# 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 project

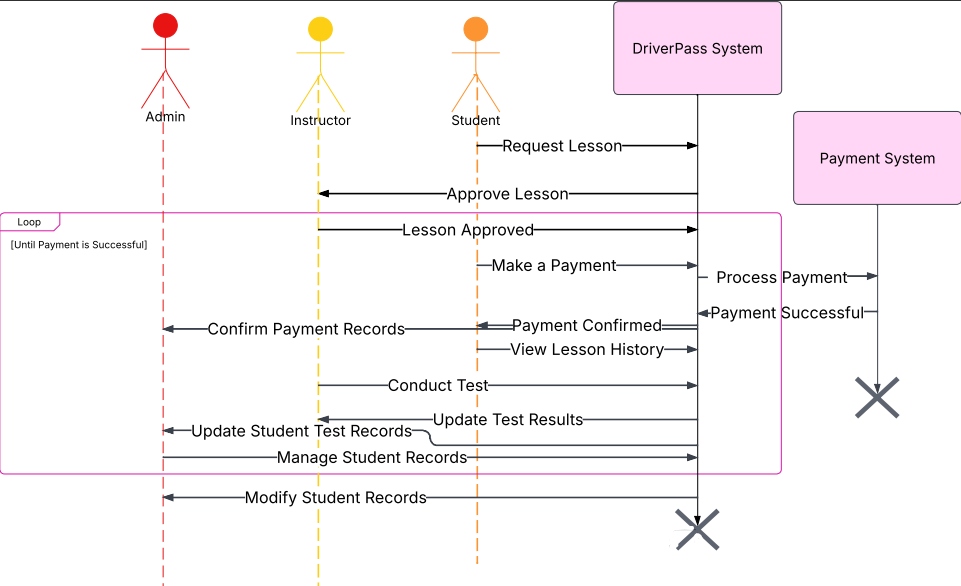
AI-generated content may be incorrect.

### UML Activity Diagrams

A diagram of a student making payment

AI-generated content may be incorrect.

### UML Sequence Diagram

**

### UML Class Diagram

*A diagram of a company

AI-generated content may be incorrect.*

## Technical Requirements

The DriverPass System requires a robust technical infrastructure to ensure smooth operations, security, and scalability. For hardware, the system will need a cloud-based or on-premise server to host the application, along with a dedicated database server for managing student records, lesson history, and payment transactions. End users, including students, instructors, and administrators, will access the system via desktops, laptops, or mobile devices. To facilitate seamless communication between system components, networking equipment such as routers, switches, and firewalls must be in place.

In terms of software, the system will run on an operating system such as Windows Server or Linux (Ubuntu or CentOS) for hosting purposes. A web server, such as Apache Tomcat or Nginx, will be required to handle HTTP requests efficiently. For data management, the system will utilize a relational database like MySQL or PostgreSQL, or alternatively, a NoSQL solution such as MongoDB. The backend will be developed using Java with the Spring Boot framework, ensuring a structured and scalable architecture, while the frontend will be designed using HTML, CSS, and JavaScript with frameworks like React.js or Angular for an interactive user experience. Hibernate will be used for database interactions, and JUnit for testing functionalities. Additionally, a payment gateway such as Stripe or PayPal will be integrated to handle secure financial transactions.

Several tools will be employed during development and maintenance. Developers will use IntelliJ IDEA or Eclipse as their primary IDEs, and version control will be managed through GitHub or GitLab. For designing and refining system diagrams, tools such as Lucidchart, Microsoft Visio, or PlantUML will be utilized. Containerization with Docker will allow for efficient deployment, while Kubernetes will help orchestrate containers in a cloud environment. Continuous integration and deployment (CI/CD) pipelines will be implemented using Jenkins, GitHub Actions, or GitLab CI/CD to streamline the deployment process.

The infrastructure of the system must be scalable and secure. Hosting will be managed through cloud platforms like AWS, Azure, or Google Cloud to ensure high availability and performance. Authentication mechanisms such as OAuth 2.0 or JWT will be implemented to protect user data and prevent unauthorized access. Regular backups of student records and payment data will be scheduled to prevent data loss, ensuring business continuity. Additionally, system monitoring tools like the ELK Stack (Elasticsearch, Logstash, Kibana) or Prometheus and Grafana will be deployed to track system performance, detect anomalies, and improve overall efficiency.

This combination of hardware, software, tools, and infrastructure will create a well-integrated and efficient DriverPass System, capable of handling lesson scheduling, payment processing, student record management, and instructor interactions seamlessly.