FYP APPOINTMENT BOOKING SYSTEM

NURUL AINA MARINA BINTI MANSOR

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FYP APPOINTMENT BOOKING SYSTEM

NURUL AINA MARINA BINTI MANSOR

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

[FYP APPOINTMENT BOOKING SYSTEM]

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

STUDENT : ________________________________ Date : 27/8/2024

(NURUL AINA MARINA BINTI MANSOR)

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 : 3/9/2024

(Noor Azilah Binti Draman @ Muda)

DEDICATION

I am deeply grateful to everyone who has supported, inspired, motivated, and uplifted me throughout this journey. Your encouragement and belief in me have been invaluable.

Special thanks to my loving parents, whose endless support and guidance have been my foundation. Your unconditional love and care have been the driving force behind my achievements. To my lecturers, your wisdom and dedication to teaching have profoundly shaped my knowledge and perspective. Your mentorship has been a beacon of light, guiding me through the complexities of my academic pursuits.

To my dear friends, your unwavering support and companionship have been a source of strength and joy. Thank you for being there through the highs and lows, celebrating my successes, and lifting me up during challenging times. Your friendship has been a cornerstone of my growth and happiness.

I extend my heartfelt appreciation to my remarkable seniors. Your influence and kindness enriched my childhood, adding vibrancy and warmth to my life. Your guidance and example have inspired me to strive for excellence and to be a better person.

To everyone who has been a part of this journey, I am forever grateful. Your contributions, no matter how big or small, have made a significant impact on my life. Thank you for helping me climb the success ladder and for being a part of my story.

ACKNOWLEDGEMENTS

I am deeply thankful to my supervisor, Noor Azilah Binti Draman @ Muda, for her invaluable assistance and guidance throughout this project. Her insightful feedback, constant encouragement, and dedicated support have been instrumental in the successful completion of this project. Despite her numerous responsibilities, she always found time to provide me with direction and motivation, for which I am immensely grateful.

I would also like to express my sincere gratitude to Dr. Norashikin Binti Ahmad, my evaluator, for her meticulous evaluation of my work and for offering valuable recommendations. Her thoughtful critiques and suggestions have significantly improved the quality of my project. Her expertise and attention to detail have been incredibly beneficial, and I am deeply appreciative of her contributions.

With great pleasure, I express my heartfelt gratitude to my family for their unwavering support and encouragement throughout my studies. Their belief in me has been a constant source of strength and motivation. My parents' unconditional love and sacrifices, along with my siblings' constant encouragement, have provided me with the confidence and determination to persevere and succeed. Their emotional and financial support has been a cornerstone of my academic journey.

Lastly, I extend my profound thanks to my lecturers, seniors, and all those who provided me with guidance that was both theoretically enriching and practically beneficial to my studies. The knowledge and skills I have gained from their teachings have been invaluable. Your collective support and encouragement have played a crucial role in my development, and I am deeply appreciative of all that you have done for me. Thank you.

ABSTRACT

The development of the Final Year Project (FYP) Appointment Booking System represents a significant leap forward in educational technology. This report addresses the challenges inherent in the FYP period, particularly the inefficiencies of traditional appointment scheduling methods like emails and in-person requests, which often lead to scheduling conflicts and missed meetings. To address these challenges, the FYP Appointment Booking System was created as a user-friendly, web-based platform. Its primary goal is to simplify the appointment scheduling process for students and supervisors. By digitizing scheduling, the system aims to streamline procedures, reduce administrative burdens, and improve communications with supervisor. Key features of the system include maintaining comprehensive records of appointments, such as the date, time slot, and appointment title, ensuring both parties stay organized. Additionally, it offers a thesis submission feature for students seeking their supervisor’s approval of their documents. Students can also search for lecturer information before requesting them as supervisors. The system provides insights into student-supervisor interactions, supporting institutions in decision-making and resource allocation. In summary, the FYP Appointment Booking System revolutionizes appointment scheduling, enhancing efficiency and collaboration in the FYP process. It ensures easy access for students and facilitates a smooth transition from project initiation to completion, benefiting both students and supervisors involved.

ABSTRAK

Pembangunan Sistem Tempahan Temu Janji Projek Tahun Akhir (FYP) mewakili satu lonjakan besar dalam teknologi pendidikan. Laporan ini menangani cabaran yang wujud dalam tempoh FYP, terutamanya ketidakcekapan kaedah penjadualan temu janji tradisional seperti e-mel dan permintaan secara bersemuka, yang sering menyebabkan konflik jadual dan pertemuan yang terlepas. Untuk mengatasi cabaran ini, Sistem Tempahan Temu Janji FYP telah diwujudkan sebagai platform berasaskan web yang mesra pengguna. Matlamat utamanya adalah untuk memudahkan proses penjadualan temu janji bagi pelajar dan penyelia. Dengan mendigitalkan penjadualan, sistem ini bertujuan untuk menstrimkan prosedur, mengurangkan beban pentadbiran, dan meningkatkan komunikasi dengan penyelia. Ciri-ciri utama sistem ini termasuk mengekalkan rekod temu janji yang komprehensif, seperti tarikh, slot masa, dan tajuk temu janji, memastikan kedua-dua pihak kekal teratur. Selain itu, ia menawarkan ciri penghantaran tesis untuk pelajar yang memerlukan kelulusan dokumen daripada penyelia mereka. Pelajar juga boleh mencari maklumat pensyarah sebelum meminta mereka sebagai penyelia. Sistem ini menyediakan pandangan tentang interaksi pelajar-penyelia, menyokong institusi dalam membuat keputusan dan peruntukan sumber. Secara ringkas, Sistem Tempahan Temu Janji FYP merevolusikan penjadualan temu janji, meningkatkan kecekapan dan kerjasama dalam proses FYP. Ia memastikan akses mudah untuk pelajar dan memudahkan peralihan yang lancar dari permulaan projek hingga selesai, memberi manfaat kepada kedua-dua pelajar dan penyelia yang terlibat.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
DBLC		**Database Life Cycle**
SDLC		**System Development Life Cycle**
DFD		**Data Flow Diagram**
ERD		**Entity Relationship Diagram**
UAT		**User Acceptance Testing**
PHP		**Hypertext Preprocessor**
HTML		**Hypertext Markup Language**
CSS		**Cascading Style Sheets**
GUI		**Graphical User Interface**
FK		**Foreign Key**
PK		**Primary Key**
SQL		**Structured Query Language**
RAM		**Random Access Memory**
DOCX		**Document Extended**
PDF		**Portable Document Format**

CHAPTER 1: INTRODUCTION

1.1 Introduction

The FYP Appointment Booking System is designed to address the shortcomings of current educational platforms in managing Final Year Projects (FYP). Recognizing that existing systems fail to meet the unique needs of FYP students, the proposed system aims to provide a comprehensive solution tailored to their requirements. Key deficiencies in the current system include the absence of a search supervisor function, lack of information on supervisor availability, and the inability to book appointments through the website. These gaps hinder students' ability to meet deadlines, find suitable mentors, and to communicate effectively. To overcome these issues, the new system will feature a search supervisor function, display supervisor capacities, and offer an appointment booking date and time slot. The goal is to create a robust, integrated platform that facilitates timely progress, transparent supervision, and efficient communication, ultimately enhancing the FYP experience for students.

1.2 Problem statement(s)

To fully understand the necessary enhancements for the system, it is crucial to identify and address the shortcomings of the existing framework. Firstly, the current system lacks a search function that allows students to gather information about potential supervisors. This issue stems from the previous developer's omission of additional features that would ease the burden on students in finding a supervisor.

Additionally, students struggle to determine the availability of supervisors because the current system does not provide information about supervisors' current capacities. This lack of transparency makes it difficult for students to identify available supervisors and initiate discussions or submit proposals for their FYP development. The problem arises from the need to manually ask supervisors about their capacity, which is particularly challenging for students in the early stages of the FYP process.

Finally, the lack of appointment booking functionality in the current system presents additional challenges for students trying to schedule meetings with their supervisors. Relying on email communication for scheduling appointments creates a cumbersome and time-consuming process. Students often have to wait days, or even weeks, for a response from their lecturers. The problem stems from the dependence on traditional communication methods, such as emails or in-person requests, which are not integrated into the system. This reliance leads to an inefficient and time-consuming procedure for students.

Considering these identified shortcomings, the imperative for a comprehensive FYP management system becomes apparent. By addressing these deficiencies and integrating essential functionalities such as search functions, supervisor availability indicators, and appointment booking capabilities, the envisioned system seeks to revolutionize the management of final year projects within educational institutions. Through enhanced efficiency, transparency, and accessibility, the proposed system aims to facilitate a smoother and more productive FYP experience for both students and supervisors alike.

1.3 Objectives

- To ensure the availability of search supervisor feature within the system that allows students to easily find and gather information about potential supervisors.
- To enable students to access information regarding the availability of supervisors' student capacity.
- To facilitate students in scheduling appointments with their supervisors via the system.

1.4 Scope

The scope of the FYP Appointment Booking System can be divided into target user and modules.

1.4.1 Modules to be developed

i. Login Module
- This module can detect user roles and log them in accordingly based on their role.

ii. Document Submission Module
- Students are able to submit their finished document for supervisor's approval.

iii. Appointment Booking Module
- Designed for students to book an appointment with their supervisor using the system

iv. Booking Supervisor Module
- This module will enable FYP students to request a supervisor.

v. Report Module
- The report will be produced in both list and graph formats to display the data statistics in the report module.

1.4.2 Scopes of the User

i. Admin
- Able to log in as administrator
- Able to manage supervisor's information
- Able to view all reports
- Able to search supervisor's information
- Able to add department and position into the system

ii. Supervisor

- Able to log in as supervisor
- Able to manage appointments requested by FYP students
- Able to change password
- Able to manage documentation

iii. FYP Student

- Able to log in as student
- Able to change password
- Able to add documentation
- Able to request appointments with their supervisor
- Able to search supervisor’s information

1.5 Project Significance

The project aims to ease students' burdens by streamlining supervisor requests through the system. Moreover, students can conveniently schedule appointments with their supervisors by choosing from available dates and time slot, eliminating the necessity for communication via WhatsApp. This approach mitigates the risk of delayed responses, which may occur if supervisors overlook messages from students. Consequently, students will find it more straightforward to manage their tasks, knowing they have a structured process in place. Lastly, students will get to have a search function to gather information about supervisors.

1.6 Expected Outcome

The system will provide search options in the system to ensure students receive well rounded information before choosing their supervisor. Additionally, students will be able to view the availability of supervisors by checking their capacity, helping them assess which supervisors are available for project guidance. Finally, the system will offer a simple and efficient way for students to schedule appointments with their supervisors.

1.7 Conclusion

In conclusion, this chapter presents an overview of the FYP Appointment Booking System, a platform designed for students to directly schedule appointments. It highlights the limitations of the existing FYP system, which prioritizes documentation submission over appointment scheduling. Additionally, this chapter details the project's scope, impact, and anticipated outcomes. The next chapter will discuss the project planning and methodologies.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

This chapter will cover the details of the development methodology used in the system. There are several types of System Development Life Cycle (SDLC) methodology such as the agile model, spiral modal and waterfall mode. These methodologies can design, develop and evaluate the project according to the principles and criteria of the project. Database Life Cycle (DBLC) was chosen for the project development because DBLC defines the database initial study, database design, implementation and loading, training and evaluation, operations, maintenance as well as evaluation. The DBLC also includes database monitoring, modification and maintenance which means the cycle never ends. These activities will continue even after the database has been successfully implemented.

2.2 Project Methodology

The FYP Appointment Booking System adheres to the stages of the Database Life Cycle (DBLC) for implementing a database, ranging from requirements analysis to maintenance. The DBLC comprises six phases: initial database study, database design, implementation, testing and evaluation, operation, and maintenance.

Database Life Cycle (DBLC) Phases for FYP Appointment Booking System:

1. Database Initial Study:

In the initial study phase, the primary objective is to understand the requirements and scope of the FYP Appointment Booking System. This involves conducting feasibility studies to determine the technical, operational, and economic viability of the project. The tasks include meeting with stakeholders such as students, supervisors, and administrative staff to gather detailed requirements. Surveys and interviews will be conducted to understand their needs and expectations from the system. The goal is to identify the key functionalities required, such as appointment scheduling, supervisor capacity visibility, and notification features. This phase will result in a clear project scope, defined goals, and a detailed requirements document that will guide the subsequent phases.

2. Database Design:

The design phase involves creating a comprehensive blueprint of the database structure. The tasks include developing data models, such as entity-relationship diagrams (ERDs), to represent the data entities, their attributes, and the relationships between them. For the FYP Appointment Booking System, this will involve defining tables for users (students and supervisors), appointments, notifications, and system settings. Schema definitions will be created to ensure data integrity, consistency, and normalization. This phase also includes designing the user interface for ease of use. Prototyping tools will be used to create wireframes and mock-ups for visualizing the user experience. The design documents produced in this phase will serve as the blueprint for the database implementation.

3. Implementation and Loading:

In the implementation and loading phase, the actual construction of the database begins based on the design specifications. Tasks include setting up the database environment using a suitable DBMS (Database Management System), creating database tables, and defining primary and foreign keys to establish relationships between tables. For the FYP Appointment Booking System, initial data such as user accounts and supervisor profiles will be loaded into the database. This phase also involves writing and testing SQL scripts for data manipulation and querying. Additionally, any required backend logic, such as stored procedures and triggers, will be implemented. Regular progress reviews and testing will ensure that the database is built correctly and efficiently.

4. Testing and Evaluation:

The testing and evaluation phase is critical to ensure the database meets all functional and performance requirements. Tasks include conducting unit testing to verify individual components, integration testing to ensure different parts of the system work together seamlessly, and user acceptance testing (UAT) with actual users to validate the system's functionality and usability. For the FYP Appointment Booking System, testing scenarios will include scheduling appointments, sending notifications, and handling concurrent user access. Any bugs or issues identified during testing will be documented and resolved. Performance testing will also be conducted to ensure the system can handle the expected load. The goal is to deliver a stable and reliable database system that meets all specified requirements.

5. Operation:

During the operation phase, the FYP Appointment Booking System is deployed for actual use by students and supervisors. Tasks include setting up the production environment, migrating data from any existing systems, and configuring the system for optimal performance and security. User training sessions will be conducted to ensure all stakeholders are familiar with the system's features and functionality. This phase also involves monitoring the system to identify any initial issues or user feedback. Support mechanisms will be established to help and address any problems that arise during the early stages of operation. The focus is on ensuring a smooth transition to the new system and achieving user satisfaction.

6. Maintenance and Evolution:

The maintenance and evolution phase involves ongoing support and enhancements to the database system. Tasks include regular monitoring of the system’s performance, applying updates and patches to fix bugs and improve security, and making enhancements based on user feedback and evolving requirements. For the FYP Appointment Booking System, this may involve adding new features, optimizing existing functionalities, and scaling the system to handle increased usage. Regular data integrity checks will be performed to ensure the system remains reliable and secure. The goal is to ensure the system continues to meet the needs of its users and adapt to any changes in the academic environment.

2.3 Project Schedule and Milestones

In Table 2.1, it shows the milestones of development of FYP Appointment Booking System. In each of the milestones there are documents that are expected to be produced as deliverables to increase transparency and accountability in each of the development processes. This project milestone also acts as a timeline for the development process to ensure all processes are delivered within the expected timeline.

Table 2.1: Project Milestone

Milestones	**Expected Documents**	**Start Date**	**End Date**
Requirement analysis	a) Collecting comprehensive information about the existing FYP system. b) Presenting the concept of the new system to the supervisor to secure approval for the project title. c) Drafting and submitting the project proposal to the supervisor for review.	11 March 2024	15 March 2024
Design of the proposed system	a) Sketching Context Diagram and Data Flow Diagram of the proposed system b) Building a complete Entity Relation Diagram (ERD)	15 March 2024	29 March 2024
Implementation of system	a) Initiating the process of system development.	29 March 2024	7 June 2024
System testing	a) Testing the system's functionality, performing validation and verification, and	7 June 2024	17 June 2024

	reviewing the system's results.		
Project demo	a) Showcase a demonstration of the system to the supervisor. b) Incorporate minor enhancements to improve the system's functionality.	19 June 2024	19 June 2024
PSM report	a) Submit final report	20 June 2024	20 June 2024
Final presentation	a) Present the fully operational system to the supervisor and evaluator, followed by a Q&A session regarding the system	20 June 2024	20 June 2024

Table 2.2: Gantt Chart of FYP Appointment Booking System

Task / Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Requirement analysis	■														
Design of the proposed system	■	■	■												
Implementation of system			■	■	■	■	■	■	■	■	■	■	■		
System testing													■	■	■
Project demo															■
PSM report															■
Final presentation															■

2.4 Conclusion

To conclude this section, this chapter covers the database development methodology for the project, known as the Database Life Cycle (DBLC). The six phases which are database initial study, database design, implementation, testing and evaluation, operation, and maintenance, have been detailed in this chapter. These phases are discussed in relation to the implementation of the FYP Appointment Booking System, along with an overview of the project schedule and milestones.

CHAPTER 3: ANALYSIS

3.1 Introduction

This chapter will explore the detailed process of collecting and analyzing data for the FYP Appointment Booking System. The objective is to ensure the system can effectively address the previously identified issues and confirm that the system requirements are appropriate and feasible.

To meet these objectives, a thorough analysis of the collected data will be performed. This analysis will involve creating various visual aids and diagrams to clarify the data and system workflows. Specifically, it will include the development of a detailed flow chart to outline the processes, a context diagram to define the system's scope and boundaries, and a data flow diagram (DFD) to show how data moves through the system.

These tools will help identify any potential gaps or inefficiencies within the system, ensuring that all aspects of the requirements are carefully examined and addressed. By employing these methodologies, the analysis will provide a solid framework to ensure that the FYP Appointment Booking System is effectively designed to achieve its intended goals.

3.2 Problem Analysis

An unstructured interview with previous FYP students and an observation were conducted to understand and analyze the weaknesses of the current FYP Appointment Booking System. The existing system requires students to contact lecturers via email to inquire about their availability for supervision, without knowing if the lecturers' quotas are already full. Additionally, tracking previous appointments is challenging. As appointments are arranged through email or WhatsApp, both students and supervisors receive numerous messages daily, making it difficult to respond promptly and causing delays in meetings. Lastly, the system lacks a search function for students to identify which lecturers are available to supervise in a particular semester. This search functionality is essential, as lecturers' information is confidential and should not be freely shared.

3.3 The Proposed Improvement / Solutions

The proposed system is to create a system that helps ease the student's problem especially during the early stages of doing the FYP. This web-based system would be an efficient tool for the students to mainly make an appointment with their designated supervisor via a website which eliminates the need for traditional methods.

3.4 Requirement Analysis of the to-be System

Requirements Analysis aims to outline the expectations of users for the developed website system. It seeks to enhance the efficiency of booking appointments between students and supervisors, display each supervisor's quota, and provide searchable options to easily find lecturer information within the system.

3.4.1 Functional Requirements

The functional requirement of a system discusses the function of the system. A function is the behavior of expected inputs and desired outputs. Functional requirement shapes the design of a system. The functional requirement of this system includes employees and administrators.

1. Admin
    - Access supervisor information
    - Review all student appointments (both rejected and accepted)
    - Monitor document submissions from all FYP students
    - Oversee supervisor registration in the system
    - Handle the management of positions available for lecturers
    - Handle the management of departments available for lecturers

2. Supervisor
   - Manage students request for supervision
   - Oversees document submissions from students under their supervision
   - Handle appointment requests for meetings

3. FYP student
   - Request a supervisor
   - Search for supervisor information
   - Schedule appointments with their designated supervisor
   - Manage document submission under supervisor's guidance

3.4.1.1 Context Diagram

A context diagram is a top level also known as "Level 0" data flow diagram. It shows how the system will receive and send data to the external entities involved.

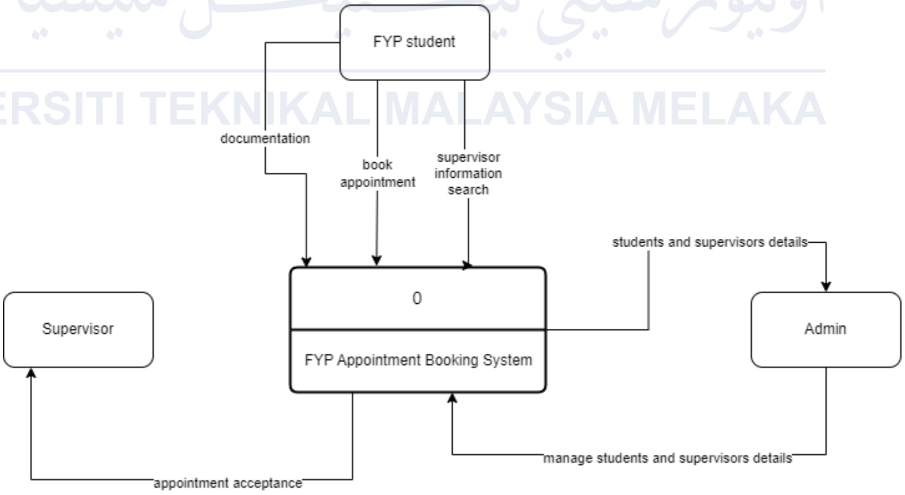

FYP student
documentation
book appointment
supervisor information search
students and supervisors details
0
FYP Appointment Booking System
Supervisor
Admin
appointment acceptance
manage students and supervisors details

Figure 3.1: Context diagram of FYP Appointment Booking System

Figure 3.1 shows the context diagram of the FYP Appointment Booking System.

3.4.1.2 Data Flow Diagram

Data Flow Diagram (DFD) is a graphical method to illustrate the processes or activities that are performed and how data moves between each function. In this project, a context diagram is first to be built to show the external entities involved and the data flows that start or end in them. Afterwards, a DFD fragment is created based on the project requirement to show how the data is flowing through every process, external entities and data stores. Figure 3.2 – Figure 3.3 show the diagrams of FYP Appointment Booking System.

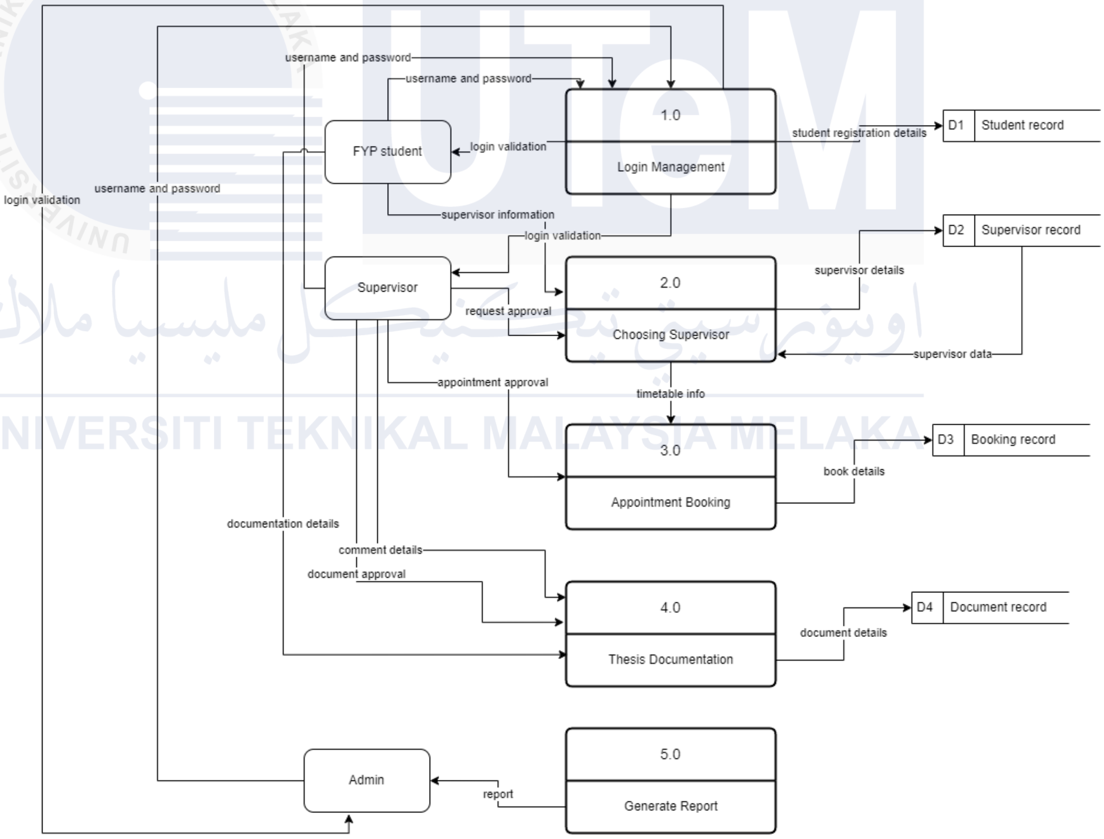


Figure 3.2: Level 0 of FYP Appointment Booking System

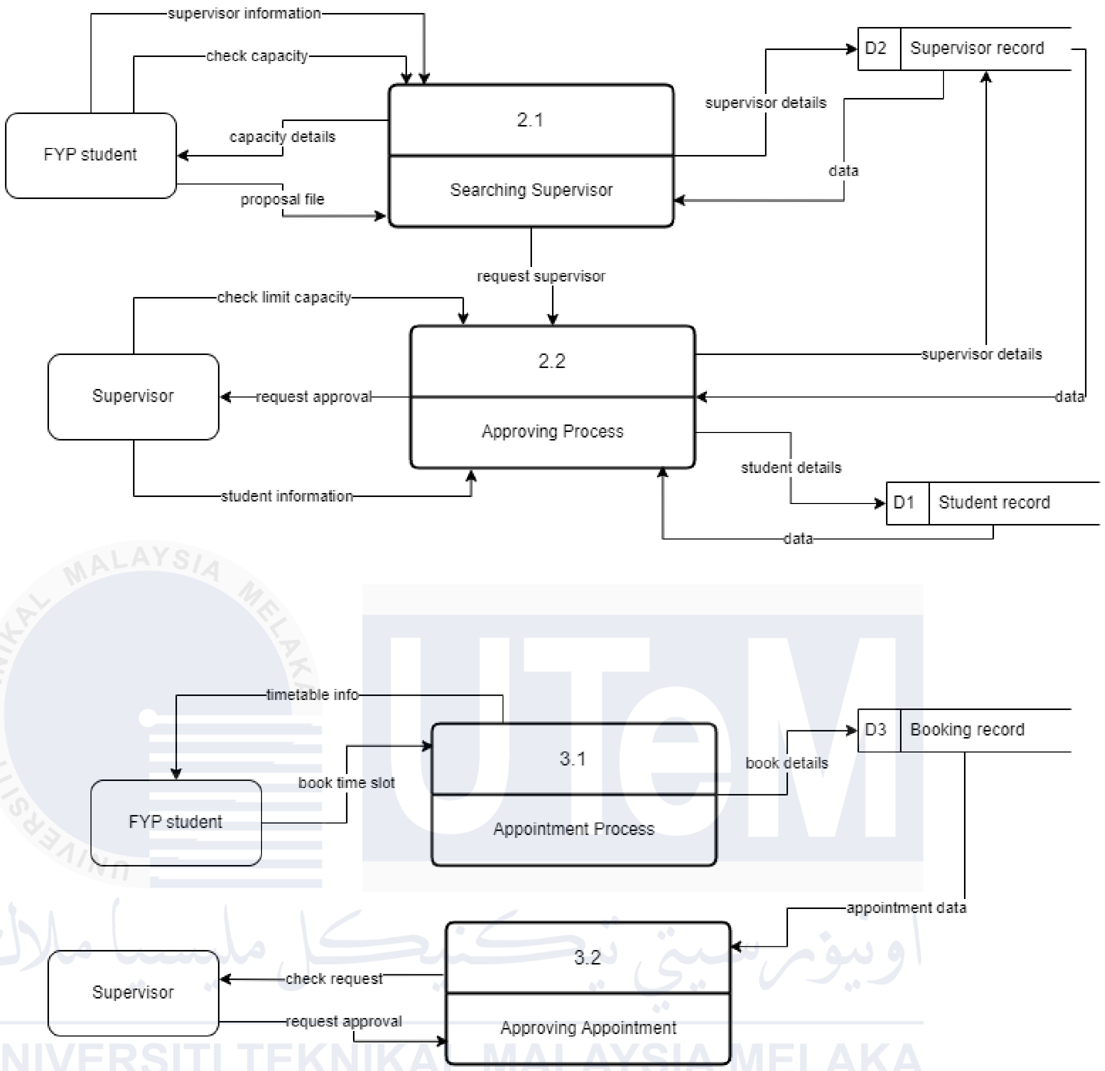


Figure 3.3: Level 1 of FYP Appointment Booking System

3.4.2 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 3.1 lists out all the non-functional requirements and its descriptions for the project.

Table 3.1: Non-Functional Requirement

Type	Requirement	Description
Coding Standards	System coding	The system is developed by using PHP, JAVA, HTML, CSS and MySQL
Integrity	Data integrity	Data should always be consistent throughout the whole system.
Security	Data Security	The system will limit the function provided for users based on their roles (Admin, Supervisor, FYP student)
Usability	Portability	The system should perform well in a browser when developed and tested using Visual Studio Code and displayed on a website page.
Reusability	Reusability	The system has user authentication module that is designed to be reusable, allowing FYP students, admin and supervisors to log in using the same underlying authentication page.

3.4.3 Other Requirements

The requirements of database system development can be split into two categories, which include the hardware requirements and software requirements. Software requirement defines the software that is used to develop the system while the hardware requirement describes the hardware used to run the software defined to build the system.

3.4.3.1 Software Requirements

Table 3.2: Software Component List in FYP Appointment Booking System

No	Software Component Name	Function
1	MySQL	Used to store and retrieve data for the system.
2	XAMPP	Used to host websites and web applications locally.
3	Draw.io	Illustrate technical diagrams such as ERD, Context Diagram, DFD and Flowchart.
4	Microsoft Word 365	To create the report documents for this report.
5	Visual Studio Code	Allows users to create code using programming languages such as PHP, HTML, CSS and JavaScript.
6	Canva	Used for creating slide presentation for the final project presentation with evaluator and supervisor.

3.4.3.2 Hardware Requirements

The list of hardware components that will be used in the FYP Appointment Booking System as shown in Table 3.3.

Table 3.3: Hardware Component List in FYP Appointment Booking System

No	Hardware Component Name	Function
1	HP Pavilion Laptop Intel 13th i5	To develop website on the project.
2	HP Printer	To print project documents.

3.5 Conclusion

The context diagram presents the FYP Appointment Booking System as a single high-level process, illustrating its relationships with other entities. In contrast, the Data Flow Diagram (DFD) details the flow of data within each function for different users, including students, supervisors, and admin. This development of data analysis simplifies the structure and meaning of data within the FYP Appointment Booking System. By employing data analysis techniques, we can transform the complexities of the appointment booking process into a comprehensible model. This model can be implemented on a computer and accessed by users, providing a clear depiction of interactions and data management within the system.

CHAPTER 4: DESIGN

4.1 Introduction

Designing the FYP Appointment Booking System is one of the most important phases in system development because the output from this phase will affect all subsequent phases. The logical system design is the result of system analysis and will be developed into the physical system design, while the physical design details the created system to solve the stated problems. During this phase, databases, schemas, processing specifications, and input/output requirements should be defined. This stage is also crucial because the data structure, control processes, procedures, and interfaces will be determined.

There are many techniques used to design the FYP Appointment Booking System. 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) will all be employed to design the system.

4.2 System Architecture Design

In this system, the architecture used is a web-based system. The web-based system approach is chosen as the architectural view in this project because it ensures that students and supervisors can access the FYP Appointment Booking System. Additionally, the platform is highly flexible and can effectively facilitate communication and scheduling between students and supervisors. However, it is important to note that the system still requires downloading and installation on the device; it cannot function as a standalone web page. This ensures that all necessary components and dependencies are available for the system to run smoothly, providing a seamless user experience.

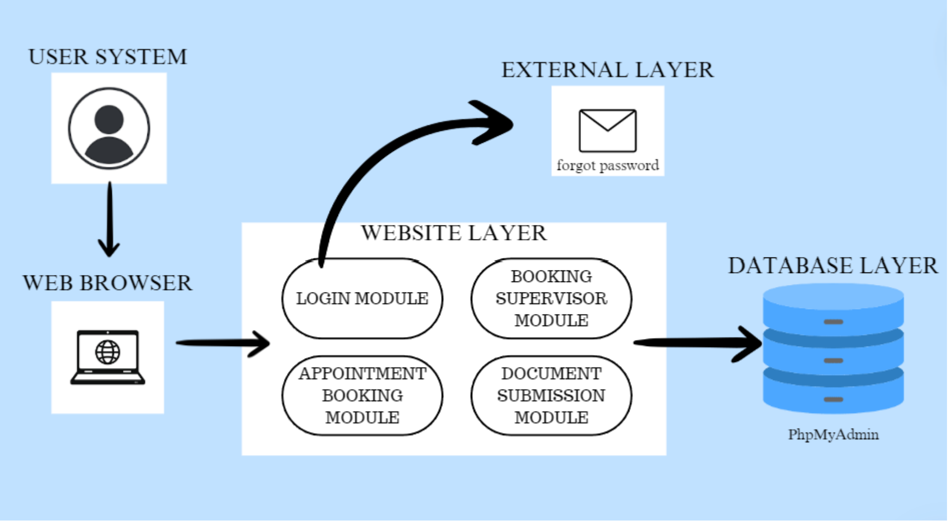


Figure 4.1: System Architecture for FYP Appointment Booking System

4.2.1 Conceptual Design

Figure 4.2 is the Entity Relationship Diagram (ERD) of FYP Appointment Booking System.

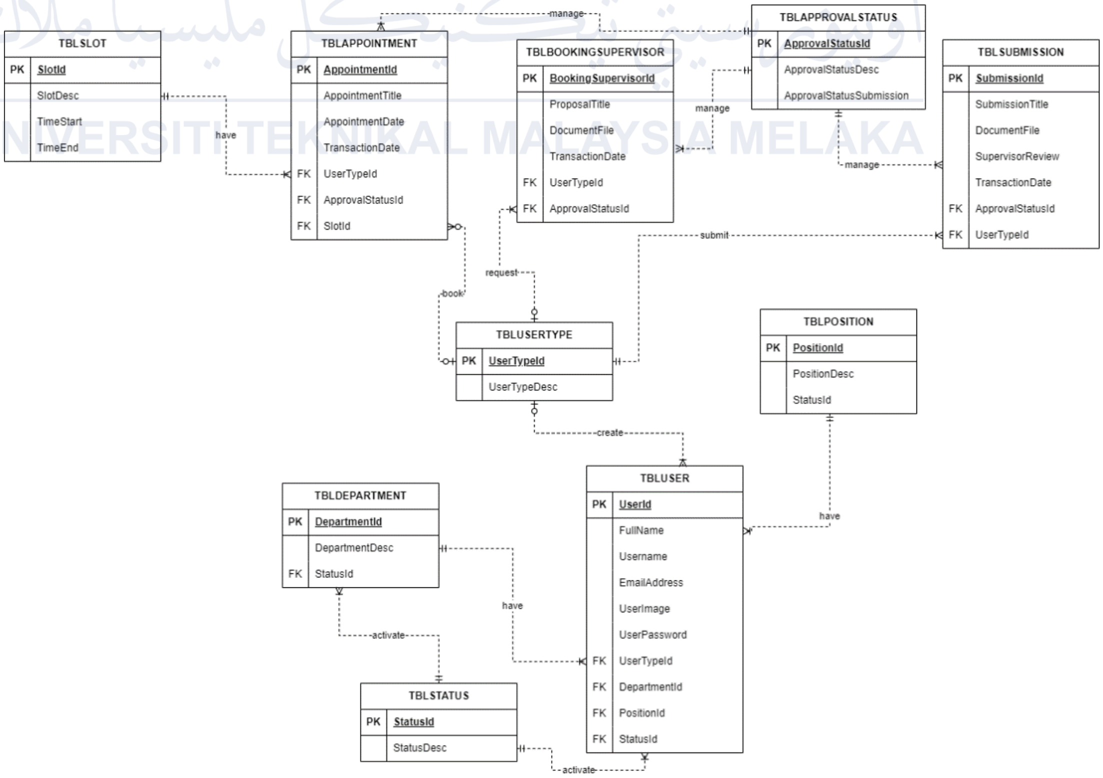


Figure 4.2: ERD for FYP Appointment Booking System

4.2.1.1 Business Rules

(a) Each student and supervisor must have one account to use.

(b) Each student can only book appointments with their designated supervisor.

(c) A student can have multiple appointments with the same supervisor.

(d) The supervisor can approve or reject appointments requested by students.

(e) Each student can only have one supervisor at a time.

(f) A student can send one to many document submissions at a time.

(g) Each supervisor can have one and only department.

(h) Each supervisor can have one and only position.

4.2.2 Logical Design

Table 4.1 – 4.10 shows the data dictionary of FYP Appointment Booking System that contains attributes of each table and brief explanation on the data type, data constraint, reference table, the attribute's description.

Table 4.1: Table tblappointment

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
AppointmentId	Id for each appointment	Int (11)	N/A	YES		PK	
AppointmentTitle	Title of appointment	Varchar (1000)	N/A	YES			
AppointmentDate	Date of appointment	date	N/A	YES			

TransactionDate	Time of appointment	timestamp	N/A	YES			
SlotId	Timeslot of appointment	Int (11)	N/A	YES		FK	tblslot
ApprovalStatusId	Approved or rejected	Int (11)	N/A	YES		FK	tblapprovestatus
UserTypeId	Student, lecturer or admin	Int (11)	N/A	YES		FK	tblusertype

Table 4.2: Table tblapprovalstatus

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
ApprovalStatusId	Id of approval status	Int (11)	N/A	YES		PK	
ApprovalStatusDesc	Description of approval status	Varchar (255)	N/A	YES			
ApprovalStatusSubmission	Submission of approval status	Varchar (255)	N/A	YES			

Table 4.3: Table tblbookingsupervisor

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
BookingSupervisorId	Id booking for supervisor	Int (11)	N/A	YES		PK	
ProposalTitle	Proposal title	Varchar (1000)	N/A	YES			
DocumentFile	Document file	Varchar(255)	N/A	YES			
TransactionDate	Time of booking	Timestamp	N/A	YES			
UserTypeId	Student, supervisor or admin	Int (11)	N/A	YES		FK	tblusertype
ApprovalStatusId	Approval status id	Int (11)	N/A	YES		FK	tblapprovestatus

Table 4.4: Table tbldepartment

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
DepartmentId	Department id	Int (11)	N/A	YES		PK	
DepartmentDesc	Department description	Varchar (255)	N/A	YES			
StatusId	Status id	Int (11)	N/A	YES		FK	tblstatus

Table 4.5: Table tblposition

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
PositionId	position id	Int (11)	N/A	YES		PK	
PositionDesc	position description	Varchar (255)	N/A	YES			
StatusId	Status id	Int (11)	N/A	YES		FK	tblstatus

Table 4.6: Table tblslot

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
SlotId	slot id	Int (11)	N/A	YES		PK	
SlotDesc	Soft description	Varchar (50)	N/A	YES			
TimeStart	Time start	time	N/A	YES			
TimeEnd	Time end	time	N/A	YES			

Table 4.7: Table tblstatus

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
StatusId	status id	Int (11)	N/A	YES		PK	
StatusDesc	status description	Varchar (10)	N/A	YES			

Table 4.8: Table tblsubmission

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
SubmissionId	Id for each submission	Int (11)	N/A	YES		PK	
SubmissionTitle	Title of submission	Varchar (1000)	N/A	YES			
DocumentFile	Document file	Varchar (255)	N/A	YES			
SupervisorReview	Supervisor review	Int (11)	N/A	YES			
TransactionDate	Date of transaction	timestamp	N/A	YES			
ApprovalStatusId	Approved or rejected	Int (11)	N/A	YES		FK	tblapprovestatus
UserTypeId	Student, supervisor or admin	Int (11)	N/A	YES		FK	tblusertype

Table 4.9: Table tbluser

Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
UserId	Id for each user	Int (11)	N/A	YES	YES	PK	
FullName	Full name	Varchar (255)	N/A	YES	YES		
Username	Username	Varchar (30)	N/A	YES			
EmailAddress	Email address	Varchar (100)	N/A	YES	YES		
UserImage	User image	Varchar (255)	N/A	YES			
UserPassword	User Password	Varchar (255)	N/A	YES			
UserTypeId	Student, lecturer or admin	Int (11)	N/A	YES		FK	tblusertype
DepartmentId	Department id	Int (11)	N/A	YES		FK	tbldepartment

PositionId	Position id	Int (11)	N/A	YES		FK	tblposition
StatusId	Status id	Int (11)	N/A	YES		FK	tblstatus

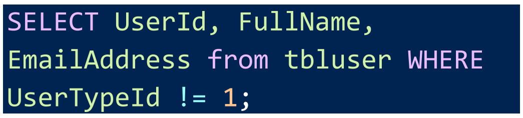
Table 4.10: Table tblusertype

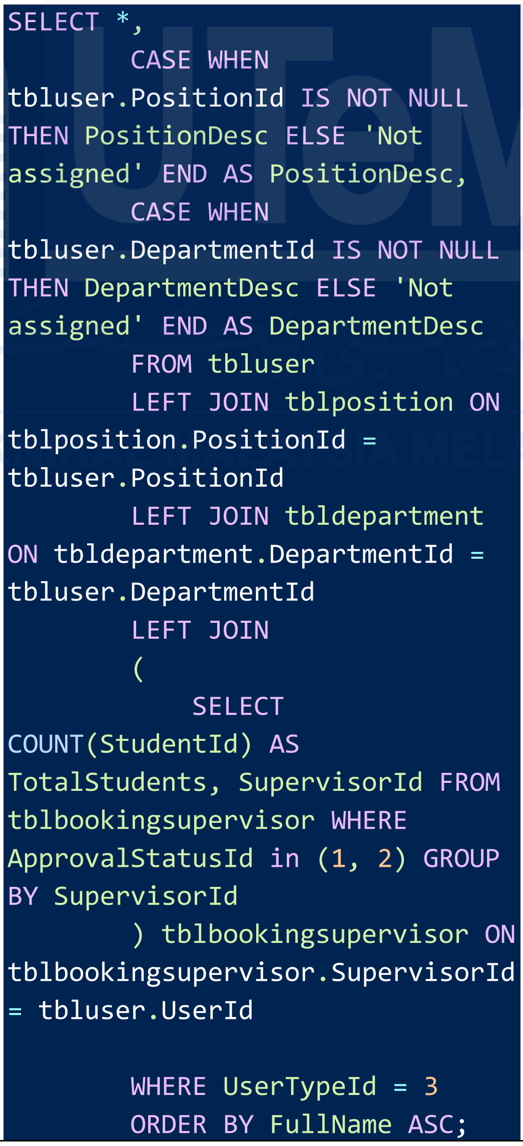
Attribute	Description	Data Type	Format	Required	Unique	PK/FK	FK References
UserTypeId	Student, lecturer or admin	Int (11)	N/A	YES		PK	
UserTypeDesc	User type description	Varchar (20)	N/A	YES			

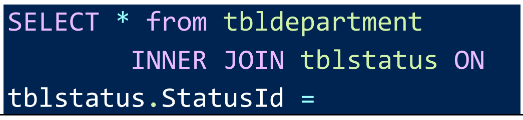
4.2.2.1 Query Design

There are many queries that have been designed to produce different types of outputs. Table 5.1 shows a few examples of query design

Table 4.11: Query Design in FYP Appointment Booking System

NO	TYPE OF QUERY	EXAMPLE	DESCRIPTION
1	SELECT	SELECT UserId, FullName, EmailAddress from tbluser WHERE UserTypeId != 1;	Retrieves information about users (UserId, FullName,

			EmailAddress) from the tbluser table. It excludes users whose UserTypeId is 1. This implies that only users with a UserTypeId other than 1 will be included in the result set.
2	LEFT JOIN	 SELECT *, CASE WHEN tbluser.PositionId IS NOT NULL THEN PositionDesc ELSE 'Not assigned' END AS PositionDesc, CASE WHEN tbluser.DepartmentId IS NOT NULL THEN DepartmentDesc ELSE 'Not assigned' END AS DepartmentDesc FROM tbluser LEFT JOIN tblposition ON tblposition.PositionId = tbluser.PositionId LEFT JOIN tbldepartment ON tbldepartment.DepartmentId = tbluser.DepartmentId LEFT JOIN (SELECT COUNT(StudentId) AS TotalStudents, SupervisorId FROM tblbookingsupervisor WHERE ApprovalStatusId in (1, 2) GROUP BY SupervisorId) tblbookingsupervisor ON tblbookingsupervisor.SupervisorId = tbluser.UserId WHERE UserTypeId = 3 ORDER BY FullName ASC;	Retrieves comprehensive information about supervisors (UserTypeId = 3) from tbluser, including their basic details, position and department descriptions (if available), and the count of students they supervise (TotalStudents).

3	INNER JOIN	
		SELECT * from tbldepartment
		        INNER JOIN tblstatus ON
		tblstatus.StatusId =
			Retrieve department details along with

		`tbldepartment.StatusId;`	the corresponding status names from table tblstatus.
4	AGGREGATE (COUNT)	`SELECT COUNT(StudentId) AS TotalStudents, SupervisorId FROM tblbookingsupervisor WHERE ApprovalStatusId in (1, 2) GROUP BY SupervisorId;`	The query counts the number of students (StudentId) assigned to each supervisor (SupervisorId) who have ApprovalStatusId values of 1 or 2 (indicating approval or pending approval). It groups the results by SupervisorId, providing a count (TotalStudents) for each supervisor.

4.2.3 Physical Design

In the physical design phase, the Database Management System (DBMS) has been chosen. The right DBMS is selected to develop the system. MySQL supports the wide range of database operations required by the system to function properly. With the reasons stated above, MySQL is easily the choice to develop the FYP Appointment Booking System.

4.2.4 Usage of Stored Procedures, Stored Functions and Triggers

Stored procedures and stored functions are used to perform tasks in database applications. They can increase performance and productivity while being easy to use too. Stored procedures can reduce redundant commands required to be repeatedly compiled hence making commands more efficient and faster to execute compared to usual commands.

4.2.4.1 Application of Stored Procedures

1. Appointment Booking: This stored procedure handles the booking of appointments by students with their supervisors. It ensures that an appointment slot is only booked if it is available.
2. Search Student List: This query retrieves a list of users who are students from the tbluser table.
3. Search Supervisor List: This query retrieves a list of users who are supervisors from the tbluser table.

4.2.4.2 Application of Triggers

1. Booking Supervisor: The trigger enforces the rule that a supervisor can have a maximum of 5 students at any given time. The trigger automatically performs this check every time a new student is requested, ensuring that the rule is consistently enforced without requiring manual intervention. It provides immediate feedback and prevents invalid data from being inserted, helping maintain the integrity of the booking system.

4.3 Graphical User Interface (GUI) Design

Graphical User Interface (GUI) design is the process of designing the aesthetics and user interaction elements of a system, focusing on visual appeal, functionality, and user experience. The design aims to ensure that users can interact with the software easily and intuitively, minimizing the learning curve and enhancing overall satisfaction.

4.3.1 Login GUI

The login interface allows FYP students, lecturers, and admin to access the system through a single, unified interface. As illustrated in Figure 4.3, the system identifies users' roles based on their registered credentials upon login. For those who have forgotten their passwords, the interface includes a "forgot password" feature. Users can enter their email addresses, and the system will send a temporary password for accessing the system. Additionally, the system provides a registration option exclusively for students, as shown in Figure 4.4.

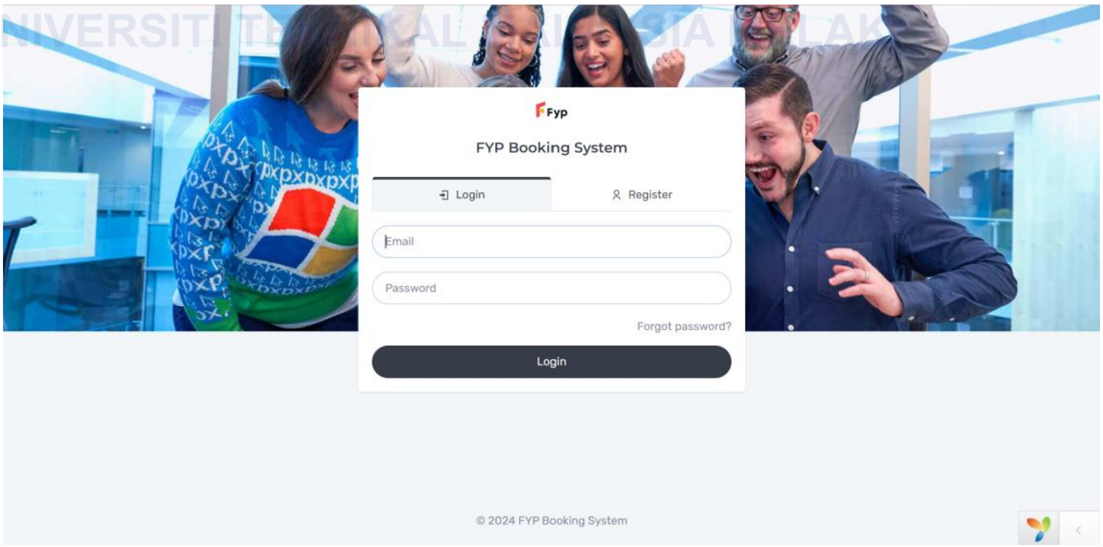
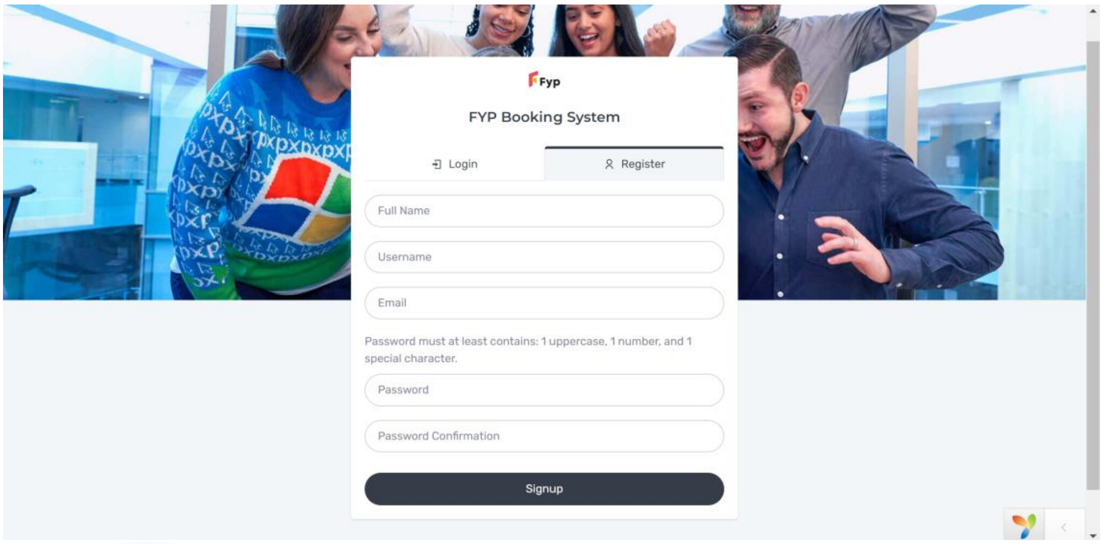


Figure 4.3: Login Interface of FYP Appointment Booking System



Password must at least contains: 1 uppercase, 1 number, and 1 special character.

Figure 4.4: Register Interface of FYP Appointment Booking System

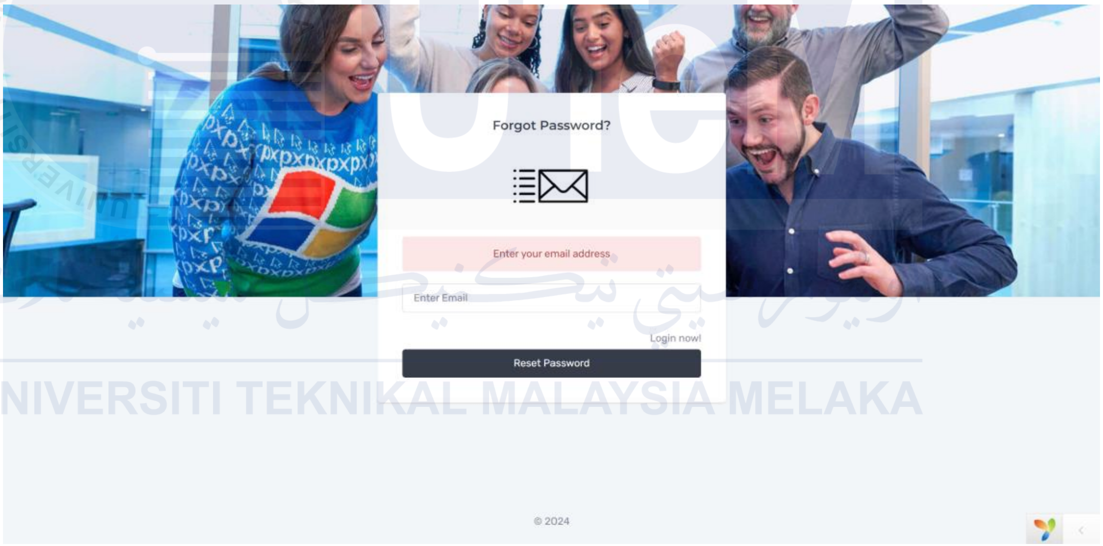
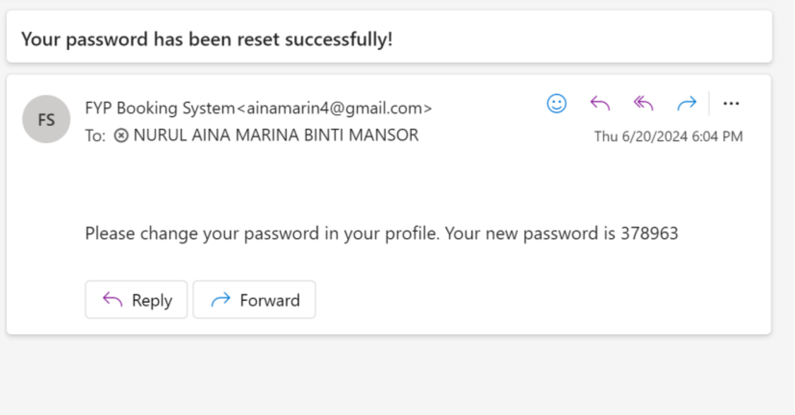


Figure 4.5: Forgot Password Interface



Your password has been reset successfully!

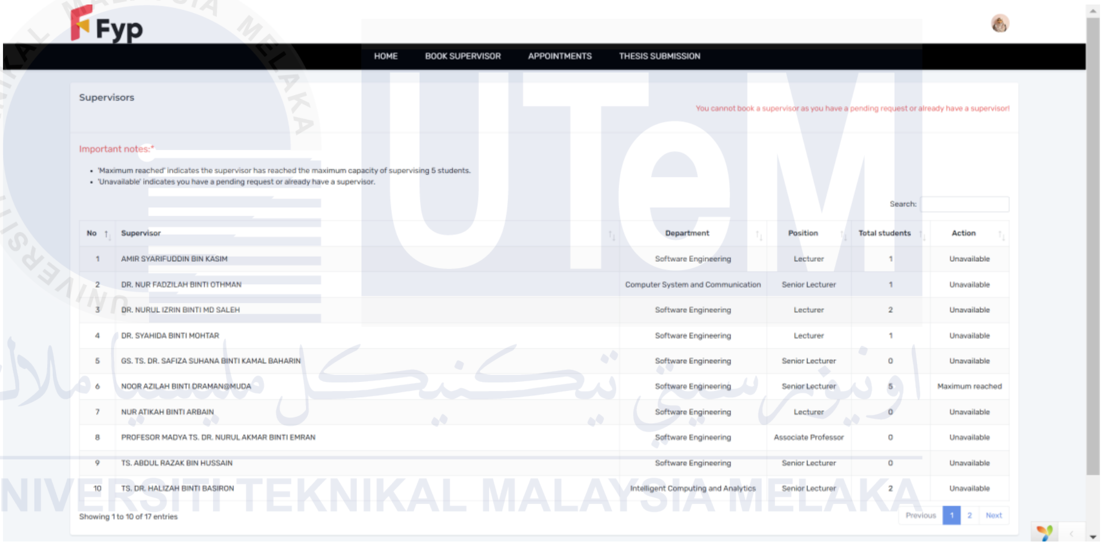
FS
FYP Booking System<ainamarin4@gmail.com>
To: NURUL AINA MARINA BINTI MANSOR
Thu 6/20/2024 6:04 PM

Please change your password in your profile. Your new password is 378963

Figure 4.6: Password Recovery Email Interface

4.3.2 FYP Student GUI

The graphical user interface (GUI) for FYP students is designed to streamline the process of requesting a supervisor, as illustrated in Figure 4.3. Once a supervisor accepts a student, the student can use the appointment booking feature, shown in Figure 4.4. Subsequently, students can submit their completed proposals via the document submission function. The proposals will either be accepted or require further improvements, as depicted in Figure 4.5.



Supervisors

You cannot book a supervisor as you have a pending request or already have a supervisor!

Important notes:*

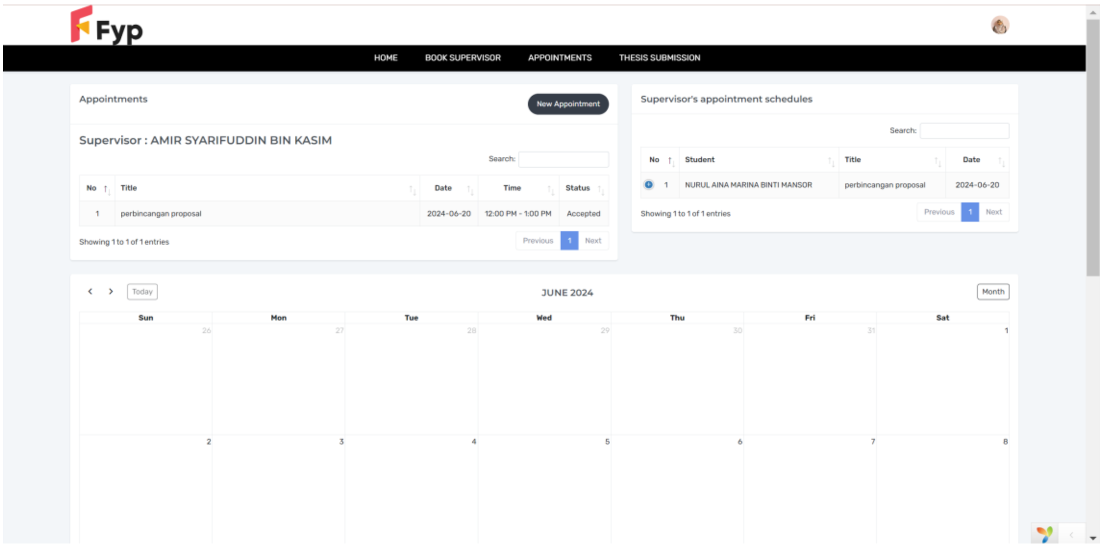
- 'Maximum reached' indicates the supervisor has reached the maximum capacity of supervising 5 students.
- 'Unavailable' indicates you have a pending request or already have a supervisor.

Search:

No	Supervisor	Department	Position	Total students	Action
1	AMIR SYARIFUDDIN BIN KASIM	Software Engineering	Lecturer	1	Unavailable
2	DR. NUR FADZILAH BINTI OTHMAN	Computer System and Communication	Senior Lecturer	1	Unavailable
3	DR. NURUL IZRIN BINTI MD SALEH	Software Engineering	Lecturer	2	Unavailable
4	DR. SYAHIDA BINTI MOHTAR	Software Engineering	Lecturer	1	Unavailable
5	GS. TS. DR. SAFIZA SUHANA BINTI KAMAL BAHARIN	Software Engineering	Senior Lecturer	0	Unavailable
6	NOOR AZILAH BINTI DRAMAN@MUDA	Software Engineering	Senior Lecturer	5	Maximum reached
7	NUR ATIKAH BINTI ARBAIN	Software Engineering	Lecturer	0	Unavailable
8	PROFESOR MADYA TS. DR. NURUL AKMAR BINTI EMRAN	Software Engineering	Associate Professor	0	Unavailable
9	TS. ABDUL RAZAK BIN HUSSAIN	Software Engineering	Senior Lecturer	0	Unavailable
10	TS. DR. HALIZAH BINTI BASIRON	Intelligent Computing and Analytics	Senior Lecturer	2	Unavailable

Showing 1 to 10 of 17 entries

Figure 4.7: Book Supervisor Interface



Appointments

Supervisor : AMIR SYARIFUDDIN BIN KASIM

Search:

No	Title	Date	Time	Status
1	perbincangan proposal	2024-06-20	12:00 PM - 1:00 PM	Accepted

Showing 1 to 1 of 1 entries

Supervisor's appointment schedules

Search:

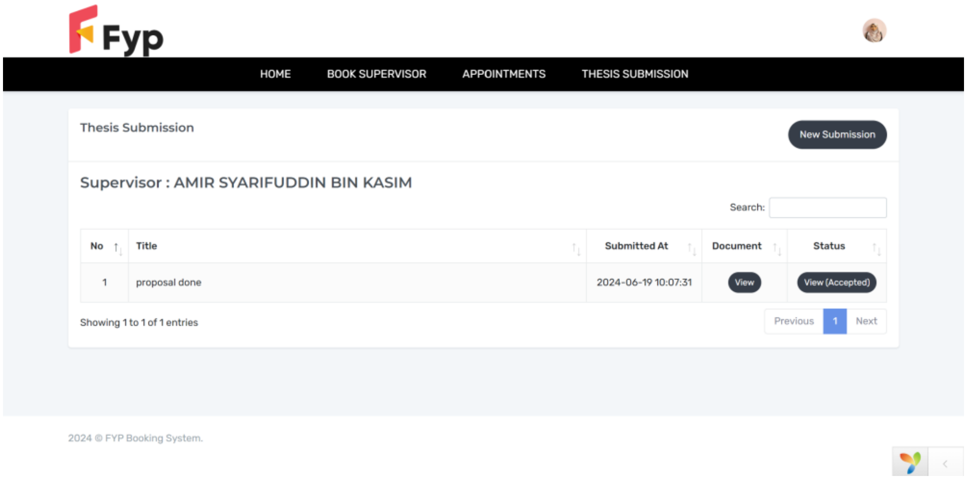
No	Student	Title	Date
1	NURUL AINA MARINA BINTI MANSOR	perbincangan proposal	2024-06-20

Showing 1 to 1 of 1 entries

JUNE 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Figure 4.8: Booking Appointment Interface



Thesis Submission

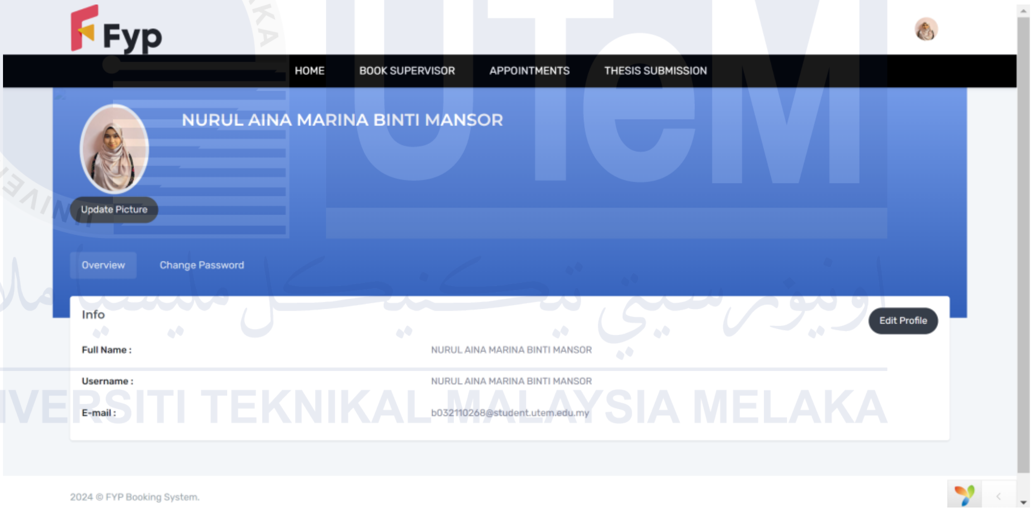
Supervisor : AMIR SYARIFUDDIN BIN KASIM

No	Title	Submitted At	Document	Status
1	proposal done	2024-06-19 10:07:31	View	View (Accepted)

Showing 1 to 1 of 1 entries

2024 © FYP Booking System.

Figure 4.9: Document Submission Interface



NURUL AINA MARINA BINTI MANSOR

Info

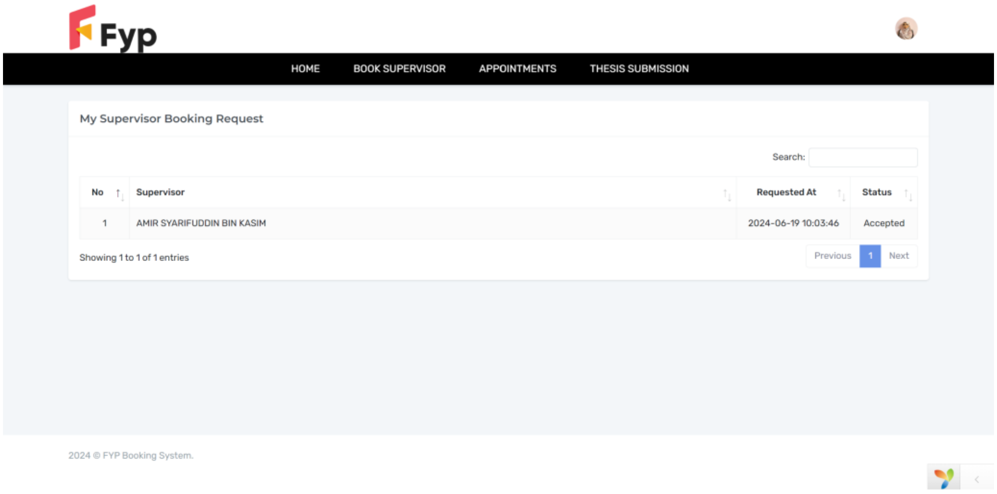
Full Name : NURUL AINA MARINA BINTI MANSOR

Username : NURUL AINA MARINA BINTI MANSOR

E-mail : b032110268@student.utem.edu.my

2024 © FYP Booking System.

Figure 4.10: Student Update Profile Interface



My Supervisor Booking Request

No	Supervisor	Requested At	Status
1	AMIR SYARIFUDDIN BIN KASIM	2024-06-19 10:03:46	Accepted

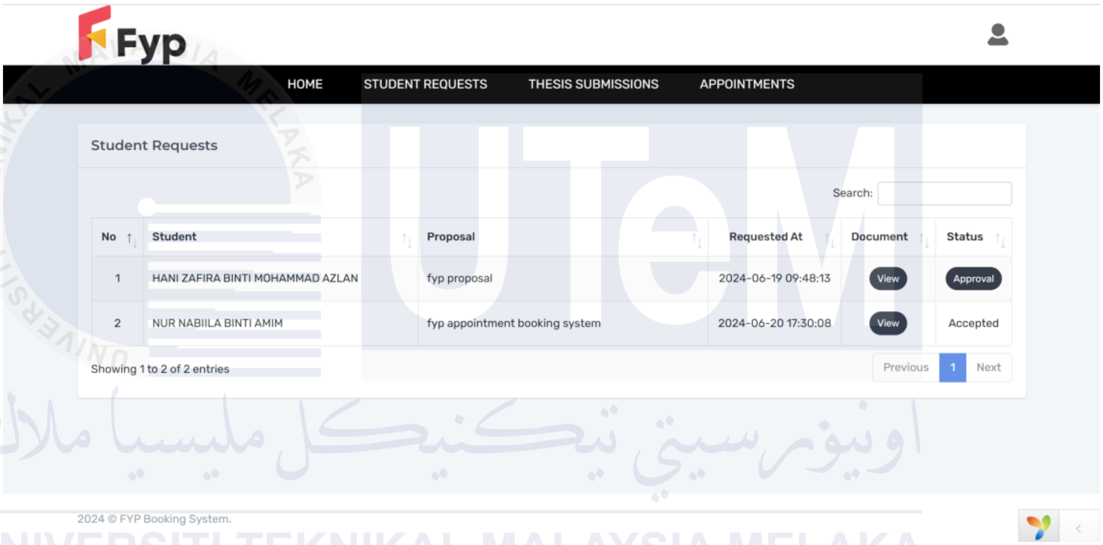
Showing 1 to 1 of 1 entries

2024 © FYP Booking System.

Figure 4.11: List Supervisor Request Interface

4.3.3 Supervisor GUI

The graphical user interface (GUI) for supervisors is designed to allow them to accept or reject student requests after reviewing their proposal documents as seen in Figure 4.12. Supervisors can also access and review the documentation submitted by their supervisees, as illustrated in Figure 4.13. Additionally, supervisors are responsible for approving student appointment requests, as depicted in Figure 4.14.

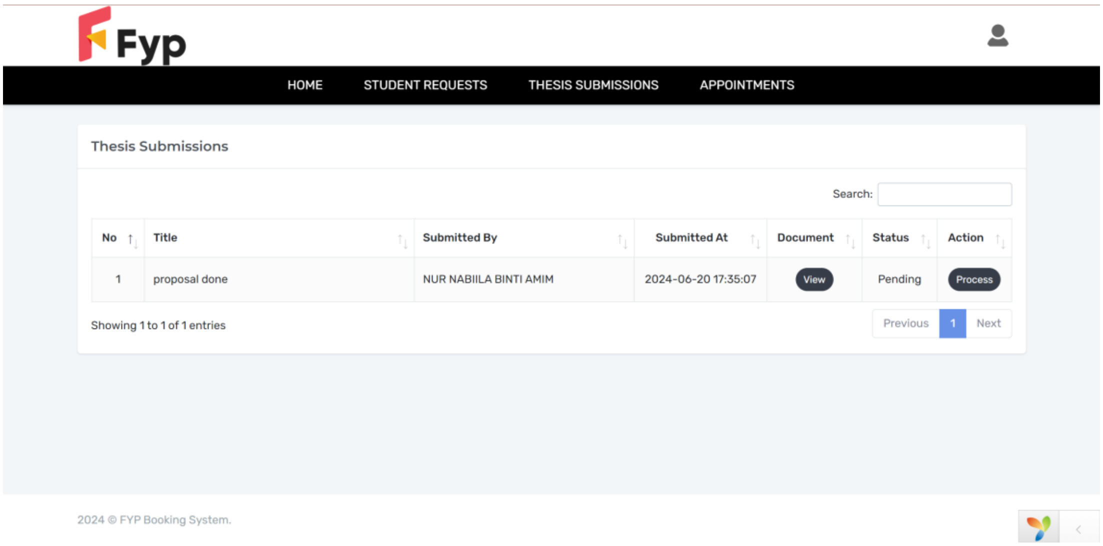


Student Requests

No	Student	Proposal	Requested At	Document	Status
1	HANI ZAFIRA BINTI MOHAMMAD AZLAN	fyp proposal	2024-06-19 09:48:13	View	Approval
2	NUR NABIILA BINTI AMIM	fyp appointment booking system	2024-06-20 17:30:08	View	Accepted

Showing 1 to 2 of 2 entries

Figure 4.12: Student Request Interface

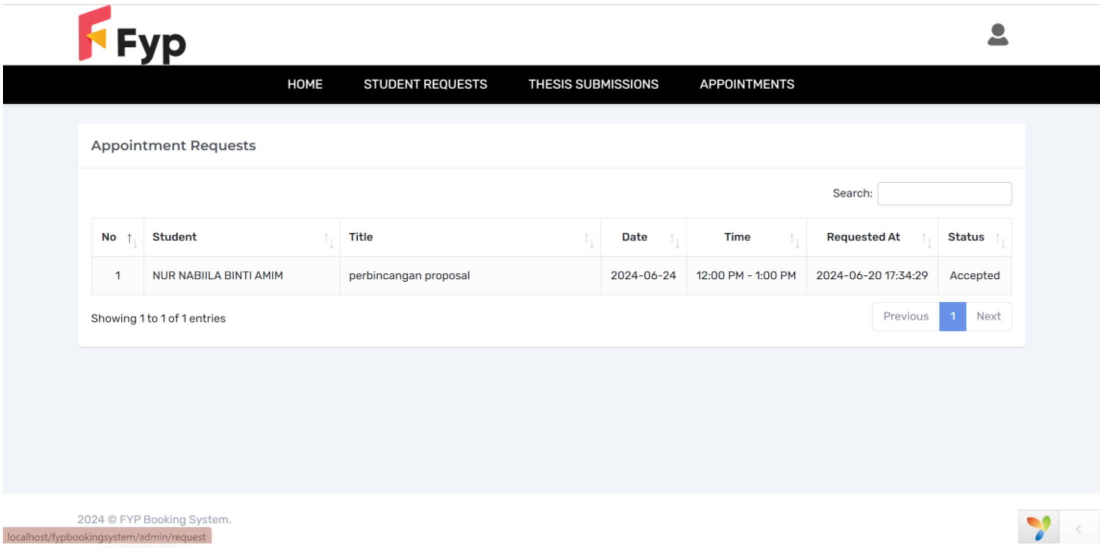


Thesis Submissions

No	Title	Submitted By	Submitted At	Document	Status	Action
1	proposal done	NUR NABIILA BINTI AMIM	2024-06-20 17:35:07	View	Pending	Process

Showing 1 to 1 of 1 entries

Figure 4.13: Student Document Submission Interface



Appointment Requests

Search:

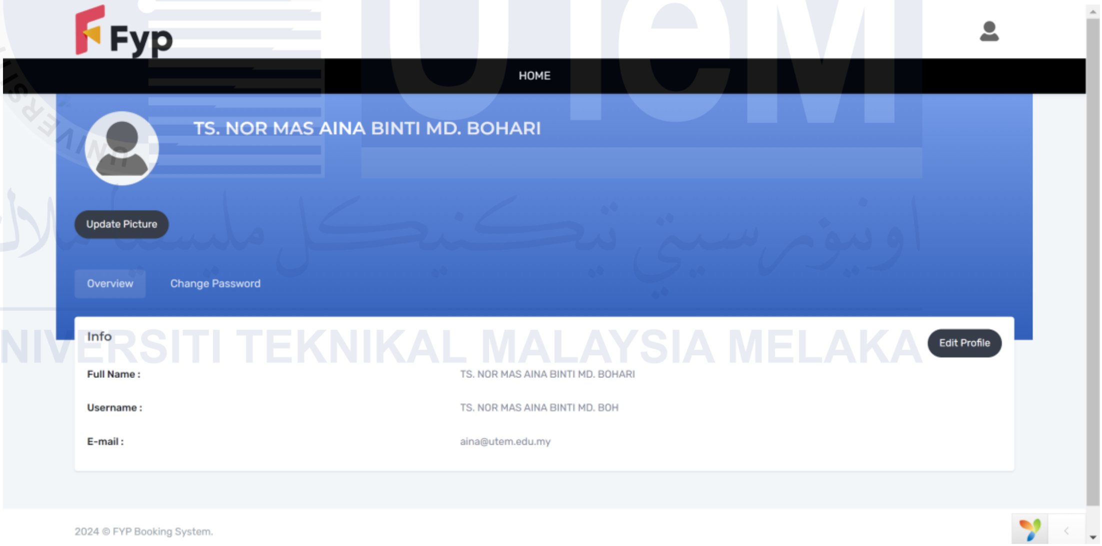
No	Student	Title	Date	Time	Requested At	Status
1	NUR NABIILA BINTI AMIM	perbincangan proposal	2024-06-24	12:00 PM - 1:00 PM	2024-06-20 17:34:29	Accepted

Showing 1 to 1 of 1 entries

2024 © FYP Booking System.

localhost/fypbookingsystem/admin/request

Figure 4.14: Student Appointment Approval Interface



Info

Full Name : TS. NOR MAS AINA BINTI MD. BOHARI

Username : TS. NOR MAS AINA BINTI MD. BOH

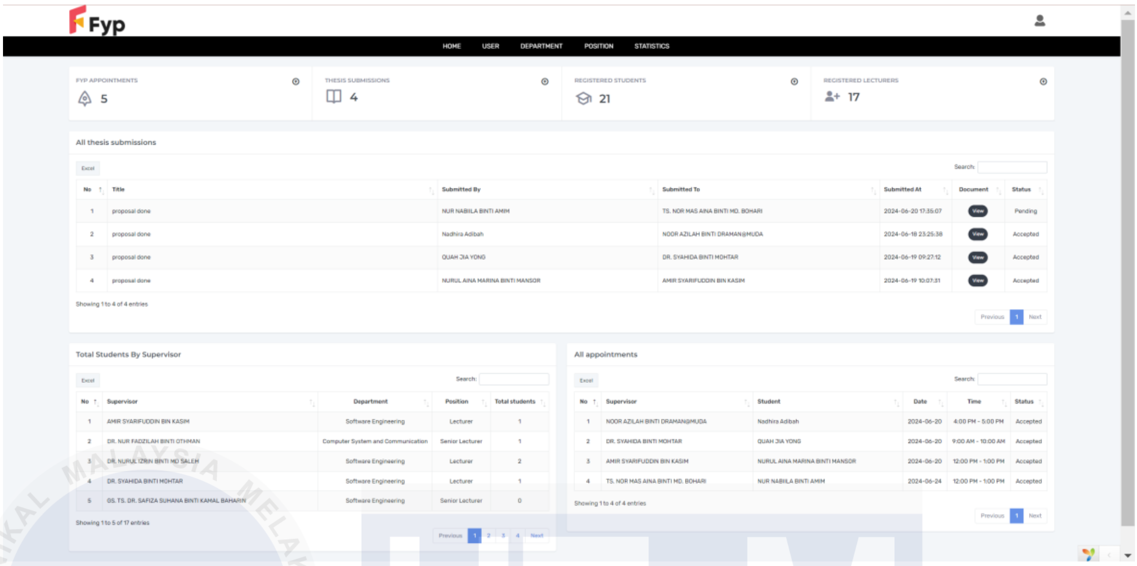
E-mail : aina@utem.edu.my

2024 © FYP Booking System.

Figure 4.15: Supervisor Update Profile Interface

4.3.4 Admin GUI

The graphical user interface (GUI) for administrators primarily allows them to view system data statistics, including all student appointments with their respective supervisors. As shown in Figure 4.17, administrators can search for user information, whether students or supervisors. Additionally, admin have the ability to activate or deactivate lists of departments and positions.



FYP APPOINTMENTS 5

THESIS SUBMISSIONS 4

REGISTERED STUDENTS 21

REGISTERED LECTURERS 17

All thesis submissions

Excel   Search:

No	Title	Submitted By	Submitted To	Submitted At	Document	Status
1	proposal done	NUR NABILA BINTI AMIM	TS. NOR MAS AINA BINTI MD. BOHARI	2024-06-20 17:35:07	View	Pending
2	proposal done	Nadhira Adibah	NOOR AZILAH BINTI DRAMAN@MUDA	2024-06-18 23:25:38	View	Accepted
3	proposal done	QUAH JIA YONG	DR. SYAHIDA BINTI MOHTAR	2024-06-19 09:27:12	View	Accepted
4	proposal done	NURUL AINA MARINA BINTI MANSOR	AMIR SYARIFUDDIN BIN KASIM	2024-06-19 10:07:31	View	Accepted

Showing 1 to 4 of 4 entries

Previous 1 Next

Total Students By Supervisor

Excel   Search:

No	Supervisor	Department	Position	Total students
1	AMIR SYARIFUDDIN BIN KASIM	Software Engineering	Lecturer	1
2	DR. NUR FADZILAH BINTI OTHMAN	Computer System and Communication	Senior Lecturer	1
3	DR. NURUL IZRIN BINTI MD SALEH	Software Engineering	Lecturer	2
4	DR. SYAHIDA BINTI MOHTAR	Software Engineering	Lecturer	1
5	GS. TS. DR. SAFIZA SUHANA BINTI KAMAL BAHARIN	Software Engineering	Senior Lecturer	0

Showing 1 to 5 of 17 entries

Previous 1 2 3 4 Next

All appointments

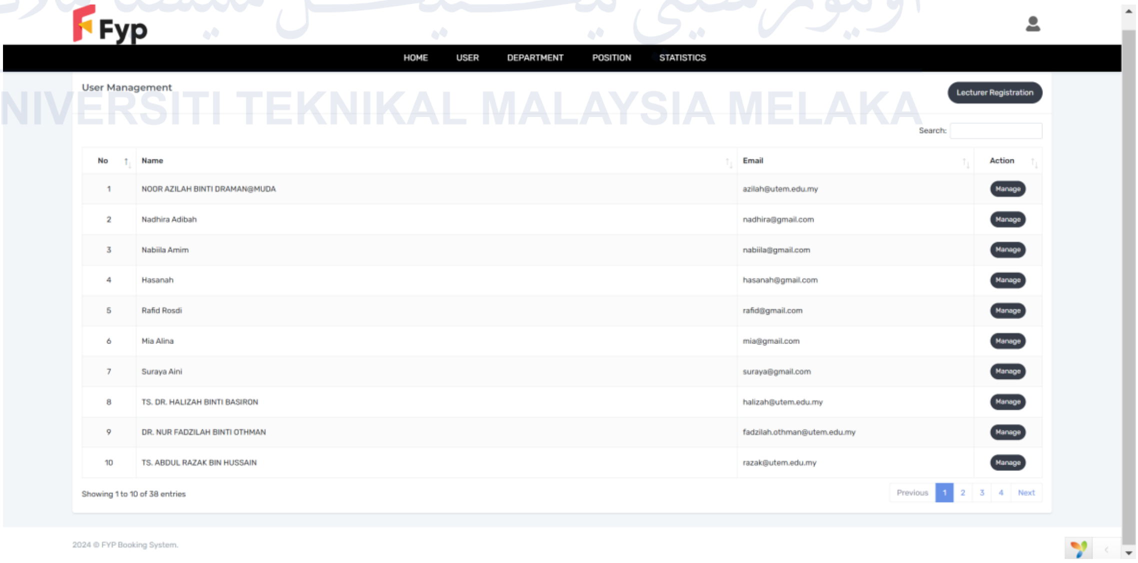
Excel   Search:

No	Supervisor	Student	Date	Time	Status
1	NOOR AZILAH BINTI DRAMAN@MUDA	Nadhira Adibah	2024-06-20	4:00 PM - 5:00 PM	Accepted
2	DR. SYAHIDA BINTI MOHTAR	QUAH JIA YONG	2024-06-20	9:00 AM - 10:00 AM	Accepted
3	AMIR SYARIFUDDIN BIN KASIM	NURUL AINA MARINA BINTI MANSOR	2024-06-20	12:00 PM - 1:00 PM	Accepted
4	TS. NOR MAS AINA BINTI MD. BOHARI	NUR NABILA BINTI AMIM	2024-06-24	12:00 PM - 1:00 PM	Accepted

Showing 1 to 4 of 4 entries

Previous 1 Next

Figure 4.16: Statistic Report Interface



User Management

Search:

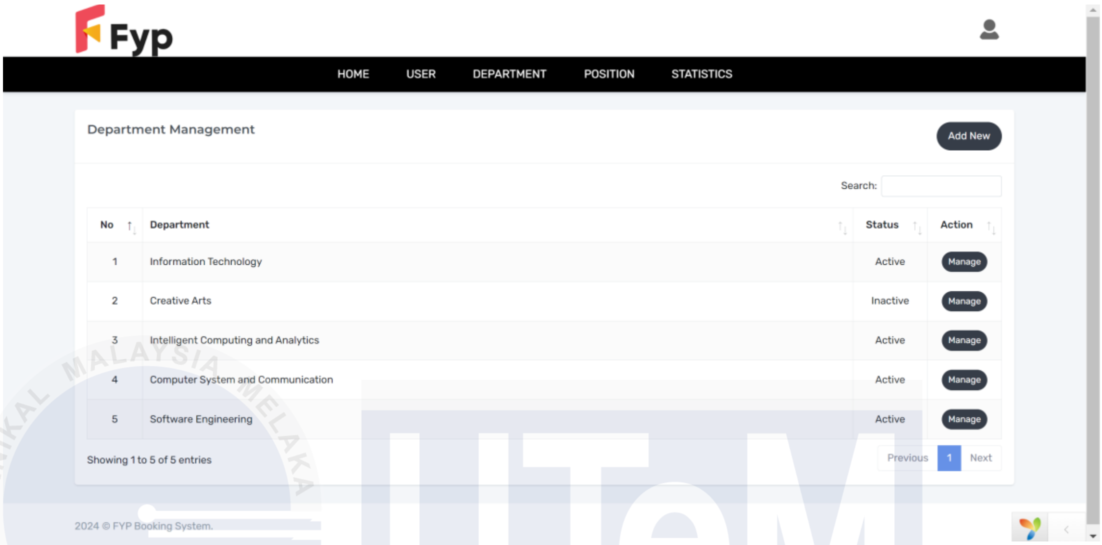
No	Name	Email	Action
1	NOOR AZILAH BINTI DRAMAN@MUDA	azilah@utem.edu.my	Manage
2	Nadhira Adibah	nadhira@gmail.com	Manage
3	Nabiila Amim	nabiila@gmail.com	Manage
4	Hasanah	hasanah@gmail.com	Manage
5	Rafid Rosdi	rafid@gmail.com	Manage
6	Mia Alina	mia@gmail.com	Manage
7	Suraya Aini	suraya@gmail.com	Manage
8	TS. DR. HALIZAH BINTI BASIRON	halizah@utem.edu.my	Manage
9	DR. NUR FADZILAH BINTI OTHMAN	fadzilah.othman@utem.edu.my	Manage
10	TS. ABDUL RAZAK BIN HUSSAIN	razak@utem.edu.my	Manage

Showing 1 to 10 of 38 entries

Previous 1 2 3 4 Next

2024 © FYP Booking System.

Figure 4.17: List User Interface



Department Management

Add New

Search:

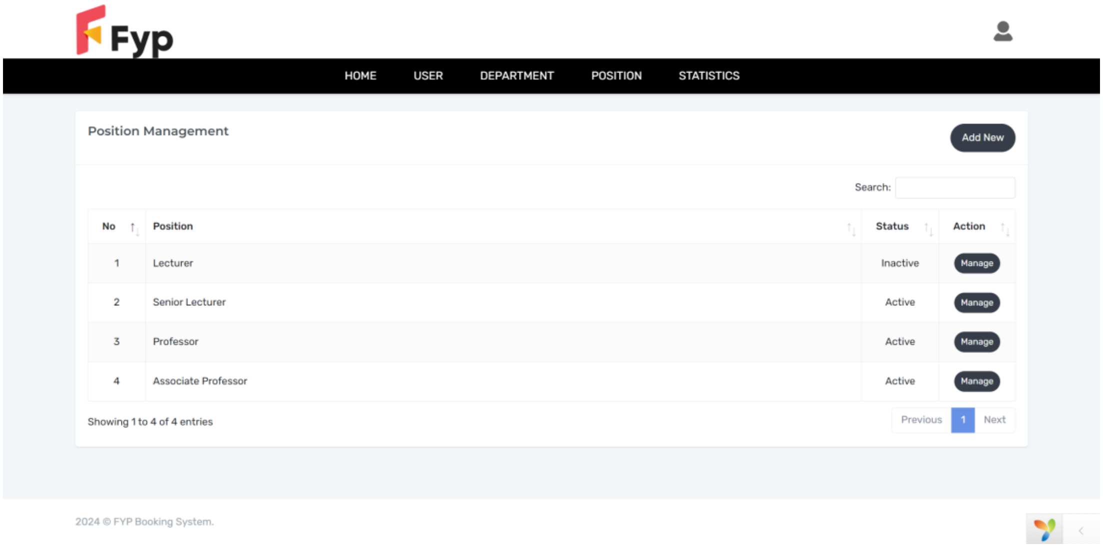
No	Department	Status	Action
1	Information Technology	Active	Manage
2	Creative Arts	Inactive	Manage
3	Intelligent Computing and Analytics	Active	Manage
4	Computer System and Communication	Active	Manage
5	Software Engineering	Active	Manage

Showing 1 to 5 of 5 entries

Previous 1 Next

2024 © FYP Booking System.

Figure 4.18: List Department Interface



Position Management

Add New

Search:

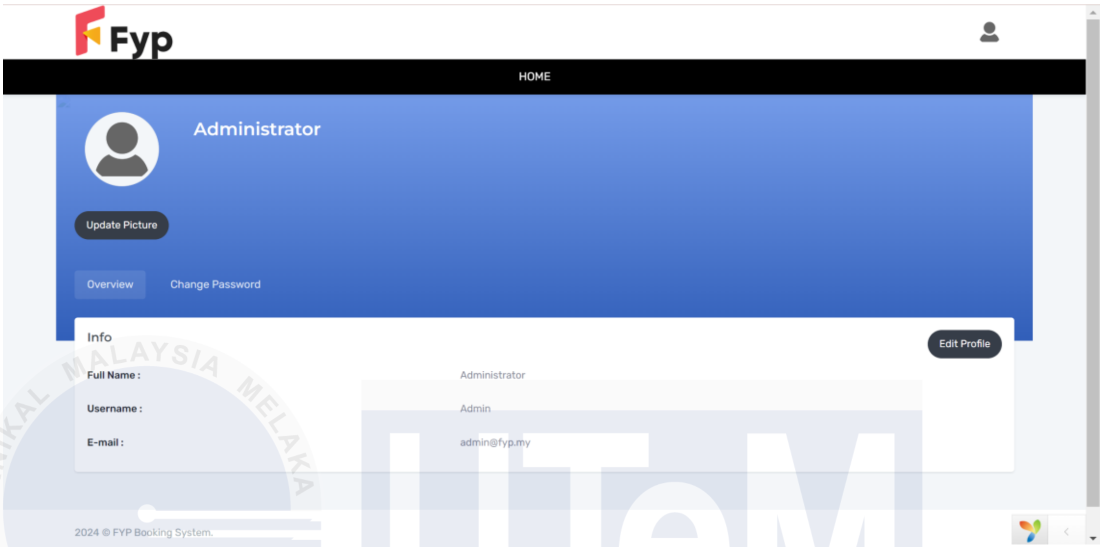
No	Position	Status	Action
1	Lecturer	Inactive	Manage
2	Senior Lecturer	Active	Manage
3	Professor	Active	Manage
4	Associate Professor	Active	Manage

Showing 1 to 4 of 4 entries

Previous 1 Next

2024 © FYP Booking System.

Figure 4.19: List Position Interface



Administrator

Info

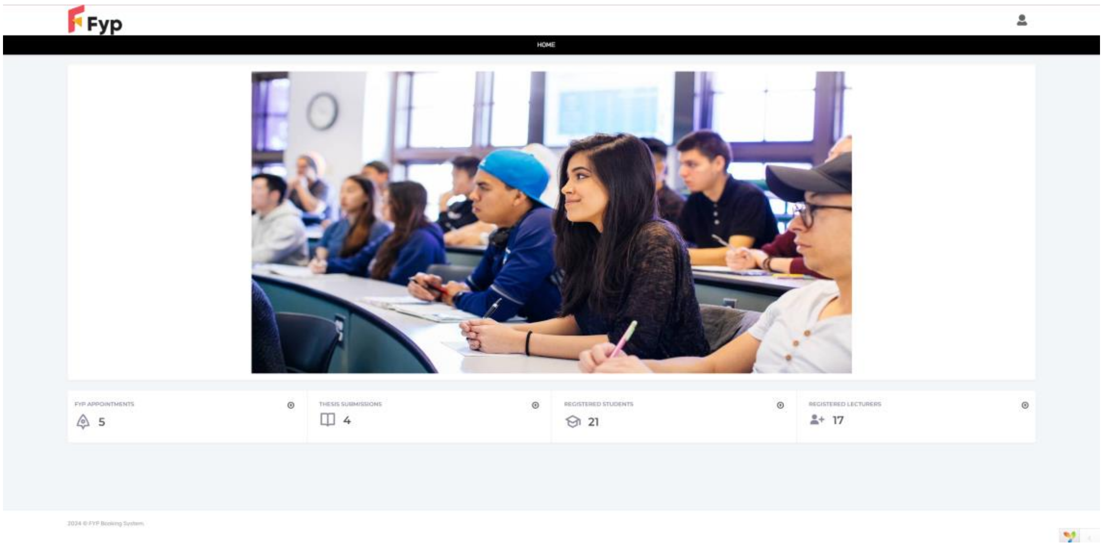
Full Name : Administrator

Username : Admin

E-mail : admin@fyp.my

2024 © FYP Booking System.

Figure 4.20: Admin Update Profile Interface



FYP APPOINTMENTS 5

THESIS SUBMISSIONS 4

REGISTERED STUDENTS 21

REGISTERED LECTURERS 17

2024 © FYP Booking System.

Figure 4.21: Admin Home Interface

4.4 Conclusion

In summary, this chapter has comprehensively discussed the project design, highlighting essential aspects for the system's development and implementation. It examined the Database Management System (DBMS) selection to ensure optimal performance, scalability, and reliability. The chapter outlined the Business Rules for Conceptual Design to maintain data integrity and consistent business logic, aligning with organizational goals.

It presented the Entity Relationship Diagram (ERD) to visualize the data model, along with a comprehensive Data Dictionary to ensure a shared understanding of data structures. The application of Triggers and Stored Procedures was explored, demonstrating their role in enforcing business rules, maintaining data integrity, enhancing performance, and promoting code reusability. This chapter has set the stage for developing a robust and efficient system, laying the groundwork for subsequent phases from design to implementation and deployment.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter, the primary objective is to effectively implement the database designed for the FYP Appointment Booking System. This will include a step-by-step guide on installing and configuring the database. XAMPP will be set up on Windows 11 to access PhpMyAdmin, and both Data Definition Language (DDL) and Data Manipulation Language (DML) commands will be executed to establish and manage the required database structures and data for the system.

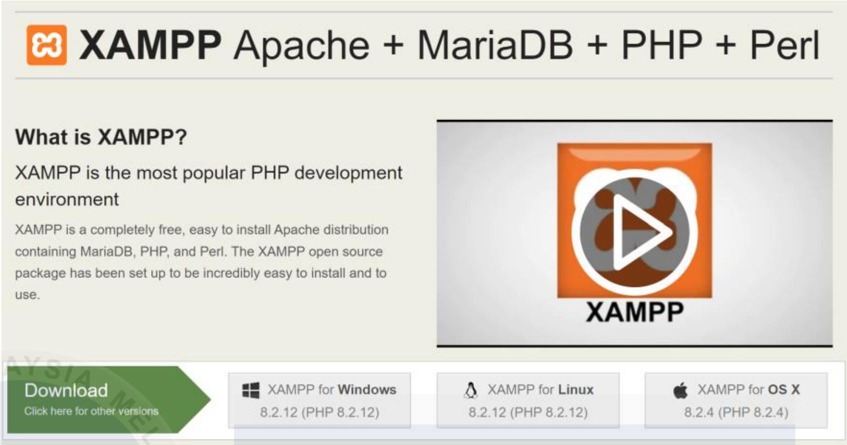
5.2 System Development Environment Setup

In the FYP Appointment Booking System, the software development must be set up before beginning website development. The project is built around three core components, each serving a distinct purpose to support the development and functionality of the system. The first component is the Web Server, for which XAMPP is used. XAMPP provides a robust and versatile local server environment that includes MySQL. This setup ensures a smooth development process and the ability to troubleshoot and debug applications in a controlled environment.

The second key component is MySQL, which serves as the database management tool for the project. MySQL is known for its reliability and efficiency in managing large volumes of data, making it an ideal choice for handling the appointment data and user information required by the FYP Appointment Booking System. Lastly, the third component involves the use of Visual Studio Code, a powerful code editor that supports multiple programming languages, including PHP, HTML, and JavaScript. Visual Studio Code is chosen for its extensive range of features, such as debugging, version control integration, and extensions that enhance development workflows.

5.2.1 Steps for installing XAMPP

Step 1: Install the latest version of XAMPP as shown in Figure 5.1 below.



XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

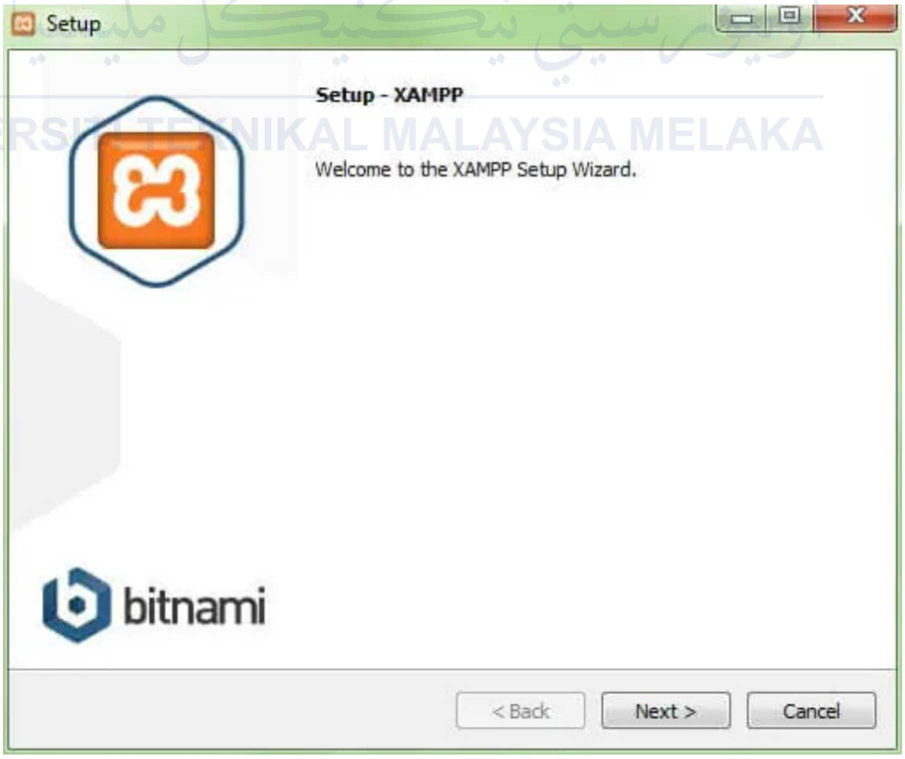
XAMPP is the most popular PHP development environment

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

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

Figure 5.1: Installation of XAMPP

Step 2: Click on 'Next' to proceed with configuring the installation settings.

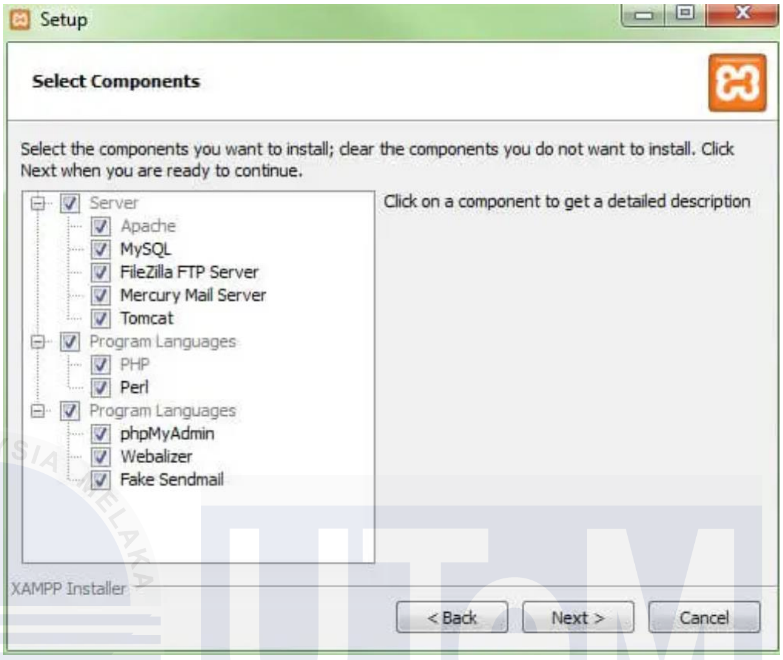


Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

Figure 5.2: Illustration setup process of XAMPP

Step 3: Choose software components to be installed



Setup

Select Components

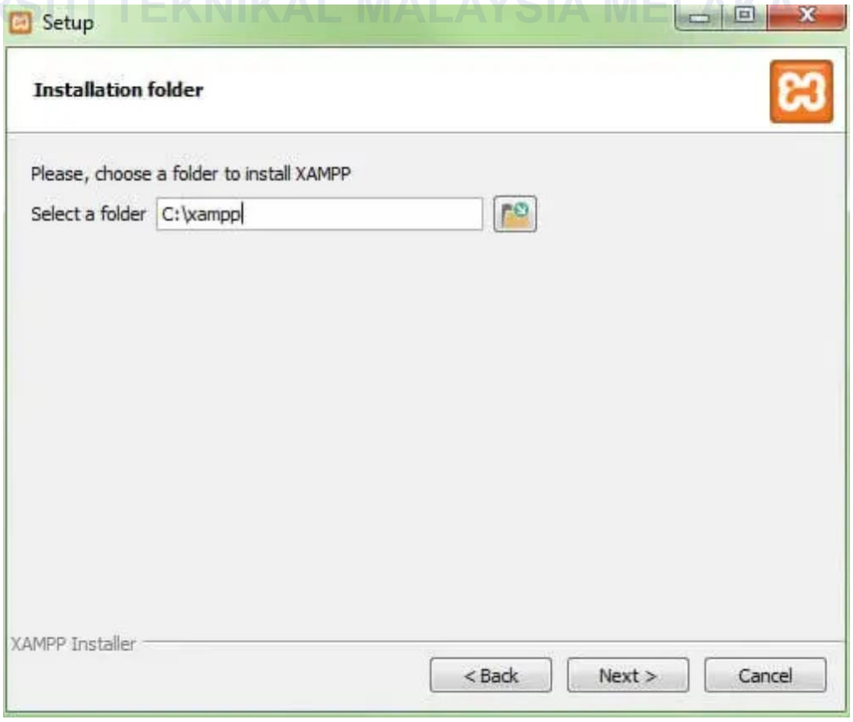
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

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

Click on a component to get a detailed description

Figure 5.3: Select components for setup XAMPP

Step 4: Choose the installation directory



Setup

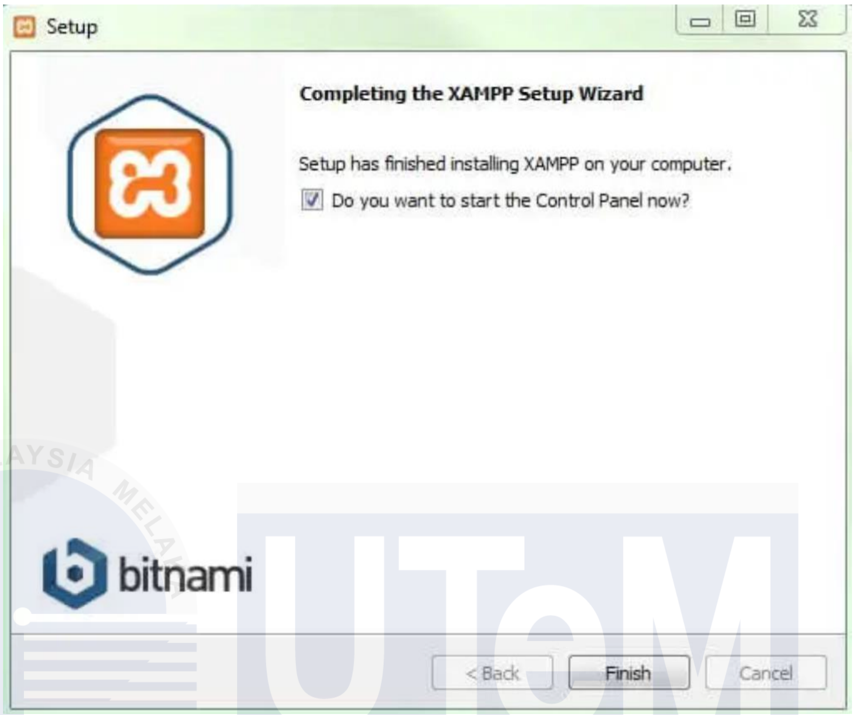
Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

Figure 5.4: Choosing a folder to install XAMPP

Step 5: Complete installation



Setup

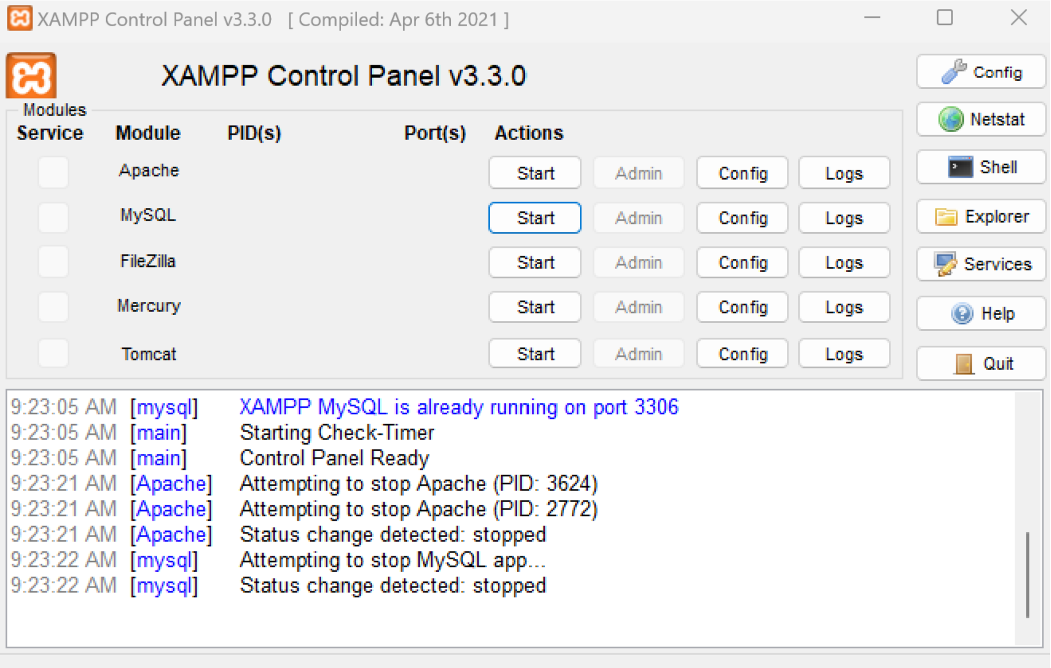
Completing the XAMPP Setup Wizard

Setup has finished installing XAMPP on your computer.

☑ Do you want to start the Control Panel now?

Figure 5.5: Completing the XAMPP setup

Step 6: Manage and control the various components of XAMPP using the control panel, such as starting Apache and MySQL to access the database.



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			Start Admin Config Logs
□	MySQL			Start Admin Config Logs
□	FileZilla			Start Admin Config Logs
□	Mercury			Start Admin Config Logs
□	Tomcat			Start Admin Config Logs

```
9:23:05 AM  [mysql]    XAMPP MySQL is already running on port 3306
9:23:05 AM  [main]     Starting Check-Timer
9:23:05 AM  [main]     Control Panel Ready
9:23:21 AM  [Apache]   Attempting to stop Apache (PID: 3624)
9:23:21 AM  [Apache]   Attempting to stop Apache (PID: 2772)
9:23:21 AM  [Apache]   Status change detected: stopped
9:23:22 AM  [mysql]    Attempting to stop MySQL app...
9:23:22 AM  [mysql]    Status change detected: stopped
```

Figure 5.6: XAMPP control panel

5.3 Database Implementation

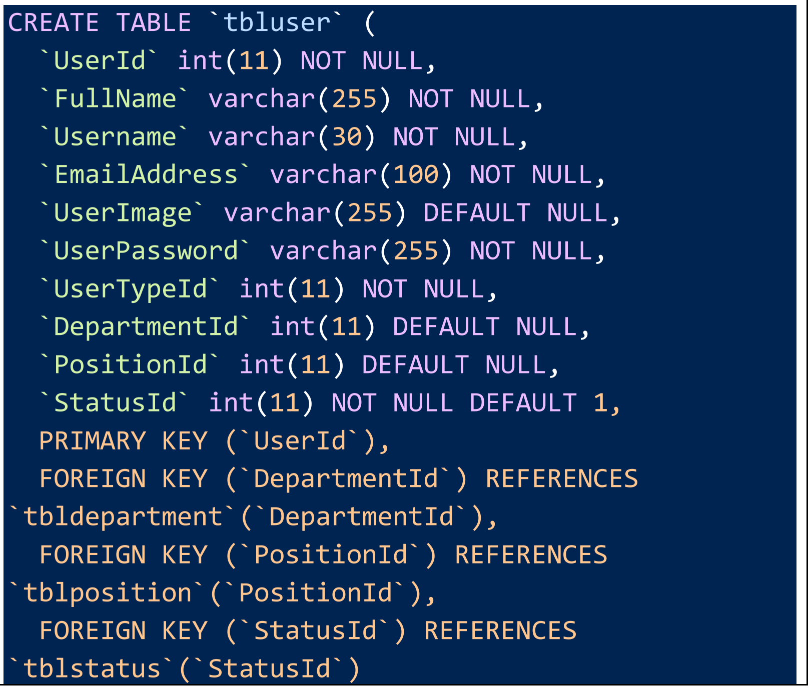
In the database implementation phase, various queries are executed to manage the database, including simple and complex queries, aggregate functions, stored procedures, and triggers. These queries are designed to insert, retrieve, validate, and verify information within the database.

5.3.1 Data Definition Language

Data Definition Language (DDL) refers to SQL commands used for creating and managing tables in a relational database. DDL statements allow the creation, alteration, and deletion of objects within the database, including tables, procedures, functions, and triggers for the FYP Appointment Booking System.

This system consists of ten tables used inside the FYP Appointment Booking System. Table 5.1 shows the DDL used to create these tables using SQL Server syntax and commands.

Table 5.1: DDL command

<table>
<tr><th>Table Name</th><th>DDL command</th></tr>
<tr><td>tbluser</td><td>
<pre>
CREATE TABLE `tbluser` (
  `UserId` int(11) NOT NULL,
  `FullName` varchar(255) NOT NULL,
  `Username` varchar(30) NOT NULL,
  `EmailAddress` varchar(100) NOT NULL,
  `UserImage` varchar(255) DEFAULT NULL,
  `UserPassword` varchar(255) NOT NULL,
  `UserTypeId` int(11) NOT NULL,
  `DepartmentId` int(11) DEFAULT NULL,
  `PositionId` int(11) DEFAULT NULL,
  `StatusId` int(11) NOT NULL DEFAULT 1,
  PRIMARY KEY (`UserId`),
  FOREIGN KEY (`DepartmentId`) REFERENCES
`tbldepartment`(`DepartmentId`),
  FOREIGN KEY (`PositionId`) REFERENCES
`tblposition`(`PositionId`),
  FOREIGN KEY (`StatusId`) REFERENCES
`tblstatus`(`StatusId`)
</pre></td></tr>
</table>

	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

Tblappointment	CREATE TABLE `tblappointment` (
	  `AppointmentId` int(11) NOT NULL,
	  `SupervisorId` int(11) NOT NULL,
	  `StudentId` int(11) NOT NULL,
	  `AppointmentTitle` varchar(1000) NOT NULL,
	  `AppointmentDate` date NOT NULL,
	  `SlotId` int(11) NOT NULL,
	  `ApprovalStatusId` int(11) NOT NULL,
	  `TransactionDate` timestamp NOT NULL DEFAULT
	current_timestamp(),
	  PRIMARY KEY (`AppointmentId`),
	  FOREIGN KEY (`SupervisorId`) REFERENCES
	`tbluser`(`UserId`),
	  FOREIGN KEY (`StudentId`) REFERENCES
	`tbluser`(`UserId`),
	  FOREIGN KEY (`SlotId`) REFERENCES
	`tblslot`(`SlotId`),
	  FOREIGN KEY (`ApprovalStatusId`) REFERENCES
	`tblapprovalstatus`(`ApprovalStatusId`)
	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
	COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

tblapprovalstatus	CREATE TABLE `tblapprovalstatus` (
	  `ApprovalStatusId` int(11) NOT NULL,
	  `ApprovalStatusDesc` varchar(255) NOT NULL,
	  `ApprovalStatusSubmission` varchar(255) NOT
	NULL,
	  PRIMARY KEY (`ApprovalStatusId`)
	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
	COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

tblbookingsupervisor	CREATE TABLE `tblbookingsupervisor` (
	  `BookingSupervisorId` int(11) NOT NULL,
	  `SupervisorId` int(11) NOT NULL,
	  `StudentId` int(11) NOT NULL,
	  `ApprovalStatusId` int(11) NOT NULL,
	  `ProposalTitle` varchar(1000) NOT NULL,
	  `DocumentFile` varchar(255) NOT NULL,
	  `TransactionDate` timestamp NOT NULL DEFAULT
	current_timestamp(),
	  PRIMARY KEY (`BookingSupervisorId`),

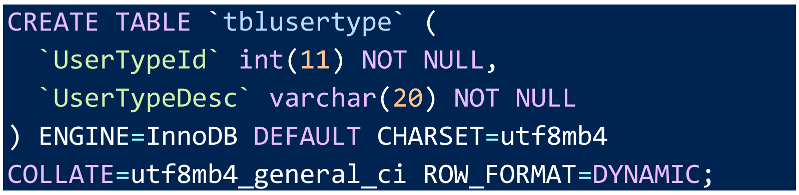
	FOREIGN KEY (`SupervisorId`) REFERENCES `tbluser`(`UserId`), FOREIGN KEY (`StudentId`) REFERENCES `tbluser`(`UserId`), FOREIGN KEY (`ApprovalStatusId`) REFERENCES `tblapprovalstatus`(`ApprovalStatusId`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

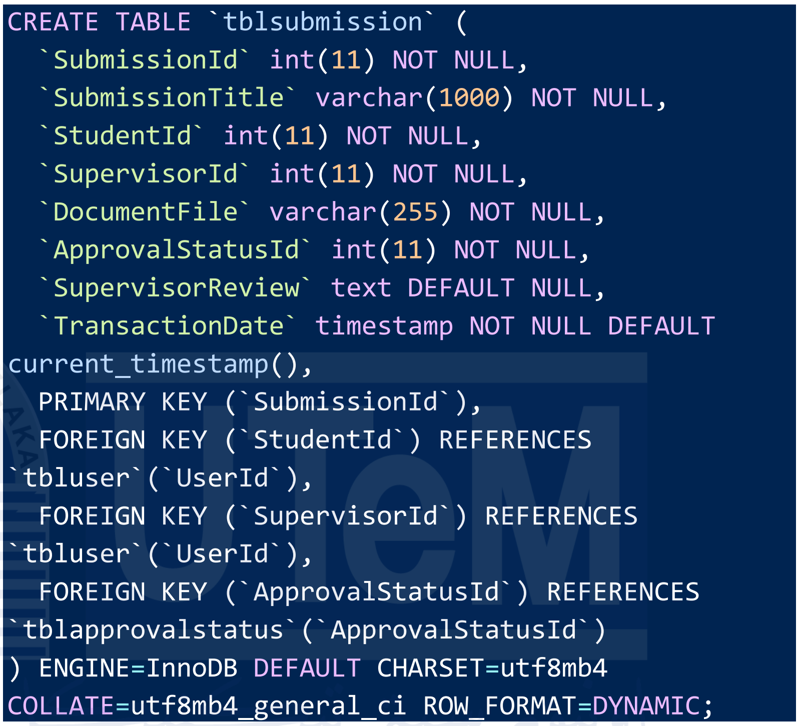
tbldepartment	CREATE TABLE `tbldepartment` (
	  `DepartmentId` int(11) NOT NULL,
	  `DepartmentDesc` varchar(255) NOT NULL,
	  `StatusId` int(11) NOT NULL,
	  PRIMARY KEY (`DepartmentId`),
	  FOREIGN KEY (`StatusId`) REFERENCES
	`tblstatus`(`StatusId`)
	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
	COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

tblposition	CREATE TABLE `tblposition` (
	  `PositionId` int(11) NOT NULL,
	  `PositionDesc` varchar(255) NOT NULL,
	  `StatusId` int(11) NOT NULL,
	  PRIMARY KEY (`PositionId`),
	  FOREIGN KEY (`StatusId`) REFERENCES
	`tblstatus`(`StatusId`)
	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
	COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

tblslot	CREATE TABLE `tblslot` (
	  `SlotId` int(11) NOT NULL,
	  `SlotDesc` varchar(50) NOT NULL,
	  `TimeStart` time NOT NULL,
	  `TimeEnd` time NOT NULL,
	  PRIMARY KEY (`SlotId`)
	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
	COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

tblstatus	CREATE TABLE `tblstatus` (
	  `StatusId` int(11) NOT NULL,
	  `StatusDesc` varchar(10) NOT NULL,
	  PRIMARY KEY (`StatusId`)
	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
	COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

tblusertype	
	CREATE TABLE `tblusertype` (
	  `UserTypeId` int(11) NOT NULL,
	  `UserTypeDesc` varchar(20) NOT NULL
	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
	COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

tblsubmission	
	CREATE TABLE `tblsubmission` (
	  `SubmissionId` int(11) NOT NULL,
	  `SubmissionTitle` varchar(1000) NOT NULL,
	  `StudentId` int(11) NOT NULL,
	  `SupervisorId` int(11) NOT NULL,
	  `DocumentFile` varchar(255) NOT NULL,
	  `ApprovalStatusId` int(11) NOT NULL,
	  `SupervisorReview` text DEFAULT NULL,
	  `TransactionDate` timestamp NOT NULL DEFAULT
	current_timestamp(),
	  PRIMARY KEY (`SubmissionId`),
	  FOREIGN KEY (`StudentId`) REFERENCES
	`tbluser`(`UserId`),
	  FOREIGN KEY (`SupervisorId`) REFERENCES
	`tbluser`(`UserId`),
	  FOREIGN KEY (`ApprovalStatusId`) REFERENCES
	`tblapprovalstatus`(`ApprovalStatusId`)
	) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
	COLLATE=utf8mb4_general_ci ROW_FORMAT=DYNAMIC;

5.3.2 Stored Procedures

Table 5.2: Stored Procedures in FYP Appointment Booking System

No	Procedure	Example
1	Appointment Booking	CREATE PROCEDURE sp_BookAppointment @StudentId INT, @SupervisorId INT, @AppointmentDate DATETIME, @AppointmentType VARCHAR(50) AS BEGIN -- Check if appointment slot is available IF NOT EXISTS (SELECT 1 FROM Appointments WHERE SupervisorId = @SupervisorId AND AppointmentDate = @AppointmentDate)

		BEGIN
		-- Insert appointment into database
		INSERT INTO Appointments (StudentId, SupervisorId, AppointmentDate, AppointmentType)
		VALUES (@StudentId, @SupervisorId, @AppointmentDate, @AppointmentType);
		
		-- Additional logic like sending notifications can be added here
		END
		ELSE
		BEGIN
		RAISERROR ('Appointment slot is already booked.', 16, 1);
		END
		END

2	Search Student List	select * from tbluser WHERE UserTypeID = 2
3	Search Supervisor List	select * from tbluser where UserTypeID = 3

5.3.3 Triggers

Table 5.3: Triggers in FYP Appointment Booking System

No	Trigger	Example
1	Booking Supervisor	CREATE TRIGGER tr_MaxStudentsPerSupervisor ON Appointments AFTER INSERT AS BEGIN -- Declare variables to hold supervisor ID and

		appointment count
		DECLARE @SupervisorId INT;
		DECLARE @AppointmentCount INT;
		-- Get the SupervisorId from the inserted row
		SELECT @SupervisorId = SupervisorId
		FROM INSERTED;
		-- Count the number of active appointments for the supervisor
		SELECT @AppointmentCount = COUNT(*)
		FROM Appointments
		WHERE SupervisorId = @SupervisorId
		AND AppointmentStatus <> 'Canceled'; -- Consider only active appointments
		-- Check if the supervisor has reached the maximum allowed number of appointments (5)
		IF @AppointmentCount >= 5
		BEGIN

		-- Raise an error to prevent the new appointment from being added
		RAISERROR ('Supervisor has reached the maximum of 5 appointments.', 16, 1);
		-- Rollback the transaction to prevent the insertion
		ROLLBACK TRANSACTION;
		END
		ELSE
		BEGIN
		-- Optionally, you can log this action or perform other tasks
		PRINT 'Appointment successfully booked.';
		END
		END;

5.4 Conclusion

In this chapter, a comprehensive overview of the development process for the FYP Appointment Booking System is provided. It begins with a detailed explanation of setting up the software development environment, which is a crucial step for ensuring that the system functions correctly on a local server. The chapter includes a step-by-step guide on how to install and configure XAMPP on a 64-bit Windows operating system. This involves downloading the correct version of XAMPP, installing it, and configuring the Apache and MySQL services to work seamlessly together. The configuration process is vital, as it establishes the necessary environment for developing and testing the appointment booking system.

Following the setup of the development environment, the chapter delves into the development process itself. This section covers the implementation of various components of the database and backend logic, including the creation of tables and relationships using Data Definition Language (DDL) commands. DDL is used to define the database schema, which involves creating tables, specifying their structure, and defining primary and foreign keys to establish relationships between different data entities.

Overall, this chapter provides a thorough account of the technical steps involved in building the FYP Appointment Booking System, emphasizing the importance of each phase in creating a reliable and user-friendly application.

CHAPTER 6: TESTING

6.1 Introduction

The testing phase of the methodology discussed in Chapter Two will be elaborated upon in this chapter, focusing on the FYP Appointment Booking System. This testing documentation includes the test plan, which outlines the initial study of testing on this system, as well as test organization, test environment, test schedule, test strategy, test design, test description, test data, and the analysis of test results. The test environment consists of the hardware and software used to test the FYP Appointment Booking System. Moreover, the test strategy is critical in this testing phase as it guides the methods of testing to be utilized.

6.2 Test Plan

A test plan is a document that outlines the approach, scope, resources, and schedule for the planned testing activities. It identifies various aspects, including the items to be tested, the features to be evaluated, the tasks involved, and who will perform each task. The document also details the level of independence required for testers, the testing environment, test design techniques, entry and exit criteria, the reasoning behind these choices, and any risks that necessitate contingency planning. Essentially, it serves as a record of the test planning process.

6.2.1 Test Organization

In this project, the testing team is composed of both developer and tester. The scope of the testing covers both functional and non-functional requirements. Developer and tester are collaborating in order to identify defects and bugs in the system, ensuring that every failure and error in the system is documented and addressed.

Table 6.1: Roles and Responsibilities of Organization

Tester ID	Position	Responsibilities
T01	System Developer	- Prepare the test plan, test environment, test schedule, test description, test data and test result and analysis - Find bugs and defect in the system - Fix error and defect in the system
T02	System Tester	- Detect system error - Provide feedback regarding enhancing the system - To read all the documents and understand what needs to be tested
T03	FYP Student	- Testing the FYP system (Login as student)

6.2.2 Test Environment

The test environment includes various components that facilitate test execution, such as software, hardware, and network configurations capable of connecting multiple elements set up by the developer. The design of the test environment should replicate the production environment to uncover any configuration or environment-related issues. Table 6.2 lists the hardware components used during this development, while Table 6.3 outlines the programs and software installed on the computer to set up the web system and configure the database.

Table 6.2: Test Environment of Hardware Components

Environment Specification	Description
Laptop	HP Pavilion Laptop Intel 13th i5
Processor	AMD Ryzen 5 5600U with Radeon Graphics
Random Access Memory (RAM)	8GB

Table 6.3: Test Environment of Software Components

Environment Specification	Description
Database	MySQL
Web Browser	Windows 11
Web Server	XAMPP
Text Editor	Visual Studio Code
Documentation	Microsoft Word 365

6.2.3 Test Schedule

A test schedule is a planned timetable outlining the software testing process, which includes a list of testing activities or tasks, their scheduled start and end dates, and the roles and responsibilities assigned to each task. This schedule should also detail how the testing progress will be monitored, assessed, and approved. Table 6.4 provides a comprehensive overview of the modules, types of testing, start and end dates, and the designated tester for each phase of the testing process.

Table 6.4: Test Schedule in FYP Appointment Booking System

Module	Start Date	End Date	Test Type	Tester
Register	1/8/2024	4/8/2024	Unit Testing	T01, T02
Login	5/8/2024	6/8/2024	Unit Testing, Integration Testing	T01, T02, T03
Forgot Password	7/8/2024	8/8/2024	Unit Testing, Integration Testing	T02, T03
Appointment Booking	9/8/2024	12/8/2024	Unit Testing, Integration Testing	T01, T02, T03
Booking Supervisor	13/8/2024	16/8/2024	Unit Testing, Integration Testing	T01, T02, T03
Document Submission	17/8/2024	20/8/2024	Unit Testing, Integration Testing	T01, T02, T03
Admin	21/8/2024	26/8/2024	Integration Testing	T01, T02

6.3 Test Strategy

A test strategy is designed to improve the efficiency of the testing process. It has developed into a set of guidelines that outline test design, testing techniques, strategies used, and the prioritization of modules. This strategy, which is a component of the test plan, functions independently. It includes black box testing, also known as Behavioural Testing, where the tester does not know the internal structure, design, or implementation of the system being tested. This approach focuses on verifying functional requirements rather than non-functional ones. Black box testing method is used during this testing phase.

Black box testing, also known as Behavioural Testing, is a software testing method where the tester does not have any knowledge of the internal structure, design, or implementation of the system being tested. This approach primarily focuses on evaluating the functional requirements of the software rather than the non-functional aspects.

In contrast, white box testing, also referred to as Code-Based Testing or Structural Testing, involves a testing approach where the tester has knowledge of the internal code and structure. The tester examines specific paths through the code and ensures that the software produces the correct outputs based on given inputs.

Unit testing is the initial level of testing performed before integration testing for the FYP Appointment Booking System. It involves testing individual components or modules of the system in isolation. Each module is tested to ensure it performs as expected according to its design and requirements. The primary goal of unit testing for the FYP Appointment Booking System is to verify that each component, such as the login, registration, appointment booking, booking supervisor, document submission and report modules, functions correctly before being integrated with other parts of the system. This testing is typically conducted by developers during the coding phase, using white box testing techniques to assess the internal logic, data flow, and code paths of the components. Unit testing helps identify bugs early in the development process, making it easier and more cost-effective to resolve issues before they impact other areas of the system.

Unit testing is a technique for software testing in which distinct parts or modules of a system are tested independently to make sure they perform as intended and meet all criteria. Unit testing for the FYP Appointment Booking System entails separating and testing individual modules, including document submission, report, supervisor booking, appointment booking, login, and registration module, to ensure they all function as intended. Early in the development phase, problems can be found and fixed by developers by concentrating on these small, isolated components of the system. By ensuring that every part functions well before combining it with others, the FYP Appointment Booking System becomes more dependable and stronger in the end.

Integration testing is the process of testing how different modules or components of a system work together as a whole. For the FYP Appointment Booking System, integration testing involves combining individual modules and testing them as a group to ensure they interact correctly and function as expected when integrated. There are two main types of integration testing: the Big Bang approach and the Incremental approach. The Big Bang approach involves integrating all components or modules of the system at once and then testing them together. In contrast, the Incremental Testing approach involves gradually integrating and testing two or more logically related modules. Additional modules are then incrementally added and tested step-by-step until the entire system has been fully integrated and tested.

The incremental approach can be further divided into two methods: the top-down approach and the bottom-up approach. Bottom-up integration testing is a software testing strategy where testing starts with the lower-level modules, components, or units of a system, which are tested first. Once these lower-level modules are verified to work correctly, higher-level modules are gradually integrated and tested, progressing upwards in the system's hierarchy. This approach allows testers to focus on ensuring the functionality and reliability of the fundamental building blocks before moving on to more complex interactions, making it easier to identify and isolate defects in lower-level modules. Top-down integration testing is a testing strategy where the process starts with the highest-level modules of a system and gradually integrates and tests lower-level modules. This method begins by testing the main control modules and then progressively integrates and tests the subordinate modules. The goal is to verify the interactions and functionality between higher-level and lower-level components as they are integrated. This approach allows for early demonstration of the system’s core functionality, even if some of the lower-level components are still under development.

For the FYP Appointment Booking System, the Top-down approach is generally better for integration testing. This approach allows for early testing of key modules, such as login and appointment booking, even if some lower-level components like document submission or report generation are still being developed. By starting with the higher-level components, you can demonstrate and validate the core features early, which facilitates gathering initial feedback. Additionally, this method helps identify issues with the overall system structure and interactions between major modules at an early stage. It also makes it easier to test and integrate crucial functionalities before the entire system is fully completed. Early validation of main functionalities reduces the risk of significant problems later in the development process and supports more effective troubleshooting and refinement.

System testing is a comprehensive testing phase where the entire system is tested to ensure it meets the specified requirements and functions correctly in all intended scenarios. For the FYP Appointment Booking System, system testing involves evaluating the integrated application to verify that all modules work together seamlessly. During system testing, the focus is on validating end-to-end processes, ensuring that the system behaves as expected under various conditions. This includes checking functionality, performance, security, and usability to ensure that the system performs well in a production-like environment. System testing aims to uncover any defects or issues that may not have been identified during earlier testing phases, ensuring the complete system operates correctly and meets user requirements before it is deployed.

User Acceptance Testing (UAT) is a critical phase in the software testing process where the end-users or stakeholders validate the system to ensure it meets their needs and requirements. For the FYP Appointment Booking System, UAT involves having actual users such as students, lecturers, and admin to test the system in a real-world scenario to confirm that it performs as expected from their perspective. During UAT, users perform tasks and interact with the system to verify that all functionalities, such as booking appointments, managing supervisors, and submitting documents, align with their expectations and requirements. This testing phase focuses on ensuring that the system is user-friendly, reliable, and ready for production. Feedback gathered from UAT helps identify any usability issues, gaps, or improvements needed before the system is officially launched. The goal is to ensure the system meets user needs and delivers a satisfactory experience before final deployment.

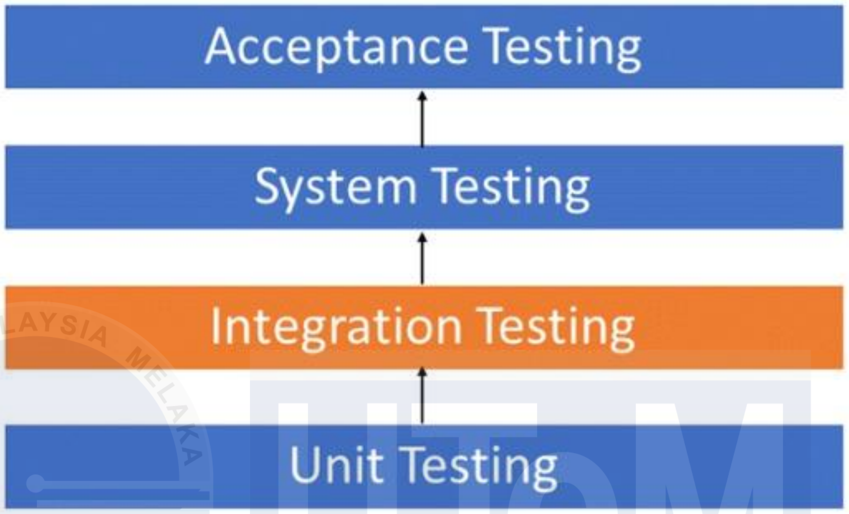


Figure 6.1: Software System Hierarchy

In conclusion, these testing phases provide a comprehensive approach to identifying and addressing issues at different stages of development, ensuring a robust, functional, and user-friendly final product of the system.

6.3.1 Classes of Tests

There are two types of test class that is implemented for this testing process in the FYP Appointment Booking System.

i. Functionality Testing

Functionality testing is a type of testing that ensures the FYP Appointment Booking System operates according to its specifications and meets all functional requirements. This testing involves verifying each function of the system, such as user registration, login, appointment booking, booking supervisor, report and document submission. It checks for the correct operation of all user interactions, database operations, and error handling processes. The primary goal is to ensure that all features work as intended and that the system performs its tasks accurately and efficiently without any functional defects.

ii. Security Testing

Security testing focuses on ensuring the system is safeguarded against unauthorized access and potential threats. The goal is to verify that only authorized users such as students, lecturers, and admin are the only users that can access specific functionalities and sensitive information within the system. Security testing evaluates measures like login system. By thoroughly testing these security features, the system can be fortified against breaches, ensuring that user data and system operations remain secure and protected from unauthorized access or malicious activities.

6.4 Test Design

Test design involves creating and writing test suites to evaluate a software application. The purpose of test design is to ensure that the software meets the specified requirements and aligns with the client's needs and expectations. Test design is divided into two main components which are test description and test data.

6.4.1 Test Description

The test description outlines the identification of each test case, the type of testing to be conducted, test date, test strategy, test description, step-by-step procedures for each test case, and the expected output results. These elements are designed and documented for every module test case. Tables 6.5 to 6.9 present the modules that have been tested to achieve the possible expected results.

Table 6.5: Test Description for Register Module (FYP Student)

Test Module	Register Module: FYP Student		
Test Type	Unit Testing	**Test Date**	1/8/2024 - 2/8/2024
Test Strategy	Black Box Testing		
Test Description	New FYP student registration		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC1_01	Ensure that a new user can successfully register with valid inputs for all required fields.	1. Enter all information needed for registration 2. Press the "Signup" button.	Registration is successful, and the user is redirected to the login page.
TC1_02	Ensure that the system displays an error message when the passwords do not match.	1. Enter all information needed for registration with the mismatch password 2. Press the "Signup" button.	Registration fails, and the system displays an error message stating "Your confirmation password does not match."
TC1_03	Validate that the registration process fails when the password does not meet complexity requirements (e.g., lacks special characters)	1. Enter all information needed for registration with the missing special characters 2. Press the "Signup" button.	Registration fails, and the system displays an error message stating "Password must contain at least one special character."
TC1_04	Ensure that the system detects and displays an error when the email address format is invalid.	1. Enter all information needed for registration with invalid email format	Registration fails, and the system displays an error message stating

		2. Press the "Signup" button.	"Please enter a valid email address."
TC1_05	Verify that the registration process does not proceed when mandatory fields are left blank.	1. Enter information needed for registration with empty fields on the username field 2. Press the "Signup" button.	Registration fails, and the system displays an error message stating "Username is required."
TC1_06	Ensure that the system does not allow registration with an email that is already in use.	1. Enter all information needed for registration with existing email address in the system 2. Press the "Signup" button.	Registration fails, and the system displays an error message stating "Email Address has already been taken.”

Table 6.6: Test Description for Register Module (Supervisor)

Test Module	Register Module: Supervisor		
Test Type	Unit Testing	**Test Date**	3/8/2024 - 4/8/2024
Test Strategy	Black Box Testing		
Test Description	New supervisor registration		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC2_01	Ensure that a new supervisor can successfully register with valid inputs for all required fields.	1. Admin required to register supervisor inside the admin account (click on user page) 2. Enter all information needed for registration	Registration is successful. The registered supervisor can successfully login into the system.

		3. Press the "Register" button.	
TC2_02	Validate that the registration process fails when the password does not meet complexity requirements (e.g., lacks special characters)	1. Enter all information needed for registration with the missing special characters 2. Press the " Register" button.	Registration fails, and the system displays an error message stating " Password must contain at least one uppercase letter, one lowercase letter, one number, and one special character."
TC2_03	Ensure that the system detects and displays an error when the email address format is invalid.	1. Enter all information needed for registration with invalid email format 2. Press the " Register" button.	Registration fails, and the system displays an error message stating "Email address is not a valid email address."
TC2_04	Verify that the registration process does not proceed when mandatory fields are left blank.	1. Enter information needed for registration with empty fields on the username field 2. Press the " Register" button.	Registration fails, and the system displays an error message stating "Username is required."
TC2_05	Ensure that the system does not allow registration with an email that is already in use.	1. Enter all information needed for registration with existing email address in the system 2. Press the " Register" button.	Registration fails, and the system displays an error message stating "Email Address has already been taken.”

Table 6.7: Test Description for Login Module

Test Module	Login Module: FYP Student/Supervisor/Admin		
Test Type	Unit Testing	**Test Date**	5/8/2024 - 6/8/2024
Test Strategy	Black Box Testing		
Test Description	Login into the system		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC3_01	Verify that users can successfully log in with valid credentials	1. Navigate to the login page 2. Enter a valid email and password 3. Click the "Login" button	Login successfully. User is redirected to the dashboard, based on their role in the system.
TC3_02	Verify that login fails with an incorrect password	1. Navigate to the login page 2. Enter a valid email and incorrect password 3. Click the "Login" button .	An error message of "Incorrect email or password" is displayed under the password form, and the user remains on the login page.
TC3_03	Verify that login fails with an unregistered email	1. Navigate to the login page. 2. Enter an unregistered email and a valid password. 3. Click the "Login" button.	Error message of "Incorrect email or password" is displayed under the email form, and the user remains on the login page.
TC3_04	Verify that login fails when required fields are left blank	1. Navigate to the login page. 2. Leave the email and password fields empty.	The message "Please fill out this field" is displayed, and the user remains on the

		3. Click the "Login" button.	login page.

Table 6.8: Test Description for Forgot Password

Test Module	Login Module: Forgot Password		
Test Type	Unit Testing	**Test Date**	7/8/2024 - 8/8/2024
Test Strategy	Black Box Testing		
Test Description	Forgot old account password		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC4_01	Verify that users able to choose option forgot password	1. Navigate to the login page 2. Click “Forgot password?” 3. Enter valid email address 4. Click button “Reset password”	Forgot password successfully. Message displayed “An email has been sent. Please check your inbox.” The default password will be sent in the email entered.
TC4_02	Verify that forgot password error if email address is left empty	1. Navigate to the login page 2. Click “Forgot password?” 3. Leave the email form empty 4. Click button “Reset password”	Message displayed “This email does not exist in our system.”

TC4_03	Verify that forgot password error if entered invalid email address	1. Navigate to the login page 2. Click “Forgot password?” 3. Enter invalid email address 4. Click button “Reset password”	Message displayed “This email does not exist in our system.”

Table 6.9: Test Description for Appointment Booking Module (FYP Student)

Test Module	Appointment Booking Module: FYP Student		
Test Type	Unit Testing	**Test Date**	9/8/2024 - 10/8/2024
Test Strategy	Black Box Testing		
Test Description	To make an appointment booking with their choice of supervisor		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC5_01	Verify the functionality of booking an appointment by selecting a date on a weekday.	1. Press “New Submission” button 2. Select a name for the appointment. 3. Choose a time slot and date, then press “Submit”.	Appointment is successfully booked, and confirmation is shown on the calendar.
TC5_02	Verify the system's behavior when booking an appointment by selecting a weekend date.	1. Press “New Submission” button 2. Select a name for the appointment. 3. Choose a time slot and date, then press “Submit”.	The system does not allow booking the appointment for the weekend date, an appropriate error message is displayed.
TC5_03	Verify booking an appointment without an	1. Press “New Submission” button	Error message “Appointment Title

	appointment title.	2. Leave appointment title blank, then press “Submit”.	cannot be blank” is displayed indicating required fields are missing.

Table 6.10: Test Description for Appointment Booking Module (Supervisor)

Test Module	Appointment Booking Module: Supervisor		
Test Type	Unit Testing	**Test Date**	11/8/2024-12/8/2024
Test Strategy	Black Box Testing		
Test Description	To accept or reject the student’s appointment booking request		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC6_01	Verify to accept the appointment request	1. Choose to accept option	Message displayed “Successfully accepted! This request has been accepted!”
TC6_02	Verify to reject the appointment request	1. Choose to reject option	Message displayed “Successfully rejected! This request has been rejected!”

Table 6.11: Test Description for Booking Supervisor Module (FYP Student)

Test Module	Booking Supervisor Module: FYP Student		
Test Type	Unit Testing	**Test Date**	13/8/2024-14/8/2024
Test Strategy	Black Box Testing		
Test	To book a supervisor for FYP student		

Description			
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC7_01	Verify that a valid PDF file with a complete proposal title is successfully submitted.	1. Enter proposal title 2. Upload a PDF file 3. Click the "Submit" button	The document is successfully submitted.
TC7_02	Verify that non-PDF file types are rejected during submission.	1. Enter proposal title 2. Upload other than a PDF file 3. Click the "Submit" button	Error message of "Only files with these extensions are allowed: pdf. " is displayed under the document file form
TC7_03	Verify that a blank proposal title, even with a valid PDF file, is rejected.	1. Upload a PDF file 2. Click the "Submit" button	Error message of " Proposal Title cannot be blank." is displayed under the submission title form
TC7_04	Verify that submissions with a filled-in proposal title but no document file are rejected.	1. Enter proposal title 2. Click the "Submit" button	Error message of " Document File cannot be blank." is displayed under the document file form

Table 6.12: Test Description for Booking Supervisor Module (Supervisor)

Test Module	Booking Supervisor Module: Supervisor		
Test Type	Unit Testing	**Test Date**	15/8/2024-16/8/2024
Test Strategy	Black Box Testing		
Test Description	To accept or reject the student's supervisor booking request		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC8_01	Verify to accept the	1. Choose to accept	Message displayed

	appointment request	option	“Successfully accepted! This request has been accepted!”
TC8_02	Verify to reject the appointment request	1. Choose to reject option	Message displayed “Successfully rejected! This request has been rejected!”

Table 6.13: Test Description for Document Submission Module (FYP Student)

Test Module	Document Submission Module: FYP Student		
Test Type	Unit Testing	**Test Date**	17/8/2024-18/8/2024
Test Strategy	Black Box Testing		
Test Description	To submit document on FYP student page		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC9_01	Verify that a valid PDF file with a complete submission title is successfully submitted.	1. Enter submission title 2. Upload a PDF file 3. Click the "Submit" button	The document is successfully submitted.
TC9_02	Verify that non-PDF file types are rejected during submission.	1. Enter submission title 2. Upload other than a PDF file 3. Click the "Submit" button	Error message of "Only files with these extensions are allowed: pdf. " is displayed under the document file form
TC9_03	Verify that a blank submission title results in rejection, even if a valid PDF file is uploaded.	1. Upload a PDF file 2. Click the "Submit" button	Error message of " Submission Title cannot be blank." is displayed under the

			submission title form
TC9_04	Verify that submissions with a filled-in submission title but no document file are rejected.	1. Enter submission title 2. Click the "Submit" button	Error message of " Document File cannot be blank." is displayed under the document file form

Table 6.14: Test Description for Document Submission Module (Supervisor)

Test Module	Document Submission Module: Supervisor		
Test Type	Unit Testing	**Test Date**	19/8/2024-20/8/2024
Test Strategy	Black Box Testing		
Test Description	To accept or reject the student’s document submission		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC10_01	Verify to accept the appointment request	1. Choose to accept option	Message displayed “Successfully accepted! This request has been accepted!”
TC10_02	Verify to reject the appointment request	1. Choose to reject option	Message displayed “Successfully rejected! This request has been rejected!”

Table 6.15: Test Description for Admin Module (Department)

Test Module	Admin Module: Department		
Test Type	Unit Testing	**Test Date**	21/8/2024-22/8/2024
Test Strategy	Black Box Testing		
Test Description	To add and view department, select a status (either inactive or active)		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC11_01	Verify to add new department with all valid input data.	1. Enter department name and choose status of department (active or inactive) 2. Click the "Save" button	A new department was successfully added.
TC11_02	Verify to add new department with department name left empty.	1. Leave department name empty and choose status of department (active or inactive) 2. Click the "Save" button	Error message of “Department Desc cannot be blank.” is displayed under the department name form
TC11_03	Verify to add new department status left unchosen.	1. Enter department name and leave status of department unchosen 2. Click the "Save" button	Error message of “Status ID cannot be blank.” is displayed under the department status form

Table 6.16: Test Description for Admin Module (Position)

Test Module	Admin Module: Position		
Test Type	Unit Testing	**Test Date**	23/8/2024-24/8/2024
Test Strategy	Black Box Testing		
Test Description	To add and view a position, select a status (either inactive or active)		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC12_01	Verify to add new position with all valid input data.	1. Enter position name and choose status of position (active or inactive) 2. Click the "Save" button	A new position was successfully added.
TC12_02	Verify to add new position with position name left empty.	1. Leave position name empty and choose status of position (active or inactive) 2. Click the "Save" button	Error message of “Position Desc cannot be blank.” is displayed under the position name form
TC12_03	Verify to add new position status left unchosen.	1. Enter position name and leave status of position unchosen 2. Click the "Save" button	Error message of “Status ID cannot be blank.” is displayed under the position status form

Table 6.17: Test Description for Admin Module (Report)

Test Module	Admin Module: Report		
Test Type	Unit Testing	**Test Date**	25/8/2024-26/8/2024
Test Strategy	Black Box Testing		
Test Description	To view report		
Test Case ID	**Test Case Description**	**Test Step**	**Expected Result**
TC13_01	Verify that the report can be exported as EXCEL file	1. Click "Excel" button to print	The report was successfully downloaded to the desktop.

6.4.2 Test Data

The test data is used to support the information of detailed test descriptions. Tables 6.10 through 6.16 provide examples of test data for each module, illustrating various scenarios and conditions under which the application will be evaluated. This data helps ensure comprehensive testing coverage and verifies that each module functions correctly under different conditions.

Table 6.18: Example of Test Data for Login Module (FYP Student)

Test Data ID	Username	Password	Description
TD01	b032110268@student.utem.edu.my	qaZ@123	Valid username and password.
TD02	b032110268@student.utem.edu.my	Password123	Valid username and incorrect password.

TD03	ainamarinaa@gmail.com	ainamarin4	Unregistered email into the system
TD04		qaZ@123	Empty username and a valid password.

Table 6.19: Example of Test Data for Login Module (Supervisor)

Test Data ID	Username	Password	Description
TD05	razak@utem.edu.my	qaZ@123	Valid username and password.
TD06	razak@utem.edu.my	Password123	Valid username and incorrect password.
TD07	ainamarin45@gmail.com	ainamarin4	Unregistered email into the system
TD08		qaZ@123	Empty username and a valid password.

Table 6.20: Example of Test Data for Login Module (Admin)

Test Data ID	Username	Password	Description
TD09	admin@fyp.my	admin_123	Valid username and password.

TD10	admin@fyp.my	123	Valid username and incorrect password.
TD11	admin123@fyp.com	admin12	Unregistered email into the system
TD12		admin_123	Empty username and a valid password.

Table 6.21: Example of Test Data for Register Module (Supervisor)

Test Data ID	Full Name	Username	Email	Password	Description
TD13	TS. MUHAMMAD SUHAIZAN BIN SULONG	TS. MUHAMMAD SUHAIZAN	suhaizan@utem.edu.my	qaZ@123	Valid registration with all correct details.
TD14	TS. MUHAMMAD SUHAIZAN BIN SULONG	TS. MUHAMMAD SUHAIZAN	suhaizan@utem.edu.my	qaZ@1234	Valid full name, username and email but mismatched passwords.
TD15	TS. MUHAMMAD	TS. MUHAMMAD	suhaizan@utem.edu.my	qaZ123	Valid username with a password

	SUHAIZAN BIN SULONG	SUHAIZAN			missing special characters.
TD16	TS. MUHAMMAD SUHAIZAN BIN SULONG	TS. MUHAMMAD SUHAIZAN	suhaizan	qaZ@123	Valid full name, password and username but invalid email address format.
TD17	TS. MUHAMMAD SUHAIZAN BIN SULONG		suhaizan@utem.edu.my	qaZ@123	Registration information is incomplete
TD18	NOOR AZILAH BINTI DRAMAN @MUDA	NOOR AZILAH	azilah@utem.edu.my	qaZ@123	Valid registration with an email that already exists in the system.

Table 6.22: Example of Test Data for Register Module (FYP Student)

Test Data ID	Full Name	Username	Email	Password	Description
TD19	SITI AISYAH BINTI	SITI AISYAH	b0321115421@student.utem	Siti_@isyah1	Valid registration with all

	FARID		.edu.my		correct details.
TD20	SITI AISYAH BINTI FARID	SITI AISYAH	b0321115421 @student.utem .edu.my	Siti_@isyah	Valid full name, username and email but mismatched passwords.
TD21	SITI AISYAH BINTI FARID	SITI AISYAH	b0321115421 @student.utem .edu.my	Siti_aisyah	Valid username with a password missing special characters.
TD22	SITI AISYAH BINTI FARID	SITI AISYAH	b0321115421	Siti_@isyah1	Valid full name, password and username but invalid email address format.
TD23	SITI AISYAH BINTI FARID		b0321115421 @student.utem .edu.my	Siti_@isyah1	Registration information is incomplete
TD24	NURUL AINA MARINA BINTI MANSOR	NURUL AINA MARINA	b032110268@ student.utem.e du.my	qaZ@123	Valid registration with an email that already exists in the system.

Table 6.23: Example of Test Data for Document Submission Module (FYP Student)

Test Data ID	Submission Title	Document Title	Description
TD25	Proposal FYP System	PDF	Valid PDF file
TD26	Proposal FYP System	DOCX	Non-PDF file type, should be rejected.
TD27		PDF	Submission title left blank with valid PDF file.
TD28	Proposal FYP System		File is left blank with submission title filled in.

Table 6.24: Example of Test Data for Appointment Booking Module (FYP Student)

Test Data ID	Appointment Title	Appointment Date	Slot	Description
TD29	Proposal meeting discussion	12/09/2024 (Thursday)	11:00AM-12:00PM	All valid information with appointment date on weekdays
TD30	Proposal meeting	14/09/2024	11:00AM-12:00PM	All valid information with

	discussion	(Saturday)		appointment date on weekends
TD31		12/09/2024 (Thursday)	11:00AM-12:00PM	Appointment title is left blank with other information filled in

Table 6.25: Example of Test Data for Booking Supervisor Module (FYP Student)

Test Data ID	Proposal Title	Document File	Description
TD32	Proposal FYP System	PDF	Valid PDF file with valid proposal title
TD33	Proposal FYP System	DOCX	Non-PDF file type, should be rejected.
TD34		PDF	Submission title left blank with valid PDF file.
TD35	Proposal FYP System		File is left blank with submission title filled in.

Table 6.26: Example of Test Data for Forgot Password

Test Data ID	Email	Description
TD36	b032110268@student.utem.edu.my	Valid email to change new password
TD37		The email field is left empty in the form.
TD38	nabihah12@gmail.com	The email entered is unregistered in the form.

Table 6.27: Test Data for Appointment Booking Module (Supervisor)

Test Data ID	Status	Description
TD39	Accept	Accept student’s appointment booking request
TD40	Reject	Reject student’s appointment booking request

Table 6.28: Test Data for Booking Supervisor Module (Supervisor)

Test Data ID	Status	Description
TD41	Accept	Accept student’s booking supervisor request
TD42	Reject	Reject student’s booking supervisor request

Table 6.29: Example of Test Data for Document Submission Module (Supervisor)

Test Data ID	Status	Description
TD43	Accept	Accept student’s document submission request
TD44	Reject	Reject student’s document submission request

Table 6.30: Example of Test Data for Admin Module (Department)

Test Data ID	Department Name	Status	Description
TD45	Engineering	Active	A new department was successfully added into the system.

TD46		Active	Department name is left blank.
TD47	Engineering		Status is left blank

Table 6.31: Example of Test Data for Admin Module (Position)

Test Data ID	Position Name	Status	Description
TD48	Senior Lecturer	Active	A new position was successfully added into the system.
TD49		Active	Position name is left blank.
TD50	Senior Lecturer		Status is left blank

Table 6.32: Example of Test Data for Admin Module (Position)

Test Data ID	File	Description
TD51	Excel	Press the "Excel" button to download the system report in Excel format.

6.5 Test Result and Analysis

Test results are crucial in testing because they reveal the actual outcomes and compare them with the expected results detailed in the test description to determine the testing status. Tables 6.17 present the identified test cases along with their actual results and indicate whether they passed or failed.

Table 6.33: Test Result for Register Module (FYP Student)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC1_01	TD19	Registration is successful, and the user is redirected to the login page.	Registration is successful, and the user is redirected to the login page.	Pass
TC1_02	TD20	Registration fails, and the system displays an error message stating "Your confirmation password does not match."	Registration fails, and the system displays an error message stating "Your confirmation password does not match."	Pass
TC1_03	TD21	Registration fails, and the system displays an error message stating "Password must contain at least one special character."	Registration fails, and the system displays an error message stating "Password must contain at least one special character."	Pass
TC1_04	TD22	Registration fails, and the system displays an error message stating "Please enter a valid email address."	Registration fails, and the system displays an error message stating "Please enter a valid email address."	Pass
TC1_05	TD23	Registration fails, and the system displays an error message stating	Registration fails, and the system displays an error message stating	Pass

		"Username is required."	"Username is required."	
TC1_06	TD24	Registration fails, and the system displays an error message stating "Email Address has already been taken.”	Registration fails, and the system displays an error message stating "Email Address has already been taken.”	Pass

Table 6.34: Test Result for Register Module (Supervisor)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC2_01	TD13	Registration is successful. The registered supervisor can successfully login into the system.	Registration is successful. The registered supervisor can successfully login into the system.	Pass
TC2_02	TD15	Registration fails, and the system displays an error message stating " Password must contain at least one uppercase letter, one lowercase letter, one number, and one special character."	Registration fails, and the system displays an error message stating " Password must contain at least one uppercase letter, one lowercase letter, one number, and one special character."	Pass
TC2_03	TD16	Registration fails, and the system displays an error message stating "Email address is not a valid email address."	Registration fails, and the system displays an error message stating "Email address is not a valid email address."	Pass

TC2_04	TD17	Registration fails, and the system displays an error message stating "Username is required."	Registration fails, and the system displays an error message stating "Username is required."	Pass
TC2_05	TD18	Registration fails, and the system displays an error message stating "Email Address has already been taken.”	Registration fails, and the system displays an error message stating "Email Address has already been taken.”	Pass

Table 6.35: Test Result for Login Module

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC3_01	TD01, TD05, TD09	Login successfully. User is redirected to the dashboard, based on their role in the system.	Login successfully. User is redirected to the dashboard, based on their role in the system.	Pass
TC3_02	TD02, TD06, TD10	An error message of "Incorrect email or password" is displayed under the password form, and the user remains on the login page.	An error message of "Incorrect email or password" is displayed under the password form, and the user remains on the login page.	Pass
TC3_03	TD03, TD07, TD11	Error message of "Incorrect email or password" is displayed under the email form, and the user remains on the login page.	Error message of "Incorrect email or password" is displayed under the email form, and the user remains on the login page.	Pass

TC3_04	TD04, TD08, TD12	The message "Please fill out this field" is displayed, and the user remains on the login page.	The message "Please fill out this field" is displayed, and the user remains on the login page.	Pass

Table 6.36: Test Result for Forgot Password Module

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC4_01	TD36	Forgot password successfully. Message displayed “An email has been sent. Please check your inbox.” The default password will be sent in the email entered.	Forgot password successfully. Message displayed “An email has been sent. Please check your inbox.” The default password will be sent in the email entered.	Pass
TC4_02	TD37	Message displayed “This email does not exist in our system.”	Message displayed “This email does not exist in our system.”	Pass
TC4_03	TD38	Message displayed “This email does not exist in our system.”	Message displayed “This email does not exist in our system.”	Pass

Table 6.37: Test Result for Appointment Booking Module (FYP Student)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC5_01	TD29	Appointment is successfully booked, and confirmation is shown on the calendar.	Appointment is successfully booked, and confirmation is shown on the calendar.	Pass
TC5_02	TD30	The system does not allow booking the	The system does not allow booking the	Pass

		appointment for the weekend date, an appropriate error message is displayed.	appointment for the weekend date, an appropriate error message is displayed.	
TC5_03	TD31	Error message “Appointment Title cannot be blank” is displayed indicating required fields are missing.	Error message “Appointment Title cannot be blank” is displayed indicating required fields are missing.	Pass

Table 6.38: Test Result for Appointment Booking Module (Supervisor)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC6_01	TD39	Message displayed “Successfully accepted! This request has been accepted!”	Message displayed “Successfully accepted! This request has been accepted!”	Pass
TC6_02	TD40	Message displayed “Successfully rejected! This request has been rejected!”	Message displayed “Successfully rejected! This request has been rejected!”	Pass

Table 6.39: Test Result for Booking Supervisor Module (FYP Student)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC7_01	TD32	The document is successfully submitted.	The document is successfully submitted.	Pass
TC7_02	TD33	Error message of "Only files with these extensions are allowed: pdf. " is displayed	Error message of "Only files with these extensions are allowed: pdf. " is displayed	Pass

		under the document file form	under the document file form	
TC7_03	TD34	Error message of " Proposal Title cannot be blank." is displayed under the submission title form	Error message of " Proposal Title cannot be blank." is displayed under the submission title form	Pass
TC7_04	TD35	Error message of " Document File cannot be blank." is displayed under the document file form	Error message of " Document File cannot be blank." is displayed under the document file form	Pass

Table 6.40: Test Result for Booking Supervisor Module (FYP Student)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC8_01	TD41	Message displayed “Successfully accepted! This request has been accepted!”	Message displayed “Successfully accepted! This request has been accepted!”	Pass
TC8_02	TD42	Message displayed “Successfully rejected! This request has been rejected!”	Message displayed “Successfully rejected! This request has been rejected!”	Pass

Table 6.41: Test Result for Document Submission Module (FYP Student)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC9_01	TD25	The document is successfully submitted.	The document is successfully submitted.	Pass
TC9_02	TD26	Error message of "Only files with these	Error message of "Only files with these	Pass

		extensions are allowed: pdf. " is displayed under the document file form	extensions are allowed: pdf. " is displayed under the document file form	
TC9_03	TD27	Error message of " Submission Title cannot be blank." is displayed under the submission title form	Error message of " Submission Title cannot be blank." is displayed under the submission title form	Pass
TC9_04	TD28	Error message of " Document File cannot be blank." is displayed under the document file form	Error message of " Document File cannot be blank." is displayed under the document file form	Pass

Table 6.42: Test Result for Document Submission Module (Supervisor)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC10_01	TD43	Message displayed “Successfully accepted! This request has been accepted!”	Message displayed “Successfully accepted! This request has been accepted!”	Pass
TC10_02	TD44	Message displayed “Successfully rejected! This request has been rejected!”	Message displayed “Successfully rejected! This request has been rejected!”	Pass

Table 6.43: Test Result for Admin Module (Department)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC11_01	TD45	A new department was successfully added.	A new department was successfully added.	Pass

TC11_02	TD46	Error message of “Department Desc cannot be blank.” is displayed under the department name form	Error message of “Department Desc cannot be blank.” is displayed under the department name form	Pass
TC11_03	TD47	Error message of “Status ID cannot be blank.” is displayed under the department status form	Error message of “Status ID cannot be blank.” is displayed under the department status form	Pass

Table 6.44: Test Result for Admin Module (Position)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC12_01	TD48	A new position was successfully added.	A new position was successfully added.	Pass
TC12_02	TD49	Error message of “Position Desc cannot be blank.” is displayed under the position name form	Error message of “Position Desc cannot be blank.” is displayed under the position name form	Pass
TC12_03	TD50	Error message of “Status ID cannot be blank.” is displayed under the position status form	Error message of “Status ID cannot be blank.” is displayed under the position status form	Pass

Table 6.45: Test Result for Admin Module (Report)

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC13_01	TD51	The report was successfully	The report was successfully	Pass

		downloaded to the desktop.	downloaded to the desktop.	

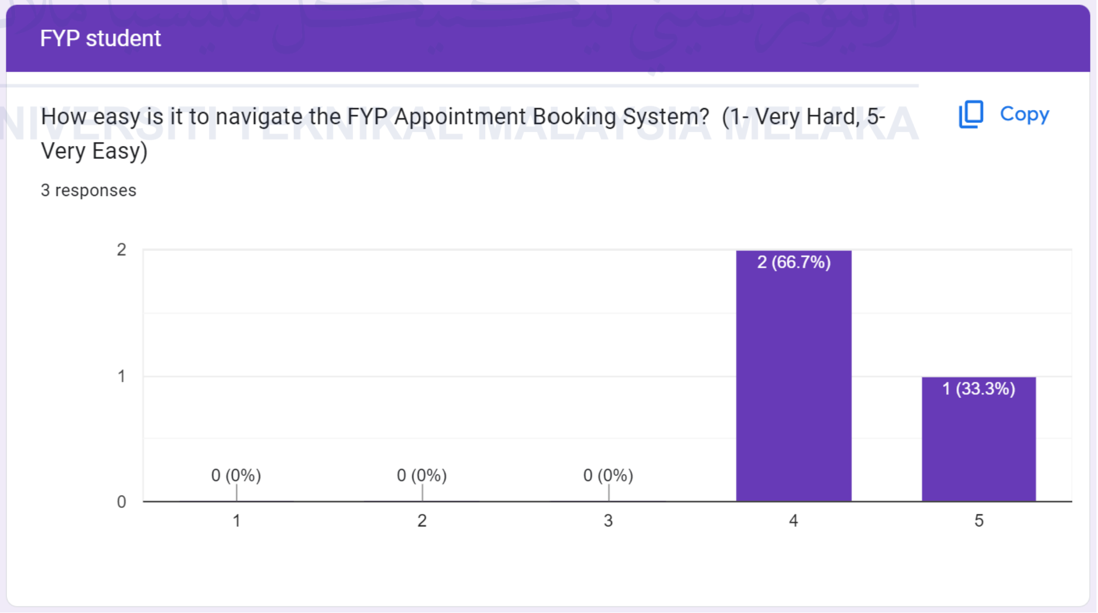
6.5.1 User Acceptance Testing

User Acceptance Testing (UAT) is essential to ensure that the FYP Appointment Booking System aligns with the needs and expectations of its users. The main goal of UAT is to confirm that the system fulfils user requirements and is prepared for deployment. The feedback collected during the UAT phase comes from users who have tested the system.

Link to feedback google form: https://forms.gle/cYSjxXKobMZimsNy9

i) Feedback from FYP Students

A total of 3 students have provided feedback on the FYP Appointment Booking System.

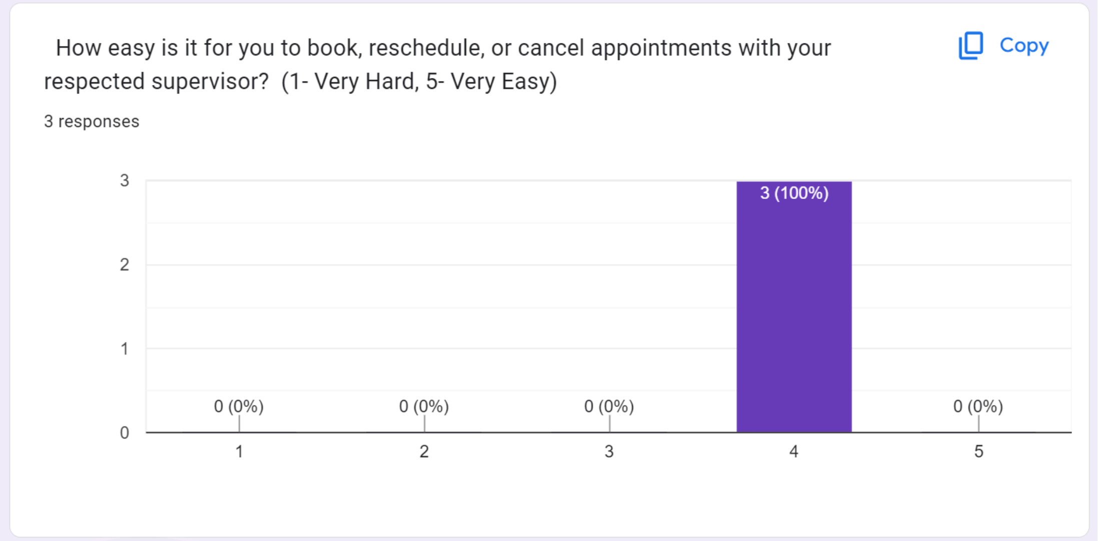


FYP student

How easy is it to navigate the FYP Appointment Booking System? (1- Very Hard, 5- Very Easy)

3 responses

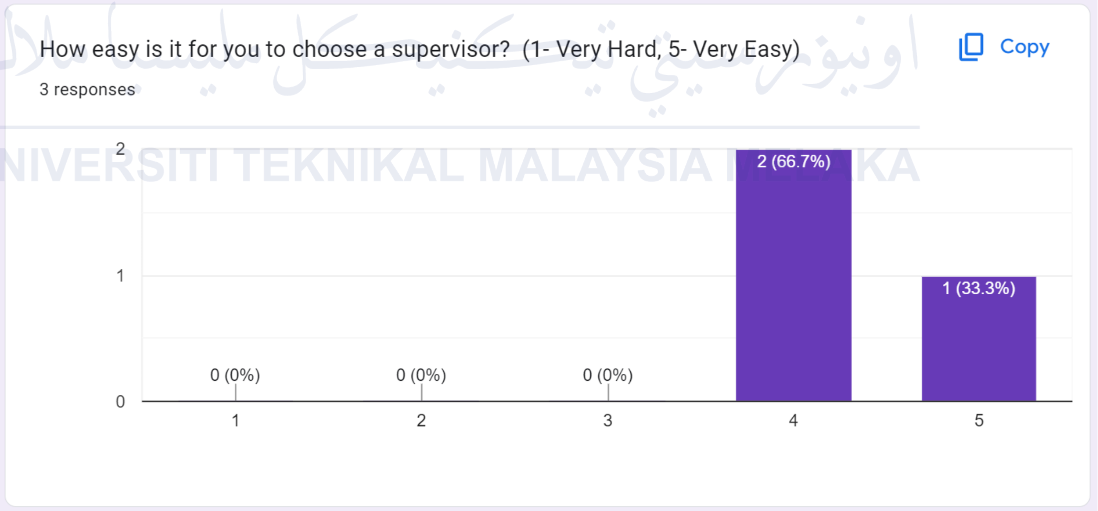
Figure 6.2: Feedback Navigation of System for FYP Student



How easy is it for you to book, reschedule, or cancel appointments with your respected supervisor? (1- Very Hard, 5- Very Easy)

3 responses

Figure 6.3: Feedback Appointment Booking for FYP Student



How easy is it for you to choose a supervisor? (1- Very Hard, 5- Very Easy)

3 responses

Figure 6.4: Feedback Booking Supervisor for FYP Student

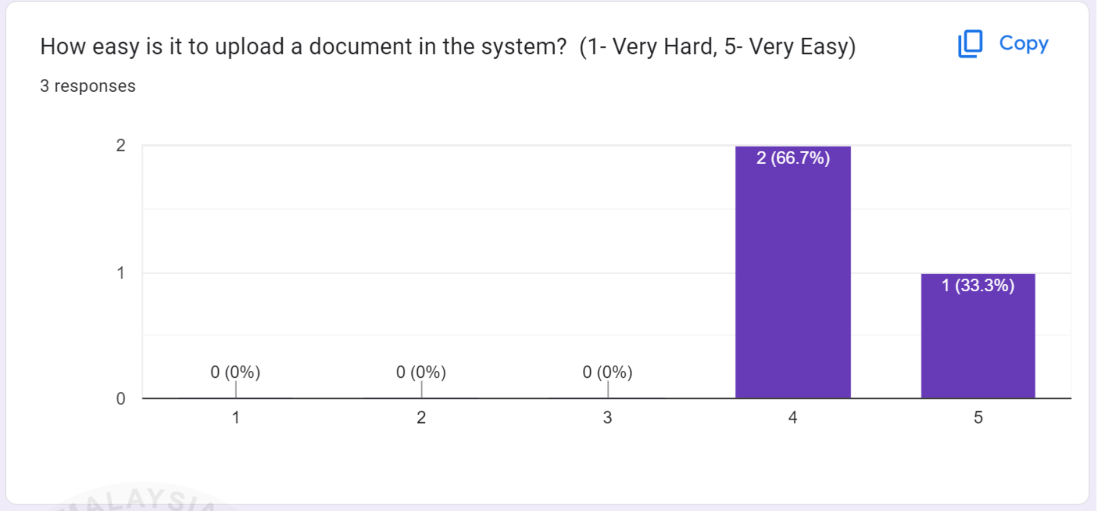


Figure 6.5: Feedback Document Submission for FYP Student

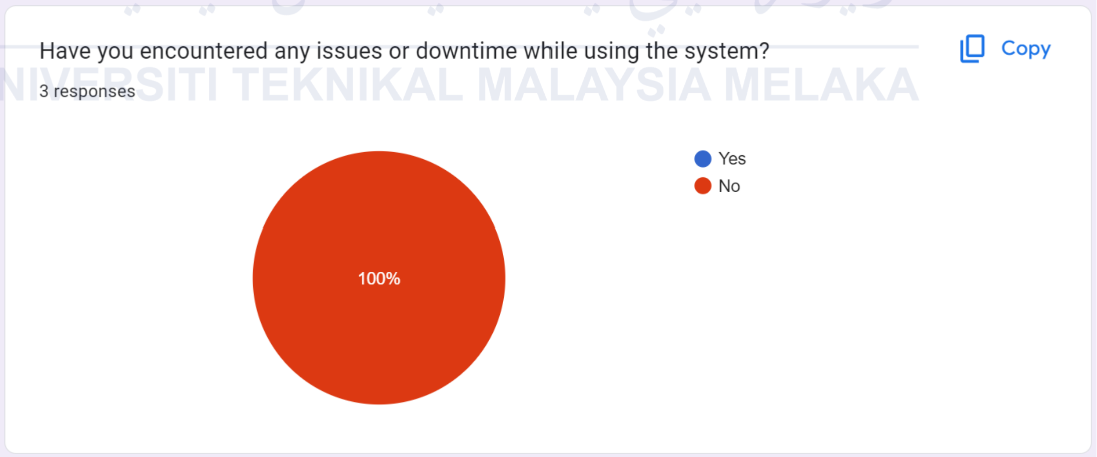
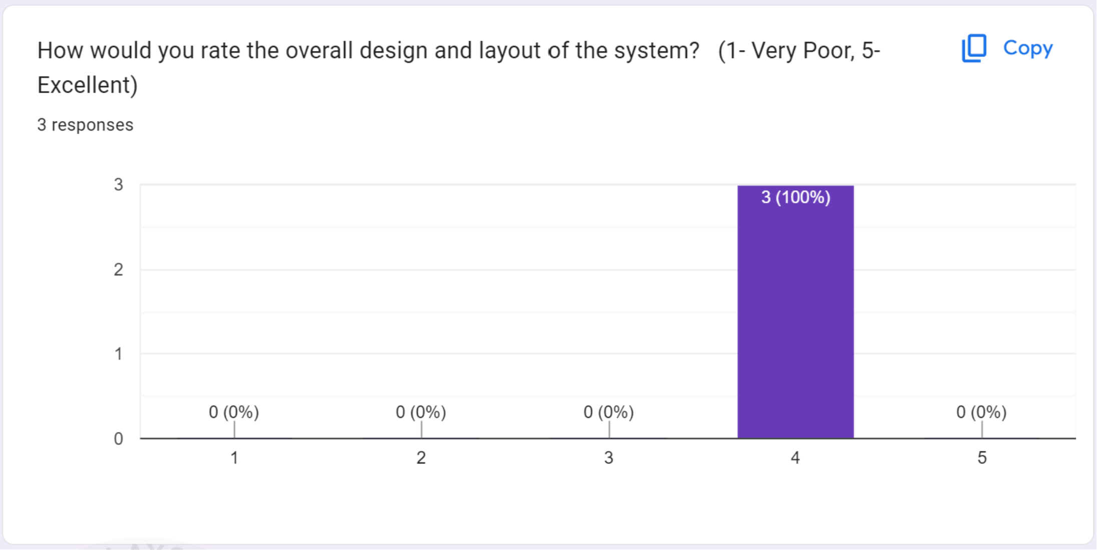


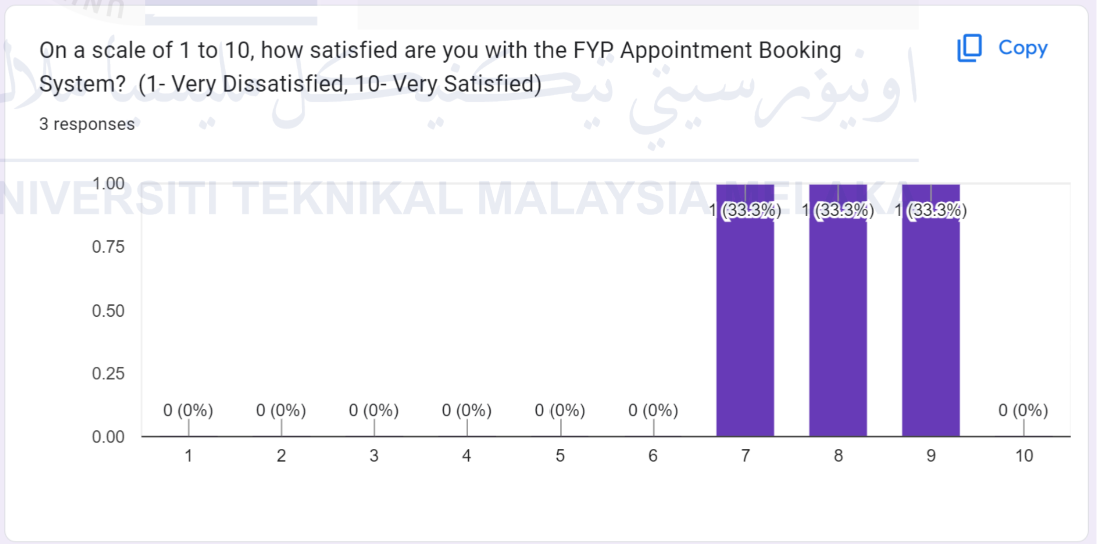
Figure 6.6: Feedback Downtime System for FYP Student



How would you rate the overall design and layout of the system? (1- Very Poor, 5- Excellent)

3 responses

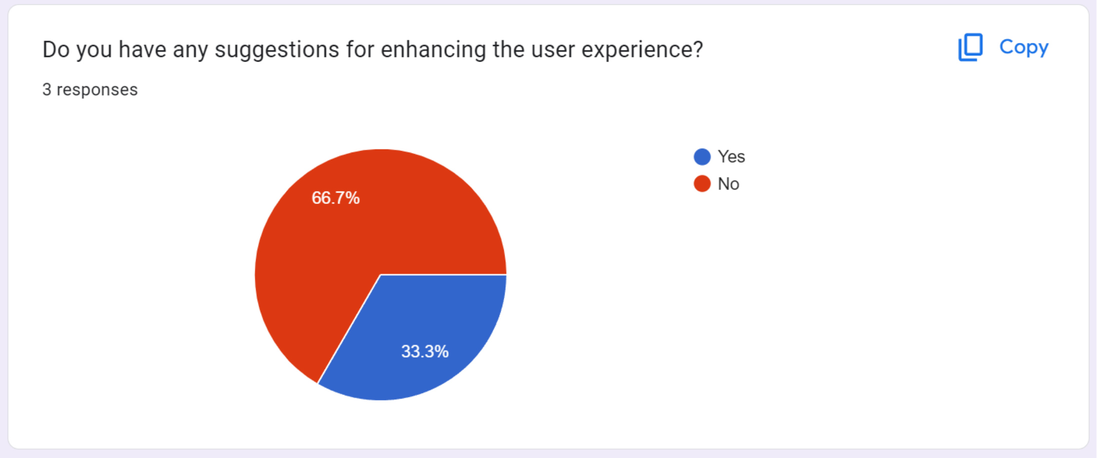
Figure 6.7: Feedback Design of System



On a scale of 1 to 10, how satisfied are you with the FYP Appointment Booking System? (1- Very Dissatisfied, 10- Very Satisfied)

3 responses

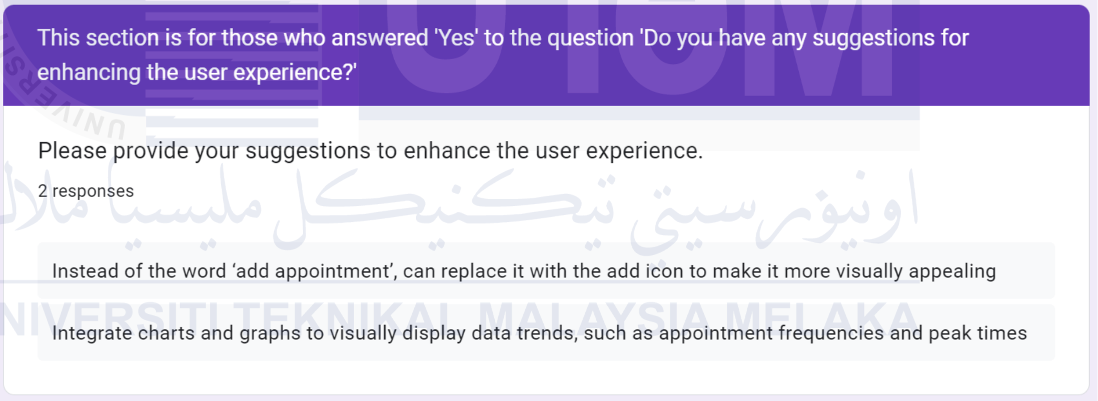
Figure 6.8: Feedback Satisfaction of System



Do you have any suggestions for enhancing the user experience?

3 responses

Figure 6.9: Feedback Suggestion of System



This section is for those who answered 'Yes' to the question 'Do you have any suggestions for enhancing the user experience?'

Please provide your suggestions to enhance the user experience.

2 responses

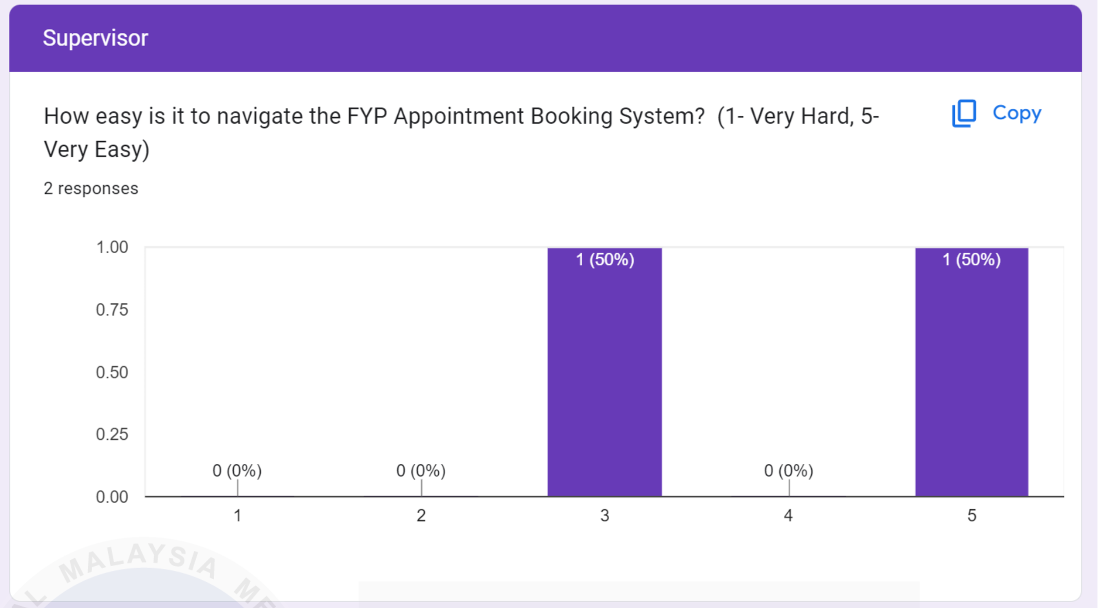
Instead of the word ‘add appointment’, can replace it with the add icon to make it more visually appealing

Integrate charts and graphs to visually display data trends, such as appointment frequencies and peak times

Figure 6.10: Suggestions to enhance User Experience from FYP Student

ii) Feedback from UTeM Supervisor

A total of 2 supervisors who have provided feedback on the FYP Appointment Booking System.

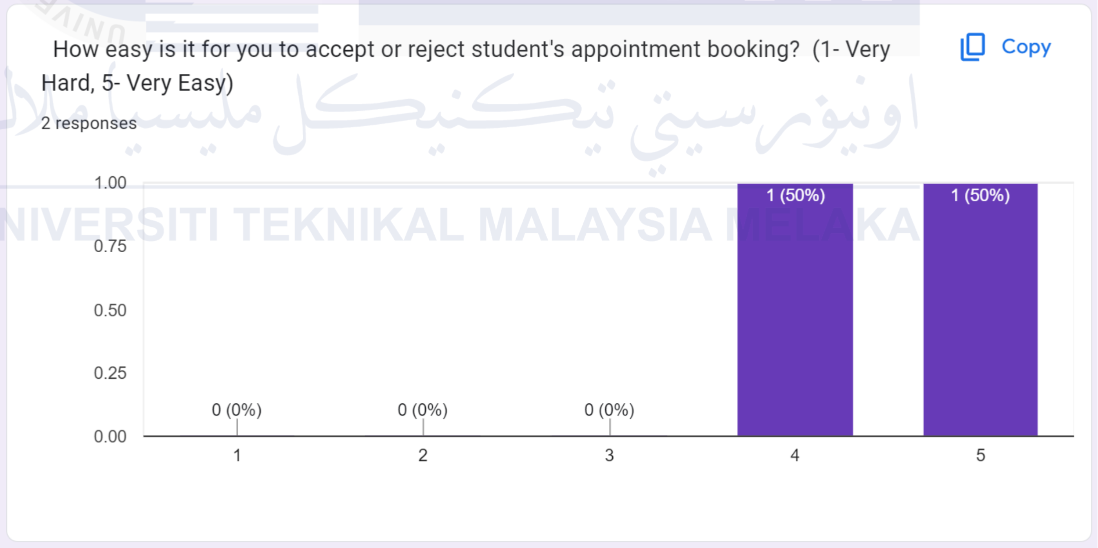


Supervisor

How easy is it to navigate the FYP Appointment Booking System? (1- Very Hard, 5- Very Easy)

2 responses

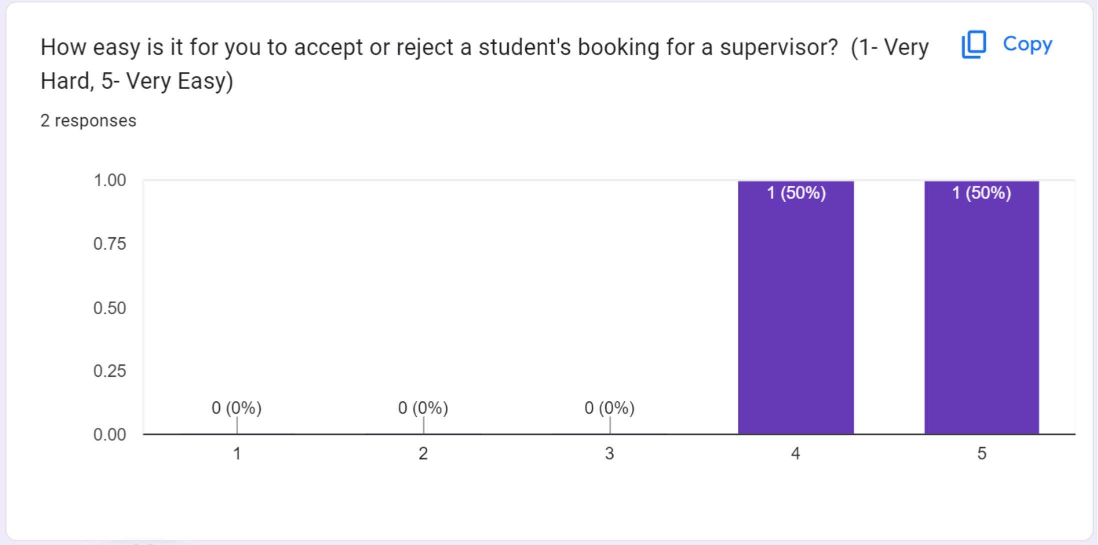
Figure 6.11: Feedback Navigation of System for Supervisor



How easy is it for you to accept or reject student's appointment booking? (1- Very Hard, 5- Very Easy)

2 responses

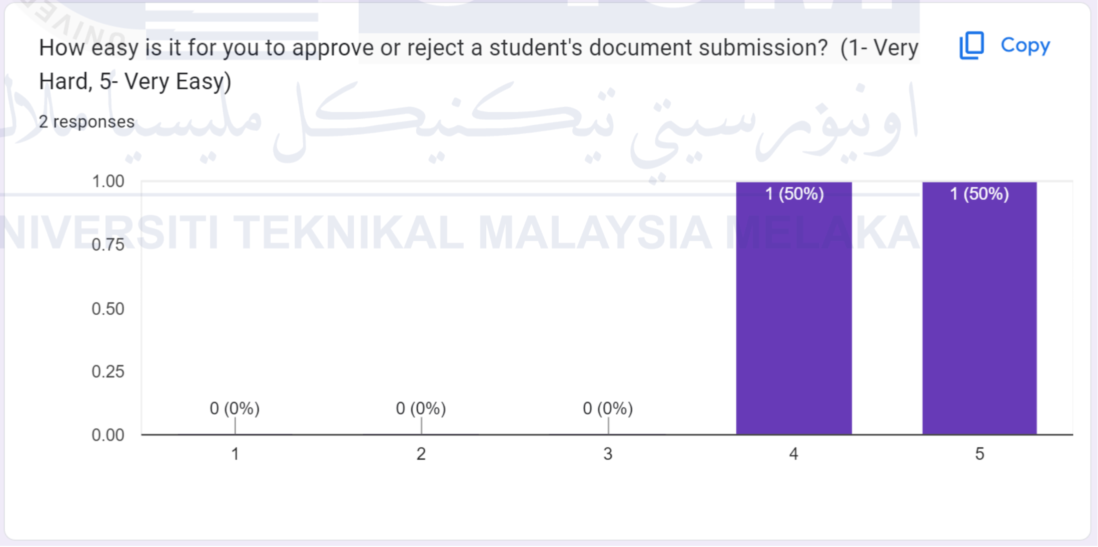
Figure 6.12: Feedback Status Appointment of System for Supervisor



How easy is it for you to accept or reject a student's booking for a supervisor? (1- Very Hard, 5- Very Easy)

2 responses

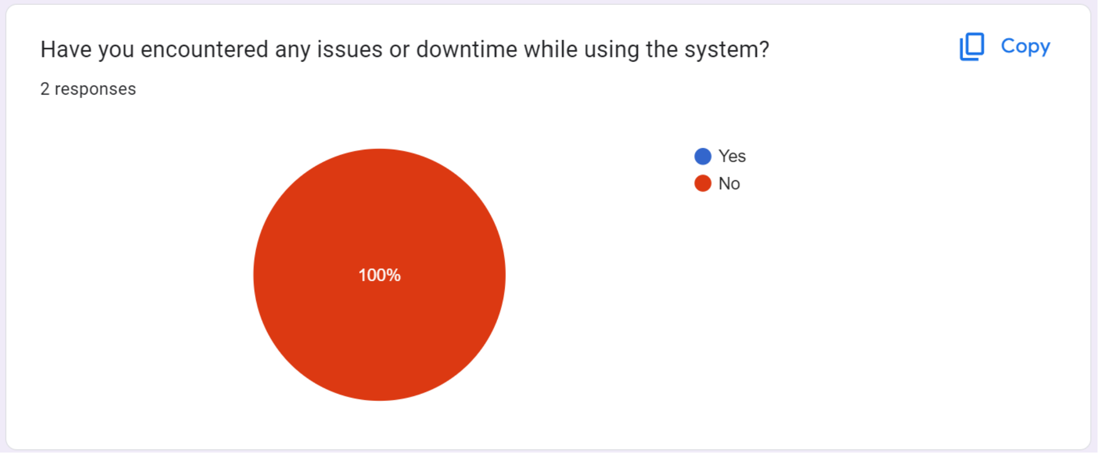
Figure 6.13: Feedback Status Booking Supervisor of System for Supervisor



How easy is it for you to approve or reject a student's document submission? (1- Very Hard, 5- Very Easy)

2 responses

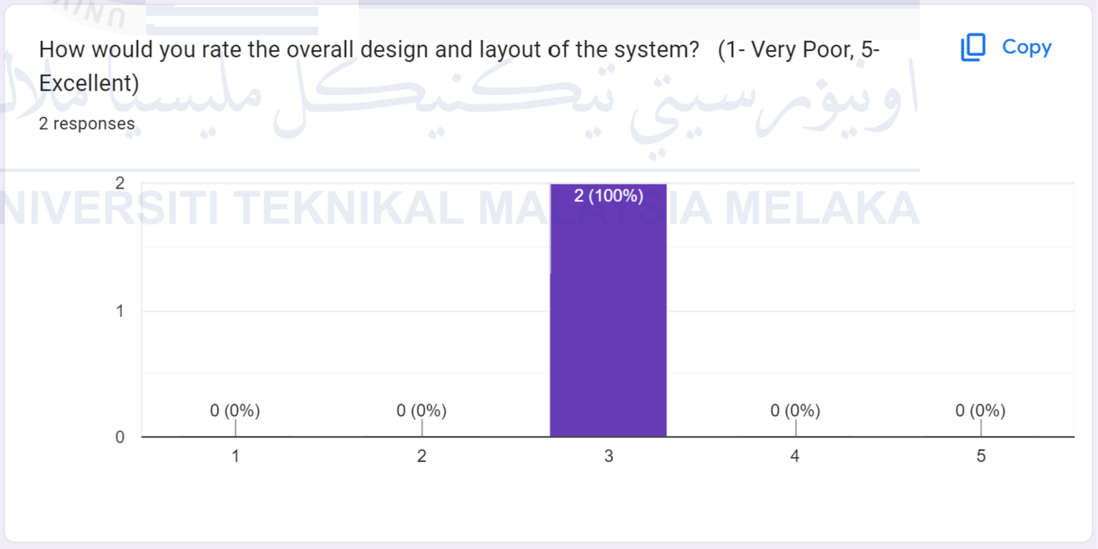
Figure 6.14: Feedback Status Document Submission of System for Supervisor



Have you encountered any issues or downtime while using the system?

2 responses

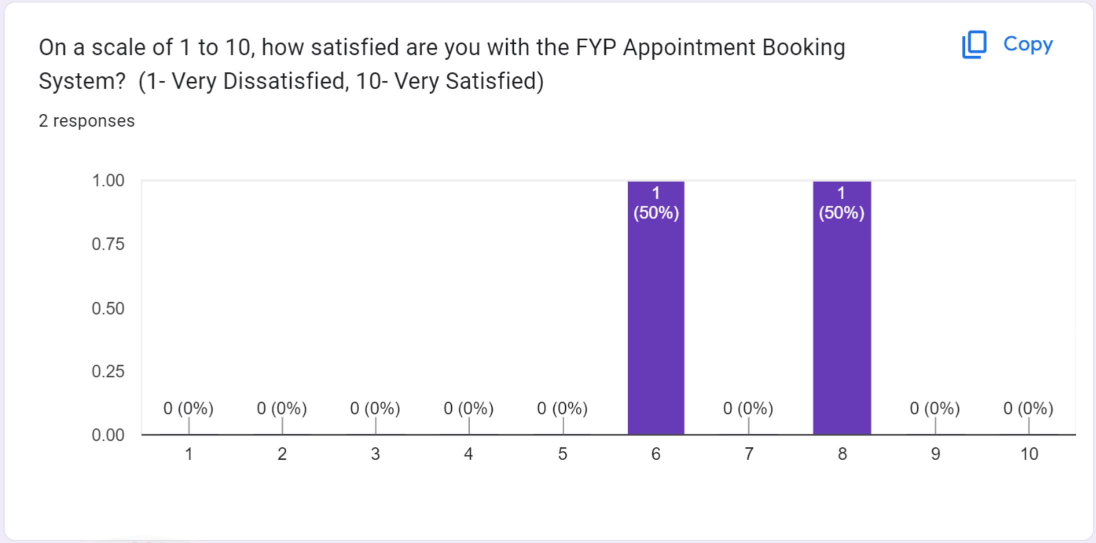
Figure 6.15: Feedback Downtime System for Supervisor



How would you rate the overall design and layout of the system?   (1- Very Poor, 5- Excellent)

2 responses

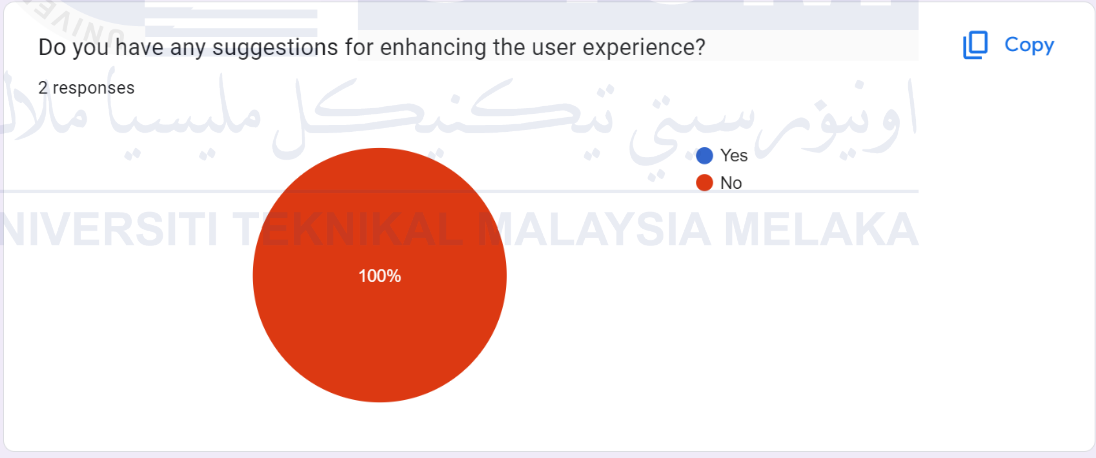
Figure 6.16: Feedback Design of System



On a scale of 1 to 10, how satisfied are you with the FYP Appointment Booking System? (1- Very Dissatisfied, 10- Very Satisfied)

2 responses

Figure 6.17: Feedback Satisfaction of System



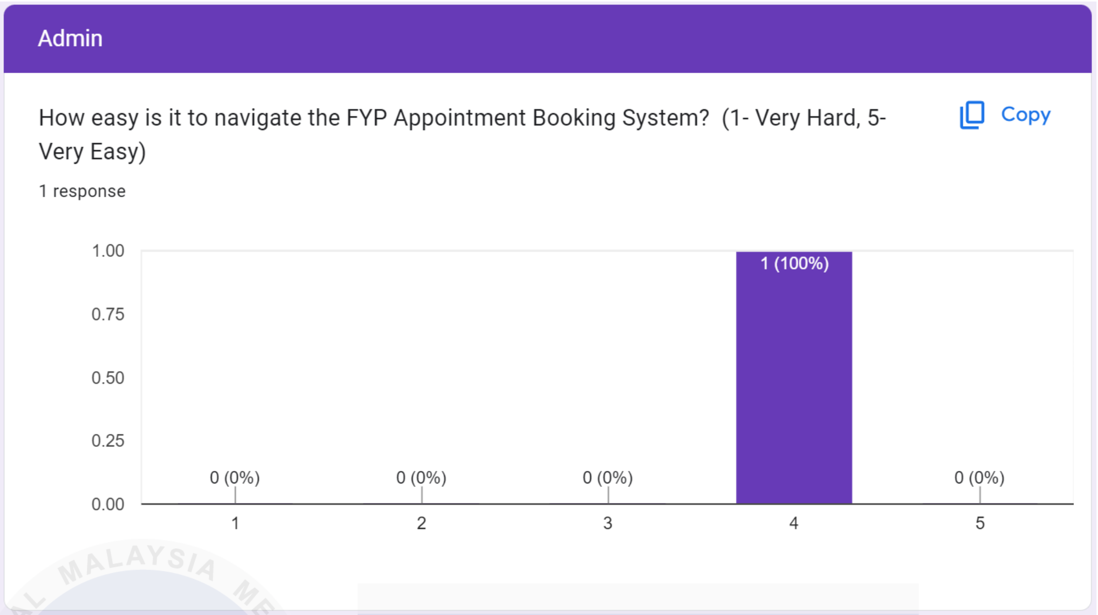
Do you have any suggestions for enhancing the user experience?

2 responses

Figure 6.18: Feedback Suggestion of System

iii) Feedback from UTeM Admin

A total of 1 admin who have provided feedback on the FYP Appointment Booking System.

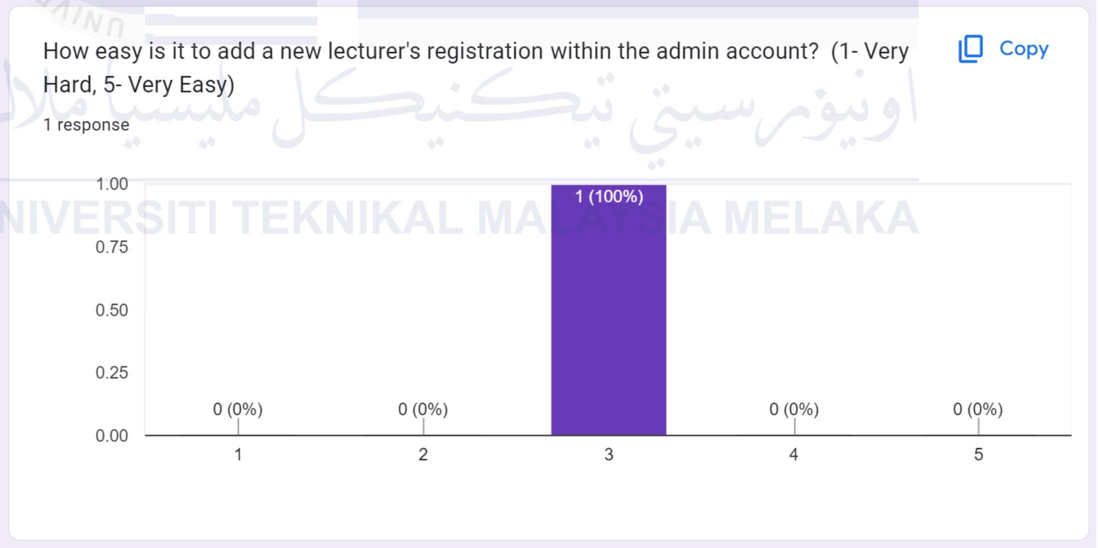


Admin

How easy is it to navigate the FYP Appointment Booking System? (1- Very Hard, 5- Very Easy)

1 response

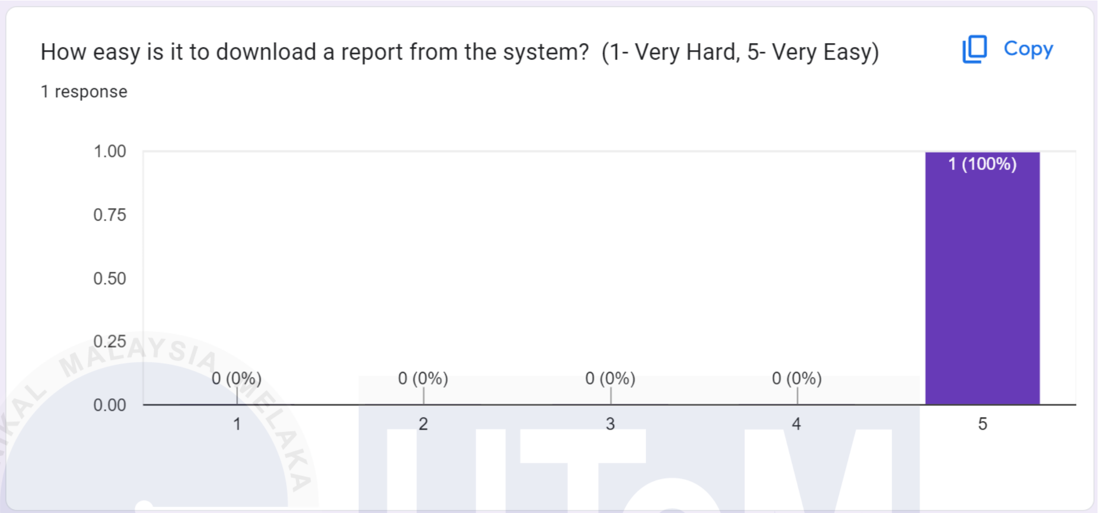
Figure 6.19: Feedback Navigation of System for Admin



How easy is it to add a new lecturer's registration within the admin account? (1- Very Hard, 5- Very Easy)

1 response

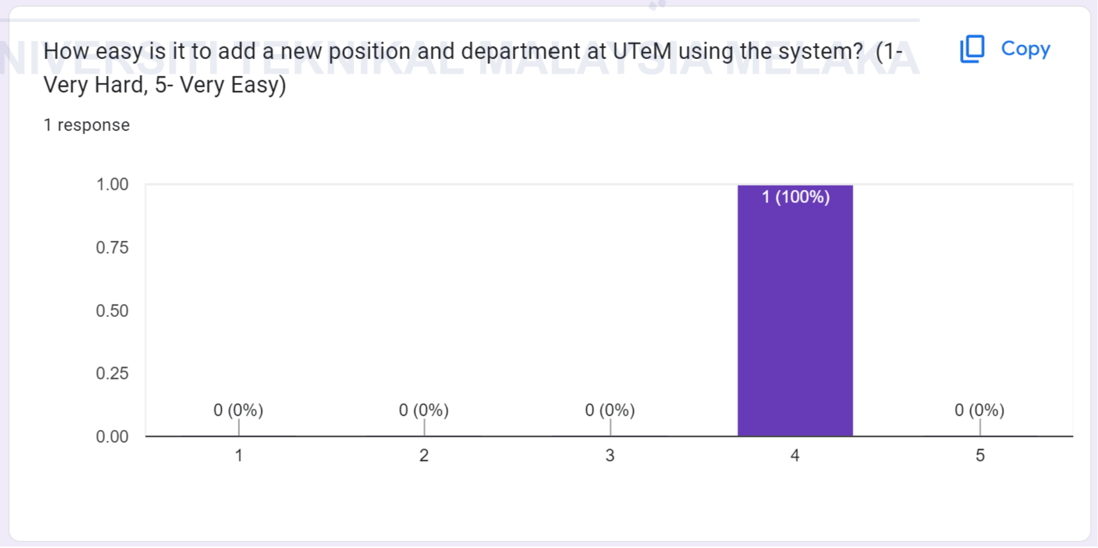
Figure 6.20: Feedback Registration Lecturer in System



How easy is it to download a report from the system? (1- Very Hard, 5- Very Easy)

1 response

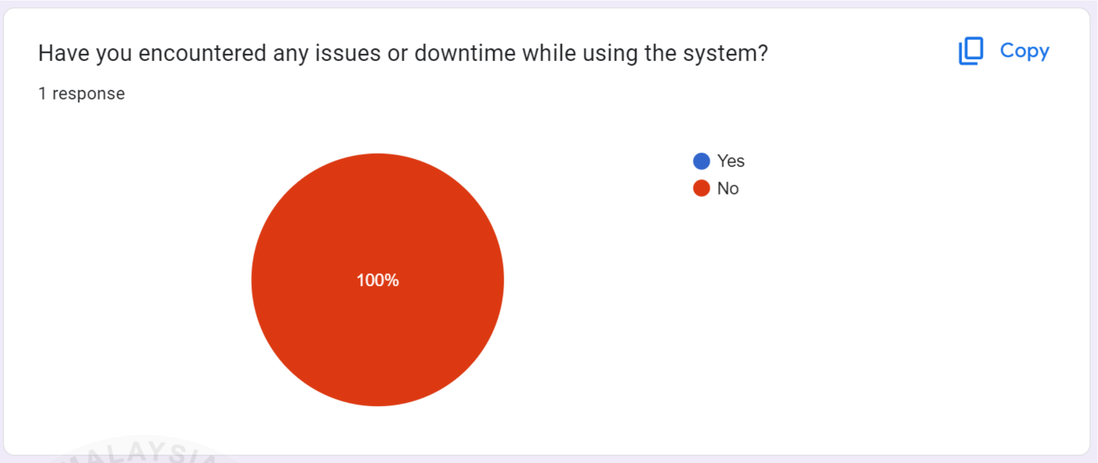
Figure 6.21: Feedback Report of System



How easy is it to add a new position and department at UTeM using the system? (1- Very Hard, 5- Very Easy)

1 response

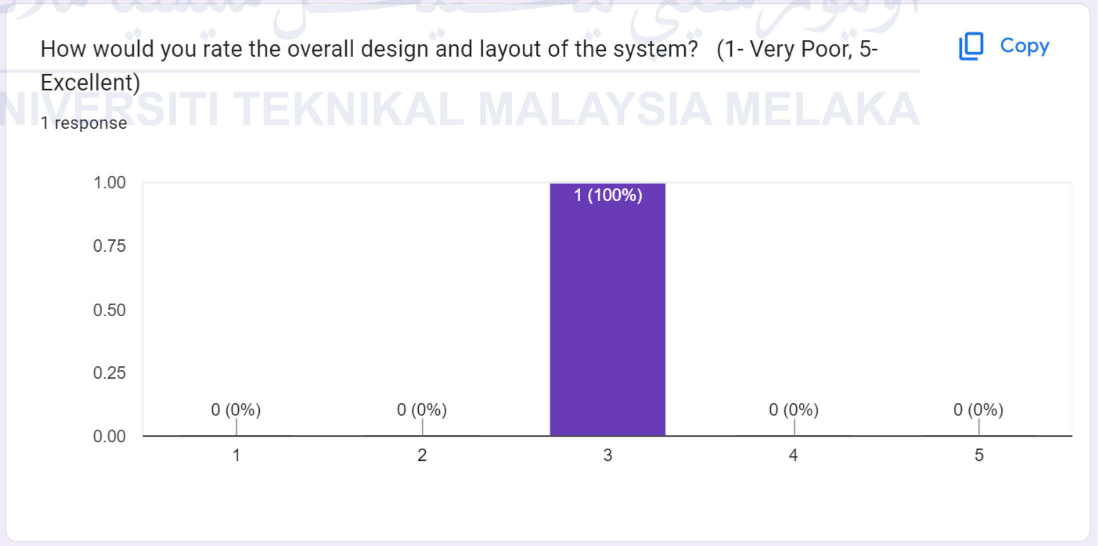
Figure 6.22: Feedback Position and Department Registration of System



Have you encountered any issues or downtime while using the system?

1 response

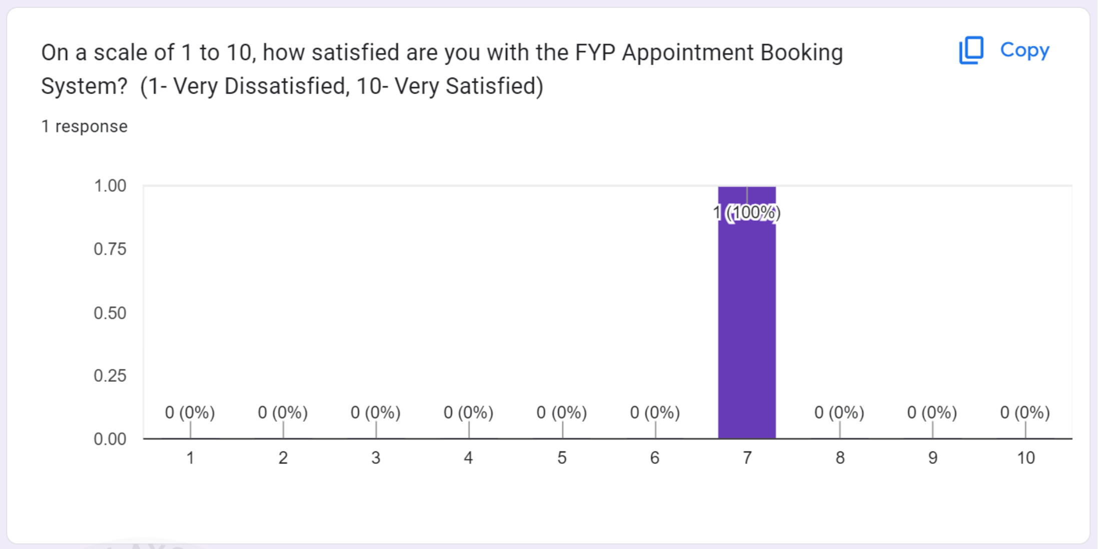
Figure 6.23: Feedback Downtime System for Admin



How would you rate the overall design and layout of the system? (1- Very Poor, 5-Excellent)

1 response

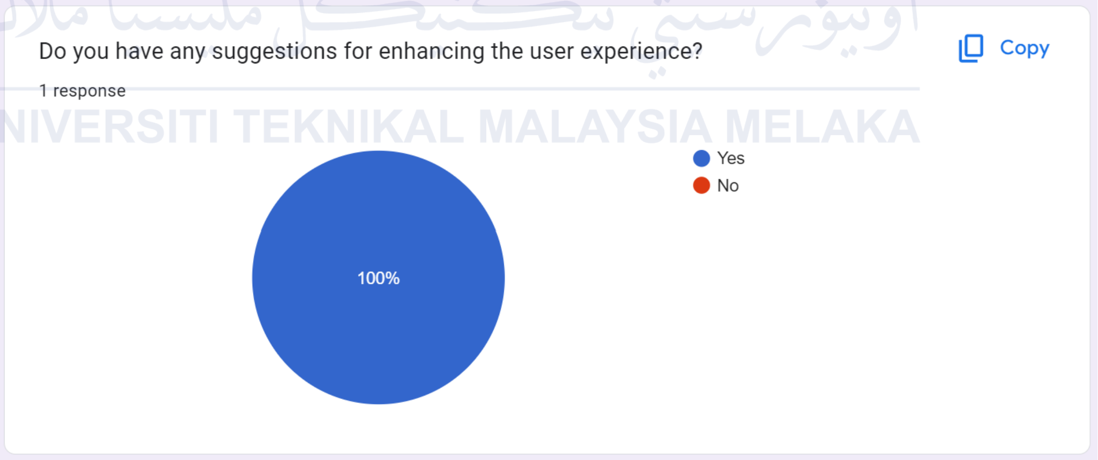
Figure 6.24: Feedback Design of System



On a scale of 1 to 10, how satisfied are you with the FYP Appointment Booking System? (1- Very Dissatisfied, 10- Very Satisfied)

1 response

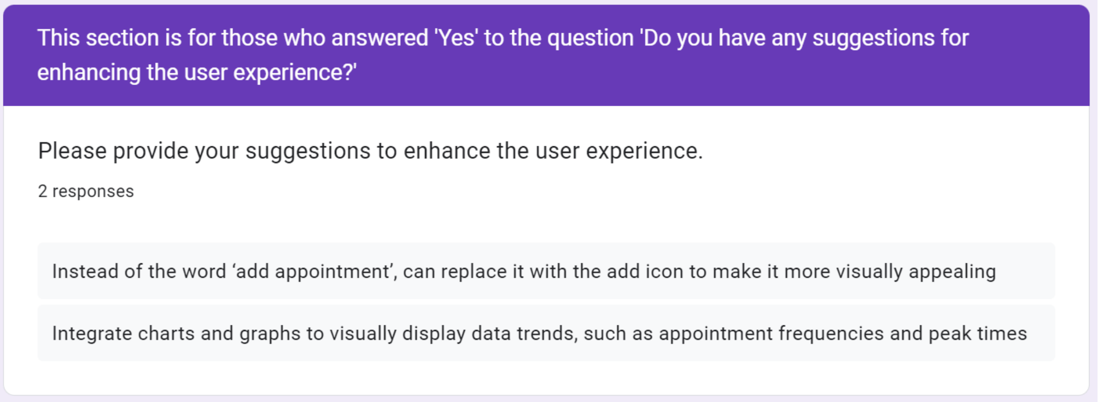
Figure 6.25: Feedback Satisfaction of System



Do you have any suggestions for enhancing the user experience?

1 response

Figure 6.26: Feedback Suggestion of System



This section is for those who answered 'Yes' to the question 'Do you have any suggestions for enhancing the user experience?'

Please provide your suggestions to enhance the user experience.

2 responses

Instead of the word ‘add appointment’, can replace it with the add icon to make it more visually appealing

Integrate charts and graphs to visually display data trends, such as appointment frequencies and peak times

Figure 6.27: Suggestions to enhance User Experience from Admin

6.6 Conclusion

This chapter outlines the techniques used to verify and validate the FYP Appointment Booking System and its functionalities. The test plan includes tests designed to assess changes made to the system. To avoid creating a system that does not meet user requirements and expectations, the test plan ensures thorough inspection. The test organization details the team involved in this project, including the developers. The test environment encompasses the hardware and software platforms used to run the system during testing. The test schedule provides a timetable for organizing each individual testing period. The test strategy describes the approach taken to evaluate the system, focusing on functionality and security. The test description outlines the expected outcomes based on the testing strategy, while the test data refers to the example of input data used for testing purposes. Finally, the test results and analysis involve collecting data from the tests conducted and analyzing it against the expected outcomes and responses.

The next chapter will cover the project’s conclusion, and the lessons learned from this project.

CHAPTER 7: CONCLUSION

7.1 Introduction

This chapter will thoroughly evaluate the overall performance of the FYP Appointment Booking System, offering a comprehensive analysis of its strengths, weaknesses, and areas where improvements are necessary. The evaluation will focus on various aspects of the system, such as its functionality, user interface, reliability, and security, to provide a well-rounded assessment of its current state. Based on these findings, specific recommendations will be proposed to enhance the system's effectiveness and user experience in the future. Additionally, this chapter will highlight the significant contributions of the project, emphasizing its impact on both users and stakeholders, and underscoring its importance in streamlining the appointment booking process mainly for FYP students.

7.2 Observation on Weaknesses and Strengths

Every system has its own strengths and weaknesses. The strengths and weaknesses of the FYP Appointment Booking System are highlighted in this section.

7.2.1 Strength

i. Streamlined Appointment Scheduling

The system streamlines the process of scheduling appointments with supervisors by enabling students to effortlessly select available time slots and confirm appointments, eliminating the need for back-and-forth communication on platforms like WhatsApp or email.

ii. User-Friendly Interface

The system is designed with an intuitive and easy-to-navigate interface, which enhances user experience by simplifying the interaction process. This user-friendly design ensures that individuals with varying levels of technical expertise, from beginners to more advanced users, can effortlessly access and utilize the system's features. The clear layout and logical organization of the interface elements reduce the learning curve, allowing users to quickly become familiar with the functionalities.

iii. Real-Time Updates

The system provides real-time updates for appointment bookings, ensuring that both students and supervisors are always informed about the updated status of appointments, reducing the risk of scheduling conflicts.

7.2.2 Weaknesses

i. Dependency on Internet Connectivity

As a web-based system, its functionality is entirely dependent on a stable internet connection. Any connectivity issues could hinder access to the system, especially during critical times like peak booking periods.

ii. Scalability Concerns

If the system wasn’t designed with scalability in mind, it might struggle to handle a large volume of users or appointments, particularly during peak times, leading to slow performance or system crashes.

7.3 Propositions for Improvement

To address scalability concerns for the FYP Appointment Booking System, it is essential to enhance both its architectural design and resource management. Currently, the system is designed to handle a limited number of users and appointments effectively. However, as user demand grows, the system's performance could be impacted. Implementing a microservices architecture could offer greater flexibility and scalability, allowing individual components of the system to be scaled independently based on load. Additionally, transitioning to a cloud-based infrastructure with auto-scaling capabilities can dynamically allocate resources in response to varying demands. Optimizing database queries and indexing strategies will also improve performance under increased loads. Furthermore, incorporating caching mechanisms and load balancing can distribute traffic efficiently and reduce server strain. By addressing these areas, the system will be better equipped to handle higher volumes of users and data, ensuring smooth operation and user experience as the number of users and appointments increases.

7.4 Project Contribution

The FYP Appointment Booking System marks a significant advancement in academic administration by simplifying the management of student-supervisor interactions. Its user-friendly interface is designed to ease appointment scheduling, making it accessible to users with varying technical skills.

Key features include real-time updates that enhance communication between students and supervisors, reducing conflicts and misunderstandings. This system addresses common scheduling challenges, offering a valuable tool for academic institutions. It enhances appointment management and supports a more organized and communicative academic environment, making it a noteworthy contribution to administrative processes.

7.5 Conclusion

In conclusion, the objectives and scope outlined in Chapter One for the FYP Appointment Booking System have been successfully achieved. The system has effectively addressed all the specified objectives, including streamlining the appointment scheduling process, enhancing user interaction, and ensuring real-time updates. The development methodology utilized was the Database Life Cycle (DBLC), encompassing database initial study, design, implementation and loading, user training, and ongoing maintenance and evaluation. The testing phase was rigorously conducted to identify and rectify any bugs and defects in the system. Ultimately, the FYP Appointment Booking System has met both its functional and non-functional requirements. Despite its successes, there remain areas for improvement to address potential weaknesses and further enhance the system’s performance and user experience. Overall, this project has been completed successfully and meets the requirements set forth for the Bachelor of Computer Science (Database Management) program.

REFERENCES

(1) PHP: MySQL Database. (n.d.). https://www.w3schools.com/php/php_mysql_intro.asp

(2) Portal i@UTeM. (n.d.). https://portal.utem.edu.my/iutem/login.asp

(3) HUB. (n.d.). https://ulearn.utem.edu.my/hub/

(4) *Download XAMPP*. (n.d.). https://www.apachefriends.org/download.html

(5) Gusev, I. (2023, January 25). How to backup SQL Server databases on Windows - SqlBak Blog. SqlBak Blog. https://sqlbak.com/blog/how-to-backup-sql-server-databases/

(6) Online Assignment submission System | Creately. (n.d.). https://creately.com/diagram/example/ispzfh3e/online-assignment-submission-system

(7) Stringfellow, A. (2023, May 1). What is Web Application Architecture? How It Works, Trends, Best Practices and More. Stackify. https://stackify.com/web-application-architecture/

(8) HTML tutorial. (n.d.). https://www.w3schools.com/html/default.asp

(9) Hamilton, T. (2024, April 12). *What is Integration Testing? (Example)*. Guru99. https://www.guru99.com/integration-testing.html