

**DEVELOPING A BLOCKCHAIN APPLICATION TO STORE PUBLIC RECORDS.**

BY

NJOROGE NAHASHON NJENGA

IF/20/14

**PROJECT REPORT SUBMITTED TO THE SCHOOL OF INFORMATION SCIENCES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF BACHELOR OF SCIENCE IN INFORMATION SCIENCE OF MOI UNIVERSITY**

AUGUST 2018

# DECLARATION

I declare that this is my original work and has not been presented in any other school for examination purposes.

Name: NAHASHON NJENGA

Signature Date

Supervisor:

Name: MR. NICHOLAS KIGET

Signature Date

# ACKNOWLEDGEMENT

Enough cannot be said to thank our peer reviewers who helped mold the final form of this project, from catching simple bugs to helping with the mental design of the implementation details.

Special thanks to Mr. Dan Samoei, who helped supervise this project and make sure it meets all academic requirements.

I also want to thank my brother John Muiruri and Damaris Simiyu, for providing the necessary documents like log book, title deed and driving license. These documents were relevant in making the system store all the required information.

I cannot forget to thank my classmates and roommate for believing in me. That encouraged me to push forward with this mammoth of an undertaking.

# ABSTRACT

Fraud in the public sector has contributed to the deep rooted corruption we see all the time in the news. Ranging from land grabbing, to use of fake academic certificates and theft, illegal sale and purchasing of vehicles. People seem to have found illegal means to generate certificates and other documents that seem authentic, and use them in the economy.

The government has gone a step further to digitize the process of acquiring these documents. This has helped accelerate the process of acquiring these documents significantly. But the issue of fraud still looms. This is because of the nature of the technologies used. Normal databases offer a way for someone to edit any records without a trace. This ideally means a person with a contact to the system administrator can easily bribe them or worse still blackmail them to edit their records.

The blockchain helps solve this problem. It promises immutability; once records are added to the blockchain, there is no changing them. And the ones that need to show change of ownership, like logbooks and title deeds, a trail of various owners over time can be obtained.

**TABLE OF CONTENTS**

Contents

[DECLARATION i](#_Toc522397899)

[ACKNOWLEDGEMENT ii](#_Toc522397900)

[ABSTRACT iii](#_Toc522397901)

[+LIST OF FIGURES AND TABLES vi](#_Toc522397902)

[LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS vii](#_Toc522397903)

[CHAPTER ONE 1](#_Toc522397904)

[INTRODUCTION 1](#_Toc522397905)

[1.1 Chapter Overview 1](#_Toc522397906)

[1.2 Background 1](#_Toc522397907)

[1.3 Statement of the problem 2](#_Toc522397908)

[1.4 Aim and Objectives of the study 3](#_Toc522397909)

[1.4.1 Aim of the study 3](#_Toc522397910)

[1.4.2 Objective of the study 3](#_Toc522397911)

[1.5 Research Questions 3](#_Toc522397912)

[CHAPTER TWO 6](#_Toc522397913)

[LITERATURE REVIEW 6](#_Toc522397914)

[2.3.1 Origin 14](#_Toc522397915)

[2.3.2 Takeover by Google 14](#_Toc522397916)

[2.3.3 Different Versions of Android OS 15](#_Toc522397917)

[Android Pros 17](#_Toc522397918)

[Android Cons 17](#_Toc522397919)

[2.4.1 United States of America 19](#_Toc522397920)

[2.4.2 Europe 20](#_Toc522397921)

[2.4.3 Current state of things globally when it comes to shopping 21](#_Toc522397922)

[2.5 Why Technology is Being Integrated in the Shopping Sector 21](#_Toc522397923)

[2.5.1 Benefits of integrating mobile applications in e-commerce 22](#_Toc522397924)

[Keeping up with competition 23](#_Toc522397925)

[2.6 A Look At Past And Present Shopping Systems Already In Use 25](#_Toc522397926)

[2.6.1 Globally 25](#_Toc522397927)

[2.6.2 Current shopping Systems in Kenya 33](#_Toc522397928)

[2.6.3 How developed system will complement existing Shopping systems 34](#_Toc522397929)

[2.7 Chapter Summary 35](#_Toc522397930)

[CHAPTER 3 35](#_Toc522397931)

[METHODOLOGY 35](#_Toc522397932)

[3.1 Chapter Overview 35](#_Toc522397933)

[3.2.1 The Waterfall Model 36](#_Toc522397934)

[3.2.2 Rapid Application Development 37](#_Toc522397935)

[3.3 Data Collection Instruments 39](#_Toc522397936)

[3.3.1 Questionnaire 40](#_Toc522397937)

[3.3.2 Interview 40](#_Toc522397938)

[3.3.3 Observation 40](#_Toc522397939)

[3.5 Study Area 40](#_Toc522397940)

[3.6 Target Population 41](#_Toc522397941)

[3.7 Sampling size and sampling procedures 41](#_Toc522397942)

[3.8 Data Collection Procedures 42](#_Toc522397943)

[CHAPTER 4 43](#_Toc522397944)

[SYSTEM ANALYSIS AND DESIGN 43](#_Toc522397945)

[4.2.1 Overview of the Proposed System 43](#_Toc522397946)

[4.2.2 Intended users 46](#_Toc522397947)

[4.2.3 Proposed system functions 46](#_Toc522397948)

[4.2.4 Functional requirements 47](#_Toc522397949)

[4.2.5 Non-Functional requirement 49](#_Toc522397950)

[4.3 System Design 50](#_Toc522397951)

[4.3.1 Structure Design 50](#_Toc522397952)

[4.4 Smart Shopping Mobile Application System Implementation 59](#_Toc522397953)

[4.4.1 Tools used 59](#_Toc522397954)

[4.5 Testing 67](#_Toc522397955)

[4.5.1 Testing Strategies 68](#_Toc522397956)

[4.6 Chapter Summary 69](#_Toc522397957)

[CHAPTER FIVE 70](#_Toc522397958)

[SUMMARY, CONCLUSION AND RECOMMENDATIONS 70](#_Toc522397959)

[5.1 Chapter Overview 70](#_Toc522397960)

[5.2 Summary of Findings 70](#_Toc522397961)

[5.3 Conclusion 71](#_Toc522397962)

[5.4 Recommendation 71](#_Toc522397963)

[5.5 Suggestion for further research 72](#_Toc522397964)

[REFERENCES 72](#_Toc522397965)

[APPENDICES 75](#_Toc522397966)

[Appendix 1: Letter to respondent/ site manager 75](#_Toc522397967)

[Appendix 2: Questionnaire 76](#_Toc522397968)

[Appendix 3: System requirements 79](#_Toc522397969)

[Appendix 4: Project Codes 79](#_Toc522397970)

# LIST OF FIGURES AND TABLES

[Figure 2.1 7](#_Toc522387328)

[Figure 2.2 8](#_Toc522387329)

[Figure 2.3 9](#_Toc522387330)

[Figure 2.4 9](#_Toc522387331)

[Figure 2.5 10](#_Toc522387332)

[Figure 2.6 10](#_Toc522387333)

[Figure 2.7 11](#_Toc522387334)

[Figure 2.8 12](#_Toc522387335)

[Figure 2.9 12](#_Toc522387336)

[Figure 2.10 22](#_Toc522387337)

[Figure 2.11 26](#_Toc522387338)

[Figure 2.12 26](#_Toc522387339)

[Figure 2.13 27](#_Toc522387340)

[Figure 2.14 28](#_Toc522387341)

[Figure 2.15 29](#_Toc522387342)

[Figure 2.16 29](#_Toc522387343)

[Figure 2.17 30](#_Toc522387344)

[Figure 2.18 31](#_Toc522387345)

[Figure 2.19 31](#_Toc522387346)

[Figure 2.20 32](#_Toc522387347)

[Figure 3.1 Waterfall model 36](#_Toc522387348)

[**Figure 3.2**: Prototyping approach used by RAD. 37](#_Toc522387349)

[Table 3.1: Sampling Frame 41](#_Toc522387350)

[Table 3.2: Sampling Design 41](#_Toc522387351)

[Figure 4.1 System’s architectural design 44](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387352)

[Figure 4.2 Data Flow Diagram 46](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387353)

[Figure 4.3 Use Case Diagram 52](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387354)

[Figure 4.4 Activity diagram for Administrator 53](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387355)

[Figure 4.5 Activity diagram for user 54](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387356)

[Figure 4.6 Sequence Diagram for User 57](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387357)

[Figure 4.7 Sequence Diagram for Administrator 58](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387358)

[Figure 4.8 Class Diagram 59](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387359)

[Figure 4.9 Home tab 61](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387360)

[Figure 4.10 Amount tab 62](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387361)

[Figure 4.11 Barcode Scanner tab 63](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387362)

[Figure 4.12 Scanned List tab 64](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387363)

[Figure 4.13 Mobile payment tab 65](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387364)

[Figure 4.14 Shopping List tab 66](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387365)

[Figure 4.15 Calculator Tab 67](file:///C:\Users\freda\Desktop\Project%20Documentation\FINAL%20PROJ%20FOLDER2\Project%20DocumentationZXA.docx#_Toc522387366)

# LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

**OS -** Operating System

**IT** -Information Technology

**MHz** -Mega Hertz

**MB** -Mega Bytes

**GB** -Giga Bytes

**K B** -Kilo Bytes

**RAM -**Random Access Memory

**RAD** -Rapid Application Development

# CHAPTER ONE

# INTRODUCTION

## 1.1 Chapter Overview

This chapter outlines and presents the background of the study, the problem statement, aim and objectives of the study, assumptions, significance and scope of the study.

## 1.2 Background

In 2017, my brother was running a car hire business. The business had been doing great, until the worst happened. A client rent the car for a week, and in the middle of the week, the client calls him notifying him that his car had been stolen. It turns out that this client went and lent out the car to somebody else who turns out to be a thief. The thief went and processed a fake log book and was in the process of selling the car to another person, when my brother received the news and turned off the car. With rental cars, and even most personal cars, you ought to install a car tracker in it, which helps you track the location of the car in real time on google maps, and also enables you to turn off the car remotely. My brother then went to the police to report this incident. And after a long process, he finally got his car back. The car tracker played a significant role in helping recover this lost car. If the car did not have this tracker, it would have taken a very long time to recover this car. This goes to show how easy it is to generate fake documents to be used for transactions in the country. If the system am proposing was in operation, the buyer would have simply gone to the site, searched the number plate of the car, and gotten the authentic owner of the car, and would have realized the seller is not the owner. This is just one case of many that go undocumented in our country. We need to come up with measures to help curb corruption and fraud in Kenya.

## 1.3 Statement of the problem

Over the years, corruption has continued to plague the Kenyan economy. Every newly elected government promises to come up with a solution. From the national government to county governments, measures such as lifestyle audits have been conducted to try and smoke out the corrupt.

We cannot have a ‘one size fits all’ mentality when it comes to solving the corruption problem. We have to identify various ways in which corruption takes place. The following are the most common methods of corruption:

**Bribery**

Bribery involves the improper use of gifts and favours in exchange for personal gain. It is the common form of corruption. The types of favours given are diverse and may include money, gifts, sexual favours, company shares, entertainment, employment and political benefits. The personal gain that is given can be anything from actively giving preferential treatment to having an indiscretion or crime overlooked.

**Embezzlement, theft and fraud**

Embezzlement and theft involve someone with access to funds or assets illegally taking control of them. Fraud involves using deception to convince the owner of funds or assets to give them up to an unauthorized party.

Examples include the misdirection of company funds into "shadow companies" (and then into the pockets of corrupt employees), the skimming of foreign aid money, scams and other corrupt activity.

**Graft**

The political act of **Graft** is a well-known and now global form of political corruption, being the unscrupulous and illegal use of a politician's authority for personal gain, when funds intended for public projects are intentionally misdirected in order to maximize the benefits to illegally private interests of the corrupted individual(s) and their cronies.

**Extortion and blackmail**

While bribery is the use of positive inducements for corrupt aims, extortion and blackmail centre around the use of threats. This can be the threat of violence or false imprisonment as well as exposure of an individual's secrets or prior crimes.

This includes such behavior as an influential person threatening to go to the media if they do not receive speedy medical treatment (at the expense of other patients), threatening a public official with exposure of their secrets if they do not vote in a particular manner, or demanding money in exchange for continued secrecy.

**Influence peddling**

Influence peddling is the illegal practice of using one's influence in government or connections with persons in authority to obtain favours or preferential treatment for another, usually in return for payment.

**Networking**

Networking can be an effective way for job-seekers to gain a competitive edge over others in the job-market. The idea is to cultivate personal relationships with prospective employers, selection panelists, and others, in the hope that these personal affections will influence future hiring decisions. This form of networking has been described as an attempt to corrupt formal hiring processes, where all candidates are given an equal opportunity to demonstrate their merits to selectors. The networker is accused of seeking non-meritocratic advantage over other candidates; advantage that is based on personal fondness rather than on any objective appraisal of which candidate is most qualified for the position.

**Abuse of discretion**

Abuse of discretion refers to the misuse of one's powers and decision-making facilities. Examples include a judge improperly dismissing a criminal case or a customs official using their discretion to allow a banned substance through a port.

**Favouritism, nepotism and clientelism**

Favouritism, nepotism and clientelism involve the favouring of not the perpetrator of corruption but someone related to them, such as a friend, family member or member of an association. Examples would include hiring or promoting a family member or staff member to a role they are not qualified for, who belongs to the same political party as you, regardless of merit.

As mentioned earlier, we cannot have one solution to solve all the forms of corruption outlined above. But as a start we should automate as many government services as possible, and even go as far as to use technologies such as blockchain that promise immutability of records once created.

## 1.4 Aim and Objectives of the study

### 1.4.1 Aim of the study

The main aim of this project is to design and develop a blockchain application to store public records. The application is intended for use by the various government bodies like E-Citizen, Kenya Revenue Authority and Kenya National Examination Council to add records to the blockchain, and the general public to query that data.

### 1.4.2 Objective of the study

1. To adopt a suitable methodology and use it to develop a blockchain application.
2. To adopt a suitable blockchain technology stack and use it to develop this system
3. To perform an analysis on requirements gathered for the application system.
4. To provide a way of storage and retrieval of public records secure, convenient and effortless.

## 1.5 Research Questions

1. What methodologies will be used to model and develop the application system?
2. What are the user requirements for the application system?
3. How is the system making of storage and retrieval of public records secure, convenient and effortless?

1.6 Assumptions of the study

The research was based on the assumption that the government will integrate this system with existing systems through APIs (Application Programming Interface). This therefore means that this system is a back end system, with no user interface.

1.7 Significance of the study

There has been numerus proposals [1] to use the blockchain to solve various problems in many sectors of our society ranging from the financial sector, medical sector and insurance, to education sector and the government as a whole. The existing technologies have helped automate processes in these sectors, but have been limited by the underlying architecture.

This application system on completion will be important to country because it will integrate with the existing technologies seamlessly yet provide the benefits of the blockchain. This means the users at the affected sectors will need very little training to get up to speed with this system. The user interface there are using now to populate data to their database will not be affected, only the processing and storage will, which as it stands they do not know or care since it is mean for the technical people to understand. Migration of data will also be made seamless through the use of APIs. The public will also visit the normal websites they do to get these services, and they will not notice any change in the way things are working at the moment. What they will get as an addition is the sense of security that the records they are accessing are authentic and tamper proof.

1.8 Scope of the study

The blockchain is a form of database by itself. As a result the scope of the system will cover the storage and retrieval of records. How they are populated and displayed on the user’s side has already been taken care of by the existing systems, as discussed in chapter 2 of this documentation. User authentication and authorization is also outside the scope of this project. The implementation of this system assumes that data coming to the system is from authorized user accounts. In addition, it is assumed that the data has passed some form of validation on the client side. Additional validation is also provided in this system to ensure only valid data is stored.

1.9 Chapter Summary

This chapter presents the background of the study, the problem statement, main aim of the study, its objectives and research questions, assumptions of the study, significance and scope. In the next chapter, a theoretical framework and related literature will be reviewed.

# CHAPTER TWO

# LITERATURE REVIEW

2.1 Chapter Overview

This chapter focuses on the study and scrutiny of existing systems this system will integrate with. The main purpose for this is to discover what has been done before, what is being done presently and what will be done in the future in regard to storage and retrieval of public records, and how this system will fill some of the gaps left by the existing systems and complement them. The chapter will also look at the history of blockchain technology, the architecture behind it, its application, advantages, disadvantages and limitations of this technology. I shall also explain how this system will complement these current systems.

2.2 The History of SmartPhones and Mobile App Development

In 1926, the scientist Nikola Tesla is quoted describing a piece of technology that would revolutionize the lives of its users,

“When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole. We shall be able to communicate with one another instantly, irrespective of distance. Not only this, but through television and telephony we shall see and hear one another as perfectly as though we were face to face, despite intervening distances of thousands of miles; and the instruments through which we shall be able to do his will be amazingly simple compared with our present telephone. A man will be able to carry one in his vest pocket.”

And this revolution we have lived to witness, is the onset and dawn of the Smartphones- very portable all in one devices that integrate different forms of multimedia in them as predicted by Tesla (Arthur, 2012).

A smartphone as defined in the Oxford dictionary is a mobile phone that performs many of the functions of a computer, typically having a touchscreen interface, internet access and an operating system.

In order to understand and appreciate the features of this smartphones, below is a summarized history of how they evolved, from the very first one ever made to the most current release.

**The IBM Simon (1993)**

This phone was invented in 1993 and it was a whole in one phone that incorporated voice and data services, Fax services, had a touchscreen that could be used to dial phon e numbers. It can be physically described as “brick-like, huge and heavy”.

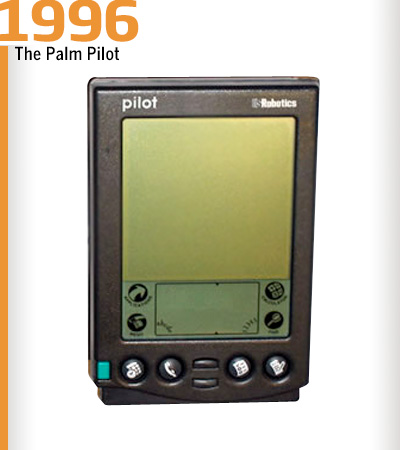


###### Figure 2.1

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

**The PALM Pilot (1996)**

This phone popularized the use of mobile data by enterprise users. It served its purpose as a true personal digital assistant (PDA), with a processing power of 16MHz and grand total memory of 128KB.



###### Figure 2.2

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

**The Nokia 9110 Communicator (1998)**

This phone had physical features that pioneered the features of current smartphones eg. the Motorola Droid that is a slider smartphone.

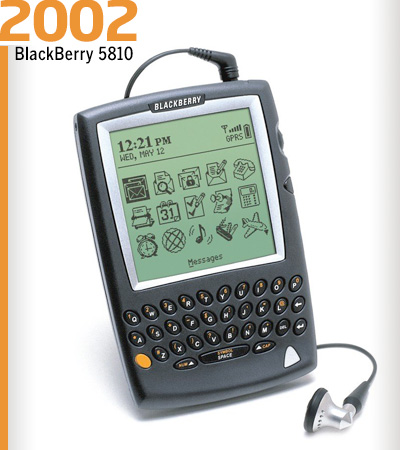


###### Figure 2.3

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

**The BlackBerry 5810 (2002)**

This phone had the ability to get e-mail and surf the web. It had a very neat design that made it to be hand-held easily. It’s downside was that you needed to plug in a headset in order to talk on the phone.



###### Figure 2.4

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

**The Palm Treo 600 (2003)**

The Treo 600 was the first smartphone released by Palm company. It featured both GSM and CDMA models and had 32MB of RAM and 144MHz of processing power.



###### Figure 2.5

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

**The Apple iPhone (2007)**

****

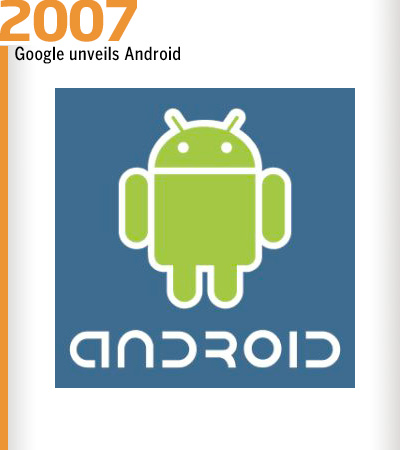
###### Figure 2.6

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

In 2007, at a heavily-hyped press event in San Francisco, Apple co-founder Steve Jobs stood on stage and unveiled a revolutionary product that not only broke the mold but also set an entirely new paradigm for computer-based phones. The look, interface and core functionality of nearly every smartphone to come along since is in some form or another derived from [the original iPhone’s](https://www.thoughtco.com/who-invented-the-iphone-1992004) innovative touchscreen-centric design.

Among some of the groundbreaking features was an expansive and responsive display from which to check email, stream video, play audio and browse the internet with a mobile browser that loaded full websites much like what’s experienced [on personal computers](https://www.thoughtco.com/history-of-computers-4082769). Apple’s unique iOS operating system allowed for a wide range of intuitive gesture-based commands and eventually a rapidly growing warehouse of downloadable third-party applications.

**Google unveils Android (2007)**



###### Figure 2.7

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

Google launched it’s Android OS in the fall of 2007. Even then it was competing with OS giants such as iPhone OS, BlackBerry OS, Windows Mobile OS and Symbian.

**The Motorola Droid (2009)**

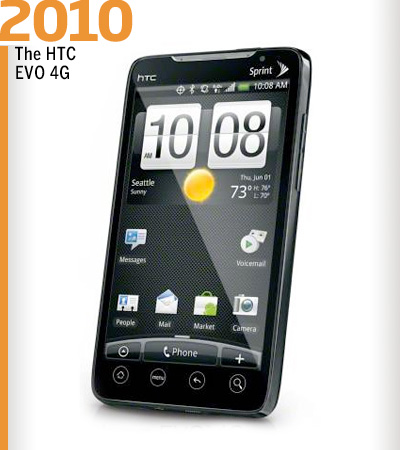
****

###### Figure 2.8

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

This phone gave the android OS the much needed popularity among users it needed. It was the first major hit that had an enduring brand recognition. The Droid was also the first Android-based smartphone to run on the Verizon network.

**The HTC EVO 4G**



###### Figure 2.9

Source: <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>

From then till now a lot of smartphone brands have been produced and sold all over the world, with the android OS running in most of these devices. According to Boulos (2010), the latest generation of smartphones are increasingly viewed as handheld computers rather than as phones, due to their powerful on-board computing capability, capacious memories, large screens and open operating systems that encourage application development. The potential for the creation of simple and easy to download apps for smartphones has created a vibrant new industry. There is now an app for just about every social, entertainment and educational requirement.

Smartphones have now achieved such a pervasive presence in society that users find it easy to self-organize themselves across large geographical areas .Many have adopted a culture where they are 'always connected' to their peer groups, communities of practice and information. The mobile phone provides an essential 'any time, any place' portal into the entire world wide web of knowledge.

This smartness has gone overboard into other products such as Televisions, watches and small electrical appliance embedded systems(Pack *et al*, 2010).

Some of the functions that can found in modern smartphones , in addition to standard mobile phone services include:- Appointment Calendar

* Appointment calendar
* Address Book
* MP3 player
* Web browser
* E-mail access, in addition to text messaging
* Mini-keyboards or onscreen keyboards
* Voice dialing
* Bluetooth
* Character recognition (allowing for handwritten input)
* Synchronization of information with desktop or laptop computers
* Voice recording
* Digital camera
* Video recording

2.3 History of Android OS

### 2.3.1 Origin

Android Inc. founded in 2003 in Palo Alto, California. It established by a team comprising of Andy Rubin, Nick Sears, Rich Miner and Chris White. Google acquired the company in 2005, which lead to the platform’s further evolution on the global platform.  
It initially developed with the goal of providing a sophisticated OS for digital cameras. The efforts later redirected to create a Smartphone operating system. Windows Mobile and Symbian were its primary competitors (Lessard, 2010).

### 2.3.2 Takeover by Google

Google acquired Android in 2005 and unveiled it in 2007. It also founded the Open Handset Alliance, a group of companies that promoted open standards within the mobile device domain. The first commercial Android device launched in September of 2008. Since then the mobile OS has seen various versions to reach the current 7.0 Nougat, which initiated in August of 2016.

Android is hailed as “the first complete, open, and free mobile platform.”

* Complete: The designers took a comprehensive approach when they developed the Android platform. They began with a secure operating system and built a robust software framework on top that allows for rich application development opportunities.
* Open: The Android platform is provided through open source licensing. Developers have unprecedented access to the handset features when developing applications.
* Free: Android applications are free to develop. There are no licensing or royalty fees to develop on the platform. No required membership fees. No required testing fees. No required signing or certification fees. Android applications can be distributed and commercialized in a variety of ways.

### 2.3.3 Different Versions of Android OS

The history of Android OS is marked by its development over the years in the form of its various versions. The different versions of this mobile operating system and their critical features are as following:

#### Android 1.0

Android v1.0 released in 2008 pioneered how to deal effectively with notifications. Its pull-down notification window was a revolutionary addition. The current Google Play Store first introduced in version 1.0 as Android Market. It also allowed the use of home screen widgets, something the iOS platform has not yet been able to include.

#### Android 1.5 (Cupcake)

This version of Android launched in 2009. It is the first Android version to have an on-screen keyboard. It also allowed third-party developers to develop and add widgets by opening the widgets SDK. Android Cupcake also allowed video captured for the first time on the platform.

#### Android 1.6 (Donut)

Android Donut was released the same year as v1.5 and introduced support for CDMA networks. It helped the mobile OS reach millions worldwide. It supported different screen sizes, allowing manufacturers to use Android on various sizes of devices. Version 1.6 also introduced the quick search box and also redesigned the Android Market.

#### Android 2.0 (Eclair)

The version 2.0 was also launched in 2009 and introduced many significant changes. In fact, it was the first mobile OS to come with Google Maps Navigation. Voice guidance and turn-by-turn navigation are some of the features introduced in Maps in Android 2.0 that continue to be in use still today. Google provided these features for free compared to paid offerings from the competition. The internet browser supported HTML5 in Android Eclair.

#### Android 2.2 (Froyo)

Android version 2.2 released in 2010 on the Nexus One device. This version refined the Android experience. Five home screen panels introduced in place of earlier 3. This version also added the mobile hotspot support and PIN lock screen.

#### Android 2.3 (Gingerbread)

Gingerbread was also introduced the same year as Froyo. The home screen and stock widgets redesigned, and the overall OS experience further refined. The version featured a better keyboard having new key colors and better multi-touch support. Android also began supporting the front-facing camera.

#### Android 3.0 (Honeycomb)

Version 3.0 released in 2011, and it was designed keeping in mind the needs of tablet users. Many of the designs from Honeycomb have been retained in the current versions too. Android began showing previews for each widget, which was a major concern with users in the earlier versions.One of the biggest changes in Honeycomb was that the virtual buttons replaced the physical button. Home, menu and back buttons became part of the software. Google also switched Android’s highlights from green to blue.

#### Android 4.0 (Ice Cream Sandwich)

The Ice Cream Sandwich carried over many of the features of Honeycomb. The legacy of the virtual buttons brought forward, and the interface became even more refined. Google released many other new small features in the update including face unlock, calendar and mail apps, and data usage details.

#### Android 4.1 (Jelly Bean)

Android Jelly Bean released in 2012. It maintained most of the visual features from its predecessor, but there were important underlying changes. Google Now was introduced, and users could open it with a quick swipe. Emails, calendar events, and weather reports brought on the same screen.  
It was also the first time when Google laid the foundations for its future digital assistants. Project Butter significantly improved the mobile OS’ touch performance. It involved developing the buffering graphics of the headset. Some of the other improvements in Android included better font, more flexibility with the widget, and expanded notifications.

#### Android 4.4 (KitKat)

Android version 4.4 was released in 2013 and introduced many big improvements and features. In fact, it accounted for a major aesthetic update to the mobile OS. The blue highlights in the Ice Cream Sandwich and Jellybean gave way to more modern and refined white highlights. The focus switched to lighter color scheme across many elements. Other new additions included the “Ok, Google” command to launch Google Now, an improved phone dialer, the Hangouts app, and full-screen apps.

#### Android 5.0 (Lollipop)

Google released Android Lollipop in 2014. It designed around Google’s popular Material Design concept. Some of the major updates included Android Runtime replacing Dalvik VM, improved notifications, support for RAW image format, and improved refinement. With Android Lollipop, Google also introduced an entirely new version of the mobile OS, Android TV.

#### Android 6.0 (Marshmallow)

Android Marshmallow launched in 2015. With Android v6.0, Google brought complete changes to the app menu. A search bar was added to search apps quickly, and the background switched from white to black. It also added memory manager, more elaborate volume controls, and began supporting fingerprint sensors.

#### Android 7.0 (Nougat)

Android version 7.0 released in August of 2016. Google Now was replaced by the Google Assistant. The notifications system received a major update with loose grouping together and screen to screen viewing. Nougat also improved multitasking. The split-screen mode allows using two apps simultaneously.

### Android Pros

* Android has a dominant share of the mobile market
* Developers generate revenue.
* You can develop on any platform.
* The environment is more open: call history is available to all apps; notifications between apps are possible as well as the sharing of content; apps can be installed from any source.
* Apps can be self-signed.
* You can publish to Google Play for a one-off fee of Ksh. 2500

.

### Android Cons

* Fragmentation between different versions of the OS, which are often significantly different, are a major problem.
* Upgrades are passed through manufacturers and carriers who add their own customizations, delaying the process.
* App developers are forced to try to accommodate users whose OS versions are way below the current version.
* Graphics are often slower.

2.4 History of Shopping

According to Alba *et al* (1997), shopping is an activity in which a customer browses the available goods or services presented by one or more retailers with the potential intent to purchase a suitable selection of them.

A [typology of shopper types](https://en.wikipedia.org/wiki/Retail#Shopper_profiles) has been developed by scholars which identifies one group of shoppers as recreational shoppers,  that is, those who enjoy shopping and view it as a [leisure](https://en.wikipedia.org/wiki/Leisure) activity.

Before there was currency, there was trading. According to archaeological finds, around 1200 BC, traders used cowrie shells as tokens similar to coins to mark the value of commercial transactions. as History of Retail Objects’ website points out. Coins later replaced cowrie shells as the currency of choice. Archaeological evidence and ancient manuscripts tell the story of ancient civilizations who used coins like Roman drachma and denarii (Davis, 2013).

According to the Encyclopedia of Ancient History, “shopping” likely consisted of purchasing necessities; for the rich, cowrie shells or gold coins would have also been used on luxuries to impress visiting dignitaries. Even then, the biggest “retail” system of which we have specific knowledge of included the agora–the Roman trading center where individual tradesmen would set up shop to offer their goods.

Shopping centers have been in existence since ancient Greek and Roman eras. In ancient Greece, the central shopping area of a city was called the agora and consisted of a large, open area in which merchants could display and sell their products. The first established shopping mall was built by [Emperor Trajan](http://library.thinkquest.org/26907/emperors/trajan.htm) in Rome nearly 2,000 years before the modern shopping center. This market was constructed in Trajan’s forum and consisted of a semi-circular building with a large vaulted hall, resembling a basilica, on top. The Emperor’s market included an array of shops, all of which faced a corridor, allowing customers to view the products and goods for sale. This center also featured several restaurants and bars, an important precursor to today’s mix of stores and shops included in shopping centers. The market consisted of several levels and more than 150 outlets that sold a wide variety of products including luxury clothing, silks, spices, and fresh food. Trajan’s Forum and the market were built nearly 2,000 years before the first recognized modern shopping center (Davies, 2013).

### 2.4.1 United States of America

1920-1940

The modern shopping center, which includes the small suburban strip center as well as the million-square-foot superregional malls, originated in the 1920s. In California, grocery stores would serve as the anchor store for a collection of smaller surrounding stores.  [According to Feinberg](http://www.acrwebsite.org/volumes/display.asp?id=7196) (n.d), shopping centers in the United States began in 1907, in a Baltimore neighborhood where a group of shops established off-street parking for their customers.  J.C. Nichols of Kansas City, Missouri is generally credited with the idea of establishing a shopping district away from a downtown. In 1922, Nichols’ [Country Club Plaza](http://www.pps.org/great_public_spaces/one?public_place_id=370) was constructed as the business district for a large-scale residential development. The Plaza featured a planned and unified architecture, a paved parking lot, and a variety of stores which were managed and maintained as a single unit. In 1928, Grandview Avenue Shopping Center in Columbus, Ohio was opened. This large center included nearly 300 stores and a parking lot which could accommodate nearly 400 cars.

Many historians consider Highland Park Shopping Village in Dallas, Texas to be the first actual planned shopping center. The Village was developed by Hugh Prather in 1931 and included a collection of stores built with a unified structure and theme (Davies, 2013).

1940s-1970s

The late 1930s and 1940s witnessed the establishment and expansion of the commercial chains Sears Roebuck & Co. and Montgomery Ward. These stores were set far away from large cities, accompanied with parking lots with ample space. In the early 1950s, Northgate in Seattle, Washington featured branches of major downtown department stores surrounded by smaller commercial shops. Soon after, Shoppers World in Framingham, Massachusetts was the first two-level center in the United States and was established in 1951. Later, in 1954, Northland Center in Detroit, Michigan utilized the “cluster layout,” which consisted of a single department store at the center and a collection of smaller stores surrounding it. Important features of Northland included surrounding parking lots and central air-conditions and heating.

In 1976, the Rouse Company developed Faneuil Hall Marketplace in Boston, Massachusetts which revolutionized the history of the shopping center. This Market was the first festival marketplace built in the United States. Festival marketplaces typically centered upon food and retail specialty items. Other similar marketplaces followed in locations such as Baltimore, New York and Miami. With the opening of Water Tower Place in Chicago, Illinois in the late 1970s, the shopping center industry had returned to a more urban location(Davies, 2013).

1980s-2008

More than 16,000 centers were built between 1980 and 1990. This decade was characterized by the increase in superregional shopping centers, which were malls measuring more than 800,000 square feet. Factory outlet centers became increasingly popular throughout 1990s. Outlet malls provided manufacturers with the opportunity to sell their own goods at discounted prices.

In addition to factory outlets, entertainment centers played a crucial role throughout the 1990s. Seeking to incorporate forms of entertainment, shopping centers began to offer a variety of activities, such as children’s play areas, live music, movies in large, multiplex cinemas, food courts, amusement parks, merchandising techniques, robotic animal displays, and other interactive demonstrations.

### 2.4.2 Europe

Renaissance Europe Markets and Fairs

During turn-of-the-century (16th century) Europe, shopping took on a new form as expansive marketplaces and fairs were developed as fundraisers and temporary celebrations. One of the largest was the Foire St. Germain in Paris, a fair for the benefit of the Abbey of St. Germain, established by Louis XI in 1482, according to History of Retail Objects’ website.

This fair lasted a few weeks around Easter. Tradesman would haggle, auction, and raffle off items for the benefit of the famous Abbey. Residents of Paris enjoyed something much like an early shopping mall. Similar fairs gained popularity and spread throughout the developed Western world.

The Industrial Age and Birth of Mass-Produced Goods

With the nascent industries of the Industrial Age came the now-familiar mass-produced products that contributed to the fast, widespread industrialization of shopping culture and the birth of the retail industry as we know it. Businessmen began to see the promise in opening general-merchandise stores to profit from the now-available needed goods.

Specialty shops, urban galleria, and department stores were born in the 18th and 19th centuries, giving rise to a culture accustomed to being able to find anything they needed at the drop of a hat. In the late 1800’s, Montgomery Ward’s first mail-order catalogs and the new invention of the cash register gave shoppers more options than ever before, according to Acudesignservices.com.

### **2.4.3 Current state of things globally when it comes to shopping**

Today, retailers can do almost all of their business virtually, with online and mobile shopping and complete [software tools like Netsuite](http://www.netsuite.com/portal/industries/retail/pos.shtml) which controls inventory, accounting, point of sales etc. Consumers can step into stores filled with European styles, Asian electronics and kitschy decor items developed by tradesmen in third-world countries. They can hop on their cutting-edge laptops or smartphones and purchase items from vendors a world away, expecting delivery within as little as one day. They can read experts’ speculations on the potential future of technological revolution while browsing for the perfect gift for their wives, brothers, and children (Jon, 2010).

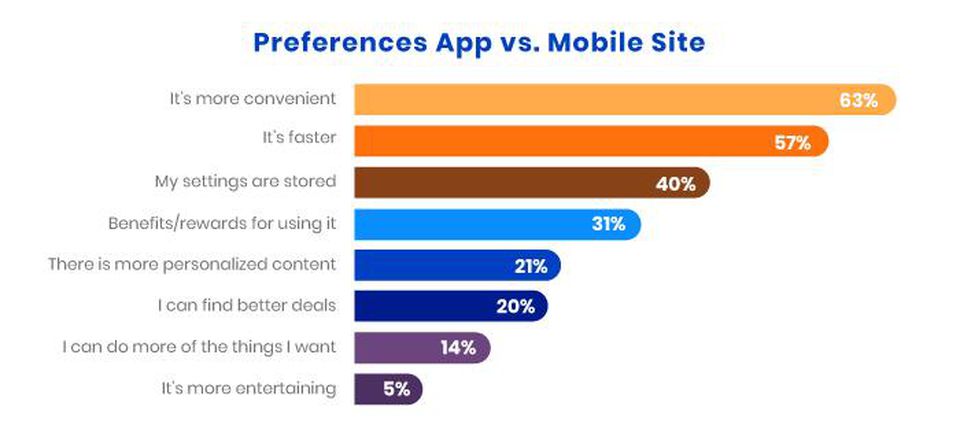
## 2.5 Why Technology is Being Integrated in the Shopping Sector

The Internet has revolutionized nearly every industry and, as technology has continued to advance to make online connectivity even more widespread, mobile tech has played a significant role in the revolution.

The shopping business sector is no exception when it comes to adopting mobile technologies to modernize their industry. To retain current customers and win over new ones, shopping store are implementing solutions that offer unique mobile experiences for users in order to provide convenience, grow brand awareness, and build customer loyalty.

The general uptrend in smartphone users lately is pushing online retailers to rethink their strategy and take a closer look at their customer's preferred way of shopping. It is much easier to reach for your phone, push a few buttons, and make a purchase, than logging into an account on a desktop. As a result, the number of smartphone users is expected to increase from 2.32 billion to 2.87 billion by 2020.

People who prefer to shop online do that because it is convenient, which is why they will consider which device better suits their needs. Take a look at the chart below that compares mobile apps to desktops and displays why some users prefer shopping on their phones:



###### Figure 2.10

Source:https://www.forbes.com/sites/quora/2017/12/19/why-many-online-shopping-sites-are-becoming-mobile-shopping-apps/

### 2.5.1 Benefits of integrating mobile applications in e-commerce

* Customers tend to spend 3 to 4 times longer in mobile apps than desktops.
* Mobile apps have lowest shopping cart abandonment rate: apps - 20%, desktops - 68%, mobile sites - 97%.
* Customers spend twice as much money on mobile apps than on desktops or mobile sites.
* The average order value on mobile apps is 140% higher than on mobile sites and 130% higher than on desktops.
* Customers are twice as likely to return to a mobile ecommerce app within thirty days than to desktop.

Each of the listed benefits are the reason why many retailers prefer to shift their business to mobile apps. In reality, an app has much more to offer, these are just some key advantages an ecommerce store can experience.

The following are reasons why technology, especially mobile solutions is being implemented in shopping stores:-

### Keeping up with competition

Many retailer chains are leveraging the power of mobile technology to keep up with competitors. Modern customers use mobile as a means of accessibility, as well as convenience. Businesses who are able to adapt to the changing world and provide seamless shopping experiences by means of integrating mobile solutions are more likely to convert users into loyal customers.

**Enhancing the brand**

By providing tech-savvy shoppers with a mobile, interactive, and high-velocity customer experience, retailers increase their chances of creating a lasting effect that amplifies their brand, while also resonating with the user.

**Data-driven marketing**

Сustomers are more value-driven than price-driven nowadays. Providing them with easy access to the necessary information about a product will increase the chances of them purchasing it.

A shopping mobile app can give customers this necessary data, while simultaneously collecting analytics to help businesses adapt their offerings to consumer needs. Data-driven marketing will enable retailers to monitor and plan product supply based on customer feedback and demand.

**Personalization**

Digitalization has drastically transformed the way shopping is done, making it more accessible than ever. By means of mobile sites and apps, businesses give their customers the opportunity to have individual and unique shopping experiences in which every consumer is at the center of their own purchasing scenario. In this way, retailers can engage with customers on a more personal level, giving them 24/7 access to products and helping them curate their shopping experience.

**Integrating the mobile app with the in-store experience**

Brands can utilize mobile apps to enhance in-store shopping experiences, making them more efficient and enjoyable for regular customers, as well as first-time shoppers. For example, retailers could use mobile apps to provide instant, detailed product information or to help customers find the location of a specific item. Apps that use location would also be able to provide alerts for special offerings as a buyer is walking down the aisle, providing extra value to the shopper while making their shopping experience more pleasant.

**Mobile payments**

Various mobile payment systems are available today in different countries and regions. Since many people are moving away from credit cards to perform transactions via apps, a mobile payment system can be easily integrated into an app to help deliver a seamless shopping experience. An integrated mobile payment system adds convenience, as well as the ability to serve more customers who want to order and purchase their items directly through an app instead of in-store.

**Getting feedback from buyers**

To better understand the way a supermarket or shopping mall operates, apps can be utilized to collect feedback from shoppers. Retailers could enable customers to send requests or provide reviews for items which will help them improve their offerings.

Accumulating data directly from customers will allow different locations to tailor their in-store shopping experience, while helping them monitor the overall demand and supply of a particular store.

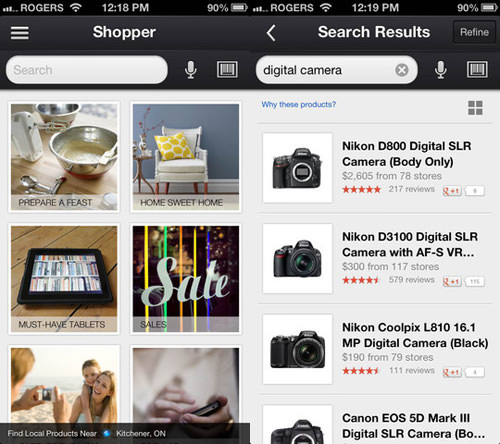
## 2.6 A Look At Past And Present Shopping Systems Already In Use

As we have seen above, both mobile application development and shopping activity have a great and rich history. Individual developers and programmers and also companies have in the past developed systems such as mobile applications, websites and stand-alone systems that make shopping much more easier, time efficient, more manageable in terms of tracking finances and expenditure rates and trends and actually more fun and engaging. You can now check for the latest prices in multiple catalogs, availability of products, comparison of prices, special promotions, discounts, reviews and lowest prices on your smartphone. This section looks at such systems that have been around.

### 2.6.1 Globally

**Google Shopper**

Google Shopper provides information like prices, reviews, videos and more of millions of products right on your smartphone, recognizing them via barcode, voice, text search or even cover art. You can easily make the best purchasing decision by comparing prices in different online stores.

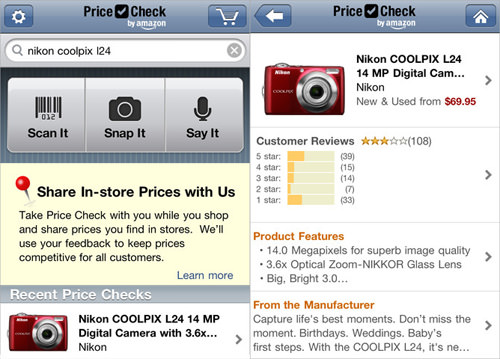


###### Figure 2.11

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

**Amazon Price Check**

Users can use its built-in scanners to scan barcodes, upload a picture of a product, speech or text search to get the product descriptions, find customer reviews and the lowest prices available from Amazon.com and its merchants.



###### Figure 2.12

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

**ShopSavvy**

Use ShopSavvy to compare prices across a number of online and local stores. You can search for products manually or even scan a barcode before you make a purchase in a store. Comparing prices like this is the easiest way to spend less when you shop.

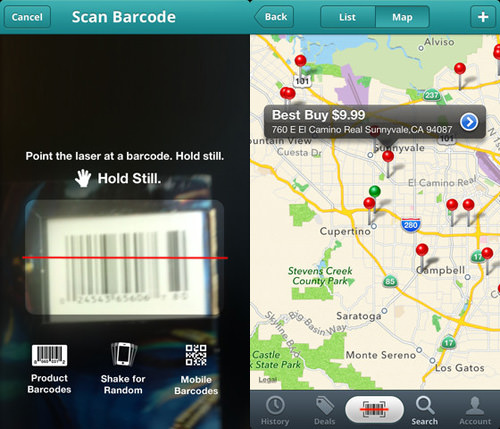
Here's how it works: open the app and either search for the product or use the scanner to scan the barcode. Immediately, you'll see the cheapest price both online and in-stores, and then you can select either to see specific retailers that are offering that item for a cheaper price.

Select an online store, and you'll immediately be taken to the product page where you can buy it - there's an option to see just new products or both new and use items. If you choose a local store, you can start navigating there or open the store's website.

If you sign up with ShopSavvy, you can even get cash back on some purchases made through certain retailers.

You can also save items in ShopSavvy so that you can get price alerts when the price changes. There's also a list of related products that shows up below the one you're looking at.

The home page of this app features the best new deals for your favorite brands, which is yet another way to find deals through ShopSavvy.

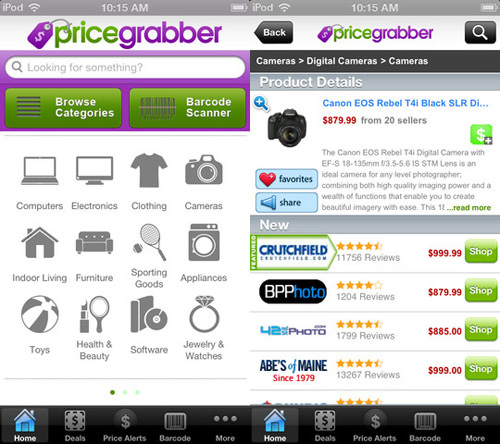


###### Figure 2.13

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

#### [PriceGrabber](http://www.pricegrabber.com/ipad-iphone-android-app.php)

Another app that allows you to find product information, compare prices and get merchant ratings. Searched results can be sorted by price, rating or popularity.

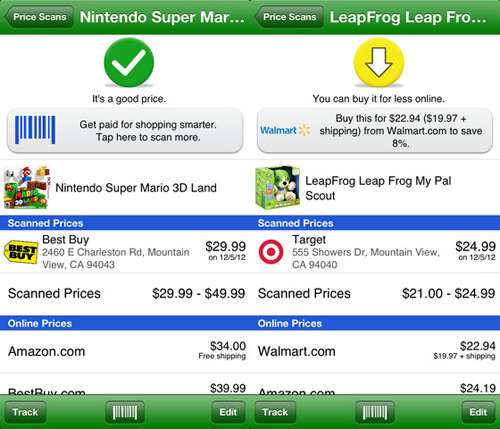


###### Figure 2.14

[Source: https://www.hongkiat.com/blog/mobile-shopping-apps/](http://www.smoopa.com/)

#### [Smoopa](http://www.smoopa.com/)

Join other shoppers to figure out what the really good prices are. This is a comparison app which lets you join foces with other shoppers to scout for the best prices for your favorite products.

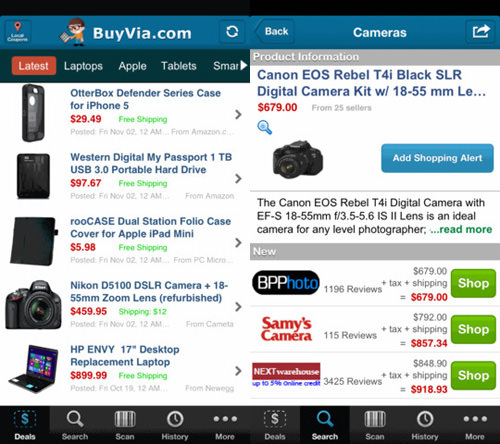


###### Figure 2.15

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

#### [BuyVia](http://www.buyvia.com/apps-downloads/)

Get the lowest price on any products from both online and local stores with BuyVia. It comes with a UPC Code and QR code scanner for you to find the best prices. Users can customize a shopping alert to receive notification when the price of a product falls.



###### Figure 2.16

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

#### [Milo](http://milo.com/about-us#mobile)

Milo has a big database for you to check the pricing and availability for over 3 million products in local stores. Users can also pay for an item from the palm of your hand and pick it up at a local store.

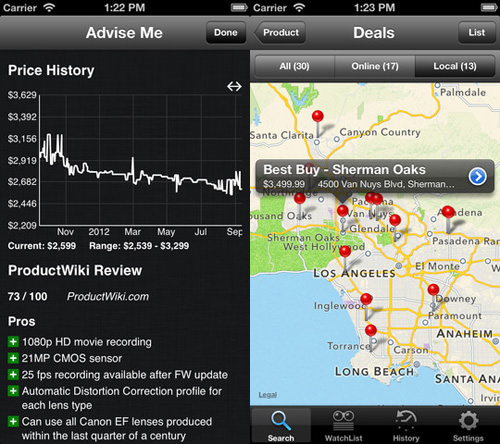


###### Figure 2.17

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

#### [Shop Advisor](https://www.shopadvisor.com/)

Shop Advisor lets you add products you want to your personal WatchList and it will send you a notification whenever the ‘price is right’.

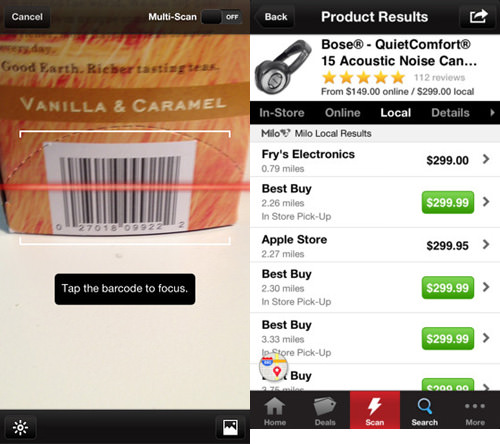


###### Figure 2.18

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

#### [RedLaser](http://redlaser.com/)

RedLaser is a shopping assistant with barcode and QR code scanners to get the best price of a product and the nearest local stores it is available in. Besides that, users can search for deals and coupons from many merchants and buy the product online via RedLaser.

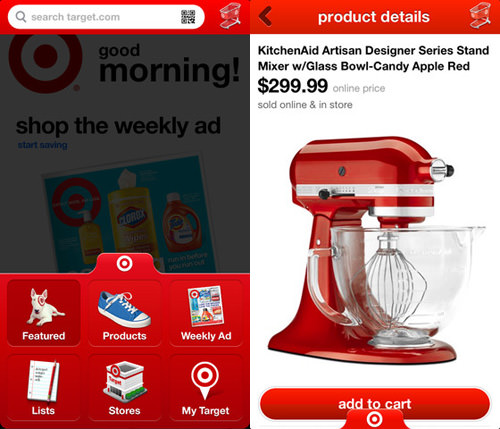


###### Figure 2.19

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

#### [Target](http://www.target.com/spot/mobile/landing?ref=sr_shorturl_anywhere)

With Target, you can create a product TargetLists to stay alert with its pricing. You can also design your shopping list, get special offers, check prices and buy a product using GiftCards and redeem coupons right from your smartphone.



###### Figure 2.20

Source: https://www.hongkiat.com/blog/mobile-shopping-apps/

**The Hunt**

Available for free on Android and iOS, The Hunt is a social shopping app. On this platform, you can exchange style advice, discover new trends and share your own shopping tips. You can post photos of something you spotted on the street and the community will help you find and buy it. You can also poll the community, asking if something is worth a purchase. Plus, you can buy things you find on the app or save them to your wish list.

#### Amazon Go

#### Amazon is a global commerce giant, and it isn’t slowing down anytime soon. The launch of Amazon Go is Amazon’s attempt to disrupt the grocery store industry with its new cashier-less way of shopping.

The “Just Walk Out” campaign introduced this new kind of store in which customers can shop and leave, no check out required. Amazon says it’s the world’s most advanced shopping technology, made possible by the same types of technologies used in self-driving cars: computer vision, sensor fusion, and deep learning. This technology automatically detects when products are taken from, or returned to, shelves and keeps track of them in a virtual cart. When a customer is done shopping, they can just leave the store. Amazon charges their account for the items they’ve walked out with and sends them the receipt.

### 2.6.2 Current shopping Systems in Kenya

**OLX Kenya**

An online marketplace that provides a platform for peer-to-peer selling. Anyone can buy and sell just about anything (legal, of course) on OLX; from cars, furniture, electronics, even livestock. Since items here are often used or second-hand, prices are generally cheaper. Unlike other online stores, prices on OLX are negotiable between the buyer and the seller.

Jumia Kenya

Jumia is still, without a doubt the largest online shop in Kenya. The site has one of the biggest selections of mobile phones, tablets, computers & laptops, women’s fashion, men’s fashion and more.

[Kilimall](https://www.kilimall.co.ke/)

An excellent alternative to Jumia. They have a wider range of items, and cheaper alternatives from China.

[Bidorbuy](https://www.bidorbuy.co.ke/)

This is an e-commerce website based on an internet auction and online marketplace model allowing individuals and businesses to trade with each other.

[Cheki](https://www.cheki.co.ke/)

You can use Cheki to find new and used cars for sale.

[Chinabuy](https://www.chinabuy.co.ke/)

Kenyans use Chinabuy to purchase electronics, fashion, home and garden appliances from China. Buyers can find very high quality products from China at very decent prices through the use of this site.

### 2.6.3 How developed system will complement existing Shopping systems

The systems that have been in existence above clearly have the following amazing features,

* Cashless transactions
* Better offers
* Free delivery
* A rich range of options
* Discounts
* Better services being offered

One similarity in almost all of them is that they are online retail stores, where the user can use their smartphones to directly buy a product remotely and have it shipped or delivered. What the Shopping Tracker mobile app will offer differently from these other apps and systems and improve on them is that it will provide the user with the very vital service of always keeping track of their cash when they shop, hence sticking within a preset budget. This in addition to the platform for writing shopping lists will enable the user to always shop within their previous set scope and reduce the impulsive purchase of items. Also integrating the normal calculator inside the app makes it easily accessible for quick random use, without the user ever leaving or closing the application.

## 2.7 Chapter Summary

This chapter presented a brief history of smartphones, Android OS and application development, Shopping as an activity: its history and current trends. The chapter also looked at the currently existing mobile shopping applications and online sites globally and also locally in Kenya and how the Shopping Tracker will be able to improve on these current systems. The next chapter will deal with the methodology used in the development of the system.

# 

# CHAPTER 3

# METHODOLOGY

## 3.1 Chapter Overview

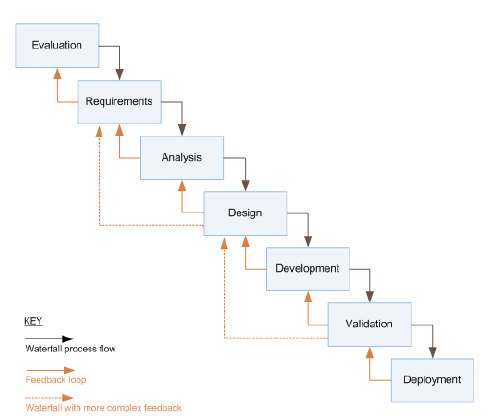
This chapter focuses on the suitable methodology to be used in the development of the Smart shopping mobile application system, the different data collection instruments, the design of the study, the study area, the target population involved, Sampling size and sampling procedures. The analysis techniques to be applied on the data are also included.

3.2 Choice of Methodology

I compared the traditional waterfall model and the Rapid Application Development (RAD) model, checked on their advantages and disadvantages and selected the most suitable and efficient model that will aid in the development of a reliable and stable Smart Shopping Mobile Application.

### 3.2.1 The Waterfall Model

The waterfall model, also known as the cascade model, was first documented by Benington, in 1956 and modified by Winston Royce in 1970. It has underpinned all other models since it created a firm foundation for requirements to be defined and analyzed prior to any design or development. It consists of stages that are cascading from one to another. (see figure 21 below). One development stage is completed before the next begins. The waterfall model brings out a very high-level view of activities taken place during development, and gives developers the sequence of events they should expect to encounter.



###### Figure 3.1 Waterfall model

Source: Software Development Lifecycle models, article in ACM SIGSOFT Software Engineering notes May 2010.

**Challenges**

1. The main challenge is that it does not reflect how the software is really developed.
2. Secondly, the model provides no guidance on how to handle changes that are likely to occur in the course of development.
3. It fails to treat software as a problem-solving process, it only presents a manufacturing view of Software development (Curtis, B.,Krasner et al., 1987).

**Advantages**

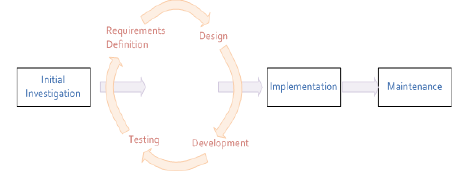
* The model is simple and easy to understand.
* Phases are processed and completed one at a time. Phases do not overlap.
* It works well for projects where requirements are well understood.
* It is manageable due to the rigidity of the model-each phase has specific deliverables and a review process.

**Disadvantages**

* Once an application is in the testing stage it is difficult to go back and change something.
* No working software is produced till late in the lifecycle.
* Poor model for long term projects.
* It is not suitable for projects where requirements are at a moderate to high risk of change.
* High amounts of risks and uncertainty.

### 3.2.2 Rapid Application Development

Principally developed by James Martin in 1991, Rapid Application Development (RAD) is a methodology that uses prototyping as a mechanism, as per Figure22, for iterative development.



###### **Figure 3.2**: Prototyping approach used by RAD.

Source: Software Development Lifecycle models, article in ACM SIGSOFT Software Engineering notes May 2010.

RAD promotes a collaborative atmosphere where business stakeholders participate actively in prototyping, creating test cases and performing unit tests. Decision making is devolved away from a centralized structure (usually comprising of the project manager and developers) to the functional team.

The open source software development model,also known as the Cathedral and the Bazaar model that was first documented by Raymond (2001), espousing a ‘release early; release often; listen to your customers’ philosophy, is quite similar to RAD and some of its spin-off methodologies such as Agile.

Recently, RAD has come to be used in a broader, generic sense that encompasses a variety of techniques aimed at speeding software development. The following are some of the RAD techniques.

**Agile**

Scope changes, as well as feature creep, are avoided by breaking a project into smaller sub-projects. Development occurs in short intervals and software releases are made to capture small incremental changes.

Applying Agile to large projects can be problematic because it emphasizes real-time communication, preferably on a personal, face-to-face basis. Also, Agile methods produce little documentation during development (requiring a significant amount of post-project documentation) whilst de-emphasizing a formal process-driven steps

**Extreme Programming (XP)**

With XP, development takes place incrementally and on the fly with a business champion acting as a conduit for user-driven requirements and functionality; there is not an initial design stage. In order to lower costs, new requirements are accounted for in short, fast spiral steps and development takes place with programmers working in pairs.

**Joint Application Development (JAD)**

JAD advocates collaborative development with the end user or customer by involving him during the design and development phases through workshops (known as JAD sessions). This has the possibility of creating scope creep in a project if the customer’s requirements are not managed well.

**Scrum**

Development takes place over a series of short iterations, or sprints, and progress is measured daily. Scrum is more suited to small projects because it places less emphasis on a process driven framework (needed on large projects) and also the central command structure can cause power struggles in large teams.

**When to use RAD Methodology**

* When the system needs to be produced in a short span of time (2 – 3 months).
* When the requirements are known.
* When the technical risk is less.

**Advantages**

* Flexible and adaptable to changes.
* Due to code generators and code reuse, there is a reduction of manual coding.
* Due to prototyping nature, there is a possibility of lesser defects.
* Each phase in RAD delivers highest priority functionality to client.
* Increased speed of development.

**Disadvantages**

* When technical risk is high, it is not suitable.
* If developers are not committed to delivering software on time, RAD projects can fail.
* Reduced scalability occurs because a RAD developed application begins as a prototype and evolves into a finished application.
* Reduced features.

The suitable methodology to use for the development of the proposed system will be the RAD model. This is because of the prototyping approach it uses as a mechanism for iterative development. The RAD technique approach used is scrum.

## 3.3 Data Collection Instruments

The data collection tools/instruments selected for this study include Questionnaires, interviews and observation. These tools will be used in the requirement gathering stage, after which the information and data collected will be used for the design of the system and eventually its actual development. The following data collection tools are discussed in detail:-

### 3.3.1 Questionnaire

Questionnaires were issued a sample selected from the general population around Moi University. The main objective of issuing them out was to collect reliable information on the state of shopping without the aid of a smart system.

### 3.3.2 Interview

Interviews were conducted on a few selected individuals who consisted of regular shoppers, supermarket counter attendants, and supermarket managers. The interview questions were both open ended and closed to give sufficient and variety of information.

### 3.3.3 Observation

A local supermarket was visited to observe the shopping activity of different shoppers. The main aim was to check on the number of individuals who either were forced to remove excess items from their shopping at the counter or those who realized they had more cash and went back to add more items.

3.4 Design of the Study

The research uses an approach that gives a detailed account and description of the problem in existence. This is achieved by conducting a detailed investigation in the shopping areas such as supermarkets and shopping malls. This approach focuses on the defined area for which this proposed system will be designed and developed.

## 3.5 Study Area

The study was based around the Moi University main campus school area, located in Uasin Gishu county, kesses constituency. The research is mainly concerned with supermarket shoppers and counter cashiers. This study area was selected due to the strategic location of the supermarket, in the university locale. This means that getting permission to interview, observe and distribute questionnaires was easy because of the nature of the learning institution.

## 3.6 Target Population

The total population selected for the research was 50 people. This comprised of general shoppers, counter cashiers and supermarket managers.

###### Table 3.1: Sampling Frame

|  |  |
| --- | --- |
|  | **Number of People** |
| General shoppers | 30 |
| Counter Cashiers | 15 |
| Supermarket Managers | 5 |
| **TOTAL** | **50** |

## 3.7 Sampling size and sampling procedures

Interviews were given and conducted to 30% of the population,while Questionnaires were conducted on 45% of the population. The study also adopted the simple random sampling method to issue these questionnaires and also to select those people who will be interviewed.

###### Table 3.2: Sampling Design

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Population** | **Sample Size for Interviews** | **Sample Size for Questionnaires** |
| General Shoppers | 30 | 30% of 30 = 9 | 45% of 30 = 13.5 therefore approximately 14 people |
| Counter Cashiers | 15 | 30% of 15 = 6 | 45% of 15 = 6.75 therefore approximately 7 people |
| Supermarket Managers | 5 | 30% of 5 = 1.5 , therefore approximate approximately 2 | 45% of 5 = 2.25 therefore approximately 2 |
| **TOTAL** | **50** | **17** | **21** |

## 3.8 Data Collection Procedures

The school drafted an introductory letter to enable me seek permission to conduct my research in the local supermarkets. The respondents were also assured that the information they gave was for research purposes only. To emphasize confidentiality, the questionnaire and interview questions did not require the user to include any personal information like names, ID numbers, phone numbers etc.

3.9 Validity and reliability of research instruments

The research tested the validity and reliability of the methodology used. Reliability was tested by administering questionnaires and getting feedback.

3.10 Data analysis techniques

The data and information collected using the data collection instruments was analyzed, sorted, categorized and given meaning in order to be used to design the actual application system.

From the analyzed data and information, the following design instruments were used to come up with the architectural design of the system:- Data Flow Diagrams (DFDs), Use case Diagrams, class diagrams, Activity diagrams, sequence diagrams, entity relationship class diagrams.

3.11 Chapter Summary

This chapter looked at the methodologies that could have best been used to develop the Smart shopping mobile application system. The data collection instruments, the design of the study and the study area were clearly outlined. The target population involved and Sampling size and sampling procedures was elaborated to show the reliability of source of the data and information. The analysis techniques to be applied on the data were also presented and clearly outlined.

# CHAPTER 4

# SYSTEM ANALYSIS AND DESIGN

4.1 Chapter Overview

This chapter presents analysis of the requirements collected, in order to identify the system’s objectives and make sure that all components required are designed to accomplish their purpose.

The actual design of the system looks at how to accomplish the objectives of the system. From the design the actual system is developed using a software CASE tool appropriate to the project.

The purpose of this phase is to transform the requirements of the system from the requirements analysis phase into design specification for its ultimate construction.

The Unified Modeling Language (UML) was used to model the data and information from the requirements obtained.

4.2 System Analysis

### 4.2.1 Overview of the Proposed System

The main aim of the proposed app is to automate shopping experience for users. This will in turn save on time and enable proper management of shopping as an activity, making it easy to track past, present and even predict future shopping patterns.

The app will consist of three main components: The Automatic Item Barcode Scanner, the calculator and the shopping notepad.

The Automatic Item Barcode Scanner will enable the user to scan item’s name, quantity and prices as one picks items from shelf to shelf. These details will be auto added in a list with each item being scanned recorded real time. The user at any one point will know what has been bought, the amount spent, and the amount of money remaining.

The Shopping notepad is to enable the user keep track of their shopping lists over time.

The Calculator is to enable the user make basic arithmetic fast and easily accessible.

In a nutshell, this application will be like a shopping assistant.

**Architectural Design**

A structural framework showing the major components the system is shown in the model below.

Input

System

Database

User GUI

.Validation

.Conditions

.Permissions

SQLite DB

Admin GUI

###### Figure 4.1 System’s architectural design

The System above is structured with three main components: The input, the processing system and the database.

**The Input and Output Design**

The Shop Smart application’s input and output interfaces were designed and developed with the user in mind. Simplicity, ease of use, user friendliness and consistency of command buttons were all considered and factored in.

**Processing System Design**

The processing system is robust, reliable and developed using well thought of algorithms. The possible conditions that a user can possibly undergo while shopping are well thought of and implemented.

For the Automatic Item Barcode Scanner, a library developed by Google to capture the barcode details of an item is imported into the code of the system. A database that contains all of the supermarket item’s barcodes is imported into the app through a Local area Network connection through wifi technology. Once an item’s barcode and the database barcode match, the details of this item i.e the name, quantity and price is fetched and displayed in the Scanned Barcode list in the app.

The calculator’s design is used to enable user to perform normal arithmetic calculations on the app.

**Database Design**

The database is designed using a SQL based language for android called SQLite. The user while using the app occasionally sends data to the database and retrieves it while scanning item’s barcodes. So a good reliable, fetch and post commands implemented in the database makes the app very robust and stable.

The administrator has access to the database and its design, and he/she can make changes and updates.

**Data Flow Diagram (DFD) for the proposed system.**

Statistics

Admin

Queries

User

Smart Shopping App System

Updates

Statistics

Report

###### Figure 4.2 Data Flow Diagram

The system has two entities that interact with the system, general users who use the app as a shopping assistant, and an administrator who performs updates on the system.

A third minor entity is the report, which is generated automatically by the application system.

### 4.2.2 Intended users

After conducting analysis through interviews and observations, the general public, in this case anyone who shops in a supermarket is a main user of the application. The user requirements and needs will therefore play a great role in development of the system.

### 4.2.3 Proposed system functions

The app will consist of three main components: The Automatic Item Barcode Scanner, the calculator and the shopping notepad.

The Automatic Item Barcode Scanner once opened will prompt the user to enter an amount of money in a text field and submit it to the system database for processing. This amount represents the money that a user wants to spend while shopping. The Automatic Item Barcode Scanner will then enable the user to scan prices of items as one picks items from shelf to shelf. The prices will be auto added in a list with each item being scanned recorded real time. The user at any one point will know what has been bought, the amount spent, and the amount of money remaining by viewing a list.

The user will be able to remove an item from the list very easily by just pressing a button.

The Shopping notepad is to enable the user keep track of their shopping lists over time. The user types a shopping list prior to the actual shopping. This makes it easy for them to have a reference on what they want to buy and not be reliant on brain memory as they can forget.

The Calculator is to enable the user make basic arithmetic fast and easily accessible. Functions supported include, addition, subtraction, division and multiplication.

In a nutshell, this application will be like a shopping assistant.

### 4.2.4 Functional requirements

These are the features describing the behavior of the system, what it should do when a certain system component is being used or is activated.

#### 4.2.4.1 Administrator

The application system’s administrator has the rights and privileges of accessing all the functions and features in the system, which include: the database, the UI and logic codes and any information that the user accepts to be accessed by the app in the terms and conditions.

##### 4.2.4.1.1 Automatic Item Barcode Scanner

1. The administrator can update the Text field that takes in amount from the user. He implements validation rules.
2. The Barcode Scanner can be updated by the administrator to scan various elements and forms of data on an item. Administrator makes sure that the scan and rescan features work as desired and intended.
3. The administrator determines the point at which the list of scanned names and prices of items will appear, which is immediately after the first item. The administrator ensures that the button for viewing the full list as selected by the user works appropriately and displays the list.
4. The administrator ensures that if a user exits the application without pressing the “Done with shopping” button, then that state of shopping is usually saved and the user is prompted to continue from where they left off.
5. Once the “Done with shopping” button is clicked, the application closes. The administrator maintains this functionality.

##### 4.2.4.1.2 Shopping List pad

The administrator ensures that the options for working with the notepad which are creating a new list, opening an existing one, saving a list and deleting a list work properly. Administrator updates through adding or removing features.

##### 4.2.4.1.3 Calculator

The administrator performs tests on the calculator to ensure that validation rules are set and the components do not make the application to crush. Administrator makes sure that the four main components of arithmetic are working well: addition, subtraction, division and multiplication.

#### 4.2.4.2 Users

The general users of the application system have access to the front-end components of the main functions of the application which include the Automatic Price Scanner, the Shopping list pad and the Calculator. The functional requirements of the users are as follows:

##### 4.2.4.2.1 Automatic Price Scanner

1. User can enter amount they have in the text field provided, before they start using the OCR scanner.
2. Users can scan prices of items and rescan if the previous scan is not satisfying.
3. Users can view the lists of total scanned prices of items at any one time.
4. Users can click the “Done with shopping” button to exit the application after successful shopping.

##### 4.2.4.2.2 Shopping List pad

The user can open the Shopping List pad to type and save their shopping lists before they actually go for the shopping. The application system provides features such as create new list, open existing list, save a list and delete a list.

##### 4.2.4.2.3 Calculator

The application allows the user to have a very easy access to a calculator inside the app. The user can perform various arithmetic calculations such as addition, subtracting, dividing and multiplication.

### 4.2.5 Non-Functional requirement

These are performance characteristics of a system as well as the constraints that pose as a hindrance to the success of the proposed system.

1. **User-friendliness**

The application should provide a very user friendly Interface. Most of the users will not technical skills, so keeping the interface self-explanatory is mandatory. This is through well labeled buttons and fields, readable font size and type and use of eye friendly colors.

1. **Ease of Use**

The application should be very simple and easy to use. A well written and simple user manual and detailed documentation should be available to the users.

1. **Functionality**

The application is designed to specifically scan prices of items whole shopping and automatically adding them for the user. The system should be able to accomplish that without fail.

1. **Reliability**

This is the extent to which a program can be expected to perform its intended function with requirement precision(Pressman S. Roger, 2001). The system should be reliable and avoid any risk of failure. It should also be accurate to give the user confidence.

1. **Response time**

The system should have a very fast response time. Users should always have the uninterrupted time while using the application caused by hanging or just slow processing. This can create frustration in a user who can end up not using the system at all.

1. **Correctness**

This refers to how accurate the data and information is handled by the application system. The speed and timeliness wont matter if the data is always incorrect. Extensive testing should be carried out to ensure correctness.

1. **Security**

Security is one of the most vital things to have in mind as this system is developed. This is because we are handling personal data and information of a user such as their shopping items, and their shopping patterns, including their favorite shopping stores. The administrator should always ensure that they are the only ones accessing this data, and therefore implement and adopt a username name and strong password policy.

1. **Maintainability**

The system should be easily maintainable by the administrator/developers. It should be one that adopts to any environmental changes, any requirements changes and any errors encountered. The system should be designed and developed in such a way that recovering from an unexpected anomaly is easy and smooth without affecting features such as data security, speed and system’s ease of use.

## 4.3 System Design

### 4.3.1 Structure Design

The structure design of the Smart Shopping application system shows an overview of the entire system. General users can use the application to aid them in their shopping through scanning prices of items as they shop, use the shopping list pad to write their shopping lists and finally if they need to make quick calculations, use the application’s inbuilt calculator.

The administrator is able from the back-end of the system access all the features and update components of the application such as the Automatic Item Barcode Scanner, the Shopping List pad and the calculator.

#### 4.3.1.2 Unified Modeling Language (UML)

Unified Modeling Language (UML) diagrams are used in this project to show detailed structural design of the system. The diagrams include, a use case diagram, activity diagram, sequence diagram and a class diagram.

##### 4.3.1.2.1 Use Case Diagrams

The use case diagram is a representation of the system to show its capability as a whole. It comprises of the use cases that are within the system boundary and the actors that are outside the boundary that interact with the system. The actors in this case are the Shoppers as the primary actor and the Administrator as the secondary actor.

Use-Case Diagram

Open App

Enter Amount

Scan Price of Item 1

Update App

Scan Price of item2..

Correct and eliminate bugs

View List of Total Items Scanned

Admin

User

Pay using Mobile phone platform

Open Shopping list

Open Calculator

Close App

###### Figure 4.3 Use Case Diagram

##### 4.3.1.2.2 Activity Diagram

An activity diagram explains diagrammatically the activities of how the users of the system interact with the system. In their interaction, it aims to bring out the decisions made from opening the application to performing the last activity and exiting the application.

For the user, once the application is open, they have access to the various features and components of the application, determined by the amount of rights given to them by the administrator.

The administrator also has access and full rights to update various features and components of the application from the back-end, and also handle data from the users

Activity Diagram for the system Administrator

Start

Login

Valid?

No

Yes

Manage Barcode Scanner

Manage Shopping pad

Manage Calculator

Logout

End

###### Figure 4.4 Activity diagram for Administrator

Activity Diagram for the system User

Start

Open App

Open and use Barcode Scanner

Access and use Shopping Pad

Open Calculator

Close App

End

###### Figure 4.5 Activity diagram for user

##### 4.3.1.2.3 Sequence Diagram

A sequence diagram models the collaboration of objects based on a time sequence. It shows how objects interact with others in a particular scenario of a use case. It also shows sequence of messages among the objects. In the Smart shopping application, it shows the various collaboration of objects on the user side, right from opening the application to using all its features, and also on the administrator side, to show all steps from login in to managing the features of the application and making updates. All steps are in a sequential manner.

Sequence Diagram for the User

User

System

Database

Open App

Send request

Display the Barcode scanner, Shopping pad and Calculator modules

Avail to user past shopping data

Select Barcode Scanner

Open Barcode Scanner

Store amount in Database

Enter amount to be spent

Successfully stored.

Barcode scanning activity is opened

Store the details of an item represented by this barcode in the DB

Scan an Item’s barcode and press SUBMIT

Stored successfully/Failure message

Display an alert to the user that Item has been scanned successfully/failed to scan

Store the details of an item represented by this barcode in the DB

If failed, Rescan

Store in Database

Scan more items

Fetch historical shopping list data from DB

Open Shopping List Pad

Connect user to a shopping list module session

Display to user Shopping List pad GUI

Open a new or existing list to update or delete

Request to store a new list, update an existing list or delete in database

Sequence Diagram for the User (Cont..)

User

Database

System

Feedback to user on state of storing, deleting and updating shopping lists.

Alert that actions of storing, deleting or updating have been actualized by the system.

Close Shopping Pad session

Connect user to a Calculator module session

Open Calculator module

Use Calculator GUI in doing shopping related calculations

Close Calculator

Terminate database connection

Close App

Disconnect from session and send user to homepage

Close app and Send user to session they were in before they started using the App

###### Figure 4.6 Sequence Diagram for User

Sequence Diagram for the Administrator

Database

System

Administrator

Login with username and password

Confirm username and password

Success, credentials match

Successful login

Manage Barcode Scanner module

Store new updates and synchronize

Successfully updated

System is up to date

Store new updates and synchronize

Manage Shopping List module

Successfully updated

System is up to date

Store new updates and synchronize

Manage Calculator module

Successfully updated

System is up to date

Generate Reports

Request for reports

View and download reports

Reports generated successfully

Logout

Terminate connection to database

Log user out of the system successfully

Logout successfully

###### Figure 4.7 Sequence Diagram for Administrator

##### 4.3.1.2.4 Class Diagrams

A class diagram shows an overview of the target system. It describes the objects and classes inside the system and the relationship between them. It also shows a wide variety of usages: from modeling the domain-specific data structure to detailed design of the target system. With the share model facilities, you can reuse your class model in the interaction diagram for modeling the detailed design of the dynamic behavior.

In the Smart Shopping application system we have four different classes that collaborate and work together in the system. The classes include: The Barcode Scanner, Calculator, Notepad and the system itself. These classes have different functions and methods within them that show how the different classes interact.

**Class Diagram**

System

Barcode Scanner

Shopping List

-Open()

-close()

-Amount

-Scanned list

-note 1

-note2

-note..n

+Scan()

+Rescan()

+pay()

+new entry()

+save()

+delete()

+open()

Calculator

-first number

-secondnumber

-result

+calculate()

###### Figure 4.8 Class Diagram

## 4.4 Smart Shopping Mobile Application System Implementation

### 4.4.1 Tools used

The tools to be used to develop the android-based application are very critical to consider. This is because of factors such as performance, storage, size of the application, security, portability etc. It is therefore important to be careful while choosing the appropriate tool.

#### 4.4.1.1 Software requirement

This application is designed and developed in the Android IDE called Android Studio. The user interface is therefore done using the extensible Markup Language (XML).

The logic of the application is done in the Java programming language, which is one of the languages supported by the IDE Android studio.

#### 4.4.1.2 External Libraries and APIs

The system imports and makes use of very useful software libraries such as the Zxing Library (Zebra Crossing), for enabling capability of making the phone camera behave as a barcode reader.

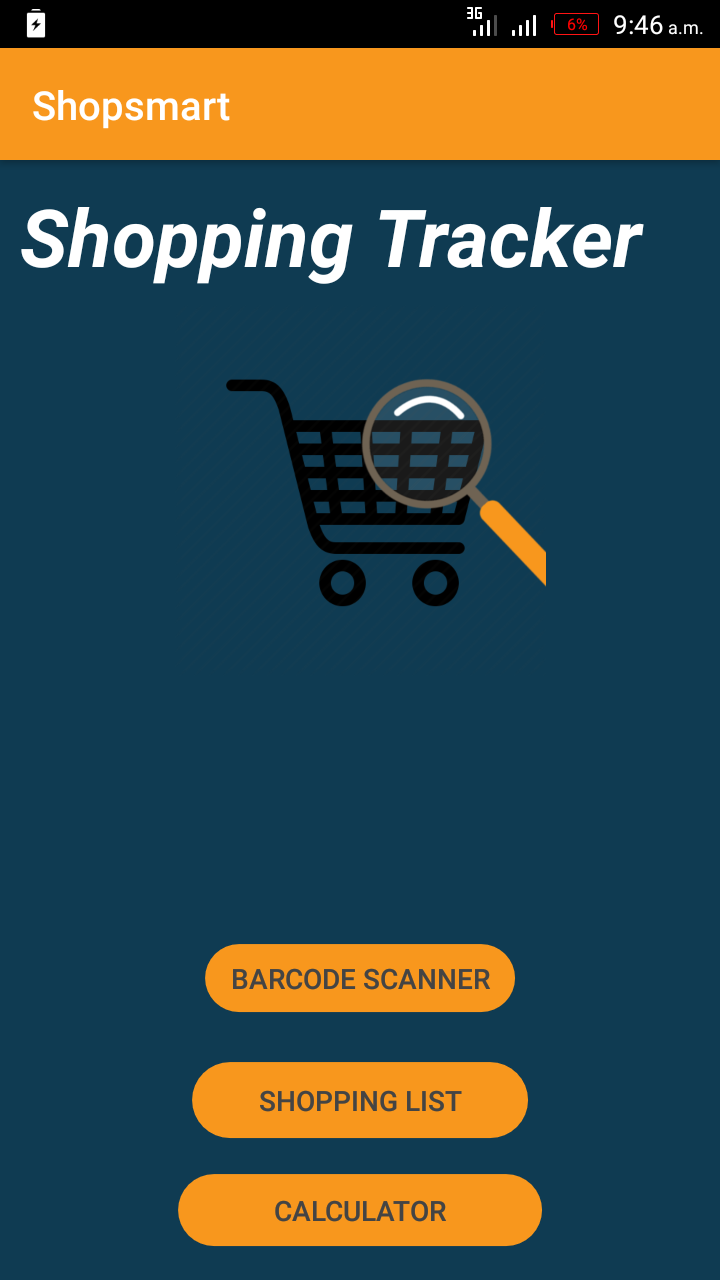
Other libraries and APIs include:

**4.4.1.3 Database management**

SQLite Database

* 1. **Graphical User Interface Home tab/activity**

The home tab is displayed when the app is opened. It contains three buttons from which the user can access the three modules of the system.

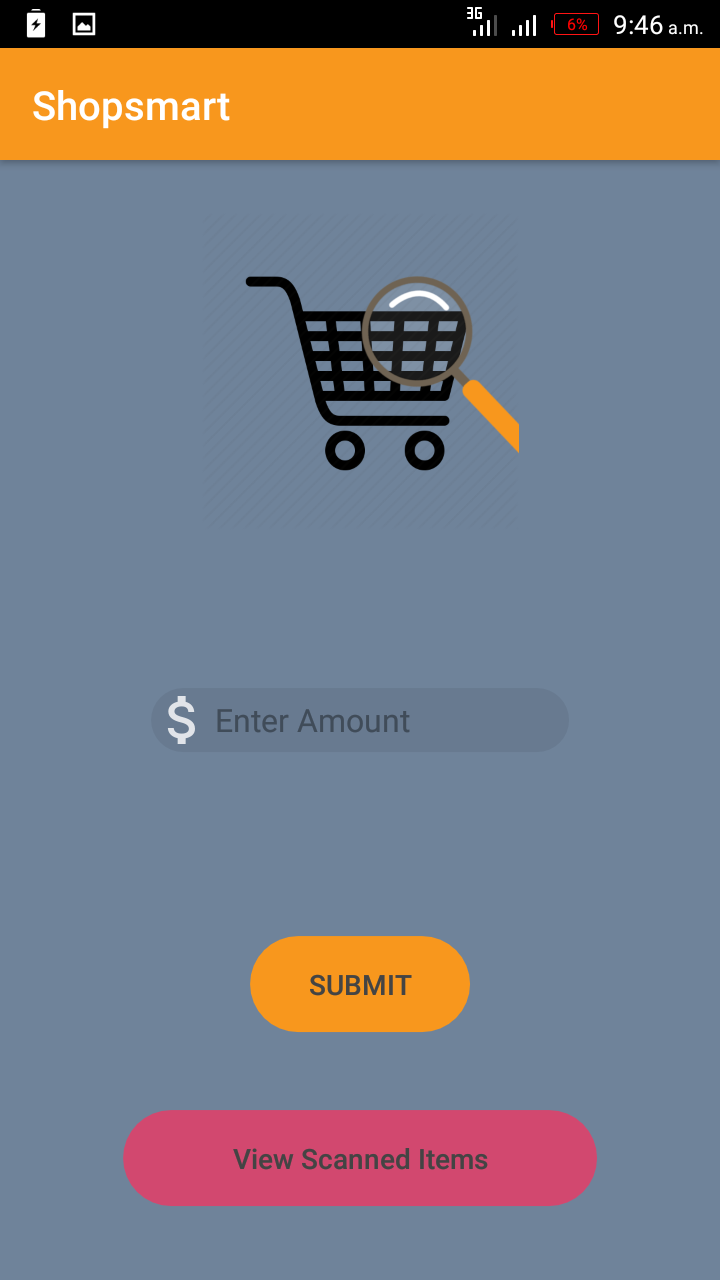


###### Figure 4.9 Home tab

* 1. **Enter Amount Tab**

After clicking the Barcode Scanner module, the user is prompted to enter an amount of money to be used in shopping. The system will manage it for them.

The user can view the shopping list from this page/tab.



###### Figure 4.10 Amount tab

* 1. **Barcode Scanner Tab**

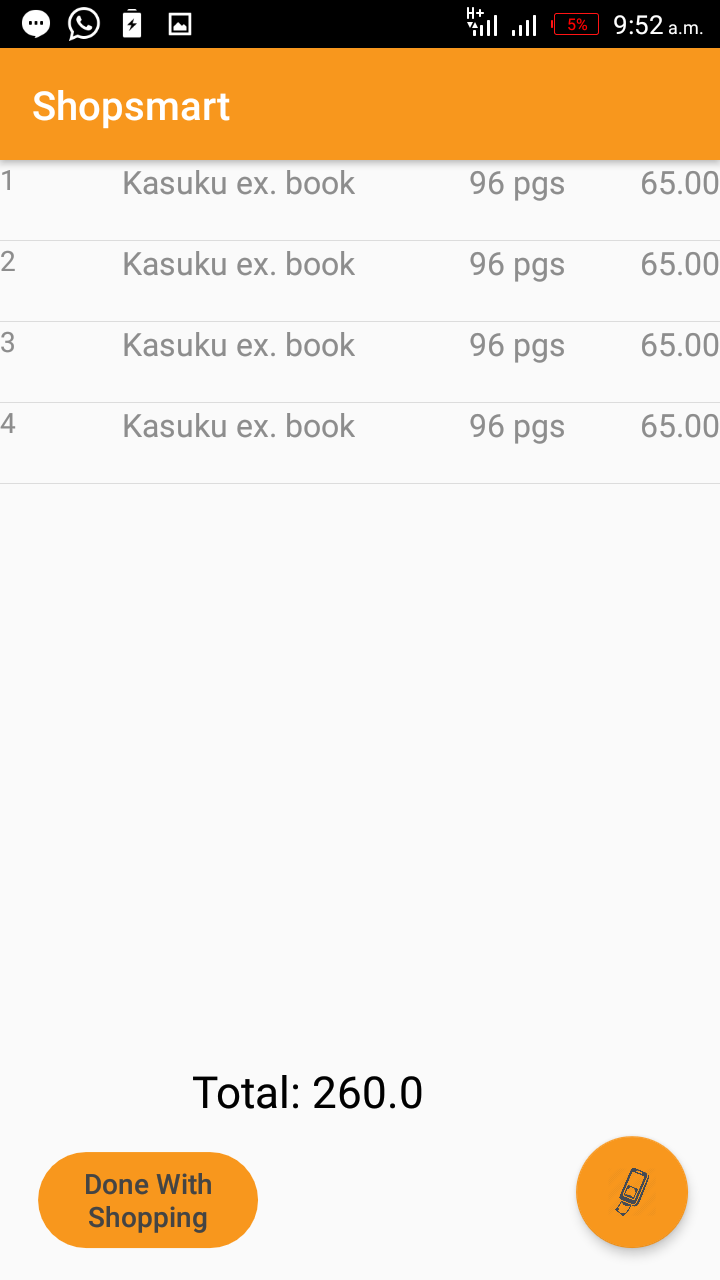
In this tab, the barcode scanner opens, and the user can be able to scan the barcode of items.



###### Figure 4.11 Barcode Scanner tab

* 1. **Scanned Lists tab**

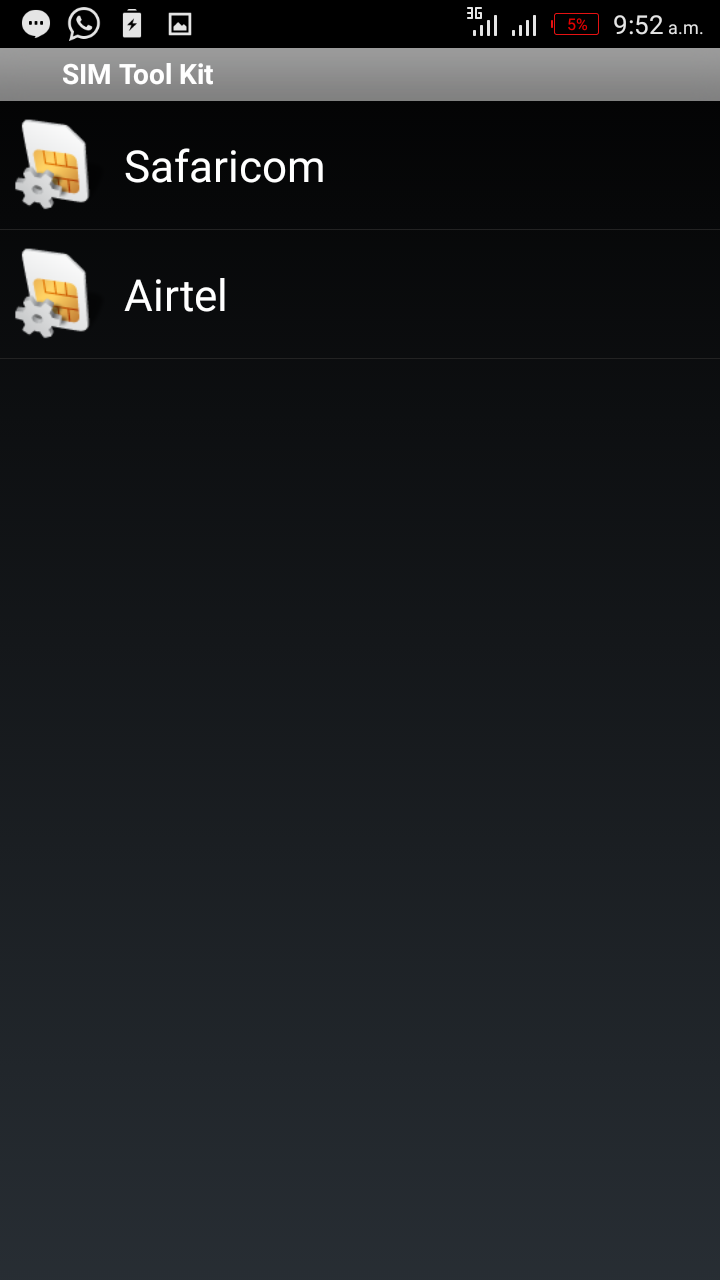
This tab displays the list of items that have been scanned by the user in a list view format. This enables the user to manage the scanned items by either adding more, or deleting an entry. The total amount of the items scanned is displayed for the user to always be in the know of their expenditures.



###### Figure 4.12 Scanned List tab

* 1. **Mobile payment tab**

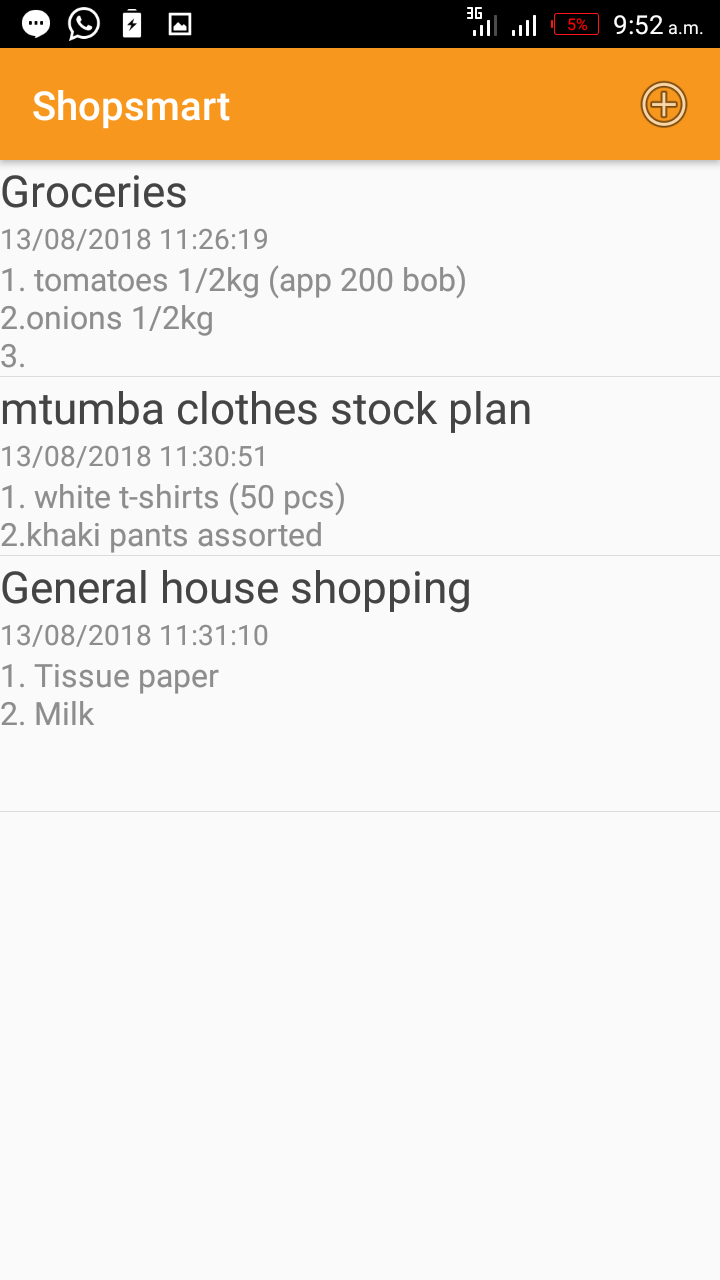
This tab is opened from the scanned lists tab, and it opens the sim tool kit for the user to pay for the total amount of items scanned using mobile money.



###### Figure 4.13 Mobile payment tab

* 1. **Shopping List Tab**

This module enables the user to create a new shopping list, edit an existing note and even delete an unwanted list. It helps a user to pre-plan a shopping activity by offering a platform to write a list before the actual shopping.



###### Figure 4.14 Shopping List tab

* 1. **Calculator Tab**

This module enables the user to have an easy access to a calculator when urgently needed and the user does not want to leave the app session. This will enable the user perform arithmetic calculations related to their shopping and hence be able to manage their money well. It acts as a backup in case the automatic barcode scanner does not work.



###### Figure 4.15 Calculator Tab

## 4.5 Testing

The testing was conducted in an iterative approach so the program design and implementation was continuously tested and evaluated thorough out the whole project. Input data was compared with the output given to check whether they were the required output. The following is a description of testing objectives, strategies that were employed during the testing of the system.

Smart Shopping mobile application testing objectives are;

1. To find errors or faults in software.
2. Test the system against user’s requirements.
3. To analyze the test results.

### 4.5.1 Testing Strategies

A test strategy is general approach to the testing process. This involved testing the system using different types of system tests. The process aimed at uncovering errors and measuring the system capability. The following system tests were performed on various areas during system development:

#### 4.5.1.1 Unit Testing

This was the first test in the development process and ensured that each unique part of the project performed accurately to the documented specifications and contained clearly defined inputs and expected results. The individual units in the system were tested separately so as to verify that they conform to the users’ requirements and operate according to the systems specification. Examples include: the enter amount tab denied user access to the barcode activity unless a valid amount was entered, when a barcode that does not exist in the database was scanned, an error message was displayed to the user. Unit testing treated each component as a stand-alone entity, which does not need other components during the testing process. Each module was tested to ascertain that it fully performed as expected.

**4.5.1.2 Integration Testing**

This involved testing integrated modules. Integration testing was done to ensure that modules could easily interact with each other and that the integration of the modules making up the system was working. The process involved a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing.

**4.5.1.3 Acceptance Testing**

This involves testing the system with the intent of confirming readiness of the product and customer acceptance.

#### 4.5.1.4 Alpha Testing

Alpha testing is done after coding. Alpha testing concentrated on verifications which included; ensuring valid entry and output for each record, system usability, amount entry validation, database and the interface consistency.

## 4.6 Chapter Summary

This chapter analyzed, interpreted and presented findings of the research. It discussed the functional and non-functional requirements of the application system. These requirements are then analyzed and transformed into design specifications and presented using UML diagrams. The chapter also discussed and outlined the software tools used in development of the system.

# CHAPTER FIVE

# SUMMARY, CONCLUSION AND RECOMMENDATIONS

## 5.1 Chapter Overview

This chapter provides a summary of the entire research findings, enables the researcher to make relevant conclusions and recommendations based on various objectives of the study.

The aim of the study was to design and develop a smart shopping application system for the android mobile smartphone. The application is intended for the use by anyone who occasionally goes to a supermarket to shop for items.

## 5.2 Summary of Findings

The first objective of the study was to adopt a suitable methodology and use it to develop a smart shopping mobile application system. In chapter 3, section 3.2 , the waterfall methodology is selected as the most convenient and appropriate methodology to design and develop the application system. This in summary is because the waterfall model identifies requirements long before programming begins.

The second objective of the study was to perform an analysis on requirements gathered for the application system. This has been achieved through the use of data collection tools to first collect the user requirements. The tools used in the study include Questionnaires, Interview questions and direct observations as specified in chapter 3 section 3.3. Analysis of the requirements is thoroughly done, especially the user requirements because the application is100% user oriented. See chapter3 section 3.10.

The third objective of the study was to provide a way of making shopping a very tireless, easy and enjoyable activity. This was to be achieved through ultimately designing and developing the application system. In the whole of chapter 4, the researcher and developer uses the requirements gathered in chapter 3, and analyzes it to come up with a design for the system. This designed is used to model and develop the actual application system using software tools. (See chapter 4 section 4.4.)

## 5.3 Conclusion

This research study looked at how people do their normal household shopping in supermarkets. It looked at the various challenges faced, the resources, tools and application systems directly and indirectly related to shopping activities.

It was established that over reliance on brain memory to recall all the items on a shopping list in most cases led to items being forgotten during the actual shopping. This meant that the shopper had to go back to the supermarket if this item is important and urgent.

It was also established in this study that doing mental calculations of prices of total items purchased and subtracting this from the cash at hand resulted in most individual cases, either overestimation or underestimation of their purchasing powers as most people could not accurately keep up with the math for long while all sorts of distractions are present in the supermarket.

Developing a smart shopping mobile application assistant will make people save on time, be accurate about their shopping, make shopping fun and interactive and relieve a person’s mind of the math that comes along with shopping.

## 5.4 Recommendation

**Networked application**

The application should in future be upgraded to fetch details of items from all the available supermarkets and compare them, and therefore give the user a price friendly option.

**Google Maps**

Google maps should be integrated in the system to give the user a view of the available supermarkets in the vicinity.

**Other Mobile OS platforms**

The application can be developed for other mobile-based platforms such as windows, blackberry and ios. This will enable a wider reach to more users and shoppers who are not on the android-based platform.

## 5.5 Suggestion for further research

1. **Integration with IOT technology**

The shopping assistant should be integrated with a smart home in the Internet of things technology, so as to always alert the user on items that have either replenished or expired.

1. **Real Time Data and Analytics technology in the app**

In a further research, the data and information being input, output and stored by this app can be tapped and utilized to give insight into customers and shoppers. Retailers will be able to know items in demand, best time in the year to stock certain items, which products to invest in advertising, potential purchasing power of a certain geographical region and also be able to determine if a new commodity can be successfully introduced to a market. This will in turn improve the shopping experience of people as shopping will become more personalized.

# REFERENCES

* Alba, J., Lynch, J., Weitz, B., Janiszewski, C., Lutz, R., Sawyer, A., & Wood, S. (1997). Interactive home shopping: consumer, retailer, and manufacturer incentives to participate in electronic marketplaces. *The Journal of Marketing*, 38-53.
* Android, O. S. (2011). Android. *Retrieved February*, *24*, 2011.
* Arthur, C. (2012). The history of smartphones: Timeline. *The Guardian*, *24*.
* Auger, C., Leduc, E., Labbé, D., Guay, C., Fillion, B., Bottari, C., & Swaine, B. (2014). Mobile Applications for Participation at the Shopping Mall: Content Analysis and Usability for Persons with Physical Disabilities and Communication or Cognitive Limitations. *International Journal of Environmental Research and Public Health*, *11*(12), 12777–12794. <http://doi.org/10.3390/ijerph111212777>
* Benington, H.D. (1956): Production of large computer programs. In Proceedings, ONR Symposium on Advanced Programming Methods for Digital Computers, June 1956, pp 15-27.
* Birrell, N. D. and Ould, M.A. (1988): A practical handbook to software
* Boulos, M. N. K., Wheeler, S., Tavares, C., & Jones, R. (2011). How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX. *Biomedical engineering online*, *10*(1), 24.
* Davis, D. (2013). *A history of shopping*. Routledge.
* Fling, B. (2009). *Mobile design and development: Practical concepts and techniques for creating mobile sites and Web apps*. " O'Reilly Media, Inc.".
* Gefen, D., Karahanna, E., & Straub, D. W. (2003). Trust and TAM in online shopping: An integrated model. *MIS quarterly*, *27*(1), 51-90.
* <http://www.onlinemarketingdegree.net/resources/history-of-shopping-center/>
* <https://expertise.jetruby.com/brief-history-of-mobile-apps-286fbbf766a9>
* <https://manifesto.co.uk/history-mobile-application-development/>
* <https://retailminded.com/the-history-of-retail-shopping/>
* <https://steelkiwi.com/blog/grocery-chains-using-mobile-tech-modernize-shoppin/>
* <https://tech.co/mobile-app-history-evolution-2015-11>
* <https://www.droidviews.com/essay-the-history-of-android-os/>
* https://www.forbes.com/sites/quora/2017/12/19/why-many-online-shopping-sites-are-becoming-mobile-shopping-apps/
* <https://www.hongkiat.com/blog/mobile-shopping-apps/>
* <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>
* <https://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>
* Inukollu, V. N., Keshamoni, D. D., Kang, T., & Inukollu, M. (2014). Factors influencing quality of mobile apps: Role of mobile app development life cycle. *arXiv preprint arXiv:1410.4537*.
* [Jon Stobart](https://www.emeraldinsight.com/author/Stobart%2C+Jon), (2010) "A history of shopping: the missing link between retail and consumer revolutions", , Vol. 2 Issue: 3, pp.342-349,
* Lee, M. K., & Turban, E. (2001). A trust model for consumer internet shopping. *International Journal of electronic commerce*, *6*(1), 75-91.
* Lessard, J., & Kessler, G. (2010). Android Forensics: Simplifying Cell Phone Examinations.
* Maia, C., Nogueira, L. M., & Pinho, L. M. (2010, July). Evaluating android os for embedded real-time systems. In *6th international workshop on operating systems platforms for embedded real-time applications* (pp. 63-70).
* Micheletti, M. (2003). Shopping with and for Virtues. In *Political virtue and shopping* (pp. 149-168). Palgrave Macmillan, New York.
* Nguyen, Tuan C. "The History of Smartphones." ThoughtCo, Jun. 14, 2018, thoughtco.com/history-of-smartphones-4096585.
* Paek, J., Kim, J., & Govindan, R. (2010, June). Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (pp. 299-314). ACM.
* Raymond, Eric (2001): Cathedral and the Bazaar, 1st Edition; O'Reilly
* Royce, Winston W. (1970): Managing the development of large software development; Cambridge University Press. ISBN 978-0521347921. pp 3-12.
* Shields, R. (Ed.). (2003). *Lifestyle shopping: The subject of consumption*. Routledge.
* Stobart, J. (2008). *Spend, spend, spend! A history of shopping*. History Press.
* TOGAF 9:The Open Group; <http://www.opengroup.org/togaf/>.
* Tracy, K. W. (2012). Mobile application development experiences on Apple’s iOS and Android OS. *Ieee Potentials*, *31*(4), 30-34.

# APPENDICES

## Appendix 1: Letter to respondent/ site manager

I am a student from Moi University, main campus undertaking research on a smart shopping mobile application system. The aim of this research is to identify whether people actually practically need a shopping assistant in the form of an phone application.

Kindly assist in filling the questionnaire. Information provided will be used for purposes of the research only and will be treated with utmost confidentiality.

## Appendix 2: Questionnaire

**Section A: General Information about you**

1. Gender

Male

Female

1. Age

15 – 25 years 26-35 years 36-45 years 46 years and above

1. Shopping experience in years

0-2 years 3-5 years 6-10 years 10 years and above

**Section B: Open Ended Questions**

1. What are some of the general challenges you face while shopping?................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................
2. Do you find it easy mentally performing calculations as you shop? Explain………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....
3. Do you write your shopping list on an actual piece of paper?.............................................................................................................................................................................................................................................................................................
4. What are some of the advantages associated with the new system?...................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................
5. What are some of the challenges that might be experienced in the event of adopting this shopping application?.................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................
6. Kindly suggest ways of improving this shopping application system to suit your needs……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....
7. General comments about the system?......................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

## Appendix 3: System requirements

**Hardware requirement**

The selection of a suitable Hardware and software is very important. This is to enable the system to run smoothly and work efficiently.

**Hardware**

A smartphone or tablet.

**Software Requirements**

Android OS 6.0 and above.

## Appendix 4: Project Codes

**Sample code for Database Tables and query functions**

public class DatabaseHelper extends SQLiteOpenHelper {

private static final int *DATABASE\_VERSION* = 26;

private static final String *DATABASE\_NAME* = "contactsManager";

//private SQLiteDatabase db;

String SQL\_CREATE\_SCANNED\_ITEM\_TABLE = "CREATE TABLE " + ColumnsContract.ScannedItemEntry.*TABLE\_SCANNED\_ITEMS* + " ("

+ ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, "+ ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_NAME* + " TEXT, "+ ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_QTY* + " TEXT, " + ColumnsContract.ScannedItemEntry.*COLUMN\_SCANNED\_ITEMS\_PRICE* + " TEXT NOT NULL " + ")";

String SQL\_DROP\_SCANNED\_ITEM\_TABLE = "DROP TABLE IF EXISTS "+ColumnsContract.ScannedItemEntry.*TABLE\_SCANNED\_ITEMS*;

String SQL\_CREATE\_BARCODES\_TABLE = "CREATE TABLE " + ColumnsContract.BarcodesReference.*TABLE\_BARCODE* + " ("

+ ColumnsContract.BarcodesReference.*COLUMN\_BARCODE* + " TEXT, " + ColumnsContract.BarcodesReference.*COLUMN\_ITEM\_NAME* + " TEXT, "

+ ColumnsContract.BarcodesReference.*COLUMN\_ITEM\_PRICE* + " TEXT, " +

ColumnsContract.BarcodesReference.*COLUMN\_QUANTITY* + " TEXT NOT NULL " + ")";

String SQL\_DROP\_BARCODES\_TABLE = "DROP TABLE IF EXISTS "+ColumnsContract.BarcodesReference.*TABLE\_BARCODE*;

public DatabaseHelper (Context context){

super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL(SQL\_CREATE\_SCANNED\_ITEM\_TABLE);

db.execSQL(SQL\_CREATE\_BARCODES\_TABLE);

Log.*e*("onCreate: ", "Payment CREATED");

Log.*e*("onCreate: ", "Barcodes CREATED");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL(SQL\_DROP\_SCANNED\_ITEM\_TABLE);

db.execSQL(SQL\_DROP\_BARCODES\_TABLE);

onCreate(db);

}

public long insertScannedItem(BarcodeQueryResponse response){

SQLiteDatabase db = this.getWritableDatabase();

ContentValues values = new ContentValues();

values.put(ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_QTY*, response.getItemQuantity());

values.put(ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_NAME*, response.getItemName());

values.put(ColumnsContract.ScannedItemEntry.*COLUMN\_SCANNED\_ITEMS\_PRICE*, response.getItemPrice());

long result = db.insert(ColumnsContract.ScannedItemEntry.*TABLE\_SCANNED\_ITEMS*,null,values);

// Log.e("saved\_item\_name", name);

Log.*e*("savedPayments", ""+result);

db.close();

return result;

}

public void insertBarcodesItem(JSONArray barcodeArray) throws JSONException{

SQLiteDatabase db = this.getWritableDatabase();

for (int i = 0; i<barcodeArray.length();i++) {

ContentValues values = new ContentValues();

values.put(ColumnsContract.BarcodesReference.*COLUMN\_BARCODE*, barcodeArray.getJSONObject(i).getString("barcode"));

values.put(ColumnsContract.BarcodesReference.*COLUMN\_ITEM\_NAME*, barcodeArray.getJSONObject(i).getString("itemName"));

values.put(ColumnsContract.BarcodesReference.*COLUMN\_ITEM\_PRICE*, barcodeArray.getJSONObject(i).getString("itemPrice"));

values.put(ColumnsContract.BarcodesReference.*COLUMN\_QUANTITY*, barcodeArray.getJSONObject(i).getString("itemQuantity"));

long result = db.insert(ColumnsContract.BarcodesReference.*TABLE\_BARCODE*, null, values);

Log.*e*("Saved\_\_Items", "" + result);

}

db.close();

}

public ArrayList<BarcodeQueryResponse> getAllAmounts(){

SQLiteDatabase db = this.getWritableDatabase();

ArrayList<BarcodeQueryResponse> data = new ArrayList<>();

String query = "SELECT \* FROM "+ ColumnsContract.ScannedItemEntry.*TABLE\_SCANNED\_ITEMS*;

Cursor cursor = db.rawQuery(query,null);

while (cursor.moveToNext()){

BarcodeQueryResponse barcodeQueryResponse = new BarcodeQueryResponse();

barcodeQueryResponse.setBarcodeName(cursor.getString(cursor.getColumnIndex(ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_ID*)));

barcodeQueryResponse.setItemName(cursor.getString(cursor.getColumnIndex(ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_NAME*)));

barcodeQueryResponse.setItemPrice(cursor.getString(cursor.getColumnIndex(ColumnsContract.ScannedItemEntry.*COLUMN\_SCANNED\_ITEMS\_PRICE*)));

barcodeQueryResponse.setItemQuantity(cursor.getString(cursor.getColumnIndex(ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_QTY*)));

data.add(barcodeQueryResponse);

}

db.close();

return data;

}

public BarcodeQueryResponse getBarcodeDetail(String barcodes){

SQLiteDatabase db = this.getWritableDatabase();

BarcodeQueryResponse result = new BarcodeQueryResponse();

// String query = "SELECT \* FROM "+ ColumnsContract.BarcodesReference.TABLE\_BARCODE +" WHERE "+ColumnsContract.BarcodesReference.COLUMN\_BARCODE + "='"+"1111111"+"'";

String query = "SELECT \* FROM "+ ColumnsContract.BarcodesReference.*TABLE\_BARCODE* +" WHERE "+ColumnsContract.BarcodesReference.*COLUMN\_BARCODE* + " = '"+barcodes+"'";

Cursor cursor = db.rawQuery(query,null);

while (cursor.moveToNext()){

result.setBarcodeName(cursor.getString(cursor.getColumnIndex(ColumnsContract.BarcodesReference.*COLUMN\_BARCODE*)));

result.setItemName(cursor.getString(cursor.getColumnIndex(ColumnsContract.BarcodesReference.*COLUMN\_ITEM\_NAME*)));

result.setItemPrice(cursor.getString(cursor.getColumnIndex(ColumnsContract.BarcodesReference.*COLUMN\_ITEM\_PRICE*)));

result.setItemQuantity(cursor.getString(cursor.getColumnIndex(ColumnsContract.BarcodesReference.*COLUMN\_QUANTITY*)));

}

db.close();

return result;

}

public void deleteAScannedEntry(int id){

SQLiteDatabase db = this.getWritableDatabase();

String query = "DELETE FROM " + ColumnsContract.ScannedItemEntry.*TABLE\_SCANNED\_ITEMS* + " WHERE " +

ColumnsContract.ScannedItemEntry.*COLUMN\_ITEM\_ID* + " = '" + id + "'";

db.execSQL(query);

}

public void deleteAll()

{

SQLiteDatabase db = this.getWritableDatabase();

db.delete(ColumnsContract.ScannedItemEntry.*TABLE\_SCANNED\_ITEMS*,null,null);

db.execSQL("DELETE FROM "+ ColumnsContract.ScannedItemEntry.*TABLE\_SCANNED\_ITEMS*);

db.close();

}

}

**Sample code for Barcode scanner implementation**

public class BarcodeActivity extends AppCompatActivity implements ZXingScannerView.ResultHandler,Serializable {

private static final int *REQUEST\_CAMERA* = 1;

private ZXingScannerView scannerView;

private static int *camId* = Camera.CameraInfo.*CAMERA\_FACING\_BACK*;

DatabaseHelper myDb;

public static Methods *myMethods*;

// static int totalAmount;

public static Double *totalAmount*;

static Double *x*;

//static int x;

static Double *totalBill* = 0.0;

public static AlertDialog.Builder *builder*;

TextView textView;

static BarcodeQueryResponse *itemDetais*;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

*builder* = new AlertDialog.Builder(this);

*itemDetais* = new BarcodeQueryResponse();

myDb = new DatabaseHelper(this);

*myMethods* = new Methods(this);

*builder*.setTitle("Scan Result" + "\nAmount Remaining : " + Double.*parseDouble*(*myMethods*.getAmount()));

textView = (TextView) findViewById(R.id.*tvTotalDisplay*);

scannerView = new ZXingScannerView(this);

setContentView(scannerView);

int currentApiVersion = Build.VERSION.*SDK\_INT*;

if(currentApiVersion >= Build.VERSION\_CODES.*M*)

{

if(checkPermission())

{

Toast.*makeText*(getApplicationContext(), "Scan an Item's Barcode!", Toast.*LENGTH\_LONG*).show();

}

else

{

requestPermission();

}

}

}

private boolean checkPermission()

{

return (ContextCompat.*checkSelfPermission*(getApplicationContext(), *CAMERA*) == PackageManager.*PERMISSION\_GRANTED*);

}

private void requestPermission()

{

ActivityCompat.*requestPermissions*(this, new String[]{*CAMERA*}, *REQUEST\_CAMERA*);

}

@Override

public void onResume() {

super.onResume();

int currentapiVersion = android.os.Build.VERSION.*SDK\_INT*;

if (currentapiVersion >= android.os.Build.VERSION\_CODES.*M*) {

if (checkPermission()) {

if(scannerView == null) {

scannerView = new ZXingScannerView(this);

setContentView(scannerView);

}

scannerView.setResultHandler(this);

scannerView.startCamera();

} else {

requestPermission();

}

}

}

@Override

public void onDestroy() {

super.onDestroy();

scannerView.stopCamera();

}

public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {

switch (requestCode) {

case *REQUEST\_CAMERA*:

if (grantResults.length > 0) {

boolean cameraAccepted = grantResults[0] == PackageManager.*PERMISSION\_GRANTED*;

if (cameraAccepted){

Toast.*makeText*(getApplicationContext(), "Permission Granted, Now you can access camera", Toast.*LENGTH\_LONG*).show();

}else {

Toast.*makeText*(getApplicationContext(), "Permission Denied, You cannot access and camera", Toast.*LENGTH\_LONG*).show();

if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M*) {

if (shouldShowRequestPermissionRationale(*CAMERA*)) {

showMessageOKCancel("You need to allow access to both the permissions",

new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M*) {

requestPermissions(new String[]{*CAMERA*},

*REQUEST\_CAMERA*);

}

}

});

return;

}

}

}

}

break;

}

}

private void showMessageOKCancel(String message, DialogInterface.OnClickListener okListener) {

new android.support.v7.app.AlertDialog.Builder(BarcodeActivity.this)

.setMessage(message)

.setPositiveButton("OK", okListener)

.setNegativeButton("Cancel", null)

.create()

.show();

}

@Override

public void handleResult(final Result result) {

final String myResult = result.getText();

Log.*d*("QRCodeScanner", result.getText());

Log.*d*("QRCodeScanner", result.getBarcodeFormat().toString());