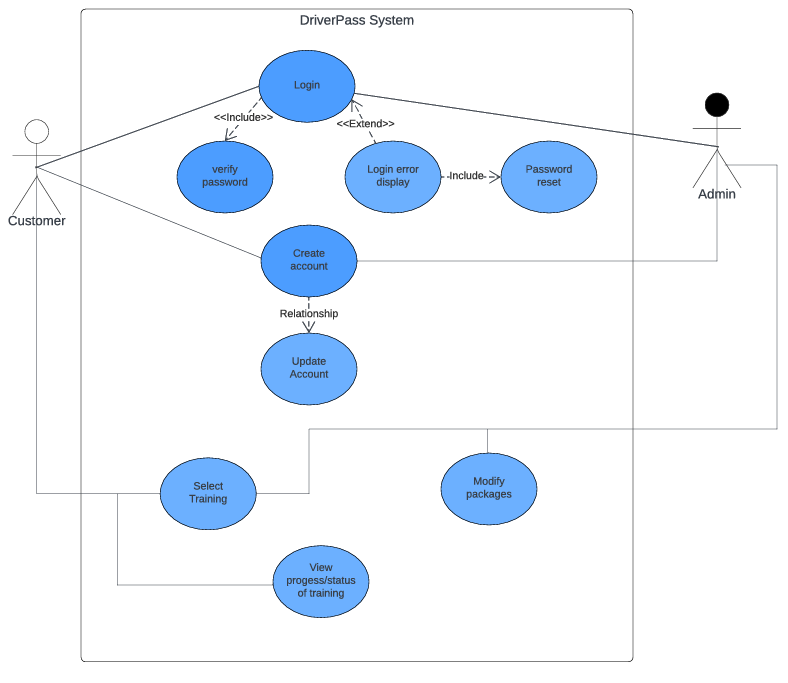
# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram – Very broad overview of how the DriverPass system functions in this subsystem. The use cases captured in the diagram encompass the core functionalities required for the DriverPass system to operate effectively. These include driver registration, scheduling driving tests, conducting tests, recording results, generating reports, and sending notifications. Additionally, the diagram ensures that all necessary actors and functionalities are accounted for, providing a comprehensive overview of the system's operations.



### 2. UML Activity Diagrams – Details the login process for DriverPass from a user perspective and the account creation process if needed.

### *A diagram of a computer process Description automatically generated*

### UML Sequence Diagram – The workflow chosen for the sequence of someone selecting a package available, scheduling, and then ultimately paying.

A diagram of a process

Description automatically generated

### UML Class Diagram

A screenshot of a computer

Description automatically generated

## Technical Requirements

1. ***Hardware Requirements:***
   * *Server Infrastructure: The system requires a robust server infrastructure to host the database and application logic. This could involve multiple servers for load balancing and redundancy.*
   * *Database Server: A dedicated database server is needed to store driver information securely. This could be a relational database management system depending on the specific requirements.*
   * *Networking Equipment: Switches, routers, and firewalls are required to facilitate communication between clients and the server infrastructure.*
2. ***Software Requirements:***
   * *Backend Framework: A backend framework is necessary to handle server-side logic and interact with the database. Popular choices include Django, Flask (Python), Express (Node.js), or Spring (Java).*
   * *Database Management System (DBMS): The system needs a reliable DBMS to store and retrieve driver information efficiently. The choice of DBMS depends on factors like scalability, data complexity, and transaction requirements.*
   * *Frontend Framework: A frontend framework is required for developing the user interface of the system. Options include React, Angular, or Vue.js for building responsive and interactive web interfaces.*
   * *Operating System: The servers may run on Linux-based distributions (e.g., Ubuntu, CentOS) for stability, security, and ease of maintenance.*