LAB MANAGEMENT SYSTEM

GROUP: 02

TEAM LEAD: PAKEEZA NASIR

MEMBER 1: ABDUL AHAD

MEMBER 2: SIBGHA AQEEL

ASSIGNMENT: 05

Software Requirements Specification (SRS) for Laboratory Management System

Following the guidelines of ISO/IEC/IEEE 830-1998

---

1. Introduction

1.1 Purpose

The purpose is to transition from a **manual** lab management process to a **digital system** that enables better organization of lab resources, tracking of hardware and software, and efficient handling of user complaints and software installation requests.

~~This document defines the functional and non-functional requirements for the Laboratory Management System (LMS). The LMS will assist the department coordinator in managing and maintaining information about laboratory resources, including computer hardware and software specifications, complaints from users, and installation requests. This document is intended to be used by developers, project stakeholders, quality assurance teams, and end users, including laboratory coordinators, technicians, and students.~~

1.2 Document Conventions

This document follows the IEEE 830-1998 SRS standard, utilizing the following conventions:

- **Shall**: Indicates a mandatory requirement.
- **Should**: Indicates a recommendation.
- **May**: Indicates an optional feature or capability.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- **System Developers**: To understand the requirements and design the system.
- **Project Stakeholders**: To ensure that all business needs are captured and agreed upon.
- **Quality Assurance Team**: To define testing criteria.
- **End Users**: To understand the system's expected functionalities and features.

1.4 ~~Product~~ Scope

- **Maintain records** of all labs and computers.
- **Store hardware specifications** of each computer.

- **Track installed software** on every system.
- **Allow students and teachers to submit complaints** about faulty computers.
- **Enable users to request software installations** and track their fulfillment.

1.4.1 OBJECTIVES

- To ensure **efficient management of laboratory resources**.
- To provide a **user-friendly platform** for complaint registration and tracking.
- To **simplify the process of software installation requests**.
- To ensure all lab **hardware and software records remain updated**.

1.5. DEFINITIONS

- **Laboratory Management System:** A **web-based platform** that helps manage laboratory resources, including computer details, hardware specifications, and installed software.
- **Complaint Registration:** The process of **logging issues** related to faulty computers by students or teachers for resolution.
- **Software Installation Request:** A feature allowing users to **request new software installations** on lab computers and track their fulfillment.
- **Hardware Specifications:** The **technical details** of a computer, including processor, RAM, storage, and other components.
- **SRS (Software Requirement Specification):** A **detailed document** that defines the software's purpose, features, and expected behaviour within its environment.

~~The Laboratory Management System (LMS) is a web-based platform designed to:~~

- ~~Manage computer records in the laboratory (including hardware specifications and installed software).~~
- ~~Register and track user complaints regarding computer functionality.~~
- ~~Handle software/hardware installation requests from students and staff.~~
- ~~Generate reports regarding lab status, hardware details, and complaint resolutions.~~

Formatted: Font: Bold

Formatted: Indent: Left: 0 cm

Formatted: List Paragraph

Formatted: Font: Bold

1.~~6~~5 References

ISO/IEC/IEEE 16326:2019, Systems and software engineering -Life cycle processes-Project management.

Case Study: 02 – Problem description provided by the teacher.~~IEEE Std 830-1998: Software Requirements Specification Standard~~

- ~~Institutional IT Policies~~

---

2. Overall Description

2.1 Product Perspective

The LMS will be a web-based system that is accessible from any modern web browser. The system will interact with a backend database to store and manage information about laboratories, computers, complaints, and installation requests.

2.2 Product Functions

- Login
- Register complaint
- Request software installation.
- View complaint status
- Generate hardware reports
- Generate lab reports
- Manage computer details
- Manage hardware details
- Track installed software
- Approve or Reject requests
- Resolve complaints
- Store and process data

~~**User Authentication**: Login and role-based access for administrators, technicians, and students/teachers.~~

Formatted: No bullets or numbering

Formatted: List Paragraph

Formatted: Font: Not Bold

Formatted: Bulleted + Level: 1 + Aligned at: 1.27 cm + Indent at: 1.27 cm

- ~~**Manage Computer Details**: CRUD (Create, Read, Update, Delete) operations for computer records, including hardware and software specifications.~~
- ~~**Complaint Registration and Tracking**: Students and teachers can register complaints regarding lab computers, with status tracking for resolution.~~
- ~~**Installation Request Management**: Students and teachers can submit requests for software or hardware installations, which administrators can approve or reject.~~
- ~~**Report Generation**: Administrators can generate reports regarding lab details, hardware, software, and complaints.~~

2.3 User Characteristics

- **Administrator (Coordinator)**: Manages the overall system functionalities, including adding/editing labs, computers, complaints, and handling installation requests.
- **Technicians**: Handle the resolution of complaints and fulfill software/hardware installation requests.
- **Students and Teachers**: Submit complaints regarding computers and request software/hardware installations.

2.4 Constraints

- The project must be **completed within the semester timeframe**.
- The development team must **consist of exactly 3 members**.
- ~~The system must be developed within the given academic semester timeframe.~~
- ~~The system must support a moderate number of concurrent users (at least 50 users at once).~~
- ~~The system must be developed using Java and compatible web technologies.~~

2.5 Assumptions and Dependencies

- All users will have internet access.
- Data entered into the system is accurate and up-to-date.
- The system will rely on a backend database for storing all data, such as MySQL or PostgreSQL.

Formatted: No bullets or numbering

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Authentication

- **Description**: The system shall allow users to log in securely, with different access levels for administrators, technicians, and students.
    - **Requirements**:
        - The system shall require a username and password for login.
        - Administrators shall have access to all system functionalities, while technicians and students will have limited access based on their roles.

3.1.2 Manage Computer Details

- **Description**: The system shall allow administrators to create, read, update, and delete computer records within labs. These records shall include both hardware and software specifications.
    - **Requirements**:
        - The system shall support adding computers to specific laboratories.
        - Each computer record shall include hardware details such as processor type, RAM size, storage, and OS type.
        - The system shall allow tracking of installed software for each computer.

3.1.3 Track Installed Software

- **Description**: The system shall track the software installed on each computer in the lab.
    - **Requirements**:
        - The system shall support adding, updating, and removing software from individual computers.
        - The system shall allow viewing a list of installed software for each computer.

3.1.4 Register and View Complaints

- **Description**: The system shall allow students and teachers to register complaints regarding computer issues. Complaints will be tracked, and users can view the status of their complaints.

- **Requirements**:
    - The system shall allow students and teachers to submit complaints about hardware or software issues.
    - The system shall track complaint status (e.g., "pending," "in-progress," "resolved").
    - The system shall send notifications to users when their complaints are resolved.

3.1.5 Installation Request Management

- **Description**: The system shall allow students and teachers to submit installation requests for software or hardware. Administrators will approve or reject requests.
    - **Requirements**:
        - The system shall allow users to submit installation requests for specific software or hardware.
        - Administrators shall approve or reject installation requests and notify the users about the decision.
        - The system shall track the status of each request (e.g., "pending," "approved," "fulfilled").

3.1.6 Report Generation

- **Description**: The system shall allow administrators to generate reports on lab status, hardware details, software details, and complaints.
    - **Requirements**:
        - The system shall generate reports for lab hardware specifications and the status of software installations.
        - The system shall allow reports to be downloaded in PDF or CSV formats.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- The system shall support at least 50 concurrent users.
- The system shall generate reports within 5 seconds of a request.

3.2.2 Security Requirements

- The system shall ensure secure login with encrypted passwords.
- The system shall implement role-based access control to ensure users only have access to the features relevant to their role.
- All sensitive data, such as passwords and personal information, shall be encrypted.

3.2.3 Usability Requirements

- The system shall have an intuitive, user-friendly interface.
- The system shall provide help documentation accessible from within the system.

3.2.4 Reliability and Availability

- The system shall be available 99.9% of the time, excluding scheduled maintenance.
- The system shall be able to recover from failures within 30 minutes.

3.2.5 Maintainability

- The system shall be designed for easy updates, with modular code that allows for efficient changes.
- The code shall be well-documented to ensure ease of maintenance.

---

4. Use Cases

Use Case 1: User Login and Authentication

- **Primary Actor**: All users (Administrator, Technician, Student/Teacher)
- **Precondition**: User has a valid account.
- **Main Flow**:
    1. User navigates to the login page.
    2. User enters username and password.
    3. The system authenticates the credentials.
    4. If valid, the user is redirected to the dashboard.
    5. If invalid, the system displays an error message.
- **Postcondition**: User is logged in and has access to authorized features based on their role.

Use Case 2: Register a Complaint

- **Primary Actor**: Student/Teacher
- **Precondition**: User is logged in.
- **Main Flow**:
    1. User selects the option to register a complaint.
    2. User provides a description of the problem and attaches relevant images (optional).
    3. User submits the complaint.
    4. The system records the complaint and sends an acknowledgment.
- **Postcondition**: Complaint is recorded, and user is notified of submission.

Use Case 3: Manage Computer Details

- **Primary Actor**: Administrator
- **Precondition**: User is logged in as an administrator.
- **Main Flow**:
    1. The administrator accesses the computer management section.
    2. The administrator creates, updates, or deletes computer records.
    3. The system updates the records and displays the latest details.
- **Postcondition**: Computer records are successfully managed and displayed.

Use Case 4: Approve or Reject Installation Request

- **Primary Actor**: Administrator
- **Precondition**: User is logged in as an administrator.
- **Main Flow**:
    1. The administrator views pending installation requests.
    2. The administrator approves or rejects each request.
    3. The system updates the status of the request.
    4. The system sends notifications to the user.

- **Postcondition**: Installation request is either approved or rejected, and the user is notified.

Use Case 5: Generate Reports

- **Primary Actor**: Administrator
- **Precondition**: User is logged in as an administrator.
- **Main Flow**:
    1. The administrator selects the report generation option.
    2. The administrator selects the type of report (e.g., hardware, software, complaints).
    3. The system generates the report and allows it to be downloaded.
- **Postcondition**: The administrator downloads or views the generated report.

---

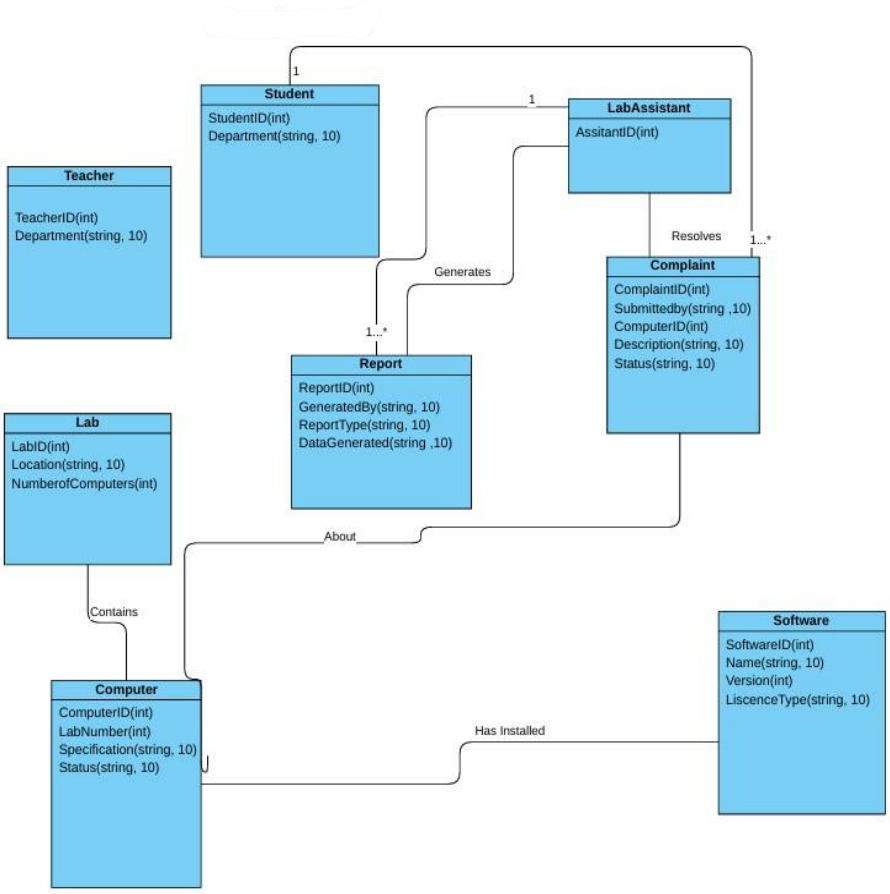
5. Appendices

A. Glossary

- **Complaint**: A report submitted by a student or teacher regarding an issue with a lab computer.
- **Installation Request**: A request submitted by a student or teacher for software or hardware installation on lab computers.

B. Acronyms

- **LMS**: Laboratory Management System

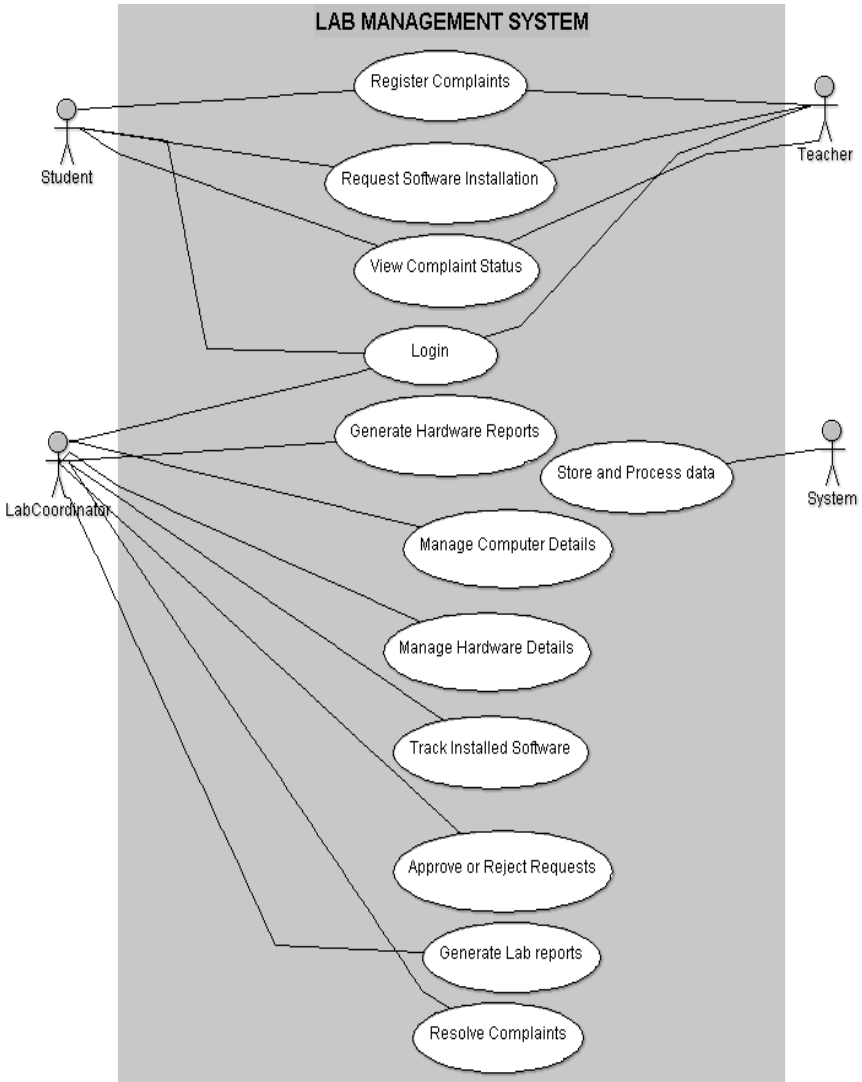


~~C.~~

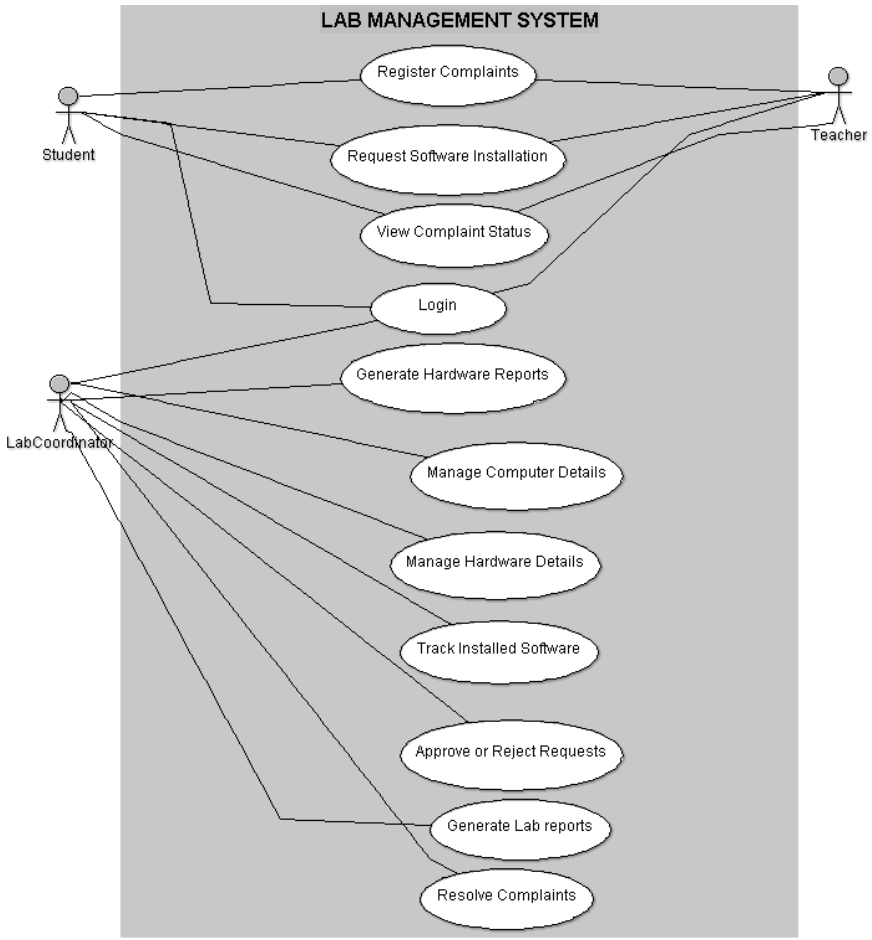
~~**External Interface Requirements**~~

- ~~The system shall support modern web browsers such as Chrome, Firefox, and Edge.~~
- ~~The system shall communicate with a backend database (e.g., MySQL, PostgreSQL).~~

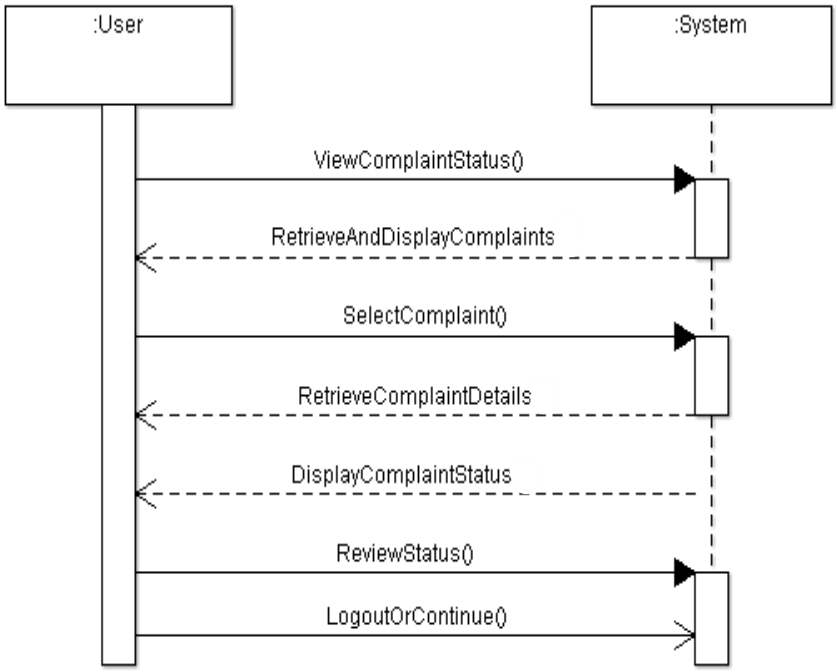
Formatted: No bullets or numbering



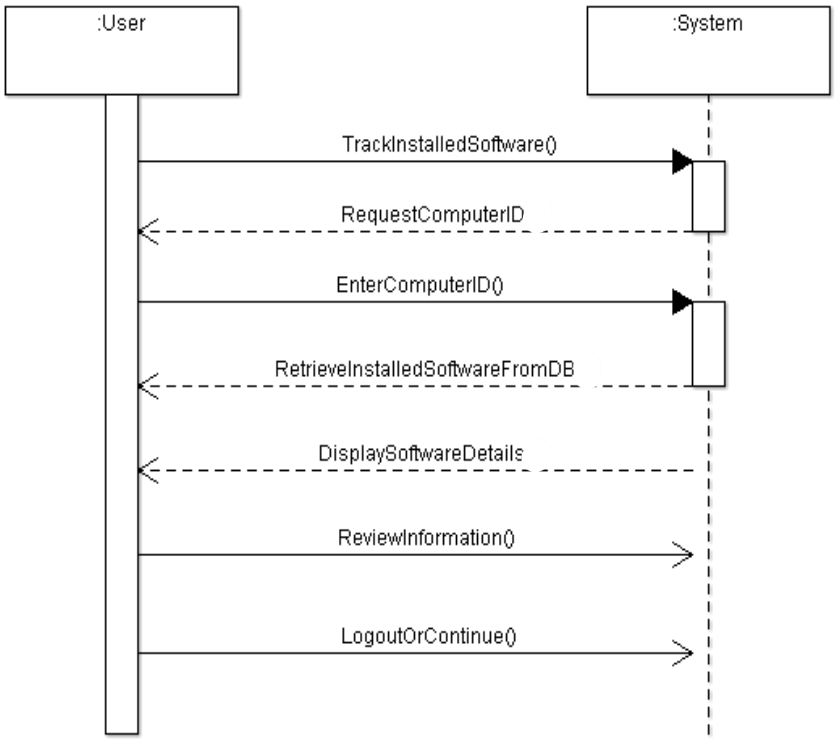
Formatted: Check spelling and grammar


LAB MANAGEMENT SYSTEM
Register Complaints
Request Software Installation
View Complaint Status
Login
Generate Hardware Reports
Manage Computer Details
Manage Hardware Details
Track Installed Software
Approve or Reject Requests
Generate Lab reports
Resolve Complaints
Student
Teacher
LabCoordinator

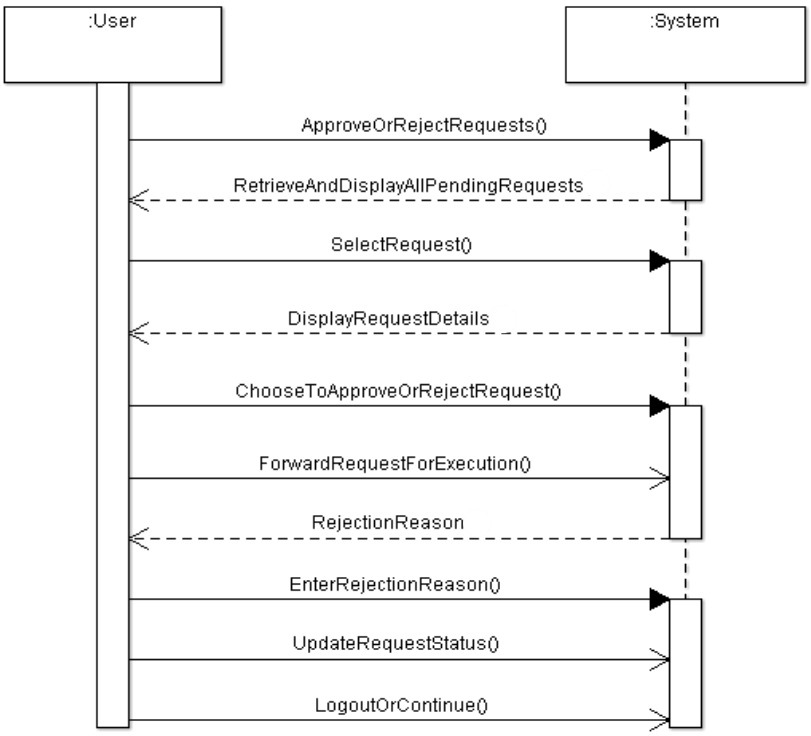
Use Case: View Complaint Status



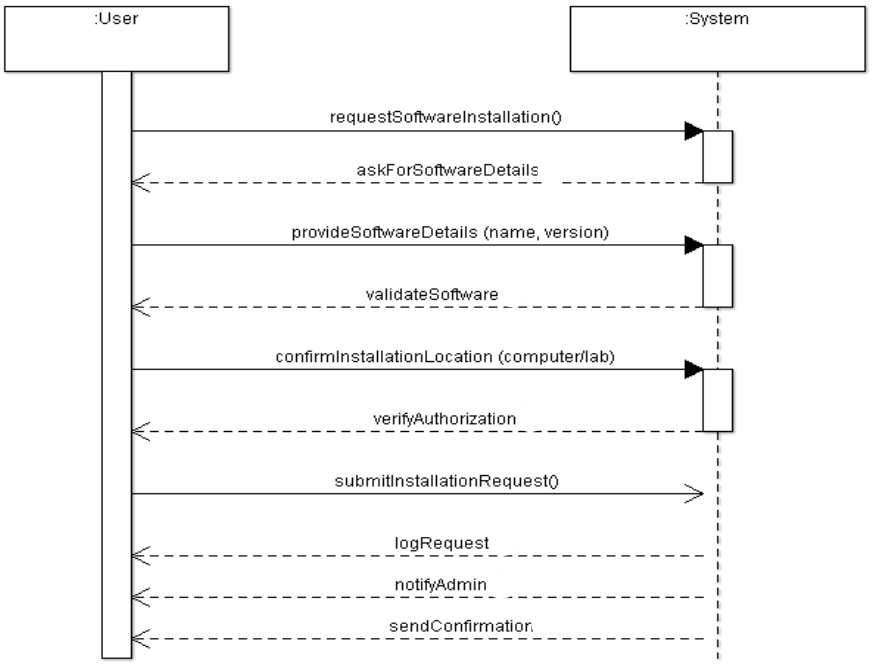
Use Case: Track Installed Software



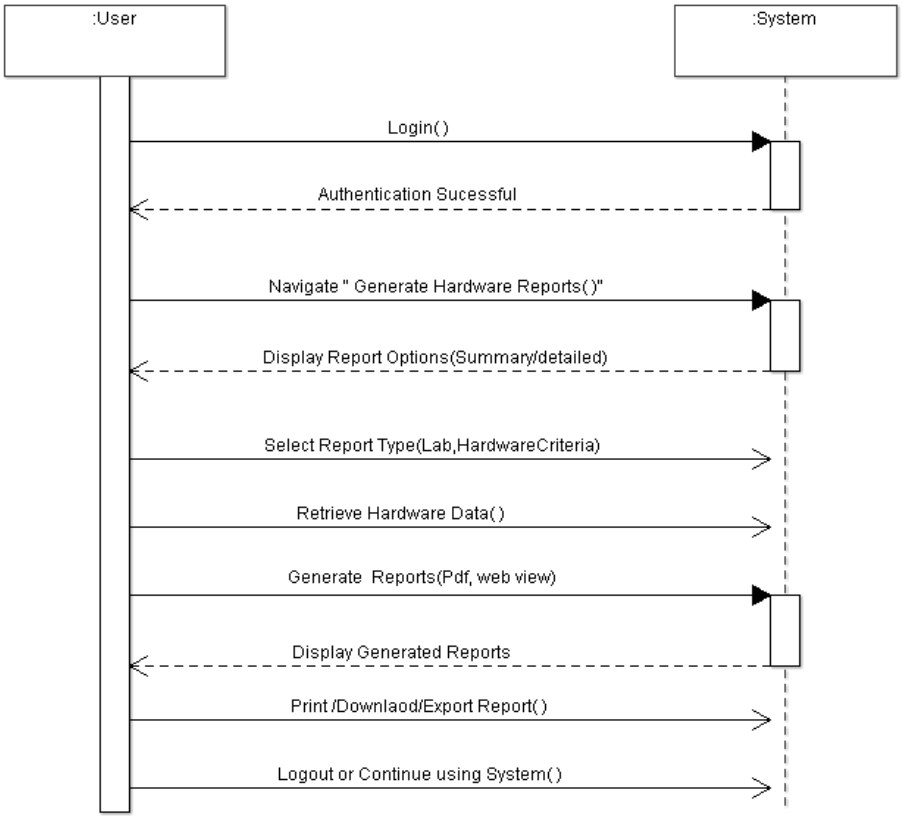
Use Case: Approve or Reject Requests



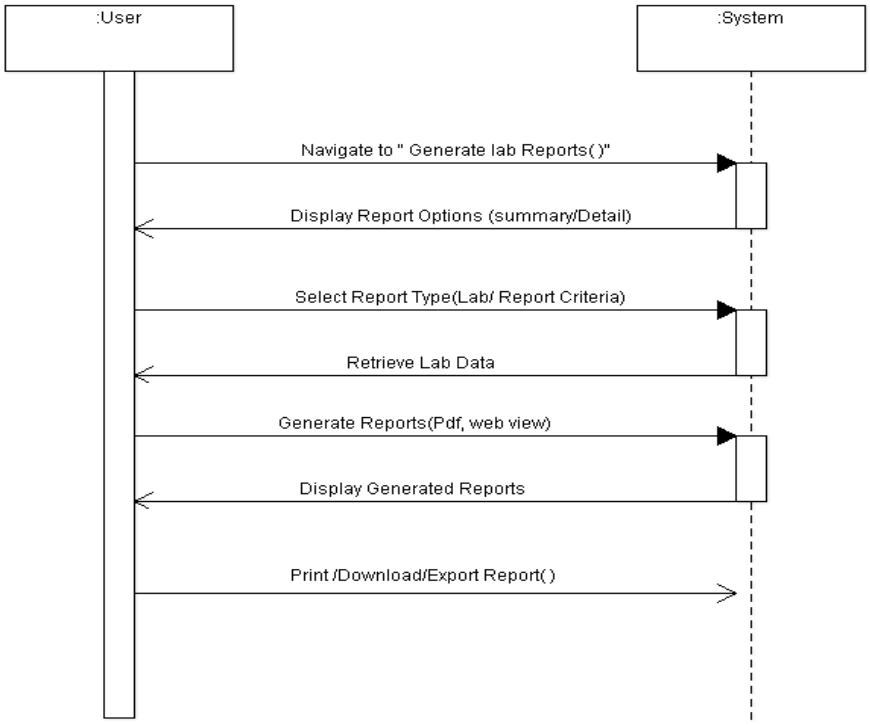
Use Case: Request Software Installation



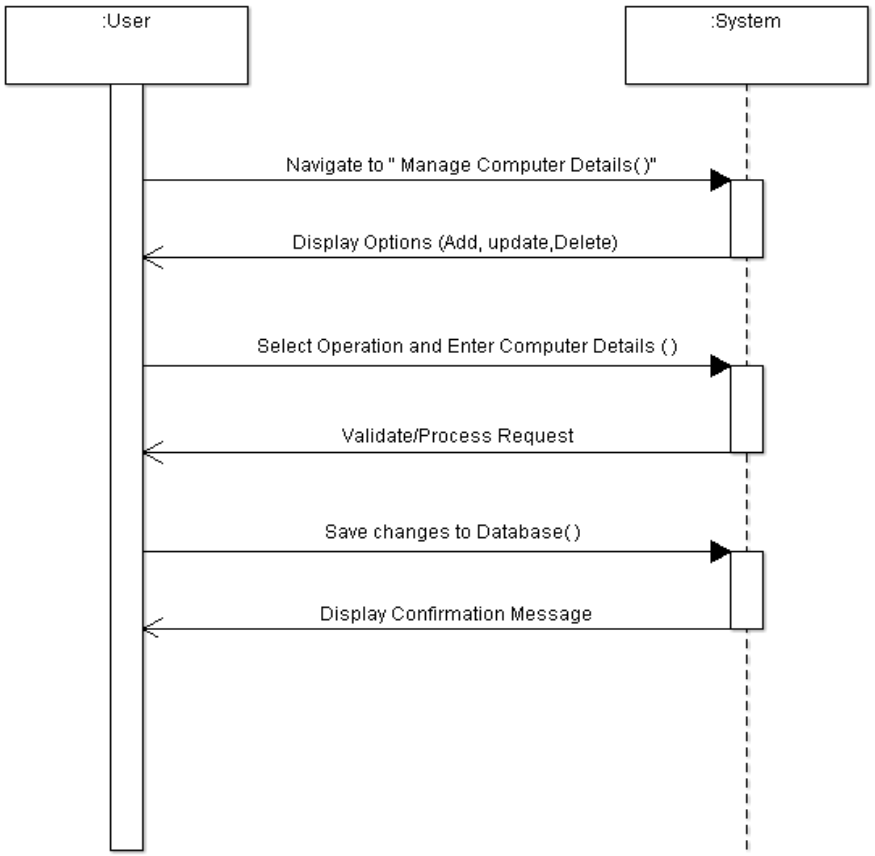
Use Case: Generate Hardware Reports



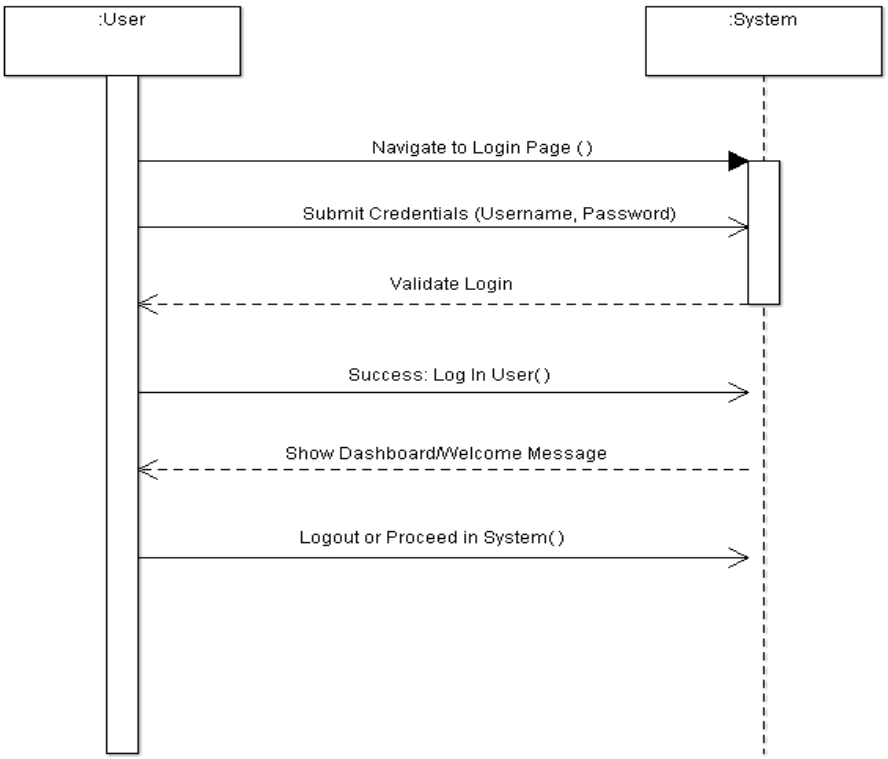
Use Case: Generate Lab Reports



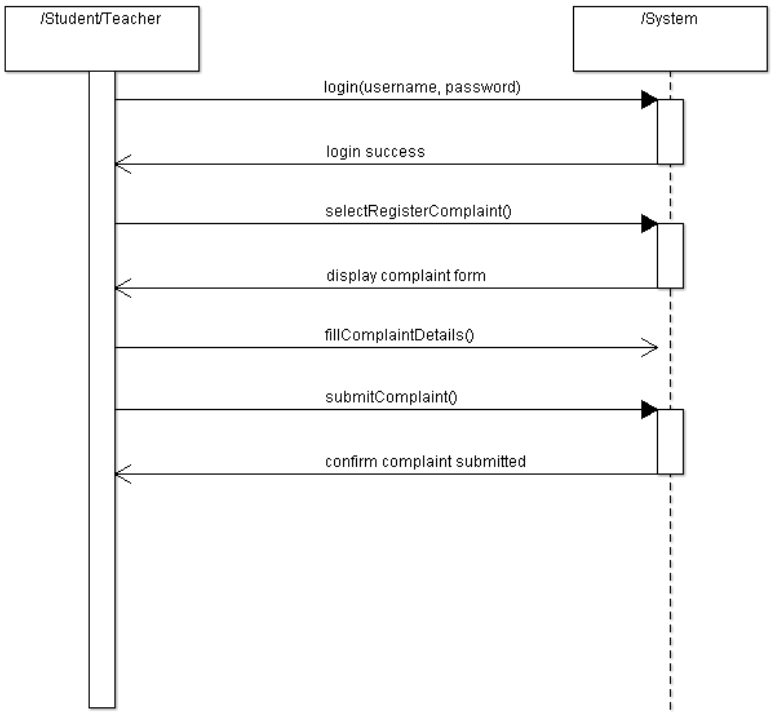
Use Case ~~Text~~: Manage Computer Details



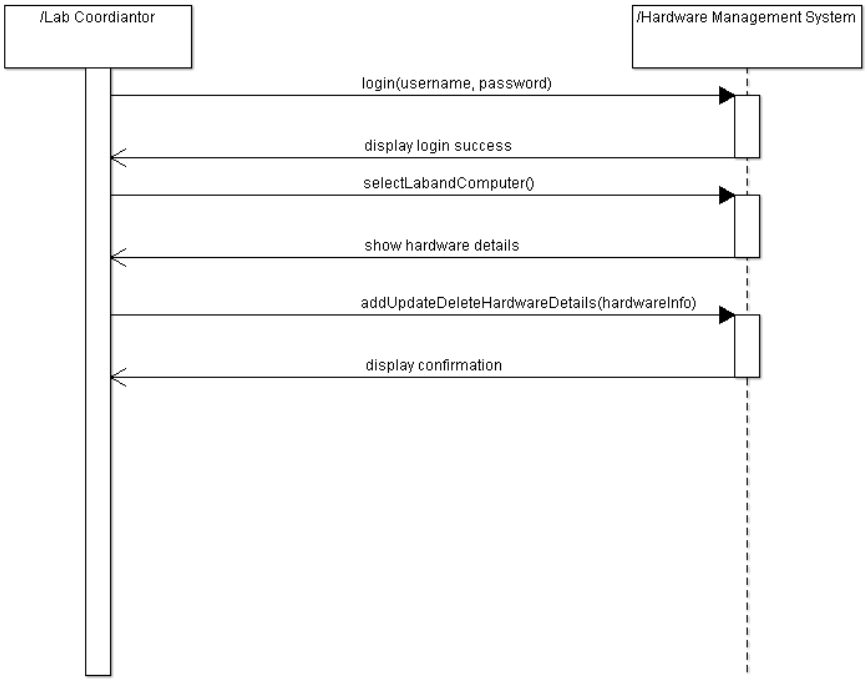
Use Case: Login



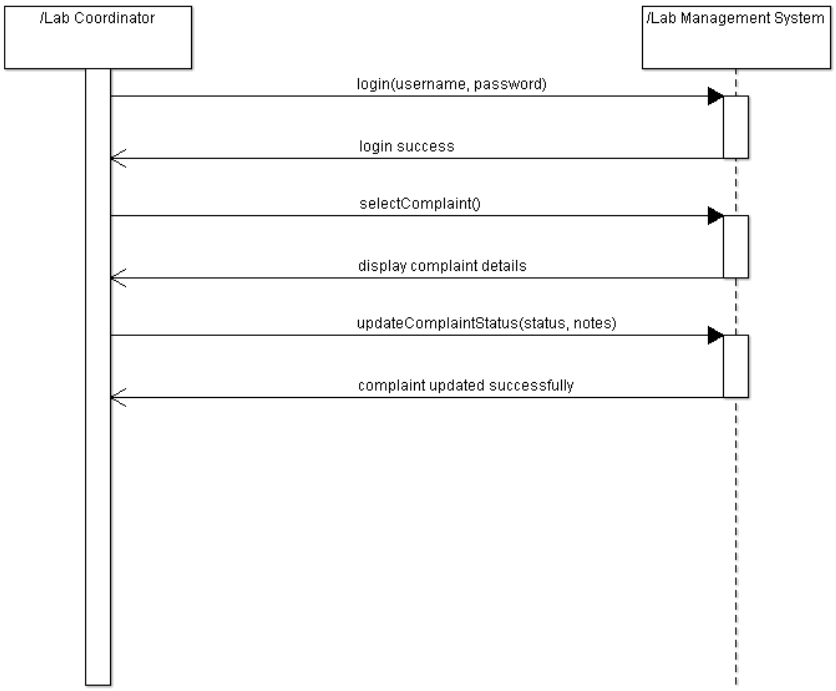
Use Case: Register Complaint



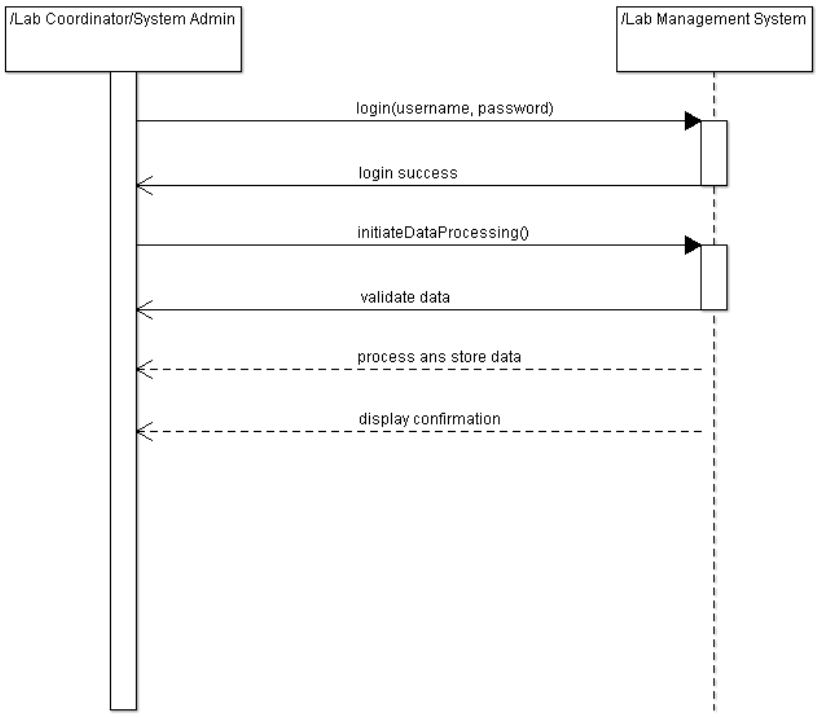
Use Case: Manage Hardware Details



Use Case: Resolve Complaints



Use Case: Store and Process Data



Formatted: Tab stops: 4.05 cm, Left