Software Requirements Specification (SRS) for Faculty Task Assignment Portal

Name	**HANISH B**
Reg. No	**7376221CS160**
Project ID	**8**
Problem Statement	**Create a template task**

Stack:

Frontend	**HTML, CSS, JavaScript**
Backend	**Apache web server, PHP with Laravel Framework**
Database	**MySQL**

Table of Contents

---

1. Introduction

1.1 Purpose

The purpose of this document is to define the Software Requirements Specification (SRS) for the Faculty Task Assignment Portal. This portal will be used by the Head of Department (HOD) to assign tasks to faculty members, ensuring that there are no duplicate assignments and all tasks are properly recorded.

1.2 Scope

This SRS covers all the functionalities required for the Faculty Task Assignment Portal. The system will allow HODs to assign tasks to faculty members, manage these tasks, and store all relevant data. The portal will ensure no duplicate tasks are assigned and will maintain a record of all tasks and their statuses.

1.3 Definitions, Acronyms, and Abbreviations

- HOD: Head of Department
- SRS: Software Requirements Specification
- Portal: Faculty Task Assignment Portal

1.4 References

- IEEE Standard for Software Requirements Specifications (IEEE 830-1998)
- Institutional guidelines for task management and faculty duties

1.5 Overview

The SRS document provides a comprehensive description of the functionalities and requirements of the Faculty Task Assignment Portal. It details the system features, external interfaces, and both functional and non-functional requirements.

---

2. Overall Description

2.1 Product Perspective

The Faculty Task Assignment Portal is a standalone web application designed to facilitate the assignment and management of tasks by the HOD to faculty members. It integrates with the institution's existing authentication system for user management.

2.2 Product Functions

- User authentication and authorization
- Task creation by HOD
- Assignment of tasks to faculty members
- Management of task details (description, type, status)
- Prevention of duplicate task assignments
- Data storage and retrieval for all task-related information

2.3 User Classes and Characteristics

- **HOD (Admin)**: Responsible for creating and assigning tasks, managing task details, and ensuring no duplicate tasks.
- **Faculty Member**: Receives assigned tasks, views task details, and updates task status.

2.4 Operating Environment

- Web-based application accessible via standard web browsers (Chrome, Firefox, Edge)

- Backend server to handle business logic and data storage
- Database for persistent storage of task and user information

2.5 Design and Implementation Constraints

- The system must be secure and protect sensitive information.
- The system must ensure data integrity and prevent duplicate task assignments.
- The user interface should be intuitive and user-friendly.

2.6 Assumptions and Dependencies

- Users will have basic knowledge of web applications.
- The institution's IT infrastructure will support the deployment of the web application.

---

3. System Features

3.1 User Authentication and Authorization

- **Description**: The system must authenticate users and authorize access based on user roles (HOD and Faculty).
- **Priority**: High
- **Stimulus/Response Sequences**: Users log in using their institutional credentials. HODs have admin privileges, while faculty have limited access.

3.2 Task Creation and Assignment

- **Description**: HODs can create tasks and assign them to specific faculty members.
- **Priority**: High

- **Stimulus/Response Sequences**: HODs input task details and select faculty members from a list. The system saves and assigns the task.

3.3 Task Management

- **Description**: HODs and faculty members can view, update, and manage task statuses.
- **Priority**: Medium
- **Stimulus/Response Sequences**: Users can update task progress, mark tasks as completed, and view task histories.

3.4 Duplicate Detection

- **Description**: The system must detect and prevent duplicate task assignments.
- **Priority**: High
- **Stimulus/Response Sequences**: Upon task creation, the system checks for existing tasks with similar details and alerts the HOD if duplicates are found.

3.5 Data Storage and Retrieval

- **Description**: The system must store all task-related data and provide efficient retrieval mechanisms.
- **Priority**: High
- **Stimulus/Response Sequences**: Data is stored in a database and can be retrieved via search functionalities.

---

4. External Interface Requirements

4.1 User Interfaces

- **Login Page**: For user authentication.
- **Dashboard**: Overview of tasks for HOD and faculty.
- **Task Creation Page**: Form for HOD to create new tasks.
- **Task Management Page**: For viewing and managing tasks.

4.2 Hardware Interfaces

- Standard web server and database server configurations.

4.3 Software Interfaces

- Integration with institutional authentication systems (e.g., LDAP, OAuth).

4.4 Communication Interfaces

- HTTPS for secure communication between client and server.

---

5. System Requirements

5.1 Functional Requirements

5.1.1 User Authentication

- The system shall authenticate users using institutional credentials.
- The system shall authorize users based on their roles.

5.1.2 Task Creation

- The system shall allow HODs to create tasks with detailed descriptions.
- The system shall provide fields for task type (lab class, subject class, etc.).

5.1.3 Task Assignment

- The system shall allow HODs to assign tasks to one or more faculty members.
- The system shall prevent the same task from being assigned to multiple faculty members simultaneously.

5.1.4 Task Management

- The system shall allow faculty to view assigned tasks.
- The system shall allow faculty to update task statuses.
- The system shall allow HODs to manage and update task details.

5.1.5 Duplicate Detection

- The system shall check for duplicate tasks during the creation process.
- The system shall alert HODs of potential duplicates and prevent assignment.

5.1.6 Data Storage

- The system shall store all task-related data in a database.
- The system shall ensure data integrity and security.

5.2 Non-Functional Requirements

5.2.1 Performance

- The system shall handle up to 1000 simultaneous users without performance degradation.
- The system shall respond to user actions within 2 seconds.

5.2.2 Reliability

- The system shall have an uptime of 99.9%.
- The system shall provide mechanisms for data backup and recovery.

5.2.3 Usability

- The system shall have an intuitive user interface.
- The system shall provide help documentation for users.

5.2.4 Security

- The system shall use HTTPS for all communications.
- The system shall protect against common web vulnerabilities (e.g., SQL injection, XSS).

5.2.5 Maintainability

- The system shall be modular to facilitate updates and maintenance.
- The system shall include logging for error tracking and debugging.

---

6. Other Non-Functional Requirements

6.1 Performance Requirements

- The system must be capable of handling peak loads without crashing.
- Response times should not exceed 2 seconds for any user action.

6.2 Safety Requirements

- The system must ensure that all data transactions are completed without loss of data.
- Regular backups must be scheduled to prevent data loss.

6.3 Security Requirements

- User data must be encrypted in transit and at rest.
- Access controls must be strictly enforced based on user roles.

---

7. Appendices

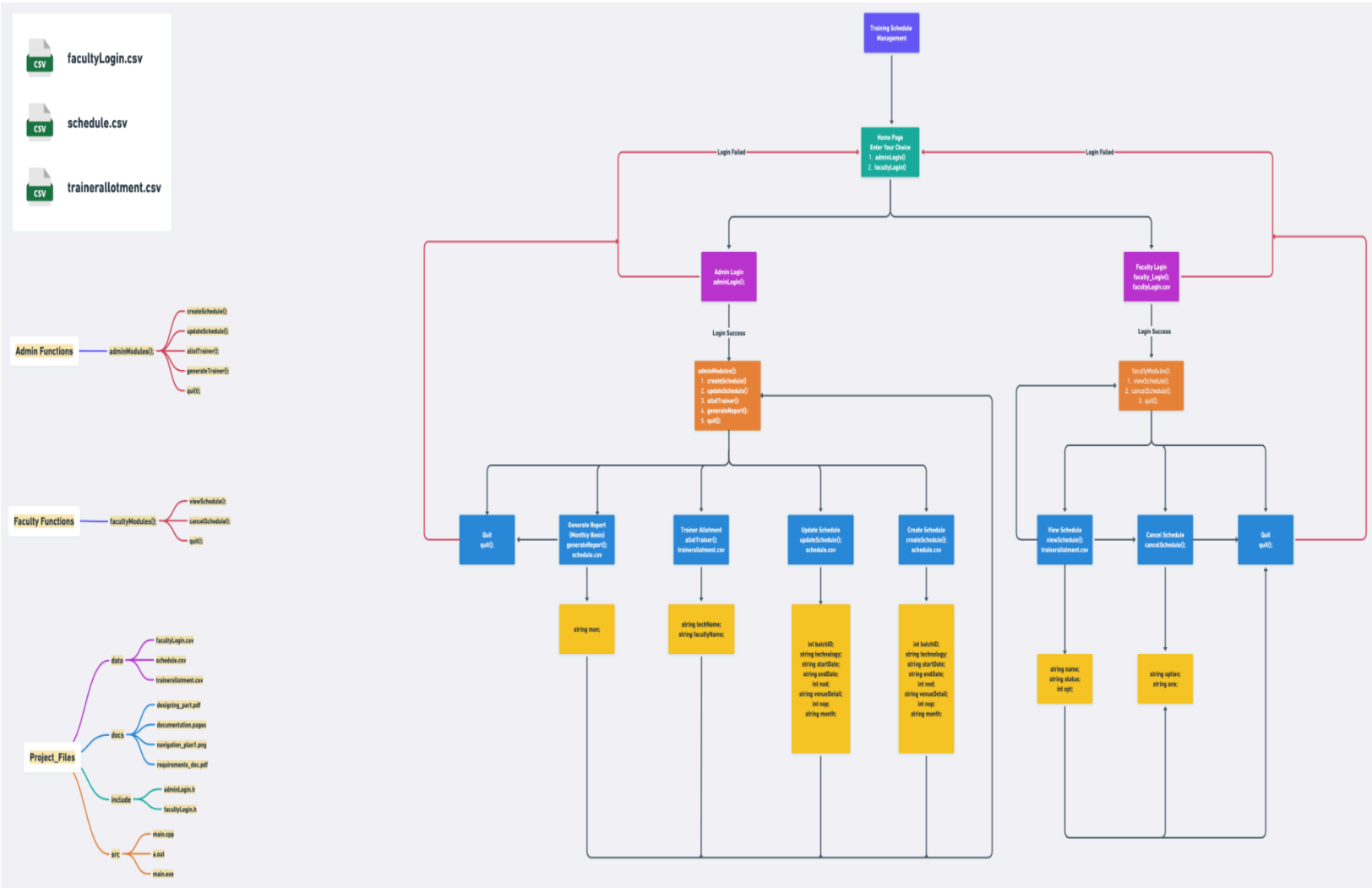
7.1 Glossary

- **HOD**: Head of Department, responsible for managing faculty and assigning tasks.

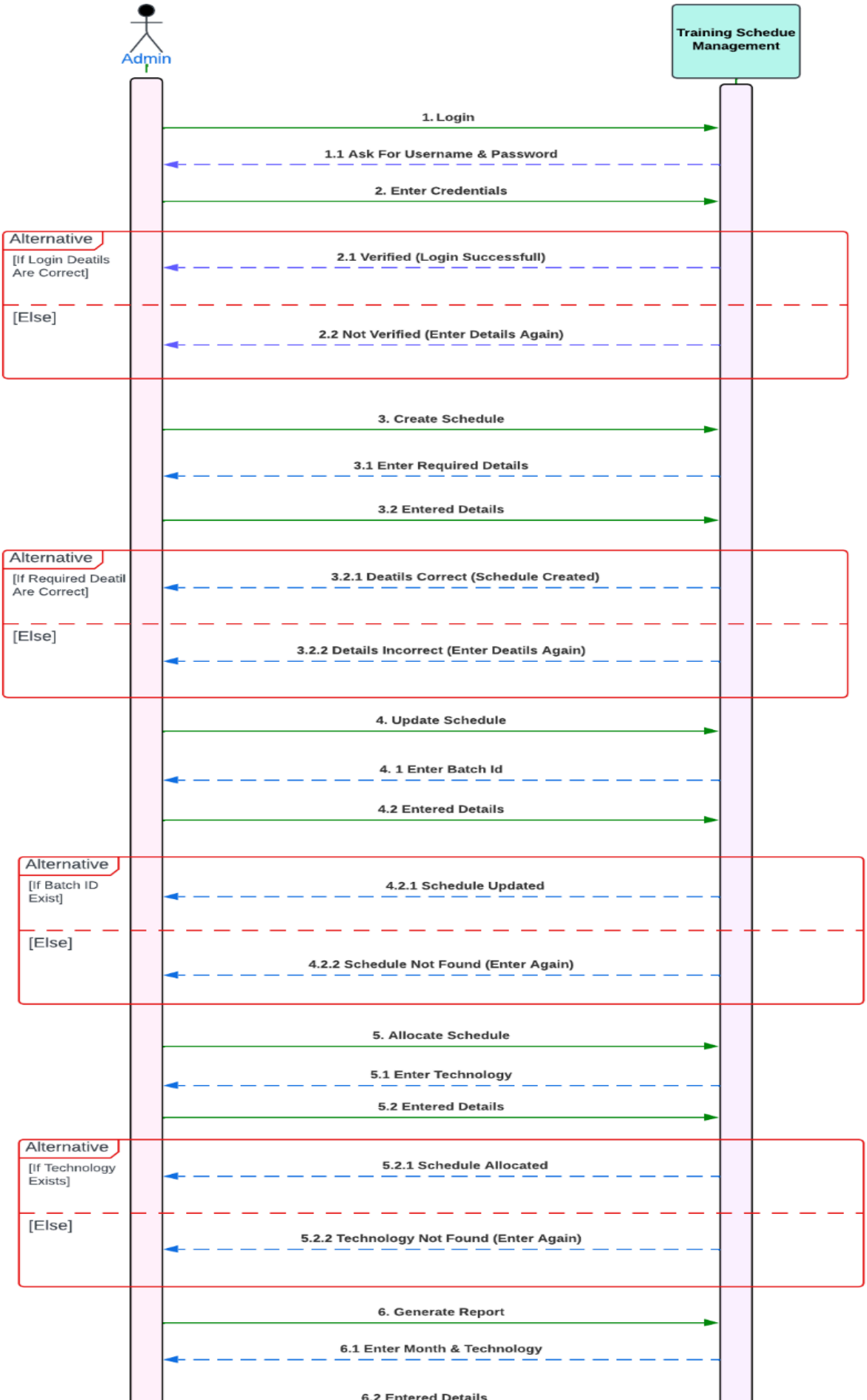
- **Faculty Member**: Staff responsible for teaching and other academic duties.

7.2 Analysis Models

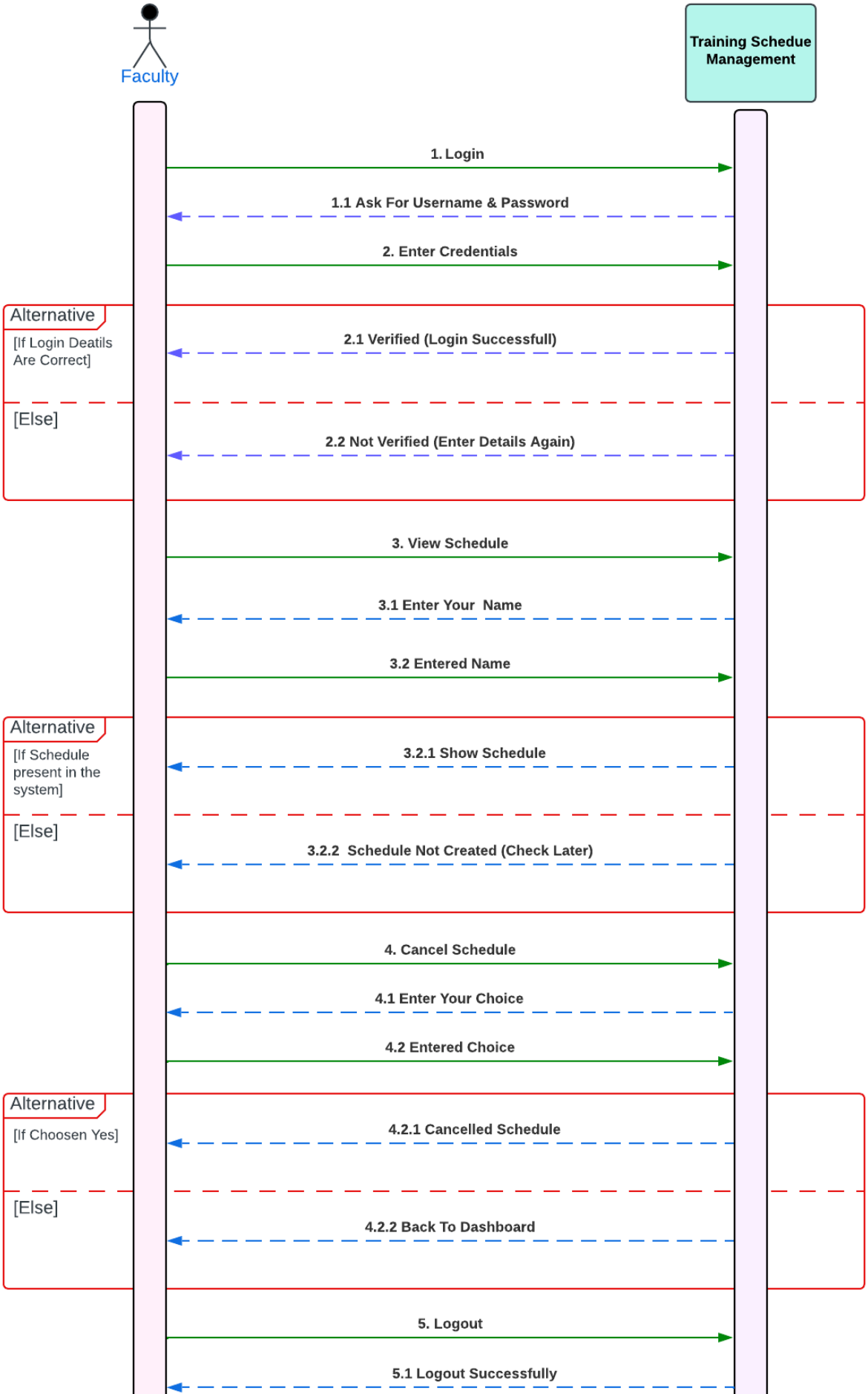
Flow Chart :-



Admin(HOD) Sequence:

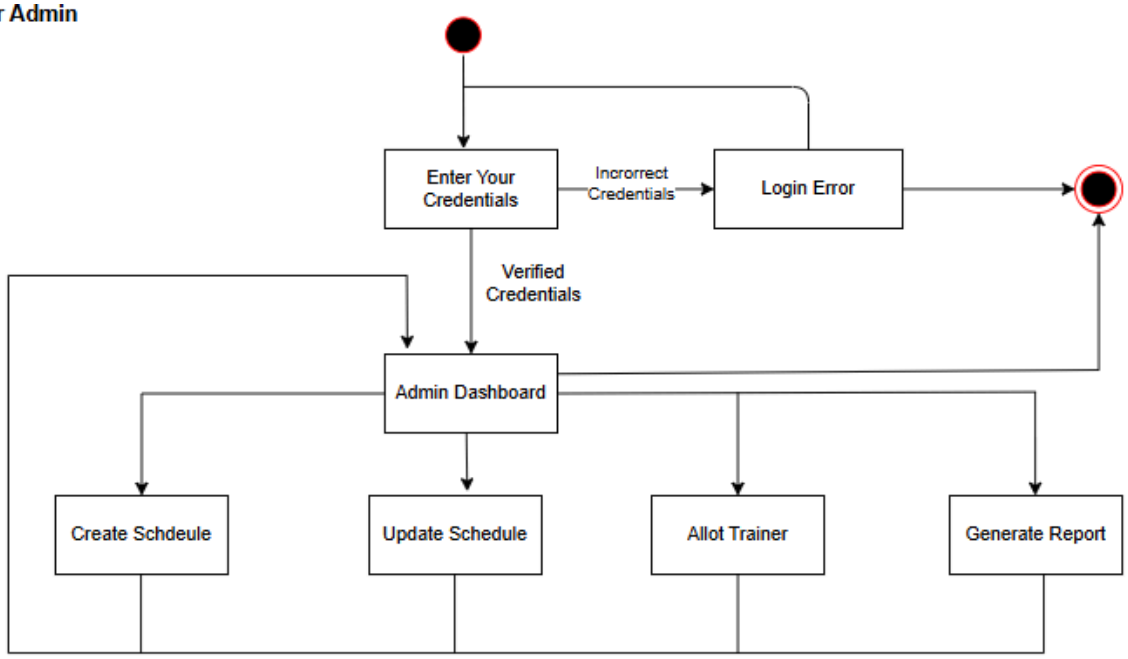

Admin
Training Schedue Management
1. Login
1.1 Ask For Username & Password
2. Enter Credentials
Alternative
[If Login Deatils Are Correct]
2.1 Verified (Login Successfull)
[Else]
2.2 Not Verified (Enter Details Again)
3. Create Schedule
3.1 Enter Required Details
3.2 Entered Details
Alternative
[If Required Deatil Are Correct]
3.2.1 Deatils Correct (Schedule Created)
[Else]
3.2.2 Details Incorrect (Enter Deatils Again)
4. Update Schedule
4. 1 Enter Batch Id
4.2 Entered Details
Alternative
[If Batch ID Exist]
4.2.1 Schedule Updated
[Else]
4.2.2 Schedule Not Found (Enter Again)
5. Allocate Schedule
5.1 Enter Technology
5.2 Entered Details
Alternative
[If Technology Exists]
5.2.1 Schedule Allocated
[Else]
5.2.2 Technology Not Found (Enter Again)
6. Generate Report
6.1 Enter Month & Technology
6.2 Entered Details

Faculty Sequence:

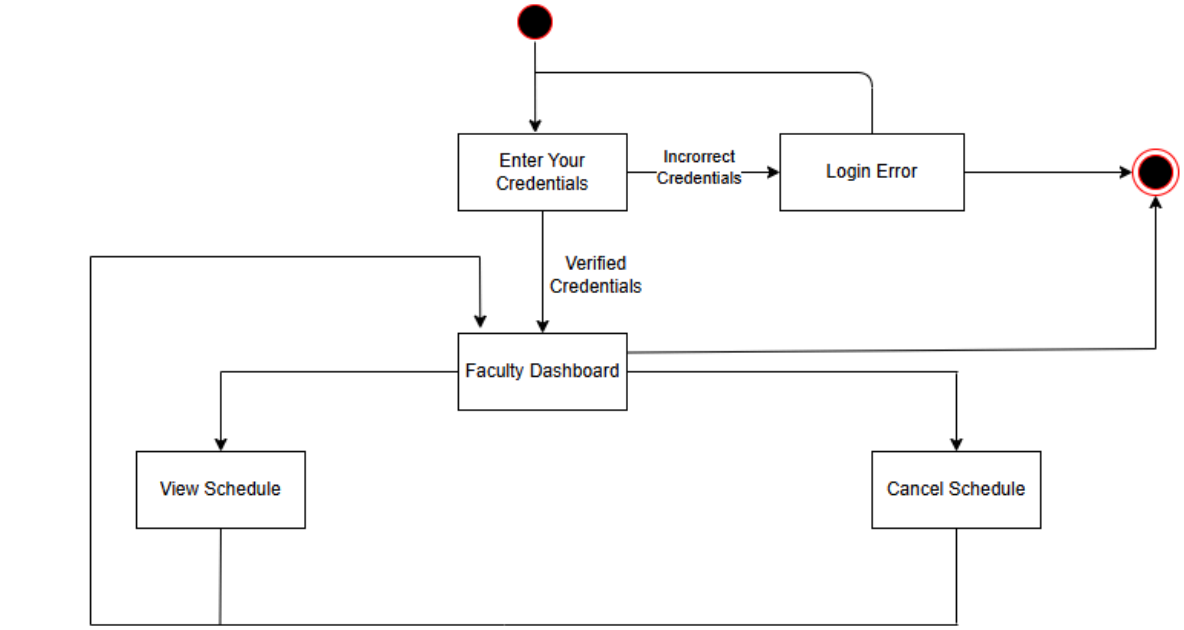

Faculty
Training Schedue Management
1. Login
1.1 Ask For Username & Password
2. Enter Credentials
Alternative
[If Login Deatils Are Correct]
2.1 Verified (Login Successfull)
[Else]
2.2 Not Verified (Enter Details Again)
3. View Schedule
3.1 Enter Your Name
3.2 Entered Name
Alternative
[If Schedule present in the system]
3.2.1 Show Schedule
[Else]
3.2.2 Schedule Not Created (Check Later)
4. Cancel Schedule
4.1 Enter Your Choice
4.2 Entered Choice
Alternative
[If Choosen Yes]
4.2.1 Cancelled Schedule
[Else]
4.2.2 Back To Dashboard
5. Logout
5.1 Logout Successfully

Project Flow:

For Admin



For Faculty



Data Flow: