[INVENTORY MAP FOR HYPERMARKET]

[NUR TAJALLI KAMALPUTRI BINTI KAMARULZAMAN]

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[INVENTORY MAP FOR HYPERMARKET]

[NUR TAJALLI KAMALPUTRI BINTI KAMARULZAMAN]

This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Database Management)] with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[2021]

DECLARATION

I hereby declare that this project report entitled

[INVENTORY MAP FOR HYPERMARKET]

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : NUR TAJALLI KAMALPUTRI BINTI KAMARULZAMAN Date : 25 MARCH 2021

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of [Computer Science (Database Management)] with Honours.

SUPERVISOR : TS. DR. YAHAYA BIN ABD RAHIM Date : 28 MARCH 2021

DEDICATION

I would like to express my special dedication to my utmost understanding and beloved parents who inspire me to always be outstanding and never give up throughout my education journey. They always support me wherever I feel down, they give me courage and advice. They are my backbone. Thank you, my beloved parents.

Last but not least, I want to thank me, I want to thank me for believing in me, I want to thank me for doing all this hard work, I want to thank me for having no days off, I want to thank me for never quitting, for just being me all times.

ACKNOWLEDGEMENTS

I would like to thank the University Technical Malaysia Melaka (UTeM) for giving us students of Faculty of Information and Communication Technology (FICTS) the chance to complete our final year project within the whole semester. I would also like to thank the committee for the briefing that was given what needs to do in the final year project.

Next, I would like to express my gratitude to my supervisor Ts. Dr Yahaya binti Abd Rahim for helping and advising me throughout my final year project journey. Without his help, I cannot complete my final year project.

Finally, I would like to dedicate my appreciation to my parents. Their inspiration word that makes me stronger during my final year project. Without them, I would not be able to be where I am standing now.

ABSTRACT

Inventory Map for Hypermarket is a web-database system proposed to keep the profile and related details to the database and the permissions to add, view, edit or delete is given only to staff and admin of the system. As we know that some hypermarket still uses paper to keep all the inventory map of products manually. Unfortunately, the data of the inventory map product could easily lose and can have redundant data. Besides, the staff cannot keep the inventory map product information. There is also no online system for shopper to search their desire product. Hence, in the proposed system, staff can keep all the detail of the product navigation map in the system. The staff can also edit, remove, and search any product using product's name and the brand. Lastly, admin can add new staff and delete the staff that no longer works with them.

ABSTRAK

Sistem Peta Inventori Produk adalah sistem pangkalan data web yang dicadangkan untuk menyimpan profil dan perincian yang berkaitan dengan pangkalan data dan kebenaran untuk menambah, melihat, mengedit atau menghapus hanya diberikan kepada kakitangan dan pentadbir sistem. Seperti yang kita ketahui bahawa beberapa pasar raya besar masih menggunakan kertas untuk menyimpan semua data peta inventori produk secara manual. Sayangnya, data tersebut boleh hilang dengan mudah dan mempunyai data berlebihan. Selain itu, kakitangan tidak dapat menyimpan maklumat peta inventori produk. Oleh itu, dalam sistem yang dicadangkan, kakitangan dapat menyimpan semua maklumat pesakit dalam sistem tersebut. Kakitangan juga boleh mengubahsuai, membuang atau mencari mana-mana produk menggunakan nama produk atau jenama produk tersebut. Akhir sekali, admin boleh menambah kakitangan baru atau membuang kakitangan yang tidak lagi bekerja di pasar raya tersebut.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	-	**Final Year Project**
IMH	-	**Inventory Map for Hypermarket**
PHP	-	**Hypertext Preprocessor**
HTML	-	**Cascading Style Sheets**
SQL	-	**Hypertext Markup Language**
CSS	-	**Structured Query Language**

LIST OF ATTACHMENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction

Over time, a lot of people loves to shop in hypermarket as it has almost everything needed for home. Since it has a lot of things sold, there are also a lot of department and rack in one hypermarket that need to be studied by the buyer or shopper every time they want to search for their needs. It would not be a problem if there are no changes in their inventory for the entire year. However, almost every month, the management of hypermarket will be deciding to change some of their inventory rack to make their design and structure more interesting and attractive. Buyer faced a lot of problems every time the hypermarket decided to change the arrangement of department and it can be annoyed for the buyer who wants to make a fast shopping due to limited time.

Each hypermarket can reach up to 20 departments with hundreds of racks and multiple floors. There is also new product added monthly and it can confuse the buyer to search for the new product. There is no any specific system that can help this issues and making the buyer life easier.

This project proposes to ensure high monitoring of the inventory map in hypermarket. The system provides a quick and user-friendly approach for the ease staff to key in new products every time there is new changes and edit the existing products. Generally, this system will be developed using web-based technology to make the system more structured and systematic. Hypertext Preprocessor (PHP) language will be used to develop the system and the database will be using MySQL to store the data.

1.2 Problem Statement

The problem statement is the description of the problem that currently exist that needs to be addressed. The offline system in conducting inventory map in hypermarket have several problems:

1.2.1 Lack of monitoring and manage product inventory map.

Every time inventory had been changed, most of the buyers and some of the staff did not even know how to search things efficiently. This can take a lot of buyer times as they need to search rack by rack. The existing system cannot properly monitor products inventory.

1.2.2 Difficult for staff to keep the new inventory map for product on the system.

Some hypermarket did not even have a specific system to keep their full list of product inventory. The current system just changes the directory offline and did not have any reference in case any problems happen. There is no online or cloud information for all staffs refers when they want to add new product on the rack.

1.2.3 Difficult to search any products.

As for some new buyers in some hypermarket, they will face a common problem that is to recognize all the inventory products placing and the department. Using search button, buyers can easily key in the product name or the department and the place of the product will be appeared.

1.3 Objectives

i. To develop a system that can manage product inventory map of the hypermarket mall.

ii. To enable the staff to easily key in the updated inventory map for product

iii. To make the generation of products inserted by date

iv. To help buyers search for their desire products easily using one click.

1.4 Scope of The Project

1.4.1 Module to developed.

1.4.1.1 Admin Module

Admin can manage all the product inventory. Admins have the authority to add and delete staff. Admins have all the features that appear in staff module since admin are the one who will in charge of the system if there is anything happens.

1.3.1.1.1 Admin Profile Module

Admin can edit their name, phone number and email here. However, the username is not editable as it is unique and permanent.

1.3.1.1.2 Admin Manage Staff Module

Admin can add and delete new staff here. Admin need to fill the form to add new staff including their username and password. If the staff are no longer works in the hypermarket, admin can delete the staff account from the database.

1.3.1.1.3 New Product Module

This is where the admin will fill the information of new products arrive. A full information including barcode, row rack and rack number are require before it can be submitted.

1.3.1.1.4 Manage Product Module

Manage product module created for admin to edit existing information of product or the inventory map for the products.

1.3.1.1.5 Product B/w Module

Admin can search up the products depending on their registration date. This module can print out all the products between the date that admin have been choose.

1.3.1.1.6 Change Password Module

This module created for the admin to change their current password to new strong password.

1.3.1.1.7 Search Module

Admin can search up the products depending on the name or the brand of the product.

1.4.1.2 Staff Module

Staff module is to insert new product inventory, update, delete or making remark at any products. Staff also can view all the list of products in Inventory Map of Hypermarket.

1.3.1.1 Staff Profile Module

Staff can edit their name, phone number and email here. However, the username is not editable as it is unique and permanent.

1.3.1.1.2 New Product Module

This is where the staff will fill the information of new products arrive. A full information including barcode, row rack and rack number are require before it can be submitted.

1.3.1.1.3 Manage Product Module

Manage product module created for staff to edit existing information of product or the inventory map for the products.

1.3.1.1.4 Product B/w Module

Staff can search up the products depending on their registration date. This module can print out all the products between the date that admin have been choose.

1.3.1.1.5 Change Password Module

This module created for the admin to change their current password to new strong password.

1.3.1.1.6 Search Module

Staff can search up the products depending on the name or the brand of the product.

1.4.1.3 Buyer Module

In this module, buyer or shopper can register to login and can search any products that have been sell in the hypermarket. The user can update their profile and view their profile.

1.3.1.3.1 Sign in Module

New user can register themselves at this module.

1.3.1.2 User Profile Module

User can edit their name, phone number and email here. However, the username is not editable as it is unique and permanent.

1.3.1.1.3 All Product Module

This is where the full list of products will be printed. The user can see the full detail of every product on this module.

1.3.1.1.4 Product B/w Module

User can search up the products depending on their registration date. This module can print out all the products between the date that admin have been choose.

1.3.1.1.5 Search Module

User can search up the products depending on the name or the brand of the product.

1.4.2 Target User

Figure 1.1: Targeted User

SYSTEM SCOPE	EXPLANATION
Admin of Inventory Map of Hypermarket	• Admin can add and delete new staff. • Admin can add new product inventory map. • View the details of all products. • Manage the details products inventory. • Admin can put remark to the product. • Edit their profile. • Change the password of their profile.
Staff of Inventory Map of Hypermarket	• Insert details of new product inventory. • Update and delete details of product inventory. • Remark any products.

	• Edit their profile. • Change the password of their profile.
Buyer or user of Inventory Map of Hypermarket	• Login • View their details. • Search any product inventory.

1.5 Project Significance

According to the system that is developed, it can prevent the data of product inventory map from easily lost and damage since the existing system that have been used are using paper form. This system also to ease staff to key in new product inventory using the web application. This system can reduce time for both staff and buyer and ease the process of searching products and keep the track of the product inventory.

1.6 Expected Output

i. Output 1: To make the system more systematic.

ii. Output 2: Can trace the product inventory map.

iii. Output 3: Can reduce time from search products.

iv. Output 4: Hypermarket inventory map will be more systematic.

1.7 Conclusion

Overall, this chapter is important to develop a system that is more efficient, stable, and more functional to the user. Moreover, the system can be developed easily because the developer will understand all about the system such as problem statement, the objective and expected output of the system. This chapter help developer a lot to understand what should develop according to the real data given. To conclude, the new system can enhance the functionality and effectiveness of the current system.

Chapter 2 will explain about the literature review and project methodology of this project.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

Database system development is the process obtaining real-world requirements, analyzing requirements, designing the data and function of the system and the implementing the operations in the system. Database Life Cycle (DBLC) approach is applied as database development methodology to ensure the system has been developed efficiently. The phases of DBLS are discussed in section 2.2 Database Methodology. The life cycle moves to the next stage after the completion of each stage.

2.2 Database Development Methodology

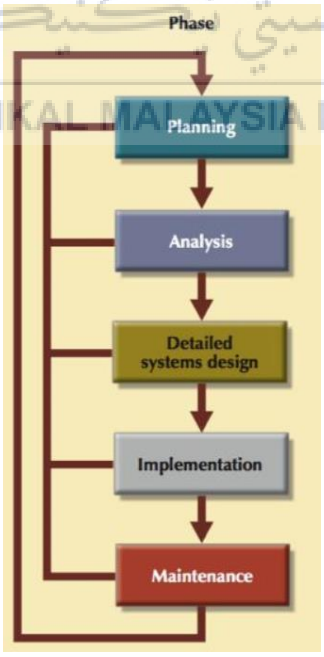

Figure 2.1: DBLC Phase

Figure 2.1 shows Database Life Cycle (DBLC) contains six phases. Starting with database initial study, database design, implementation, and loading, testing and evaluation, operation, and maintenance and evolution (THIRU, n.d.)

2.2.1 Database Initial Study

In this phase, the mission objectives of database project are defined. Mission statement defines major aims of database application while objective identify a particular task that the database support. Then, scope and boundaries of database project are defined. Knowing these helps in defining the required data structures, the type and number of entities and so on. The information of what the planned system supposed to function for now and in the future also gathered in this phase. In this phase, analyze of company situation is being made and defines problem and constraints.

2.2.2 Database Design

The second phase focuses on the design of the database model that will support safety inventory operations and objectives. In this phase the conceptual design the logical design and physical design are created and the DBMS software selection is also being prepared. Three phase of database design are conceptual database design, logical database design and physical database design. Conceptual database design is to build the conceptual representation of the database, which has the identification of the important entities, relationship and attributed. Data modeling assist in understanding data and facilitate communication about information requirement. Logical database design converts the conceptual representation to the logical structure of the database, which includes designing the relations. Physical database design decides how the logical structure is to be physically implemented, tailored to specific DBMS.

2.2.3 Implementation and Loading

This phase is the physical realization of the database and application design. Used DDL and DML to implement database in DBMS. After the database has been created, the data must be loaded into the database tables. The application is implemented using programming language.

2.2.4 Testing and evaluation

Testing and evaluation are the process of running the database system with intent of finding errors. The database is tested to ensure that it maintains the integrity and security of the data. This phase demonstrates that database and application program appear to be working according to requirements.

2.2.5 Operation

Once the database has passed the evaluation stage, it is operational. At that point, the database, its management, it's users and its application program constitute a complete information system. In this phase the required information flows such as all the queries are produced.

2.2.6 Maintenance and Evolution

The database should be maintained from time to time. Some of the required periodic maintenance activities include preventive and corrective maintenance (backup and recovery), adaptive maintenance and assignment of access permission for old and new users. In this phase, the database administrator must be prepared to perform routine maintenance activities within the database.

2.3 Project Schedule and Milestone

There are specific schedule and milestone that have been followed during implementation of PSM 1.

Table 2.1: Gantt Chart of Implementation PSM

WEEK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Proposal FYP Proposal assessment & verification Proposal correction																
List of supervisors																
Proposal Presentation and Submission Chapter 1 (System Development Begins)																
Report Chapter 1 & 2																
Report Chapter 1 & 2 Correction																
Progress Presentation Report Chapter 3 & 4																
Report 3 & 4 Correction																
MID TERM SEM BREAK																
Implementation Project Demo																
Project Correction Report FYP 1																
Report Correction																
Project Demo Updated Report FYP																
Final Presentation																
Correction Report FYP 1 Submission of Logbook																

2.4 Conclusion

As conclusion, this chapter discuss database development methodology for the project. The methodology is based on Database Life Cycle (DBLC) approach to develop system. Furthermore, the DBLC never end because database monitoring, modification and maintenance are part of the life cycle and these activities continue long after a database has been implemented. So, the DBLC encompasses the lifetime of the database. For the next chapter (Chapter 3) is about project analysis, will be discussed into details. The problem analysis, proposed improvements or solutions, non-functional requirements, functional requirements, and other requirements will be included.

CHAPTER 3: ANALYSIS

3.1 Introduction

The analysis stage involves critical evaluation of the database development planning carried out beforehand. In this stage, developer assess the plan of database development against the elements like cost, time-period, development platform, programming languages, and forecasted development results, to analyze the effectiveness associated with the planned database system. This phase involves analyzing the current system and the system to be developed.

3.2 Problem Analysis

As I want the best for my system, I went to Giant Hypermarket at Klang, Selangor to observe the system that they used there. Giant Hypermarket are on of the largest hypermarket in Malaysia. They even have launched near 2000 new products and have hundreds of rows of racks in their building. This hypermarket focusses on two types of users that is staff and shopper and both users have a lot of business rules that need to be go thoroughly. When new products come in, staff need to manually check the products and add it up to rows and have a lot of documents and papers if the structure of rows have been changed. Some new shopper will face a lot of issues as they need to go through every rack and rows one by one just to search for their desire products. Through Inventory Map for Hypermarket, shopper can search their products and immediately get the exact position of the product.

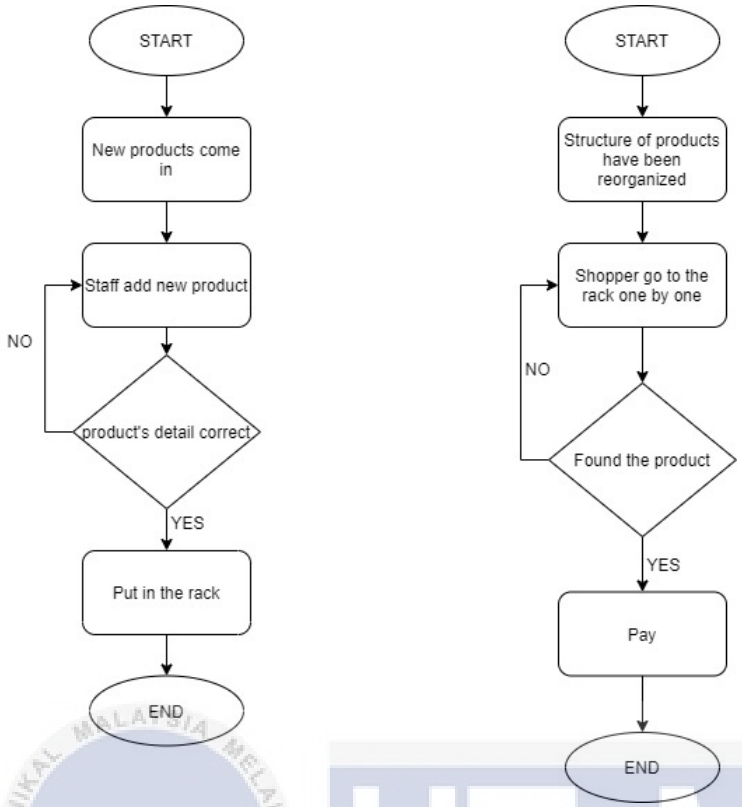

Figure 3.1: Flowchart of the current system

Figure 3.1 shows the flowchart of the current system in hypermarket in Malaysia.

3.3 The Proposed Improvements/ Solutions

New system proposed to help shopper easily find the products that they find within a minute. This system developed for make ease the staff and shopper job without using any paper.

3.4 Requirement Analysis of the To-Be-System

3.4.1 Functional Requirement (Process Model)

3.4.1.1 Context Diagram

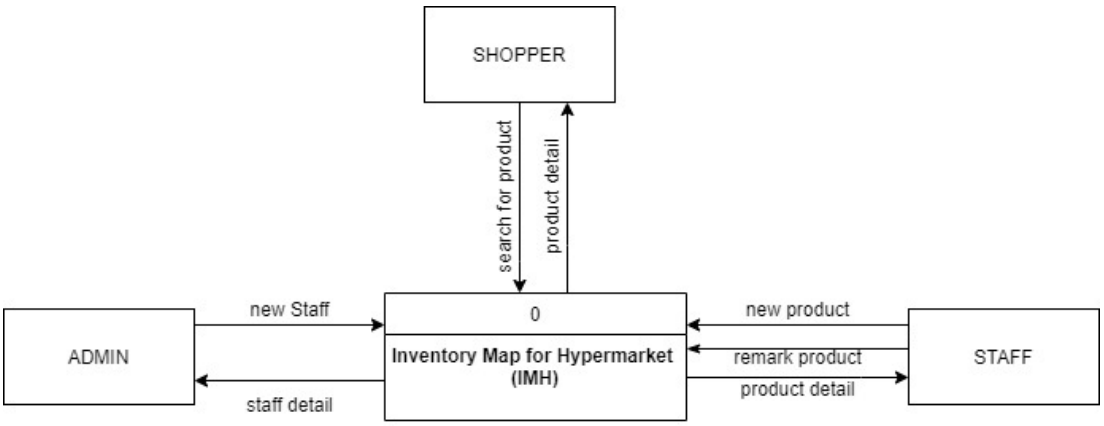

Figure 3.2: IMH Context Diagram

Figure above shown the context diagram for the Inventory Map for Hypermarket. Admin can add new staff and view the detail of the staff. Staff can add new product, remark product, and get to view the details of the products. Shopper can search for the product detail and will get to see the details of the products.

3.4.1.2 Data Flow Diagram Level 0

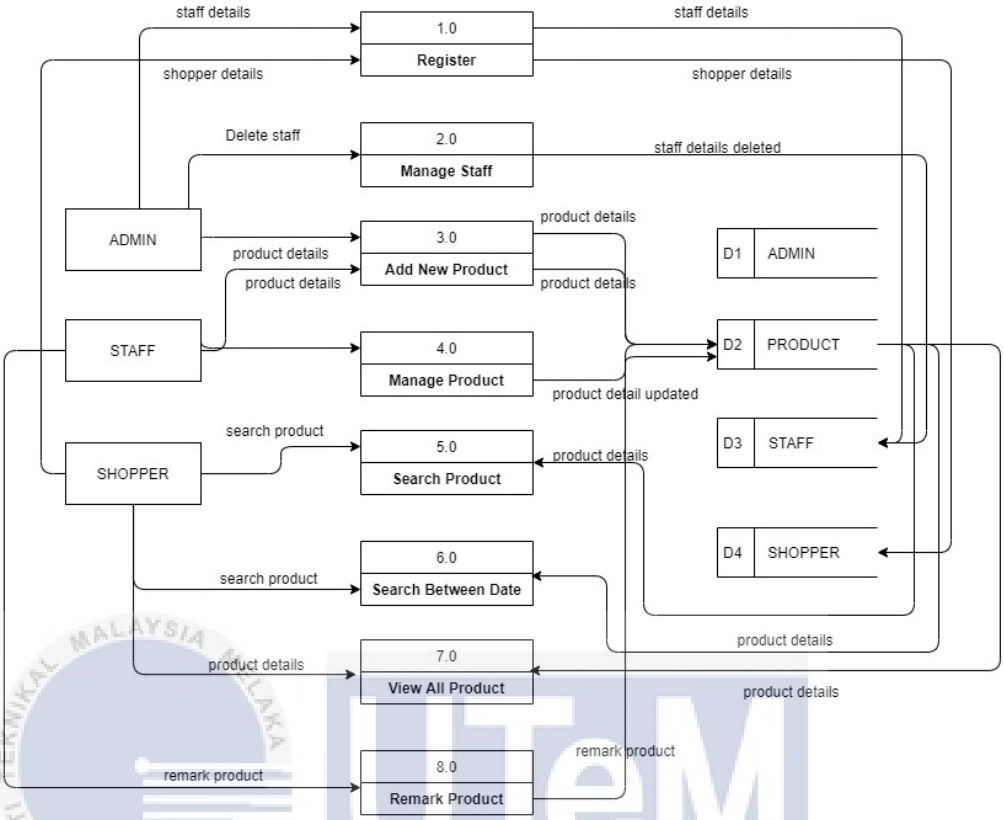

Figure 3.3: IMH Data Flow Diagram Level 0

Figure 3.3 shows the data flow diagram for Inventory Map for Hypermarket. This system consists of three main users: admin, staff and shopper.

3.4.1.3 Data Flow Diagram Level 1

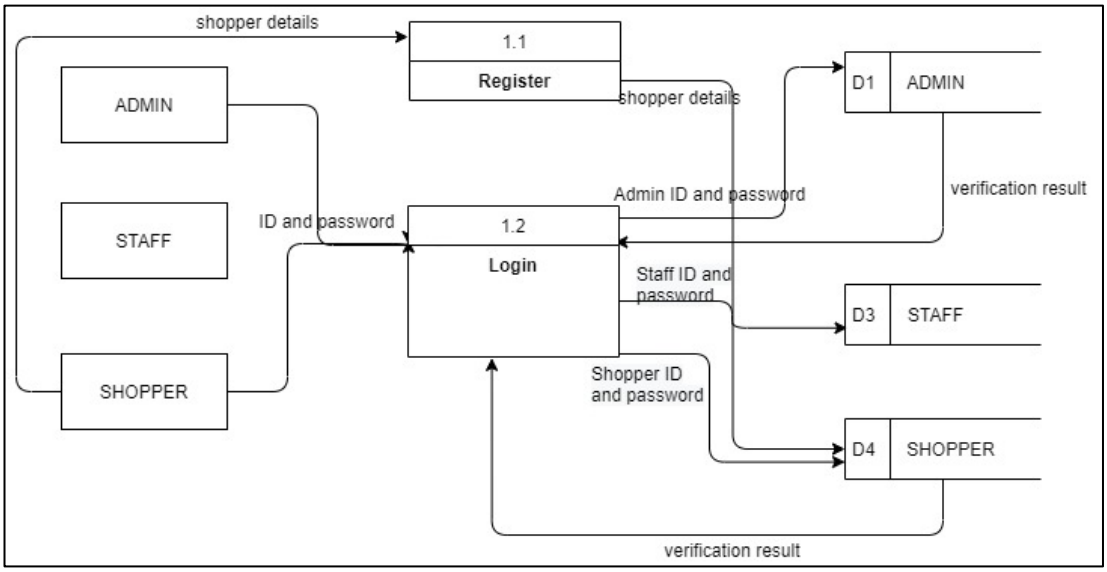

Figure 3.4: Register Process Level 1

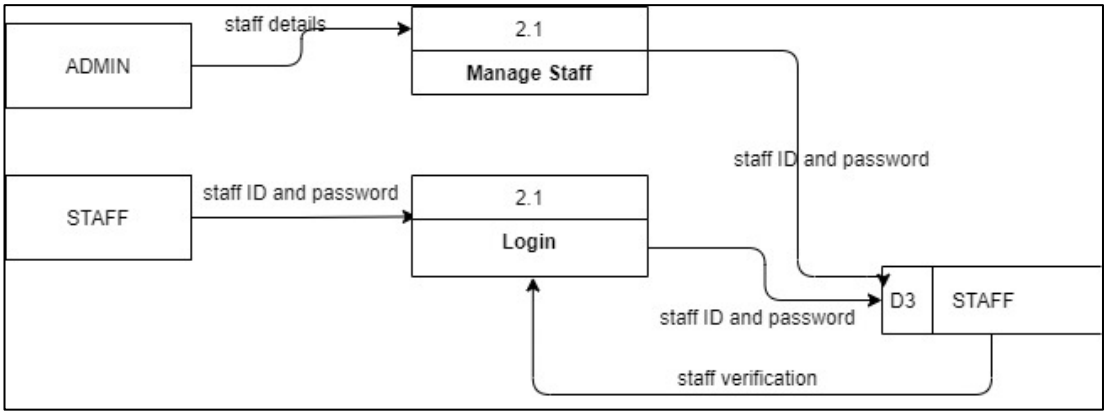

Figure 3.5: Manage Staff Process Level 1

Figure 3.6: Add New product Process Level 1

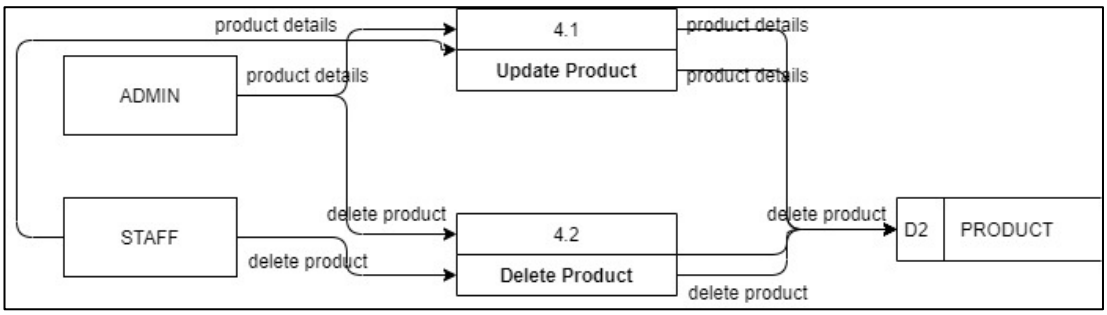

Figure 3.7:Manage Product Process Level 1

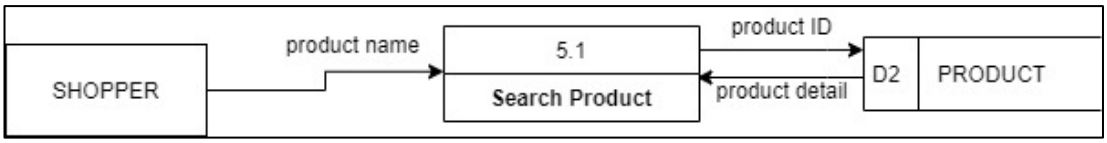

Figure 3.8: Search Product Process Level 1

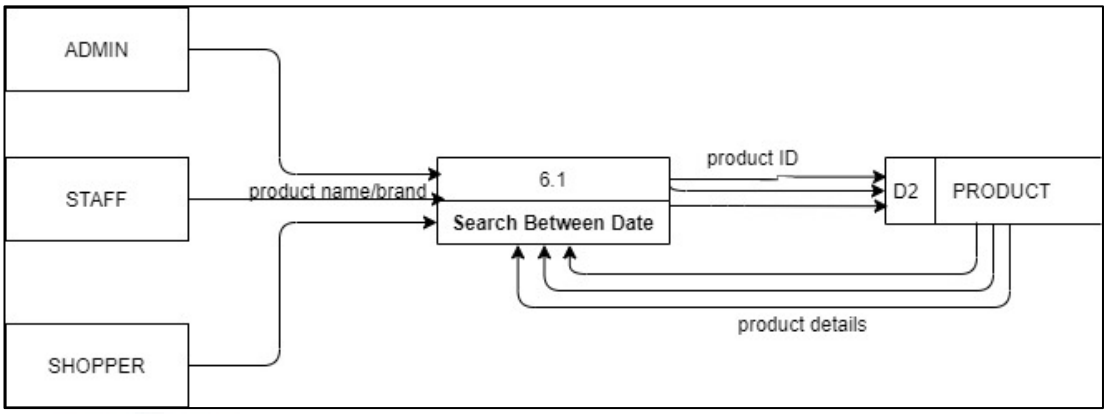

Figure 3.9: Search Between Date Process Level 1

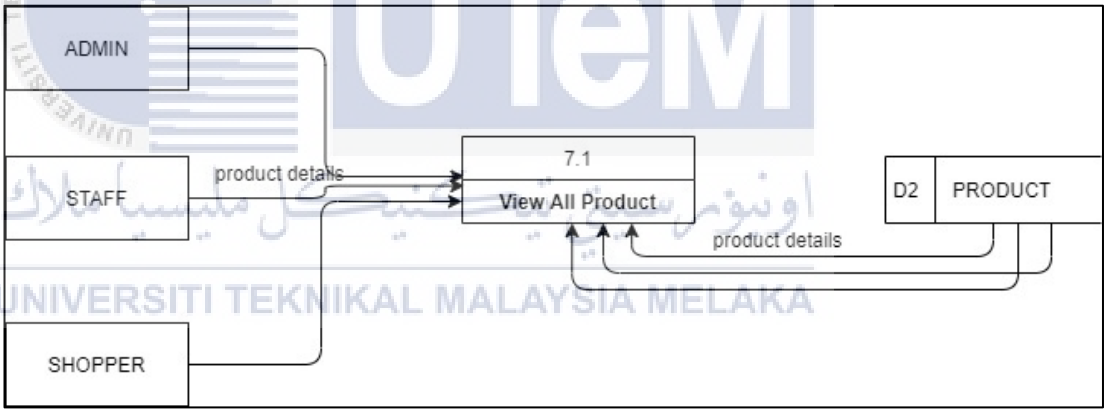

Figure 3.10:View All Product Process Level 1

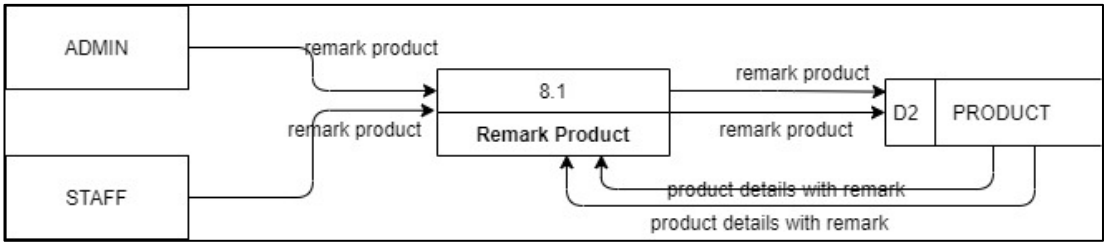

Figure 3.11: Remark Product Process Level 1

3.4.2 Non-functional Requirements

3.4.2.1 Maintainability

Amount of effort required to maintain and enhance application solution in production.

3.4.2.2 Access Security

i. Staff can only be registered by the admin.

ii. Password shall never be viewable at the point of entry or at any other time.

3.4.3 Other Requirements

3.4.3.1 Software Requirements

3.4.3.1.1 Documentation

i. Microsoft Office 365

Figure 3.12: Microsoft Office 365 Logo

Office 365 is a version of Office that's available Office 365. This product contains Access, Excel, OneNote, Outlook, PowerPoint, Publisher, Skype for Business and Word. Microsoft Word is a word processor software that used in this project for documentation purpose.

ii. Draw.io

Figure 3.13: Draw.io Logo

Draw.io is a free diagramming application that allows users to create and share diagrams within a web browser. Users can create and edit a variety of diagrams including flowcharts, org charts, process diagrams, ER diagrams, UML, network diagrams, and more.

3.4.3.1.2 Programming Tools

i. Sublime Text 3

Figure 3.14: Sublime Text 3 Logo

Sublime Text is a shareware cross-platform source code editor with a Python application programming interface. It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

3.4.3.1.3 Programming Language

i. PHP

Figure 3.15: PHP language Logo

PHP is a server-side scripting language. that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages. PHP is open source and free. PHP has in built support for working hand in hand with MySQL; this doesn't mean you can't use PHP with other database management systems. You can still use PHP with Postgres, Oracle, MSSQL Server, ODBC and etc. PHP is cross platform this means you can deploy your application on a number of different operating systems such as windows, Linux, Mac OS etc.

ii. HTML

Figure 3.16: HTML 5 language Logo

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as PHP.

iii. CSS

Figure 3.17: CSS language Logo

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and PHP.

iv. Java Script

Figure 3.18: Java Script language Logo

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

3.4.3.1.4 DBMS

i. MySQL

Figure 3.19: MySQL Logo

MySQL is an Oracle-backed open-source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing. MySQL is an important component of an open-source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

ii. phpMyAdmin

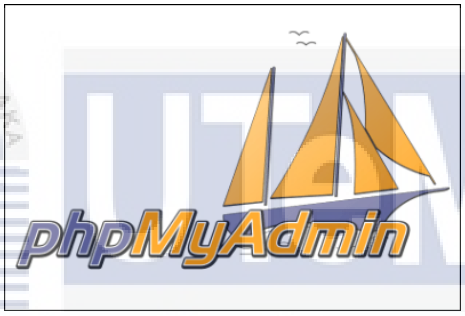

Figure 3.20: phpMyAdmin Logo

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

3.4.3.1.5 Operating System

Figure 3.21: Windows 10 Logo

Microsoft Windows is a group of several graphical operating system families, all of which are developed, marketed and sold my Microsoft. The version that used is Windows 10.

3.4.3.1.6 Server

Figure 3.22: XAMPP Logo

Xampp is a free and open-source cross-platform web server software for installing and using the Apache Web server. It has provided component including MYSQL database, PHP, HTTPS (SSL), Common gateway interface (CGI), Server side includes (SSI).

3.4.3.2 Hardware Requirements

The hardware that used in this project is laptop. Below listed the hardware requirement in developing system.

Table 3.1: Hardware Requirements for PSM 1

COMPONENT	SPECIFICATION
Microprocessor	(1.1 GHz base frequency, up to 2.6 GHz burst frequency, 4 MB cache, 2 cores)

Hard Drive	500 GB 5400 rpm
Keyboard	Full-size island-style keyboard with numeric keypad
Display	15.6" diagonal HD SVA BrightView WLED-backlit (1366 x 768)

Table 3.1 shows the hardware requirements for developing this project. The processor used is Intel Core i7 and the memory of the hardware is 4GB DDR4.

3.5 Conclusion

Problems of the current existing system has been analysed during analysis phase. This chapter discusses the project requirement which includes the software requirements, hardware requirements and other requirements. In conclusion, the deliverable of this chapter which are the flow chart of the current system, flow chart of the proposed system, context diagram and data flow diagrams can help to achieve a good analysis. The structure and meaning of the data in the organizations can be understood easily by reviewing this analysis. All this analysis will be used as a guideline to design the database.

Chapter 4 will explain about the design phase for this system.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter summarizes design phase is concerned with the physical construction of the system. Included are the design or configuration of the network such as hardware, operating system, programming, design of user interfaces, design of system interfaces and security issues. It is important that the proposed design be tested for performance, and to ensure that it meets the requirements outlined during the analysis phase. In other words, the main objective of this phase is to transform the previously defined requirements into a complete and detailed set of specifications which will be used during the next phase.

4.2 Introductory Preview to This Chapter

4.2.1 Database Design

This phase consists of three parts which is the conceptual design, the logical design, and the physical design. Some methodologies merge the logical design phase into the other two phases. This section is not aimed at being a definitive discussion of database design methodologies (there are whole books written on that!); rather it aims to introduce you to the top.

4.2.1.1 Conceptual Design

Conceptual design phase is to build a conceptual model based upon the previously identified requirements, but closer to the final physical model.

4.2.1.1.1 Entity Relationship Diagram

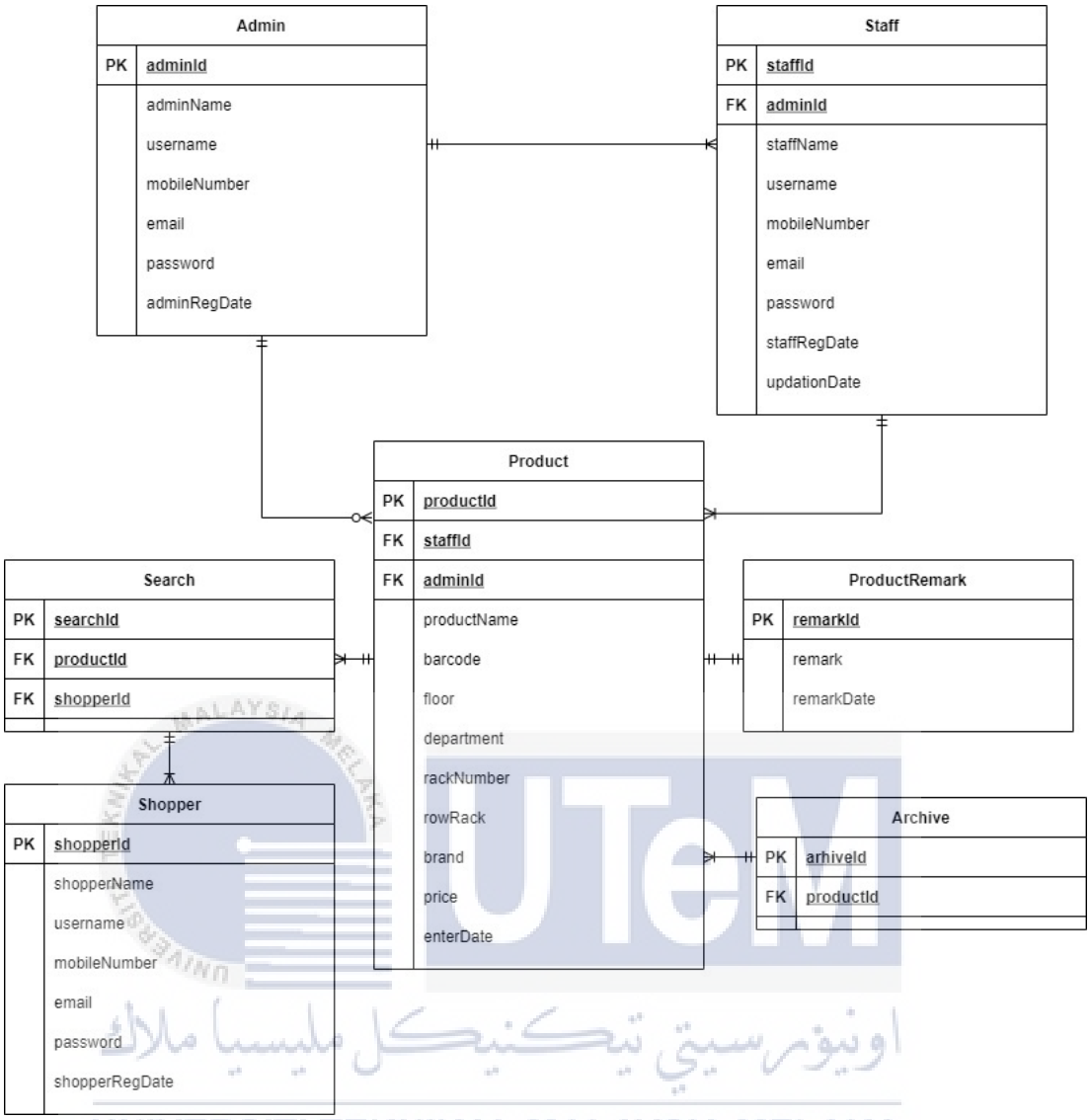

Table 4.1: IMH Entity Relationship Diagram (ERD)

4.2.1.1.2 Business Rules

Business Rule	Description
1	Each staff will be registered by the admin.
	One admin can register more than one staff.
2	Each username and password are unique.
	Password should consist of letters, numbers, and special characters with minimum 6 characters.

3	Each staff can add new product in the system.
	The staff can add more than one product.
4	Each staff can edit and delete products.
	Edit product will be use if there are changes in navigation map while delete product will be use when the product has been stopped produce.
5	Each staff can search product between selected dates.
	The detail of the product will be appeared.
6	Each user can edit some of their information including full name, phone number and email.
	They cannot edit their username as it is unique and different for each user.
7	Each user can reset their password by entering their current password and their new password.
8	Product that has been deleted by admin or staff will be save in archive.
	After 24 hours, the product will be permanently deleted form table.
9	Admin can delete staff who did not work with the company anymore.
	The list of the staff will be deleted permanently.

4.2.1.2 Logical Design

4.2.1.2.1 Data Dictionary

Table	Column	Data Type	Constraint	Reference Table
Admin	adminId	Int(11)	Primary Key	-
	adminName	Varchar(100)	Not Null	-
	username	Varchar(100)	Not Null	-
	mobileNumber	Bigint(11)	Not Null	-
	email	Varchar(100)	Not Null	-
	Password	Varchar(100)	Not Null	-
	adminregDate	timestamp	Not Null	-
Staff	staffId	Int(11)	Primary Key	-
	adminId	Int(11)	Foreign Key	Admin
	staffname	Varchar(100)	Not Null	-
	username	Varchar(100)	Not Null	-
	mobileNumber	Bigint(11)	Not Null	-
	email	Varchar(100)	Not Null	-
	password	Varchar(100)	Not Null	-
	staffRegDate	timestamp	Not Null	-
	updationDate	timestamp	Not Null	-
Shopper	shopperId	Int(11)	Primary Key	-
	shoppername	Varchar(100)	Not Null	-
	username	Varchar(100)	Not Null	-
	mobileNumber	Bigint(11)	Not Null	-
	email	Varchar(100)	Not Null	-
	password	Varchar(100)	Not Null	-
	shopperRegDate	timestamp	Not Null	-
	productId	Int(11)	Primary Key	-
	adminId	Int(11)	Foreign Key	Admin

Product	staffId	Int(11)	Foreign Key	Staff
	productName	Varchar(100)	Not Null	-
	barcode	Int(11)	Not Null	-
	floor	Varchar(100)	Not Null	-
	department	Varchar(100)	Not Null	-
	rackNumber	Varchar(100)	Not Null	-
	rowRack	Varchar(100)	Not Null	-
	brand	Varchar(100)	Not Null	-
	price	Varchar(100)	Not Null	
	enterDate	timestamp	Null	

ProductRemark	remarkId	Int(11)	Primary Key	-
	remark	Varchar(100)	Not Null	-
	remarkDate	timestamp	Null	-

Search	searchId	Int(11)	Primary Key	-
	productId	Int(11)	Foreign Key	Product
	shopperId	Int(11)	Foreign Key	Shopper

Archive	archiveId	Int(11)	Primary Key	-
	productId	Int(11)	Foreign Key	Product

4.2.1.3 Physical Design

4.2.1.3.1 Data Dictionary Language

TABLE	**DDL**
Admin	CREATE TABLE `admin` (`adminId` int(11) NOT NULL, `AdminName` varchar(100) NOT NULL, `Username` varchar(100) NOT NULL, `MobileNumber` bigint(11) NOT NULL, `Email` varchar(100) NOT NULL,

	`Password` varchar(100) NOT NULL, `AdminRegDate` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Archive	CREATE TABLE `archive` (`archiveId` int(11) NOT NULL, `productId` int(11) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Product	CREATE TABLE `product` (`productId` int(11) NOT NULL, `ProductName` varchar(255) NOT NULL, `Barcode` int(11) NOT NULL, `Floor` int(100) NOT NULL, `Department` varchar(100) NOT NULL, `RackNumber` varchar(100) NOT NULL, `RowRack` varchar(100) NOT NULL, `Brand` varchar(100) NOT NULL, `Price` varchar(255) NOT NULL, `EnterDate` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(), `remark` varchar(100) DEFAULT NULL, `outtime` timestamp NULL DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
productRemark	CREATE TABLE `productremark` (`id` int(11) NOT NULL, `productId` int(11) NOT NULL, `status` varchar(255) NOT NULL, `remark` mediumtext NOT NULL, `remarkDate` timestamp NOT NULL DEFAULT current_timestamp()) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Shopper	CREATE TABLE `shopper` (`shopperId` int(11) NOT NULL,

	`shopperName` varchar(100) NOT NULL, `username` varchar(100) NOT NULL, `mobileNumber` bigint(11) NOT NULL, `email` varchar(100) NOT NULL, `password` varchar(100) NOT NULL, `shopperRegDate` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(), `updationdate` timestamp NULL DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Staff	CREATE TABLE `staff` (`staffId` int(11) NOT NULL, `adminId` int(11) DEFAULT NULL, `staffName` varchar(100) NOT NULL, `username` varchar(100) NOT NULL, `mobileNumber` bigint(11) NOT NULL, `email` varchar(100) NOT NULL, `password` varchar(100) NOT NULL, `staffRegDate` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(), `updationdate` timestamp NULL DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

4.2.1.3.2 The Usage of Stored Procedures, Triggers, and Other Related Database

Procedure
RemarkProduct
	BEGIN
	SELECT *
	FROM productremark
	WHERE status = seqStatus;
	END
	
	call productremark();

Trigger	$sql1= ”CREATE OR REPLACE TRIGGER staffId

	BEFORE INSERT ON staff FOR EACH ROW SET NEW.staffId = ‘S00’;”;

4.3 Graphical User Interface (GUI) Design

- **Shopper**

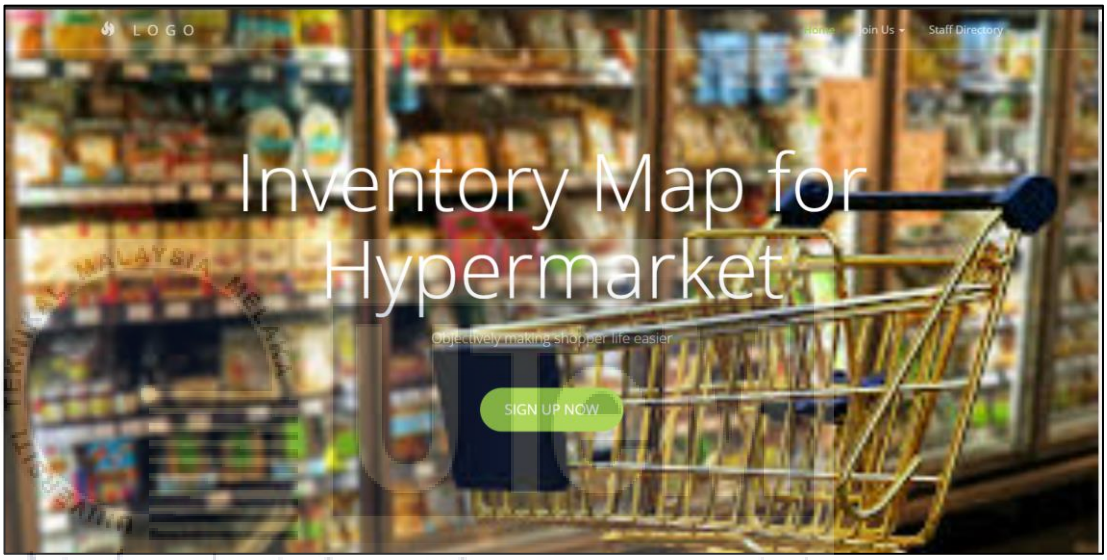

Figure 4.1: Main Page for IMH

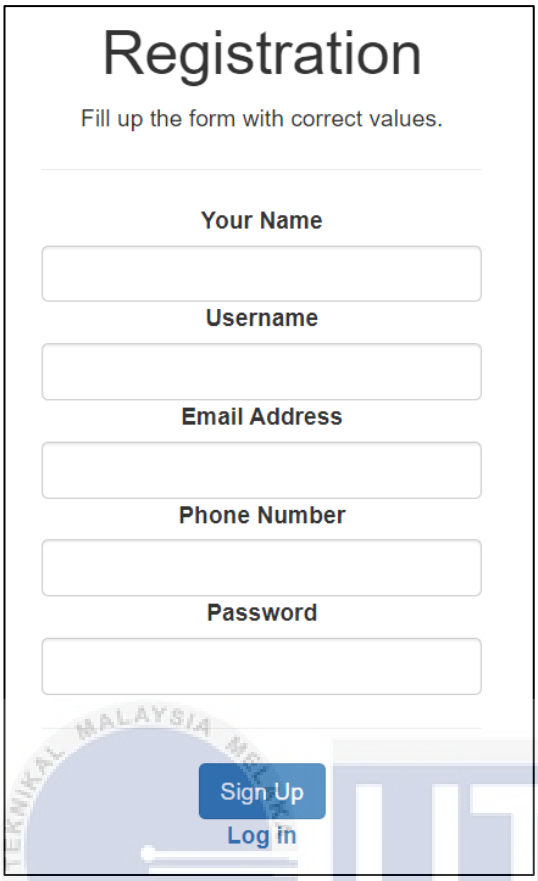

Figure 4.2: Registration Page for IMH

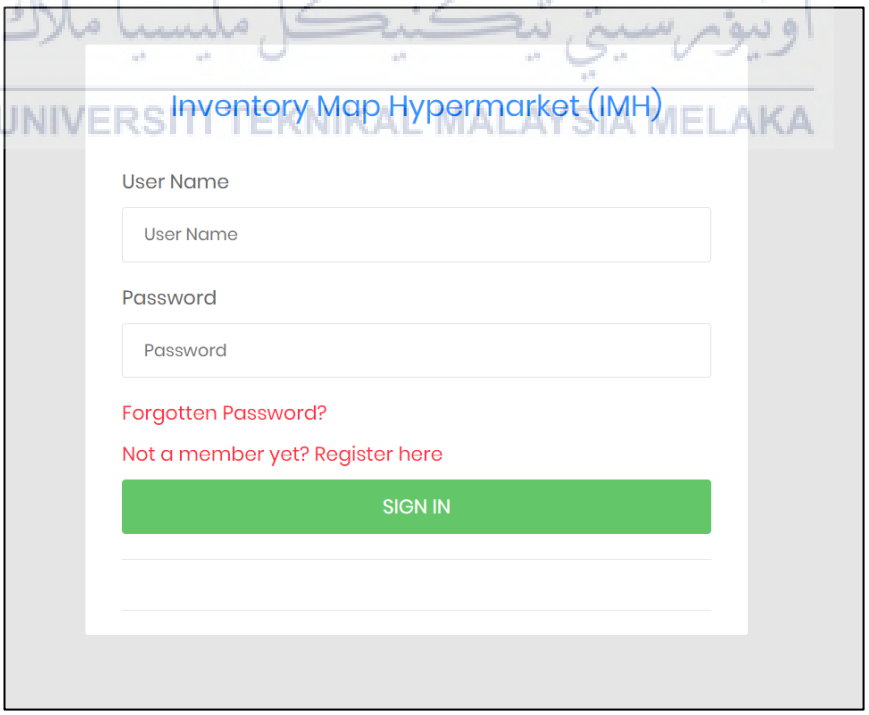

Figure 4.3: Login Page for IMH

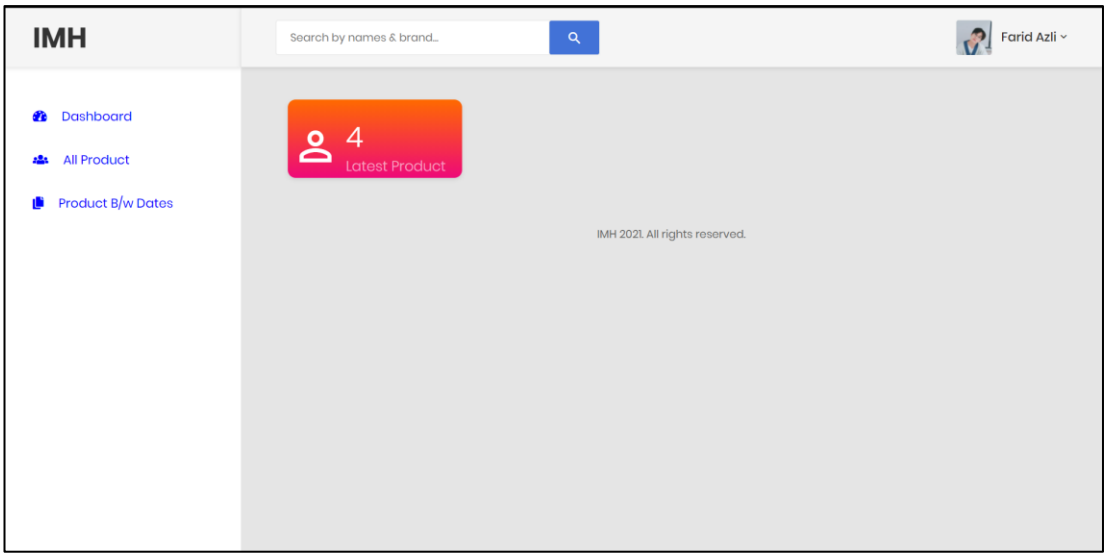

Figure 4.4: Dashboard Page for shopper for IMH

Figure 4.5: All Product Page for shopper for IMH

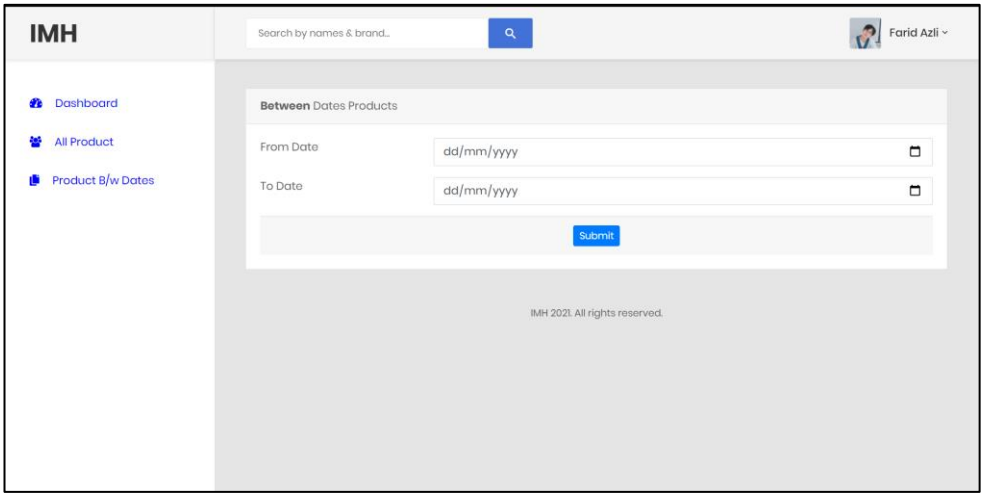

Figure 4.6: Product B/w Dates Page for shopper for IMH

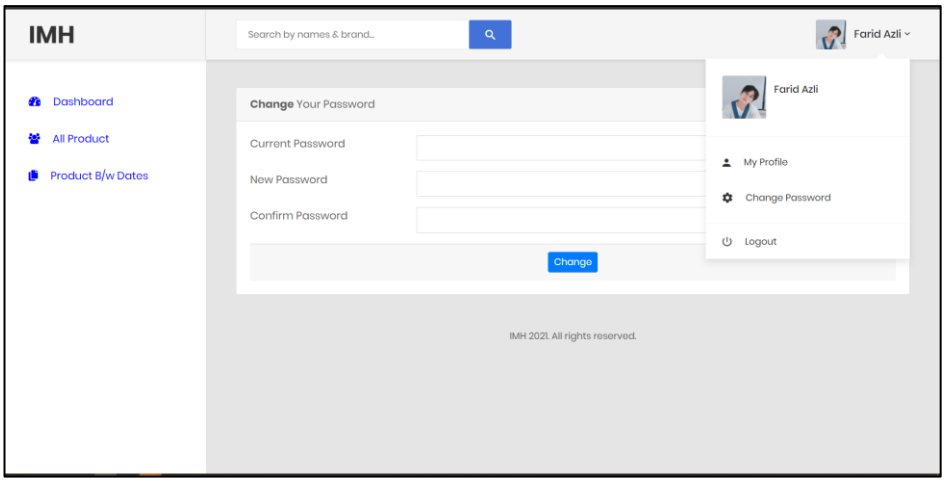

Figure 4.7: Navigation Bar Page for shopper for IMH

Figure 4.8: Update Profile Page for shopper for IMH

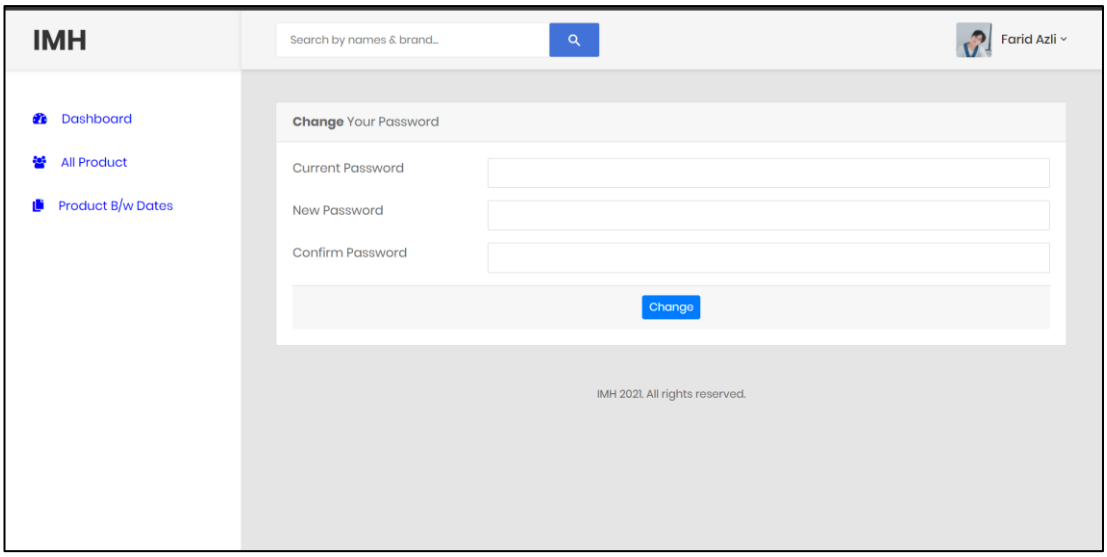

Figure 4.9: Change Password Page for shopper for IMH

- **Admin Module**

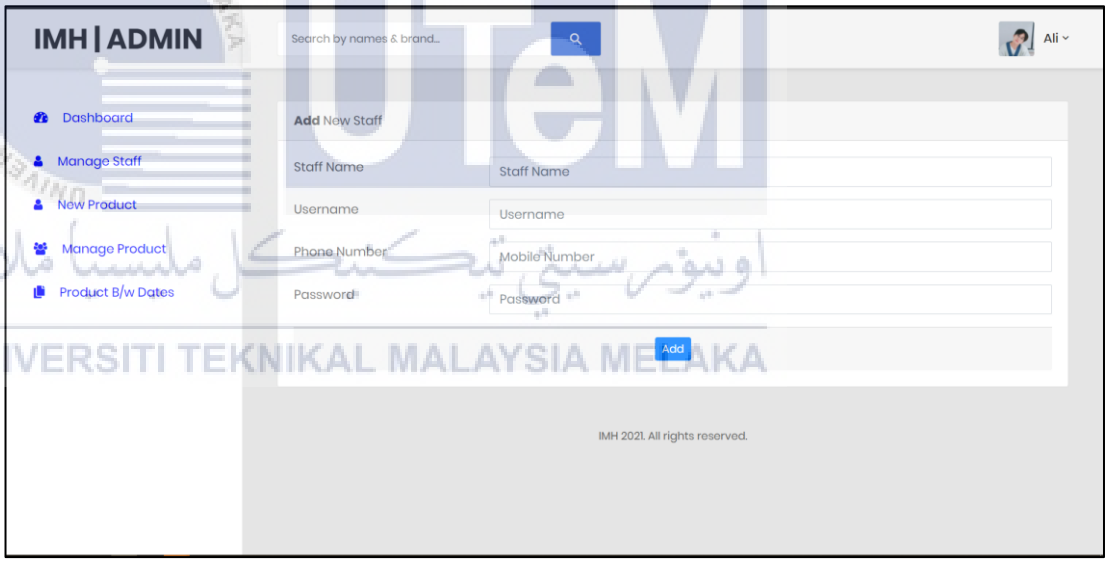

Figure 4.10: Add New Staff Page for admin for IMH.

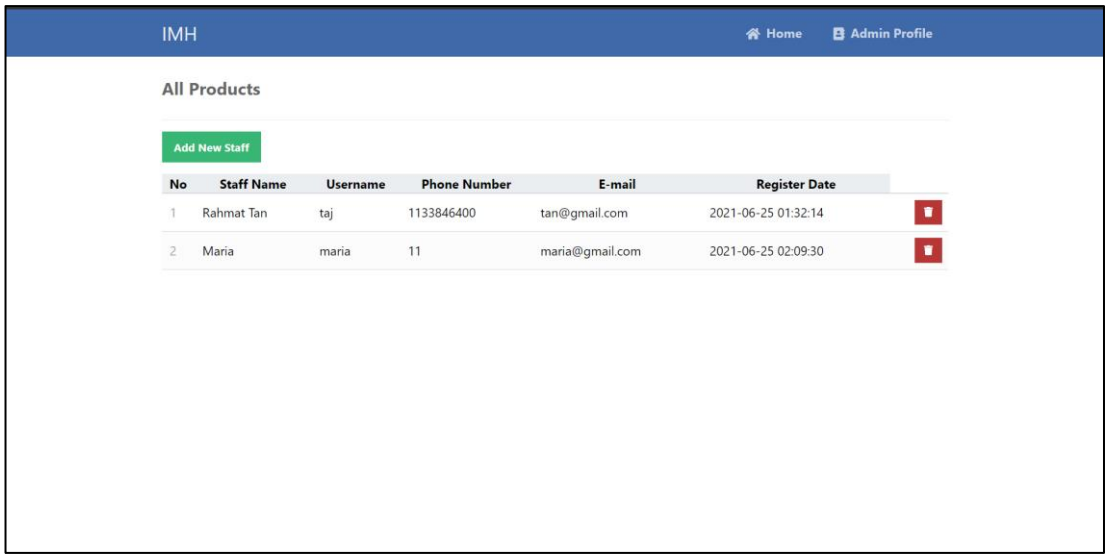

Figure 4.11: Manage Staff Page for admin for IMH.

- **Staff Module**

Figure 4.12: Add New Product Page for staff for IMH.

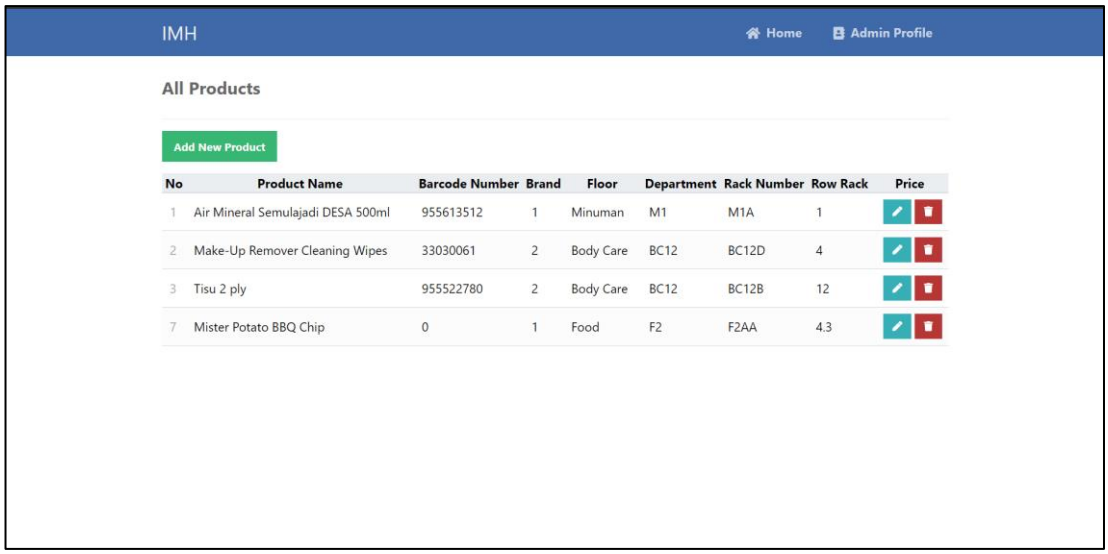

Figure 4.13: Manage Product Page for staff for IMH.

Figure 4.14: All Product Page in Admin and Staff

4.4 Conclusion

In conclusion, designing the database is important because this can provide a solution to the problems specified in the requirement document in analysis phase. The design document for this system act as a plan or blueprint for the solution and will be used later for implementation, testing and maintenance. Based on the logical, physical and data dictionary that are provided in this document, overall, the database structure are created with the specific modules that will be developed in MySQL database using

phpMyAdmin. The relationship between each entity that depends on each other will be shown by an overall view of the conceptual design.

Chapter 5 will explain about the implementation phase of this project.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter discusses about the implementation of software development environment setup and system database implementation. The software developing environment will be explained the steps of installation of the software, executing and configuring the software and database.

For the database implementation, it will be explained about DBMS that was the schedule, which is MySQL Database using phpMyAdmin with XAMPP Server, the Data Definition Language (DDL), Data Manipulation Language (DML) and lastly about the main processes such as stored procedures and trigger that implemented on this system using PLSQL programming language.

5.2 Software Development Environment Setup

The software that is used for the development process of the Inventory Map for Hypermarket is the Sublime Text 3 as a platform to write a PHP code. Then for database management, this system was used MySQL Database to store the data and the database object using phpMyAdmin to handle the administration of MySQL. Lastly, for the server platform, this system used XAMPP Server to be a server that will handle the database and the system together.

5.2.1 Software Environment Setup XAMPP Server

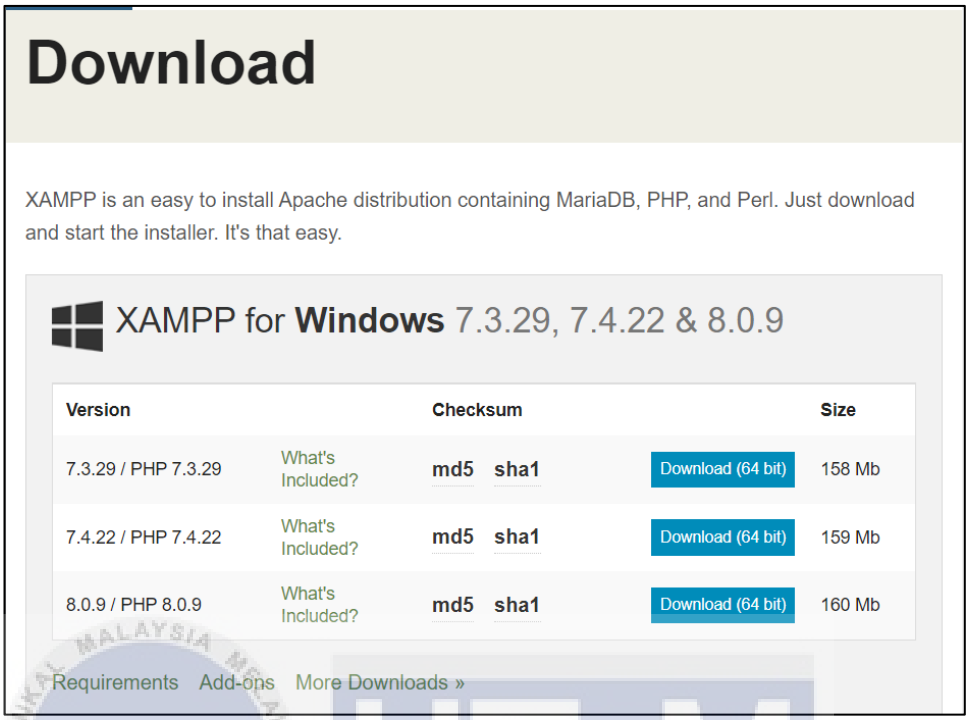

Figure 5.1: XAMPP Server Web Sites Download Folder

Step 1: Go to https://www.apachefriends.org/download.html and start to download the XAMPP Server installer. Figures 5.1 show the web site to download the XAMPP Server installer. Make sure before downloading, choose the right version of the XAMPP Server installer that will be used on the computer configuration either 64 bits (x64) or 32 bits (x86), and also choose the right operating system for the computer.

Figure 5.2: User Account Control (UAC)

Step 2: Click 'OK' to continue the installation if windows pop up warning about User Account Control (UAC).

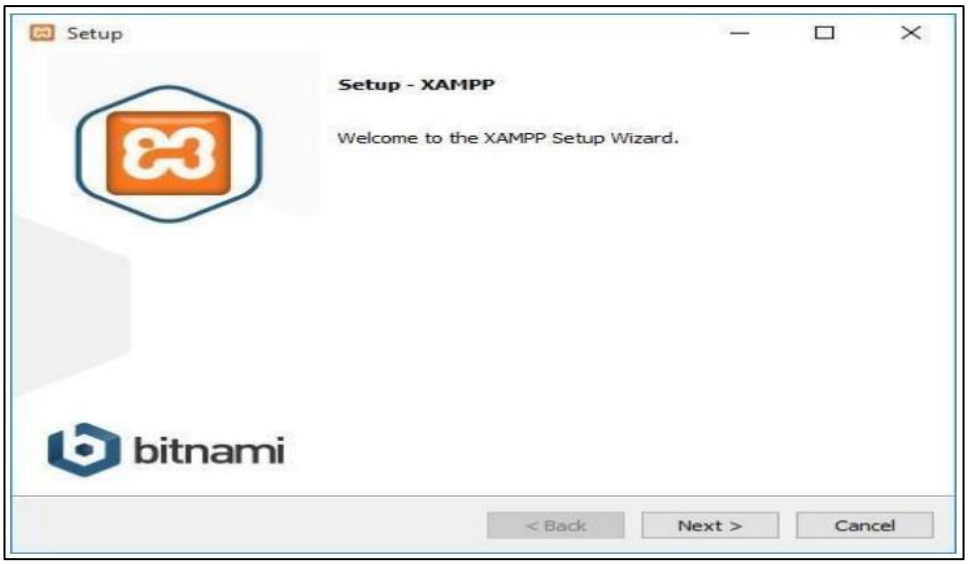

Figure 5.3: XAMPP Setup Wizard

Step 3: Next, Welcome to the XAMPP Setup Wizard will appear. Click 'Next' to continue the installation.

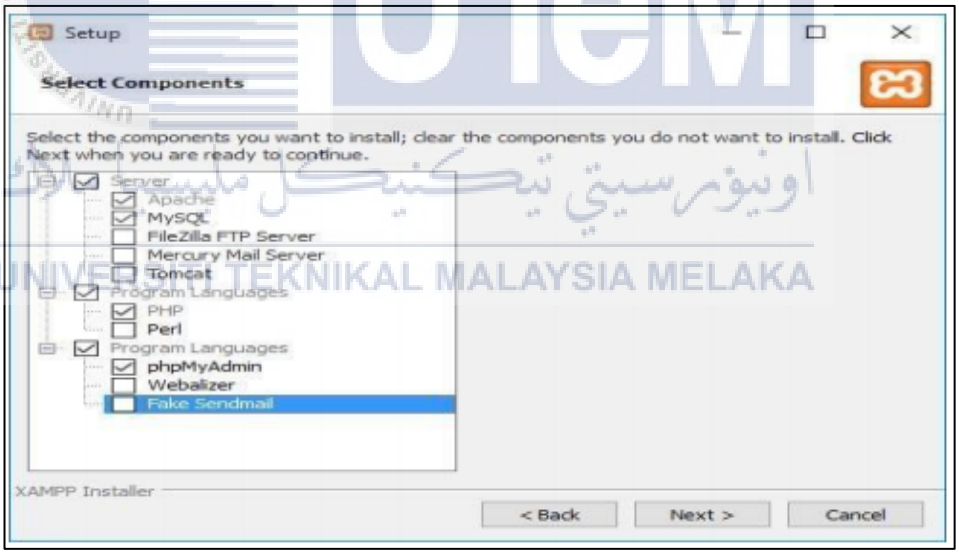

Figure 5.4: Components Selection Interface

Step 4: The components selection screen will appear next. This screen asked to make the selection which components of the software would like to install and which ones do not want. Click 'Next' to continue.

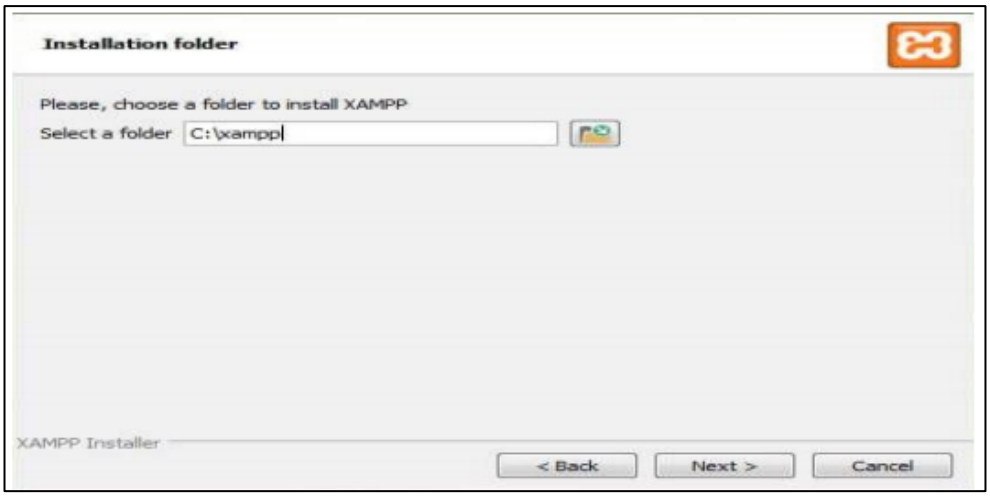

Figure 5.5: Choose the Installation Directory

Step 5: In this next step, users need to choose the directory for the XAMPP software packet to be installed. If the user chooses for the standard setup, then the folder name XAMPP will be created under C:\. To continue the installation, click 'Next'.

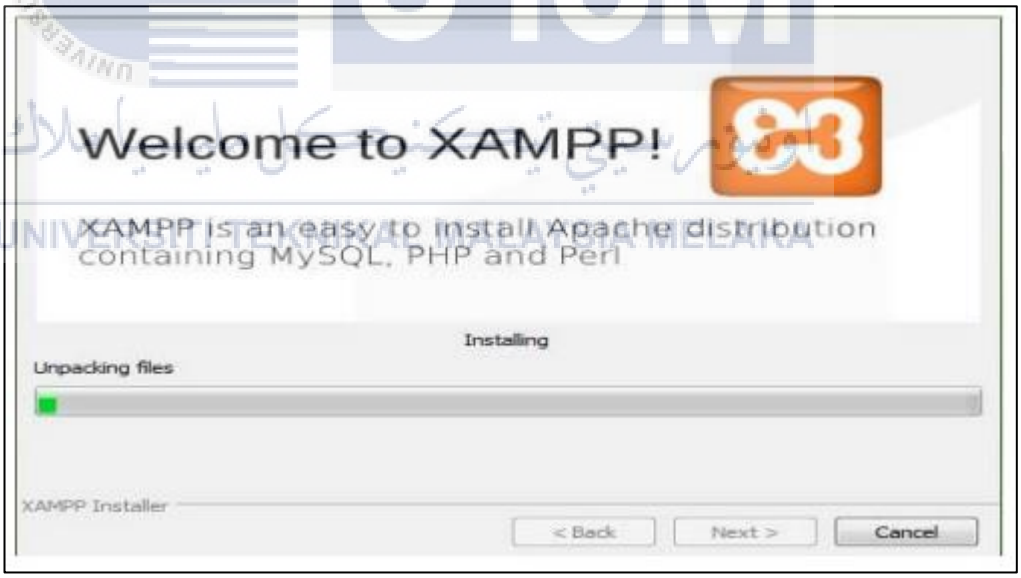

Figure 5.6: Installation Process Started

Step 6: XAMPP will be started installing the files to the location selected in the previous step.

Figure 5.7: Complete Installation

Step 7: After all the components are unpacked and installed, users can click on 'Finish'. Click to tick the corresponding and open the XAMPP Control Panel.

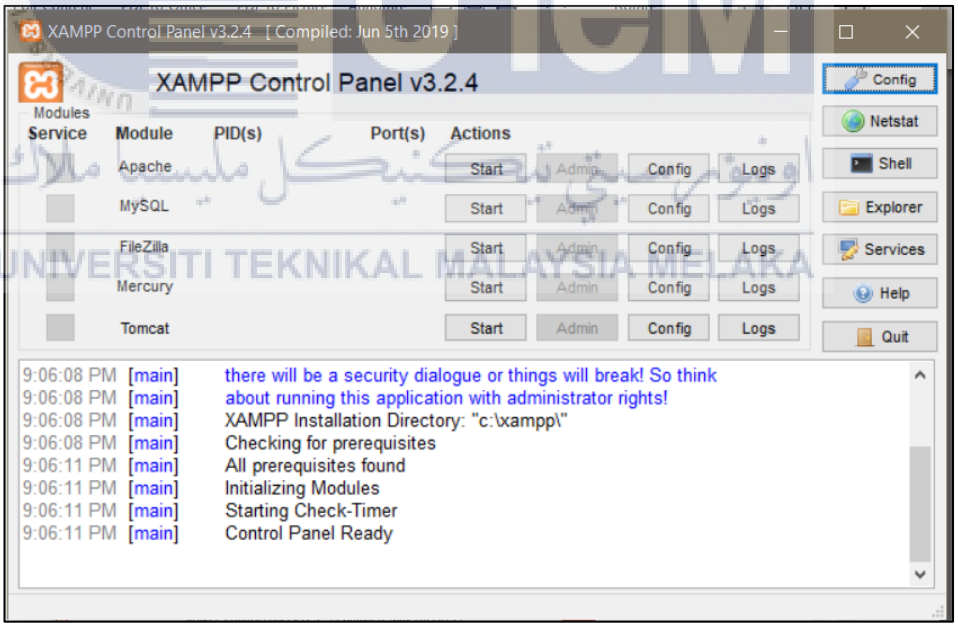

Figure 5.8: XAMPP Control Panel

Step 8: The Control Panel will automatically open, but if users unchecked the option in the previous window, users can search the XAMPP folder and open the XAMPP Control Panel manually.

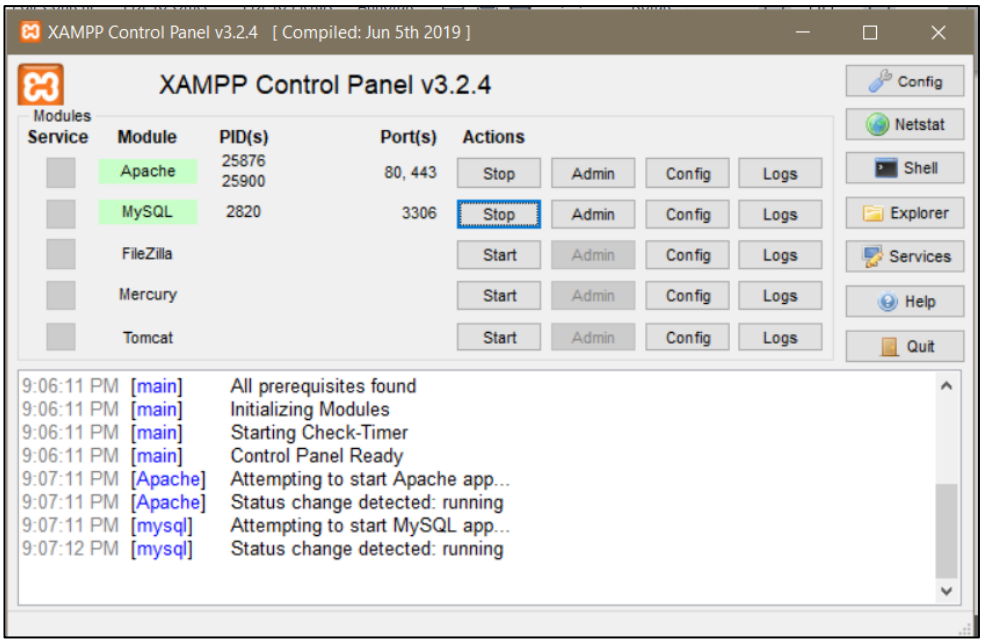

Figure 5.9: Starting XAMPP Modules

Step 9: Modules can be started or stopped on the XAMPP Control Panel on the buttons under 'Actions'. Users can detect which modules have been started because the names are highlighted green under the 'Module' title.

5.2.2 Database Environment Setup

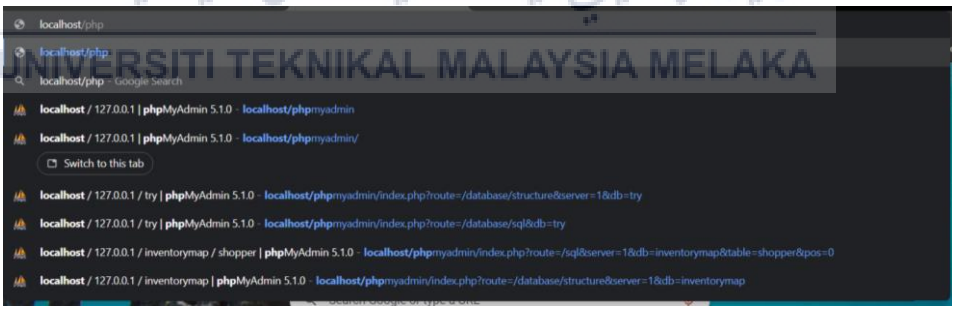

Figure 5.10: phpMyAdmin Link Search

Step 1: Users can access phpMyAdmin by entering http://localhost/phpmyadmin/ in the web server. The phpMyAdmin first screen will appear.

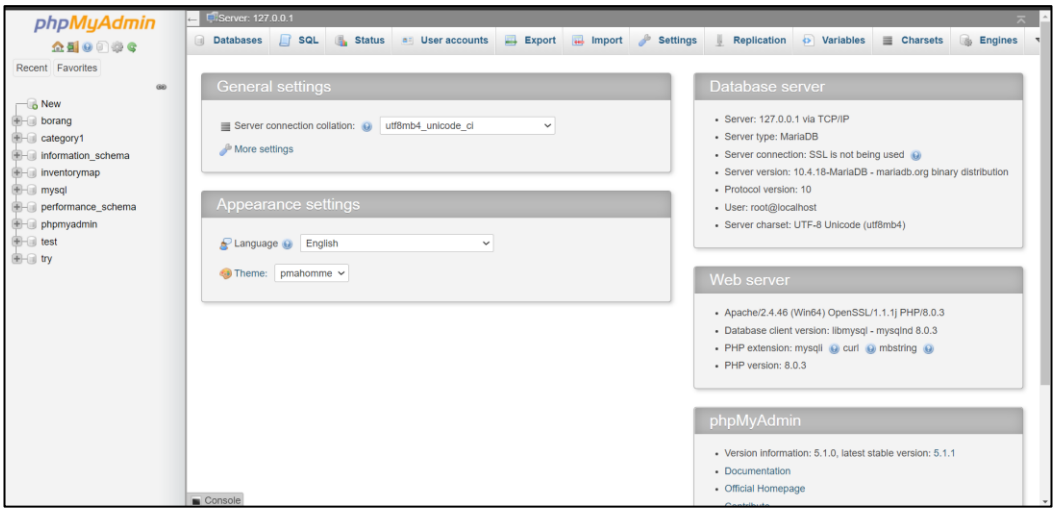

Figure 5.11: phpMyAdmin First Page

Step 2: The main phpMyAdmin first-page screen will appear. On the left is a list of databases that already exist. Users need to create a new database, click Databases in the top navbar, or click the SQL.

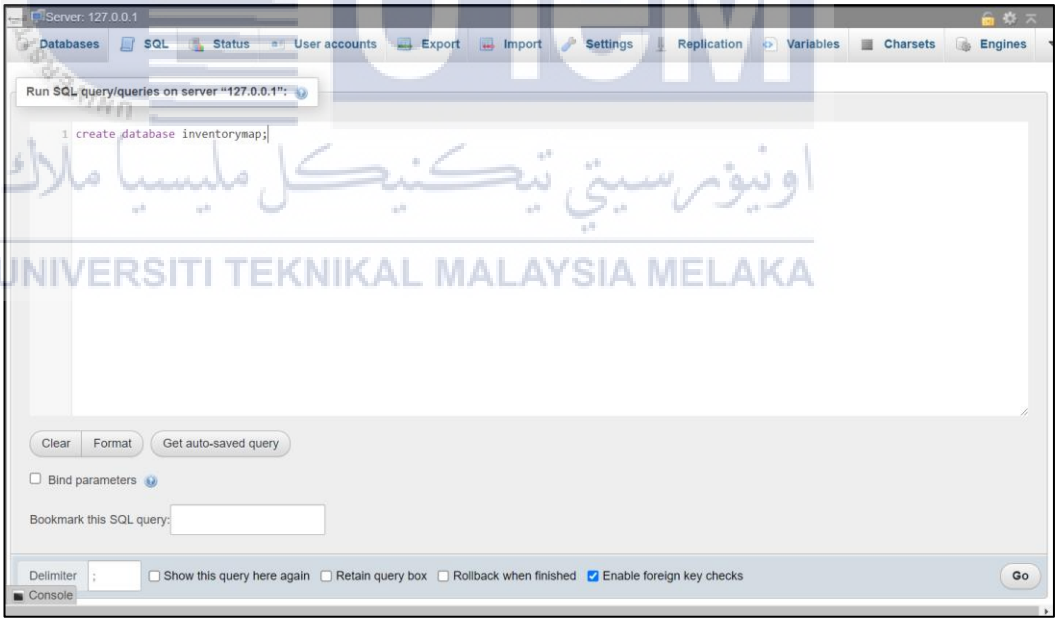

Figure 5.12: Creating Database for IMH as 'inventorymap'.

Step 3: Using SQL language, type the command to create a new database such as 'psm1' for IMH.

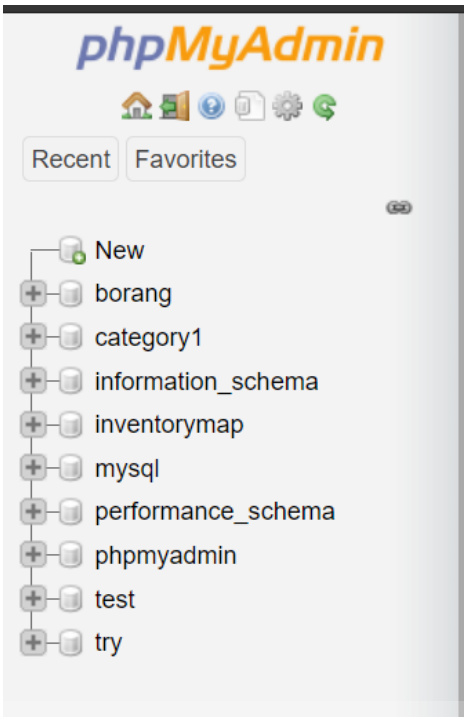

Figure 5.13: List Databases that have been Created

Step 4: After the database is successfully created, the message successful will pop up. Users can view the databases that are created and can start using the database such as create a new table.

5.3 Database Implementation

The database implementation phase explained the Data Definition Language (DDL), trigger, and stored procedures to shows the functionality in this system.

5.3.1 Data Definition Language (DDL)

In the implementation phase of the database, Data Definition Language (DDL) is primarily used by database administrators to create the database and table. Below are shown the DDL that is used in Pahang Pharmacy Gate Pass Management System.

5.3.1.1 Create Database

The first thing users need to create the database by using the 'Create Database' command. After that user can create any new table in the database. The database name

for this system is ‘inventorymap’. Figure 5.12 shows the command for create a database.

5.3.1.2 Create Table

After the database is successfully created, the user needs to create the table in the database by using the ‘Create Table’ command and need to add the attribute name, data type, field length, constraint, the primary key, and lastly the foreign key. Table 5.1 to figure 5.6 shows all the tables that is created into the inventorymap that required for develop IMH.

Table 5.1: DDL for Table Admin

1. TABLE ADMIN

```
CREATE TABLE `admin` (
 `adminId` int(11) NOT NULL,
 `AdminName` varchar(100) NOT NULL,
 `Username` varchar(100) NOT NULL,
 `MobileNumber` bigint(11) NOT NULL,
 `Email` varchar(100) NOT NULL,
 `Password` varchar(100) NOT NULL,
 `AdminRegDate` timestamp NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```


Table 5.2: DDL for TABLE Archieve

2. TABLE ARCHIVE

```
CREATE TABLE `archive` (
 `archiveId` int(11) NOT NULL,
 `productId` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table 5.3: DDL for Table Product

```
3. TABLE PRODUCT

        CREATE TABLE `product` (
         `productId` int(11) NOT NULL,
         `ProductName` varchar(255) NOT NULL,
         `Barcode` int(11) NOT NULL,
         `Floor` int(100) NOT NULL,
         `Department` varchar(100) NOT NULL,
         `RackNumber` varchar(100) NOT NULL,
         `RowRack` varchar(100) NOT NULL,
         `Brand` varchar(100) NOT NULL,
         `Price` varchar(255) NOT NULL,
         `EnterDate` timestamp NOT NULL DEFAULT
        current_timestamp() ON UPDATE current_timestamp(),
         `remark` varchar(100) DEFAULT NULL,
         `outtime` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```


Table 5.4: DDL for Table ProductRemark

```
4. TABLE PRODUCTREMARK

        CREATE TABLE `productremark` (
         `id` int(11) NOT NULL,
         `productId` int(11) NOT NULL,
         `status` varchar(255) NOT NULL,
         `remark` mediumtext NOT NULL,
         `remarkDate` timestamp NOT NULL DEFAULT
        current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table 5.5: DDL for Table Shopper

```
5. TABLE SHOPPER
CREATE TABLE `shopper` (
 `shopperId` int(11) NOT NULL,
 `shopperName` varchar(100) NOT NULL,
 `username` varchar(100) NOT NULL,
 `mobileNumber` bigint(11) NOT NULL,
 `email` varchar(100) NOT NULL,
 `password` varchar(100) NOT NULL,
 `shopperRegDate` timestamp NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp(),
 `updationdate` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table 5.6: DDL for Table Staff

```
6. TABLE STAFF
CREATE TABLE `staff` (
 `staffId` int(11) NOT NULL,
 `adminId` int(11) DEFAULT NULL,
 `staffName` varchar(100) NOT NULL,
 `username` varchar(100) NOT NULL,
 `mobileNumber` bigint(11) NOT NULL,
 `email` varchar(100) NOT NULL,
 `password` varchar(100) NOT NULL,
 `staffRegDate` timestamp NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp(),
 `updationdate` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

5.3.1.3 Create Trigger

The trigger is a condition that automatically checks the data manipulation event had happened on the selected table. The Trigger is performing before or after operations occurred. For this system, trigger insert 'S00' before the primary key in the Staff table. Figure 5.1 shows triggers list implemented in IMH.

Table 5.7: List of Triggers in IMH

Types of triggers	Implemented in table	Use of trigger
Before insert	Staff	To insert 'S00' everytime before the staff id will be insert.

Table 5.8: Create Trigger Insert Staff

```
1. BEFORE INSERT STAFF

$sql1= "CREATE OR REPLACE TRIGGER staffId
BEFORE INSERT ON staff
FOR EACH ROW
SET NEW.staffId = 'S00';";
```

5.3.1.4 Create Store Procedure

The stored procedures will accept or not accept input parameters and can be used by the users using different parameters. Table 5.2 shows list of the procedures that are created in this system.

Table 5.9: List of Procedures in IMH

Type of procedures	Implemented in table	Use of procedures
Stored procedure (DML) inserts, update and delete.	productRemark	View the status of product remark

Table 5.10: Create Stored Procedure View ProductRemark

```
1. PROCEDURE REMARKPRODUCT

BEGIN
SELECT *
FROM productremark
WHERE status = seqStatus;
END
call productremark();
```

5.4 Conclusion

In conclusion, the implementation phase is the final phase before goes to the testing phase. Inventory Map for Hypermarket is developed using the XAMPP Server and MySQL using phpMyAdmin to control the database. The database for this project has 6 tables, triggers, and stored procedures that data storage implemented database.

Chapter 6 will explain about the testing phase of this project.

CHAPTER 6: TESTING

6.1 Introduction

Software testing may be a prepare of assess the usefulness of a program application with an expectation to find whether the created software met the required necessities or not and to recognize the error or bugs to guarantee that the software is defect-free in arrange to create the quality software (SM, n.d.).

This chapter explained about the testing phase of Inventory Map for Hypermarket System. This phase is for make sure the system developed have all the requirements. The testing that has been done on Inventory Map for Hypermarket System to check and confirm all the functionality meets the objective of proposed system that requested by user.

In this venture, the testing methodologies to be embraced are black-box and white-box testing. Black-box testing is performed utilizing computer program interfacing to guarantee that they work as anticipated whereas white-box testing looks interior the program (coding) and employments that information as portion of the testing prepare. White-box testing requires inside information of the framework and programming abilities.

6.2 Test Plan

Test plan reflects the complete extensive testing schedule and approach that the developer utilizes within the testing stage. A test plan consists of test organization, test environment and test schedule. Test plan also help the testing steps and check the effectiveness of Inventory Map for Hypermarket System.

Test organization is to explain the involvement of person within the testing procedure. Test environment is to explain the environment of testing to be carried out, and to characterize hardware, configuration, arrangements and preparing the testing. Test schedule is to characterize how numerous cycles and duration of the test to be conducted.

6.2.1 Test Organization

Test organization is a group of people who is responsible to handle the test procedure of this project. Test procedures will be great if people in different background involve because few points of view may be delivered due to their own knowledge in Information Technology.

Software Developer is the person that responsible in developing Inventory Map for Hypermarket System. Then the client is targeted person who will be the end user of this system such as staff and visitor. Table 6.1 shows the list of testers that involve for testing this system and their responsibilities.

Table 6.1: List of Tester and Their Responsibilities

Tester ID	Roles	Responsibilities
Tester_1	Software Developer	- Responsible to developing, executing, reviewing the integration and component of current system. - Elaborate ideas of improvement the systems. - Writing the system code. - Maintaining the system. - Preparing the user's manual.
Tester_2	Client	- Act as the end user to get some feedbacks, which is staff and visitors. - Testing the module of system developed. - Give feedbacks or opinion for improvements of the system.

6.2.2 Test Environment

A testing environment may be a setup of software and hardware for the testing to execute test cases of Inventory Map for Hypermarket System. In other words, it bolsters test execution with hardware, software, and network configuration (99, n.d.). The test environment setup must imitate the generation environment in arrange to reveal any configuration related issues. It will figure out whether this system can be adaptable to run on different platform of hardware and software.

6.2.2.1 Environment Setup

The environment setup depicts the environment of testing which can be carried out reliably all through the testing, the testing modules include, the software and hardware utilized. Table 6.2 shows the environment setup specification in testing phase.

Table 6.2: Environment Setup Specification in IMH

Environment Configuration	Specification
Operating System	Windows 10
Processor	Intel® Celeron® CPU N2840 @2.16GHz 2.16 GHz
Random Access Memory (RAM)	4GB
Database	MySQL
Server	Apache
Programming Language	Hypertext Preprocessor (PHP) Hypertext Markup Language (HTML)

6.2.2.2 Software Application

Software applications consist of all the application or module inside Inventory Map for Hypermarket System. Table 6.3 shows all the application in IMH.

Table 6.3: IMH Application Environment

System Application	1. System Login 2. Add, update, delete and search in each module for staff and admin. 3. Report Chart 4. Generate PDF for the report graph

6.2.2.3 System Software

System software consists of all the tools that have been used in the Inventory Map for Hypermarket System. Table 6.4 shows all the software that have been used in this system.

Table 6.4: System Software

System Software	1. Windows 10 Education 2. XAMPP Server v3.2.1 (Apache, MySQL) 3. SQL Developer (MySQL) 4. Adobe Dreamweaver 5. Google Chrome (Browser)

6.2.2.4 System Hardware

System hardware consist of all the hardware that have been used in this system. Table 6.5 shows the hardware used for developed Technician Management System.

Table 6.5: System Hardware Tools

System Hardware	1. Laptop 2. Mouse 3. Keyboard 4. Printer

6.2.3 Test Schedule

Test plan is the technique of testing all the data collected within the length of time which has been set. The test schedule is to assign about the schedule for testing of Inventory Map for Hypermarket System such as when and by whom the test will be conduct. The schedule is function as guideline for developer to perform the testing on time as in project timeline. Table 6.6 shows the test schedule for admin and staff.

Table 6.6: Test Schedule of IMH Testing

Module/Component	Activity	Duration (Day(s))
Registration	Error handling test and integration test	1
Login	Error handling test and integration test	1
Change Password	Error handling test and integration test	1
Update Profile	Error handling test and integration test	2
User	Error handling test and integration test	2
Staff	Error handling test and integration test	5
Admin	Error handling test and integration test	6

6.3 Testing Strategies

Test strategy is characterized as a set of directing guideline that illuminates the test plan and controls how testing should be done.

White Box Testing also called as glass box or structural testing. White box testing is checking at the structure of the code inside the system and use that knowledge as process of testing. Should know about the internal structure of system to perform this testing smoothly and can make sure the system run according to specification.

Black Box Testing is more to interacting with the system. It means that to perform this testing techniques it does not require any knowledge about the internal part of the system but need to know how it should perform. Table 6.8 shows explanation about White Box Testing Vs Black Box Testing.

Table 6.7: Differences of White Box Testing and Black Box Testing

Approach	Explanation
White Box Testing	• The test that has been performed to check and evaluate the internal structure of the system through the developer using internal structure (code). • Known as "Structural testing". • Developer and tester (have programming knowledge) will involve in this testing
Black Box Testing	• The test that has been performed to test the system through the functional and non-functional (based on its behaviour) without knowing about internal structure of the system. • Known as "Functional Testing" • Testers involve in this testing

6.3.1 Classes of Test

Two types of tests were selected to use in this phase which are error handling test and integration test.

i. Error Handling Test

Error handling testing is a type of testing that is performed to check whether the system is capable of or able to handle the errors. This testing performed with the help of both developers and the testers and not only focuses on error but also focuses on the exception handling. This test will validate only correct and accurate data from the client (No null values in any form). Error message will pop up on the screen to inform user.

ii. Integration Test

Integration testing is to test the interfaces between the modules. This test is to ensure this system captures data into the database correctly based on input from user. If the process is successful, means that the system is well integrated with the database.

6.4 Test Design

There are two parts of test design which are test description and test data. Test design will clarify about the test that has been performed for each module. Test description is the test case identification, test cases and result for each testing module. As for the test data, it incorporates the genuine life information that will be chosen. Test design will explain about the test has been performed for each module.

6.4.1 Test Description

Test description is a test case that documented set of the data input and operating condition needs to run a test item. Table 6.8 to Table 6.14 shows the test cases and expected result for each module.

Table 6.8:Registration Module

Test Case ID	Description	Testing Type	Expected Output
IMH_01-1	All field is blank	Error Handling Test	“Please fill out this field” will pop up

IMH_01-2	Valid input for each field	Unit Testing/Integration	"Successfully saved" will pop up

Table 6.9: Login Module

Test Case ID	Description	Testing Type	Expected Output
IMH_02-1	All field is blank	Error Handling Test	"Please fill out this field" will pop up
IMH_02-2	Valid input for each field	Unit Testing/Integration	User can login to the system successfully

Table 6.10: Reset Password Module

Test Case ID	Description	Testing Type	Expected Output
IMH_03-1	Confirm new password field blank	Error Handling Test	"Please fill out this field" will pop up
IMH_03-2	Valid mobile number, email, current new password and confirm new password with match requested format	Unit Testing/Integration	"Password successfully change" will pop up

Table 6.11: Update Profile Module

Test Case ID	Description	Testing Type	Expected Output
IMH_04-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
IMH_04-2	Valid input for each field	Unit Testing/Integration	"Successfully saved" will pop up

Table 6.12: Manage Staff Module (Add New Staff)

Test Case ID	Description	Testing Type	Expected Output
IMH_05-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
IMH_05-2	Valid input for each field	Unit Testing/Integration	"Successfully saved" will pop up

Table 6.13: Manage New Product

Test Case ID	Description	Testing Type	Expected Output
IMH_06-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
IMH_06-2	Valid input for each field	Unit Testing/Integration	"Successfully saved" will pop up

Table 6.14: Manage Product (Edit Product)

Test Case ID	Description	Testing Type	Expected Output
IMH_07-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
IMH_07-2	Valid input for each field	Unit Testing/Integration	"Successfully saved" will pop up

Table 6.15: Search Product Between Date

Test Case ID	Description	Testing Type	Expected Output
IMH_08-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
IMH_08-2	Valid input for each field	Unit Testing/Integration	"Successfully saved" will pop up

6.4.2 Test Data

Test data are used to get expected output based on test design of this system. This test data is performed by validating, testing and verifying the software behavior using the real data from user. Table 6.16 to Table 6.23 shows the test data.

Table 6.16: Test Data for Registration Module

Test Data ID	TD_01-1	TD_01-2
Test Case ID	IMH_01-1	IMH_01-2

User registration form	User does not fill up the fields correctly Your Name: Ahmad bin Ali Username: ahmad Email Address: ahmad Phone Number: 1 Password: abc	Users fill up the fields completely Your Name: Ahmad bin Ali Username: ahmad Email Address: ahmad@gmail.com Phone Number: 01234567890 Password: Abc_123
Test Data Result	New user registration fails due to incomplete form	New user is successfully saved and submitted

Table 6.17: Test Data of Login Module

Test Data ID	**TD_02-1**	**TD_02-2**
Test Case ID	**IMH_02-1**	**IMH_02-2**
Username	ahmad	ahmad
Password	abc	Abc_123
Test Data Result	Login fail because of incorrect password	Successfully login to IMH

Table 6.18: Test Data of Reset Password Module

Test Data ID	**TD_03-1**	**TD_03-2**
Test Case ID	**IMH_03-1**	**IMH_03-2**

Current Password	Abc_123	Abc_123
New Password	123_Abc	123_Abc
Confirm New Password	111_Abc	123_Abc
Test Data Result	New user registration fails due to incorrect password	Changed Password Successful

Table 6.19: Test Data for Update Profile Module

Test Data ID	**TD_04-1**	**TD_04-2**
Test Case ID	**IMH_04-1**	**IMH_04-2**
Update Profile Form	User does not fill up the fields correctly Admin Name: Abu bin Ali Email: abu@gmail.com Phone Number:	Users fill up the fields completely Admin Name: Abu bin Ali Email: abu@gmail.com Phone Number: 011223344
Test Data Result	Update profile fails due to incomplete form	Profile changed is successfully saved and submitted

Table 6.20: Test Data for Manage Staff (Insert Staff) Module

Test Data ID	**TD_05-1**	**TD_05-2**
Test Case ID	**IMH_05-1**	**IMH_05-2**

Insert New Staff Form	User does not fill up the fields correctly Staff Name: Gopal a/l Kumar Username: gopal Email: gopal@gmail.com Phone Number:	Users fill up the fields completely Staff Name: Gopal a/l Kumar Username: gopal Email: gopal@gmail.com Phone Number: 01987654321
Test Data Result	Insert new staff fails due to incomplete form	Insert new staff is successfully saved and submitted

Table 6.21: Test Data for New Product Module

Test Data ID	**TD_06-1**	**TD_06-2**
Test Case ID	**IMH_06-1**	**IMH_06-2**
Insert New Product Form	Detail of product does not fill up correctly Product Name: Ikan Selar Barcode: 02233817891918272 Floor: 1 Department: Barangan Basah Rack: BS2 Row: BS12A Brand: Nelayan Canggih Price:	Detail of product fill up the fields completely Product Name: Ikan Selar Barcode: 02233817891918272 Floor: 1 Department: Barangan Basah Rack: BS2 Row: BS12A Brand: Nelayan Canggih Price: 9

Test Data Result	Insert new product fails due to incomplete form	Insert new product is successfully saved and submitted

Table 6.22: Test Data of Manage Product (Edit Product) Module

Test Data ID	**TD_07-1**	**TD_07-2**
Test Case ID	**IMH_07-1**	**IMH_07-2**
Update Product Form	Detail of product does not fill up correctly Product Name: Ikan Selar Barcode: Floor: 1 Department: Barangan Basah Rack: BS2 Row: BS12A Brand: Nelayan Canggih Price: 9	Detail of product fill up the fields completely Product Name: Ikan Selar Barcode: 02233817891918272 Floor: 1 Department: Barangan Basah Rack: BS2 Row: BS12A Brand: Nelayan Canggih Price: 9
Test Data Result	Update product fails due to incomplete form	Update product is successfully saved and submitted

Table 6.23: Test Data of Search Product Between Date Module

Test Data ID	**TD_08-1**	**TD_08-2**
Test Case ID	**IMH_08-1**	**IMH_08-2**

Search Product Between Date	Detail of product does not fill up correctly From Date: 09/1/2021 To Date:	Detail of product fill up the fields completely From Date: 09/1/2021 To Date:
Test Data Result	Search product between date fails due to incomplete form	Search product between date is successfully saved and submitted

6.5 Test Result and Analysis

For the test result, it clarifies the result of testing that followed using test data from user. This part will explain each modules test case results which consists of the success or failure. All test case will be tested, and the result will be shown on the table below. Table 6.23 until Table 6.31 shows the test result and analysis.

Table 6.24: Registration Module Test Result

Module/Component			**Result**		
TEST CASE ID	**TEST DATA ID**	**Testing Date**	**Description**	**Success**	**Fail**
IMH_01-1	TD_01-1	31/08/2021	Some field blank	/	
IMH_01-2	TD_01-2	31/08/2021	All fields were filled	/	

Table 6.25: Login Module Test Result

Module/Component	**Result**

TEST CASE ID	TEST DATA ID	Testing Date	Description	Success	Fail
IMH_02-1	TD_02-1	01/09/2021	User key in incorrect username and password.	/	
IMH_02-2	TD_02-2	01/09/2021	User key in incorrect username and password.	/	

Table 6.26: Reset Password Test Result

Module/Component			Result		
TEST CASE ID	**TEST DATA ID**	**Testing Date**	**Description**	**Success**	**Fail**
IMH_03-1	TD_03-1	01/09/2021	User does not fill confirm password field.	/	
IMH_03-2	TD_03-2	01/09/2021	All fields were filled.	/	

Table 6.27: Update Profile Module Test Result

Module/Component	Result

TEST CASE ID	TEST DATA ID	Testing Date	Description	Success	Fail
IMH_04-1	TD_04-1	01/09/2021	Some field is blank	/	
IMH_04-2	TD_04-2	01/09/2021	All fields were filled.	/	

Table 6.28: Manage Staff (Add New Staff) Module Test Result

Module/Component			Result		
TEST CASE ID	**TEST DATA ID**	**Testing Date**	**Description**	**Success**	**Fail**
IMH_05-1	TD_05-1	01/09/2021	Some field blank	/	
IMH_05-2	TD_05-2	01/09/2021	All fields were filled	/	

Table 6.29: Add New Product Module Test Result

Module/Component			Result		
TEST CASE ID	**TEST DATA ID**	**Testing Date**	**Description**	**Success**	**Fail**
IMH_06-1	TD_06-1	01/09/2021	Some field blank	/	

IMH_06-2	TD_06-2	01/09/2021	All fields were filled	/	

Table 6.30: Manage Product (Edit Product)

Module/Component			**Result**		
TEST CASE ID	**TEST DATA ID**	**Testing Date**	**Description**	**Success**	**Fail**
IMH_07-1	TD_07-1	01/09/2021	Some field blank	/	
IMH_07-2	TD_07-2	01/09/2021	All fields were filled	/	

Table 6.31: Search Between Date Module Test Data

Module/Component			**Result**		
TEST CASE ID	**TEST DATA ID**	**Testing Date**	**Description**	**Success**	**Fail**
IMH_08-1	TD_08-1	01/09/2021	Some field blank	/	
IMH_08-2	TD_08-2	01/09/2021	All fields were filled	/	

6.6 Conclusion

To conclude, testing stage clarified around the procedure that has been utilized to approve and confirm IMH to create beyond any doubt all work meets the necessity and module that have been created. Software testing is the hard part in every software development. All testing action must be arranged well and conducted through due period and the fetched of settling the bug and error. Test arrange comprise the test cases that are utilized to look at changes anticipate of the system.

Chapter 7 will explain about the conclusion of the project. A conclusion will be explained by pointing out the strengths and the weakness.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

To conclude this project, this chapter will explain the weakness and strengths of this system. In this chapter also explain about the proposition for improvement on IMH which are any suggestion that can make this system better for the user. All the strength and weakness of this system is based on a module that developed in this system.

Lastly, this chapter also explains about project contribution such as to university, faculty, company, or individual that relate to this project.

7.2 Observation on Weaknesses and Strengths

Every system that developed always have their strength and weakness. Below are a list of the strength and weakness of Pahang Pharmacy Gate Pass Management System.

i. Strength

- Ease for user to search any inventory product in a hypermarket
- Easily for the staff to key in the updated inventory map for any product
- Easily for the staff to add new product into the system

ii. Weakness

- Does not have backup and recovery procedure in the database.

7.3 Prepositions for Improvement

After analyzing the strength and weaknesses that state above, there are several ideas for improvements that can help in future use. Inventory Map for Hypermarket System will be better with implement a backup and recovery procedure in the database to ensure the data safely store if have something happen to the system or database such as misbehave, corruption, and database problem. Backup and recovery procedures can be implemented by developer for this system in many ways such as daily, weekly, or monthly backup either using full or incremental backup.

It would be better for this system if it can generate real time mapping for the user such as Waze application. This will help user from hustle and straightly get the location of the product that they want. Lack of time being one of my biggest reasons why I did not get to achieve to generate real time mapping.

7.4 Project Contribution

Project contribution for this project is divide into the university, company, and individual. For the first is the university, this project is belonging to the university in University Technical Malaysia Melaka. This also can be the source of ideas, finding, and information to other university students in developing management system, especially for gate pass management systems.

Furthermore, for hypermarket contribution, the hypermarket can manage its system and process to handle their inventory product easily and more efficiently. The data of product or any information will store in the database and can ease management to track and manage the product.

Finally, for individual contribution is during project documentation to get any knowledge or information and ideas to develop and implement another system or make any improvement of their system.

7.5 Conclusion

The conclusion of this project that can be concluded is this system has been developed to ease the manage product inventory using the Inventory Map for Hypermarket System. The system has been developed also meets objective and can solve the problem statement that has been studied in the first chapter in this project and still need improvement for the better in future use. The proposition of improvement that has been analyzed need to be implemented in this project to make sure this system more efficient, reliable, and more secure.

APPENDICES

USER MANUAL FOR ADMIN

1. Add New Staff
    - Click on Manage Staff on left navigation bar
    - Click button Add New Staff
    - Insert the full name, username, phone number and the temporary password for the staff.
    - Click Add
    - New staff created

2. Delete Existing Staff
    - Click on Manage Staff on left navigation bar
    - Choose the existing that want to delete and click the red dustbin emoticon
    - Confirmation needed. Simply click ‘yes’ if it is the correct staff to be delete
    - Staff deleted

3. Add New Product
    - Click on New Product on left navigation bar
    - Fill in the form (Product Name, Barcode, Floor, Department, Rack, Row, Brand, Price) and click the Add button
    - New product added successfully

4. Read All Product
    - Click on All Product on left navigation bar
    - All product and its detail will be appeared

5. Add Remark on Product
    - Click on All Product on left navigation bar
    - List of products will be appeared
    - Scroll the detail of the product to the right
    - Choose the product that want to put remark and scroll to the right.
    - Click on the pencil emoticon

- Insert remark
- Click Update
- The remark successfully updated

6. Delete Product
   - Click on Manage Product on left navigation bar
   - Choose the existing product that want to delete and click the red dustbin emoticon
   - Confirmation needed. Simply click ‘yes’ if it is the correct product to be delete
   - Product deleted

7. Update Detail Product
   - Click on Manage Product on left navigation bar
   - Choose the existing product that want to update and click the blue pencil emoticon
   - Only Department, Rack Number, Row Rack and Price can be edited.
   - Click on the Update button
   - Product successfully updated

8. Search Product Between Dates
   - Click on Product B/w Dates on left navigation bar
   - Choose the From Date and To Date to get the list of products between the date
   - Click Submit button
   - List of products appeared

9. Update Profile
   - Click Admin Profile in admin photo
   - Only Admin Name, Email and phone number can be edited
   - Click update
   - Profile updated

10. Search Product or Brand
    - Click on search bar on the top navigation
    - Input the product name or product brand

- Click enter or the blue search emoticon
- The result will be appeared

USER MANUAL FOR STAFF

1. Add New Product
   - Click on New Product on left navigation bar
   - Fill in the form (Product Name, Barcode, Floor, Department, Rack, Row, Brand, Price) and click the Add button
   - New product added successfully

2. Read All Product
   - Click on All Product on left navigation bar
   - All product and its detail will be appeared

3. Add Remark on Product
   - Click on All Product on left navigation bar
   - List of products will be appeared
   - Scroll the detail of the product to the right
   - Choose the product that want to put remark and scroll to the right.
   - Click on the pencil emoticon
   - Insert remark
   - Click Update
   - The remark successfully updated

4. Delete Product
   - Click on Manage Product on left navigation bar
   - Choose the existing product that want to delete and click the red dustbin emoticon
   - Confirmation needed. Simply click ‘yes’ if it is the correct product to be delete
   - Product deleted

5. Update Detail Product
    - Click on Manage Product on left navigation bar
    - Choose the existing product that want to update and click the blue pencil emoticon
    - Only Department, Rack Number, Row Rack and Price can be edited.
    - Click on the Update button
    - Product successfully updated

6. Search Product Between Dates
    - Click on Product B/w Dates on left navigation bar
    - Choose the From Date and To Date to get the list of products between the date
    - Click Submit button
    - List of products appeared

7. Update Profile
    - Click Staff Profile in admin photo
    - Only Staff Name, Email and phone number can be edited
    - Click update
    - Profile updated

8. Search Product or Brand
    - Click on search bar on the top navigation
    - Input the product name or product brand
    - Click enter or the blue search emoticon
    - The result will be appeared

USER MANUAL FOR USER

1. Read All Product
    - Click on All Product on left navigation bar
    - All product and its detail will be appeared

2. Search Product Between Dates

- Click on Product B/w Dates on left navigation bar
- Choose the From Date and To Date to get the list of products between the date
- Click Submit button
- List of products appeared

3. Update Profile

- Click User Profile in admin photo
- Only User's Full Name, Email and phone number can be edited
- Click update
- Profile updated

8. Search Product or Brand

- Click on search bar on the top navigation
- Input the product name or product brand
- Click enter or the blue search emoticon
- The result will be appeared

REFERENCES

Hasliza Hassan, Muhammad Sabbir Rahman. (2012). Transformation of Hypermarket Retailing Industry in Malaysia. Conference: International Conference on Innovation, Management and Technology Research (ICIMTR 2012).