

Software Requirement Specification (SRS)

Group-12

Govinda Rohith Y (CS21BTECH11062)

R Bhargava Ram (CS21BTECH11052)

P Pranav (CS21BTECH11062)

Teja (CS21BTECH11059)

Contents

1	Introduction:	2
1.1	Purpose of Hostel Room Management System:	2
1.2	Scope:	2
1.3	Abbreviations:	2
1.4	Overview:	3
2	Description:	3
2.1	Product Perspective:	3
2.2	Product Functions:	4
2.3	Principal Actors and their Characteristics:	5
2.3.1	Student :	5
2.3.2	Hostel Representative(HR) :	5
2.3.3	Hostel Office(HO) :	5
2.4	Assumptions :	5
3	Requirements:	6
3.1	Functional Requirements:	6
3.1.1	Use cases related to authorization:	6
3.1.2	Use cases related to room management:	7
3.1.3	Use cases related to profiles:	9
3.1.4	Use cases related to complaints/tickets:	9
3.1.5	Use cases related to announcements:	11
3.2	Non-Functional Requirements:	11
3.2.1	Performance:	11
3.2.2	Security:	12
3.2.3	Usability:	12
4	Future Extensions:	12
5	Appendix:	12
5.1	Login page of the website	12
5.2	Home page for student interface	13
5.3	Home page for HR interface	13
5.4	Home page for HO interface	14

1 Introduction:

1.1 Purpose of Hostel Room Management System:

A software program called a "hostel room management system" is made to automate and simplify the operation of hostels and dorms. Enhancing efficiency, organization, and general effectiveness in managing many parts of hostel rooms is the main goal of this kind of system.

The key purpost of the HOrtel Room Management System are:

- Provide an easy-to-use interface for both HR and users to access relevant information and perform necessary tasks.
- Efficiently allocating rooms according to the administrative criteria.
- Manage resident profiles, ensuring accurate and up-to-date records.
- Facilitate smooth check-in and check-out processes for residents.
- Make Announcements for important events, announcements, or policy updates.
- Can raise tickets for complaints.
- Can exchange rooms on mutual Agreement.

1.2 Scope:

In-Scope:

- Room Allocation, Availability and Maintenance.
- Registration and Profile Management.
- Check-In and Check-Out Process.
- Notifications(Announcements).
- Complaints(Raise Tickets).
- Room exchange facility.
- Block Reviews.

Out-Scope:

- Fee and Billing Management.
- Room Reservation for Non-Residents.
- Food Services Management.

1.3 Abbreviations:

- **SRS** : Software Requirements Specification.
- **HRMS** : Hostel Room Management System.
- **HR** : Hostel Representative (per Block).
- **HO** : Hostel Office (Admin).

1.4 Overview:

The rest of the SRS is as follows: Section 2 provides a comprehensive overview of the software, delineating the anticipated user proficiency level, outlining general constraints governing the software development process, and articulating various assumptions and dependencies inherent in the project. Section 3 provides the basic abbreviations using in the coding part. Finally Section 4 gives brief overview about the software which we are developing.

2 Description:

2.1 Product Perspective:

The system provides user interfaces tailored for Students, HR, and Hostel Office, ensuring an intuitive and user-friendly experience. The interfaces support functionalities such as room requests, complaint management, and announcements.

The HRMS will be a standalone system that interacts with the hostel database for storing and retrieving information. It may also integrate with other systems, such as accounting software, AIMS/ERP etc.

2.2 Product Functions:

Class of use cases	Use cases	Description of use cases
Authorization	Login	Login as a student/HR/HO.
	Change password	Change password for user.
	Forgot password	Old password is sent to students email ID.
	Logout	Logout from the website.
	Allot HR	Allot a HR to a block by HO.
Room Management	New room booking	Student can book a room based on availability.
	Room exchange request	Request for room exchange is given by students on mutual agreement.
	Room exchange approval	Approval of room exchange request by HO.
	Check-in/out request	Check-in request after a long break or check-out request for a long break to HO by students.
	Check-in/out approval	Check-in/out approval by HO.
	Hostel block rating	At the time of leaving the block rating is given by students.
Profiles	My profile	Students can see their profile.
	View others profile	Students/HR/HO can View details at pod/block/hostel level respectively.
Tickets	Raise ticket	Students can raise tickets and request for service.
	View ticket status	Students can see status of their tickets. HR/HO can see status at block/hostel level.
	Solve tickets by HR	Close tickets which are solvable by HR.
	Pass tickets	Filter and pass unsolved tickets by HR based on importance to HO.
	Close tickets	The tickets which are filtered by HR are solved and closed by HO.
	Reply to tickets	HR/HO can reply to student tickets accordingly.
Announcements	View announcements	Students can see announcements made by HR/HO. HR can see announcements made by HO.
	Make announcements	Announcements are made by HR at block level/HO at hostel level.

Table 1: Function table

2.3 Principal Actors and their Characteristics:

2.3.1 Student :

- The primary user who resides in the hostel.
- Interacts with the system for tasks such as room requests, room exchanges, and check-in/out procedures.
- Engages with the complaint system for raising and checking the status of issues within the room, pod, or floor.
- Utilizes the facility status feature to check the availability and last serviced date of various amenities.
- Accesses personal room details and reservation information.
- Provides reviews of hostel blocks during check-out.
- Receives announcements from HR and Hostel Office.

2.3.2 Hostel Representative(HR) :

- HR is a student who has been elected to serve and represent a specific hostel block.
- Acts as an intermediary between students and Hostel Office.
- Resolves complaints within their capacity and escalates unresolved issues to the Hostel Office.
- Makes announcements for their assigned hostel block.
- Communicates with Hostel Office for issue escalation.

2.3.3 Hostel Office(HO) :

- Administers and oversees the entire hostel management system.
- Has access to all functionalities available to students and HR.
- Authorizes room exchange and check-in/out requests.
- Manages hostel facilities, their servicing, and resolves escalated complaints.
- Makes announcements for the entire hostel.
- Communicates with HR for issue resolution and block-level announcements.

2.4 Assumptions :

- The HO creates student accounts using the data provided by the academic administration, assuming all the details are correct.
- Users have access to internet-connected devices with standard web browsers.
- The institution's student database provides accurate and up-to-date information.

3 Requirements:

3.1 Functional Requirements:

3.1.1 Use cases related to authorization:

- **Use case 1:** Login

Primary Actor : Student, HR and HO.

Pre-Conditions : Already enrolled student by Academics, HR is already allotted by HO.

Main Scenario :

1. When the user accesses the website, a login and password prompt appears.
2. User enters the user ID and password.
3. System does authentication.
4. Main screen is displayed.

Alternate Scenario :

1. Authorization fails due to wrong username or password.
 - (a) Prompt the user that they typed the wrong username or password.
 - (b) User enters the username and password again.

- **Use case 2:** Change Password

Primary Actor : Student, HR and HO.

Pre-Conditions : User has already logged in.

Main Scenario :

1. User initiates the change password command.
2. User is prompted for new password.
3. User gives the new password and clicks conform.
4. New password is registered with the system.

Alternate Scenario :

1. User deny.
 - (a) After giving new password, user can disagree to change.

- **Use case 3:** Forgot Password

Primary Actor : Student, HR and HO.

Pre-Conditions : Already enrolled student by Academics, HR is already allotted by HO.

Main Scenario :

1. User enters the username and clicks on forgot password.
2. User is prompted for user ID.
3. User gives User ID and clicks conform.
4. System verifies user ID and it sends the old password to connected email.

Alternate Scenario :

1. User ID is wrong.

- (a) As the User ID given is wrong mail is not sent to the user.
- (b) So user re-enters user ID again.

- **Use case 4:** Logout

Primary Actor : Student, HR and HO.

Pre-Conditions : User has already logged in.

Main Scenario :

1. User initiates the log out command.
2. User is logged out.
3. Login page is displayed.

- **Use case 5:** Allot HR

Primary Actor : HO

Pre-Conditions : Already enrolled student by Academics.

Main Scenario :

1. HO is prompted for user ID of student for each block.
2. HO gives user ID of student for each block respectively.
3. Then system sends mail to the user ID of respective student with credentials of HR account.

Alternate Scenario :

1. Allotted HR is wrong.
 - (a) If the some other person is allotted as HR by mistake.
 - (b) HO can re-allot the correct one.

3.1.2 Use cases related to room management:

- **Use case 6:** New room booking

Primary Actor : Student

Pre-Conditions : New student just logged in.

Main Scenario :

1. Student is prompted to select a block.
2. Based on the selected block, system prompts to select a floor.
3. After selecting floor and sending request, if there is a available room system allocates room to the student.

Alternate Scenario :

1. When room is not allocated.
 - (a) Rooms are allocated in first-come first-serve basis, if all rooms are filled in the requested floor.
 - (b) System prompts student to select block again.

Note : Through this whole process the no.of available room in each block/floor and rating of each block is shown to the student. Updates dynamically.

- **Use case 7:** Room exchange request

Primary Actor : Student

Pre-Conditions : Old students (Already have a room).

Main Scenario :

1. Student is prompted to select a block and enter room number.
2. System validates the room number.
3. Student then sends the request to HO. (NO multiple requests)

Alternate Scenario :

1. When room number entered is not valid.
 - (a) If entered room number is invalid, system prompts to re-enter correct room number.

- **Use case 8:** Room exchange approval

Primary Actor : HO

Pre-Conditions : Old students (Already have a room).

Main Scenario :

1. System finds the mutual requests and then show HO.
2. HO is prompted to agree/disagree for exchange for each request shown.
3. Then HO gives agree/disagree.

Alternate Scenario :

1. When system can't find mutual requests.
 - (a) If system can't find mutual requests (Time constraint), then denial is sent to the students request.

- **Use case 9:** Check-in/out request

Primary Actor : Student

Pre-Conditions : Old students (Already have a room).

Main Scenario :

1. Student is prompted to give reason, duration and proof(File) for reason.
2. Student gives reason, duration(from-to) and proof.
3. Student then sends the request to HO. (NO multiple requests)

Alternate Scenario :

1. Invalid proof/duration.
 - (a) If attached proof is invalid or not sufficient/ duration is incorrect.
 - (b) HO rejects the request.
 - (c) After rejection, once again request can be sent.

- **Use case 10:** Check-in/out approval

Primary Actor : HO

Pre-Conditions : Old students (Already have a room).

Main Scenario :

1. HO is prompted to agree/disagree for check-in/out for each request shown.
2. After checking the valid proof and duration.
3. HO approves the request.

Alternate Scenario :

1. Invalid proof/duration.
 - (a) If attached proof is invalid or not sufficient/ duration is incorrect.
 - (b) HO can reject the request.

- **Use case 11:** Hostel block rating

Primary Actor : Student

Pre-Conditions : Old students (Already have a room).

Main Scenario :

1. Student is prompted to give rating for the block in which stays only.
2. student gives rating.

3.1.3 Use cases related to profiles:

- **Use case 12:** My profile

Primary Actor : Student

Pre-Conditions : Old students (Already have a room).

Main Scenario :

1. Student can view his personal details like Name, User ID, block, floor, room number.
2. Status of amenities in the room like bed, table, chair, LAN, Current, fan etc.

Alternate Scenario :

1. Invalid data.
 - (a) If the details show are incorrect, then the student can correct his personal details.

- **Use case 13:** View others profile

Primary Actor : Student

Pre-Conditions : Old students (Already have a room).

Main Scenario :

1. Student can view his pod mates details like Name, User ID etc.
2. Status of amenities/services in the floor like house keeping, water purifier, washing machine etc.

3.1.4 Use cases related to complaints/tickets:

- **Use case 14:** Raise ticket

Primary Actor : Student

Pre-Conditions : Old students (Already have a room).

Main Scenario :

1. Student is prompted to select type, subject and description of ticket.

2. Student selects the type, gives subject and description of ticket.
3. Finally student raises the ticket.

Alternate Scenario :

1. If ticket is closed.
 - (a) If the ticket is closed without solving the problem completely.
 - (b) New ticket can be raised.
2. Student can close ticket.
 - (a) If the problem is solved before the service by HO.
 - (b) Student can close the ticket by himself.

- **Use case 15:** View ticket status

Primary Actor : Student, HO and HR

Pre-Conditions : One or more than one tickets raised.

Main Scenario :

1. Students can view all the tickets they raised.
2. HR/HO can view all the active tickets raised at block/hostel level respectively.
3. HR/HO can sort the tickets based on the type of ticket.
4. Further the status of each ticket can be viewed.

- **Use case 16:** Solve tickets by HR

Primary Actor : HR

Pre-Conditions : One or more than one tickets raised.

Main Scenario :

1. HR can view all the active tickets raised at block level.
2. HR can solve the problem concerned by the ticket and then close it.
3. Further HR can sort for a particular type of ticket and solve them.

- **Use case 17:** Pass ticket

Primary Actor : HR

Pre-Conditions : One or more than one tickets raised.

Main Scenario :

1. HR can view all the active tickets raised at block level.
2. The tickets which are important and cannot be solved by HR.
3. Then HR can forward those tickets to HO to solve them.

- **Use case 18:** Close ticket

Primary Actor : HO

Pre-Conditions : One or more than one tickets raised.

Main Scenario :

1. The tickets which are important and cannot be solved by HR can be viewed by the HO.
2. Then HO can sort and solve the problems by send the required staff for the problem.

3. After solving the problem, the ticket is closed by HO.

- **Use case 19:** Reply to tickets

Primary Actor : HO and HR

Pre-Conditions : One or more than one tickets raised.

Main Scenario :

1. HR/HO can reply to the ticket raised by student before closing the ticket by sending the details of staff who will solve the work.
2. Or can add some message at the time of closing ticket.(at most one reply)

3.1.5 Use cases related to announcements:

- **Use case 20:** View announcements

Primary Actor : Student, HR and HO

Pre-Conditions : Already logged into the site.

Main Scenario :

1. Students can see announcements made by HR/HO.
2. HR can see announcements made by him and HO.
3. HO can see announcements made by him.

- **Use case 21:** Make announcements

Primary Actor : HR and HO

Pre-Conditions : Already logged into the site.

Main Scenario :

1. HR/HO is prompted to give title, description and live-date.
2. HR/HO gives title, description and live-date for the announcement.
3. HR can make announcements for students at block level.
4. HO can make announcements for students at hostel level.
5. HO can make announcements for HR's only.

Alternate Scenario :

1. If announcement made is wrong.
 - (a) HR/HO can check the announcements as said above, if they find the details wrong.
 - (b) They can delete the announcement and re-post with correction.

3.2 Non-Functional Requirements:

3.2.1 Performance:

- System should support a minimum of 100 simultaneous users.
- Response time should be fast enough.

3.2.2 Security:

- Data should be stored securely with appropriate encryption.
- Access controls should be in place to restrict unauthorized access.

3.2.3 Usability:

- The user interface should be intuitive and easy to navigate.
- The system should be accessible on multiple devices and browsers.

4 Future Extensions:

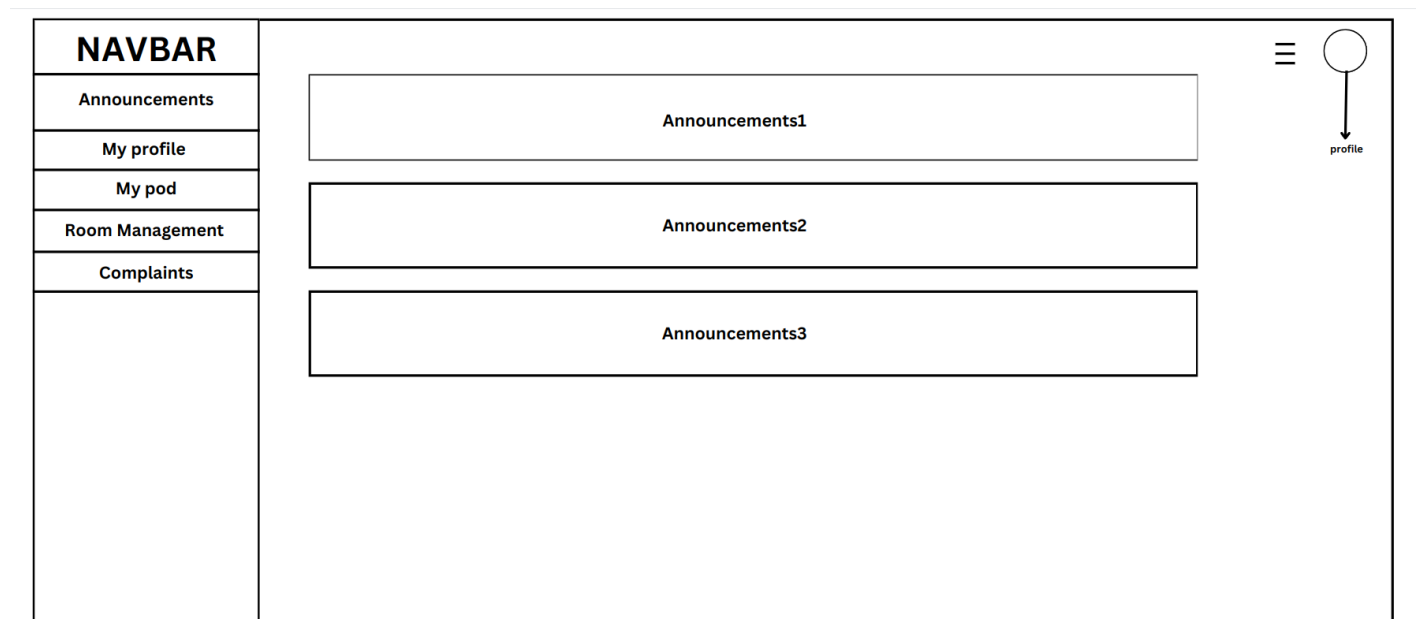
1. We can create a new role for staff, such that it would be much easy to ask for service we can by-pass HO and directly talk to the concerned staff as required.
2. We can add time constraints for solving the complaints with in given time based on type of problem.
3. We can integrate with Mess service portal/site.
4. We can integrate with accounts, to check if the hostel fee is paid.

5 Appendix:

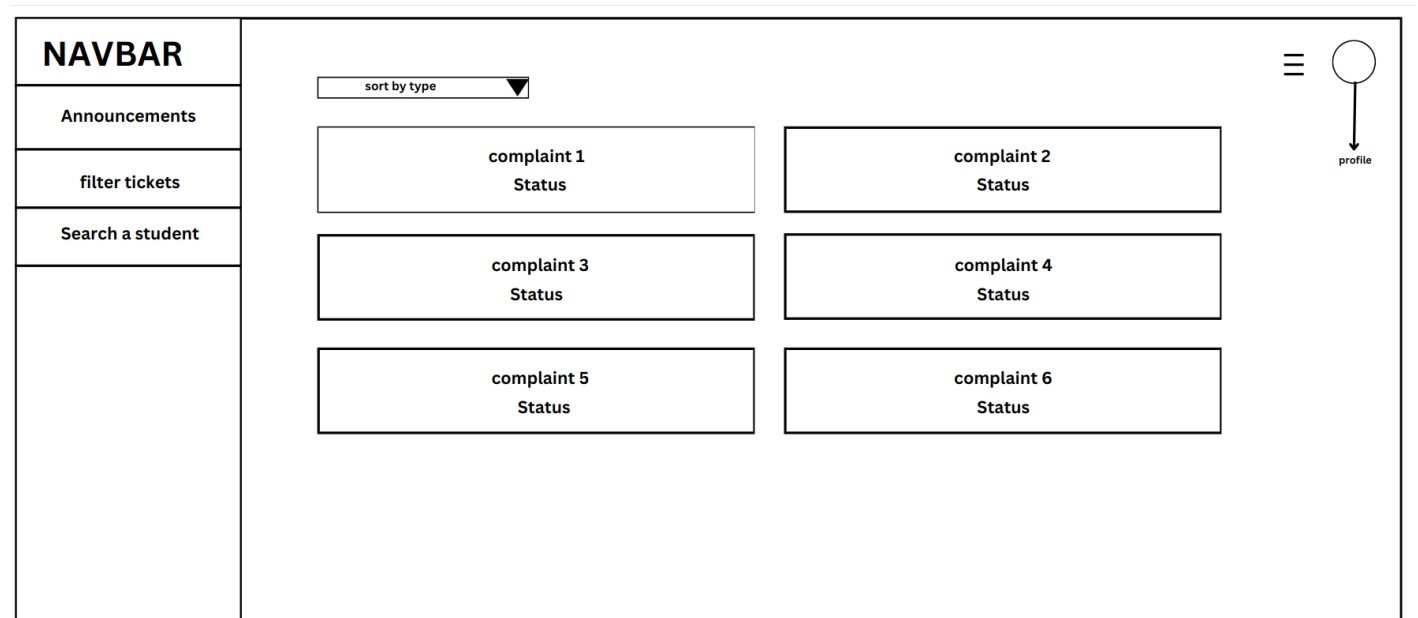
5.1 Login page of the website

The diagram illustrates the layout of a login page. It consists of a large outer rectangle representing the page area. Centered within this area is a smaller rectangle representing the login form. Inside the form rectangle, the elements are arranged vertically: a text input field with the placeholder text 'user id', another text input field with the placeholder text 'password', a button labeled 'login', a text link labeled 'forgot password?', and a text link labeled 'signup'.

5.2 Home page for student interface



5.3 Home page for HR interface



5.4 Home page for HO interface

