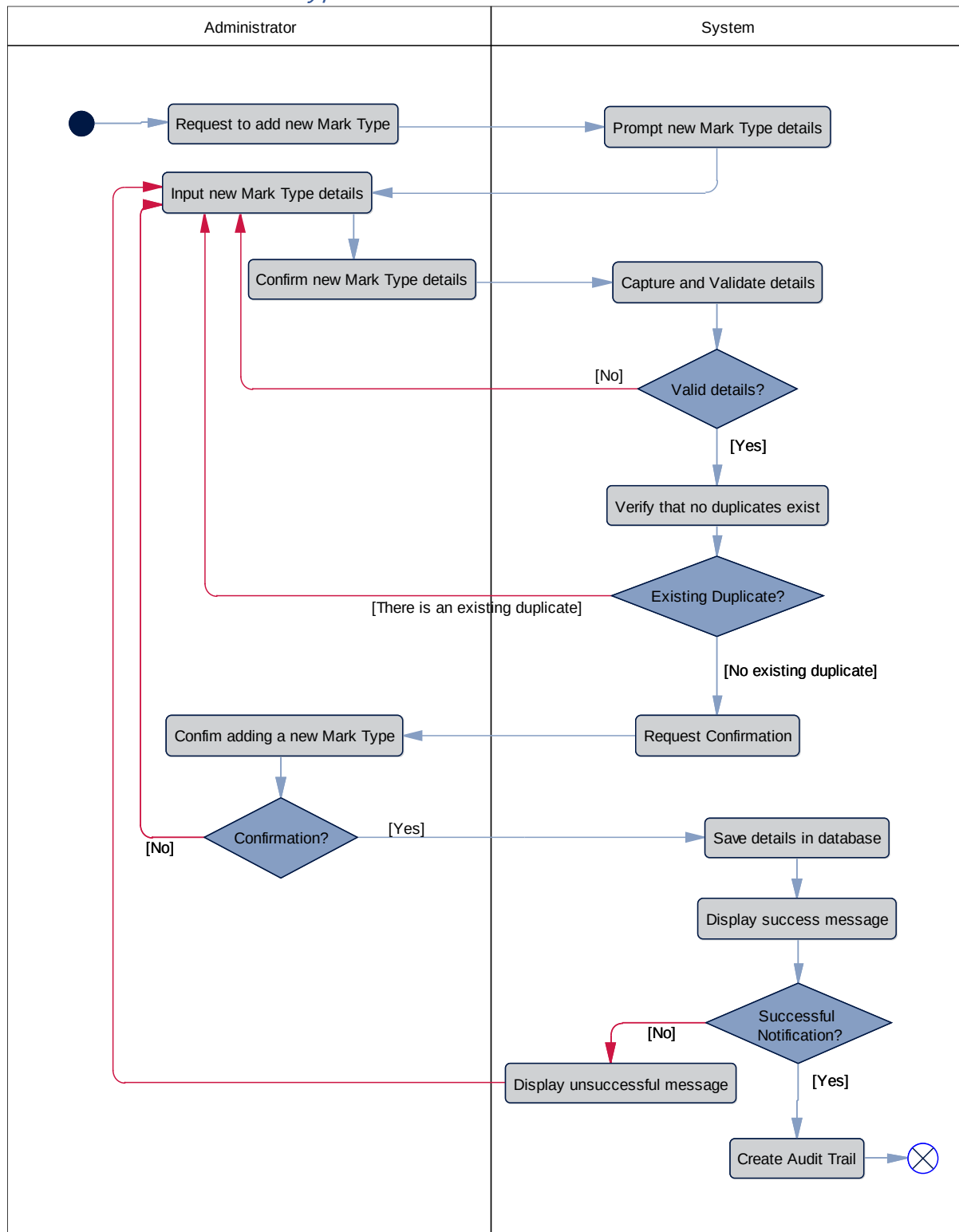


Figure 234 Add Mark Type Activity Diagram



4.2.4.8 Search Mark Type

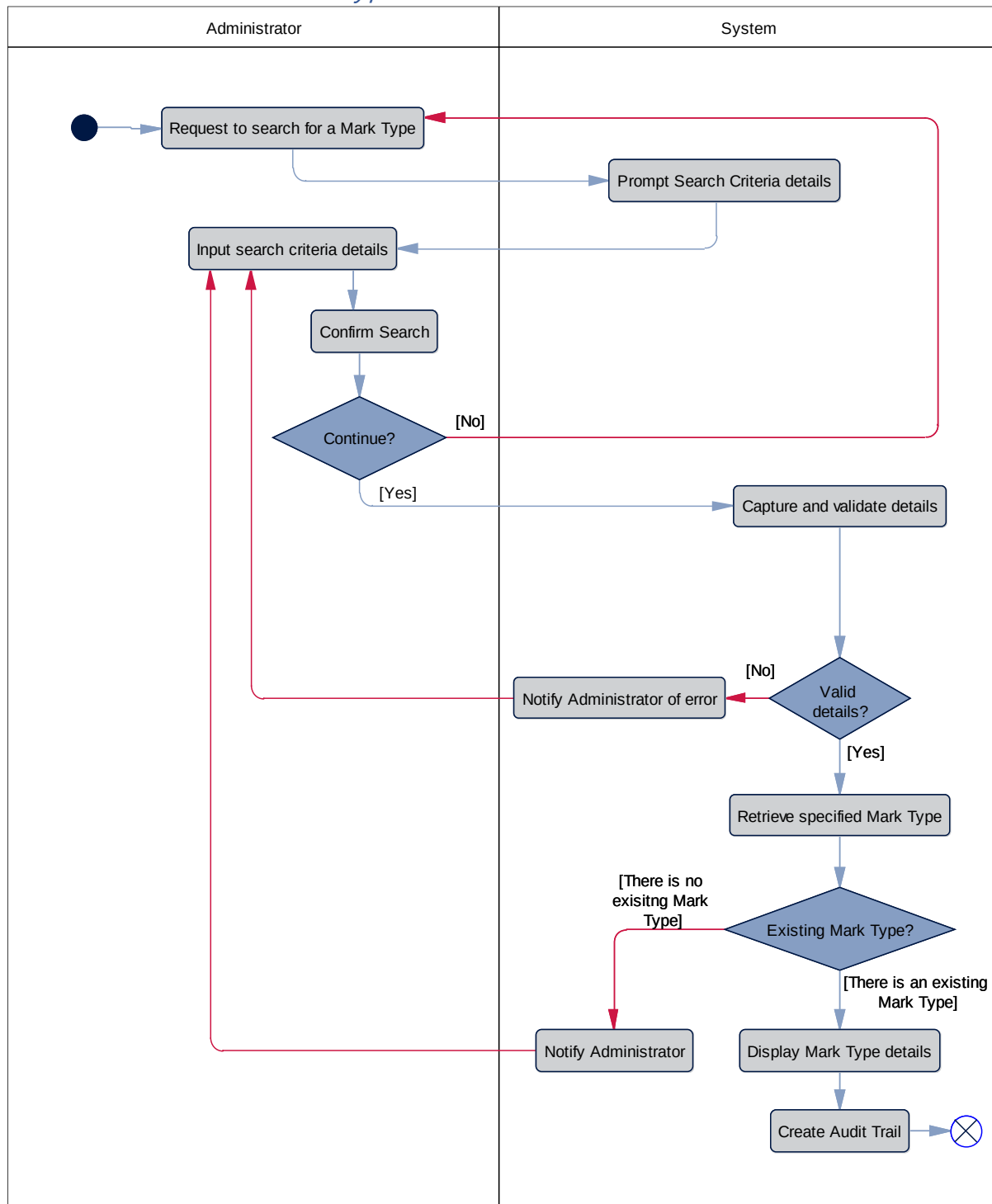


Figure 235 Search Mark Type Activity Diagram



4.2.4.9 Update Mark Type

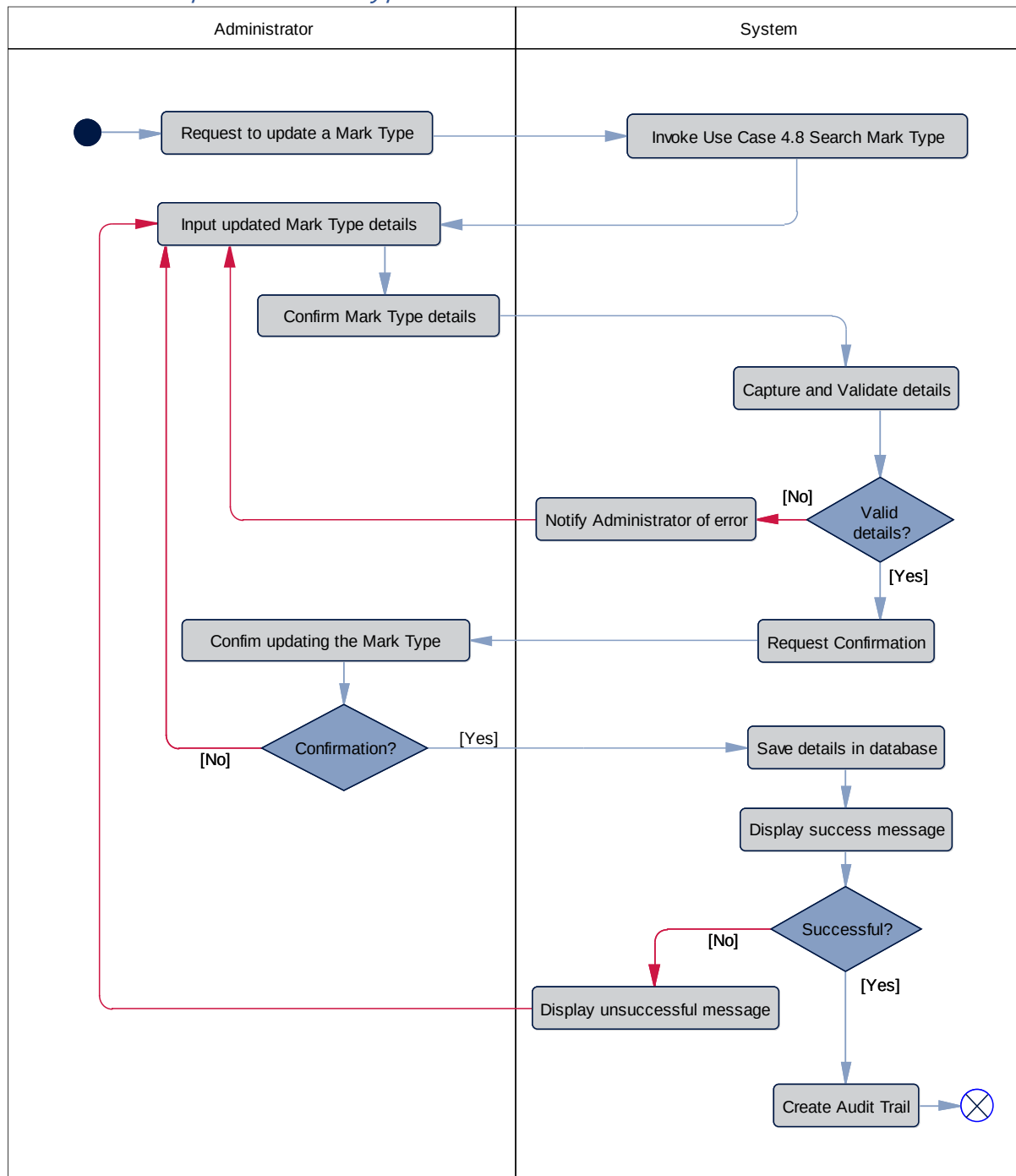


Figure 236 Update Mark Type Activity Diagram



4.2.4.10 Delete Mark Type

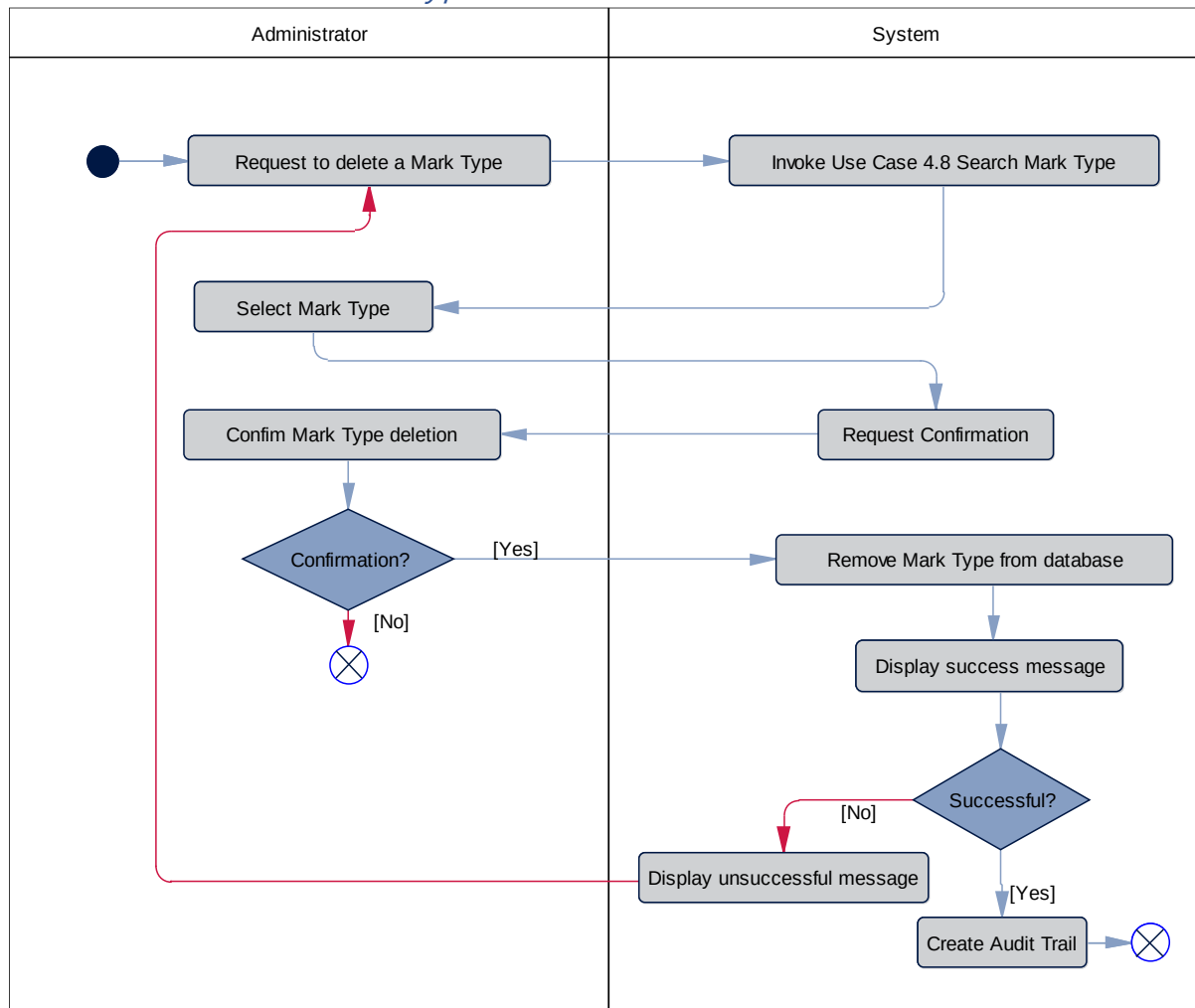


Figure 237 Delete Mark Type Activity Diagram



4.2.5 Sub-system 5: Bicycle

4.2.5.1 Add Bicycle

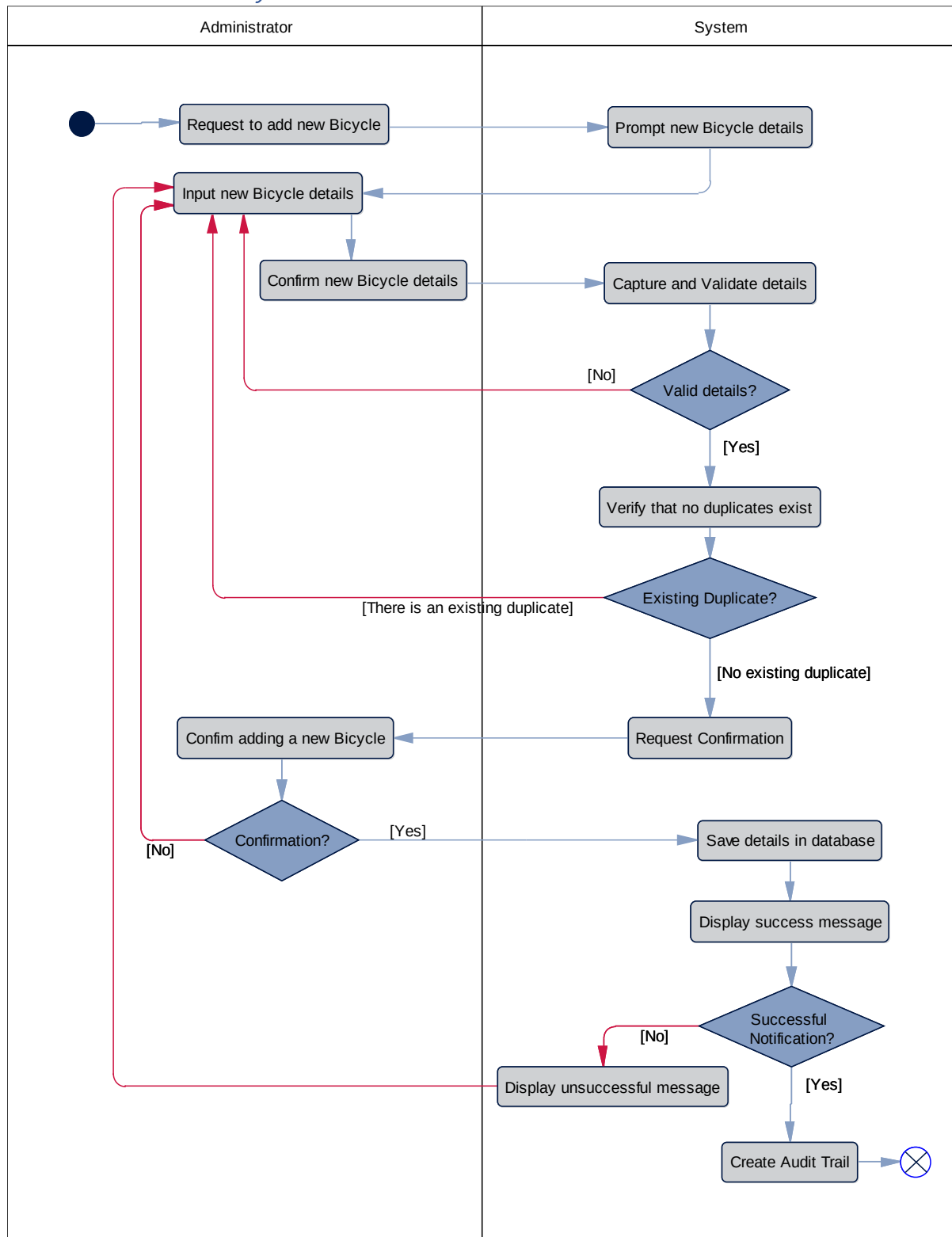


Figure 238 Add Bicycle Activity Diagram



4.2.5.2 Search Bicycle

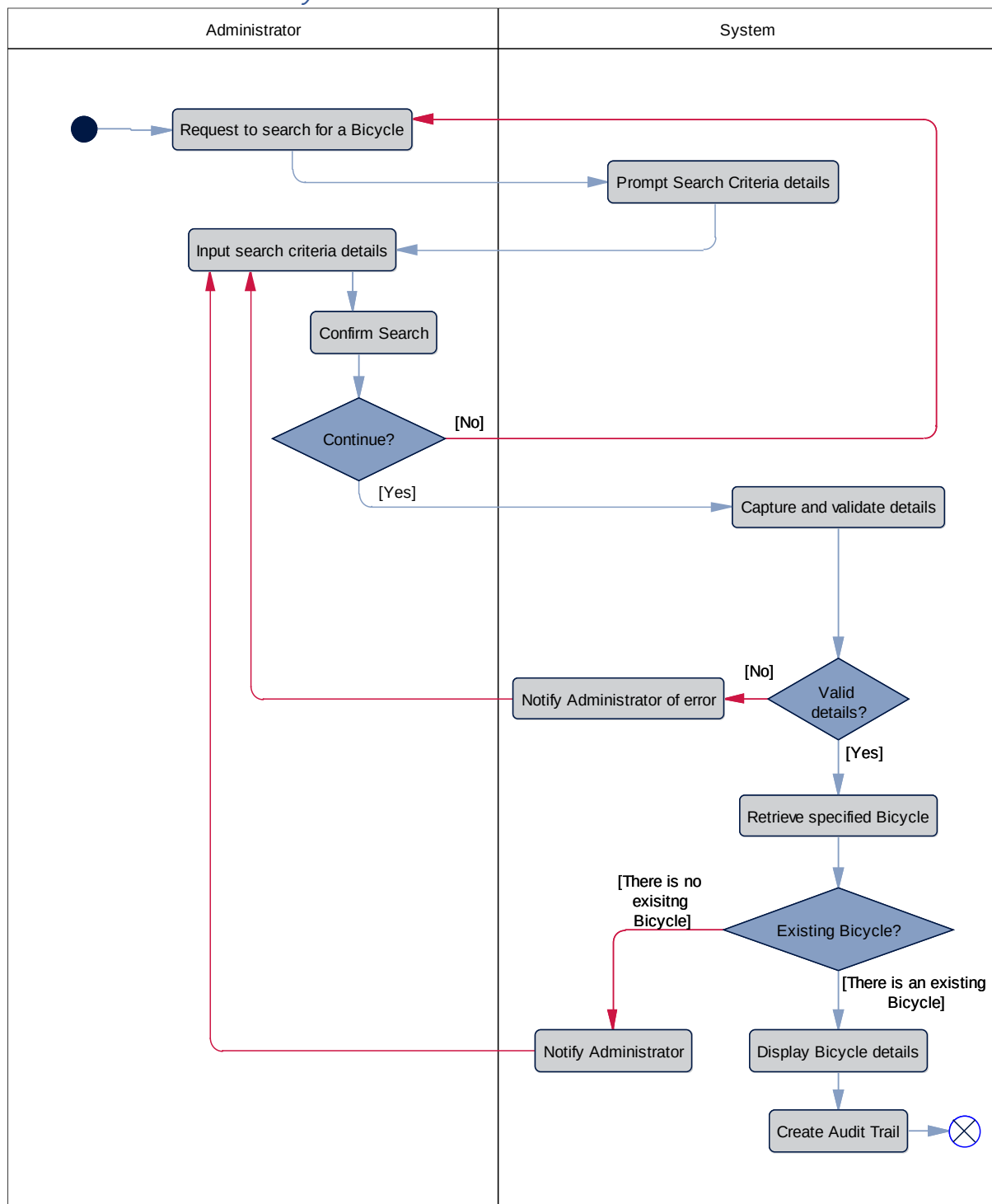


Figure 239 Search Bicycle Activity Diagram



4.2.5.3 Update Bicycle

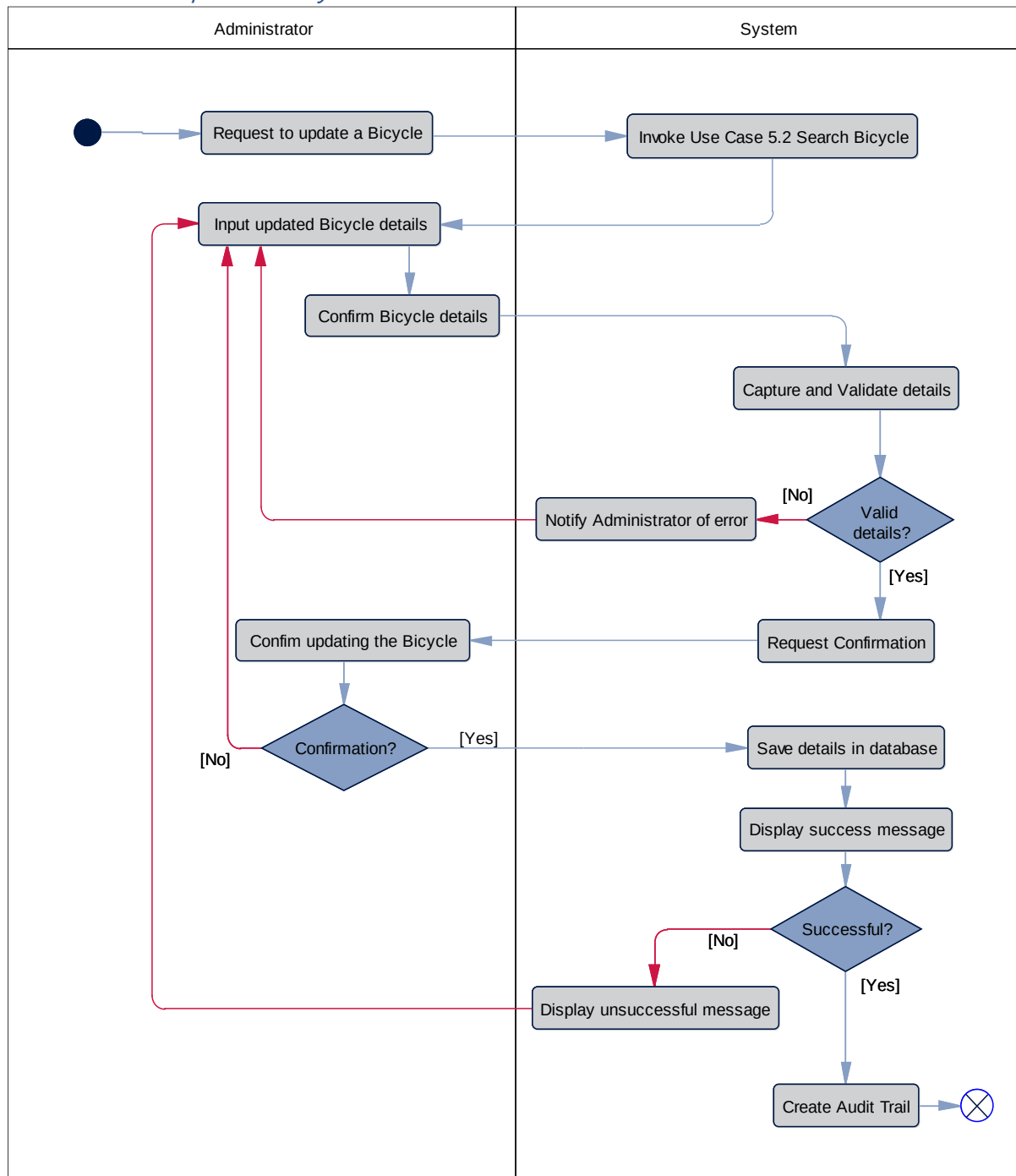


Figure 240 Update Bicycle Activity Diagram



4.2.5.4 Delete Bicycle

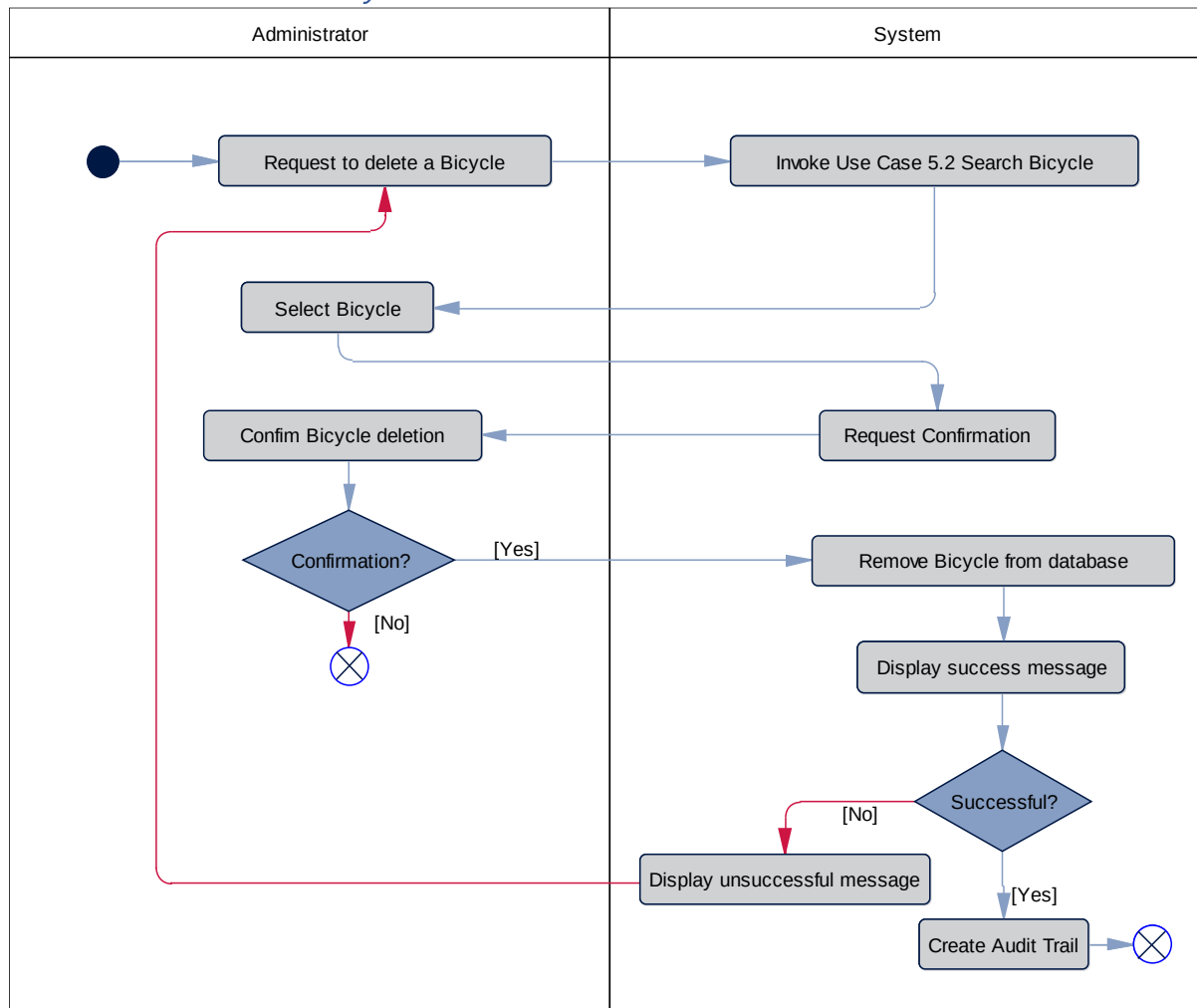


Figure 241 Delete Bicycle Activity Diagram



4.2.5.5 Assign Bicycle to Student

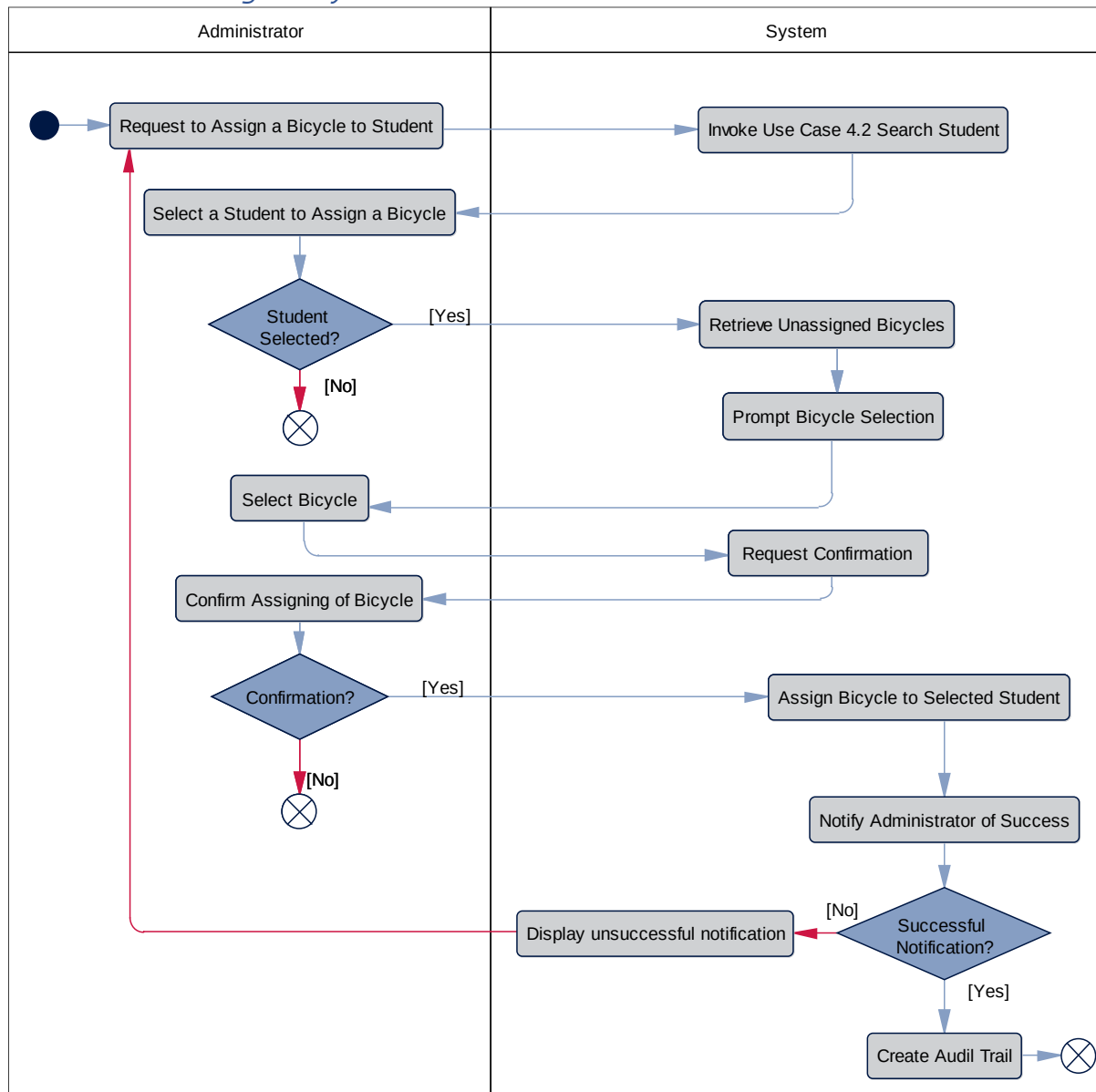


Figure 242 Assign Bicycle to Student Activity Diagram



4.2.5.6 Unassign Bicycle from Student

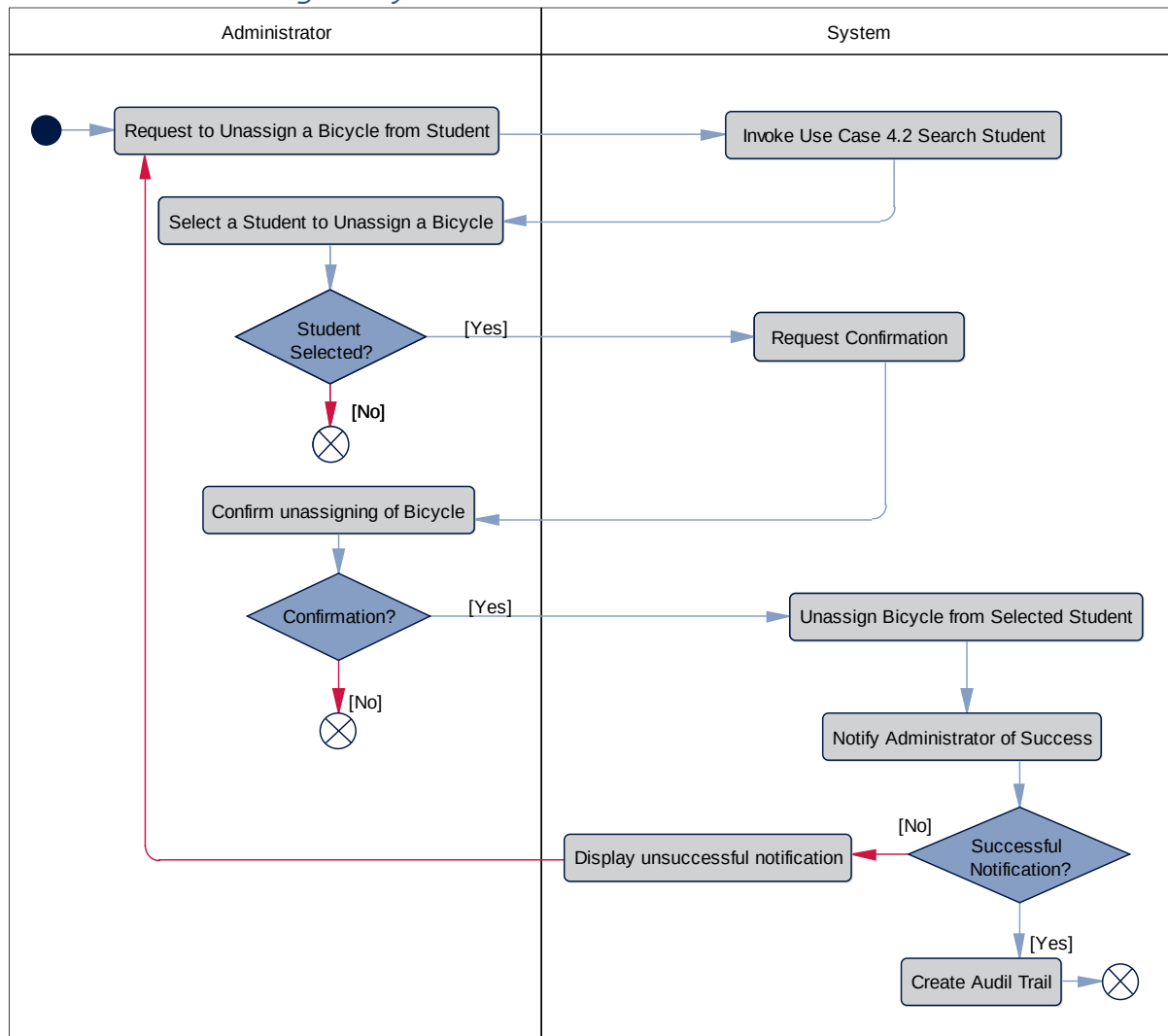


Figure 243 Unassign Bicycle from Student Activity Diagram



4.2.5.7 Assign Tag to Bicycle

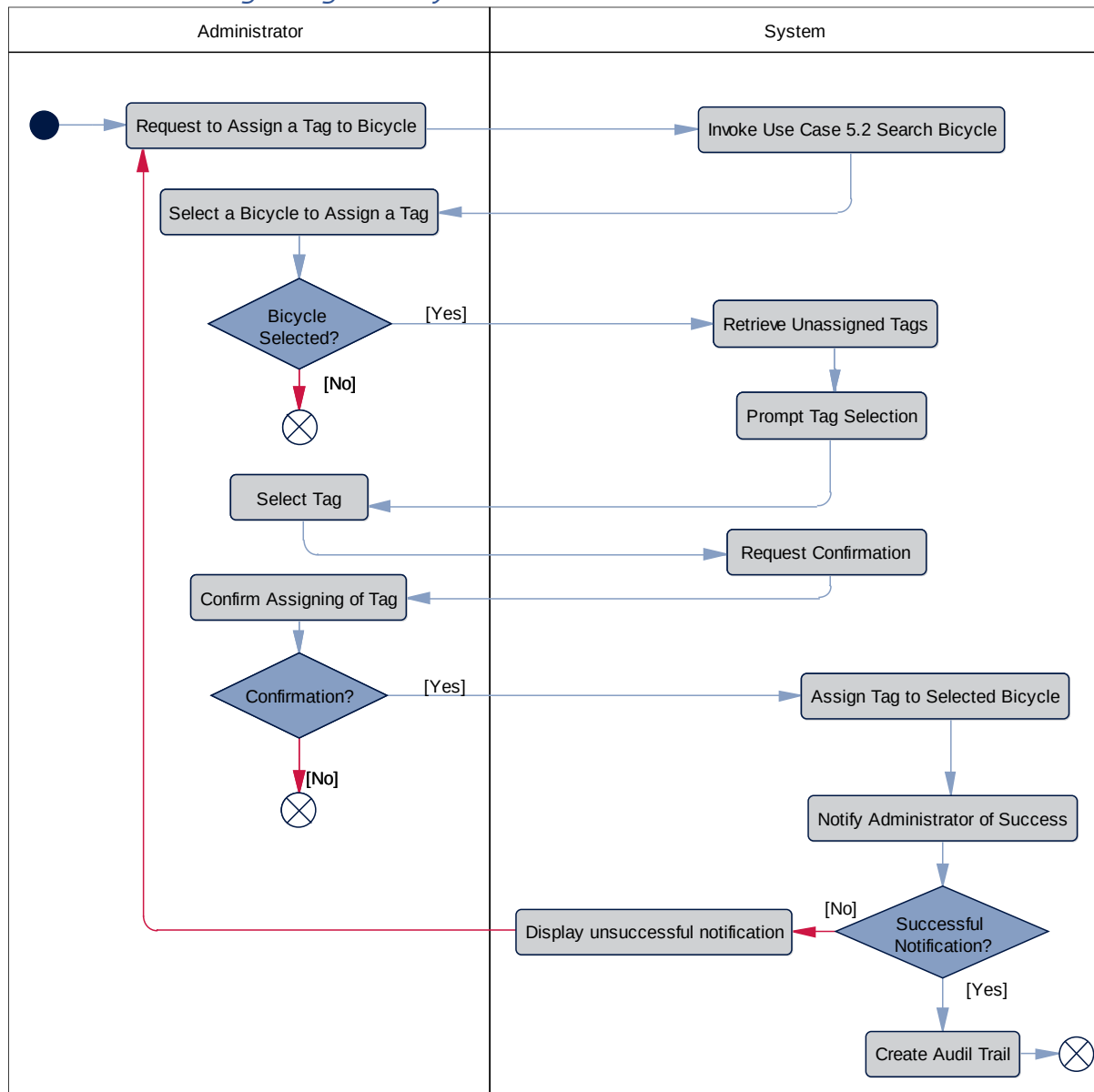


Figure 244 Assign Tag to Bicycle Activity Diagram



4.2.5.8 Unassign Tag from Bicycle

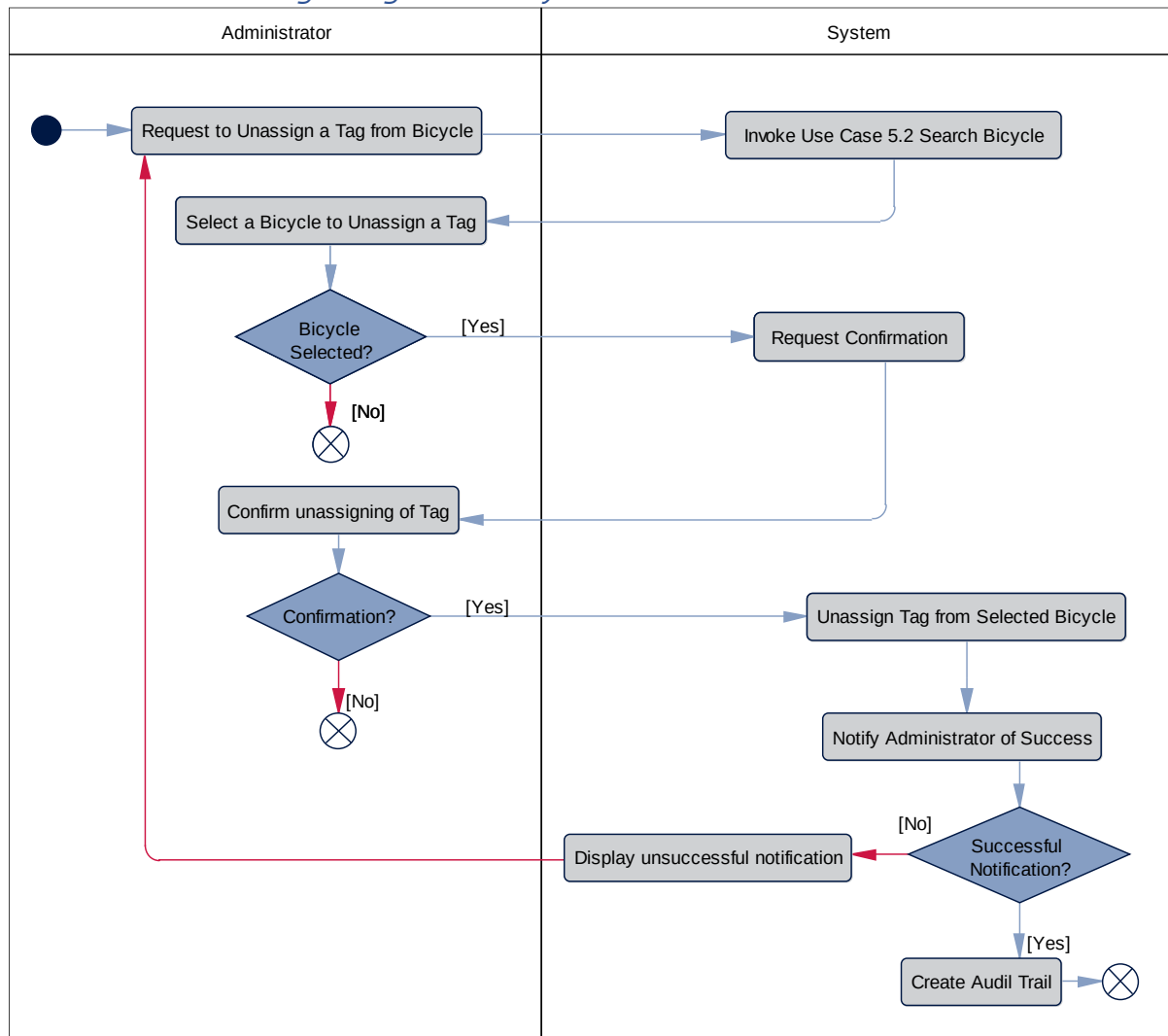


Figure 245 Unassign Tag from Bicycle Activity Diagram



4.2.6 Sub-system 6: Bicycle Part

4.2.6.1 Add Bicycle Part

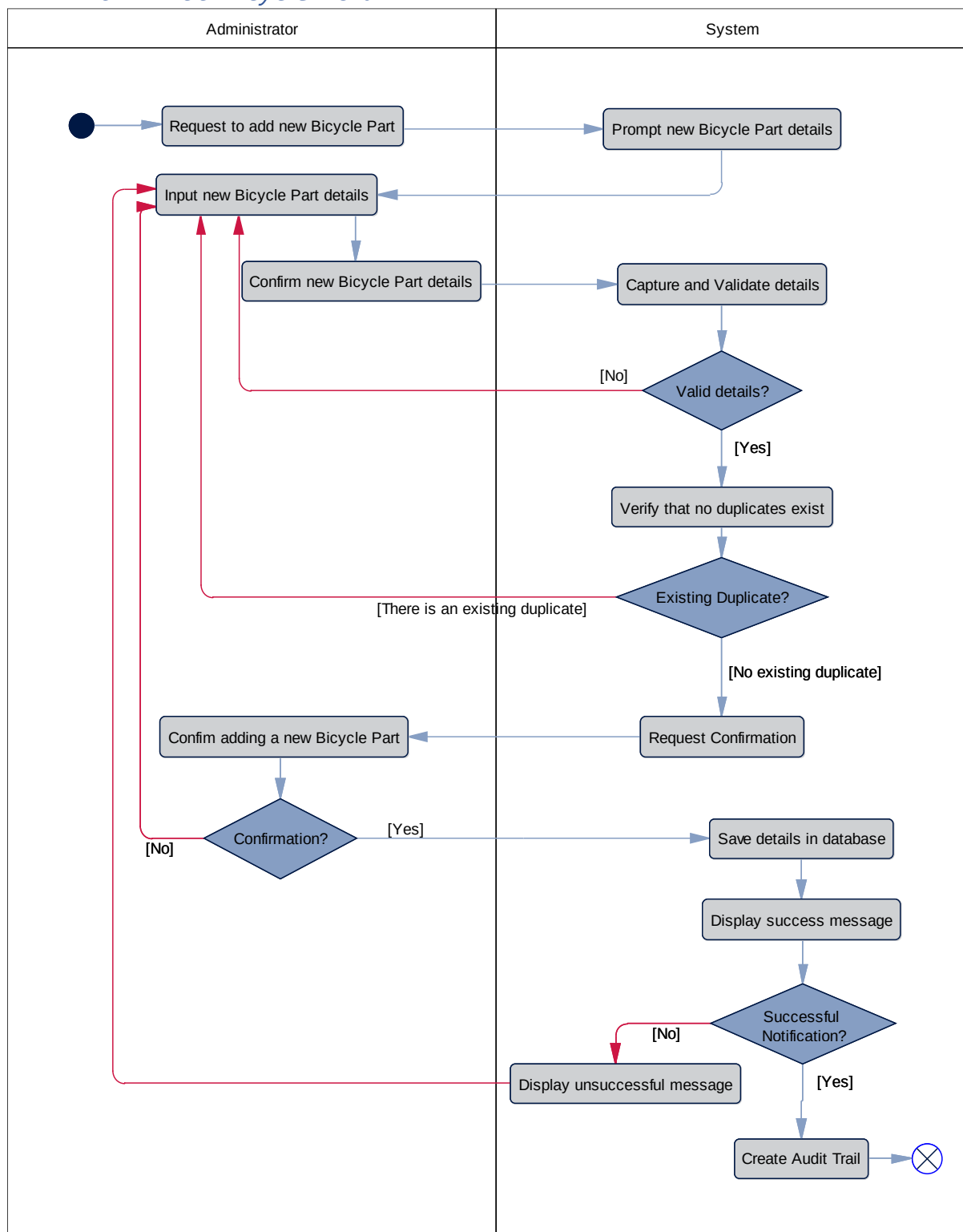


Figure 246 Add Bicycle Part Activity Diagram



4.2.6.2 Search Bicycle Part

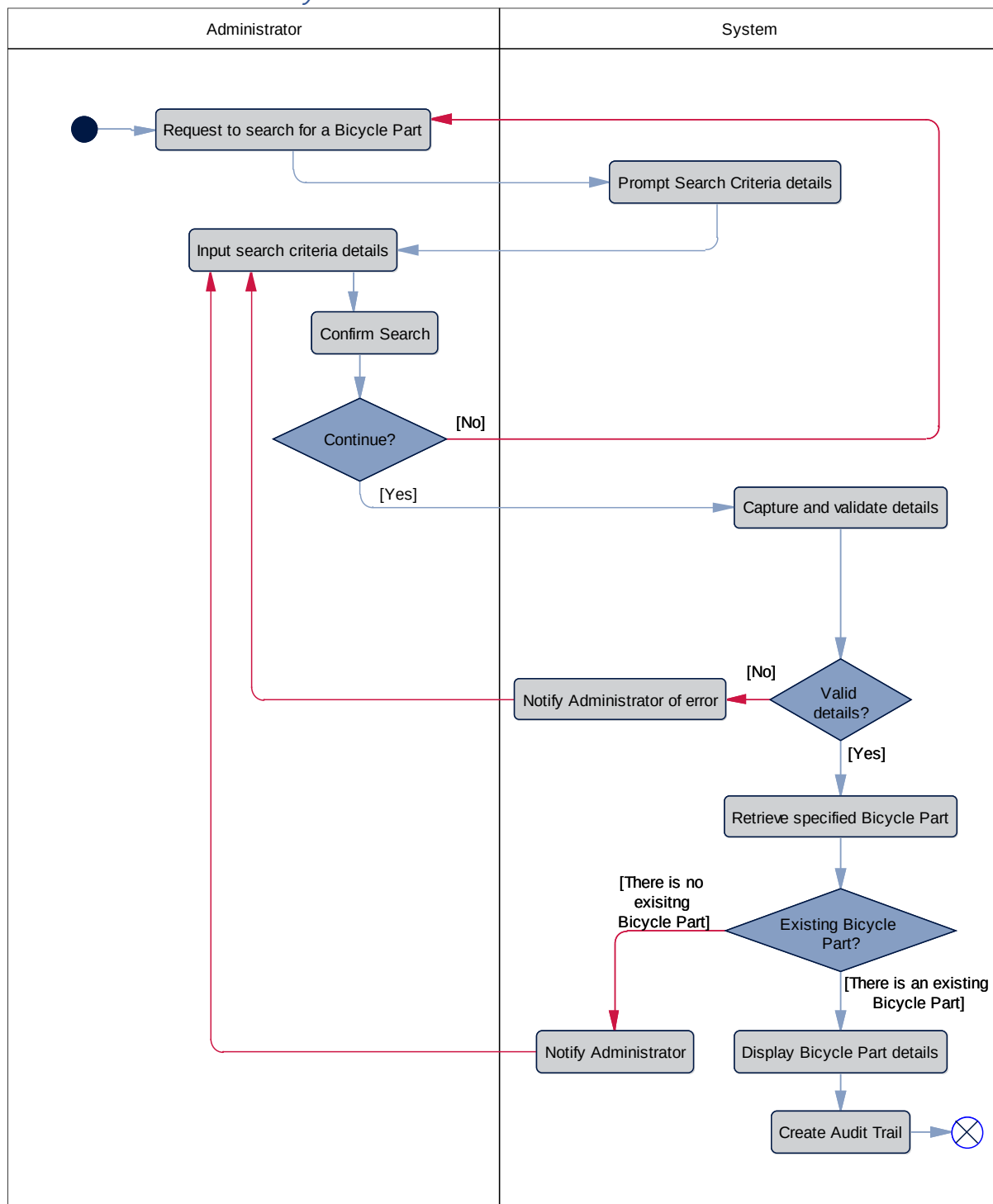


Figure 247 Search Bicycle Part Activity Diagram



4.2.6.3 Update Bicycle Part

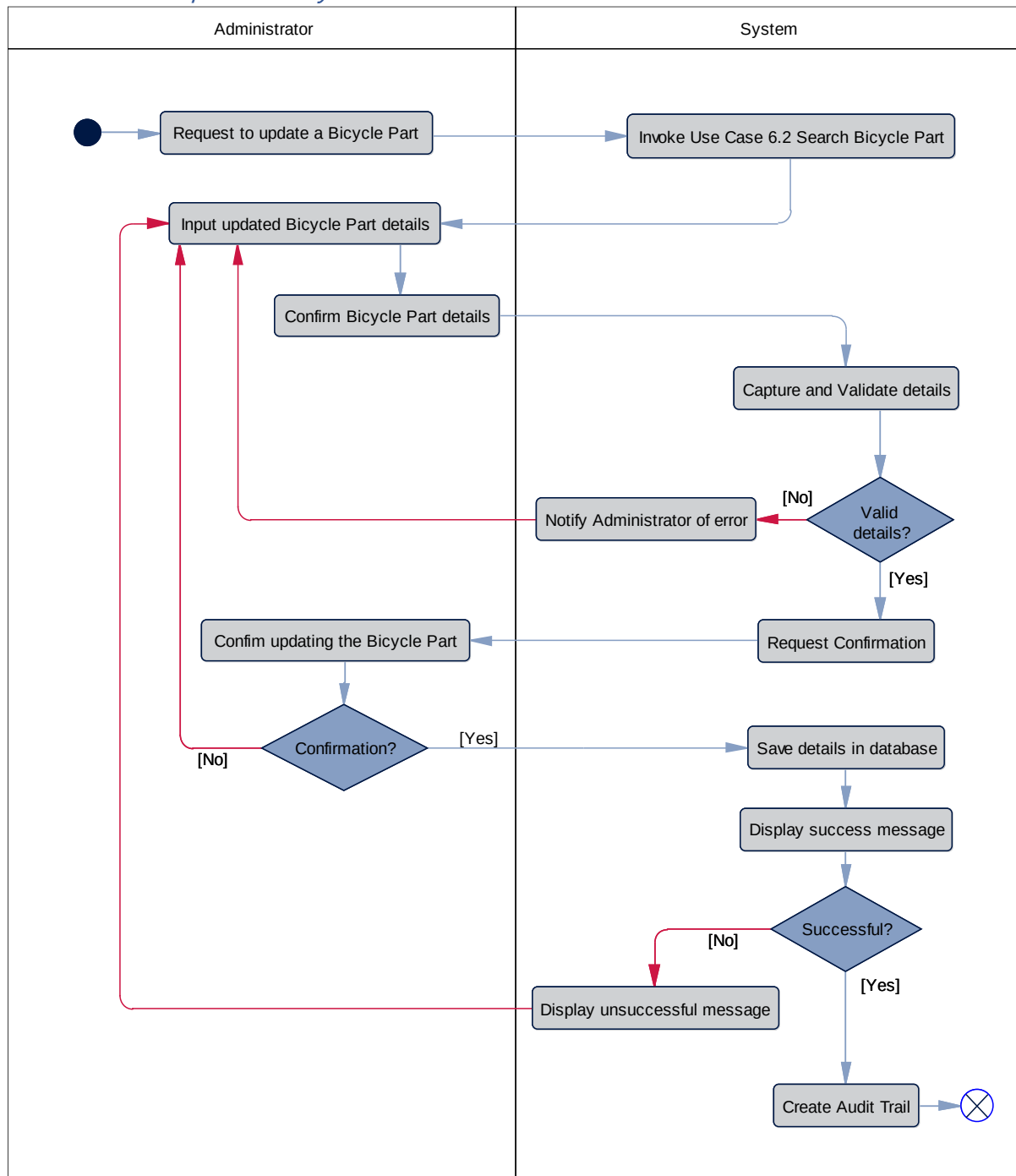


Figure 248 Update Bicycle Part Activity Diagram



4.2.6.4 Delete Bicycle Part

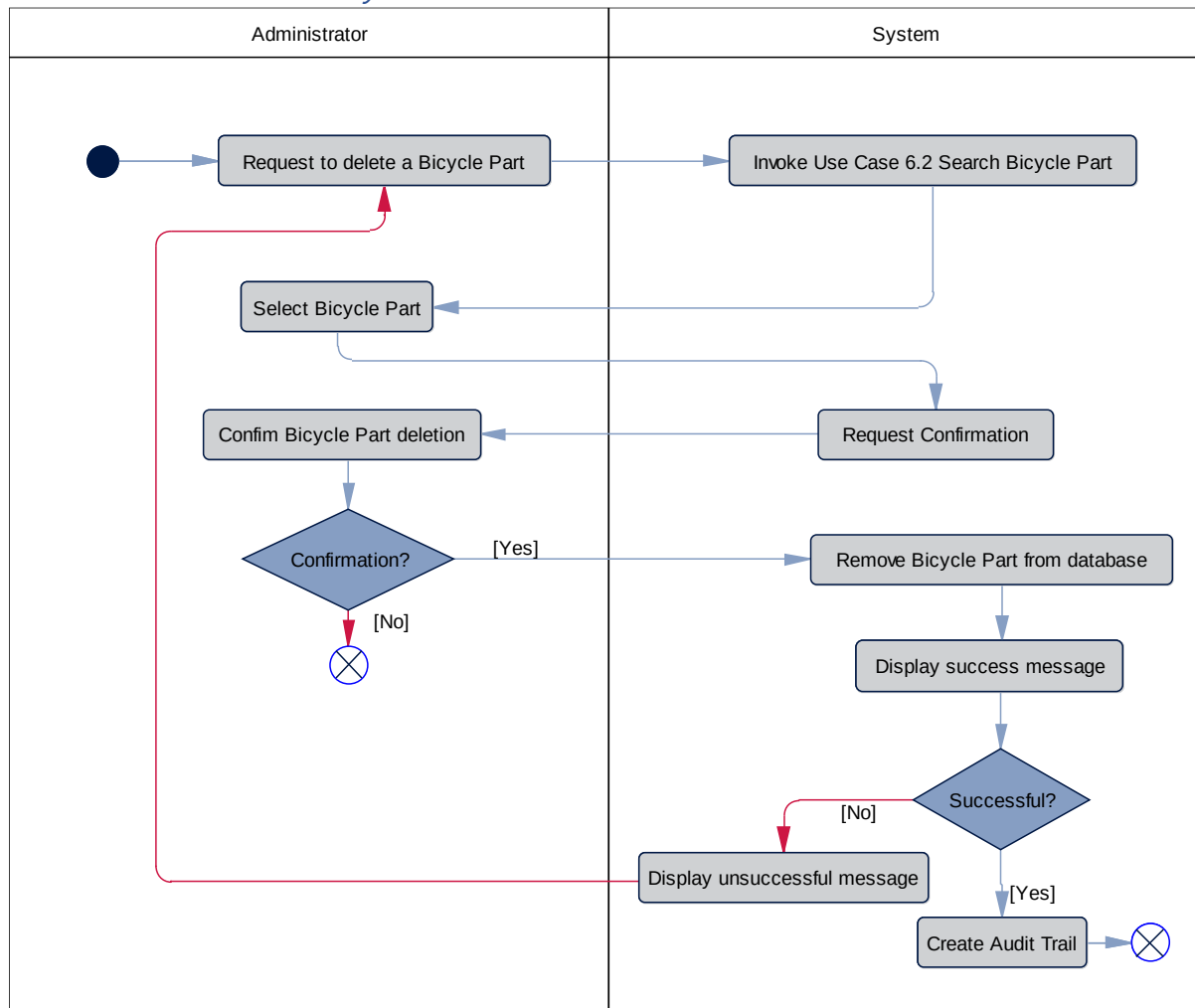


Figure 249 Delete Bicycle Part Activity Diagram



4.2.6.5 Add Bicycle Section

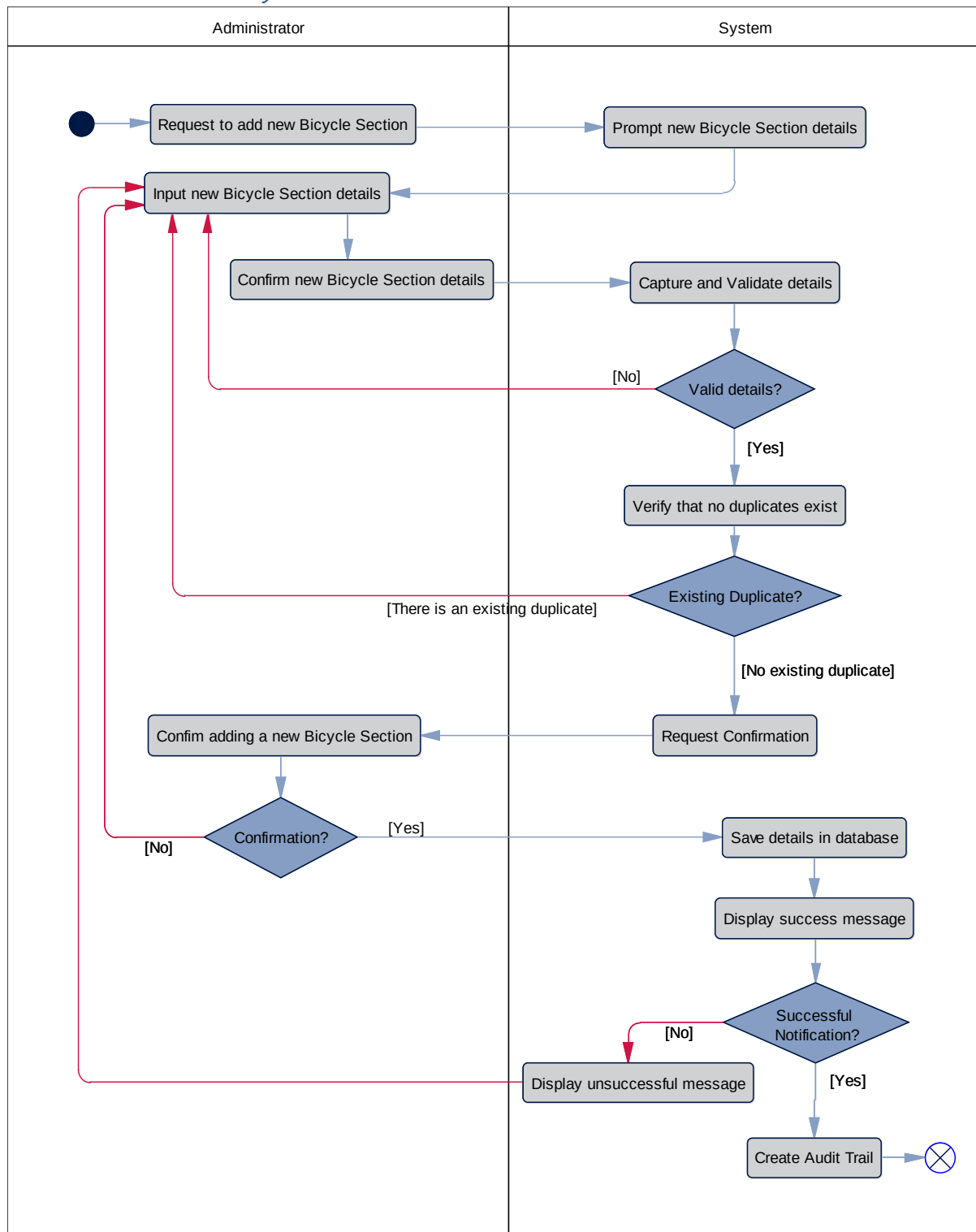


Figure 250 Add Bicycle Section Activity Diagram



4.2.6.6 Search Bicycle Section

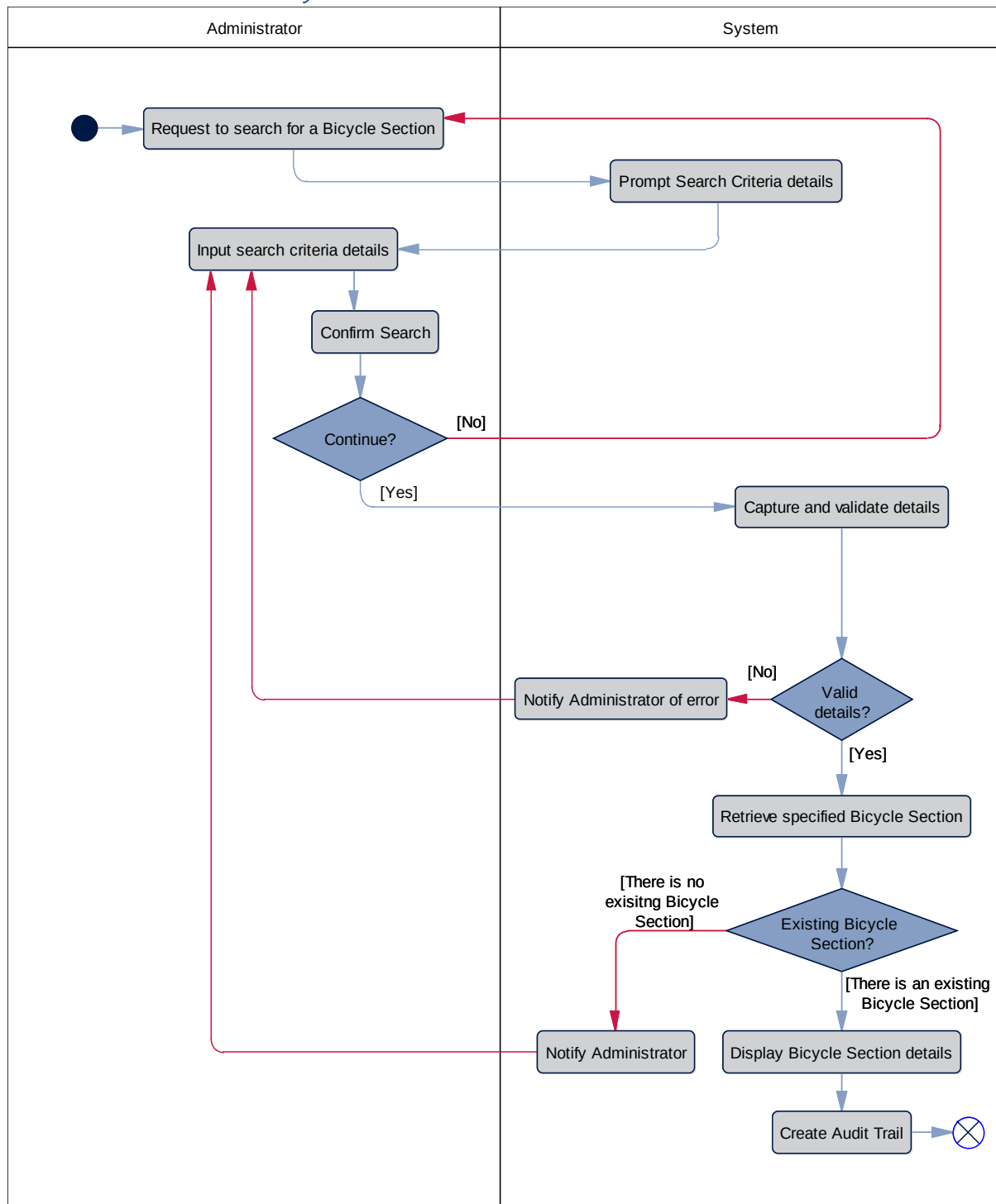


Figure 251 Search Bicycle Section Activity Diagram



4.2.6.7 Update Bicycle Section

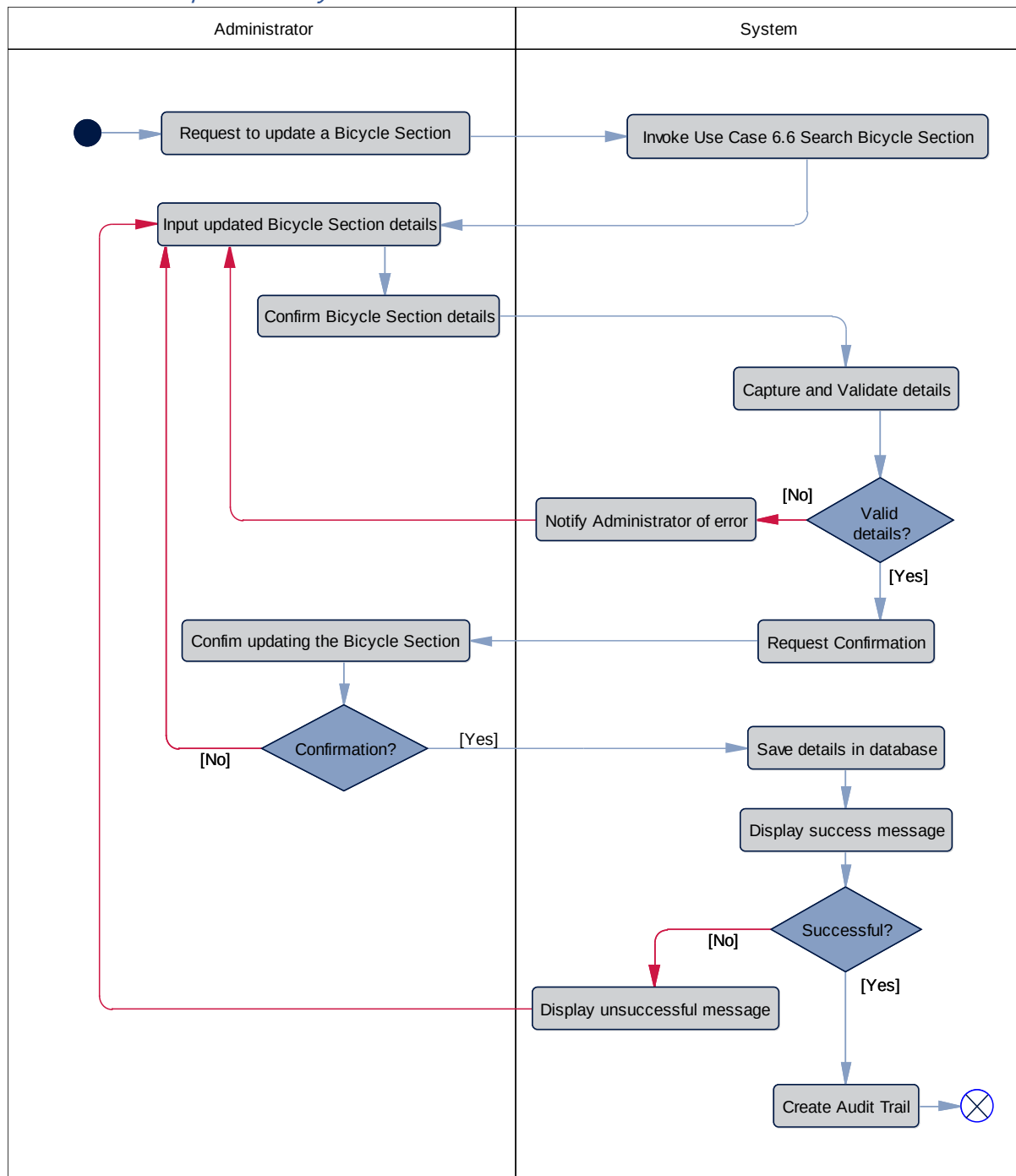


Figure 252 Update Bicycle Section Activity Diagram

4.2.6.8 Delete Bicycle Section

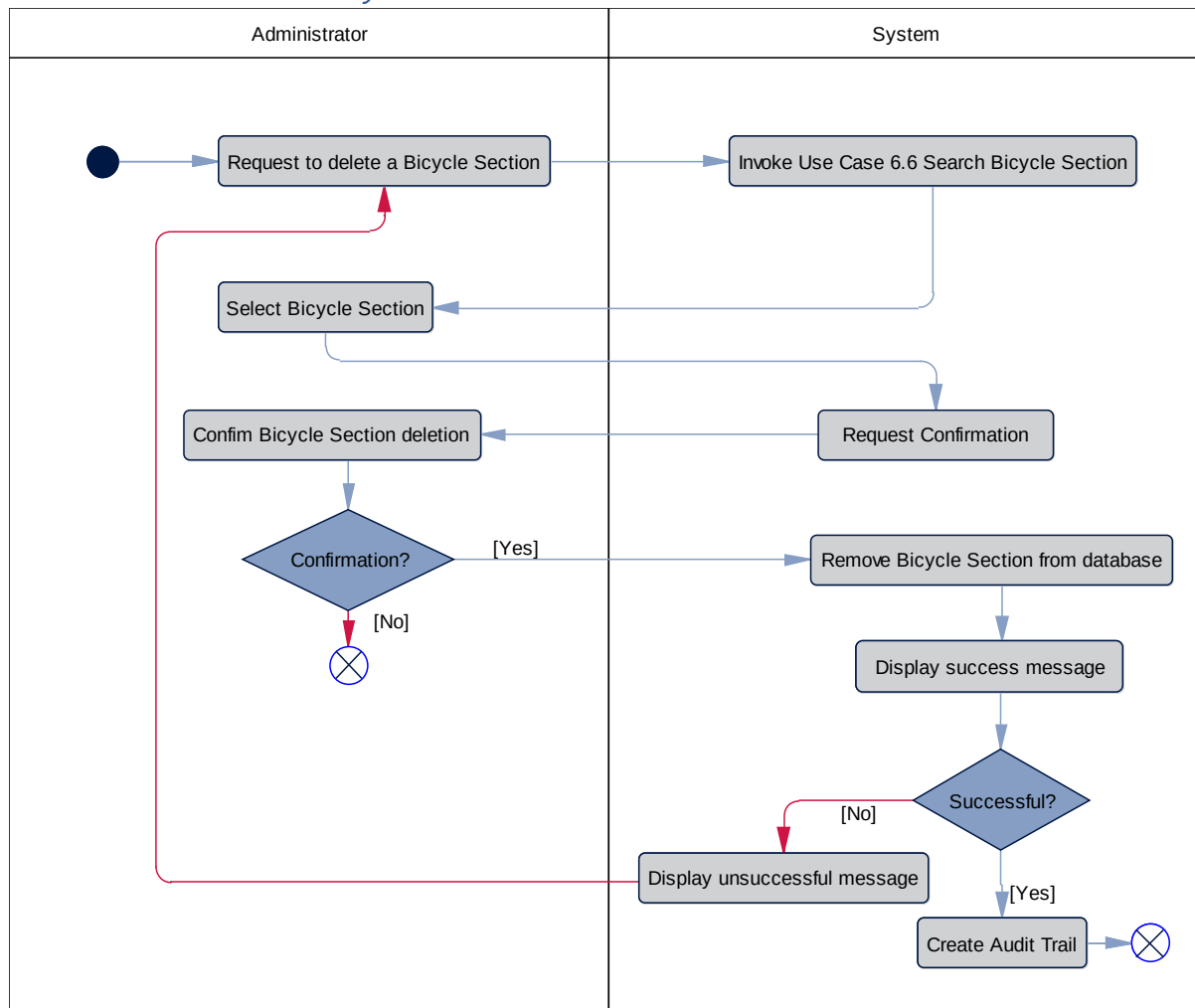


Figure 253 Delete Bicycle Section Activity Diagram



4.2.7 Sub-system 7: Tracking

4.2.7.1 Add Log Entry

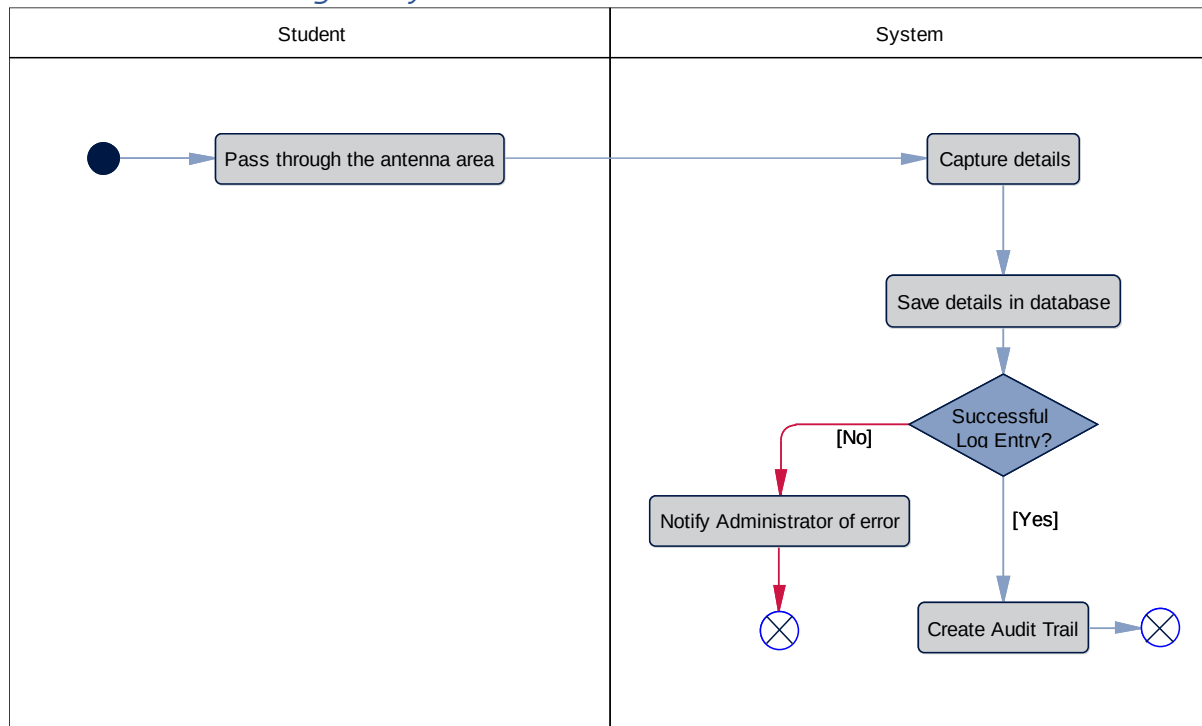


Figure 254 Add Log Entry Activity Diagram



4.2.7.2 Search Log Entry

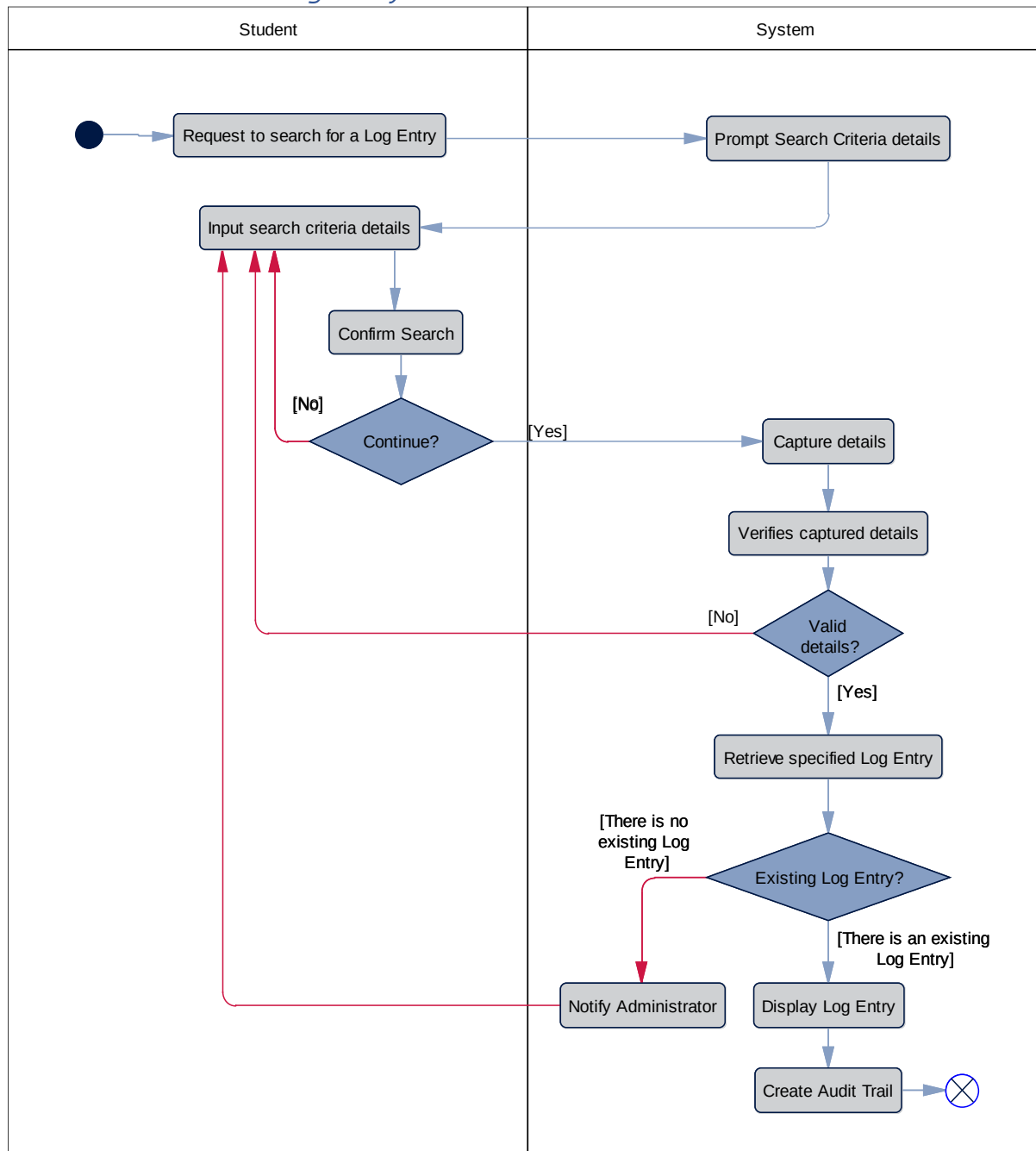


Figure 255 Search Log Entry Activity Diagram



4.2.8 Sub-system 8: Job Tasks

4.2.8.1 Add Fault Task

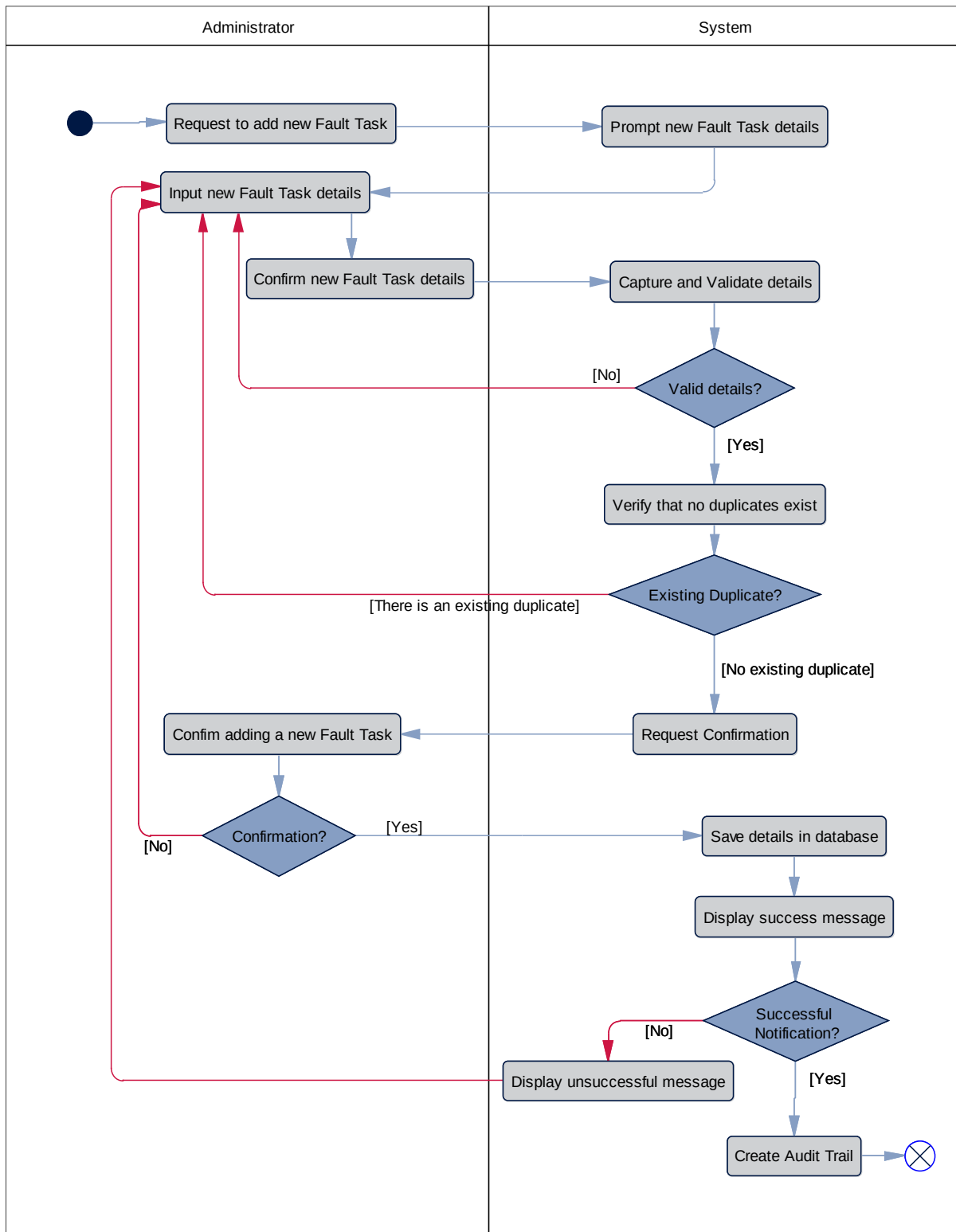


Figure 256 Add Fault Task Activity Diagram



4.2.8.2 Search Fault Task

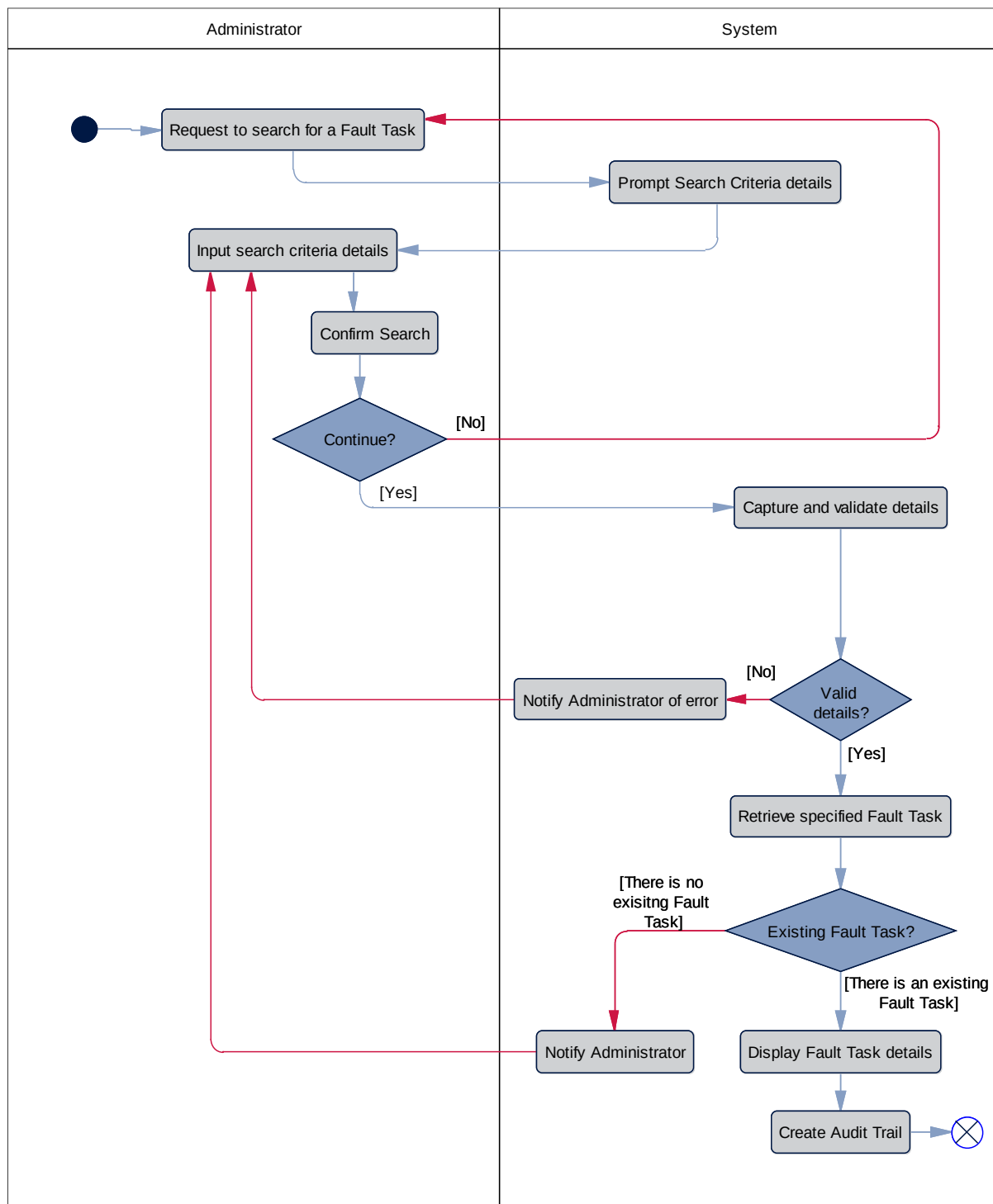


Figure 257 Search Fault Task Activity Diagram



4.2.8.3 Update Fault Task

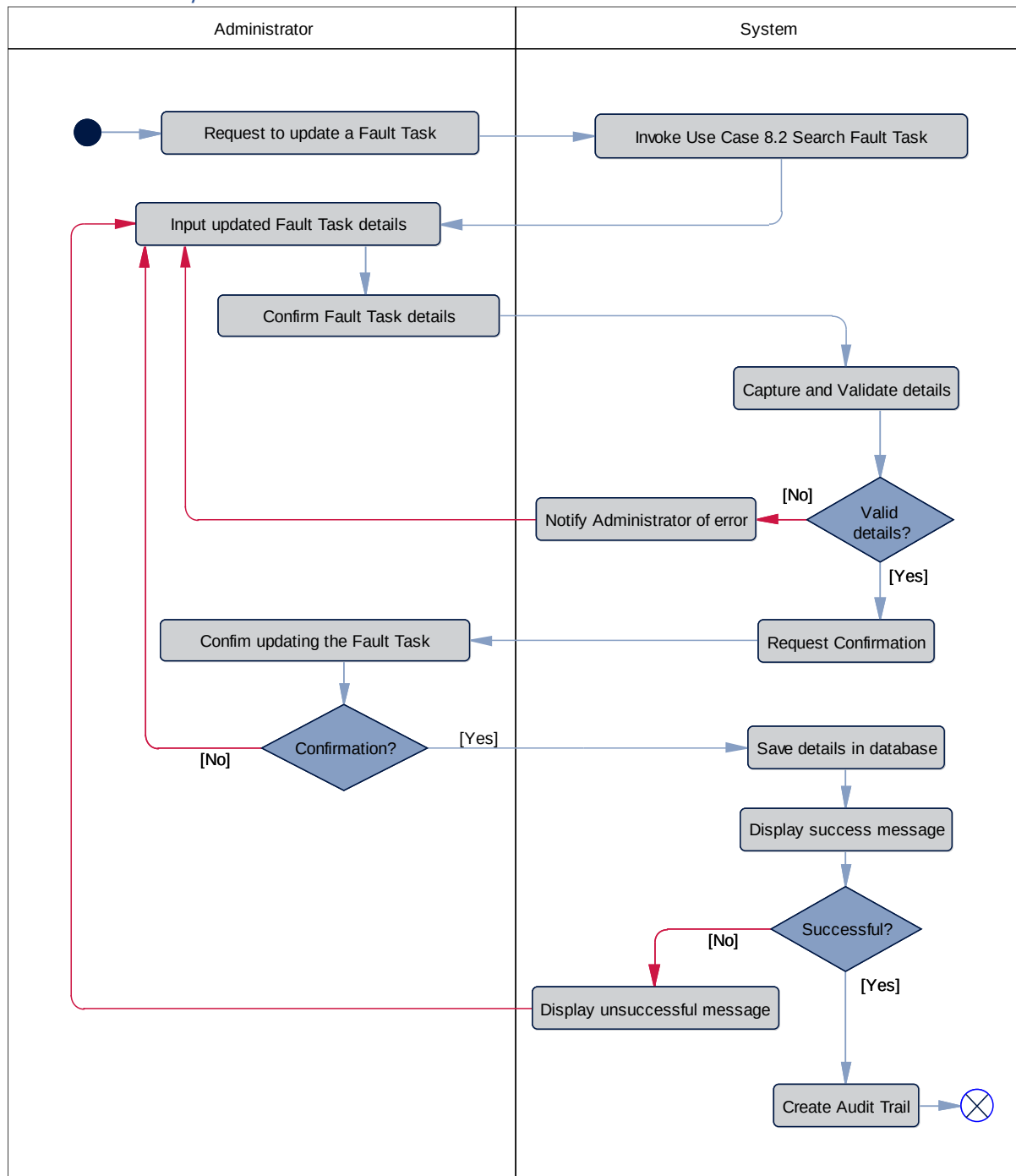


Figure 258 Update Fault Task Activity Diagram



4.2.8.4 Delete Fault Task

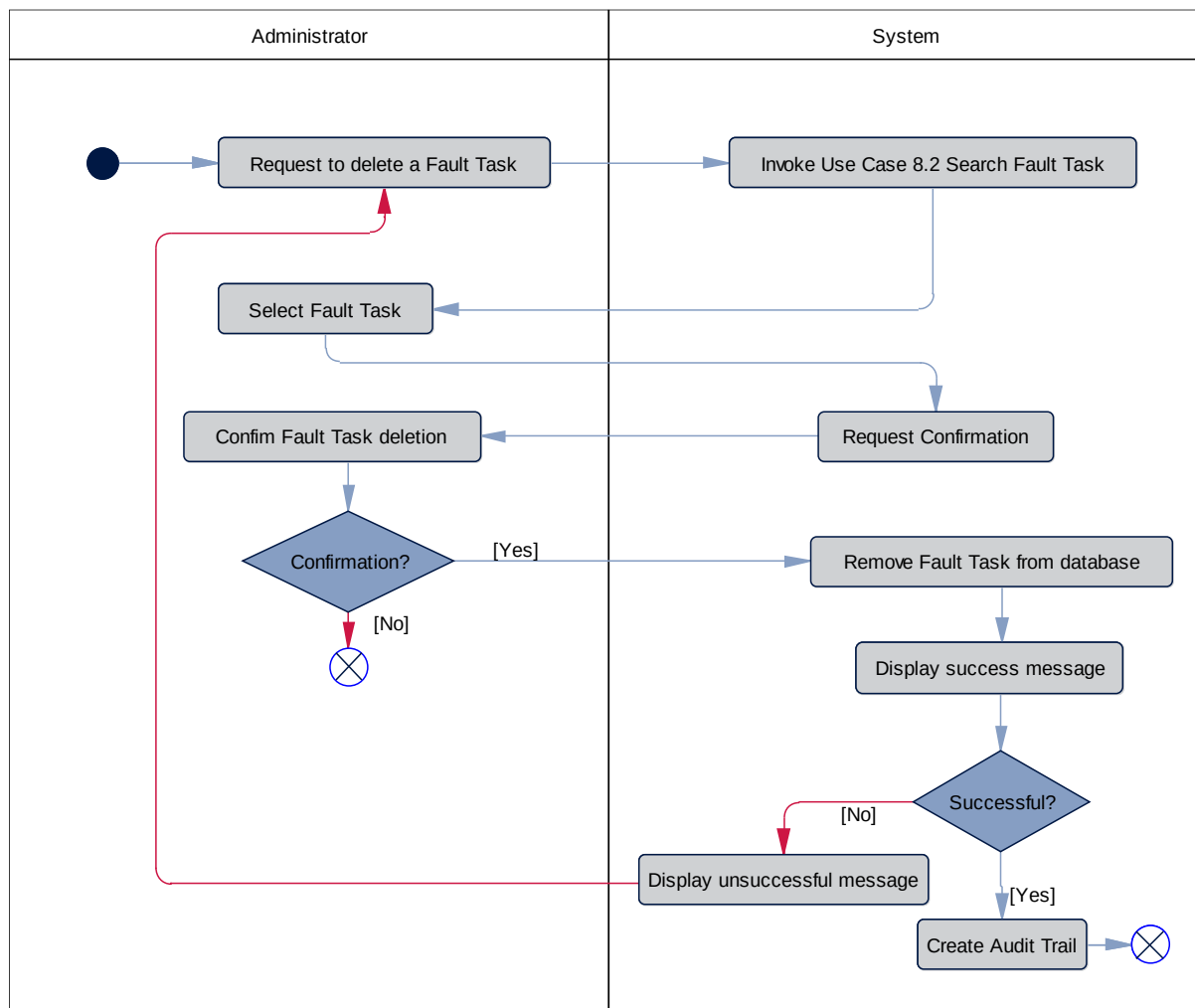


Figure 259 Delete Fault Task Activity Diagram



4.2.8.5 Add Maintenance Task

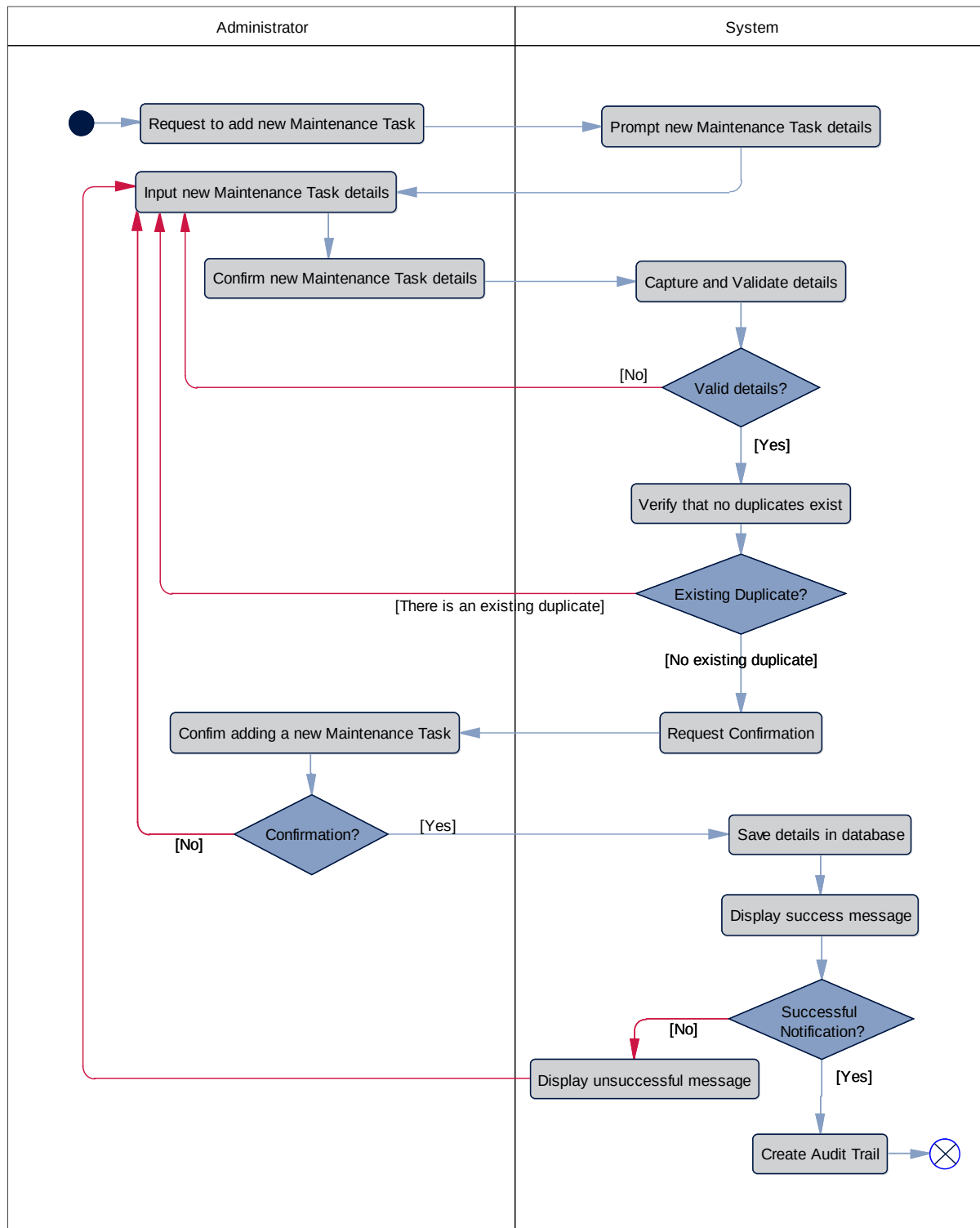


Figure 260 Add Maintenance Task Activity Diagram



4.2.8.6 Search Maintenance Task

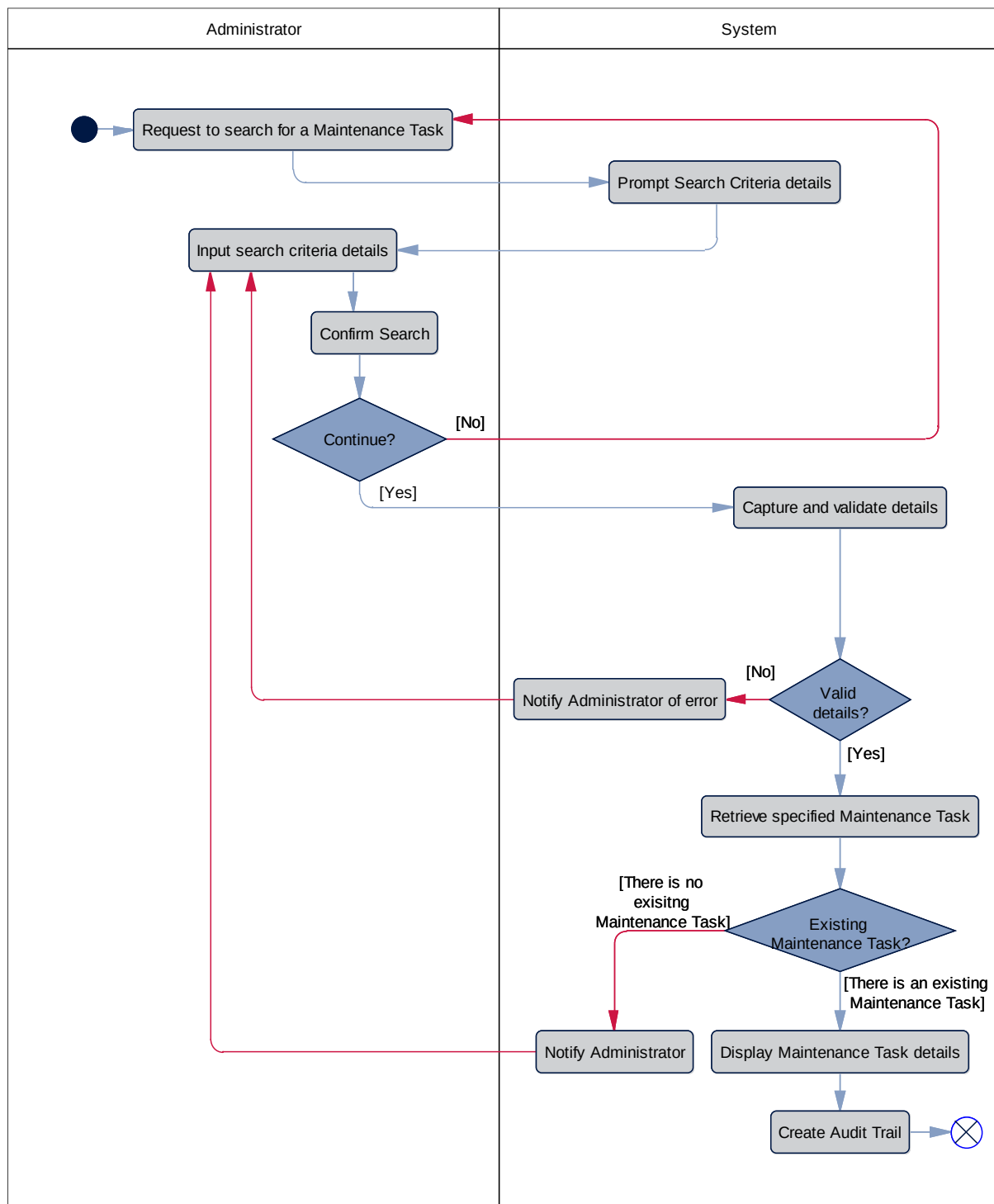


Figure 261 Search Maintenance Task Activity Diagram



4.2.8.7 Update Maintenance Task

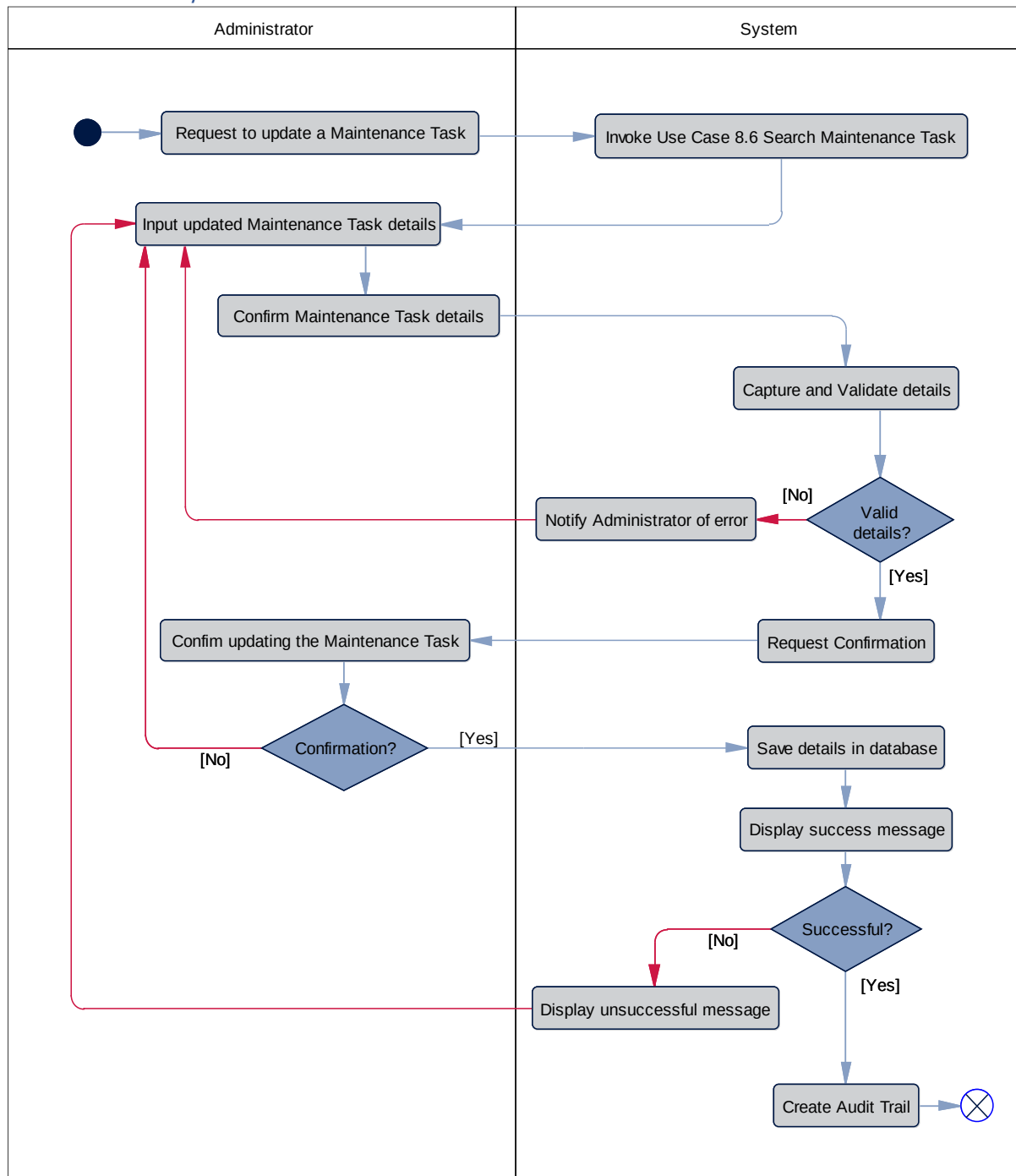


Figure 262 Update Maintenance Task Activity Diagram



4.2.8.8 Delete Maintenance Task

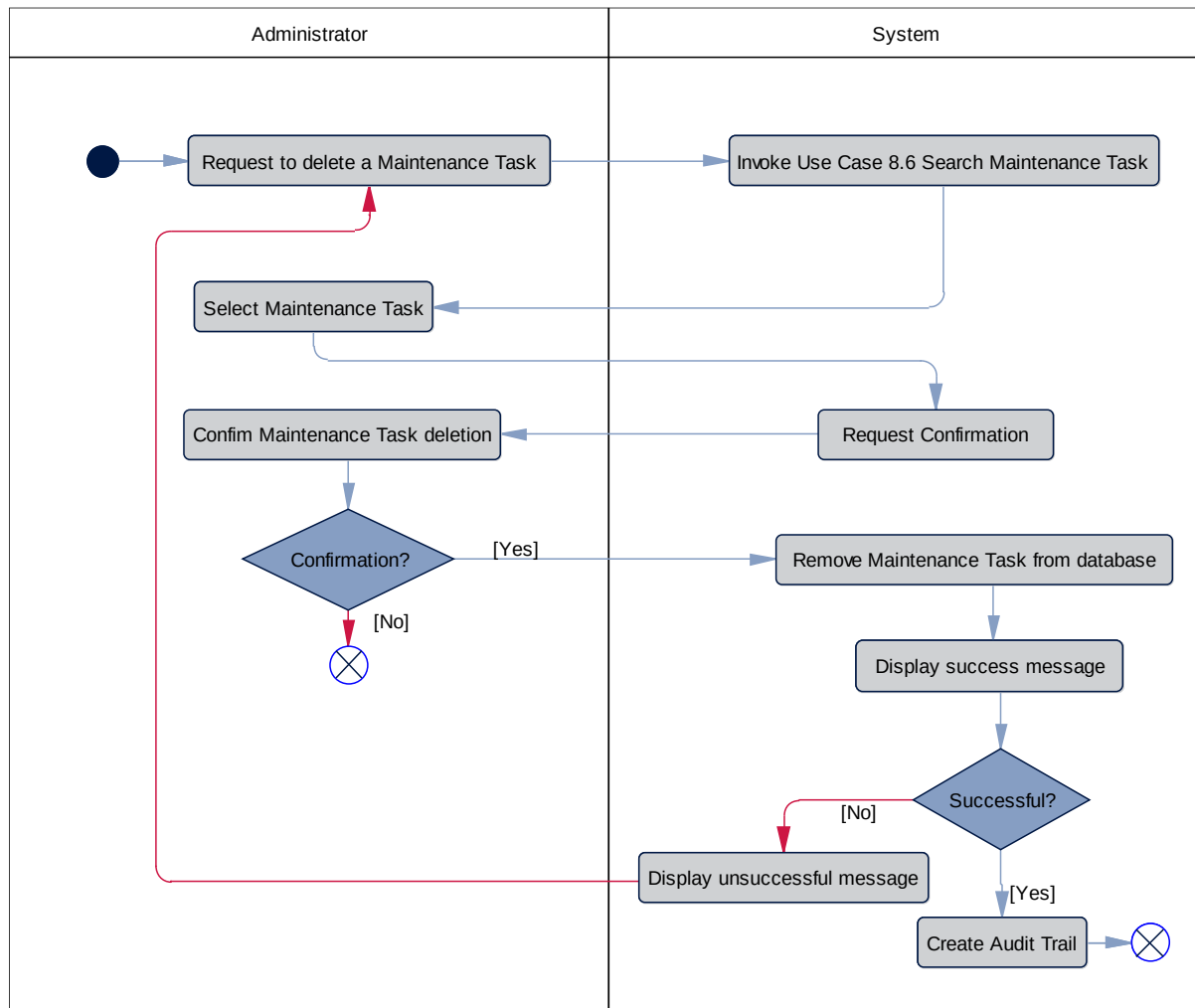


Figure 263 Delete Maintenance Task Activity Diagram



4.2.9 Sub-system 9: School

4.2.9.1 Add School

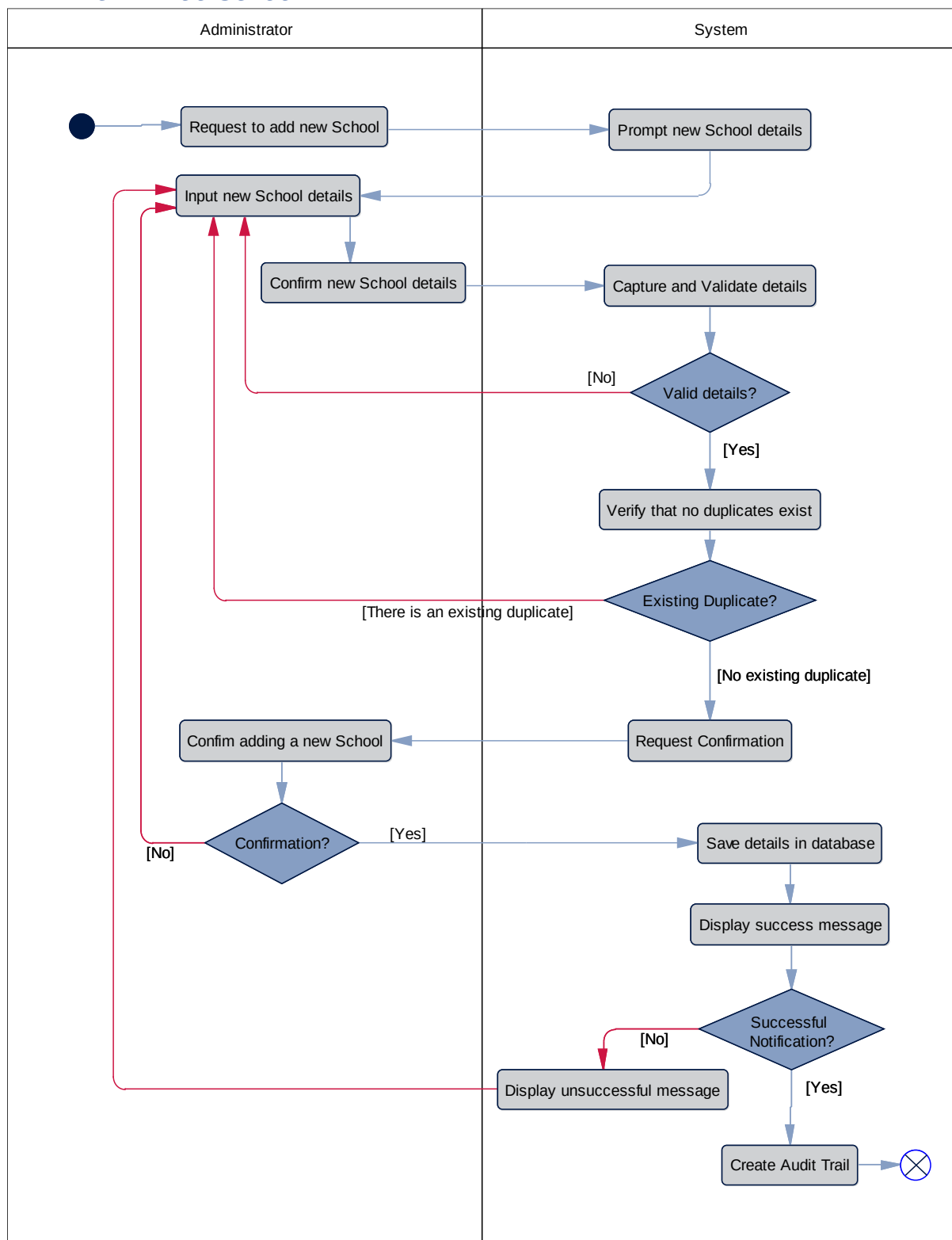


Figure 264 Add School Activity Diagram



4.2.9.2 Search School

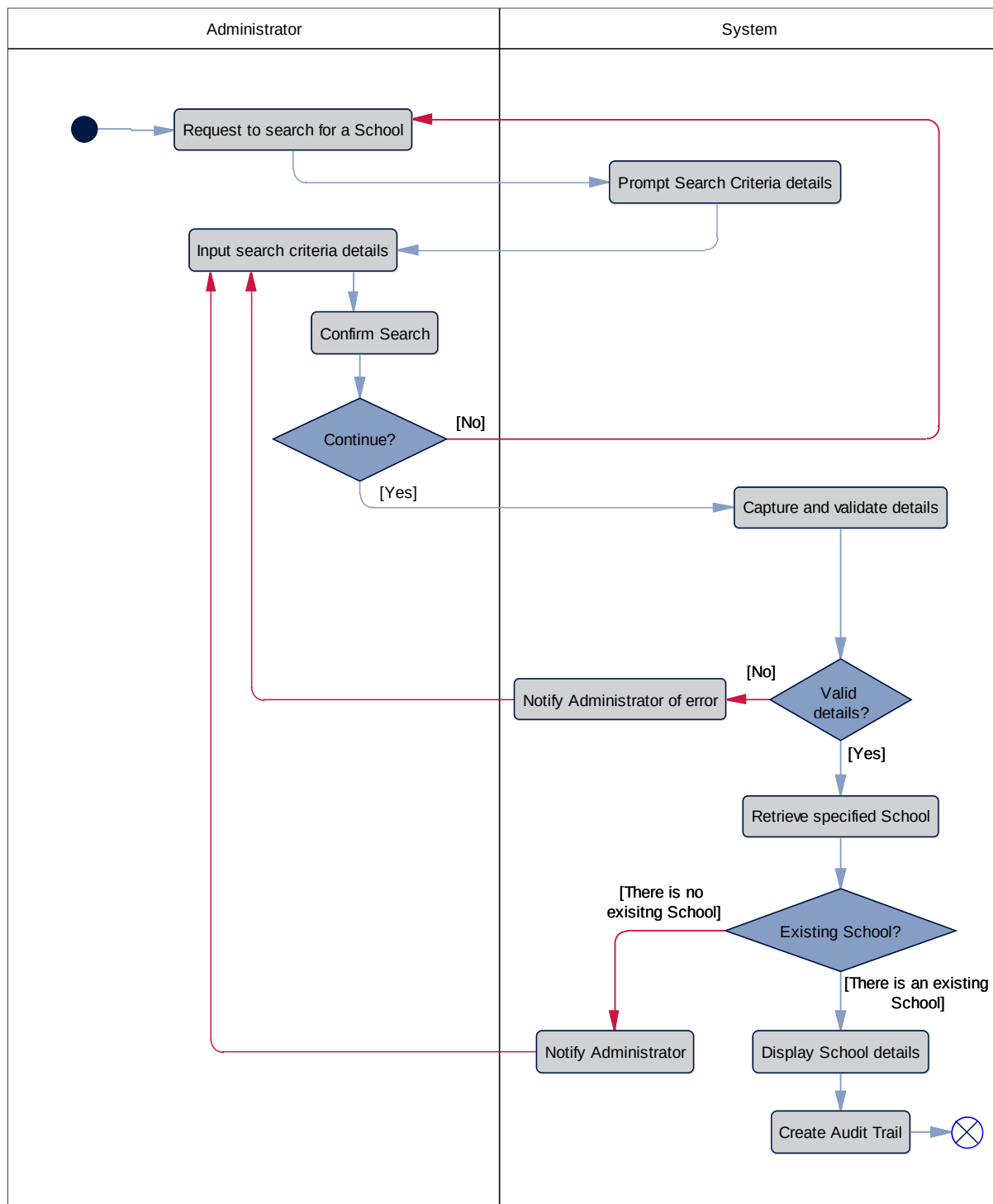


Figure 265 Search School Activity Diagram



4.2.9.3 Update School

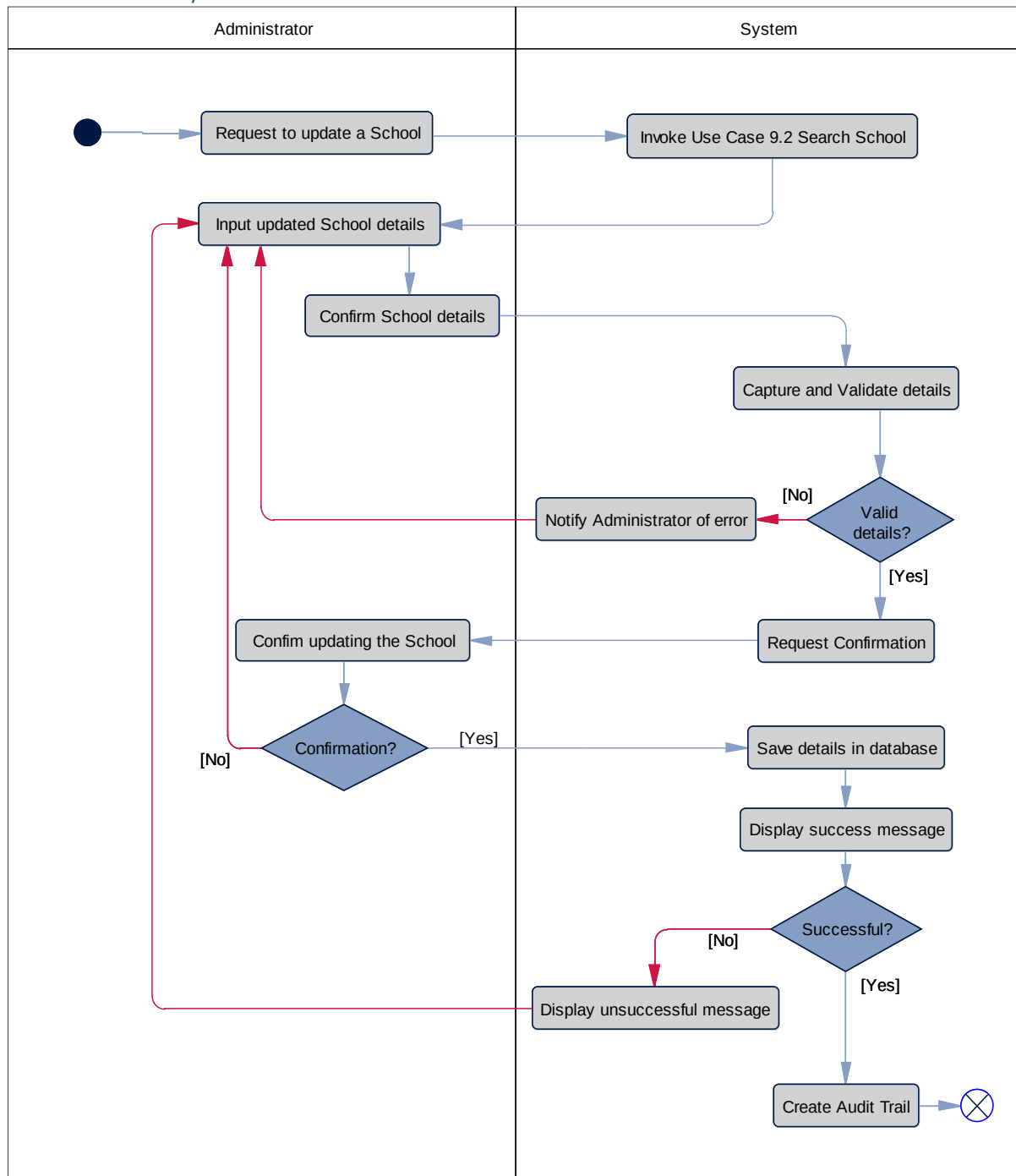


Figure 266 Update School Activity Diagram



4.2.9.4 Delete School

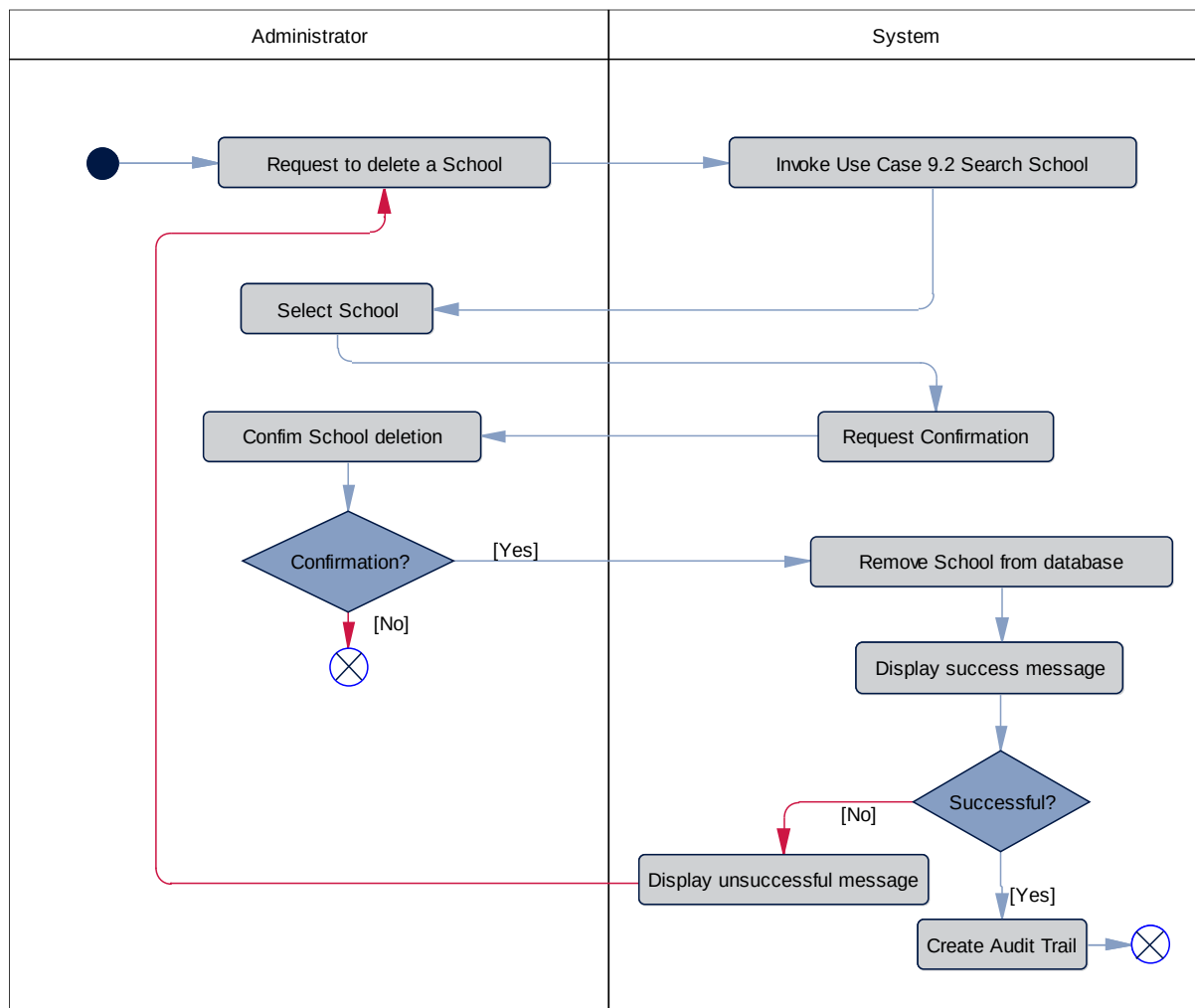


Figure 267 Delete School Activity Diagram



4.2.10 Sub-system 10: Job

4.2.10.1 Report Fault

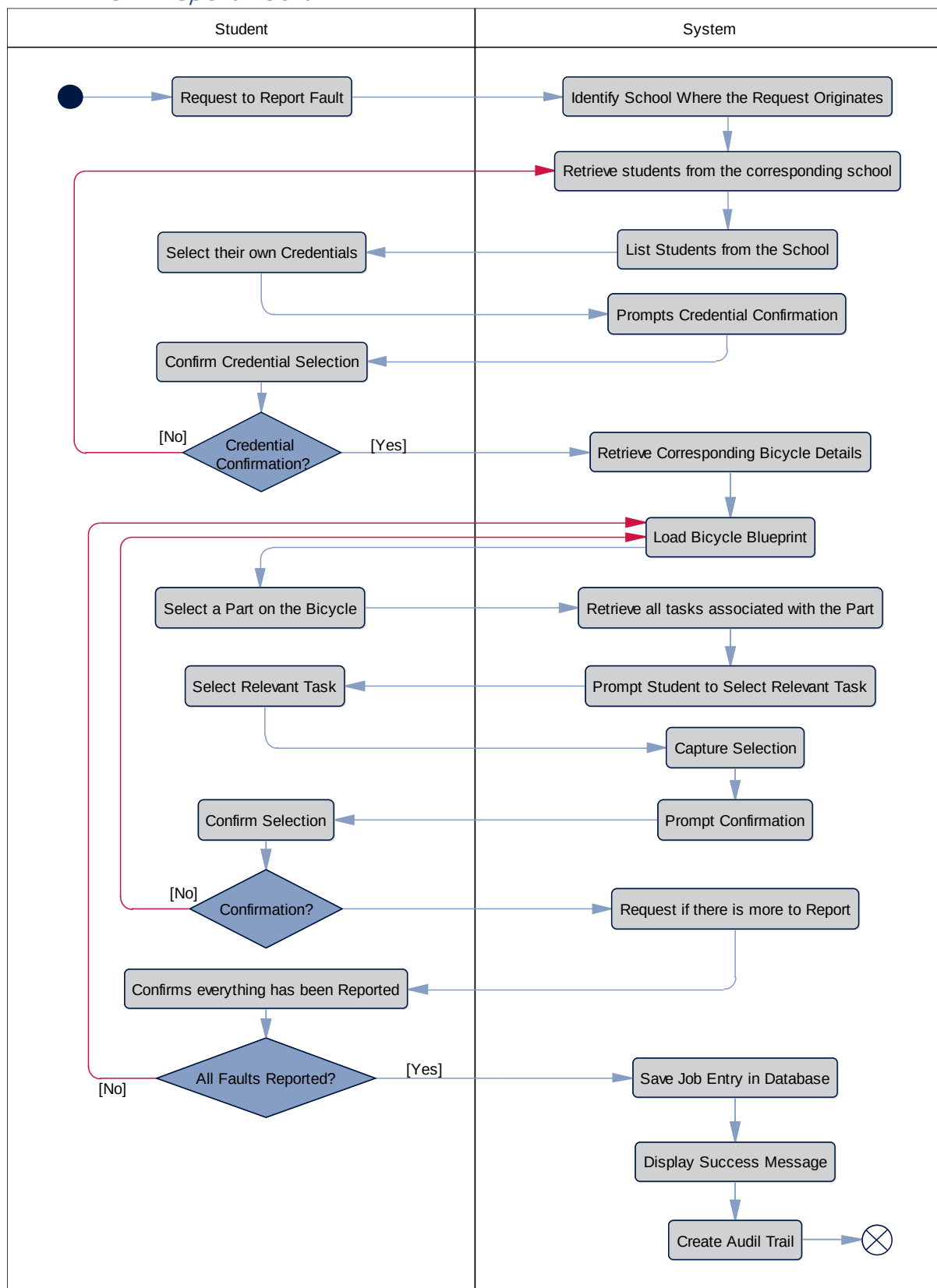


Figure 268 Report Fault Activity Diagram



4.2.10.2 View Jobs

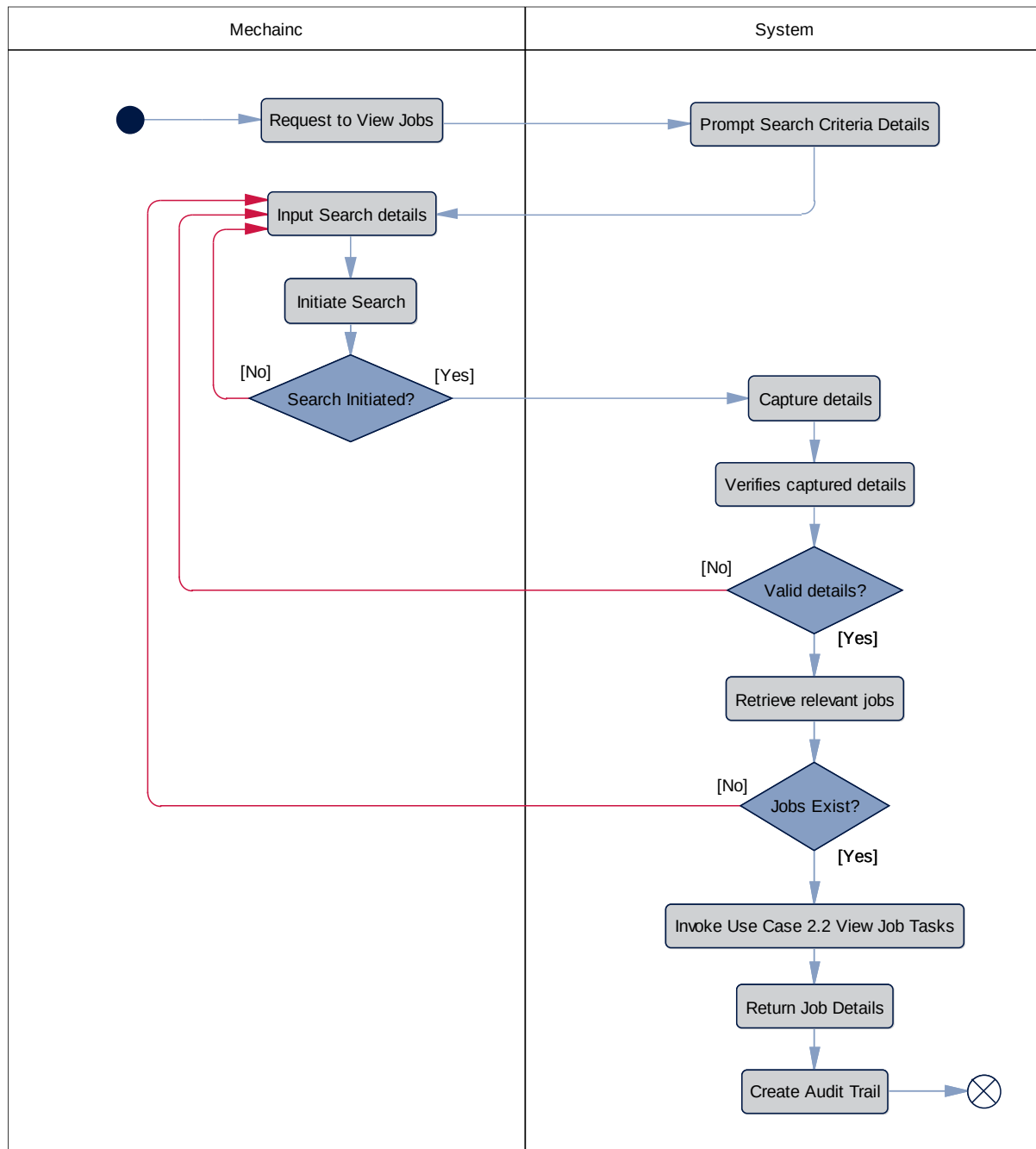


Figure 269 View Jobs Activity Diagrams



4.2.10.3 Check-in Bicycle

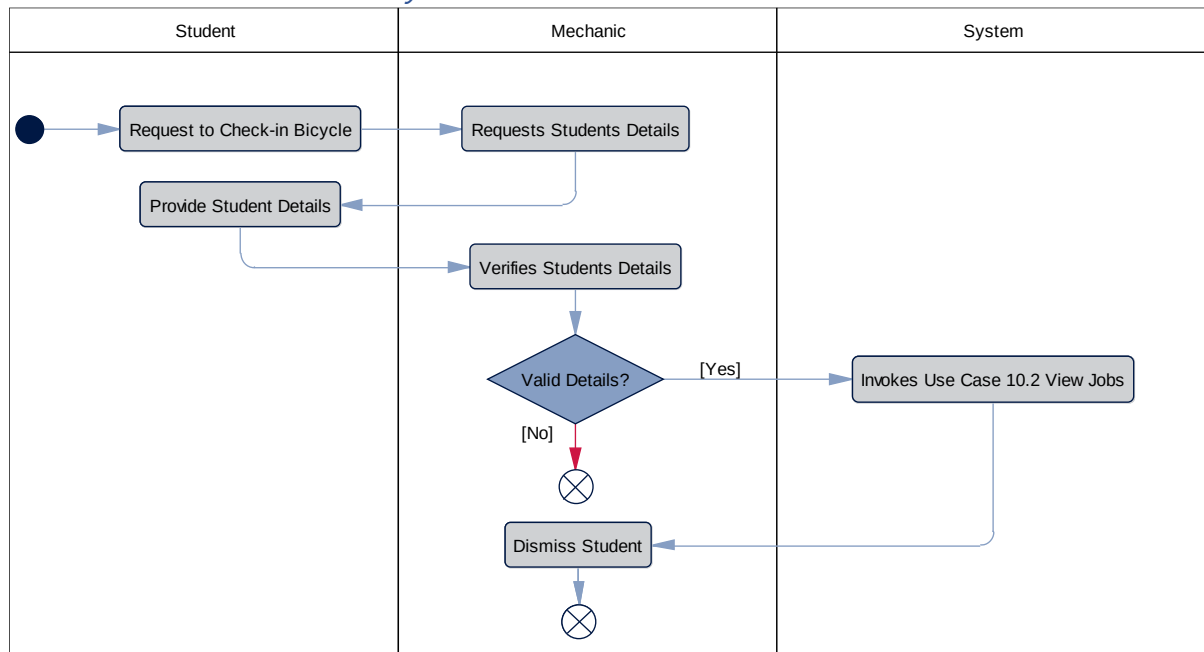


Figure 270 Check-in Bicycle Activity Diagram



4.2.10.4 Report Maintenance

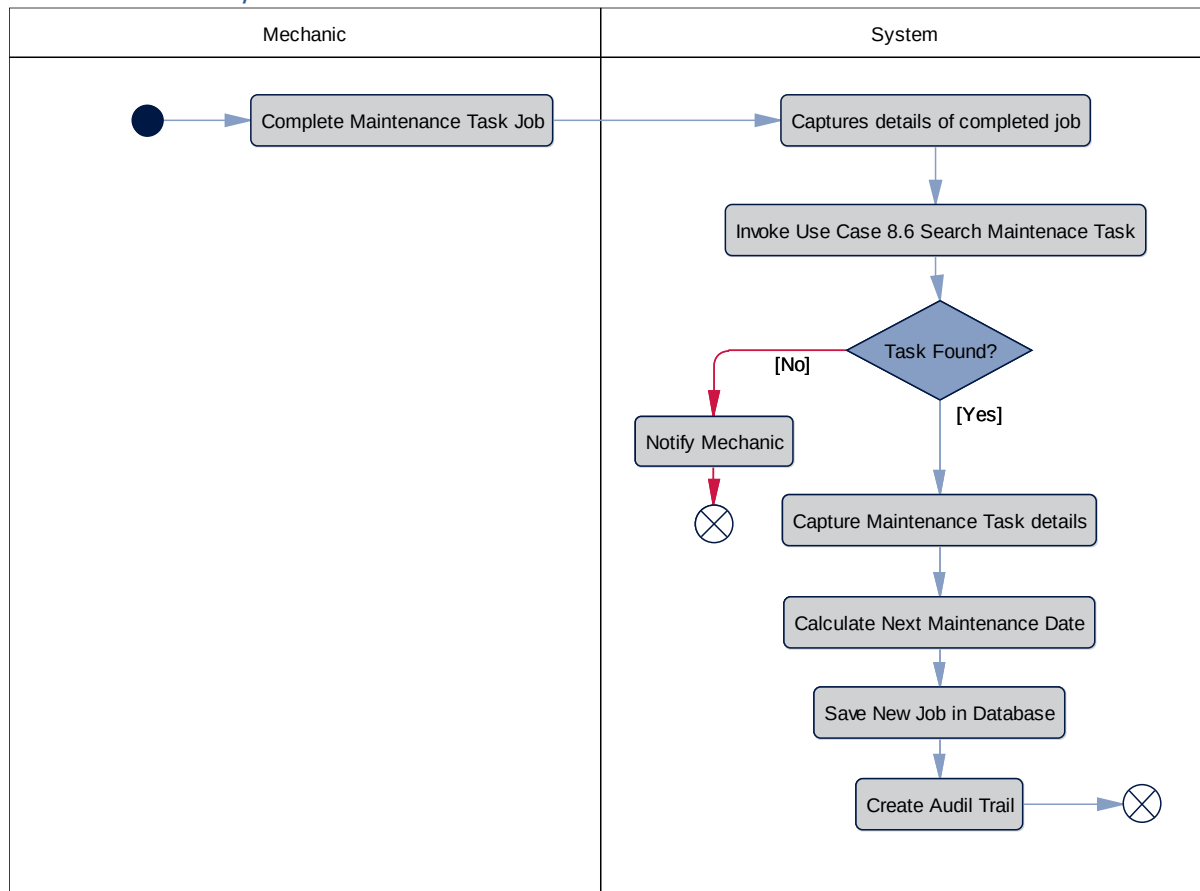


Figure 271 Report Maintenance Activity Diagram



4.2.11 Sub-system 11: Reporting

4.2.11.1 Generate Maintenance Schedule

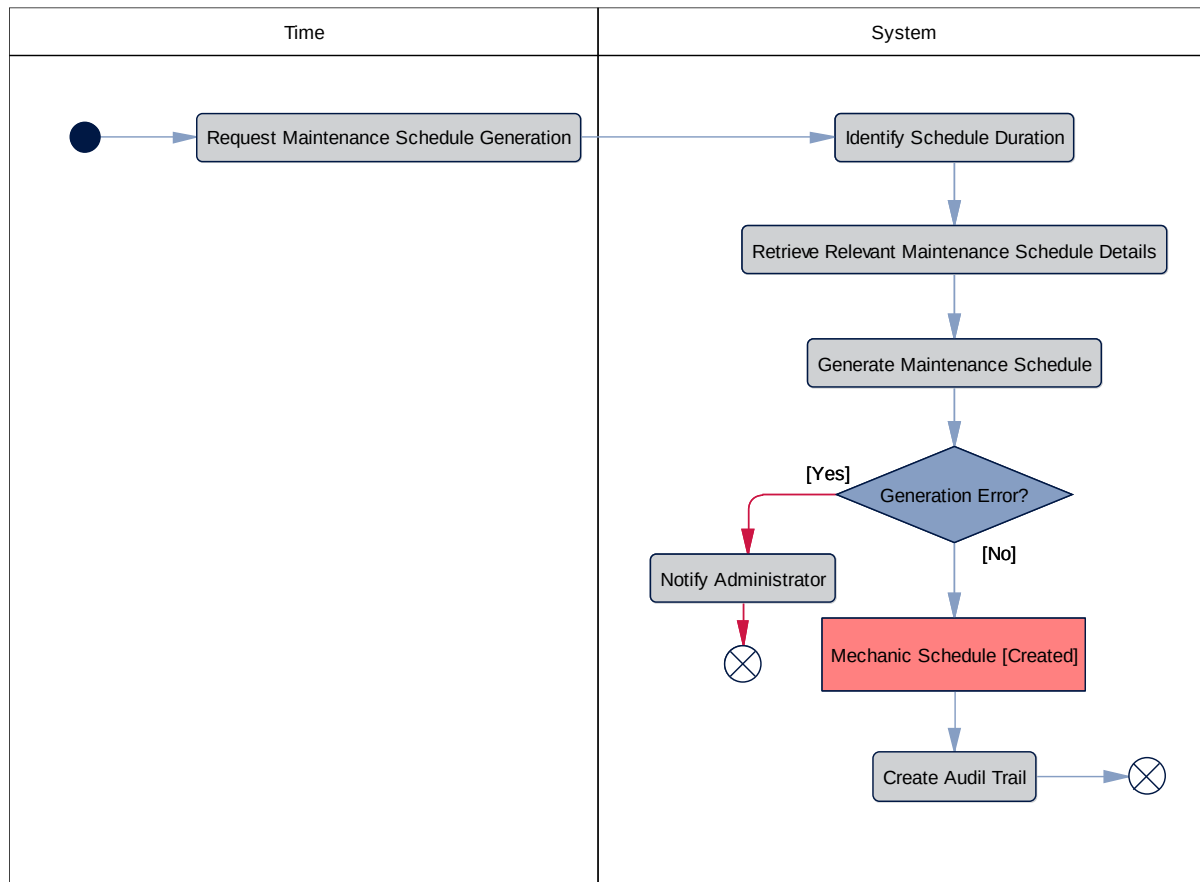


Figure 272 Generate Maintenance Schedule Activity Diagram



4.2.11.2 Generate Maintenance Requirements Report

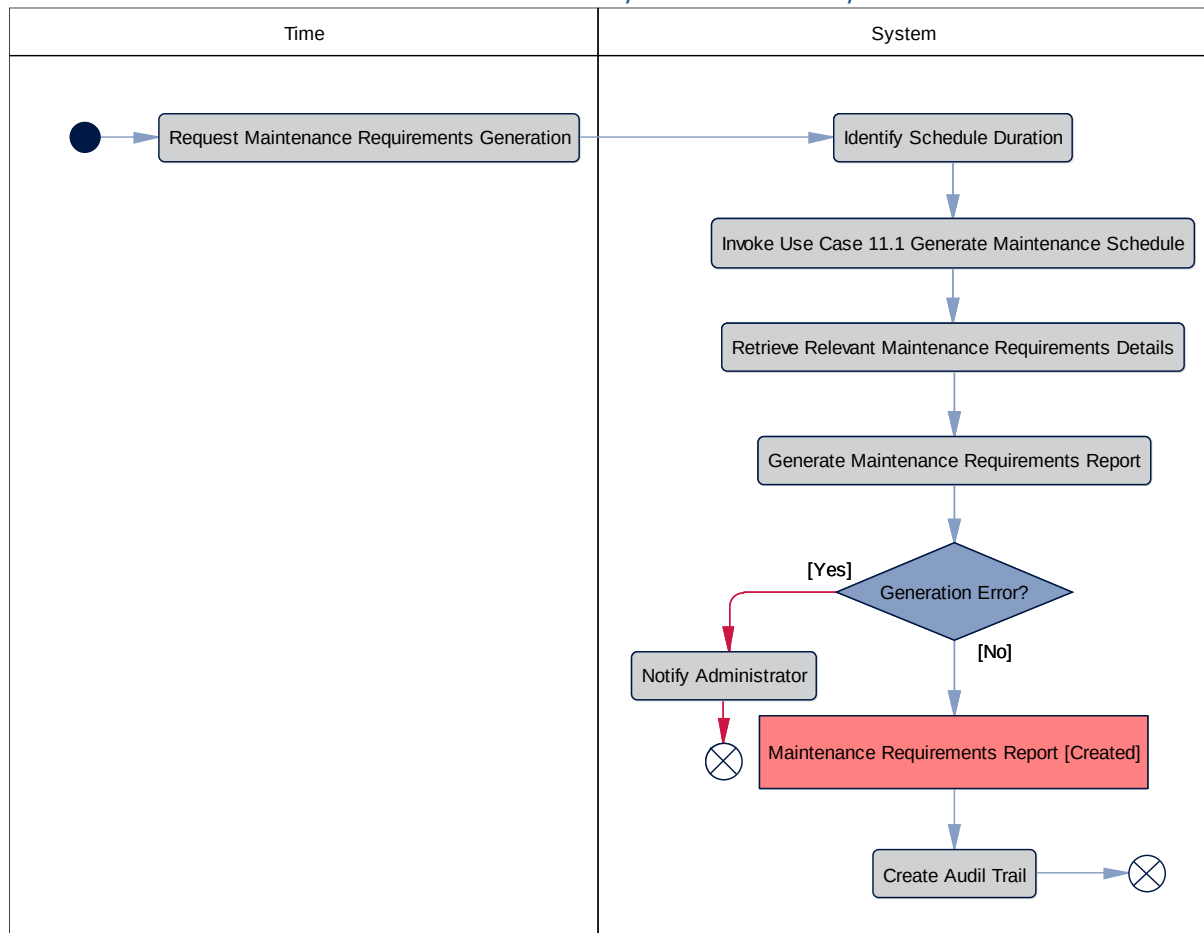


Figure 273 Generate Maintenance Requirements Report Activity Diagram



4.2.11.3 Generate Student Report

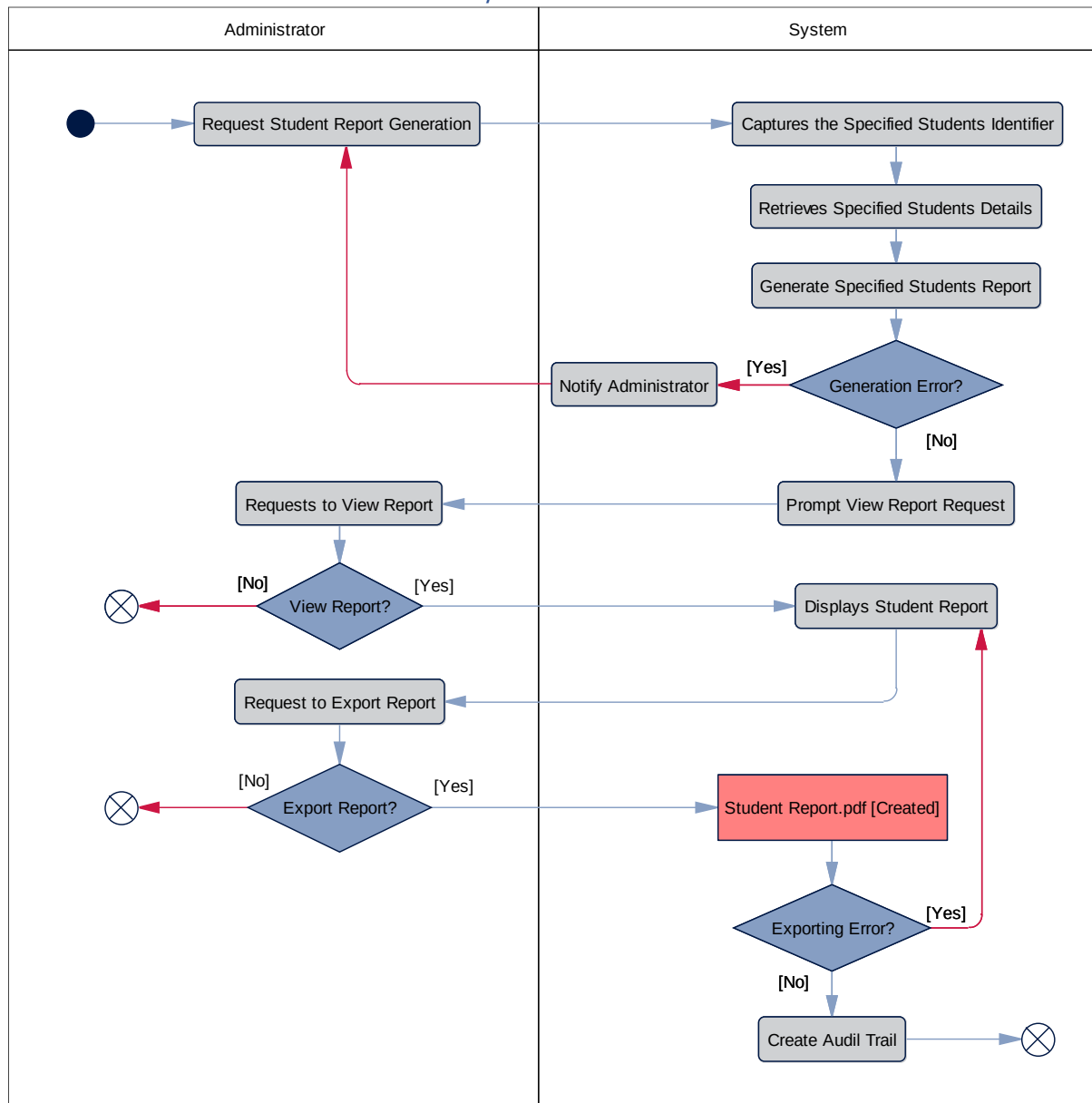


Figure 274 Generate Student Report Activity Diagram



4.2.11.4 Generate Bicycle History Report

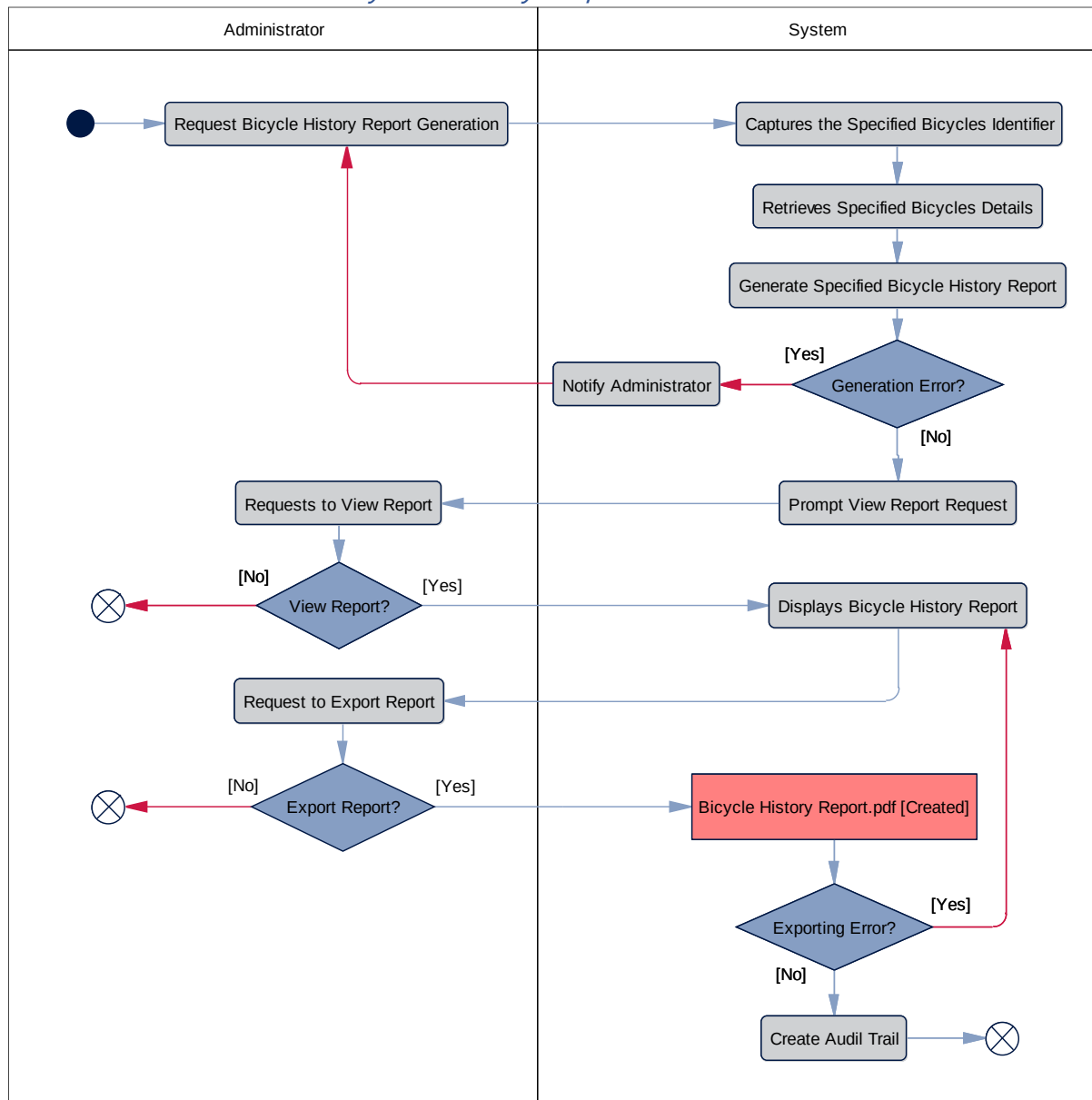


Figure 275 Generate Bicycle History Activity Diagram



4.2.11.5 Generate Log Report

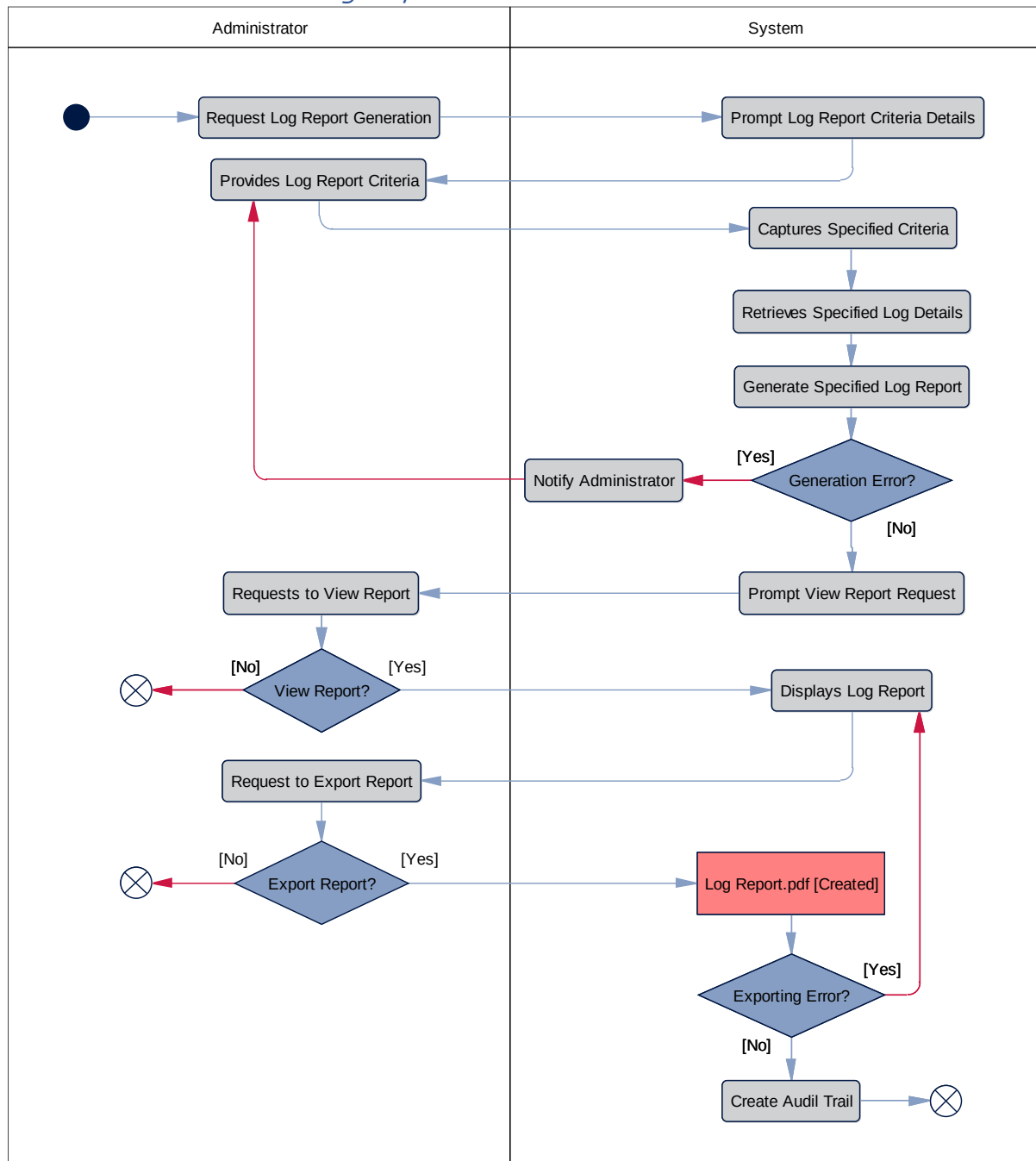


Figure 276 Generate Log Report Activity Diagram



4.2.11.6 Generate Parts Used Report

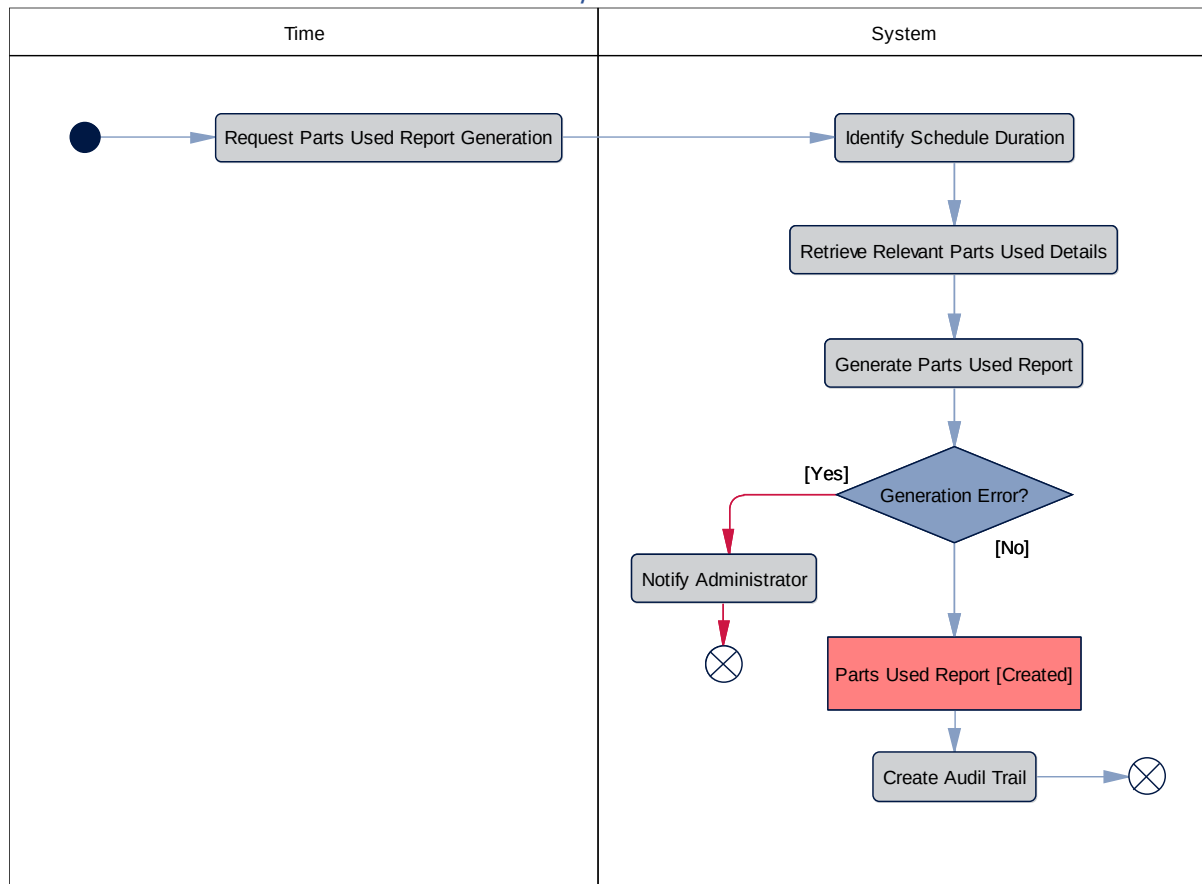


Figure 277 Generate Parts Used Report Activity Diagram



4.2.12 Sub-system 12: Notification

4.2.12.1 Send Student Did Not Arrive Notification

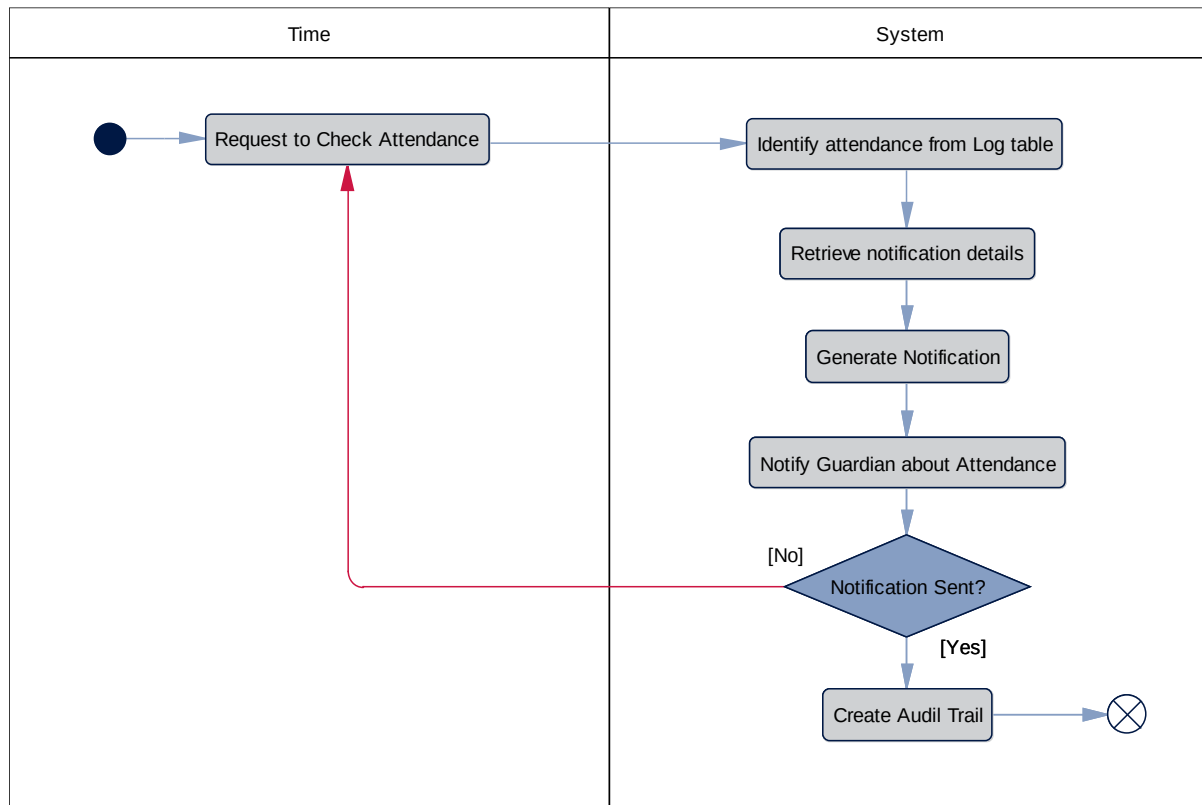


Figure 278 Send Student Did Not Arrive Notification Activity Diagram



4.2.12.2 Send Student Maintenance Notification

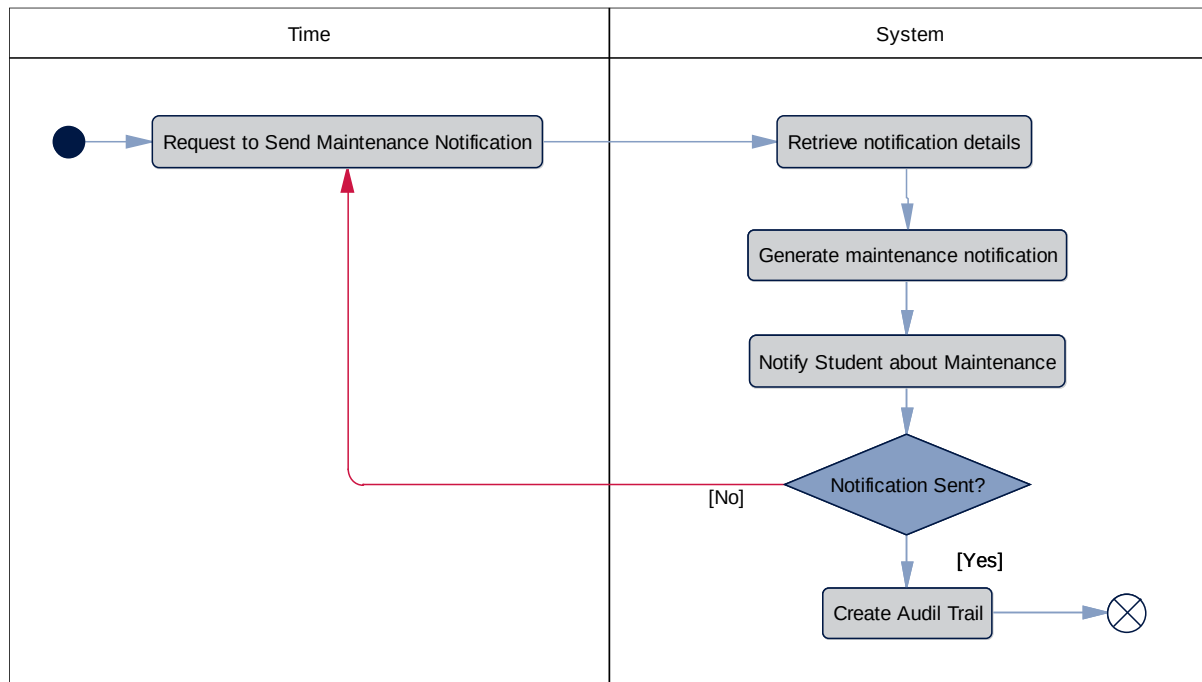


Figure 279 Send Student Maintenance Notification Activity Diagram



4.2.13 Sub-system 13: Administration

4.2.13.1 Backup Data

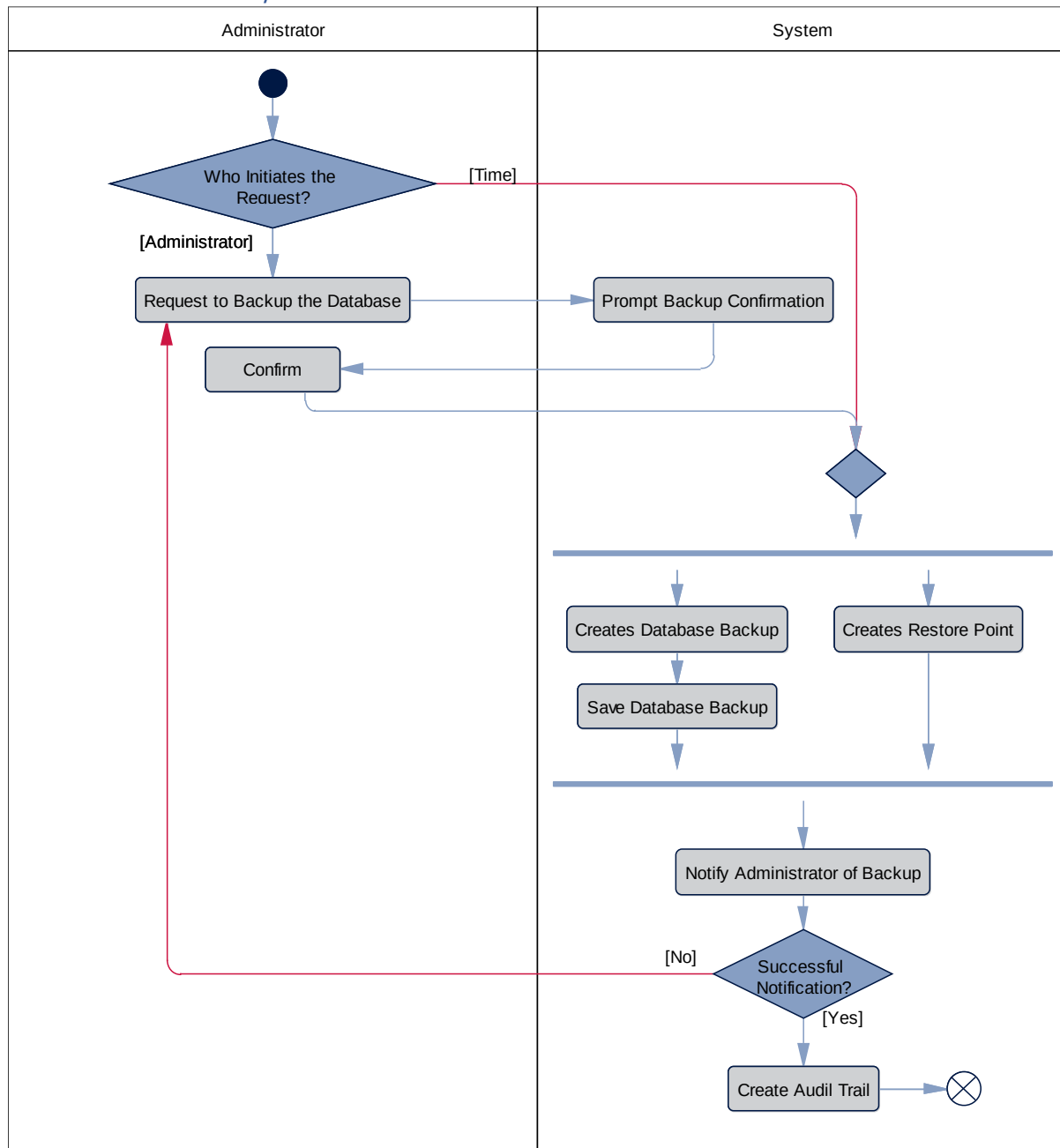


Figure 280 Backup Data Activity Diagram



4.2.13.2 Restore Data

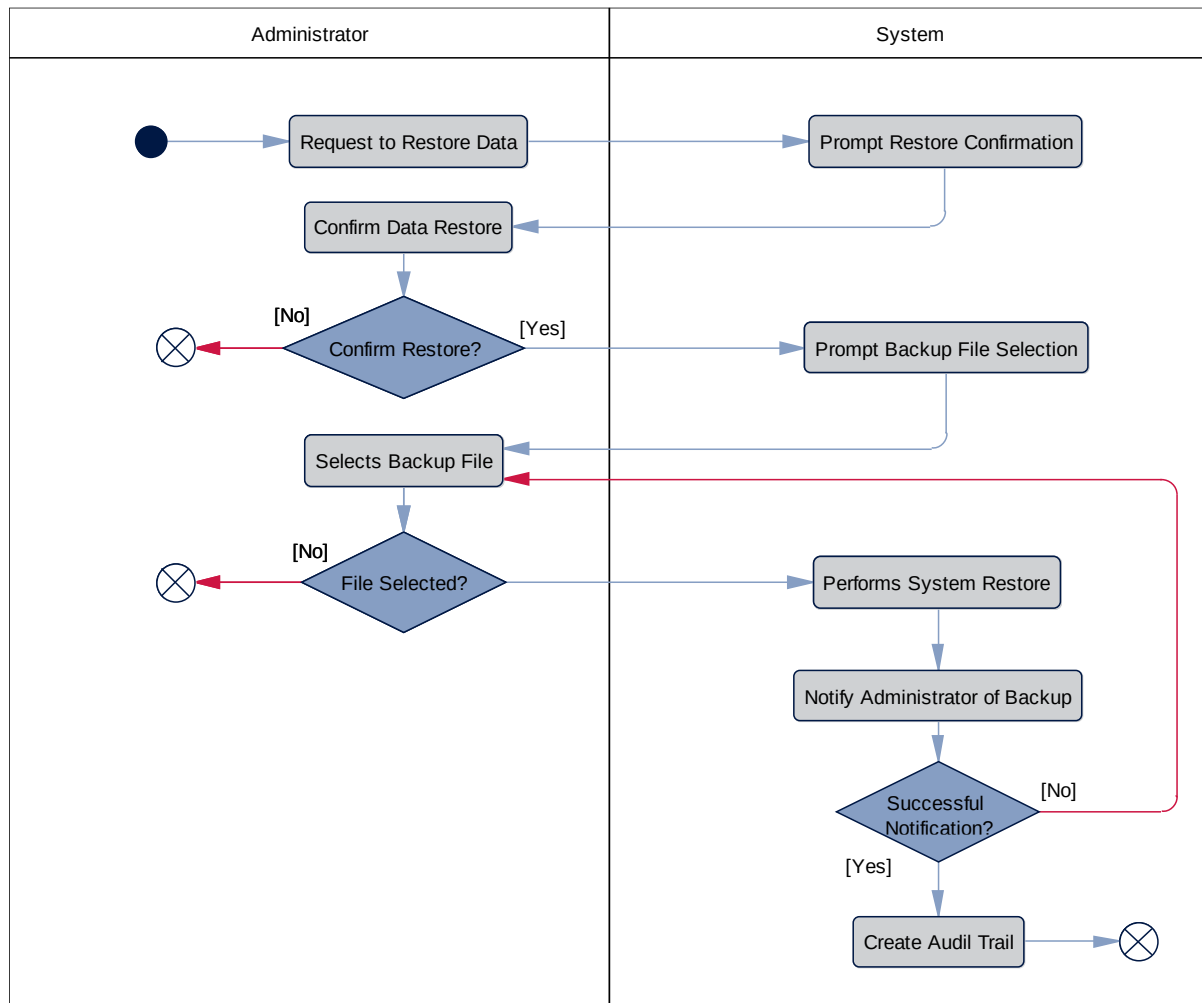


Figure 281 Restore Data Activity Diagram



4.2.13.3 View Audit Trail

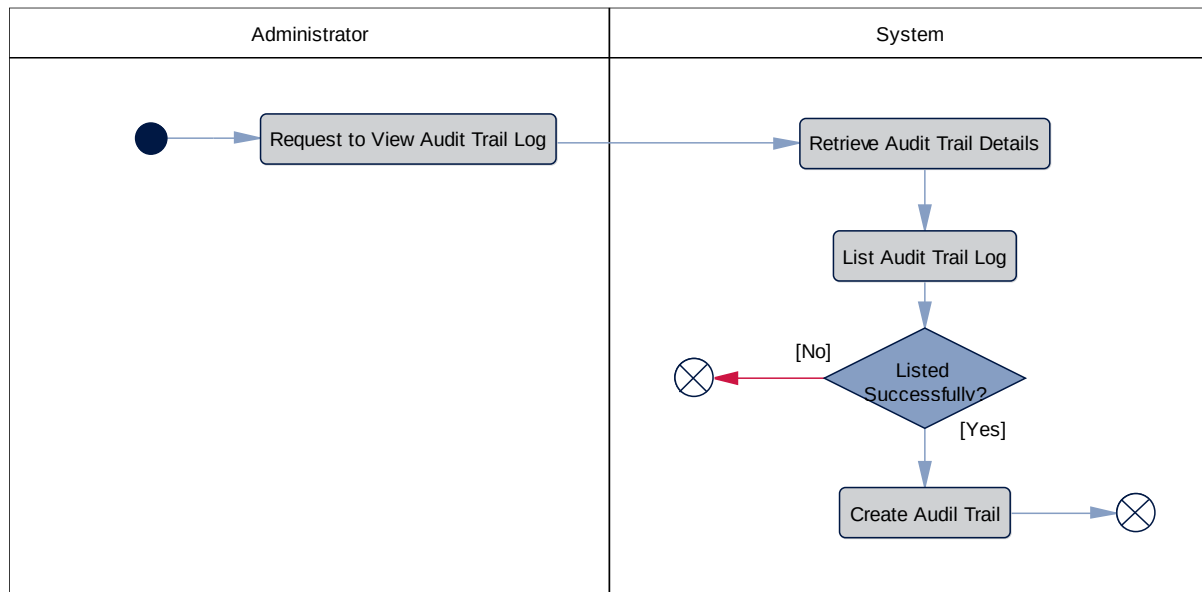


Figure 282 View Audit Trail Activity Diagram



4.2.13.4 Search Audit Trail Entry

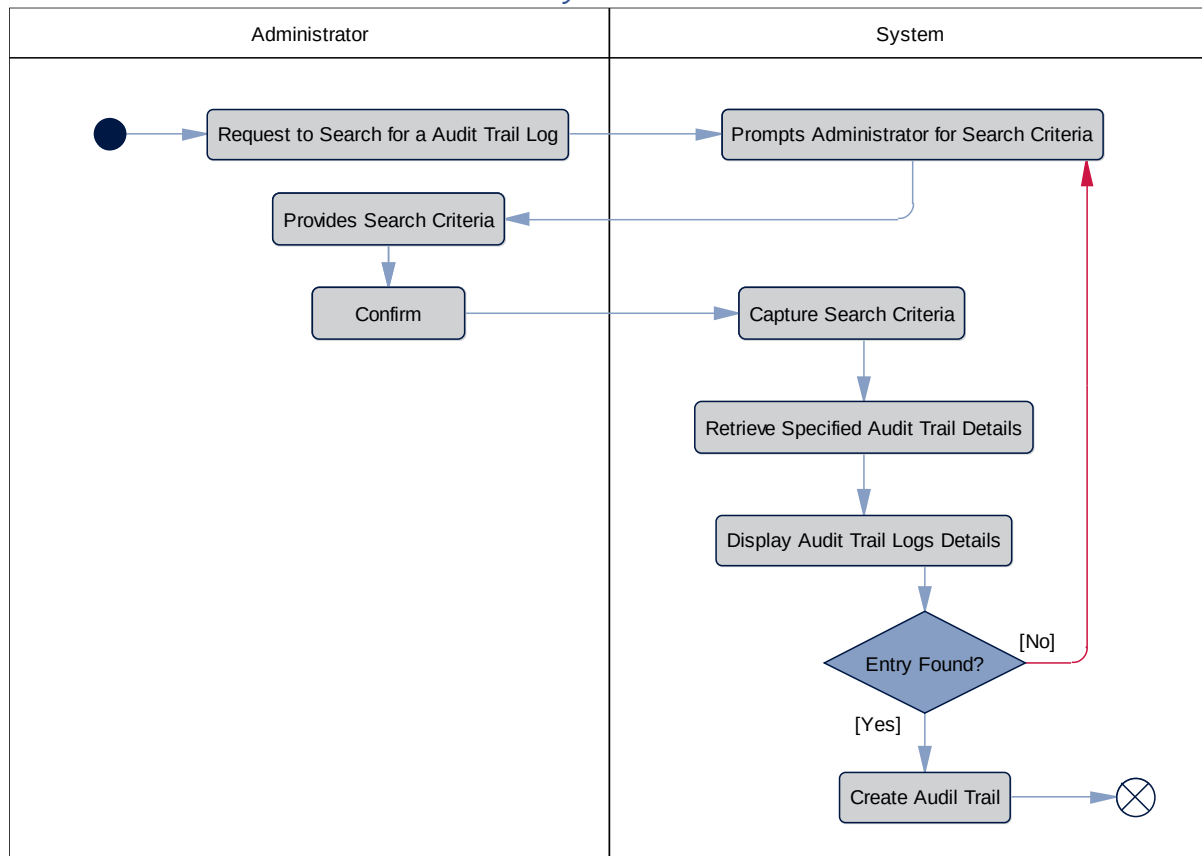


Figure 283 Search Audit Trail Entry Activity Diagram



4.2.13.5 Add Route

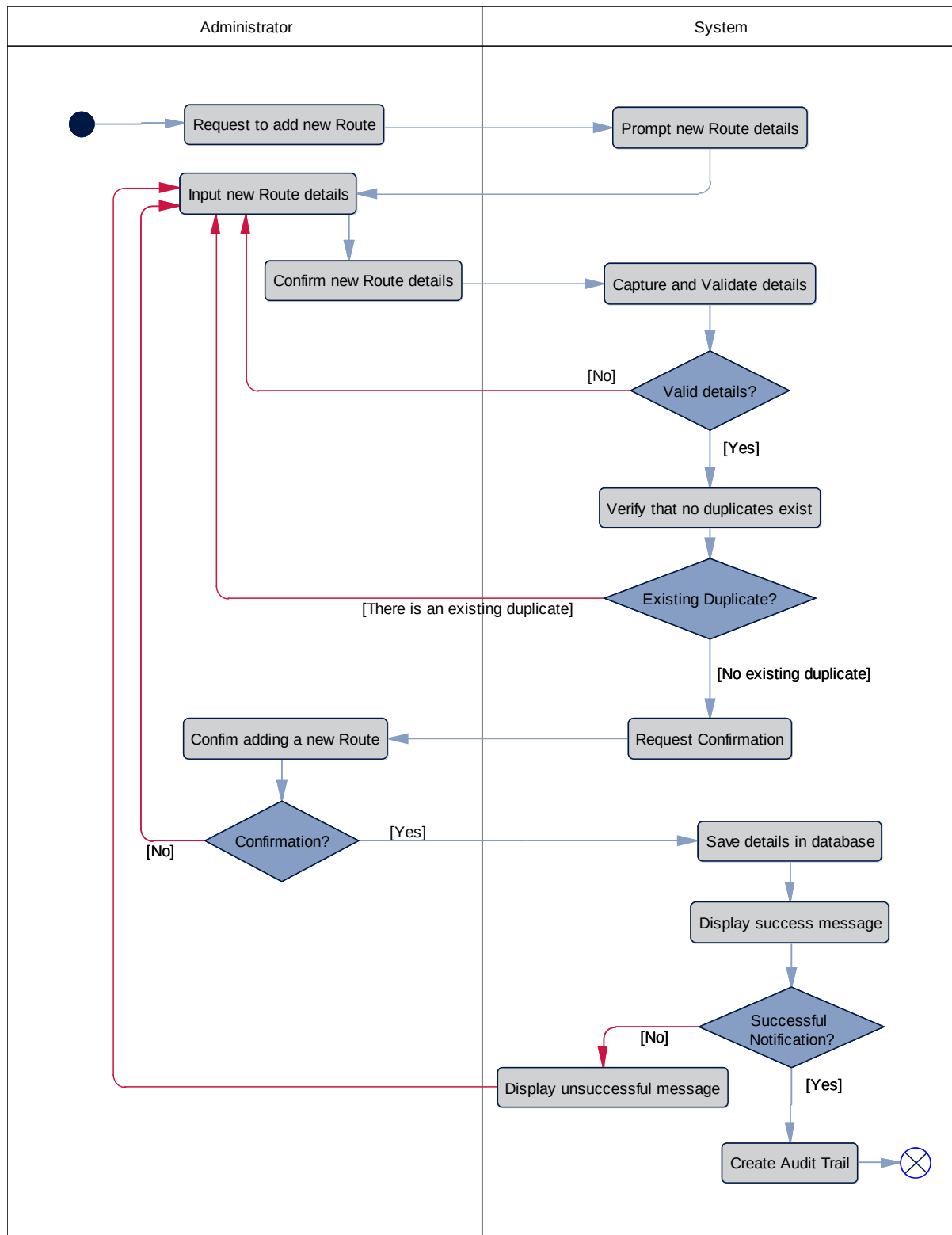


Figure 284 Add Route Activity Diagram



4.2.13.6 Search Route

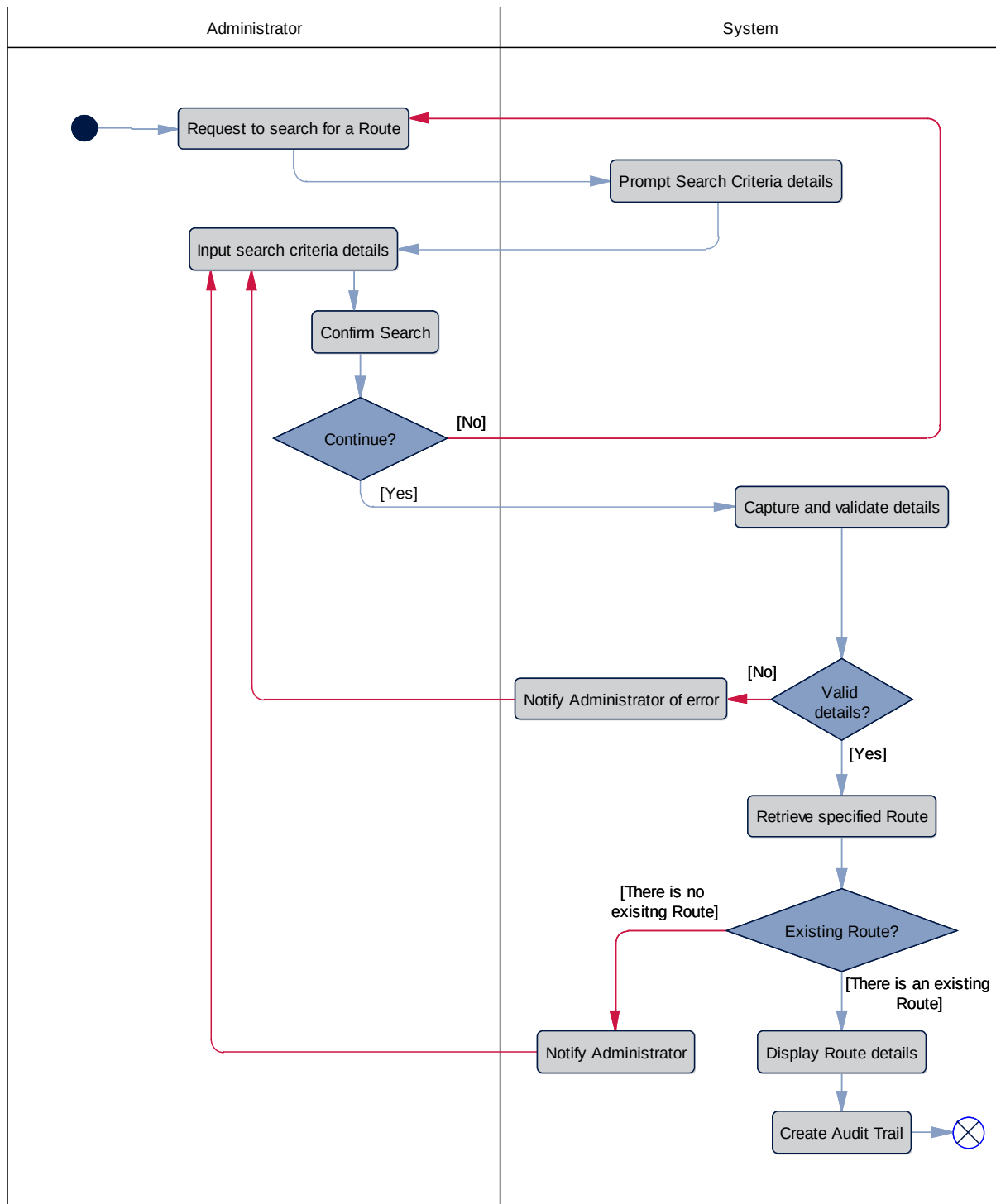


Figure 285 Search Route Activity Diagram



4.2.13.7 Update Route

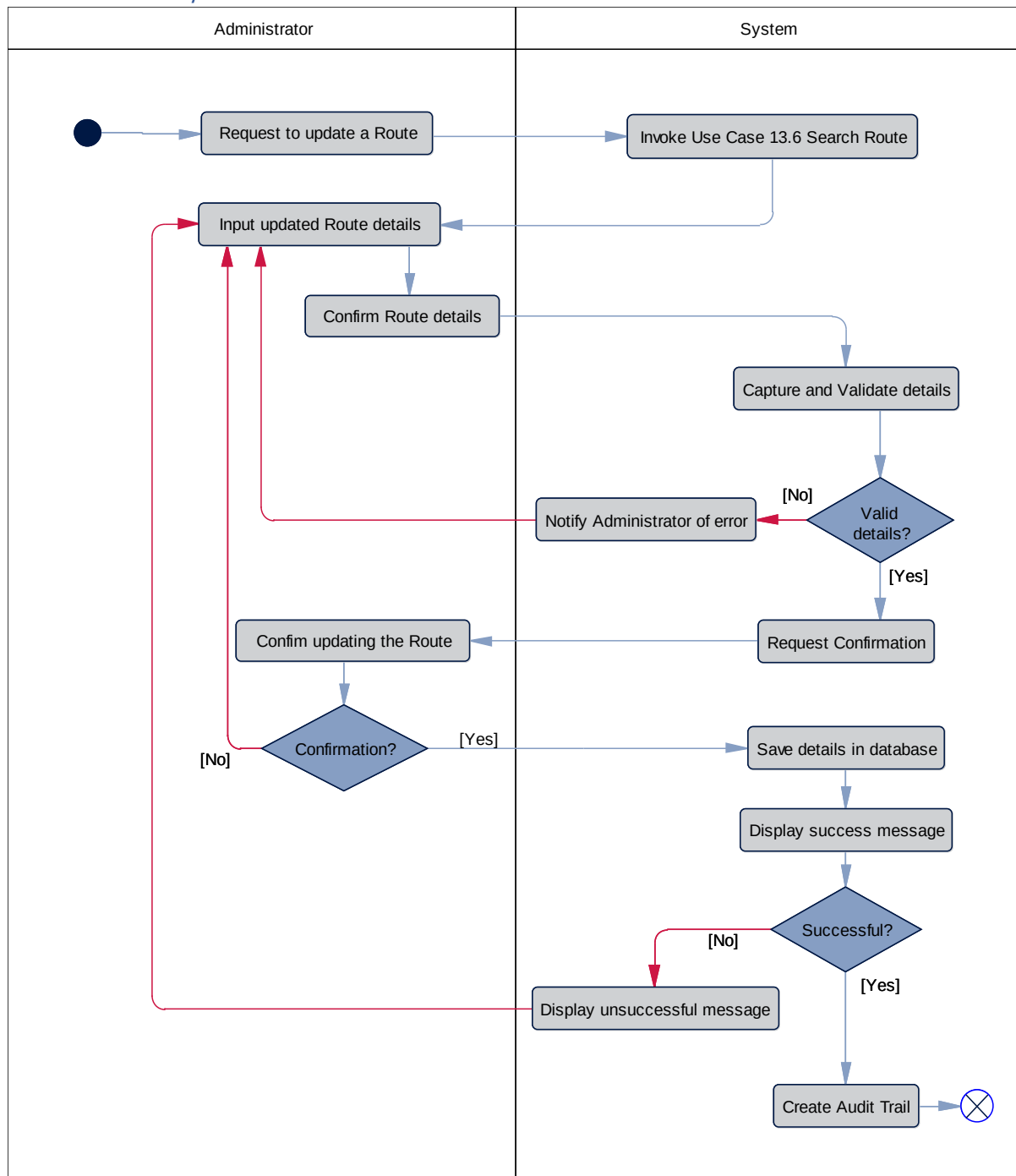


Figure 286 Update Route Activity Diagram



4.2.13.8 Delete Route

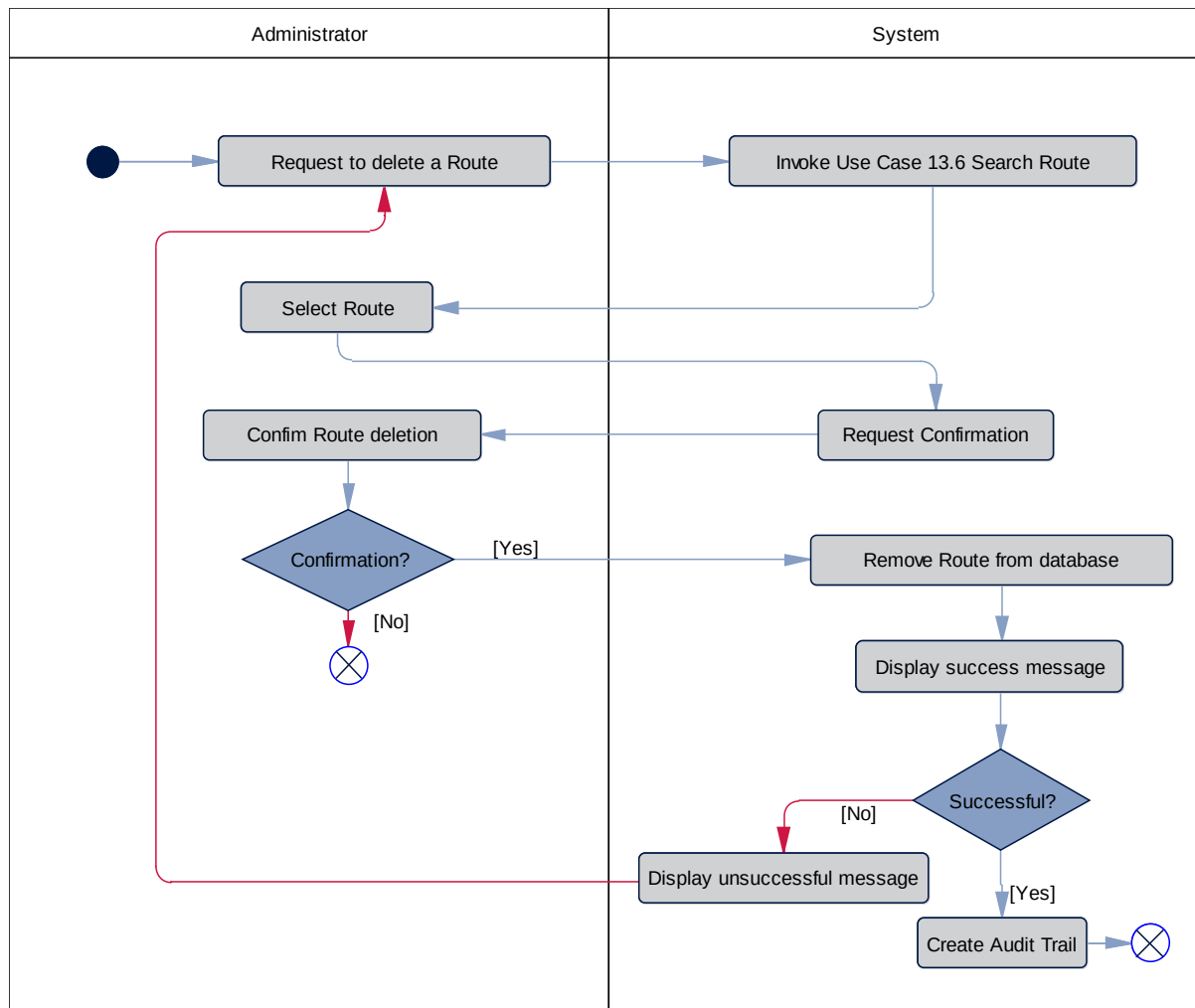


Figure 287 Delete Route Activity Diagram



4.3 Conclusion

After looking at the UML diagrams we can see how each and every single process flows from beginning to end including decisions.



5. Data Model

5.1 Introduction

The entity relationship diagram (ERD for short) is basically all the data stores of the system and essentially forms the backbone of the system as all data is going to be stored in tables that are created using the ERD. The ERD is a useful way to map out all the data stores which the system will have and allows us to see the relationships between them so that when we design the database, we will already have a blueprint for it.

5.2 Logical Entity Relationship Diagram

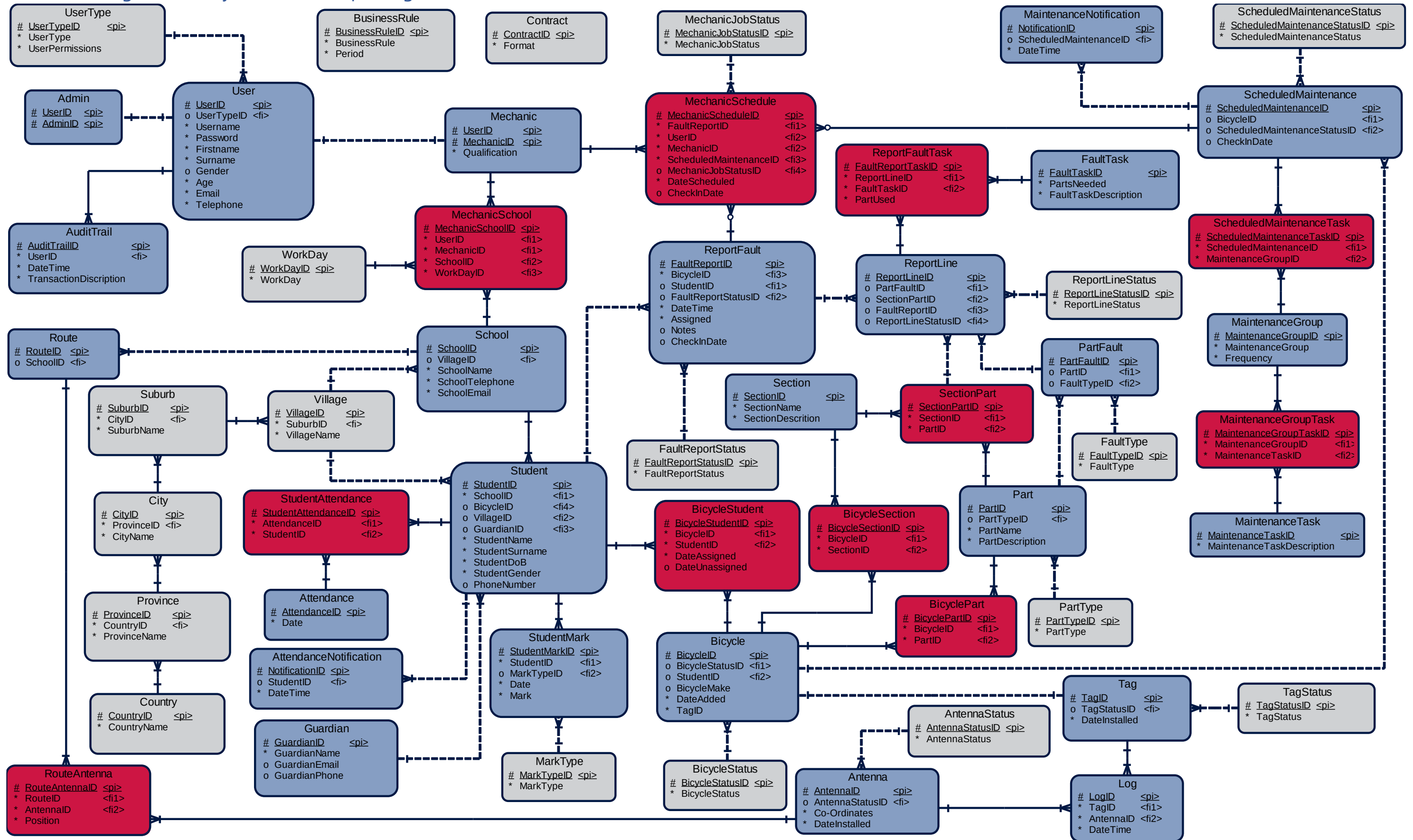


Figure 288 Logical Entity Relational Diagram

5.1 Conclusion

After designing our ERD we now have a much better understanding of the system overall and its technical aspects.



6. Interfaces and other inputs

6.1 Introduction

The Interfaces and other Inputs section of this deliverable contains a Table view. This Table details all the input flows into our system corresponding with the Context Diagram. It shows all the input interactions with the system interface and the basic actions that take place to trigger certain system activity. Purposes for each action flow and the Usage is shown clearly per flow. Tied together to this, are the Entities and Attributes related to that particular flow, in the Entity Relational Diagram. The triggering actors are also shown.

6.2 Interfaces and Inputs Table

Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
1.1 Login	1.1 Request to Login details	user requests to log in to the system.	When a registered user wants to log in to the system.	None	Requested by the user
	1.1 Input user Login details	user enters specific details that are used to log in.	When a user has requested to log in to the system.	Entities: user Attributes: username Password	Entered by the user
	1.1 [alt] Request forgot password details	User requests to reset their password if they forgot it	When a user cannot access their account due to a forgotten password.	None	Requested by the user
1.2 Logout	1.2 Request to Logout user details	user requests to Logout of the system.	When a user wants to Logout of the system.	None	Requested by the user
	1.2 Logout Confirmation accepted details	The user confirms the continuation of Logging out.	When a user has requested to Logout	None	Entered by the user
	1.2 [alt] Logout confirmation declined details	user declines the continuation of Logging out.	When a user has requested to Logout	None	Entered by the user
1.3 Forgot Password	1.3 user Email address details	user enters their Email address.	When a user has requested to reset their password.	Entities: user Attributes: Email	Entered by the user
	1.3 Select reset password link details	User requests to reset the password by selecting the link	When the user is trying to reset their password	None	Entered by the user
	1.3 [alt] Reset User password declined details	User declines to proceed with the password reset process.	When the user no longer wants to reset their password	None	Entered by the user
	1.3 New Password details	user enters their new password details.	When a user is setting a new password.	Entities: user	Entered by the user



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: Password	
	1.3 Reset user password accepted details	user accepts the new passwords details.	When a user has entered a new password.	Entities:	Entered by the user
	1.3 [alt] Reset user password declined details	user declines the new password details.	When a user has entered a new password.	None	Entered by the user
1.4 Add user	1.4 Request to add new user details	The Administrator captures a new user on the database.	Administrator requests to Add a user's information for a bicycle to be worked on by the mechanic.	None	Requested by the Administrator
	1.4 Input new user details	The Administrator enters the specific details of the user being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: user Attributes: username Password Firstname Surname Gender Age Email Telephone	Entered by the Administrator
	1.4 Add user confirmation accepted	Administrator confirms the Add user action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	1.4 [alt] Decline Add user confirmation details	Administrator declines or cancels the Add user action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
1.5 Search user	1.5 Request to search user details	The Administrator wishes to view specific user details and prompts this action.	Administrator requests to search for a specific user by triggering this process.	None	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	1.5 Input user search criteria details	Administrator searches for a specific user in the system.	It contains the criteria specified by the Administrator when requesting to view a specific user's details.	Entities: user Attributes: username Password Firstname Surname Gender Age Email Telephone	Entered by the Administrator
1.6 Update user	1.6 Request to Update user details	Administrator requests to edit a user's details	Administrator requests to edit user information to the required changes	None	Requested by the Administrator
	1.6 Input Updated user details	Administrator enters the specific details that are to change for the user being updated	Administrator inputs the changes to be made by filling in the required entries.	Entities: user Attributes: username Password Firstname Surname Gender Age Email Telephone	Entered by the Administrator
	1.6 Update user details confirmation accepted details	Administrator confirms the Update user action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	1.6 [alt] Decline Update user	Administrator declines or cancels the Update user action before that	Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	confirmation details	information captured is committed to the database.			
1.7 Delete user	1.7 Request to Delete user details	Administrator desires to remove a certain user from the system	The Administrator selects the user to be deleted via the search function.	Entities: user Attributes: username Password Firstname Surname Gender Age Email Telephone	Requested by the Administrator
	1.7 Accept Delete user confirmation details	Administrator declines or cancels the Delete user action before that removal is committed to the database.	Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
	1.7 [alt] Decline Delete user confirmation details	Administrator confirms the Delete user action before that information captured is committed to the database. Then the action is done.	The Administrator declines by selecting the related option on the screen, then the user and its related details get deleted from the system.	None	Entered by the Administrator
1.8 Add user Type	1.8 Request to add a new user Type details	The Administrator captures a new user Type on the database.	Administrator requests to Add a user Type's information for a bicycle to be worked on by the mechanic.	None	Requested by the Administrator
	1.8 New user Type details	The Administrator enters the specific details of the user Type being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: userType Attributes: userTypeID userType userTypePermissions	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	1.8 Add user Type confirmation accepted details	Administrator confirms the Add user Type action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	1.8 [alt] Add user type confirmation declined details	Administrator declines or cancels the Add user Type action before it is committed to the database.	Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
1.9 Search user Type	1.9 Request to search user Type details	The Administrator wishes to view specific user Type details and prompts this action.	Administrator requests to search for a specific user Type by triggering this process.	None	Requested by the Administrator
	1.9 Input user Type search criteria details	Administrator searches for a specific user Type in the system.	It contains the criteria specified by the Administrator when requesting to view a specific user Type's details.	Entities: userType Attributes: userId userTypePermissions	Entered by the Administrator
1.10 Update user Type	1.10 Request to Update user type details	Administrator requests to edit a user Type's details	Administrator requests to edit user Type information to the required changes	None	Requested by the Administrator
	1.10 Input Updated user type details	Administrator enters the specific details that are to change for the user Type being updated	Administrator inputs the changes to be made by filling in the required entries.	Entities: userType Attributes: userId userTypePermissions	Entered by the Administrator
	1.10 Update user Type confirmation accepted details	Administrator confirms the Update user Type action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	1.10 [alt] Decline Update user Type confirmation details	Administrator declines or cancels the Update user Type action before that information captured is committed to the database.	Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
1.11 Delete user Type	1.11 Request to Delete user Type details	Administrator desires to remove a certain user Type from the system	The Administrator selects the user Type to be deleted via the search function.	Entities: userType Attributes: userTypeID userType userTypePermissions	Requested by the Administrator
	1.11 Accept Delete user type details	Administrator declines or cancels the Delete user Type action before that removal is committed to the database.	Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
	1.11 [alt] Decline Delete user Type confirmation details	Administrator confirms the Delete user Type action before that information captured is committed to the database. Then the action is done.	The Administrator declines by selecting the related option on the screen, then the user Type and its related details get deleted from the system.	None	Entered by the Administrator
2.1 Complete Job	2.1 Request to complete Job details	Mechanic requests to complete a job.	When a mechanic has completed a job.	None	Requested by the Mechanic
	2.1 Select specific Job details	Mechanic selects a specific job to complete.	When a mechanic has requested to complete a job.	Entities: MechanicSchedule ReportFault ReportLine ScheduledMaintenance Attributes: MechanicScheduleID FaultReportID MechanicID	Entered by the Mechanic



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				ScheduledMaintenanceID MechanicJobStatusID BicycleID ReportLineID ReportLineStatusID	
	2.1 Required Part used confirmation accepted	The mechanic confirms that a part was used.	When a mechanic has used the part required to complete the job.	None	Entered by the Mechanic
2.2 View Job Tasks	2.2 Request to view Job tasks details	Mechanic requests to view all the tasks that make up a job.	When a mechanic wants to view all the tasks that are required to complete a job.	None	Requested by the Mechanic
	2.2 Input search criteria details	Mechanic wants to view all the job tasks for a specific job.	When a mechanic is looking for a specific job based on certain criteria.	Entities: MechanicSchedule ReportFault ReportLine ScheduledMaintenance Attributes: MechanicScheduleID FaultReportID ReportLineID ReportLineStatusID	Entered by the Mechanic
3.1 Add Antenna	3.1 Request to add new Antenna details	The Administrator captures a new Antenna on the database.	Administrator requests to Add an Antenna's information.	None	Requested by the Administrator
	3.1 New Antenna details	The Administrator enters the specific details of the Antenna being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: Antenna AntennaStatus Attributes: AntennaID AntennaStatusID	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				AntennaStatus Co-ordinates DateInstalled	
	3.1 Add Confirmation accepted	The Administrator confirms the Add Antenna action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	3.1 [alt] Add Confirmation declined	Administrator declines or cancels the Add Antenna action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
3.2 Search Antenna	3.2 Request to search antenna details	The Administrator wishes to view specific Antenna details and prompts this action.	Administrator requests to search for a specific Antenna by triggering this process.	None	Requested by the Administrator
	3.2 Input Antenna search criteria details	Administrator searches for a specific Antenna in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Antenna's details.	Entities: Antenna AntennaStatus Attributes: AntennaID AntennaStatusID AntennaStatus Co-ordinates DateInstalled	Entered by the Administrator
3.3 Update Antenna	3.3 Request to Update Antenna details	Administrator requests to edit an Antenna's details.	Administrator requests to edit Antenna information to the required changes	None	Requested by the Administrator
	3.3 Input Updated Antenna details	The Administrator enters the specific details that are to change for the Antenna being updated.	Administrator inputs the changes to be made by filling in the required entries.	Entities: Antenna AntennaStatus Attributes: AntennaID AntennaStatusID AntennaStatus Co-ordinates	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				DateInstalled	
	3.3 Update Antenna confirmation accepted	Administrator confirms the Update Antenna action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	3.3 [alt] Decline Update Antenna confirmation	Administrator declines or cancels the Update Antenna action before that information captured is committed to the database.	Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
3.4 Delete Antenna	3.4 Request to delete Antenna details	Administrator desires to remove a certain Antenna from the system	The Administrator selects the Antenna to be deleted via the search function.	Entities: Antenna AntennaStatus Attributes: AntennaID AntennaStatusID AntennaStatus Co-ordinates DateInstalled	Requested by the Administrator
	3.4 Accept Delete Antenna confirmation	The Administrator confirms the Delete Antenna action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Antenna and its related details get deleted from the system.	None	Entered by the Administrator
	3.4 [alt] Decline Delete Antenna confirmation	Administrator declines or cancels the Delete Antenna action before that removal is committed to the database.	Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
3.5 Add Tag	3.5 Request to add new Tag	The Administrator captures a new Tag on the database.	Administrator requests to Add a Tag's information.	None	Requested by the Administrator
	3.5 New Tag details	The Administrator enters the specific details of the Tag being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: Tag TagStatus	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: TagID TagStatusID TagStatus DateInstalled	
	3.5 Add Tag confirmation accepted	The Administrator confirms the Add Tag action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	3.5 [alt] Decline Add Tag confirmation	Administrator declines or cancels the Add Tag action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
3.6 Search Tag	3.6 Request to search Tag details	The Administrator wishes to view specific Tag details and prompts this action.	Administrator requests to search for a specific Tag by triggering this process.	None	Requested by the Administrator
	3.6 Input Tag search criteria details	The Administrator searches for a specific Tag in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Tag's details.	Entities: Tag TagStatus Attributes: TagID TagStatusID TagStatus DateInstalled	Entered by the Administrator
3.7 Update Tag	3.7 Request to update Tag details	The Administrator requests to edit a Tag's details.	Administrator requests to edit Tag information to the required changes	None	Requested by the Administrator
	3.7 Input Updated Tag details	The Administrator enters the specific details that are to change for the Tag being updated.	Administrator inputs the changes to be made by filling in the required entries.	Entities: Tag TagStatus Attributes: TagID TagStatusID TagStatus	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				DateInstalled	
	3.7 Update Tag details confirmation accepted	Administrator confirms the Update Tag action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	3.7 [alt] Decline Update Tag confirmation	The Administrator declines or cancels the Update Tag action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
3.8 Delete Tag	3.8 Request to delete Tag details	The Administrator desires to remove a certain Tag from the system	The Administrator selects the Tag to be deleted via the search function.	Entities: Tag TagStatus Attributes: TagID TagStatusID TagStatus DateInstalled	Requested by the Administrator
	3.8 Accept Delete Tag confirmation	The Administrator confirms the Delete Tag action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Tag and its related details get deleted from the system.	None	Entered by the Administrator
	3.8 [alt] Decline Delete Tag confirmation	The Administrator declines or cancels the Delete Tag action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
4.1 Add Student	4.1 Request to add new Student details	The Administrator captures a new Student on the database.	The Administrator requests to Add a Student's information.	None	Requested by the Administrator
	4.1 New Student details	The Administrator enters the specific details of the Student being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: Student Guardian	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: StudentID SchoolID GuardianID StudentName StudentSurname StudentDoB StudentGender GuardianName GuardianEmail GuardianPhone	
	4.1 Add Student confirmation accepted	The Administrator confirms the Add Student action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	4.1 [alt] Decline Add Student confirmation	The Administrator declines or cancels the Add Student action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
4.2 Search Student	4.2 Request to search Student details	The Administrator wishes to view specific Student details and prompts this action.	The Administrator requests to search for a specific Student by triggering this process.	None	Requested by the Administrator
	4.2 Input Student search criteria details	The Administrator searches for a specific Student in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Student's details.	Entities: Student Guardian Attributes: StudentID SchoolID GuardianID StudentName StudentSurname StudentDoB StudentGender GuardianName	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				GuardianEmail GuardianPhone	
4.3 Update Student	4.3 Request to update Student details	The Administrator requests to edit a Student's details.	The Administrator requests to edit Student information to the required changes	None	Requested by the Administrator
	4.3 Input Updated Student details	The Administrator enters the specific details that are to change for the Student being updated.	The Administrator inputs the changes to be made by filling in the required entries.	Entities: Student Guardian Attributes: StudentID SchoolID GuardianID StudentName StudentSurname StudentDoB StudentGender GuardianName GuardianEmail GuardianPhone	Entered by the Administrator
	4.3 Update Student details confirmation accepted	The Administrator confirms the Update Student action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	4.3 [alt] Decline Update Student confirmation	The Administrator declines or cancels the Update Student action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
4.4 Delete Student	4.4 Request to delete Student details	The Administrator desires to remove a certain Student from the system	The Administrator selects the Student to be deleted via the search function.	Entities: Student Guardian Attributes: StudentID	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				SchoolID GuardianID StudentName StudentSurname StudentDoB StudentGender GuardianName GuardianEmail GuardianPhone	
	4.4 Accept Delete Student confirmation	The Administrator confirms the Delete Student action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Student and related details get deleted from the system.	None	Entered by the Administrator
	4.4 [alt] Decline Delete Student confirmation	The Administrator declines or cancels the Delete Student action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
4.5 Capture Student Marks	4.5 Request to capture a Student's Mark details	The Administrator captures a Student's marks on the database.	The Administrator requests to capture a Student's Marks' information.	None	Requested by the Administrator
	4.5 New Student Mark details	The Administrator enters the specific details of the Student's Marks being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: StudentMark MarkType Attributes: StudentMarkID StudentID MarkTypeID Date Mark MarkType	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	4.5 Add Student Mark confirmation accepted	The Administrator confirms the Add Student Mark action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	4.5 [alt] Decline Add Student Mark confirmation	The Administrator declines or cancels the Add Student Mark action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
4.6 Search Student Marks	4.6 Request to search Student mark details	The Administrator wishes to view specific Student Marks details and prompts this action.	The Administrator requests to search for a specific Student Marks by triggering this process.	None	Requested by the Administrator
	4.6 Input Student Mark search criteria details	The Administrator searches for a specific Student Marks's in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Student's Mark details.	Entities: StudentMark MarkType Attributes: StudentMarkID StudentID MarkTypeID Date Mark MarkType	Entered by the Administrator
4.7 Add Mark Type	4.7 Request to add new Mark Type details	The Administrator captures a new Mark Type on the database.	The Administrator requests to Add a Mark Type information.	None	Requested by the Administrator
	4.7 Input New Mark Type details	The Administrator enters the specific details of the Mark Type being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: MarkType Attributes: MarkTypeID MarkType	Entered by the Administrator
	4.7 Add Mark Type	The Administrator confirms the Add Mark Type action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	confirmation accepted details				
	4.7 [alt] Add Mark Type confirmation decline details	The Administrator declines or cancels the Add Mark Type action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
4.8 Search Mark Type	4.8 Request to search Mark Type details	The Administrator wishes to view specific Mark Type details and prompts this action.	The Administrator requests to search for a specific Mark Type by triggering this process.	None	Requested by the Administrator
	4.8 Input Mark Type search criteria details	The Administrator searches for a specific Mark Type in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Mark Types' details.	Entities: MarkType Attributes: MarkTypeID MarkType	Entered by the Administrator
4.9 Update Mark Type	4.9 Request to update Mark Type details	The Administrator requests to edit a Mark Type's details.	The Administrator requests to edit Mark Type information to the required changes	None	Requested by the Administrator
	4.9 Input Updated Mark Type details	The Administrator enters the specific details that are to change for the Mark Type being updated.	The Administrator inputs the changes to be made by filling in the required entries.	Entities: MarkType Attributes: MarkTypeID MarkType	Entered by the Administrator
	4.9 Update Mark Type details confirmation accepted details	The Administrator confirms the Update Mark Type action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	4.9 [alt] Update Mark Type confirmation declined details	The Administrator declines or cancels the Update Mark Type action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	4.10 Request to delete Mark Type details	The Administrator desires to remove a certain Mark Type from the system	The Administrator selects the Mark Type to be deleted via the search function.	Entities: MarkType Attributes: MarkTypeID MarkType	Requested by the Administrator
	4.10 Delete Mark Type confirmation accepted details	The Administrator confirms the Delete Mark Type action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Student and related details get deleted from the system.	None	Entered by the Administrator
	4.10 Delete Mark Type confirmation declined details	The Administrator declines or cancels the Delete Mark Type action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
5.1 Add Bicycle	5.1 Request to Add new Bicycle details	The Administrator captures a new Bicycle on the database.	The Administrator requests to Add a Bicycle's information.	None	Requested by the Administrator
	5.1 New Bicycle details	The Administrator enters the specific details of the Bicycle being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: Bicycle BicycleStatus Attributes: BicycleID BicycleStatusID BicycleStatus BicycleMake DateAdded	Entered by the Administrator
	5.1 Add Bicycle confirmation accepted details	The Administrator confirms the Add Bicycle action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	5.1 [alt] Add Bicycle	The Administrator declines or cancels the Add Bicycle action	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	Confirmation declined details	before it is committed to the database.			
5.2 Search Bicycle	5.2 Request to search Bicycle details	The Administrator wishes to view specific Bicycle details and prompts this action.	The Administrator requests to search for a specific Bicycle by triggering this process.	None	Requested by the Administrator
	5.2 Input Bicycle search criteria details	The Administrator searches for a specific Bicycle in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Bicycle's details.	Entities: Bicycle BicycleStatus Attributes: BicycleID BicycleStatusID BicycleStatus BicycleMake DateAdded	Entered by the Administrator
5.3 Update Bicycle	5.3 Request to Update Bicycle details	The Administrator requests to edit a Bicycle's details.	The Administrator requests to edit Bicycle information to the required changes	None	Requested by the Administrator
	5.3 Updated Bicycle details	The Administrator enters the specific details that are to change for the Bicycle being updated.	The Administrator inputs the changes to be made by filling in the required entries.	Entities: Bicycle BicycleStatus Attributes: BicycleID BicycleStatusID BicycleStatus BicycleMake DateAdded	Entered by the Administrator
	5.3 Update Bicycle confirmation accepted details	The Administrator confirms the Update Bicycle action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	5.3 [alt] Update Bicycle confirmation declined details	The Administrator declines or cancels the Update Bicycle action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
5.4 Delete Bicycle	5.4 Request to delete Bicycle details	The Administrator desires to remove a certain Bicycle from the system	The Administrator selects the Bicycle to be deleted via the search function.	Entities: Bicycle BicycleStatus Attributes: BicycleID BicycleStatusID BicycleStatus BicycleMake DateAdded	Requested by the Administrator
	5.4 Delete Bicycle confirmation accepted details	The Administrator confirms the Delete Bicycle action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Bicycle and its related details get deleted from the system.	None	Entered by the Administrator
	5.4 [alt] Delete Bicycle confirmation declined details	The Administrator declines or cancels the Delete Bicycle action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
5.5 Assign a Bicycle to Student	5.5 Request to assign Bicycle details	The Administrator requests to assign a bicycle to a student.	When a student needs to get a bicycle assigned to them.	None	Requested by the Administrator
	5.5 Assign option selected details	The Administrator selects which student the bicycle will be assigned to.	When the Administrator has decided which student should get a bicycle.	Entities: Bicycle Student BicycleStudent Attributes: BicycleID StudentID	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				BicycleStudentID	
5.6 Unassign Bicycle to Student	5.6 Request to Unassign Bicycle details	The Administrator requests to unassign a bicycle from a student.	When a student needs to get a bicycle unassigned from them.	None	Requested by the Administrator
	5.6 Unassign option selected clicked details	The Administrator selects which student the bicycle will be unassigned from.	When the Administrator has decided which student should have their bicycle unassigned.	Entities: Bicycle Student BicycleStudent Attributes: BicycleID StudentID BicycleStudentID	Entered by the Administrator
5.7 Assign Tag to bicycle	5.7 Request to assign Tag	The Administrator requests to assign a tag to a bicycle.	When a bicycle needs to get a tag assigned to it.	None	Requested by the Administrator
	5.7 Assign option selected clicked details	The Administrator selects which bicycle the tag will be assigned to.	When the Administrator has decided which bicycle should get a tag.	Entities: Bicycle Tag Attributes: BicycleID TagID	Entered by the Administrator
5.8 Unassign Tag from Bicycle	5.8 Request to Unassign Tag details	The Administrator requests to unassign a tag from a bicycle.	When a bicycle needs to get a tag unassigned from it.	None	Requested by the Administrator
	5.8 Unassign option selected clicked details	The Administrator selects which bicycle the tag will be unassigned from.	When the Administrator has decided which bicycle should have its tag unassigned.	Entities: Bicycle Tag Attributes: BicycleID TagID	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
6.1 Add Bicycle Part	6.1 Request to add a new BicyclePart details	The Administrator captures a new Bicycle Part on the database.	Administrator requests to Add a Bicycle Part's information for a bicycle.	None	Requested by the Administrator
	6.1 New Bicycle Part details	The Administrator enters the specific details of the Bicycle Part being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: Part PartType SectionPart Section Attributes: PartID PartTypeID PartName PartDescription PartType SectionName SectionPartID SectionID	Entered by the Administrator
	6.1 Add Confirmation accepted	The Administrator confirms the Add Bicycle Part action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	6.1 [alt] Add Confirmation declined	The Administrator declines or cancels the Add Bicycle Part action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
6.2 Search Bicycle Part	6.2 Request to Search Bicycle Part details	The Administrator wishes to view specific Bicycle Part details and prompts this action.	Administrator requests to search for a specific Bicycle Part by triggering this process.	None	Requested by the Administrator
	6.2 Input Bicycle Part Search criteria details	Administrator searches for a specific Bicycle Part in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Bicycle Part's details.	Entities: Part PartType SectionPart Section	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: PartID PartTypeID PartName PartDescription PartType SectionName SectionPartID SectionID	
6.3 Update Bicycle Part	6.3 Request to Update Bicycle Part details	The Administrator requests to edit a Bicycle Part's details	The Administrator requests to edit Bicycle Part information to the required changes	None	Requested by the Administrator
	6.3 Input Updated Bicycle Part details	The Administrator enters the specific details that are to change for the Bicycle Part being updated	The Administrator inputs the changes to be made by filling in the required entries.	Entities: Part PartType SectionPart Section Attributes: PartID PartTypeID PartName PartDescription PartType SectionName SectionPartID SectionID	Entered by the Administrator
	6.3 Update Bicycle Part confirmation accepted	The Administrator confirms the Update Bicycle Part action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	6.3 [alt] Update Bicycle Part confirmation declined	The Administrator declines or cancels the Update Bicycle Part action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
6.4 Delete Bicycle Part	6.4 Request to Delete Bicycle Part details	The Administrator desires to remove a certain Bicycle Part from the system	The Administrator selects the Bicycle Part to be deleted via the search function.	Entities: Part PartType SectionPart Section Attributes: PartID PartTypeID PartName PartDescription PartType SectionName SectionPartID SectionID	Requested by the Administrator
	6.4 Delete Bicycle Part confirmation accepted details	The Administrator confirms the Delete Bicycle Part action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Bicycle Part and its related details get deleted from the system.	None	Entered by the Administrator
	6.4 [alt] Delete Bicycle Part confirmation declined	The Administrator declines or cancels the Delete Bicycle Part action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
6.5 Add Bicycle Section	6.5 Request to add new Bicycle Section details	The Administrator captures a new Bicycle Section on the database.	The Administrator requests to Add a Bicycle Section's information for a bicycle.	None	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	6.5 Input New Bicycle Section details	The Administrator enters the specific details of the Bicycle Section being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: Section Attributes: SectionID SectionName SectionDescription	Entered by the Administrator
	6.5 Add Confirmation	The Administrator confirms the Add Bicycle Section action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	6.5 [alt] Add Confirmation declined details	The Administrator declines or cancels the Add Bicycle Section action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
6.6 Search Bicycle Section	6.6 Request to Search Bicycle Section details	The Administrator wishes to view specific Bicycle Section details and prompts this action.	The Administrator requests to search for a specific Bicycle Section by triggering this process.	None	Requested by the Administrator
	6.6 Input Bicycle Section Search criteria details	The Administrator searches for a specific Bicycle Section in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Bicycle Section's details.	Entities: Section Attributes: SectionID SectionName SectionDescription	Entered by the Administrator
6.3 Update Bicycle Section	6.7 Request to Update Bicycle Section details	The Administrator requests to edit a Bicycle Section's details	The Administrator requests to edit Bicycle Section information to the required changes	None	Requested by the Administrator
	6.7 Input Updated Bicycle Section details	The Administrator enters the specific details that are to change for the Bicycle Section being updated	The Administrator inputs the changes to be made by filling in the required entries.	Entities: Section Attributes: SectionID SectionName SectionDescription	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	6.7 Update Bicycle Section confirmation accepted details	The Administrator confirms the Update Bicycle Section action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	6.7 [alt] Update Bicycle Section confirmation declined details	The Administrator declines or cancels the Update Bicycle Section action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
6.8 Delete Bicycle Section	6.8 Request to Delete Bicycle Section details	The Administrator desires to remove a certain Bicycle Section from the system	The Administrator selects the Bicycle Section to be deleted via the search function.	Entities: Section Attributes: SectionID SectionName SectionDescription	Requested by the Administrator
	6.8 Delete Bicycle Section confirmation accepted details	The Administrator confirms the Delete Bicycle Section action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Bicycle Section and its related details get deleted from the system.	None	Entered by the Administrator
	6.8 [alt] Delete Bicycle Section confirmation declined details	The Administrator declines or cancels the Delete Bicycle Section action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
7.1 Add Log Entry	7.1 Pass-through antenna range details	Student passes an antenna which allows the system to track their progress.	When a student passes an antenna with a tag.	Entities: Log Tag Antenna Attributes: LogID TagID AntennaID	Student passes an Antenna with a Tag



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				DateTime	
7.2 Search Log entry	7.2 Request to search Log entry details	Student requests to search for a Log entry on the system.	When a Student requests to find a specific log entry.	None	Requested by the Student
	7.2 Input Log entry search criteria details	Student enters the search criteria that provides them with the entry they are looking for.	When a Student knows what log entry, they are looking for and search for it.	Entities: Log Tag Antenna Attributes: LogID TagID AntennaID DateTime	Entered by the Student
8.1 Add Fault Task	8.1 Request to add new Fault Task details	The Administrator captures a new Fault Task on the database.	The Administrator requests to Add a Fault Task's information for a bicycle to be worked on by the mechanic.	None	Requested by the Administrator
	8.1 Input new fault Task details	The Administrator enters the specific details of the Fault Task being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: FaultTask Part Section Attributes: FaultTaskID Frequency PartsNeeded FaultTaskDescription PartName SectionName	Entered by the Administrator
	8.1 Add Confirmation accepted details	The Administrator confirms the Add Fault Task action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	8.1 [alt] Add Confirmation declined details	The Administrator declines or cancels the Add Fault Task action before it is committed to the database.	Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
8.2 Search Fault Task	8.2 Request to Search Fault task details	The Administrator wishes to view specific Fault Task details and prompts this action.	Administrator requests to search for a specific Fault Task by triggering this process.	None	Requested by the Administrator
	8.2 Input Fault Task Search criteria details	Administrator searches for a specific Fault Task in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Fault Task's details.	Entities: FaultTask Part Section Attributes: FaultTaskID Frequency PartsNeeded FaultTaskDescription PartName SectionName	Entered by the Administrator
8.3 Update Fault Task	8.3 Request to Update Fault Task details	The Administrator requests to edit a Fault Task's details	The Administrator requests to edit Fault Task information to the required changes	None	Requested by the Administrator
	8.3 Input Updated Fault Task details	The Administrator enters the specific details that are to change for the Fault Task being updated	The Administrator inputs the changes to be made by filling in the required entries.	Entities: FaultTask Part Section Attributes: FaultTaskID Frequency PartsNeeded FaultTaskDescription PartName SectionName	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	8.3 Update Fault Task confirmation accepted details	The Administrator confirms the Update Fault Task action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	8.3 [alt] Update Fault Task confirmation declined details	The Administrator declines or cancels the Update Fault Task action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
8.4 Delete Fault Task	8.4 Request to Delete Fault Task details	The Administrator desires to remove a certain Fault Task from the system	The Administrator selects the Fault Task to be deleted via the search function.	Entities: FaultTask Part Section Attributes: FaultTaskID Frequency PartsNeeded FaultTaskDescription PartName SectionName	Requested by the Administrator
	8.4 [alt] Delete Fault Task confirmation declined details	The Administrator declines or cancels the Delete Fault Task action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
	8.4 Delete Fault Task confirmation accepted details	The Administrator confirms the Delete Fault Task action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Fault Task and its related details get deleted from the system.	None	Entered by the Administrator
8.5 Add Maintenance Task	8.5 Request to Add new Maintenance Task details	The Administrator captures a new Maintenance Task on the database.	The Administrator requests to Add a Maintenance Task's information for a bicycle to be worked on by the mechanic.	None	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	8.5 New Maintenance Task details	The Administrator enters the specific details of the Maintenance Task being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: MaintenanceTask Part Section Attributes: MaintenanceTaskID Frequency MaintenanceTaskDescription PartName SectionName	Entered by the Administrator
	8.5 Add Maintenance Task confirmation accepted details	The Administrator confirms the Add Maintenance Task action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	8.5 [alt] Add Maintenance Task Confirmation declined details	The Administrator declines or cancels the Add Maintenance Task action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
8.6 Search Maintenance Task	8.6 Request to Search Maintenance Task details	The Administrator wishes to view specific Maintenance Task details and prompts this action.	The Administrator requests to search for a specific Maintenance Task by triggering this process.	None	Requested by the Administrator
	8.6 Input Maintenance Task Search criteria details	The Administrator searches for a specific Maintenance Task in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Maintenance Task's details.	Entities: MaintenanceTask Part Section Attributes: MaintenanceTaskID Frequency	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				MaintenanceTaskDescription PartName SectionName	
8.7 Update Maintenance Task	8.7 Request to Update Maintenance Task details	The Administrator requests to edit a Maintenance Task's details	The Administrator requests to edit Maintenance Task information to the required changes	None	Requested by the Administrator
	8.7 Updated Maintenance Task details	The Administrator enters the specific details that are to change for the Maintenance Task being updated	The Administrator inputs the changes to be made by filling in the required entries.	Entities: MaintenanceTask Part Section Attributes: MaintenanceTaskID Frequency MaintenanceTaskDescription PartName SectionName	Entered by the Administrator
	8.7 Update Maintenance Task confirmation accepted details	The Administrator confirms the Update Maintenance Task action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	8.7 [alt] Update Maintenance Task confirmation declined details	The Administrator declines or cancels the Update Maintenance Task action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
8.8 Delete Maintenance Task	8.8 Request to Delete Maintenance Task details	The Administrator desires to remove a certain Maintenance Task from the system	The Administrator selects the Maintenance Task to be deleted via the search function.	Entities: MaintenanceTask Part Section	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: MaintenanceTaskID Frequency MaintenanceTaskDescription PartName SectionName	
	8.8 Delete Maintenance Task confirmation accepted details	The Administrator confirms the Delete Maintenance Task action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Maintenance Task and its related details get deleted from the system.	None	Entered by the Administrator
	8.8 [alt] Delete Maintenance Task confirmation declined details	The Administrator declines or cancels the Delete Maintenance Task action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
9.1 Add School	9.1 Request to add new School details	The Administrator captures a new School on the database.	The Administrator requests to Add School information for students.	None	Requested by the Administrator
	9.1 New School details	The Administrator enters the specific details of the School being added.	The Administrator inputs the details to be created by filling in the required entries.	Entities: School Attributes: SchoolID VillageID SchoolName SchoolTelephone SchoolEmail	Entered by the Administrator
	9.1 Add School confirmation accepted details	The Administrator confirms the Add School action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
9.2 Search School	9.2 Request to search School details	The Administrator wishes to view specific School details and prompts this action.	The Administrator requests to search for a specific School by triggering this process.		
	9.2 Input School search criteria details	The Administrator searches for a specific School in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Schools' details.	Entities: School Attributes: VillageID SchoolName SchoolTelephone SchoolEmail	Entered by the Administrator
9.3 Update School	9.3 Request to update School details	The Administrator requests to edit a School's details	The Administrator requests to edit School information to the required changes	None	Requested by the Administrator
	9.3 Input Updated School details	The Administrator enters the specific details that are to change for the School being updated	The Administrator inputs the changes to be made by filling in the required entries.	Entities: School Attributes: SchoolID VillageID SchoolName SchoolTelephone SchoolEmail	Entered by the Administrator
	9.3 Update School details confirmation accepted details	The Administrator confirms the Update School action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	9.3 [alt] Decline Update School confirmation details	The Administrator declines or cancels the Update School action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
9.4 Delete School	9.4 Request to delete School details	The Administrator desires to remove a certain School from the system	The Administrator selects the School to be deleted via the search function.	Entities: School	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: SchoolID VillageID SchoolName SchoolTelephone SchoolEmail	
	9.4 Accept Delete School confirmation details	The Administrator confirms the Delete School action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the School and its related details get deleted from the system.	None	Entered by the Administrator
	9.4 [alt] Decline Delete School confirmation details	The Administrator declines or cancels the Delete School action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator
10.1 Report Fault	10.1 Request to report fault details	Student wishes to report a fault on the bicycle	When the Bicycle needs a Fault to be repaired by the mechanic.	None	Requested by a Student
	10.1 Select personal credential details	To be able to identify which student is reporting and which bicycle is being reported.	Chooses the BicycleID from the ones detected within proximity of the antenna where they're reporting the fault.	Entities: Bicycle Student Attributes: BicycleID StudentID	Requested by a Student
	10.1 Select Bicycle part details	This is for the specific bicycle section with the fault to be fixed to be selected. Then the related part in that section that needs fixing.	Student taps on the bicycle section on the image on-the screen to capture the section containing parts with possible faults that can be repaired. Then they can select the part in the drop-down list.	Entities: Bicycle Section Part Student Attributes: BicycleID SectionID	Requested by a Student



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				SectionName PartID PartName StudentID	
	10.1 Select task details	The student sheds light on what kind of task might need to be performed on the bicycle to fix the Fault.	Student selects the possible task to be performed on the bicycle if they know, if not, they choose "other".	Entities: FaultTask FaultType Attributes: FaultTaskID FaultTypeID FaultTaskDescription	
	10.1 Submit Fault confirmation accepted details	The student confirms the Report Fault action before that information captured is committed to the database.	Student confirms by selecting the related option on the screen.	None	Requested by a Student
	10.1 [alt] Submit Fault confirmation declined details	The student declines or cancels the Report Fault action before that information captured is committed to the database.	The student declines the Report Fault action by selecting the related option on the screen.	None	Requested by a Student
10.2 View Jobs	10.2 Request to view Job details	The mechanic needs to see Jobs to be done in that period at that school.	Mechanic needs this before carrying out any repair or maintenance job so that he can carry all necessary parts & tools.	None	Requested by the Mechanic
	10.2 Input search criteria details	Specify the period and school whose jobs need to be viewed.	The mechanic needs to view relevant jobs.	Entities: MechanicSchedule Attributes: MechanicScheduleID DateScheduled	Requested by the Mechanic
10.3 Check-in Bicycle	10.3 Request to check-in bicycle details	Student Checks-In their bicycle when it is due for a Mechanic Job.	Check-in information needed by the Mechanic for the jobs he is going to do and Administrator for record-keeping sake.	None	Requested by the Student



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	10.3 Input student details	Student captures their details and those for the bicycle for identification of the Job to be done.	Facilitates the search of the bicycle that needs to be worked on.	Entities: Student Bicycle MechanicSchedule Attributes: StudentID BicycleID DateScheduled	Requested by the Student
10.4 Report Maintenance	10.4 Completed maintenance task job details	Mechanic Reports a Maintenance Job completed, and the system automatically adds the next maintenance task on the system for that Bicycle.	Mechanic does this after a maintenance Job has been completed on a bicycle. The system sets the next Maintenance Job due based on this information.	Entities: Bicycle MechanicSchedule MechanicJobStatus ScheduledMaintenance MaintenanceTask Attributes: BicycleID MechanicStatusID MechanicJobStatusID CheckInDate Frequency	Requested by the Mechanic
11.1 Generate Maintenance Schedule	11.1 Initiate Report details	Time triggers this so that a Maintenance Schedule is generated for the Mechanic.	Maintenance Schedule is needed by the mechanic to know his Schedule for that period.	Entities: MechanicSchedule Attributes: MechanicScheduleID DateScheduled MechanicJobStatus	Requested by Time
11.2 Generate Maintenance Requirements Report	11.2 Initiate Report details	Time triggers this so that a Maintenance Requirements Report is generated for the Mechanic.	Maintenance Requirements are extracted for the Maintenance Schedule for that period to enable the Mechanic to bring all necessary equipment.	Entities: MechanicSchedule FaultTask	Requested by Time



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes MechanicScheduleID DateScheduled MechanicJobStatus PartsNeeded	
11.3 Generate Student Report	11.3 Request Generate Student Report details	Administrator generates this report for the tracking and assessment of the Student's academic progress and bicycle usage.	Used to determine if bicycle ownership will be handed over or not, as well as what areas need attention. This is done upon request.	Entities: Student StudentAttendance Attendance Studentmark MarkType Attributes: StudentID AttendanceID Date StudentAttendanceID Marktype Mark Date	Requested by the Administrator
	11.3 View report confirmation accepted details	The Administrator confirms the View Generated Student Report action before that information is displayed on the screen.	The Administrator confirms by selecting the related option on the screen to view the report.	None	Requested by the Administrator
	11.3 [alt] View Report confirmation declined details	The Administrator declines or cancels the View Report action before that information is displayed on the screen.	The Administrator declines to View the generated report by selecting the related option on the screen.	None	Requested by the Administrator
	11.3 Request to Export Report details	The Administrator selects to export and save the report generated in a different format.	Administrator stores this exported report for further assessment	None	
11.4 Generate Bicycle History Report	11.4 Request Generate	The Administrator generates this report for the tracking and	Used to determine the state of the bicycle and assess its history. This is done upon request.	Entities: Bicycle BicycleStatus	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	Bicycle History Report details	assessment of the Bicycle usage, previous users, and jobs done.		MechanicSchedule Student Attributes: BicycleID BicycleStatus CheckInDate StudentID	
	11.4 View report confirmation accepted details	The Administrator confirms the View Generated Bicycle History Report action before that information is displayed on the screen.	The Administrator confirms by selecting the related option on the screen to view the report.	None	Requested by the Administrator
	11.4 [alt] View Report confirmation declined details	The Administrator declines or cancels the View Report action before that information is displayed on the screen.	Administrator declines to View the generated report by selecting the related option on the screen.	None	Requested by the Administrator
	11.4 Request to Export Report details	The Administrator selects to export and save the report generated in a different format.	Administrator stores this exported report for further assessment	None	Requested by the Administrator
11.5 Generate Log Report	11.5 Generate Log Report request details	The Administrator generates this report for the tracking and assessment of Activity at certain Antennas along routes.	Used to determine the activity at an antenna. This is done upon request.	None	Requested by the Administrator
	11.5 Input Report criteria details	The Administrator enters the criteria for this Log report.	The Administrator is able to filter the log report to be generated per certain criteria, e.g., time period or per Antenna.	Entities: Antenna AntennaStatus Log Tag Attributes: AntennaID AntennaStatus DateInstalled Co-Ordinates	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				DateTime TagID	
	11.5 View report confirmation accepted details	The Administrator confirms the View Generated Log Report action before that information is displayed on the screen.	Administrator confirms by selecting the related option on the screen to view the report.	None	Requested by the Administrator
	11.5 [alt] View Report confirmation declined details	Administrator declines or cancels the View Report action before that information is displayed on the screen.	The Administrator declines to View the generated report by selecting the related option on the screen.	None	Requested by the Administrator
	11.5 Request to Export Report details	The Administrator selects to export and save the report generated in a different format.	Administrator stores this exported report for further assessment	None	Requested by the Administrator
11.6 Generate Parts Used Report	11.6 Initiate Report details	Time triggers this so that a Parts Used Report is generated for Admin purposes. Helps in inventory tracking.	Parts used in the Mechanic Job details are extracted for the Parts Used Report for that period Mechanic Schedule.	Entities: MechanicSchedule ReportFaultTask Attributes MechanicScheduleID FaultReportID MechanicJobStatus PartsUsed	Requested by Time
12.1 Send student did not arrive notification	12.1 Student has not arrived in time limit details	Notification sent to the student's guardian to inform them of the student's absence.	If no reason for absenteeism is given, the system will send a notification to the guardian to inform them of the student not coming to school that day.	Entities: Student Guardian Antenna Attributes: StudentID GuardianID GuardianPhone AntennaID	Requested by Time



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
12.2 Send student Maintenance Notification	12.2 Week before scheduled maintenance details	Notification is sent to the student to inform them of a maintenance job due on their bicycle.	The system sends a notification via SMS to the student a week before the maintenance date.	Entities: Student Bicycle Guardian ScheduledMaintenance MechanicSchedule Attributes: StudentID BicycleID GuardianPhone DateScheduled	Requested by Time
13.1 Backup Data	13.1 Request to Backup system details	Administrator uses the system to backup data onto the database for data security.	Administrator requests to backup system activities to the system database	None	Requested by the System Administrator
	13.1 Backup confirmation details	The Administrator provides details to confirm the backup procedure.	When the Administrator would want to make a backup.	None	Entered by the Administrator
13.2 Restore Data	13.2 Request to restore system details	The Administrator uses the system to restore data that was stored in a previous backup.	The Administrator requests to restore system activities that are stored in the database as a backup.	None	Requested by the Administrator
	13.2 Restore file confirmation details	The Administrator provides confirmation details for the backup procedure by selecting the backup to restore.	When the Administrator wants to restore a stored backup.	None	Entered by the Administrator
13.3 View Audit Trail Log	13.3 Request to View Audit Trail log details	The Administrator views the Audit Trail to see what activity has been taking place in the system.	The Administrator requests to view the Audit Trail Log over a certain period.	Entities: AuditTrail Attributes: AuditTrailID DateTime AuditTrailDescription	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
13.4 Search Audit Trail Entry	13.4 Request to search Audit Trail details	The Administrator requests to search the Audit Trail for activity.	The Administrator wants to search for an entry in the Audit Trail	None	Requested by the Administrator
	13.4 Input Audit Trail search details	The Administrator searches for a specific entry in the Audit Trail	It contains the criteria specified by the Administrator when requesting to view a specific audit trail entry.	Entities: AuditTrail Attributes: AuditTrailID DateTime AuditTrailDescription	Entered by the Administrator
13.5 Add Route	13.5 Request to add new Route details	The Administrator sets up a route trail marked by a sequence of Antennas	The Administrator requests to Add Route information for tracking.	None	Requested by the Administrator
	13.5 Input New Route details	The Administrator enters the specific details of the route being added by selecting the antenna positions in the order they occur.	The Administrator inputs the details to be created by filling in the required entries.	Entities: Route Attributes: RouteID SchoolID AntennaID Position	Entered by the Administrator
	13.5 Add Route confirmation accepted details	The Administrator confirms the Add Route action before it is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	13.5 [alt] Decline Add Route confirmation details	The Administrator declines or cancels the Add Route action before it is committed to the database.	The Administrator declines the Add action by selecting the related option on the screen.	None	Entered by the Administrator
13.6 Search Route	13.6 Input search Route criteria details	The Administrator searches for a specific Route sequence in the system.	It contains the criteria specified by the Administrator when requesting to view a specific Route trail.	Entities: Route RouteAntenna	Entered by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: RouteID AntennaID SchoolID Position	
13.7 Update Route	13.7 Request to update Route details	The Administrator requests to edit a route trail's details	The Administrator requests to edit Route information to the required changes	None	Requested by the Administrator
	13.7 Input Updated Route details	The Administrator enters the specific details that are to change for the route being updated	The Administrator inputs the changes to be made by filling in the required entries.	Entities: Route RouteAntenna Attributes: RouteID SchoolID AntennaID Position	Entered by the Administrator
	13.7 Update Route details confirmation accepted details	The Administrator confirms the Update Route action before that information captured is committed to the database.	The Administrator confirms by selecting the related option on the screen.	None	Entered by the Administrator
	13.7 [alt] Update Route confirmation declined details	The Administrator declines or cancels the Update Route action before that information captured is committed to the database.	The Administrator declines the Update action by selecting the related option on the screen.	None	Entered by the Administrator
13.8 Delete Route	13.8 Request to delete Route details	The Administrator desires to remove a certain route from the system	The Administrator selects the route to be deleted via the search function.	Route RouteAntenna Attributes: RouteID SchoolID AntennaID Position	Requested by the Administrator



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	13.8 Delete Route confirmation accepted details	The Administrator confirms the Delete Route action before that information captured is committed to the database. Then the action is done.	The Administrator confirms by selecting the related option on the screen, then the Route get deleted from the system.	None	Entered by the Administrator
	13.8 [alt] Delete Route confirmation declined details	The Administrator declines or cancels the Delete Route action before that removal is committed to the database.	The Administrator declines the Delete action by selecting the related option on the screen.	None	Entered by the Administrator



6.3 Conclusion

That table has now painted a clearer picture and explains the context diagram inputs or rather “in-flows” with further detail. This is important because the Data Flow Diagrams that follow the Context Diagram, build up from there, and those flows stay consistent and balanced at each level.



7. Reports and other outputs

7.1 Introduction

The Reports and other Outputs section of this deliverable contains a Table view. This Table details all the output flows from our system corresponding with the Context Diagram. It shows all the output interactions from the system and Reports generated. The basic actions that take place to trigger certain system output or display activity are also explicitly shown. Purposes for each action flow and the Usage is shown clearly per flow. Tied together to this, are the Entities and Attributes related to that particular flow, in the Entity Relational Diagram.

7.2 Reports and other outputs

Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
1.1 Login	1.1 Request Login details	System requests the User to provide their Login details.	When a User request to Login to the system.	Entities: None Attributes: None	Prompt displayed by system.
	1.1 [alt] Validation Error notification details	Notifies the Administrator that the Login details provided are invalid.	When a User provides invalid Login details.	Entities: None Attributes: None	Notification displayed by the system.
	1.1 Displayed User home details	System displays details based on the login details provided.	To display a User's home page.	Entities: User UserType Attributes: Username Password UserType UserPermissions	Details displayed by the system.
1.2 Logout	1.2 Request Logout Confirmation details	System prompts the User to confirm they want to Logout.	When the User requests to Logout.	Entities: None Attributes: None	Prompt displayed by the system.
	1.2 Displayed Logout confirmation details	Notifies the User they have been logged out.	When the User has been successfully Logged out.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
1.3 Forgot Password	1.3 Request User email address details	System requests User to provide their email address.	When the User requests to Reset their password.	Entities: None Attributes: None	Prompt Displayed by system.
	1.3 [alt] Invalid Email Address notification details	Notifies the User that the email provided is invalid.	When the User provides and invalid Email address.	Entities: None Attributes: None	Notification displayed by the system.
	1.3 Password Reset link sent notification details	Notifies the User that the Password Reset link has been sent to their email address.	When the system has sent the Reset Password link.	Entities: None Attributes: None	Notification displayed by the system.
	1.3 Request new Password details	System request the User to provide their new Password.	When the system requests the User to provide their new password.	Entities: None Attributes: None	Prompt displayed by the system.
	1.3 Request Reset Password confirmation details	System requests the User to confirm they wish to reset their password.	When the system request confirmation from the User that they wish to reset their password.	Entities: None Attributes: None	Prompt displayed by the system.
	1.3 [alt] Verification Error notification details	System notifies the User that their new password is invalid.	When the User provides invalid User details.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	1.3 Display reset password success notification	System notifies the User that their password has been reset successfully.	When the Users password has been successfully updated.	Entities: None Attributes: None	Notification displayed by the system.
1.4 Add User	1.4 Request User details	System requests the Administrator to enter the new Users details.	When the Administrator requests to add a new User.	Entities: None Attributes: None	Prompt displayed by the system.
	1.4 Request new User confirmation details	System prompts the Administrator to confirm they wish to add the new User.	When the system requests the Administrator to confirm they wish to add a new User.	Entities: None Attributes: None	Prompt displayed by the system.
	1.4 [alt] Invalid Format notification details	System notifies the Administrator that the User details provided are invalid.	When the Administrator provides invalid User details.	Entities: None Attributes: None	Notification displayed by the system.
	1.4 Display Add successful notification details	System notifies the Administrator that the new User was added successfully.	When the User has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
	1.4 [alt] User already exist notification details	System notifies the Administrator that the User already exists in the system.	When the User already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	1.4 [alt] Display add unsuccessful notification details	System notifies the Administrator that the addition was unsuccessful.	When the system incurs an error when adding a new User.	Entities: None Attributes: None	Notification displayed by the system.
1.5 Search User	1.5 Request User search criteria details	System requests the Administrator to provide the details of the User they wish to search for.	When the Administrator requests to search for a User.	Entities: None Attributes: None	Prompt displayed by system.
	1.5 [alt] Invalid Format notification details	System notifies the Administrator that the User details are invalid.	When the User details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	1.5 Display User details	System displays details of the User that the Administrator searched for.	When the system displays the User details.	Entities: User UserType Attributes: UserID UserType Username Password Firstname Surname Gender Age Email Telephone	Details displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	1.5 [alt] User does exist notification details	System notifies the Administrator that the User cannot be found.	When the User does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
1.6 Update User	1.6 Request Updated User details	System requests the Administrator to provide the updated details of the User.	When the Administrator requests to update a User's details.	Entities: None Attributes: None	Prompt displayed by system.
	1.6 Request update user confirmation details	System prompts the Administrator to confirm they wish to update the selected User.	When the system requires confirmation to update the User details.	Entities: None Attributes: None	Prompt displayed by system.
	1.6 [alt] Invalid format notification details	System notifies the Administrator that the new User details are invalid.	When the updated User details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	1.6 [alt] Save unsuccessful notification details	System notifies the Administrator that the Users details were not updated successfully.	When the system did not successfully update the Users details.	Entities: None Attributes: None	Notification displayed by the system.
	1.6 Display Successful notification details	System notifies the Administrator that the Users details were updated successfully.	When the system has successfully updated the Users details.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
1.7 Delete User	1.7 [alt] User does not exist notification details	Notifies the Administrator that the User they wish to delete does not exist.	When the Administrator requests to delete a User that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	1.7 Request Delete User confirmation details	System prompts the Administrator to confirm they wish to delete the selected User.	When the system requests confirmation from the Administrator to delete the User.	Entities: None Attributes: None	Prompt displayed by system.
	1.7 Display success notification details	System notifies the Administrator that the User was deleted successfully.	When the system has successfully deleted the User from the system.	Entities: None Attributes: None	Notification displayed by the system.
	1.7 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the User was not successfully deleted.	When the system incurs an error while deleting the User.	Entities: None Attributes: None	Notification displayed by the system.
1.8 Add User Type	1.8 Request User type details	System requests the Administrator to enter the new User Types details.	When the Administrator requests to add a new User Type.	Entities: None Attributes: None	Prompt displayed by system.
	1.8 Request new User type confirmation details	System prompts the Administrator to confirm they wish to add the new User Type.	When the system requests the Administrator to confirm they wish to add a new User Type.	Entities: None Attributes: None	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	1.8 [alt] Invalid Format Notification details	System notifies the Administrator that the User Type details provided are invalid.	When the Administrator provides invalid User Type details.	Entities: None Attributes: None	Notification displayed by the system.
	1.8 [alt] User Type already exists message	System notifies the Administrator that the User Type already exists in the system.	When the User Type already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	1.8 Display success notification details	System notifies the Administrator that the new User Type was added successfully.	When the User Type has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
	1.8 [alt] Display add unsuccessful notification details	System notifies the Administrator that the addition was unsuccessful.	When the system incurs an error when adding a new User Type.	Entities: None Attributes: None	Notification displayed by the system.
1.9 Search User Type	1.8 Request User Type search details	System requests the Administrator to provide the details of the User Type they wish to search for.	When the Administrator requests to search for a User Type.	Entities: None Attributes: None	Prompt displayed by system.
	1.9 [alt] Invalid format notification details	System notifies the Administrator that the User Type details are invalid.	When the User Type details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	1.9 Display User Type details	System displays details of the User Type that the Administrator searched for.	When the system displays the User Type details.	Entities: UserType Attributes: UserTypeID UserType UserTypePermissions	Details displayed by the system.
	1.9 [alt] User Type does not exist notification	System notifies the Administrator that the User Type cannot be found.	When the User Type does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
1.10 Update User Type	1.10 Request Update User Type details	System requests the Administrator to provide the updated details of the User Type.	When the Administrator requests to update a User Type details.	Entities: None Attributes: None	Prompt displayed by system.
	1.10 Request Update User type confirmation details	System prompts the Administrator to confirm they wish to update the selected User Type.	When the system requires confirmation to update the User Type details.	Entities: None Attributes: None	Prompt displayed by system.
	1.10 [alt] Save unsuccessful notification details	System notifies the Administrator that the User Type details were not updated successfully.	When the system did not successfully update the User Type details.	Entities: None Attributes: None	Notification displayed by the system.
	1.10 Display Success notification details	System notifies the Administrator that the User Type details were updated successfully.	When the system has successfully updated the User Type details.	Entities: None Attributes:	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				None	
1.11 Delete User Type	1.11 [alt] User Type does not exist notification	Notifies the Administrator that the User Type they wish to delete does not exist.	When the Administrator requests to delete a User that does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
	1.11 Request Delete User Type confirmation details	System prompts the Administrator to confirm they wish to delete the selected User Type.	When the system requests confirmation from the Administrator to delete the User Type.	Entities: None Attributes: None	Prompt displayed by system.
	1.11 Display Success notification details	System notifies the Administrator that the User Type was deleted successfully.	When the system has successfully deleted the User Type from the system.	Entities: None Attributes: None	Notification displayed by the system.
	1.11 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the User was not successfully deleted.	When the system incurs an error while deleting the User type.	Entities: None Attributes: None	Notification displayed by the system.
2.1 Complete Job	2.1 Request select Specific job details	System prompts the Mechanic to select the Job they wish to complete.	When the system requests the Mechanic to select a specific job.	Entities: None Attributes: None	Prompt displayed by system.
	2.1 Request if part required was used confirmation	System prompts the mechanic to confirm the parts used.	When the system requests the mechanic to confirm if he used the specified parts.	Entities: ReportFault Attributes: PartUsed	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
2.2 View Job Tasks	2.2 Displayed job tasks list successfully details	System displays the Job Task list details.	When the Job Task list is displayed.	Entities: ReportLine Attributes: PartfaultID SectionPartID FaultReportID ReportLineStatusID	Details Displayed by system.
	2.2 [alt] No entries found notification error details	System notifies the mechanic that the Job tasks cannot be found.	When the job task does not exist in the system.	Entities: None Attributes: None	Notification displayed by the system.
3.1 Add Antenna	3.1 Request Antenna details	System requests the Administrator to enter the new Antennas details.	When the Administrator requests to add a new Antenna.	Entities: None Attributes: None	Prompt displayed by system.
	3.1 Request Add Antenna confirmation details	System prompts the Administrator to confirm they wish to add the new Antenna.	When the system requests the Administrator to confirm they wish to add a new Antenna.	Entities: None Attributes: None	Prompt displayed by system.
	3.1 [alt] Invalid format notification details	System notifies the Administrator that the Antenna details provided are invalid.	When the Administrator provides invalid Antenna details.	Entities: None Attributes: None	Notification displayed by the system.
	3.1 [alt] Antenna already exist notification details	System notifies the Administrator that the Antenna already exists in the system.	When the Antenna already exists in the system.	Entities: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: None	
	3.1 Display success notification details	System notifies the Administrator that the new Antenna was added successfully.	When the Antenna has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
	3.1 [alt] Display add unsuccessful notification details	System notifies the Administrator that the addition was unsuccessful.	When the system incurs an error when adding a new antenna.	Entities: None Attributes: None	
3.2 Search Antenna	3.2 Request Antenna search criteria details	System requests the Administrator to provide the details of the Antenna they wish to search for.	When the Administrator requests to search for an Antenna.	Entities: None Attributes: None	Prompt displayed by system.
	3.2 [alt] Invalid format notification details	System notifies the Administrator that the Antenna details are invalid.	When the Antenna details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	3.2 Display Antenna details	System displays details of the Antenna that the Administrator searched for.	When the system displays the Antenna details.	Entities: Antenna Attributes: AntennaID AntennaStatusID Co-ordinates DateInstalled	Details displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	3.2 [alt] Antenna does not exist notification details	System notifies the Administrator that the Antenna cannot be found.	When the Antenna does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
3.3 Update Antenna	3.3 Request Updated Antenna details	System requests the Administrator to provide the updated details of the Antenna.	When the Administrator requests to update an Antennas details.	Entities: None Attributes: None	Prompt displayed by system.
	3.3 [alt] Invalid format notification details	System notifies the Administrator that the new Antenna details are invalid.	When the updated Antenna details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	3.3 [alt] Save Unsuccessful notification details	System notifies the Administrator that the Antennas details were not updated successfully.	When the system did not successfully update the Antennas details.	Entities: None Attributes: None	Notification displayed by the system.
	3.3 Display Success notification details	System notifies the Administrator that the Antennas details were updated successfully.	When the system has successfully updated the Antennas details.	Entities: None Attributes: None	Notification displayed by the system.
3.4 Delete Antenna	3.4 [alt] Antenna does not exist notification details	Notifies the Administrator that the Antenna they wish to delete does not exist.	When the Administrator requests to delete an Antenna that does not exist.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	3.4 Request Delete Antenna confirmation details	System prompts the Administrator to confirm they wish to delete the selected Antenna.	When the system requests confirmation from the Administrator to delete the Antenna.	Entities: None Attributes: None	Prompt displayed by system.
	3.4 Display success notification details	System notifies the Administrator that the Antenna was deleted successfully.	When the system has successfully deleted the Antenna from the system.	Entities: None Attributes: None	Notification displayed by the system.
	3.4 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the User was not successfully deleted.	When the system incurs an error while deleting the antenna.	Entities: None Attributes: None	Notification displayed by the system.
3.5 Add Tag	3.5 Request Tag details	System requests the Administrator to enter the new Tags details.	When the Administrator requests to add a new Tag.	Entities: None Attributes: None	Prompt displayed by system.
	3.5 Request new Tag confirmation details	System prompts the Administrator to confirm they wish to add the new Tag.	When the system requests the Administrator to confirm they wish to add a new Tag.	Entities: None Attributes: None	Prompt displayed by system.
	3.5 [alt] Invalid Format notification details	System notifies the Administrator that the Tag details provided are invalid.	When the Administrator provides invalid Tag details.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	3.5 [alt] Tag already exist message details	System notifies the Administrator that the Tag already exists in the system.	When the Tag already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	3.5 Display Success Notification details	System notifies the Administrator that the new Tag was added successfully.	When the Tag has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
	3.5 [alt] Display add unsuccessful notification details	System notifies the Administrator that the addition was unsuccessful.	When the system incurs an error when adding a new tag.	Entities: None Attributes: None	Notification displayed by the system.
3.6 Search Tag	3.6 Request Tag search criteria details	System requests the Administrator to provide the details of the Tag they wish to search for.	When the Administrator requests to search for a Tag.	Entities: None Attributes: None	Prompt displayed by system.
	3.6 [alt] Invalid Format notification details	System notifies the Administrator that the Tag details are invalid.	When the Tag details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	3.6 Display Tag Details	System displays details of the Tag that the Administrator searched for.	When the system displays the Tag details.	Entities: Tag Attributes: TagID TagStatusID	Details displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				DateInstalled	
	3.6 [alt] Tag does not exist notification details	System notifies the Administrator that the Tag cannot be found.	When the Tag does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
3.7 Update Tag	3.7 Request Updated Tag details	System requests the Administrator to provide the updated details of the Tag.	When the Administrator requests to update a Tags details.	Entities: None Attributes: None	Prompt displayed by system.
	3.7 Request Update Tag confirmation details	System prompts the Administrator to confirm they wish to update the selected Tag.	When the system requires confirmation to update the Tag details.	Entities: None Attributes: None	Prompt displayed by system.
	3.7 [alt] Invalid format notification details	System notifies the Administrator that the new Tag details are invalid.	When the updated Tag details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	3.7 [alt] Save unsuccessful notification details	System notifies the Administrator that the Tag details were not updated successfully.	When the system did not successfully update the Tag details.	Entities: None Attributes: None	Notification displayed by the system.
3.8 Delete Tag	3.8 [alt] Tag does not exist notification details	Notifies the Administrator that the Tag they wish to delete does not exist.	When the Administrator requests to delete a Tag that does not exist.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	3.8 Request Delete Tag confirmation details	System prompts the Administrator to confirm they wish to delete the selected Tag.	When the system requests confirmation from the Administrator to delete the Tag.	Entities: None Attributes: None	Prompt displayed by system.
	3.8 Display Success notification details	System notifies the Administrator that the Tag was deleted successfully.	When the system has successfully deleted the Tag from the system.	Entities: None Attributes: None	Notification displayed by the system.
	3.8 [alt] Tag does not exist notification details	Notifies the Administrator that the Tag they wish to delete does not exist.	When the Administrator requests to delete a Tag that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	3.8 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the tag was not successfully deleted.	When the system incurs an error while deleting the tag.	Entities: None Attributes: None	Notification displayed by the system.
4.1 Add Student	4.1 Request Student details	System requests the Administrator to enter the new Students details.	When the Administrator requests to add a new Student.	Entities: None Attributes: None	Prompt displayed by system.
	4.1 Request new Student confirmation details	System prompts the Administrator to confirm they wish to add the new Student.	When the system requests the Administrator to confirm they wish to add a new Student.	Entities: None Attributes: None	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	4.1 [alt] Invalid Format notification details	System notifies the Administrator that the Student details provided are invalid.	When the Administrator provides invalid Student details.	Entities: None Attributes: None	Notification displayed by the system.
	4.1 [alt] Student already exist message details	System notifies the Administrator that the Student already exists in the system.	When the Student already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	4.1 Display Success Notification details	System notifies the Administrator that the new Student was added successfully.	When the Student has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
	4.1 [alt] Display add unsuccessful notification details	System notifies the Administrator that the addition was unsuccessful.	When the system incurs an error when adding a new student.	Entities: None Attributes: None	Notification displayed by the system.
4.2 Search Student	1.11 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the User was not successfully deleted.	When the system incurs an error while deleting the User.	Entities: None Attributes: None	Notification displayed by the system.
	4.2 [alt] Invalid Format notification details	System notifies the Administrator that the Student details are invalid.	When the Student details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	4.2 Display Student Details	System displays details of the Student that the Administrator searched for.	When the system displays the Student details.	Entities: Student Bicycle Attributes: StudentID BicycleID SchoolName VillageName StudentName StudentSurname StudentDOB StudentGender	Details displayed by the system.
	4.2 [alt] Student does not exist notification details	System notifies the Administrator that the Student cannot be found.	When the Student does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
4.3 Update Student	4.3 Request Update Student details	System requests the Administrator to provide the updated details of the Student.	When the Administrator requests to update a Student details.	Entities: None Attributes: None	Prompt displayed by system.
	4.3 Request Update Student confirmation details	System notifies the Administrator that the new Student details are invalid.	When the updated Student details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	4.3 [alt] Invalid format notification details	System notifies the Administrator that the Student details were not updated successfully.	When the system did not successfully update the Student details.	Entities: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: None	
	4.3 Display Success notification details	System notifies the Administrator that the Student details were updated successfully.	When the system has successfully updated the Student details.	Entities: None Attributes: None	Notification displayed by the system.
4.4 Delete Student	4.4 [alt] Student does not exist notification details	Notifies the Administrator that the Student they wish to delete does not exist.	When the Administrator requests to delete an Student that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	4.4 Request Delete Student confirmation details	System prompts the Administrator to confirm they wish to delete the selected Student.	When the system requests confirmation from the Administrator to delete the Student.	Entities: None Attributes: None	Prompt displayed by system.
	4.4 Display Success notification details	System notifies the Administrator that the Student was deleted successfully.	When the system has successfully deleted the Student from the system.	Entities: None Attributes: None	Notification displayed by the system.
	4.4 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the User was not successfully deleted.	When the system incurs an error while deleting the student.	Entities: None Attributes: None	Notification displayed by the system.
4.5 Capture Student Marks	4.5 Request Student Mark details	System requests the Administrator to enter the new Student Mark details.	When the Administrator requests to add new Student Marks.	Entities: None Attributes:	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				None	
	4.5 Request new Student Mark confirmation details	System prompts the Administrator to confirm they wish to add the new Student Marks.	When the system requests the Administrator to confirm they wish to add new Student Marks.	Entities: None Attributes: None	Prompt displayed by system.
	4.5 [alt] Invalid Format notification details	System notifies the Administrator that the Student Mark details provided are invalid.	When the Administrator provides invalid Tag details.	Entities: None Attributes: None	Notification displayed by the system.
	4.5 [alt] Save Unsuccessful notification details	System notifies the Administrator that the Student Marks were not saved correctly.	When the system was unsuccessful in saving the Student Marks.	Entities: None Attributes: None	Notification displayed by the system.
	4.5 Display Success Notification details	System notifies the Administrator that the new Student Mark has been added successfully.	When the Student Mark has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
4.6 Search Student Marks	4.6 Request Student Mark search criteria details	System requests the Administrator to provide the details of the Student Mark they wish to search for.	When the Administrator requests to search for a Student Mark.	Entities: None Attributes: None	Prompt displayed by system.
	4.6 [alt] Invalid Format notification details	System notifies the Administrator that the Student Mark details are invalid.	When the Student Mark details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	4.6 Display Student Mark details	System displays details of the Student Mark that the Administrator searched for.	When the system displays the Student Mark details.	Entities: StudentMark Attributes: StudentMark Date	Details displayed by the system.
	4.6 [alt] Student Mark does not exist notification	System notifies the Administrator that the Student Mark cannot be found.	When the Student Mark does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
4.7 Add Mark Type	4.7 Request Mark Type details	System requests the Administrator to enter the new Mark Type details.	When the Administrator requests to add a new Mark Type.	Entities: None Attributes: None	Prompt displayed by the system.
	4.7 Request Add Mark Type confirmation details	System prompts the Administrator to confirm they wish to add the new Mark Type.	When the system requests the Administrator to confirm they wish to add a new Mark Type.	Entities: None Attributes: None	Prompt displayed by system.
	4.7 [alt] Invalid format notification details	System notifies the Administrator that the Mark Type details provided are invalid.	When the Administrator provides invalid Mark Type details.	Entities: None Attributes: None	Notification displayed by the system.
	4.7 [alt] Mark type already exist notification details	System notifies the Administrator that the Mark Type already exists in the system.	When the Mark Type already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	4.7 Display success notification details	System notifies the Administrator that the new Mark Type was added successfully.	When the Mark Type has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
	4.7 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the User was not successfully deleted.	When the system incurs an error while deleting the Mark Type.	Entities: None Attributes: None	Notification displayed by the system.
4.8 Search Mark Type	4.8 Request Mark Type search criteria details	System requests the Administrator to provide the details of the Mark Type they wish to search for.	When the Administrator requests to search for a Mark Type.	Entities: None Attributes: None	Prompt displayed by system.
	4.8 [alt] Invalid format notification details	System notifies the Administrator that the Mark Type details are invalid.	When the Mark Type details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	4.8 Display Mark Type details	System displays details of the Mark Type that the Administrator searched for.	When the system displays the Mark Type details.	Entities: MarkType Attributes: MarkTypeID MarkType	Details displayed by the system.
	4.8 [alt] Mark Type does not exist notification details	System notifies the Administrator that the Mark Type cannot be found.	When the Mark Type does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
4.9 Update Mark Type	4.9 Request Updated Mark Type details	System requests the Administrator to provide the updated details of the Mark Type.	When the Administrator requests to update a Mark Type details.	Entities: None Attributes: None	Prompt displayed by system.
	4.9 [alt] Invalid format notification details	System notifies the Administrator that the new Mark Type details are invalid.	When the updated Mark Type details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	4.9 [alt] Save Unsuccessful notification details	System notifies the Administrator that the Mark Type details were not updated successfully.	When the system did not successfully update the Mark Type details.	Entities: None Attributes: None	Notification displayed by the system.
	4.9 Display Success notification details	System notifies the Administrator that the Mark Type details were updated successfully.	When the system has successfully updated the Mark Type details.	Entities: None Attributes: None	Notification displayed by the system.
4.10 Delete Mark Type	4.10 [alt] Antenna does not exist notification details	Notifies the Administrator that the Delete Mark Type they wish to delete does not exist.	When the Administrator requests to delete a Mark Type that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	4.10 Request Delete Mark Type confirmation details	System prompts the Administrator to confirm they wish to delete the selected D Mark Type.	When the system requests confirmation from the Administrator to delete the Mark Type.	Entities: None Attributes: None	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	4.10 Display success notification details	System notifies the Administrator that the Mark Type was deleted successfully.	When the system has successfully deleted the Mark Type from the system.	Entities: None Attributes: None	Notification displayed by the system.
	4.10 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the mark type was not successfully deleted.	When the system incurs an error while deleting the mark type.	Entities: None Attributes: None	Notification displayed by the system.
5.1 Add Bicycle	5.1 Request Bicycle details	System requests the Administrator to enter the new Bicycle details.	When the Administrator requests to add a new Bicycle.	Entities: None Attributes: None	Prompt displayed by system.
	5.1 Request Add Bicycle confirmation details	System prompts the Administrator to confirm they wish to add the new Bicycle.	When the system requests the Administrator to confirm they wish to add a new Bicycle.	Entities: None Attributes: None	Prompt displayed by system.
	5.1 [alt] Invalid Format notification details	System notifies the Administrator that the Bicycle details provided are invalid.	When the Administrator provides invalid Bicycle details.	Entities: None Attributes: None	Notification displayed by the system.
	5.1 [alt] Bicycle already exist notification details	System notifies the Administrator that the Bicycle already exists in the system.	When the Bicycle already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	5.1 Display Success Notification details	System notifies the Administrator that the new Bicycle was added successfully.	When the Bicycle has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
	5.1 [alt] Display add unsuccessful notification details	System notifies the Administrator that the addition was unsuccessful.	When the system incurs an error when adding a new bicycle.	Entities: None Attributes: None	Notification displayed by the system.
5.2 Search Bicycle	5.2 Request Bicycle Search criteria details	System requests the Administrator to provide the details of the Bicycle they wish to search for.	When the Administrator requests to search for a Bicycle.	Entities: None Attributes: None	Prompt displayed by system.
	5.2 [alt] Invalid Format notification details	System notifies the Administrator that the Bicycle details are invalid.	When the Bicycle details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	5.2 Display Bicycle Details	System displays details of the Bicycle that the Administrator searched for.	When the system displays the Bicycle details.	Entities: Bicycle Attributes: BicycleID BicycleStatus Student BicycleMake DateAdded Tag	Details displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	5.2 [alt] Bicycle does not exist notification details	System notifies the Administrator that the Bicycle cannot be found.	When the Bicycle does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
5.3 Update Bicycle	5.3 Request Update Bicycle details	System requests the Administrator to provide the updated details of the Bicycle.	When the Administrator requests to update a Bicycle details.	Entities: None Attributes: None	Prompt displayed by system.
	5.3 Request Update Bicycle confirmation details	System prompts the Administrator to confirm they wish to add the new Bicycle.	When the system requires confirmation to update the Bicycle details.	Entities: None Attributes: None	Notification displayed by the system.
	5.3 [alt] Invalid Format notification details	System notifies the Administrator that the new Bicycle details are invalid.	When the updated Bicycle details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	5.3 [alt] Save unsuccessful notification details	System notifies the Administrator that the Bicycle details were not updated successfully.	When the system did not successfully update the Bicycle details.	Entities: None Attributes: None	Notification displayed by the system.
	5.3 Display Success notification details	System notifies the Administrator that the Bicycle details were updated successfully.	When the system has successfully updated the Bicycle details.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
5.4 Delete Bicycle	5.4 [alt] Bicycle does not exist notification details	Notifies the Administrator that the Bicycle they wish to delete does not exist.	When the Administrator requests to delete a Bicycle that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	5.4 Request Delete Bicycle confirmation details	System prompts the Administrator to confirm they wish to delete the selected Bicycle.	When the system requests confirmation from the Administrator to delete the Bicycle.	Entities: None Attributes: None	Prompt displayed by system.
	5.4 Display Success notification details	System notifies the Administrator that the Bicycle was deleted successfully.	When the system has successfully deleted the Bicycle from the system.	Entities: None Attributes: None	Notification displayed by the system.
	5.4 [alt] Display delete unsuccessful notification details	System notifies the Administrator that the Bicycle was not successfully deleted.	When the system incurs an error while deleting the bicycle.	Entities: None Attributes: None	Notification displayed by the system.
5.5 Assign a Bicycle to Student	5.5 Display Success notification details	System notifies the Administrator that the Bicycle was Assigned to the student successfully	When the system has successfully assigned the Bicycle to the Student.	Entities: None Attributes: None	Notification displayed by the system.
	5.5 [alt] Display Assign unsuccessful notification details	System notifies the Administrator that the Bicycle was not Assigned to the Student successfully.	When the system did not successfully Assign the Bicycle to the Student.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
5.6 Unassign Bicycle to Student	5.6 Display Success notification details	System notifies the Administrator that the Bicycle was Unassigned from the student successfully	When the system has successfully Unassigned Bicycle from the Student.	Entities: None Attributes: None	Notification displayed by the system.
	5.6 [alt] Display Unassign unsuccessful notification details	System notifies the Administrator that the Bicycle was not Unassigned from the Student successfully.	When the system did not successfully Unassign the Bicycle from the Student.	Entities: None Attributes: None	Notification displayed by the system.
5.7 Assign Tag to bicycle	5.7 Display Success notification details	System notifies the Administrator that the Tag was Assigned to the bicycle successfully	When the system has successfully assigned the Tag to the bicycle.	Entities: None Attributes: None	Notification displayed by the system.
	5.7 [alt] Display Assign unsuccessful notification details	System notifies the Administrator that the Tag was not Assigned to the Bicycle successfully.	When the system did not successfully Assign the Tag to the Bicycle.	Entities: None Attributes: None	Notification displayed by the system.
5.8 Unassign Tag from Bicycle	5.8 Display Success notification details	System notifies the Administrator that the Tag was Unassigned from the Bicycle successfully	When the system has successfully Unassigned Tag from the Bicycle.	Entities: None Attributes: None	Notification displayed by the system.
	5.8 [alt] Display Unassign unsuccessful notification details	System notifies the Administrator that the Tag was not Unassigned from the Bicycle successfully.	When the system did not successfully Unassign the Tag from the Bicycle.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
6.1 Add Bicycle Part	6.1 Request Bicycle Part details	System requests the Administrator to enter the new Bicycle Part details.	When the Administrator requests to add a new Bicycle Part.	Entities: None Attributes: None	Prompt displayed by system.
	6.1 Request Add Bicycle Part confirmation details	System prompts the Administrator to confirm they wish to add the new Bicycle Part.	When the system requests the Administrator to confirm they wish to add a new Bicycle Part.	Entities: None Attributes: None	Prompt displayed by system.
	6.1 [alt] Invalid format notification details	System notifies the Administrator that the Bicycle Part details provided are invalid.	When the Administrator provides invalid Bicycle Part details.	Entities: None Attributes: None	Notification displayed by the system.
	6.1 [alt] Bicycle Part already exists notification details	System notifies the Administrator that the Bicycle Part already exists in the system.	When the Bicycle Part already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	6.1 Display Success Notification details	System notifies the Administrator that the new Bicycle Part was added successfully.	When the Bicycle Part has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
6.2 Search Bicycle Part	6.2 Request Bicycle Part search criteria details	System requests the Administrator to provide the details of the Bicycle Part they wish to search for.	When the Administrator requests to search for a Bicycle Part.	Entities: None Attributes: None	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	6.2 [alt] Invalid Format notification details	System notifies the Administrator that the Bicycle Part details are invalid.	When the Bicycle Part details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	6.2 Display Bicycle Part details	System displays details of the Bicycle Part that the Administrator searched for.	When the system displays the Bicycle Part details.	Entities: Part Attributes: PartID PartName PartDescription Quantity	Details displayed by system.
	6.2 [alt] Bicycle Part does not exist notification details	System notifies the Administrator that the Bicycle Part cannot be found.	When the Bicycle Part does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
6.3 Update Bicycle Part	6.3 Request Updated Bicycle Part details	System requests the Administrator to provide the updated details of the Bicycle Part.	When the Administrator requests to update a Bicycle Part details.	Entities: None Attributes: None	Prompt displayed by system.
	6.3 Request Update Bicycle Part confirmation details	System prompts the Administrator to confirm they wish to update the selected Bicycle Part.	When the system requires confirmation to update the Bicycle Part details.	Entities: None Attributes: None	Prompt displayed by system.
	6.3 [alt] Invalid format notification details	System notifies the Administrator that the new Bicycle Part details are invalid.	When the updated Bicycle Part details are invalid.	Entities: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: None	
	6.3 [alt] Update Bicycle Part confirmation declined	System notifies the Administrator that the Bicycle Part details were not updated successfully.	When the system did not successfully update the Bicycle Part details.	Entities: None Attributes: None	Notification displayed by the system.
6.4 Delete Bicycle Part	6.4 [alt] Bicycle Part does not exist notification details	Notifies the Administrator that the Student they wish to delete does not exist.	When the Administrator requests to delete a Bicycle Part that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	6.4 Request Delete Bicycle Part confirmation details	System prompts the Administrator to confirm they wish to delete the selected Bicycle Part.	When the system requests confirmation from the Administrator to delete the Bicycle Part.	Entities: None Attributes: None	Prompt displayed by system.
	6.4 Display Success notification details	System notifies the Administrator that the Bicycle Part was deleted successfully.	When the system has successfully deleted the Bicycle Part from the system.	Entities: None Attributes: None	Notification displayed by the system.
6.5 Add Bicycle Section	6.5 Request Bicycle Section details	System requests the Administrator to enter the new Bicycle Section details.	When the Administrator requests to add a new Bicycle Section.	Entities: None Attributes: None	Prompt displayed by system.
	6.5 Request Add Bicycle Section confirmation details	System prompts the Administrator to confirm they wish to add the new Bicycle Section.	When the system requests the Administrator to confirm they wish to add a new Bicycle Section.	Entities: None Attributes:	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				None	
	6.5 [alt] Invalid Format notification details	System notifies the Administrator that the Bicycle Section details provided are invalid.	When the Administrator provides invalid Bicycle Section details.	Entities: None Attributes: None	Notification displayed by the system.
	6.5 [alt] Bicycle already exist notification details	System notifies the Administrator that the Bicycle Section already exists in the system.	When the Bicycle Section already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	6.5 Display Success Notification details	System notifies the Administrator that the new Bicycle Section was added successfully.	When the Bicycle Section has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
6.6 Search Bicycle Section	6.6 Request Bicycle Section Search criteria details	System requests the Administrator to provide the details of the Bicycle Section they wish to search for.	When the Administrator requests to search for a Bicycle Section.	Entities: None Attributes: None	Prompt displayed by system.
	6.6 [alt] Invalid Format notification details	System notifies the Administrator that the Bicycle Section details are invalid.	When the Bicycle Section details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	6.6 Display Bicycle Section Details	System displays details of the Bicycle Section that the Administrator searched for.	When the system displays the Bicycle Section details.	Entities: Section Attributes: SectionID	Details displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				SectionNam SectionDescription	
	6.6 [alt] Bicycle does not exist notification details	System notifies the Administrator that the Bicycle Section cannot be found.	When the Bicycle Section does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
6.7 Update Bicycle Section	6.7 Request Update Bicycle details	System requests the Administrator to provide the updated details of the Bicycle.	When the Administrator requests to update a Bicycle details.	Entities: None Attributes: None	Prompt displayed by system.
	6.7 Request Update Bicycle Section confirmation details	System prompts the Administrator to confirm they wish to delete the selected Bicycle Section.	When the system requests confirmation from the Administrator to delete the Bicycle Section.	Entities: None Attributes: None	Prompts displayed by the system.
	6.7 [alt] Invalid Format notification details	System notifies the Administrator that the new Bicycle Section details are invalid.	When the updated Bicycle Section details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	6.7 Display Success notification details	System notifies the Administrator that the Bicycle Section details were updated successfully.	When the system has successfully updated the Bicycle Section details.	Entities: None Attributes: None	Notification displayed by the system.
6.8 Delete Bicycle Section	6.8 [alt] Bicycle Section does not exist notification details	Notifies the Administrator that the Bicycle Section they wish to delete does not exist.	When the Administrator requests to delete a Bicycle Section that does not exist.	Entities: None Attributes:	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				None	
	6.8 Request Delete Bicycle Section confirmation details	System prompts the Administrator to confirm they wish to delete the selected Bicycle Section.	When the system requests confirmation from the Administrator to delete the Bicycle Section.	Entities: None Attributes: None	Prompt displayed by system.
	6.8 Display Success notification details	System notifies the Administrator that the Bicycle Section was deleted successfully.	When the system has successfully deleted the Bicycle Section from the system.	Entities: None Attributes: None	Notification displayed by the system.
7.2 Search Log entry	7.2 Request Log entry search criteria details	System requests the Administrator to provide the details of the Log Entry they wish to search for.	When the Administrator requests to search for a Log Entry.	Entities: None Attributes: None	Prompt displayed by system.
	7.2 [alt] Invalid Format notification details	System notifies the Administrator that the Log Entry details are invalid.	When the Log Entry details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	7.2 Display Log entry Details	System displays details of the Log Entry that the Administrator searched for.	When the system displays the Log Entry details.	Entities: Log Antenna Tag Attributes: LogID AntennaID TagID DateTime	Details displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	7.2 [alt] Log entry does not exist notification details	System notifies the Administrator that the Log Entry cannot be found.	When the Log Entry does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
8.1 Add Fault Task	8.1 Request Fault Task details	System requests the Administrator to enter the new Fault Task details.	When the Administrator requests to add a new Fault Task.	Entities: None Attributes: None	Prompt displayed by system.
	8.1 Request Add Fault Task confirmation details	System prompts the Administrator to confirm they wish to add the new Fault Task.	When the system requests the Administrator to confirm they wish to add a new Fault Task.	Entities: None Attributes: None	Prompt displayed by system.
	8.1 [alt] Invalid Format Request details	System notifies the Administrator that the Fault Task details provided are invalid.	When the Administrator provides invalid Fault Task details.	Entities: None Attributes: None	Notification displayed by the system.
	8.1 [alt] Fault Task already exist notification details	System notifies the Administrator that the Fault Task already exists in the system.	When the Fault Task already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	8.1 Display Success Notification details	System notifies the Administrator that the new Fault Task was added successfully.	When the Fault Task has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
8.2 Search Fault Task	8.2 Request Fault Task search criteria details	System requests the Administrator to provide the details of the Fault Task they wish to search for.	When the Administrator requests to search for a Fault Task.	Entities: None Attributes: None	Prompt displayed by system.
	8.2 [alt] Invalid Format notification details	System notifies the Administrator that the Fault Task details are invalid.	When the Fault Task details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	8.2 Display Fault Task details	System displays details of the Fault Task that the Administrator searched for.	When the system displays the Fault Task details.	Entities: FaultTask Attributes: FaultTaskID PartsNeeded FaultTaskDescription	Details displayed by the system.
	8.2 [alt] Fault Task does not exist notification details	System notifies the Administrator that the Fault Task cannot be found.	When the Fault Task does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
8.3 Update Fault Task	8.3 Request Updated Fault Task details	System requests the Administrator to provide the updated details of the Fault Task.	When the Administrator requests to update a Fault Task details.	Entities: None Attributes: None	Prompt displayed by system.
	8.3 Request Update Fault Task confirmation details	System prompts the Administrator to confirm they wish to update the selected Fault Task.	When the system requires confirmation to update the Fault Task details.	Entities: None Attributes:	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				None	
	8.3 [alt] Invalid format notification details	System notifies the Administrator that the new Fault Task details are invalid.	. When the updated Fault Task details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	8.3 [alt] Save unsuccessful notification details	System notifies the Administrator that the Fault Task details were not updated successfully.	When the system did not successfully update the Fault Task details.	Entities: None Attributes: None	Notification displayed by the system.
	8.3 Display success notification details	System notifies the Administrator that the Fault Task details were updated successfully	When the system has successfully updated the Fault Task details.	Entities: None Attributes: None	Notification displayed by the system.
8.4 Delete Fault Task	8.4 [alt] Fault Task does not exist notification details	Notifies the Administrator that the Fault Task they wish to delete does not exist.	When the Administrator requests to delete a Fault Task that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	8.4 Request Delete Fault Task confirmation details	System prompts the Administrator to confirm they wish to delete the selected Fault Task.	When the system requests confirmation from the Administrator to delete the Fault Task.	Entities: None Attributes: None	Prompt displayed by system.
	8.4 Display Success notification details	System notifies the Administrator that the Fault Task was deleted successfully.	When the system has successfully deleted the Fault Task from the system.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
8.5 Add Maintenance Task	8.5 Request Maintenance Task details	System requests the Administrator to enter the new Maintenance Task details.	When the Administrator requests to add a new Maintenance Task.	Entities: None Attributes: None	Prompt displayed by system.
	8.5 Request Add Maintenance Task confirmation details	System prompts the Administrator to confirm they wish to add the new Maintenance Task.	When the system requests the Administrator to confirm they wish to add a new Maintenance Task.	Entities: None Attributes: None	Prompt displayed by system.
	8.5 [alt] Invalid Format notification details	System notifies the Administrator that the Maintenance Task details provided are invalid.	When the Administrator provides invalid Maintenance Task details.	Entities: None Attributes: None	Notification displayed by the system.
	8.5 [alt] Maintenance Task already exist notification details	System notifies the Administrator that the Maintenance Task already exists in the system.	When the Maintenance Task already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	8.5 Display Success Notification details	System notifies the Administrator that the new Maintenance Task was added successfully.	When the Maintenance Task has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
8.6 Search Maintenance Task	8.6 Request Maintenance Task search criteria details	System requests the Administrator to provide the details of the Maintenance Task they wish to search for.	When the Administrator requests to search for a Maintenance Task.	Entities: None Attributes: None	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	8.6 [alt] Invalid Format notification details	System notifies the Administrator that the Maintenance Task details are invalid.	When the Maintenance Task details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	8.6 Display Maintenance Task details	System displays details of the Fa Maintenance ult Task that the Administrator searched for.	When the system displays the Maintenance Task details.	Entities: MaintenanceTask Attributes: Description Frequency PartsNeeded	Details Displayed by system.
	8.6 [alt] Maintenance Task does not exist notification details	System notifies the Administrator that the Maintenance Task cannot be found.	When the Maintenance Task does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
8.7 Update Maintenance Task	8.7 Request Updated Maintenance Task details	System requests the Administrator to provide the updated details of the Maintenance Task.	When the Administrator requests to update a Maintenance Task details.	Entities: None Attributes: None	Prompt displayed by system.
	8.7 Request Update Maintenance Task confirmation details	System prompts the Administrator to confirm they wish to update the selected Maintenance Task.	When the system requires confirmation to update the Fault Maintenance details.	Entities: None Attributes: None	Prompt displayed by system.
	8.7 [alt] Invalid format notification details	System notifies the Administrator that the new Maintenance Task details are invalid.	. When the updated Maintenance Task details are invalid.	Entities: None Attributes:	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				None	
	8.7 [alt] Save Unsuccessful notification details	System notifies the Administrator that the Maintenance Task details were not updated successfully.	When the system did not successfully update the Maintenance Task details.	Entities: None Attributes: None	Notification displayed by the system.
	8.7 Display Success notification details	System notifies the Administrator that the Maintenance Task details were updated successfully	When the system has successfully updated the Maintenance Task details.	Entities: None Attributes: None	Notification displayed by the system.
8.8 Delete Maintenance Task	8.8 [alt] Maintenance Task does not exist notification details	Notifies the Administrator that the Maintenance Task they wish to delete does not exist.	When the Administrator requests to delete a Maintenance Task that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	8.8 Request Delete Maintenance Task confirmation details	System prompts the Administrator to confirm they wish to delete the selected Maintenance Task.	When the system requests confirmation from the Administrator to delete the Maintenance Task.	Entities: None Attributes: None	Prompt displayed by system.
	8.8 Display Success notification details	System notifies the Administrator that the Maintenance Task was deleted successfully.	When the system has successfully deleted the Maintenance Task from the system.	Entities: None Attributes: None	Notification displayed by the system.
9.1 Add School	9.1 Request School details	System requests the Administrator to enter the new School details.	When the Administrator requests to add a new School.	Entities: None Attributes: None	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	9.1 Request new School confirmation details	System prompts the Administrator to confirm they wish to add the new School.	When the system requests the Administrator to confirm they wish to add a new School.	Entities: None Attributes: None	Prompt displayed by system.
	9.1 [alt] Invalid Format notification details	System notifies the Administrator that the School details provided are invalid.	When the Administrator provides invalid School details.	Entities: None Attributes: None	Notification displayed by the system.
	9.1 [alt] School already exist message details	System notifies the Administrator that the School already exists in the system.	When the School already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	9.1 Display Success Notification details	System notifies the Administrator that the new School was added successfully.	When the School has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
9.2 Search School	9.2 Request School Search criteria details	System requests the Administrator to provide the details of School they wish to search for.	When the Administrator requests to search for a School.	Entities: None Attributes: None	Prompt displayed by system.
	9.2 [alt] Invalid Format notification details	System notifies the Administrator that the School details are invalid.	When the School details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	9.2 Display School Details	System displays details of the School that the Administrator searched for.	When the system displays the School details.	Entities: School Attributes: SchoolID SchoolName SchoolTelephone SchoolEmail	Details displayed by the system.
	9.2 [alt] School does not exist notification details	System notifies the Administrator that the School cannot be found.	When the School does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
9.3 Update School	9.3 Request to update School details	System requests the Administrator to provide the updated details of the School.	When the Administrator requests to update a School details.	Entities: None Attributes: None	Prompt displayed by system.
	9.3 Request update School confirmation details	System prompts the Administrator to confirm they wish to add the new School.	When the system requires confirmation to update the School details.	Entities: None Attributes: None	Prompt displayed by the system.
	9.3 [alt] Invalid format notification details	System notifies the Administrator that the new School details are invalid.	When the updated School details are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	9.3 [alt] Save unsuccessful notification details	System notifies the Administrator that the School details were not updated successfully.	When the system did not successfully update the School details.	Entities: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: None	
	9.3 Display Success notification details	System notifies the Administrator that the School details were updated successfully.	When the system has successfully updated the School details.	Entities: None Attributes: None	Notification displayed by the system.
9.4 Delete School	9.4 [alt] School does not exist notification details	Notifies the Administrator that the School they wish to delete does not exist.	When the Administrator requests to delete a School that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	9.4 Request Delete School confirmation details	System prompts the Administrator to confirm they wish to delete the selected School.	When the system requests confirmation from the Administrator to delete the School.	Entities: None Attributes: None	Prompt displayed by system.
	9.4 Display Success notification details	System notifies the Administrator that the School was deleted successfully.	When the system has successfully deleted the School from the system.	Entities: None Attributes: None	Notification displayed by the system.
10.1 Report Fault	10.1 Request Submit Fault confirmation details	System prompts the Student to confirm they wish to report the selected Fault.	When the system requests confirmation from the Student to report the fault.	Entities: None Attributes: None	Prompt displayed by system.
	10.1 Display Success notification details	System notifies the Student that their fault was reported successfully.	When the fault has been successfully reported.	Entities: None Attributes:	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				None	
10.2 View Jobs	10.2 [alt] Validation error Notification details	System notifies the Mechanic the details entered are invalid.	When the mechanic enters invalid details.	Entities: None Attributes: None	Notification displayed by the system.
	10.2 Displayed job list successfully details	The system displays the job list.	When the system displays the Job List details.	Entities: Bicycle Student Attributes: Date Notes	Details displayed by the system.
10.3 Check-in Bicycle	10.3 Request Student details	The system will prompt the Student to provide their Student details.	When the system requests the Student to provide their details.	Entities: None Attributes: None	Prompt displayed by the system.
	10.3 [alt] Provided details do not appear on the system notification details	The system notifies the student that their details are invalid and do not exist on the system.	When the student enters invalid details.	Entities: None Attributes: None	Notification displayed by the system.
11.1 Generate Maintenance Schedule	11.1 View Maintenance Schedule Report details	The system displays the Maintenance Schedule to the Mechanic.	When the system displays the report with its details.	Entities: School Student Bicycle BicyclePart Part	Report displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				MaintenanceTask Attributes: SchoolName StudentName StudentSurname BicycleID BicyclePartID PartID Description	
11.2 Generate Maintenance Requirements Report	11.2 View Maintenance Requirements Report details	The system will display the Maintenance Requirements Report.	When the system displays the report with its details.	Entities: School BicyclePart Part Attributes: SchoolName BicyclePartID PartDescription	Report displayed by the system.
11.3 Generate Student Report	11.3 View Student Report details	The system will display the Student Report details to the Administrator.	When the system displays the report with its details.	Entities: Student School Bicycle StudentAttendance StudentMark Attributes: StudentName StudentSurname StudentDoB	Report displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				StudentGender SchoolName AttendanceID Date Mark	
	11.3 [alt] Export error notification details	System notifies the Administrator that the Report Export failed.	When the Report is not exported successfully.	Entities: None Attributes: None	Notification displayed by the system.
11.4 Generate Bicycle History Report	11.4 Request to View report confirmation details	The system prompts the Administrator to confirm the wish to view the report.	When the system needs confirmation to display the report.	Entities: None Attributes: None	Prompt displayed by system.
	11.4 [alt] Export error notification details	System notifies the Administrator that the Report Export failed.	When the Report is not exported successfully.	Entities: None Attributes: None	Notification displayed by the system.
	11.4 View Bicycle History Report details	The system will display the Student Report details to the Administrator.	When the systems display the report with its details.	Entities: Student School Tag BicycleStatus Mechanic Job MechanicJobStatus JobTask Task FaultTask	Report displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				MaintenanceTask Attributes: StudentID SchoolName TagID BicycleStatus JobID TaskDescription FaultTaskID MaintenanceTaskID	
11.5 Generate Log Report	11.5 Request Report criteria details	The system prompts the Administrator to enter the criteria they wish to generate the report by.	When the system requests search criteria details from the Administrator.	Entities: None Attributes: None	Prompt displayed by system.
	11.5 Request to View report confirmation details	The system prompts the Administrator to confirm the wish to view the report.	When the system needs confirmation to display the report.	Entities: None Attributes: None	Prompt displayed by system.
	11.5 [alt] Export error notification details	System notifies the Administrator that the Report Export failed.	When the Report is not exported successfully.	Entities: None Attributes: None	Notification displayed by the system.
	11.5 View Log Report details	The system will display the Student Report details to the Administrator.	When the systems display the report with its details.	Entities: Antenna AntennaStatus Log Bicycle	Report displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Student Attributes: AntennaID Coordinates AntennaStatus LogID DateTime BicycleID StudentID	
11.6 Generate Parts Used Report	11.6 View Parts Used Report details	The system will display the Student Report details to the User.	When the systems display the report with its details.	Entities: BicyclePart BicycleSection Attributes: PartName PartDescription SectionDescription	Report displayed by the system.
12.1 Send student did not arrive notification	12.1 Send Student alert notification SMS details	The system sends an SMS to the guardian of the child alerting them that the student did not arrive at school on the day.	When the system alerts the Guardian.	Entities: Student Guardian Attributes: StudentID StudentName GuardianID GuardianName GuardianPhone	Alert by the system.
12.2 Send student Maintenance Notification	12.2 Send Maintenance	The system sends an SMS to the student alerting them that their bicycle is due for maintenance.	When the system alerts the Student.	Entities: Student	Alert by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
	notification SMS details			Attributes: StudentName PhoneNumber	
13.1 Backup Data	13.1 Request Backup confirmation details	System prompts the Administrator to confirm they wish to Back up the system.	When the system requires confirmation to Back up the system.	Entities: None Attributes: None	Prompt displayed by the system.
	13.1 Display success notification details	System notifies the Administrator that the System was Backed up successfully.	When the system has been successfully Backed up.	Entities: None Attributes: None	Notification displayed by the system.
	13.1 [alt] Display Unsuccessful notification details	System notifies the Administrator that the system was not successfully Backed up.	When the system Backup was unsuccessful.	Entities: None Attributes: None	Notification displayed by the system.
13.2 Restore Data	13.2 Request Restore file confirmation details	System prompts the Administrator to confirm they wish to Restore the system.	When the system requires confirmation to Restore the system.	Entities: None Attributes: None	Prompt displayed by the system.
	13.2 Display success notification details	System notifies the Administrator that the System was Restored successfully.	When the system has been successfully Restored.	Entities: None Attributes: None	Notification displayed by the system.
	13.2 [alt] Display unsuccessful notification details	System notifies the Administrator that the system was not successfully Restored.	When the system Restore was unsuccessful.	Entities: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: None	
13.3 View Audit Trail Log	13.3 Display Audit Log details	System displays the details of the Audit Trail log.	When the system displays the Audit Trail log.	Entities: AuditTrail Attributes: AuditID DateTime TransactionDescription UserID	Details displayed by the system.
	13.3 [alt] Cannot display details	System notifies the Administrator that it cannot display the Audit Trail log.	When the system cannot display the Audit Trail log.	Entities: None Attributes: None	Notification displayed by the system.
13.4 Search Audit Trail Entry	13.4 Request Audit Trail search details	System requests the Administrator to provide the details of Audit Trail entry they wish to search for.	When the Administrator requests to search for an Audit Trail entry.	Entities: None Attributes: None	Prompt displayed by the system.
	13.4 Display Audit Trail entry details	System displays the details of the Audit Trail entry that the Administrator searched for.	When the system displays the Audit Trail entry details.	Entities: AuditTrail Attributes: AuditID DateTime TransactionDescription UserID	Details displayed by the system
	13.4 [alt] Audit Trail entry does not exist details	System notifies the Administrator that the Audit Trail entry cannot be found.	When the Audit Trail entry does not exist on the system.	Entities: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: None	
13.5 Add Route	13.5 Request Route details	System requests the Administrator to enter the new Route details.	When the Administrator requests to add a new Route.	Entities: None Attributes: None	Prompt displayed by the system.
	13.5 Request new Route confirmation details	System prompts the Administrator to confirm they wish to add the new Route.	When the system requests the Administrator to confirm they wish to add a new Route.	Entities: None Attributes: None	Prompt displayed by system.
	13.5 [alt] Invalid Format notification details	System notifies the Administrator that the Route details provided are invalid.	When the Administrator provides invalid Route details.	Entities: None Attributes: None	Notification displayed by the system.
	13.5 [alt] Route already exist message details	System notifies the Administrator that the Route already exists in the system.	When the Route already exists in the system.	Entities: None Attributes: None	Notification displayed by the system.
	13.5 Display Success Notification details	System notifies the Administrator that the new Route was added successfully.	When the Route has been added successfully.	Entities: None Attributes: None	Notification displayed by the system.
13.6 Search Route	13.6 Request Route Search criteria details	System requests the Administrator to provide the details of Route they wish to search for.	When the Administrator requests to search for a Route.	Entities: None Attributes:	Prompt displayed by system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				None	
	13.6 [alt] Invalid Format notification details	System notifies the Administrator that the Route details are invalid.	When the Route details provided are invalid.	Entities: None Attributes: None	Notification displayed by the system.
	13.6 Display Route Details	System displays details of the Route that the Administrator searched for.	When the system displays the Route details.	Entities: Route School Attributes: RouteID SchoolID	Details displayed by the system.
	13.6 [alt] Route does not exist notification details	System notifies the Administrator that the Route cannot be found.	When the Route does not exist on the system.	Entities: None Attributes: None	Notification displayed by the system.
13.7 Update Route	13.7 Request to update Route details	System requests the Administrator to provide the updated details of the Route.	When the Administrator requests to update a Route details.	Entities: None Attributes: None	Prompt displayed by system.
	13.7 Request update Route confirmation details	System prompts the Administrator to confirm they wish to add the new Route.	When the system requires confirmation to update the Route details.	Entities: None Attributes: None	Notification displayed by the system.
	13.7 [alt] Invalid format notification details	System notifies the Administrator that the new Route details are invalid.	When the updated Route details are invalid.	Entities: None	Notification displayed by the system.



Requirement No. and Title	Requirement Description	Purpose	Usage	Entities and Attributes	Logical Layout
				Attributes: None	
	13.7 [alt] Save unsuccessful notification details	System notifies the Administrator that the Route details were not updated successfully.	When the system did not successfully update the Route details.	Entities: None Attributes: None	Notification displayed by the system.
	13.7 Display Success notification details	System notifies the Administrator that the Route details were updated successfully.	When the system has successfully updated the Route details.	Entities: None Attributes: None	Notification displayed by the system.
13.8 Delete Route	13.8 [alt] Route does not exist notification details	Notifies the Administrator that the Route they wish to delete does not exist.	When the Administrator requests to delete a Route that does not exist.	Entities: None Attributes: None	Notification displayed by the system.
	13.8 Request Delete Route confirmation details	System prompts the Administrator to confirm they wish to delete the selected Route.	When the system requests confirmation from the Administrator to delete the Route.	Entities: None Attributes: None	Prompt displayed by system.
	13.8 Display Success notification details	System notifies the Administrator that the Route was deleted successfully.	When the system has successfully deleted the Route from the system.	Entities: None Attributes: None	Notification displayed by the system.



Virtual Visionaries
We See What You Don't

11/09/20

#BIKES4ERP



7.1 Conclusion

That table has now painted a clearer picture and explains the context diagram outputs or rather “out-flows” with further detail. This is important because the Data Flow Diagrams that follow the Context Diagram, build up from there, and those flows stay consistent and balanced at each level.



8. Validation

8.1 Introduction

The validation lists every requirement in our system. It shows which use cases, processes on the primitive level of the Data Flow Diagrams, entities in the Entity Relational Diagram satisfy the requirements individually to detail. In short, the table below validates the need and usage of each Requirement per subsystem.



8.2 Validate the details of the functional specification against the Requirements

Sub System	Requirement and Use Case	Process (DFD)	Entities
1. User	1.1 Login	1.1.1 Display User login details	None
		1.1.2 Capture and Verify User login details	None
		1.1.3 Validate user Login details	User
		1.1.4 Grant user access and Display User Home details	AuditTrail
	1.2 Logout	1.2.1 Display Logout confirmation notification details	None
		1.2.2 Display Logout successful notification details	AuditTrail
	1.3 Forgot Password	1.3.1 Display Forgot Password details	None
		1.3.2 Capture and Verify User Email Address details	None
		1.3.3 Validate inputted Email Address details	user
		1.3.4 Send Reset password link details	None
		1.3.5 Display Reset password details	None
		1.3.6 Capture and Verify new User password details	User
		1.3.7 Display Reset password Successful notification	AuditTrail
	1.4 Add User	1.4.1 Display Add User details	None
		1.4.2 Captures and validates User Details	None
		1.4.3 Verify User Details	User
		1.4.4 Generate User ID and Save User details	User
		1.4.5 Display Add Successful Notification	AuditTrail
	1.5 Search User	1.5.1 Display User Search Criteria details	None
		1.5.2 Capture User Search criteria details	None
		1.5.3 Validate User Search criteria details	None
		1.5.4 Display Searched User details	User AuditTrail
	1.6 Update User	1.6.1 Display Searched User details	None
		1.6.2 Capture Updated User details	None
		1.6.3 Validate updated User details	None
		1.6.4 Save updated user details	User
		1.6.5 Display Update Successful Notification	AuditTrail
	1.7 Delete User	1.7.1 Display Searched User details	None
		1.7.2 Delete Searched User details	User
		1.7.3 Display User Deleted notification	AuditTrail
		1.8.1 Display Add User Type details	None



Sub System	Requirement and Use Case	Process (DFD)	Entities
	1.8 Add User Type	1.8.2 Capture and validate User Type Details	None
		1.8.3 Verify User Type Details	User
		1.8.4 Generate User Type ID and Save User Type details	User
		1.8.5 Display Add Successful Notification	AuditTrail
	1.9 Search User Type	1.9.1 Display User Type Search Criteria details	None
		1.9.2 Capture User Type Search criteria details	None
		1.9.3 Validate User Type Search criteria details	None
		1.9.4 Display Searched User Type details	UserType AuditTrail
	1.10 Update User Type	1.10.1 Display Searched User Type details	None
		1.10.2 Capture Updated User Type details	None
		1.10.3 Validate updated User Type details	None
		1.10.4 Save updated User Type details	UserType
		1.10.5 Display Update Successful Notification	AuditTrail
	1.11 Delete User Type	1.11.1 Display Searched User Type details	None
		1.11.2 Delete Searched User Type details	UserType
		1.11.3 Display User Type Deleted notification	AuditTrail
2. Mechanic Schedule	2.1 Complete Job	2.1.1 Retrieve Job list details	None
		2.1.2 Display Job Tasks details	None
		2.1.3 Update job Status details	ReportFault ReportLine
		2.1.4 Display Job Completed successfully notification	AuditTrail
	2.2 View Job Tasks	2.2.1 Display Search criteria details	None
		2.2.2 Capture Search criteria details	None
		2.2.3 Validate inputted criteria details	None
		2.2.4 Display Job list details	ReportLine AuditTrail
3. Tracking Hardware	3.1 Add Antenna	3.1.1 Display Add Antenna details	None
		3.1.2 Capture and validate Antenna details	None
		3.1.3 Verify Antenna details	Antenna
		3.1.4 Generate Antenna ID and Save Antenna details	Antenna
		3.1.5 Display Add Successful Notification	AuditTrail
	3.2 Search Antenna	3.2.1 Display Antenna Search Criteria details	None



Sub System	Requirement and Use Case	Process (DFD)	Entities
		3.2.2 Capture Antenna Search criteria details	None
		3.2.3 Validate Antenna Search criteria details	None
		3.2.4 Display Searched Antenna details	Antenna AuditTrail
	3.3 Update Antenna	3.3.1 Display Searched Antenna details	None
		3.3.2 Capture Updated Antenna details	None
		3.3.3 Validate updated Antenna details	None
		3.3.4 Save updated Antenna details	Antenna
		3.3.5 Display Update Successful Notification	AuditTrail
	3.4 Delete Antenna	3.4.1 Display Searched Antenna details	None
		3.4.2 Delete Searched Antenna details	Antenna
		3.4.3 Display Antenna Deleted notification	AuditTrail
	3.5 Add Tag	3.5.1 Display Add Tag details	None
		3.5.2 Capture and validate Tag Details	None
		3.5.3 Verify Tag Details	Tag
		3.5.4 Generate Tag ID and Save Tag details	Tag
		3.5.5 Display Add Successful Notification	AuditTrail
	3.6 Search Tag	3.6.1 Display Tag Search Criteria details	None
		3.6.2 Capture Tag Search criteria details	None
		3.6.3 Validate Tag Search criteria details	None
		3.6.4 Display Searched Tag details	Tag AuditTrail
	3.7 Update Tag	3.7.1 Display Searched Tag details	None
		3.7.2 Capture Updated Tag details	None
		3.7.3 Validate updated Tag details	None
		3.7.4 Save updated Tag details	Tag
		3.7.5 Display Update Successful Notification	AuditTrail
	3.8 Delete Tag	3.8.1 Display Searched Tag details	None
		3.8.2 Delete Searched Tag details	Tag
		3.8.3 Display Tag Deleted notification	AuditTrail
4. Student	4.1 Add Student	4.1.1 Display Add Student details	None
		4.1.2 Capture and validate Antenna details	None
		4.1.3 Verify Student Details	Student
		4.1.4 Generate Student ID and Save Student details	Student
		4.1.5 Display Add Successful Notification	AuditTrail



Sub System	Requirement and Use Case	Process (DFD)	Entities
	4.2 Search Student	4.2.1 Display Student Search Criteria details	None
		4.2.2 Capture Student Search criteria details	None
		4.2.3 Validate Student Search criteria details	None
		4.2.4 Display Searched Student details	Student AuditTrail
	4.3 Update Student	4.3.1 Display Searched Student details	None
		4.3.2 Capture Updated Student details	None
		4.3.3 Validate updated Student details	None
		4.3.4 Save updated Student details	Student
		4.3.5 Display Update Successful Notification	AuditTrail
	4.4 Delete Student	4.4.1 Display Searched Student details	None
		4.4.2 Delete Searched Student	Student
		4.4.3 Display Student Deleted notification	AuditTrail
	4.5 Capture Student Marks	4.5.1 Display Capture Student Marks details	None
		4.5.2 Capture and validate Student Marks Details	None
		4.5.3 Verify Student Marks Details	StudentMark
		4.5.4 Save Student Marks details	StudentMark
		4.5.5 Display Capture Successful Notification	AuditTrail
	4.6 Search Student Marks	4.6.1 Display Student Mark Search Criteria details	None
		4.6.2 Capture Student Mark Search criteria details	None
		4.6.3 Validate Student Mark Search criteria details	None
		4.6.4 Display Searched Student Mark details	Student
	4.7 Add Mark Type	4.7.1 Display Add Mark Type details	None
		4.7.2 Capture and validate Mark Type Details	None
		4.7.3 Verify Mark Type Details	MarkType
		4.7.4 Generate MarkTypeID and Save Mark Type details	MarkType
		4.7.5 Display Add Successful Notification	AuditTrail
	4.8 Search Mark Type	4.8.1 Display Mark Type Search Criteria details	None
		4.8.2 Capture Mark Type Search criteria details	None
		4.8.3 Validate Mark Type Search criteria details	None
		4.8.4 Display Searched Mark Type details	MarkType AuditTrail
	4.9 Update Mark Type	4.9.1 Display Searched Mark Type details	None



Sub System	Requirement and Use Case	Process (DFD)	Entities
		4.9.2 Capture Updated Mark Type details	None
		4.9.3 Validate updated Mark Type details	None
		4.9.4 Save updated Mark Type details	MarkType
		4.9.5 Display Update Successful Notification	AuditTrail
	4.10 Delete Mark Type	4.10.1 Display Searched Mark Type details	None
		4.10.2 Delete Searched Mark Type	MarkType
		4.10.3 Display Mark Type Deleted notification	AuditTrail
5. Bicycle	5.1 Add Bicycle	5.1.1 Display Add Bicycle details	None
		5.1.2 Capture and validate Bicycle details	None
		5.1.3 Verify Bicycle details	Bicycle
		5.1.4 Generate BicycleID and Save Bicycle details	Bicycle
		5.1.5 Display Add Successful Notification	AuditTrail
	5.2 Search Bicycle	5.2.1 Display Bicycle Search Criteria details	None
		5.2.2 Capture Bicycle Search criteria details	None
		5.2.3 Validate Bicycle Search criteria details	None
		5.2.4 Display Searched Bicycle details	Bicycle AuditTrail
	5.3 Update Bicycle	5.3.1 Display Searched Bicycle details	None
		5.3.2 Capture Updated Bicycle details	None
		5.3.3 Validate Updated Bicycle details	None
		5.3.4 Save Updated Bicycle details	Bicycle
		5.3.5 Display Update Successful Notification	AuditTrail
	5.4 Delete Bicycle	5.4.1 Display Searched Bicycle details	None
		5.4.2 Delete Searched Bicycle details	Bicycle
		5.4.3 Display Bicycle Deleted notification	AuditTrail
	5.5 Assign A Bicycle To Student	5.5.1 Display list of Students with no Bicycle	None
		5.5.2 Display list of bicycles with available BicycleStatus	BicycleStatus
		5.5.3 Assign Bicycle to Student	Bicycle BicycleStudent BicycleStatus Student
		5.5.4 Display Assign Successful notification	AuditTrail
	5.6 Unassign Bicycle From Student	5.6.1 Display list of Students with a Bicycle assigned	None
		5.6.2 Unassign Bicycle from Student	Bicycle BicycleStatus



Sub System	Requirement and Use Case	Process (DFD)	Entities
			BicycleStudent Student
		5.6.3 Display Unassign Successful notification	AuditTrail
	5.7 Assign Tag To Bicycle	5.7.1 Display list of Bicycle without a tag details	None
		5.7.2 Display list of tags with Available TagStatus	TagStatus
		5.7.3 Assign Tag to Bicycle	Bicycle Tag TagStatus
		5.7.4 Display Assign Successful notification	AuditTrail
	5.8 Unassign Tag From Bicycle	5.8.1 Display list of Assigned Bicycle details	None
		5.8.2 Unassign Tag from Bicycle	Bicycle Tag TagStatus
		5.8.3 Display Unassign Successful notification	AuditTrail
6. Bicycle Part	6.1 Add Bicycle Part	6.1.1 Display Add Bicycle Part details	None
		6.1.2 Capture and validate Bicycle Part details	None
		6.1.3 Verify Part details	Part
		6.1.4 Generate PartID and Save Part details	Part
		6.1.5 Display Add Successful Notification	AuditTrail
	6.2 Search bicycle Part	6.2.1 Display Bicycle Part Search criteria details	None
		6.2.2 Capture Bicycle Part Search criteria details	None
		6.2.3 Validate Bicycle Part Search criteria details	None
		6.2.4 Display Searched Bicycle Part details	Part AuditTrail
	6.3 Update Bicycle Part	6.3.1 Display Searched Bicycle Part details	None
		6.3.2 Capture Updated Bicycle Part details	None
		6.3.3 Validate Updated Bicycle Part details	None
		6.3.4 Save updated Bicycle Part details	Part
		6.3.5 Display Update Successful notification	AuditTrail
	6.4 Delete Bicycle Part	6.4.1 Display Searched Bicycle Part details	None
		6.4.2 Delete Searched Bicycle Part details	BicyclePart
		6.4.3 Display Bicycle Part deleted notification	AuditTrail



Sub System	Requirement and Use Case	Process (DFD)	Entities
	6.5 Add Bicycle Section	6.5.1 Display Add Bicycle Section details	None
		6.5.2 Capture and validate Bicycle Section details	None
		6.5.3 Verify Bicycle Section details	Section
		6.5.4 Generate SectionID and Save Bicycle Section details	Section
		6.5.5 Display Add Successful Notification	AuditTrail
	6.6 Search Bicycle Section	6.6.1 Display Bicycle Section Search criteria details	None
		6.6.2 Capture Bicycle Section Search criteria details	None
		6.6.3 Validate Bicycle Section Search criteria details	None
		6.6.4 Display Searched Bicycle Section details	BicycleSection
	6.7 Update Bicycle Section	6.7.1 Display Searched Bicycle Section details	None
		6.7.2 Capture Updated Bicycle Section details	None
		6.7.3 Validate Updated Bicycle Part details	None
		6.7.4 Save updated Bicycle Section details	BicycleSection
		6.7.5 Display Update Successful notification	AuditTrail
	6.8 Delete Bicycle Section	6.8.1 Display Searched Bicycle Section details	None
		6.8.2 Delete Searched Bicycle Section details	BicycleSection
		6.8.3 Display Bicycle Section deleted notification	AuditTrail
7. Tracking	7.1 Add Log Entry	7.1.1 Capture tag details	Tag
		7.1.2 Capture and Save Log entry details	Log AuditTrail
	7.2 Search Log Entry	7.2.1 Display Log entry Search Criteria details	None
		7.2.2 Capture Log entry Search criteria details	None
		7.2.3 Validate Log Entry Search criteria details	None
8. Job Tasks	8.1 Add Fault Task	7.2.4 Display Searched Log Entry details	Log AuditTrail
		8.1.1 Display Add Fault Task details	None
		8.1.2 Capture and validate Fault task details	None
		8.1.3 Verify Fault task details	FaultTask
		8.1.4 Generate Fault ID and Save FaultTask details	FaultTask
		8.1.5 Display Add Successful Notification	AuditTrail



Sub System	Requirement and Use Case	Process (DFD)	Entities
	8.2 Search Fault Task	8.2.1 Display Fault Task Search criteria details	None
		8.2.2 Capture Fault Task Search criteria details	None
		8.2.3 Validate Fault Task Search criteria details	None
		8.2.4 Display Searched Fault Task details	FaultTask AuditTrail
	8.3 Update Fault Task	8.3.1 Display Searched Fault Task details	None
		8.3.2 Capture Updated Fault Task details	None
		8.3.3 Validate Updated Fault Task details	None
		8.3.4 Save updated Fault Task details	FaultTask
	8.4 Delete Fault Task	8.4.1 Display Searched Fault Task details	None
		8.4.2 Delete Searched Fault Task details	FaultTask
		8.4.3 Display Fault Task deleted notification	AuditTrail
	8.5 Add Maintenance Task	8.5.1 Display Add Maintenance Task details	None
		8.5.2 Capture and validate Maintenance Task details	None
		8.5.3 Verify Maintenance Task details	None
		8.5.4 Generate MaintenanceTaskID and Save MaintenanceTask details	MaintenanceTask
		8.5.5 Display Add Successful Notification	AuditTrail
	8.6 Search Maintenance Task	8.6.1 Display Maintenance Task Search criteria details	None
		8.6.2 Capture Maintenance Task Search criteria details	None
		8.6.3 Validate Maintenance Task Search criteria details	None
		8.6.4 Display Searched Maintenance Task details	MaintenanceTask AuditTrail
	8.7 Update Maintenance Task	8.7.1 Display Searched Maintenance Task details	None
		8.7.2 Capture Updated Maintenance Task details	None
		8.7.3 Validate Updated Maintenance Task details	None
		8.7.4 Save Updated Maintenance Task details	MaintenanceTask
		8.7.5 Display Update Successful notification	AuditTrail
	8.8 Delete Maintenance Task	8.8.1 Display Searched Maintenance Task details	None
		8.8.2 Delete Searched Maintenance Task details	MaintenanceTask



Sub System	Requirement and Use Case	Process (DFD)	Entities
		8.8.3 Display Maintenance Task deleted notification	AuditTrail
9. School	9.1 Add School	9.1.1 Display Add School Details	None
		9.1.2 Capture and validate School Details	None
		9.1.3 Verify School Details	School
		9.1.4 Generate SchoolID and Save School details	School
		9.1.5 Display Add School Successful Notification	AuditTrail
	9.2 Search School	9.2.1 Display School Search Criteria details	None
		9.2.2 Capture School Search criteria details	None
		9.2.3 Validate School Search criteria details	None
		9.2.4 Display Searched School details	School AuditTrail
	9.3 Update School	9.3.1 Display Searched School details	None
		9.3.2 Capture Updated School details	None
		9.3.3 Validate updated School details	None
		9.3.4 Save updated School details	School
		9.3.5 Display Update Successful Notification	AuditTrail
	9.4 Delete School	9.4.1 Display Searched School details	None
		9.4.2 Delete Searched School details	School
		9.4.3 Display School Deleted notification	AuditTrail
10. Job	10.1 Report Fault	10.1.1 Retrieve SchoolID where the request was made	School
		10.1.2 Retrieve a list of Students with that SchoolID	Student
		10.1.3 Display a List of Students in that School	None
		10.1.4 Capture BicycleID details	None
		10.1.5 Display Bicycle Blueprint with different part selection details	BicyclePart
		10.1.6 Capture Part selection details	None
		10.1.7 Display selection option details	FaultTask
		10.1.8 Capture Student selection details	None
		10.1.9 Generate JobID and Save Job details	Job
		10.1.10 Display submitted successfully notification	AuditTrail
	10.2 View Jobs	10.2.1 Display Search criteria details	None
		10.2.2 Capture Search criteria details	None



Sub System	Requirement and Use Case	Process (DFD)	Entities
		10.2.3 Validate inputted criteria details	Jobs
		10.2.4 Display Job list details	AuditTrail
	10.3 Check-In Bicycle	10.3.1 Capture Student details	None
		10.3.2 Verify Student and Bicycle details	None
	10.4 Report Maintenance	10.4.1 Capture Job and Maintenance task details	None
		10.4.2 Capture Retrieved Maintenance task details	None
		10.4.3 Generate Next maintenance date details	Maintenancetask Job
		10.4.4 Generate JobID and Save Job details	AuditTrail Job
11. Reporting	11.1 Generate Maintenance Schedule	11.1.1 Retrieve Maintenance Schedule details	Bicycle Part BusinessRule Student MaintenanceTask School Section
		11.1.2 Generate Maintenance Schedule details	None
		11.2.3 Email Maintenance Requirements schedule Report	AuditTrail
	11.2 Generate Maintenance Requirements Report	11.2.1 Retrieve Maintenance Requirements details	School Part Section BusinessRule
		11.2.2 Generate Maintenance requirements report	None
		11.2.3 Email Maintenance Requirements Report	AuditTrail
	11.3 Generate Student Report	11.3.1 Capture StudentID	None
		11.3.2 Retrieve Student Report details	Student Tag School Bicycle StudentMark StudentAttendance
		11.3.3 Generate Student Report details	None
		11.3.4 Display Student Report details	AuditTrail
		11.3.5 Export Report details	AuditTrail
	11.4 Generate Bicycle History Report	11.4.1 Capture BicycleID details	None
		11.4.2 Retrieve Bicycle History Report details	Student School



Sub System	Requirement and Use Case	Process (DFD)	Entities
			Tag FaultTask MechanicJobStatus BicycleStatus MechanicJob ReportLine MaintenanceTask
		11.4.3 Generate Bicycle History Report	None
		11.4.4 Display Bicycle History Report	AuditTrail
		11.4.5 Export Report details	AuditTrail
	11.5 Generate Log Report	11.5.1 Capture Report criteria	None
		11.5.2 Retrieve Log Report details	Antenna Log Student AntennaStatus Bicycle
		11.5.3 Generate Log Report details	None
		11.5.4 Display Log Report details	AuditTrail
		11.5.5 Export Report details	AuditTrail
	11.6 Generate Parts Used Report	11.6.1 Retrieve Parts Used details	Section Part BusinessRule
		11.6.2 Generate Parts used report	None
		11.6.3 Display Maintenance Requirements Report	AuditTrail
12. Notification	12.1 Send Student Did Not Arrive Notification	12.1.1 Retrieve a list of bicycles that haven't arrived details	Log
		12.1.2 Retrieve Student details	Bicycle Student Guardian
		12.1.3 Generate and Send Student alert notification SMS	None
	12.2 Send Student Maintenance Notification	12.2.1 Retrieve Maintenance task details	Student ScheduleMaintenance
		12.2.2 Generate Maintenance notification details	None
		12.2.3 Maintenance notification SMS details	None
13. Administrator	13.1 Backup Data	13.1.1 Display system backup details	None
		13.1.2 Perform system backup	BackupFile
		13.1.3 Display Backup successful notification	AuditTrail
	13.2 Restore Data	13.2.1 Display System Restore details	None
		13.2.2 Perform System Restore	BackupFile
		13.2.3 Display Restore Successful notification	AuditTrail
		13.3.1 Retrieve Audit trail details	AuditTrail



Sub System	Requirement and Use Case	Process (DFD)	Entities
	13.3 View Audit Trail Log		
		13.3.2 Display requested Audit trail Log details	AuditTrail
	13.4 Search Audit Trail Entry	13.4.1 Display Audit Trail search details	None
		13.4.2 Capture Audit Trail search criteria	None
		13.4.3 Display Audit Trail details	AuditTrail
	13.5 Add Route	13.5.1 Display Add Route Details	None
		13.5.2 Capture and validate Route Details	None
		13.5.3 Verify Route Details	None
		13.5.4 Generate RouteID and Save Route details	Route
		13.5.5 Display Add Route Successful Notification	AuditTrail
	13.6 Search Route	13.6.1 Display Search Route Criteria details	None
		13.6.2 Capture Search Route criteria details	None
		13.6.3 Validate Search Route criteria details	None
		13.6.4 Display Searched Route details	Route AuditTrail
	13.7 Update Route	13.7.1 Display Searched Route details	None
		13.7.2 Capture Updated Route details	None
		13.7.3 Validate updated Route details	None
		13.7.4 Save updated Route details	Route
		13.7.5 Display Update Successful Notification	AuditTrail
	13.8 Delete Route	13.8.1 Display Searched Route details	None
		13.8.2 Delete Searched Route details	Route
		13.8.3 Display Route Deleted notification	AuditTrail



8.3 Conclusion

That table has now painted a clearer picture and explains the system requirements and their usage at the lowest level of the system to detail. This is important because those Processes are the moving parts to what our system is intended to do.



9. Deliverable Conclusion & Sign-off

9.1 Conclusion

This concludes the second deliverable of the proposed system. This deliverable is purposed to provide a broad overview of the functional specifications of each Functional Requirement, along with descriptions, data models and data that go along with them. The goal is that, now you can fully envision how the core parts of our system will work and what data will be flowing through it.

9.2 Client Sign-off

I, _____, on behalf of EPI-USE, hereby confirm that I understand and agree to the stated scope of work, specifications, diagrams and requirements set out in this document – Visual Visionaries Deliverable 2 Functional Specification.

Date: _____

Signature: _____

9.3 Group Sign-off

We, as Group 17 Visual Visionaries, hereby agree and declare that each member in the group contributed equally and fully to this document, Deliverable 2 Functional Specification for INF 370 2020.

Name: _____

Signature: _____

Date: _____

Name: _____

Signature: _____

Date: _____

Name: _____

Signature: _____

Date: _____

Name: _____

Signature: _____

Date: _____

Name: _____

Signature: _____

Date: _____



10. Complexity

10.1 Complexity Matrix

INF370 - Complexity Matrix - 2020				
Topic	Level	✓	Marks	MAX
1. Special GUI	For online applications: Responsive web design	✓	3	42
	For desktop applications: Form design according to design principles (Schneiderman's golden rule on navigation applies here)			
	Appropriate use of grids/tables	✓	3	
	Appropriate use of tabs/links	✓	3	
	Use of graphs in an appropriate business context	✓	4	
	The storage and display of graphical information, like photos with a good business reason	✓	3	
	Working e-mail automatically generated from the database in an appropriate business context	✓	2	
	SMS messages automatically generated from the system in an appropriate business context	✓	2	
	Extensive user-friendly search facility	✓	3	
	At least one use of a tree to display data from the database	✓	3	
	Able to dynamically modify a data tree structure and in doing so adjusting the data in the database	✓	4	
	At least one use of a calendar view of data (not a date/time picker; not a plug-in such as Google calendar)		3	
	Uploading a file into the system with appropriate business reason		3	
	The use of audio/video in an appropriate business context		3	
	At least one use of an administrator configurable timer in an appropriate business context	✓	3	
2. Database access	At least 30 tables used (4 member groups) or 40 tables used (5 member groups)	✓	6	15
	Full referential integrity on all tables	✓	6	
	At least one use of master-detail table relationships (Schneiderman's golden rule on system status applies here)	✓	3	
3. Reports	At least 3 simple list reports in a reporting tool (no control breaks, no graphs, single table)	✓	3	15
	At least 2 transactional report with 2 or more control breaks (with heading and calculated values/totals, multiple tables)	✓	6	
	At least 1 report with adjustable criteria	✓	3	
	At least 1 management report using a graph	✓	3	
4. Flexibility	All data that can change in future should not be hard coded but maintained in a sub-module of the system (e.g. Lookup tables)	✓	6	12



	Some business rules are not hard coded but maintained in a sub-module of the system.	✓	6	
5. Error handling	All system-generated errors are trapped, and consistent, user-friendly error messages are displayed	✓	6	12
	Appropriate data validation on all input fields	✓	6	
6. Help	At least one menu item or other control that opens up a complete help document (HTML, PDF, Help-file)	✓	3	15
	Extensive context-sensitive help. E.g. calling Help on a specific screen/function will automatically open the specific help for that screen/function.	✓	6	
	Search Facility on Help		3	
	Extensive use of hints	✓	3	
7. Security	Logon screen with user ID and password and fixed user profiles	✓	3	13
	Applying two factor authentication with applicable business reason.		3	
	Encrypted passwords in database	✓	1	
	Flexible user profiles (i.e. you can dynamically add user profiles that will enable/disable access to certain parts of the system)	✓	6	
8. Audit Trail	An audit trail of all transactions in the system showing at least date, time, user, transaction type, critical data (such as amount and quantity of transaction)	✓	6	9
	Able to search the audit trail on any of the following: date, user, transaction type	✓	3	
9. Deployment	For a desktop application: Fully functional installation disks that take care of application installation requirements (install and uninstall)		3	15
	For an online application: Deployment of application to a publicly accessible web server	✓	3	
	For a mobile application: Deployment to an App Market place (such as the Play Store or the AppStore)		6	
	Deployment of the database to a remote database server	✓	3	
10. Backup and Restore	A backup and restore subsystem exist that backup/restore all data (system may exit during restore)	✓	3	3
11. Import/Export Data	Able to open Word or Excel and automatically place data in it based on the parameters provided (with a good business reason)	✓	6	9
	XML or JSON: At least 1 XML or JSON file for Importing or Exporting of data (with good business reason)	✓	3	
12. External INPUT device	Simple Link to an external INPUT device using plug-and-play technology, such as a swipe card reader, bar code reader, etc. or a native component such as a QR reader, a GPS component, etc.	✓	3	18
	Loose Link to an external INPUT device using device specific software. Data or images must seamlessly be stored in the database, but device specific software is visible to the user. (This could include a digital		6	



	camera, scanner, voice recording device, thump print reader, etc.)			
	Tight Link to an external INPUT device using device specific software. Data or images must seamlessly be stored in the database, but device specific software is not visible to the user. (This could include a digital camera, scanner, voice recording device, thump print reader, etc.)	✓	9	
13. External APPLICATION / Services	Integrate an existing web service into your application (with good business reason)		3	9
	A fully functional link to an installed external application system exists and the interface must be shown to work on the external system. Note that this excludes Microsoft Office Applications		6	
14. Multiplatform processing for an appropriate business reason	Appropriate business use of static views on an alternative platform.		3	27
	Appropriate use of dynamic views on an alternative platform (i.e. data is displayed from the system's database)		3	
	Appropriate use of substantial dynamic views on an alternative platform (i.e. both reading and writing data from the system's database)		9	
	Uploading a file through an alternative platform onto the system's database.		3	
	Substantial processing on a third platform (i.e. both reading and writing data from the system's database)		9	
15. Programming Principles	The use of a data layer to facilitate interaction between your database and your business layer	✓	3	12
	The use of an API to facilitate interaction between your business layer and your presentation layer	✓	6	
	Comprehensive use of stored procedures and/or triggers and/or jobs.	✓	3	
16. Innovative addition to the system	Any very advanced innovative addition to the system (e.g. machine learning, AI, block chain, text mining, IOT, etc.)	✓	9	9
	Maximum Complexity Marks			235
	Our Projects Complexity Marks			166