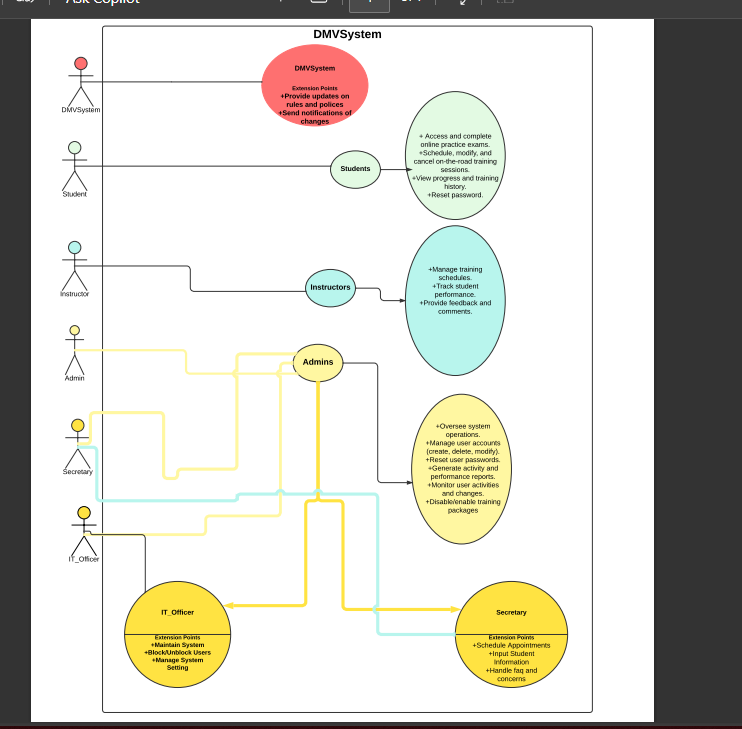
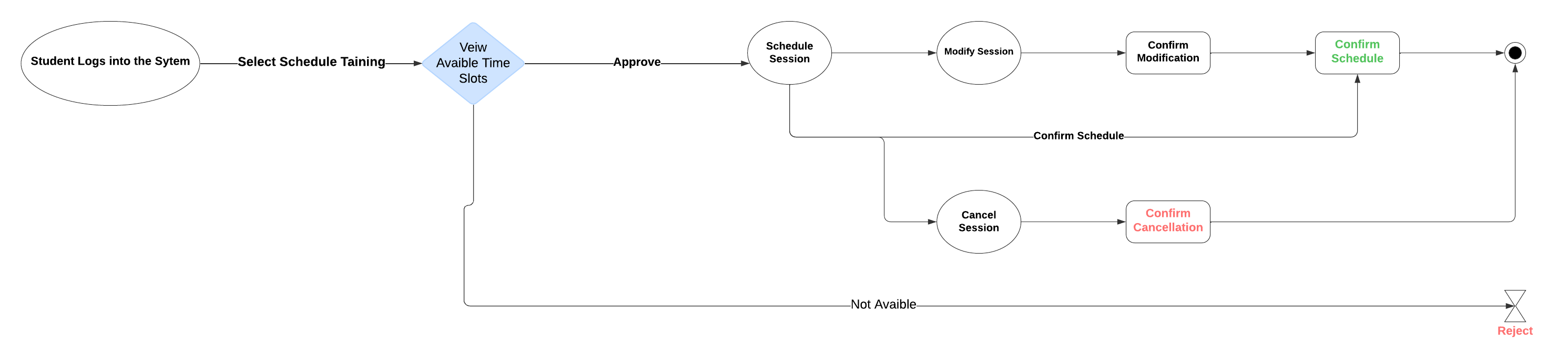
# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram



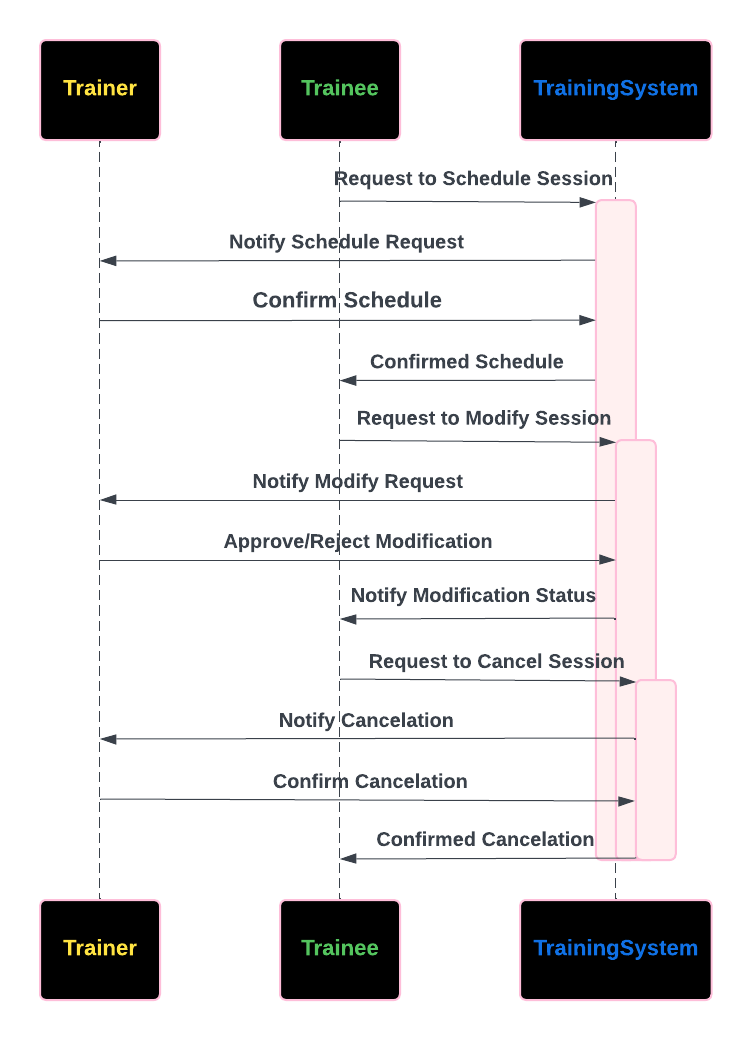
### UML Activity Diagrams



A computer diagram with a black arrow

Description automatically generated with medium confidence

### UML Sequence Diagram



### UML Class Diagram

A computer screen with many black squares

Description automatically generated

## Technical Requirements

**Hardware Requirements:**

1. **Servers**:

**Web Server:** A server capable of hosting a web-based application, supporting various user roles like students, instructors, admins, and IT officers.

**Database Server**: A robust server to manage and store user data, schedules, training progress, and system settings.

**Backup Server**: For regular backups to prevent data loss and ensure disaster recovery.

2. **Client** **Devices**:

**Desktop/Laptop:** Users (students, instructors, admins) should be able to access the system via desktop or laptop computers.

**Mobile Devices:** Compatibility with smartphones and tablets (iOS and Android) is required to allow users to access and interact with the system on-the-go.

**Software Requirements:**

**1. Operating System:**

**Server:** Linux-based systems (ex. Ubuntu Server) are preferred for the server environment, although Windows Server could also be supported.

**Client**: The system should be accessible from major operating systems including Windows, macOS, iOS, and Android.

2. **Web Application**:

**Front-End Framework:** React.js, Angular.js, or Vue.js for building a responsive and user-friendly interface.

**Back-End Framework**: Node.js, Django (Python), or ASP.NET Core (C#) to handle server-side logic, data processing, and integration with the database.

**Web Server:** Apache or Nginx for serving the web application.

3. **Database**:

**Database Management System:** MySQL, PostgreSQL, or Microsoft SQL Server for storing user information, schedules, training progress, and other relevant data.

**Data Backup and Recovery:** Automated tools for regular backups and recovery, such as AWS Backup.

4. **Security Software:**

**SSL/TLS Certificates**: To secure data transmission between the client and server.

**Authentication and Authorization:** Implementing OAuth 2.0, or JWT.

**Tools and Infrastructure:**

1. Development Tools:

**Integrated Development Environment (IDE):** Visual Studio, PyCharm, or VS Code for coding and development.

**Version Control:** Git with platforms like GitHub or GitLab for managing code versions and collaboration.

**2. Deployment and Monitoring:**

**CI/CD Pipeline:** Jenkins, GitHub Actions, or GitLab CI/CD for automated testing, building, and deployment.

**Monitoring Tools:** Tools like Prometheus and Grafana or New Relic for system monitoring and performance tracking.

**Logging:** ELK Stack or similar logging tools for tracking system activity and debugging issues.

3. **Infrastructure**:

**Cloud Services:** AWS, Azure, or Google Cloud Platform for hosting web applications, database, and other related services, make sure scalability and availability.

**Load Balancing:** Implementing load balancers to distribute traffic evenly across servers.