BANNARI AMMAN INSTITUTE OF TECHNOLOGY
An Autonomous Institution Affiliated to Anna University - Chennai, Accredited by NAAC with A+ Grade
Sathyamangalam - 638401 Erode District, Tamil Nadu, India
Stay Ahead

SOFTWARE REQUIREMENT SPECIFICATION

NAME	KAVIRAJ K R
ROLL. NO	7376222AL157
SEAT. NO	165
PROJECT ID	5
PROJECT NAME	RESOURCE BOOKING SYSTEM

1. PROBLEM STATEMENT:

Implement a resource booking system for reserving facilities such as classrooms, labs, auditoriums, and seminar hall. Allow users to check availability, submit booking requests, receive confirmation notifications, and manage bookings.

2. INTRODUCTION:

Project Overview:

The Resource Booking System is a web-based application designed to streamline the process of reserving facilities such as classrooms, labs, auditoriums, and seminar halls. The system provides an easy-to-use platform for users to check availability, submit booking requests, and manage their bookings efficiently.

Purpose of the Project:

The purpose of this project is to develop a centralized system that automates and simplifies the process of booking facilities within an organization or institution. By implementing this system, we aim to eliminate the need for manual booking processes, reduce conflicts in scheduling, and improve overall efficiency in managing resources.

Scope of the Project:

- Creating a user-friendly interface for users to interact with the system.
- Implementing a booking management system that allows users to check availability, submit booking requests, and receive confirmation notifications.
- Providing administrators with tools to manage facilities, approve or reject booking requests, and generate reports.
- Ensuring the system is scalable, secure, and easy to maintain.

Goals and Objectives:

The primary goals of the Resource Booking System are to:

- Improve the efficiency of booking facilities by providing a centralized and automated system.
- Reduce conflicts in scheduling by allowing users to view real-time availability of facilities.
- Enhance user experience by providing a seamless and intuitive interface for booking and managing facilities.

The objectives of the project are to:

- Develop a robust and reliable system that meets the needs of users and administrators.
- Ensure the system is scalable to accommodate future growth and changes in requirements.
- Implement security measures to protect user data and prevent unauthorized access.

3. PROJECT TIMELINE:

PHASE	DURATION	DESCRIPTION
1	Week 1 - 2	• Define project scope, objectives, and requirements. • Develop a project plan and timeline.
2	Week 3 - 4	• Design and UI/UX prototyping • Frontend Development
3	Week 5 - 6	• Database design and implementation • Integrate the database using MySQL for data storage.
4	Week 6 - 8	• Implement the backend logic using Apache and Laravel framework. • Backend development
5	Week 8 - 9	• Integration and testing
6	Week 9 – 10	• Deployment

4. SYSTEM OVERVIEW:

System Architecture:

The Resource Booking System is designed using a client-server architecture. The client-side is a web application accessible through a browser, while the server-side consists of a backend server that handles the business logic and interacts with the database.

The system follows a three-tier architecture, with the presentation layer (client-side), application layer (backend server), and data layer (database) separated for scalability and maintainability.

Components:

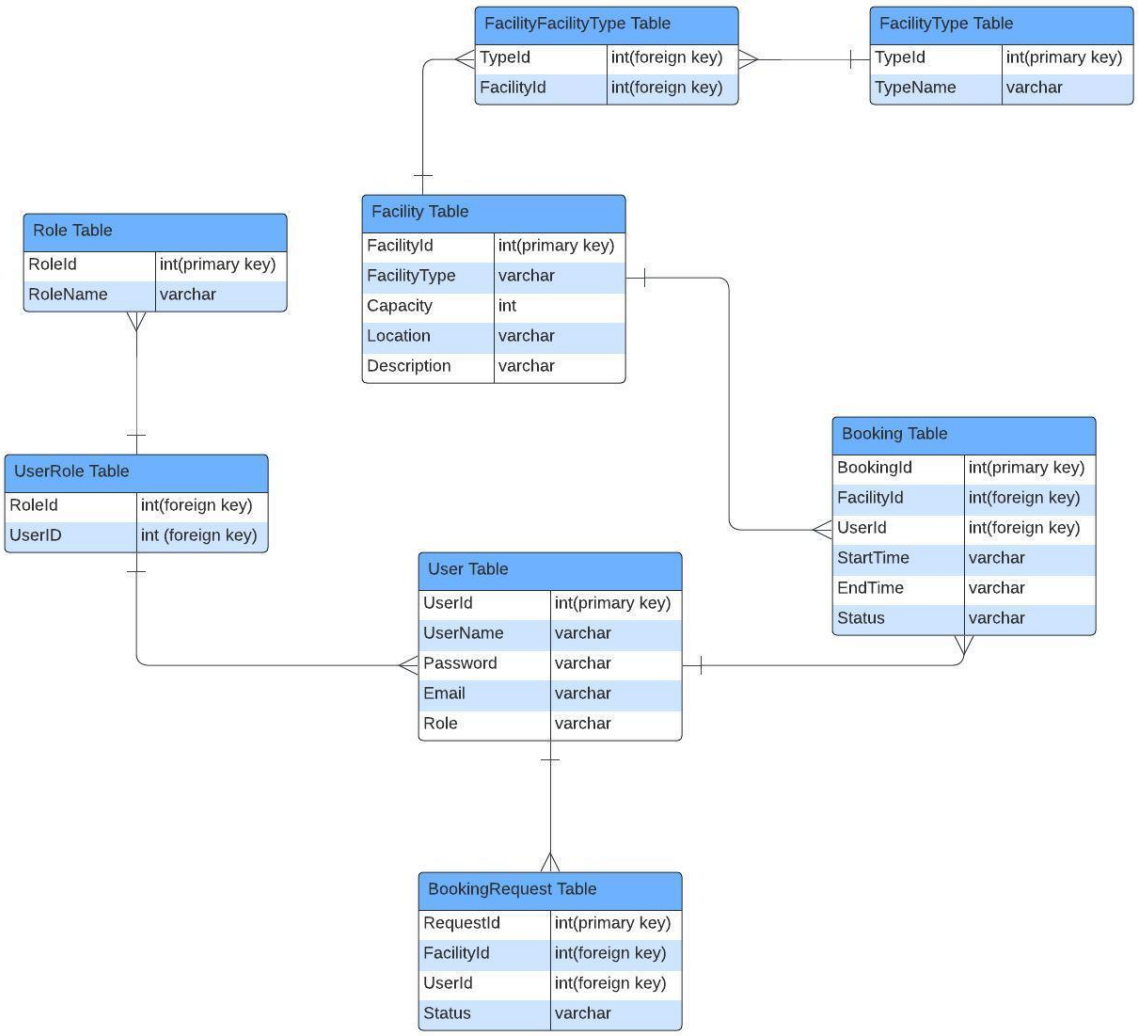
1. **Frontend (Client-side):** The frontend of the system is developed using HTML, CSS, and JavaScript. It provides the user interface for users to interact with the system, including viewing available facilities, submitting booking requests, and managing bookings.
2. **Backend (Server-side):** The backend of the system is developed using Apache and PHP with Laravel framework. It handles the business logic of the application, including processing booking requests, managing user authentication, and interacting with the database.
3. **Database:** The database used in the system is MySQL. It stores information about facilities, bookings, users, and other relevant data. The database design is normalized to reduce redundancy and improve data integrity.

Technological Stack:

Frontend	HTML, CSS, JAVASCRIPT
Backend	PHP WITH LARAVEL FRAMEWORK, APACHE
Database	MYSQL
API	RESTFUL API

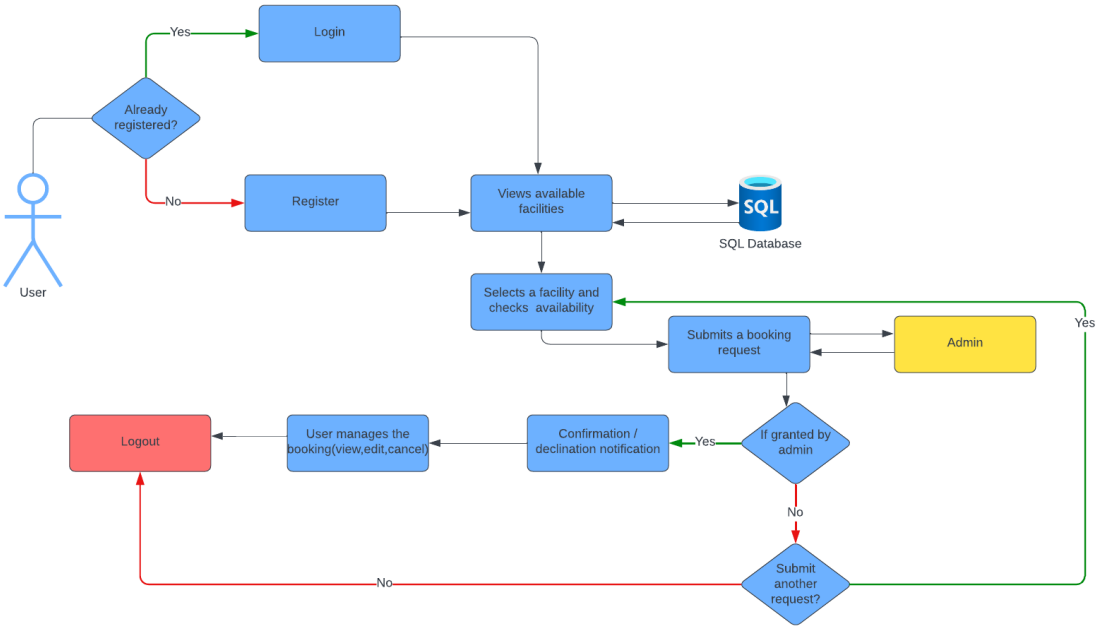
Database Design:

ER diagram:

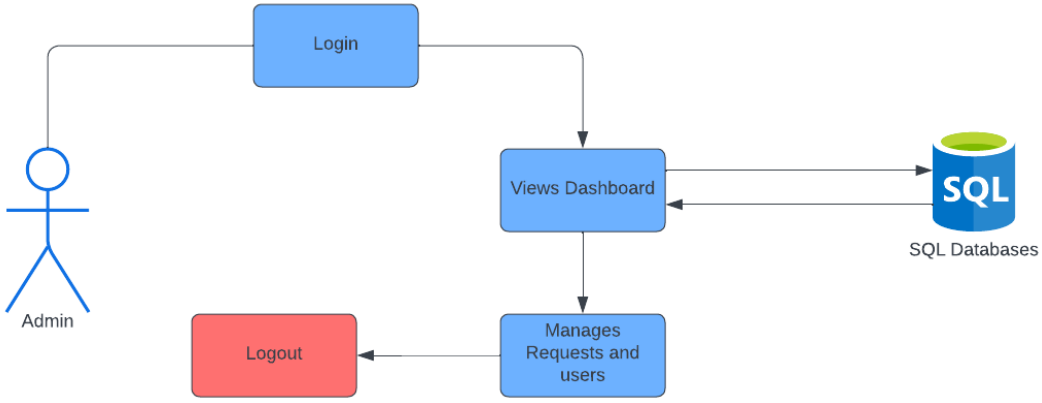


Diagrams:

User's Interface:



Admin's Interface:



5. FUNCTIONAL REQUIREMENTS:

1. **User Registration and Authentication:**
    - Users should be able to register for an account and authenticate themselves to access the system.
2. **Facility Management:**
    - Admins should be able to add, edit, and delete facility details such as name, capacity, and location.
3. **Booking Management:**
    - Users should be able to view available facilities and submit booking requests.
    - Admins should be able to view, approve, or reject booking requests.
4. **Notification System:**
    - Users and admins should receive notifications for booking requests, approvals, and rejections.

6. NON – FUNCTIONAL REQUIREMENTS:

1. **Usability:**
    - The user interface should be intuitive and easy to use for both users and admins.
2. **Performance:**
    - The system should be able to handle a large number of concurrent users and booking requests without significant performance degradation.
3. **Security:**
    - The system should implement secure authentication mechanisms to protect user accounts and data.
    - Data should be stored securely and access should be restricted to authorized users only.

Use Cases:

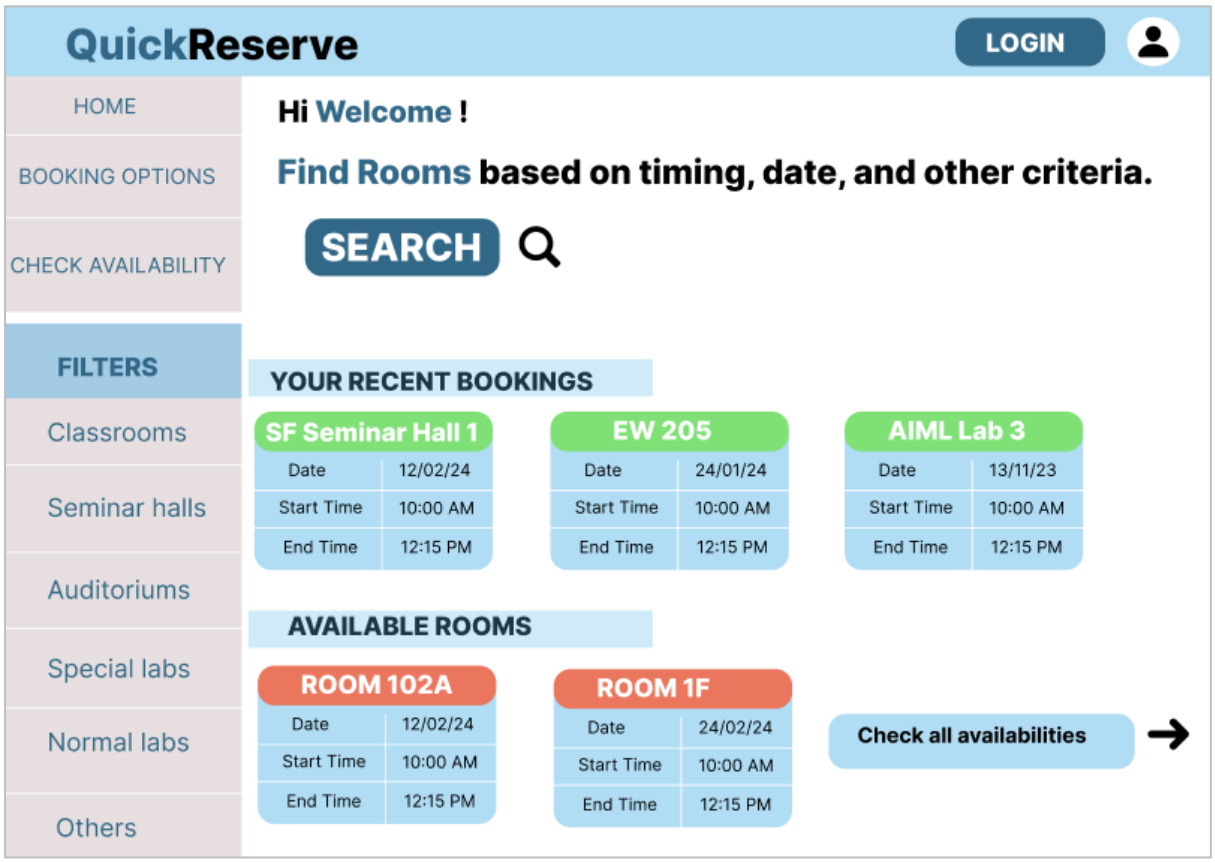
1. **Facility Booking:**
    - **Actor:** User
    - **Description:** User wants to book a facility for a specific date and time.
    - **Steps:**
        1. User logs in to their account.

2. User navigates to the list of available facilities.
3. User selects a facility and specifies the booking start and end time.
4. User submits the booking request.
5. System checks the facility's availability and sends a notification to the user.

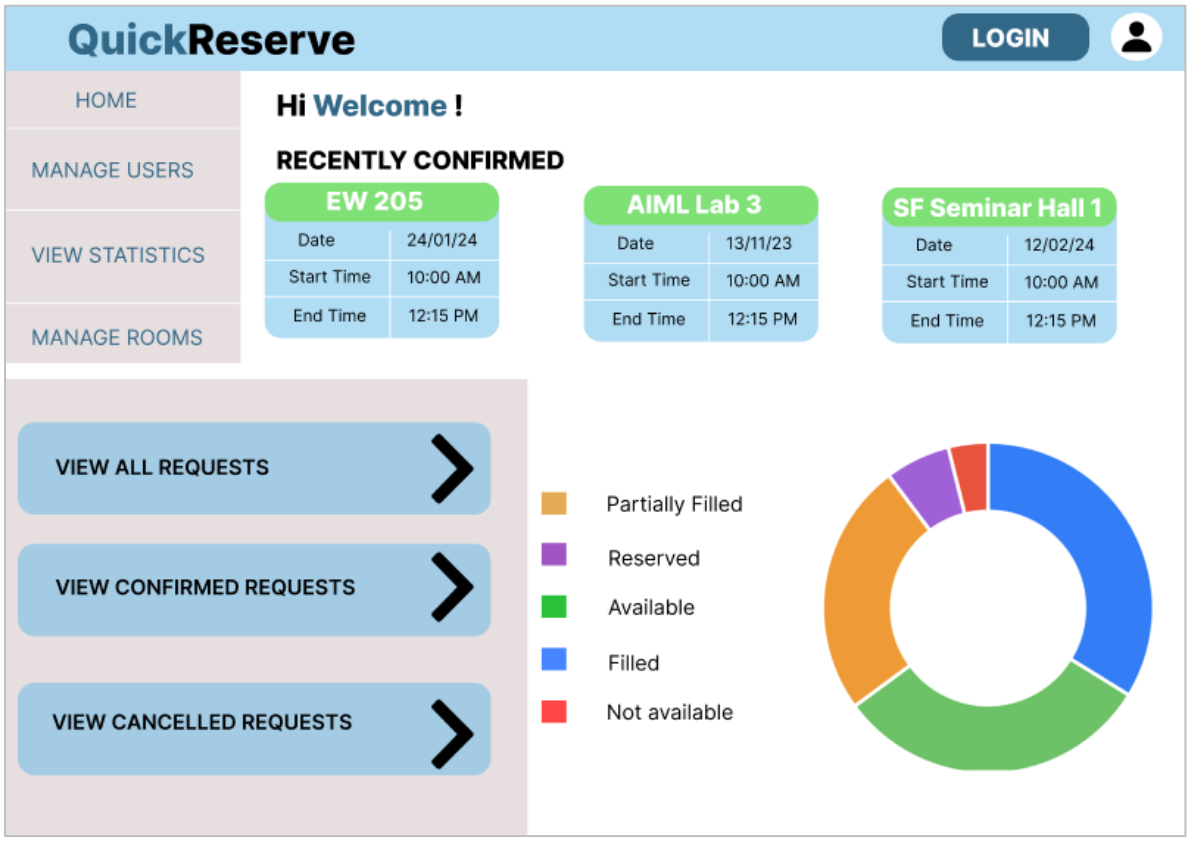
2. **Booking Approval:**
   - **Actor:** Admin
   - **Description:** Admin wants to approve or reject a booking request.
   - **Steps:**
     1. Admin logs in to their account.
     2. Admin navigates to the list of pending booking requests.
     3. Admin selects a booking request and reviews the details.
     4. Admin approves or rejects the booking request.
     5. System sends a notification to the user with the booking status.

7. PROTOTYPE:

User's View:



Admin's View:



Login Page: