EPI-USE®

11/09/2020

#BIKES4ERP

Informatics 370 Deliverable 1 – Project Proposal

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

A MEMBER OF: groupelephant.com™ BEYOND CORPORATE PURPOSE

ERP™

Table of Contents

Table of Figures

Table of Tables

1. Deliverable Introduction

This deliverable of the INF 370 Project is the Project Proposal. It involves a first-hand introduction to the problem that we will be addressing, as well as a proposed solution. It will include detailed client information, a clear project request, preliminary investigation of the problem as well as problem itself. Additional to that, requirements for the proposed solution will be discussed in depth and a feasibility analysis will be done to analyze and contrast available solution options.

The aim of this deliverable is to provide an overview of the problems that we have identified as well as exploring the potential solutions to the identified problems.

2. Client Information

2.1 Introduction

This section gives an overview of the client, EPI-USE as well as ERP in terms of their general and background information as well as the organizational structure, management structure, business environment and area of business. The client information section also includes the organization contact details, as well as the two EPI-USE employees we will be in correspondence with.

2.2 Client History and Background

2.2.1 EPI-USE Background

EPI-USE was founded in 1998 and is a global organization with local presence in over 32 countries and are best known as the world's largest and most experienced independent SAP HR/Payroll specialist responsible for designing, building and implementing the Cloud-based, hybrid and on premises HR/Payroll systems for large, complex multinational corporations. Their focus is in Human Capital Management (HCM) surrounding Human Resources, Talent as well as Payroll. They have played an instrumental role in creating Payroll technology solutions and are the leader in deploying SAP's S/4 HANA Finance applications. In addition to their involvement with SAP they have also created tightly integrated payroll developments for more than 35 additional countries.

EPI-USE is part of groupelephant.com (ERP) which employs more than 2,700 people across 32 countries, with over 20 years of experience in community relations, poverty alleviation, and impact investment in infrastructure. ERP comprises mostly of employee-owned group of companies, NGO's and impact investment organizations – their primary strategic imperative being 'Beyond Corporate Purpose' in day-to-day activities.

EPI-USE donates 1% of their revenue to ERP with focus on the protection and preservation of Elephants, Rhinos and alleviating poverty among rural people.

2.2.2 ERP Background

ERP is a non-profit organization with over 20 years of experience in community relations, poverty alleviation, and impact investment in infrastructure. The main focus of ERP is the protection of Elephants and Rhinos, and poverty alleviation among rural People within South Africa. Groupelephant.com, ERP's parent company donates 1% of its revenue to cover all of ERP's operational expenses.

2.2.3 EPI-USE History

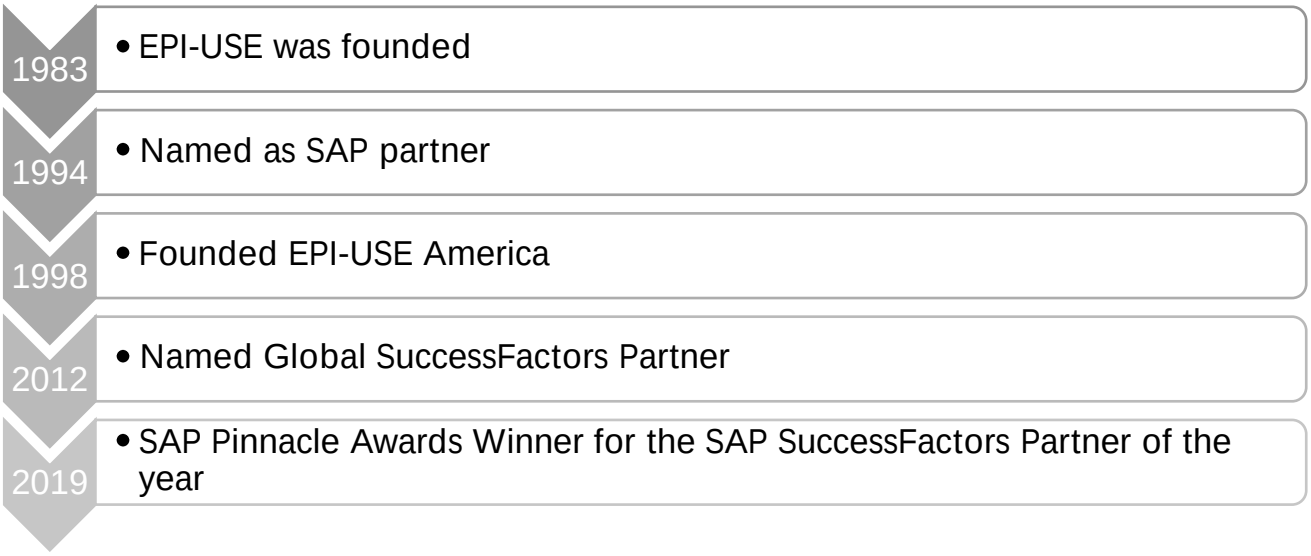

Figure 1 EPI-USE History

2.3 ERP Organizational Information

2.3.1 Organization

EPI-USE is a large independent HR/SAP specialist who form part of groupelephant.com.

groupelephant.com owns and funds the NGO that is ERP (Elephants, Rhinos and People), and the #Bikes4ERP system will be done on behalf of ERP.

2.3.2 Organizational Structure

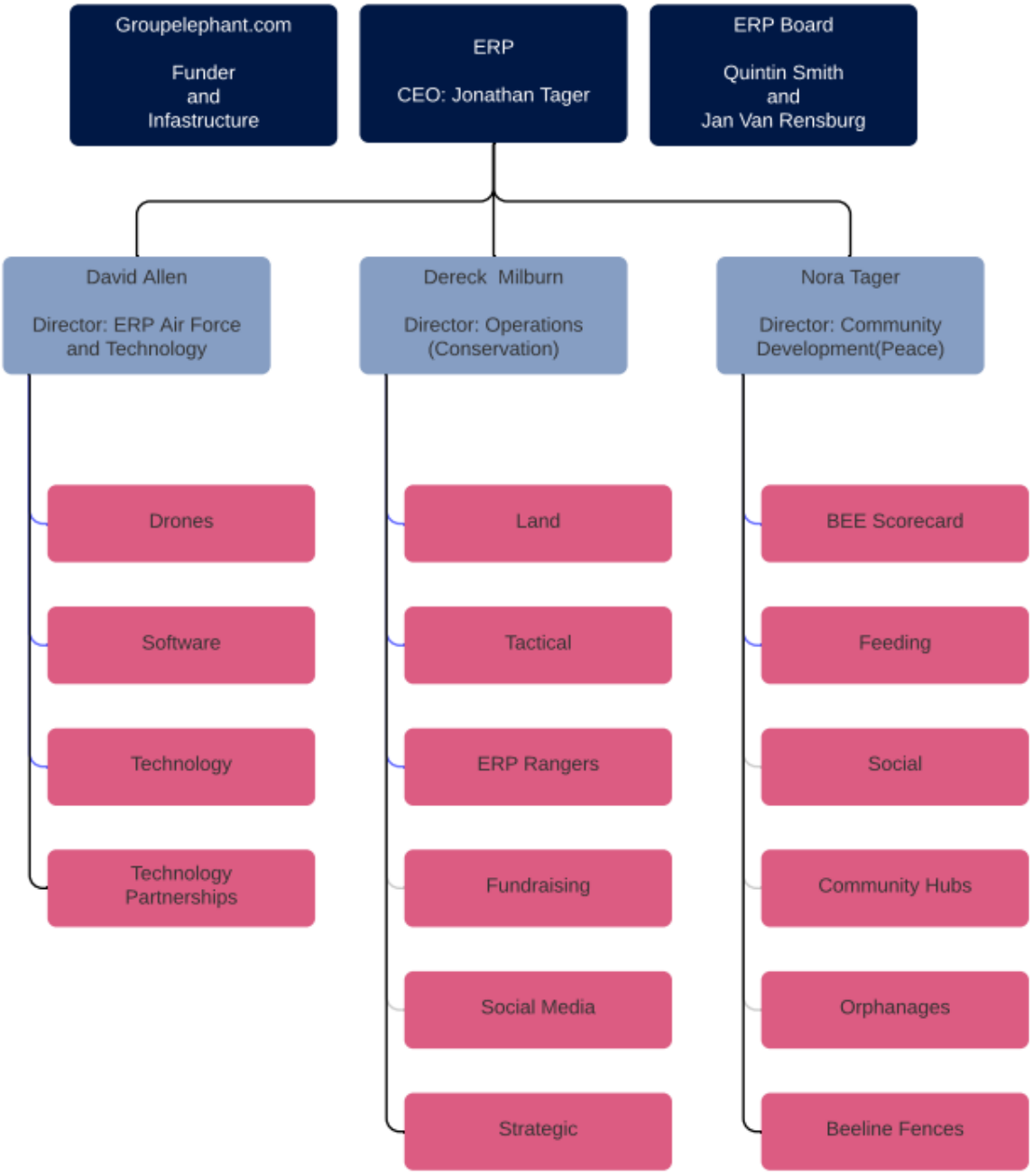

Figure 2 ERP Organizational Structure

2.3.3 Area of Business

EPI-USE operates globally, in 30 countries. EPI-USE Africa has offices that are located in Cape Town, Johannesburg and Pretoria. The offices we will be visiting for our meetings is based in Pretoria:

- 46 Ingersol Road,
  Lynnwood Glen,
  Pretoria,
  South Africa

2.3.4 Contact Details

EPI-USE contact details:
Email: info@epiuse.com
Tel: +27 12 470 2200
Fax: +27 86 669 189
Pretoria Address: 46 Ingersol Road, Lynnwood Glen, Pretoria, South Africa

EPI-USE contact person details:
Manie Kilian
Email: manie.kilian@epiuse.com
Background: Manie started with EPI-USE in 2006. He is currently the Data Migration Team Lead – ADP Faurecia. His experience includes SAP HCM and Employee Central Implementation with more than 15 years' experience in SAP, he has also implemented systems for various organizations such as Standard Bank, SARS and Barloworld.

Anelda Crous
Email: anelda.crous@epiuse.com
Mobile: +27 723837202
Office: +27 12 470 2200
Background: Anelda currently works as an HCM Associate Consultant at EPI-USE. She graduated from the University of Pretoria with Honors in BIS Information Sciences in 2018.

2.4 Conclusion

In conclusion, EPI-USE is a global organization contributing 1% of their revenue to fund ERP projects. EPI-USE has requested a system for the #Bikes4ERP initiative.

This section included EPI-USE's organizational overview and information. Contact details for contact persons were provided namely, EPI-USE Africa, Manie Kilian and our main point of contact, Anelda Crous.

3. Project Request

3.1 Introduction

In this section we will be describing the client's request for a new system and what they need from this system. An overview of the project's specifications through the project request section is also provided to further elaborate in detail, the specifications.

3.2 Project Request information

Through the #Bikes4ERP project, ERP raises funds to purchase and distribute bicycles to students in remote rural areas. Children must walk more than 10 km daily to and from school which impacts their daily attendance at school and restricting them from focussing on their educational achievements. Students enter into a 'Study to Own' contract where after 2 years of using a bicycle (provided by the project) to go to school they have the opportunity to own the bicycle provided they perform well enough academically. By addressing this issue, the short-term impact of this is that problems such as exhaustion and absenteeism is immediately negated, whilst the long-term impact is being able to enable the students to finish their education thereby reducing the likelihood of extreme poverty in rural areas.

For this 2-year period we want to deploy technology to track the children's progress to and from school by placing markers along the routes to school. Once they arrive or leave school again their caretakers or parent must be informed if they do not arrive at either destination in a certain period of time. We have challenges with cell phone coverage and data availability which will play a big role in the consideration of our solution.

In addition, we also want statistical data such as the time they spend on the bicycles for this period in order to receive data so that our clients can adjust things such as routes to further improve the system. This information needs to be recorded and available for analysis.

The next piece of data crucial to the system will be a maintenance aspect as the client has requested that a report of information such as components needed for repairs, be generated for a mechanic which will go to the school once a week to fix bikes. We therefore need to integrate a feature into our system which can received an input such as which bike has which issue and present it in a report form (for example) as the output for the mechanic.

Lastly the client would also like to track academic information so that they can further gather whether or not the project is actually producing the required results or if the project is needs adjustments.

3.3 Conclusion

In this section the client's brief request is outlined as well as what they require from our system. It is just a summary and we will elaborate further on in the document about the specifications of the system.

4. Preliminary Investigation

4.1 Introduction

In this section the focus is on defining the business' need for the new system through discussing the problem statement, goals and objectives. The problems, opportunities and directives matrix are included, and preliminary assumptions and constraints are listed. The requirement elicitation techniques checklist, descriptions and proof are also included in this section.

4.2 Problem Statement

The problem of:	School children in rural schools often walk distances of up to 16 kilometres to get to and from school daily. This leads to fatigue which affects their concentration levels in class and their overall results and performance. This situation often results in absenteeism and in some instances complete withdrawal from school. Bikes4ERP aims to solve this by providing bikes to children in these rural areas. The current problem regarding the bikes is that students school attendance and academic progress needs to be monitored to ensure that the bicycles are being used for their intended purpose. In addition to the monitoring, the bicycles require periodic maintenance by a mechanic.
Affects:	• Students (Rural Schools) • EPI-USE • Parents • Administrator • School • Mechanic
The impact of which is:	• Students (Rural Schools) – Because the students are not currently being monitored their safety is a concern. • EPI-USE – The bicycles might not be used for their intended purpose, defeating the ERP efforts with this project. • Administrator – The administrator is not able to monitor and track the student's use of the bikes and academic progress, thus unable to also carry out the 2-year use-to-own contract. • Parents – The parents are concerned with the safety as well as the whereabouts of their children when commuting to and from school. • Schools

	– The school is concerned with the overall academic performance of students as well as their daily class attendance. • Mechanic – The mechanic does not have an efficient maintenance schedule or routine for the bicycles to keep them in good working condition.
A successful solution would:	• Increase student safety and bicycle security • Enhance and Improve student academic performance • Better the mechanics work and efficiency • Help EPI-USE/ERP to better advance the #Bikes4ERP project in a more effective manner • Enable the administrator to monitor bicycle usage and academic progress • Notify parents on critical communications pertaining to student security and absence from school • Improve the involved schools pass rates

Table 1: Problem Statement

4.3 Business Goals and Objectives

From the previous information stated in this project we need to identify objectives for this project so that we can design our system in such a way that the objectives will be met when designing the system for the client.

Below will be a SMART analysis of the objective for this project:

The objective:	The objective of our system is to allow the Bikes4ERP project to gain tracking and statistical analysis capabilities so that by the end of 2020 ERP has a fully working bicycle tracking system to ensure the bike's and children's safety. To measure the impact of this aspect of the system we will be looking at statistics such as number of children which arrive safely and on time as well as number of bikes gone missing.
Specific:	The above objective is specific as it states exactly what data we need.
Measurable:	It is measurable because the data we need is numerical and can be measured.
Achievable:	This objective is achievable as we have the resources, knowledge and means to create a system which will be able to track the bikes and student's arrivals and departures.
Relevant:	This objective is extremely relevant as this is one of the key requirements of the projects (increased student safety).
Timely:	This objective is timely because it specifies that the tracking needs to be done by the end of 2020.

The objective:	The next key objective of the system would be tracking their academic performance so that our clients can monitor whether or not the project needs adjustment or not at the end of 2020.

Specific:	The above objective is specific as it states exactly what data we are tracking.
Measurable:	It is measurable because marks are considered measurable data.
Achievable:	This objective is achievable as the student's academic records are available to us and available for us to use.
Relevant:	This objective is relevant because without this data, the client cannot gauge whether or not the project is having the desired impact.
Timely:	This objective is timely because it specifies when this data needs to be available by.

The objective:	Another part of this system will be the capability for the mechanic to get reports every week on what repairs he will need to perform based on the data received from students so that he can adequately prepare for when he goes to the school for repairs.
Specific:	The above objective is specific as it states exactly what needs to be generated for the mechanic.
Measurable:	It is measurable because we can use see whether or not the system is producing the required reports every week for the mechanic.
Achievable:	This objective is achievable as we have the resources, knowledge and means to create a system which will be able to generate reports.
Relevant:	This objective is relevant as the mechanic currently has no way of preparing what he needs for the following week.
Timely:	This objective is timely because it specifies that a report needs to be generated every week.

Table 2:Business Goals and Objectives

4.4 Problems, Opportunities and Directives

PROJECT: #Bikes4ERP	PROJECT MANAGER: Tapiwarufaro Ndoro
CREATED BY: Hansen Li	LAST UPDATED BY: Hansen Li
DATE CREATED: 02/21/2020	DATE LAST UPDATED: 02/21/2020

Brief Statement of Problem	The impact the problem is causing	Expected benefits from any potential solution	How quickly can the problem potentially be resolved	What is the underlying source of the problem?	What will it cost to solve the problem?
1. Tracking students on their bikes	The administrator is unable to determine	If the system is able to track the	6 months	The current system does not have any	R0; business knowledge input

	whether or not students are arriving at school	students, then not only will the student's safety be increased but so will the bikes		tracking features	from the client; 9 months of hard dedicated work by 5 INF370 students
2. Lack of data available to EPI-USE	EPI-USE is unable to determine whether or not the project is actually reaching its goals because they do not have data to work with	If the system works as it should then EPI-USE will be able to use the data, they receive to modify and improve the project	6 months	The current system does not have any tracking features	R0; business knowledge input from the client; 9 months of hard dedicated work by 5 INF370 students
3. Bikes are going missing due to theft or misplacement	Since bikes are going missing, the next students will either not have bikes to use of NGO's will have to replace the missing bikes and increasing costs	If the system can successfully track the whereabouts of bikes as well as its users, then the amount of missing bike cases will decrease	6 months	Due to the environment or neglect, bikes are going missing	R0; business knowledge input from the client; 9 months of hard dedicated work by 5 INF370 students
4. The mechanic often does not know what components he needs	Because the mechanic doesn't know what parts he needs or what needs fixing he is unable to	The system will allow the mechanic to prepare in advance the components he needs for	6 months	There is no way for the mechanic to know what components he requires in	R0; business knowledge input from the client; 9 months of hard

	adequately prepare for when he goes to do repairs	the next week		this current setup	dedicated work by 5 INF370 students

Brief Statements of Opportunity	Urgency	Visibility	Annual Benefits	Priority or Rank	Proposed Solution
1. Implement a tracking system for the bikes	High	Low	----	1	New development
2. Design a new mechanic maintenance system	High	Medium	-----	2	New development
3. Keep track of whether or not the student's academics are improving	Moderate	Low	------	3	New development
4. Notify parents when students don't arrive at school	High	Medium	-----	2	New development
5. Keep track of bike parts	Moderate	Low	----	2	Digitalize the how the mechanic keeps track of parts now

Brief Statements of Directive	Urgency	Visibility	Annual Benefits	Priority or Rank	Proposed Solution
1. EPI-USE needs to track the academic data so that reports	High	Low	------	2	New development

can be generated for them.					
2. EPI-USE wants to know the traveling distances of the bikes on average	Medium	Low	------	1	New development
3. Group elephant is a NGO which means that everything from this system is a donation therefore no profits are to be made	Low	High	-------	3	We are not allowed to take any capital from this program

Table 3: Problems, Opportunities and Directives

4.5 Preliminary Assumptions

4.5.1 Constraints

After being briefed on the project, the group identified the following constraints that we will face in this project:

1. The largest constraint is a lack of connection in the area which we are working in. Because there is minimal communications infrastructure, technologies such as Wi-Fi and 3G were already ruled out when designing the project.
2. Due to the education levels of this community, we need to design this system as simple as possible and in such a way that someone which is illiterate or not adequate in English can still use the system.
3. The tracking devices will have to be stored on bikes which means that they need to be rugged and durable as well.
4. The trackers themselves must also be easily serviceable and relatively inexpensive so that if something does happen to them, they can be either easily fixed or easily replaced.
5. This is a fairly big system which means that time constraints will be a big challenge in this project.
6. Although this project is funded by an NGO, we still need to ensure low costs in order to make the system feasible for the client.

4.5.2 Assumptions

1. We are assuming that the school is currently working off a paper-based system for admin.
2. The administrator will have the majority of control over the backend of the system.
3. The children's literacy level is not high
4. The mechanic's literacy level is not high
5. The students have to travel long distances to school
6. The academic records are recorded

4.6 Elicitation Techniques

Our elicitation strategy involved a mixture of different techniques interwoven together in several meetings:

- 11 February 2020: Received Project Request from EPI-USE for the #Bikes4ERP Project
- 12 February 2020: Contact meeting as the Virtual Visionaries
- 13 February 2020: First meeting with Client plus Interview with Mr. Manie Killian
- 18 February 2020: Brainstorming session meeting as the Virtual Visionaries
- 20 February 2020: Requirements Workshop with EPI-USE representatives

Below are the elicitation techniques that were utilized to gather user requirements/stakeholder needs:

- Brainstorming
  - Virtual Visionaries opted for this because as it allowed the opportunity to sit in and get answers to specific questions pertaining to how the stakeholders would prefer to have their problem to be resolved. It helped in finding the actual source of certain underlying issues as well as providing the opportunity to ask further what options would be feasible as solutions to meet those needs.
- Interviews
  - This helped to have a more direct interaction with individuals directly concerned with this system (in this case, it was with Manie Killian who is directly overseeing this project, thus making him a major stakeholder). The Interview with him allowed us to be able to get a clear picture of what is currently at hand.
- Requirements workshop
  - This helped when we had to clarify what the real requirements are. It gave us an opportunity to have experts sit in with us who have had experience with this kind of "tracking-project" as well as understand the project enough to give us clarity on what major focus areas require most of our attention.

4.7 Elicitation Technique Checklist, Details and Proof

Brainstorming

Preparation checklist

No	Task	Done	Date Completed
1	Define area of interest - Bike tracking, Maintenance System as per project request	✓	Tuesday 11/02/2020
2	Define time limit - Must be completed by Mid-October 2020	✓	Thursday 13/02/2020
3	Identify participants - All Group 17 Members	✓	Wednesday 12/02/2020
4	Identify facilitator - Mr Manie Killian	✓	Wednesday 12/02/2020
5	Invite participants - Invites sent via google calendar invite	✓	Wednesday 12/02/2020 .
6	Invite facilitator - Google calendar invite sent by Ms Anelda Crous	✓	Wednesday 12/02/2020
7	Meet with participants to explain expectations - Major expectation was to confirm the project scope.	✓	Wednesday 12/02/2020
8	Establish evaluation criteria - Evaluate based on fundamental organizational needs	✓	Wednesday 12/02/2020
9	Book venue/meeting room - Ms Anelda Crous booked a boardroom for us at EPI-USE	✓	Thursday 13/02/2020

During Elicitation checklist

No	Task	Done	Date Completed
1	Share new ideas - New tracking technologies: RFID (Radio-Frequency Identification) active tags, GPS (Global Positioning System) - New Notification ideas on the table: SMS notifications.	✓	Friday 14/02/2020
2	Record all ideas - Meeting minutes captured in detail showing all ideas suggested & considered, regardless of cost constraints. -> Designing a maintenance system for the admin -> having no login features for the student, just USSD codes	✓	Tuesday 13/02/2020

	-> eliminate need for smartphones/data -> RFID tags have open source server-side code we can manipulate		
3	Build on ideas	✓	Thursday 13/02/2020
	- Further questions were noted down to further probe on the potential technologies to be pursued. E.g. -> How feasible is the GPS option cost wise? -> is		
4	Elicit as many ideas as possible	✓	Friday 14/02/2020
	- Several ideas from USSD codes, RFID active tags, SMS notifications, road markers, GPS, etc.		

After Elicitation checklist

No	Task	Done	Date Completed
1	Discuss and evaluate ideas	✓	Tuesday 18/02/2020
	- Discussed the needs and analysed what we would be able to tackle and how we would.		
2	Create list of ideas	✓	Tuesday 18/02/2020
	- A shortlist of technologies and ideas for implementation were captured, awaiting the next meeting with EPI-USE. ->GPS (Global Positioning System) -> USSD (Unstructured Supplementary Service Data) -> RFID (Radio Frequency Identification)		
3	Rate ideas	✓	Tuesday 18/02/2020
	- The listed ideas were ranked according to the following criteria: cost, operational feasibility, team skills.		
4	Distribute final list of ideas	✓	Wednesday 19/02/2020
	- Sent a copy of our shortlist to Ms. A. Crous for review in preparation for the next meeting/requirements workshop.		
5	Schedule follow-up if needed	✓	Tuesday 18/02/2020
	- Booked a requirements workshop with EPI-USE: representatives from the functional and operational team		

Interviews

Preparation checklist

No	Task	Done	Date Completed

1	Decide which type of interview - Carried out an informal interview with Mr. Manie Killian.	✓	Wednesday 12/02/2020
2	Decide on interview goal - Have a clearer understanding of the core user needs, their level of understanding and core competencies. Priorities of needs also needed.	✓	Wednesday 12/02/2020
3	Create list of questions -> What are the main users of the system? -> What will they use the system for? -> What resources are available? -> What are the time constraints? ->What is the budget at hand? -> What are the business rules? -> What would be additional desirable features? -> Is it necessary to track the student on the bicycle the entire time they're cycling to school and back?	✓	Wednesday 12/02/2020
4	Identify potential interviewees - Mr Manie Killian, representative from EPI-USE on the #Bikes4ERP project.	✓	Thursday 13/02/2020
5	Decide on location for interview - We requested a meeting with Ms. Anelda Crous and she booked the venue for us at EPI-USE.	✓	Thursday 13/02/2020
6	Invite interviewees - Ms. Anelda Crous invited Mr. Killian to our interview.	✓	Wednesday 12/02/2020
7	Send questions to interviewees (optional) - Opted not to do this.	-	-

During Elicitation checklist

No	Task	Done	Date Completed
1	Describe purpose of interview - Obtain a clearer understanding of the core user requirements from someone who interacts with the users.	✓	Thursday 13/02/2020
2	Confirm interviewees' roles - Reiterated Mr. Killian's role during this meeting (as the representative of ERP/EPI-USE #Bikes4ERP Project).	✓	Thursday 13/02/2020
3	Address any concerns - Concerns on technological limitations were raised in light of the budget and team abilities	✓	Thursday 13/02/2020
4	Explain how information will be recorded and shared - Mr Ndoro (Project Manager) will handle all group communications with Ms Crous on behalf of EPI-USE.	✓	Thursday 13/02/2020
5	Ask predefined questions	✓	Thursday 13/02/2020

	- List of questions prepared prior were posed. Further questions were asked as they came up,		
6	Summarize the session	✓	Thursday 13/02/2020
	- Understood that the main users of the system were the Administrator(s), Student(s) and Mechanic. The major needs for the Administrator are to track the student movements on the bicycle as well as their academic progress. The student's needs are those of reporting bicycle faults. The mechanics main requirement is to be able to carry out his maintenance and repairs more efficiently. Student does not need to be fully tracked the entire trip, but their departure and arrivals from a certain checkpoint and school need to be captured.		

After Elicitation checklist

No	Task	Done	Date Completed
1	Organize information	✓	Friday 14/02/2020
	- Made a formal capture of meeting minutes, as well as the collection of all interview responses, making them relevant to our needs and making it functional.		
2	Confirm results with interviewees	✓	Monday 17/02/2020
	- Sent follow-up emails to confirm the conclusions drawn from the meeting on behalf of Mr. Killian.		
3	Share information with stakeholders	✓	Monday 02/03/2020
	- Sent Final requirements/needs to the client for review before final submission of project proposal (Deliverable 1).		
4	Schedule follow-up interview if needed	✓	Tuesday 18/02/2020
	- Pending interview with the technical and functional team representatives due for the 20th of February in a requirements workshop.		

Workshops

Preparation checklist

No	Task	Done	Date Completed
1	Define purpose and desired outcomes	✓	Tuesday 18/02/2020
	- To further narrow down the technology options and ideas on how to tackle the project.		
2	Identify stakeholders	✓	Monday 17/02/2020
	- Mr. Christian Wheeler, Mr. Ruan Maass, Mr Manie Killian.		
3	Identify facilitator and scribe	✓	Tuesday 18/02/2020

	- Project Manager (Mr. Ndoro) facilitated whilst Mr. Li captured meeting minutes the scribe.		
4	Create agenda	✓	Tuesday 18/02/2020
	- Run through a list of questions compiled and going through the expectations from the technical team (ERP). ->Sign up restrictions -> Should we incorporate donations? -> Parents & Students level of usage of the system? -> How to go about verification of users or parents (security) -> How limited does our system have to be to accommodate for the possible lack of technology & devices? ->What are the system priorities? -> Is it just a bike management system or a full website?		
5	Determine how outputs will be captured	✓	Tuesday 18/02/2020
	- Set a scribe in place who will document every activity of the meeting, including the feedback and answers received.		
6	Arrange venue/meeting room	✓	Monday 17/02/2020
	- EPI-USE arranged a conference room for the meeting		
7	Arrange equipment	✓	Tuesday 18/02/2020
	- All equipment was provided by EPI-USE, notebooks, pens. No need to use any projectors/screens at this point.		
8	Send agenda	✓	Monday 17/02/2020
	- Agenda was communicated to all meeting participants via Google Calendar invite/ Google mail.		
9	Conduct pre-workshop interviews	×	-
	- Our group did not carry out any of these.		
10	Define workshop roles	✓	Thursday 20/02/2020
	- Definitive roles were informally set-up and communicated to all parties, so as to come ready.		

During Elicitation checklist

No	Task	Done	Date Completed
1	State purpose and desired outcomes	✓	Thursday 20/02/2020
	- Iron out the technologies and ideas to be considered for implementation. Have better clarity on the way forward.		
2	Establish agreed-upon ground rules	✓	Thursday 20/02/2020
	- Informal rules were set out, basically allowing each speaker their time to speak freely without interruption.		
3	Work through agenda	✓	Thursday 20/02/2020

- BLE Beacons to be consider for tracking, together with more research to be done on USSD codes.
Questions asked:
-> Sign up restrictions
-> Should we incorporate donations?
-> Parents & Students level of usage of the system?
->How to go about verification of users or parents (security)
-> How limited does our system have to be to accommodate for the possible lack of technology & devices?
-> What are the system priorities?
-> Is it just a bike management system or a full website?

After Elicitation checklist

No	Task	Done	Date Completed
1	Document results	✓	Tuesday 25/02/2020
	- Meeting minutes were captured extensively. Captured in both audio and text format, plus personal scribble notes by each team member: -> Cost is not critical, as long as it integrates into the system -> SMS messaging system is viable -> Account for central sensor -> We need a maintenance table so that the mechanic can gather enough information (i.e. we need a full-on maintenance system) -> The more simplified the system, the better -> Do not concern yourself with actual tracking too much -> Consider using BLE Beacons instead of GPS tracking -> The school should be the central point of data (collection) -> The main points will be the mechanic, students and data --->> Logging in the students ->> Keep in mind the literacy levels of every user of the system ->> Bikes are a big part of our system. Focus on them more. ->> Mechanics data is very important so that reports can be generated ->> Academic reports are another system as well for the school for the tracking of student academic progress (facilitated by the Administrator)		
2	Follow-up on open action items	✓	Tuesday 25/02/2020

	- Emailed Ms. Crous to inquire on specific questions pertaining to which student data is to be captured, what weightings to utilize for the feasibility matrix, etc.		
3	Distribute documentation to attendees and stakeholders - Copies were sent of outcomes of the requirements workshop to the group members, and any loopholes were clarified via email.	✓	Tuesday 25/02/2020

Table 4: Elicitation Technique checklists

4.8 Conclusion

In conclusion, the preliminary investigation of the project drew more focus and clarity for defining the business' need for the system through defining the goals as well as the problems and opportunities by making use of various requirement elicitation techniques.

5. Problem Analysis

5.1 Introduction

The problem analysis section focuses on assessing capability gaps, it includes a rich picture which represents the current Bikes4ERP system, it also includes a table that assesses and compares the current and required system capabilities, capability gaps and recommendations.

5.2 Rich Picture

Note: There is not many technical aspects in this system as it currently does not have any.

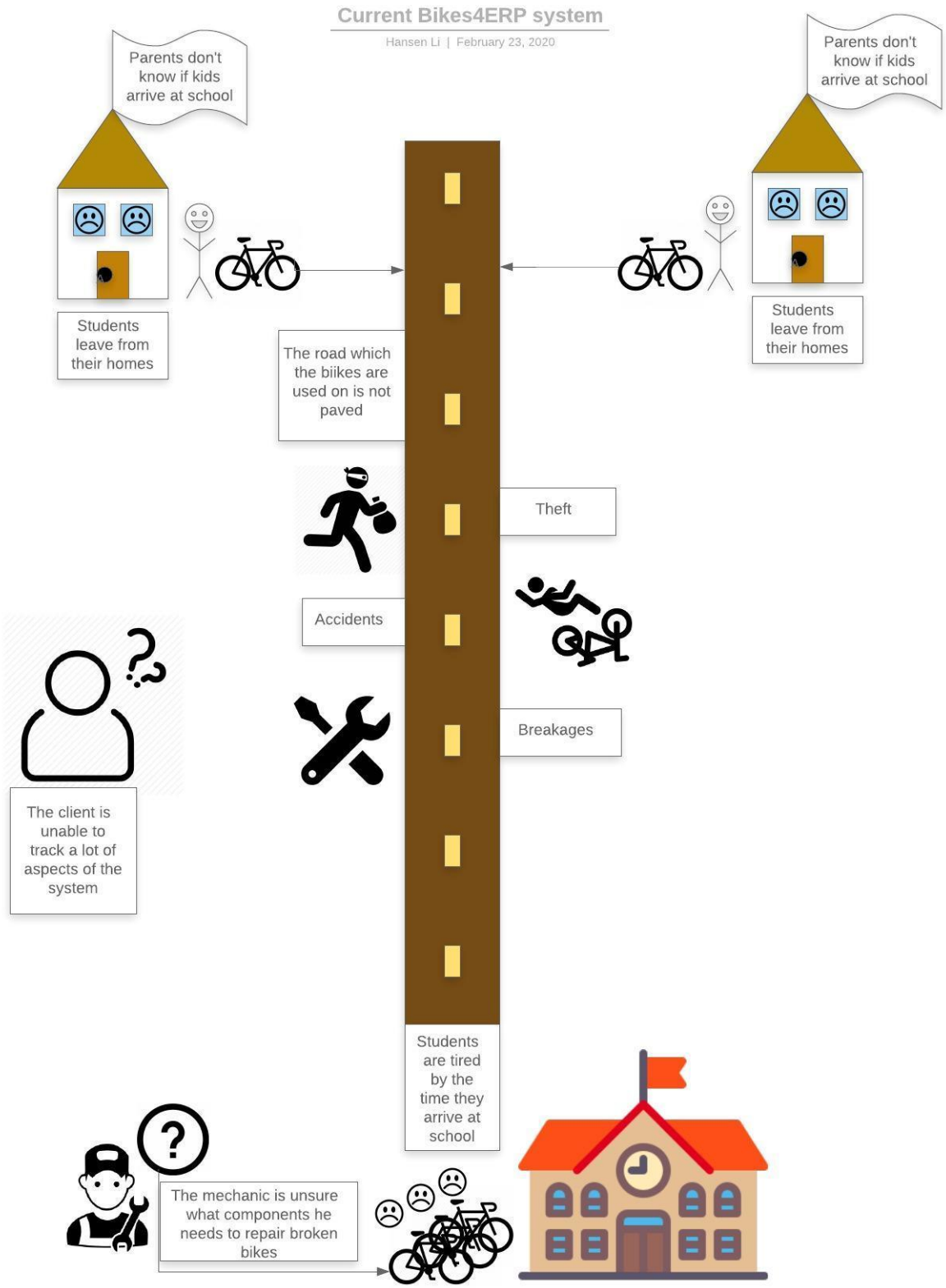

Figure 3 Current System Rich Picture

5.3 System Capabilities

Current system's capabilities	Required system's capabilities	Capability gaps	Recommendations for each gap
Introduction: The current system's capabilities are extremely limited and technologically underdeveloped. Below will be more specific capabilities of the current system.	Introduction: The required system would need to address all the previously mentioned shortcomings of each individual capability of the existing system.	Introduction: Below will be a list of capability gaps that we have identified between the current system and the required system.	Introduction: The recommendations for each gap will be fairly the same in the sense that the required system needs to address all of the capability gaps individually
Currently Bikes4ERP provides the students of the school with bikes so that they can ride to and from school every day but since there is no system in place to track the arrival and departure times of bikes, parents do not know the whereabouts of their children.	The required system will need to be able to track the arrival and departure times of the bikes so that if a child does not arrive or arrives extremely late, the parents or guardians can be contacted and informed about this information.	The gap between the current system and the required system with regards to this capability is the lack of knowing when a child departs and when (or even if) they arrive at school.	To address the tracking capability gap we have decided to implement a RFID system whereby each time a bike passes a checkpoint, an entry will be created in the new system so that the whereabouts of the child and their bike is known.
There is a mechanic which repairs the broken bikes but because he cannot be notified of what components he needs and what maintenance he is to perform, he is un-prepared most of the times. He also does not know what maintenance he will have to perform the following week.	The next capability required by the system would be the need for a way to track the maintenance and repairs of the bikes so that the mechanic can adequately prepare before he goes on site to fix the bikes. This will include information such as what is wrong with each bike and the parts and components, he'll	The gap between the current system and the required system is that the required system will have a way to record information such as parts needed, repair required, which bikes needs to be repaired and other relevant data. This will allow the	The way to address this gap is that the required system will need to include a function to record all the entries of the bikes which need servicing as well as the repairs that are needed for those bikes and their respective components. Once recorded, the mechanic needs to have a way to view this information in a

	need to fix the bikes so that he can prepare beforehand.	mechanic to adequately prepare in advance when using the required system compared to not knowing what is needed in the current system.	understandable form.
Currently if a student has a problem with their bike, they have no way of letting anyone know what is wrong with their bike.	The required system is going to introduce an interface where the students can specify what is wrong with their bike once they get to school. It will be easy to use and make use of pictures and symbols instead of words so that children of all ages will be able to use the form.	The current system has no way of students letting the mechanic know what is wrong with the bike but in the required system the students have an easy and convenient of reporting their problems and an easy way for the mechanic to see the problems.	The required system will offer an easy to use interface for the students to report their problems. Students can use the system to say which component is wrong with the bike, when it broke and other additional notes they can tell the mechanic that they think will help him.
In the current system the only form of information the clients have about the students is which student has a bike.	The required system will need to record data such as which student has which bike, when the student received the bike and whether or not the student can keep his/her bike after the 2 years.	There is no way to record data in a reliable way which means that the clients do not have enough information to improve the system	The required system will record data of the students marks so that clients can decide whether or not the child has performed well enough to keep their bike or not. The required system will also have to track average time traveled, average distance traveled and average departure and arrival times so that

			a report can be generated for the client.

Table 5: System Capabilities

5.4 Conclusion

In conclusion, an overview of the current system as well as the current and required system capabilities have been provided and discussed in this section.

6. Requirements Analysis

6.1 Introduction

The requirements analysis section of the deliverable forms a critical part of the project as it deals with the functional and non-functional requirements of the new system to be developed. This section also discusses clearly identified user acceptance criteria for each functional requirement.

6.2 Functional Requirements list

1. User Sub-system
   - 1.1. Login
   - 1.2. Logout
   - 1.3. Reset user password
   - 1.4. Add User
   - 1.5. Search User
   - 1.6. Update User
   - 1.7. Delete User
   - 1.8. Add User Role
   - 1.9. Search User Role
   - 1.10. Update User Role
   - 1.11. Delete User Role

2. Mechanic tasks Sub-system
   - 2.1. Add mechanic task
   - 2.2. Search mechanic task
   - 2.3. Delete mechanic task
   - 2.4. Update mechanic task

3. Tracking Sub-system
   - 3.1. Add Beacon
   - 3.2. Search Beacon
   - 3.3. Update Beacon
   - 3.4. Delete Beacon
   - 3.5. Add Beacon status
   - 3.6. Search Beacon status
   - 3.7. Update Beacon status
   - 3.8. Delete Beacon status
   - 3.9. Add Tag
   - 3.10. Search Tag
   - 3.11. Update Tag
   - 3.12. Delete Tag

4. Student Sub-system
   - 4.1. Add student

4.2. Search Student
4.3. Update Student
4.4. Delete Student
4.5. Add Student Marks
4.6. Search Student Marks
4.7. Add student status
4.8. Search student status
4.9. Update student status
4.10. Delete student status

5. Bicycle Sub-system
   5.1. Add Bicycle
   5.2. Search Bicycle
   5.3. Update Bicycle
   5.4. Delete Bicycle
   5.5. Assign Student
   5.6. Add bicycle status
   5.7. Search bicycle status
   5.8. Update bicycle status
   5.9. Delete bicycle status

6. Bicycle Part Sub-system
   6.1. Add Bicycle Part
   6.2. Search Bicycle Part
   6.3. Update Bicycle Part
   6.4. Delete Bicycle Part

7. Tracking Log Sub-system
   7.1. Create Log entry
   7.2. Search Log entry

8. Maintenance log sub-system *(A report of all maintenance requirements will be generated from this)*
   8.1. Report Fault
   8.2. Delete Fault
   8.3. Search Fault
   8.4. Add maintenance task
   8.5. Update maintenance task
   8.6. Search maintenance task
   8.7. Delete maintenance task

9. Mechanic Sub-system
   9.1. Add Mechanic
   9.2. Search Mechanic
   9.3. Update Mechanic

9.4. Delete Mechanic

10. School Sub-system
    10.1. Add School
    10.2. Search School
    10.3. Update School
    10.4. Delete School

11. Job Sub-system
    11.1. Add Job
    11.2. View Job

12. Reporting Sub-system
    12.1. Generate Maintenance schedule
    12.2. Generate Maintenance requirements report
    12.3. Generate Student report
    12.4. Generate Bicycle History report
    12.5. Generate Log report

13. Backup and Restore Sub-system
    13.1. Backup Data
    13.2. Restore Data

6.3 Functional Requirement Description and Detail

6.3.1 Sub-system 1: User

6.3.1.1 Login

Functional Requirement	Explanation
Requirement number:	1.1
Requirement name:	Login
Requirement short description:	The system must allow a User to login.
Requirement detailed description and constraints:	• The system must allow all registered Users to log into the system via their username and password. • The system must compare the User's password and username with the User data stored on the database to allow or deny access.

	• During the login process, the system will determine the User's level of access and will take the User to the relevant functionality.
Business rules applicable to this requirement	• Only Users that are registered on the system must be able to login. • Administrator and Mechanic have different levels of access regarding the functionally of the system.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.1.2 Logout

Functional Requirement	Explanation
Requirement number:	1.2
Requirement name:	Logout
Requirement short description:	The system must allow a User to logout of the system.
Requirement detailed description and constraints:	• The User must be logged into the system. • Users must be logged out if the "logout" button is clicked. Users must also be automatically logged out if they reach 15 minutes of inactivity on the system. • Once logged out, the User will be redirected to the login screen.
Business rules applicable to this requirement	• The user must already be logged into the system in order to logout.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.1.3 Reset User Password

Functional Requirement	Explanation

Requirement number:	1.3
Requirement name:	Reset User Password
Requirement short description:	The system must allow the User to reset their password if it is forgotten.
Requirement detailed description and constraints:	• If the "Forgot Password" link is clicked, they will be prompted to enter the email address. • The system will verify that the User is registered on the system. • The User will receive a reset password link and be prompted to enter their new password. • The system will update the Users password.
Business rules applicable to this requirement	• The email address entered must match a registered User's password on the database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.1.4 Add User

Functional Requirement	Explanation
Requirement number:	1.4
Requirement name:	Add User
Requirement short description:	The system provides an interface for adding a new User into the system.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a new User and their relevant information. • The Administrator will have to specify the user role of the new User. • The user role must already exist on the system before the new User can be assigned. • The new User must not already exist on the system.

Business rules applicable to this requirement	• Only an Administrator can add a new User. • The Administrator must be logged in to add a new User.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.1.5 Search User

Functional Requirement	Explanation
Requirement number:	1.5
Requirement name:	Search User
Requirement short description:	The system must have functionality to search for a User.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for a User through the use of a search function. • The searched User must already exist on the system.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can use the search User function. • The Administrator must be logged into search for a User.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.1.6 Update User

Functional Requirement	Explanation
Requirement number:	1.6
Requirement name:	Update User
Requirement short description:	The system provides an interface to update a User on the system.

Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to update a User's information on the system. • The User must exist on the system. • The updated User information must be valid.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can update a User. • The Administrator must be logged in to update a User.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.1.7 Delete User

Functional Requirement	Explanation
Requirement number:	1.7
Requirement name:	Delete User
Requirement short description:	The system provides an interface to delete a User from the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to delete a User's information from the system. • The user must exist on the system. • The delete function must satisfy all referential integrity constraints.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can delete a User and their information. • The Administrator must be logged in the delete a User. • A deleted User's information must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.1.8 Add User Role

Functional Requirement	Explanation

Requirement number:	1.8
Requirement name:	Add User Role
Requirement short description:	The system provides an interface for adding a new User Role into the system.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a new User role. • The Administrator will have to specify the access levels of the new User role. • The new User role must not already exist on the system.
Business rules applicable to this requirement	• Only an Administrator can add a new User role. • The Administrator must be logged in to add a new User role.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.1.9 Search User Role

Functional Requirement	Explanation
Requirement number:	1.9
Requirement name:	Search User Role
Requirement short description:	The system must have functionality to search for a User role.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for a User role through the use of a search function. • The searched User role must already exist on the system.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can use the search User role function. • The Administrator must be logged into search for a User role.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.1.10 Update User Role

Functional Requirement	Explanation
Requirement number:	1.10
Requirement name:	Update User Role
Requirement short description:	The system provides an interface to update a User role on the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to update a User role on the system. • The User role must exist on the system. • The updated User role access levels must be valid.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can update a User role. • The Administrator must be logged in to update a User role.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.1.11 Delete User Role

Functional Requirement	Explanation
Requirement number:	1.11
Requirement name:	Delete User role
Requirement short description:	The system provides an interface to delete a User role from the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to delete a User role from the system. • The User role must exist on the system. • The delete function must satisfy all referential integrity constraints.

Business rules applicable to this requirement	• Only an Administrator or an authorized function can delete a User role from the system. • The Administrator must be logged in the delete a User role. • A deleted User role must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.2 Sub-system 2: Mechanic tasks

6.3.2.1 Add Mechanic tasks

Functional Requirement	Explanation
Requirement number:	2.1
Requirement name:	Add Mechanic tasks
Requirement short description:	The system provides an interface for adding a new User Role into the system.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a new Mechanic task. • The Administrator will • The new mechanic task must not already exist on the system.
Business rules applicable to this requirement	• Only an Administrator can add a new User role. • The Administrator must be logged in to add a new User role.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.2.2 Search Mechanic tasks

Functional Requirement	Explanation
Requirement number:	2.2

Requirement name:	Search Mechanic tasks
Requirement short description:	The system provides an interface for the search functionality to find a mechanic task in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for an existing mechanic task and the relevant information will be displayed. • The student being searched needs to be registered in the system. • The administrator must be logged in to the system to search an existing student
Business rules applicable to this requirement	• Only an administrator can search a student in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.2.3 Update Mechanic task

Functional Requirement	Explanation
Requirement number:	2.3
Requirement name:	Update Mechanic task
Requirement short description:	The system provides an interface to update a mechanic task on the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to update a mechanic tasks information on the system. • The mechanic tasks must exist on the system. • The updated mechanic tasks information must be valid.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can update a mechanic task. • The Administrator must be logged in to update a mechanic task.

Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.2.4 Delete Mechanic task

Functional Requirement	Explanation
Requirement number:	2.4
Requirement name:	Delete Mechanic task
Requirement short description:	The system provides an interface to delete a Mechanic task from the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to delete a mechanic task information from the system. • The mechanic task must exist on the system. • The delete function must satisfy all referential integrity constraints.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can delete a mechanic task and its information. • The Administrator must be logged in the delete a mechanic task. • A deleted mechanic tasks information must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.3 Sub-system 3: Tracking

6.3.3.1 Add Beacon

Functional Requirement	Explanation
Requirement number:	3.1
Requirement name:	Add Beacon

Requirement short description:	The system provides an interface for adding a new beacon into the system.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a new beacon to the system and its relevant information including the location of the beacon, the Beacon number as well as the date it was added. • The Administrator will have to specify the location of the beacons. • The location needs to be in the system already before it can be assigned to a beacon. • The new beacon must not already exist on the system.
Business rules applicable to this requirement	• Only an Administrator can add a new beacon. • The Administrator must be logged in to add a new beacon.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.3.2 Search Beacon

Functional Requirement	Explanation
Requirement number:	3.2
Requirement name:	Search Beacon
Requirement short description:	The system provides an interface to search for a beacon.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for a beacon through the use of a search function. • The searched beacon must already exist on the system.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can search for a beacon. • The Administrator must be logged in to search for a beacon.
Revision date and Revision number:	2019-02-27 Version 1.0

Criticality/Priority:	Medium

6.3.3.3 Update Beacon

Functional Requirement	Explanation
Requirement number:	3.3
Requirement name:	Update Beacon
Requirement short description:	The system provides an interface to update a Beacon on the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to update a Beacon's information on the system. • The Beacon must exist on the system. • The updated Beacon's information must be valid.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can update a Beacon. • The Administrator must be logged in to update a Beacon.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.3.4 Delete Beacon

Functional Requirement	Explanation
Requirement number:	3.4
Requirement name:	Delete Beacon
Requirement short description:	The system provides an interface to delete a Beacon from the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to delete a Beacon's information from the system. • The Beacon must exist on the system.

	• The delete function must satisfy all referential integrity constraints.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can delete a Beacon and its information. • The Administrator must be logged in to delete a Beacon. • A deleted Beacon's information must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.3.5 Add Beacon status

Functional Requirement	Explanation
Requirement number:	3.5
Requirement name:	Add Beacon status
Requirement short description:	The system provides an interface for adding a new beacon status into the system.
Requirement detailed description and constraints:	• The system will allow the Administrator to add new beacon statuses to the system and its relevant information. • The Administrator will have to specify the states of the beacon statuses. • The new beacon status must not already exist on the system.
Business rules applicable to this requirement	• Only an Administrator can add a new beacon status. • The Administrator must be logged in to add a new beacon status.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.3.6 Search Beacon status

Functional Requirement	Explanation

Requirement number:	3.6
Requirement name:	Search Beacon status
Requirement short description:	The system provides an interface to search for a beacon status.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for a beacon status through the use of a search function. • The searched beacon status must already exist on the system.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can search for a beacon status. • The Administrator must be logged in to search for a beacon status.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.3.7 Update Beacon status

Functional Requirement	Explanation
Requirement number:	3.7
Requirement name:	Update Beacon status
Requirement short description:	The system provides an interface to update a Beacon status on the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to update a Beacon's status information on the system. • The Beacon status must exist on the system. • The updated Beacon status information must be valid.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can update a Beacon status. • The Administrator must be logged in to update a Beacon status.

Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.3.8 Delete Beacon status

Functional Requirement	Explanation
Requirement number:	3.8
Requirement name:	Delete Beacon status
Requirement short description:	The system provides an interface to delete a Beacon status from the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to delete a Beacon status from the system. • The Beacon status must exist on the system. • The delete function must satisfy all referential integrity constraints.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can delete a Beacon status and its information. • The Administrator must be logged in the delete a Beacon status. • A deleted Beacon status must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.3.9 Add Tag

Functional Requirement	Explanation
Requirement number:	3.9
Requirement name:	Add Tag
Requirement short description:	The system provides an interface for adding a new Tag into the system.

Requirement detailed description and constraints:	• The system will allow the Administrator to add a new Tag to the system and its relevant information which includes the Tag ID, the bike which the tag is assigned to and the date which the Tag was assigned. • The Administrator will have to specify which bike the Tag is assigned to • The new Tag must not already exist on the system.
Business rules applicable to this requirement	• Only an Administrator can add a new Tag. • The Administrator must be logged in to add a new Tag.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.3.10 Search Tag

Functional Requirement	Explanation
Requirement number:	3.10
Requirement name:	Search Tag
Requirement short description:	The system provides an interface to search for a Tag.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for a Tag through the use of a search function. • The searched Tag must already exist on the system.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can search for a Tag. • The Administrator must be logged in to search for a Tag. •
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.3.11 Update Tag

Functional Requirement	Explanation
Requirement number:	3.11
Requirement name:	Update Tag
Requirement short description:	The system provides an interface to update a Tag on the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to update a Tag's information on the system. • The Tag must exist on the system. • The updated Tag information must be valid.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can update a Tag. • The Administrator must be logged in to update a Tag.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.3.12 Delete Tag

Functional Requirement	Explanation
Requirement number:	3.12
Requirement name:	Delete Tag
Requirement short description:	The system provides an interface to delete a Tag from the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to delete a Tag's information from the system. • The Tag must exist on the system. • The delete function must satisfy all referential integrity constraints.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can delete a Tag and its information. • The Administrator must be logged in to delete a Tag.

	• A deleted Tag's information must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.4 Sub-system 4: Student

6.3.4.1 Add Student

Functional Requirement	Explanation
Requirement number:	4.1
Requirement name:	Add Student
Requirement short description:	The system provides an interface for the adding of a new student into the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a new Student and their relevant information including Name, Surname, Grade, Age, Gender, etc. • The Administrator will have to specify the school that this student is enrolled at. • Student status must be selected from a list of options. • Student address & Contact details of the student's Parent/Guardian need to be captured. • The student must not already exist in the system. • The Administrator must be logged in to the system to add a new student
Business rules applicable to this requirement	• Only an administrator can add a new student into the system/database. • The student must be currently enrolled at a school under the ERP scope of the #Bikes4ERP project.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.4.2 Search Student

Functional Requirement	Explanation
Requirement number:	4.2
Requirement name:	Search Student
Requirement short description:	The system provides an interface for the search functionality to find an existing student in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for an existing Student and their information will be displayed. • The student being searched needs to be registered in the system. • The administrator must be logged in to the system to search an existing student.
Business rules applicable to this requirement	• Only an administrator can search a student in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.4.3 Update Student

Functional Requirement	Explanation
Requirement number:	4.3
Requirement name:	Update Student
Requirement short description:	The system provides an interface for the updating of the details of a selected student in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to edit/update the details of a particular searched/selected student. • The student being updated needs to be registered in the system and have been searched. • The Administrator must be logged in to update any student.

Business rules applicable to this requirement	• Only the Administrator or authorized functions can edit/update the details of a particular searched/selected student.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.4.4 Delete Student

Functional Requirement	Explanation
Requirement number:	4.4
Requirement name:	Delete Student
Requirement short description:	The system provides an interface for the removal of a selected student and all their details from the system/database.
Requirement detailed description and constraints:	• The system allows the Administrator to delete/remove the details of a particular searched/selected student. • The student being deleted needs to be currently existing and registered in the system and have been searched or selected. • The Administrator must be logged in to delete any student.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can delete the details of a particular searched/selected student. • This is to be rarely utilized since data is to be stored for historical records purposes.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.4.5 Add Student Marks

Functional Requirement	Explanation

Requirement number:	4.5
Requirement name:	Add Student Marks
Requirement short description:	The system provides an interface for the adding of a new student's mark into the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a student's average marks throughout their academic term • The student's marks must correspond to the relevant student • The student must already exist in the system before they can have their marks recorded. • The Administrator must be logged in to the system to add a student's marks.
Business rules applicable to this requirement	• Only an administrator can add a student's marks into the system/database. • The student must be currently enrolled at a school under the ERP scope of the #Bikes4ERP project.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.4.6 Search Student Marks

Functional Requirement	Explanation
Requirement number:	4.6
Requirement name:	Search Student Marks
Requirement short description:	The system provides an interface for the search functionality or administrator to find an existing student's marks in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for an existing Student's marks and their marks will be displayed. • The student's marks being searched needs to exist on the system.

	• The administrator must be logged in to the system to search a student's marks.
Business rules applicable to this requirement	• Only an administrator or an authorized function can search a student's marks in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.4.7 Add Student status

Functional Requirement	Explanation
Requirement number:	4.7
Requirement name:	Add Student Status
Requirement short description:	The system provides an interface for the adding of a student status option that can be chosen in the "Add Student" Use Case.
Requirement detailed description and constraints:	• The system allows the Administrator to add a status option for a student. • The status being added needs to be consistent with the rest of the business rules and constraints of the #Bikes4ERP project and our system. • The Administrator must be logged in to add a student status.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can add a student status.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.4.8 Search Student status

Functional Requirement	Explanation

Requirement number:	4.8
Requirement name:	Search Student Status
Requirement short description:	The system provides an interface for the search functionality to find an existing student status in the system to assign to a student or edit.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for an existing Student status option. • The student status being searched needs to exist in the system prior to the search • The administrator must be logged in to the system to search a student status
Business rules applicable to this requirement	• Only an administrator can search a student status option in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.4.9 Update Student status

Functional Requirement	Explanation
Requirement number:	4.9
Requirement name:	Update Student Status
Requirement short description:	The system provides an interface for the updating of a student status option in the system.
Requirement detailed description and constraints:	• The system will allow the Administrator to edit/update a student status option. • The student status being updated needs to exist in the system prior to this action • The Administrator must be logged in to update any student status.

Business rules applicable to this requirement	• Only the Administrator or authorized functions can edit/update the details of a particular student status option.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.4.10 Delete Student status

Functional Requirement	Explanation
Requirement number:	4.10
Requirement name:	Delete Student status
Requirement short description:	The system provides an interface for the removal of a selected student status option from the system.
Requirement detailed description and constraints:	• The system allows the Administrator to delete/remove a student status option. • The student status option being deleted needs to be currently existing in the system and have been searched or selected. • The Administrator must be logged in to delete any student status option.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can delete a particular searched/selected student status option.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.5 Sub-system 5: Bicycle

6.3.5.1 Add Bicycle

Functional Requirement	Explanation
Requirement number:	5.1

Requirement name:	Add Bicycle
Requirement short description:	The system provides an interface for the adding of a new bicycle into the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a new Bicycle and its relevant information including Model, Color, status, years in use, etc. • The Administrator will have to specify the school that this bicycle will be assigned/dispatched to. • Bicycle status must be selected from a list of options. • The bicycle must not already exist in the system. • The Administrator must be logged in to the system to add a new bicycle.
Business rules applicable to this requirement	• Only an administrator can add a new bicycle into the system/database. • The bicycle must be in a good working condition.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.5.2 Search Bicycle

Functional Requirement	Explanation
Requirement number:	5.2
Requirement name:	Search Bicycle
Requirement short description:	The system provides an interface for the search functionality to find an existing bicycle in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for an existing Bicycle and its information will be displayed. • The bicycle being searched needs to be registered in the system. • The administrator must be logged in to the system to search an existing bicycle

Business rules applicable to this requirement	• Only an administrator can search a bicycle in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.5.3 Update Bicycle

Functional Requirement	Explanation
Requirement number:	5.3
Requirement name:	Update Bicycle
Requirement short description:	The system provides an interface for the updating of the details of a selected bicycle in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to edit/update the details of a particular searched/selected bicycle. • The bicycle being updated needs to be registered in the system and have been searched. • The Administrator must be logged in to update any bicycle.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can edit/update the details of a particular searched/selected bicycle.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.5.4 Delete Bicycle

Functional Requirement	Explanation
Requirement number:	5.4
Requirement name:	Delete Bicycle

Requirement short description:	The system provides an interface for the removal of a selected bicycle and all their details from the system/database.
Requirement detailed description and constraints:	• The system allows the Administrator to delete/remove the details of a particular searched/selected bicycle. • The bicycle being deleted needs to be currently existing and registered in the system and have been searched or selected. • The Administrator must be logged in to delete any bicycle.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can delete the details of a particular searched/selected bicycle. • This is to be rarely utilized since data is to be stored for historical records purposes.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.5.5 Assign Student

Functional Requirement	Explanation
Requirement number:	5.5
Requirement name:	Assign Student
Requirement short description:	The system provides an interface for the assigning of a student to a bicycle.
Requirement detailed description and constraints:	• The system allows the administrator to assign a student to a bicycle. • The student being assigned to a bicycle must be pre-existing in the system. • The bicycle being assigned needs to be in a good working condition. • The bicycle status needs to be updated to accommodate for it being assigned
Business rules applicable to this requirement	• Only the Administrator or authorized functions can assign a student/bicycle or unassign one to a bicycle/student. • The bicycle must have a tag assigned to it already before a student can be assigned to it.

Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.5.6 Add Bicycle status

Functional Requirement	Explanation
Requirement number:	5.6
Requirement name:	Add Bicycle Status
Requirement short description:	The system provides an interface for the adding of a bicycle status option that can be chosen in the "Add Bicycle" Use Case.
Requirement detailed description and constraints:	• The system allows the Administrator to add a status option for a bicycle. • The status being added needs to be consistent with the rest of the business rules and constraints of the #Bikes4ERP project and our system. • The Administrator must be logged in to add a bicycle status.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can add a bicycle status.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.5.7 Search Bicycle status

Functional Requirement	Explanation
Requirement number:	5.7
Requirement name:	Search Bicycle Status

Requirement short description:	The system provides an interface for the search functionality to find an existing bicycle status in the system to assign to a bicycle or edit.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for an existing Bicycle status option. • The Bicycle status being searched needs to exist in the system prior to the search • The administrator must be logged in to the system to search a Bicycle status
Business rules applicable to this requirement	• Only an administrator can search a Bicycle status option in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.5.8 Update Bicycle status

Functional Requirement	Explanation
Requirement number:	5.8
Requirement name:	Update Bicycle Status
Requirement short description:	The system provides an interface for the updating of a bicycle status option in the system.
Requirement detailed description and constraints:	• The system will allow the Administrator to edit/update a bicycle status option. • The bicycle status being updated needs to exist in the system prior to this action • The Administrator must be logged in to update any bicycle status.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can edit/update the details of a particular bicycle status option.
Revision date and Revision number:	2019-02-27 Version 1.0

Criticality/Priority:	Medium

6.3.5.9 Delete Bicycle status

Functional Requirement	Explanation
Requirement number:	5.9
Requirement name:	Delete Bicycle Status
Requirement short description:	The system provides an interface for the removal of a selected bicycle status option from the system.
Requirement detailed description and constraints:	• The system allows the Administrator to delete/remove a bicycle status option. • The bicycle status option being deleted needs to be currently existing in the system and have been searched or selected. • The Administrator must be logged in to delete any bicycle status option.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can delete a particular searched/selected bicycle status option.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.6 Sub-system 6: Bicycle Part

6.3.6.1 Add Bicycle Part

Functional Requirement	Explanation
Requirement number:	6.1
Requirement name:	Add Bicycle Part
Requirement short description:	The system provides an interface for adding a new bicycle part into the system.

Requirement detailed description and constraints:	• The system will allow the Administrator to add new bicycle part to the system and its relevant information. • The Administrator will have to specify the location of the bicycle part on the fault report visual. • The new bicycle part must not already exist on the system.
Business rules applicable to this requirement	• Only an Administrator can add a new bicycle part. • The Administrator must be logged in to add a new bicycle part.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.6.2 Search Bicycle Part

Functional Requirement	Explanation
Requirement number:	6.2
Requirement name:	Search Bicycle part
Requirement short description:	The system provides an interface to search for a bicycle part.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for a bicycle part through the use of a search function. • The searched bicycle part must already exist on the system.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can search for a bicycle part. • The Administrator must be logged in to search for a bicycle part.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.6.3 Update Bicycle Part

Functional Requirement	Explanation

Requirement number:	6.3
Requirement name:	Update Bicycle part
Requirement short description:	The system provides an interface to update a bicycle part on the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to update a bicycle parts information on the system. • The bicycle part must exist on the system. • The updated bicycle parts information must be valid.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can update a bicycle part. • The Administrator must be logged in to update a bicycle part.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.6.4 Delete Bicycle Part

Functional Requirement	Explanation
Requirement number:	6.4
Requirement name:	Delete Bicycle Part
Requirement short description:	The system provides an interface to delete a Bicycle from the system.
Requirement detailed description and constraints:	• The system will allow the Administrator or authorized functions to delete a bicycle parts information from the system. • The bicycle part must exist on the system. • The delete function must satisfy all referential integrity constraints.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can delete a bicycle part and its information.

	• The Administrator must be logged in the delete a bicycle part. • A deleted bicycle parts information must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.7 Sub-system 7: Tracking Log

6.3.7.1 Create Entry

Functional Requirement	Explanation
Requirement number:	7.1
Requirement name:	Create Log Entry
Requirement short description:	The system provides an interface for adding a new entry into the system.
Requirement detailed description and constraints:	• The system will allow the Beacon to add a new log entry to the system and its relevant information which includes: The date, the arrival time, the departure time, the bike ID, the student name and the time traveled. • The bike needs to be in the system already before it can be assigned to an entry. • The new entry must not already exist on the system.
Business rules applicable to this requirement	• An entry needs to be created every time a bike goes past a beacon. • All bikes need to be installed with a tracker.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.7.2 Search Entry

Functional Requirement	Explanation

Requirement number:	7.2
Requirement name:	Search Log Entry
Requirement short description:	The system provides an interface to search for an entry on the system
Requirement detailed description and constraints:	• The system will allow the Administrator to search for an entry by the use of a search function. • The searched entry must already exist on the system.
Business rules applicable to this requirement	• Only an Administrator or an authorized function can search for an Entry. • The Administrator must be logged in to search for an Entry.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.8 Sub-system 8: Maintenance Log

6.3.8.1 Add Fault

Functional Requirement	Explanation
Requirement number:	8.1
Requirement name:	Add Fault
Requirement short description:	The system provides an interface for adding a new fault into the system.
Requirement detailed description and constraints:	• The system will allow a user to add a new fault with the relevant information such as the bike id and detain. • The user will have to specify what is wrong with the bike. • The user will have to specify a bike that is registered to a student.
Business rules applicable to this requirement	• The bike has to be registered on the system.

Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.8.2 Delete Fault

Functional Requirement	Explanation
Requirement number:	8.2
Requirement name:	Delete Fault
Requirement short description:	The system provides an interface to delete a fault from the system.
Requirement detailed description and constraints:	• The system will allow an authorized user to delete a fault on the system. • The fault must exist on the system. • The delete fault must satisfy all referential integrity constraints. • The user must specify the reason for deleting the fault.
Business rules applicable to this requirement	• Only an authorized user can delete a fault. • The user must be logged into the system. • A deleted fault's information must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.8.3 Search Fault

Functional Requirement	Explanation
Requirement number:	8.3
Requirement name:	Search Fault
Requirement short description:	The system must have functionality to search for a fault with specific parameters.

Requirement detailed description and constraints:	• The system will allow an authorized user to search for a fault on the system using a search function. • The user must provide valid information for the search function.
Business rules applicable to this requirement	• Only authorized functions or users can use the search fault function. • The user must be logged in to search for a fault.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.8.4 Add Maintenance task

Functional Requirement	Explanation
Requirement number:	8.4
Requirement name:	Add Maintenance Task
Requirement short description:	The system provides an interface for adding a new Maintenance Task into the system.
Requirement detailed description and constraints:	• The system will allow an authorized user to add a new Maintenance Task. • The authorized user will have to specify the Maintenance Task's information such as the name, time span and requirements of the new Maintenance Task. • The new Maintenance Task must not already exist on the system.
Business rules applicable to this requirement	• Only an authorized user can add a new Maintenance Task. • The user must be registered in the system. • The Administrator must be logged in to add a Maintenance Task.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.8.5 Update Maintenance task

Functional Requirement	Explanation
Requirement number:	8.5
Requirement name:	Update Maintenance Task
Requirement short description:	The system provides an interface for updating a Maintenance Task on the system.
Requirement detailed description and constraints:	• The system will allow an authorized user to update a Maintenance Task. • The authorized user will have to specify the new information for the specified Maintenance Task. • The Maintenance Task must already exist on the system. • The new information must be valid.
Business rules applicable to this requirement	• Only an authorized user can update a Maintenance Task. • The user must be registered in the system. • The User must be logged in to update a Maintenance Task.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.8.6 Search Maintenance task

Functional Requirement	Explanation
Requirement number:	8.6
Requirement name:	Search Maintenance Task
Requirement short description:	The system must have functionality to search for a Maintenance Task with specific parameters.
Requirement detailed description and constraints:	• The system will allow an authorized user to search for a Maintenance Task on the system using a search function. • The user must provide valid information for the search function.

Business rules applicable to this requirement	• Only authorized functions can use the search Maintenance Task. • The user must be logged in to search for a Maintenance Task.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.8.7 Delete Maintenance task

Functional Requirement	Explanation
Requirement number:	8.7
Requirement name:	Delete Maintenance Task
Requirement short description:	The system provides an interface to delete a Maintenance Task from the system.
Requirement detailed description and constraints:	• The system will allow an authorized user to delete a Maintenance Task on the system. • The Maintenance Task must exist on the system. • The delete function must satisfy all referential integrity constraints. • The user must specify the reason for deleting the Maintenance Task.
Business rules applicable to this requirement	• Only an authorized user can delete a Maintenance Task. • The user must be logged into the system. • A deleted Maintenance Task's information must still be backed up.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.9 Sub-system 9: Mechanic

6.3.9.1 Add Mechanic

Functional Requirement	Explanation

Requirement number:	9.1
Requirement name:	Add Mechanic
Requirement short description:	The system provides an interface for the adding of a new mechanic into the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a new Mechanic and their relevant information including Name, Surname, Cellphone, Email, Gender, ID Number etc. • Mechanic ID number & Contact details of the mechanic need to be captured. • The Mechanic must not already exist in the system. • The Administrator must be logged in to the system to add a new Mechanic
Business rules applicable to this requirement	• Only an administrator can add a new mechanic into the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.9.2 Search Mechanic

Functional Requirement	Explanation
Requirement number:	9.2
Requirement name:	Search Mechanic
Requirement short description:	The system provides an interface for the search functionality to find an existing mechanic in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to search for an existing Mechanic and its information will be displayed. • The mechanic being searched needs to be registered in the system. • The administrator must be logged in to the system to search an existing mechanic.

Business rules applicable to this requirement	• Only an administrator can search a mechanic in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.9.3 Update Mechanic

Functional Requirement	Explanation
Requirement number:	9.3
Requirement name:	Update Mechanic
Requirement short description:	The system provides an interface for the updating of the details of a selected mechanic in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to edit/update the details of a particular searched/selected mechanic. • The mechanic being updated needs to be registered in the system and have been searched. • The Administrator must be logged in to update any mechanic.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can edit/update the details of a particular searched/selected mechanic.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.9.4 Delete Mechanic

Functional Requirement	Explanation
Requirement number:	9.4
Requirement name:	Delete Mechanic

Requirement short description:	The system provides an interface for the removal of a selected mechanic and all their details from the system/database.
Requirement detailed description and constraints:	• The system allows the Administrator to delete/remove the details of a particular searched/selected mechanic. • The mechanic being deleted needs to be currently existing and registered in the system and have been searched or selected. • The Administrator must be logged in to delete any mechanic.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can delete the details of a particular searched/selected mechanic. • This is to be rarely utilized since data is to be stored for historical records purposes.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.9.5 Assign Mechanic

Functional Requirement	Explanation
Requirement number:	9.5
Requirement name:	Assign Mechanic
Requirement short description:	The system provides an interface for the assigning of a Mechanic to a School.
Requirement detailed description and constraints:	• The system allows the administrator to assign a mechanic to a school. • The mechanic being assigned to a school must be pre-existing in the system.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can assign a mechanic or unassign a mechanic to a school. • The bicycle must have a tag assigned to it already before a student can be assigned to it.

Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.10 Sub-system 10: School

6.3.10.1 Add School

Functional Requirement	Explanation
Requirement number:	10.1
Requirement name:	Add School
Requirement short description:	The system provides an interface for the adding of a new school into the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to add a new School and its relevant information including name and location. • The School must not already exist in the system. • The Administrator must be logged in to the system to add a new School
Business rules applicable to this requirement	• Only an administrator can add a new school into the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.10.2 Search School

Functional Requirement	Explanation
Requirement number:	10.2
Requirement name:	Search School
Requirement short description:	The system provides an interface for the search functionality to find an existing school in the system/database.

Requirement detailed description and constraints:	• The system will allow the Administrator to search for an existing school and its information will be displayed. • The school being searched needs to be registered in the system. • The administrator must be logged in to the system to search an existing school.
Business rules applicable to this requirement	• Only an administrator can search a school in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.10.3 Update School

Functional Requirement	Explanation
Requirement number:	10.3
Requirement name:	Update School
Requirement short description:	The system provides an interface for the updating of the details of a selected school in the system/database.
Requirement detailed description and constraints:	• The system will allow the Administrator to edit/update the details of a particular searched/selected school. • The school being updated needs to be registered in the system and have been searched. • The Administrator must be logged in to update any school.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can edit/update the details of a particular searched/selected school.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.10.4 Delete School

Functional Requirement	Explanation
Requirement number:	10.4
Requirement name:	Delete School
Requirement short description:	The system provides an interface for the removal of a selected school and all their details from the system/database.
Requirement detailed description and constraints:	• The system allows the Administrator to delete/remove the details of a particular searched/selected school. • The school being deleted needs to be currently existing and registered in the system and have been searched or selected. • The Administrator must be logged in to delete any school.
Business rules applicable to this requirement	• Only the Administrator or authorized functions can delete the details of a particular searched/selected school. • This is to be rarely utilized since data is to be stored for historical records purposes.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.11 Sub-system 11: Job

6.3.11.1 Add Job

Functional Requirement	Explanation
Requirement number:	11.1
Requirement name:	Add Job
Requirement short description:	The system provides an interface for the adding of a new job into the system/database when the students make a request.
Requirement detailed description and constraints:	• The system will allow the authorized function to add a new job and its relevant information including Date created, Bike ID, Student ID, Mechanic assigned, Job status. • The problem with the bike needs to be captured. • The Job must not already exist in the system.

	• The function invoking "Add job" must be authorized to do so.
Business rules applicable to this requirement	• Only the authorized function can add a new job into the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.11.2 View Job

Functional Requirement	Explanation
Requirement number:	9.2
Requirement name:	View Job
Requirement short description:	The system provides an interface for the view functionality to find an existing job in the system/database and display it.
Requirement detailed description and constraints:	• The system will allow the Mechanic to search for an existing job and its information will be displayed. • The job being searched needs to exist in the system. • The Mechanic must be logged in to the system to view an existing Job.
Business rules applicable to this requirement	• Only a Mechanic or authorized function can view a mechanic in the system/database.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.12 Sub-system 12: Reporting

6.3.12.1 Generate Maintenance schedule

Functional Requirement	Explanation

Requirement number:	12.1
Requirement name:	Generate weekly Maintenance schedule
Requirement short description:	The system must generate a maintenance schedule automatically every week.
Requirement detailed description and constraints:	• The system must automatically generate a maintenance schedule every Friday. • The maintenance schedule will include the all the student's maintenance requests scheduled for the upcoming week.
Business rules applicable to this requirement	• The maintenance schedule must be generated in an easy to read and use format as the Mechanic (primary User of the report) could potentially not be highly and may have limited technological skills. • The report must be generated weekly and automatically.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.12.2 Generate Maintenance Requirements Report

Functional Requirement	Explanation
Requirement number:	12.2
Requirement name:	Generate Maintenance Requirements report
Requirement short description:	The system must automatically generate a Maintenance Requirements report every week.
Requirement detailed description and constraints:	• The system must automatically generate a Maintenance requirement report every Friday. • The Maintenance requirement report will include the all the bicycle parts and maintenance tools needed for the bicycle repair requests scheduled for the upcoming week.
Business rules applicable to this requirement	• The Maintenance requirement report must be generated in an easy to read and use format as the Mechanic (primary User of the report) could potentially not be highly and may have limited technological skills.

	• The report must be generated weekly and automatically.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.12.3 Generate Student Report

Functional Requirement	Explanation
Requirement number:	12.3
Requirement name:	Generate Student Report
Requirement short description:	The system must have the functionality for the Administrator to generate a student report.
Requirement detailed description and constraints:	• The Administrator must be able to generate a Student report using the system based on given criteria. • The system will retrieve the student report information from the database and display in a report view as well as have the option to export the report.
Business rules applicable to this requirement	• Student report must be generated half yearly as an overall average of the student academic performance. • Only Administrator must be able to generate the Student report.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.12.4 Generate Bicycle History report

Functional Requirement	Explanation
Requirement number:	12.4
Requirement name:	Generate Bicycle History Report

Requirement short description:	The system must generate a bicycle history report on the Administrators request.
Requirement detailed description and constraints:	• The system must have the functionality such that the Administrator can generate a report displaying each bicycle registered on the system and relevant information. • The summarized report will include maintenance history, ownership history.
Business rules applicable to this requirement	• Only the Administrator must be able to generate the report.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Low

6.3.12.5 Generate Log report

Functional Requirement	Explanation
Requirement number:	12.5
Requirement name:	Generate Log report
Requirement short description:	The system must generate a Log schedule on the Administrators request.
Requirement detailed description and constraints:	• The system must have the functionality such that the Administrator can generate a beacon Log report based on given criteria. • The summarized report will include each beacons arrival and departure times for each student.
Business rules applicable to this requirement	• Only the Administrator must be able to generate the report.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	High

6.3.13 Sub-system 13: Backup and Restore

6.3.13.1 Backup Data

Functional Requirement	Explanation
Requirement number:	13.1
Requirement name:	Backup Data
Requirement short description:	The system must backup all the data in the system on a regular basis.
Requirement detailed description and constraints:	• The system must have the functionality to duplicate all the data to external servers. • Ensure referential integrity of the data. • The administrator should be able to initiate a backup
Business rules applicable to this requirement	• The data must be backed up weekly.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.3.13.2 Restore Data

Functional Requirement	Explanation
Requirement number:	13.2
Requirement name:	Restore Data
Requirement short description:	The system must restore all the data from the most recent backup that appears in the system.
Requirement detailed description and constraints:	• The system must have the functionality to override the local data with the most recent version on the external backup. • The system will allow the administrator to initiate the restore process.

Business rules applicable to this requirement	• Only an authorized user can restore the data. • The user must be logged into the system.
Revision date and Revision number:	2019-02-27 Version 1.0
Criticality/Priority:	Medium

6.4 User Acceptance Criteria

The user acceptance criteria allow us to define the boundaries of the system requirements. This helps the system designers to prioritize the most critical aspects of the system. Ensuring that the requirements are implemented completely so that the system can serve to complete its purpose.
The following are the priority indicators uses for the individual requirements.

- High: These are the requirements that are critical for the system to function and serve its intended purpose. These requirements are essential for the system to run and require the system designers to implement it completely.
- Moderate: These are the requirements that increase the efficiency of our system but do not prevent our system from running if they are not there.
- Low: These are the requirements that are optional, this means that the system can still run and serve its intended purpose if these requirements are not implemented.

6.5 Non-functional Requirements

6.5.1 Fast response times

Requirement	Explanation
Number and Title:	1. Fast response times
Text:	The system will have fast response times.
Type:	Non-Functional
Details and Constraints:	• System will always respond quickly when not affected by external factors. • System must make use of the least amount of computer resources as possible.
Version Date & Number:	24 February 2020 | Version 1.0
Criticality/Priority:	High

6.5.2 User-friendly

Requirement	Explanation
Number and Title:	2. User-Friendly design
Text:	The system must have a simple and easy to use design so that it can be used by users of all ages.
Type:	Non-Functional
Details and Constraints:	• The system must be designed in such a way that it is easy for the user to use. • The system must have help functionality.
Version Date & Number:	24 February 2020 | Version 1.0
Criticality/Priority:	High

6.5.3 Data integrity and consistency

Requirement	Explanation
Number and Title:	3. Data integrity and consistency
Text:	The system database must maintain its database integrity in a consistent manner.
Type:	Non-Functional
Details and Constraints:	• The system and database must be designed in such a way that all data maintains its integrity. • The system and database must be designed in such a way that all data maintains consistent.
Version Date & Number:	24 February 2020 | Version 1.0
Criticality/Priority:	High

6.5.4 Easy maintained

Requirement	Explanation

Number and Title:	4. Easily maintained
Text:	The system must be designed so that maintenance is quick and doesn't prevent the system from serving its purpose.
Type:	Non-Functional
Details and Constraints:	• The system will be designed in segments so that the system will not lose functionality when different segments are under maintenance.
Version Date & Number:	24 February 2020 | Version 1.0
Criticality/Priority:	Low

6.5.5 System flexibility

Requirement	Explanation
Number and Title:	5. System Flexibility
Text:	The system must not have fixed variables and rules.
Type:	Non-Functional
Details and Constraints:	• The system must store any variables or rules that will restrict a function in a table. • These variables must be updateable so that it can be changed accordingly.
Version Date & Number:	24 February 2020 | Version 1.0
Criticality/Priority:	High

6.5.6 Security protocols

Requirement	Explanation
Number and Title:	6. Strict security protocols
Text:	Ensure that strict security protocols are used and implemented to ensure he systems security.
Type:	Non-Functional
Details and Constraints:	• The database must be encrypted ensuring data security.

	• Different users must have different restrictions to prevent unauthorized use of different functions.
Version Date & Number:	24 February 2020 | Version 1.0
Criticality/Priority:	High

6.5.7 Easily accessible

Requirement	Explanation
Number and Title:	7. Easily accessible
Text:	The system must be easy to access.
Type:	Non-Functional
Details and Constraints:	• The system will be web based so that it can be accessed from anywhere. • The database will be hosted online to allow data access from anywhere.
Version Date & Number:	24 February 2020 | Version 1.0
Criticality/Priority:	High

6.6 Conclusion

In conclusion, the requirements analysis section represents the setup and analysis of the functional and non-functional requirements of the system as well as user acceptance criteria.

7. Feasibility/Decision Analysis

7.1.1 Introduction

The feasibility analysis section deals with the new system proposal to the client by comparing and analyzing different potential system candidates. Alternatives and recommendations are also discussed. The following is applicable:

Purpose: The purpose of this analysis is to help determine which of the four options stated will be the most viable option for the business, this is done by assessing the Operational analysis deals with the functionality of the system as well as the political side of the system. The technical analysis deals with the various technologies used as well as the expertise needed to develop, operate and maintain the system. The economic analysis deals with the cost to develop the system as well as the hardware and software required to operate the system. The Schedule analysis deals with the time the system will take to be developed and implemented.

Background: There is currently no Bikes4Erp system as there is currently no way to track the bicycles, no data backup.

Scope: The feasibility analysis will analyze 4 different systems. Two of the systems will be Virtual Visionaries prosed systems and the other two systems are already developed and available systems.

Structure: It analyzes each candidate on 4 criteria's that each have a specific weighting percentage which helps give a score for each candidate on each criterion as well as an overall score.

Weightings: When discussing the weightings, we discussed 3 different options and came to the following conclusion. operational feasibility has a weighting of 40% as it is the most important aspect of the required system as the client will need every stated functionality done in order for the system to operate at an optimal manner. Technical feasibility has a weighting of 30% as there are many technical factors which include signal and service as well as the teams coding experience, these and more will have to be considered as they will have a major impact on the running and success of the system. Economical feasibility has a weighting of 20% as the client informed us that cost was not a major concern leading us to choose 20% as it isn't as important as Operational and technical feasibility but still has an important role as to high cost will have a negative impact on the system. Schedule feasibility has a weighting of 10% since the time frame given was acceptable and fair making it a less key factor than the other feasibilities.

7.2 Feasibility Analysis/Matrix

Feasibility Criteria	Weight	Visybl	AssetGather/ Booqable	VVS (RIFD)	VVS (BLE beacons)

Operational Feasibility Functionality A description of to what degree the candidate would benefit the organization and how well the system would work.	40%	- Visybl is an existing system that is already available for hire - It benefits the client as it has a user-friendly website and app for tracking your data as well as it already has existing tracking reports and interface to add devices. - Visybl already has all functionality with tracking and generating reports on that data	- AssetGather/ Booqable are two already existing systems that are independent of each other. - The AssetGather system catches all the data from the tag reader and stores it in a readable format to allow the reader to view the data - The AssetGather system can generate reports based on the tracking data. - The system is designed for growth making it easy to expand the tracking system. - The system will also use Booqable to handle the booking and scheduling for the mechanic.	- The system will be able to create the reports required for the admin to monitor the students and their bicycles as well as maintenance and repair reports for the maintenance staff. - The system will also have a feature to allow students to book a bicycle repair. - The system will be able to notify the school and parent if the child did not arrive at school. - The system will be able to log and keep track of the times the bicycles depart and arrive at school as well as the amount of time spent on the bicycles. - The system will be linked to a cloud database that is expandable.	- The system will be able to create the reports required for the admin to monitor the students and bike as well maintenance and inventory reporting for the maintenance team. - The system will also have a feature to allow students to book a bike repair. - The system will be able to notify the school and parent if the child did not arrive at school. - The system will be able to log and keep track of the times the bikes leave and arrive at school as well as the time spent on the bike. - The system will be linked to a cloud database that is expandable. - The system will consist of a web

			- Booqable will allow the mechanic to view and plan his current schedule. - As well as allow students to book a bicycle repair. - The users won't use all the functionality provided by AssetGather as it is not designed to track other types of data. - Some of the functionality is missing in AssetGather this is where Booqable comes in to fill the maintenance booking requirements. - But Booqable will have a lot of unused functions.	- The system will consist of a web application that will include all the required functionality regarding the admin, the mechanic and the student. - The system will generate the various reports that will be required by the admin and the mechanic.	application for admin and the mechanic usage to allow them to generate necessary reports to track bike usage, student progress and maintenance schedules
Political A description of how well received this solution would be		- The solution will be received well by admin as it is a simple easy use	- The system will be received in an okay manner as you need two different	- The mechanic and Admin will easily adapt to the system as it was	- The mechanic and Admin will easily adapt to the system as it was

from both user management, user, and organization perspective.		process to get reports of all the data - The system is Missing key functionality making it difficult to keep track of all data on separate systems. - Overall the system won't be accepted well as it doesn't have all the desired functions and will still require a lot of manual tasks. Score: 20/40	programs to do majority of the functions. - Some Key functionality is missing as a result some functions will still be a manual process. Score: 30/40	developed with their stated required functionality. - The student will receive the system well as it was developed to ensure it is easy and user friendly for them to log a repair - Quick tutorial may be required for the mechanic to use the system. - The overall system should be accepted well as it possesses all the stated functionality. Score: 38/40	developed with their needs in mind to ensure easy usability and user-friendly layout. - The student will receive the system well as it was developed to ensure it is easy and user friendly to use. - Quick tutorial may be required for the mechanic to use the system. Score: 37/40
Technical Feasibility	30%	- Requires a standard	- A stable internet	- The system will be	- The system will use

Technology An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed to support this candidate.	computer as well as a smartphone that supports BLE services. - A stable Internet connection is required to interact with the system as the whole system is online - All you need to purchase is the Asset beacons and cloud nodes. - Visybl use desired technology that can be integrated with many systems. - Visybl is an online system that is already developed with the user in mind to ensure their website is user friendly. - It uses high quality BLE beacons called cloud	connection is required to interact with the system. - Requires RFID Tags and readers to be purchased - This candidate uses technology that is desirable and easy to implement - Will need a computer that's strong enough to run two programs simultaneously . - AssetGather/ booqable is an already existing system that is easy to use and designed for ease of access.	integrating mobitech's CUHk software which is freely available as well as using googles web API to perform various tracking functions. These functions are easily obtainable as well as easy to integrate. - As well as making use of RFID sensors and tags which are easily obtainable. - The system will be built for web use by using available technologies which include: Visual Studio Code and SQL Server Manager. - This application will be available online and will require internet connection to perform various functions on the system. - The system will be able to run on any	googles API to integrate BLE Beacons into our system - The system will be built for web using available technology like: Visual Studio code and SQL Server manager. - The system will integrate googles beacon technology called Eddystone. - This system will incorporate BLE Beacon technology to keep track of the times the bikes arrive. - This application will be fully web based and will require an internet connection to perform various functions on the system.

		nodes and asset beacons that will be attached to the bike. - Visybl also has an app that can be downloaded on android and IOS to help track your assets.		device that can support web browsing. - This system will require additional hardware for each bike in the form of a tag.	- The system will be able to run on any device that supports web browsing - The system will require additional devices for the bike as well as devices to transmit the beacon data.
Expertise An assessment to the technical expertise needed to develop, operate, and maintain the candidate system.		- Visybl will be well accepted by the user as it is an easy to navigate site that can be accessed through the website or the mobile app as long as there is internet connection making it easy and convenient for the administrator to interact	- Maintenance of the system can be done by AssetGather/Booqable. - No development is needed as you pay a subscription for the service. - Not too complicated to integrate Booqable and AssetGather. - Users of the system may require a few tutorials to be	- Knowledge of C# and mobitech's CUHk software will be needed in the development and maintenance of the system. - The system will be user friendly with basic computer literacy needed to interact with the system	- C# and Eddystone knowledge is required for the development and maintenance of the system. - The system will be user friendly with basic computer literacy needed to interact with the system.

		with the system. - The organization will accept it highly as it is an existing and tested system that is guaranteed to benefit the organization Score: 19/30	able to operate both systems simultaneously. Score: 20/30	Score: 27/30	Score: 25/30
Economic Feasibility Cost to develop:	20%	- No development cost	- No development cost	- Free Software: Firebase Visual studio Visual studio code - Paid Software: SQL server manager - R12000 - TOTAL development cost: R12000	- Free Software: FireBase Visual studio Visual studio code Eddystone - Paid Software: SQL server manager - R12000 - TOTAL development cost: R12000
Hardware Cost		R4683 for 6 beacons and one cloud node	Tags and readers R16 per tag	RFID Tags = R16 per tag R988 per RFID antenna	BLE Beacon = R502 BLE Sensor = R472

Subscription Fee		R60 per bike per month Score: 14/20	R988 per antenna AssetGather = R2890 Booqable = R3780 Total Rental cost = R6670 Score: 10/20	- No Subscription Fee Score: 17/20	- No Subscription Fee Score: 15/20
Schedule Feasibility An assessment of how long the solution will take to design and implement.	10%	- This system will take 2-5 months to implement as well as train necessary users of the system. Score: 10/10	- The system will take 3-6 moths to implement as well as train all necessary users of the system. Score: 9/10	- The system will take around 9 months for Virtual Visionaries to design and implement as well as train the necessary users of the system. Score: 8/10	- The system will take around 9 months for Virtual Visionaries to design implement as well as train the necessary users of the system. Score: 8/10
Ranking:	100%	63%	69%	90%	85%

Table 6: Feasibility Analysis Matrix

7.3 Details of Alternatives

CANDIDATE 1 – VISYBL

Feasibility	Details
Operational	Visybl is a tracking managing system, Visybl allows you to track your assets as well as generating various reports. Visybl lacks key functionality regarding the mechanic and student side of the system and will require another separate system or manual functions.
Technical	Visybl is an already developed and available for purchase which makes it quick and easy to obtain the system. This system is compatible with IOS, Android and windows. This system requires a smart device in order to pick up and transfer beacon data. Visybl is an easy to use system that requires basic computer literacy. Visybl does not provide all desired functionality as well as requiring a smart device. This system will be used in rural areas were smart devices may not be located.
Economical	Visybl is available on a monthly subscription of R60 per month for each bicycle. The R60 includes a BLE Beacon.

Schedule	Visybl is an already developed system therefore there is no development needed only time needed is for setup and training therefore making it a quick installation and setup.

Table 7: Alternatives Candidate 1

CANDIDATE 2 – ASSETGATHER/BOOQABLE

Feasibility	Details
Operational	AssetGather is an asset tracking system that was created by AssetPulse is designed to manage RFID tag tracking data. AssetGather allows you to track the time of arrival and departure of each bicycle and generate reports based on the gathered data. AssetGather lacks key functionality but that is where Booqable comes in as Booqable will be able to create a schedule for the mechanic as well as allow a student to add a repair.
Technical	AssetGather and Booqable are both already available systems with AssetGather being compatible with Booqable making it quick and easy to obtain the two systems. These systems are web based and will require internet connection and a browser. This system uses RFID technology to track the departure and arrival times of the bicycles. These systems are easy to use but may require some knowledge to operate both simultaneously, but all functionality will require basic computer literacy to operate.
Economical	AssetGather and Booqable are available on a monthly subscription plan of R2890 and R3780 this is only for the software you will need to purchase RFID antennas and tags which cost R988 per antenna and R16 per tag.
Schedule	AssetGather and Booqable are both already developed systems therefore there is no development needed only time needed is for setup and training therefore making it a quick installation and setup.

Table 8: Alternatives Candidate 2

CANDIDATE 3 – VVS (RFID)

Feasibility	Details
Operational	VVS is a system that will be designed from scratch, VVS will be tailor-made to the needs of the client to ensure all requirements listed by the client will be fulfilled. VVS will use a web application which will allow the admin to generate various reports on student performance as well as arrival and departure times and overall bicycle usage time. The system will be able to notify the school and parents if the child has not arrived at school. The system will also generate a maintenance schedule for the mechanic as well as allow him to add repair as well as generate a maintenance report. This system will also allow the student to book a repair slot for his bicycle.
Technical	VVS will require a basic Computer at the school to allow the mechanic to operate the system as well as the students to book a repair. The system will require an internet connection to use the system as well as to update the database. VVS will require additional software to read the RFID tags as well as to develop the system, the system will require each bike to have an RFID tag as well as two locations that will have antennas installed. VVS will have a

	simple and easy to understand user interface. The system will require basic knowledge of C# and Mobitech's CUHk software.
Economical	This system uses a lot of free software except for yearly subscription for SQL Server Manager. VVS will also require RFID tags for each bicycle as well as antennas in the school areas.
Schedule	This system will be designed from scratch therefore requiring a lot of development and setup time as well as a brief time to train the users of the system. Therefore, taking the team around 9 months to complete the system.

Table 9: Alternatives Candidate 3

CANDIDATE 4 – VVS (BLE)

Feasibility	Details
Operational	VVS is a system that will be designed from scratch, VVS will be tailor-made to the needs of the client to ensure all requirements listed by the client will be fulfilled. VVS will allow the admin to generate various reports on student performance as well as arrival times to school and overall bicycle usage time. The system will also generate a maintenance schedule for the mechanic as well as allow him to add repair as well as generate a maintenance report. This system will also allow the student to book a repair slot for his bicycle.
Technical	VVS will require a smart device at the school to allow the mechanic to operate the system as well as the students to book a repair as well as to read the BLE beacon data. The system will require an internet connection to use the system as well as to update the database. VVS will require additional software to read the BLE beacons as well as to develop the system, the system will require each bike to have a BLE beacon as well as a smart Device at the school and a beacon by a point in the village. This system will be used in rural areas were smart devices may not be located. VVS will have a simple and easy to understand user interface. The system will require basic C# and Eddystone knowledge.
Economical	This system uses a lot of free software except for yearly subscription for SQL Server Manager. VVS will also require BLE beacons for each bicycle as well as a beacon in the village and a smart device at the school.
Schedule	This system will be designed from scratch therefore requiring a lot of development and setup time as well as a brief time to train the users of the system. Therefore, taking the team around 9 months to complete the system.

Table 10: Alternatives Candidate 4

7.4 Recommendations

CANDIDATE 3 – VVS (RFID)

VVS is a fully customized system that was tailor made to fit the requirements stated by the client. VVS will be designed from scratch to ensure that all the requirements listed by the client will be met in an optimal manner to ensure that the system runs smoothly and effectively. VVS includes functionality that is desirable for the student, mechanic and Administrator. VVS had the highest score for functionality as it meets the

requirements listed whereas the other candidates lack in key areas making them less viable than VVS. VVS has very appealing technology as it is easily obtainable as well as how well it will integrate into our system than the other options. VVS will only require one annual subscription for Microsoft SQL server manager as well as a once off purchase of RFID tracking technology. VVS scheduled time frame satisfies the time frame stipulated by the client (schedule feasibility) making it the most feasible option as it had the highest score out of all the candidates.

7.5 Conclusion

After performing our feasibility analysis, we have gathered that the best option would be candidate 3 overall as it provides the best balance of all the feasibility factors. Candidate 3 is not only more feasible, but it is better than the other candidates in important aspects which will help us create the best system possible to meet our client's problem.

8. Deliverable Conclusion

The objective of deliverable 1 was to figure out what the clients wanted from us compared to what they currently have. We analyzed their current system together with them to identify key features that they're missing such as a lack of tracking capabilities and found solutions to each of these problems through the use of multiple elicitation techniques and research.

After a rough idea of the required system came into being, we started refining it and focusing on more individual components of it so that it is easier for us to expand on in further deliverables. This deliverable has served as a basis on what our system will look like and do but we know that we will have changes in this deliverable as our system will change and different components and aspects will be added and removed.

In summary this deliverable is essentially a way for our client and us to gain clarity into what the new system's features and capabilities will be as well as the problems we need to address and how to solve these problems.

1. Appendices

1.1 Appendix A: Deliverable Sign-off

1.1.1 Client Sign-off

I, _____, on behalf of EPI-USE, hereby confirm that I understand and agree to the stated scope of work, specifications and requirements set out in this document – Visual Visionaries Deliverable 1 Project Proposal.

Date: _____

Signature: _____

1.1.2 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 1 Project Proposal for INF 370 2020.

Name: _____

Signature: _____

Date: _____

Name: _____

Signature: _____

Date: _____

Name: _____

Signature: _____

Date: _____

Name: _____

Signature: _____

Date: _____

Name: _____

Signature: _____

Date: _____

1.2 Appendix B: Client Documentation

1.2.1 ERP Facts

100% OF YOUR DONATION GOES DIRECTLY TO ERP'S FIELDWORK

ERP's parent company, groupelephant.com, donates 1% of its revenues to cover all of ERP's operational expenses, and to fund projects. This means that 100% of every dollar you donate will go straight into our fieldwork projects.

ERP WORKS DIRECTLY WITH THE COMMUNITY

ERP empowers communities that live side by side with wild elephants and rhinos because we believe community members are the only viable long-term solution to the survival of elephants and rhinos.

WITHIN REASON ERP WILL TRY ANYTHING THAT MAY WORK

We have an appetite for well-managed risk, so we pursue a broad range of potential solutions and rapidly scale the ones that work.

ERP IS RUN WITH THE DISCIPLINE OF A BUSINESS, RATHER THAN LIKE A CONVENTIONAL NONPROFIT

Drawing on the resources in groupelephant.com, we have expertise in building partnerships, executing projects and growing economic wealth, and we apply these skills to the challenge of poverty in rural communities, with reference to elephant and rhino conservation.

ERP BUILDS UNCONVENTIONAL PARTNERSHIPS

We have no preconceived notions about which players should be involved in this fight, so we seek out partners who are best positioned to help, whether they are businesses, other conservation organizations, governments, or local communities.

ERP PROVES EVERY PROJECT

By measuring basic impact numbers, providing project details and photos, ERP strives to prove every field project it completes.

Dedicated to protecting elephants and rhinos in the wild through alleviating poverty
ERP is a 501(c)(3) | EIN 81-1973249

1.2.2 ERP Overview

1.2.3 ERP #Bikes4ERP Document – Project Request

February 10, 2020

EPI-USE Office
46 Ingersol Road,
Lynnwood Glen,
Pretoria,
0081

Dear Student,

#Bikes4ERP Tracking Initiative

ERP's strategy is the protection and preservation of elephants and rhinos in the wild, through poverty alleviation. Our poverty alleviation methodology covers Planning, Education, Agriculture, Cooperatives and Environment (P.E.A.C.E.).

ERP brings over 20 years of experience in community relations, poverty alleviation, and impact investment in infrastructure, to achieve sustainable, non-lethal wildlife conservation.

#Bikes4ERP wants to leverage on this by introducing a method of tracking their ERP bikes handed to children in rural committees.

As further background find the attached flyer highlighting what ERP represents.

Yours sincerely,
Manie Kilian

February 10, 2020

Problem Statement

School children in rural schools often walk distances of up to 16 kilometers to get to and from school daily. This leads to fatigue which affects their concentration levels in class and their overall results and performance. This situation often results in absenteeism and in some instances complete withdrawal from school

Applying Technology

Through #Bikes4ERP, we raise funds to purchase and distribute bicycles to students, in remote rural areas, for children who must walk more than 10 kms, daily to and from school, to improve their daily attendance at school, thereby improving their educational achievements.
Students enter into a 'Study to Own' contract where after 2 years of using the bicycles to go to school, they own it. The short-term impact of this is that we address exhaustion and absenteeism, whilst the long-term impact is enabling the students to finish their education thereby reducing the likelihood of extreme poverty in rural areas.

For this 2-year period we want to deploy technology to track the children's progress to and from school by placing markers along the routes to school. Once they arrive or leave school again their caretakers or parent must be informed that they are on route and be able to monitor their progress. We have challenges with cellphone coverage and data availability. These needs to be considered in the solution.

In addition, we also want to track the time they spend on the bicycles for this period in order to monitor and track our maintenance schedules on the bikes. These info needs to be recorded and available for analysis.

Sessions with the applicable parties involved will be arranged to fully scope the various requirements and desired outcomes.

Conclusion

Groupelephant.com continuously strives to improve on the current processes we have and to deploy new initiatives and would like to pair this with whatever technology can make the journey for our #Bikes4ERP children as pleasant and efficient as possible. To this end, any meaningful improvements and innovations not detailed above will be encouraged. We look forward to your participation in our cause!

1.2.4 EPI-USE System Documentation: Beneficiary Contract

BENEFICIARY CONTRACT
#BIKES4ERP PROGRAM

Personal Information and Beneficiary Baseline **Please complete all sections honest and in full. NO SPACES**				
Name and Surname of Student				
What is your Age				
What is your Grade and Class?			*Example Gr 8 A/Gr9 B/Gr 10 C*	
What is your Gender	1.Male		2.Female	
Name of School you attend				
Name of your village?				
What is your average travel time going to school?		Hours		Minutes
What is the distance between your home and school?				KM
How do you travel to school?	1.Walking	2.Using a bus/taxi	3.Using a bicycle	4.Using a Car

Attendance (Please provide us with the most relevant answer)		
On average how many days per month are you absent from school?		Average Number of Days per Month
What is the primary reason for your absence from school?	1. I walk a long distance to school 2. I have no transport to school 3. Unreliable transport to school 4. Transport to school is to expensive 5. I never miss school 6. Other Please specify _____	Answer:(Write only one number)
Punctuality (Please provide us with the most relevant answer)		
On average how many days per month are you late for school?		Average Number of Days Per month
What is the primary reason for you coming late to school?	1. I have to walk a long distance to school 2. I have no transport to school 3. Unreliable transport to school 4. Transport to school is to expensive 5. I am never late for school 6. Other Please specify _____	Answer (Write only one number)

Academic (Please provide us with the most relevant answer)

On average how many hours per day do you spend studying?				Hours per day
On average how many hours per day do you spend doing homework/assignments?				Hours per day
What was your overall academic performance at the end of the last term?	1.Excellent (100%-80%)	2.Good (80%-60%)	3.Average (60%-40%)	4.Poor (40%-10%)
Household Activities				
On average how many hours per day do you spend doing household work?				Average Hours per day
Please select what type of household chores you must do at home?	1. Clean the house 2. Cook for the family 3. Fetch water 4. Look after my brother and sister 5. Work in the family garden 6. Look after sheep/ goats /cattle 7. Fetching younger sibling from school/pre-school 8. None 9. Other please specific			Answer(Write only the number)
Safety (Please provide us with the most relevant answer)				
Have you been attacked in last month?	1.Yes		2.No	
If Yes what type of attack	1.Robbery	2.Sexual	3.Harassment	4.Armed Robbery
If Yes how many times in last month?	1-3	3-6	6-10	10+

CONTRACT TERMS AND CONDITIONS

1 As custodian of this bicycle, I agree that this bicycle shall be used primarily for the purpose of improving my access to education, which involves students to attend school faithfully and timely using the provided bicycle.

2 I further agree that the bicycle will remain property of the ERP and that:

2.1 The bicycle shall not be sold or traded or exchanged for anything.

2.2 I must protect, maintain and take full responsibility for the cost of service and repair of the bicycle. Any theft or damage of the bicycle must be communicated to the Bicycle Supervisory Committee/School Principal or the Bicycle Supervisor and police within 24 hours.

2.3 The bicycle shall not be misused which may involve not achieving the primary purpose stated above. The misuse may involve but is not limited to overloading, transportation to bars or beer parties, defacing the applied stickers, etc.

2.4 I will allow the Bicycle Supervisory Committee or appointed Supervisor, to supervise the usage of the bicycle.

2.5 The Bicycle Supervisory Committee or appointed Supervisor will have full authority and sole discretion in determining if the received bicycle is being used inappropriately or not maintained satisfactorily.

2.6 The Bicycle Supervisory Committee or appointed Supervisor has sole authority to withdraw the bicycle without negotiation if they determine the bicycle has been misused or not maintained.

2.7 ERP or its partners will not replace or insure the bicycle and are not liable for any injuries or damages while using the bicycle.

2.8 The bicycle can be used by members of my household only if the conditions of faithful, timely school attendance using the provided bicycle are maintained and no misuse occurs. Misuse by family members

will result in bicycle being removed immediately. Misuse may be exemplified as a situation where members of the household deny the beneficiary of its right to use the bicycle to travel to school and back.

PARENT/CAREGIVER

3 I further hereby give permission for the child to receive a bicycle and form part of the #Bikes4ERP programme. By signing this contract, I take responsibility of educating my child on the responsible use of a bicycle and commit to ensuring that my child understands the risk involved in using the bicycle.

4 I fully understand and give permission for my child(ren) listed below to participate in events that are developmental to him/her and that photos taken can be used for reporting, publications as deemed necessary.

5 I understand that my child(ren) will not receive any money as a result of participating in events or being published in any documents.

6 I understand that if at any time I am not satisfied or uncomfortable with my child participation, through events that are taking place, I may withdraw my child(ren) from such participation.

6.1 In such an event I will return all items delivered to me including but not limited to the bicycle, helmet, pump, lock, spanner and tool.

7 The undersigned authorizes ERP to permit the use and display of said photographs and video in any publication, multimedia production, display, advertisement, or world-wide web publication.

8 The undersigned agrees that ERP may use name, likeness or biographical information supplied by the undersigned.

9 The undersigned understands that no compensation is offered in exchange for the images/information.

10 The undersigned releases and forever discharges ERP, their agents, officers and employees from any and all claims and demands arising out of or in connection with the use of said photographs/images, including but not limited to, any claims for invasion of privacy or defamation.

11 I further agree that after each calendar year, I will report on the use of the bicycle and when I leave my current school after completion of my studies, I will return the bicycle back to the school for other learners to use and benefit

12 I further agree to have my bicycle serviced and repaired the appointed Supervisor and Field Mechanic during the period of use

Having understood the terms and conditions of this contract, I hereby bind myself thereto by appending my signature. This _____ day of _____, 20___

Signature - Beneficiary	Name and Surname of Beneficiary
Signature - Parent/Guardian for Student	Name and Surname of Parent/Guardian for Students
Signature of BSC Chairperson	Signature of Program Manager

1.3 Appendix C: Other Systems Investigated

1.3.1 Literature Studies

1.3.1.1Visybl Literature Study

1.3.1.1.1 Introduction

Visybl is a modern-day asset tracking solution that is designed to automate tracking and reporting functions to ensure the system is quick and easy to use. Visybl makes use of Bluetooth low energy device technology to track the assets. Visybl will allow the admin to track when an asset leaves and arrives at a certain location as well as to notify the client on movements and activity as well as allowing the client to generate reports on the specified asset.

1.3.1.1.2 Economic feasibility

- Visybl will have a cost of R4683 for 6 beacons and one cloud node and will require a further cost of R60 per month per bicycle which includes:
    - Visybl Beacon cost is included within subscription plan.
    - Visybl Beacon Battery management and beacon swap prior to battery depletion.
    - Fully managed CloudNode with OTA Firmware updates.
    - Unlimited no. of users in a single company account.
    - Unlimited use of Visybl Cloud, Visybl iOS, Android, and Amazon Alexa Apps.
    - Unlimited push notifications and email alerts.
    - Quarterly new features/software updates to Visybl Cloud and mobile Apps.
- This makes it a very expensive system as more bicycles are added to the system.

1.3.1.1.3 System Features

- Visybl comes with many features but a lot of features will not be used by the user. The following are a list of the features:
    - Visybl Beacon cost is included within subscription plan.
    - Visybl Beacon Battery management and beacon swap prior to battery depletion.
    - Fully managed CloudNode with OTA Firmware updates.
    - Unlimited no. of users in a single company account.
    - Unlimited use of Visybl Cloud, Visybl iOS, Android, and Amazon Alexa Apps.
    - Unlimited push notifications and email alerts.
    - Quarterly new features/software updates to Visybl Cloud and mobile Apps.

1.3.1.1.4 Operational Features

- Asset management
- Tracking management
- Beacon management

1.3.1.1.5 Lacking Features

- Visybl lacks key features that are required which include:
    - Monitoring of bike maintenance.
    - Scheduling mechanic tasks.
    - Adding a bicycle repair.
    - Tracking student progress.
    - Generating various reports.
    - Scheduling mechanic job

1.3.1.1.6 Conclusion

Visybl is a very impressive system but it was designed for the purpose of tracking business assets but lacks features that are key for the client as it is not a customized system. The tracking software is appealing but lacks maintenance features which are vital for the system.

1.3.1.2 AssetGather and Booqable

1.3.1.2.1 Introduction

AssetGather is a software solution created by AssetPulse which is an RFID service provider who offer customizable RFID tracking solutions, they also assist in asset tracking and inventory processing. They are relevant as they offer solutions that could be sustainable in extreme outdoor conditions. Booqable is a rental software that turns complex processes into simple streamlined processes.

The RFID based tracking system consists of 3 basic components – the RFID reader, RFID tags and RFID software.

1.3.1.2.2 Economic Feasibility

- AssetGather costs R2890 per month which includes all software required to run the system.
- Booqable will cost R3780 per month which includes all the software required to run the system. The client will have to purchase RFID tags for each bike as well as Antennas to install in the areas.

1.3.1.2.3 System Features

- AssetGather comes with many features with regards to tracking an asset. Most of the features are required by client as well as features that won't be used by the client:
    - Comprehensive, customizable web-based dashboard & reports
    - Intelligent rules and alerts engine
    - Track and trace asset history
    - Track and receive missing inventory alerts
    - Role-based views - multi-level user access

- Secure and scalable enterprise-class architecture
- Flexible integration with other enterprise applications

- Booqable comes with many features with regards to Scheduling, reporting and booking.
- Most of the features are required by client as well as features that won't be used by the client.
  - Visual planning and zero overbooking.
  - Optimize the way you manage the bicycles
  - Report generation

1.3.1.2.4 Operational Features

- Asset management
- Schedule management
- Tracking management
- RFID antenna management

1.3.1.2.5 Lacking Features

- AssetGather/Booqable lacks key features that are required which include:
  - No notification processes
  - Tracking student progress
  - No Bicycle usage report

1.3.1.2.6 Conclusion

AssetGather is a very impressive system which was designed for the specific purpose of tracking business assets but lacks features that are key for the client as it is not a customized system. The tracking software is appealing but lacks maintenance features which are vital for the system. Booqable is a good system which was designed for the specific purpose of scheduling and asset management. But these systems have to integrate together in order to meet all client requirements.

1.4 Appendix D: 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 camera, scanner, voice recording device, thump print reader, etc.)		6	
	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