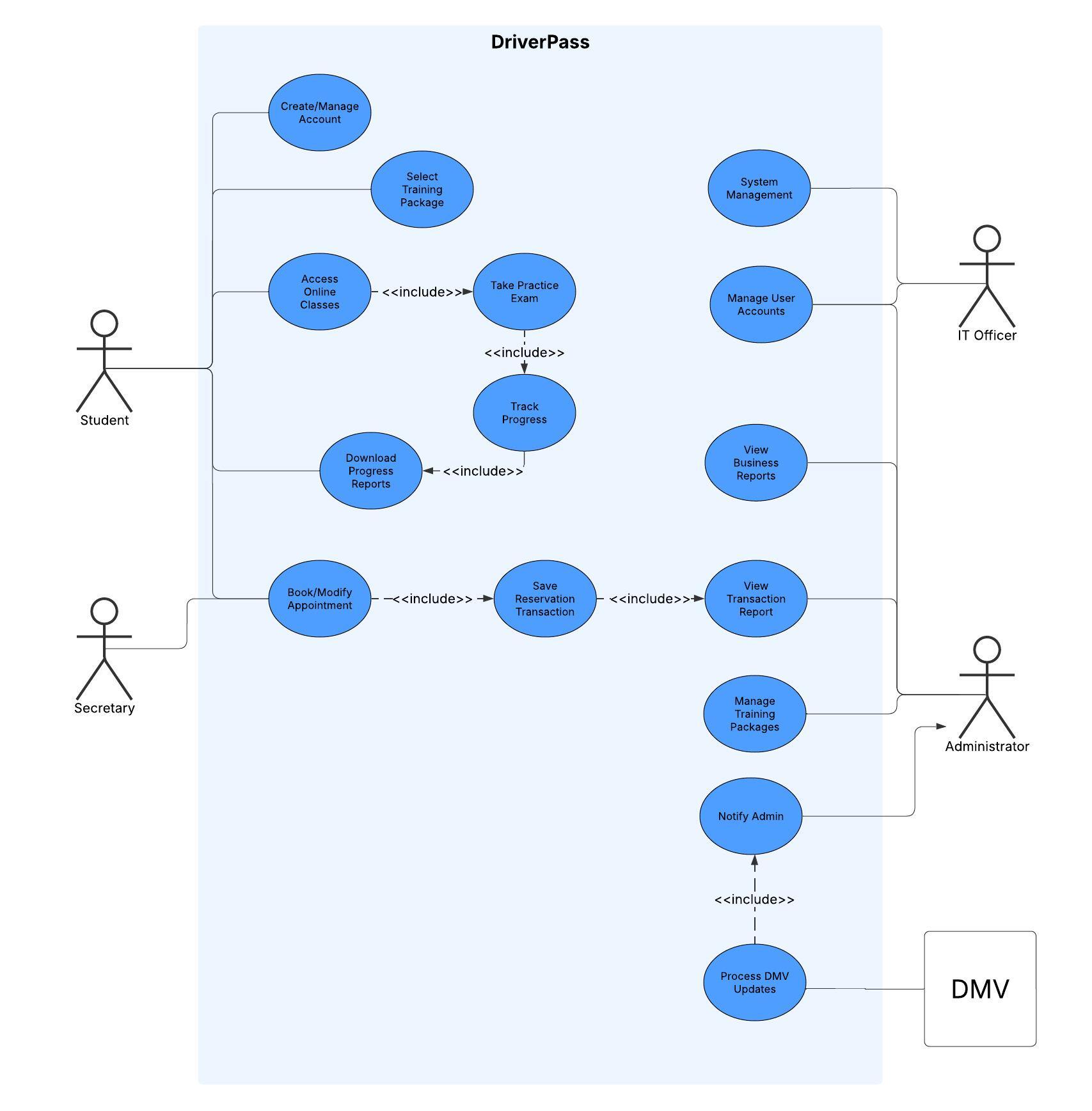
# CS 255 System Design Document Template

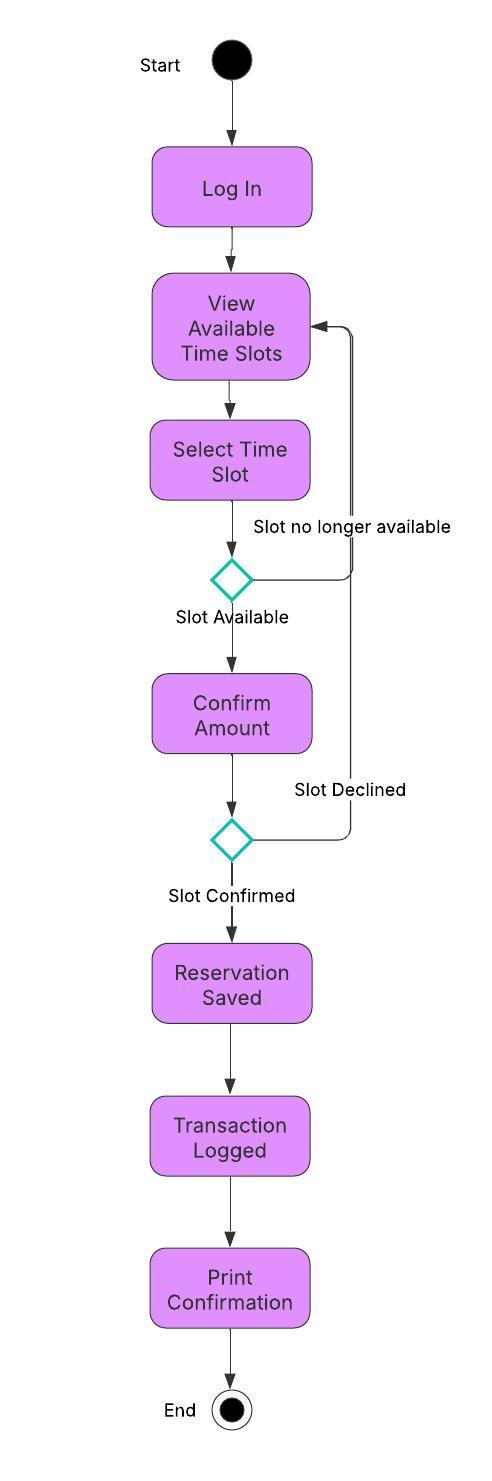
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length 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

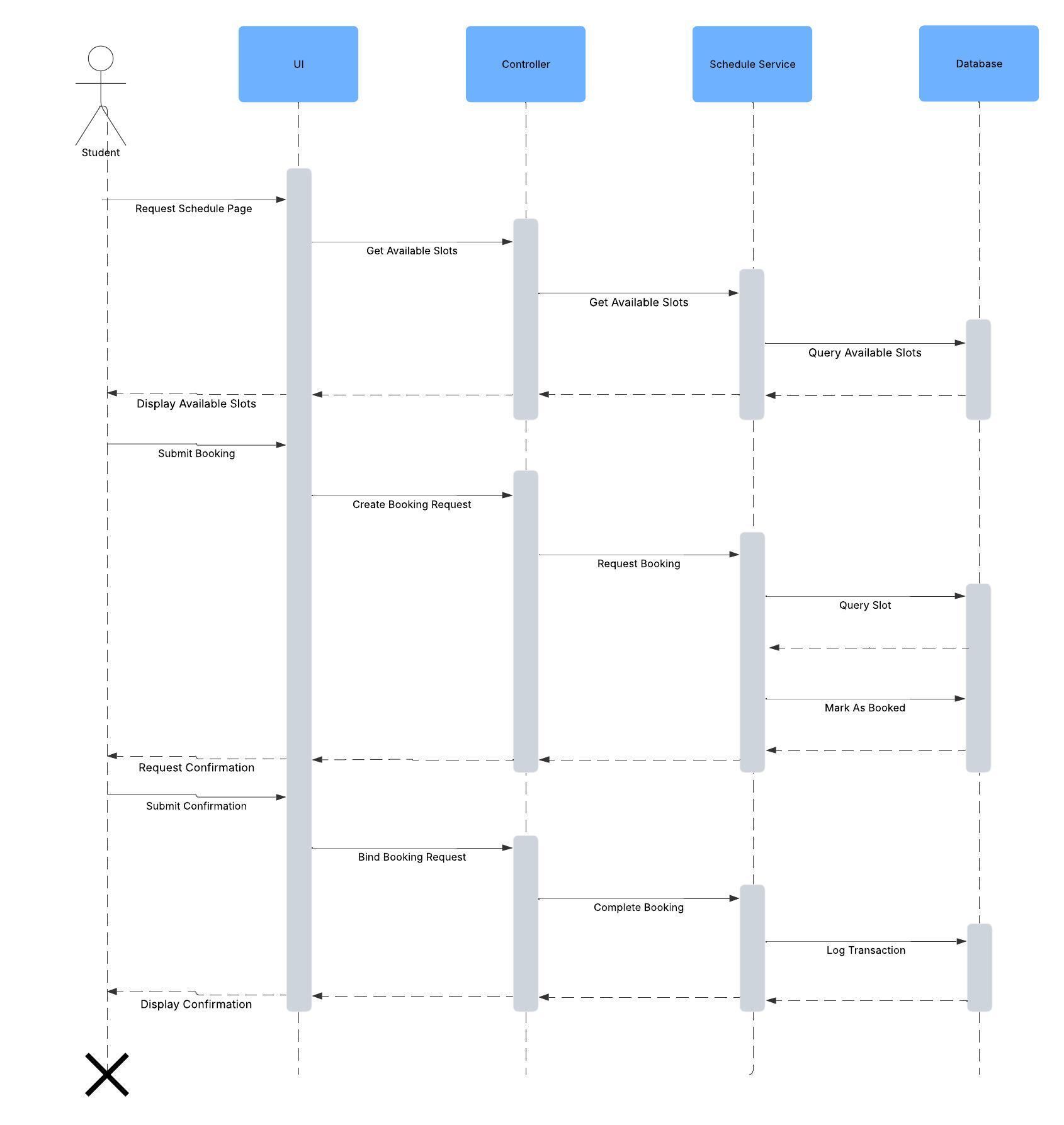
### UML Use Case Diagram

**

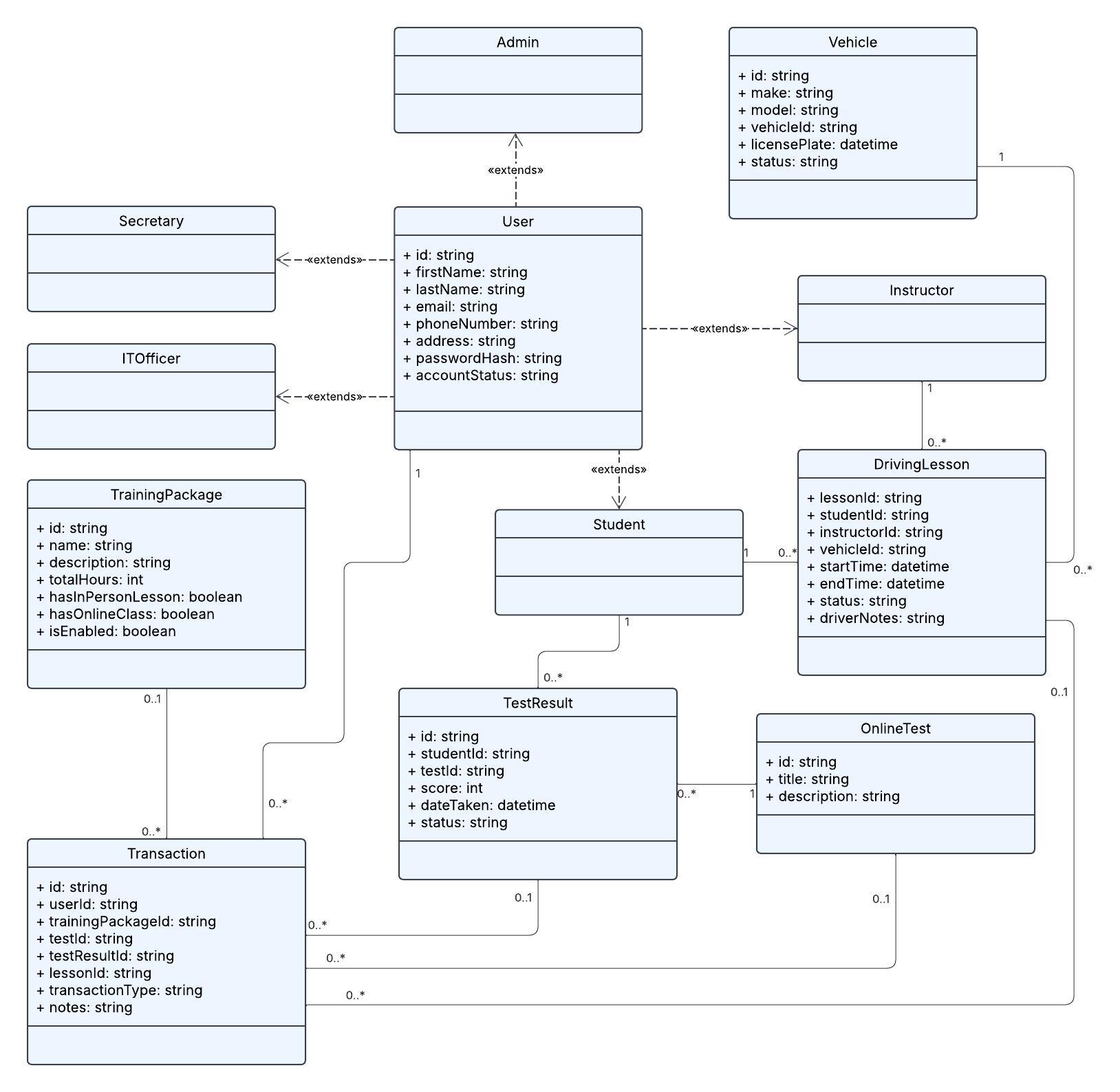
### UML Activity Diagrams

**

### UML Sequence Diagram

**

### UML Class Diagram

**

## Technical Requirements

#### **1. Hardware Requirements**

* **Client Devices**:  
  + Any modern desktop or laptop computer with a supported web browser.
  + Mobile devices, both iOS and Android.
* **Server Infrastructure**:  
  + Cloud hosted virtual server instances running Linux.
  + Scalable hardware support via AWS EC2 or comparable cloud services.
  + Load balancer and auto-scaling groups to manage spikes in student scheduling activity.
* **Storage**:  
  + Cloud-based object storage for backups and reports.
  + Relational database.

#### **2. Software Requirements**

* **Front-End**:  
  + A modern web framework for a responsive user interface.
* **Back-End**:  
  + Web server framework
  + RESTful APIs for communication between frontend and backend.
  + Email service for account confirmations and password resets.
* **Database**:  
  + Relational Database for storing user profiles, lesson bookings, test results, and instructor schedules.
* **Authentication & Authorization**:  
  + OAuth or token-based security.
  + Role-based access controls to differentiate permissions between students, IT officers, secretaries, and administrators.

#### **3. Tools**

* **Development**:  
  + Visual Studio Code for local development.
  + Git and GitHub for version control.
  + Postman for API testing.
* **Design**:  
  + Lucidchart for UML diagrams.
  + Figma for wireframing UI mockups.
* **Monitoring & Logging**:  
  + AWS CloudWatch for tracking system health and activity logs.
  + Error logging via tools like Sentry.

#### **4. Infrastructure**

* **Cloud Hosting**:  
  + AWS for hosting the entire system.
  + Includes services like:  
    - EC2 for compute
    - RDS for database
    - S3 for file storage
    - CloudFront for static file delivery
* **Security Infrastructure**:  
  + SSL/TLS certificates for HTTPS
  + Web Application Firewall to block malicious requests
  + IAM roles and permissions to control internal access
* **Compliance & Backup**:  
  + Scheduled automatic backups for database and reports
  + Endpoint for receiving DMV updates
  + System audit logs for compliance tracking