I-KOOP MANAGEMENT SYSTEM

AININ SOFEA BINTI AZAHARI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

I-KOOP MANAGEMENT SYSTEM

AININ SOFEA BINTI AZAHARI

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

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2020/2021

DECLARATION

I hereby declare that this project report entitled

I-KOOP MANAGEMENT SYSTEM

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ainin azahari Date : 07/09/2021

(AININ SOFEA BINTI AZAHARI)

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 : _____ Date : 07/09/2021

(TS. ABDUL RAZAK BIN HUSSAIN)

DEDICATION

To my beloved parents, family and all my 3 BITD friend whose support, encouragement and inspired me to complete this project. Next, big thank you go to my supervisor, Ts. Abdul Razak Bin Hussain for this project for helping me doing my PSM 1. I also would like to thank for his kind great advice and brilliant ideas. Not forgetting to all the lecturer that involve directly or indirectly, the Universiti Teknikal Malaysia Melaka especially Faculty Information Technology and Communication for giving me chance to study here.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Ts. Abdul Razak Bin Hussain for giving me the opportunity to work on this project. I would also like to express my gratitude to my assistant for successfully completing this project. I am so grateful that I managed and able to complete the project within the time given. Thank you so much to everyone that give me amazing advice, encouragement and criticism. I am extremely grateful and thankful for supporting me to complete this project. This project cannot be completed without guidance from my supervisor and encouragement from my friends.

ABSTRACT

The i-KOOP Management System is designed to help facilitate the school cooperative management system, process of buying and selling goods in schools and how to generate the report in the system. The irregular and neglected staff attendance record where staff have to take turns to manage or looking after the school cooperative without any fixed schedule. This problem is caused by difficulty in allocating time or schedule for staff who will manage the school cooperative. To solve the problem, system admin will be able to maintain comprehensive human resource records that includes details of staff and supplier. Next, rely on visual inspection of what product to replenish that may cause over stocking. The staff and supplier need to record the product or items using paper. This problem also may cause missing data and duplicate. To solve the problem is the staff need to facilitate product inventory that includes product sales, product re-ordering and product disposal. It is crucial to have an online cooperative. Finally, the staff also able to obtain the reports easily to save time.

ABSTRAK

Sistem pengurusan i-KOOP dirancang untuk membantu memudahkan sistem pengurusan koperasi sekolah, proses jual beli barang di sekolah dan bagaimana menghasilkan laporan dalam sistem tersebut. Pertama, rekod kehadiran kakitangan yang tidak teratur dan diabaikan di mana kakitangan perlu bergilir-gilir mengurus atau menjaga koperasi sekolah tanpa jadual yang tetap. Masalah ini disebabkan oleh kesukaran dalam memperuntukkan masa atau jadual untuk staf yang akan menguruskan koperasi sekolah. Untuk menyelesaikan masalah tersebut, pentadbir sistem akan menyimpan rekod sumber manusia yang menyeluruh yang merangkumi perincian kakitangan dan pembekal. Seterusnya, system secara manual bergantung pada pemeriksaan visual produk yang akan diisi semula yang mungkin menyebabkan stok berlebihan. Kakitangan dan pembekal perlu merekodkan produk atau barang menggunakan kertas. Masalah ini juga boleh menyebabkan kehilangan data dan pendua. Untuk menyelesaikan masalah tersebut, kakitangan perlu memfasilitasi inventori produk yang merangkumi penjualan produk, pesanan semula produk dan pelupusan produk. Akhir sekali, kakitangan dapat memperoleh laporan dengan mudah untuk menjimatkan masa.

TABLE OF CONTENTS

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
DFD	**-**	**Data Flow Diagram**
ERD	**-**	**Entity Relationship Diagram**
XAMPP	**-**	**Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P).**
PHP	**-**	**Hypertext Preprocessor**
HTTP	**-**	**Hypertext Transfer Protocol**
HTML	**-**	**Hypertext Markup Language**

Chapter 1: INTRODUCTION

1.1 Introduction

The i-KOOP Management System is a system that developed to help the school shop to manage the stock more efficiently. This system is pleasant automated system for stock management. The objectives are to help industry to manage inventory competently, improve performance and efficiency in handling inventory. It will help users collect and store key-in data for their products and services.

The i-KOOP Management System is invent to help ease the process of buying and selling goods in schools, school cooperative management system and how to calculate the total stock of goods accumulated. Besides, with the school cooperative system, it can help facilitate the affairs of employees as well as can save time and reduce errors.

The project is to improve the school cooperative management system to be more efficient. It is crucial to have an online cooperative management system to help improvised the current manual system because the data can be missing or duplicate. Using the manual management make the data is not stored properly and it can be lost.

1.2 Problem Statement

i. Irregular and neglected staff attendance record.

- The staff have to take turns to manage or looking after the school cooperative without any fixed schedule. This problem is caused by difficulty in allocating time or schedule for staff who will manage the school cooperative.

ii. Rely on visual inspection of what product to replenish that may cause over stocking.

- The staff and supplier need to record the product or items using paper. This problem also may cause data redundancy and it will be able to be duplicate.

iii. Staff difficult to manage or view the sales in higher quantity.

- The staff are difficult to summarize the highest sale in month because the report is recorded by manually using the book or paper. The manual system also does not able to display the sales in higher quantity in chart.

1.3 Objective

i. To maintain comprehensive human resource records that includes details of staff and supplier.

ii. To facilitate product inventory that includes product sales, product re-ordering and product disposal.

iii. To generate a report of the sales.

1.4 Scope

The scopes are functionality, user system and the database. The system can be accessed by teacher, system admin and supplier where the teacher can make order for the cooperatives products or items. The scope can be divided into target user and modules.

1.4.1 User Scope

(a) Teacher

The teacher is a staff that will manage the system related to the list of products where teacher can insert, view, update and delete the product. Teacher also can make order and manage sales at the end of the day. After that, teacher can view the time table and view the sales inventory report.

(b) System admin

System admin is responsible to manage the schedule or time table for the teacher to decide the staff that will manage the cooperative based on predetermined schedule. System admin also will be able to register new staff and view the sales inventory report.

(c) Supplier

The supplier is people that supplies the product for the cooperative at the school. The supplier can view, insert, update and delete the product. Next, supplier also will be able to view the history of order time table.

1.4.2 Module Scope

(a) Product modules

Product modules will be able to view all the product by name or category. It will display the details of the product name, description, category and price.

(b) Order modules

Order modules will be able to track the total or quantity of the items or product in the system. The order modules will display the product name, quantity and price. Order can be made by choosing the list of supplier.

(c) Schedule modules

Schedule module will be able to display the schedule of staff that will manage the cooperative at school based on predetermined schedule. Furthermore, the schedule also created so that suppliers have a regular schedule to come check the inventory products easily.

(d) Report modules

Report modules will be able to display the sales of the product or items in line graph and pie chart. The line graph will display the report selling by weekly while the pie chart will display the list quantity of the product based on category.

1.5 Project significance

The system helps the staff to control and monitor the goods in stock. It also allows them to keep track of the inventory and make sure that the goods are well-organized. The i-KOOP Management System is developed to manage the stock efficiently. This can solve the problem of staff to order the product or items and calculate the stock of the product easily and save a lot of time. Also, it will help to determine financial position because it is very important to have the correct valuation.

Furthermore, the system of i-KOOP Management System also assists the schedule or time table to make sure the staff will follow the schedule that has been given so that they are more organized in managing the cooperative at school.

1.6 Expected Output

The project aims is to provide an inventory tracking for the product or item in school shop. Furthermore, the staff can place order for the stock or item and manage the stock using the system. The system will help the user to increase the performance of the work. After that, the system will be able to view or display the report of sales by graph.

The i-KOOP Management System is developed to provide the system to reduce cost and improve the service level from using the manual system. The system is to focuses on the inventory management to help the user to manage the stocks or product efficiently. The database system that are using to develop the i-KOOP Management System is SQL.

1.7 Conclusion

Online services play an important role in today's daily life. The i-KOOP Management System is facing technological changes, and the current system is still using the old traditional methods to manage its services. A central platform for employees and suppliers to manage inventory and even make payments. This system will very effectively avoid problems such as information errors or loss of registry. Using this system, data is easy to use and securely stored in the system and database.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 About Figures

A project methodology is basically a process that integrates scientific approaches to systems, supported by systems thinking and simulation methods, implemented by people with appropriate systems and experience in the application. It is a standard process that is followed to perform all the steps required to analyze, design, implement and maintain information systems.

As a software methodology, it is the framework for planning, organizing and managing system development processes, an appropriate system development methodology is needed. With the right methodology, it can lead to more effective and profitable software development.

2.2 Project Methodology

To develop i-KOOP Management System, Database Life Cycle (DBLC) has been choose as the project methology. It includes six phases in the Database Life Cycle (DBLC) which is database initial study, database design, implementing and loading, testing and evaluation, operating and maitenance and evaluation.

a) Database Initial Study

Analysis the scope, user groups involved and required data. It is necessary to collect facts whose authority include interview, investigation and other related archives.

i. Identify the objective, project aim, scope of the project and the boundaries project of i-KOOP Management System.
ii. Analyze the project system and identify the problems.

b) Database Design

Based on database initial study, carry out the database design process or model that will support the operations. There are two database design approach which is top-down uses an entity-relationship (ER) model and bottom-up design uses the process of normalization.

i. Create and identify logical database design, physical database design and conceptual data modelling.
ii. Define entities, attributes, relationship, process, identification and requirements.
iii. Identify the Database Management System (DBMS), the language and framework for the project.

c) Implementing and Loading

Implement all design specifications from previous phase. The output of the database design phase is a series of guidance detailing the establishment of tables, attributes, views, indexes, security constraints, storage and achievement guidelines.

i. Install the required software for the improvement process.
ii. Database system design.

d) Testing and Evaluation

The database used to prototype application and occurs with the programming. The integrity to enforced via proper use of primary, foreign key rules.

i. Run or test the database to ensure the integrity and security of data.
ii. Evaluate the database.

e) Operating

It is considered to be usable once the database has passed the assessment stage. The process of system evolution inevitably begins at the beginning of the operating period.

i. If there is any problem, it has to resolved according to severity.

f) Maintenance and Evaluation

Inside the database, the database administrator must be able to conduct routine preservation tasks.

i. Preventive maintenance or backup, recovery, the attributes, security and the integrity of i-KOOP Management System.

2.3 Project Schedule & Milestone

Table 2.1 Gant Chart

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Planning the Project															
Problem Identification and Analysis															
Database Design															
Development and Implementation of Project															
Testing and Evaluation															
Presentation and Demonstration															

Table 2.2 Table for Overall Milestone

Milestone	Expected Outcome	Dates
1. Planning the Project	• Discuss project proposal • Do a correction and updated the proposal.	08 March 2021
2. Problem Identification and Analysis	• Identify the problem of the project. • Identify the objective for the project.	19 March 2021
3. Database Design	• Context Diagram. • Data Flow Diagram (DFD). • Entity Relationship Diagram (ERD). • Data Dictionary.	23 March 2021
4. Development and Implementation of Project	• User Interface. • Source code.	15 Jun 2021
5. Testing and Evaluation	• Project testing.	20 June 2021
6. Presentation and Demonstration	• PSM1 Log Book. • PSM1 Final Report. • Project Demonstration. • Final Presentation.	30 June 2021 – 2 July 2021

2.4 Conclusion

In this chapter, we can define the project methodology and planning according to the database life cycle method used. It contains six stages. In addition, we can also define project schedule and milestone. The goal and scope of the concept are included in the final standard. Estimated time for the implementation of the planning system. Completed assure that the entire project is finished on time.

CHAPTER 3: ANALYSIS

3.1 Introduction

The analysis report should describe the analysis of the current system. System analysis isa rough description of the system being developed. The analysis aims to address aspects related to the existing system, which will continue to envolve. Existing systems over-time, system deficiencies and quality improvements.

The analysis of the system is split into two stages which is the analysis of the present system and also the analysis of the new system to be developed. This analysis are often represented in context and information flow diagrams. The analysis section is that the method shaping needs or system from the attitude of users, organizations or established standards.

3.2 Problem Analysis

The current system does not provide a schedule for staff where they can take turn to manage the school cooperative and because of that staff are difficult to manage due to schedule that has not been fixed. Moreover, the system does not have automatic calculation to calculate the inventory where it is fully done by writing on paper. Furthermore, the system may be duplicate and will cause data loss because it does not have backup. Other than that, the manual system does not able to display the sales in higher quantity.

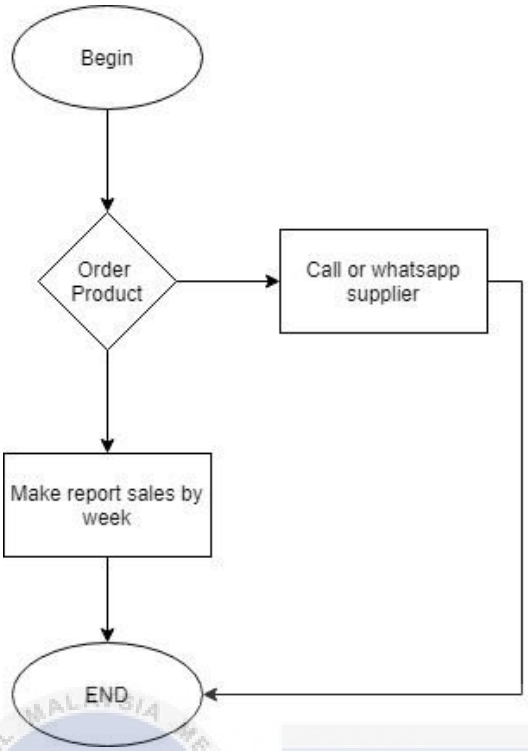

Figure 3.1: Flowchart order product for staff.

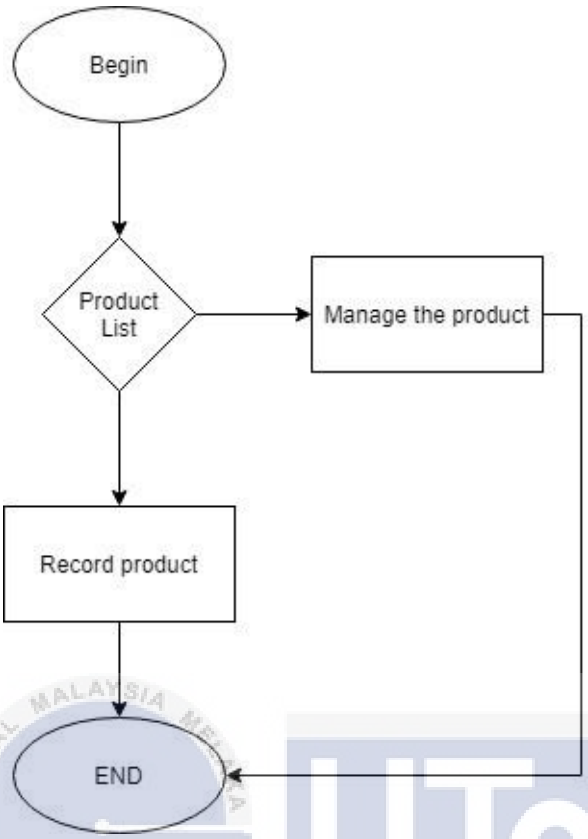

Figure 3.2: Flowchart product list for staff.

3.3 The proposed improvements/solutions

The improvement that has been proposed is to provide the schedule or time table for the staff to help to manage the school cooperative without any problem. After that, the solution for the data that might be duplicate and will cause data loss, the system has facilitate the product inventory that include product sales, product re-ordering and product disposal. Lastly, the improvements or solution is to generate a report of sales in higher quantity.

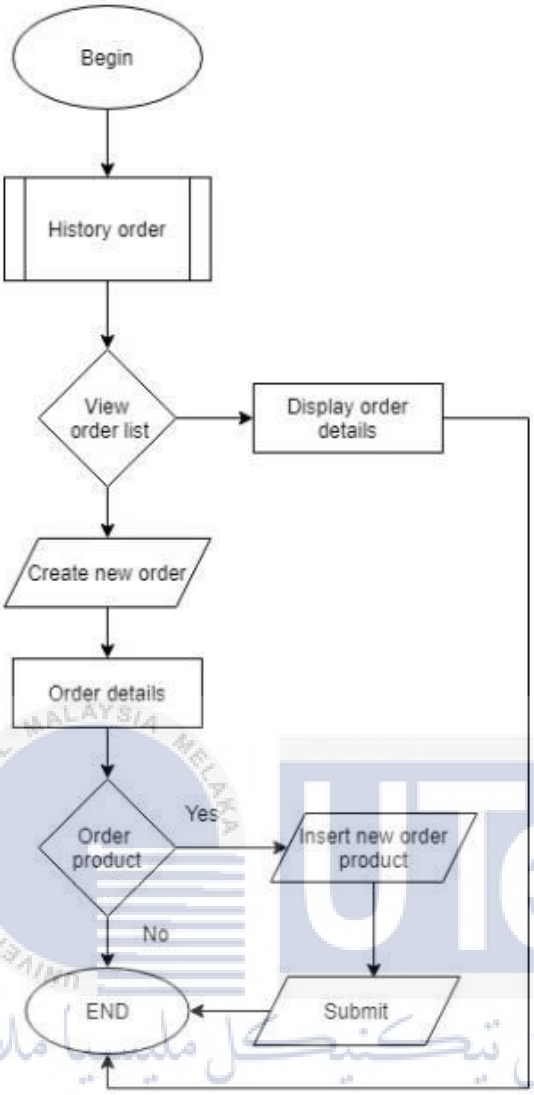

Figure 3.3: Flowchart to-be-system order product.

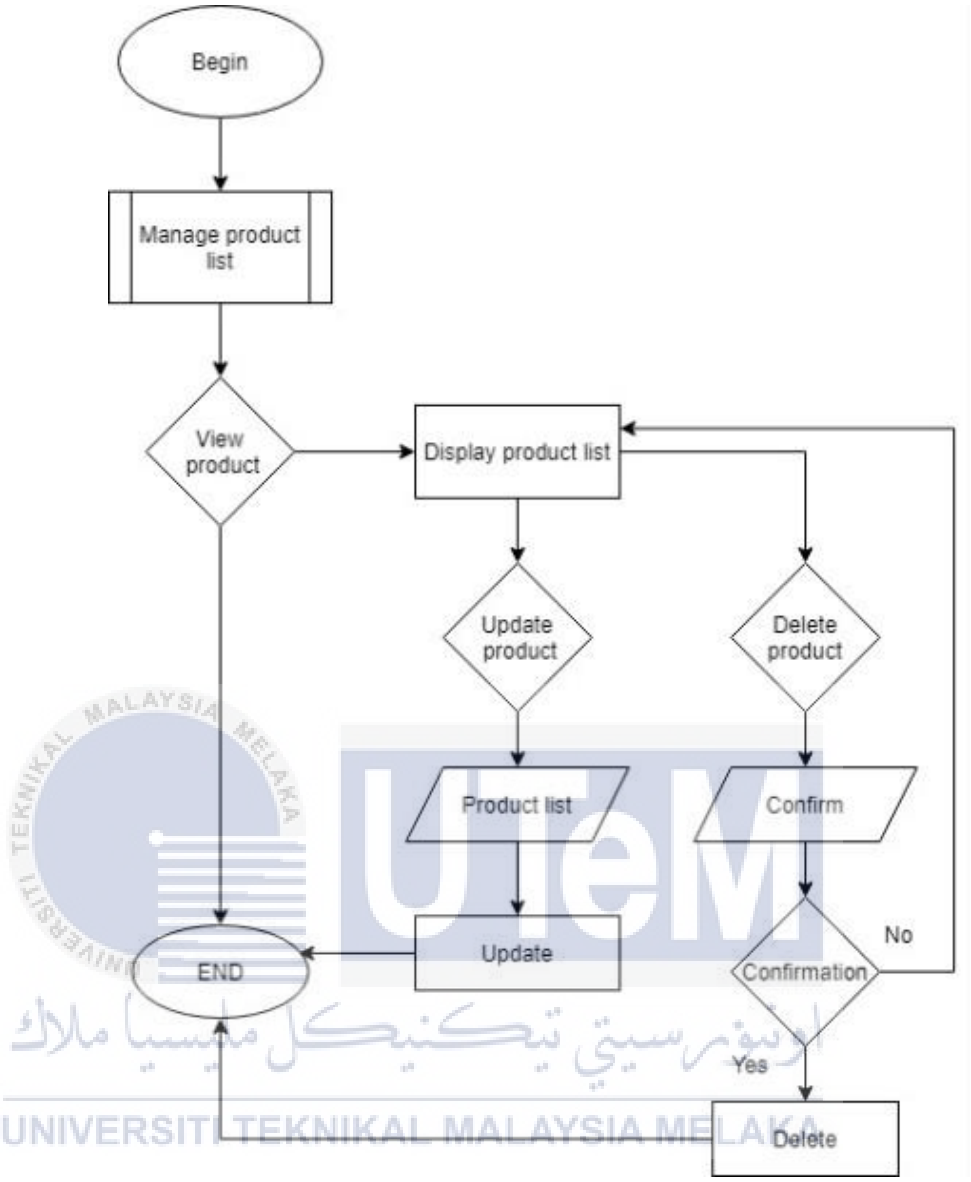

Figure 3.4: Flowchart to-be-system product list.

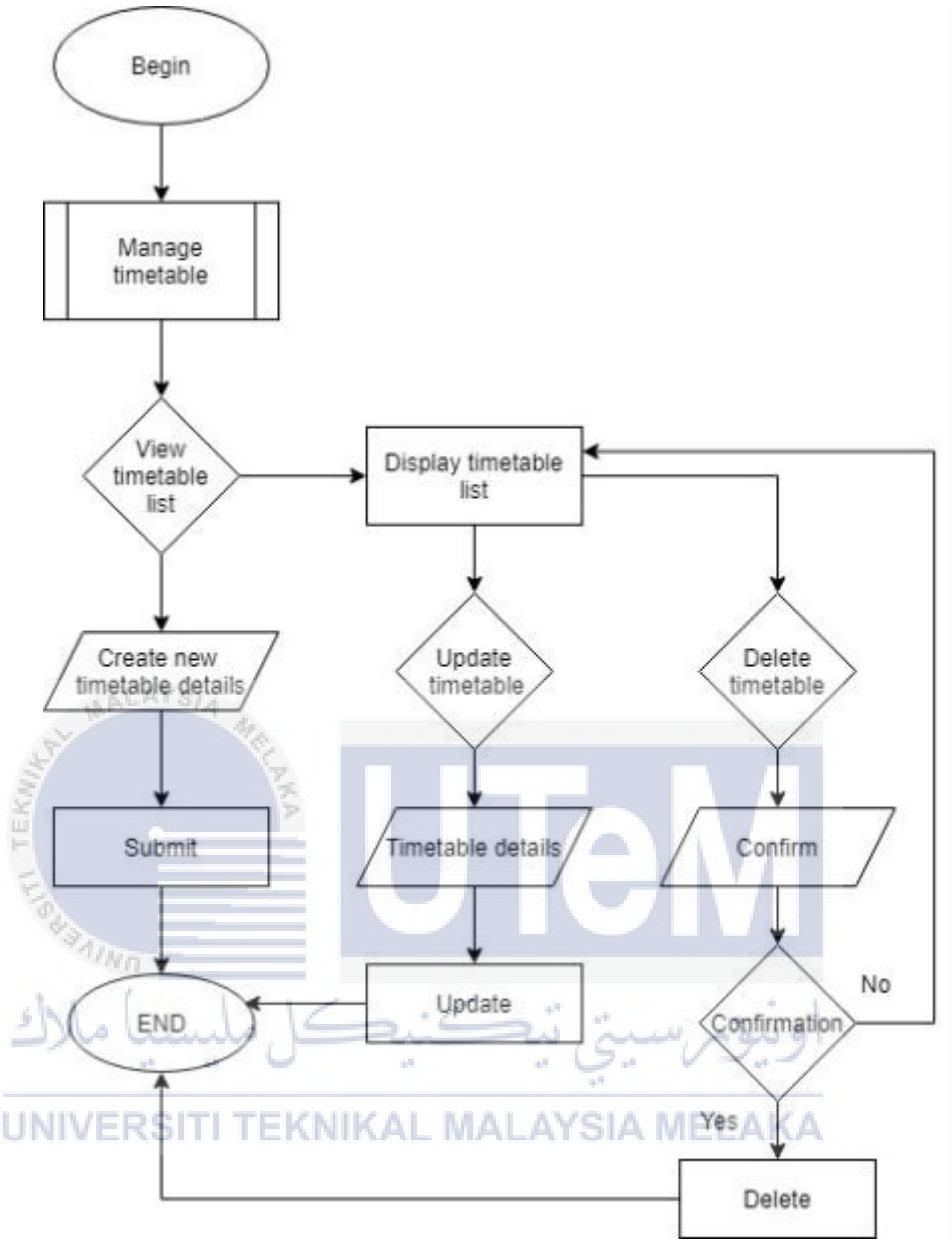

Figure 3.5: Flowchart to-be-system timetable list.

3.4 Requirement analysis of the to-be system

3.4.1 Functional Requirements (Process Model)

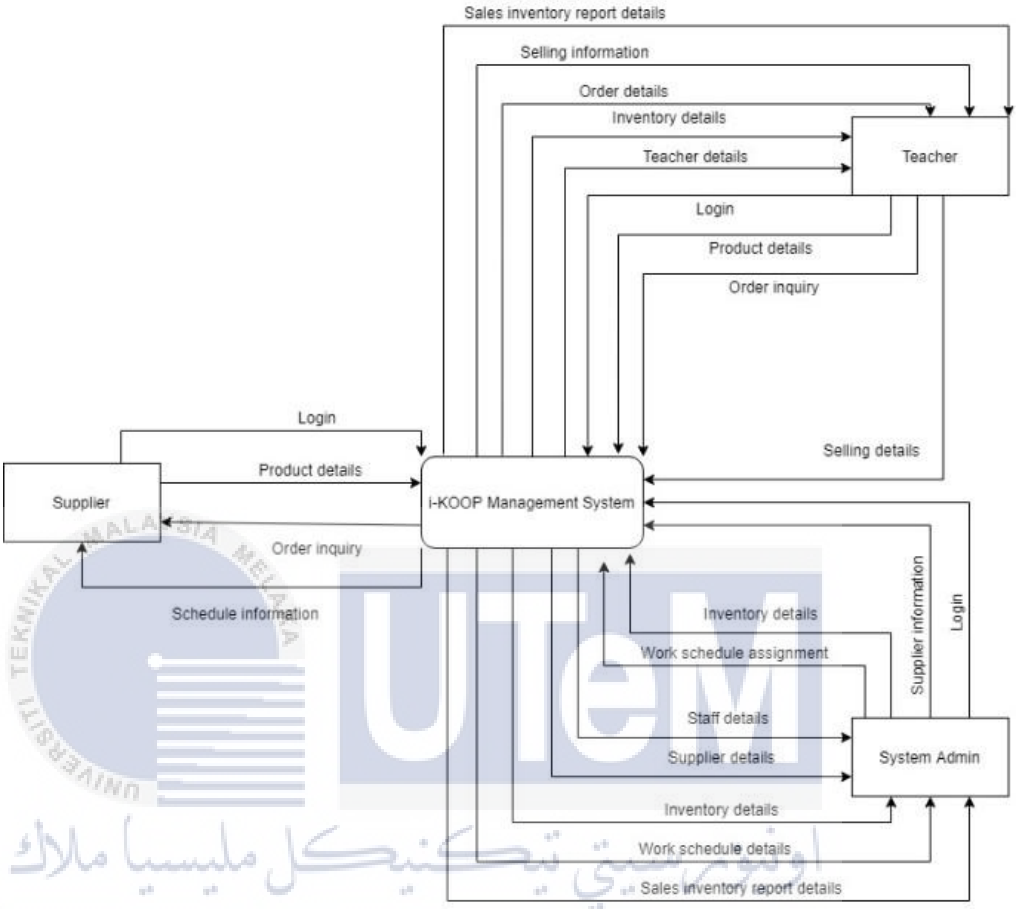

Figure 3.6: Context Diagram.

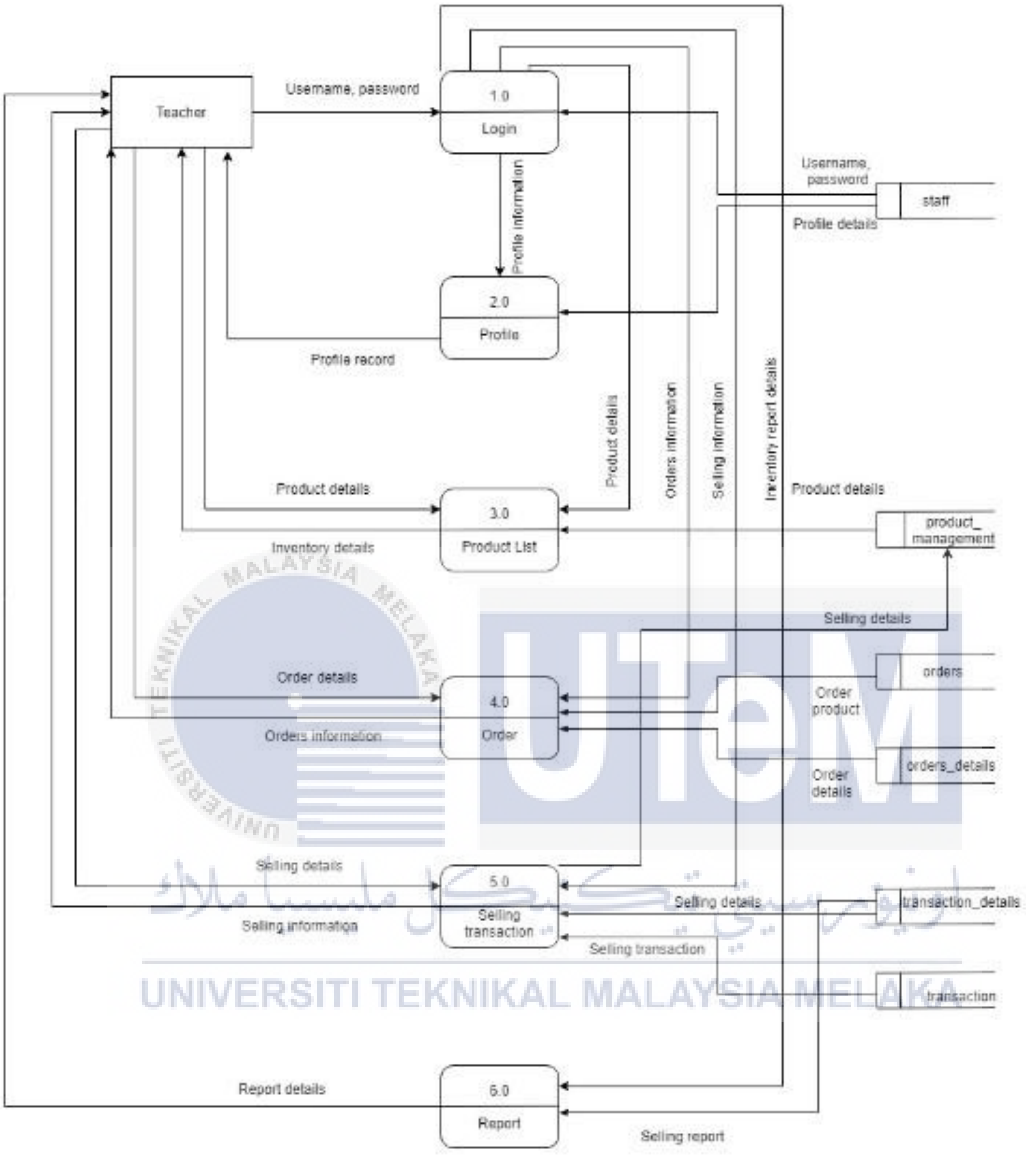

Figure 3.7: Data Flow Diagram Level 0

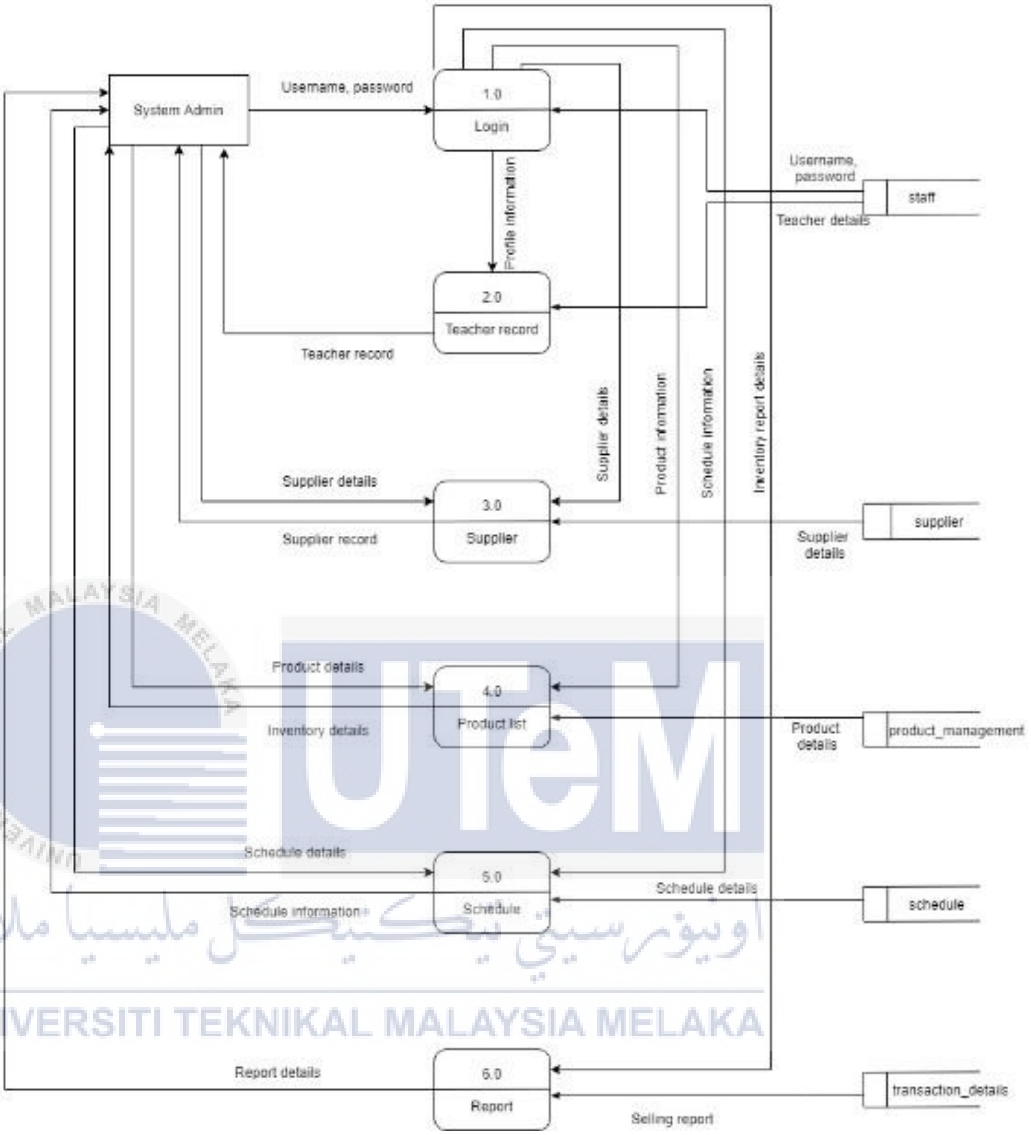

Figure 3.8: Data Flow Diagram Level 0

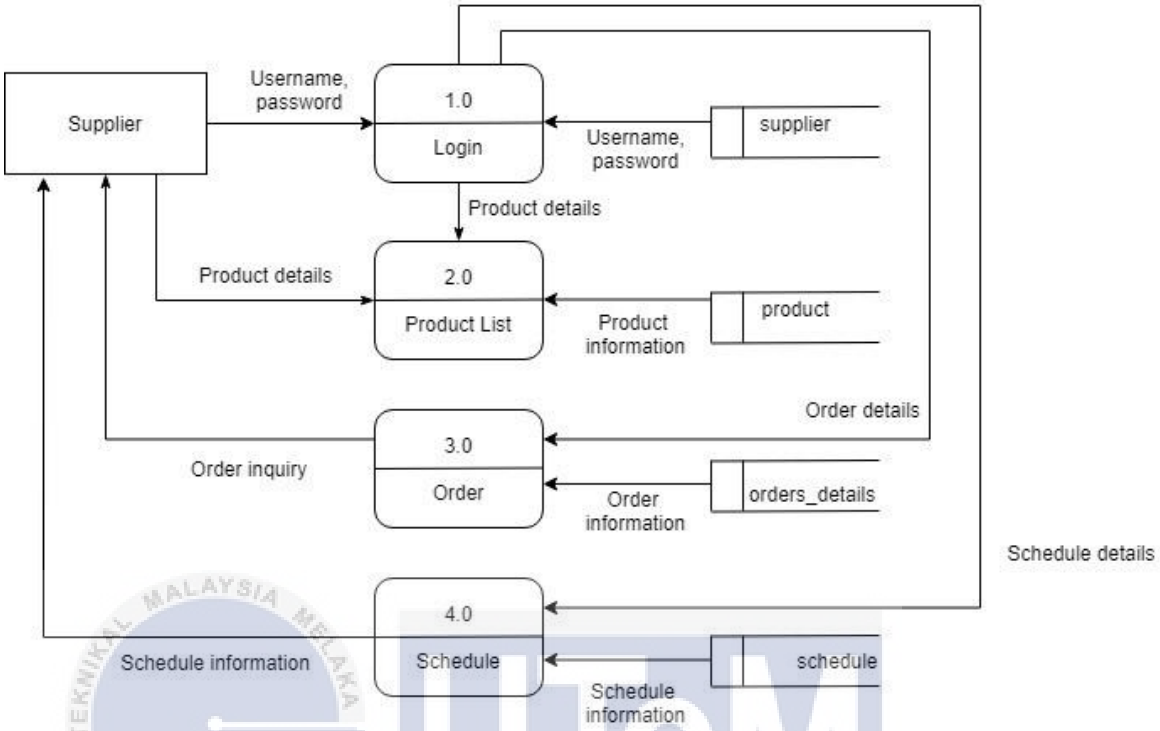

Figure 3.9: Data Flow Diagram Level 0

3.4.2 Non-Functional Requirement

- The system should authorized users with the correct username and password.
- The calculation of the quantity product should be correct.
- The system has a user-friendly interfce and is easy to learn to work with the system.

3.4.3 Others Requirement

i. Software

Table 3.1: Software Requirement.

No.	Software	Description
1.	Sublime Text	Sublime text is a cross-platform shared software editor. It is intuitive and easy to learn and supports multi-line and block editing.
2.	XAMPP	XAMPP is a free and ASCII text file cross-platform internet server resolution stack package developed by Apache Friends, consisting chiefly of the Apache hypertext transfer protocol Server, MariaDB database, and interpreters for scripts written within the PHP and Perl programming languages.
3.	PHP	PHP is stand for Hypertext Preprocessor. PHP is a widely-used, open source scripting language. PHP is a web scripting language embedded in HTML. It allows to collect, process and utilize data to create output
4.	Windows 10	Windows 10 is to unify the Windows experience across multiple devices, such desktop computers, tablets, and smartphones.

ii. Hardware

Table 3.2: Hardware Requirement.

Name	Description
Brand	HP
Processor	AMD A6-5200 APU with Radeon(TM) HD Graphics 2.00 GHz
Memory (RAM)	8.00 GB
System Type	64-bit operating system, x64-based processor

3.5 Conclusion

The context diagram shows that the system should be designed to be composed of all the inputs and outputs that the system needs. The flow sheet shows the various details of the system operation.

CHAPTER 4: DESIGN

4.1 Introduction

Design is the planning of appearance and function of a product or service. The design process follows a linear process of investigating the problem, solving the problem, and then implementing the appropriate solution. The system is designed to meet demand. The demand determined in the essential analysis stage are translated into system design, which accurately describe the system design, and can be used as absorption for the next stage of system design.

4.2 Database Design

Database design is defined as a series of steps that help design, create, implement, and manage the data management system. The data model contains the need of logical and physical design. It defines the database structure for planning, storing, and managing information. Data accuracy can only be achieved if the database is designed to store only relevant and required data.

A database design is essential to ensure the consistency of information, eliminate redundant data, execute queries efficiently, and improve database performance. Database design also allows easy access and retrieval of data when needed.

4.2.1 Conceptual Design

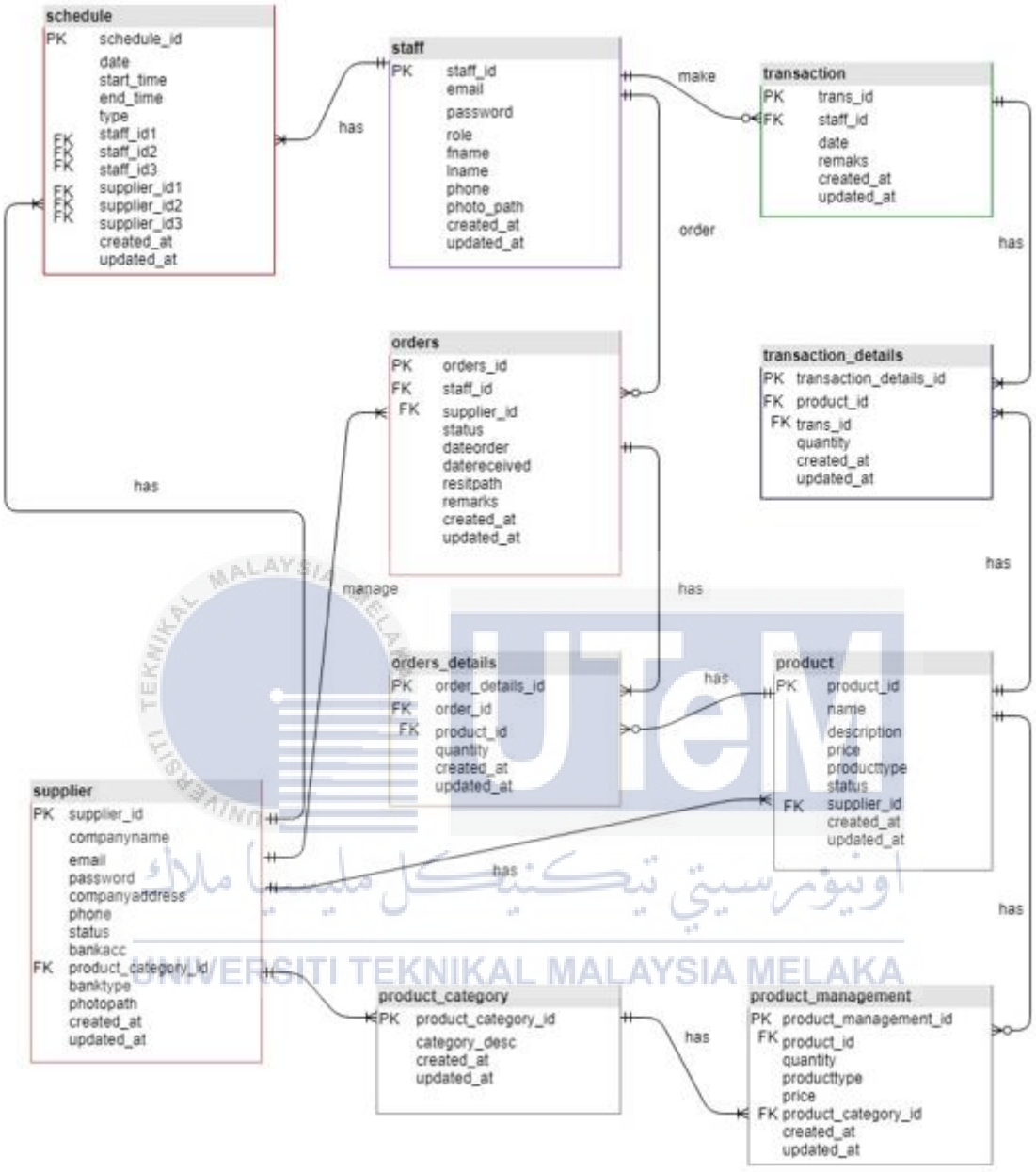

Figure 4.2: ERD of i-KOOP Management System

4.2.1.1 Business Rule

1. One or many schedule can only have one staff. One staff can has one or many schedule.
2. One staff can make zero or many transaction. Zero or many transaction can be make by only one staff.
3. One staff can make zero or many order. Zero or many order can be make by only one staff.
4. One order can has one or many order details. One or many order details can have only one order.
5. Zero or many order details can only has one product. One product can has zero or many order details.
6. One product has zero or many product management. Zero or many product management can has one product.
7. One product category can has one or many product management. One or many product management can only has one product category.
8. One transaction has one or many transaction details. One or many transaction details can has only one transaction.
9. One or many transaction details can has only one product. One product can has one or many transaction details.
10. One supplier can handle one or many order. One or many order can be handle by only one supplier

4.2.2 Logical Design

Table 4.1: Data Dictionary for Staff Table.

STAFF						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	staff_id	Bigint(20)	Yes	-	PK	-
2.	email	Varchar(50)	Yes	-	-	-
3.	password	Varchar(255)	Yes	-	-	-
4.	role	Smallint(6)	Yes	-	-	-
5.	fname	Varchar(100)	Yes	-	-	-
6.	lname	Varchar(100)	Yes	-	-	-
7.	phone	Varchar(20)	Yes	-	-	-
8.	photo_path	Varchar(200)	Yes	-	-	-
9.	created_at	Timestamp	Yes	-	-	-
10.	updated_at	Timestamp	Yes	-	-	-

Table 4.2: Data Dictionary for Supplier Table.

SUPPLIER						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	supplier_id	Bigint(20)	Yes	-	PK	-
2.	companyname	Varchar(100)	Yes	-	-	-
3.	email	Varchar(50)	Yes	-	-	-
4.	password	Varchar(255)	Yes	-	-	-
5.	companyaddress	Varchar(255)	Yes	-	-	-
6.	Phone	Varchar(20)	Yes	-	-	-
7.	status	Smallint(6)	Yes	-	-	-
8.	bankacc	Varchar(100)	Yes	-	-	-
9.	category	Bigint(20)	Yes	-	-	-
10.	banktype	Smallint(6)	Yes	-	-	-
11.	photo_path	Varchar(200)	Yes	-	-	-
12.	created_at	Timestamp	Yes	-	-	-
13.	updated_at	Timestamp	Yes	-	-	-

Table 4.3: Data Dictionary for Orders Table.

ORDERS						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	orders_id	Bigint(20)	Yes	-	PK	-
2.	staff_id	Bigint(20)	Yes	-	FK	staff
3.	supplier_id	Bigint(20)	Yes	-	FK	supplier
4.	status	Smallint(6)	Yes	-	-	-
5.	dateorder	Date	Yes	-	-	-
6.	datereceived	date	Yes	-	-	-
7.	resitpath	Varchar(200)	Yes	-	-	-
8.	remarks	Varchar(1000)	Yes	-	-	-
9.	created_at	Timestamp	Yes	-	-	-
10.	updated_at	Timestamp	Yes	-	-	-

Table 4.4: Data Dictionary for Orders_Details Table.

ORDERS_DETAILS						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	order_details_id	Bigint(20)	Yes	-	PK	-
2.	order_id	Bigint(20)	Yes	-	FK	order
3.	product_id	Bigint(20)	Yes	-	FK	product
4.	quantity	Int(11)	Yes	-	-	-
5.	created_at	Timestamp	Yes	-	-	-
6.	updated_at	Timestamp	Yes	-	-	-

Table 4.5: Data Dictionary for Product_Categeory Table.

PRODUCT_CATEGORY						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	Product_category_id	Bigint(20)	Yes	-	PK	-
2.	Category_desc	Varchar(100)	Yes	-	-	-
3.	Created_at	Timestamp	Yes	-	-	-
4.	Updated_at	Timestamp	Yes	-	-	-

Table 4.6: Data Dictionary for Product Table.

PRODUCT						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	product_id	Bigint(20)	Yes	-	PK	-
2.	name	Varchar(200)	Yes	-	-	-
3.	description	Varchar(1000)	Yes	-	-	-
4.	price	Double(8,2)	Yes	-	-	-
5.	producttype	Smallint(6)	Yes	-	-	-
6.	status	Smallint(6)	Yes	-	-	-
7.	supplier_id	Bigint(20)	Yes	-	-	-
8.	created_at	Timestamp	Yes	-	-	-
9.	updated_at	Timestamp	Yes	-	-	-

Table 4.7: Data Dictionary for Product_Management Table.

PRODUCT_MANAGEMENT						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	product_management_id	Bigint(20)	Yes	-	PK	-
2.	product_id	Bigint(20)	Yes	-	FK	product
3.	quantity	Int(11)	Yes	-	-	-
4.	producttype	Smallint(6)	Yes	-	-	-
5.	price	Double(8,2)	Yes	-	-	-
6.	category	Bigint(20)	Yes	-	-	-
7.	created_at	Timestamp	Yes	-	-	-
8.	updated_at	Timestamp	Yes	-	-	-

Table 4.8: Data Dictionary for Transaction Table.

TRANSACTION						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	trans_id	Bigint(20)	Yes	-	PK	-
2.	staff_id	Bigint(20)	Yes	-	FK	staff
3.	date	date	Yes	-	-	-
4.	remarks	Varchar(1000)	Yes	-	-	-
5.	created_at	Timestamp	Yes	-	-	-
6.	updated_at	Timestamp	Yes	-	-	-

Table 4.9: Data Dictionary for Transaction_Details Table.

TRANSACTION_DETAILS						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	transaction_details_id	Bigint(20)	Yes	-	PK	-
2.	trans_id	Bigint(20)	Yes	-	FK	Transaction
3.	product_id	Bigint(20)	Yes	-	FK	Product
4.	quantity	Int(11)	Yes	-	-	-
5.	created_at	Timestamp	Yes	-	-	-
6.	updated_at	Timestamp	Yes	-	-	-

Table 4.10: Data Dictionary for Schedule Table.

SCHEDULE						
No	Attribute Name	Data Type	Required	Default Value	Constraint	FK Table
1.	schedule_id	Bigint(20)	Yes	-	PK	-
2.	date	Date	Yes	-	-	-
3.	start_time	Time	Yes	-	-	-
4.	end_time	Time	Yes	-	-	-
5.	type	Smallint(6)	Yes	-	-	-
6.	staff_id1	Bigint(20)	Yes	-	-	-
7.	staff_id2	Bigint(20)	Yes	-	-	-
8.	staff_id3	Bigint(20)	Yes	-	-	-
9.	created_at	Timestamp	Yes	-	-	-
10.	updated_at	Timestamp	Yes	-	-	-

4.2.3 Query Design

a. Insert

- $create_productmanage = "INSERT INTO product_management(quantity,product_id,created_at,updated_at) VALUES ('$data_listorder[quantity]','$data_listorder[product_id]','$current_timestamp','$current_timestamp')"; }

  $result_productmanage = $conn->query($create_productmanage); }

b. Update

- while($data_productmanage = $get_product_manage->fetch_assoc())
  {
  $total_quantity = $data_productmanage['quantity'] +
  $data_listorder['quantity'];

  $create_productmanage = "UPDATE product_management
  SET quantity='$total_quantity',updated_at='$current_timestamp'
  WHERE product_management_id='$data_productmanage
  [product_management_id]' ";

c. Error Handling

- if (password_verify($_POST['password'],
  $row["password"]))
  {
  if($table_name == 'staff') {
  $_SESSION["role"] = $row['role'];
  $_SESSION["user_id"] = $row["staff_id"]; }
  else {

  $_SESSION["role"] = 3;
  $_SESSION["user_id"] = $row["supplier_id"]; }

  Echo json_encode(['messagecode'=>1,'message_title'=>
  'Success!', 'message'=>'Success Login']);

  $break_foreach = true;
  break; }
  else {
  echo json_encode(['messagecode'=>0,'message_title'=>
  'Error!', 'message'=>'Wrong Password']);

  $break_foreach = true;
  break;
  }
  }
  }
  }

  ($email_invalid) echo
  json_encode(['messagecode'=>0,'message_title'=>'Error!',
  'message'=>'Email Invalid']);
  }

4.2.4 Physical Design

Physical design the effective method of storing and retrieving objects. It represents the entities of the system. It is a graphical illustration of the system, representing external and internal entities of the system with to and data flow. Physical design also relevant to the input and output of the system. Physical design is how the input data is produce, how the input is prepared, and the way the output is displayed.

4.3 Graphical User Interface (GUI) Design

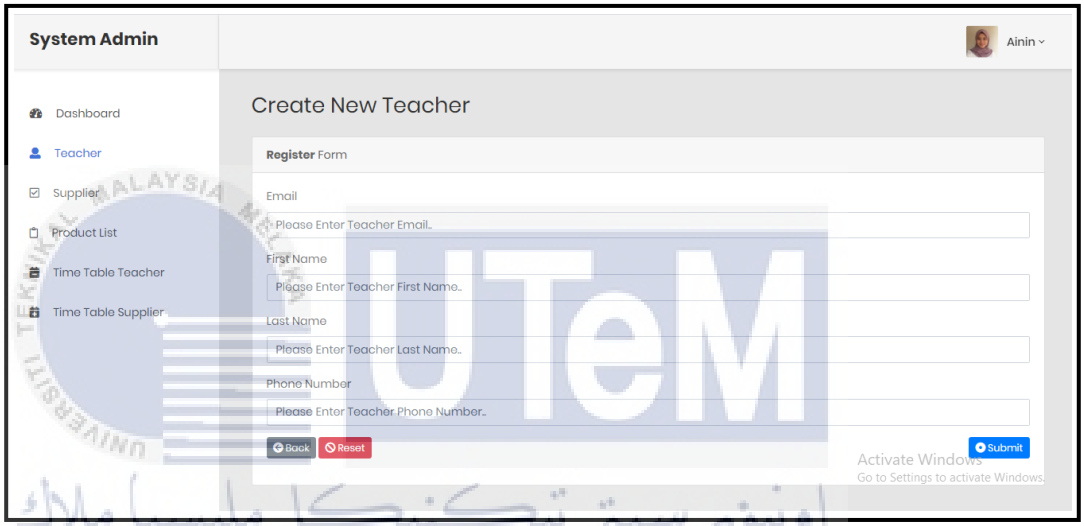

Figure 4.3: Register Form.

The system admin will insert or create the new staff details forthe new user of the system. Therefore, system admin need to fill in the form which is email, first name, last name and phone number.

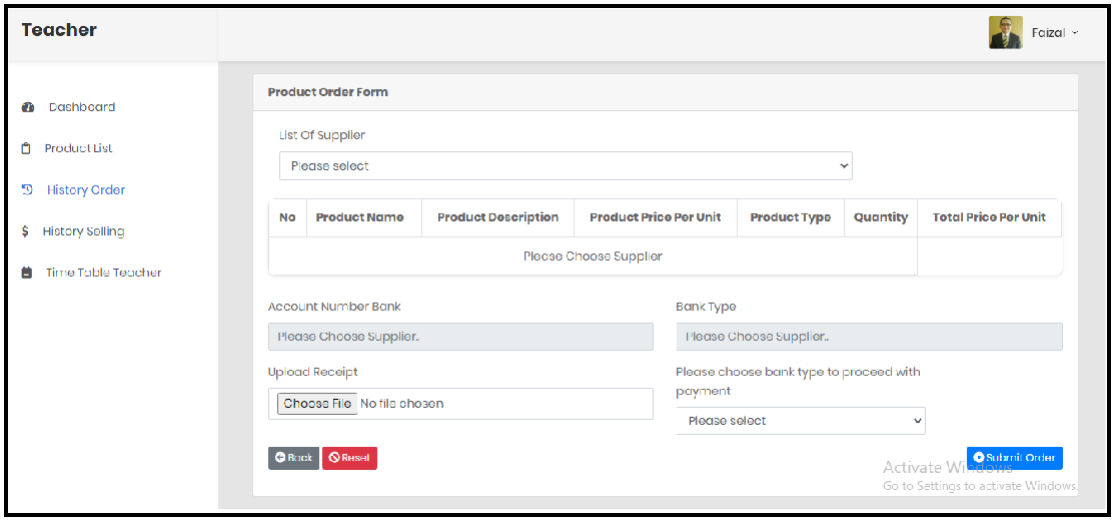

Figure 4.4: Order Form.

The teacher need to insert the details in the form to order the product from the supplier.

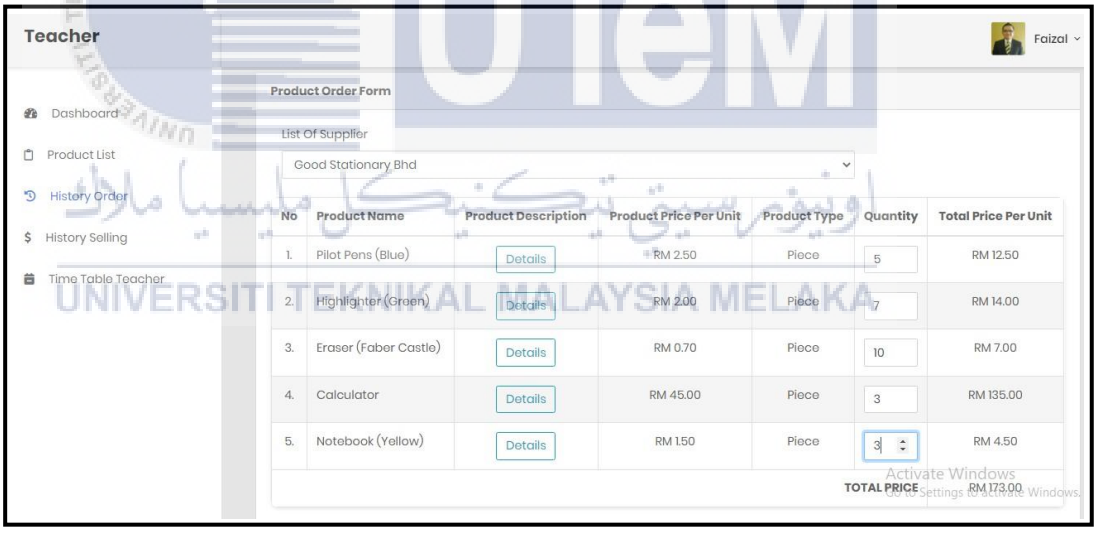

Figure 4.5: Insert order details.

The teacher need to select the supplier company and after that set the quantity of the product that need to order. After that, teacher have to make payment by choosing the bank type. Lastly, teacher need to upload the receipt.

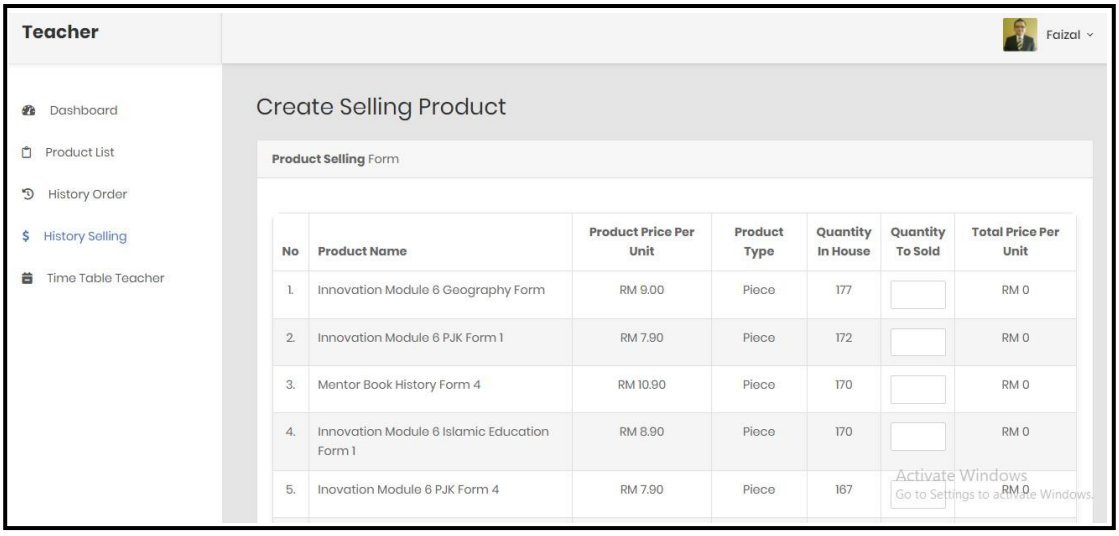

Figure 4.6: Form of selling product.

The teacher need to insert the quantity of selling product in the form at the end of the day.

Figure 4.7: Insert timetable for teacher.

System admin need to insert the time table for teacher to assign which teacher will manage the school shop.

4.4 Conclusion

This chapter is concluded the design phase models which is conceptual, logical and physical design. It is to specified the requirement. Without a design, the system can not be constructed, enforced or operated. The purpose of design phase is to develop detailed specifications that emphasize physical solution. Furthermore, system requirements, logical descriptions of entities, relationships, and data attributes recorded in the analysis phase.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

The implementation chapter introduces the task for the employment of the information design, shows the installation and therefore the configuration procedure of the database, within the implementation part of the database MySQL is put in on the Windows 10 platform, during this phase knowledge Definition Language (DDL) and knowledge Manipulation Language (DML) in the sort of SQL statements are enforced in the database. The operation standing is delineate for every module.

5.2 Software Development Environment Setup

In the i-KOOP Management System, the software developmenet environment contain few components which is Database Server in MySQL, programming languange in PHP and phpMyAdmin which is Database Management Tool for the system.

5.2.1 Installation Step

Step 1: Go to the browser and download the XAMPP for Windows.

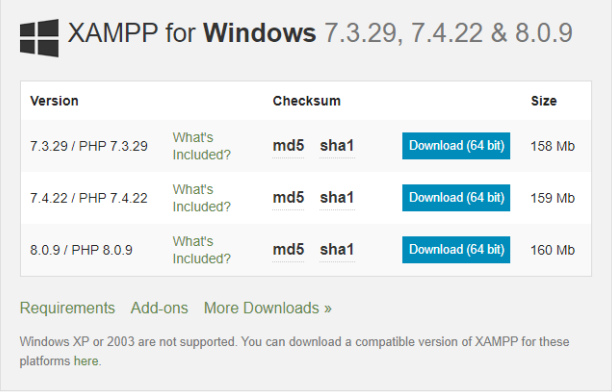

Figure 5.1: Download XAMP.

Step 2: Click the Yes button.

Step 3: Click the Next button.

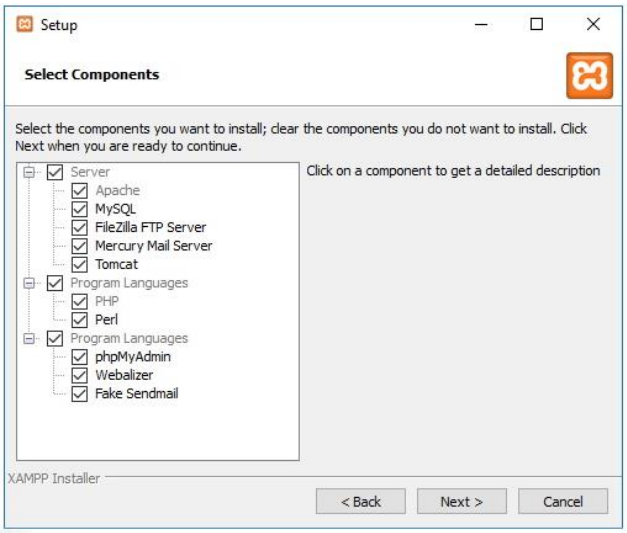

Figure 5.2: Choose Component.

Step 4: Click Next button.

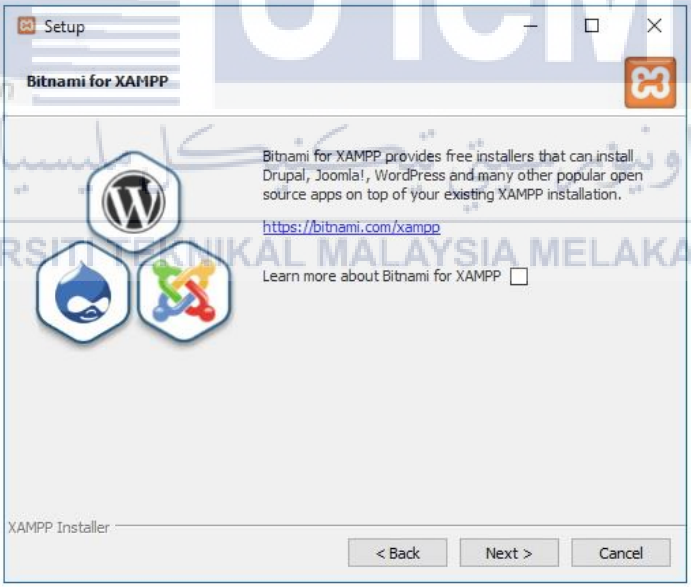

Figure 5.3: Bitnami for XAMPP.

Step 5: Select folder for the location and click Next.

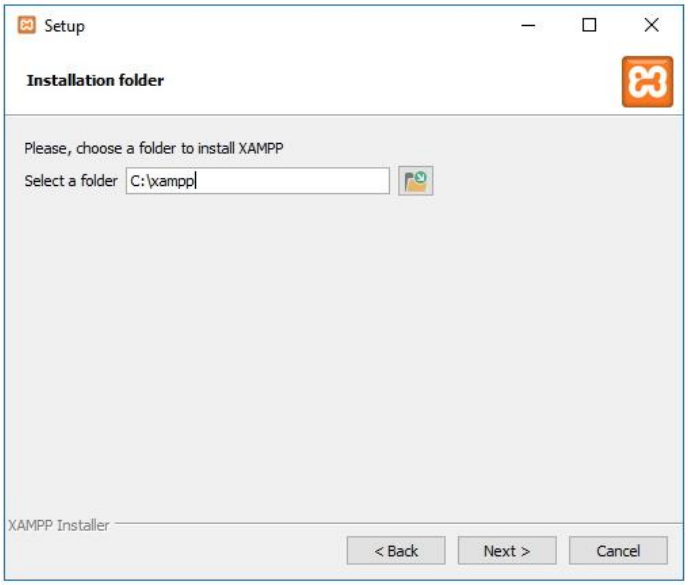

Figure 5.4: Installation of the folder.

Step 6: Click Finish Button.

Figure 5.5: Installation of XAMPP has finished.

Step 7: Click Start button at Apache and MySQL to run.

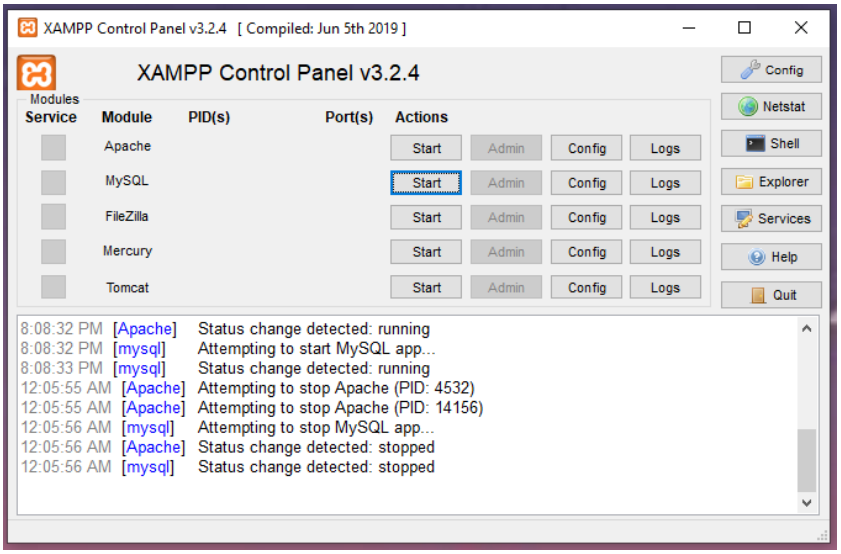

Figure 5.6: Click Start of Apache and MySQL.

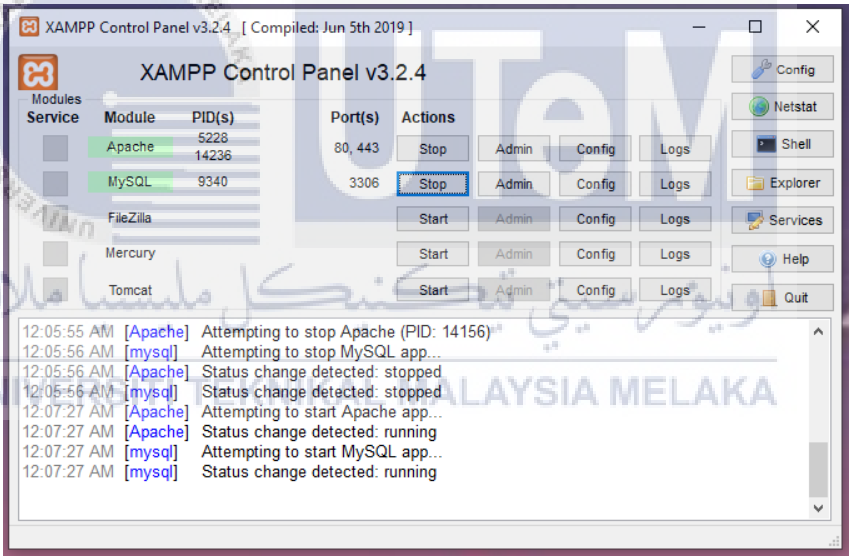

Figure 5.7: Apache and MySQL are running.

5.2.2 Database creation

Step 1: Click New to create new database.

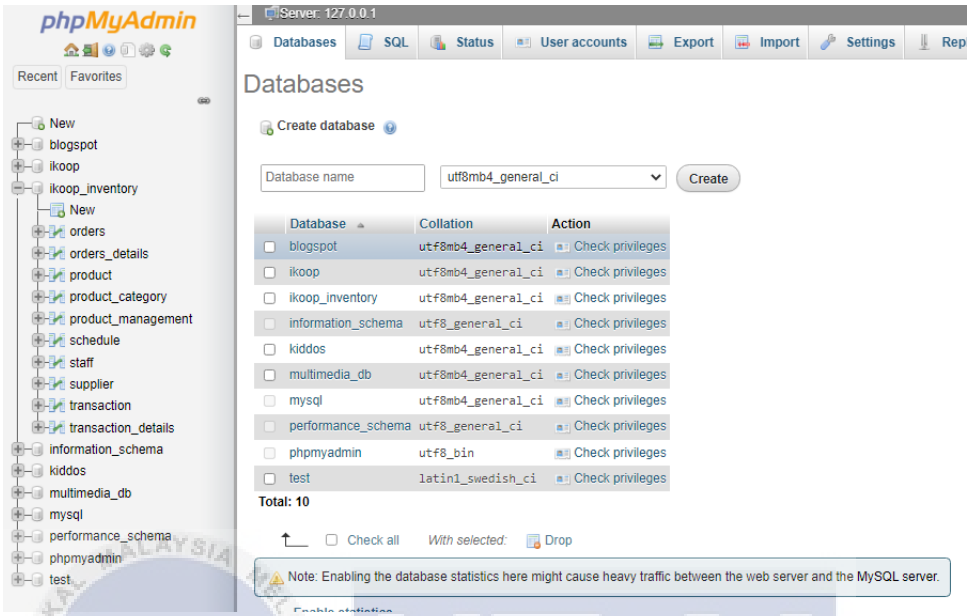

Figure 5.8: PHPMyAdmin page.

Step 2: Insert the name of the database.

Figure 5.9: Create Database.

Step 3: Click New on the left to create new table.

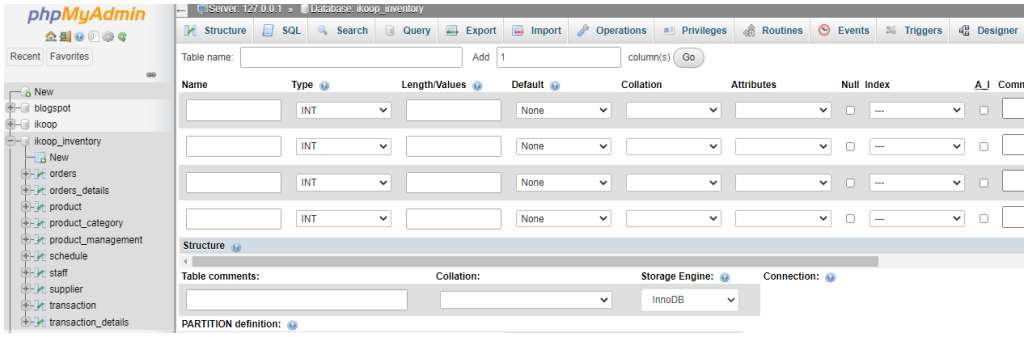

Figure 5.10: Add new table in the database.

Step 4: Insert the table name and fill in the attribute in the field and then click Save button.

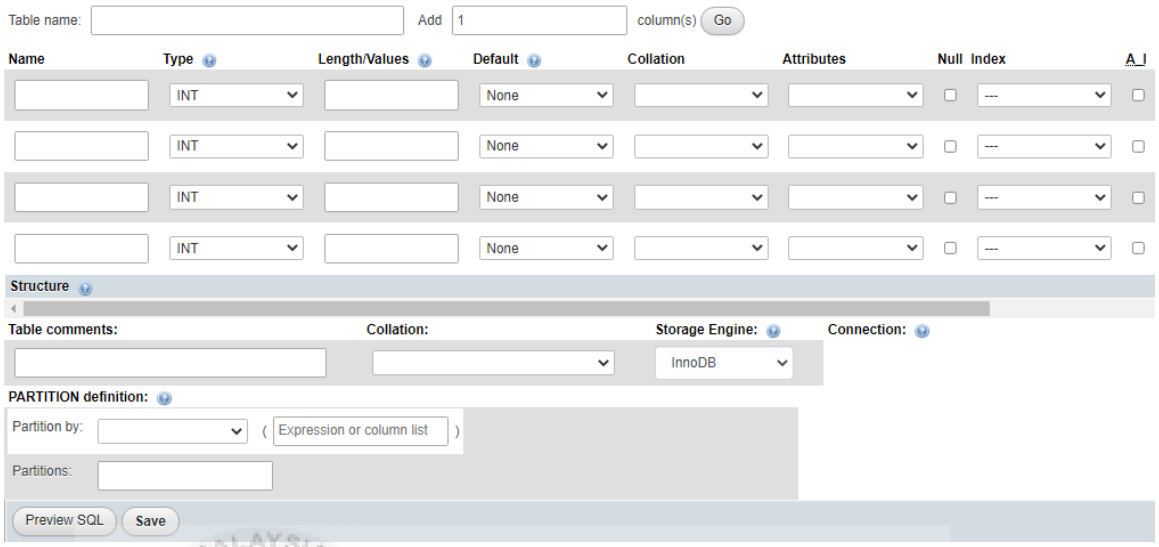

Figure 5.11: Add atrribute into table.

5.3 Database Implementation

5.3.1 Data Definition Language (DDL)

Data Definition Language (DDL) could be a SQL command that make and manipulate the table in relative info. It wont to create, alter and drop from the database wherever it includes table and indexes.

5.3.1.1 Create Table Command

```
CREATE TABLE `orders` (
  `orders_id` bigint(20) UNSIGNED NOT NULL,
  `staff_id` bigint(20) UNSIGNED DEFAULT NULL,
  `supplier_id` bigint(20) UNSIGNED DEFAULT NULL,
  `status` smallint(6) DEFAULT NULL,
  `dateorder` date DEFAULT NULL,
  `datereceived` date DEFAULT NULL,
  `resitpath` varchar(200) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `remarks` varchar(1000) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
```

```
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

Figure 5.12: Create table Orders

```
CREATE TABLE `orders_details` (
  `orders_details_id` bigint(20) UNSIGNED NOT NULL,
  `order_id` bigint(20) UNSIGNED DEFAULT NULL,
  `product_id` bigint(20) UNSIGNED DEFAULT NULL,
  `quantity` int(11) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

Figure 5.13: Create table Orders Details


```
CREATE TABLE `product` (
  `product_id` bigint(20) UNSIGNED NOT NULL,
  `name` varchar(200) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `description` varchar(10000) COLLATE utf8mb4_unicode_ci
DEFAULT NULL,
  `price` double(8,2) DEFAULT NULL,
  `productype` smallint(6) DEFAULT NULL,
  `status` smallint(6) DEFAULT NULL,
  `supplier_id` bigint(20) UNSIGNED DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

Figure 5.14: Create table Product

```
CREATE TABLE `product_category` (
  `product_category_id` bigint(20) UNSIGNED NOT NULL,
  `category_desc` varchar(100) COLLATE utf8mb4_unicode_ci
DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

Figure 5.15: Create table Product Category

```
CREATE TABLE `product_management` (
  `product_management_id` bigint(20) UNSIGNED NOT NULL,
  `product_id` bigint(20) UNSIGNED DEFAULT NULL,
  `quantity` int(11) DEFAULT NULL,
  `productype` smallint(6) DEFAULT NULL,
  `price` double(8,2) DEFAULT NULL,
  `category` bigint(20) UNSIGNED DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```


Figure 5.16: Create table Product Management

```
CREATE TABLE `schedule` (
  `schedule_id` bigint(20) UNSIGNED NOT NULL,
  `date` date DEFAULT NULL,
  `start_time` time DEFAULT NULL,
  `end_time` time DEFAULT NULL,
  `type` smallint(6) DEFAULT NULL,
  `staff_id1` bigint(20) UNSIGNED DEFAULT NULL,
  `staff_id2` bigint(20) UNSIGNED DEFAULT NULL,
  `staff_id3` bigint(20) UNSIGNED DEFAULT NULL,
  `supplier_id1` bigint(20) UNSIGNED DEFAULT NULL,
  `supplier_id2` bigint(20) UNSIGNED DEFAULT NULL,
  `supplier_id3` bigint(20) UNSIGNED DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

Figure 5.17: Create table Schedule

```
CREATE TABLE `staff` (
  `staff_id` bigint(20) UNSIGNED NOT NULL,
  `email` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `password` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `role` smallint(6) NOT NULL,
  `fname` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `lname` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `phone` varchar(20) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `photo_path` varchar(200) COLLATE utf8mb4_unicode_ci
DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```


Figure 5.18: Create table Staff

```
CREATE TABLE `supplier` (
  `supplier_id` bigint(20) UNSIGNED NOT NULL,
  `companyname` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `email` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `password` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `companyaddress` varchar(255) COLLATE utf8mb4_unicode_ci
DEFAULT NULL,
  `phone` varchar(20) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `status` smallint(6) DEFAULT NULL,
  `bankacc` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `category` bigint(20) UNSIGNED DEFAULT NULL,
  `banktype` smallint(6) DEFAULT NULL,
  `photo_path` varchar(200) COLLATE utf8mb4_unicode_ci DEFAULT
NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

Figure 5.19: Create table Supplier

```
CREATE TABLE `transaction` (
  `trans_id` bigint(20) UNSIGNED NOT NULL,
  `staff_id` bigint(20) UNSIGNED DEFAULT NULL,
  `date` date DEFAULT NULL,
  `remarks` varchar(10000) COLLATE utf8mb4_unicode_ci
DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

Figure 5.20: Create table Transaction


```
CREATE TABLE `transaction_details` (
  `transaction_details_id` bigint(20) UNSIGNED NOT
NULL,
  `trans_id` bigint(20) UNSIGNED DEFAULT NULL,
  `product_id` bigint(20) UNSIGNED DEFAULT NULL,
  `quantity` int(11) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

Figure 5.21: Create table Transaction Details

```
ALTER TABLE `staff`
  ADD PRIMARY KEY (`staff_id`);
```

Figure 5.22: Index for table staff

```
ALTER TABLE `supplier`
 ADD CONSTRAINT `supplier_category_foreign` FOREIGN KEY (`category`) REFERENCES `product_category` (`product_category_id`) ON DELETE CASCADE;
```

Figure 5.23: Constraint for table supplier

5.3.2 Stored Procedures

Description	Stored Procedures
To insert supplier details	CREATE DEFINER=`root`@`localhost` PROCEDURE `create_supplier` (IN `V_EMAIL` VARCHAR(50), IN `V_PASSWORD` VARCHAR(255), IN `V_ADDRESS` VARCHAR(255), `V_PHONE` VARCHAR(20), IN `V_NAME` VARCHAR(100), IN `V_BANKACC` VARCHAR(100), IN `V_CATEGORY` INT, IN `V_BANKTYPE` INT) BEGIN DECLARE C_DATE VARCHAR (20) ; DECLARE V_STATUS INT ; SET V_STATUS = 1; SET C_DATE=NOW( ); INSERT INTO supplier (COMPANYNAME,EMAIL,PASSWORD,COMPANYADDRESS,PHONE,STATUS,BANKACC,CATEGORY,BANKTYPE,CREATED_AT,UPDATED_AT) VALUES (V_NAME,V_EMAIL,V_PASSWORD,V_ADDRESS,V_PHONE,V_STATUS,V_BANKACC,V_CATEGORY,V_BANKTYPE,C_DATE,C_DATE);
To insert teacher details	CREATE DEFINER=`root`@`localhost` PROCEDURE `create_teacher` (IN `V_EMAIL` VARCHAR(50), IN `V_PASSWORD` VARCHAR(255), IN `V_FNAME` VARCHAR(100), IN `V_LNAME` VARCHAR(100), IN `V_PHONE` VARCHAR(20)) BEGIN DECLARE C_DATE VARCHAR (20) ; DECLARE V_ROLE INT; SET C_DATE=NOW( ); SET V_ROLE = 2;

	INSERT INTO staff (EMAIL,PASSWORD,ROLE,FNAME,LNAME,PHONE,CREATED_AT,UPDATED_AT) VALUES (V_EMAIL,V_PASSWORD,V_ROLE,V_FNAME,V_LNAME,V_PHONE,C_DATE,C_DATE);

5.3.3 Data Loading Process

The i-KOOP Management System is a system that store data and process data to provide the accurate information of the product, sales and staff. It is to support issues with regards to incorrect stock values. All the data will be stored into the database.

5.4 Conclusion

In conclusion, the chapter is all over regarding the code development atmosphere setup wherever it provides a way to install XAMPP and MySQL in Windows 10 platform. In addition, the system are going to be designed on business logic, which incorporates the information Data Definition Language (DDL), info table creation, constraint commands and data loading process.

CHAPTER 6: TESTING

6.1 Introduction

Testing is that the way of classify a system or its part to see whether or not or not it accomodated the mandatory needs. In another words, testing is the proceeding of running a system to spot any gaps, errors, or missing requirements that are inconsistent with the particular requirements. In this chapter, the i-KOOP Management System will be verify and validate.

6.2 Test Plan

The test plan is product description, test objectives, test strategy, scope, schedule, procedures, check resources and product. Test plans are essential within the development of the software package as define what testing desires doing to confirm the software is straightforward. Test plans also has detailed of understanding workflow where it will be tested to determine the design and find bugs.

6.2.1 Test Organization

In i-KOOP Management System, the test organization is include two users which is teacher and the system admin. In the table below, it will provide the testing based on the responsibilities.

Table 6.1: The list of responsibility.

Tester ID	Users	Responsibilities
Tester1	Teacher	- Test system based on the test script. - Testing teacher modules. - Error detection.
Tester2	System Admin	- Test system based on the test script. - Test system admin modules. - Error detection.

6.2.2 Test Environment

A take a look at setting could be a assortment of software, hardware, operative system, tools and network that enables testing groups to execute the test cases.

Table 6.2: Test Environment Hardware

Name	Description
Brand	HP
Processor	AMD A6-5200 APU with Radeon(TM) HD Graphics 2.00 GHz
Memory (RAM)	8.00 GB
System Type	64-bit operating system, x64-based processor

Table 6.3: Test Environment Software

No.	Software	Description
1.	Sublime Text	To write the PHP code.
2.	Web Server	XAMPP is to create the web server and for the deployment.
3.	Operating System	Windows 10 is to manage the hardware and software to run the system.
4.	Web browser	Google Chrome which is to test the system interface and run the PHP source code.
5.	Microsoft Word	Use to write the final report.

6.2.3 Test Schedule

Table 6.4: Test Schedule

Testing task	Testing Activity
Login	Unit testing and integration testing.
Registration	Unit testing and integration testing.
Time table	Unit testing and integration testing.
History selling	Unit testing and integration testing.
Product list	Unit testing and integration testing.
History order	Unit testing and integration testing.

6.3 Test Strategy

A test strategy is a rule to be followed to achieve the test objective and execution of test variety mentioned in the testing plan. Test strategy could be a set of tips that explains the look and determineshowever testing has be done and regarding the final approaches.

6.3.1 Classess of tests

i. Error Handling Test

The error handling test is ensure that valid and exact data or information form user is entered into the input field. It is to ensure the user insert the correct information before the data will be stored in the database. After that, the error message will be displayed on the screen to notify that the users insert the incorrect data into the system.

ii. Security Test

Security test is to detect the security mechanism of the system. In i-KOOP Management System login page, it will verify and validate the email and password of the user login to protects their data or information.

6.4 Test Design

Test design is development and creation test which is to specifying the test from test condition to test software. The aim is to ensure the requirement is same with the client needed.

6.4.1 Test Description

The test description can make a case for the test suit definition, test type, prerequisites, test requirements, gradual procedure in every test case, identification and expected results. Develop and document each unit test case.

1) Module login

Login is a process where user login into the account to manage the product. User login into their account using email and password. The password must be accurate and correct password. Table below show how the test case and will explain more.

Table 6.5: Test Description of user login

Test Case ID	Test Requirement	Step	Expected Output
TC1_1	Verify login operation is possible if the email and password are accurate	1. Login page 2. Insert accurate email 3. Insert accurate password 4. Click login button	User login success. Display “Success! Login success”.
TC1_2	Verify login operation is possible if the email and password are wrong	1. Login page 2. Insert accurate email 3. Insert incorrect password 4. Click login button	Login failed. It will display error message “Error! Wrong password”
TC1_3	Verify login operation is not possible if the email and password are blank	1. Login page 2. Click login button	Login failed. It will display error message “Error! Email invalid”

2) Module registration

Registration is a process where user create a new account in the system for the new user. The user need to fill out the field which is email, first name, last name and phone number. The information must be in correct format and accurate data. The table 6.6 and table 6.7 below will show how the test of registration are conducted.

Table 6.6: Test Description of teacher registration

Test Case ID	Test Requirement	Step	Expected Output
TC2_1	Verify the teacher registration is possible if all the input field data are accurate	1. Operate to new teacher register page. 2. Fill in the accurate data in input field email, first name, last name and phone number. 3. Click on Submit button.	New teacher register successful. Display "Success! This data has been created"
TC2_2	Verify the teacher registration is not possible if few input field data are wrong	1. Operate to new teacher register page. 2. Insert incorrect format of email in the input field. 4. Fill in the correct data in input field first name, last name and phone number. 3. Click on Submit button.	Failed to register. It will display the error message "Please include an '@' in the email address".
TC2_3	Verify the teacher registration is not possible if the input field are blank	1. Operate to new teacher register page. 2. Click on Submit button.	Failed to register. It will display the error message "Please fill out this field".

Table 6.7: Test Description of supplier registration

Test Case ID	Test Requirement	Step	Expected Output
TC3_1	Verify the supplier registration is possible if all the input field data are accurate	1. Operate to new supplier register page. 2. Fill in the correct data in input field email, company name, company address, phone number, supplier category, bank type and bank account number. 3. Click on Submit button.	New supplier register successful. Display “Success! This data has been created”.
TC3_2	Validate the supplier registration is not available if few input field data are invalid	1. Operate to new supplier register page. 2. Insert incorrect format of email in the input field. 3. Fill in the correct data in input field company name, company address, phone number, supplier category, bank type and bank account number. 4. Click on Submit button.	Failed to register. It will display the error message “Please include an ‘@’ in the email address”.
TC3_3	Verify the supplier registration is not available if the input field are blank	1. Navigate to new supplier register page. 2. Click on Submit button.	Failed to register. It will display the error message “Please fill out this field”.

3) Module time table

Time table is a process where admin need to create schedule for teacher. The time table will help the teacher to manage the cooperative on the date and time given based on the time table. Table 6.8 below show how the test is being conducted.

Table 6.8: Test Description of time table

Test Case ID	Test Requirement	Step	Expected Output
TC4_1	Verify the time table or schedule is possible if the time table form is valid	1. Operate to new time table page. 2. Fill in the correct data in the field of date, start time, end time, teacher number 1, teacher number 2 and teacher number 3. 3. Click on Submit button.	New time table successfully added. Display “Success! This data has been created”.
TC4_2	Verify the time table is not possible if certain field are blank	1. Operate to new time table page. 2. Do not insert the date. 3. Fill in the correct data in the field of start time, end time, teacher number 1, teacher number 2 and teacher number 3. 4. Click on Submit button.	Failed to add new time table. It will display the error message “Please fill out this field”.
TC4_3	Verify the time table is not possible if certan field are invalid	1. Operate to new time table page. 2. Fill in the data in the field of date, start time, and end time. 3. Do not select the teacher. 4. Click on Submit button.	Failed to add new time table. It will display the error message “Please fill out this field”.
TC4_4	Verify the time table is not possible if the input field are blank	1. Operate to new time table page. 2. Click on Submit button.	Failed to add new time table. It will display the error message “Please fill out this field”.

4) Module history selling

History selling is a process where user need to create or update the product which has been sold. It is to help staff to manage the quantity of the product in the system. For history selling, the user need to insert the quantity of the product. Table 6.9 show how the test are conducted.

Table 6.9: Test Description of history selling

Test Case ID	Test Requirement	Step	Expected Output
TC5_1	Verify the history selling is possible if the table form is valid	1. Operate to selling product page. 2. Fill in the correct quantity of sold and remarks. 3. Click on Submit button.	Selling product successfully added. Display “Success! Selling has been created”.
TC5_2	Verify the the history selling is not possible if the input field are blank	1. Opertae to new time table page. 2. Click on Submit button.	Failed to add new selling product. It will display the error message “Your quantity is empty. Please fill the quantity.”.

5) Module product list

Product list is a process where supplier create a new product in the system. Supplier need to fill out the correct information to insert new product which is product name, product description, product type and product price. Table 6.10 below show how the test are conducted.

Table 6.10: Test Description of product list by supplier

Test Case ID	Test Requirement	Step	Expected Output
TC6_1	Verify the product list is possible if the table form is accurate	1. Operate to product list page. 2. Fill in the correct information of product name, product description, product type and product price. 3. Click on Submit button.	New product successfully added.
TC6_2	Verify the the product list is not possible if the input field are wrong	1. Operate to product list page. 2. Fill in the accurate information of product name, product description and product type. 3. Product price is blank. 4. Click on Submit button.	Failed to add new selling product. It will display the error message “Please fill out this field.”.
TC6_3	Verify the the product list is not	1. Operate to create new product page.	Failed to add new selling product. It

	possible if the input field are blank	2. Click on Submit button.	will display the error message “Please fill out this field.”.

6) Module history order

History order is a process where supplier received order from the teacher. Supplier need to fill out the field which is deliver date, start time, end time and remarks. Table 6.11 below show how the test are conducted.

Table 6.11: Test Description of history order by supplier

Test Case ID	Test Requirement	Step	Expected Output
TC7_1	Verify the order details is possible if the input field is accurate	1. Operate to order details page. 2. Fill in the accurate information date to deliver order, start time, end time, status order and remarks. 3. Click on Submit button.	Order details successfully approved.
TC7_2	Verify the order details is not possible if the field are blank	1. Operate to order details page. 2. Fill in the correct information date to deliver order, start time, end time and remarks. 3. Status order are blank. 4. Click on Submit button.	Failed to add new order details. It will display the error message “Oops.. Something went wrong”.
TC7_3	Verify the order details is not possible if the input field are blank	1. Operate to add new order details page. 2. Click on Submit button.	Failed to add new order details. It will display the error message “Please fill out this field.”.

6.4.2 Test Data

1) Module login

Table 6.12: Testing data of user login

Test Case ID	Email	Password
TC1_1	Faizal71@gmail.com	123
TC1_2	Faizal71@gmail.com	abc
TC1_3	Blank form	Blank form

2) Module registration

Table 6.13: Testing data of teacher registration

Test Case ID	Email	First name	Last name	Phone No
TC2_1	Anuarabdullah73@gmail.com	Anuar	Abdullah	0194471011
TC2_2	Anuarabdullah73	Anuar	Abdullah	0194471011
TC2_3	Blank form	Blank form	Blank form	Blank form

Table 6.14: Testing data of supplier registration

Test Case ID	Email	Company name	Company address	Phone No	Supplier category	Bank type	Bank account number
TC3_1	Schooloutfitsdnbhd @gmail.com	School outfit	no 17G tingkat 2, jalan merbau, sg besi. selangor	+0366687191	Clothes	Maybank	514112180116
TC3_2	Schooloutfitsdnbhd @’’gmail.com	School outfit	no 17G tingkat 2, jalan	+0366687191	Clothes	Maybank	514112180116

			merbau, sg besi. selangor				
TC3_3	Blank form	Blank form	Blank form	Blank form	Blank form	Blank form	Blank form

3) Module time table

Table 6.15: Testing data of time table

Test Case ID	Date	Start time	End time	Teacher number 1	Teacher number 2	Teacher number 3
TC4_1	28/08/2021	9.00 am	11.00 am	Siti Zubaidah	Anuar Abdullah	Blank form
TC4_2	Blank form	2.00 PM	4.00 PM	Kairol Amin	Blank form	Blank form
TC4_3	30/8/2021	1.30 pm	Blank form	Muhd Faizal	Blank form	Blank form
TC4_4	Blank form	Blank form	Blank form	Blank form	Blank form	Blank form

4) Moduel history selling

Table 6.16: Testing data of history selling

Test Case ID	Quantity of sold	Remarks
TC5_1	5	need to increase the stock
TC5_2	Blank form	Blank form

5) Module product list

Table 6.17: Testing data of product list by supplier

Test Case ID	Product name	Product description	Product type	Product price
TC6_1	Faber castle eraser	Good quality of product	Piece	RM 0.70
TC6_2	Faber castle eraser	Good quality of product	Piece	Blank form
TC6_3	Blank form	Blank form	Blank form	Blank form

6) Module history order

Table 6.18: Testing data of history order by supplier

Test Case ID	Date to deliver order	Start time	End time	Status order	Remarks
TC7_1	30/8/2021	1.35 pm	2.30 pm	Accept	The product will be deliver on the stated date.
TC7_2	30/8/2021	1.30 pm	2.30 pm		The product will be deliver on the stated date.
TC7_3	Blank form	Blank form	Blank form	Blank form	Blank form

6.5 Test Result and Analysis

Table 6.19: Testing result of user login

Test Case ID	Description	Result
TC1_1	Display popup message “Login Success”	Pass
TC1_2	Display popup message “Error! Wrong password”	Pass
TC1_3	Display popup message “Error! Email invalid”	Pass

Table 6.20: Testing result teacher registration

Test Case ID	Description	Pass/Fail
TC2_1	Display popup message “Success! This data has been created”	Pass
TC2_2	Display error “Please include an ‘@’ in the email address”.	Pass
TC2_3	Display error “Please fill out this field”	Pass

Table 6.21: Testing result supplier registration

Test Case ID	Description	Pass/Fail
TC3_1	Display popup message “Success! This data has been created”	Pass
TC3_2	Display error “Please include an ‘@’ in the email address”.	Pass
TC3_3	Display error “Please fill out this field”	Pass

Table 6.22: Testing result of time table

Test Case ID	Description	Pass/Fail
TC4_1	Display popup message “Success! This data has been created”	Pass
TC4_2	Display error “Please fill out this field”.	Pass
TC4_3	Display error “Please fill out this field”	Pass
TC4_4	Display error “Please fill out this field”	Pass

Table 6.23: Testing result of history selling

Test Case ID	Description	Pass/Fail
TC5_1	Display popup message “Success! Selling has been created”	Pass
TC5_2	Display error “Your quantity is empty. Please fill the quantity”.	Pass

Table 6.24: Testing result of product list by supplier

Test Case ID	Description	Pass/Fail
TC6_1	Display popup message “Success! Product has been created”	Pass
TC6_2	Display error “Please fill out the field”.	Pass
TC6_3	Display error “Please fill out the field”.	Pass

Table 6.25: Testing result of history order by supplier

Test Case ID	Description	Pass/Fail
TC7_1	Display popup message “Success! Order has been updated”	Pass
TC7_2	Display error “Oops.. Something went wrong”.	Pass
TC7_3	Display error “Please fill out the field”.	Pass

6.6 Conclusion

In conclusion, in this chapter is to do testing phase where to test the bugs and defect of the system. Testing is to help identifies the error and by checking the system. It consist test schedule, test strategy, test design, test description and test result and analysis.

CHAPTER 7: CONCLUSION

7.1 Introduction

The chapter is about the conclusion of i-KOOP Management System where it consists of the weakness and strengths. After that, there are a propositions for improvements on how to improved the system to be better according on the weakness and strength. Next is to defined the project contribution.

7.2 Observation on Weaknesses and Strengths

7.2.1 Weakness

a. Do not has notification features

The system does not have any reminder or notification to notify the time table. The user might forgot their schedule.

b. Backup and recovery features

The system does not have automatic recovery for data backup functions. The data stored in the system may not be recoverable if the system damaged, loss, or data is accidentally deleted by user.

c. Forgot password features

The system does not have or provide forgot password function where to allow the user to get their new password and verify through their email.

7.2.2 Strengths

a. Error reduction

The i-KOOP Management System has a functional to update the quantity of the product where the result will automatically calculate and keep or store in the database to avoid the wrong calculation by the user. Next, the system also will generate a report based on the database.

7.3 Propositions for Improvements

a. Notifications

Notification to remind the user of the schedule or time table through email. For example, the system will send the notification reminder to the user one or two days before the schedule to remind the time and date.

b. Backup and recovery

Add automatic backup and recovery function. For example, the data will be automatically backup and done at 12.00 am every day to confirm the information is up to date. Once that, the system are recover mechanically when the system is corrupted.

c. Forgot password

Forgot password function can be improved in the system. For example, the system will send their new password through their email if the user forgot the password.

7.4 Project Contribution

The project contribution has several parties which is university, company and individual. First, the project is owned and belong to Universiti Teknikal Malaysia Melaka (UTeM). In addition, other universities can also use this project as a reference material in developing a project such as i-KOOP Management System.

Next, a contribution for company to manage the quantity of the product, to designed an accurate system and more efficient. In the system, it will automatically calculate the quantity of the product and store the data into the database. Therefore, it will help to enhance the effectiveness on manage the information easily and reduce the human error.

Finally, a contribution for individual where it can be used to gain more knowledge on how to develop a system. Plus, the system also may help to give benefits and encourage the individual to make a improvement on their system.

7.5 Conclusion

In conclusion, the target and also the scope has expressed within the Chapter 1 are ready to achieve. The objective of the i-KOOP Management System is to maintain comprehensive human resource records that includes details of staff and supplier, facilitate product inventory that includes product sales, product re- ordering and product disposal and lastly to generate a report of the sales. After that, Database Life Cycle (DBLC) is choose as the methology in i-KOOP Management System.

REFERENCES

1. W3schools. Retrieved from w3schools.com
   https://www.w3schools.com/

2. School cooperatives (2021). Retrieved from www.sutama.edu.my
   https://www.sutama.edu.my/2021/01/koperasi-sekolah-akan-mula-beroperasi/

3. Data Flow Diagram. Retrieved from www.lucidchart.com
   https://www.lucidchart.com/blog/data-flow-diagram-tutorial

4. Database Life Cycle (DBLC). Retrieved from http://www.myreadingroom.co.in/
   http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/506-database-development-life-cycle.html

5. Software testing (2021). Retrieved from www.softwaretestinghelp.com
   https://www.softwaretestinghelp.com/database-testing-process/