--

ROLL NUMBER	7376221CD157
STUDENT NAME	VIBEESH ST
SEAT NUMBER	216
PROJECT IB	16
PROJECT TITLE	HOSTEL MANAGEMENT SYSTEM

FRONTEND	React JS
BACKEND	Spring Boot Java
DATABASE	MySQL
API	Restful services

1. PURPOSE:

The Hostel Management System is aimed at automating and streamlining the hostel management process at our college. This system will provide a centralized platform for hostel managers, administrators, and students to efficiently manage hostel-related activities, such as room allocation, vacancy status tracking, and room booking.

By replacing manual processes with automated solutions, it aims to improve efficiency, reduce errors, and enhance the overall hostel experience for both students and staff.

2. SCOPE:

- Registration and management of hostels, including hostel block details and building area.
- Management of room details, including room types, capacity, and area.
- Allocation and tracking of room occupancy status (vacant/occupied).
- Booking and management of rooms for events.
- User authentication and authorization mechanisms to ensure data security and privacy.
- Integration with college databases and systems, as required.

3.OBJECTIVES:

- To automate and streamline hostel management processes to improve efficiency.
- To provide real-time visibility into hostel occupancy status and room availability.
- To enhance the hostel experience for students by providing a user-friendly platform for room booking.
- To aid hostel management in decision-making and resource planning.

4.USERS:

1. Hostel Managers and staff:
    - Responsible for managing hostel information (capacity, room types).
    - Manage student bookings during the annual room allocation process and handle outsider booking requests.
2. Students (Yearly):
    - Book rooms during the annual shift process to secure their preferred accommodation.

5.FUNCTIONAL REQUIREMENTS:

1. Hostel Management:
    - Create, edit, and view hostel details (names, capacities, building areas).
    - Manage room types (single, double, quadruple, etc.) and capacities.
    - Track room allocation details (student/staff/warden, hostel, floor, room number).
2. Room Availability:
    - Display real-time availability of rooms in each hostel by type (single, double, etc.).
    - Allow filtering based on hostel name, room type, and floor.
3. Student Room booking:
    - Facilitate student booking of rooms during the annual shift process.
    - Integrate a user-friendly search function for available rooms.
    - Enable admins to confirm and cancel rooms booked by students.
4. Room allocation for outsiders:
    - Enable admin to book available rooms for outsiders.
5. Reporting:
    - Generate reports on total admitted students per hostel.
    - Provide room occupancy reports (occupied/vacant) for each hostel, with options for filtering by date range or other criteria.
    - Allow for custom report generation based on user-defined filters (hostel name, year, etc.).

6. NON-FUNCTIONAL REQUIREMENTS

- Performance: The system should have fast loading times and responsive user interactions.
- Security: User authentication (login) with role-based access control, data encryption, and secure data storage are essential.
- Usability: The user interface should be intuitive, user-friendly, and accessible for all user groups. Users with disabilities should be able to navigate the system effectively.

7. USER STORIES:

Hostel administrators:

- **Effortless Data Access:** Retrieve and manage hostel information instantly and efficiently.

Student**:**

- **Quick Room Selection:** Find and book my desired room during the shift without delays.

8. USER INTERFACE DESIGN:

- Develop user-friendly interfaces for hostel managers, administrators, and students.
- Ensure intuitive navigation and easy access to key functionalities.

9. DATA REQUIREMENTS:

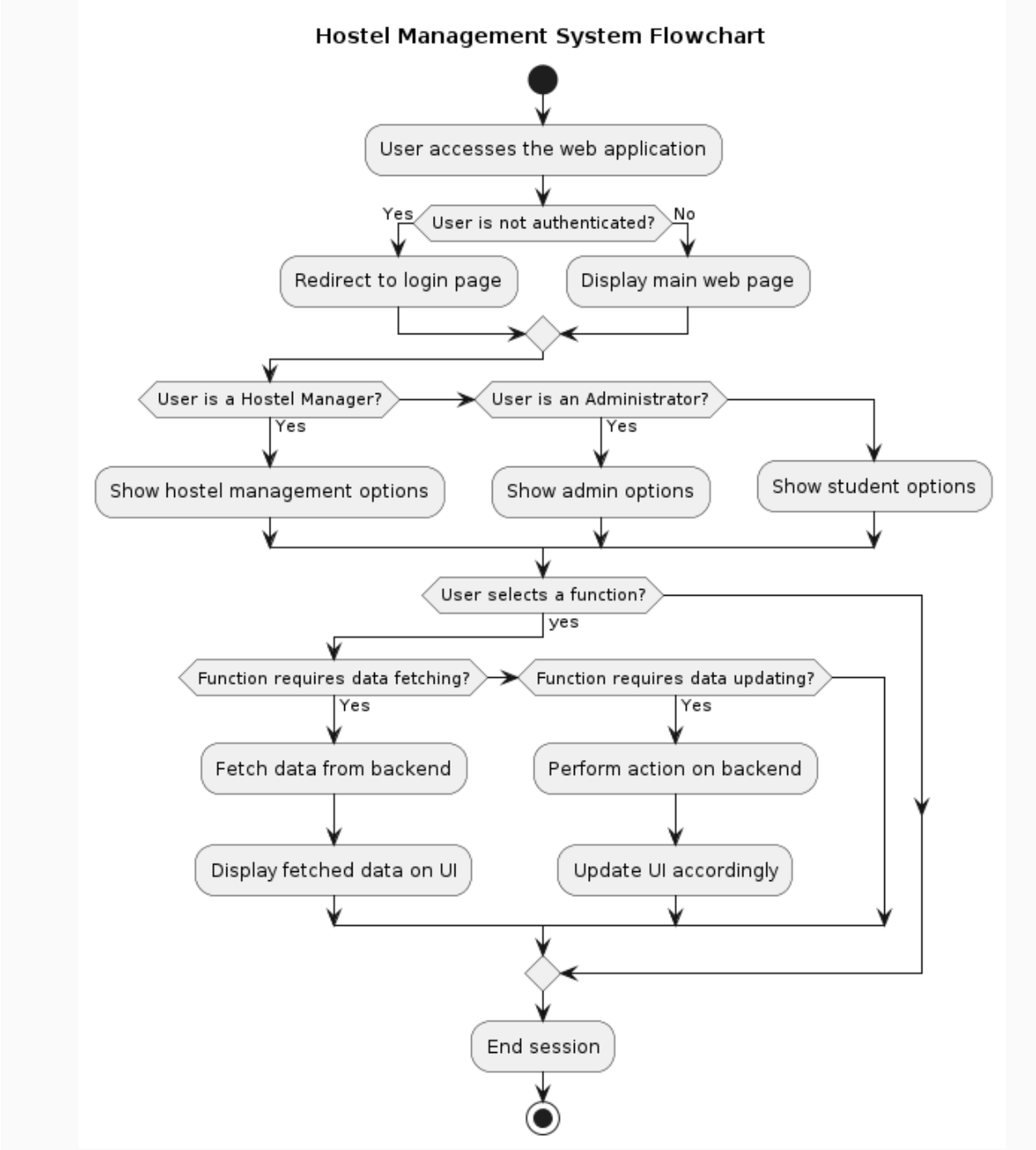
- Store hostel, room, occupancy, booking, and user data in the database.
- Define clear data entities and relationships in the database schema.
- Ensure data integrity and consistency through proper validation and constraints.

10. RISKS AND MITIGATION STRATEGIES:

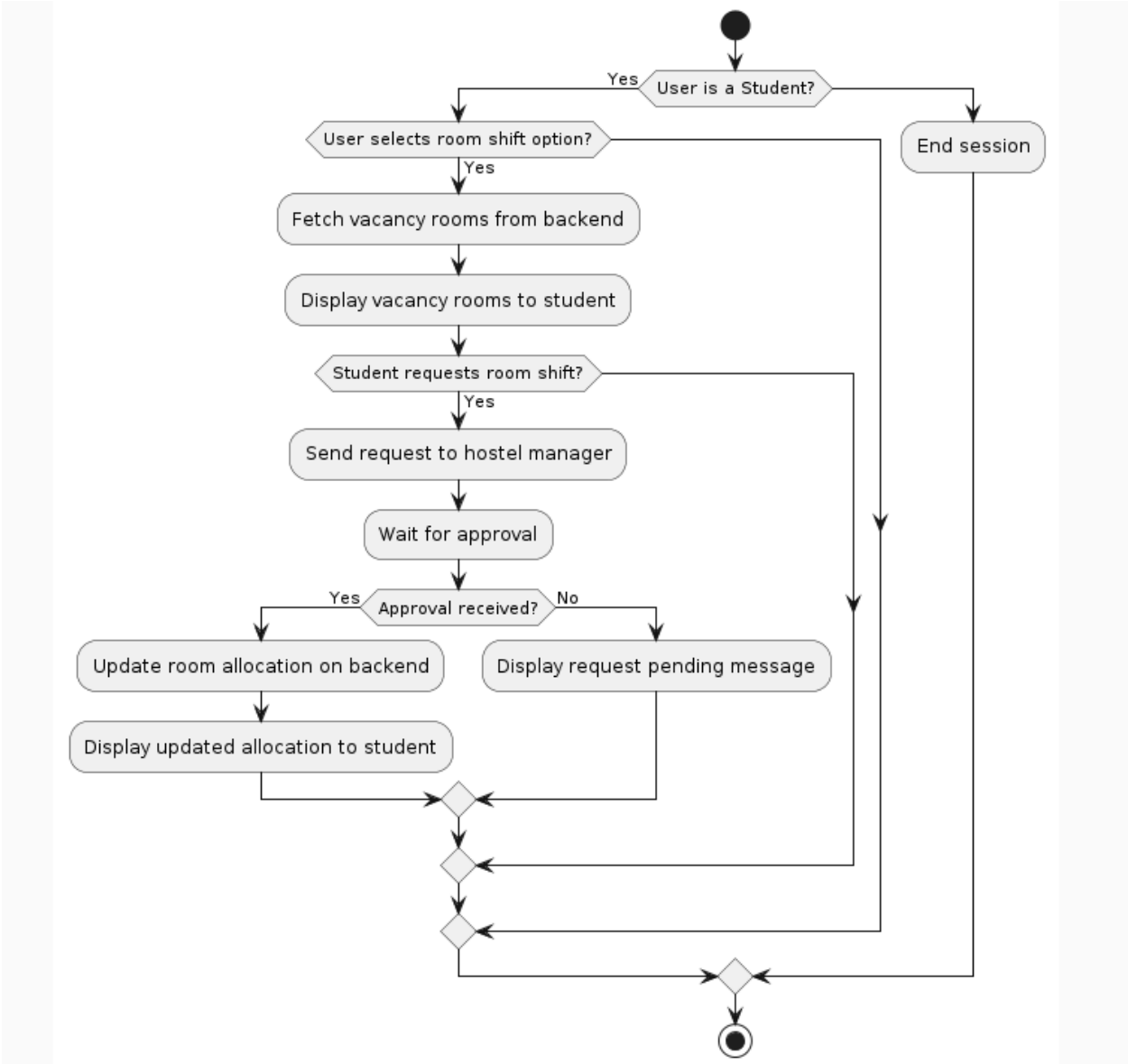
- Address potential user resistance through training and support.
- Tackle technical challenges through thorough testing and debugging.
- Maintain open communication channels to address any emerging risks promptly.

11)WORKFLOW:

- General

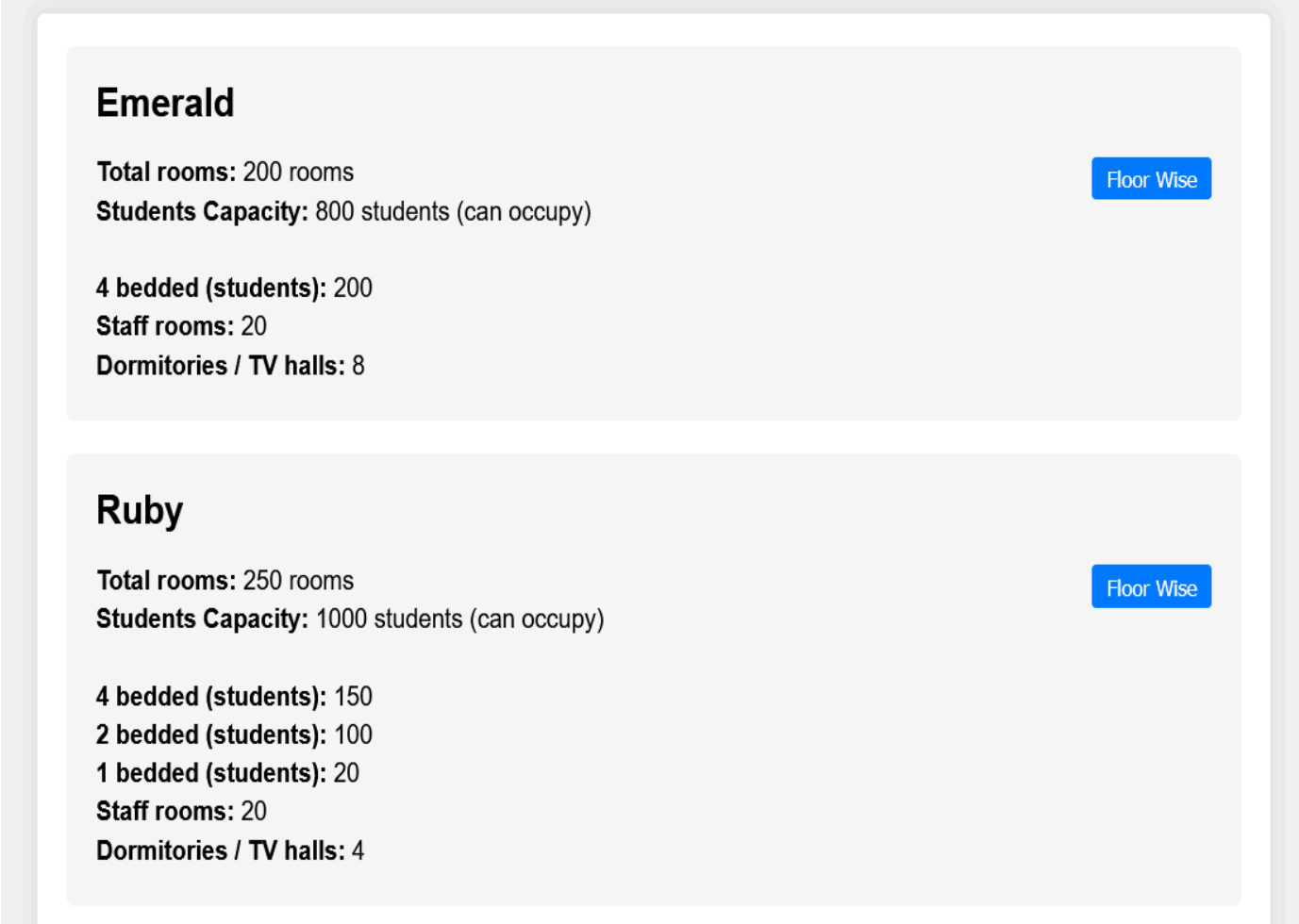


- Students booking rooms:



12) USER INTERFACE (MODEL):

Simple model page to show hostel details and room booking feature:



Ruby

Total rooms: 200 rooms
Students Capacity: 800 students (can occupy)

Floor Wise

4 bedded (students): 101
2 bedded (students): 73
1 bedded (students): 26
Staff rooms: 20
Dormitories / TV halls: 8

Floor	Total rooms	4-Bedded	2-Bedded	1-Bedded
Ground	50	20	25	5
First	50	30	15	5
Second	50	24	20	6
Third	50	27	13	10

Room Vacancy - Cauvery

Select Floor: All Select Room Type: All

Floor	Room Number	Room Type	Occupied	Vacant Spaces	Allocate
	101	Single	1	0	Allocate
1	102	Double	1	1	Allocate
	103	Four Bedded	3	1	Allocate
	201	Single	1	0	Allocate
2	202	Double	1	1	Allocate
	203	Four Bedded	4	0	Allocate