BETTA FARM MANAGEMENT SYSTEM

FARID HAZEEQ BIN BURHANUDDIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BETTA FARM MANAGEMENT SYSTEM

FARID HAZEEQ BIN BURHANUDDIN

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

BETTA FARM MANAGEMENT SYSTEM

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

STUDENT : _____ Date : 31/08/2021

(FARID HAZEEQ BIN BURHANUDDIN)

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 (Software Development) with Honours.

SUPERVISOR : _____ Date : 12/9/2021

(ANIZA BINTI OTHMAN)

DEDICATION

I dedicate this project to Allah SWT my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength throughout this semester. I also dedicate this work to my parents; Burhanuddin Bin Zainal and Sharifah Zainab Binti Ahmad Zambri who has encouraged me all the way and whose encouragement has made sure that I give it all it takes to finish that which I have started. Furthermore, I also dedicate this work to my supervisor, Mrs. Aniza Binti Othman who have been giving me guidance throughout my project. Thank you. My love for you all can never be quantified. May Allah bless you.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful

Alhamdulillah , all praises to Allah SWT for the strengths and His blessings in completing this project. I would like to reflect on the people who have supported and helped me so much throughout this period

Firstly, I would like to express my sincere appreciation to my supervisor, Mrs. Aniza Binti Othman for the continuous support of my project, for her guidance, encouragement, patience and immense knowledge during research process and coding writing. Without her valuable assistance, this work might not have completed well.

Secondly, my sincere thank you to the examiner of my proposal and final year project, Professor Dr. Mohd Khanapi Bin Abd Ghani for the support and suggestions pertaining to the improvement of my thesis.

I also would like to express gratitude to all lectures of FTMK for their willingness in participating in my research.

Sincere thanks to all my friends especially my course mate, Ahmad Zaim Bin Zulkifli and my classmates for their help and encouragement.

Last but not the least, it is my utmost pleasure to dedicate this work to my dear parents; Mr. Burhanuddin Bin Zainal and Mrs. Sharifah Zainab Binti Ahmad Zambri and to my siblings, for their endless love, prayers and their unwavering belief in my ability to accomplish my goal.

May Allah grant the best rewards for all of you.

Farid Hazeeq Bin Burhanuddin.

ABSTRACT

Betta Farm Management System is a management system to support the management of betta, record the order history for breeders and users. In the current situation of the management, the order history is recorded manually by using hand. All the information relate to the betta and the order is recorded in a logbook or papers. Through this method the management does not have an effective information management for the order history. It is also hard for the management to track the order history. The objective of this project is to develop and design an information system that able to store the order history, betta information and users information. The scope of the project covers a few modules, which are login module, administrator module and user module. The significance of the project is to help the management in storing information relate to the betta with higher efficiency and less time consuming.

ABSTRAK

Sistem Pengurusan Ladang Betta adalah sistem pengurusan untuk menyokong pengurusan betta, mencatat sejarah pesanan untuk penternak dan pengguna. Biar kita ambil contoh dimana pada ketika ini semua pengurusan, sejarah pesanan direkodkan secara manual dengan menggunakan tangan. Semua maklumat berkaitan dengan betta dan pesanan dicatatkan dalam buku log atau kertas. Melalui kaedah ini pihak pengurusan tidak mempunyai pengurusan maklumat yang berkesan untuk sejarah pesanan dan ia juga sukar bagi pihak pengurusan untuk mengesan sejarah pesanan. Objektif projek ini adalah untuk membangun dan merancang sistem maklumat yang dapat menyimpan sejarah pesanan, maklumat betta dan maklumat pengguna. Skop projek merangkumi beberapa modul, iaitu modul log masuk, modul pentadbir dan modul pengguna. Kepentingan projek ini adalah untuk membantu pihak pengurusan dalam menyimpan maklumat berkaitan dengan betta dengan kecekapan yang lebih tinggi dan memakan masa yang lebih sedikit.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

BFMS	-	BETTA FARM MANAGEMENT SYSTEM
SDLC	-	SOFTWARE DEVELOPMENT LIFE CYCLE
TB	-	TERABYTE
GB	-	GIGABYTE
FR	-	FUNCTIONAL REQUIREMENT
SQL	-	STRUCTURED QUERY LANGUAGE
GUI	-	GRAPHICAL USER INTERFACE
HTML	-	HYPERTEXT MARKUP LANGUAGE
DDL	-	DATA DEFINITION LANGUAGE
ERD	-	ENTITY RELATIONSHIP DIAGRAM
CD	-	CONTEXT DIAGRAM
DFD	-	DATA FLOW DIAGRAM

CHAPTER 1: INTRODUCTION

1.1 Introduction

Betta Farm Management System(BFMS) is a management system that support the management of the farm. Nowadays many breeders still use manual method like recorded manually by using hand. All the information relate to the customer and the order is recorded in a logbook or papers. Via this approach the management does not have an appropriate information management for the order history. It is also hard for the management to track the order history. The objective of this project is to develop and design an information system that able to store user details, user's order and betta information. The scope of the project covers a few modules, which are login module, administrator module and user module.

1.2 Problem Statement

1) Takes a lot of time to fill records by using hand.

   All the records a save by using hand and it can It can occur in handwritten manuscripts as a result of forgetfulness or an unintentional stroke of the pen.

2) The records are poorly managed.

   All the information relate to the betta and the order is recorded in a logbook or papers which can be damaged and lost easily through human error and natural disasters.

3) Waste of time to find the records when are needed.

   It is hard for the management to track the order history because all the orders history has been save on paper.

1.3 Objective

1) To develop and design management system that able to store the order details.

2) To make system that can asses the order's record instantly

3) To make system that can generate the order report.

1.4 Scope

1.4.1 Modules to Be Develop

1.4.1.1 Login Module

To authenticate user to access the system by logging in using username and password to administrator page or user page.

1.4.1.2 Administrator Module

(a) Product – Admin can add new product and admin has right to delete the product.

(b) Customer - Admin can add new customer and view details of the customer, admin has the right to updates the detail and delete the user.

(c) Order - Admin can view all orders from the customer and print the orders report.

1.4.1.3 User Module

(a) Product – User can view details of the product, customer has no right to updates the detail and delete the product.

(b) Order - User can add new order and view details of the order.

1.4.2 Target User

(a) Administrator

(b) User

1.5 Project Significance

The significance of the project is to help the management in storing information relate to the product and user with higher efficiency and less time consuming. The responsible admin also capable to maintain the information easily.

Furthermore, the system will also improve the management access to retrieve all order history or relevant data. With the data, they can save any order easily.

1.6 Expected Output

The expected outcome of the project is a system that are able to help a Betta's farmer to manage the business easily. The admin can use all the functions provided by the system. The customer can easily make an order for the Betta. This will eliminates manual recording with a digitalized system.

1.7 Conclusion

Finally, the BFMS allows management to manage their orders in an efficient and convenient manner. Administrators can view information about each order's income and sales. As a result, they are able to determine the number of orders that will be distributed based on the information provided.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

A literature review is a summary of existing research on the topic of your investigation. This synthesis brings together information from a variety of sources to provide a more comprehensive understanding of the topic, laying the groundwork for both the research question and primary research. Because it will cite sources and should discuss the sources' reliability, a literature review differs from an annotated bibliography. A methodology is a framework that project managers use to create, plan, implement, and achieve project goals. There are a variety of project management approaches that can be applied to a variety of projects.

2.2 Facts and Findings

2.2.1 Domain

The domain of Betta Farm Management System(BFMS) is web development. The task of developing, maintaining, and updating a website is known as web development. Web development also entails the creation of front-end pages as well as back-end server-side code that connects the web system to the database and the system's front end. Website has been implemented in many ways, such as advertisement, e-commerce, management and many more. Implementing the project as web development enable the system to be used in multiplatform with the terms of the device has an internet connection and able to use the web browser.

2.2.2 Existing System

GHV Betta Farm is an betta farm for Selangor state that situated in Kuala Selangor. It sells betta, food pellet, betta aid, and care equipment. Normally, all customers must come to the farm to make an order and purchase of betta and they also like to ask the price of the products and what item should use to taking care of their betta. Since the pandemic, all order must be made through phone only.

Currently, GHV Betta Farm still does not have any computerized system. All operation done manually and important information's are recorded by written. All records are kept in the paper and logbook.

The workflow in GHV Betta Farm starts with customer ask the availability of the product and then the person in charge will go through from logbook, fish rack and item cabinet to check their availability.

If the customer want to see the new stock of the product, they must come to the farm by themselves which is hard during this pandemic.

Currently, the person in charge faces many problems using this manual system. The problem occurred when the staff needs to open all records and find the answers for the customer requirement. Sometimes, the information needed is not included in the logbooks. The staff has to find it in the fish racks or item cabinet. All these things take a long time. The staff could give the wrong information.

The way to record the new information is not well managed. Sometimes, the staff just write down anywhere in the logbooks. It is a big problem if the handwriting cannot be read. To find data through the logbooks are not easy, the staff needs the patience to do it.

All information of the operations of the manual system are recorded in paper. The possibility for the data to be lost or damaged is very high. Old records might be already lost.

2.2.3 Technique

The technique used to gather information for this project is user observation. User observation aids the analyst by providing a firsthand understanding of how the user interacts with the system. The analyst can observe the user and how their surroundings affect their interaction with the system when the goal is to improve a task. User observation can also be useful for validating previously collected data. It could be in situations where users provide false information or are unable to recall all of their tasks when using the system.

2.3 Project Methodology

In software development, there are a chain of activities need to be completed which known as System Development Life Cycle (SDLC). For system development, many organizations employ the SDLC methodology, which consists of several phases that track the progress of the systems analysis and design effort.

The Agile methodology has been used in development this project. Agile is currently the most effective method for managing project development. In agile software development, there are four core values. Individual and team interactions are prioritized over processes and tools. Working software takes precedence over thorough documentation, and customer collaboration takes precedence over contract negotiations. Finally, responding to change over according to a strategic plan. The advantages of this method are that the customer has an early and regular opportunity to inspect the goods and make decisions about the project. The iterative process governs the Agile software development lifecycle. Each iteration progresses to the next stage of software development by incorporating useful elements such as client documents until the project is completed, and each iteration has a deadline for completion.

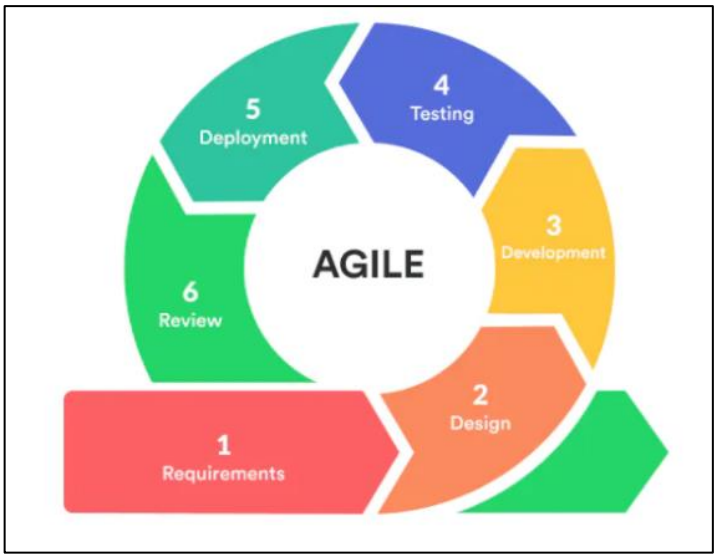

Figure 2.1 Agile Methodology Phase

There are six phase using the agile methodology to complete this project. Firstly, the requirements phase are for the project teams to identify the requirements for the system development. It would focus on the need for a web-based framework as well as the end-user analysis required to develop the system based on the requirements. It can be done using the problem statement that the project teams discovered, as well as the project's objective goal, to determine the deliverables that must be completed and to simplify the objectives into a single purpose. Next, the design phase is when the project teams focus on web-based system architecture, the database design and interface. The requirements gathered in the first phase will serve as a guide in design phase for properly designing the system and focusing on how to provide the required functionality.

Thirdly, on the development phase is begin when a project teams start writing code and converting design documentation into the actual software. This is the most time-consuming stage of the SDLC because it is the foundation of the entire process. After that, the testing phase is for the development teams to ensure that the software is bug-free and compatible with everything else they've written previously.

Fifthly, the deployment and implementation are when the working iteration must be integrated and delivered into production by project teams. The system will be deploy on the servers and made available to users either for demonstration or actual

use. Lastly, once all of the previous stages of development have been completed the project teams are gathered again by the stakeholders, who review the progress made toward completing the requirements. The team presents their ideas for resolving the issues that arose during the previous phases, and the stakeholders consider their suggestions. After that, either with a new iteration or by progressing to the next stage, the Agile software development lifecycle phases begin anew.

2.4 Project Requirements

2.4.1 Software Requirement

A software requirement is a list of software programs or hardware devices that are needed to run the system.

1) Microsoft Visual Studio Code

   Microsoft Visual Studio Code is a reimagined and optimized code editor for developing and debugging modern web and cloud applications.

2) XAMPP

   XAMPP stands for cross-platform, Apache, MySQL, PHP, and Perl, and it allows you to create a WordPress site on your computer's local web server.

3) Microsoft Windows Operating System

   Microsoft Windows Operating System is software that manages computer hardware and software resources while also providing common services to computer programs.

4) Google Chrome

   Google Chrome is software that allows you to access the World Wide Web.

2.4.2 Hardware Requirement

Hardware requirements are the hardware devices needed to develop this system. The hardware used in this project is listed below.

1) Laptop - Asus ROG GL552VW

2) CPU - Intel Core i7 6th Gen 6700HQ

3) Hard Drive - 1TB

4) Memory - 16GB

5) Screen Resolution - 1920x1080 pixels

2.4.3 Other Requirements

Table 2.1 Other Requirement

Description	Tools
Image editing	Adobe Photoshop 2020
Communication Application	Discord

2.5 Project Schedule and Milestone

A project schedule and milestones is a timetable that organizes project tasks and milestones. During the project development, project scheduling is used to keep track of resources and deliverables. It is necessary for completing projects on time and on budget.

2.5.1 Milestones

The project starts at 15/3/2021 and there are milestones that need to be completed before the due date.

Table 2.2 Milestones of BTMS project

No	Milestone	Start date	Due Date
1	Start the proposal documentation	15/3/2021	25/3/2021
2	Submit project proposal	15/3/2021	26/3/2021
3	Begin project introduction documentation	27/3/2021	7/4/2021
4	Project development start	7/3/2021	20/6/2021
5	Literature review documentation	8/4/2021	18/4/2021
6	Project methodology documentation	19/4/2021	25/4/2021
7	Project analysis documentation	26/4/2021	2/5/2021
8	Project design documentation	3/5/2021	23/5/2021
9	Project implementation documentation	17/5/2021	20/6/2021
10	Final project demo	21/6/2021	27/6/2021

2.5.2 Gantt Chart

TASK	WEEK													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Proposal														
Project introduction														
Literature review														
Project methodology														
Project Analysis														
Project Design														
Project Implementation														

Figure 2.2 Gantt Chart of BFMS project

2.6 Conclusion

This chapter concludes with a discussion of the project methodology. To develop a system, the methodology is based on the Software Development Life Cycle (SDLC). Furthermore, because database monitoring, modification, and maintenance are all part of the life cycle, and these activities continue long after a database has been implemented, the SDLC never ends.

CHAPTER 3: ANALYSIS

3.1 Introduction

The analysis stage entails a thorough examination of the prior software development planning. To analyze the effectiveness associated with the planned database system, the developer evaluates the database development plan against elements such as cost, time-period, development platform, programming languages, and forecasted development results. This phase involves analyzing the current system and the system to be developed.

3.2 Problem Analysis

3.2.1 Current Business Workflow

The current system lacks a few functionalities such as online betta management, order report generation and online order management because during this pandemic all the order a made over the phone only.

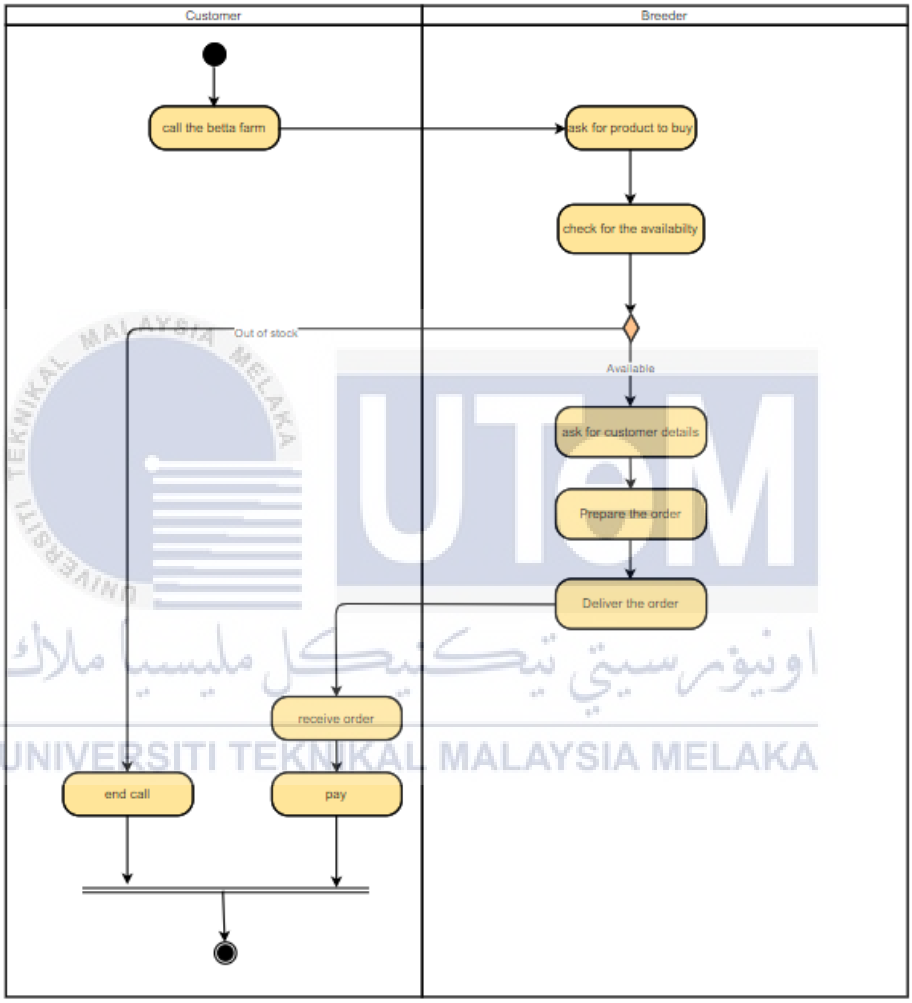

Figure 3.1 The Activity Diagram for The Current System

3.2.2 Problem of Current System

All transaction like ordering and purchasing betta is still done through on the phone or meetup. Sometimes, when customer want to order the betta, the person in charge have to manually check the availability of the betta which is really time consuming.

During the ordering process, person in charge do not know whether the betta is available or not. Person in charge need to count manually the total order and they do not have a proper system to manage their system.

3.2.3 To-Be Analysis

A new system is proposed to improve some of the current system's functionality. The system allows the person in charge to check the availability of betta directly. The person in charge and the user can use the system to obtain order information. The proposed system will display all customers order on a digital report.

3.2.4 Context Diagram

The figure shows how the system interact with external entities.

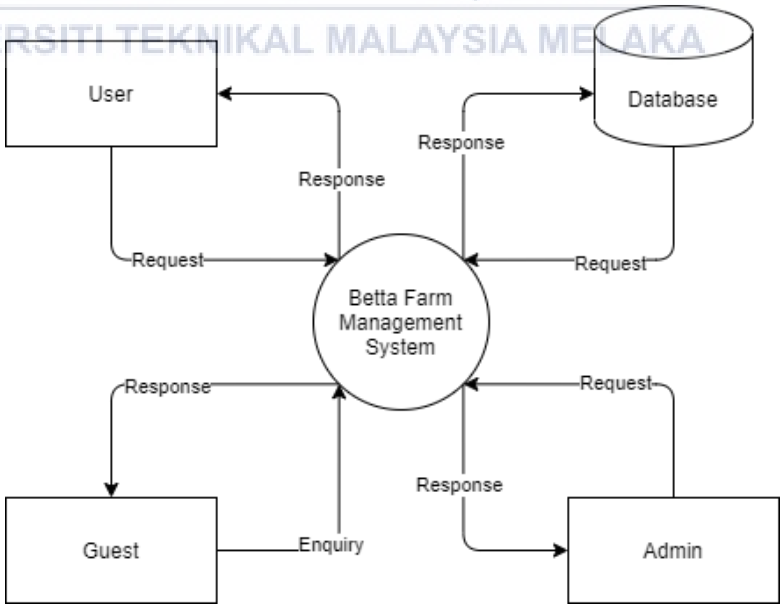

Figure 3.2 Proposed System Context Diagram

3.2.5 Data Flow Diagram

This figure shows data flow of main process on admin side and user side in the proposed Betta Farm Management System(BFMS).

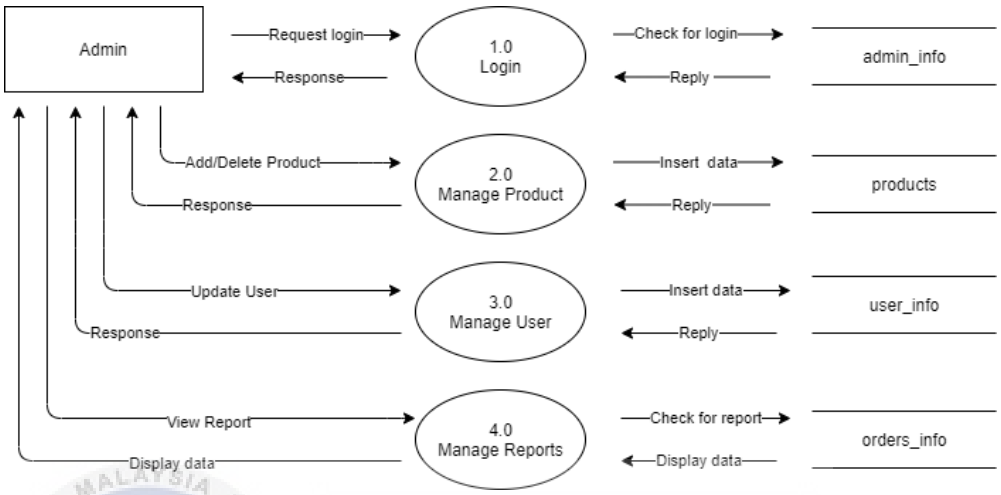

Figure 3.3 The 1st Level Admin Data Flow Diagram

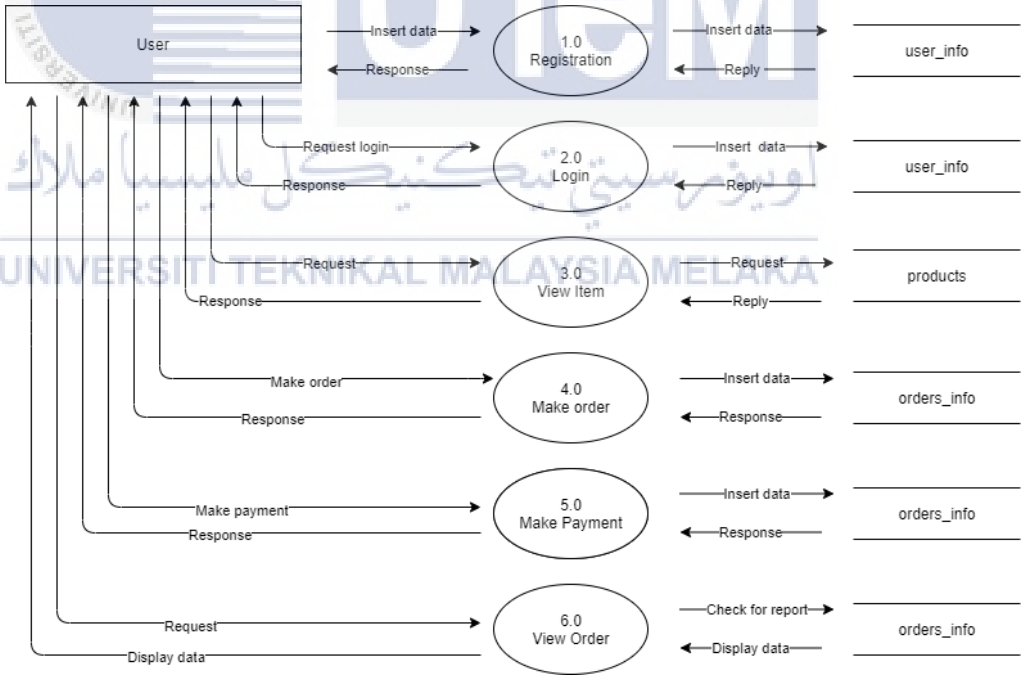

Figure 3.4 The 1st Level User Data Flow Diagram

3.3 Requirement analysis

3.3.1 Data Requirement

Admin information, brands, cart, categories, order information, order product information, products, and user information are all data requirements for this project. The data dictionary for Betta Farm Management System is shown in the table below.

Table 3.1 Data Dictionary

	Attribute Name	Data Type	Length	Constraint	Description
Admin Info	admin_id	int	10	Primary key	Admin ID will be auto increment
	admin_name	varchar	100	No default	Admin name
	admin_email	varchar	300	No default	Admin email
	admin_password	varchar	300	No default	Admin password
Brands	brand_id	int	100	Primary key	Brand ID will be auto increment
	brand_title	text		No default	Brand title
Cart	id	int	10	Primary key	Cart ID will be auto increment
	p_id	int	10	No default	Product ID
	ip_add	varchar	250	No default	Add product to cart
	user_id	int	10	Null	User ID
	qty	int	10	No default	Quantity of the product that has been selected
Categories	cat_id	int	100	Primary key	Category ID
	cat_title	text		No default	Category title
Order Info	order_id	int	10	Primary key	Order ID will be auto increment

	user_id	int	11	Foreign key	User ID
	f_name	varchar	255	No Default	User name
	email	varchar	255	No default	User email
	address	varchar	255	No Default	User address
	city	varchar	255	No default	User city
	state	varchar	255	No Default	User state
	zip	int	10	No default	User zip
	cardname	varchar	255	No Default	Name on card
	carnumber	varchar	20	No default	Card number
	expdate	varchar	255	No Default	Card expired date
	prod_count	int	15	Null	Total product to make order
	total_amt	int	15	Null	Total amount to pay
	cvv	int	5	No Default	Card CCV number
	date_order	timestamp		No default	Date of order
Order Product	order_pro_id	int	10	Primary key	Order product ID
	order_id	int	11	Foreign key	Order ID
	product_id	int	11	Foreign key	Product ID
	qty	int	15	Null	Quantity product of order
	amt	int	15	Null	Amount product of order
Products	product_id	int	100	Primary key	Product ID will be auto increment
	product_cat	int	100	No default	Product category
	product_brand	int	100	No default	Product brand
	product_title	varchar	255	No default	Product title
	product_price	int	100	No default	Product price
	product_desc	text		No default	Product description

	product_image	text		No default	Product image
	product_image2	varchar	200	No default	Product image
	product_image3	varchar	200	No default	Product image
	product_image4	varchar	200	No default	Product image
	product_image5	varchar	200	No default	Product image
	product_keywords	text		No default	Product keywords
User Info	user_id	int	10	Primary key	User ID will be auto increment
	first_name	varchar	10	No default	User first name
	last_name	varchar	10	No default	User last name
	email	varchar	10	No default	User email
	password	varchar	10	No default	User password
	mobile	varchar	10	No default	User phone number
	address	varchar	10	No default	User address
	address2	varchar	10	No default	User address

3.3.2 Functional Requirement

This section will define high-level requirements and features of the proposed system. It focuses on the capabilities of the system as required by the stakeholders and the target users.

Table 3.2 Functional Requirement Table

FR No	Requirement	Description	Phase
BFMS 1.1	Login	The system shall enable user to login into the system by enter valid user identification and password	
BFMS 1.2	Logout	The system shall enable user to logout from the system at anytime	
BFMS 2.1	Register	The system shall allow user to register for their account	

BFMS 2.2		The system shall enable user to click button “Sign Up” to insert all detail in the database	
BFMS 3.1	Manage product availability	The system shall enable administrator to manage the products	
BFMS 3.2		The system shall enable to add new product	
BFMS 3.3		The system shall enable administrator to insert data manually	
BFMS 4.1	View data	The system shall enable user to view user profile	
BFMS 4.2		The system shall enable to view all products	
BFMS 4.3		The system shall enable user to view order details	
BFMS 4.4		The system shall enable administrator to view order details and report	

3.3.2.1 Use Case Diagram

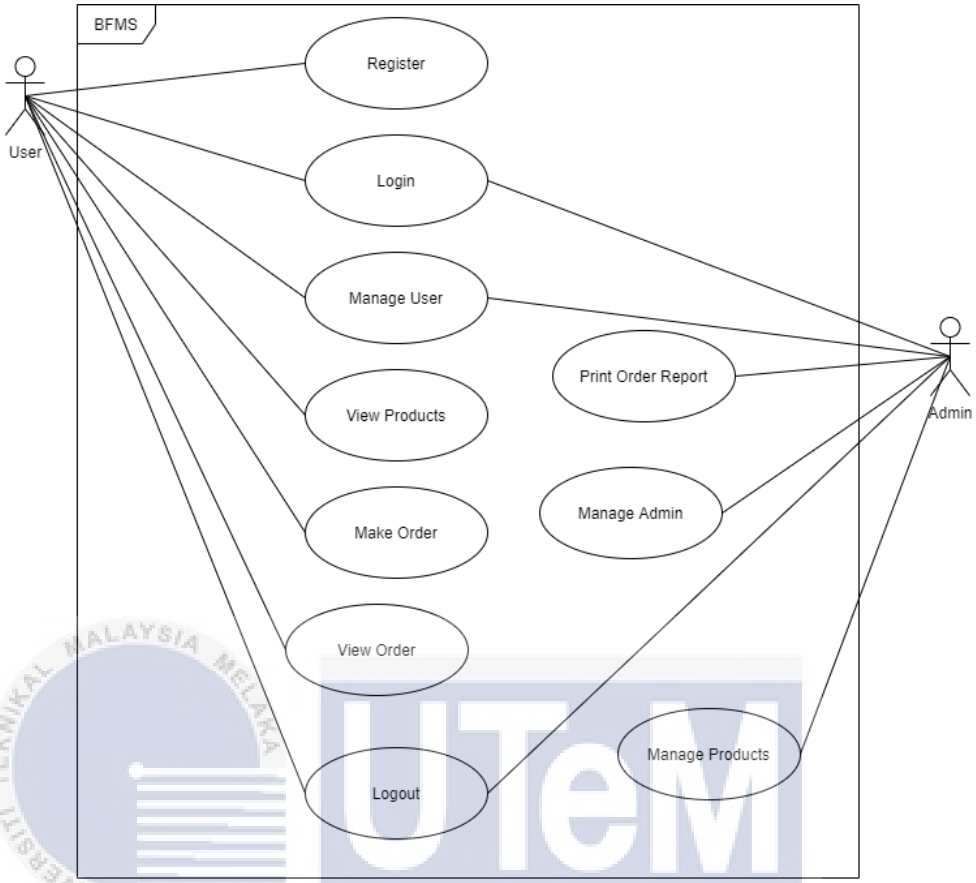

Figure 3.5 Use Case Diagram for Betta Farm Management System

The Figure 3.5 above shows the use case diagram for Betta Farm Management System(BFMS). There are two user of this system which is administrator and user. Administrator is a person who will handle the system by manage the products for the user while user has right to use overall of the system except manage the products.

3.3.2.2 Data Flow Diagram

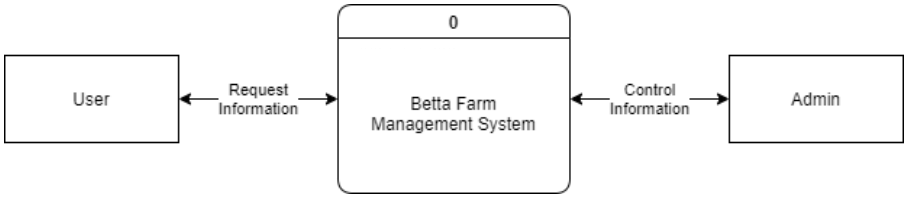

Figure 3.6 The 0 Level Data Flow Diagram for Betta Farm Management System

Figure 3.6 shows the 0 level data flow diagram of Betta Farm Management System(BFMS). The context diagram includes the high-level process of both user which are user and admin.

3.3.3 Non Functional Requirement

Non-functional requirements talk about the performance and characteristics of the requirement and what the system should be able to provide. This aid in measuring specific performance must be used without affecting the entire system and must improve the usability.

3.3.3.1 Performance Requirement

The system is designed to load the initial page in less than 3 seconds. It should also be ensured that the system will not obstruct user input.

3.3.3.2 Maintainability

To ensure the quality of the services provided, the system should be maintained on a regular basis.

3.3.3.3 Reliability

The system should be able to cope with a large number of people attempting to use it at the same time.

3.3.4 Other Requirement

The software and hardware requirements that are involved in technical justification are discussed in this section.

3.3.4.1 Software Requirement

To develop the BFMS, some of the software used is identified. The software required for developing system application are listed below.

Table 3.3 List of Software Requirement

Software	Description
Google Chrome	To view the web page
Microsoft Word 2020	It is use to prepare the report
Draw.io	Drawing tools for modelling
XAMPP	Use for Apache web server, MySQL database (actually MariaDB), Php and Perl (as command-line executables and Apache modules)
Microsoft Visual Studio Code	It use to writing the code

3.3.4.2 Hardware Requirement

A hardware compatibility list (HCL) is frequently included with a hardware requirements list, especially in the case of operating systems. The following is a list of the hardware requirements for developing web application.

Table 3.4 List of Hardware Requirement

Hardware	Description
Laptop : Asus gl552v	Hardware tool to storing and processing data. and used to install all software requirement use for Betta Farm Management System.

Optical Mouse	Control cursor in a GUI (graphical user interface) and can move and select text, files, icons and folders on laptop to assists the development of Betta Farm Management System.

3.4 Conclusion

This chapter concluded with information on the requirement specification. Following the feasibility studies on the overall available technologies, this requirement specification and analysis section provides a more detailed description of the functionality and constraints on the system. This is a critical phase to ensure that the project meets the project's real requirements and to reduce misunderstanding and misinterpretation of the entire system.

CHAPTER 4: DESIGN

4.1 Introduction

The system design document will be briefly described in this chapter. The process of defining the elements of a system, such as the architecture, modules, and components, as well as the various interfaces between those components and the data that flows through it, is known as system design.

Conceptual system design is the first step in the system design process. The entity relationship model is a graphical representation of the logical relationship between entities in order to create a database.

The logical database design phase is the next step in the system design process. A data dictionary is a document that is used to control system access and manipulation. Simultaneously, integrity constraints are defined, and the local logical data model is reviewed with the user.

Lastly, the physical database which produces a description of the system's implementation, is the final phase of the system design methodology. It describes the file organization, base relations, and indexes design, as well as any associated integrity constraints and security measures, that help to achieve efficient data access.

4.2 High-Level Design

In high-level design, it is important to concern the architecture of the system. Users will be more likely to use this system if it has an interactive interface. The modules cover database architecture, application architecture, application flow, and technology architecture, all of which are used to design interactive systems. It also aids in the detection of inconsistencies prior to coding and can be used as a high-level reference manual for how the modules interact.

4.2.1 System Architecture

A system architecture is a conceptual model that defines a system's structure, behavior, and other views. A formal description and representation of a system organized in a way that supports reasoning about the system's structures and behaviors is known as an architecture description. To send data and receive HTML responses from web-based applications, the user must have a device and an internet connection. The server will use read and write permissions to communicate with the file system and database.

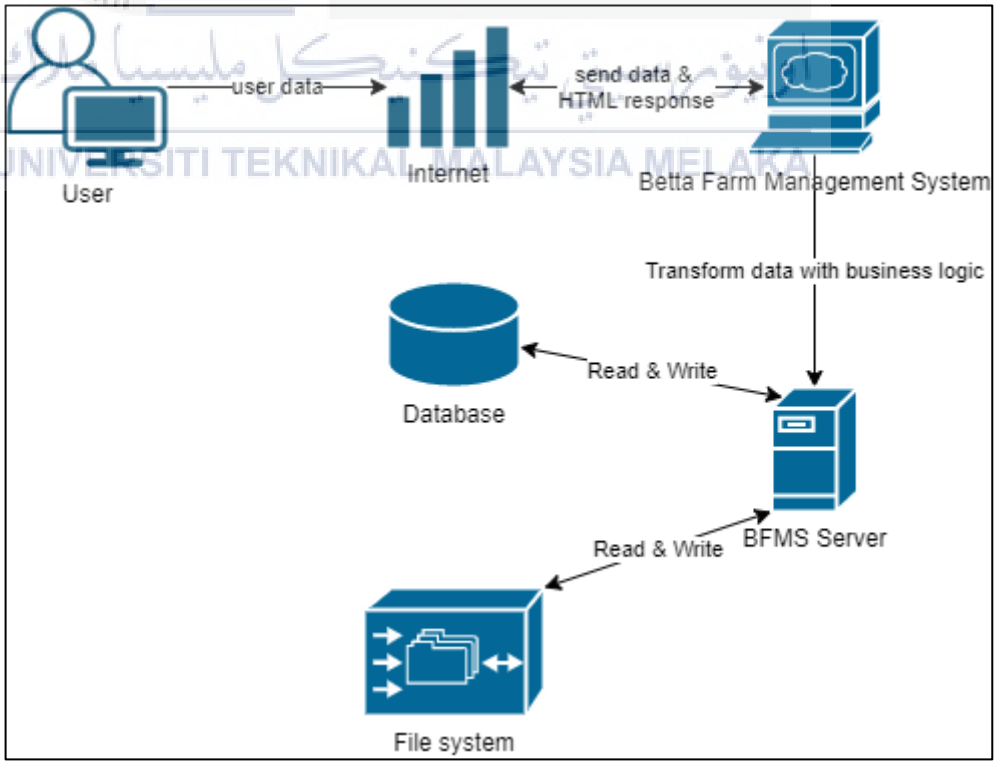

Figure 4.1 System Architecture for BFMS

4.2.2 User Interface Design

4.2.2.1 Navigation Design

The discipline of navigation design entails developing, analyzing, and implementing methods for users to navigate through a website or app. The design gives users an overview of the BFMS method in general. The navigation layout stream identified during the implementation is shown in the diagram below.

- Search Field

Figure 4.2 Search Field Navigation

Users can use the search box to enter a keyword or phrase (query) and submit it to the index, which will return the most relevant results.

- Pagination

Figure 4.3 Pagination Navigation Design

Pagination divides content into pages and allows users to skip between pages or go through the content in chronological order.

- Vertical Navigation

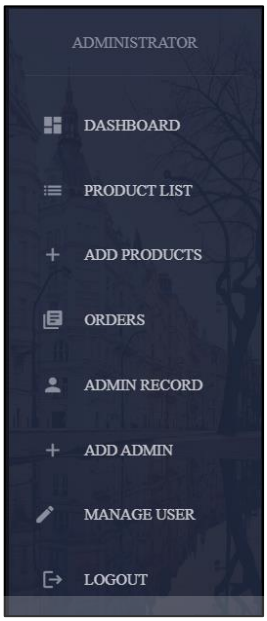

Figure 4.4 Vertical Navigation Design

When the number of categories is large or user-customizable, vertical navigation is appropriate. Because it is familiar, flexible, and takes up little space, it is considered a "safe" navigation pattern.

4.2.2.2 Input Design

The input design is a component of the overall system design that necessitates special consideration. The goal of designing input data is to make it simple and error-free. It's the process of converting user-generated data into a computer-readable format. The user interface for inputs in this project is shown in the diagram below.

- Text Field

Text fields allow users to enter text.

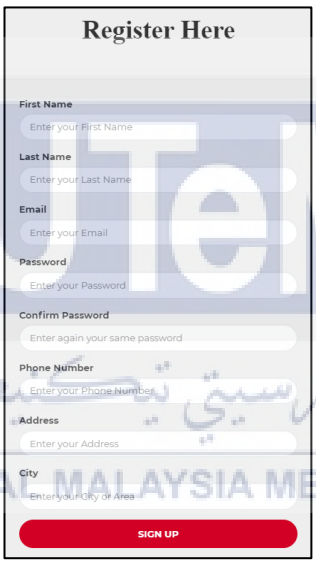

Figure 4.5 Registration Text Field

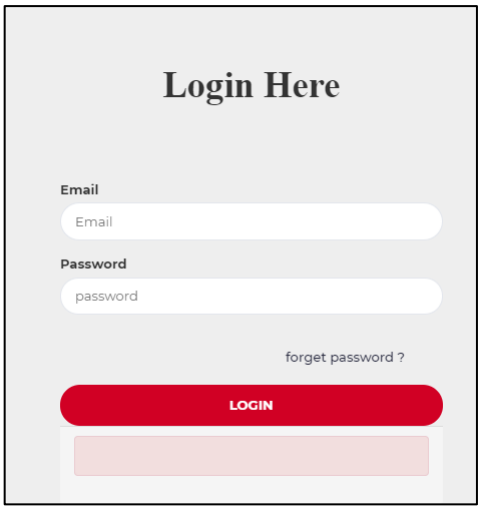

Figure 4.6 Login Text Field

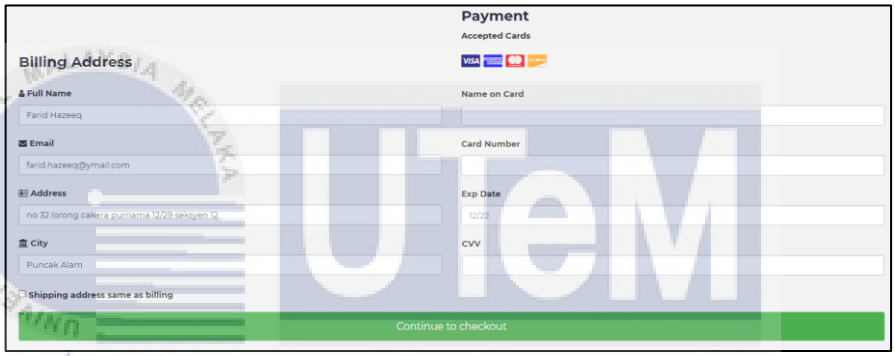

Figure 4.7 Checkout Text Field

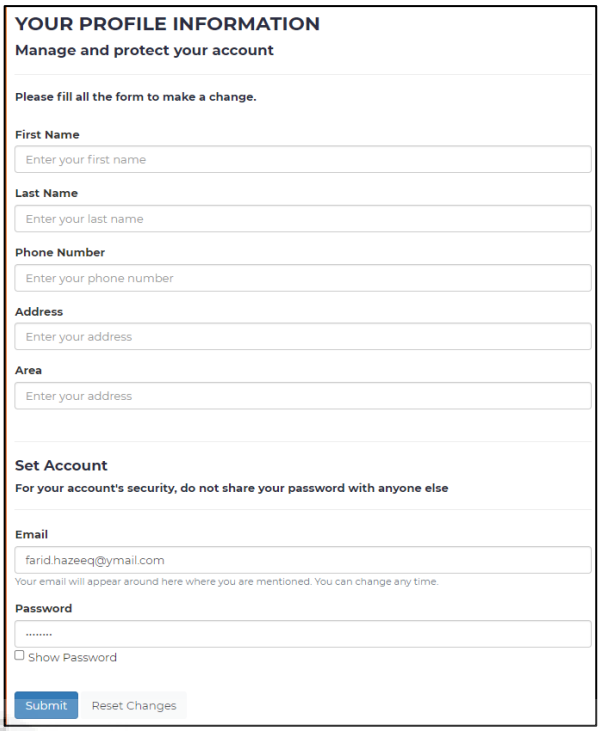

Figure 4.8 Edit Profile Text Field

Figure 4.9 Add Product Text Field

- Dropdown button

The dropdown button is a button that, when user clicked, displays a list of mutually exclusive items as a drop-down menu.

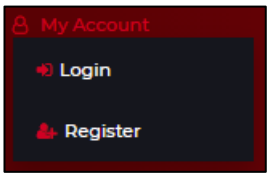

Figure 4.10 Dropdown Button (MyAccount)

4.2.2.3 Output Design

The structure was created to demonstrate the scheme's ability to produce results through application. The results and information generated by the system for users are referred to as output design in BFMS. An output layout can be in terms of a message box, a warning error, output, and so on.

Figure 4.11 Warning Error Output

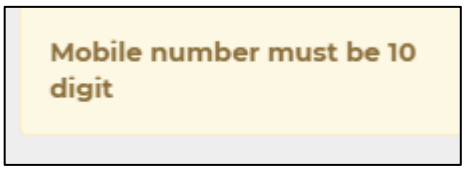

Figure 4.12 Alert Message Output

Figure 4.13 Order Report History

4.2.3 Database Design

4.2.3.1 Conceptual and Logical Database Design

Conceptual database development is a technique for creating an enterprise-wide data modal that is independent of all physical constraints. The modelling and business rules of the Entity Relationship Diagram (ERD) are used to create the conceptual design of the database used in the development of BFMS. Logical design is used to translate the conceptual design into an internal DBMS model. The information dictionary and standardization used in BFMS are explained by the database's logical layout.

4.2.3.2 Entity Relationship Diagram

There are two users involved in this system: admin and user. admin_info, categories, products, brands, user_info, cart, order_info and order_products are the eight entities (tables) that make up the system's ERD.

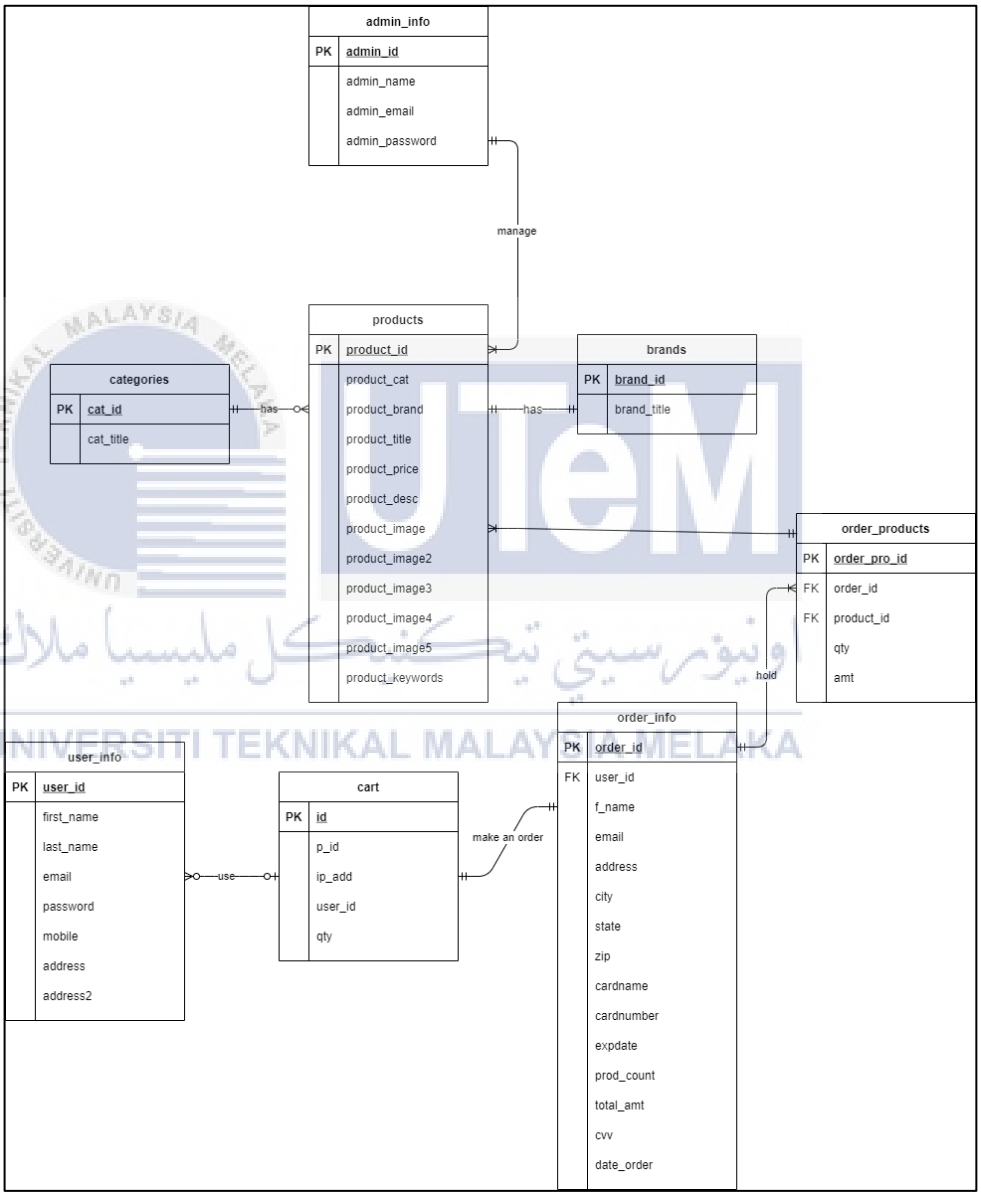

Figure 4.14 Entity Relationship Diagram for BFMS

4.2.3.3 Data Dictionary

Table 4.1 Data Dictionary for Admin

admin_info							
Attribute Name	Contents	Data Type	Size	Required?	Default Value	PK/FK	FK Referenced Table
admin_id	Admin ID	int	10	Yes	Auto Increment	PK	
admin_name	Admin name	varchar	100	Yes	No default		
admin_email	Admin email	varchar	300	Yes	No default		
admin_password	Admin password	varchar	300	Yes	No default		

Table 4.2 Data Dictionary for Brands

brands							
Attribute Name	Contents	Data Type	Size	Required?	Default Value	PK/FK	FK Referenced Table
brand_id	Brand ID	int	100	Yes	Auto Increment	PK	
brand_title	Brand title	text		Yes	No default		

Table 4.3 Data Dictionary for Cart

cart							
Attribute Name	Contents	Data Type	Size	Required?	Default Value	PK/FK	FK Referenced Table
id	Cart ID	int	10	Yes	Auto Increment	PK	
p_id	Product ID	int	10	Yes	No default		
ip_add	Add product to cart	varchar	250		No default		
user_id	User ID	int	10	Yes	Null		
qty	Quantity of the product that has been selected	int	10	Yes	No default		

Table 4.4 Data Dictionary for Categories

categories							
Attribute Name	Contents	Data Type	Size	Required?	Default Value	PK/FK	FK Referenced Table
cat_id	Category ID	int	100	Yes	No default	PK	
cat_title	Category title	text		Yes	No default		

Table 4.5 Data Dictionary for Order Info

order_info							
Attribute Name	Contents	Data Type	Size	Required?	Default Value	PK/FK	FK Referenced Table
order_id	Order ID	int	10	Yes	No default	PK	
user_id	User ID	int	11	Yes	No default	FK	user_info
f_name	User name	varchar	255	Yes	No default		
email	User email	varchar	255	Yes	No default		
address	User address	varchar	255	Yes	No default		
city	User city	varchar	255	Yes	No default		

state	User state	varchar	255	Yes	No default		
zip	User zip	int	10	Yes	No default		
cardname	Name on card	varchar	255	Yes	No default		
carnumber	Card number	varchar	20	Yes	No default		
expdate	Card expired date	varchar	255	Yes	No default		
prod_count	Total product to make order	int	15	Yes	Null		
total_amt	Total amount to pay	int	15	Yes	Null		
cvv	Card CCV number	int	5	Yes	No default		
date_order	Date of order	timestamp		Yes	No default		

Table 4.6 Data Dictionary for Product

products							
Attribute Name	Contents	Data Type	Size	Required?	Default Value	PK/FK	FK Referenced Table
product_id	int	int	100	Yes	Auto Increment	PK	

product_cat	int	int	100	Yes	No default		
product_brand	int	varchar	100	Yes	No default		
product_title	varchar	varchar	255	Yes	No default		
product_price	int	varchar	100	Yes	No default		
product_desc	text	varchar		Yes	No default		
product_image	text	varchar		Yes	No default		
product_image2	varchar	int	200	Yes	No default		
product_image3	varchar	varchar	200	Yes	No default		
product_image4	varchar	varchar	200	Yes	No default		
product_image5	varchar	varchar	200	Yes	No default		
product_keywords	text	int		Yes	Null		

Table 4.7 Data Dictionary for Order Products

order_products							
Attribute Name	Contents	Data Type	Size	Required?	Default Value	PK/FK	FK Referenced Table

order_pro_id	Order product ID	int	10	Yes	Auto Increment	PK	
order_id	Order ID	int	11	Yes	No default	FK	order_info
product_id	Product ID	int	11	Yes	No default	FK	products
qty	Quantity product of order	int	15	Yes	No default		
amt	Amount product of order	int	15	Yes	No default		

Table 4.8 Data Dictionary for User Info

user_info							
Attribute Name	Contents	Data Type	Size	Required?	Default Value	PK/FK	FK Referenced Table
user_id	User ID	int	10	Yes	Auto Increment	PK	
first_name	User first name	varchar	10	Yes	No default		

last_name	User last name	varchar	10	Yes	No default		
email	User email	varchar	10	Yes	No default		
password	User password	varchar	10	Yes	No default		
mobile	User phone number	varchar	10	Yes	No default		
address	User address	varchar	10	Yes	No default		
address2	User address	varchar	10	Yes	No default		

4.3 Detailed Design

4.3.1 Software Design

Every method or operation, including its responsibility, input or output parameter, pre or post condition, and algorithm, is described in software design.

4.3.1.1 Login User

Program name : BFMS_001

Description :

- Receive information of users and save into the database.
- Admin/Users can log on to the system as the database contain all information of users in the system.

Input/Output :

- Login Page
    - Input : Email and password of user.
    - Output : The admin will link to the admin main page while user will link to user main page.

Pseudo code :

- Login Page
    - Step 1 : Initializes all working variables to zeroes.
    - Step 2 : Fill in the needed inputs (email and password)
    - Step 3 : Click the button “Login” to the system.

Screen format：

Login page

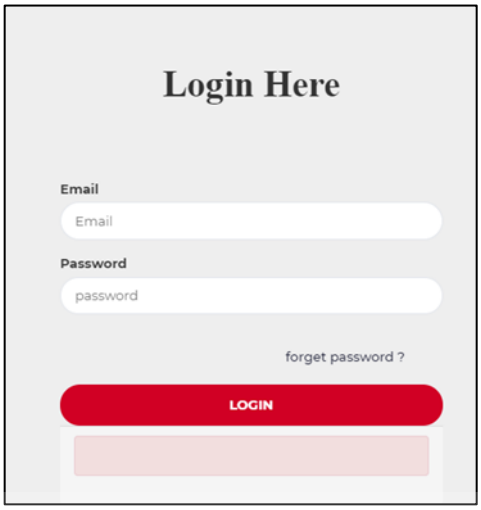

Figure 4.15 Login Page

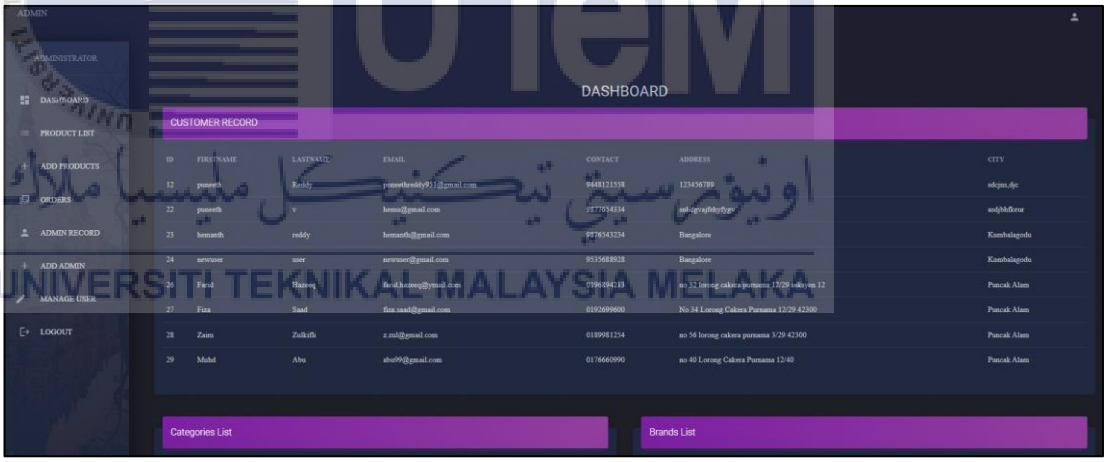

Figure 4.16 Admin Main page

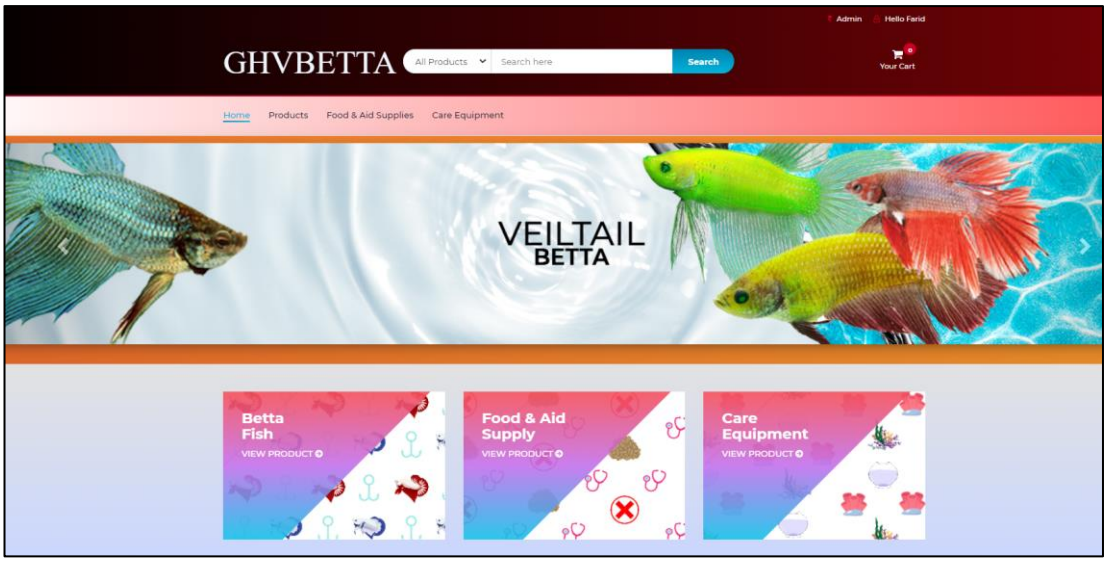

Figure 4.17 User Main Page

4.3.1.2 Manage Products

Program name : BFMS_002

Description :

- Sent and receive information of users and save into the database.
- Admin can add and add and delete.

Input/Output :

- Add product
  - Input : Product title, Product image, Description, Product price, Product category, Product brand and Product keywords.
  - Output : The information of new product has successfully added in the product list page.

- Delete product

    - Output : The information of product has successfully deleted in the product list page

Pseudo code :

- Add product

    - Step 1 : Click the button “Add New” to add product in product list page.

    - Step 2 : Fill in the needed inputs (Product title, Product image, Description, Product price, Product category, Product brand and Product keywords).

    - Step 3 : Click the button “Update Product” to add the product to the system.

- Delete product

    - Step 1 : Click the button “Delete” on product list page to delete the product.

Screen format：

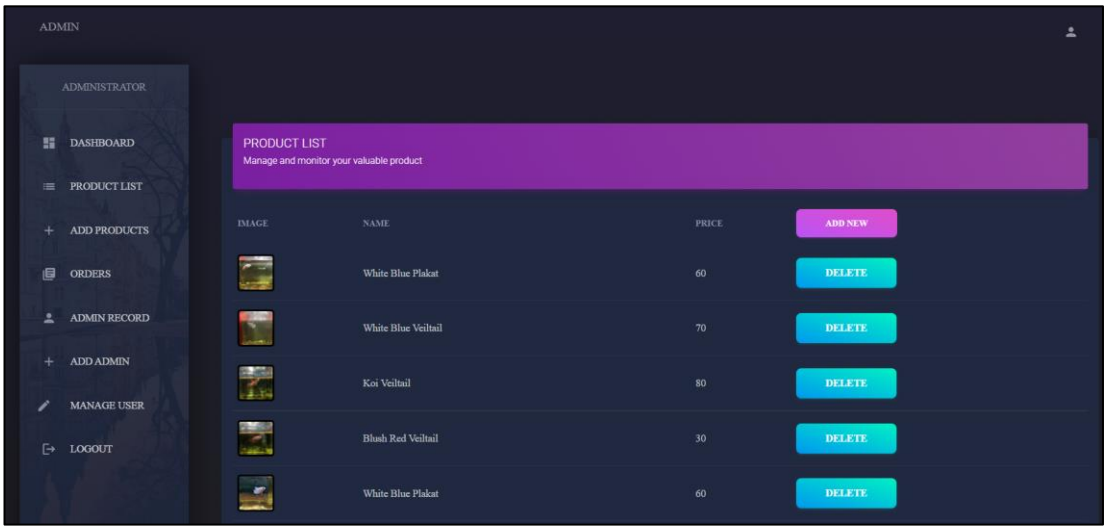

Figure 4.18 Product List Page

Figure 4.19 Add Product Form

4.3.1.3 Manage User

Program name : BFMS_003

Description :

- Sent and receive information of the user and save into the database.
- Admin can add, update and delete customer in the system.

Input/Output :

- Add user
  - Input : First name, Last name, Email, Password, Phone number, City Address.
  - Output : The information of the new user has successfully added in the manage user page.
- Update user
  - Input : First name, Last name, Email, Password.
  - Output : The information of the user has successfully updated in the manage user page.
- Delete user
  - Output : The information of user has successfully deleted in the manage user page.

Pseudo code :

- Add user
    - Step 1 : Click the button “Add New” to user in the system.
    - Step 2 : Fill in the needed inputs (First name, Last name, Email, Password, Phone number, City Address.)
    - Step 3 : Click the button “Update User” to add new user to the system.
- Update user
    - Step 1 : Click the “Pencil” icon to update the details of user.
    - Step 2 : Fill in the needed inputs (First name, Last name, Email, Password.)
    - Step 3 : Click the button “Update” to update the details of user
- Delete user
    - Step 1 : Click the “X” icon to delete the user.

Screen format :

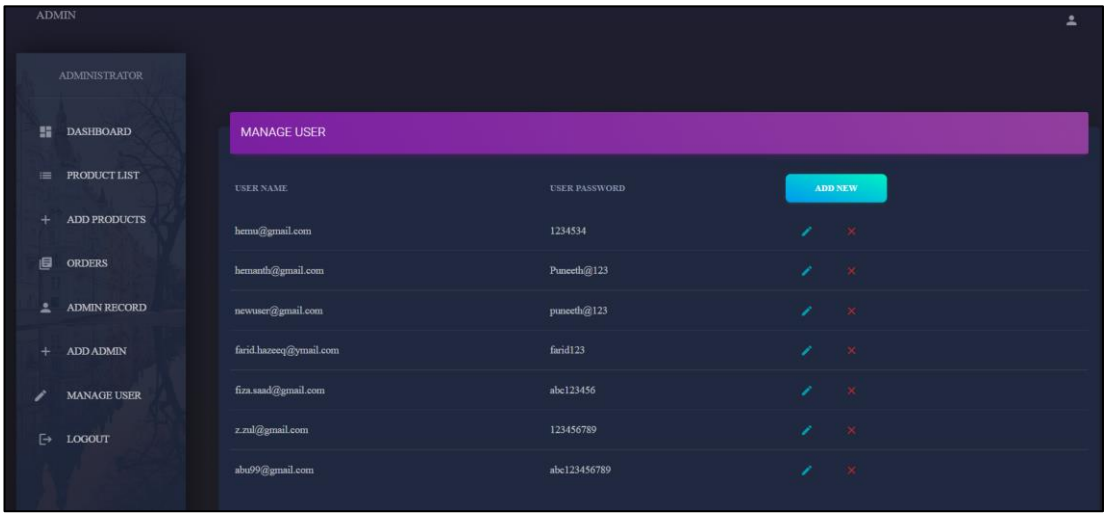

Figure 4.20 Manage User Page

Figure 4.21 Add User Form Page

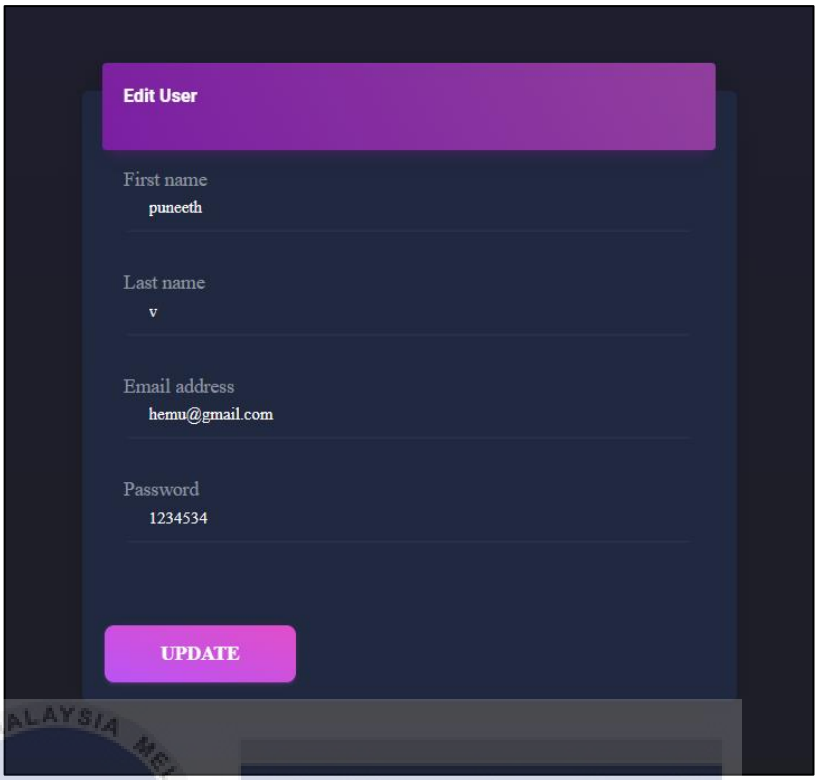

Figure 4.22 Update User Form

4.3.1.4 Manage Cart

Program name : BFMS_004

Description :

- Sent and receive information of product and save in the cart.
- User can add and remove product in the cart.

Input/Output :

- Add product
  - Input : Product ID, Product title.
  - Output : Number of item in cart will increase and the product has successfully added in cart.

- Remove product

    - Output : Number of item in cart will decrease and the product has successfully removed in cart.

Pseudo code :

- Add product

    - Step 1 : Click button “Add To Cart” to add product in cart.

- Remove product

    - Step 1 : Click the “Shopping Cart” icon.
    - Step 2 : Click the button “View Cart”.
    - Step 3 : Click the “Bin” icon to remove product from cart.

Screen format :

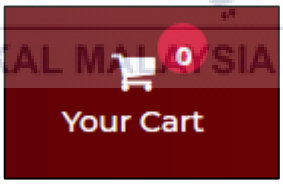

Figure 4.23 View of Cart when is empty

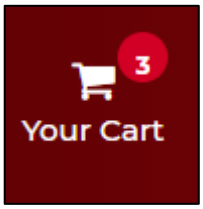

Figure 4.24 View of Cart has an Products

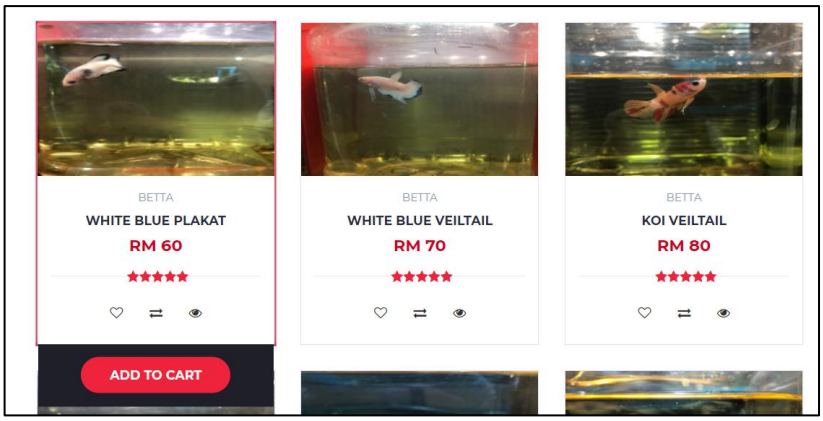

Figure 4.25 View Add To Cart Button on the Product

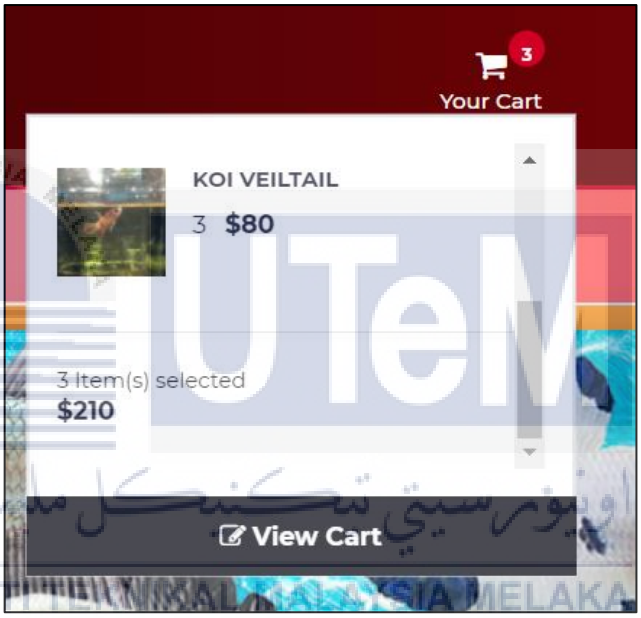

Figure 4.26 Shopping Cart Interface

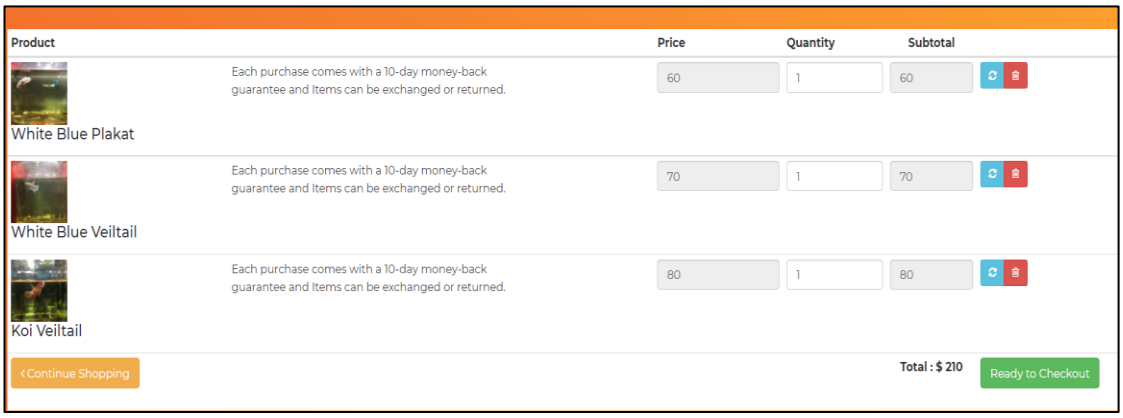

Figure 4.27 Manage Cart Page

4.3.1.5 Manage Profile

Program name : BFMS_005

Description :

- Sent and receive information of user and save into database.
- User can update their details in the system.

Input/Output :

- Update profile
  - Input : First name, Last name, Phone number, Address, Area, Email, Password.
  - Output : The information of user has successfully updated in user profile page.

Pseudo code :

- Update profile
  - Step 1 : Click the button “My profile”.
  - Step 2 : Click the button “Edit Profile”.
  - Step 3 : Fill in the needed inputs (First name, Last name, Phone number, Address, Area, Email, Password).
  - Step 4 : Click the button “Submit” to update user profile in the system.

Screen format :

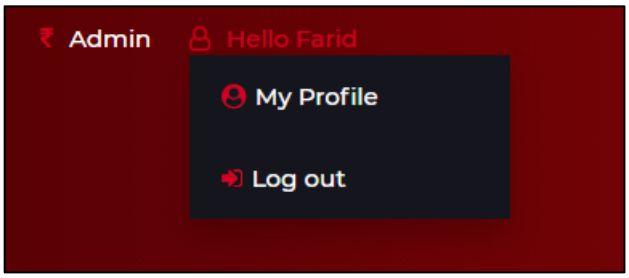

Figure 4.28 My Profile Button View

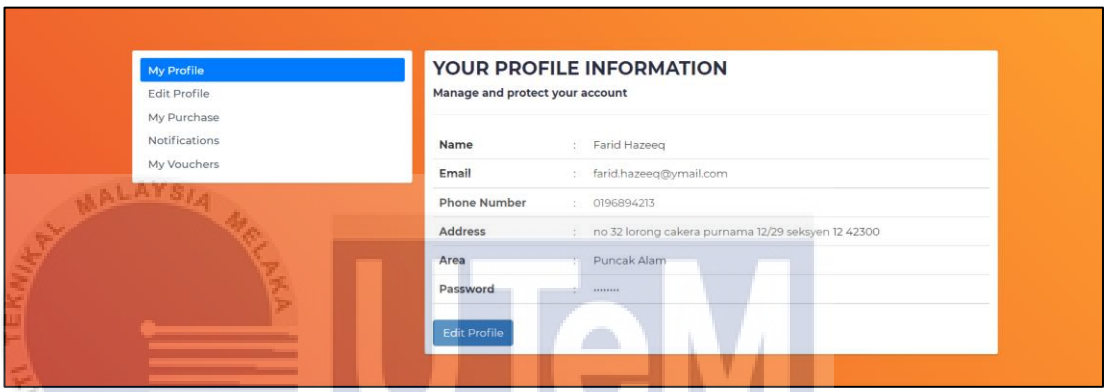

Figure 4.29 User Profile Page

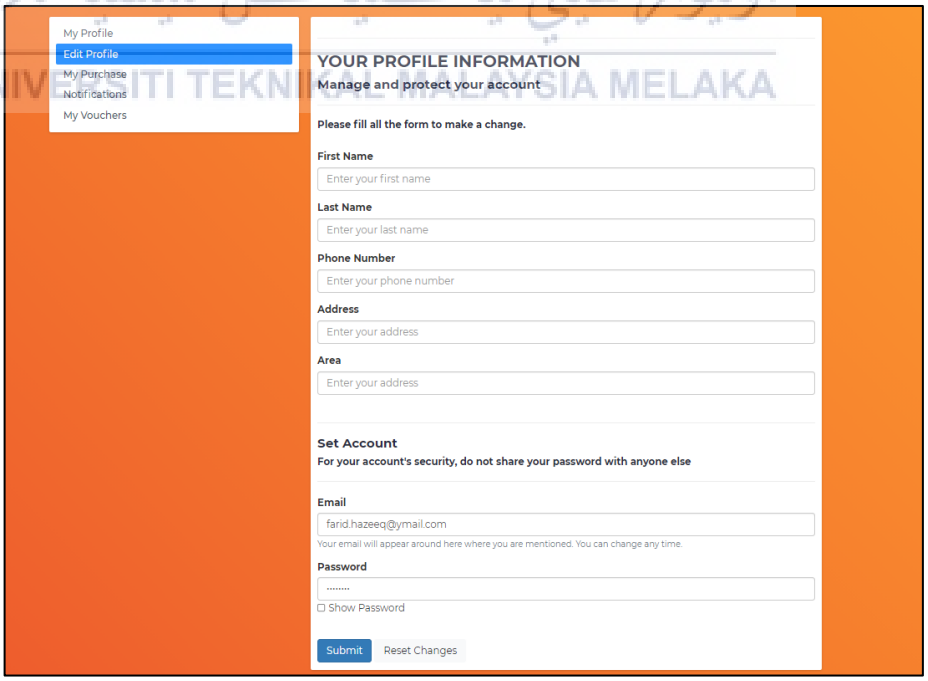

Figure 4.30 Edit Profile Page

4.3.2 Physical Database Design

4.3.2.1 Data Definition Language (DDL)

i. Table Admin Info

```
CREATE TABLE `admin_info` (
  `admin_id` int(10) NOT NULL,
  `admin_name` varchar(100) NOT NULL,
  `admin_email` varchar(300) NOT NULL,
  `admin_password` varchar(300) NOT NULL
)
```

ii. Table Brands

```
CREATE TABLE `brands` (
  `brand_id` int(100) NOT NULL,
  `brand_title` text NOT NULL
)
```

iii. Table Cart

```
CREATE TABLE `cart` (
  `id` int(10) NOT NULL,
  `p_id` int(10) NOT NULL,
  `ip_add` varchar(250) NOT NULL,
  `user_id` int(10) DEFAULT NULL,
  `qty` int(10) NOT NULL
)
```

iv. Table Categories

```
CREATE TABLE `categories` (
  `cat_id` int(100) NOT NULL,
  `cat_title` text CHARACTER SET latin1 NOT NULL
)
```

v. Table Orders Info

```
CREATE TABLE `orders_info` (
  `order_id` int(10) NOT NULL,
  `user_id` int(11) NOT NULL,
  `f_name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `address` varchar(255) NOT NULL,
  `city` varchar(255) NOT NULL,
  `state` varchar(255) NOT NULL,
  `zip` int(10) NOT NULL,
  `cardname` varchar(255) NOT NULL,
  `cardnumber` varchar(20) NOT NULL,
  `expdate` varchar(255) NOT NULL,
  `prod_count` int(15) DEFAULT NULL,
  `total_amt` int(15) DEFAULT NULL,
  `cvv` int(5) NOT NULL,
  `date_order` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
)
```


vi. Table Order Products

```
CREATE TABLE `order_products` (
  `order_pro_id` int(10) NOT NULL,
  `order_id` int(11) NOT NULL,
  `product_id` int(11) NOT NULL,
  `qty` int(15) DEFAULT NULL,
  `amt` int(15) DEFAULT NULL
)
```

vii. Table Products

```
CREATE TABLE `products` (
  `product_id` int(100) NOT NULL,
  `product_cat` int(100) NOT NULL,
  `product_brand` int(100) NOT NULL,
  `product_title` varchar(255) NOT NULL,
  `product_price` int(100) NOT NULL,
  `product_desc` text NOT NULL,
  `product_image` text NOT NULL,
  `product_image2` varchar(200) NOT NULL,
  `product_image3` varchar(200) NOT NULL,
  `product_image4` varchar(200) NOT NULL,
  `product_image5` varchar(200) NOT NULL,
  `product_keywords` text NOT NULL
)
```

viii. Table User Info

```
CREATE TABLE `user_info` (
  `user_id` int(10) NOT NULL,
  `first_name` varchar(100) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `email` varchar(300) NOT NULL,
  `password` varchar(300) NOT NULL,
  `mobile` varchar(10) NOT NULL,
  `address1` varchar(300) NOT NULL,
  `address2` varchar(11) NOT NULL
)
```


4.4 Conclusion

To conclude, designing the database is critical because it can help solve the problems identified in the requirement document during the analysis phase. The design document serves as a roadmap for the solution, and it will be used for implementation, testing, and maintenance in the future. This chapter's output will be used in the next chapter, which will be about the individual database system.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This section covers the setup of the software production environment, database execution, system configuration governance, and execution status. The goal of the implementation phase is to design and implement a system that meets business and design requirements. Writing the entry into the system is generally recognized as an important component during this phase.

5.2 Software Development Environment Setup

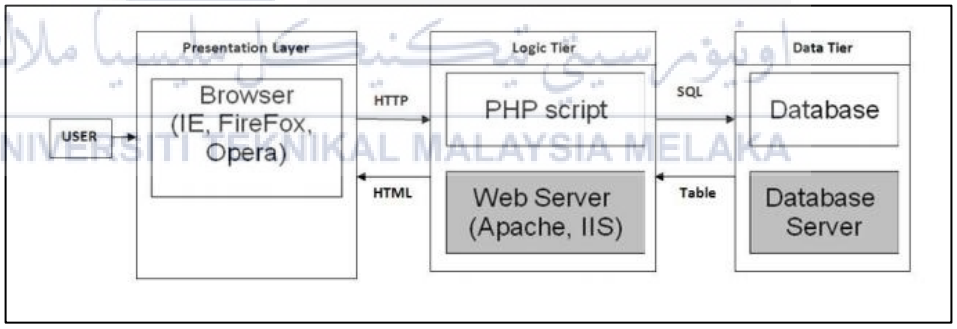

Figure 5.1 Three Tier System Architecture

The three-layer tier system architecture that was used to develop this system is shown in Figure 5.1. The presentation layer represents the user's client computer and software for interacting with the system. The logic tier then represents the programming language as well as the server that allows the user to access the web server, such as Apache. The XAMPP server is used in this system because it comes with a package that includes the PHP language and the Apache server.

5.3 Software Configuration Management

5.3.1 Configuration Environment Setup

This section will describe the software used to set up the configuration management system in the project.

5.3.1.1 Microsoft Visual Studio Code

To create a web application system, BFMS uses Microsoft Visual Studio Code as a development tool. The advantages of this tool include a lightning-fast source code editor that is ideal for everyday use. Syntax highlighting, bracket matching, auto-indentation, box selection, snippets, and more are all available with support for hundreds of languages.

5.3.1.2 Database Configuration

BFMS uses phpMyAdmin because it is a free and open source MySQL and MariaDB administration tool. It has become one of the most popular MySQL administration tools, especially for web hosting services, as a portable web application written primarily in PHP.

5.3.2 Version Control Procedure

The procedure for controlling BFMS source code management is described in the version control procedure. The BFMS version control procedure consists of several steps. The initial development of BFMS is done without the use of version control. Following the completion of the initial development, BFMS is subjected to a version control procedure.

Table 5.1 Version Control Procedure

Version	Description
BFMS v0.1	This is the first version, which has no functionality. Only the navigation flow and interface design are shown in this version. This version is intended for user acceptance testing.
BFMS v0.5	This version includes a number of system modules. Unit testing is used to ensure the functionality of these modules.
BFMS v0.8	This version improves on the previous version by incorporating more modules. More unit testing has been completed, and integration testing has begun. This version corrects errors from previous versions.
BFMS v1.0	The system in its entirety. The entire system is tested. The system includes verification and error handling.

5.4 Implementation Status

The development status of each component or module is depicted in the table below.

Module	Description	Duration	Month completed	Size (%)
Interface design	Develop the user interface for the system.	5 days	February 2021	5
Database design	Develop database and the relationship between them.	7 days	February 2021	5
User Login	User enter the email and password to get an access the system	7 days	March 2021	10
Registration	User can register to make an account to access the system	7 days	March 2021	10
Manage User	Admin can manage the user while user can manage their information	7 days	March 2021	10
Manage Product	Admin can manage the products while	7 days	April 2021	10

	user can view the product			
Manage Cart	User can manage their shopping cart and make a checkout	12 days	May 2021	10
Manage order	Admin can view the order and order's report while user can see their personal order.	5 days	June 2021	10

5.5 Conclusion

After installing multiple software types in the development setting, BFMS is introduced or coded as intended in the implementation phase. The software environment configuration is the tool and process environment that enables BFMS to be developed, validated, and released quickly and consistently. Finally, the testing phase will lead to the next chapter, which will focus on testing and debugging the system.

CHAPTER 6: TESTING

6.1 Introduction

This chapter discusses BFMS testing. Software testing is a method used to evaluate the performance of software. Quality is divided into parts such as accuracy, completeness, and safety, but it also includes more technical specifications such as capacity, compatibility, and usability. The testing stage encompasses not only the method of running a software or application, but also the goal of discovering errors. The purpose of testing is to ensure that each functionality in the system has been completed and that each functionality in the BFMS works as expected. There are three types of testing: test plan, test strategy, and test design. Test organization, test environment, and test schedule are all part of the test plan. There are two types of testing in terms of test design: test description and test data. Finally, the test results and analysis are written down.

6.2 Test Plan

One of the significant measures to ensure that these significant trials are not registered and that exams are documented for future reference is the test plan. A test plan is typically a piece of paper that will be used during a training attempt to provide and record important data about an experiment venture, such as relevant background data or assets. A successful test plan would aid in improving the accuracy and reliability of system testing. The operations of a test plan are test organization, sample setting, and test timeline.

6.2.1 Test Organization

This section will serve two functions. The first is for the test manager, and the second is for the tester. The test manager will be fully responsible for ensuring the smooth and successful development of the project. The tester, on the other hand, will test the system using the interface and functions developed within the system. Any testing results will be recorded for future improvement.

Table 6.1 Test Organization

Tester ID	Name	Roles
Test_01	Farid Hazeeq Bin Burhanuddin	Test Manager
Test_02	Ahmad Zaim Bin Zulkifli	Test User

6.2.2 Test Environment

The test is carried out in-house within online session under the supervision of the test manager. The computer, internet, and mobile phone are used to administer the test. The internet is used to ensure that the system interface runs smoothly, and the computer is used to launch the system via the localhost server. A mobile phone is also used to receive messages while the tester is performing the test.

6.2.3 Test Schedule

The time it took to complete the testing is shown in the table below. Modules are classified based on their scope and are assigned a unique identifier to make them easily identifiable.

Table 6.2 Test Schedule

Test Case	Total Module	Duration
Login Module	1	1 Day
Customer Module	2	1 Day
Administrator Module	3	1 Day

6.3 Test Strategy

This section will go over the strategy that was used during the testing phase. Dynamic testing will be used during the testing phase. White-box and black-box testing methods are used in dynamic testing. Each test will be thoroughly tested using the appropriate method, either white-box or black-box testing. This method is important because it will examine the Betta Farm Management System.

Table 6.3 Black-box and White-box Testing Method

Type of method	Description
Black-box testing	The testing involved an internal structure/implementation that the tester was unfamiliar with. A software tester will perform this testing. This testing includes both functional and non-functional tests.
White-box testing	The testing involved an internal structure/implementation that the tester is familiar with. Testing will take place following the completion of detailed design documentation. The tester must be familiar with programming because he or she will be testing the logic of the software. Path testing and loop testing are included in this testing.

6.3.1 Classes of tests

The Betta Farm Management System is put to the test during the testing process. It is performed to ensure that the system's features function as expected. The tester will look over the application's functionality and user input. The following section will go over all of the relevant test cases in detail.

Aside from that, non-functionality testing, such as security testing, is carried out in the system. Its purpose is to ensure that all data within the system is secure. It

will determine and confirm whether or not the system is vulnerable to a hacker attack. The attack could occur in a variety of ways, such as hackers logging in without being authenticated by the system, among others.

6.4 Test Design

This section will show the process of identifying test case for each module in the test description. To obtain an accurate result, both correct and incorrect data are prepared, and the result is recorded in the test case.

6.4.1 Test Description

The purpose of the test description section is to check and verify that the system function returns the expected result. Each test description will include a unique identifier, a description, and the expected result of the system. The test cases for each module are listed in the table below.

6.4.1.1 Test Case for Administrator

Table 6.4 Test Case for Administrator

Module	Test Case ID	Description	Expected Result
Login	AD001_01	To authenticate user credential when logging into the system with correct user id and password.	System will direct the user into admin dashboard.
	AD001_02	To authenticate user credential when incorrect admin id or password is inserted.	System will display an error message of 'Wrong email or password, Maybe'.

	AD001_03	To authenticate user when field is blank.	System will ask user to fill out the blank filed.
Manage Product	AD002_01	To authenticate user can view list of products.	System display list of products.
	AD002_02	To authenticate user can add new products by click on ‘add new’ and fill in the required field and click on ‘update product’ button.	System will display message of ‘Your Product was added successfully’.
	AD002_03	To authenticate when user left one field blank.	System will ask user to fill out the blank field.
	AD002_04	To authenticate user can delete by click on delete button.	System will display message of ‘Product Have Been Removed’.
Manage User	AD003_01	To authenticate user can view list of users.	System will display list of users.
	AD003_02	To authenticate user can add new user by click on ‘add new’ fill in the required field and click on ‘update user’ button.	System will add new user on the database.

	AD003_03	To authenticate when user left one field blank.	System will ask user to fill out the blank field.
	AD003_04	To authenticate user can delete by click on delete icon.	System will delete the user from the database.

6.4.1.2 Test Case for Customer

Table 6.5 Test Case for Customer

Module	Test Case ID	Description	Expected Result
Login	C001_01	To authenticate user credential when logging into the system with correct user id and password.	User will login into the system.
	C001_02	To authenticate user credential when incorrect user id or password is inserted.	System will display an error message of 'Wrong email or password, Maybe'.
	C001_03	To authenticate user when field is blank.	System will ask user to fill out the blank filed.

Register	C002_01	To create an account to make a product purchase by register in register form.	User be able to purchase product in the system.
	C002_02	To authenticate user when field is blank.	System will ask user to fill out the blank filed.
	C002_03	To authenticate user when symbol and number are not valid on name text field.	System will display an error message of ‘This “ ” is not valid..!’.
	C002_04	To authenticate user when only email format can be accept on email text field.	System will display an error message.
	C002_05	To authenticate user when the email already register into the system.	System will display an error message of ‘Email Address is already available Try Another email address’.
	C002_06	To authenticate user when the password are too weak for register.	System will display an error message of ‘Password is weak’.

	C002_07	To authenticate user when the mobile number is invalid.	System will display an error message of 'Mobile number at least must be 10 digit'.
Shopping Cart	C003_01	To authenticate user when the product have been add in the cart	System will display a message of 'Product is Added..!'.
	C003_02	To authenticate user when the product already in the cart.	System will display a message of 'Product is already added into the cart Continue Shopping..!'.
	C003_03	To authenticate user when the product will be remove from cart.	System will remove the product from user's cart.
Checkout Process	C004_01	To authenticate user when field is blank.	System will ask user to fill out the blank filed.
	C004_02	To authenticate user when payment has been made.	System will display a message 'Payment Successfully.'.

Edit Profile	C005_01	To authenticate user when changes on profile has been made.	System will display a message 'Update Successfully.'.

6.4.2 Test Data

This section describes how the system reacts when valid or invalid data is entered or left blank. The test data will be displayed in the table below.

Table 6.6 Test Data for Administrator

Test Case ID	Test Data	Steps
AD001_01	Email : admin@gmail.com Password : abc12345678	1. Fill in the required field with given data. 2. Click 'login' button.
AD001_02	Email : admin@gmail.com Password : abc123 (incorrect data)	
AD001_03	Email : (left blank) Password : abc12345678	
AD002_01	Image : (picture of product) Name : White Blue Plakat Price : 60	1. Click on 'Product List' page.

AD002_02	Product Title : Plakat Add Image : plakat.img Description : Male plakat Pricing : 40 Product Category : 1 Product Brand : 1 Product Keyword : Betta ADD	1. Click on 'Product List' page. 2. Click 'Add/Delete' button. 3. Fill in the required field with given data.
AD002_03	Product Title : (left blank) Add Image : plakat.img Description : Male plakat Pricing : 40 Product Category : 1 Product Brand : 1 Product Keyword : Betta	
AD002_04	Product Title : Plakat Add Image : plakat.img Description : Male plakat	

	Pricing : 40 Product Category : 1 Product Brand : 1 Product Keyword : Betta DELETE	
AD003_01	User Name : farid.hazeeq@ymail.com Password :farid123	1. Click on 'Manage User' page.
AD003_02	First Name : Zaim Last Name : Zulhilmi Email : z.zul@gmail.com Password : zaim12345678 Phone Number : 0192234567 City : Cheras Address : No. 1 Lorong Makhota Impian 12/80 5600 Cheras, Selangor ADD	1. Click on 'Manage User' page. 2. Click 'Add/Delete' button. 3. Fill in the required field with given data.
AD003_03	First Name : (left blank) Last Name : Zulhilmi	

	Email : z.zul@gmail.com Password : zaim12345678 Phone Number : 0192234567 City : Cheras Address : No. 1 Lorong Makhota Impian 12/80 5600 Cheras, Selangor	
AD003_04	First Name : Zaim Last Name : Zulhilmi Email : z.zul@gmail.com Password : zaim12345678 Phone Number : 0192234567 City : Cheras Address : No. 1 Lorong Makhota Impian 12/80 5600 Cheras, Selangor DELETE	

Table 6.7 Test Data for Customer

Test Case ID	Test Data	Steps
C001_01	Email : farid.hazeeq@ymail.com Password : farid123	1. Fill in the required field with given data. 2. Click 'login' button.
C001_02	Email : farid.hazeeq@ymail.com Password : abc123 (incorrect data)	
C001_03	Email : (left blank) Password : farid123	
C002_01	First Name : Fatin Last Name : Nadiah Email : f.nad@gmail.com Password : Rumputhijau123 Confirm Password : Rumputhijau123 Phone Number : 0196894213 Address : No. 33, Lorong Cakera Purnama 16/78 42300 City : Puncak Alam	1. Click on 'My Account'. 2. Click on 'Register'. 3. Fill in the required field with given data. 4. Click on 'Sign Up' button.

C002_02	First Name : (left blank) Last Name : Nadiah Email : f.nad@gmail.com Password : Rumputhijau123 Confirm Password : Rumputhijau123 Phone Number : 0196894213 Address : No. 33, Lorong Cakera Purnama 16/78 42300 City : Puncak Alam	
C002_03	First Name : Fatin_ Last Name : N@diah Email : f.nad@gmail.com Password : Rumputhijau123 Confirm Password : Rumputhijau123 Phone Number : 0196894213 Address : No. 33, Lorong Cakera Purnama 16/78 42300 City : Puncak Alam	

C002_04	First Name : Fatin Last Name : Nadiah Email : f.nadgmail.com Password : Rumputhijau123 Confirm Password : Rumputhijau123 Phone Number : 0196894213 Address : No. 33, Lorong Cakera Purnama 16/78 42300 City : Puncak Alam	
C002_05	First Name : Fatin Last Name : Nadiah Email : farid.hazeeq@ymail.com Password : Rumputhijau123 Confirm Password : Rumputhijau123 Phone Number : 0196894213 Address : No. 33, Lorong Cakera Purnama 16/78 42300 City : Puncak Alam	

C002_06	First Name : Fatin Last Name : Nadiah Email : f.nad@gmail.com Password : 123 Confirm Password : 123 Phone Number : 0196894213 Address : No. 33, Lorong Cakera Purnama 16/78 42300 City : Puncak Alam	
C002_07	First Name : Fatin Last Name : Nadiah Email : f.nad@gmail.com Password : Rumputhijau123 Confirm Password : Rumputhijau123 Phone Number : 01968942 Address : No. 33, Lorong Cakera Purnama 16/78 42300 City : Puncak Alam	

C003_01	CART Product : White Blue Plakat Price : 60 Quantity : 1 Subtotal : 60 ADD	1. Click on ‘add to cart’ at one of product. 2. Click on ‘Your Cart’. 3. Click on ‘View Cart’.
C003_02	CART Product : White Blue Plakat Price : 60 Quantity : 1 Subtotal : 60 DECLINE	
C003_03	CART Product : White Blue Plakat Price : 60 Quantity : 1 Subtotal : 60 DELETE	1. Click on ‘add to cart’ at one of product. 2. Click on ‘Your Cart’. 3. Click on ‘View Cart’. 4. Click on bin icon to delete product from cart.

C004_01	Name on Card : Farid Hazeeq Card Number : 1234 1234 1234 1234 Exp Date : (left blank) CVV : 898	1. Click on 'add to cart' at one of product. 2. Click on 'Your Cart'. 3. Click on 'View Cart'. 4. Click on 'ready to Checkout' button. 5. Click on 'Continue to checkout'
C004_02	Name on Card : Farid Hazeeq Card Number : 1234 1234 1234 1234 Exp Date : 12/90 CVV : 898	
C005_01	First Name : Fatin Last Name : Amirah (New Name) Phone Number : 01968942 Address : No. 33, Lorong Cakera Purnama 16/78 42300 City : Puncak Alam Email : f.nad@gmail.com Password : Rumputhijau123	Hover to 'Hello, (username). Click on my profile. Click on 'Edit Profile'. Update details that want to update Click on 'Submit' button.

6.5 Test Result and Analysis

The preceding section outlines the requirements for the system to pass the test. All of the test case's results will be recorded in this section. If one of the test cases did not pass the test, it was considered a 'fail'. If the test case passes the test, it is deemed a 'success.' This test result and analysis are critical in ensuring that the system behaves and responds appropriately to each user action.

6.5.1 Test Result for Administrator

Table 6.8 Test Result for Administrator

Test Case ID	Actual Result	Result (Success/Fail)
AD001_01	System will direct the user into admin dashboard.	Success
AD001_02	System will display an error message of 'Wrong email or password, Maybe'.	Success
AD001_03	System will ask user to fill out the blank filed.	Success
AD002_01	System display list of products.	Success
AD002_02	System will display message of 'Your Product was added successfully'.	Success
AD002_03	System will ask user to fill out the blank field.	Success
AD002_04	System will display message of 'Product Have Been Removed'.	Success
AD003_01	System will display list of users.	Success

AD003_02	System will add new user on the database.	Success
AD003_03	System will ask user to fill out the blank field.	Success
AD003_04	System will delete the user from the database.	Success

6.5.2 Test Result for Customer

Test Case ID	Actual Result	Result (Success/Fail)
C001_01	User will login into the system.	Success
C001_02	System will display an error message of 'Wrong email or password, Maybe'.	Success
C001_03	System will ask user to fill out the blank filed.	Success
C002_01	User be able to purchase product in the system.	Success
C002_02	System will ask user to fill out the blank filed.	Success
C002_03	System will display an error message of 'This " " is not valid..!'.	Success
C002_04	System will display an error message.	Success

C002_05	System will display an error message of ‘Email Address is already available Try Another email address’.	Success
C002_06	System will display an error message of ‘Password is weak’.	Success
C002_07	System will display an error message of ‘Mobile number at least must be 10 digit’.	Success
C003_01	System will display a message of ‘Product is Added..!’.	Success
C003_02	System will display a message of ‘Product is already added into the cart Continue Shopping..!’.	Success
C003_03	System will remove the product from user’s cart.	Success
C004_01	System will ask user to fill out the blank filed.	Success
C004_02	System will display a message ‘Payment Successfully.’.	Success
C005_01	System will display a message ‘Update Successfully.’.	Success

6.5.3 Summary of Recorded Test Case

Table 6.9 Summary of Recorded Test Case

Total Test Cases	Success
Test Case for Administrator	11
Test Case for Staff	16
Total	27

6.6 Conclusion

Finally, this chapter completes all of the system's required testing before it is deployed to the end user. Testing is critical to ensuring that the system behaves as specified in the requirements. The following chapter will explain the Betta Farm Management System's strengths and weaknesses, as well as future improvements that can be made to the system.

CHAPTER 7: PROJECT CONCLUSION

7.1 Observation on Weaknesses and Strengths

During the testing phase of the Betta Farm Management System, a few strengths and weaknesses were discovered. These strengths and weaknesses will distinguish the system from any other existing system. The advantages and disadvantages will be discussed further below.

Strength

- The authorized staff can manage the user inside the system.
- The authorized staff can manage the product and stock inventory in one system.
- The authorized staffs can monitor the orders made in the system.
- The customer will be able to make purchase inside the system.
- The authorized staffs will be able to view and monitor all the order from customers.

Weaknesses

- The system cannot provide invoice.
- The system did not have any loyalty or reward program.
- The customer did not have notification for any update for their orders.
- The system doesn’t have an intelligence catalogue.

7.2 Propositions of Improvement

Based on the shortcomings mentioned in the preceding section, the Betta Farm Management System can be improved in a variety of ways. There are only a few suggestions for system enhancements. One suggestion is that the system be integrated into a mobile application, which will make it easier and faster for staff and customers to complete tasks. Aside from that, the system can offer real-time chat to customers, giving them a way to reach staff at the precise moment they have questions or problems they can't solve. This is much more satisfying than sending an email to a support team; with email, customers never know when he or she will get a response. Following that, the system can generate invoices for all purchases to remind customers of the work completed or goods provided. It's an itemized bill, so the customer can see exactly what they're paying for. They're a good way to keep track of things. Next, the system will provide an intelligence catalogue that can provide a good interface and reliability for the user to use the system.

7.3 Project Contribution

The project contribution can be split between the farmer and the individual. The first is for the farmer; this project is applicable to all of Betta's farmers. As a result, when used in computer system, this system will be more effective. Next, as a contribution to the individual who wants to open a betta farm on their own, they have their own system development to store all of the important customer information. As a result, the individual is when a project documentation can be used by anyone to gain

knowledge and to expand ideas to develop another system or improve their current system.

7.4 Conclusion

The conclusion that can be drawn after completing this system is that the developed system has made it easier for user to use and manage their business. This system achieved its goal and solved the main problem identified earlier in this report, but improvements for improved performance and future use are still required. To make the system more reliable and secure, all improvement suggestions must be implemented.

REFERENCES

Mamoun Eid, M. (2015). Requirement Gathering Methods. Retrieved 23 June 2021, from https://www.umsl.edu/~sauterv/analysis/F2015/Requirement%20Gathering%20Methods.html.htm

Agile Software Development Lifecycle Phases Explained. (2021). Retrieved 23 June 2021, from https://relevant.software/blog/agile-software-development-lifecycle-phases-explained/

Agile Software Development Lifecycle Phases Explained. (2021). Retrieved 23 June 2021, from https://relevant.software/blog/agile-software-development-lifecycle-phases-explained/

dfd diagram for online shopping website. (2021). Retrieved 23 June 2021, from https://meeraacademy.com/dfd-diagram-for-online-shopping-website/

Code, V. (2021). Why Visual Studio Code?. Retrieved 23 June 2021, from https://code.visualstudio.com/docs/editor/whyvscode