



ASHESI UNIVERSITY COLLEGE

**A USER-CENTRED APPROACH TO A BUSINESS MANAGEMENT
APPLICATION FOR SMALL BUSINESS OWNERS**

APPLIED PROJECT

B.Sc. Management Information Systems

Rahila Sarah Sule

2016

Running head: BUSINESS MANAGEMENT APPLICATION FOR SMALL BUSINESS
OWNERS

ASHESI UNIVERSITY COLLEGE

**A User-Centred Approach to a Business Management Application for
Small Business Owners**

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
Management Information Systems

Rahila Sarah Sule

April 2016

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 thesis laid down by Ashesi University College.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

Acknowledgements

I would first like to thank my applied project supervisor, Mr Aelaf Dafla, for his assistance and dedicated involvement through the process of this project. He consistently allowed this paper to be my own work but provided guidance and substantial feedback when needed.

I would also like to thank Mr Kwadwo Osafo-Mafo whose doors were always open to me when I needed help during the course of this project. Also, I am grateful to Mr Stephan Nwolley and his team at Nola for help with achieving a user-centred design for this project.

Additionally, I must express my profound gratitude to Safiya Sule, Umar Sule and my parents for their moral support throughout my years of study. I would also like to thank my friends, Andrew Bimpong, Wendy Amartey, Ameh Dan-Ugo and Olamide Bello for their continuous support and encouragement through the process of writing this applied project. This accomplishment will not have been possible without all aforementioned individuals.

Finally, I would like to thank God for strength, wisdom and inspiration throughout my years of study and through the process of writing this dissertation.

Abstract

Due to the expected changes in the business industry, businesses are placing greater emphasis on innovation. Most businesses are taking advantage of technology to achieve innovation. As such, a number of business tools have been developed to help business owners manage their businesses. However, these tools are not utilised by several well-educated small business owners because they claim to prefer their current methods. This is because they believe that paper or a simple Excel sheet is less complicated than an application that will stand as a challenge for them and disconnect them from their business.

This project employs a user-centred design to present a simple web application to help small business owners manage their sales, customer, inventory and receivables. The application has an integrated notification system that keeps the business owner updated on activities occurring within their business. The application follows design principles in order to achieve a simple, useful and usable application for the small business owners who are novice technology users.

After conducting several tests, it can be concluded that the solution is a highly usable and useful application for its target users. Usability tests revealed that the application saves time and effort and hence increases operational efficiency for small businesses and their owners. By adopting this solution and implementing further features, small business owners can be kept abreast with their businesses and can sustain their businesses over the years to come.

Keywords: retail, small businesses, small business owners, technology, business processes, business process automation, user-centred design, notification system

Table of Contents

Declaration.....	iii
Acknowledgements	iv
Abstract.....	v
Table of Contents	vi
List of Tables	viii
List of Figures.....	ix
Chapter 1: Introduction	1
1.1 Introduction	1
1.2 Background	2
1.3 Motivation	5
1.4 Objectives	6
1.5 Outline of Dissertation	7
Chapter 2: Design	8
2.1 Design Overview	8
2.2 Requirements Elicitation and Analysis	12
2.2.1 Requirements Discovery	12
2.2.2 User Requirements.....	17
2.2.3 System requirements.....	18
2.3 System Architecture	20
2.4 Database Architecture.....	23
Chapter 3: Implementation.....	25

3.1 Implementation overview	25
3.2 Implementation	25
3.3 Technology, Tools and Platform	34
Chapter 4: Testing	36
4.1 Overview	36
4.2 Development Testing	36
4.2.1 Unit Testing	36
4.2.2 Component Testing	37
4.2.3 System Testing	39
4.3 Release Testing	40
Chapter 5: Conclusion	46
5.1 Overview	46
5.2 Challenges	46
5.2.1 Integration of Facebook as a Notification System	46
5.2.2 Ensuring Usability of System	47
5.3 Future Work	47
5.4 Conclusion	48
References	50
Appendix	52
Appendix A	52
Appendix B	54
Appendix C	56
Appendix D	57
Appendix E	61

List of Tables

Table	Description	Page
Table 3.1	Technologies used for implementation	34
Table 4.1	System testing	39
Table 4.2	Release testing	41
Table 4.3	User testing	43
Table D1	Plan for requirement analysis	57

List of Figures

Figure	Description	Page
Figure 1.1	Technologies which are used today in retail industry	3
Figure 2.1	Use case diagram for business owner	16
Figure 2.2	Use case diagram for employee	17
Figure 2.3	System architecture	20
Figure 2.4	Database architecture	24
Figure 3.1	Homepage	26
Figure 3.2	Shortcut on homepage	27
Figure 3.3	Sequence diagram for add sale	28
Figure 3.4	PHP and query for customer and employee dropdown list	29
Figure 3.5	SQL query for adding sale	30
Figure 3.6	View for adding sale	30
Figure 3.7	Various product views	31
Figure C1	Retail sales GDP contribution, 2013	56

Chapter 1: Introduction

1.1 Introduction

With the prevalent nature of technology in the business world and its increasing importance to the overall operation of many businesses, technology has become a key input used to achieve business goals such as increased revenue and lower costs. This is due to the fact that most businesses, specifically the retail and wholesale industry, thrive on margins.

The use of technology such as Radio Frequency Identification(RFID), smart display systems, video analytics and inventory management systems have made it possible to maximize margins. In addition, technology has made remarkable improvements in customer experience, supply chain management, operations management, loss prevention and inventory management (Mathaba, Dlodlo, Smith, & Adigun, 2011). However, managing different business and operating functions to create value for customers can be complicated and difficult for business owners; even those that operate small businesses.

Although technology is important there are some small businesses who do not use technology in their business because they are unaware of the advantages coupled with the adoption of technology and thus believe that the methods they currently use in running their businesses are effective and simple. However, these businesses that are not adopting technology are missing out on the rewards from investing in technology.

From interviews conducted with some business owners, it has been observed that the harnessing of technology can be daunting for some business owners. Therefore, this project presents mBridge, a management information software for small-to-medium sized retail businesses. It provides a cost-effective, simple, learnable and usable platform to streamline the key business processes of sales management, customer management, inventory

management and receivables management. This will be done by creating a link between already functioning methods and technology by modifying such methods to integrate technology that makes business operations more competent. Also, mBridge's integration of a notification system is important because unlike similar applications, it will help to keep the business owners abreast with the operations of their businesses.

Nonetheless, just like the integration of information technology in any business, the success of the mBridge application is based on how well it is assimilated by its users. For this reason, simplicity, utility and usability are the key factors considered in developing this application.

1.2 Background

As at 2013, Planet Retail reported that about three-quarter of West African countries attributed approximately fifty percent of GDP to retail sales (as cited in Deloitte, 2015). This is because small businesses constitute a significant percentage of the private sector. The growth of small businesses is therefore very important.

As technology is evolving there are numerous technologies available for use within the retail industry and African small business owners are gradually employing these technologies in their businesses. As illustrated in Figure 1 below, there are Point-Of-Sale scanning systems for more efficient customer checkout, inventory management systems for better inventory tracking and electronic invoicing systems that automate invoice processing and reduce paper waste amongst others, available for use within the retail industry. Business automation processes are also being utilised to improve business efficiency by reducing time spent on inventory reconciliation.

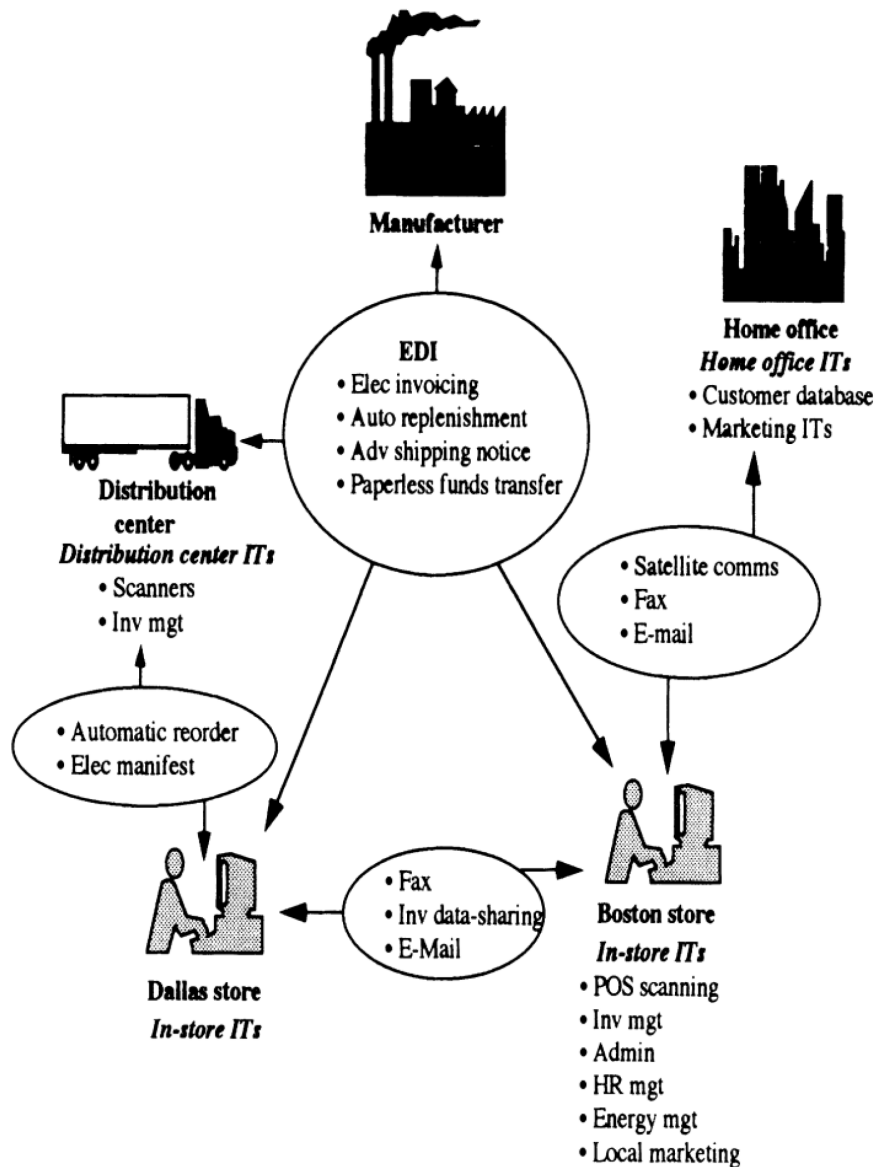


Figure 1.1 - Technologies which are used today in retail industry

Yet, there are some business people, including some who are well-educated, who do not use technology for business for many reasons such as fear of change, privacy issues, security reasons or just because they are uncomfortable with it. However, almost everything

we use (such as laptops, mobile phones, ATMs) was once new technology to us but we learned to embrace it as an integral part of our daily lives in this information age. This is because new technology seems complex due to pre-conceptions in our minds. Thus, some people become uncomfortable and frustrated because they are afraid of change, privacy or security. However, an observation of how this group of people perform daily tasks could help in developing software or systems that will help make their activities more efficient. This is because it has been noted that this group of people classify familiar environments as the only safe environments.

Business processes are valuable organizational assets that offer businesses the opportunity to organize the way they conduct business. As such the efficiency and productivity of such processes are potential challenges for businesses. Different organizations have created applications to help businesses manage their processes efficiently however, these applications are still not effective enough for mBridge's target users because they are mostly performance based rather than user-centred. Losoftware's RightControl, inflow and Carta are business management systems for small businesses that were reviewed for this project.

RightControl is an inventory management software for small businesses (Losoftware, 2016). It has a simply designed user interface and has good functionality, however, it lacks a good notification system and therefore still gives room for a disconnect between the business owner and the business. It also has quite a number of unessential features that require the user to input a lot of data which can be overwhelming to small business owners, therefore making it seem like a complex application.

inflow is an inventory management software for small and medium sized businesses to help track inventory, reorder stocks, fill customer orders and create customized reports

(Archon Systems, 2016). The software is an easy to use software that handles basic inventory management tasks and a few other tasks. Conversely, there are some drawbacks to it. The application is a desktop application meaning it cannot be accessed remotely by business owners. It also only works on Windows computers, which means it is not OS-neutral, so it cannot be used on operating systems like Mac OS X except by using a virtual machine that can access a Windows environment. This will eventually lead to a poor performance of the application in the market as there has been a move away from Windows-specific applications to OS-neutral applications, according to Michael Silver (Zachary, 2009). Furthermore, while inFlow can be started out with a free version, it requires up-front payment that is at a steep price.

Carta is a cloud-based order and inventory management system for small businesses. It has a simple and user-friendly user interface and the application employs some automation of business processes (Carta, 2016). However, as with most similar systems, Carta does not offer a well integrated notification system for the business owner.

In a nut shell, although these applications have good functionality, they have some drawbacks that make them unsuitable and inefficient for this project's target users. mBridge's notification system is a functionality that sets it apart from similar applications.

1.3 Motivation

Change has become a trademark for success in the business world. In fact, the pace of change is so intense that it is critical for the future success of a business. In the short run, the impact of not using technology in businesses might seem negligible but in the long run the

impact could be negative and high. This is due to the fact that it is impossible to predict what the vastly growing change in technology will lead to in the next few years.

According to research by the Economist Intelligence Unit, business leaders are expecting evident change to the business industry in the future due to the impact of technology. The research reveals that business leaders see a shift towards more virtual working environments (Ricoh, 2012). David Mills, Executive Vice President, Operations Ricoh Europe said, “The successful businesses of 2020 will be those that place greater emphasis upon process innovation. The key to building a more agile business that is ready to embrace the changes is to have the right infrastructure in place to manage it successfully” (Ricoh, 2012). Therefore, businesses that employ the use of technology now stand a better chance of survival in the next few years. However, it is evident that technologies by themselves cannot help businesses keep up with changes required for survival and growth within their industry. The business processes being coupled with the technology must be assessed and improved to add real value to the businesses.

Also, as stated by KPMG, robust economic growth relative to the rest of the world will boost Africa’s retail sector over the long term (KPMG Africa Limited, 2015). Hence, in order to stay relevant and run successful businesses, business owners must innovate and harness technology to improve their business operations.

1.4 Objectives

The objective of this project is to create a user-centred web application to provide the following functionality to cater to the effective and efficient operation of small retail businesses. The target user group believe that the current method of using paper to manage

their business processes is adequate. As such the web application aims to change the target user group's perception and close the gap between the target users and technology by offering a tool that is simple, learnable and seamlessly usable. These functionalities include:

- Allow business owners to manage sales, customers, inventory and receivables.
- Provide efficient coordination between a business and a business owner through an efficient and simple notification system.
- Employ business automation processes to reduce complexity in business procedures such as the reconciliation of inventory and sales, the reconciliation of sales and receivables.

1.5 Outline of Dissertation

This report will provide more details about the accentuating aspects of the system and how it operates. Chapter 2 will discuss the design of the system including the requirements elicitation and analysis process. It will also review the architectural decisions and considerations made as well as the database design. Chapter 3 will discuss the implementation of the system. It will review the technology, tools and steps taken to build the entire system. Chapter 4 will review the different tests conducted on the system and the results of the tests conducted. Finally, Chapter 5 will conclude the report and discuss challenges encountered while developing the system. It will also include future work that could expand the functionality of the system.

Chapter 2: Design

2.1 Design Overview

System type

A web application will be used for the business management software rather than a desktop application. This is because of the several advantages that a web application will offer for such a system. A web application will require no installation because it is offered as a service over the internet and hence it will have high availability and accessibility. Therefore, using web applications does not restrict users to one device because it can be used on any device that is connected to the internet. Since all that is needed to use a web application is an internet connection, then it can be concluded that web applications are operating system independent, so no specific operating system is required to run a web application. Therefore, web applications can be used on mobile phones and tablets.

Interfaces

In architecture design it is important to define the components and interfaces between them. Since the web application does not have any designated hardware, it does not need special direct hardware interfaces that require designing and implementation. The hardware connection is managed by the underlying operating system on the web server. The communication between the database and the web application involves operation concerning both reading and modifying data. Communication among the different system parts is important since they depend on each other. However, the means by which the communication is achieved is not vital for the system and is therefore also handled by the underlying operating system for the web application.

Scope

The software is a business management software that is a web application. It will help retail business owners to manage and improve their inventory management, sales recording, debt recording and customer database. The application should be readily available online and can be accessed on a laptop, desktop or tablet.

Business owners will have total administrator access and can administer the system, assign privileges to employees and keep the information accurate. In the case where the software will be used mostly by employees rather than the business owner, the software's good notification system handles the problem of employees dishonestly modifying certain data without the business owner's consent.

Furthermore, the software needs Internet connection to be accessible to users. All system information is maintained in a database, which is located on a web-server. The application has the capability of representing both summary and detailed information on inventory, sales, debtors and customers.

Feasibility Study

- **Economic feasibility:** Based on the resources stated in the requirements plan, the project is feasible because all stated resources are available and can easily be obtained. The benefits of the project outweigh the costs because the major costs involved in development are time and money (phone calls, transportation costs).
- **Technical feasibility:** The small business retail stores currently possess technology such as laptops, desktops or tablets that can be used to operate the proposed software. Based on

interviews conducted, the owners of the businesses, the major users of the software, currently possess the necessary technical expertise required to operate the proposed software. According to the survey of five businesses, they are all currently novice users so they will be able to learn and use an easy software. However, some retail businesses will require some employees to use the software and a few of the employees are beginners so some training will be required.

The software developer currently possess the necessary technology required to develop the proposed software. A laptop and internet access are readily available and Ashesi's server will be used to create dummy data and test entries.

The technology that will be used in development are mature and proven technology, such as a web server and already existing programming languages. These required technology is available and has the capacity to handle the software.

- Legal feasibility: The software does not violate any legal laws and so can be regarded as feasible based on legal requirements.

Product functions

With the web application, business owners and employees will be able to search for items in inventory, specific sales and customers. The result will be based on the criteria the user inputs. The results of the search will be viewed in a list view, depending on what criteria is included in the search. This list view will provide basic attributes as specified by business owners. However, the user will be able to select an item or entry or name to view further information.

Business owners can also add and edit each entry made into the system. In order to achieve its aim of being easy and simple, the software has a reconciliation feature that automatically reconciles inventory based on sales records. It also automatically creates new receivable records when a customer buys on credit.

User characteristics

From interviews conducted during requirements gathering, it has been deduced that there are two types of users that will interact with the system: business owners and employees. Each of these two types of users has different use of the system so each of them has their own requirements.

The business owners are the system administrators and can manage all information about their business pertaining to inventory, sales, debtors and customers. They are managing the overall system to ensure correctness of information within it. The target users are small retail business owners are novice users with university degrees, between the ages of 30 and 50 years old.

The employees can also manage all information on the application. They may either be beginner or novice technology users.

The application is not intended to be used by disabled people.

Constraints

The Internet connection is a constraint for the application since it is a web application. As the application fetches data from the database over the Internet, it is critical that there is an Internet connection for the application to function.

The application will be constrained by the capacity of the database. Since the database is shared by all users of the system, it may be forced to queue incoming requests and therefore may increase the time it takes to fetch data.

Assumptions

A major assumption about the software is that it will always be used on laptops, desktops or tablets that have enough performance. Another assumption is that the web browsers on all laptops, desktops and tablets work in the same way.

2.2 Requirements Elicitation and Analysis

After the feasibility study, the next stage involved working with the system end-users to discover the application domain, the services it should provide and constraints. Hence, the purpose of this section is to describe the steps taken to further understand the problem being solved, derive functions of the software and understand the performance of the software and any constraints.

2.2.1 Requirements Discovery

The first actions taken before gathering requirements from users was establishing the objectives of the software. Similar business management systems, Losoftware's RightControl, inflow and Carta were reviewed and general business processes such as sales recording, inventory management, debt management, customer management, supply chain management and customer service were studied. This was done in order to gain background knowledge of such an application.

Next, information about the system was gathered from end-users.

2.2.1.1 Interviews

Interviews were conducted with five end-users. Before embarking on the interviews, the user characteristics were specified to guide the search for interviewees. The characteristics of the end-users were identified to be:

- individuals between 30 and 50 years of age
- either Ghanaian or Nigerian nationals
- either male or female
- university degree holders
- small or medium retail business owners.

The interviewees were selected based on convenience sampling. A consent form which sought the approval of interviewees was issued to all interviewees. The consent form clearly stated that participation was voluntary and that interviewees may have withdrawn at any time without negative consequences. The interviews were a mixture of closed and open interviews because a set of questions needed to be answered but some questions led on to other important issues that were discussed in a less structured way.

During the interview, questions were asked about the current business processes involved in the daily operations of the retail shops, this led to discussions on systems currently used by the business owners, which were mostly paper or Microsoft Excel. Questions were also asked about the software to be developed. The results of the interviews helped in getting an overall understanding of the businesses the end-users own and how they operate, it also helped in giving an understanding of how the end-users might interact with the

new software and the difficulties that they face with the current systems they use. Also, the interviews gave better insight to why the end-users currently do not use technology in their businesses.

2.2.1.2 Scenarios

To better understand how end-users might interact with the new software, scenarios were used. Different scenarios were used for the interviewing process to better understand the business processes involved in the businesses' daily activities. Additionally, three scenarios were deduced from the interaction with end-users to outline example interaction sessions with the application to end-users. These scenarios helped in giving a clearer understanding of how the application to be developed will be used and a description of what could go wrong. Below is a short scenario that was used during the process of requirements gathering:

Scenario: 2

Prerequisites: Business owner has a valid username and password

At the end of the week, the owner of the store wants to reconcile the sales and the inventory. Rather than checking each receipt in the receipt book, like she did previously, she simply logs in to mBridge on her iPad and navigates to the sales section. She simply checks all sales recorded for the week and checks inventory page to ensure that all reconciliations are accurate. She does not need to manually update her inventory with sales transactions because mBridge does that for her already.

The above scenario and the two other scenarios included in Appendix B were analyzed by first identifying actors, objects and activities. In deducing requirements from the scenario above, the key actor was identified as a business owner. Next, the task the actor is trying to

execute was identified. In this case, the task intended to be performed by the actor was reconciliation of inventory. After identifying the task, the task was analysed based on the parts facilitated by the system and those independent of the system. Due to the implementation of SQL queries that perform business logic for reconciling inventory after every sale is recorded, the task was facilitated by the application and involved some business logic behind the scenes.

All scenarios can be found in Appendix B.

2.2.1.3 Use Cases

Below are use cases of the major interactions between the business owners and employees and the software.

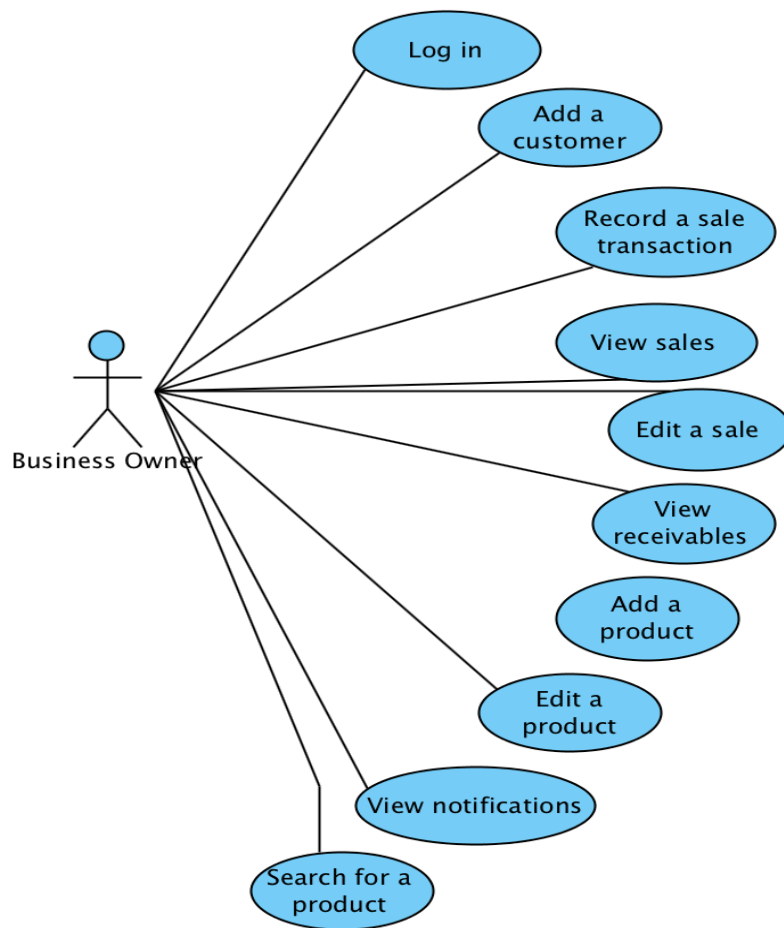


Figure 2.1 - Use case diagram for business owner

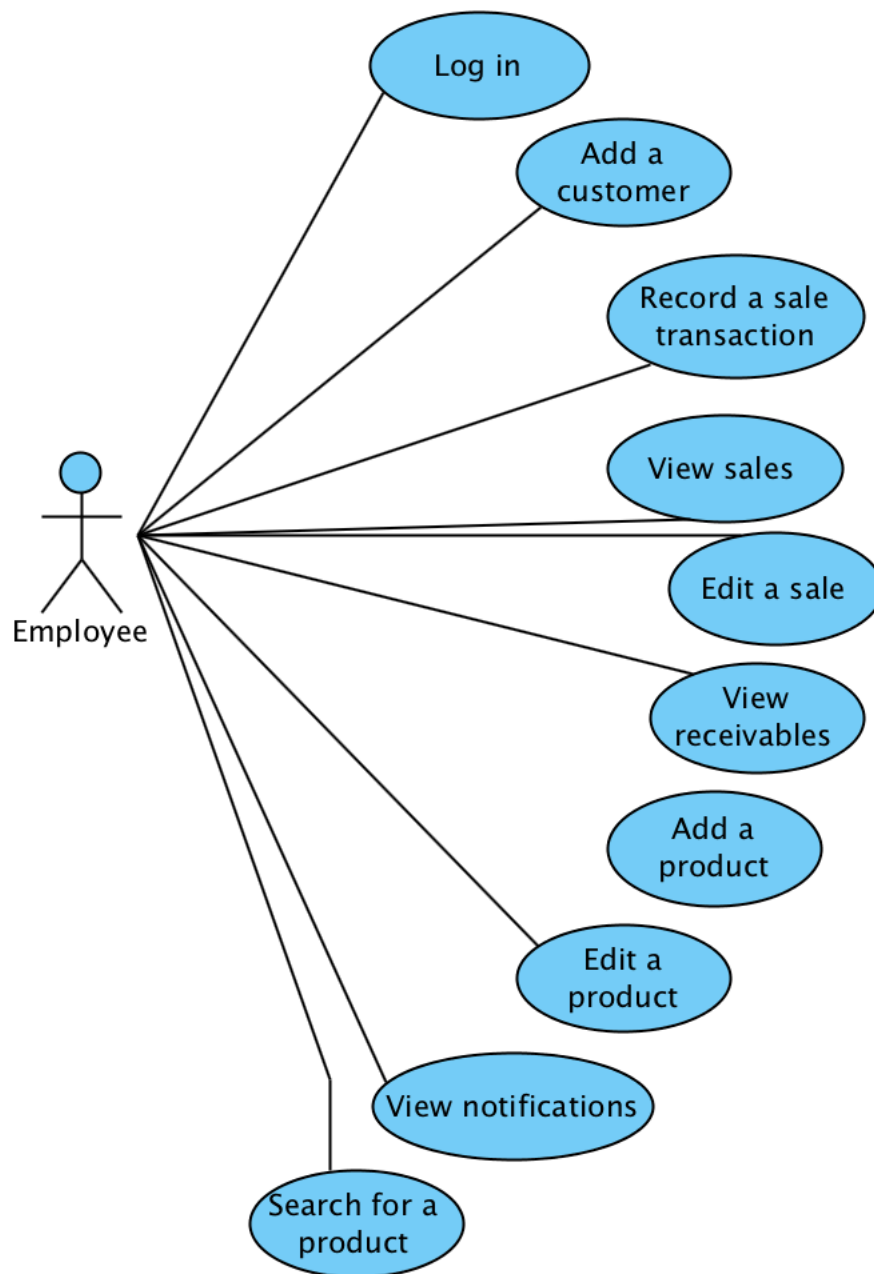


Figure 2.2 - Use case diagram for employee

2.2.2 User Requirements

- After analyzing results from the interviews conducted, the following
- Easily record sales transactions

- Manage inventory of all goods
- Easily locate an item
- Manage and keep track of all debtors
- Easily determine quantity remaining of an item

2.2.3 System requirements

The software's system requirements have been classified into functional and non-functional requirements.

2.2.3.1 Functional requirements

- The following are functional requirements for business owners: Log in, log out, search for item, search for sale, search for customer, add item, edit item, view item, add sale, edit sale, view sale, view receivable, add customer, edit customer, view customer, view notifications, receive notifications.
- The following are functional requirements for employees: Log in, log out, search for item, search for sale, search for customer, add item, edit item, view item, add sale, edit sale, view sale, view receivable, add customer, edit customer, view customer.

2.2.3.2 Non-functional requirements

Non-functional requirements define the required characteristics of the system such as security, operability, performance and usability. The non-functional requirements were defined to specify how well system should operate because they place limitations on the system being developed, the development process and highlight external constraints the

system must satisfy. As such, product and organizational requirements are defined below for the system.

Product requirements

The software should be available to business owners and employees during normal working hours. Unavailability of the business application will lead to the unavailability of vital business information, stopping the flow of information and slowing down or, in some cases, halting business processes. Therefore, the application's downtime should not exceed 4.5 hours per year which is less than the retail industry average of 7.75 hours per year (Boggs, Bozman, & Perry, 2009). In order to ensure confidentiality, the software should be secure and require a valid username and password in order to access software functions. 95% of business owners should be able to use all the system functions after five hours of training without requiring assistance because the system should be learnable and easy to use after a few uses. After this training, the average number of errors made by these novice users shall not exceed three per hour of system use.

Organizational requirements

Business owners and employees should authenticate themselves using their given username and password. System should not allow for deletion of data entries. The software should be upgradeable.

2.3 System Architecture

This section describes the software architecture by providing a comprehensive architectural overview of the software being developed. It captures and conveys the significant architectural decisions which have been made on the software. The software will implement the Model-View-Controller (MVC) architecture as shown in Figure 2.3 below.

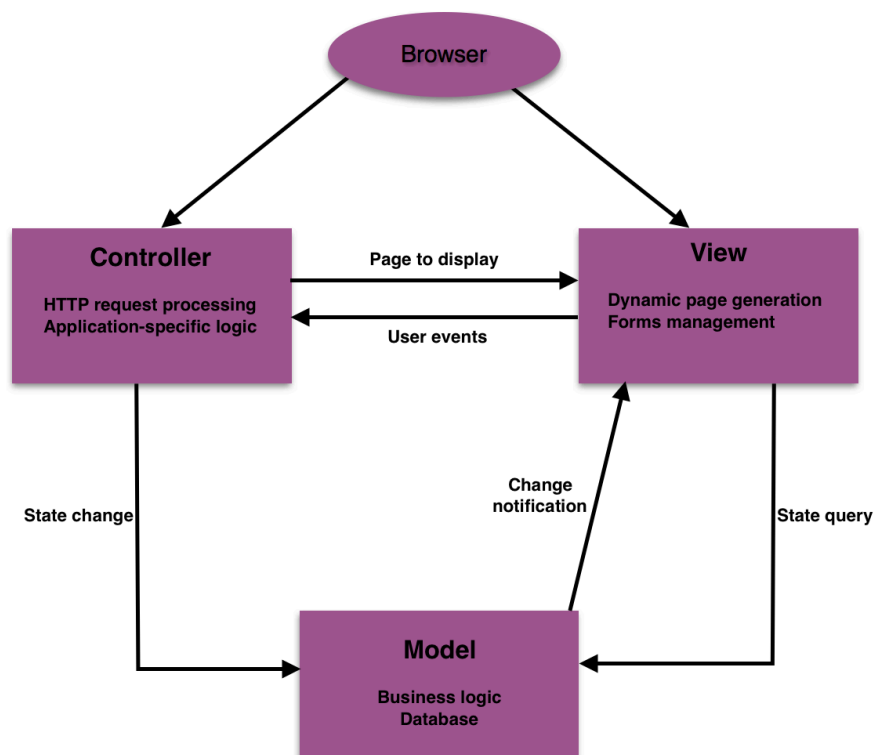


Figure 2.3 - System architecture

The Model-View-Controller architecture is a popularly used architectural pattern for web-based applications and allows data to change independent of its representation. It was chosen for this project because consolidation into three distinct modules will help in development, debugging, error detecting and maintenance. Using this architectural pattern,

the application's presentation and interaction are separated from the system data. The application is structured into three logical modules that interact with each other. The Model manages the application data and related operations on data. The Controller manages user interactions by handling application-specific logic and passes these interactions to the View and Model. The View defines and manages how data is presented to end-users.

- Model

This part of the application defines the application's basic functionality behind a set of business logic. The model comprises of classes based on the key objects of the application which are:

- adb: this class a root class for managing all classes. This class has functions to connect to the database, log errors, fetch rows from the database tables, run queries and get number of rows in a current dataset. This class is included in most other classes and in controller files. It can therefore be identified as the backbone class of the application.
- Customer: This is a class to manage all customer functions. It contains queries to the database for adding customers, editing customer details, searching for customers, viewing single customer details and viewing all customers in database.
- Employee: This is a class to manage all employee functions. It contains queries to the database for adding employees, editing employee details, searching for employees and getting details of an employee.

- Item: This is a class to manage all item functions. It contains queries to the database for adding items, editing item details, searching database records for an item based on item name, viewing details of a single item, viewing all items in the database, viewing of all items in stock, viewing of all items low in stock and viewing of all items that are sold out.
 - Notification: This is a class to manage all notification functions. It contains queries to the database for adding notifications, editing notifications, viewing single notifications and viewing all notification records in database.
 - Sale: This is a class to manage all sale functions. It contains queries to the database for adding sales, editing sales, viewing single sale transactions, viewing all sale transactions, getting last sale transaction record and adding receivable record to database based on action from a new sale.
- View
- The View functions as the user interface for the application and is made with HTML, CSS, JavaScript and a template. A template was used in order to maintain a consistent look and feel of the application and CSS was used to style and beautify the application. View comprises of different html pages for presentation of data to users. The pages range from the login page to the application's index page to the major pages for Sales, Inventory, Customers and Receivables. The Sales, Inventory and Customer pages each have buttons that navigate to pages for adding, editing and viewing. The Receivables page has a button that navigates to a page for recording payments.

- Controller

The Controller manipulates and processes data from Model by deciding which View to display based on the user's request and other considerations. This sequencing of calls to the Model and Views based on input from the user defines the application's workflow. The controllers consist of PHP pages that handle different application-specific logics. The PHP pages are used to create instances of the Models for adding, editing, and viewing.

2.4 Database Architecture

In building the database for the system, the major entities of the system and key processes were considered. The scenarios included in Appendix B were used as the starting point. Entities and relationships between entities were identified from the scenarios. Based on the entities and relationships identified, a simple ER diagram was drawn on paper and enterprise rules were realised. The visualization of the database design on paper helped in recreating a high-level EER diagram using MySQLWorkbench. After designing an EER diagram, the logical database, which includes the tables of the database, was designed and the queries for creating the database and tables were executed using phpMyAdmin.

The database is currently hosted on Ashesi's server but will be hosted for business owners on a MySQL database that requires internet connectivity. The database architecture is modelled in Figure 2.4.

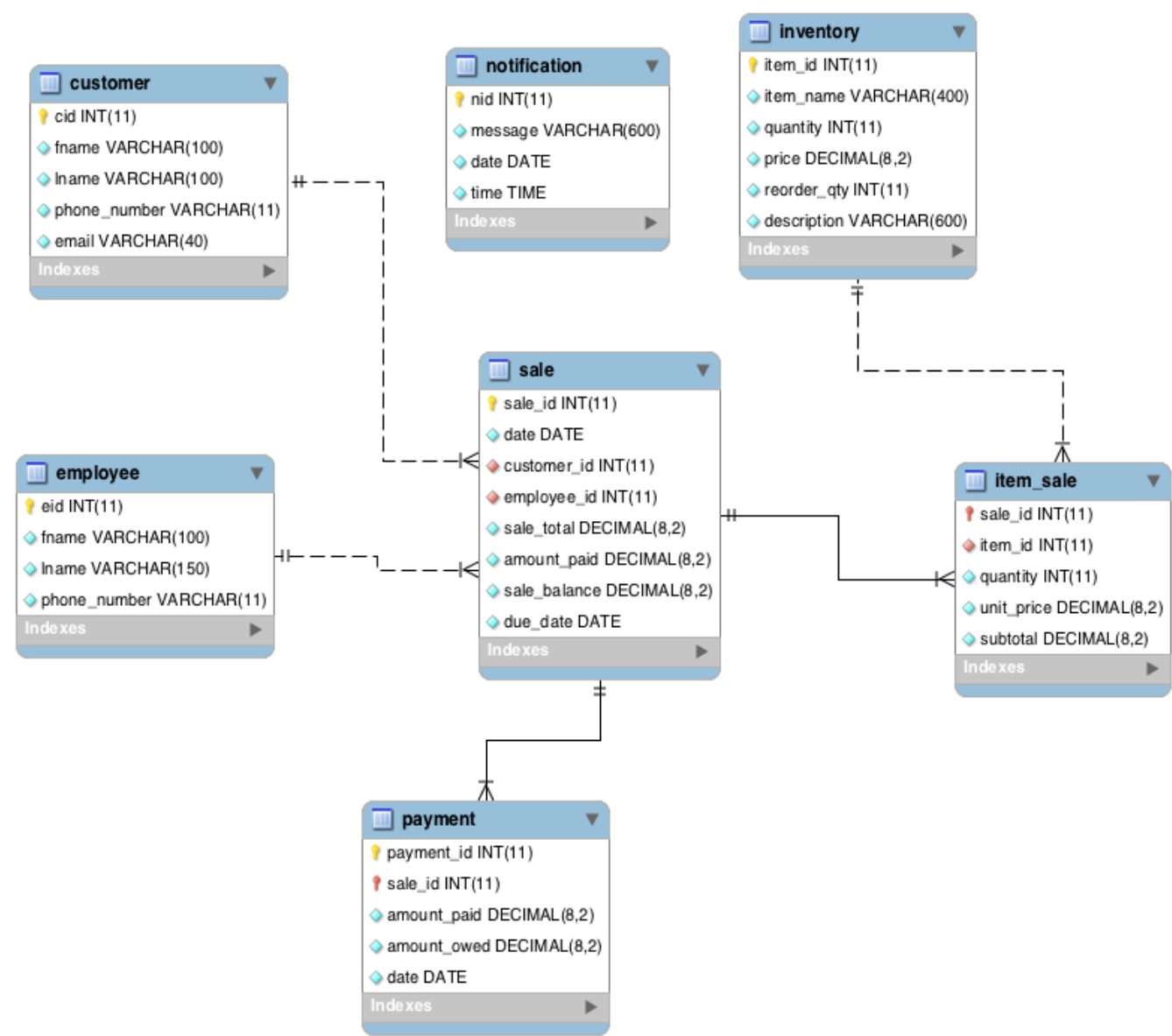


Figure 2.4 - Database architecture

Chapter 3: Implementation

3.1 Implementation overview

Considering the type of end-users, a user-centred approach was used in the development of the application because it is a widely accepted way to build interfaces with good usability properties. The implementation process was an iterative process which started off with low level prototype based on requirements gathered during the requirements engineering process. After a couple of iterations involving designing, testing and evaluating, a template was selected and modified. The different views were implemented and then the controllers for each were linked to the views. The implementation was done using MySQL, HTML, JavaScript and PHP elements.

3.2 Implementation

The first step in implementing was the design of a low level prototype. Sketches were drawn to help visualize the application. The sketches were used to create a storyboard based on the scenarios deduced in the design stage. The storyboard served as a paper prototype for the application. The paper prototype helped to focus attention on the bigger picture and issues of usability and simplicity rather than font size, colour, alignment etc. It was also easy to build and changes were easily made between user tests and in some instances, during the user test. The paper prototype was tested on three end-users using the formative evaluation process which required the users to think aloud while using the prototype. The test revealed that the users understood the conceptual model of the application and its functionality. However, there were a few issues concerning some terminology used and a few navigation confusions. These

issues were resolved and second tests were carried out to ensure that the changes were effective.

The web application was built based on feedback from the paper prototype. In implementing the web application, three key factors were considered: simplicity, learnability and usability because the end-users are novice computer users and are not interested in using complex applications.

As with most social networking websites and applications, after the business owner logs in to the web application, they will be directed to the application's homepage which presents an overview of recent activities within the application, as shown in Figure 3.1. This design choice was made in order to present the user with a familiar user interface similar to Facebook's homepage. This is to ensure that once the users log in to the application they are able to reacquaint themselves with recent activities by quickly browsing through.

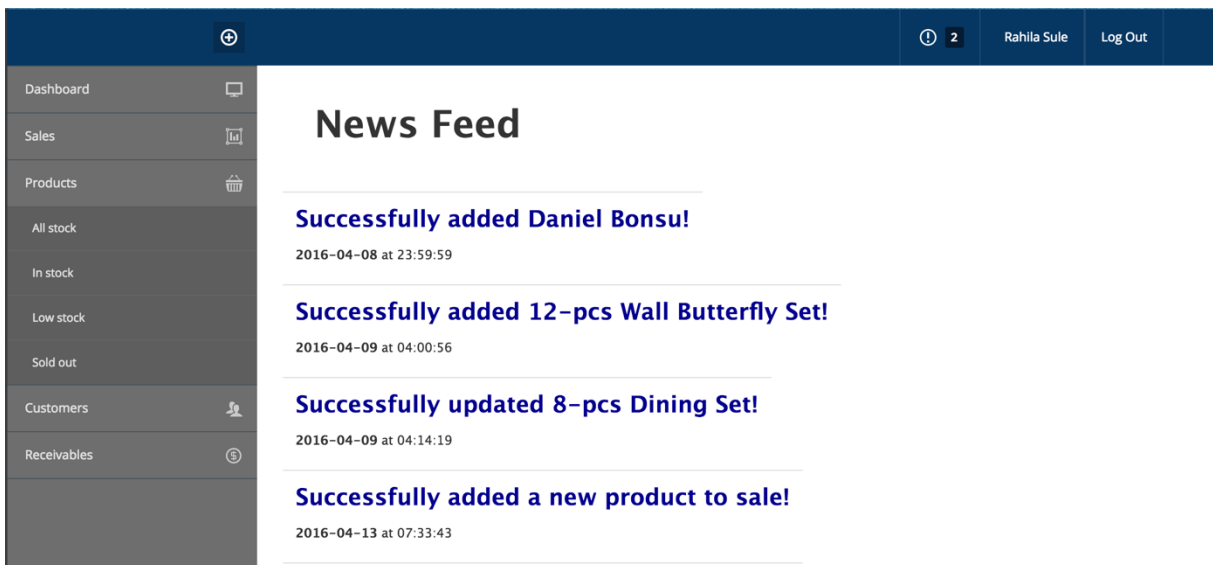


Figure 3.1 - Homepage

The homepage has a very clean and simple look with a header and a side navigation bar for accessing sections of the application. Thus, maintaining simplicity will ensure that the

interface is easy for them to learn because there are less things for them to learn. To enhance the application's simplicity, the theme comprises of cool colours that are easy on the eyes.

To improve efficiency, the homepage has a shortcut, beside the application's logo, which allows the business owner to easily record a new sale, add a new product or add a new customer. The shortcut which is depicted by a plus icon is illustrated in Figure 3.2.

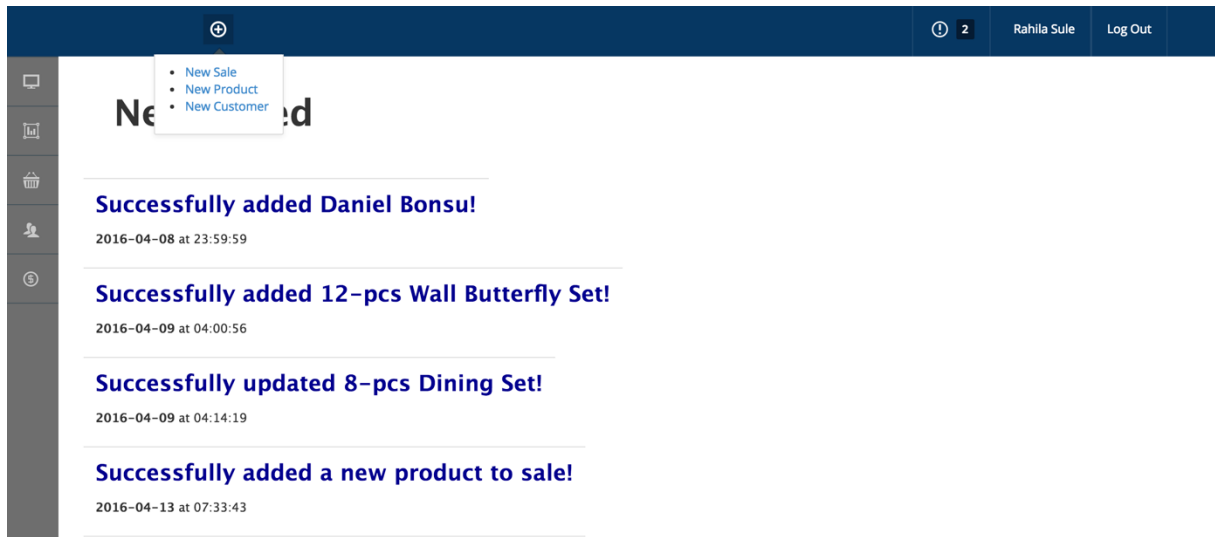


Figure 3.2 - Shortcut on homepage

After clicking on any of the four sections, the initial view is a table view of all recorded entries of the details of that section. Hence the sales section will navigate to a page with a table view of all recorded sales and accordingly for the other sections. Each table column can be sorted in ascending or descending order and the results can be filtered based on text input into the search textbox.

Sales

The Sales section will be the most used section of the application since sales are frequent and key events in businesses. Hence this section was implemented to be a stress-free

and easy process for the end-user that requires only a few inputs. Figure 3.3 below is a sequence diagram to illustrate the sequence of activities that take place when a new sale is recorded.

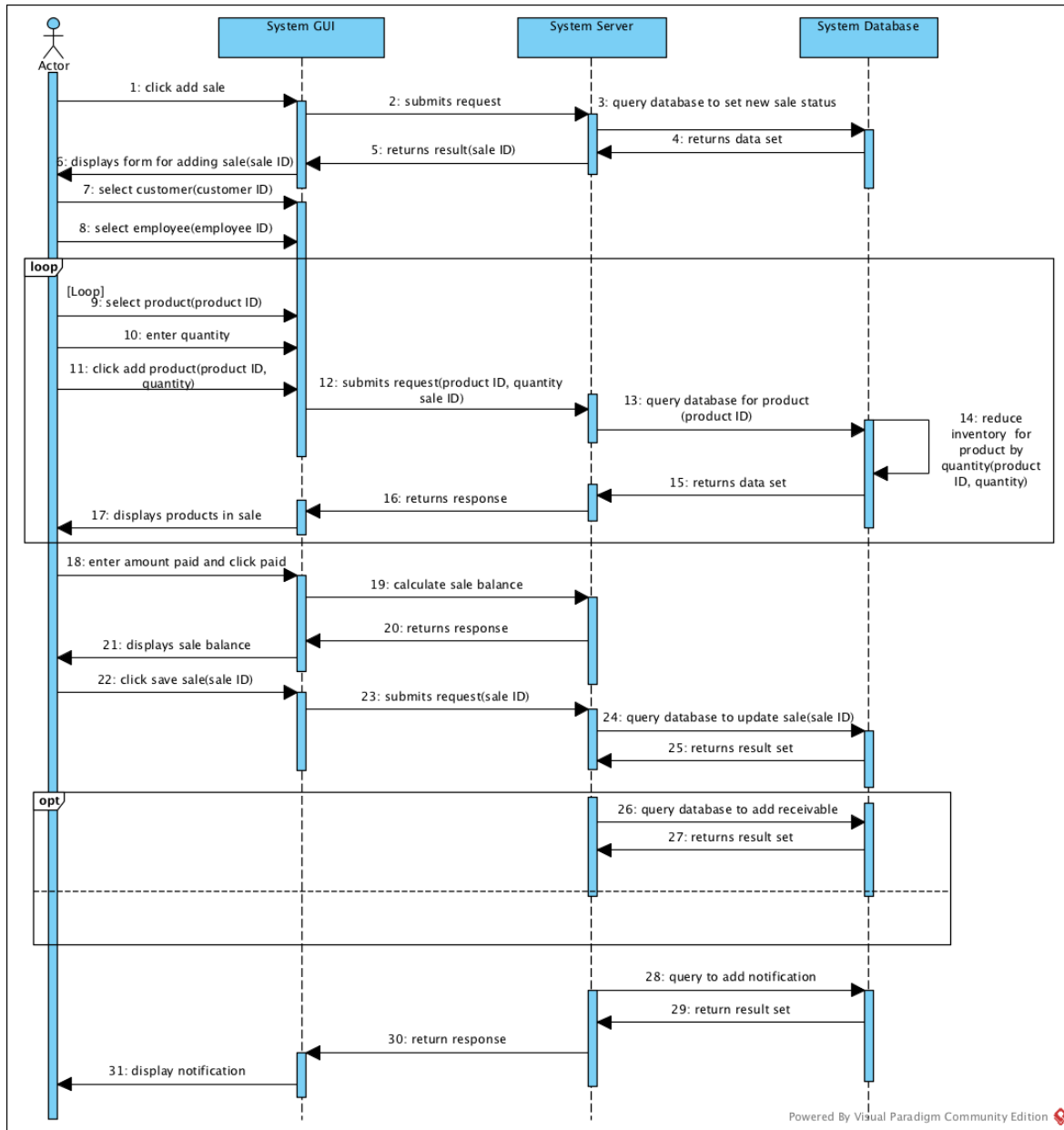


Figure 3.3 - Sequence diagram for add sale

To add a sale, the user needs to select the customer and employee from dropdown lists implemented using PHP and SQL query as shown in Figure 3.4.

```

<div class="form-group">
  <label class="col-sm-3 control-label" style="margin-left:10px"> Customer: </label>
  <div class="col-sm-4">
    <select data-placeholder="Choose a customer..." class="form-control select-liquid" id="cid">
      <option value="0">--Select Customer--</option>
      <?php
        include_once("../application/models/customer.php");
        $arow = new customer();

        $arow->view_all_customers();
        while($row=$arow->fetch()){
          echo "<option value='{$row['cid']}'>{$row['fname']} {$row['lname']}</option>";
        }
      ?>
    </select>
  </div>
</div>
</div>

<div class="col-sm-6">
  <div class="form-group">
    <label class="col-sm-3 control-label">Employee: </label>
    <div class="col-sm-4">
      <select data-placeholder="Choose an employee..." class="form-control select-liquid" id="eid">
        <option value="0">--Select Employee--</option>
        <?php
          include_once("../application/models/employee.php");
          $srow = new employee();

          $srow->get_employees();
          while($row=$srow->fetch()){
            echo "<option value='{$row['eid']}'>{$row['fname']} {$row['lname']}</option>";
          }
        ?>
      </select>
    </div>
  </div>
</div>

```

Figure 3.4 - PHP and query for customer and employee dropdown list

If the customer is a new customer, he or she can easily be added using the shortcut at the top of the screen and then the sale process can be continued. The user also has to select the products that are to be included in the sale by typing the name of the product in a textbox and selecting the product from the dropdown list that appears. This section was implemented to calculate the sale total by adding the subtotals of each item. Beneath the sale total is a textbox for the user to input the amount paid by the customer. The sale balance is then calculated by subtracting the amount paid from the sale total. This process was implemented using SQL query within the sale model and PHP. Snippets of the code used for implementation of this section are shown in Figure 3.5.

```

<?php
/**
 * author: Rahila Sule
 * description: A class to manage all sale functions.
 * date: 23-02-2016
 */

include_once("adb.php");
class sale extends adb
{
    //constructor for sale class
    function sale()
    {

    }

    //adds sale record to database
    function add_sale($sid, $cid, $eid, $total, $paid)
    {
        $str_query="UPDATE sale SET date=DATE(CURDATE()), customer_id=$cid, employee_id=$eid,
        sale_total=$total, amount_paid= $paid, sale_balance = sale_total - amount_paid, due_date=DATE_ADD(date, INTERVAL 14 DAY)
        WHERE sale_id=$sid";
        $this->query($str_query);
        $str_query="select last_insert_id() last_id";
        $this->query($str_query);
        $data = $this->fetch();
        //var_dump($data);
        $str_query="INSERT INTO payment SET sale_id = {$data['last_id']},
        amount_paid=(SELECT sale.amount_paid FROM sale WHERE sale_id={$data['last_id']}),
        amount_owed=(SELECT sale.sale_balance FROM sale
        WHERE sale.sale_id={$data['last_id']}, date=DATE(CURDATE()))";
        return $this->query($str_query);
    }
}

```

Figure 3.5 - SQL query for adding sale

A snippet of the view of the add sale section is illustrated in Figure 3.6.

The screenshot shows a web application interface for adding a new sale. On the left is a sidebar with navigation links: Dashboard, Sales (highlighted), Products, Customers, and Receivables. The top header bar shows the user's name 'Rahila Sule' and a 'Log Out' button. The main content area is titled 'New Sale' and displays the current date 'Thursday, April 28, 2016'. Below this, there are dropdown menus for 'Customer' and 'Employee'. A table-like structure is used for adding products, with columns for 'Product', 'Quantity', 'Unit Price', and 'Subtotal'. A row is shown with a dropdown for 'Add a product--' and an 'Add product' button. Below the table is a 'Notes' section with a text area. On the right side of the form, there are input fields for 'Total:', 'Subtotal:', and 'Paid:'. At the bottom right, there are 'Save' and 'Cancel' buttons.

Figure 3.6 - View for adding sale

Products

For easy inventory management, the products section is subdivided into four parts: all stock, in stock, low stock and sold out. Figure 3.7 contains a snippet of the code that was used to implement the various product views.

```
//allows view of a single item
function view_item($item_id)
{
    $str_query="SELECT * FROM inventory WHERE item_id=$item_id";
    return $this->query($str_query);
}

//allows view of all item records
function view_all_items()
{
    $str_query="SELECT * FROM inventory";
    return $this->query($str_query);
}

//allows view of all items in stock
function view_items_instock()
{
    $str_query="SELECT * FROM inventory WHERE quantity > 0";
    return $this->query($str_query);
}

//allows view of all items low in stock
function view_lowstock()
{
    $str_query="SELECT * FROM inventory WHERE quantity < reorder_qty AND quantity !=0";
    return $this->query($str_query);
}

//allows view of sold out stock
function view_soldout()
{
    $str_query="SELECT * FROM inventory WHERE quantity = 0";
    return $this->query($str_query);
}
```

Figure 3.7 - Various product views

The Products page also has a button that navigates to a page for adding new products. This page was implemented using an HTML form and SQL query that adds a new product to the database. To edit an item, the user can click the edit button beside a product to edit details of the product. The edit was implemented using a modal implemented with HTML and JavaScript code.

The process of inventory reconciliation is automated by the application. When a sale is recorded, the amount of each item associated with the sale is reconciled in the inventory table

of the database. Hence, when a sale is made, inventory is reduced by the quantity of each item that is recorded for the sale.

Customers

The Customers section gives a table view of all the business' customers. The user can add a new customer through the 'add customer' button that navigates to a new page with an HTML form for the essential customer details: customer first name and last name, email address and phone number. These details can also be edited by clicking the edit hyperlink on the record to be edited. The details requested from the user are minimal because the process of data collection has to be short in order to make it an efficient process that can be done in a few minutes since the mode of data collection is verbal.

Receivables

Records in the receivables table are added automatically based on sale transactions. If the amount paid for a sale is not the same as the total sale, a new receivable record is created and the balance (the sale total minus the amount paid) and sale total are added to the new receivable record.

Notifications

In an effort to make the business processes more efficient for end-users and to close the gap created by similar applications, the mBridge application implements a notification system. This module is a major part of the application which distinguishes mBridge from similar applications. A research conducted by Muangneon and Erjongmanee (2015)

characterized Facebook http requests into two sessions. One was which activity was performed first and the other was which activity had more http requests per session. Notification and newsfeed were two of the five major activities the results of the research revealed. This result was not surprising because users perceive notifications as a means of validating that activities are being performed on the user's account.

Therefore, social media applications have notification systems that keep the user updated on activities within the user's social circle and activities on the user's profile such as likes and comments. Similarly, the application's notification system provides the business owner with updates to their business such as new sales, new customers, new receivable, new products and low stock. Hence, the application's user interface was slightly modelled after Facebook, although mBridge is an application that has only essential features in order to achieve simplicity of the application. The homepage is a display of recent activities and serves as a news feed for the user. Aside the homepage, notifications for the application can be viewed at the top of the page beside the user's name. Also, the notification system addresses the security concern about employees having full access to the application since the business owner can monitor all activities. This is because every change made is logged and a notification is recorded for the change.

Notification messages were structured to be simple single line success or error messages. The success messages appear on-screen in a green banner while the error messages appear on-screen in a red banner.

3.3 Technology, Tools and Platform

MAMP (Macintosh, Apache, MySQL, PHP) is a relatively new technology that is used for building web applications. It is completely open-source, easy to learn and allows for quick and easy development. The MAMP framework was chosen for implementing this web application as it is widely used to solve business problems because as most businesses require, it allows for access to information, processing of information and the availability of information anywhere. The development platform uses Mac OS X as the operating system, Apache as the web server, MySQL as the database and PHP as the object-oriented scripting language. Mac OS X, which is the operating system, runs all the other three components. Apache provides the mechanism for providing web pages because PHP resides in Apache and both work together to provide dynamic pages. MySQL provides the data storage for the system and uses phpMyAdmin as the database for this application. PHP holds the parts of the MAMP together and is used to code dynamic content that can access data in the phpMyAdmin database.

Table 3.1 - Technologies used for implementation

Behavioral	PHP, JavaScript
Presentation	CSS, HTML5
Structure	HTML5, PHP

- Behavioural:

JavaScript is the scripting language of the web that allows the addition of dynamic behaviour to web documents. It was used to implement popover dropdown menus and the dynamic

dropdown in the application's sidebar. PHP is a server scripting language for making interactive and dynamic web pages. It was used for interacting with HTML forms, communicating with the database to store information from the user or to retrieve information displayed and to generate secure web pages.

- Presentation:

A Bootstrap template was used to implement the user interface for the application. The Bootstrap template uses Cascading Style Sheets (CSS) for presentation. CSS is the presentation language for web content that applies styling, layout and animation to mark up content such as HTML. It was used to enable consistency of presentation in the web application because one change to the web application's CSS will automatically make it to every page of the web application. Also, it tailors the appearance of the web application to be available for viewing on numerous media by allowing the same mark up page to be presented in different viewing styles. CSS was used for implementing front-end web design.

- Structure:

HTML5 is a mark up language that allows more diverse and powerful websites and applications. It was used to give the user interface a frame for how information will be displayed to user. PHP was used to structure the back-end model which deals with accessing and manipulating data in database.

Chapter 4: Testing

4.1 Overview

The first goal of testing leads to validation and verification testing to ensure that the software is fit for its purpose. Validation testing, which started as soon as development started, was done to ensure that the application meets the end-user's needs. From the derivation of the application's functional and non-functional requirements to implementation and testing, all procedures have been ensured to reiterate the user's requirements. All decisions and methods used were chosen to ensure that the application is a simple, learnable and usable application for business owners.

Development, release and user tests were conducted to detect any bugs within the application and to show that the application does what it is intended to do.

4.2 Development Testing

This section discusses the testing activities carried out by the application developer. Unit tests, component tests and system tests were conducted. The aim of these tests were to check the system for bugs. Hence, the development tests were interleaved with the application's debugging procedure.

4.2.1 Unit Testing

For this testing, individual object classes or methods of the system were tested to ensure that the functionality of the objects is correct. The test was conducted by calls to the different object functions using different input parameters. For each object class, similar test cases were generated for similar functions. All functions for creating new records (adding a

sale, adding a customer, adding a product, adding a notification) were tested to show that the new records exist in the database and that the fields have been set as specified. All editing functions were tested to ensure that the modifications made were updated accordingly within the database. For all units, abnormal inputs were tested to check how each unit handles such cases to avoid crashing the component. After revising all errors, all units were certified to be fully operational.

4.2.2 Component Testing

This testing involved the integration of several units of the software to create composite components. This test was conducted based on the application's four major sections. For each section four test cases which included one with valid inputs, one with invalid inputs, one with some null values and one with boundary values were executed.

For the customer section, new customers were added and crosschecked against the database to ensure they were actually added correctly then the customers' information were edited and crosschecked against the database again. The notifications were also checked to ensure that the notifications displayed to the user after adding and editing reflected in the notifications tab at the top right corner of the screen and in the notifications table in the database.

For the products section, new products were added and crosschecked against the database to ensure they were actually added correctly then the products' information were edited and crosschecked against the database again. The notifications were also checked to ensure that the notifications displayed to the user after adding and editing reflected in the notifications tab at the top right corner of the screen and in the notifications table in the

database. Some items were reduced to quantities lower than their reorder quantities to test if the notification system alerts for a low stock and notifications table in database contains record of item as low stock. Some item quantities were also reduced to zero to check if the notification system will alert item as sold out and notifications table in database contains record of item as sold out.

For the sales section, new sales were added and crosschecked against the database to ensure they were actually added correctly then the sales' information were edited and crosschecked against the database again. The notifications were also checked to ensure that the notifications displayed to the user after adding and editing reflected in the notifications tab at the top right corner of the screen and in the notifications table in the database. Instances of sales involving full payment and partial payments were tested. For instances involving partial payments, the receivables table was crosschecked, both on the application and on the database, to ensure concurrency and accuracy.

For the receivables section, payments were made and the values were crosschecked with the payment table in the database to ensure that values were concurrent. The notifications were also checked to ensure that the notifications displayed to the user after payment is added reflected in the notifications tab at the top right corner of the screen and in the notifications table in the database.

After correcting all errors and modifying some components, all components were certified to be fully operational.

4.2.3 System Testing

This involved testing the system with all of the components of the system integrated as a whole system. For the system testing, the first initial system test was conducted by the developer and then consequent system tests were executed by three professional testers. All testers were allowed to freely explore the application and perform tasks based on their own discretion whilst observing the performance of the entire application. One tester examined the products section of the application. She tried adding multiple new products and constantly checked the notifications tab to ensure that a notification was recorded for each new product. Another tester examined how the inventory is reconciled after a sale. She recorded a sale transaction for some items and added various quantities of the items. Afterwards, she checked the inventory for those items to ensure that the items had been appropriately reduced by the amounts recorded for the sale. She also tested reducing the quantities of two different items below their reorder quantities and checked to ensure that the notifications tab was updated accordingly. Then she tried to restock an item to ensure the quantity increases accordingly. The last tester explored the whole application and took note of the ease of navigating between sections. He also observed the number of button clicks to perform tasks as well as how responsive the application was to errors. Comments from all testers were collated and appropriate revisions were made to the application as highlighted in Table 4.1 below.

Table 4.1 - System testing

Comments from testers	Actions taken
Tester suggested that single view of product be implemented as a pop up to avoid requiring the user navigate back-and-forth between pages.	Single view of a product was implemented using a modal.

Comments from testers	Actions taken
Tester suggested that to restock a product, a button should be placed on each row of the products and a pop up be used to restock.	A restock button was added on each row, beside the edit button. The restock button was implemented as a modal and requires the user to indicate how many new quantities of the product are being added.
Tester commented that application does not require user to make a lot of button clicks hence making it efficient for the user because navigation is easy.	None.

After applying all modifications and suggestions made, the system was certified to be fully operational.

4.3 Release Testing

For the release testing, the aim was to check that the system meets its requirements and is good enough for use by the end-users. Scenario testing was the approach employed. This method was selected because it is a quick way to test several requirements and it tests based on realistic ways in which the system will be used. The following scenario describes one way that the system may be used when a sale transaction is to be recorded.

Justice is the owner of Cooke and Co retail shop. He handles all customer related procedures of his shop and so he handles the recording of transactions. On a Monday afternoon, Justice receives a customer in his shop and the customer has picked out 3 vases to purchase. Justice logs into the mBridge application on his laptop. He uses the shortcut for recording sales located in the navigation bar at the top of the screen. As he starts the process and types the customer's name he realises the customer's details are not in the system. He then uses the shortcut at the top of screen to add the customer's details. He gets the required information from the customer and uses the shortcut to get back to recording the sale. He adds

the three vases to the sale and tells the customer the total sale amount. The customer indicates that she will pay half the amount now and the rest later. Justice inputs the amount she has paid into the system and adds the sale. He tells her that she has a 14-day payment window and tells her the due date of her payment.

Later in the day, the lady returns and indicates that she wants to make a payment on her tab. Justice uses the application to navigate to the receivables section, he searches for the lady's name, finds the sale and clicks to make a payment. He inputs the amount the lady has just paid and tells her the amount she owes.

After he closes for the day, Justice goes to view all sales made for the day. He also checks his receivables for any upcoming due dates.

The above scenario tests a number of features of the mBridge application as indicated in the Table 4.2 below:

Table 4.2 – Release testing

Test	Action taken to test	Result
Authentication by logging into the application.	To test, two random usernames and passwords were used. One set consisted of valid characters and the other consisted of invalid characters. A valid username and password was inputted in the application.	The random username and password failed to allow access into the application. Both cases of valid and invalid characters failed. The valid username and password granted access to the application.
Creation of a new customer record.	A new customer was added and email and phone number fields were not filled. A new customer was added with all required fields filled correctly.	The new customer was not added because some required fields were left blank. When all fields were filled correctly, the new customer was added and a new notification of a new user was displayed.

Test	Action taken to test	Result
Recording of a new sale transaction with partial payment.	A new sale was added and GHC 0 was indicated as the amount paid.	The sale was added and the sale was added to the receivables table with the sale total displayed as the sale balance.
Receivable record retrieval.	From the test above, the receivables table was checked and a receivable record was viewed.	The receivable record was displayed and matched the sale record with the same sale id.
Recording of debt payment.	Using the same receivable record from above, a new payment of half the sale balance was made by clicking the make payment button on the row of the record. The form was filled accordingly. Another payment was made on the same sale.	A new payment record was added to the receivable. This was viewed by clicking on the specific receivable. The result were two records of payments for that receivable and the balances were updated accordingly.
Retrieval of sales records.	After recoding the sale above, the sales table was checked to view the sale record.	The sale record was displayed and matched values inputted by tester.

During the release test, the release tester read the observed how the application behaved in response to different inputs. Some deliberate mistakes were made to check the application's response to errors. All performance problems were noted and corrected.

After release testing, it can be concluded that the application meets its requirements and is suitable for use by end-users.

4.4 User Testing

In order to ensure that the web application fulfils user requirements, it was tested on five users because this number is adequate for finding as many usability problems as with more test participants (Nielsen, 2012). The user testing mainly examined the usability,

simplicity and utility of the application. Utility refers to whether the application provides the necessary features and hence whether the application fulfils all requirements. For the purpose of this project, usability was defined based on the following five components:

- Learnability: how easy it is to use the application the first time
- Efficiency: how quickly users are able to perform tasks
- Memorability: how easy it is for users to reestablish proficiency when users return to the application after a period of inactivity
- Errors: the number of errors, severity of the errors and how users are able to recover from errors
- Satisfaction: how satisfying it is to use the design

Based on the above components, the following factors were considered during each test: time taken to complete tasks, number of errors made and ease of navigation. Findings were ranked based on low, medium or high severity to determine the criticality of the results the test revealed. Also, feedback from the test users were recorded to to determine satisfaction and to check for similarity in feedback.

Table 4.3 - User testing

User	Test	Findings	Rank of findings	Feedback
User 1	Scenario 1 from appendix B	Within less than a minute, user was able to easily search inventory for item. User easily navigated to sales section and recorded a sale within 3 minutes. After recording the sale, user easily	Medium	Easy to search through inventory. The presentation of data gives a good overview for each section.

User	Test	Findings	Rank of findings	Feedback
		viewed upcoming receivables based on due dates within one minute.		
User 2	Scenario 2 from appendix B	User was able to navigate the sales section and narrow sales list to only sales for the week. User crosschecked with inventory and was able to easily navigate between the inventory and sales sections. User made no errors throughout the activity.	Medium	The different groupings of products based on quantity available makes it easier to keep user updated on status of products. The reconciliation of inventory after every sale, is a feature that saves a lot of time for business owners since inventory reconciliation is a time-consuming task.
User 3	Scenario 3 from appendix B	User was able to add a sale within 3 minutes and made no errors while recording the sale. After receiving notification of low quantity, user checked product listing to confirm correctness of notification.	High	Notifying the business owner when an item has reached below its reorder quantity is very helpful. The notification system is a great idea for such an application.
User 4	User was given a scenario to test recording payments made on two receivable records for a certain customer.	User was able to easily navigate to add sales and to view receivables. However, user was unsure what the "Make payment" button would do but tried it anyway.	High	User expected this task to be a bit confusing but was relieved that it was an easy process due to a functional search and simple presentation of data. Also, user suggests that the "make payment" button be rephrased to "record payment" or "add payment".

User	Test	Findings	Rank of findings	Feedback
User 5	User was given a scenario to record a sale transaction in which partial payment was made. User was then asked to record a payment. After completing both tasks, the next day, the user was asked to repeat the process for another transaction.	User was able to easily navigate and record a sale. The next day when the user was asked to re-enact the same tasks, the user executed the task within a shorter time (3 minutes) than the previous day when it had taken the user six minutes.	High	User stated that the tasks are easy to perform and do not require much effort because the information being recorded is similar to what would have been recorded on paper. User also commented that although there is a similarity to paper, recording using the application is more efficient. Furthermore, the user remarked that the logic employed to update receivables record on each sale is efficient because it prevents errors that might have been recorded if paper was being used.

Chapter 5: Conclusion

5.1 Overview

This section concludes this report by discussing key challenges faced while developing the system, future work that can be done to improve the efficiency and functionality of the application and provides a brief conclusion to the report.

5.2 Challenges

In developing the mBridge software, a few challenges were encountered, the following highlights the two major challenges encountered during development and implementation of the application and actions taken to manage these challenges.

5.2.1 Integration of Facebook as a Notification System

From research, notifications were observed to play a major role in getting the attention of users. It has been observed that notifications represent the future app and Facebook is making efforts to own them. Research has shown that people tend to use their Facebook notifications mostly because the notifications serve as great and simple reminders (Muangngeon & Erjongmanee, 2015). Interviews conducted revealed that small business owners want to be fully connected to their businesses, hence, Facebook would serve as an added benefit to the mBridge application's notification system which seeks to accomplish this.

Facebook's developer API includes a Graph API which is the primary way to get data in and out of Facebook. The Graph API is a low-level HTTP API that can be used to query data and post new stories. This Graph API will be used to integrate the notification system to

post a new story to the business owner's Facebook wall. The posted story will contain only a brief notification message which will be the same as the notification displayed in the notifications tab of the application. However, the main challenge is figuring out how to make HTTP requests using PHP because PHP does not have a library for HTTP requests.

5.2.2 Ensuring Usability of System

Considering the application's target users, the usability of the application is very important. This is because a key reason identified why these users do not currently use technology in their businesses is because they believe their current methods are reliable and easy to use. Creating an application for users with such a mindset is quite challenging because the success of the application can only be realised if the application is usable and learnable to them. In managing this challenge, a user-centred design was used to develop the application. Human Computer Interaction principles were critically considered to ensure simplicity, usability and learnability of the application.

5.3 Future Work

Although the results of this project demonstrate its effectiveness, there are a few points that can be added to improve the work done and increase the functionality of the application.

- Integration with RFID technology

Radio Frequency Identification technology have been observed to generate remarkable results for the process of inventory management. RFID technology can provide source automation features that will increase the speed of data collection for analysis of products.

The RFID technology will also introduce easier and more efficient tracking of products for the business.

- Integration of a Point-Of-Sale system

A Point-Of-Sale(POS) system can be integrated with the sale section of the application. The integration of the POS will help the business with transaction processing. This system could also help business owners gain insights to customer's buying behaviors by providing variables and data on which analysis can be conducted.

- Integration of supply chain management

Currently the mBridge application does not manage suppliers or interactions with suppliers. To further ease the operational efficiency of a business, the application can be extended to include a supply management system that will manage the business' suppliers, instantly notify suppliers when stock is low and can be used for easier communication between business owners and suppliers.

5.4 Conclusion

With the rapid pace of information technology growth, there will be remarkable changes to the way of life and how things are done over the next decade. Businesses are already integrating technology into their operational processes and are leveraging technology to achieve effective competitive strategies. Different information technology solutions have been designed to help business owners manage their business processes however, these

existing solutions have limitations that make them unsuitable to business owners, especially small business owners.

mBridge is a web-based business management tool for small retail businesses that provides an easy to use inventory, sale, customer and receivable management system coupled with a notification system that enables coordination between small business owners and their businesses. To ensure the application's usability, it was developed to reduce the effort required to read or input data and as such prevent users from making errors that result in decreased operational efficiency. Testing revealed that the application, which was developed following a user-centred design approach, is a highly usable system for these novice users that reduces the effort required to manage their key business processes. Also, the testing results reveal that the application's notification system increases operational efficiency.

References

- Archon Systems. (2016). *Homepage*. Retrieved March 2016, from inFlow Inventory Software: <https://www.inflowinventory.com/>
- Boggs, R., Bozman, J., & Perry, R. (2009, August). *Reducing Downtime and Business Loss: Addressing Business Risk with Effective Technology*. Retrieved from IDC: http://www.hp.com/hpinfo/newsroom/press_kits/2009/CompetitiveEdge/ReducingDowntime.pdf
- Carta. (2016). *About*. Retrieved March 2016, from Carta: <https://getcarta.com/about/>
- Deloitte. (2015). *African Powers Of Retailing*. Retrieved from <http://www2.deloitte.com/content/dam/Deloitte/ng/Documents/consumer-business/ng-african-powers-of-retailing-new-horizons-for-growth.pdf>
- KPMG Africa Limited. (2015). *African Consumer and Retail Sector Report 2015*. Retrieved from <https://www.kpmg.com/Africa/en/IssuesAndInsights/Articles-Publications/General-Industries-Publications/Documents/Africa's%20Consumer%20Story.pdf>
- Losoftware. (2016). *About RightControl*. Retrieved March 2016, from Losoftware: <http://www.losoftware.co.uk/>
- Muangngeon, A., & Erjongmanee, S. (2015). Analysis of Facebook Activity Usage through Network and Human Perspectives. *Knowledge and Smart Technology (KST), 2015 7th International Conference* (pp. 13-18). Chonburi: IEEE.
- Nielsen, J. (2012, June 4). *How Many Test Users in a Usability Study?* Retrieved April 16, 2016, from Nielsen Norman Group: <https://www.nngroup.com/articles/how-many-test-users/>

Ricoh. (2012). *Businesses unable to keep up with impacts of technological change may disappear by 2020*. Retrieved March 27, 2016, from Ricoh:

<http://thoughtleadership.ricoh-europe.com/eu/thenextdecade/impacts/>

Ricoh. (2012, March). *The next decade of technology in business*. Retrieved March 27, 2016, from Ricoh: <http://thoughtleadership.ricoh-europe.com/eu/thenextdecade/>

Sommerville, I. (2011). *Software Engineering* (9th Edition ed.). Boston, MA, USA: Pearson Education Inc.

Zachary, P. (2009, August 10). *An Operating System for the Cloud*. Retrieved from MIT Technology Review: <https://www.technologyreview.com/s/414671/an-operating-system-for-the-cloud/>

Appendix

Appendix A

The following questions were asked during the interviews with different business owners. Some follow-up questions were also asked as based on answers provided to the questions below.

1. Personal questions

- What is your highest level of education?
- Have you taken any computer training courses?
- What types of technology do you currently use?
- How often do you use technology?
- Is any of your family members highly interested in technology?

2. Business questions

- What are daily operations you frequently perform?
- What are these processes like?
- How easy is it for you to perform your daily activities with the use of paper?
- Do you currently use any sort of office software? (e.g. business management software, Microsoft office)
- What are the most time consuming activities? What can be done to make these better?
- What are the most challenging activities? What can be done to make these better?

- Do you have an IT officer? Is he/she an employee that works everyday?

3. Requirement questions

- How will you use a feature that lets you manage inventory?
- How will you use a feature that lets you manage supply chain?
- How will you use a feature that lets you monitor sales, costs and profit?
- What reports do you use (e.g. sales)?
- What should you be able to conclude from your report? (most bought, least bought)
- How often do you need a report generated? (i.e. when? Daily, weekly, monthly, yearly?)

Appendix B

Scenarios

The following are scenarios were used in obtaining system requirements.

Scenario: 1

A customer, Mrs. Y, walks into a small retail store. She wants to buy a gift for her daughter and some household items. The business owner suggests some items to her and she makes her choice. The business owner searches inventory for item to ensure that the quantity she requires is available. After checking, Mrs. Y is notified that she can buy her requested quantity. The business owner records the sale and Mrs. Y pays half of the sale total. The sale balance is added as a receivable. At the end of the day, the business owner looks through her receivables list to see all upcoming receivables.

Scenario: 2

At the end of the week, the owner of the store wants to reconcile the sales and the inventory. Rather than checking each receipt in the receipt book, like she did previously, she simply logs in to mBridge on her iPad and navigates to the sales section. She simply checks all sales recorded for the week and checks inventory page to ensure that all reconciliations are accurate. She does not need to manually update her inventory with sales transactions because mBridge does that for her already.

Scenario: 3

A customer, Mr. B buys some products from a shop and pays part of the sale total. An employee of the shop records a debt sale for Mr. B. The business owner, who is currently not in the country, receives a notification that a sale has been recorded and that a receivable record has been added. The business owner is also notified that one of the products in the sale has reached below its reorder quantity.

Appendix C

The table below shows West African countries with contribution of country retail sales to Gross Domestic Product (% , 2013) as reported by Deloitte in their 2015 African Powers of Retailing report. Deloitte reported that the GDP numbers were taken from Planet Retail (Deloitte, 2015).

Region	Country	Retail sales GDP contribution, 2013
North Africa	Algeria	22.2
	Egypt	43.3
	Libya	21.3
	Morocco	40.2
	Sudan	42.0
	Tunisia	34.9
	Western Sahara •	63.2
Southern Africa	Botswana	41.8
	Lesotho	62.0
	Namibia	31.0
	South Africa	32.8
	Swaziland	44.4
West Africa	Benin	57.4
	Burkina Faso	51.9
	Cape Verde	43.0
	Côte d'Ivoire	41.2
	Gambia	72.1
	Ghana	39.8
	Guinea	62.0
	Guinea-Bissau	58.9
	Liberia	63.2
	Mali	63.1
	Mauritania	46.8
	Niger	57.3
	Nigeria	23.6
	Senegal	53.1
	Sierra Leone	68.9
	Togo	64.8

Figure C1 - Retail sales GDP contribution, 2013

Appendix D

Plan for requirement analysis

The problem identified as the basis for this project is that there are retail business owners who currently do not utilize technology in their businesses because they are uncomfortable with the use of technology. Therefore, this project seeks to curtail this problem by developing a management information software that incorporates technology that such business owners are comfortable with and are currently using. This document contains a plan for the requirements analysis of the project.

Aim: Develop a management information software for non-tech savvy retail business owners to improve operations and reduce overhead.

Objectives:

- Easier and more efficient tracking of key business processes
- Reduce the use of paper
- Increase business' operational efficiency

Table D1 - Plan for requirement analysis

Resource	Information	Method	Timeline	Cost
Sales reports	Understanding of format and required output	Hard data	5 days	0
Inventory records	Understanding of current methods of keeping inventory and required data to be included in an inventory record	Hard data	4 days	0
Business owners	Understanding of background of interviewee and the business	Interview	10 days	Phone credit:

				GHC 30
Direct observation	General operation of business, daily activities, technology currently being used	Direct observation	5 days	Travel costs: GHC 500
Internet	General research	Internet research	Whole project period	0
Existing management information software	Understanding of domain, advantages, disadvantages, common faults with such software	Internet research	9 days	0
Human Computer Interaction expert	User interface requirements and principles for target users	Interview and discussion	2 days	0
Human Computer Interaction principles	Basic HCI principles for MIS domain and user interface	Internet and library research	3 days	0

Resources

The resources stated above are explained in depth below.

- Sales reports: Sales reports generated for 2013 and 2014 will be reviewed for each business. This information is important to get an understanding of the required output of sales reports and also to understand the general format of sales reports. This document will be obtained from the business and is estimated to cost GHC 0 because it will be given freely by the business owners after a confidentiality agreement is signed.

- **Inventory books:** The current methods the businesses use to keep inventory and the required data to be included in inventory records will be noted. This information is important because inventory management is a key activity for retail businesses and so will be a key component of the software. Good reports should be able to be generated from inventory. This document will be obtained from businesses and is estimated to cost GHC 0 because it will be given freely by business owners after a confidentiality agreement is signed.
- **Business owners:** An interview, which will be divided into two parts, will be conducted to get some information from business owners. The adequate sample size for this type of management information software is 100, however, only 5 participants will be used for this project due to constraints of resources and time. For the first part of the interview, business owners will be interviewed in order to gain understanding of their backgrounds, i.e. their educational background, society background and technological experience. This information is important to understand the business owners' technological competence and their basis for not utilizing technology in their businesses. The second part of the interview will be based on the background of the business. This information is important to determine the general aspects of the business that could be improved through the use of technology. After conducting the interviews, a report will be generated which will indicate similarities among all the interviewees such as general reasons for not using technology and general areas that could be optimized with technology. The interviews will be both physical and phone interviews. The estimated cost in conducting this task is GHC 30 for all phone interviews.

- Direct observation: Direct observation was conducted for a business before this project was created. Due to the fact that a business was observed before the project was structured, there may be some gaps in the report generated from the direct observation because at the time of the observation, there were no defined things to look out for. It is estimated that some travel costs will be incurred but an exact amount cannot be determined now.
- Internet: The internet will be used through the course of this project as a major resource for information on the project. It will be used for research purposes alongside the library.
- Existing management information software: Already existing management information software will be reviewed to gain a general understanding of the expected domain. Also, the advantages, disadvantages, and common mistakes of existing management information software will be reviewed in order to generate system requirements.
- Human computer interaction(HCI) expert: A human computer interaction expert will be interviewed in order to generate HCI specifications that the software should meet in order to be usable by the target users.
- Human computer interaction(HCI) principles: In addition to the interview and discussion with the HCI expert, a document containing basic and required HCI principles will be used when designing the software's interface and interactions.

Appendix E

Functional requirements

Log in: In order to use the system, a business owner should be logged in to the web application as an administrator.

Log out: After using the system, the business owner should be able to successfully log out.

Search for item: A business owner should be able to search for an item based on item name.

Search for sale: A business owner should be able to search for a sale based on certain criteria, i.e. customer's name, date of sale, item purchased etc.

Search for customer: A business owner should be able to search for a customer based on name.

Add item: Given the business owner is logged in, the business owner should be able to fill in the required data to add an item to inventory and the item should be added to inventory list.

Edit item: A business owner can edit the details of an item. When a business owner edits an item, the item should be updated in the inventory list.

View item: A business owner should be able to view the details of an inventory item.

Add transaction: Given the business owner is logged in, the business owner should be able to fill in the required data to add a sale transaction and the transaction should be added to sales records.

Edit transaction: A business owner can edit the details of a sale. When a business owner edits a sale, the transaction should be updated in the sales records.

View transaction: A business owner should be able to view the details of a sale transaction.

View receivable: A business owner should be able to view the details of a debt entry.

Add customer: Given the business owner is logged in, the business owner should be able to fill in the required data to add a customer and the customer should be added to customer database.

Edit customer: A business owner can edit the details of a customer. When a business owner edits a customer's details, the customer's details should be updated in the customer database.

View customer: A business owner should be able to view the details of a customer.

View notifications: A business owner should be able to view all notifications within the application.

The following are functional requirements for employees:

Log in: In order to use the system, an employee should be logged in to the web application as an employee.

Log out: After using the system, the employee should be able to successfully log out.

Add item: Given the employee is logged in, the employee should be able to fill in the required data to add an item to inventory and the item should be added to inventory list.

View item: An employee should be able to view the details of an inventory item.

Add transaction: Given the employee is logged in, the employee should be able to fill in the required data to add a sale transaction and the transaction should be added to sales records.

View transaction: An employee should be able to view the details of a sale transaction.

View receivables: An employee should be able to view receivables.

Add customer: Given the employee is logged in, the employee should be able to fill in the required data to add a customer and the customer should be added to customer database.

View customer: An employee should be able to view the details of a customer.

Search for item: An employee should be able to search for an item based on certain criteria, i.e. name.

Search for sale: An employee should be able to search for a sale based on certain criteria, i.e. customer's name, date of sale, item purchased etc.

Search for customer: An employee should be able to search for a customer based on either first name or last name.