PROJECT REPORT

Hostel Management System

Group Members

Afreen Sudheer (TKM22CS014)

Alan George (TKM22CS018)

K R Abhinand Babu (TKM22CS090)

Mohammed Afeef (TKM22CS102)

Contents

1. Introduction

The Hostel Management System is designed to streamline and enhance the management of hostel facilities, providing an efficient solution for administrators and students alike. This project addresses the common challenges faced in hostel management, such as room allocation, fee tracking, and student data management.

By leveraging modern web technologies, the system offers a user-friendly interface that allows administrators to easily monitor and manage various aspects of hostel operations. Key features include real-time statistics on room availability, comprehensive fee overviews, and detailed student records, all aimed at improving operational efficiency and enhancing the user experience.

This report outlines the objectives, methodologies, and outcomes of the project, highlighting the development process and the technologies utilized. The findings and insights presented herein aim to contribute to the ongoing improvement of hostel management practices and serve as a foundation for future enhancement.

2. Abstract

The Hostel Management System website is designed to streamline the administration and organization of hostel facilities for students, faculty, and staff. This platform facilitates the efficient management of resident information, room assignments, fee payments, and maintenance requests. With a user-friendly interface, administrators can easily add or update records, assign rooms, and monitor hostel occupancy. Students and faculty members can log in to check room availability, view hostel rules, and manage their profiles.

The website integrates a secure database for data storage, ensuring all resident and operational data is organized and accessible in real time. By digitizing and automating key hostel processes, the system reduces administrative workload, enhances communication, and improves the overall hostel experience for residents and staff alike.

3. System Requirements

3.1. Software Requirements

- **Frontend Development**:

→ HTML5, CSS3, JavaScript: For building and styling the website interface.

- **Backend Development**:

→ MySQL: Database management system for storing hostel data

- **Web Server**:

→ Apache or Nginx: For hosting the website and serving files to users.

- **Operating System**:

→ Windows, macOS, or Linux: Support basic web dev tools and servers.

- **Development Tools**:

→ Code Editor (e.g., VS Code, Sublime Text)

→ Database Management Tool (e.g., phpMyAdmin, MySQL Workbench)

- **Browser**:

→ Any modern web browser (e.g., Chrome, Firefox, Safari) for testing and viewing the website.

3.2. Hardware Requirements

- **Processor**:

→ Intel i3 or AMD equivalent, or better.

- **RAM**:

→ 4 GB minimum (8 GB recommended for smoother development).

- **Storage**:

→ 500 MB - 1 GB free disk space for project files, database, and server configurations.

4. Index

5. GUI Design

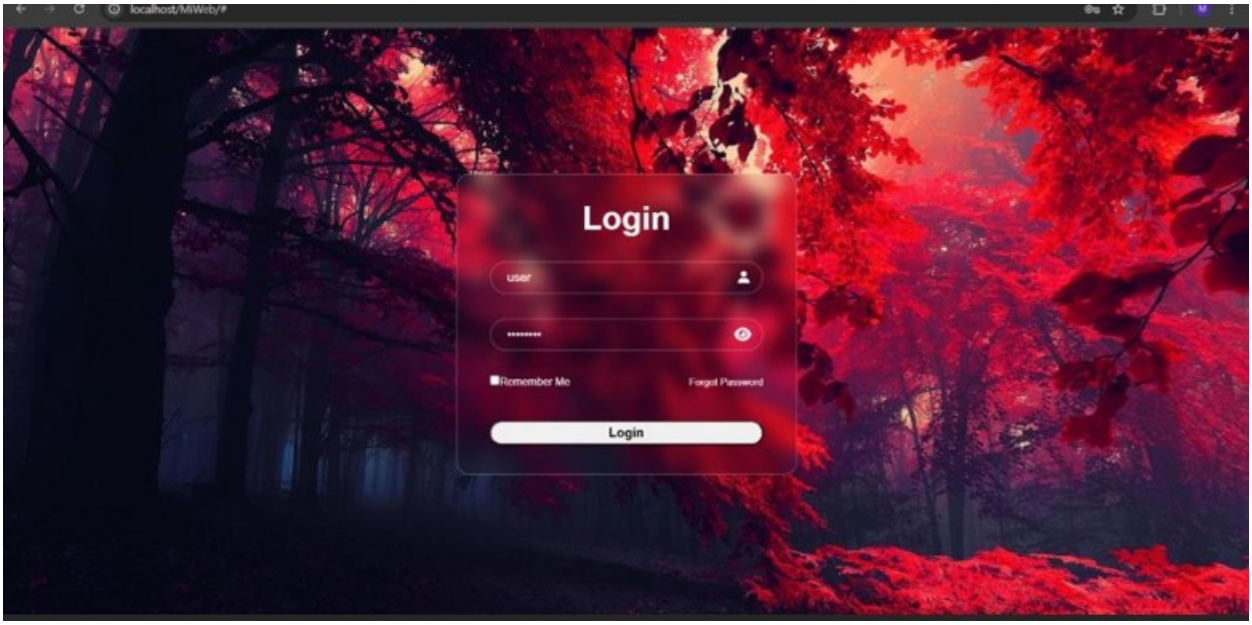


Fig. 1 Login Page

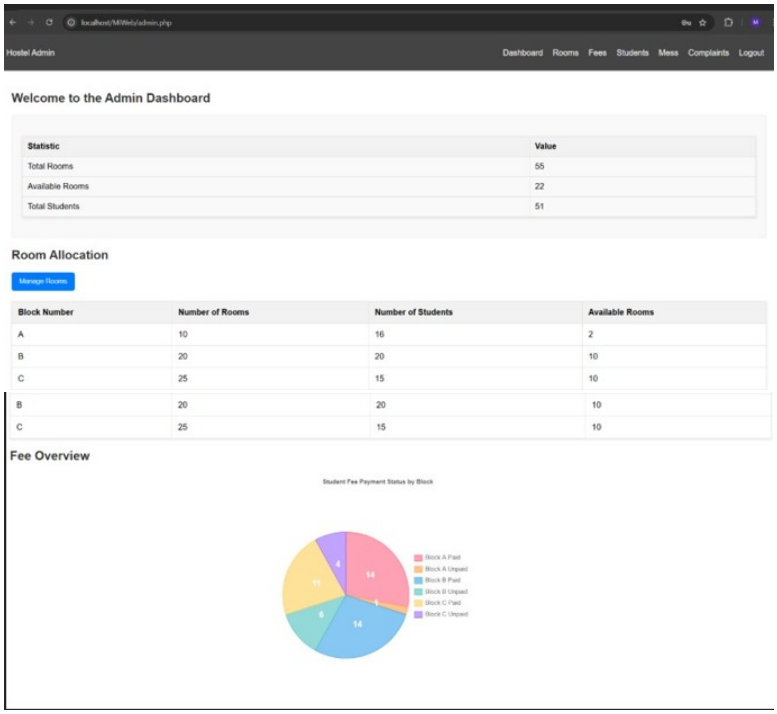


Fig. 2 Dashboard Pages

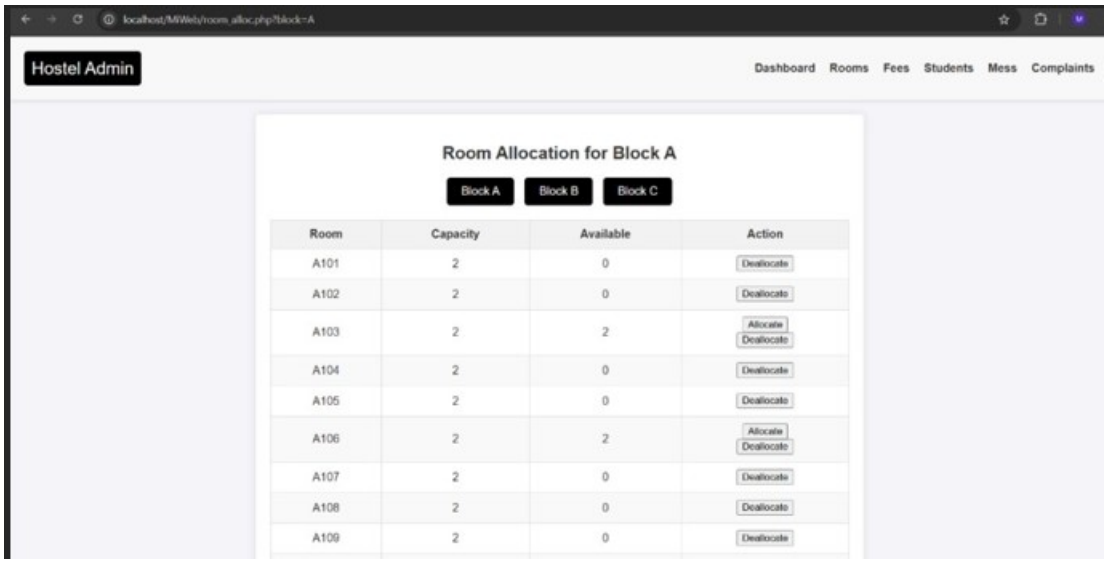
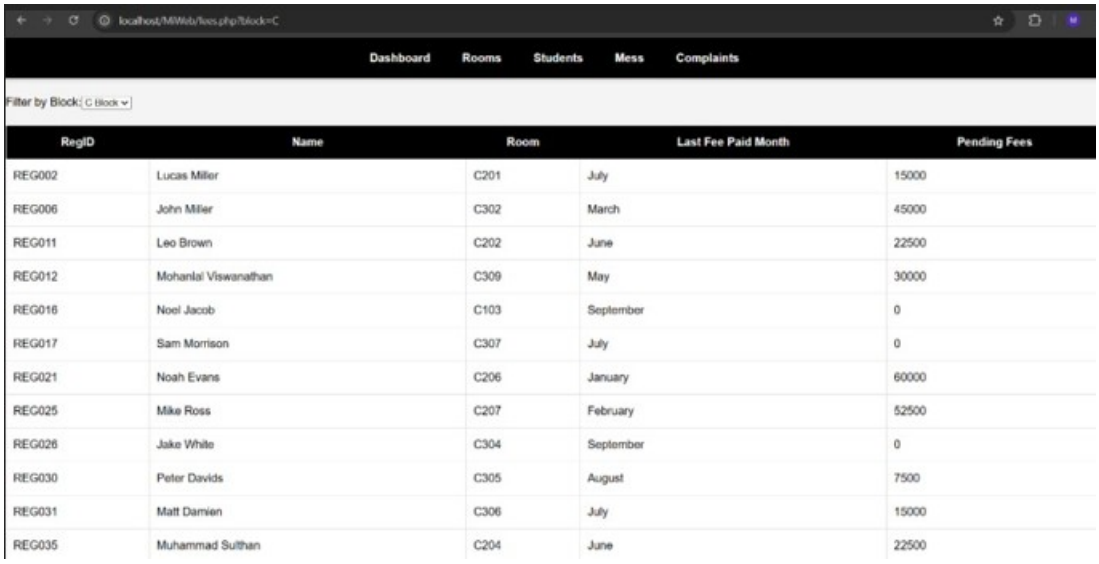
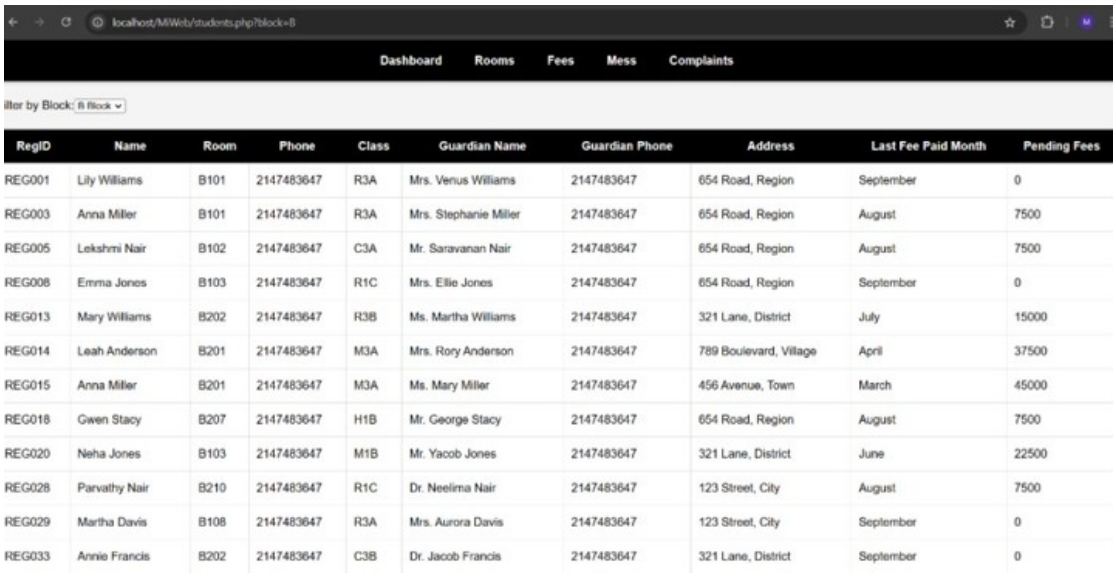


Fig. 3 Room Allocation Page



RegID	Name	Room	Last Fee Paid Month	Pending Fees
REG002	Lucas Miller	C201	July	15000
REG006	John Miller	C302	March	45000
REG011	Leo Brown	C202	June	22500
REG012	Mohanlal Viswanathan	C309	May	30000
REG016	Noel Jacob	C103	September	0
REG017	Sam Morrison	C307	July	0
REG021	Noah Evans	C206	January	60000
REG025	Mike Ross	C207	February	52500
REG026	Jake White	C304	September	0
REG030	Peter Davids	C305	August	7500
REG031	Matt Damien	C306	July	15000
REG035	Muhammad Sulthan	C204	June	22500

Fig. 4 Fee Payment Overview Page



RegID	Name	Room	Phone	Class	Guardian Name	Guardian Phone	Address	Last Fee Paid Month	Pending Fees
REG001	Lily Williams	B101	2147483647	R3A	Mrs. Venus Williams	2147483647	654 Road, Region	September	0
REG003	Anna Miller	B101	2147483647	R3A	Mrs. Stephanie Miller	2147483647	654 Road, Region	August	7500
REG005	Lekshmi Nair	B102	2147483647	C3A	Mr. Saravanan Nair	2147483647	654 Road, Region	August	7500
REG008	Emma Jones	B103	2147483647	R1C	Mrs. Ellie Jones	2147483647	654 Road, Region	September	0
REG013	Mary Williams	B202	2147483647	R3B	Ms. Martha Williams	2147483647	321 Lane, District	July	15000
REG014	Leah Anderson	B201	2147483647	M3A	Mrs. Rory Anderson	2147483647	789 Boulevard, Village	April	37500
REG015	Anna Miller	B201	2147483647	M3A	Ms. Mary Miller	2147483647	456 Avenue, Town	March	45000
REG018	Gwen Stacy	B207	2147483647	H1B	Mr. George Stacy	2147483647	654 Road, Region	August	7500
REG020	Neha Jones	B103	2147483647	M1B	Mr. Yacob Jones	2147483647	321 Lane, District	June	22500
REG028	Parvathy Nair	B210	2147483647	R1C	Dr. Neelima Nair	2147483647	123 Street, City	August	7500
REG029	Martha Davis	B108	2147483647	R3A	Mrs. Aurora Davis	2147483647	123 Street, City	September	0
REG033	Annie Francis	B202	2147483647	C3B	Dr. Jacob Francis	2147483647	321 Lane, District	September	0

Fig. 5 Student Information Page

Fig. 6 Student Room Allocation Page

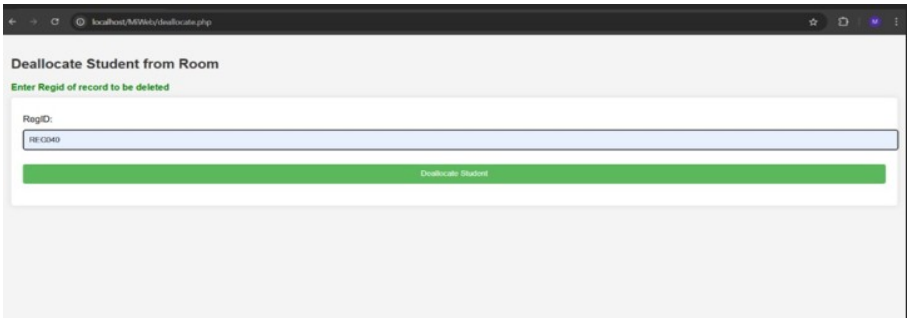


Fig.7 Student Room Deallocation Page

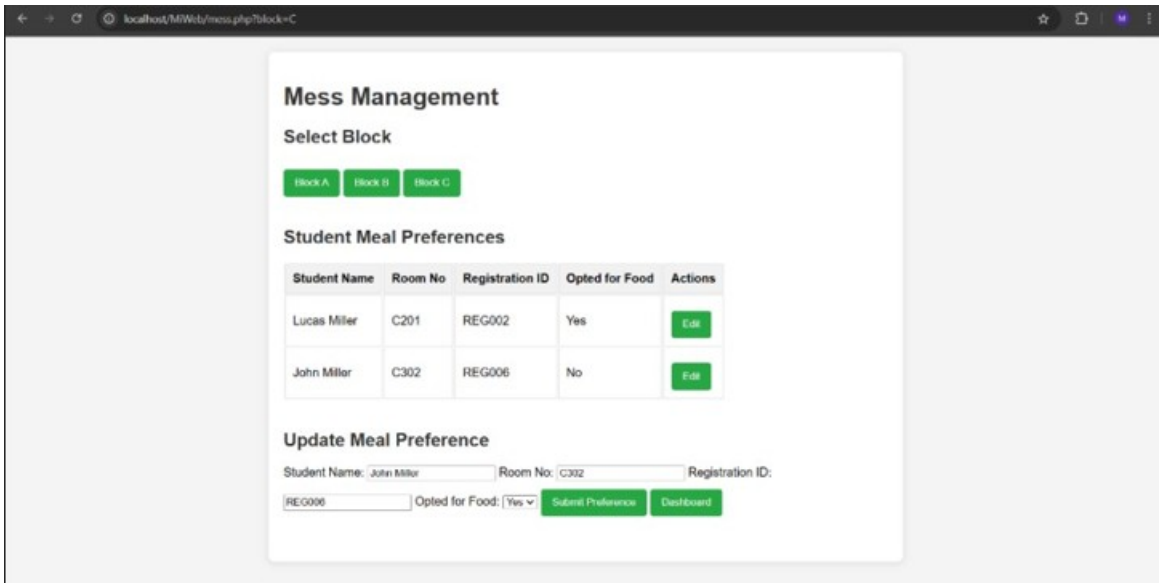


Fig. 8 Mess Details Page

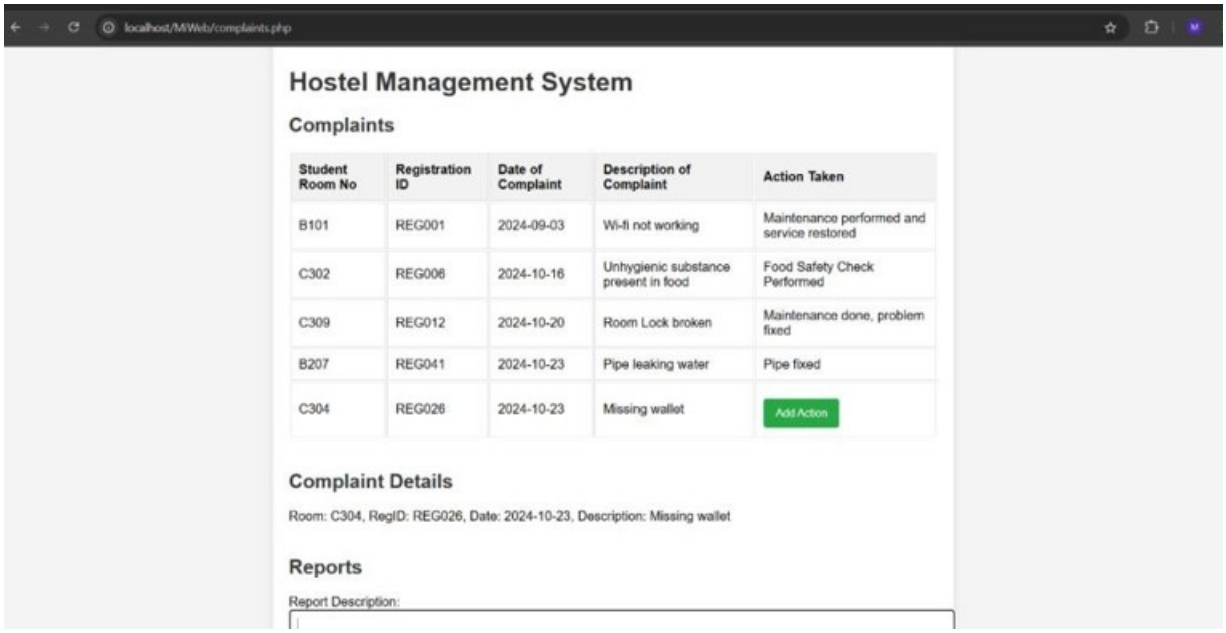


Fig. 9 Complaints Page

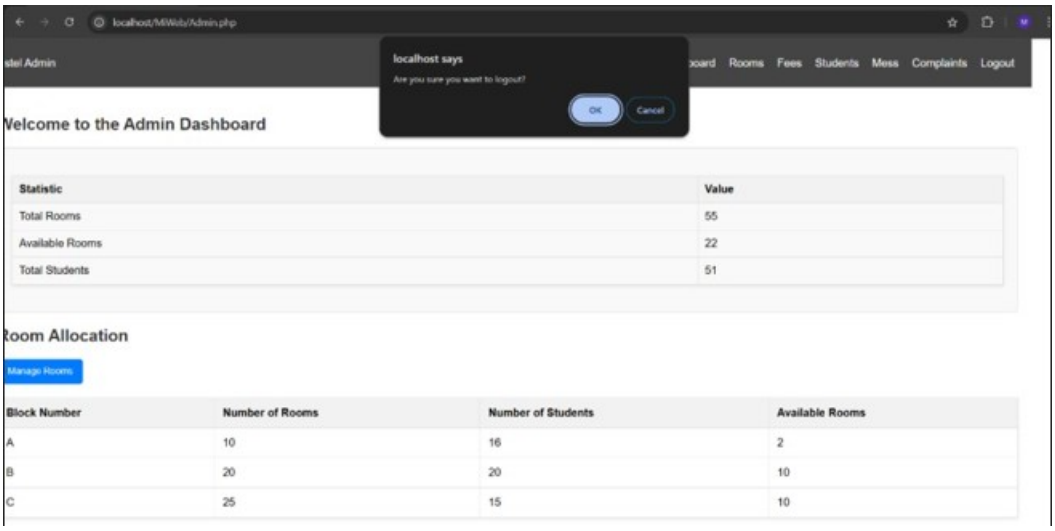


Fig. 10 Logout Page

6. Conclusion

The Hostel Management System developed in this project provides a comprehensive solution for managing hostel operations efficiently. By integrating functionalities such as user authentication, room allocation, student management, fee tracking, and complaint handling, the system addresses the common challenges faced by hostel administrators and enhances the overall user experience for students.

The implementation of a web-based interface ensures accessibility and ease of use, allowing administrators to manage hostel activities from any device with internet access. The system's design prioritizes security, performance, and usability, making it a robust tool for hostel management.

Future enhancements could include the integration of additional features such as automated notifications for fee payments, a mobile application for students, and advanced reporting capabilities. Overall, this project lays a solid foundation for further development and improvement in hostel management practices.

7. Appendices

7.1. Analysis Model

1. Use Case Diagram

- Purpose: Illustrates the interactions between users and the system.
- Actors:

i. Admin: Manages hostel operations, such as assigning rooms, handling payments, and responding to maintenance requests.

ii. Resident: Views room assignments, submits maintenance requests, and checks payment status.

- Use Cases:

i. Manage Residents: Admin can add, update, or remove resident records.

ii. Room Assignment: Admin assigns or reassigns rooms to residents.

iii. Payment Processing: Admin manages payment records, and residents can view payment

status.

iv. Maintenance Request Handling: Residents submit requests, and admins track and update request status.

v. View Profile: Residents can view personal details, room assignments, and payment history.

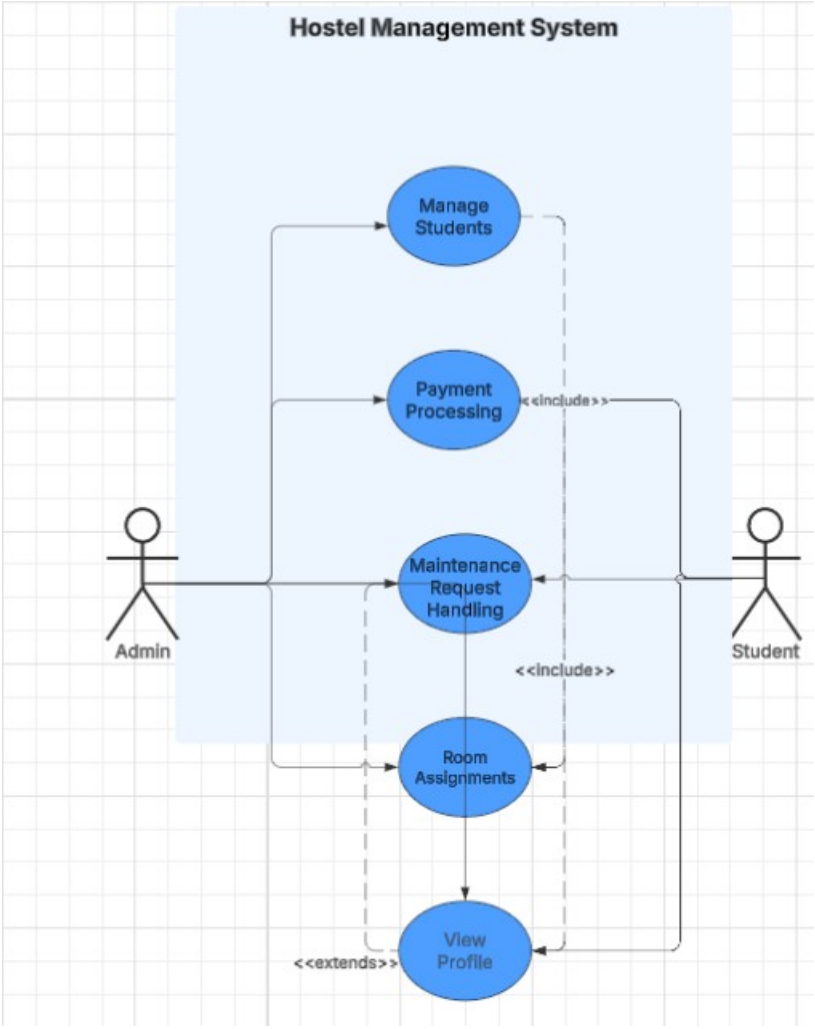


Fig. 11 Use Case Diagram

2. Entity-Relationship (ER) Diagram

5. Purpose: Shows the database structure, including tables and relationships between entities.

- Entities:

i. Resident: Stores resident information

ii. Room: Contains details about hostel rooms

iii. Payment: Tracks payment details

iv. MaintenanceRequest: Records maintenance issues

- Relationships:

i. Resident–Room: Each resident is assigned a room (one-to-one relationship).

ii. Resident–Payment: Residents have multiple payments (one-tomany relationship).

iii. Resident–MaintenanceRequest: Each resident can submit multiple requests (one-to-many relationship).

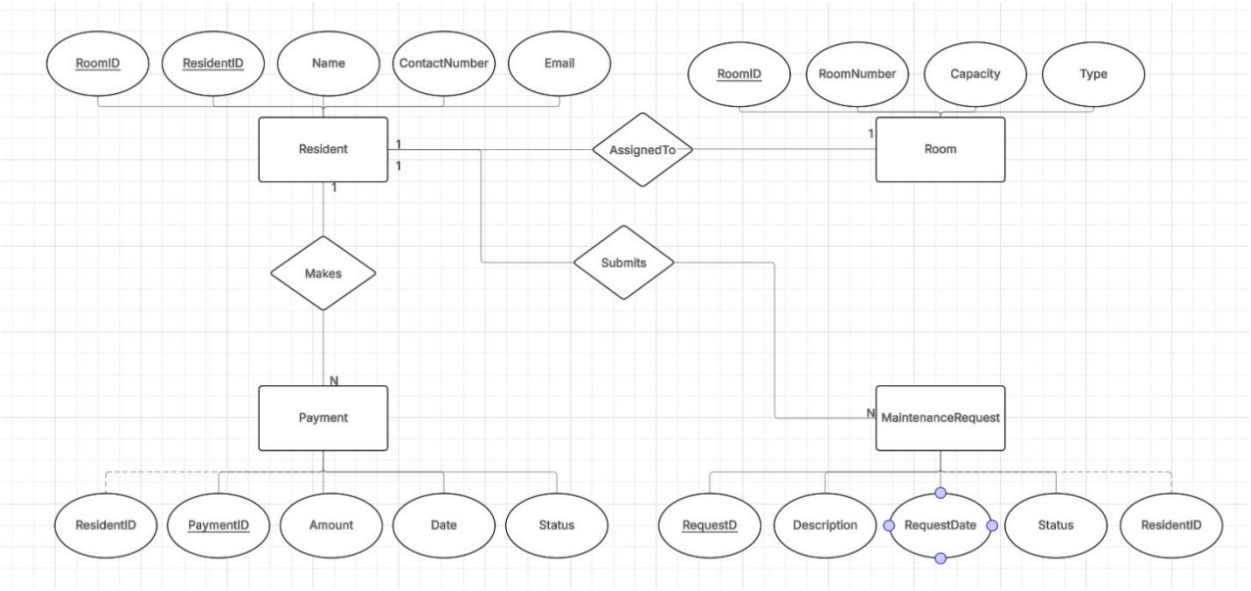


Fig.1 2 Entity Relationship Diagram

3. Data Flow Diagram (DFD)

- Purpose: Shows the flow of information within the system and identifies sources, processes, and data stores.
- Levels:

i. Level 0 (Context Diagram):

→ Gives an overview of the entire system, showing interactions between actors (Admin, Resident) and major processes.

ii. Level 1:

→ Resident Management: Data flow for adding, updating, and managing resident details.

→ Room Allocation: Data flow for assigning and updating room status.

→ Payment Processing: Data flow for recording and viewing payment details.

→ Maintenance Handling: Data flow for submitting and tracking maintenance requests.

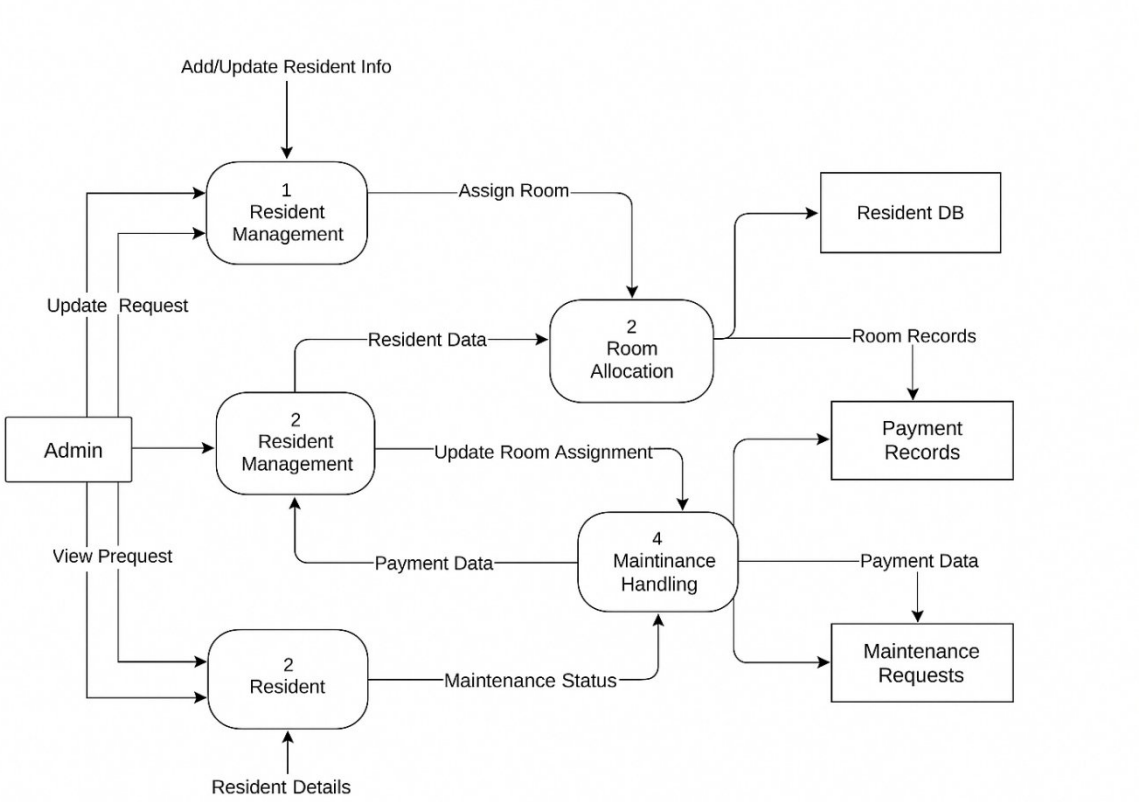


Fig. 13 Data Flow Diagram

7.2. Future Scope

i. User Authentication and Roles

- The exact authentication method (e.g., standard login, single sign-on, or multi-factor authentication).

ii. Payment Integration

- Whether the system will support online payment processing and, if so, which payment gateway to integrate (e.g., PayPal, Stripe, Razorpay).

iii. Maintenance Request Workflow

- Define the specific statuses and stages for maintenance requests (e.g., Pending, In Progress, Completed).

iv. Data Backup and Recovery

- The method for data backup (e.g., automated daily backups, manual weekly backups).

8. References

1. IEEE Std 830-1998, "IEEE Recommended Practice for Software Requirements Specifications," IEEE, 1998.
2. W. S. Humphrey, "Managing the Software Process," Addison-Wesley, 1989.
3. Sommerville, I. (2011). "Software Engineering" (9th ed.). Boston: Addison-Wesley.
4. Pressman, R. S. (2014). "Software Engineering: A Practitioner's Approach" (9th ed.). New York: McGraw-Hill.
5. MySQL Documentation. (n.d.). Retrieved from https://dev.mysql.com/doc/ .
6. PHP Manual. (n.d.). Retrieved from https://www.php.net/manual/en/ .

Appendix I

SOFTWARE REQUIREMENTS SPECIFICATION

Hostel Management System

Version : 1.0

Prepared by

Afreen Sudheer(TKM22CS014)

Alan George(TKM22CS018)

K R Abhinand Babu(TKM22CS090)

Mohammed Afeef(TKM22CS102)

Contents

1. Introduction

1.1. Purpose

The purpose of this document is to outline the requirements for the Hostel Management System, which aims to facilitate the management of hostel operations, including room allocation, student management, fee tracking, and complaint handling.

1.2. Scope

The Hostel Management System will provide a web-based interface for administrators to manage hostel facilities efficiently. The system will allow for user authentication, room and student management, fee tracking, and reporting functionalities.

1.3. Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- Admin: Administrator
- DB: Database
- RegID: Registration ID

1.4. References

- IEEE 830-1998 Standard for SRS
- Project files and documentation

1.5. Overview

This document provides a detailed description of the system's requirements, including functional and non-functional requirements, user interface requirements, and system features.

2. Overall Description

2.1. Product Perspective

The Hostel Management System will be a web-based application accessible via standard web browsers. It will interact with a MySQL database to store and retrieve data related to students, rooms, fees, and complaints.

2.2. Product Functions

- User authentication and session management
- Room management (allocation and deallocation)
- Student management (registration, updates, and deletions)
- Fee management (tracking and reporting)
- Complaint management (submission and resolution)
- Reporting and statistics generation

2.3. User Classes and Characteristics

- Administrators: Users who manage the hostel system, including room allocation, student records, and fee management.
- Students: Users who can view their information and fee status.

2.4. Operating Environment

The system will operate on a web server with a MySQL database. It will be compatible with major web browsers (Chrome, Firefox, Safari, Edge) and various operating systems (Windows, macOS, Linux).

2.5. Design and Implementation Constraints

The system must comply with security standards to protect user data and prevent unauthorized access.

2.6. Assumptions and Dependencies

The system assumes that users have internet access and that the web server is properly configured.

3. Specific Requirements

3.1. Functional Requirements

1. User Authentication

- Users must be able to register and log in securely.
- Admin users should have different access levels compared to regular users.

2. Room Management

- Admins must be able to add, update, and delete room information.
- Users should be able to view available rooms and their statuses.

3. Student Management

- Admins must be able to add, update, and remove student records.
- Admins should be able to view student details, including room assignments and fee status.

4. Fee Management

- The system must track and display pending and paid fees for each student.
- Admins should be able to generate reports on fee payments.

5. Complaint Management

- Users must be able to submit complaints.
- Admins should be able to view and update the status of complaints.

6. Statistics and Reporting

- The system must provide real-time statistics on total rooms, available rooms, and total students.
- Visual representations (charts) of fee payment statuses should be available.

3.2. Non-Functional Requirements

1. Performance

- The system should handle multiple concurrent users without significant performance degradation.
- Page load times should be optimized for a smooth user experience.

2. Security

- User data must be stored securely, with encryption for sensitive information (e.g., passwords).
- The system should implement measures to prevent SQL injection and cross-site scripting (XSS) attacks.

3. Usability

- The user interface should be intuitive and easy to navigate for both administrators and students.

- Help documentation should be available to assist users in navigating the system.

4. Scalability

- The system should be designed to accommodate future growth, allowing for the addition of new features and increased user load.

5. Compatibility

- The system should be compatible with major web browsers (Chrome, Firefox, Safari, Edge).
- It should also be compatible with various operating systems (Windows, macOS, Linux).

6. Maintainability

- The codebase should be well-documented and organized to facilitate future maintenance and updates.

4. External Interface Requirements

4.1. User Interfaces

The system will have a web-based interface with forms for user login, room allocation, student registration, fee management, and complaint submission.

4.2. Hardware Interfaces

The system does not require any specialized hardware and is designed to operate on standard user devices with internet connectivity. It supports desktops, laptops, tablets, and smartphones that can run modern browsers.

4.3. Software Interfaces

The system will interface with a MySQL database for data storage and retrieval.

4.4. Communication Interfaces

The system will use HTTPS for secure communication.

5. System Features

5.1. User Authentication

Description: Users can register and log in to access the system.

Priority: High

5.2. Room Management

Description: Admins can manage room allocations and view room availability.

Priority: High

5.3. Student Management

Description: Admins can manage student records and view their details.

Priority: High

5.4. Fee Management

Description: The system tracks student fees and generates reports.

Priority: High

5.5. Complaint Management

Description: Users can submit complaints, and admins can manage them.

Priority: Medium

5.6. Reporting and Statistics

Description: The system provides real-time statistics and visual reports.

Priority: Medium

6. Other Non-Functional Requirements

6.1. Performance Requirements

The system should support up to 100 concurrent users without performance issues.

6.2. Security Requirements

User passwords must be hashed and salted before storage.

Program Outcomes

→ PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

→ PO2: Problem analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

→ PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

→ PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

→ PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

→ PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse, and in multidisciplinary settings.

→ PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in a multidisciplinary environment.