# 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

*[In Module Six, you were asked to complete a use case diagram based on your system design. If you would like to make any adjustments to your diagram, please do so. Please insert your use case diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the clien*A diagram of a driver pass

AI-generated content may be incorrect.*t’s needs.]*

### UML Activity Diagrams

*[You were asked to choose* ***two*** *use cases and create* ***two*** *activity diagrams, one for each use case. Please insert* ***both*** *of your activity diagrams here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s needs.]* A diagram of a student

AI-generated content may be incorrect.

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

UML Sequence Diagram

*[You were asked to create a sequence diagram based on* ***one*** *of the use cases you chose. Please insert your sequence diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s needs.]A diagram of a sequence diagram

AI-generated content may be incorrect.*

### UML Class Diagram

*[You were asked to create a class diagram based on the different classes and attributes needed for your system design. You are* ***not*** *required to include methods, but you may if you wish. Please insert your class diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s requirements.]A screenshot of a computer program

AI-generated content may be incorrect.*

## Technical Requirements

*[Based on the diagrams you have created, describe the technical requirements of your system. These requirements should address the required hardware, software, tools, and infrastructure necessary for your system design.]*

Hardware Requirements

* Cloud-hosted servers provided through AWS or Microsoft Azure for scalability and uptime.
* End-user devices: desktops, laptops, tablets, or smartphones capable of running modern web browsers.
* Instructor devices: tablets or laptops with internet access for managing driving lesson schedules and student progress.
* Backup servers to ensure redundancy and disaster recovery.

Software Requirements

* Operating Systems: Windows, macOS, Android, and iOS (web-based platform—no local installation required).
* Web Browsers: Google Chrome, Firefox, Edge, Safari (latest versions supported).
* Server Environment: Linux or Windows Server environment configured for web hosting.
* Database: MySQL or PostgreSQL for persistent storage of user accounts, lessons, exams, packages, and reports.
* Web Framework: Java Spring Boot or Node.js for backend logic and REST API implementation.
* Frontend Framework: React.js or Angular for the user interface, ensuring responsive design for mobile and desktop.
* Authentication & Security Libraries: OAuth 2.0 / HTTPS with SSL encryption and hashed passwords (bcrypt or SHA-256).

Network & Infrastructure

* Cloud deployment: Amazon Web Services (AWS) EC2 and S3 or Microsoft Azure equivalents.
* Load balancing and autoscaling to handle peak traffic (e.g., simultaneous exam sessions).
* Continuous integration tools such as Jenkins or GitHub Actions for testing and deployment.
* Content Delivery Network (CDN) for optimized delivery of static assets (images, scripts, course materials).

Tools & Development Environment

* IDE: Eclipse, Visual Studio Code, or IntelliJ IDEA for development.
* Diagramming & Documentation: Lucidchart for UML diagrams and Microsoft Word for documentation.
* Version Control: Git with GitHub or GitLab repository.
* Testing Tools: JUnit or Jest for automated testing; Postman for API validation.