“HOSTEL MANAGEMENT SYSTEM”

PROJECT REPORT

BY: TEAM ASTERIKS 1

SL NO	NAME	ROLL NO
1	ANAND RAJ	IIT2022047
2	ROHAN PHUTKE	IIT2022048
3	AVANISH GURJAR	IIT2022049
4	AMAN RAJ	IIT2022050
5	PIYUSH PRIYADARSHI	IIT2022051
6	SAURAV GITTE	IIT2022066

1) Introduction:

This project aims to digitalize the room allotment process for students facing difficulties in securing accommodation after semester breaks. It offers a comprehensive solution by integrating features like an in-out system and a help desk for room maintenance, alongside providing essential hostel facilities and amenities. Leveraging web development technologies, the project streamlines hostel management operations and enhances the overall living experience for students.

1.1 Purpose:

The purpose of this project is to address the challenges students encounter in securing hostel rooms after semester breaks. By digitalizing the room allotment process and introducing additional features such as an in-out system and a help desk, the project aims to streamline hostel management operations, reduce confusion in room allotment, and improve the overall living experience for students

1.2 Scope:

Inscope:

1 - Integration with academic databases for student information.

2 - Development of a digital room allotment system.
3 - Implementation of an in-out system for tracking students' movements.
4 - Integration of a help desk ticketing system for room maintenance requests.
5 - Real-time room status updates.
6 - User authentication and secure login methods.
7 - Provision of essential hostel facilities and amenities.
8 - Communication platform for students and administrators.
9 - Feedback mechanism for continuous improvement.
10 - Training sessions/documentation for users.
11 - Support system for technical assistance.

Outscope:

Physical infrastructure changes in hostels.

- Maintenance of hardware devices (e.g., sensors for room occupancy).

- Legal and administrative processes beyond digitalization.

- Student behavior management beyond the scope of the system.

1.3 Definitions, Acronyms, and Abbreviations:

NodeJS: A runtime environment for executing JavaScript code server-side.

- Express: A web application framework for Node.js.

- MySQL: An open-source relational database management system.

- HTML/CSS/JavaScript: Standard technologies for building web applications.

- ReactJS: A JavaScript library for building user interfaces.

- Git: A version control system for tracking changes in source code.

- VS Code: Visual Studio Code, a popular code editor.

- Inscope: Features and functionalities included in the project's development scope.

- Outscope: Activities or requirements not included in the project's development scope

1.4 References

IEEE SRS Format

2 Overview

2.1 Product Perspective:

The project aims to provide a comprehensive digital solution to address the challenges faced by students in securing hostel accommodations after semester breaks. It offers an integrated platform that streamlines room allotment processes, enhances hostel management, and improves the overall living experience for students. From a product perspective, several key aspects need to be considered:

Integration with Existing Systems: The digital room allotment system needs to seamlessly integrate with existing academic databases and management systems to ensure accurate and up-to-date information regarding student accommodations and room availability.

User Authentication and Security: Robust user authentication mechanisms are essential to ensure that only authorized individuals can access the system. Data security measures must be implemented to protect sensitive information such as student records and personal details.

User-Friendly Interface: The user interface should be intuitive and easy to navigate, catering to the diverse needs of students, hostel administrators, and maintenance staff. Clear and concise displays of room availability, maintenance requests, and in-out statuses are crucial for usability.

Real-Time Updates and Communication: The system should provide real-time updates on room availability and status changes to minimize confusion and streamline communication between students and hostel authorities. Integration with communication platforms can facilitate instant notifications and alerts.

Scalability and Reliability: As the number of users and data volume may vary significantly, the system should be designed to scale effectively and handle peak loads without compromising performance. Robust backend architecture and database management are essential for reliability.

Continuous Improvement and Feedback Mechanisms: Regular feedback from users should be incorporated into the development process to identify areas for improvement and prioritize feature enhancements. Continuous iteration and refinement are necessary to meet evolving user needs and address emerging challenges.

Training and Support: Comprehensive training sessions and documentation should be provided to users to ensure they can effectively utilize the system. Additionally, a dedicated support system should be in place to address user queries, troubleshoot issues, and provide assistance as needed.

Overall, the product perspective emphasizes the need for a user-centric approach, seamless integration with existing systems, robust security measures, scalability, and continuous improvement to deliver a successful digitalized room allotment system and hostel management platform

2.2 Product Functions

User Authentication	Ensure only authorized users (students, hostel administrators, maintenance staff) can access the system. This includes login functionality, password management, and user role assignment.
Room Availability	Display up-to-date information on available rooms, including room types, occupancy status, and any special features. Users can search for vacant rooms based on their preferences and requirements.
Room Allotment	Facilitate the digital allotment of rooms to students after semester breaks. Hostel administrators can assign rooms to students based on availability and eligibility criteria, ensuring a fair and efficient allocation process
Maintenance Requests	Provide a help desk ticketing system for students to report maintenance issues or request assistance with room-related problems. Hostel maintenance staff can receive, prioritize, and resolve these requests in a timely manner, improving overall hostel maintenance.
In-Out System	Implement a system to track the entry and exit of students from hostel premises. This allows hostel authorities to monitor occupancy levels, ensure security, and manage guest access effectively.
Real-Time Updates	Enable real-time updates on room availability and status changes to minimize confusion and facilitate quick decision-making. This includes notifications for new allotments, room

	maintenance updates, and any changes in occupancy status.
Communication Platform	Integrate with communication platforms (e.g., email, messaging apps) to facilitate instant notifications, alerts, and announcements. This enhances communication between students, hostel authorities, and maintenance staff, improving overall coordination and information sharing
Feedback Mechanism	Provide a mechanism for users to submit feedback and suggestions for system improvements. Hostel administrators can review and prioritize these suggestions to enhance the functionality and usability of the system based on user input
Training and Support	Offer comprehensive training sessions and documentation to educate users on how to effectively utilize the system. Additionally, establish a dedicated support system to address user queries, troubleshoot issues, and provide assistance as needed

2.3 User Characteristics

1. Students: The primary users of the system are students who require hostel accommodations after semester breaks. They may vary in technological proficiency but generally seek an intuitive and user-friendly interface to easily navigate through room availability, submit room allotment requests, report maintenance issues, and access essential hostel facilities.

2. Hostel Administrators: These users are responsible for managing hostel operations, including room allotment, maintenance, and overall administration. They require access to administrative features such as managing room availability, assigning rooms to students, tracking maintenance requests, monitoring in-out statuses, and communicating with students.

3. Maintenance Staff: Maintenance staff members are tasked with addressing maintenance issues and ensuring the upkeep of hostel facilities. They interact with the system to receive, prioritize, and resolve maintenance requests submitted by students, updating the status of each request as it progresses.

4.System Administrators: These users are responsible for the overall management and maintenance of the digitalized room allotment system. They handle tasks such as user authentication, database management, system configuration, and ensuring data security and privacy.

5. Academic Authorities: In some cases, academic authorities may also interact with the system, especially in terms of providing data for integration with academic databases and overseeing the overall functioning of hostel management systems within educational institutions.

Overall, the user characteristics encompass a diverse range of stakeholders involved in hostel management, each with specific roles and requirements within the digitalized room allotment system. The system must cater to the needs of these users by providing relevant features, accessibility, and ease of use to effectively address the challenges faced during semester breaks and streamline hostel management operations.

2.4 General constraints of the project include:

1. Time limitations within a semester timeframe.

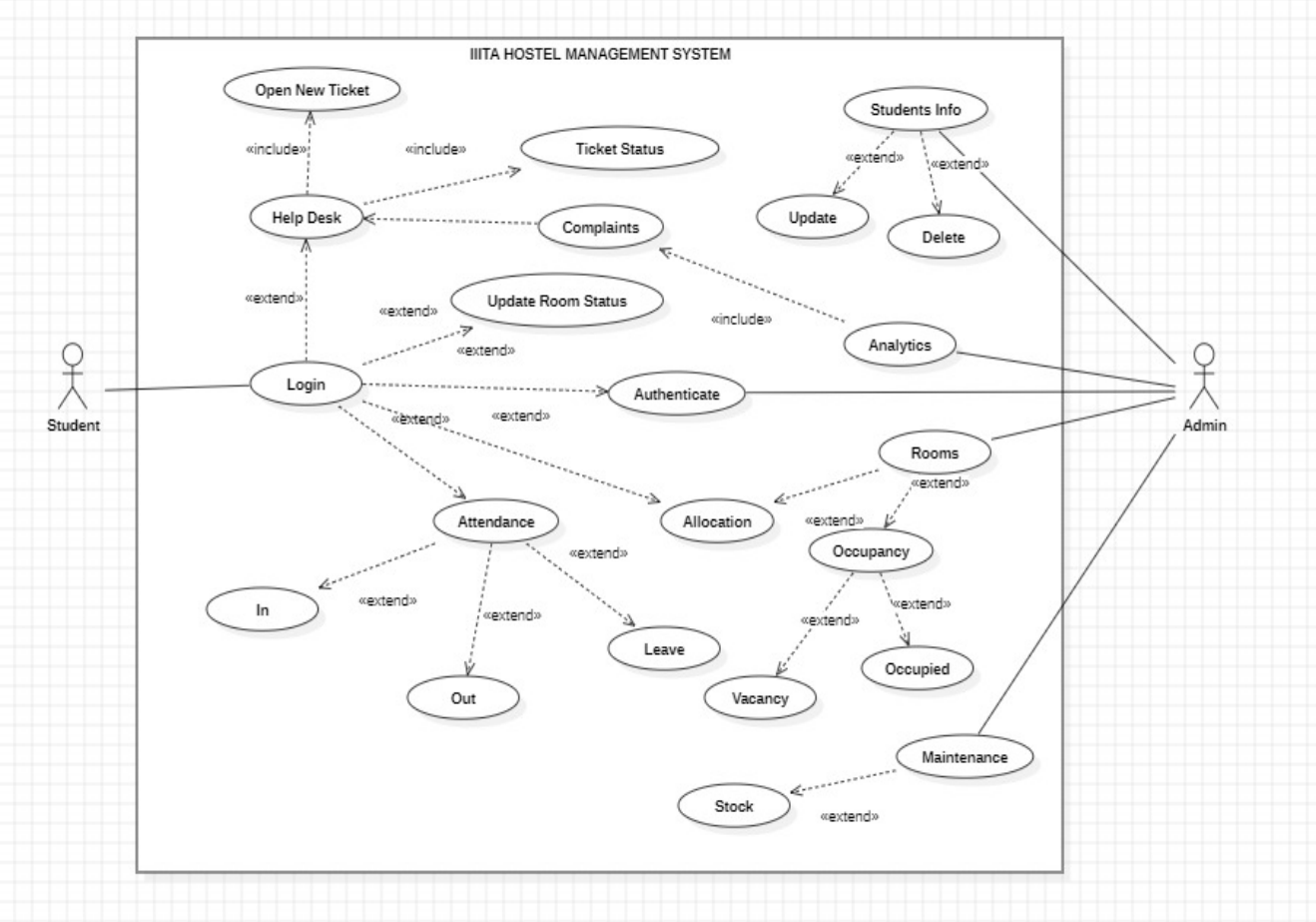
2. Resource constraints with 4-6 individuals collaborating.

3. Technological limitations in compatibility and scalability.

4. Data security and privacy regulations.

5. Operational challenges in integrating existing systems and managing real-time updates

APPENDIX A: USE CASE DIAGRAM



3. Specific Requirements:

3.1 Functional Requirements:

We describe the functional requirements by giving various use cases.

Use Case 1: Open New Ticket

Summary: Allows users to open a new ticket for hostel-related issues or requests.

Actors: Students, Hostel Residents

Preconditions-:

- User must be logged in.

- Hostel management system must be accessible.

Main Success Scenario:

1. User navigates to the "Open New Ticket" section.

2. User provides necessary details such as issue description, room number, and contact information.

3. User submits the ticket.

4. The system records the ticket and assigns a unique identifier.

5. Confirmation message is displayed to the user.

Extensions:

- If required information is missing or invalid, the system prompts the user to correct the errors.

Postconditions:

- A new ticket is created and added to the help desk queue for resolution by administrators.

Use Case 2: Check Ticket Status

Summary: Allows users to check the status of their previously submitted tickets.

Actors: Students, Hostel Residents

Preconditions:

- User must be logged in.

- At least one ticket must have been previously submitted by the user.

Main Success Scenario:

1. User navigates to the "Ticket Status" section.

2. User enters the unique ticket identifier or selects from a list of submitted tickets.

3. The system retrieves and displays the current status of the ticket.

4. User views details such as resolution progress and updates.

Extensions:

- If the ticket identifier is not found or invalid, the system notifies the user accordingly.

Postconditions:

- User is informed of the current status of their ticket, facilitating communication and resolution tracking

Use Case 3: Update Room Status

Summary: Allows administrators to update the status of hostel rooms (e.g., vacant, occupied, under maintenance).

Actors: Hostel Administrators

Preconditions:

- Administrator must be logged in.

- Access permissions must be granted for room status updates.

Main Success Scenario:

1. Administrator navigates to the "Update Room Status" section.

2. Administrator selects the room to be updated.

3. Administrator chooses the new status from predefined options (e.g., vacant, occupied, under maintenance).

4. The system updates the room status accordingly.

5. Confirmation message is displayed to the administrator.

Extensions:

- If the selected room is unavailable or inaccessible, the system notifies the administrator.

Postconditions:

- Room status is successfully updated in the system, reflecting the current occupancy and maintenance status.

These use cases are tailored to the functionalities ,focusing on key actions such as ticket management, room status updates, and ticket status tracking.

3.2 Non-Functional Requirements:

1. Performance:

- The system responds to user actions within 2 seconds under normal load conditions.
- Real-time room status updates to be displayed without delay.

2. Usability:

- The user interface to be intuitive and easy to navigate, requiring minimal training for hostel staff and students.
- The system supports multiple languages to accommodate users from diverse backgrounds.

3. Reliability:

- The system has a uptime of at least 99.9% to ensure availability at all times, especially during critical periods like room allotment.
- Data integrity to be maintained to avoid errors in room allotment or in-out system records.

4. Security:

- User authentication to be implemented securely to prevent unauthorized access to sensitive information.
- Data transmission to be encrypted using SSL/TLS protocols to protect user privacy.

5. Scalability:

- The system is able to handle a growing number of users and hostel rooms without a significant decrease in performance.

- Database schema and server architecture to be designed for scalability to accommodate future growth.

3.3 Hardware Requirements:

- Server: A dedicated server or cloud hosting service capable of running NodeJS with Express and MySQL.

- Storage: Sufficient storage space to store hostel and user data, including room availability, in-out records, and help desk tickets.

- Network: Reliable internet connectivity to ensure seamless access to the system from hostel premises and beyond.

3.4 Software Requirements:

- Operating System: Compatible with Linux-based/Windows operating systems for server deployment.

- NodeJS: Latest stable version for server-side application development.

- Express: Web application framework for NodeJS to handle routing and middleware.

- MySQL: Database management system for storing hostel and user data securely.

- HTML/CSS/JavaScript: For frontend development, with additional use of ReactJS for building interactive user interfaces.

- Git: Version control system for collaborative development and code management.

- IDE: Visual Studio Code (VS Code) or any other preferred integrated development environment for coding and debugging.

3.5 Design Constraints:

- Data Security: Compliance with data protection regulations such as GDPR or local privacy laws.

- Scalability: Architecture and database design to be scalable to accommodate future growth in user base and hostel infrastructure.

- User Accessibility: Design to should be inclusive and accessible to users with disabilities, complying with accessibility standards.

- *Integration: Compatibility with existing academic databases or systems for seamless data exchange and integration.

- User Experience: Focus on delivering a seamless and intuitive user experience to enhance adoption and usability among hostel staff and students.

By addressing these non-functional requirements, hardware and software requirements, and design constraints, the hostel management system can be developed effectively, ensuring reliability, performance, security, and scalability while meeting the needs of users and stakeholders.