# 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 company structure

Description automatically generated*

### UML Activity Diagrams

*A screenshot of a diagram

Description automatically generated* A diagram of a process

Description automatically generated

### UML Sequence Diagram

*A diagram of a process

Description automatically generated*

### UML Class Diagram

*A screenshot of a computer screen

Description automatically generated*

## Technical Requirements

The DriverPass system requires a robust infrastructure to ensure smooth operation and scalability. At the hardware level, the system will need cloud-based servers with high availability and redundancy to host the platform, ensuring accessibility across multiple devices. These servers must be equipped with sufficient processing power, memory, and storage to handle concurrent users, high data volumes, and the generation of real-time reports. End-users, including customers and staff, can access the system through standard devices such as desktop computers, tablets, and smartphones, requiring modern browsers like Google Chrome, Safari, or Firefox.

The software requirements include a secure, web-based application framework supported by a relational database management system (e.g., PostgreSQL or MySQL) to manage user accounts, reservations, and lesson data. The backend system will use a programming language like Python or Java for logic and integrations, while the frontend will rely on HTML5, CSS3, and JavaScript frameworks (e.g., React or Angular) to create a user-friendly interface. The system must also include APIs for external integrations, such as pulling DMV updates and compliance requirements.

To support development, deployment, and maintenance, the system will rely on tools like GitHub for version control, CI/CD pipelines for automated testing and deployment, and cloud platforms such as AWS or Azure for hosting. Security tools, including SSL/TLS encryption, multi-factor authentication (MFA), and firewalls, will be necessary to protect sensitive data and ensure compliance with security standards. The infrastructure should also support routine updates, monitoring, and backups to maintain system reliability and adaptability to future needs.