E-FEELICIOUS ORDERING SYSTEM

FATIN NAJIEHAH BINTI MOHD ZAMELI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2020/2021

E-FEELICIOUS ORDERING SYSTEM

FATIN NAJIEHAH BINTI MOHD ZAMELI

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

2020/2021

DECLARATION

I hereby declare that this project report entitled

E-FEELICIOUS ORDERING SYSTEM

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

STUDENT : FATIN NAJIEHAH BINTI MOHD ZAMELI Date : 14 JUNE 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 : 05 September 2021

DECLARATION

I hereby declare that this project report entitled

DEDICATION

My Final Year Project is dedicated to my beloved parents, Encik Mohd Zameli and Puan Zaliha, whose always give blessings and put me in their prayers. Words of encouragement from them I still remember and hold on tight to it.

I also dedicate this Final Year Project to my supervisor Ts. Dr. Yahaya bin Abd Rahim who have supported me, kept me motivated, and assisted me with his ideas along the process. I will always be grateful for all of his initiatives in assisting me throughout developing my Final Year Project.

I also want to express my gratitude to all of my friends for supporting me throughout the development of my final year project. You all been my biggest supporters.

ACKNOWLEDGEMENTS

This is to acknowledge everyone who contributed to making this final year project a reality without whom this would not be possible. To begin, I would like to express my gratitude to my supervisor, Ts. Dr. Yahaya bin Abd Rahim, who provided invaluable assistance, spent his time and effort, and taught me how to accomplish it. This final year project would never have been completed without his assistance.

I want to express my appreciation to who have always been encouraging of me and have clarified my doubts, as well as my parents, who have also significantly contributed to the completion of my final year project. I want to take this opportunity to thank them for their support, and I hope that they will always continue to do so in the future.

A project is a bridge between theoretical and practical learning and with this thinking I worked on my final year project and was able to complete it because of the assistance and efforts of everyone who assisted me.

Again, I would like to express my gratitude to my students and friends for their support and encouragement in developing and completing my project. All of these owe me a debt of gratitude. I was only able to complete my project and make it a positive and pleasurable journey because of them.

Thanks for all your encouragement!

ABSTRACT

Here, an E-Feelicious Ordering system is proposed to make the ordering procedure easier. The proposed system provides a user interface and updates the menu to include all accessible alternatives, making customer task become easier. Customers can place orders for several desserts and check order details before checking out. The customer receives an order confirmation. The order is queued and updated in the database before being returned in real time. This system helps the employees in checking orders in real time and processing them effectively and easily with minimal mistakes. This project aims to replace the old and inefficient ordering method which is commonly used by all sellers with a new and more efficient approach. The current ordering process is inconvenient for both employees and customers since it involves a great deal of human labor. The manual work done by the employees will result in certain human mistakes, such as giving the wrong bill to the clients, the waiter's handwriting being unattractive, and the order sequence being improper. Customers will be dissatisfied with the seller as a consequence of all of these human mistakes. As a result, this ordering system was created to enable the seller in better managing their product. The time taken to place an order has been cut in half thanks to this ordering system.

ABSTRAK

Di sini, E-Feelicious Ordering System dicadangkan untuk mempermudahkan prosedur pesanan. Sistem yang dicadangkan menyediakan antara muka pengguna dan mengemas kini menu untuk memasukkan semua alternatif yang dapat diakses, menjadikan tugas pelanggan menjadi lebih mudah. Pelanggan boleh membuat pesanan untuk beberapa pencuci mulut dan memeriksa perincian pesanan sebelum mendaftar keluar. Pelanggan menerima pengesahan pesanan. Pesanan itu diatur dalam barisan dan dikemas kini dalam pangkalan data sebelum dikembalikan dalam masa nyata. Sistem ini membantu pekerja dalam memeriksa pesanan dalam masa nyata dan memprosesnya dengan berkesan dan mudah dengan kesalahan minimum. Projek ini bertujuan untuk menggantikan kaedah pesanan lama dan tidak cekap yang biasanya digunakan oleh semua penjual dengan pendekatan baru dan lebih cekap. Proses pesanan semasa tidak selesa bagi pekerja dan pelanggan kerana melibatkan banyak tenaga kerja manusia. Kerja manual yang dilakukan oleh pekerja akan mengakibatkan kesilapan manusia tertentu, seperti memberi tagihan yang salah kepada pelanggan, tulisan tangan pelayan tidak menarik, dan urutan pesanan tidak tepat. Pelanggan akan tidak berpuas hati dengan penjual akibat daripada kesilapan manusia ini. Hasilnya, sistem pesanan ini dibuat untuk membolehkan penjual menguruskan produk mereka dengan lebih baik. Masa yang diambil untuk membuat pesanan telah dipotong separuh berkat sistem pesanan ini.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FYP	-	Final Year Project
PSM	-	Projek Sarjana Muda
EFOS	-	E-Feelicious Ordering System
PHP	-	Hypertext Preprocessor
CSS	-	Cascading Style Sheets
HTML	-	Hypertext Markup Language
SQL	-	Structured Query Language

CHAPTER 1: INTRODUCTION

1.1 Introduction

This system has been developed to fulfill the "Project Sarjana Muda" requirement. This system is based on the case study from the Feelicious.co Dessert online business. Customers can place the order via the internet. This will make the ordering process which had been done manually become much easier, whereby with the internet access or connection, customers are able to make the purchasing of desserts within seconds. On the other hand, this system also helps the Feelicious.co Dessert management to manage the ordering information such as products and customers information in more effective and precise way. Finally, all the information will be inserted into the Feelicious.co Dessert databases, which previously all the ordering details and product information of the Feelicious.co Dessert will be keep into the file-based system. This system allows the manager to view, add and update information regarding the desserts (products). For examples, admin can insert and update the stock of products that available, availability of the dessert, can edit the price of the dessert. The staff can view the order list and update the order. They also can view the customer details. Customer can order by their own by looking at the menu of the dessert offered by the Feelicious.co Dessert. They can order by insert their information needed, choose the desserts, and complete the order. They also can view the order status and the staff who prepared the order.

The E-Feelicious Ordering System (EFOS) is a system for a business to make the record of the orders and product. The current system is very inefficient since they must record each orders manually. With this system the operations become more organized and less time-consuming. This system will allow admin to maintain their record in one place which will be easy to manage and will also reduce the risk of lost and errors. This system makes the daily tasks become easier, simpler, and faster for everyone from a small independent business.

1.2 Problem Statement

1. Time consuming

   The administrator must keep track of the previous data of the customer. Extra time is needed to search the history of the order information.

2. Wrong order

   Due to a misinterpretation of handwriting order, the potential of making a mistake was higher.

3. Inconvenience

   The staff might lose the conversation with customer or misplace the order paper.

1.3 Objective

I. To provide a system that able to do online ordering which is via the internet for customers

II. To systemize all the ordering information and customers information and store the data into the database

III. To provide online menu for the customer

IV. To calculated price of order

1.4 Scope of the Project

Module to be developed

i. Login Module

Login function is needed to identify whether the user is customer, manager, or staff. Different user login can perform different tasks.

ii. Update profile

Customers can update their profile. They can update their name, email, phone number and address.

iii. Order module

The customer can place the order after they view the menu and add to the cart.

iv. View order details module

The admin and staff can view the full order details of the customer.

v. View menu

The customers can view the menu through this system. The customers will get to know about the desserts' name, description, and price.

vi. Add new menu

The admin can add new menu to the system. Admin can add the name, price, description, and picture of the dessert. After admin inserted, the customer can view it through the menu page.

vii. Update menu

Admin can update any changes to the dessert details. The admin can update the name, price, picture, description, stock, and availability of the menu.

viii. View staff

Admin can view all staff details including their name, email, phone number and their status.

ix. Update order details
Admin can update the order details to be display to the customer. So that customer will know the order status.

Table 1.1: Target users

System scope	Explanation
Admin of the Feelicious.co Dessert	- View all the orders from customer - Update the order status - View all the dessert details - Edit the dessert details - Update the availability of the dessert - View all staff information’s
Staff of the Feelicious.co Dessert	- View all the orders from customer - Update the staff information in orders
Customer	- Signup as new user - Login - View all the desserts - Add the dessert to cart - Place order for the dessert - View order status

1.5 Project Significance

According to the technology that has been designed, data can be simply lost and damaged. Because they are currently taking order through WhatsApp and a paper form in their manual or existing system. This system is made to also can reduce time, example it will ease the process of taking order that usually is taken manually before this that consume time. The staff are no longer must spend time processing clients' orders, worry less about communication problems, and simplify their order operation process. This system also secure and store information of the orders and customer. So that the order will not lose or damaged.

1.6 Expected Output

i. Output 1: To simplify the whole ordering process.

ii. Output 2: To improve the efficiency of the order management

iii. Output 3: To accept a larger number of orders with ease

iv. Output 4: All the customers' information will be more secure.

1.7 Conclusion

Overall, this chapter is crucial in the development of a system that is more efficient, stable, and user-friendly. Additionally, the system may be constructed quickly because the developer will be familiar with the system's issue statement, objective, and expected output. This chapter is helpful in assisting developers in understanding what should be developed based on the real data provided. To summarize, the new system that will be constructed will be able to improve the current system's functionality and efficacy.

This project's literature review and approach will be discussed in Chapter 2.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

The database life cycle (DBLC) defines the stages involved for implementing a database, starting with requirements analysis, and ending with monitoring and modification. Furthermore, the DBLC never ends because database monitoring, modification, and maintenance are part of the life cycle, and these activities continue long after a database has been implemented. The process of gathering real-world requirements, assessing requirements, creating the data and function of the system, and then executing the operations in the system is known as database system development. To ensuring that the system has been designed efficiently, the Database Life Cycle (DBLC) technique is used as a database development methodology. In section 2.2 Database Methodology, the phases of DBLC are explained. After each stage is completed, the life cycle proceeds to the next stage.

2.2 Database Development Methodology

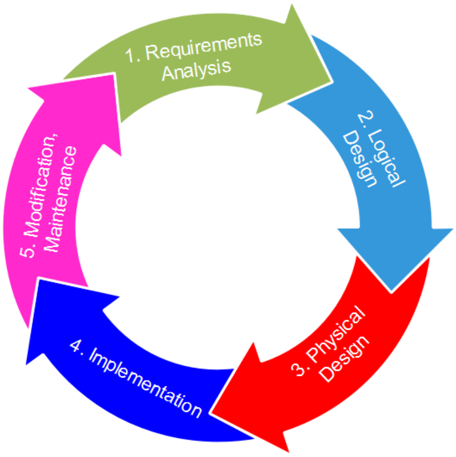


Figure 2.1 Database Life Cycle (DBLC)

Figure 2.1 shows entire Database Life Cycle (DBLC). It consists of requirement analysis, logical design, physical design, implementation, and modification maintenance.

I. Requirement's analysis

Requirements Analysis is the first and most important stage in the *Database Life Cycle*. It is the most labor-intensive for the database designer. This stage involves assessing the informational needs of an organization so that a database can be designed to meet those needs.

II. Logical design

During the first part of Logical Design, a *conceptual model* is created based on the need's assessment performed in stage one. A conceptual model is typically an *entity-relationship (ER) diagram* that shows the tables, fields, and primary keys of the database, and how tables are related (linked) to one another. The tables sketched in the ER diagram are then *normalized*. The normalization process resolves any problems associated with the database design, so that data can be accessed quickly and efficiently.

- Conceptual model: A description of the structure of a database.
- Entity-relationship (ER) diagram: A diagram used during the design phase of database development to illustrate the organization of and relationships between data during database design.
- Normalization: The process of applying increasingly stringent rules to a relational database to correct any problems associated with poor design.

III. Physical Design

The Physical Design stage has only one purpose: to maximize database efficiency. This means finding ways to speed up the performance of the RDBMS. Manipulating certain database design elements can speed up the two slowest operations in an RDBMS: retrieving data from and writing data to a database.

IV. Implementation

During the implementation stage of the DBLC, the tables developed in the ER diagram (and subsequently normalized) are converted into SQL statements. These SQL statements are then executed in the RDBMS to create a database. By this stage in the database life cycle, the System Administrator has installed and configured an RDBMS.

V. Monitoring, modification, and maintenance

A successfully implemented database must be carefully monitored to ensure that it is functioning properly and that it is secure from unauthorized access. The RDBMS usually provides utilities to help monitor database functionality and security. Database modification involves adding and deleting records, importing data from other systems (as needed), and creating additional tables, user views, and other objects and tools. As an organization grows, its information system must grow to remain useful.

Project Schedule and Milestone

Table 2. 1 Gantt Chart of PSM

Activity	Week														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. Introduction and project methodology	■	■													
2. Analysis and design		■	■	■											
3. Implementation					■	■	■	■	■	■	■				
4. Testing											■	■			
5. Maintenance												■	■	■	

Table 2.2 Milestone of PSM

WEEK	START DATE	END DATE	KEY MILESTONE
W1	15/3/2021	21/3/2021	- Final Year Project proposal - Proposal discussion with supervisor - Proposal assessment & verification by supervisor
W2	22/3/2021	28/3/2021	- Proposal submission to Committee via email - Proposal Approval - List of supervisor & title
W3	29/3/2021	4/4/2021	- Submission via PSM Ulearn - Chapter 1: System Development Begins
W4	5/4/2021	11/4/2021	- Development of Chapter 1 - Report and correction for Chapter 1
W5	12/4/2021	18/4/2021	- Development of Chapter 2 begins
W6	19/4/2021	25/4/2021	- Development of Chapter 2 - Report and correction for Chapter 2
W7	26/4/2021	2/5/2021	- Development of Chapter 3 begins
W8	3/5/2021	9/5/2021	- Development of Chapter 3 - Report and correction for Chapter 3
W9	**Mid Semester Break**		
W10	17/5/2021	23/5/2021	- Development of Chapter 4

			- Report and correction for Chapter 4
W11	24/5/2021	30/5/2021	- Project demonstration
W12	31/5/2021	6/6/2021	- Project demonstration - PSM 1 report in progress
W13	7/6/2021	13/6/2021	- Project demonstration - PSM 1 report in progress - Presentation schedule
W14	14/6/2021	20/6/2021	- Project demonstration - Complete PSM 1 draft report
W15	21/6/2021	27/6/2021	- Final presentation
W16	28/6/2021	4/7/2021	- Correction on the PSM 1 draft report - Submit logbook
W17	**Final Examination Weeks**		

2.3 Conclusion

To conclude, this chapter discusses the project's database development technique. To design the system, the methodology is based on the Database Life Cycle (DBLC) approach. Moreover, since database monitoring, modification, and maintenance are all part of the life cycle, and these tasks remain long after a database has been deployed, the DBLC never ends. As a result, the DBLC includes the database's whole lifecycle.

The project analysis will be addressed in detail in the next chapter (Chapter 3). The problem analysis will contain non-functional requirements, functional requirements, and other requirements, as well as potential improvements or solutions.

CHAPTER 3: ANALYSIS

3.1 Introduction

The analysis phase involves a detailed check of the database development strategy that was developed previously. To measure the efficiency relating to the proposed software system, the developer evaluates the database development plan against elements such as expense, period, development platform, programming languages, and predicted the outcomes.

3.2 Problem Analysis

To complete this analysis for Final Year Project, I went to meet Ms. Aina Sufi which is the owner of Feelicious.co to understand and study more about the process of current system and the level of efficiency which is manual system for her business. This business has been operating for 3 years. Ms. Aina took an order from a customer who ordered through the WhatsApp app and jotted it down on paper. It is a difficult process because Ms. Aina manages her business on her own. Ordering and preparation are all done by him.

3.3 The Proposed Improvements/Solutions

The proposed system helps in many ways. It helps customers to easily place their order. Traditionally, people had to make calls to place orders or drive to the restaurants for a take-out, then wait for the food to be prepared and delivered. Sometimes, placing an order on the phone means that there could be mistakes in order. This system developed for make sure you generate more profit and better organize work at the premise, while you also make savings. The system is very easy and convenient for customers to use.

3.4 Requirement Analysis of To-Be System

3.4.1 Functional Requirement (Process Model)

3.4.2 Context Diagram

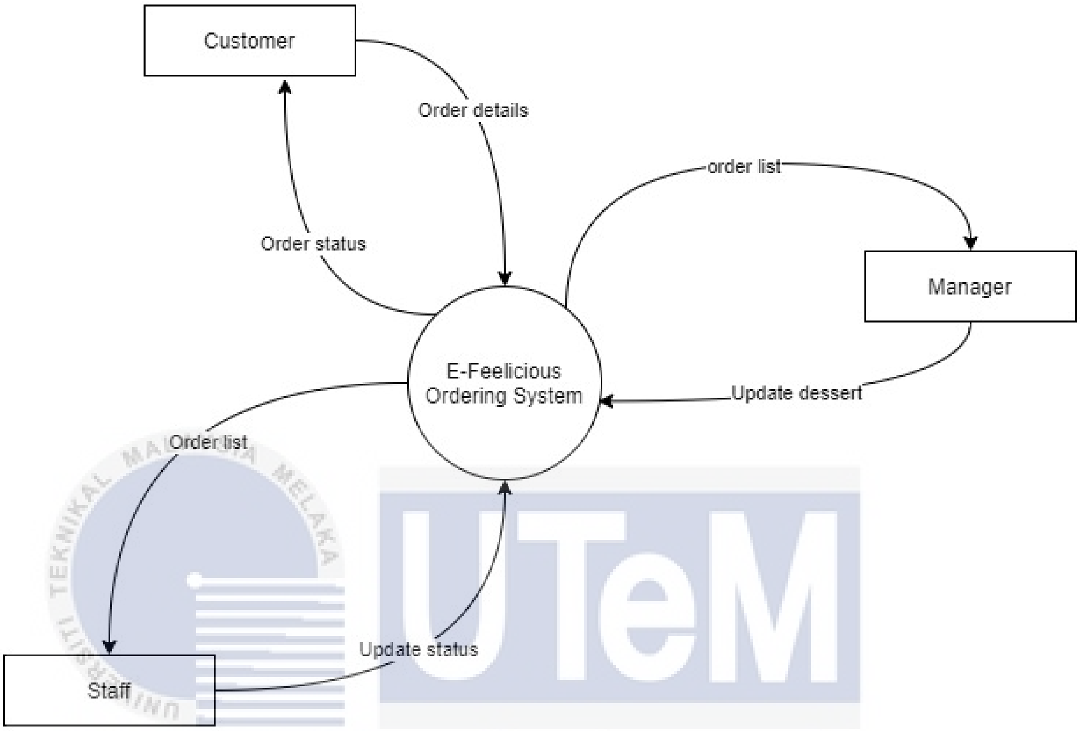


Figure 3.1: EFOS Context Diagram

The figure shows how the system interact with external entities. Customer will place the order and the manager will receive the order. Manager can update the desserts. Manager can edit the desserts, add new dessert, view the details, and update the availability of the dessert. Staff can update the order status and view all the order list

3.4.3 Data Flow Diagram Level 0

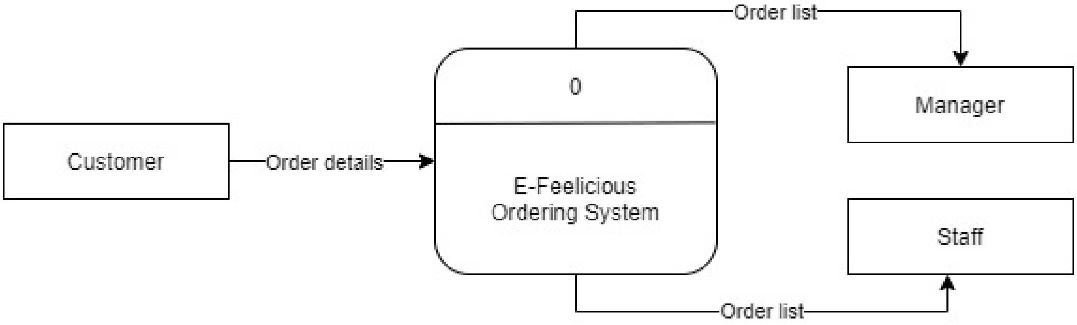


Figure 3.2: EFOS Data Flow Diagram Level 0

At this level, the Input and Output of the system are shown. The system is designed and established across the world with input and output at this level.

3.4.4 Data Flow Diagram Level 1

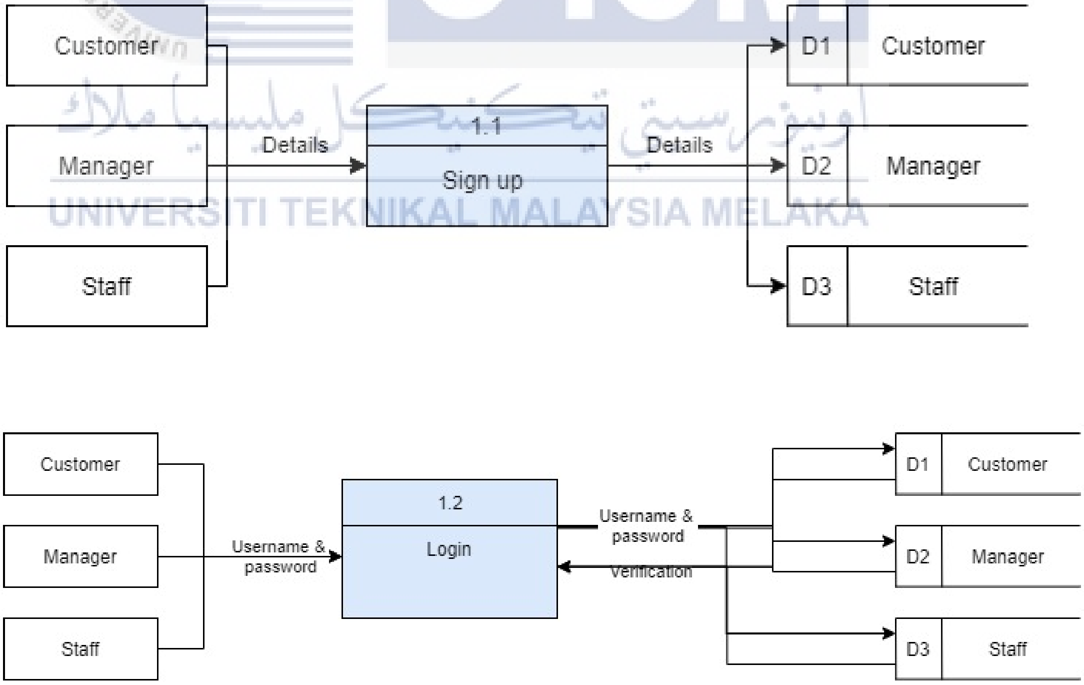


Figure 3.3: Sign up and login process Level 1

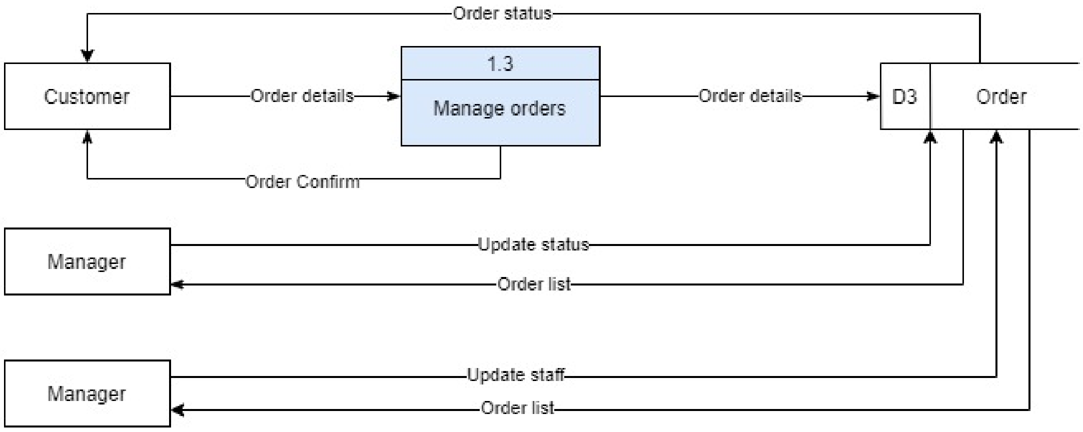


Figure 3.4: Manage order process Level 1

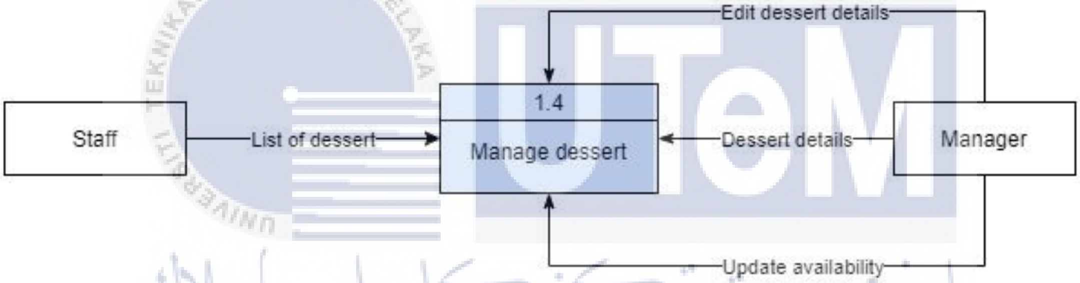


Figure 3.5: Manage dessert process Level 1

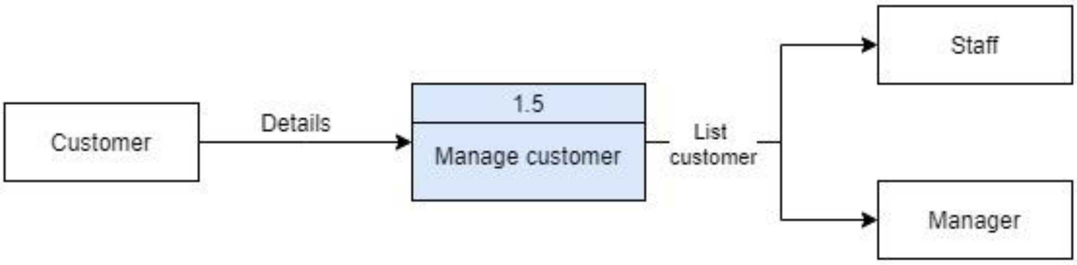
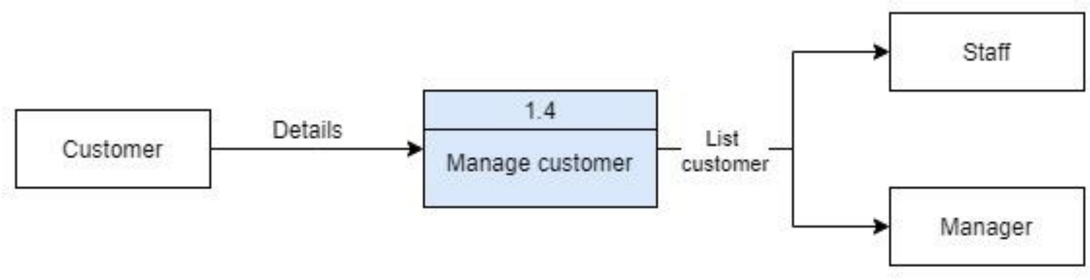


Figure 3.6: Manage customer process Level 1



3.4.5 Non-Functional Requirement

I. Availability

The system should be always available, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system run

II. Performance

The system should complete perform the ordering process in a short period of time.

III. Security

Secure all the confidential data (customer information)

IV. Safety

All the data of the order should not loss or damage

V. Maintainability

The amount of time and effort required to maintain and improve a functional software required.

3.4.6 Other Requirement

3.4.7 Software Requirement

Documentation

1. Microsoft Office 365

Microsoft Office, or simply Office, is a family of client software, server software, and services developed by Microsoft. Initially a marketing term for an office suite (bundled set of productivity applications), the first version of Office contained Microsoft Word, Microsoft Excel, and Microsoft PowerPoint.

2. Draw.io

Diagrams.net (formerly draw.io) is free online diagram software. You can use it as a flowchart maker, network diagram software, to create UML online, as an ER diagram tool, to design database schema, to build BPMN online, as a circuit diagram maker, and more. draw.io can import. vsdx, Gliffy™ and Lucidchart™ files .

Programming Language

1. Php

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely used, free, and efficient alternative to competitors such as Microsoft's ASP.

2. Html

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.

3. CSS

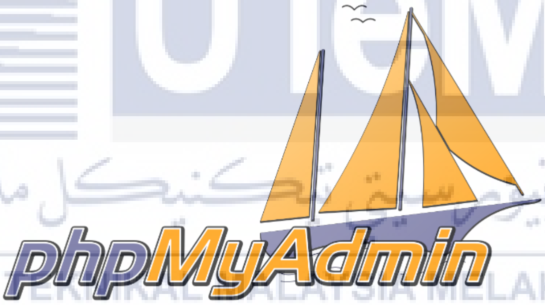
Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

DBMS

1. Mysql

MySQL is a relational database management system based on SQL – Structured Query Language. The application is **used** for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common **use** for **mySQL** however, is for the purpose of a web database

2. PhpMyAdmin



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 can directly execute any SQL statement.

Operating system

Windows 10

Microsoft Windows is a collection of graphical operating system versions that Microsoft created, promoted, and sold. The operating system utilized was Windows 10.

Server

Xampp

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

Table 3. 1 Hardware Requirements for PSM 1

COMPONENT	SPECIFICATION
Microprocessor	AMD Ryzen 7 3700U
Hard Drive	512 GB PCle SSD
Memory	8 GB DDR4
Keyboard	Chiclet keyboard, full-size keyboard, support touchpad
Display	Radeon ™ RX Vega 10 Graphics

3.5 Conclusion

During the analysis phase, the problems of the current system were examined. The project requirements, which include software, hardware, and other requires, are discussed in this chapter. By analyzing this evaluation, the structure and meaning of the data in the organization may be fully understood. All this information will be used to design the database.

The design phase of this system will be discussed in Chapter 4.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter summarizes the design phase, which is focused with the system's physical structure. Hardware, operating system, programming, design of user interfaces, design of system interfaces, and security considerations are all included in the network design or configuration. It is necessary to analyze the suggested design for performance and make sure it meets the guidelines defined during the analysis process. To put it another way, the primary goal of this phase is to turn previously specified requirements into a precise and accurate set of specifications that will be used in the future phase.

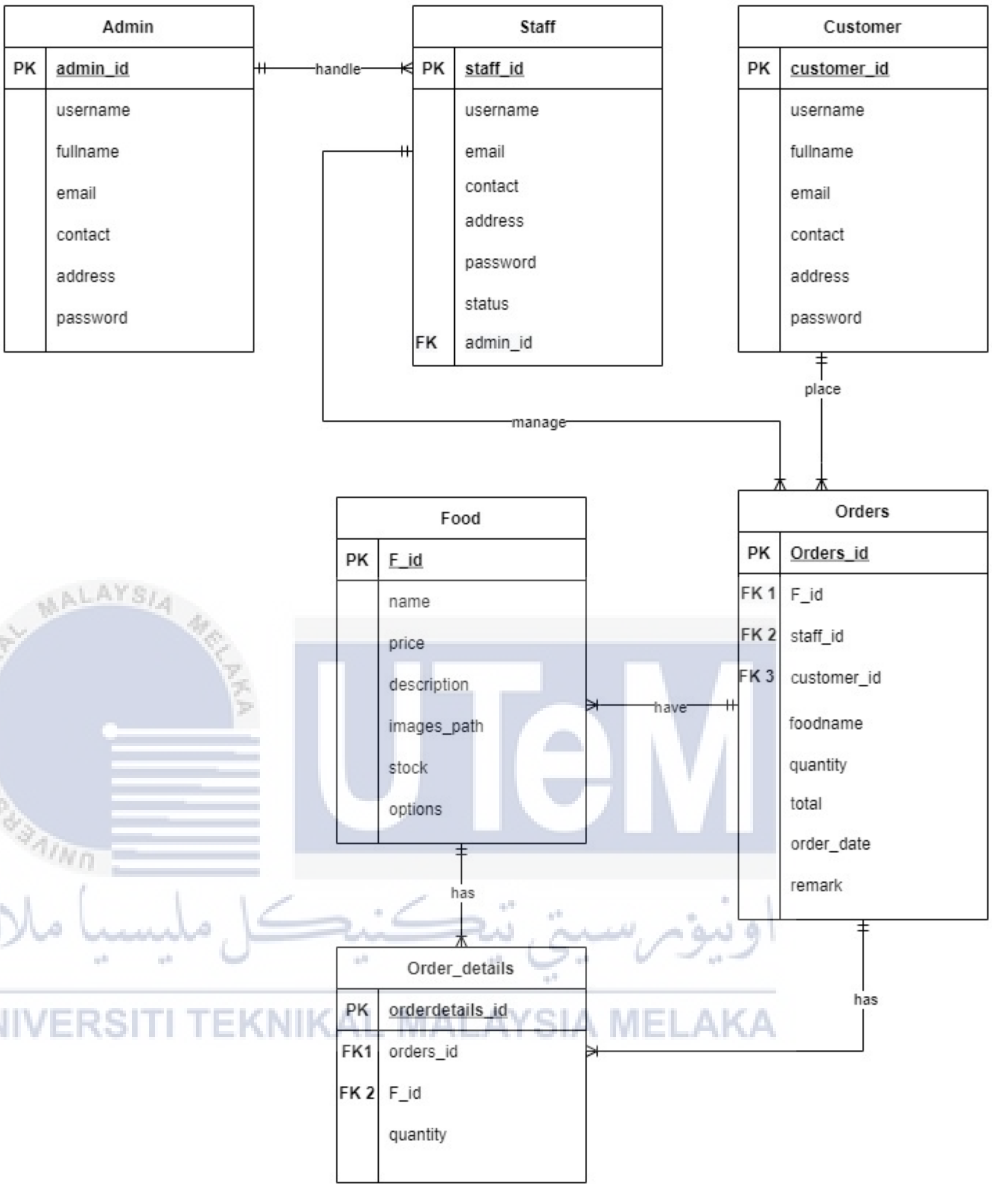
4.2 Database Design

This phase is divided into three sections: conceptual design, logical design, and physical design. The logical design step can sometimes be combined with the other two phases in some approaches. This section is not intended to be a comprehensive study of database design approaches rather, it is intended to introduce you to the subject

4.2.1 Conceptual design

Conceptual design is a stage in the design process in which the full implications of a product's purpose and form are defined. Interactions, experiences, procedures, and strategies are all part of the process.

4.2.2 Entity Relationship Diagram (ERD)



4.2.3 Business Rules

- Each manager can handle one or more staff
  Each staff can be handled by only one manager

- Each staff can manage one or more orders
  Each order can be managed by only one staff

- Each order can have one or more foods
  One food can only belong to only one orders

- Each customer can order one or multiple order
  Each order can belong to only one customer

4.2.4 Logical Design

4.2.5 Data Dictionary

CUSTOMER								
Key	Attribute	Content	Data Type and Size	Format	Default Value	Unique	Required	FK Reference Table
PK	Customer_id	Customer ID	INT (30)	999999		YES	YES	
	Username	Username	VARCHAR (30)	XXXXX			YES	
	Fullname	Customer name	VARCHAR (30)	XXXXX			YES	
	Email	Customer email	VARCHAR (30)	XXXXX			YES	
	Contact	Phone number	VARCHAR (30)	XXXXX			YES	
	Address	Customer Address	VARCHAR (50)	XXXXX			YES	
	Password	Password	VARCHAR (30)	XXXXX			YES	

FOOD								
Key	Attribute	Content	Data Type and Size	Format	Default Value	Unique	Required	FK Reference Table
PK	Food_id	Food id	INT (100)	99999		YES	YES	
	Name	Food name	VARCHAR (100)	XXXXX			YES	
	Price	Food price	INT (30)	99999			YES	

	Description	Food description	VARCHAR (500)	XXXXX			YES	
	Images_path	Image's path	VARCHAR (200)	XXXXX			YES	
	Stock	Stock of food	INT (100)	99999			YES	
	Options	Availability	VARCHAR (10)	XXXXX	ENABLE		YES	

MANAGER								
Key	Attribute	Content	Data Type and Size	Format	Default Value	Unique	Required	FK Reference Table
PK	Manager_id	Manager id	INT (30)	99999		YES	YES	
	Username	Username	VARCHAR (30)	XXXXX			YES	
	Fullname	Manager name	VARCHAR (30)	XXXXX			YES	
	Email	Manager email	VARCHAR (30)	XXXXX			YES	
	Contact	Phone number	VARCHAR (30)	XXXXX			YES	
	Address	Manager Address	VARCHAR (50)	XXXXX			YES	
	Password	Password	VARCHAR (30)	XXXXX			YES	

STAFF								
Key	Attribute	Content	Data Type and Size	Format	Default Value	Unique	Required	FK Reference Table
PK	Staff_id	Staff id	INT (30)	99999		YES	YES	
FK 1	Manager_id	Manager_id	INT (30)	99999			YES	MANAGER
FK 2	Orders_id	Order id	INT (30)	99999			YES	ORDERS
	Username	Username	VARCHAR (30)	XXXXX			YES	
	Fullname	Manager name	VARCHAR (30)	XXXXX			YES	
	Email	Manager email	VARCHAR (30)	XXXXX			YES	
	Contact	Phone number	VARCHAR (30)	XXXXX			YES	
	Address	Manager Address	VARCHAR (50)	XXXXX			YES	
	Password	Password	VARCHAR (30)	XXXXX			YES	
	Status	Status	VARCHAR (100)	XXXXX	ONLINE		YES	

ORDERS								
Key	Attribute	Content	Data Type and Size	Format	Default Value	Unique	Required	FK Reference Table
PK	Orders_id	Order id	INT (30)	99999		YES	YES	
FK 1	Food_id	Food id	INT (100)	99999			YES	FOOD
FK 2	Staff_id	Staff id	INT (30)	99999			YES	STAFF
FK 3	Customer_id	Customer id	INT (30)	99999			YES	CUSTOMER
	Foodname	Food name	VARCHAR (30)	XXXXX			YES	
	Price	Food price	VARCHAR (100)	XXXXX			YES	
	Quantity	Food quantity	INT (30)	XXXXX			YES	
	Order_date	Order date	DATE	YYYY-MM-DD			YES	
	Username	Customer username	VARCHAR (30)	XXXXX			YES	
	Remark	Status order	VARCHAR (100)	XXXXX	TO BE PROCESS		YES	

4.2.6 Physical Design

4.2.7 Data Definition Language (DDL)

i. Table Customer

```
CREATE TABLE `customer` (
  `customer_id` int(30) NOT NULL,
  `username` varchar(30) NOT NULL,
  `fullname` varchar(30) NOT NULL,
  `email` varchar(30) NOT NULL,
  `contact` varchar(30) NOT NULL,
  `address` varchar(50) NOT NULL,
  `password` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ii. Table Food

```
CREATE TABLE `food` (
  `F_ID` int(100) NOT NULL,
  `name` varchar(100) NOT NULL,
  `price` int(30) NOT NULL,
  `description` varchar(500) NOT NULL,
  `images_path` varchar(200) NOT NULL,
  `stock` int(100) NOT NULL,
  `options` varchar(10) NOT NULL DEFAULT 'ENABLE'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

iii. Table Manager

```
        CREATE TABLE `manager` (
          `manager_id` int(30) NOT NULL,
          `username` varchar(30) NOT NULL,
          `fullname` varchar(30) NOT NULL,
          `email` varchar(30) NOT NULL,
          `contact` varchar(30) NOT NULL,
          `address` varchar(50) NOT NULL,
          `password` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

iv. Table Staff

```
        CREATE TABLE `staff` (
         `staff_id` int(30) NOT NULL,
         `username` varchar(30) NOT NULL,
         `email` varchar(30) NOT NULL,
         `contact` varchar(30) NOT NULL,
         `address` varchar(50) NOT NULL,
         `password` varchar(30) NOT NULL,
         `manager_id` int(30) NOT NULL,
         `orders_ID` int(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

v. Table Orders

```
        CREATE TABLE `orders` (
         `orders_ID` int(30) NOT NULL,
         `F_ID` int(100) NOT NULL,
         `foodname` varchar(100) NOT NULL,
         `price` int(30) NOT NULL,
         `quantity` int(30) NOT NULL,
         `order_date` date DEFAULT current_timestamp(),
         `username` varchar(30) NOT NULL,
         `contact` varchar(30) NOT NULL,
         `remark` varchar(100) NOT NULL DEFAULT 'TO BE PROCESS'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

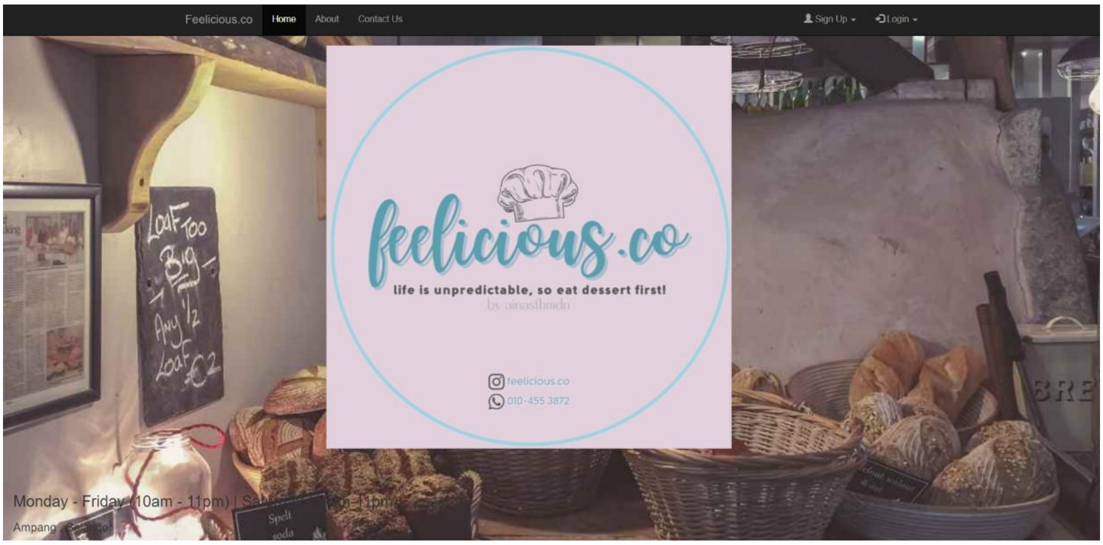
4.2.8 The Usage of Stored Procedures, Triggers and Other Related Database

Table 4.1 The Usage of Stored Procedures and Triggers

Procedure	CREATE PROCEDURE GetOrderdetails
	AS
	BEGIN
	SET NOCOUNT ON
	
	SELECT A.Orders_id , B.Foodname , A.Quantity FROM
	Orders A
	INNER JOIN Food B ON A.F_ID = B.F_ID
	
	END

Trigger	CREATE OR REPLACE TRIGGER staff_id
	
	BEFORE INSERT ON staff
	
	BEGIN
	
	DBMS_OUTPUT.PUT_LINE('New staff are about to be added');
	
	END;

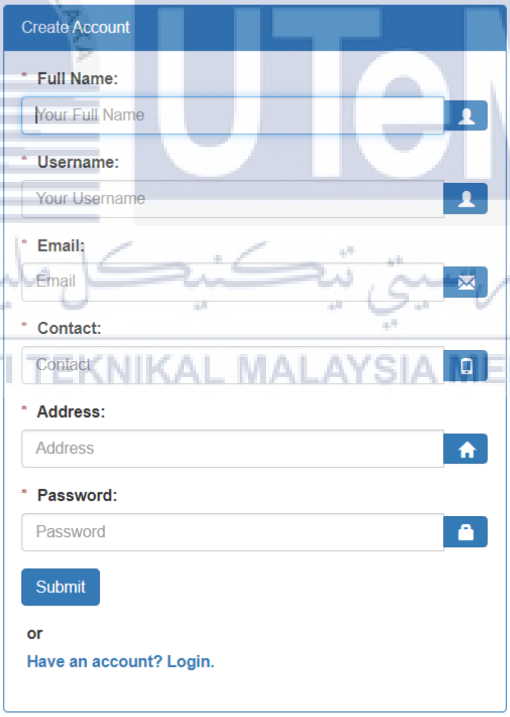
4.3 Graphical User Interface (GUI) Design



Monday - Friday (10am - 11pm) | Sat[illegible]pm - 11pm

Ampang, Selangor

Figure 4.2: EFOS Home Page Interface



Create Account

* Full Name:

Your Full Name

* Username:

Your Username

* Email:

Email

* Contact:

Contact

* Address:

Address

* Password:

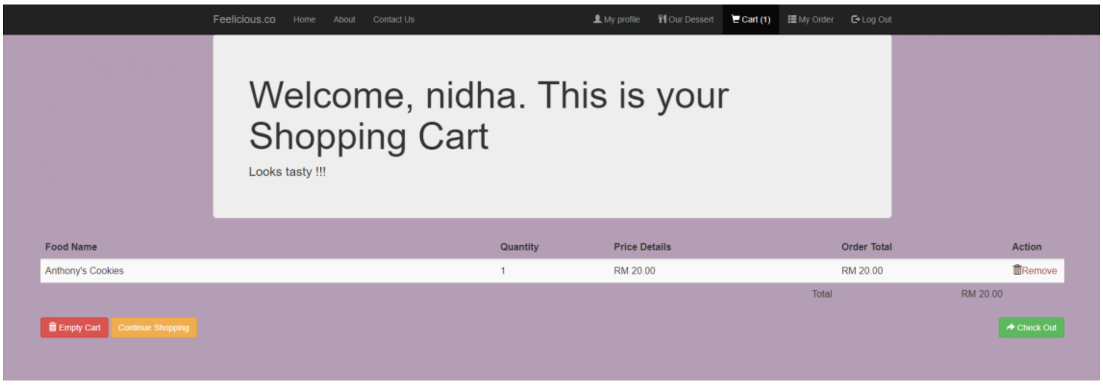
Password

Submit

or

Have an account? Login.

Figure 4.3: Sign up account interface

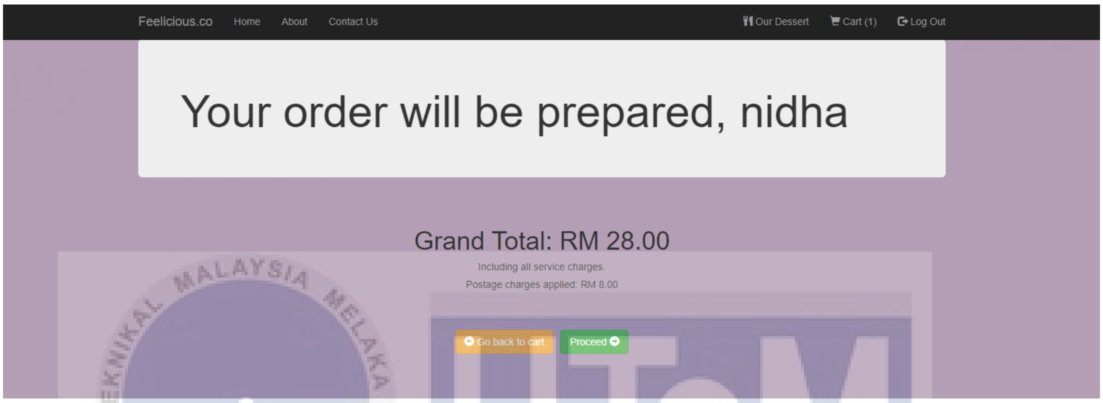


Welcome, nidha. This is your Shopping Cart

Looks tasty !!!

Food Name	Quantity	Price Details	Order Total	Action
Anthony's Cookies	1	RM 20.00	RM 20.00	Remove
			Total	RM 20.00

Figure 4.4: Customer cart interface



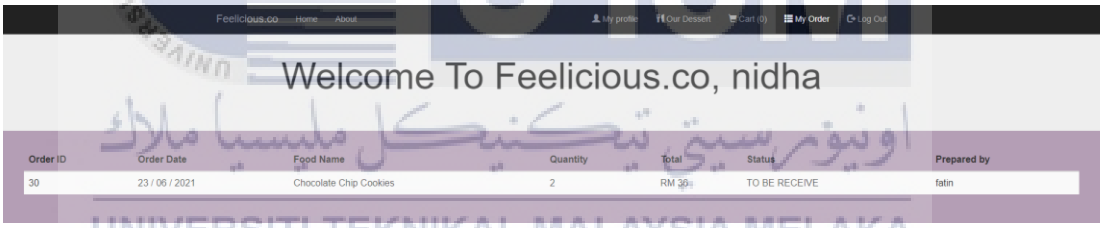
Your order will be prepared, nidha

Grand Total: RM 28.00

Including all service charges

Postage charges applied: RM 8.00

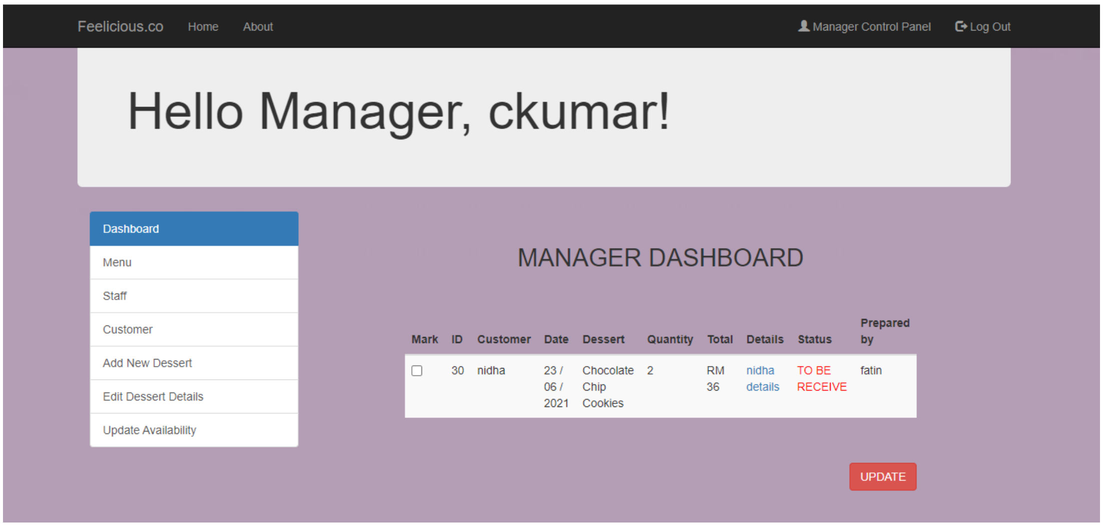
Figure 4.5: Customer order total payment interface



Welcome To Feelicious.co, nidha

Order ID	Order Date	Food Name	Quantity	Total	Status	Prepared by
30	23 / 06 / 2021	Chocolate Chip Cookies	2	RM 36	TO BE RECEIVE	fatin

Figure 4.6: Customer order status interface

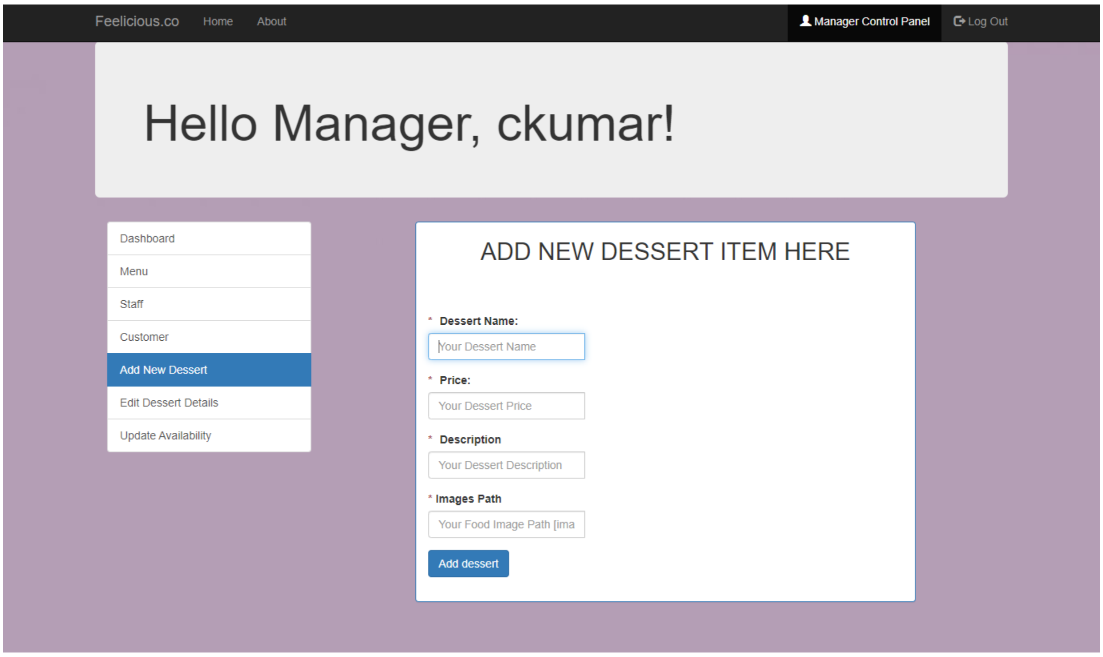


Hello Manager, ckumar!

MANAGER DASHBOARD

Mark	ID	Customer	Date	Dessert	Quantity	Total	Details	Status	Prepared by
□	30	nidha	23 / 06 / 2021	Chocolate Chip Cookies	2	RM 36	nidha details	TO BE RECEIVE	fatin

Figure 4.7: Manager dashboard interface



Hello Manager, ckumar!

ADD NEW DESSERT ITEM HERE

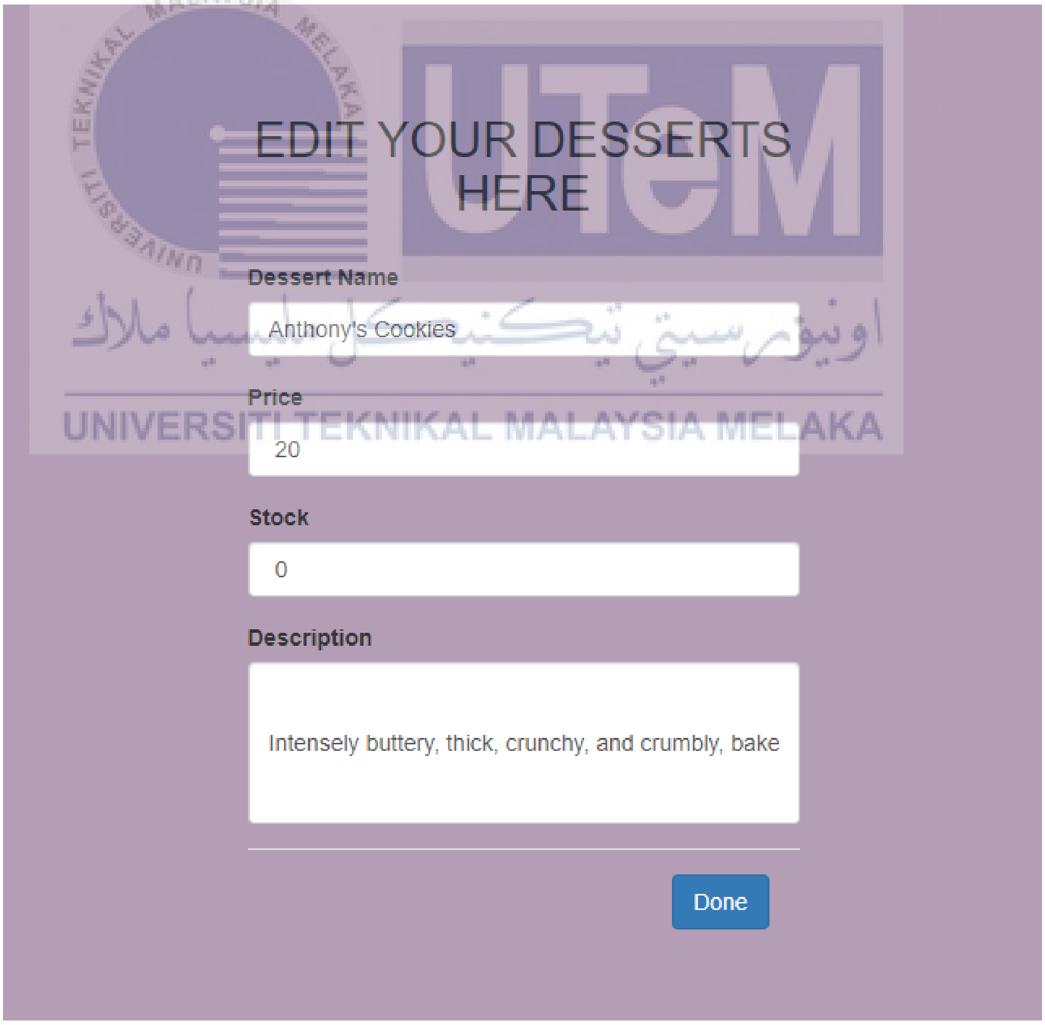
* Dessert Name: Your Dessert Name

* Price: Your Dessert Price

* Description: Your Dessert Description

* Images Path: Your Food Image Path [ima

Figure 4.8: Manager add new dessert interface



EDIT YOUR DESSERTS HERE

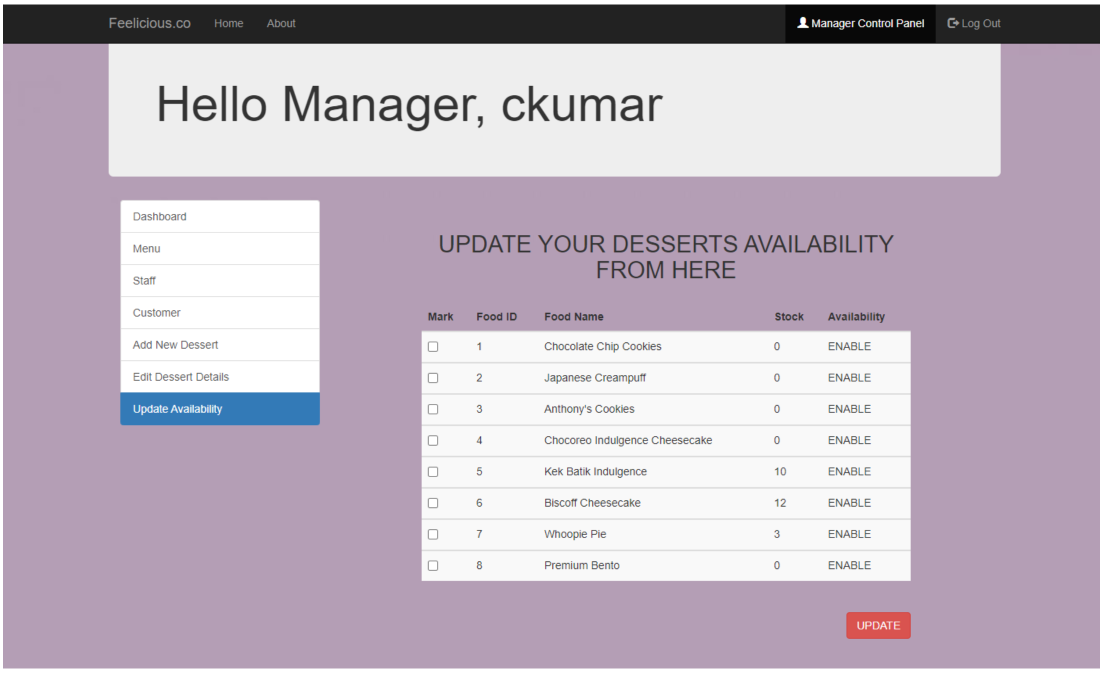
Dessert Name: Anthony's Cookies

Price: 20

Stock: 0

Description: Intensely buttery, thick, crunchy, and crumbly, bake

Figure 4.9: Manager edit dessert details interface

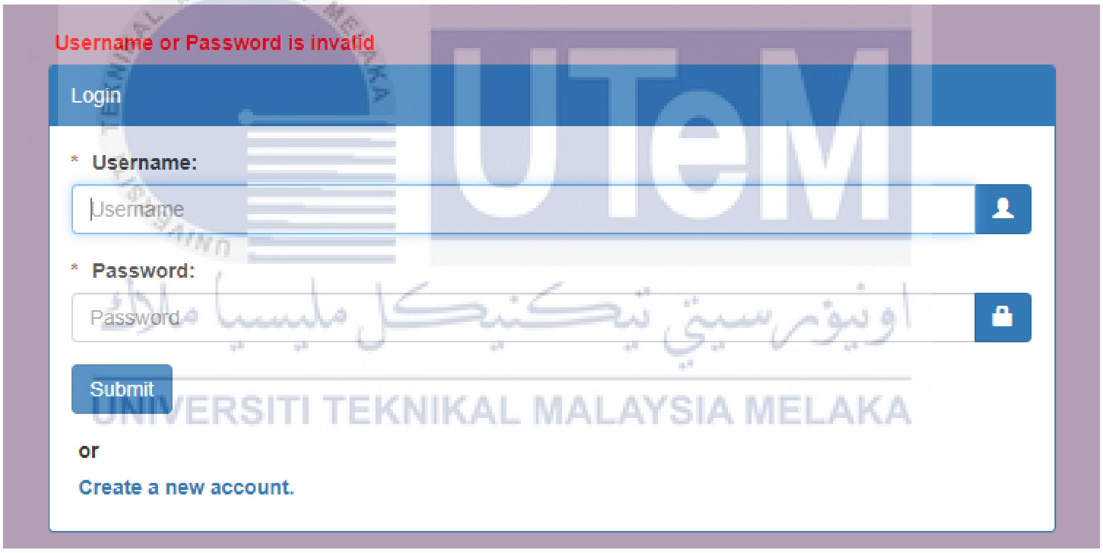


Hello Manager, ckumar

UPDATE YOUR DESSERTS AVAILABILITY FROM HERE

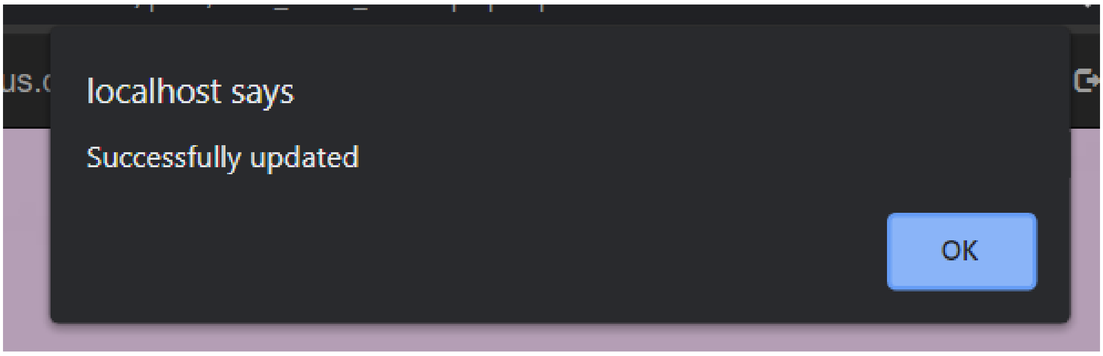
Mark	Food ID	Food Name	Stock	Availability
□	1	Chocolate Chip Cookies	0	ENABLE
□	2	Japanese Creampuff	0	ENABLE
□	3	Anthony's Cookies	0	ENABLE
□	4	Chocoreo Indulgence Cheesecake	0	ENABLE
□	5	Kek Batik Indulgence	10	ENABLE
□	6	Biscoff Cheesecake	12	ENABLE
□	7	Whoopie Pie	3	ENABLE
□	8	Premium Bento	0	ENABLE

Figure 4.10: Manager update dessert availability interface



Username or Password is invalid

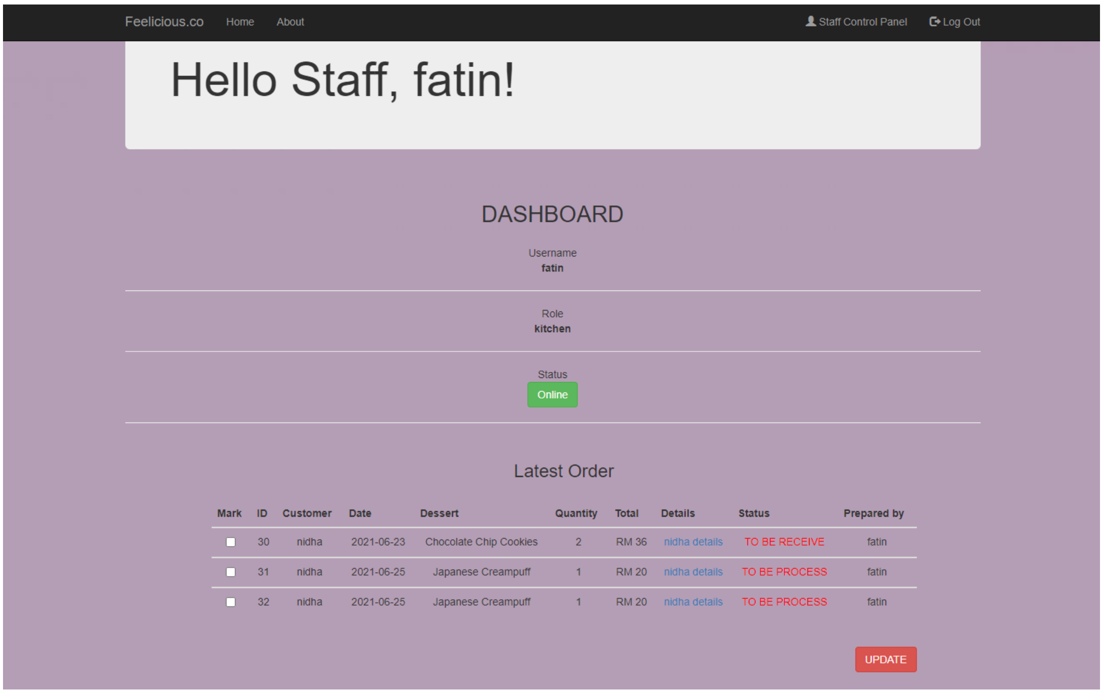
Figure 4.11: Error handling of username and password



localhost says

Successfully updated

Figure 4.12: Success message popup



Hello Staff, fatin!

DASHBOARD

Username
fatin

Role
kitchen

Status

Latest Order

Mark	ID	Customer	Date	Dessert	Quantity	Total	Details	Status	Prepared by
□	30	nidha	2021-06-23	Chocolate Chip Cookies	2	RM 36	nidha details	TO BE RECEIVE	fatin
□	31	nidha	2021-06-25	Japanese Creampuff	1	RM 20	nidha details	TO BE PROCESS	fatin
□	32	nidha	2021-06-25	Japanese Creampuff	1	RM 20	nidha details	TO BE PROCESS	fatin

Figure 4.13: Staff dashboard interface

4.4 Conclusion

Finally, designing the database is crucial because it can help solve the difficulties identified in the requirement document during the analysis process. This system's design document serves as a blueprint or blueprint for the solution, and it will be utilized for implementation, testing, and maintenance in the future. Using the logical, physical, and data dictionaries supplied in this text. Overall, the database structure is built using certain modules that will be produced in the MySQL database using phpMyAdmin. An overall picture of the conceptual design will highlight the relationship between each item that is dependent on the other.

The implementation phase of this project will be discussed in Chapter 5.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter describes how to set up a software development system and how to create a database for it. In the software development system, the techniques for installing, executing, and modifying the system and databases will be described.

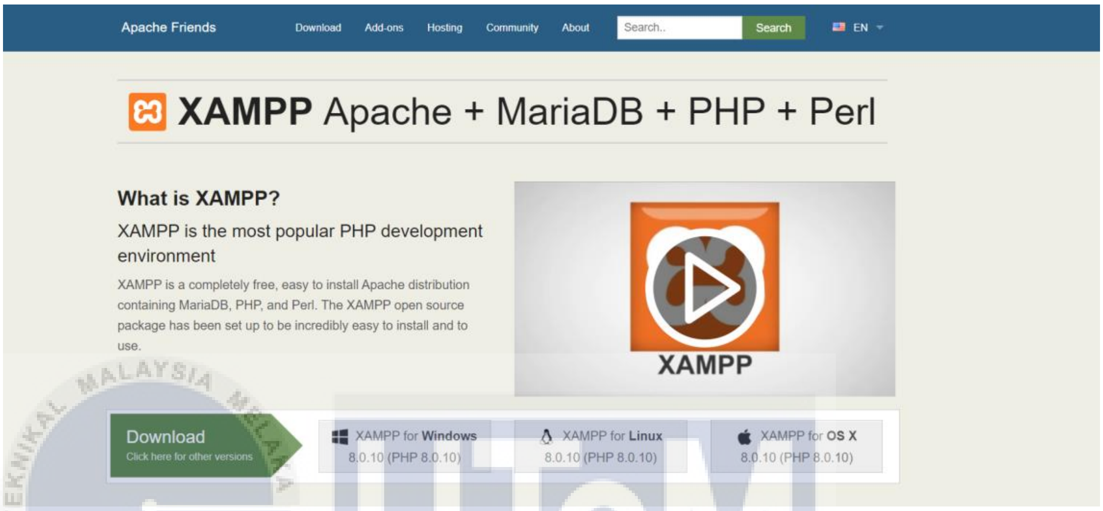
The DBMS, which is MySQL Database with phpMyAdmin and XAMPP Server, is discussed in the database implementation, as well as the Data Definition Language (DDL), Data Manipulation Language (DML), and the main processes, such as stored procedures and triggers, that are implemented towards this system using PL/SQL programming language.

5.2 Software Development Environment Setup

The software that is used for the development process of the E-Feelicious Ordering System is Sublime Text 3 as a platform to write the PHP coding. For the database, this system is using phpMyAdmin to manage all the administration of MySQL. Beside that, for the server platform, this system used XAMPP Server to be a server that will manage all the database as well as the system.

5.2.1 Software Environment Setup XAMPP Server

XAMPP

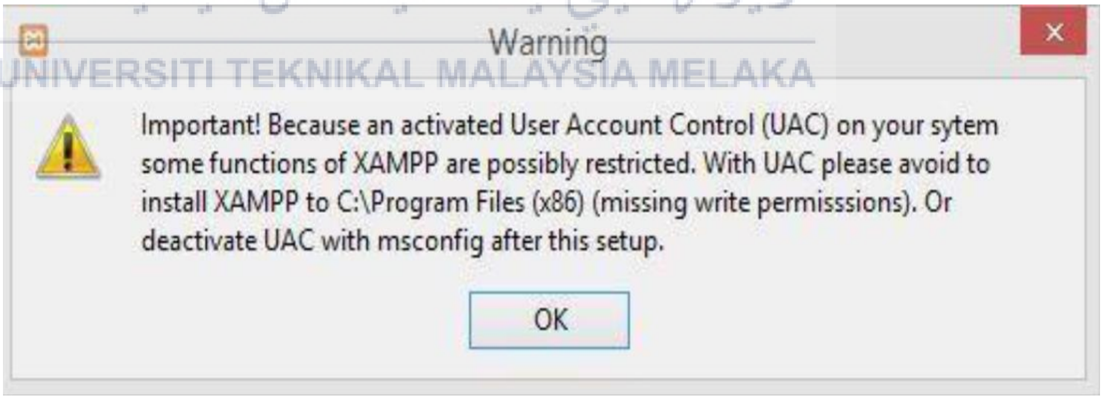


What is XAMPP?

XAMPP is the most popular PHP development environment

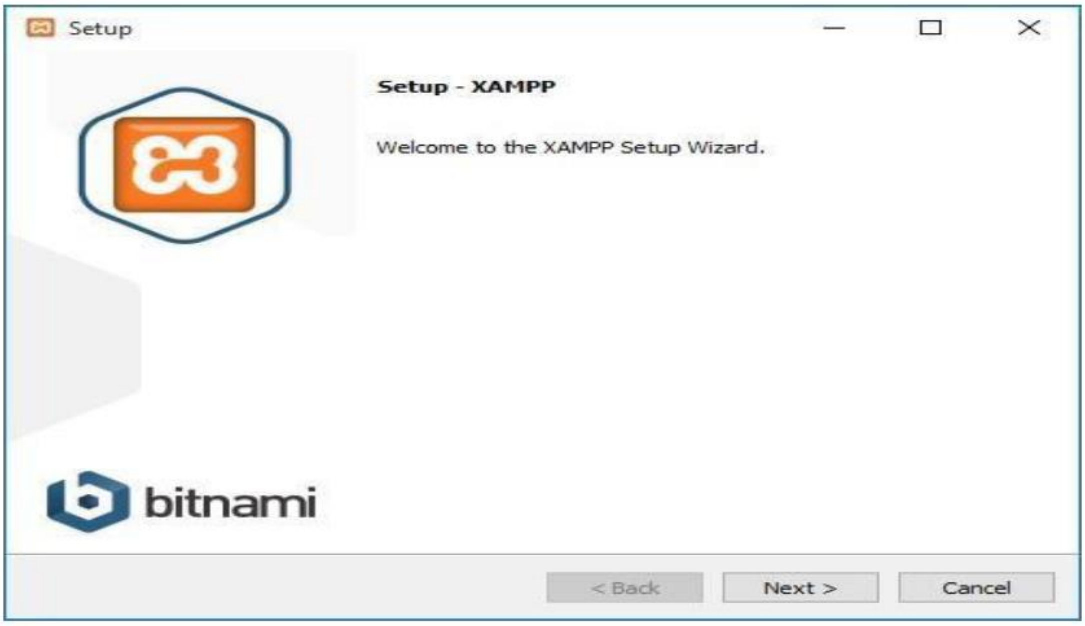
XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

Step 1: Go to https://www.apachefriends.org/index.html and start download the XAMPP Server. It has to be download based on the computer configuration.



Important! Because an activated User Account Control (UAC) on your sytem some functions of XAMPP are possibly restricted. With UAC please avoid to install XAMPP to C:\Program Files (x86) (missing write permisssions). Or deactivate UAC with msconfig after this setup.

Step 2: The pop-up message will be shown. Click ‘OK’

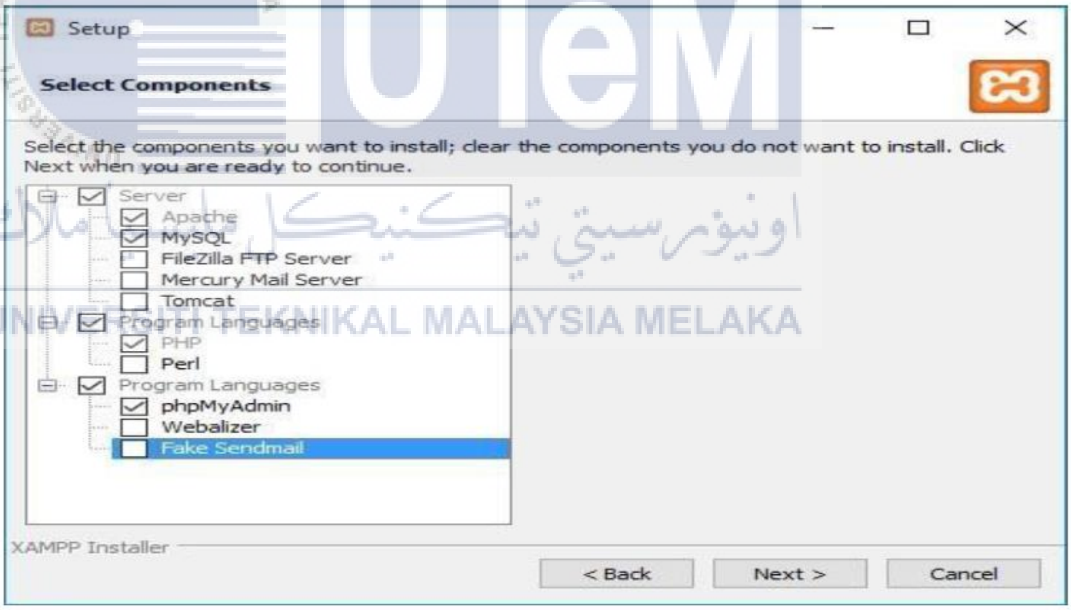


Setup

Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

Step 3: Click ‘Next >’ to continue the installation



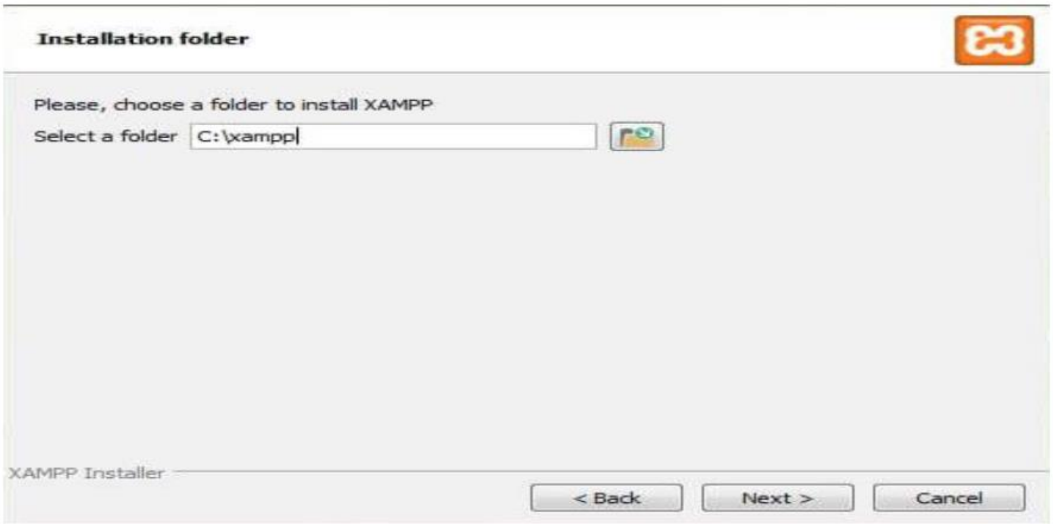
Setup

Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Server
  - ☑ Apache
  - ☑ MySQL
  - □ FileZilla FTP Server
  - □ Mercury Mail Server
  - □ Tomcat
- ☑ Program Languages
  - ☑ PHP
  - □ Perl
- ☑ Program Languages
  - ☑ phpMyAdmin
  - □ Webalizer
  - □ Fake Sendmail

Step 4: The components selection section will pop-up. This screen is to select the components that need to be install. Click ‘Next >’ to continue

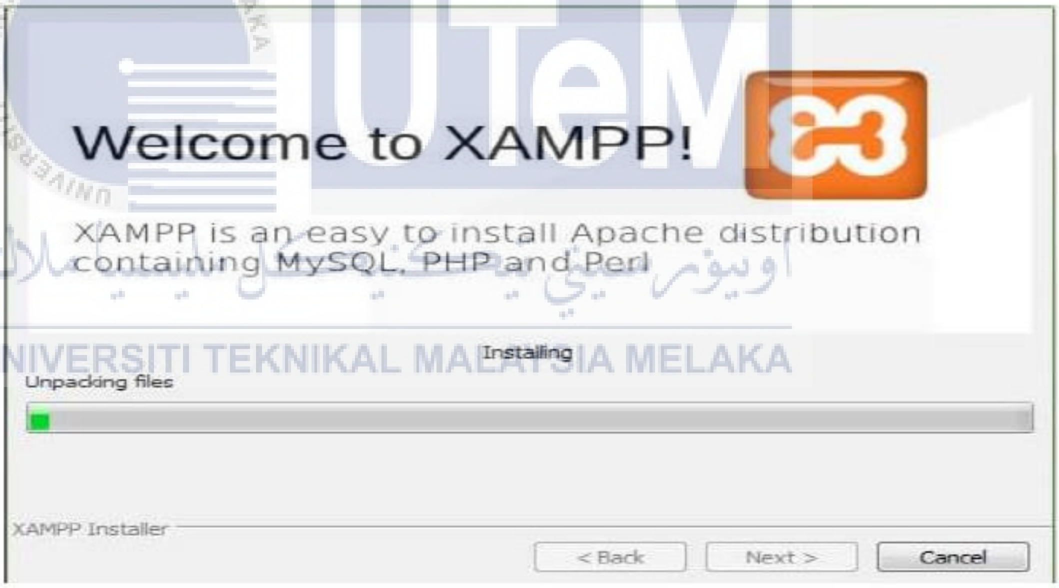


Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

Step 5: Next, choose the directory for the XAMPP server to be installed. To continue the installation, click 'Next >'



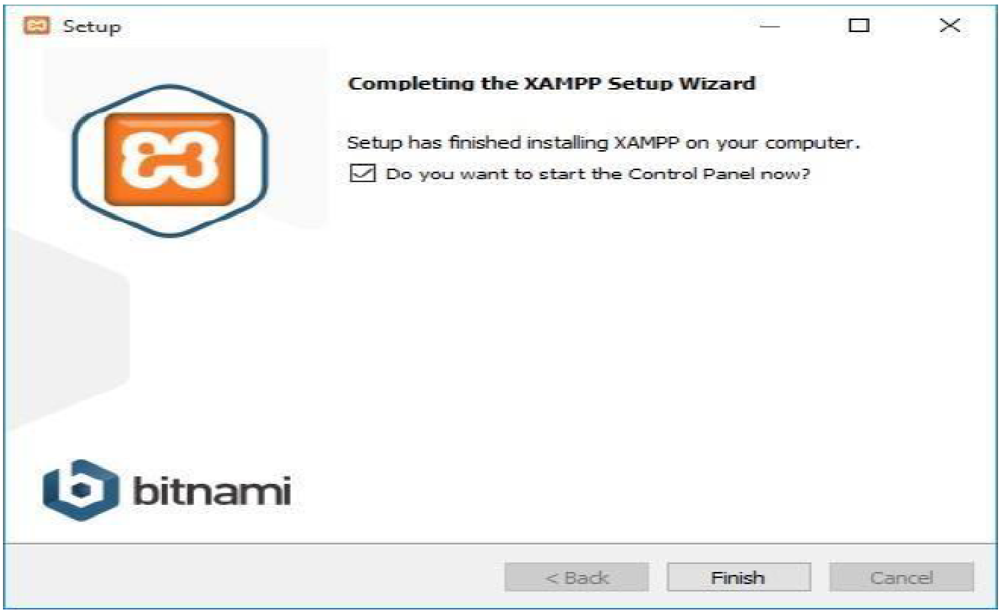
Welcome to XAMPP!

XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl

Installing

Unpacking files

Step 6: The XAMPP server will be started to install into the computer

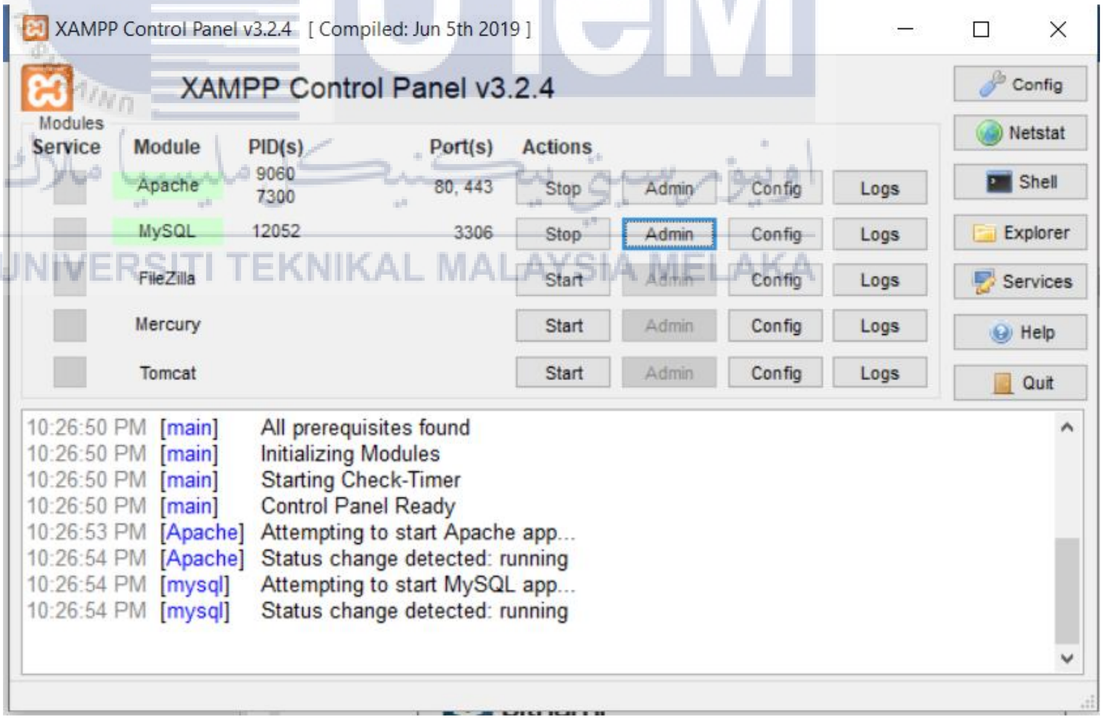


Completing the XAMPP Setup Wizard

Setup has finished installing XAMPP on your computer.

☑ Do you want to start the Control Panel now?

Step 7: After all the components are successfully installed, click ‘Finish’. Tick the box of corresponding and start the Control Panel XAMPP

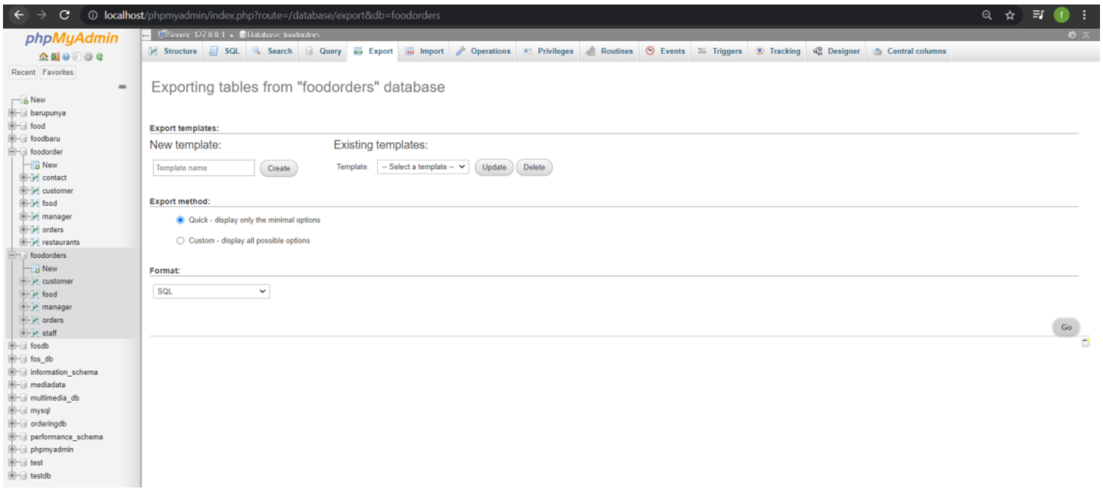

XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]
XAMPP Control Panel v3.2.4
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	9060 7300	80, 443	Stop	Admin	Config	Logs
	MySQL	12052	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
10:26:50 PM  [main]    All prerequisites found
10:26:50 PM  [main]    Initializing Modules
10:26:50 PM  [main]    Starting Check-Timer
10:26:50 PM  [main]    Control Panel Ready
10:26:53 PM  [Apache]  Attempting to start Apache app...
10:26:54 PM  [Apache]  Status change detected: running
10:26:54 PM  [mysql]   Attempting to start MySQL app...
10:26:54 PM  [mysql]   Status change detected: running
```

Step 8: The control panel of the XAMPP will be automatically opened. Modules can be started or stopped on the control panel using the actions button. The modules that have been started will be highlighted to green under the Module column.

5.2.2 Database Environment Setup



Exporting tables from "foodorders" database

Export templates:

New template: Template name Create

Existing templates: Template: -- Select a template -- Update Delete

Export method:

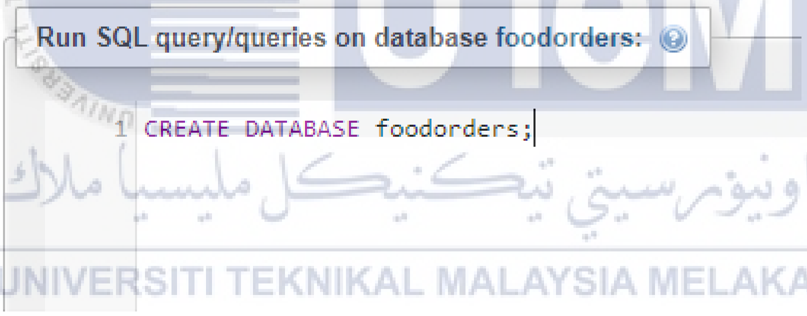
● Quick - display only the minimal options

○ Custom - display all possible options

Format: SQL

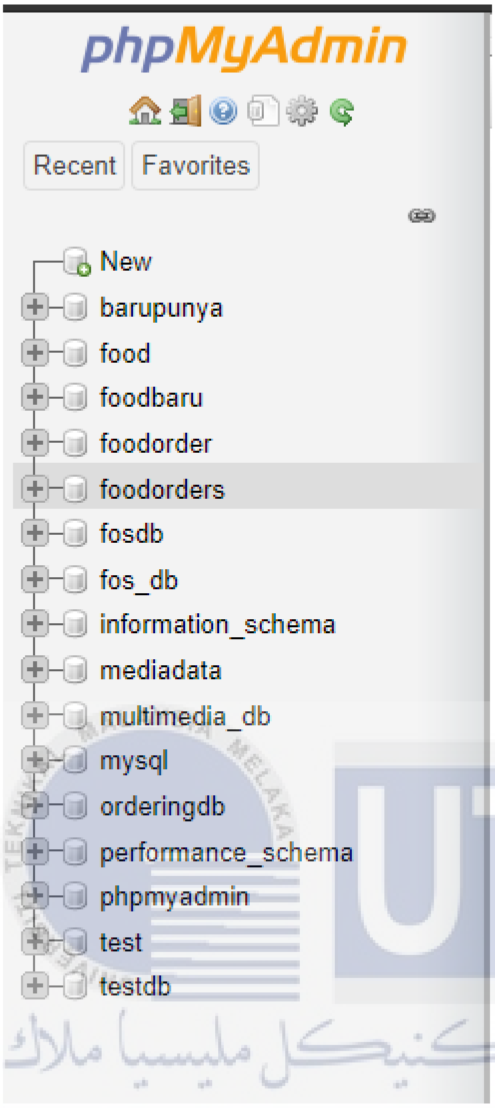
Go

Step 1: By clicking on the admin button MySQL, it will straight away go to the main page of the phpMyAdmin. Users need to create a new database. Click the 'Databases' at the top of navbar. Users also can click the 'SQL'


Run SQL query/queries on database foodorders:

```
CREATE DATABASE foodorders;
```

Step 2: Using SQL language, type the command to create a new database.



New
barupunya
food
foodbaru
foodorder
foodorders
fosdb
fos_db
information_schema
mediadata
multimedia_db
mysql
orderingdb
performance_schema
phpmyadmin
test
testdb

Step 3: New database that successfully created will be appear at the sidebar. Users can view the databases that are created. Now, users can start create tables in the database.

5.3 Database Implementation

The database implementation phase demonstrated the capability of the system by explaining the Data Definition Language (DDL), triggers, and stored procedures.

5.3.1 Data Definition Language (DDL)

Data Definition Language (DDL) is mainly used by admin to configure the database and tables during the implementation phase of the database. The DDL used in E-Feelicious Ordering System is shown below.

5.3.1.1 Create Database

```
CREATE DATABASE ‘FOODORDERS’;
```

5.3.1.2 Create Table

After the database has been built correctly, the user must use the ‘Create Table' command to add the attribute name, data type, field length, constraint, primary key, and foreign key to the table in the database. Figures 5.15 to 5.25 illustrate all of the tables that were built in PSM1 in order to develop EFOS.

```
        1. Table Customer

        CREATE TABLE `customer` (
          `customer_id` int(30) NOT NULL,
          `username` varchar(30) NOT NULL,
          `fullname` varchar(30) NOT NULL,
          `email` varchar(30) NOT NULL,
          `contact` varchar(30) NOT NULL,
          `address` varchar(50) NOT NULL,
          `password` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
2. Table Food


        CREATE TABLE `food` (
          `F_ID` int(100) NOT NULL,
          `name` varchar(100) NOT NULL,
          `price` int(30) NOT NULL,
          `description` varchar(500) NOT NULL,
          `images_path` varchar(200) NOT NULL,
          `stock` int(100) NOT NULL,
          `options` varchar(10) NOT NULL DEFAULT 'ENABLE'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

3. Table Manager

```
      CREATE TABLE `manager` (
        `manager_id` int(30) NOT NULL,
        `username` varchar(30) NOT NULL,
        `fullname` varchar(30) NOT NULL,
        `email` varchar(30) NOT NULL,
        `contact` varchar(30) NOT NULL,
        `address` varchar(50) NOT NULL,
        `password` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4. Table Orders

```
      CREATE TABLE `orders` (
        `orders_ID` int(30) NOT NULL,
        `F_ID` int(100) NOT NULL,
        `staff_id` varchar(30) NOT NULL,
        `foodname` varchar(100) NOT NULL,
        `price` int(30) NOT NULL,
        `quantity` int(30) NOT NULL,
        `order_date` date DEFAULT current_timestamp(),
        `username` varchar(30) NOT NULL,
        `remark` varchar(100) NOT NULL DEFAULT 'TO BE PROCESS'
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
5. Table Staff

        CREATE TABLE `staff` (
          `staff_id` int(30) NOT NULL,
          `username` varchar(30) NOT NULL,
          `email` varchar(30) NOT NULL,
          `contact` varchar(30) NOT NULL,
          `address` varchar(50) NOT NULL,
          `password` varchar(30) NOT NULL,
          `manager_id` int(30) NOT NULL,
          `status` varchar(30) NOT NULL,
          `role` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

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 staff id before the primary key in the Staff table. Figure 5.1 shows triggers list implemented in EFOS

Table 5.7: List of Triggers in EFOS

Types of triggers	Table	Use of Trigger
BEFORE INSERT	Staff	To create staff id every time before the new staff added

Table 5.8: Create Trigger Insert New Staff

```
CREATE OR REPLACE TRIGGER staff_id
        BEFORE INSERT ON staff
        BEGIN
           DBMS_OUTPUT.PUT_LINE ('New staff are about to be added');
END;
```

5.3.1.4 Create Stored 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 Procedure in EFOS

Types of procedures	Table	Use of Procedure
Stored procedure (DML) inserts, update and delete.	Orders	View order details

Table 5.10: Create Stored Procedure View Order details

```
CREATE PROCEDURE GetOrderdetails
        AS
        BEGIN
        SET NOCOUNT ON

        SELECT A.Orders_id , B.Foodname , A.Quantity FROM
        Orders A
        INNER JOIN Food B ON A.F_ID = B.F_ID

END
```

5.4 Conclusion

Finally, before moving on to the testing step, the implementation phase must be completed. The E-Feelicious Ordering System was created with the XAMPP Server and MySQL, and the database was managed by using phpMyAdmin. This project's database contains 5 tables, triggers, and stored procedures, as well as data storage.

The testing phase of this project will be discussed in Chapter 6.

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 E-Feelicious Ordering System. This phase is for make sure the system developed have all the requirements. The testing that has been done on Pahang Pharmacy Gate Pass Management 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 E-Feelicious Ordering 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 E-Feelicious Ordering 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 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 E-Feelicious Ordering 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 EFOS

Environment Configuration	Specification
Operating System	Windows 10
Processor	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
Random Access Memory (RAM)	8.00 GB (6.94 GB usable)
Database	MySQL
Server	Apache
Programming Language	Hypertext Processor (PHP) Hypertext Markup Language (HTML)

6.2.2.2 Software Application

Software applications consist of all the application or module inside E-Feelicious Ordering System. Table 6.3 shows all the application in EFOS.

Table 6.3: EFOS Application Environment

System Application	1. System login (user authentication) 2. Add, update, delete and view in each module for staff and customer 3. Display order summary 4. Menu management

6.2.2.3 System Software

System software consists of all the tools that have been used in E-Feelicious Ordering 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 2. XAMPP Server (Apache and MySQL) 3. SQL Developer (MySQL) 4. Sublime Text 3 5. Google Chrome (Browser) 6. Microsoft Word

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 E-Feelicious Management System.

Table 6.5: System Hardware Tools

System Hardware	1. Laptop 2. Mouse 3. Keyboard

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 E-Feelicious Ordering 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 EFOS Testing

Module/Component	Activity	Duration (Day)
Registration	Error handling test and integration test	1
Login	Error handling test and integration test	1
Password	Error handling test and integration test	1
Update Profile	Error handling test and integration test	2
Customer	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 Test Strategy

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 behaviors) 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

This section explains each test module by describing its test case description, test type and expected results

6.4.1.1 Test Module 1: User registration (Sign Up)

Table 6.8: Test Description of User Registration Module

Test Case ID	Description	Testing type	Expected result
EFOS_01-1	Field is blank for user name, password and confirm password	Error Handling Test	"Please fill out this field" message will be popped up
EFOS_01-2	Valid input for each field	Unit testing / Integration	New user's information is successfully recorded into the database

6.4.1.2 Test Module 2: Login (User Authentication)

Table 6.9: Test Description of Login Module

Test Case ID	Description	Testing type	Expected result
EFOS_02-1	Invalid username and password	Error Handling Test	"Username or Password may wrong" message will be popped up
EFOS_01-2	Valid input for each field	Unit testing / Integration	User can log into the system Successfully.

6.4.1.3 Test Module 3: Update Profile Module

Table 6.10: Test Description of Update Profile Module

Test Case ID	Description	Testing type	Expected result
EFOS_03-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
EFOS_03-2	Valid input for each field	Unit testing / Integration	"Successfully updated" will pop up

6.4.1.4 Test Module 4: Add New Menu Module

Table 6.11: Test Description of Add New Menu Module

Test Case ID	Description	Testing type	Expected result
EFOS_04-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
EFOS_04-2	Valid input for each field	Unit testing / Integration	"Successfully added" will pop up

6.4.1.5 Test Module 5: Order Module

Table 6.12: Test Description of Order Module

Test Case ID	Description	Testing type	Expected result
EFOS_05-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
EFOS_05-2	Valid input for each field	Unit testing / Integration	"Successfully placed order" will pop up

6.4.1.6 Test Module 6: Update Menu Module

Table 6.13: Test Description of Update Menu Module

Test Case ID	Description	Testing type	Expected result
EFOS_06-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
EFOS_06-2	Valid input for each field	Unit testing / Integration	"Successfully updated" will pop up

6.4.1.7 Test Module 7: Update Order Module

Table 6.14: Test Description of Update Order Status Module

Test Case ID	Description	Testing type	Expected result
EFOS_07-1	Some field is blank	Error Handling Test	"Please fill out this field" will pop up
EFOS_07-2	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 6.15: Test Data for User Registration Module

Test Data ID	TD_01-1	TD_01-2
Test Case ID	EFOS_01-1	EFOS_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 6.16: Test Data for Login Module

Test Data ID	TD_02-1	TD_02-2
Test Case ID	EFOS_02-1	EFOS_02-2
Username	Ahmad	Ahmad
Password	abc	Abc_123
Test Data Result	Login fails because of incorrect password	Successfully login to EFOS

Table 6.17: Test Data for Update Profile Module

Test Data ID	TD_03-1	TD_03-2
Test Case ID	EFOS_03-1	EFOS_03-2
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 6.18: Test Data for Add New Menu Module

Test Data ID	TD_04-1	TD_04-2
Test Case ID	EFOS_04-1	EFOS_04-2
Insert New Menu Form	User does not fill up the fields correctly Food Name: Cake Price:	Users fill up the fields completely Food Name: Cake Price: RM30
Test Data Result	Insert new menu fails due to incomplete form	Insert new menu is successfully saved and submitted

Table 6.19: Test Data for Order Module

Test Data ID	TD_05-1	TD_05-2
Test Case ID	EFOS_05-1	EFOS_05-2
Place Order Form	User does not fill up the fields correctly Username: Ahmad Order: Cake Quantity:	Users fill up the fields completely Username: Ahmad Order: Cake Quantity: 5
Test Data Result	Place order fails due to incomplete form	Order placed successfully and submitted

Table 6.20: Test Data for Update Menu Module

Test Data ID	TD_06-1	TD_06-2
Test Case ID	EFOS_06-1	EFOS_06-2
Update Menu Form	User does not fill up the fields correctly Food Name: Cake Price:	Users fill up the fields completely Food Name: Cake Price: RM45
Test Data Result	Update menu fails due to incomplete form	Menu updated successfully saved and submitted

Table 6.20: Test Data for Update Order Status Module

Test Data ID	TD_06-1	TD_06-2
Test Case ID	EFOS_06-1	EFOS_06-2
Update Order Status	User does not fill up the fields correctly Username: Ahmad Order: Cake Quantity: Status:	Users fill up the fields completely Username: Ahmad Order: Cake Quantity: 5 Status: To be delivered
Test Data Result	Update order fails due to incomplete form	Order status updated successfully saved and submitted

6.5 Test Results 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 6.21: Registration Module Test Result

Module/Component			Result		
Test Case ID	**Test Data ID**	**Testing date**	**Description**	**Success**	**Fail**
EFOS_01-1	TD_01-1	31/08/2021	Some field blank	/	
EFOS_01-2	TD_01-2	31/08/2021	All fields were filled	/	

Table 6.22: Login Module Test Result

Module/Component			Result		
Test Case ID	**Test Data ID**	**Testing date**	**Description**	**Success**	**Fail**
EFOS_02-1	TD_02-1	01/09/2021	User key in incorrect username or password	/	
EFOS_02-2	TD_02-2	01/092021	User key in correct username and password.	/	

Table 6.23: Update Profile Module Test Result

Module/Component			Result		
Test Case ID	**Test Data ID**	**Testing date**	**Description**	**Success**	**Fail**
EFOS_03-1	TD_03-1	01/092021	Some field blank	/	
EFOS_03-2	TD_03-2	01/092021	All fields were filled	/	

Table 6.24: Add New Menu Module Test Result

Module/Component			Result		
Test Case ID	**Test Data ID**	**Testing date**	**Description**	**Success**	**Fail**
EFOS_04-1	TD_04-1	02/092021	Some field blank	/	
EFOS_04-2	TD_04-2	02/092021	All fields were filled	/	

Table 6.25: Order Module Test Result

Module/Component			Result		
Test Case ID	**Test Data ID**	**Testing date**	**Description**	**Success**	**Fail**
EFOS_05-1	TD_05-1	02/092021	Some field blank	/	
EFOS_05-2	TD_05-2	02/092021	All fields were filled	/	

Table 6.26: Update Menu Module Test Result

Module/Component			Result		
Test Case ID	**Test Data ID**	**Testing date**	**Description**	**Success**	**Fail**
EFOS_06-1	TD_06-1	03/092021	Some field blank	/	
EFOS_06-2	TD_06-2	03/092021	All fields were filled	/	

Table 6.27: Update Order Status Module Test Result

Module/Component			Result		
Test Case ID	Test Data ID	Testing date	Description	Success	Fail
EFOS_07-1	TD_07-1	03/092021	Some field blank	/	
EFOS_07-2	TD_07-2	03/092021	All fields were filled	/	

6.6 Conclusion

Software testing is necessary because we make mistakes. Some of these errors are not very important, but some of them are dangerous. We need to check everything we produce because things always go wrong. Humans always make mistakes.

Software testing is necessary because we make mistakes. Some of these errors are not very important, but some of them are dangerous. We need to check everything we produce because things always go wrong. Humans always make mistakes.

To conclude, testing stage clarified around the procedure that has been utilized to approve and confirm EFOS 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

This chapter will explain the system's weaknesses and strengths to bring the project to a completion. This chapter will also discuss EFOS improvement's suggestions, which are any suggestions that can improvise the user's experience with this system. All of the strengths and weaknesses are focused on a module that was built within it.

Finally, this chapter discusses project contributions, such as those made to the university, faculty, company, or individual.

7.2 Observation on Weakness and Strengths

Every system that has developed has strengths and weaknesses. The following is a list of E-Feelicious Ordering System's strengths and weaknesses:

Strengths

- All data about customers and orders are kept in the appropriate database compared to manual systems that are no longer suitable for the information technology era.
- Allows customers to place an order at virtually any time, from anywhere, saving the time and resources typically spent on travelling to pick up a meal.
- The system has different levels of user access, that are admin, staff, and customer. This is to split and control user access to the system. Each user has different access rights when they log in to the system.

Weakness

- Customers cannot cancel their own orders
- The system's shopping cart only offers basic functions, and it does not allow to modify the order
- Does not have backup and recovery procedure in the database

7.3 Propositions for Improvement

After analyzing the strength and weaknesses, there are several ideas for improvements that can help in future use. E-Feelicious Ordering 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 ordering system to have a secured payment system with a few options of payment methods i.e., via online banking, cash on delivery or credit/debit card. After the transaction, the receipt will be print out. So, this way also can improve for the staff to organized sales report.

Next, to help improve customer experience, this system can be upgraded to let customer choose layout to view the menu. This will make it easier for customers to choose the menu display according to their convenience.

7.4 Project Contribution

The purpose of this E-Feelicious Ordering System is to help the online business by minimizing the workloads. This ordering system eliminates many of the issues that plague from the previous standard ordering systems. The initial goal of this system is to create a system which includes dessert preparation time. Customers may view the status of the order for the desserts to be delivered to them under this system.

By using the method of taking the previous order, it takes quite a long time. Orders taken are easily lost or damaged. Customers may be more satisfied because of this. This system is designed to offer customers or users with the most up-to-date information. The menu can be changed by the admin depending on the availability of desserts. It is also necessary to have a user-friendly system because it will affect the business's brand image. This ordering system's user interface is simple and straightforward. The dashboard for the crew and the chef is similarly compact and organized.

I would like to express my gratitude to my supervisor, Dr Yahaya Abd Rahim because I would not be able to achieve what I have been achieving in this project without her help and support. Thank you

7.5 Conclusion

People nowadays enjoy a great deal of ease due to the advances in technology. Many businesses use a system to improvise their operations since they are efficient for both sellers and customers. Customer experience will be improved by making this order process much easier than previous method. This E-Feelicious Ordering System will secure the order’s information from being lost or misplaced.

The previous method is a bit difficult because the customer does not know whether to stock for the desserts still available or not. The staffs must recall the most up-to-date stock so that they may inform the customers. Customers may be disappointed with the services if the staff forgets to tell them about the unavailability. This system allows staff to update the menu.

By replacing the inefficient process of taking orders, this system will increase the productivity of the business. Therefore, this system contributes to increasing the business’s quality and performance. It reduces the staff's manual labor. Customers can place orders through the system without any fault.

REFERENCES

Khairunnisa, K., Ayob J., Mohd. Helmy A. Wahab, M. Erdi Ayob, M. Izwan Ayob & M. Afif Ayob 2009, ‘The application of wireless food ordering system’, MASAUM Journal of Computing, vol. 1, no. 2.