

**MOBILE BASED FOR INVENTORY  
SYSTEM (MBIS)**

**CHE NORMADIAA BINTI IBRAHIM**

**Bachelor of Software Engineering with Honors**

**UNIVERSITI MALAYSIA PAHANG**

**UNIVERSITI MALAYSIA PAHANG****DECLARATION OF THESIS AND COPYRIGHT**

Author's Full Name : CHE NORMADIAA BINTI IBRAHIM

Date of Birth : 15 AUGUST 1996

Title : MOBILE BASED FOR INVENTORY SYSTEM (MBIS)

---

---

Academic Session : SEM II 2017/2018

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)\*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

---

(Student's Signature)

960815-11-5078

---

New IC/Passport Number

---

(Supervisor's Signature)

FAUZIAH BINTI ZAINUDDIN

---

Name of Supervisor



## **SUPERVISOR'S DECLARATION**

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Software Engineering with Honors.

---

(Supervisor's Signature)

Full Name : FAUZIAH BINTI ZAINUDDIN

Position : LECTURER

Date :



## STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at University Malaysia Pahang or any other institutions.

---

(Student's Signature)

Full Name : CHE NORMADIAA BINTI IBRAHIM

ID Number : CB15070

Date :

MOBILE BASED FOR INVENTORY SYSTEM (MBIS)

CHE NORMADIAA BINTI IBRAHIM

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Bachelor of Software Engineering with Honors

Faculty of Computer System and Software Engineering

UNIVERSITI MALAYSIA PAHANG

MAY 2018

## **ACKNOWLEDGEMENTS**

First and foremost, praise to Almighty Allah for all his blessing for giving me patience and good health throughout the duration of this project, Mobile Based for Inventory System. I would like to dedicate my highest gratitude for those who have involved directly or indirectly during this project. A million thanks to Madam Fauziah Binti Zainuddin, my supervisor in helpful and supporting me in prepared the document, supervision, suggestion, ideas and all the given guidelines for me to complete the project successfully. Do not forget also to my parents, family and friends, sincere thanks for their affectionate support, valuable information and advices in order to help from start to finish preparation of the report and develop the system. Thank you very much to supporter and helper that give me encouragement to continue with this project. May Allah bless al of you.

## ABSTRAK

Tajuk projek ini adalah Aplikasi Mobil untuk Sistem Inventori (MBIS). Sistem ini memainkan peranan penting untuk membantu penjual terutama stokis untuk menguruskan inventori mereka, masuk dan keluar produk dan menjejak stok produk yang tersedia. Sebelum ini, kebanyakan penjual merekodkan inventori mereka hanya dalam buku log sahaja. Dengan cara ini, kebarangkalian kehilangan data mungkin berlaku dan terus perkara ini dapat mempengaruhi inventori mereka. Oleh itu, objektif untuk membangunkan sistem ini adalah untuk menganalisis masalah sistem semasa dengan menghasilkan Aplikasi Mobil untuk Sistem Inventri, untuk merekabentuk dan membangunkan sistem prototaip kepada Aplikasi Mobil untuk Sistem Inventori menggunakan Google Excel sebagai pengurusan pangkalan data sistem dan untuk menguji aliran prototaip Aplikasi Mobil untuk Sistem Inventori. MBIS dicipta menggunakan Android Studio dengan Bahasa program iaitu Java. Di samping itu, pengkomputeran awan akan digunakan untuk menyokong pengurusan pangkalan data sistem dan membuat sistem lebih mesra. Metodologi yang digunakan dalam membangunkan sistem ini ialah Metodologi Agile yang terdiri daripada lima fasa iaitu merancang, kehendak, reka bentuk, membangunkan dan ujian yang membolehkan sistem dibangunkan dengan cepat dan mudah melaksanakan perubahan walaupun semasa pembangunan sistem. Sebelum menyampaikan sistem, ujian penuh ke atas sistem akan dilakukan untuk memastikan sistem memenuhi keperluan pengguna dan objektif.

## **ABSTRACT**

The title of this project is Mobile Based for Inventory System (MBIS). This system plays an important role to help seller especially stockist to manage their inventory, inbound and outbound of the product and track the availability of product. Before this, most of the seller recorded their inventory just in log book only. With this way, the probability loses of data may happen and directly this matter can effected their inventory. So, the objectives of developing this system is to analyse the problem of current system by produce Mobile Based for Inventory System, to design and develop prototype system for Mobile Based for Inventory System using Google Excel as a system database management and to test the prototype of Mobile Based for Inventory System. MBIS is developed by using Android Studio with Java language. In addition, cloud computing will be used to support a system database management and make a system more friendly. The methodology used in developing this system is Agile Methodology that consist of five phases which are planning, requirement, design, building and testing that allows the system to be developed quickly and easily implement the changes even during the development of the system. Before deliver the system, full testing toward the system will be performed to ensure the system has meet the user requirements and objectives.



**TABLE OF CONTENT**

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF SYMBOLS</b>	<b>x</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xi</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>17</b>
<b>1.1 BACKGROUND OF STUDY</b>	<b>17</b>
<b>1.2 PROBLEM STATEMENT</b>	<b>18</b>
<b>1.3 OBJECTIVE</b>	<b>19</b>
<b>1.4 SCOPE</b>	<b>19</b>
<b>1.5 THESIS ORGANIZATION</b>	<b>20</b>
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>21</b>
<b>2.1 INTRODUCTION</b>	<b>21</b>
<b>2.2 EXISTING SYSTEM</b>	<b>21</b>
2.2.1 MY BUSINESS	22
2.2.2 SMART INVENTORY MANAGEMENT	23

2.2.3 SMART INVENTORY SYSTEM	28
<b>2.3 COMPARING EXISTING SYSTEM</b>	<b>33</b>
<b>CHAPTER 3 METHODOLOGY</b>	<b>38</b>
<b>3.1 INTRODUCTION</b>	<b>38</b>
<b>3.2 SOFTWARE DEVELOPMENT METHODOLOGY</b>	<b>38</b>
3.2.1 Context Diagram	41
3.2.2 Use Case Diagram	42
3.2.3 Overall System Activity Diagram	43
3.2.4 Class Diagram	45
3.2.5 Entity Relationship Diagram (ERD)	44
3.2.6 Propose System Interface	46
3.2.7 Modules / Functions of the System	47
3.2.8 Software Requirement Specification (SRS)	48
3.2.9 Software Design Document (SDD)	48
<b>3.3 HARDWARE AND SOFTWARE</b>	<b>49</b>
3.3.1 Software Items	49
3.3.2 Hardware Items	50
<b>3.4 GANTT CHART</b>	<b>45</b>
<b>CHAPTER 4 IMPLEMENTATION, TESTING AND RESULT DISCUSSION</b>	<b>51</b>
<b>4.1 INTRODUCTION</b>	<b>51</b>
<b>4.2 IMPLEMENTATION</b>	<b>51</b>
4.2.1 Development Environment	52

4.2.2 System Functionality	58
<b>4.3 TESTING AND RESULT DISCUSSION</b>	<b>70</b>
4.3.1 Functional Testing	70
4.3.2 User Acceptance Testing	76
<b>4.4 USER MANUAL</b>	<b>76</b>
<b>CHAPTER 5 CONCLUSION</b>	<b>77</b>
<b>5.1 INTRODUCTION</b>	<b>77</b>
<b>5.2 PROJECT CONSTRAINTS</b>	<b>79</b>
<b>5.3 FUTURE WORK</b>	<b>79</b>
<b>REFERENCES</b>	<b>80</b>
<b>APPENDIX A</b>	
<b>APPENDIX B</b>	
<b>APPENDIX C</b>	
<b>APPENDIX D</b>	
<b>APPENDIX E</b>	

**LIST OF TABLES**

Table 2.1 Advantages and disadvantages for My Business	24
Table 2.2 Advantages and disadvantages for Smart Inventory Management	27
Table 2.3 Advantages and disadvantages for Smart Inventory System	31
Table 2.4 Comparison between three (3) existing system	33
Table 3.1 Software Items	49
Table 3.2 Hardware Items	50
Table 4.1 Test Cases in Functional Testing	70

**LIST OF FIGURES**

Figure 2.1 Main Interface for My Business system	22
Figure 2.2 Main interface of Smart Inventory Management	25
Figure 2.3 Home page for Smart Inventory System	29
Figure 2.4 Scanner mode page	30
Figure 2.5 Notification of item lower than critical leve	31
Figure 3.1 Illustration of agile development model	39
Figure 3.2 Context diagram for Mobile Based for Inventory System	41
Figure 3.3 : Use case diagram of Mobile Based for Inventory System	42
Figure 3.4 Overall System Activity Diagram of Mobile Based for Inventory System	43
Figure 3.5 Class diagram of Mobile based for Inventory System	44
Figure 3.6 Entity Relationship Diagram (ERD) of Mobile Based for Inventory System	45
Figure 3.7 Dialog Diagram for Mobile Based for Inventory System	46
Figure 4.1 Android Studio Environment Interface for Design Layout	51
Figure 4.2 Android Studio Environment Interface for Text Layout	52
Figure 4.3 Android Studio Environment Interface for Java Layout	53
Figure 4.5 Google Excel Environment Interface	54
Figure 4.6 Google Apps Script Environment Interface	55
Figure 4.7 Product Configuration Interface	56
Figure 4.8 Login Interface	57
Figure 4.9 Register Interface	58
Figure 4.10 Product Interface	59

Figure 4.11 Add Product Information Interface	59
Figure 4.12 Update and Delete Product Interface	60
Figure 4.13 Sale Interface	61
Figure 4.14 Add Sale Information Interface	61
Figure 4.15 Sale History Interface	62
Figure 4.16 Purchase Interface	63
Figure 4.17 Add Purchase Information Interface	63
Figure 4.18 Purchase History Interface	64
Figure 4.19 Customer Interface	65
Figure 4.20 Add Customer Information Interface	65
Figure 4.21 Update and Delete Interface	66
Figure 4.22 Supplier Interface	67
Figure 4.23 Add Supplier Information Interface	67
Figure 4.24 Update and Delete Interface	68

**LIST OF APPENDIX**

Appendix A	Gantt Chart
Appendix B	Software Requirement Specification (SRS)
Appendix C	Software Design Document (SDD)

**LIST OF ABBREVIATIONS**

MBIS	Mobile Based for Inventory System
SRS	Software Requirement Specifications
SDD	Software Design Document
iOS	iPhone OS
PC	Personal Computer
CSV	Comma Separated Values
SDLC	System Development Life Cycle
GUI	Graphical User Interface



## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 BACKGROUND OF STUDY**

In the era of technology, business is a competitive and creative activity that continuously contributes to the shaping of society and plays a major role. Business are not only important because provide goods services for consumers, but it also the ways of the companies conducting the business. Before this, most of the companies or sellers used manual book to manage or record the sales of their business. Therefore, Mobile Based for Inventory System is new approach of the system application that will be developed to help the stockist to manage their business more efficient and effective in the systematic ways. For examples can track the costs of the inventory throughout the manufacture and sales process, tell the seller when to add stock, allow to track profits and used to forecast inventory levels.

As we known, inventory management system are central on how companies or seller track and control the inventories. It is the process of overseeing and controlling the flow of inventory units a business uses in the manufacture of good for sale or distribution. The different types of the inventory have different function in terms of the input, process, output and the storage. In this project, the focus will be on implementing and developing the inventory system by using Google Excel as system database management where it is one of the medium that provided high-level services where separates a physical computing devices into one or more virtual devices that can make it easier to used and manage to perform computing task everywhere and anytime. To access the system database, the user need to active the internet connection because it is secure and private

network which is the user need to have the active google account to access the Google Excel to prevent an unauthorized access. The proposed system can help seller to manage stock, store and retrieved data with efficient inventory control and high accuracy.

So, Mobile Based for Inventory System (MBIS) developed to help the stockist to manage and handling their business properly using Google Excel to save, retrieve, update, view the availability stock, manage sale, manage purchase and manage customer and supplier.

## **1.2 PROBLEM STATEMENT**

Today there are a lot of inventory systems that have been developed in order to help seller to manage their inventory or business. Mostly, of the existing system that seller used are web-based system which mean they only can access to the system by using laptop or computer only. Thus, it is not efficient to do the work at outside because they need to bring the laptop around. But, each system have their advantages and disadvantages itself to make the system working properly. So in this project, three existing application will be choose to study on how it working and the process. After studies this three systems, then will be analyse the advantages and disadvantages of the system that can be apply into the Mobile Based for Inventory System.

Besides that, several stores used manual forms of a log book to keep record of product available in stock and transaction made. In this situation, the seller facing some difficulty while handling inbound and outbound of the products and sometimes the stock is out of control and difficult to trace. Moreover, this method also has a high risk of data loss due to difficulty of making backup because all the data are in the form of hard copy. In order to solve this problem, Mobile Based for Inventory System will be develop to enhance buying and selling process with the customers and save more time in order to identify items and product existing because any unpredictable mistakes can increase cost losses to the companies.

Lastly, the process of inventory is difficult to track inbound, outbound and availability of the product using manual log book and tendency to gain error or mistake during calculation. It is difficult to the seller or user to trace and aware the availability of stock. In addition, the computerized system is more efficient and accurate in calculation such as calculate price of sale, purchase, profit, and loss and sometimes need to apply several formula to calculate all of that compare to human energy that have a lot of difficulties in calculate price.

### **1.3 OBJECTIVE**

The purpose in developing this Mobile Based for Inventory System (MBIS) is based on several objectives. Main objectives of this project are :

- i. To analyse the problem of current system by produce Mobile Based for Inventory System.
- ii. To design and develop prototype system for Mobile Based for Inventory System using Google Excel as a system database management.
- iii. To test the prototype of Mobile Based for Inventory System.

### **1.4 SCOPE**

In this section, scopes of Mobile Based for Inventory System (MBIS) are defined. There are several elements that involves which is user, system and database. These elements are important to make sure that the scope of the system not override the boundaries of the system. Only one person that involve in Mobile Based for Inventory System (MBIS) which is user named stockist. This system is mobile application that is easier to the user to access. In addition, this system will be developed using Android Studio for develop Java language. Then, Google Excel will be used to support a system database management and make a system more friendly to seller to check stock and view the history by getting the requirement from the client.

## **1.5 THESIS ORGANIZATION**

This thesis consists of 5 chapters and each chapter discuss different issue of project. Chapter 1 elaborates the introduction of the Mobile Based for Inventory System that is about the overall overview of the system. It include introduction, problem statement, objective, scope and thesis organization which is objective and scope will be defined based on problem statement. Chapter 2 discuss about the literature review which is supposed to elaborate and compare the current system and previous system. Meanwhile, methodology in chapter 3, where explained about the process, approach and method used to build a project. In the same time, introduction, Gantt chart, software development tools and module/function also explained in this chapter. Next, chapter 4 consist of implementation or development of the project. Lastly, chapter 5 summarize overall from chapter 1 to chapter 4 and give the conclusion about the system.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION**

In this chapter will describe the comparison of the existing system. There are three inventory system that already existing will be compare in this project which is My Business, Smart Inventory Management and Smart Inventory System. This chapter consist of three parts. First part of this chapter is a review of the inventory system. Three inventory system or existing system that will be review which are My Business, Smart Inventory Management and Smart Inventory System. After describe a review between three existing system, the advantages and disadvantages will come out in this part too where to enhance Mobile Based for Inventory System. Second part of this chapter is make the comparison between three existing system. Last part will be concerns on techniques, hardware and tools used within three existing inventory system.

#### **2.2 EXISTING SYSTEM**

Nowadays, many companies used the inventory management system to avoid the product out of stock and handling inbound and outbound of the product. The following are three existing system that can be differentiate based on feature that provided.

### 2.2.1 MY BUSINESS

My Business is one of the inventory system that have been used by sellers for small business administration where have a point of sales (Green, 2018) This application has an inventory section where seller can add product, update and check the products that are in minimum stock quantity. In addition, My Business also has a section to manage sales, purchases, revenue, extra expenses, and generate report and graphs. Then, seller can create receipts for sales and create sales receivables.

My Business inventory system is a mobile application. This system is offline control which is offline management system software that take care of the inventory such as sales orders, payments, buying the product and fulfilment orders efficiently and faster. Figure 2.1 shows main interface of My Business system.

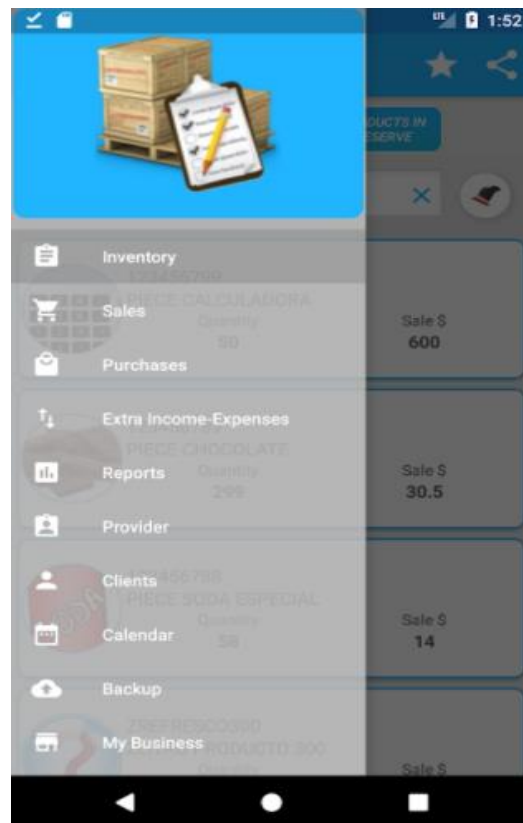


Figure 2.1 Main Interface for My Business system

On the whole, there are 4 modules in this system which are manage sales, purchases, revenue, and generate report and graphs. First module is manage sales. Before that, seller must add their product into the list in order to make them easier to manage sales for each product. They can add and search their product by inserting the barcode while using a barcode reader to get the product easily, same as in the sales. In the sales section, seller can add sales, check their sales history and add the quotation. For each sale, seller needs to insert the quantity and availability of the product, sale for each product and for all. In the same time, the user can set the customer payment whether paid or receivables in order to separate them in report and to make seller easier to track debtors. Then, every transaction occurring between customer and seller will be recorded in the sales history.

Second module is purchases. Once sales occurred, the availability of the items or product will decrease and sometimes out of stock. Because in this system does not apply alert system, so seller needs to be aware every time about the availability of the products. In the purchases section of My Business, the method and form used are the same as when the seller needs to add new items. But a little bit different, the seller can search the product that they want to add and the information is automatically filled, just seller needs to insert the quantity of the added product only. The seller also can check and review back their history of purchases.

Third module is manage revenue. To make it easier for sellers to know their profit and amount losses for the whole, My Business separated revenue into extra incomes and expenses. Example the extra incomes is investment, rental building and other while extra expenses like clothes, food gasoline and others. In this situation, seller can manage their budget and the amount of incomes and expenses also provided in the report to make the user easier to track and check the date, time and total of the transaction.

Lastly, generate report and graphs. There are many categories of report in My Business such as report of transaction per day, total transaction, sales receivables, purchases payable, extra income receivable, extra expenses payable, transaction by product, top product, product details, inventory report, sales per day, ABC analysis, purchase register, extra income and expenses, clients report, top clients and provider report. These all reports have their own differences although they come from the same function like extra incomes and expenses, extra income receivables and extra expenses payable where

extra income and expenses is the general report for the user get the information overall amount in certain time. To prevent the data loss, seller can backup it to the storage of the device used and also can sent to other devices.

There are the advantages and disadvantages for this system. This can be seen in Table 2.1.

Table 2.1 Advantages and disadvantages for My Business

<b>ADVANTAGES</b>	<b>DISADVANTAGES</b>
<ul style="list-style-type: none"> <li>➤ Can install this application in Android and formerly iPhone OS (iOS).</li> <li>➤ Can used the application without internet connection.</li> <li>➤ Can create a backup and store it in the device or send to another device.</li> <li>➤ Can keep track of buyer and receivables.</li> <li>➤ Possibility used barcode reader.</li> <li>➤ Can browse and use this application without signing up.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Inventory details are not secure.</li> <li>➤ Unable to view report more than 15 times.</li> <li>➤ The receipts can be save in pdf only cannot save in png. or jpeg.</li> <li>➤ Do not have alert for the minimum stock quantity.</li> </ul>

In conclusion, there are several features included in My Business inventory system such as the current version for this mobile application is 6.1.0 and type of android that can be used for this application is 4.4.3 and up. Next, the person can installs My Business up to 100,000. Meanwhile, My Business inventory system have the advantages which is this application can be used without internet connection and this system is friendly which is can install this application in Android and formerly iPhone OS (iOS). Then, this system has possibility used barcode scanner to add the products so that it is easier and save time. Others, able to track buyers and receivables, in addition the sellers



can create a backup of the report and store it in the device or send to another device. This features can be apply in mobile based Inventory System but to make it more safe, the report also can save to the cloud computing. Contradict than that, this application did not send any alert for the minimum stock quantity or the products is out of stock so that sellers cannot know if there have products out of stock. My Business only can save the receipt in pdf only cannot save in picture such as jpeg or png.

### 2.2.2 SMART INVENTORY MANAGEMENT

Smart Inventory Management is suitable used for any type of small, medium business and warehouse inventory management (Shana, 2018). In addition, this system is simple and easy to manage inventory for android devices and web based solution. But in this comparison, inventory for android devices used. Figure 2.2 is the main interface of the Smart Inventory Management by using android device.

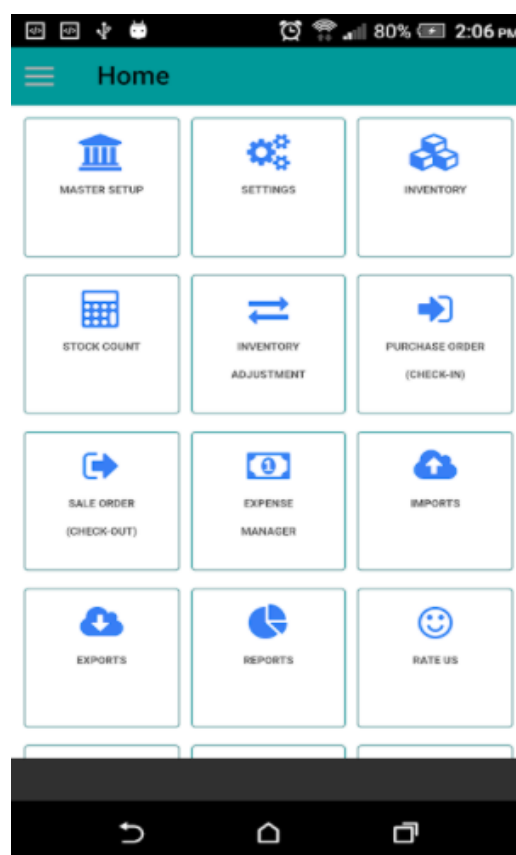


Figure 2.2 Main interface of Smart Inventory Management

Figure 2.2 shows the main interface of Smart Inventory Management system. Smart Inventory Management is a business management system that take care of seller inventory such as inventory tracking, stock take or control, inventory adjustment, purchase or Sale orders and transfers.

The module of inventory tracking focus more on viewing, adding, and editing an inventory which is this system the user can add the product based on the their own categories. For example, product Mary Kay have many product from the different categories such as skin care, lipstick, makeup and etc. Other than that, to insert the barcode one by one is difficult and waste the time, moreover if the user need to add more new item. In this matter, barcode scanner can be used to make it easier to the user. As usual the user need to insert the quantity of stock, unit cost and unit price in each item want to add.

Besides, stock take or control where in the Smart Inventory named as the stock count. In this system, user can count the product inbound and outbound easily by adjust the inventory count which is in this section provided the status name, outer and inner unit and the. Before that, user need to search the specific products that want to adjust. This way is easier to the user entered and get the product out without do the same step for multiple products.

Next module is inventory management. In inventory management, vendors or seller only need to choose the product that existing in the system and specific the quantity of each product to be managed as the product is can be any states such as waste, damage, returned, expired, lost, sample or in promo. In addition the seller also can choose any date to add it in any states.

Other than that, purchase and sales where seller can select the product want to purchase and also can select the supplier they want. In order to purchase, seller need specified required date of purchase product to make sure the product is not out of stock. In the meantime, to ensure sales are always recorded and have complete data, the section of sales order is needed where the seller needs to enter the name of the customer who made the purchase, as well as the quantity. If the purchase has a discount, the seller can place the discount given and the net price for the purchase will come out automatically.

Then the seller can also choose the payment method whether using Check, Visa, Master, Amex, UnionPay, Diners, Debit, Discover, JCB, Cash and pending payment.

Lastly, sellers can manage their inventory by transfer their inventory between stores and warehouses. Sellers need to create a new account for each transferable stores or warehouses before transfer the items. In the same time, need to make sure the item code for each products are same in each stores or warehouses to transfer with add new item. Then, if the system is ready to transfer, there has one button on top tap for the user to add new received transfer. However, after the process done, seller can review and check back the history that have been transfer by select the account and date range and click search.

There are the advantages and disadvantages for this system. This can be seen in Table 2.2.

Table 0.1 Advantages and disadvantages for Smart Inventory Management

ADVANTAGES	DISADVANTAGES
<ul style="list-style-type: none"> <li>➤ Do not required additional hardware or software.</li> <li>➤ Used cloud based and can access via any android device.</li> <li>➤ Can share the exports, purchase order, sale order using social networking and via email.</li> <li>➤ Have web version which can access from PC.</li> <li>➤ Can classified the item into its own categories.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Cannot access the system if have no internet connection.</li> <li>➤ This system need to register first then login to proceed to the next step.</li> <li>➤ Cannot view the daily sales in a single report.</li> <li>➤ Cannot delete pictures after added.</li> </ul>

In a nutshell, there are some advantages that really interesting to been develop in Mobile Based Inventory System. First, Smart Inventory Management app is cloud based

system where no backup required and can be access via any android devices. That make the user no need to worry of losing data. Second, this system is an online application, so developer provided web version for user get the clear application which can be access in their own personal computer, in addition can access the system anytime and anywhere. Third, seller can export the report or data in the CSV format and can share the report with anyone or even store in the Google drive. Lastly, in form add new item, user can classified the item that will be added into its own categories.

There also have some disadvantages in this system that unusually happen in other system or application like seller or user cannot view the daily sales in a single report. This matter will make more complicated to the user if anything happen which user need to open full report and review one by one all of the sales to get daily sales.

### **2.2.3 SMART INVENTORY SYSTEM**

Smart Inventory System is developed in mobile application platform which is one of an application that function to keep track of the product easily where using the QR Codes and barcode. This system is different from other system which is the divide the inventory into three part that is items, groups, and tags. For the items, this part is contrast to others where the user need to enter the quantity for each item want to add. In addition, the user also can see the quantity changes from the detail page of the items. Then, in the group, the items are collected in together and specified it in one group followed their categories. Last one is tag where it is used for specified item or group attribute or alternative to group. For example; old, new, important and etc. In tag do not have to insert the quantity information same as group. Figure 2.3 shown home interface of Smart Inventory System.

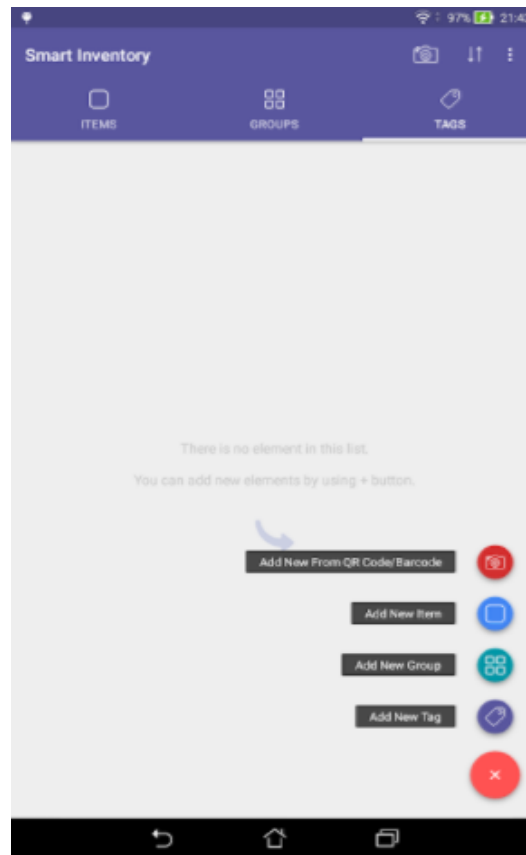


Figure 2.3 Home page for Smart Inventory System

Next, Smart Inventory System can be classified into four modules that are registration, manage inventory, purchase and sales order and summary information. First module is registration. After Smart Inventory System has been download in the devices, automatic every user will be the basic user. So, the function of registration is not for basic user but only for premium user where the basic user need to buy for RM9.99 to get the premium user. The specials of premium user has been stated as :

- i. Premium user get unlimited cloud storage where they can backup all data in the device to the cloud without having any limitation.
- ii. Summary and detailed information about the products can be visible.
- iii. Quantity change chart and object history can be visible.
- iv. Premium user can export all lists in the format of CSV.
- v. Premium user can see and modify all inventory information and data by using web page.

Second module is manage inventory which is in this section the user can add their item depends on which part they want to at whether in items, groups and tags. To add the products into the item, user can used easier way with scan the QR Code or barcode and also can add manually with insert QR Code or barcode value, name, description, quantity, and critical quantity level. Other than that, user can classified the new item that want to add follow the each category and can review or search in the part of groups. Then, user also can add the new tag where user can review their item.

Third module is purchase and sales order. This section user just only use scanner mode where they just need to scan the barcode or QR code if want make purchases or sales. Before scan, the user need to specify the amount of change as Figure 2.4. For example, if user choose to make sales and the ready stock of that item is 20, then user specified their amount of change is 2. When user scan of the item, automatically the stock of item will decrease to 18. If the item is lower than critical level, the system will sent the notification of pop up as Figure 2.5.

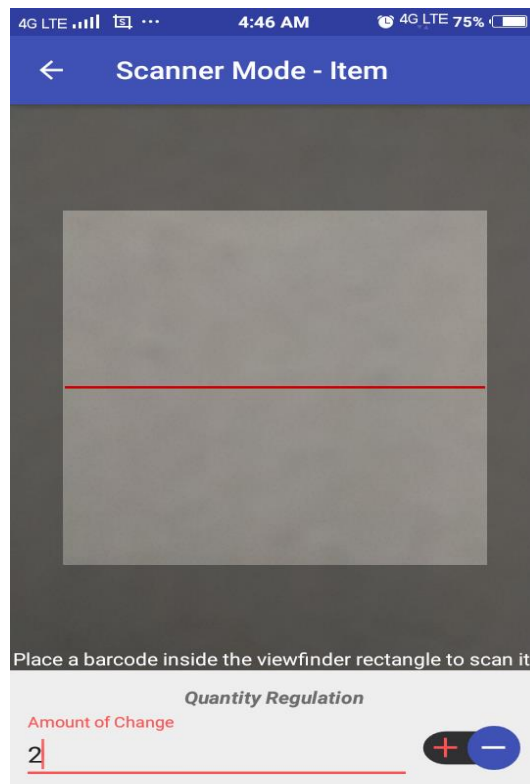


Figure 2.4 Scanner mode page



Figure 2.5 Notification of item lower than critical level

Last module is summary information where in this part consists of pie chart summary, total object summary, item specific summary, group specific summary and tag specific summary where the user can know all the related information of the Smart Inventory System. Basic user only can see the pie chart summary and others can be access by premium user.

There are the advantages and disadvantages for this system. This can be seen in Table 2.3.

Table 2.3 Advantages and disadvantages for Smart Inventory System

ADVANTAGES	DISADVANTAGES
<ul style="list-style-type: none"> <li>➤ Do not required additional hardware or software.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Cannot access in PC and iPhone OS (iOS).</li> </ul>

<ul style="list-style-type: none"> <li>➤ Cannot access the system if have no internet connection.</li> <li>➤ System backup data to the cloud.</li> <li>➤ Easier to transfer present list to the application by using import and export.</li> <li>➤ Can find the item easily by added the item to favourite.</li> <li>➤ Premium user get unlimited cloud storage.</li> <li>➤ Sent notification if the quantity of item lower that critical level.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Cannot store the picture of item on the cloud.</li> <li>➤ Only premium user can export data and view the object history.</li> </ul>
---	--

From the differentiation of the advantages and disadvantages from Table 2.3 above can be conclude that this system is a mobile application platform that can be access and install in Android only cannot in PC or iPhone OS (iOS). Smart Inventory System is platform where the user can keep track product easily, in addition can keep track the product fast by added product to favourite list. By registering to the system, the status of the user will change from basic user to the premium user where the premium user can backup the data to the cloud, get unlimited cloud storage, export data and view object history while basic user just can track and store the items in system only. However, just data can stored, the pictures of item cannot stored in the cloud. Then, the advantage seems clearly in this system is the system sent the notification to the user if the level of item quantity is lower than critical level. Not all user noticed the items is in minimum quantity if a lots of item added. By having this notification, the selling will be more productive and the item can be restock constantly.



### 2.3 COMPARING EXISTING SYSTEM

Table 2.4 shows the comparison between three existing system which are My Business, Smart Inventory Management and Smart Inventory System. In this table are divide into several aspects such as current version, requires android, installs, platform, features, advantages and disadvantages.

Table 2.4 Comparison between three (3) existing system

	<b>MY BUSINESS</b>	<b>SMART INVENTORY MANAGEMENT</b>	<b>SMART INVENTORY SYSTEM</b>
Current Version	6.1.0	2.5.1	1.2.2.2
Installs	4.4.3 and up	4.1 and up	4.1 and up
Size	100,000 and up	10,000 and up	10,000 and up
Updated	11 February 2018	17 February 2018	7 April 2018
Platform	Mobile application platform	Mobile application platform	Mobile application platform

Features	<ul style="list-style-type: none"> <li>➤ Can manage products, sales, purchases, revenue, and extra expenses.</li> <li>➤ Can generate reports and graphs.</li> <li>➤ Can create receipts for sales.</li> <li>➤ Can modify the numeric format in Configuration section.</li> <li>➤ Can register client name.</li> <li>➤ Can export in pdf format for products in reserve.</li> <li>➤ Providers and clients can be exported in CSV format.</li> <li>➤ Able to create sales receivables.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Can add unlimited items into the system</li> <li>➤ Multi user access.</li> <li>➤ Multi language support.</li> <li>➤ Send alert when level of stock is minimum.</li> <li>➤ Send alert for pending purchase order.</li> <li>➤ Quotation management.</li> <li>➤ Barcode scan integrated.</li> <li>➤ Consist graphical reports.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Easy to keep track product.</li> <li>➤ Can used QR codes and barcodes to search an item</li> <li>➤ Need to register via Google, Email, Facebook and Twitter to get premium user.</li> <li>➤ No limitation for details.</li> <li>➤ Can add objects to favourite list</li> </ul>
Advantages	<ul style="list-style-type: none"> <li>➤ Can install this</li> </ul>	<ul style="list-style-type: none"> <li>➤ Do not required</li> </ul>	<ul style="list-style-type: none"> <li>➤ Do not required</li> </ul>

	<p>application in Android and formerly iPhone OS (iOS).</p> <ul style="list-style-type: none"> <li>➤ Can used the application without internet connection.</li> <li>➤ Can create a backup and store it in the device or send to another device.</li> <li>➤ Can keep track of buyer and receivables.</li> <li>➤ Possibility used barcode reader.</li> <li>➤ Can browse and use this application without signing up.</li> </ul>	<p>additional hardware or software.</p> <ul style="list-style-type: none"> <li>➤ Can share the exports, purchase order, sale order using social networking and via email.</li> <li>➤ Have web version which can access from PC.</li> <li>➤ Can classified the item into its own categories.</li> </ul>	<p>additional hardware or software.</p> <ul style="list-style-type: none"> <li>➤ Cannot access the system if have no internet connection.</li> <li>➤ System backup data to the cloud.</li> <li>➤ Easier to transfer present list to the application by using import and export.</li> <li>➤ Can find the item easily by added the item to favourite.</li> <li>➤ Premium user get unlimited cloud storage.</li> <li>➤ Sent notification</li> </ul>
--	---	--	--

			if the quantity of item lower than critical level.
Disadvantages	<ul style="list-style-type: none"> <li>➤ Inventory details are not secure.</li> <li>➤ Unable to view report more than 15 times.</li> <li>➤ The receipts can be save in pdf only cannot save in png. or jpeg.</li> <li>➤ Do not have alert for the minimum stock quantity.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Cannot access the system if have no internet connection.</li> <li>➤ This system need to register first then login to proceed to the next step.</li> <li>➤ Cannot view the daily sales in a single report.</li> <li>➤ Cannot delete pictures after added.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Cannot access in PC and iPhone OS (iOS).</li> <li>➤ Cannot store the picture of item on the cloud.</li> <li>➤ Only premium user can export data and view the object history.</li> </ul>

In brief, table 2.4 is a comparison between the three online inventory system that have existed based on current version, requires android, installs, size, updated system date, platform, features, advantages and disadvantages. These all three system have their own weakness and strengths. A well-run system helps user understand their assets and maximize their potential, thus improving their business operations and increasing profits. Based on the three existing system mentioned, database management system is important in order to manage data efficiently and allows users to perform multiple tasks with ease. Besides that, with implemented this project by using Google Excel as database, data protection is taken importantly. With the existence of cloud database, the user don't have to worry about losing any data or document anymore because the system maintains an automatic paper trail of all transaction. In addition, by using this database, the user also can check their business history.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 INTRODUCTION**

In this chapter will briefly describe about the methodology used during the development of Mobile Base for Inventory System. To ensure an effectiveness of inventory management in order to develop information system and act as guidelines to satisfy the business's objectives, this project will be develop according to System Development Life Cycle (SDLC).

#### **3.2 SOFTWARE DEVELOPMENT METHODOLOGY**

The Agile Software Development Methodology is suitable method to be used in development of the system. The Agile methodology is the process of changeable requirement based on customer needs because this method is an iterative and incremental based development in which customer satisfaction is a high priority (Hneif & Ow,2009). It means at the end of development, customer will provide the feedback to ensure the project is successful and it can be implement without any problems. Moreover, agile development absolutely the best choice for mobile development so that the resulting mobile apps are adaptable after release (Sharma, Sarkar & Gupta, 2012). Figure 3.1 below shows an illustration of agile development model.

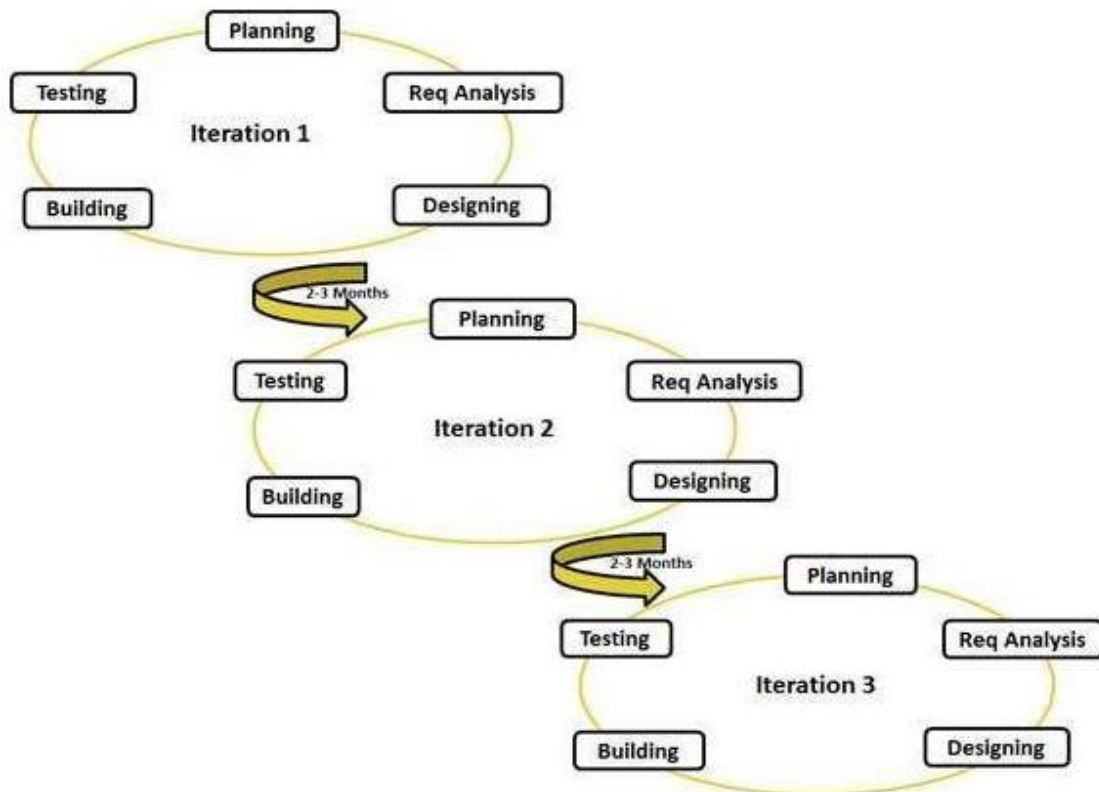


Figure 3.1 Illustration of agile development model (Bowerman, 2015)

In the process of Agile methodology, there are several stages that can be follow to achieve goal of the project which are planning, requirement analysis, design, coding or building and lastly is testing. All of these stages will be conducting based in iterative development that means customer can change their requirements in every step of the development. In addition, incremental development will be applied in this project when requirement change at the end of development.

i. Planning Phase

Planning phase is a first stage in Agile development which is the collection of user's requirements happen. The development will observe the common idea for client value, company value and risk that might occur in the future time in terms of resources, time, benefits, costs and others. Besides, in this phase also need to find out the scope of the problem and determine the solutions.

## ii. Requirement Analysis Phase

In the second phase, the developer need to analyzing the needs of the customers to make sure the system can meet their requirement. To meet customer expectations, developer need to show all possible requirement such as functional requirement and non-functional requirement. A functional requirement described what a system need to do while non-functional requirement place constrains on how the system will do.

## iii. Design Phase

In the third phase will translate all the requirement specification of the system into a design that will be implement .The hardware and software, system modules and graphical user interface (GUI) are defined for development of prototype for the proposed system. This step will help developer to illustrate the system to the customer in simple way and easy to understand. Normally, more than one approach is proposed in this stage such as use case diagram, context diagram, activity diagram, entity relationship diagram (ERD) and dialog diagram as shown in figure 3.1, 3.2, 3.3, 3.4, 3.5, and 3.6.

## iv. Building Phases

In fourth phases will start coding according to the requirements and the design discussed in previous phases. Then, the database will be created to stall the information of every transaction happen and created necessary interfaces and GUI to interact with the back-end.

## v. Testing

In the last phase, the full testing toward the system will be performed before the system is delivered to customers. This is to ensure the system has meet the user's requirements and objectives as well as to ensure that the system is free from error and bug to what was documented in the requirements analysis phase.



### 3.2.1 Context Diagram

Context diagram shows the interaction between Mobile Based for Inventory system with the user which is stockist. Stockist should register and login to the system in order to manage the inventory, manage sales, manage purchases, manage customer and manage supplier. Figure 3.2 shows the context diagram for Mobile Based for Inventory System.

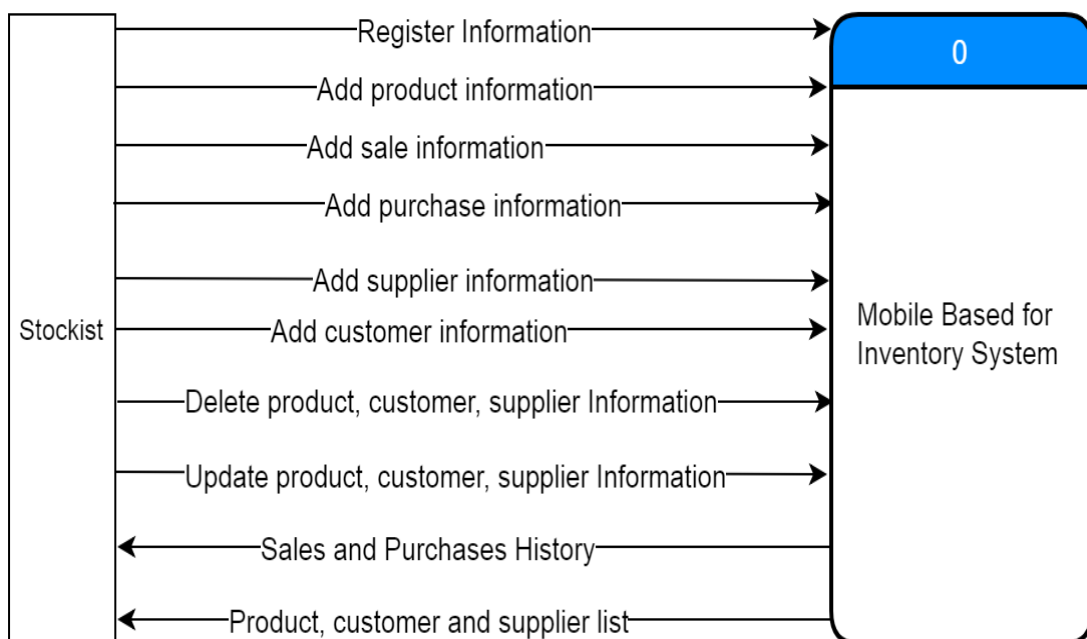


Figure 3.2 Context diagram for Mobile Based for Inventory System

### 3.2.2 Use Case Diagram

Use case diagram is used to describe functionality and the behaviour of system which show a group of use cases and actors and their relationship. Mobile Based for Inventory System consists of one actor, which is stockist that allow to manage inventory, manage sales, manage purchases, manage customer and manage supplier. Figure 3.3 is use case diagram for Mobile Based for Inventory System.

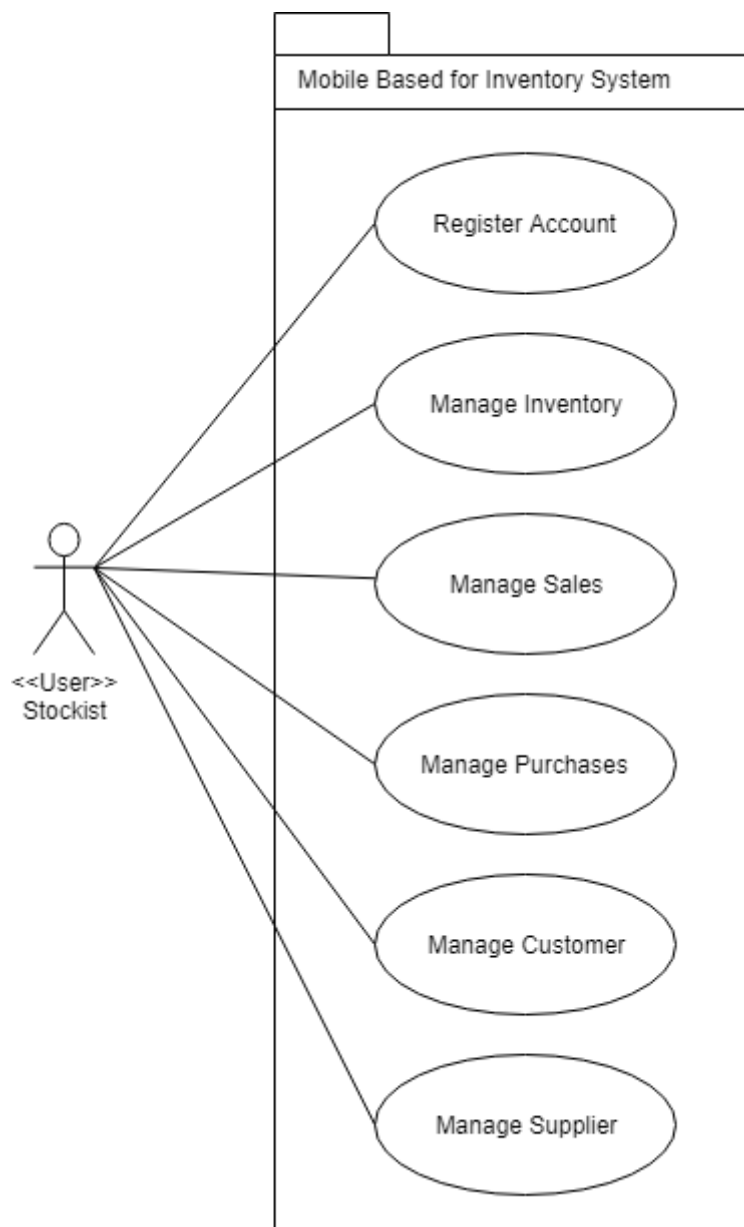


Figure 3.3 : Use case diagram of Mobile Based for Inventory System

### 3.2.3 Overall System Activity Diagram

Activity Diagram is significant UML diagram which explains the flow of the system that can connect between one activity of system to another activity. Figure 3.4 shows the overall system activity diagram of Mobile Based for Inventory System.

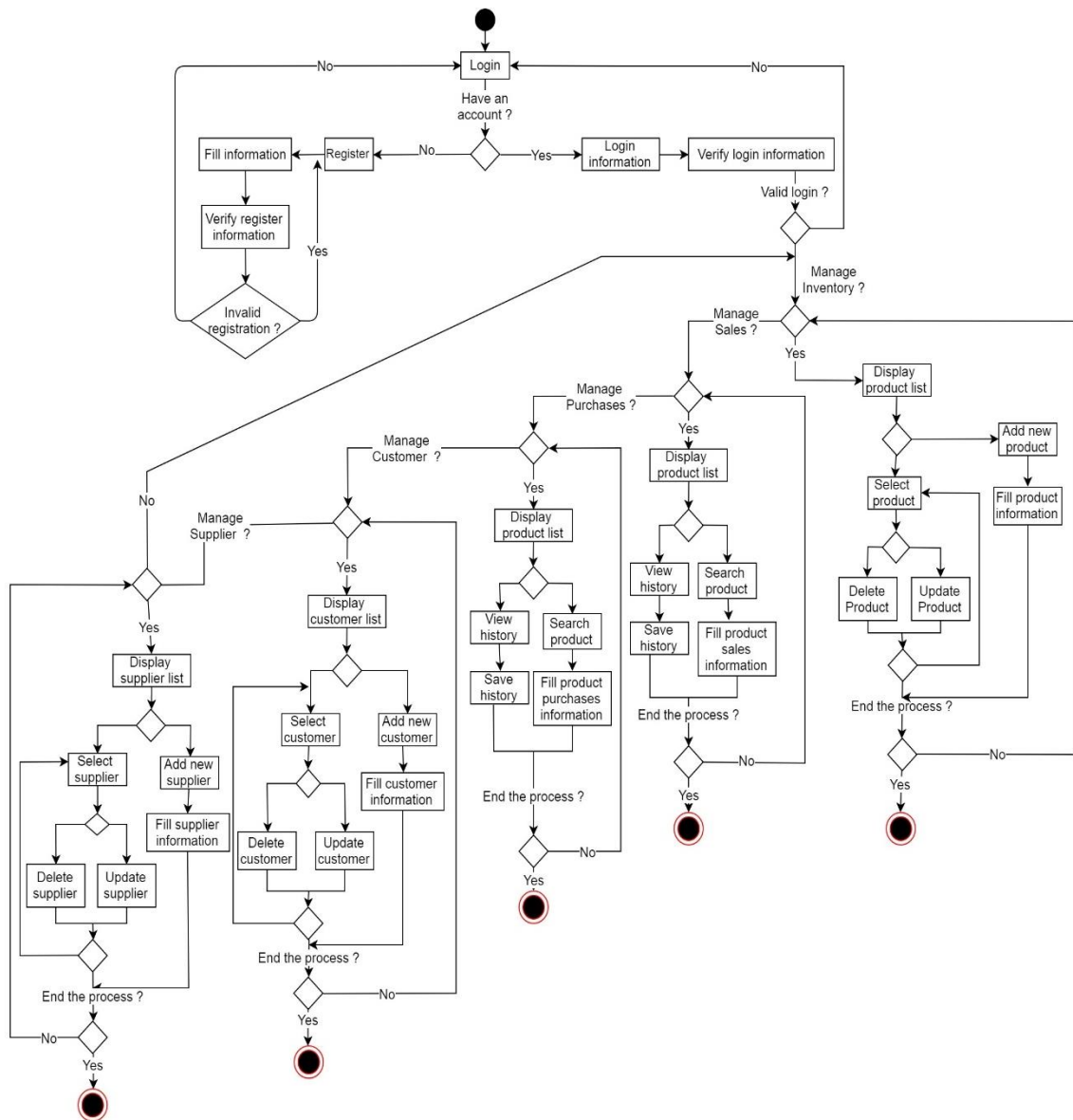


Figure 3.4 Overall System Activity Diagram of Mobile Based for Inventory System

### 3.2.4 Class Diagram

Figure 3.5 shown the class diagram of Mobile based for Inventory System.

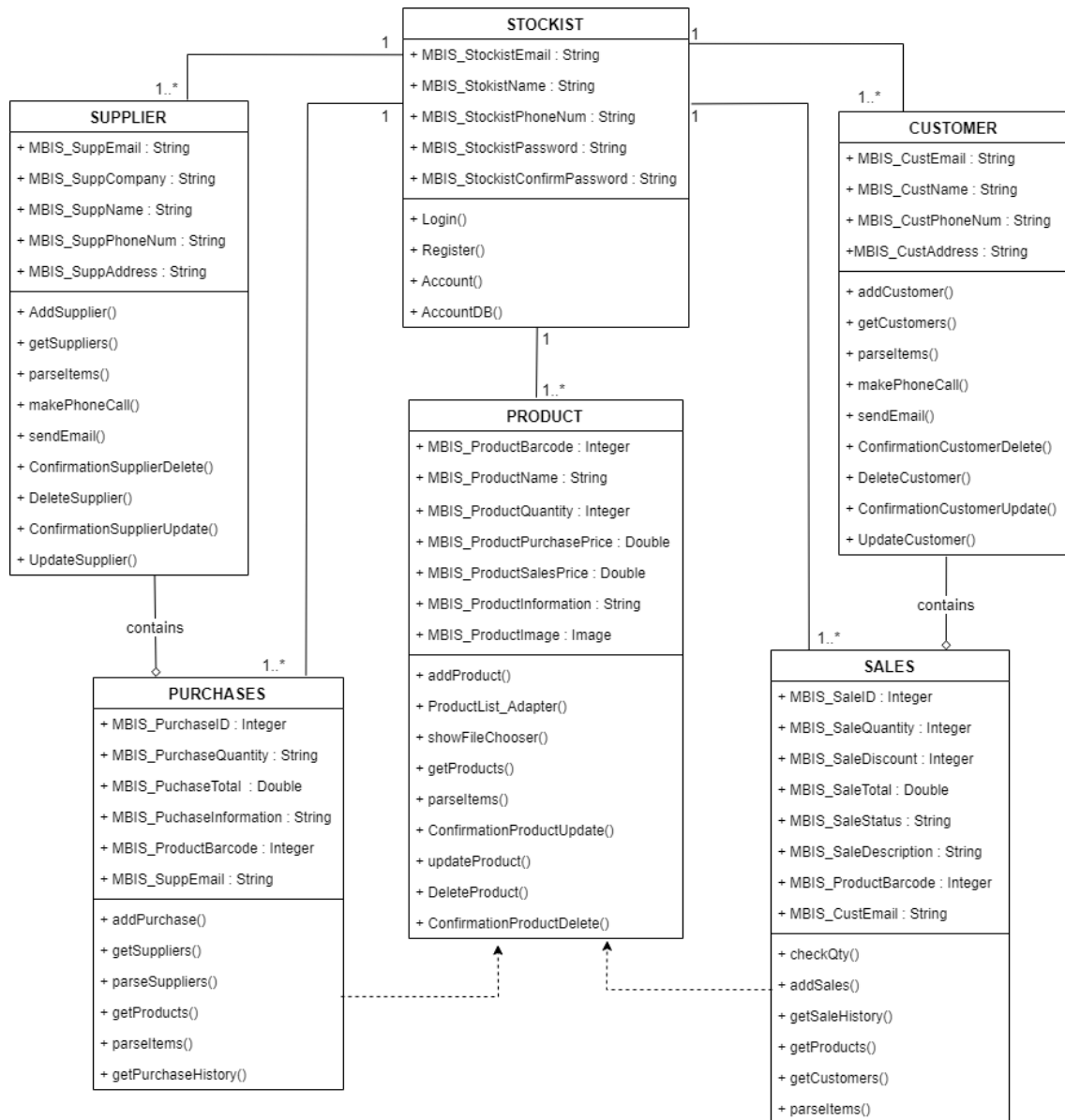


Figure 3.5 Class diagram of Mobile based for Inventory System

### 3.2.5 Entity Relationship Diagram (ERD)

Entity relationship diagram (ERD) is a flowchart that represent the relationship between database entities. In figure 3.6 shown entity relationship diagram of Mobile Based for Inventory System.

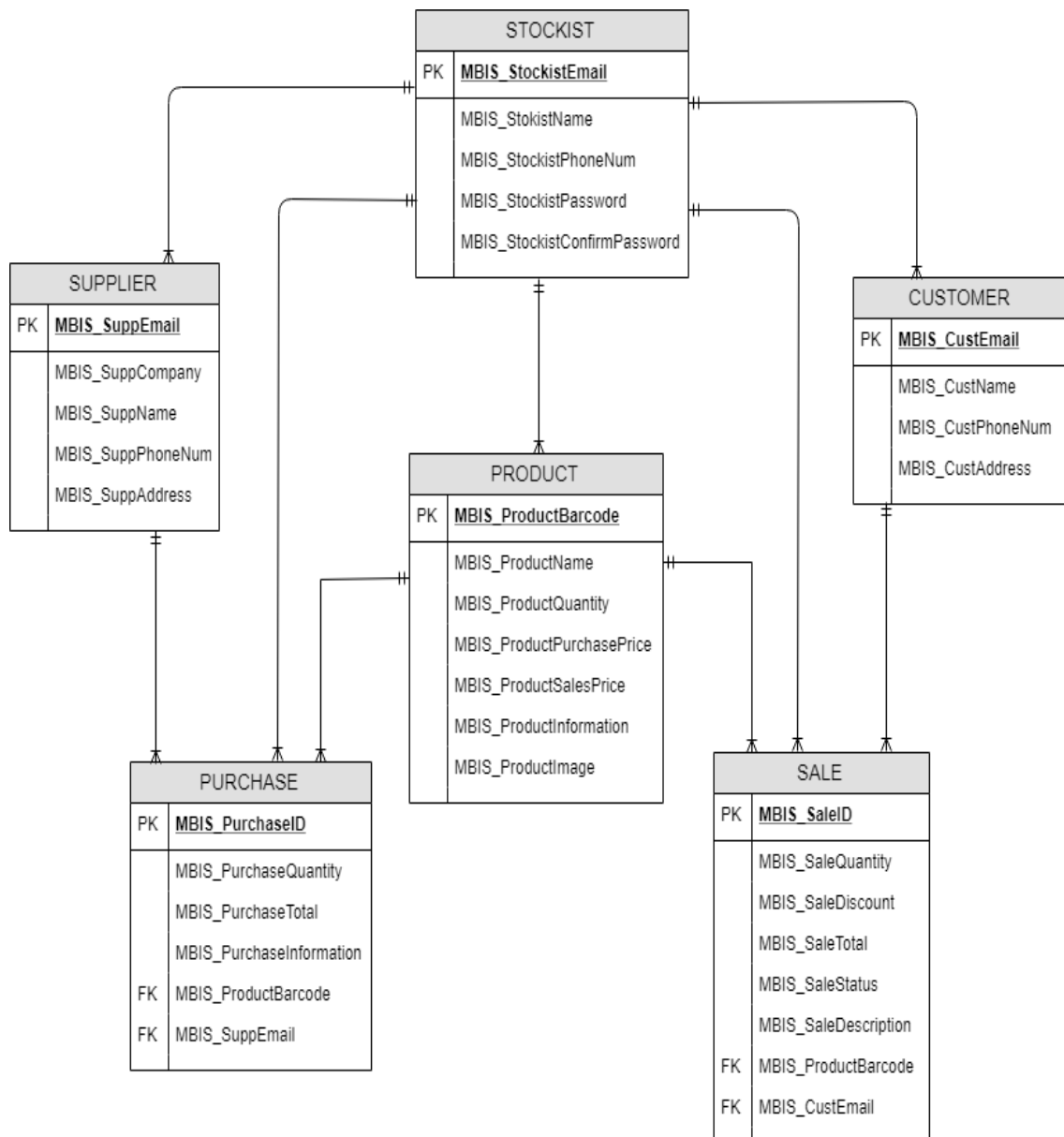


Figure 3.6 Entity Relationship Diagram (ERD) of Mobile Based for Inventory System

### 3.2.6 Propose System Interface

In propose system interface, dialog diagram will provided to show the interaction between each interface in Mobile Based for Inventory System. Figure 3.7 show the dialog diagram and Appendix B-2 shown the Graphical User Interface (GUI) for Mobile Based for Inventory System.

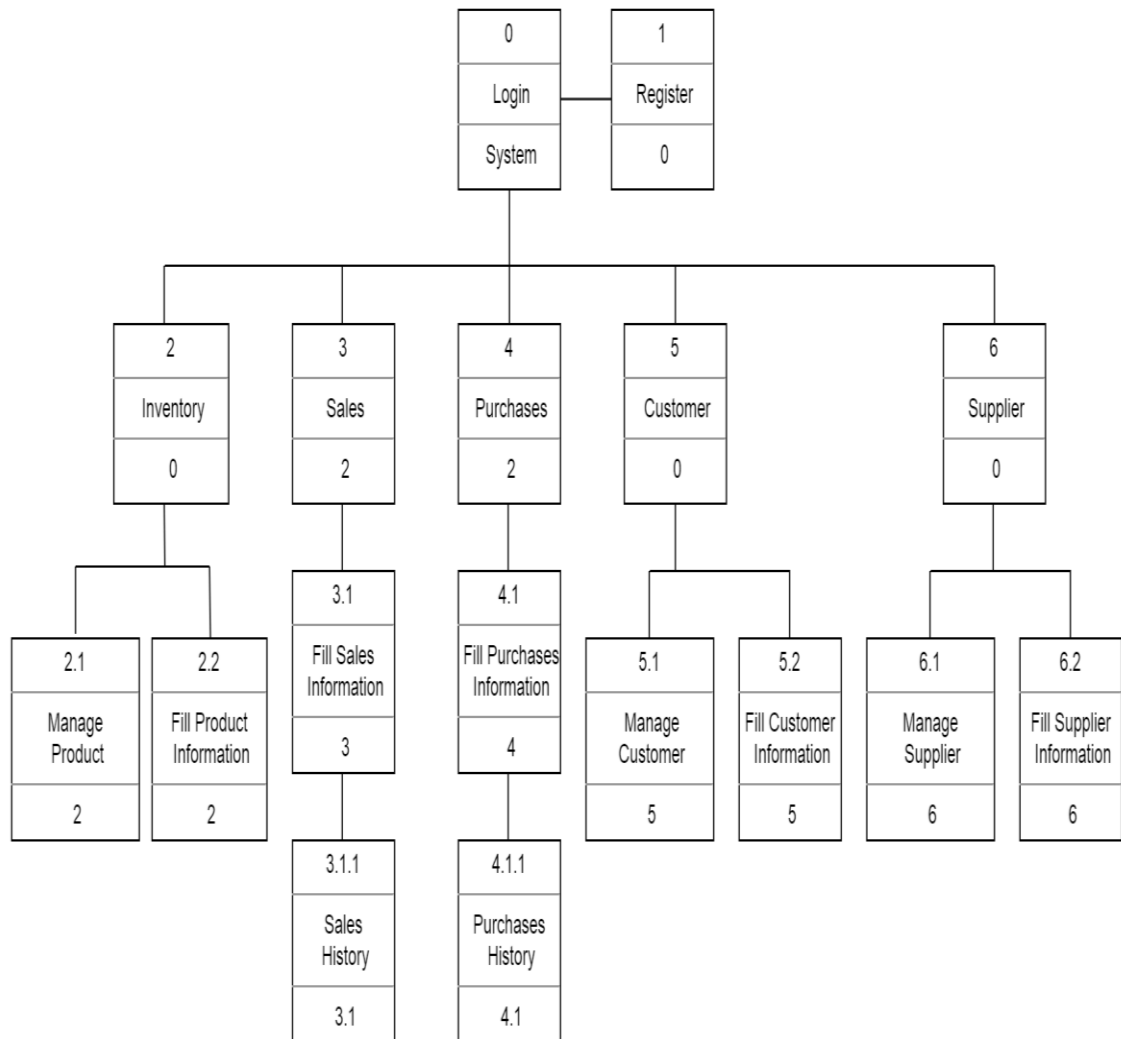


Figure 3.7 Dialog Diagram for Mobile Based for Inventory System

### 3.2.7 Modules / Functions of the System

The modules or functions that are used in this system, Mobile Based for Inventory System are register account, manage inventory, manage sales, manage purchases, manage customer and manage supplier.

First, register account, stockist need to register first in order to get password and registered email to login into the system.

Second, manage inventory which is stockist can view the list of product by searching according to name of product and barcode. In addition, stockist can add the product into the system by entering image, barcode, name, quantity, purchase price, sale price, supplier name and the description about the product. In addition, stockist can update and also delete the product information.

Third, manage sales. In this section, stockist can search and choose the products that have been sold, then enter the quantity, customer name and description of product sold. Moreover, stockist can classified their customer into paid or receivables to make it easier to the stockist recognize their customer's status. To know the information of the customer based on product sales, stockist have optional to save customer name and additional sale information. In addition, stockist able to insert percentage discount if there have the discount on the certain product. Next, Stockist also can view the sales history.

Fourth, stockist able to add and record the purchase product from the supplier. In order to manage the purchases, stockist able to search and choose the product that want to purchase then insert the quantity and able to state the name of the provider or supplier. Then, stockist can view the purchases history.

Fifth, manage customer. Stockist can record their customer information like name, phone number, email and address. Stockist can view the list of their customer including the information when needed and directly can call and email customer by clicking the button call and email provided. However, stockist also can update and delete their customer information anytime.

Last but not least, stockist able to display the list of the supplier information searching name of supplier or name of the product. The provider or supplier of the products can be save and record their information such as company name, supplier name, phone number and also email needed and directly can call and email customer by clicking the button call and email provided. Stockist also can manage their supplier information by update and delete the supplier information.

### **3.2.8 Software Requirement Specification (SRS)**

The Software Requirement Specification (SRS) is documentation which describe the user interaction and the requirement of Mobile Based for Inventory System and to ensuring the flow of the system is followed the requirement. In this project, the documentation contains the product description, interfaces requirements, software product features, requirement traceability and also system requirement approval. This document can be referring in Appendix B.

### **3.2.9 Software Design Document (SDD)**

Software Design Document (SDD) are recorded the result of the system design process of Mobile Based for Inventory System. The document completely describes the data dictionary, preliminary design, detailed design and system design approval. This SDD document can be referring in Appendix C.



### 3.3 HARDWARE AND SOFTWARE

Mobile Based for Inventory System is implement in the computer and it requires some hardware and software requirement for developing the complete system which is it has been decided after analysed the system and based on the system need.

#### 3.3.1 Software Items

The software items will be used in the development process as well as deployment of Mobile Based for Inventory System was listed in Table 3.1.

Table 3.1 Software Items

Software Items	Detail and Purpose
Microsoft Word 2013	Documentation
Microsoft Project 2016	Gantt chart for documentation
Microsoft Power Point Presentation	Presentation of documentation
Android Studio	For development prototype of Mobile Based for Inventory System
Google Chrome	For search the resources
Windows 8 64-bits	The development platform OS
Google Drive	Backup the documents

### 3.3.2 Hardware Items

Table 3.2 are listed of hardware will be used in the project implementation and development.

Table 3.2 Hardware Items

Hardware Items	Detail and Purpose
Acer Aspire E1-470PG (Laptop)	Planning, documenting, development and testing.
Pendrive 16GB	User temporarily data transfer the project.

### 3.4 GANTT CHART

The development of Mobile Based for Inventory System can be present in form of Gantt chart starting from the requirement, analysis, design and implementation. The Gantt chart is created to show the process planning for the whole project from the first week start project until the presentation day as shown in Appendix A.

## **CHAPTER 4**

### **IMPLEMENTATION, TESTING AND RESULT DISCUSSION**

#### **4.1 INTRODUCTION**

In this chapter, it describe about the development of the system and some of the implementation that have been used in Mobile Based for Inventory System. The development process involved of creating the interface using Android Studio that connects with database which is Google Excel. Generally, the focus in this chapter is to ensure the functionality of the system and to apply the methodology as mention in previous chapter. The discussion in this chapter will be initiated from the system interface.

#### **4.2 IMPLEMENTATION**

This section describes about the steps in developing and specific implementation requirements of the application. Each process involved in the project development has been recorded. In general, Mobile Based for Inventory System used two development phases, front end development and back end development which Android Studio and Google Apps Script respectively.

## 4.2.1 Development Environment

First and foremost, Android Studio is front-end development tools that has been used in this project. Front end development manage everything that users visually can see in the application or browser. Android Studio is android's official integrated development environment (IDE) for application development exclusively for android platform based on IntelliJ IDEA. Android studio also support for Google Cloud Platform where making it easy to integrate google Cloud Messaging and Application Engine. As this project is android application which is native platform, java and xml have been used along the development of this project. Moreover, it enables to run in Android Virtual Device as emulator. So, it is suitable for Mobile Based for Inventory System to develop by using Android Studio.

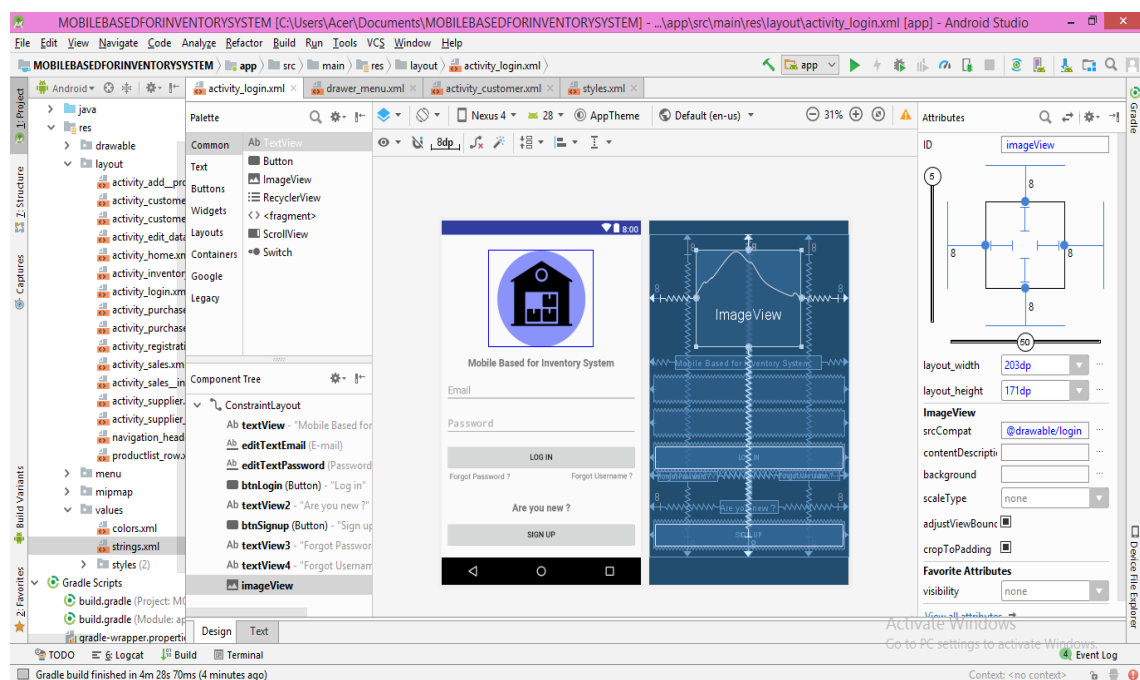


Figure 4.1 Android Studio Environment Interface for Design Layout

Figure 4.1 shows the Android Studio environment interfaces for XML layout. In this interface, it include the Design layout and Text layout which is Design layout is rich layout editor that allow users to drag and drop UI components into a visual design editor instead of writing the layout XML by hand and have option to preview layout on multiple screen configurations.

In this interface, it includes palette for easy to drag list of views and view groups into the layout, component tree for view hierarchy for the layout, toolbar for configure the layout appearance in the editor and to change some layout attributes, design editor for view layout in Design or Blueprint view and lastly, attributes for change the selected view's attributes.

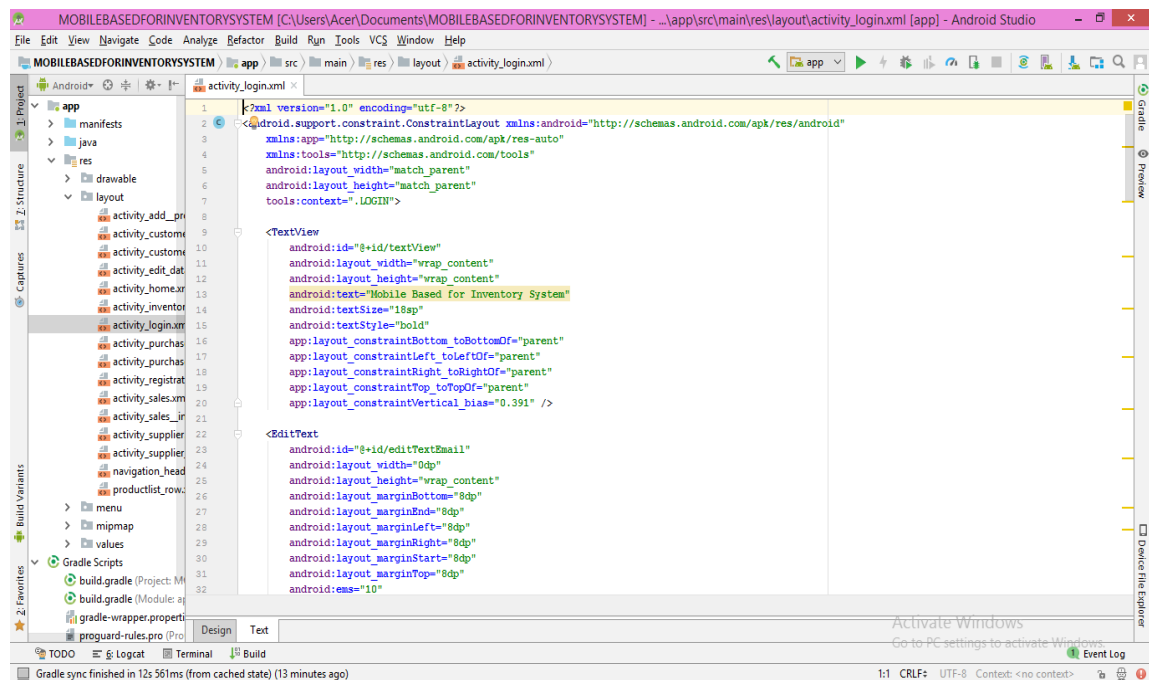


Figure 4.2 Android Studio Environment Interface for Text Layout

Figure 4.2 shows the Android Studio environment interfaces for Text layout. A layout defines the user interface's structure of application, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects. Usually, View objects called "widgets" and can be one of many subclasses like TextView or Button. Then, the ViewGroup objects are usually called "layouts" which is can be one of many types that provide a different layout structure, such as ConstraintLayout or LinearLayout.

User can quickly design UI layouts and the screen elements by using Android's XML vocabulary. In the same way user can create web pages in HTML with a series of nested elements. Each layout file must consists exactly one root element either View or ViewGroup object. User can add additional layout objects or widgets as child elements once defined the root element to gradually build a View hierarchy that defines the layout.

It will properly compile after declared the layout in XML and save the file with .xml extension in Android project's res/layout/directory.

Each View and ViewGroup object support their own variety of XML attributes. Some attributes are specific to a View object such as TextView supports the textSize attributes and some are common to all View objects because its inherited from the root View class, for example the id attribute. Then, other attributes are considered “layout parameters” which are attributes that describe certain layout orientations of the View object, as defined by that object's parent ViewGroup object.

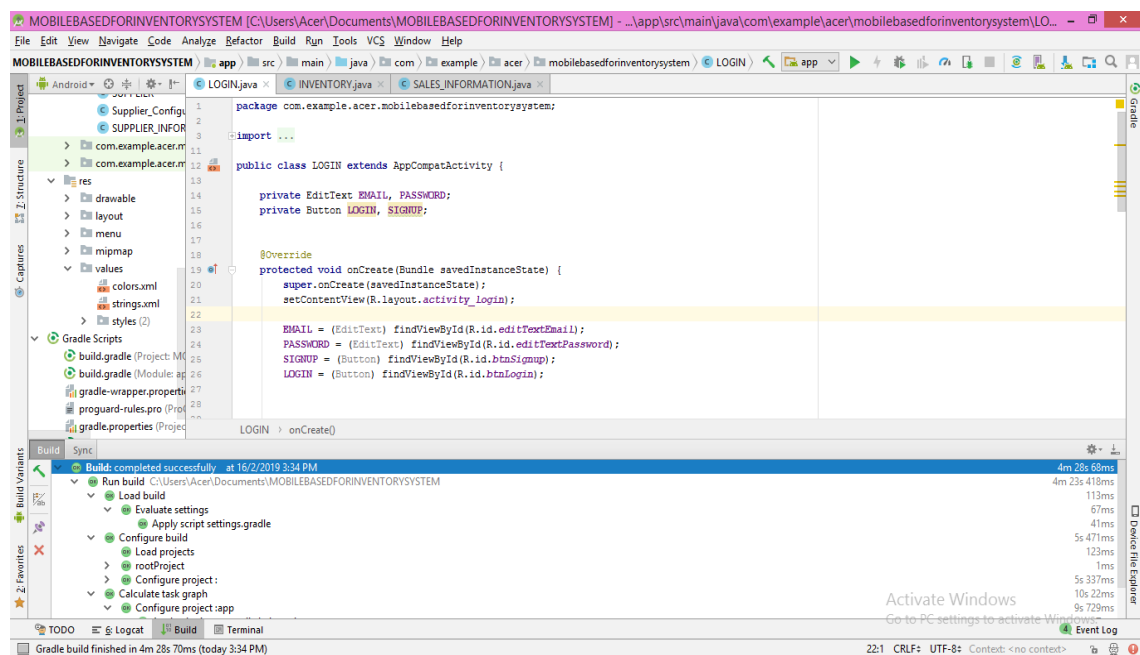
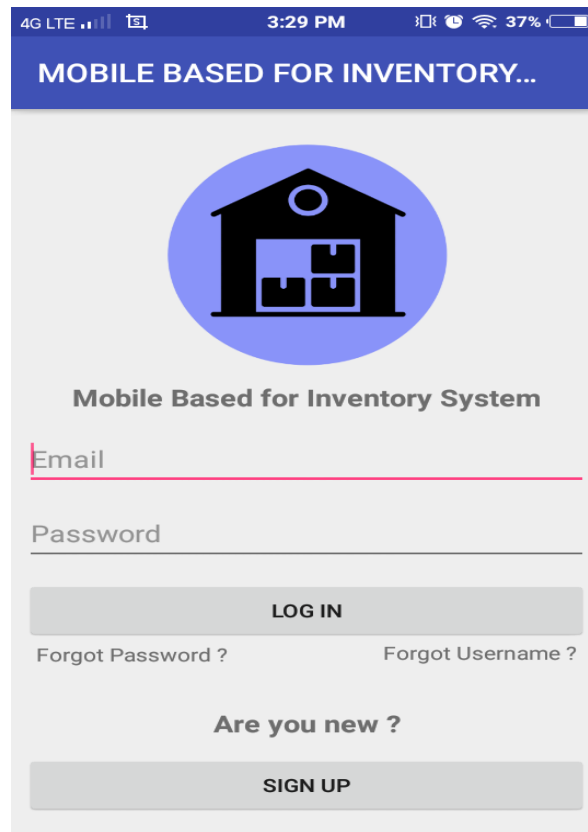


Figure 4.3 Android Studio Environment Interface for Java Layout


Figure 4.2 shows the Android Studio environment interfaces for Java layout which is main window to develop the system. In this interface consist the tool window placed at the below of the window that state the successful task and notice user if have errors during running an application. Furthermore, it shows code colour which is easier for user to detect errors if the code not suitable with the code functionality.

Mobile Based for Inventory System developed by using Java language. Java is an object-oriented programming language and designed to be platform independent and secure by using virtual machines that allows to create compiled programs that run on nearly every platform. Moreover, Java a very popular programming language developed by Sun Microsystems.



4G LTE 3:29 PM 37%

**MOBILE BASED FOR INVENTORY...**



**Mobile Based for Inventory System**

Email

Password

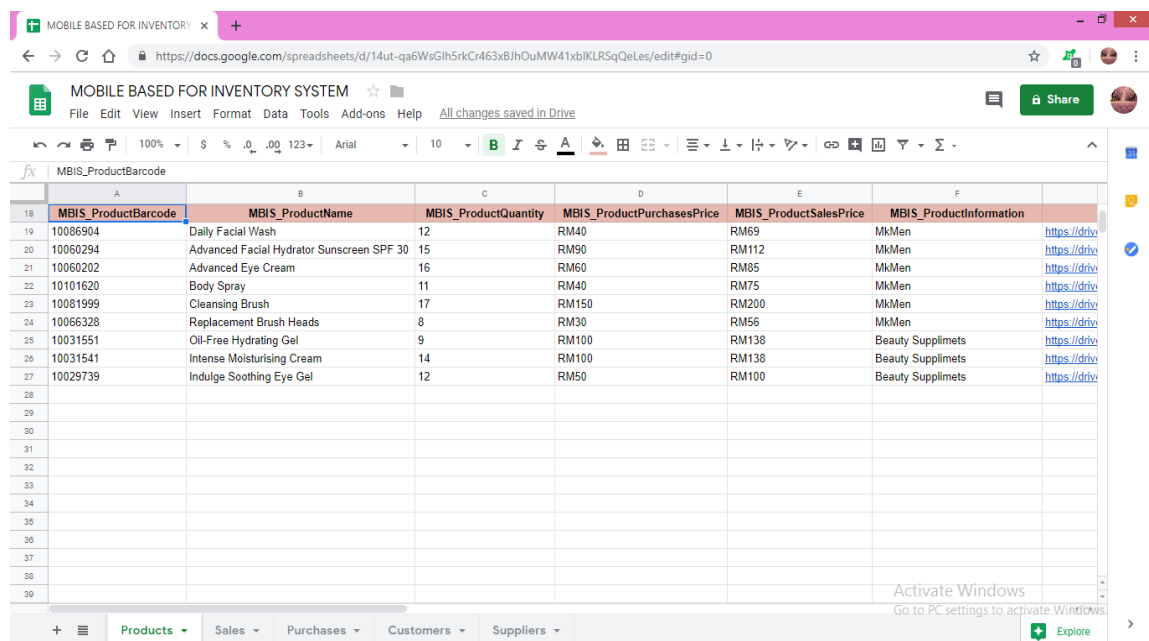
**LOG IN**

Forgot Password ?      Forgot Username ?

**Are you new ?**

**SIGN UP**

So, Java code will apply in this project to make the validation of the user input and to add more action in the system. For the user input, it helps to validate whether user input is correct data or not. This help the system to become more efficient in handling the data from user. Figure 4.4 shows how Java works to validate the data from user.



MOBILE BASED FOR INVENTORY SYSTEM

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

	A	B	C	D	E	F
	MBIS_ProductBarcode	MBIS_ProductName	MBIS_ProductQuantity	MBIS_ProductPurchasesPrice	MBIS_ProductSalesPrice	MBIS_ProductInformation
19	10086904	Daily Facial Wash	12	RM40	RM69	MkMen <a href="https://drive">https://drive</a>
20	10060294	Advanced Facial Hydrator Sunscreen SPF 30	15	RM90	RM112	MkMen <a href="https://drive">https://drive</a>
21	10060202	Advanced Eye Cream	16	RM60	RM85	MkMen <a href="https://drive">https://drive</a>
22	10101620	Body Spray	11	RM40	RM75	MkMen <a href="https://drive">https://drive</a>
23	10081999	Cleansing Brush	17	RM150	RM200	MkMen <a href="https://drive">https://drive</a>
24	10066328	Replacement Brush Heads	8	RM30	RM56	MkMen <a href="https://drive">https://drive</a>
25	10031551	Oil-Free Hydrating Gel	9	RM100	RM138	Beauty Supplimets <a href="https://drive">https://drive</a>
26	10031541	Intense Moisturising Cream	14	RM100	RM138	Beauty Supplimets <a href="https://drive">https://drive</a>
27	10029739	Indulge Soothing Eye Gel	12	RM50	RM100	Beauty Supplimets <a href="https://drive">https://drive</a>
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						

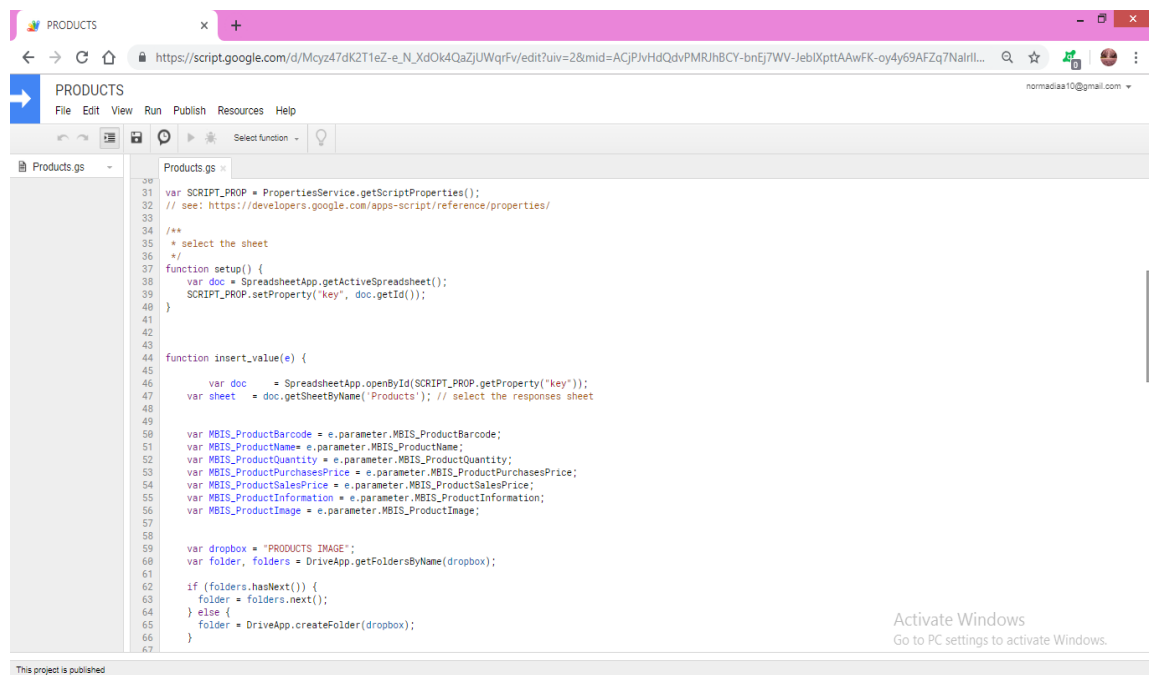
Activate Windows  
Go to PC settings to activate Windows

Products Sales Purchases Customers Suppliers

Figure 4.5 Google Excel Environment Interface

Figure 4.5 shows the Google Excel environment interfaces act as database server for Mobile Based for Inventory System. Database is most important in this project to store all the data from user and easier to the user to check the information at anytime and anywhere without having the system. Besides, the file and each sheet should named to create a connection. The file can be set up to any names and create a many sheet for the system use.

Additionally, Google Apps Script acts as Application Programming Interface (API). APIs just application to communicate with one another between Android Studio and Google Excel. Google Apps Script has to be code to insert, delete and update the information automatically into Google Excel by using the system. The example of Google Apps Script is showing in the figure below



```

31 var SCRIPT_PROP = PropertiesService.getScriptProperties();
32 // see: https://developers.google.com/apps-script/reference/properties/
33
34 /**
35  * select the sheet
36  */
37 function setup() {
38   var doc = SpreadsheetApp.getActiveSpreadsheet();
39   SCRIPT_PROP.setProperty("key", doc.getId());
40 }
41
42
43
44 function insert_value(e) {
45
46   var doc = SpreadsheetApp.openById(SCRIPT_PROP.getProperty("key"));
47   var sheet = doc.getSheetByName("Products"); // select the responses sheet
48
49
50   var MBIS_ProductBarcode = e.parameter.MBIS_ProductBarcode;
51   var MBIS_ProductName = e.parameter.MBIS_ProductName;
52   var MBIS_ProductQuantity = e.parameter.MBIS_ProductQuantity;
53   var MBIS_ProductPurchasesPrice = e.parameter.MBIS_ProductPurchasesPrice;
54   var MBIS_ProductSalesPrice = e.parameter.MBIS_ProductSalesPrice;
55   var MBIS_ProductInformation = e.parameter.MBIS_ProductInformation;
56   var MBIS_ProductImage = e.parameter.MBIS_ProductImage;
57
58
59   var dropdown = "PRODUCTS IMAGE";
60   var folders = DriveApp.getFoldersByName(dropdown);
61
62   if (folders.hasNext()) {
63     folder = folders.next();
64   } else {
65     folder = DriveApp.createFolder(dropdown);
66   }
67

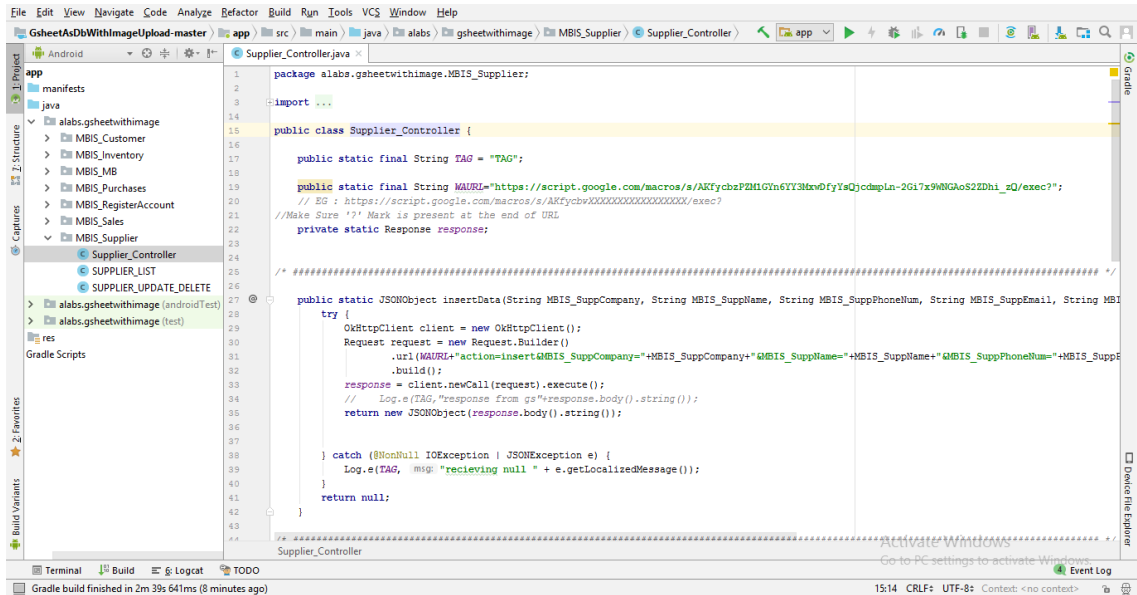
```

Figure 4.6 Google Apps Script Environment Interface

Google Apps Script is a rapid application development platform that makes it fast and easy to create business applications that integrate with G Suite. The language has been used to write code is JavaScript and have access to built-in libraries for favourite G Suite applications like Calendar, Drive, Gmail and more. Next, it can write custom functions and macros for Google Sheets then can publish web applications either standalone or embedded in Google Sites.



This script need to be publish first as a web application then requires authorization of the user before distributing the URL. After authorization success, user need to copy the web application URL and put into the system so that the system and Google Excel has connection to insert, update and delete the data as in Figure 4.7.



```
1 package alabs.gsheetswithimage.MBIS_Supplier;
2
3 import ..
4
5 public class Supplier_Controller {
6
7     public static final String TAG = "TAG";
8
9     public static final String MAURL="https://script.google.com/macros/s/ARFyCzbzF2M1G1n6TY3kxwOfy1sQjcdpLn-2G17x9WNgAoS2Eh1_zQ/exec?";
10    // BG : https://script.google.com/macros/s/ARFyCzbzF2M1G1n6TY3kxwOfy1sQjcdpLn-2G17x9WNgAoS2Eh1_zQ/exec?
11    //Make Sure '?' Mark is present at the end of URL
12    private static Response response;
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27    /** *****
28
29     * public static JSONObject insertData(String MBIS_SuppCompany, String MBIS_SuppName, String MBIS_SuppPhoneNum, String MBIS_SuppEmail, String MBIS_SuppAddress)
30     * @return
31     */
32    public static JSONObject insertData(String MBIS_SuppCompany, String MBIS_SuppName, String MBIS_SuppPhoneNum, String MBIS_SuppEmail, String MBIS_SuppAddress)
33    {
34        try {
35            OkHttpClient client = new OkHttpClient();
36            Request request = new Request.Builder()
37                .url(MAURL+"action=insert&MBIS_SuppCompany="+MBIS_SuppCompany+"&MBIS_SuppName="+MBIS_SuppName+"&MBIS_SuppPhoneNum="+MBIS_SuppPhoneNum+"&MBIS_SuppEmail="+MBIS_SuppEmail+"&MBIS_SuppAddress="+MBIS_SuppAddress)
38                .build();
39            response = client.newCall(request).execute();
40            // Log.e(TAG,"response from gs"+response.body().string());
41            return new JSONObject(response.body().string());
42        }
43        catch (NullPointerException | JSONException e) {
44            Log.e(TAG, "recieving null " + e.getMessage());
45        }
46        return null;
47    }
48 }
```

The screenshot displays an IDE window for 'Supplier\_Controller.java'. The code defines a class with a static TAG, a MAURL constant, and an insertData method that sends an HTTP request to a Google Sheet script to insert supplier data. The IDE interface includes a project structure on the left, a status bar at the bottom, and a 'Terminal' tab.

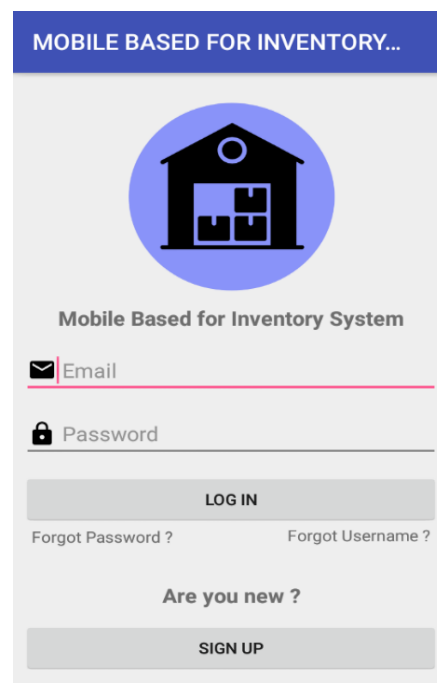
Figure 4.7 Product Configuration Interface

## 4.2.2 System Functionality

This section explain the system functionality that is the interaction between user and the system. User interface of the system should be friendly in order to ensure the understandability of the user.

### a. Register Account

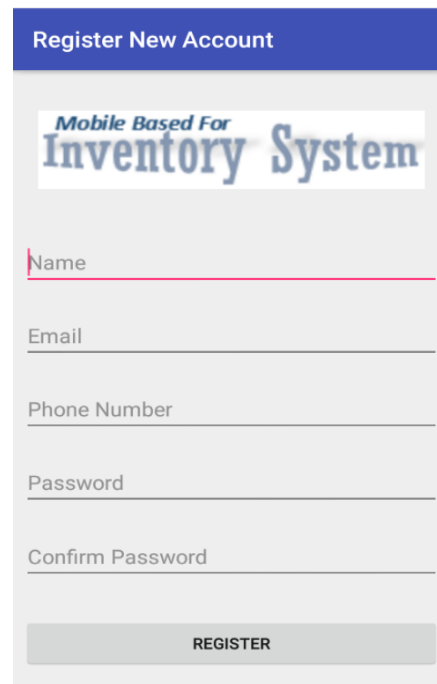
Figure 4.8 shown login interface where stockist need to enter their registered or valid email and password. For new user, they can register new account by clicking button “SIGN UP” in order to get their email and password.



The screenshot displays the login interface for the 'Mobile Based for Inventory System'. At the top, a blue header bar contains the text 'MOBILE BASED FOR INVENTORY...'. Below this is a circular logo featuring a black house icon with a white window and a white door, set against a blue background. Underneath the logo, the text 'Mobile Based for Inventory System' is displayed. The interface includes two input fields: 'Email' with an envelope icon and 'Password' with a lock icon. A 'LOG IN' button is positioned below the password field. Below the 'LOG IN' button, there are two links: 'Forgot Password ?' and 'Forgot Username ?'. At the bottom, there is a section titled 'Are you new ?' with a 'SIGN UP' button.

Figure 4.8 Login Interface

Figure 4.9 shown register interface which is user need to register their account by fill in their valid information including the password to secure their account. Then user need click button “REGISTER” to proceed to the login page.



The image shows a mobile application interface for registering a new account. At the top, there is a blue header with the text "Register New Account". Below the header is a logo that reads "Mobile Based For Inventory System". The main form area contains five input fields, each with a label and a horizontal line for text entry: "Name", "Email", "Phone Number", "Password", and "Confirm Password". At the bottom of the form is a grey button with the text "REGISTER" in capital letters.

Figure 4.9 Register Interface

### **b. Manage Inventory**

In manage inventory function, stockist is able to manage the product information. Figure 4.10 shown an inventory interface. In this interface, the registered item will be display automatically in list view by barcode, name, quantity and sale price of product. User can edit and delete the information of the items by click on the specific item if there have any changes. Next, the new product can be add into the system by clicking the floating button and it will go to the product details interface.










Inventory		
	<b>10049797</b> Freshen Quantity : 15      Sales (RM) : 72	
	<b>10054040</b> Mary Kay Liquid Foundation Brush Quantity : 12      Sales (RM) : 53	
	<b>10060294</b> Advanced Facial Hydrator Sunscreen SPF Quantity : 16      Sales (RM) : 112	
	<b>10101620</b> Body Spray Quantity : 15      Sales (RM) : 75	
	<b>10078697</b> CITYSCAPE Cologne Spray Quantity : 9      Sales (RM) : 139	
	<b>10075680</b> MK High Intensity Sport Cologne Spray Quantity : 15      Sales (RM) : 159	
	<b>10060202</b> Advanced Eye Cream Quantity : 12      Sales (RM) :	

Figure 4.10 Product Interface

Figure 4.11 shown add product information interface which user need to insert the information of the products such as the picture, product code, product name, quantity, purchases, sales, and additional information of the product. The price of the sale must be higher than purchase price.

Add Product Details

Product Code :

Product Name :

Quantity :	Purchases (RM) :	Sales (RM) :
<input style="width: 30%;" type="text"/>	<input style="width: 30%;" type="text"/>	<input style="width: 30%;" type="text"/>

Additional Product Information :

✕
📄

Figure 4.11 Add Product Information Interface

Figure 4.12 below shows the interface of update and delete product information. The information of the product automatically displayed when user click on the specific product to delete and update.

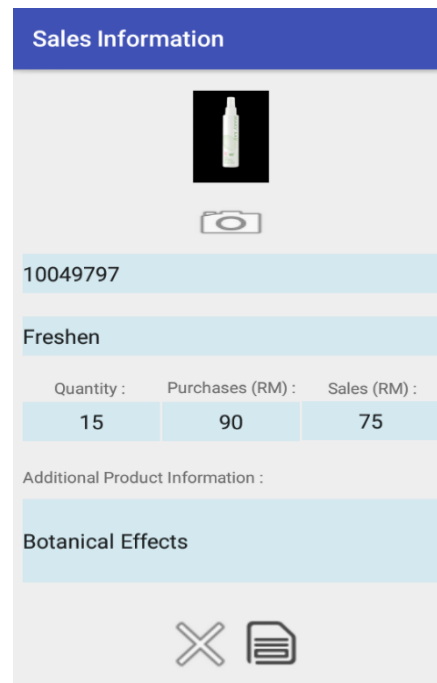


Figure 4.12 Update and Delete Product Interface

### c. Manage Sales

In this manage sales function, stockist is able to manage the sales information by clicking to the specific product or search by product barcode and it will go to the sales information interface. In the same time, stockist also can check the history of the sales by click button "SALES HISTORY" as shown in Figure 4.13.

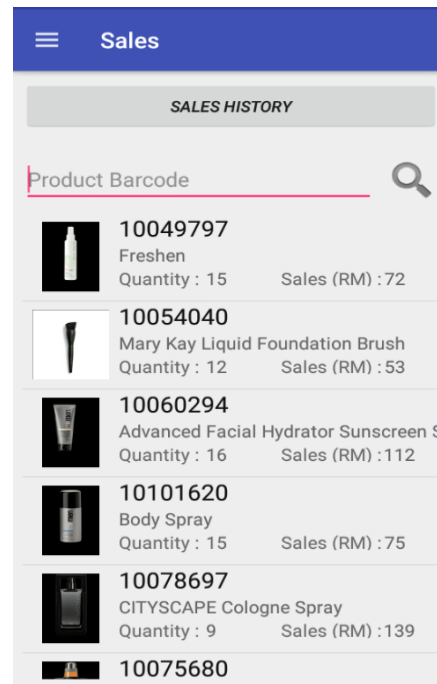


Figure 4.13 Sale Interface

Figure 4.14 below shown the interface of add sale information which included quantity, total sale, paid or receivable, client name and additional sale information.

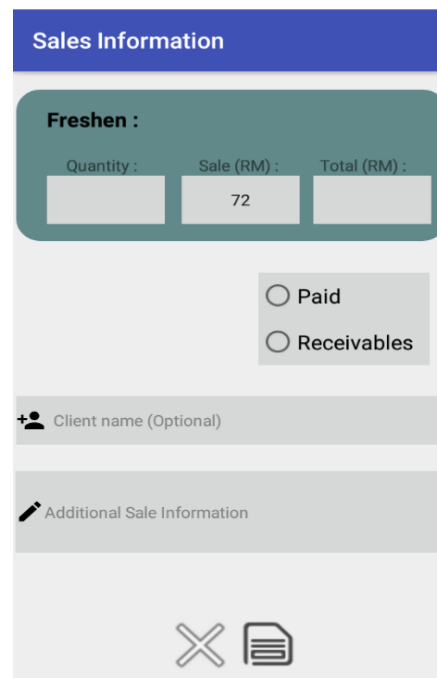


Figure 4.14 Add Sale Information Interface

Figure 4.15 shown the list of the sales history including product name, quantity, client name, information, total sale price, status sale and date and time.

Sales History	
<b>Product Name : Freshen</b>	
Quantity : 245	Total (RM) : 86
Client Name :	Receivables
Information : a	
April 5, 2019 1:28:55 AM HKT	
<b>Product Name : Mary Kay Liquid Foundation Brush</b>	
Quantity : 45	Total (RM) : 58
Client Name :	Receivables
Information : b	
April 5, 2019 2:32:48 AM HKT	
<b>Product Name : Advanced Facial Hydrator Sunscreen SF</b>	
Quantity : 25	Total (RM) : 44
Client Name :	Paid
Information : c	
April 5, 2019 2:33:50 AM HKT	
<b>Product Name : Body Spray</b>	
Quantity : 55	Total (RM) : 2
Client Name :	Paid
Information : d	
April 5, 2019 2:34:28 AM HKT	
<b>Product Name : CITYSCAPE Cologne Spray</b>	
Quantity : 619	Total (RM) : 848
Client Name :	Receivables
Information : a	

Figure 4.15 Sale History Interface

#### d. Manage Purchases

In this manage purchases function, stockist is able to manage the purchase information by clicking to the specific product or search by product barcode and it will go to the purchases information interface. In a meanwhile, stockist also can check the history of the purchases product by click button “SALES HISTORY” as shown in Figure 4.16.

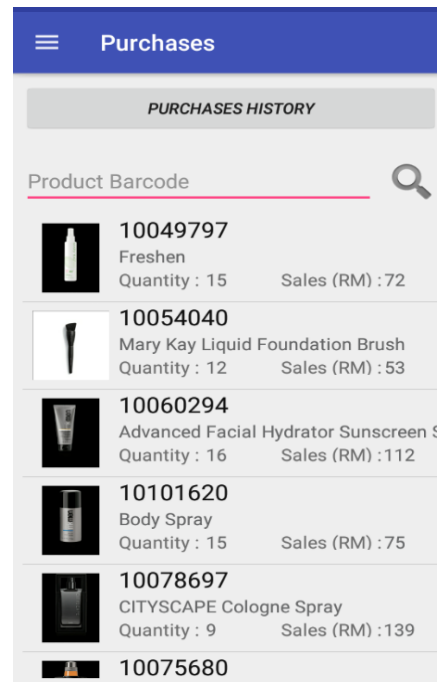


Figure 4.16 Purchase Interface

Figure 4.17 below shown the interface of add purchase information which included quantity to add, total quantity, purchase price, sale price, total purchase, supplier name and additional information of purchase.

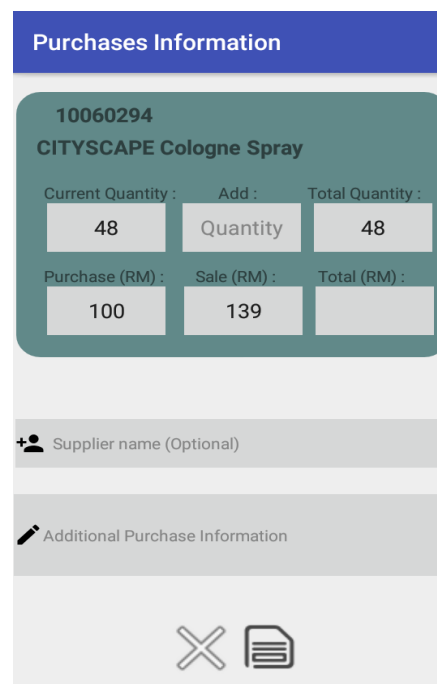


Figure 4.17 Add Purchase Information Interface



Figure 4.18 shown the list of the purchase history including product name, quantity, supplier name, information, total purchase price, status sale and date and time.

Purchases History	
<b>Product Name : Mary Kay Liquid Foundation Brush</b>	
Quantity : 2	Total (RM) : 130
Supplier Name :	
Information : a	
<i>April 7, 2019 2:17:22 AM HKT</i>	
<hr/>	
<b>Product Name : Freshen</b>	
Quantity : 2	Total (RM) : 200
Supplier Name :	
Information : b	
<i>April 7, 2019 3:36:35 AM HKT</i>	
<hr/>	
<b>Product Name : Body Spray</b>	
Quantity : 5	Total (RM) : 450
Supplier Name :	
Information : c	
<i>April 14, 2019 12:10:40 PM HKT</i>	
<hr/>	
<b>Product Name : CITYSCAPE Cologne Spray</b>	
Quantity : 3	Total (RM) : 450
Supplier Name :	
Information : d	
<i>April 22, 2019 8:22:55 PM HKT</i>	
<hr/>	
<b>Product Name : CITYSCAPE Cologne Spray</b>	
Quantity : 3	Total (RM) : 450
Supplier Name :	
Information : a	

Figure 4.18 Purchase History Interface

#### e. Manage Customer

Stockist able to manage the customer information in manage customer such as stockist able to add, update, delete the customer information. Figure 4.19 below shown customer interface where stockist can recorded their customer information. In this page were displayed the list of the customers that have been recorded by stockists and can add new customers by clicking floating button. In the same time, user also can update and delete the information customers that they want by clicking on the specific name of the customer.

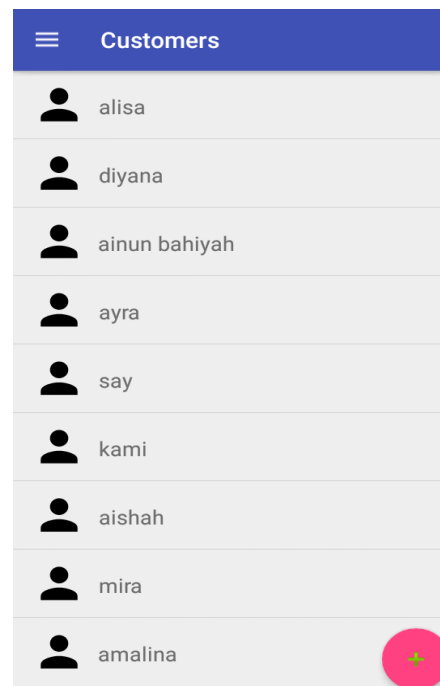


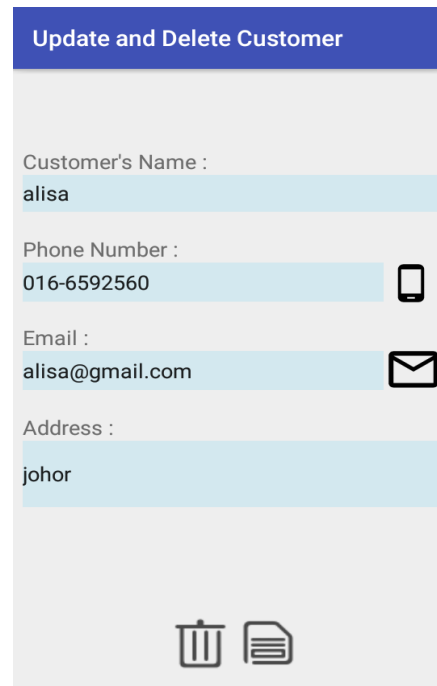
Figure 4.19 Customer Interface

Figure 4.20 shown add customer information interface where user need to fill in their customer's information such as name, phone number, email and address in order to save customer information either regular customers or not.

A screenshot of a mobile application interface titled "Add Customer Details". The interface has a blue header bar with the text "Add Customer Details" in white. Below the header is a light gray background with four light blue input fields stacked vertically. The labels for these fields are "Customer's Name", "Phone Number", "Email", and "Address". At the bottom of the form, there are two icons: a red "X" icon and a document icon.

Figure 4.20 Add Customer Information Interface

Stockist able to see the customer information including name of customer, phone number, email, and address as shown in Figure 4.21. In this interface, stockist able to delete and update customer information. Then, email and phone button also provided to ease stockist make a phone call and send an email to the customers.



The screenshot displays a mobile application interface titled "Update and Delete Customer". It features a form with the following fields and values:

- Customer's Name : alisa
- Phone Number : 016-6592560 (with a phone icon)
- Email : alisa@gmail.com (with an email icon)
- Address : johor

At the bottom of the form, there are two icons: a trash can (delete) and a document (update).

Figure 4.21 Update and Delete Interface

#### f. Manage Supplier

In this manage supplier function, Stockist able to manage the supplier information. Figure 4.22 below shown supplier interface. Stockist can recorded their supplier information. In this page were displayed the list of the suppliers that have been recorded by stockists and can add new supplier by clicking floating button. In the same time, user also can update and delete the information supplier that they want by clicking on the specific name of the supplier.

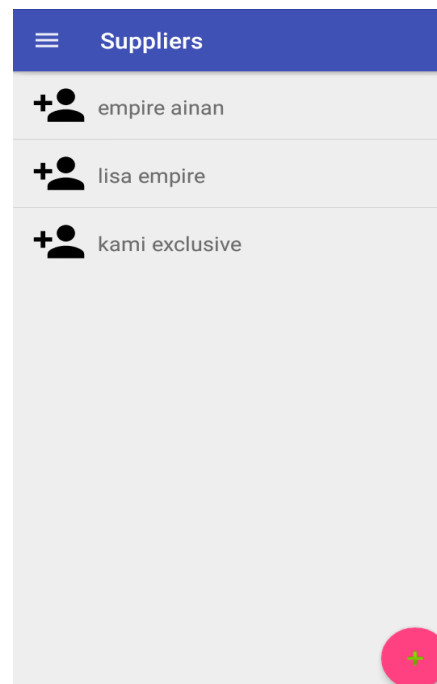


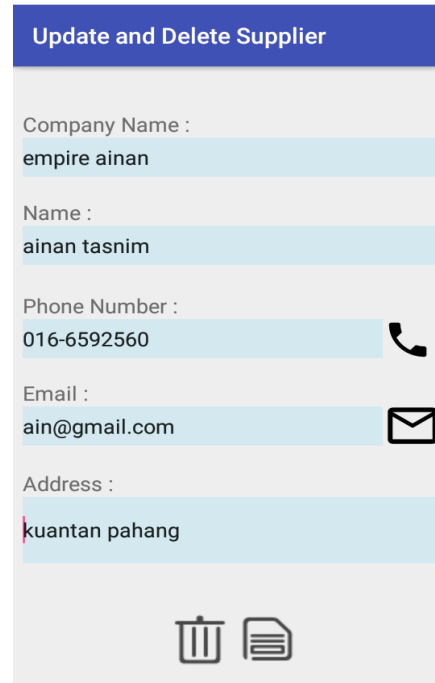
Figure 4.22 Supplier Interface

Figure 4.23 shown add supplier interface where user need to fill in their supplier's information such as name of company, name of supplier, phone number, email and address.

A mobile application interface titled "Add Supplier Details" with a blue header. Below the header, there are five light blue input fields stacked vertically, labeled "Company", "Supplier's Name", "Phone Number", "Email", and "Address". At the bottom of the form, there are two icons: a grey "X" icon for closing and a grey document icon for saving.

Figure 4.23 Add Supplier Information Interface

Stockist able to see the supplier information including company name, supplier name, phone number, email, and address as shown in Figure 4.21. In this interface, stockist able to delete and update supplier information. Then, email and phone button also provided to ease stockist make a phone call and send an email to the suppliers.



The screenshot displays a mobile application interface titled "Update and Delete Supplier". It features a list of supplier details for "empire ainan". Each detail is presented in a light blue input field with a corresponding label and icon. The details include: Company Name (empire ainan), Name (ainan tasnim), Phone Number (016-6592560) with a phone icon, Email (ain@gmail.com) with an envelope icon, and Address (kuantan pahang). At the bottom of the form, there are two icons: a trash can for deletion and a document for editing.

Field	Value	Icon
Company Name :	empire ainan	
Name :	ainan tasnim	
Phone Number :	016-6592560	Phone
Email :	ain@gmail.com	Email
Address :	kuantan pahang	

Figure 4.24 Update and Delete Interface

### 4.3 TESTING AND RESULT DISCUSSION

This project use Android Studio to build and test the application codes on smartphone. In order to implement this, the mobile phone used is Vivo Y51 5.0.2 on Android 3.0 operating system. The targeted user to test this application is stockist.

Next, the testing process takes a few steps where the developer need to assure that every interface and written code goes smoothly without any errors ad bugs. Moreover, it is necessary for the application to meet all the functions and requirements that has been specified since the early development.

#### 4.3.1 Functional Testing

Table 4.1 Test Cases in Functional Testing

Use Case	Test Case	Function Being Tested	Input	Expected Output	Actual Result	Status
Register Account	T1	System insert a valid user information	Insert user information and click register button	Registration successful	Registration successful	Pass
Register Account	T2	System insert previous user information	Insert previous user information and click register button	Registration unsuccessful	Registration unsuccessful	Pass

Register Account	T3	System handles a valid username and password properly	Enter correct password and username	Login successful	Login successful	Pass
Register Account	T4	System handles an invalid username and password properly	Enter incorrect password and username	Login unsuccessful	Login unsuccessful	Pass
Manage Inventory	T5	System insert valid product details and display list of products	Insert product details and click save button	Insert product details successful	Insert product details successful	Pass
Manage Inventory	T6	System insert previous product details and don't display list of products	Insert previous product details and click save button	Insert product details unsuccessful	Insert product details unsuccessful	Pass
Manage Inventory	T7	System able to display product	Update product information	Update successful	Update successful	Pass

		information and update the details of the product	and click save button			
Manage Inventory	T8	System able to display product information and delete the product information	Click delete button	Delete successful	Delete successful	Pass
Manage Sales	T9	System able to display product information and search product by product code	Insert product barcode and click button search	Display product successful	Display product successful	Pass
Manage Sales	T10	System insert valid product sale details	Insert sale details and click save button	Insert product sale successful	Insert product sale successful	Pass
Manage Sales	T11	System displays all the sales history	Click the sales history button	Display all product sales history.	Display all product sales history.	Pass



Manage Purchases	T12	System able to display product information and search product by product code	Insert product barcode and click button search	Display product successful	Display product successful	Pass
Manage Purchases	T13	System insert valid product purchase details	Insert purchase details and click save button	Insert product purchase successful	Insert product purchase successful	Pass
Manage Purchases	T14	System displays all the purchases history	Click the purchases history button	Display all product purchases history.	Display all product purchases history.	Pass
Manage Customer	T15	System insert valid customer details and display list of customers	Insert customer details and click save button	Insert customer details successful	Insert customer details successful	Pass
Manage Customer	T16	System insert previous customer details and don't	Insert previous customer details and	Insert customer details unsuccessful	Insert customer details unsuccessful	Pass

		display list of customers	click save button			
Manage Customer	T17	System able to display customer information and update the details of the customer	Update customer information and click save button	Update successful	Update successful	Pass
Manage Customer	T18	System able to display customer information and delete the customer information	Click delete button	Delete successful	Delete successful	Pass
Manage Customer	T19	System able to call customer	Click button call	Call successful	Call successful	Pass
Manage Customer	T20	System able to email customer	Click button email	Email send successful	Email send successful	Pass
Manage Supplier	T21	System insert valid supplier details and	Insert supplier details and click save button	Insert supplier details successful	Insert supplier details successful	Pass

		display list of supplier				
Manage Supplier	T22	System insert previous supplier details and don't display list of suppliers	Insert previous supplier details and click save button	Insert supplier details unsuccessful	Insert supplier details unsuccessful	Pass
Manage Supplier	T23	System able to display supplier information and update the details of the supplier	Update supplier information and click save button	Update successful	Update successful	Pass
Manage Supplier	T24	System able to display supplier information and delete the supplier information	Click delete button	Delete successful	Delete successful	Pass
Manage Supplier	T25	System able to call supplier	Click button call	Call successful	Call successful	Pass

Manage Supplier	T26	System able to email supplier	Click button email	Email send successful	Email send successful	Pass
-----------------	-----	-------------------------------	--------------------	-----------------------	-----------------------	------

### 4.3.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is the last phase of this application testing process. During UAT, actual application users test the application to make sure it can handle the required tasks in real environment, according to specifications. UAT is one of the final and critical application project procedures that must occur before the application can be used commercially. In general, UAT is the usage of the software by people from the intended audience and recording and correcting of any defects which are discovered. It gives users the chance to interact with the software and find out if everything works as it should if features have been overlooked, miscommunicated, not communicated, and so on.

The purpose of this section is to outline the User Acceptance Testing (UAT) process for the system. Approval of this testing implies that reviewers are confident that following the execution of the test plan, the resulting system will be considered fully-tested and eligible for implementation. Any error or problem found by the user must be noted on the form as stated in Appendix E (UAT Form).

## 4.4 USER MANUAL

User Manual generally known as User Guide which is intended to provide assistance for end user using a particular system that has been produce after the system prototype has been done. The user manual can be referring in Appendix D.

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 INTRODUCTION**

In this chapter will summarized all chapter that have been discussed before including summary of problem statements, objectives of the project, methodologies used, implementation and testing.

Generally, Mobile Based for Inventory System is developed for stockist to manage their inventory, inbound and outbound of the product and track the availability of product. This system used Google Excel to support a system database management which is easier to the stockist to view product sales trends like track best sellers or slow movers and prevent out-of-stock scenarios by maintaining minimum stock levels. This system also can prevent the record from missing or lost because all the data had been save in the cloud. In addition, the computerized system more efficient and accurate in calculation such as calculate price of sale, purchase, profit, and loss and sometimes need to apply several formula to calculate all of that.

The important and crucial effort was focused on designing and developing the system in order to achieve all the objectives that has been states from the earlier of the proposal. The first objective is to analyse the problem of current system by produce Mobile Based for Inventory System. The second objective is to design and develop prototype system for Mobile Based for Inventory System using Google Excel as the system database management. The third objective is to test the prototype of Mobile Based

for Inventory System. Three objectives that have been stated from the earlier of the system had been achieved successfully.

Then, methodology is essential method in a development of the project. There are many methodologies that can be used for the development but it has a few suitable methodologies that can be applied for this project development. The suitable methodology for this project development is Agile in order to make sure the process of the project development smoothly. Agile methodology consists five (5) phases which are planning, requirement analysis, designing, testing and building. Agile is an iterative and incremental based development which are process of changeable requirement based on customer needs. Therefore, the proper planning and following the phase is very important to evade delay of the project.

Hereby, there are several enhancements that can be made for future improvement of this Mobile Based for Inventory System to ensure that the usage throughout the system is more reliable and dependable in the future. Further details can be preferred in subtopics 5.3.

## 5.2 PROJECT CONSTRAINT

There are certain of constraints arise using development of this project and the constraint as follows :

- i. System cannot run without internet connection  
To save, retrieve, update and delete data from Google Excel cannot be access without internet connection.
- ii. Cannot run in web based system and IOS  
Since MBIS is mobile based, it can be use in operating system only and cannot be used for web based and IOS.
- iii. Using text watcher  
The system will stopped if stockist delete the information that already inserted in the text watcher.

## 5.3 FUTURE WORK

There are several enhancement can be applied on the Mobile Based for Inventory System for future improvement.

- i. Implementing barcode reader. It is really help stockist inserting true information of the product such as name and barcode number. Furthermore, it can save stockist time.
- ii. Enhancing the system into web based system
- iii. Implementing camera when insert the image of the new product or in update product. It may ease the stockist by taking picture directly without opening and searching in gallery.
- iv. Implement sales and purchases report which easier stockist to analyse the higher demand product.

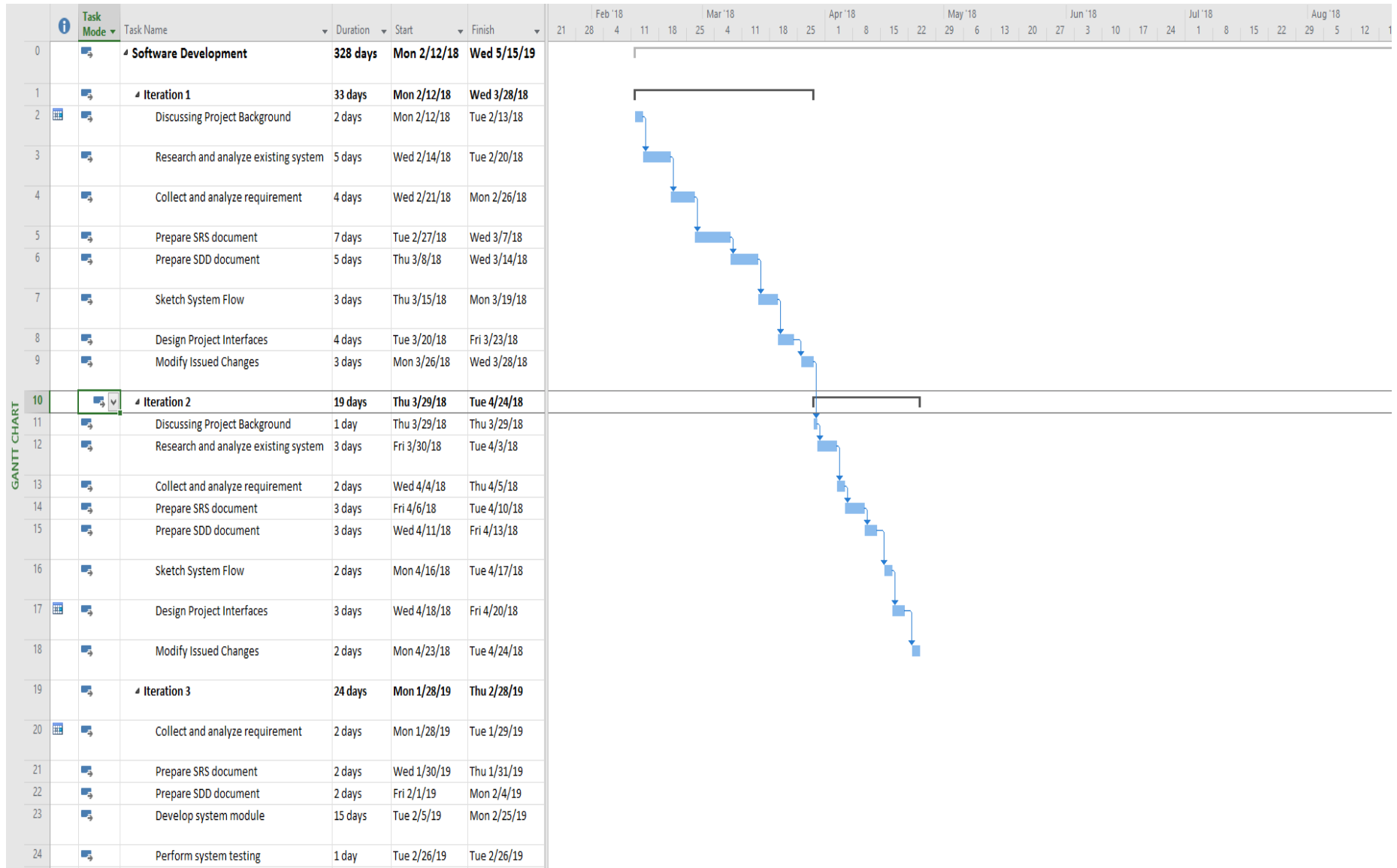
## **REFERENCES**



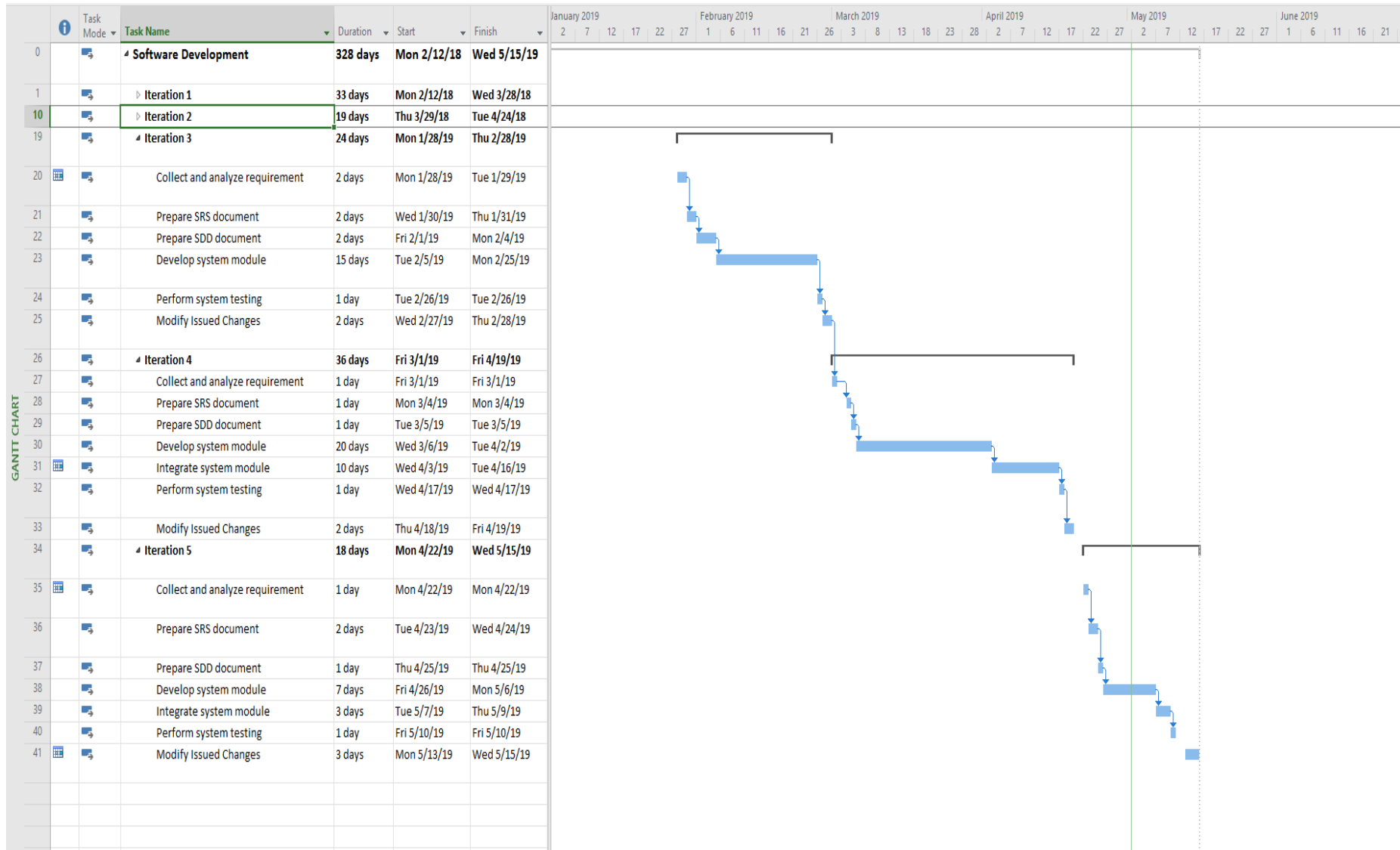
- [1] Green, J. (2018). 4.6 5,970, 3–5. "My Business". Retrieved from :  
[https://play.google.com/store/apps/details?id=com.segb\\_d3v3l0p.minegocio](https://play.google.com/store/apps/details?id=com.segb_d3v3l0p.minegocio)
- [2] Shana, A. (2018). 4.3 87, 3–5. "Smart Inventory management". Retrieved from :  
<https://play.google.com/store/apps/details?id=com.smartapps.smartinventory>
- [3] Toth, J. (2018). 4.4 185, 3–5. "Smart Inventory System". Retrieved from :  
<https://play.google.com/store/apps/details?id=com.nonzeroapps.android.smartinventory>
- [4] Hneif, M., & Ow, S. H. (2009). Review of Agile Methodologies in Software Development 1. *International Journal of Research and Reviews in Applied Sciences*, 1(1), 2076–2734. [https://doi.org/ISSN: 2076-734X, EISSN: 2076-7366](https://doi.org/ISSN:2076-734X,EISSN:2076-7366). Retrieved from :  
[http://www.arpapress.com/Volumes/Vol1/IJRRAS\\_1\\_01.pdf](http://www.arpapress.com/Volumes/Vol1/IJRRAS_1_01.pdf)
- [5] Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile Processes and Methodologies : A Conceptual Study. *International Journal on Computer Science and Engineering*, 4(5), 892–899. Retrieved from : <https://pdfs.semanticscholar.org/5af8/fc8a2be4a16da4ffbe11d5699f71a37f6969.pdf>
- [6] Bowerman, E. (2015). What is Agile Development for Mobile Apps?, 1–7. Retrieved from :  
<http://sourcebits.com/app-development-design-blog/what-is-agile-development-for-mobile-apps/>
- [7] KUMAR, B. R. (2018, MARCH 13). *Coding is Love*. Retrieved from :  
<https://codingislove.com/google-sheets-database/>
- [8] Nilanchala. (2015, JUN 27). *STACK TIPS*. Retrieved from  
<https://stacktips.com/tutorials/android/download-and-display-image-in-android-gridview>

**APPENDIX A**

**GANTT CHART**



GANTT CHART



## **APPENDIX B**

### **SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

**Version**

**1**

MOBILE BASED FOR INVENTORY SYSTEM (MBIS)

---

Faculty Computer Systems & Software Engineering (FSKKP)

# Software Requirement Specification (SRS)

# Table of Contents

1.0	PRODUCT DESCRIPTION	3
1.1	User Characteristics	3
1.2	Constraints	3
2.0	INTERFACES REQUIREMENTS	4
2.1	User Interfaces	4
2.2	Hardware Interfaces	7
2.3	Software Interfaces	7
3.0	SOFTWARE PRODUCT FEATURES	8
3.1	Register Account	8
3.2	Manage Inventory	10
3.3	Manage Sales	13
3.4	Manage Purchases	16
3.5	Manage Customer	19
3.6	Manage Supplier	21
4.0	REQUIREMENTS TRACEABILITY	23
5.0	SYSTEM REQUIREMENTS APPROVAL	27
	Appendix B-1	
	Appendix B-2	

## **1.0 PRODUCT DESCRIPTION**

### **1.1 User Characteristics**

Users who can access to the Mobile Based for Inventory System is only one person that is stockist. In order to access the system, stockist need to register first and create username and password. Any stockists can use this system because there is no education level. In this system stockist can access all services such as manage inventory, manage sales, manage purchases, manage customer and also manage supplier. Moreover, no experience is needed in using Mobile Based for Inventory System because the system provide sufficient information to user.

### **1.2 Constraints**

Below is the list of constraints for the Mobile Based for Inventory System :

- i. The system need internet connection to in order to access and save data from Google excel.
- ii. The language that use in the Mobile Based for Inventory System is in English only.
- iii. Mobile Based for Inventory System cannot run in web based system because this system is mobile based system.



## 2.0 INTERFACES REQUIREMENT

### 2.1 User Interface

The user interface is the stage of the interaction between user and the system. Figure 2.1 shown the dialog diagram together with user interface description of Mobile Based for Inventory System that shown in Table 2.1.

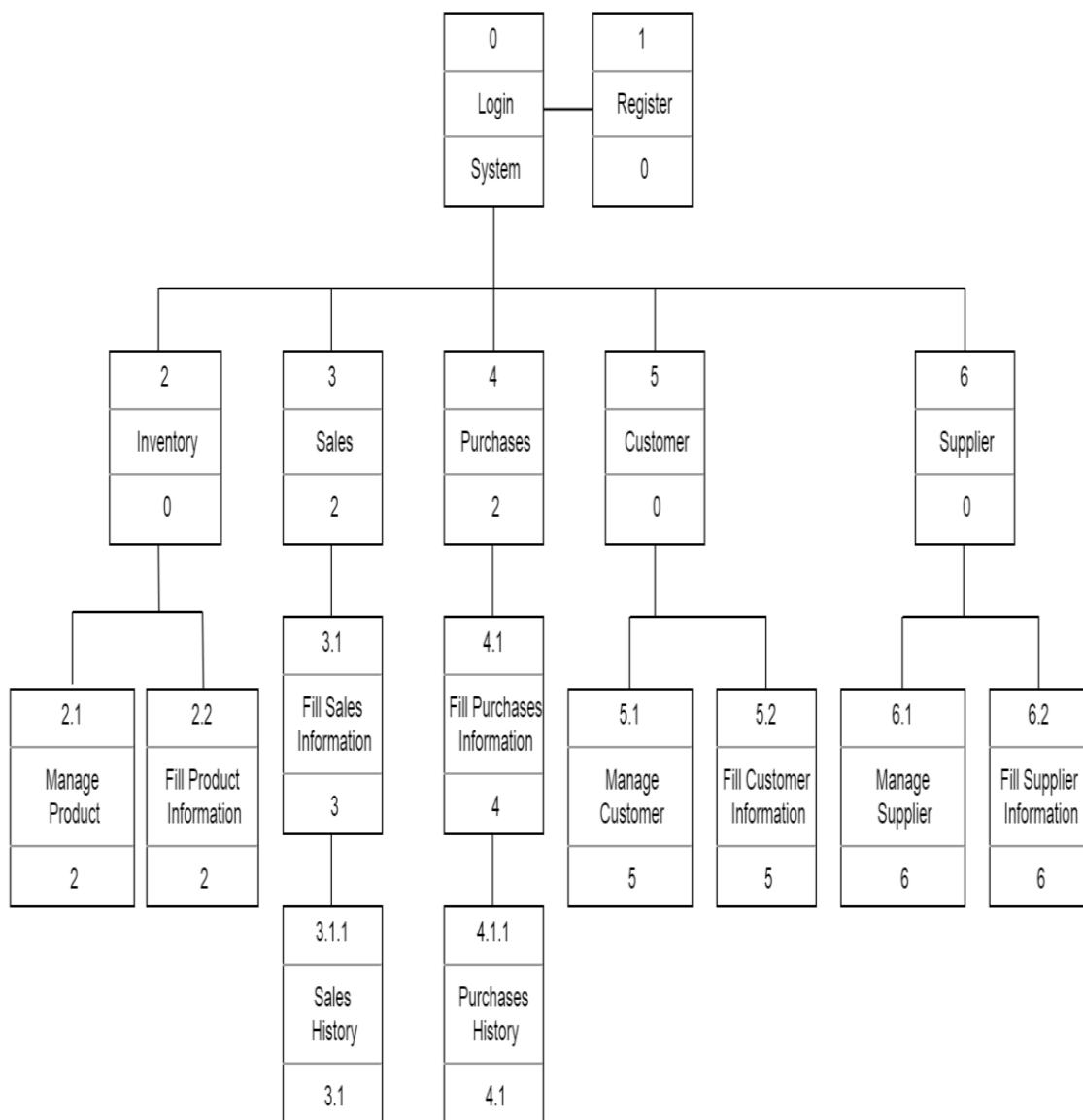


Figure 2.1 : Dialog Diagram of Mobile Based for Inventory System

Table 2.1 : User Interface Description

User Interface Name	Description	User Interface Layout
Login	In this page, stockist can access to the system by enter his/her email and password that already registered.	Refer Appendix B-2.1
Register	Stockist need to register in order to access and get email and password registered.	Refer Appendix B-2.2
Product	This page display the list of products available and stockist can click the specific name of product to manage that product info and also have button to add new product.	Refer Appendix B-2.3
Update and Delete Product	In this page, stockist can manage his/her product by update the information or delete the product.	Refer Appendix B-2.4
Add Product Information	This page display the product form which stockist can add the information of his/her new products.	Refer Appendix B-2.5
Sale	In this page, stockist can search the product or can choose to view the sales history.	Refer Appendix B-2.6
Add Sales Information	This page display the sale form which stockist can add the information of his/her new sale.	Refer Appendix B-2.7
Sale History	This page display the history of all sales.	Refer Appendix B-2.8
Purchase	In this page, stockist can search the product or can choose to view the purchases history.	Refer Appendix B-2.9

Add Purchase Information	This page display the purchase form which stockist can add the information of his/her new purchase.	Refer Appendix B-2.10
Purchase History	This page display the history of all purchases.	Refer Appendix B-2.11
Customer	This page display the list of customers available and stockist can click specific name of customer to manage that customer info and also have button to add new customer.	Refer Appendix B-2.12
Update and Delete Customer	In this page, stockist can manage his/her customer by update the information or delete their customer.	Refer Appendix B-2.13
Add Customer Information	This page display the customer form which stockist can add the information of his/her new customer.	Refer Appendix B-2.14
Supplier	This page display the list of suppliers available and stockist can click specific name of supplier to manage that supplier info and also have button to add new supplier.	Refer Appendix B-2.15
Update and Delete Supplier	In this page, stockist can manage his/her supplier by update the information or delete their supplier.	Refer Appendix B-2.16
Add Supplier Information	This page display the supplier form which stockist can add the information of his/her new supplier.	Refer Appendix B-2.17

## 2.2 Hardware Interface

In developing Mobile Based for Inventory System, Android Smart Phone will be used as hardware interface to display the contents to the user.

## 2.3 Software Interface

Software interface is the most important part in developing system which the interaction between user and the system will happen. Table 2.2 shown the software used to develop Mobile Based for Inventory System interface.

Table 2.1 : Software Interface of Mobile Based for Inventory System

Name of Software	Description
Android Studio	This software is most suitable to be used and develop an interface for mobile based application and to display to the user.

### 3.0 SOFTWARE PRODUCT FEATURES

#### 3.1 Register Account

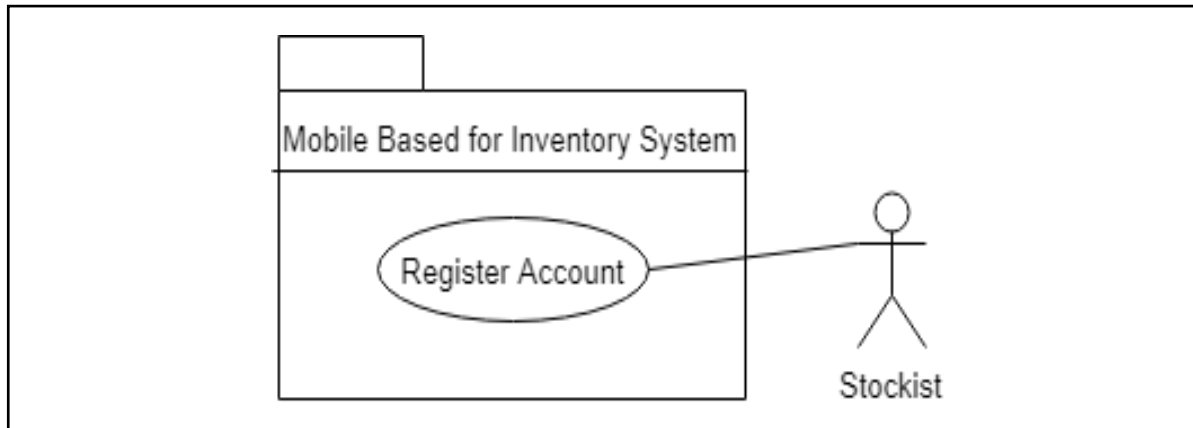


Figure 3.1 Use Case Diagram for Register Account

Table 3.1 Use Case Description for Register Account

Use Case ID	MBIS_UC1_100
Brief Description	This use case is used by stockist to register account in Mobile Based for Inventory System
Actor	Stockist
Pre-condition	System must be in state ready to access data and shall display the interface of registration page.
Basic Flow	<ol style="list-style-type: none"> <li>1. The use case start when stockist clicking the register button.</li> <li>2. Stockist enter details information in register form.</li> <li>3. Stockist click button register or <b>[A1 : Click button cancel]</b>.</li> <li>4. The system verify username and password that entered by stockist.</li> <li>5. If the verification succeeded, an account will be created, else the verification failure is executed <b>[E1 : Failed to register]</b>.</li> <li>6. The use case end.</li> </ol>
Alternative Flow	<p><b>A1 : Click button cancel.</b></p> <p><b>[MBIS_UC1_101]</b></p> <ol style="list-style-type: none"> <li>1. Stockist click button cancel.</li> <li>2. Continue with step 1 -5 from basic flow.</li> </ol>

Exception Flow	<b>E1 : Failed to register</b> <b>[MBIS_UC1_102]</b> <ol style="list-style-type: none"> <li>1. The system display an error message.</li> <li>2. Cancel the registration.</li> <li>3. Continue with step 1-3 from basic flow.</li> </ol>
Post-condition	<ol style="list-style-type: none"> <li>1. Stockist is notify for the successfully registered account.</li> <li>2. Stockist can login into the system</li> </ol>
Rules	<b>R2 : Stockist can register only one account for one email.</b>
Constraints	N/A
Sequence Diagram	Refer Appendix B-1.1 1.1.1 : Basic Flow 1.1.2 : Alternative Flow 1.1.3 : Exception Flow
Interface	Refer Appendix B-2.1

### 3.2 Manage Inventory

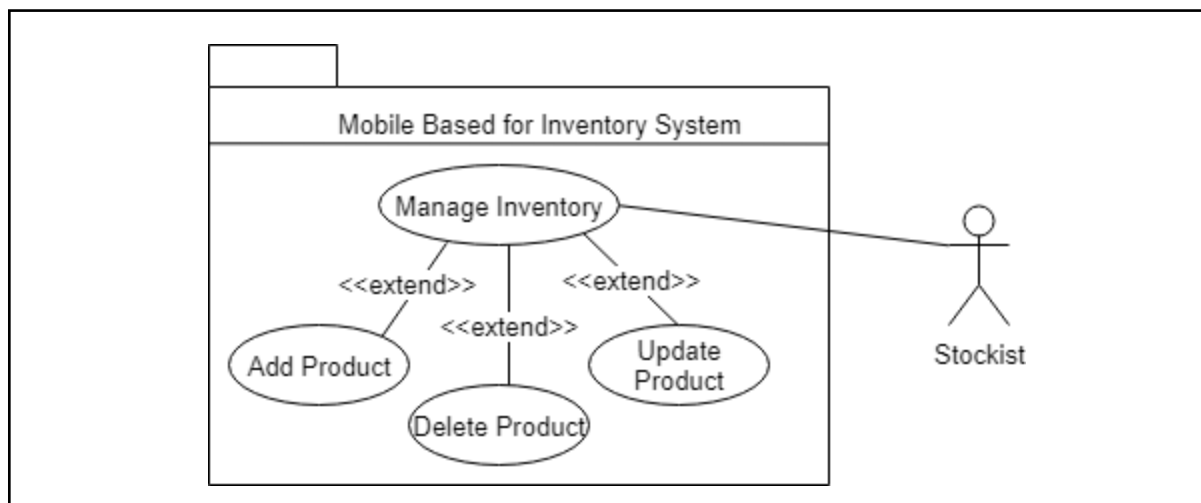


Figure 3.2 Use Case Diagram for Manage Inventory

Table 3.2 Use Case Description for Manage Inventory

Use Case ID	MBIS_UC2_200
Brief Description	This use case is used by stockist to manage their inventory by view add, update and delete the products information.
Actor	Stockist
Pre-condition	<ol style="list-style-type: none"> <li>1. System must be in state ready to access and save data.</li> <li>2. System display the interface of manage inventory.</li> </ol>
Basic Flow	<p><b>B1 : Add product</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist clicking button add product.</li> <li>2. Stockist able to insert barcode number manually.</li> <li>3. Stockist enter the details information of the product.</li> <li>4. Stockist click button save or [<b>A1 : Click button cancel</b>].</li> <li>5. The system verifies the product information [<b>E1 : Incomplete fill up information</b>].</li> <li>6. The system will save the product details data.</li> <li>7. The use case ends.</li> </ol> <p><b>B2 : Update product</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist clicking specific product.</li> <li>2. The system display the product information.</li> <li>3. Stockist update the information of that specific product by entering new data.</li> <li>4. Stockist click button save or [<b>A2 : Click button delete</b>].</li> <li>5. The system display confirmation message.</li> <li>6. The system commit the data into the database.</li> <li>7. The use case ends.</li> </ol>
Alternative Flow	<p><b>A1 : Click button cancel</b> [MBIS_UC2_201]</p> <ol style="list-style-type: none"> <li>1. Stockist click button cancel.</li> <li>2. Continue with step 1-7 from basic flow.</li> </ol> <p><b>A2 : Click button delete</b> [MBIS_UC2_202]</p>

	<ol style="list-style-type: none"> <li>1. Stockist click button delete.</li> <li>2. Display confirmation message.</li> <li>3. The system deleted that specific product information from database.</li> <li>4. The use case ends.</li> </ol>
Exception Flow	<p><b>E1 : Incomplete fill up information</b> [MBIS_UC2_203]</p> <ol style="list-style-type: none"> <li>1. The system display an alert message.</li> <li>2. Continue with step 4-7 from basic flow.</li> </ol>
Post-condition	<ol style="list-style-type: none"> <li>1. The information's was recorded in the database and will be display in the system.</li> <li>2. Stockist able to make sales and purchases.</li> </ol>
Rules	Sales price must be greater than purchases price.
Constraints	N/A
Sequence Diagram	<p>Refer Appendix B-1.2</p> <p>1.2.1 : Basic Flow</p> <p>1.2.2 : Alternative Flow</p> <p>1.2.3 : Exception Flow</p>
Interface	Refer Appendix B-2.2

### 3.3 Manage Sales

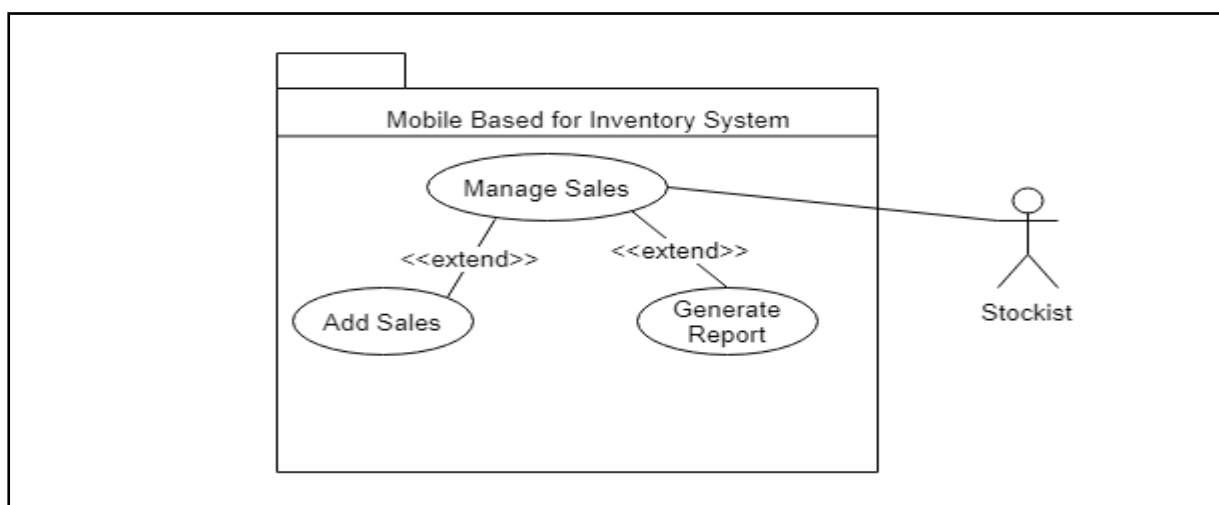


Figure 3.3 Use Case Diagram for Manage Sales



Table 3.3 Use Case Description for Manage Sales

Use Case ID	MBIS_UC3_300
Brief Description	This use case is used by stockist to manage their sales by search the product and insert the information of products that have been sold. In other hand, stockist also can view the history of product sales.
Actor	Stockist
Pre-condition	<ol style="list-style-type: none"> <li>1. System must be in state ready to access and save data.</li> <li>2. System display the interface of manage sales.</li> </ol>
Basic Flow	<p><b>B1 : Add sales</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist clicking on specific product.</li> <li>2. Stockist click on specific product to make sale or [<b>A1 : Insert product barcode or name in search field</b>].</li> <li>3. The system display sales form.</li> <li>4. Stockist enter the details information of the sale product.</li> <li>5. Stockist click button save or [<b>A2 : Click button cancel</b>].</li> <li>6. The system verifies the product sales information. [<b>E1 : Incomplete fill up information</b>] [<b>E2 : Product achieve critical quantity</b>].</li> <li>7. The system will save the product sales data.</li> <li>8. The use case ends.</li> </ol> <p><b>B3 : Generate report</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist clicking button view history.</li> <li>2. The system display sale history of all products.</li> <li>3. Stockist click button cancel.</li> <li>4. The use case ends.</li> </ol>
Alternative Flow	<p><b>A1 : Insert product barcode or name in search field</b></p> <p>[MBIS_UC3_301]</p> <ol style="list-style-type: none"> <li>1. Stockist insert product barcode or name in search field.</li> <li>2. The system display product searched.</li> <li>3. Stockist click on specific product to make sale.</li> </ol>

	<p>4. Continue with step 3-8 from basic flow.</p> <p><b>A2 : Click button cancel</b></p> <p>[MBIS_UC3_302]</p> <p>5. Stockist click button cancel.</p> <p>6. Continue with step 1-8 from basic flow.</p>
Exception Flow	<p><b>E1 : Incomplete fill up information</b></p> <p>[MBIS_UC3_303]</p> <p>1. The system display an alert message.</p> <p>2. Continue with step 4-8 from basic flow.</p> <p><b>E2 : Product achieve critical quantity</b></p> <p>[MBIS_UC3_304]</p> <p>3. The system display an alert message.</p> <p>4. Stockist click button “OK”.</p> <p>5. Continue with step 7-8 from basic flow.</p>
Post-condition	The information's was recorded in the database and will be display in the system.
Rules	The process cannot proceed if the product out of stock.
Constraints	N/A
Sequence Diagram	<p>Refer Appendix B-1.3</p> <p>1.3.1 : Basic Flow</p> <p>1.3.2 : Alternative Flow</p> <p>1.3.3 : Exception Flow</p>
Interface	Refer Appendix B-2.3

### 3.4 Manage Purchases

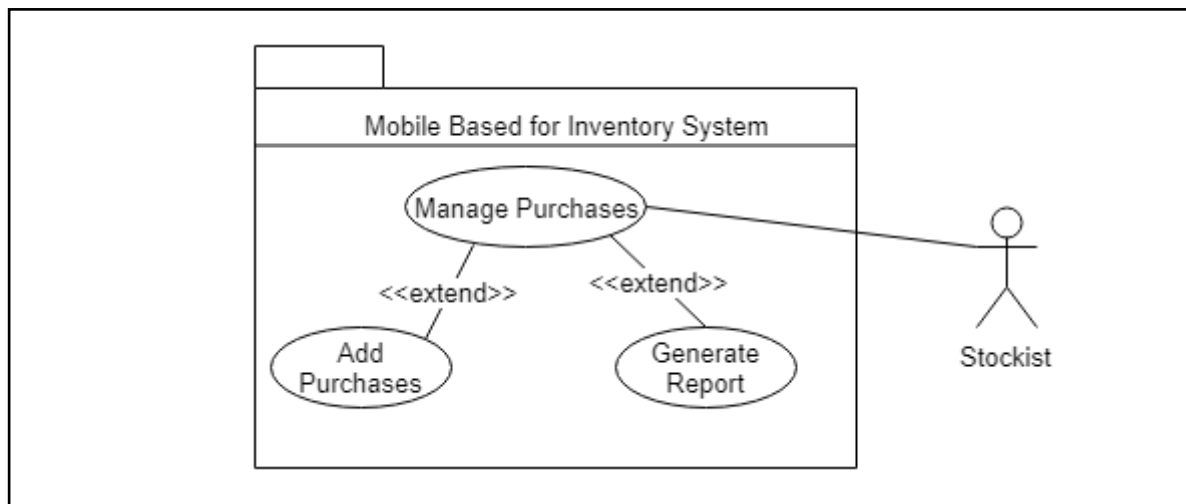


Figure 3.4 Use Case Diagram for Manage Purchases

Table 3.4 Use Case Description for Manage Purchases

Use Case ID	MBIS_UC4_400
Brief Description	This use case is used by stockist to manage their purchases by search the product and insert the information of products that have been purchased. In other hand, stockist also can view the history report of product purchases.
Actor	Stockist
Pre-condition	<ol style="list-style-type: none"> <li>1. System must be in state ready to access and save data.</li> <li>2. System display the interface of manage purchases.</li> </ol>
Basic Flow	<p><b>B1 : Add purchases</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist clicking button add purchases.</li> <li>2. Stockist click on specific product to make purchase or <b>[A1 : Insert product barcode or name in search field]</b>.</li> <li>3. The system display purchases form.</li> <li>4. Stockist enter the details information of the purchases product.</li> <li>5. Stockist click button save or <b>[A2 : Click button cancel]</b>.</li> <li>6. The system verifies the product purchases information. <b>[E1 : Incomplete fill up information]</b>.</li> </ol>

	<p>7. The system will save the product sales data.</p> <p>8. The use case ends.</p> <p><b>B3 : Generate report</b></p> <p>9. The use case start when stockist clicking button view history.</p> <p>10. The system display history of all products.</p> <p>11. Stockist click button cancel.</p> <p>12. The use case ends.</p>
Alternative Flow	<p><b>A1 : Insert product barcode or name in search field</b></p> <p>[MBIS_UC3_401]</p> <ol style="list-style-type: none"> <li>1. Stockist insert product barcode or name in search field.</li> <li>2. The system display product searched.</li> <li>3. Stockist click on specific product to make purchase.</li> <li>4. Continue with step 3-8 from basic flow.</li> </ol> <p><b>A2 : Click button cancel</b></p> <p>[MBIS_UC4_402]</p> <ol style="list-style-type: none"> <li>1. Stockist click button cancel.</li> <li>2. Continue with step 1-8 from basic flow.</li> </ol>
Exception Flow	<p><b>E1 : Incomplete fill up information</b></p> <p>[MBIS_UC4_403]</p> <ol style="list-style-type: none"> <li>1. The system display an alert message.</li> <li>2. Continue with step 4-8 from basic flow.</li> </ol>
Post-condition	The information's was recorded in the database and will be display in the system.
Rules	N/A
Constraints	N/A
Sequence Diagram	<p>Refer Appendix B-1.4</p> <p>1.4.1 : Basic Flow</p> <p>1.4.2 : Alternative Flow</p> <p>1.4.3 : Exception Flow</p>
Interface	Refer Appendix B-2.4

### 3.5 Manage Customer

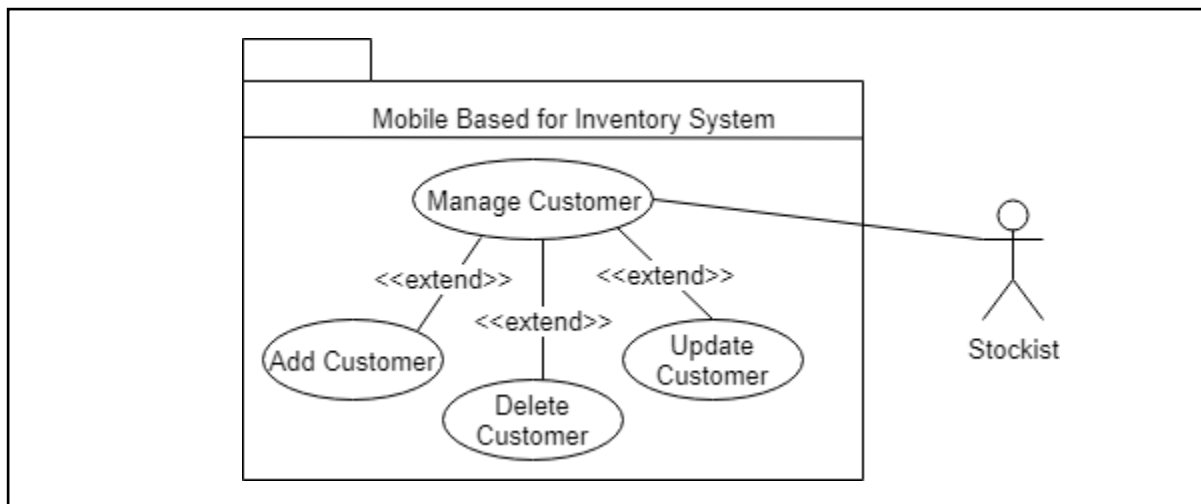


Figure 3.5 Use Case Diagram for Manage Customer

Table 3.5 Use Case Description for Manage Customer

Use Case ID	MBIS_UC5_500
Brief Description	This use case is used by stockist to manage their customer by view, add, update and delete the customer information.
Actor	Stockist
Pre-condition	<ol style="list-style-type: none"> <li>1. System must be in state ready to access and save data.</li> <li>2. System display the interface of manage customer.</li> </ol>
Basic Flow	<p><b>B1 : Add customer</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist clicking button add customer.</li> <li>2. The system display details form of customer.</li> <li>3. Stockist enter the details information of customer.</li> <li>4. Stockist click button save or <b>[A1 : Click button cancel]</b>.</li> <li>5. The system verifies the customer information <b>[E1 : Incomplete fill up information]</b>.</li> <li>6. The system will save the customer data.</li> <li>7. The use case ends.</li> </ol>

	<p><b>B2 : Update Customer</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist click on specific customer</li> <li>2. The system display customer information.</li> <li>3. Stockist update the information of that specific customer by entering new data.</li> <li>4. Stockist click button save or [<b>A2 : Click button delete</b>].</li> <li>5. The system display confirmation message.</li> <li>6. The system commit the data into the database.</li> <li>7. The use case ends.</li> </ol>
Alternative Flow	<p><b>A1 : Click button cancel</b> [MBIS_UC5_501]</p> <ol style="list-style-type: none"> <li>1. Stockist click button cancel.</li> <li>2. Continue with step 1-6 from basic flow.</li> </ol> <p><b>A2 : Click button delete</b> [MBIS_UC5_502]</p> <ol style="list-style-type: none"> <li>1. Stockist click button delete.</li> <li>2. Display confirmation message.</li> <li>3. The system deleted that specific customer information from database.</li> <li>4. The use case ends.</li> </ol>
Exception Flow	<p><b>E1 : Incomplete fill up information</b> [MBIS_UC5_503]</p> <ol style="list-style-type: none"> <li>3. The system display an alert message.</li> <li>4. Continue with step 3-6 from basic flow.</li> </ol>
Post-condition	<ol style="list-style-type: none"> <li>1. The information's was recorded in the database and will be display in the system.</li> <li>2. Stockist able to add customer in make sales.</li> </ol>
Rules	N/A
Constraints	N/A
Sequence Diagram	Refer Appendix B-1.5 1.5.1 : Basic Flow

	1.5.2 : Alternative Flow 1.5.3 : Exception Flow
Interface	Refer Appendix B-2.5

### 3.6 Manage Supplier

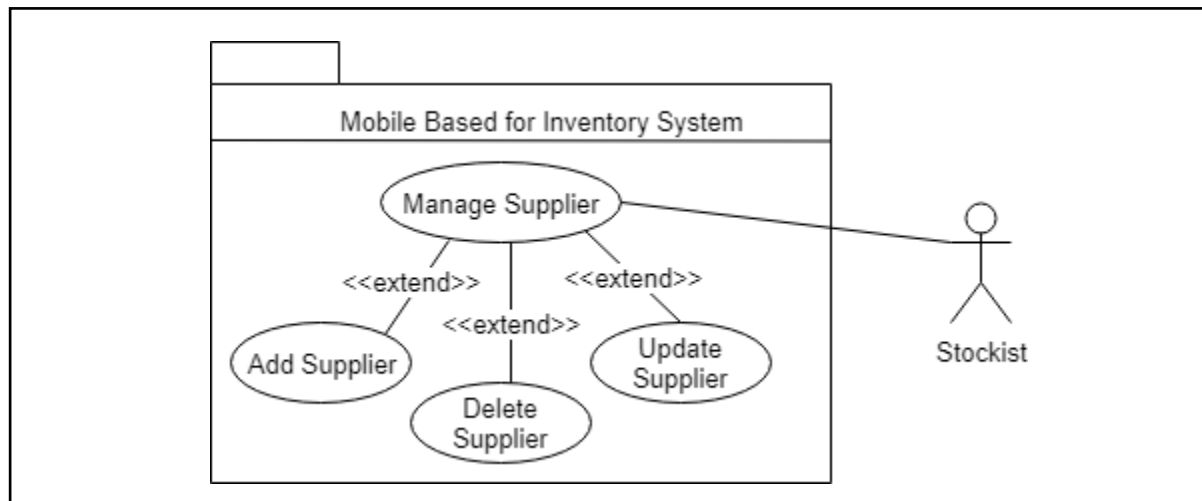


Figure 3.6 Use Case Diagram for Manage Supplier

Table 3.6 Use Case Description for Manage Supplier

Use Case ID	MBIS_UC6_600
Brief Description	This use case is used by stockist to manage their supplier by view, add, update and delete the supplier information.
Actor	Stockist
Pre-condition	<ol style="list-style-type: none"> <li>1. System must be in state ready to access and save data.</li> <li>2. System display the interface of manage supplier.</li> </ol>
Basic Flow	<p><b>B1 : Add supplier</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist clicking button add supplier.</li> <li>2. The system display details form of supplier.</li> <li>3. Stockist enter the details information of supplier.</li> <li>4. Stockist click button save or [<b>A1 : Click button cancel</b>].</li> <li>5. The system verifies the supplier information [<b>E1 : Incomplete fill up information</b>].</li> </ol>

	<ol style="list-style-type: none"> <li>6. The system will save the supplier data.</li> <li>7. The use case ends.</li> </ol> <p><b>B2 : Update supplier</b></p> <ol style="list-style-type: none"> <li>1. The use case start when stockist click on specific supplier.</li> <li>2. The system display supplier information.</li> <li>3. Stockist update the information of that specific supplier by entering new data.</li> <li>4. Stockist click button save or [<b>A2 : Click button delete</b>].</li> <li>5. The system display confirmation message.</li> <li>6. The system commit the data into the database.</li> <li>7. The use case ends.</li> </ol>
Alternative Flow	<p><b>A1 : Click button cancel</b></p> <p>[MBIS_UC6_601]</p> <ol style="list-style-type: none"> <li>1. Stockist click button cancel.</li> <li>2. Continue with step 1-4 from basic flow.</li> </ol> <p><b>A2 : Click button delete</b></p> <p>[MBIS_UC6_602]</p> <ol style="list-style-type: none"> <li>1. Stockist click button delete.</li> <li>2. Display confirmation message.</li> <li>3. The system deleted that specific supplier information from database.</li> <li>4. The use case ends.</li> </ol>
Exception Flow	<p><b>E1 : Incomplete fill up information</b></p> <p>[MBIS_UC6_603]</p> <ol style="list-style-type: none"> <li>1. The system display an alert message.</li> <li>2. Continue with step 3-6 from basic flow.</li> </ol>
Post-condition	<ol style="list-style-type: none"> <li>1. The information's was recorded in the database and will be display in the system.</li> </ol>



	2. Stockist able to add supplier in make purchases.
Rules	N/A
Constraints	N/A
Sequence Diagram	Refer Appendix B-1.6 1.6.1 : Basic Flow 1.6.2 : Alternative Flow 1.6.3 : Exception Flow
Interface	Refer Appendix B-2.3

#### 4.0 REQUIREMENTS TRACEABILITY

Table 4.1 shown the requirements traceability of Mobile Based for Inventory System.

Table 4.1 Requirement Traceability of Mobile Based for Inventory System

Requirement	Description
[MBIS_UC1_100]	Use case register account : Stockist able to register an account fill up all the information needed in order to get username and password.
[MBIS_UC1_101]	Click button cancel : If stockist able to click button cancel if he/she didn't want to register.
[MBIS_UC1_102]	Failed to register : The system able failed the registration if system failed verifies the register information.
[MBIS_UC2_200]	Use case manage inventory : Stockist able to manage inventory by fill up all the information needed, view, update, delete and generate report of products.
[MBIS_UC2_201]	Click button cancel :

	Stockist able to click button cancel if he/she didn't want to save the information of product.
[MBIS_UC2_202]	Click button delete : Stockist able to delete the product if he/she don't have that product anymore
[MBIS_UC2_203]	Incomplete fill up information: The system checks all the information whether all fill up or not in adding new products information.
[MBIS_UC3_300]	Use case manage sales : Stockist able to manage sales by fill up all the information needed, view, update, delete and generate report of sales.
[MBIS_UC3_301]	Insert product barcode or name in search field : Stockist able to search specific product to make sale.
[MBIS_UC3_302]	Click button cancel : Stockist able to click button cancel if he/she didn't want to save the information of sales.
[MBIS_UC3_303]	Incomplete fill up information : The system checks all the information whether all fill up or not in adding new sales information.
[MBIS_UC3_304]	Product achieve critical quantity : The system notify if the quantity after make new sale below than 3.
[MBIS_UC4_400]	Use case manage purchases : Stockist able to manage purchases by fill up all the information needed, view, update, delete and generate report of purchases.
[MBIS_UC3_401]	Insert product barcode or name in search field : Stockist able to search specific product to make purchase.
[MBIS_UC4_402]	Click button cancel : Stockist able to click button cancel if he/she didn't want to save the information of purchases.

[MBIS_UC4_403]	<p>Incomplete fill up information:</p> <p>The system checks all the information whether all fill up or not in adding new purchases information.</p>
[MBIS_UC5_500]	<p>Use case manage customer :</p> <p>Stockist able to manage customer by fill up all the information needed, view, update, delete and generate report of customers.</p>
[MBIS_UC5_501]	<p>Click button cancel :</p> <p>Stockist able to click button cancel if he/she didn't want to save the information of customer.</p>
[MBIS_UC5_502]	<p>Click button delete :</p> <p>Stockist able to delete the customer if he/she don't have that customer anymore.</p>
[MBIS_UC5_503]	<p>Incomplete fill up information:</p> <p>The system checks all the information whether all fill up or not in adding new customer information.</p>
[MBIS_UC6_600]	<p>Use case manage supplier :</p> <p>Stockist able to manage supplier by fill up all the information needed, view, update, delete and generate report of suppliers.</p>
[MBIS_UC6_601]	<p>Click button cancel :</p> <p>Stockist able to click button cancel if he/she didn't want to save the information of supplier.</p>
[MBIS_UC6_602]	<p>Click button delete :</p> <p>Stockist able to delete the product if he/she don't have that supplier.</p>
[MBIS_UC6_603]	<p>Incomplete fill up information :</p> <p>The system checks all the information whether all fill up or not in adding new supplier information.</p>

**5.0 SYSTEM REQUIREMENTS APPROVAL**

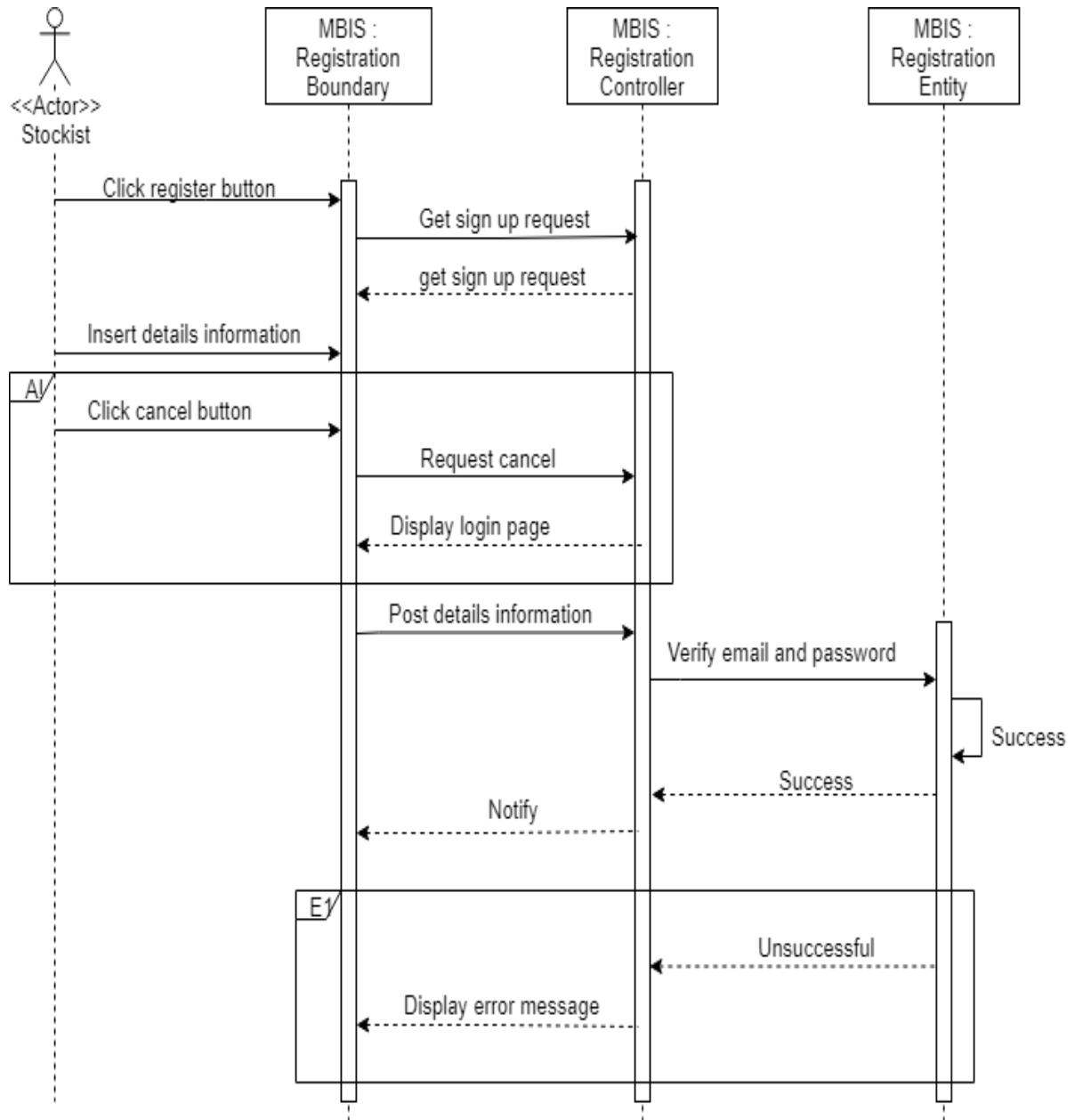
	<b>Name</b>	<b>Date</b>
<b>Verified by:</b>  _____  Developer	CHE NORMADIAA BINTI IBRAHIM	
<b>Approved by:</b>  _____  Client		

## **Appendix B-1**

### **Sequence Diagrams**

**Refer Use Case ID : MBIS\_UC1\_100**

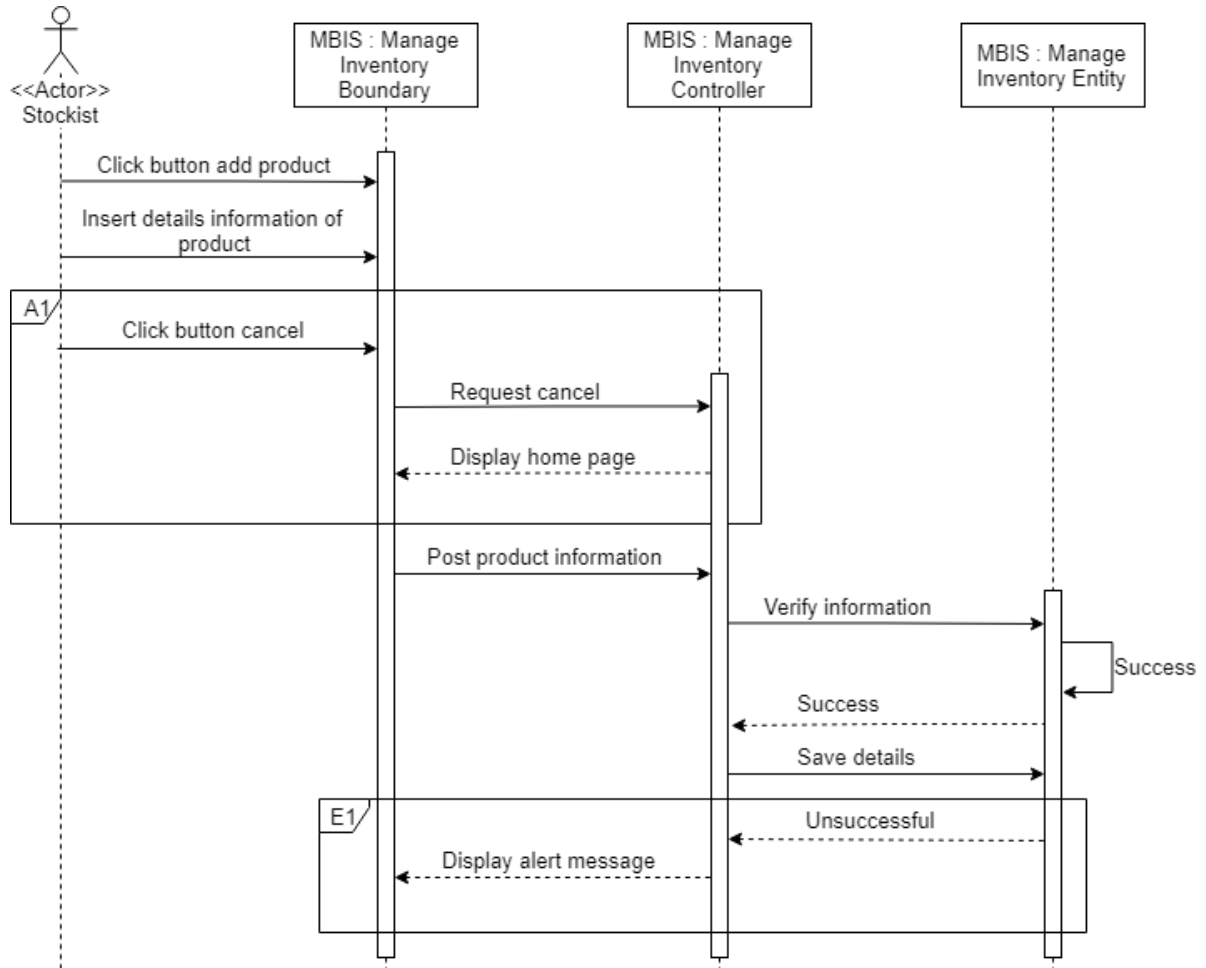
1. Sequence Diagram for Register Account



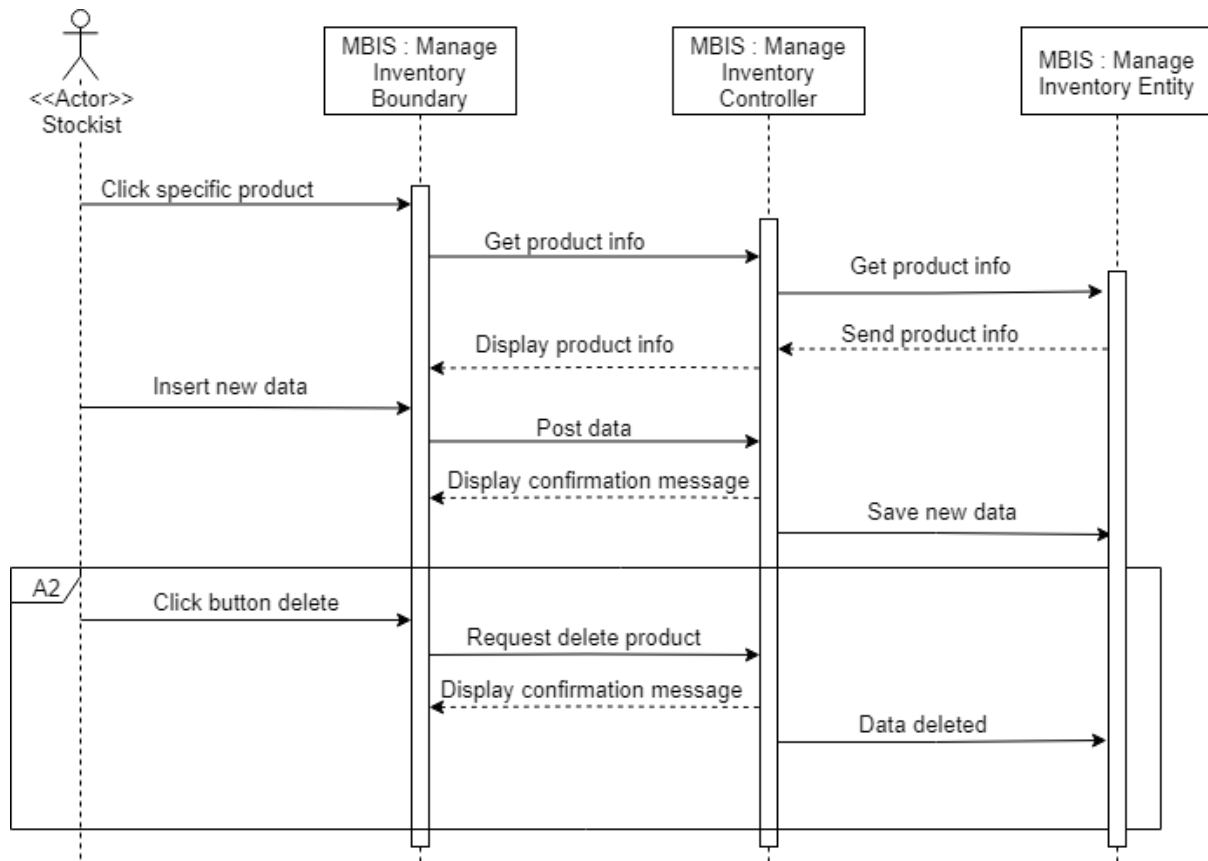
Appendix B-1.1.1: Sequence diagram for Register Account

**Refer Use Case ID : MBIS\_UC2\_200**

2. Sequence Diagram for Manage Inventory



Appendix B-1.2.1: Sequence diagram for Manage Inventory (Add Product)

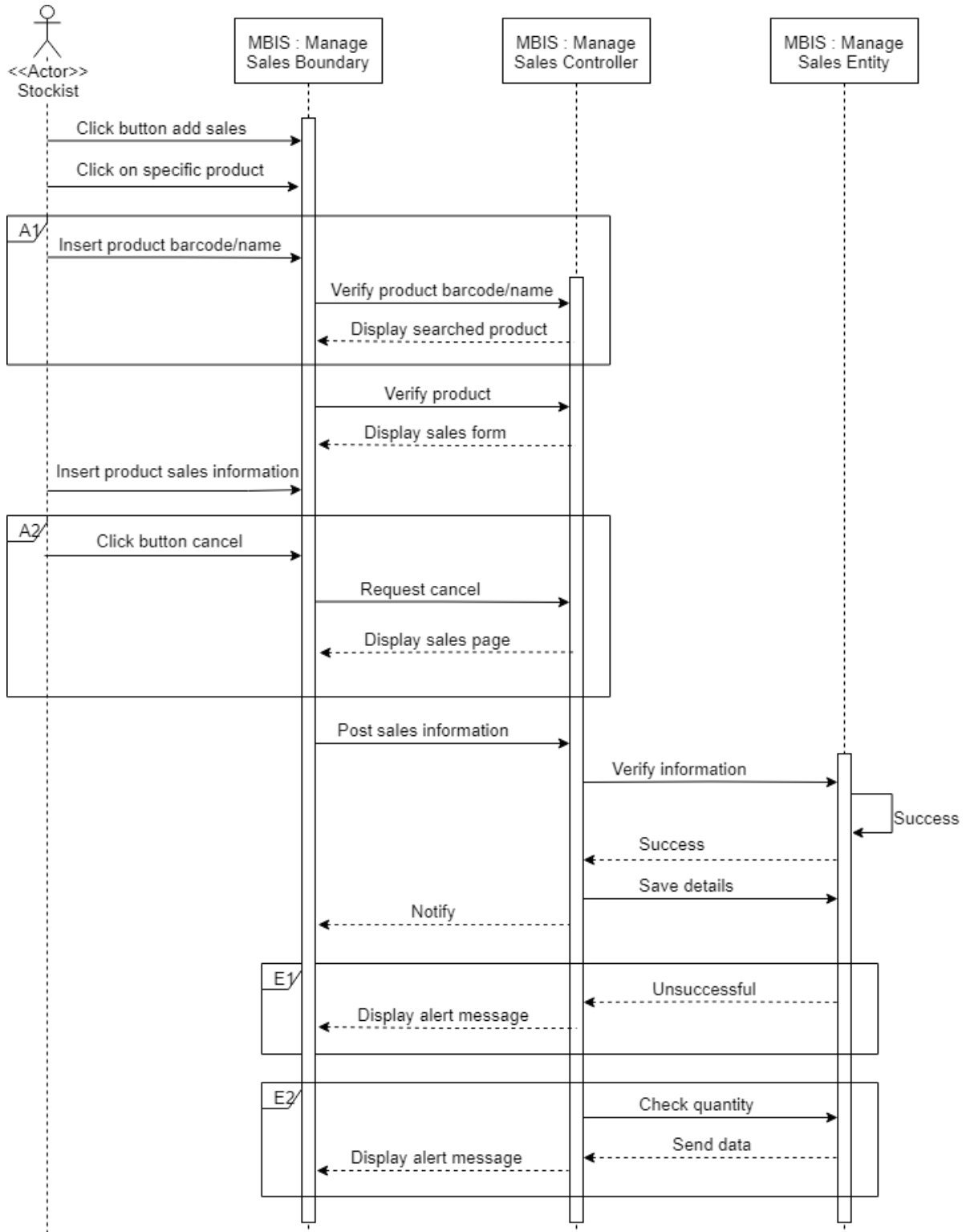


Appendix B-1.2.2: Sequence diagram for Manage Inventory (Update Product)

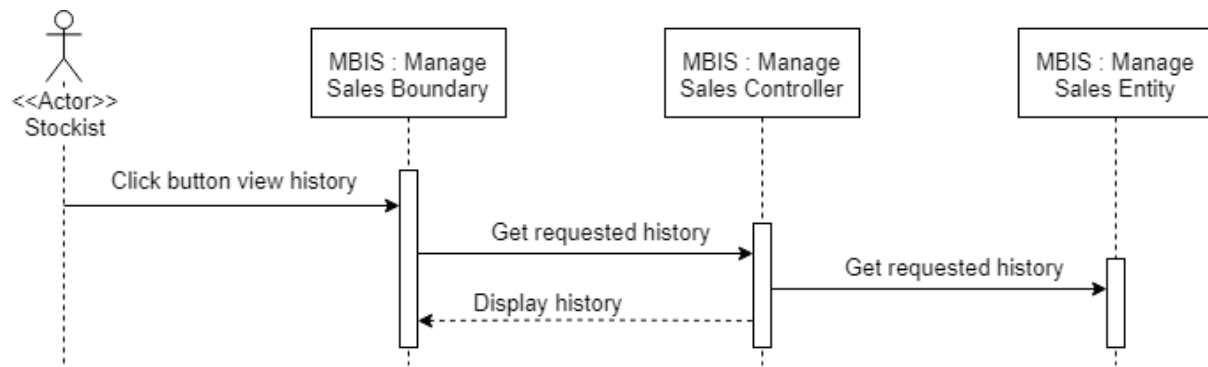


Refer Use Case ID : MBIS\_UC3\_300

3. Sequence Diagram for Manage Sales



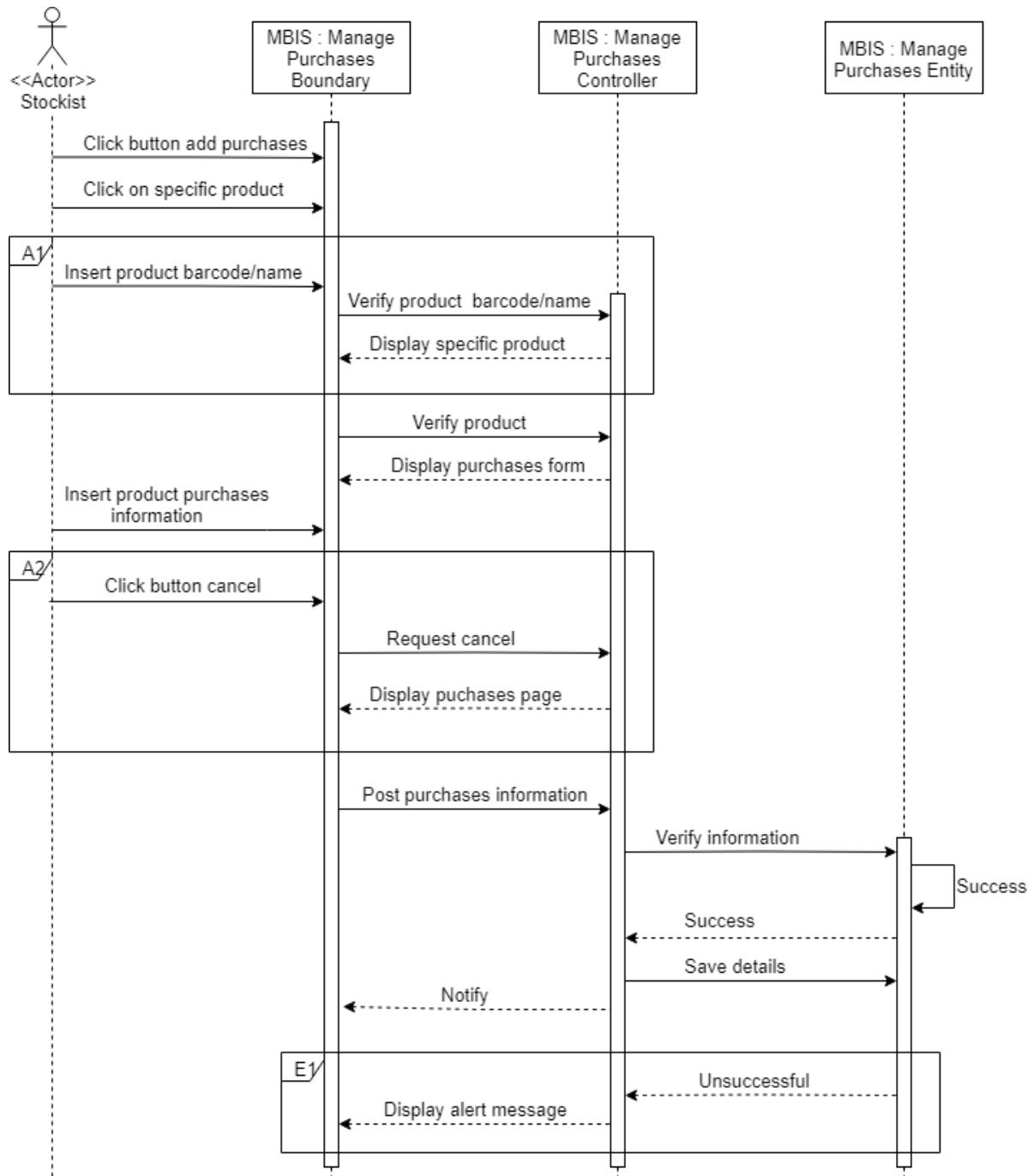
Appendix B-1.3.1: Sequence diagram for Manage Sales (Add Sales)



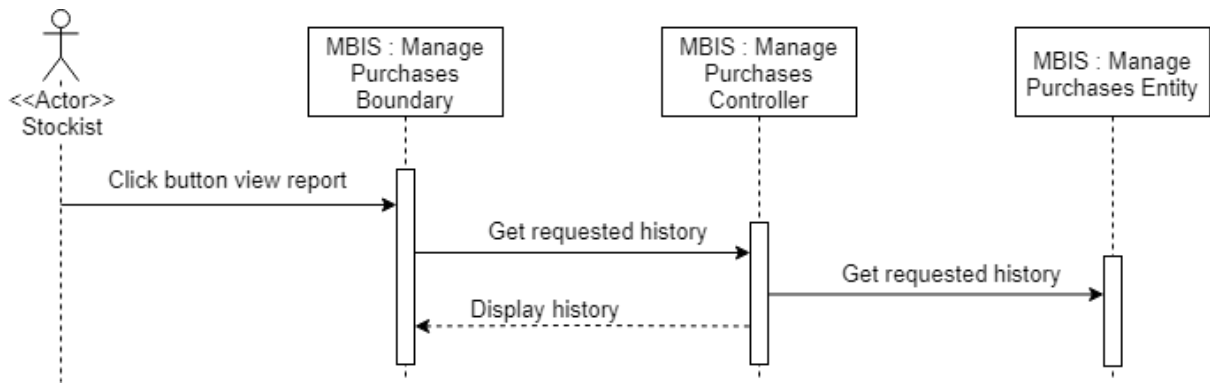
Appendix B-1.3.2: Sequence diagram for Manage Sales (Generate Report)

Refer Use Case ID : MBIS\_UC4\_400

4. Sequence Diagram for Manage Purchases



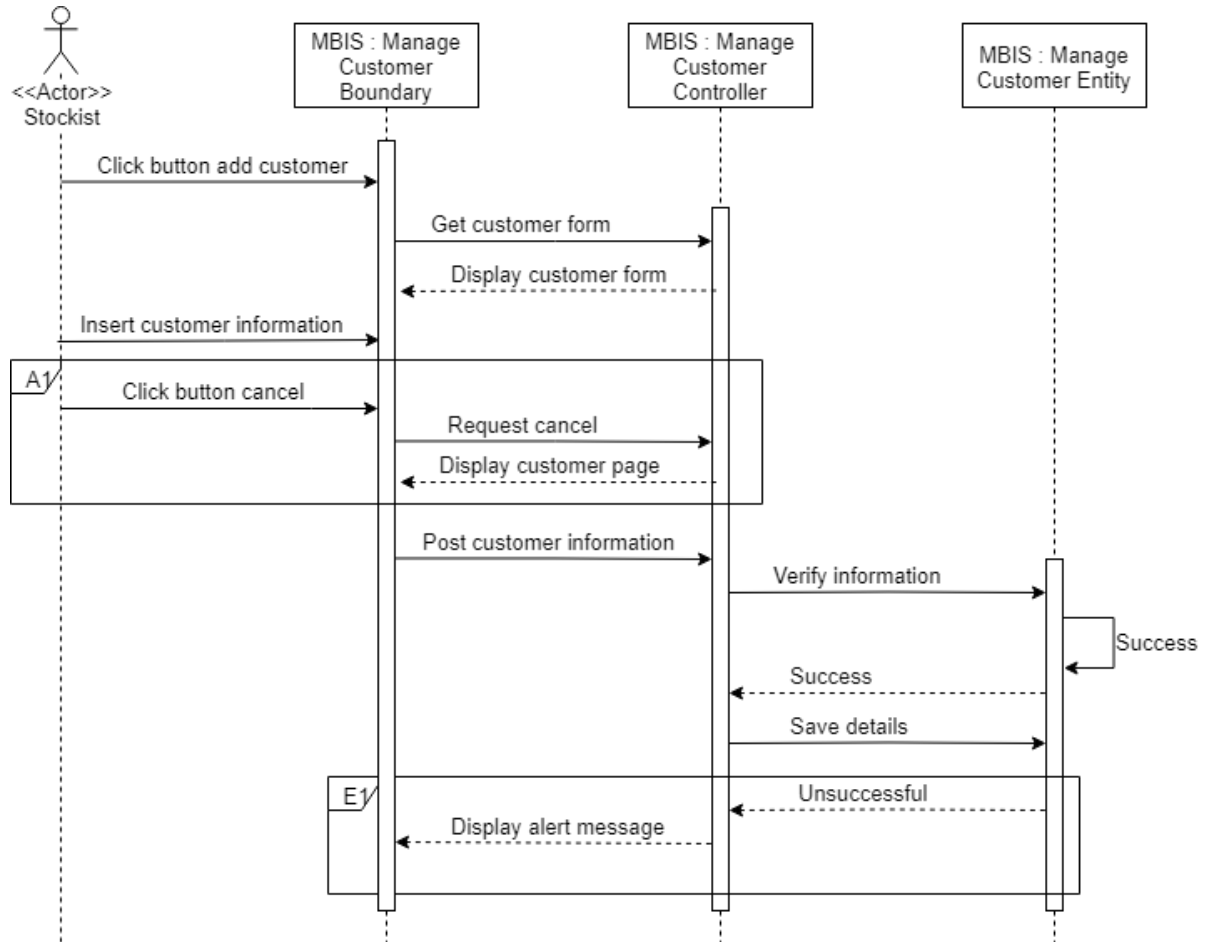
Appendix B-1.4.1: Sequence diagram for Manage Purchases (Add Purchases )



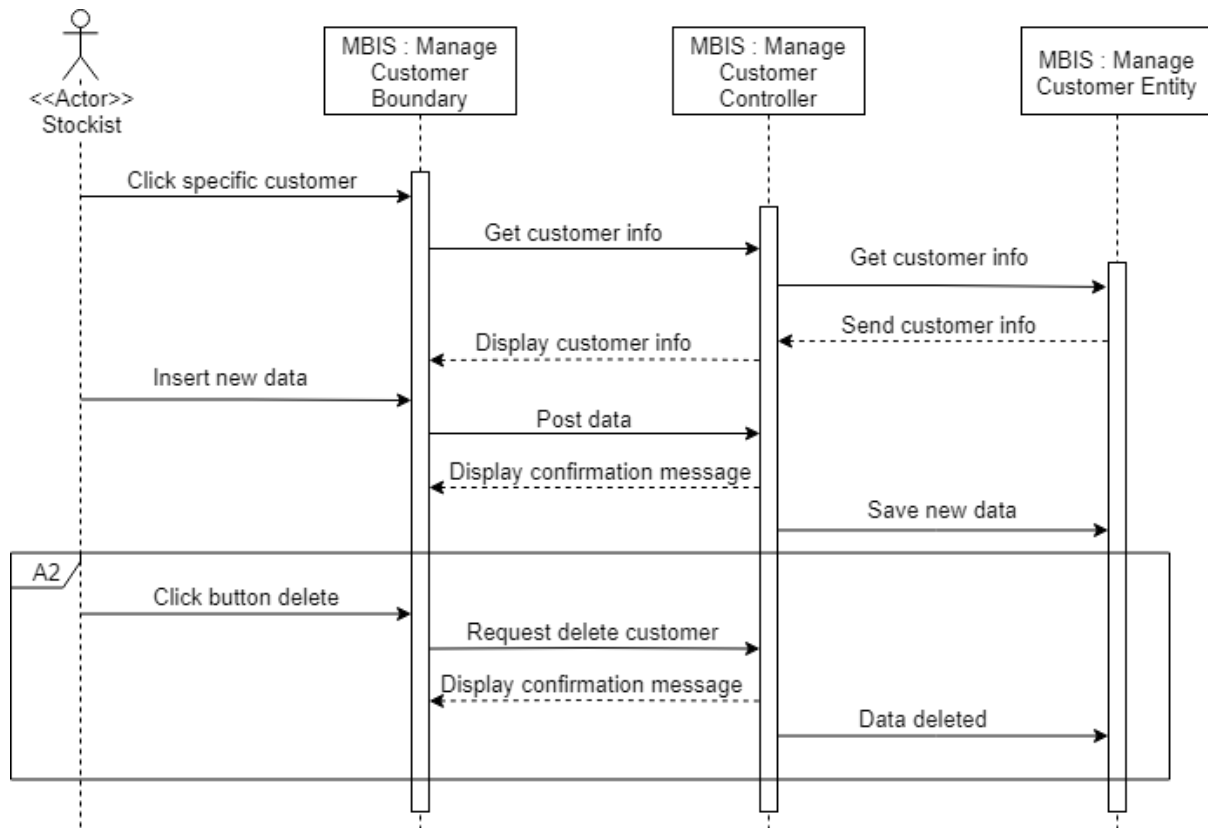
Appendix B-1.4.2: Sequence diagram for Manage Purchases (Generate Report)

**Refer Use Case ID : MBIS\_UC5\_500**

5. Sequence Diagram for Manage Customer



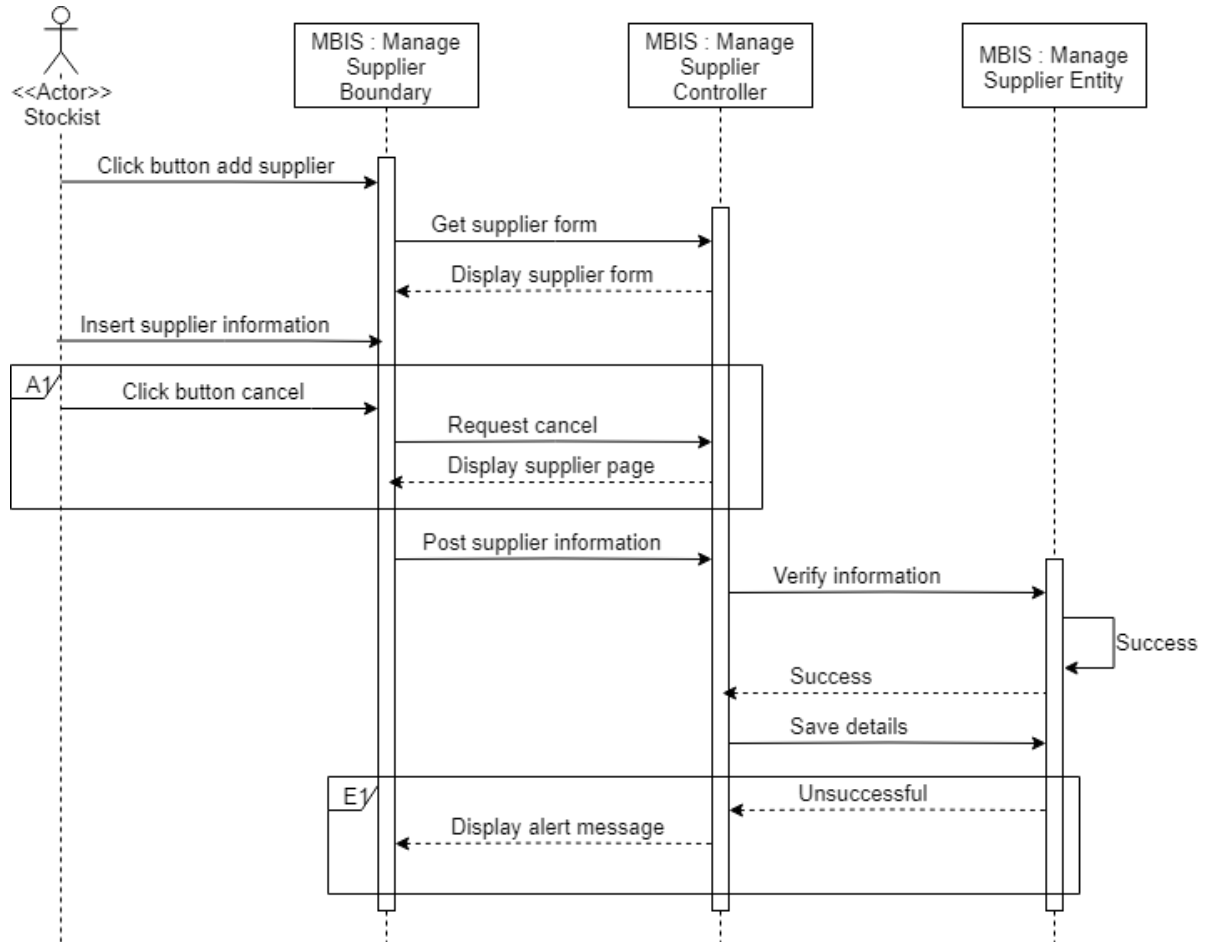
Appendix B-1.5.1: Sequence diagram for Manage Customer (Add Customer )



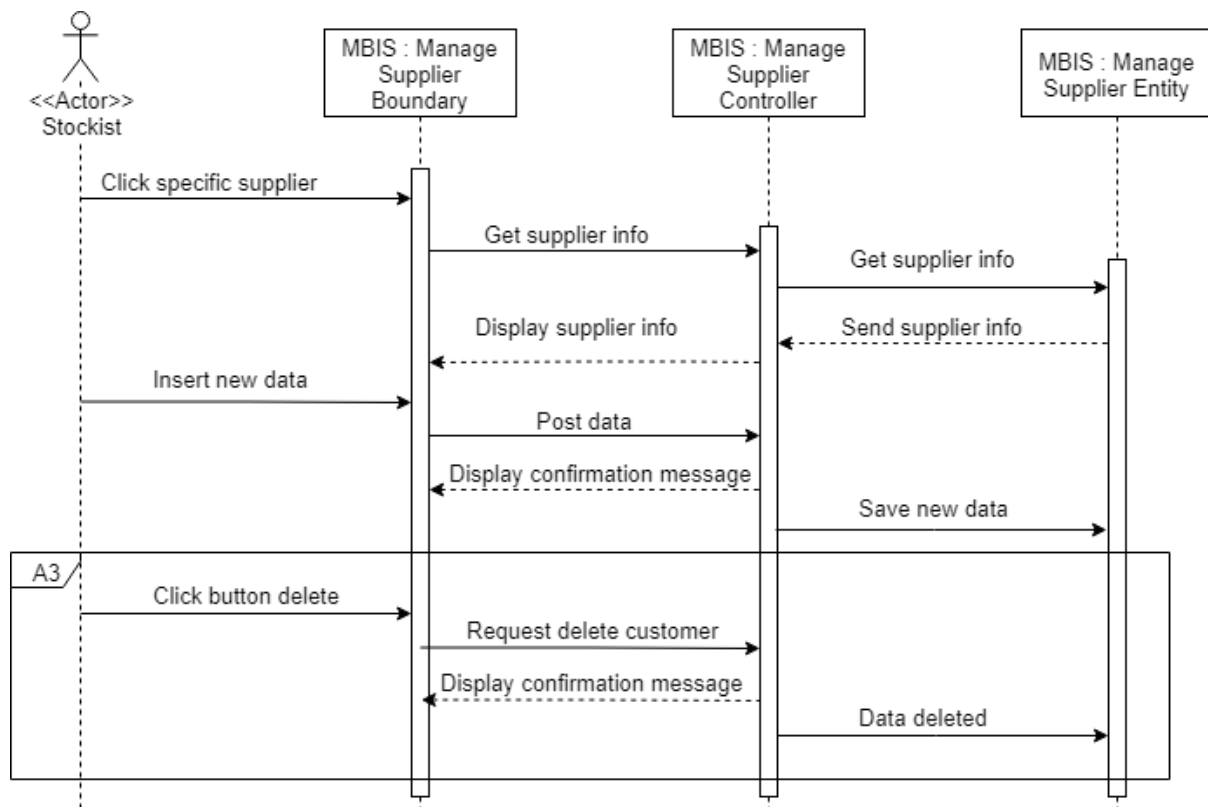
Appendix B-1.5.2: Sequence diagram for Manage Customer (Update Customer)

**Refer Use Case ID : MBIS\_UC6\_600**

6. Sequence Diagram for Manage Supplier



Appendix B-1.6.1: Sequence diagram for Manage Supplier (Add Supplier )

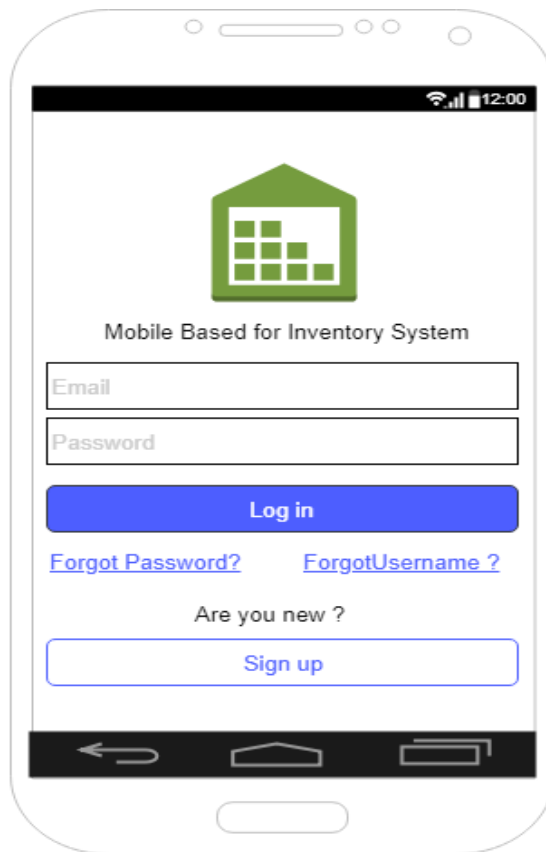


Appendix B-1.6.2: Sequence diagram for Manage Supplier (Update Supplier)

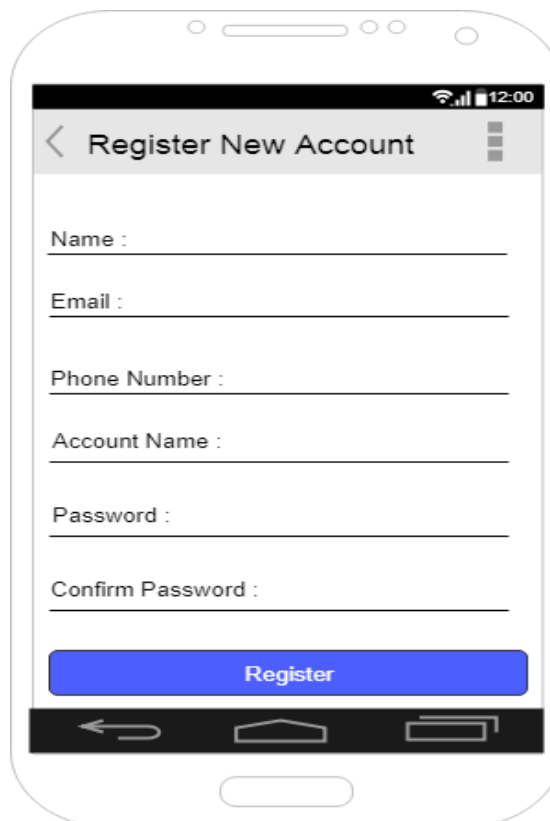


## **Appendix B-2**

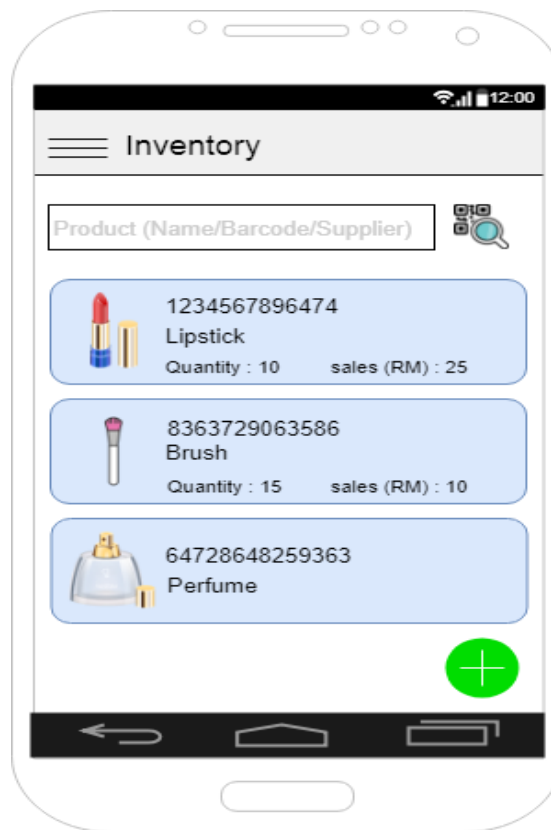
### **User Interfaces**



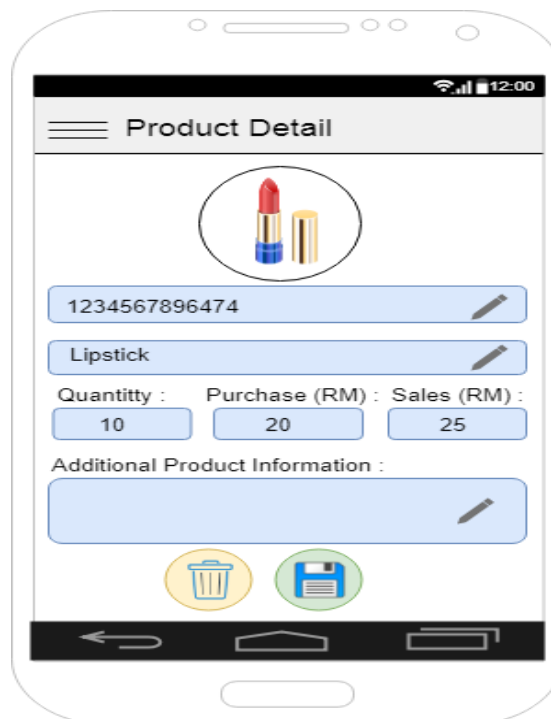
Appendix B-2.1: Login page



Appendix B-2.2 : Register page



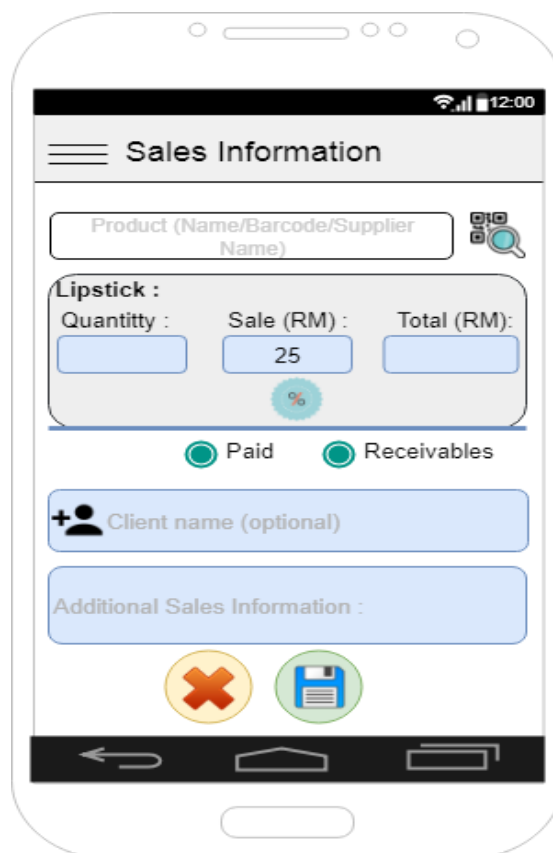
Appendix B-2.3 : Prodduct page



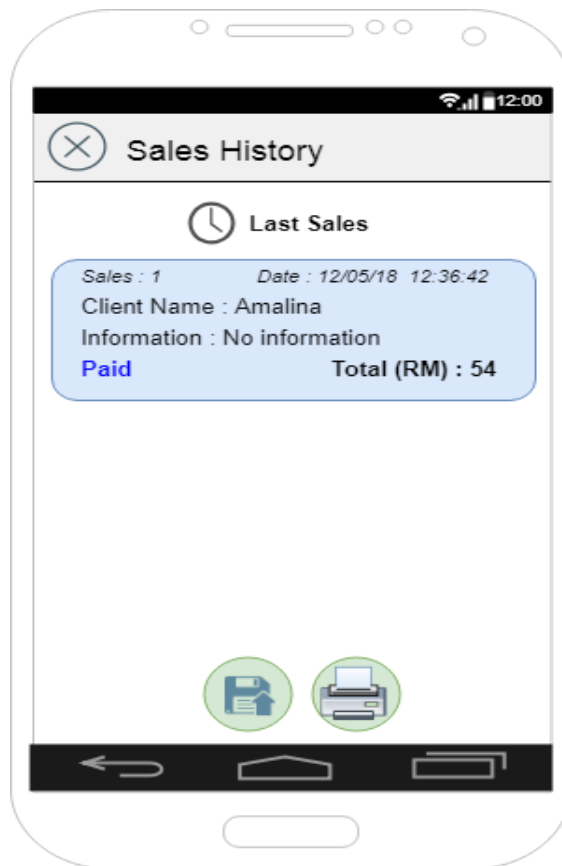
Appendix B-2.4 : Add Product Information page



Appendix B-2.6 : Sale page



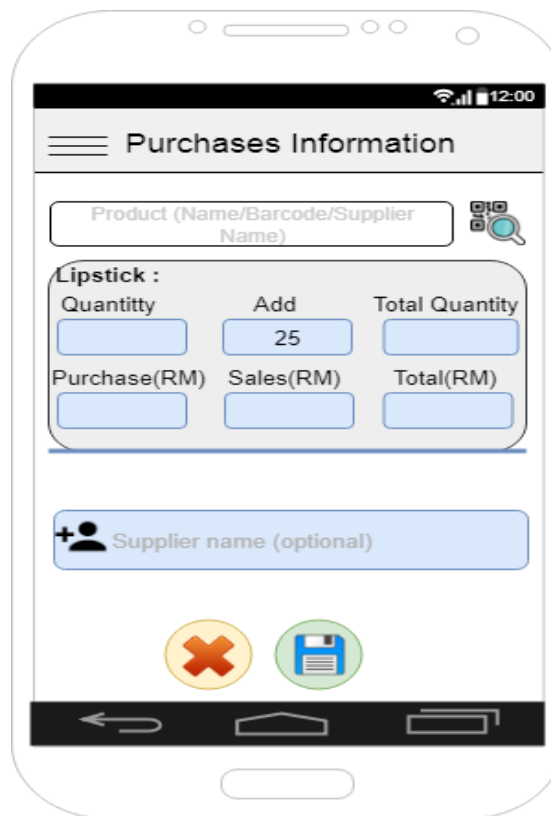
Appendix B-2.7 : Add Sale Information page



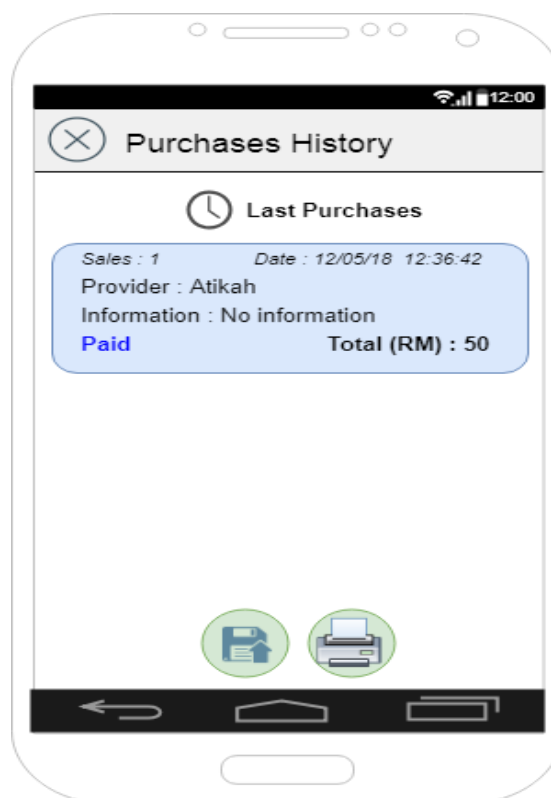
Appendix B-2.8 : Sale History page



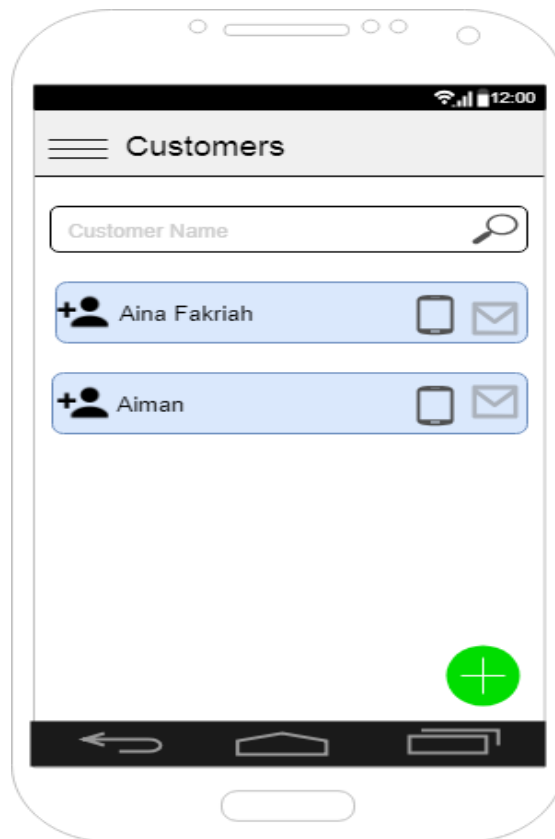
Appendix B-2.9 : Purchases page



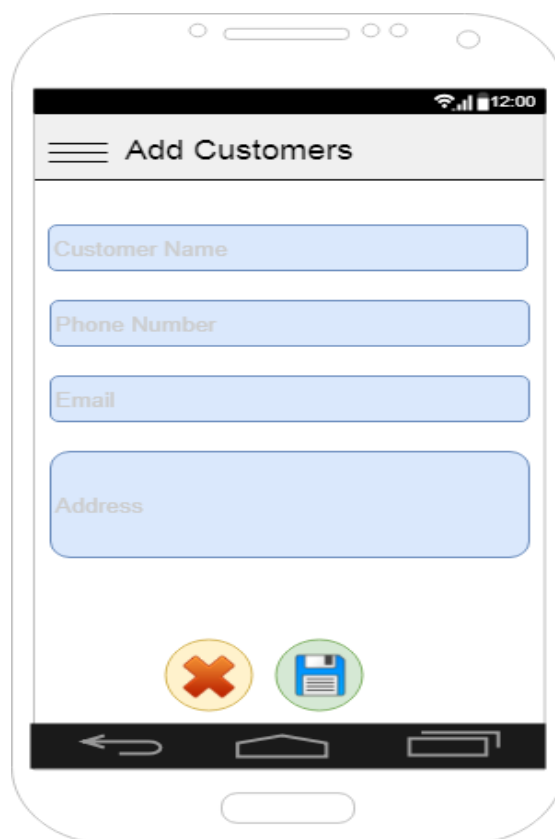
Appendix B-2.10 : Add Purchase Information page



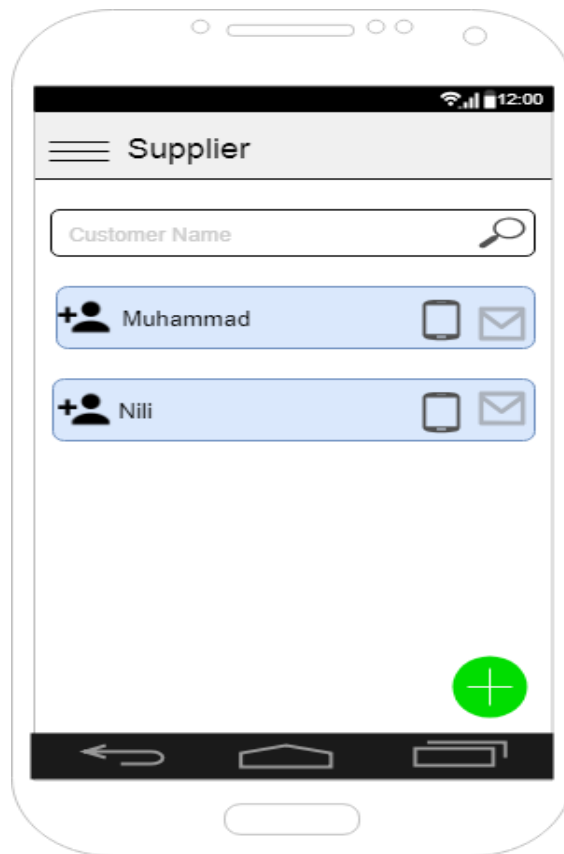
Appendix B-2.11 : Purchases History page



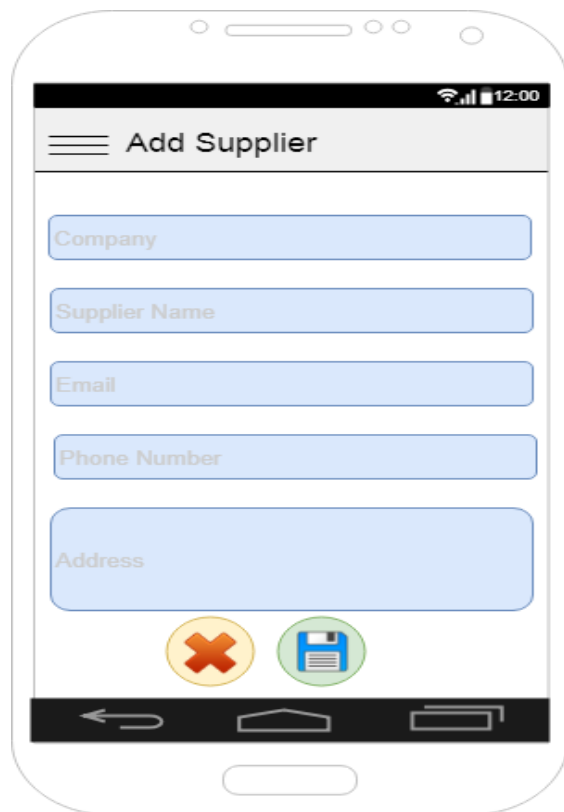
Appendix B-2.12 : Customer page



Appendix B-2.14 : Add Customer Information page



Appendix B-2.15 : Supplier page



Appendix B-2.17 : Add Supplier Information page



## **APPENDIX C**

### **SOFTWARE DESIGN DOCUMENT (SDD)**

**Version**

**1**

MOBILE BASED FOR INVENTORY SYSTEM (MBIS)

---

Faculty Computer Systems & Software Engineering (FSKKP)

# Software Design Document (SDD)

# Table of Contents

1.0	DATA DICTIONARY	3
2.0	PRELIMINARY DESIGN	7
	2.1 System Architecture	7
	2.1.1 Static Organization	7
	2.2.2 Dynamic Organization	11
3.0	DETAILED DESIGN	12
	3.1 Register Account	12
	3.2 Manage Inventory	14
	3.3 Manage Sales	18
	3.4 Manage Purchases	22
	3.5 Manage Customer	25
	3.6 Manage Supplier	29
4.0	SYSTEM REQUIREMENTS APPROVAL	33

## 1.0 DATA DICTIONARY

Data dictionary is the tables of database that contains the information in each table. The information in each table consists of field name, description, data type and constraint Table 1.1 to Table 1.6 shown the data dictionary in Mobile based for Inventory System.

### 1.1.1 Stockist Information

Table 1.1 Data Dictionary of Stockist Information

Field Name	Description	Data Type	Constraint
MBIS_StockistEmail	Stockist Email	String	PK
MBIS_StockistName	Stockist Name	String	
MBIS_StockistPhoneNum	Stockist Phone Number	String	
MBIS_StockistPassword	Stockist Password	String	
MBIS_StockistConfirmPassword	Stockist Password	String	

### 1.1.2 Product Information

Table 1.2 Data Dictionary of Inventory Management

Field Name	Description	Data Type	Constraint
MBIS_ProductBarcode	Product Barcode	Integer	PK
MBIS_ProductName	Product Name	String	

MBIS_ProductQuantity	Product Quantity	Integer	
MBIS_ProductPurchasePrice	Product Purchase Price	Double	
MBIS_ProductSalesPrice	Product Sales Price	Double	
MBIS_ProductInformaton	Product Description	String	
MBIA_ProductImage	Product Image	Image	

### 1.1.3 Sales Information

Table 1.3 Data Dictionary of Sales Information

Field Name	Description	Data Type	Constraint
MBIS_SaleID	Sales Id	Integer	PK
MBIS_SaleQuantity	Sales Quantity	Integer	
MBIS_SaleDiscount	Sales Discount	Integer	
MBIS_SaleTotal	Sales Total Price	Double	
MBIS_SaleType	Sales Type	String	
MBIS_SaleDescription	Sales Description	String	
MBIS_ProductBarcode	Product Barcode	Integer	FK
MBIS_CustEmail	Customer Email	String	FK

## 1.1.4 Purchases Information

Table 1.4 Data Dictionary of Purchases Information

Field Name	Description	Data Type	Constraint
MBIS_PurchaseID	Purchase Id	NUMBER	PK
MBIS_PurchaseQuantity	Purchase Quantity	NUMBER	
MBIS_PurchaseTotal	Purchase Total Price	DOUBLE	
MBIS_PurchaseInformation	Purchase Information	DOUBLE	
MBIS_ProductBarcode	Product Barcode	NUMBER	FK
MBIS_SuppEmail	Supplier Email	NUMBER	FK

## 1.1.5 Customer Information

Table 1.5 Data Dictionary of Customer Information

Field Name	Description	Data Type	Constraint
MBIS_CustEmail	Customer Email	String	
MBIS_CustName	Customer Name	String	
MBIS_CustPhoneNum	Customer Phone Number	String	
MBIS_CustAddress	Customer Address	String	

## 1.1.6 Supplier Information

Table 1.6 Data Dictionary of Supplier Information

Field Name	Description	Data Type	Constraint
MBIS_SuppEmail	Supplier Email	String	FK
MBIS_SuppCompany	Supplier Company Name	String	
MBIS_SuppName	Supplier Name	String	
MBIS_SuppPhoneNum	Supplier Phone Number	String	
MBIS_SuppAddress	Supplier Address	String	

## 2.0 PRELIMINARY DESIGN

### 2.1 System Architecture

In system organization consist of two organization that is static and dynamic which identifies the internal organizational structure of the system

#### 2.1.1 Static Organization

Figure 2.1 shown the static organization of Mobile Based for Inventory System. It consists of :

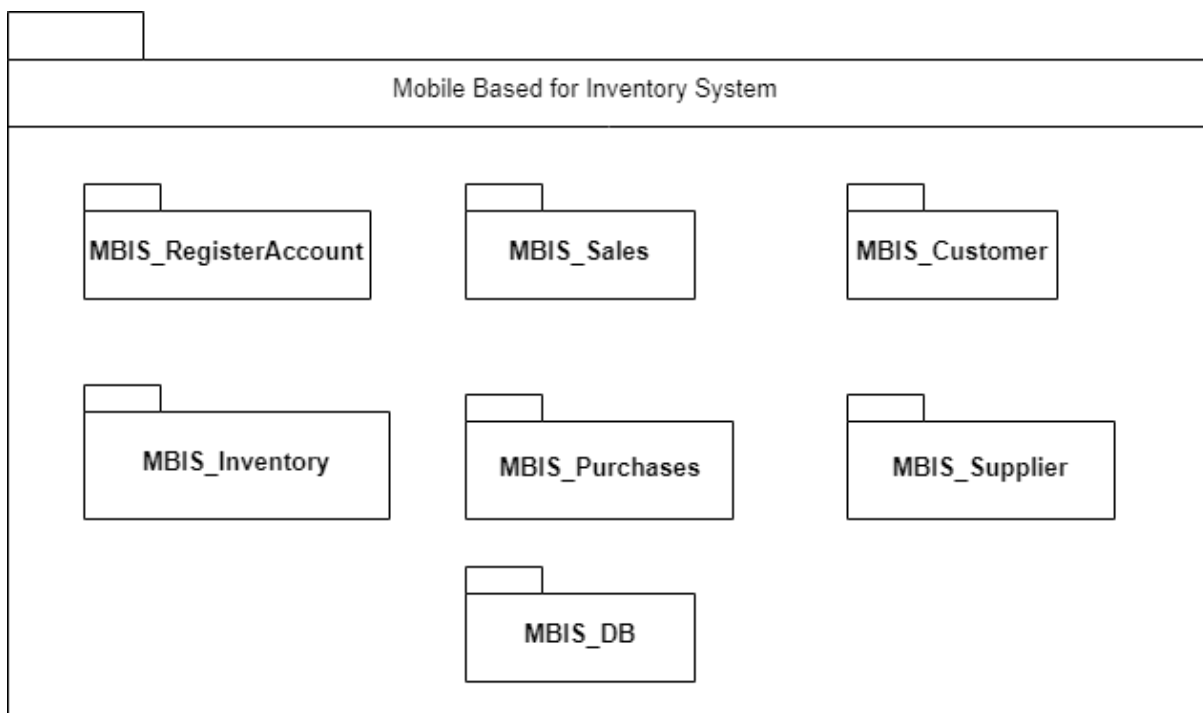


Figure 2.1 Static Organization of Mobile Based for Inventory System



This section describe the details for each subsystem/package.

#### 1. MBIS\_RegisterAccount

This package is responsible to control the information about stockist when interacting with of Mobile Based for Inventory System. This package consists of the following classes or unit :

- a) LOGIN Class
- b) activity\_login View Class
- c) REGISTRATION Class
- d) activity\_registration View Class
- e) Account Class
- f) AccountDB Class

#### 2. MBIS\_Inventory

This package is responsible to control the product of Mobile Based for Inventory System. This package consists of the following classes or unit :

- a) PRODUCT\_INFORMATION Class
- b) activity\_add\_product\_information View Class
- c) PRODUCT\_LIST Class
- d) activity\_product View Class
- e) PRODUCT\_UPDATE\_DELETE Class
- f) activity\_product\_update\_delete View Class
- g) ProductList\_Adapter Class
- h) Product\_Controller Class

### 3. MBIS\_Sales

This package is responsible to control sales of Mobile Based for Inventory System. This package consists of the following classes or unit :

- a) SALE Class
- b) activity\_sale View Class
- c) SALE\_HISTORY Class
- d) activity\_sale\_history View Class
- e) SALE\_INFORMATION Class
- f) activity\_sale\_information View Class

### 4. MBIS\_Purchases

This package is responsible to control purchases of Mobile Based for Inventory System. This package consists of the following classes or unit :

- a) PURCHASE Class
- b) activity\_purchase View Class
- c) PURCHASE\_HISTORY Class
- d) activity\_purchase\_history View Class
- e) PURCHASE\_INFORMATION Class
- f) activity\_purchase\_information View Class

## 5. MBIS\_Customer

This package is responsible to control the customer's information of Mobile Based for Inventory System. This package consists of the following classes or unit :

- a) CUSTOMER\_LIST Class
- b) activity\_customer\_list View Class
- c) CUSTOMER\_UPDATE\_DELETE Class
- d) activity\_customer\_update\_delete View Class
- e) Customer\_Controller Class
- f) customer\_information View Class

## 6. MBIS\_Supplier

This package is responsible to control the supplier's information of Mobile Based for Inventory System. This package consists of the following classes or unit :

- a) SUPPLIER\_LIST Class
- b) activity\_supplier\_list View Class
- c) SUPPLIER\_UPDATE\_DELETE Class
- d) activity\_supplier\_update\_delete View Class
- e) Supplier\_Controller Class
- f) Supplier\_information View Class

## 7. MBIS\_DB

This package does not consist any classes. This package just acts a reuse database that consist many table.

**2.1.2 Dynamic Organization**

Figure 2.2 shown the component and their relationships between each other in the system.

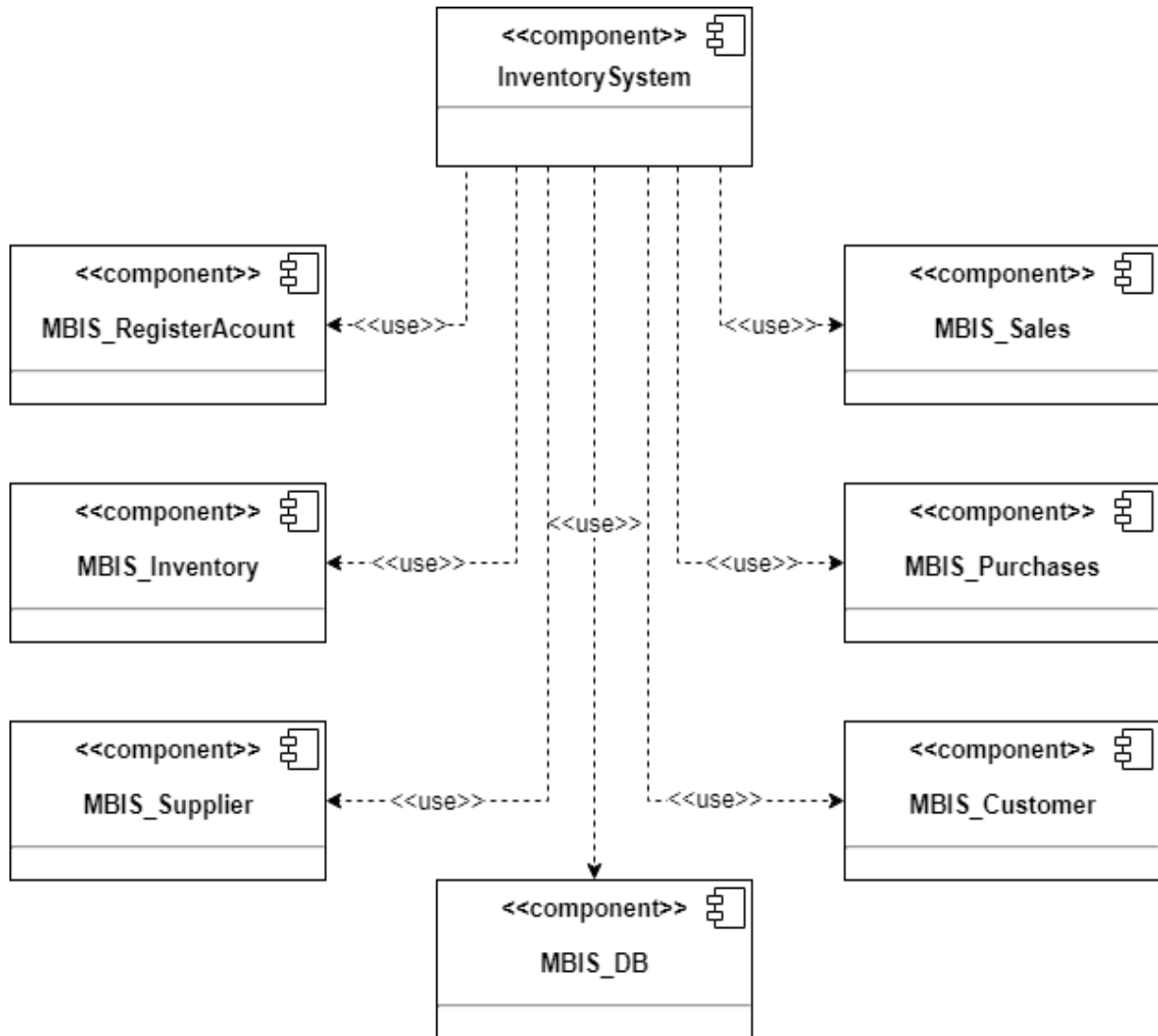


Figure 2.2 Component Diagram of Mobile Based for Inventory System

### 3.0 DETAILED DESIGN

#### 3.1 MBIS\_RegisterAccount

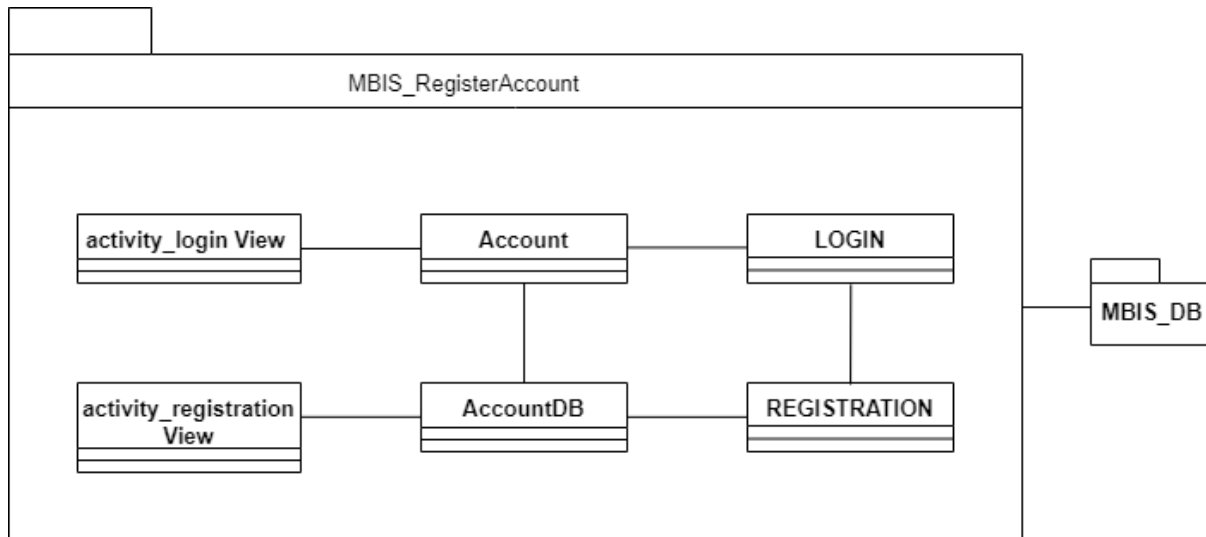


Figure 3.1 MBIS\_Register Account Package

##### 3.1.1 LOGIN Class

Table 3.1 Login Class

Class Type	Entity Class	
Responsibility	This class is responsible to login into the system	
Attribute	- MBIS_StockistEmail : String - MBIS_StockistPassword : String	
Method	- Login()	- To login into the system
Algorithm	BEGIN Stockist click button login The system display a list of product END	

## 3.1.2 activity\_login View Class

Table 3.2 Login View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view login form	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button login The system display a list of product END	

## 3.1.3 REGISTRATION Class

Table 3.3 Registration Class

Class Type	Entity Class	
Responsibility	This class is responsible to add new user information.	
Attribute	-MBIS_StockistEmail ; String -MBIS_StokistName : String -MBIS_StockistPhoneNum : String -MBIS_StockistPassword : String -MBIS_StockistConfirmPassword : String	
Method	- Register()	- To add product information
Algorithm	BEGIN Stockist click button register The system display login page and saved information END	

3.1.4 activity\_registration View Class

Table 3.4 Registration View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view registration details form	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button register The system display login page and saved information END	

3.2 MBIS\_Inventory

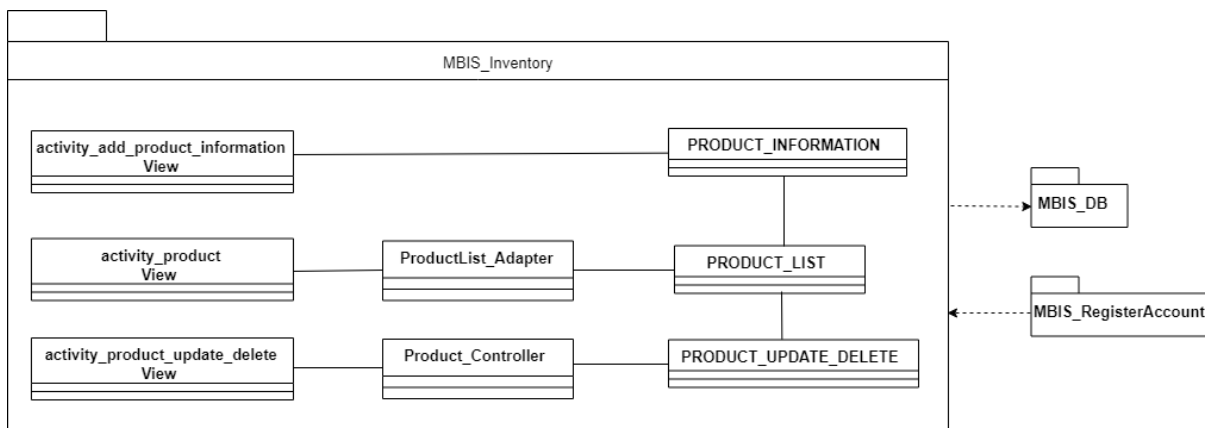


Figure 3.2 MBIS\_Inventory Package

3.2.1 PRODUCT\_INFORMATION Class

Table 3.5 Product Information Class

Class Type	Entity Class
Responsibility	This class is responsible to add product information.
Attribute	-MBIS_ProductBarcode : Integer

	-MBIS_ProductName : String -MBIS_ProductQuantity : Integer -MBIS_ProductPurchasePrice : Double -MBIS_ProductSalesPrice : Double -MBIS_ProductInformation : String - MBIS_ProductImage : Image	
Method	- addProduct()	- To add product information
Algorithm	BEGIN Stockist click button save product informations The system display a list of product with the button to manage product END	

### 3.2.2 activity\_add\_product\_information View Class

Table 3.6 Product Information View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view product details form.	
Attribute	-	
Method	-	
Algorithm	BEGIN Stockist click button add new product The system display product information form END	

### 3.2.3 PRODUCT\_LIST Class

Table 3.7 Product List Class

Class Type	Entity Class	
Responsibility	This class is responsible to display list of product	
Attribute	-MBIS_ProductBarcode : Integer	



	-MBIS_ProductName : String -MBIS_ProductQuantity : Integer -MBIS_ProductSalesPrice : Double -MBIS_ProductImage : Image	
Method	- getProducts() -parseItems()	- To get product - To get product
Algorithm	BEGIN Stockist click button inventoy The system display a list of product and add product button END	

### 3.2.4 activity\_product\_list View Class

Table 3.8 Product List View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view product list	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button inventoy The system display a list of product and add product button END	

### 3.2.5 PRODUCT\_UPDATE\_DELETE Class

Table 3.9 Product Update Delete Class

Class Type	Entity Class	
Responsibility	This class is responsible to update and delete product.	
Attribute	-MBIS_ProductBarcode : Integer -MBIS_ProductName : String	

	-MBIS_ProductQuantity : Integer -MBIS_ProductPurchasePrice : Double -MBIS_ProductSalesPrice : Double -MBIS_ProductInformation : String - MBIS_ProductImage : Image	
Method	- ConfirmationProductDelete() -DeleteProduct() - ConfirmationProductUpdate() -UpdateProduct()	-Confirmation on Deleting product information - Delete product information -Confirmation on updating product information - Update product information
Algorithm	BEGIN Stockist click button update and delete on product information The system update the product list and the product information END	

### 3.2.6 activity\_product\_update\_delete View Class

Table 3.10 Product Update Delete View Class

Class Type	View	
Responsibility	This class is allow stockist to view product informations.	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click on specific product The system display product informations END	

3.2.7 Product\_Controller Class

Table 3.11 Product ControllerClass

Class Type	Controller Class	
Responsibility	This class is responsible to manage the product.	
Attribute	-	-
Method	- deleteData() - updateData()	- To delete product information -To update product information
Algorithm	BEGIN Stockist click to view product information Stockist click to add product Stockist click to update product Stockist click to delete product Stockist click to generate the product report END	

3.3 MBIS\_Sales

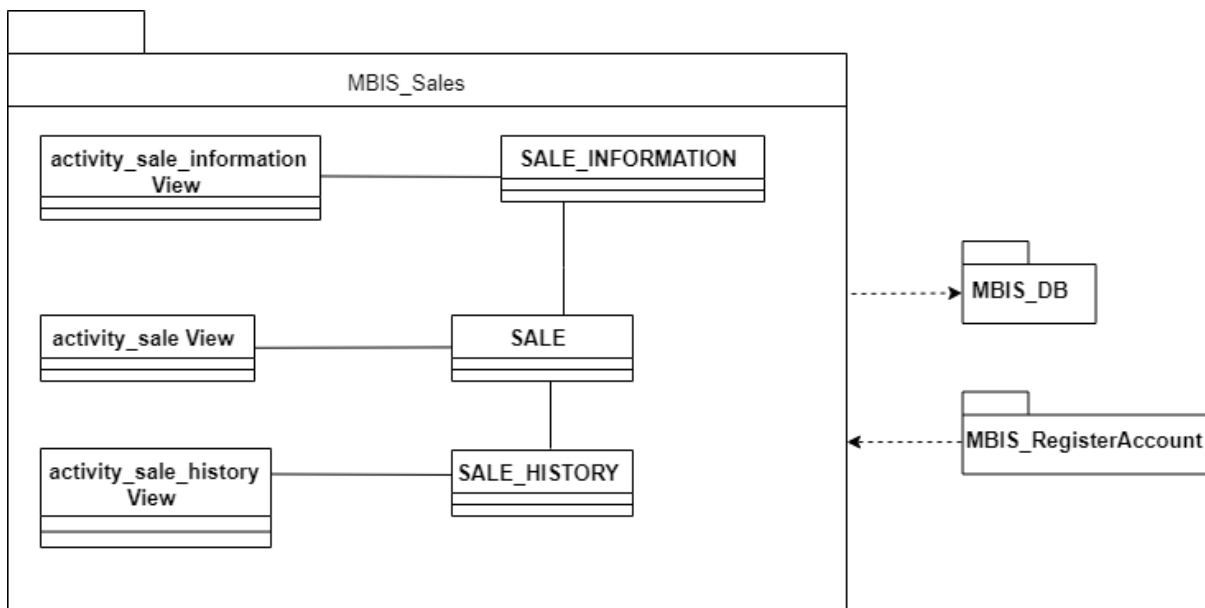


Figure 3.3 MBIS\_Sales Package

## 3.3.1 SALE Class

Table 3.12 Sale Class

Class Type	Entity Class	
Responsibility	This class is responsible to display list of product	
Attribute	-MBIS_ProductBarcode : Integer -MBIS_ProductName : String -MBIS_ProductQuantity : Integer -MBIS_ProductSalesPrice : Double - MBIS_ProductImage : Image	
Method	- getProducts() -parseItems()	- To get product - To get product
Algorithm	BEGIN Stockist click button sale The system display a list of product, sale history button and search button END	

## 3.3.2 activity\_sale View Class

Table 3.13 Sale View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view product list	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button sale The system display a list of product, sale history button and search button END	

## 3.3.3 SALE\_HISTORY Class

Table 3.14 Sale History Class

Class Type	Entity Class	
Responsibility	This class is responsible to display sale history	
Attribute	-MBIS_SaleQuantity : Integer - MBIS_SaleTotal : Double - MBIS_SaleStatus : String - MBIS_SaleDescription : String - MBIS_ProductName : String - MBIS_CustName : String	
Method	- getSaleHistory() -parseItems()	- To get sale history - To get sale history
Algorithm	BEGIN Stockist click button sale history The system display a list of sale history END	

## 3.3.4 activity\_sale\_history View Class

Table 3.15 Sale History View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view sale history	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button sale history The system display a list of sale history END	

## 3.3.5 SALE\_INFORMATION Class

Table 3.16 Sale Information Class

Class Type	Entity Class	
Responsibility	This class is responsible to display sale information form	
Attribute	-MBIS_SaleQuantity : Integer -MBIS_SaleDiscount : Integer - MBIS_SaleTotal : Double - MBIS_SaleStatus : String - MBIS_SaleDescription : String - MBIS_CustEmail : String	
Method	- checkQty() -addSale() -getCustomers() -parseItems()	-To check quantity of product -To add new sale - To get Customer name - To get Customer name
Algorithm	BEGIN Stockist click button save sale informations The system display a list of product with updated quantity END	

## 3.3.6 activity\_sale\_information View Class

Table 3.17 Sale Information View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view sale information form	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button save sale informations The system display a list of product with updated quantity END	

### 3.4 MBIS\_Purchases

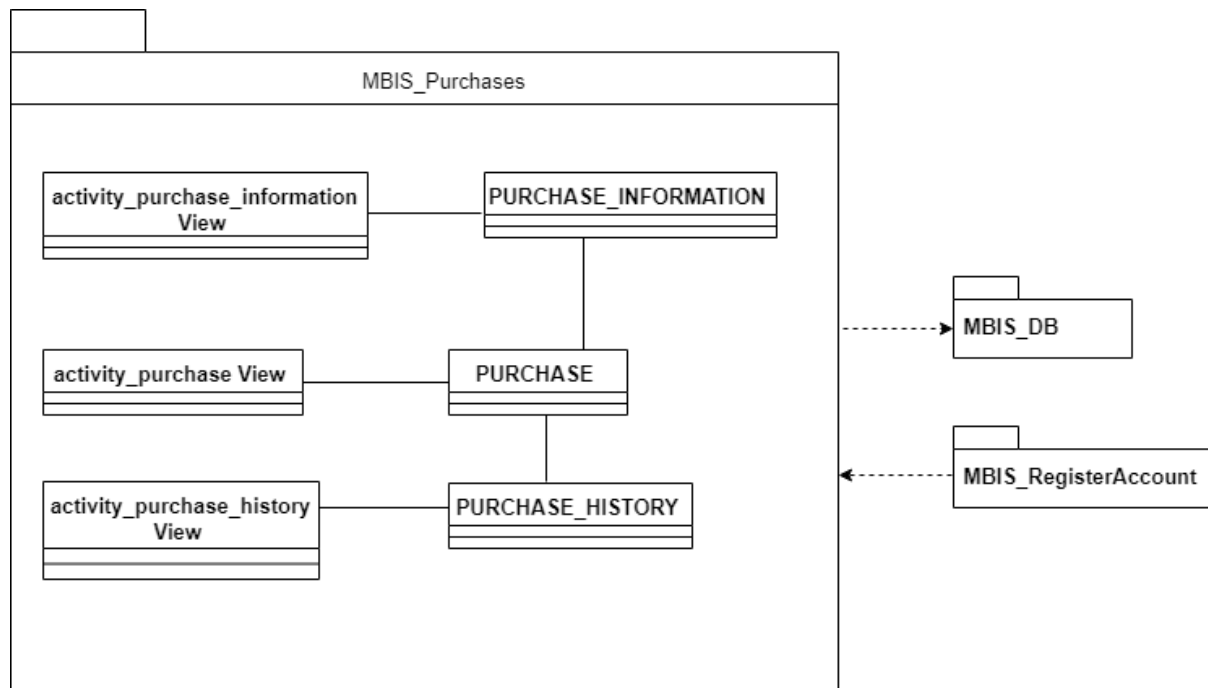


Figure 3.4 MBIS\_Purchases Package

#### 3.4.1 PURCHASE Class

Table 3.18 Purchase Class

Class Type	Entity Class	
Responsibility	This class is responsible to display list of product	
Attribute	-MBIS_ProductBarcode : Integer -MBIS_ProductName : String -MBIS_ProductQuantity : Integer -MBIS_ProductSalesPrice : Double -MBIS_ProductImage : Image	
Method	- getProducts() -parseItems()	- To get product - To get product
Algorithm	BEGIN Stockist click button purchase	

	<p>The system display a list of product, purchase history button and search button</p> <p>END</p>
--	---

### 3.4.2 activity\_purchase View Class

Table 3.19 Purchase View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view product list	
Attribute	-	
Method	-	-
Algorithm	<p>BEGIN</p> <p>Stockist click button purchase</p> <p>The system display a list of product, purchases history button and search button</p> <p>END</p>	

### 3.4.3 PURCHASE\_HISTORY Class

Table 3.20 Purchase History Class

Class Type	Entity Class	
Responsibility	This class is responsible to display purchase history	
Attribute	<p>- MBIS_PurchaseQuantity : String</p> <p>- MBIS_PurchaseTotal : Double</p> <p>- MBIS_PurchaseInformation : String</p> <p>-MBIS_SuppName : String</p> <p>-MBIS_ProductName : String</p>	
Method	<p>- getPurchaseHistory()</p> <p>-parseItems()</p>	<p>- To get purchase history</p> <p>- To get purchase history</p>
Algorithm	<p>BEGIN</p> <p>Stockist click button purchase history</p>	



	The system display a list of purchase history END
--	--

#### 3.4.4 activity\_purchase\_history View Class

Table 3.21 Purchase History View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view purchase history	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button purchase history The system display a list of purchase history END	

#### 3.4.5 PURCHASE\_INFORMATION Class

Table 3.22 Purchase Information Class

Class Type	Entity Class	
Responsibility	This class is responsible to display purchase information form	
Attribute	-MBIS_PurchaseQuantity : String - MBIS_PurchaseTotal : Double - MBIS_PurchaseInformation : String - MBIS_SuppName : String	
Method	-addPurchase() -getSupplier() -parseSupplier()	-To add new purchase - To get supplier name - To get supplier name
Algorithm	BEGIN Stockist click button save purchase informations The system display a list of product with updated quantity END	

3.4.6 activity\_purchase\_information View Class

Table 3.23 Purchase Information View Class

Class Type	Boundary Class	
Responsibility	This class is responsible to view purchase information form	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button save purchase informations The system display a list of product with updated quantity END	

3.5 MBIS\_Customer

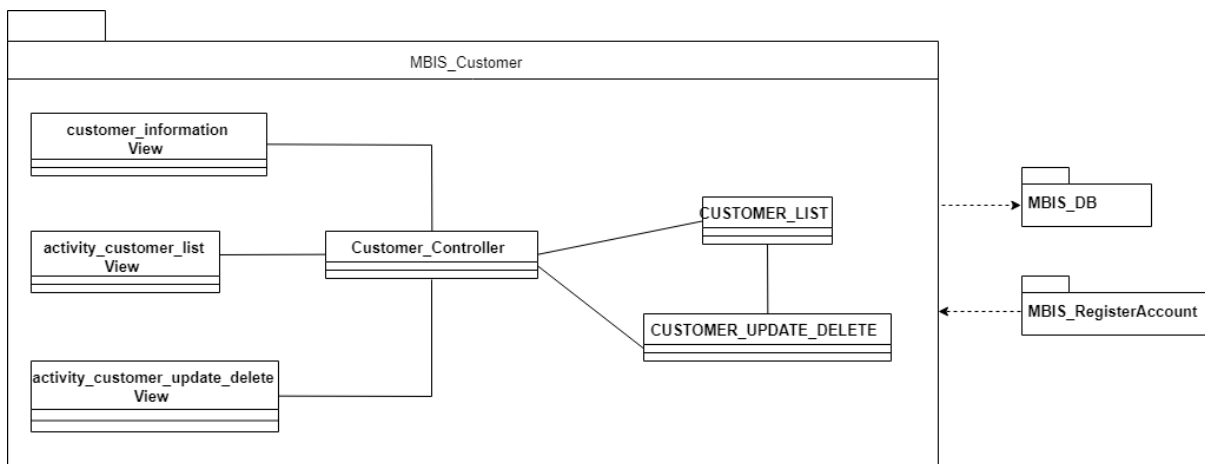


Figure 3.5 MBIS\_Customer Package

## 3.5.1 CUSTOMER\_LIST Class

Table 3.24 Customer List Class

Class Type	Entity Class	
Responsibility	This class is responsible to display customer and add new customer	
Attribute	-MBIS_CustName : String -MBIS_CustPhoneNum : String -MBIS_CustEmail : String -MBIS_CustAddress : String	
Method	-getCustomers() -parseItems() -addCustomer()	-To get the product information - To get the product information -To add customer information
Algorithm	BEGIN Stockist click button save on add new supplier The system get the product Information END	

## 3.5.2 activity\_customer\_list View Class

Table 3.25 Customer List View Class

Class Type	View Class	
Responsibility	This class allow stockist to view list of customer	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button customer The system display a list of customer END	

## 3.5.3 CUSTOMER\_UPDATE\_DELETE Class

Table 3.26 Customer Update Delete Class

Class Type	Entity Class	
Responsibility	This class is responsible to update and delete customer.	
Attribute	-MBIS_CustName : String - MBIS_CustPhoneNum : String -MBIS_CustEmail : String -MBIS_CustAddress : String	
Method	-makePhoneCall() -SendEmail() - ConfirmationCustomerDelete() -DeleteCustomer() - ConfirmationCustomerUpdate() -UpdateCustomer()	-To make a phone call -To send email -Confirmation on Deleting Customer Information - Delete Customer Information -Confirmation on Updating Customer Information - Update Customer Information
Algorithm	BEGIN Stockist click button update and delete on customer information The system update the customer list and the customer information END	

## 3.5.4 activity\_customer\_update\_delete Class

Table 3.27 Customer Update Delete View Class

Class Type	View Class	
Responsibility	This class is allow stockist to view customer informations.	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click on specific customer The system display customer informations	

	END
--	-----

### 3.5.5 Customer\_Controller Class

Table 3.28 Customer Controller Class

Class Type	Controller	
Responsibility	This class is responsible to manage the process occur under customer	
Attribute	-	-
Method	- insertData() -deleteData() -updateData()	-To insert customer information -To delete customer information -To update customer information
Algorithm	BEGIN Stockist click button add customer Stockist click button delete customer Stockist click button add new customer END	

### 3.5.6 customer\_information View Class

Table 3.29 Customer Informatio View Class

Class Type	View Class	
Responsibility	This class is allow stockist to view customer information form	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click on button add customer The system display add customer information form END	

### 3.6 MBIS\_Supplier

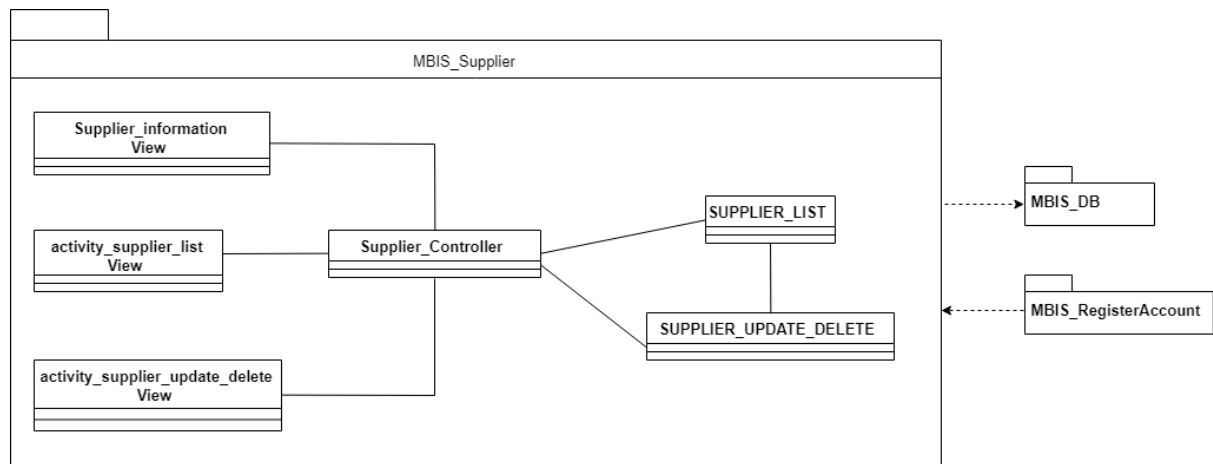


Figure 3.6 MBIS\_Supplier Package

#### 3.6.1 SUPPLIER\_LIST Class

Table 3.30 Supplier List Class

Class Type	Entity Class	
Responsibility	This class is responsible to display supplier and add new supplier	
Attribute	-MBIS_SuppCompany : String -MBIS_SuppName : String -MBIS_SuppPhone : String -MBIS_SuppEmail : String -MBIS_SuppAddress : String	
Method	-getSuppliers() -parseItems() -addSupplier()	-To get the supplier information -To get supplier information -To add supplier information
Algorithm	BEGIN Stockist click button save on add new supplier The system update supplier list END	

## 3.6.2 activity\_supplier\_list View Class

Table 3.31 Supplier List View Class

Class Type	View Class	
Responsibility	This class allow stockist to view list of supplier	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click button supplier The system display a list of supplier END	

## 3.6.3 SUPPLIER\_UPDATE\_DELETE Class

Table 3.32 Supplier Update Delete Class

Class Type	Entity Class	
Responsibility	This class is responsible to update and delete supplier.	
Attribute	-MBIS_SuppName : String -MBIS_SuppCompany : String -MBIS_SuppEmail : String -MBIS_SuppPhoneNum : String -MBIS_SuppAddress : String	
Method	-makePhoneCall() -SendEmail() - ConfirmationSupplierDelete() -DeleteSupplier() - ConfirmationSupplierUpdate() -UpdateSupplier()	-To make a phone call -To send email -Confirmation on Deleting Supplier Information - Delete Supplier Information -Confirmation on Updating Supplier Information - Update Supplier Information
Algorithm	BEGIN Stockist click button update and delete on supplier information	

	The system update the supplier list and the supplier information END
--	---

### 3.6.4 activity\_supplier\_update\_delete View Class

Table 3.33 Supplier Update Delete View Class

Class Type	View Class	
Responsibility	This class is allow stockist to view supplier informations.	
Attribute	-	
Method	-	-
Algorithm	BEGIN Stockist click on specific supplier The system display supplier informations END	

### 3.6.5 Supplier\_Controller Class

Table 3.34 Supplier Controller Class

Class Type	Controller Class	
Responsibility	This class is responsible to manage the process occur under supplier	
Attribute	-	-
Method	-insertData() -deleteData() -updateData()	-To insert supplier information -To delete supplier information -To update supplier information
Algorithm	BEGIN Stockist click button add supplier Stockist click button delete supplier Stockist click button add new supplier END	



## 3.6.6 Supplier\_information View Class

Table 3.35 Supplier Information View Class

Class Type	View Class	
Responsibility	This class is allow stockist to view supplier information form	
Attribute	-	-
Method	-	-
Algorithm	BEGIN Stockist click on button add supplier The system display add supplier information form END	


**4.0 SYSTEM DESIGN APPROVAL**

	<b>Name</b>	<b>Date</b>
<b>Verified by:</b>  _____  Developer	CHE NORMADIAA BINTI IBRAHIM	
<b>Approved by:</b>  _____  Client		

**APPENDIX D**

USER MANUAL

**MOBILE BASED FOR INVENTORY...**



**Mobile Based for Inventory System**

Email

Password

**LOG IN**

Forgot Password ?

**Are you new ?**

**SIGN UP**

### Appendix D-1.1 Login Page

1. User insert email and password to enter into the system.
2. Then click “LOG IN” button.
3. If user do not have account, user need to click “SIGN UP” button to create new account.

Register New Account

*Mobile Based For*  
**Inventory System**

Name

Email

Phone Number








Password

Confirm Password

REGISTER

#### Appendix D-1.2 Register Page

1. User insert all information needed : name, email, phone number, password and confirm password.
2. The information cannot be empty.
3. Then click “REGISTER” button in order to get registered email and password and to get new account.

Inventory	
	<p><b>10060294</b>            CITYSCAPE Cologne Spray            Quantity : 48      Sales (RM) : 139</p>
	<p><b>10101620</b>            Body Spray test            Quantity : 15      Sales (RM) : 75</p>
	<p><b>10078697</b>            CITYSCAPE Cologne Spray            Quantity : 93      Sales (RM) : 139</p>
	<p><b>10075680</b>            MK High Intensity Sport Cologne Spray            Quantity : 14      Sales (RM) : 159</p>
	<p><b>10060202</b>            Advanced Eye Cream            Quantity : 20      Sales (RM) : 85</p>
	<p><b>10091570</b>            MK High Intensity Ocean Cologne Spray            Quantity : 2      Sales (RM) : 139</p>
	<p><b>10105438</b>            Clarifying Cleanser for Acne-Prone            Quantity : 3      Sales (RM) : 139</p>

### Appendix D-1.3 Product Page

1. System automatically displayed the list of the saved product.
2. User can add new product by clicking add floating button at the bottom.
3. User can see the information, update and delete of each product by clicking particular product.
4. User can choose the other function (sale, purchase, customer and supplier) by clicking drawer on the left top.

**Add Product Details**

Product Code :

Product Name :

Quantity :	Purchases (RM) :	Sales (RM) :
<input type="text"/>	<input type="text"/>	<input type="text"/>

Additional Product Information :

#### Appendix D-1.4 Add Product Information Page

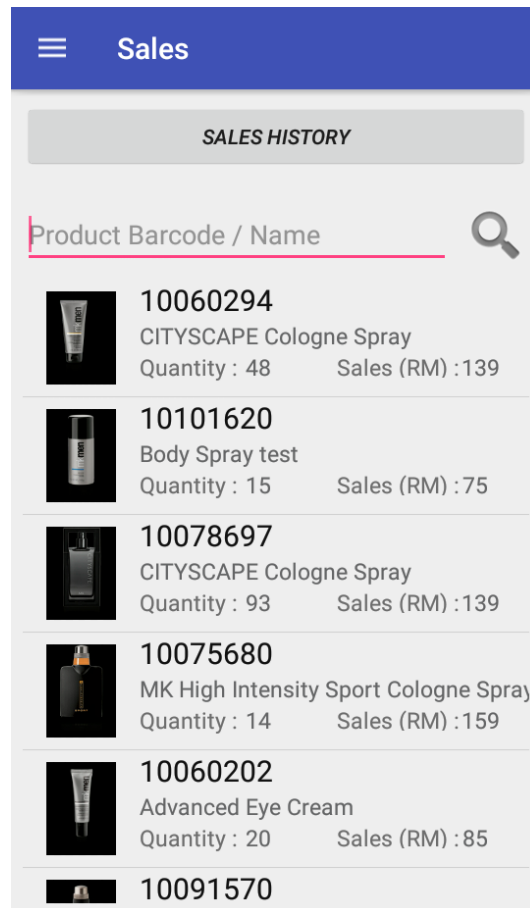
1. User insert all the product information : image, product code, product name, quantity, purchase price, sale price, and additional product information
2. Purchase price must be lower than sale price.
3. User click “SAVE” button to keep the new product information in the Google Excel.
4. User can click “CANCEL” button if want to cancel adding new product.



#### Appendix D-1.5 Update and Delete Product Page

1. This page appear once user clicking on the particular product in the Product page.
2. User can delete the existing product by clicking “DELETE” button on the buttom.
3. User can update the information of the existing product and then click “SAVE” button at the buttom to save new product information.





#### Appendix D-1.6 Sale Page

1. The list of product automatically displayed on the sale page
2. User can add new sale by clicking on the particular product.
3. User can use search to find the specific product by inserting product name or code.
4. User click "SALES HISTORY" button to see the history of the sales product.

#### Appendix D-1.7 Add Sale Information Page

1. This page appear once user clicking on the particular product in the Sale page.
2. In order to make new sale, user need to click “SAVE” button after inserted the quantity of sale product.
3. The discount, sale type client name and additional sale information is an optional either user want to enter or not.
4. User can click “CANCEL” button if want to cancel adding new sale.

## Sales History

Product Name : **Freshen**

Quantity : 245                      Total (RM) : 86  
Client Name :                      Receivables  
Information : a  
*April 5, 2019 1:28:55 AM HKT*

---

Product Name : **Mary Kay Liquid Foundation Brush**

Quantity : 45                      Total (RM) : 58  
Client Name :                      Receivables  
Information : b  
*April 5, 2019 2:32:48 AM HKT*

---

Product Name : **Advanced Facial Hydrator Sunscreen SF**

Quantity : 25                      Total (RM) : 44  
Client Name :                      Paid  
Information : c  
*April 5, 2019 2:33:50 AM HKT*

---

Product Name : **Body Spray**

Quantity : 55                      Total (RM) : 2  
Client Name :                      Paid  
Information : d  
*April 5, 2019 2:34:28 AM HKT*

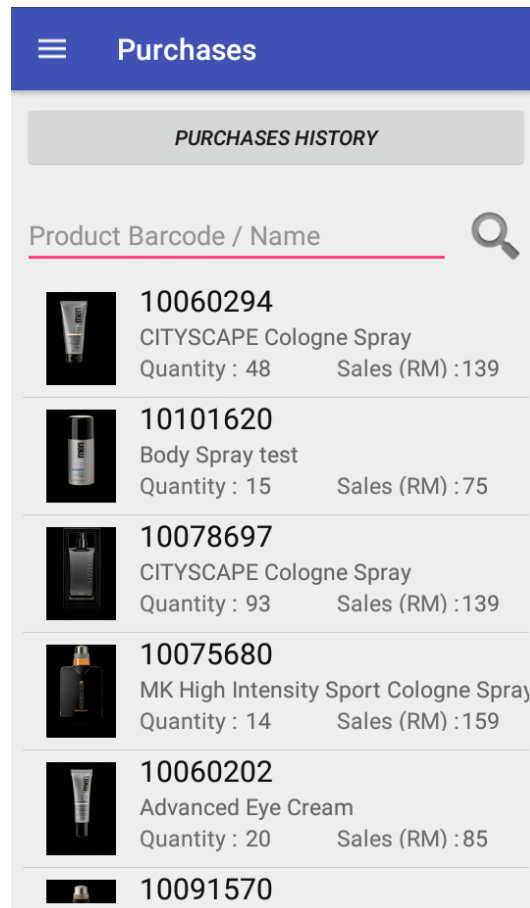
---

Product Name : **CITYSCAPE Cologne Spray**

Quantity : 619                      Total (RM) : 848  
Client Name :                      Receivables  
Information : e

### Appendix D-1.8 Sale History Page

1. The list of the sale history automatically displayed once user clicking button “SALE HISTORY” on the Sale page.



#### Appendix D-1.9 Purchase Page

1. The list of product automatically displayed on the purchase page.
2. User can add new purchase product by clicking on the particular product.
3. User can use search to find the specific product by inserting product name or code.
4. User click "PURCHASES HISTORY" button to see the history of the purchase product.

**Purchases Information**

**10060294**  
**CITYSCAPE Cologne Spray**

Current Quantity :	Add :	Total Quantity :
48	Quantity	48
Purchase (RM) :	Sale (RM) :	Total (RM) :
100	139	

+ 👤 Supplier name (Optional)

✎ Additional Purchase Information

✕
📄

#### Appendix D-1.10 Add Purchase Information Page

1. This page appear once user clicking on the particular product in the Purchase page.
2. In order to make new purchase product, user need to click “SAVE” button after inserted the quantity of purchased product.
3. The supplier name and additional purchase information is an optional either user want to enter or not.
4. User can click “CANCEL” button if want to cancel adding new purchase.

## Purchases History

**Product Name : Mary Kay Liquid Foundation Brush**

Quantity : 2 **Total (RM) : 130**

Supplier Name :

Information : a

*April 7, 2019 2:17:22 AM HKT*

**Product Name : Freshen**

Quantity : 2 **Total (RM) : 200**

Supplier Name :

Information : b

*April 7, 2019 3:36:35 AM HKT*

**Product Name : Body Spray**

Quantity : 5 **Total (RM) : 450**

Supplier Name :

Information : c

*April 14, 2019 12:10:40 PM HKT*

**Product Name : CITYSCAPE Cologne Spray**

Quantity : 3 **Total (RM) : 450**

Supplier Name :

Information : d

*April 22, 2019 8:22:55 PM HKT*

**Product Name : CITYSCAPE Cologne Spray**

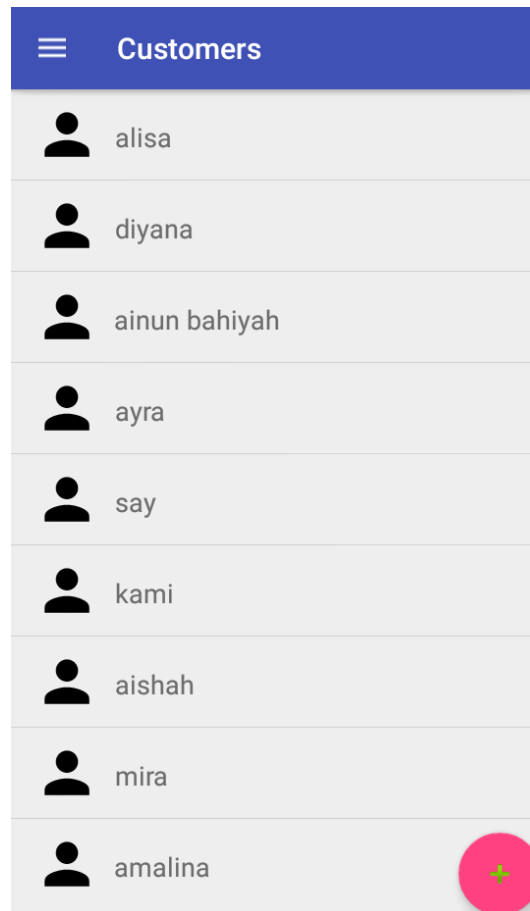
Quantity : 3 **Total (RM) : 450**

Supplier Name :

Information : e

### Appendix D-1.11 Purchase History Page



1. The list of the purchase history automatically displayed once user clicking button “PURCHASE HISTORY” on the Purchase page.



Appendix D-1.12 Customer Page

1. System automatically displayed the list of the saved customer.
2. User can add new customer by clicking add floating button at the bottom.
3. User can see the information, update and delete of each customer by clicking particular customer name.

### Add Customer Details


#### Appendix D-1.13 Add Customer Information Page


1. User insert all the customer information : name, phone number, email, and address.
2. User click “SAVE” button to keep the new customer information in the Google Excel.
3. User can click “CANCEL” button if want to cancel adding new customer.





### Update and Delete Customer

Customer's Name :  
alisa

Phone Number :  
016-6592560 

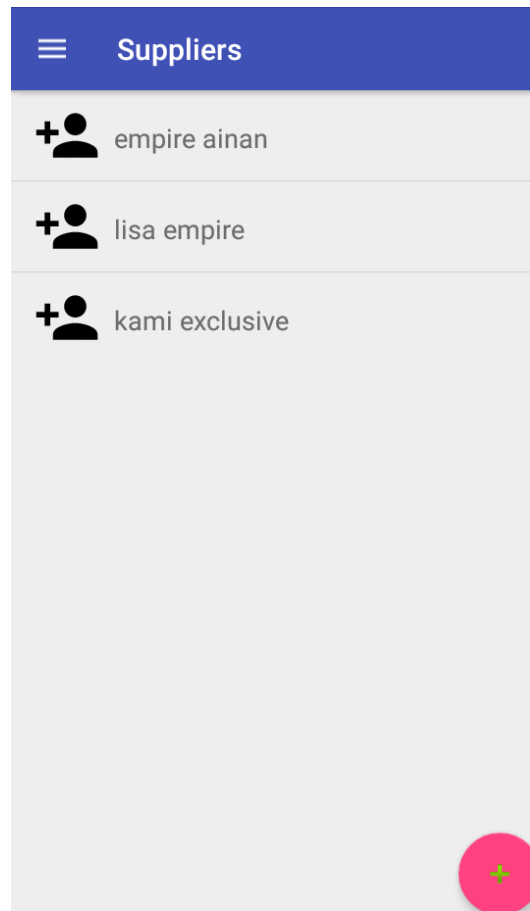
Email :  
alisa@gmail.com 

Address :  
johor

#### Appendix D-1.14 Update and Delete Customer Page

1. This page appear once user clicking on the particular customer in the Customer page.
2. User can delete the existing customer by clicking “DELETE” button at the buttom.
3. User can update the information of the existing customer and then click “SAVE” button at the buttom to save new customer information.
4. User can directly call or email customer from this page by clicking “PHONE” and “EMAIL” button.



Appendix D-1.15 Supplier Page

1. System automatically displayed the list of the saved supplier.
2. User can add new supplier by clicking add floating button at the bottom.
3. User can see the information, update and delete of each supplier by clicking particular supplier name.

**Add Supplier Details**

Company

Supplier's Name

Phone Number

Email

Address

X [Document Icon]


#### Appendix D-1.16 Add Supplier Information Page


1. User insert all the supplier information : company name, supplier name, phone number, email, and address.
2. User click “SAVE” button to keep the new supplier information in the Google Excel.
3. User can click “CANCEL” button if want to cancel adding new supplier.

### Update and Delete Supplier



Company Name :  
empire ainan

Name :  
ainan tasnim

Phone Number :  
016-6592560 

Email :  
ain@gmail.com 

Address :  
kuantan pahang

Appendix D-1.17 Update and Delete Supplier Page

1. This page appear once user clicking on the particular supplier in the Supplier page.
2. User can delete the existing supplier by clicking “DELETE” button at the buttom.
3. User can update the information of the existing supplier and then click “SAVE” button at the buttom to save new supplier information.
4. User can directly call or email supplier from this page by clicking “PHONE” and “EMAIL” button.

## **APPENDIX E**

### **USER ACCEPTANCE TEST (UAT)**

No.	Module	Activities	Status		Comments
			Yes	No	
1	Register Account	Register new account	√		
2		User login	√		
3	Manage Inventory	Insert new product	√		
4		Display inserted product	√		
5		Update product information	√		
6		Delete product	√		
7	Manage Sales	Insert new sale	√		
8		View sale history	√		
19		Search product	√		
10	Manage Purchases	Insert new purchase	√		
11		View purchase history	√		
12		Search product	√		
13	Manage Customer	Insert new customer	√		
14		Display inserted customer	√		
15		Update customer information	√		
16		Delete customer	√		
17		Call customer	√		
18		Email customer	√		
19	Manage Customer	Insert new supplier	√		
20		Display inserted supplier	√		
21		Update supplier information	√		
22		Delete supplier	√		
23		Call supplier	√		
24		Email supplier	√		

This test has been performed by:

Name: Nur Amalina Abdul Halim

Date: 03/05/2019

No.	Module	Activities	Status		Comments
			Yes	No	
1	Register Account	Register new account	√		
2		User login	√		
3	Manage Inventory	Insert new product	√		
4		Display inserted product	√		
5		Update product information	√		
6		Delete product	√		
7	Manage Sales	Insert new sale	√		
8		View sale history	√		
19		Search product	√		
10	Manage Purchases	Insert new purchase	√		
11		View purchase history	√		
12		Search product	√		
13	Manage Customer	Insert new customer	√		
14		Display inserted customer	√		
15		Update customer information	√		
16		Delete customer	√		
17		Call customer	√		
18		Email customer	√		
19	Manage Customer	Insert new supplier	√		
20		Display inserted supplier	√		
21		Update supplier information	√		
22		Delete supplier	√		
23		Call supplier	√		
24		Email supplier	√		

This test has been performed by:

Name: Nur Aqilah Osman

Date: 03/05/2019