ONE DORMITORY MANAGEMENT SYSTEM

DINIE AZZAHRAA BINTI KHAIRULZAMAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ONE DORMITORY MANAGEMENT SYSTEM

DINIE AZZAHRAA BINTI KHAIRULZAMAN

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

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

ONE DORMITORY MANAGEMENT SYSTEM

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

STUDENT: [illegible] Date: 6 September 2024

(DINIE AZZAHRAA BINTI KHAIRULZAMAN)

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

SUPERVISOR: [illegible] Date: 6 September 2024

(DR NUR ATIKAH BINTI ARBAIN)

DEDICATION

To my beloved parents, thank you for supporting me through this journey. There are no words to describe how much both of you have supported me without any complaint, always praying for me and encouraging me to continue this journey to the end. Your unwavering love and support have been my foundation and strength, guiding me through the toughest times and celebrating my successes with boundless joy. To my friends who helped me walk through this journey, thank you. Your companionship, patience and relentless assistance in finding and correcting the errors in my learning have been invaluable. You have been my study partners, my confidants and my motivators, ensuring that I never felt alone on this challenging path. And of course, to my supervisor, Dr Nur Atikah binti Arbain, thank you for guiding me throughout this journey. Your expertise, advice and unwavering support have been instrumental in shaping my academic and professional growth. You provided me with the direction and insight needed to navigate the complexities of my research and your encouragement pushed me to strive for excellence. To everyone who has been a part of this journey, thank you. Your collective support, wisdom and encouragement have made this achievement possible and I am deeply grateful for each one of you.

ACKNOWLEDGEMENTS

Special appreciation goes to my supervisor, Dr Nur Atikah binti Arbain, for her supervision and constant support. She has been an inspirational role model for this topic, offering invaluable guidance and encouragement throughout the entire process. Her insightful comments and suggestions during the tentative and proposal phases have significantly contributed to the successful completion of this project. Her expertise and dedication have not only enhanced the quality of my work but have also inspired me to strive for excellence.

Additionally, I would like to extend my heartfelt thanks to my beloved parents, who have provided unwavering support and motivation throughout my project. Their constant encouragement and belief in my abilities have been a continuous source of strength, helping me to persevere through challenges and stay focused on my goals. Their prayers, love and sacrifices have been fundamental to my achievements, and I am deeply grateful for their presence in my life. This project would not have been possible without the collective support and contributions of both my supervisor and my parents and I am profoundly thankful for their roles in this journey.

ABSTRACT

The One Dormitory Management System is designed to streamline the administrative processes of managing student dormitories, offering a comprehensive solution to the challenges faced by hostel administrators. Featuring a user-friendly interface, this system efficiently handles room allocation, fee payment tracking, maintenance requests, and occupancy records. Built using PHP as the programming language and MySQL for the database, it integrates key functionalities like automated room assignment, real-time updates, and secure data storage. Automated room assignments ensure swift and fair allocation, while real-time updates keep administrators and students informed about current statuses and changes, reducing confusion and delays. The robust database securely stores all student-related data, facilitating easy access and management. Implementing this system significantly reduces administrative workload, allowing staff to focus on critical tasks and improving data management through centralized records. It simplifies processes and minimizes errors, providing a seamless experience for both students and administrators. By modernizing dormitory management practices, the system ensures a well-organized and efficient living environment for students. Educational institutions adopting this system can expect improved management of student housing, higher student satisfaction, and a more streamlined operational workflow for administrators. This system has undergone a comprehensive testing phase, with all functionalities thoroughly evaluated prior to deployment to end users. User acceptance testing was carried out through a survey to gather feedback from real users. The survey is divided into three parts: user information, system functionality and overall feedback. The results were highly satisfactory, and I received a significant amount of positive feedback from the participants, the majority of whom are students and colleagues from the university.

ABSTRAK

Sistem Pengurusan Satu Asrama direka untuk menyelaraskan proses pentadbiran pengurusan asrama pelajar, menawarkan penyelesaian yang komprehensif kepada cabaran yang dihadapi oleh pentadbir asrama. Menampilkan antara muka yang mesra pengguna, sistem ini mengendalikan peruntukan bilik, penjejakan pembayaran yuran, permintaan penyelenggaraan dan rekod penghunian dengan cekap. Dibina menggunakan PHP sebagai bahasa pengaturcaraan dan MySQL untuk pangkalan data, ia menyepadukan fungsi utama seperti tugasan bilik automatik, kemas kini masa nyata dan penyimpanan data selamat. Tugasan bilik automatik memastikan peruntukan yang pantas dan adil, manakala kemas kini masa nyata memastikan pentadbir dan pelajar dimaklumkan tentang status dan perubahan semasa, mengurangkan kekeliruan dan kelewatan. Pangkalan data yang teguh menyimpan semua data berkaitan pelajar dengan selamat, memudahkan akses dan pengurusan yang mudah. Melaksanakan sistem ini dengan ketara mengurangkan beban kerja pentadbiran, membolehkan kakitangan memberi tumpuan kepada tugas kritikal dan menambah baik pengurusan data melalui rekod terpusat. Ia memudahkan proses dan meminimumkan ralat, memberikan pengalaman yang lancar untuk pelajar dan pentadbir. Dengan memodenkan amalan pengurusan asrama, sistem ini memastikan persekitaran hidup pelajar yang teratur dan cekap. Institusi pendidikan yang menggunakan sistem ini boleh mengharapkan pengurusan perumahan pelajar yang lebih baik, kepuasan pelajar yang lebih tinggi dan aliran kerja operasi yang lebih diselaraskan untuk pentadbir. Sistem ini telah melalui fasa ujian menyeluruh, dengan semua fungsi dinilai secara menyeluruh sebelum digunakan kepada pengguna akhir. Ujian penerimaan pengguna telah dijalankan melalui tinjauan untuk mengumpul maklum balas daripada pengguna sebenar. Tinjauan ini dibahagikan kepada tiga bahagian: maklumat pengguna, fungsi sistem dan maklum balas keseluruhan. Keputusannya amat memuaskan, dan saya menerima sejumlah besar maklum balas positif daripada para peserta, yang majoritinya pelajar dan rakan sekerja dari universiti.

TABLE OF CONTENTS

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATION

FYP	–	**Final Year Project**
DBLC	–	**Database Life Cycle**
ERD	–	**Entity Relationship Diagram**
DDL	–	**Data Definition Language**
DML	–	**Data Manipulation Language**
UTA	–	**User Testing Acceptance**

CHAPTER 1: INTRODUCTION

1.1 Introduction

In today's rapidly evolving world, the demand for reliable and efficient hostel management services has never been more critical. For many students, a hostel is not just a place to stay, it serves as a second home, offering a sense of comfort and stability while they are away from their families. A well-managed hostel typically features large, well-ventilated dormitories and rooms, situated within the college premises, providing students with a safe and conducive environment for their academic pursuits. Ensuring that these accommodations are clean, calm and well-maintained is a key responsibility of college management, as it directly impacts the well-being and academic success of the students.

To address these needs, advanced platforms have been developed to connect hostel administrators with students and staff, streamlining various aspects of hostel operations. The introduction of a web-based hostel management system has revolutionized the way accommodations are provided to university students. This system is designed to manage all facets of hostel life more efficiently, from the initial room allocation to ongoing maintenance and student support. By centralizing information and automating routine tasks, the system reduces administrative burden, allowing staff to focus on creating a better living environment for students.

The primary purpose of a hostel management system is to ensure that all aspects of hostel operations are handled in an efficient and effective manner. By leveraging this technology, hostel administrators can offer a smoother and more enjoyable experience for both themselves and the students. The system facilitates the allocation of rooms based on student preferences and availability, ensuring a fair and optimized distribution of accommodations. It also enhances communication between students, staff, and administrators, making it easier to address any issues that may arise. Despite these advancements, significant challenges persist in the realm of hostel management. One of the most pressing issues is optimizing operational efficiency to ensure that resources are used effectively.

1.2 Problem Statement

1. The data stored in manual files involves physical storage or documents but not in a database, leading to data duplication, repetitive data, and data loss.
2. Manual recording is prone to errors, and it takes a significant amount of time to search through the student hostel records one by one.
3. Generating reports manually is time-consuming, error-prone, and lacks real-time insights into dormitory management.

1.3 Objectives

This project embarks on the following objectives:

1. To store student dormitory records in a centralized database, minimizing data duplication and reducing the risk of data loss.
2. To improve the efficiency of record-keeping processes in student hostels by transitioning to a computerized recording system.
3. To develop a system that automates the generation of reports, reducing the labor and errors associated with manual report creation.

1.4 Scope

1. **Target Users**

a) Admin

b) Staff

c) Student

2. **Modules to be developed**

a) Login Module

- In this module, users will securely log in using their unique user ID and password, ensuring that only authorized individuals can access or modify the system. The module incorporates robust security measures, including password encryption and user authentication protocols.
- It also allows new users, such as students and staff, to register by providing necessary personal and contact information, followed by the creation of unique login credentials. Additionally, it includes password recovery options to assist users in regaining access if they forget their credentials.

b) Student Module

- Students can access detailed information about their assigned room. This module provides a comprehensive view of the student's living arrangements within the hostel.
- It also allows students to view billing information, including a breakdown of hostel fees, payment schedules, and any outstanding balances. Students can easily monitor their financial obligations.
- Additionally, students can review their payment history and download receipts for each transaction, ensuring transparency and easy record-keeping for future reference.

c) Staff Module

- The staff module grants authorized personnel access to essential student information, such as room assignments, contact details, and emergency contacts. This data is crucial for managing student welfare and responding to any issues.
- Staff can also modify existing room details, including updating room assignments, managing occupancy and ensuring room readiness before student check-ins. This feature helps maintain accurate and up-to-date room records.
- The module allows staff to track hostel fees, monitor payment deadlines, and manage outstanding balances efficiently.

d) Room Module

- This module is responsible for the comprehensive management of all hostel rooms, covering everything from room allocation to maintenance. It ensures that each room is properly recorded and tracked within the system.
- It stores and organizes detailed information about each room in the hostel, including room number and type. This helps in making informed decisions regarding room assignments.
- The module also includes functionalities for scheduling room inspections, handling maintenance requests, and tracking room availability, ensuring that the hostel operates smoothly.

e) Payment Module

- The payment module allows administrators to configure fee structures based on various criteria such as room type, amenities provided, duration of stay, and any additional services or charges. It can also accommodate discounts for early payments or scholarships.
- This module is designed to notify students about late payment penalties through automated reminders and notifications, ensuring timely payments. It also includes functionalities for taking appropriate actions to recover overdue amounts, such as restricting access to certain services until payments are made.
- Additionally, it can generate detailed financial reports for administrators, helping in budget planning and financial auditing.

f) Complaint Module

- The complaint module empowers students to submit complaints or issues related to their hostel experience through an easy-to-use online form within the system. These can range from maintenance requests to grievances about room conditions or other hostel-related matters.
- This module facilitates communication between staff and students regarding specific complaints, allowing staff to acknowledge receipt, provide updates, and offer resolutions. It ensures that student concerns are addressed promptly and effectively.
- Staff members can also change the read status of a complaint, marking it as pending or approved ensuring transparency and accountability in the complaint resolution process.

g) Hostel Module

- The hostel module manages comprehensive information about the hostel, including overall capacity, room availability, and maintenance schedules. It helps administrators keep track of hostel resources and plan for future needs.
- Students can view detailed hostel information, such as the total number of rooms, available facilities, common areas, and maintenance updates. This transparency allows students to make informed decisions about their accommodation.
- Additionally, this module can generate reports on hostel occupancy rates, maintenance activities, and future capacity planning, assisting administrators in efficiently managing hostel operations.

1.5 Project Significant

1. **Effective Communication Channels:** Enable efficient communication between hostel staff, students and management through features such as complaint submission and real-time notifications.

2. **User-Friendly Interface:** Design an intuitive and user-friendly interface accessible through web or mobile platforms, catering to the diverse needs of administrators, staff and students and enhancing overall user experience and satisfaction.

3. **Optimized Resource Management**: Implement efficient resource management tools that allow administrators to effectively allocate rooms, track occupancy, and manage hostel resources, leading to better utilization of facilities and improved operational efficiency.

1.6 Expected Outputs

Seamless and Efficient Operation:

- Streamlined hostel-related processes leading to smoother daily operations.

Improved User Experience:

- Enhanced satisfaction for both students and staff through an intuitive and user-friendly system interface.

Student Capabilities:

- Students can easily register, book rooms, make payments, and submit complaints through the system.

Staff Efficiency:

- Staff will have access to streamlined tools for managing room allocations, tracking payments, and resolving student issues.

Enhanced Communication:

- Improved communication between students and staff, facilitating quick responses and better service.

Reduction of Manual Errors:

- Automated processes reduce the likelihood of errors typically associated with manual data handling.

Smooth Operation of All Processes:

- Ensuring that all aspects of hostel management run efficiently, leading to a more organized and well-managed environment.

1.7 Summary

The One Dormitory system has emerged with new goodies compared to the experience where every activity concerning the hostel business is limited to a physical location only. Even though the physical location has not been eradicated, the nature of functions and how these functions are achieved have been reshaped by the power of the internet. The web-based hostel management system has offered an advantage to both students as well as staff to efficiently and effectively manage the business and satisfy students' needs at the click of a button.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

The process of developing a database system involves obtaining precise requirements, assessing them as well as the system’s capabilities and data design, and then putting each function into practice inside its modules. A database system’s development through the Database Life Cycle technique’s stages helps ensure it has undergone sufficient preparation before usage. More information about this phase is covered in this chapter.

2.2 Project Methodology

The choice of methodology is crucial for the success of any system development. The One Dormitory Management System project opted for the waterfall model due to its organized structure, which allows for early issue detection and rectification. Its simplicity and clarity facilitate productive teamwork and stakeholder communication. In this model, stages are developed sequentially, including requirements, concept development, design, creation, testing, launch, and maintenance. Each phase begins only after the previous one is completed, ensuring a methodical development lifecycle. This approach aligns with the Database Life Cycle (DBLC), reinforcing our commitment to completing the project on schedule and to a high standard.

The DBLC contains six phases: initial study, design, implementation, testing, operation, and maintenance, each ensuring the database system's efficiency, reliability, and adaptability. The initial study examines current operations to identify inadequacies. The design phase creates a supporting database model. Implementation transforms the design into a functional database, followed by rigorous testing. Once operational, the database integrates with management and user applications, initiating system evolution. The maintenance phase includes routine activities like backups and access management, ensuring the system remains efficient, secure, and up-to-date.

a) Database Initial Study

In the initial study phase, the current operations and procedures within the dormitory management system are closely examined through research, observation, and user feedback. This examination seeks to identify the shortcomings of the existing system, such as inefficiencies in room allocation, difficulties in managing student preferences, or challenges in maintaining real-time updates. The analysis focuses on understanding the problems faced by administrators and students, as well as identifying any operational constraints that hinder the overall performance. Through this comprehensive analysis, the study defines the scope, purpose, and boundaries of the new One Dormitory Management System, aiming to address these issues with a modernized solution. By outlining the key objectives and limitations, this phase provides a solid foundation for moving forward with the design and development of the new system, ensuring that it aligns with the needs of its users.

b) Database Design

Database design is a critical phase in the development of the One Dormitory Management System, as it defines how data will be structured, stored, and accessed. This phase begins with the conceptual design, which involves the creation of an Entity-Relationship Diagram (ERD) to visually represent the system's main entities, such as students, rooms, and administrators, along with the relationships between them. This step helps clarify the system's data requirements and the interactions between different entities. Moving on to logical design, this stage delves into the specific schema definitions, determining the data types, relationships, and constraints that will ensure data integrity and consistency. Additionally, a detailed data dictionary is created to define every field in the database. The physical design phase focuses on the actual implementation of the database structure, optimizing storage, and enhancing performance by organizing tables, indexes, and other database objects. Furthermore, database objects like triggers, stored procedures, and views are developed to automate processes, enforce business rules, and ensure efficient data manipulation. This thorough approach ensures that the database not only meets the system's current needs but is also scalable and adaptable to future requirements.

c) Implementation

The implementation phase is where the theoretical database design transforms into a functional system. During this phase, the conceptual, logical, and physical designs are translated into a working database. This begins with the creation of database tables and attributes according to the defined schemas. Each table represents an entity from the ERD, and relationships between these entities are implemented using foreign keys and constraints. Alongside table creation, the development of triggers and stored procedures is critical to ensuring that automated actions are performed when specific conditions are met, such as sending notifications when a room is vacated or enforcing business rules related to room allocation. The implementation of these database objects helps streamline operations and maintain data integrity throughout the system. This phase is crucial for ensuring that the system is tailored to the unique needs of the One Dormitory Management System and is ready for deployment.

d) Testing

Testing is one of the most critical phases of the project lifecycle, as it ensures that the One Dormitory Management System functions as expected and meets all predefined requirements.

This stage involves conducting a series of rigorous tests, beginning with unit testing, which checks the functionality of individual components such as tables, queries, triggers, and procedures. Integration testing follows, where the interactions between different components are tested to ensure that they work together seamlessly. User Acceptance Testing (UAT) is also conducted, allowing actual users to interact with the system and provide feedback on its usability, performance, and overall experience. This stage ensures that the system is user-friendly and meets the needs of the dormitory administrators and students. Performance testing is also vital in this phase, as it checks the database's ability to handle large volumes of data and simultaneous user interactions without lag or failure. By conducting thorough testing, any bugs or issues are identified and resolved before the system goes live, ensuring a reliable and efficient database for dormitory management.

e) Operation

Once the database has successfully passed all testing and evaluation phases, it is deployed into the production environment, marking the beginning of the operational phase. During this phase, the One Dormitory Management System becomes fully functional, allowing users to perform real-time operations such as room allocation, student record management, and reporting. The operational phase is not just about maintaining the database; it also involves continuous monitoring to ensure that the system runs smoothly and efficiently within its intended environment. Regular performance checks, security audits, and user feedback are essential during this phase to identify potential improvements and to ensure that the system adapts to the growing needs of the dormitory. The start of the operational phase also signals the beginning of system evolution, as future updates, enhancements, and feature additions may be required to meet changing requirements.

f) Maintenance

Maintenance is the final, ongoing phase of the database lifecycle, crucial to ensuring that the One Dormitory Management System remains efficient, secure, and responsive over time. During this phase, the database administrator performs routine tasks such as data backups to prevent data loss in case of system failure, database optimization to maintain performance, and updating user permissions to accommodate new students, staff, or system modifications. Security patches and software updates are also applied to protect the system against vulnerabilities and to ensure that it complies with the latest standards. Additionally, as the system evolves and the number of users grows, maintenance may involve expanding storage capacity, refining queries, or adding new functionalities to meet the ever-changing demands of the dormitory environment. Through continuous monitoring, updates, and improvements, this phase ensures that the system stays reliable and meets the long-term needs of its users.

2.3 Project Schedule and Milestone

This part explains how the project has been managed from start to end.

MONTH / TASK	YEAR: 2024						
	3	4	5	6	7	8	9
Database Initial Study	🟪						
Database Design	🟪	🟪	🟪				
Implementation				🟪			
Testing					🟪	🟪	
Operation						🟪	
Maintenance						🟪	🟪

2.4 Summary

To sum up, a project's success depends on the methodical approach to database development, which guarantees that the data is arranged, dependable, and accessible. The planning process and waterfall model that was selected as the approach have a significant impact on how the database development project turns out. Through our exploration of database development methodology, it is evident that each approach brings its own set of advantages and challenges.

By adhering to the waterfall model, the project benefits from a clear roadmap that helps in managing time, resources, and expectations. The methodical progression through each phase minimizes the risk of errors and omissions, leading to a more robust and well-structured database system

CHAPTER 3: ANALYSIS

3.1 Introduction

Database system analysis involves evaluating a company's data management infrastructure with a focus on database development, application, and optimization. This includes determining data needs, identifying sources, constructing data structures, and setting up systems for efficient processing and storage. The goal is to investigate problems in current systems to inform improvements. This ensures the database meets corporate objectives, adheres to security and performance best practices, and provides timely information, enhancing data management and supporting informed decision-making.

Database system analysis also involves assessing how well the existing infrastructure aligns with the organization's strategic goals and operational needs. By examining data flow, user interactions, and system performance, analysts can identify bottlenecks or inefficiencies that may hinder productivity. This phase also includes a review of data integrity measures, such as backup processes and disaster recovery plans, ensuring that critical information is protected. Additionally, system scalability is evaluated to determine if the current database can support future growth and increased data demands. Ultimately, this comprehensive evaluation helps tailor the database system to better support decision-making, streamline processes, and improve overall business performance.

3.2 Problem Analysis

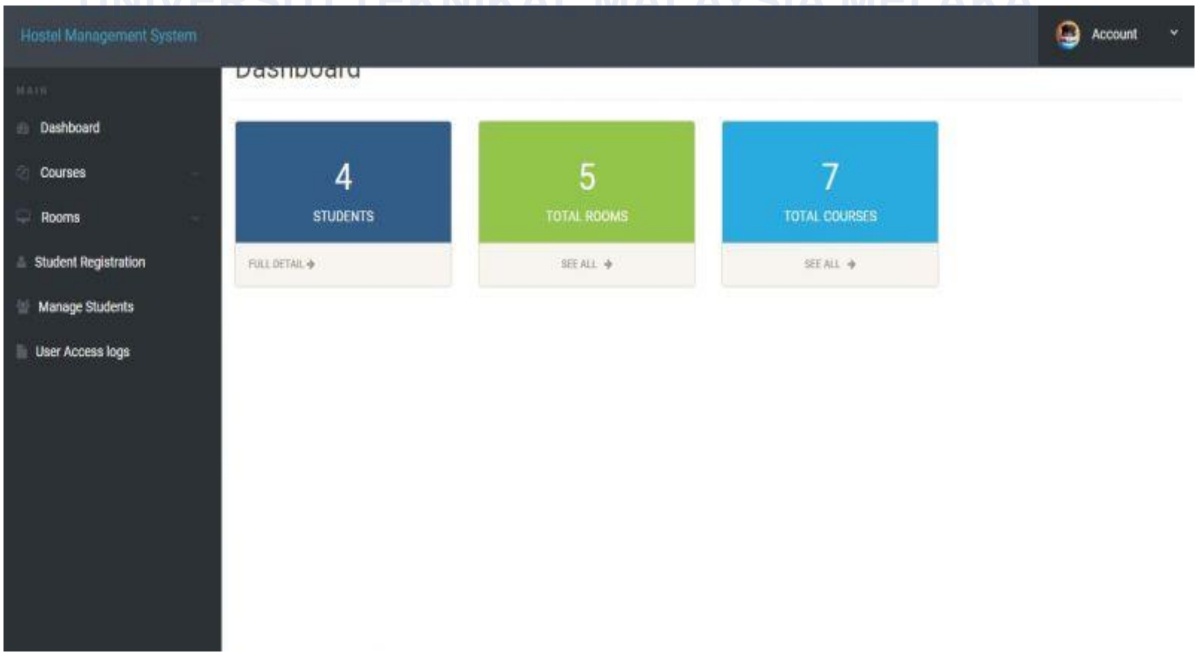


Figure 3.1 : This is an interface of the existing system for the hostel management system

The figure 3.1 is the existing system for staff

- The dashboard typically features a simple, grid-based layout or a series of basic sections, focusing on displaying key information rather than a polished visual design
- Uses a limited and often dull color palette, such as shades of grey, white or light blue. The colors are usually chosen for functionality rather than aesthetics.
- Information is displayed in static tables or lists
- Limited navigation options with basic menus or links to other system sections. Navigation is functional but not designed for a smooth user experience.
- The focus is on functionality, so the dashboard may lack modern design elements like visual hierarchy, attractive icons or engaging animations.

Table 3.1 : Summary feature of the current system and the new system

Existing System	New System
Uses basic layout and styling with minimal visual enhancements	Incorporates improved layout, styling, and visual elements for better aesthetics and user experience
Displays hostel options with images and names only	Enhances hostel display by adding "View More" buttons for additional information
Does not provide pop-out details for hostels	Implements pop-out details with hostel capacity and status upon clicking the "View More" button
Limited use of colors, fonts and graphical elements	Use of a modern design with appealing color schemes, varied fonts, and engaging graphical elements

3.3 The proposed improvements/solutions

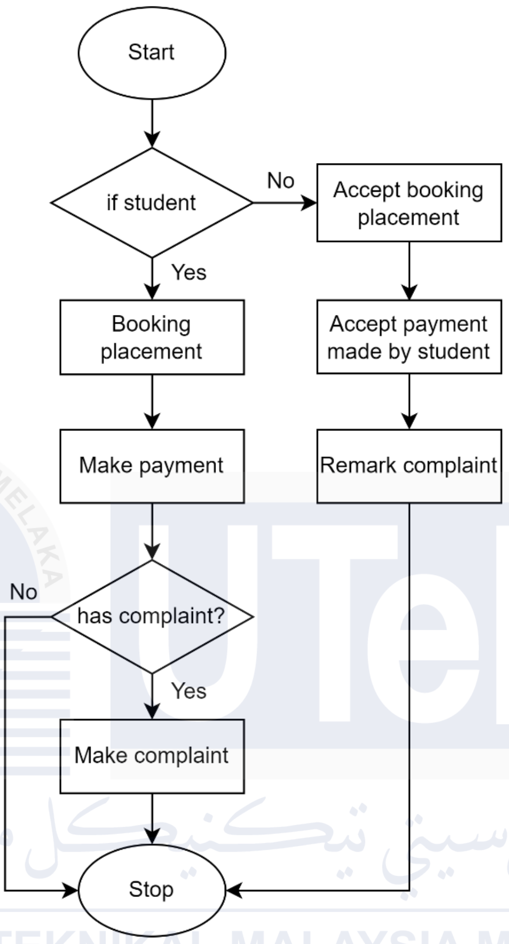


Figure 3.2 : Flowchart of the proposed system

As shown in Figure 3.2, the process begins when a student identifies the need for a hostel room. This triggers the student to make a booking request using the system's automated booking function. Once the booking request is made, it becomes visible to all available rooms within the hostel. At this point, a decision is made by the system based on room availability and booking preferences. If a suitable room is found, the process moves forward.

Once a room is assigned, the student must first gain access to the room. This may involve obtaining keys or other necessary credentials. After gaining access, the student makes a payment using the system's secure and automated payment processing function. This ensures that the student is occupying the room before finalizing the payment. If the student has any complaints or issues with the room, they can submit a complaint through the system's designated complaint channel. The complaint will be reviewed and addressed by the hostel management.

In addition, the system provides administrators with tools to track room occupancy, manage bookings, and monitor student activities, ensuring compliance with hostel policies and maintaining a safe and comfortable environment. This marks the end of the process. The proposed system aims to automate these steps, improving efficiency and communication between students, administrators, and hostel staff.

3.4 Requirement analysis of the to-be system

3.4.1 Functional Requirement

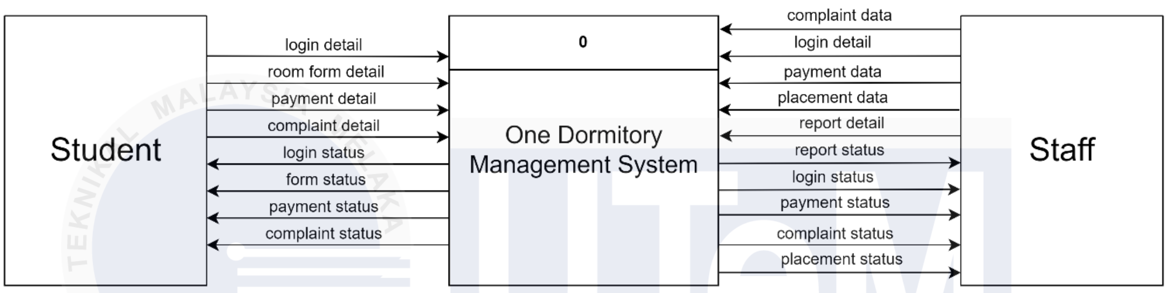


Figure 3.3 : Context Diagram for One Dormitory

Student: The student is provided with personalized access to the system through secure login credentials. Once logged in, they are presented with a user-friendly interface where they can conveniently view essential information such as their room allocation details, the status of their payments, and the current progress of any complaints they have submitted. If the student encounters any issues or concerns, they can use the built-in complaint submission feature. By filling out the complaint form with the necessary details, such as the nature of the issue and any supporting information, the student can ensure that their complaint is formally lodged. Once submitted, the complaint is directed to the appropriate staff for review, and the system provides updates on its status, ensuring the student is informed throughout the resolution process.

Staff: Staff members have dedicated access to the system through their secure login details, which grant them administrative privileges tailored to their role. This centralized access allows staff to efficiently manage student data and ensure that all information is accurate and up to date. In terms of complaints, the system provides staff with a clear view of all submitted issues, including the specific details entered by the students. Staff can track the status of each complaint from the moment it is submitted until it is resolved, enabling them to prioritize urgent matters and ensure timely responses. Their actions may involve communicating directly with the student for clarification, coordinating with other staff members to resolve the issue, or taking hands-on steps to address the problem. The system enhances staff efficiency by organizing and streamlining complaint management, improving overall service quality for the students

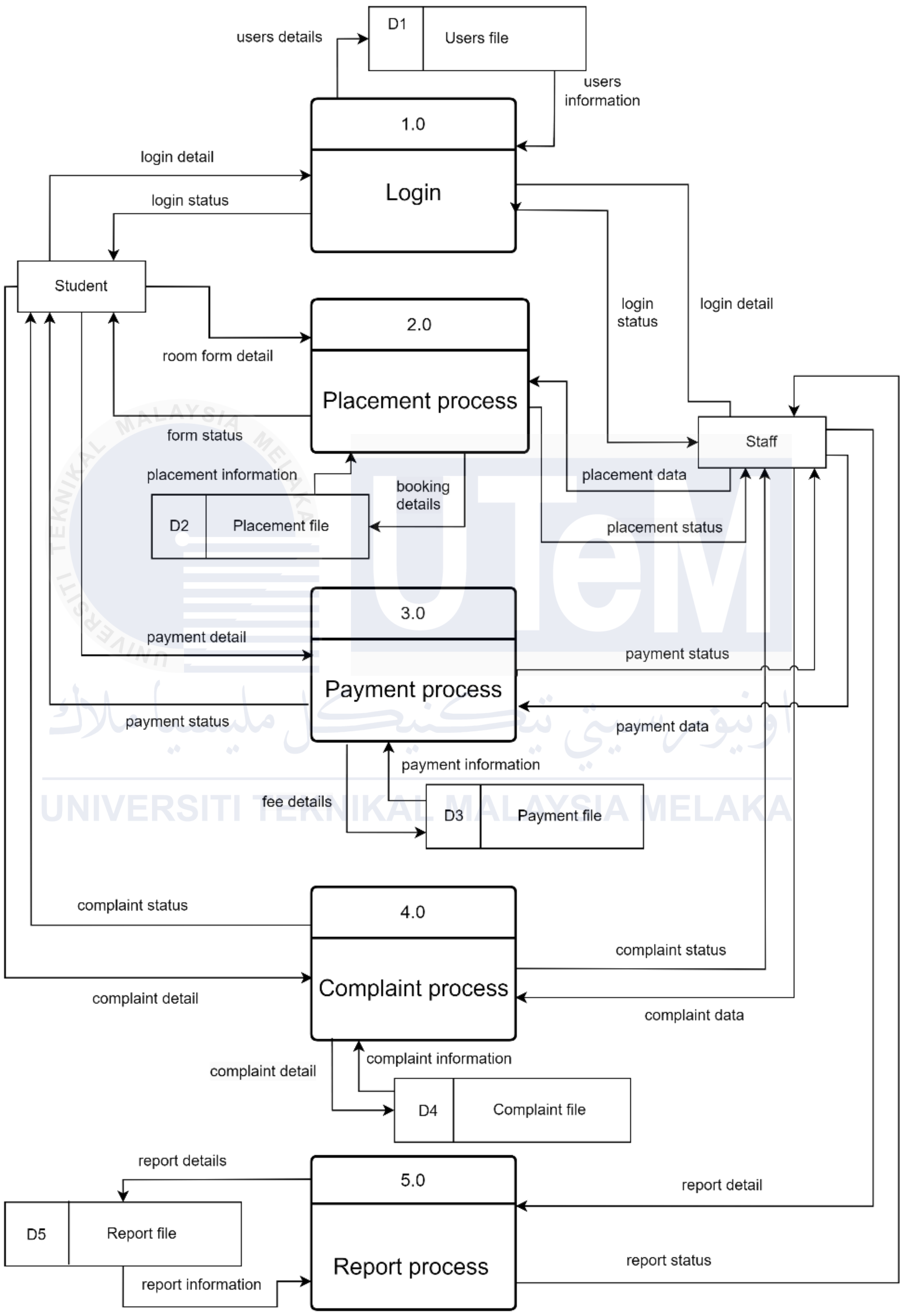


Figure 3.4 : Data Flow Diagram Level 1 for One Dormitory

3.4.2 Non-functional Requirement

Quality Standards

1. **Reliability:** One of the primary quality standards for the One Dormitory Management System is ensuring high reliability, which is crucial for maintaining the trust of users. The system must operate with minimal disruptions, targeting a system availability of 99.9%. This means that downtime, whether for maintenance or unexpected failures, should be minimized to ensure continuous and smooth operations for both students and staff. Implementing redundancy measures, such as backup servers or failover systems, can help maintain availability, reducing the risk of significant disruptions in service. Additionally, proactive monitoring of system health can help identify potential issues before they affect users, further enhancing reliability.
2. **Usability:** Usability is key to ensuring that the system is effective and widely adopted by both students and staff. The interface should be designed with simplicity and intuitiveness in mind, making it easy to navigate without requiring extensive training. A well-thought-out user interface (UI) design that includes clear labels, intuitive navigation paths, and responsive elements can help users quickly understand how to use the system. Incorporating features like tooltips, help sections, and user-friendly error messages can also improve the overall experience, ensuring that both tech-savvy users and those less familiar with digital platforms can efficiently perform their tasks.
3. **Maintainability:** To ensure the system can be easily updated and debugged in the future, maintainability is a crucial quality standard. The system should be designed in a modular fashion, where different components can be updated or fixed independently without affecting the entire system. This modular design simplifies the process of troubleshooting and makes it easier to add new features or make modifications as needed. Additionally, explicit and detailed documentation is essential for maintaining system longevity. The documentation should cover system architecture, code annotations, and guidelines for future developers or administrators to follow when making updates or diagnosing issues.

Security Requirements

1. **Security:** Security is a top priority for the One Dormitory Management System, as it handles sensitive data related to both students and staff. The system should employ role-based access control (RBAC), which restricts access to data and system functionalities based on the user's role within the organization. For instance, students should only be able to view their personal details, while staff members may have broader access to manage student data and resolve complaints. Additionally, multi-factor authentication (MFA) should be implemented to further

secure the login process, requiring users to provide two or more verification factors, such as a password and a one-time code sent to their phone. This reduces the risk of unauthorized access and ensures that only verified users can interact with the system.

2. **Vulnerability Management:** To protect the system against emerging threats, regular security audits and vulnerability assessments are necessary. These processes help identify potential weaknesses in the system's defenses, such as outdated software, misconfigured settings, or exploitable code. By conducting these audits regularly, the system administrators can proactively address vulnerabilities before they are exploited by malicious actors. Furthermore, timely application of security patches and system updates is essential to maintaining the system's resilience against new threats. A robust vulnerability management strategy ensures that the system remains secure, safeguarding both the data and the functionality of the hostel management system.

Operational Requirements

1. **Performance:** The performance of the One Dormitory Management System must be optimized to ensure smooth operations, even during periods of peak load. The system should maintain acceptable response times under heavy user activity, such as during the beginning of a new academic term when many students may be accessing the system simultaneously to check room assignments or make payments. To achieve this, performance monitoring tools should be implemented to track key system metrics such as response time, server load, and database query times. Continuous performance optimization, such as database indexing or server load balancing, can help prevent bottlenecks and ensure the system operates efficiently under all conditions.

2. **Scalability:** As the number of students and the size of the hostel facilities grow, the system must be able to scale accordingly without significant performance degradation. Horizontal scalability, which involves adding more servers or resources to handle increased demand, should be considered in the system design. This ensures that as student enrollments rise or additional hostel buildings are introduced, the system can handle the larger data volumes and increased user activity seamlessly. A scalable design also allows for future expansions, such as the inclusion of new features or integration with other university systems, ensuring the system remains adaptable to the institution's evolving needs.

3.4.3 Others Requirements

Table 3.2 : Software requirement description

Software	Description
Microsoft VisualCode	Projects fully utilize Visual Studio's many time-saving and productivity features, maximizing efficiency and effectiveness in development. The numerous built-in tools streamline workflows, enhancing overall performance and output.
XAMPP	XAMPP is selected for its all-in-one package that includes Apache, MySQL, PHP and Perl, simplifying the setup process. Compared to other web servers like WAMP, it is much simpler to set up, offering an easier and more user-friendly installation and configuration process for developers.

Table 3.3 : Hardware requirement description

Hardware	Specification	Reason of choosing
Laptop (Acer Spin 3)	• Windows 10 Home 64-bit • Intel® Core™ i5-1135G7 processor Quad-core 2.40 GHz • 13.3" WQXGA (2560 x 1600) 16:10 IPS Touchscreen • RAM: 8 GB, LPDDR4X 512 GB SSD	Speedy Connectivity

3.5 Summary

In conclusion, a comprehensive database system analysis empowers organizations to optimize their data management practices by aligning systems with strategic business objectives. This analysis enables the implementation of effective data governance policies, ensuring that data is collected, stored, and accessed in a manner that complies with security and integrity requirements.

Moreover, a well-executed database system analysis fosters a culture of data-driven decision-making. By streamlining data retrieval processes and improving data quality, organizations can extract valuable insights that inform strategic planning, operational efficiency, and risk management. This ultimately leads to more informed and effective decision-making, driving business growth and innovation.

CHAPTER 4: DESIGN

4.1 Introduction

In this chapter, conceptual design is an early phase of the design process. It involves an understanding of people's needs and how to meet them with products, services, and processes. Next, Logical database design is the process of transforming (or mapping) a conceptual schema of the application domain into a schema for the data model underlying a particular relational DBMS. Finally, Physical database design is the process of transforming logical data models into physical data models.

4.2 Database Design

4.2.1 Conceptual Database Design

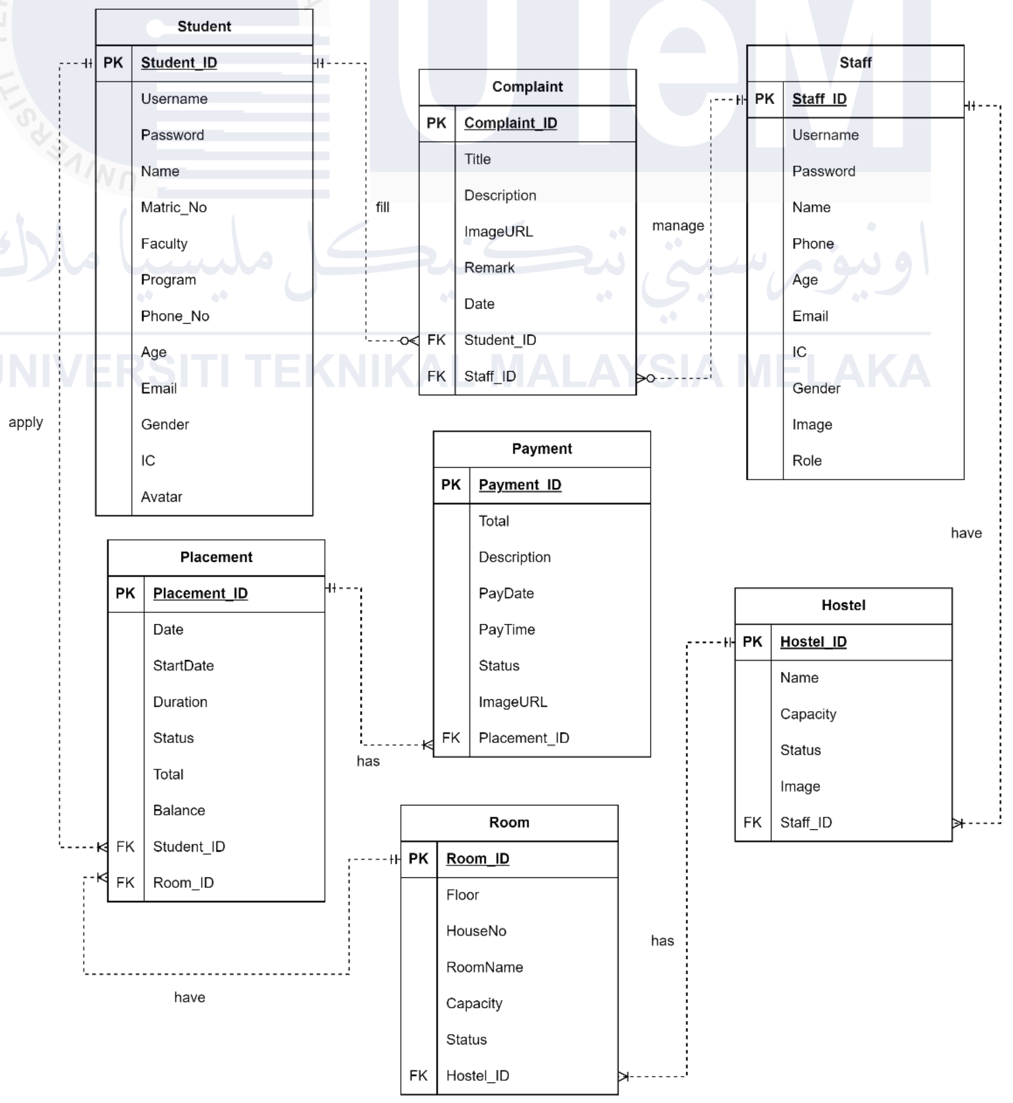


Figure 4.1 : Entity Relationship Diagram

Business Rules

- **Student-Placement Relationship:** A student can apply for multiple placements, allowing them to explore various options and preferences. However, each placement can only be assigned to a single student to ensure fairness and prevent conflicts.
- **Student-Complaint Relationship:** Students may encounter issues or concerns during their stay, prompting them to file complaints. While a student can submit multiple complaints if necessary, each complaint is typically addressed by a specific individual to ensure accountability and efficient resolution.
- **Placement-Payment Relationship:** A placement can involve multiple payments, such as upfront fees, monthly rent, or additional charges for services. However, each payment must be clearly linked to a specific placement to maintain accurate financial records and avoid confusion.
- **Placement-Room Relationship:** Each placement is assigned to a particular room, providing the student with a designated living space. However, a room can accommodate multiple placements, especially in shared or dormitory-style arrangements.
- **Room-Hostel Relationship:** A room is always located within a specific hostel, providing students with access to shared facilities and amenities. A hostel can house multiple rooms, catering to varying student needs and preferences.
- **Staff-Complaint Relationship:** Each complaint is typically assigned to a specific staff member to ensure that it is addressed in a timely and appropriate manner. A staff member can handle multiple complaints, demonstrating their ability to manage workload and resolve issues effectively.
- **Staff-Hostel Relationship:** A staff member can be responsible for managing multiple hostels, overseeing operations, and ensuring the well-being of students. However, each hostel is typically assigned to a primary staff member who is accountable for its day-to-day management.

4.2.2 Logical Database Design

4.2.2.1 Normalization

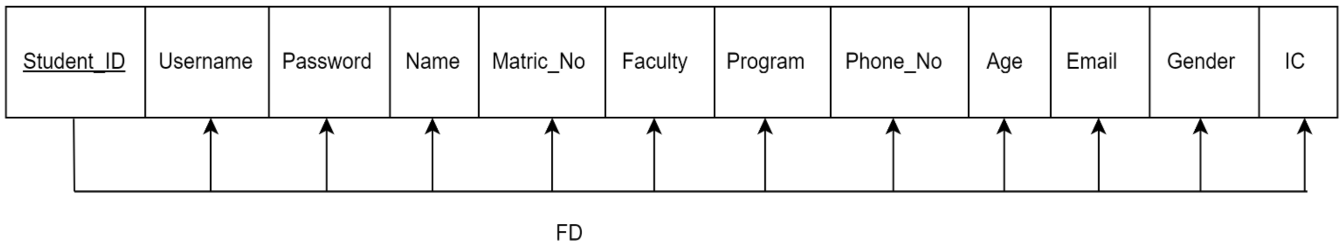


Figure 4.2 : The normalization of student

Based on figure 4.2, the normalization of the student entity is a critical step in ensuring that the data is organized efficiently and redundancy is minimized. By applying normalization techniques, all attributes related to the student, such as U**sername, Password, Name, Matric_No, Faculty, Program, Phone_No, Age, Email, Gender,** and **IC (Identification Card)**, are made functionally dependent on the primary key, which in this case is the **Student ID**. This approach ensures that each piece of data is stored only once, and any changes made to the data are reflected consistently across the system.

For instance, if a student changes their contact number or email address, the system will update the information in a single location, preventing discrepancies across multiple records. By eliminating any repetitive or unnecessary data storage, normalization also enhances the system's performance by reducing storage requirements and improving query efficiency. Furthermore, this structured organization of student data ensures that the database is easier to maintain and less prone to anomalies during updates, deletions, or insertions. Overall, the normalization of the student entity contributes significantly to data integrity and the long-term sustainability of the system.

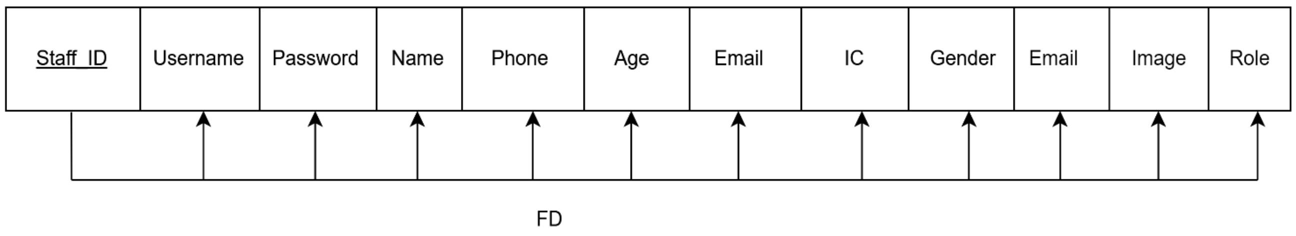


Figure 4.3 : The normalization of staff

Based on figure 4.3, the normalization of the staff entity plays a crucial role in ensuring that the data is structured efficiently and free from redundancy. In this process, all staff-related attributes such as **Username, Password, Name, Phone, Age, Email, Gender, IC (Identification Card), Image**, and **Role** are made functionally dependent on the **Staff ID**, which serves as the primary key. This means that each piece of information associated with a particular staff member is linked directly to their unique **Staff ID**, ensuring that no data is unnecessarily duplicated across the system.

By ensuring that attributes like **Phone, Email,** and **Role** are stored in a single, well-organized table, the system eliminates the potential for inconsistencies. For example, if a staff member changes their contact details or role, the system can update this information in one location, ensuring the entire database reflects these changes accurately without requiring multiple updates in various places.

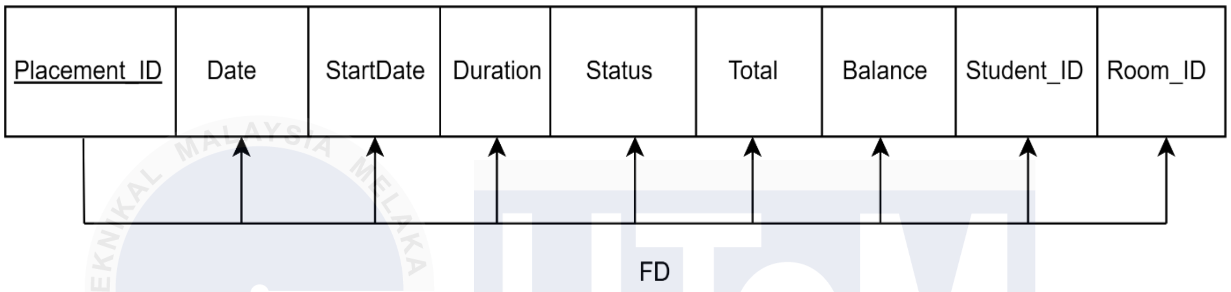


Figure 4.4 : The normalization of placement

Based on figure 4.4, the normalization of the placement entity is a crucial step in ensuring that the data is organized efficiently and without redundancy. This process involves ensuring that all placement-related attributes such as **Date, StartDate, Duration, Status, Total**, and **Balance** are functionally dependent on the primary key, which is the **Placement ID**. The **Placement ID** serves as a unique identifier for each placement record, ensuring that each placement is distinct and can be easily referenced within the system.

Additionally, the inclusion of the **Student_ID** and **Room_ID** attributes establishes key relationships between the placement entity and the respective students and rooms. The **Student_ID** links the placement record to a specific student, ensuring that all information related to the student's placement such as the duration of their stay, the room they are assigned to, and the financial details like total cost and balance is stored in one place. Similarly, the **Room_ID** connects the placement record to the assigned room, which is essential for tracking room allocation and availability.

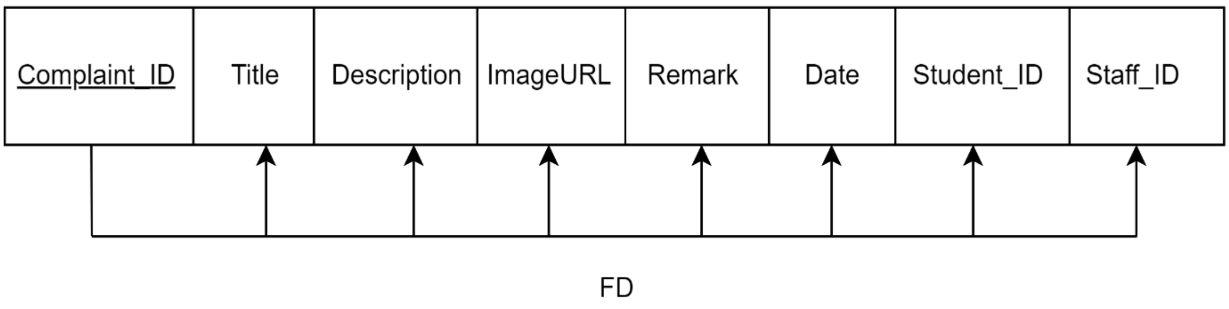


Figure 4.5 : The normalization of complaint

Based on figure 4.5, the normalization of the complaint entity is an essential step in organizing complaint-related data efficiently while eliminating redundancy. In this process, all attributes related to a complaint such as **Title, Description, ImageURL, Remark,** and **Date** are made functionally dependent on the **Complaint ID**, which serves as the primary key. This ensures that each complaint is uniquely identifiable and that its associated data, including the description of the issue and any related images or remarks, is directly linked to this unique identifier.

Furthermore, the inclusion of **Student_ID** and **Staff_ID** establishes crucial relationships between the complaint entity and both the student who lodged the complaint and the staff member responsible for handling it. The **Student_ID** ensures that the complaint is tied to a specific student, making it easy to track which students have submitted complaints and the details of their grievances. On the other hand, the **Staff_ID** links the complaint to the staff member assigned to address or resolve the issue, facilitating better tracking of complaint resolution and accountability.

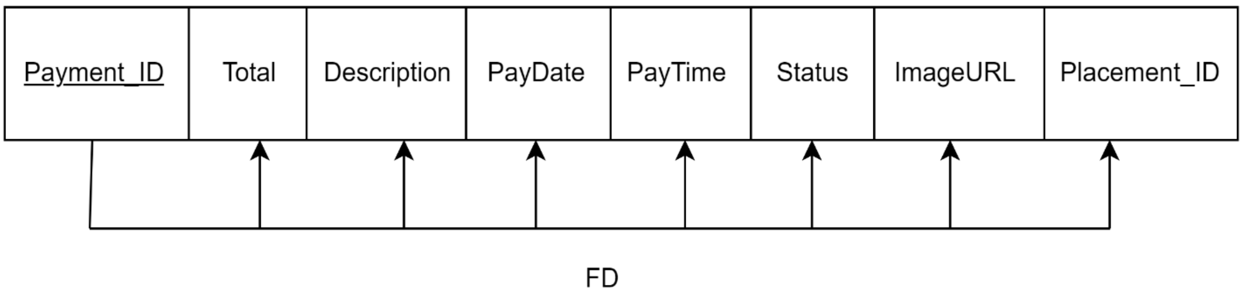


Figure 4.6 : The normalization of payment

In the normalization process depicted in Figure 4.6, the payment entity undergoes a structured transformation to ensure that all attributes related to payments namely **Total, Description, PayDate, PayTime, Status,** and **ImageURL** are functionally dependent on the primary key, which is the **Payment ID**. This normalization process is essential for reducing redundancy and enhancing data integrity within the database.

The **Payment ID** serves as the unique identifier for each payment entry, ensuring that each attribute is correctly linked to a specific payment record. Additionally, the inclusion of **Placement_ID** in this schema establishes a crucial relationship between the payment entity and the projects associated with these payments. This relationship is fundamental for tracking and managing the financial transactions related to specific projects, thereby providing a comprehensive view of payment activities in relation to their respective projects.

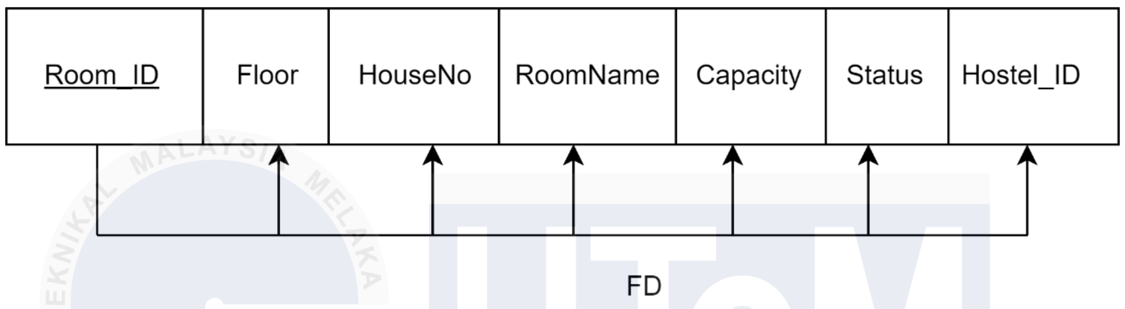


Figure 4.7 : The normalization of room

In the normalization process illustrated in Figure 4.7, the room entity is carefully organized to ensure that all attributes associated with rooms specifically **Floor, HouseNo, RoomName, Capacity,** and **Status** are functionally dependent on the primary key, which is the **Room ID**. This normalization is a critical step in optimizing the database structure, as it effectively reduces redundancy and enhances the accuracy of data storage and retrieval.

The **Room ID** serves as the unique identifier for each room record, ensuring that every room-related attribute is consistently linked to a specific room entry. This means that attributes such as Floor, **HouseNo, RoomName, Capacity,** and **Status** are all tied directly to the **Room ID**, ensuring that each room's details are uniquely and accurately represented.

Furthermore, the inclusion of **Hostel_ID** in the schema plays a vital role by establishing a relationship between the room entity and the hostels where these rooms are located. This relationship is crucial for effectively managing and tracking rooms within the context of their respective hostels. By associating each room with a specific **Hostel_ID**, the database design facilitates a clear and organized view of room allocations within different hostels, allowing for more efficient management and administration of hostel resources.

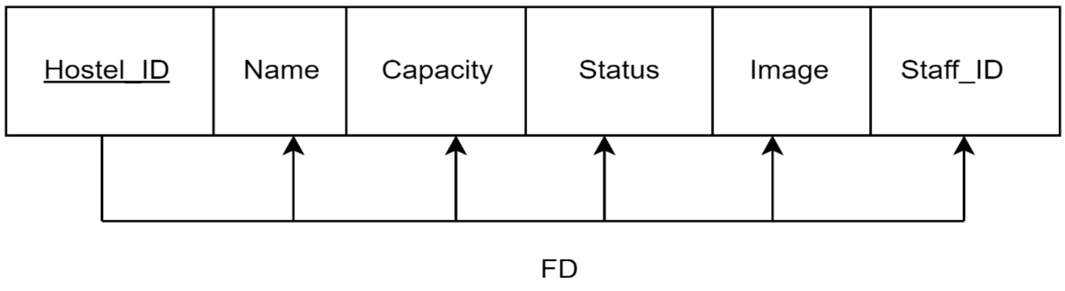


Figure 4.8 : The normalization of hostel

In the normalization process depicted in Figure 4.8, the hostel entity is meticulously structured to ensure that all attributes related to the hostel namely **Name, Capacity, Status,** and **Image** are functionally dependent on the primary key, which is the **Hostel ID**. This normalization is crucial for achieving an organized and efficient database design, as it minimizes redundancy and enhances the integrity of data within the system.

The **Hostel ID** serves as the unique identifier for each hostel record, ensuring that each attribute associated with the hostel is directly and accurately linked to a specific hostel entry. By making sure that attributes such as **Name, Capacity, Status,** and **Image** are dependent on the **Hostel ID**, the database design guarantees that each hostel's details are consistently and uniquely represented. This setup prevents issues such as duplicate records or inconsistent data, thereby maintaining a high level of data quality.

Additionally, the schema includes **Staff_ID**, which establishes a critical relationship between the hostel entity and the staff members assigned to manage or oversee the hostel. This relationship is essential for linking hostels with their respective staff, thereby facilitating a clear and organized connection between the management team and the hostel operations. By associating each hostel with a specific **Staff_ID**, the database design enables effective management and monitoring of hostel activities, ensuring that staff responsibilities and assignments are well-documented and managed.

4.2.2.2 Data Dictionary

Table 4.1 : Table Student data dictionary

Table Name	**Attribute Name**	**Data Type**	**Required**	**PK or FK**	**FK (Referenced table)**
Student	Student_ID	Int (11)	Yes	PK	
	Username	Varchar (30)	Yes		
	Password	text	Yes		
	Name	Varchar (255)	Yes		
	Matric_No	Varchar (30)	Yes		
	Faculty	Varchar (11)	Yes		
	Program	Varchar (11)	Yes		
	Phone	Varchar (30)	Yes		
	Age	Int (11)	Yes		
	Email	Varchar (40)	Yes		
	IC	Int (15)	Yes		
	Gender	Varchar (11)	Yes		
	Avatar	Text	Yes		

Table 4.2 : Table Room data dictionary

Table Name	**Attribute Name**	**Data Type**	**Required**	**PK or FK**	**FK (Referenced table)**
Room	Room_ID	Int (11)	Yes	PK	
	Floor	Int (1)	Yes		
	House_No	Int (2)	Yes		
	Room_Name	Char (1)	Yes		
	Capacity	Int (1)	Yes		
	Status	Int (11)	Yes		
	Hostel ID	Int (11)	Yes	FK	Hostel

Table 4.3 : Table Staff data dictionary

Table Name	Attribute Name	Data Type	Required	PK or FK	FK (Referenced table)
Staff	Staff_ID	Int (11)	Yes	PK	
	Username	Varchar (30)	Yes		
	Password	Text	Yes		
	Name	Varchar (255)	Yes		
	Phone	Varchar (30)	Yes		
	Age	Int (11)	Yes		
	Email	Varchar (40)	Yes		
	Image	Mediumblob	Yes		
	IC	Int (15)	Yes		
	Gender	Varchar (11)	Yes		
	Role	Varchar (11)	Yes		

Table 4.4 : Table Complaint data dictionary

Table Name	Attribute Name	Type	Required	PK or FK	FK (Referenced table)
Complaint	Complaint_ID	Int (11)	Yes	PK	
	Title	Varchar (255)	Yes		
	Description	Text	Yes		
	ImageURL	Text	Yes		
	Remark	Varchar (255)	Yes		
	Student_ID	Int (11)	Yes	FK	Student
	Staff_ID	Int (11)	Yes	FK	Staff

Table 4.5 : Table Placement data dictionary

Table Name	Attribute Name	Data Type	Required	PK or FK	FK (Referenced table)
Placement	Placement_ID	Int (11)	Yes	PK	
	Date	Date	Yes		
	StartDate	Date	Yes		
	Duration	Int (2)	Yes		
	Total	Decimal (10,2)	Yes		
	Balance	Decimal (10,2)	Yes		
	Status	Int (11)	Yes		
	Student_ID	Int (11)	Yes	FK	Student
	Room_ID	Int (11)	Yes	FK	Room

Table 4.6 : Table Payment data dictionary

Table Name	Attribute Name	Data Type	Required	PK or FK	FK (Referenced table)
Payment	Payment_ID	Int (11)	Yes	PK	
	Total	Decimal (6,2)	Yes		
	Description	Varchar (255)	Yes		
	PayDate	Date	Yes		
	PayTime	Time	Yes		
	Status	Int (11)	Yes		
	ImageURL	Text	Yes		
	Placement_ID	Int (11)	Yes	FK	Placement

Table 4.7 : Table Hostel data dictionary

Table Name	Attribute Name	Data Type	Required	PK or FK	FK (Referenced table)
Hostel	Hostel_ID	Int (11)	Yes	PK	
	Name	Varchar (255)	Yes		
	Capacity	Int (11)	Yes		
	Status	Int (11)	Yes		
	Staff_ID	Int (11)	Yes	FK	Staff

4.2.3 Physical Database Design

i) DDL (Create Table)

```
CREATE TABLE `student` (
  `Student_ID` int(11) NOT NULL,
  `Username` varchar(30) NOT NULL,
  `Password` text NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Matric_No` varchar(30) DEFAULT NULL,
  `Faculty` varchar(11) DEFAULT NULL,
  `Program` varchar(11) DEFAULT NULL,
  `Phone_No` varchar(30) NOT NULL,
  `Age` int(11) DEFAULT NULL,
  `Email` varchar(40) DEFAULT NULL,
  `Gender` varchar(11) DEFAULT NULL,
  `Avatar` text DEFAULT NULL
)
```

```
CREATE TABLE `staff` (
  `Staff_ID` int(11) NOT NULL,
  `Username` varchar(30) NOT NULL,
  `Password` text NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Phone` varchar(30) NOT NULL,
  `Age` int(11) NOT NULL,
  `Email` varchar(40) NOT NULL,
  `Image` mediumblob DEFAULT NULL
)
```

```
CREATE TABLE `room` (
  `Room_ID` int(11) NOT NULL,
  `Floor` int(1) NOT NULL,
  `House_No` int(2) NOT NULL,
  `Room_Name` char(1) NOT NULL,
  `Capacity` int(1) NOT NULL DEFAULT 2,
  `Status` int(11) DEFAULT 1,
  `Hostel_ID` int(11) DEFAULT NULL
)
```

```
CREATE TABLE `payment` (
  `Payment_ID` int(11) NOT NULL,
  `Total` decimal(6,2) NOT NULL,
  `Description` varchar(255) DEFAULT NULL,
  `PayDate` date NOT NULL,
  `PayTime` time NOT NULL,
  `Status` int(11) DEFAULT 2,
  `ImageURL` text NOT NULL,
  `Placement_ID` int(11) NOT NULL
)
```

```
CREATE TABLE `placement` (
  `Placement_ID` int(11) NOT NULL,
  `Date` date NOT NULL,
  `StartDate` date NOT NULL,
  `Duration` int(2) NOT NULL,
  `Total` decimal(10,2) NOT NULL,
  `Balance` decimal(10,2) NOT NULL,
  `Status` int(11) DEFAULT 2,
  `Student_ID` int(11) DEFAULT NULL,
  `Room_ID` int(11) DEFAULT NULL
)
```

```
CREATE TABLE `hostel` (
  `Hostel_ID` int(11) NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Capacity` int(11) NOT NULL,
  `Status` int(11) NOT NULL,
  `Staff_ID` int(11) DEFAULT NULL,
  `Address_ID` int(11) DEFAULT NULL
)
```

```
CREATE TABLE `complaint` (
  `Complaint_ID` int(11) NOT NULL,
  `Title` varchar(255) NOT NULL,
  `Description` text NOT NULL,
  `Remark` varchar(255) DEFAULT NULL,
  `ImageURL` text NOT NULL,
  `Date` date DEFAULT NULL,
  `Student_ID` int(11) NOT NULL,
  `Staff_ID` int(11) DEFAULT NULL
)
```

```
Constraints for table `complaint`

ALTER TABLE `complaint`
  ADD CONSTRAINT `complaint_ibfk_1` FOREIGN KEY
(`Student_ID`) REFERENCES `student` (`Student_ID`),
  ADD CONSTRAINT `complaint_ibfk_2` FOREIGN KEY
(`Staff_ID`) REFERENCES `staff` (`Staff_ID`);

Constraints for table `hostel`

ALTER TABLE `hostel`
  ADD CONSTRAINT `hostel_ibfk_1` FOREIGN KEY
(`Staff_ID`) REFERENCES `staff` (`Staff_ID`),
  ADD CONSTRAINT `hostel_ibfk_2` FOREIGN KEY
(`Address_ID`) REFERENCES `address` (`Address_ID`);

Constraints for table `payment`

ALTER TABLE `payment`
  ADD CONSTRAINT `payment_ibfk_1` FOREIGN KEY
(`Placement_ID`) REFERENCES `placement`
(`Placement_ID`);


Constraints for table `placement`

ALTER TABLE `placement`
  ADD CONSTRAINT `placement_ibfk_1` FOREIGN KEY
(`Student_ID`) REFERENCES `student` (`Student_ID`),
  ADD CONSTRAINT `placement_ibfk_2` FOREIGN KEY
(`Room_ID`) REFERENCES `room` (`Room_ID`);


Constraints for table `room`

ALTER TABLE `room`
  ADD CONSTRAINT `room_ibfk_1` FOREIGN KEY (`Hostel_ID`)
REFERENCES `hostel` (`Hostel_ID`);
```

ii) DML

a) Insert Statement

```
INSERT INTO `student` (`Student_ID`, `Username`,
`Password`, `Name`, `Matric_No`, `Faculty`, `Program`,
`Phone_No`, `Age`, `Email`, `Gender`, `Avatar`) VALUES
(1, 'faizal',
'$2y$10$a6SC/PE1g10REr5GZBQuR.Mt49yTxPg/0QlbmUNPcFmjuRh
twwzJO', 'Muhammad Faizal Bin Abdullah', 'B032110321',
'5', 'BITD', '01952331421', 21,
'b032110321@student.utem.edu.my', '1',
'../student/avatar/null.png');
```

```
INSERT INTO `placement` (`Placement_ID`, `Date`,
`StartDate`, `Duration`, `Total`, `Balance`, `Status`,
`Student_ID`, `Room_ID`) VALUES
(19, '2024-06-17', '2024-06-17', 2, '0.00', '260.00',
1, 1, 34),
(20, '2024-06-19', '2024-06-19', 3, '0.00', '390.00',
2, 1, 19),
(21, '2024-06-19', '2024-06-20', 1, '0.00', '130.00',
1, 6, 24),
(22, '2024-06-19', '2024-06-19', 6, '0.00', '780.00',
3, 9, 43);
```

b) Update Statement

```
UPDATE `hostel`

SET `Hostel_ID`='12',`Name`='LEKIR',`Capacity`=
300,`Status`='Available',`Staff_ID`='2',`image`='170391
50591703044692hostel-malaysia-
4.jpg',`description`='Gender: Female'

WHERE Hostel_ID = '4';
```

```
UPDATE `staff`

SET
`Staff_ID`='1',`Username`='amin',`Password`='$2y$10$0Kx
eOxs2T9av0FpLYCeQeO/ueb5pntv7yoLaZVbs6ow',`Name`='AminA
bid',`Phone`='01291029120',`Age`='32',`Email`='amin@gma
il.com',`Image`='desk-staff-png-image_401612265_wh1200'

WHERE Staff_ID = 1;
```

c) Delete Statement

```
DELETE FROM `complaint` WHERE Complaint_ID = 2;
```

```
DELETE FROM `room` WHERE Room_ID = 15;
```

d) Trigger

```
CREATE TRIGGER `after_payment_update` AFTER UPDATE ON
`payment` FOR EACH ROW BEGIN
    IF OLD.Status = 2 AND NEW.Status = 1 THEN
        -- Update the placement table
        UPDATE placement
        SET Total = Total + NEW.Total,
            Balance = Balance - NEW.Total
        WHERE Placement_ID = NEW.Placement_ID;
    END IF;
END
```

```
CREATE TRIGGER `SetPayTimeBeforeInsert` BEFORE INSERT
ON `payment` FOR EACH ROW BEGIN
    SET NEW.PayTime = CURRENT_TIME();
END
```

```
CREATE TRIGGER `SetPayDateBeforeInsert` BEFORE INSERT
ON `payment` FOR EACH ROW BEGIN
    SET NEW.PayDate = CURDATE();
END
```

```
CREATE TRIGGER `update_room_capacity` BEFORE UPDATE ON
`room` FOR EACH ROW BEGIN
    IF NEW.Capacity = 0 THEN
        SET NEW.Status = 2;
    ELSE
        SET NEW.Status = 1;
    END IF;
END
```

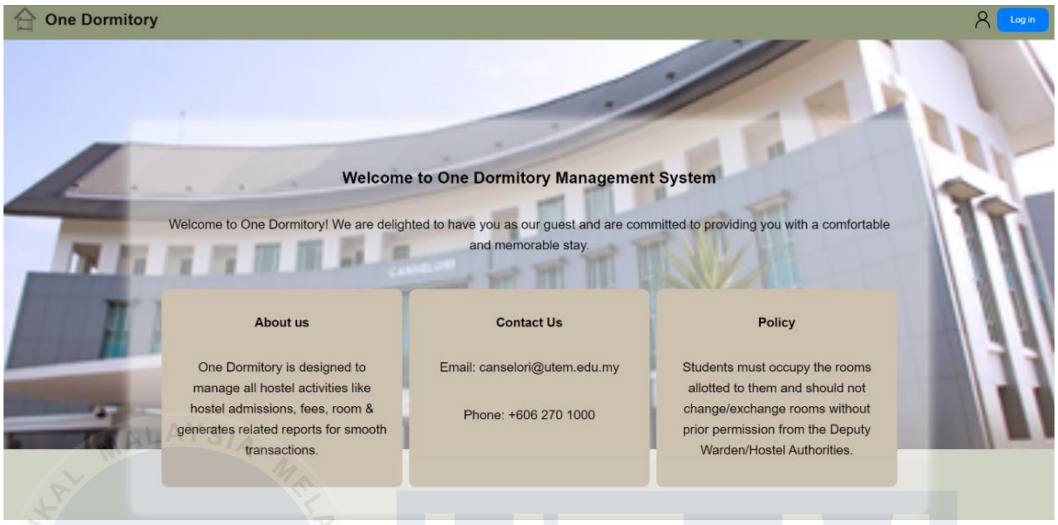
e) Join Table

```
SELECT

    student.Name,

    hostel.Name

FROM student

JOIN placement ON student.Student_ID =
placement.Student_ID

JOIN room ON placement.Room_ID = room.Room_ID

JOIN hostel ON room.Hostel_ID = hostel.Hostel_ID;
```

f) Procedure

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`RegisterStudent` (
IN `p_Username` VARCHAR(30),
IN `p_Name` VARCHAR(255),
IN `p_Matric_No` VARCHAR(30),
IN `p_Faculty` VARCHAR(11),
IN `p_Program` VARCHAR(30),
IN `p_Phone_No` VARCHAR(30),
IN `p_Age` INT,
IN `p_Gender` VARCHAR(11),
IN `p_Email` VARCHAR(40),
IN `p_Password` TEXT,
IN `p_Avatar` TEXT)

BEGIN
    INSERT INTO student (Username, Name, Matric_No,
Faculty, Program, Phone_No, Age, Gender, Email,
Password, Avatar)
VALUES (p_Username, p_Name, p_Matric_No, p_Faculty,
p_Program, p_Phone_No, p_Age, p_Gender, p_Email,
p_Password, p_Avatar);
END
```

4.3 User Interface


One Dormitory

Log in

Welcome to One Dormitory Management System

Welcome to One Dormitory! We are delighted to have you as our guest and are committed to providing you with a comfortable and memorable stay.

About us

One Dormitory is designed to manage all hostel activities like hostel admissions, fees, room & generates related reports for smooth transactions.

Contact Us

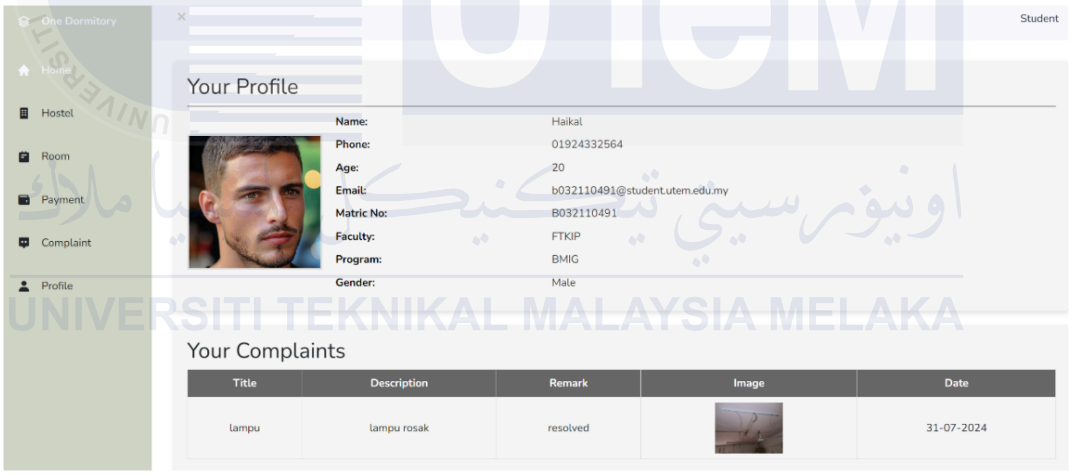
Email: canselori@utem.edu.my

Phone: +606 270 1000

Policy

Students must occupy the rooms allotted to them and should not change/exchange rooms without prior permission from the Deputy Warden/Hostel Authorities.

Figure 4.9 : The mainpage of the system



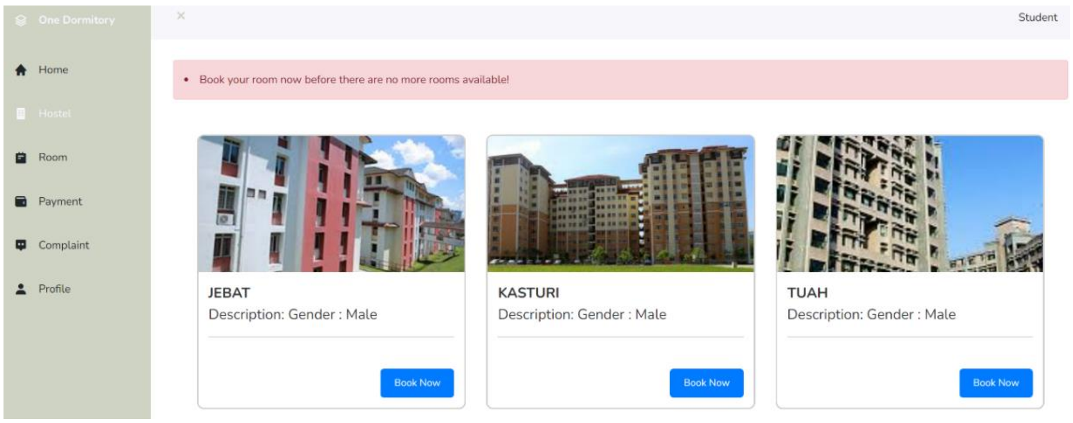
Your Profile

Name:	Haikal
Phone:	01924332564
Age:	20
Email:	b032110491@student.utem.edu.my
Matric No:	B032110491
Faculty:	FTKIP
Program:	BMIG
Gender:	Male

Your Complaints

Title	Description	Remark	Image	Date
lampu	lampu rosak	resolved		31-07-2024

Figure 4.10 : The homepage of the student



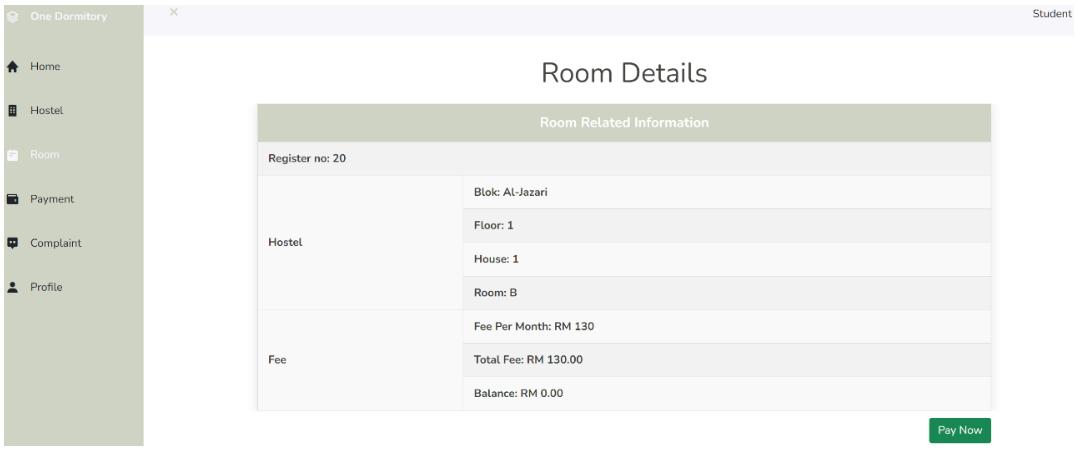
• Book your room now before there are no more rooms available!

JEBAT
Description: Gender : Male
Book Now

KASTURI
Description: Gender : Male
Book Now

TUAH
Description: Gender : Male
Book Now

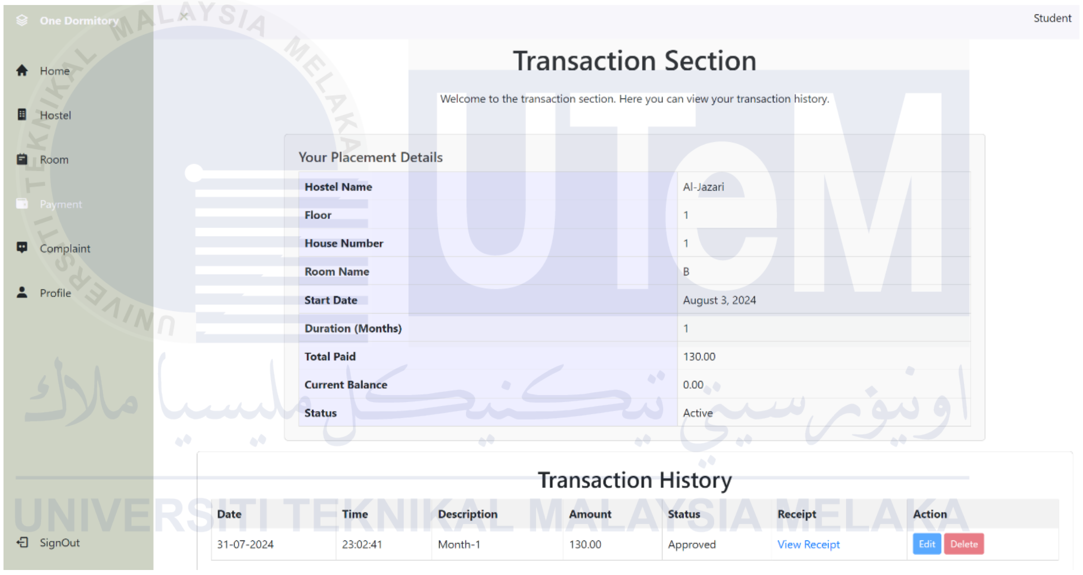
Figure 4.11 : The hostel placement page for student



Room Details

Room Related Information	
Register no: 20	
Hostel	Blok: Al-Jazari
	Floor: 1
	House: 1
	Room: B
Fee	Fee Per Month: RM 130
	Total Fee: RM 130.00
	Balance: RM 0.00

Figure 4.12 : The room details page for student



Transaction Section

Welcome to the transaction section. Here you can view your transaction history.

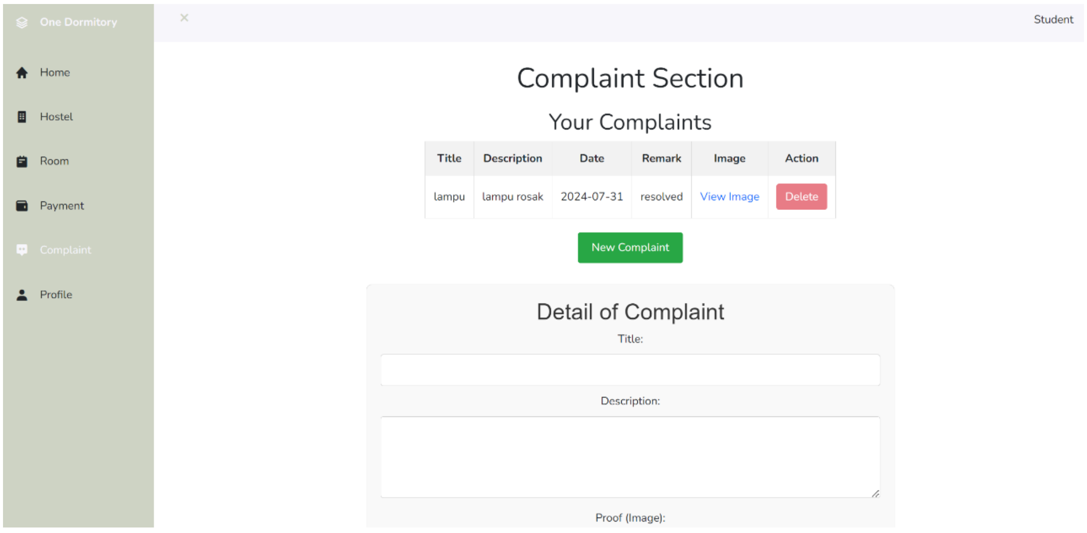
Your Placement Details

Hostel Name	Al-Jazari
Floor	1
House Number	1
Room Name	B
Start Date	August 3, 2024
Duration (Months)	1
Total Paid	130.00
Current Balance	0.00
Status	Active

Transaction History

Date	Time	Description	Amount	Status	Receipt	Action
31-07-2024	23:02:41	Month-1	130.00	Approved	View Receipt	Edit Delete

Figure 4.13 : The transaction page for student



Complaint Section

Your Complaints

Title	Description	Date	Remark	Image	Action
lampu	lampu rosak	2024-07-31	resolved	View Image	Delete

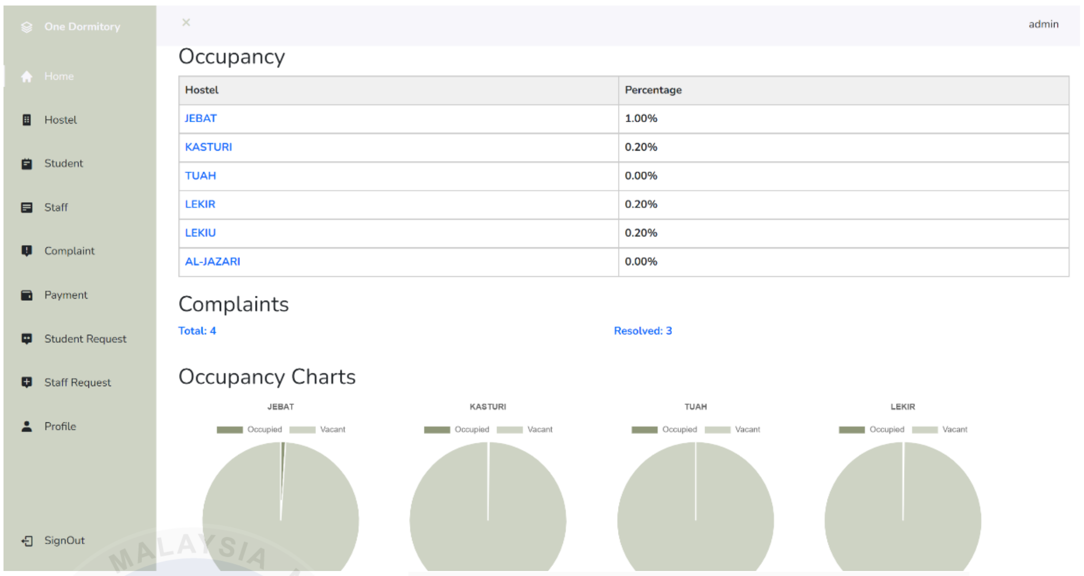
Detail of Complaint

Title:

Description:

Proof (Image):

Figure 4.14 : The complaint section for student



Occupancy

Hostel	Percentage
JEBAT	1.00%
KASTURI	0.20%
TUAH	0.00%
LEKIR	0.20%
LEKIU	0.20%
AL-JAZARI	0.00%

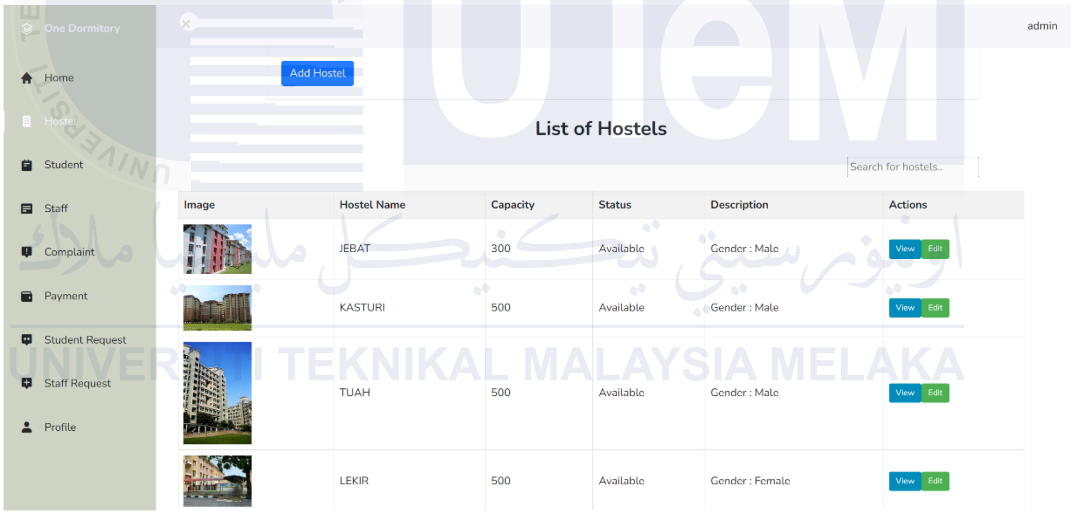
Complaints

Total: 4

Resolved: 3

Occupancy Charts

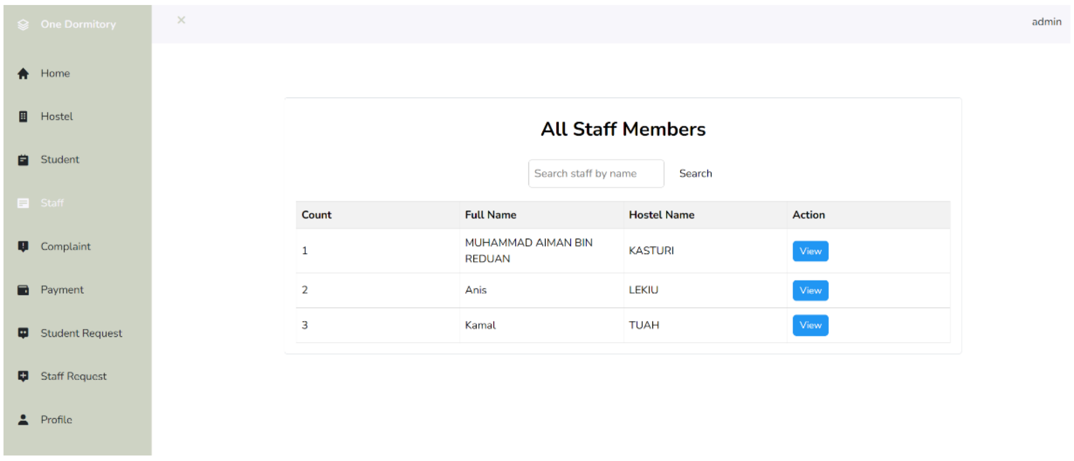
Figure 4.15 : The homepage of the admin



List of Hostels

Image	Hostel Name	Capacity	Status	Description	Actions
	JEBAT	300	Available	Gender : Male	View Edit
	KASTURI	500	Available	Gender : Male	View Edit
	TUAH	500	Available	Gender : Male	View Edit
	LEKIR	500	Available	Gender : Female	View Edit

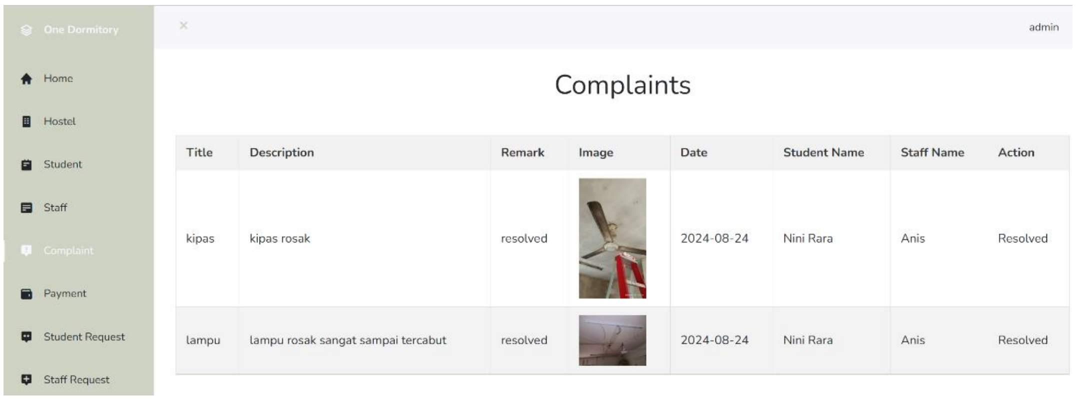
Figure 4.16 : The list of hostel page for admin



All Staff Members

Count	Full Name	Hostel Name	Action
1	MUHAMMAD AIMAN BIN REDUAN	KASTURI	View
2	Anis	LEKIU	View
3	Kamal	TUAH	View

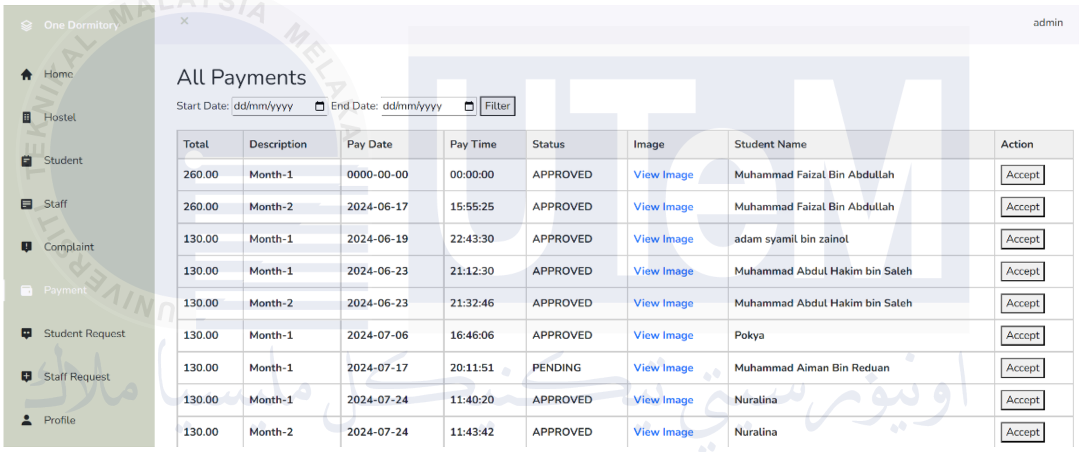
Figure 4.17 : The list of staff members for admin



Complaints

Title	Description	Remark	Image	Date	Student Name	Staff Name	Action
kipas	kipas rosak	resolved		2024-08-24	Nini Rara	Anis	Resolved
lampu	lampu rosak sangat sampai tercabut	resolved		2024-08-24	Nini Rara	Anis	Resolved

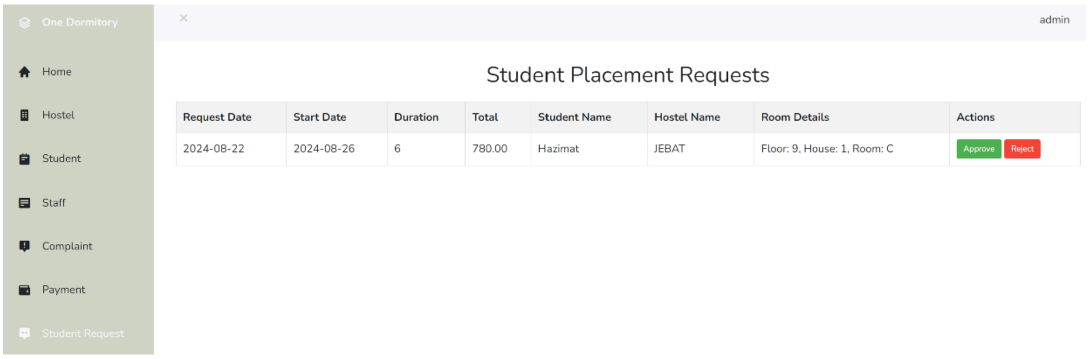
Figure 4.18 : The complaint page for admin



All Payments

Total	Description	Pay Date	Pay Time	Status	Image	Student Name	Action
260.00	Month-1	0000-00-00	00:00:00	APPROVED	View Image	Muhammad Faizal Bin Abdullah	Accept
260.00	Month-2	2024-06-17	15:55:25	APPROVED	View Image	Muhammad Faizal Bin Abdullah	Accept
130.00	Month-1	2024-06-19	22:43:30	APPROVED	View Image	adam syamil bin zainol	Accept
130.00	Month-1	2024-06-23	21:12:30	APPROVED	View Image	Muhammad Abdul Hakim bin Saleh	Accept
130.00	Month-2	2024-06-23	21:32:46	APPROVED	View Image	Muhammad Abdul Hakim bin Saleh	Accept
130.00	Month-1	2024-07-06	16:46:06	APPROVED	View Image	Pokya	Accept
130.00	Month-1	2024-07-17	20:11:51	PENDING	View Image	Muhammad Aiman Bin Reduan	Accept
130.00	Month-1	2024-07-24	11:40:20	APPROVED	View Image	Nuralina	Accept
130.00	Month-2	2024-07-24	11:43:42	APPROVED	View Image	Nuralina	Accept

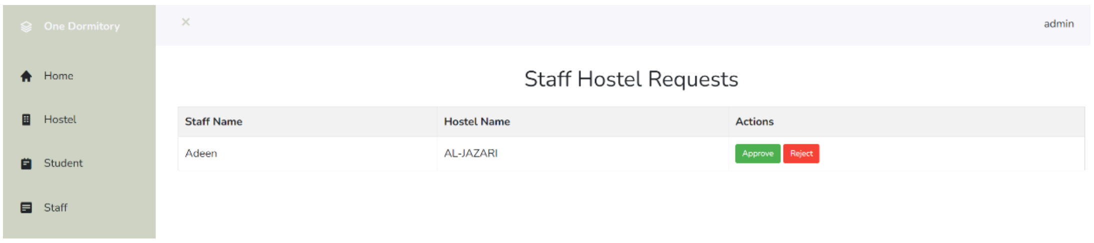
Figure 4.19 : The transaction page for admin



Student Placement Requests

Request Date	Start Date	Duration	Total	Student Name	Hostel Name	Room Details	Actions
2024-08-22	2024-08-26	6	780.00	Hazimat	JEBAT	Floor: 9, House: 1, Room: C	Approve Reject

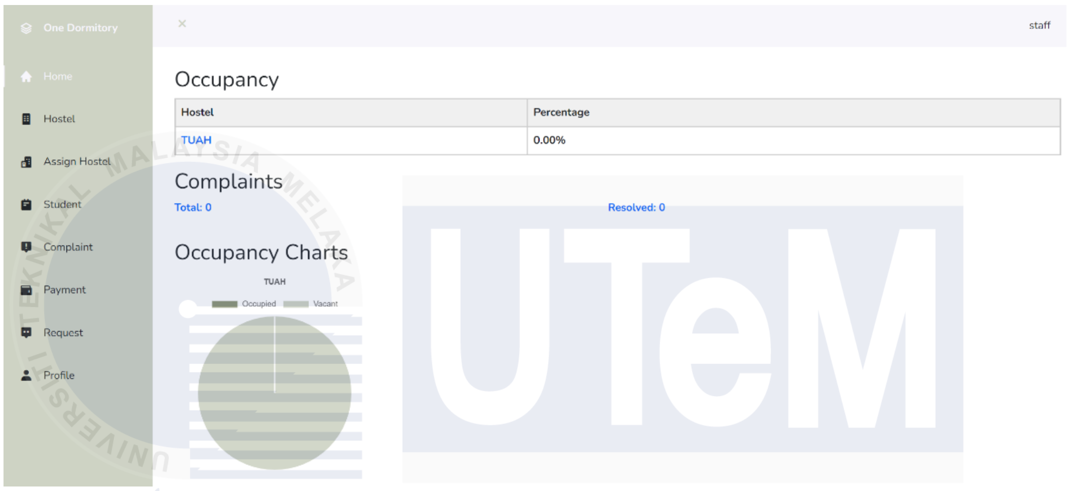
Figure 4.20 : The placement request page for admin



Staff Hostel Requests

Staff Name	Hostel Name	Actions
Adeen	AL-JAZARI	Approve Reject

Figure 4.21 : The hostel request page for admin



Occupancy

Hostel	Percentage
TUAH	0.00%

Complaints

Total: 0

Resolved: 0

Figure 4.22 : The homepage of the staff

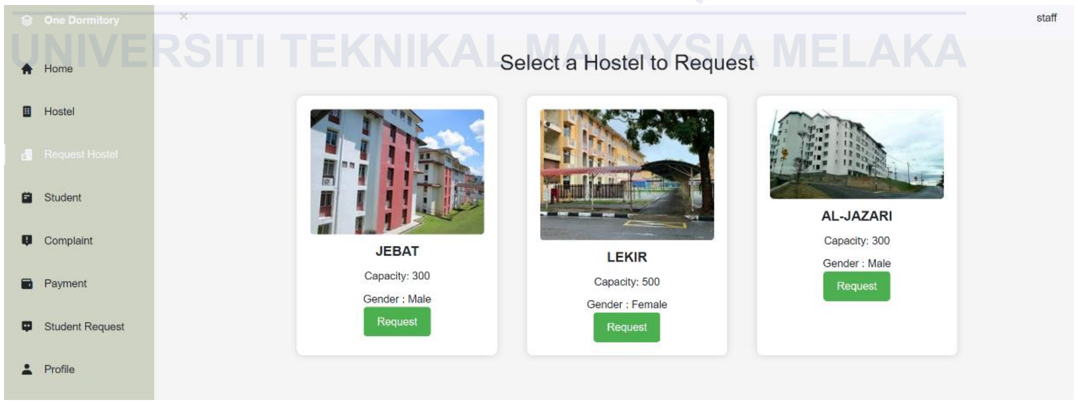


Figure 4.23 : The hostel request page for staff

4.4 Summary

In conclusion, a carefully crafted database design forms the backbone of an effective house rental management system. By addressing the unique needs of individual managers, incorporating scalability and security measures, and supporting efficient data management, the database design significantly contributes to the application's success and longevity.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

During the implementation phase, it is essential to ensure that the system is not only installed and configured correctly but also that it is transitioned smoothly from the development stage to the testing phase. This phase is critical as it involves placing the developed system into a beta stage, where it undergoes rigorous testing and bug fixing. This transition is a pivotal moment in the project lifecycle, marking the shift from development to production readiness. The chapter is meticulously divided into four subtopics: software development environment setup, database implementation, software configuration management, and implementation status. Within the software development environment setup, two key subtopics will be explored in detail, focusing on the establishment of a robust development environment and the preparation necessary for a successful implementation. These subtopics are vital as they lay the groundwork for the system's effective deployment and subsequent testing, ensuring that the system is prepared for real-world application and operational success.

5.2 Software Development Environment Setup

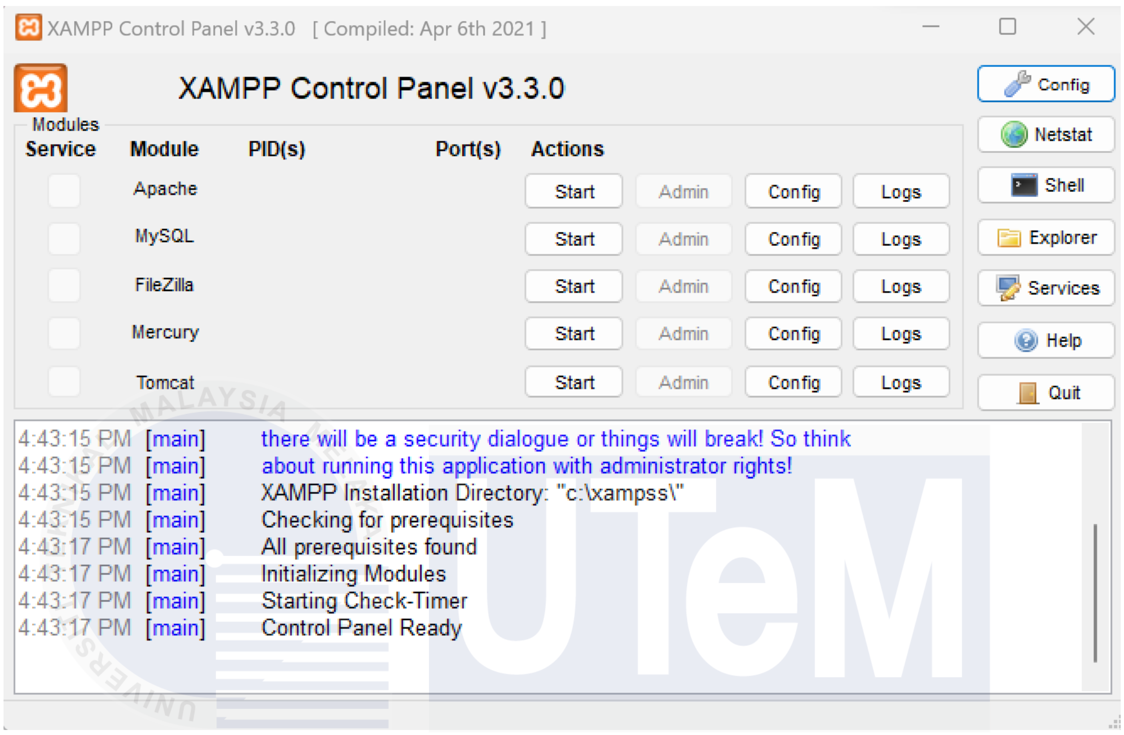
This subtopic will go over the environmental architecture that was mentioned in Chapter 3. During the phase, Microsoft Visual Code will be utilized as the IDE (Integrated Development Environment) and XAMPP as the database management system (DBMS).

Table 5.1 : List of hardware, software, database requirement

Hardware Requirement	• Laptop (Acer Spin 3) • Windows 10 Home 64-bit • Intel® Core™ i5-1135G7 processor Quad-core 2.40 GHz • 13.3" WQXGA (2560 x 1600) 16:10 IPS Touchscreen • 8 GB, LPDDR4X • 512 GB SSD
Software Requirement	• Microsoft Visual Code • XAMPP
Database Requirement	• PHPMyAdmin (MySQL)

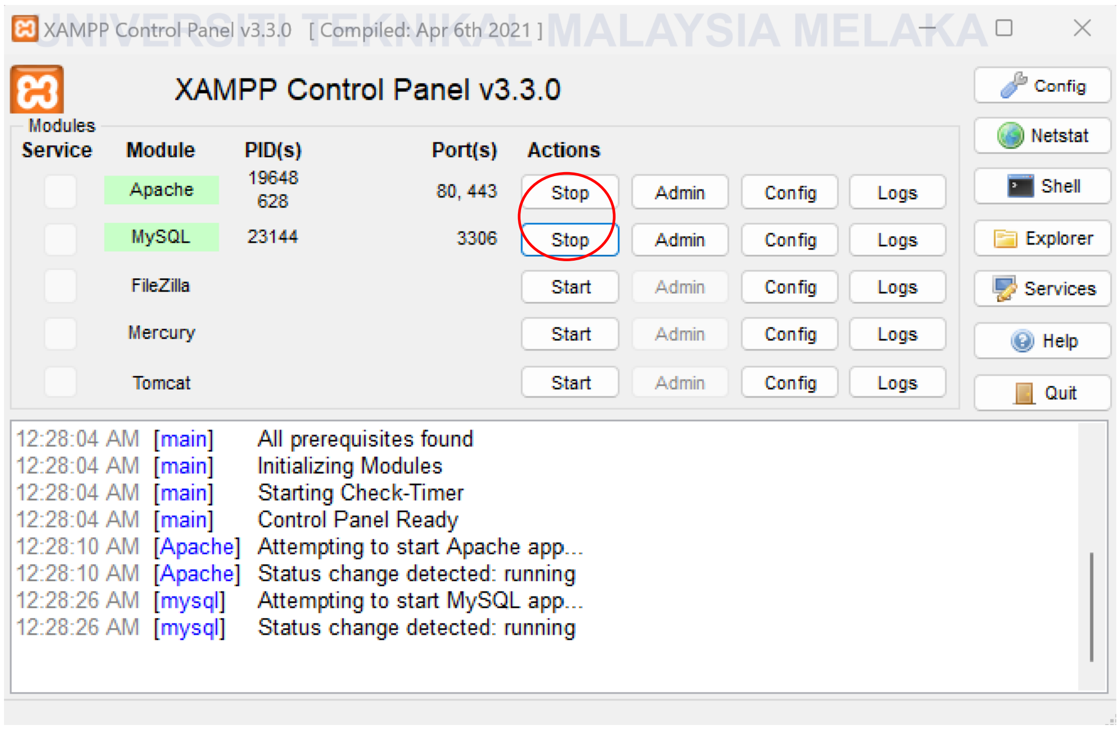
5.3 Database Implementation

This section explains how to activate the Xampp service.



```
4:43:15 PM  [main]   there will be a security dialogue or things will break! So think
4:43:15 PM  [main]   about running this application with administrator rights!
4:43:15 PM  [main]   XAMPP Installation Directory: "c:\xampss\"
4:43:15 PM  [main]   Checking for prerequisites
4:43:17 PM  [main]   All prerequisites found
4:43:17 PM  [main]   Initializing Modules
4:43:17 PM  [main]   Starting Check-Timer
4:43:17 PM  [main]   Control Panel Ready
```

Figure 5.1 : Starting the Xampp Services in Local Computer


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

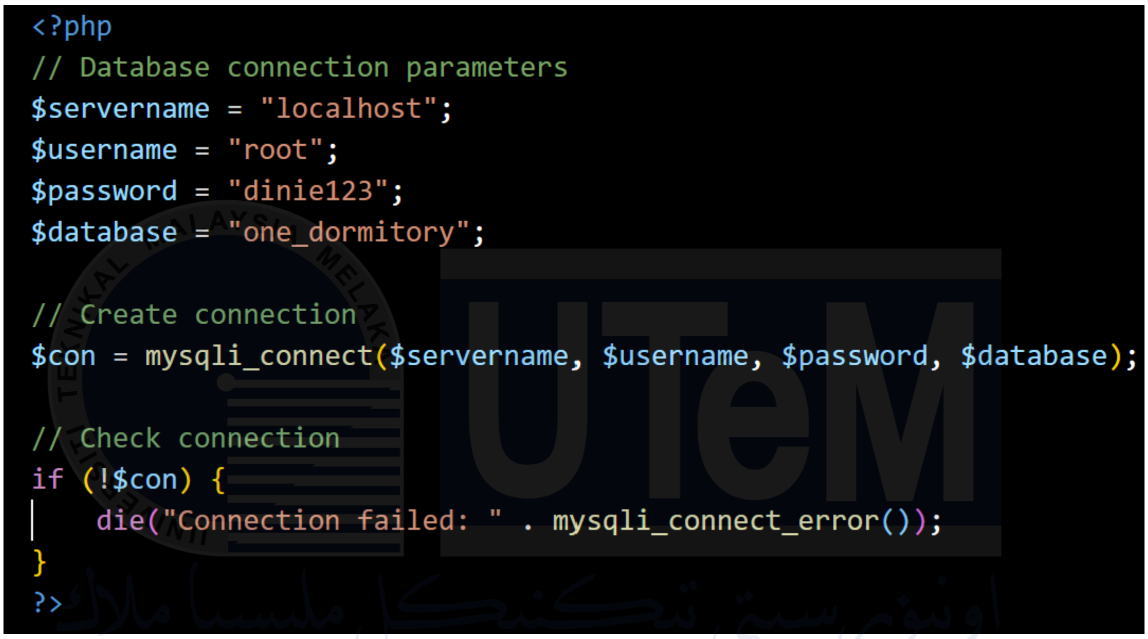
Service	Module	PID(s)	Port(s)	Actions			
	Apache	19648 628	80, 443	Stop	Admin	Config	Logs
	MySQL	23144	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
12:28:04 AM  [main]     All prerequisites found
12:28:04 AM  [main]     Initializing Modules
12:28:04 AM  [main]     Starting Check-Timer
12:28:04 AM  [main]     Control Panel Ready
12:28:10 AM  [Apache]   Attempting to start Apache app...
12:28:10 AM  [Apache]   Status change detected: running
12:28:26 AM  [mysql]    Attempting to start MySQL app...
12:28:26 AM  [mysql]    Status change detected: running
```

Figure 5.2 : Click start at Apache and MySQL

Step:

1. Open XAMPP Control Panel: Launch the XAMPP Control Panel from your start menu or applications folder.
2. Start Apache and MySQL: Click "Start" next to Apache and MySQL to begin the web server and database server.



```php
<?php
// Database connection parameters
$servername = "localhost";
$username = "root";
$password = "dinie123";
$database = "one_dormitory";

// Create connection
$con = mysqli_connect($servername, $username, $password, $database);

// Check connection
if (!$con) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

Figure 5.3 : Source code for connection to database

Figure 5.3 shows a PHP script designed to establish a connection to a MySQL database. It defines the connection parameters, including the server name (`localhost`), username (`root`), password (`dinie123`), and database name (`one_dormitory`). The script then attempts to connect using `mysqli_connect()` and stores the connection in the `$con` variable. It also includes a check to ensure the connection is successful; if it fails, an error message is displayed using `mysqli_connect_error()`, and the script stops executing.

5.4 Implementation Status

Table 5.2 : Implementation status

Module	Description	Duration
Login	Secure access for users with authentication	1 week
Student	Manages student hostel profiles and room placement	3 weeks
Staff	Handles staff details, payment and placement of the hostel	3 weeks
Room	Manage room bookings and availability	2 weeks
Hostel	Manage hostel room availability and occupancy	2 weeks
Payment	Facilitates hostel fee	2 weeks
Complaint	Track and manage student complaint	1 week

5.5 Summary

In conclusion, the success of implementing One Dormitory Management System relies heavily on a well-thought-out and robust database design. The database serves as the foundation for storing, retrieving, and managing critical information related to hostel rooms, bookings and payment transactions. Ensuring the database is designed efficiently is crucial for the smooth operation and overall effectiveness of the hostel management system.

In addition to robust design, regular maintenance and optimization of the database are equally vital to the system's long-term success. As the One Dormitory Management System scales, handling increased data volumes and user interactions will require continuous monitoring and fine-tuning. Ensuring data integrity, implementing security measures, and optimizing queries will help maintain high performance and reliability, ultimately leading to a more seamless user experience and better management of the hostel's operations.

CHAPTER 6: TESTING

6.1 Introduction

The testing phase is crucial for verifying system functionality and ensuring that the system's performance meets the specified requirements. This chapter will cover the test plan, test strategy, test design, and test results of the system. Testing involves running the database system with the goal of identifying errors. The testing process will start with smaller components and gradually expand to larger parts. These smaller components will be integrated with other modules, forming the foundation for testing the entire system.

6.2 Test Plan

6.2.1 Test Organization

The test organization outlines the roles and responsibilities of the personnel involved in the testing phases. In this section, the testing team will be established, with responsibilities including managing, executing, designing, reviewing and completing all testing tasks. In the context of the One Dormitory system, the system developer primarily acts as the main tester. The developer can identify bugs and errors firsthand, allowing for immediate resolution.

The system developer will conduct tests on all modules to ensure the system's integrity throughout the development process, aiming to minimize errors and reduce the likelihood of bugs in the final product.

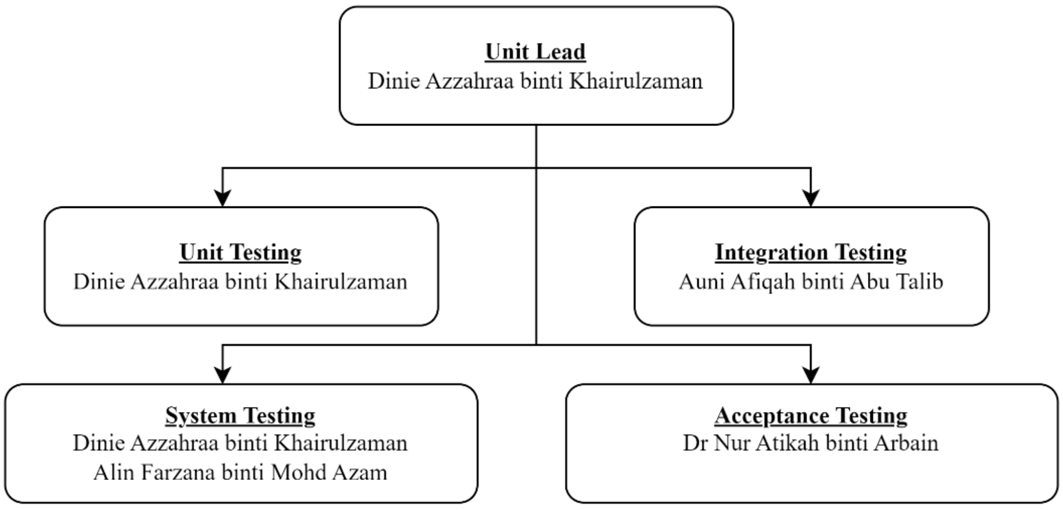

Unit Lead
Dinie Azzahraa binti Khairulzaman
Unit Testing
Dinie Azzahraa binti Khairulzaman
Integration Testing
Auni Afiqah binti Abu Talib
System Testing
Dinie Azzahraa binti Khairulzaman
Alin Farzana binti Mohd Azam
Acceptance Testing
Dr Nur Atikah binti Arbain

Figure 6.1 : Hierarchy of Test Organization

The provided image, Figure 6.1, illustrates a basic organizational chart outlining the structure of a testing team within a software development project. The Unit Lead, Dinie Azzahraa binti Khairulzaman, occupies a pivotal role, overseeing the entire team's activities while also actively participating in unit testing. This dual responsibility underscores the compact nature of the team or the specialized focus of their unit testing efforts.

Auni Afiqah binti Abu Talib takes charge of integration testing, a critical phase that ensures the smooth interaction and compatibility of various software components. Her role is instrumental in guaranteeing the seamless integration and operation of the entire system. System testing, a comprehensive evaluation of the software's overall functionality and performance, is entrusted to Dinie Azzahraa binti Khairulzaman and Alin Farzana binti Mohd Azam. Their collaborative efforts are essential in verifying the software's adherence to the intended design and its ability to deliver the desired outcomes.

The final stage of testing, acceptance testing, is conducted by Dr Nur Atikah binti Arbain. They meticulously assess the software against the predefined requirements, ensuring its readiness for deployment and delivery to end-users. Their sign-off signifies the successful completion of the testing process and paves the way for the software's release.

6.2.2 Test Environment

Table 6.1 : Test Environment Specification

System Configuration	**Specification (Server)**
Operating System	Windows 10 Home 64-bit
Database	PHPMyAdmin (MySQL)
Random Access Memory (RAM)	8 GB
Hard Disk	500 GB
Processor	Intel® Core™ i5-1135G7 processor Quad-core 2.40 GHz
Software	- Microsoft Visual Code - XAMPP

6.2.3 Test Schedule

The testing process is divided into three types: unit testing, integration testing, and user acceptance testing. In this section, each test output will be documented to ensure that the deliverables can be refined and improved after each testing phase.

Table 6.2 : Test Schedule

Module	Type	Duration
Registration	✓ Unit Test ✓ Integration Test ✓ User Acceptance Test	5 days
Login System	✓ Integration Test ✓ User Acceptance Test	3 days
Placement Module	✓ Integration Test ✓ User Acceptance Test	7 days
Payment Module	✓ Integration Test ✓ User Acceptance Test	8 days
Complaint Module	✓ Integration Test ✓ User Acceptance Test	4 days

6.3 Test Strategy

Several strategies can be employed for testing the One Dormitory System, including the bottom-up approach, top-down approach, white-box approach, and black-box approach. For this project, the white-box approach and black-box approach has been chosen for testing the One Dormitory System.

Table 6.3 : Description of approaches

Approaches	Explanation
White-Box	White-box testing is a method where the tester has detailed knowledge of the system's internal structure, design, and implementation. This approach is particularly suited for Unit and Integration Testing, as it requires programming expertise to assess and verify the accuracy of the system's internal processes

Black-Box	Black-box testing is a method where the tester lacks knowledge of the system's internal structure, design, or implementation. This approach is primarily applied in higher-level testing, such as Acceptance and System Testing, and evaluates the system's functionality based on inputs and expected outputs without needing programming expertise.

6.3.1 Classes of tests

The types of tests employed to evaluate the system's capabilities and ensure it meets the desired outcomes include functional testing, unit testing, and integration testing. These tests are crucial in assessing various aspects of the system's performance, ensuring that each component functions correctly and integrates seamlessly with other parts of the system, ultimately confirming that the system fulfills its intended requirements.

Table 6.4 : Description of classes of tests

Class of Test	Explanation
Functional Testing	The aim of running a functional test is to determine that the outputs developed by the system as the one required. Black-Box approach is a get ready to functional requirements of the system. During the test, the navigation process of the system is tested to determined that the connection go to the correct page
Unit Testing	This test helps to validate the individual units of source code is working properly or not. The aim is to fetch the tiny piece of testable software in the application and ensure whether it functioning similar as expected
Integration Testing	This test is a cross-check of the correctness interaction to determine the stable and stick the shifting between related modules

6.4 Test Design

6.4.1 Test Description

Tables 6.5 through 6.9 offer an in-depth description of the testing procedures applied to each module within the system. These tables meticulously detail the specific tests that have been conducted, including the methodologies used and the criteria for evaluation. Each table outlines the expected outcomes for the tests, providing a clear benchmark against which the system's performance is measured. Additionally, the tables illustrate how each test is related to the various components of the system, ensuring a comprehensive approach to system evaluation.

By examining each module in detail, these tables play a crucial role in guaranteeing that every aspect of the system is rigorously assessed. They help to ensure that all functionalities are tested in alignment with their intended design and requirements. This thorough examination is essential for identifying and addressing any potential issues or discrepancies that may arise during the testing phase. By documenting the testing processes and expected results in such a structured manner, these tables contribute significantly to the overall quality assurance efforts, ensuring that the system performs reliably and meets all necessary standards before it is fully deployed.

6.4.1.1 Registration Module

The Registration Module is a fundamental component of the One Dormitory System, serving as the initial step for users to gain access to the system. This module facilitates the process through which new users can create accounts by providing essential personal details. During registration, users are required to input their information, such as names, contact details, and other relevant data, which is necessary to generate a unique username. This username, along with a password created during the registration process, becomes the primary means of accessing the system.

Upon successful registration, users receive their new usernames and passwords, which they can then use to log in to the system. This ensures that only registered individuals can access the system's features and services. In the event that registration is unsuccessful due to errors or issues with the provided information, users will need to reattempt the registration process. This ensures that only accurate and complete details are captured, maintaining the integrity of user accounts and system security. Overall, the Registration Module is crucial for managing user access and ensuring a secure and organized process for onboarding new users into the One Dormitory System.

Table 6.5 : Description of Registration Module

Test Case ID	Description	Action	Expected Output

R01	Username = blank
	Password = blank
	Name = blank
	IC = blank
	Gender = -Select-
	Email = blank
	Role = -Select-
		No input provided	ERROR

R02	Username = Shasha
	Password = blank
	Name = blank
	IC = blank
	Gender = -Select-
	Email = blank
	Role = -Select-
		Password, Name, IC, Gender, Email and Role are left blank	ERROR

R03	Username = Shasha
	Password = ***
	Name = blank
	IC = blank
	Gender = -Select-
	Email = blank
	Role = -Select-
		Name, IC, Gender, Email and Role are left blank	ERROR

R04	Username = Shasha
	Password = ***
	Name = Shashabila
	IC = blank
	Gender = -Select-
	Email = blank
	Role = -Select-
		IC, Gender, Email and Role are left blank	ERROR

R05	Username = Shasha
	Password = ***
	Name = Shashabila
	IC = 010728101299
	Gender = -Select-
	Email = blank
		Gender, Email and Role are left blank	ERROR

	Role = -Select-		

R06	Username = Shasha
	Password = ***
	Name = Shashabila
	IC = 010728101299
	Gender = Female
	Email = blank
	Role = -Select-
		Email and Role are left blank	ERROR

R07	Username = Shasha
	Password = ***
	Name = Shashabila
	IC = 010728101299
	Gender = Female
	Email = shasha@gmail.com
	Role = -Select-
		Role is left blank	ERROR

R08	Username = Shasha
	Password = ***
	Name = Shashabila
	IC = 010728101299
	Gender = Female
	Email = shasha@gmail.com
	Role = Staff
		All necessary input is inserted	OK

6.4.1.2 Login System Module

The login system module is a critical component of the One Dormitory System, providing a secure entry point for users. To access the system, users must enter their usernames and passwords, which are essential for authenticating their identities. This process ensures that only authorized individuals can log in and interact with the system, thereby protecting sensitive information and maintaining overall system security.

Beyond basic authentication, the login system module may also include additional security features such as password recovery options, multi-factor authentication, and account lockout mechanisms. These features enhance security by safeguarding user accounts from unauthorized access and potential breaches. Overall, the login system module plays a vital role in managing access and ensuring that the system operates securely and efficiently.

Table 6.6 : Description of Login System Module

Test Case ID	Description	Action	Expected Output

L01	Username = blank
	Password = blank
		No input provided	ERROR

L02	Username = nana
	Password = blank
		Password is left blank	ERROR

L03	Username = blank
	Password = ***
		Username is left blank	ERROR

L04	Username = nana
	Password = ***
		All necessary input is inserted	OK

6.4.1.3 Placement Module

The Placement Module in the One Dormitory System is designed to allocate students to available hostel rooms based on predefined criteria such as room preferences and availability. It ensures efficient and fair distribution of accommodations, streamlining the process of room assignment for both students and administrators.

Table 6.7 : Description of Placement Module

Test Case ID	Description	Action	Expected Output

P01	Date = blank
	Duration = blank
		No input provided	ERROR

P02	Date = 1/9/2024
	Duration = blank
		Duration is left blank	ERROR

P03	Date = blank
	Duration = 4 months
		Date is left blank	ERROR

P04	Date = blank
	Duration = blank
		All necessary input is inserted	OK

6.4.1.4 Payment Module

The payment module in the One Dormitory System handles all financial transactions. It ensures secure and efficient management of fees, providing real-time updates on payment status and enabling automated reminders for pending dues.

Table 6.8 : Description of Payment Module

Test Case ID	Description	Action	Expected Output

P01	Description = -Select-
	Amount = auto
	Receipt = blank
		No input provided	ERROR

P02	Description = month 1
	Amount = auto
	Receipt = blank
		Receipt is left blank	ERROR

P03	Description = month 1
	Amount = auto
	Receipt = picture
		All necessary input is inserted	OK

6.4.1.5 Complaint Module

The Complaint Module within the One Dormitory System is designed to streamline the process through which students can submit and monitor complaints related to their accommodation. This module plays a crucial role in ensuring that any issues or concerns raised by students are efficiently logged and addressed. When a student encounters a problem with their accommodation, they can easily submit a detailed complaint through the system, specifying the nature of the issue and any relevant information.

Once a complaint is submitted, the module ensures that it is systematically recorded and assigned to the appropriate staff member for resolution. This assignment process helps to ensure that each issue is handled by personnel with the necessary expertise and responsibility.

Table 6.9 : Description of Complaint Module

Test Case ID	Description	Action	Expected Output

C01	Title = blank
	Description = blank
	Proof = blank
		No input provided	ERROR

C02	Title = Fan problem
	Description = blank
	Proof = blank
		Description and proof are left blank	ERROR

C03	Title = Fan problem
	Description = Fan too noisy
	Proof = blank
		Proof is left blank	ERROR

C04	Title = Fan problem
	Description = Fan too noisy
	Proof = 'picture'
		All necessary input is inserted	OK

6.4.2 Test Data

In this critical stage of the testing process, real data is employed to rigorously evaluate the system's accuracy and effectiveness. This phase is essential for ensuring that the system performs as expected under realistic conditions and with actual data inputs. By using real data, the testing process can more accurately simulate the system's performance in a real-world environment, providing a more reliable assessment of its functionality and effectiveness. Table 6.10 presents an example of the test data used during this phase

Table 6.10 : Description of Login test data

COMPONENT : LOGIN		
Test Number	**Attribute**	**Data**
TEST01	**Admin**	
	Username	admin
	Password	***
TEST02	**Staff**	
	Username	shasha
	Password	***
TEST03	**Student**	
	Username	nini
	Password	***

6.5 Test Result and Analysis

System : One Dormitory Management System

Version : 1.0

Module : Registration Module

Table 6.11 : Test Result and Analysis for Registration Module

Test Number	Action	Result	Pass Initials (OK / Fail)
TEST01	Valid input: Based on each input type Condition: User enters personal details	System will prompt with new username based on user priority	OK

System : One Dormitory Management System

Version : 1.0

Module : Login Module

Table 6.12 : Test Result and Analysis for Login Module

Test Number	Action	Result	Pass Initials (OK / Fail)
TEST01	Valid input: Condition: username and password already in database Input: Username: admin Password: ***	Able to access the system	OK
TEST02	Valid input: Condition: username and password not in database Input: Username: testing Password: testing	Display error message	OK

System : One Dormitory Management System

Version : 1.0

Module : Placement Module

Table 6.13 : Test Result and Analysis for Placement Module

Test Number	Action	Result	Pass Initials (OK / Fail)
TEST01	Valid input: Based on each input type Condition: User enters placement details	System will prompt the message and display back selected	OK

System : One Dormitory Management System

Version : 1.0

Module : Payment Module

Table 6.14 : Test Result and Analysis for Payment Module

Test Number	Action	Result	Pass Initials (OK / Fail)
TEST01	Valid input: Based on each input type Condition: User enters payment details	System will prompt the message and display back selected	OK

System : One Dormitory Management System

Version : 1.0

Module : Complaint Module

Table 6.15: Test Result and Analysis for Complaint Module

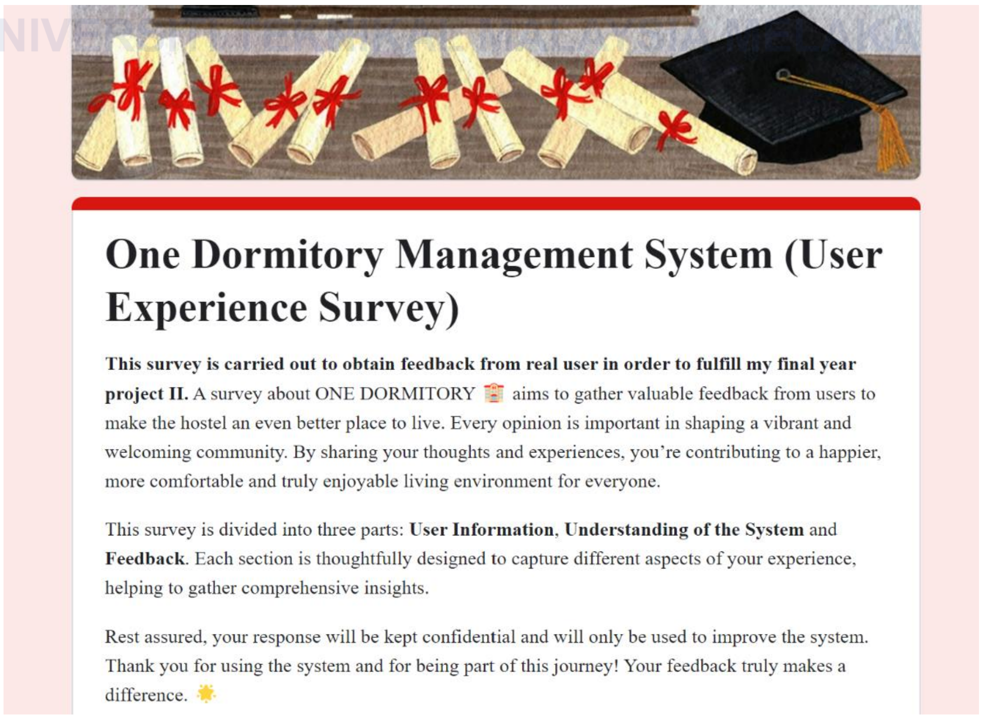
Test Number	Action	Result	Pass Initials (OK / Fail)
TEST01	Valid input: Based on each input type Condition: User enters complaint details	System will prompt the message and display back selected	OK

6.6 User acceptance testing

User Acceptance Testing (UAT) is a vital phase in the software development process, where the system is tested by end users to ensure it meets their needs and functions as expected in a real-world environment. During UAT, users such as students and staff interact with the system in a test environment that mirrors the production setup, performing tasks like booking rooms or managing student information to validate the system's usability and functionality. To gather feedback efficiently, a survey was created by using Google form, allowing users to provide detailed input on their experience with the system, helping to identify any remaining issues or areas for improvement before final deployment. This feedback was crucial in making necessary adjustments, ensuring the system is ready for launch.

a) Survey Questionnaires

Figure 6.2 – 6.7 depicts a survey titled "One Dormitory Management System (User Experience Survey)." This survey aims to gather feedback from real users to improve the hostel's living environment. The survey is divided into three parts: User Information, Understanding of the System, and Feedback. It emphasizes the importance of user opinions in shaping a vibrant and welcoming community and assures confidentiality of responses. The overall goal is to enhance the user experience and create a more enjoyable living environment for everyone.



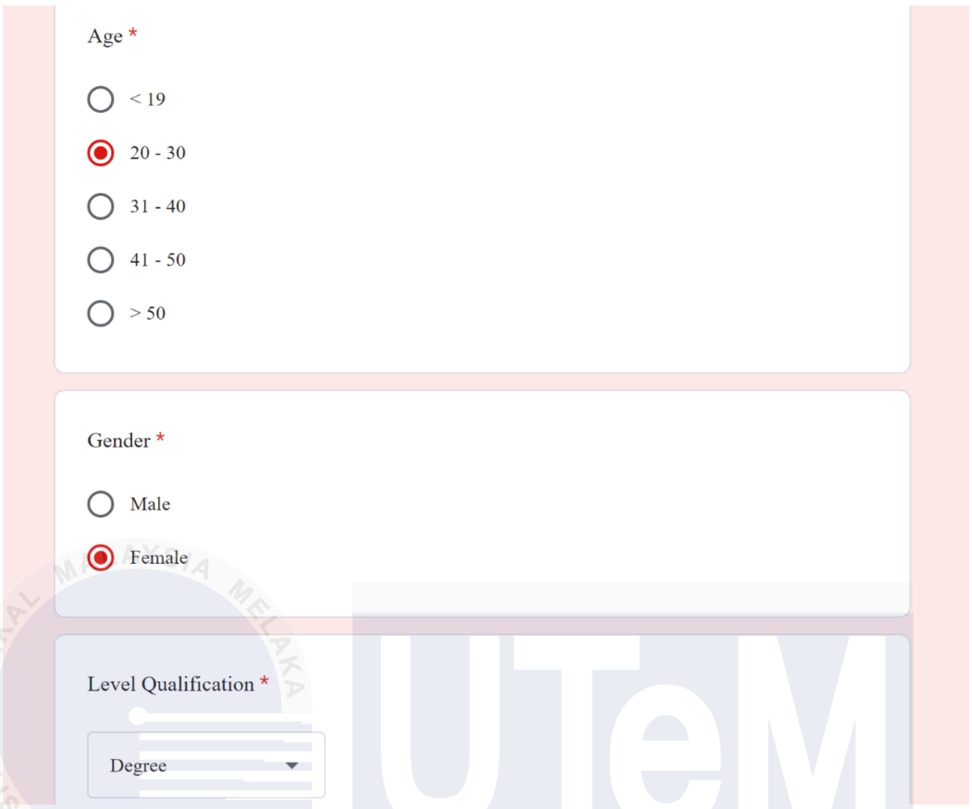
One Dormitory Management System (User Experience Survey)

This survey is carried out to obtain feedback from real user in order to fulfill my final year project II. A survey about ONE DORMITORY 🏨 aims to gather valuable feedback from users to make the hostel an even better place to live. Every opinion is important in shaping a vibrant and welcoming community. By sharing your thoughts and experiences, you're contributing to a happier, more comfortable and truly enjoyable living environment for everyone.

This survey is divided into three parts: **User Information**, **Understanding of the System** and **Feedback**. Each section is thoughtfully designed to capture different aspects of your experience, helping to gather comprehensive insights.

Rest assured, your response will be kept confidential and will only be used to improve the system. Thank you for using the system and for being part of this journey! Your feedback truly makes a difference. ☀️

Figure 6.2 : Survey Questionnaires Initiation for User Feedback During UAT



Age *

- ○ < 19
- [✓ 20 - 30]
- ○ 31 - 40
- ○ 41 - 50
- ○ > 50

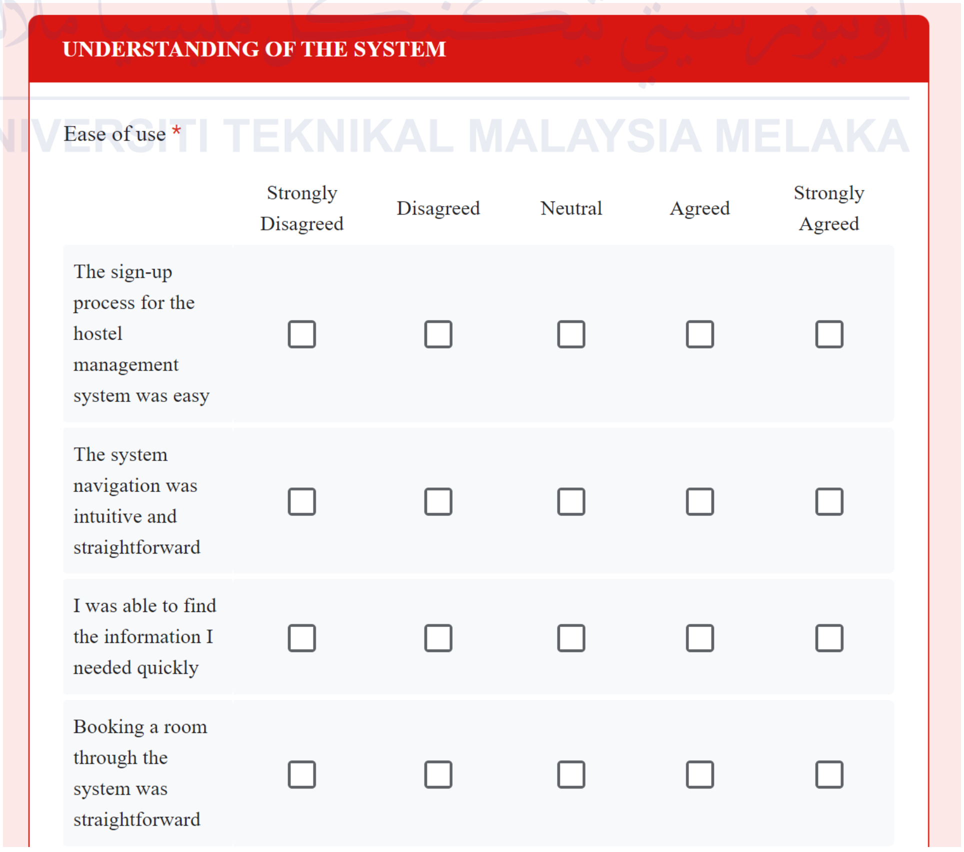
Gender *

- ○ Male
- [✓ Female]

Level Qualification *

Degree

Figure 6.3 : The list of question for user information

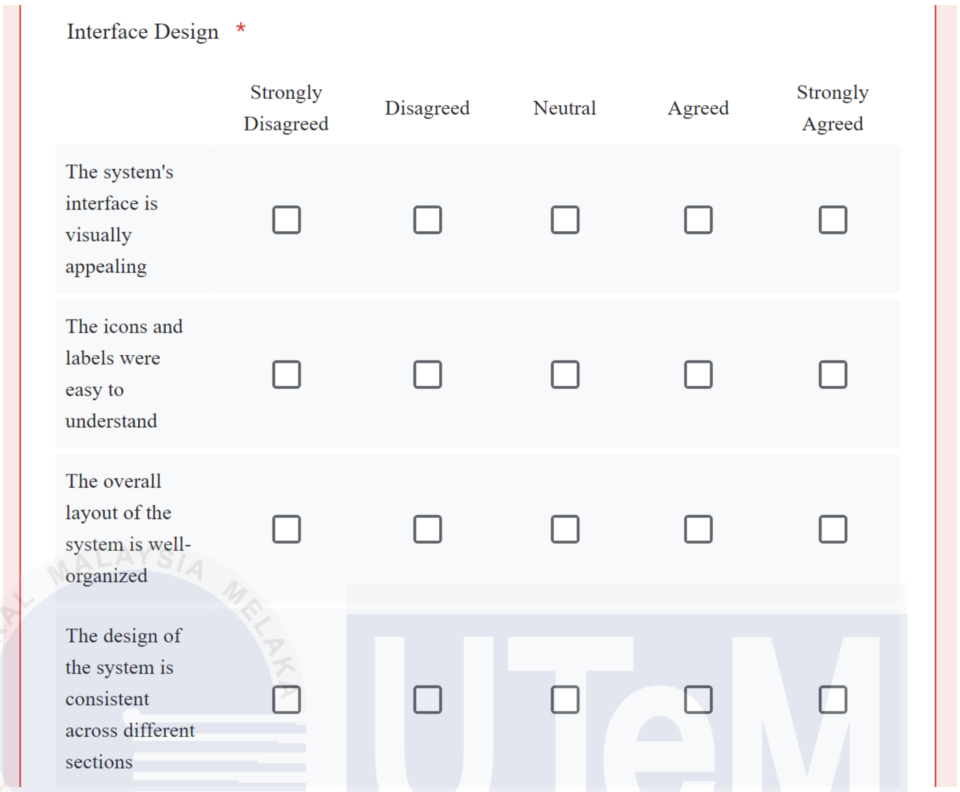


UNDERSTANDING OF THE SYSTEM

Ease of use *

	Strongly Disagreed	Disagreed	Neutral	Agreed	Strongly Agreed
The sign-up process for the hostel management system was easy	☐	☐	☐	☐	☐
The system navigation was intuitive and straightforward	☐	☐	☐	☐	☐
I was able to find the information I needed quickly	☐	☐	☐	☐	☐
Booking a room through the system was straightforward	☐	☐	☐	☐	☐

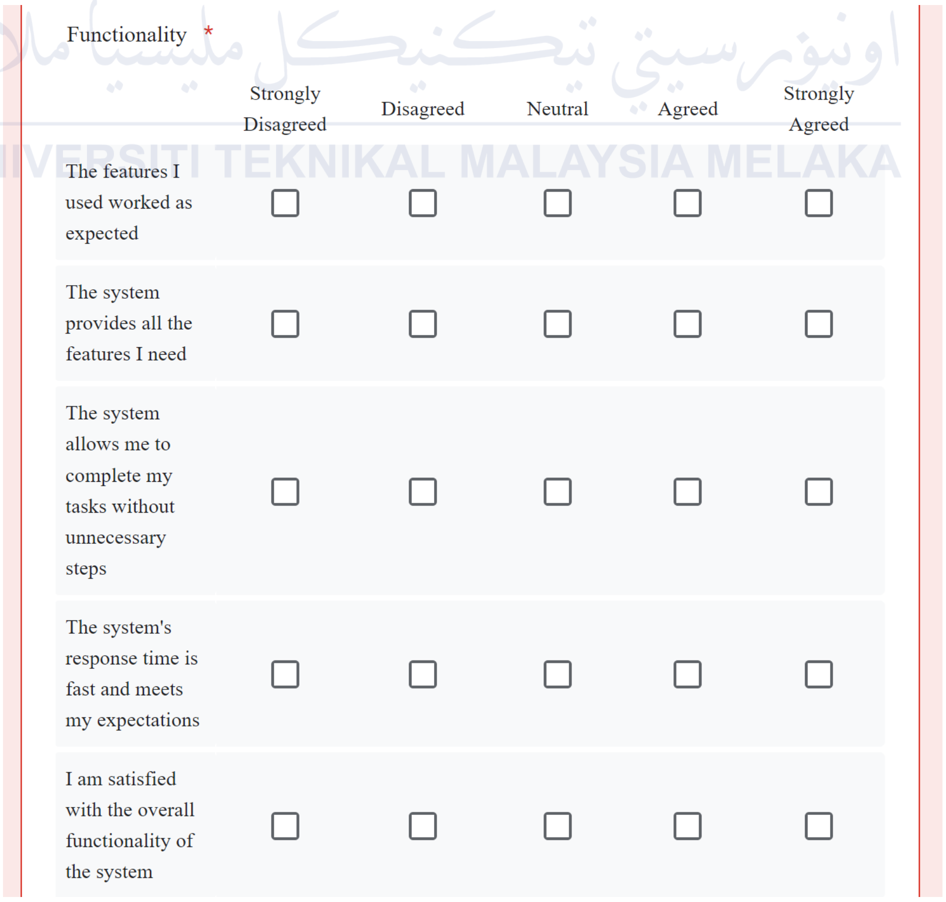
Figure 6.4 : The list of question for ease of use



Interface Design *

	Strongly Disagreed	Disagreed	Neutral	Agreed	Strongly Agreed
The system's interface is visually appealing	☐	☐	☐	☐	☐
The icons and labels were easy to understand	☐	☐	☐	☐	☐
The overall layout of the system is well-organized	☐	☐	☐	☐	☐
The design of the system is consistent across different sections	☐	☐	☐	☐	☐

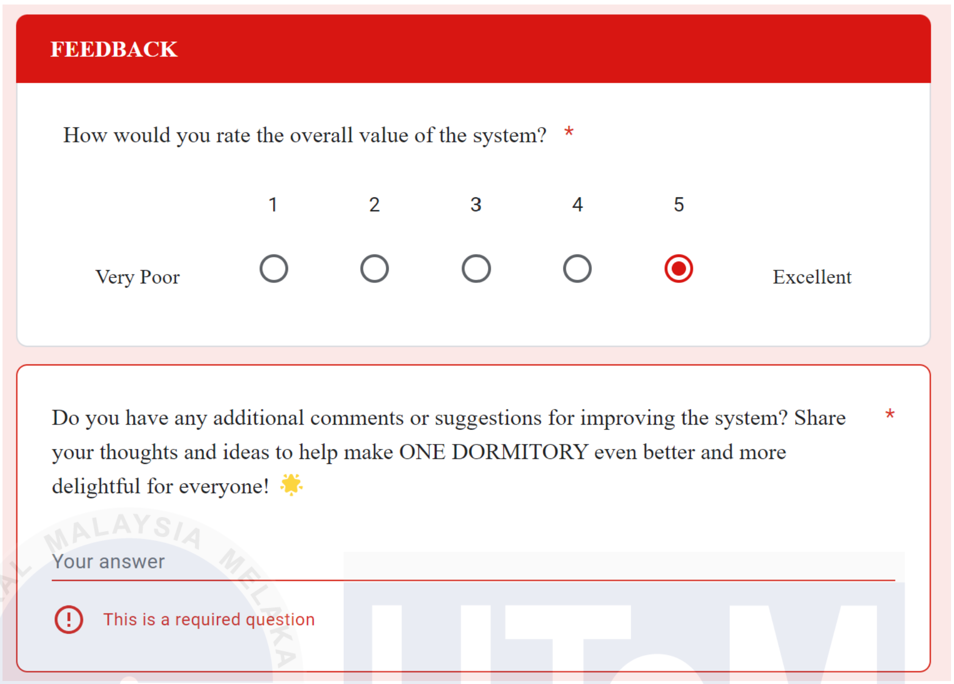
Figure 6.5 : The list of question for interface design



Functionality *

	Strongly Disagreed	Disagreed	Neutral	Agreed	Strongly Agreed
The features I used worked as expected	☐	☐	☐	☐	☐
The system provides all the features I need	☐	☐	☐	☐	☐
The system allows me to complete my tasks without unnecessary steps	☐	☐	☐	☐	☐
The system's response time is fast and meets my expectations	☐	☐	☐	☐	☐
I am satisfied with the overall functionality of the system	☐	☐	☐	☐	☐

Figure 6.6 : The list of question for functionality of the system



FEEDBACK

How would you rate the overall value of the system? *

Very Poor 1 2 3 4 [✓ 5] Excellent

Do you have any additional comments or suggestions for improving the system? Share your thoughts and ideas to help make ONE DORMITORY even better and more delightful for everyone! *

Your answer

This is a required question

Figure 6.7 : Question for user feedback

b) Results for survey

The survey results, depicted in Figures 6.8-6.15, provide valuable insights into the user experience of our dormitory management system. These findings will inform future improvements and enhancements to the system, ensuring it continues to meet the needs and expectations of our users.

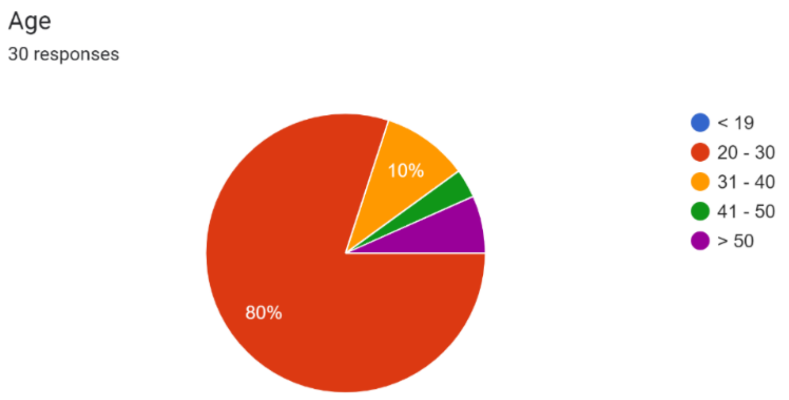


Figure 6.8 : Pie Chart representing age distribution of users

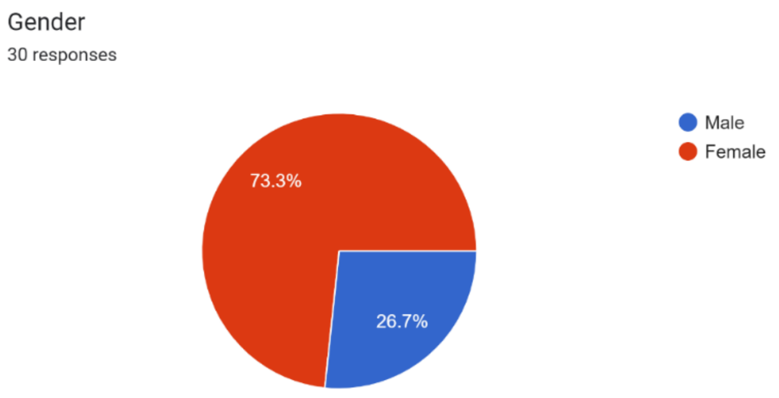


Figure 6.9 : Pie Chart representing gender of users

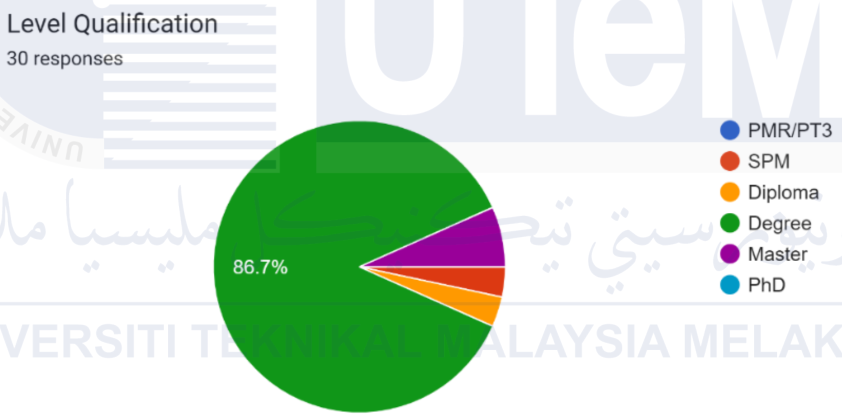


Figure 6.10 : Pie Chart representing level qualification of users

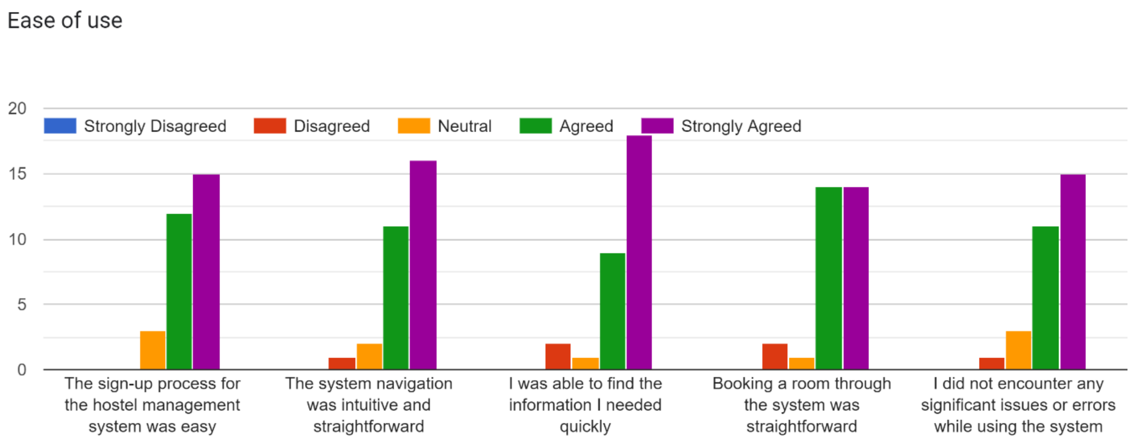


Figure 6.11 : Graft bar representing questions for ease of use

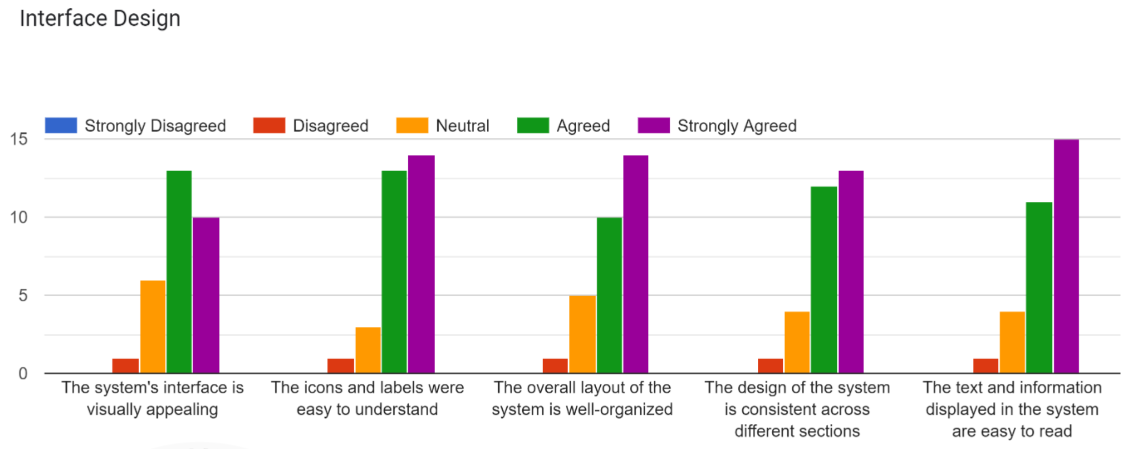


Figure 6.12 : Graft bar representing questions for interface design

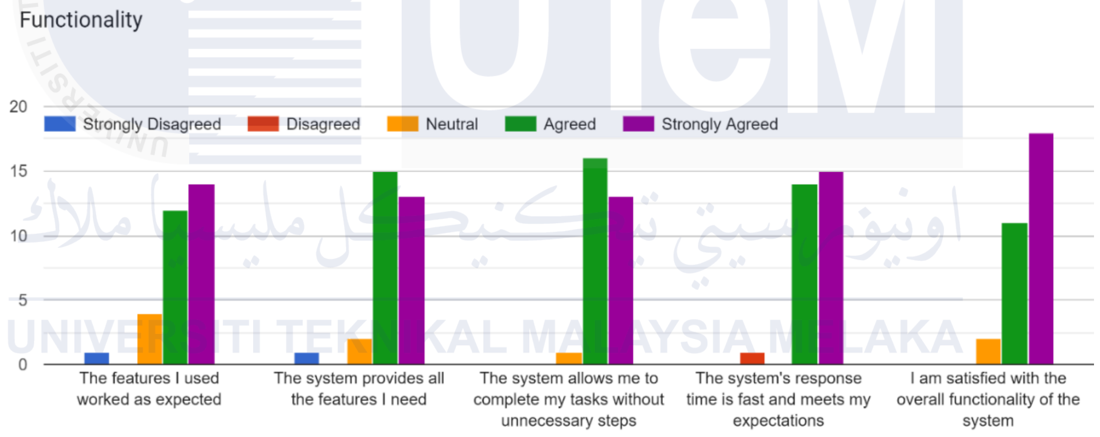


Figure 6.13 : Graft bar representing questions for functionality of the system

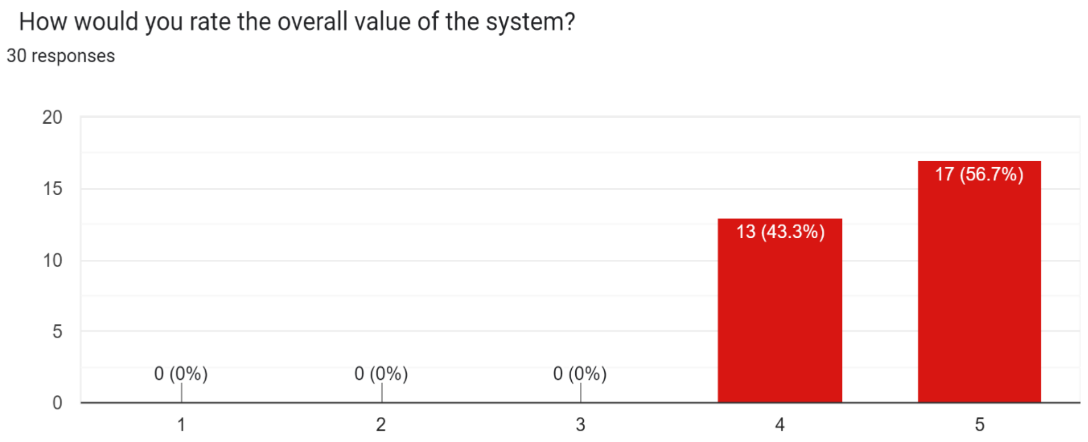
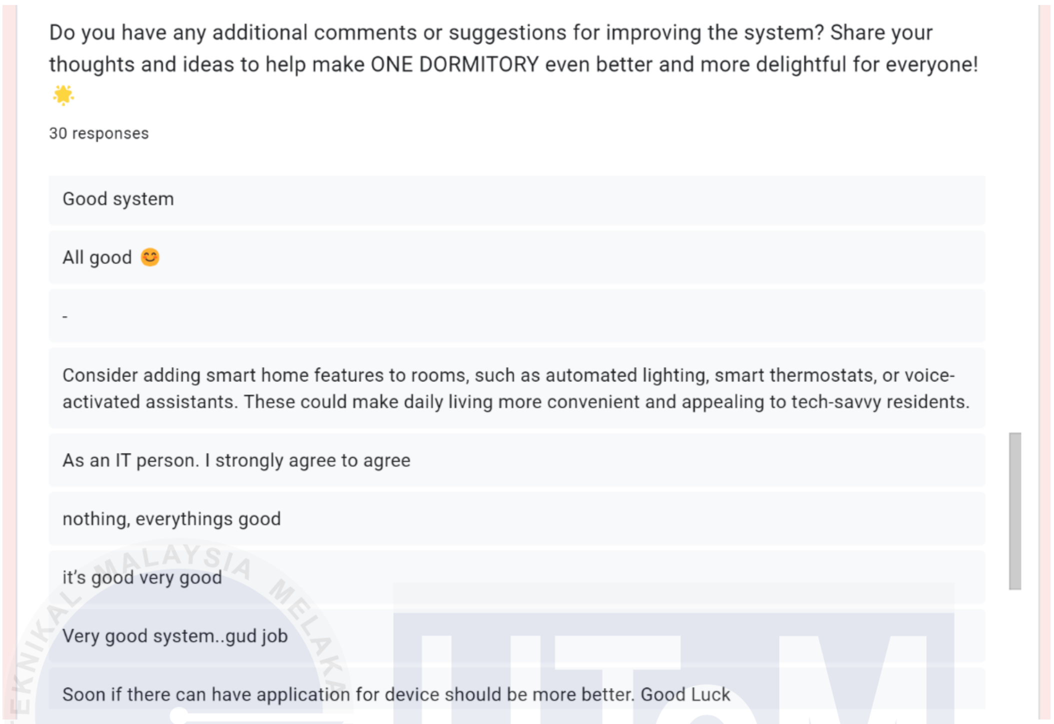


Figure 6.14 : Graft bar representing questions for feedback section


Do you have any additional comments or suggestions for improving the system? Share your thoughts and ideas to help make ONE DORMITORY even better and more delightful for everyone! 🌞

30 responses

Good system

All good 😊

-

Consider adding smart home features to rooms, such as automated lighting, smart thermostats, or voice-activated assistants. These could make daily living more convenient and appealing to tech-savvy residents.

As an IT person. I strongly agree to agree

nothing, everythings good

it's good very good

Very good system..gud job

Soon if there can have application for device should be more better. Good Luck

Figure 6.15 : Feedback from users

The survey results are highly favorable, reflecting that the system meets user expectations across several key areas. Respondents, predominantly aged 20 to 30, with a higher proportion of females and degree holders, generally agreed that the system is user-friendly, well-designed and functional. This feedback indicates that the system's ease of use, interface design and overall functionality are well-received, suggesting that it is effective and ready for broader implementation.

6.7 Summary

This chapter provides an in-depth examination of the comprehensive testing processes that have been meticulously carried out to ensure the system is released free of bugs. These rigorous testing procedures are designed not only to uncover and address any potential issues but also to systematically improve the system's overall performance and reliability. By identifying and resolving a range of bugs, these tests play a crucial role in enhancing the system's quality and ensuring a smooth user experience upon release.

CHAPTER 7: CONCLUSION

7.1 Introduction

The hostel management system is designed to assist users in efficiently managing room bookings, student information, and financial records. The primary goal of the system is to streamline and organize data within the database to meet user requirements effectively. By automating these processes, the system aims to reduce manual errors, save time, and improve overall operational efficiency. This chapter provides a comprehensive overview of the system, highlighting its strengths and weaknesses, and offers insights into how it can enhance the management of hostel operations.

7.2 Strengths and Weaknesses

Table 7.1 : Strengths and weaknesses for One Dormitory

Strengths	Weaknesses
The system can help hostel staff keep track of important information, such as student details and payment transactions	Some students may be concerned about the privacy of their personal information stored in the one dormitory management system
The system can facilitate communication between hostel staff and student, allowing for quicker responses to inquiries and maintenance requests	Technical problems, such as system crashes or data loss, can disrupt hostel operations and cause frustration for both staff and students
The system can automate many tasks, such as managing room availability, processing bookings and tracking payments.	If the system experiences downtime, hostel staff and students may encounter difficulties with tasks such as managing room bookings and processing payments

7.3 Proposition for improvement

There is always room for improvement to make the system more appealing and user-friendly for the target user. This can be done by adding new features or adopting a more attractive and intuitive design. By focusing on both functional and aesthetic upgrades, the system can better engage users and provide a more enjoyable and efficient experience.

To improve the hostel management system, several key upgrades can be made. First, strengthen security by using advanced encryption and access controls to protect student data. Regular updates and vulnerability checks will help keep the system secure. Additionally, set up a reliable backup and recovery system with automatic backups and regular tests to ensure data can be quickly restored if something goes wrong. Also, enhance user support with thorough training for staff and a responsive helpdesk to quickly address any issues.

7.4 Project Contribution

The project will significantly contribute to the hostel management system by streamlining the management of One Dormitory-related tasks. This system is designed to accommodate various users, each with distinct levels of access and authority to handle and retrieve data. For instance, staff possess special authority, granting them access to all data within the hostel management system. This comprehensive access allows administrators to efficiently oversee and manage all aspects of hostel operations, from room bookings and student information to complaint request and payment transactions. By providing tailored access levels, the system ensures that each user can perform their specific duties effectively while maintaining the security and integrity of the data.

7.5 Summary

The One Dormitory Management System is designed for various users, including staff and students. It is built to be flexible in handling room bookings, student information, complaint requests and payment transactions. Throughout the project, One Dormitory has achieved its set objectives, which are to store student dormitory records in a centralized database, minimizing data duplication and reducing the risk of data loss, to improve the efficiency of record-keeping processes in student hostels by transitioning to a computerized recording system and to develop a system that automates the generation of reports, reducing the labor and errors associated with manual report creation. The project output has successfully assisted in the comprehensive management of hostel operations, ensuring all aspects receive the necessary attention. To meet future requirements, the system will need to be continually revised and improved.

REFERENCES

Wan Ja'afar, W. N. H. (2012). Hostel Management System (HMS). Universiti Malaysia Pahang.

Tan, S. W., & The, S. S. (2014). E-Hostel Service System. Universiti Malaysia Sabah, Faculty of Computing and Informatics.

Rogue Plus Publishing. (2018). Hotel Reservation Log Book. CreateSpace Independent Publishing Platform.

Gudadhe, M., Bhoyar, A., Karwade, A., & Dhaware, V. (2022). Hostel Management System. Department of Information Technology, Priyadarshini College of Engineering, Nagpur, India.

SharmikhaSree, R., Meera, S., Kavya, K. K., & Harina, P. (2023). Dormitory Management System. IEEE.

Clark, H. (2017). User-Centered Design in Hostel Management Systems: Improving Student Satisfaction. International Journal of Human-Computer Interaction.

Yang, Y., & Chen, S. (2022). Design and Implementation of College Dormitory Management System. IEEE.

Kamal Acharya, (January, 2022). Hostel Management System Project. Tribhuvan University

Wong, P. L., & Mahdin, H. (2022). Design and Development of Online Hostel Management System to Improve Application and Management Process.

A, Manoj & Shah, Yogesh & Luitel, Chandra & Jaiswal, Ankit & S., Sajat. (2023). Design and implementation of an automated system for hostel management using cloud technology and app development.

Leonard Wandera. (2016). A Project Report On Hostel Management System. Zetech University

Brown, L., & Green, K. (May, 2022). System Analysis and Design for Student Hostel Management System Project. Journal of Communication and Information Technology

Williams, R., & Taylor, S. (2019). Room Allocation Optimization in Hostel Management Systems. Journal of Systems and Software

Philip Badaszewski. (2022). The Journal of College and University Student Housing. Association of College & University Housing Officers

Israt Jahan, Samsunnahar Chow, Md Towfiqul. (June, 2020). Hostel Management System. Department of Electronics and Communication Engineering

Universiti Teknikal Malaysia Melaka
Hang Tuah Jaya,
76100 Durian Tunggal,
Melaka, Malaysia.

+606 270 1000
+606 270 1022
www.utem.edu.my

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI
Tel : +606 270 2411 | Faks : +606 270 1048

Rujukan Kami (Our Ref) : UTEM.600-5/1/5 Jld.3 (87)
Rujukan Tuan (Your Ref):
Tarikh (Date): 8 Ogos 2024 | 3 Safar 1446H

KEPADA SESIAPA YANG BERKENAAN

Tuan,

PERMOHONAN MENJALANKAN KAJIAN KES BAGI TUGASAN PROJEK

Dengan hormatnya saya merujuk perkara di atas.

2. Dimaklumkan bahawa penama tersebut adalah pelajar Universiti Teknikal Malaysia Melaka. Maklumat terperinci adalah seperti berikut :

Nama	:	Dinie Azzahra binti Khairulzaman
No. Matrik	:	
Kursus	:	Ijazah Sarjana Muda Sains Komputer (Pengurusan Pangkalan Data) dengan Kepujian
Fakulti	:	Fakulti Teknologi Maklumat & Komunikasi

3. Pelajar ini perlu menyiapkan satu tugasan projek bagi kursus BITU3983 (Projek Sarjana Muda II). Sehubungan dengan ini, saya sangat berbesar hati sekiranya pihak tuan dapat memberi peluang kepada pelajar ini untuk membuat kajian kes tersebut di organisasi tuan.

Segala kerjasama daripada pihak tuan didahului dengan ucapan terima kasih.

Sekian.

"BIJAK LAKSANA TUAH, BERANI LAKSANA JEBAT"
"MALAYSIA MADANI"
"BERKHIDMAT UNTUK NEGARA"
"KOMPETENSI TERAS KEGEMILANGAN"

Saya yang menjalankan amanah,

[illegible]

SHARIFAH NURUL FARIDAH BINTI SYED ABU BAKAR
Timbalan Pendaftar Kanan
Fakulti Teknologi Maklumat dan Komunikasi
Universiti Teknikal Malaysia Melaka

nas/ua