# 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 system

Description automatically generated*

### UML Activity Diagrams

*A diagram of a flowchart

Description automatically generated*

A diagram of a flowchart

Description automatically generated

### UML Sequence Diagram

*A grid of lines with words

Description automatically generated*

### UML Class Diagram

*A diagram of a test

Description automatically generated*

## Technical Requirements

***Technical Requirements for the DriverPass System***

***Hardware Requirements:***

1. ***Client Devices****:*
   * *Devices such as laptops, desktops, tablets, or smartphones to access the DriverPass platform.*
   * *Minimum specifications:*
     + ***Processor****: 2 GHz or faster.*
     + ***RAM****: At least 4 GB.*
     + ***Storage****: Minimum of 500 MB free for browser or application data.*
   * *Compatibility with modern web browsers or dedicated mobile applications.*
2. ***Server Infrastructure****:*
   * *A cloud-based server or dedicated server for hosting the application and database.*
   * *Recommended server specifications:*
     + ***Processor****: Multi-core CPU, 3 GHz or faster.*
     + ***RAM****: 16 GB or higher for efficient performance under load.*
     + ***Storage****: 500 GB SSD for fast access to data.*
   * ***Backup System****: Redundant servers or a cloud backup solution to ensure data integrity.*
3. ***Networking****:*
   * *Reliable internet connection with a minimum speed of 10 Mbps for users.*
   * *Server infrastructure with a high-bandwidth connection to handle concurrent users efficiently.*

***Software Requirements:***

1. ***User Interface****:*
   * *Front-end built using a responsive framework like* ***React*** *to ensure compatibility across devices.*
2. ***Back-End Systems****:*
   * ***Programming Language****: Python, Java, or Node.js for scalable server-side logic.*
   * ***Framework****: Django, Flask, or Express.js for efficient application development.*
3. ***Database****:*
   * *Relational Database Management System (RDBMS) like* ***MySQL*** *to store user, test, instructor, and appointment data.*
   * *Alternatively, a NoSQL database like* ***MongoDB*** *if flexible data structures are needed.*
4. ***Middleware and APIs****:*
   * *RESTful APIs for communication between the front-end and back-end.*
   * *Third-party APIs for features like email confirmations, payment processing, or calendar integration.*
5. ***Operating System****:*
   * *Server-side OS: Linux (Ubuntu, CentOS) for stability and cost-effectiveness.*
   * *Client-side compatibility: Windows, macOS, iOS, and Android.*

***Tools:***

1. ***Development Tools****:*
   * *IDEs: Visual Studio Code, IntelliJ IDEA, or Eclipse.*
   * *Version control:* ***Git*** *with GitHub or GitLab for collaboration and version tracking.*
   * *Testing tools: Selenium for automated UI testing and Postman for API testing.*
2. ***Deployment Tools****:*
   * *CI/CD pipelines using GitHub Actions*
   * *Hosting platforms like AWS, Microsoft Azure, or Google Cloud for production environments.*
3. ***Monitoring and Analytics****:*
   * *Tools like Google Analytics to monitor application performance and user behavior.*

***Infrastructure Requirements:***

1. ***Authentication and Security****:*
   * *Secure login system with two-factor authentication (2FA).*
   * *Use HTTPS for secure communication.*
   * *Encryption protocols for sensitive data (e.g., passwords, personal information).*
2. ***Scalability and Availability****:*
   * *Load balancing to handle high traffic during peak hours.*
   * *Auto-scaling servers to ensure system reliability as user demand grows.*
3. ***Data Storage and Backup****:*
   * *Cloud-based storage solution for scalability (e.g., AWS S3 or Azure Blob Storage).*
   * *Daily automated backups to prevent data loss.*
4. ***User Support****:*
   * *Help desk software for managing user inquiries.*
   * *Live chat or chatbot integration for quick assistance.*