Software Laboratory Management System

Presented by :

1. SHIVAM RAI [2022CSB011]
2. ANKITA TRIPATHI [2022CSB012]
3. ANGSHUMAN ROY [2022CSB014]
4. SAYANDIP ROY [2022CSB015]
5. ANUSREE MANDAL [2022CSB016]

Problem Understanding:

The **Software Laboratory Management System** facilitates **students, lab-in-charge, faculty members, and lab assistants** in managing lab resources, scheduling, issue reporting, and monitoring.

🔍 Introduction

- Effective management of software laboratories is critical for smooth academic operations.
- Challenges such as resource conflicts, manual scheduling, lack of transparency, and poor communication hinder productivity.
- Our system proposes a **centralized, user-friendly platform** to streamline operations across all roles:
  👩‍🎓 Students, 👨‍🏫 Faculty Members, 👩‍🔧 Lab Assistants, and 👨‍💼 Lab-in-Charge.

Use-Case Diagram

A **Use-Case Diagram** represents the interactions between **users (actors)** and the **system functions**.

Actors:

- **Student** 👩‍🎓 (Requests lab access, reports issues)
- **Faculty Member** 👨‍🏫 (Reserves labs, monitors student activities)
- **Lab Assistant** 🛠️ (Maintains equipment, assists users)
- **Lab In-Charge** 👨‍💼 (Manages schedules, approvals, and reports)

Use Cases:

- **Lab Reservation:** Faculty reserves labs; lab-in-charge approves.
- **Resource Allocation:** Students request software/hardware; lab-in-charge processes requests.
- **Issue Reporting:** Students and faculty report issues; lab assistants fix them.
- **Monitoring & Logs:** Lab-in-charge tracks usage logs and generates reports.

Use Cases and Interactions

1. Students

- **Book Lab Slot** (Requests lab access)
- **View Software Availability** (Checks available software in the lab)
- **Report Issues** (Logs hardware/software issues)
- **Access Lab Resources** (Uses lab for assignments/projects)

2. Faculty Members

- **Schedule Lab Sessions** (Reserves lab for classes)
- **Monitor Attendance** (Marks students' presence in labs)
- **View Lab Resources** (Checks hardware/software availability)
- **Report Issues** (Logs maintenance requests)

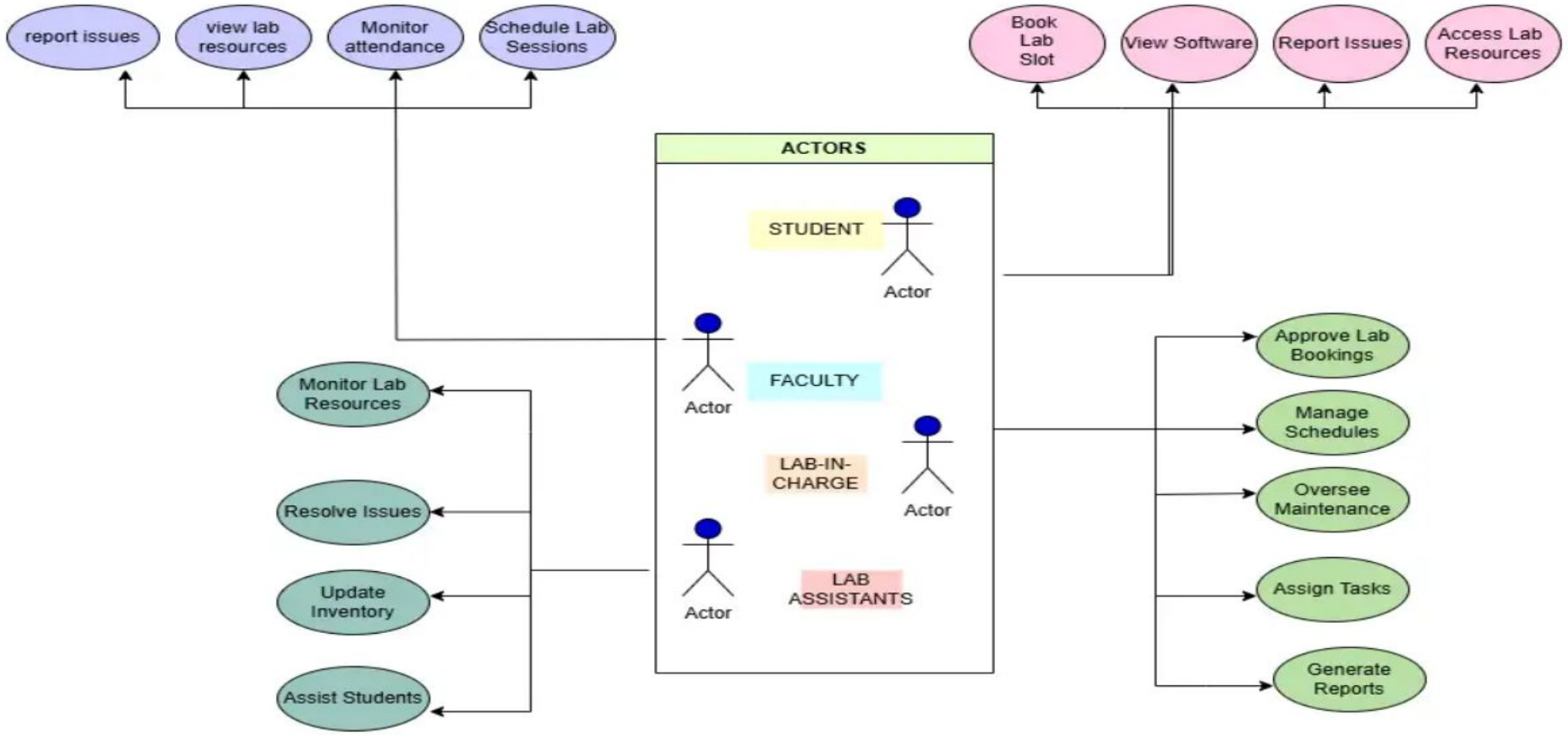
3. Lab-in-Charge

- **Approve/Reject Lab Bookings** (Manages student booking requests)
- **Manage Lab Schedules** (Oversees lab usage and schedules)
- **Oversee Maintenance Requests** (Handles reported issues)
- **Assign Tasks to Lab Assistants** (Delegates maintenance tasks)
- **Generate Reports** (Analyzes lab usage and issues)

4. Lab Assistants

- **Monitor and Maintain Lab Resources** (Ensures lab equipment/software is functional)
- **Address Reported Issues** (Resolves hardware/software problems)
- **Update Software Inventory** (Keeps track of installed software)
- **Assist Students** (Provides technical help)

Use-Case Diagram Representation



Functional Requirements

- **User Management**

✅ Secure login system for students, faculty, lab assistants, and in-charge.
✅ Profile management (update password, user details).

- **Lab Scheduling & Reservations**

✅ Faculty members can **reserve labs** for specific time slots.
✅ Lab-in-charge can **approve or reject** reservations.
✅ Students can **check lab availability** online.

- **Equipment & Resource Management**

✅ Lab assistants can **track hardware/software** availability.
✅ Students & faculty can **request new software/hardware**.
✅ Lab-in-charge can **approve/disapprove** resource requests.

Functional Requirements

🔹 Issue Reporting & Maintenance

✅ Students & faculty can **report issues** (e.g., system failure, software crash).
✅ Lab assistants get **notified of reported issues**.
✅ Lab-in-charge **assigns tasks** for issue resolution.

🔹 Monitoring & Reports

✅ Lab-in-charge can **view logs** (who accessed the lab, duration, activities).
✅ System generates **resource usage reports** (e.g., which software is used the most).
✅ Reports on **issue resolution history** for tracking efficiency.

Non-Functional Requirements

These ensure the **quality** of the system.

- **Performance Requirements**

✅ System should support **at least 100 concurrent users**.
✅ Lab reservation requests should be **processed within 2 seconds**.

- **Security Requirements**

✅ Role-based access: Students cannot modify lab schedules.
✅ Passwords must be **encrypted** and stored securely.

Non-Functional Requirements

🔹 Usability Requirements

✅ User-friendly dashboard for easy **lab booking & issue reporting**.
✅ Accessible via **desktop and mobile devices**.

🔹 Reliability Requirements

✅ System must be **available 99.9% of the time**.
✅ Automatic **backup** of data every **24 hours**.