



ASHESI UNIVERSITY COLLEGE

INVENTORY MANAGEMENT SYSTEM FOR CAMFED ZIMBABWE

APPLIED PROJECT

B.Sc. Computer Science

Linda Bhebhe

2018

ASHESI UNIVERSITY COLLEGE

Inventory Management System for Camfed Zimbabwe

Applied Project

Applied Project submitted to the Department of Computer Science, Ashesi
University College in partial fulfilment of the requirements for the award of
Bachelor of Science degree in Computer Science

Linda Bhebhe

April 2018

Declaration

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

.....

Candidate's Name:

.....

Date:

.....

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University College.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

.....

Acknowledgements

My profound gratitude goes to my project supervisor Mr David Sampah for constantly checking on my progress and giving me feedback. The successful completion of the project would not have been possible without his assistance and expertise. I would also like to thank all the faculty members in the Computer Science Department at Ashesi for supporting me throughout my study and MasterCard Foundation for funding my study.

I am also indebted to my study mates and friends Abdul Razak and Lynn Mumia for ensuring that I remain focused at all times . In a special way I also thank Molife Chaplin for the motivation to start this project and helping me stay awake to complete all tasks during the difficult times of the semester.

I would also like to thank all the staff members at Camfed Zimbabwe for the permission to work on this project and their support and constant feedback.

To all my family members I express my gratitude for being there for me and supporting me in various ways.

Above all I thank God for the wisdom, strength and good health.

Abstract

In this digital age, it is unanticipated that some organisations and firms still use the manual book system to keep records. Research on the reasons why some organisations are still using the manual way proved that usability is an important aspect when developing a record keeping system. Most available record keeping or inventory systems are tailored for businesses involved in buying and selling goods. Since for such businesses there will be a designated person or few people responsible for record keeping, they take some time to learn the system and become familiar with it. Resultantly, they can navigate even on seemingly sophisticated user interfaces. Due to the complexity of the inventory systems available, non-retailers prefer to use the manual system which requires no proficiency and less or no time to learn.

This paper details the development of an inventory management system for Camfed (Campaign for Female Education) Zimbabwe following a request made by the organisation. With more emphasis on usability the system was developed ensuring that each interface has the minimal functionality. Frequent testing was done to ensure that the system meets the requirements of the user. It was implemented using Html, CSS, PHP, JavaScript and other web technology languages and frameworks. The system has a very usable and interactive interface which ensures a highly satisfying user experience.

Table of Contents

Declaration	i
Abstract	iii
Table of Contents	iv
List of Tables.....	vii
List of Figures	viii
List of Abbreviations.....	x
Chapter 1: Introduction	1
1.1 Introduction	1
1.2 Background Information –Camfed.....	1
1.3 Problem Description and Motivation	2
1.4 Related Work and Existing Solution	3
1.5 Aims and Objectives	5
1.6 Project Outline.....	5
Chapter 2: Requirements	7
2.1 Requirements Design Overview.....	7
2.2 Data Collection.....	8
2.2.1 Interviews	8
2.2.2 Observational Study	9
2.3 System Scope	10
2.4 Application Features	10
2.5 User Characteristics.....	10
2.6 Operating Environment	11
2.7 Functional Requirements.....	11
2.8.1 Usability and Performance	12
2.8.2 System Configuration.....	13
2.8.3 Security.....	13
Chapter 3: Architecture and Design	14
3.1 Architecture Outline	14
3.2 Modules and Mock-ups	15
3.2.1 Login	15
3.2.2 Request Form	16
3.2.3 Admin Operations	17

3.4 System Architecture	18
3.5 Database and Data Design.....	19
3.6 Activity Diagrams	20
Chapter 4: Implementation.....	22
4.1 Implementation Description.....	22
4.2 Software and Tools.....	22
4.2.1 Front End.....	22
4.2.2 Back End	23
4.2.3 Database and Server	23
4.2.4 API	24
4.3 Major Implementation Decisions	24
4.4 Classes and Methods	24
4.4.1 Stationery Class.....	24
4.4.2 Equipment Class.....	25
4.4.3 IT Support Class.....	25
4.4.4 Database connection class	26
4.4.5 User Class.....	26
4.5 Interfaces	26
4.5.1 Dashboard Landing Page	26
4.5.7 Web App Interface	31
4.5.8 Mobile App Interface	33
4.6 Sequence Diagram.....	35
4.7 Code Snippets.....	35
Chapter 5: Testing	39
5.1 Development Testing	39
5.1.1 Unit Testing.....	39
5.1.2 Component Testing	40
5.1.3 System Testing	41
5.2 Usability Testing	41
5.3 Acceptance Testing	42
Chapter 6: Conclusion and Recommendations	43
6.1 Conclusion.....	43
6.2 Challenges	44
6.3 Future Work	44

References 45

Appendices 46

List of Tables

Table 2.1. Requirements analysis overview	8
Table 2.2.1. Interview responses	9
Table 4.4.1. Stationery class.....	25
Table 4.4.2. Equipment class	25
Table 4.4.2. IT support class	25
Table 4.4.4. Database connection class.....	26
Table 4.4.5. User class	26
Table 5.2. Usability testing	41

List of Figures

Figure 2.7.1: IT inventory use case diagram	11
Figure 2.7.2: Stationery inventory use case diagram	12
Figure 3.1: MVC architecture	14
Figure 3.2.1: Sign-in prototype	16
Figure 3.2.2: Stationery requests form prototype.....	17
Figure 3.2.3: Admin operations prototype	18
Figure 3.4: : System architecture.....	19
Figure 3.5: ER diagram	20
Figure 3.6.1:Admin officer activity diagram	21
Figure 3.6.2: District officer activity diagram.....	21
Figure 4.5.1.1: Dashboard badges	27
Figure 4.5.1.2: Dashboard sample pie chart.....	28
Figure 4.5.1.3: Dashboard sample graph.....	28
Figure 4.5.2: User registration.....	29
Figure 4.5.3: IT support requests	29
Figure 4.5.4: Stationery dispatch table.....	30
Figure 4.5.5: Equipment requests view	30
Figure 4.5.6: Dashboard Notifications	31
Figure 4.5.7.1: Landing Page	31
Figure 4.5.7.2: User login.....	32
Figure 4.5.7.3: Stationery request interface	33
Figure 4.5.7.4: Request status	33
Figure 4.5.8: Mobile interface.....	34
Figure 4.6: Sequence diagram.....	35

Figure 4.7.1: Code snippet 1 36

Figure 4.7.2: Code snippet 2 37

Figure 4.7.3: Code snippet 3 38

Figure 5.1.1: Registration form validation 40

List of Abbreviations

HTML - Hypertext Mark-up Language

CSS – Cascading Style Sheet

AJAX – Asynchronous JavaScript and XML

API – Application Programming Interface

HTTP – Hypertext Transfer Protocol

ER – Entity Relationship

SQL– Structured Query language

Chapter 1: Introduction

1.1 Introduction

Inventory management is often related to retailers. It is perceived to be a way of keeping goods, recording sales and orders as well as measuring profits. Due to this perception, most non-retailers who keep track of stock, tend to use the traditional way of recording in a book. This is also due to the fact that inventory management systems are designed to suite retailers. This limits the non-retailers, since these systems will be too sophisticated regarding the data they intend to keep track of and the expertise of their workers. Basically, an inventory management system is a method or system that allows you to have critical information on your physical goods and their movements. The benefits of such a system should not be overlooked by non-retail organisations.

This report details the development of the Camfed equipment request application, which has been named CamBeep. This is a hybrid mobile application developed for Camfed; an organisation that supports young women from marginalised areas through school. The application will help the organisation to keep track of the IT equipment and manage the disbursement of client entitlements to the various districts in the country. The goal is to increase efficiency and accountability in the process of disbursing entitlements as well as aiding every member of the organisation to report and get assistance on any IT related complaint promptly.

1.2 Background Information –Camfed

Camfed is a non-profit organization tackling poverty and inequality by supporting marginalized girls to go to school and succeed, and empowering young women to step up as leaders of change. Founded in 1993 in Zimbabwe, supporting 32 girls through school, in 2017 the support has extended to more than 1.5million children through a network of schools in Zimbabwe, Ghana, Malawi, Tanzania and Zambia. Girls supported through

Secondary School are provided with either holistic or targeted support covering needs that include school fees, uniforms, books, pens, boarding fees and disability aids.

Camfed works with partner communities which are among the most deprived. Communities which are far from hospitals, public infrastructure and in the poorest land. The organisation measures its impact through rigorous monitoring and evaluation, longitudinal surveys and in-depth research. To maximise its impact, the organisation works with several partner schools and community volunteers which include traditional leaders, government education officials, teachers, parents and former students. Apart from providing school fees and other required supplies for education, the organisation also offers trainings for women. The trainings help the women with financial management and entrepreneurship (Camfed, 2017).

1.3 Problem Description and Motivation

Some of the core values of Camfed are transparency and accountability. The organisation ensures that all its processes are transparent and everyone is accountable in the handling of financial resources and equipment. However, accountability is not fully achieved due to the absence of a resources management system for the organization. This realisation was made during the developer's internship period with the organisation between June and August 2017. For instance the process of reconciling the stationery that was disbursed to the clients was cumbersome. It required the administration officer to spend more than a day going through the organization's stationery request book.

Despite the factors such as time efficiency in resource reconciliation and auditing, transparency and accountability, the security of the organisation's data is of much concern. In the event that the "stationery request book" is lost or stolen, the organisation would have lost all the data concerning the disbursements. The use of files to store data also increases

the amount of paper in the organisation which then requires a large amount of storeroom space to store the files.

Given that there are various inventory systems developed already, it remained a question to me why Camfed was not using any of these. The user requirements analysis proved that usability is the most important feature for the organisation. Most of the people in the organisation require a very simple and usable system. This makes it problematic for the organisation to adopt any of the existing systems.

1.4 Related Work and Existing Solution

Interviews with the employees in the organisation revealed that prior to using the current log book system, the organisation had a Salesforce inventory system. 90% of the employees cannot use the system due to the difficulty of understanding it. This system was also integrated with an application that reports cases requiring IT support to the IT administrator and a platform for making announcements in the organisation. Even though this system had important features, the user interface was not friendly. Improving the existing application to suit the needs of the organisation was another option. However, given that modifying the user interface required rebuilding most parts of the system, developing a new usable system was the best option.

According to Holzinger (2005), system usability is the ease of use and acceptability of a system, for a particular class of users, carrying out specific tasks in a specific environment. Usability characteristics include learnability, efficiency, memorability, low error rate and satisfaction. These are important factors for the product goals to be achieved. Ensuring these reduces the chances of system failure.

A system can be designed such that it is easy to deploy, install and access but still not achieve its purpose. If it requires the users to be experienced before they can navigate through it, they will still bypass some of the proper channels of using it or not use it at all.

The user interface design should be considered as equally important as the other features like the ease to install and deploy. This was evident in the analysis of the usability of the ILab inventory system which manages the inventory from interconnected laboratories supporting workflows, requests and searches. The study which was divided into four parts: planning, analysing, testing and redefining the system, proved that the user interface design is the main determinant of whether the system will be used or not. Poor user interface design is generally one of the weaknesses of the existing inventory systems (Rhinho, R., Sousa, A., & Restivo, A., 2010).

Most often, inventory system developers tend to separate themselves from the users with the assumption that they clearly know what the users want, since an inventory system is usually straightforward. This prevents the designers from receiving important feedback from the users. While coming up with the system can be challenging, Shackel (1991) identified that designing what real users really want is a problem. Given that usually those who give requirements may be the managers in the organisation and not the users who use the system for daily activities, it is the duty of the developer to interact with the real users such that the system will be usable, useful and used. In the case of developing the system for Camfed, this is a loophole to take note of. Since the real users are mostly novices, there are high chances of having the system failing if the requirements are only determined by the organisation's management.

With the new paradigm of human computer interaction which incorporates virtual reality and augmented reality, user interaction with applications should be optimised using such knowledge. The shift from traditional web applications to modern ones with aspects of virtual and augmented reality will involve incorporating new functionality into applications like gesture control and voice interaction (Lo, Y.-C., Liu, X.-W., & Yang, S.-H, 2017). Such inventions improve the user interactions and help to reduce errors.

According to Karat (1997), usability is not only limited to the display and keyboard, but encompasses several artefacts that fits into a complex work or home environment. He defines usability of a system as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. There is need to focus on the environment into which the system will fit. Regarding this, any generic inventory system will not meet the needs of Camfed as an organisation.

1.5 Aims and Objectives

This project aims at developing a system that will change the perceptions of its target users, who see the traditional book system as adequate for the management of the inventory. By using Cambeep, the users will not only become familiar with this particular application but will be inspired to seek more technological solutions for the organisation.

Given that in Zimbabwe, Camfed operates in 24 districts, the hybrid mobile app will ease the synchronization of requests from each district to the main office. This will enable users to make requests on mobile devices even when travelling. To reduce the number of enquiries to the IT department for new workers' orientation, videos that can guide new users will also be incorporated. The application will also allow, among other functionalities, the constant update of the equipment database, and the generation of weekly reports for administration and auditing purposes. The application will have different views, each user will get a view based on their login credentials. The user view will be designed to suit each user's experience level.

1.6 Project Outline

This report will provide a description of how the application works and how it was designed. Chapter 2 discusses the requirements of the system; the design overview, the data collection process and the functional and non-functional requirements of the application.

Chapter 3 discusses the architecture and design of the system, the tools used and what influenced the choice of tools. It also comprises of the description of all the implementation modules, the database design, sequence and activity diagrams. Chapter 4 discusses the implementation of the system, the tools used and the snippets of some codes. Chapter 5 reports on the implementation results and system testing. Finally, Chapter 6 has the conclusion of the report, recommendations and plans for future work.

Chapter 2: Requirements

Requirements were gathered by interacting with the users of the system and from personal reflections on the work experience while interning with the organisation. Other requirements were also identified from studying the existing solutions and realising the loopholes in the systems. Observing the users as they interact with the existing tools and understanding the nature of the working environment in the organisation as well as its values was very instrumental.

2.1 Requirements Design Overview

A requirement analysis plan was outlined which helped to gather the requirements from the users and other relevant people. Requirements and information for a successful completion of the project were gathered from various people as shown in the table below:

Category	Resource people	Information required	Method used
Camfed	IT Manager	<ul style="list-style-type: none">• How the systems should be designed to be easily configured with the existing systems in the organisation.• The functionality that will be important and why.• The challenges currently faced with the existing systems.	Interviews
	District Operations Officers	<ul style="list-style-type: none">• The challenges they face with the existing systems and how the problem can be solved	Interviews and Observation
	Departmental Managers	<ul style="list-style-type: none">• The important functionality of the system.	Interviews and observation
	Administration officer	<ul style="list-style-type: none">• How the system could be designed such that it does not deviate from the existing methods.	Interviews

	Executive members	<ul style="list-style-type: none"> • How the system should be designed and installed not to bring extra costs in the organisation. 	Interviews
	Interns	<ul style="list-style-type: none"> • The problems they have identified in the organisation in terms of inventory management. 	Interviews
Ashesi University College	Project Supervisor	<ul style="list-style-type: none"> • Guidelines for a successful completion of the project 	Scheduled meetings
	Other Staff and Faculty	<ul style="list-style-type: none"> • Ideas on how to successfully complete the project and the required expertise for developing the system. 	Open Discussions
	Colleagues	<ul style="list-style-type: none"> • Expertise and ideas 	Open Discussions
Other	Software Developers	<ul style="list-style-type: none"> • The best practices when designing a modernized inventory system. 	Open Discussions

Table 2.1. Requirements analysis overview

2.2 Data Collection

Data was collected through interviews, open discussions and observational study.

2.2.1 Interviews

The interviewees responded with different perspectives based on their roles and how they interact with the existing system as shown below;

<p>Role A</p> <ul style="list-style-type: none"> • The system that was initially used by the organisation had a sophisticated user interface which resulted in most users failing to navigate on it. • The system required some procedures to open and login which most users were not able to follow • The users required training before they can use the system and were prone to errors. 	<p>Role B</p> <ul style="list-style-type: none"> • Suggested that a web based system that manages the stock database would aid in generating reports and getting timely prompts if the organisation is running out of stock. • Requested that the system should send reminders to the persons responsible if they are supposed to return equipment to the National Office and on the day of collection of the equipment they requested for.
<p>Role C</p> <ul style="list-style-type: none"> • Highlighted that the system aid the users to make or approve requests from their respective locations. 	<p>Role D</p> <ul style="list-style-type: none"> • The system will increase efficiency in the organisation.

Table 2.2.1. Interview responses

2.2.2 Observational Study

Interning in the IT department gave the developer an opportunity to observe how the District Office Secretariats and other staff interact with the systems. Following this, the need to incorporate audio visuals in the application was identified. This will reduce the number of times they contact the IT manager for help. This will also limit the time spent

on orienting every new member in the organisation since they will be able to learn on their own how to use the system.

2.3 System Scope

The system to be implemented will be able to manage the stationery, IT equipment and IT support requests for Camfed Zimbabwe. It will be hosted on a local server during the earlier stages of development and will be later transferred to an online server to aid the users to test and give feedback to the developer. The users will be able to use the system for a week to identify areas of improvement before the final version of the application is dispatched to them. The mobile version of the application will be compiled and installed on the mobile devices of the users. Finally, the application will be transferred to the Camfed server and installations will be done accordingly with the help of the organisation's IT department.

2.4 Application Features

Each user's login credentials will be used to determine the pages the user is given access to. Each user interface will have the least functionality possible to minimize having redundant functionality for the user.

2.5 User Characteristics

The main users of CamBeep are the District Office Secretariats, the Administration Officer and the IT Manager. They will use the system to make requests and have their requests approved if the items they requested for are available. The Administration officer will disburse the items requested, update the database and generate reports for the executive team. The IT Manager will use the application for all the IT related requests like troubleshooting by the members of the Organisation and for keeping track of the IT equipment.

2.6 Operating Environment

CamBeep is a hybrid mobile application that will work on all mobile platforms i.e. IOS, Android and Windows. The application will be supported by all browsers.

2.7 Functional Requirements

The use case diagram below shows the functional requirements of the system in relation to the IT inventory and how the stakeholders in the IT department and the Administration will interact with system.

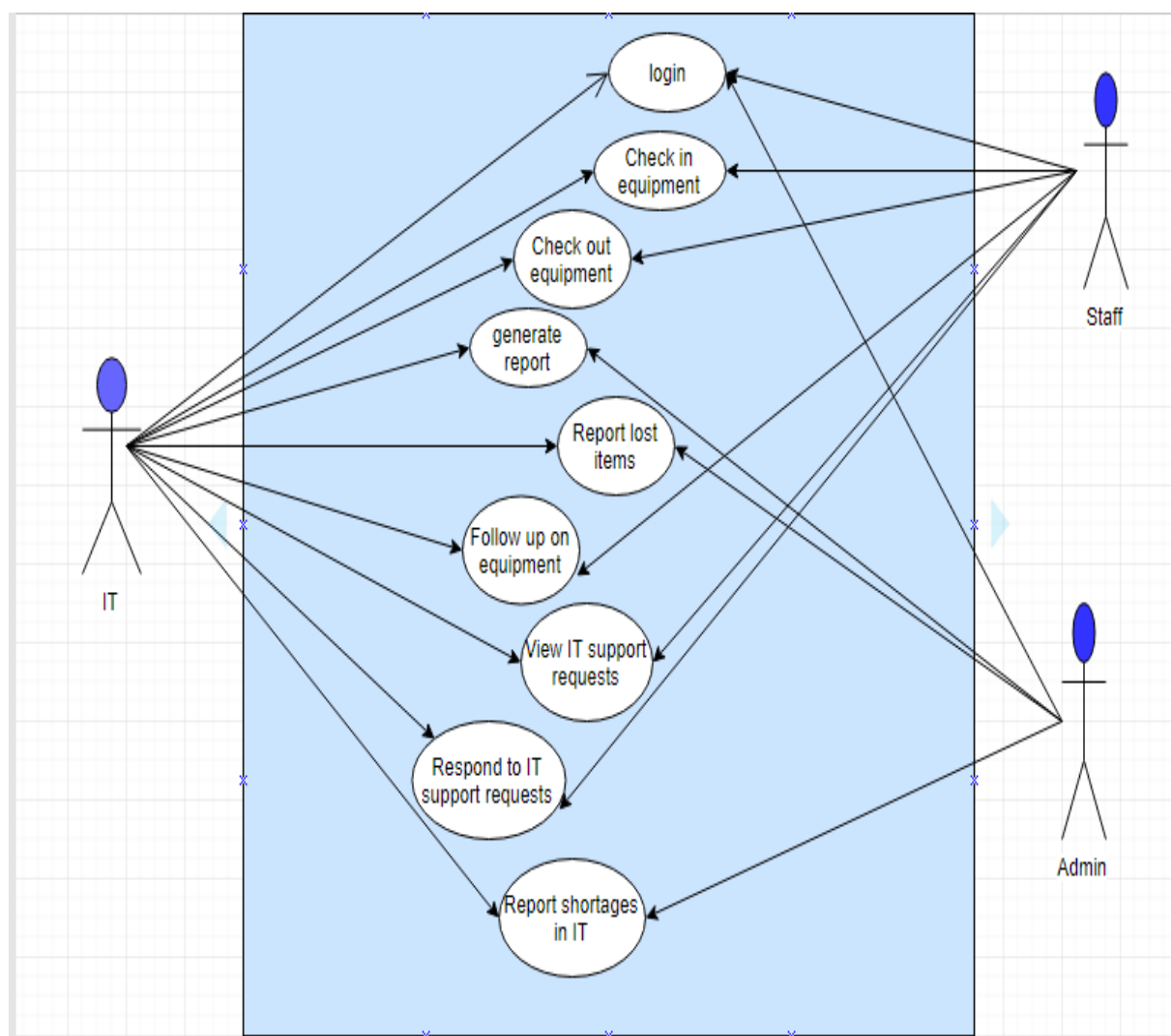


Figure 2.7.1 IT inventory use case diagram

The use case diagram below shows how the users will interact with the system in managing the stationery inventory;

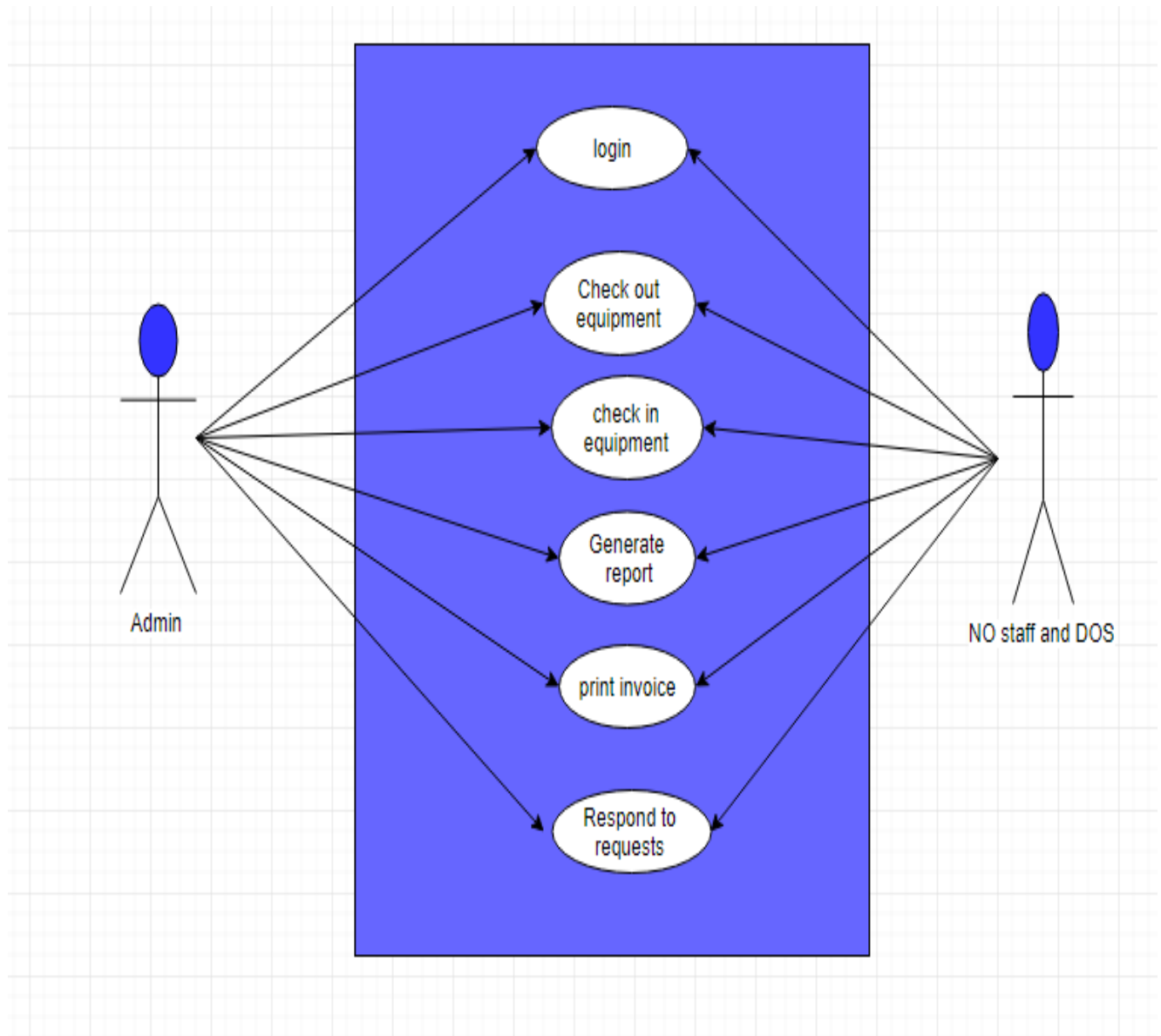


Figure 1.7.2. Stationery inventory use case diagram

2.8 Non- Functional Requirements

The application is expected to meet non-functional requirements that include usability, good performance and easy configuration.

2.8.1 Usability and Performance

The user interface should not have redundant information for a user. Each user should only be able to access what they need and have an interface that is friendly and easy to navigate. Users should be able to access functionalities like generating reports from the system and printing without any difficulty.

The records kept by the system should be up to date and available all the time. Information should be readily available. The system should be running at all times to enable users to make requests at any given time. The admin officer and the IT officer should receive notifications each time requests are made for entitlements and for IT equipment respectively.

2.8.2 System Configuration

The system should not bring extra cost to the organisation during installation. It should be designed to merge perfectly with the existing system in the organisation. The design of the system should be easy to understand such that future maintenance will be easy.

2.8.3 Security

Each user must have unique identification in the system to enable accountability. The administrator should be the one to register users in the system and only users who are registered should be able to login and use the application.

Chapter 3: Architecture and Design

This chapter details the structure of the system, the system design, database design and the data required for the system. The system adopts the incremental model involving various testing, prototyping and gathering feedback from the users on how they expect the system to work.

3.1 Architecture Outline

To be able to clearly see the interaction between the various components of the system, the system uses the Model, View, Controller (MVC) approach. This enables the developer to monitor the system easily and prevent errors. It also enables the developer to easily follow the code to make changes when errors occur at each component without affecting other components.

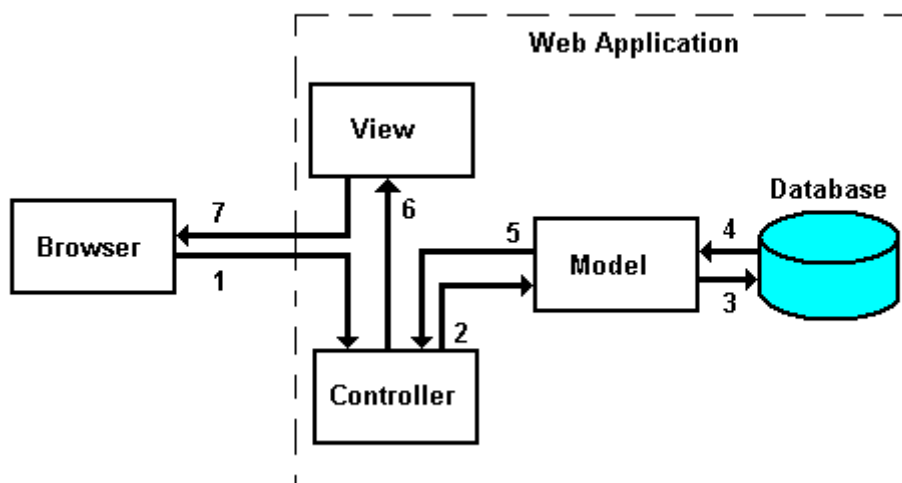


Figure 2.1. MVC architecture

Figure 3.1 above illustrates the architecture and how information flows in the application. In step 1 the user clicks a button to submit a form. Using the URL, the form will be sent to the controller which communicates with the model. Step 3 and 4 shows how data is written and read from the database respectively. In step 5 the results from the model are passed to the controller and in step 6, the results are passed to the view which then

presents the output in a well formatted output display to the user in the browser (Tech, 2015).

3.2 Modules and Mock-ups

In designing the system using the incremental model, key modules to be executed were identified on a high level and mock-ups of the system were built for the developer to have a clear view of the flow of the system. The modules identified were as follows;

- Login page
- Main page
- Add and remove a user
- Password change
- Make a stationery request
- Make an IT support request
- View IT support requests
- View stationery requests
- View IT equipment requests
- Add equipment
- Add stationery
- Generate report
- Send email reminder

3.2.1 Login

At the login page, the user enters their username and password. Based on the privileges that they have, they are redirected to a user interface corresponding to their access level. The login page is the most important part of the system as it authenticates the users accordingly and directs them only to the interfaces that they are allowed to access. By correctly redirecting the user, the system eliminates user errors by giving the user a wrong

interface or an interface with functionality beyond/below their access level. Fig 3.1 below shows the mock-up of the login interface.

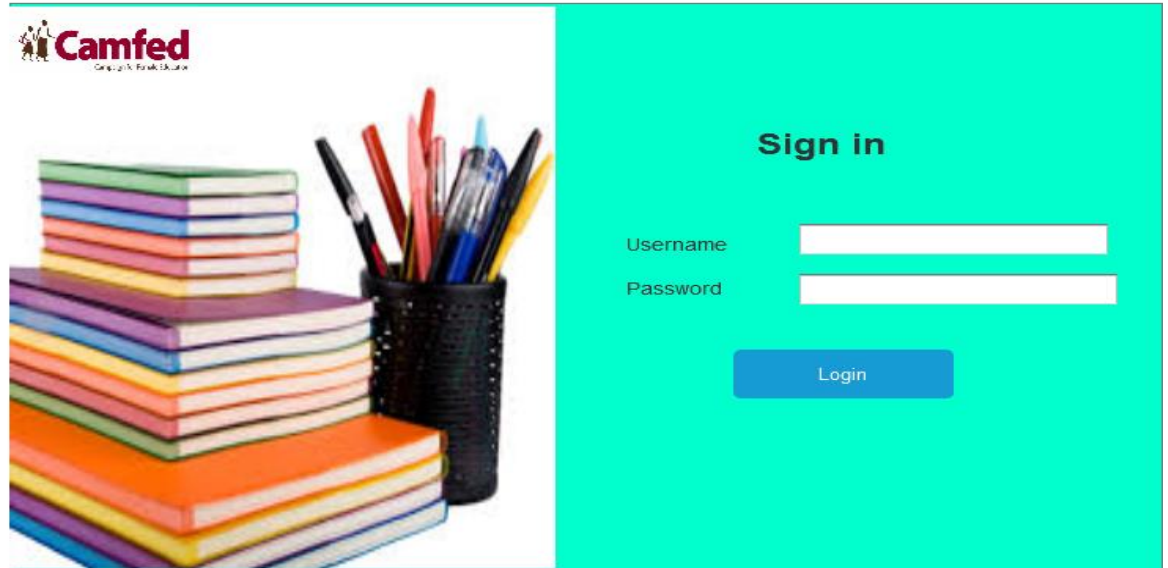
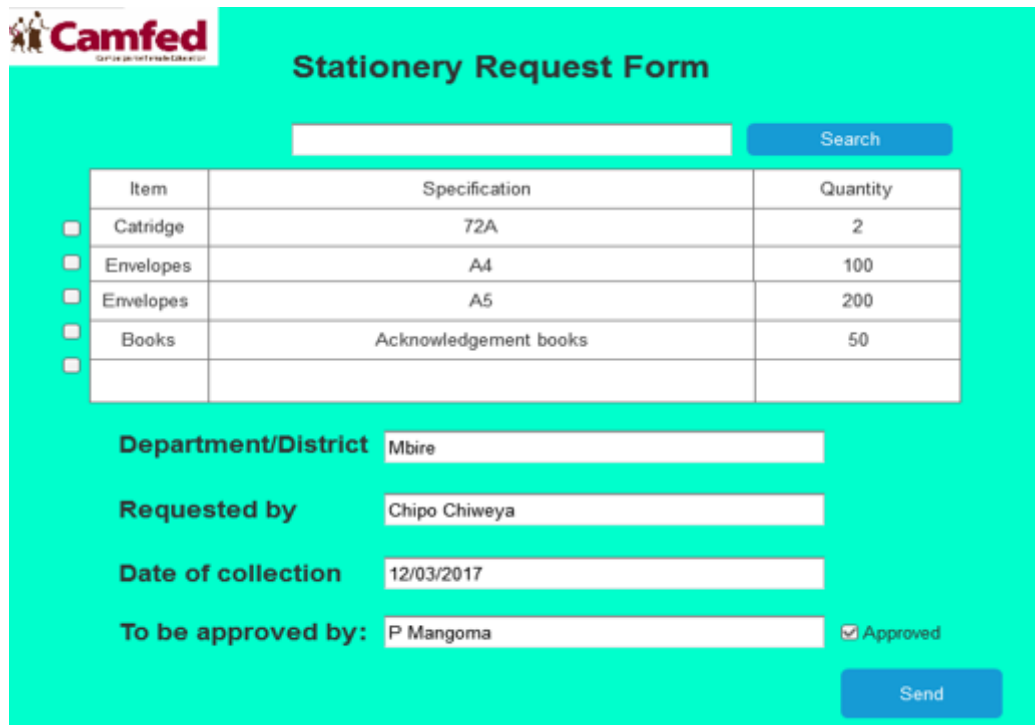


Figure 3.2.1. Sign-in prototype

3.2.2 Request Form

The prototype of the request form was designed to look like a page in the existing book for making requests. To reduce user confusion, they will have the same form layout on the request page as the manual system (book) which is in Fig 1 of the appendix. They will search for the name of what they want to request for. The search results from the database will be populated in a table form and they will check against the item they want to request for, specify the quantity and select the admin member to approve the request, before submitting the form. The system will automatically send a prompt email to the approver to approve the request. The request form is as shown in Fig 3.2.2 below;



Camfed
GIVING WOMEN THE EDUCATION

Stationery Request Form

	Item	Specification	Quantity
<input type="checkbox"/>	Catridge	72A	2
<input type="checkbox"/>	Envelopes	A4	100
<input type="checkbox"/>	Envelopes	A5	200
<input type="checkbox"/>	Books	Acknowledgement books	50
<input type="checkbox"/>			

Department/District

Requested by

Date of collection

To be approved by: ☒ Approved

Figure 3.2.2. Stationery requests form prototype

3.2.3 Admin Operations

The Admin Officer is the main user of the system. Unlike other users who will only use the system to make requests, the Admin Officer will have to be able to check-in new equipment received by the organisation. The application should allow the admin to read the barcode of the new equipment and insert it into the database with other relevant information. Also, the admin will have to be able to view the requests and check-out the stationery requested for. At the end of the month, the admin should be able to generate a report about all the stationery transactions made. This report should also allow the executive staff and auditors to monitor how the stationery is disbursed in the organisation. While the admin is responsible for checking out equipment, they can also checkout the equipment to themselves. Fig 3.3.3 below shows a mock-up of the functionality for the admin.

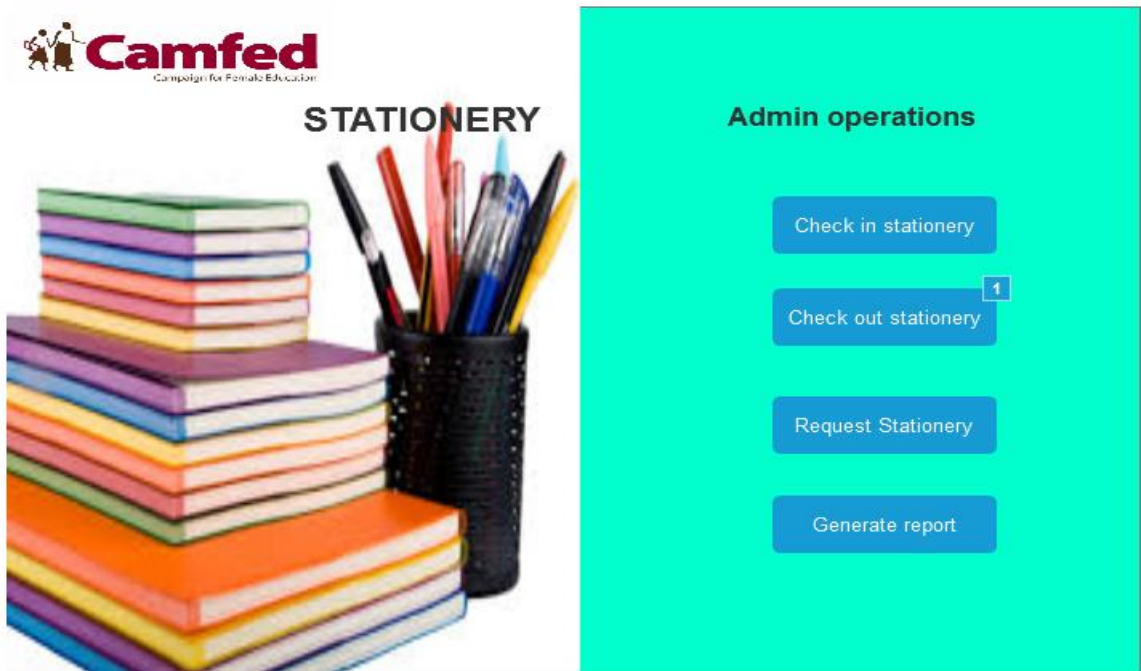


Figure 3.2.3. Admin operations prototype

By clicking the checkout stationery button, the admin will view all the requests made.

3.4 System Architecture

Figure 3.5 below shows the architecture of the system. The model has a layered structure showing how data moves from one layer to the other in the system. The user interacts with the application layer which has the user interface. The application layer has the logic of how the system works and is also responsible for sending the data to be displayed in the application layer. The database layer communicates with the database when queries executed.

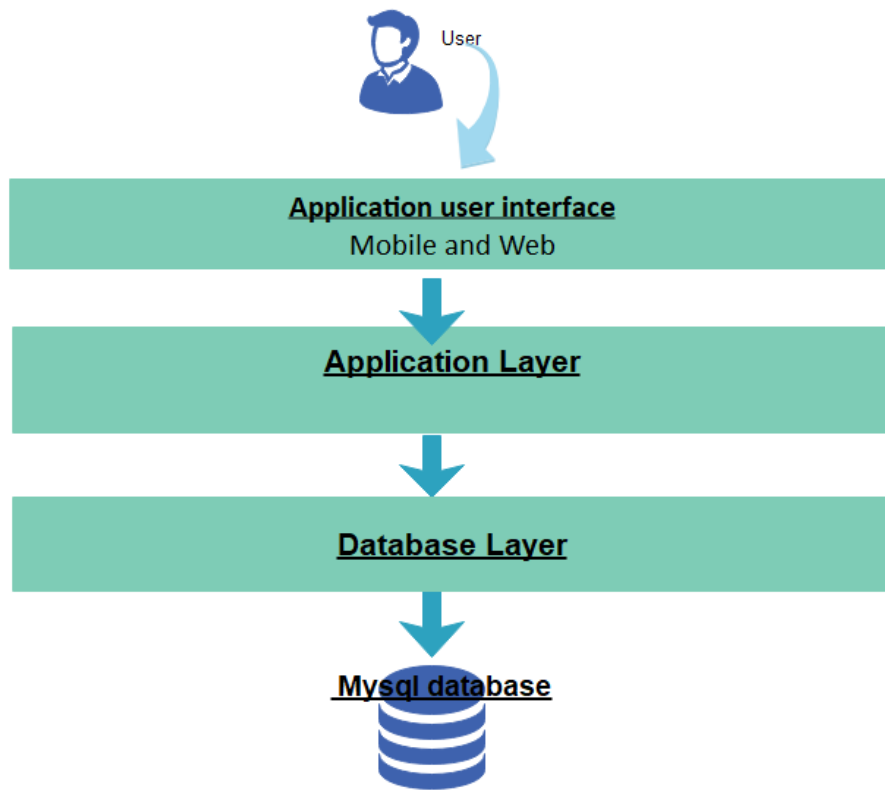


Figure 3.4. System architecture

3.5 Database and Data Design

The database system of CamBeep includes entities like staff members, the administration members, and the inventory equipment that the system monitors. It also keeps track of the transactions that are made for easy retrieval. The design of the database is as shown below:

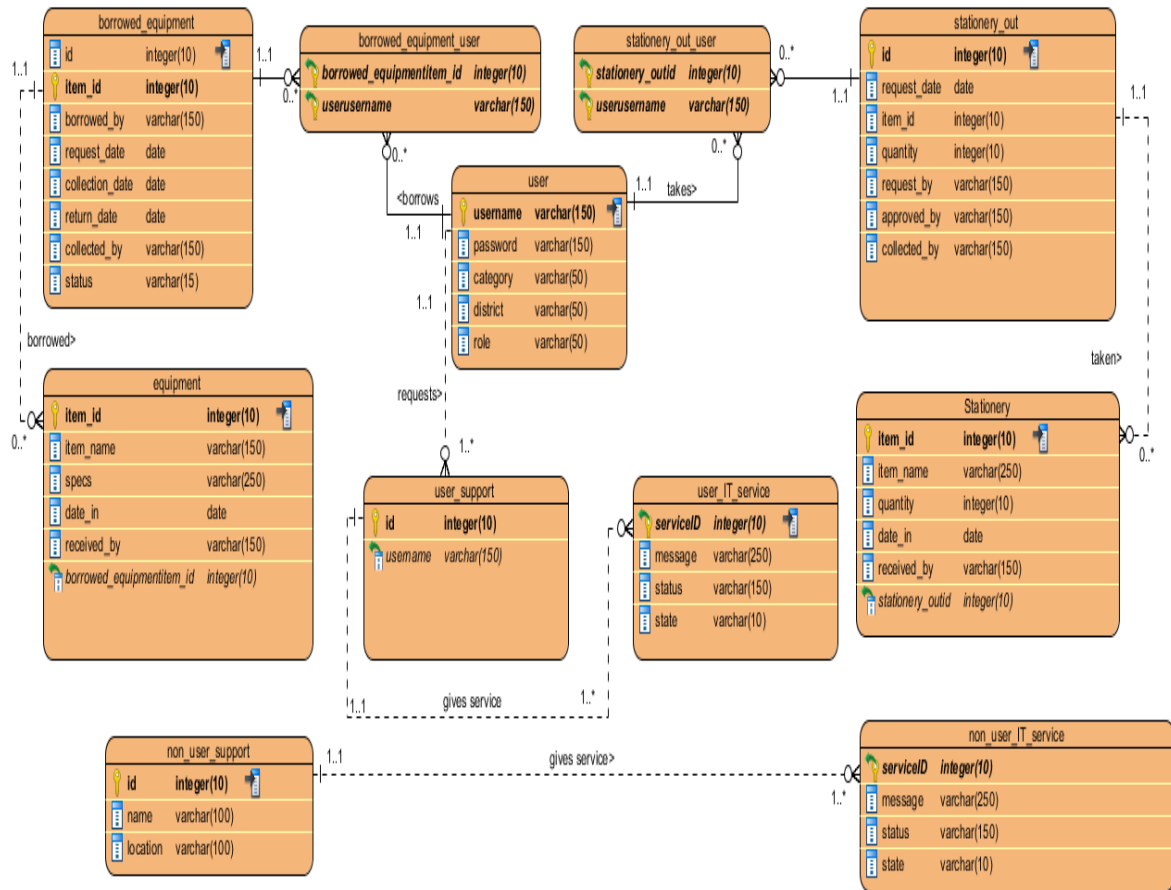


Figure 3.5. ER diagram

3.6 Activity Diagrams

The activity diagrams show how the admin officer and the district officer will interact with the system respectively on the main functionality that they will be using.

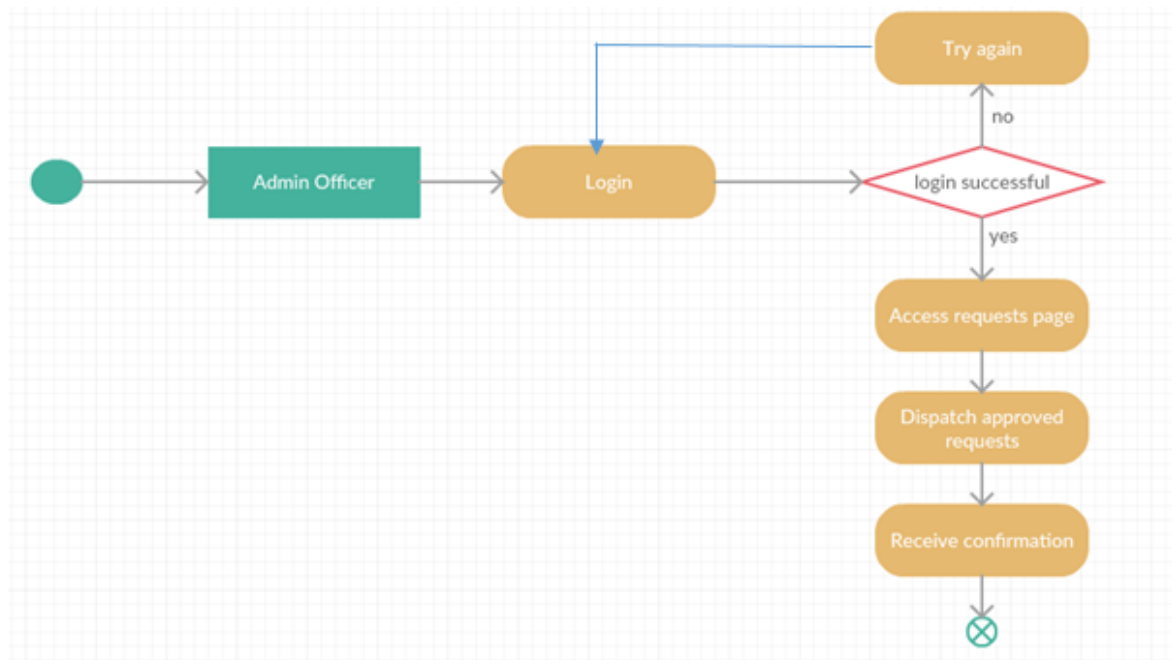


Figure 3.6.1.Admin officer activity diagram

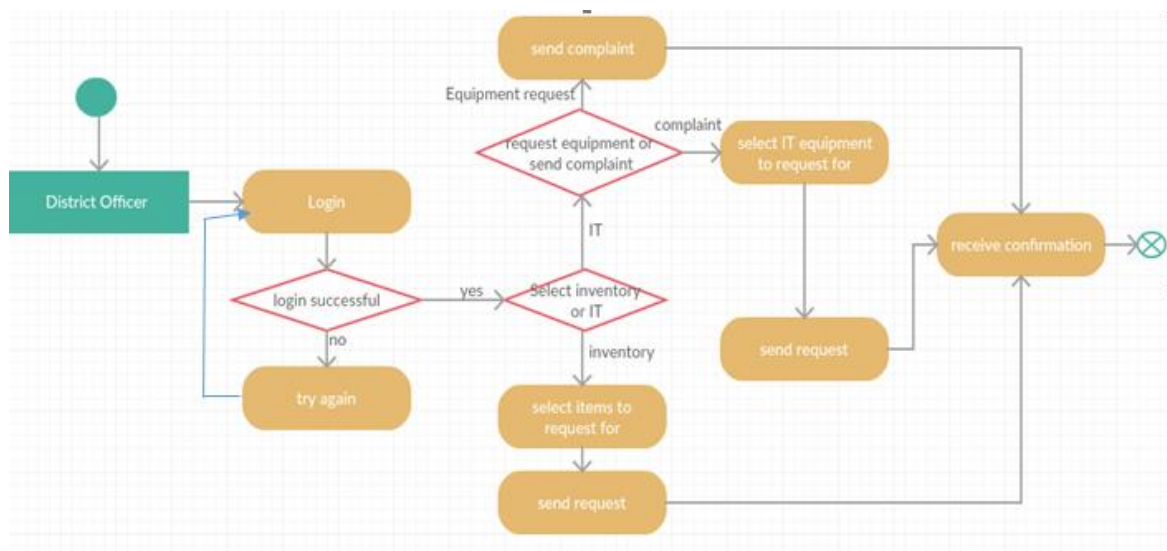


Figure 3.6.2. District officer activity diagram

Chapter 4: Implementation

This chapter describes how the Cambeep app was implemented. It highlights all the resources that were used, the tasks that were of major importance and the languages and libraries that were utilized. It also includes the snapshots of the various pages that were developed and some important decisions that were made at this stage.

4.1 Implementation Description

The application was designed to be both a web app and a mobile app. These two versions were implemented using the same codes. However, the mobile version required an additional compilation to be deployed as an app on mobile devices, while the web app can be accessed on a browser. The implementation followed the MVC approach which enabled developing each component of the application independently.

4.2 Software and Tools

Various tools and software were used during the implementation as listed below;

4.2.1 Front End

HTML: Html is a programming language for formatting web pages. By combining it with JavaScript and CSS, the web page can be given a better design as desired by the users. Since the app is web based, HTML was the option. It is supported by all browsers allowing the users to access the app from any browser. It is very easy to use and understand. This will make it easy for the client to get cost effective solution providers to update the application in the future.

CSS: CSS was used to detail the presentation of the html pages to give a more desirable design. It allowed editing the colours, fonts and layouts for each page. CSS is a styling language and like HTML, it is compatible with all browsers.

JAVASCRIPT: To make the html pages more responsive and articulate, JavaScript was used. It was used to send requests to Ajax making the URL responsive.

4.2.2 Back End

PHP: Hypertext pre-processor is an open source language which can be embedded into HTML. PHP was used in this implementation because it is compatible with all platforms like windows and UNIX and offers relatively fast data processing on a web application.

AJAX: Ajax was used to send data to the server in the background of a page without reloading the page. It allows the loaded page to request and receive data without being reloaded making the page more interactive. It was also used because it works well along with HTML, JavaScript and CSS.

4.2.3 Database and Server

MYSQL: MySQL was used to create the database for the system. It was also the choice because it can handle large amounts of data and is easy to combine with PHP. Server:

Apache Server (XAMPP): XAMPP was used to run the application and build a database locally during the development.

000webhosting: This is a free web hosting which works with PHP and MySQL. This server was used to enable the initial testing and debugging of the mobile version of the app during implementation. It can be easily connected to FileZilla through the port numbers and the hosting credentials making the upload of documents to the server easier. Even though it is not secure, it was still used because at that particular state of implementation the security of the application was not of great importance.

Ashesi University CS server: This server was used in the final testing of the application because it is secure.

4.2.4 API

Apache Cordova/ PhoneGap: This is a mobile application development framework for building iOS and Android applications. It enables the use of standard web technologies and languages like html, CSS and JavaScript to develop an application and deploy it as a mobile app without it losing the native features.

4.3 Major Implementation Decisions

During development, the major decision made was to make the mobile application specifically for Android. This decision was made after the realization that all mobile devices within the organisation are currently android and there is no plans for changing this yet.

4.4 Classes and Methods

The tables below show the classes and methods for each class implemented in the back end.

4.4.1 Stationery Class

This class handles all the operations done on the stationery object. These operations include adding new stationery, updating the stationery details searching and viewing all the stationery in the database.

Method	Use
<i>view_stationery()</i>	Viewing a specified equipment in the database.
<i>request_stationery ()</i>	Inserting a request for equipment into the database
<i>add_stationery ()</i>	Adding a new equipment in the database
<i>delete_stationery ()</i>	Deleting an equipment from the database
<i>search_stationery ()</i>	Searching for an equipment in the database
<i>update_stationery ()</i>	Updating details on an equipment in the database
<i>view_all stationery ()</i>	Viewing all the equipment in the database

Table 4.4.1. Stationery class

4.4.2 Equipment Class

This class handles all the operations done on the equipment object. These operations include adding new equipment, updating the equipment details, searching and viewing all the equipment in the database.

Method	Use
<i>view_equipment()</i>	Viewing a specified equipment in the database.
<i>request_equipment()</i>	Inserting a request for equipment into the database
<i>add_equipment()</i>	Adding a new equipment in the database
<i>delete_equipment ()</i>	Deleting an equipment from the database
<i>search_equipment()</i>	Searching for an equipment in the database
<i>update_equipment()</i>	Updating details on an equipment in the database
<i>view_all_equipment()</i>	Viewing all the equipment in the database

Table 4.4.2. Equipment class

4.4.3 IT Support Class

This class handles the IT support requests.

Method	Use
<i>view()</i>	View a specified support request
<i>request()</i>	Inserting a new request in the database
<i>update()</i>	Updating details on a particular request
<i>search()</i>	Searching requests made
<i>respond()</i>	Sending an email to the sender of the support request

Table 4.4.2. IT support class

4.4.4 Database connection class

This class handles the connection to the database and is required by all the other classes that have operations that include the database. The class holds the name of the server, username, password and the database name for the application.

Method	Use
<i>getdb()</i>	Establish a connection to the database
<i>query()</i>	Execute a query

Table 4.4.4. Database connection class

4.4.5 User Class

The user class has methods for adding, deleting and updating the details of a user and allowing the user to login if the login credentials entered are correct.

Method	Use
<i>add_user()</i>	Adding a new user in the system
<i>delete()</i>	Deleting a user
<i>login()</i>	Allowing the user to login and authenticating them accordingly
<i>getUser()</i>	Retrieving information on a particular user
<i>all_users()</i>	Viewing all the users
<i>update-User()</i>	Updating information on a particular user

Table 4.4.5. User class

4.5 Interfaces

Some of the interfaces that the user will interact with are discussed and shown below:

4.5.1 Dashboard Landing Page

In general, the dashboard has the functionality that is specifically for the admin member. It allows the admin member to view notifications at a glance and to see the trend

of requests for each category: IT support, IT Equipment and Stationery, on a graphical view. The figure below shows the badges on the landing page of the dashboard which informs the admin the number of pending requests, requests made, number of lost items, etc.

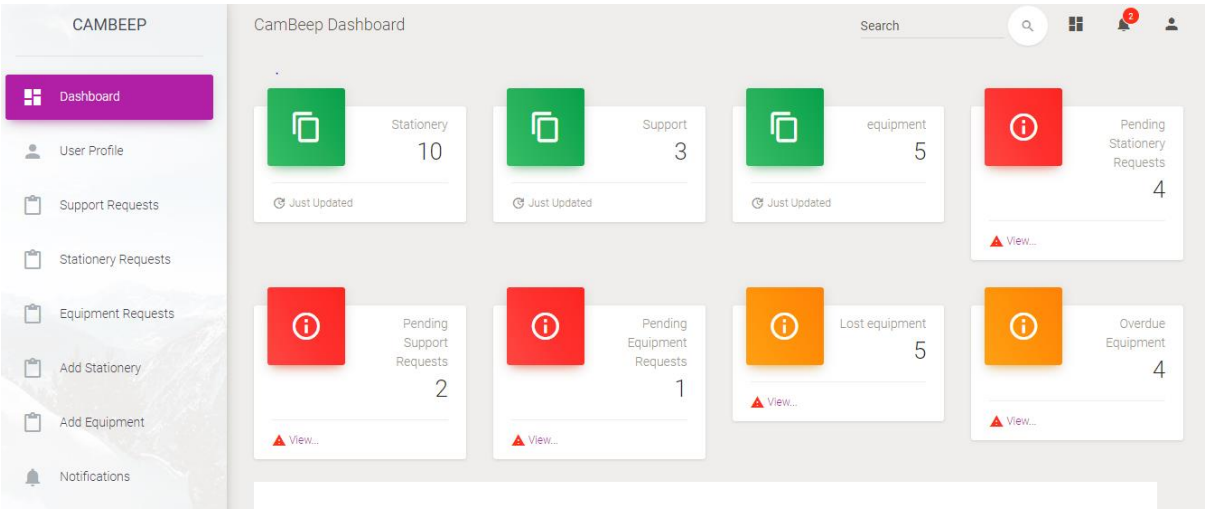


Figure 4.5.1.1.Dashboard badges

The Admin can also have a graphical view of the number of stationery and equipment available as shown below;

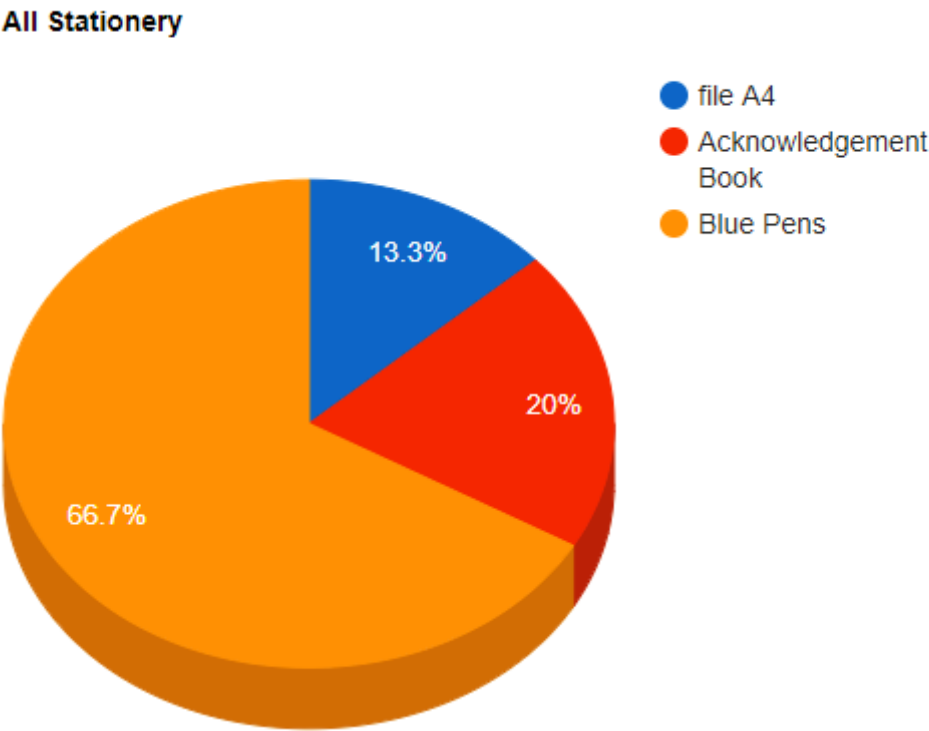


Figure 4.5.1.2. Dashboard sample pie chart

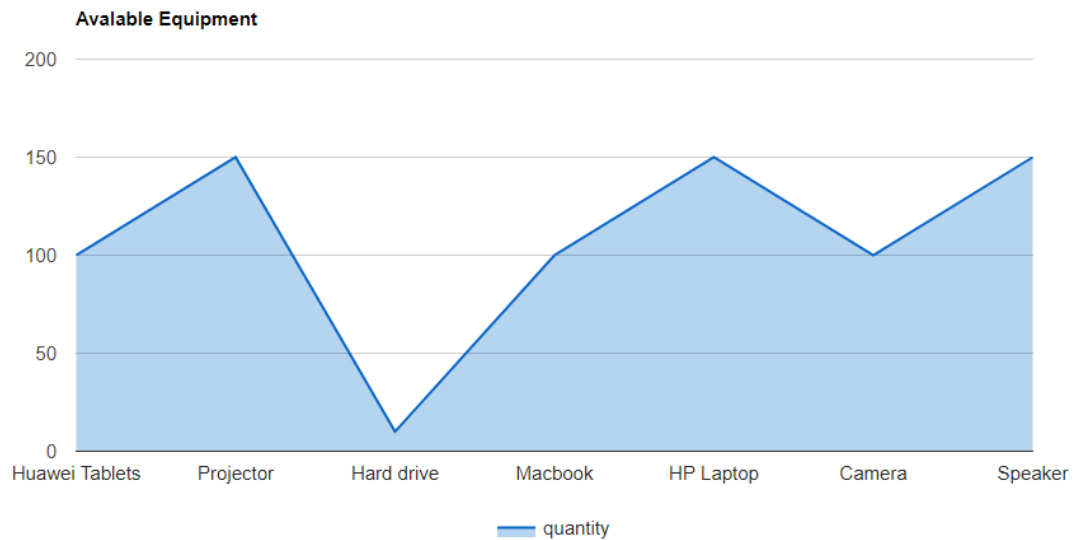


Figure 4.5.1.3. Dashboard sample graph

4.5.2 User Registration

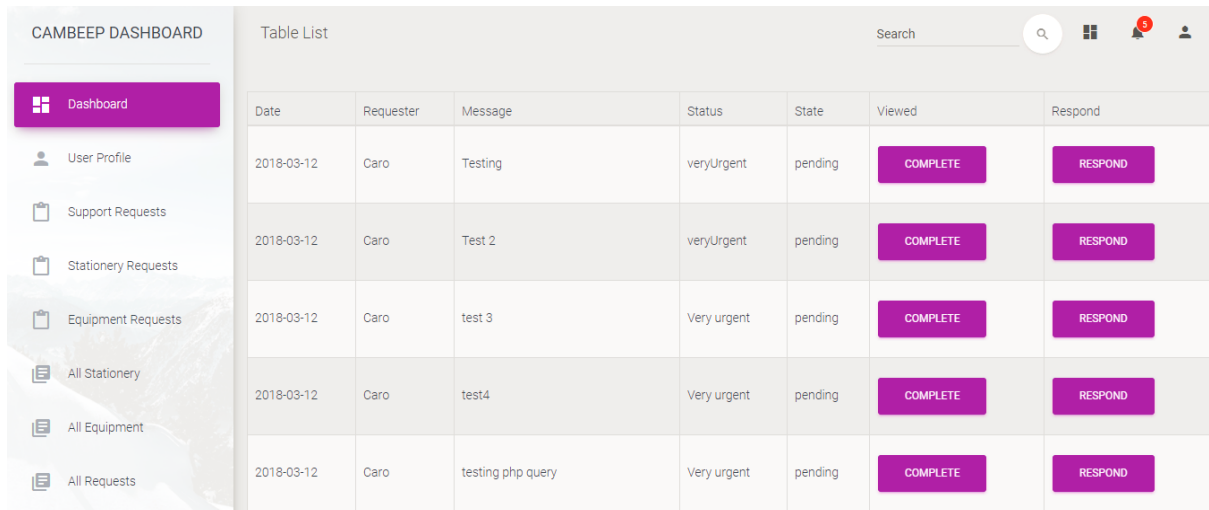
The admin will have to use the dashboard to register users. No user is able to use the application unless they have been registered by the admin. The admin updates the user credentials or registers the user by entering their credentials in an interface shown below:

The screenshot shows the 'CAMBEEP' dashboard interface. On the left is a sidebar menu with options: Dashboard (selected), User Profile, Support Requests, Stationery Requests, Equipment Requests, All Stationery, All Equipment, and All Requests. The main content area is titled 'User Profile' with a sub-header 'Add a new user'. It contains a registration form with the following fields: First Name, Last Name, Username, Password, Category, Role, Email, and District. An 'ADD PROFILE' button is located at the bottom right of the form.

Figure 4.5.2. User registration

4.5.3 Support Requests

The admin can view the support requests, respond to each request by sending an email to the user through clicking the respond button or mark the request as complete; this clears the request off the table. The view of the support requests table is as shown below;



Date	Requester	Message	Status	State	Viewed	Respond
2018-03-12	Caro	Testing	veryUrgent	pending	COMPLETE	RESPOND
2018-03-12	Caro	Test 2	veryUrgent	pending	COMPLETE	RESPOND
2018-03-12	Caro	test 3	Very urgent	pending	COMPLETE	RESPOND
2018-03-12	Caro	test4	Very urgent	pending	COMPLETE	RESPOND
2018-03-12	Caro	testing php query	Very urgent	pending	COMPLETE	RESPOND

Figure 4.5.3. IT support requests

4.5.4 Stationery Requests

The table of the Stationery Requests allows the Admin to view all the requests that have been made and not yet cleared. The admin can clear the request from the table after dispatching the requested stationery and by clicking the button of the particular request and indicating the remainder if the user could not get the quantity that they requested for. The table is as shown below;

Dashboard	Date	By	Item	Quantity	Collection_by	Rem	Clear
User Profile	2018-04-15	Bhebe	file A4	23	Bhebe	<input type="text"/>	<button>CLEAR</button>
Support Requests	2018-04-15	Bhebe	file A4	100	Bhebe	<input type="text"/>	<button>CLEAR</button>
Stationery Requests	2018-04-15	Bhebe	file A4	100	Bhebe	<input type="text"/>	<button>CLEAR</button>
Equipment Requests	2018-04-15	Bhebe	file A4	100	Bhebe	<input type="text"/>	<button>CLEAR</button>
Add Stationery	2018-04-15	Bhebe	file A4	100	Bhebe	<input type="text"/>	<button>CLEAR</button>
Add Equipment	2018-04-15	Bhebe	file A4	100	Bhebe	<input type="text"/>	<button>CLEAR</button>
Notifications	2018-04-15	Bhebe	file A4	100	Bhebe	<input type="text"/>	<button>CLEAR</button>

Figure 4.5.4. Stationery dispatch table

4.5.5 Equipment Requests

The Equipment Requests table works in a similar way to the Stationery Requests table. The figure below shows how the interface will appear if no request was made;

CAMBEEP DASHBOARD	Table List
Dashboard	you have no records
User Profile	
Support Requests	
Stationery Requests	
Equipment Requests	
Add Stationery	
Add Equipment	
Notifications	

Figure 4.5.5. Equipment requests view

4.5.6 Notifications

The dashboard also allows the admin to view notifications. Notifications are generated if there is one or more requests made for stationery, equipment or support. They are also made if any of the equipment is running low.

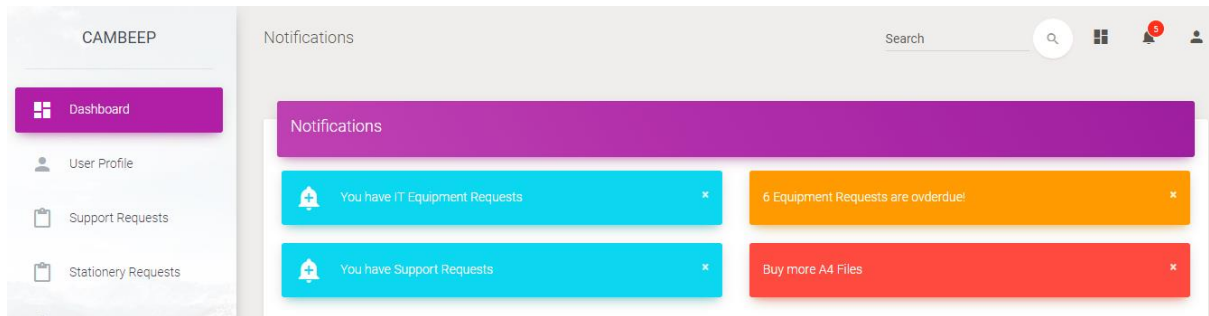


Figure 4.5.6. Dashboard Notifications

4.5.7 Web App Interface

The web application has a simple interface with a screencast to guide users on the homepage. The navigation bar has the links to the stationery, IT support, equipment and the login and logout functionality.

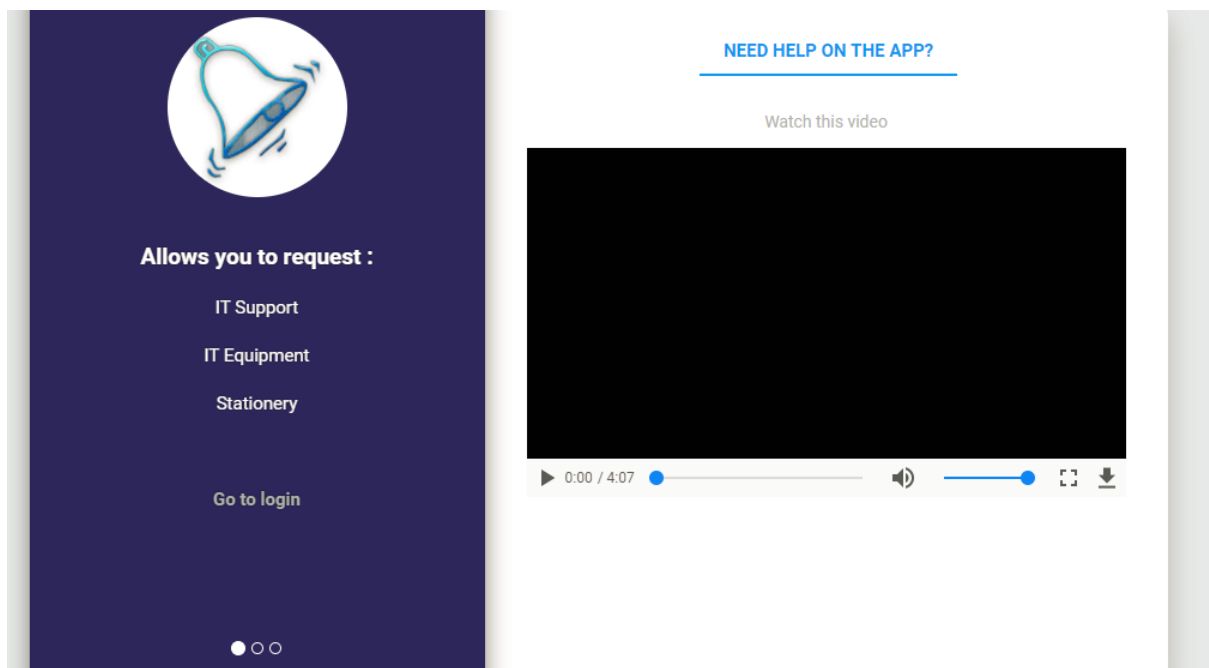


Figure 4.5.7.1. Landing Page

The interface below is the login page of the web app. The user can login with their username and password.

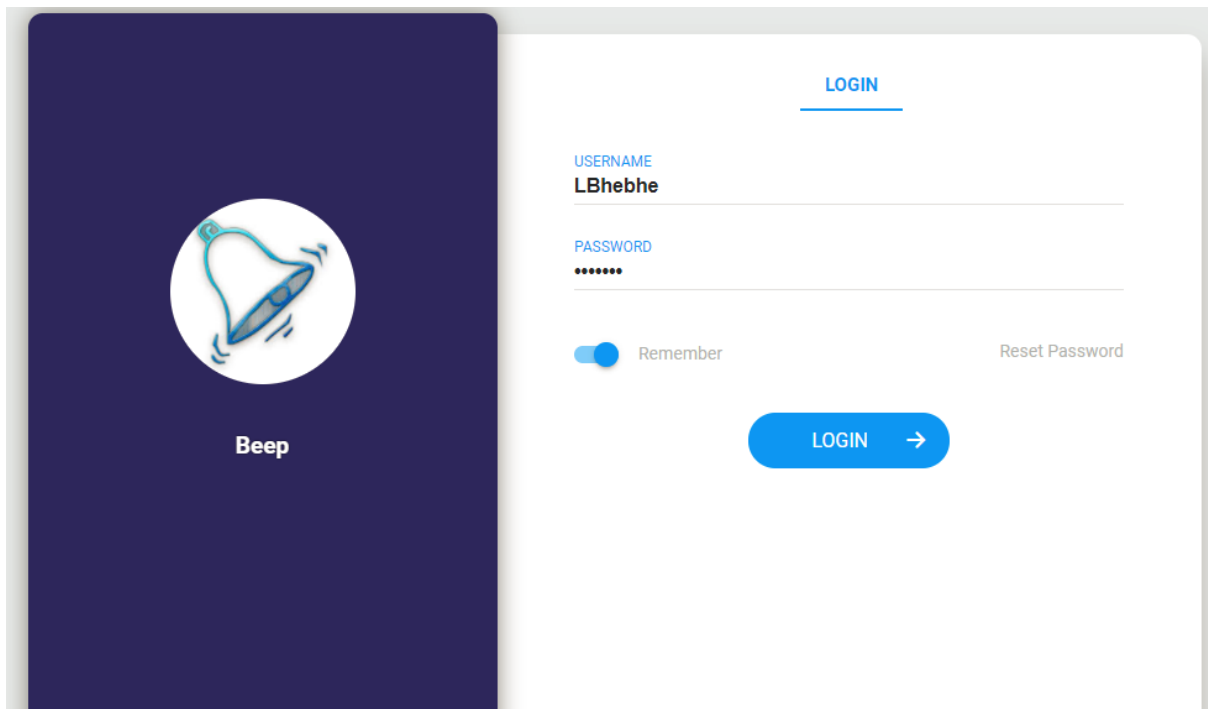
The image shows a web application login page. On the left is a dark blue vertical panel. In the center of this panel is a white circular icon containing a blue bell with motion lines, and the word "Beep" is written in white text below it. To the right of this panel is a white login form. At the top of the form, the word "LOGIN" is written in blue and underlined. Below this, there are two input fields. The first is labeled "USERNAME" in blue, with the text "LBhebhhe" entered. The second is labeled "PASSWORD" in blue, with a series of dots representing a masked password. Below the password field, there is a "Remember" toggle switch which is currently turned on (blue), and a "Reset Password" link in a lighter blue font. At the bottom of the form is a blue rounded rectangular button with the text "LOGIN" and a right-pointing arrow.

Figure 4.5.7.2. User login

The stationery link on the navigation bar takes the user to a page where they can request for stationery or view the status of their requests.

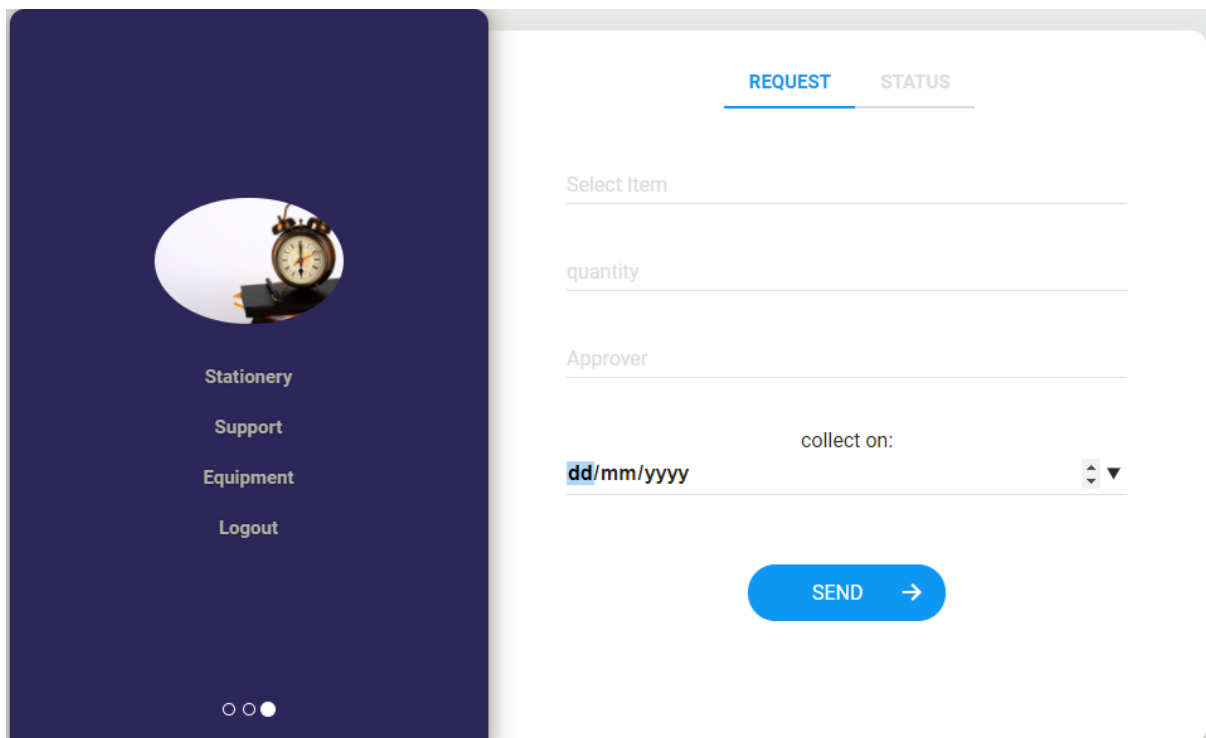


Figure 4.5.7.3. Stationery request interface

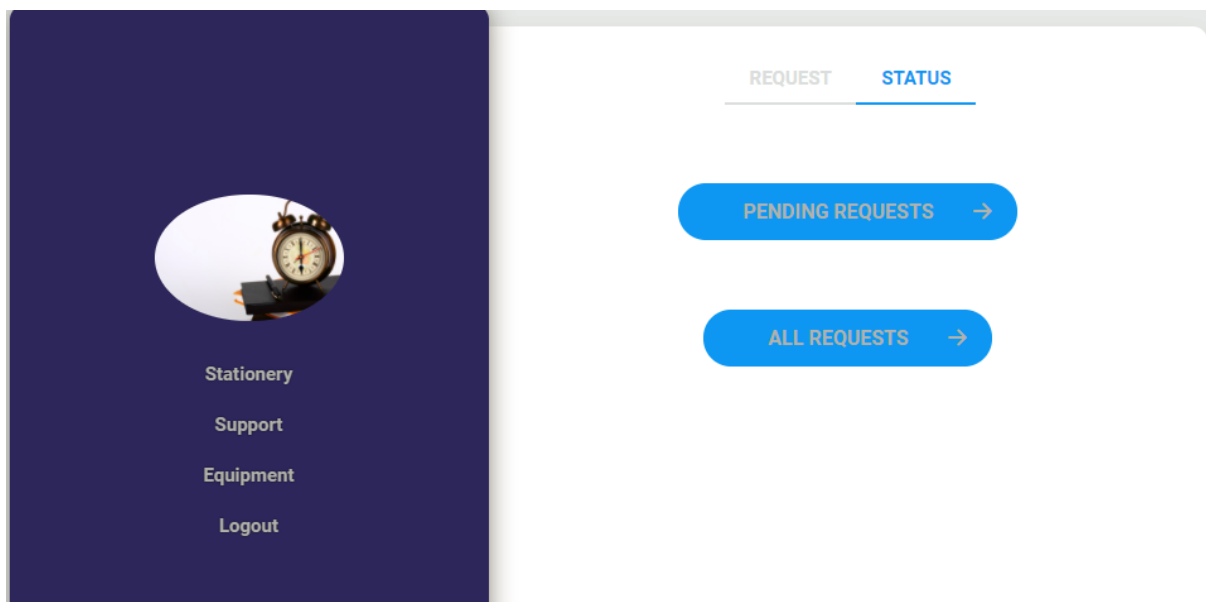


Figure 4.5.7.4. Request status

4.5.8 Mobile App Interface

The Figure below shows the interface of the mobile version.

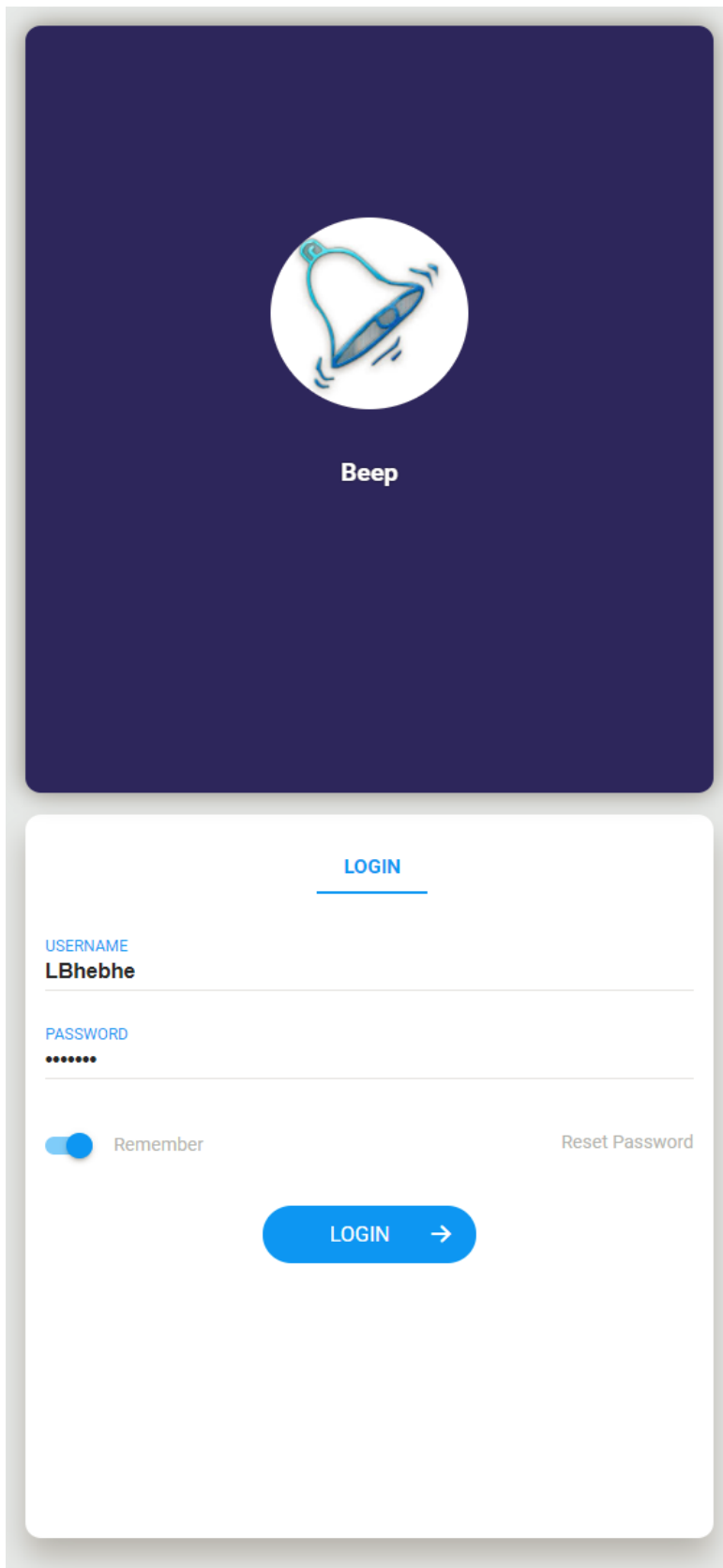


Figure 4.5.8. *Mobile app interface*

4.6 Sequence Diagram

The sequence diagram shows the flow of the system when a user completes the login form. The form sends the data to a PHP file through Ajax. The PHP file connects with the database to verify if the user is valid and responds. If the user is valid, the user is redirected to the home page based on their credentials else they will be prompted to enter valid credentials. The other forms in the application for making requests also have the same flow in handling the data given by the user and communicating with the database.

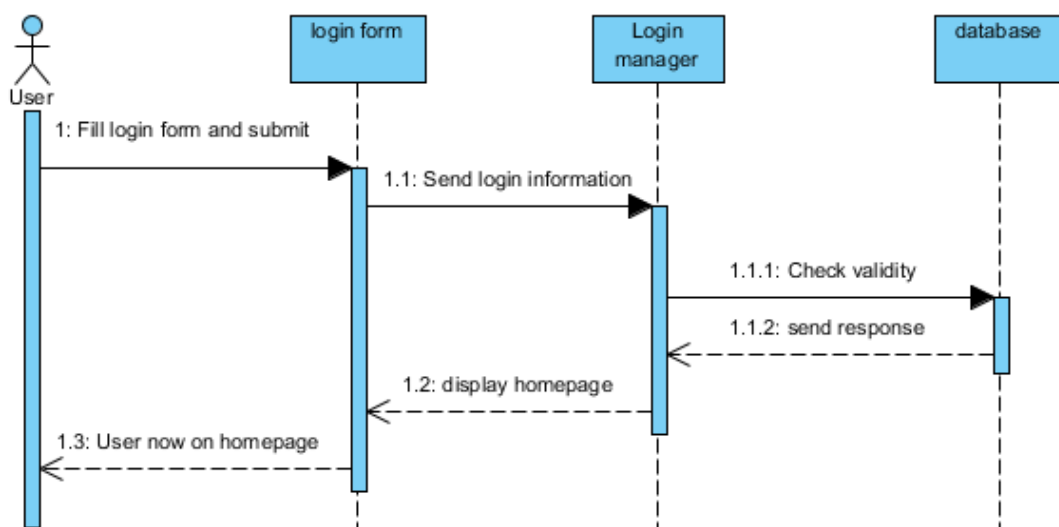


Figure 4.6. Sequence diagram

4.7 Code Snippets

To enable an easy conversion of the web app to a mobile app, there was no direct communication between html and PHP. As highlighted above, Ajax was used to send data from the html forms to PHP. This snippet below shows an example of this for the login functionality.

```

function loginAsUser(){

    var usern= document.getElementById('username').value;
    var pass= document.getElementById('password').value;

    if (window.XMLHttpRequest) {
        xmlhttp = new XMLHttpRequest();
    } else {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            alert(this.readyState);
            if (this.responseText == "code1"){
                window.location.href = "ITadminHome.php";
            }
            else if(this.responseText == "code2") {

                window.location.href = "generalHome.php";
            }
            else if(this.responseText == "code3") {
                window.location.href = "adminHome.php";
            }
        }
    };
    xmlhttp.open("GET", "http://localhost:81/cambeep/model/cambeeplogin.php?username="+usern+"&password="+pass, true);
    xmlhttp.send();
}

```

Figure 4.7.1. Code snippet 1

The following snippet shows how the code dynamically draws the charts using data from the database. The JavaScript shown is included in a PHP script which calls the database class to connect to the database and retrieve the data to be used for drawing the charts.


```

<script type="text/javascript">

    google.charts.load('current', {'packages':['corechart']});

    google.charts.setOnLoadCallback(drawChart);
    function drawChart(){
        var data = new google.visualization.DataTable();
        var data = google.visualization.arrayToDataTable([
            ['item_name','quantity'],
            <?php
                while($row = mysqli_fetch_assoc($areault)){
                    echo "[".$row["item_name"]."', '.$row["quantity"]."',";
                }
            ?>
        ]);

        var options = {
            title: 'Avalable Equipment',
            curveType: 'function',
            legend: { position: 'bottom' }
        };

        var chart = new google.visualization.AreaChart(document.getElementById('areachart'));
        chart.draw(data, options);
    }

</script>

```

Figure 4.7.2. Code snippet 2

The snippet below shows part of the configuration file of the mobile version of the application. Important details like the App ID shown on the snippet were changed for security purposes.

```

<?xml version='1.0' encoding='utf-8'?>
<widget id="camBeepApp" version="1.1.0" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://
cordova.apache.org/ns/1.0">
  <name>CamBeep</name>
  <description>
    An application to send and receive notifications on inventory movement promptly.
  </description>
  <author email="leebhebhe002@gmail.com" href="lindabhebhe@wordpress.com">
    Apache Cordova Team
  </author>
  <content src="generalHome.php" />

  <access origin="*" />
  <allow-intent href="http://*/*" />
  <allow-intent href="https://*/*" />
  <allow-intent href="tel:*" />
  <allow-intent href="sms:*" />
  <allow-intent href="mailto:*" />
  <allow-intent href="geo:*" />
  <platform name="android">
    <allow-intent href="market:*" />
  </platform>
  <platform name="ios">
    <allow-intent href="itms:*" />
    <allow-intent href="itms-apps:*" />
  </platform>
  <engine name="browser" spec="^5.0.1" />
  <engine name="android" spec="^6.3.0" />
  <plugin name="cordova-plugin-whitelist" spec="^1.3.3" />
  <plugin name="cordova-plugin-camera" spec="^3.0.0" />
</widget>

```

Figure 4.7.3. Code snippet 3

Chapter 5: Testing

During and after implementation, testing was done to ensure that the application was working as expected. Development, usability and acceptance testing were the forms of testing employed. Testing ensured that the software meets all the requirements and helped the developer to identify areas that required corrections.

5.1 Development Testing

During the implementation stages of the software, development testing was employed. This involved testing the functionality continually as the implementation progressed. This testing comprised of unit, component and system testing.

5.1.1 Unit Testing

Each function was tested before being integrated with other functions in the entire application. Unit testing involved searching for all possible loopholes in a particular functionality. Checks were made to see if the system works in instances where duplicate values were to be inserted into the database and when an attempt to insert wrong inputs was made or submitting a form without all the required data.

The figure below shows an instance where the user clicked the button to add a new account in the system without completing the details.

The screenshot shows the 'User Profile' registration form in the CAMBEEP system. The form is located on the right side of the interface, with a sidebar on the left containing navigation links. The form fields are as follows:

- First Name:** Input field with a red prompt above it.
- Last Name:** Input field.
- Username:** Input field.
- Password:** Input field.
- Category:** Input field.
- Role:** Input field.
- Email:** Input field.
- District:** Input field.

A purple button labeled 'ADD PROFILE' is positioned at the bottom right of the form. The sidebar on the left includes the following items:

- Dashboard
- User Profile
- Support Requests
- Stationery Requests
- Equipment Requests
- Add Stationery
- Add Equipment
- Notifications

Figure 5.1.1. Registration form validation

For every input that is required the user receives a prompt in red colour to indicate the need to complete the field. All the SQL queries were tested before integrating them in the system to ensure that they retrieve the correct data as required. After adding the queries to the entire code, the functions were also tested to check if the queries were working correctly within the code.

5.1.2 Component Testing

The units of the system were combined and tested as a group to identify any defects between the modules. This was to ensure that the units worked efficiently when implemented together. The units were tested in three distinct groups which were the stationery management, IT support and IT equipment modules. For each module, the functions to add, view requests, update, approved and delete request were tested. Hence, more errors were addressed in unit testing before component testing. The only error was discovered and fixed at this stage was the failure of the system to authenticate correctly the IT Admin and the Executive Admin to their respective interfaces.

5.1.3 System Testing

The system was tested as a whole after completing unit and component testing. In this testing, the developer gave the system to colleagues who had not played any role in the development of the system. These testers tracked if the system met both the non-functional and functional requirements. At this stage, the testers reported that the system was relatively slow due to the lengthy queries that were running one after the other for some functionality. Some queries were revised to make them more efficient which increased the speed of the system.

5.2 Usability Testing

The system was tested by users who are not software developers to see if the system is user friendly. The users gave feedback on the flow and navigation of the system. They ranked the application on a scale of 1-5 (5 for best, 1 for worst) in various qualities as shown on the table below;

Quality	Description	Average rank
Learnability	How easy it was for them to accomplish a task the first time they encountered the system	5
Memorability	How effectively they used the system after they have stayed for a while without using it.	5
Errors	The severity and number of errors that the user made.	4.5
Efficiency	How fast they used the system after gaining experience on the system.	5
Satisfaction	How satisfied they were using the system in general.	5

Table 5.2. Usability testing

From the feedback on usability testing, the testers highlighted that the system was usable and only needed improvement in making the forms more interactive. The ranking on the severity and number errors made was lower than the other rankings because initially

users had to type in the names of the approvers but this was fixed. Users can now select the names of the approvers from a drop down menu auto generated from the database. This was fixed by making the user confirm the details on the request form before submitting it and giving them the chance to edit if they had made a mistake.

5.3 Acceptance Testing

To determine if the system was ready for release, acceptance testing was conducted. Since some changes were made during development, the staff at Camfed had the chance to verify if the produced application was in line with the business needs. At this stage, they realised that there was a need of separation of interface on the dashboard such that the general user IT Admin officer only has access to IT requests.

Chapter 6: Conclusion and Recommendations

6.1 Conclusion

Through working on this project, the developer was able to acquire more skills on project management. Working towards meeting the deadlines of the project and the expectations of the client enabled the developer to gain more skills in time management and prioritizing tasks. The developer was also able to see the interlink of the courses studied such as Research Methods, Mobile Web Development, Web Technologies, Software Engineering, Databases, Algorithm Design and Analysis, Human Computer Interaction and all other programming courses. Employing the knowledge from all these courses helped to produce a complete project. Apart from having a hands on application of the knowledge acquired in the four years of study, the developer was able to gain more research skills especially in acquiring user needs for a software.

The system developed met most of the requirements as expected. After testing and fixing the errors that were discovered, the following functions were working as required;

- ✓ A user can be added.
- ✓ Users can log in and be authenticated and redirected correctly.
- ✓ Users send requests for IT support, stationery and equipment on the web App and mobile devices.
- ✓ The admin can view all requests.
- ✓ The admin can check out completed requests.
- ✓ The admin can send emails in response to the requests made from the system.
- ✓ The system can prompt users with overdue equipment to return.
- ✓ The Admin can be prompted on the stationery and equipment running low.
- ✓ The status of an equipment is known at each moment whether it is on loan, lost or in the office.

6.2 Challenges

The main challenge faced in the project was the decision to change the framework to use during the implementation stage which gave the developer limited time to complete the project. This deprived the developer the time to complete the most important parts of the project and explore other innovations in inventory systems like incorporating augmented reality and virtual reality to optimise the user interaction as suggested by the literature review.

6.3 Future Work

- The system should be developed further to enable the admin to print a report in a pdf format for auditing purposes. The report should comprise the names and quantity of the items that were requested and those still in stock.
- The developer will work with the IT Administrator at Camfed to install and configure the system.
- It is also recommended that the system should be constantly upgraded since the technologies in web development continue to advance.

References

- Camfed. (2017, 11 01). Retrieved from Camfed: <https://camfed.org/>
- Karat, J. (1997). Evolving the scope of user-centered design. *Magazine Communications of the ACM*, 33-38.
- Kulkarni, R., Padmanabham, P. P., Sagare, V., & Maheshwari, V. (2013). Usability evaluation of PS using SUMI (Software Usability Measurement Inventory). *Advances in Computing Communications and Informatics* (pp. 1270 - 1273). Mysore: IEEE.
- Lo, Y.-C., Liu, X.-W., & Yang, S.-H. (2017). A personalized learning system with smart interaction. *Consumer Electronics* (pp. 1-7). Taiwan: IEEE.
- Phinho, R., Sousa, A., & Restivo, A. (2010). Applying usability principles to the design of a web interface for the “iLab - inventory manager for electronics laboratory”. *Iberian Conference* (pp. 1-6). Santiago de Compostela: IEEE.
- Shackel, B. (1991). *Human Factors for Informatics Usability*. New York : Cambridge University Press.
- Tech. (2015, October 21). Retrieved March 31, 2016, from Zytrax: <http://www.zytrax.com/tech/lang/ruby/ror.html>

Appendices

Appendix A

Camfed Zimbabwe Letter of Approval



When you educate a girl
...everything changes

Ashesi University College
PMB CT3 Cantonments,
Accra,
Ghana

August 18th 2017

Dear Linda

Re: Letter of approval

I am pleased to inform you that your request to implement a project on how to handle stationery requests, IT equipment and IT Support requests in Camfed Admin department has been approved. Please find attached a copy of the consumables page to help you in your implementation process.

We are looking forward to receiving feedback from you on the progress as you go along.

Yours faithfully,

Faith Nkala
National Director

Sample Request Form

[illegible]

Goods received note

48