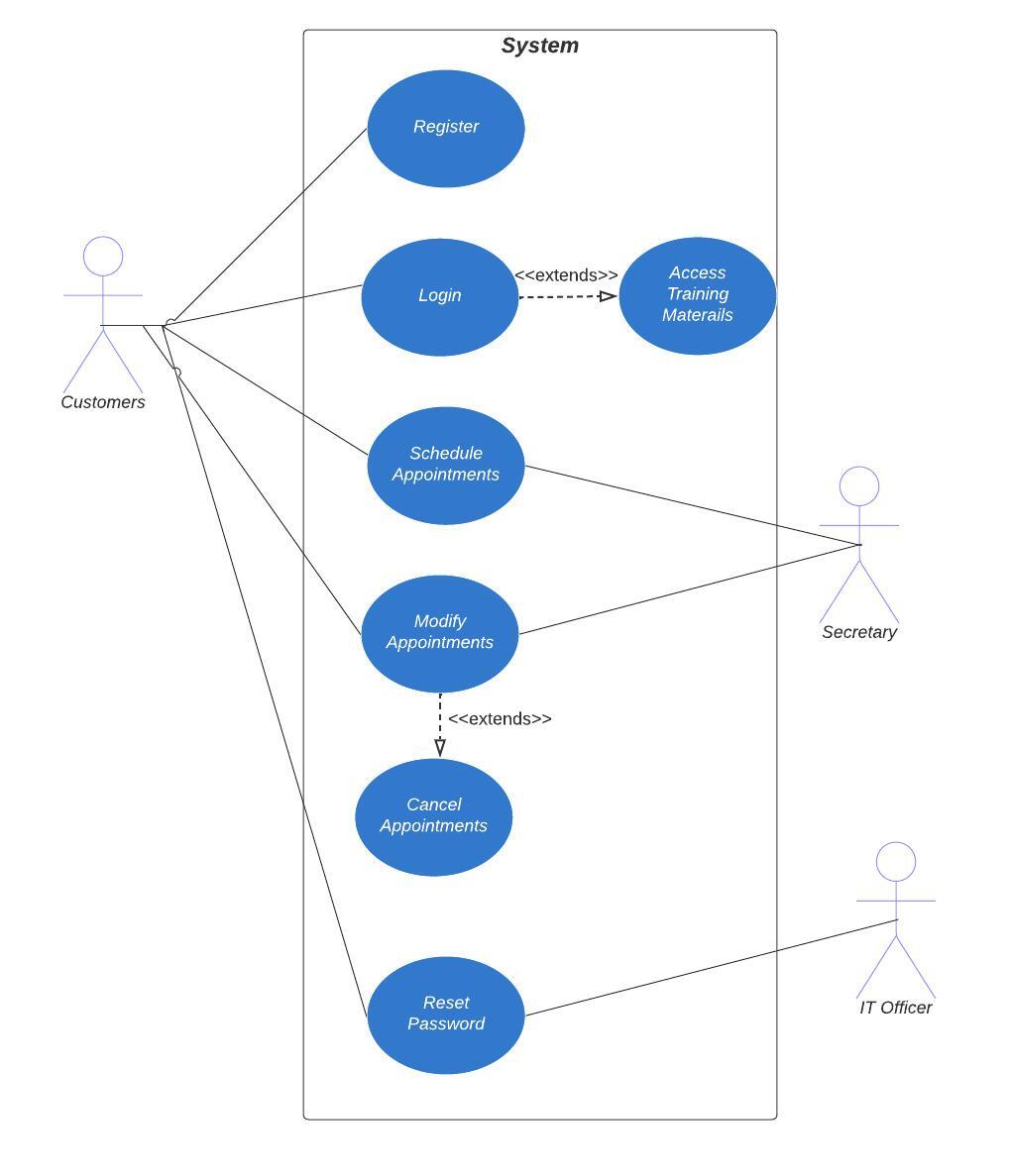
# CS 255 System Design Document

## UML Diagrams

### UML Use Case Diagram

**

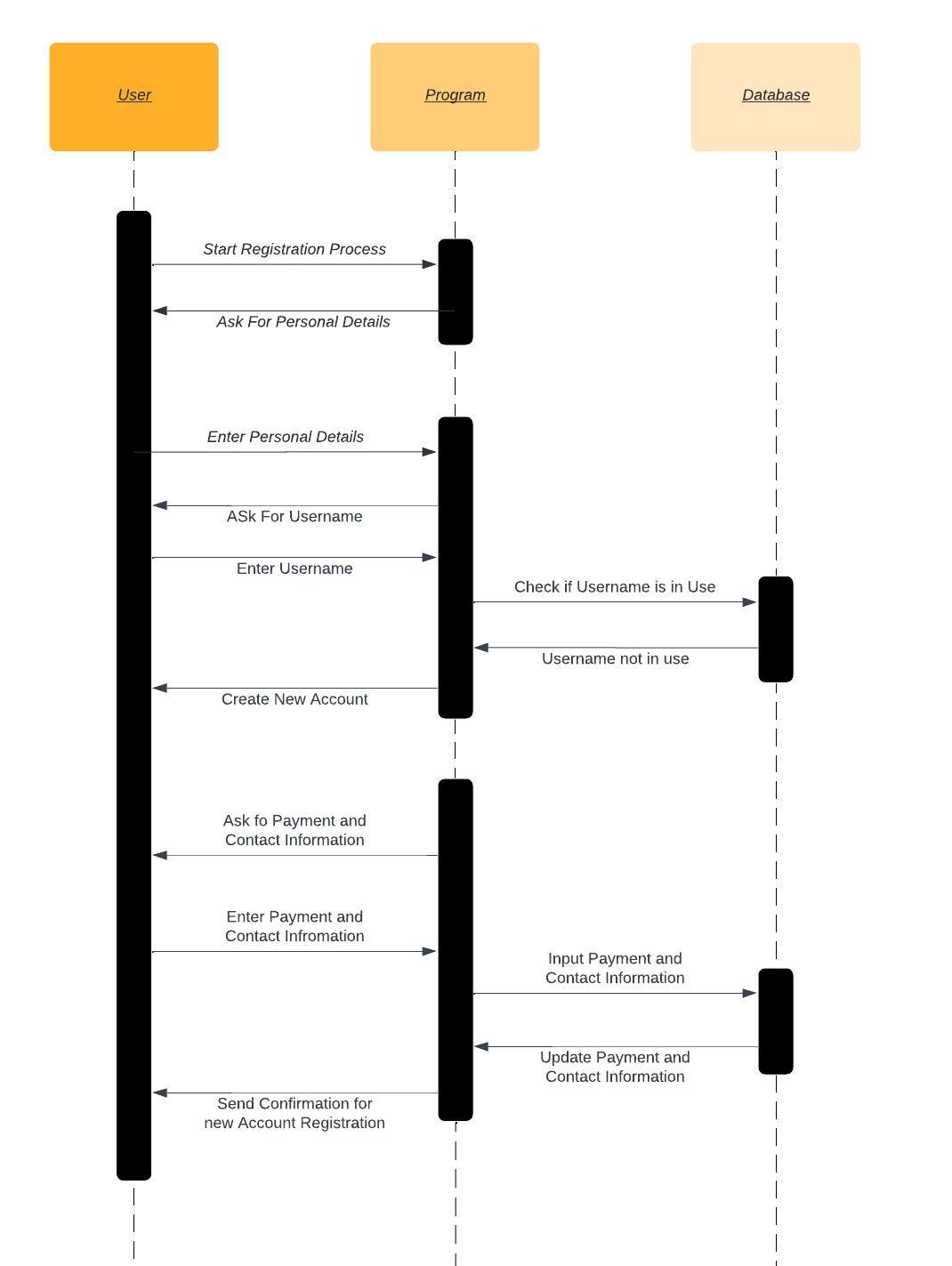
### UML Activity Diagrams

*A picture containing text, screenshot, diagram, receipt

Description automatically generatedA diagram of a process

Description automatically generated with low confidence*

### UML Sequence Diagram

**

### UML Class Diagram

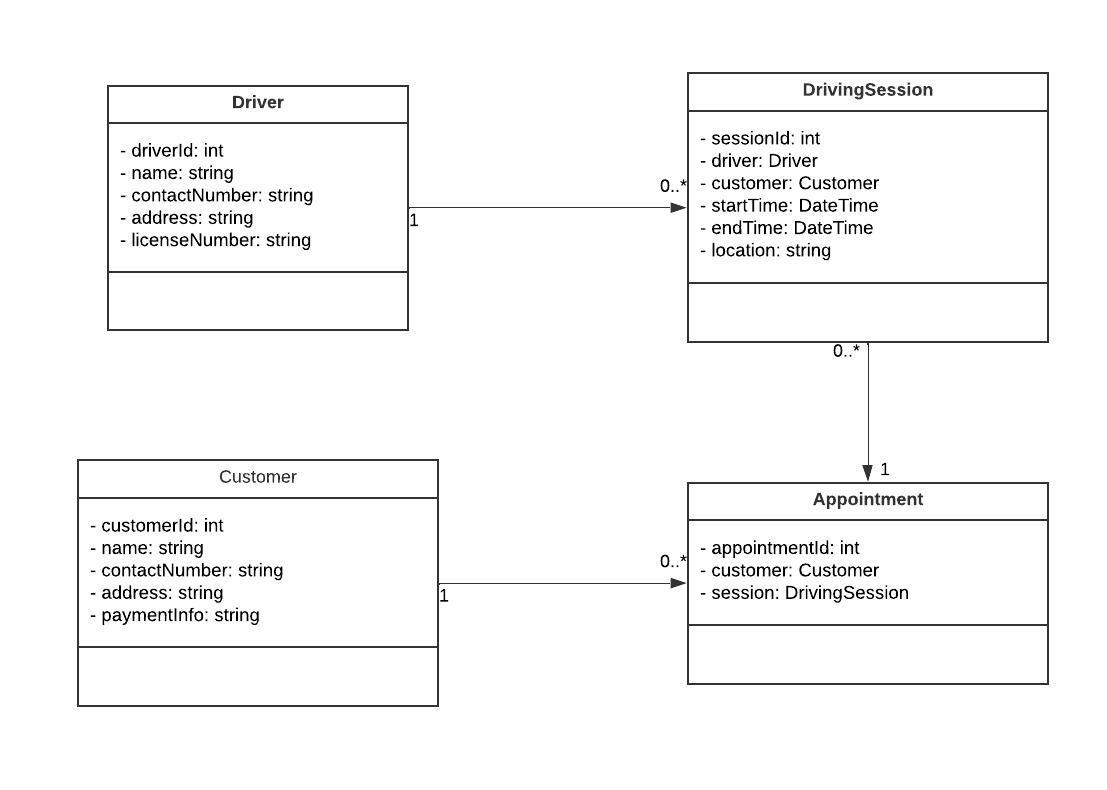
In this class diagram, we have four main classes: Driver, DrivingSession, Customer, and Appointment. Here's a brief description of each class:

Driver: Represents a driver involved in the driving sessions. It has attributes such as driverId, name, contactNumber, address, and licenseNumber.

DrivingSession: Represents a specific driving session that takes place between a driver and a customer. It has attributes such as sessionId, driver (reference to the Driver class), customer (reference to the Customer class), startTime, endTime, and location.

Customer: Represents a customer who schedules driving lessons. It has attributes such as customerId, name, contactNumber, address, and paymentInfo.

Appointment: Represents an appointment made by a customer for a driving session. It has attributes such as appointmentId, customer (reference to the Customer class), and session (reference to the DrivingSession class).



## Technical Requirements

Based on the diagrams and information provided, here are some technical requirements for the system design:

1. **Hardware Requirements:**

* Servers or cloud infrastructure to host the system and database.
* Sufficient computing resources to handle expected user traffic and data storage.
* Network infrastructure to ensure connectivity between different components and users.
* Client devices such as computers, laptops, tablets, and mobile devices for accessing the system.

2. **Software Requirements:**

* Operating system compatible with the chosen server infrastructure (e.g., Linux, Windows Server).
* Web server software (e.g., Apache, Nginx) to serve web pages and handle HTTP requests.
* Database management system (e.g., MySQL, PostgreSQL) to store and manage data.
* Programming languages and frameworks for developing the system components (e.g., Java, Python, PHP, .NET).
* Development tools and integrated development environments (IDEs) for coding, debugging, and testing (e.g., Eclipse, Visual Studio, PyCharm).

3. **Infrastructure and Services:**

* Internet connectivity to ensure accessibility of the system from anywhere.
* Cloud services for hosting the system, providing scalability, and managing infrastructure (e.g., Amazon Web Services, Microsoft Azure, Google Cloud Platform).
* Security measures, including firewalls, encryption, and secure socket layer (SSL) certificates, to protect sensitive user data and ensure secure communication.
* Regular backups and disaster recovery plans to safeguard data and ensure business continuity.

4. **Integration Requirements:**

* Integration with external systems or APIs for accessing DMV updates, rules, policies, and sample questions.
* Integration with payment gateway services to securely process credit card transactions.
* Integration with email or notification services for sending appointment confirmations and notifications.

5. **User Interface and Experience:**

* Responsive web design to ensure the system is accessible and user-friendly across different devices and screen sizes.
* Compatibility with popular web browsers (e.g., Chrome, Firefox, Safari) to ensure consistent user experience.
* Consideration for accessibility standards to accommodate users with disabilities.