# 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 people with a diagram

Description automatically generated with medium confidence

### UML Activity Diagrams

A diagram of a flowchart

Description automatically generated

A diagram of a process

Description automatically generated

### UML Sequence Diagram

*A diagram of a system

Description automatically generated*

### UML Class Diagram

*A diagram of a computer program

Description automatically generated*

## Technical Requirements

### ****1. Hardware Requirements****

* **Server Infrastructure**:
  + A dedicated web server to host the DriverPass application. It should have sufficient CPU, memory, and storage to handle multiple simultaneous users and data processing tasks.
  + A database server to manage and store user data, lesson schedules, and reservation information. The server should be equipped with redundant storage and backup capabilities.
  + Workstations for secretaries, IT officers, and admins with standard configurations (modern processors, 8GB+ RAM, 256GB+ storage) for accessing the system and performing day-to-day operations.

### ****2. Software Requirements****

* **Operating System**:
  + Server: Linux (e.g., Ubuntu Server) or Windows Server, depending on the organizational preference.
  + Client Machines: Windows, macOS, or Linux.
* **Web Server**:
  + Apache or Nginx to serve the DriverPass application, handling HTTP requests and routing them to the appropriate application components.
* **Database Management System (DBMS)**:
  + MySQL or PostgreSQL to store and manage relational data such as user information, lesson schedules, reservations, and tracking logs.
* **Application Framework**:
  + A web application framework such as Django (Python) or Spring Boot (Java) for building and deploying the DriverPass system. This framework should support MVC (Model-View-Controller) architecture to separate concerns and improve maintainability.
* **Development Tools**:
  + An Integrated Development Environment (IDE) like Visual Studio Code, Eclipse, or PyCharm for development.
  + Version control tools such as Git, with repositories hosted on GitHub or GitLab.

### ****3. Tools and Utilities****

* **CASE Tool**:
  + Lucidchart or draw.io for creating and maintaining UML diagrams (use case, class, sequence, activity diagrams).
* **Security Tools**:
  + SSL/TLS certificates for securing data in transit (HTTPS).
  + Firewalls and Intrusion Detection Systems (IDS) for protecting the network and application servers from unauthorized access and attacks.
  + User authentication tools, such as OAuth or LDAP, to manage secure access and roles within the system.

### ****4. Infrastructure Requirements****

* **Networking**:
  + A reliable and secure network infrastructure that supports both wired and wireless connections for all devices interacting with the system.
  + Virtual Private Network (VPN) access for remote employees or users to securely connect to the system.
* **Backup and Recovery**:
  + A comprehensive backup solution that regularly backs up the database, application data, and configuration files. Off-site storage or cloud-based backup options should be considered for disaster recovery.
* **Scalability and Redundancy**:
  + The infrastructure should be scalable to handle increased traffic and data as the company grows. Load balancers and redundant servers should be implemented to ensure high availability and minimize downtime.
* **Compliance and Updates**:
  + Integration with DMV systems for updates on rules, policies, and test content. This will require secure APIs or data feeds from the DMV.
  + Regular system updates and patches to maintain security and functionality, particularly in compliance with any regulatory requirements.