



# MY RECEIPTS

## Cpit 499 Final Report

By

Nedaa Omar	1607450
Hania AL-Hafeez	1607412
Shahad Alattas	1605472

Supervised By

Dr.Abeer Al-Makky

Department of Information Technology  
Faculty of Computing and Information Technology  
King Abdulaziz University  
Jeddah – Saudi Arabia  
Spring 2020

### **DECLARATION by AUTHORS**

"I/we certify that this work has not been accepted in substance for any degree and is not concurrently being submitted for any degree other than that of BS Information Technology being studied at King Abdulaziz University, Jeddah. I/we also declare that this work is the result of my/our own findings and investigations except where otherwise identified by references and that I/we have not plagiarized another's work".

[Signature]

[Nedaa Omar]

[Signature]

[Hania Al-Hafeez]

[Signature]

[Shahad Alattas]

---

### **DECLARATION by SUPERVISOR**

I, the undersigned hereby certify that I have read this project report and finally approve it with recommendation that this report may be submitted by the authors above to the final year project evaluation committee for final evaluation and presentation, in partial fulfillment of the requirements for the degree of BS Information Technology at the Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah.

[Signature]

[Dr.Abeer Al-Makky]

## **Abstract**

Mobile applications are a major part of everyone's life as everything has developed and has been eligible for use over the internet or with your smartphone. One area that will require the usage of these developments and life improvements due to the existing of applications is shopping/selling retail.

Most shops use paper receipt in terms of any operations (refund/exchange) that is needed by their customers which is not very practical since it can get lost or even forgotten by the customer which stops them from refunding/exchanging, another thing is that these paper receipts is not environment friendly and the fact that sometimes it is issued to be thrown away the next day will eventually leave us with tons of non-recyclable paper receipts.

To solve such problems we have introduced My Receipt Application which will serve in solving these problems by using some techniques such Cloud Computing, QR codes, and data visualization that will be used to provide a better experience to the user as well as saving the environment from a possible pollution due to the usage of paper receipts. To make My Receipt possible we have conducted interviews and surveys, collected information in the literature review, which left us with the needed information to specify the requirements and techniques that we intend to work with.

My Receipts will work in a manner where all the customers receipts will be transferred automatically to the cloud computing as QR codes and then into the My Receipt Application, each receipt will be organized in a way user can retrieve the maximum intended information regarding their purchases . these receipts can be further used for exchange and refund processes. Also, it will allow the user to set a monthly limit to gain more control over their spending as well as displaying monthly purchases by a graph to track the spending.

## Table of Contents

Abstract .....	III
List of Figures .....	IX
List of Tables .....	XII
Chapter 1: General Introduction .....	1
1.1 Acronyms.....	1
1.2 Project Background.....	1
1.3 Problem Statement.....	1
1.4 Feasibility Study .....	2
1.4.1 Success Criteria.....	2
1.4.2 Completion Criteria.....	2
1.4.3 Weakness .....	2
1.4.4 Deliverables .....	2
1.5 Project Objectives .....	2
1.6 Proposed Solution .....	3
1.7 Stakeholders.....	3
1.8 Project Scope .....	3
1.8.1 Main Features.....	4
1.8.2 Methodology .....	4
1.8.3 project plan.....	5
1.8.4 Designed tools.....	7
1.9 Conclusion .....	7
Chapter 2: Literature Review .....	8
2.1 Introduction.....	8
2.2 Technology Background.....	8
2.2.1 Cloud Computing .....	9
2.2.1.2 Cloud Computing Deployment Model.....	9
2.2.3 Automatic Identification and Data Capture Technique .....	10
2.2.4 QR Code.....	10
2.2.4.1 Why QR Code? .....	10

2.2.4.1 Uses of QR code .....	11
2.2.4.2 How QR code works? .....	12
2.2.5 Data Visualization.....	12
2.3 Similar IT Solutions.....	13
2.3.1 Genius Scan .....	13
2.3.2 Receipt Hog .....	14
2.3.3 Invoice Maker by NorthOne .....	15
2.3.4 Invoice2go.....	16
2.3.5 Invoice Creator.....	17
2.4 Comparing Existing Solutions .....	18
2.5 Conclusion .....	18
<b>Chapter 3: Elicitation .....</b>	<b>19</b>
3.1 Introduction.....	19
3.2 Elicitation Techniques .....	19
3.3 Participations .....	19
3.4 Elicitation Result and Analysis of Collected Data.....	19
3.4.1 Interview .....	19
3.4.2 Questionnaire analysis .....	20
3.5 Requirement Specification.....	22
3.5.1 Functional Requirements .....	23
3.5.2 Non-functional Requirements .....	25
3.5.2.1 Usability .....	25
3.5.2.2 Availability .....	25
3.5.2.3 Scalability .....	25
3.5.2.4 Maintainability.....	25
3.5.2.5 Security .....	25
3.6 Conclusion .....	25
<b>Chapter 4: Analysis Model .....</b>	<b>26</b>
4.1 Introduction.....	26
4.2 Use case Diagram .....	26
4.3 Elaborated Use case .....	26

4.3.1 Elaborated Use case “Login” .....	27
4.3.2 Elaborated Use case “Display All Receipts of User” .....	27
4.3.3 Elaborated Use case “Register User” .....	27
4.3.4 Elaborated Use case “Generate Identity QR” .....	28
4.3.5 Elaborated Use case “Set Monthly Purchase Limit” .....	28
4.3.6 Elaborated Use case “Display Details of Specific Receipt” .....	28
4.3.7 Elaborated Use case “Choose to Display Monthly Purchase” .....	29
4.4 Traceability Matrix: .....	29
4.5 Sequence Diagram .....	31
4.6 Data Model .....	34
4.7 Design Choices .....	34
4.8 Class Diagram.....	35
4.9 Conclusion .....	36
<b>Chapter 5: Design Model .....</b>	<b>37</b>
5.1 Introduction.....	37
5.2 Low level Architecture .....	37
5.3 Graphical User Interface (GUI) Design.....	38
5.4 Conclusion .....	43
<b>Chapter 6: Implementation .....</b>	<b>44</b>
6.1 Introduction.....	44
6.2 Tools .....	44
6.2.1 Hardware.....	44
6.2.2 Software .....	44
6.2.2.1 Firebase .....	44
6.2.2.2 Command Line (CMD).....	44
6.2.2.3 Microsoft Visual Studio.....	45
6.3 System implementation.....	45
6.3.1 Register Page Implementation .....	45
6.3.2 Login Page Implementation .....	47
6.3.3 User Identity QR Generation .....	47
6.3.4 Storing Receipts to Specific User .....	48

6.3.5 Display List of User Receipts .....	50
6.3.6 View Receipt Details .....	50
6.3.7 Update Receipt QR .....	51
6.5 Code debugging and Troubleshooting issues .....	52
6.5.1 QR Generation and Update .....	52
6.5.2 Linking Application to firebase storage.....	52
6.5.3 Calculating the Total of Receipts.....	53
6.6 Conclusion .....	53
<b>Chapter 7: Testing.....</b>	<b>54</b>
7.1 Introduction.....	54
7.2 Unit Test .....	54
7.2.1 Test Running app and its component.....	54
7.2.2 initialize app code .....	55
7.2.3 Test generating QRcode from URL.....	56
7.3 Integration Testing .....	57
7.3.1 Tested integration Functions.....	57
7.3.2 Integration Testing Result.....	57
7.4 Conclusion .....	57
<b>Chapter 8: Results and Discussion.....</b>	<b>58</b>
8.1 Introduction.....	58
8.2 Graphical User Interfaces .....	58
8.2.1 Start Screen .....	58
8.2.2 Reset-Password Screen .....	59
8.2.3 Sign up Screen .....	60
8.2.4 User Profile Screen .....	61
Figure 8.4 shows Profile screen which refer to user identity QR page that will be generated automatically after user create an account on MyReceipts application by displaying his/her name and user will have the ability to get back to profile page when he/she want, this page will contain: .....	61
8.2.5 Receipts List Screen.....	61
8.2.6 Receipt Details Screen .....	62

8.3 Objective Accomplished.....	62
8.4 Limitations of the Project .....	63
8.5 Conclusion .....	63
Chapter 9: Conclusion and Future Work .....	64
9.1 Challenges and limitations.....	64
9.2 Future work.....	65
10. References.....	66
Appendix A - Interview .....	68
Appendix B - Questionnaire .....	72
Appendix C - Questionnaire Result .....	73
Appendix D - Members Contribution .....	75

## List of Figures

Figure 1.1 Proposed Solution idea .....	3
Figure 1.2 agile methodology process [1].....	4
Figure 1.3 Gant chart Summary .....	5
Figure 1.4 Gant chart pre-planning.....	5
Figure 1.5 Gant chart Planning phase .....	5
Figure 1.6 Gant chart analysis phase .....	6
Figure 1.7 Gant chart design phase.....	6
Figure 1.8 Gant chart Implementation phase .....	6
Figure 1.9 Gant chart Testing phase .....	6
Figure 1.10 Gant chart project submission phase .....	7
Figure 2.1 Devices and Servers that Use Cloud Computing [4].....	9
Figure 2.2 encoding data into QR [8] .....	10
Figure 2.3 reconstructing multiple QR codes into a single QR code [8].....	11
Figure 2.4 position detection patterns in QR symbol [8].....	11
Figure 2.5 Type of Data visualization will use in Project [10] .....	12
Figure 2.6 Application that Arrange the Receipts [11].....	13
Figure 2.7 Application that Give Points in any Purchases [12].....	14
Figure 2.8 Application Create and Arrange the Receipts [13].....	15
Figure 2.9 Application Show Notification and Status of Receipts [14] .....	16
Figure 2.10 Application Create and Just Display Created Receipts [15] .....	17
Figure 3.1 Q1 survey result.....	20
Figure 3.2 Q2 survey result.....	20
Figure 3.3 Q3 survey result.....	21
Figure 3.4 Q4 survey result.....	21
Figure 3.5 Q5 survey result.....	22
Figure 3.6 Q6 survey result.....	22
Figure 4.1 My Receipts Application Use case of the customer .....	26
Figure 4.2 Sequence Diagram Register User.....	31
Figure 4.3 Sequence Diagram User Login.....	32

Figure 4.4 Sequence Diagram Receipts Details.....	32
Figure 4.5 Sequence Diagram Set Monthly Purchase Limit.....	33
Figure 4.6 Sequence Diagram choose to display monthly purchase .....	33
Figure 4.7 observer pattern for My Receipts .....	34
Figure 4.8 Class Diagram for My Receipts Application.....	35
Figure 5.1 My Receipts Low-Level Architecture .....	37
Figure 5.2 Signup interface.....	38
Figure 5.3 User Login interface .....	39
Figure 5.4 user identity QR interface.....	39
Figure 5.5 menu or navigation bar interface .....	40
Figure 5.6 Arranged Receipts interface .....	40
Figure 5.7 Detailed receipt interface.....	41
Figure 5.8 Detailed Expired Receipt interface.....	41
Figure 5.9 adjust monthly limit interface.....	42
Figure 5.10 monthly purchase visualization interface .....	42
Figure 6.1 Loading user variables code .....	45
Figure 6.2 User account creation condition code.....	46
Figure 6.3 user Login condition code .....	47
Figure 6.4 functions of profile page.....	48
Figure 6.5 specified tag to display QR.....	48
Figure 6.6 Adding receipts to user by code .....	48
Figure 6.7 Adding receipts to user by firebase storage.....	49
Figure 6.8 code of displaying list of user receipts .....	50
Figure 6.9 code of receipts details page.....	51
Figure 6.10 Receipts firebase storage .....	51
Figure 6.11 Receipts attributes stored in firebase .....	52
Figure 7.1 Unit Test Code for Creating and Initiate the App .....	54
Figure 7.2 App Code of Creating and Initiate 1.....	55
Figure 7.3 App Code of Creating and Initiate 2.....	55
Figure 7.4 Unit Test Result of Initiate and Creating the App.....	56
Figure 7.5 Unit Test Code of QR code Generation .....	56

Figure 7.6 Result of QR unit test .....	56
Figure 8.1 MyReceipts Login or Home Page .....	58
Figure 8.2 MyReceipts Reset Password Page.....	59
Figure 8.3 MyReceipts Sign up Page.....	60
Figure 8.4 MyReceipts User Profile Page.....	61
Figure 8.5 MyReceipts User Receipts List Page .....	61
Figure 8.6 MyReceipts Receipt Details Page .....	62

## **List of Tables**

Table 2.1 Summary of the features between My Receipts Application and related Application .....	18
Table 3.1 Functional Requirements of My Receipts .....	23
Table 3.2 Functional Requirements of My Receipts (cont.) .....	24
Table 4.1 traceability matrix of the project.....	29
Table 4.2 traceability matrix of the project (cont.) .....	30
Table 4.3 traceability matrix of the project (cont.) .....	31
Table 7.1 Unit Test Table .....	54
Table 7.2 Table of Integration Test.....	57

# **Chapter 1: General Introduction**

## **1.1 Acronyms**

QR	Quick Response Code
Identity QR	A QR that will be used to identify a specific user which will be unique code
IOS	iPhone operation system
GUI	Graphical user interface

## **1.2 Project Background**

Technology is advancing every second, Smartphones are the most important invention ever found, this leaded many aspect of life to rely on smartphones, it has changed the way everything works to an easier faster way.

Mobile applications are a major part of smartphones as they are the life of it because it provides us with most of the facilities through the smartphones, this gives the opportunity to use to enhance to customer experience in many fields one of them is the buying/selling field.

My Receipts application will help people by providing them an access to an electronic interface through their phones containing all the receipts using the cloud computing technology for transferring from the Seller System to the Costumer Phone automatically, giving them constant access to all of their previously purchases receipts.

## **1.3 Problem Statement**

Most of the costumers has faced a problem a day when it comes to refund or exchange terms, as well as the sellers which faces several issues like arguing with costumers about the importance of the receipt existence for any operation to be done.

We will display some reasons in the means of the importance of (my receipt) and how it can be very helpful:

- Receipts could get lost or damaged.
- Expiration of refund and exchange.
- Constant using of paper that will be thrown away after a short period of time which will end up only worsen the environmental problems.
- Thermal Paper uses heat rather than ink and contains BPA and the thermal paper can't be recycled.

## **1.4 Feasibility Study**

### **1.4.1 Success Criteria**

The shoppers can receive all their receipts on the (My Receipts) application that will be retrieved from the cloud to their interface in an organized manner which can be used further for any intended operation regarding the refund/exchange , as well as their ability to set a monthly limit for each month to keep their spending's within their monthly budget. Moreover, the shoppers can use the application to keep track of their purchases in each month by viewing the graphical representation including all months and how much they spent in it.

### **1.4.2 Completion Criteria**

Our application can retrieve receipts from the cloud of the specific user, display detail of each receipt, notify user of receipts status, display monthly spending's as graphical visualization and notify the user whenever he is about to exceed the monthly limit once its set.

### **1.4.3 Weakness**

-Cloud computing only works upon the existence of internet.  
-Storing receipts is crucial since is sensitive information a maximum security must be applied and only few attributes that can not disclose any important information must be dealt with and stored on the cloud.

### **1.4.4 Deliverables**

We intend to provide a complete research regarding this project and implement it in such it includes: mobile application connected to the cloud computing storage to store receipts as QR codes that will be retrieved to user interfaces for further use.

## **1.5 Project Objectives**

This project will develop an application using cloud computing for helping customers to have a better shopping experience.

### **The main objectives are:**

- Improving the refund/exchange process from traditional to electronic.
- Displaying each month purchases using a graph to give the user the ability to track their spending to manage their budget.
- Allow setting a monthly limit that will help the customer to not exceed the amount of money they should spend within a month.
- Allow the user to access the existing receipts for refund and exchange.

- Organize all the receipts of each customer in one place for easier accessibility and usability.
- Different kind of notification regarding the receipt status (exchange date is about to expire/ refund date is about to expire).
- Supports organizing the receipts into categories for a better accessibility.

## 1.6 Proposed Solution

We tend to implement and develop an application using cloud computing technology to facilitate refunding and exchanging manners. This allows customers have a better and easier shopping experience. Figure 1.1 gives an overview how the combination of different technologies, smartphone, and computer device will make refund/exchange process for an item fast and easy .

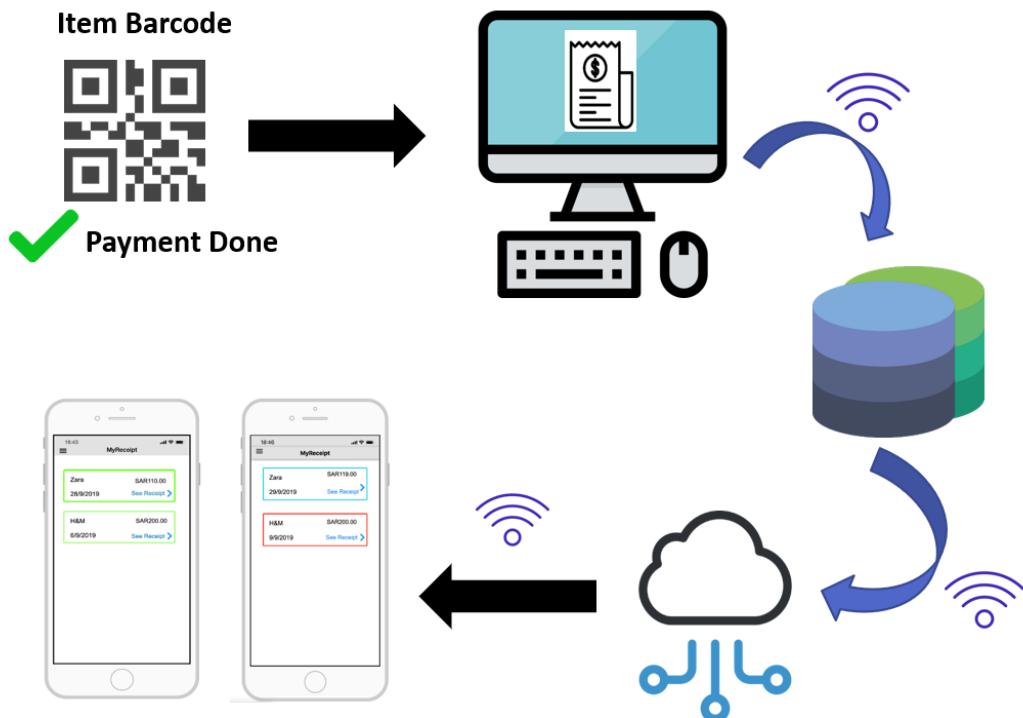


Figure 1.1 Proposed Solution idea

## 1.7 Stakeholders

The application will be used by shoppers age 12 and above who would prefer tracking their receipts for further usage.

## 1.8 Project Scope

Project scope is a huge part of planning phase of any project and it gives a detailed set features of this project that is collected and defined upon the project requirements.

“My Receipt” is intended to provide an interactive way of facilities to shoppers using the cloud computing and the QR (quick response) technologies. Our application main purpose is to enhance the selling/refunding experience to be more convenient and to follow this era of technology rather than the exhausting traditional way.

#### **What our project is aiming to deliver:**

- An IOS/android application that works in plus to the Seller System using the cloud that stores the needed data to be displayed to the user.

#### **1.8.1 Main Features**

##### **Our project tends to have the following as main features:**

- Using cloud computing to store all of user’s data
- Connecting to Seller System and transferring data to the cloud
- Organization feature: allowing user to organize receipts
- Setting budget limit for each month
- Notifications about the receipt’s expiry dates.

#### **1.8.2 Methodology**

This project may involve further future changes and additives to enhance each feature the application is providing, and it must be open to respond to the changes in an easy fast way. The systematic method that is going to be used is the Agile model. a framework to software development that enhance testing throughout the software development lifecycle and the continuous iteration of development with an eye toward being able to respond to change easily and quickly[1]. We are going to use the agile methodology to reflect any changes that may occur in any period of time in the next iteration which will leave us with more flexibility.



Figure 1.2 agile methodology process [1]

### 1.8.3 project plan

In this section, we will display the plan of the entire project. The project will go through two semesters which will include several phases such as planning, analysis, design, implementation and then monitoring. We will display the phases regarding this semester which are the planning, analysis and design.

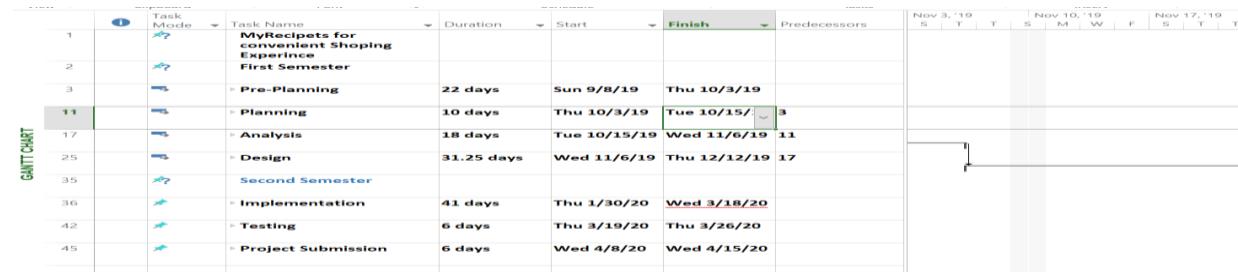


Figure 1.3 Gant chart Summary

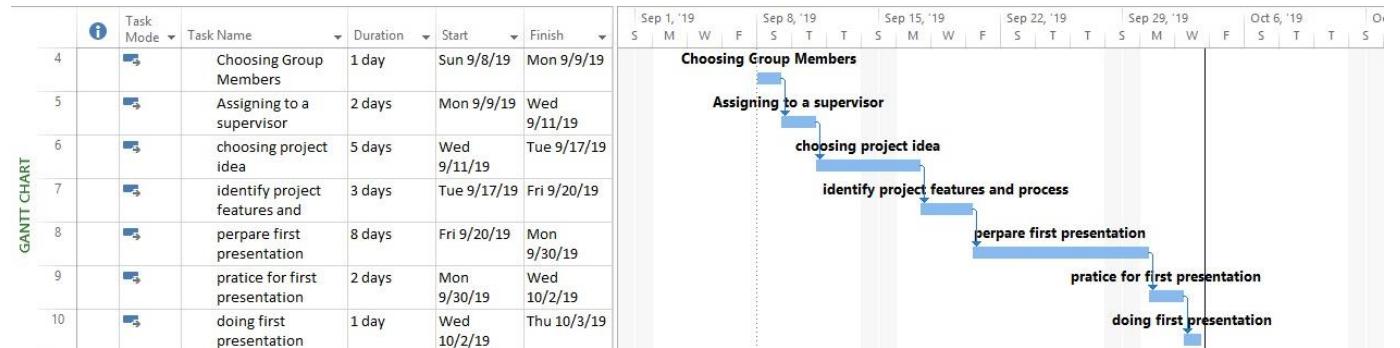


Figure 1.4 Gant chart pre-planning

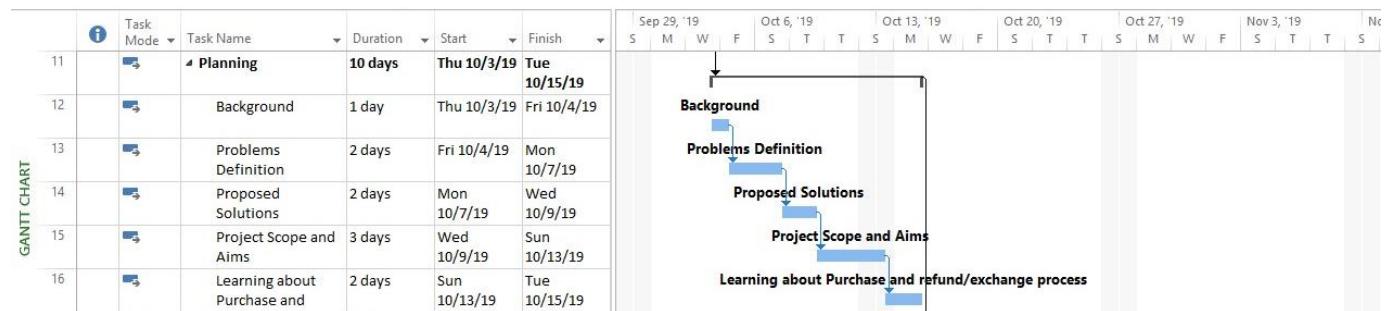


Figure 1.5 Gant chart Planning phase

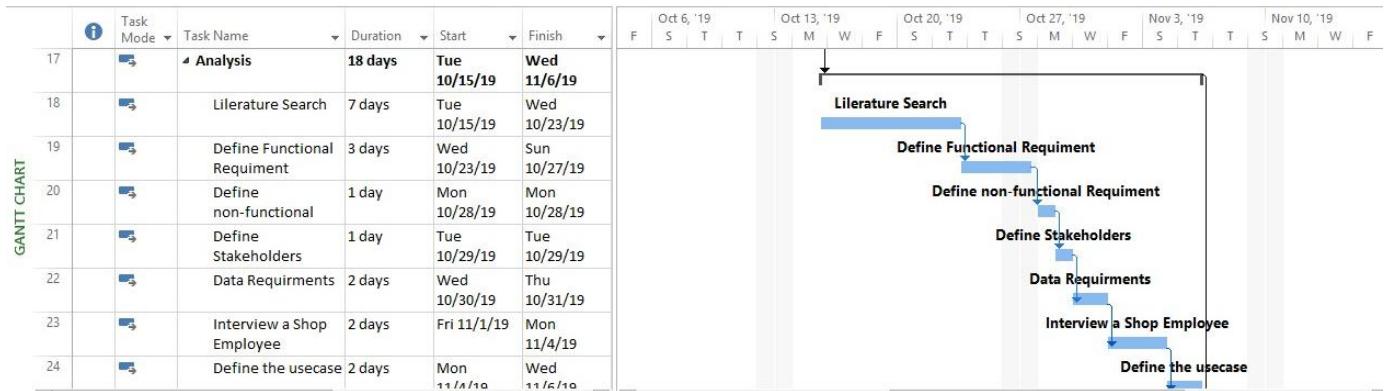


Figure 1.6 Gant chart analysis phase

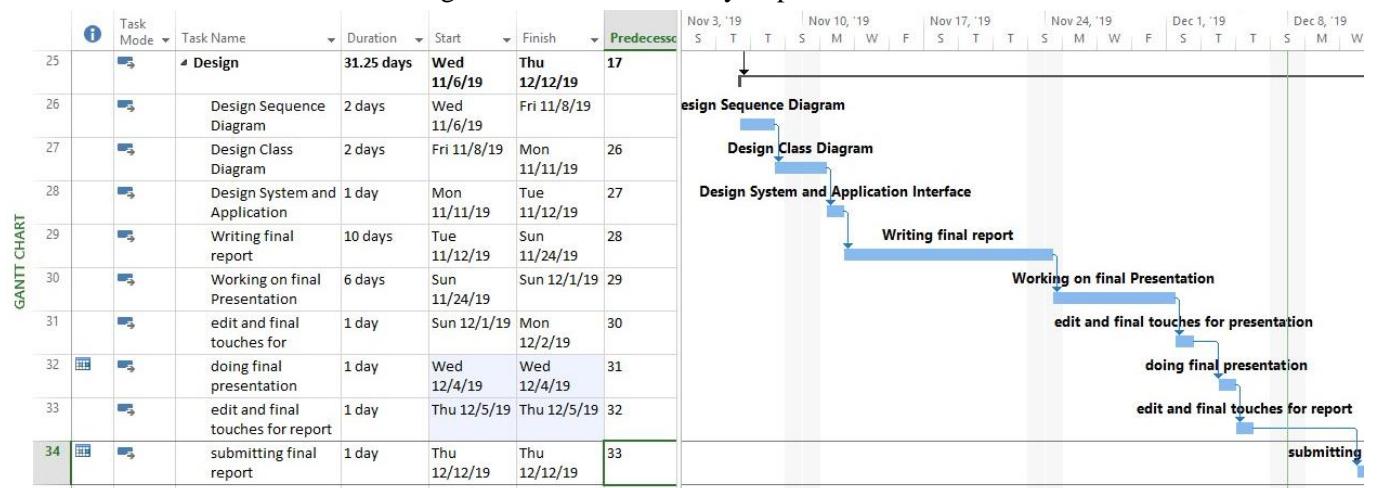


Figure 1.7 Gant chart design phase

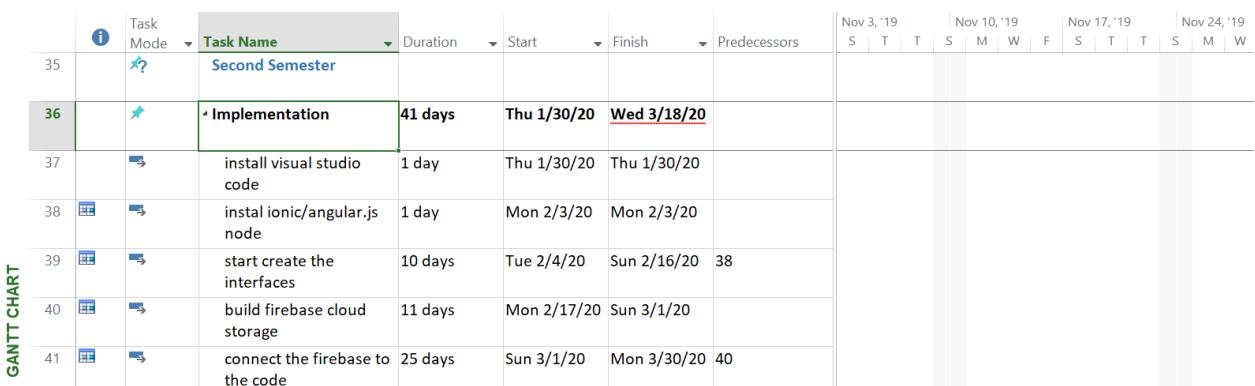


Figure 1.8 Gant chart Implementation phase



Figure 1.9 Gant chart Testing phase

**GANTT CHART**

45		Project Submission	6 days	Wed 4/8/20	Wed 4/15/20		
46	➡	Final report submission	1 day	Wed 4/8/20	Wed 4/8/20		
47	➡	Final Application submission	1 day	Fri 4/10/20	Sun 4/12/20		
48	➡	Final presentation	1 day	Tue 4/14/20	Tue 4/14/20		

Figure 1.10 Gant chart project submission phase

### 1.8.4 Designed tools

We are going to use different tools to draw the diagrams of the system and design the initially web application this semester.

The tools that going to be used are:

- Star UML
- Proto.io
- Lucid Chart

## 1.9 Conclusion

This chapter had discussed general description of My Receipts application. It described the application background, problem statement, Feasibility study, Proposed solution, stakeholders, and the project scope. We introduced the idea of our project, goals, its main objective, also we have mentioned the targeted user and displayed the project plan along with the methodology that is intended to be used and followed during this project.

In the next chapter, the literature review will be discussed.

## **Chapter 2: Literature Review**

### **2.1 Introduction**

This study has conducted a research on the implementation of cloud computing in shopping stores, concentrating on making an application having costumers' receipts information using QR code which is an automatic identification and data capture technique. The QR which is type of barcode that contains a matrix of dots for identifying each purchasing operation that will be stored on the cloud system including all the data that will be used by costumers for refund and exchange instead of using paper receipts.

We conducted that using paper is the worst thing to do. As to what Katherine Martinko feistyredhair (2018) said “Thermal paper uses heat rather than ink, relies on bisphenol. BPA is a hormone disruptor and is absorbed through the skin”.[2] which indicates that paper receipts paper type is thermal that works with ink rather than heat to write, thermal paper has this chemical bisphenol A (BPA) that coats the paper receipts that causes endocrine disruptor and even the recycling of these kind of paper will release extra bisphenol A into the environment which ends up causing further damage .

My Receipts is an application that will limit the usage of paper receipts and it will be used in several situations where it receives the receipts of costumers directly from the cloud system and organize them as cards in a user-friendly manner. Limitation propriety in the application can be used to help costumers keep up with their monthly budget plan. Also, the receipts will be displayed as a graph that visualize monthly activities of purchases that made by the user.

In this chapter we are going to provide basic knowledge on technology in section 2.2 that we are going to use in the application. Followed by section 2.3 which will identify the relationship or similarities between existing solutions with My Receipts application. Then will compare some existing solutions with My Receipts application in section 2.4.

### **2.2 Technology Background**

Technology is growing every day by day; most people are using these modern technologies, and everything has become easier and what once took weeks or days to preform we can now do it with one click with no effort or time wasted. The development of the technology has made people lead more comfortable lives.

Uber for example is a successful app that provide rides to its customers in the easiest and fastest way through a simple click. “In addition helping you get from point A to point B, we're working to bring the future closer with self-driving technology”[3].

Due to the huge growth of the technology there is various types of technologies that can be applied to create change and impact business and society cloud computing is one of the examples.

### 2.2.1 Cloud Computing

According to NIST definition "Cloud computing is a model access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction." [4] cloud computing relies on shared computer resources for example networks, servers that provides services and applications rather than having a local server to handle applications. Also, it describes the data centers that will be available for access to many users over the internet

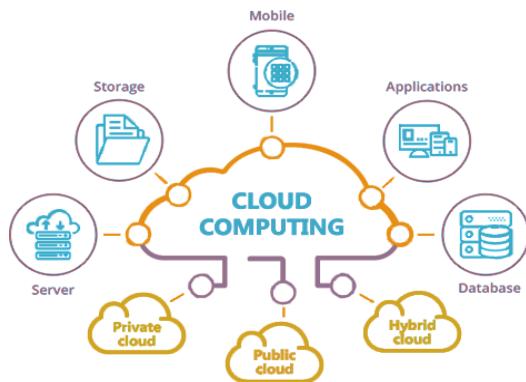


Figure 2.1 Devices and Servers that Use Cloud Computing [4]

#### 2.2.1.2 Cloud Computing Deployment Model

The deployment model based on the community "community cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns. [5]",

Community cloud will be the best model to serve the purpose of My Receipts application since it will be used by a specific community of shared concerns (Selling Stores, Costumers) and it can be owned by different companies or brands and managed by different employees for the refund and exchange process.

### **2.2.3 Automatic Identification and Data Capture Technique**

AIDC used to identify and track items, automatic identification and data collection, store information on discrete, packaged or containerized items[6]. AIDC which refers to automatic identification data capture is a method used to capture and collect data in whatever form which is transferred automatically to the system and it includes QR codes.

### **2.2.4 QR Code**

QR Code “is a matrix symbol which has been developed as the one enabling all of high capacity”[7].. Its stands for quick response which can be read quickly by smartphones or scanning machine for instant access to hidden included information in it. Also, QR is two-dimensional symbol that stores data in both horizontal and vertical manners.

#### **2.2.4.1 Why QR Code?**

- 1- High Capacity of encoding data:

QR code can work with different types of data and it can take up to 7,089 digits or 4296 characters to be encoded in one QR symbol[8] as shown in Figure 2.2. based on what was mentioned it is the most appropriate serving the huge amount of receipts for each costumer and each item included with its names and id numbers as well as storing information about the costumer.

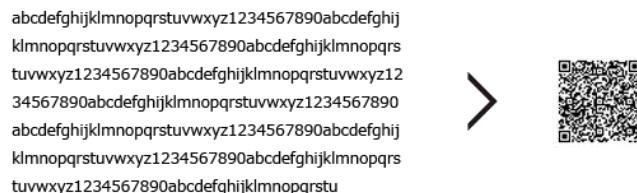


Figure 2.2 encoding data into QR [8]

- 2- Small printout size:

QR can represent and store data vertically and horizontally so it can encode double the amount of data a regular barcode can with minimal space required [8].

- 3- Structured appending feature Multiple QR codes as shown in Figure 2.3 can be reconstructed as a single QR symbol that contains multiple QR codes information, the user can have a single QR code symbol displayed on their application containing all the receipts for a specific date rather than having an individual barcode for each receipt due to this feature.

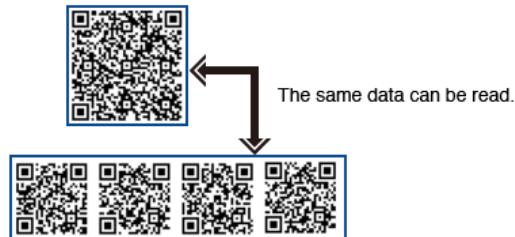


Figure 2.3 reconstructing multiple QR codes into a single QR code [8]

- 4- Readable from any direction in 360 degrees

QR code unlike the traditional barcode support high speed reading thanks to the position detection patterns placed on the three corners of the QR symbol which can be used easily for scanning by stores in case of an operation is required such as refund/exchange shown in the Figure 2.4.

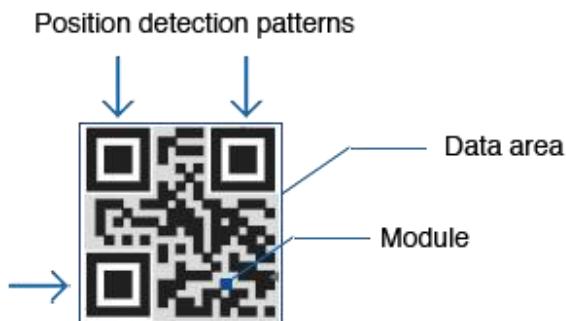


Figure 2.4 position detection patterns in QR symbol [8]

#### 2.2.4.1 Uses of QR code

QR is contributing to make many functions and operations more convenient and its used in a wide variety of business process such as factories distribution and manufacturing.

- **QR in retail sales:**

QR is used in coupon service rather than sending people paper coupons they work with coded coupons that is sent to costumer's smartphones and then scan them with a scanner, this will help raising customer satisfaction because they don't have to bring paper coupons with them that can be forgotten or lost.

QR code can be used in our application in the same manner as coupons is used in QR by encoding all the receipts and costumer details in a QR code symbol which is sent to the costumer

smartphone for further use of exchange/refund by only scanning the QR symbol displayed on the smartphone to complete the operation.

#### 2.2.4.2 How QR code works?

QR require two technologies in order to function:

1- QR Generator:

Which is either programming the QR for a specific function for example displaying the receipt whenever it is scanned or can be generated using existing website that encode some required details and information and then provide them for the desired user.

2- QR code scanner:

Scanner is used to decode the QR code to extract the stored information or programmed functionality which is done by a specific kind of scanners as the (AT30Q) scanner.

#### 2.2.5 Data Visualization

Data visualization is “the graphical representation of information and data by using visual elements like charts, graphs, and maps” [9] This visual graph serves variety of business purpose and it represents value interests.

My Receipts application will use data visualization as a feature to show monthly purchases for each customer which help them to keep track of their monthly spending’s as shown in Figure 2.5 . Also, if a user exceeded the limit of monthly budget then the user can minimize their purchases the next month. My Receipts application will deploy the column chart visual element graph that “use vertical columns to show numerical comparisons between categories.” [10], it would be the most appropriate way of displaying monthly purchases money amount.

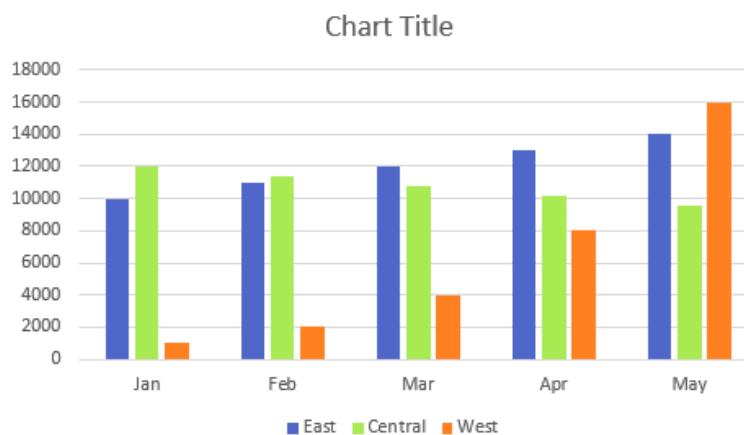


Figure 2.5 Type of Data visualization will use in Project [10]

## 2.3 Similar IT Solutions

There are many IT similar solutions that were designed for receipt creation, sending, or arranging them. These solutions will be presented in the following for five selected applications, their advantages and disadvantages will be discussed.

### 2.3.1 Genius Scan



Figure 2.6 Application that Arrange the Receipts [11]

Genius Scan an application used as smart algorithms to detect users' documents and apply perspective correction automatically, this paper documents can be digitized into a PDF file. Also users can add more pages ,merge and group documents together with tags to be stored on archive field [11]

#### Advantages:

- Genius Scan corrects any perspective distortion in users' photos.
- All the documents automatically appear in all users accounts on different devices.
- Scans are auto detected.

#### Disadvantages:

- Any paper documents or receipts when its received to the application must be scanned manually.
- Scanned paper receipt can't be used again for refund or exchange the item is just stored on the application as PDF file or image.

### 2.3.2 Receipt Hog

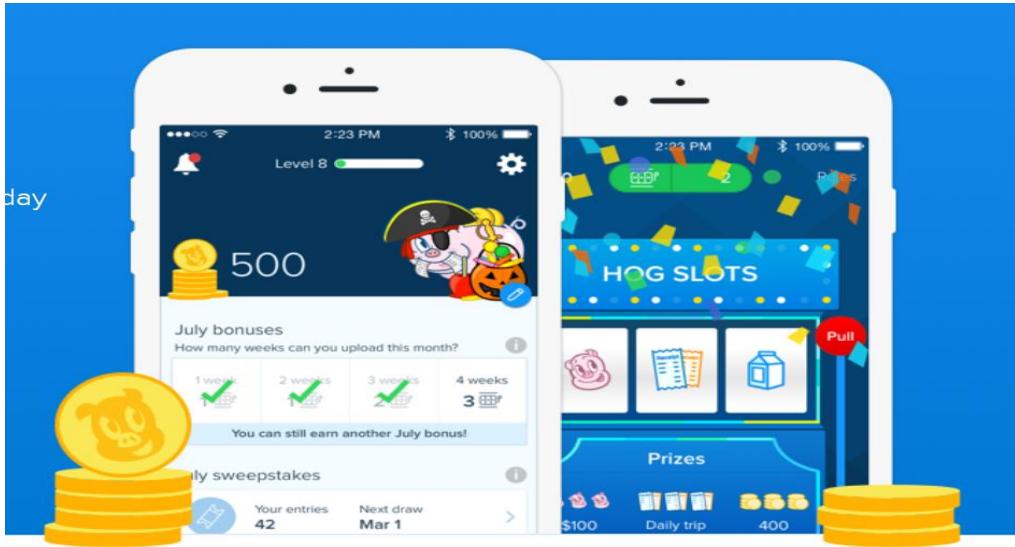


Figure 2.7 Application that Give Points in any Purchases [12]

Receipt Hog mobile app its purpose to turn users daily shopping receipts into easy cash reward so each receipt would be snap by user as a picture and submit it into the app to earn coins or snips.

#### Advantages:

- It requires a picture of a receipt of any shop regardless of the stuff the user bought.
- Users can win shopping trips.
- application works with Amazon so users can take their coins and use them in Amazon as gift card.

#### Disadvantages:

- Exhausting Process to snap the receipts every time just to get coins, not everyone interested or excited to do so.
- Not all users have accounts on amazon so not everyone will benefit from this amazon gift card feature.

### 2.3.3 Invoice Maker by NorthOne

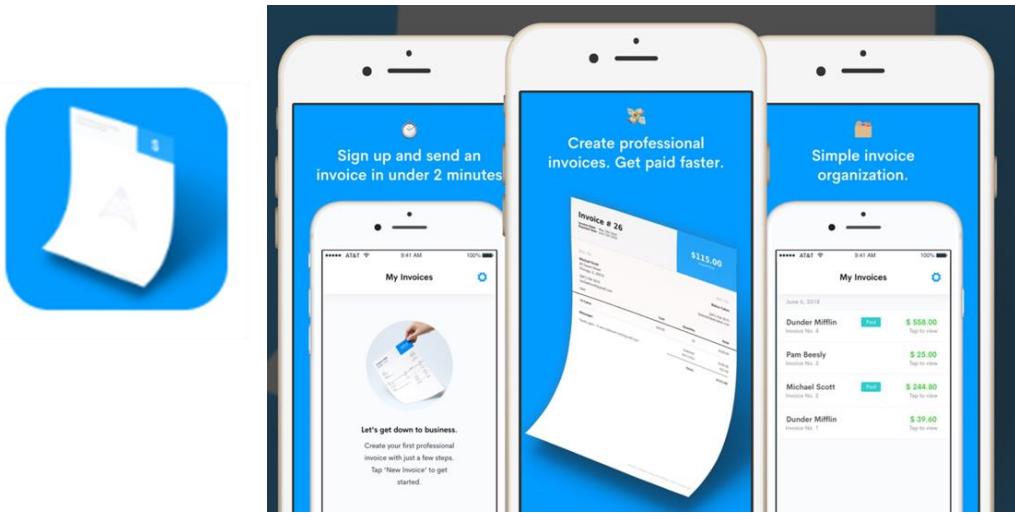


Figure 2.8 Application Create and Arrange the Receipts [13]

Free Invoice allows business owners to create and send a beautiful, professional invoice [13]. Free invoice is an application that is aimed to business owners to generate and send invoices in a small amount of time with no further subscription or fees required.

#### Advantages:

- Easy creation of invoices.
- Invoices are generated in the form of the PDF.
- It's reliable for various receipt situations like credit memos, and credit notes with the functionality of payment tracking.

#### Disadvantages:

- It's used by business owners only.
- Invoices need to be created manually each time.

## 2.3.4 Invoice2go

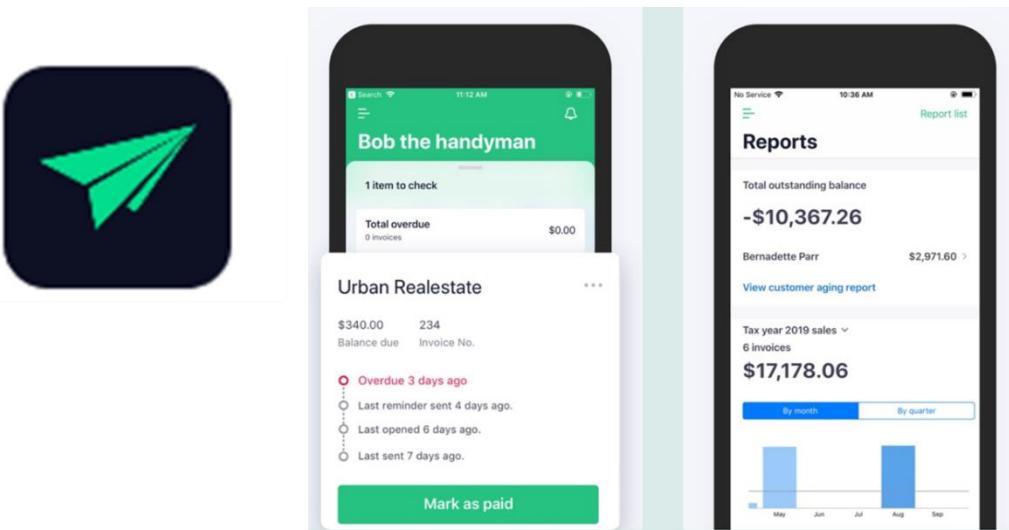


Figure 2.9 Application Show Notification and Status of Receipts [14]

An application used by business owners to send invoices to customers with an automatic synchronization to their smart devices.

### Advantages:

- Invoices are transferred in no time.
- Online payment feature.
- Available statuses for different invoices.

### Disadvantages:

- Doesn't support notification as status need to be checked manually.
- Invoices can't be used for other operations.
- Can't be used by regular customers.

## 2.3.5 Invoice Creator

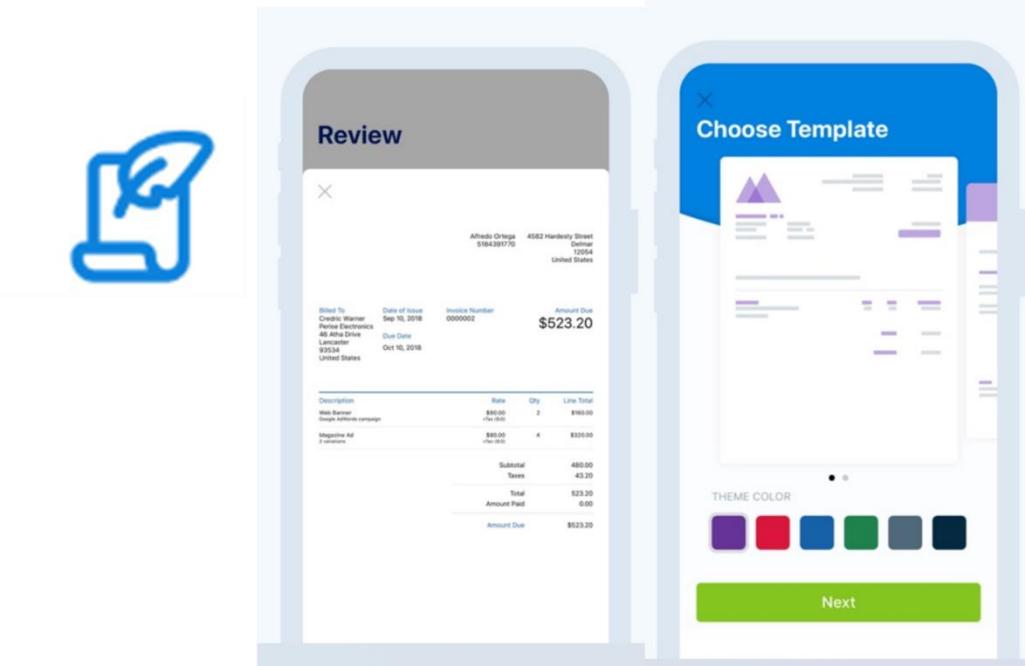


Figure 2.10 Application Create and Just Display Created Receipts [15]

Invoice Creator FreshBooks is a free and easy app just to allow users to create professional receipts in short duration to look as their needs by using existing templates.[15]

### Advantages:

- Support Many languages so the users can create their receipt with their main languages.
- Build own receipt templates.
- Voice recognition.

### Disadvantages:

- User create the receipt manually.
- Receipts stored as PDF file so it's just used to display purchasing item[15]

## 2.4 Comparing Existing Solutions

In Table 2.1 below summarization of similar features between My Receipts application and existing application.

Feature/Application	Genius Scan	Receipt Hog	Invoice Maker	Invoice2go	Invoice Creator	My Receipts
Mobile Application	√	√	√	√	√	√
Data Visualization				√		√
Cloud Computing						√
Refund Exchange Process						√
Adjust Monthly Limit						√
Notification for Receipts						√
Arranging Receipts	√	√	√	√	√	√
Different Receipts Status			√	√		√
Display Receipt information	√	√	√	√	√	√

Table 2.1 Summary of the features between My Receipts Application and related Application

## 2.5 Conclusion

In this chapter we have discussed several technologies background that will be used in My Receipts application such as Cloud computing, Automatic Identification, QR code and Data visualization. Secondly, we discussed about QR history in the applications industry where QR code is an automatic detection and capturing technique. That will be used by My Receipts application to store receipts as QR on the cloud system. We also have talked about the similar IT solution with their benefits and limitations. In the end, we compared My Receipts application with other existing applications.

## **Chapter 3: Elicitation**

### **3.1 Introduction**

Considering the previously mentioned techniques and technologies that is intended to be used in this project, in this chapter we will discuss the analysis phase. The analysis of data that was gathered by interview and online survey. the data collection from some potential stores and users using several elicitation techniques will help us identifying the requirements, these requirements will be analyzed furthermore to conduct the functional and nonfunctional requirements.

### **3.2 Elicitation Techniques**

An essential aspect of data collection is to collect requirements needed to meet the objective of the project as well as meeting the needs of the user. Moreover, data can be gathered from several resources and using multiple elicitation techniques such as interview, observations, and questionnaires.

The first method we used to gather data was to conduct individual interviews with several stores that helped us to identify and define the features and functions of our desired product and how the refund/exchange process flow within any of their systems.

The second method used was the questionnaire which can work with a large sample of the specified population with a relatively low cost and it's easy to analyze.

### **3.3 Participations**

- In interview to gather data we went to a shopping center contacted different stores employees to gather the requirements we need for our project.
- In Questionnaire, we published our online questionnaire to the all type of users in different social media. This allowed us to get an appropriate feedback about our application, Project and idea from a valuable set of users.

### **3.4 Elicitation Result and Analysis of Collected Data**

#### **3.4.1 Interview**

We conducted interviews with several shops in order to get the appropriate information that can serve the idea of My Receipts. So, At the beginning we have compiled the questions that we want to be answered by the shops and we were keen to hold the interview with supervisor of the system in the shops face-to-face in order to understand and realize the seriousness of our project and to provide us about all the technique's that used on the system and how it works given to our

desire to cooperate with the store that has the appropriate system. Refer Appendix A - to check interview details.

### 3.4.2 Questionnaire analysis

A questionnaire is used for collecting data which consist of a set of standardized questions called items that follow a fixed scheme to gather data from users. The form is made via Google forms and then was distributed through different social media applications to be filled with various customers. The questionnaire includes different questions regarding to the purchasing/refunding process, to help define the requirements and the needs of most of the users. The questionnaire is attached in appendix A, and the Analysis are as follows:

- We received 110 responses
- As shown in Figure 3.1 the first question (How often do you return the item you purchased from the shop (example: shirt)?) result 67.3% of people answered they return the item rarely, 22.7% always, and 10% never.

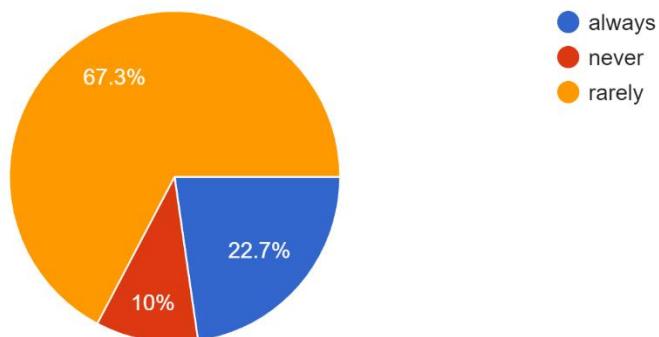


Figure 3.1 Q1 survey result

- The Figure 3.2 shown the second question (How many times do you go to shopping in a month?) result 60% of people answered they go to shopping once, 27.3% twice, and 12.7% more than 3 times a month.

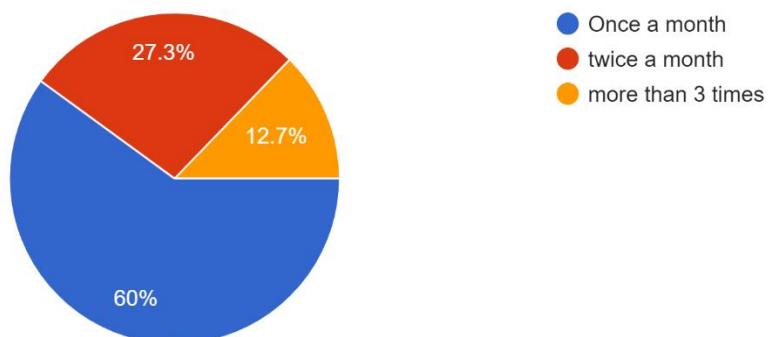


Figure 3.2 Q2 survey result

- As shown in Figure 3.3 the third question (Have you ever been in a situation where you wanted to refund/exchange but didn't find the receipt?) result 87.3% people answered that they didn't found the receipts, and 12.7% answered no they didn't been in forgetting receipt situation.

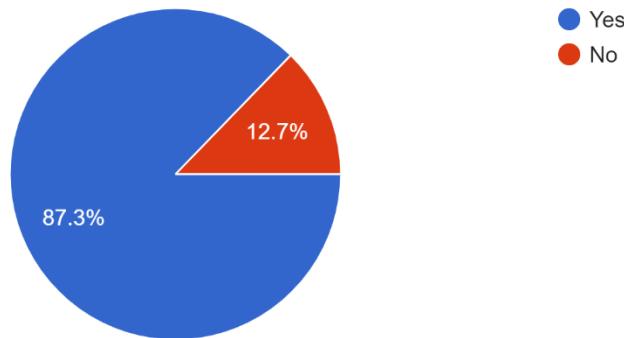


Figure 3.3 Q3 survey result

- The Figure 3.4 show the fourth question (Would you prefer using electronic receipts over paper receipts?) result 90.9% preferred the electronic receipts, and 9.1% are good with paper receipts.

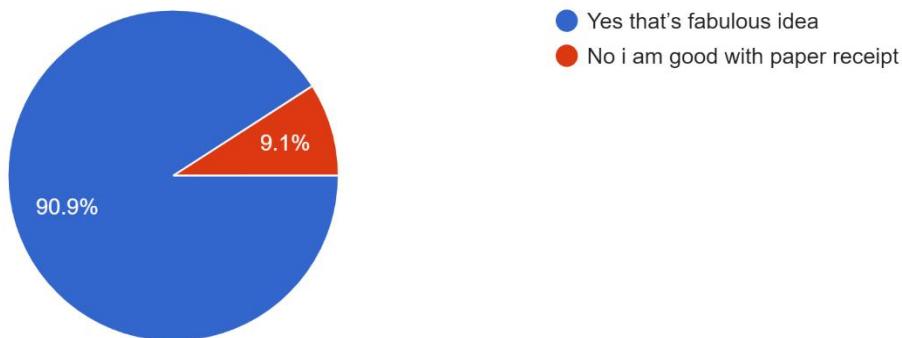


Figure 3.4 Q4 survey result

- As shown in Figure 3.5 the fifth question (Do you think that electronic receipts will solve the problem of forgetting and losing paper receipts?) result 94.5% people answered yes it will solve the problem, and 5.5% answered no.

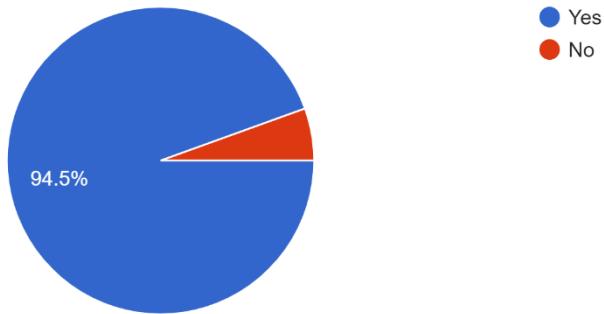


Figure 3.5 Q5 survey result

- The Figure 3.6 shown the sixth question (Have you ever reached the end of the month with no money left for further purchases?) result 69.1% people answered yes, they reach end of the month with no left for purchase purchases, and 30.9% answered no.

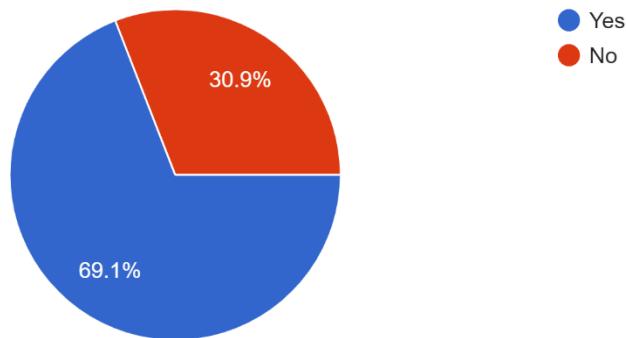


Figure 3.6 Q6 survey result

The information that was gathered using the questionnaire has left us with a better understanding about how the customer sees the application.

Based on the questionnaire, most of the responses was positively supporting the idea of My Receipts regarding question number 5 which had a (94.5%) of agreement upon the idea.

We can conclude from these statistics that most people face issues in their shopping experience when it comes to refunding/exchanging. Thus, our system will solve most of the problems that occur to shoppers with the refunding/exchanging process. Refer to Appendix B - Appendix C - for questionnaire details.

### **3.5 Requirement Specification**

This section will discuss both the functional and nonfunction requirement for the project.

### **3.5.1 Functional Requirements**

This part of the requirements explains description of the service that the software must offer based on the gathered data. The data was gathered by interviewing the shop employees and questionnaire different type of users in beginning of this chapter in section 3.4 Elicitation Result and Analysis of Collected Data. The table below displaying the user and system requirements with the priority of each requirement where the highest indicates what should be implemented as an essential in My Receipts Application.

<b>Requirement ID</b>	<b>Description</b>	<b>Priority</b>
Req 1	The user shall have an account to get the receipts on the System.	High
Req.1.1	The system shall generate QR code as an identity QR that contain user information (email, name, phone number, password) to be used as a unique id when login into system.	High
Req 1.2	The system shall organize the receipt to be displayed as cards in system interface with different border color based on the status of the receipts.	High
Req 1.2.1	The green card border on receipt shall allow user to refund and exchange the item	High
Req 1.2.2	The yellow card border on receipt shall allow user to exchange the item only.	High
Req 1.2.3	The red card border on receipt shall notify the user that its last day to exchange the item only.	High
Req 1.2.4	The gray card border on receipt shall display the user that receipts is expired.	High
Req 1.2.4.1	The system shall make the QR code of receipts faded.	High

Table 3.1 Functional Requirements of My Receipts

<b>Requirement ID</b>	<b>Description</b>	<b>Priority</b>
Req 2	The system should retrieve item information (Quantity, name, price,date) and Identity QR from the store system database.	High
Req 3	The system should generate the receipt QR containing (receipt barcode number, items barcodes numbers)	High
Req 3.1	The system shall store the receipt QR next to the identity QR and item information (Quantity, name, price) of a specific user in the cloud.	High
Req 3.2	The system shall retrieve from cloud the item information and receipt QR to be displayed on the App when click to see receipt field based on the specified user Identity QR.	High
Req 4	The system shall update the receipt QR whenever refund or exchange process occurs.	High
Req 5	The system shall ask the user to set the monthly limit for purchases	Medium
Req 5.1	The system shall notify the user if the monthly limit has been exceeded.	High
Req 6	The system shall calculate monthly limit from the total of each receipt.	High
Req 6.1	The system shall display the data visualized of the purchases on each month.	High
Req 7	The system shall scan the IdentityQR to start purchasing and transferring receipts to application.	High

Table 3.2 Functional Requirements of My Receipts (cont.)

### **3.5.2 Non-functional Requirements**

The non-functional requirement are quality attributes and constraints that is required to be met and achieved by our system to ensure that the final product is within the needed performance and quality.

#### **3.5.2.1 Usability**

The user (Customer) shall be able to use all the functions of the application easily by displaying monthly purchases budget, archive and not archive receipts, and setting monthly limit for purchases.

#### **3.5.2.2 Availability**

Application is free of cost for customers and should operate on any smartphone devices that has IOS or Android Operating System.

#### **3.5.2.3 Scalability**

The cloud should be suitable for accepting any amount of data that is created.

#### **3.5.2.4 Maintainability**

The application should be flexible when it comes to accepting any future improvements and any additive features.

#### **3.5.2.5 Security**

The system shall not disclose any personal information about customers.

## **3.6 Conclusion**

This chapter discussed the Election Techniques, used for data collection and gathering which are the questionnaire and the interview. After that we have analyzed the result of election techniques. Based on the result of the election we determined the requirement specification and analysis which are functional and non-functional requirements of My Receipts Application.

## Chapter 4: Analysis Model

### 4.1 Introduction

This chapter we will be starting the analysis phase using different diagrams. We will elaborate each use case by describing it as well as displaying the traceability matrix mapping between each use case and the appropriate functional requirements.

### 4.2 Use case Diagram

The use case defines the interactions between the actors and the system by which will clarify the system functional requirements mentioned in the previous chapter section 3.5.1 Functional Requirements.

The Figure 4.1 shows the use case diagram of My Receipts Application.

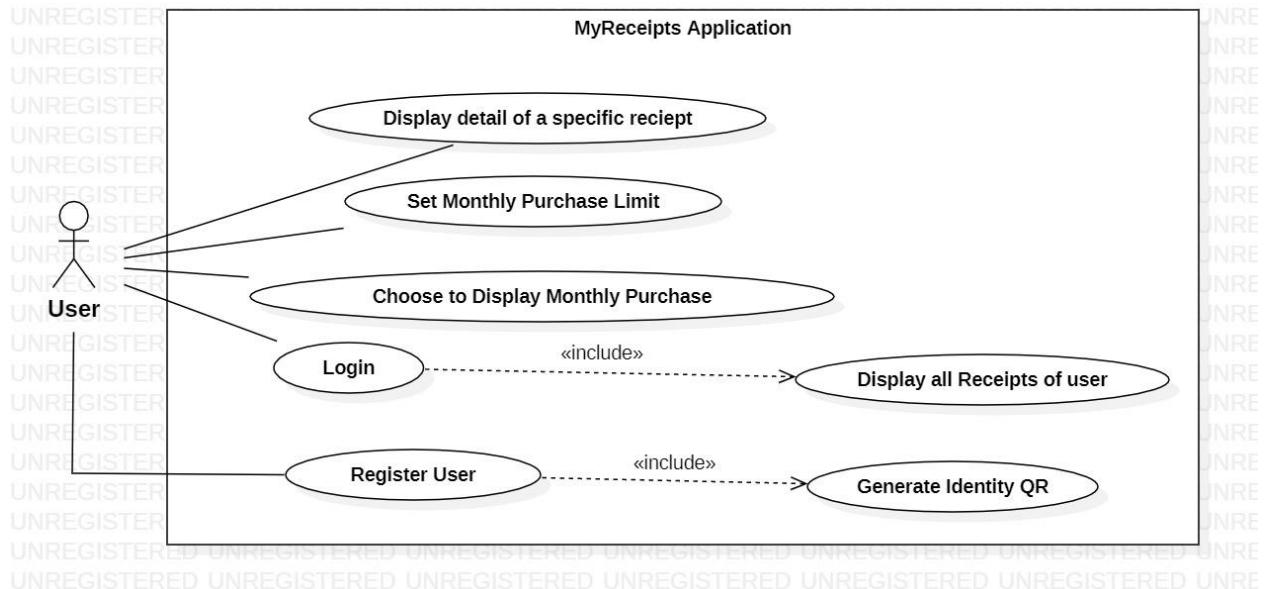


Figure 4.1 My Receipts Application Use case of the customer

### 4.3 Elaborated Use case

This section will provide a description for each use case including the trigger actions, the preconditions, the success condition, the basic flow, and the possible exceptions that may occur and what actions should be taken by the application to react to it.

#### **4.3.1 Elaborated Use case” Login”**

- **Actors:** Application User
- **Summary:** The Application User Will Login into My Receipts Application
- **Precondition:** The User Should has an Account to Login
- **Success End Condition:** User Has Entered the Application Successfully
- **Include:** Display All Receipts of User
- **Extend:** None
- **Basic Flow:**
  1. The Application prompt the user to enter his/her phone number and password.
  2. The application will validate user information.
  3. Application will display all user receipts.
- **Exceptions:**
  1. If Password or Phone Number is wrong.  
Error Message Should Appear.

#### **4.3.2 Elaborated Use case “Display All Receipts of User”**

- **Actors:** Application User.
- **Summary:** The Application will display all the receipts of the specific user.
- **Precondition:** to load user receipts user should be logged in successfully into My Receipts.
- **Success End Condition:** all receipts are displayed for the specific user.
- **Include:** none.
- **Extend:** none.
- **Basic Flow:** none.
- **Exceptions:** none.
- 

#### **4.3.3 Elaborated Use case “Register User”**

- **Actors:** Application User.
- **Summary:** the user who do not has an account will create an account in My Receipts Application.
- **Precondition:** user doesn't have an account.
- **Success End Condition:** user account successfully created in My Receipts.
- **Include:** Generate Identity QR.
- **Extend:** none.
- **Basic Flow:**
  1. The application prompt user to enter his required information.
  2. The filled information is sent to QR generator to create unique identity QR.
  3. The generated QR is sent to cloud.
  4. Created Identity QR of user is Displayed.

- **Exceptions:**
  1. If Phone Number already registered  
Error Message Should Appear.

#### **4.3.4 Elaborated Use case “Generate Identity QR”**

- **Actors:** Application User
- **Summary:** Identity QR is generated for newly user account.
- **Precondition:** the required information to create account is filled.
- **Success End Condition:** the QR is created for newly created account.
- **Include:** none.
- **Extend:** none.
- **Basic Flow:** none.
- **Exceptions:** none.

#### **4.3.5 Elaborated Use case “Set Monthly Purchase Limit”**

- **Actors:** Application User.
- **Summary:** the user can set monthly amount for purchases for a specific month to not exceed and keep up within their budget.
- **Precondition:** user shall have an account in My Receipts.
- **Success End Condition:** Limit amount is successfully adjusted and set.
- **Include:** none.
- **Extend:** none.
- **Basic Flow:**
  1. The user selects the month that where to set the limit to his/her budget.
  2. Type the specified amount for the limit.
  3. Limit has been assigned successfully.
- **Exceptions:** none.

#### **4.3.6 Elaborated Use case “Display Details of Specific Receipt”**

- **Actors:** Application User.
- **Summary:** application display the details of receipts with items and total amount of receipt along with the receiptQR for a specific receipt based on the user selection.
- **Precondition:** user shall have an account in My Receipts.
- **Success End Condition:** View Details of Receipts.
- **Include:** none.
- **Extend:** none.
- **Basic Flow:**
  1. User logged into application.
  2. List of receipts are displayed.
  3. Select the appropriate receipt.

4. Details of receipt are displayed.
- **Exceptions:** none.

#### **4.3.7 Elaborated Use case “Choose to Display Monthly Purchase”**

- **Actors:** Application User.
- **Summary:** the amount of purchases that has been spent in a month will be displayed as visualized graph.
- **Precondition:** user shall have an account in My Receipts.
- **Success End Condition:** a visualization graph is displayed.
- **Include:** none.
- **Extend:** none.
- **Basic Flow:** none.
- **Exceptions:** none.

### **4.4 Traceability Matrix:**

In Table 4.1 to Table 4.3 below the tractability matrix maps the diagram of use cases with functional reequipments that we have discussed above in section 3.5.1 Functional Requirements and 4.2 Use case Diagram.

Test cases	Requirement					
	Req 1	Req 1.1	Req 1.2	Req 1.2.1	Req 1.2.2	Req 1.2.3
Use case	Already Registered User	Generate User Identity QR	Organize Receipts as Cards	Refund and Exchange Card Receipts	Exchange Card Receipts Only	Last Day to Exchange Card Receipts
UC1	x		x	x	x	x
UC2	x		x	x	x	x
UC3		x				
UC4	x	x				
UC5	x					
UC6	x					
UC7	x					

Table 4.1 traceability matrix of the project

Test cases	Requirement					
	Req 1.2.4	Req 1.2.4.1	Req 2	Req 3	Req 3.1	Req 3.2
Use case	Receipt Card Expired	QR code is Faded	Retrieval of information and IDQR from DB	Receipts will have a QR Code	Store Generated QR in the Cloud	Retrieve Stored QR code from Cloud
UC1	x					x
UC2	x					
UC3						
UC4						x
UC5						
UC6		x	x	x	x	x
UC7			x			x

Table 4.2 traceability matrix of the project (cont.)

Test cases	Requirement					
	Req 4	Req 5	Req 5.1	Req 6	Req 6.1	Req 7
Use case	QR Code is Updated	Set Purchase Limit Monthly	Reminder if purchase limit Exceed	Calculate Total Receipts Amount Monthly	Display Monthly Purchase	Scan Purchase Item
UC1		x	x	x	x	
UC2	x					
UC3						x
UC4						
UC5		x	x	x		

UC6	x					
UC7				x	x	

Table 4.3 traceability matrix of the project (cont.)

## 4.5 Sequence Diagram

Sequence diagram shows the order of interactions between classes in term of exchanging the messages and it describes each use case in a step by step manner to reach the specified purpose by the application.

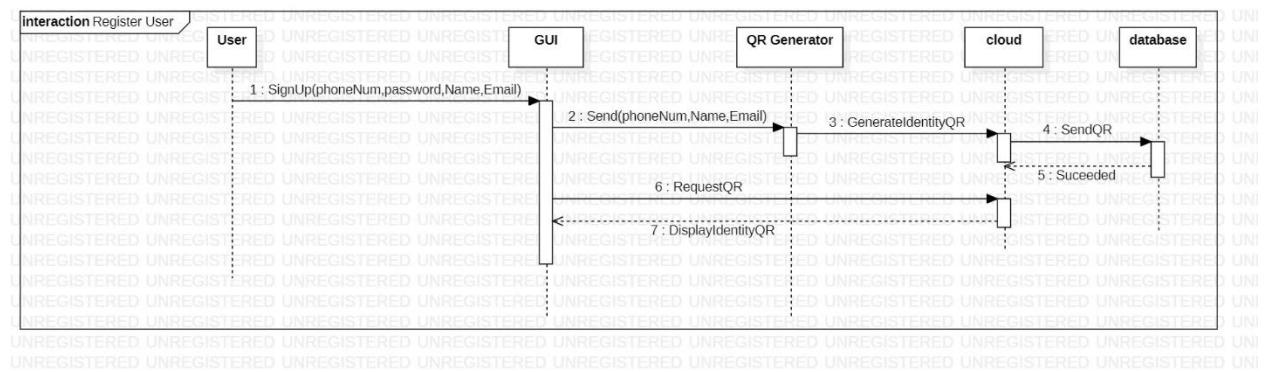


Figure 4.2 Sequence Diagram Register User

As shown in Figure 4.2 above, the register user sequence is Signing up new user with his information

Send the filled information to QR generator to generate identity QR of user. The generated IdentityQR will be sent to the cloud to be stored, then the cloud will send the IdentityQR that was generated to the system database to be stored in the appropriate table.

After that the database notify the cloud that QR has been saved successfully stored in appropriate table.

GUI will send a request to get identityQR, then the cloud will return the stored IdentityQR of the user back to GUI.

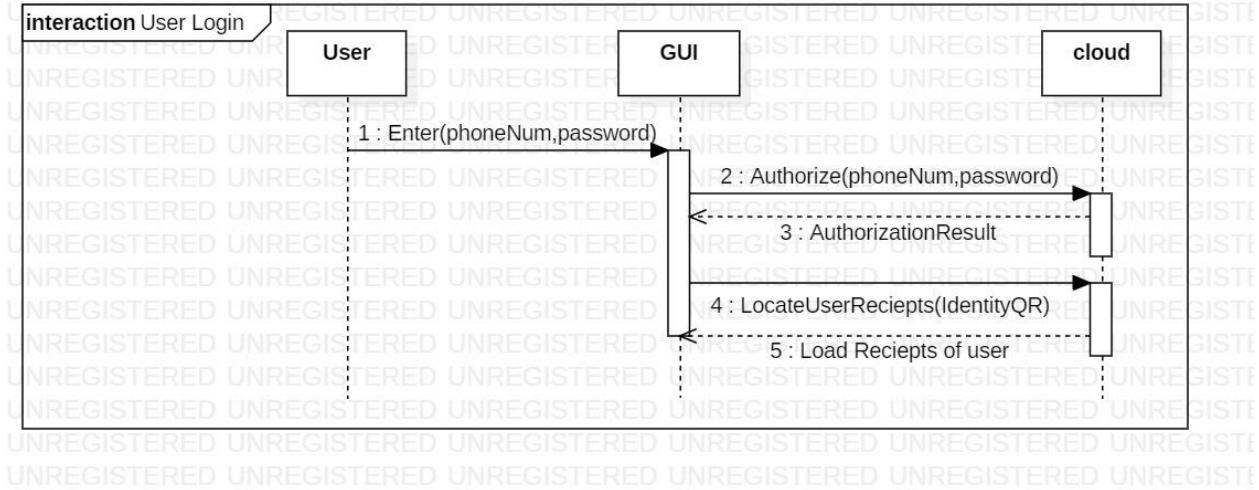


Figure 4.3 Sequence Diagram User Login

The Figure 4.3 above shows the sequence diagram of user login in My Receipts Application. The user will enter his phone number and password to enter the application then will check in the cloud if its saved and will get back a response massage to inform the user that the logging process was successful, then locate the user receipts based on the IdentityQR and display it back on the application.

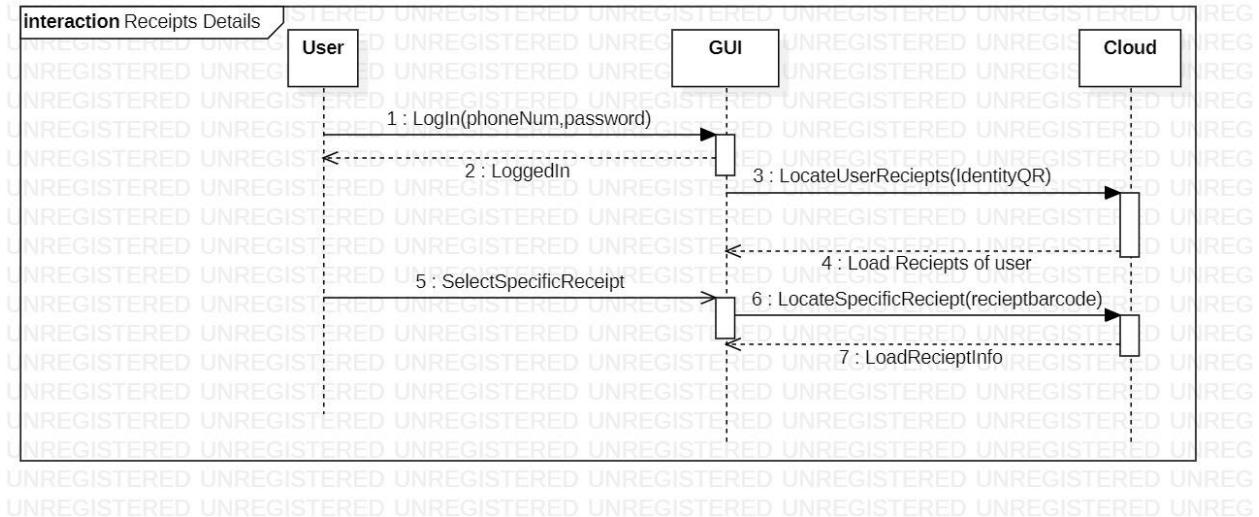


Figure 4.4 Sequence Diagram Receipts Details

Figure 4.4 above shows Receipts Details Sequence.

After user login and locate unique identity QR of the user, the cloud will retrieve and load all receipts of user that are saved in cloud.

The user will select a specific receipt, then it will be located in the cloud and based on selected receipts items and information will be retrieved and loaded from the cloud.

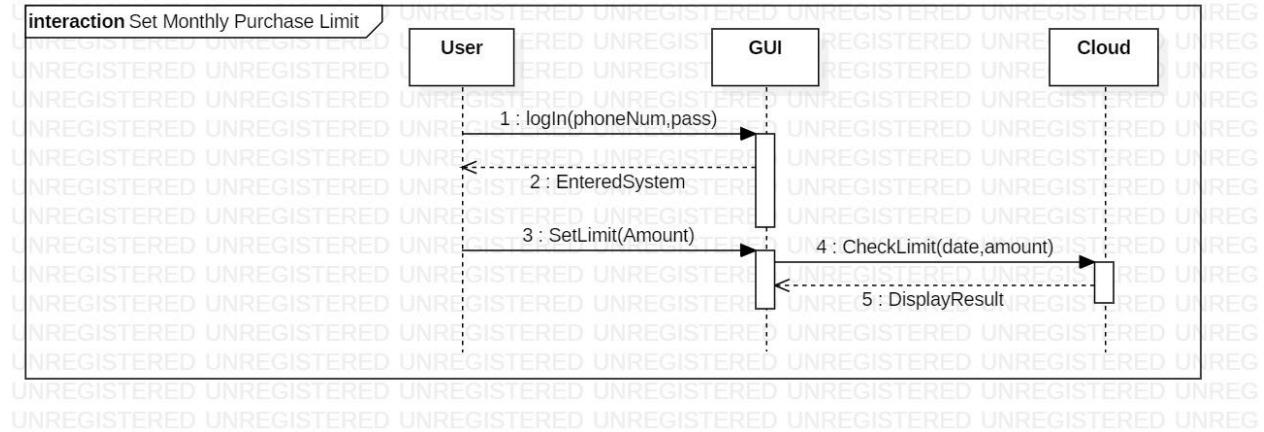


Figure 4.5 Sequence Diagram Set Monthly Purchase Limit

Figure 4.5 above shows set monthly purchase interaction.

The user will set amount of money to spend in the selected month which will be the monthly limit for that month, after that the system will check the limit by calculating all the receipts of a specific month using the date parameter and comparing it with the amount and then display result to the user.

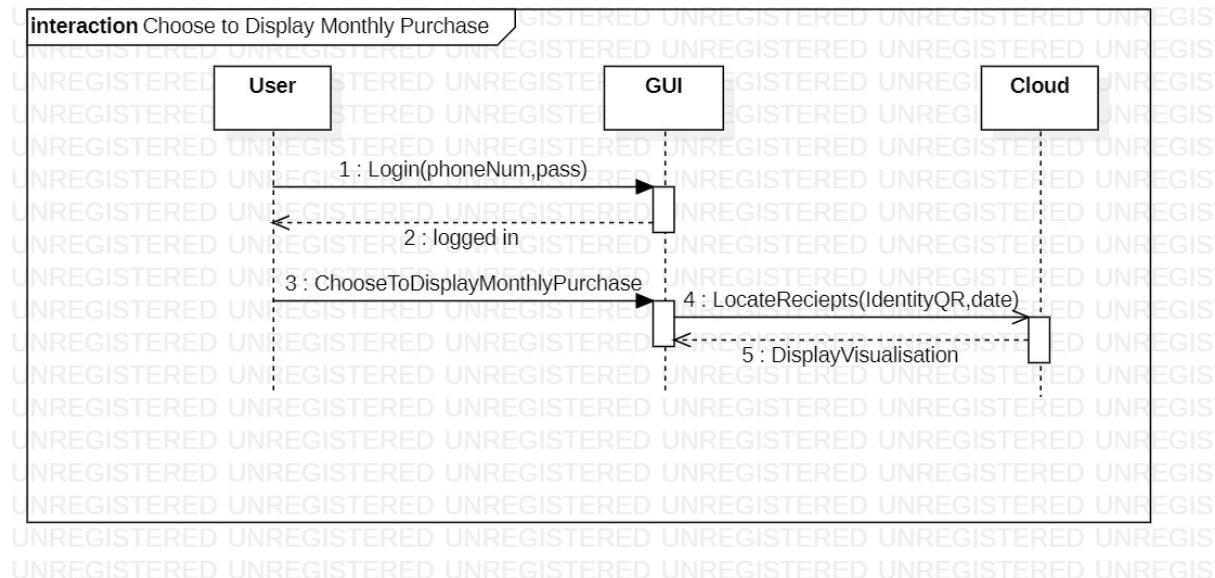


Figure 4.6 Sequence Diagram choose to display monthly purchase

The Figure 4.6 above shows choose month to display monthly purchase interaction, is to show the result of amount of purchase spent each month.

After the user login he/she will choose to display the monthly visualization, the purchases will be located from the cloud then it will be displayed as a graph containing the purchases of all months within a year.

## 4.6 Data Model

Data model won't be required in our project since we are going to use our cloud to store all information for any further operation. all data will be retrieved at first from the store existing database to our cloud.

## 4.7 Design Choices

Design pattern is a description for how to solve a problem that can be used in many different situations [16]. There are 3 types of design pattern creational, structural, and behavior.

In this project the behavior type is used by observer pattern. As the observer defines one to many dependencies between object so that one object changes the state, all its dependents are notified and updated automatically.

In My Receipts Application we have one receipts will contain many items. The total amount of receipt can increased/decreased when the refund/exchange process occur.

So, the receipt class is the subject will have operations attach, detach and notify using observer item object.

And the item class is the observer will have update operation using receipt class object.

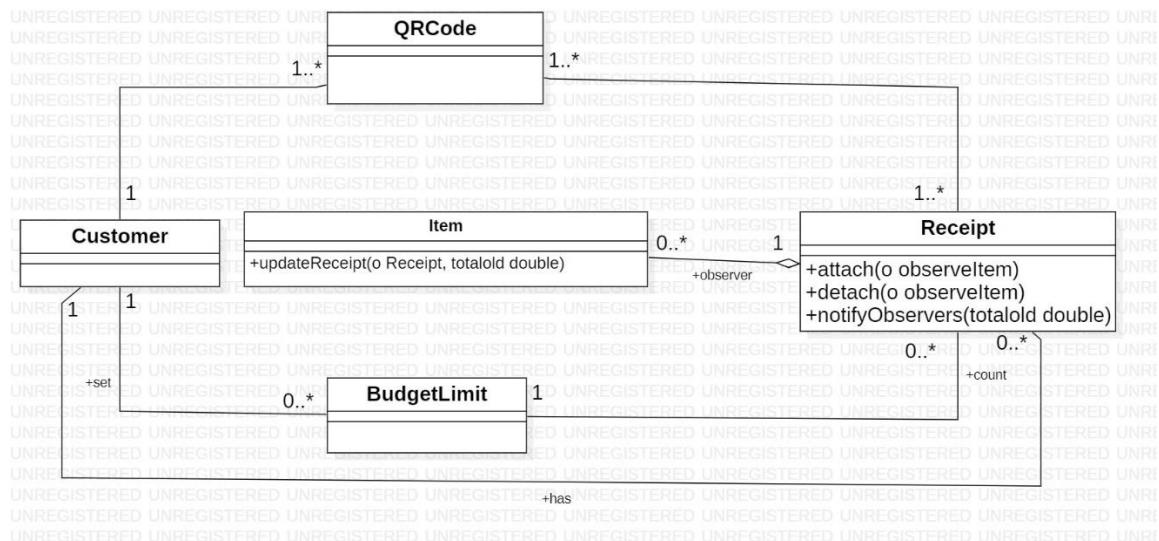


Figure 4.7 observer pattern for My Receipts

## 4.8 Class Diagram

In software engineering, a class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects [17].

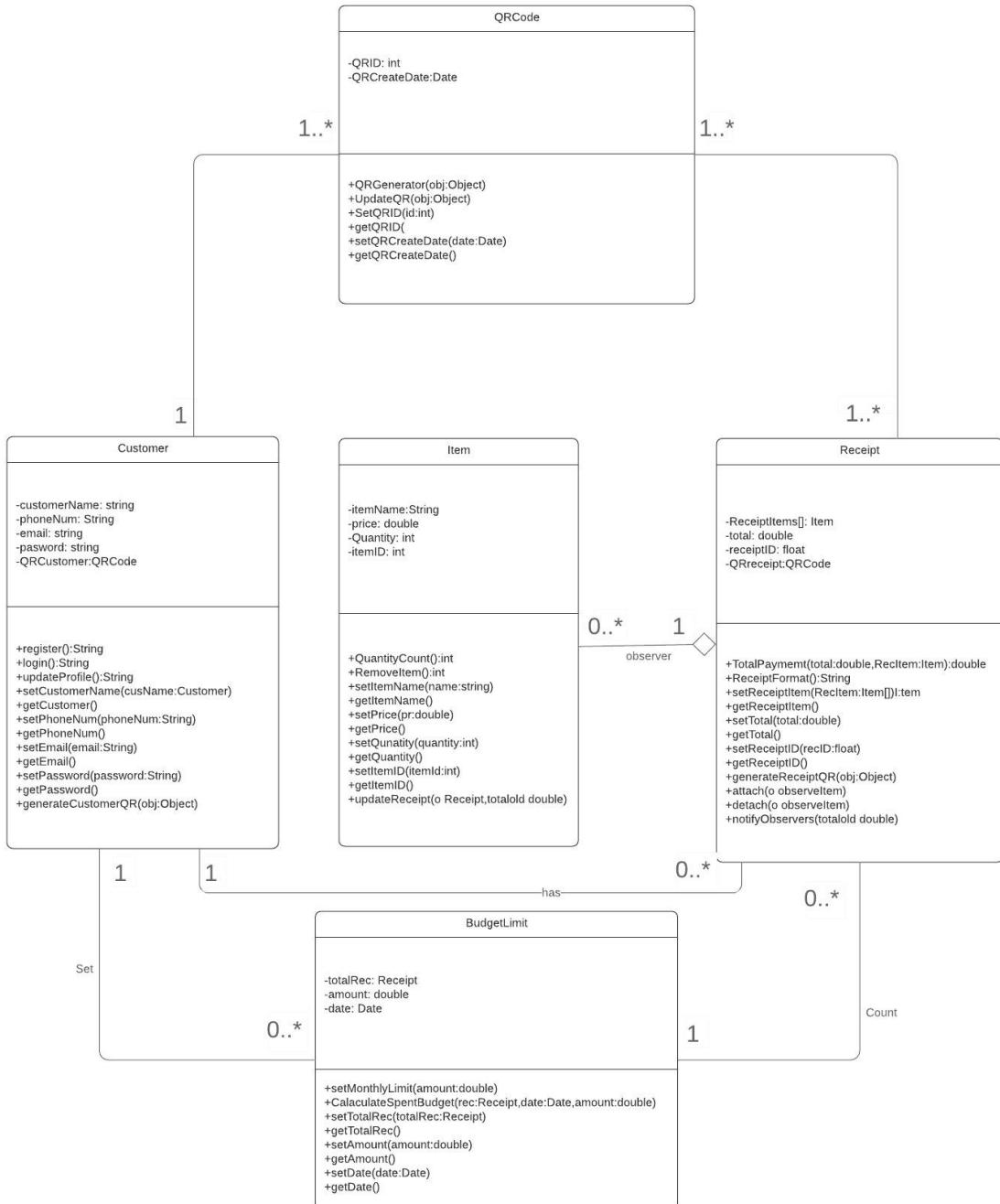


Figure 4.8 Class Diagram for My Receipts Application

The class diagram of My Receipts Application has five main classes (Customer, BudgetLimit, Item, Receipt, QRCode). The QR class will be responsible on working with the QR generating and transferring functions, the Customer class will deal with most of the users functions regarding the registrations, login and the user associated information, the Item class which is concerned with items within the receipts, then the receipt class that includes all operations that will be applied on the receipts and lastly the BudgetLimit class to manipulate the setting amount for each month and other related calculations.

## 4.9 Conclusion

In this chapter we presented various diagrams which include, the use case that defined the interaction between the actor , the application and the system as well as elaborating each use case, sequence diagram to describe the sequential ordering of interactions between the objects, and lastly to understand the software concept we displayed the class diagram.

# Chapter 5: Design Model

## 5.1 Introduction

This chapter will include the design phase of our project, which has the low-level architecture, and graphical user interface prototype.

## 5.2 Low level Architecture

Figure shows the low-level architecture of our application.

The low-level architecture is the stage where the actual software components are designed. We have highlighted the main component which are user, smartphone, and shop purchase system. We are using cloud technology to store all user, items, and receipt information.

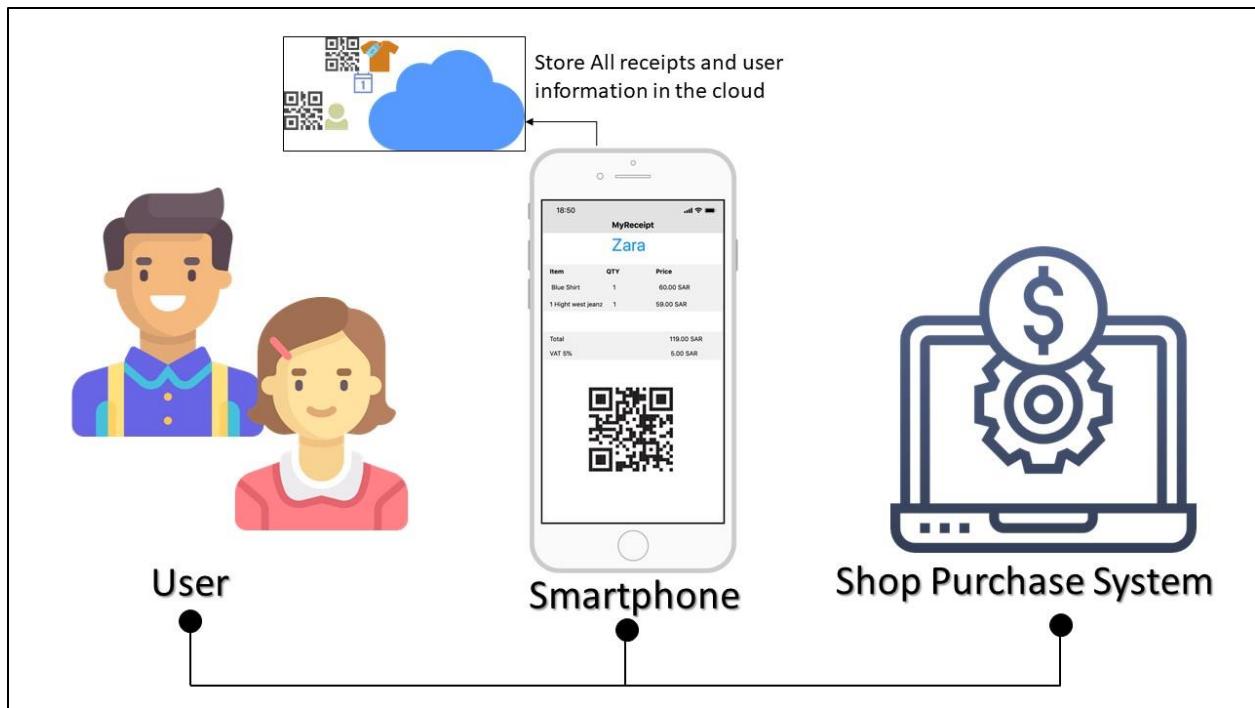


Figure 5.1 My Receipts Low-Level Architecture

### 5.3 Graphical User Interface (GUI) Design

Graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical elements such as windows, icon and buttons [18].

In this section we will going to provide our suggested software design that going to show the main functionality of the system.

Figure 5.2 represent the sign-up page into My Receipts Application.

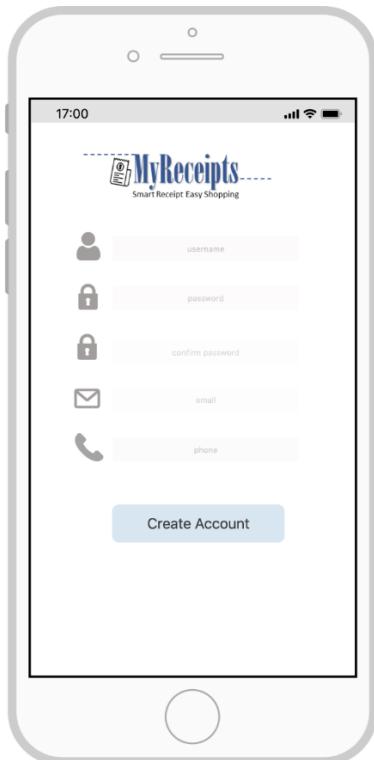


Figure 5.2 Signup interface

Figure 5.3 shows the login page of the user when he/she wants to login to My Receipts application.

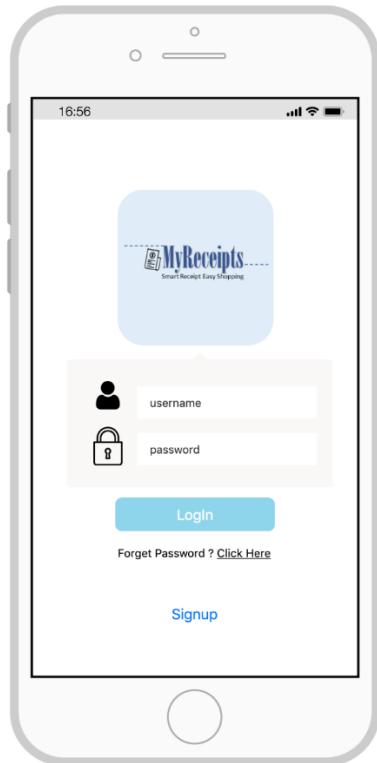


Figure 5.3 User Login interface

Figure 5.4 represent the identity QR for recently user just signup into the My Receipts.



Figure 5.4 user identity QR interface

Figure 5.5 show the list of tabs in our application.

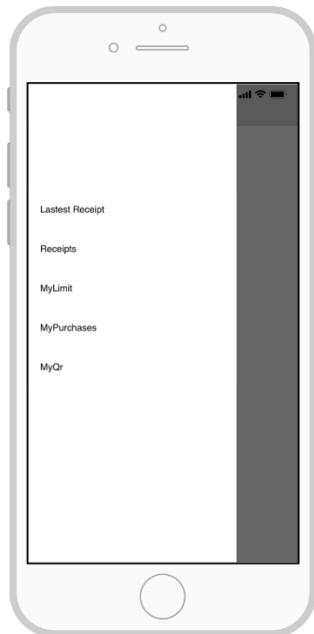


Figure 5.5 menu or navigation bar interface

Figure 5.6 shows the arranged receipts from newest to oldest

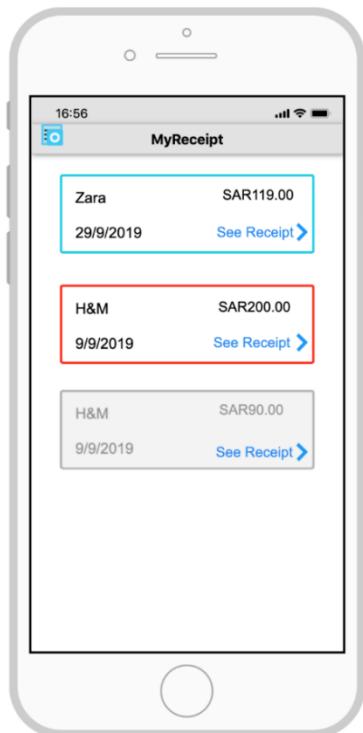


Figure 5.6 Arranged Receipts interface

Figure 5.7 shows the details of the selected receipt with QR so customer can use it for refund/exchange and see details of purchases for that receipt.



Figure 5.7 Detailed receipt interface

Figure 5.8 shows the details of selected receipt with QR faded user can't use it any more for refund/exchange because the date is exceeded, but user can use it to see purchase details of that receipt.



Figure 5.8 Detailed Expired Receipt interface

Figure 5.9 represents the adjusting monthly purchase limit that user use to select the specific month with amount to spent in that month.

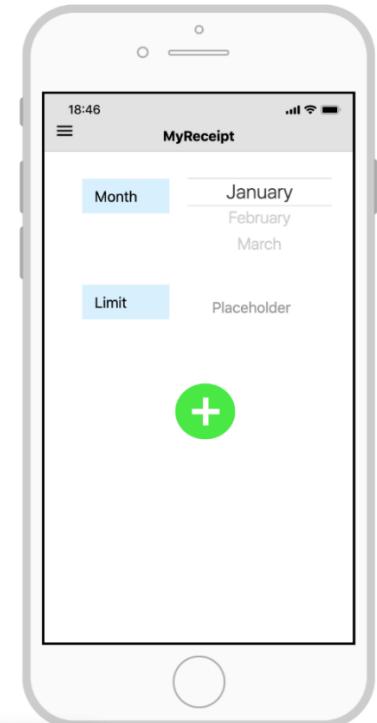


Figure 5.9 adjust monthly limit interface

Figure 5.10 shows the monthly purchase spent on each month.



Figure 5.10 monthly purchase visualization interface

## **5.4 Conclusion**

Overall, in this chapter, we previewed the low-level architecture to display the main components details of the project, then the prototype to display each functionality provided by our application interface.

# **Chapter 6: Implementation**

## **6.1 Introduction**

In this chapter, we will be presenting the implementation phase of MyReceipts application, by presenting the tools, software, and languages that have been used to develop MyReceipts application, as well later on we have discussed the problems that we have faced during implementing MyReceipts application.

## **6.2 Tools**

Main hardware and software tools will be discussed here.

### **6.2.1 Hardware**

The main hardware used in the development phase are:

- Laptop: used in implementing, emulating, testing MyReceipts application.
- Smartphone: To check design preferences, check the availability on stores, and test for any errors.

### **6.2.2 Software**

We used three software elements throughout the development phase: firebase for cloud storage of receipts, Command Line (CMD) for running the ionic software, Microsoft visual studio to implement the MyReceipt Application Code.

#### **6.2.2.1 Firebase**

Firebase is Google's mobile platform that helps you quickly develop high-quality apps and grow your business, it provides you with the ability to use the cloud firebase storage, which is a flexible, scalable database for mobile, web, and server development from **Firebase** and **Google Cloud Platform**[19].

Cloud firebase is used to save receipts of users along with their purchased items with the allowance of it to be automatically linked to user devices using user id.

#### **6.2.2.2 Command Line (CMD)**

CMD is a command line interpreter application available in most Windows operating systems. It's **used to** execute entered commands. Most of those commands automate tasks via scripts and batch files [20].

We used CMD to run and emulate our ionic code on a Connected device via ionic cordova *run* command which helped in Building the app and deploying it to devices and emulators.

### 6.2.2.3 Microsoft Visual Studio

It is a Microsoft professional development tool that provides a full integrated environment that enables the programmers and testers to program and code with different programming languages [21].

It is mainly used to develop Websites, mobile applications and computer games.

## 6.3 System implementation

In this section, we will be presenting the implementation of the MyReceipts application by using the features that was already defined in chapter 1 and the prototype which was designed in chapter 4 as a starter.

### 6.3.1 Register Page Implementation

Register page is where the user begins his/her experience in MyReceipts application. It's one of the main features without it the user can't use the application.

Figure 6.1 loads a snapshot of all the users on the database to the (users' array) to check it in case of any duplication of new registration.

Figure 6.2 shows creation of user account.

Every user is stored to firebase storage by clicking the button create account based on id.

```
--  
52  ngOnInit() {  
53    firebase.database().ref('/users').once('value').then((snapshot) => {  
54      this.users = snapshot.val();  
55      if (this.users == null) {  
56        this.users = [];  
57      }  
58      console.log(this.users);  
59    });  
60    // this.firestore.collection(`users`).valueChanges().subscribe(res => {  
61    //   this.users = res;  
62    //   console.log(this.users);  
63    // });  
64  }  
65  
66  
67  showPassword() {  
68    this.passwordShown = !this.passwordShown;  
69  }  
70  
71  showPassword2() {  
72    this.passwordShown2 = !this.passwordShown2;  
73  }
```

Figure 6.1 Loading user variables code

```

76  async signup() {
77    var error = false;
78    if (!error && this.username.length <= 0) {
79      this.presentToast("error_username");
80      error = true;
81    }
82    if (!error && this.password.length <= 0) {
83      this.presentToast("error_password");
84      error = true;
85    }
86    if (!error && this.password != this.confirm_password) {
87      this.presentToast("error_password_mismatch");
88      error = true;
89    }
90    if (!error && !this.validator.validateEmail(this.email)) {
91      this.presentToast("error_email");
92      error = true;
93    }
94    if (!error && this.phone.length <= 0) {
95      this.presentToast("error_phone");
96      error = true;
97    }
98    if (!error) {
99      const username = this.username;
100     const password = this.password;
101     const email = this.email;
102     const phone = this.phone;
103     error = false;
104     console.log(this.users);
105     for (var i = 0; i < this.users.length; i++) {
106       if (this.users[i].email == this.email) {
107         this.presentToast("Email Used Before");
108         error = true;
109         break;
110       } else if (this.users[i].username == this.username) {
111         this.presentToast("Username Used Before");
112         error = true;
113         break;
114       } else if (this.users[i].phone == this.phone) {
115         this.presentToast("Phone Used Before");
116         error = true;
117         break;
118     }
119     if (!error) {
120       let id = this.users.length;
121       firebase.database().ref('users/' + id).set({
122         "username": username,
123         "password": password,
124         "email": email,
125         "phone": phone
126       }).then((snapshot: any) => {
127         let user = new User(
128           id,
129           username,
130           password,
131           email,
132           phone);
133         this.loggedUser.data = user;
134         this.loggedUser.replaceInLocalStorage();
135         this.router.navigate(['/profile']);
136         this.presentToast("User Registered Successfully!");
137       });
138     }
139   }
140 }
141 }
142 } else {
143   // await loading.dismiss();
144 }
145 }
146 }
147 }
148 }
149 }
150 async presentToast(field) {
151   this.translateService.get(field).subscribe(
152     async value => {
153       const toast = await this.toastController.create({
154         message: value,
155         duration: 2000
156       });
157       toast.present();
158     }
159   }
160 }
161 }

```

Figure 6.2 User account creation condition code

### 6.3.2 Login Page Implementation

The Login page allow user to log into its profile and use the system application functions.

Figure 6.3 Shows the implementation of the login function where it takes the user entered values (username, password) and check if its correct and then navigate user to the profile page.

```
136 var error = false;
137
138 if (!error && this.username.length <= 0) {
139   this.presentToast("error_username");
140   error = true;
141 }
142
143 if (!error && this.password.length <= 0) {
144   this.presentToast("error_password");
145   error = true;
146 }
147
148 if (!error) {
149   const username = this.username;
150   const password = this.password;
151
152   firebase.database().ref('/users').once('value').then((snapshot) => {
153     let users: any = snapshot.val();
154     let found = null;
155     let found_id = 0;
156
157     for (var i = 0; i < users.length; i++) {
158       if (users[i].username == this.username && users[i].password == this.password) {
159         found = users[i];
160         found_id = i;
161         break;
162       }
163     }
164
165     if (found != null) {
166       let user = new User(found_id,
167         found.username,
168         found.password,
169         found.email,
170         found.phone);
171       this.loggedUser.data = user;
172       this.loggedUser.replaceInLocalStorage();
173       this.router.navigateByUrl('/profile');
174       this.presentToast("User_Login_Successfully!");
175     } else {
176       this.presentToast("User name or password not correct!");
177     }
178   }));
}
```

Figure 6.3 user Login condition code

### 6.3.3 User Identity QR Generation

User QR generation is one of the main features in our project that when the QR of the user is created after register on MyReceipts application the QR will be used later as identity QR to exchange/refund an item in the store instead of paper receipt.

Figure 6.4 show the main functions of profile page.

Figure 6.5 shows the display of the QR code of the user with the specified attributes.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Figure 6.4 functions of profile page

```

-- | -- | --
29 | <ion-row>
30 |   <ion-col size="12" class="ion-text-center">
31 |     <qrcode [qrdta="encodeData" [width]="256" [errorCorrectionLevel]="'M'" 
32 |       (click)="printQR()"></qrcode>
33 |   </ion-col>

```

Figure 6.5 specified tag to display QR

#### 6.3.4 Storing Receipts to Specific User

There is 2 way to adding the receipt to specific user by the firebase cloud storage or using the visual studio to implement it through the code.

Figure 6.6 shows how to add the receipt to specific user by id (because each user has id based on it the receipts are added). This part is commented in code now since it was used for the first creation only, but any further changes will be through the firebase.

```

-- | -- | --
61 let receipts = [];
62 firebase.database().ref('/receipts').once('value').then((snapshot) => {
63   receipts = snapshot.val();
64   if(receipts==null){
65     receipts=[];
66   }
67 });
68 setTimeout(() => {
69   let id = receipts.length;
70   firebase.database().ref('receipts/' + id).set({
71     "title": "Zara1",
72     "total": 250,
73     "vat": "3",
74     "date": "2020-03-15",
75     "user id": 5,
76     "items": [
77       {
78         "title": "T32",
79         "price": 22,
80         "qty": 2
81       }
82     ]
83   }).then((snapshot: any) => {
84   });
85 }, 5000);
86

```

Figure 6.6 Adding receipts to user by code

Figure 6.7 show how to add specific stored receipt to specific user in firebase by id just change the id of user and the receipt will be added to that user.

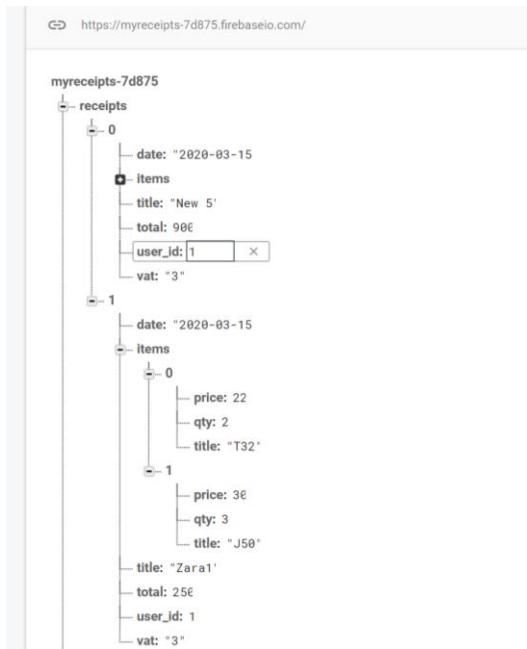


Figure 6.7 Adding receipts to user by firebase storage

### 6.3.5 Display List of User Receipts

Display list of receipt is feature that shows all user receipt in the page.

Figure 6.8 shows how all receipts of specific user is retrieved to the page based on the user id as well as the total of the receipt calculation, clicking on the “see receipt “button will navigate to the receipt detail page.

```
42  firebase.database().ref('/receipts').once('value').then((snapshot) => {
43    this.receipts = snapshot.val();
44    if (this.receipts == null) {
45      this.receipts = [];
46    }
47    this.receipts.sort(this.dynamicSort("date"));
48
49    for (var i = 0; i < this.receipts.length; i++) {
50      this.receipts[i].id = i;
51    }
52
53    for (var i = this.receipts.length - 1; i >= 0; i--) {
54      if (parseInt(this.receipts[i].user_id) != parseInt(userid)) {
55        this.receipts.splice(i, 1);
56      }
57    }
58  });
59}
60
61
62
63
64
65
66  goToReceipt(receipt) {
67    let navigationExtras: NavigationExtras = { state: { receipt: receipt } };
68    this.router.navigate(['receipt-details'], navigationExtras);
69  }
70
71  dynamicSort(property) {
72    var sortOrder = 1;
73    if (property[0] === "-") {
74      sortOrder = -1;
75      property = property.substr(1);
76    }
77    return function (a, b) {
78      /* next line works with strings and numbers,
79       * and you may want to customize it to your needs
80       */
81      var result = (a[property] < b[property]) ? -1 : (a[property] > b[property]) ? 1 : 0;
82      return result * sortOrder;
83    }
84  }
85
86  getTotal(receipt) {
87    let total = 0;
88
89
90    for (var i = 0; i < receipt.items.length; i++) {
91      total += parseFloat(receipt.items[i].price);
92    }
93
94
95    total = total - (total * 0.05);
96
97
98    return this.round2Digit(total);
99  }
100
101  round2Digit(num){
102    return Math.round((num + Number.EPSILON) * 100) / 100;
103  }
104}
105
```

Figure 6.8 code of displaying list of user receipts

### 6.3.6 View Receipt Details

Figure 6.9 shows the details of the receipt regarding retrieving the QR and displaying it as well as the items, total and vat of the specified receipt.

```

60
61     getQRCode(receipt) {
62       let str =
63         "https://myreceipts-7d875.firebaseio.com/receipts/" +
64         receipt.id;
65       return str;
66     }
67
68     getTotal(receipt) {
69       let total = 0;
70       for (var i = 0; i < receipt.items.length; i++) {
71         total += parseFloat(receipt.items[i].price);
72       }
73       total = total - (total * 0.05);
74       return this.round2Digit(total);
75     }
76
77     getVat(receipt) {
78       let total = 0;
79       for (var i = 0; i < receipt.items.length; i++) {
80         total += parseFloat(receipt.items[i].price);
81       }
82       return this.round2Digit((total * 0.05));
83     }
84
85     printQR(receipt){
86       let str =
87         "https://myreceipts-7d875.firebaseio.com/receipts/" +
88         receipt.id;
89       console.log(str);
90     }
91
92     round2Digit(num){
93       return Math.round((num + Number.EPSILON) * 100) / 100;
94     }
95
96   }
97 }
```

Figure 6.9 code of receipts details page

### 6.3.7 Update Receipt QR

Updating the receipt QR is the process that happens when an item is refunded/exchanged from the shop.

The process happens by going to the link of the created firebase storage and select the item to perform the intended operation on the specific receipt.

Figure 6.10 shows that storage has many receipts, each receipt is linked to a specific user to display it in user screen.

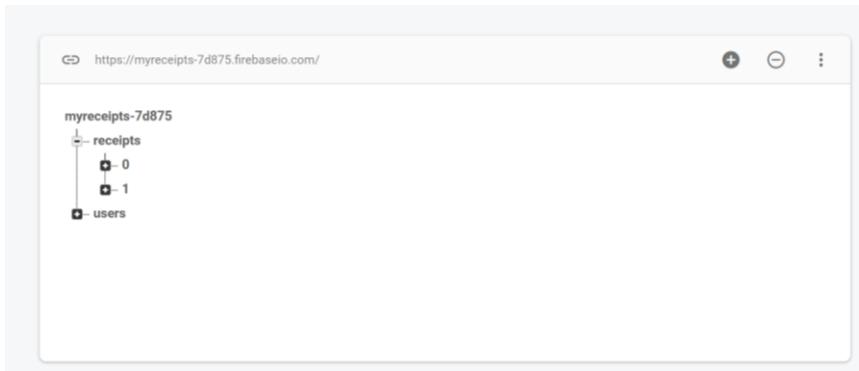


Figure 6.10 Receipts firebase storage

Figure 6.11 shows how the receipts attributes is saved in the cloud firebase storage when delete/update option is clicked the total of receipt is changed to that user.

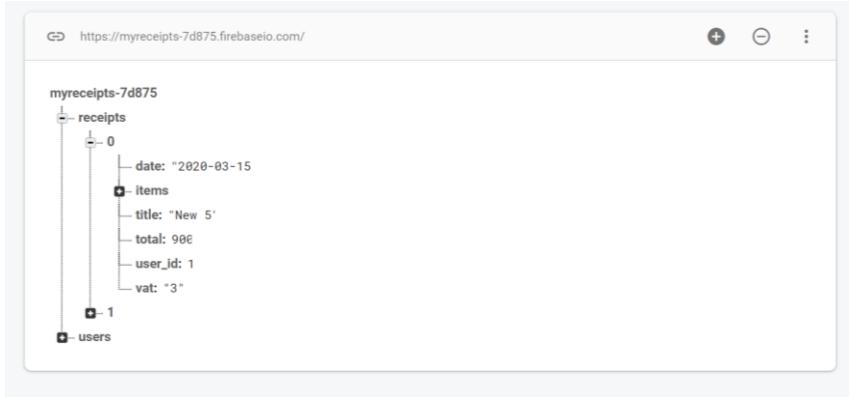


Figure 6.11 Receipts attributes stored in firebase

## 6.5 Code debugging and Troubleshooting issues

Throughout the journey of developing our project we have encountered many problems; details will be discussed below about the outcomes and the problems and how it was dealt with and fixed.

### 6.5.1 QR Generation and Update

QR code was a new thing to learn about we have faced many problems in finding the way of generating and updating the QR for receipts and user. By searching in different resources and videos we have resolved this challenge.

In order for the application to process and flow properly the QR needed to be properly and correctly linked and coded to the pages.

First, we have just used QR libraries in ionic to store one text field in the QR after that we have got further and combined all needed thing in one QR.

### 6.5.2 Linking Application to firebase storage

Linking the application was the most challenging thing that we faced because we have never worked with firebase storage and ionic framework.

The main problem was I finding the way of linkage between pages and firebase, after searching we have found the libraries needed to be exist in each code page.

This problem was dealt with by importing the angularfirestore package to each page.

### **6.5.3 Calculating the Total of Receipts**

We have faced a problem in calculating the total of specific receipt, which is the mismatch of the total value, which was correctly shown on the console but when it was retrieved to the application page when run it was false.

The problem was due to missing of setting the variable on the html code page and was solved by just adding the missed variable.

## **6.6 Conclusion**

In this chapter firstly, we have introduced the tools and programs that was used to implement the project function. Also, we have discussed the implementation phase of MyReceipts application based on what we have specified before regarding the features and the design. Later on, have displayed the problems that we have encountered during creation of the project

# Chapter 7: Testing

## 7.1 Introduction

A software lifecycle includes testing phase which is a mandatory part of any software development to ensure that it works properly and correctly regarding the requirement specified by the user before the final product is delivered for use. In this chapter we will discuss two types of testing that we have applied on our project MyReceipts application.

## 7.2 Unit Test

Unit test is one of the most important types of tests that can be applied on software that tests every single function in the software application separately, it can be done manually and or by using automated testing tool. Table 7.1 shows the summary of unit testing applied.

Test	Input	Expected Output	Actual Output	Result
Initiate the application	No input	Initiate the app	Worked successfully	pass
Create application	No input	Creating the Work	Created successfully	pass
Create identity QR	Database URL	QRCode	QRCode generated successfully	pass

Table 7.1 Unit Test Table

the following figures shows some of the automated tests that were conducted through karma and jasmine programs that tests applied unit test in the application. For instance, Figure 7.1 below shows the unit test code for running and crating component and Figure 7.5 shows test user QR generation

### 7.2.1 Test Running app and its component

```
35
36  it('should create the app', () => {
37    const fixture = TestBed.createComponent(AppComponent);
38    const app = fixture.debugElement.componentInstance;
39    expect(app).toBeTruthy();
40  });
41
42  it('should initialize the app', async () => {
43    TestBed.createComponent(AppComponent);
44    expect(platformSpy.ready).toHaveBeenCalled();
45    await platformReadySpy;
46    expect(statusBarSpy.styleDefault).toHaveBeenCalled();
47    expect(splashScreenSpy.hide).toHaveBeenCalled();
48  });

```

Figure 7.1 Unit Test Code for Creating and Initiate the App

## 7.2.2 initialize app code

```
36
37     constructor(
38         private storage: SharedStorage,
39         private platform: Platform,
40         private splashScreen: SplashScreen,
41         private statusBar: StatusBar,
42         private translate: TranslateService,
43         public alertController: AlertController,
44         private translateConfigService: TranslateConfigService,
45         private location: Location
46     ) {
47         this.initializeApp();
48         this.statusBar.styleDefault();
49         this.splashScreen.hide();
50         // this.presentConfirm();
51     }
52
53     getTranslations() {
54         this.translate.get('confirm').subscribe(
55             value => {
56                 this.confirm = value;
57             }
58         );
59         this.translate.get('ok').subscribe(
60             value => {
61                 this.ok = value;
62             }
63         );
64         this.translate.get('cancel').subscribe(
65             value => {
66                 this.cancel = value;
67             }
68         );
69         this.translate.get('are_you_sure_exit_app').subscribe(
70             value => {
71                 this.are_you_sure_exit_app = value;
72             }
73         );
74     }
--
```

Figure 7.2 App Code of Creating and Initiate 1

```
75
76     initializeApp() {
77         this.platform.ready().then(() => {
78             this.platform.backButton.subscribe(() => {
79                 document.addEventListener('backbutton', function (event) {
80                     event.preventDefault();
81                     this.presentConfirm();
82                     event.stopPropagation();
83                 }, false);
84             });
85             this.statusBar.styleDefault();
86             this.splashScreen.hide();
87             this.translateConfigService.setDefaultLanguage();
88             // var lng = this.storage.getValue(SharedStorage.APP_LANG);
89             var lng = 'en';
90             if (lng) {
91                 this.selectedLanguage = lng;
92             } else {
93                 this.selectedLanguage = 'ar';
94             }
95             localStorage.setItem(SharedStorage.APP_LANG, this.selectedLanguage);
96             //this is to determine the text direction depending on the selected language
97             this.translate.onLangChange.subscribe((event: LangChangeEvent) => {
98                 this.selectedLanguage = event.lang;
99                 this.getTranslations();
100            });
101            this.getTranslations();
102        });
103        if (!firebase.apps.length) {
104            firebase.initializeApp(this.config);
105        }
106    }
--
```

Figure 7.3 App Code of Creating and Initiate 2

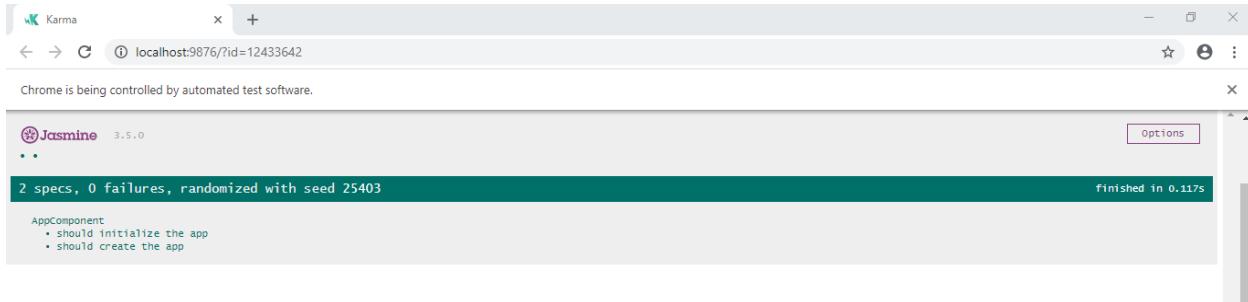


Figure 7.4 Unit Test Result of Initiate and Creating the App

### 7.2.3 Test generating QRcode from URL

```

34
35   it('Create New QR Code', () => {
36     let url = "https://myreceipts-7d875.firebaseio.com/users/7";
37     let barcodeScanner = new BarcodeScanner();
38
39
40     barcodeScanner
41       .encode(barcodeScanner.Encode.TEXT_TYPE, url)
42       .then(
43         encodedData => {
44           console.log(encodedData);
45         },
46         err => {
47           console.log("Error occurred : " + err);
48         }
49       );
50       expect(1 + 1).toBe(2);
51     });
52

```

Figure 7.5 Unit Test Code of QR code Generation

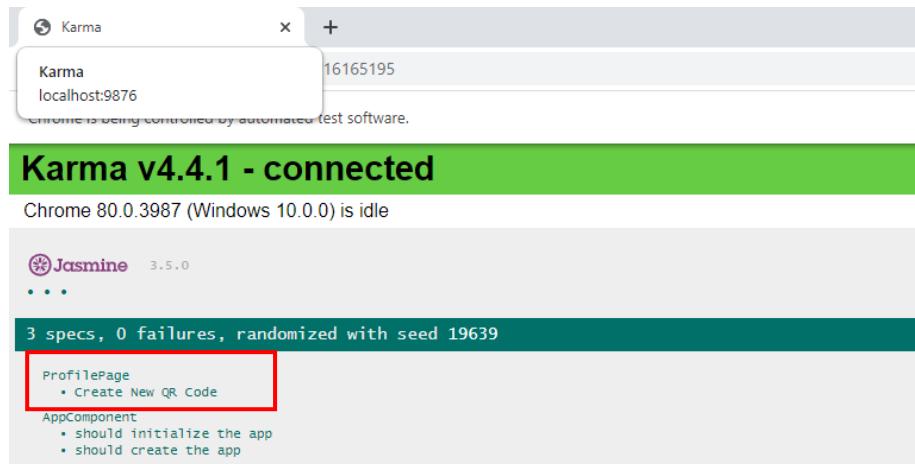


Figure 7.6 Result of QR unit test

## 7.3 Integration Testing

Another way of software testing is integration testing which is considered an important part of the software lifecycle which aims to test the integration and relationship between components to check if they work correctly.

### 7.3.1 Tested integration Functions

Table 7.2 shows some test cases for some tasks to perform the integration test.

Tasks	Test case Scenarios	Results
login	Open application >> enter username & password >> press login	Pass
Show user identity QR	Open application >> do login >> profile page with user identity QR	Pass
signup	Open application >> enter username, password, email, and phone number >> press create account	Pass
Show receipts	Open application >> do login >> profile page >> press receipts button	Pass
Show receipt details with receipt QR	Open application >> do login >> profile page >> press receipts button >> see receipt button	Pass

Table 7.2 Table of Integration Test

### 7.3.2 Integration Testing Result

As a result, we have reached a successful output that shows achieving our goals and requirements correctly and fully.

## 7.4 Conclusion

In this chapter we have tested our project different components by using different testing types, tools and methods to evaluate it and make sure it meets the user requirements and expectations.

## Chapter 8: Results and Discussion

### 8.1 Introduction

In this chapter, we will show the interfaces of MyReceipts application and the main features. Also, we will provide main objectives accomplished and some of the limitations in the project.

### 8.2 Graphical User Interfaces

This section will show main screen of MyReceipts application with description of its main functions.

#### 8.2.1 Start Screen

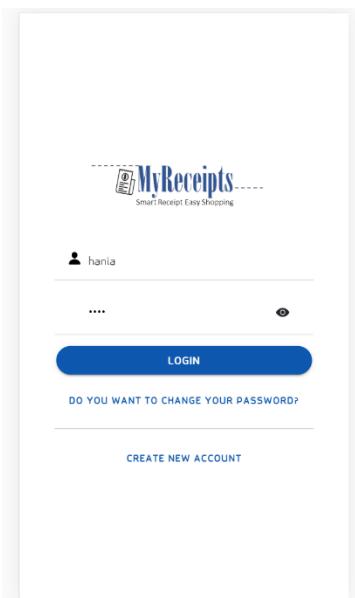


Figure 1.1 shows the start page of MyReceipts application it's the login page which allow user to get into the application to display its receipts. The Login page contain:

- Username and password field to allow user goes to Receipts page.
- User ability to change their password by click change password button then it will allow user to change password in reset-password page.
- User can't goes to the receipts page if they don't have account on MyReceipts app so they must create an account by clicking create new account and its will allow user goes to sign up page.

Figure 8.1 MyReceipts Login or Home Page

### 8.2.2 Reset-Password Screen

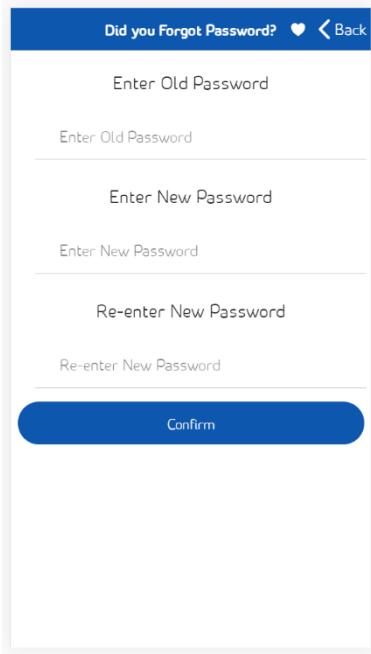


Figure 8.2 MyReceipts Reset Password Page

Figure 8.2 shows reset-password screen its will be displayed when user forget his/her password and it contain 3 field and 2 button which is:

- **Enter old password field:** User will press label to enter his/her old password.
- **Enter new password field:** User will press label to enter his/her new password.
- **Re-enter new password field:** User will press label to re-enter his/her new password for the confirmation.
- **Confirm button:** This will make the update when user press confirm button by changing user password.
- **Back button:** User will press it when he/she doesn't want to change password.

### 8.2.3 Sign up Screen

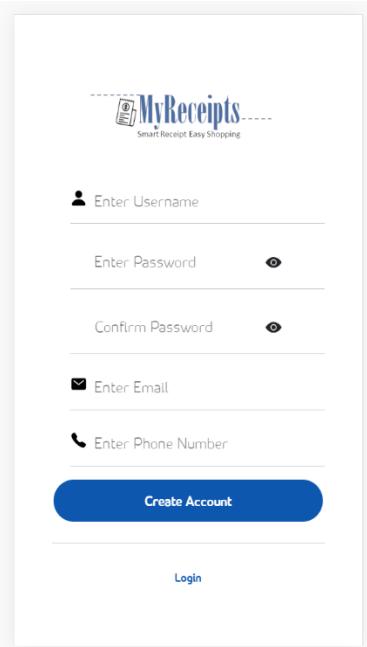


Figure 8.3 MyReceipts Sign up Page

Figure 8.3 shows Sign up page this page is visited if user doesn't have an account on MyReceipts application and it has many field that will displayed on user Identity QR to be known for the application system need and for shopping store and this user information will be stored on firebase Also, 2 buttons which is :

- **Username field:** User press this field to enter his/her username that will be used when he/she is logging into the application.
- **Password field:** User press this field to enter his/her password that will be used when he/she is logging into the application.
- **Confirm password field:** User press this field to re-enter his/her password for the confirmation.
- **Email field:** User press this field enter his/her email for the application system need.
- **Phone Number field:** User press this field enter his/her phone number application system need.
- **Create account button:** when user press create account filed the user account will be created and save into the firebase and will allow user goes to profile page that shows user Identity QR which have been generated automatically.
- **Login button:** if user already have account on MyReceipts application he/she go back into login page.

## 8.2.4 User Profile Screen



Figure 8.4 MyReceipts User Profile Page

Figure 8.4 shows Profile screen which refer to user identity QR page that will be generated automatically after user create an account on MyReceipts application by displaying his/her name and user will have the ability to get back to profile page when he/she want, this page will contain:

- **User identity QR:** That contain user information and it will be used by shopping stores to save all user receipts.
- **Sign out button icon:** User will press this button when he/she want to sign out from application.
- **Receipts list button icon:** User will press this button if he/she want to display all receipts.

## 8.2.5 Receipts List Screen

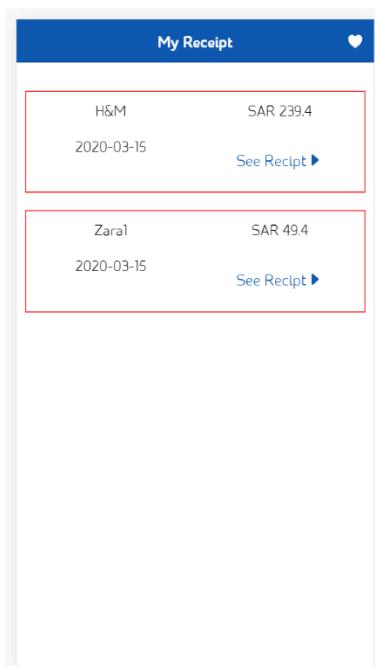


Figure 8.5 MyReceipts User Receipts List Page

Figure 8.5 shows user receipts screen will display all user receipts as list of cards each card will shows basic information of receipts such as title of store, date and price also each one will contain:

- **See Receipt button:** user will press this button if he/she want to display whole receipts details with main QR that will be used for refunding/exchanging item.
- **Profile button icon:** At the top of the page the user will have the ability to return to the profile page by press this button.

### 8.2.6 Receipt Details Screen



Figure 8.6 MyReceipts Receipt Details Page

The screen as shown in **Figure 8.6** will show the whole receipts details as they displayed on shopping store system with items, quantity, price, total and VAT also with the main QR that will be used by the user for refunding/exchanging item from stores. All this information will be saved on the firebase with and the QR can be updated if user make any change on the receipt. The following page will contain two buttons:

- **Profile button icon:** User will press it if he/she want to return to the profile page which contain user identity QR.
- **Back button:** User will press it when he wants to get back to the receipts card list.

### 8.3 Objective Accomplished

MyReceipts application has main objectives that have been achieved through implementation phase and make sure about correction in testing phase, here are the main objectives:

- User must have an account to start MyReceipts application experience.
- User QR generation after creating an account in MyReceipts application.
- Displaying user receipts as cards.
- Each receipt card has an ability to display receipts details with main QR that will be used for refunding/exchanging the items.

Also, there is an additional objective to provide a security perspective that help MyReceipts application to gain customer satisfaction which is user ability to change his/her account password when needed or forgotten.

## **8.4 Limitations of the Project**

Every developer faces problem during implantation phase due to some situation we have faced, the project has some limitation which are not achieved as following:

- The ability of setting the receipts cards with different colors based on the status of each receipt.
- The shortage of time caused us to let some of the function unimplemented such as the monthly limit, notifications, data visualization.

## **8.5 Conclusion**

In this chapter, we discussed the interfaces and how it works regarding our application, describe the objectives that have met. Lastly, we presented the limitation that we have faced.

## **Chapter 9: Conclusion and Future Work**

At the beginning of this senior project, we got the idea of MyReceipts application that is a very helpful system for a lot of people and based on this idea we have analyzed, designed, implemented and tested MyReceipts application. Firstly, planned the phases that needed to be accomplished by defining the project scope, stakeholders, budget and time. After that, we have conducted the requirements by using the different elicitation techniques (surveys, interviews), which helped in defining the operations and functions that is related to the application throughout the analysis phase. After the analysis phase we started designing and structuring the interfaces of the application. In implementation phase we used for programming ionic framework with combination of angular Node.js, and java script, after we make sure that the code works correctly, we started testing our application by two techniques: unit and integration testing.

### **9.1 Challenges and limitations**

We have faced some challenges while we were working along the project phases, most of the problems were during the implementation phase, we will discuss some of them below:

- Programming with the new language “ionic framework”.
- Using and downloading angular/node.js.
- Dealing with the QR code and linking it to the pages.
- Using the firebase storage and learning how to store and link each item.

Most of the challenges were dealt with and correctly overcome. On the other side the application has some limitation due to the time pressure and shortage, these limitations will be discussed in the future works.

## **9.2 Future work**

MyReceipts in the future can achieve many features based on the limitations that we have faced, and it will help customers to have a better shopping experience, some of these features are:

- Change the cards color based on the status of the receipt.
- Setting the monthly limit of each month for the user to keep up within the budget and not exceed it
- Data visualization to show the spending of each month.
- Notifications regarding the status of the receipts.
- Notifications on any sales regarding the stores we partnered with.

MyReceipts application has faced many problems during implementation of the software some of the problem was solved during that phase but few was addressed in the future work such as: the notification, status, monthly limit, and data visualization. To conclude with, some of these limitations that were not solved and handled were to put in mind to be solved and developed in the future.

## References

- [1] "Agile Methodology Guide: Understanding Agile Testing - QASymphony." [Online]. Available: <https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/>. [Accessed: 08-Dec-2019].
- [2] "The problem with paper receipts," *TreeHugger*. [Online]. Available: <https://www.treehugger.com/corporate-responsibility/problem-paper-receipts.html>. [Accessed: 04-Nov-2019].
- [3] "About Uber - Our Story - Vision for Our Future," *Uber*. [Online]. Available: <https://www.uber.com/sa/en/about/>. [Accessed: 04-Nov-2019].
- [4] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," p. 7.
- [5] U. J. Bora and M. Ahmed, "E-Learning using Cloud Computing," *Int. J. Sci. Mod. Eng.*, pp. 9–12.
- [6] "Automatic Identification and Data Collection (AIDC)." [Online]. Available: <http://www.mhi.org/fundamentals/automatic-identification>. [Accessed: 04-Nov-2019].
- [7] "Three\_QR\_Code.pdf."
- [8] "What is a QR Code? | QRcode.com | DENSO WAVE."
- [9] "What is data visualization? A definition, examples, and resources." [Online]. Available: <https://www.tableau.com/learn/articles/data-visualization>. [Accessed: 04-Nov-2019].
- [10] L. Chou, "Top 16 Types of Chart in Data Visualization," *Medium*, 23-Jul-2019. [Online]. Available: <https://towardsdatascience.com/top-16-types-of-chart-in-data-visualization-196a76b54b62>. [Accessed: 04-Nov-2019].
- [11] "Genius Scan - A scanner app in your pocket," *The Grizzly Labs*. [Online]. Available: <https://thegrizzlylabs.com/genius-scan>. [Accessed: 04-Nov-2019].
- [12] "What is Receipt Hog?," *Receipt Hog Help Center*. [Online]. Available: <http://receipthog.zendesk.com/hc/en-us/articles/227155648-What-is-Receipt-Hog->. [Accessed: 04-Nov-2019].
- [13] "Invoice Maker by NorthOne by NorthOne." [Online]. Available: <https://appadvice.com/app/invoice-maker-by-northone/1368542810>. [Accessed: 04-Nov-2019].
- [14] A. S. Author, "Invoice2go Invoice & Estimate by Invoice2go.com," *AppAdvice*. [Online]. Available: [/app/invoice2go-invoice-estimate/540236748](https://appadvice.com/app/invoice2go-invoice-estimate/540236748). [Accessed: 04-Nov-2019].
- [15] A. S. Author, "Invoice Creator by FreshBooks by FreshBooks," *AppAdvice*. [Online]. Available: [/app/invoice-creator-by-freshbooks/1421252234](https://appadvice.com/app/invoice-creator-by-freshbooks/1421252234). [Accessed: 04-Nov-2019].
- [16] "Software design pattern - Wikipedia."
- [17] "What is Class Diagram?" [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>. [Accessed: 12-Dec-2019].
- [18] "Graphical user interface - Wikipedia." [Online]. Available: [https://en.wikipedia.org/wiki/Graphical\\_user\\_interface](https://en.wikipedia.org/wiki/Graphical_user_interface). [Accessed: 12-Dec-2019].
- [19] "What is Firebase? The complete story, abridged. - Firebase Developers - Medium." <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0> (accessed Apr. 20, 2020).

- [2] Facebook, Twitter, and LinkedIn, “What Is the Command Prompt in Windows, and How Do I Open It?,” *Lifewire*. <https://www.lifewire.com/command-prompt-2625840> (accessed Apr. 20, 2020).
- [3] “Visual Studio Code,” *Wikipedia*. Apr. 13, 2020, Accessed: Apr. 20, 2020. [Online]. Available:  
[https://en.wikipedia.org/w/index.php?title=Visual\\_Studio\\_Code&oldid=950773435](https://en.wikipedia.org/w/index.php?title=Visual_Studio_Code&oldid=950773435).

## Appendix A - Interview

First, We made interview by visiting few stores on mall of Arabia and we discovered that each system on stores work differently from other stores even if it's under same company like AlHokair Group company that include many stores each of them has its own techniques on the system such as Promode and Massimo Dutti stores whereas both we realized their systems works in a very unprofessional way, but the store that we prefer and interested to work with their system its femi9 store its under Saudi company called AlMashat they have a great and very professional system that could be helpful to our project.

<b>Interviewee:</b>	<b>Interviewer:</b>
Supervisor of femi9 Shop	Shahad Alattas, Nedaa Saleh, Hania Hafeez
<b>Location:</b>	<b>Appointment Date:</b>
AL Arab mall, femi9 store	10/10/2019 Start Time: 2:00PM End Time: 2:30;
<b>Objectives:</b>	Understand how the system works in terms of storing and the use of barcode.
<b>Agenda:</b> Introduction Background to our project Starting an interview: Question #1 Question #2 Question #3 Question #4 Question #5 Question #6 Question #7 Closing	<b>Approximate Time:</b> 2 min 2 min 5 min 3 min 3 min 5 min 3 min 2 min 2 min 1 min
<b>Question #1</b> How your System works?	<b>Answer:</b> System of the store has a fixed functions that performs the required operations such as purchase, exchange or return operations and only me and the employees of the store has the authority to use these functions without knowing the storage process of this data as this process is confidential known only by employees who work on IT section at Company.
<b>Question #2</b>	<b>Answer:</b>

Do you use barcode/QR on your system process?	Yes we have 2 barcode that been used on our operation , first barcode its for receipt we scan it to make the operation , the second one its called item code its unique for each item only the last 3 digit will be different even if it's the same item there will be many size from it ( S,M,L) that will be known from this unique item code .
<p><b>Question #3</b></p> <p>Do you have records for all the operation on the system?</p>	<p><b>Answer:</b></p> <p>No , on the store system we just made the operations such as purchasing, refunding or exchange items but each operation that have been made on the system it's not stored here, maybe stored by the employees who is responsible for storage data and operations of the system at IT management but not on shop system.</p>
<p><b>Question #4</b></p> <p>How could be refund/exchange items process on your system?</p>	<p><b>Answer:</b></p> <p>Ok first we scan the barcode of the receipt then choose refund function from the system and select the item that need to refund it and answer the reason of refund this item because its damage or just for changing the size , Also to be sure from the item that need to be refund by the customer from checking the item code after that we print the new receipt for the customer with that update without the receipt barcode.</p>
<p><b>Question #5</b></p> <p>Do you have an update for every purchasing, refunding, exchanging item?</p>	<p><b>Answer:</b></p> <p>Yes , when the customer buy 2 item and she need to refund one of them , when I scan the barcode of the receipt I will delete that item and the receipt will be update it on the system directly , so the customer can use the receipt again for the other item.</p>
<p><b>Question #6</b></p> <p>what is the information you take it from customer when refunding/exchanging an item?</p>	<p><b>Answer:</b></p> <p>No, We didn't take information from customer.</p>
<p><b>Question #7</b></p> <p>Where is the information of purchasing is stored? Database, Cloud, Excel Sheet?</p>	<p><b>Answer:</b></p> <p>As I told you we don't know where the data stored after the purchasing or any process, we have authority only to use this fixed function. Its IT management job.</p>

<b>Interviewee:</b> Supervisor of Massimo dutti Shop	<b>Interviewer:</b> Shahad Alattas, Nedaa Saleh, Hania Hafeez
<b>Location:</b> AL Arab mall, femi9 store	<b>Appointment Date:</b> 10/10/2019 Start Time: 2:40PM End Time: 3:00
<b>Objectives:</b>	Understand how the system works in terms of storing and the use of barcode
<b>Agenda:</b> Introduction Background to our project Starting an interview: Question #1 Question #2 Question #3 Question #4 Question #5 Closing	<b>Approximate Time:</b> 2 min 2 min 3 min 4 min 3 min 2 min 1 min 1 min
<b>Question #1</b> How your System on store works?	<b>Answer:</b> Our system work without using database it's based on the RFID reader that placed on each item so when we made scan to this reader we make the purchasing process, after that all daily purchasing process collected together and stored on USB. and send it by email to the company at Spain.
<b>Question #2</b> When you stored the receipts on the USB then what you doing with that USB?	<b>Answer:</b> After was store all daily purchases receipt on USB, then IT employee his responsibility to take the USB and send it by email to the base at Spain company so and the one who is responsible to open and close the references of the system by the base on Spain so when we add receipt and we need to make change to this receipt we can't do it without the permission from the base on Spain company
<b>Question #3</b> Do you use barcode/QR on your system process?	<b>Answer:</b>

	We use RFID reader on each item that provide the items details and barcode that displayed on the receipt will be used on refund/exchange the items.
<b>Question #4</b>  Do you have an update for every purchasing, refunding, exchanging item?	<b>Answer:</b> No update on the system, its will just use the paper receipt and check to the item the has been refunded.
<b>Question #5:</b>  what is the information you take it from customer when refunding/exchanging an item?	<b>Answer:</b> Yes, there Phone numbers and Signature

## Appendix B - Questionnaire



### My Receipts

My Receipts is our senior project that will make the process of shopping experience easy  
Help us to provide more reliable and smart Experience.

\* Required

Q1: How often do you return the item you purchased from the shop (example: shirt)? \*

always  
 never  
 rarely

Q2: How many times do you go to shopping in a month ? \*

Once a month  
 twice a month  
 more than 3 times

Q3: Have you ever been in a situation where you wanted to refund/exchange but didn't find the receipt? \*

Yes  
 No

Q4: Would you prefer using electronic receipts over paper receipts? \*

Yes that's fabulous idea  
 No i am good with paper receipt

Q5: Do you think that electronic receipts will solve the problem of forgetting and losing paper receipts? \*

Yes  
 No

Q6: Have you ever reached the end of the month with no money left for further purchases? \*

Yes  
 No

Q7: do you have any suggestions to improve our application ?

Your answer \_\_\_\_\_

Submit

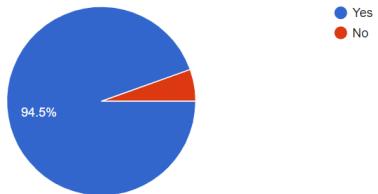
## Appendix C - Questionnaire Result



Questions Responses 110

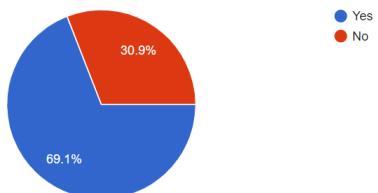
Q5: Do you think that electronic receipts will solve the problem of forgetting and losing paper receipts?

110 responses



Q6: Have you ever reached the end of the month with no money left for further purchases?

110 responses



## **Appendix D - Members Contribution**

<b>Chapter</b>	<b>Team Members Contribution</b>
1: General Introduction	
2: Literature Review	
3: Elicitation	
4: Analysis Model	All team members have worked together in writing, designing the diagrams, and gathering data for each chapter in the project.
5: Design Model	