# 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

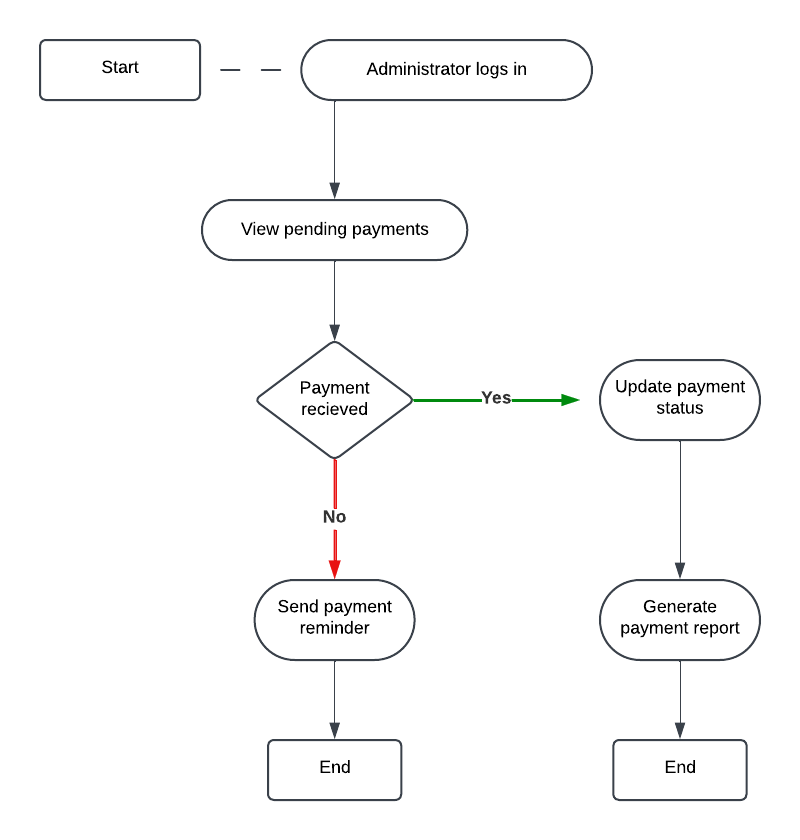
### UML Use Case Diagram

A diagram of a user flow

Description automatically generated

### UML Activity Diagrams

Manage payments/schedule lessons



A flowchart of a program

Description automatically generated

### UML Sequence Diagram

Schedule lesson

A diagram of a diagram

Description automatically generated

### UML Class Diagram

A diagram of a string

Description automatically generated

## Technical Requirements

**Hardware Requirements:**

1. Servers:

* A cloud-based server to host the application and database
* Adequate CPU (e.g., Intel Xeon processors) and RAM (16–32 GB) to handle multiple user requests simultaneously.

1. Workstations:

* Computers for system administrators, secretaries, and instructors. Minimum specifications:
* CPU: Intel i5 or higher
* RAM: 8–16 GB
* Storage: 256 GB SSD
* Network: High-speed internet access (10–100 Mbps)

1. Devices for end users:

* Laptops, desktops, tablets, or smartphones

1. Networking hardware:

* Routers and switches for internal networking
* A reliable internet connection with redundant failover to ensure high availability

**Software Requirements:**

1. Operating system:

* Servers: Linux (Ubuntu) or Windows Server.
* Workstations: Windows 10/11 or macOS.

1. Database Management System (DBMS):

* MySQL, PostgreSQL, or MongoDB to store user data, lesson schedules, payment information, and DMV updates.

1. Web server:

* Apache HTTP Server or NGINX for hosting the DriverPass application

1. Application frameworks:

* Backend: Node.js, Python with Django/Flask, or Java Spring Boot for server-side processing.
* Frontend: React.js, Angular, or Vue.js for the user interface.

1. Payment gateway integration:

* Software to securely integrate payment gateways (e.g., Stripe or PayPal) to process transactions.

1. Development tools:

* CASE tools like Lucidchart for UML diagrams.
* IDEs such as Visual Studio Code, Eclipse, or IntelliJ IDEA for development.
* Version control systems like Git (GitHub, GitLab, or Bitbucket).

**Tools:**

* Lucidchart: For creating UML diagrams (use case, activity, sequence, and class diagrams).
* GitHub or GitLab: For version control and collaborative coding.
* Docker: For containerizing the application to ensure consistent deployment across different environments.
* CI/CD Pipelines: Tools like Jenkins, GitHub Actions, or GitLab CI/CD for automating testing and deployment.
* Monitoring Tools: Software such as Prometheus and Grafana for real-time performance monitoring.

**Infrastructure requirements:**

* Cloud-Based Deployment
* Load Balancer
* Data Backup
* Security Infrastructure