# Dylan Drake

CS 255

August 11th, 2025

# CS 255 System Design Document

## UML Diagrams

### UML Use Case Diagram

**Description:**

* Actors: Student, Driving Instructor, Admin
* Use cases: Create Account, Schedule Lessons, Take Practice Tests, View Progress, Manage Accounts, Upload Training Materials

A diagram of a diagram

AI-generated content may be incorrect.

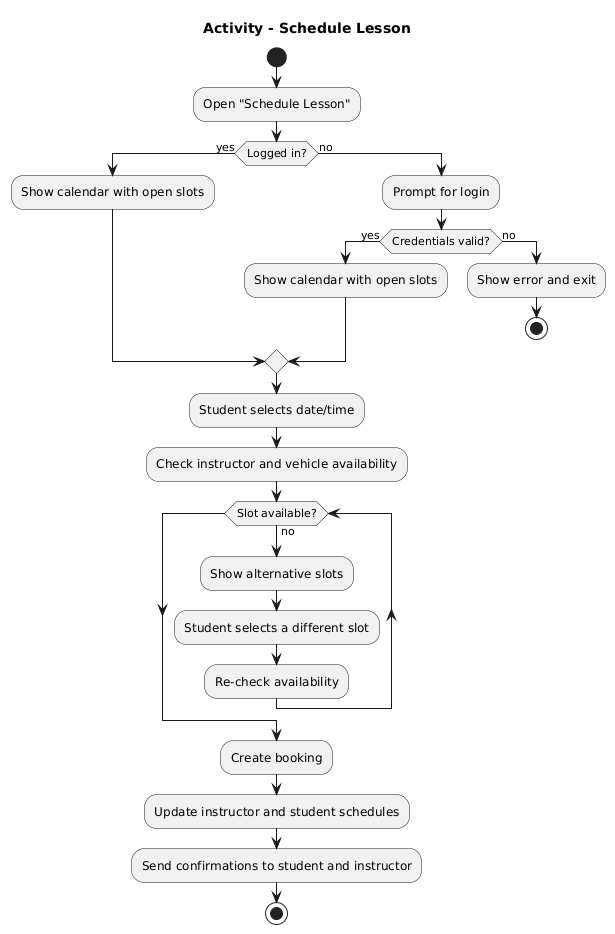
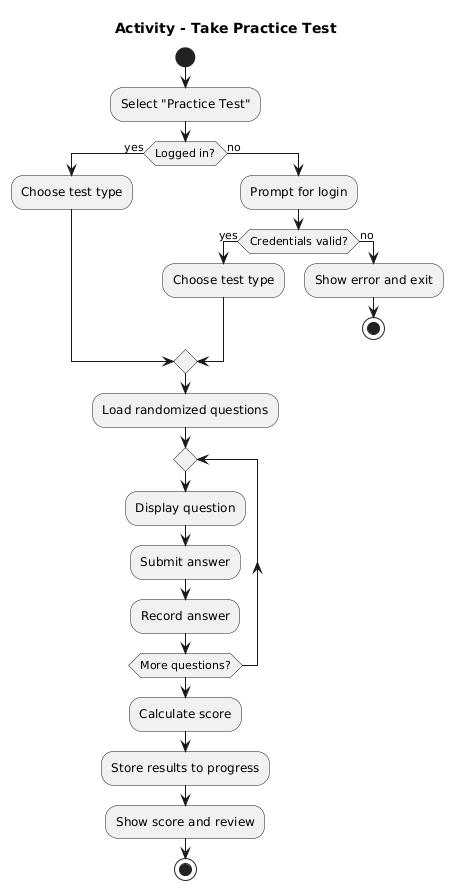
### UML Activity Diagrams

**Activity Diagram 1: Schedule Lessons**

1. Student logs in
2. System verifies account
3. Student selects “Schedule Lesson”
4. Calendar displays available times
5. Student selects date/time
6. System confirms booking and updates schedule

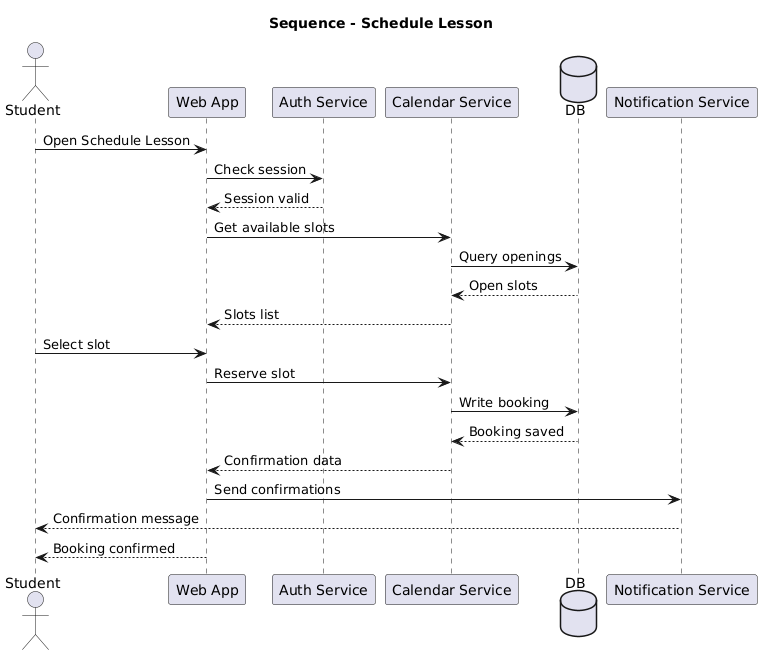
**Activity Diagram 2: Take Practice Test**

1. Student logs in
2. System verifies account
3. Student selects “Take Practice Test”
4. System loads randomized test questions
5. Student completes test
6. System grades test and displays results



### UML Sequence Diagram

* Student → System: Login Request
* System → Database: Verify Credentials
* Student → System: Request Lesson Scheduling
* System → Calendar Service: Retrieve Available Slots
* Calendar Service → System: Return Slots
* Student → System: Select Slot
* System → Database: Save Booking
* System → Student: Confirm Lesson Scheduled



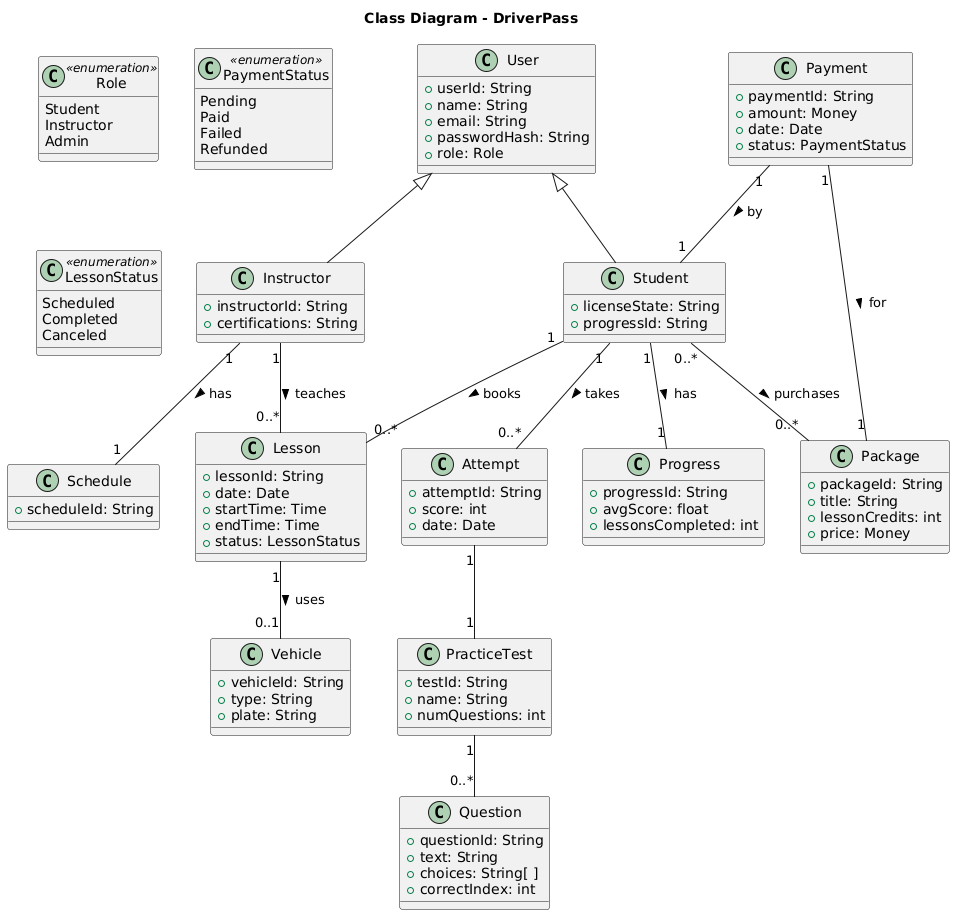
### UML Class Diagram

Classes:

* **Student** (Attributes: studentID, name, email, password, progressReport)
* **Instructor** (Attributes: instructorID, name, schedule, certifications)
* **Lesson** (Attributes: lessonID, date, time, instructorID, studentID)
* **PracticeTest** (Attributes: testID, questions, score)
* **Admin** (Attributes: adminID, name, role)

Associations:

* Student schedules Lesson (1..\* relationship)
* Instructor teaches Lesson (1..\* relationship)
* Student takes PracticeTest (1..\* relationship)



## Technical Requirements

**Hardware:**

* Web server with 16GB RAM, quad-core processor
* User devices: desktop, laptop, or mobile with internet access

**Software:**

* Server OS: Windows Server or Linux
* Database: MySQL
* Web Framework: Java Spring Boot
* Browser support: Chrome, Firefox, Edge, Safari

**Tools:**

* Lucidchart for diagramming
* Eclipse or IntelliJ for Java development
* GitHub for version control

**Infrastructure:**

* Cloud hosting (AWS or Azure)
* Secure HTTPS with SSL certificate
* Daily data backups