



Developing a Hostel Room Swap Management System - (Case Study of Bayero University Kano)

¹Hafsat Muttaka, *^{1,2}Kabir Umar, ^{2,3}Ahmad Abdulhamid Said and ²Aisha Kabir Umar

¹Department of software Engineering, Bayero University Kano, Nigeria

²Himma Computer Technologies, Kano, Nigeria.

³Department of software Engineering, Northwest University Kano State, Nigeria.

*Corresponding authors' email: ukabir.se@buk.edu.ng

ABSTRACT

In some Nigerian universities such as Bayero University Kano the current systems of swapping hostel rooms are uncoordinated and inefficient. Students often faced several challenges such as difficulty in finding other students who are willing to swap rooms, time wasted in searching manually through word of mouth or hostel posters. This paper presents a hostel room swap management system that enables students to exchange hostel rooms conveniently based on their preferences by automating and streamlining the process of requesting and approving room swaps. The system is developed using the python flask framework, HTML, CSS, Bootstrap and SQLite databases managed through SQLAlchemy. Usability evaluation was conducted using questionnaire-based approach to assess ease of use and overall user satisfaction. Overall satisfaction was notably high, with 64.5% of participants reporting they were very satisfied and 22.6% satisfied with their experience. Additionally, 90.4% of participants indicated their willingness to recommend the platform to others, underscoring its potential for wider adoption. The results indicate positive user acceptance, demonstrating that the system effectively meets its intended objective. Future improvements should include intelligent swap matching, automated notifications, and integration of the system with the university hostel management system to enhance efficiency.

Keywords: Student Accommodation, University Hostel, Digital Solution, Swap Matching

INTRODUCTION

Accommodation plays a significant and important role in the academic success and wellbeing of students in tertiary institutions. Hostels are provided to the students in tertiary institution to ensure affordable and accessible housing, influence their wellbeing, study habits and general campus experience but in many cases, the allocation of hostel rooms does not meet the preferences of students. Some students may be allocated to a room with roommates whose lifestyle or study pattern are incompatible. Sometimes, student may want to live in a specific block due to facilities available, and some may wish to move to a quieter or more special environment. Discrepancies in lifestyle, study patterns, and facility needs frequently lead to interpersonal conflicts, overcrowding, and a general decline in student satisfaction with hostel life (Ojetola & Gbadegesin, 2021).

Traditionally, Students who want to change rooms (i.e room swap) rely on informal and manual processes. This often includes word of mouth, posting handwritten notes on hostel notice boards or negotiating swaps through personal networks to find someone willing to exchange rooms. This manual process is not only time-consuming but also unreliable and inefficient, as it may not connect students with suitable matches and there is no guarantee that swap arrangements will be honoured. Furthermore, the absence of an official platform increases the risk of misinformation, duplication of requests, and exploitation by third parties who charge unauthorized fees for "securing" bed spaces (Okeke & Musa, 2023). With the advancement of web technologies, there is an opportunity to develop a hostel room swapping system that automates the process and bring solution to these challenges. Web-based systems have been widely adopted to streamline services such as course registration, hostel allocation, library access and e-learning. Research has shown that such systems improve transparency, speed and user satisfaction compared to manual approaches (Ogunyinka & Hassan, 2022).

Introducing a hostel room swapping websites aligns with this trend by reducing the stress faced by students in seeking room changes and by enabling administrators to manage hostel data more efficiently. It will also help reduce paperwork and support better hostel planning in future. In addition, the system promotes equity and accessibility. Instead of relying on privilege, social connections, or luck, every student would have equal access to swap opportunities once their request is logged into the system. This fosters fairness and transparency in hostel management, qualities that are often lacking in traditional methods (Salami & Obasi, 2023). Moreover, the system can serve as a model for other student accommodation services, highlighting how technology can address practical campus challenges.

Therefore, this study proposes an automated and student-centred hostel room swapping system that is aimed at addressing the aforementioned challenges inherent in the manual room swap approach. The system serves as a digital platform which seeks to contribute in improving hostel management, enhanced student satisfaction, and technology-driven campus environment. The remaining of this paper is organized as follows: Review of related literature is presented in Section 2.0. Thereafter, Methodology is presented in Section 3.0, followed by Implementation in section 4.0, and Testing and Results in Section 5.0. The paper is concluded in Section 6.0.

Literature Review

This section reviews existing literature relevant to hostel room management and swapping systems. It explores different approaches, technologies, and methodologies adopted in related studies.

Bista and Karki (2021) conducted a study on digitalized hostel systems with a focus on space optimization and data integration. The system was developed as a web application using the Python programming language and Django framework. It identifies key functionalities such as sorting

algorithms for room allocation and centralized inventory management. Anis & Mohd Safar (2022) designed a web-based hostel application system to manage student applications. Built with PHP and MySQL, their work focused on form submission and student data management but lacked features Room swapping or roommate matching.

Waibi (2022) implemented a web-based hostel allocation system with HTML, CSS, PHP, MySQL, proving login and database management. However, the design was more centred on administrator, leaving students with little control over reallocation or swaps. Bhardwaj et al. (2022) reviewed the integration of AI and IoT systems in automating dormitory environments. While primarily a system study, it identifies the use of intelligent frameworks for predictive booking and smart environment management. The work highlighted how modern systems move toward "intelligent" allocation rather than just simple manual entries.

Ukper and Ibekwe (2023) designed an integrated role-based HMS to enhance efficiency and transparency in university accommodation. The system was developed using Javascript and MySQL as a web-based solution. The application features robust Role-Based Access Control (RBAC) and audit trails, which are essential for managing high-level administrative approvals like room swaps. Oloyede et al. (2024) developed an off-campus hostel management system that automated hostel allocation, registration and reporting using HTML, CSS, JavaScript and MySQL. Their work highlighted how digital solutions can streamline allocation processes but did not address flexibility in room Swapping.

Diyaolu et al. (2024) proposed an e-based hostel management system that supported automated registration and room allocation. The system was built with PHP and MySQL, it streamlined processes but excluded mechanisms for swapping between students. Adekola et al. (2024) explored the design and implementation of a student hostel management system at Babcock University to solve ineffective room assignments. The system was built using PHP, HTML, and CSS, focusing on creating a responsive web environment for students. Key functionalities identified include an automated allocation engine and a student dashboard for viewing accommodation status.

Pawar and Shaikh (2025) implemented a hostel management system focused on real-time web application and database architecture. Their software uses JavaScript and Node.js to ensure high-speed data processing. The system provides real-time updates on room availability, which is a critical functionality for ensuring data integrity during a room exchange process. Eweoya et al. (2025) investigated the

development of a web-based hostel management system aimed at reducing manual processing delays and improving transparency. Their software is a web application developed using the PHP programming language and MySQL database. The application provides functionalities such as real-time room status tracking, automated allocation, and digital record management to eliminate paper-based inefficiencies

From the reviewed works, it shows that most hostel management systems prioritize allocation and registration but does not address flexibility such as room swap. The review also outlines the importance of adaptability, while few reviews show peer-to-peer frameworks that align more closely with room swapping, thus suggesting the need for developing a new platform which focuses on room swaps.

MATERIALS AND METHODS

This section explains the methodology used for the research work. The methodology includes the analysis and design phases that were crucial in shaping the functionality and structure of the system. This section also presents both functional and non-functional requirement that guided the development process. Various model diagrams such as Use Case, Class, Activity, and Entity-Relationship Diagrams (ERD) are used to illustrate the system's structure and functionality.

Research Workflow: The research followed a systematic workflow from conception to completion. The various stages of the study are illustrated in Fig. 1.

The research adopted a structured and systematic workflow to guide the development of the Hostel room swap system from conception to evaluation. The process began with an Introduction, which established the problem context and objectives of the study. This was followed by a Literature Review, where existing works on hostel room management and swapping systems were examined. The Analysis and Design phase then translated the gathered insights into system requirements, use case models, and architectural blueprints. Subsequently, the Implementation and Testing phase involved the actual development of the platform using the selected technologies, accompanied by rigorous usability testing to validate system functionality. The Evaluation and Result phase assessed the platform's performance against the defined requirements, with findings documented to draw conclusions and inform recommendations. Each phase fed directly into the next, ensuring a coherent and traceable development process throughout the study.

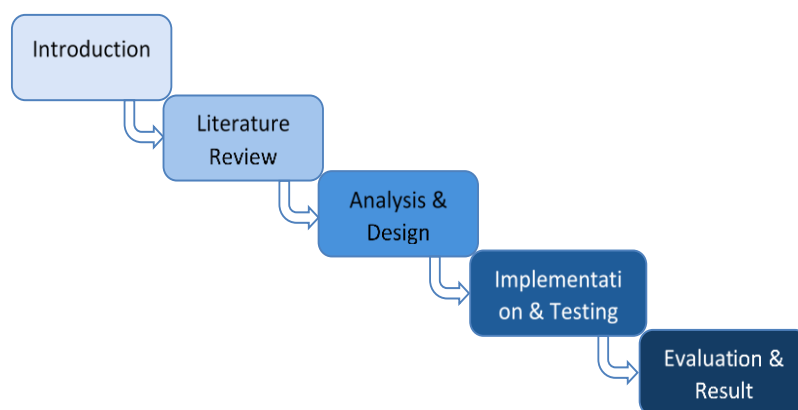


Figure 1: Research Workflow

Software Development Model

In this study, the Agile Software Development Life Cycle (SDLC) Model was adopted for the hostel room swap application project. Agile SDLC was chosen because it supports iterative development, continuous feedback, flexibility for changing requirements, continuous

improvement and frequent testing. This approach enables rapid prototyping, early delivery of functional features, and the ability to adapt to changing requirements. It is particularly suitable for web applications where features can be developed and tested in iterations, allowing for quick adjustments based on user feedback and stakeholder needs.

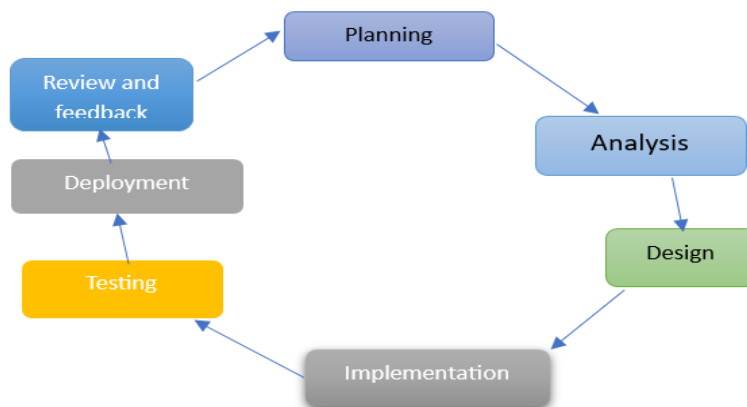


Figure 2: Software Development Life Cycle

As shown in figure 2 above, the platform development project is carried out in a number of phases, namely; planning phase, analysis phase, system design phase, implementation phase, testing phase, deployment phase, review and feedback phase.

- i. **Planning:** This phase involves general scheduling of project activities and resource allocation. The planning is revisited at the beginning of each iteration so as to refine schedules based on risk analysis and priorities.
- ii. **Analysis:** This phase involves gathering detailed functional and non-functional requirements from stakeholders (Students and Hostel Admin).
- iii. **Design:** This phase involves creating architectural diagrams, database design, and layouts based on the requirements.
- iv. **Implementation:** This phase involves coding the Hostel room swap management system based on the detailed design and specification.
- v. **Testing:** This phase involves conducting usability testing to validate functionalities.
- vi. **Deployment:** This phase involves preparing the application for deployment with proper configuration and documentation. The hostel rooms swap system was deployed as a web application accessible via the internet.
- vii. **Review and feedback:** This phase involves collecting review and feedback from students that use the system during usability testing so as to identify areas that require further enhancement.

Analysis of existing system

This section presents an analysis of the existing hostel room swap system and the proposed system. The existing system is largely manual and inefficient, while the proposed system introduces an automated and web-based approach aimed at improving efficiency, transparency and overall user experience.

Description of Existing System

The existing system for managing hostel room swaps at Ramat Hostel Bayero University Kano is predominantly manual, involving physical paperwork, face-to-face meetings,

and email communications. Students who wish to swap rooms must fill out paper forms, submit them to the hostel administration office, and wait for manual processing. The administration manually matches swap requests by reviewing paper records and coordinating between students. The existing system for managing hostel room swaps is used by the following actors:

Administrative Staff:

Role: Handle all room allocation and swap requests, which involve managing physical forms and emails. They are responsible for data recording, physical matching of swap requests, giving approvals, and manually notifying students.

Students

Role: Initiate a swap by submitting a request to the administration with their current and desired room. They repeatedly follow up administration to check the status of their request or to find a suitable match.

Requirement Elicitation

Requirement Elicitation is the process of gathering and defining the requirements for a software system. In this study, Requirements were gathered through multiple methods as highlighted below.

Interviews with Hostel Administrator understand s and Students

This involves direct conversation with selected student and hostel administrators so as to current processes, pain points, and desired features.

Observation of Current Manual Process

We carried out direct observation of the existing workflow at the hostel administration office. This enabled us to see how students submit requests, and how their request is being processed.

Findings of the requirement elicitation process in terms of key needs of students as well as hostel administrators are summarized below.

- i. **Student's key needs:** The student generally wants to have the following features in the proposed system; a self-service portal to create room swap requests, view matches and receive real-time notifications on the status of their swap; want to see available swap options

- ii. Hostel Admin key needs: The hostel administrators want to have the following in the proposed system; Centralized tools for approving or rejecting swaps; bulk data import or export and a clear activity log for audit purposes; want to have automated matching system to reduce manual work; Require secure authentication and authorization.

Requirement Definition

Requirement definition is the process of identifying and documenting what a system should do and how it should perform it. It helps ensure the system meets user needs. Based on findings from requirement elicitation exercise, the following functional and non-functional requirement are defined for the proposed system

Functional Requirements; the key functional requirements for the proposed system are listed below.

- i. Admin shall be able to Login with admin credentials
- ii. Admin shall be able to view dashboard with statistics (total students, pending swaps, approved/rejected counts)
- iii. Admin shall be able to view all swap requests with detailed information

- iv. Admin shall be able to approve or reject swap requests with reason comments
- v. Admin shall be able to Bulk import students from CSV file
- vi. Admin shall be able to Export student data to CSV
- vii. The system should allow students to login with student ID, and logout securely
- viii. The system should allow the students to view and update personal profile (hostel block, room number, preferences)
- ix. The system should allow students to create swap requests with current and target room information
- x. The students should be able to view matching swap suggestions from other students
- xi. The students should be able to send proposals to other students swap requests
- xii. The students should be able to view all their swap requests with status (pending, approved, rejected)
- xiii. The students should be able to receive notifications about swap status changes
- xiv. The students should be able to change password securely

A use case diagram shown in Figure 3 below illustrates the interaction between the users (actors) and the system. It highlights the key functionalities that users can perform, such as login, creating swap requests, viewing matches, managing approvals (for admins).

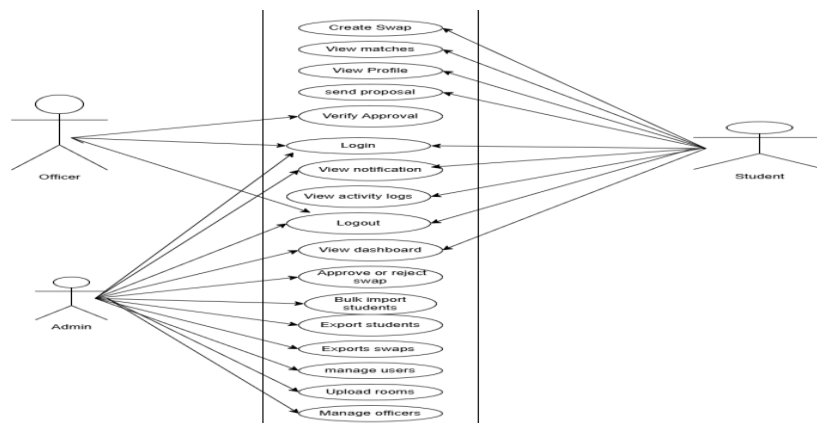


Figure 3: Use Case Diagram

Non-functional Requirements: The desired non-functional requirements for the proposed system are defined below:

- i. Security: The system must have role-based access control, encrypt user passwords, and restrict unauthorized access to sensitive data
- ii. Performance: The system shall respond to user requests within a few seconds and handle multiple users concurrently without performance degradation
- iii. Reliability: The system shall be available 24/7, with mechanisms to prevent data loss and ensure consistent operation
- iv. Usability: The system shall have a user-friendly interface, allowing users to perform tasks easily with minimal guidance.
- v. Maintainability: The system shall be designed to allow easy updates, modification, and debugging without major downtime.
- vi. Scalability: The system shall support future growth, accommodating additional users, hostels, or rooms without redesigning the architecture

Design of Proposed System

This section presents the key designs of the proposed hostel room swap management system. It explains how the system is structured, how users interact with it, and how different components work together to achieve the system objectives. The system design provides a clear blueprint of the proposed solution using activity diagrams and class diagrams. These diagrams help to illustrate the system workflow, user interactions, and processing logic in a clear and understandable manner.

Description of Proposed System

The proposed Hostel Room Swap Management System is a web-based application built using Flask framework. Students can log in, create swap requests specifying their current room and desired room. The system automatically matches swap requests based on preferences (students wanting each other's rooms), students can view matches and send proposals. Administrators review all swap requests, can see linked proposals, and approve or reject them. The system sends notifications to users about status changes and maintains

activity logs for audit purposes. The system supports bulk data import/export through CSV files for efficient management. Officers and admin can verify whether the approval is given or not.

Activity Diagram

We present the activity diagram for the two key actors namely, Hostel admin and Student.

Hostel Admin Activity Diagram

The hostel admin activity diagram illustrates the sequence of actions performed by the administrator in the system. It

includes activities such as logging into the system, managing users and officers, uploading students, approving or rejecting swap, verifying approval, exporting students, exporting swaps. Figure 4 below shows the hostel admin activity diagram

Student Activity Diagram

The student activity diagram illustrates the sequence of actions performed by the student in the system. It includes activities such as logging in, creating swap requests, sending proposals, viewing swap status. Figure 5 below shows the student activity diagram

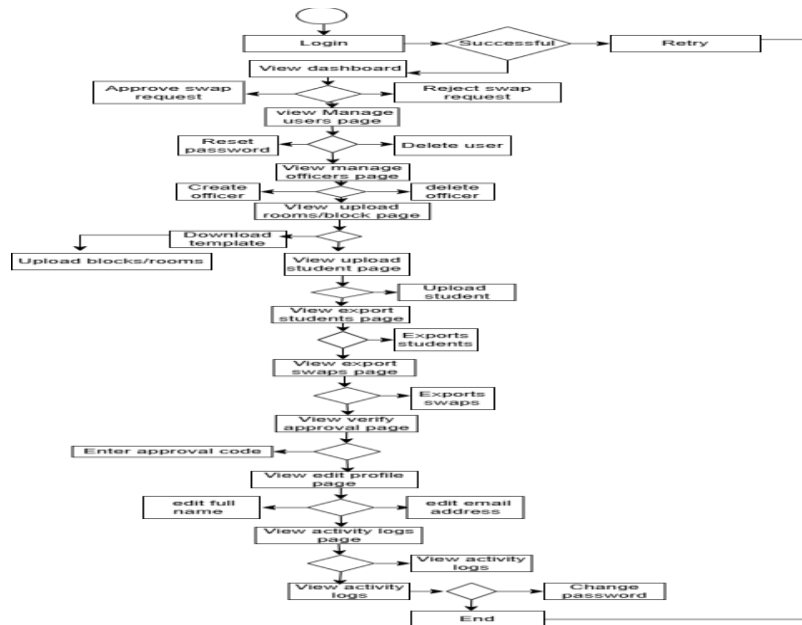


Figure 4: Hostel Admin Activity Diagram

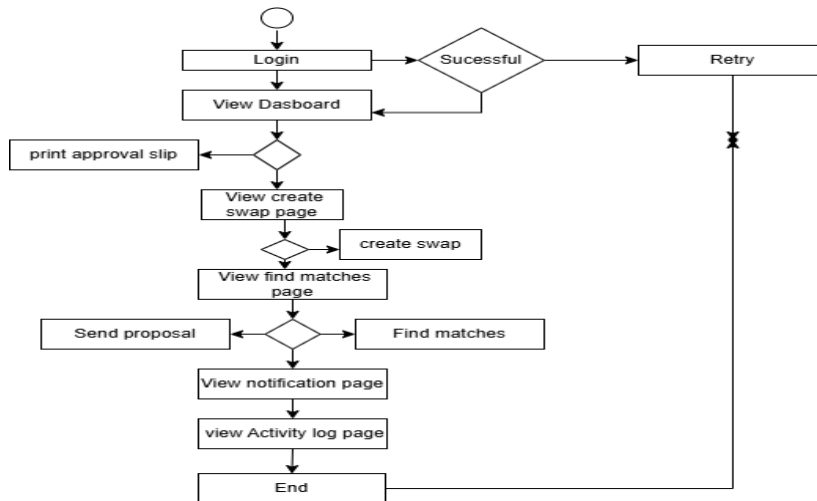


Figure 5: Student Activity Diagram

Class Diagram

The Class Diagram shown in Figure 6 illustrates the structure of the hostel room swap application structure in terms of the classes or objects and their relationships. Each class represents an entity in the system, such as a user, swap request, notification, or activity log. The diagram shows the attributes and methods of major classes, as well as how they interact with one another.

Entity Relationship (ER) Diagram

The Entity-Relationship Diagram shown in Figure 7 illustrates the database structure for the hostel room swap application. The diagram shows how different entities are related. This is crucial for database design as it ensures data integrity and efficient data retrieval.

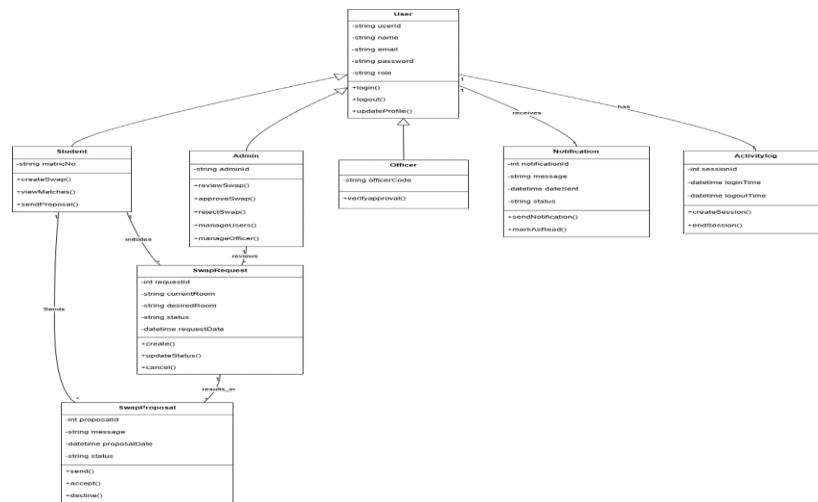


Figure 6: Class diagram

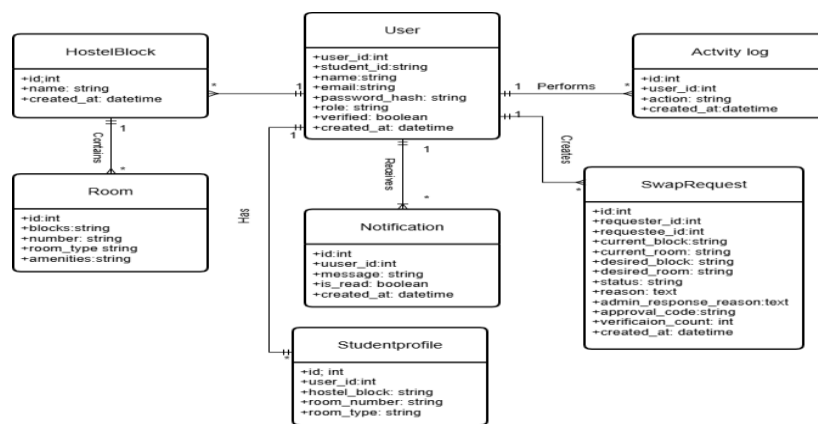


Figure 7: Entity Relationship Diagram

Implementation

This section focuses on the implementation phase of the Hostel Room Swap Management System. The implementation phase involves transforming the system design into a working web application using Python Flask framework and SQLAlchemy ORM. The development process adheres to the requirements and design models

presented in the previous sections, ensuring that the system meets its functional and non-functional goals.

Implementation tools

To implement the system, the following tools listed in table 1 below were used:

Table 1: Implementation tools

Category	Software Tools	Purpose
Programming Language	Python 3.11+	Used for backend development.
Database	SQLite, MySQL and SQLAlchemy ORM 3.0+:	SQLite is used for development and testing while MySQL is used for production deployment.
Frontend Technologies	Bootstrap 5, Tailwind CSS, Font Awesome Icons, Sweet Alert 2	For designing the layout and ensuring a responsive, user-friendly front-end experience.
Development Environment	Visual Studio Code	Primary IDE used for coding, with extensions for Python, and Flask.
Backend Framework	Flask 2.1+	Used as the python framework for backend development
Version Control	Git	Version control system for managing codebase changes and enabling effective team collaboration.

Description of System Operation

The system operation includes various functionalities to support hostel admins and students. Screenshots illustrating system operations are presented and described below in figures 8-13:

Student Dashboard

The student dashboard is the central hub for students after logging in. It displays their swap requests, pending proposals, recent notifications, and provides quick access to create new swap requests or view matches. Screenshot is shown in Figure 8.

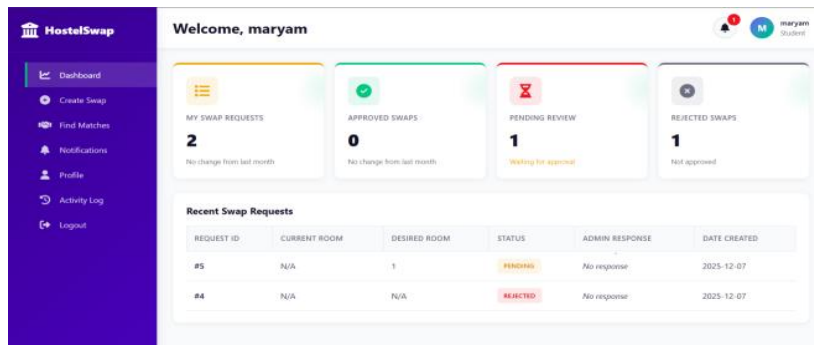


Figure 8: Student Dashboard

Create Swap Request Interface

This interface allows students to create new swap requests by specifying their desired block and room number. The system

automatically retrieves the student's current room information from their profile. Screenshot is shown in Figure 9

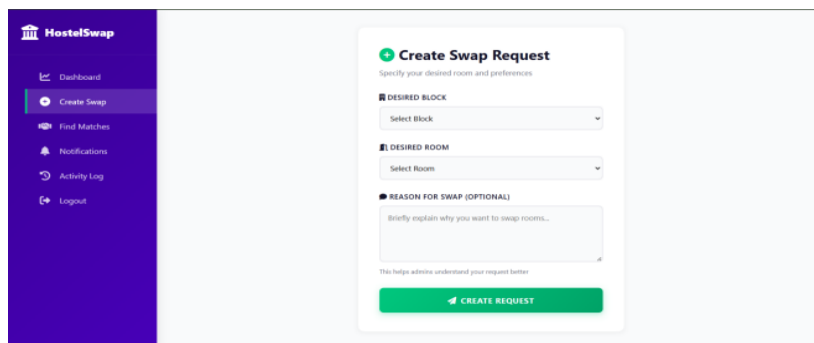


Figure 9: Create Swap Request Interface

View Matches Interface

This allows students to view matching swap requests from other students. The matching algorithm finds swaps where

other students want the current user's room or where their current room matches what the user wants. Screenshot is shown in Figure 10.

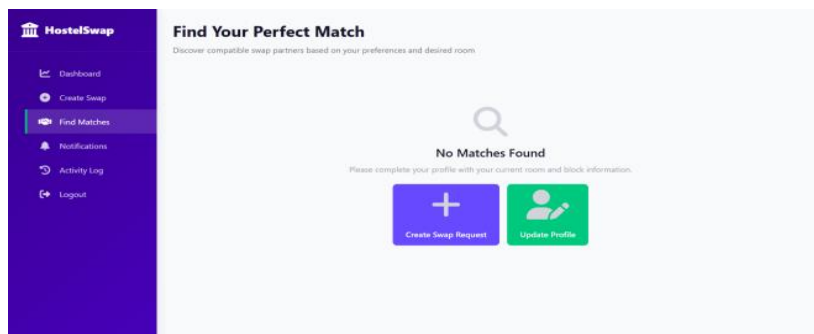


Figure 10: View Matches Interface

Admin Dashboard

The admin dashboard provides administrators with a comprehensive overview of the system, including statistics,

pending swap requests, and management tools. Screenshot is shown in Figure 11.

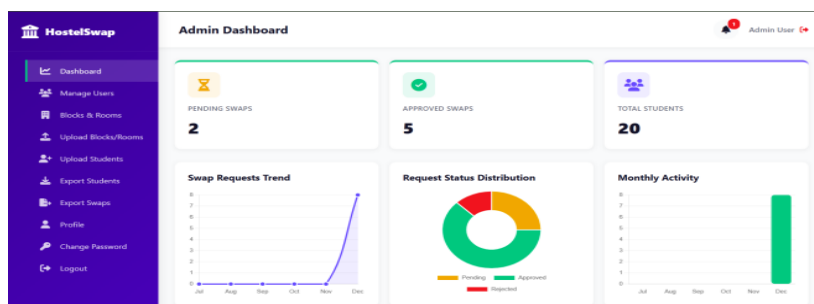


Figure 11: Admin Dashboard

Admin Approval/Rejection Interface

Admins can review swap requests in detail and approve or reject them with reason comments. The interface uses

SweetAlert2 for confirmation dialogs. Screenshot is shown in Figure 12.

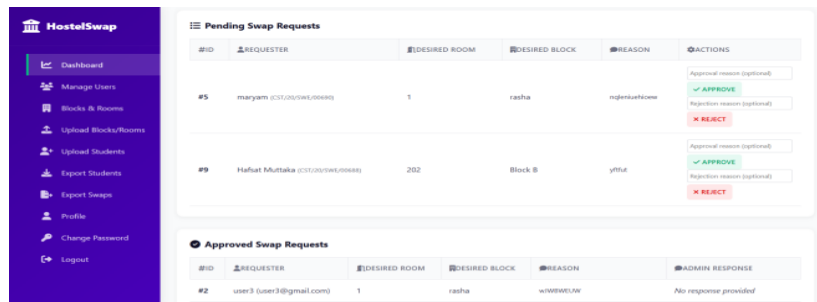


Figure 12: Admin Approval or Rejection Interface

Notifications Centre

The notifications centre displays all system notifications for the user, including swap status updates, new proposals, and system alerts. Screenshot is shown in Figure 13.



Figure 13: Notification Centre

RESULTS AND DISCUSSION

This section presents results of the usability testing which was conducted to verify the system's functionalities, performance, and user experience.

Results of Usability Testing

The usability evaluation was carried out using structured questionnaire. The participants consist of hostel administrators and students who are the primary users of the

system. Each participant was allowed to use the system and then asked to respond to the usability questionnaire based on their experience. The questionnaire consists of 8 questions that are focused on key usability attributes such as ease of use, interface clarity, system responsiveness, task completion, and overall user satisfaction. Results of the questionnaire are summarised in the following tables, and for each table visualisation is provided using bar chart.

Table 2: Q1: What Is Your Role In The System?

S/N	Responses	Score	Percentage
1	Admin	6	19.40
2	Officer	2	6.50
3	Student	23	74.10
	Total	31	100%

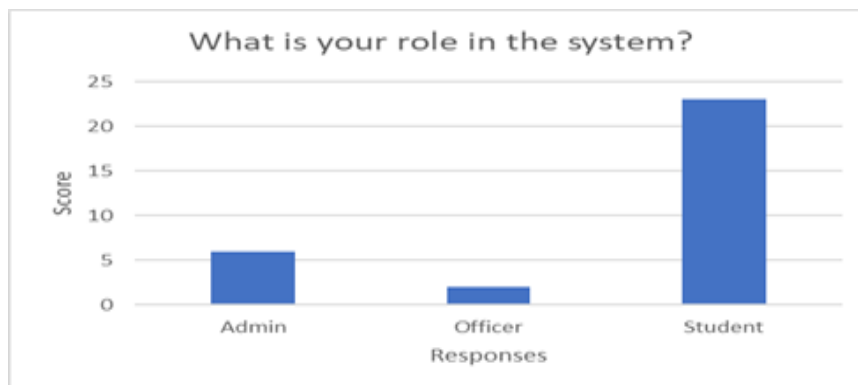


Figure 14: Result of Q1

Table 3; Q2: How Easy Was It To Navigate The System?

S/N	Responses	Score	Percentage
1	Very easy	9	29.00
2	Easy	17	54.90
3	Neutral	4	12.90
4	Difficult	0	0.00
5	Very difficult	1	3.20
	Total	31	100%

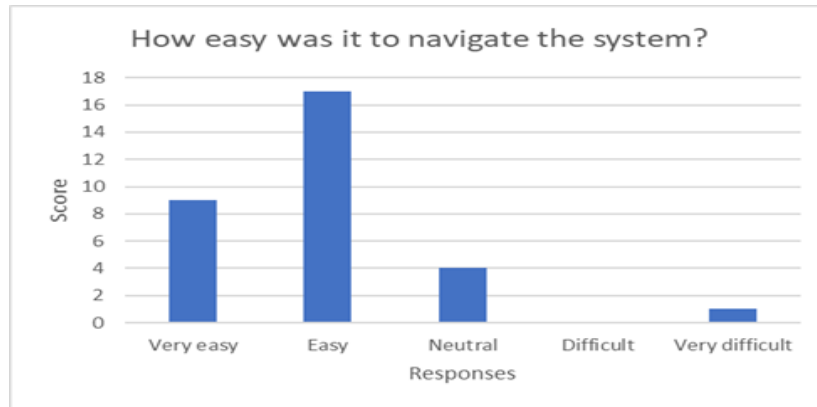


Figure 15: Result of Q2

Table 4; Q3: Is the Interface Clear and Understandable?

S/N	Responses	Score	Percentage
1	Very Clear	15	48.40
2	Clear	13	41.90
3	Neutral	2	6.50
4	Unclear	1	3.20
5	Very unclear	0	0.00
	Total	31	100%

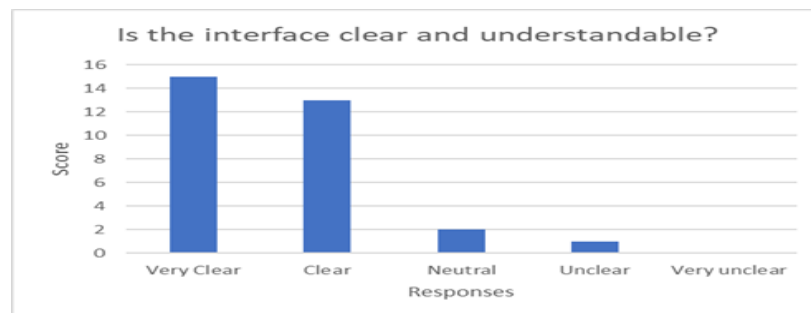


Figure 16: Result of Q3

Table 5; Q4: How Easy Was It to Learn How to Use the System For The First Time?

S/N	Responses	Score	Percentage
1	Very easy	13	43.40
2	Easy	14	46.70
3	Neutral	2	6.60
4	Difficult	1	3.30
5	Very difficult	0	0.00
	Total	30	100%

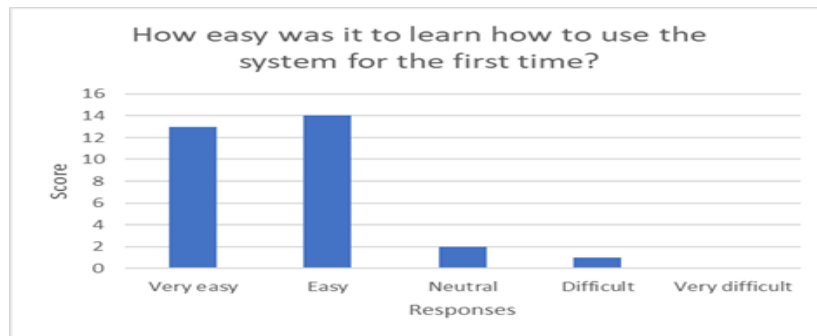


Figure 17: Result of Q4

Table 6: Can You Complete The Tasks Without Any Assistance?

S/N	Responses	Score	Percentage
1	Very easily	18	56.25
2	Easily	8	25.00
3	Neutral	2	6.25
4	With difficulty	2	6.25
5	Not at all	2	6.25
	Total	32	100%

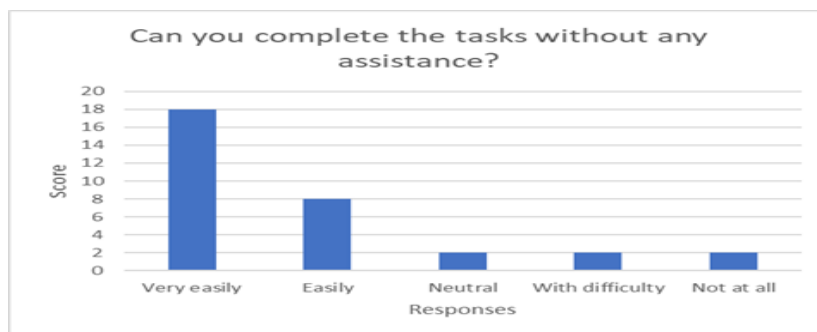


Figure 18: Result of Q5

Table 7; Q6: How Is The System Response?

S/N	Responses	Score	Percentage
1	Very Fast	16	51.60
2	Fast	10	32.30
3	Average	5	16.10
4	Slow	0	0.00
5	Very slow	0	0.00
	Total	31	100%

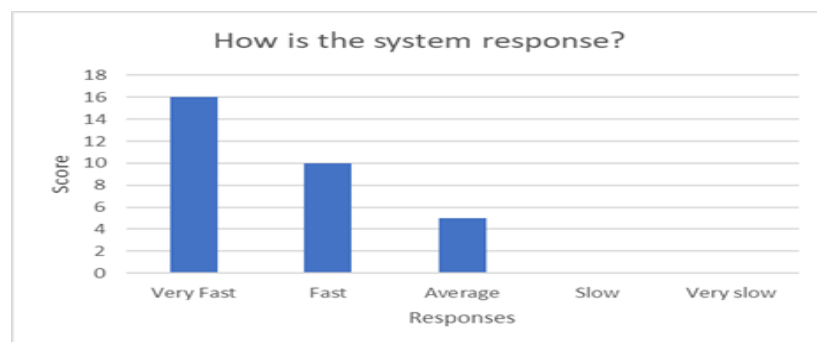


Figure 19: Result of Q6

Table 8: Q7: Overall, How Satisfied Are You With The System?

S/N	Responses	Score	Percentage
1	Very dissatisfied	0	0.00
2	Dissatisfied	3	9.70

S/N	Responses	Score	Percentage
3	Neutral	1	3.20
4	Satisfied	7	22.60
5	Very satisfied	20	64.50
	Total	31	100%

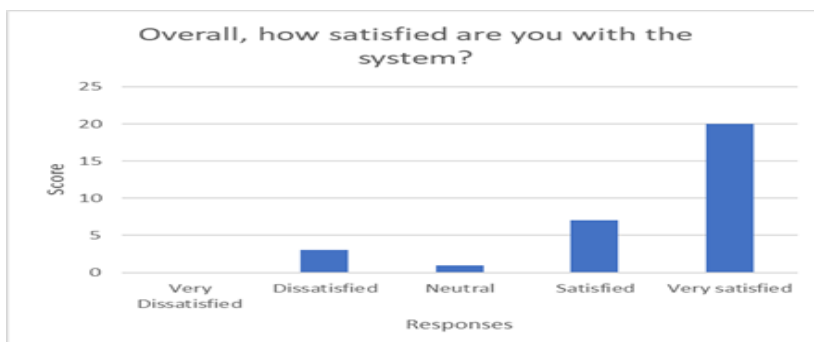


Figure 20: Result of Q7

Table 9: How Likely Are You To Recommend This System To Other Students?

S/N	Responses	Score	Percentage
1	Very Likely	22	71.00
2	Likely	6	19.40
3	I do not know	1	3.20
4	Unlikely	2	6.40
5	Very Unlikely	0	0.00
	Total	31	100%

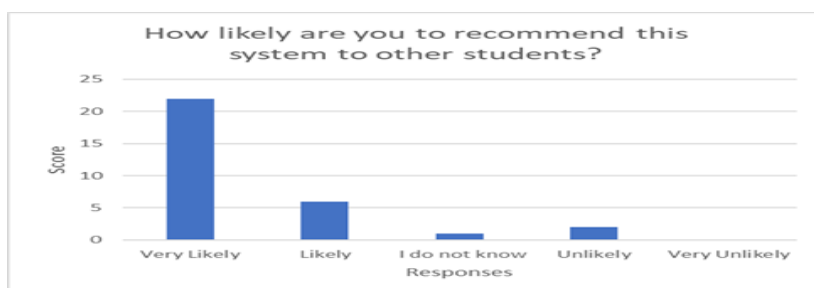


Figure 21: Result of Q8

Discussion

Usability testing was carried out using 31 anonymous users consisting of 23 students and 8 hostel admin. The users conducted thorough testing and filled the usability testing questionnaire. Overall, the results demonstrate a strong positive reception of the platform across all evaluated dimensions. Regarding navigation, 29% of participants found the platform very easy to navigate and 54.8% found it easy to navigate, with only 3.2% reporting difficulty in navigation. This indicates that the interface design is accessible and well-structured for both user roles.

Similarly, 48.4% of respondents confirmed that the system features are very clear to understand, and 41.9% reported that the system features are clear to understand, thus confirming overall clarity and understandability of system features. In terms of system learnability, 41.9% of users rated the system as very easy to learn and use, while 45.2% considered the system as easy to learn. This reflects the success of the user-centred design approach adopted during development. Performance in terms of system responses also received strong ratings, with 51.6% rating the system as very fast and 32.3% as fast, affirming acceptable system performance. Overall satisfaction was notably high, with 64.5% of participants reporting they were very satisfied and 22.6%

satisfied with their experience. Additionally, 90.4% indicated their willingness to recommend the platform to others, underscoring its potential for wider adoption.

Analysis of the responses shows that most participants found the system easy to navigate and simple to understand. The majority of the users rated the system interface as clear, indicating that the layout and design were user-friendly. Most participants responded that they were able to complete tasks without assistance, which shows that the system is easy to learn. The system response time was rated fast or very fast by most users, indicating good system performance. Furthermore, the overall satisfaction level of users was high, with many participants indicating that they were satisfied or very satisfied with the system. A large number of users also expressed a strong likelihood of recommending the system to others.

CONCLUSION

This study addressed the problem of dissatisfaction among students with their allocated hostel rooms due to incompatibility, location, and personal preferences. The Hostel Room Swap System was developed as a web-based solution to enable students to request and exchange hostel rooms in a convenient and transparent manner. An

appropriate software development methodology was adopted, covering requirement analysis, system design, implementation, and testing. The system successfully implemented core functionalities such as user authentication, room swap request submission, viewing available swap options, and interaction between users. The results obtained demonstrate that the system effectively simplifies the hostel room swap process and enhances student satisfaction. Future improvements should include intelligent swap matching, automated notifications, and interaction with the hostel management system to enhance efficiency.

REFERENCES

- Adegbola, A. A., Adekola, O. A., Ebitigha, O. F., Nwabueze, U. F., Imade, U. D., & Adebajo, A. (2024). Design and implementation of a student hostel management system. *Babcock University College of Postgraduate Studies Journals*, 3(1), 49-58. <https://cpgsjournals.babcock.edu.ng/journals/CTICTR/document?aid=43606026>
- Anis, N. F., & Mohd Safar, N. Z. (2022). Web based hostel management system for Sekolah Menengah Agama Parit Sulong. *Applied Information Technology and Computer Science*, 3(1), 279-291. <https://doi.org/10.30880/aitcs.2022.03.01.018>
- Bamidele, A. O., & Yusuf, T. A. (2024). Design and implementation of web-based hostel management systems in Nigerian universities. *International Journal of Computer Applications*, 186(12), 25–32.
- Bhardwaj, S., Gupta, P., & Sharma, M. (2022). AI-IoT system to automate dormitory environments: A review. *International Journal of Computer Applications*, 184(12), 40-45. <https://www.researchgate.net/publication/368057802>
- Bista, R. K., & Karki, A. J. (2021). Hostel management system: Study of digitalized systems and database integration. *Journal of Science and Engineering*, 8(2), 112-119. <https://www.semanticscholar.org/paper/91c78ca433867f139c3dd3acebc481ce936befb0>
- Diyaolu, O. A., & Muftau, A. K. (2024). Development of an E-based hostel management system. *International Journal of Innovative Science and Research Technology*, 9(6), 1000-1008. <https://doi.org/10.38124/ijisrt/IJISRT24JUN147>
- Eweoya, I. O., Awoniyi, A., Adeniyi, O., Okesola, K., Udosen, A., Adigun, T., & Fatade, O. (2025). Development of web-based hostel management system. *British Journal of Computer, Networking and Information Technology*, 8(1), 40-52. <https://www.researchgate.net/publication/389336918>
- Eze, S. C., & Abdullahi, M. S. (2023). Digital transformation and smart campus initiatives in higher education institutions. *Journal of Information Technology in Education*, 18(2), 44–58.
- Ogunyinka, O. I., & Hassan, R. A. (2022). Adoption of web-based information systems in tertiary institutions. *Nigerian Journal of Computing and Informatics*, 14(1), 60–71.
- Ojetola, A. A., & Gbadegesin, K. O. (2021). Student accommodation and its impact on academic performance in tertiary institutions. *Journal of Educational Management*, 9(3), 112–120.
- Okeke, C. N., & Musa, I. A. (2023). Challenges of hostel accommodation management in Nigerian universities. *African Journal of Educational Technology*, 7(1), 89–98.
- Oloyede, A. A., Adekunle, O. A., & Lawal, S. A. (2023). Efficiency of digital platforms in off-campus hostel records. *Nigerian Journal of Technological Development*, 20(3), 145–152. <https://www.researchgate.net/publication/381783262>
- Pawar, A., & Shaikh, J. (2025). Hostel management system: Web application and database architecture. *International Journal of Multidisciplinary Research in Science, Engineering and Technology*, 8(4), 5161-5168. <https://ijarce.com/wp-content/uploads/2025/06/IJARCE.2025.14634.pdf>
- Salami, A. T., & Obasi, F. U. (2023). Equity and transparency in student accommodation allocation systems. *Journal of Social and Management Sciences*, 15(4), 101–110.
- Ukpere, P. O., & Ibekwe, C. (2023). Integrated role-based HMS for efficiency and transparency in university accommodation. *Journal of Digital Innovation and Computing*, 4(2), 88-97. <https://www.researchgate.net/publication/398910719>
- Waibi, Z. (2022). *Design and implementation of a web-based hostel management system* [Thesis, ResearchGate]. <https://www.researchgate.net/publication/383260643>

