

Software Requirements Specification



Ahmedabad
University

CSE300 Software Engineering

Submitted to Faculty: Prof. Khushru Doctor

Name	Enrollment Number	Program
Digant Patel	AU2040086	B Tech (CSE)
Madhvendrasinh Jhala	AU2040162	B Tech (CSE)
Yug Patel	AU2040181	B Tech (CSE)
Deep Patel	AU2040250	B Tech (CSE)
Parth Zinzuwadia	AU2140233	B Tech (CSE)

Table of Contents

1. Introduction.....	3
1.1 Problem Statement.....	3
1.2 Product Scope.....	3
1.3 Aims & Objectives.....	4
1.4 Definitions, Acronyms and Abbreviations.....	4
2. Description.....	4
2.1 Product Perspective: Hostel Management Software.....	4
2.1.1 Existing System Function.....	4
2.2 Operating Environment.....	5
2.2.1 Server side.....	5
2.2.2 Client side.....	5
2.3 Design and implementation constraints.....	6
2.4 User Documentation.....	6
2.5 Assumptions and Dependencies.....	6
2.5.1 Regulatory Policies.....	6
2.5.2 Hardware Limitations.....	7
3. Proposed System.....	7
3.1 Stakeholder Requirements.....	7
3.2 Business Requirements.....	8
3.3 Functional Requirements.....	8
3.3.1 User(FR1).....	8
3.3.2 Administration(FR2).....	8
3.3.3 Student(FR3).....	8
3.3.4 Mess Management(FR4).....	8
3.3.5 Housekeeping(FR5).....	9
3.3.6 Security(FR6).....	9
3.4 Non-Functional Requirements.....	9
3.5 External-interface Requirements.....	10
3.6 Performance Requirements.....	10
3.7 Intended Audience.....	11

1. Introduction

In the dynamic landscape of educational institutions, managing hostel facilities efficiently is a crucial aspect of ensuring a seamless student experience. Recognizing the complexities involved in hostel administration and the need for a centralized solution, we present a web-based application designed to streamline hostel management processes and foster effective communication between students, administrators, and support staff. This intuitive software is dedicated to optimizing the hostel experience by automating administrative tasks, enhancing communication channels, and providing a seamless interface for all stakeholders.

1.1 Problem Statement

The current scenario in hostel management lacks a centralized platform that seamlessly connects hostel administrators, students, and support staff. Hostel operations involve numerous processes, such as student registrations, room assignments, mess management, and housekeeping requests, which can be cumbersome to fulfill manually and less efficient without a dedicated system. The unified solution will solve the problem in communication gaps, manual errors, and the lack of transparency.

1.2 Product Scope

This software takes into account most of the problems present in the current scenario and hence helps to automate the process of many hostel services with very little overhead for the management team.

The areas covered are as follows:

- Students residing in the hostel: The new software would help the students to easily access the facilities and make their stay better.
- The Hostel Managers: The managers would be able to easily manage and maintain hostel facilities efficiently.
- The University Cafeteria Team: The team would be able to get a better and clearer idea for planning the logistics for the meals.

1.3 Aims & Objectives

- To enhance communication channels between hostel administrators and students through a dedicated portal.
- Streamline the process of new student registration and room assignments.
- Provide students to use the facility at their convenience.
- An admin (hostel manager) dashboard for efficient management of hostel operations.
- Implement features to optimize mess management processes, including accurate headcount monitoring and credential verification during meals.

1.4 Definitions, Acronyms and Abbreviations

TERM	DEFINITION
Hostel Manager	Hostel Manager who supervises the smooth functioning of all facilities and is the admin of the proposed system.
Student	Student who accesses the facilities.
Mess Manager	Mess-Manager who is responsible for the logistic part of the meals.
Housekeeping Service Manager	Housekeepers who are responsible for the cleaning part of the hostel.

2. Description

2.1 Product Perspective: Hostel Management Software

2.1.1 Existing System Function

- In the present scenario there are many demands that are not being fulfilled and leads to dissatisfaction for the students.

- In the current scenario the students have to manually find cleaners and request the cleaners to come and clean the room. And many times the situation arises that already a large number of students have made a request and hence sometimes the students have to wait a long time for getting their room cleaned.
- Other than that the students have to manually collect food-coupons from the admin office for their lunch and high-tea and the admin has to keep a count of the number of students who collected the respective coupons. After the meal time the coupons have to be manually checked.
- Also the ADMIN has a lot of workload related to allotment and changing requests of rooms by the student and there is no way to supervise the hostel management efficiently.

2.2 Design and implementation constraints

- The application will use Next.js, React router, React Hooks, Node.js, MySQL, and Apollo GraphQL as web technologies.
- Since the website is not made responsive to the screen-size and hence it should be accessed by only large screen-sized devices only.
- HTTP protocol is used as a communication protocol.
- Several types of validations and security measures make this web application a secure one.
- Since the Connecting platform is a web-based application, an internet connection must be established.
- The Connecting platform will be used on PCs and will function via the internet or intranet in any web browser and can be accessed from anywhere and with any devices like laptops or tablets.

2.3 User Documentation

The website is made in such a way by keeping the concept of Human-Computer Interaction that it is made very easy for any user to access the services and navigate between web services. Hence no documentation or further help might be required by any users. In case there is a problem or difficulty that is being faced by a student or any other user then the person can reach-out to the hostel manager and their queries would be solved.

2.4 Assumptions and Dependencies

2.4.1 Regulatory Policies

All the students who would be taking admission to the university hostel would have to register themselves with the admin and then they would be able to access the website. After that, with the proper authentication model, the students will be able to use the facilities provided by the website. The admin would also create accounts for cleaners as well.

2.4.2 Hardware Limitations

There is no limitation in the operating system in which the connecting system will work. However, a proper internet connection is required to access the facilities provided by the hostel management. Users can access the system with any internet browser. The server needs a continuous and stable internet connection to provide services to its users. There will also be some minimum hardware requirements for the server based on the number of users.

3. Proposed System

3.1 Stakeholder Requirements

Admin Requirements

- Admin should be able to view all the student requests made.
- Admin should be able to make registration of new students.
- Admins should be able to allot rooms to the students.

Student Requirements

- The student should be able to book slots for room cleaning.
- The student should be able to select a place for their lunch and high-tea.

Mess Manager Requirements

- The mess manager should be able to view the lunch and high-tea preferences made by the students.
- The mess manager would scan the qr-code of the student so that they have a note on the number of students who had completed their meal.

Housekeeper Requirements

- The housekeepers would be able to view the requests made by the students for cleaning their rooms.
- They would also get the requests for changing of bed-sheets made by the students.

3.2 Business Requirements

Providing hostellers facilities

Make all the requests online through the website.

Making hostel managers work easy

Direct access to the student's requests and automatic allotment of rooms to the new students. No need to communicate from the hostel to the university cafeteria where the meals are being prepared.

Making mess managers work easy

The mess managers would be able to view the total count of the students who would be having their meals at the university and hostel mess.

3.3 Functional Requirements

3.3.1 User (FR1)

1. The user would be able to login into the system with proper credentials. (FR1.1)
2. Based on the privileges of the user, the user would be redirected to their respective interfaces. (FR1.2)
3. The user would also be able to reset the password associated with their account. (FR1.3)

3.3.2 Administration (FR2)

1. Admin has the functionality of registering different types of new users. (i.e., students, house-cleaners etc.) (FR2.1)
2. Admin can also view different requests being made by the students. (FR2.2)
3. Also be able to make changes in the slots of cleaning service. (FR2.3)
4. When the admin makes credentials for students, the room should be assigned to him automatically. (FR2.4)

3.3.3 Student (FR3)

1. The students will be able to choose the place where they will be having their meals. (FR3.1)
2. The student would be able to make a request for a room-cleaning service. (FR3.2)
3. They would be able to make requests for bed-sheet change also. (FR3.3)
4. They would be able to view their booking for room-cleaning. (FR3.4)

3.3.4 Mess Management (FR4)

1. The mess manager would be able to view the count of student's meal place preference. (FR4.1)
2. The mess manager would be able to mark the students who would mark the students who have taken their meal. (FR4.2)

3.3.5 Housekeeping (FR5)

1. The housekeeping people would be able to view the room-cleaning request made by the student. (FR5.1)
2. They would also be able to view the bed-sheet change requests made by students. (FR5.2)

3.3.6 Security (FR6)

1. A user will not be able to access the interface of another type of user. (FR6.1)
2. JWT authentication would be implemented, and hence every time there would be an API call, the authentications would be made. (FR6)

3.4 Non-Functional Requirements

Usability: The software should have an intuitive and user-friendly interface to ensure that both hostellers and hostel/mess managers can easily navigate and perform their tasks without extensive training.

Performance: The system should be responsive and capable of handling concurrent requests from multiple users, especially during peak hours when many hostellers may be making reservations or meal transactions.

Security: The software should implement robust security measures to protect user data and payment information. It should also ensure that only authorized personnel can access management features.

Scalability: The system should be designed to accommodate an increasing number of users and requests over time as the hostel population grows. It should also support easy integration with additional hostels or mess facilities if needed.

Reliability: The software should be available 24/7, with minimal downtime for maintenance. It should have backup and recovery mechanisms in place to prevent data loss.

Data Privacy: Hosteller information, including personal preferences and meal transaction history, should be kept confidential and comply with relevant data privacy regulations.

Reporting: The system should provide reporting capabilities for mess managers and hostel management to track trends in reservations, meal usage, and cleaning requests, aiding in decision-making and resource planning.

Accessibility: The software should adhere to accessibility standards to ensure that it can be used by individuals with some disabilities, including those who rely on assistive technologies.

Integration: The software should be capable of integrating with existing hostel and university systems, such as student databases, to streamline processes and data exchange.

Maintenance and Support: There should be a well-defined plan for ongoing maintenance, updates, and user support to address any issues or enhancements required by users.

3.5 External-interface Requirements

User-Interfaces: There are four user interfaces namely one for the student user where they would be having screens related to meal-place booking, room cleaning etc. The cleaners would be having a screen where all the requests for room cleaning are there. The mess-manager would have a screen interface to get the students meal preference and the university manager would be having the interface to add new students and view student requests.

Backend: Communication between the user interface and the backend will be provided by Apollo Client. And JWT authentication has been implemented for security purposes.

3.6 Performance Requirements

Static

User Authentication: The system must handle authentication for up to 100 concurrent users, ensuring low latency and high performance during peak usage times.

Storage: The database must store student and room information for a period of 5 years. Regular backups should be performed seamlessly during non-peak hours to ensure data integrity.

Compatibility: The software should be compatible with popular web browsers (Chrome, Firefox, Safari, and Edge).

Dynamic

Facility Booking: The software must support up to 100 simultaneous facility booking requests without degrading performance. The booking confirmation should be provided to users within 10 seconds.

Mess Management: Granting the student access to the cafeteria after scanning their QR within 5 seconds.

3.7 Intended Audience

- **FLH Hostel Management team:** The dean of student housing and the entire team would be able to use the website.
- **Hostel Manager:** The hostel manager supervising the facilities of the hostel would be an intended audience for the website.
- **Student:** The students who are admitted into the university hostel.

