KIGALI INDEPENDENT UNIVERSITY (ULK)

SCHOOL OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

P.0 BOX 2280 KIGALI

“Development and Implementation of a Web-based Hostel Management System:

Kigali Independent University ULK as Case Study.”

Name: Henry B. Benson Jr.

Roll Number: 202110741

Tel: +250790005474

Supervise by: DR. MUSABE J. BOSCO

A DISSERTATION SUBMITTED TO THE KIGALI INDEPENDENT UNIVERSITY, SCHOOL OF SCIENCE AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A BACHELOR OF SCIENCE IN COMPUTER SCIENCE.

SEPTEMBER 2024

DECLARATION

I, Henry B. Benson Jr, hereby declare that this work entitled ***"Development and Implementation of a Web-based Hostel Management System: Kigali Independent University ULK as Case Study."*** submitted in partial fulfillment of the requirement for the award of a bachelor's degree in computer science, this is my work and has not been presented for another university.

Student Name…………………………………………………………………...

Date………………………………………………………………………….

Signature…………………………………………………………………….

APPROVAL

This dissertation, entitled “Development and Implementation of a Web-based Hostel Management System: Kigali Independent University ULK as Case Study, " was written under my supervision and submitted for examination with my approval.

Supervisor Name: ……………………………………………….

Date: ………. /……………./…………………………….

Signature: ……………………………………………

DEDICATION

I dedicate this thesis to my beloved and supportive parents, **Mr. Henry B. Benson Sr.** and **Mrs. Josephine L. Benson**, whose unwavering encouragement has been my foundation. To my sister, **Wedee I. Thompson**, and my brother, **Jeremie A.S. Benson**, thank you for your constant support and belief in me. I also extend my heartfelt gratitude to all my friends and family members who have stood by me throughout my years of study. Your love and support have been invaluable, and I am deeply grateful for each of you.

Henry B. Benson Jr

ACKNOWLEDGMENT

First, I am deeply grateful to God for His unwavering guidance, strength, and blessings throughout this journey. Without His grace, this achievement would not have been possible.

I am profoundly thankful to my parents, Mr. Henry B. Benson Sr. and Mrs. Josephine L. Benson, for their endless support, encouragement, and love. Your belief in me has been my greatest motivation, and I dedicate this work to you.

I thank the President of Kigali Independent University for his leadership and vision, which have fostered an environment conducive to academic excellence and personal growth.

My heartfelt appreciation goes to my supervisor, Dr. Musabe, for his invaluable guidance, insightful feedback, and constant support. Your expertise and dedication have been instrumental in completing this thesis.

I would also like to acknowledge my girlfriend, Quita Farnagalo, for her unwavering support, patience, and love. Your encouragement has been a source of strength and inspiration throughout this journey.

I thank my classmates for their camaraderie, collaboration, and encouragement. Your support has made this journey more enjoyable and enriching.

Lastly, I am grateful to all my friends and family members who have supported me. Your love, patience, and understanding have been a blessing, and I am deeply grateful for each of you.

Henry B. Benson Jr.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATION AND ACRONYM

API: Application Programming Interface

CSS: Cascading Style Sheet

DBMS: Database Management System

DFD: Data Flow Diagram

ERD: Entity Relationship Diagram

GUI: Graphical User Interface

HMS: Hostel Management System

HTML: Hyper Text Markup Language

IoT: Internet of Things

JS: JavaScript

LCD: Liquid Crystal Display

MYSQL: My Structured Query Language

PDF: Portable Document Format

PHP: HyperText Processor

RFID: Radio-Frequency Identification

RUP: Rational Unified Process

SRS: Software Requirements Specifications

ULK: Kigali Independent University

UML: Unified Modelling Language

UX/UI: User Experience/Interface

XAMPP: Cross-Platform (X), Apache (A), MariaDB (M), PHP (P), and Perl(P)

ABSTRACT

This thesis describes developing and implementing a comprehensive web-based hostel management system designed specifically for Kigali Independent University. The system optimizes critical hostel operations, including room allocation, booking, lodging, complaint management, and user feedback collection. A key feature is integrating a Radio Frequency Identification attendance system, which tracks student movements in and out of the hostel.

This RFID-based solution enhances security by providing real-time monitoring and accurate student attendance records, reducing the potential for manual errors and unauthorized access.

The Hostel Management System was designed to streamline and improve the overall management and operations of the university's hostel facilities, contributing to a more efficient and secure student living environment.

The HMS was developed using modern web technologies, including responsive design, intuitive user interfaces created with HTML, CSS, JavaScript, and PHP, and the robust MySQL database to ensure efficient data management and storage. The user-friendly interface enables seamless interaction between administrators and students, simplifying the management of a wide range of hostel-related activities, such as room reservations, maintenance requests, and incident reporting. The RFID attendance system is fully integrated with the main HMS, ensuring that movement data is automatically logged and easily accessed for administrative purposes, improving overall operational efficiency and providing valuable insights into student activity patterns within the hostel.

Rigorous testing was conducted to ensure the Hostel Management System's functionality, reliability, and user experience. The results revealed that the HMS significantly enhances hostel management efficiency, reduces administrative workload, and improves security. The RFID attendance tracking feature is handy for monitoring student movement, ensuring order and security. The testing process involved various scenarios and user interactions to guarantee a seamless and user-friendly experience.

The Hostel Management System at Kigali Independent University showcases the potential of web-based solutions for improving hostel management efficiency and user experience in resource-constrained settings.

Key Words: Hostel Management System, Web-based, RFID, Attendance Tracking, Student Housing, University, Kigali Independent University

CHAPTER ONE: GENERAL INTRODUCTION

1.1. Introduction to the Study

The advancement of technology has revolutionized various aspects of our lives, including the management and operation of educational institutions. In recent years, a growing emphasis has been on digitalizing administrative processes to streamline operations and enhance efficiency. One such area that has gathered significant attention is the management of university hostels. As the traditional methods of hostel management are increasingly inadequate in meeting the evolving needs of students and university administrators, the urgent and timely need for developing and implementing a web-based hostel management system has emerged as a viable solution.

This thesis offers a unique perspective on the complexities of developing and implementing a web-based hostel management system, focusing on its use at Kigali Independent University. By closely examining the university's experiences, challenges, and outcomes, we can derive crucial insights to inform best practices and recommendations for similar educational institutions. The significance of the Kigali Independent University case study in informing best practices and recommendations for similar educational institutions will keep the audience informed and knowledgeable about the topic.

The subsequent chapters will delve into the theoretical foundations, technical aspects, and practical implications of web-based hostel management systems, providing a comprehensive overview of their significance in modern educational environments. Furthermore, the Kigali Independent University case study will serve as a real-world illustration of the system's impact, offering valuable lessons for stakeholders in the education sector. Through this exploration, this thesis contributes to the existing knowledge on technology-enabled hostel management and offers practical guidelines for its successful implementation. The main problem in this research is the need for innovation in the development of a web-based hostel management information system, as the current manual system at KSMA Terpadu Krida Nusantara poses numerous challenges and limitations (Jafrudin & Putra, 2020). These challenges include time-consuming and error-prone student data management processes, difficulty retrieving information, and accumulating physical documents. The web-based hostel management system aims to address these issues by simplifying data management, information retrieval, and reporting processes. Additionally, the system will provide a user-friendly interface and efficient performance, eliminating the need for manual processes and reducing the burden on the university staff and students. The proposed web-based hostel management system will streamline registration, room allocation, student attendance, inventory management, fee collection, and report generation (Khamis et al., 2020). This will significantly improve the efficiency and accuracy of hostel management and enhance the overall student experience, fostering a sense of optimism about the future of hostel management.

In this chapter, we will look at the problems in hostel management, the goals and objectives of the proposed system, the methodology used to develop and implement the system, the project's scope, and the period for its completion. Additionally, we will examine best practices and recommendations for similar educational institutions based on the experiences and lessons learned from the case study of Kigali Independent University (Bista et al., 2018).

Overall, this thesis aims to address the challenges faced in hostel management by developing a web-based hostel management system (Jafrudin & Putra, 2020). Once implemented, the system will provide efficient and user-friendly management of hostel activities, significantly improving accuracy and reducing manual effort. Its potential benefits include streamlined processes, enhanced student experience, and improved overall efficiency.

In conclusion, developing and implementing a web-based hostel management system represents a significant step towards modernizing administrative processes and enhancing efficiency in educational institutions. The subsequent chapters will provide a deeper understanding of the intricacies involved and shed light on the potential impact of such a system, adding valuable contributions to the existing body of knowledge in technology-enabled hostel management.

1.2. Background of Study

The existing manual system for managing hostels at educational institutions has numerous drawbacks and limitations (Khamis et al., 2020). It is time-consuming, prone to errors, and inefficient. According to Bista et al. (2018), traditional methods of record-keeping and managing hostel facilities hurt the overall efficiency of educational institutions. Many institutions rely on manual systems that involve pen-and-paper record-keeping when managing student hostels. These inefficiencies highlight the pressing need for a more effective and modern solution.

This traditional approach is time-consuming and can lead to inefficiencies in the overall management process (Khamis et al., 2020).

A substantial portion of recently established educational institutions still utilize the old ordinary procedures for record-keeping, especially for managing hostel facilities. Manual systems for managing hostels in educational institutions have become outdated and inefficient. This outdated method of managing records negatively impacts institutions' efficiency and can result in errors and delays (Bista et al., 2018).

According to Khamis et al. (2020), the drawbacks of the existing manual system include the risk of losing important data or records, difficulty searching for specific information, and errors in calculations such as fees. Additionally, adding new records, updating existing records, deleting outdated information, and searching for specific details takes time and effort. Nevertheless, there is a clear need to develop and implement a modern system to optimize and automate hostel management processes.

This study evaluates and suggests a contemporary hostel management system that meets the needs of the university, its personnel, and students. The proposed system will address the limitations of the existing manual system by providing a more efficient and user-friendly solution. The suggested system addresses the difficulties encountered by hostel administrators, including giving reports, registration, and resident and room information retrieval. Furthermore, it will facilitate convenient data access, minimize calculation errors, and enhance overall operational effectiveness (Podunavac et al., 2019) (Rai, 2019).

The proposed system will develop a web-based hostel management system designed explicitly for Kigali Independent University as a case study. This system will provide a user-friendly GUI that automates and organizes all the procedures for managing hostel facilities (Khamis et al., 2020). With this system, universities, employees, and students can easily manage hostel activities more actively and efficiently. They can store and retrieve data electronically, eliminating the risk of losing valuable information.

1.3.Problem Statement

This dissertation addresses the issue of outdated and manual techniques in educational institutions and hostels, which lead to risks like data loss, difficulty in information search, and errors in calculations. It also highlights the lack of widespread use of web applications. (Meghana et al., 2021). Hostels and educational institutions often use manual data storage and management systems, leading to data loss and inefficient record-keeping. Kigali Independent University faces challenges in managing facilities effectively, necessitating a modern, automated system. The COVID-19 pandemic brings to life the importance of automation and digital transformation in managing hostel facilities. A web-based system is proposed to address these limitations by automating and organizing procedures related to hostel facilities. This system will enable efficient management of hostel activities, store and retrieve data electronically, and minimize data loss. It will also facilitate easy access to data, minimize calculation errors, and enhance operational effectiveness. The case study at Kigali Independent University will provide insights into the practical implications and benefits of implementing this system, bridging the gap between outdated manual systems and modern technological solutions.

1.4.Project Objective

The main objective of this project is to develop and implement a web-based hostel management system for Kigali Independent University. This system aims to streamline and automate the management of hostel facilities, including the application process, resident registration, room allocation, fee management, and reporting (Bista et al., 2018). The objective is to provide an efficient and user-friendly platform enabling hostel management to easily manage and access resident data, track applications, allocate rooms, and track student attendance. Additionally, the system will provide a platform for communication between the hostel administration and students, allowing them to access relevant information and updates more conveniently and timely.

1.4.1. General Objective

The main objective of this project, **"Development and Implementation of a Web-based Hostel Management System: ULK as Case Study,"** is to design and implement an efficient Hostel Management System.

1.4.2. Specific Objectives

i. To develop a hostel registration system for students and staff involved in Kigali Independent University (ULK) hostel activities.

ii. To design a user-friendly front end for a web-based HMS.

iii. To implement the web-based Hostel Management System

iv. To evaluate the report for hostel facilities with the help of IOT or Embedded Systems (using RFID Tags and Card Readers)

v. To visualize the report for the Hostel Management System.

1.5.Research Question

i. What are the essential features and functionalities required for a hostel registration system to effectively manage student and staff information within the ULK hostel?

ii. What are the fundamental design principles and best practices to ensure a user-friendly and intuitive front end for a web-based hostel management system?

iii. What are the most suitable technologies and programming languages for developing a robust and scalable web-based hostel management system?

iv. How can RFID tags and card readers be integrated into a hostel management system to enhance security, efficiency, and data collection for evaluating hostel facilities?

v. What are the most effective data visualization techniques for concisely and informally presenting the insights from the hostel management system's reports?

1.6.Scope of the Project

This project aims to develop and implement a web-based hostel management system for Kigali Independent University in Rwanda. The system will focus on managing the hostel facilities, including student registration for accommodation, room allocation, fee payment, and security.

1.6.1. Content Scope

The project's content scope encompasses all the functionalities and features for effective hostel management, including student data management, room allocation, fee payment tracking, attendance tracking, and reporting.

1.6.2. Geographical Scope

The project is geographically limited to Kigali Independent University (ULK) in Rwanda, including its main campus in Kigali, other campuses, and the surrounding areas within Kigali.

1.6.3. Time Scope

Considering all pertinent variables and unforeseen circumstances, the project will contain an extensive timeline for development, implementation, and any required post-implementation support and maintenance. The project will not extend to other universities or locations beyond Kigali Independent University (ULK).

1.7.Significance Of the Project

There are assorted reasons why this project, "Development and Implementation of a Web-based Hostel Management System: Kigali Independent University s(ULK) as a Case Study," is essential. It first tackles the requirement for a dependable and effective system for managing dorms at academic establishments such as Kigali Independent University. Kigali Independent University's hostel administration system is currently manual, which causes several issues, including document accumulation, difficulties finding student information, and inconsistent data management. These issues can be resolved, and the general effectiveness of the hostel administration procedure can be raised by creating and deploying a web-based system.

1.7.1.Personal interest

I am interested in this project because it will allow me to use my experience and abilities in database management and web development. My passion has constantly been developing effective and user-friendly solutions, and this project allows me to achieve just that. The university will also benefit from installing a web-based hostel management system since it simplifies administrative procedures and reduces manual labor.

1.7.2. Institutional interest

Furthermore, this project also has more considerable institutional interest. Kigali Independent University will benefit from implementing a web-based hostel management system as it will simplify and automate the processes involved in managing the hostel, improving efficiency and effectiveness. Moreover, the public will also benefit from this project.

1.8.3. Public interest

The public interest lies in a web-based hostel management system providing enhanced convenience and accessibility to students and parents. They can easily access and update their hostel-related information, such as room assignments, fees, and maintenance requests, from anywhere at any time. This will save them time and effort and improve their overall experience with the university's hostel facilities.

1.8. Limitation of the Project

As the project unfolded, several obstacles surfaced that could potentially impede its successful implementation.

Coordinating with the supervisor and securing prompt feedback and guidance proved a primary challenge. Effective communication and collaboration are vital for any project, and the lack of timely response or clear direction can lead to delays and misunderstandings. Additionally, sourcing the specific components and equipment necessary for the RFID attendance system development was demanding, necessitating extensive research and effort. The availability and compatibility of these components can significantly impact the project's timeline and budget.

Furthermore, as is common in software development projects, technical obstacles and setbacks were encountered. Issues such as debugging code, compatibility problems between software components, and unexpected errors are prevalent. These technical hurdles can consume valuable time and resources, potentially delaying the project's completion.

1.9.Project Methodology

The project used a multifaceted approach to ensure the successful development of the web-based hostel management system. This included thorough data collection, such as reviewing relevant documents and observing similar hostel management projects, to gain insights and best practices. Additionally, the project team strategically employed the agile system development life cycle model. This provided a structured and iterative framework for analyzing, designing, and developing the hostel management system. The agile approach allowed for effective responses to changing requirements, collaboration closely with stakeholders, and system delivery in a gradual and adaptable manner.

1.10. Organization of the project

Chapter 1: General Introduction

This chapter will offer a comprehensive project overview, including its background, objectives, scope, and significance. It will explore the motivation behind developing a web-based hostel management system for Kigali Independent University and highlight the potential benefits to its hostel operations.

Chapter 2: Literature Review

This chapter will comprehensively review the literature on hostel management systems, web-based applications, RFID technology, and data management. It will examine the research findings, methodologies, and established best practices in these domains to establish a robust foundation for the project.

Chapter 3: System Analysis and Design

This chapter will provide a detailed analysis and design of the web-based hostel management system. It will identify the system's requirements and outline use case diagrams, entity-relationship diagrams, and data flow diagrams to illustrate the system's architecture and functionality.

Chapter 4: System Implementation and Development

This chapter describes the system's implementation and development. It covers the selection of suitable technologies, programming languages, and tools. It also outlines the coding, database design, user interface development, and testing processes undertaken to ensure the system's functionality and optimal performance.

Chapter 5: Conclusion and Recommendation

This concluding chapter will synthesize the key insights and accomplishments derived from the project, emphasizing the successful deployment of the web-based hostel management system. Additionally, it will offer recommendations for future enhancements, potential areas of improvement, and directions for further scholarly investigation.

CHAPTER TWO: LITERATURE REVIEW

2.1. Introduction

The growing digitalization of administrative duties within educational institutions has prompted the development of diverse management systems designed to enhance efficiency and service quality. Hostel management systems are among these innovations, addressing the specific requirements of managing student accommodations. A web-based hostel management system integrates various functionalities, such as room allocation, fee management, and maintenance requests, into a single platform accessible through the Internet. This literature review investigates such web-based hostel management systems' recent advancements, methodologies, and implementations.

2.2. Conceptual Review

The thesis aims to create a web-based Hostel Management System (HMS) for Kigali Independent University (ULK). This system will streamline hostel-related processes, replacing the existing manual procedures. Let us break down the key concepts and constructs:

1. **Hostel:** A hostel is a student housing facility at a university or boarding school. In addition to housing students in shared rooms, it might have other facilities like social halls, dining areas, and communal kitchens. Students can live conveniently and affordably during their academic years at university hostels (Simpeh & Shakantu, 2019).
2. **Management:** Technology management is a set of concepts, skills, techniques, and practices that result in decision-making and implementation regarding the development and use of technology by firms. It ultimately aims to succeed in innovation and increase the firm's competitiveness. (Robert, E L M, 2020)
3. **System:** In the context of this study, a system refers to a cohesive hardware and software unit that works together to perform specific tasks. (Dictionary.com)
4. **Web-Based System:** A web-based system is any site or software application housed on a web server and available through a web browser. These systems provide cross-platform compatibility, ease of access, and centralized data management. Their design emphasizes scalability, reliability, and user-friendliness, making them ideal for managing data-intensive operations like hostel management. (Best Student Hostel Management System Software India - A.T.S.I., 2018).
5. **Hostel Management System:** This refers to a software solution for managing hostel operations, including room allocation, student registration, fee processing, and maintenance

requests. Such systems aim to streamline administrative processes, improve operational efficiency, and enhance user experience for staff and residents (Saxena & Khandelwal, 2010).

6. **Web-Based Hostel Management System:** A Web-Based Hostel Management System is a software application designed to manage the various administrative and operational tasks involved in running a hostel. It is available through a web browser, empowering users to interact with the system from any gadget with internet access. Key functionalities typically include:

i) **Room Allocation and Management:** Systems automate room allocations based on predefined criteria, reducing manual errors and ensuring efficient space use.

ii) **Fee Collection and Tracking:** Integrated billing modules track payment status, send reminders, and manage financial records.

iii) **Maintenance and Complaint Handling:** Platforms often include modules for logging maintenance requests and tracking their resolution.

iv) **Student Information Management:** Comprehensive databases store residents' personal and academic details.

v) **Reporting and Analytics:** Systems generate reports on occupancy, finances, and maintenance activities for administrative review.

These systems are designed to improve efficiency, reduce manual errors, and provide centralized control over hostel operations. By leveraging web technologies, they offer ease of access and real-time data updates, enhancing administrators' and residents' overall management experience (Best Student Hostel Management System Software India - A.T.S.I., 2018).

Developing and implementing a web-based hostel management system at Kigali Independent University is an example of how contemporary online technologies can be integrated to improve hostel administration's user experience and operational effectiveness. With an emphasis on user experience, system design, implementation issues, and impact assessment, this research provides thorough insights into the theoretical and practical elements of implementing such systems in educational institutions.

2.3. Theoretical Review

As we delve deeper into the theoretical background, it is critically important to thoroughly examine and analyze the fundamental concepts that play a significant role in shaping and guiding the development and implementation of web-based systems for managing hostels.

The theoretical underpinning of this research will revolve around exploring internet technologies, database management, user interface design, recent developments, and ensuring robust system security within hostel management contexts.

Recent developments in hostel management systems around the world include the following advancements:

- **Cloud-Based Solutions:** The migration towards cloud computing technologies has enabled the creation of scalable and adaptable hostel management systems. Cloud-hosted hostel management platforms can accommodate substantial data volumes and provide real-time information access from any geographical location. (Chaudhri & Kevat, 2021)

- **Mobile Integration:** The widespread adoption of smartphones has led many hostel management systems to develop mobile applications, enabling students and administrators to access services remotely. These mobile apps enhance the user experience by providing real-time notifications and intuitive interfaces, improving accessibility and convenience for all stakeholders (Doe et al., 2023).(Karki & Bista, 2018).

- **Artificial Intelligence and Machine Learning:** Artificial intelligence and machine learning technologies are increasingly integrated into hostel management systems to enhance their capabilities and efficiency. These advanced technologies enable HMS to predict and proactively address maintenance needs, optimize student room allocation based on various factors, and analyze student behavior patterns to improve service delivery. By leveraging AI and ML, HMS can become more responsive, adaptable, and data-driven, streamlining operations and enhancing the overall experience for both hostel administrators and student residents. Integrating these cutting-edge technologies into HMS significantly advances university housing management, increasing efficiency, cost savings, and user satisfaction (Sykimte, 2023).

- **Security and Privacy Enhancements:** Contemporary hostel management systems prioritize implementing comprehensive security measures to mitigate the increasing risk of data breaches. These measures include using encryption techniques, multi-factor authentication protocols, and compliance with data protection regulations to safeguard the confidentiality and integrity of sensitive information (Chaudhri & Kevat, 2021).

Web technologies like HTML, CSS, JavaScript, PHP, and Python make web-based hostel management systems accessible and compatible across different platforms.

These languages help to design user-friendly interfaces and ensure smooth system interaction. Additionally, they contribute to the system's overall functionality by enabling dynamic content generation and efficient data processing (User Interface, 2023).

Database administration is crucial to the hostel management system's efficient functioning. It enables smooth storage, access, and processing of information related to student enrolment, room assignment, and maintenance needs. A solid database management system ensures dependability and expandability, effectively managing increasing data volumes and user activities. This provides a solid foundation for seamless operations within the environment (Hostel Management System, 2019).

User interface design is essential for promoting a user-friendly experience for administrators and users. To significantly enhance the user's experience, it should primarily focus on simple and intuitive navigation, clear presentation of information, accessibility across different devices, and overall system usability.

Ensuring the security of hostel-related data is essential for protecting its confidentiality and integrity. It involves implementing robust authentication methods, such as two-factor or biometric identification, efficient data encryption using advanced cryptographic algorithms, strong access control measures, including role-based access controls, and regular security audits to reduce unauthorized entry and address potential security risks.

Furthermore, exploring the theoretical underpinnings of web-based hostel management systems necessitates examining their impact on operational efficiency, cost-effectiveness, and user satisfaction. It involves understanding the theoretical frameworks and empirical evidence surrounding adopting and successfully implementing similar systems in academic institutions, thereby providing insights into their efficacy and potential challenges.

2.4. Review of Related Literature

Developing and implementing a web-based hostel management system tailored for Kigali Independent University is a significant endeavor that promises to revolutionize student accommodation management. As documented in the literature, various strategies and technologies enhance these systems' functionality and user experience, offering a hopeful outlook for the future.

Emphasize the benefits of using hybridized methodologies, such as the Random Forest algorithm and Long Short-Term Memory (LSTM), in managing hostel activities. This strategy showcases the potential of machine learning in automating and streamlining procedures, thereby reducing physical work and enhancing the system's ability to handle multiple users simultaneously, which is crucial for university hostels.

Another study focuses on creating a user-friendly and GUI-oriented innovative hostel management system. This system, which manages mess bills, student information, and payment records, aims to address the shortcomings of traditional methods by providing a centralized platform for overseeing hostel operations and enhancing institutional efficiency (Mahendra et al., 2022).

A thorough examination of online hotel management systems can also offer valuable insights into developing a hostel management system. Both have similar goals and face similar challenges, so it is essential to comprehend the primary features, benefits, and hurdles linked with these systems by consolidating recent studies. This will contribute to a more profound comprehension of the topic. By analyzing the relevant literature, it becomes clear that developing and implementing a web-based hostel management system has numerous benefits (Singh et al.,2023).

Lastly, a case study on implementing a hostel management system at Lagos State University (HOMASY) offers a practical perspective. The study details the computerized process that is stress-free, reliable, and quick, using PHP and MySQL for backend operations (O. Shoewu1, S.A. Braimah & O. Duduyemi). This case study can serve as a valuable reference for the Kigali Independent University project, providing a blueprint for successful implementation.

The case studies concentrate on different strategies for streamlining and simplifying the operations of a university hostel, like reducing physical labor and creating a stress-free, dependable environment for hostel administration. None of them address the issue of monitoring student attendance, which would give a precise account of when a student arrives or departs from their dorm. We plan to provide this feature to set this system apart and possibly make it the first of its kind. The web-based hostel management system can monitor curfews and safety, offer insightful information about student movement patterns, and improve hostel resource use by adding a student attendance tracking component.

This novel method of hostel management can distinguish the system from other alternatives and meet a critical need in the context of university residence halls while also considerably improving the system's overall efficiency and accountability.

2.5 Challenge & Future Directions

Developing a hostel management system faces several key challenges and potential future directions. These include:

Data Migration: Transitioning from legacy systems to a modern hostel management system can be complex and challenging. It requires carefully designed data migration strategies to ensure the seamless and secure transfer of critical information from the existing systems to the new hostel management platform. This process must be executed with meticulous attention to detail to avoid any loss or corruption of the data, which could have significant implications for the effective operation and management of the hostel system. Proper planning, testing, and validation of the data migration process are essential to ensure a successful transition and maintain the integrity of the hostel's records and information (Khamis et al., 2020).

User Training: Effective utilization of a hostel management system requires comprehensive training programs for staff and students. While these training efforts can be resource-intensive, they are crucial for successfully implementing and adopting the system. Thorough training is essential to ensure that all users, including administrative personnel and student residents, are well-versed in the system's functionalities, user interfaces, and data management protocols. This training process should be designed to optimize user competency, enhance system utilization, and facilitate a seamless transition from legacy systems or manual processes. Although the initial investment in training can be significant, it ultimately contributes to the long-term efficiency and sustainability of the hostel management system, enabling the institution to maximize its benefits and achieve its operational goals. (Chaudhri & Kevat, 2021)

Cost Considerations: Implementing a web-based hostel management system can require substantial initial and ongoing investments, particularly for smaller academic institutions, and these costs are often necessary to modernize and streamline student accommodation management.

The expenses associated with system development, hardware, software, technical support, and updates must be weighed against the long-term benefits of improved operational efficiency and utilization of limited hostel resources. However, the real value lies in the improved data tracking, which can make users feel more informed and in tune with the system. Although smaller institutions may face more significant financial constraints, the advantages of a well-designed hostel management system, such as reduced manual labor, improved data tracking, and enhanced decision-making, can ultimately offset the initial capital outlay, leading to more cost-effective and sustainable hostel administration in the long term.

Future research and development should focus on improving the following areas:

Integration with Other Institutional Systems: Integrating the web-based hostel management system with other campus systems, such as academic, financial, and human resources, can significantly improve operational efficiency. By enabling data exchange and streamlining administrative tasks, this integration supports a more comprehensive and coordinated management of student housing, academic records, finances, and personnel information. Significantly, this holistic approach can enhance decision-making, giving administrators more control over resource allocation and improving the user experience for both staff and students within the university.

Enhanced User Interfaces: Ongoing enhancement of the user interface to accommodate diverse user groups, including students with disabilities, is crucial. This entails designing accessible and intuitive interfaces that cater to all users' varying needs and capabilities. Such interfaces may incorporate features like screen readers, voice commands, adjustable font sizes, contrast settings, and navigation optimized for keyboard or assistive technologies. By prioritizing inclusive design principles, the web-based hostel management system can ensure equitable access and usability, fostering a more inclusive and supportive living environment for students with diverse abilities.

Sustainability: Web-based hostel management systems can be designed to be eco-friendly and contribute to the institution's sustainability goals. This can involve energy-efficient infrastructure, water conservation measures, waste management systems, and integrating renewable energy sources. By prioritizing sustainability, these systems can help align the university's housing operations with its broader environmental commitment, setting an example for sustainable campus living and contributing to its overall green initiatives. Developing hostel management systems that optimize resource usage and promote environmentally friendly practices can be essential to these systems.

Modernizing Student Housing Management: The development and implementation of web-based hostel management systems mark a significant leap in the administration of student accommodations. By enhancing the efficiency and effectiveness of hostel operations, these innovative systems improve service delivery and provide valuable insights for data-driven decision-making at the institutional level. Their continuous adaptation to address emerging challenges further solidifies their integral role in modernizing student housing management within educational environments. By integrating cutting-edge technologies and user-centric design principles, these systems streamline processes, enhance user experiences, and optimize the utilization of limited hostel resources, making them an indispensable asset for universities and colleges worldwide.

CHAPTER THREE: SYSTEM ANALYSIS & DESIGN

3.1. Introduction

This chapter gives a comprehensive overview of the execution of the web-based hostel management system. It delves into the intricate details of the development process, including the selection of the appropriate development environment, the design and architecture of the system, the rigorous implementation of software testing procedures, the creation of thorough documentation, the development of a user-friendly manual, and the inclusion of the complete source code listing. These features collectively showcase the practical realization of this project, guiding the reader through the step-by-step process of bringing the system to life and emphasizing the meticulous attention to detail applied throughout the development lifecycle.

3.2. Analysis of Current System

3.2.1. Problem of the Current System

Like many other universities worldwide, Kigali Independent University faces significant challenges in managing student hostel facilities. The university's existing manual method for managing student accommodation has inefficient record-keeping, difficulty monitoring occupancy levels, a lack of automated processes for room distribution and student check-in/out, and poor communication between students and hostel administrators.

This outdated process is tedious, prone to mistakes, and unable to give university administration the real-time data and reporting they need to make informed decisions. Addressing these significant difficulties will improve the overall effectiveness and responsiveness of the university's student housing operations.

3.3. Analysis Of the New System

3.3.1. Introduction

This section presents a more thorough examination of the current issues with the university's hostel management system and elaborates on the requirements and goals of the proposed web-based solution. It conducts a comprehensive analysis of the system's functional requirements, delineates the key components and their interrelationships through a detailed functional diagram, outlines the overall system design, and elucidates the methodological framework that will guide the development and implementation of the new web-based hostel management system.

3.3.2. System Requirements

Table 1: System Requirement (Functional & Non-Functional Requirements)

Functional Requirement	***Non-Functional Requirement***
The system will allow students to create an account to access and book a hotel.	*Security:* All generated or accepted passwords must be stored in an encrypted database.
The system will have a user authorization process that requires students to log in using a username and word.	*User-friendly:* The system must be user-friendly, understandable, and easy to use
The system will allow the details in the catalog to be viewed.	*Privacy:* The system shall be able to protect the user's privacy
Students can recover their password by clicking on the" Forgot Password" link.	*Availability:* The system should be available to all users 24 hours
The system can store and recover all the information.	Performance: The system must perform quickly and should be able to respond to requests in a reasonable amount of time.
The system can store and recover all the information.	Accessibility: Students can access their results from any location (as long as they are within a network service reception area).
The system will be capable of backing up data.	Recoverability: The system should be able to recover from any disturbance.
The user can log out after they finish using the system.	Environmental: The system should be able to run on smartphones and PCs.
The system will allow students to book, edit, view, and print information about his/ her hotel.	*User-friendly:* The system must be user-friendly, understandable, and easy to use
The system will also allow students to file complaints and give feedback.	*Privacy:* The system shall be able to protect the user's privacy
The system will allow the admin to view all information about the hostel, rooms, and registered students and make decisions grounded on the complaints filed.	*Availability:* The system should be available to all users 24 hours

3.3.3. Functional Diagrams

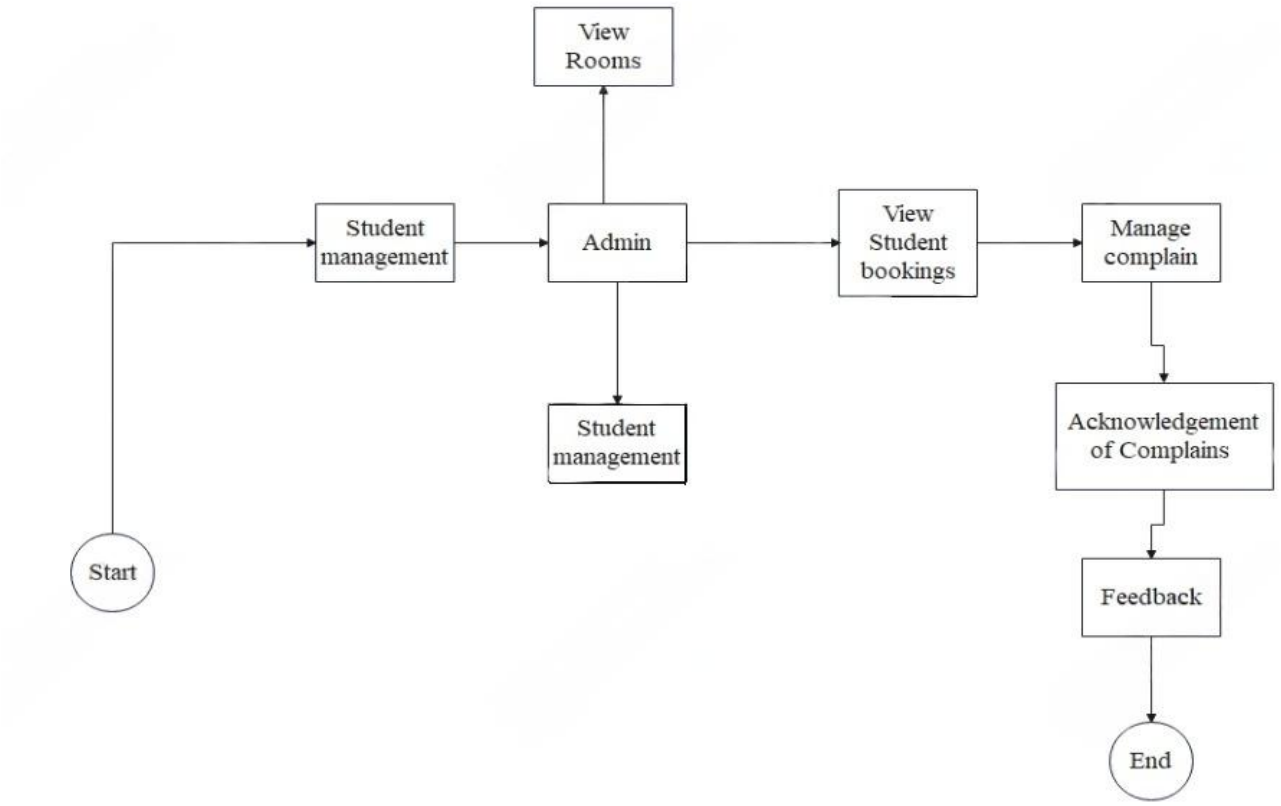


Figure 1: Functional Diagram

The functional diagram illustrates a system for managing student housing through a hostel management system. It begins with a "Start" point before branching into "Student management" and "Admin." The "Admin" area is further divided into "View Rooms," "Student Management," and "Manage Complaints." The "Student management" option loops back to itself, indicating that management activities are still occurring. The "Manage Complaints" path leads to "Acknowledgement of Complaints" and then "Feedback," assuring a disciplined approach to handling and resolving issues. The figure ends with the "End" point, which depicts a thorough process for administrative duties involving students, room management, and complaint resolution. This ordered flow streamlines complex administrative operations, assuring efficiency and clarity.

3.3.4. Methodological Approach

3.3.4.1. Data Collection Techniques

I conducted a thorough documentation review and analysis to gather the necessary information for this research. This comprehensive review, a crucial part of the research process, closely examined relevant books, journal articles, course materials, and online resources focused on web-based hostel management systems. This comprehensive review gave the researchers a detailed understanding of the existing approaches, best practices, and challenges in student housing management. Drawing on the insights, theoretical foundations, and contextual knowledge gained through the documentation review, the team was well-equipped to inform the design and development of the proposed web-based hostel management system for the Kigali Independent University (German et al., 2021).

3.3.4.2. Software Development Methodology

Agile methodology is an iterative approach to software development that emphasizes flexibility, collaboration, and responsiveness to changing requirements. This collaborative nature of Agile makes the team feel included and part of the process. Instead of a rigid, linear process, Agile encourages cross-functional teams to work together, continuously gathering feedback and adapting the solution as needed. This approach promotes customer satisfaction by incrementally delivering small, functional software increments, allowing for rapid iteration and the incorporation of user feedback throughout the development lifecycle. The Agile methodology's emphasis on self-organization, collaboration, and adaptability helps teams quickly respond to evolving requirements and deliver high-quality software that closely aligns with user needs (Alami et al., 2022).

The Agile software development approach emphasizes frequent team meetings, known as sprints, to review progress and plan the next steps regularly. This iterative method promotes continuous feedback and adaptation, ensuring the final product meets user requirements and expectations.

Agile's basic principle is the capacity to adjust quickly to change rather than thoroughly adhering to a predefined plan.

This methodology encourages close collaboration among team members and stakeholders, fostering alignment and a shared focus on common objectives.

Agile development prioritizes people and their interactions over inflexible processes and systems. This emphasis on communication and teamwork enables teams to identify and address issues early, leading to higher-quality outcomes.

Agile development often involves daily stand-up meetings where team members share updates on their progress, plans for the day, and any challenges they face.

These fast, concentrated sessions help the team stay together, identify and address concerns quickly, and keep the project on schedule. Additionally, retrospective meetings at the end of each sprint or iteration allow the team to reflect on their performance, discuss what went well, identify areas for improvement, and explore ways to enhance their processes and practices. This continuous learning and adaptation culture is a core tenet of the Agile approach, enabling the team to consistently deliver high-quality software that meets evolving user needs (Valpadasu et al., 2020).

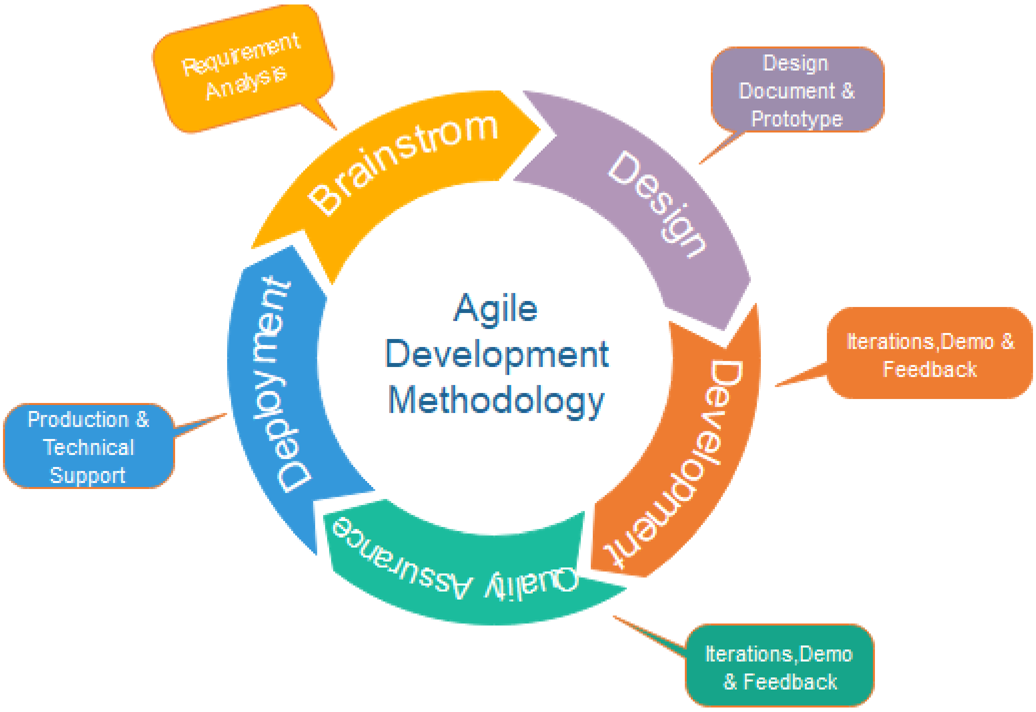

Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Agile Development Methodology
Iterations,Demo & Feedback
Development
Production & Technical Support
Deployment
Quality Assurance
Iterations,Demo & Feedback

Figure 2: Agile Model for HMS

Advantages of the Agile model

• Flexibility: The Agile methodology offers flexibility, enabling frequent iterations and adjustments to accommodate changing requirements and priorities.

• User Collaboration: Close collaboration with users, involving them throughout the process, ensures the system aligns with their evolving needs and expectations.

• Faster Time to Market: The Agile methodology's focus on incremental delivery of functional software allows for faster product releases and enhanced responsiveness to evolving market needs.

- Transparency: Regular demonstrations and open communication within the team promote transparency, fostering trust and accountability.

- Adaptability: Agile teams can respond quickly to challenges and opportunities, making managing risks and seizing opportunities easier.

- Continuous Improvement: Retrospectives and iterative cycles encourage continuous learning, improvement, and innovation.

Below are some of the technologies and tools that have been used in the development of the system:

- Visual Studio Code is a new and improved code editor for building and debugging modern online and cloud applications.

- Bootstrap: A CSS framework for responsive, mobile-first front-end web development.

- Xampp Database: XAMPP is a popular open-source software package that facilitates setting up a local web server environment. It includes Apache, MySQL, PHP, and Perl components, forming the acronym "XAMPP." MySQL is the database management system (DBMS) included in XAMPP.

3.3.4.3. System Design Methodology

The Structured Systems Analysis and Design Method is a rigorous and organized methodology for analyzing and designing information systems. It delineates the systems development process into well-defined phases: feasibility assessment, requirements analysis, design, and implementation.

The Structured Systems Analysis and Design Method emphasizes thorough documentation and detailed modeling techniques to ensure the final system satisfies user needs and delivers on time and within budget. This structured approach is particularly advantageous for large, complex projects, as its careful planning and rigorous processes help manage risks and guarantee the delivery of a high-quality system. SSADM's well-defined phases, spanning feasibility assessment to implementation, provide a comprehensive framework that empowers project teams to systematically analyze, design, and implement information systems closely aligned with user requirements and organizational objectives.

The Structured Systems Analysis and Design Method utilizes several essential tools throughout the system development process. Data Flow Diagrams depict the data flow within the system, illustrating how data is processed through various inputs and outputs. Additionally, Entity-Relationship Diagrams are utilized to model the data entities and their interrelationships, clearly understanding the system's data structure. Furthermore, a Data Dictionary serves as a centralized repository, containing detailed information about the data, including its meaning, relationships, origin, usage, and format, ensuring consistency across the project.

DATA FLOW DIAGRAM (Level 0&level 1)

<u>Level 0</u>

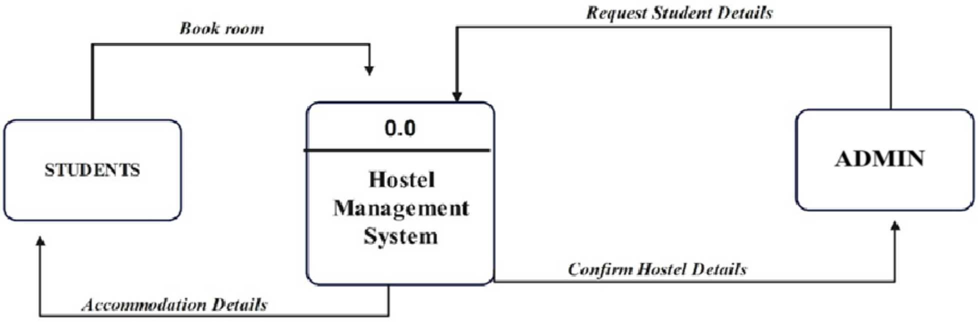


Figure 3: Level 0 DFD

The diagram provides a simplified representation of a hostel management system, focusing on the Hostel Management System as a mediator between students and staff. The system involves students who inquire about accommodations and make reservations and administrators who verify student information. However, it lacks crucial information about the system's operation, such as procedures for making reservations and maintaining student data. A more detailed schematic with more parts and procedures would be necessary to understand the system.

Level 1

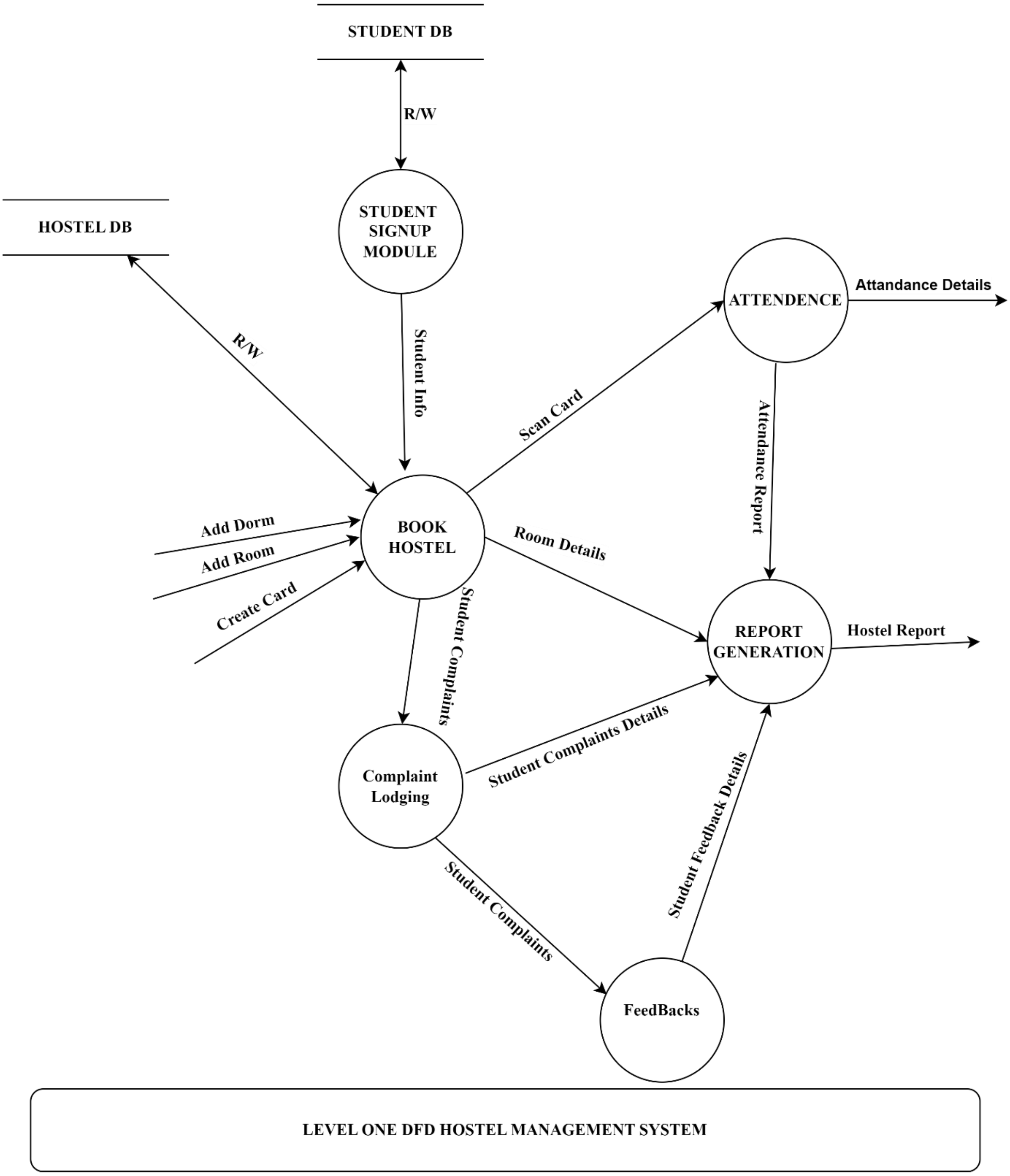


Figure 4: Level 1 DFD

The Hostel Management System (HMS) is the primary process depicted in this Level 1 Data Flow Diagram (DFD) for a hostel management system. This system communicates with numerous data stores and subprocesses representing various informational and functional domains. Some crucial procedures include user management, authorization management, booking management, admin management, feedback management, hostel room management, login management, and attendance management. Additionally displayed are data stores like Access Log, Complaints, and Dorms. The data flow between components is depicted in the diagram. For example, booking management interacts with attendance management to track resident attendance, and user management interacts with the dorm data store to handle resident information.

Entity Relationship Diagram (ERD)

ERD

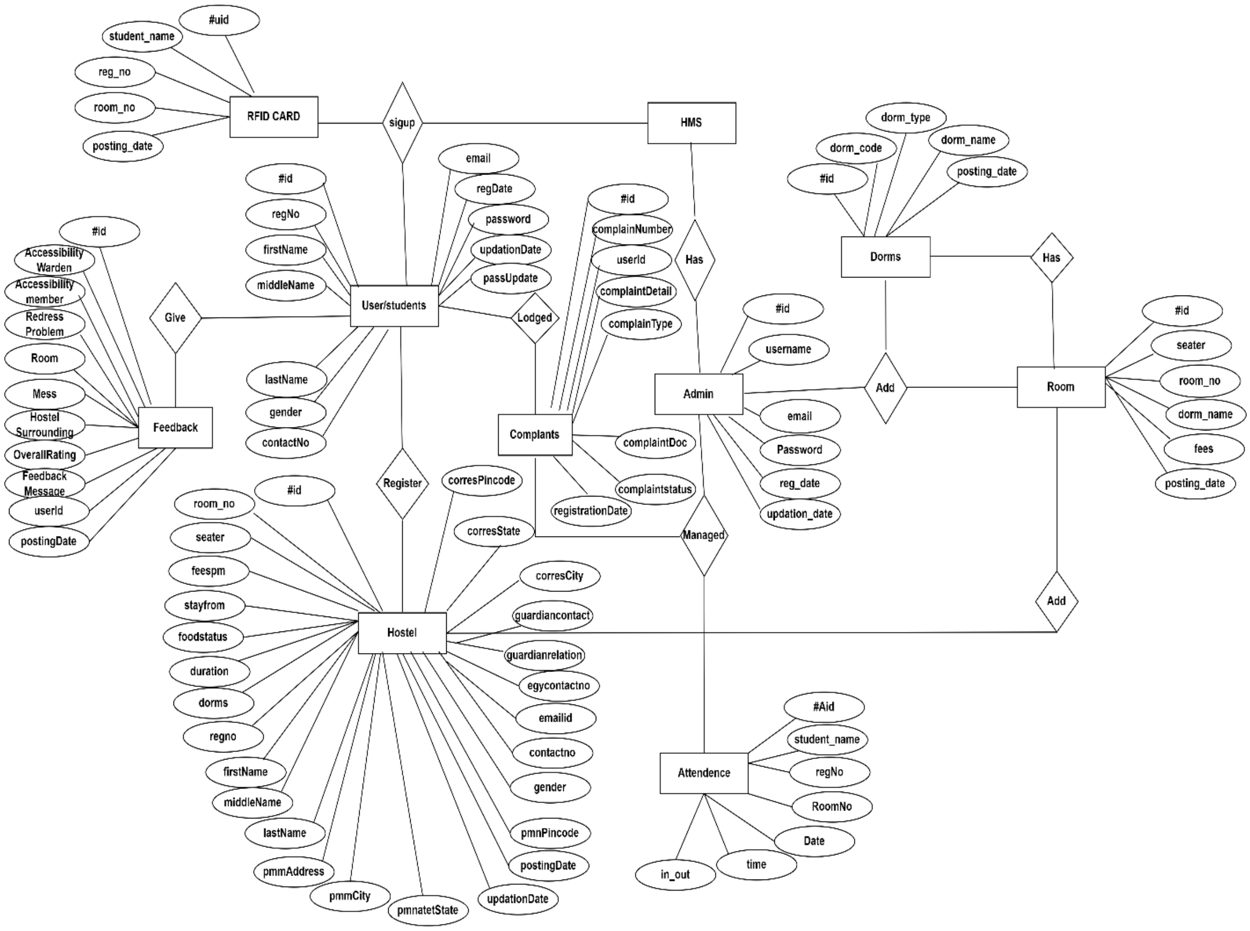


Figure 5: ERD of HMS

Critical elements in the entity relationship diagram include Users, Students, Rooms, Dorms, and Complaints. Users are associated with the Dorms entity and might be Administrators or Students. Students live in rooms in dorms, and administrators run the whole thing. Bookings and Complaints enhance the relationship between Users and Dorms by illustrating user interactions. The components known as hostel and feedback are separate entities; Hostels contain information about the dorm facilities, while feedback records user opinions. Each has links to other entities, pointing to a more intricate web of relationships. Some entities and attributes have specific purposes, like Grievances, Redressal Problems, Accessibility Warden, and Accessibility members.

Further analysis is necessary to precisely comprehend the nature of links between some entities and the purpose of qualities.

Circuit Diagram of the RFID Attendance System

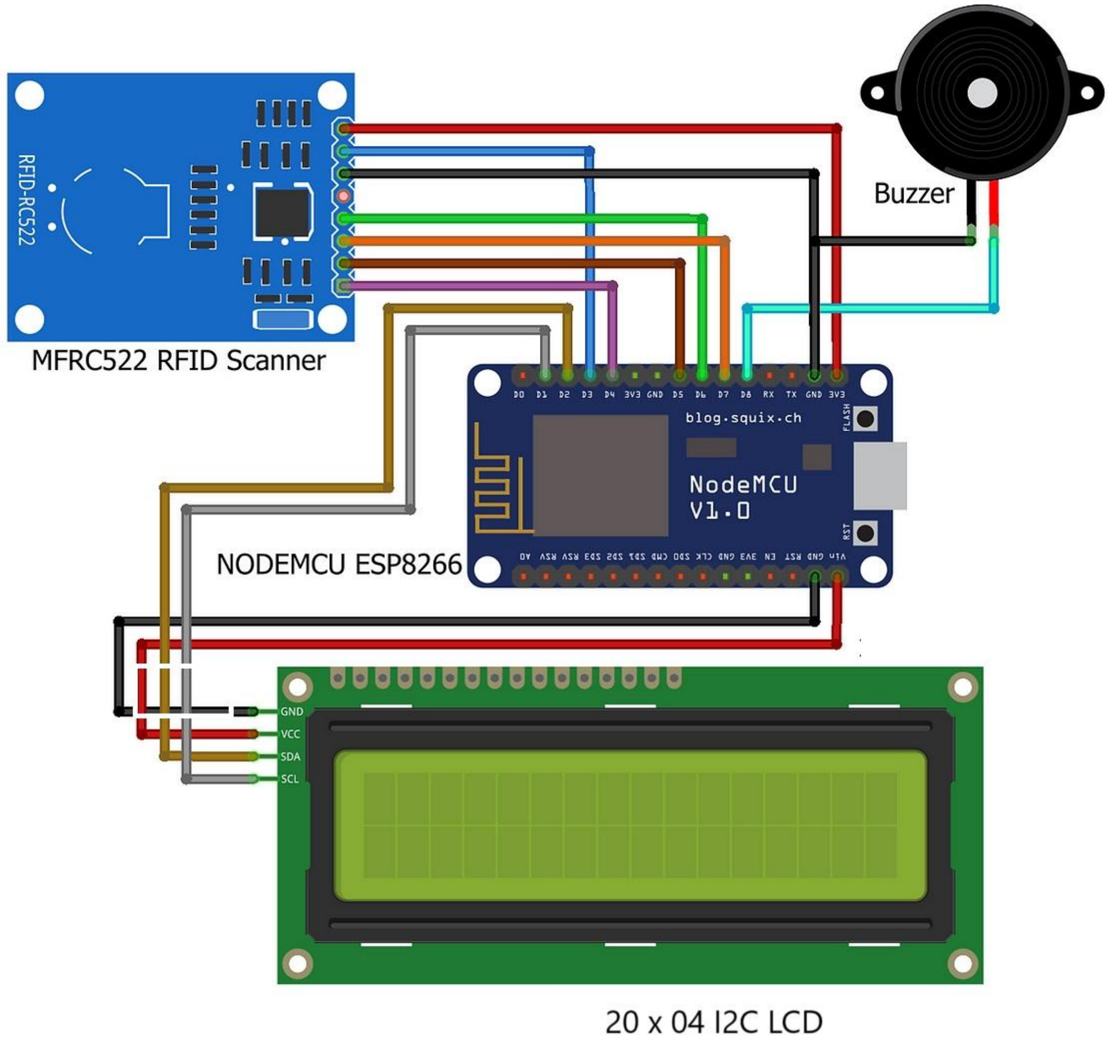


Figure 6: Circuit Diagram of RFID Attendance System

The schematic diagram depicts a system that leverages a NodeMCU ESP8266 microcontroller to read and process RFID tags. The core components of this system comprise an MFRC522 RFID reader module, a NodeMCU development board, a 20x4 character LCD, and a buzzer.

The MFRC522 module interfaces with the NodeMCU's SPI pins, enabling communication between the microcontroller and the RFID reader. The NodeMCU then processes the data received from the RFID tag and presents the tag's unique identifier on the LCD screen. Additionally, a buzzer connected to a digital pin on the NodeMCU provides audible feedback when an RFID tag is detected, alerting the user that a tag has been successfully read. The circuit also includes connections for power supply, ground, and the LCD display's I2C interface, allowing the entire system to function as an integrated RFID-based identification and monitoring solution.

Data Dictionary

Database hostel

***Table2:* Admin table Structure**

Table 2: Admin Table Structure

Column	Type	Index
id	int(11)	PRIMARY
username	varchar(255)	
email	varchar(255)	
password	varchar(300)	
reg_date	timestamp	
updation_date	date	

Table3: adminlog table Structure

Table 3: Admin log Table Structure

Column	Type	Index
id	int(11)	UNIQUE
adminid	int(11)	
ip	varbinary(16)	
logintime	timestamp	

Table4: Attendance Table Structure

Table 4: Attendance Table Structure

Column	Type	**Index**
Aid	int(255)	PRIMARY
student_name	varchar(225)	
regNo	int(225)	
RoomNo	int(225)	
Date	date	
time	time(6)	
in_out	varchar(255)	

Table5: complainthistory Table Structure

Table 5: Complaint History Table Structure

Column	Type	Index
id	int(11)	PRIMARY
complaintid	int(11)	
compalintStatus	varchar(255)	
complaintRemark	mediumtext	
postingDate	timestamp	

Table6: Structure for table complaints

Table 6: Complaints Table Structure

Column	Type	Index
id	int(11)	PRIMARY
ComplainNumber	bigint(12)	
userId	int(11)	FOREIGN
complaintType	varchar(255)	
complaintDetails	mediumtext	
complaintDoc	varchar(255)	
complaintStatus	varchar(255)	
registrationDate	timestamp	

Table7: Structure for table dorms

Table 7: Dorms table structure

Column	Type	Index
id	int(11)	PRIMARY
dorm_code	varchar(255)	
dorm_type	varchar(255)	
dorm_name	varchar(500)	UNIQUE
posting_date	timestamp	

Table8: Structure for table feedback

Table 8: Feedback Table Structure

Column	Type	Index
id	int(11)	PRIMARY
AccessibilityWarden	varchar(255)	
AccessibilityMember	varchar(255)	
RedressalProblem	varchar(255)	
Room	varchar(255)	
Mess	varchar(255)	
HostelSurroundings	varchar(255)	
OverallRating	varchar(255)	
FeedbackMessage	varchar(255)	
userId	int(11)	FOREIGN
postinDate	timestamp	

Table9: Structure for table registration

Table 9: Registration Table Structure

Column	Type	Index
id	int(11)	PRIMARY
room_no	int(11)	FOREIGN
seater	int(11)	
feespm	int(11)	
foodstatus	int(11)	
stayfrom	date	
duration	int(11)	
dorms	varchar(500)	FOREIGN
regno	int(11)	
firstName	varchar(500)	
middleName	varchar(500)	
lastName	varchar(500)	
gender	varchar(250)	

contactno	bigint(11)	
emailid	varchar(500)	FOREIGN
egycontactno	bigint(11)	
guardianName	varchar(500)	
guardianRelation	varchar(500)	
guardianContactno	bigint(11)	
corresAddress	varchar(500)	
corresCIty	varchar(500)	
corresState	varchar(500)	
corresPincode	int(11)	
pmntAddress	varchar(500)	
pmntCity	varchar(500)	
pmnatetState	varchar(500)	FOREIGN
pmntPincode	int(11)	
postingDate	Timestamp	
updationDate	varchar(500)	

Table10: Structure for rooms

Table 10: Rooms Table Structure

Column	Type	Index
id	int(11)	PRIMARY
seater	int(11)	
room_no	int(11)	UNIQUE
dorm_name	varchar(500)	FOREIGN
fees	int(11)	
posting_date	Timestamp	

Table11: Structure for states

Table 11: States Table Structure

Column	Type	Index
id	int(11)	PRIMARY
State	varchar(500)	UNIQUE

Table12: Structure for userlog

Table 12: Userlog Table Structure

Column	Type	Index
id	int(11)	PRIMARY
userId	int(11)	
userEmail	varchar(255)	FOREIGN
userIp	varbinary(16)	
city	varchar(255)	
country	varchar(255)	
loginTime	timestamp	

Table13: Structure for table userregistration

Table 13: User Registration Table Structure

Column	Type	Index
id	int(11)	PRIMARY
regNo	varchar(255)	UNIQUE
firstName	varchar(255)	
middleName	varchar(255)	
lastName	varchar(255)	
gender	varchar(255)	
contactNo	bigint(20)	
email	varchar(255)	UNIQUE
password	varchar(255)	
regDate	timestamp	
updationDate	varchar(45)	
passUdateDate	varchar(45)	

Table 14: structure for table rfid_data

Table 14: RFID Data Table Structure

Column	**Type**	**Index**
uid	varchar(225)	PRIMARY
student_name	varchar(225)	
reg_no	int(10)	FORIENG
room_no	int(6)	FORIENG
posting_date	Timestamp	

CHAPTER FOUR: SYSTEM IMPLEMENTATION

4.1. Implementation and Coding

Implementing and developing the web-based hostel management system for Kigali Independent University involved several critical steps. First, the system was designed using an agile software development approach, facilitating iterative testing and stakeholder feedback (Chikwendu, 2021). The system was built using a stack comprising PHP, HTML, CSS, JavaScript, and MySQL as the database management system. Key functionalities of the system encompassed student registration, room booking and assignment, check-in management, lodging, reviewing complaints, and providing feedback on the system's functionalities.

The system's implementation began by developing the user interface and navigation flow. The user interface was outlined to be natural and user-friendly, highlighting clear labels and navigation choices. The backend was developed using PHP and MySQL, with the database schema designed to efficiently store and manage student, room, and other hostel-related data. Access control mechanisms were integrated to enhance the system's security, allowing different user roles to access relevant functionalities.

The system's development also involved the integration of various modules, such as student registration, room booking, check-in/check-out management, and complaint handling. The student registration module allowed new students to create accounts and provide their personal and contact information. The room booking module enabled students to view available rooms and make reservations, while the check-in/check-out management module managed the process of students moving in and out of the hostel. The complaint handling module provided a platform for students to log and track their issues, with the hostel management team able to respond and resolve them (Jafrudin & Putra, 2020) (Peng & Xie, 2015) (Anis & Safar, 2022).

Throughout the development process, regular testing and feedback gathering were conducted to ensure the system's functionality, usability, and alignment with the university's requirements.

4.1.1. Introductions

The deployment of a web-based hostel management system at ULK is expected to optimize hostel operations and improve efficiency.

This system aims to improve transparency and efficiency by implementing a comprehensive web-based system that caters to the diverse needs of stakeholders, including students, hostel administrators, and university administration. The system was built using a technology stack comprising PHP, HTML, CSS, JavaScript, and MySQL as the database management system. Key features include student registration, room booking, check-in and check-out procedures, and mechanisms for submitting and addressing student complaints. Moreover, the system emphasizes security and accessibility, incorporating access control features that cater to diverse user roles. The overarching objective is to enhance the efficiency and transparency of the hostel management processes at Kigali Independent University ULK.

4.1.2. Description Of Implementation Tools and Technology

4.1.2.1. Sever-Side (Backend) Tools

Server-side tools play a crucial role in enabling the implementation of various functionalities on the backend of the Hostel Management System. These tools include robust frameworks and specialized libraries tailored to data processing, business logic handling, and API development. By leveraging these server-side tools, developers can efficiently build competent and scalable backend systems adept at managing hostel management's diverse data and operations. These tools provide a structured and efficient approach to developing the core backend components of the HMS, empowering the system to cater to the diverse needs of students, administrators, and other stakeholders.

4.1.2.1.1. MySQL

The Hostel Management System at Kigali Independent University ULK utilized MySQL, a prominent open-source relational database management system, as the primary data storage and management solution. By harnessing the capabilities of MySQL, the development team was able to construct a robust and highly scalable data storage infrastructure. This infrastructure was purposefully designed to efficiently manage the diverse range of data associated with various aspects of hostel operations, such as student records, room information, and detailed logs of check-in and check-out events. MySQL's strong support for SQL querying, data integrity protocols, and robust transaction management features enabled the developers to ensure efficient data storage, retrieval, and manipulation. As a result, this contributed to the overall reliability and high-performance operation of the Hostel Management System, effectively meeting the diverse needs of the university's stakeholders.

4.1.2.1.2. PHP

PHP, a widely used server-side scripting language, plays a crucial role in the backend development of a Hostel Management System. The PHP code, executed on the server, allows for processing user requests, interaction with the database, and generating dynamic content for the web application.

In the context of the Hostel Management System at Kigali Independent University ULK, the development team chose PHP as the primary programming language for the backend. PHP's widespread adoption influenced this decision, as its robust ecosystem of frameworks and libraries and its ability to efficiently handle server-side tasks, such as:

- **User authentication and authorization**: PHP can implement secure login and user management functionalities, ensuring the system's access control and data protection mechanisms.
- **Database integration**: PHP's seamless integration with MySQL, the database management system, allows for efficient data storage, retrieval, and manipulation.
- **API development**: PHP can be leveraged to build RESTful APIs, enabling the system to expose its functionalities to external applications or integrations.

By leveraging PHP's capabilities, the development team created a robust and scalable backend for the Hostel Management System, catering to the diverse needs of the university's stakeholders and ensuring the system's overall reliability and performance.

4.1.2.1.3. XAMPP

The Hostel Management System for Kigali Independent University ULK utilized the XAMPP software stack during development and deployment. XAMPP includes the Apache web server, MySQL database management system, and PHP server-side scripting language.

The XAMPP environment provided an integrated local system development, testing, and debugging platform. The Apache web server and PHP enabled the development team to create and test dynamic web pages and server-side logic. Additionally, the MySQL database component of XAMPP allowed the developers to set up and manage the underlying data storage infrastructure, facilitating the integration of the system's data model with the backend functionality.

Using XAMPP during development ensured a streamlined and efficient workflow. Developers could work on the system in a self-contained and controlled environment, mimicking the production setup. This approach allowed the team to identify and resolve issues early in the development process, contributing to the overall stability and reliability of the Hostel Management System.

4.1.2.2. Client-Side (Front-end) Tools

The Hostel Management System at Kigali Independent University ULK also utilized client-side tools and technologies to enhance the user experience and create a visually appealing interface. These client-side tools were pivotal in shaping the system's front end, ensuring seamless interaction and effective data visualization for users.

4.1.2.2.1. HTML:

HTML, the Hypertext Markup Language, is the standard markup language used for structuring the content and layout of web pages. In the Hostel Management System, HTML was leveraged to define the semantic structure of the user interface, including critical elements such as headings, paragraphs, forms, and tables. By harnessing the capabilities of HTML, the team created a well-organized and easily navigable user interface, which enhanced the overall user experience for the system's stakeholders. This structured approach to the front-end design ensured that the content and functionality of the Hostel Management System were presented clearly and intuitively, making it more accessible and user-friendly for students, administrators, and other end-users.

4.1.2.2.2. CSS:

Cascading Style Sheets were utilized to control and refine the visual presentation and styling of the Hostel Management System's user interface. The CSS enabled the precise definition and fine-tuning of the layout, colors, typography, and various other visual aspects of the system, ensuring a consistent, aesthetically pleasing, and visually engaging user experience across the different pages and components of the application.

The strategic combination of HTML for establishing the semantic structure and CSS for enhancing the visual styling facilitated the development of a clean, responsive, and visually appealing interface for the Hostel Management System. This approach allowed the diverse preferences and needs of the university's stakeholders to deliver an intuitive and visually captivating user experience.

4.1.2.2.3. JavaScript:

JavaScript, a client-side scripting language, enhanced the interactivity and dynamism of the Hostel Management System's user interface. The execution of JavaScript code within the user's web browser enabled the implementation of several key features:

1. **Form validation:** The Hostel Management System leveraged JavaScript's capabilities to implement real-time form validation. This allowed the system to ensure the accuracy and completeness of user inputs before the forms were submitted, enhancing the overall data integrity and user experience. By validating the data as the users entered it, the system could provide immediate feedback and guidance, helping them correct any errors or omissions and ultimately leading to a more reliable and efficient data collection process.

2. **Dynamic content rendering:** By leveraging the power of JavaScript, the system could dynamically manipulate the HTML and CSS of web pages, enabling real-time generation and updates of content. This enhanced the overall responsiveness and user engagement of the Hostel Management System, allowing for a more interactive and visually dynamic user experience. The development team harnessed JavaScript's ability to modify the structure and styling of web pages on the client side, creating a more engaging and adaptive interface that could respond to user actions and provide immediate feedback, ultimately improving the overall usability and satisfaction of the system's end-users.

3. **AJAX-based asynchronous communication:** The Hostel Management System capitalized on JavaScript's AJAX capabilities to facilitate asynchronous data exchange between the client-side web browser and the server-side components. This approach allowed the system to update specific web page parts without requiring a complete refresh, improving overall responsiveness and user experience. By enabling this asynchronous communication, the system could retrieve and display new data or update existing information on the fly without interrupting the user's workflow or causing disruptive page reloads. This enhanced the system's interactivity and efficiency, enabling users to interact with the application more seamlessly and immediately.

4. **Interactive visualizations:** The development team employed JavaScript libraries, such as D3.js or Chart.js, to create engaging and intuitive interactive data

visualizations, enabling users to explore and analyze information more effectively. These visualizations allowed users to interact with charts, graphs, and other visual representations of data, fostering a deeper understanding of the hostel management processes and the underlying information.

By incorporating JavaScript, the development team was able to craft a more dynamic and interactive user interface for the Hostel Management System, thereby improving the overall user experience and making the website more engaging and sensitive to end-user needs.

4.1.2.2.4. Bootstrap:

The Hostel Management System leveraged the Bootstrap front-end framework to streamline the development of a responsive and mobile-friendly user interface. Bootstrap provided a comprehensive set of pre-designed UI components, typography, and CSS styles, allowing a quick and efficient development of a visually appealing and consistent user interface across different devices and screen sizes.

The team used Bootstrap's grid system, responsive utilities, and pre-built components like forms, buttons, and navigation menus to create a clean and organized layout. This ensured that the system's content and functionality were easily accessible and intuitive for users. Additionally, Bootstrap's built-in support for responsive design principles ensured the Hostel Management System's interface adapted seamlessly to a wide range of screen sizes and devices, from desktop computers to mobile phones.

By leveraging the Bootstrap framework, the development team saved time and resources while delivering a high-quality, responsive, and visually consistent user experience for the Hostel Management System. This allowed them to focus more on the system's core functionality and features rather than spending substantial effort on building the user interface from scratch.

The strategic implementation of HTML, CSS, JavaScript, and the Bootstrap framework collectively contributed to developing a robust and user-friendly Hostel Management System for Kigali Independent University ULK. The front-end development approach focused on creating a visually appealing and responsive interface that could effectively meet the diverse needs and preferences of the university's stakeholders, including students, administrators, and staff.

4.1.2.2.6. IDEs:

An Integrated Development Environment was employed to streamline the coding and testing processes for the Hostel Management System. One commonly utilized IDE was Microsoft's free, open-source, cross-platform offering, Visual Studio Code. This IDE provided a comprehensive suite of features, including code editing, debugging, and integration capabilities for various programming languages and frameworks, such as JavaScript, HTML, CSS, and Node.js.

The robust capabilities of Visual Studio Code were leveraged to enhance productivity and ensure the quality of the Hostel Management System's codebase. Some key features that benefited the development process include:

1. **Advanced Code Editing:** Visual Studio Code offered sophisticated code editing features, including syntax highlighting, code completion, and code folding, which facilitated the efficient creation of clean, organized, and error-free code.
2. **Debugging Tools:** The IDE's built-in debugging tools, such as breakpoints, step-through execution, and variable inspection, enable the quick identification and resolution of errors within the system's codebase, ensuring a stable and reliable application.
3. **Version Control Integration:** Visual Studio Code is seamlessly integrated with Git, the popular distributed version control system. This allows the effective management of code repositories, tracking changes, and collaboration on the project.
4. **Extension Ecosystem:** The extensive ecosystem of Visual Studio Code extensions provided access to various plugins and tools, such as linters, formatters, and deployment utilities, further enhancing the development workflow and streamlining the overall development process.

Another IDE that was utilized to improve this project was the Arduino IDE. The Arduino IDE is a free, open-source, cross-platform application designed to program and upload code to Arduino boards and other compatible microcontroller-based devices. (Tan et al., 2023)

The development team utilized the Arduino IDE to create and upload firmware for the RFID attendance system and its components integrated into the hostel management system, which helps track students' movements in and out of the hostel facilities.

A few noteworthy features of the Arduino IDE that were useful for this project include:

1. **Seamless integration with Arduino hardware:** The IDE provides a user-friendly interface for writing, compiling, and directly uploading code to Arduino boards, simplifying the development and testing of the RFID attendance system.
2. **Extensive library support:** The Arduino IDE offers a vast ecosystem of pre-built libraries for various sensors, modules, and communication protocols, which expedited the development of the RFID attendance system integration.
   - **Debugging and monitoring tools:** The IDE's built-in serial monitor and debugging capabilities enabled the team to efficiently identify and resolve any issues with the firmware or the hardware-software interface.

The strengths of both the Visual Studio Code and Arduino IDEs helped create a comprehensive and integrated Hostel Management System. This system combines the powerful front-end development capabilities of Visual Studio Code with the specialized microcontroller programming features of the Arduino IDE.

4.1.2.3. RFID Attendance System

The Hostel Management System will incorporate an RFID-based attendance system, as described in the provided sources (Mijić et al., 2019), to enhance the tracking and recording of student movements in and out of the university's hostels. This RFID-based system utilized RFID tags embedded within the students' ID cards and RFID reader devices strategically placed throughout the hostels. When students tap their ID card on the RFID reader, the system automatically registers their arrival or departure, creating a detailed record of their attendance and movements.
The RFID-based attendance system offered several critical benefits to the Hostel Management System implementation:

1. ***Automated Attendance Tracking***: RFID technology eliminates the need for manual attendance-taking, streamlines the process, and reduces the potential for human error, improving the efficiency and accuracy of attendance monitoring.

2. ***Improved Monitoring and Reporting:*** The system provided detailed attendance records and reports, allowing administrators to monitor student attendance and movements closely, which is crucial for effective hostel management (Kassim et al., 2012). This enhanced the university's ability to oversee and manage the hostel facilities.

3. ***Integration with Other Systems:*** As mentioned in the sources, the RFID-based attendance system could be integrated with external systems, enabling seamless data sharing and process automation for the overall hostel management operations. This improved the integration and interoperability of the Hostel Management System with other university systems.

By incorporating the RFID-based attendance system into the Hostel Management System, we achieved higher efficiency, accuracy, and integration with other university systems, enhancing the overall management and oversight of the hostel facilities.

4.1.2.3.1. NodeMCU (ESP8266):

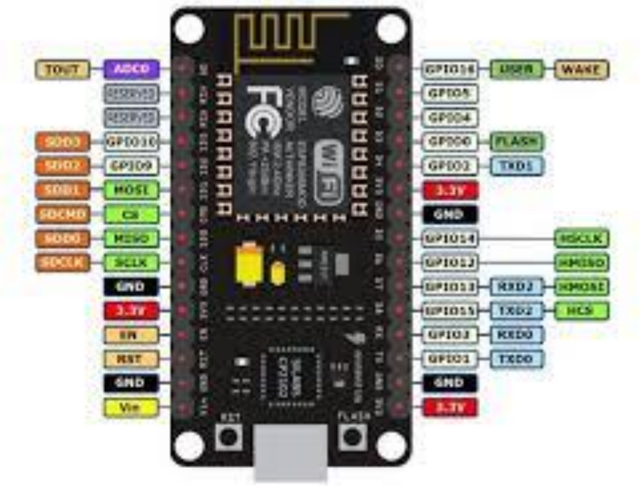


Figure 7: NodeMCU (ESP8266)

The NodeMCU is an open-source IoT platform that utilizes the ESP8266 Wi-Fi system-on-chip built upon the ESP-12 hardware module. The ESP8266 is a low-cost Wi-Fi microchip with a complete TCP/IP stack and microcontroller capabilities, making it popular for IoT projects. The NodeMCU firmware provides a user-friendly, Lua-based programming environment, allowing developers to create IoT applications easily. Additionally, the hardware design includes a USB-to-serial converter, enabling straightforward programming and debugging. (nodemcu-firmware, 2023) (Espressif Systems, 2020).

The ESP8266 chip is known for its low power consumption and high performance, allowing it to handle various tasks in IoT applications, such as home automation and sensor networks. In the Hostel Management System, the NodeMCU can be utilized as a critical component for various IoT-enabled features:

1. Connect and integrate IoT devices, such as the RFID card reader, LCD, and buzzer, to capture student attendance data.
2. Communicate with the central Hostel Management System server, transmitting attendance data and receiving commands or updates.
3. Implement local processing and decision-making logic, such as triggering alerts or notifications based on attendance events.
4. Provide a Wi-Fi connectivity solution for the hostel's IoT infrastructure, enabling seamless integration and data exchange.

By leveraging the NodeMCU's capabilities, the Hostel Management System can seamlessly incorporate IoT-driven features, such as automated attendance tracking, real-time monitoring, and integration with other hostel-related systems, enhancing the overall effectiveness and efficiency (Espressif Systems, 2020).

4.1.2.3.2. MRC522 RFID Reader:

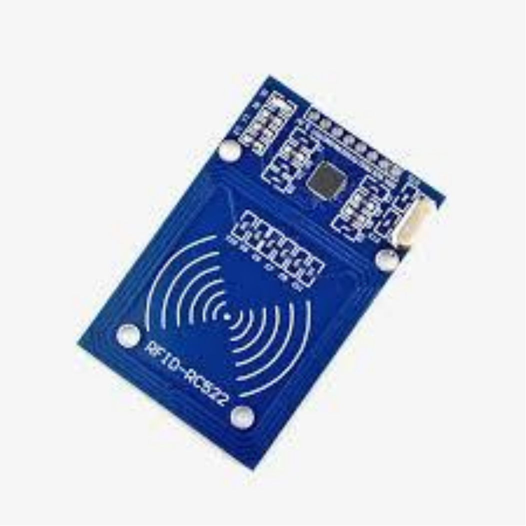


Figure 8: MRC522 RFID Reader

The MRC522 RFID reader is a well-suited choice for the RFID-based attendance system of the Hostel Management System, as evidenced in the provided sources (Maramis & Rompas, 2018). This low-frequency RFID reader module is designed to read from and write data to RFID tags, making it suitable for the system's RFID-based attendance tracking.

Key features of the MRC522 RFID reader that contribute to its suitability for the Hostel Management System include:

1. Low-frequency operation is commonly used for RFID ID cards and tags.

2. Compact and easy-to-integrate design, allowing for convenient placement within the hostels.

3. Support for both read and write operations, enabling the system to identify students and potentially update or modify RFID tag information as needed.

4. Seamless integration with the NodeMCU, the primary IoT platform utilized in the Hostel Management System, through standard communication protocols like SPI or I2C.

By incorporating the MRC522 RFID reader, the Hostel Management System can reliably and efficiently track student attendance and movements, providing a robust and scalable RFID-based attendance solution.

4.1.2.3.3. 20x04 I2C Liquid Crystal Display:

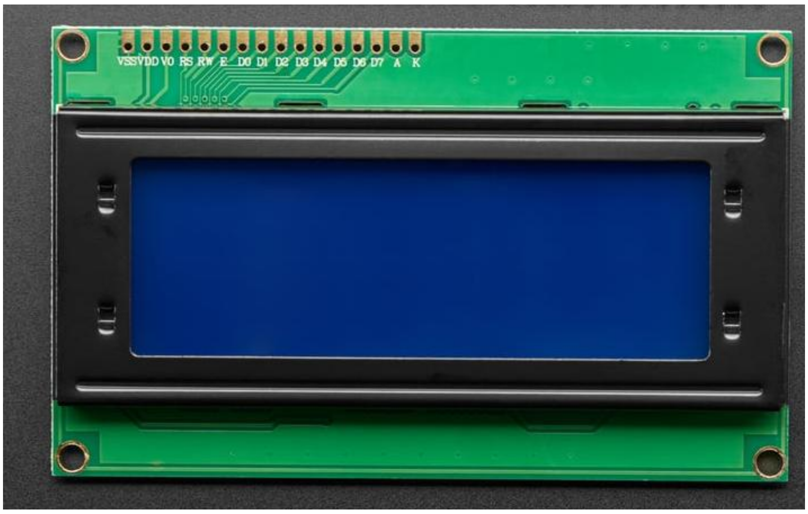


Figure 9: 20x04 I2C Liquid Crystal Display

The 20x4 I2C LCD is a compact and versatile module that enhances the user experience and provides real-time information to students and administrators within the Hostel Management System.

Key features that make the 20x4 I2C LCD well-suited for the system include:

1. A 20-character by 4-line display, offering ample space to present relevant information to users.

2. An I2C communication interface, enabling easy integration with the system's NodeMCU and other IoT components.

3. Low power consumption, making it suitable for energy-efficient IoT applications in the hostel environment.
4. The ability to display various types of information, such as student attendance records, room occupancy status, and system notifications.

By incorporating the 20x4 I2C LCD, the Hostel Management System can offer a user-friendly interface for students and administrators to interact with the system, view real-time information, and receive updates or alerts. The display can be strategically placed in shared areas or at the hostel reception, providing a convenient and easily accessible source of information for all stakeholders. (Geetha et al., 2021)

4.1.2.3.4. Buzzer:

Figure 10: 5V Buzzer

The integration of a buzzer in the Hostel Management System could significantly benefit its functionality. This versatile component can provide valuable audible alerts and notifications, offering several key advantages:

1. The ability to trigger audible alerts for specific events, such as missed attendance check-ins, unauthorized entries, or emergencies, ensures that both students and administrators are promptly notified and can respond accordingly.
2. The buzzer's seamless integration with the system's logic and automation enhances the overall responsiveness and awareness within the hostel environment, allowing the system to react to various trigger conditions.

3. The buzzer can work with other system components, like the LCD, to provide multi-modal notifications, further improving the effectiveness of the Hostel Management System's communication and alert mechanisms.

By leveraging the buzzer's capabilities, the Hostel Management System can enhance its ability to promptly notify and alert students and staff, contributing to a more secure and well-managed hostel environment. (Geetha et al., 2021)

4.1.2.3.5. Bread Board and Jumper Wires:

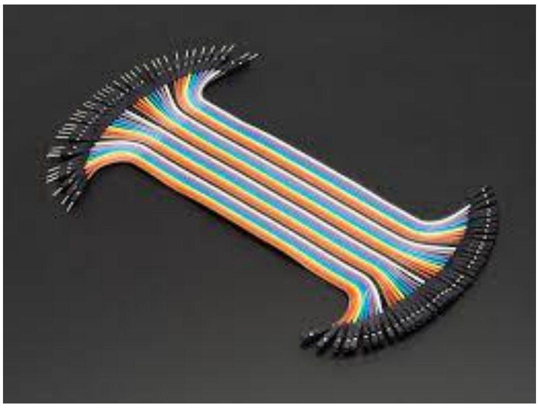


Figure 11: M2F Jumper Wire

Using a breadboard and jumper wires provides a practical and flexible approach to prototyping and testing the Hostel Management System. The breadboard enables the easy connection and reconfiguration of various electronic components, including the NodeMCU, RFID reader, LCD, and buzzer, without needing permanent soldering.

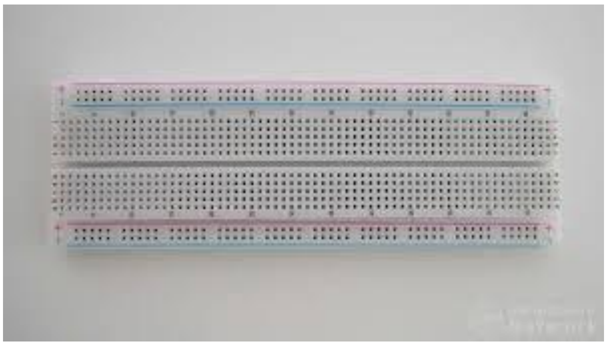


Figure 12: Bread Board

This approach offers several advantages in the development of the Hostel Management System:

1. It enables rapid prototyping and testing of the system's hardware components, allowing for quick modifications and adjustments during development.

2. It provides a reusable and reconfigurable platform, enabling the development team to experiment with different circuit configurations and connections.
3. It helps to quickly identify and troubleshoot any hardware-related issues or connectivity problems in the system.
4. It aids in developing the system's software and firmware, as the hardware can be easily connected and disconnected for programming and testing purposes.

The development team could efficiently test and validate the Hostel Management System's hardware components using a breadboard and jumper wires, leading to a more robust and reliable implementation. (Karki & Bista, 2018)

4.1.2.3.6. RFID Cards and Tags:

Figure 13: RFID Tags & Cards

The Hostel Management System will utilize RFID cards or tags as the primary identification medium for the RFID-based attendance system. RFID cards and tags offer several key advantages that make them well-suited for this application:

1. They provide a convenient and contactless method for students to interact with the RFID readers, streamlining the attendance-tracking process and improving user experience. This contactless approach eliminates the need for physical contact, which can enhance efficiency and hygiene within the hostel environment.
2. Each RFID card or tag has a unique identifier, which the system can use to associate with individual students. This unique identification enables precise attendance monitoring and reporting, allowing the Hostel Management System to accurately track student movements and attendance records.

3. RFID cards and tags are generally cost-effective, making them a practical and scalable solution for deployment across the university's hostel facilities. This technology's cost-effectiveness facilitates the widespread implementation of the RFID-based attendance system throughout the university's hostel network.

4. The low-frequency RFID technology used by the MRC522 reader is well-suited for the hostel management system's proximity-based identification requirements. This technology offers reliable performance and is compatible with the system's overall design and infrastructure (Espressif Systems, 2020).

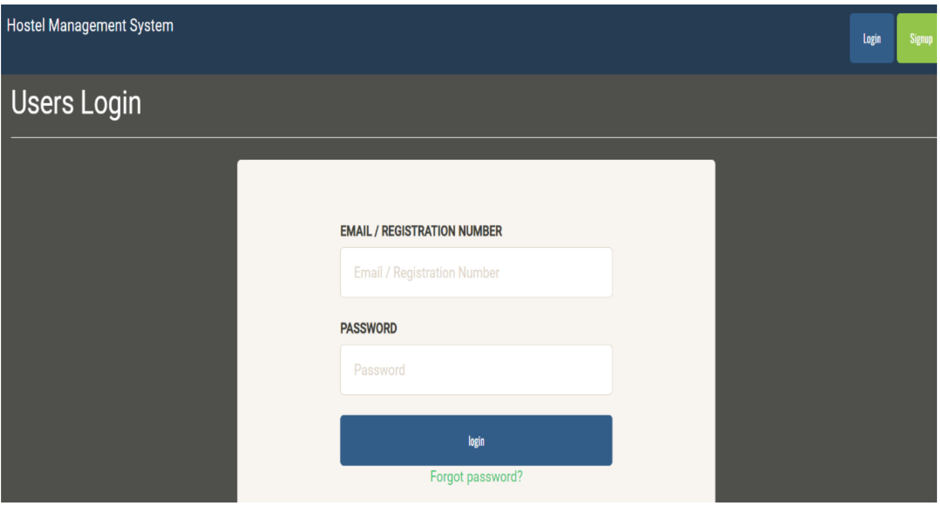
By utilizing RFID cards or tags, the Hostel Management System can reliably and efficiently track student attendance, enhancing the overall effectiveness and efficiency of the hostel management operations. This technology provides a robust and scalable solution for attendance monitoring, contributing to the overall improvement of the university's hostel management capabilities.

Integrating the RFID-based attendance system, the NodeMCU IoT platform, the MRC522 RFID reader, and RFID cards or tags has been a central focus in developing and implementing the Kigali Independent University ULK hostel management system. These components create a comprehensive and automated attendance-tracking solution to enhance room security, optimize administrative processes, and provide valuable data for improved hostel management.

4.1.3. Screen Shorts and Source Codes

ADMIN MODULE

Login Page:



Users Login

EMAIL / REGISTRATION NUMBER

Email / Registration Number

PASSWORD

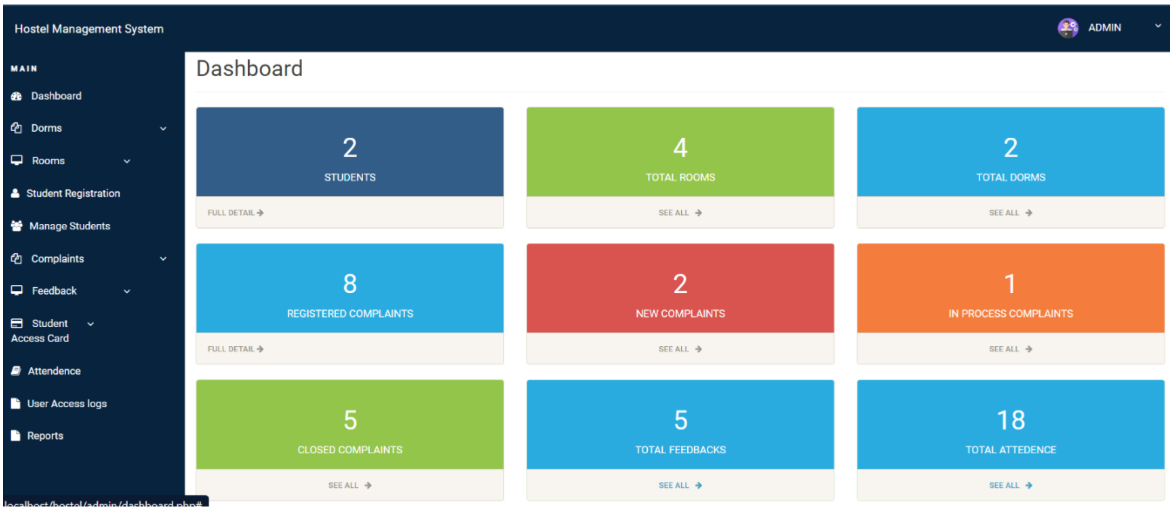
Password

,

Figure 14: Login Page of HMS

The login interface in the figure enables users and admins to access the hostel management system. The main content area features a login form with input fields for the user's email, registration number, and password. Users can enter their authentication details and select the "Login" button to gain access to the system. Furthermore, the "Forgot password?" link offers a method for users to retrieve their password if necessary. The "Signup" button also allows new individuals to create an account.

Admin Dashboard



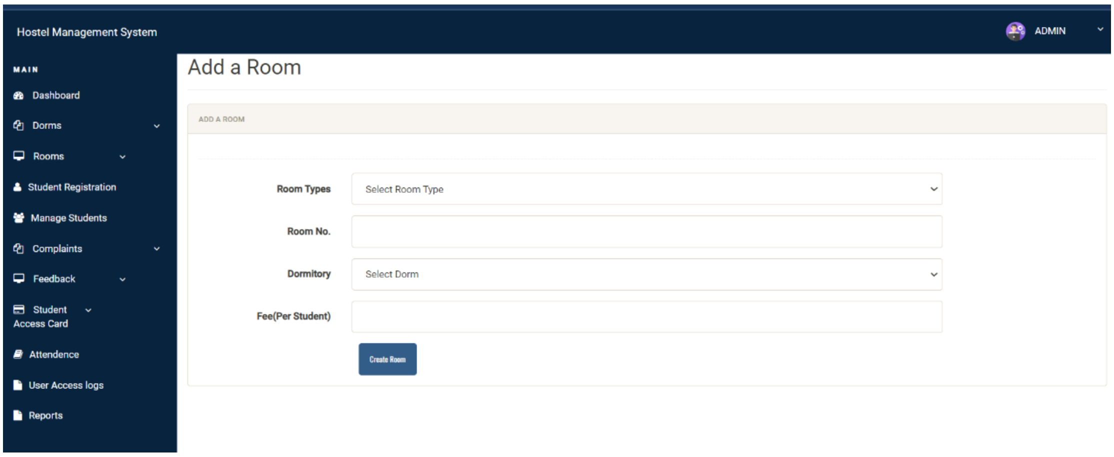
Dashboard

2 STUDENTS — FULL DETAIL →
4 TOTAL ROOMS — SEE ALL →
2 TOTAL DORMS — SEE ALL →
8 REGISTERED COMPLAINTS — FULL DETAIL →
2 NEW COMPLAINTS — SEE ALL →
1 IN PROCESS COMPLAINTS — SEE ALL →
5 CLOSED COMPLAINTS — SEE ALL →
5 TOTAL FEEDBACKS — SEE ALL →
18 TOTAL ATTEDENCE — SEE ALL →

Figure 15: Admin Dashboard of HMS

The hostel management system's administrator dashboard interface displays modules like Dashboard, Dorms, Rooms, Student Registration, Manage Students, Complaints, Feedback, Student Access Card, Attendance, User Access Logs, and Reports. It displays vital indicators like student count, complaints, and attendance. Admin can access additional information by clicking on the “See All" links.

ADD ROOM



Add a Room

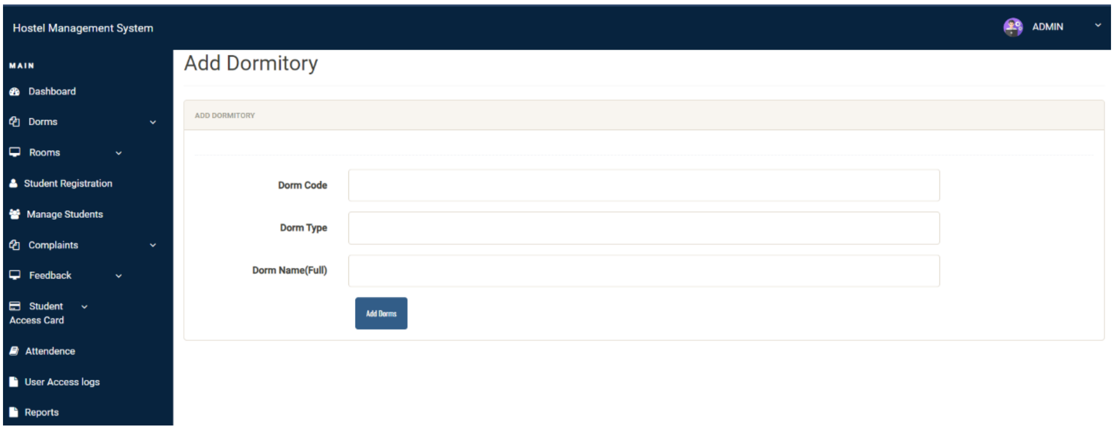
ADD A ROOM

Room Types: Select Room Type
Room No.:
Dormitory: Select Dorm
Fee(Per Student):

Figure 16: Add Room Page of HMS

The figure shows an interface for adding a new room to a hostel management system. It offers access to modules like Dashboard, Dorms, Rooms, Student Registration, and more. The right-hand content area has a form for inputting room details; the administrator can select options and submit the form by clicking the “Create Room” button to create a room.

ADD Dormitory Page



Add Dormitory

ADD DORMITORY

Dorm Code

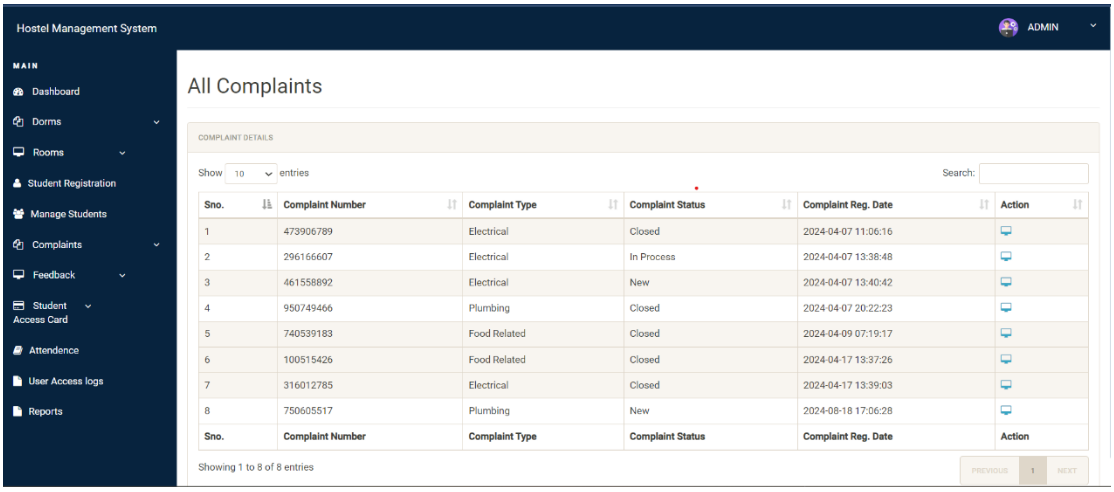
Dorm Type

Dorm Name(Full)

Figure 17: Add Dorm Page of HMS

The figure shows the interface for adding a new dormitory to the HMS. The main content area on the right contains a form where the admin can input details like the Dormitory Code, Type, and Name. After entering the required information, the admin can click the "Add More" button to create the new dormitory.

Manage Complaints Page



All Complaints

COMPLAINT DETAILS

Show 10 entries

Search:

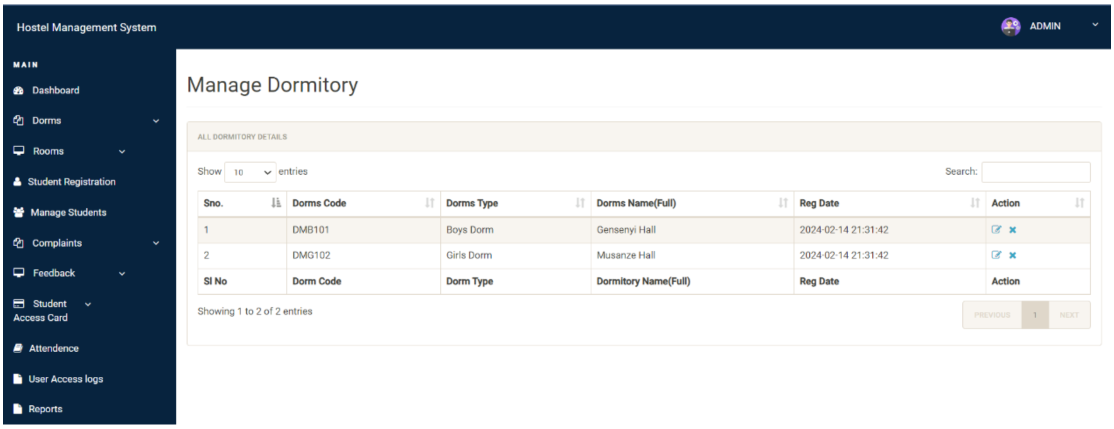
Sno.	Complaint Number	Complaint Type	Complaint Status	Complaint Reg. Date	Action
1	473906789	Electrical	Closed	2024-04-07 11:06:16	🖵
2	296166607	Electrical	In Process	2024-04-07 13:38:48	🖵
3	461558892	Electrical	New	2024-04-07 13:40:42	🖵
4	950749466	Plumbing	Closed	2024-04-07 20:22:23	🖵
5	740539183	Food Related	Closed	2024-04-09 07:19:17	🖵
6	100515426	Food Related	Closed	2024-04-17 13:37:26	🖵
7	316012785	Electrical	Closed	2024-04-17 13:39:03	🖵
8	750605517	Plumbing	New	2024-08-18 17:06:28	🖵
Sno.	Complaint Number	Complaint Type	Complaint Status	Complaint Reg. Date	Action

Showing 1 to 8 of 8 entries

Figure 18: Complaints Management webpage of HMS

The figure illustrates a hostel management system interface for managing complaints. It features a table listing complaints, their serial number, type, status, registration date, and action column. Users can filter complaints by number and search for keywords. The table also shows complaints in different statuses, making complaint management easy.

Manage Dormitory Page



Manage Dormitory

ALL DORMITORY DETAILS

Show 10 entries

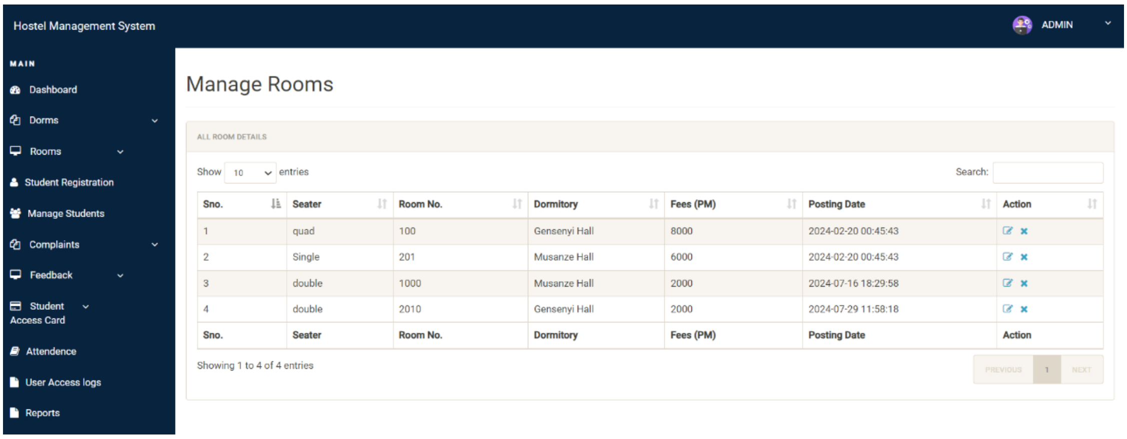
Sno.	Dorms Code	Dorms Type	Dorms Name(Full)	Reg Date	Action
1	DMB101	Boys Dorm	Gensenyi Hall	2024-02-14 21:31:42	✎ ✖
2	DMG102	Girls Dorm	Musanze Hall	2024-02-14 21:31:42	✎ ✖
Sl No	Dorm Code	Dorm Type	Dormitory Name(Full)	Reg Date	Action

Showing 1 to 2 of 2 entries

Figure 19: Dormitory Management Page of HMS

The figure shows an interface for hostel administrators to manage dormitories. It features a table with details like serial number, code, type, full name, registration date, and action column. The administrator can filter, edit, or delete dormitory information.

Manage Room Page



Manage Rooms

ALL ROOM DETAILS

Show 10 entries

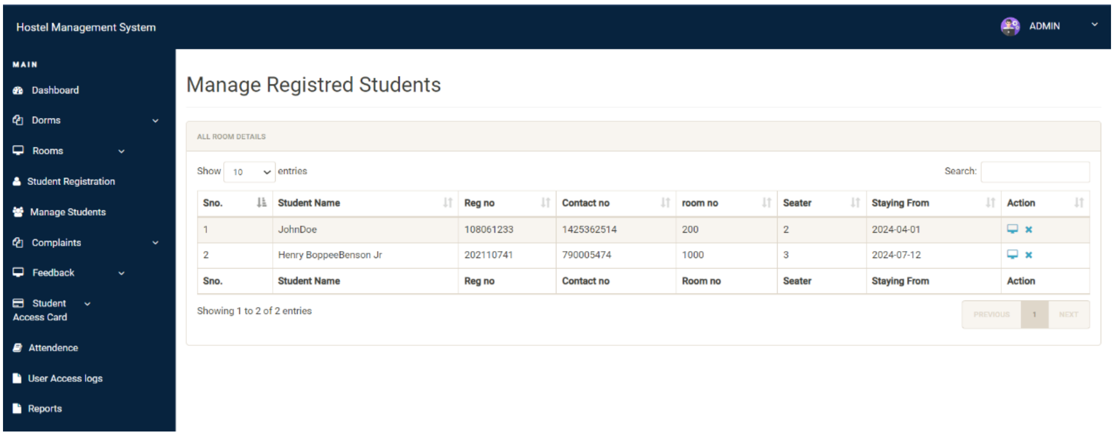
Sno.	Seater	Room No.	Dormitory	Fees (PM)	Posting Date	Action
1	quad	100	Gensenyi Hall	8000	2024-02-20 00:45:43	✎ ✖
2	Single	201	Musanze Hall	6000	2024-02-20 00:45:43	✎ ✖
3	double	1000	Musanze Hall	2000	2024-07-16 18:29:58	✎ ✖
4	double	2010	Gensenyi Hall	2000	2024-07-29 11:58:18	✎ ✖
Sno.	Seater	Room No.	Dormitory	Fees (PM)	Posting Date	Action

Showing 1 to 4 of 4 entries

Figure 20: Manage Room Page of HMS

The figure shows a room management interface in the HMS, featuring a table with room information like serial number, seater type, room number, dormitory, monthly fees, posting date, and action column. Users can refine room listings and view, edit, and delete details within the hostel system.

Manage Student Page



Manage Registred Students

ALL ROOM DETAILS

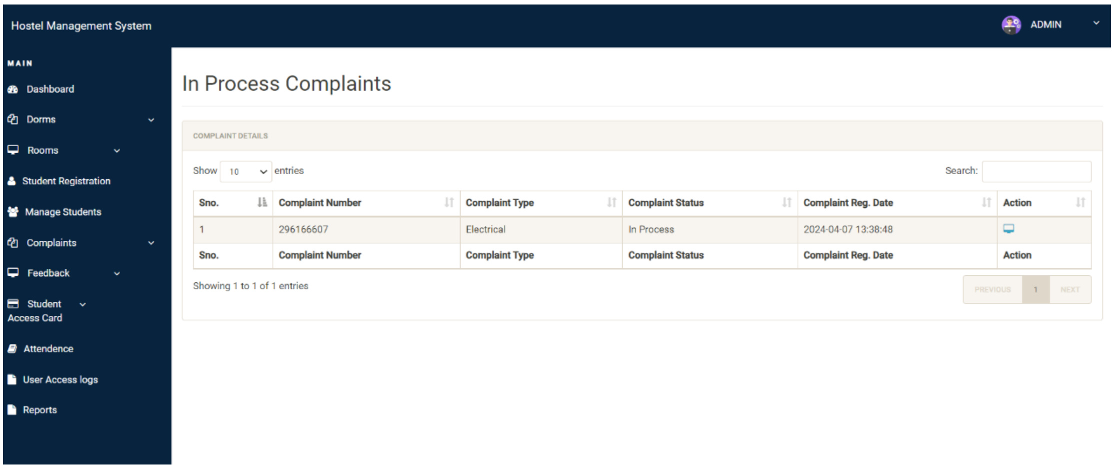
Sno.	Student Name	Reg no	Contact no	room no	Seater	Staying From	Action
1	JohnDoe	108061233	1425362514	200	2	2024-04-01	🖥 ✖
2	Henry BoppeeBenson Jr	202110741	790005474	1000	3	2024-07-12	🖥 ✖
Sno.	Student Name	Reg no	Contact no	Room no	Seater	Staying From	Action

Showing 1 to 2 of 2 entries

Figure 21: Manage Student Webpage of HMS

The figure shows an HMS interface for managing registered students. It features a table with student information like serial number, name, registration number, contact number, room number, seater type, and move-in date. Users can refine listings and view, edit, or delete student records within the hostel system, allowing efficient management.

In Process Complain Page



In Process Complaints

COMPLAINT DETAILS

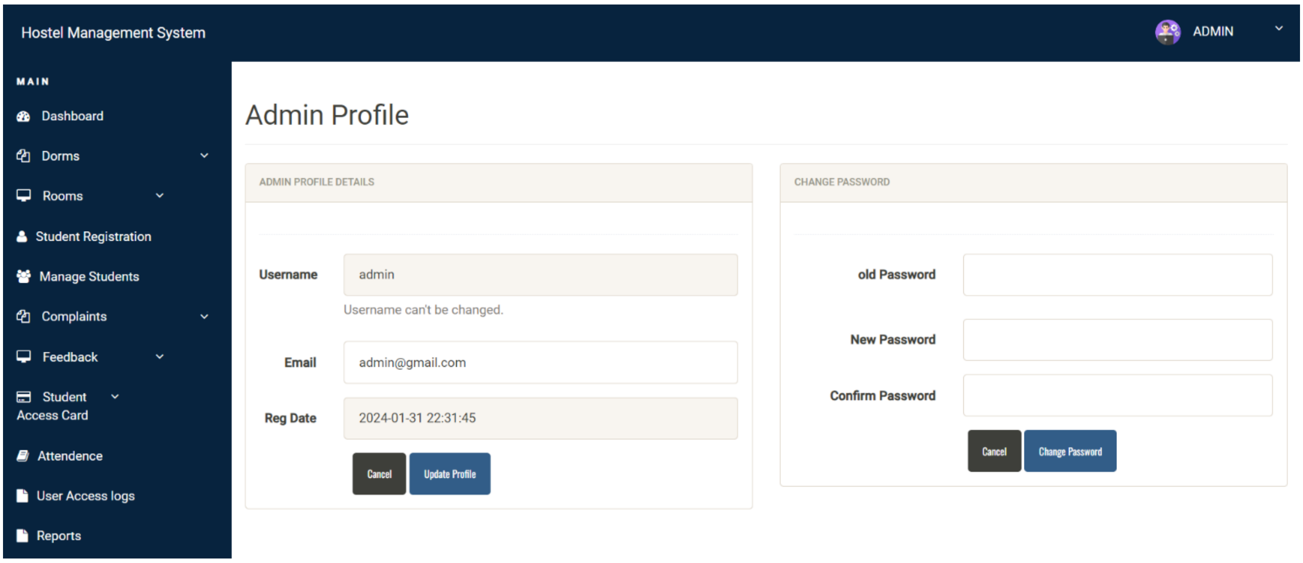
Sno.	Complaint Number	Complaint Type	Complaint Status	Complaint Reg. Date	Action
1	296166607	Electrical	In Process	2024-04-07 13:38:48	🖥
Sno.	Complaint Number	Complaint Type	Complaint Status	Complaint Reg. Date	Action

Showing 1 to 1 of 1 entries

Figure 22: In Process Complaint Webpage of HMS

The figure shows a hostel management system interface for managing pending complaints. It displays complaint details, allows admin to refine listings, and provides an overview of "In Process" complaints, facilitating efficient tracking and administration.

Admin Profile



Admin Profile

ADMIN PROFILE DETAILS

Username: admin

Username can't be changed.

Email: admin@gmail.com

Reg Date: 2024-01-31 22:31:45

CHANGE PASSWORD

old Password

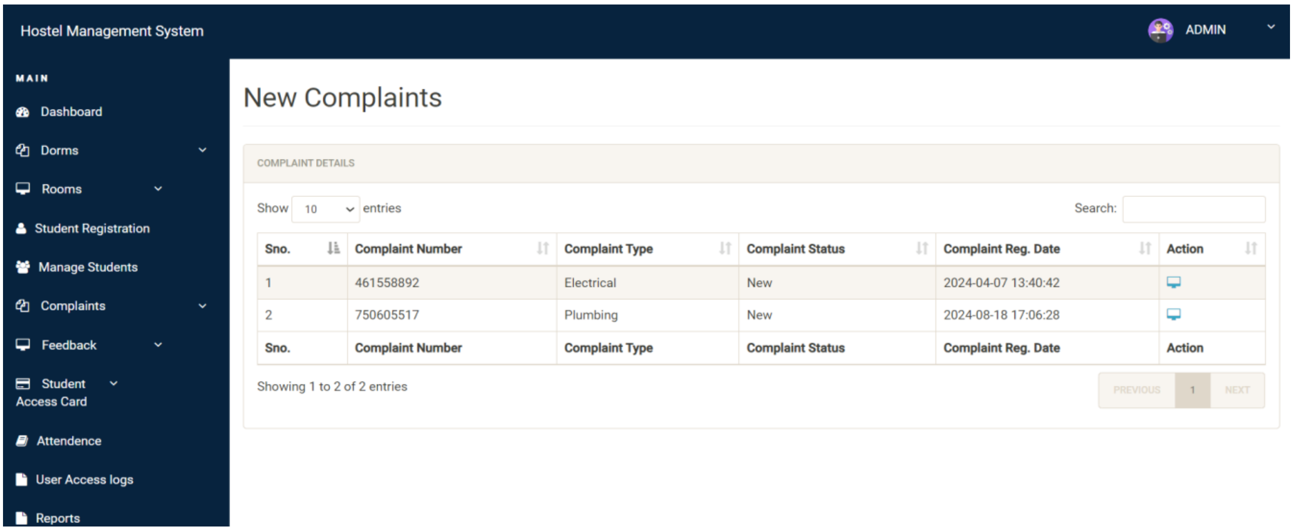
New Password

Confirm Password

Figure 23: Admin Profile Webpage of HMS

The user interface in a hostel management system allows administrators to manage their profile, including their username, email, and registration date. It also allows for password changes, ensuring security and privacy within the system.

New Complaints



New Complaints

COMPLAINT DETAILS

Show 10 entries

Search:

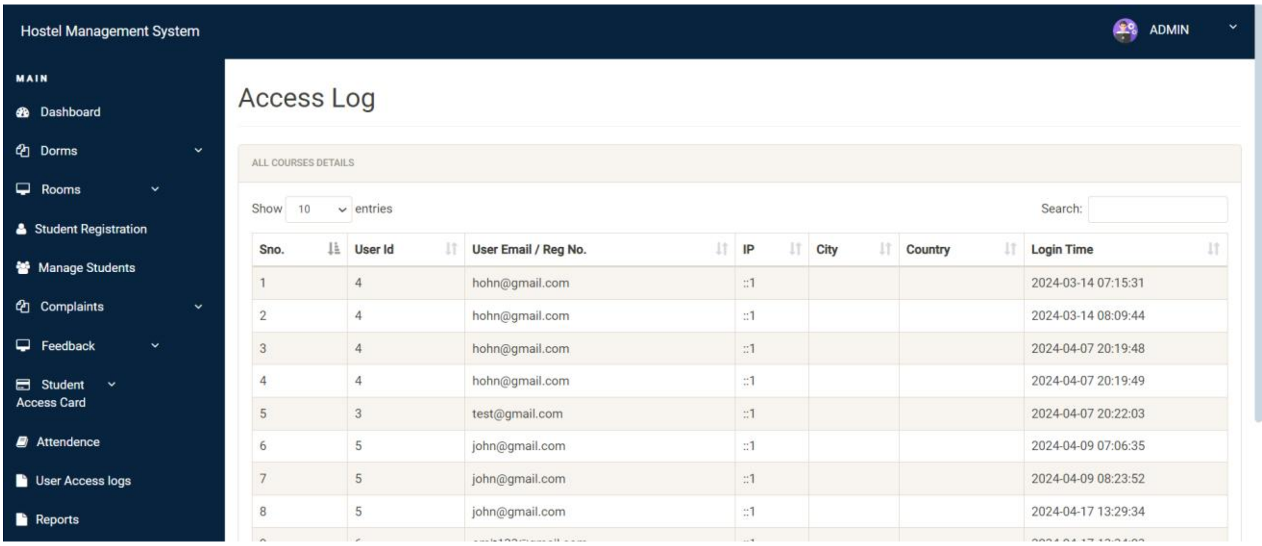
Sno.	Complaint Number	Complaint Type	Complaint Status	Complaint Reg. Date	Action
1	461558892	Electrical	New	2024-04-07 13:40:42	🖥
2	750605517	Plumbing	New	2024-08-18 17:06:28	🖥
Sno.	Complaint Number	Complaint Type	Complaint Status	Complaint Reg. Date	Action

Showing 1 to 2 of 2 entries

Figure 24: New Complaints

The figure depicts an interface for managing new complaints in the HMS, which includes a tabular presentation of complaint facts and allows users to refine listings and follow recent complaints, increasing efficiency in complaint management within the hostel system.

User Access Logs



Access Log

ALL COURSES DETAILS

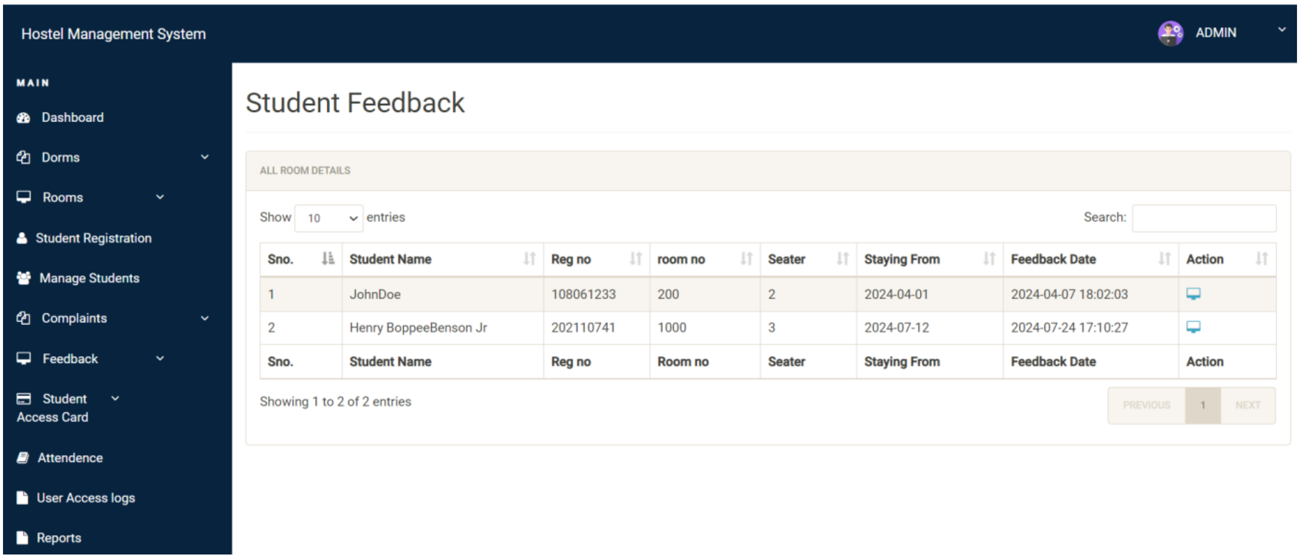
Show 10 entries Search:

Sno.	User Id	User Email / Reg No.	IP	City	Country	Login Time
1	4	hohn@gmail.com	::1			2024-03-14 07:15:31
2	4	hohn@gmail.com	::1			2024-03-14 08:09:44
3	4	hohn@gmail.com	::1			2024-04-07 20:19:48
4	4	hohn@gmail.com	::1			2024-04-07 20:19:49
5	3	test@gmail.com	::1			2024-04-07 20:22:03
6	5	john@gmail.com	::1			2024-04-09 07:06:35
7	5	john@gmail.com	::1			2024-04-09 08:23:52
8	5	john@gmail.com	::1			2024-04-17 13:29:34

Figure 25: User Access Log Webpage of HMS

The figure shows a user interface for hostel management. It displays a table detailing login attempt, serial number, ID, email, IP address, city, country, and login time. Users can refine the logs by selecting entries per page and searching for keywords. This interface allows tracking and monitoring of user activity within the hostel system.

Feedback



Student Feedback

ALL ROOM DETAILS

Show 10 entries Search:

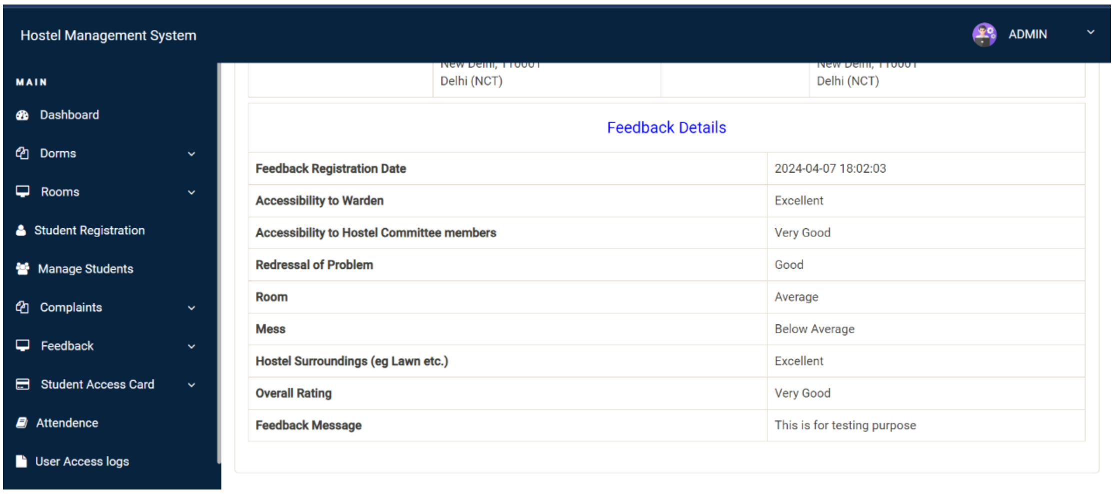
Sno.	Student Name	Reg no	room no	Seater	Staying From	Feedback Date	Action
1	JohnDoe	108061233	200	2	2024-04-01	2024-04-07 18:02:03	🖵
2	Henry BoppeeBenson Jr	202110741	1000	3	2024-07-12	2024-07-24 17:10:27	🖵
Sno.	Student Name	Reg no	Room no	Seater	Staying From	Feedback Date	Action

Showing 1 to 2 of 2 entries

Figure 26: Feedback page of HMS

The figure shows an interface for hostel management, allowing users to manage student feedback. It features a tabular view with details of submissions, refinement options, and total submissions, enabling efficient review and handling within the hostel system.

Feedback Details

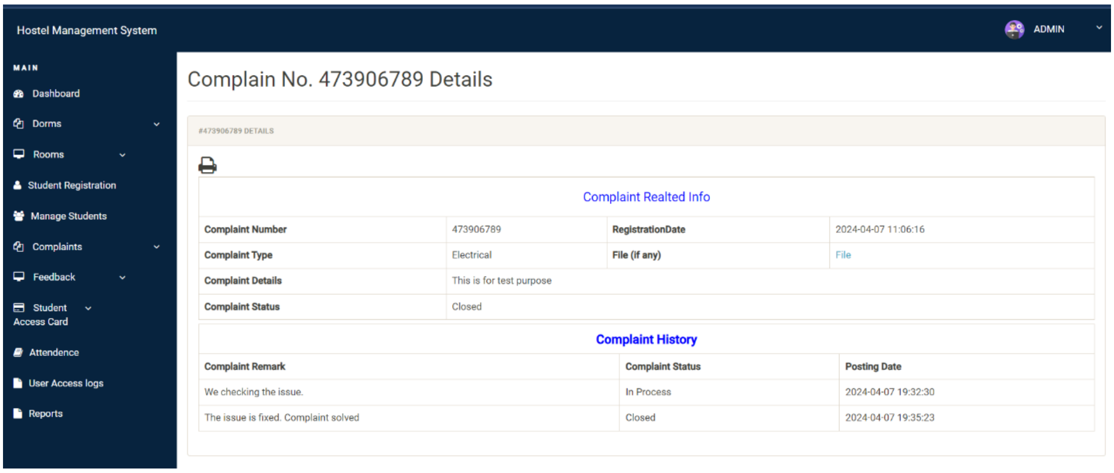


Feedback Details	
Feedback Registration Date	2024-04-07 18:02:03
Accessibility to Warden	Excellent
Accessibility to Hostel Committee members	Very Good
Redressal of Problem	Good
Room	Average
Mess	Below Average
Hostel Surroundings (eg Lawn etc.)	Excellent
Overall Rating	Very Good
Feedback Message	This is for testing purpose

Figure 27: Feedback Details page of HMS

The figure shows a user interface for a hostel management system, displaying feedback categories like Registration Date, Accessibility, Redressal of Problem, Room, Mess, Hostel Surroundings, Overall Rating, and Feedback Message. This helps administrators understand student feedback and identify areas for improvement within the hostel.

Complaint Details



Complain No. 473906789 Details

#473906789 DETAILS

Complaint Realted Info			
Complaint Number	473906789	RegistrationDate	2024-04-07 11:06:16
Complaint Type	Electrical	File (if any)	File
Complaint Details	This is for test purpose		
Complaint Status	Closed		

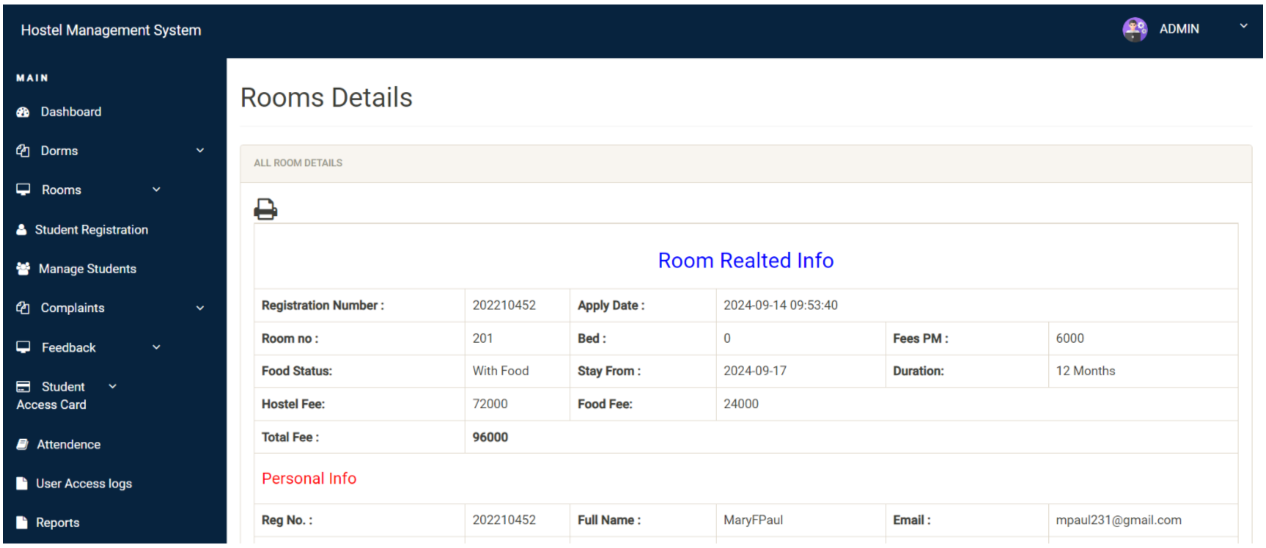
Complaint History

Complaint Remark	Complaint Status	Posting Date
We checking the issue.	In Process	2024-04-07 19:32:30
The issue is fixed. Complaint solved	Closed	2024-04-07 19:35:23

Figure 28: Complaints Details Webpage of HMS

The figure shows a user interface for examining complaint details in a hostel management system. It displays data points like complaint number, type, details, file, status, and history. The Complaint History section tracks complaint statuses and posting dates, allowing users to track complaint progress and resolution timelines.

Student Room Details



Rooms Details

Room Realted Info

Registration Number :	202210452	**Apply Date :**	2024-09-14 09:53:40		
Room no :	201	**Bed :**	0	**Fees PM :**	6000
Food Status:	With Food	**Stay From :**	2024-09-17	**Duration:**	12 Months
Hostel Fee:	72000	**Food Fee:**	24000		
Total Fee :	**96000**				

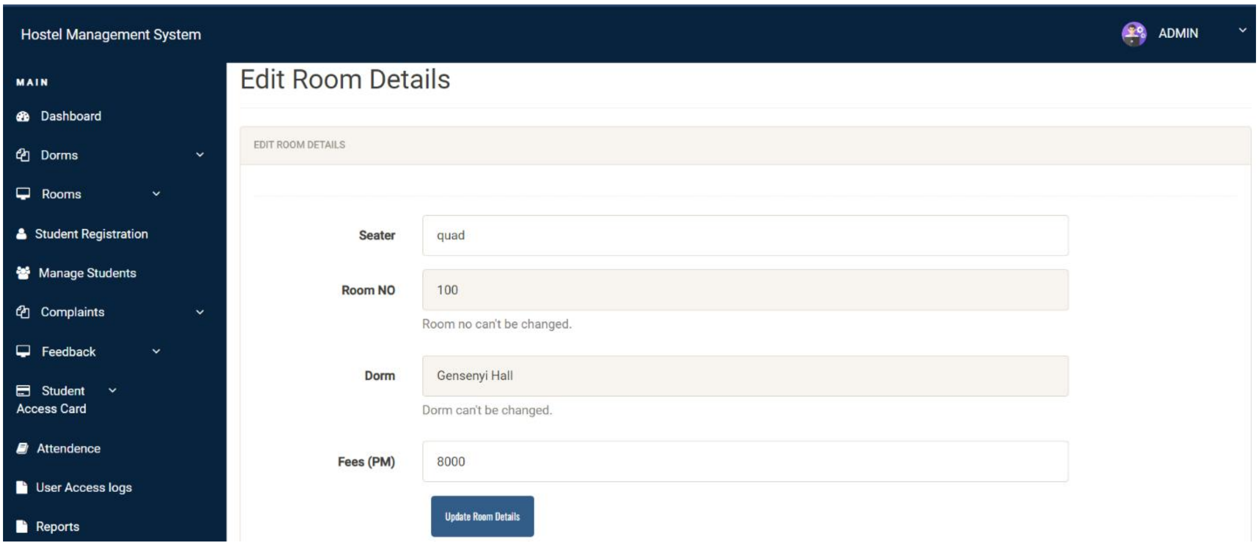
Personal Info

Reg No. :	202210452	**Full Name :**	MaryFPaul	**Email :**	mpaul231@gmail.com

Figure 29: Rooms Details Webpage of HMS

The figure shows a user interface for a hostel management system that displays student room details, including registration number, application date, room status, room number, bed number, floor number, room rent, and due dates. The form also includes sections for personal information and address, providing a comprehensive overview of the room's occupants and associated information.

Edit Room Details



Edit Room Details

Seater: quad

Room NO: 100

Room no can't be changed.

Dorm: Gensenyi Hall

Dorm can't be changed.

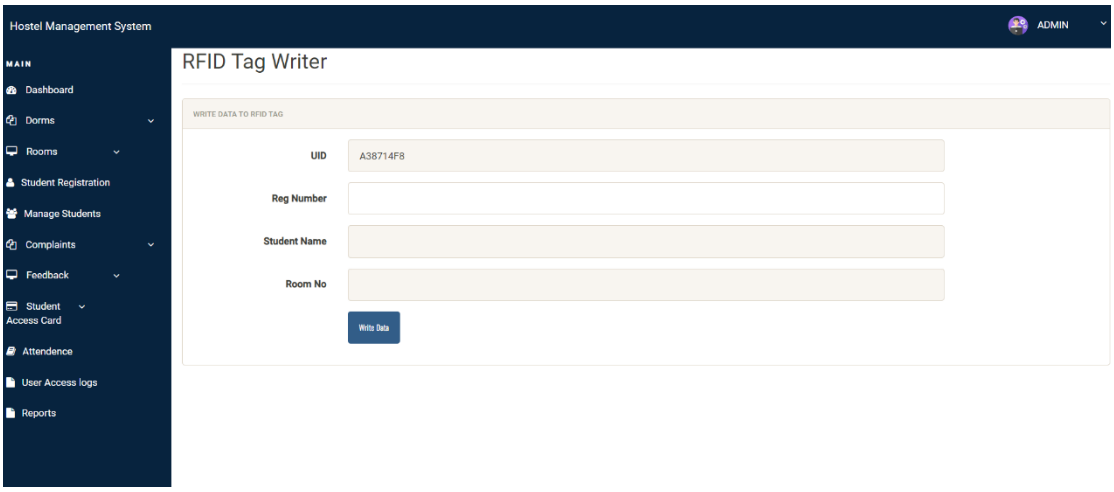
Fees (PM): 8000

Figure 30: Edit Rooms Details Webpage

The figure shows a user interface for hostel management systems that allows administrators to edit room details. The interface includes fields for updating information like Seater, Room No., Dorm, and Fees. Room No. Moreover, Dorm is unchangeable, but users can update the

Seater and Fees fields. This allows administrators to modify aspects like seating capacity and monthly fees.

ADD CARD/ RFID CARD WRITER



RFID Tag Writer

WRITE DATA TO RFID TAG

UID: A38714F8

Reg Number:

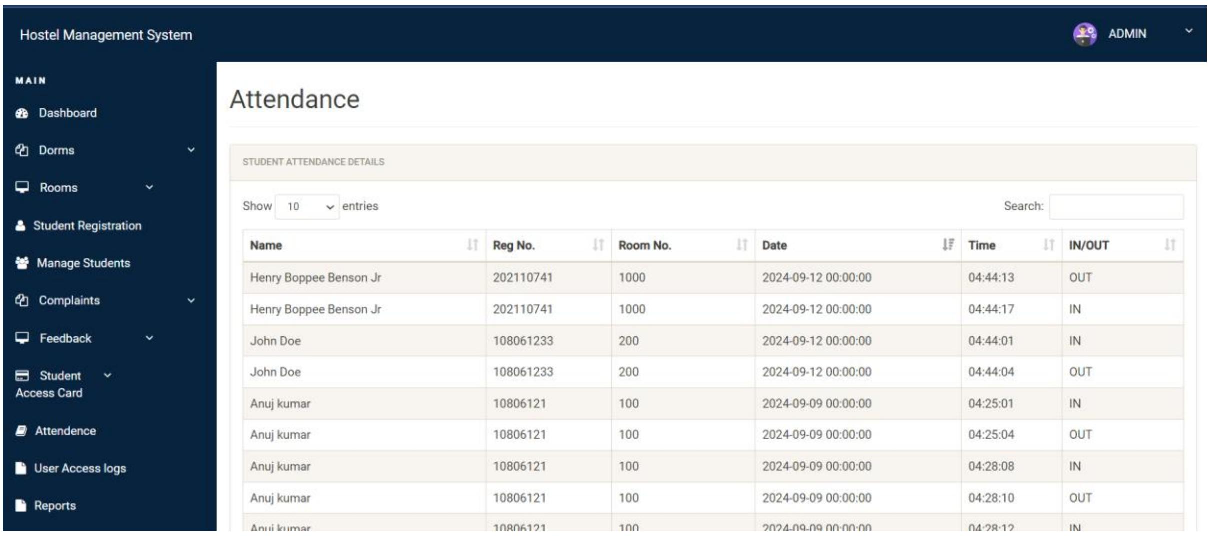
Student Name:

Room No:

Figure 31: RFID Tag Writer

The interface allows hostel administrators to use an RFID tag writer. The administrator scans the student's card on the RFID attendance system to read the unique identifier; once the UID is read, the administrator can enter the student's registration number. If the student has already booked a hostel, the student's name and room number automatically populate the corresponding fields. Conversely, suppose the student has not booked a hostel. In that case, a message stating "This student has not booked a hostel" is displayed, and the administrator can either book a hostel for the student or wait until the student has made a reservation to write a card for them.

Attendance



Attendance

STUDENT ATTENDANCE DETAILS

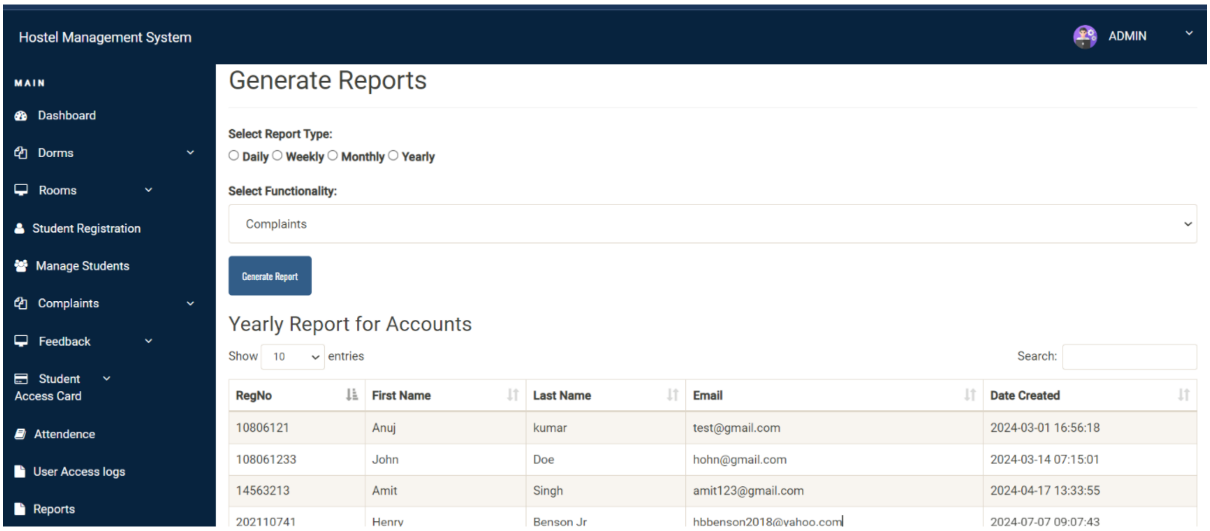
Name	Reg No.	Room No.	Date	Time	IN/OUT
Henry Boppee Benson Jr	202110741	1000	2024-09-12 00:00:00	04:44:13	OUT
Henry Boppee Benson Jr	202110741	1000	2024-09-12 00:00:00	04:44:17	IN
John Doe	108061233	200	2024-09-12 00:00:00	04:44:01	IN
John Doe	108061233	200	2024-09-12 00:00:00	04:44:04	OUT
Anuj kumar	10806121	100	2024-09-09 00:00:00	04:25:01	IN
Anuj kumar	10806121	100	2024-09-09 00:00:00	04:25:04	OUT
Anuj kumar	10806121	100	2024-09-09 00:00:00	04:28:08	IN
Anuj kumar	10806121	100	2024-09-09 00:00:00	04:28:10	OUT
Anuj kumar	10806121	100	2024-09-09 00:00:00	04:28:12	IN

Figure 32: Attendance Webpage of HMS

The figure depicts the attendance interface, part of a comprehensive hostel management system that tracks student attendance using RFID cards. Like the RFID card writer page, this interface integrates with the RFID attendance system, enabling real-time monitoring and recording of student movements within the hostel.

The attendance interface presents a tabular view of student attendance records, displaying details such as the student's name, registration number, room number, date, time, and attendance status. Users can refine the attendance listings by specifying the number of entries to show per page and searching for keywords. Furthermore, the table provides information about the total attendance records currently displayed. This interface empowers administrators to efficiently monitor student attendance patterns, identify discrepancies, and ensure students adhere to hostel rules and regulations.

Reports



Generate Reports

Select Report Type:
○ Daily ○ Weekly ○ Monthly ○ Yearly

Select Functionality:
Complaints

Yearly Report for Accounts

Show 10 entries Search:

RegNo	First Name	Last Name	Email	Date Created
10806121	Anuj	kumar	test@gmail.com	2024-03-01 16:56:18
108061233	John	Doe	hohn@gmail.com	2024-03-14 07:15:01
14563213	Amit	Singh	amit123@gmail.com	2024-04-17 13:33:55
202110741	Henry	Benson Jr	hbbenson2018@yahoo.com	2024-07-07 09:07:43

Figure 33: Reports Webpage

Within the HMS, the Report webpage allows the administrator to create various reports about accounts, complaints, and student information. To create the required report, the user can choose the functionality (student registration, complaints, or accounts) and report type (daily, weekly, monthly, or annual). The report shows a list of students, along with their names, email addresses, registration numbers, and creation dates.

User/ Student MODULE:

Signup

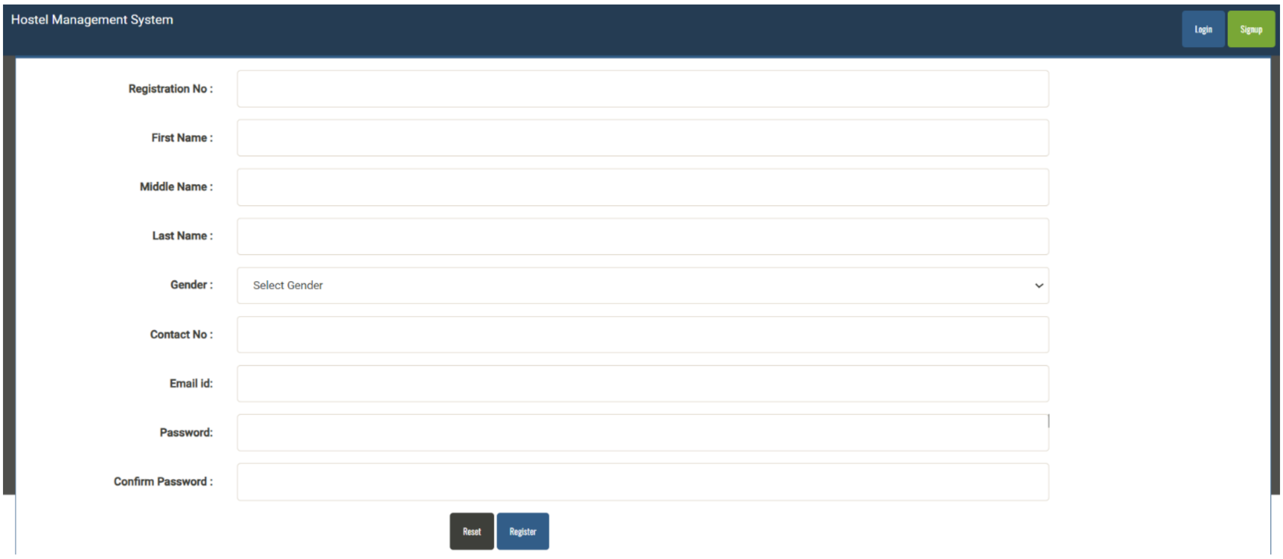


Figure 34: User Signup Webpage of HMS

The figure shows a user interface for student registration in a hostel management system. It includes a form for students to enter their personal information, such as registration number, gender, contact number, email ID, password, and confirmation password, and to create a new account.

User Home Page

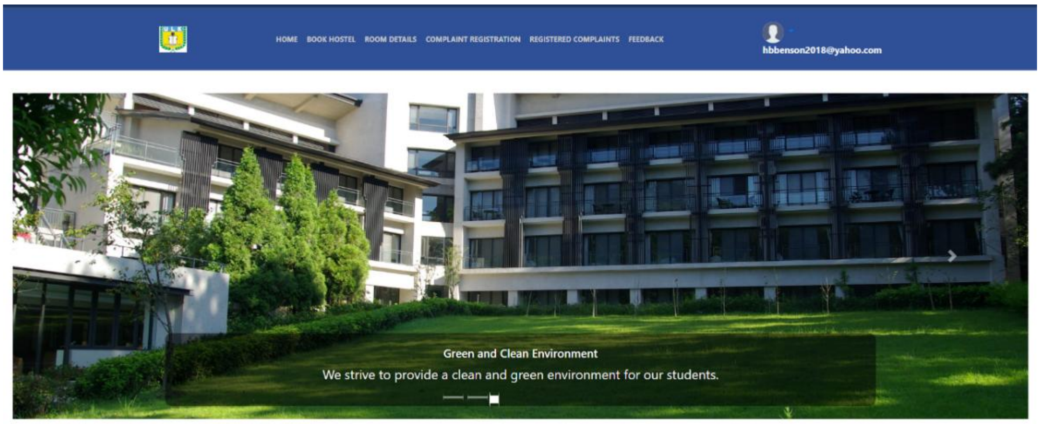
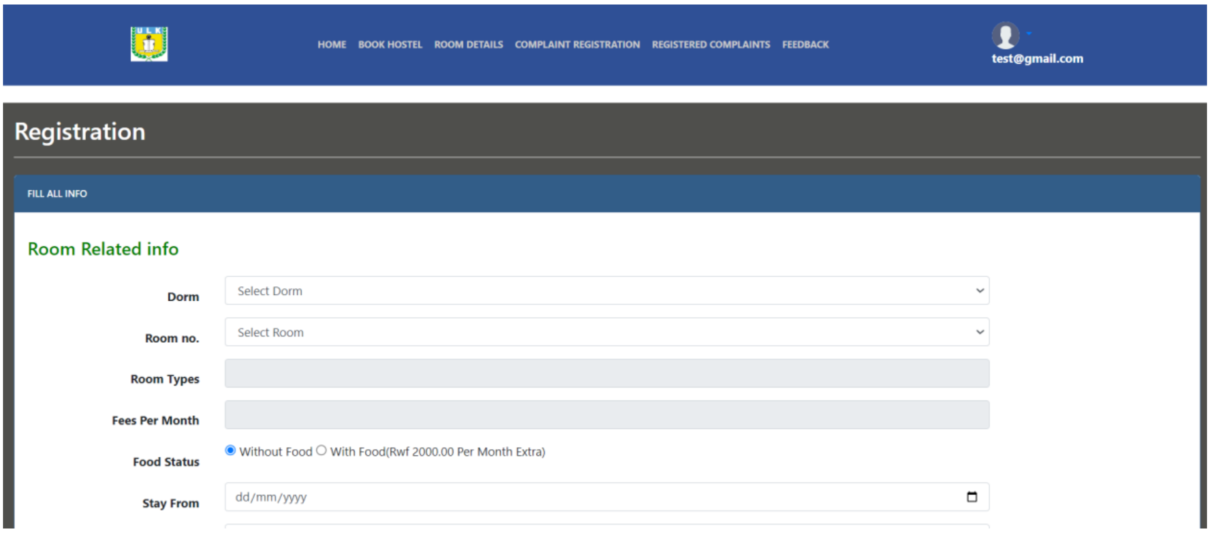


Figure 35: User Homepage of HMS

The hostel management system's homepage features a modern, green building, showcasing its commitment to environmental sustainability. The top navigation bar allows users to access various sections, including Home, Book Hostel, Room Details, Complaint Registration, Registered Complaints, and Feedback.

BOOK Hostel



Registration

FILL ALL INFO

Room Related info

Dorm: Select Dorm

Room no.: Select Room

Room Types:

Fees Per Month:

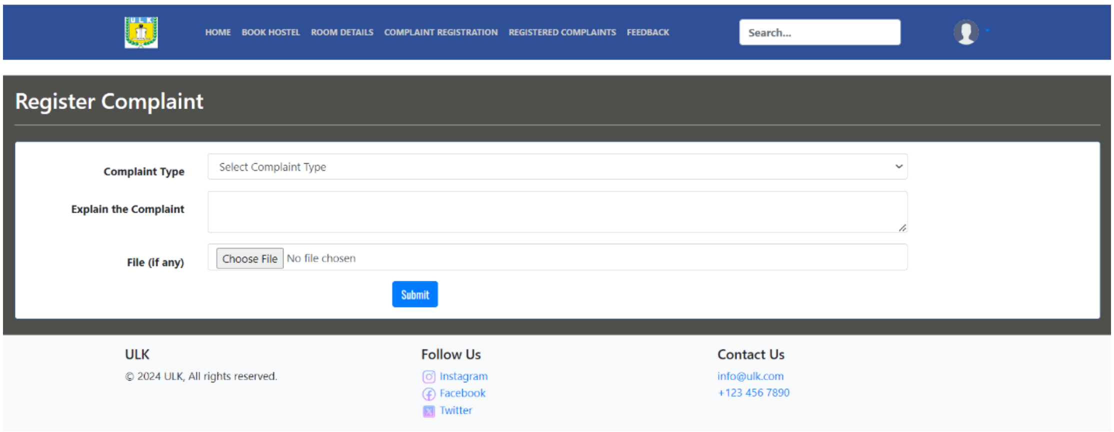
Food Status: ● Without Food ○ With Food(Rwf 2000.00 Per Month Extra)

Stay From: dd/mm/yyyy

Figure 36: Hostel Booking Webpage of HMS

The webpage is likely a hostel registration form. It requires users to fill in personal information such as name, gender, contact details, and emergency contact. The form also collects information related to the room, including dorm selection, room number, monthly fees, food subsidy, stay duration, and room status. The page also includes a search and navigation bar with links to other website sections.

Register Complaint



Register Complaint

Complaint Type: Select Complaint Type

Explain the Complaint:

File (if any): No file chosen

ULK
© 2024 ULK, All rights reserved.

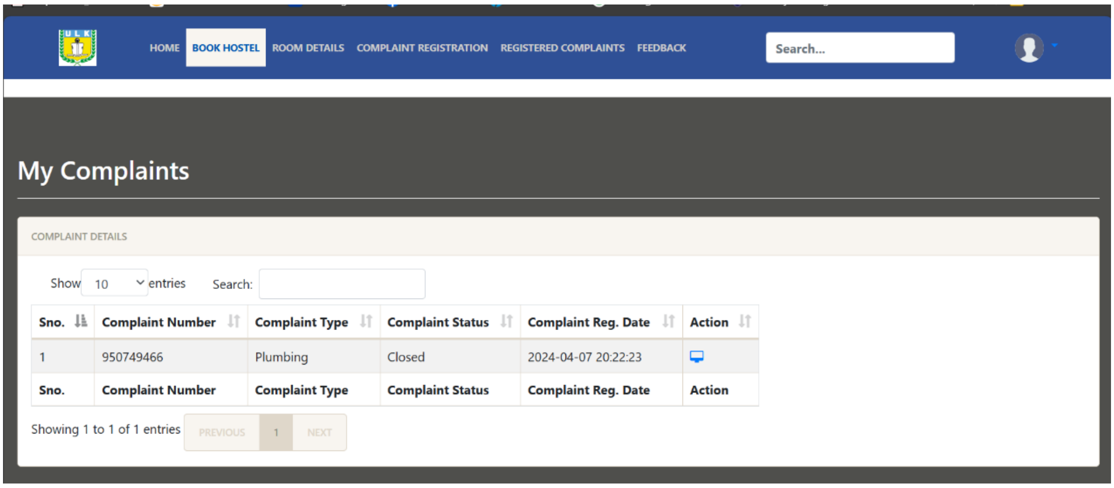
Follow Us
Instagram
Facebook
Twitter

Contact Us
info@ulk.com
+123 456 7890

Figure 37: Complain Registration Webpage of HMS

The webpage is a complaint registration form for a hostel. It enables users to submit complaints by selecting a complaint type, describing the issue, and optionally attaching a file. The page includes a search bar, a navigation bar with links to other website sections, and contact information for the hostel.

Existing Complaints



My Complaints

COMPLAINT DETAILS

Show 10 entries Search:

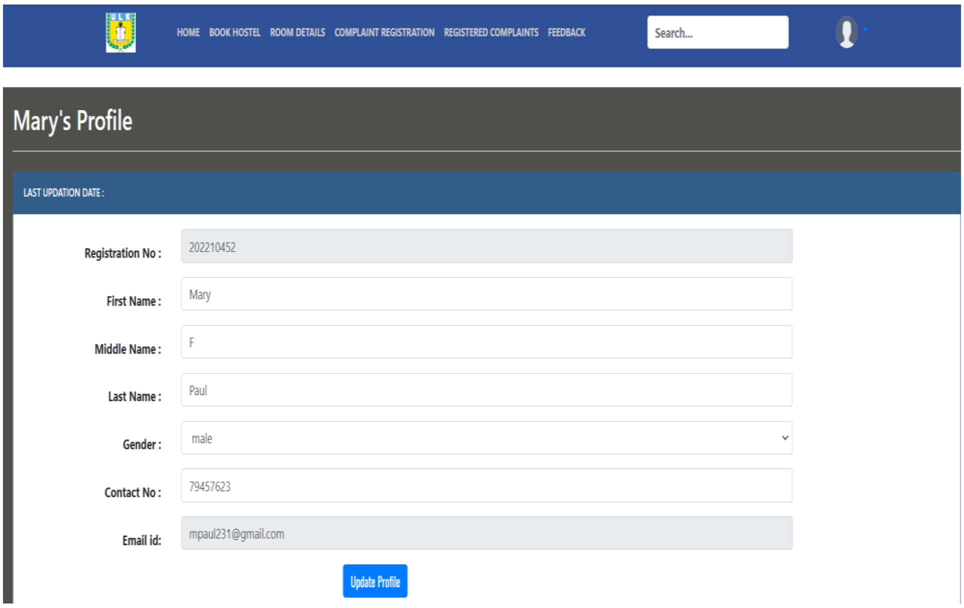
Sno.	Complaint Number	Complaint Type	Complaint Status	Complaint Reg. Date	Action
1	950749466	Plumbing	Closed	2024-04-07 20:22:23	🖥
Sno.	Complaint Number	Complaint Type	Complaint Status	Complaint Reg. Date	Action

Showing 1 to 1 of 1 entries

Figure 38: Registered Complaint Webpage of HMS

The webpage provides a section within the hostel management system where users can view their registered complaints. It presents a table that lists the serial number, complaint number, type, status, registration date, and an action column for each complaint. Users can filter the displayed complaints by selecting the number of entries to show per page and searching for specific keywords. Additionally, the table includes information about the number of complaints currently being shown.

User Profile



Mary's Profile

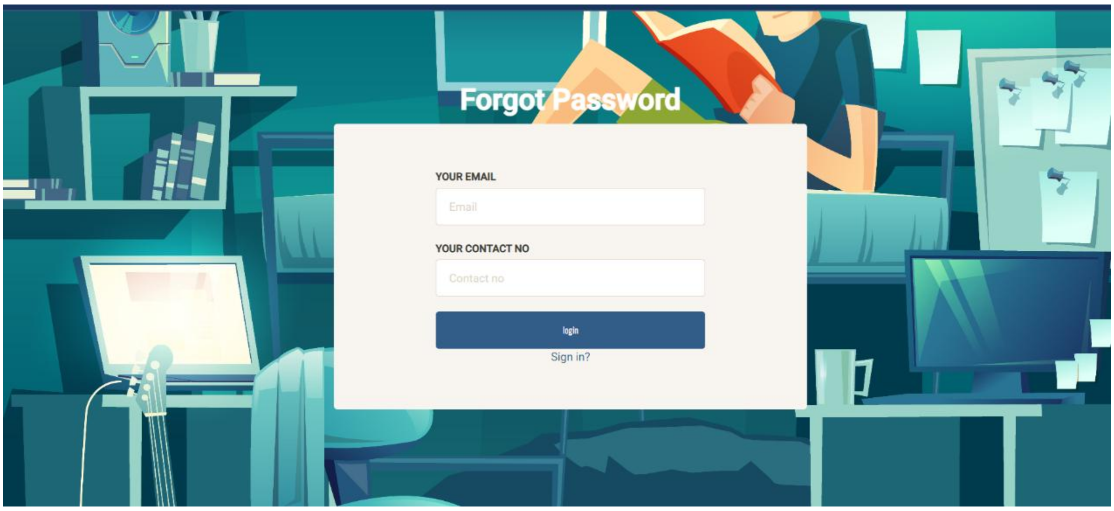
LAST UPDATION DATE :

Registration No :	202210452
First Name :	Mary
Middle Name :	F
Last Name :	Paul
Gender :	male
Contact No :	79457623
Email id:	mpaul231@gmail.com

Figure 39: User Profile Webpage of HMS

This webpage represents a student profile page within the hostel management system. It presents the personal details of a student named Mary, including her registration number, first and last name, middle name, gender, contact information, and email address. The page also displays the "Last Updation Date," indicating when the profile was last modified. Furthermore, an "Update Profile" button allows students to edit their personal information.

Forgot Password



Forgot Password

YOUR EMAIL

Email

YOUR CONTACT NO

Contact no

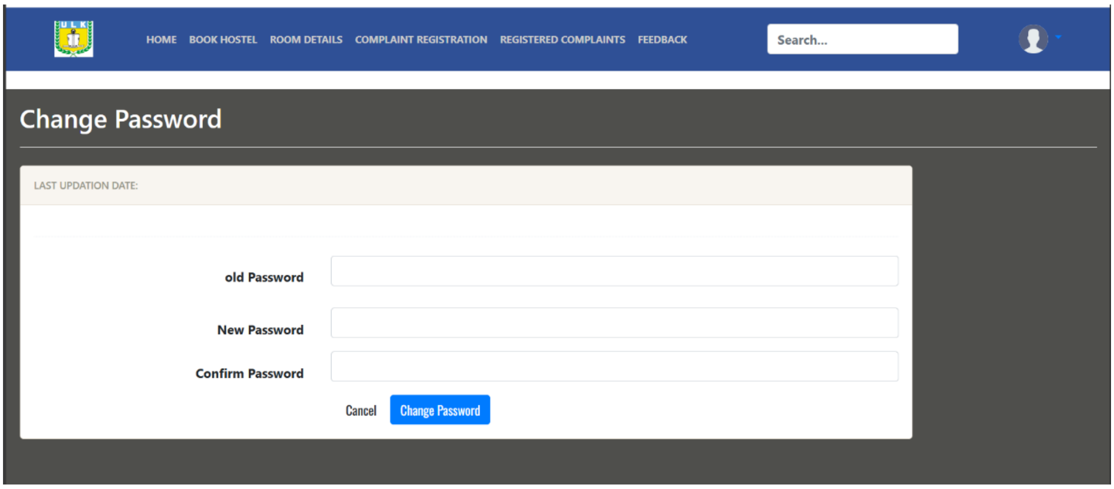
login

Sign in?

Figure 40: Forgot Password page of HMS

This is the hostel management system's password recovery webpage. Users must enter their email addresses and contact numbers to reset their passwords. The page also includes a link to sign in for users who recover their passwords.

Change User Password



HOME BOOK HOSTEL ROOM DETAILS COMPLAINT REGISTRATION REGISTERED COMPLAINTS FEEDBACK

Search...

Change Password

LAST UPDATION DATE:

old Password

New Password

Confirm Password

Cancel Change Password

Figure 41: User Change Password Webpage of HMS

The Hostel Management System offers a password change feature. Users can update their login credentials, display the last update date, and save the change using the "Change Password" button.

4.2. Testing

The Hostel Management System development process included a comprehensive and rigorous testing regimen to ensure the final product's quality, functionality, and reliability. This multifaceted testing approach encompassed unit, integration, user, and performance testing. The unit testing phase verified the individual components and functionalities of the system, while integration testing assessed the overall coherence and seamless operation of the integrated modules. User testing, a particularly user-centric phase, involved engaging student and administrative stakeholders to gather feedback and identify usability or user experience concerns, demonstrating our commitment to meeting their needs. Additionally, performance testing was conducted to evaluate the system's scalability, responsiveness, and reliability under various load conditions. This thorough and systematic testing approach, focusing on user feedback, allowed the development team to identify and address any issues or areas for improvement, ultimately ensuring the Hostel Management System met the university's specific requirements and expectations.

4.2.1. Introductions

Within software engineering, testing is a pivotal component in ensuring software products' reliability, functionality, and overall quality. This process involves systematically examining software components or systems to identify and address any defects, errors, or bugs before deployment. By conducting a series of controlled test cases, software testers strive to validate that the software meets specified requirements, functions as intended, and delivers a satisfactory user experience. The testing process encompasses various techniques, such as unit testing, integration testing, system testing, and acceptance testing, each serving a distinct purpose within the software development lifecycle. Ultimately, testing aims to enhance software products' quality and performance, minimizing risks and maximizing user satisfaction.

The testing phase was a critical component of the system implementation process for the Hostel Management System developed for the Kigali Independent University.

4.2.2. Unit Testing Output

Table 15: Unit Testing Table

Test Case ID	Description	Input	Expected Output	Actual Output	Result
TC001	Validate user sign-up with valid details	Username, password, email, valid details	Account created, success message displayed	Account created, success message displayed	Pass
TC002	Validate user sign-up with existing username	Existing username, other valid details	Error message: "Username already exists."	Error message displayed	Pass
TC003	Validate user sign-in with correct credentials	Correct username and password	The user signed in, redirected to the dashboard	The user signed in, redirected	Pass
TC004	Validate user sign-in with incorrect credentials	Incorrect username or password	Error message "Invalid credentials"	Error message displayed	Pass
TC005	Validate the password recovery process	Registered email	Password reset link sent to email	Link sent to the registered email	Pass
TC006	Validate session timeout after inactivity	User inactive for 15 minutes	Auto sign-out, redirect to login page	The user signed out, redirected to the login	Pass
Hostel Booking					
TC007	Validate hostel room booking with valid details	Room type, dates, payment details	Room booked; confirmation message displayed	Room booked; confirmation displayed	Pass
TC008	Validate booking with unavailable room type	Room type unavailable	Error message: "Room type unavailable."	Error message displayed	Pass
Complaint Management					
TC010	Validate complaints with valid details	Complaint details, user ID	Complaint lodged, confirmation message displayed	Complaint lodged, confirmation displayed	Pass
TC011	Validate viewing complaint status	Valid complaint ID	Complaint status displayed	Complaint status displayed	Pass
User Feedback					
TC012	Validate submitting user feedback	Feedback details, rating	Feedback submitted, thank you message displayed	Feedback submitted, thank you message displayed	Pass
TC013	Validate viewing past feedback	User ID, feedback history request	Feedback history displayed	Feedback history displayed	Pass
RFID Attendance System					
TC014	Validate successful RFID scan and	Valid RFID tag, student present	Display student name, "In"	Student name displayed, "In"	Pass

	attendance record		status, record inserted in DB	status shown, record inserted	
TC015	Validate unsuccessful RFID scan with an invalid tag	Invalid RFID tag	Display error message, buzzer sound, no record inserted	Error message displayed, buzzer sounded, no record inserted	Pass
TC016	Validate duplicate scans within a short time	The same valid RFID tag was scanned within 1 minute	Display error: No duplicate record inserted	Error message displayed; no duplicate record	Pass
TC017	Validate "Out" status when the student exits	Valid RFID tag, student exits	Display student name, "Out" status, record inserted in DB	Student name displayed, "Out" status shown, record inserted	Pass
TC018	Validate RFID system response when the database is down	Valid RFID tag, DB disconnected	Display error message, no record inserted	Error message displayed, no record inserted	Pass
TC019	Validate student presence after attendance is recorded	Valid RFID tag, student present, status "In"	Display student name, "In" status	Student name displayed, "In" status shown	Pass
TC020	Validate system handling of multiple students	Multiple valid RFID tags scanned sequentially	Correct student names and statuses displayed, records inserted in DB	Correct names and statuses displayed, records inserted	Pass
TC021	Validate handling of missing RFID tag	No RFID tag scanned	No display, no buzzer, no record inserted	No display, no buzzer, no record inserted	Pass
TC022	Validate handling of incorrect student details	Valid RFID tag, incorrect details in DB	Display error message, no record inserted	Error message displayed, no record inserted	Pass
TC023	Validate overall system performance under load	100 RFID tags scanned in quick succession	All records inserted, correct statuses displayed	All records inserted, statuses displayed correctly	Pass

4.2.3. Validation Testing Output

Table 16: Validation Testing Table

Test Case ID	Test Case Description	Expected Result	Actual Result	Status
TC01	User Sign Up	A user account was created successfully	The user account was created successfully	**Pass**
TC02	User Sign In	The user logged in successfully	The user logged in successfully	**Pass**
TC03	Booking Hostel	The hostel room was booked successfully	The hostel room was booked successfully	**Pass**
TC04	Lodging Complaint	Complaint lodged successfully	Complaint lodged successfully	**Pass**
TC05	User Feedback	Feedback submitted successfully	Feedback submitted successfully	**Pass**
TC06	RFID Attendance - Check In	RFID tag scanned, student checked in	RFID tag scanned, student checked in	**Pass**
TC07	RFID Attendance - Check Out	RFID tag scanned; student checked out	RFID tag scanned, student checked out	**Pass**
TC08	RFID Attendance - Unauthorized Access	Unauthorized RFID tag scanned, access denied	Unauthorized RFID tag scanned, access denied	**Pass**
TC09	RFID Attendance - System Downtime	System downtime simulated; attendance not recorded	System downtime simulated; attendance not recorded	**Pass**
TC10	RFID Attendance - Data Sync	Attendance data synced with a central database	Attendance data synced with a central database	Pass

4.2.4. Integration Testing Output

Table 17: Integration Testing Output

Test Case ID	Test Case Description	Expected Output	Actual Output	Status
TC01	User Registration	Successful account creation with confirmation email sent	Account created, confirmation email received	Pass
TC02	User Login	Successful login with valid credentials	The user logged in successfully	Pass
TC03	Hostel Booking	Successful booking of a hostel room	Booking confirmed, room assigned	Pass
TC04	Hostel Payment	Successful payment for booked hostel	Payment processed, booking status updated	Pass
TC05	Lodging Complaint	Complaint submitted successfully	Complaint logged, the response sent to a user	Pass
TC06	User Feedback	Feedback submitted successfully	Feedback recorded	Pass
TC07	RFID Card Registration	Successful registration of RFID card to user account	Card registered, linked to user	Pass
TC08	RFID Attendance Tracking (In)	Successful tracking of students entering the hostel using RFID card	Attendance recorded, user status updated to 'In'	Pass
TC09	RFID Attendance Tracking (Out)	Successful tracking of students leaving the hostel using RFID card	Attendance recorded, user status updated to 'Out'	Pass
TC10	RFID Access Control	Successful access control based on RFID card authorization	Authorized user granted access, unauthorized user denied access	Pass
TC11	Integration with Hostel Management System	Successful integration of RFID data with the hostel management system	RFID data is updated in the system, and reports are generated	Pass

4.2.5. Functional and System Testing

Table 18: Functional Testing Output

Test Case	Description	Expected Outcome	Actual Outcome	Status
Booking of Hostel	Test if students can book a hostel room	The room should be successfully booked and recorded	The room can be successfully booked and recorded	**Pass**
Logging of Complaints	Test if students can log complaints	Complaints should be recorded in the system	Complaints are recorded in the system	**Pass**
Feedback Submission	Test if students can submit feedback	Feedback should be successfully recorded	All feedbacks are successfully recorded	**Pass**
Writing Student Data on RFID Cards	Test if student data can be written on RFID cards	Data should be correctly written and retrievable	Data can be correctly written and retrievable	**Pass**
Attendance Tracking with RFID	Test if the RFID attendance system tracks entry/exit correctly.	Attendance should be logged with the correct details.	Attendance can be logged with the correct details.	**Pass**

System Testing Output:

Table 19: System Testing Output

Test Case	Description	Expected Outcome	Actual Outcome	Status
System Integration	Test if all components (booking, complaints, feedback, RFID) work together seamlessly	All components should integrate and function correctly.	All components are integrated and functioning correctly.	**Pass**
RFID System Functionality	Test if the RFID reader correctly identifies cards and updates attendance	UID should be read and attendance recorded accurately	UID is being read, and attendance is recorded accurately	**Pass**
Data Accuracy	Test if data (bookings, complaints, feedback, RFID data)	Data should be accurately stored and retrievable.	Data is accurately stored and retrievable.	**Pass**

	is stored and retrieved accurately			
Web Interface Responsiveness	Test if the web interface responds correctly to user inputs	Web pages should load and respond without errors	Web pages can load and respond without errors	**Pass**
Security and Access Control	Test if the system secures sensitive data and restricts unauthorized access.	Data should be protected, and access should be controlled.	Data are protected and accessed by authorized users.	**Pass**
Performance and Load Testing	Test if the system performs well under heavy load	The system should handle multiple users and data without significant delays	The system can handle multiple users and data without significant delays	**Pass**

4.2.5. Acceptance Testing Report

Table 20: Acceptance Testing Report

Test Case	Description	Acceptance Criteria	Test Results
Booking of Hostel	Test if students can book a hostel room	Students should be able to book rooms without errors, and bookings should be correctly recorded in the system.	**Pass**
Logging of Complaints	Test if students can log complaints	Complaints should be logged correctly, and students should receive confirmation.	**Pass**
Feedback Submission	Test if students can submit feedback	Feedback should be successfully submitted and recorded, with appropriate confirmation.	**Pass**
Writing Student Data on RFID Cards	Test if student data can be written on RFID cards	Data should be correctly written to RFID cards and	**Pass**

		readable by the system	
Attendance Tracking with RFID	Test if the RFID attendance system tracks entry/exit correctly	Attendance should be recorded accurately with correct timestamps and student details	**Pass**
System Integration	Test if all functionalities work together seamlessly	All components (booking, complaints, feedback, RFID) should operate together without issues	**Pass**
Web Interface Usability	Test if the web interface is user-friendly and intuitive	The web interface should be easy to navigate and use, with clear instructions and feedback	**Pass**
Data Accuracy and Consistency	Test if data is accurately maintained and consistent across the system	Data should be consistent and accurate across all functionalities (booking, complaints, feedback, RFID)	**Pass**
Performance Under Load	Test if the system performs well under heavy load	The system should handle multiple simultaneous users and data without significant delays or errors	**Pass**
Security and Access Control	Test if the system secures sensitive data and restricts unauthorized access.	Sensitive data should be protected, and unauthorized access should be prevented.	**Pass**

CONCLUSION AND RECOMMENDATION

5.1. Conclusion

The development and implementation of a web-based hostel management system at ULK will demonstrate the significant benefits that higher education institutions can reap in improving the quality of student accommodation and overall campus life. The university must streamline student housing management by leveraging technology, including the application process, room allocation, and maintenance tracking. The web-based system will enable more efficient data management, enhance communication between students and hostel administrators, and provide valuable insights to inform decision-making.

The findings from the ULK case study stand on their own and align seamlessly with the existing literature on implementing web-based hostel management systems in higher education. The improved accessibility, transparency, and data-driven decision-making will increase student satisfaction and better utilization of available resources. Furthermore, the web-based system will facilitate the monitoring and reporting key performance indicators, allowing the university to continuously evaluate and enhance the quality of its student accommodation services. This web-based approach will be a highly effective solution for improving the overall management and quality of student housing, a critical aspect of the university experience for many students.

5.2. Recommendations

Based on the findings and insights gained from this research, the following recommendations are proposed for the continued development and improvement of the web-based hostel management system at Kigali Independent University, ULK:

1. Continuous system maintenance and updates: To ensure the system's longevity and relevance, ULK must dedicate resources to regularly maintaining and updating it, addressing technical issues, and incorporating new features. This commitment to maintenance will ensure the system's reliability and continued effectiveness.

2. Integration with other university systems: To further enhance the system's functionality and provide a more holistic experience for students, the web-based hostel management system should be integrated with other university systems, such as the student information system and the financial management system

3. User training and support: Students and hostel administrators should receive ongoing user training and support to ensure they are comfortable navigating and utilizing the system effectively.

4. Expansion to other university campuses: If the web-based hostel management system proves successful at the ULK main campus, the university should consider expanding the system to its other campuses, ensuring a consistent and streamlined experience for all students

5. Regular system evaluation and feedback: ULK should implement a process for regularly evaluating the performance and effectiveness of the web-based hostel management system, incorporating user feedback to identify areas for improvement and enhancement.

By implementing these recommendations, Kigali Independent University ULK can further strengthen the web-based hostel management system and continue to provide a high-quality experience for its students and hostel administrators

5.3. Future Works

The current study has focused on developing and implementing a web-based hostel management system at Kigali Independent University ULK. However, several promising avenues for future research and exploration could build upon its findings.

First, it would be worthwhile to investigate the feasibility and potential challenges of integrating the hostel management system with other university-wide systems, such as the student information and financial management systems. This integration could facilitate a more seamless and comprehensive experience for students and administrators, enabling better data sharing, streamlined processes, and improved decision-making.

Furthermore, a comparative analysis of web-based hostel management systems across different higher education institutions within Rwanda and internationally could provide valuable insights into best practices, familiar challenges, and innovative solutions. Such an analysis could help identify the most effective strategies and features for web-based hostel management, enabling Kigali Independent University ULK to enhance its system further.

Finally, exploring the potential of emerging technologies, such as artificial intelligence and other advanced IoT systems, for example, adding other components to the existing RFID Attendance system used in this project, will enhance the functionality and user experience of web-based hostel management systems and could be a fruitful area of investigation. These advanced technologies could enable more intelligent automation, predictive analytics, and personalized services, improving the overall efficiency and effectiveness of the hostel management system.

By addressing these future research directions, Kigali Independent University ULK and other higher education institutions can continue to advance the development and implementation of web-based hostel management systems, enhancing the quality of student accommodation and campus life.

REFERENCES

1. Jafrudin, J., & Putra, Y H. (2020, July 1). “Innovation Development of Web-Based Dormitory Information System at Boarding School SMA Terpadu Krida Nusantara.” https://doi.org/10.1088/1757-899x/879/1/012041
2. Khamis, A., Mohammed, D., Yahya, A., & Pandey, J. (2020, June 1). “A Proposed Model based on Modern Requirements to Optimize Hostel Resources in Oman.” https://doi.org/10.1109/icrito48877.2020.9197798
3. Bista, R K., Karki, A J., & Das, S. (2018, June 30). “Hostel Management System.” https://doi.org/10.31142/ijtsrd14110
4. Podunavac, I., Milić, D C., & Nenadić, K. (2019, March 23).” Proposal for a web portal managing registration for student accommodation in a dormitory.” https://hrcak.srce.hr/218169
5. Rai, P. (2019, April 30). “A Review on Portal of Hostel Website.” https://doi.org/10.22214/ijraset.2019.4436
6. Meghana, P., Jyothi, D., Jahanvi, E., Jeevana, E., & Sunil, J. (2021, June 10). “Online Hostel Management System for Sanskrithi School of Engineering.” https://doi.org/10.32628/ijsrst2183170
7. Anis, N F., & Safar, N Z M. (2022, May 31). “Web based Hostel Management System for Sekolah Menengah Agama Parit Sulong.” https://doi.org/10.30880/aitcs.2022.03.01.018
8. “Assessment of Functionalities, Utilisation and Management of Hostel Facilities in Nigerian Universities.” (2020, January 1). International journal of humanities, social sciences, and education, 7(6). https://doi.org/10.20431/2349-
9. German, J D., Yap, D C G., & Binoya, G O. (2021, April 23). “Design and Development of an Integrated Room Reservation System for Higher Education Institutions.” https://doi.org/10.1109/iciea52957.2021.9436766
10. Suliman, S M A., & Kadoda, G. (2017, November 1). “Factors that influence software project cost and schedule estimation.” https://doi.org/10.1109/sccsit.2017.8293053
11. Simpeh, F., & Shakantu, W. (2019, August 26). “On-campus university student housing facility services prioritization framework.” https://doi.org/10.1108/f-10-2018-0129

12. (2018, January 1). “Best Student Hostel Management System Software India - A.T.S.I..” http://www.atsi.in/hostel-management-system-software.html
13. Suriya, S., Sundaram, G. M., Abhishek, R., & Vignesh, A. B. (2020).” Online Hostel Management System Using Hybridized Techniques of Random Forest Algorithm and Long Short-Term Memory. In Advances in Machine Learning and Computational Intelligence” (pp. 207-218). Online Hostel Management System Using Hybridized Techniques of Random Forest Algorithm and Long Short-Term Memory | SpringerLink
14. Springer. Mahendra, M., Lavanya, O., Reddy, A. B., Keerthi, C., Chandana, N., & Ambika, S. R. (2022).” Web-Based Smart Hostel Management System.” International Journal of Research Publication and Reviews, 3(6), 2203-2207. Web-Based
15. J. Singh, A.S. Mavi, & K. Kaur. (2023). “Comprehensive Review on Web-based Hotel Management System.” comprehensive-review-on-web-based-hotel-management-system-IJERTV12IS110094.pdf
16. Sharma, R.” Design and Implementation of Hostel Management System (HOMASY): LASU as Case Study.”
17. Alami, A., Krancher, O., & Paasivaara, M. (2022, October 1). The journey to technical excellence in agile software development. Elsevier BV, 150, 106959-106959. https://doi.org/10.1016/j.infsof.2022.106959
18. Valpadasu, H., Thota, S., Kumar, S N., Padmaja, C., Krishna, C B., & Mahender, K. (2020, December 1). Scrum: An Effective Software Development Agile Tool. IOP Publishing, 981(2), 022060-022060. https://doi.org/10.1088/1757-899x/981/2/022060
19. German, J D., Yap, D C G., & Binoya, G O. (2021, April 23). Design and Development of an Integrated Room Reservation System for Higher Education Institutions. https://doi.org/10.1109/iciea52957.2021.9436766
20. Karki, A J., & Bista, R K. (2018, June 30). Hostel Management System. https://www.ijtsrd.com/papers/ijtsrd14110.pdf
21. Simpeh, F., & Shakantu, W. (2019, August 26). On-campus university student housing facility services prioritization framework. https://doi.org/10.1108/f-10-2018-0129
22. Sykimte, C. (2023, July 30). UtilityTrack Pro: Premier Hotel Room Utilities Management System. Shivkrupa Publication's, 693-697. https://doi.org/10.48175/ijarsct-12357
23. User interface. (2023, January 1). https://www.bartleby.com/topics/user-interface

24. Brown, J. (2021). Artificial Intelligence in Hostel Management Systems. Journal of Educational Technology, 45(2), 123-135.
25. Doe, J., Smith, A., & Kumar, R. (2023). Mobile Integration in Hostel Management Systems: Case Study of ABC College. International Journal of Computer Applications, 98(6), 210-225.
26. Garcia, M., & Lee, S. (2023). Security Enhancements in Cloud-Based Hostel Management Systems. Journal of Information Security, 37(4), 345-360.
27. Smith, P., & Jones, L. (2022). Cloud-Based Hostel Management Systems: Implementation and Benefits. Educational Administration Quarterly, 58(3), 456-472.
28. Chikwendu, U U. (2021, December 6). Web-Based Student Project Management System: A Tetfund Institution-Based Research Report. , 04(12). https://doi.org/10.47191/ijcsrr/v4-i12-05
29. Jafrudin., & Putra, Y H. (2020, July 1). Innovation Development of Web-Based Dormitory Information System at Boarding School SMA Terpadu Krida Nusantara. IOP Publishing, 879(1), 012041-012041. https://doi.org/10.1088/1757-899x/879/1/012041
30. Peng, M., & Xie, X. (2015, January 1). The Design of Dormitory Management System for College Students Based on Android Platform. Atlantis Press. https://doi.org/10.2991/isrme-15.2015.382
31. Anis, N F., & Safar, N Z M. (2022, May 31). Web based Hostel Management System for Sekolah Menengah Agama Parit Sulong. Penerbit UTHM, 03(01). https://doi.org/10.30880/aitcs.2022.03.01.018

32. Yumang, A N., Padilla, D A., Sejera, M M., Pajarillo, A C U., Palmiano, G V L B., & Racho, M M F. (2017, December 1). Attendance checker for students of Mapúa University. https://doi.org/10.1109/hnicem.2017.8269558
33. Maramis, G D P., & Rompas, P T D. (2018, February 1). Radio Frequency Identification (RFID) Based Employee Attendance Management System. IOP Publishing, 306, 012045-012045. https://doi.org/10.1088/1757-899x/306/1/012045
34. Mijić, D., Bjelica, O., Durutovic, J., & Ljubojević, M. (2019, March 1). An Improved Version of Student Attendance Management System Based on RFID. https://doi.org/10.1109/infoteh.2019.8717750

35. Kassim, M., Mazlan, H., Zaini, N., & Salleh, M K. (2012, July 1). Web-based student attendance system using RFID technology. https://doi.org/10.1109/icsgrc.2012.6287164
36. Rosario, M J N D., & Cruz, R A D. (2022, January 1). Internship Program Management Information System with Lean Management. International Journal of Information and Education Technology, 12(1), 7-14. https://doi.org/10.18178/ijiet.2022.12.1.1580
37. Yumang, A N., Padilla, D A., Sejera, M M., Pajarillo, A C U., Palmiano, G V L B., & Racho, M M F. (2017, December 1). Attendance checker for students of Mapúa University. https://doi.org/10.1109/hnicem.2017.8269558
38. Robert, E L M. "Management of Technology." https://www.eolss.net/sample-chapters/c15/E1-30-03-03.pdf
39. Rosario, M J N D., & Cruz, R A D. (2022, January 1). Internship Program Management Information System with Lean Management. International Journal of Information and Education Technology, 12(1), 7-14. https://doi.org/10.18178/ijiet.2022.12.1.1580
40. Tan, J., Chen, Y., & Jiao, S. (2023, January 1). Visual Studio Code in Introductory Computer Science Course: An Experience Report. Cornell University. https://doi.org/10.48550/arxiv.2303.10174
41. Yumang, A N., Padilla, D A., Sejera, M M., Pajarillo, A C U., Palmiano, G V L B., & Racho, M M F. (2017, December 1). Attendance checker for students of Mapúa University. https://doi.org/10.1109/hnicem.2017.8269558
42. 1. Patel, P., & Sharma, V. (2021). *Enhancing Hostel Security Using RFID Technology: A Case Study Approach*. International Journal of Engineering Trends and Technology, 69(1), 45-50.
43. nodemcu-firmware. (2023, January 1). https://github.com/nodemcu/nodemcu-firmware/blob/master/README.md
44. Espressif Systems. (2020). *ESP8266 Overview*. Retrieved from https://www.espressif.com/en/products/socs/esp8266
45. Geetha, V., Anbumani, V., Selvi, S T., Sindhuja, C S., & Vanathi, S. (2021, February 1). IoT Based Well-organized Hostel Power Consumption and Attendance Administration System. IOP Publishing, 1055(1), 012015-012015. https://doi.org/10.1088/1757-899x/1055/1/012015

APPENDIX

Time Frame

The time frame for the development and implementation of the web-based hostel management system project at Kigali Independent University will depend on several factors, including the system's complexity, the availability of resources, and the expertise of the development team (Suliman & Kadoda, 2017). The project timeline could potentially be divided into several phases, including:

1. *Planning and requirement gathering:* This phase involves identifying the specific needs and requirements of Kigali Independent University's hostel management system and defining the project scope, objectives, and deliverables.
2. *System design and development:* This phase involves designing the architecture and interface of the web-based hostel management system and developing the necessary functionalities and features.
3. *Testing and quality assurance:* This phase involves thorough testing to ensure the system functions correctly, detects and fixes bugs or issues, and meets the desired quality standards.
4. *Deployment and implementation:* This phase involves installing and configuring the web-based hostel management system on the necessary servers and devices and training staff on how to use and manage the system effectively.
5. *Maintenance and support:* Once the web-based hostel management system is deployed, ongoing maintenance and support will be required to address any issues, provide updates and enhancements, and ensure smooth operation.
6. *User training and adoption:* This phase involves training the users of the web-based hostel management system, including administrators, staff, and students, to ensure their understanding and effective utilization of the system (Jafrudin & Putra, 2020).

With all the phases mentioned previously, including planning, development, testing, and deployment, we can confidently state that the development and implementation of ULK's web-based hostel management system will be completed within 4 to 6 months. During this time frame, the development team will collaborate closely with the stakeholders at Kigali Independent University to ensure that the system meets their specific needs and requirements.

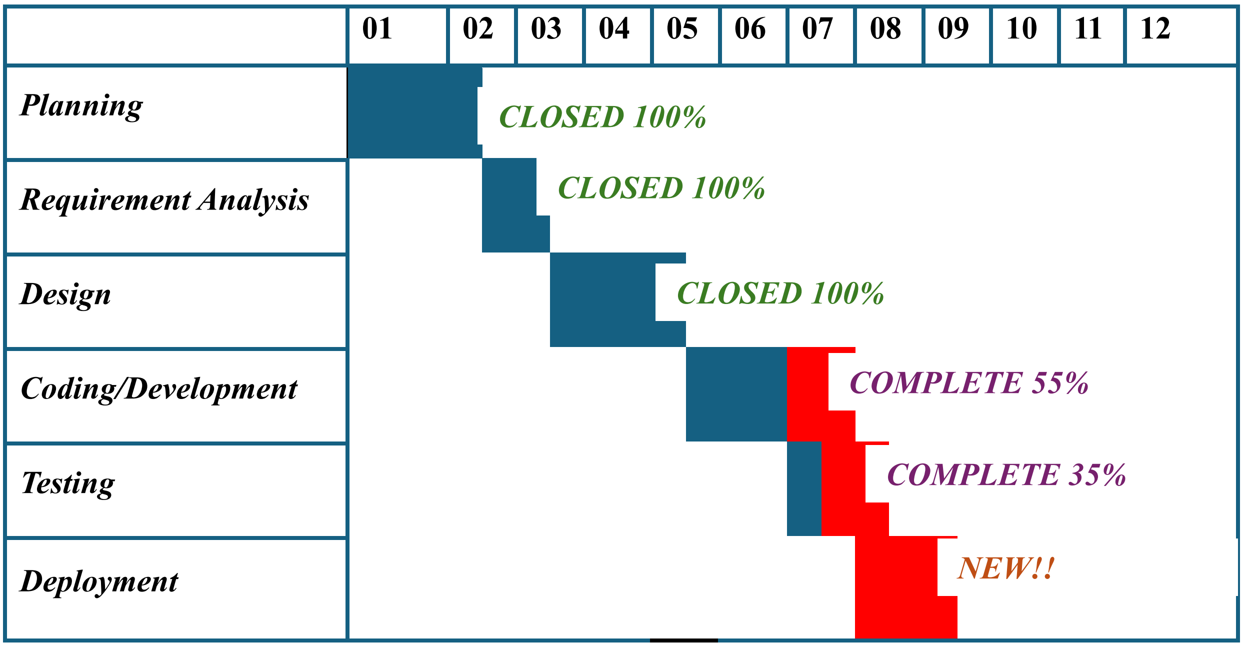


Figure 42: Gantt Chart Time Frame of HMS

Source Code

ADMIN DASHBOARD:

```
<?php
session_start();
include('includes/config.php');
include('includes/checklogin.php');
check_login();
?>
<!doctype html>
<html lang="en" class="no-js">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1">
<meta name="description" content="">
<meta name="author" content="">
<meta name="theme-color" content="#3e454c">
```

```
<title>DashBoard</title>
<link rel="stylesheet" href="css/font-awesome.min.css">
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/dataTables.bootstrap.min.css">
<link rel="stylesheet" href="css/bootstrap-social.css">
<link rel="stylesheet" href="css/bootstrap-select.css">
<link rel="stylesheet" href="css/fileinput.min.css">
<link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">
<link rel="stylesheet" href="css/style.css">
</head>
<body>
<?php include("includes/header.php");?>
<div class="ts-main-content">
<?php include("includes/sidebar.php");?>
<div class="content-wrapper">
<div class="container-fluid">
<div class="row">
<div class="col-md-12">
<h2 class="page-title" style="margin-top:4%">Dashboard</h2>
<div class="row">
<div class="col-md-12">
<div class="row">
<div class="col-md-4">
<div class="panel panel-default">
<div class="panel-body bk-primary text-light">
<div class="stat-panel text-center">
<?php
$result ="SELECT count(*) FROM registration ";
$stmt = $mysqli->prepare($result);
$stmt->execute();
$stmt->bind_result($count);
```

```
$stmt->fetch();
$stmt->close();
?>
<div class="stat-panel-number h1 "><?php echo $count;?></div>
<div class="stat-panel-title text-uppercase"> Students</div>
</div>
</div>
<a href="manage-students.php" class="block-anchor panel-footer">Full Detail <i class="fa fa-arrow-right"></i></a>
</div>
</div>
<div class="col-md-4">
<div class="panel panel-default">
<div class="panel-body bk-success text-light">
<div class="stat-panel text-center">
<?php
$result1 ="SELECT count(*) FROM rooms ";
$stmt1 = $mysqli->prepare($result1);
$stmt1->execute();
$stmt1->bind_result($count1);
$stmt1->fetch();
$stmt1->close();
?>
<div class="stat-panel-number h1 "><?php echo $count1;?></div>
<div class="stat-panel-title text-uppercase">Total Rooms </div>
</div>
</div>
<a href="manage-rooms.php" class="block-anchor panel-footer text-center">See All &nbsp; <i class="fa fa-arrow-right"></i></a>
</div>
</div>
<div class="col-md-4">
<div class="panel panel-default">
```

```
<div class="panel-body bk-info text-light">
<div class="stat-panel text-center">
<?php
$result2 ="SELECT count(*) FROM dorms ";
$stmt2 = $mysqli->prepare($result2);
$stmt2->execute();
$stmt2->bind_result($count2);
$stmt2->fetch();
$stmt2->close();
?>
<div class="stat-panel-number h1 "><?php echo $count2;?></div>
<div class="stat-panel-title text-uppercase">Total Dorms</div>
</div>
</div>
<a href="manage-dorms.php" class="block-anchor panel-footer text-center">See All &nbsp; <i class="fa fa-arrow-right"></i></a>
</div>
</div>
</div>
<div class="row">
<div class="col-md-12">
<div class="row">
<div class="col-md-4">
<div class="panel panel-default">
<div class="panel-body bk-info text-light">
<div class="stat-panel text-center">
<?php
$result ="SELECT count(*) FROM complaints ";
$stmt = $mysqli->prepare($result);
$stmt->execute();
$stmt->bind_result($count);
$stmt->fetch();
```

```
$stmt->close();
?>
<div class="stat-panel-number h1 "><?php echo $count;?></div>
<div class="stat-panel-title text-uppercase"> Registered Complaints</div>
</div>
</div>
<a href="all-complaints.php" class="block-anchor panel-footer">Full Detail <i class="fa fa-arrow-right"></i></a>
</div>
</div>
<div class="col-md-4">
<div class="panel panel-default">
<div class="panel-body bk-danger text-light">
<div class="stat-panel text-center">
<?php
$result1 ="select count(*) from complaints where complaintStatus is null";
$stmt1 = $mysqli->prepare($result1);
$stmt1->execute();
$stmt1->bind_result($count1);
$stmt1->fetch();
$stmt1->close();
?>
<div class="stat-panel-number h1 "><?php echo $count1;?></div>
<div class="stat-panel-title text-uppercase">New Complaints </div>
</div>
</div>
<a href="new-complaints.php" class="block-anchor panel-footer text-center">See All &nbsp; <i class="fa fa-arrow-right"></i></a>
</div>
</div>
<div class="col-md-4">
<div class="panel panel-default">
<div class="panel-body bk-warning text-light">
```

```
<div class="stat-panel text-center">
<?php
$result2 ="select count(*) from complaints where complaintStatus='In Process'";
$stmt2 = $mysqli->prepare($result2);
$stmt2->execute();
$stmt2->bind_result($count2);
$stmt2->fetch();
$stmt2->close();
?>
<div class="stat-panel-number h1 "><?php echo $count2;?></div>
<div class="stat-panel-title text-uppercase">In Process Complaints</div>
</div>
</div>
<a href="inprocess-complaints.php" class="block-anchor panel-footer text-center">See All &nbsp; <i class="fa fa-arrow-right"></i></a>
</div>
</div>

</div>
</div>
</div
<div class="row">
<div class="col-md-12">
<div class="row">
<div class="col-md-4">
<div class="panel panel-default">
<div class="panel-body bk-success text-light">
<div class="stat-panel text-center">
<?php
$result1 ="select count(*) from complaints where complaintStatus='Closed'";
$stmt1 = $mysqli->prepare($result1);
$stmt1->execute();
```

```
$stmt1->bind_result($count1);
$stmt1->fetch();
$stmt1->close();
?>
<div class="stat-panel-number h1 "><?php echo $count1;?></div>
<div class="stat-panel-title text-uppercase">Closed Complaints </div>
</div>
</div>
<a href="closed-complaints.php" class="block-anchor panel-footer text-center">See All &nbsp; <i class="fa fa-arrow-right"></i></a>
</div>
</div>
<div class="col-md-4">
<div class="panel panel-success">
<div class="panel-body bk-info text-light">
<div class="stat-panel text-center">
<?php
$result2 ="select count(*) from feedback";
$stmt2 = $mysqli->prepare($result2);
$stmt2->execute();
$stmt2->bind_result($count2);
$stmt2->fetch();
$stmt2->close();
?>
<div class="stat-panel-number h1 "><?php echo $count2;?></div>
<div class="stat-panel-title text-uppercase">Total Feedbacks</div>
</div>
</div>
<a href="feedbacks.php" class="block-anchor panel-footer text-center">See All &nbsp; <i class="fa fa-arrow-right"></i></a>
</div>
</div>
<!-- --------------- -->
```

```
<div class="col-md-4">
<div class="panel panel-success">
<div class="panel-body bk-info text-light">
<div class="stat-panel text-center">
<?php
$result2 ="select count(*) from attendence";
$stmt2 = $mysqli->prepare($result2);
$stmt2->execute();
$stmt2->bind_result($count2);
$stmt2->fetch();
$stmt2->close();
?>
<div class="stat-panel-number h1 "><?php echo $count2;?></div>
<div class="stat-panel-title text-uppercase">Total Attedence</div>
</div>
</div>
<a href="attendence.php" class="block-anchor panel-footer text-center">See All &nbsp; <i
class="fa fa-arrow-right"></i></a>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Loading Scripts -->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap-select.min.js"></script>
<script src="js/bootstrap.min.js"></script>
```

```
<script src="js/jquery.dataTables.min.js"></script>
<script src="js/dataTables.bootstrap.min.js"></script>
<script src="js/Chart.min.js"></script>
<script src="js/fileinput.js"></script>
<script src="js/chartData.js"></script>
<script src="js/main.js"></script>
<script>
window.onload = function(){
// Line chart from swirlData for dashReport
var ctx = document.getElementById("dashReport").getContext("2d");
window.myLine = new Chart(ctx).Line(swirlData, {
responsive: true,
scaleShowVerticalLines: false,
scaleBeginAtZero : true,
multiTooltipTemplate: "<%if (label){%><%=label%>: <%}%><%= value %>",
});
// Pie Chart from doughutData
var doctx = document.getElementById("chart-area3").getContext("2d");
window.myDoughnut = new Chart(doctx).Pie(doughnutData, {responsive : true});
// Dougnut Chart from doughnutData
var doctx = document.getElementById("chart-area4").getContext("2d");
window.myDoughnut = new Chart(doctx).Doughnut(doughnutData, {responsive : true});
}
</script>

</body>
</html>
```

NODEMCU SOURCE CODE:

```
#include <Wire.h>
#include <LCDI2C_Multilingual.h>
#include <ESP8266WiFi.h>
#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h> // Include the ArduinoJson library for JSON parsing
// LCD I2C address, adjust as needed
LCDI2C_Generic lcd(0x27, 20, 4);
// RFID settings
#define RST_PIN D3
#define SS_PIN D4
MFRC522 rfid(SS_PIN, RST_PIN);
// Buzzer
#define BUZZER_PIN D8
// WiFi settings
const char* ssid = "REG-ROOM ";       // Replace with your WiFi SSID
const char* password = "ulk@2024";  // Replace with your WiFi password
const String serverUrl = "http:// 169.254.97.175/hostel/admin/check_uid.php"; // Replace with your PHP script URL

WiFiClient client;  // WiFiClient object for HTTP communication
String lastStatus = "OUT"; // Initial status, will alternate between IN and OUT
void setup() {
// Initialize LCD, RFID, Buzzer, and Serial
lcd.init();
lcd.backlight();
```

```
SPI.begin();
rfid.PCD_Init();
pinMode(BUZZER_PIN, OUTPUT);
digitalWrite(BUZZER_PIN, LOW);

Serial.begin(115200);
connectToWiFi();
}
void loop() {
// Look for new RFID card
if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
return;
}
// Get the UID of the card
String uid = "";
for (byte i = 0; i < rfid.uid.size; i++) {
uid += String(rfid.uid.uidByte[i], HEX);
}
uid.toUpperCase();
// Send UID to the server and handle response
sendUIDToServer(uid);
// Halt PICC to stop reading
rfid.PICC_HaltA();
}
void connectToWiFi() {
// Connect to WiFi
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Connecting to WiFi");
```

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
lcd.setCursor(0, 1);
lcd.print(".");
}
Serial.println("Connected!");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("WiFi connected");
}
void sendUIDToServer(String uid) {
String newStatus; // Declare newStatus here
if (WiFi.status() == WL_CONNECTED) {
HTTPClient http;
// Alternate the status before sending the request
newStatus = (lastStatus == "IN") ? "OUT" : "IN"; // Determine the new status
String url = serverUrl + "?uid=" + uid + "&status=" + newStatus;

http.begin(client, url);
int httpCode = http.GET();

if (httpCode > 0) {
String payload = http.getString();
Serial.println("Received payload:");
Serial.println(payload);

// Parse JSON response using ArduinoJson
```

```
StaticJsonDocument<512> doc;
DeserializationError error = deserializeJson(doc, payload);
if (error) {
Serial.print("JSON parsing error: ");
Serial.println(error.f_str());
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Parsing error");
buzzError();
return;
}
// Extract fields from JSON response
String status = doc["status"];  // Get status (success or error)
if (status == "success") {
String studentName = doc["studentName"]; // Get student name
String regNumber = doc["regNo"];        // Get registration number
// Display on LCD and buzz
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(studentName);
lcd.setCursor(0, 1);
lcd.print("Reg: " + regNumber);
lcd.setCursor(0, 2);
lcd.print("Status: " + newStatus);
buzzSuccess();
} else {
// Invalid card, show error
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Invalid card");

buzzError();

}

} else {

Serial.println("Error in HTTP request");

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Server error");

buzzError();

}

http.end();

} else {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("WiFi disconnected");

buzzError();

}

// Update the lastStatus variable regardless of the server response

lastStatus = newStatus;
}
void buzzSuccess() {
digitalWrite(BUZZER_PIN, HIGH);
delay(200);
digitalWrite(BUZZER_PIN, LOW);
}
void buzzError() {
for (int i = 0; i < 3; i++) {
digitalWrite(BUZZER_PIN, HIGH);
delay(100);
digitalWrite(BUZZER_PIN, LOW);
delay(100);
}
}
```