# Christopher Mooers

# CS 255 System Design Document

# 10/18/2025

## UML Diagrams

### UML Use Case Diagram

*A diagram of a driverpass system

AI-generated content may be incorrect.*

**Description**: The UML use case diagram for the DriverPass system shows how different users interact with the system to perform key functions. The primary users include customers, trainers, the IT officer, the secretary, the DMV, and the business owner. Customers can register for accounts, log in, purchase packages, schedule or cancel appointments, and access online training materials. Trainers can view assigned lessons and record progress, while the IT officer manages accounts and resets passwords. The DMV updates testing policies, and the owner can view reports and overall system activity. This diagram provides an overview of how all user roles connect and contribute to the DriverPass service.

### UML Activity Diagrams

A screenshot of a computer

AI-generated content may be incorrect.

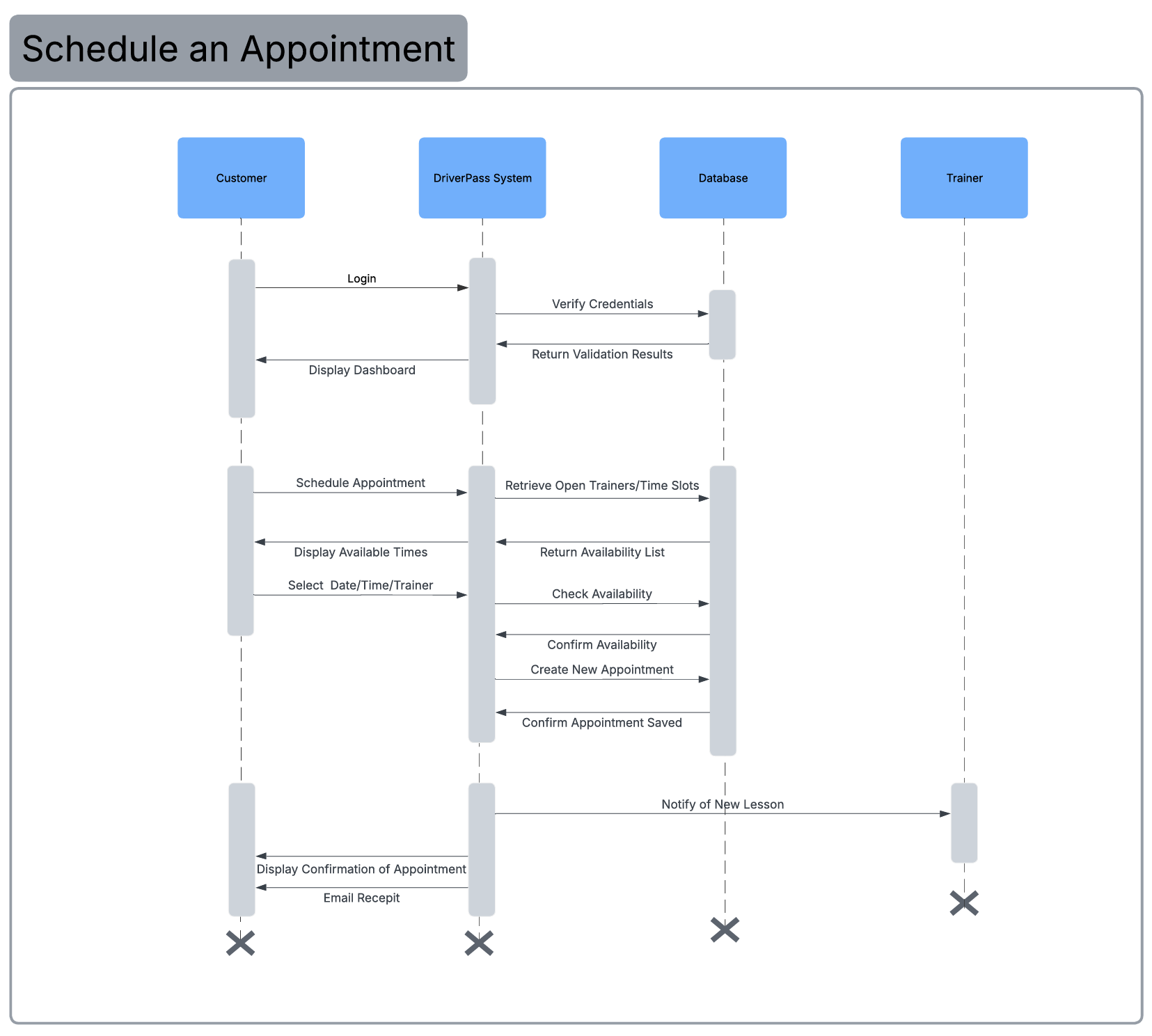
**Description**: This activity diagram illustrates how customers and the DriverPass system work together to manage driving lesson appointments. The process begins when a customer logs in and selects whether to schedule, modify, or cancel an appointment. The system verifies login credentials, displays available lesson times and instructors, and checks availability before confirming changes. If a selected time slot is unavailable, the customer is prompted to choose another option. Once the appointment is finalized, the system updates the database and sends confirmation messages to both the customer and the assigned instructor. This workflow ensures a smooth scheduling experience while keeping all lesson information accurate and up to date.

A screenshot of a computer

AI-generated content may be incorrect.

**Description:** This activity diagram outlines the steps a customer follows to purchase a training package through the DriverPass system. After logging in, the customer reviews available packages, selects one, and proceeds to payment. The system validates login credentials, displays pricing details, and securely communicates with the payment gateway for authorization. If payment is approved, the DriverPass system activates the customer’s package, updates account information, and emails a confirmation receipt. In the event of a declined payment, the customer can retry, change methods, or cancel the transaction. This process ensures secure and efficient payment handling while immediately granting customers access to their training materials.

### UML Sequence Diagram



**Description:** This sequence diagram shows the step-by-step interaction between the customer, DriverPass system, database, and trainer during the appointment scheduling process. The customer begins by logging in, and the system verifies the credentials with the database. Once validated, the customer selects the option to schedule an appointment. The system retrieves available trainers and time slots from the database and displays them to the customer. After the customer selects a preferred date, time, and trainer, the system checks availability, creates a new appointment record, and confirms that it has been saved. Finally, the system notifies the assigned trainer and provides the customer with an on-screen confirmation and an email receipt. This diagram demonstrates how the DriverPass system coordinates real-time communication between users and the database to ensure accurate and efficient scheduling.

### UML Class Diagram

A computer screen shot of a diagram

AI-generated content may be incorrect.

**Description:** The UML class diagram shows the primary classes and their relationships within the DriverPass system. At the core, an abstract User class defines shared attributes and operations, while specialized subclasses, such as Customer, Trainer, IT Officer, and Owner, extend these functions to meet their specific roles. Customers can purchase packages, schedule or cancel appointments, and complete practice exams. Trainers conduct appointments, record lesson notes, and track customer progress. Appointments are associated with customers, trainers, and vehicles, with related LessonNotes created for each session. Payments are authorized through a secure PaymentGateway, while customers enroll in Packages managed through PackageEnrollments. The Owner oversees reports and monitors system performance, and IT Officers manage accounts and system configurations. A DMVUpdate class allows administrators to publish policy and rule changes to ensure content accuracy. Collectively, these classes establish the structure and behavior needed to support DriverPass’s scheduling, training, and administrative operations.

## Technical Requirements

The DriverPass system will be built as a web-based application that can be accessed through a secure internet connection on any modern device such as a desktop, laptop, tablet, or smartphone. The goal is to create a system that is reliable, easy to use, and scalable as the number of users grows.

**Hardware Requirements:**  
DriverPass will be hosted on cloud-based servers that provide dependable uptime and data storage. The servers should have enough memory and processing power to support multiple users logging in, scheduling appointments, and accessing course materials at the same time. End users will only need an internet connection and a standard web browser.

**Software Requirements:**  
The application will use common web technologies such as HTML, CSS, and JavaScript for the front end, and a programming language like Java for the back end. A relational database,such as MySQL, will be used to store user accounts, appointments, packages, and lesson information. The system will also include a payment component connected to a secure third-party service for processing transactions.

**Development Tools and Infrastructure:**  
Developers can use familiar tools such as Visual Studio Code or IntelliJ IDEA for coding, GitHub for version control, and Lucidchart for UML design. The system will follow a three-tier architecture that separates the user interface, business logic, and data layers to make the system easier to maintain and update. Security will include password encryption, HTTPS connections, and role-based permissions to protect user data.

By combining these technologies and design practices, DriverPass will run efficiently, support all user roles, and ensure data security and system reliability for both customers and administrators.