

Hostel Mess Management System: A Web-Base Solution For Efficient Hostel Operations

Suyash Kakade¹, Aniket Kandekar², Rushikesh Khetmalis³, Sachin Jadhav⁴, Karim Mulani⁵

^{1, 2, 3, 4, 5} Dept of Information Technology

^{1, 2, 3, 4, 5} Savitribai Phule Pune University, Sinhgad Institute of Technology, Lonavala, India

Abstract- *The Hostel Mess Management System is a web-based application designed to automate and streamline hostel and mess-related operations in educational institutions. Traditional hostel management relies on manual processes such as registers, paper forms, and offline communication, which often lead to inefficiencies, data errors, and lack of transparency.*

This system provides a centralized platform where students and administrators can interact efficiently through a digital interface. It includes modules for room allotment, attendance tracking, mess menu voting, complaint management, and night-out permissions.

The system is developed using modern technologies such as Spring Boot for backend, React.js for frontend, and MySQL for database management.

Students can mark attendance, vote for preferred meals, and submit complaints online, while administrators can monitor and manage all activities in real time.

The implementation of a menu voting system enhances student participation and satisfaction in mess services. Additionally, the complaint tracking feature ensures accountability and timely resolution of issues. Secure authentication mechanisms protect user data and ensure role-based access control.

The system reduces manual workload, improves data accuracy, and enhances operational efficiency. Overall, the proposed solution provides a scalable and user-friendly approach to modern hostel management, ensuring transparency, efficiency, and improved communication between students and administration.

I. INTRODUCTION

Hostel management is an essential part of educational institutions, involving activities such as room allotment, attendance tracking, mess management, complaint handling, and permission control. In many institutions, these operations are still carried out manually using registers, paper forms, and

verbal communication, which often leads to inefficiencies, data inconsistencies, and delays in decision-making.

With the increasing number of students and hostel facilities, managing these activities manually becomes more complex and error-prone. Issues such as incorrect room allocation, proxy attendance, unorganized complaint handling, and lack of transparency in mess operations are common. Additionally, students have limited involvement in decision-making processes like menu planning, which often leads to dissatisfaction.

To overcome these challenges, there is a need for a centralized and automated system that can manage all hostel-related operations efficiently. The Hostel Mess Management System is designed as a web-based platform that integrates students and administrators into a single digital environment. It allows students to access services such as attendance marking, menu voting, complaint submission, and night-out requests, while administrators can monitor and control all activities through an admin dashboard.

The use of modern web technologies enables real-time updates, secure data management, and improved communication between users. By replacing traditional manual methods with a digital solution, the system enhances accuracy, reduces workload, and ensures transparency in hostel operations. Overall, this project aims to provide a scalable, efficient, and user-friendly system that meets the evolving needs of modern hostel management.

II. RELATEDWORK

Several research studies and systems have been developed to improve hostel and mess management through digital solutions. These works focus on automating administrative tasks, improving data accuracy, and enhancing communication between students and authorities.

A study on digital hostel management systems proposed a web-based platform that automates room allocation, attendance tracking, and complaint handling. The

system demonstrated improved efficiency and transparency compared to manual methods, highlighting the importance of real-time data access and centralized record management.

A full-stack web-based hostel administration system utilized modern technologies such as React and Node.js to develop scalable and responsive applications. The study emphasized role-based access control, real-time updates, and efficient database management for handling large amounts of student data.

Research on mess management systems focused on features like menu voting and complaint submission. These systems allow

Student to participate in decision-making processes and provide feedback, leading to higher satisfaction and reduced food wastage.

Additionally, studies on secure authentication and database optimization highlight the importance of protecting user data and ensuring efficient data retrieval. Techniques such as role-based access control, encryption, and optimized queries improve system reliability and performance.

Overall, existing research shows that digital hostel management systems significantly improve operational efficiency, transparency, and user satisfaction. However, many systems lack full integration of all modules. The proposed Hostel Mess Management System addresses this gap by combining multiple functionalities into a single, unified platform.

III. METHODOLOGY

The Hostel Mess Management System is developed using a structured and modular approach to ensure scalability, efficiency, and maintainability. The system follows a three-tier architecture consisting of frontend, backend, and database layers.

Initially, system requirements were analyzed based on common problems in manual hostel management such as inefficient room allocation, inaccurate attendance tracking, and lack of transparency. Based on these requirements, the system was divided into independent modules including room allotment, attendance management, mess menu voting, complaint handling, and night-out permission.

The frontend layer is developed using React.js, providing an interactive and user-friendly interface for both students and administrators. Students can perform actions such

as marking attendance, voting for menus, submitting complaints, and requesting permissions. Administrators access a dashboard to manage rooms, monitor attendance, approve requests, and resolve complaints.

The backend layer is implemented using Spring Boot, which handles all business logic and processes user requests through REST APIs. It validates user inputs, manages authentication using JWT, and ensures secure communication between frontend and database.

The database layer uses MySQL to store all system data, including student records, room details, attendance logs, menu votes, complaints, and permission requests. CRUD operations are performed efficiently through the backend using Spring Data JPA.

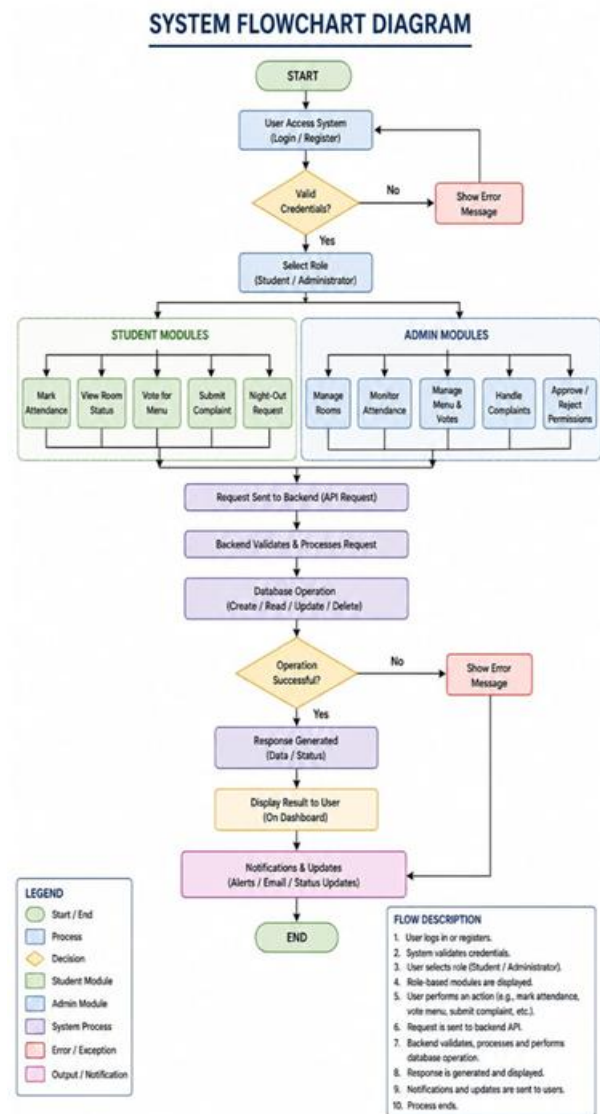


Figure1: System Flowchart of Hostel Mess Management System.

IV. ARCHITECTURE

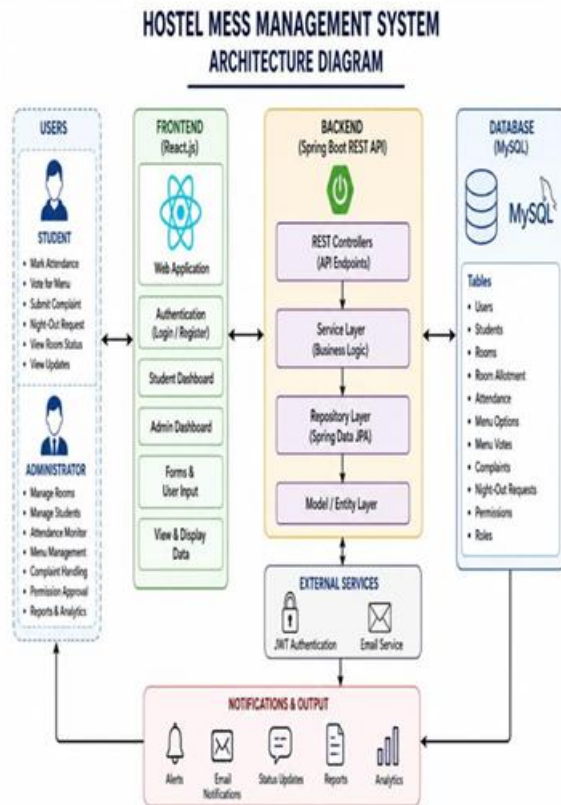


Figure2: System Architecture Of Hostel Mess Management System.

V. BACKEND AND URL HANDLING

The backend of the Hostel Mess Management System is developed using Spring Boot, which manages all core functionalities and business logic. It exposes RESTful APIs that allow the frontend to communicate with the server efficiently. When a user performs an action (such as marking attendance or submitting a complaint), a request is sent to a specific API endpoint. The backend first validates the request, including user authentication and authorization using JWT tokens.

URL handling is performed through well-defined API routes (e.g., /login, /attendance, /complaints), which map user requests to corresponding controller methods. These endpoints process

HTTP methods like GET, POST, PUT, and DELETE. The backend then forwards the request to the service layer, where the main logic is executed.

After processing, the system interacts with the database through the repository layer to store or retrieve data.

Finally, a structured response (JSON format) is sent back to the frontend. This approach ensures secure, organized, and efficient handling of all user requests within the system.

VI. CONTENT EXTRACTION

Content extraction in the Hostel Mess Management System refers to the process of collecting, validating, and structuring user input data before it is processed or stored in the system. Data is received from the frontend through forms such as login, attendance marking, menu voting, complaint submission, and permission requests.

The backend extracts this data from HTTP requests (JSON format) using REST API controllers. It then performs input validation to ensure correctness, completeness, and security, such as checking required fields, formats, and user authorization. After validation, the data is transformed into appropriate models or entities for further processing.

The service layer processes the extracted content based on business logic, such as updating attendance records or storing complaints.

VII. USER INTERFACE DESIGN

The Hostel Mess Management System utilizes a dashboard-centric user interface (UI) designed for clarity and role-specific functionality. The interface is built using React.js, ensuring a responsive and modular design that provides real-time status feedback for both students and administrators.

1. Student Interface

The student interface acts as a personalized service portal, allowing residents to interact with hostel and mess operations digitally.

Service Cards: A grid-based layout provides quick access to core services such as "View Hostels," "Mess Menu," "Attendance," "Night Out," "Complaints," and "Holiday Request."

Mess Menu Voting: Students can view current options and participate in democratic menu selection via a dedicated voting module.

Request Submissions: Students can submit digital forms for night-out permissions or food-related complaints, eliminating the need for physical registers.

Real-Time Status: The interface displays immediate updates on the status of submitted requests (e.g., "Pending," "Approved," or "Resolved").

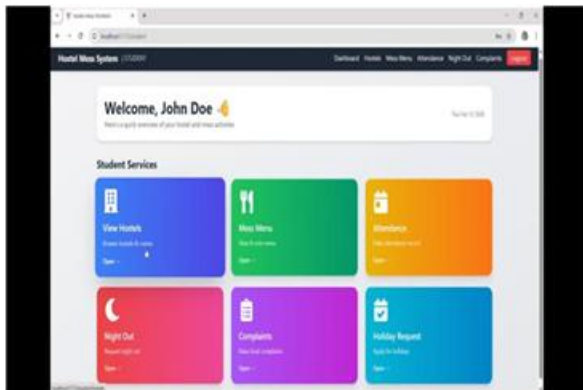


Figure3: Student Interface

2. Admin Interface

The admin interface is a management-heavy dashboard designed to provide wardens and hostel heads with full visibility and control over institutional operations.

Management Modules: The dashboard is categorized into administrative tasks including "Hostel Management," "Room Management," and "Attendance Management".

Approval Pipeline: Admins can instantly review, approve, or reject pending student requests for night-outs and holidays.

Mess Control: Administrators use this interface to monitor voting trends, declare the final mess menu, and respond to grievances logged in the complaint system.

Analytics and Monitoring: The system provides a bird's-eye view of room occupancy, student behavior patterns, and overall hostel presence.



Figure3: Admin Interface

VIII. RESULTS AND EVALUATION

The Hostel Mess Management System was successfully implemented and tested to evaluate its performance, usability, and efficiency. The system demonstrated significant improvements over traditional manual methods by automating key hostel operations such as attendance tracking, room allotment, complaint handling, and mess menu management.

During testing, the system showed accurate data processing with minimal errors due to automated validation and structured workflows. The response time for user requests was observed to be fast, ensuring smooth interaction between frontend and backend. Real-time updates allowed both students and administrators to access the latest information without delays.

IX. PERFORMANCE AND RELIABILITY

The Hostel Mess Management System is designed to deliver high performance and reliable operation under regular usage conditions. The system efficiently handles multiple user requests simultaneously through optimized backend processing using Spring Boot and REST APIs. Fast response time is achieved by minimizing unnecessary data processing and using efficient database queries, ensuring smooth interaction for both students and administrators.

The use of a structured database (MySQL) ensures quick data retrieval and consistency across all modules such as attendance, complaints, and room allocation. Caching and proper indexing techniques further enhance system performance by reducing load time. The frontend, developed using React.js, provides a responsive user interface that updates dynamically without reloading entire pages.

In terms of reliability, the system includes robust error handling and validation mechanisms to prevent system failures and incorrect data entries. Authentication and authorization using JWT ensure secure and controlled access. The system is capable of handling concurrent users without performance degradation, making it suitable for large-scale hostel environments.

X. USER INTERFACE DEMONSTRATION

Pilot testing of the system demonstrates significant advantages over traditional methods:

Metric	Manual System	HostelConnect (Automated)
Room Allotment	Manual register slow	Real time instant
Attendance	Paper-based, prone to proxy	Digital, Secure records
Mess Menu	Decided by authorities	Student-led voting
Security	Hard to track night-outs	Structured approval history

XI. LIMITATION AND FUTURE WORK

Limitations: The current system has some limitations. It depends on continuous internet connectivity for accessing features and real-time updates. The system does not include biometric or hardware-based attendance, which may allow misuse in some cases. It is primarily designed for web use and lacks a dedicated mobile application. Additionally, handling very large-scale hostel data may require further optimization and cloud-based deployment.

Future Work: Future enhancements can improve the system significantly. A mobile application can be developed for better accessibility. Integration of biometric or RFID-based attendance can increase accuracy and security. Online payment features for mess fees and hostel charges can be added. Advanced analytics and reporting dashboards can help administrators make better decisions. Cloud deployment and AI-based features like predictive meal planning can further enhance system performance and usability.

XII. CONCLUSION

The Hostel Mess Management System provides an efficient and reliable solution to automate hostel and mess-related operations. By replacing manual processes with a digital platform, the system improves accuracy, transparency, and overall management efficiency. Features such as attendance tracking, menu voting, complaint handling, and permission management enhance both administrative control and student participation.

The use of modern technologies ensures scalability, security, and real-time data access. The system reduces workload for administrators while providing a user-friendly interface for students.

In conclusion, the proposed system successfully addresses the limitations of traditional hostel management and offers a practical, scalable, and effective solution for modern institutions.

REFERENCES

- [1] Patil, R., & Deshmukh, A. (2022). Design and Implementation of Digital Hostel Management System. *International Journal of Computer Applications*.
- [2] Sharma, P., & Nair, S. (2022). IoT-Based Smart Hostel Monitoring System. *International Journal of Engineering Research & Technology*.
- [3] Kulkarni, M., & Jadhav, P. (2023). Full Stack Web-Based Hostel Management Application. *Journal of Web Engineering and Technology*.
- [4] Pawar, S., & Khandekar, R. (2023). Mess Management System with Menu Voting Feature. *International Journal of Innovative Research in Computer Science*.
- [5] Gupta, R., & Singh, P. (2022). Secure Web Applications Using Role-Based Access Control. *Journal of Information Security*.
- [6] Oracle Corporation. (2023). MySQL Documentation. Available at: <https://dev.mysql.com/doc/>
- [7] Spring Framework. (2023). Spring Boot Documentation. Available at: <https://spring.io/projects/spring-boot>
- [8] Facebook Inc. (2023). React.js Documentation. Available at: <https://react.dev/>
- [9] Postman Inc. (2023). Postman API Platform Documentation. Available at: <https://www.postman.com/>
- [10] Microsoft. (2023). Visual Studio Code Documentation. Available at: <https://code.visualstudio.com/>