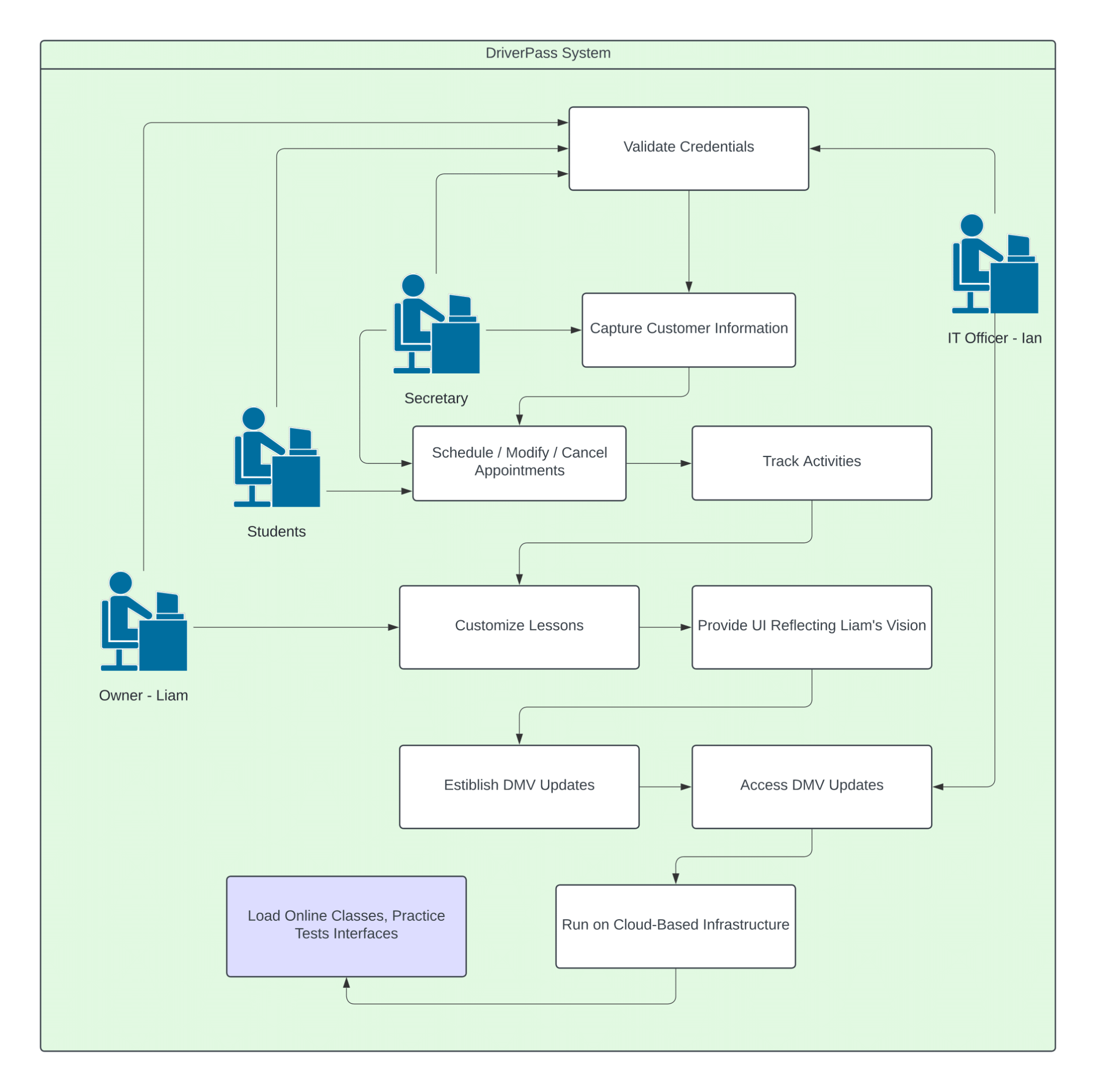
# CS 255 System Design Document

Joseph Dengler

## UML Diagrams

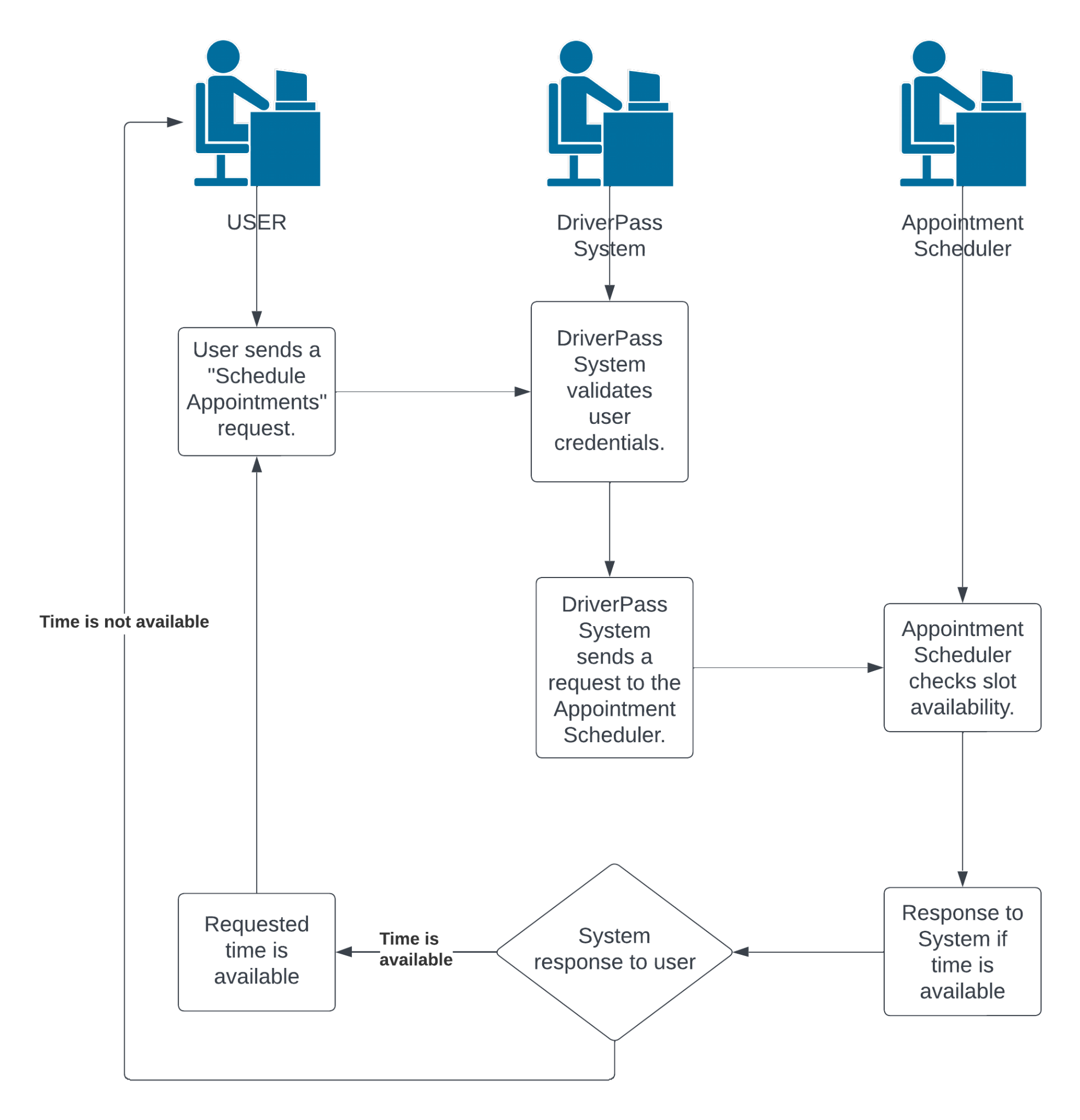
### UML Use Case Diagram



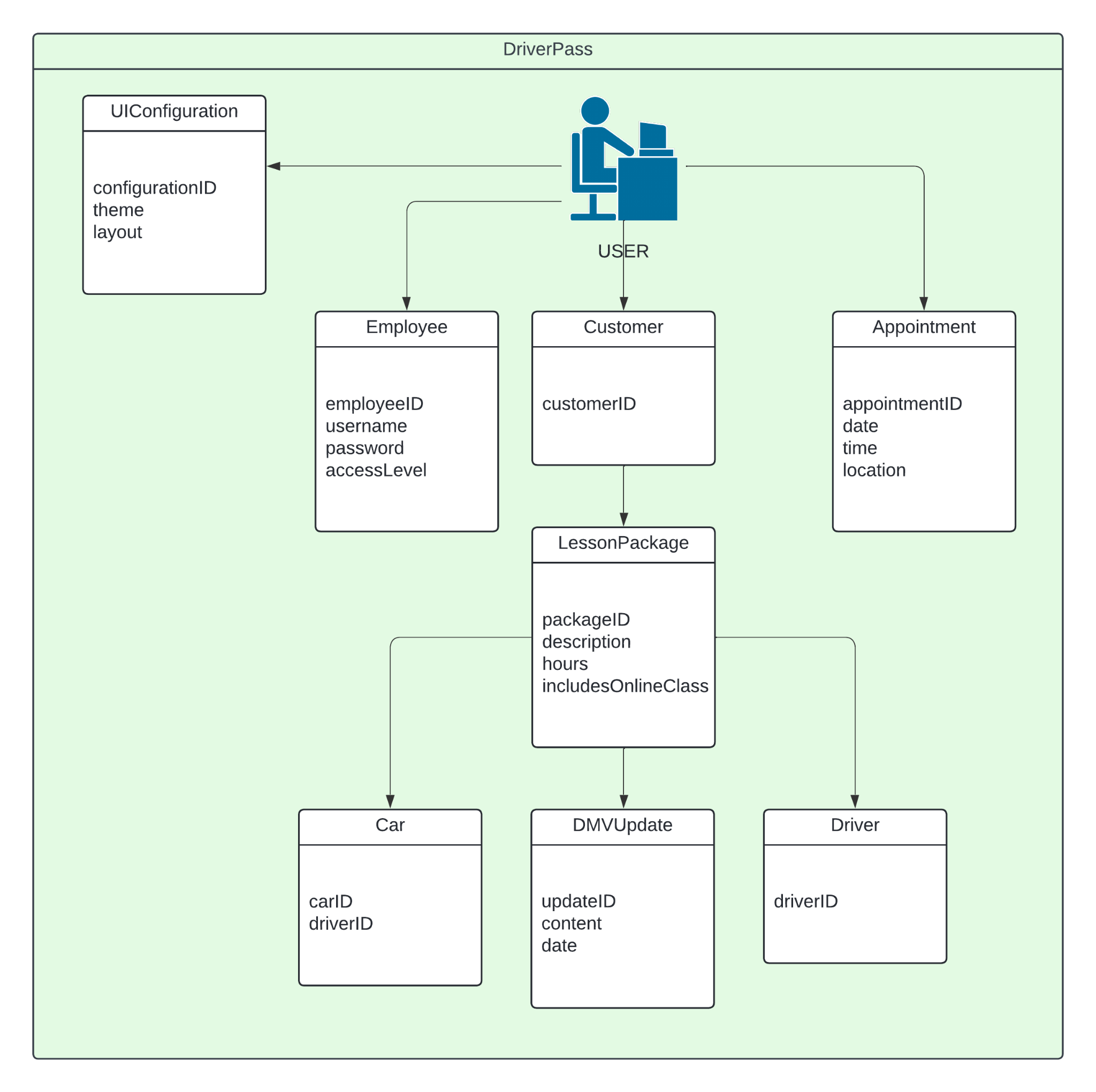
### UML Activity Diagram

### 

### UML Sequence Diagram

**

### UML Class Diagram

**

## Technical Requirements

**Hardware Requirements:**

Servers: The system needs robust servers capable of handling both online and offline data access, ensuring data integrity and avoiding duplication.

Database Server: A dedicated server for storing and managing the database, ensuring efficient data retrieval and storage.

Cloud Infrastructure: The system should be designed to run on a cloud-based infrastructure, eliminating the need for local server management.

**Software Requirements:**

Web Application: A web-based application to allow users (customers, employees) to access the system online. The application should be compatible with major web browsers.

Database Management System (DBMS): A reliable DBMS for managing the database, ensuring data consistency, security, and ease of retrieval.

Security Software: Robust security measures, including encryption, firewalls, and access controls, to protect user data and system integrity.

Backup and Recovery Software: Automated backup and recovery tools to prevent data loss and ensure system availability.

Programming Languages: Depending on the technology stack, programming languages like Java, Python, or others may be required for application development.

**Tools:**

CASE Tools: The system design and development can benefit from CASE (Computer-Aided Software Engineering) tools for creating and maintaining UML diagrams. Lucidchart has already been used so far.

Version Control: Version control tools (e.g., Git) to manage source code and track changes made by the development team.

Issue Tracking: An issue tracking system (e.g., Jira) to manage and prioritize development tasks and address any reported issues.

**Infrastructure:**

Internet Connectivity: A reliable and high-speed internet connection for seamless online access to the system.

DMV Connectivity: Integration with the DMV systems, requiring a secure and stable connection to receive updates on rules, policies, and sample questions.

Cloud Service Provider: Selection of a reputable cloud service provider (e.g., AWS, Azure) for hosting the application and ensuring scalability and reliability.