**CS 255 System Design**

Edgar I. Pena

Southern New Hampshire University

CS-255: System Analysis and Design

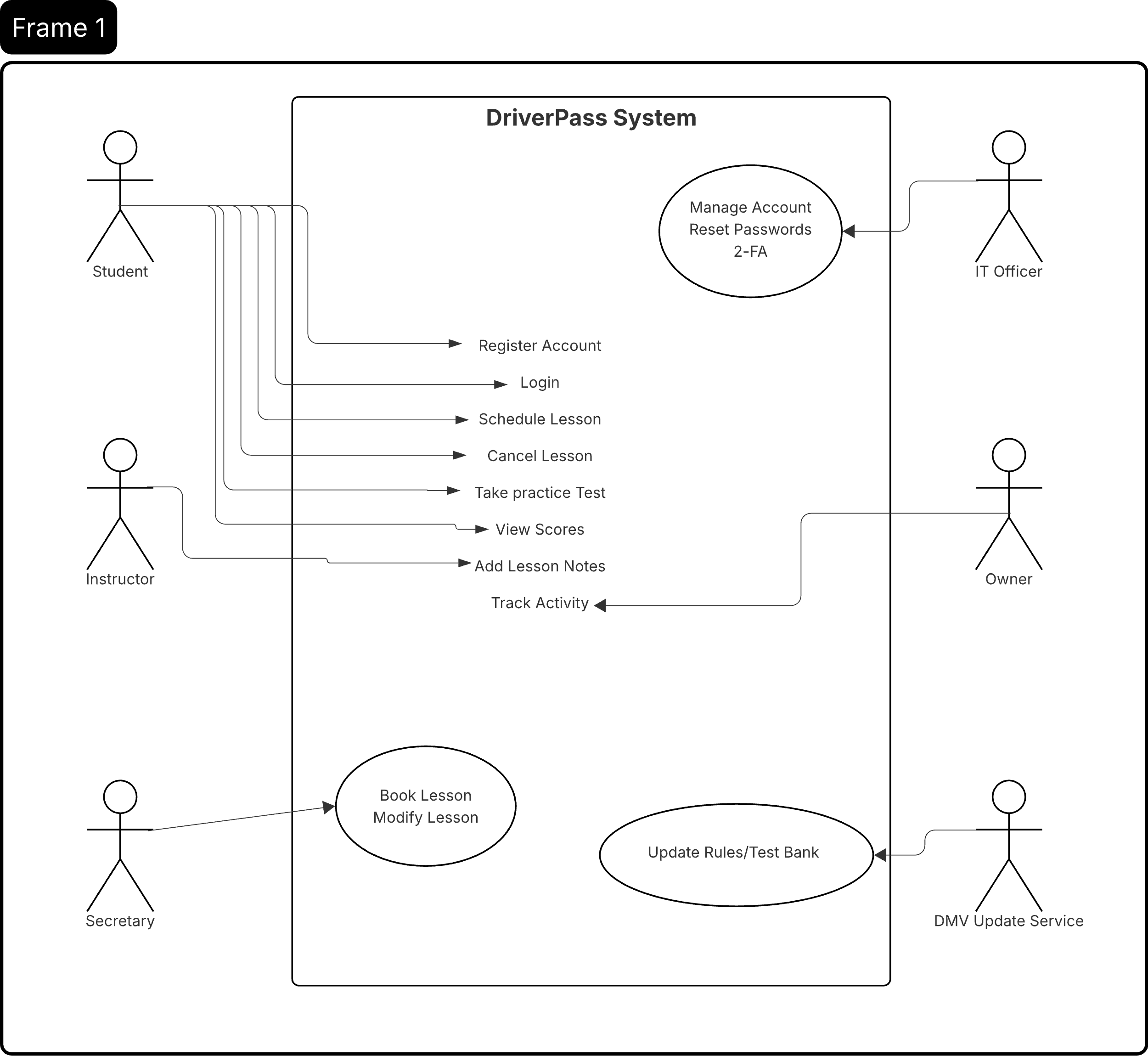
Professor Antonio Fredericks

August 20, 2025

# CS 255 System Design

## UML Diagrams

### UML Use Case Diagram



This use case diagram outlines the primary interactions between system actors and the DriverPass platform. Students are supported with account creation, authentication, lesson scheduling and cancellation, practice testing, and score retrieval. Instructors contribute lesson notes tied to appointments. The secretary can create and update lesson bookings on behalf of students. The IT officer manages account security functions, including password resets and two-factor authentication. The owner has access to system activity tracking, while the DMV update service provides new rules and test updates to keep practice exams aligned with current standards.

### UML Activity Diagrams

A diagram of a driver pass system

Description automatically generated

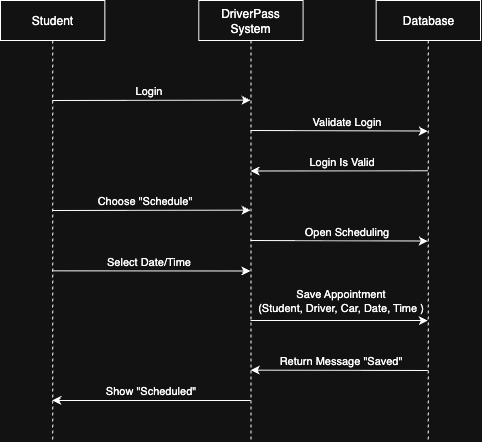
This activity diagram illustrates how lesson appointments are managed in the DriverPass system. After authentication, the student can choose to schedule, modify, or cancel a lesson. Scheduling involves selecting a lesson date and time, which the system validates and persists to the database. Modifying retrieves an existing appointment, allows updates to the driver, car, or time, and commits changes to the record. Cancellation prompts confirmation and then marks the appointment as canceled in the system. All operations ensure the database is updated consistently with student, driver, car, and time details.

A diagram of a student

Description automatically generated

This activity diagram shows the workflow for a student completing a practice test. After login validation, the student selects a test, and the system loads the corresponding questions. As the student answers, responses are recorded, and once complete, the system grades the test. The final score is stored in the database, and results are displayed back to the student.

### UML Sequence Diagram



This sequence diagram shows the process of scheduling a lesson. The student logs in, and the system validates the login with the database. After selecting the scheduling option and choosing a date and time, the system saves the appointment details, such as the student, driver, car, date, and time, to the database. Once stored, the system returns a confirmation message, and the student sees the lesson marked as scheduled.

### UML Class Diagram

A screenshot of a computer

Description automatically generated

This class diagram defines the main entities of the DriverPass system and their relationships. User accounts are the foundation, supporting roles such as student, instructor, secretary, IT officer, and owner. Students are linked to training packages and can schedule appointments with instructors and cars. Each appointment may generate a lesson note and is tracked with status values like scheduled, modified, or canceled. Practice tests are managed separately, with test attempts tied to students and updated by DMV rule changes. Finally, audit logs capture actions across the system to provide traceability of account and scheduling activity.

## Technical Requirements

The DriverPass system will need a reliable hardware and software setup to support its users and features. On the hardware side, the system should run on a web server with at least mid-range processing power, enough memory to handle multiple users at once, and secure storage for account information, lesson data, and test results. Since the system is used online, internet access with strong uptime guarantees will also be necessary.

On the software side, a web application framework will be required to support the user interface and backend logic. The system will also need a relational database to store structured data such as user accounts, appointments, lesson notes, and practice test results. For security, the software should include account management features such as password hashing and optional two-factor authentication.

In terms of tools, UML modeling tools like Lucidchart are useful during design, while development will require an IDE and version control system such as Git. Automated testing tools should be in place to confirm that scheduling, account management, and test scoring all work as expected.

For infrastructure, the application should be hosted on a secure cloud service or an on-premises server with proper backup and recovery processes. Secure communication protocols such as HTTPS are required to protect user data in transit. Regular updates from the DMV must also be supported, which means the system should provide integration points to load new rules and update the practice test bank.