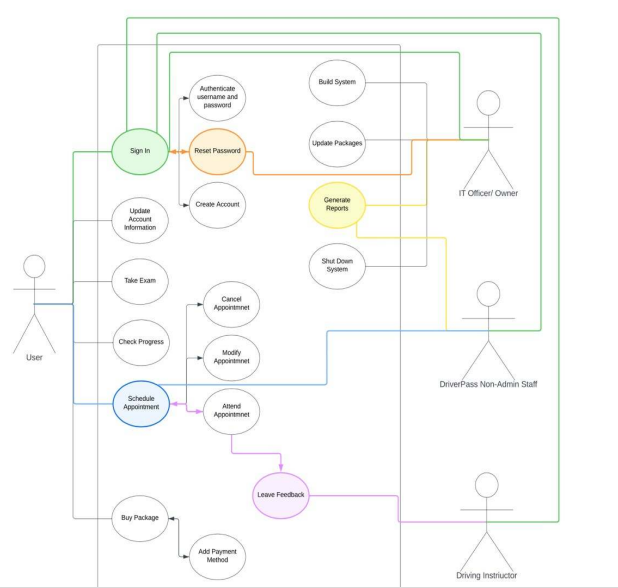
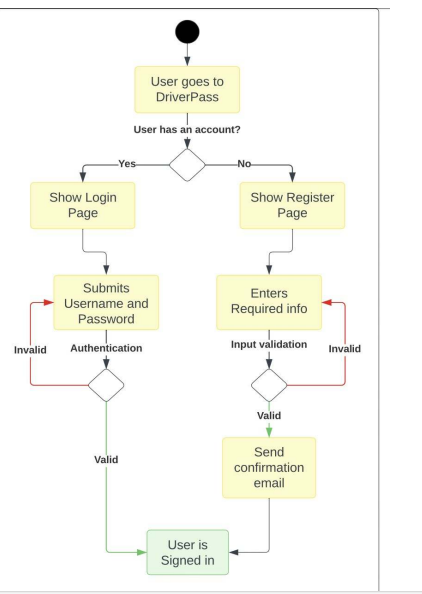
# CS 255 System Design Document Template

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

*[*

**

*Description: The use case diagram illustrates the interactions between different users (actors) and the system. Key use cases include:*

*Customer Registration*

*Lesson Reservation*

*Lesson Cancellation*

*Password Reset*

*Admin Account Management*

*DMV Updates Notification]*

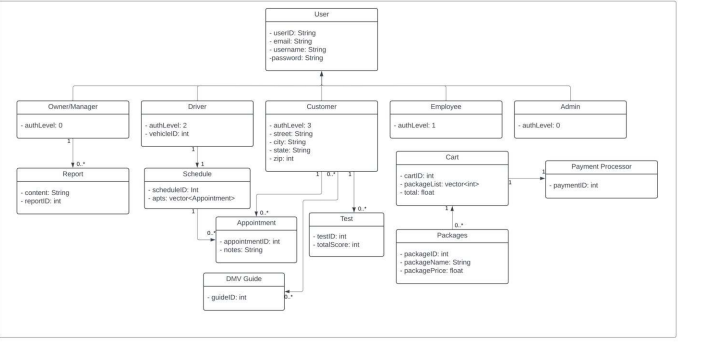
### UML Activity Diagrams

[

### UML Sequence Diagram

*[Description: This sequence diagram details the interactions between the customer, the system, and the admin during the lesson reservation process, showing the order of operations and data exchanges.]*

### UML Class Diagram

*[*

Description: The class diagram defines the main classes and attributes needed for the DriverPass system, including:

Customer: Attributes like first name, last name, address, email, and credit card information.

Lesson: Attributes for lesson date, time, duration, and associated driver.

Package: Attributes for package name and description (e.g., Package One, Package Two).

Admin: Attributes for admin role and permissions.

DMV Notification: Attributes for compliance updates and notifications.

## Technical Requirements

*[Hardware Requirements*

*Server Specifications:*

*A cloud-based server to host the application (e.g., AWS, Azure).*

*Minimum requirements: 8GB RAM, 4 CPU cores, and 100GB SSD storage to handle user data and application load.*

*User Devices:*

*Users should be able to access the system from any computer or mobile device with internet connectivity.*

*Software Requirements*

*Operating System:*

*A modern web server OS (e.g., Linux, Windows Server) compatible with web applications.*

*Database Management System:*

*A relational database management system (RDBMS) like MySQL or PostgreSQL for storing user data, lesson reservations, and reports.*

*Programming Languages:*

*Frontend: HTML, CSS, JavaScript (React or Angular).*

*Backend: Java or Python (Django or Flask) for handling business logic.*

*Frameworks:*

*Use web frameworks (like Spring for Java or Django for Python) for building the application structure.*

*Security Tools:*

*Implementation of SSL certificates for secure data transmission.*

*Use of secure password hashing algorithms (e.g., bcrypt) for user password management.*

*Tools*

*Development Tools:*

*IDEs like IntelliJ IDEA (Java) or PyCharm (Python) for software development.*

*Version control system (e.g., Git) for managing code changes.*

*Testing Tools:*

*Automated testing tools (e.g., Selenium) for end-to-end testing of the web application.*

*Performance testing tools (e.g., JMeter) to ensure system stability under load.*

*Infrastructure Requirements*

*Cloud Infrastructure:*

*The system should be hosted on a reliable cloud service provider (e.g., AWS, Azure, Google Cloud) to ensure scalability and reliability.*

*Backup Solutions:*

*Automated backup solutions to regularly back up user data and application states to prevent data loss.*

*Monitoring Tools:*

*Use of application performance monitoring (APM) tools (e.g., New Relic, Datadog) to monitor application performance and user activity.]*