# CS 255 System Design Document Template – Dustin Davis

## UML Diagrams

### UML Use Case Diagram

A diagram of a company structure

AI-generated content may be incorrect.

### UML Activity Diagrams

A diagram of a flowchart

AI-generated content may be incorrect. A diagram of a flowchart

AI-generated content may be incorrect.

### UML Sequence Diagram

A diagram of a company

AI-generated content may be incorrect.

### UML Class Diagram

### A diagram of a company AI-generated content may be incorrect.

## Technical Requirements

**Hardware Requirements**

1. **Servers**
   * A cloud-based server will handle the system’s backend, allowing secure access from any location.
   * The server must support scalable infrastructure to accommodate increasing user activity.
2. **User Devices**
   * Students and staff will need access via desktops, laptops, tablets, or smartphones.
   * Devices must have stable internet access and support modern web browsers.
3. **Backup and Storage Devices**
   * Automated cloud storage should be implemented for regular backups.
   * Secure storage solutions must be in place for sensitive data, including payment records and personal user information.

**Software Requirements**

1. **Operating Systems**
   * The system should run on a cloud platform that supports Linux-based servers or Windows Server.
2. **Database Management System (DBMS)**
   * A cloud-compatible DBMS such as MySQL or PostgreSQL will be used for managing user data, lesson scheduling, and payment records.
3. **Web Framework**
   * A robust web application framework like Django (Python) or Spring Boot (Java) will be used to develop and maintain the backend system.
4. **Frontend Technologies**
   * Technologies such as HTML5, CSS3, and JavaScript frameworks like React or Angular will be used for developing the user interface.
5. **Security Software**
   * SSL certificates will be used for secure web communication.
   * Encryption tools will protect sensitive data, including passwords and payment information.

**Tools**

1. **Development Tools**
   * Lucidchart for creating diagrams and visual design elements.
   * GitHub for version control and collaborative development.
2. **Testing Tools**
   * JUnit (for Java) or PyTest (for Python) will be used for unit testing.
   * Selenium for automated testing of the web interface.
3. **Monitoring and Logging Tools**
   * New Relic or Datadog for monitoring system performance.
   * ELK Stack for logging errors and tracking system activity.

**Infrastructure Requirements**

1. **Cloud Hosting**
   * The system will be hosted on a platform such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure.
2. **Network Requirements**
   * A stable internet connection with a secure firewall is required.
   * VPN access should be provided for administrative and IT staff.
3. **Security Infrastructure**
   * Role-Based Access Control (RBAC) will ensure users can only access functions relevant to their roles.
   * Two-Factor Authentication (2FA) will be implemented for admin and IT accounts.
4. **Backup and Disaster Recovery**
   * Automated daily backups will be stored securely in the cloud.
   * A disaster recovery plan will be in place to restore data and system functionality in case of failure.

**Compliance Requirements**

1. **Data Privacy Laws**
   * The system will comply with relevant data protection regulations, including GDPR (if applicable) and CCPA for user data protection.
2. **Payment Compliance**
   * Compliance with PCI DSS standards for secure handling of credit card transactions will be ensured.