# CS 255 System Design Document Template

This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead the goal is to complete each section based on what your client’s needs are. Remove this note when you are finished, and replace all bracketed text with the relevant information.

## UML Diagrams

### UML Use Case Diagram

*A diagram of a driver pass

AI-generated content may be incorrect.*

### UML Activity Diagrams

A diagram of a test

AI-generated content may be incorrect.*A diagram of a student registration

AI-generated content may be incorrect.*

### UML Sequence Diagram

A diagram of a student

AI-generated content may be incorrect.

### UML Class Diagram

A diagram of a software company

AI-generated content may be incorrect.

## Technical Requirements

**Hardware Requirements**

* **Client Devices:**  
  Students, staff, and administrators will access the system using personal computers, tablets, or smartphones. No specialized hardware is needed beyond basic internet-enabled devices.
* **Server Infrastructure:**  
  The backend will be hosted on cloud servers to allow 24/7 uptime, backups, and remote access. The infrastructure must support secure data storage and handle concurrent user interactions.
* **Storage:**  
  Structured relational data will be stored in an SQL-based database system. Exported reports may be stored on cloud object storage such as AWS S3 or Azure Blob.

**Software Requirements**

* **Frontend Technology:**
  + HTML5, CSS3, JavaScript
  + Mobile-first design for accessibility across screen sizes
* **Backend Technology:**
  + Node.js with Express, Django, or Spring Boot
  + RESTful API endpoints to connect frontend actions with backend processing
  + JSON format for API data exchange
* **Database:**
  + **Relational Database Management System (RDBMS):** MySQL or PostgreSQL
    - Tables represent entities shown in class diagram: Student, Driver, Lesson, PracticeTest, Package, etc.
    - Relationships will be enforced via foreign key constraints.
* **Authentication:**
  + Password hashing
  + Support for secure password reset via email
  + Two-factor authentication

**Development Tools**

* **UML Modeling:** Lucidchart
* **Version Control:** Git with GitHub or GitLab for collaboration and tracking
* **IDE:** Visual Studio Code, IntelliJ IDEA, or PyCharm depending on backend stack
* **Testing Tools:**
  + Unit tests (Jest, Mocha, or JUnit)
  + UI tests (Selenium or Cypress)

**Infrastructure Requirements**

* **Cloud Hosting:** AWS, Microsoft Azure, or Google Cloud Platform
  + Supports autoscaling, secure data storage, daily backups, and high availability
* **SSL/TLS Encryption:** All web traffic must use HTTPS

**Security Requirements**

* Passwords must follow complexity rules (uppercase, lowercase, symbol, number)
* Two-factor authentication for critical roles (IT, Owner)
* Account lockout on multiple failed login attempts
* Role-based access control with distinct permissions for:
  + Owner
  + IT Officer
  + Secretary
  + Student
  + Driver
* All sensitive actions will be logged with timestamps and user info

**Compatibility and Platform Constraints**

* Runs on all major operating systems via browser
* Must perform reliably in Chrome, Firefox, Safari, and Edge
* Compatible with desktop and mobile browsers