ROSEWOOD RESORT RESERVATION SYSTEM
(RRRS)

SANDRA MOH SHU YUAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN

JUDUL: ROSEWOOD RESORT RESERVATION SYSTEM (RRRS)

SESI PENGAJIAN: 2023 / 2024

Saya: SANDRA MOH SHU YUAN

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan unituk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. * Sila tandakan (□)

________	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
________	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)
✓	TIDAK TERHAD	

SANDRA	
(TANDATANGAN PELAJAR)	(TANDATANGAN PENYELIA)
Alamat tetap: 44,JALAN TEMBIKAI 28, TAMAN KOTA MASAI, PASIR GUDANG,81700,JOHOR.	TS. ABDUL RAZAK BIN HUSSAIN
Tarikh: 22 JANUARI 2023	Tarikh: 22 January 2024

ROSEWOOD RESORT RESERVATION SYSTEM

(RRRS)

SANDRA MOH SHU YUAN

This report is submitted in partial fulfilment of the requirement for the Bachelor of Computer Science (Database Management)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

DECLARATION

I hereby declare that this project report entitled

ROSEWOOD RESORT RESERVATION SYSTEM (RRRS)

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

STUDENT : SANDRA Date : 22 JAN 2024

(SANDRA MOH SHU YUAN)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Database Management) with Honours.

SUPERVISOR : Date : 22 January 2024

(TS. ABDUL RAZAK BIN HUSSAIN)

DEDICATION

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

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest thanks to University Technical Malaysia Melaka (UTeM) and Faculty of Information and Communication Technology for giving me a great chance to gain knowledge and skills that I had learnt to develop a project.

Besides, I would like to express my great appreciation and unlimited thanks to my supervisor, Mr Ts.Abdul Razak bin Hussain for taking part in the useful decision and giving necessary advice and guidance during the planning and development of this project. Despite being extraordinarily busy with her duties, she took time out to share her knowledge, guided and kept me on the correct path during the project. I am fortunate enough to get constant encouragement, support, and guidance who helped me to successfully complete my project diploma.

In addition, I would like to thanks for my family and friends for supporting and encouraging me during the development of the project. Hence, I would also like to thank everyone who involves directly or indirectly in the project. I might not have done this project successfully without the help received from the people mentioned above.

ABSTRACT

The Rosewood Resort Reservation System is an essential component of the resort's operations, facilitating efficient and seamless booking processes for guests. However, the system has encountered several challenges that have affected its functionality and reliability. This report presents a comprehensive analysis of the problems identified within the current system and proposes potential solutions for their resolution. The primary issues identified include a cumbersome user interface, inaccurate real-time database updates, and inadequate security measures. These problems have resulted in customer dissatisfaction, operational inefficiencies, and potential data breaches. To address these challenges, this report recommends the implementation of a user-friendly interface, an improved database management system, and enhanced security protocols. By addressing these issues, the Rosewood Resort Reservation System can achieve greater customer satisfaction, improved operational efficiency, and ensure the security of guest data.

ABSTRAK

Sistem Tempahan Rosewood Resort adalah komponen penting dalam operasi resort, memudahkan proses tempahan yang cekap dan lancar untuk tetamu. Walau bagaimanapun, sistem telah menghadapi beberapa cabaran yang telah menjejaskan fungsi dan kebolehpercayaannya. Laporan ini membentangkan analisis komprehensif masalah yang dikenal pasti dalam sistem semasa dan mencadangkan penyelesaian yang berpotensi untuk penyelesaiannya. Isu utama yang dikenal pasti termasuk antara muka pengguna yang rumit, kemas kini pangkalan data masa nyata yang tidak tepat dan langkah keselamatan yang tidak mencukupi. Masalah ini telah mengakibatkan ketidakpuasan hati pelanggan, ketidakcekapan operasi dan potensi pelanggaran data. Untuk menangani cabaran ini, laporan ini mengesyorkan pelaksanaan antara muka mesra pengguna, sistem pengurusan pangkalan data yang dipertingkatkan dan protokol keselamatan yang dipertingkatkan. Dengan menangani isu ini, Sistem Tempahan Rosewood Resort boleh mencapai kepuasan pelanggan yang lebih besar, kecekapan operasi yang lebih baik dan memastikan keselamatan data tetamu.

TABLE OF CONTENTS PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

CHAPTER I: INTRODUCTION

1.1 Introduction

The Rosewood resort reservation system is a local resort reserve application for the resort's staff and admin to revolutionize the traditional reserving so that the reservation process could be carried out quickly and easily. This system is reduced as much as possible to avoid errors while reserve in resort. No formal knowledge is needed for the user to use this system. Thus, this all proves the Rosewood Resort Reservation system is user-friendly. Rosewood Resort Reservation system can lead to error-free, secure, reliable, and fast management

The computerized resort reservation system deals with reservation inquiry such as single, group, cancel and recall reservation. During reservation, the details of the customers, type of room required, and number of room required are fed into the system. Once these in formations are entered, the system searches for the unoccupied rooms and displays the result. In reservation inquiry, customer can get the information such as rent of rooms and details of rooms available. Besides, the resort reservation system also allowed special request by customers.

Resort operation will be easy for the resort's admin and staff since all data and information will store in the database and it can access anytime, apart from that it has been constructed to dealing with a large number of reserve simultaneously to prevent room overload. This system can calculate the bill automatically instead of calculating it manually. This can make sure that the resort will not sustain any loss by calculating all the bills correctly. Besides, it can generate a graph automatically to see the book status for every room type in the resort. This reserving project illustrates how to supervise for good performance and better services for small or medium resort.

The system will become an important tool use for a resort to improve the management aspect by utilizing a computerized system to coordinate every room reservation transaction instead of the traditional method. Every organization has challenges to overcome and manage the information of category of room, customer book details and so on. Rosewood Resort Reservation system has user-friendly navigation which is easy to understand and use to ensure that the resort can equip the right level of information and details. In terms of the integrity and availability of the system provided, it can be concluded that this system is a suitable solution for the resort.

1.2 Problem Statement

The current booking system is manual as all the work is done and kept in files. The bookings are done by filling in forms manually which are submitted to custodians therefore taking a lot of time to book a resort meaning performance of the current system is slow and insufficient. They face the problem of data accuracy and not being able to collect the required data in time. Customers have to come early enough and walk around looking for places to rent and do booking. Therefore, it is necessary to develop a resort reservation system that records the room and user by tracking discrete details about of each client, update status each time a booking is made, save clients information into database and generate reports

1.3 Objective

The objective of this project is to study and develop rosewood resort reservation system:

- To keep reservation details for a single and group resort guest.
- To create a hall reservation module for staff to handle hall reservation and payment transactions
- To provide a graph statistic for management to view data easily

1.4 Scope

The scope of this project focuses on three major points which are modules to be developed and target users as well as platform which is to build and publish the outcome of the website.

1.4.1 System Target User

- System Admin
- Resort’s Customer Module
- Resort’s Staff

1.4.2 Module to be developed

The following are admin, resort's customer and staff module in system.

A. System Admin

1. Manage Profile

- Admin can manage their profile such as view and update their username, email address, contact number.

2. Manage User such as Admin, Customer and Staff.

- Admin can manage user by searching username and view user details such as email address, username, contact number, user's role and account activation.
- Admin can update user role to user, admin or staff.
- Admin can block user restrict their activation.
- Admin can delete user.

3. Manage Room & Hall Availability

- Admin can search room and hall availability by select room type , start date and end date to view the room and hall availability.

4. Manage Room Key or Card

- Admin can manage room key by insert, delete, update room key or card's information.

5. Manage Customer's Booking

- Admin can manage customer's booking by searching booking status or username to update the booking details or download the receipt of booking.

6. Generate Report

- Admin can generate report by searching room type, start date and end date of the statistic they want to generate.

7. Manage Staff Work and Leave Record (Attendance)

- Admin can view staff work and leave record.

8. Monitoring Staff's Performance

- ➢ Admin can monitor staff's performance such as they work record they update daily.

B. Resort's Customer Module

1. Manage Profile

- ➢ Customers can manage their profile such as view and update their username, email address, contact number.

2. Room Reservation for personal or Group

- ➢ Customers can book a room by selecting the room type and insert the number of people, start and end date they want to stay based on the availability of the room.

3. Hall Reservation

- ➢ Customers can book a hall by selecting the start and end date they want to stay based on the availability of the hall.

4. Manage Own Room and Hall Booking

- ➢ Customers can manage their own hall and room booking by view, cancel or make payment transfer of the booking

C. Staff Module

1. Manage Profile

- ➢ Staff can manage their profile such as view and update their username, email address, contact number.

2. Manage Own Work Attendance

- ➢ Staff can update their work attendance daily.

3. Manage Own Work Progress

- ➢ Staff can update their work progress in the system to view their own performance.

1.5 Project Significant

The rosewood resort reservation system is used to increase the productivity of the resort. By simplifying the process of room reserve, make a payment, and minimize human error. On the other hand, the resort can provide better customer services to its valued customers by fully utilizing this system. With good customer services, is a good starting point to fulfil customers' satisfaction as well as customers' wants and needs. The benefit earned can stand out from competitors by automating daily operations which will allow resort service providers to increase sales. Besides, it can increase efficiency by shortening the reserving time and eliminating paperwork. Besides, the system also lightens the workload on the resort's end. Once, user place a reservation, the data will send to the resort database and place in a queue in real-time. Besides, the data will be display on the computer screen along with the corresponding option. It allows admin to easily manage the reservation sequentially, produce the necessary item with minimal delay and help reduce human error.

1.6 Expected Output

Rosewood resort reservation system comes online as a publicly listed website and collects the information of customers who come to the resort.

i. **Output one**: provide accurate statistical graph that retrieve from database and able to print the graph and save it in image format and pdf format.

ii. **Output two**: retrieve more relevance information that can be easy to view and read in one row by using join queries and sub-queries statement.

iii. **Output three**: produce simple user interface that ensure user can control this system efficiently.

1.7 Conclusion

Chapter 1 introduces the basic idea of the project, which is rosewood resort reservation system. This system should be useful and convenient in managing resort information records and generating statistical reports for each customer's booking as well as lay down the groundwork for the development of the system to come.

The next chapter will describe the methodology that is applied in this project, followed by planning which is one of important parts to start developing this documentation systematically.

CHAPTER II: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

A system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. Many such frameworks have existed and continually evolve over many years, with its own strengths and weaknesses for each and every type of methodology. Among these methodologies some stand out as the tried and true examples, such as the waterfall methodology, prototyping methodology, incremental methodology, agile methodology, and Rapid Application Development (RAD) methodology. On the other hand, planning is the initial study of this documentation. It is able to lead the system in the shortest way to be developed and complete the task on time.

2.2 Facts and Findings

2.2.1 Domain

The domain for a Rosewood Resort reservation system encompasses various aspects related to managing reservations, bookings, and guest information specifically tailored for the Rosewood Resort. The primary focus of the domain revolves around facilitating the reservation process, ensuring seamless interactions between guests and the resort, and optimizing the overall guest experience.

Within this domain, the reservation system would encompass features such as online booking capabilities, allowing guests to make reservations conveniently through the resort's website or a dedicated online portal. It would also include functionalities for managing room availability, pricing, and different accommodation options, providing guests with a range of choices to suit their preferences.

The domain would further involve managing guest information and profiles, allowing the resort to store and access guest details, such as contact information, stay history, and specific preferences. This information can be used to personalize the guest experience, including providing tailored services, room preferences, and special requests.

Additionally, the domain would encompass features related to managing reservations, modifications, and cancellations efficiently. It would enable guests to make changes to their bookings, ensuring flexibility and convenience. The system would also assist the resort staff in tracking and managing reservations, avoiding any double bookings or conflicts.

The domain would extend to generating reports and analytics for the system administrators. This would include information on occupancy rates, revenue management, and booking trends, helping the resort make informed decisions, implement marketing strategies, and optimize operations.

2.2.2 Existing System

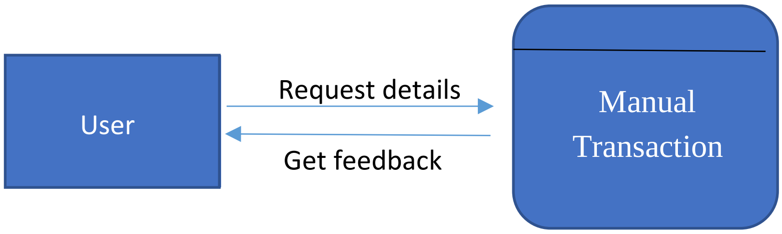


Figure 2.0 Existing System

In the context of resort reservation systems, the manual system refers to the traditional method of managing reservations and bookings without the aid of automated tools or technology. The manual system involves the use of paper-based processes, physical logbooks, and face-to-face interactions. However, relying solely on a manual system can have various implications for different users involved in the resort reservation process: the resort's customers, staff, and system administrators.

For resort customers, the manual system can lead to inefficiencies and inconveniences. Customers may have to physically visit or call the resort to inquire about availability, make reservations, or modify their bookings. This can be time-consuming and frustrating, especially during peak seasons when demand is high. Additionally, the lack of real-time updates in the manual system can result in instances where customers arrive at the resort only to find out that their reservation was not properly recorded or that there was a double booking, leading to dissatisfaction and negative experiences.

Resort staff members also face challenges when using a manual system. They need to maintain physical records, such as logbooks or reservation forms, which can be prone to errors, misplacement, or damage. The process of manually updating and cross-referencing information can be time-consuming and increase the likelihood of human errors. This can result in difficulties in managing reservations accurately, leading to overbooking or missed bookings. Furthermore, the absence of a centralized system makes it challenging for staff members to access up-to-date information about room availability, guest preferences, or special requests, which can hinder their ability to provide personalized services.

System administrators responsible for overseeing the resort reservation process also encounter limitations with a manual system. They face challenges in generating reports, analyzing data, and making informed decisions without access to comprehensive and real-time information. Monitoring inventory, tracking revenue, and identifying trends or patterns become arduous tasks, making it difficult to optimize operations or implement effective marketing strategies. Furthermore, maintaining the security and privacy of customer data can be challenging when using physical records, as they are more susceptible to loss or unauthorized access.

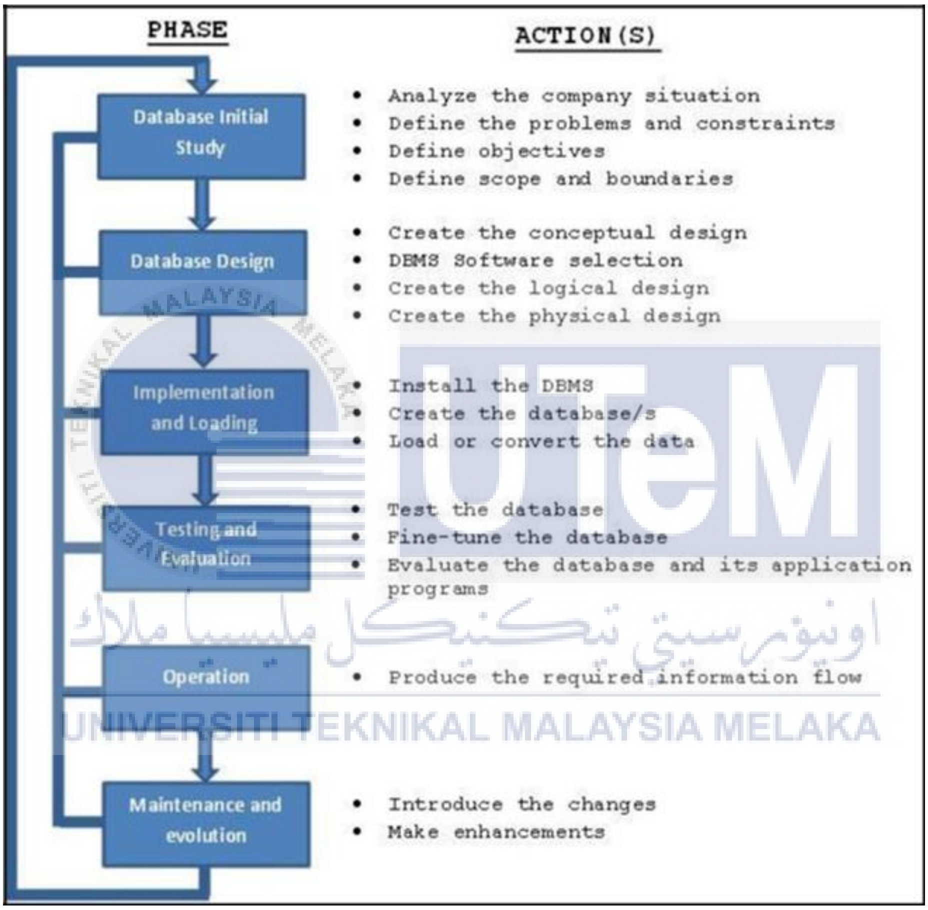
2.2.3 Technique

Projects that are successful are well-managed. To effectively manage a project, the management or development must identify the software development techniques that are most suited to the project at hand. Each technique has its own set of strengthsand drawbacks, and it exists for a variety of reasons. Here are the system developmenttechniques we have chosen to be implemented for this project.

2.3 Project Methodology

Many tests will be conducted as this project is developed in order to detect defects and errors in each software iteration earlier. This ensures that the system can meet all the requirements and can avoid wasting time and resources over a long term period.

As for the database methodology on Database Development Life Cycle (DBLC), it is going to be developed as a top down approach and implemented separately with SDLC.



PHASE
ACTION(S)
Database Initial Study
Analyze the company situation
Define the problems and constraints
Define objectives
Define scope and boundaries
Database Design
Create the conceptual design
DBMS Software selection
Create the logical design
Create the physical design
Implementation and Loading
Install the DBMS
Create the database/s
Load or convert the data
Testing and Evaluation
Test the database
Fine-tune the database
Evaluate the database and its application programs
Operation
Produce the required information flow
Maintenance and evolution
Introduce the changes
Make enhancements

Figure 2.1: Database Life Cycle Illustration

2.3.1 Methodology in Developing Methodology

I. Database Initial Study

Database initial study is the earliest stage to gather and analyse the collected data to provide the system can properly figure out the obstacles and ensure the system requirements are suitable and can be fulfilled. Analysis of the data gathered is done by charting a flow chart, a context diagram and a data flow diagram (DFD) in Figure 3.0 until Figure 3.1.

II. Database Design

i. Entity Relationship Diagram

ER diagrams are used to model and create relational databases, both in terms of logic and business rules and the technology to be employed. An ER diagram is frequently used as the first step in developing requirements for an information systems project in software engineering. It's also used tomodel a specific database or databases. These entities can have properties defined via attributes. An ER diagram depicts the logical structure of databases by identifying entities, their attributes, and the interactions between them. A relational database has a relational table equivalent that can be stated that way if necessary. For this project development, the crow's foot model of ERD will be used to illustrate the relational database.

ii. Data Dictionary

A Data Dictionary is a list of names, definitions, and attributes for data objects in the ER Diagram that are being used or recorded. It explains the meanings and goals of data elements in the context of a project, as well as interpretation, accepted meanings, and representation. A Data Dictionary also contains information about data pieces in the form of metadata. A DataDictionary's metadata can help define the scope and properties of data items, as well as the rules that govern their use and application. Furthermore, it aids in the avoidance of data discrepancies across a projectand the definition of project-wide conventions

III. Implementation

In the implementation stage, this system is installed MySQL as the database for save the records and uses Data Definition Language (DDL) to create the database and include all the tables needed. On the other hands, Data Manipulation Language (DML) is helping to insert, update and delete the information properly in the database. The programming language or the software use to develop the interface is using Adobe Dreamweaver and AppServer 8.6.0 (Win64) through PHP 5.6.30 with local server that mention in 5.2.1.

IV. Testing

This testing phase will conduct two types of tests which are unit testing and system testing. Unit testing will be conducted to each of the functions individually which include login function, forget password functionality, send request function, upload ".csv" batch file function and generate statistical graph function. The unit testing is conducted by inputting sample data into the system to test the connections between interface, database and error messages. If the database implementation fails to meet the system's evaluation criteria or requirement, several options will be considered to enhance the system are as follows:

i. For performances related issues, specific system and DBMS configuration parameter is finely tuned to optimise work cycles. The best sources of information are the hardware and software technical reference manuals.

ii. Modify the logical design.

iii. Upgrade or change the DBMS software or the hardware platform.

V. Maintenance

At this stage, the system is put into practical use to discover problems that were not found in the earlier stages. Therefore, maintenance involves correcting errors, improving system implementation and enhancing the system's services as new requirements are discovered. However, this maintenance phase is not cover in this project.

2.4 Project Requirement

2.4.1 Software Requirement

Table 2.0 Software Requirement

Software	Description
Subline Text 3	To use HTML, CSS and PHP programming language to make a system.
MySQL and phpMyAdmin.	To develop the system database. The database used to store and retrieve the data of a system.
Microsoft Visio	To produce Milestone and Gantt Chart

2.4.2 Hardware Requirements

Table 2.1 Hardware Requirement

Hardware	Description
Laptop	To run and develop a system
HP Printer	To print project documents

2.5 Project Schedule and Milestones

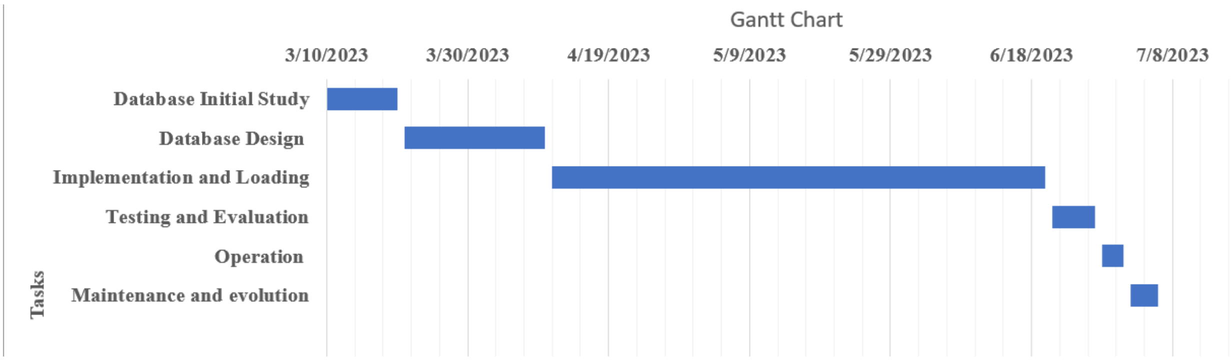
In Table 2.2, it shows the milestones of development of Rosewood Resort Reservation System. In each of the milestone there are documents that are expected to be produced as deliverables or documentation for this project for a smoother development process. This project milestone also acts as a timeline for the development process to ensure all processes is completed in the expected timeline and should be on time with the relevant documents.

Table 2.2: Rosewood Resort Reservation System Schedule

Tasks	Start Date	End Date	Duration
Database Initial Study	3/10/2023	3/20/2023	10
Database Design	3/21/2023	4/10/2023	20
Implementation and Loading	4/11/2023	6/20/2023	70
Testing and Evaluation	6/21/2023	6/27/2023	6
Operation	6/28/2023	7/1/2023	3
Maintenance and evolution	7/2/2023	7/6/2023	4

Figure 2.2 shows the overall development timeline for the project. Aligning the activities and number of weeks keeps the project development on schedule and set targets to be reached by each week.

Figure 2.2: Gantt chart of Rosewood Resort Reservation System



2.6 Conclusion

This chapter mainly discuss the project methodology and planning that was used during the completion of this project. It includes the introduction of project methodology, database development methodology, and project schedule with appropriate milestones. The agile development methodology is used for the system development life cycle (SDLC) approach while a top down approach is implemented separately for the Database Development Life Cycle (DBLC).

For the next chapter, the project analysis will be discussed in more detail. The problem analysis, proposed improvements and solutions, non-functional requirements, functional requirements, as well as other requirements will be included.

CHAPTER III: ANALYSIS

3.1 Introduction

This chapter will show the system analysis. System analysis is a crucial phase in the development and improvement of any system, aiming to understand its functionalities, requirements, and limitations. It involves a comprehensive examination and evaluation of the system's components, processes, and interactions to identify areas for enhancement and propose effective solutions. System analysis plays a vital role in bridging the gap between user needs and technological capabilities, ensuring that the system aligns with the organization's objectives. By employing various techniques such as interviews, observations, and data gathering, system analysts can gain a deep understanding of the system's current state, challenges, and potential improvements. The insights gained from system analysis serve as the foundation for making informed decisions, designing robust systems, and optimizing performance, ultimately leading to more efficient and effective operations.

3.2 Problem analysis

The problem and the limitations that is faced while developing the system is that the system must have a high data integrity so that only authorized person can interact with the interrelated data. Besides, the rosewood hotel reservation system has a lot of confidential records of customer which need to be accessed only by themselves or it will cause an act of security and privacy breach therefore the login system needs to be secured. Next problem, the user need to log into their account manually to check the status of booking request. Hence, this will cause some inconvenience to some users because they cannot notify by the system and then they have to check it anytime by themselves.

3.3 Requirement analysis

3.3.1 Data Requirement (Data Dictionary)

3.3.1.1 TABLE NAME: BOOKING

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
booking_id	Booking ID	int	11	Primary Key	Auto_Increment
room_id	Room ID	int	11	Foreign Key	
user_id	User ID	int	11	Foreign Key	
approver_id	Approver ID	int	11	Foreign Key	
booking_status_id	Status ID	int	11	Foreign Key	
meal_id	Meal ID	int	11	Foreign Key	
booking_start_date	Start Date of booking	datetime		Not Null	
booking_end_date	End Date of booking	datetime		Not Null	
booking_register_date	Register Date of booking	datetime		Not Null	
booking_payment	Booking Payment	varchar	255	Allow Null	
booking_payment_date	Payment Date of booking	datetime		Allow Null	
booking_price	Booking Price	decimal	(10,2)	Not Null	
booking_special_request	Special Request by Customers	varchar	255	Not Null	
booking_cancel_reason	Booking Cancel Reason	varchar	255	Allow Null	

Table 3.0 Data Dictionary of Table Booking

3.3.1.2 TABLE NAME: CUSTOMER

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
customer_id	Customer ID	int	11	Primary Key	Auto_Increment
customer_icno	Customer Identification Card Number	varchar	12		
customer_name	Customer Name	varchar	255	Not Null	
customer_age	Customer Age	int	11	Not Null	
customer_gender	Customer Gender	varchar	255	Not Null	
booking_id	Booking ID	int	11	Foreign Key	

Table 3.1 Data Dictionary of Table Customer

3.3.1.3 TABLE NAME: FLOOR

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
floor_id	Floor ID	int	11	Primary Key	Auto_Increment
floor_name	Floor Name	varchar	255	Not Null	

Table 3.2 Data Dictionary of Table Floor

3.3.1.4 TABLE NAME: ROLE

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
role_id	Role ID	int	11	Primary Key	Auto_Increment
role_name	Role Name	varchar	255	Not Null	

Table 3.3 Data Dictionary of Table Role

3.3.1.5 TABLE NAME: ROOM

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
room_id	Room ID	int	11	Primary Key	Auto_Increment
room_type_id	Room Type ID	int	11	Foreign Key	
floor_id	Floor ID	int	11	Foreign Key	
room_name	Room Name	varchar	255	Not Null	
room_pax	Room Pax	int	11	Null	
room_active	Room Active	int	1	Not Null	
staff_id	Staff ID	int	11	Foreign Key	
last_update	Last Update	datetime		Not Null	

Table 3.4 Data Dictionary of Table Room

3.3.1.6 TABLE NAME: ROOM_TYPE

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
room_type_id	Room Type ID	int	11	Primary Key	Auto_Increment
room_type_name	Room Type Name	varchar	255	Not Null	
room_type_price	Room Type Price	decimal	(10,2)	Not Null	
room_type_image	Room Type Image	varchar	255	Not Null	

Table 3.5 Data Dictionary of Table Room Type

3.3.1.7 TABLE NAME: USER

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
user_id	User ID	int	11	Primary Key	Auto_Increment
user_name	User Name	varchar	255	Not Null	
user_password	User Password	varchar	255	Not Null	
user_email	User Email	varchar	255	Not Null	
user_phone	User Phone	varchar	30	Not Null	
user_active	User Activation	int	1	Not Null	
enforce_change_password	Enforce User to Change Password	int	1	Not Null	
role_id	Role ID	int	11	Foreign Key	

Table 3.6 Data Dictionary of Table User

3.3.1.8 TABLE NAME: BOOKING_STATUS

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
booking_status_id	Status ID	int	11	Primary Key	Auto_Increment
booking_status_name	Status Name	varchar	255	Not Null	

Table 3.7 Data Dictionary of Table Status

3.3.1.9 TABLE NAME: ATTENDANCE_TYPE

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
attendance_type_id	Attendance Type ID	int	11	Primary Key	Auto_Increment
attendance_type_name	Attendance Type Name	varchar	255	Not Null	

Table 3.8 Data Dictionary of Table ATTENDANCE_TYPE

3.3.1.10 TABLE NAME: ATTENDANCE

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
attendance_id	Attendance ID	int	11	Primary Key	Auto_Increment
attendance_datetime	Attendance Date Time	datetime		Not Null	
attendance_type_id	Attendance Type ID	int	11	Foreign Key	
staff_id	Staff ID	int	11	Foreign Key	

Table 3.9 Data Dictionary of Table ATTENDANCE

3.3.1.11 TABLE NAME: MEAL

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
meal_id	Meal ID	int	11	Primary Key	Auto_Increment
meal_name	Meal Name	varchar	255	Not Null	
meal_description	Meal Description	varchar	255	Not Null	
meal_price	Meal Price	decimal	(10,2)	Not Null	

Table 3.10 Data Dictionary of Table MEAL

3.3.1.12 TABLE NAME: WORK_PROGRESS

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
work_progress_id	Work Progress ID	int	11	Primary Key	Auto_Increment
work_progress_remark	Work Progress Remark	varchar	255	Null	
work_progress_datetime	Work Progress Datetime	datetime		Not Null	
work_status_id	Work Status ID	int	11	Foreign Key	
work_progress_details	Work Progress Details	Varchar	255	Null	
work_type_id	Work Type ID	int	11	Foreign Key	
staff_id	Staff ID	int	11	Foreign Key	
admin_id	Admin ID	int	11	Foreign Key	

Table 3.11 Data Dictionary of Table WORK_PROGRESS

3.3.1.13 TABLE NAME: WORK_STATUS

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
work_status_id	Work Status ID	int	11	Primary Key	Auto_Increment
work_status_name	Work Status Name	varchar	255	Not Null	

Table 3.12 Data Dictionary of Table WORK_STATUS

3.3.1.14 TABLE NAME: WORK_TYPE

ATTRIBUTE NAME	DESCRIPTION	DATA TYPE	FIELD SIZE	CONSTRAINT KEY	REMARKS
work_type_id	Work Type ID	int	11	Primary Key	Auto_Increment
work_type_name	Work Type Name	varchar	255	Not Null	

Table 3.13 Data Dictionary of Table WORK_TYPE

3.3.2 Functional Requirement

The functional requirement of a system shows the intended behaviour of the system. A function is described as a set of input, the behaviour and the output. It shapes the design of a system. The Rosewood Resort Reservation System functional requirement includes resort's customer and staff and also system administrators.

A. System Admin

Functional Requirement ID	**Functional Requirement**	**Functional Requirement Description**
FRA_01	Manage Profile	➢ Admin can manage their profile such as view and update their username, email address, contact number.
FRA_02	Manage User such as Admin, Customer and Staff.	➢ Admin can manage user by searching username and view user details such as email address, username, contact number, user's role and account activation. ➢ Admin can update user role to user, admin or staff. ➢ Admin can block user restrict their activation. ➢ Admin can delete user.
FRA_03	Manage Room & Hall Availability	➢ Admin can search room and hall availability by select room type, start date and end date to view the room and hall availability.
FRA_04	Manage Room Key or Card	➢ Admin can manage room key by insert, delete, update room key or card's information.
FRA_05	Manage Customer's Booking	➢ Admin can manage customer's booking by searching booking status or username to update the booking details or download the receipt of booking.

FRA_06	Generate Report	➢ Admin can generate report by searching room type, start date and end date of the statistic they want to generate.
FRA_07	Manage Staff Work and Leave Record	➢ Admin can view staff work and leave record.
FRA_08	Monitoring Staff's Performance	➢ Admin can monitor staff's performance such as they work record they update daily.

Table 3.14 Functional Requirement for System Admin

B. Resort's Customer Module

FRC_01	Manage Profile	➢ Customers can manage their profile such as view and update their username, email address, contact number.
FRC_02	Room Reservation for personal or Group	➢ Customers can book a room by selecting the room type and insert the number of people, start and end date they want to stay based on the availability of the room.
FRC_03	Hall Reservation	➢ Customers can book a hall by selecting the start and end date they want to stay based on the availability of the hall.

Table 3.15 Functional Requirement for Resort's Customer Module

C. Staff Module

FRS_01	Manage Profile	➢ Staff can manage their profile such as view and update their username, email address, contact number.
FRS_02	Manage Own Work Attendance	➢ Staff can update their work attendance daily.
FRS_03	Manage Own Work Progress	➢ Staff can update their work progress in the system to view their own performance.

Table 3.16 Functional Requirement for Staff Module

3.3.2.1 Context Diagram

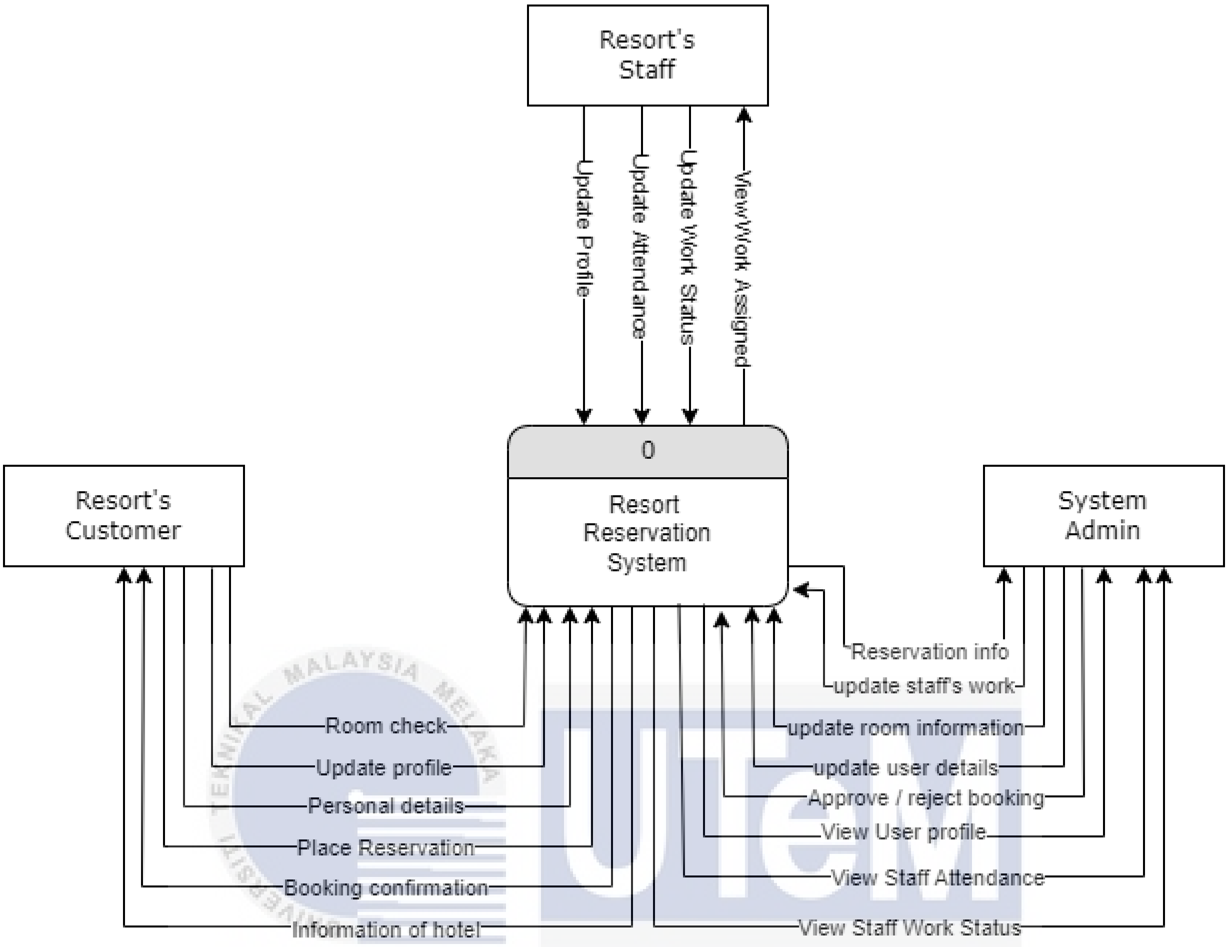


Figure 3.0 Context Diagram

3.3.2.2 Data Flow Diagram – Lvl 1.

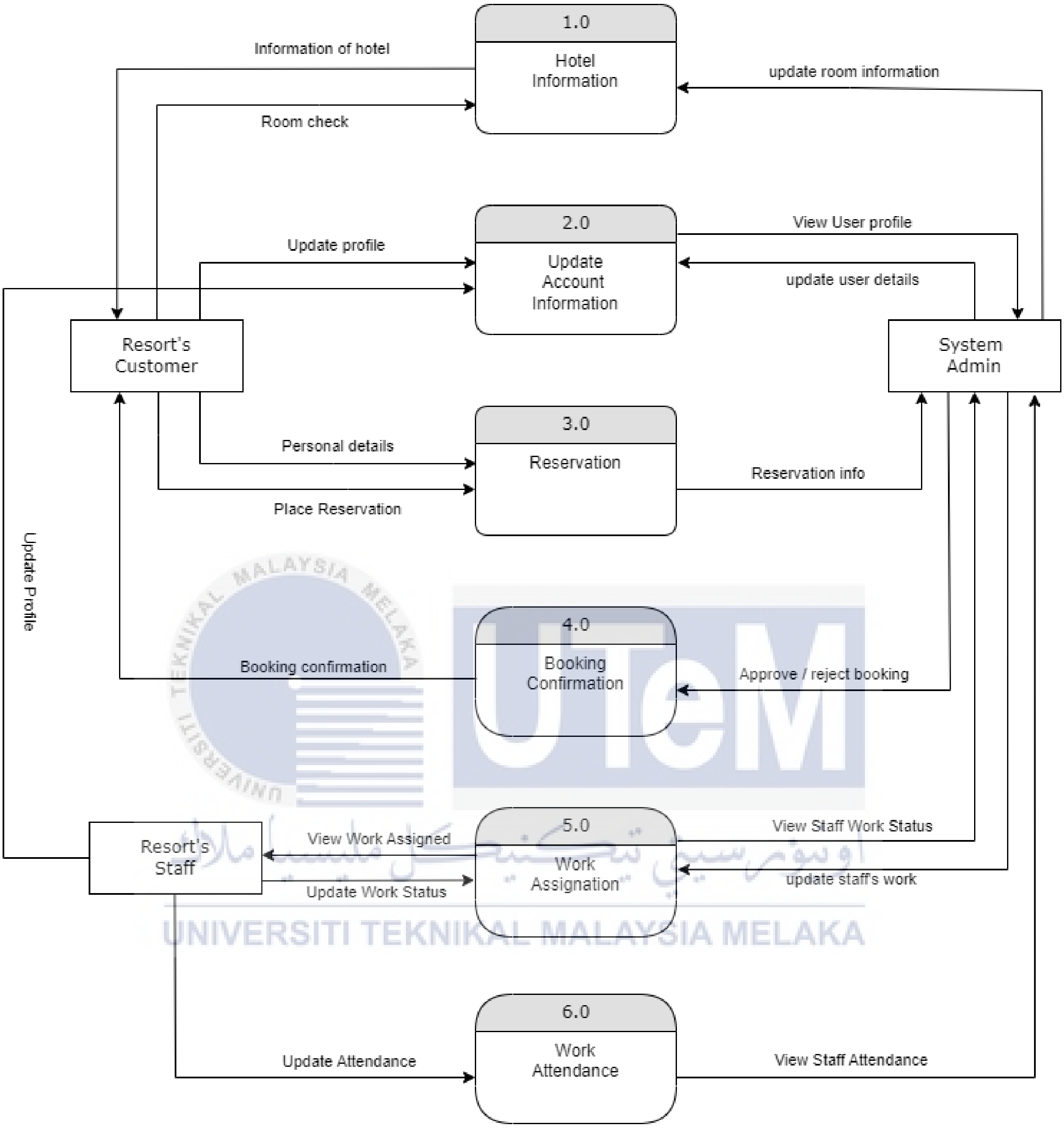


Figure 3.1 Data Flow Diagram – Lvl 1.

3.3.3 Non-Functional Requirements

Non-functional requirement is a group of requirements that describes a variety of system characteristics to attain higher user satisfaction toward the system. Table 3.17 lists out all the non-functional requirements and its descriptions for RoseWood Resort Reservation System

Non Functional Requirement ID	Type	Requirement	Description
NFR_01	Coding Standards	System coding	The system is developed using PHP, JavaScript, HTML, CSS and MySQL.
NFR_02	Integrity	Data integrity	Data should always be consistent 100% through all the interface components.
NFR_03	Security	Data Security	Precautions are taken to ensure users do not have access to higher-level functions and administrative access.
NFR_04	Usability	Portability	The system should be able to operate well on various platforms.
NFR_05	Reusability	Reusability	The system must have common components that are shared across the system on various levels (such as Login page).

Table 3.17: Non-functional requirement

3.3.4 Other Requirements

The requirements of database system development are divided into two categories which is software requirements and hardware requirements. Software requirement describes the software that is used in developing the system while the hardware requirement is the hardware used to run the software described to create the system.

3.3.4.1 Software Requirements

Table 3.18 have listed the requirement and specification of software components, which have been used in Rosewood Resort Reservation System, there are:

Software	Description
Subline Text 3	To use HTML, CSS and PHP programming language to make a system.
MySQL and phpMyAdmin.	To develop the system database. The database used to store and retrieve the data of a system.
Microsoft Visio	To produce Milestone and Gantt Chart

Table 3.18: Software Component List in Rosewood Resort Reservation System

3.3.4.2 Hardware Requirement

The list of hardware component that will be used in the Rosewood Resort Reservation System is as shown in the Table 3.19.

No	Hardware	Description
1.	Asus Vivo Book Laptop	To develop website on the project
2.	HP Printer	To print project documents

Table 3.19: Hardware Requirement Used in Rosewood Resort Reservation System

3.4 Conclusion

The context Diagram shows the system under consideration as a single high-level process and then illustrates the relationship that the system has with its other entities. Meanwhile, the Data Flow Diagram explains each data flow in each function for different users.

This development of data analysis has help simplify the structure and meaning of data in the system. Data analysis techniques can be used as the first step of extrapolating the complexities of a real-world scenario into an understandable model that can be executed on a computer and be accessed by users.

In the next chapter, this project will cover the architecture of this web system such as the design of the database is used to this website and the navigation system that able to guild user to use the web page through the user interface.

CHAPTER IV: DESIGN

4.1 Introduction

Designing the system is the most crucial phase in system development. The logical system design is a result of system analysis and is converted into the physical system design. Physical design is a detailed description of what is needed to solve the stated problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up into details. Data structure, control process, interface, documentation, and procedures are decided at this stage.

There are several techniques that can be used to describe the design of the system. In this case, it can be illustrated using Entity Relationship Diagrams (ERD), business rules, data dictionary, data normalization, the selection of a suitable Database Management System (DBMS), and the creation of a Graphical User Interface (GUI).

4.2 High-level Design

4.2.1 System Architecture Design

In this system, the architecture used is a web-based database application system. This system is accessed over a network connection using a website, rather than running through device storage. This application requires a web browser to run.

The web-based system approach is chosen as the architecture view in this project as it suits the needs of customers who will usually access the system online and on the go through a smartphone web browser. This requires a system that is scalable between a desktop and mobile version that using browser in mobile phone, as well as being accessible anywhere. These specific requirements can easily be fulfilled through the use of a website for the proposed system.

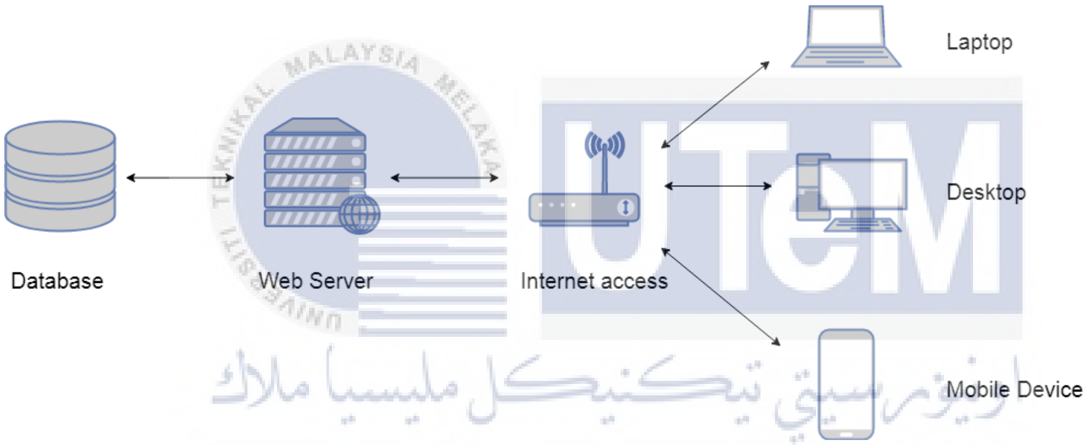


Figure 4.0 Component to Access Rosewood Resort Reservation System

4.2.2 User Interface Design

4.2.2.1 Index Design

4.2.2.1.1 Index Page

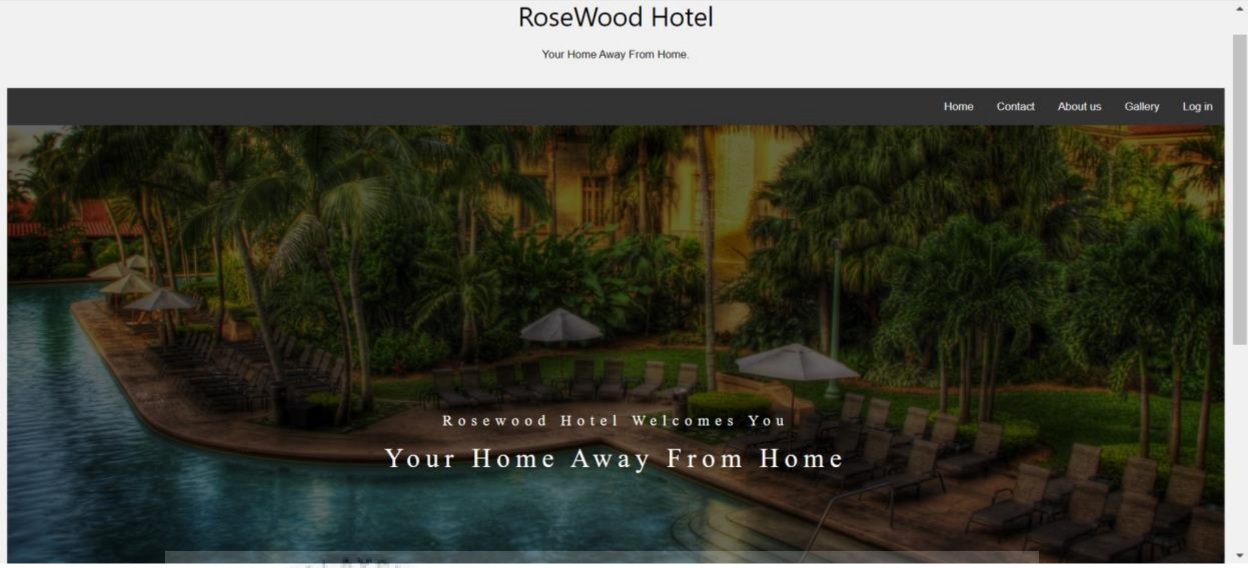
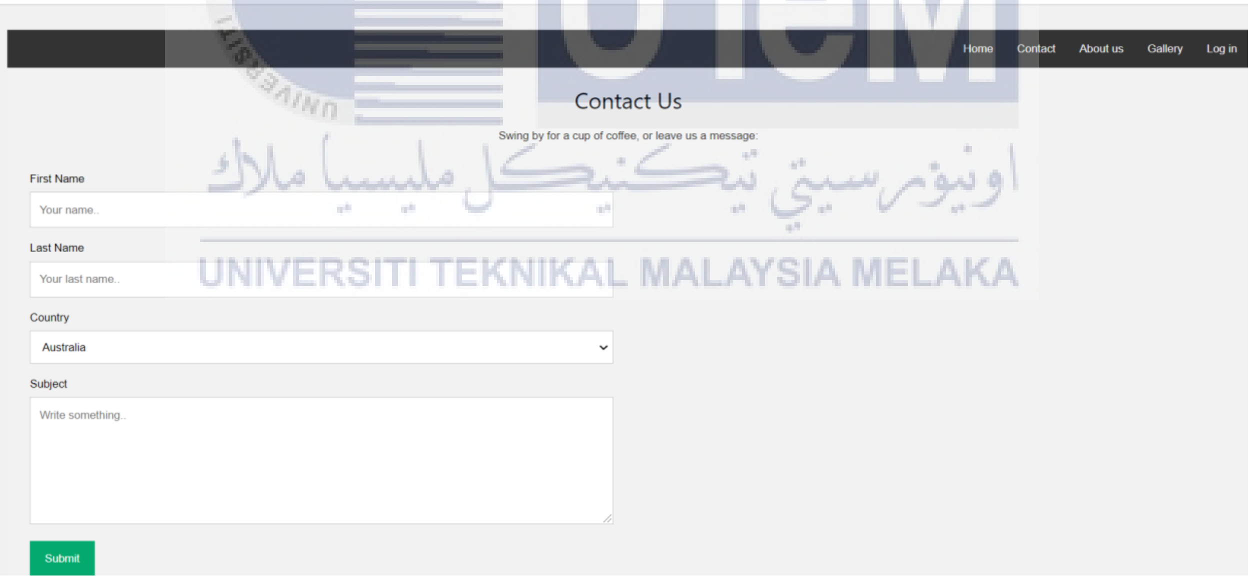


Figure 4.1 Index Page

4.2.2.1.2 Contact Us Page



Contact Us

Swing by for a cup of coffee, or leave us a message:

First Name

Your name..

Last Name

Your last name..

Country

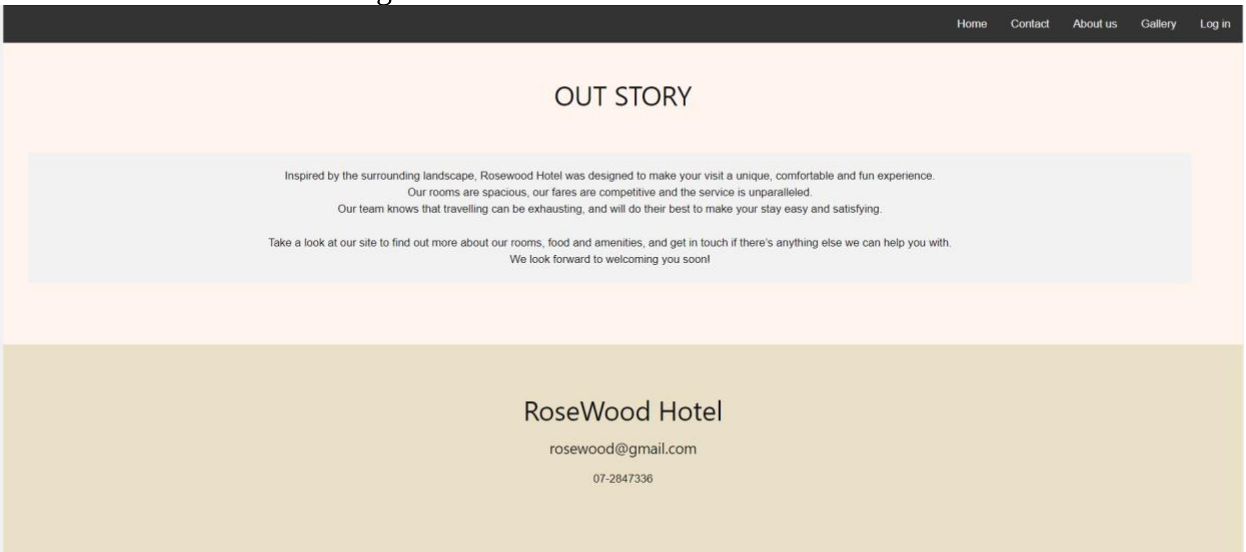
Australia

Subject

Write something..

Figure 4.2 Contact Us Page

4.2.2.1.3 About Us Page



OUT STORY

Inspired by the surrounding landscape, Rosewood Hotel was designed to make your visit a unique, comfortable and fun experience.
Our rooms are spacious, our fares are competitive and the service is unparalleled.
Our team knows that travelling can be exhausting, and will do their best to make your stay easy and satisfying.

Take a look at our site to find out more about our rooms, food and amenities, and get in touch if there's anything else we can help you with.
We look forward to welcoming you soon!

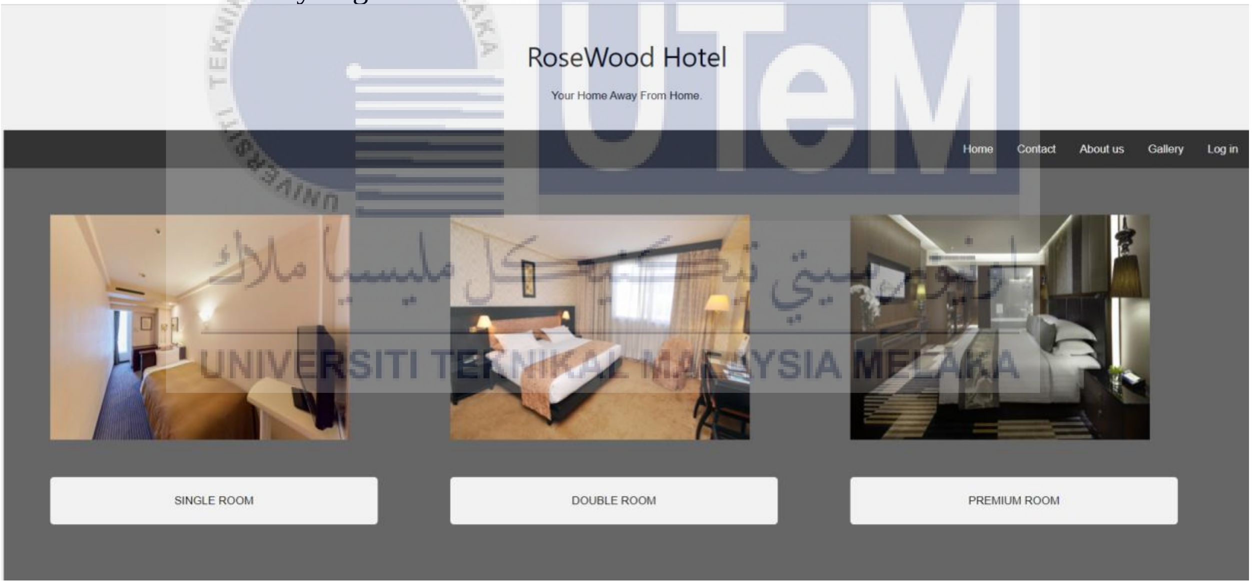
RoseWood Hotel

rosewood@gmail.com

07-2847336

Figure 4.3 About Us Page

4.2.2.1.4 Gallery Page

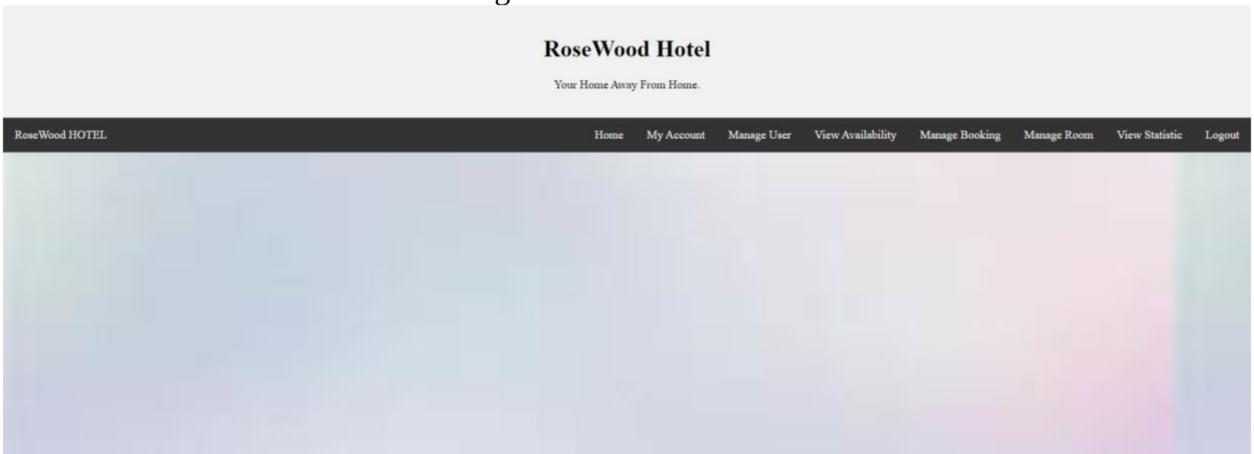


RoseWood Hotel

Your Home Away From Home.

Figure 4.4 Gallery Page

4.2.2.1.5 Admin Dashboard Page

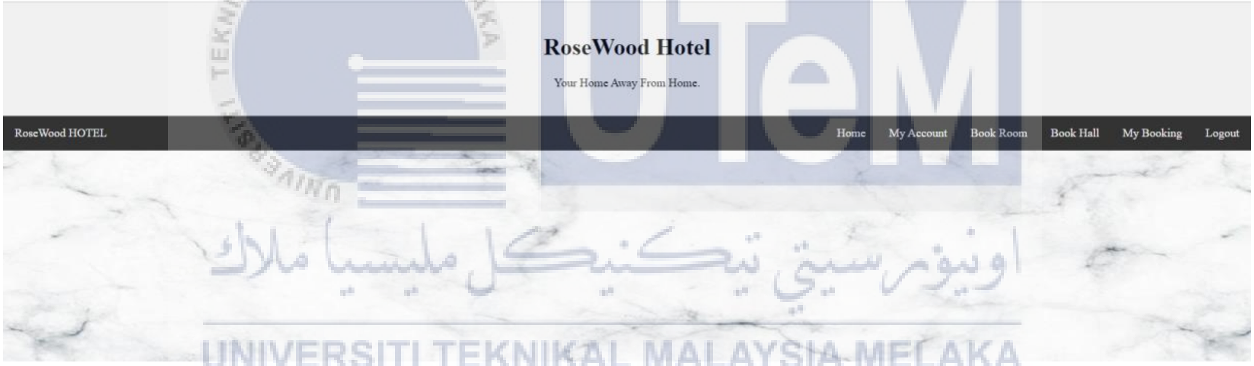


RoseWood Hotel

Your Home Away From Home.

Figure 4.5 Admin Dashboard Page

4.2.2.1.6 User Dashboard Page

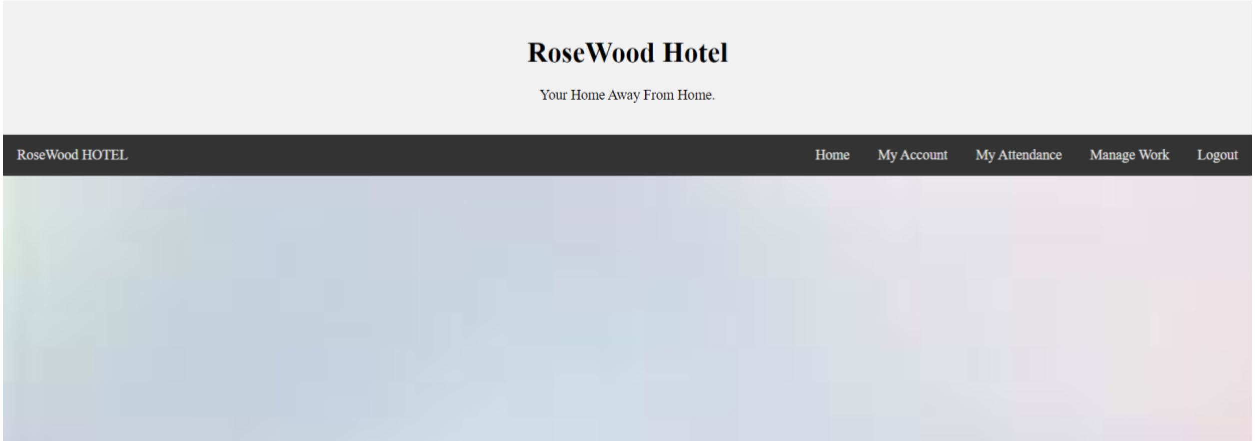


RoseWood Hotel

Your Home Away From Home.

Figure 4.6 User Dashboard Page

4.2.2.1.7 Staff Dashboard Page



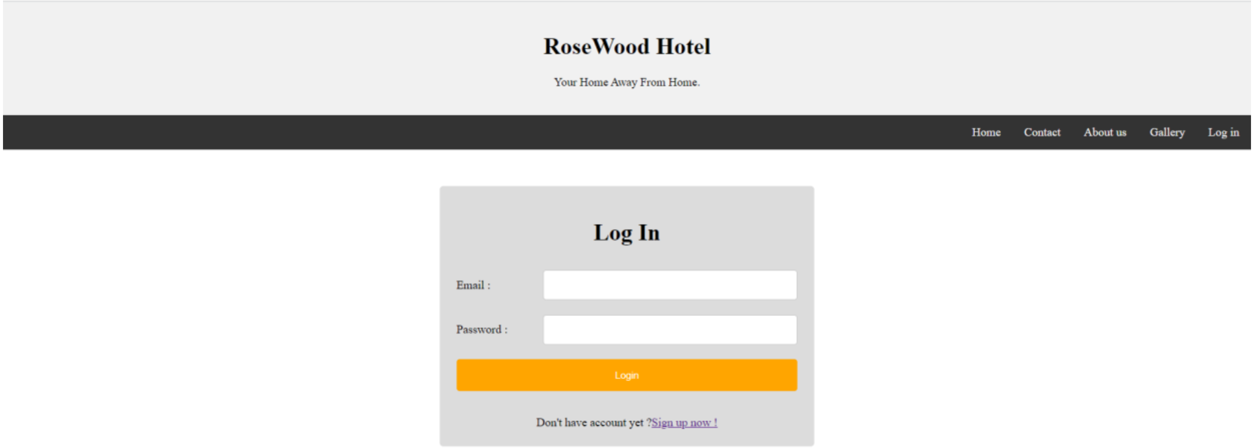
RoseWood Hotel

Your Home Away From Home.

Figure 4.7 Staff Dashboard Page

4.2.2.2 Input Design

4.2.2.2.1 Login Page



RoseWood Hotel

Your Home Away From Home.

Log In

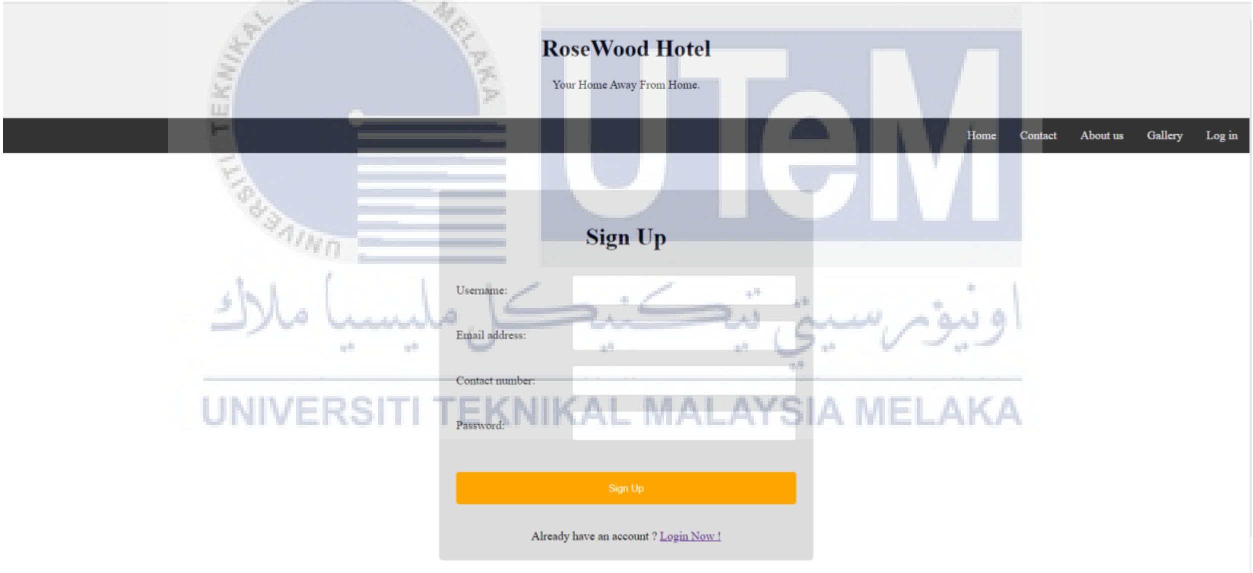
Email :

Password :

Don't have account yet ?Sign up now !

Figure 4.8 Login Page

4.2.2.2.2 Register Page



RoseWood Hotel

Your Home Away From Home.

Sign Up

Username:

Email address:

Contact number:

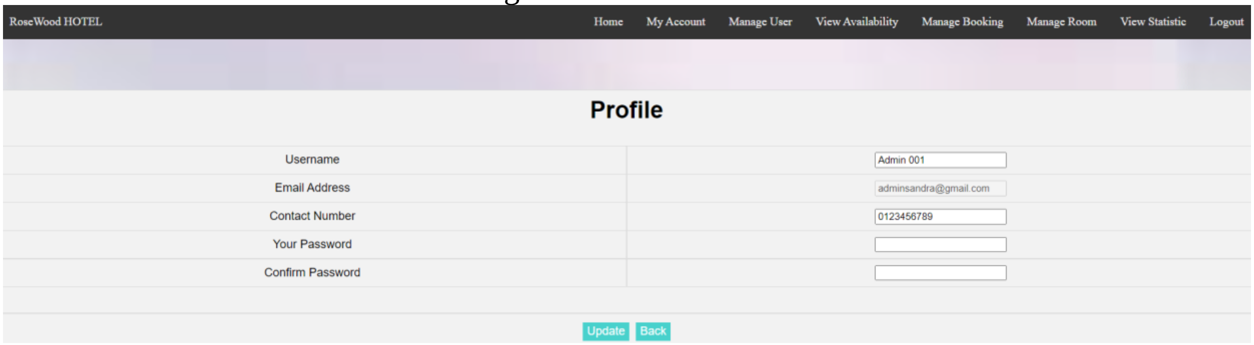
Password:

Already have an account ? Login Now !

Figure 4.9 Register Page

Admin Page

4.2.2.2.3 Admin Edit Profile Page

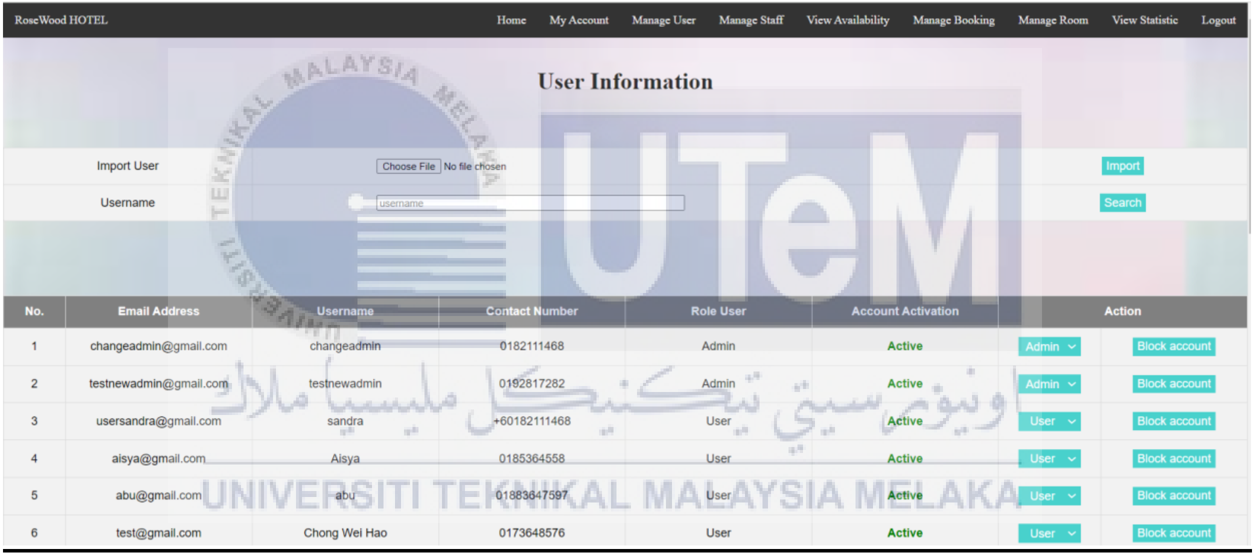


Profile

Username	Admin 001
Email Address	adminsandra@gmail.com
Contact Number	0123456789
Your Password	
Confirm Password	

Figure 4.10 Admin Edit Profile Page

4.2.2.2.4 Admin Manage User Page (Update Active Status, Block user and Add new user)



User Information

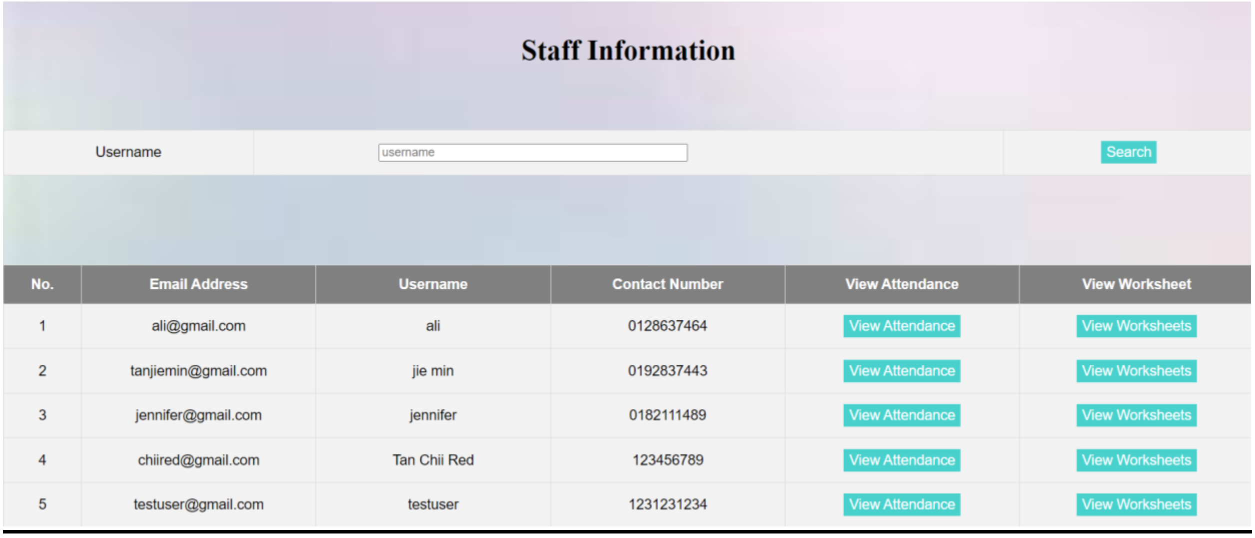
Import User — Choose File No file chosen — Import

Username — username — Search

No.	Email Address	Username	Contact Number	Role User	Account Activation	Action	
1	changeadmin@gmail.com	changeadmin	0182111468	Admin	Active	Admin	Block account
2	testnewadmin@gmail.com	testnewadmin	0192817282	Admin	Active	Admin	Block account
3	usersandra@gmail.com	sandra	+60182111468	User	Active	User	Block account
4	aisya@gmail.com	Aisya	0185364558	User	Active	User	Block account
5	abu@gmail.com	abu	01883647597	User	Active	User	Block account
6	test@gmail.com	Chong Wei Hao	0173648576	User	Active	User	Block account

Figure 4.11 Admin Manage User Page (Update Active Status, Block user and Add new user)

4.2.2.2.4 Admin Manage Staff Page

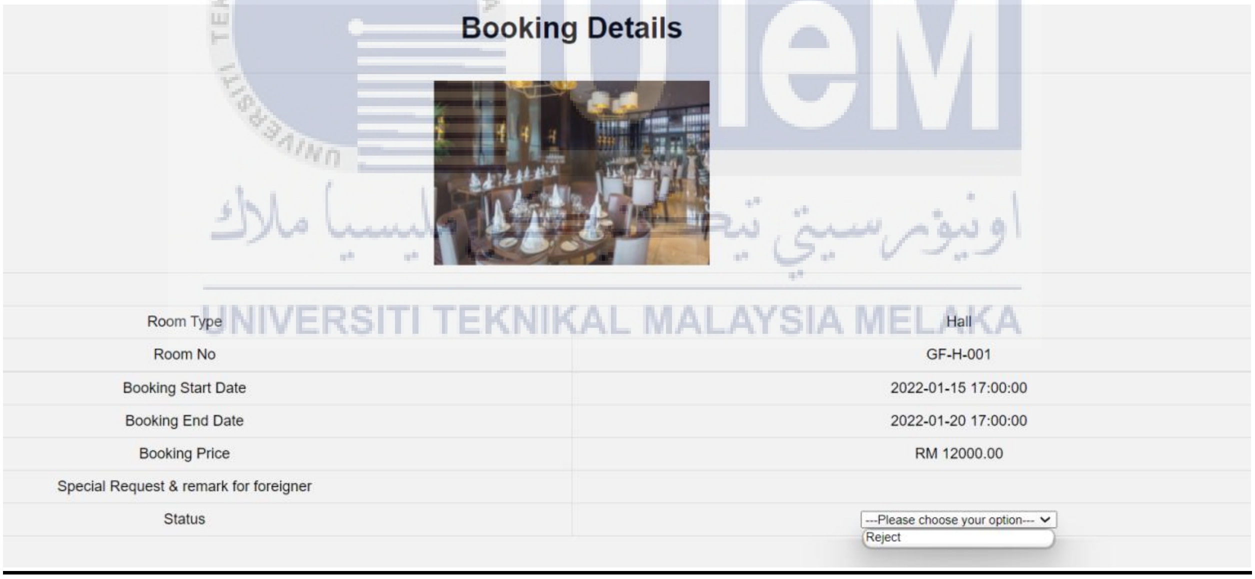


Staff Information

No.	Email Address	Username	Contact Number	View Attendance	View Worksheet
1	ali@gmail.com	ali	0128637464	View Attendance	View Worksheets
2	tanjiemin@gmail.com	jie min	0192837443	View Attendance	View Worksheets
3	jennifer@gmail.com	jennifer	0182111489	View Attendance	View Worksheets
4	chiired@gmail.com	Tan Chii Red	123456789	View Attendance	View Worksheets
5	testuser@gmail.com	testuser	1231231234	View Attendance	View Worksheets

Figure 4.12 Admin Manage Staff Page (Update Active Status, Block user account)

4.2.2.2.5 Admin Manage Booking Page (update booking status)

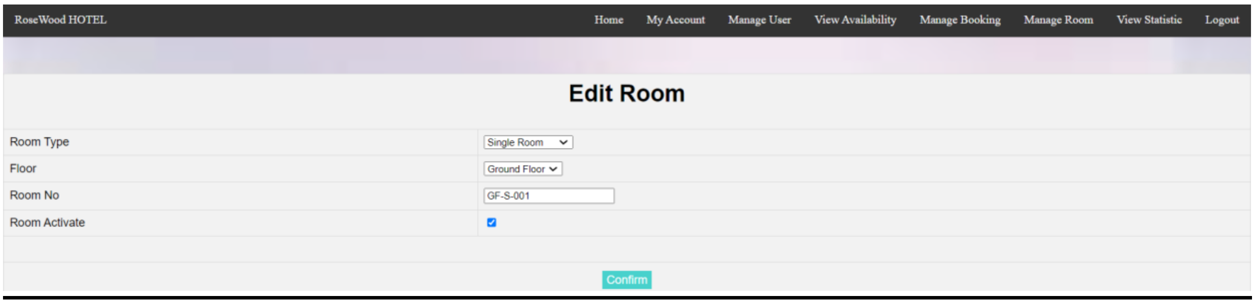


Booking Details

Room Type	Hall
Room No	GF-H-001
Booking Start Date	2022-01-15 17:00:00
Booking End Date	2022-01-20 17:00:00
Booking Price	RM 12000.00
Special Request & remark for foreigner	
Status	---Please choose your option--- Reject

Figure 4.13 Admin Manage Booking Page (update booking status)

4.2.2.2.6 Admin Edit Room Page

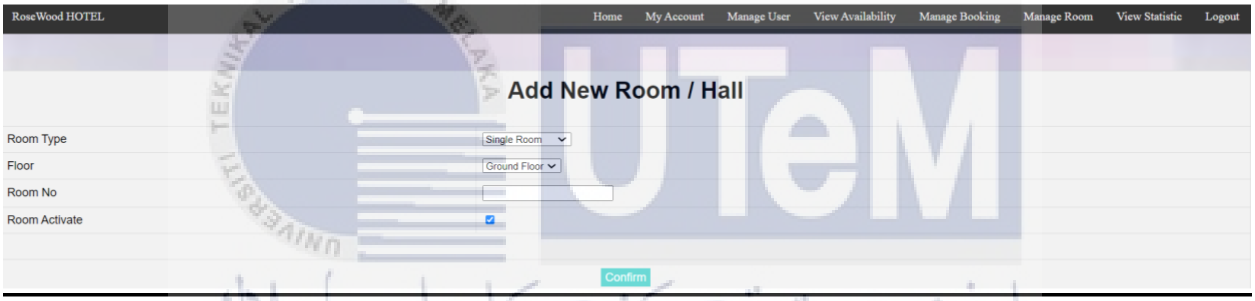


Edit Room

Room Type	Single Room
Floor	Ground Floor
Room No	GF-S-001
Room Activate	☑

Figure 4.14 Admin Edit Room Page

4.2.2.2.7 Admin Add New Room Page

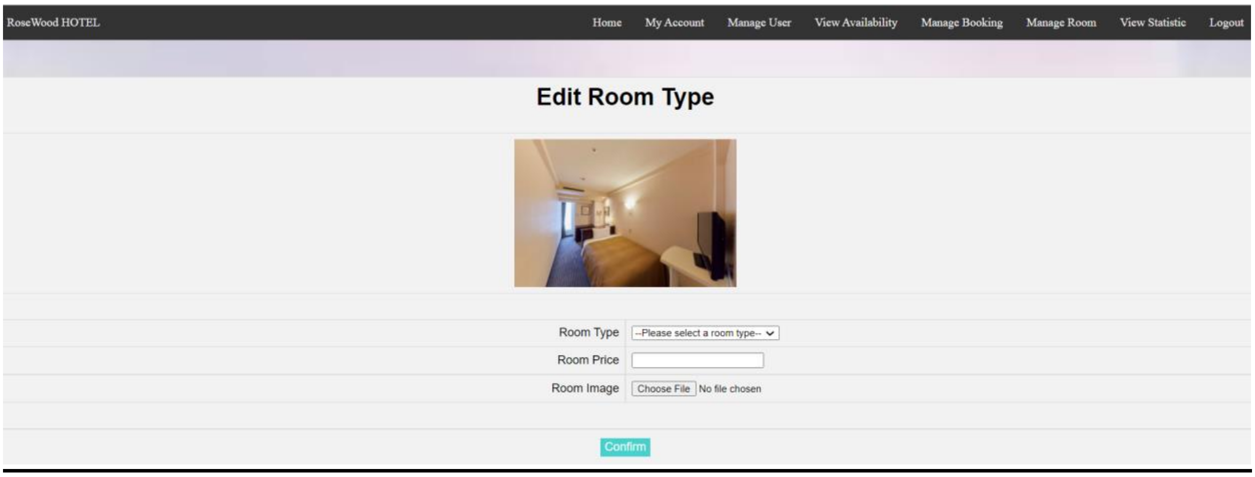


Add New Room / Hall

Room Type	Single Room
Floor	Ground Floor
Room No	
Room Activate	☑

Figure 4.15 Admin Add New Room Page

4.2.2.2.8 Admin Edit Room Type Page

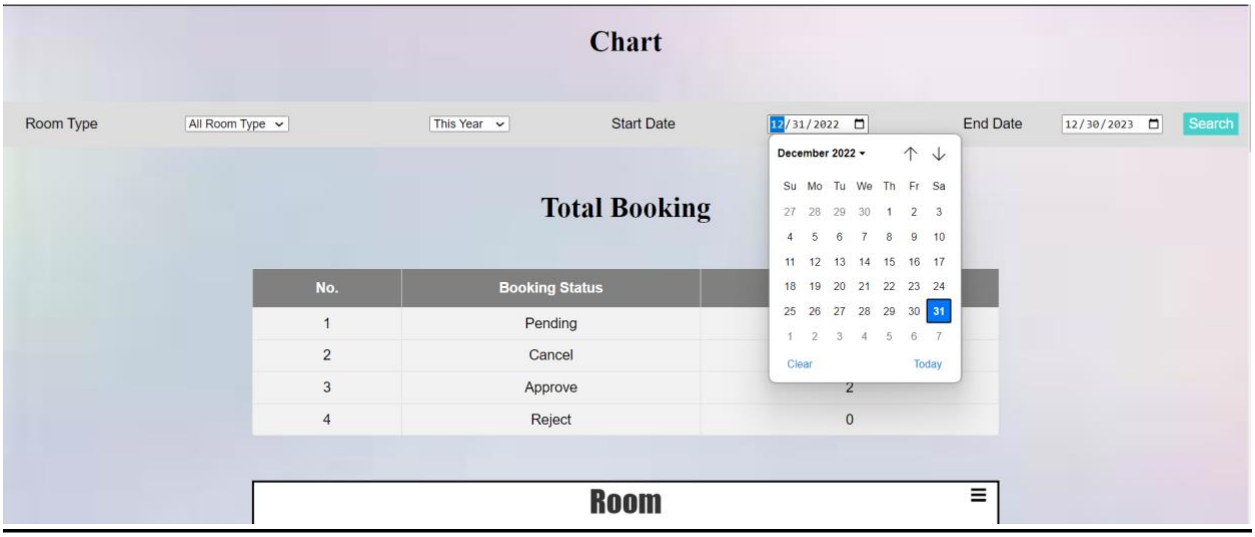


Edit Room Type

Room Type	--Please select a room type--
Room Price	
Room Image	No file chosen

Figure 4.16 Admin Edit Room Type Page

4.2.2.2.9 Admin Generate Report Page



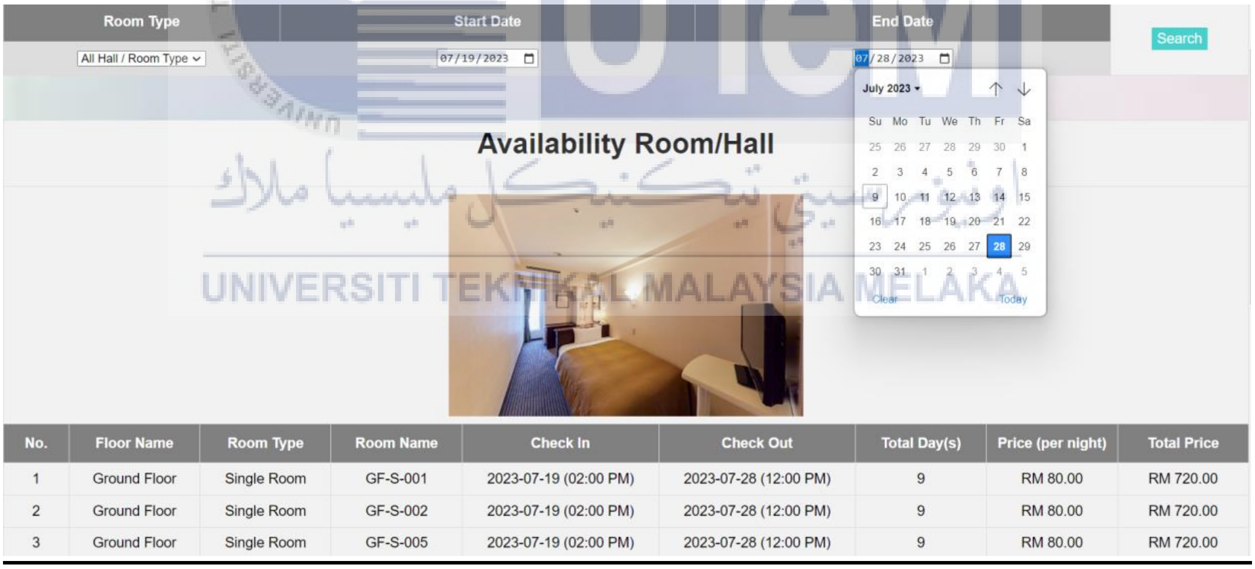
Room Type: All Room Type, This Year
Start Date: 12/31/2022
End Date: 12/30/2023
Search

Total Booking

No.	Booking Status	
1	Pending	
2	Cancel	
3	Approve	2
4	Reject	0

Figure 4.17 Admin Generate Report Page

4.2.2.2.10 Admin View Availability Page



Room Type	Start Date	End Date
All Hall / Room Type	07/19/2023	07/28/2023

Search

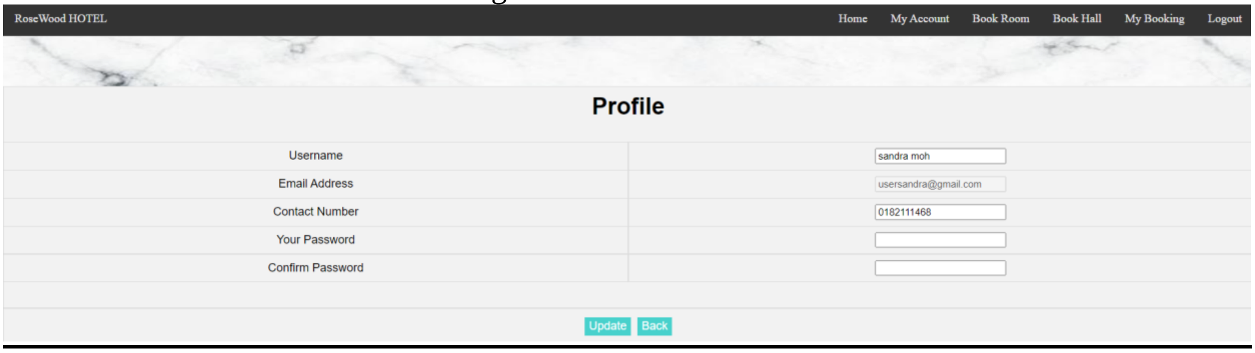
Availability Room/Hall

No.	Floor Name	Room Type	Room Name	Check In	Check Out	Total Day(s)	Price (per night)	Total Price
1	Ground Floor	Single Room	GF-S-001	2023-07-19 (02:00 PM)	2023-07-28 (12:00 PM)	9	RM 80.00	RM 720.00
2	Ground Floor	Single Room	GF-S-002	2023-07-19 (02:00 PM)	2023-07-28 (12:00 PM)	9	RM 80.00	RM 720.00
3	Ground Floor	Single Room	GF-S-005	2023-07-19 (02:00 PM)	2023-07-28 (12:00 PM)	9	RM 80.00	RM 720.00

Figure 4.18 Admin View Availability Page

User Page

4.2.2.2.11 User Edit Profile Page

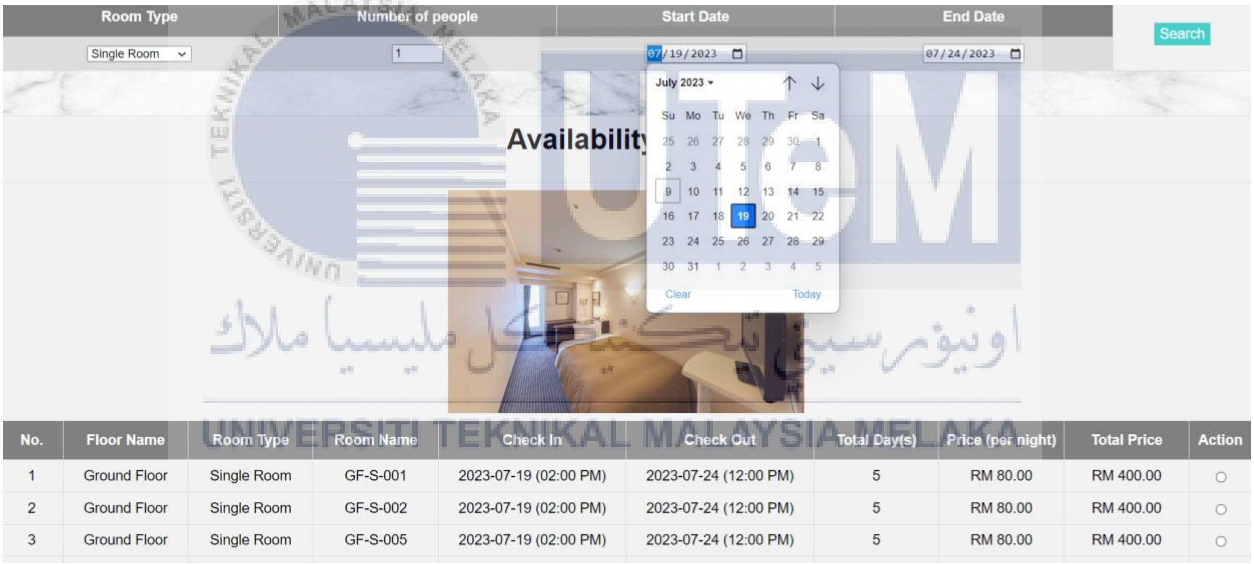


Profile

Username	sandra moh
Email Address	usersandra@gmail.com
Contact Number	0182111468
Your Password	
Confirm Password	

Figure 4.19 User Edit Profile Page

4.2.2.2.12 User Insert Booking Information Page (Room)

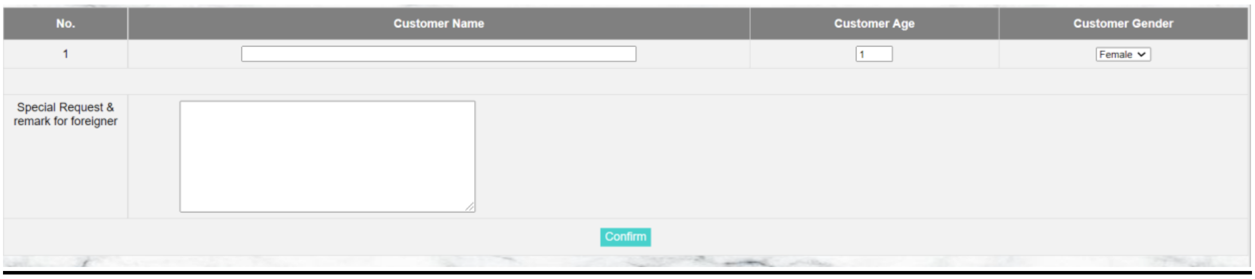


Room Type	Number of people	Start Date	End Date
Single Room	1	07/19/2023	07/24/2023

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

No.	Floor Name	Room Type	Room Name	Check In	Check Out	Total Day(s)	Price (per night)	Total Price	Action
1	Ground Floor	Single Room	GF-S-001	2023-07-19 (02:00 PM)	2023-07-24 (12:00 PM)	5	RM 80.00	RM 400.00	○
2	Ground Floor	Single Room	GF-S-002	2023-07-19 (02:00 PM)	2023-07-24 (12:00 PM)	5	RM 80.00	RM 400.00	○
3	Ground Floor	Single Room	GF-S-005	2023-07-19 (02:00 PM)	2023-07-24 (12:00 PM)	5	RM 80.00	RM 400.00	○

Figure 4.20 User Insert Booking Information Page (Room)

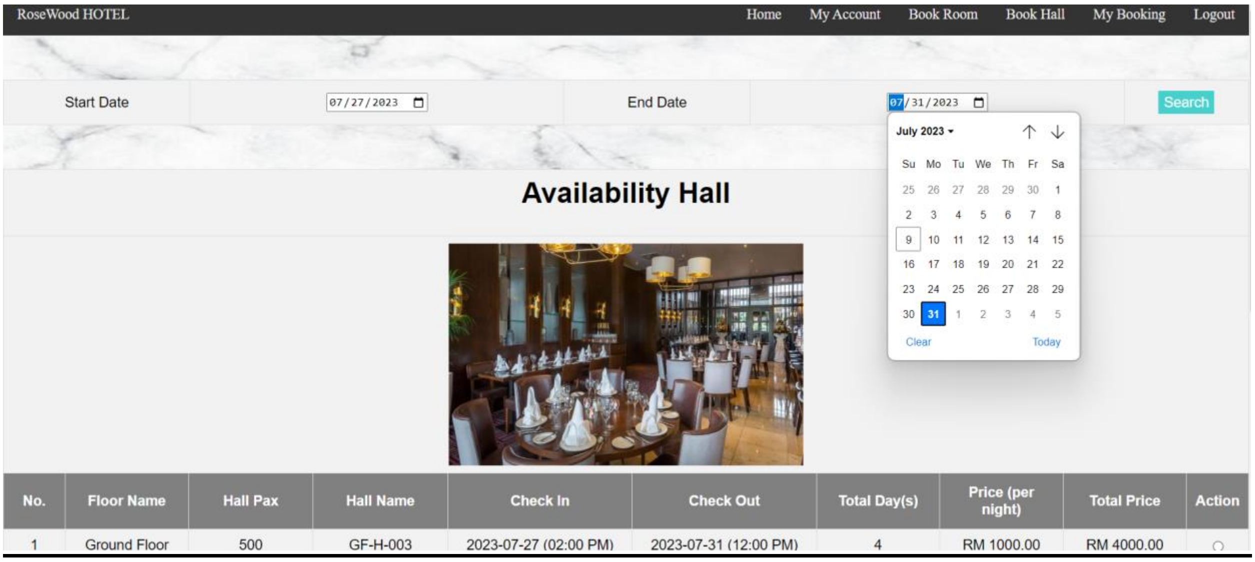


No.	Customer Name	Customer Age	Customer Gender
1		1	Female

Special Request & remark for foreigner

Figure 4.21 User Insert Booking Information Page (Room)

4.2.2.2.13 User Insert Booking Information Page (Hall)

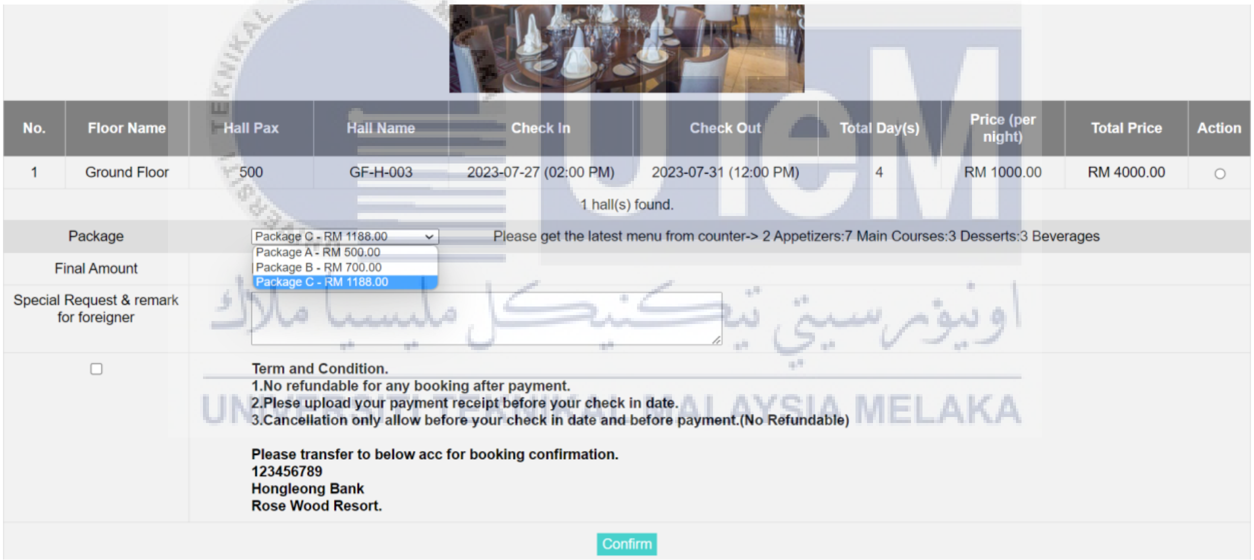


Start Date 07/27/2023 End Date 07/31/2023

Availability Hall

No.	Floor Name	Hall Pax	Hall Name	Check In	Check Out	Total Day(s)	Price (per night)	Total Price	Action
1	Ground Floor	500	GF-H-003	2023-07-27 (02:00 PM)	2023-07-31 (12:00 PM)	4	RM 1000.00	RM 4000.00	○

Figure 4.22 User Insert Booking Information Page (Hall)



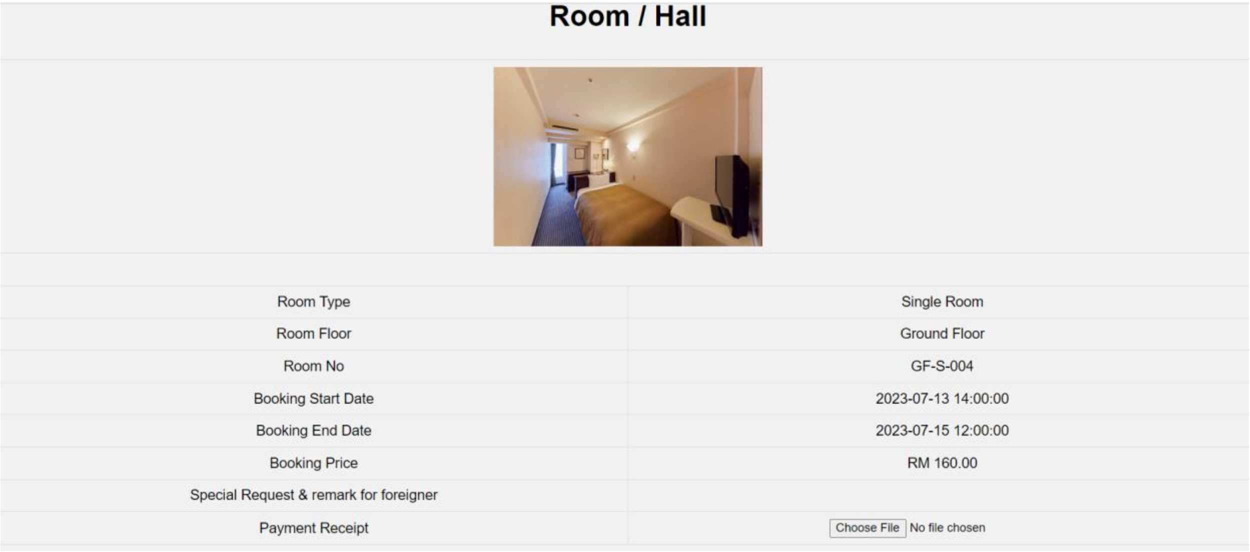
No.	Floor Name	Hall Pax	Hall Name	Check In	Check Out	Total Day(s)	Price (per night)	Total Price	Action
1	Ground Floor	500	GF-H-003	2023-07-27 (02:00 PM)	2023-07-31 (12:00 PM)	4	RM 1000.00	RM 4000.00	○

1 hall(s) found.

Package	Package C - RM 1188.00 Package A - RM 500.00 Package B - RM 700.00 Package C - RM 1188.00	Please get the latest menu from counter-> 2 Appetizers:7 Main Courses:3 Desserts:3 Beverages
Final Amount		
Special Request & remark for foreigner		
□	**Term and Condition.** **1.No refundable for any booking after payment.** **2.Plese upload your payment receipt before your check in date.** **3.Cancellation only allow before your check in date and before payment.(No Refundable)** **Please transfer to below acc for booking confirmation.** **123456789** **Hongleong Bank** **Rose Wood Resort.**	

Figure 4.23 User Insert Booking Information Page (Hall)

4.2.2.2.14 User Upload Payment Page



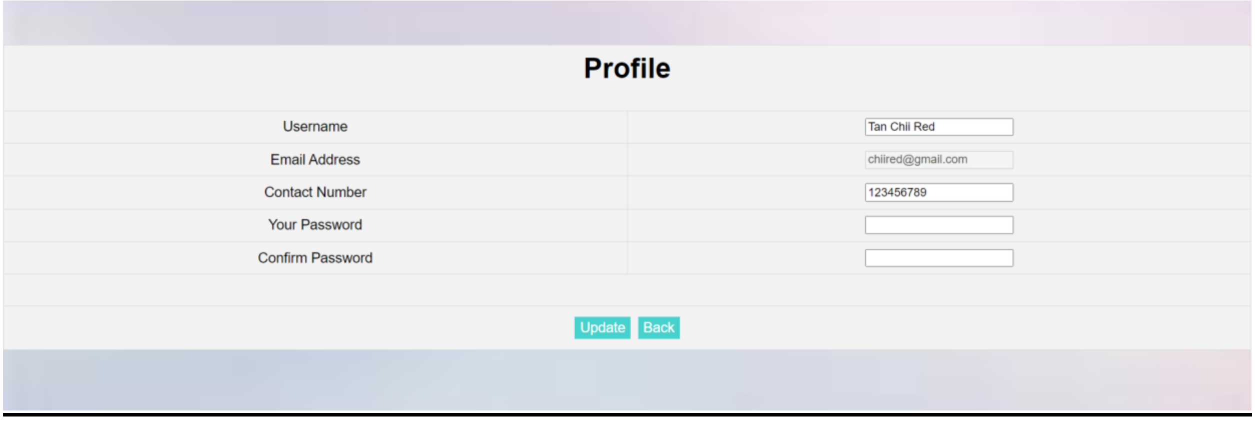
Room / Hall

Room Type	Single Room
Room Floor	Ground Floor
Room No	GF-S-004
Booking Start Date	2023-07-13 14:00:00
Booking End Date	2023-07-15 12:00:00
Booking Price	RM 160.00
Special Request & remark for foreigner	
Payment Receipt	Choose File No file chosen

Figure 4.24 User Upload Payment Page

Staff Page

4.2.2.2.15 Staff Update Profile Page

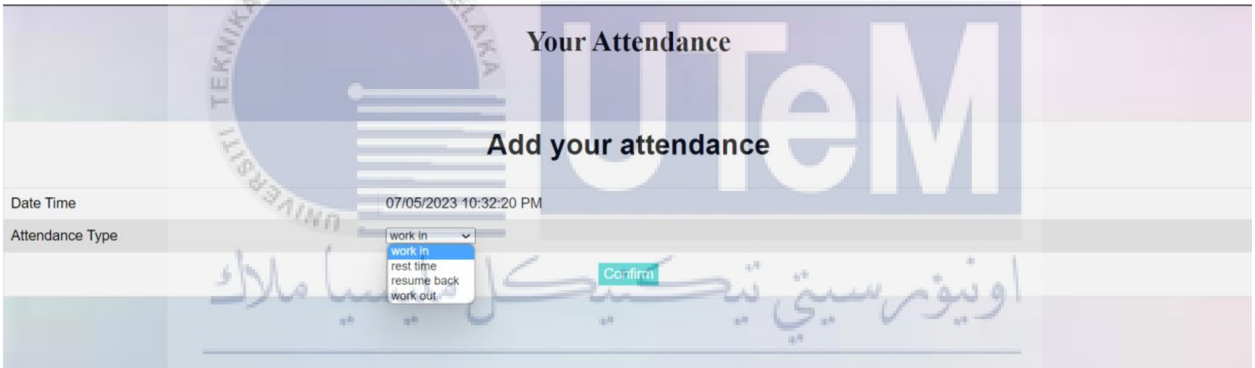


Profile

Username	Tan Chii Red
Email Address	chiired@gmail.com
Contact Number	123456789
Your Password	
Confirm Password	

Figure 4.25 Staff Update Profile Page

4.2.2.2.16 Staff Add Attendance Page



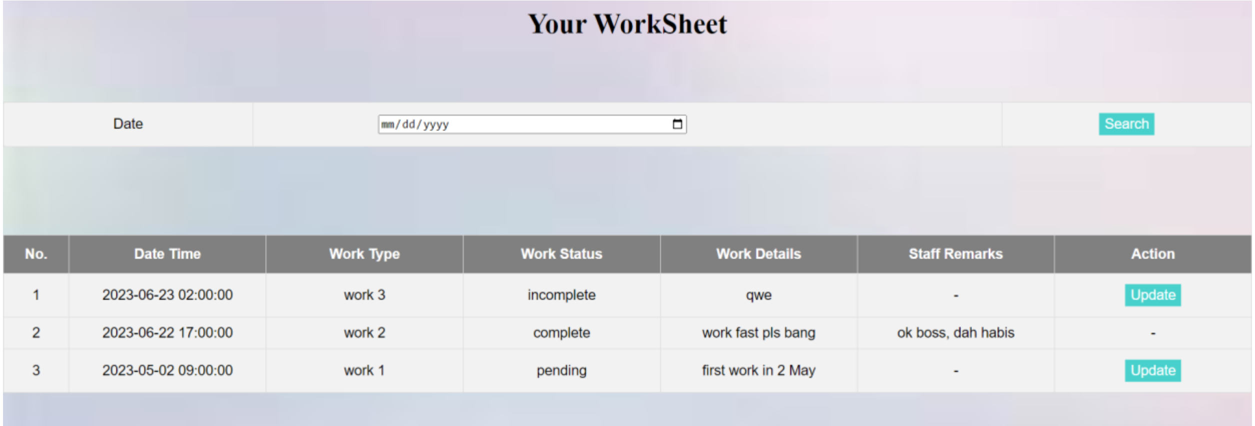
Your Attendance

Add your attendance

Date Time	07/05/2023 10:32:20 PM
Attendance Type	work in

Figure 4.26 Staff Add Attendance Page

4.2.2.2.17 Staff Update Work Progress Page



Your WorkSheet

Date	mm/dd/yyyy	Search

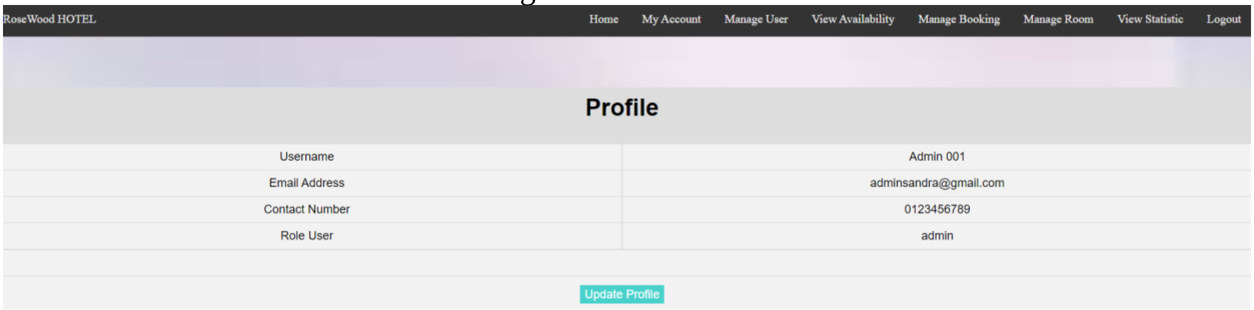
No.	Date Time	Work Type	Work Status	Work Details	Staff Remarks	Action
1	2023-06-23 02:00:00	work 3	incomplete	qwe	-	Update
2	2023-06-22 17:00:00	work 2	complete	work fast pls bang	ok boss, dah habis	-
3	2023-05-02 09:00:00	work 1	pending	first work in 2 May	-	Update

Figure 4.27 Staff Update Work Progress Page

4.2.2.3 Output Design

Admin Page

4.2.2.3.1 Admin View Profile Page

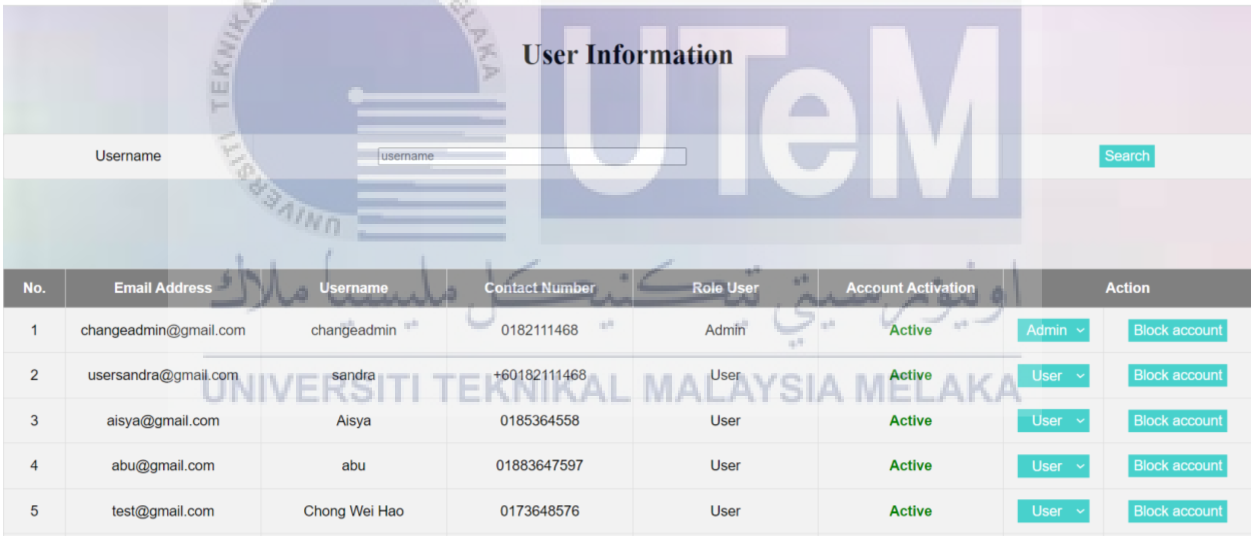


Profile

Username	Admin 001
Email Address	adminsandra@gmail.com
Contact Number	0123456789
Role User	admin

Figure 4.28 Admin View Profile Page

4.2.2.3.2 Admin Manage User Page (View Information)

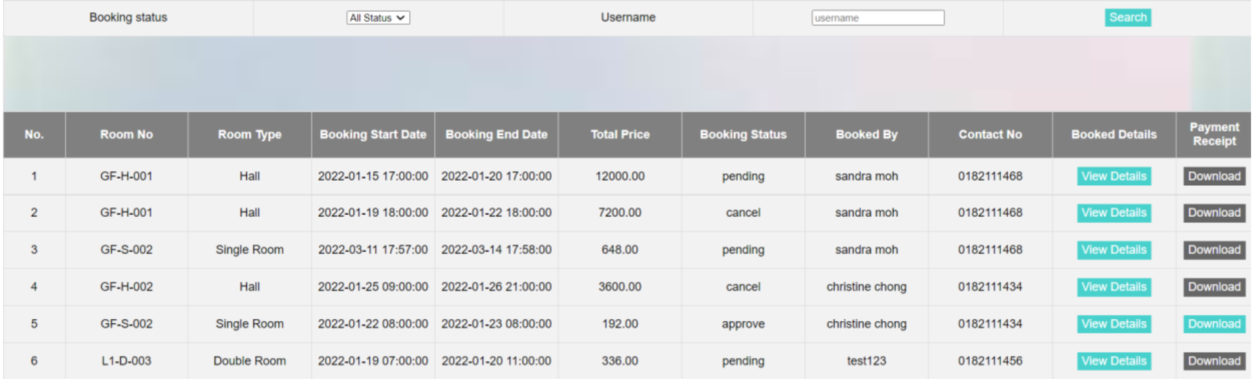


User Information

No.	Email Address	Username	Contact Number	Role User	Account Activation	Action	
1	changeadmin@gmail.com	changeadmin	0182111468	Admin	Active	Admin	Block account
2	usersandra@gmail.com	sandra	+60182111468	User	Active	User	Block account
3	aisya@gmail.com	Aisya	0185364558	User	Active	User	Block account
4	abu@gmail.com	abu	01883647597	User	Active	User	Block account
5	test@gmail.com	Chong Wei Hao	0173648576	User	Active	User	Block account

Figure 4.29 Admin Manage User Page (View Information)

4.2.2.3.3 Admin Manage Booking Page (View Information)

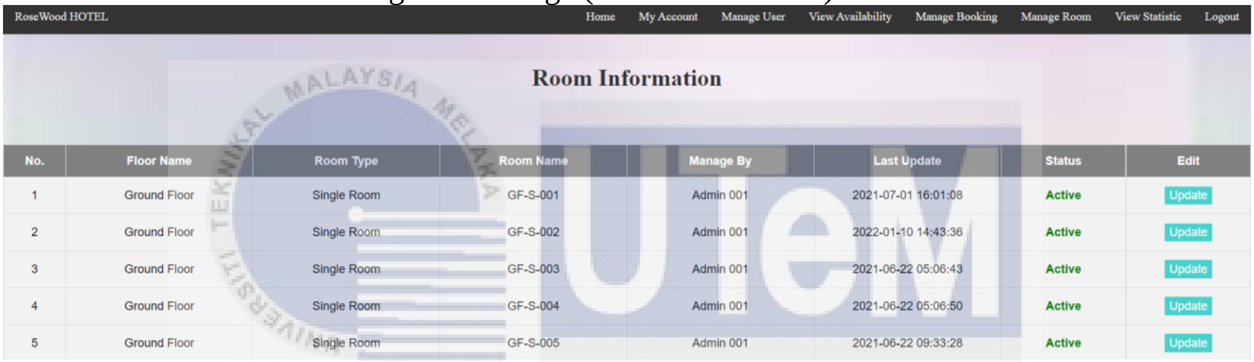


Booking status	All Status	Username	username	Search

No.	Room No	Room Type	Booking Start Date	Booking End Date	Total Price	Booking Status	Booked By	Contact No	Booked Details	Payment Receipt
1	GF-H-001	Hall	2022-01-15 17:00:00	2022-01-20 17:00:00	12000.00	pending	sandra moh	0182111468	View Details	Download
2	GF-H-001	Hall	2022-01-19 18:00:00	2022-01-22 18:00:00	7200.00	cancel	sandra moh	0182111468	View Details	Download
3	GF-S-002	Single Room	2022-03-11 17:57:00	2022-03-14 17:58:00	648.00	pending	sandra moh	0182111468	View Details	Download
4	GF-H-002	Hall	2022-01-25 09:00:00	2022-01-26 21:00:00	3600.00	cancel	christine chong	0182111434	View Details	Download
5	GF-S-002	Single Room	2022-01-22 08:00:00	2022-01-23 08:00:00	192.00	approve	christine chong	0182111434	View Details	Download
6	L1-D-003	Double Room	2022-01-19 07:00:00	2022-01-20 11:00:00	336.00	pending	test123	0182111456	View Details	Download

Figure 4.30 Admin Manage Booking Page (View Information)

4.2.2.3.4 Admin Manage Room Page (View Information)

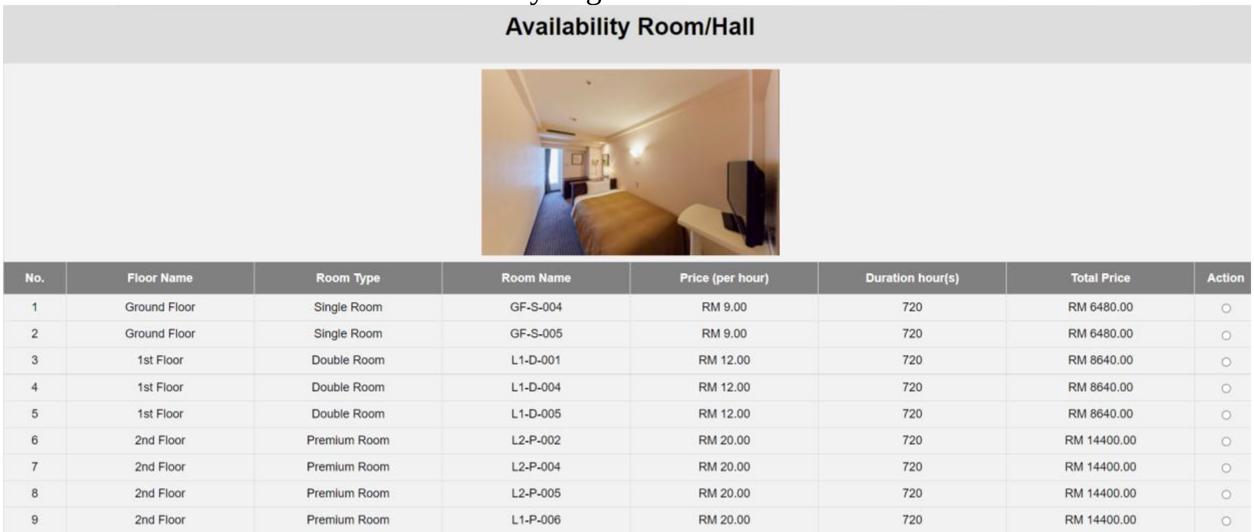


Room Information

No.	Floor Name	Room Type	Room Name	Manage By	Last Update	Status	Edit
1	Ground Floor	Single Room	GF-S-001	Admin 001	2021-07-01 16:01:08	Active	Update
2	Ground Floor	Single Room	GF-S-002	Admin 001	2022-01-10 14:43:36	Active	Update
3	Ground Floor	Single Room	GF-S-003	Admin 001	2021-06-22 05:06:43	Active	Update
4	Ground Floor	Single Room	GF-S-004	Admin 001	2021-06-22 05:06:50	Active	Update
5	Ground Floor	Single Room	GF-S-005	Admin 001	2021-06-22 09:33:28	Active	Update

Figure 4.31 Admin Manage Room Page (View Information)

4.2.2.3.5 Admin View Availability Page

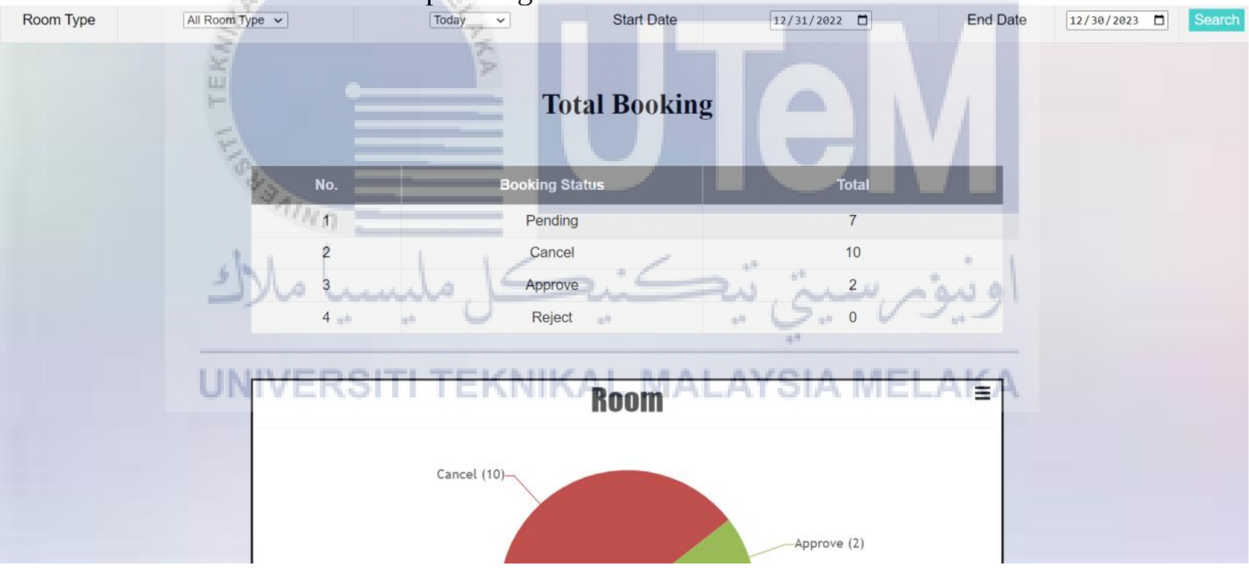


Availability Room/Hall

No.	Floor Name	Room Type	Room Name	Price (per hour)	Duration hour(s)	Total Price	Action
1	Ground Floor	Single Room	GF-S-004	RM 9.00	720	RM 6480.00	○
2	Ground Floor	Single Room	GF-S-005	RM 9.00	720	RM 6480.00	○
3	1st Floor	Double Room	L1-D-001	RM 12.00	720	RM 8640.00	○
4	1st Floor	Double Room	L1-D-004	RM 12.00	720	RM 8640.00	○
5	1st Floor	Double Room	L1-D-005	RM 12.00	720	RM 8640.00	○
6	2nd Floor	Premium Room	L2-P-002	RM 20.00	720	RM 14400.00	○
7	2nd Floor	Premium Room	L2-P-004	RM 20.00	720	RM 14400.00	○
8	2nd Floor	Premium Room	L2-P-005	RM 20.00	720	RM 14400.00	○
9	2nd Floor	Premium Room	L1-P-006	RM 20.00	720	RM 14400.00	○

Figure 4.32 Admin View Availability Page

4.2.2.3.6 Admin View Report Page



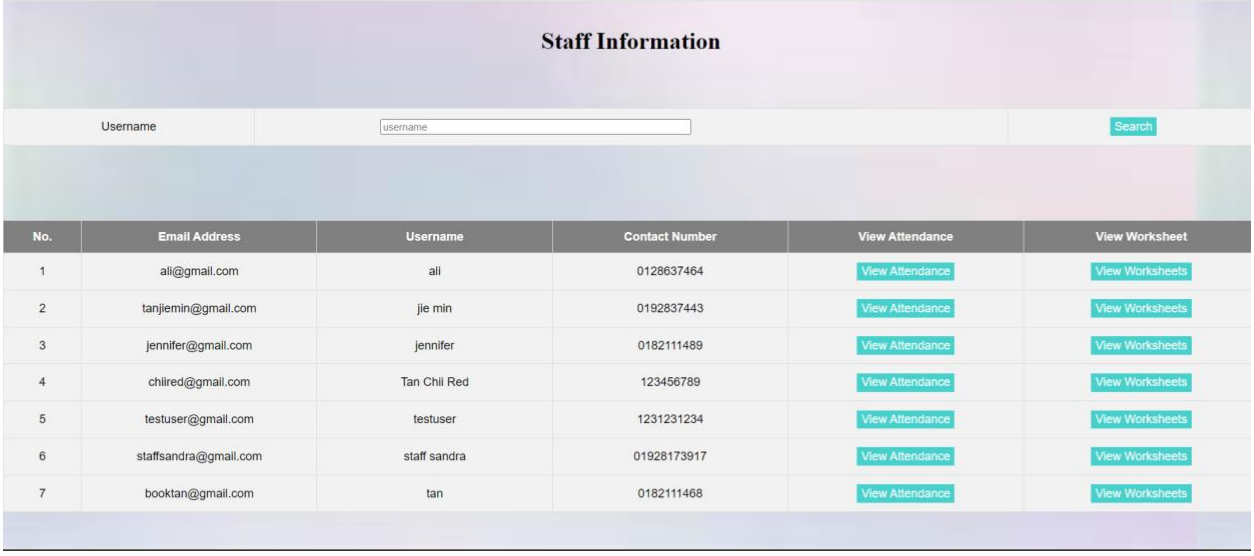
Total Booking

No.	Booking Status	Total
1	Pending	7
2	Cancel	10
3	Approve	2
4	Reject	0

Room

Figure 4.33 Admin View Report Page (View Report)

4.2.2.3.7 Admin View Staff Page



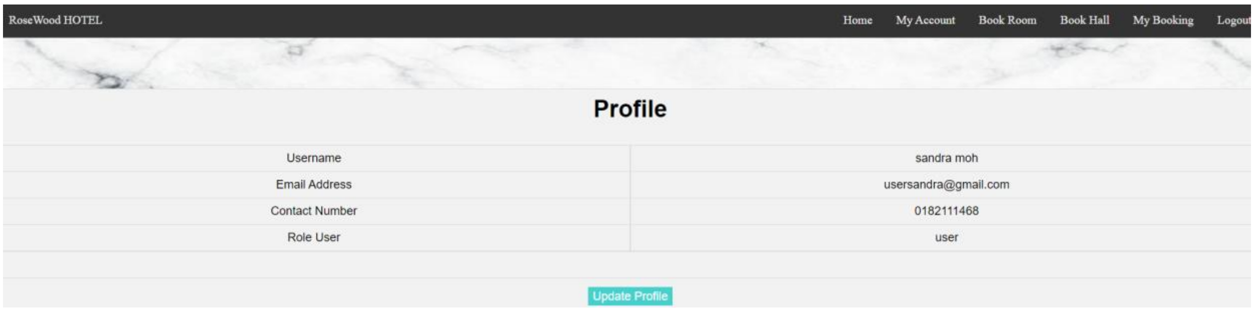
Staff Information

No.	Email Address	Username	Contact Number	View Attendance	View Worksheet
1	ali@gmail.com	ali	0128637464	View Attendance	View Worksheets
2	tanjiemin@gmail.com	jie min	0192837443	View Attendance	View Worksheets
3	jennifer@gmail.com	jennifer	0182111489	View Attendance	View Worksheets
4	chiired@gmail.com	Tan Chii Red	123456789	View Attendance	View Worksheets
5	testuser@gmail.com	testuser	1231231234	View Attendance	View Worksheets
6	staffsandra@gmail.com	staff sandra	01928173917	View Attendance	View Worksheets
7	booktan@gmail.com	tan	0182111468	View Attendance	View Worksheets

Figure 4.34 Admin View Staff Page

User Page

4.2.2.3.8 User View Profile Page

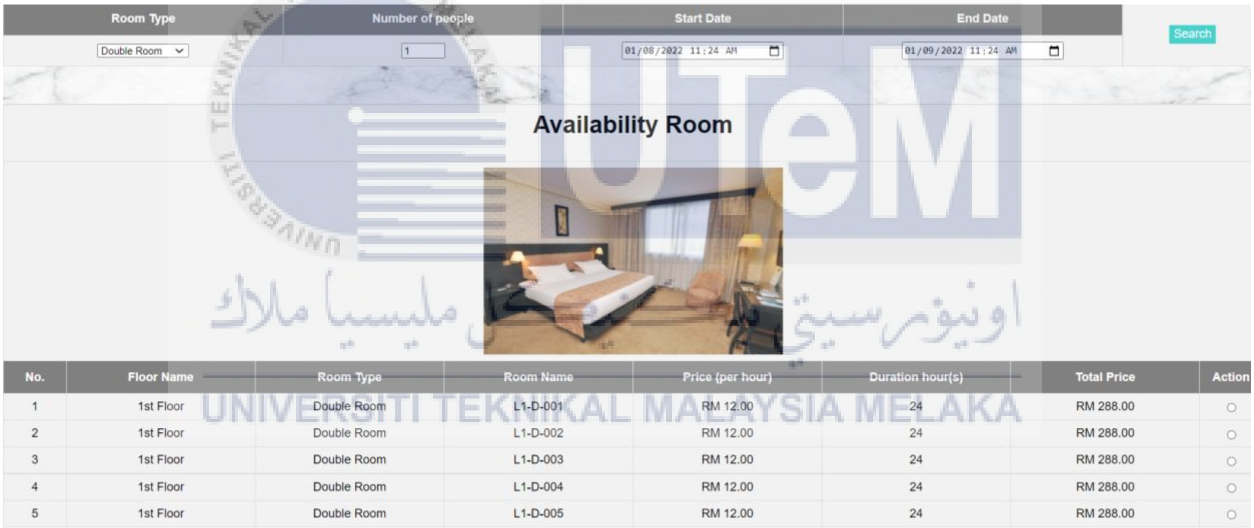


Profile

Username	sandra moh
Email Address	usersandra@gmail.com
Contact Number	0182111468
Role User	user

Figure 4.35 User View Profile Page

4.2.2.3.9 User Booking Room Page



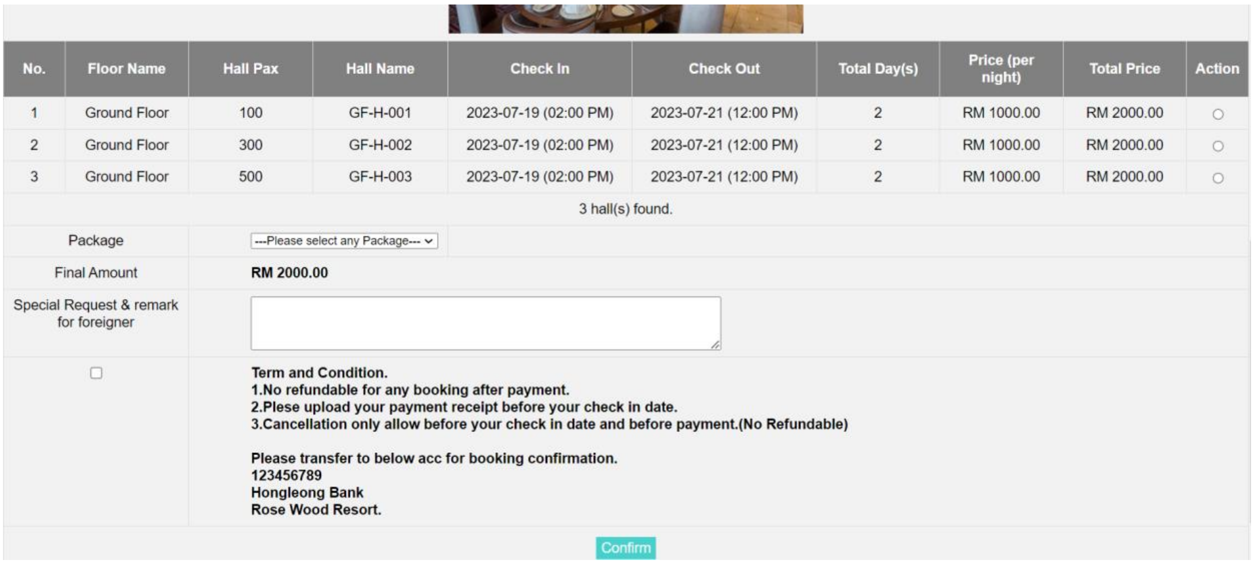
Room Type	Number of people	Start Date	End Date
Double Room	1	01/08/2022 11:24 AM	01/09/2022 11:24 AM

Availability Room

No.	Floor Name	Room Type	Room Name	Price (per hour)	Duration hour(s)	Total Price	Action
1	1st Floor	Double Room	L1-D-001	RM 12.00	24	RM 288.00	○
2	1st Floor	Double Room	L1-D-002	RM 12.00	24	RM 288.00	○
3	1st Floor	Double Room	L1-D-003	RM 12.00	24	RM 288.00	○
4	1st Floor	Double Room	L1-D-004	RM 12.00	24	RM 288.00	○
5	1st Floor	Double Room	L1-D-005	RM 12.00	24	RM 288.00	○

Figure 4.36 User Booking Room Page

4.2.2.3.10 User Booking Hall Page



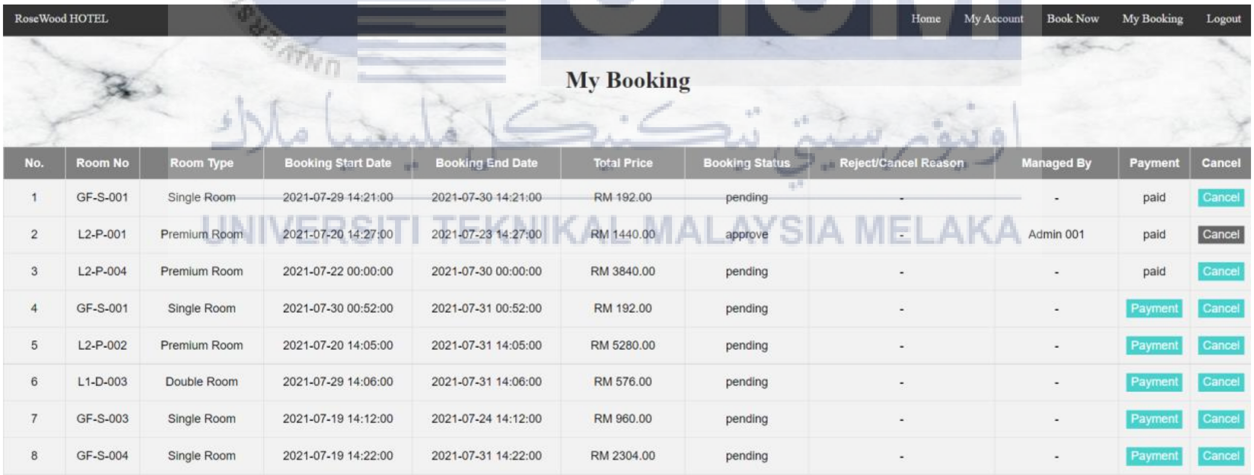
No.	Floor Name	Hall Pax	Hall Name	Check In	Check Out	Total Day(s)	Price (per night)	Total Price	Action
1	Ground Floor	100	GF-H-001	2023-07-19 (02:00 PM)	2023-07-21 (12:00 PM)	2	RM 1000.00	RM 2000.00	○
2	Ground Floor	300	GF-H-002	2023-07-19 (02:00 PM)	2023-07-21 (12:00 PM)	2	RM 1000.00	RM 2000.00	○
3	Ground Floor	500	GF-H-003	2023-07-19 (02:00 PM)	2023-07-21 (12:00 PM)	2	RM 1000.00	RM 2000.00	○

3 hall(s) found.

Package	---Please select any Package---
Final Amount	**RM 2000.00**
Special Request & remark for foreigner	
□	**Term and Condition.** **1.No refundable for any booking after payment.** **2.Plese upload your payment receipt before your check in date.** **3.Cancellation only allow before your check in date and before payment.(No Refundable)** **Please transfer to below acc for booking confirmation.** **123456789** **Hongleong Bank** **Rose Wood Resort.**

Figure 4.37 User Booking Hall Page

4.2.2.3.11 User View My Booking Page



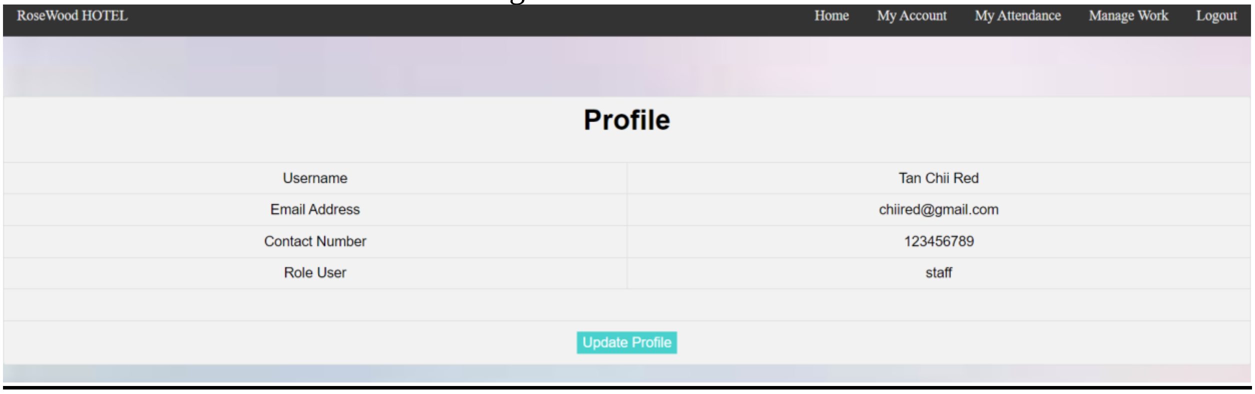
My Booking

No.	Room No	Room Type	Booking Start Date	Booking End Date	Total Price	Booking Status	Reject/Cancel Reason	Managed By	Payment	Cancel
1	GF-S-001	Single Room	2021-07-29 14:21:00	2021-07-30 14:21:00	RM 192.00	pending	-	-	paid	Cancel
2	L2-P-001	Premium Room	2021-07-20 14:27:00	2021-07-23 14:27:00	RM 1440.00	approve	-	Admin 001	paid	Cancel
3	L2-P-004	Premium Room	2021-07-22 00:00:00	2021-07-30 00:00:00	RM 3840.00	pending	-	-	paid	Cancel
4	GF-S-001	Single Room	2021-07-30 00:52:00	2021-07-31 00:52:00	RM 192.00	pending	-	-	Payment	Cancel
5	L2-P-002	Premium Room	2021-07-20 14:05:00	2021-07-31 14:05:00	RM 5280.00	pending	-	-	Payment	Cancel
6	L1-D-003	Double Room	2021-07-29 14:06:00	2021-07-31 14:06:00	RM 576.00	pending	-	-	Payment	Cancel
7	GF-S-003	Single Room	2021-07-19 14:12:00	2021-07-24 14:12:00	RM 960.00	pending	-	-	Payment	Cancel
8	GF-S-004	Single Room	2021-07-19 14:22:00	2021-07-31 14:22:00	RM 2304.00	pending	-	-	Payment	Cancel

Figure 4.38 User View My Booking Page

Staff Page

4.2.2.3.12 Staff View Profile Page

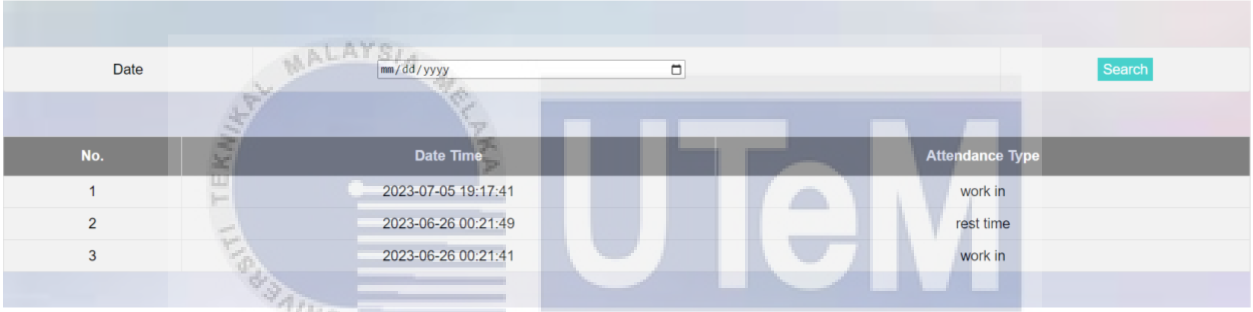


Profile

Username	Tan Chii Red
Email Address	chiired@gmail.com
Contact Number	123456789
Role User	staff

Figure 4.39 Staff View Profile Page

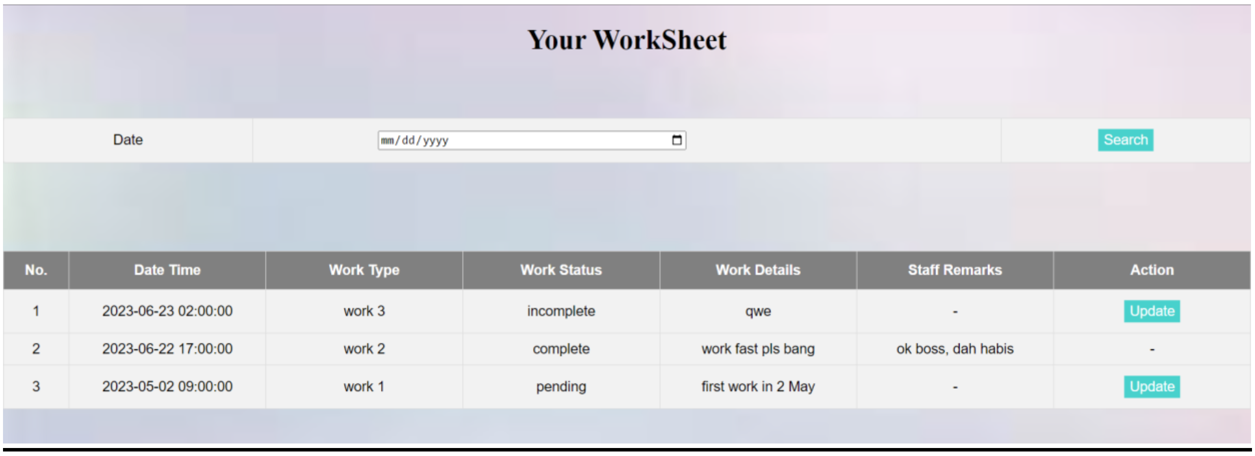
4.2.2.3.13 Staff View Attendance Page



No.	Date Time	Attendance Type
1	2023-07-05 19:17:41	work in
2	2023-06-26 00:21:49	rest time
3	2023-06-26 00:21:41	work in

Figure 4.40 Staff View Attendance Page

4.2.2.3.14 Staff View Work Progress Page



Your WorkSheet

Date	mm/dd/yyyy	Search

No.	Date Time	Work Type	Work Status	Work Details	Staff Remarks	Action
1	2023-06-23 02:00:00	work 3	incomplete	qwe	-	Update
2	2023-06-22 17:00:00	work 2	complete	work fast pls bang	ok boss, dah habis	-
3	2023-05-02 09:00:00	work 1	pending	first work in 2 May	-	Update

Figure 4.41 Staff View Work Progress Page

4.2.3 Conceptual and Logical Database Design

4.2.3.1 ERD

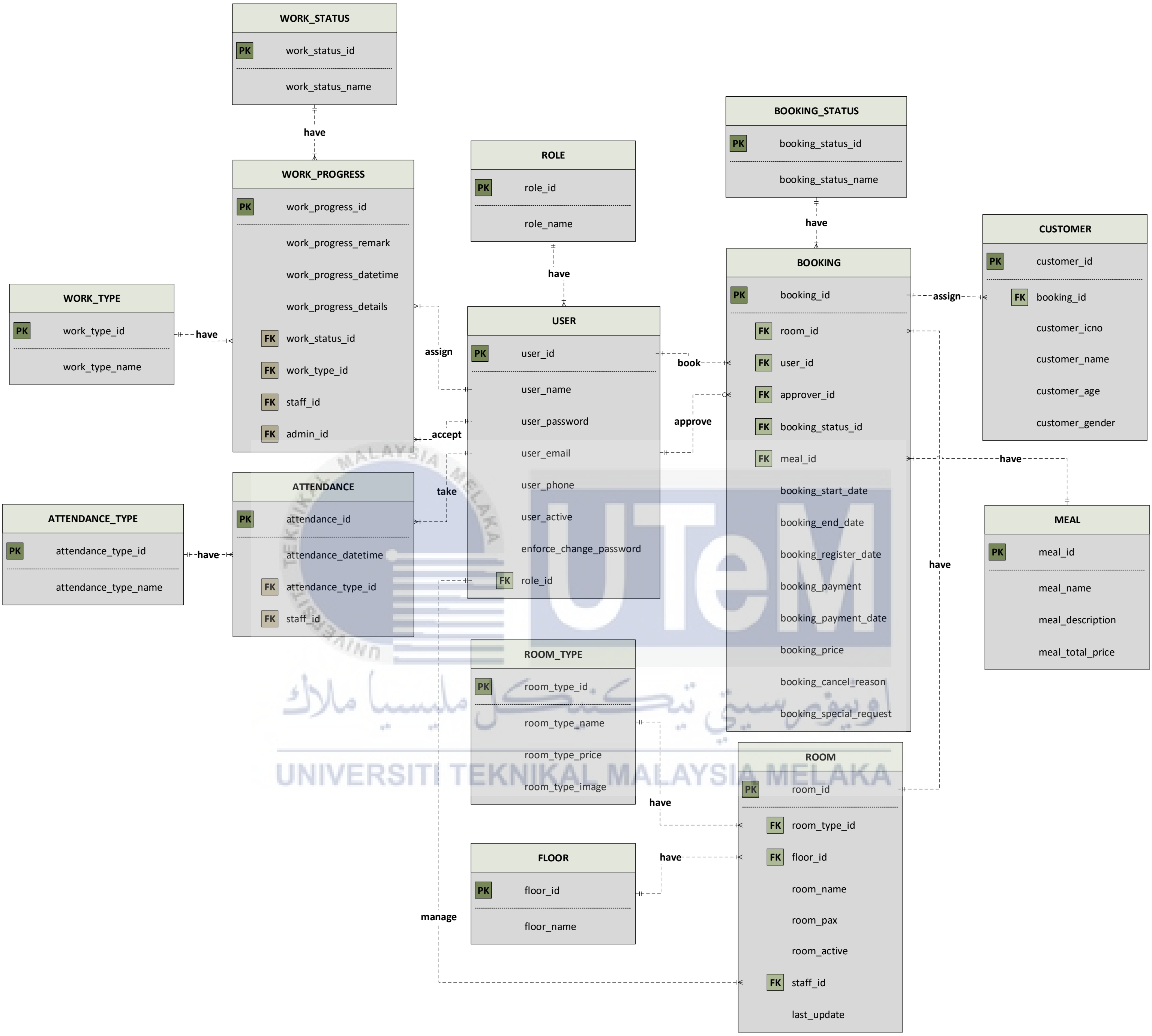


Figure 4.42: ERD of RoseWood Resort Reservation System

4.2.3.2 Business Rule

No	Description
1	A user have one and only one role
	A role have one or many user
2	A user assign one or many work progress
	A work progress assigned by one and only one user
3	A user accept one or many work progress
	A work progress accept by one user only.
4	A user take one or many attendance
	An attendance take by one user only.
5	A user book one or many booking.
	A booking book by one user only.
6	A user approve zero or many booking
	A booking approved by one user only.
7	A booking have only one booking status.
	A booking status have one or many booking.
8	A booking have assign by one or many customer.
	A customer assign by one booking only.
9	A booking have one meal only
	A meal have one or many booking.
10	A booking have one room only.
	A room have one or many booking
11	A room have only one room type.
	A room type have one or many room.
12	A room have only one floor.
	A floor have one or many room.
13	A user manage one or many room
	A room manage by one user only.
14	A work progress have one work status only
	A work status have one or more work progress.
15	A work type have one or many work progress
	A work progress have only one work type.

16	A attendance type have one or many attendance
	A attendance have one attendance only

Table 4.01 Business Rule

4.3 System Architecture

Figure 4.43 below illustrates the flowchart of main processes from this system. It reveals the actions and processes that are to be associated with the main users in this system.

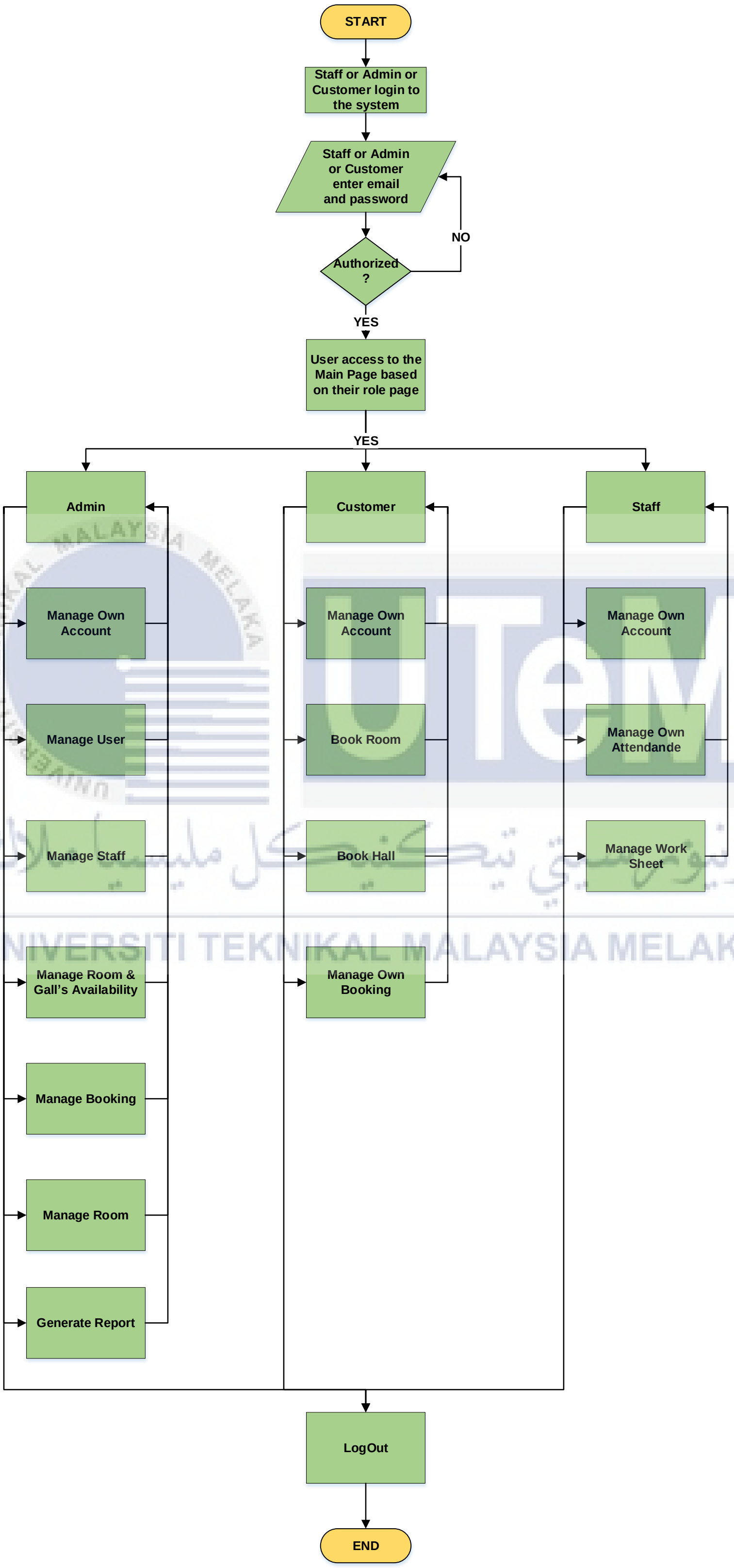


Figure 4.43 Flowchart of the System

4.3.1 Module Specification

Table 4.0 Login

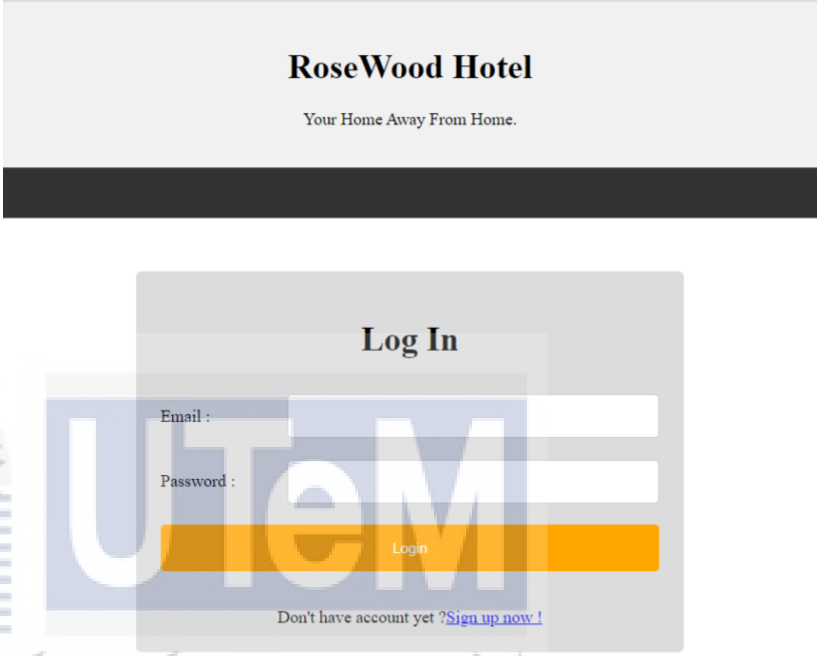
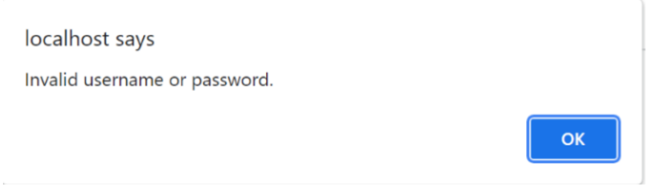
No of Module:	1
Module Name:	Login
Purpose:	To allow admin and user to log in and use the system.
Input:	[EMAIL] [PASSWORD]
Input Screen:	 RoseWood Hotel Your Home Away From Home. Log In Email : Password : Login Don't have account yet ?Sign up now !
Output:	Output : Warning message will pop out if key in wrong user ID and password, else go to the respective interface based on the user's role  localhost says Invalid username or password. OK

Table 4.1 Register

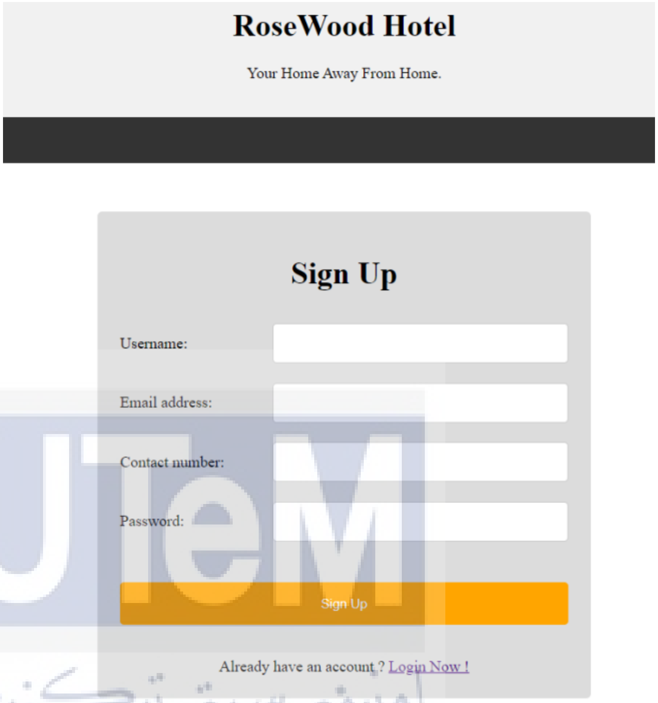
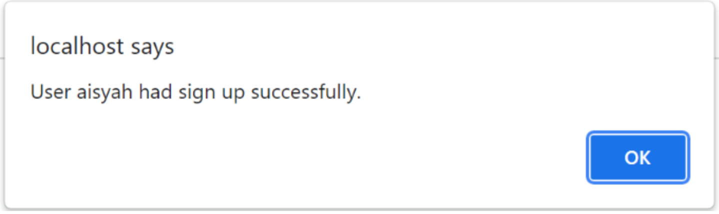
No of Module:	2
Module Name:	Register
Purpose:	To register user so that user able to access system.
Input:	[USERNAME] [EMAIL] [CONTACTNUMBER] [PASSWORD]
Input Screen:	 RoseWood Hotel Your Home Away From Home. Sign Up Username: Email address: Contact number: Password: Sign Up Already have an account ? Login Now !
Output:	Output : Message box will pop out, if the user successfully registers on system and will redirect to login page  localhost says User aisyah had sign up successfully. OK

Table 4.2 View My Account

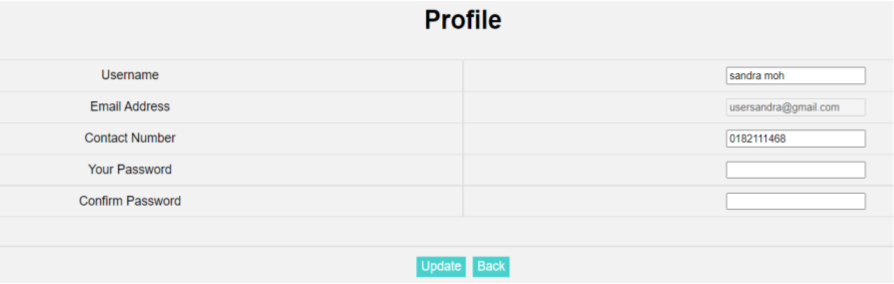
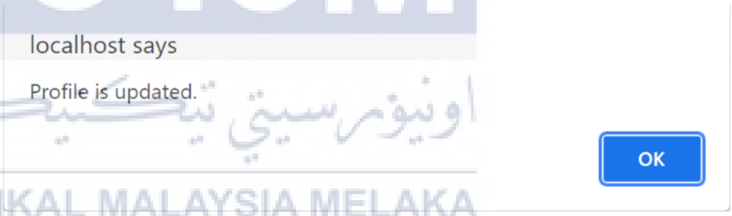
No of Module:	3
Module Name:	Update Account Information
Purpose:	To update personal information and profile
Input:	[USERNAME] [CONTACTNUMBER] [NEWPASSWORD] [CONFIRMPASSWORD]
Input Screen:	 Profile Username: sandra moh Email Address: usersandra@gmail.com Contact Number: 0182111468 Your Password: Confirm Password:
Output:	Output : Update user information.  localhost says Profile is updated.

Table 4.3 Manage User

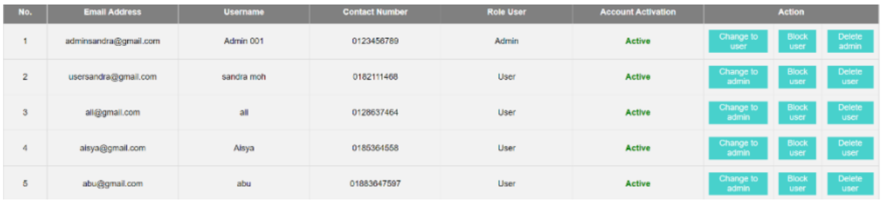
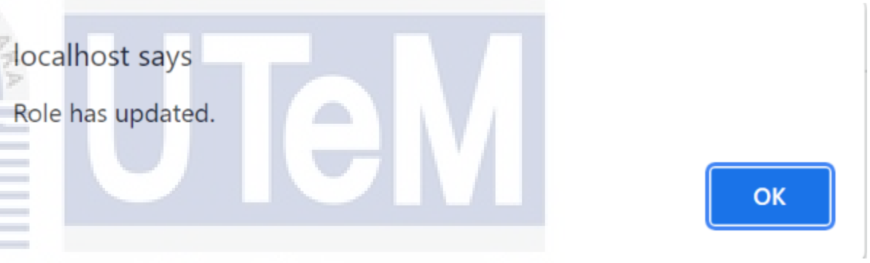
<table>
<tr><td><b>No of Module:</b></td><td>4</td></tr>
<tr><td><b>Module Name:</b></td><td>Manage User</td></tr>
<tr><td><b>Purpose:</b></td><td>To allow admin to manage user.</td></tr>
<tr><td><b>Input:</b></td><td>-</td></tr>
<tr><td><b>Input Screen:</b></td><td>
<table>
<tr><th>No.</th><th>Email Address</th><th>Username</th><th>Contact Number</th><th>Role User</th><th>Account Activation</th><th>Action</th><th></th><th></th></tr>
<tr><td>1</td><td>adminsandra@gmail.com</td><td>Admin 001</td><td>0123456789</td><td>Admin</td><td>Active</td><td>Change to user</td><td>Block user</td><td>Delete admin</td></tr>
<tr><td>2</td><td>usersandra@gmail.com</td><td>sandra moh</td><td>0182111468</td><td>User</td><td>Active</td><td>Change to admin</td><td>Block user</td><td>Delete user</td></tr>
<tr><td>3</td><td>ali@gmail.com</td><td>ali</td><td>0128637464</td><td>User</td><td>Active</td><td>Change to admin</td><td>Block user</td><td>Delete user</td></tr>
<tr><td>4</td><td>aisya@gmail.com</td><td>Aisya</td><td>0185364558</td><td>User</td><td>Active</td><td>Change to admin</td><td>Block user</td><td>Delete user</td></tr>
<tr><td>5</td><td>abu@gmail.com</td><td>abu</td><td>01883647597</td><td>User</td><td>Active</td><td>Change to admin</td><td>Block user</td><td>Delete user</td></tr>
</table></td></tr>
<tr><td><b>Output:</b></td><td>Output : Update user information.

localhost says
Role has updated.
 </td></tr>
</table>

Table 4.4 Manage Booking

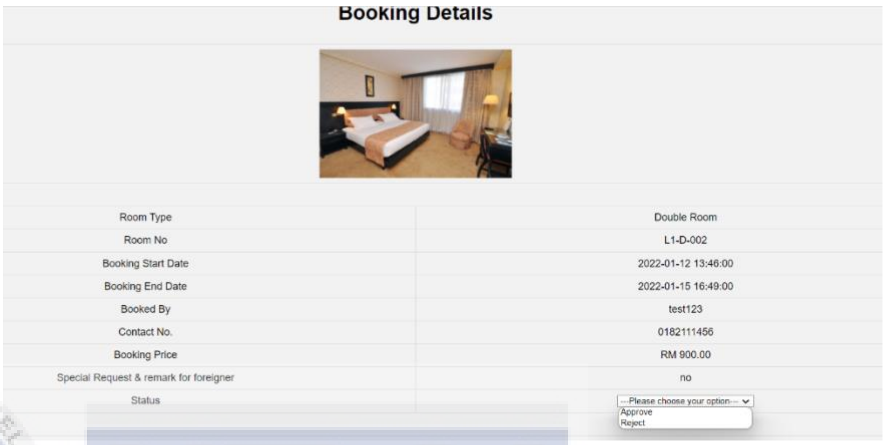
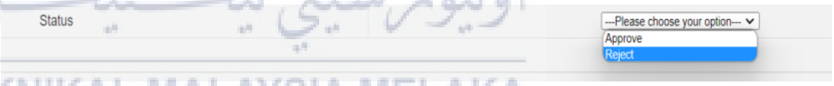
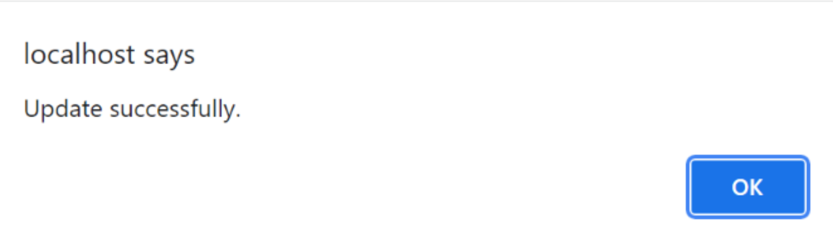
No of Module:	5
Module Name:	Manage Booking
Purpose:	To allow admin manage booking .
Input:	[ROOMAPPROVESTATUS]
Input Screen:	 Booking Details Room Type: Double Room Room No: L1-D-002 Booking Start Date: 2022-01-12 13:46:00 Booking End Date: 2022-01-15 16:49:00 Booked By: test123 Contact No.: 0182111456 Booking Price: RM 900.00 Special Request & remark for foreigner: no Status: ---Please choose your option---
Output:	Output : Update booking status. (Reject or approve)  Status: ---Please choose your option---  localhost says Update successfully.

Table 4.5 Manage Room

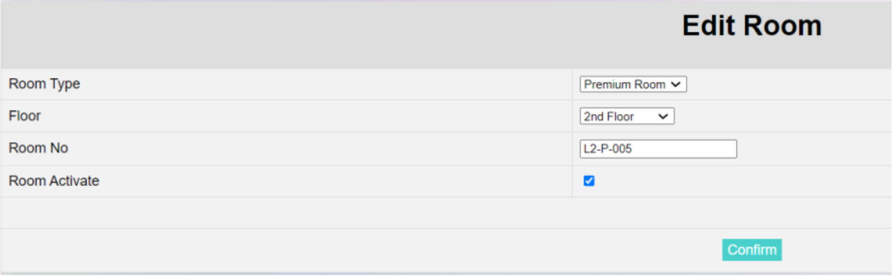
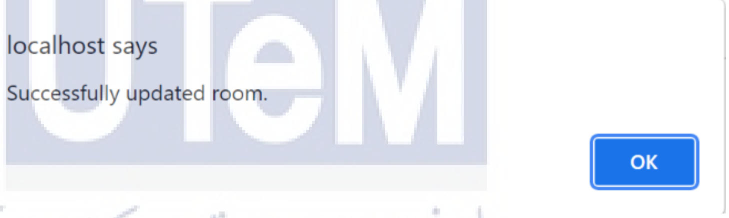
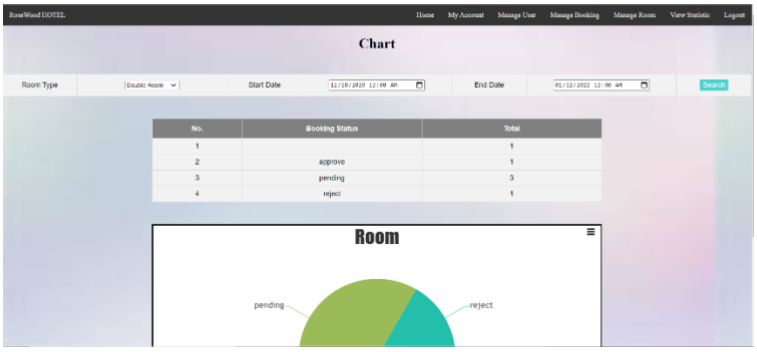
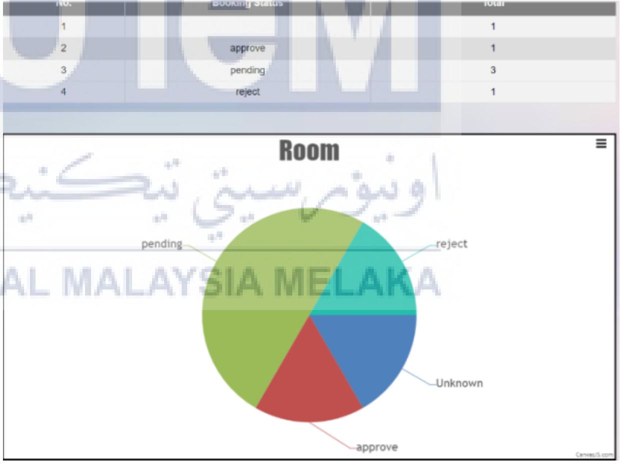
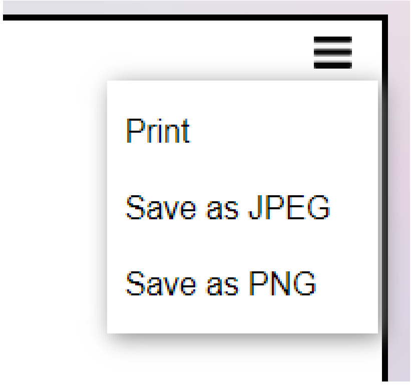
No of Module:	6
Module Name:	Manage Room
Purpose:	To allow admin manage room.
Input:	[ROOMTYPE] [FLOOR] [ROOMNO] [ROOM_ACTIVE]
Input Screen:	 Edit Room Room Type: Premium Room Floor: 2nd Floor Room No: L2-P-005 Room Activate: ☑ Confirm
Output:	Output : Update room information.  localhost says Successfully updated room. OK

Table 4.6 Generate Report

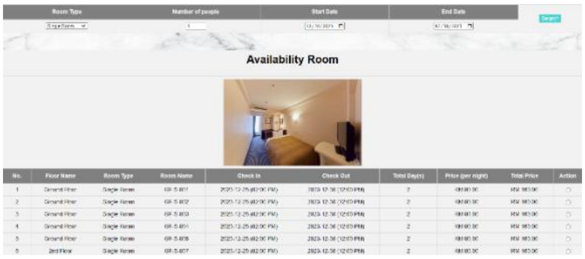
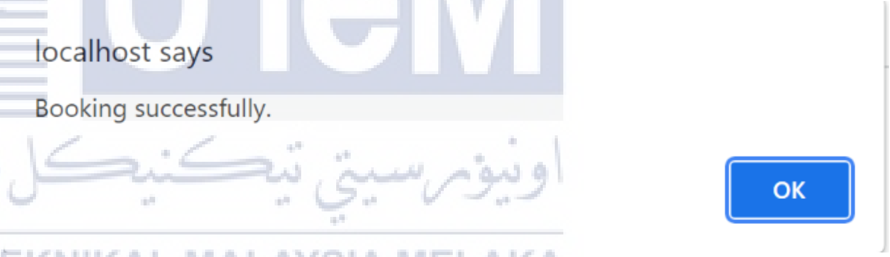
<table>
<tr><td><b>No of Module:</b></td><td>7</td></tr>
<tr><td><b>Module Name:</b></td><td>Generate Report</td></tr>
<tr><td><b>Purpose:</b></td><td>To view the statistic and save the report.</td></tr>
<tr><td><b>Input:</b></td><td>[ROOMTYPE] [STARTDATE] [ENDDATE]</td></tr>
<tr><td><b>Input Screen:</b></td><td>

RoseWood HOTEL
Home My Account Manage User Manage Booking Manage Room View Statistic Logout
Chart
Room Type: Double Room
Start Date: 12/18/2020 12:00 AM
End Date: 01/12/2022 12:00 AM
Search
<table>
<tr><th>No.</th><th>Booking Status</th><th>Total</th></tr>
<tr><td>1</td><td></td><td>1</td></tr>
<tr><td>2</td><td>approve</td><td>1</td></tr>
<tr><td>3</td><td>pending</td><td>3</td></tr>
<tr><td>4</td><td>reject</td><td>1</td></tr>
</table>

</td></tr>
<tr><td><b>Output:</b></td><td>
Output : View and Save report .

<table>
<tr><th>No.</th><th>Booking Status</th><th>Total</th></tr>
<tr><td>1</td><td></td><td>1</td></tr>
<tr><td>2</td><td>approve</td><td>1</td></tr>
<tr><td>3</td><td>pending</td><td>3</td></tr>
<tr><td>4</td><td>reject</td><td>1</td></tr>
</table>



</td></tr>
</table>

Table 4.7 Booking Room

No of Module:	8
Module Name:	Booking Room
Purpose:	To let user make a room reservation.
Input:	[ROOMTYPE] [NUMBEROFPEOPLE] [STARTDATE] [ENDDATE]
Input Screen:	
Output:	Output : Make a room reservation.  localhost says Booking successfully. OK

Room Type	Number of people	Start Date	End Date	
Single Room	1	12/25/2023	12/30/2023	Search

Availability Room

No.	Floor Name	Room Type	Room Name	Check In	Check Out	Total Day(s)	Price (per night)	Total Price	Action
1	Ground Floor	Single Room	GF-S-001	2023-12-25 (02:00 PM)	2023-12-30 (12:00 PM)	2	RM 80.00	RM 160.00	○
2	Ground Floor	Single Room	GF-S-002	2023-12-25 (02:00 PM)	2023-12-30 (12:00 PM)	2	RM 80.00	RM 160.00	○
3	Ground Floor	Single Room	GF-S-003	2023-12-25 (02:00 PM)	2023-12-30 (12:00 PM)	2	RM 80.00	RM 160.00	○
4	Ground Floor	Single Room	GF-S-004	2023-12-25 (02:00 PM)	2023-12-30 (12:00 PM)	2	RM 80.00	RM 160.00	○
5	Ground Floor	Single Room	GF-S-005	2023-12-25 (02:00 PM)	2023-12-30 (12:00 PM)	2	RM 80.00	RM 160.00	○
5	2nd Floor	Single Room	GF-S-007	2023-12-25 (02:00 PM)	2023-12-30 (12:00 PM)	2	RM 80.00	RM 160.00	○

Table 4.8 Booking Hall

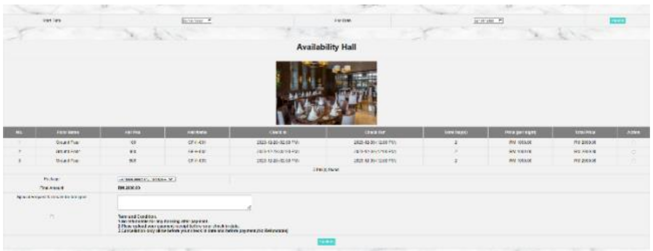
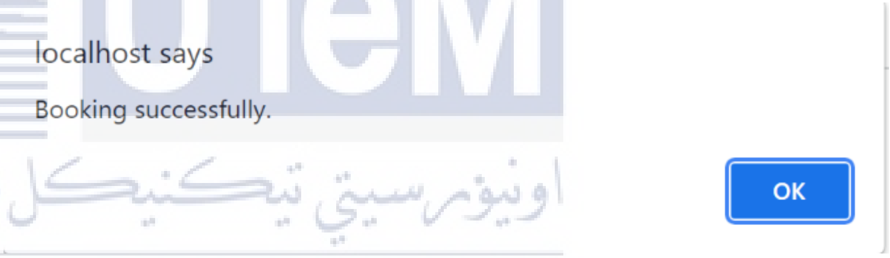
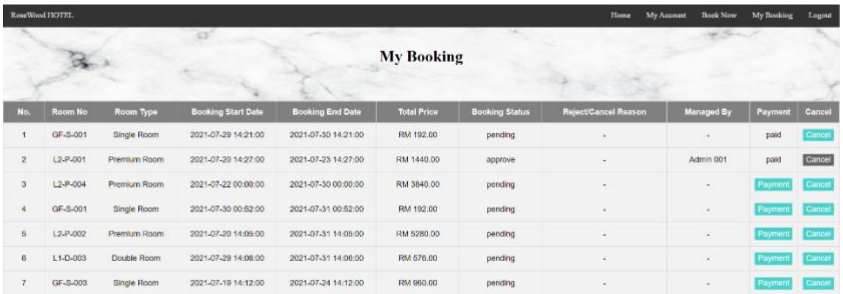
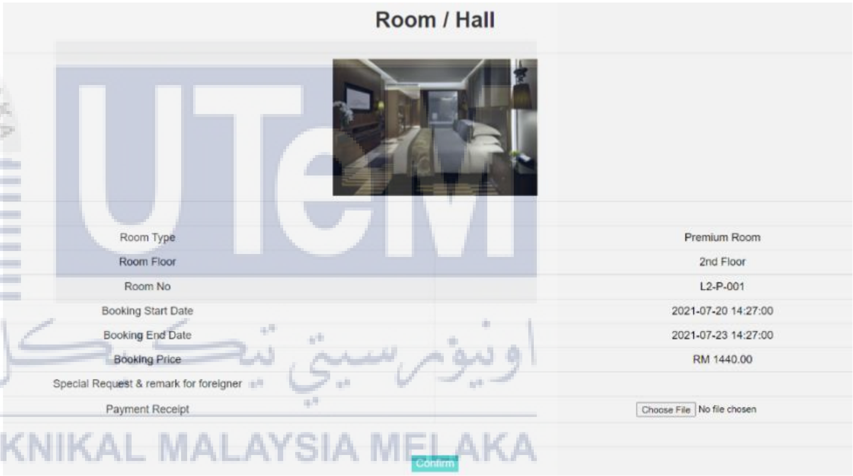
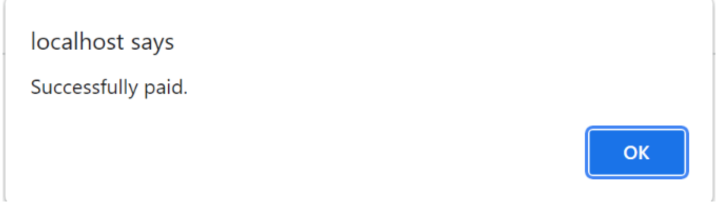
No of Module:	9
Module Name:	Booking Hall
Purpose:	To let user make a hall reservation.
Input:	[STARTDATE] [ENDDATE]
Input Screen:	 Availability Hall
Output:	Output : Make a hall reservation.  localhost says Booking successfully.

Table 4.9 View Booking History

No of Module:	10
Module Name:	View Booking History
Purpose:	To view the booking the made by customers and make payment.
Input:	[PAYMENT]
Input Screen:	 
Output:	Output : Upload payment receipt. 

My Booking

No.	Room No	Room Type	Booking Start Date	Booking End Date	Total Price	Booking Status	Reject/Cancel Reason	Managed By	Payment	Cancel
1	GF-S-001	Single Room	2021-07-29 14:21:00	2021-07-30 14:21:00	RM 192.00	pending	-	-	paid	Cancel
2	L2-P-001	Premium Room	2021-07-20 14:27:00	2021-07-23 14:27:00	RM 1440.00	approve	-	Admin 001	paid	Cancel
3	L2-P-004	Premium Room	2021-07-22 00:00:00	2021-07-30 00:00:00	RM 3840.00	pending	-	-	Payment	Cancel
4	GF-S-001	Single Room	2021-07-30 00:52:00	2021-07-31 00:52:00	RM 192.00	pending	-	-	Payment	Cancel
5	L2-P-002	Premium Room	2021-07-20 14:05:00	2021-07-31 14:05:00	RM 5280.00	pending	-	-	Payment	Cancel
6	L1-D-003	Double Room	2021-07-29 14:06:00	2021-07-31 14:06:00	RM 576.00	pending	-	-	Payment	Cancel
7	GF-S-003	Single Room	2021-07-19 14:12:00	2021-07-24 14:12:00	RM 960.00	pending	-	-	Payment	Cancel

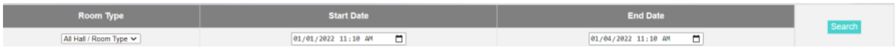
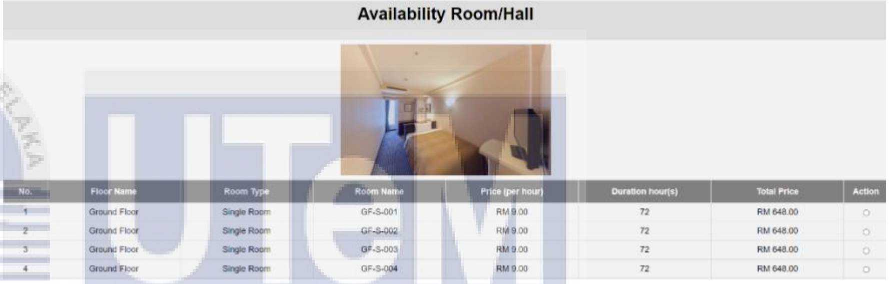
Room / Hall

Room Type	Premium Room
Room Floor	2nd Floor
Room No	L2-P-001
Booking Start Date	2021-07-20 14:27:00
Booking End Date	2021-07-23 14:27:00
Booking Price	RM 1440.00
Special Request & remark for foreigner	
Payment Receipt	Choose File No file chosen

localhost says

Successfully paid.

Table 4.10 View Availability

No of Module:	11
Module Name:	View Availability
Purpose:	To view the Availability of the room.
Input:	[ROOMTYPE] [STARTDATE] [ENDDATE]
Input Screen:	
Output:	Output : View Availability 

Input Screen:

Room Type	Start Date	End Date	
All Hall / Room Type	01/01/2022 11:10 AM	01/04/2022 11:10 AM	Search

Output : View Availability

Availability Room/Hall

No.	Floor Name	Room Type	Room Name	Price (per hour)	Duration hour(s)	Total Price	Action
1	Ground Floor	Single Room	GF-S-001	RM 9.00	72	RM 648.00	○
2	Ground Floor	Single Room	GF-S-002	RM 9.00	72	RM 648.00	○
3	Ground Floor	Single Room	GF-S-003	RM 9.00	72	RM 648.00	○
4	Ground Floor	Single Room	GF-S-004	RM 9.00	72	RM 648.00	○

Table 4.11 Add Attendance

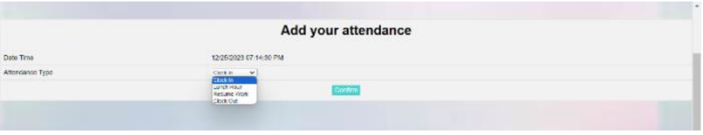
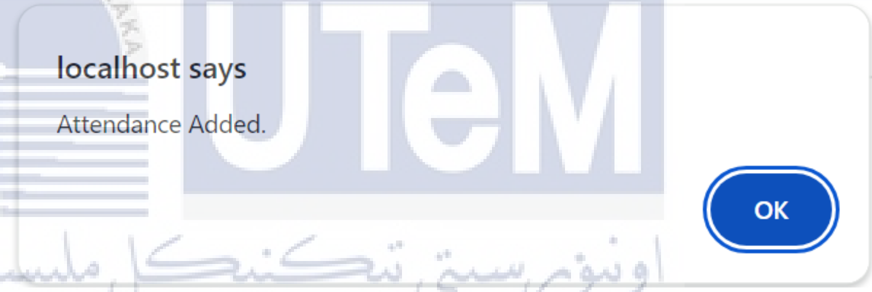
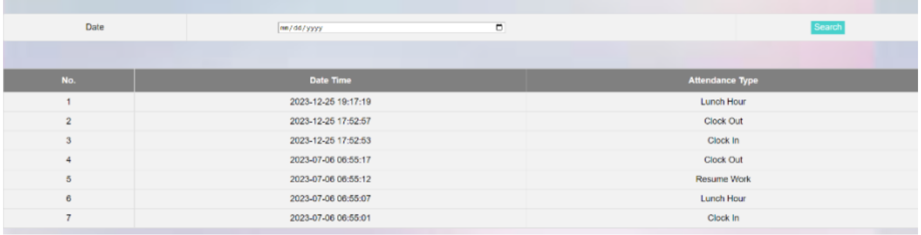
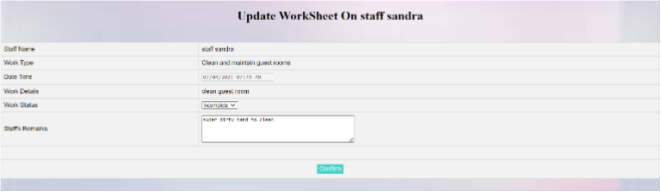
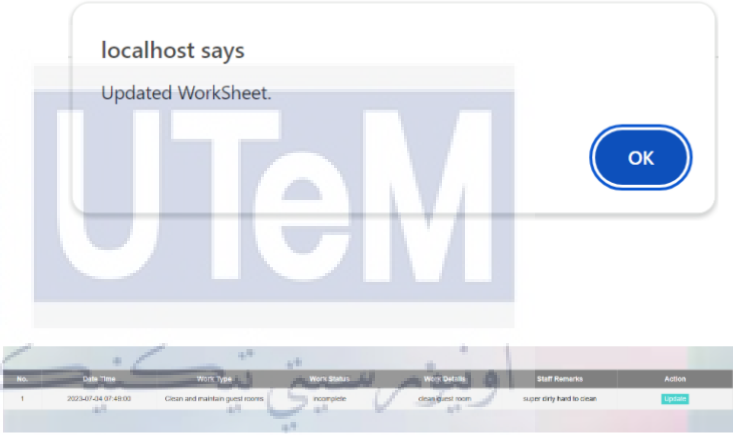
No of Module:	12
Module Name:	Add Work Attendance
Purpose:	To add daily Attendance for staff
Input:	[ATTENDANCETYPE]
Input Screen:	 Add your attendance Date Time 12/25/2023 07:14:30 PM Attendance Type Clock In Clock In Lunch Hour Resume Work Clock Out Confirm
Output:	 localhost says Attendance Added.  Date mm/dd/yyyy No. | Date Time | Attendance Type 1 | 2023-12-25 19:17:19 | Lunch Hour 2 | 2023-12-25 17:52:57 | Clock Out 3 | 2023-12-25 17:52:53 | Clock In 4 | 2023-07-06 06:55:17 | Clock Out 5 | 2023-07-06 06:55:12 | Resume Work 6 | 2023-07-06 06:55:07 | Lunch Hour 7 | 2023-07-06 06:55:01 | Clock In

Table 4.12 Manage Work Sheet for Staff

<table>
<tr><td><b>No of Module:</b></td><td>13</td></tr>
<tr><td><b>Module Name:</b></td><td>View Availability</td></tr>
<tr><td><b>Purpose:</b></td><td>To view the Availability of the room.</td></tr>
<tr><td><b>Input:</b></td><td>[ROOMTYPE] [STARTDATE] [ENDDATE]</td></tr>
<tr><td><b>Input Screen:</b></td><td>

<b>Update WorkSheet On staff sandra</b>
<table>
<tr><td>Staff Name</td><td>staff sandra</td></tr>
<tr><td>Work Type</td><td>Clean and maintain guest rooms</td></tr>
<tr><td>Date Time</td><td>07/04/2023 07:49 AM</td></tr>
<tr><td>Work Details</td><td>clean guest room</td></tr>
<tr><td>Work Status</td><td>incomplete</td></tr>
<tr><td>Staffs Remarks</td><td>super dirty hard to clean</td></tr>
</table>
Confirm
</td></tr>
<tr><td><b>Output:</b></td><td>Output : View Availability

localhost says
Updated WorkSheet.
OK
<table>
<tr><th>No.</th><th>Date Time</th><th>Work Type</th><th>Work Status</th><th>Work Details</th><th>Staff Remarks</th><th>Action</th></tr>
<tr><td>1</td><td>2023-07-04 07:49:00</td><td>Clean and maintain guest rooms</td><td>incomplete</td><td>clean guest room</td><td>super dirty hard to clean</td><td>Update</td></tr>
</table>
</td></tr>
</table>

4.4 Conclusion

In the design phase, all design requirements are accurately depicted to provide developers with clear guidelines. These requirements serve as a reference throughout the project development process, ensuring consistency and alignment with the main objective of the Rosewood Resort Reservation System. The design phase acts as a bridge to the next phase, the implementation phase, where the actual development takes place.

By thoroughly documenting the design requirements, the development team can effectively translate them into tangible system components. This includes designing user interfaces, database structures, and system modules. The design phase facilitates communication and collaboration between stakeholders, ensuring that everyone involved has a clear understanding of the system's design.

Furthermore, the design phase served as a foundation for subsequent development activities. It helps streamline the implementation process by providing a roadmap for developers to follow. This promotes efficiency, reduces errors, and facilitates a smooth transition from design to implementation. Ultimately, the design phase plays a vital role in the overall success of the system development project.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter is dedicated to the implementation of the system, with a particular focus on establishing the software development environment and configuring the database. The section concerning the setup of the software development environment encompasses tasks like database creation and the definition of its constituent elements. In tandem, the database implementation phase will delve into both Data Definition Language (DDL) operations and the subsequent data loading procedures. A comprehensive exploration of database design and execution is presented within this chapter. The principal objective here is to deliver a comprehensive understanding of the intricate processes involved in actualizing the database itself.

5.2 Software Development Environment Setup

5.2.1 Database Environment Setup

The creation of the local database for this project is facilitated through the utilization of phpMyAdmin MySQL.

5.2.1.1 Installation Step for Local Database

Setting up phpMyAdmin MySQL involves installing XAMPP, a software application that enables programmers to write and test code utilizing MySQL databases on a local web server within their host system. XAMPP can be installed across Windows, Linux, and Mac platforms. Given the Windows laptop environmentfor this project, XAMPP for Windows has been selected as the appropriate choice. Refer to Figure 5.0 for the download location of XAMPP.

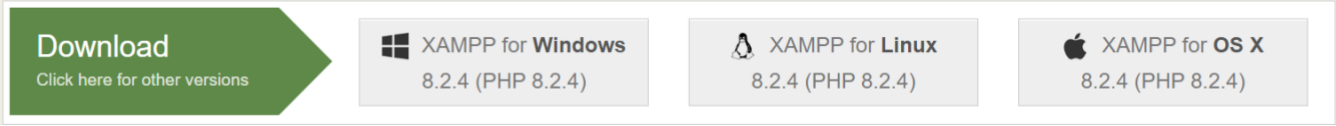
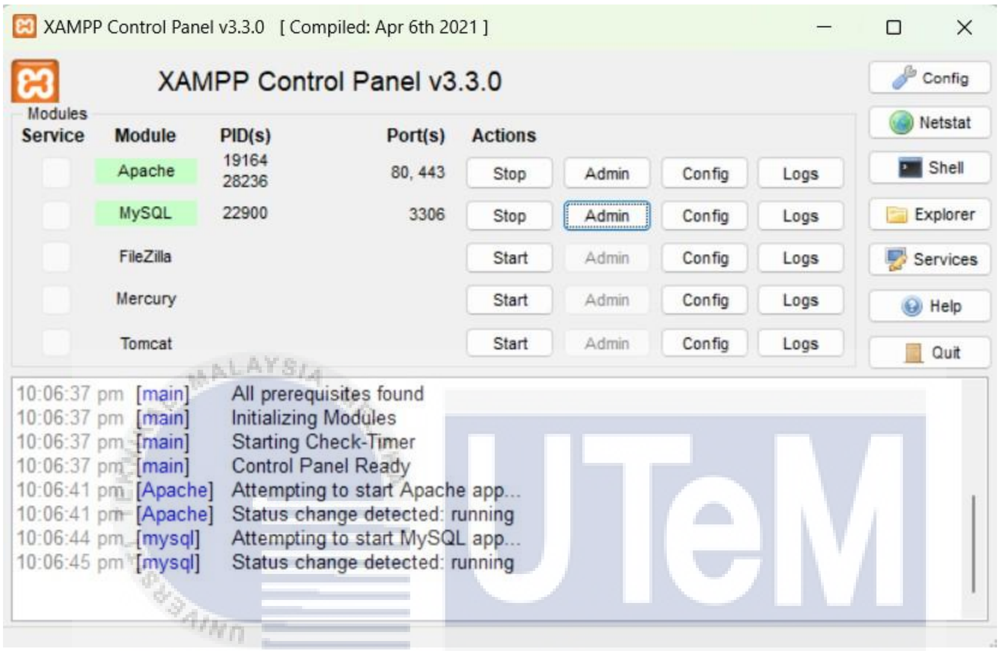


Figure 5.0 XAMPP for Windows

To establish a connection with the MySQL database, two essential buttons, namely Apache and MySQL, need to be initiated. Successful activation or currently running instances will be visually highlighted in green. To access the database administration or DBA, the "admin" button needs to be clicked, unveiling the MySQL administration interface. Refer to Figure 5.1 for an illustrative depiction of the administrative access procedure.



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

XAMPP Control Panel v3.3.0

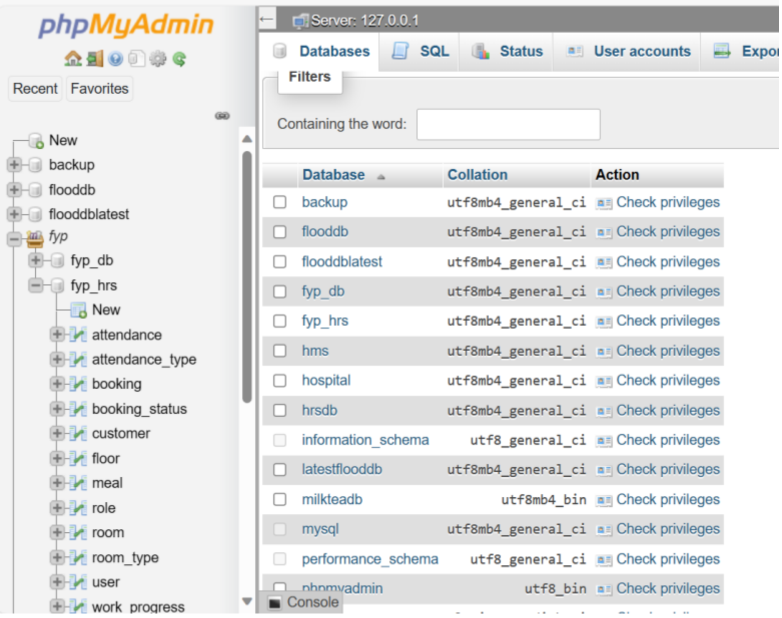
Modules

Service	Module	PID(s)	Port(s)	Actions			
	Apache	19164 28236	80, 443	Stop	Admin	Config	Logs
	MySQL	22900	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
10:06:37 pm  [main]    All prerequisites found
10:06:37 pm  [main]    Initializing Modules
10:06:37 pm  [main]    Starting Check-Timer
10:06:37 pm  [main]    Control Panel Ready
10:06:41 pm  [Apache]  Attempting to start Apache app...
10:06:41 pm  [Apache]  Status change detected: running
10:06:44 pm  [mysql]   Attempting to start MySQL app...
10:06:45 pm  [mysql]   Status change detected: running
```

Figure 5.1 XAMPP Control Panel

As demonstrated in Figure 5.2, successful connectivity to the phpMyAdmin MySQL Administrator has been established.



	Database	Collation	Action
□	backup	utf8mb4_general_ci	Check privileges
□	flooddb	utf8mb4_general_ci	Check privileges
□	flooddblatest	utf8mb4_general_ci	Check privileges
□	fyp_db	utf8mb4_general_ci	Check privileges
□	fyp_hrs	utf8mb4_general_ci	Check privileges
□	hms	utf8mb4_general_ci	Check privileges
□	hospital	utf8mb4_general_ci	Check privileges
□	hrsdb	utf8mb4_general_ci	Check privileges
□	information_schema	utf8_general_ci	Check privileges
□	latestflooddb	utf8mb4_general_ci	Check privileges
□	milkteadb	utf8mb4_bin	Check privileges
□	mysql	utf8mb4_general_ci	Check privileges
□	performance_schema	utf8_general_ci	Check privileges
□	phpmyadmin	utf8_bin	Check privileges

Figure 5.2 phpMyAdmin server

The configuration of the system's local database is outlined below:

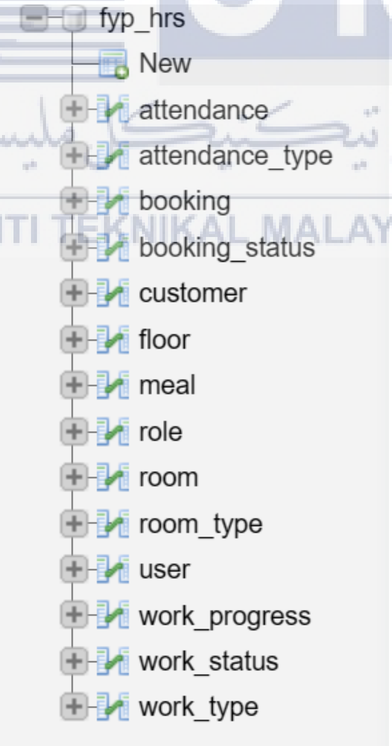
```
<?php
$servername="localhost";
$username="root";
$password="";
$dbname="fyp_hrs";

//Create connection
$conn=new mysqli($servername,$username,$password,$dbname);

//Check connection
if($conn->connect_error){
        die("Connection failed: ".$conn->connect_error);
}
//echo "Connected successfully";

?>
```

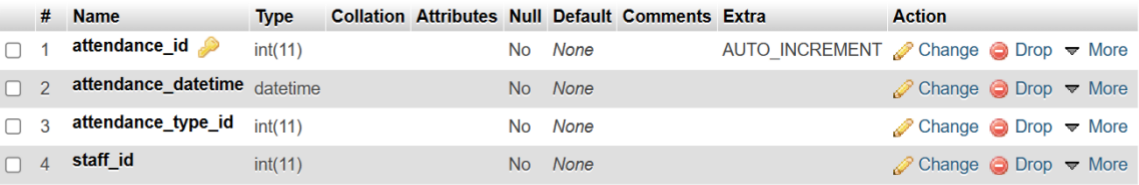
5.2.2 Database Creation and Database Object Creation



```
fyp_hrs
  New
  attendance
  attendance_type
  booking
  booking_status
  customer
  floor
  meal
  role
  room
  room_type
  user
  work_progress
  work_status
  work_type
```

Figure 5.3 showcases the successful creation of 14 tables within the MySQL database

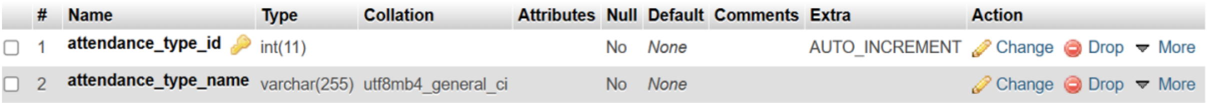
Attendance



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	attendance_id 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop ▾ More
□	2	attendance_datetime	datetime			No	*None*			Change Drop ▾ More
□	3	attendance_type_id	int(11)			No	*None*			Change Drop ▾ More
□	4	staff_id	int(11)			No	*None*			Change Drop ▾ More

Figure 5.4 Attendance Table.

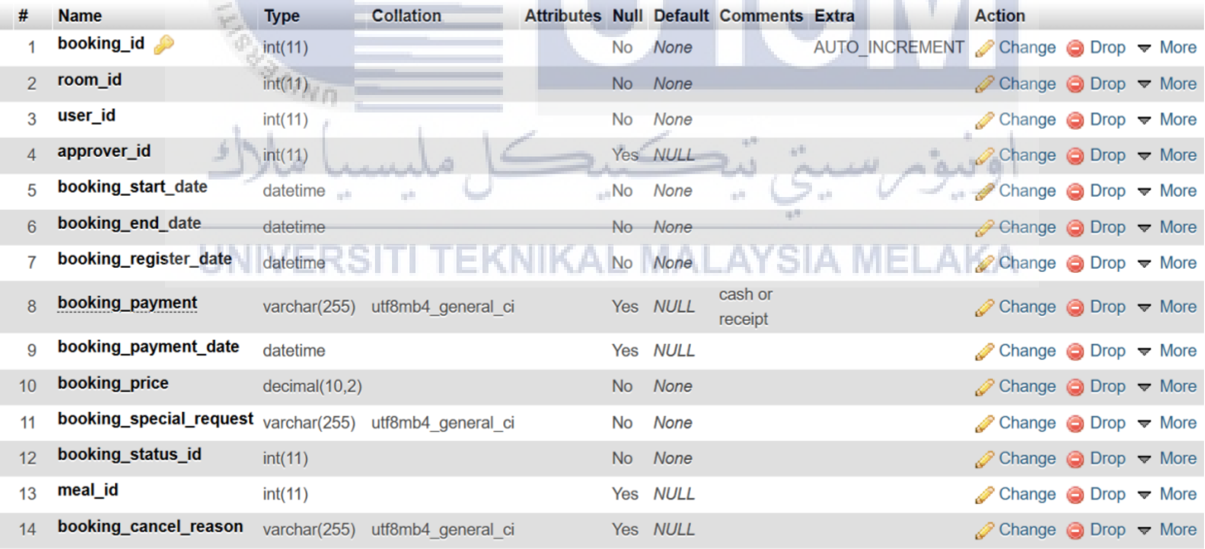
Attendance_type



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	attendance_type_id 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop ▾ More
□	2	attendance_type_name	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop ▾ More

Figure 5.5 Attendance_Type Table.

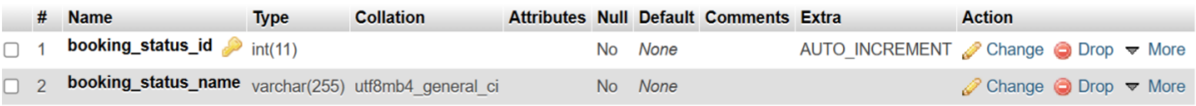
Booking



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	booking_id 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop ▾ More
2	room_id	int(11)			No	*None*			Change Drop ▾ More
3	user_id	int(11)			No	*None*			Change Drop ▾ More
4	approver_id	int(11)			Yes	*NULL*			Change Drop ▾ More
5	booking_start_date	datetime			No	*None*			Change Drop ▾ More
6	booking_end_date	datetime			No	*None*			Change Drop ▾ More
7	booking_register_date	datetime			No	*None*			Change Drop ▾ More
8	booking_payment	varchar(255)	utf8mb4_general_ci		Yes	*NULL*	cash or receipt		Change Drop ▾ More
9	booking_payment_date	datetime			Yes	*NULL*			Change Drop ▾ More
10	booking_price	decimal(10,2)			No	*None*			Change Drop ▾ More
11	booking_special_request	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop ▾ More
12	booking_status_id	int(11)			No	*None*			Change Drop ▾ More
13	meal_id	int(11)			Yes	*NULL*			Change Drop ▾ More
14	booking_cancel_reason	varchar(255)	utf8mb4_general_ci		Yes	*NULL*			Change Drop ▾ More

Figure 5.6 Booking Table.

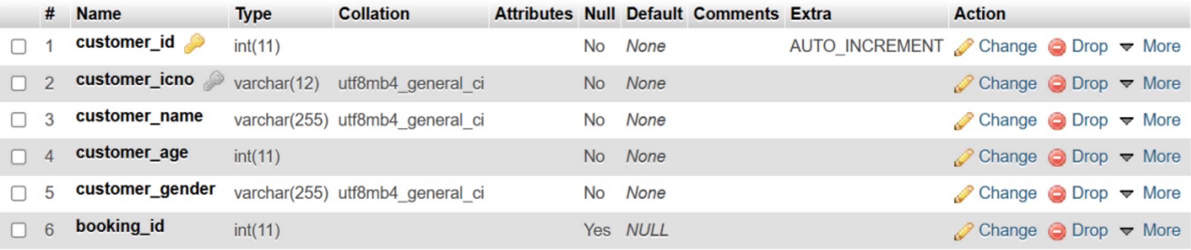
Booking_status



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	booking_status_id 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop ▾ More
□	2	booking_status_name	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop ▾ More

Figure 5.7 Booking_Status Table.

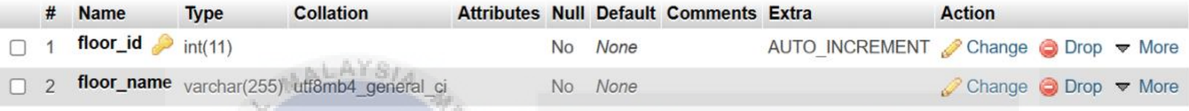
Customer



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**customer_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**customer_icno**	varchar(12)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	**customer_name**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	**customer_age**	int(11)			No	*None*			Change Drop More
☐	5	**customer_gender**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	6	**booking_id**	int(11)			Yes	*NULL*			Change Drop More

Figure 5.8 Customer Table.

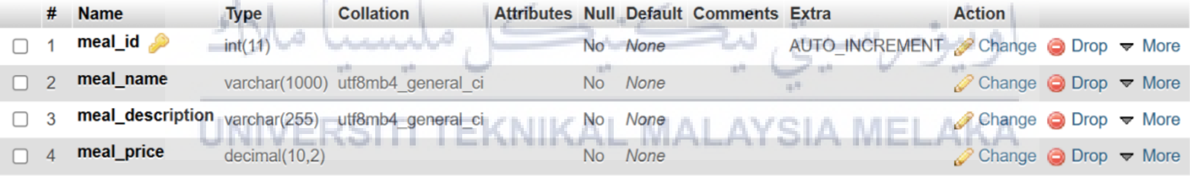
Floor



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**floor_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**floor_name**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

Figure 5.9 Floor Table.

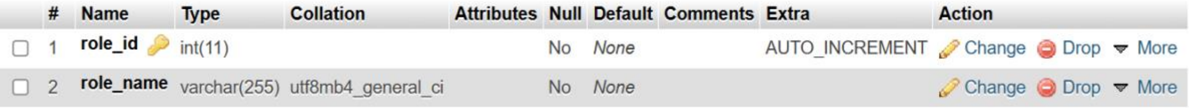
Meal



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**meal_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**meal_name**	varchar(1000)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	**meal_description**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	**meal_price**	decimal(10,2)			No	*None*			Change Drop More

Figure 5.10 Meal Table.

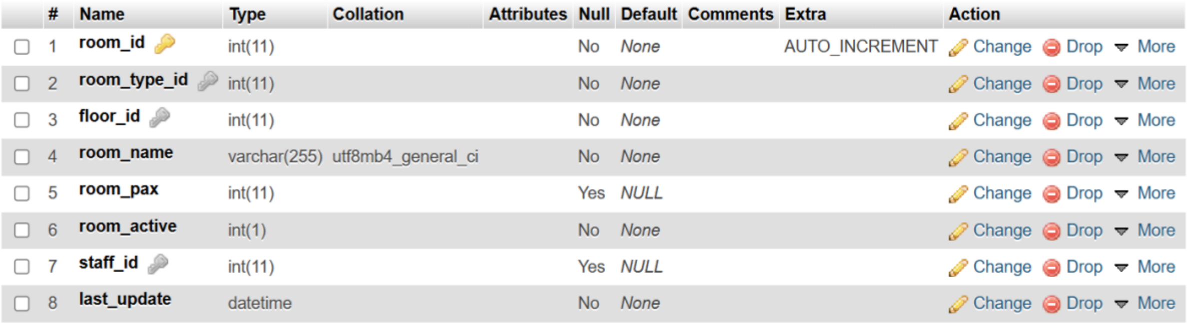
Role



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**role_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**role_name**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

Figure 5.11 Role Table.

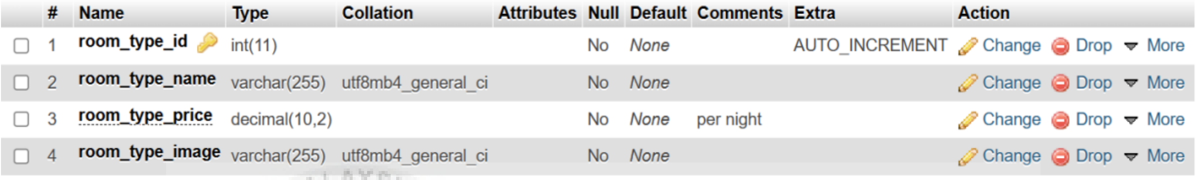
Room



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	room_id 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	room_type_id 🔑	int(11)			No	*None*			Change Drop More
□	3	floor_id 🔑	int(11)			No	*None*			Change Drop More
□	4	room_name	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
□	5	room_pax	int(11)			Yes	*NULL*			Change Drop More
□	6	room_active	int(1)			No	*None*			Change Drop More
□	7	staff_id 🔑	int(11)			Yes	*NULL*			Change Drop More
□	8	last_update	datetime			No	*None*			Change Drop More

Figure 5.12 Room Table.

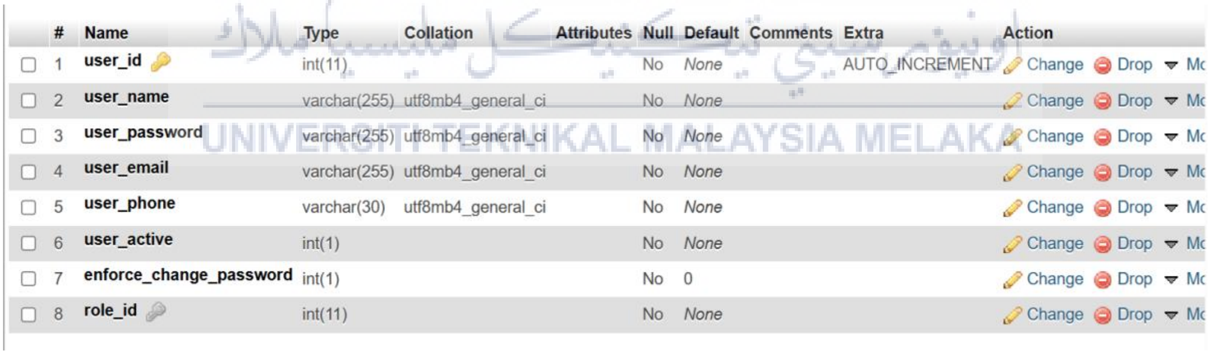
Room_type



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	room_type_id 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	room_type_name	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	room_type_price	decimal(10,2)			No	*None*	per night		Change Drop More
□	4	room_type_image	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

Figure 5.13 Room_Type Table.

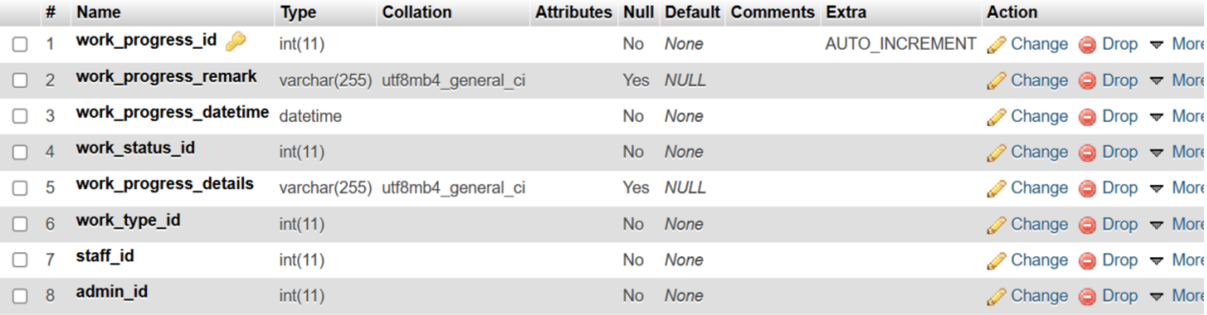
User



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	user_id 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop Mo
□	2	user_name	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop Mo
□	3	user_password	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop Mo
□	4	user_email	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop Mo
□	5	user_phone	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop Mo
□	6	user_active	int(1)			No	*None*			Change Drop Mo
□	7	enforce_change_password	int(1)			No	0			Change Drop Mo
□	8	role_id 🔑	int(11)			No	*None*			Change Drop Mo

Figure 5.14 User Table.

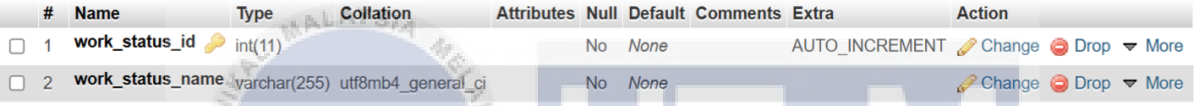
Work_Progress



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	work_progress_id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop Mor
□	2	work_progress_remark	varchar(255)	utf8mb4_general_ci		Yes	*NULL*			Change Drop Mor
□	3	work_progress_datetime	datetime			No	*None*			Change Drop Mor
□	4	work_status_id	int(11)			No	*None*			Change Drop Mor
□	5	work_progress_details	varchar(255)	utf8mb4_general_ci		Yes	*NULL*			Change Drop Mor
□	6	work_type_id	int(11)			No	*None*			Change Drop Mor
□	7	staff_id	int(11)			No	*None*			Change Drop Mor
□	8	admin_id	int(11)			No	*None*			Change Drop Mor

Figure 5.15 Work_Progress Table.

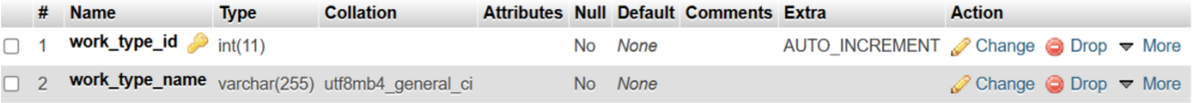
Work_Status



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	work_status_id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	work_status_name	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

Figure 5.16 Work_Status Table.

Work_Type



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	work_type_id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	work_type_name	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

Figure 5.17 Work_Type Table.

5.3 Database Implementation

5.3.1 Data Definition Language (DDL) Statement

Data Definition Language (DDL) is a subset of Structured Query Language (SQL) that focuses on defining and managing the structure and attributes of a database. DDL commands are used to create, modify, and delete database objects such as tables, indexes, views, and schemas. Unlike Data Manipulation Language (DML) which deals with the manipulation of data within these objects, DDL is concerned with the overall organization and design of the database. DDL statements enable users to establish the data types, constraints, relationships, and other properties that govern how data is stored, ensuring data integrity and consistency. Common DDL commands include "CREATE" for generating new database objects, "ALTER" for modifying existing structures, and "DROP" for removing objects. DDL plays a crucial role in database management systems, providing a foundation for creating the logical framework in which data is stored and accessed.

DDL for Rosewood Resort Reservation System:

A. Create Table

Attendance Table

```
CREATE TABLE `attendance` (
  `attendance_id` int(11) NOT NULL,
  `attendance_datetime` datetime NOT NULL,
  `attendance_type_id` int(11) NOT NULL,
  `staff_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Attendance_Type Table

```
CREATE TABLE `attendance_type` (
  `attendance_type_id` int(11) NOT NULL,
  `attendance_type_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Booking Table

```
CREATE TABLE `booking` (
  `booking_id` int(11) NOT NULL,
  `room_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `approver_id` int(11) DEFAULT NULL,
  `booking_start_date` datetime NOT NULL,
  `booking_end_date` datetime NOT NULL,
  `booking_register_date` datetime NOT NULL,
  `booking_payment` varchar(255) DEFAULT NULL COMMENT 'cash or receipt',
  `booking_payment_date` datetime DEFAULT NULL,
  `booking_price` decimal(10,2) NOT NULL,
  `booking_special_request` varchar(255) NOT NULL,
  `booking_status_id` int(11) NOT NULL,
  `meal_id` int(11) DEFAULT NULL,
  `booking_cancel_reason` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Booking_Status Table

```
CREATE TABLE `booking_status` (
  `booking_status_id` int(11) NOT NULL,
  `booking_status_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Customer Table

```
CREATE TABLE `customer` (
  `customer_id` int(11) NOT NULL,
  `customer_icno` varchar(12) NOT NULL,
  `customer_name` varchar(255) NOT NULL,
  `customer_age` int(11) NOT NULL,
  `customer_gender` varchar(255) NOT NULL,
  `booking_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Floor Table

```
CREATE TABLE `floor` (
  `floor_id` int(11) NOT NULL,
  `floor_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Meal Table

```
CREATE TABLE `meal` (
  `meal_id` int(11) NOT NULL,
  `meal_name` varchar(1000) NOT NULL,
  `meal_description` varchar(255) NOT NULL,
  `meal_price` decimal(10,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Role Table

```
CREATE TABLE `role` (
  `role_id` int(11) NOT NULL,
  `role_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Room Table

```
CREATE TABLE `room` (
  `room_id` int(11) NOT NULL,
  `room_type_id` int(11) NOT NULL,
  `floor_id` int(11) NOT NULL,
  `room_name` varchar(255) NOT NULL,
  `room_pax` int(11) DEFAULT NULL,
  `room_active` int(1) NOT NULL,
  `staff_id` int(11) DEFAULT NULL,
  `last_update` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Room_Type Table

```
CREATE TABLE `room_type` (
  `room_type_id` int(11) NOT NULL,
  `room_type_name` varchar(255) NOT NULL,
  `room_type_price` decimal(10,2) NOT NULL COMMENT 'per night',
  `room_type_image` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

User Table

```
CREATE TABLE `user` (
  `user_id` int(11) NOT NULL,
  `user_name` varchar(255) NOT NULL,
  `user_password` varchar(255) NOT NULL,
  `user_email` varchar(255) NOT NULL,
  `user_phone` varchar(30) NOT NULL,
  `user_active` int(1) NOT NULL,
  `enforce_change_password` int(1) NOT NULL DEFAULT 0,
  `role_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Work_Progress Table

```
CREATE TABLE `work_progress` (
  `work_progress_id` int(11) NOT NULL,
  `work_progress_remark` varchar(255) DEFAULT NULL,
  `work_progress_datetime` datetime NOT NULL,
  `work_status_id` int(11) NOT NULL,
  `work_progress_details` varchar(255) DEFAULT NULL,
  `work_type_id` int(11) NOT NULL,
  `staff_id` int(11) NOT NULL,
  `admin_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Work_Status Table

```
CREATE TABLE `work_status` (
  `work_status_id` int(11) NOT NULL,
  `work_status_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Work_Type Table

```
CREATE TABLE `work_type` (
  `work_type_id` int(11) NOT NULL,
  `work_type_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

B. Index For Table

Attendance Table

```
ALTER TABLE `attendance`
  ADD PRIMARY KEY (`attendance_id`);
```

Attendance_Type Table

```
ALTER TABLE `attendance_type`
  ADD PRIMARY KEY (`attendance_type_id`);
```

Booking Table

```
ALTER TABLE `booking`
  ADD PRIMARY KEY (`booking_id`);
```

Booking_Status Table

```
ALTER TABLE `booking_status`
  ADD PRIMARY KEY (`booking_status_id`);
```

Customer Table

```
ALTER TABLE `customer`
  ADD PRIMARY KEY (`customer_id`),
  ADD UNIQUE KEY `unique_customer_icno` (`customer_icno`);
```

Floor Table

```
ALTER TABLE `floor`
  ADD PRIMARY KEY (`floor_id`);
```

Meal Table

```
ALTER TABLE `meal`
  ADD PRIMARY KEY (`meal_id`);;
```

Role Table

```
ALTER TABLE `role`
  ADD PRIMARY KEY (`role_id`);
```

Room Table

```
ALTER TABLE `room`
  ADD PRIMARY KEY (`room_id`),
  ADD KEY `fk_room_floor` (`floor_id`),
  ADD KEY `fk_room_room_type` (`room_type_id`),
  ADD KEY `fk_room_staff` (`staff_id`);
```

Room_Type Table

```
ALTER TABLE `room_type`
  ADD PRIMARY KEY (`room_type_id`);
```

User Table

```
ALTER TABLE `user`
  ADD PRIMARY KEY (`user_id`),
  ADD KEY `fk_user_role` (`role_id`);
```

Work_Progress Table

```
ALTER TABLE `work_progress`
  ADD PRIMARY KEY (`work_progress_id`);
```

Work_Status Table

```
ALTER TABLE `work_status`
  ADD PRIMARY KEY (`work_status_id`);
```

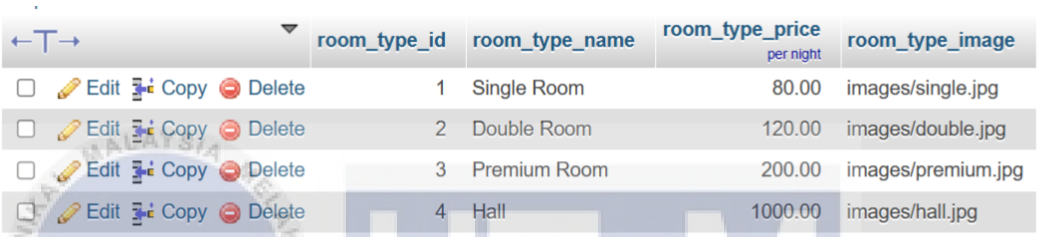
Work_Type Table

```
ALTER TABLE `work_type`
  ADD PRIMARY KEY (`work_type_id`);
```

5.3.2 Data Manipulation Language (DML) Statement

Data Manipulation Language (DML) is a subset of Structured Query Language (SQL) used for interacting with and modifying data within a relational database. DML is focused on performing operations that manipulate the content of database tables, such as inserting, updating, retrieving, and deleting records. These operations enable users to interact with the data stored in the database, shaping it according to various requirements. The primary DML commands include "SELECT" for querying data, "INSERT" for adding new records, "UPDATE" for modifying existing data, and "DELETE" for removing records. DML statements offer powerful control over data transformation and retrieval, allowing users to apply filters, join tables, aggregate information, and more. Unlike Data Definition Language (DDL), which concentrates on the structure of the database, DML is concerned with the actual data stored within it, making it a foundational tool for database applications and data-driven operations.

5.3.2.1 Select Command

Query	SELECT * FROM room_type
Description	To check the room type details
Output	

←T→	room_type_id	room_type_name	room_type_price per night	room_type_image
□ Edit Copy Delete	1	Single Room	80.00	images/single.jpg
□ Edit Copy Delete	2	Double Room	120.00	images/double.jpg
□ Edit Copy Delete	3	Premium Room	200.00	images/premium.jpg
□ Edit Copy Delete	4	Hall	1000.00	images/hall.jpg

Table 5.0 Select Room Type Query.

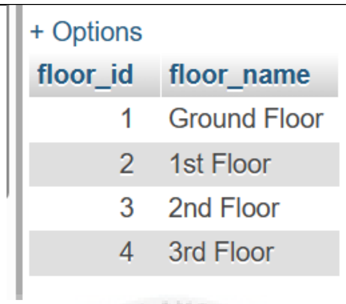
<table>
<tr><td><b>Query</b></td><td>SELECT * FROM floor</td></tr>
<tr><td><b>Description</b></td><td>To check the floor type details</td></tr>
<tr><td><b>Output</b></td><td>
+ Options
<table>
<tr><th>floor_id</th><th>floor_name</th></tr>
<tr><td>1</td><td>Ground Floor</td></tr>
<tr><td>2</td><td>1st Floor</td></tr>
<tr><td>3</td><td>2nd Floor</td></tr>
<tr><td>4</td><td>3rd Floor</td></tr>
</table></td></tr>
</table>

Table 5.1 Select Floor Type Query.

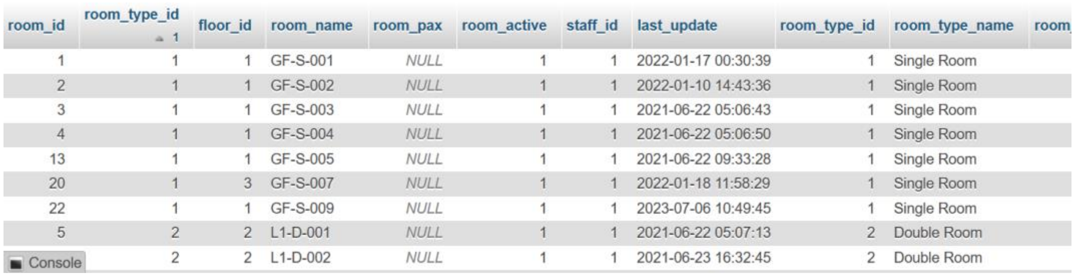
<table>
<tr><td><b>Query</b></td><td>SELECT * FROM room r
LEFT JOIN room_type rt ON r.room_type_id = rt.room_type_id
LEFT JOIN floor f ON r.floor_id = f.floor_id
LEFT JOIN user u ON r.staff_id = u.user_id
ORDER BY r.room_type_id</td></tr>
<tr><td><b>Description</b></td><td>Select the details of room and join with room_type, floor and user table.</td></tr>
<tr><td><b>Output</b></td><td>
<table>
<tr><th>room_id</th><th>room_type_id ▲ 1</th><th>floor_id</th><th>room_name</th><th>room_pax</th><th>room_active</th><th>staff_id</th><th>last_update</th><th>room_type_id</th><th>room_type_name</th><th>room</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>GF-S-001</td><td>NULL</td><td>1</td><td>1</td><td>2022-01-17 00:30:39</td><td>1</td><td>Single Room</td><td></td></tr>
<tr><td>2</td><td>1</td><td>1</td><td>GF-S-002</td><td>NULL</td><td>1</td><td>1</td><td>2022-01-10 14:43:36</td><td>1</td><td>Single Room</td><td></td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>GF-S-003</td><td>NULL</td><td>1</td><td>1</td><td>2021-06-22 05:06:43</td><td>1</td><td>Single Room</td><td></td></tr>
<tr><td>4</td><td>1</td><td>1</td><td>GF-S-004</td><td>NULL</td><td>1</td><td>1</td><td>2021-06-22 05:06:50</td><td>1</td><td>Single Room</td><td></td></tr>
<tr><td>13</td><td>1</td><td>1</td><td>GF-S-005</td><td>NULL</td><td>1</td><td>1</td><td>2021-06-22 09:33:28</td><td>1</td><td>Single Room</td><td></td></tr>
<tr><td>20</td><td>1</td><td>3</td><td>GF-S-007</td><td>NULL</td><td>1</td><td>1</td><td>2022-01-18 11:58:29</td><td>1</td><td>Single Room</td><td></td></tr>
<tr><td>22</td><td>1</td><td>1</td><td>GF-S-009</td><td>NULL</td><td>1</td><td>1</td><td>2023-07-06 10:49:45</td><td>1</td><td>Single Room</td><td></td></tr>
<tr><td>5</td><td>2</td><td>2</td><td>L1-D-001</td><td>NULL</td><td>1</td><td>1</td><td>2021-06-22 05:07:13</td><td>2</td><td>Double Room</td><td></td></tr>
<tr><td>Console</td><td>2</td><td>2</td><td>L1-D-002</td><td>NULL</td><td>1</td><td>1</td><td>2021-06-23 16:32:45</td><td>2</td><td>Double Room</td><td></td></tr>
</table></td></tr>
</table>

Table 5.2 Select Room Query.

5.3.2.2 Insert Command

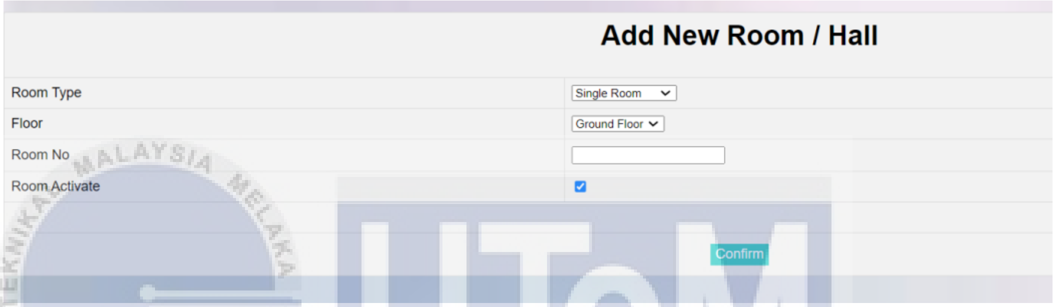
Query	"INSERT INTO `room` (room_type_id, floor_id, room_name, room_pax, room_active, staff_id, last_update) VALUES ('$room_type', '$floor', '$room_no', " .$room_pax. ", '$room_activate', '$user_id', NOW())";
Description	Admin insert new room.
Output	 Add New Room / Hall Room Type: Single Room Floor: Ground Floor Room No: Room Activate: ☑ Confirm

Table 5.3 Insert Room Query.

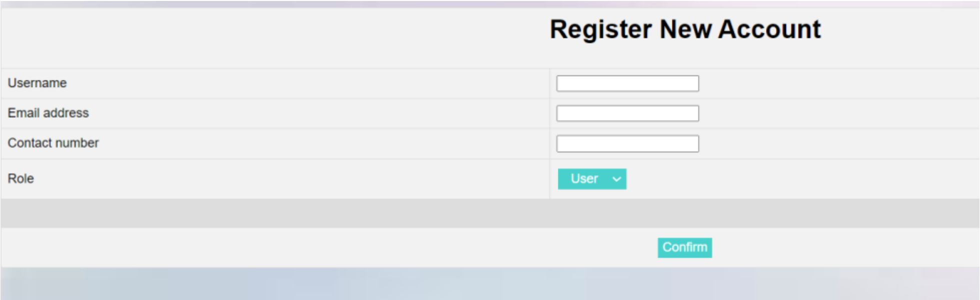
Query	"INSERT INTO user (user_name, user_password, user_email, user_phone, user_active, enforce_change_password, role_id) VALUES ('$username','$password','$email','$phone','1', '1', $role_id)";
Description	Admin insert new user.
Output	 Register New Account Username: Email address: Contact number: Role: User Confirm

Table 5.4 Insert User Query.

5.3.2.3 Update Command

Query	UPDATE \`work_progress\` SET admin_id = '$admin_id',
	work_type_id = '$work_type_id',
	work_progress_datetime = '$work_progress_datetime',
	work_progress_details = '$work_progress_details'
	WHERE work_progress_id = '$work_id';

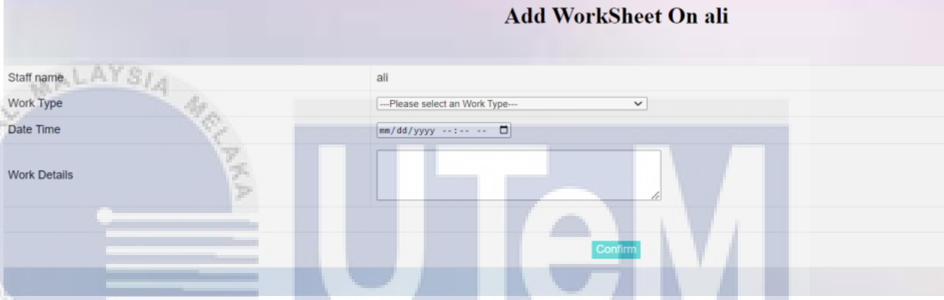
Description	Update the work progress details with the datetime and details.
Output	 Add WorkSheet On ali Staff name ali Work Type ---Please select an Work Type--- Date Time mm/dd/yyyy --:-- -- Work Details

Table 5.5 Update Work Progress Query.

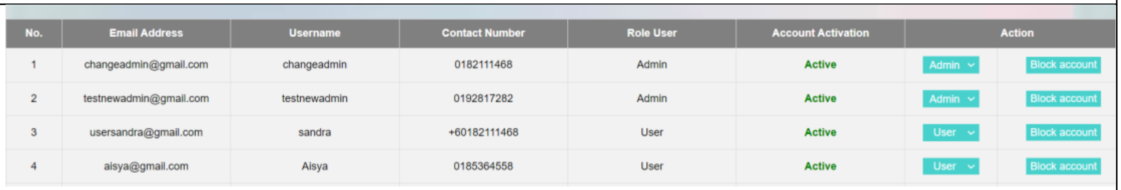
<table>
<tr><td><b>Query</b></td><td>"update user set user_active = '$active' where user_id = '$uid'";</td></tr>
<tr><td><b>Description</b></td><td>Admin Update user account active status.</td></tr>
<tr><td><b>Output</b></td><td>
<table>
<tr><th>No.</th><th>Email Address</th><th>Username</th><th>Contact Number</th><th>Role User</th><th>Account Activation</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td>changeadmin@gmail.com</td><td>changeadmin</td><td>0182111468</td><td>Admin</td><td>Active</td><td>Admin</td><td>Block account</td></tr>
<tr><td>2</td><td>testnewadmin@gmail.com</td><td>testnewadmin</td><td>0192817282</td><td>Admin</td><td>Active</td><td>Admin</td><td>Block account</td></tr>
<tr><td>3</td><td>usersandra@gmail.com</td><td>sandra</td><td>+60182111468</td><td>User</td><td>Active</td><td>User</td><td>Block account</td></tr>
<tr><td>4</td><td>aisya@gmail.com</td><td>Aisya</td><td>0185364558</td><td>User</td><td>Active</td><td>User</td><td>Block account</td></tr>
</table></td></tr>
</table>

Table 5.6 Update User Account Active Status Query.

5.3.2.4 Delete Command

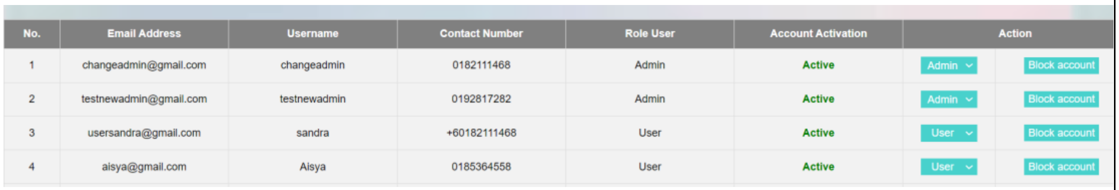
<table>
<tr><td><b>Query</b></td><td>delete from user where user_id = '$u_id'</td></tr>
<tr><td><b>Description</b></td><td>Delete the user.</td></tr>
<tr><td><b>Output</b></td><td>
<table>
<tr><th>No.</th><th>Email Address</th><th>Username</th><th>Contact Number</th><th>Role User</th><th>Account Activation</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td>changeadmin@gmail.com</td><td>changeadmin</td><td>0182111468</td><td>Admin</td><td>Active</td><td>Admin</td><td>Block account</td></tr>
<tr><td>2</td><td>testnewadmin@gmail.com</td><td>testnewadmin</td><td>0192817282</td><td>Admin</td><td>Active</td><td>Admin</td><td>Block account</td></tr>
<tr><td>3</td><td>usersandra@gmail.com</td><td>sandra</td><td>+60182111468</td><td>User</td><td>Active</td><td>User</td><td>Block account</td></tr>
<tr><td>4</td><td>aisya@gmail.com</td><td>Aisya</td><td>0185364558</td><td>User</td><td>Active</td><td>User</td><td>Block account</td></tr>
</table></td></tr>
</table>

Table 5.7 Delete User Account Query.

5.3.3 Trigger

A trigger in SQL is a database object that automatically executes a set of actions in response to a specific event occurring within a database. This event could be an INSERT, UPDATE, DELETE, or other data-related operation performed on a table. Triggers are defined to monitor these events and can be configured to initiate actions like modifying data, enforcing data integrity, logging events, or performing other custom operations. They provide a way to automate certain tasks, maintain data consistency, and enforce business rules without requiring manual intervention. Triggers play a vital role in database management by allowing developers to define behavior that responds to changes in data, enhancing the database's efficiency, integrity, and functionality. Table

Table 5.8 List for Trigger

Desc	Query

Trigger to Update Room Status:	DELIMITER //
	CREATE OR REPLACE TRIGGER AfterBookingInsert
	AFTER INSERT ON booking
	FOR EACH ROW
	BEGIN
	  UPDATE room
	  SET room_active = 0
	  WHERE room.room_id = NEW.room_id;
	END //
	DELIMITER ;

Trigger to Update Room Status on Booking Cancellation:	DELIMITER //
	CREATE OR REPLACE TRIGGER AfterBookingUpdate
	AFTER UPDATE ON booking
	FOR EACH ROW
	BEGIN
	  IF NEW.booking_status_id = 3 THEN -- Assuming status code 3 represents canceled
	    UPDATE room
	    SET room_active = 1

	WHERE room.room_id = NEW.room_id;
	END IF;
	END //
	DELIMITER ;

Trigger to Enforce Password Change:	DELIMITER //
	CREATE OR REPLACE BeforeUserUpdate
	BEFORE UPDATE ON user
	FOR EACH ROW
	BEGIN
	IF NEW.enforce_change_password = 1 AND NEW.user_password = OLD.user_password THEN
	SIGNAL SQLSTATE '45000'
	SET MESSAGE_TEXT = 'Password change required.';
	END IF;
	END //
	DELIMITER ;

Trigger to Update Last Update Time for Rooms:	DELIMITER //
	CREATE OR REPLACE TRIGGER BeforeRoomUpdate
	BEFORE UPDATE ON room
	FOR EACH ROW
	BEGIN
	SET NEW.last_update = NOW();
	END //
	DELIMITER ;

Trigger to Track Work Progress:	DELIMITER // CREATE OR REPLACE TRIGGER BeforeWorkProgressInsert BEFORE INSERT ON work_progress FOR EACH ROW BEGIN IF NEW.admin_id IS NULL THEN SIGNAL SQLSTATE '45000'

	SET MESSAGE_TEXT = 'Admin ID cannot be NULL.';
	END IF;
	END //
	DELIMITER ;

5.3.4 Procedure

A procedure in the context of databases is a reusable and self-contained block of code that encapsulates a series of SQL statements. It is defined within a database to perform a specific task or set of tasks. Procedures help streamline database operations by allowing complex operations to be executed with a single call. They can accept input parameters, process data, and return results. Procedures enhance code modularity, as they can be called from various parts of an application without needing to replicate the same code each time. This promotes code reusability, reduces redundancy, and simplifies maintenance. Procedures are particularly useful for performing routine tasks, complex calculations, or enforcing business logic within a database.

Table 5.9 List of Procedure

Desc	**Query**

Procedure to Insert a Booking Record:	DELIMITER //
	CREATE OR REPLACE PROCEDURE InsertBooking(
	  IN room_id_param INT,
	  IN user_id_param INT,
	  IN booking_start_date_param DATETIME,
	  IN booking_end_date_param DATETIME,
	  IN booking_price_param DECIMAL(10,2),
	  IN booking_special_request_param VARCHAR(255)
	)
	BEGIN
	  INSERT INTO booking (room_id, user_id, booking_start_date, booking_end_date, booking_price, booking_special_request, booking_status_id)
	  VALUES (room_id_param, user_id_param, booking_start_date_param, booking_end_date_param, booking_price_param, booking_special_request_param, 1);
	END //
	DELIMITER ;

Procedure to Update Customer Information:	DELIMITER //
	CREATE OR REPLACE PROCEDURE UpdateCustomerInfo(
	  IN customer_id_param INT,
	  IN new_customer_name_param VARCHAR(255),
	  IN new_customer_age_param INT,
	  IN new_customer_gender_param VARCHAR(255)
	)
	BEGIN
	  UPDATE customer
	  SET customer_name = new_customer_name_param,
	      customer_age = new_customer_age_param,
	      customer_gender = new_customer_gender_param
	  WHERE customer_id = customer_id_param;
	END //
	DELIMITER ;

Procedure to Cancel Booking:	DELIMITER //
	CREATE OR REPLACE PROCEDURE CancelBooking(
	  IN booking_id_param INT,
	  IN cancel_reason_param VARCHAR(255)
	)
	BEGIN
	  UPDATE booking
	  SET booking_status_id = 3, -- Assuming status code 3 represents canceled
	      booking_cancel_reason = cancel_reason_param
	  WHERE booking_id = booking_id_param;
	END //
	DELIMITER ;

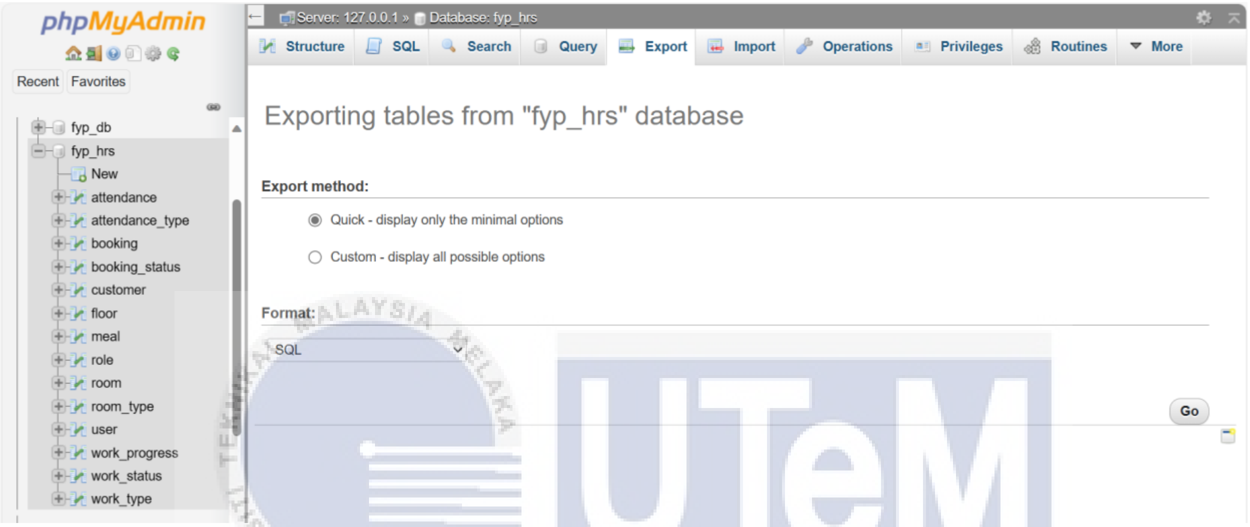
Procedure to Log Attendance:	DELIMITER //
	CREATE OR REPLACE PROCEDURE LogAttendance(
	  IN staff_id_param INT,
	  IN attendance_type_id_param INT
	)
	BEGIN
	  INSERT INTO attendance (attendance_datetime, attendance_type_id, staff_id)
	  VALUES (NOW(), attendance_type_id_param, staff_id_param);
	END //
	DELIMITER ;

Procedure to Get Available Rooms:	DELIMITER //
	CREATE OR REPLACE PROCEDURE GetAvailableRooms(
	  IN start_date_param DATETIME,
	  IN end_date_param DATETIME
	)
	BEGIN
	  SELECT room.*
	  FROM room
	  WHERE room.room_id NOT IN (
	    SELECT booking.room_id
	    FROM booking
	    WHERE NOT (
	      booking.booking_end_date <= start_date_param
	      OR booking.booking_start_date >= end_date_param
	    )
	  );
	END //
	DELIMITER ;

5.3.5 Data Loading Process

5.3.5.1 Export Data

Upon constructing a database, it can be exported, along with all its associated database objects, utilizing the export function in phpMyAdmin. The resulting database script can subsequently be employed on any computer or operating system. The export functionality of the DBMS is illustrated in Figure 5.18:



Exporting tables from "fyp_hrs" database

Export method:

● Quick - display only the minimal options

○ Custom - display all possible options

Format:

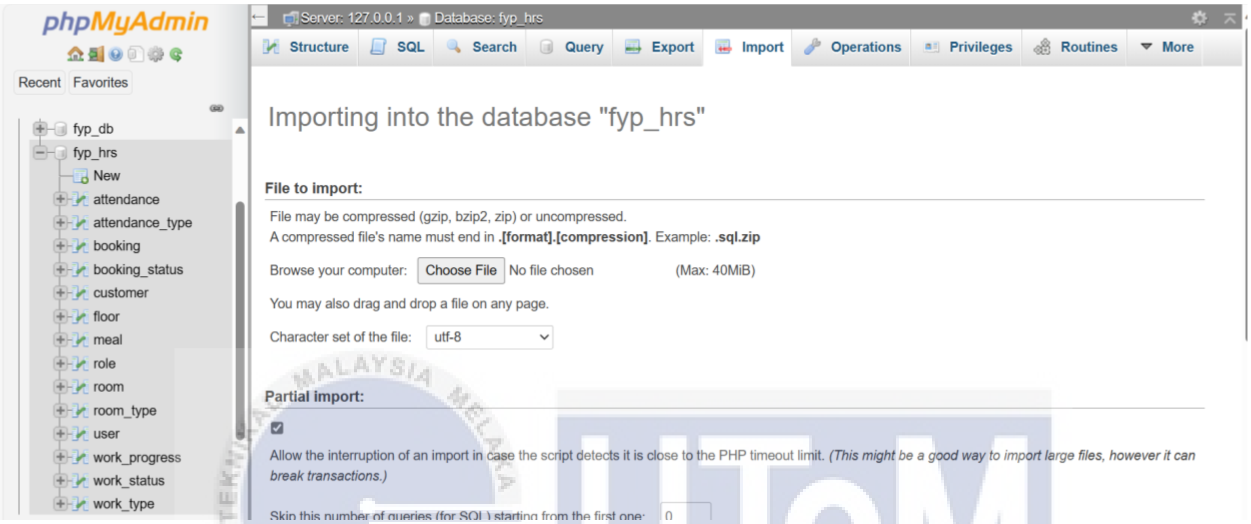
SQL

Figure 5.18 Export Database.

To initiate the export process, begin by clicking the "Export" button. Verify the accuracy of the database you intend to export. For a swift and straightforward export, opt for the "Quick" radio button and ensure that the format is set to "SQL." Click the "Go" button to generate the SQL script for the database, which is crucial for enabling the utilization of the database on alternate computer systems.

5.3.5.2 Import Data

Utilizing the phpMyAdmin import function, it becomes feasible to upload an entire database alongside its complete range of objects. This approach allows for universal database access across diverse computer systems, irrespective of their operating systems. The import function of the DBMS is visualized in Figure 5.19:



Importing into the database "fyp_hrs"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Choose File No file chosen (Max: 40MiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) starting from the first one: 0

Figure 5.19 Import Process

To initiate the utilization of the import function, it is necessary to first establish the database. Following this, click on the "Import" button and meticulously confirm that the designated target database for importing the database script is accurate. Subsequently, from your computer's storage, select the SQL file that requires loading. To execute the SQL file within the DBMS, finalize the process by pressing the "Go" button at the bottom of the page.

5.4 Conclusion

In conclusion, this chapter delves into the system's implementation, encompassing the setup of the software development environment and the database. It offers a comprehensive guide to establishing and executing both local and cloud databases on various servers within a system. The significance of maintaining multiple backups, spanning both local and cloud repositories, is underscored as a preventive measure against potential data loss, which could prove detrimental to the project's integrity.

Further insights into the system testing phase will be provided in the subsequent chapter. This will encompass the test planning process, incorporating aspects such as organizational considerations, the testing environment, hardware and firmware configurations, as well as pre-testing arrangements and training. Additionally, the forthcoming chapter will expound upon the testing strategy, design, test outcomes, and the subsequent in-depth analysis .

CHAPTER 6: TESTING

6.1 Introduction

In this chapter, the modules will undergo rigorous testing to determine their proficiency in effectively processing the dataset and yielding accurate outcomes. The testing phase, which holds paramount importance, serves as the final evaluation to ensure the robustness of the modules and the system's overall functionality. This critical testing phase aims to verify that the system operates seamlessly and remains resilient against any potential failures. Consequently, a comprehensive examination of key testing elements such as test strategy, test planning, test results, and other related aspects will be elaborated upon in depth.

6.2 Test Plan

A test plan is a crucial document that serves as a blueprint for the entire testing process. It meticulously outlines the approach, scope, resource allocation, and scheduling of planned testing activities. Within its pages, it identifies key elements such as the items slated for testing, specific features to be scrutinized, the assignment of testing tasks, the level of autonomy granted to individual testers, the designated test environment, the chosen testing methodologies, as well as the strategies and criteria for entering and exiting the testing phases. Additionally, it provides a detailed rationale for these decisions and includes a comprehensive assessment of any potential risks, warranting the need for contingency planning. In essence, the test plan serves as a comprehensive record of the entire test planning process, ensuring that testing efforts are well organized, transparent, and aligned with project objectives.

6.2.1 Test Organization

In Rosewood Reservation System, the test organization consists of developer. Scope that is tested by developer includes functional requirement and non-functional requirement. Moreover, tester finds the bugs in the web application and catches the failure or error of the outcome in this system as well as gives the rights privileges to admin together with limits the permission for user with or without account system.

Table 6.1 Roles and Responsibilities of organization

Name	Roles	Responsibilities
Sandra Moh Shu Yuan	Developer	- Develop this system using certain of programming language to design and build this whole system - Manage input and output of data by using stated database. - Fix error and upgrade the functionality system
Sandra Moh Shu Yuan & Tan Chee Hon	Tester	- Prepare the test plan, test environment, test schedule, test description, test data and test result and analysis - Find bugs
Low Zi Lun	Analyser	- Identify the requirement from clients - Collect data

6.2.2 Test Environment

Test environment comprises of components that help test execution with software, hardware and network configuration which is able to connect more than two components that setup by developer. Test environment design must copy the creation condition so as to reveal any condition or arrangement related issues. Table 6.2 presents the hardware components that being use during this development. On the other hand, Table 6.3 shows the programs and software computer which setup the web system and configuration of database.

Table 6.2 Test Environment of Hardware Components

Environment Specification	Description
Laptop	Fujitsu Lifebook L series
Central Processor Unit (CPU)	Intel i7 - 3612 QM
Keyboard	Aula Wings of Liberty RGB
Mouse	Logitech M100r
Radom Access Memory (RAM)	8 GB ddr3 1333mhz

Table 6.3 Test Environment of Software

Environment Specification	Description
Database	MySQL
Web Server	AppServer 8.6.0
Operating System	Windows 11 64-bit
Web Browser	Google Chrome
Development Tools	Adobe Dreamweaver CS6
Documentation	Microsoft Word 2016

6.2.3 Test Schedule

A test schedule is timetable for software testing that incorporates the testing steps or undertakings, the objective begin and finish dates, and duties. It ought to likewise depict how the test will be evaluated, followed, and affirmed. In table 6.4 states the activity, testing description, date of start and date of finish as well as the duration that have been taken during this testing.

Table 6.4 Test Schedule in RoseWood Resort Reservation System

Activities	Description	Start Date	End Date	Duration
Security Testing	Security testing is a sort of programming testing that means to reveal vulnerabilities of the framework and verify that its information and assets are shielded from potential interlopers or unauthorized user.	10/12/2023	17/12/2023	7 days
Functionality Testing	Functionality Testing is characterized as a sort of testing which checks that each capacity of the product application works in conformance with the requirement specification.	18/12/2023	27/12/2023	7 days

6.3 Test Strategy

Test Strategy has four types of testing including black box testing, white box testing, bottom-up testing and top-down testing. Black box testing, otherwise called Behavioural Testing, is a software testing approach in which the internal structure, design or implementation of the things being tested is not known to the tester. This testing basically is used to test the functional requirements compare to non-functional requirements. On the other hand, white box testing that known as Code-Based Testing or Structural Testing is software testing approach that is selected by tester to practice paths through the programming language and determines the appropriate outputs from the input resources. Moreover, bottom-up testing tests the each segment at lower progressive system is tested separately and after that the segments that depend upon these segments are tested. At the same time, top-down testing is a coordination testing system utilized so as to recreate the conduct of the lower-level modules that are not yet integrated. However, this project only covers the black box testing and top-down testing that test from the highest of hierarchy system until the lowest.

6.3.1 Classes of tests

There are two types of test class that is implemented for this testing process in Rosewood Reservation System.

i. **Functionality Testing**

This testing will check that each capacity of the product application works in conformance with the requirement specification.

ii. **Security Testing**

This testing will cover programming testing that means to reveal vulnerabilities of the framework and verify that its information and assets are shielded from potential interlopers or unauthorized user.

6.4 Test Design

Test design is a process that defines the method of testing should be done. It shows the possibility of testing that may happen in this system. This testing also involves the description of testing and the data that been to test for this web system.

6.5 Testing Results and Analysis

In this section, the system will be tested with many possible errors that can occurred such as human error that needs to be avoided. Furthermore, the system must have a strict policy so that the data that is stored must be in complete and in legal format. The testing phase will be started from the insert module. The testing should cover many aspects such as leaving fields empty and hitting the submit button will trigger an alert to pop up. Most of the testing is crucial to the system's flow, to make sure that everything is running smoothly.

6.5.1 Module/Unit: Login

System: Rosewood Resort Reservation System
Module/Unit: Login
Made by: Sandra
Actor: User, Admin or Staff

Test ID	Test Data	Expectation Result	Actual Result	Status
L_001	**Email:** usersandra@gmail.com **Password:** 123456	System will display "Successfully Login".	System display "Successfully Login".	Pass
L_002	**Email:** usersandra **Password:** 123456	System will display "Invalid username or password".	System display "Invalid username or password".	Pass
L_003	**Email:** usersandra@gmail.com **Password:** 1	System will display "Invalid username and password".	System display "Invalid username and password".	Pass
L_004	**Email:** **Password:**	System will display "Please fill out this field".	System display "Please fill out this field".	Pass
L_005	**Email:** usersandra@gmail.com **Password:**	System will display "Please fill out this field".	System display "Please fill out this field".	Pass
L_006	**Email:** **Password:** 12345	System will display "Please fill out this field".	System display "Please fill out this field".	Pass

Table 6.5 Test Case for Login

6.5.2 Module/Unit: Registration

System: Rosewood Resort Reservation System

Module/Unit: Registration

Made By: Sandra

Actor: User

Test ID	Test Data	Expectation Result	Actual Result	Status
R_001	**Username:** sandra **Email Address:** sandra12345@gmail.com **Contact Number:** 0182111234 **Password:** 123456	System will display "User sandra had sign up successfully.".	System display "User sandra had sign up successfully.".	Pass
R_002	**Username:** **Email Address:** sandra12345@gmail.com **Contact Number:** 0182111234 **Password:** 123456	System will display "Please fill out this field."	System display "Please fill out this field."	Pass
R_003	**Username:** sandra **Email Address:** **Contact Number:** 0182111234 **Password:** 123456	System will display "Please fill out this field."	System display "Please fill out this field."	Pass
R_004	**Username:** sandra **Email Address:** sandra12345@gmail.com **Contact Number:** **Password:** 123456	System will display "Please fill out this field."	System display "Please fill out this field."	Pass

R_005	**Username:** **Email Address:** sandra12345@gmail.com **Contact Number:** 0182111234 **Password:**	System will display "Please fill out this field."	System display "Please fill out this field."	Pass
R_006	**Username:** sandra **Email Address:** shuyuan010726gmail.com **Contact Number:** 0182111234 **Password:** 123456	System will display "Please include an '@' in the email address. shuyuan010726gmail.com is missing an '@'.	System display "Please include an '@' in the email address. shuyuan010726gmail.com is missing an '@'.	Pass
R_007	**Username:** **Email Address:** **Contact Number:** **Password:**	System will display "Please fill out this field."	System display "Please fill out this field."	Pass
R_008	**Username:** sandra **Email Address:** shuyuan010726@gmail.com **Contact Number:** 0182111234 **Password:** 12	System will display "Please lengthen this text to 5 characters or more (you are currently using 2 characters). "	System display "Please lengthen this text to 5 characters or more (you are currently using 2 characters). "	Pass

Table 6.6 Test Case for Register

6.5.3 Module/Unit: Update Account Information

System: Rosewood Resort Reservation System

Module/Unit: Update Account Information

Made By: Sandra

Actor: User, Admin and Staff

Test ID	Test Data	Expectation Result	Actual Result	Status
UA_01	**Username:** sandra **Email Address:** sandra12345@gmail.com **Contact Number:** 0182111234 **Your Password:** 123456 **Confirm Password:** 123456	System will display “Profile is updated.”.	System display “Profile is updated.”.	Pass
UA_02	**Username:** **Email Address:** sandra12345@gmail.com **Contact Number:** 0182111234 **Your Password:** 123456 **Confirm Password:** 123456	System will display “Please fill up this field.”.	System display “Please fill up this field.”.	Pass
UA_03	**Username:** sandra **Email Address:** sandra12345@gmail.com **Contact Number:** **Your Password:** 123456 **Confirm Password:** 123456	System will display “Please fill up this field.”.	System display “Please fill up this field.”.	Pass

UA_04	**Username:** sandra **Email Address:** sandra12345@gmail.com **Contact Number:** 0182111234 **Your Password:** **Confirm Password:** 123456	System will display “New password and confirm password is different. Please fill it both same if want to update password. ”.	System display “New password and confirm password is different. Please fill it both same if want to update password. ”.	Pass
UA_05	**Username:** sandra **Email Address:** sandra12345@gmail.com **Contact Number:** 0182111234 **Your Password:** 123456 **Confirm Password:**	System will display “New password and confirm password is different. Please fill it both same if want to update password.”.	System display “New password and confirm password is different. Please fill it both same if want to update password.”.	Pass
UA_06	**Username:** **Email Address:** sandra12345@gmail.com **Contact Number:** **Your Password:** **Confirm Password:**	System will display “Please fill up this field.”.	System display “Please fill up this field.”.	Pass

Table 6.7 Test Case for Update Account Information

6.5.4 Module/Unit: Manage User.

System: Rosewood Hotel Reservation System

Module/Unit: Manage User.

Made By: Sandra

Actor: Admin

Test ID	Test Data	Expectation Result	Actual Result	Status
		Update User Role		
UR_01	**Email:**Admin001@gmail.com **Username:** Admin01 **Contact Number:** 0182736485 **Role User:** Admin **Account Activation:** Active ->Click change to user button.	System will display "Role has updated.".	System display "Role has updated.".	Pass
UR_02	**Email:**User001@gmail.com **Username:** user01 **Contact Number:** 0182736485 **Role User:** User **Account Activation:** Active -> Click change to admin button.	System will display "Role has updated.".	System display "Role has updated.".	Pass
UR_03	**Email**:User001@gmail.com **Username**: user01 **Contact Number:** 0182736485 **Role User:** User **Account Activation:** Active -> Click change to Staff button.	System will display "Role has updated.".	System display "Role has updated.".	Pass
		Block User		
BU_01	**Email:** User001@gmail.com **Username:** user01 **Contact Number:** 0182736485 **Role User:** User **Account Activation:** Active -> Click block user button.	System will display "Account has been locked.".	System display "Account has been locked.".	Pass
BU_02	**Email:** User001@gmail.com **Username:** user01 **Contact Number:** 0182736485 **Role User:** User **Account Activation:** Inactive -> Click unblock user button.	System will display "Account has been unlocked.".	System display "Account has been unlocked.".	Pass

Add User				
AU_01	**Username:** Sandra **Email Address:** shuyuan072601@gmail.com **Contact Number:** 0182111468 **Role:** User -> Click Confirm button.	System will display “Successfully added account.”	System display “Successfully added account.”	Pass
AU_02	**Username:** **Email Address:** shuyuan072601@gmail.com **Contact Number:** 0182111468 **Role:** User -> Click Confirm button.	System will display “Please fill up this field.”.	System display “Please fill up this field.”.	Pass
AU_03	**Username:** Sandra **Email Address:** shuyuan072601 **Contact Number:** 0182111468 **Role:** User -> Click Confirm button.	System will display “Please include an ‘@’ in the email address. shuyuan07260 is missing an ‘@’.	System display “Please include an ‘@’ in the email address. shuyuan07260 is missing an ‘@’.	Pass
AU_04	**Username:** Sandra **Email Address:** **Contact Number:** 0182111468 **Role:** User -> Click Confirm button.	System will display “Please fill up this field.”.	System display “Please fill up this field.”.	Pass
AU_05	**Username:** Sandra **Email Address:** shuyuan072601@gmail.com **Contact Number:** **Role:** User -> Click Confirm button.	System will display “Please fill up this field.”.	System display “Please fill up this field.”.	Pass
AU_06	**Username:** Sandra **Email Address:** adminaisyah@gmail.com **Contact Number:** **Role:** Admin -> Click Confirm button.	System will display “Successfully added account.”	System display “Successfully added account.”	Pass

Table 6.8 Test Case for Manage User

6.5.5 Module/Unit: Manage Booking.

System: Rosewood Resort Reservation System

Module/Unit: Manage Booking.

Made By: Sandra

Actor: Admin

Test ID	Test Data	Expectation Result	Actual Result	Status
MB_01	**Room Type:** Single Room **Room No:** GF-S-001 **Phone Number :**0182937453 **Booking Start Date:** 2023-01-22 **Booking End Date**: 2023-01-31 **Booking Price:** RM 1782 **Status:** Reject	System will display "Update successfully.".	System display "Update successfully.".	Pass
MB_02	**Room Type:** Single Room **Room No:** GF-S-001 **Booking Start Date**: 2022-01-22 **Booking End Date:** 2022-01-31 **Booking Price:** RM 1782 **Status:** Approve	System will display "Update successfully.".	System display "Update successfully.".	Pass

Table 6.9 Test Case for Manage Booking.

6.5.6 Module/Unit: Manage Room

System: Rosewood Resort Reservation System

Module/Unit: Manage Room

Made By: Sandra

Actor: Admin

Test ID	Test Data	Expectation Result	Actual Result	Status
Update Room				
UR_01	**Room Type:** Premium Room **Floor:** 2nd Floor **Room No:** L2-P-005 **Room Activation:** ☑	System will display "Successfully updated room.".	System display "Successfully updated room.".	Pass
UR_02	**Room Type:** Single Room **Floor:** Ground Floor **Room No:** GF-S-001 **Room Activation:** ☑	System will display "Successfully updated room.".	System display "Successfully updated room.".	Pass
UR_03	**Room Type:** Double Room **Floor:** 1st Floor **Room No:** L1-D-003 **Room Activation:** ☑	System will display "Successfully updated room.".	System display "Successfully updated room.".	Pass
UR_04	**Room Type:** Double Room **Floor:** 1st Floor **Room No:** **Room Activation:** ☑	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass
UR_05	**Room Type:** Double Room **Floor:** 1st Floor **Room No:** L1-D-003 **Room Activation:**	System will display "Successfully updated room.".	System display "Successfully updated room.".	Pass

Edit Room Type				
ER_01	**Room Type:** Premium Room **Room Price:** 15 **Room Image:** premiumroom.jpg	System will display “Successfully updated room type.”.	System display “Successfully updated room type.”.	Pass
ER_02	**Room Type:** Single Room **Room Price:** 8 **Room Image:** singleroom.jpg	System will display “Successfully updated room type.”.	System display “Successfully updated room type.”.	Pass
ER_03	**Room Type:** Double Room **Room Price:** 10 **Room Image:** doubleroom.jpg	System will display “Successfully updated room type.”.	System display “Successfully updated room type.”.	Pass
ER_04	**Room Type:** **Room Price:** 12 **Room Image:** doubleroom.jpg	System will display “Please select an item in the list.”.	System display “Please select an item in the list.”.	Pass
ER_05	**Room Type:** Double room **Room Price:** **Room Image:** doubleroom.jpg	System will display “Please fill out this field.”.	System display “Please fill out this field.”.	Pass

Add Room				
AR_01	**Room Type:** Premium Room **Floor:** 2nd Floor **Room No:** L2-P-005 **Room Activation:** ☑	System will display “Successfully added room.”.	System display “Successfully added room.”.	Pass
AR_02	**Room Type:** Single Room **Floor:** Ground Floor **Room No:** GF-S-001 **Room Activation:** ☑	System will display “Successfully added room.”.	System display “Successfully added room.”.	Pass
AR_03	**Room Type:** Double Room **Floor:** 1st Floor **Room No:** L1-D-003 **Room Activation:** ☑	System will display “Successfully added room.”.	System display “Successfully added room.”.	Pass
AR_04	**Room Type:** Double Room **Floor:** 1st Floor **Room No:** **Room Activation:** ☑	System will display “Please fill out this field.”.	System display “Please fill out this field.”.	Pass
AR_05	**Room Type:** Double Room **Floor:** 1st Floor **Room No:** L1-D-003 **Room Activation:**	System will display “Successfully added room.”.	System display “Successfully added room.”.	Pass

Table 6.10 Test Case for Manage Room.

6.5.7 Module/Unit: View Statistic

System: Rosewood Resort Reservation System

Module/Unit: View Statistic

Made By: Sandra

Actor: Admin

Test ID	Test Data	Expectation Result	Actual Result	Status
VS_01	**Room Type:** Premium Room **Start Date:** 01/12/2022 **End Date**: 09/01/2023	System will display statistic report of the booking.	System display statistic report of the booking.	Pass
VS_02	**Room Type:** Premium Room **Start Date:** **End Date:** 09/01/2023	System will display “Please fill out this field.”.	System display “Please fill out this field.”.	Pass
VS_03	**Room Type**: Premium Room **Start Date:** 01/12/2023 **End Date:**	System will display “Please fill out this field.”.	System display “Please fill out this field.”.	Pass
VS_04	**Room Type**: Single Room **Start Date:** 01/12/2023 **End Date:** 09/01/2023	System will display statistic report of the booking.	System display statistic report of the booking.	Pass
VS_05	**Room Type:** Double Room **Start Date:** 01/12/2023 **End Date:** 09/01/2023	System will display statistic report of the booking	System display statistic report of the booking	Pass
VS_06	**Room Type:** All Room Type **Start Date:** 01/12/2022 **End Date:** 09/01/2023	System will display statistic report of the booking.	System display statistic report of the booking.	Pass

Table 6.11 Test Case for View Statistic.

6.5.8 Module/Unit: View Availability (Admin Page)

System: Rosewood Resort Reservation System

Module/Unit: View Availability (Admin Page)

Made By: Sandra

Actor: Admin

Test ID	Test Data	Expectation Result	Actual Result	Status
VA_01	**Room Type:** Single Room **Start Date:** 01/13/2023 **End Date**: 01/14/2023	System will display availability of Single Room.	System display availability of Single Room.	Pass
VA_02	**Room Type:** Double Room **Start Date:** 01/13/2023 **End Date:** 01/14/2023	System will display availability of Double Room.	System display availability of Double Room.	Pass
VA_03	**Room Type:** Premium Room **Start Date:** 01/13/2023 **End Date:** 01/14/2023	System will display availability of Premium Room.	System display availability of Premium Room.	Pass
VA_04	**Room Type:** Hall **Start Date:** 01/13/2023 **End Date:** 01/14/2023	System will display availability of Hall.	System display availability of Hall.	Pass
VA_05	**Room Type:** Single Room **Start Date:** 01/13/2023 **End Date:** 01/10/2023	System will display "Search End Date must greater than Search Start Date" .	System display "Search End Date must greater than Search Start Date" .	Pass
VA_06	**Room Type:** Single Room **Start Date:** 01/13/2023 **End Date:**	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass
VA_07	**Room Type**: All Room / Hall **Start Date:** 01/13/2023 **End Date:** 01/14/2023	System will display availability of All Room /Hall.	System display availability of All Room /Hall.	Pass

Table 6.12 Test Case for View Availability (Admin Page)

6.5.9 Module/Unit: Book Room

System: Rosewood Resort Reservation System

Module/Unit: Book Room

Made By: Sandra

Actor: User

Test ID	Test Data	Expectation Result	Actual Result	Status
BR_01	**RoomType:** Single Room **Number of People:** 2 **Start Date:** 3/11/2023 2:00 pm **End Date:** 3/12/2023 12:00 pm **Customer Name 1:** Ali **Customer Name 2:** Abu **Customer Age 1:** 20 **Customer Age 2**: 22 **Customer Gender 1:** Male **Customer Gender 2:** Male **Term and Condition :** ☑	System will display "Booking successfully."	System display "Booking successfully."	Pass
BR_02	**RoomType:** Double Room **Number of People:** 2 **Start Date:** 3/11/2023 2:00 pm **End Date:** 3/12/2023 12:00 pm **Customer Name 1:** Ali **Customer Name 2:** Abu **Customer Age 1:** 20 **Customer Age 2:** 22 **Customer Gender 1:** Male **Customer Gender 2:** Male **Term and Condition :** ☑	System will display "Booking successfully."	System display "Booking successfully."	Pass

BR_03	**RoomType:** Premium Room **Number of People:** 2 **Start Date:** 3/11/2023 2:00 pm **End Date:** 3/12/2023 12:00 pm **Customer Name 1:** Ali **Customer Name 2:** Abu **Customer Age 1:** 20 **Customer Age 2:** 22 **Customer Gender 1:** Male **Customer Gender 2:** Male **Term and Condition :** ☑	System will display "Booking successfully."	System display "Booking successfully."	Pass
BR_04	**RoomType:** Single Room **Number of People:** 1 **Start Date:** 3/11/2023 2:00 pm **End Date:** 3/12/2023 12:00 pm **Customer Name 1:** Ali **Customer Age 1:** 20 **Customer Gender 1:** Male **Table 6.5.**☑	System will display "Booking successfully."	System display "Booking successfully."	Pass
BR_05	**RoomType:** Single Room **Number of People:** 2 **Start Date:** **End Date:** **Customer Name 1:** Ali **Customer Name 2:** Abu **Customer Age 1:** 20 **Customer Age 2:** 22 **Customer Gender 1:** Male **Customer Gender 2:** Male **Term and Condition :** ☑	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass

BR_06	**RoomType:** Single Room **Number of People:** 2 **Start Date:** 3/12/2023 2:00 pm **End Date:** 3/11/2023 12:00 pm **Customer Name 1:** **Customer Name 2:** **Customer Age 1:** **Customer Age 2:** **Customer Gender 1:** **Customer Gender 2:** **Term and Condition :** ☑	System will display “Search End Date must greater than Search Start Date.”	System display “Search End Date must greater than Search Start Date.”	Pass
BR_07	**RoomType:** Single Room **Number of People:** 2 **Start Date:** 3/11/2023 2:00 pm **End Date:** 3/12/2023 12:00 pm **Customer Name 1:** Ali **Customer Name 2:** **Customer Age 1:** **Customer Age 2:** 22 **Customer Gender 1:** **Customer Gender 2:** Male **Term and Condition :** ☑	System will display “Please fill out this field.”.	System display “Please fill out this field.”.	Pass
BR_08	**RoomType**: Double Room **Number of People:** 2 **Start Date:** 3/11/2023 2:00 pm **End Date:** 3/12/2023 12:00 pm **Customer Name 1**: Ali **Customer Name 2**: Abu **Customer Age 1:** 20 **Customer Age 2**: 22 **Customer Gender 1**: Male	System will display “Please check this box if you want to proceed.”	System display “Please check this box if you want to proceed.”	Pass

	Customer Gender 2: Male **Term and Condition:** ☐			

Table 6.13 Test Case for Book Room.

6.5.10 Module/Unit: Book Hall

System: Rosewood Resort Reservation System

Module/Unit: Book Hall

Made By: Sandra

Actor: User

Test ID	Test Data	Expectation Result	Actual Result	Status
BH_01	**Start Date:** 3/12/2023 2:00 pm **End Date:** 4/12/2023 12:00 pm **Hall:** GF-H-03 **Package:** Package A **Term and Condition:** ☑	System will display "Booking successfully."	System display "Booking successfully."	Pass
BH_02	**Start Date:** 3/10/2023 2:00 pm **End Date:** 3/9/2023 12:00 pm **Hall:** GF-H-03 **Package:** Package A **Term and Condition:** ☑	System will display "Search End Date must greater than Search Start Date."	System display "Search End Date must greater than Search Start Date."	Pass
BH_03	**Start Date:** 3/12/2023 2:00 pm **End Date:** **Hall:** GF-H-03 **Package:** Package A **Term and Condition:** ☑	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass
BH_04	**Start Date:** **End Date:** 3/11/2023 12:00 pm **Hall:** GF-H-03 **Package:** Package A **Term and Condition:** ☑	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass

BH_05	**Start Date:** 3/12/2023 2:00 pm **End Date:** 4/12/2023 12:00 pm **Hall:** **Package:** Package A **Term and Condition:** ☑	System will display “Please select one of these option.”.	System display “Please select one of these option.”.	Pass
BH_06	**Start Date:** 3/12/2023 2:00 pm **End Date:** 4/12/2023 12:00 pm **Hall:** GF-H-03 **Package:** **Term and Condition:** ☑	System will display “Please select an item in the list.”.	System display “Please select an item in the list.”.	Pass
BH_07	**Start Date:** 3/12/2023 2:00 pm **End Date:** 4/12/2023 12:00 pm **Hall:** GF-H-03 **Package:** Package A **Term and Condition:** ☑	System will display “Please check this box if you want to proceed.”	System display “Please check this box if you want to proceed.”	Pass

Table 6.14 Test Case for Book Hall.

6.5.11 Module/Unit: View My Booking

System: Rosewood Resort Reservation System

Module/Unit: View My Booking

Made By: Sandra

Actor: User

Test ID	Test Data	Expectation Result	Actual Result	Status
VB_01	->User upload payment receipt	System will display “Successfully paid.”.	System display “Successfully paid.”.	Pass
VB_02	->User click cancel booking button	System will display “Booking has been cancelled.”.	System display “Booking has been cancelled.”.	Pass

Table 6.15 Test Case for View My Booking.

6.5.12 Module/Unit: Manage Staff

System: Rosewood Resort Reservation System

Module/Unit: Manage Staff

Made By: Sandra

Actor: Admin

Test ID	Test Data	Expectation Result	Actual Result	Status
		View Attendance		
VA_01	-> Click View Attendance button.	System will display Attendance Record of that staff.	System display Attendance Record of that staff.	Pass
		View Work Sheets		
VWS_01	-> Click View Work Sheet button.	System will display Work Sheet Record of that staff.	System display Work Sheet Record of that staff.	Pass
		Add Work Sheets		
AWS_01	**Work Type:** Clean & Maintain guest rooms. **Date Time:** 26/12/23 17:39 PM **Work Details :** clean room 5	System will display "Successfully Added Work Sheet.".	System display "Successfully Added Work Sheet.".	Pass
AWS_02	**Work Type:** Clean & Maintain guest rooms. **Date Time:** **Work Details :** clean room 5	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass
AWS_03	**Work Type:** Clean & Maintain guest rooms. **Date Time:** 26/12/23 17:39 PM **Work Details :**	System will display "Successfully Added Work Sheet.".	System display "Successfully Added Work Sheet.".	Pass

AWS_04	**Work Type:** **Date Time:** 26/12/23 17:39 PM **Work Details :** clean room 5	System will display “Please select work type.”.	System display “Please select work type.”.	Pass

Table 6.16 Test Case for Manage Staff.

6.5.13 Module/Unit: My Attendance

System: Rosewood Resort Reservation System

Module/Unit: My Attendance

Made By: Sandra

Actor: Staff

Test ID	Test Data	Expectation Result	Actual Result	Status
ATD_01	**Date Time:** 12/26/2023 08:32:36 PM **Attendance Type:** Clock In	System will display "Attendance successfully added.".	System display "Attendance successfully added.".	Pass
ATD_02	**Date Time:** 12/26/2023 12:30:40 PM **Attendance Type:** Lunch Hour	System will display "Attendance successfully added.".	System display "Attendance successfully added.".	Pass
ATD_03	**Date Time:** 12/26/2023 1:32:50 PM **Attendance Type:** Resume Work	System will display "Attendance successfully added.".	System display "Attendance successfully added.".	Pass
ATD_04	**Date Time:** 12/26/2023 06:32:33 PM **Attendance Type:** Clock Out	System will display "Attendance successfully added.".	System display "Attendance successfully added.".	Pass

Table 6.17 Test Case for My Attendance.

6.5.14 Module/Unit: Manage Work

System: Rosewood Resort Reservation System

Module/Unit: Manage Work

Made By: Sandra

Actor: Staff

Test ID	Test Data	Expectation Result	Actual Result	Status
MW_01	**Work Status:** pending **Staff's Remark:**	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass
MW_02	**Work Status:** incomplete **Staff's Remark:**	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass
MW_03	**Work Status:** complete **Staff's Remark:**	System will display "Please fill out this field.".	System display "Please fill out this field.".	Pass
MW_04	**Work Status:** complete **Staff's Remark:** done clean	System will display "Work Sheet successfully updated.".	System display "Work Sheet successfully updated.".	Pass

Table 6.18 Test Case for Manage Work Sheet.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

The project conclusion is the pinnacle of the whole endeavor and summarizes the major discoveries, revelations, and results attained throughout the course of the project's lifespan. It gives a chance to consider the accomplishments, difficulties, and lessons discovered along the way. Project stakeholders can evaluate whether the project's aims and objectives have been reached during this ending phase and come to significant conclusions that will guide their future work. Additionally, comments, areas for improvement, and a final assessment of the project's effect and contributions are frequently included in the project conclusion. It is an essential component of project closure since it enables a thorough evaluation of the project's overall performance and offers insightful information for future endeavors.

7.2 Observation on Weaknesses and Strengths

A critical evaluation that reveals areas for improvement and areas of excellence within a project or process involves observation of weaknesses and strengths. The strengths and weaknesses that contribute to Henry Sports Equipment Rental Web Based System are:

7.2.1 Strengths of the Rosewood Resort Reservation System

The Rosewood Resort Reservation System stands as a formidable solution with a multitude of strengths, positioning it as an invaluable asset for both customers and resort administrators alike. At the forefront is its intuitive and user-friendly interface, offering a seamless booking experience for customers seeking travel accommodations or event spaces. The system's versatility shines through its ability to cater to a diverse range of needs, allowing customers to reserve not only rooms for travel but also halls for weddings or parties. The inclusion of a comprehensive booking history feature adds a layer of transparency, enabling customers to effortlessly track and manage their reservations over time.

Administratively, the system demonstrates remarkable strength by empowering resort administrators with efficient account management tools. Admins can easily oversee and manage both customer and staff accounts, streamlining the administrative process. The robust reporting functionality is another highlight, providing admins with the capability to generate detailed sales reports categorized by year, month, or week. This analytical power enables administrators to gain valuable insights into resort performance, facilitating informed decision-making for strategic planning and resource allocation.

The system's task assignment feature fosters seamless communication and coordination among staff members. Admins can efficiently assign tasks to the resort's staff, who, in turn, provide real-time updates on their work status. The attendance tracking feature ensures accurate timekeeping, recording clock-in, breaks, and clock-out times, contributing to enhanced workforce management.

In summary, the strengths of the Rosewood Resort Reservation System lie in its user-centric design, versatile booking options, robust administrative tools, and analytical capabilities. These features collectively contribute to a well-rounded and efficient system that caters to the diverse needs of customers and empowers resort administrators to manage operations with precision and insight.

7.2.2 Weaknesses of the Rosewood Resort Reservation System

While the Rosewood Resort Reservation System exhibits notable strengths, it is not without its share of weaknesses that warrant attention and improvement. One significant drawback lies in the manual verification process for customer payment receipts. The current system requires administrators to individually inspect and validate each uploaded payment receipt before updating booking statuses, introducing a potential bottleneck during peak reservation periods and increasing the likelihood of delays. This manual verification process can be time-consuming and resource-intensive, calling for a more streamlined and automated approach to enhance efficiency.

Another weakness arises in the system's room and hall management features. While it allows for the editing of details and addition of new accommodations, there's a risk of errors if not executed with precision. The system could benefit from a more user-friendly interface and additional validation checks to reduce the likelihood of inaccuracies. Additionally, the lack of real-time updates on room and hall availability may lead to discrepancies between displayed information and actual availability, potentially affecting the accuracy of bookings.

Furthermore, the system's dependence on manual intervention for certain administrative tasks, such as editing room and hall details, and the verification of payment receipts, poses a challenge to scalability. As the resort's operations expand, these manual processes may become increasingly cumbersome, highlighting the need for automated solutions to handle a growing volume of transactions seamlessly.

In conclusion, addressing these weaknesses, such as implementing automated verification processes, refining room and hall management features, and reducing dependency on manual interventions, will be crucial for the Rosewood Resort Reservation System to reach its full potential, ensuring a smoother and more efficient experience for both customers and administrators.

7.3 Propositions for Improvement in the Rosewood Resort Reservation System

To address the identified weaknesses in the Rosewood Resort Reservation System, several strategic propositions for improvement are essential. Firstly, the manual verification process for customer payment receipts can be streamlined through the implementation of automated verification algorithms. By incorporating advanced algorithms, the system can autonomously validate payment receipts, reducing the burden on administrators and significantly expediting the approval or rejection of bookings. This enhancement not only mitigates delays during peak reservation periods but also ensures a more efficient and error-free process, enhancing overall customer satisfaction.

Secondly, refining the room and hall management features is pivotal for improving accuracy and user experience. A more intuitive user interface with built-in validation checks can minimize errors during the editing of details or addition of new accommodations. Real-time updates on room and hall availability should be integrated to provide customers and administrators with accurate and up-to-date information, eliminating discrepancies between displayed availability and the actual status of accommodations. These improvements will contribute to a more reliable booking process and streamline administrative tasks related to resource management.

Additionally, the system's scalability can be enhanced by reducing manual interventions in administrative tasks. Introducing automation for routine processes, such as editing room and hall details, can ensure that the system remains efficient and effective as the resort's operations expand. Automation not only reduces the potential for errors but also optimizes resource utilization, allowing administrators to focus on more strategic aspects of resort management.

In summary, these propositions for improvement aim to fortify the Rosewood Resort Reservation System by introducing automation, refining user interfaces, and ensuring real-time updates. By addressing these key areas, the system can overcome its weaknesses, providing a more seamless experience for both customers and administrators and positioning itself as a robust and scalable solution for the resort's growing needs.

7.4 Contribution

The Rosewood Resort Reservation System makes significant contributions to the overall efficiency, transparency, and management of the resort's operations. One of its primary contributions lies in providing customers with a seamless and user-friendly booking experience. The intuitive interface allows customers to effortlessly reserve travel accommodations or event spaces like halls for weddings or parties. The inclusion of a comprehensive booking history feature enhances transparency, enabling customers to track and manage their reservations over time, fostering trust and satisfaction.

From an administrative standpoint, the system contributes by streamlining account management for both customers and staff. Admins can efficiently oversee and manage accounts, add new staff members, and generate detailed sales reports. The reporting functionality empowers administrators with valuable insights into resort performance, enabling data-driven decision-making for strategic planning and resource allocation. The system's task assignment and attendance tracking features further contribute to improved workforce management, ensuring efficient communication and accurate timekeeping.

Moreover, the system enhances the overall organization of the resort by automating and centralizing key processes. The ability to assign tasks to staff and receive real-time updates on work status facilitates smoother communication and coordination. The attendance tracking feature adds an additional layer of organization, ensuring accurate time records for staff, which is crucial for payroll and workforce planning.

In summary, the Rosewood Resort Reservation System contributes significantly to the resort's success by providing a seamless experience for customers, streamlining administrative tasks, offering valuable insights through reporting, and enhancing overall organizational efficiency. Its multifaceted capabilities contribute to a more efficient, transparent, and well-managed resort operation, positively impacting both customer satisfaction and the resort's bottom line.

7.5 Conclusion

In conclusion, the Rosewood Resort Reservation System has met the objective that is stated in chapter one. The Rosewood Resort Reservation System emerges as a pivotal and transformative tool in the management and operation of the resort. Its strengths, including an intuitive user interface, versatile booking options, and robust administrative features, contribute significantly to customer satisfaction and efficient resort administration. The system's ability to streamline tasks, automate processes, and provide real-time insights through reporting enhances organizational efficiency and strategic decision-making.

While the system exhibits strengths that underscore its positive impact, it is not without weaknesses. The manual verification process for payment receipts and the potential for errors in room and hall management highlight areas for improvement. However, these weaknesses present opportunities for refinement and optimization, ensuring that the system can evolve to meet the growing demands of the resort.

The propositions for improvement, such as the introduction of automated verification algorithms, user interface enhancements, and increased automation for administrative tasks, provide a roadmap for addressing these weaknesses. By implementing these improvements, the Rosewood Resort Reservation System can further solidify its role as a cornerstone in the resort's success, offering an enhanced and streamlined experience for both customers and administrators.

In essence, the Rosewood Resort Reservation System stands as a dynamic and adaptive solution, integral to the resort's operational success. As technology continues to advance, ongoing improvements and strategic enhancements will be key to ensuring the system's resilience and continued positive contribution to the resort's overall efficiency, customer satisfaction, and long-term success in the competitive hospitality industry.

REFERENCES

JavaTpoint. (n.d.). Unit testing - javatpoint. www.javatpoint.com. Retrieved from: https://www.javatpoint.com/unit-testing

W3schools.com - Retrieved from : http://www.w3schools.com/html/default.asp

HTML color codes - Retrieved from : https://htmlcolorcodes.com/

Xampp - Retrieved from :

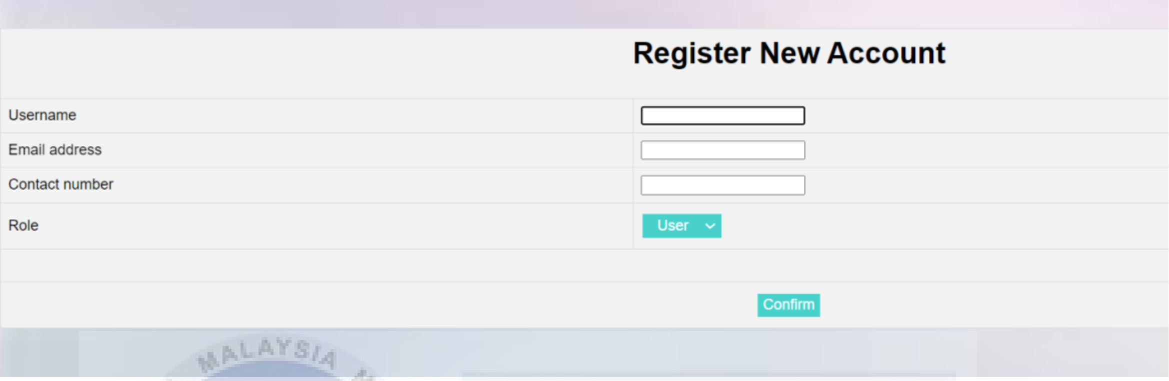
- https://www.apachefriends.org/index.html

- https://en.wikipedia.org/wiki/XAMPP#:~:text=XAMPP%20(%2F%CB%88z

APPENDIX A

Admin add new account and email will be sent process

1.Admin add new user (one account)

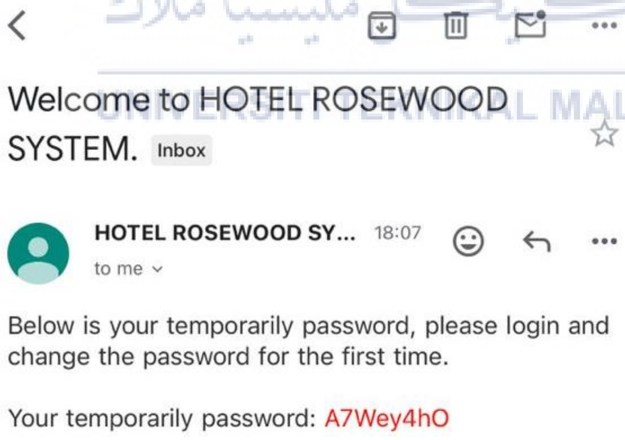


Register New Account

Username	
Email address	
Contact number	
Role	User

Figure A.01 Admin Add New User (one account)

2.User will receive an email to change password when their First login.



Welcome to HOTEL ROSEWOOD SYSTEM.

HOTEL ROSEWOOD SY... 18:07

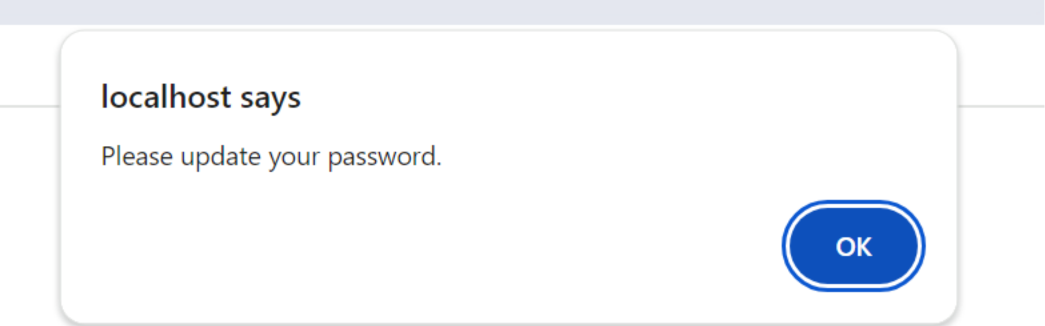
to me

Below is your temporarily password, please login and change the password for the first time.

Your temporarily password: A7Wey4hO

Figure A.02 User Receive Temporarily Password in Email

3. **System will ask user to change password when first login**

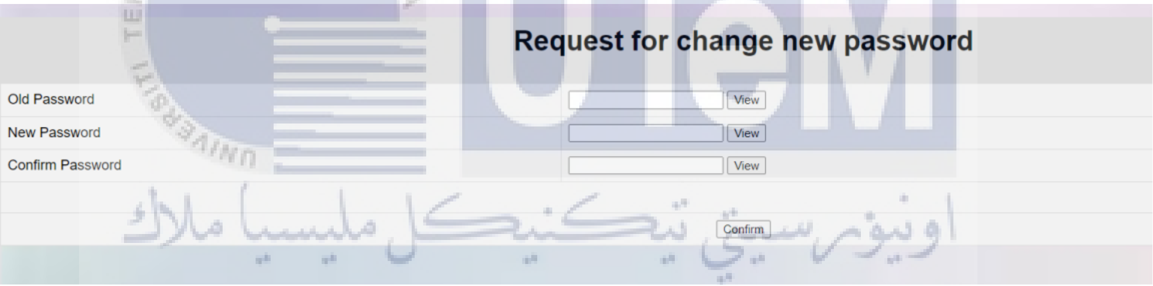


localhost says

Please update your password.

Figure A.03 Notification Change Password for Fist Login

4. **User need to change password when first login.**



Request for change new password

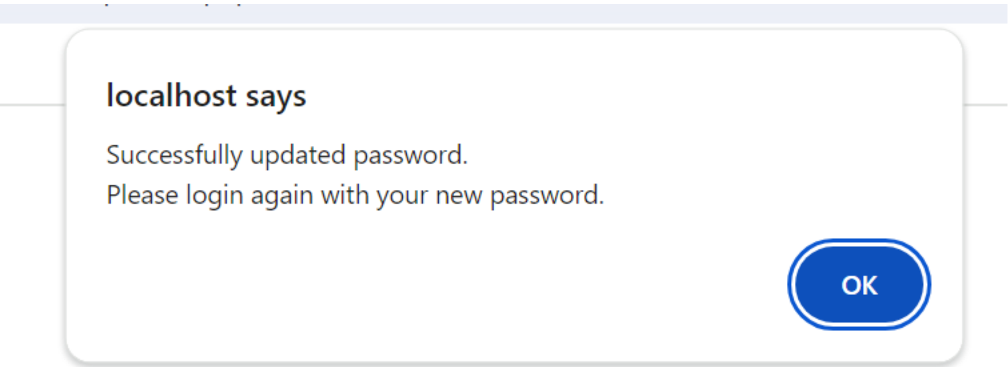
Old Password

New Password

Confirm Password

Figure A.04 Change Password for Fist Login

5. **Notification will display the password updated**

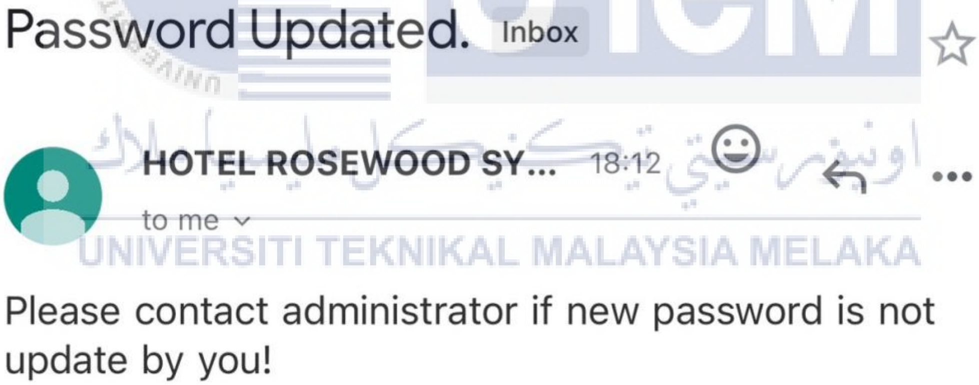


localhost says

Successfully updated password.
Please login again with your new password.

Figure A.05 Notification Password Updated.

6. **New email will be sent to user.**

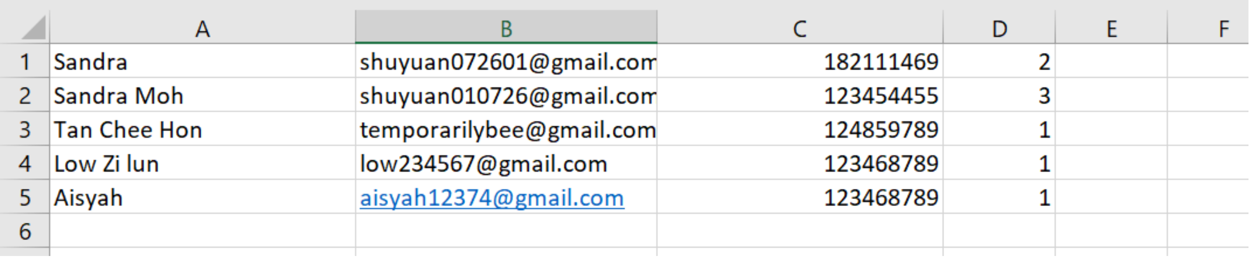


Password Updated.

Please contact administrator if new password is not update by you!

Figure A.06 Email Password Updated.

7. Admin can also import new user by using csv file when admin wish to import a group of people.



	A	B	C	D	E	F
1	Sandra	shuyuan072601@gmail.com	182111469	2		
2	Sandra Moh	shuyuan010726@gmail.com	123454455	3		
3	Tan Chee Hon	temporarilybee@gmail.com	124859789	1		
4	Low Zi lun	low234567@gmail.com	123468789	1		
5	Aisyah	aisyah12374@gmail.com	123468789	1		
6						

Figure A.07 Using Excel import Group of People.