APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 6 No. 2 (2025) 130-149
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Online Accommodation Management System for UTHM Students

Jocelyn Song Zhi Wei[1], Noryusliza Abdullah[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: yusliza@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.008

Article Info

Received: 11 June 2025
Accepted: 3 November 2025
Available online: 30 November 2025

Keywords

Online Accommodation Management System, Booking, Iterative Waterfall Model.

Abstract

The Online Accommodation Management System for UTHM students is built to address inefficiencies in the manual registration process. The new system allows students to register, view roommates, manage profiles, and handle online bookings. By eliminating queues and introducing key collection through time slots, the system reduces administrative workload, prevents document loss, and saves space. Developed with PHP and MySQL using the iterative waterfall model, it streamlines operations, ensuring data accuracy and a better student and staff experience. This system aims to improve the overall efficiency of the hostel management process.

1. Introduction

Students stay in the hostel to pursue formal education away from their homes. A hostel is a budget-friendly accommodation focusing on a shared social experience [1]. A hostel is a large house generally owned by local government authorities or charities where people can stay cheaply for a short period [2].

The Online Accommodation Management System is a system that handles hostel activities such as applying for a hostel room, online registration, room allocation, and generating timeslots for students to take the room key. The Online Accommodation Management System aims to provide a platform for Universiti Tun Hussein Onn (UTHM) hostel students to apply for hostel rooms. Moreover, this system will enable students to book for the Wisdom Inn without queueing or engaging with hostel staff. Students can book the room at any time. It offers a more efficient and convenient way to manage the check-in and out of the room process and enhance satisfaction while ensuring secure and real-time data updates for the hostel administration.

The current system used by the Student Housing Management Department is the Housing Office Management System **(**HOMS). HOMS is a platform for hostel students to apply for hostels and check for approval. However, it does not provide a platform for Wisdom Inn students to self-check in and out of the hostel. The staff manually completes the current registration process on the hostel's registration day. This causes problems, such as students queueing up and waiting for a long time to take the room keys and check into the hostel. Also, the staff needs to collect a pile of student registration forms, which causes lots of administrative burdens. In the current system, students are not allowed to view their roommates. Hence, they cannot discuss managing their room together, which causes them to worry about their housing situation.

The proposed solution is an Online Accommodation Management System designed to streamline the Wisdom Inn check-in and out process for students, eliminating the need to queue physically in the staff room. This system will enable students to register for hostel check-in remotely, from any location and anytime, using their devices. Automating the process reduces the administrative workload, minimizes delays, and ensures a

This is an open access article under the CC BY-NC-SA 4.0 license.

more efficient and user-friendly experience for students and staff. It enhances convenience, promotes self-service, and optimizes resource management within the hostel registration process. Also, the other hostel students can use the system to apply for their next semester's accommodation on campus.

2. Related Work

2.1 UTHM Student Housing Management Department

The UTHM Student Housing Management Department primarily serves students by offering accommodations within the university's various hostels, collectively as UTHM hostels or residential colleges. These hostels are allocated based on availability and the eligibility of students who apply for housing each semester. The department coordinates with other university bodies to maintain high standards in facilities, ensuring that accommodations are secure, clean, and well-maintained.

The UTHM Student Housing Management Department operates a structured set of processes to manage each phase of the student housing experience, from the initial application to check-out. Students submit their applications for hostel accommodations at the start of each semester through the Housing Office Management System (HOMS). The HOMS system provides several functions, such as making room reservations, cancelling rooms, registering new students, managing room types and services, and logging into the homepage [3]. This system verifies each student's eligibility and processes room allocations based on the availability of spaces, prioritizing applications as needed. Once a room is allocated, students can track the status of their application and approval details via HOMS. When allocation is confirmed, students proceed to the check-in phase, which occurs on-site at the hostel. A letter will be sent to the student's email, and they must bring the letter to submit on the hostel registration day. On registration day, students receive their room keys and other essential information while staff manually verify student details and collect the letters. This check-in process can often lead to long waiting lines and administrative difficulties.

Despite the comprehensive structure, the department's current processes face several challenges. Notably, the lack of a self-registration feature for returning students results in delays and congestion, as check-ins are handled manually on registration day. The reliance on manual processes further increases the workload for staff and creates the potential for errors, such as room misallocations. Additionally, students cannot access information about their housemates after room assignments are made, leading to anxiety about their living arrangements. Addressing these limitations could significantly enhance the department's efficiency, reduce administrative burdens, and improve the housing experience for students.

2.2 Study on Existing Related System

The existing systems involved in this study are the Housing Office Management System (HOMS), University Hostel Management System (UHMS), and Hostel Management System. The Housing Office Management System (HOMS) is a modern student accommodation platform developed by the Student Housing Centre (PRP), Information Technology Centre (PTM), and an external provider to enhance UTHM Residential Colleges' digital systems. Accessible via multiple platforms and optimized for mobile use, HOMS initially supports online applications for senior students' hostel placements and pre-registration for new students, with plans for broader functionality. Key features include hostel applications, booking status tracking, pre-registration for room selection, housemate details viewing, and a demerit system for checking penalties. HOMS also allows students to report hostel issues with detailed descriptions and photos. It offers an Off-Campus check-in function for external renters, including a rental house search by zone and location. Login security is ensured through matric number and password authentication, safeguarding student access [3[.

The University Hostel Management System (UHMS) is a comprehensive platform designed to streamline hostel operations in universities and colleges [4]. It automates room allocation, meal and lodging records, fee tracking, and feedback collection while managing attendance, disciplinary logs, and check-in/out processes. The system reduces staff workload and paper usage by maintaining detailed records of student accommodations, including room preferences, transfers, food usage, and payments, while gathering feedback on facilities and sending fee reminders. It also tracks building details, room availability, and visitor records, supports mess bill and maintenance charge management, and monitors daily expenses. UHMS ensures efficient administration, financial management, and a well-organized residential environment that meets the needs of students and staff.

The hostel management system is a web-based platform designed to streamline and automate hostel operations, addressing the challenges of traditional manual methods in managing the growing complexity of student accommodations [5]. It reduces administrative workload, minimizes errors, and improves efficiency by integrating processes such as student registration, administrator login, and data management. Key modules include an administrator login for managing room assignments, evictions, fee verification, and record updates; a student registration module for collecting and maintaining vital student details; and a dashboard for centralized access to student records and hostel information. The system enhances transparency, organization, and user

convenience, meeting the needs of modern academic institutions. Table 1 shows the comparison between the existing systems and the developed system.

Table 1 *System Comparison*

Feature	**Housing Office Management System (HOMS)**	**University Hostel Management System (UHMS)**	**Hostel Management System**	**Online Accommodation Management System**
Dashboard	√	√	√	√
Hostel Application Module	√	√	X	√
Registration Module	√	√	√	√
Login Module	√	√	√	√
Student Profile Module	√	√	√	√
Check-in/out Module	X	√	X	√
Demerit Module	√	√	X	X
Off-Campus Checkin	√	X	X	X
E-Complaint Module	√	√	X	X
Key Collection Time Module	√	X	X	√
Announcement Module	√	√	X	√

3. Methodology

The Software Development Life Cycle (SDLC) is a systematic process by software development organizations to plan, design, build, test, deploy, and maintain software applications [6]. This project uses an iterative waterfall methodology. The iterative waterfall approach is selected in software development projects because the project has relatively precise and stable requirements but might gain feedback and refinement at different stages. Iterative waterfall methodology combines the waterfall model and iterative development elements to increase software development efficiency [7]. Unlike the traditional waterfall, the iterative waterfall allows the author to go back to previous phases, update the criteria, and make changes based on client feedback. The stages of iterative waterfall include planning, analysis, design, implementation, testing, and maintenance, as shown in Figure 1, and are further discussed in this section.

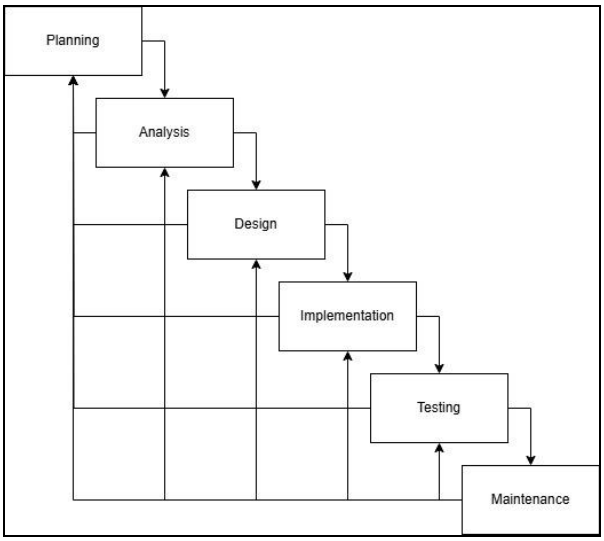
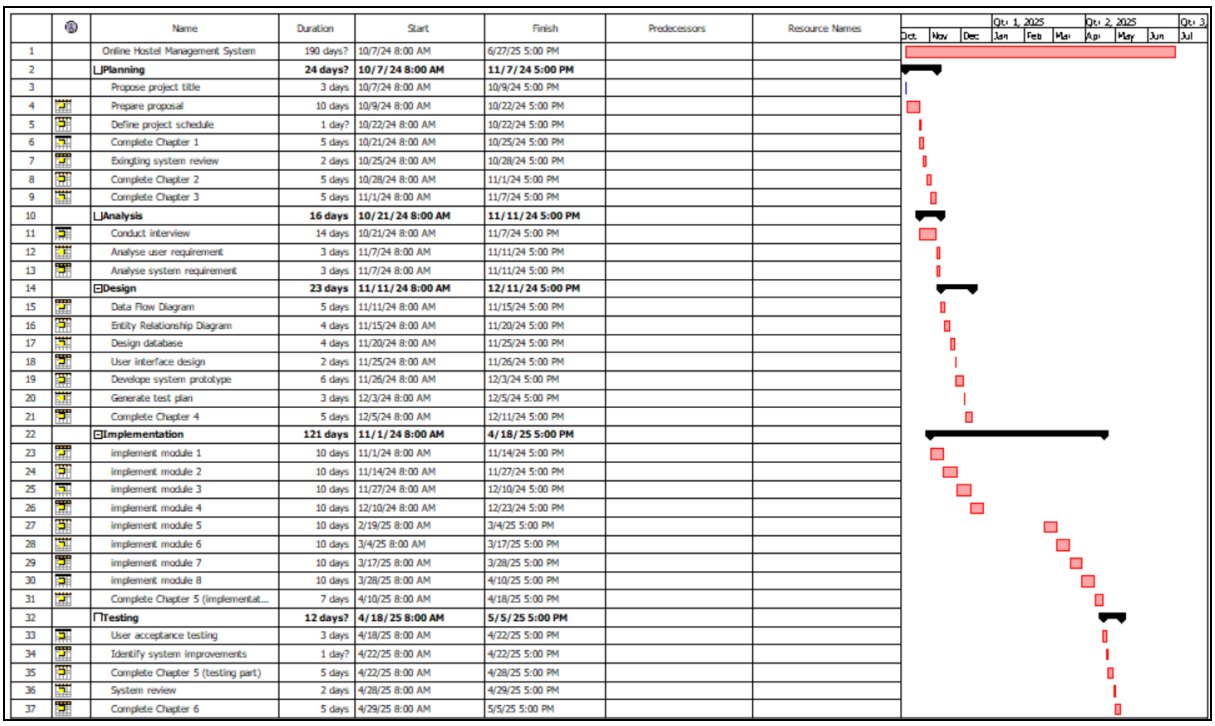


Fig.1 *Iterative Waterfall Model*

The planning phase is a critical stage in a project lifecycle where the objectives, scope, and requirements are clearly defined and documented to establish a solid foundation for its execution. This phase involves identifying the project's goals, determining deliverables, specifying the target users, and creating a detailed plan that

outlines the tasks, timelines, resources, and processes needed to complete the project successfully. It ensures all stakeholders have a shared understanding of the project's purpose and approach while minimizing risks, improving efficiency, and setting the direction for subsequent phases. A Gantt chart is set up to list all the activities and durations that will be carried out throughout the project development, as discussed in Figure 2.



	Name	Duration	Start	Finish	Predecessors	Resource Names
1	Online Hostel Management System	190 days?	10/7/24 8:00 AM	6/27/25 5:00 PM		
2	**Planning**	**24 days?**	**10/7/24 8:00 AM**	**11/7/24 5:00 PM**		
3	Propose project title	3 days	10/7/24 8:00 AM	10/9/24 5:00 PM		
4	Prepare proposal	10 days	10/9/24 8:00 AM	10/22/24 5:00 PM		
5	Define project schedule	1 day?	10/22/24 8:00 AM	10/22/24 5:00 PM		
6	Complete Chapter 1	5 days	10/21/24 8:00 AM	10/25/24 5:00 PM		
7	Exingting system review	2 days	10/25/24 8:00 AM	10/28/24 5:00 PM		
8	Complete Chapter 2	5 days	10/28/24 8:00 AM	11/1/24 5:00 PM		
9	Complete Chapter 3	5 days	11/1/24 8:00 AM	11/7/24 5:00 PM		
10	**Analysis**	**16 days**	**10/21/24 8:00 AM**	**11/11/24 5:00 PM**		
11	Conduct interview	14 days	10/21/24 8:00 AM	11/7/24 5:00 PM		
12	Analyse user requirement	3 days	11/7/24 8:00 AM	11/11/24 5:00 PM		
13	Analyse system requirement	3 days	11/7/24 8:00 AM	11/11/24 5:00 PM		
14	**Design**	**23 days**	**11/11/24 8:00 AM**	**12/11/24 5:00 PM**		
15	Data Flow Diagram	5 days	11/11/24 8:00 AM	11/15/24 5:00 PM		
16	Entity Relationship Diagram	4 days	11/15/24 8:00 AM	11/20/24 5:00 PM		
17	Design database	4 days	11/20/24 8:00 AM	11/25/24 5:00 PM		
18	User interface design	2 days	11/25/24 8:00 AM	11/26/24 5:00 PM		
19	Develope system prototype	6 days	11/26/24 8:00 AM	12/3/24 5:00 PM		
20	Generate test plan	3 days	12/3/24 8:00 AM	12/5/24 5:00 PM		
21	Complete Chapter 4	5 days	12/5/24 8:00 AM	12/11/24 5:00 PM		
22	**Implementation**	**121 days**	**11/1/24 8:00 AM**	**4/18/25 5:00 PM**		
23	implement module 1	10 days	11/1/24 8:00 AM	11/14/24 5:00 PM		
24	implement module 2	10 days	11/14/24 8:00 AM	11/27/24 5:00 PM		
25	implement module 3	10 days	11/27/24 8:00 AM	12/10/24 5:00 PM		
26	implement module 4	10 days	12/10/24 8:00 AM	12/23/24 5:00 PM		
27	implement module 5	10 days	2/19/25 8:00 AM	3/4/25 5:00 PM		
28	implement module 6	10 days	3/4/25 8:00 AM	3/17/25 5:00 PM		
29	implement module 7	10 days	3/17/25 8:00 AM	3/28/25 5:00 PM		
30	implement module 8	10 days	3/28/25 8:00 AM	4/10/25 5:00 PM		
31	Complete Chapter 5 (implementat...	7 days	4/10/25 8:00 AM	4/18/25 5:00 PM		
32	**Testing**	**12 days?**	**4/18/25 8:00 AM**	**5/5/25 5:00 PM**		
33	User acceptance testing	3 days	4/18/25 8:00 AM	4/22/25 5:00 PM		
34	Identify system improvements	1 day?	4/22/25 8:00 AM	4/22/25 5:00 PM		
35	Complete Chapter 5 (testing part)	5 days	4/22/25 8:00 AM	4/28/25 5:00 PM		
36	System review	2 days	4/28/25 8:00 AM	4/29/25 5:00 PM		
37	Complete Chapter 6	5 days	4/29/25 8:00 AM	5/5/25 5:00 PM		

Fig.2 *Gantt chart plan*

The analysis phase is crucial in establishing a strong system design and development foundation by thoroughly examining the project's requirements, objectives, and challenges. This involves extensive research, including reviewing related research papers, comparing three existing systems, and identifying areas for improvement. Interviews with Student Hostel Management staff are conducted to gather insights into workflow and functional requirements. The phase also includes defining software and hardware needs, aligning them with system functionality, and outlining the architecture of the Online Accommodation Management System. Preparations for the proposal defense are finalized, ensuring the project's scope is well-defined, and stakeholder needs are addressed.

The design phase is essential in project development, focusing on planning the system's structure, functionality, and user experience. This phase defines the specifications, features, user interfaces, architecture, and technology stack. Key components include process design, database design, and interface design. Tools like context diagrams and Level 0 Data Flow Diagrams (DFDs) model system operations, while Entity-Relationship Diagrams (ERDs) outline database structure. Wireframes provide low-fidelity visual layouts of the user interface, highlighting navigation and content placement. Interface design emphasizes user-friendliness, consistent navigation, responsive design, and accessibility, ensuring a seamless and intuitive user experience.

The implementation phase transforms designs and plans into a functional system by writing the code using selected programming languages, frameworks, and tools. Front-end development uses HTML, CSS, and JavaScript, while PHP is used for backend development. This phase involves creating system functionality, integrating components, and ensuring proper operation. Additionally, the database is designed and set up using MySQL, based on the Entity-Relationship Diagram (ERD) from the design phase. Documentation, including system architecture, database design, UI wireframes, and functional specifications, is the foundation for development and is updated throughout the process.

The testing phase is a crucial stage in the software development lifecycle, where the system is evaluated to ensure it meets requirements and functions correctly. It identifies and addresses bugs, errors, and issues related to functionality, performance, usability, security, and compatibility. Activities include creating a test plan, designing test cases, executing tests, logging defects, and conducting retesting or regression testing. In this project, functional testing and user acceptance testing (UAT) were performed with the participation of staff from the Student Hostel Management Department and UTHM hostel students. This phase ensures the final product meets user expectations, delivers quality, and minimizes the risk of failure during deployment.

The maintenance phase is the final stage in the software development lifecycle, where the product or system is monitored, updated, and improved after its deployment to ensure it continues to function effectively and meets user needs. During this phase, the author addresses issues such as bug fixes and performance enhancements and adapts the system to changes in the environment, such as new hardware, software, or regulatory requirements. Additionally, this phase involves implementing new features or upgrades based on user feedback to keep the system relevant and efficient. Regular maintenance ensures the system remains reliable, secure, and capable of supporting organizational goals over time, extending its usability and value.

3.1 System Requirement Analysis

The software development process begins with requirement analysis, a critical phase determining the project's success. Continuous communication with stakeholders and end users is key in defining expectations, resolving potential issues, and recording all requirements [5]. Therefore, functional and non-functional requirement analysis is a pivotal phase in the software development process, enabling the understanding, documentation, and definition of user and stakeholder expectations for the software application. The functional requirement of the developed system is analysed as shown in Table 2.

Table 2 *Functional Requirement*

Main Module	Requirements
Registration	1. Allow students to create accounts. 2. Validate user details during registration (e.g., email, username, password). 3. Send confirmation notifications upon successful registration.
Login and logout	1. Enable secure login for students using hashed credentials. 2. Provide a logout function to terminate user sessions securely.
Student site	1. Enable students to apply for a hostel. 2. Allow students to choose room. 3. Provide a feature for booking rooms at Wisdom Inn. 4. Display a list of roommates. 5. Update the list dynamically as changes occur. 6. Allow students to view and edit their profiles.
Admin site	1. Generate and assign key slots for students. 2. Review and approve or reject student hostel applications. 3. Notify students of the application status. 4. Create and publish notifications or announcements for students. 5. Assign room for students.

Including non-functional requirements is a vital aspect of software development. These requirements encompass performance, security, reliability, compatibility with various devices, browsers, or operating systems, usability, and scalability. Each requirement should be detailed alongside its evaluation criteria [8]. The non-functional requirements specify the criteria for evaluating the system's performance, unlike the functional requirements, which specify certain activities and behaviours of the system. Table 3 shows the non-functional requirements for the developed system.

Table 3 Non-*Functional Requirement*

Requirement	Description
Operational	1. The system must be available 24/7 with minimal downtime for maintenance. 2. The system must support simultaneous access from different locations and devices without disruption.
Performance	1. The system should ensure page load times under 3 seconds for optimal user experience.
Security	1. User passwords must be stored securely using hashing algorithms and not accessible in plaintext. 2. The developed system can only be accessed with the correct username and password.
Scalability	1. The system must scale to accommodate increased users and hostel applications without significant performance degradation. 2. It should be designed to handle additional modules and features in the future with minimal code changes.

Table 3 Non-*Functional Requirement (continued)*

Requirement	Description
Usability	1. The user interface must be intuitive and accessible, with easy navigation for students and admins.
Maintainability	1. The system should be modular and well-documented to allow for easy updates, bug fixes, and future enhancements.

4. Results and Discussion

This section discusses the results and discussion of the project, which are the system analysis and design, implementation, and system testing results.

4.1 System Analysis and Design

System analysis and design is the process of studying and planning the structure of a system to ensure it meets specified requirements. The system's objectives, problems, and user needs are identified during analysis through research, interviews, and requirement gathering. The system's architecture, features, and user interfaces are defined in the design phase, including database structures and process flows, often represented through diagrams and wireframes. This phase ensures a clear blueprint for development, aligning functionality with user needs and technical capabilities.

4.1.1 Data Flow Diagram

A Context Diagram is a high-level visual representation of a system and its interaction with external entities. It provides an overview of the system's boundaries, inputs, outputs, and the entities interacting with it. The diagram uses a single process node to represent the system and shows how data flows between it and its external environment. By offering a high-level overview, the context diagram enables stakeholders and developers to grasp how the system integrates into its larger environment quickly. Figure 3 shows the context diagram and the two entities in the developed system: students and administrators. The Online Accommodation Management System context diagram illustrates the interactions between two external entities, students and administrators, and the system. Students can register, log in, apply for a hostel, check in or out, and edit their profiles, while administrators manage key slot generation, approve applications, and create announcements. The diagram outlines the data flow between these entities and the central system, providing a clear overview of system boundaries and key processes.

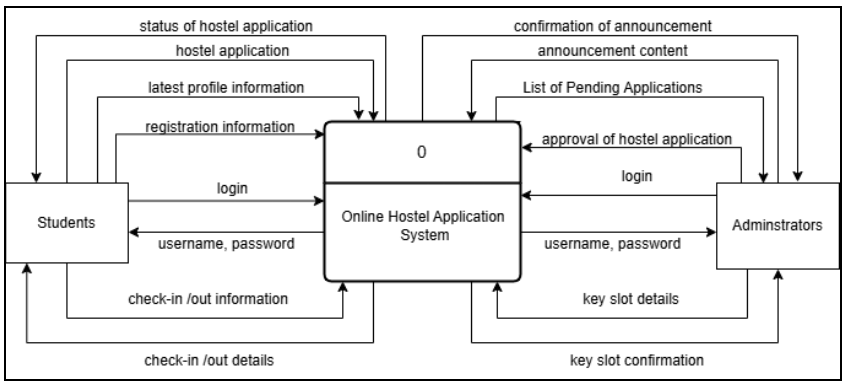


Fig.3 *Context diagram for the developed system*

4.1.2 Level 0 DFD

A DFD Level 0 (Data Flow Diagram Level 0) provides a high-level overview of a system, illustrating its significant processes and how they interact with external entities like users or other systems. It shows the flow of data between these external entities and the central system, as well as between the system's leading processes, without delving into internal details. The diagram helps define system boundaries, identifies key data inputs and outputs, and offers a foundation for more detailed system analysis in subsequent levels. Figure 4 shows the DFD level 0 for the developed system. The developed system consists of seven processes, which are register (process

1.0), login (process 2.0), apply hostel (process 3.0), check-in/out Wisdom Inn (process 4.0), manage student profile (process 5.0), generate key slot time (process 6.0), and make announcements (process 7.0).

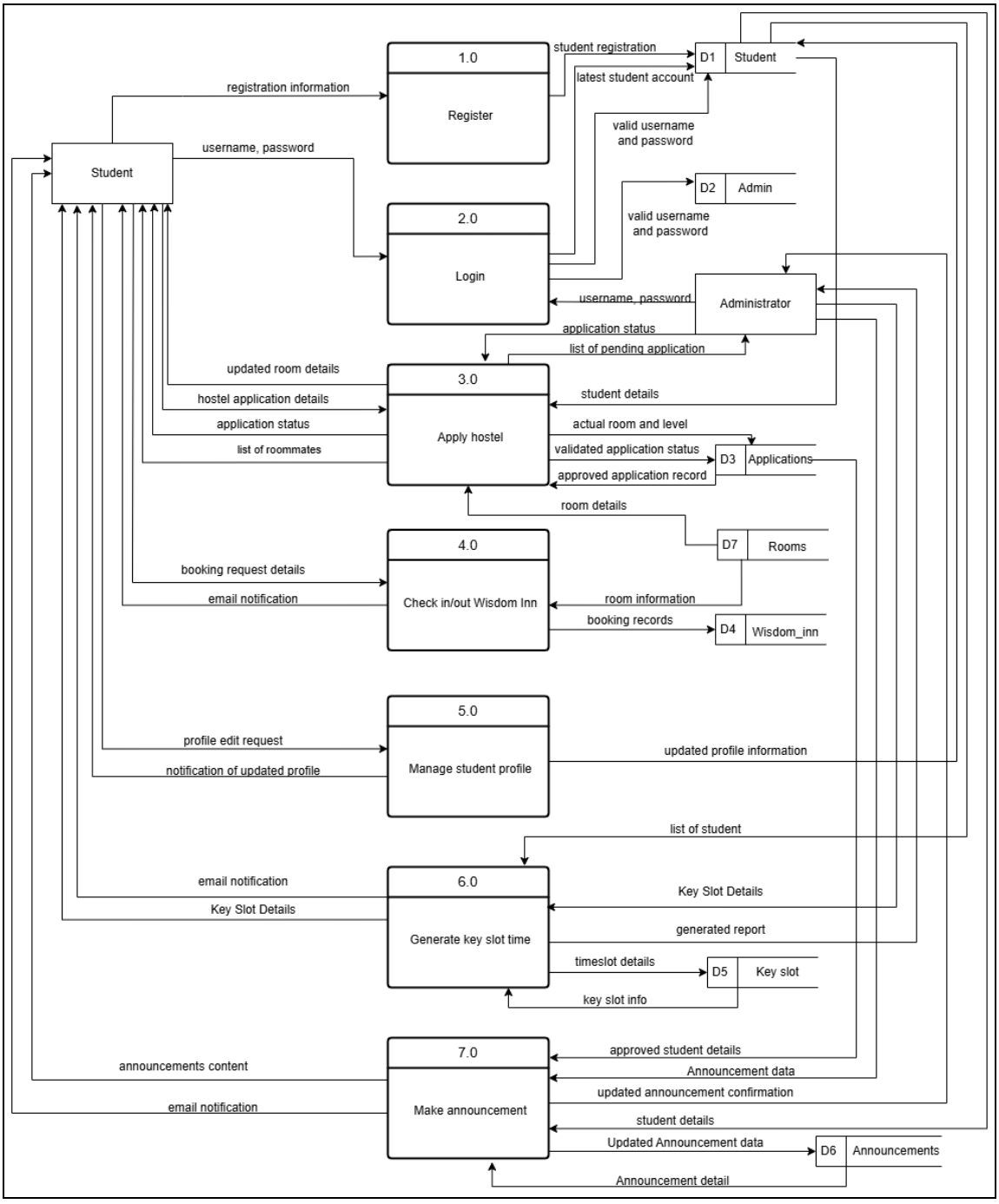


Fig.4 *Data Flow Diagram Level 0*

4.2 Entity Relationship Diagram

An Entity-Relationship Diagram (ERD) is a visual representation of the data and relationships within a database. It is a critical tool in database design, used to illustrate the system's entities, attributes, and connections. ERDs help in understanding and communicating the database structure during the design phase. Figure 5 shows the ERD diagram for the developed system.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 130-149

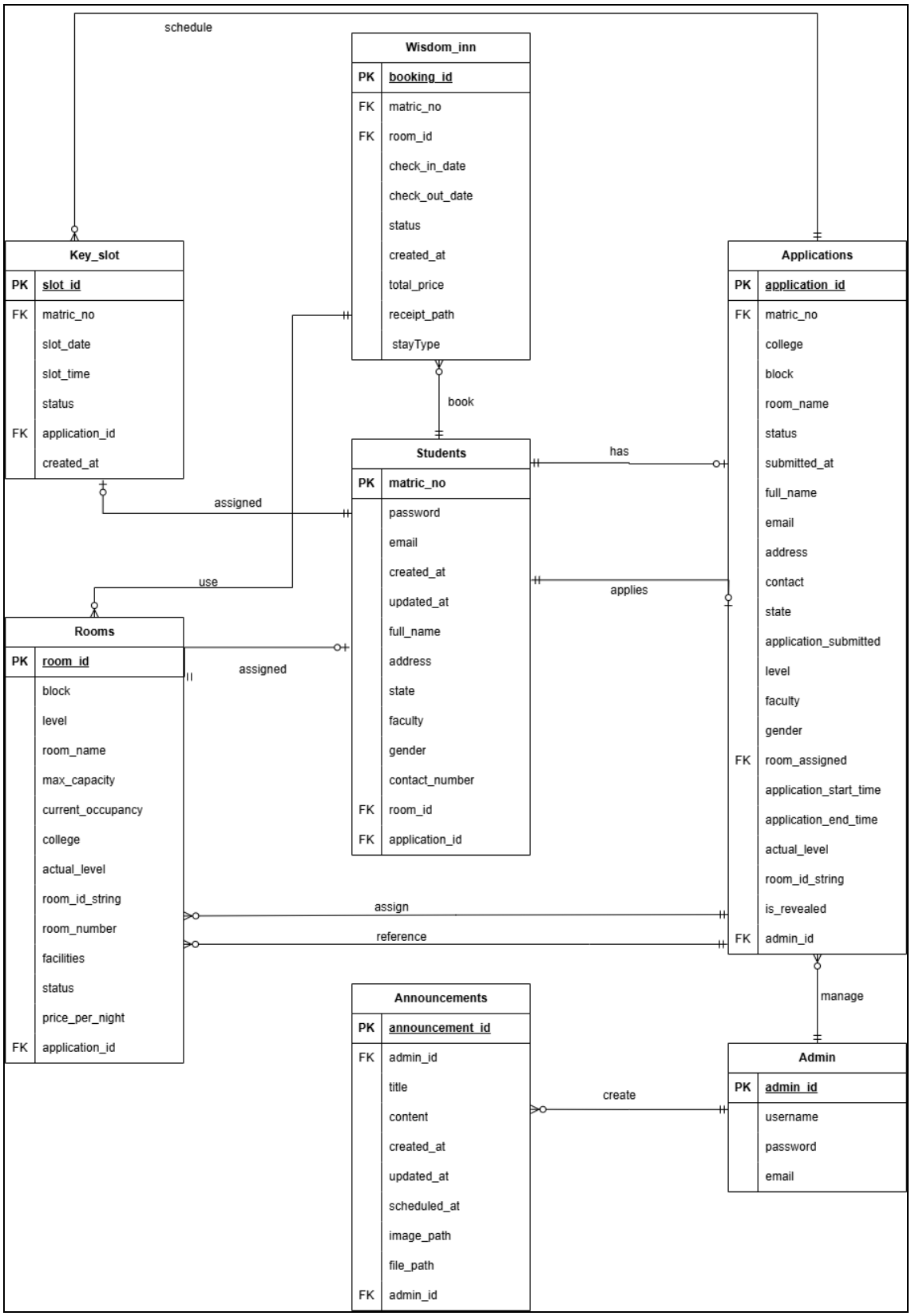


Fig.5 *ERD diagram for the developed system*

4.3 Implementation of Module

The implementation phase of a developed system is the stage in the system development lifecycle where the designed and developed system is put into operation. The system was implemented with a front-end built in HTML, CSS, and JavaScript, and a back end powered by PHP integrated with a MySQL database.

4.3.1 Login/Logout and Registration Module

Figure 6 shows the interface for the students to log in to the developed system. The students are required to log in with their matric number and password. If the login credentials are correct, the students will be directed to the homepage, or else the students need to enter the correct details again.

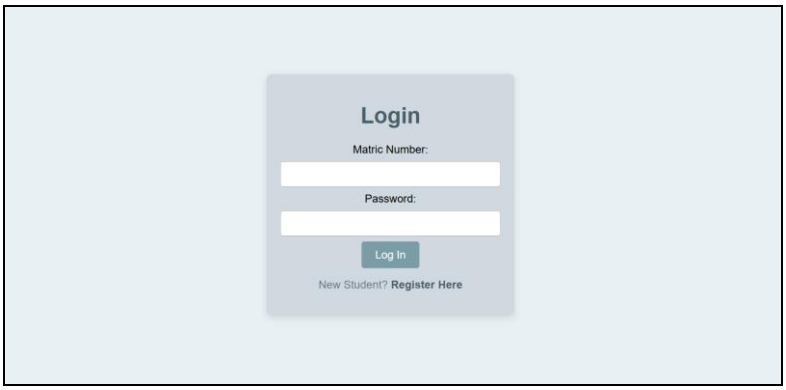


Fig.6 *Login page*

Figure 7 shows the interface for the students to register for the account. The students can register using a form requiring their matric number, email, and password. A successful sign-up message will be shown upon a successful registration.

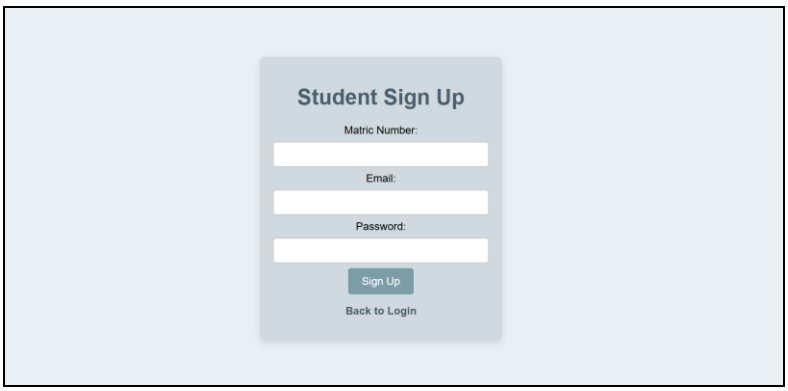


Fig.7 *Student registration page*

4.3.2 Hostel Application Module

In the hostel application module, students can submit applications for UTHM hostels and track their application status. After applying for a room, they can also view their assigned roommates' names. Figure 8 illustrates the hostel application interface, where students can view their application status and roommates' details. When students get approved by the admin, their status will turn from “Pending” to “Approved”, but the level and the room will remain “Pending” until the admin reveals the actual level and room for the student. The assign and reveal room function is explained in Figure 11. Also, the student can delete the application within the pre-registration period by clicking on the “Delete” button.

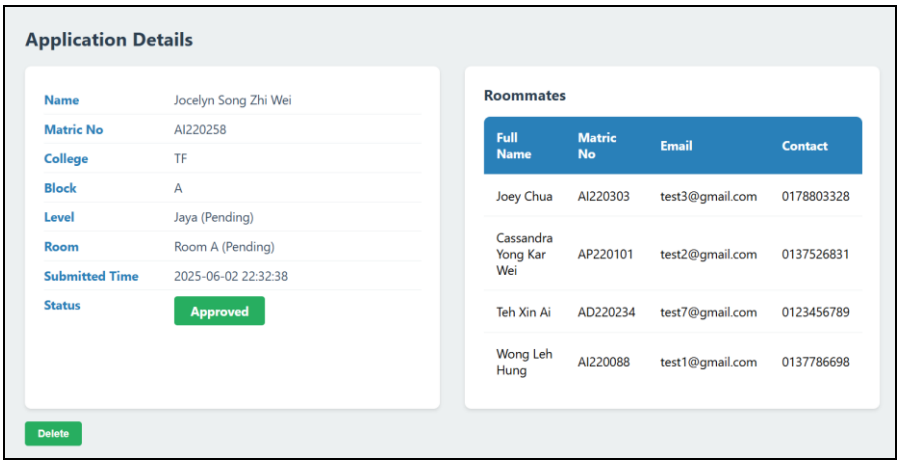
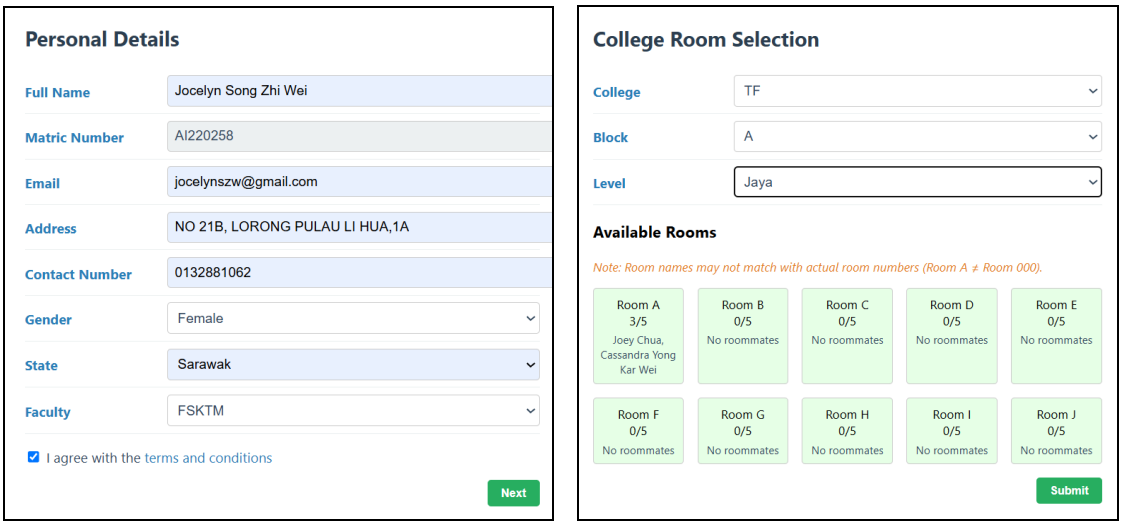


Fig.8 *Dashboard for application status and roommates' details*

Figure 9 (a) and (b) show the application form and room selection form. Students can fill in their personal information on the application form page. Then, they can click "Next" button to direct to the room selection page to select their preferred college, block, level, and room. The level and room are assigned anonymous names, with the final allocation determined by the administrator.



(a) **(b)**

Fig.9 *Forms to handle room submit (a)Application form; (b)Room selection form*

For the administrator site, administrators can review applications through the "Manage Application" dashboard. Figure 10 shows the interface for the admin to approve, reject, or recover the students' applications based on availability and criteria. The admin can approve and reject the student's application by clicking on the "Approve" and "Reject" buttons. Moreover, the admin can recover the student's application if they approve or reject it due to a mistake. The "Approved Selected" and "Reject Selected" buttons allow the admin to approve or reject the selected student by selecting multiple students at the same time. The admin can use the search bar to search for the students.

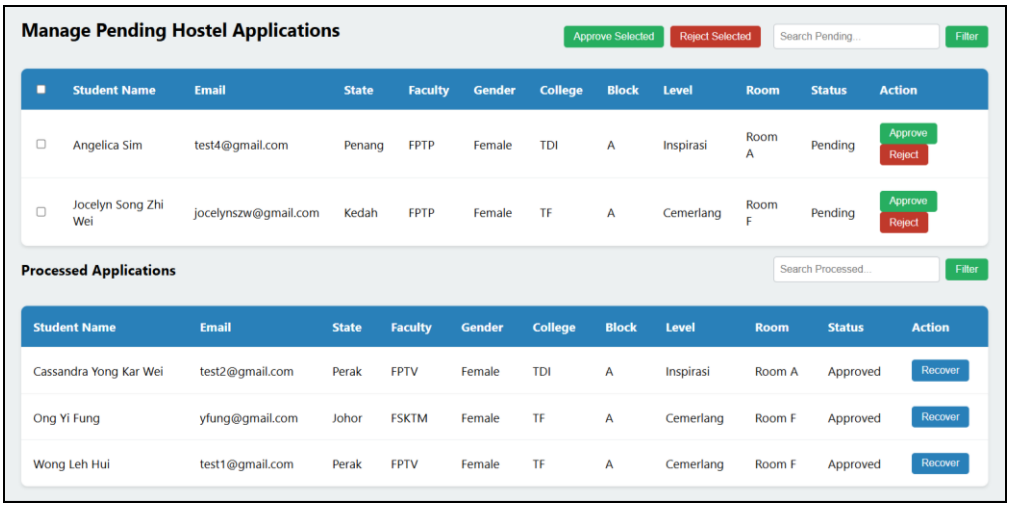


Fig.10 *Application management page*

The admin can assign a room for the students after they have approved the student's application. Figure 11 shows the pending rooms, which is the group of students who have not been assigned and revealed the room. The group of students who applied for the same college, block, level, and room is displayed in the same section. Admin needs to assign and reveal the actual level name and room number to the students because when students are choosing and applying for the hostel, they cannot view and choose the actual level and room number. The levels are shown with unique words such as “Gemilang, Jaya, and Bistari” while the rooms are shown as “Room A, Room B, and Room C” to prevent students from selecting the levels and rooms by themselves to make sure it is fair for every student when applying to the hostels. Figure 12 shows that the admin can unrevealed the student if they reveal the student accidentally.

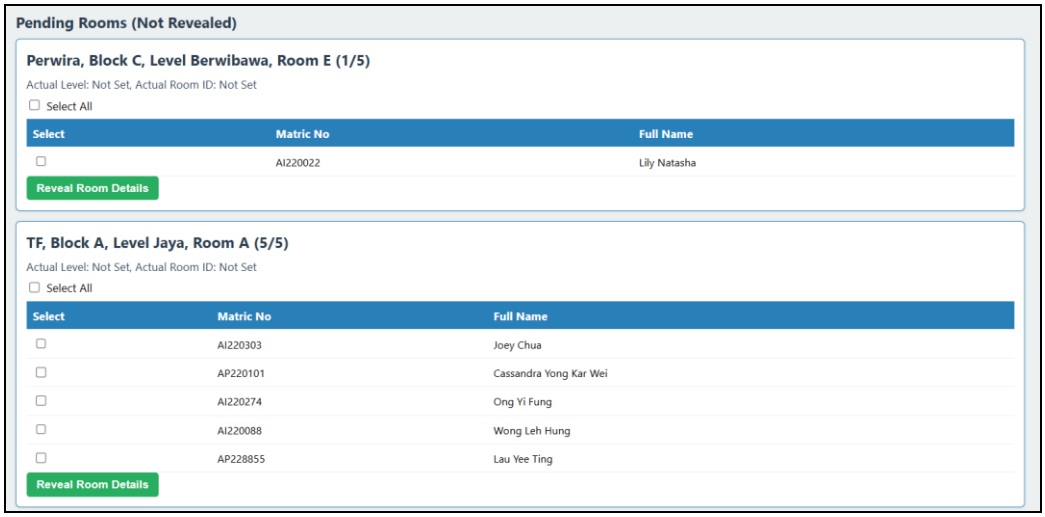


Fig.11 *Assign and reveal the room function*

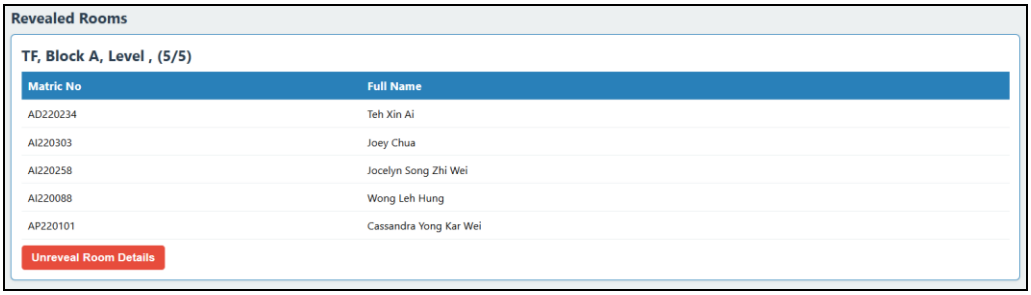


Fig.12 *Unrevealed room function*

Moreover, the admin can merge the students if they find the room is not fully occupied by students. Figure 13 shows the function of merging students into one room. The merging concept is followed by the majority of the students, which means that if one room has more students, then the system will merge other students into the room with majority of the students. The admin can click the checkbox to select all the students and merge them into one room.

Fig.13 *Merge room function*

4.3.3 Student Profile Module

The student profile module enables students to modify their personal information whenever updates are needed. Students can conveniently revise their details by completing a form to ensure their information is accurate. Figure 14 displays the form used by students to view and edit their information.

Fig.14 *Edit form*

4.3.4 Key Collection Time Module

Administrators generate key collection timeslots in the key collection time module and organize them based on students' faculties. This structured approach ensures efficient key distribution. Figure 15 shows the key timeslot management module's interface. The admin can generate a key timeslot and date based on the students' faculty. Multiple faculties can be selected by holding the "Ctrl" key on the keyboard.

After generating the key timeslots, the admin can choose the timeslots and generate timeslot reports to view all the timeslot's student details, such as students' name, matric number, and their respective faculties, and download the report in the PDF format. When the timeslot report is generated, an email will automatically be sent to the email address that the students registered with. The admin can also delete the timeslot if they want.

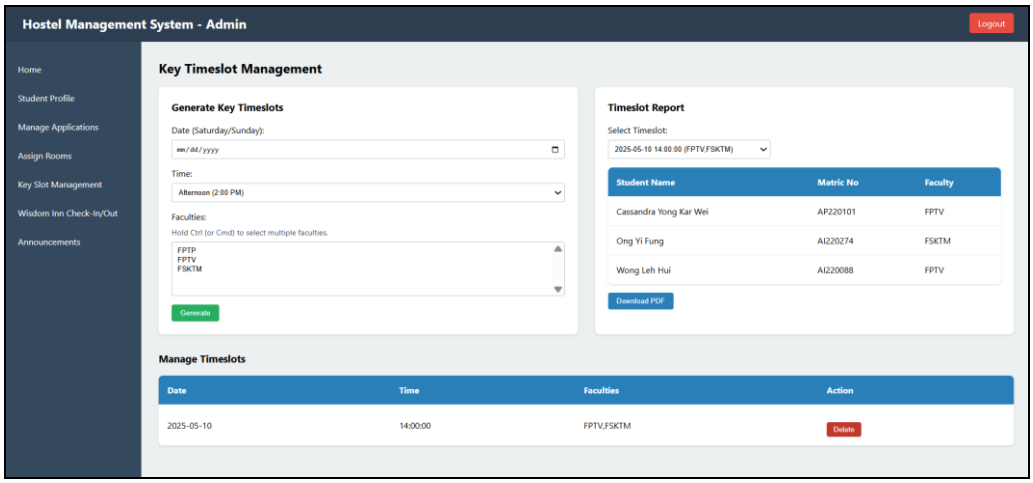


Fig.15 *Key Timeslot Management page*

4.3.5 Wisdom Inn Check-in/out Module

The Wisdom Inn Check-in/out module provides a function for students to book their long-term or short-term stay in UTHM. Students can choose the staying type, which is short-term or long-term as shown in Figure 16. For short-term staying, students can choose the check-in and check-out dates that they want to stay in the Wisdom Inn. Students cannot choose the dates if they choose the check-out dates earlier than the check-in dates or choose the check-out dates more than 14 days. For long-term staying, which is four months for one semester, students can directly go to the next page to view available rooms when they click the button.

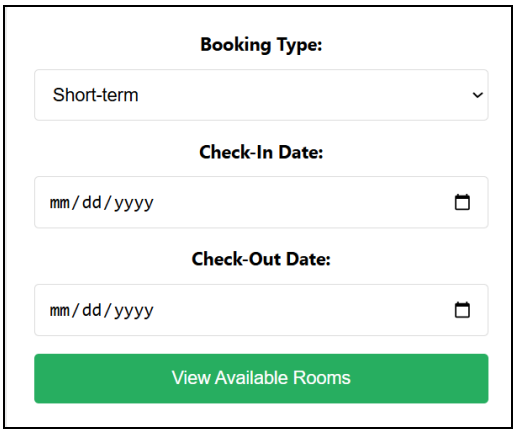


Fig.16 *Booking form*

Figure 17 shows the available rooms page where students can choose rooms and check for the total price based on their booking type choices. The total price and the booking days for short-term booking will be calculated automatically and shown on the page. For long-term booking, the page displays the rental per month and the total price for one semester. After students click the "Choose" button, a booking details is displayed. Students need to pay the rental in UTHM's Sistem Maklumat Akademik Pelajar (SMAP) first. After that, they need to upload the e-receipt in the upload receipt section shown in Figure 18. Then they can click the "Confirm Booking" button to complete the booking process.

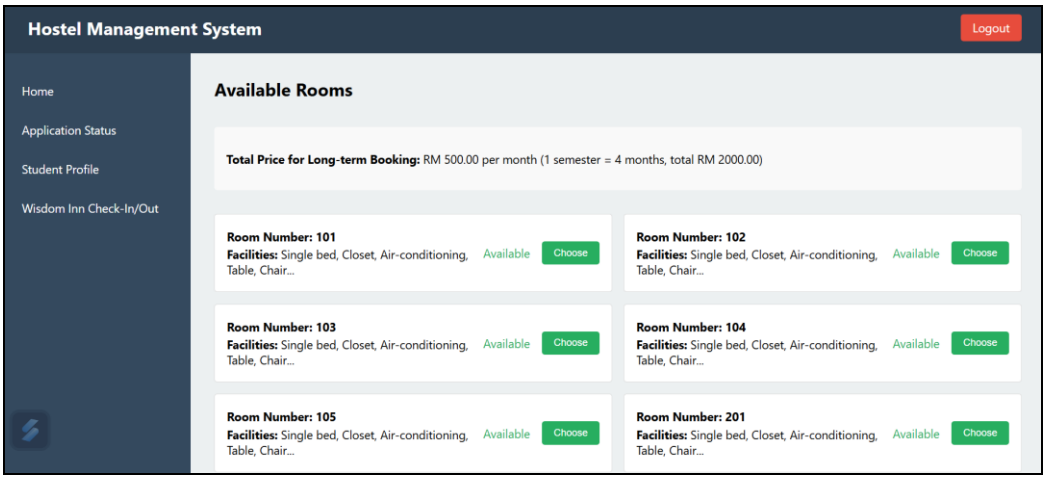


Fig.17 *Available room page*

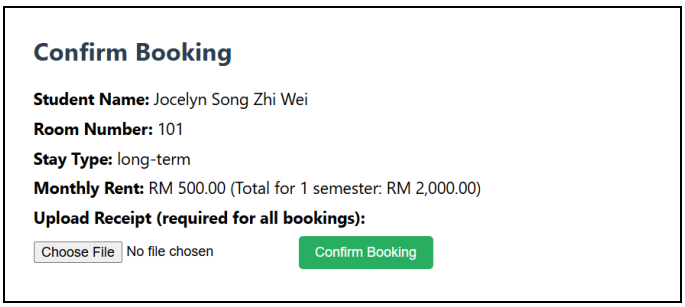


Fig.18 *Confirmation of booking page*

For the administrator site, the admin can view the booking details and click to view the receipt uploaded by the students as shown in Figure 19. This can double check whether the students make payment or not. There is a search function for the admin to search a student from the list of bookings.

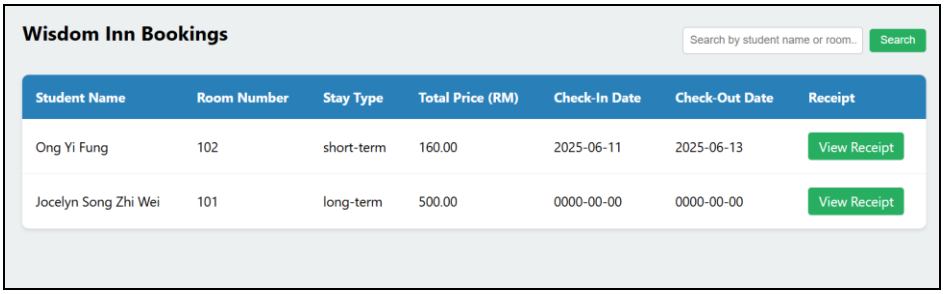


Fig.19 *Wisdom Inn bookings details page*

4.3.6 Announcement Module

Administrators utilize the dashboard to create and post announcements, such as deadlines and events, which are stored in the announcements table within the system. This allows for efficient communication of important updates. Figure 20 shows the announcement posting interface for the admin to post the announcement, and Figure 21 shows the interface for the admin to edit and delete the announcement. Figure 22 shows that the admin can edit the title, content, upload a new file, and reschedule the time. When an announcement is edited, it shows an “edited” sign in orange colour on both the admin and the student site to remind them that this announcement has been edited. Students can easily access and view these announcements on the system's homepage, as depicted in Figure 23, ensuring they stay informed about relevant hostel-related information.

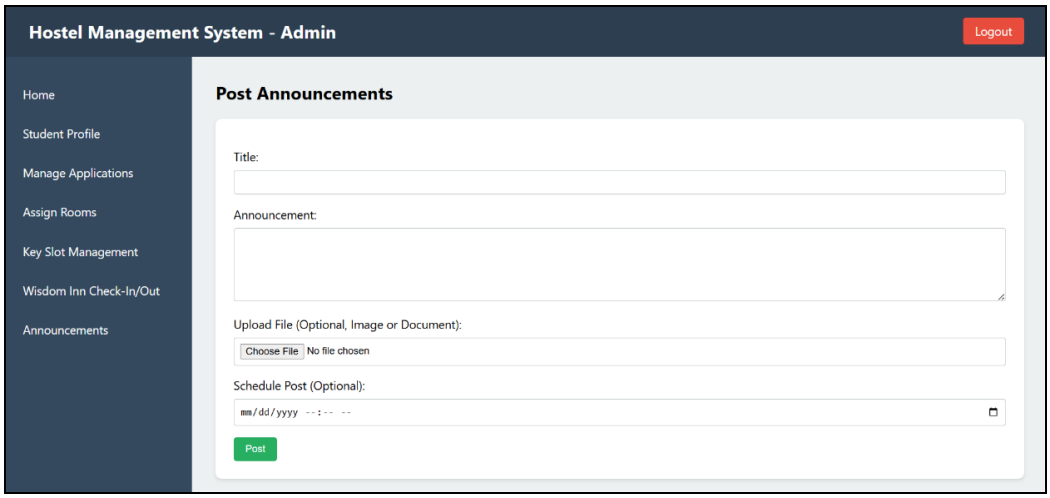


Fig.20 *Announcement posting page*

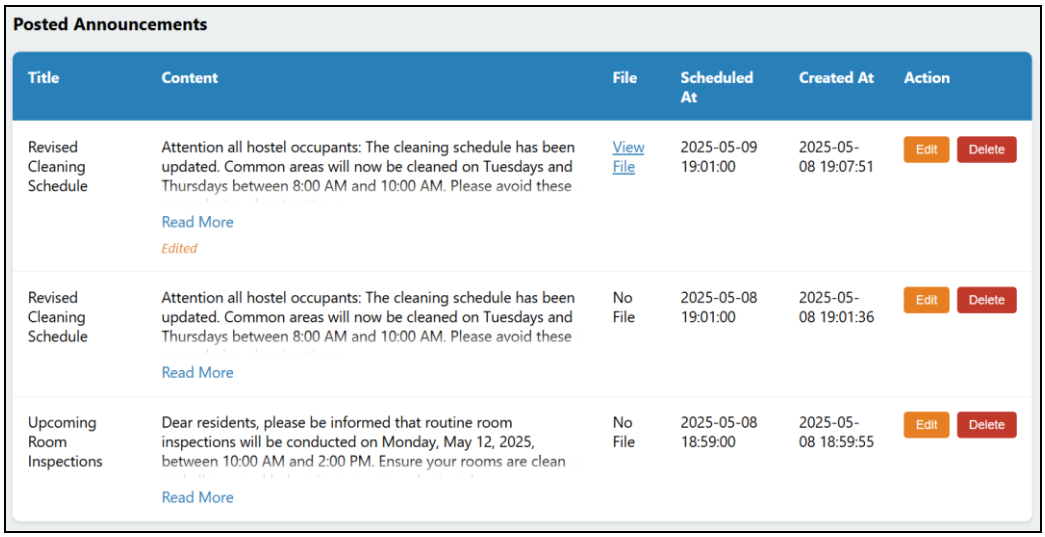


Fig.21 *Edit and delete action page*

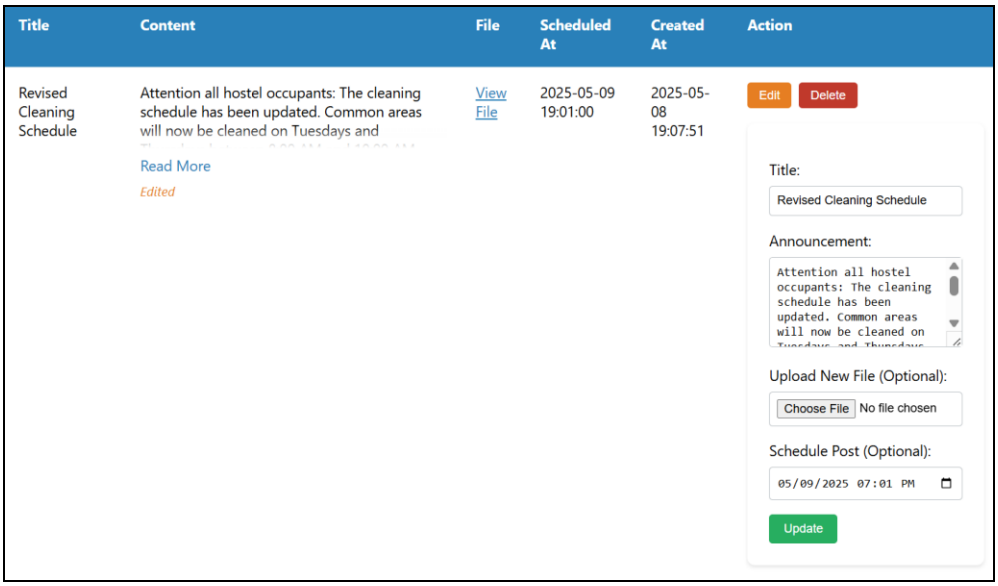


Fig.22 *Edit function*

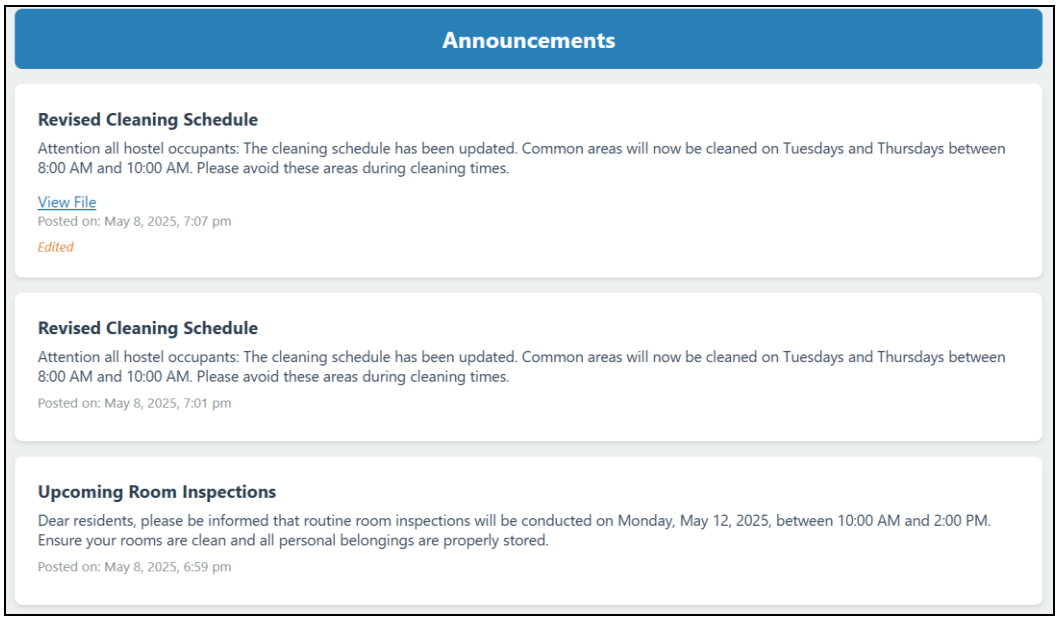


Fig.23 *Announcement page (Student site)*

4.4 Testing

4.4.1 Functional Testing

Functional testing is a type of software testing that focuses on verifying that the software operates according to its defined requirements and specifications. It involves testing each function of the system by providing input and examining the output, without considering the internal code structure. One of the key aspects of functional testing is the use of test cases, which are predefined sets of inputs, execution conditions, and expected results used to validate system behaviour under multiple conditions. These test cases help ensure that all features work correctly across different scenarios, including normal, boundary, and error conditions. Functional testing covers areas such as user interface interactions, data processing, security features, and integration with external systems, ensuring the application performs as expected from the end-user's perspective. Table 4 and Table 5 show the result for the developed system.

Table 4: Functionality test result for the developed system - Student site

No.	Test case	Expected result	Result
Login/logout function			
1	Valid matric number and password.	User is successfully logged in and redirected to the dashboard, else an error message is shown.	Pass
2	Logout	User is successfully logged out and redirected to the login page.	Pass
Registration function			
1	Complete registration information.	A successful sign-up message is displayed else an error message is shown.	Pass
Home page			
1	Click "Check" to do pre-registration for hostel.	Redirect to the application status page to apply for a hostel.	Pass
2	Click "Check" to view application status.	Redirect to the application status page to view the status.	Pass
Apply hostel function			
1	If under pre-registration period, click "Apply Now" link to apply hostel, or else no link is displayed.	During the pre-registration period, the "Apply Now" link is visible and clickable. Outside the period, the link is hidden or disabled.	Pass
2	Complete the personal details form.	Form submission is successful. User proceeds to the next step of hostel selection.	Pass

Table 4: Functionality test result for the developed system - Student site (continued)

No.	Test case	Expected result	Result
3	Choose the college, block, level, and room accordingly.	Hostel selection is saved successfully, and a confirmation message is shown.	Pass
Student profile page			
1	Edit data such as email, contact number, address, and state.	The data are updated, and a successfully updated message is shown on the profile page.	Pass
Wisdom Inn booking function			
1	Click “Book Now” button.	Redirect to the available rooms page.	Pass
2	Complete booking form and click “Choose” button.	Redirect to the page of the room details.	Pass
3	Upload receipt and click “Confirm Booking”.	Confirmation message is sent to the student.	Pass

Table 5: Functionality test result for the developed system - Admin site

No.	Test case	Expected result	Result
Login/logout function			
1	Valid username and password.	Admin is successfully logged in and redirected to the dashboard.	Pass
Dashboard			
1	The “click” hyperlink can linked to the respective pages	Each hyperlink successfully redirects the user to the correct and corresponding page.	Pass
2	Set application dates and time.	Admin can set the application start and end dates and times. Settings are updated and a successfully updated message is displayed.	Pass
Manage students’ application function			
1	Approve student application.	Student site application status updates to "Approved", and a successful message is displayed.	
2	Reject student application.	Student site application status updates to "Rejected", and a successful message is displayed.	Pass
3	Recover student application.	Recover the student successfully and a message is displayed.	Pass
4	Search student based on their respective details.	Filter out the student who matches the keywords entered by the admin.	Pass
5	Reveal the pending rooms.	The students can view the actual level and room at the application status dashboard.	Pass
6	Unrevealed the room.	The group of students are displayed in the pending room section.	Pass
7	Merge the students into one room.	The students are merged into one room followed by the majority of students within the room.	Pass
Manage key timeslot			
1	Generate key timeslots for multiple faculties	Timeslots are successfully generated based on the number of students in each faculty, distributed across Saturday and Sunday only.	Pass
2	Generate the timeslot report.	A report is created summarizing the generated timeslots per faculty, including student counts and assigned slots.	Pass
3	Generate email to student once the timeslot report is generated.	Each student receives an email notification containing their assigned key collection timeslot and relevant details.	Pass
4	Delete timeslot.	The selected timeslot is removed from the system; associated students are updated or notified of the change accordingly.	Pass

Table 5: Functionality test result for the developed system - Admin site (continued)

No.	Test case	Expected result	Result
Wisdom Inn booking management			
1	View receipt uploaded by the students.	Direct to a new page to view the receipt uploaded by the students.	Pass
2	Search the student based on the student details.	The system successfully retrieves and displays the matching students based on the entered search criteria	Pass
Announcement management			
1	Post announcement.	A new announcement with title, content, file and date is successfully created, saved in the system, and immediately visible to all intended users.	Pass
2	Edit announcement.	The selected announcement is updated with the new content, title, file, or rescheduled time, and the changes are reflected in real time.	Pass
3	Delete announcement.	The selected announcement is permanently removed from the system and no longer visible to users.	Pass

4.4.2 User Acceptance Test

User Acceptance Testing (UAT) is the final phase of the software testing process where actual end-users test the system to verify that it meets their requirements and is ready for deployment. It focuses on validating the software's functionality, usability, and overall performance from the user's perspective in real-world scenarios. This section presents an analysis form of the UAT form used to evaluate the test cases for the developed system. The testing involved 10 respondents, and the results collected are evaluated and presented in graphs. The respondents are requested to complete the survey form to assess their acceptance and satisfaction with the developed system. Table 6 and Figure 24 shows the UAT based on user interface, while Table 7 and Figure 25 shows the UAT based on the system functionalities.

Table 6: User acceptance test based on user interface

No	Features	Ranking					Total
		1	2	3	4	5	
1	The system interface is visually appealing and clean.	-	-	1	5	4	10
2	The layout and navigation are intuitive and easy to understand.	-	-	1	7	2	10
3	Buttons, menus, and labels are clear and well-organized.	-	-	2	4	4	10
4	The system is responsive and loads pages quickly.	-	1	2	3	4	10
5	The color scheme and font style make the content easy to read.	-	-	-	3	7	10

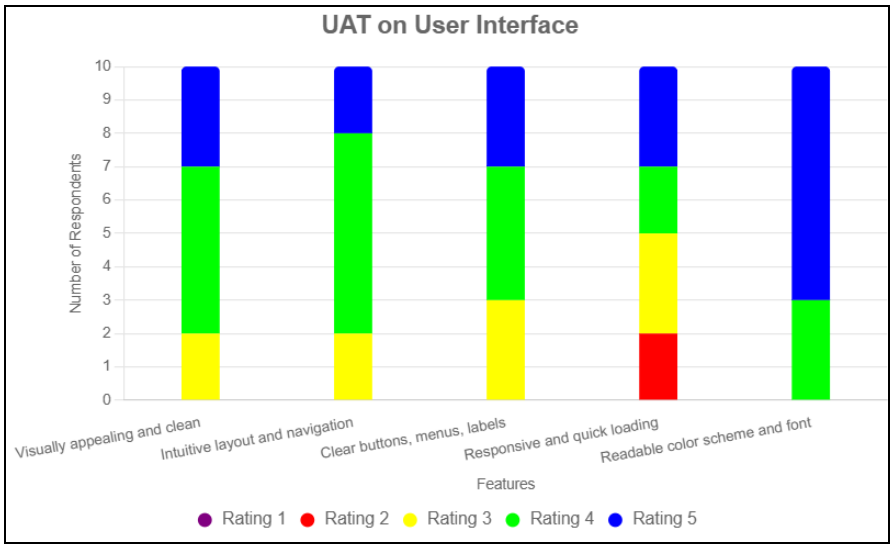


Figure 24: UAT result based on user interface

Table 7: User acceptance test based on functionalities

No	Features	Ranking					Total
		1	2	3	4	5	
1	Login and Registration	-	-	2	2	6	10
2	Student profile management	-	-	3	4	3	10
3	Hostel application management	-	-	1	4	5	10
4	Key timeslots and report management	-	-	4	4	2	10
5	Wisdom Inn check-in/check-out	-	1	3	2	4	10
6	Announcement	-	1	-	2	7	10

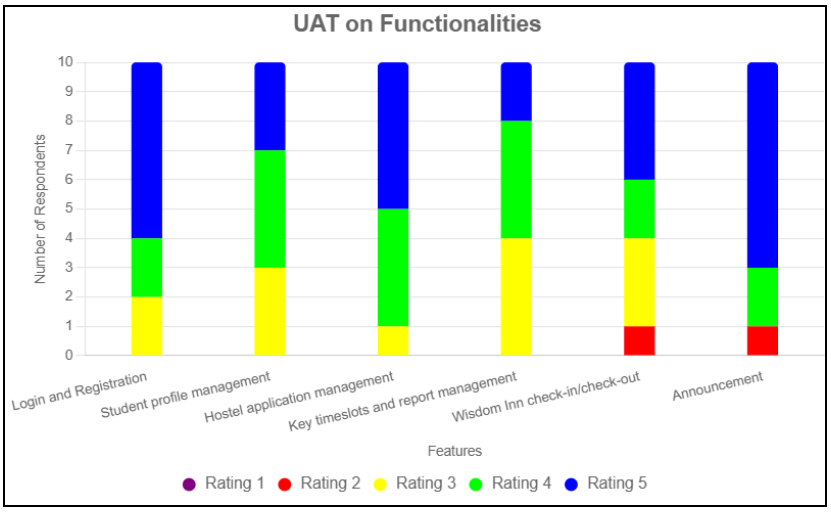


Figure 25: UAT result based on functionalities

5.0 Conclusion

The Online Accommodation Management System for UTHM students effectively addresses the inefficiencies of traditional hostel processes by offering a digital solution that enhances convenience and efficiency for students and staff. Through its key features, including self-registration, online applications, and time-slot-based key collection, the system achieves its objectives of design, development, and usability testing. While the system is robust and user-friendly, proposed future enhancements, such as mobile app development, analytics integration, multilingual support, and real-time notifications, will further improve its functionality and inclusivity. This project marks a significant step toward modernizing hostel management and sets a strong foundation for future advancements.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Jocelyn Song Zhi Wei, Noryusliza Binti Abdullah;* ***data collection:*** *Jocelyn Song Zhi Wei;* ***analysis and interpretation of results:*** *Jocelyn Song Zhi Wei, Noryusliza Binti Abdullah;* ***draft manuscript preparation:*** *Jocelyn Song Zhi Wei, Noryusliza Binti Abdullah. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Hostelworld, "What is a hostel? The answer will change your travels forever," *Hostelworld Travel Blog*, Aug. 04, 2023. https://www.hostelworld.com/blog/what-is-a-hostel/
[2] "Collins Dictionary - Hostel Meaning," *Collins*. https://www.collinsdictionary.com/dictionary/english/hostel
[3] Portal Rasmi PRP UTHM, "Pengenalan," *Portal Rasmi PRP UTHM*, Dec. 26, 2024. https://prp.uthm.edu.my/index.php/
[4] Creativ Eras - Software Development Company, "University College Hostel Management System | IFNOSS," *Ifnoss Campus Management System for University, College, School and Institute*. https://www.ifnoss.com/edu/college-university-software/hostel-management-system.aspx
[5] "Hostel management," *IEEE Conference Publication | IEEE Xplore*, Nov. 09, 2022. https://ieeexplore.ieee.org/document/10040481
[6] "View of UTHM Residential booking system." https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/12018/4324
[7] "System Architecture Diagram: Tutorial & Examples," *Multiplayer*. https://www.multiplayer.app/system-architecture/system-architecture-diagram/
[8] N. Ahmad, "What is requirement analysis: best practices and examples," Nov. 17, 2023. https://www.lambdatest.com/learning-hub/requirement-analysis