# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram

*[A diagram of a driver pass system

AI-generated content may be incorrect.]*

### UML Activity Diagrams

*[A diagram of a program

AI-generated content may be incorrect.*

*Schedule Lesson Activity Diagram.*

*A diagram of a test

AI-generated content may be incorrect.*

*Take Practice Test Diagram]*

### UML Sequence Diagram

*[A diagram of a project

AI-generated content may be incorrect.]*

### UML Class Diagram

*[A diagram of a computer

AI-generated content may be incorrect.]*

## Technical Requirements

*[*The DriverPass system will be developed using a three-tier architecture consisting of a presentation layer, an application layer, and a data layer. The presentation layer will be a responsive web application that communicates with the application layer through RESTful APIs, allowing for modularity, scalability, and potential future integration with mobile platforms. The data layer will use a relational database to store information about students, instructors, lessons, vehicles, and test results. This architecture supports maintainability and efficient communication between system components while keeping data secure and consistent.

The system will be hosted in a cloud-based environment such as AWS or Microsoft Azure to provide reliability, scalability, and automatic backup functionality. Load balancing will be used to manage concurrent user activity, and separate servers will handle web application traffic and database transactions. Object storage will be used for learning materials, reports, and data exports, ensuring that large files can be accessed efficiently. The infrastructure will include automatic scaling and high availability to maintain strong performance under peak usage.

The software stack will use industry-standard technologies to promote efficiency and security. The backend will be built using Java with Spring Boot or Python with FastAPI for handling business logic and API services. The frontend will be created using a modern JavaScript framework such as React or Angular to deliver a user-friendly, accessible experience. Data will be stored in a MySQL or PostgreSQL database, and GitHub will be used for version control and collaboration. Lucidchart will support UML modeling and design documentation, while Jenkins or GitHub Actions will be used to implement continuous integration and deployment.

Security will be a top priority throughout the design and implementation of DriverPass. The system will use role-based access control to ensure that students, instructors, secretaries, administrators, and IT staff have permissions appropriate to their responsibilities. All communication will be secured using SSL/TLS encryption, and sensitive data will be encrypted both in transit and at rest. Passwords will be hashed, and account lockout policies will protect against brute-force attempts. Additionally, audit logs will track administrative actions, and all database queries will be parameterized to prevent SQL injection attacks.

The system is designed to perform efficiently and reliably, maintaining an average page load time of under two seconds and a response time of less than 500 milliseconds for lesson scheduling and test operations. Uptime will be maintained at or above 99.9% through redundant architecture and rolling deployment strategies that minimize downtime during maintenance. Cloud monitoring and logging tools will ensure continuous visibility into system performance.

Certain assumptions and constraints guide the design of the DriverPass system. The first release will support only web browsers, with mobile support planned for future versions. Instructors will manage their own availability calendars, while lesson booking will automatically validate time and resource conflicts. Payment processing will be handled through a PCI-compliant third-party service. DMV data will be integrated securely through scheduled API connections, and system maintenance will be conducted by IT personnel with limited administrative privileges to preserve system integrity.*]*