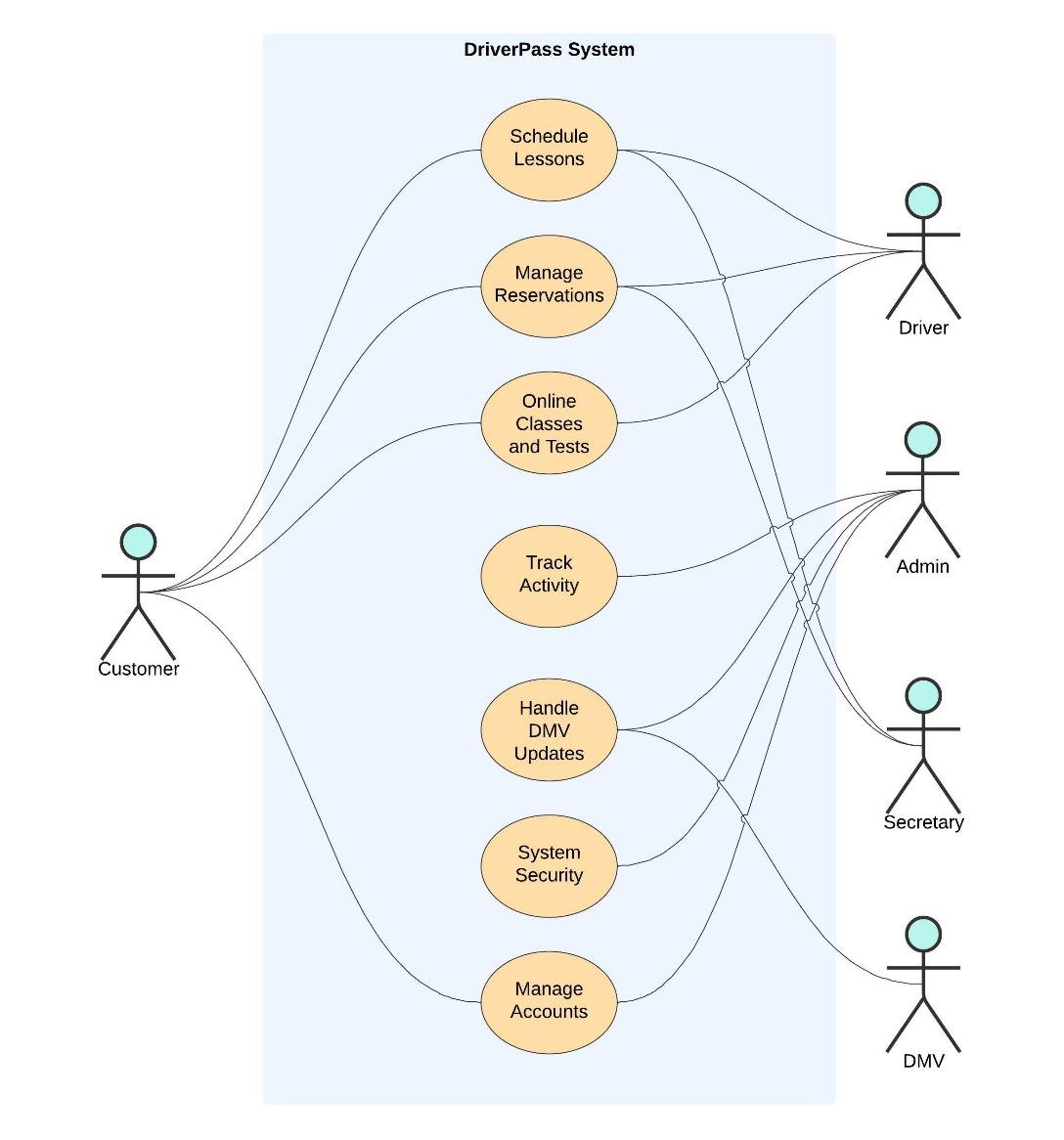
# CS 255 System Design Document Template

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

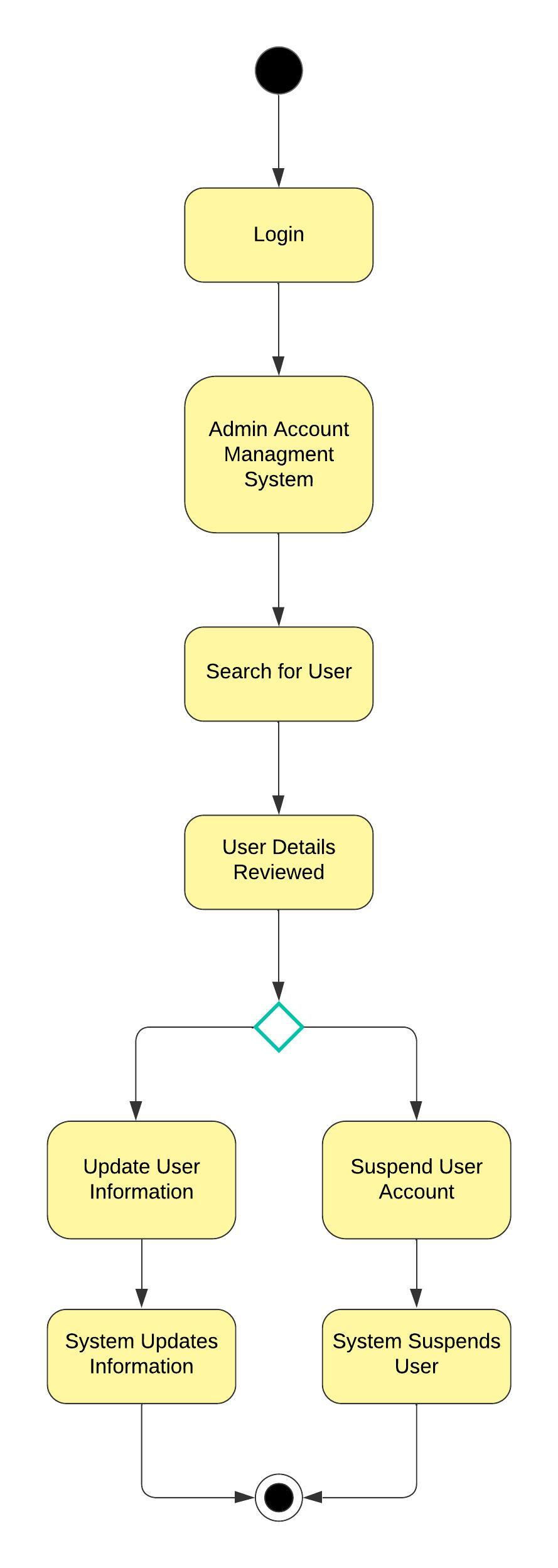
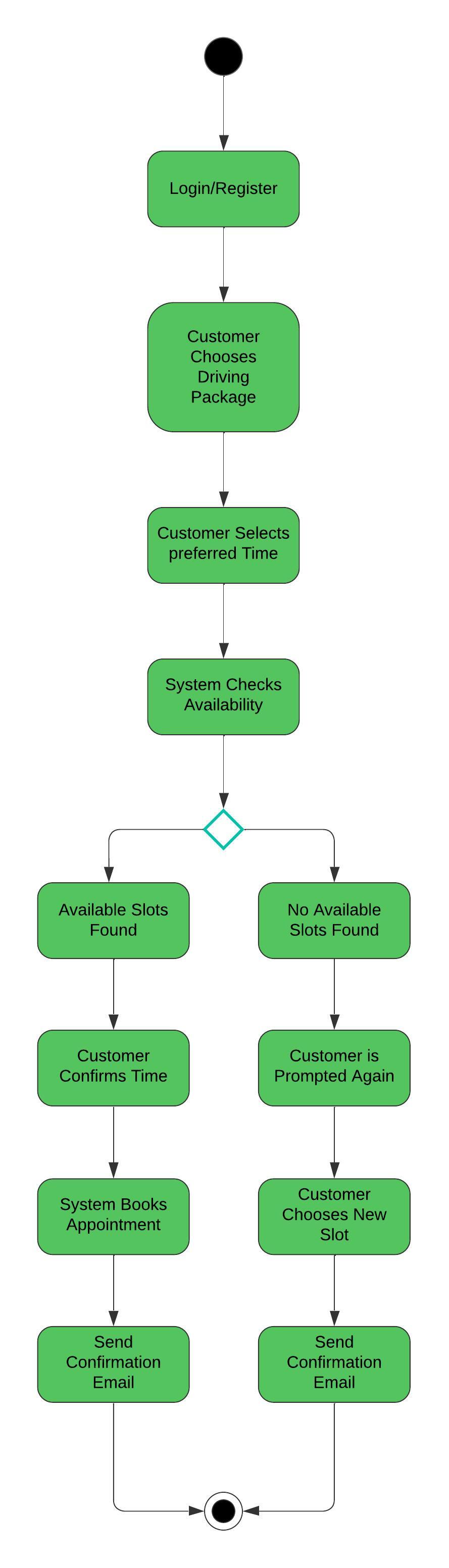
### UML Use Case Diagram

**

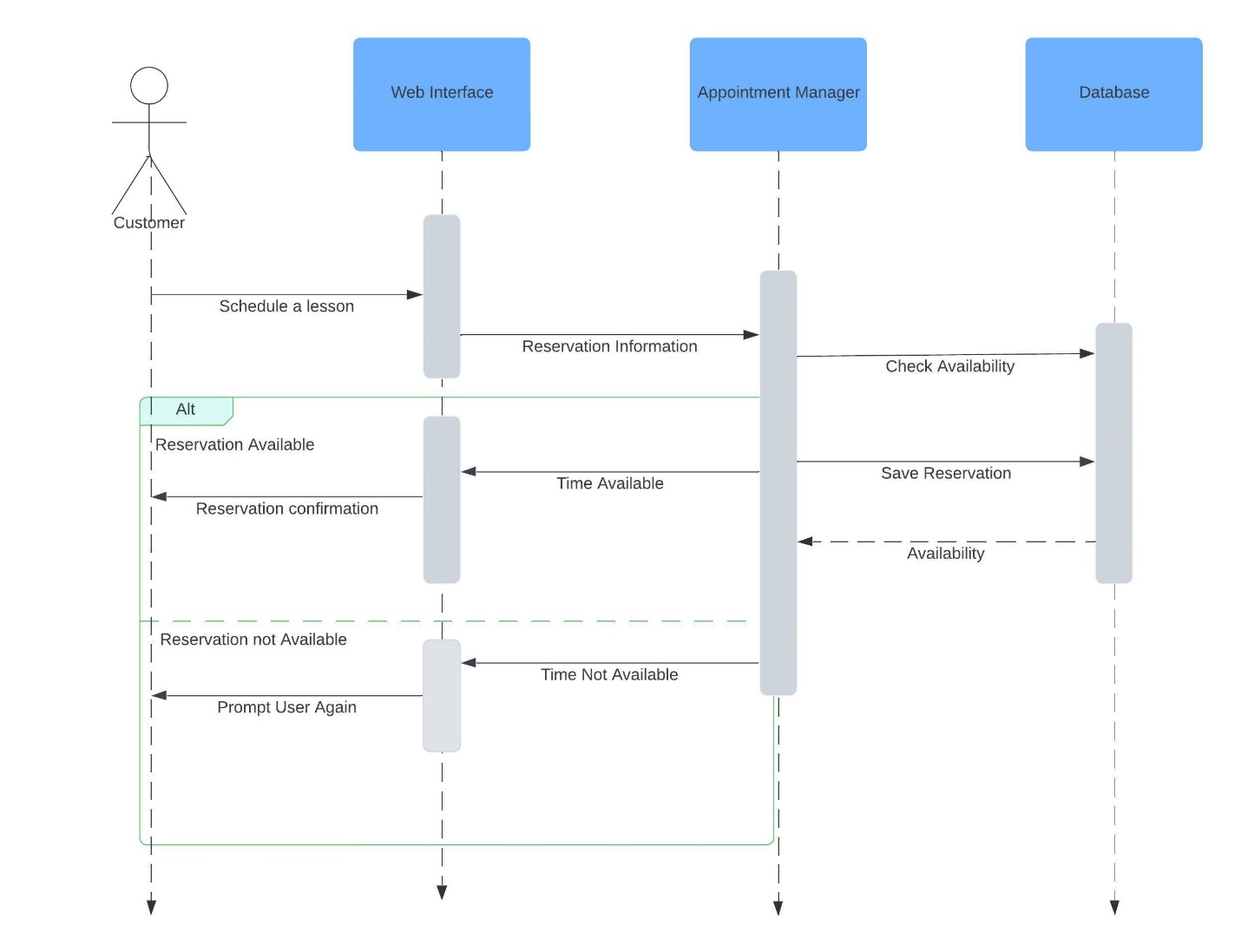
*A diagram of a driving lesson

Description automatically generated*

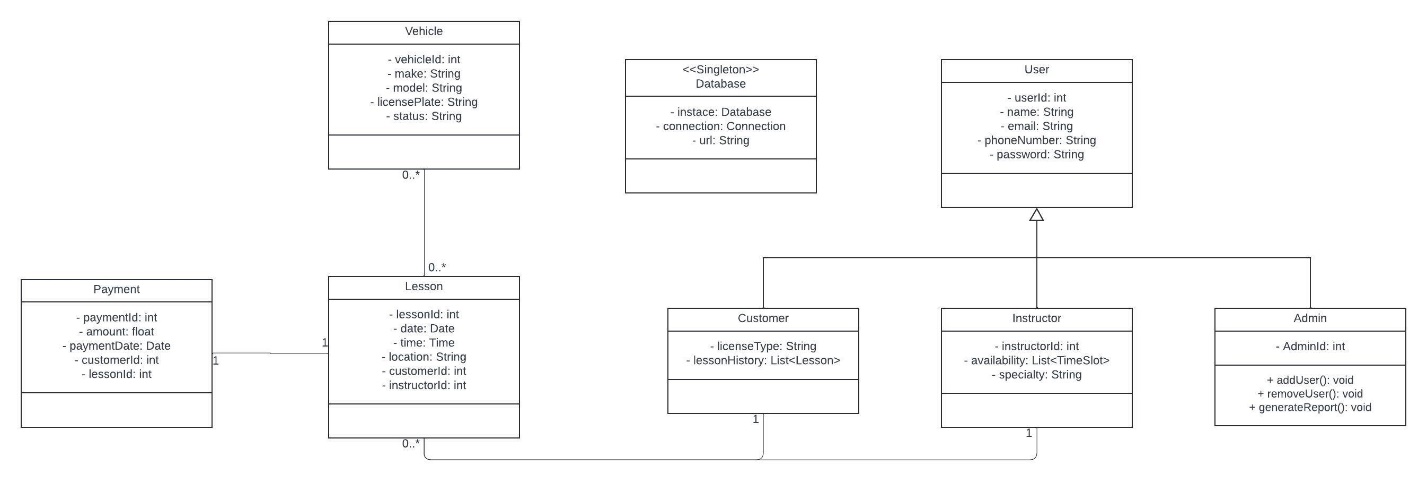
### UML Activity Diagrams

**

### UML Sequence Diagram

**

### UML Class Diagram

**

## Technical Requirements

The hardware requirements for the DriverPass system involve utilizing cloud-based servers, such as those provided by AWS, to host both the application and the database. Client devices include desktops, laptops, and mobile devices that support modern web browsers. Additionally, implementing a cloud-based backup infrastructure is essential to ensure data integrity and enable effective disaster recovery mechanisms.

Regarding software, the system should operate on a Linux (Ubuntu) or Windows Server operating system for the hosting environment. The web server can be Apache or Nginx, and PostgreSQL will be used as the database management system. The backend development will be conducted using programming languages such as Python, Java, or Node.js, while the frontend will leverage HTML, CSS, and JavaScript, with frameworks like React or Angular for user interfaces. The system should also integrate with cloud platforms such as AWS or Azure and incorporate security protocols like OAuth for authentication and email integration for password management and password resets.

For the development process, tools such as Jira will be used for task tracking, while Git will serve for version control. Communication among team members will be facilitated by Slack or Microsoft Teams, and UI/UX design will be managed using Figma. For database management, phpMyAdmin or MySQL Workbench will be helpful, and automated testing can be performed using tools such as Selenium or Postman to ensure the system’s functionality and reliability.

The infrastructure requirements focus on scalable cloud hosting services like AWS or Google Cloud to efficiently manage varying levels of traffic. A load balancer will be implemented to distribute traffic across multiple servers, and data synchronization tools will ensure consistency between online and offline environments. Additional infrastructure considerations include the use of a Content Delivery Network (CDN) to optimize content delivery, security measures such as firewalls and SSL certificates, and regular backups coupled with disaster recovery solutions to safeguard the system’s performance, security, and reliability.