# 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 program

Description automatically generated with medium confidence

### UML Activity Diagrams

A screenshot of a computer screen

Description automatically generated

A diagram of a software process

Description automatically generated

### UML Sequence Diagram

A diagram of a software project

Description automatically generated

### UML Class Diagram

A diagram of a software application

Description automatically generated with medium confidence

## Technical Requirements

With the development of the DriverPass system, I have outlined both functional and nonfunctional requirements, which have directly informed the technical requirements needed to implement the system. Below, I will discuss the key technical requirements based on the system’s needs.

First is Hardware Requirements. There are a few that need to be taken into consideration for this system, such as Servers, Databases and Client Devices. The DriverPass system requires robust server infrastructure to handle data storage, processing, and network requests. Given the need for scalability, the servers must be capable of supporting a growing number of students and test scheduling requests. Initially, a cloud-based server solution such as Amazon Web Services or Microsoft Azure, would be ideal due to it scalability and flexibility. Next separate database servers are necessary for managing data related to students, driving tests, practice tests, and schedules. These servers should be optimized for fast data retrieval and updates, given the nature of the real-time scheduling and test scoring. Finally End users, primarily students, will access the system through a variety of devices (desktops, laptops, tablets, smartphones.) The system needs to be optimized for cross-platform compatibility, requiring responsive design and client-side performance optimization.

Next, I will address the many software requirements for this system including Operating systems, Web Server Software, Database Management systems, and Programming Languages. The server infrastructure should run on a Linux-based system for stability, security and cost-effectiveness. Apache or Nginx will be used to serve the web applications proving reliable handling of HTTPS request and integration with other system components. MySQL or PostgreSQL could be used as the Database system, providing reliable data storage, ACID compliance and performance optimization capabilities. The system should be primarily developed in PHP for back-end logic, with front-end development using HTML, CSS, and JavaScript. A framework such as Laravel could be employed to streamline development and enforce best practices. Finally, the system must support modern web browsers such as Chrome, Firefox and Safari to ensure that broad accessibility is available.

There are a few tools that could be used in this instance to help ensure proper code handling, version control and testing. There are various Integrated Development Environments that could be used in this instance. Such as Visual Studio code or PHPStrom to write, debug and maintain the codebase. Version control should be implemented in this case for tracking changes, collaboration and maintaining code integrity. Things such as GitHub or GitLab could be used. Tools such as PHPUnit for PHP unit testing, Selenium for automated browser testing and Postman for API testing will ensure the system’s reliability and performance.

Finally, is Infrastructure. There are a few things that could help with this such as Cloud Infrastructure, Continuous Integration/Continuous Deployment and Load Balancing. As mentioned, AWS or Azure will provide the cloud infrastructure necessary for scalable storage, computing power, and networking. These services will also facilitate disaster recovery, data backup and system monitoring. A Continuous Integration/Continuous Deployment pipeline using tools like Jenkins or GitLab CLI will automate testing and deployment, ensuring that updates to the system are efficiently rolled out with downtime. Finally Load Balancing will help to maintain high performance and availability, especially as the user base grows, load balancers will distribute incoming traffic across multiple servers, preventing any single server from becoming a bottleneck.

In conclusion, the technical requirements stated above are designed to ensure that the DrivePass system meets both its functional and nonfunctional requirements. The combination of robust hardware, reliable software, essential development tools, and scalable infrastructure provides a solid foundation for delivering a system that is efficient, secure, and capable of growing with the user base.