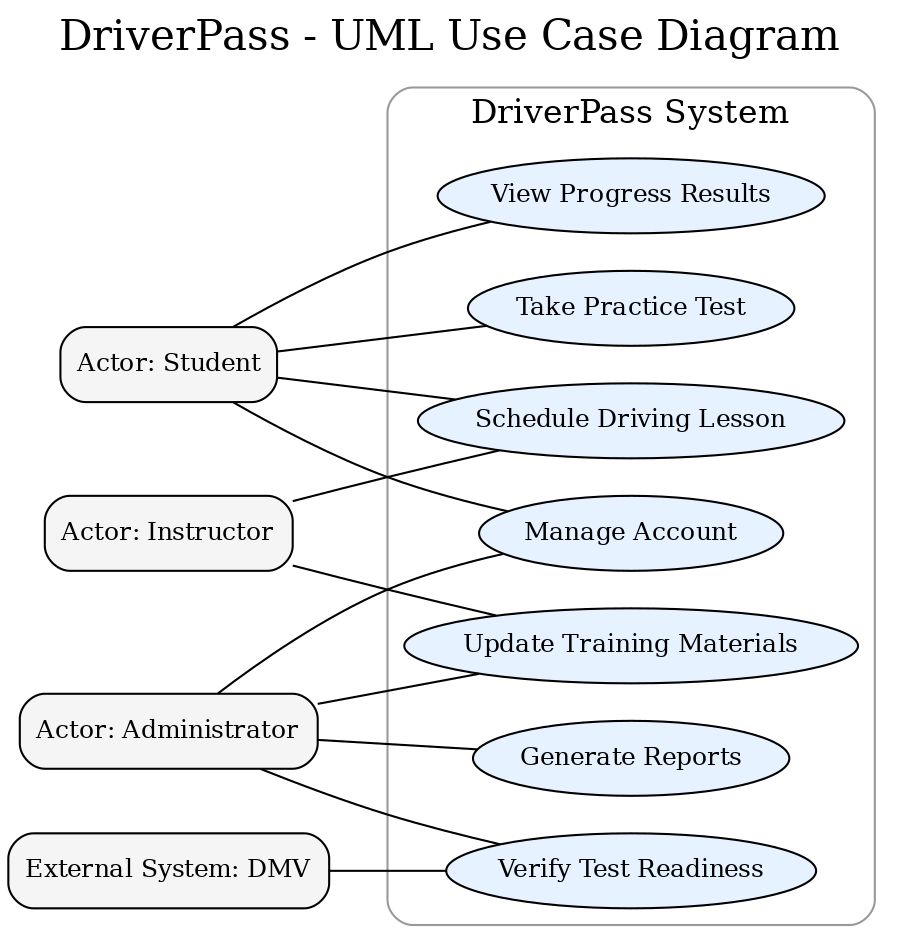
# CS 255 System Design Document Template

This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead the goal is to complete each section based on what your client’s needs are. Remove this note when you are finished, and replace all bracketed text with the relevant information.

## UML Diagrams

### UML Use Case Diagram

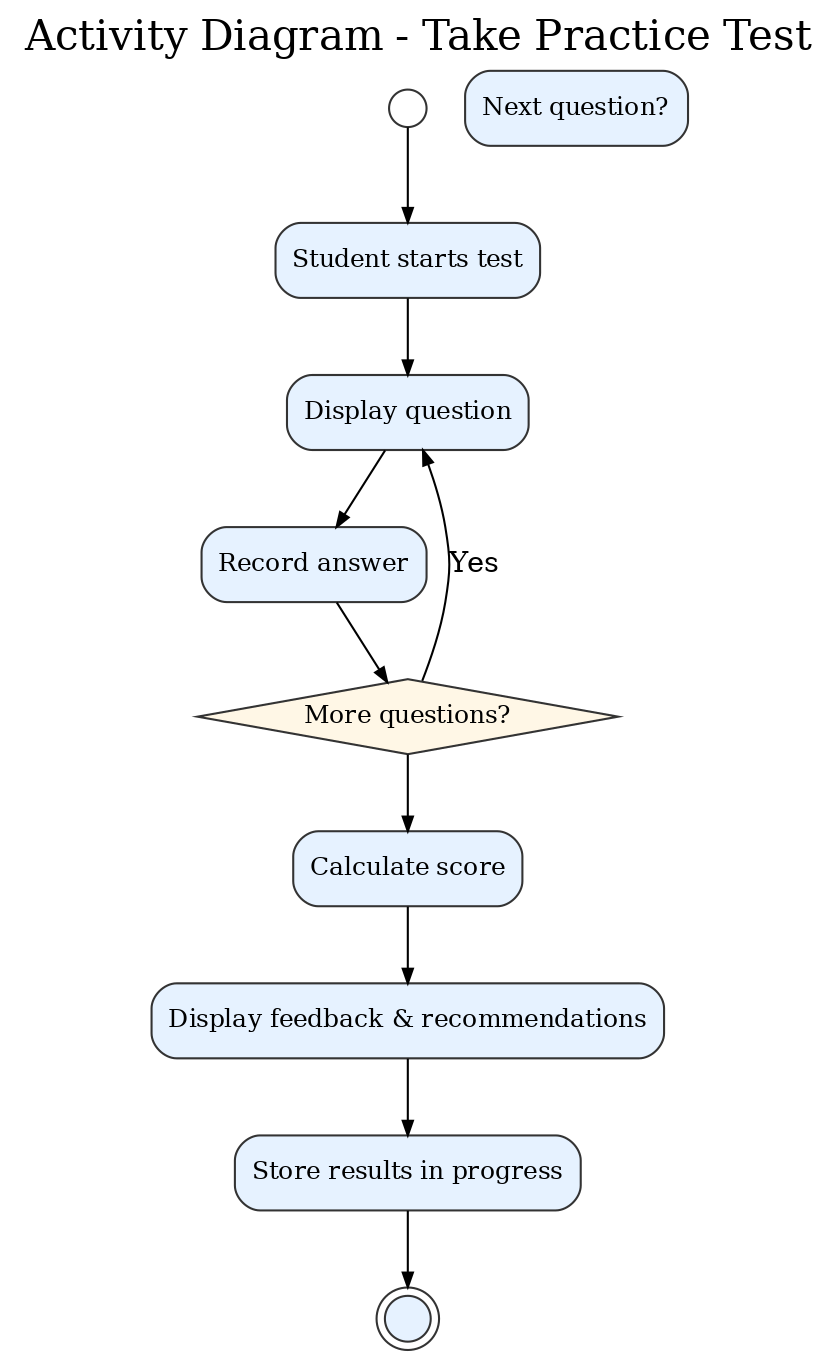
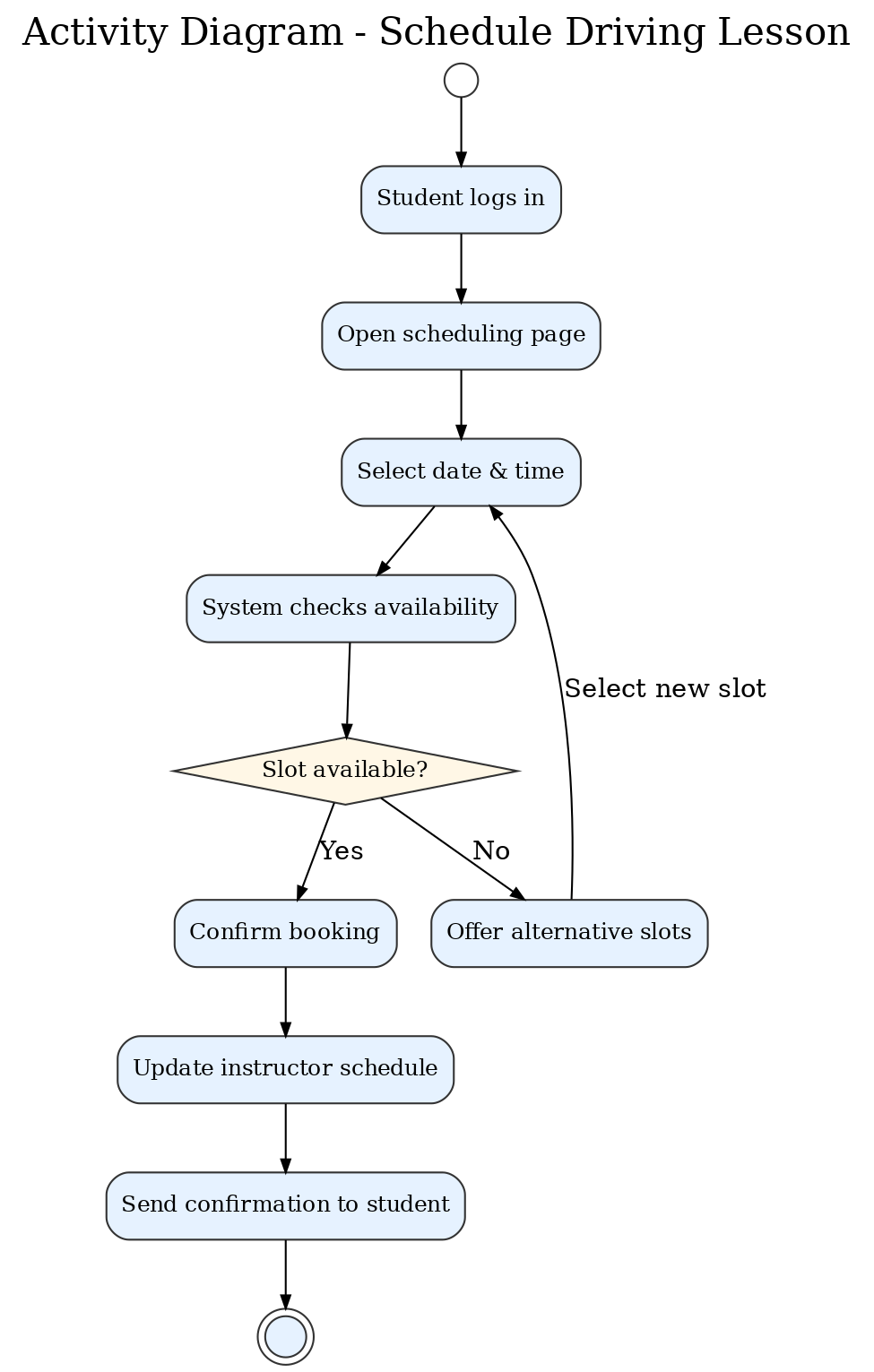
The use case diagram for the DriverPass system illustrates the primary functions available to each type of user. The main actors in the system are the Student, Instructor, Administrator, and the DMV System. The Student can take practice tests, schedule driving lessons, view progress reports, and manage their account. Instructors can manage their schedules, update lesson availability, and provide feedback to students. The Administrator can update training materials, generate system reports, and manage user accounts. The DMV System is connected for verification of test readiness.  
 This diagram ensures that all necessary user interactions are clearly defined so developers can implement features that meet the client’s needs while keeping the system intuitive for end users.



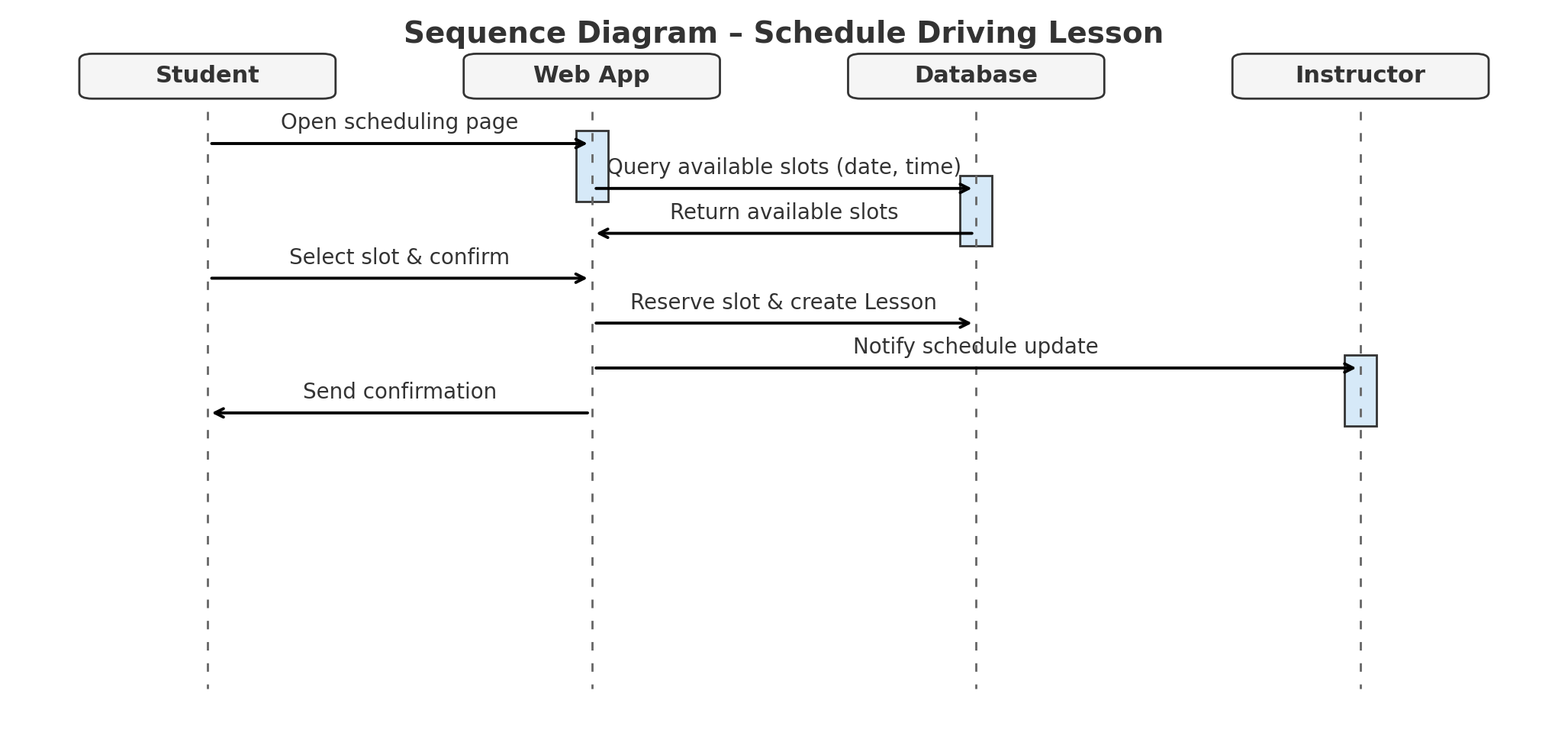
### UML Activity Diagrams

Two activity diagrams were created to show the step-by-step flow of specific processes.

1. **Schedule Driving Lesson**  
    This diagram shows how a student selects a lesson date, checks for availability, confirms the booking, and receives a confirmation. It includes decision points for slot availability and payment requirements (if applicable).
2. **Take Practice Test**  
    This diagram outlines how a student begins a test, navigates through questions, submits answers, and receives immediate feedback. It also includes the decision point where the system determines whether the student passes or fails.

These diagrams help visualize each process so both technical and nontechnical stakeholders can understand the exact steps the system will handle.  


### UML Sequence Diagram

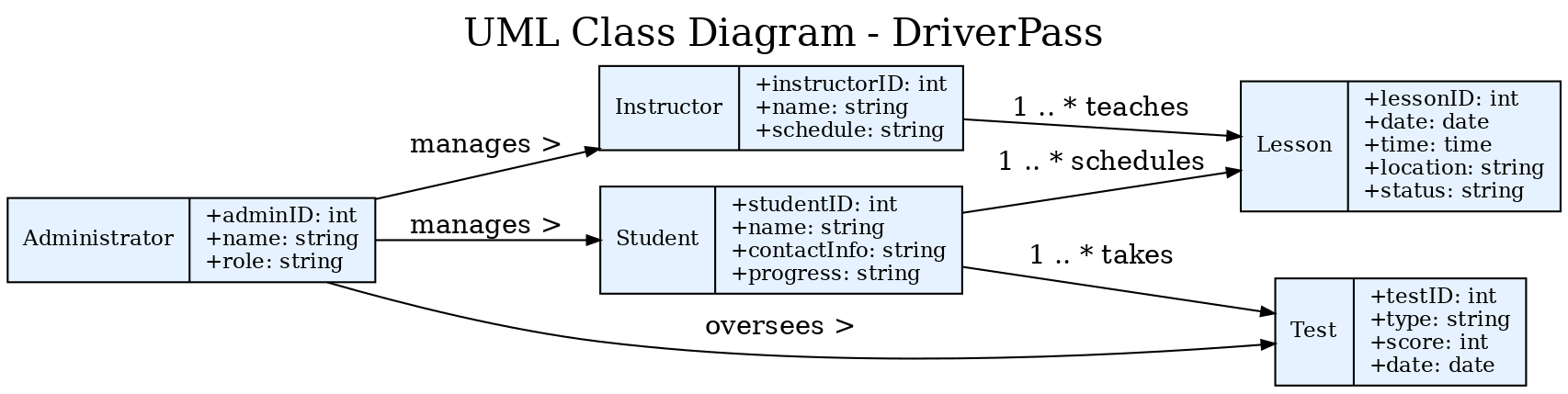
The sequence diagram for the *Schedule Driving Lesson* use case shows the order of interactions between the Student, Scheduling System, Instructor, and Database. It begins with the student requesting available lesson times, continues with the system checking the database for openings, and ends with the instructor’s schedule being updated and the student receiving confirmation.  
 This diagram is valuable for developers because it defines the exact communication flow between components, ensuring a smooth, predictable user experience.  


### UML Class Diagram

The class diagram for DriverPass defines the structure of the system by identifying the key classes and their attributes:

* **Student**: studentID, name, contactInfo, progress
* **Instructor**: instructorID, name, schedule
* **Lesson**: lessonID, date, time, location, status
* **Test**: testID, type, score, date
* **Administrator**: adminID, name, role

The relationships are clearly shown: a Student can schedule many Lessons, an Instructor can teach many Lessons, and Lessons are associated with Tests for tracking readiness. This diagram provides the blueprint for how data is organized and related in the system, making it a cornerstone for the development phase.



## Technical Requirements

The DriverPass system will require the following components to operate efficiently and securely:

**Hardware**

* Web server with at least a quad-core processor, 16 GB of RAM, and 500 GB SSD storage.
* End-user devices including desktop computers, laptops, tablets, or smartphones with internet access.

**Software**

* Web application framework such as Java Spring Boot.
* Relational database management system (MySQL or PostgreSQL).
* Modern, secure web browsers for client access.

**Tools**

* Lucidchart for system diagramming.
* GitHub for version control and collaboration.
* IDE such as Eclipse or IntelliJ for application development.

**Infrastructure**

* Cloud hosting service (AWS or Azure) with HTTPS enabled.
* Secure authentication protocols and encryption for all user data.
* Automated backup system and disaster recovery plan.

By meeting these requirements, the DriverPass system will be capable of supporting users reliably, maintaining data integrity, and protecting sensitive information in compliance with industry standards.