

Software Requirements Specification Document (SRS)

Lab Space and Desktop Allocation for
Research Labs & Server Compute Usage
Management

AN_07

Advaith Girish	Adithyan D	Y Venkateshwara Rao	Prithvi Ganapathi Bhat
----------------	------------	------------------------	---------------------------

Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms, and Abbreviations
- 1.4 Overview

2. Functional Requirements

- 2.1 User Roles & Permissions
- 2.2 F1 – Lab Layout Management
- 2.3 F2 – Server Usage Monitoring
- 2.4 F3 – User Management
- 2.5 F4 – Notification System
- 2.6 F5 – Allocation Tracking
- 2.7 F6 – Real-Time Alerts
- 2.8 F7 – Reporting
- 2.9 F8 – System Configuration

3. Non-Functional Requirements

- 3.1 Performance Requirements
- 3.2 Security Requirements
- 3.3 Usability Requirements

4. System Design

- 4.1 Technologies to be Used
- 4.2 Database Design

5. Assumptions & Constraints

6. Future Enhancements

7. Use Case Diagram

8. Class Diagram

9. Version History



1. Introduction

1.1 Purpose

This document defines the requirements for the **Lab Space & Desktop Allocation Management System**, a web-based solution designed to optimize lab resource allocation, monitor server compute usage, and provide real-time alerts for efficient resource management in academic research environments.

1.2 Scope

The system will:

- Enable **Admins** to upload lab layouts, define seating positions, and set server usage thresholds.
- Allow **Faculty** to allocate/deallocate seats and monitor server metrics.
- Permit **Students** to request seats and view usage statistics.
- Generate real-time alerts for server overuse and periodic resource utilization reports.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
Admin	Superuser with privileges to manage lab layouts, users, and server thresholds.
Faculty	Authorized user to allocate seats and track resource usage.
RBAC	Role-Based Access Control: Restricts system access based on user roles.
SRS	Software Requirements Specification.

1.4 Overview

This SRS outlines the functional and non-functional requirements, system architecture, and operational guidelines for the system.

2. Functional Requirements

2.1 User Roles & Permissions

Role	Permissions
Admin	Upload lab layouts, define seating, set server thresholds, manage users.
Faculty	Allocate/deallocate seats, view allocations, monitor server usage.
Student	Request seats, view allocated seat details, check server metrics.

3. Functional Requirements

F1:Admin Functional Requirements

- **F1.1 Upload Lab Layouts:** Admins can upload scanned images of lab layouts.
- **F1.2 Define Seating Positions:** Admins can mark and label seating positions on the uploaded layouts.
- **F1.3 Manage Allocations:** Admins can assign seating to students and faculty with statuses such as Available, Allocated, or Blocked.
- **F1.4 Monitor Resource Usage:** Admins can track CPU and memory usage in real-time for desktop-based labs.
- **F1.5 Set Usage Thresholds:** Admins can define acceptable usage limits for server resources.
- **F1.6 Generate Alerts:** Admins receive notifications when usage exceeds predefined thresholds.
- **F1.7 Manage User Roles:** Admins can assign roles (Admin, Faculty, Student) with specific permissions.
- **F1.8 Manage Authentication and Authorization:** Admins can implement secure login mechanisms for users.

- **F1.9 Receive System Alerts:** Admins get real-time notifications for system issues and exceed resource usage.

F2:Faculty Functional Requirements

- **F2.1 View Lab Layouts:** Faculty can access uploaded lab layouts and seating arrangements.
- **F2.2 Request Seating Modifications:** Faculty can request changes in seating allocations if necessary.
- **F2.3 Monitor Assigned Resources:** Faculty can view resource usage for assigned systems.
- **F2.4 Receive Notifications:** Faculty get alerts related to resource usage, lab status, or system issues.

F3:Student Functional Requirements

- **F3.1 View Assigned Seat:** Students can check allocated seating positions in the lab layout.
 - **F3.2 Monitor Personal System Usage:** Students can view real-time CPU and memory usage of the assigned system.
 - **F3.3 Receive Personal Alerts:** Students get notifications if assigned system usage exceeds set thresholds.
-

Use Case 1: Lab Space Allocation

Description: The system automatically assigns lab spaces based on availability and predefined rules.

Steps:

1. The user logs into the system.
 2. The user selects the desired lab and time slot.
 3. The system checks availability.
 4. If available, the system assigns the lab space and confirms the booking.
 5. If unavailable, the system suggests alternative slots.
 6. The user receives a confirmation message.
-

Use Case 2: Lab Layout Management

Description: Manage lab layouts and seating allocations.

Steps:

1. The administrator logs into the system.
 2. The administrator uploads a scanned lab layout (JPEG/PNG).
 3. The system processes and displays the layout.
 4. The administrator defines and labels seating positions.
 5. The administrator assigns seat statuses as Available, Allocated, or Blocked.
-

Use Case 3: Administrator Override Allocation

Description: Administrators can manually override automatic lab allocations.

Steps:

1. The administrator logs into the system.
2. The administrator selects the lab and time slot.

3. The system displays the current allocation status.
 4. The administrator overrides the existing allocation.
 5. The system updates the records and notifies affected users.
-

Use Case 4: Server Usage Monitoring

Description: Monitor real-time server compute resources.

Steps:

1. The administrator accesses the server monitoring dashboard.
 2. The system displays CPU and memory usage for desktop-based labs.
 3. The administrator sets usage thresholds (e.g., 90% CPU).
 4. The system continuously tracks resource consumption.
 5. If a threshold is exceeded, the system triggers an alert.
-

Use Case 5: User Cancellation of Lab Booking

Description: Users can cancel their lab reservations if needed.

Steps:

1. The user logs into the system.
 2. The user navigates to their bookings.
 3. The user selects the booking to cancel.
 4. The system asks for confirmation.
 5. Upon confirmation, the system releases the lab space.
 6. The user receives a cancellation confirmation.
-

Use Case 6: Viewing Lab Availability

Description: Users can check lab availability before making a booking.

Steps:

1. The user logs into the system.

2. The user selects the lab and date.
 3. The system fetches and displays available time slots.
-

Use Case 7: Lab Usage Reporting

Description: The system generates reports on lab space utilization.

Steps:

1. The administrator logs into the system.
 2. The administrator selects the reporting module.
 3. The system compiles lab usage data.
 4. The report is generated and displayed/downloaded.
-

Use Case 8: User Management

Description: Manage user roles and authentication.

Steps:

1. The user accesses the login page.
 2. The user enters credentials (LDAP/NITC-based authentication).
 3. The system verifies login credentials.
 4. If valid, access is granted based on role (Admin, Faculty, Student).
 5. The user can reset passwords or enable multi-factor authentication if required.
-

Use Case 9: Role-Based Access Control (RBAC)

Description: The system restricts functionalities based on user roles.

Steps:

1. The user logs in.
2. The system verifies the user's role.
3. Based on the role, the system grants access to permitted features.

Use Case 10: Notification and Alerts

Description: Deliver system alerts and reports to users/admins.

Steps:

1. A critical event occurs (e.g., booking confirmation, server overload).
 2. The system generates a notification.
 3. The notification is sent via email/SMS to relevant users.
-

Use Case 11: Allocation Tracking

Description: Track and display seat allocation history.

Steps:

1. The administrator logs into the system.
 2. The system displays real-time seat occupancy per lab.
 3. The administrator accesses historical allocation records.
 4. The system allows downloading archived allocation data.
-

Use Case 12: Real-Time Alerts

Description: Notify users of critical events immediately.

Steps:

1. The system continuously monitors server and seat allocations.
 2. If server usage exceeds a threshold, an alert is triggered.
 3. If a seat allocation is changed, the affected user is notified.
-

Use Case 13: Reporting and Analytics

Description: Generate analytical reports on lab allocations and server usage.

Steps:

1. The administrator logs into the reporting module.
 2. The administrator selects the report type (seat allocation/server usage).
 3. The system compiles and displays data.
 4. Reports can be exported in PDF/CSV formats.
 5. Graphs visualize usage trends.
-

Use Case 14: System Configuration

Description: Configure system parameters and alerts.

Steps:

1. The administrator accesses system settings.
 2. The administrator modifies server monitoring intervals (e.g., 5s, 10s).
 3. The administrator customizes alert templates and recipient lists.
 4. Changes are saved and applied system-wide.
-

3. Non-Functional Requirements

3.1 Performance Requirements

- Support **500+ concurrent users** with response times under **2 seconds**.
- Server usage data updated every **5 seconds**.

3.2 Security Requirements

- Encrypt user data and lab layouts using **AES-256**.
- **JWT authentication** for API endpoints.
- **RBAC** to restrict access based on roles.

3.3 Usability Requirements

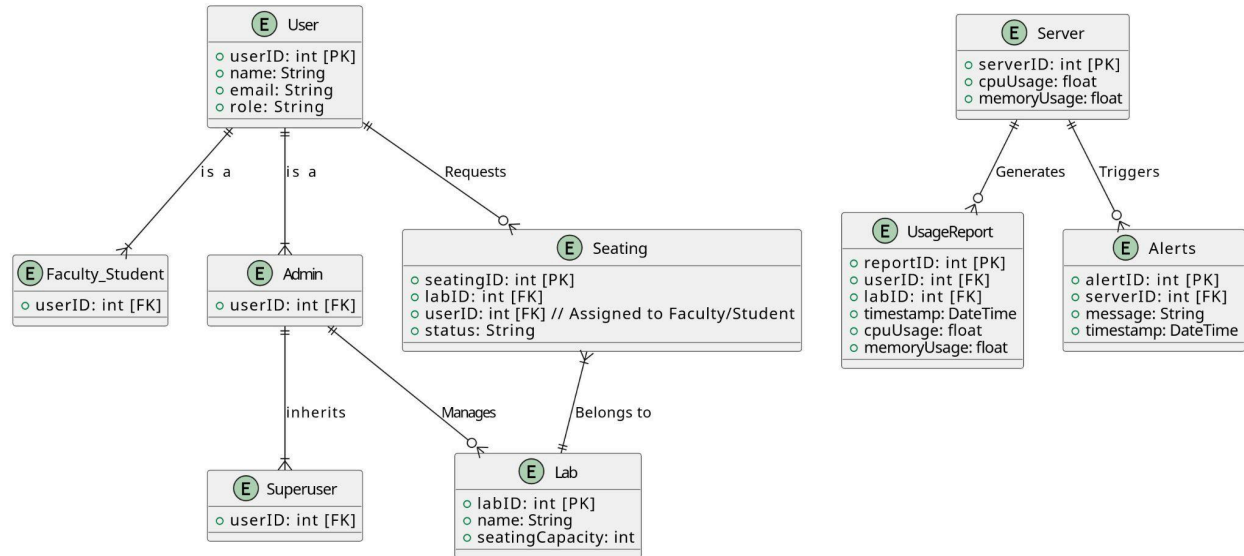
- Intuitive drag-and-drop interface for lab layout upload.
- Mobile-responsive dashboard for faculty and students.

4. System Design

4.1 Technologies to be Used

- **Frontend:** React.js (Material-UI for layouts).
- **Backend:** Java Spring(Spring Boot, REST APIs).
- **Database:** MongoDB (store layouts, allocations, server logs).

4.2 Database Design



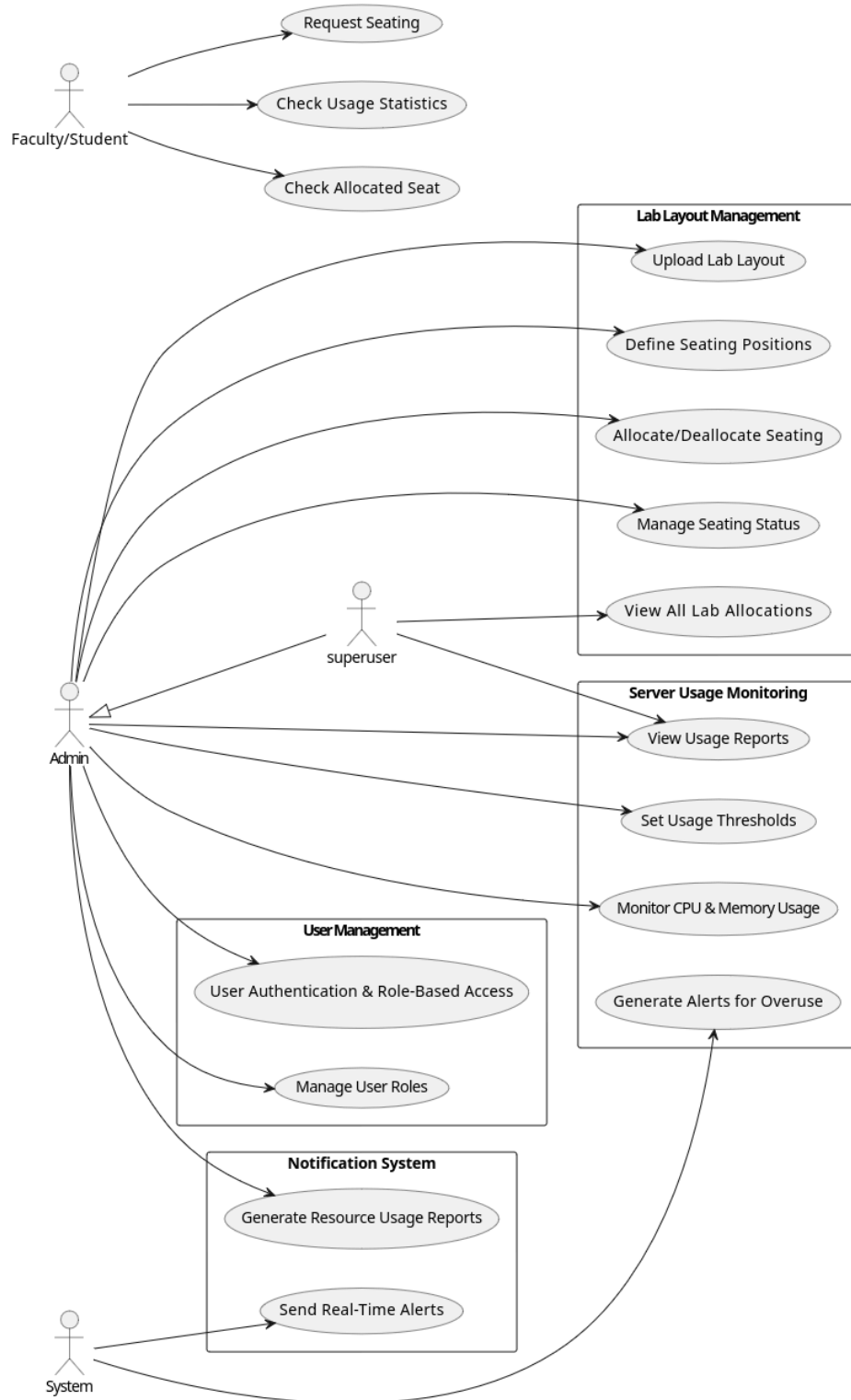
Entities:

1. **Users** (*userId*, *email*, *role*, *encryptedPassword*)
 2. **Labs** (*labId*, *layoutImage*, *seatingCapacity*)
 3. **Seats** (*seatId*, *labId*, *status*, *allocatedTo*)
 4. **ServerUsage** (*timestamp*, *labId*, *cpuUsage*, *memoryUsage*)
 5. **Alerts** (*alertId*, *labId*, *message*, *severity*)
-

5. Assumptions & Constraints

- Only admins can modify lab layouts and server thresholds.
- Students cannot request multiple seats simultaneously.
- Server thresholds are predefined and non-editable by faculty/students.

7. Use Case Diagram

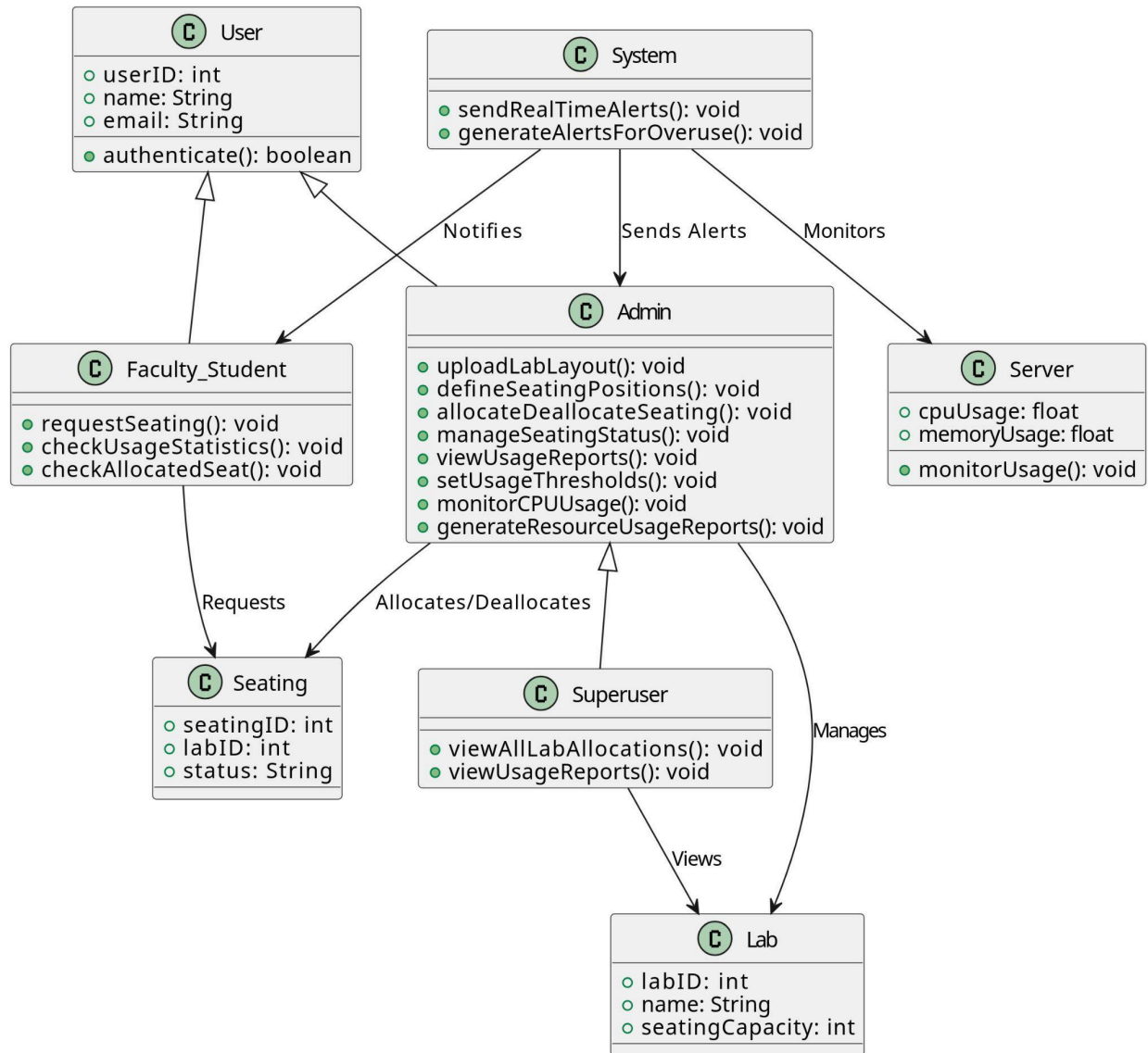


Actors: Admin, Faculty, Student

Key Use Cases:

- Admin: *Upload Layout, Set Server Thresholds, Manage Users.*
 - Faculty: *Allocate Seats, View Allocation Reports.*
 - Student: *Request Seat, Check Usage Statistics.*
-

8. Class Diagram



9. Version History

Version	Date	Author	Changes
1.0	10-Dec-2024	Advaith Girish	Initial draft.
1.1	12-Feb-2024	Adithyan D	Added use case diagrams, refined NFRs.
1.2	24-Feb-2025	Prithvi Ganapathi	Refined the SRS, added Database design, and Class diagram.
