PIZZA ORDERING MANAGEMENT SYSTEM (POMS)

SITI AISYAH BINTI MUHAMAD NAZAR

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

PIZZA ORDERING MANAGEMENT SYSTEM (POMS)

SITI AISYAH BINTI MUHAMAD NAZAR

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

2023/2024

DECLARATION

I hereby declare that this project report entitled

PIZZA ORDERING MANAGEMENT SYSTEM (POMS)

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

STUDENT : [illegible] Date : 25/9/2024
(SITI AISYAH BINTI MUHAMAD NAZAR)

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 : [illegible] Date : 25/9/2024
(PROF. DR NURUL AKMAR BINTI EMRAN)

DECLARATION

DEDICATION

I would like to express my special dedication to my beloved parents and supervisor who have been giving me guidance and encouragement throughout my project. Especially, please allow me to dedicate my greatest gratitude to the following significant advisors and contributors.

ACKNOWLEDGEMENTS

I express my gratitude to Dr. Nurul Akmar binti Emran for providing assistance in the effective completion of this research. I am so appreciative of her knowledge, counsel, support, and encouragement. She has consistently and persuasively portrayed the spirit of adventure in research and has provided us with insightful direction and heartfelt support.

I also want to express my gratitude to my buddy and parents, who have been there for me throughout this process, providing encouragement and support. I would especially like to express my sincere gratitude to the following important mentors and collaborators.

Finally, but just as importantly, I would want to express my gratitude to my professor, my buddy, and the Faculty of Information and Communications Technology at Universiti Teknikal Malaysia Melaka for all of their help and advice with this report. This report might have remained a faraway reality if it weren't for their assistance.

ABSTRACT

Pizza Ordering Management System (POMS) is a web-based system designed to streamline the process of ordering and managing pizza orders. In the current system, all data collected during the ordering process are stored in a paper format, which is time-consuming for the staff as they have to manually take orders, prepare pizzas, and manage payments and deliveries. Additionally, manual data handling increases the risk of errors, such as incorrect orders and lost data. Therefore, POMS is developed to increase the efficiency of the ordering process by replacing the existing manual paper-based system. Customers can place their orders through the company website, fill in their personal details, and choose their desired pizzas. At the same time, the system minimizes human errors by automatically processing orders and calculating totals, and it reduces data redundancy by enforcing key constraints in the database. This system allows staff to efficiently manage order details, prepare pizzas, handle payments, and organize deliveries. Furthermore, POMS utilizes the Database Life Cycle (DBLC) methodology for development due to its structured approach, which includes specific deliverables and review processes for each phase. DBLC consists of six important phases: database initial study, database design, implementation & loading, training & evaluation, operation, and maintenance & evaluation. By following these phases, the Pizza Ordering Management System ensures a systematic and efficient management of the ordering process, enhancing the overall customer experience and operational effectiveness of the restaurant.

ABSTRAK

[Sistem Pengurusan Pesanan Pizza (POMS) ialah sistem berasaskan web yang direka untuk menyederhanakan proses memesan dan menguruskan pesanan pizza. Dalam sistem semasa, semua data yang dikumpulkan semasa proses pesanan disimpan dalam format kertas, yang memakan masa bagi kakitangan kerana mereka perlu mengambil pesanan secara manual, menyediakan pizza, dan menguruskan pembayaran dan penghantaran. Selain itu, pemprosesan data manual meningkatkan risiko kesilapan, seperti pesanan yang salah dan data yang hilang. Oleh itu, POMS telah dibangunkan untuk meningkatkan kecekapan proses pesanan dengan menggantikan sistem berasaskan kertas manual yang sedia ada. Pelanggan boleh meletakkan pesanan mereka melalui laman web syarikat, mengisi butiran peribadi mereka, dan memilih pizza yang mereka inginkan. Pada masa yang sama, sistem ini mengurangkan kesilapan manusia dengan memproses pesanan secara automatik dan mengira jumlah, dan ia mengurangkan redundansi data dengan menguatkuasakan sekatan utama dalam pangkalan data. Sistem ini membolehkan kakitangan untuk menguruskan butiran pesanan dengan cekap, menyediakan pizza, menangani pembayaran, dan mengatur penghantaran. Selain itu, POMS menggunakan kaedah kitaran hayat pangkalan data (DBLC) untuk pembangunan kerana pendekatan terstrukturnya, yang termasuk penghantaran dan proses tinjauan tertentu untuk setiap fasa. DBLC terdiri daripada enam fasa penting: kajian permulaan, reka bentuk pangkalan data, pelaksanaan & muat turun, latihan & penilaian, operasi, dan pemeliharaan & evaluasi. Dengan mengikuti fasa ini, Sistem Pengurusan Pesanan Pizza memastikan pengurusan proses pesanan yang sistematik dan cekap, meningkatkan pengalaman pelanggan secara keseluruhan dan kecekapan operasi restoran.

TABLE OF CONTENTS

LIST OF TABLES

PAGE

LIST OF FIGURES

LIST OF ABBREVIATIONS

FYP	-	Final Year Project
POMS	-	Pizza Ordering Management System
DBLC	-	Database Life Cycle
DFD	-	Data Flow Diagram
GUI	-	Graphical User Interface
DDL	-	Data Definition Language
DML	-	Data Manipulation Language
ERD	-	Entity Relationship Diagram
NF	-	Normal Form
CPU	-	Central Processing Unit
OS	-	Operating System
IDE	-	Integrated Development Environment

LIST OF ATTACHMENTS

CHAPTER 1: INTRODUCTION

1.1 Project Background

A pizzeria specialized in custom made pizzas is currently talking orders by phone. The current system where the customer calls the pizzeria takes times of staff to answer the phone or via through counter and is more work consuming than necessary. They want to allow customers to customize and order their pizzas online. The pizzeria also aims to increase the sales, due to the easy to use order online website. The system will give the staff more time to 'work' rather than to accept orders by phone or via through counter, also the potential increase in customers are enough reason for the pizzeria to accept the change information system where customers can order their customized pizza that they wanted.

1.2 Problem Statement

i. **Difficulty in managing orders due to manual recording:**

It's challenging to manage pizza orders manually, leading to potential errors and inefficiencies in the process. There's a risk of losing order information, and the manual system requires excessive paper usage.

ii. **Human error in order processing and assignment:**

Errors may occur in processing pizza orders, such as incorrect toppings, sizes, or delivery addresses. Mistakes in order assignment to the wrong delivery person or kitchen staff can lead to delays and customer dissatisfaction.

iii. Difficulty in searching and generating reports:

Locating specific orders or customer details within a paper-based system is time-consuming and prone to oversight. Generating summary reports on order statistics, popular toppings, or sales trends is challenging due to the manual nature of record-keeping.

1.3 Objectives

Here are some of the objectives that need to achieve when creating this system:

1) **To ease pizza ordering and provide searching by having computerize order pizza online system and to improve the efficiency of the ordering system.**
   Enable customers to search for and select pizza options easily through the online platform which is to simplify the process for customers and optimize order processing times to ensure quick and accurate fulfilment of customer requests by reducing manual steps and streamlining the order placement process.

2) **To computerized record payment**
   Enable automatic generation of payment receipts for customers' pizza order records to enhance accuracy and transparency and integrate secure computerized payment processing methods to facilitate seamless transactions for customers and ensure compliance with payment industry standards and regulations for data security and privacy.

3) **To implement an integrated digital record-keeping and reporting system**
   Streamline locating orders and customer details, and to simplify generating reports on order statistics, popular toppings, and sales trends. This system will replace the time-consuming and error-prone paper-based methods, providing quick and accurate access to information. Automated tools will generate customizable reports, improving data accuracy and decision-making.

1.4 Scope

The scope of this system can be divided by module and user.

1.4.1 System Scope

1.4.1.1 System Module

• Able to login as a administrator.

• Able to login as a customer.

1.4.1.2 Registration Module

In this module, the new customer needs to register into the system that required their name, id, number phone, address and password. The details about the customers will be stored into the database.

1.4.1.3 Custom Pizza Management Menu

In this module, admin can adding, editing, deleting, modifying managing details of custom pizza menu by list all pizza menu, pizza name, prices and menu availability .In addition, in this system ,customer will be notified if pizza menu is not available.

1.4.1.4 Pizza Ordering Module

In this module, the customers will have to choose the pizza that they wanted based on pizza type including the price that are stated. In addition ,the admin need to fetch detail of customer order such as name, address and telephone number and generate bill charge of pizzas.

1.4.1.5 Order Processing Module

This module is for a customer for confirmation order processing begins. In this module the order status will be updated and the customer will be notified. New orders will show up in updated the employees in the kitchen. The kitchen employees have to look at the order and decide if they can prepare the order or not. In addition, when the decision is made, a customer will be notified.

1.4.1.6 Payment Module

In this module, the admin used to manage payment details that required calculation such as payment based on the customer's order.

1.4.1.7 Report Module

In this module, a administrator can generate a report that shows the top sales pizza and the total pizza sold in this system.

1.4.2 User Scope

1.4.2.1 Administrator

In this module, the administrator are responsible for adding, updating, deleting and searching pizza menu also can generate a report that shows the top sales pizza and the total pizza sold in this system.

1.4.2.2 Customer

- able to view the information of pizza menu.
- able to order and view total amount of orders.

1.5 Project Significance

The Pizza Ordering Management System (POMS) is developed to streamline the pizza ordering process. This system enables customers to submit their orders and customize their pizzas online. The customization options help the pizzeria meet specific customer preferences. Hence, the system allows customers to easily place orders for delivery or pick-up. Additionally, admin can manage order schedules and track order statuses through the system. After fulfilling the orders, admin can insert delivery or service comments. This system helps to keep and manage customer and order data. By developing this system, the management of the ordering process will become more systematic and effective.

1.6 Expected Output

POMS will feature a user-friendly online interface where customers can easily navigate the menu, customize their pizzas, and complete their orders. Admin will have access to a comprehensive order management dashboard for handling incoming orders, assigning tasks, and tracking order statuses. The system will include real-time inventory management to monitor stock levels. Customers can create and manage accounts, view order history, and choose payment methods for quick checkout. Detailed sales and performance reports will be generated to help management analyze business performance. The system will optimize delivery and pick-up scheduling with real-time updates and route optimization for drivers. Customers can provide feedback and seek support through a dedicated system, while admin can review and respond to inquiries. This system aims to streamline operations, enhance customer satisfaction, and provide valuable business insights.

1.7 Conclusion

In this chapter is about the basic idea of the POMS. POMS is very useful and convenient for managing the pizza ordering process. Furthermore, this system is easy to use and increases data consistency. The management of the ordering process becomes more efficient and systematic with the development of the POMS.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

Database system development is the process of gathering accurate requirements, analysing requirements, data design and the system's capability, and then implementing each operation into practice within the modules. It is known as Database Life Cycle (DBLC) approach, this feature is used as the development methodology of a database system and ensures that the system has been developed adequately. This phase is discussed in this chapter in more detail.

2.2 Project Methodology

To develop POMS, the Database Life Cycle (DBLC) methodology is chosen. There are six phases in the DBLC: database initial study, database design, implementation & loading, training & evaluation, operations, and maintenance & evolution.

2.2.1 Database Initial Study

1) In this phase, a comprehensive review of existing pizza ordering systems was conducted to identify areas for improvement and determine the requirements for a new system. The review included a thorough literature review to understand the current state of pizza ordering systems, focusing on their functionality, user experience, data management, order processing, payment gateways, and security. The research highlighted key challenges such as difficulty in managing orders due to manual recording, human error in order processing and assignment and difficulty in searching and generating reports. Based on these findings, the objectives for the new system were defined: to ease pizza ordering and provide searching by having computerize order pizza online system and improve the efficiency of the ordering system, to computerize record payment and Implement an integrated digital record-keeping and reporting system. The scope of the system was outlined, including the required data structures, data types, number of entities, and the physical size of the database. This phase resulted in a clear understanding of the requirements and objectives for the new pizza ordering system, guiding its design and development.

2.2.2 Database Design

The second phase focuses on designing the database model to support the system operations and objectives of the Pizza Ordering Management System (POMS). This is a critical phase to ensure the final product meets system requirements. This phase consists of four sub-phases: conceptual design, DBMS software selection, logical design, and physical design. In conceptual design, an entity-relationship diagram (ERD) is created and normalized based on the system requirements to represent the data entities and relationships within the POMS. For DBMS software selection, MySQL is chosen as the DBMS software for this system due to its reliability and efficiency. Then, for logical design an agreement on data definitions and business logic is established. A relational data model is created to specify data and queries based on the data requirements. In physical design, the system is represented in tables, columns, indexes, sequences, and constraints after grouping attributes from the logical database. This design represents core business rules and data relationships at a detailed level.

2.2.3 Implementation and Loading

In this phase, MySQL is installed on the computer. Setup variables are tuned according to hardware, software, and usage conditions. Databases are created using Data Definition Language (DDL) based on the ERD. Data Manipulation Language (DML) is used to populate the database tables. The interface is developed using XAMPP APACHE 8.2.12 through PHP 8.2.12 with the hostname: Post (localhost: 80). Database performance, security standards, data integrity, and backup recovery procedures are also addressed during this phase.

2.2.4 Testing and Evaluation

This phase includes three sub-phases: testing the database, fine-tuning the database,and evaluating the database and its application programs.

a. Test the Database

Ensure data integrity and security using primary keys, foreign keys, unique constraints, password security, data encryption, privileges, and access rights. Conduct unit testing for individual functions and system testing for overall business processes.

b. Fine-Tune the Database

Correct errors and adjust the database to meet data requirements and improve performance.

c. Evaluate the Database

Ensure all components interact properly to meet data requirements through comprehensive testing.

2.2.5 Operations

After evaluation, the system becomes a complete information system. Users begin operating the system to produce the required information flow, such as loading data, managing data, and generating reports.

2.2.6 Maintenance and Evolution

The maintenance phase includes tasks such as maintaining indexes, optimizing tables, managing user access, and performing backups and restorations. As new requirements or changes arise, the system evolves, involving adaptive maintenance to add new fields or tables and enhance performance.

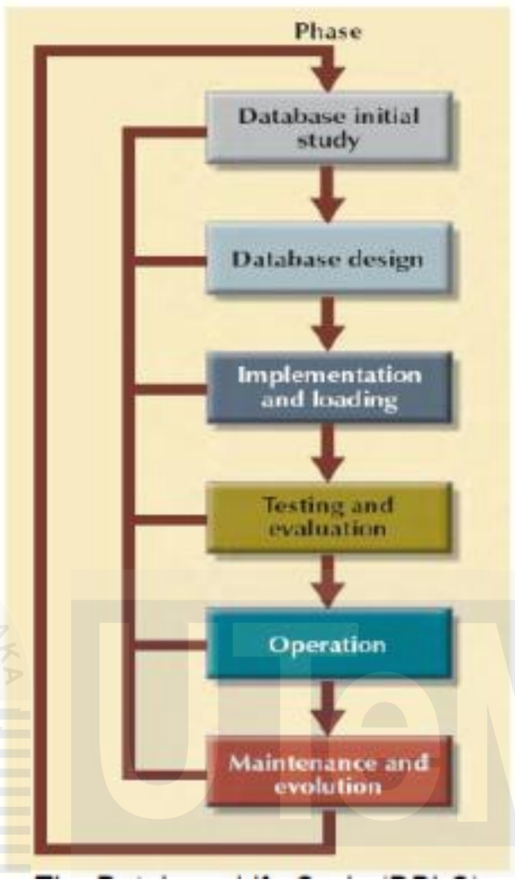


Figure 2.1 Database Life Cycle Illustration (JHCSC, 2016)

2.3 Project Schedule and Milestones

In Table 2.1, it shows the milestones of development of pizza ordering management system (POMS). IN each milestones, there are ouutcomes that are expected to be produced to ensure all process with the expected timeline and should be on time with relevant.

Table 2.1 Project Milestones

Milestones	Expected Outcomes	Date
1. Problem Identification and analysis	1. Problem statement, objective 2. Flow chart of the current system and the proposed system 3. Requirement and module of proposed system.	30 March 2024
Conceptual design of the proposed system	1. DFD and ERD of the proposed system 2. Business rules 3. Data Definition Language 4. Data Manipulation Language 5. Normalization and query	20 April 2024
Implementation of the proposed system	1. Database environment set up 2. Graphical User Interface (GUI) of the proposed system 3. Source code of the proposed system each function	25 May 2024
Testing of the proposed system functionality	1. Test plan 2. Test case 3. Test result and analysis 4. Solution solving of error message from testing process	15 July 2024
Completed documentation and log book	1. PSM 1 final report 2. Project Demonstration	30 August 2024
Project Demonstration	1. Final Presentation	5 September 2024

Table 2.2 it shows the overall development timeline for the progress.

Table 2.2 Gantt Chart of Pizza Ordering Management System

Week Task	1	2	3	4	5	6	7	8	9	1 0	1 1	1 2	1 3	1 4	1 5
Planning the Project	■	■													
Problem Identification and Analysis			■	■											
Database Design					■	■	■								
Development And Implementation the project						■	■	■	■	■	■	■			
Testing and Evaluation												■	■	■	
Presentation and Demonstration															■

2.4 Conclusion

In conclusion, Chapter 2 provides a comprehensive overview of the methodology employed in developing the database system, emphasising the Database Life Cycle (DBLC) approach. The eight phases of the Database Development Life Cycle (DDLC) are outlined, covering planning to deployment and maintenance.

CHAPTER 3: ANALYSIS

3.1 Introduction

This chapter will explore the analysis phase of the project, which involves breaking down the entire system into sub-modules and gathering factual data. It aims to understand the processes involved, identify existing problems, and propose practical suggestions for system improvement. This includes a detailed study of the current pizza ordering management system, gathering operational data, comprehending the flow of necessary information, pinpointing bottlenecks experienced by users, and devising solutions to streamline their tasks. The focus here lies in comprehending the intricacies of the current recruitment procedures, identifying weaknesses, and proposing enhancements for the new system that align with user requirements.

3.2 Problem Analysis

To delve into the shortcomings of the current pizza ordering process, an unstructured interview and observational study were conducted. Currently, the pizzeria operates on a manual system where all customer orders are documented using paper formats. This reliance on physical files creates significant inefficiencies as staff must manually sift through stacks of paperwork to locate specific customer orders. This not only increases the workload for administrative personnel but also consumes valuable time that could be better spent on improving customer service and operational efficiency.

Furthermore, the manual calculation of order totals increases the likelihood of human error, potentially resulting in incorrect billing amounts for customers. This not only impacts the accuracy of financial transactions but also affects the trust and satisfaction levels of customers.

In terms of customer service, delays often occur as staff wait for customers to complete order forms, especially during peak times when there is a high volume of orders. This waiting period not only affects the efficiency of service delivery but also contributes to customer frustration and dissatisfaction.

Additionally, the manual process of updating order statuses and generating handwritten summaries for sales reports adds further complexity and time consumption to the operational workflow. Administrative personnel must manually update and track each order's status, which can be prone to oversight and delays. Handwritten summaries of sales reports are not only time-consuming to produce but also prone to errors in data entry and interpretation.

3.2.1 Current System’s Flow

A flowchart is a visual representation that uses shapes and arrows to illustrate the sequence of steps, processes, or decisions in a system or procedure. It provides a clear and easy-to-understand overview of the flow of information or tasks, making it a helpful tool for planning, documenting, and understanding processes in various fields such as software development, business, and engineering.

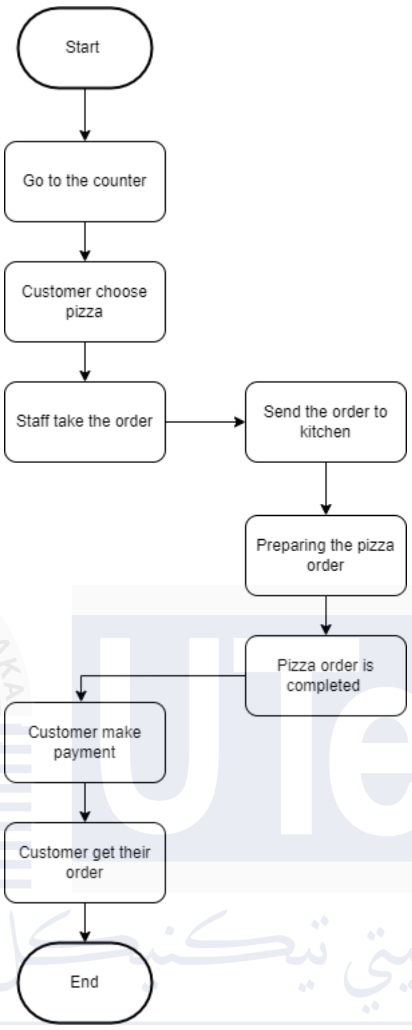


Figure 3.1 Current flow of the process for customer to order the pizza

Based on the scenario described in Figure 3.1, several problems in the current system can be identified. Firstly, the process of staff members manually recording the customer's order details can lead to errors in capturing the correct information. This manual recording also results in transmission delays as order details are manually sent from the counter to the kitchen, which can lead to miscommunication and affect order accuracy and preparation time. Additionally, there is a risk of preparation errors if the kitchen staff misinterpret the order details or overlook specific customer preferences and toppings.

The payment processing in the current system requires customers to make payment after the order is prepared. This could cause delays and congestion at the counter, especially during busy periods, increasing the overall wait time and reducing customer satisfaction. Moreover, the lack of a digital record of orders may make it difficult to track orders, manage inventory, and analyze sales data for business insights and improvements. The entire process heavily relies on staff members accurately recording, transmitting, and preparing the orders, leaving room for human error.

Furthermore, the system does not provide customers with real-time updates on their order status, which can leave them uncertain about the wait time and overall process. The sequential steps of order taking, preparation, and payment create bottlenecks and inefficiencies in the workflow, especially during peak hours. Identifying these problems highlights the areas where improvements can be made to streamline the pizza ordering process, reduce errors, and enhance the overall customer experience.

3.3 The proposed improvements/ solutions

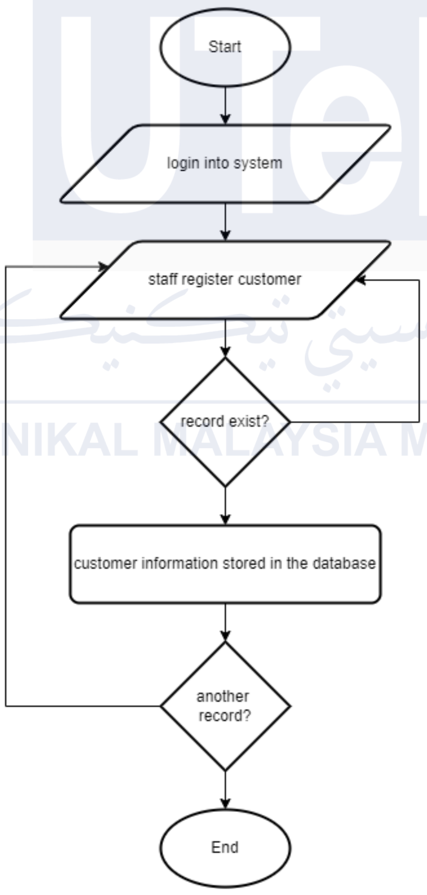


Figure 3.2 Current flow of the process for customer to order the pizza

Based on the scenario described in Figure 3.2, the proposed system aims to overcome the identified problems in the current pizza ordering process by introducing a more efficient and automated approach. Firstly, the system enables staff to login and access a digital interface to input all required customer details, reducing the risk of manual errors in recording orders. This interface also includes a validation feature that checks for existing records, preventing duplicate entries and ensuring data accuracy. If a record already exists, the system prompts the staff to re-enter the details, streamlining the registration process and avoiding unnecessary repetitions.

Furthermore, once a new customer's details are successfully registered, the information is automatically saved in a centralized database. This digital storage allows for easy retrieval and management of customer data, enhancing overall data organization. The proposed system also simplifies the addition of new records by allowing staff to select an option to add another record, repeating the streamlined process efficiently. This ensures that the process is consistent and reduces the time spent on administrative tasks, enabling staff to focus more on customer service and order management.

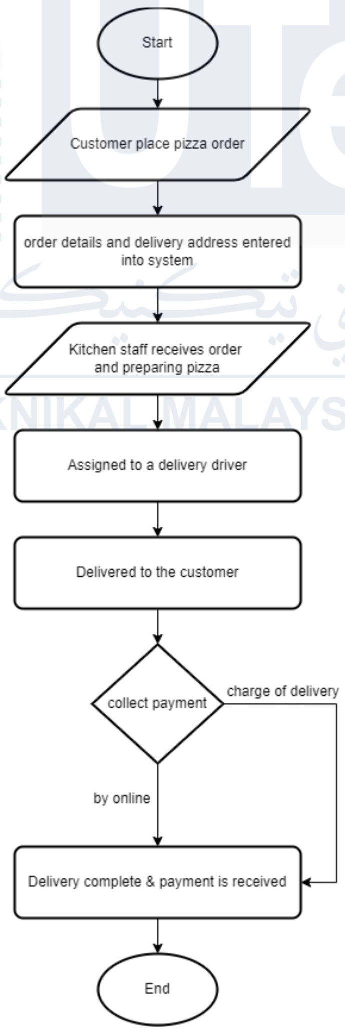

Start
Customer place pizza order
order details and delivery address entered into system
Kitchen staff receives order and preparing pizza
Assigned to a delivery driver
Delivered to the customer
collect payment
charge of delivery
by online
Delivery complete & payment is received
End

Figure 3.3 The proposed flow of process from the staff came to deliver the pizza until the process from order placement to delivery

Based on the scenario described in Figure 3.3, the proposed system for pizza ordering and delivery addresses the current issues by integrating a streamlined and automated process that enhances efficiency and accuracy. When a customer places a pizza order through the phone, online platform, or mobile app, the order details and delivery address are directly entered into the system, eliminating the need for manual entry and reducing the risk of errors. The system immediately sends an order confirmation to the customer, including an estimated preparation and delivery time, which improves communication and sets clear expectations.

The kitchen staff receives the order electronically, allowing them to start preparing the pizza without delay. This electronic transmission ensures that the specifications are accurately followed, minimizing preparation errors. Once the pizza is ready, it is packed in a heat-insulating box to maintain its temperature during delivery, ensuring the customer receives a hot and fresh product.

The system optimizes the delivery process by assigning the prepared pizza to a delivery driver and potentially optimizing delivery routes for efficiency. This reduces delivery times and improves overall service. The delivery driver then picks up the pizza and proceeds to the customer's location. If the customer has not prepaid online, the driver collects payment upon delivery, ensuring a smooth transaction.

After the delivery is complete and payment is received, the order is marked as complete in the system. This comprehensive tracking ensures that all steps of the order process are monitored and recorded, providing valuable data for future improvements. By automating these steps, the proposed system reduces manual errors, improves communication, enhances delivery efficiency, and ensures a better overall customer experience.

3.4 Requirement analysis of the to-be system

Requirement analysis will be describe on system requirement for each user and the functional requirement.

3.4.1 Functional Requirement (Process Model)

Customer:

1. Sign Up (only for new customer)

Input : "SignUp" option selected.

Output: customer prompted to enter the details.

2. Login

   Input : "Login" option selected.

   Output: customer prompted to enter the username and password.

3. Forgot Password

   Input : "Forgot Password" option selected.

   Output: customer prompted to enter the email and new password.

4. Select pizza items

   State : The customer has logged in and the main menu has been displayed.

   Input : Items are selected customer feel free to order.

   Output: system will display selected items.

5. Change order

   Input : "Go to Cart" option selected.

   Output: customer can add or delete pizza item in order.

6. Review the pizza order before submitting

   Input : "Order Place" option selected.

   State: customer name, phone number, location (address) display or enter the all of information.

   Output :customer promoted to pay the bill.

7. Payment

   State : The different types of payment method are display.

   Input: Choose any payment method.

   Output: customer prompted to enter the verification code if choose online payment.

   State: Display order number, payment details and confirmation of delivery.

8. Logout

   Input : "Logout" option selected.

Output: You are successfully logout.

State: System display login page.

Administrator

1. Login (Admin login page)

Input: “Login” option selected.

Output: Admin prompted to enter the username and password.

2. Modify Menu

State: In the system all the items are displayed with their rates.

Input: “Change” option selected.

Output: Admin can make changings in menu like adding or removing food items which are not available.

3. Order list

Input: “Order list” option selected.

State: System display all order details.

Output: Admin can view all order page .

4. Delivery

Input: “ Delivery” option selected.

State: System display the delivery details.

Output: Admin can select the staff role (delivery) make the delivery to send the order to the customer.

5. Logout

Input: “Logout” option selected.

Output: You are successfully logout.

State: System display login page.

3.4.1.1 Data Flow Diagram (DFD)

i. Context Diagram

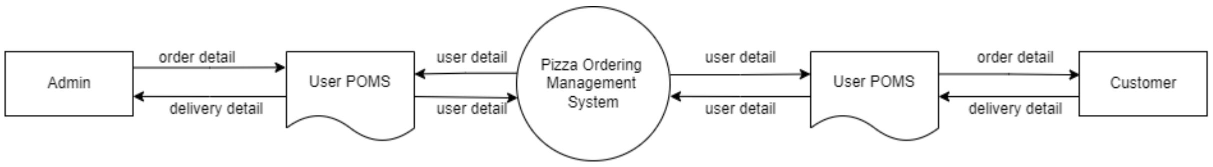


Figure 3.4 Context Diagram

Table shows a more detailed breakout of pieces of the Context Level Diagram. It highlight the main functions carried out by the system, such as breaking down the high-level process of the Context Diagram into its sub processes.

ii. Data Flow Diagram (DFD) Level 1

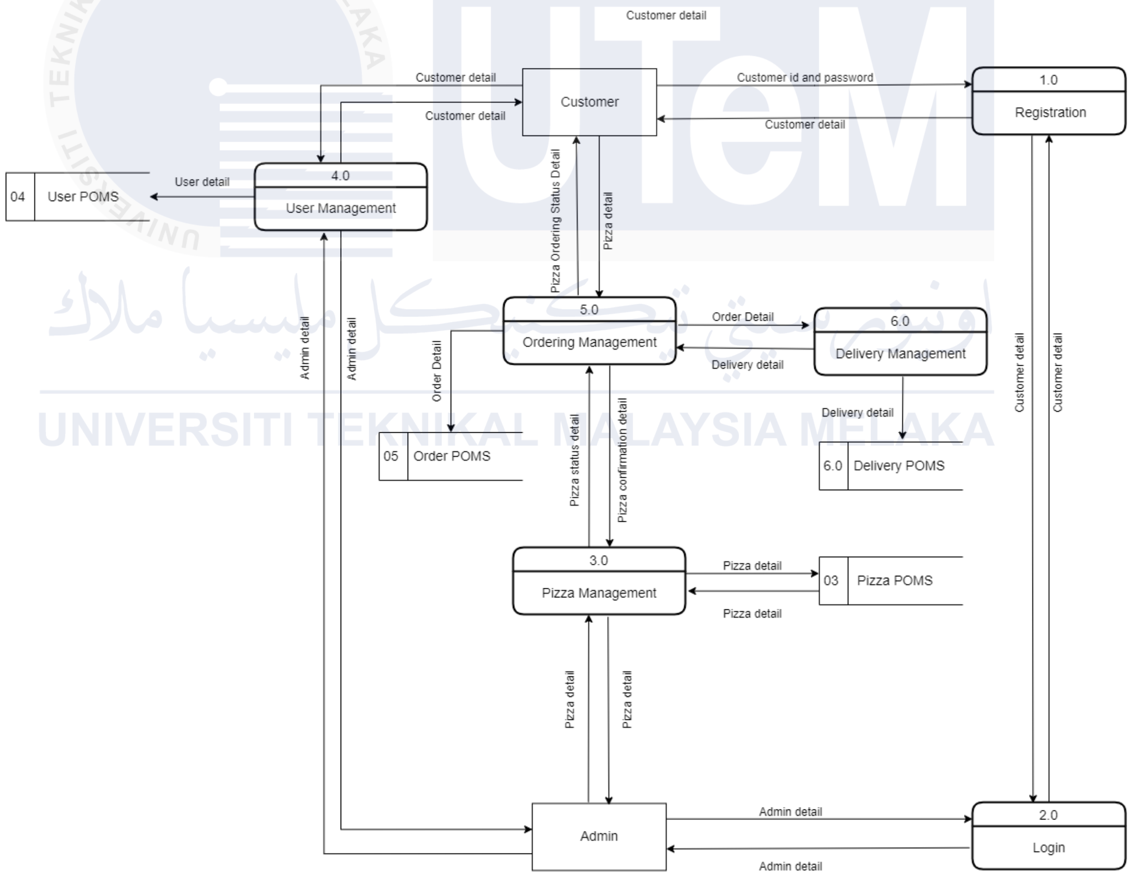


Figure 3.5 Data Flow Diagram (DFD) Level 1 POMS

Figure 3.5 shows the detailed workflow and interactions within the Pizza Ordering Management System (POMS), covering various modules including registration, pizza management, ordering, delivery, and customer management.

The process begins with the Admin PODS (01), where admin details are managed and provided. These details are sent to the Registration module (1.0), which processes and stores the admin information, allowing admins to be registered in the system. This ensures that only authorized personnel can manage the system's backend operations.

Customers start by logging in through the Login module (2.0) by providing their ID and password. The login details are authenticated, and once verified, customer information is managed by the Customer Management module (4.0). This module ensures that all customer details are accurate and up-to-date, with the information stored and retrieved from the Customer POMS database (04). This process facilitates seamless customer interactions with the system.

The Pizza Management module (3.0) is responsible for handling the details of the pizzas available in the system. Admins provide the necessary pizza details to this module, which then updates and manages this information. The Pizza Management module interacts with the Pizza POMS database (03) to store and retrieve pizza details, ensuring that the information is current and accurate. This module also manages the status and confirmation details of the pizzas.

The Ordering Management module (5.0) oversees the process of placing and managing pizza orders. It receives pizza details and order status updates from customers and processes these orders. The module then sends the order details to the Delivery Management module (6.0), which handles the delivery logistics. This module ensures that the orders are delivered accurately and efficiently, managing all delivery-related details.

iii. Data Flow Diagram (DFD) Level 2

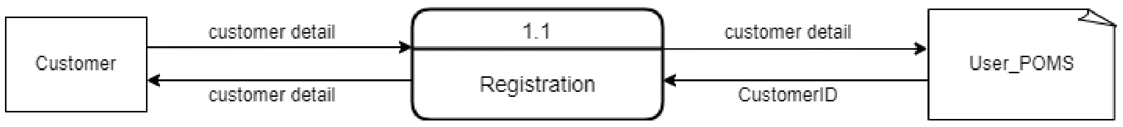


Figure 3.6 DFD Level 2 Registration of POMS

Table shows the registration process within the POMS. Firstly, the process begins with the customer who provides their details and then the customer details are sent to the Registration module, which processes the information. Next, the processed customer details are then stored in the User_POMS database , and a unique CustomerID is generated. Then, the registration module sends the customer details, including the newly generated CustomerId, back to the customer.

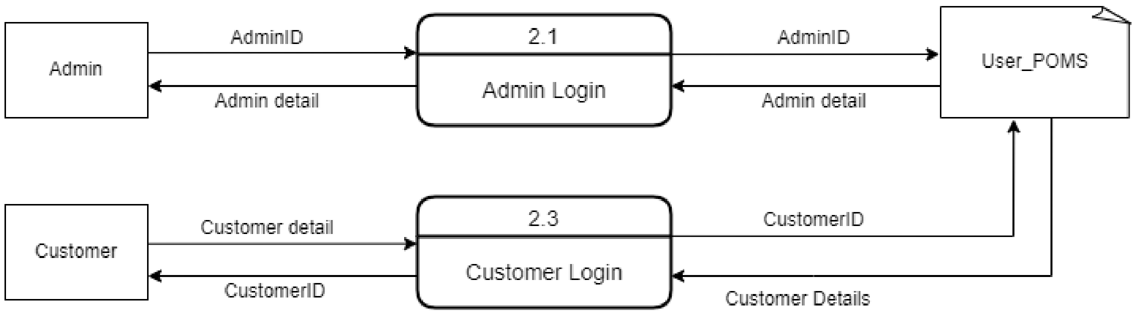


Figure 3.7 DFD Level 2 Login of POMS

Table shows the login processes for both admin and customer within the POMS. For Admin role, the admin initiates the login process by providing their AdminID then the AdminID is sent to the Admin Login module for authentication which then it communicates with the User_POMS database to retrieve the admin details associated with the provided AdminID. Once authenticated, the admin details are sent back to the Admin Login module which then provides the authenticated admin details back to the admin. For Customer role, The customer initiates the login process by providing their customer details. The customer details are sent to the Customer Login module for authentication. The Customer Login module communicates with the User_POMS database to retrieve the customer details associated with the provided CustomerID. Once authenticated, the customer details are sent back to the Customer Login module. The Customer Login module then provides the authenticated customer details back to the customer.

3.4.2 Non-Functional Requirements

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Table 3.1 shows the non-functional requirements and its description for POMS.

Table 3.1 Non-Functional Requirements

No	Non-functional	Description

1	Performance	-The system need to be reliable -Appropriate error message to notify user if the request is unable to process
2	Security	-The system should authorized users with username and password and identify the user access. -The password of the user must be encrypt
3	Usability	-The system have a friendly user interface and easy to learn how to operate the system -The system will guide the user on how to user and what is the next step of the process
4	Reliability	-The calculation of personality test result should be correct -The summary report generated should be accurate base on the data in database

3.4.3 Other Requirements

The requirements of database system development are dividing for two categories. A first category is software requirements and a second category is hardware requirement. Software requirement can describe the software that used to develop the system while the hardware requirement that used to support the system.

3.4.3.1 Software requirements

There have listed the requirements and specification of software components, which have been used in system. There are:

Table 3.2 Software Requirements

Software	Description

Visual Studio Code	Visual Studio Code (or VS Code for short) is a free, open-source, and powerful code editor developed by Microsoft. It is designed to be a lightweight and highly customizable IDE (Integrated Development Environment) that can be used for a wide range of programming languages and software development tasks.
Drawio	DrawIO (also known as Diagrams.net) is a free, open-source, and browser-based diagramming and flowchart tool. It allows users to create a wide variety of diagrams, flowcharts, organizational charts, network diagrams, UML diagrams, and more.
Microsoft Word 19	Microsoft Word is a word processor to write report.
XAMPP	XAMPP is used to run the MySQL database

PHP	PHP is a general-purpose programming language originally designed for web development.
Sublime text	Sublime text 3 is a source code editor to design the interface and implementation coding.

3.4.3.2 Hardware requirements

The list of hardware component that will be used in the system is as shown in the Table 3.10.

Table 3.3 Hardware Requirement Detail

No	Hardware	Description
1	Processor	Intel Core i5 10 th Gen
2	Memory	12 GB
3	Drive	SSD
4	Printer	PIXMA G2010 -Inkjet Printer

3.5 Conclusion

In conclusion, system analysis for the "Pizza Ordering System," breaking down the existing system to identify its strengths and weaknesses. Utilising flowcharts, context diagram (CD) and data flow diagrams (DFD). It provides a visual representation of the current system's processes. This analysis serves as a foundation for the upcoming system design, ensuring that the new system addresses existing flaws while leveraging strengths for optimal efficiency.

CHAPTER 4: DESIGN

4.1 Introduction

Systems analysis, architectural patterns, application programming interfaces, and design patterns all contribute to system designs that can be created without writing any code at all. A proper design that does not necessitate extraneous expenses and maintenance efforts and provides a better experience for the end-users as our application manages architectural load is only possible if all the requirements are met.

There are many different types of system design, but one of the most crucial for our POMS was the interface design, which gave us as developers insight into how to make the system more intuitive and easier to use. Regarding that, the system is governed by the restrictions of quality attributes. Depending on the specifics of the use case, these characteristics could include reliability, scalability, efficiency, cost, and many others.

In addition, POMS uses a 3-Tier architecture comprising the presentation tier (user interface), logic tier (process management), and data tier (data storage and retrieval). The database design starts with a conceptual phase, creating a high-level blueprint using an Entity Relationship Diagram (ERD) to outline entities, relationships, and attributes. This is followed by the logical design phase, which details tables, relationships, keys, and constraints without considering specific database management systems. Normalization is applied to organize data, reduce redundancy, and improve data integrity by dividing large tables into smaller, related ones, ensuring an efficient and scalable database structure.

4.2 System Architecture Design

Online Pizza Management System is designed based on a 3-Tier design architecture. This architecture consists of three different types of functionalities: the presentation tier, the logic tier, and the data tier.

The presentation tier is the topmost level of the application, serving as the user interface that translates tasks and results to the user. The logic tier coordinates the application's processes, executes commands, makes logical decisions and evaluations, and performs calculations. It also transfers and processes data between the two surrounding layers. Lastly, the data tier involves storing and retrieving data from a database or file system. This information is then passed back to the logic tier for processing, and eventually, the processed data is presented to the user.

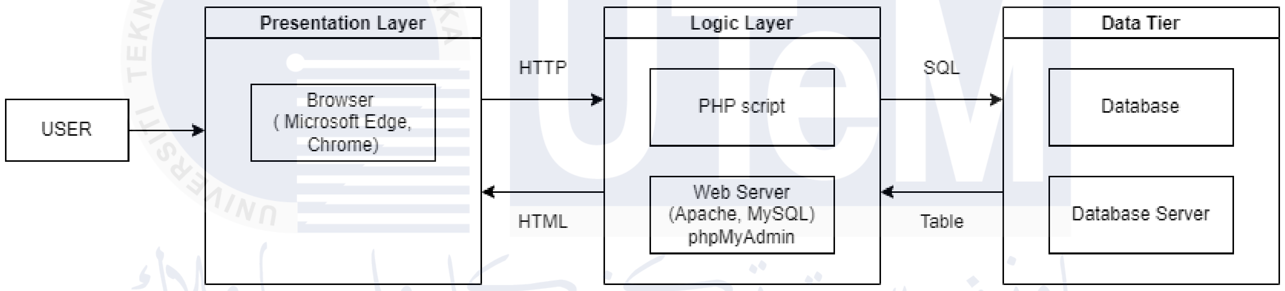


Figure 4.1 3-Tier Architecture

4.3 Database Design

Conceptual database design involves creating a high-level blueprint for a database, outlining essential entities, their relationships, and attributes. This data model has all the physical storage attributes and logical and physical design options required to produce a design in a data definition language, which can then be utilized to build a database. Each entity's specific properties are included in a fully attributed data model.

Most distinct elements of an overall database system's design can be referred to by the phrase "database design." It can be primarily—and fairly—thought of as the logical layout of the foundational data structures that are utilized to hold the data. These are the tables and views in the relational model. Relationships and entities in an object database map exactly to named relationships and object classes.

4.3.1 Conceptual Design

The database design approach starts with conceptual database design. To represent the database's structure, the Entity Relationship Diagram (ERD) was created. An entity relationship diagram demonstrates the connections between entity sets that are kept in a database. It serves as an example of the logical layout of a database. It also discusses how the database's tables interact with one another.

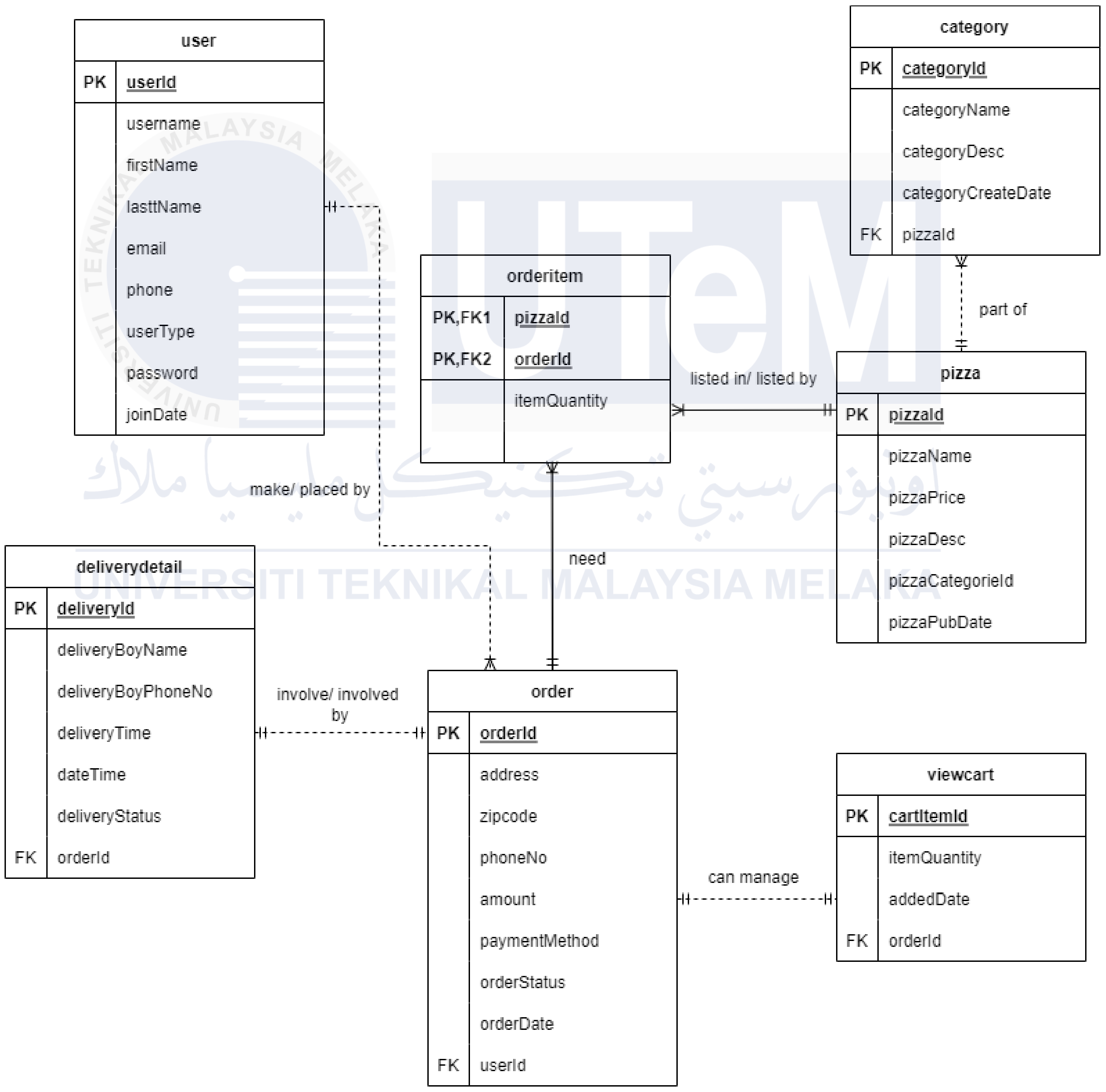


Figure 4.2 ERD of Pizza Ordering Management System (POMS)

Business rule of Pizza Ordering Management System

1. Each order can have one or many pizza and each pizza need in one or many orders.
2. Each user can make one or many orders and each order is placed by one and only one customer.
3. Each delivery detail can have one and only one order and each order can have one and only one deliverydetail.
4. Each pizza must have one and only one category and each category must have one or more pizza.
5. Each viewcart cab have one and only one order and each order can have one and only one viewcart

4.3.2 Logical Design

Logical database design is the process of translating the conceptual database model into a detailed structure, specifying tables, relationships, keys, and constraints. It defines the data organization and relationships without considering the specific database management system. In logical design, the ERD is translated into relational model. In relational model, entities are translated into tables. Data dictionary consists of the structure design of the tables following the data types of the targeted DBMS (MySQL).

4.3.2.1 Data Dictionary

a) Categories Table

This table keep each Categories detail information.

Table 4.1 Table Categories

Attribute	Description	Data Type	Required	PK/FK	Reference Table
categorieId	Category ID	Int(12)	Yes	PK	
categorieName	Category Name	Varchar(255)	Yes		
categorieDesc	Category Description	text	Yes		
categorieCreateDate	CategoryCreateDate	datetime	Yes		

b) Contact Table

This table keep each contact detail information.

Table 4.2 Table Contact

Attribute	Description	Data Type	Required	PK/FK	Reference Table
contactId	Contact Id	Int(21)	Yes	PK	
userId	User Id	Int(21)	Yes	FK	Users
email	Email	Varchar (35)	Yes		
phoneNo	Phone Number	Bigint(21)	Yes		
orderId	Order Id	Int(21)	Yes	FK	Orders
message	Message	text	Yes		
time	Time	datetime	Yes		

c) ContactReply Table

This table keep each contact reply detail information.

Table 4.1 Table ContactReply

Attribute	Description	Data Type	Required	PK/FK	Reference Table
Id	Contact reply ID	Int(21)	Yes	PK	
contactId	Contact ID	Int(21)	Yes	FK	contact
userId	UserID	Int(23)	Yes	FK	users
message	Message	text	Yes		
datetime	DateTime	datetime	Yes		

d) DeliveryDetails Table

This table keep each delivery detail information.

Table 4.2 Table DeliveryDetails

Attribute	Description	Data Type	Required	PK/FK	Reference Table
Id	Delibery ID	Int(21)	Yes	PK	
orderId	Order Id	Int(21)	Yes	FK	orders
deliveryBoyName	Deliver Name	Varchar(35)	Yes		
deliveryBoyPhoneNo	Deliver Phone Number	Bignit(25)	Yes		

deliveryTime	Delivery Time	Int(200)	Yes		
datetime	DateTime	datetime	Yes		

e) OrderItems Table

This table keep each orderItems detail information.

Table 4.3 Table OrderItems

Attribute	Description	Data Type	Required	PK/FK	Reference Table
id	Order Item ID	Int(21)	Yes	PK	
orderId	Order ID	Int(21)	Yes	FK	orders
pizzaId	Pizza Id	Int(21)	Yes	FK	pizza
itemQuantity	Quantity of Item	Int(100)	Yes		

f) Orders Table

This table keep each orders detail information.

Table 4.4 Table Orders

Attribute	Description	Data Type	Required	PK/FK	Reference Table
orderId	Order Id	Int(21)	Yes	PK	
userId	User Id	Int(21)	Yes	FK	users
address	Address user	Varchar(255)	Yes		
zipCode	Zipcode User	Int(21)	Yes		
phoneNo	Phone Number	Bigint(21)	Yes		
amount	Amount to pay	Int(200)	Yes		
paymentMode	Payment Method	Enum('0','1')	Yes		

g) Pizza Table

This table keep each pizza detail information.

Table 4.5 Table Pizza

Attribute	Description	Data Type	Required	PK/FK	Reference Table
pizzaId	Pizza Id	Int(12)	Yes	PK	

pizzaName	Name of pizza	Varchar(255)	Yes		
pizzaPrice	Price pizza	Int(12)	Yes		
pizzaDesc	Description of pizza	text	Yes		
pizzaCategorieId	Pizza Category Id	Int(12)	Yes		
pizzaPubDate	Pizza Publish Date	datetime	Yes		

h) SiteDetail Table

This table keep each Site detail information.

Table 4.6 Table SiteDetail

Attribute	Description	Data Type	Required	PK/FK	Reference Table
tempId	Site Detail Id	Int(11)	Yes	PK	
systemName	Name of site system	Varchar(21)	Yes		
email	Email of site system	Varchar(35)	Yes		
contact1	Phone Number site system 1	Bigint(21)	Yes		
contact2	Phone Number site System 2	Bigint(21)			
address	Address system	text	Yes		
dateTime	Datetime	datetime			

i) Users Table

This table keep each Users detail information.

Table 4.7 Table Users

Attribute	Description	Data Type	Required	PK/FK	Reference Table
id	User Id	Int(21)	Yes	PK	
username	Name of user	Varchar(21)	Yes		
firstName	User Firstname	Varchar(21)	Yes		
lastName	User Lastname	Varchar(21)	Yes		
email	User’s email	Varchar(35)	Yes		
phone	User’s phone	Bigint(20)	Yes		

	number				
userType	Usertype “customer’, admin’,’staff’	enum(‘0’, ‘1’, ‘2’)	Yes		
password	User’s password	Varchar(255)	Yes		
joinDate	User joindate	datetime	Yes		

j) Viewcart Table

This table keep each viewcart detail information.

Table 4.8 Table ViewCart

Attribute	Description	Data Type	Required	PK/FK	Reference Table
cartItemId	Cart Item Id	Int(11)	Yes	PK	
pizzaId	Pizza Id	Int(11)	Yes	FK	Pizza
itemQuantity	Quantity item	Int(100)	Yes		
userId	User Id	Int(11)	Yes	FK	users
addedDate	Added Date	datetime	Yes		

4.3.2.2 Normalization

The conceptual design are using normalization displayed, Each table shows attribute, primary key and foreign key. Figure 4.3 to 4.7 will show the 3NF of Pizza Ordering Management System (POMS).

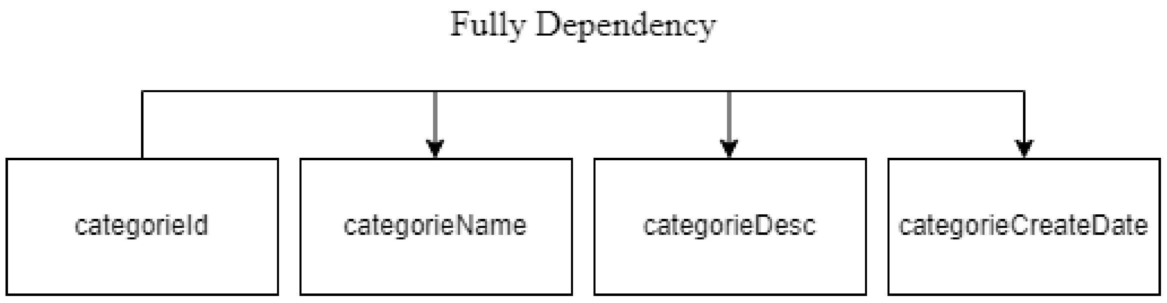


Figure 4.3 3 NF of Categories Table

Figure 4.3 illustrates that categories table is in 3NF.

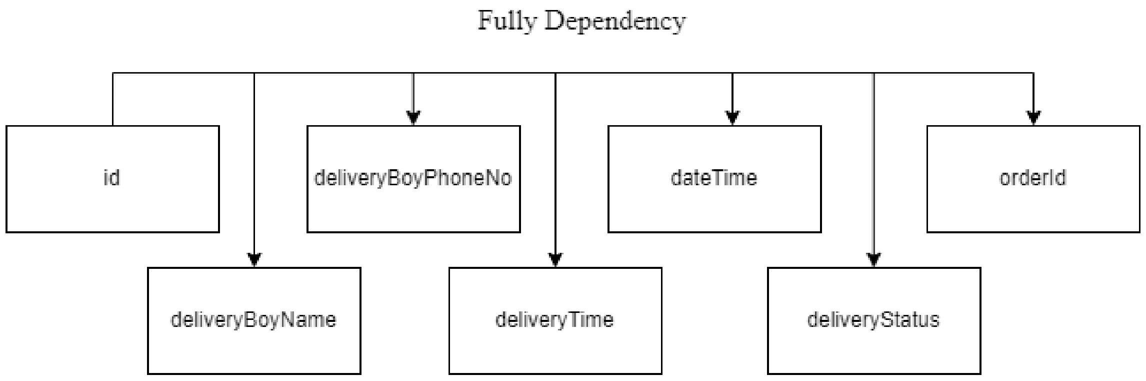


Figure 4.4 3NF of DeliveryDetails Table

Figure 4.4 illustrates that deliverydetails table is in 3NF.

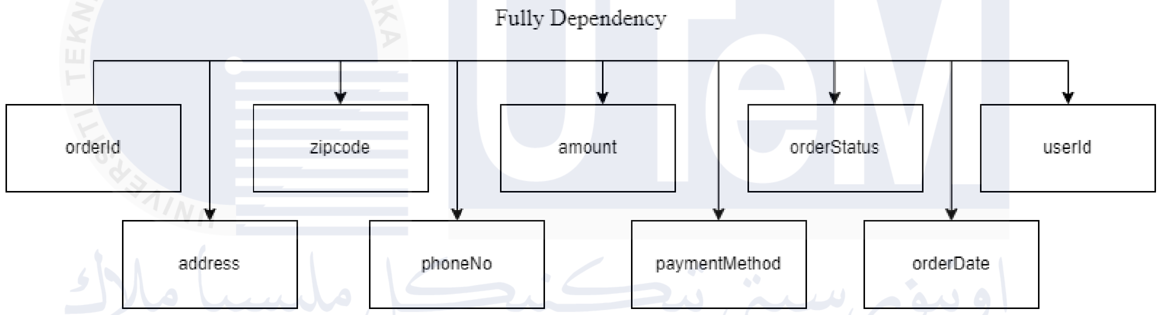


Figure 4.5 3 NF of Orders Table

Figure 4.5 illustrates that orders table is in 3NF.

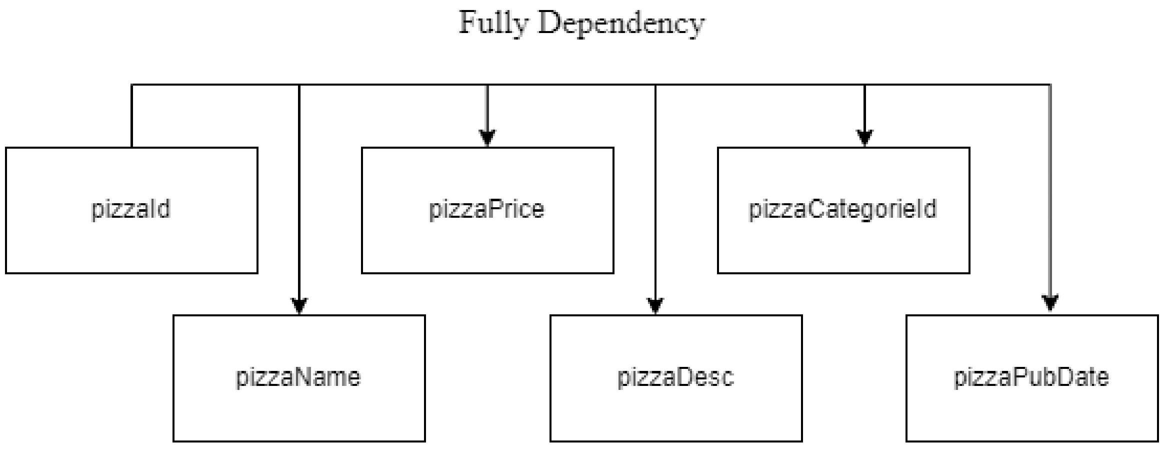


Figure 4.6 3 NF of Pizza Table

Figure 4.6 illustrates that pizza table is in 3NF.

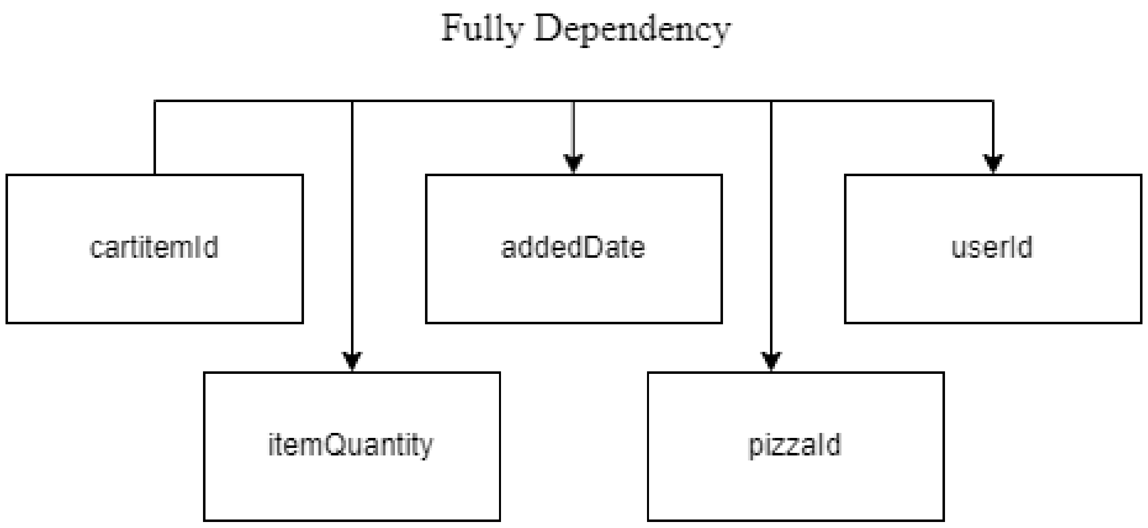


Figure 4.7 3NF of ViewCart Table

Figure 4.7 illustrates that viewcart table is in 3NF.

4.3.2.3 Query Design

There are many query design that can be carried out to produce many different types output. Each query have own requirement, reason and purpose. Table 4.11 will shows some examples of query design.

Table 4.9 Query Design of POMS

Type of Query	Query	Explanation
Simple Query	SELECT * FROM orders WHERE order_status = 'Completed';	To retrieve all orders where the order status is marked as "Completed."
Join Table Query	SELECT c.customer_id, c.customer_name, o.order_id, o.total_amount FROM customers c JOIN orders o ON c.customer_id = o.customer_id WHERE o.order_date = '2024-08-24';	To retrieve the customer information and their respective orders for a specific date (e.g., '2024-08-24').

Aggregate and Grouping Query	SELECT p.pizzaName, COUNT(oi.pizzaId) as count
	FROM orderitems oi
	JOIN pizza p ON oi.pizzaId = p.pizzaId
	GROUP BY oi.pizzaId
	ORDER BY count DESC
	LIMIT 5;
		To retrieve the top 5 most popular pizzas based on the number of times they were ordered.

4.3.2.4 Usage of Stored Procedures and Triggers

Table 4.10 Stored Procedure Relates Database Object Detail

Procedure	Database Table	Query	Explanation

Select	Orderitems	SELECT oi.pizzaId, p.pizzaName, SUM(oi.itemQuantity) AS totalQuantity
		INTO topPizzaId, topPizzaName, topPizzaQuantity
		FROM orderitems oi
		JOIN pizza p ON oi.pizzaId = p.pizzaId
		GROUP BY oi.pizzaId, p.pizzaName
		ORDER BY totalQuantity DESC
		LIMIT 1;
		
		SELECT topPizzaId AS PizzaID, topPizzaName AS PizzaName, topPizzaQuantity AS TotalQuantity;
			This SQL query involves two main parts: the first part selects the pizza with the highest total quantity ordered, and the second part retrieves that information for display.

Select	Users	SELECT COUNT(*) AS totalUsers FROM users;	Representing the total number of registered users in the system.

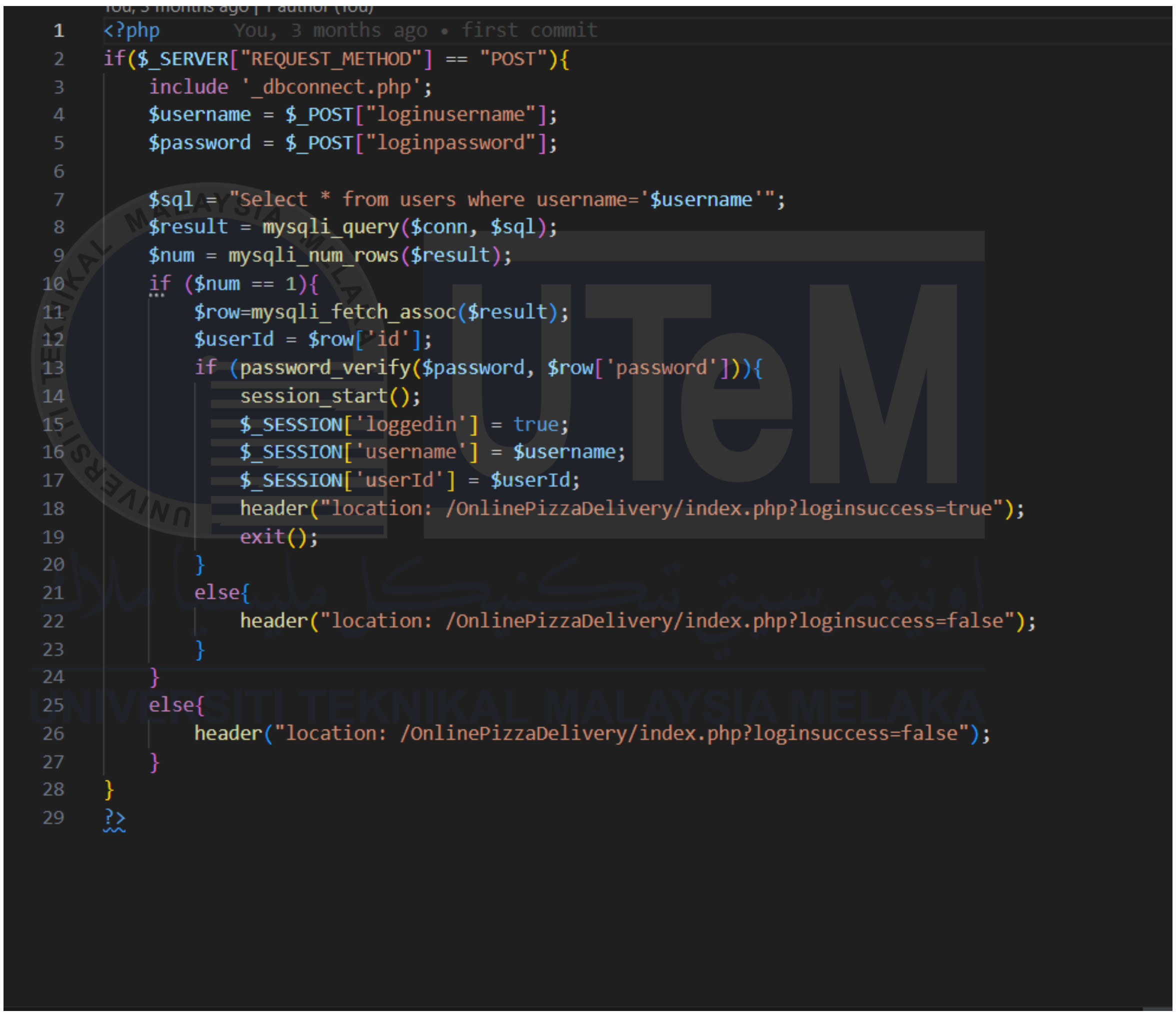
Select	Orders	SELECT COUNT(*) AS totalOrders FROM orders;	Representing the total count of orders in the system.

Table 4.11 Triggers Related Database Object Detail

Trigger	Database Table	Query	Explanation
After Insert	Orders	CREATE TRIGGER \`after_insert_orders_log_recent\` AFTER INSERT ON \`orders\` FOR EACH ROW BEGIN INSERT INTO recent_orders_log (orderId, orderDate, amount) VALUES (NEW.orderId, NEW.orderDate, NEW.amount); END	To insert the order's details into the recent orders log after inserting into the orders table.
After Update	Orders	CREATE TRIGGER \`log_order_updates\` AFTER UPDATE ON \`orders\` FOR EACH ROW BEGIN IF OLD.orderStatus <> NEW.orderStatus THEN INSERT INTO order_history (orderId, oldStatus, newStatus) VALUES (OLD.orderId, OLD.orderStatus, NEW.orderStatus); END IF; END	To log the old and new order statuses in the order history after updating the order's status in the orders table.

4.3.2.5 Security Mechanism

The system uses security mechanisms to verify that the user's email address and password are correct, identify the role of the user (admin or customer), grant access to the system, and direct users to the appropriate user page based on their role. For instance, the user will login as admin if the user enter valid username and password for admin account.



```
You, 3 months ago | 1 author (You)
<?php        You, 3 months ago • first commit
if($_SERVER["REQUEST_METHOD"] == "POST"){
    include '_dbconnect.php';
    $username = $_POST["loginusername"];
    $password = $_POST["loginpassword"];

    $sql = "Select * from users where username='$username'";
    $result = mysqli_query($conn, $sql);
    $num = mysqli_num_rows($result);
    if ($num == 1){
        $row=mysqli_fetch_assoc($result);
        $userId = $row['id'];
        if (password_verify($password, $row['password'])){
            session_start();
            $_SESSION['loggedin'] = true;
            $_SESSION['username'] = $username;
            $_SESSION['userId'] = $userId;
            header("location: /OnlinePizzaDelivery/index.php?loginsuccess=true");
            exit();
        }
        else{
            header("location: /OnlinePizzaDelivery/index.php?loginsuccess=false");
        }
    }
    else{
        header("location: /OnlinePizzaDelivery/index.php?loginsuccess=false");
    }
}
?>
```

Figure 4.8 Validate Username and Password to identify the role of user

4.4 Graphical User Interface (GUI) design

A graphical user interface design defines the overall process of designing how a user will be able to interact with a system and the nature of the inputs that the system accepts and produces. It includes the screen displays, which provide navigation through the system, the screens and forms that capture data. In graphical user interface design consists of three parts which are navigation design, input design and output design.

4.4.1 Navigation Design

The navigation component of the interface design provides a straightforward access and clearly direction for the user. The following Figure illustrates the navigation design and path for Pizza Ordering Management System (POMS).

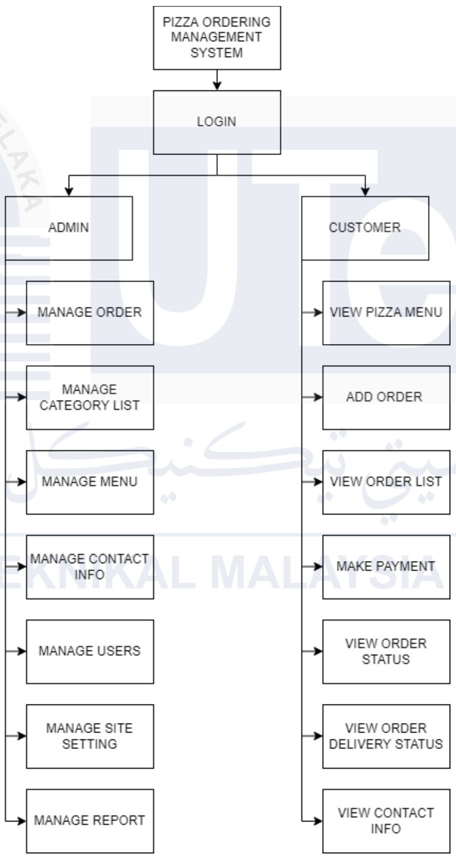


Figure 4.9 Navigation Path of POMS

4.4.2 Input Design

The input design is concerned with how users enter both organized and unstructured data into the system. The screen and forms are designed and used to store information for the system for an action performed. The Figure 6.2 until Figure 6.10 illustrates the input design can be referred in Appendix A.

4.4.3 Output Design

The output design concerns on presenting retrieved information of the system on the screen or form. The output design was showed at Figure 6.11 until 6.19 can be referred end in Appendix A.

4.5 Conclusion

This chapter concludes with information on creating a more systematic POMS design that satisfies both functional and non-functional needs, as well as features derived from the analysis in chapter III. In addition, the goal of the design phase is to find a solution for the issue that the requirement document in the analysis phase identified. The transition from the problem domain to the solution domain begins with this phase. The design document is the result of this step. Later, during the phases of implementation, testing, and maintenance, this design document is employed.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter, the task in implementing the database design is presented. The database installation and configuration procedure were showed. In the database implementation phase, MySQL is installed on Windows 11 platform. In this phase, Data Definition Language (DDL) and Data Manipulation Language (DML) in form of SQL statements are implemented in database. Implementation status are describe for each modules.

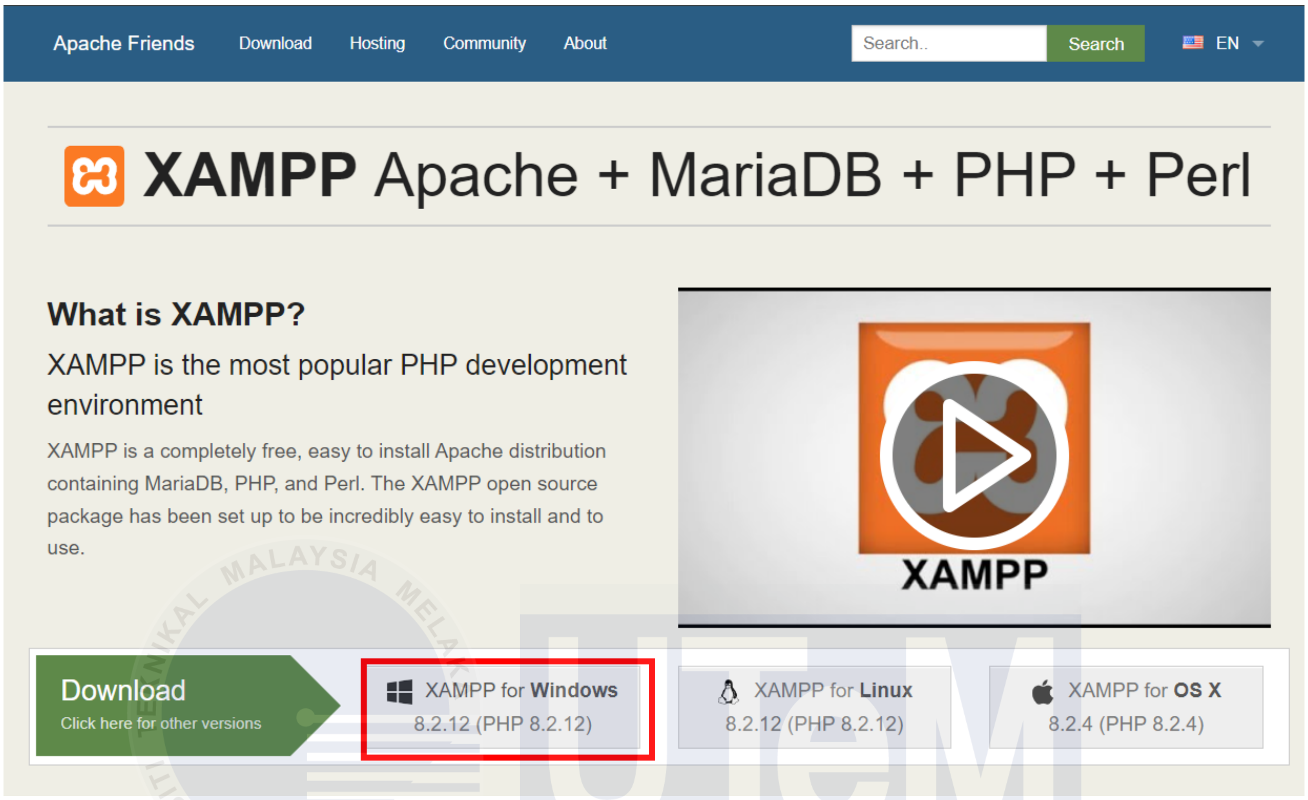
5.2 System Development Environment Setup

In POMS, the software development environment needs to be established prior to developing system, where we can create web applications. Additionally, the system development environment comprises four primary components: a Web Server using Apache, a Database Server utilizing MySQL, a Web Programming Language with PHP, and a Database Management Tool using phpMyAdmin. Furthermore, Windows users can download Apache Friends to acquire the free XAMPP package. Similarly, the key software components mentioned above can be easily installed via XAMPP.

5.2.1 Steps of Installation Setup

Installation of XAMPP

Step 1 : In web browser, go to https://www.apachefriends.org/ and download XAMPP for Windows.


Apache Friends Download Hosting Community About Search.. Search EN

XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

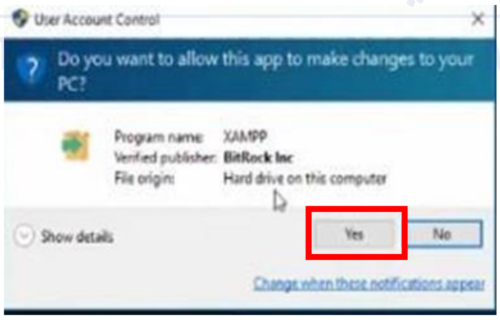
XAMPP is the most popular PHP development environment

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.

XAMPP
Download
Click here for other versions
XAMPP for Windows 8.2.12 (PHP 8.2.12)
XAMPP for Linux 8.2.12 (PHP 8.2.12)
XAMPP for OS X 8.2.4 (PHP 8.2.4)

Figure 5.1 Download XAMPP for Windows

Step 2 : Based on the Figure 5.2 , click the Yes button at XAMPP installer.

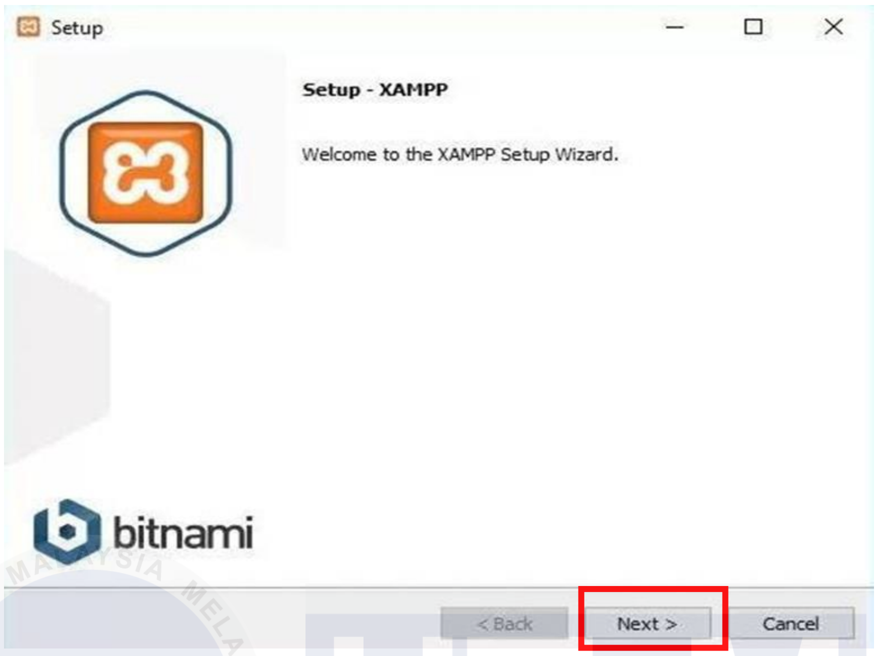

User Account Control

Do you want to allow this app to make changes to your PC?

Program name: XAMPP
Verified publisher: BitRock Inc
File origin: Hard drive on this computer

Figure 5.2 XAMPP installer

Step 3 : Click the Next button to setup XAMPP in Figure 5.3

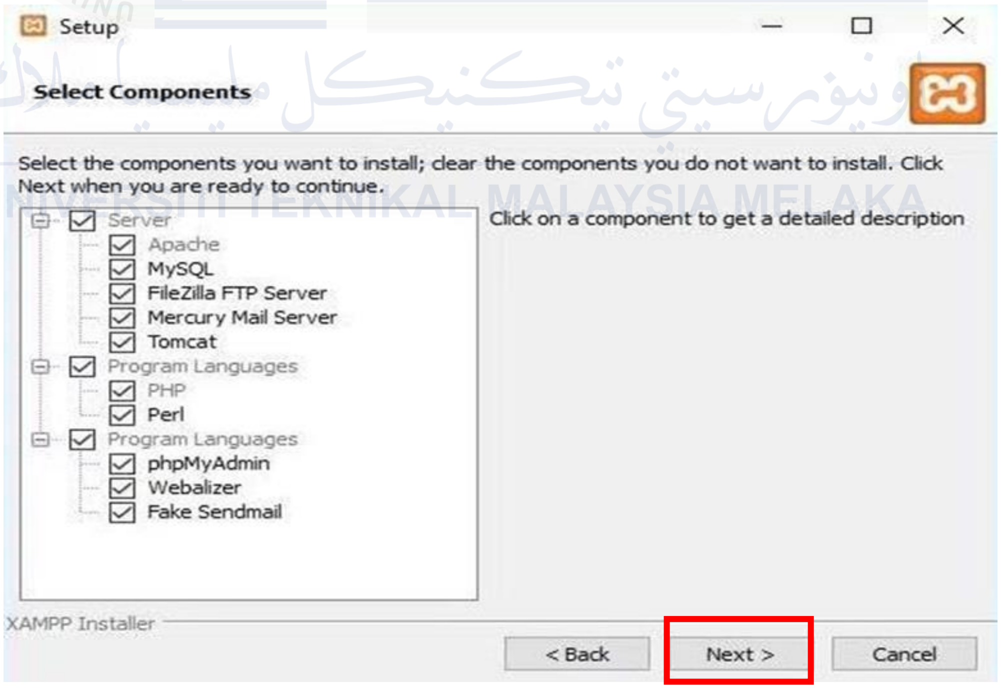

Setup

Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

Figure 5.3 XAMPP Setup Wizard

Step 4 : In Figure 5.4, click the Next button 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

Click on a component to get a detailed description

XAMPP Installer

Figure 5.4 Select Components

Step 5: To install XAMPP, select the location button in figure 5.6 and click the Next button to proceed. After that, the installation of XAMPP will begin, as seen in Figure 5.7.

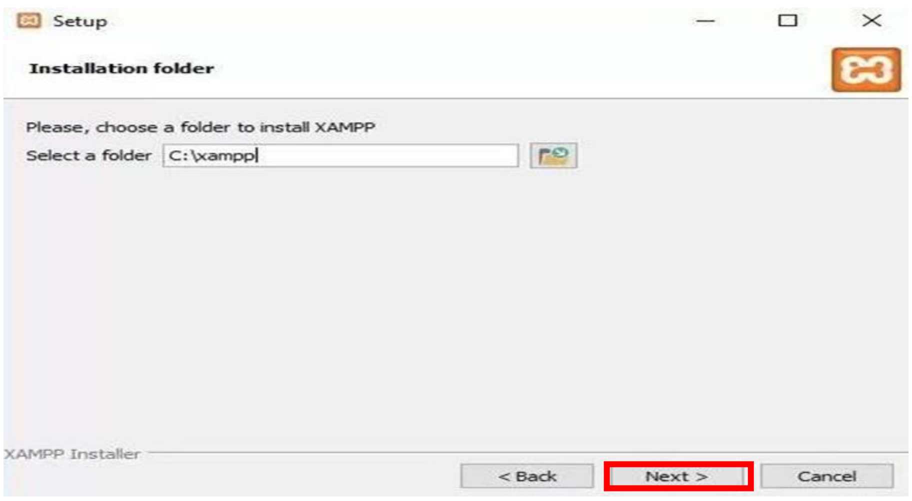


Figure 5.5 Select Installation Folder Location

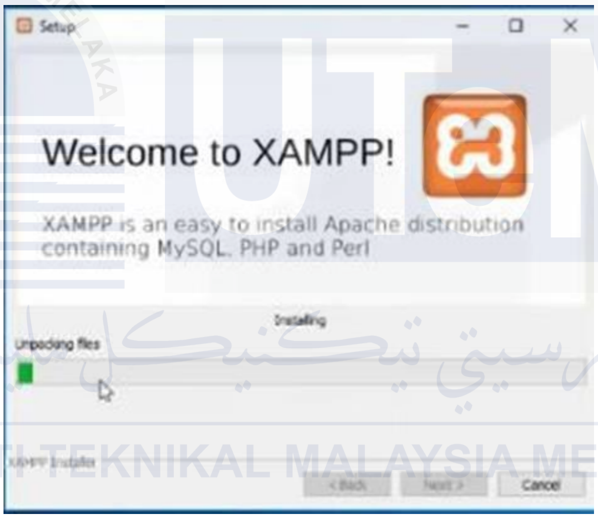


Figure 5.6 Waiting Installation XAMPP Installation

Step 6: Figure 5.7 shows the XAMPP setup wizard is completing. Click Finish button to start with XAMPP control panel.

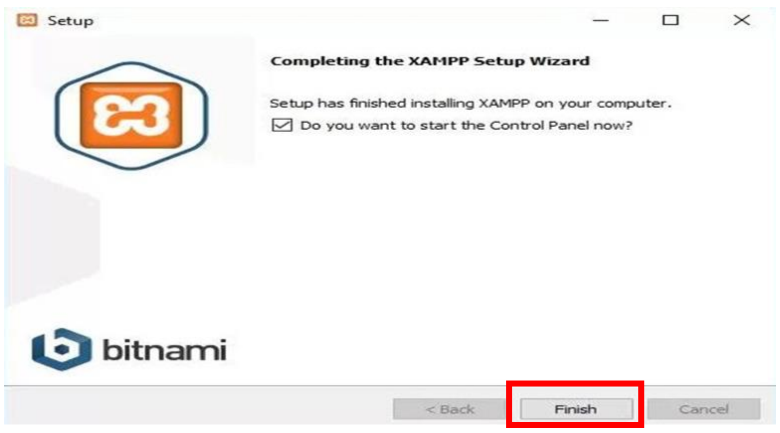
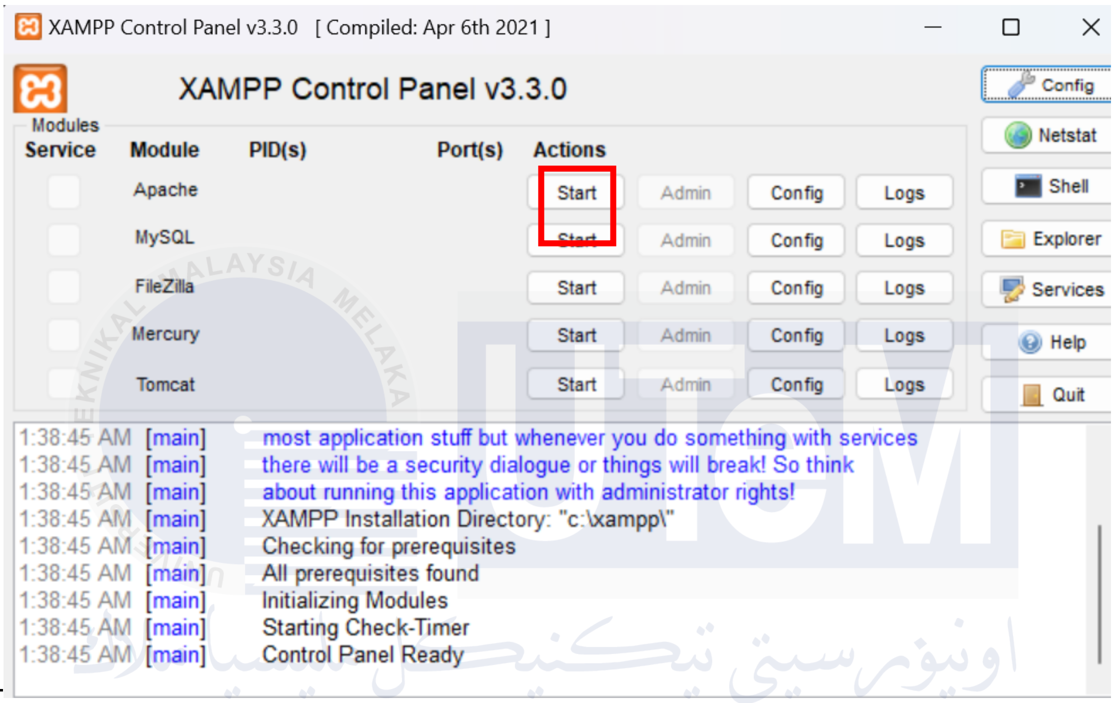

Setup
Completing the XAMPP Setup Wizard
Setup has finished installing XAMPP on your computer.
Do you want to start the Control Panel now?
bitnami
< Back
Finish
Cancel

Figure 5.7 Completing Installation of XAMPP Setup Wizard

Step 7 : Figure 5.8 shows the XAMPP control panel. Click the start button in Module Apache and MySQL to run the modules and wait until the modules show green color at the modules and this indicated that the both modules run successfully as shown in figure 5.9

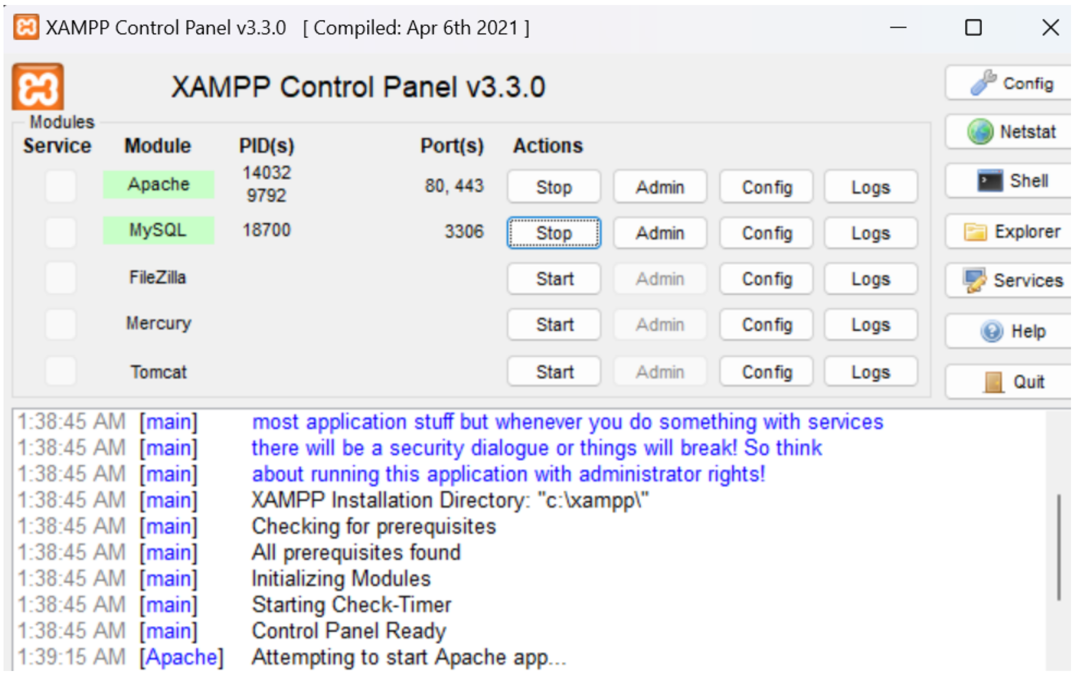


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

Service	Module	PID(s)	Port(s)	Actions			
	Apache			Start	Admin	Config	Logs
	MySQL			Start	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
1:38:45 AM  [main]   most application stuff but whenever you do something with services
1:38:45 AM  [main]   there will be a security dialogue or things will break! So think
1:38:45 AM  [main]   about running this application with administrator rights!
1:38:45 AM  [main]   XAMPP Installation Directory: "c:\xampp\"
1:38:45 AM  [main]   Checking for prerequisites
1:38:45 AM  [main]   All prerequisites found
1:38:45 AM  [main]   Initializing Modules
1:38:45 AM  [main]   Starting Check-Timer
1:38:45 AM  [main]   Control Panel Ready
```

Figure 5.8 Run the Apache and MySQL modules



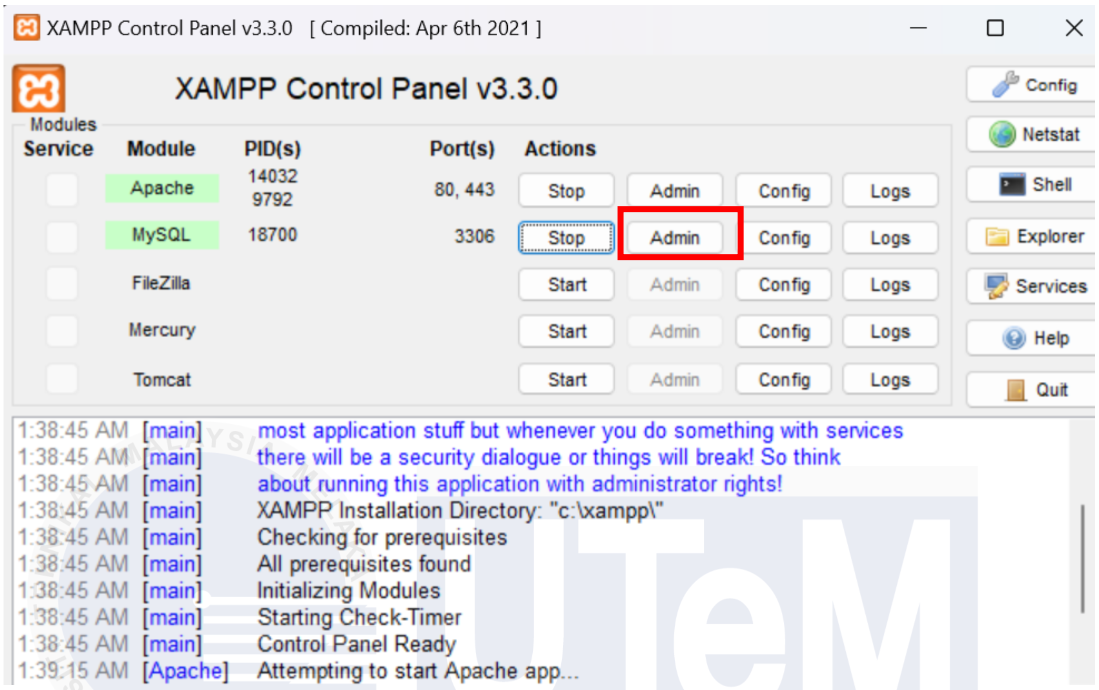
XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

Service	Module	PID(s)	Port(s)	Actions			
	Apache	14032 9792	80, 443	Stop	Admin	Config	Logs
	MySQL	18700	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
1:38:45 AM  [main]   most application stuff but whenever you do something with services
1:38:45 AM  [main]   there will be a security dialogue or things will break! So think
1:38:45 AM  [main]   about running this application with administrator rights!
1:38:45 AM  [main]   XAMPP Installation Directory: "c:\xampp\"
1:38:45 AM  [main]   Checking for prerequisites
1:38:45 AM  [main]   All prerequisites found
1:38:45 AM  [main]   Initializing Modules
1:38:45 AM  [main]   Starting Check-Timer
1:38:45 AM  [main]   Control Panel Ready
1:39:15 AM  [Apache] Attempting to start Apache app...
```

Figure 5.9 Apache and MySQL modules are running

Step 8 : Click the Admin button of the MySQL module to enter phpMyAdmin page in figure 5.10.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Config
Netstat
Shell
Explorer
Services
Help
Quit
Modules

Service	Module	PID(s)	Port(s)	Actions			
	Apache	14032 9792	80, 443	Stop	Admin	Config	Logs
	MySQL	18700	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
1:38:45 AM  [main]    most application stuff but whenever you do something with services
1:38:45 AM  [main]    there will be a security dialogue or things will break! So think
1:38:45 AM  [main]    about running this application with administrator rights!
1:38:45 AM  [main]    XAMPP Installation Directory: "c:\xampp\"
1:38:45 AM  [main]    Checking for prerequisites
1:38:45 AM  [main]    All prerequisites found
1:38:45 AM  [main]    Initializing Modules
1:38:45 AM  [main]    Starting Check-Timer
1:38:45 AM  [main]    Control Panel Ready
1:39:15 AM  [Apache]  Attempting to start Apache app...
```

Figure 5.10 Enter phpMyAdmin page

Step 9 : Figure 5.13 shown phpMyAdmin page. Click New button to create database.

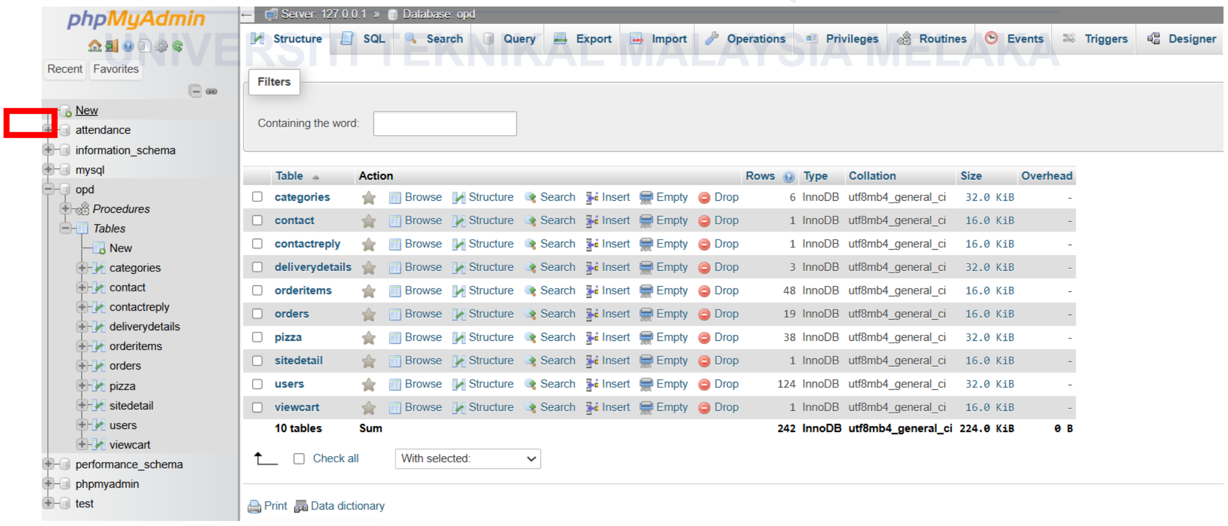


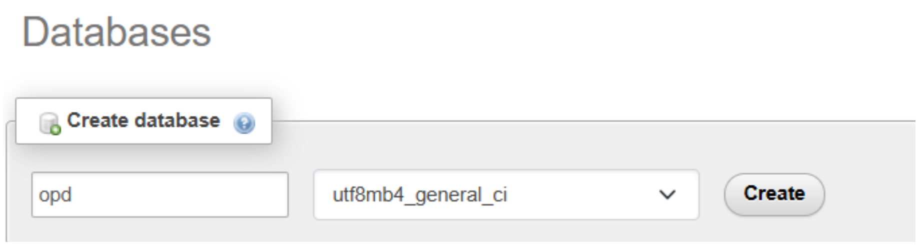
Table	Action							Rows	Type	Collation	Size	Overhead
categories	☆	Browse	Structure	Search	Insert	Empty	Drop	6	InnoDB	utf8mb4_general_ci	32.0 KiB	-
contact	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
contactreply	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
deliverydetails	☆	Browse	Structure	Search	Insert	Empty	Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
orderitems	☆	Browse	Structure	Search	Insert	Empty	Drop	48	InnoDB	utf8mb4_general_ci	16.0 KiB	-
orders	☆	Browse	Structure	Search	Insert	Empty	Drop	19	InnoDB	utf8mb4_general_ci	16.0 KiB	-
pizza	☆	Browse	Structure	Search	Insert	Empty	Drop	38	InnoDB	utf8mb4_general_ci	32.0 KiB	-
sitedetail	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
users	☆	Browse	Structure	Search	Insert	Empty	Drop	124	InnoDB	utf8mb4_general_ci	32.0 KiB	-
viewcart	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
10 tables	Sum							242	InnoDB	utf8mb4_general_ci	224.0 KiB	0 B

Check all — With selected:

Print Data dictionary

Figure 5.11 phpMyAdmin page

Step 10 : In figure 5.14, type the database name on text field and click Create. Then, create database was successful.

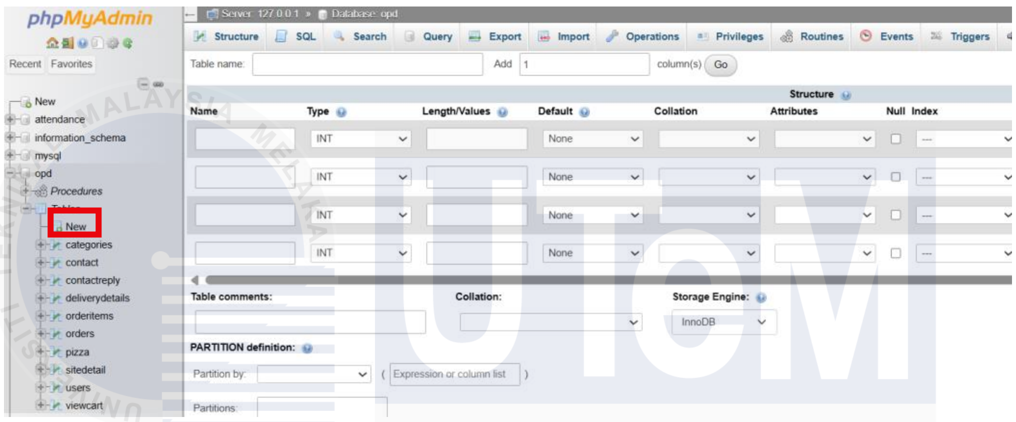


opd

utf8mb4_general_ci

Figure 5.12 Create Database

Step 11: In figure 5.15, click the new button to create new table.



Add 1 column(s)

Type	Default	Index
INT	None	---
INT	None	---
INT	None	---
INT	None	---

Storage Engine: InnoDB

Figure 5.13 Add New Table into Database

Step 12 : Enter each attribute's name, type, length, and index into the text field next to the database table name, then click Save. In figure 5.14, the addition of an attribute to the database table was then accomplished.

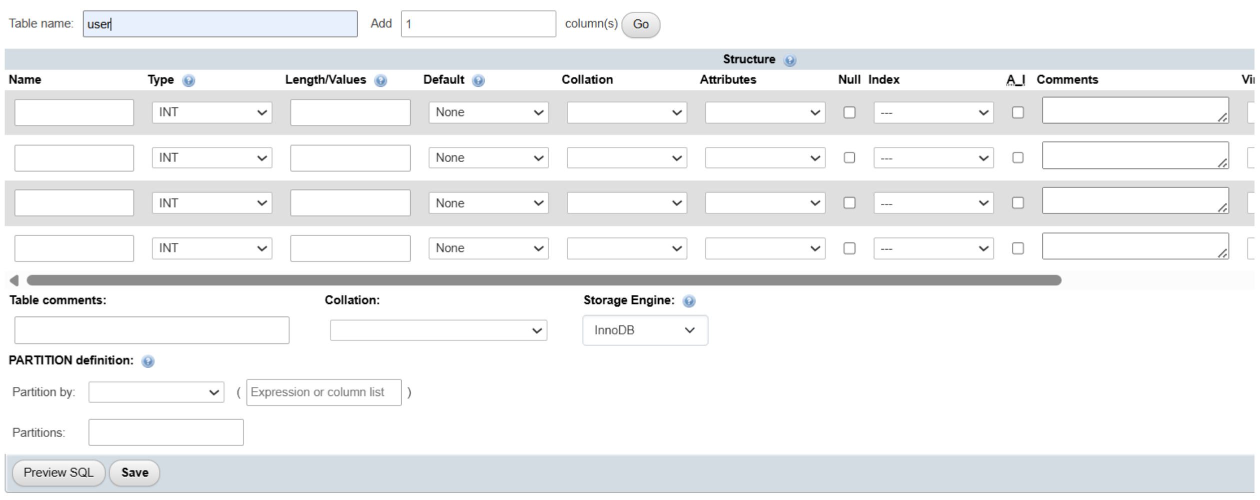


Table name: user

Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
	INT		None			☐	---	☐	
	INT		None			☐	---	☐	
	INT		None			☐	---	☐	
	INT		None			☐	---	☐	

Storage Engine: InnoDB

Figure 5.14 Add Attrribute into Table

5.3 Database Implementation

In POMS database implementation, the database are used to test the database query such as simple query, aggregate function, stored procedure and triggers into the system.

5.3.1 Data Definition Languange (DDL)

In a relational database, the SQL command used to build and modify tables is known as Data Definition Language, or DDL. Tables, indexes, and triggers for POMS can all be created, altered, and dropped from databases using DDL statements.

5.3.1.1 Create Table Commands

Figure 5.15 to Figure 5. Will show the system of each create table commands.

```
CREATE TABLE `categories` (
  `categorieId` int(12) NOT NULL,
  `categorieName` varchar(255) NOT NULL,
  `categorieDesc` text NOT NULL,
  `categorieCreateDate` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.15 Create Table categories

```
CREATE TABLE `contactreply` (
  `id` int(21) NOT NULL,
  `contactId` int(21) NOT NULL,
  `userId` int(23) NOT NULL,
  `message` text NOT NULL,
  `datetime` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.16 Create Table contactreply

```
CREATE TABLE `contact` (
  `contactId` int(21) NOT NULL,
  `userId` int(21) NOT NULL,
  `email` varchar(35) NOT NULL,
  `phoneNo` bigint(21) NOT NULL,
  `orderId` int(21) NOT NULL DEFAULT 0 COMMENT 'If problem is not related to the
order then order id = 0',
  `message` text NOT NULL,
  `time` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.17 Create Table contact

```
CREATE TABLE `deliverydetails` (
  `id` int(21) NOT NULL,
  `orderId` int(21) NOT NULL,
  `deliveryBoyName` varchar(35) NOT NULL,
  `deliveryBoyPhoneNo` bigint(25) NOT NULL,
  `deliveryTime` int(200) NOT NULL COMMENT 'Time in minutes',
  `dateTime` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.18 Create Table deliverydetails

```
CREATE TABLE `orderitems` (
  `id` int(21) NOT NULL,
  `orderId` int(21) NOT NULL,
  `pizzaId` int(21) NOT NULL,
  `itemQuantity` int(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.19 Create Table orderitems

```
CREATE TABLE `orders` (
  `orderId` int(21) NOT NULL,
  `userId` int(21) NOT NULL,
  `address` varchar(255) NOT NULL,
  `zipCode` int(21) NOT NULL,
  `phoneNo` bigint(21) NOT NULL,
  `amount` int(200) NOT NULL,
  `paymentMode` enum('0','1') NOT NULL DEFAULT '0' COMMENT '0=cash on delivery,
\r\n1=online ',
  `orderStatus` enum('0','1','2','3','4','5','6') NOT NULL DEFAULT '0' COMMENT '0=Order
Placed.\r\n1=Order Confirmed.\r\n2=Preparing your Order.\r\n3=Your order is on the
way!\r\n4=Order Delivered.\r\n5=Order Denied.\r\n6=Order Cancelled.',
  `orderDate` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.20 Create Table orders

```
CREATE TABLE `pizza` (
  `pizzaId` int(12) NOT NULL,
  `pizzaName` varchar(255) NOT NULL,
  `pizzaPrice` int(12) NOT NULL,
  `pizzaDesc` text NOT NULL,
  `pizzaCategorieId` int(12) NOT NULL,
  `pizzaPubDate` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.21 Create Table pizza

```
CREATE TABLE `sitedetail` (
  `tempId` int(11) NOT NULL,
  `systemName` varchar(21) NOT NULL,
  `email` varchar(35) NOT NULL,
  `contact1` bigint(21) NOT NULL,
  `contact2` bigint(21) DEFAULT NULL COMMENT 'Optional',
  `address` text NOT NULL,
  `dateTime` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.22 Create Table sitedetail

```
CREATE TABLE `users` (
  `id` int(21) NOT NULL,
  `username` varchar(21) NOT NULL,
  `firstName` varchar(21) NOT NULL,
  `lastName` varchar(21) NOT NULL,
  `email` varchar(35) NOT NULL,
  `phone` bigint(20) NOT NULL,
  `userType` enum('0','1','2') NOT NULL DEFAULT '0' COMMENT
'0=user\n1=admin\n2=staff',
  `password` varchar(255) NOT NULL,
  `joinDate` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.23 Create Table users

```
CREATE TABLE `viewcart` (
  `cartItemId` int(11) NOT NULL,
  `pizzaId` int(11) NOT NULL,
  `itemQuantity` int(100) NOT NULL,
  `userId` int(11) NOT NULL,
  `addedDate` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
  `address` text NOT NULL,
  `dateTime` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 5.24 Create Table viewcart

5.3.2 Data Manipulation Languange

```
INSERT INTO `users` (`id`, `username`, `firstName`, `lastName`,
`email`, `phone`, `userType`, `password`, `joinDate`) VALUES

(1, 'admin', 'admin', 'admin', 'admin@gmail.com', 1111111111, '1',
'$2y$10$AAfxRFOYbl7FdN17rN3fgeiPu/xQrx6MnvRGzqjVHlGqH
AM4d9T1i', '2021-04-11 11:40:58'),

(2, 'adam', 'ahmad', 'yazid', 'ahmad@gmail.com', 123456789, '2', 'staff2',
'2023-05-20 04:42:49'),

(3, 'ali', 'atan', 'ali', 'atan@gmail.com', 123456789, '0',
'$2y$10$Zgx4kddyNmwxsfYzWv.cFOnLa6TzjvQUSWXjuHPxk.l8sY
.GmTqn6', '2023-05-29 04:43:40'),
```

Figure 5.25 Insert Statement – insert data into table user

```
UPDATE categories SET  categorieName = 'Updated VEG PIZZA',
categorieDesc = 'An updated description for veggie lovers!'

WHERE   categorieId = 1;
```

Figure 5.26 Update Statement -update table categories

```
DELETE FROM `categories` WHERE categorieId = 5
```

Figure 5.27 Delete Statement – delete table categories

```
SELECT pizzaId, pizzaName, pizzaPrice, pizzaDesc FROM items

WHERE categorieId = 1;
```

Figure 5.28 Select Statement – retrieve table categories

5.3.3 Stored Procedures

Table 5.1 Stored Procedures Query

Decription	Stored Procedure Query

This SQL query involves two main parts: the first part selects the pizza with the highest total quantity ordered, and the second part retrieves that information for display.	SELECT oi.pizzaId, p.pizzaName, SUM(oi.itemQuantity) AS totalQuantity
	INTO topPizzaId, topPizzaName, topPizzaQuantity
	FROM orderitems oi
	JOIN pizza p ON oi.pizzaId = p.pizzaId
	GROUP BY oi.pizzaId, p.pizzaName
	ORDER BY totalQuantity DESC
	LIMIT 1;
	
	SELECT topPizzaId AS PizzaID, topPizzaName AS PizzaName, topPizzaQuantity AS TotalQuantity;

Representing the total number of registered users in the system.	SELECT COUNT(*) AS totalUsers FROM users;
Representing the total count of orders in the system.	SELECT COUNT(*) AS totalOrders FROM orders;

5.3.4 Triggers

In POMS, triggers are used to maintain the integrity of the information on the database.

Table 5.2 Triggers Query

Description	Query
To insert the order's details into the recent orders log	CREATE TRIGGER after_insert_orders_log_recent AFTER INSERT ON orders FOR EACH ROW

after inserting into the orders table.	BEGIN
	INSERT INTO recent_orders_log (order_id, order_date, total_amount)
	VALUES (NEW.order_id, NEW.order_date, NEW.total_amount);
	END;

To log the old and new order statuses in the order history after updating the order's status in the orders table.	CREATE TRIGGER log_order_updates
	AFTER UPDATE ON orders
	FOR EACH ROW
	BEGIN
	IF OLD.order_status <> NEW.order_status THEN
	INSERT INTO order_history (order_id, old_status, new_status)
	VALUES (OLD.order_id, OLD.order_status, NEW.order_status);
	END IF;
	END;

5.3.5 Data Loading Process

The Pizza Ordering Management System (POMS) is a system for storing and processing data to provide useful information to customers, kitchen staff, and managers. All data, such as customer details, order details, payment information, and inventory status, is stored in a database. In POMS, the Extract, Transform, Load (ETL) process is used to manage large amounts of structured and unstructured data. Extract involves collecting and reading data from multiple sources. Transform converts the required data into the appropriate format and stores it in another database. Load creates the data entries in the database, ensuring that all information is accurately recorded and easily accessible for efficient order management.

5.4 Implementation Status

Implementation status describe the module,description, duration to complete and date completes along the implementation process.

Table 5.3 Implementation Status

Module	**Description**	**Duration to complete/ days**	**Date completes**
System	Allows users (customers and admin) to interact with the Pizza Ordering Management System. It includes navigation, authentication, and user role management (admin, customer).	3	15/4/2024
Registration	To allow new users (customers) to create an account and register with the system.	5	21/4/2024
Custom Pizza Management Menu	To allow customers to create and customize their pizzas by selecting based on preferred categories pizza menu	5	28/4/2024
Pizza Ordering	To enable customers to select pizzas, customize them, and place orders.	10	10/5/2024
Order Processing	To handle the processing of customer orders, from placing an order to its preparation.	10	21/5/2024
Payment	To manage payments for pizza orders.	5	27/5/2024
Delivery Detail	To handle the delivery process, including managing delivery details and tracking.	10	12/6/2024
Report	To provide reports and analytics on various aspects of the Pizza Ordering Management System on total user, total user in the system and top selling pizza.	5	20/6/2024

5.5 Conclusion

In summary, this chapter outlines the setup of the pizza ordering management development, including procedures for installing Apache, XAMPP, PHP, and MySQL on a Windows platform. Additionally, it describes the database implementation to manage the system's processes. The system is built on business logic using Data Definition Language (DDL), Data Manipulation Language (DML), stored procedures, triggers, and commands to create database tables and constraints.

CHAPTER 6: TESTING

6.1 Introduction

Testing is the process of assessing a software application's functioning to determine whether or not it complies with the requirements and to uncover errors to guarantee that the system is error-free. This chapter will carry out the Pizza Ordering Management System (POMS) validation and verification.The two primary objectives of POMS testing are:

i. To show end users that POMS satisfies their needs;

ii. To find any errors or bugs in POMS using a different test approach.

Furthermore, system testing are important phase in Database Life Cycle (DBLC). POMS testing phase consists test plan which included test organization, test schedule and test environment.

6.2 Test Plan

A test plan is a technical document that outlines the test strategy and scope. It specifies the resources required for testing, including manpower, software, and hardware. Additionally, the test plan includes a schedule and details on test deliverables. It provides a comprehensive understanding of the system's workflow and functions. Furthermore, the document explains how each aspect will be tested to verify if the system operates according to its design. It aims to identify bugs and determine the system's actual limitations.

6.2.1 Test Organisation

In POMS, the test organization includes three users which are candidate, recruiter and recruiter manager. Functional requirements and non-requirements will be test on each users. InTable 6.1, it illustrates the four users undergoes testing based on their user responsibilities.

Table 6.1 User Responsibilities List

TesterID	Users	Responsibilities
T1	Admin	• Testing the system by following the test script given • Testing the admin module • Managing orders, updating menu items, and handling customer queries • Defect and bug detection
T2	Customer	• Testing the system by following the test script given • Testing the customer module • Placing pizza orders, tracking order status, and providing feedback • Defect and bug detection

6.2.2 Test Environment

Test Environment is a setup of software, hardware, operating system, tool required and network configuration for the testing teams to execute test cases. In table 6.2 and 6.3 show the hardware and software list required for POMS test environment.

Table 6.2 Test Environment Hardware List

Environment Specification	Description
Laptop	ASUS Laptop X415JP

Processor	Intel Core i5 th Gen
Random Access Memory (RAM)	12GB
Printer	PIXMA G2010- Inkjet Printer

Table 6.3 Test Environment Software List

Environment	Description
Database	MySQL • To manage data in the database table that runs on a server
Web Server	XAMPP Server • To create a local web server and use to deployment and testing purposes
Operating System / Platform	Windows 11 • To manage computer hardware and software resources and then provide the service or tool for computer program operate
Web Browser	Google Chrome / Microsoft Edge • To use run the PHP source code and test the system interface functionality
Sublime	• To use write PHP code
Microsoft Word 2019	• To use write final report
Canva	• To make slide for presentation

6.2.3 Test Schedule

A test schedule was created by outlining the main test milestones and providing a schedule summary. One test activity that is dependent on the system's completion date is scheduling. When creating the test schedule, dates should be estimated and revised as needed. The develop POMS test schedule is shown in Table 6.4.

Table 6.4 Test Schedule

Testing Task	**Testing Activity**	**Start Date**	**End Date**
Registration	Unit Testing, Integration Testing and User acceptance	18-7-2024	20-7-2024
Login	Unit Testing, Integration Testing and User Acceptance	20-7-2024	21-7-2024
Pizza Menu Searching	Unit Testing, Integration Testing and User Acceptance	23-7-2024	27-7-2024
CheckOut Page	Unit Testing and Integration Testing	25-7-2024	27-7-2024
Order Manage for Order Status and Delivery Details	Unit Testing and Integration Testing	31-7-2024	8-8-2024
New Category Pizza	Unit Testing and Integration Testing	7-8-2024	10-8-2024
New Item Pizza	Unit Testing and Integration Testing	12-8-2024	15-8-2024
Contact Us Form Page	Unit Testing and Integration Testing	23-8-2024	28-8-2024

6.3 Test Strategy

Test strategy is a set of guidelines that explains test design and determine showtesting needs to be done, the technique to be used and which modules to test. There are two types of testing which are black box testing and white box testing.

Black box testing is a software testing method in which the software is tested without knowledge of its internal code structure. This type of testing is typically carried out by testers who may not have implementation or programming knowledge. The primary focus of black box testing is to test the functionality of the system, including both functional and non-functional requirements. Furthermore, black box testing is generally applicable at a high level, such as in system testing and user acceptance testing. This method of testing is useful for simulating real user scenarios and ensuring that the software meets user needs.

White box testing is a software testing method in which the internal structure of the code is known to the tester. This type of testing is typically carried out by software developers who possess implementation and programming knowledge. The primary focus of white box testing is to test the program code of the system, including code structure, branches, and loops.In addition, White box testing is generally applicable at a lower level, such as in unit testing and integration testing. This method of testing is useful for identifying errors and defects in the code, and ensuring that the software meets its technical requirements.

Unit testing in the Pizza Ordering Management System (POMS) involves testing individual modules such as user registration, login functionality, order placement, and payment processing to ensure each unit performs as designed. For example, the unit module for user registration would be tested to validate user input and create a new user account. This level of testing helps identify and fix defects early in the development cycle, reducing the overall cost of defect fixing.

Integration testing in POMS involves testing the interactions between multiple modules, such as when a user places an order, the system updates the order status and sends a confirmation email to the user. This level of testing exposes faults in the interaction between integrated units, ensuring that the entire system functions as expected and provides a seamless user experience. For instance, integration testing would involve testing the workflow of a user placing an order, including logging in, selecting a pizza, proceeding to checkout, and processing payment.

System testing in the Pizza Ordering Management System (POMS) involves testing the complete and integrated software to evaluate its compliance with the specified requirements.

The purpose of this test is to ensure that the entire system, from user registration to order delivery, functions as expected and meets the desired quality and functionality standards. In POMS, system testing would involve testing the whole system flow, including user registration, login, order placement, payment processing, and order tracking, to ensure that it meets the specified requirements.

Table 6.5 illustrates the types of testing and test design techniques used in the Pizza Ordering Management System (POMS), which includes both white box and black box testing. In POMS, white box testing and black box testing were implemented to ensure the system's reliability and accuracy. In POMS, white box testing was used to test internal workings, focusing on correct data input types and data storage/transfer among modules, with examples including testing user registration module's input validation and database storage. Black box testing evaluated system functionality by inputting various data, such as valid/invalid payment info, to verify correct output responses.

Table 6.5 Type of test and test design technique for white box and black box testing

	White Box	Black Box
Type of Test	• **Unit Testing**: Verifies the functionality of individual components within the software. • **Integration Testing**: Focuses on testing the interaction between integrated modules to ensure proper data flow. • **System Testing**: Assesses whether the system as a whole behaves correctly by checking the flow of	• **Unit Testing**: Ensures the functionality of individual modules based on their specifications. • **Integration Testing**: Examines how different modules interact with each other within the system. • **System Testing**: Tests the entire system in a fully integrated environment, covering end-to-end scenarios.

	data and integration points.	• **User Acceptance Testing (UAT)**: Validates the system against user requirements, confirming it meets the business needs.
Test Design Techniques	• **Fault Insertion**: Intentionally introducing faults to evaluate system resilience. • **Error Handling**: Assessing the software's ability to handle and recover from errors. • **String Testing**: Evaluating the handling of strings and text-based inputs in the system. • **Statement Coverage**: Ensuring each statement in the code is executed at least once during testing. • **Decision Coverage**: Testing every possible decision point in the code to ensure all outcomes are evaluated.	• **Equivalence Partitioning**: Dividing input data into valid and invalid partitions to reduce the number of test cases while maximizing coverage. • **Boundary Value Analysis**: Testing at the boundaries between valid and invalid input ranges. • **Use Case Testing**: Verifying system behavior against real-world scenarios and workflows. • **Decision Table Testing**: Creating a table of conditions and actions to identify different combinations of inputs and outputs.

	• **Condition Coverage**: Ensuring every possible logical condition in the code is tested. • **Path Coverage**: Validating every possible execution path through the code to ensure all branches are covered.	

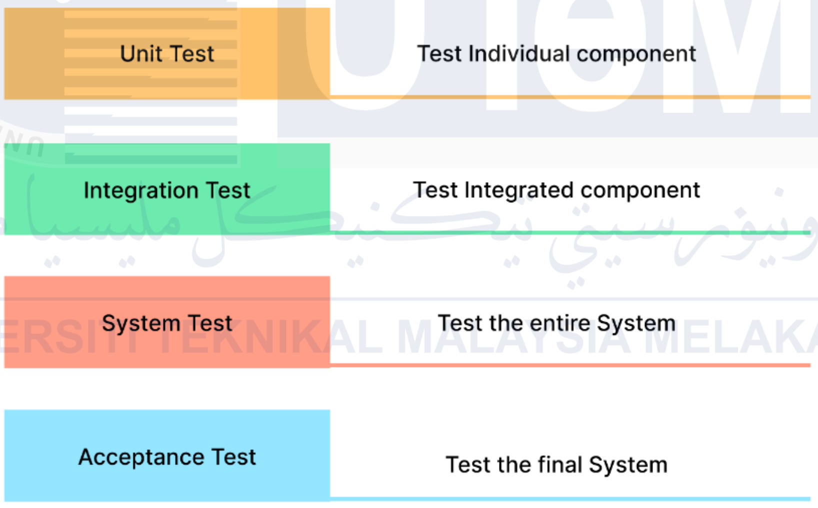


Figure 6.1 Level of Testing (Devaraj, 2023)

6.3.1 Classes of Tests

i. Error Handling Test

This testing is used to validate only the correct and accurate data input from the user can be filled in into the input field. For example, the user can only enter a valid email address in the "Email" field. In POMS, this testing was implemented to ensure the user enters the correct data before the order is processed and error message will be popped up on the screen to inform the user regarding their wrong or invalid input.Security Test

ii. Security Test

Security test will be conducted to validate and verify user login credentials, such as username and password, at the login process. In POMS, this testing was implemented to ensure that only authorized users can access the system and make orders.

iii. Integration Test

Integration test is to ensure that this system captures data into the database correctly based on what key in the data. It is done by moving through each and every menu item in the interface. Therefore, this testing was implemented in POMS to ensure the data are stored correctly and focus on the data transfer among the modules. For example, only the orders with complete and valid information will be stored in the database and displayed in the order list.

6.4 Test Design

Test design are describe to create and write test suites for testing a software. Purpose of the test design is to make sure that the requirements given are met that is parallel with what the client need and want. There are two parts of test design which are test description and test data.

6.4.1 Test Description

The test description outlines the identification of test cases, types of testing, preconditions, testing requirements, step-by-step procedures for each test case, and the expected output. Each module's test cases are carefully designed and documented to ensure thorough coverage and accuracy.Table 6.6 until table 6.13 will describe the test description in detail according to the system module.

Table 6.6 Test Description of Registration (User - Customer)

Test ID	T001 - Registration

Testing Type	Unit testing and integration testing			
Test Strategy	White box Testing			
Test Class	Security and error handling testing			
Test Case ID	**Test Requirements**	**Pre-condition**	**Test/Step Procedure**	**Expected Output**
TC1_1	Validate the registration function is available if the username, first name, last name, email, phone number and password provide are valid	User has valid username, first name, last name, email, phone number and password	1. Navigate to Sign up page 2. Provide valid username 3. Provide first and last name 4. Provide valid email 5. Provide phone number 6. Provide valid password 7. Click on Submit button	Sign up successful. Display message "Success! You can now login."
TC1_2	Validate the registration function is **Not** available if the username, first name, last name,email, phone number and password are blank		1. Navigate to Signup page 2. Click on Submit button	Sign up failed. Display error message "Please fill up this field."
TC1_3	Validate the registration function is **Not** available if the email provide are invalid		1. Navigate to Signup page 2. Provide invalid email 3. Click on Submit button	Sign up failed. Display error message "Please include an @ in the email address."
TC1_4	Validate the registration function is **Not** availableif the username provide are same as other user		1. Navigate to Signup page 2. Provide valid username 3. Click on Submit button	Sign up failed. Display error message "Warning!Username Already Exists."

Table 6.7 Test Description of User Login (User - Admin and Customer)

Test ID	T002 - Login			
Testing Type	Unit testing and integration testing			
Test Strategy	White box Testing			
Test Class	Security and error handling testing			
Test Case ID	**Test Requirements**	**Pre-condition**	**Test/Step Procedure**	**Expected Output**
TC2_1	Validate the login function is available if the username and password provide are valid	User has valid username and password	1. Navigate to login page 2. Provide valid username 3. Provide valid password 4. Click on Login button	Login successful .Display message "**Success!** You are logged in."
TC2_2	Validate the login function is **Not** available if the username or password are blank		1. Navigate to login page 2. Click on Login button	Login failed. Display error message "Please fill out this field"
TC2_3	Validate the login function is **Not** available if the username and password provide are invalid		1. Navigate to login page 2. Provide valid username 3. Provide invalid password 4. Click on Login button	Login failed. Display error message "Incorrect Username or Password! Please Enter Again."

Table 6.8 Test Description of Pizza Menu Searching

Test ID	T003 – Pizza Menu Searching

Testing Type	Unit testing and error handling testing		
Test Strategy	White Box Testing		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC3_1	Validate that the pizza menu search function returns results when searching by some keywords of pizza name is valid	1. Navigate to searching form page 2. Input the valid keywords of pizza name 3. Click on Search button	Searching Successful.Display the searching pizza menu information based on the keyword entered.
TC3_2	Validate that the pizza menu search function does not return results if the input data is a non-valid keyword or using number.	1. Navigate to searching form page 2. Input a non-valid keyword or number to the input search field 3. Click the Search button	Searching Failed. Display error message "No Result Found" with "Suggestions: • Make sure that all words are spelled correctly. • Try different keywords. • Try more general keywords."

Table 6.9 Test Description of CheckOut Page

Test ID	T004 – CheckOut Page		
Testing Type	Unit testing and error handling testing		
Test Strategy	White Box Testing		
Pre-condition	1. User must login as customer 2. User (customer) must at least one order pizza menu in the cart to checkout		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**

TC4_1	Validate the checkout function is available if the address, address line 2, zipcode and password are valid	1. Navigate to checkout page 2. Provide valid address and address line 2 (if needed) 3. Provide valid zipcode 4. Provide valid password 5. Click on Order button	CheckOut successful. Display message "Thanks for ordering with us. Your order id is ' '."
TC4_2	Validate the checkout function is NOT available if the address, address line 2, zipcode and password are blank	1. Navigate to checkout page 2. Click on Order button	CheckOut failed. Display error message "Please fill up this field."
TC4_3	Validate the checkout function is Not available if the password is invalid	1. Navigate to checkout page. 2. Provide invalid password 3. Click on Order button	Checkout failed. Display error message "Incorrect Password! Please enter correct Password."
TC4_4	Validate the checkout function is Not available if the zipcode is invalid	1. Navigate to checkout page. 2. Provide invalid password and zipcode 3. Click on Order button	Checkout failed. Display error message "Please match with requested format."

Table 6.10 Test Description of Order Manage of Order Status and Delivery Details

Test ID	T005 – Order Manage of Order Status and Delivery Details
Testing Type	Unit testing and error handling testing
Test Strategy	White Box Testing

Pre-condition	1. User must login as administrator 2. User have done order and payment in the ChechkOut page		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC5_1	Validate the order manage function is available if the delivery boy name is selected	1. Navigate to order status and delivery details 2. Select delivery boy 3. Click on Update button	Order manage successful. Display message “ Update successfully”
TC5_2	Validate the order manage function is Not available if the delivery boy name is not selected	1. Navigate to order status and delivery details 2. Click on Update button	Order manage failed. Display error message “Please select an item in the list.”

Table 6.11 Test Description of New Category Pizza

Test ID	T006 – Create New Category Pizza		
Testing Type	Unit testing and error handling testing		
Test Strategy	White Box Testing		
Pre-condition	User must login as administrator		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC6_1	Validate the category pizza function is available if all category name, category description and image is filled in	1. Navigate to Create new Category page 2. Enter category name, category description and image 3. Click on Create button	Create new category successful. Display message “New Category Has Been Added”

TC6_2	Validate the category pizza function is Not available if not all category name, category description and image is filled in	1. Navigate to Create new Category page 2. Enter whether is category name or category description and image 3. Click on Create button	Create new category failed. Display error message “Please fill out this field.” for category name and category description while for error message for image that not inserted “Please select a file.”

Table 6.12 Test Description of New Item Pizza

Test ID	T007 – Create New Item of Pizza Menu		
Testing Type	Unit testing and error handling testing		
Test Strategy	White Box Testing		
Pre-condition	User must login as administrator		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC7_1	Validate the item of pizza menu function is available if all name, description, price, category and image is filled in	4. Navigate to Create new item page 5. Enter category name, category description and image 4. Click on Create button	Create new item successful. Display message “New Item Has Been Added”
TC7_2	Validate the item of pizza menu function is Not available if not all name, description, price, category and image is filled	4. Navigate to Create new item page 5. Enter whether is name, or description or price or category or image 5. Click on Create button	Create new item failed. Display error message “Please fill out this field.” for name, description and price and category display the error message “ Please select an item in the list.” while for error message for image that not inserted “Please select a file.”

Table 6.13 Test Description of Contact Us Form Page

Test ID	T008 – Fill in the Contact Us Form Page		
Testing Type	Unit testing and error handling testing		
Test Strategy	White Box Testing		
Pre-condition	User must login as customer		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC8_1	Validate the contact us form function is available if all password and message is filled in	1. Navigate to Contact Us Form Page 2. Enter valid password and some message 3. Click on Submit button	Filled in successful. Display message “Thanks for Contact us. Your contact id is 3. We will contact you very soon.”
TC8_2	Validate the contact us form function is Not available if not all password and message is filled in	1. Navigate to Contact Us Form Page 2. Click on Submit button	Filled in failed. Display error message “Please fill out this field”
TC8_3	Validate the contact us form function is Not available if password entered is invalid	1. Navigate to Contact Us Form Page 2. Enter valid message 3. Enter invalid password 4. Click on Submit button	Filled in failed. Display error message “Password incorrect!”

6.4.2 Test Data and Test Result

Table 6.14 Test Data of Sign Up Registration (Customer)

Test Data ID	Username	First Name	Last Name	Email	Phone No	Password
TD1_1	hajar13	siti	hajar	shajar@gmail.com	191223458	hajar123
TD1_2	blank	blank	blank	blank	blank	blank
TD1_3	hajar13	siti	hajar	shajar	191223458	hajar123
TD1_4	hajar13	siti	hajar	shajar@gmail.com	191223458	hajar123

Table 6.15 Test Data of Login

Test Data ID	Username	Password
TD2_1	hajar13	hajar123
TD2_2	blank	blank
TD2_3	hajar13	hajar

Table 6.16 Test Data of Pizza Searching Menu

Test Data ID	Keyword for Search
TD3_1	Chicken
TD3_2	Beef

Table 6.17 Test Data of CheckOut page

Test Data ID	Address	Address Line 2	Zipcode	Password
TD4_1	No 13 Jalan Haji Soten	Kampung Padang Temu	75050	hajar123
TD4_2	blank	blank	blank	blank
TD4_3	No 13 Jalan Haji Soten	Kampung Padang Temu	75050	hajar
TD4_4	No 13 Jalan Haji Soten	Kampung Padang Temu	7505	hajar123

Table 6.18 Test Data of Order Manage

Test Data ID	Delivery Boy Name	Phone No
TD5_1	Ahmad Yazid	123456789
TD5_2	blank	blank

Table 6.19 Test Data of New Category Pizza

Test Data ID	Category Name	Category Desc	Image
TD6_1	Promotion	Raya Haji	Raya haji.png
TD6_2	blank	blank	blank

Table 6.20 Test Data of New Item Pizza

Test Data ID	Name	Description	Price	Category	Image
TD7_1	Matcha Latte Bobba	Liking good for taste of Japanese matcha	RM 9	Promotion	Matcha.png
TD7_2	Matcha Latte Bobba	Liking good for taste of Japanese matcha	RM9	Promotion	blank

Table 6.21 Test Data of Contact Us Form Page

Test Data ID	Password	Message
TD8_1	hajar123	Good service!
TD8_2	blank	blank
TD8_3	hajar	Good service!

6.5 Test Result and Analysis

Table 6.22 Test Result of Registration (User- Customer)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass /Fail
TC1_1	TD1_1	Sign up successful	Display message “Success! You can now login.”	Pass

TC1_2	TD1_2	Sign up failed	Display error message "Please fill up this field."	Pass
TC1_3	TD1_3	Sign up failed	Display error message "Please include an @ in the email address."	Pass
TC1_4	TD1_4	Sign up failed	Display error message "Warning! Username Already Exists."	Pass

Table 6.23 Test Result of User Login (User – Admin and Customer)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass /Fail
TC2_1	TD2_1	Login successful	Display message "Success! You are logged in."	Pass
TC2_2	TD2_2	Login failed	Display error message "Please fill out this field."	Pass
TC2_3	TD2_3	Login failed	Display error message "Incorrect Username or Password! Please Enter Again."	Pass

Table 6.24 Test Result of Pizza Menu Searching

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass /Fail

TC3_1	TD3_1	Searching successful	Display the searching pizza menu information based on the keyword entered	Pass
TC3_2	TD3_2	Searching failed	Display error message “No Result Found” with “Suggestions: • Make sure that all words are spelled correctly. • Try different keywords. • Try more general keywords.”	Pass

Table 6.25 Test Result of CheckOut Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass /Fail
TC4_1	TD4_1	CheckOut successful	Display message “Thanks for ordering with us. Your order id is ‘21’.”	Pass
TC4_2	TD4_2	CheckOut failed	Display error message “Please fill up this field.”	Pass
TC4_3	TD4_3	CheckOut failed	Display error message “Incorrect Password! Please enter correct Password.”	Pass
TC4_4	TD4_4	CheckOut failed	Display error message “Please match with requested format.”	Pass

Table 6.26 Test Result of Order Manage of Order Status and Delivery Details

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass /Fail

TC5_1	TD5_1	Order manage successful.	Display message " Update successfully"	Pass
TC5_2	TD5_2	Order manage failed.	Display error message "Please select an item in the list."	Pass

Table 6.27 Test Result of New Category Pizza

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass /Fail
TC6_1	TD6_1	Create new category successful.	Display message "New Category Has Been Added"	Pass
TC6_2	TD6_2	Create new category failed.	Display error message "Please fill out this field." for category name and category description while for error message for image that not inserted "Please select a file."	Pass

Table 6.28 Test Result of New Item Pizza

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass /Fail
TC7_1	TD7_1	Create new item successful.	Display message "New Item Has Been Added'	Pass
TC7_2	TD7_2	Create new item failed.	Display error message "Please fill out this field." for name, description and price and category display the error message " Please select an item in the list." while for error message for image that not inserted "Please select a file."	Pass

Table 6.29 Test Result of Contact Us Form Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass /Fail
TC8_1	TD8_1	Filled in successful.	Display message "Thanks for Contact us. Your contact id is 3. We will contact you very soon."	Pass
TC8_2	TD8_2	Filled in failed	Display error message "Please fill out this field"	Pass
TC8_3	TD8_3	Filled in failed	Display error message "Password Incorrect"	Pass

6.6 Conclusion

In conclusion, system testing is crucial phase in the part of DBLC. DBLC testing phase is used to help on checking and identifies error and defect in early stage. In POMS, are used on testing. Test Schedule, test description and test result are done in this chapter

CHAPTER 7 CONCLUSION

7.1 Introduction

This chapter will be discussing about the overall conclusion for Pizza Ordering Management System (POMS), which will consist of strength and weaknesses analysisof the system. Besides that, propositions on improvements according to the strength and weakness analysis, and contributions to this project will be defined and elaborate in this chapter as well.

7.2 Observation on Weakness and Strengths

Every system has its own advantages and disadvantages. An illustration in section 7.2.1 explained the strengths of the POMS, while section 7.2.2 was discussed for the weakness of this system.

7.2.1 Strengths

The strength of POMS are:

i. **Enhance Customer Experience**

POMS enhances the overall customer experience by offering a user-friendly interface that allows customers to easily navigate the menu, customize their orders, and choose preferred payment methods. The system's ability to provide order tracking and real-time updates on the status of the order, including preparation and delivery time, adds transparency and reliability to the service. Additionally, customers can save their preferences and past orders, making future ordering faster and more personalized.

ii. Efficient Data Management and Reporting

The POMS efficiently manages and stores all relevant data, including customer details, order history, and inventory levels, in a centralized database. This centralized data management allows for easy access and retrieval of information, aiding in better decision-making and customer service. The system also generates detailed reports on sales trends, popular menu items, and delivery performance, which helps management identify areas for improvement and optimize operations. These reports can be automatically generated and customized based on specific needs, saving time and ensuring accuracy.

7.2.2 Weakness

The weakness of POMS are:

i. Recovery and backup features

There are no automated data backup or recovery features on the system. When a system corrupts or a user unintentionally deletes data, the data stored within the system may not be recoverable.

ii. Forgot password function

This system does not provide forgot password function to allow the user get the new password by sending to their verify email.

iii. Lack of Automated Order Tracking Notifications

The POMS lacks an automated notification feature that updates customers on the status of their orders, such as preparation, dispatch, and delivery. Currently, customers must manually refresh the order tracking page to receive updates, which can be inconvenient and lead to frustration if they are unsure of the current status of their order. Without timely notifications via email, SMS, or in-app alerts, customers might feel uncertain about when their order will arrive, reducing the overall satisfaction with the service.

7.3 Propositions of Improvement

The following are statements of position for improving POMS based on strengths and weakness above.

i. **Automatic Backup and Recovery**

The POMS can be enhanced by implementing an automatic backup function. For instance, the system can be configured to perform automatic backups at a specific time, such as 12:00 AM daily, ensuring that all order data, customer details, and transaction records are securely backed up and up to date. The admin can customize the frequency of these backups to occur daily, weekly, or monthly, depending on the system's needs. In case of system corruption or failure, the system will automatically recover the most recent backup, minimizing data loss and ensuring business continuity.

ii. **Forgot Password Functionality**

To improve the user experience, POMS can introduce a forgot password function. If a customer forgets their password, the system can send a password reset link or a temporary password to their verified email address. Once they log in with the new or temporary password, the system will prompt them to reset their password to something more secure. This functionality would improve account security and ease of access, ensuring users can quickly regain access to their accounts without needing to contact customer support.

iii. **Order Status Notifications**

POMS can be improved by incorporating automated order status notifications. Instead of requiring customers to manually refresh the tracking page, the system can send real-time updates via email, SMS, or in-app notifications at key stages of the order process. For example, customers would receive a notification when their order is confirmed, when it is being prepared, when it is out for delivery, and upon completion. This improvement would enhance customer experience by keeping them informed and reducing uncertainty about their order's status.

7.4 Contribution

The Pizza Ordering Management System (POMS) is primarily designed to streamline and manage the pizza ordering process more effectively. Firstly, POMS plays a crucial role in handling the complexity of order management by organizing and processing customer orders systematically and efficiently. Automated triggers are implemented in the system to update order statuses as they progress through various stages, such as preparation and delivery. This ensures that customers have accurate and up-to-date information on the status of each order.

Secondly, POMS will automatically calculate the total cost of an order, including taxes, discounts, and delivery fees, and store this information in the database once the order is placed. This function provides accurate billing information to both the customers and the admin management, ensuring transparency and aiding in the financial tracking of sales. Additionally, this automation reduces the likelihood of human error, resulting in a more reliable and efficient ordering process.

7.5 Conclusion

In conclusion, the objectives and scope outlined in Chapter 1 of this project have been successfully achieved. The primary objectives of the POMS were to replace the traditional manual ordering process with an efficient, web-based system; reduce human errors in order processing and billing; streamline the search for customer information; and generate various types of reports. The Database Life Cycle (DBLC) methodology was chosen for development, as it provides a comprehensive approach covering the initial database study, database design, implementation and loading, training and evaluation, as well as operations, maintenance, and evaluation.

The Testing Phase of POMS was conducted to identify and rectify any bugs or defects in the system. Ultimately, POMS successfully met both the functional and non-functional requirements set out at the beginning of the project. However, some weaknesses were identified in certain areas of the system, which should be addressed in future iterations to enhance performance and user satisfaction. Overall, this project has been successfully completed and meets the requirements for the Bachelor of Computer Science (Database Management). Furthermore, the contributions of this project can serve as a foundation for future improvements in pizza ordering systems.

REFERENCES

Computers Professor. (2019). Explain Database Life Cycle DBLC. Retrieved from http://www.computersprofessor.com/2019/01/explain-database-life-cycle-dblc.html

Devaraj, K (2023). Levels of Testing: A Complete Approach to Quality Assurance. Retrieved from https://testsigma.com/blog/levels-of-testing/

GeeksforGeeks (2020). DFD for Food Ordering System. Retrieved from https://www.geeksforgeeks.org/dfd-for-food-ordering-system/

GeeksforGeeks (2024). Difference between Black Box and White Box Testing. Retrieved from https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/

Jackson, K. (2022). Last Call Pizza Information System.

JHCSC (2016). Database Life Cycle (DBLC) and System Development Life Cycle (SDLC). Retrieved from https://jhcscset.wordpress.com/about/

LovelyCoding.org (2022).Online Pizza Ordering System, Final Year Project – LovelyCodin. Retrieved from https://www.lovelycoding.org/online-pizza-ordering-system-project-for-finalyear/?srsltid=AfmBOopZxhpHRGBZWDI2GiS7mJTqnouwkOiIUt973IQkSsM5omL6dLpl

Tutorialspoint.com. (2011.). Software Testing Quick Guide. Retrieved from https://www.tutorialspoint.com/software_testing/software_testing_quick_guide

Sotnik, S., Manakov, V., & Lyashenko, V. (2023). Overview: PHP and MySQL features for creating modern web projects.

What is User Acceptance Testing (UAT)? with Examples. (n.d.). Retrieved from https://www.guru99.com/user-acceptance-testing.html

APPENDIX A

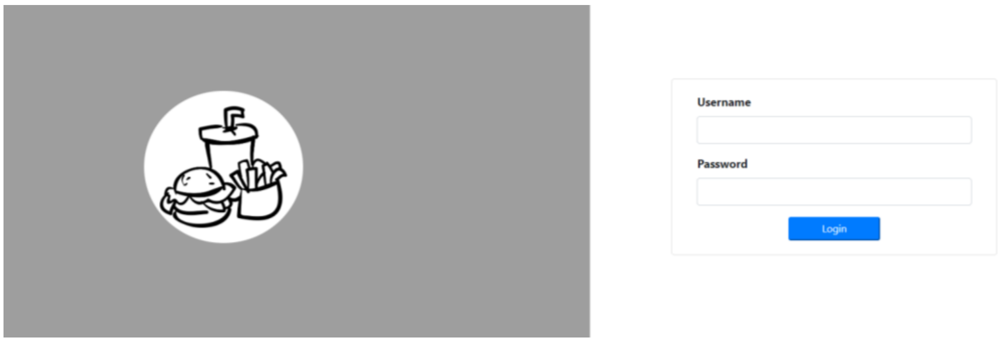
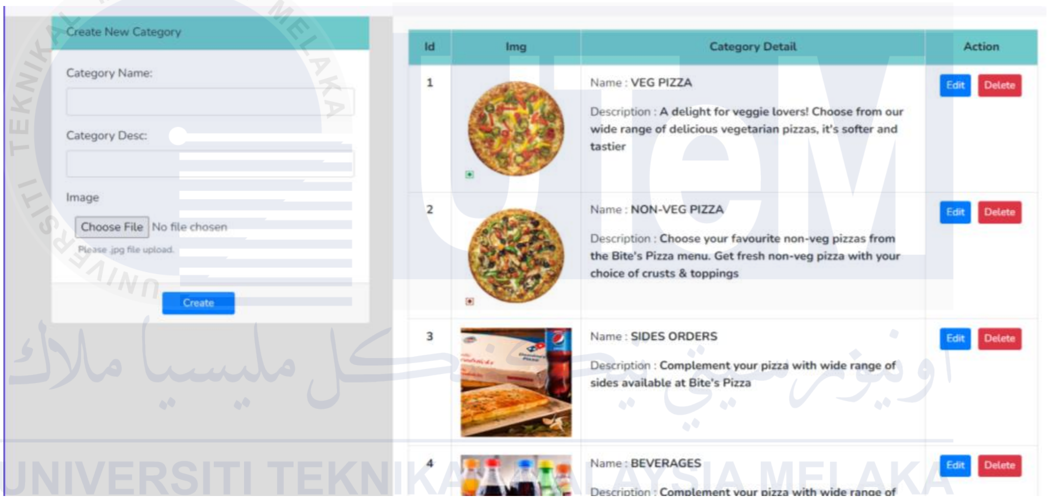
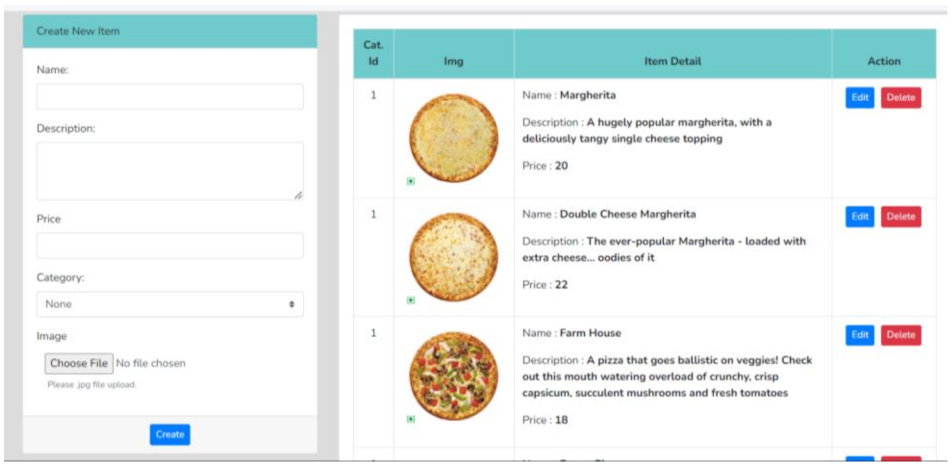


Figure 6.2 Login Page for Administrator Role



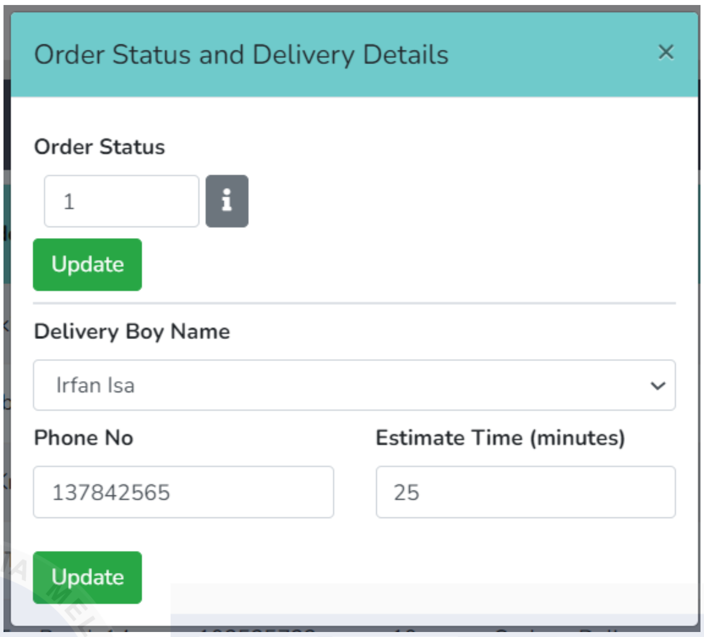
Id	Img	Category Detail	Action
1		Name : VEG PIZZA Description : A delight for veggie lovers! Choose from our wide range of delicious vegetarian pizzas, it's softer and tastier	Edit Delete
2		Name : NON-VEG PIZZA Description : Choose your favourite non-veg pizzas from the Bite's Pizza menu. Get fresh non-veg pizza with your choice of crusts & toppings	Edit Delete
3		Name : SIDES ORDERS Description : Complement your pizza with wide range of sides available at Bite's Pizza	Edit Delete
4		Name : BEVERAGES Description : Complement your pizza with wide range of	Edit Delete

Figure 6.3 Create New Category Pizza Form



Cat. Id	Img	Item Detail	Action
1		Name : Margherita Description : A hugely popular margherita, with a deliciously tangy single cheese topping Price : 20	Edit Delete
1		Name : Double Cheese Margherita Description : The ever-popular Margherita - loaded with extra cheese... oodies of it Price : 22	Edit Delete
1		Name : Farm House Description : A pizza that goes ballistic on veggies! Check out this mouth watering overload of crunchy, crisp capsicum, succulent mushrooms and fresh tomatoes Price : 18	Edit Delete

Figure 6.4 Create New Item Pizza Form



Order Status and Delivery Details

Order Status

1

Update

Delivery Boy Name

Irfan Isa

Phone No

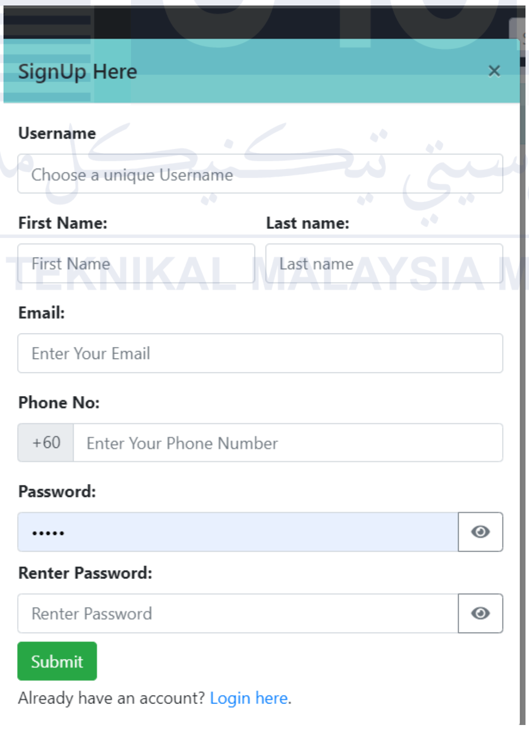
137842565

Estimate Time (minutes)

25

Update

Figure 6.5 Order Manage of Status Order and Delivery Details



SignUp Here

Username

Choose a unique Username

First Name:

First Name

Last name:

Last name

Email:

Enter Your Email

Phone No:

+60 Enter Your Phone Number

Password:

•••••

Renter Password:

Renter Password

Submit

Already have an account? Login here.

Figure 6.6 Sign Up Page Form for New Customer

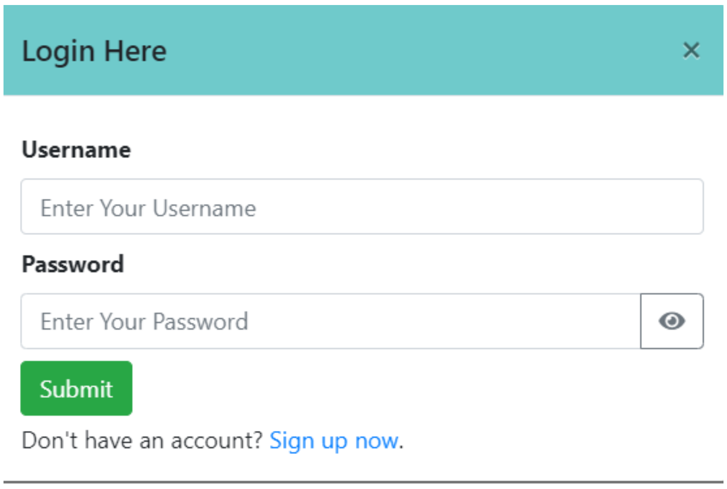


Figure 6.7 Login Page for Customer Role

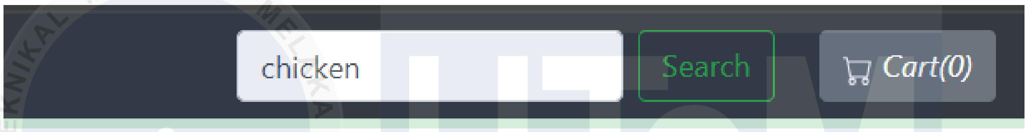


Figure 6.8 Pizza Menu Searching

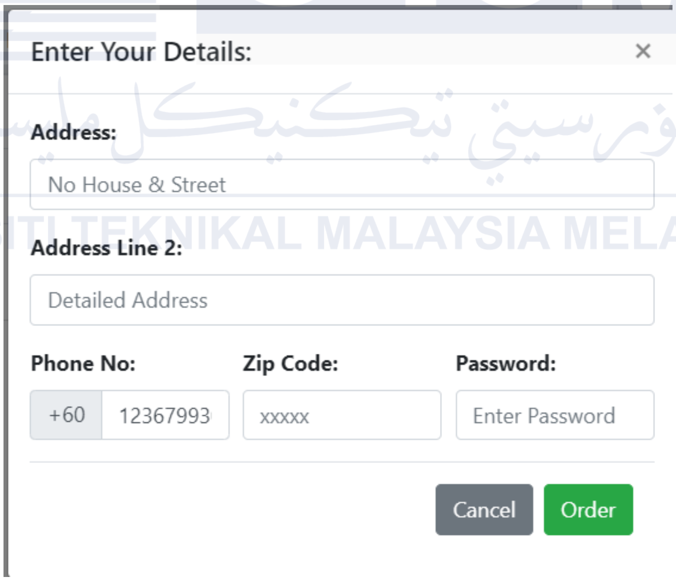
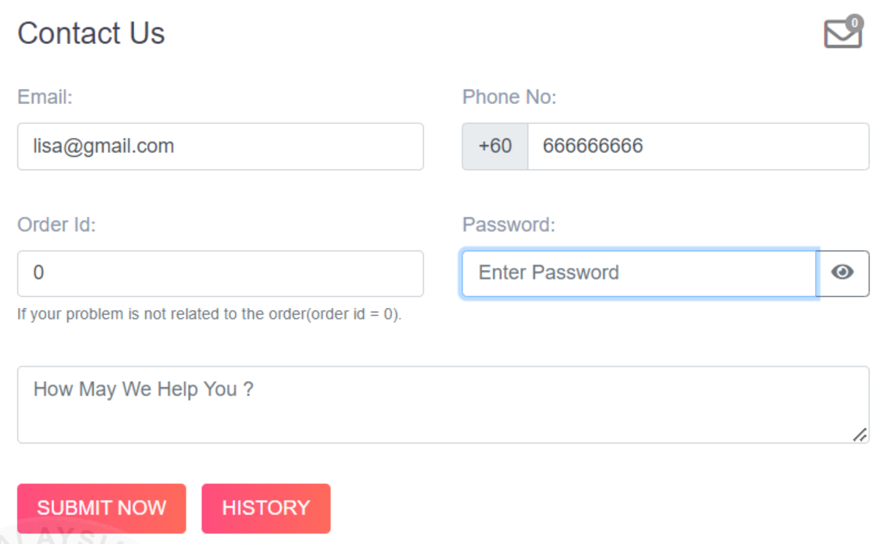


Figure 6.9 CheckOut Page


Contact Us

Email: lisa@gmail.com	Phone No: +60 666666666
Order Id: 0	Password: Enter Password

If your problem is not related to the order(order id = 0).

How May We Help You ?

Figure 6.10 Contact Us Form Page

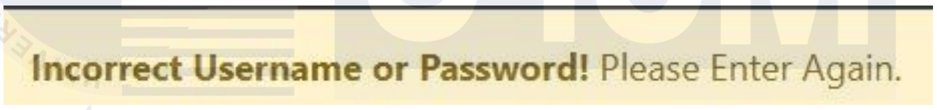
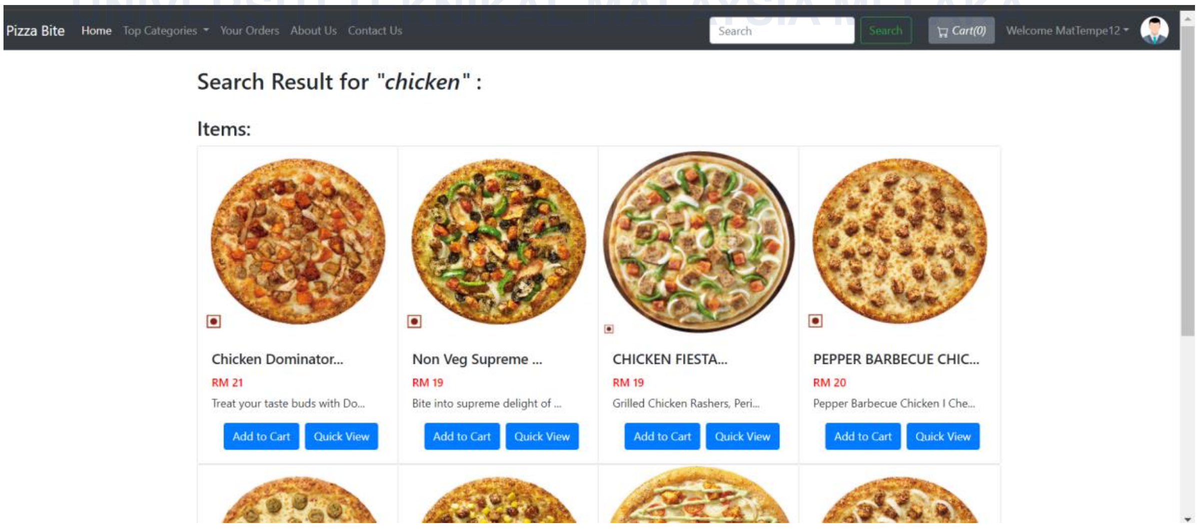

Incorrect Username or Password! Please Enter Again.

Figure 6.11 Login Failed Error Message

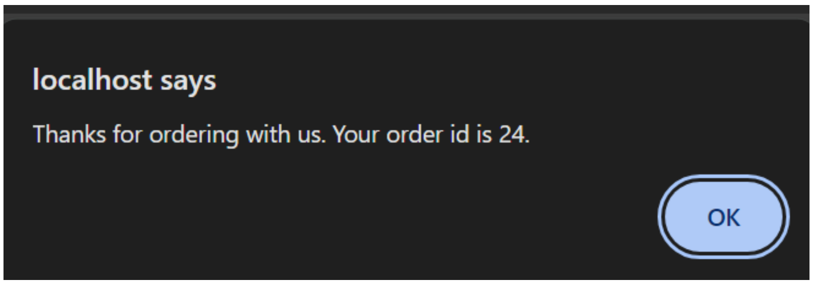


Search Result for "*chicken*" :

Items:

Chicken Dominator...	Non Veg Supreme ...	CHICKEN FIESTA...	PEPPER BARBECUE CHIC...
RM 21	RM 19	RM 19	RM 20
Treat your taste buds with Do...	Bite into supreme delight of ...	Grilled Chicken Rashers, Peri...	Pepper Barbecue Chicken I Che...
Add to Cart Quick View	Add to Cart Quick View	Add to Cart Quick View	Add to Cart Quick View

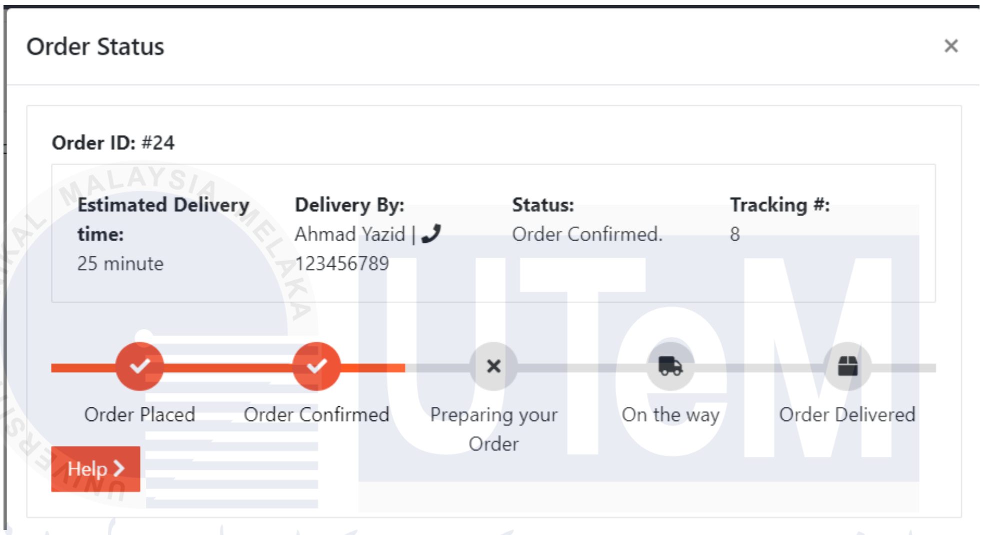
Figure 6.12 Result of Searching Pizza Menu



localhost says

Thanks for ordering with us. Your order id is 24.

Figure 6.13 CheckOut Success With Order ID Message

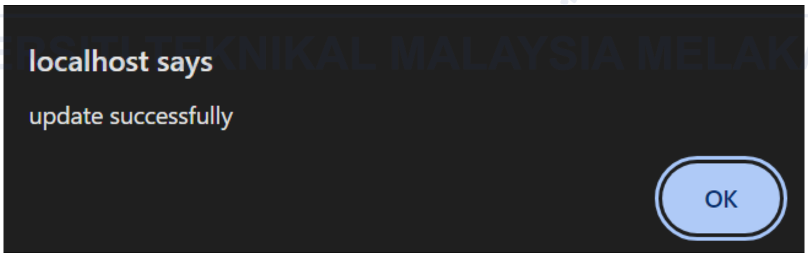


Order Status

Order ID: #24

Estimated Delivery time:	Delivery By:	Status:	Tracking #:
25 minute	Ahmad Yazid | 123456789	Order Confirmed.	8

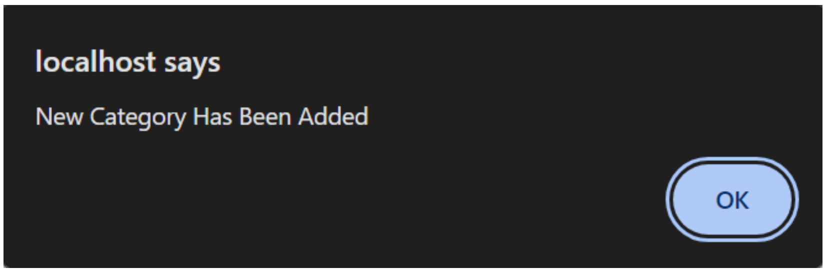
Figure 6.14 Order Status Result from Customer Account



localhost says

update successfully

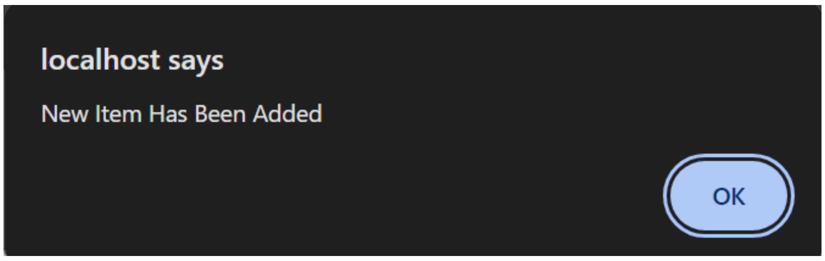
Figure 6.15 Update Order Status Success Message



localhost says

New Category Has Been Added

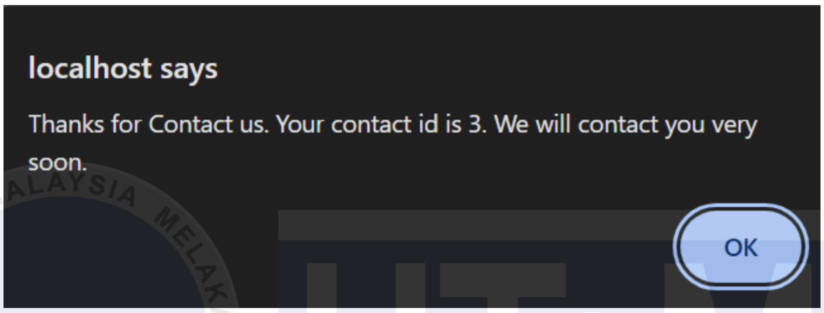
Figure 6.16 New Category Added Message



localhost says

New Item Has Been Added

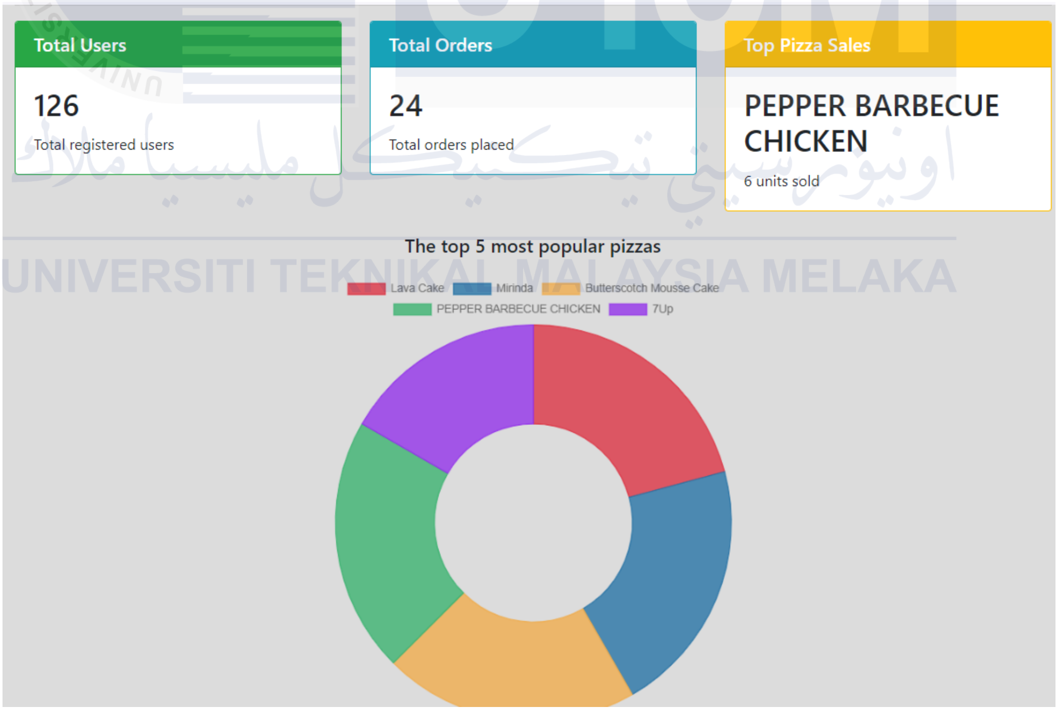
Figure 6.17 New Item Added Message



localhost says

Thanks for Contact us. Your contact id is 3. We will contact you very soon.

Figure 6.18 Filled In Succes of Contact Us Message



Total Users

126

Total registered users

Total Orders

24

Total orders placed

Top Pizza Sales

PEPPER BARBECUE CHICKEN

6 units sold

The top 5 most popular pizzas

Figure 6.19 POMS Report