# CS 255 System Design Document Template

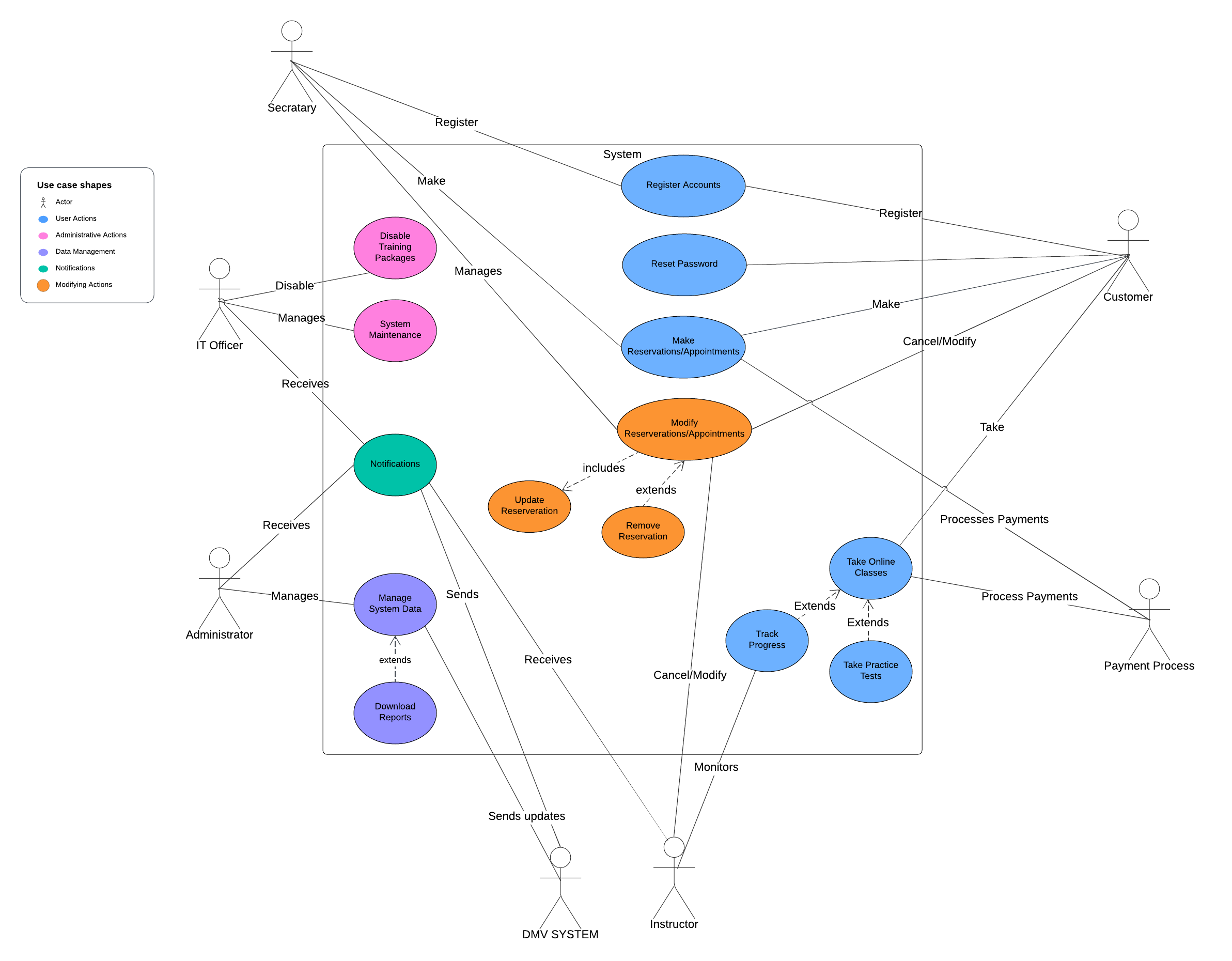
Alexander Flood

23rd, June 2024

# CS-255 System Analysis and Design

## UML Diagrams

### UML Use Case Diagram

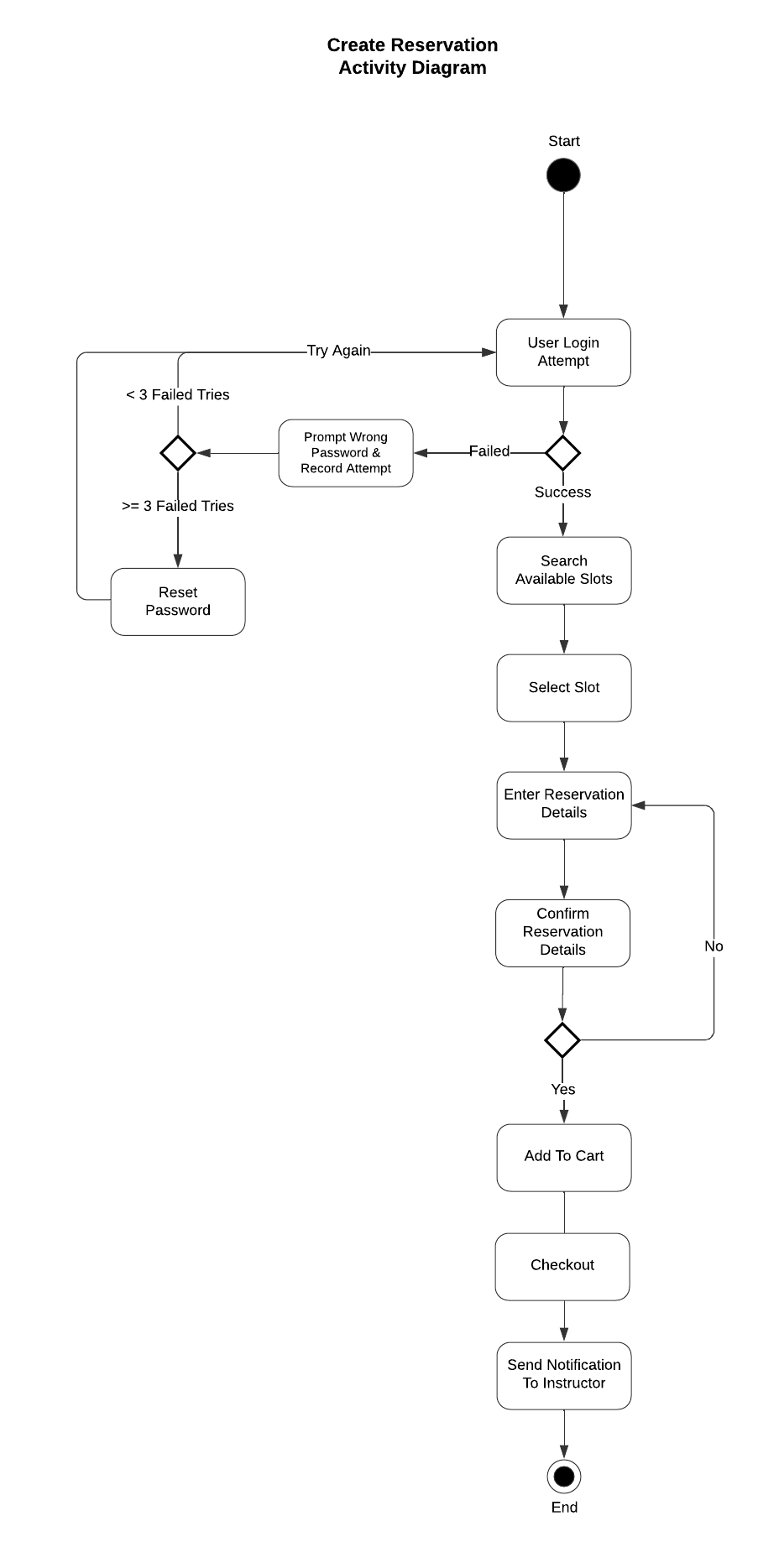


The use case diagram for the DMV outlines various actions performed by different actors, including customers, secretaries, IT officers, administrators, instructors, and the payment processor. It includes key actions users, staff, and administrators can expect to perform within the system such as creating, modifying, and canceling reservations. Enrolling in different packages/courses, taking practice tests, and tracking progress.

This diagram shows the relationship between each actor and these key actions, showing how the actors and use cases interact and relate to one another. Together it provides an overview of the key actors and functionality of our system.

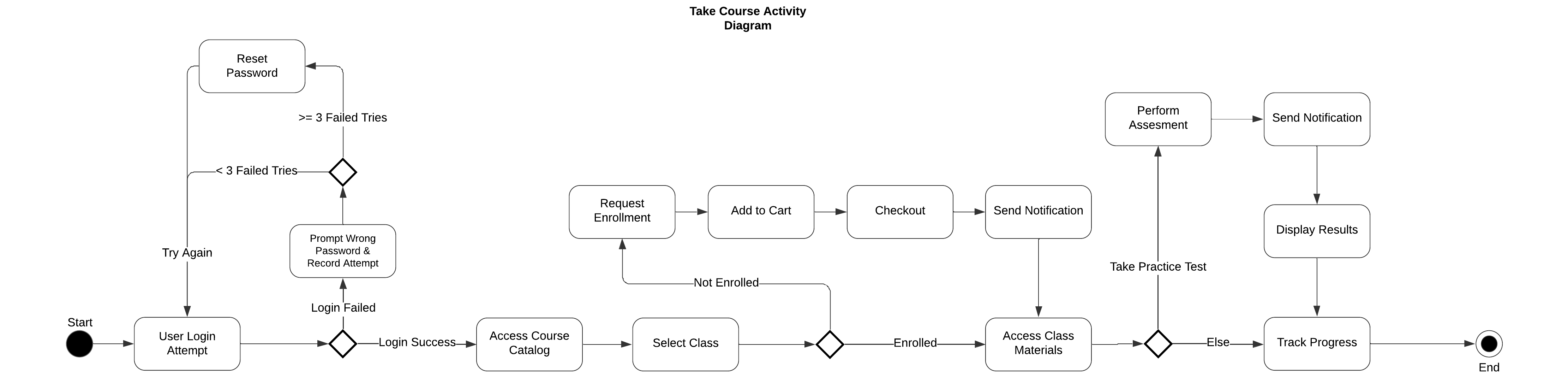
### UML Activity Diagrams

**Create Reservation Activity Diagram**



The Create Reservation activity diagram outlines the process for a user to log in, search available time slots, and create a reservation. This flow includes handling failed login attempts, eventually prompting the user to reset their password after a failed number of attempts. Upon login, the user can search for and select an available slot, enter reservation details, and confirm them. They add the reservation to the cart and the user goes to checkout. The instructor is notified after the user pays for the reservation to finalize the process. This diagram ensures a user-friendly and seamless process that includes error handling and notifications.

**Take Course Activity Diagram**

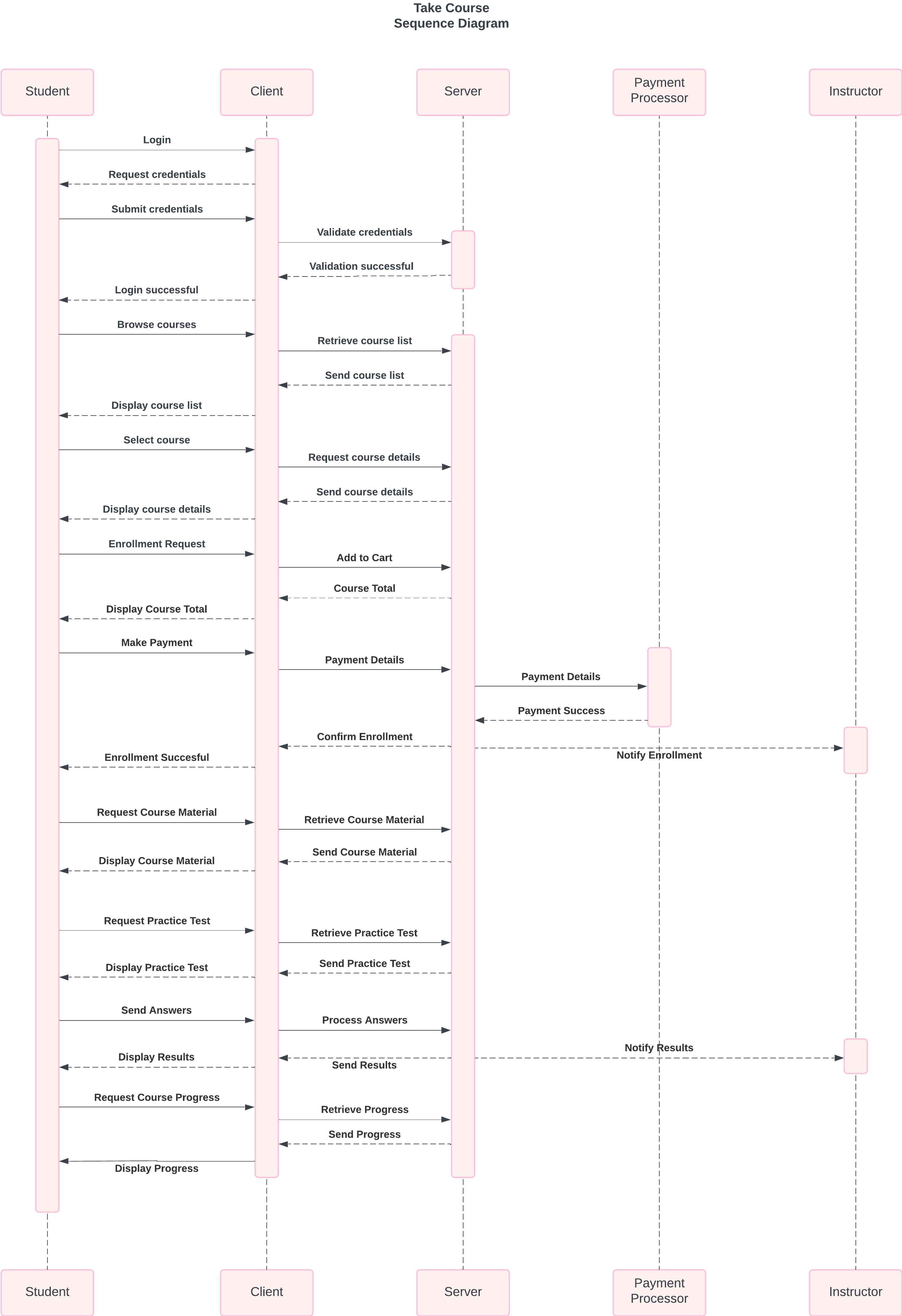


The Take Course activity diagram outlines the process for a user to log in, enroll in a course, access class materials, and track their progress. It also includes handling of failed login attempts and the use of a checkout process for enrolling in courses, ensuring all courses are paid for prior to access.

This diagram provides a comprehensive flow from user login to course enrollment, material access and process tracking, including error handling and notifications. This ensures customers can easily customers are provided with an intuitive and user-friendly process that is easy to understand, leading to increased customer enrollment.

### UML Sequence Diagram

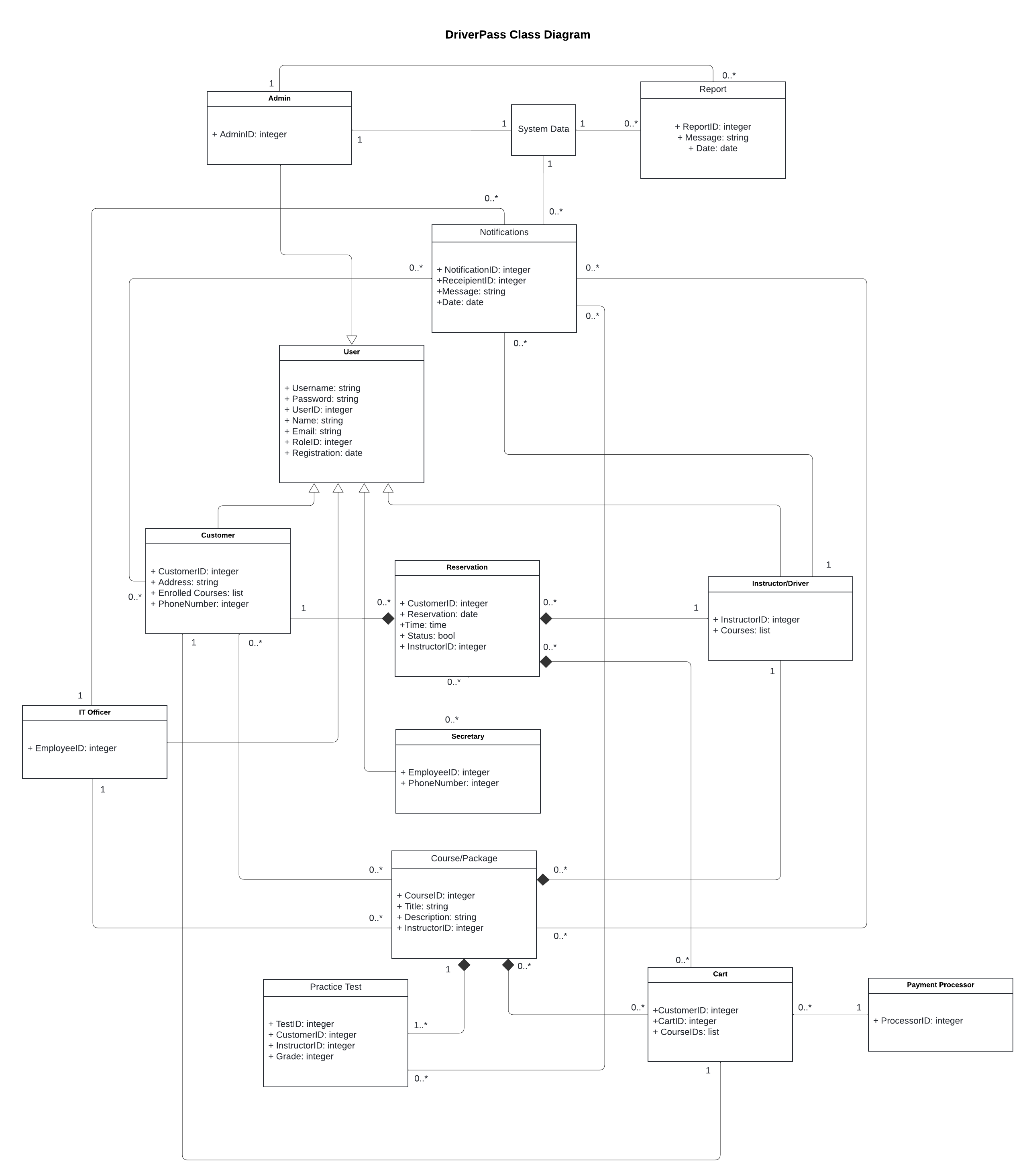
**Take Course Sequence Diagram**



The Take Course sequence diagram shows the interactions between a student, client, server, payment processor, and instructor to facilitate course/package enrollment and participation. The process starts with the student logging in through the client which validates the user credentials with the server. Upon successful login, the student can browse and select courses. The client retrieves and displays the course details from the server.

The student can then request enrollment, add the course to their cart, and checkout. After the payment is processed by the payment processor, the server notifies the instructor of a new enrollment and sends the course materials to the client. The student can request and complete a practice test, with the server processing and sending the results to the student and notifying the instructor. Finally, the student can request and view their course progress. This diagram provides a smooth and interactive process for general course enrollement and participation, keeping the student and instructor in the loop.

### UML Class Diagram



The DriverPass system is designed to efficiently manage reservations, courses, and related processes for various user roles, including customers, instructors, IT officers, secretaries, and administrators. Customers can enroll in multiple courses, manage reservations, and take practice tests, all facilitated by instructors who oversee courses, grade tests, and fulfill reservations. IT officers enable or disable course packages as needed, while secretaries can create reservations on behalf of customers.

Administrators oversee the system data and create reports as needed. The system provides notifications for key events such as reservations, course enrollments, and practice tests. The system also provides a streamlined checkout process through an external payment processor. This design ensures comprehensive management of educational and reservation activities.

## Technical Requirements

***Hardware Requirements***

* ***Servers:*** *High performance servers to host the application, manage user login sessions, process data such as practice tests, and handle transactions.*
* ***Storage:*** *Adequate storage should be provided via solid state drives for storing user data, course materials, system logs, and user data.*
* ***Client Devices:*** *User devices such as PCs, laptops, and smartphones capable of accessing a modern web browser.*

***Software Requirements***

* ***Operating System:*** *A Linux based operating system for the server environment is recommended for stability, security, and savings on licensing fees.*
* ***Database Management:*** *A modern and robust relation database management system such as MySQL or PostgreSQL is recommended for storing user accounts, course data, reservations, and system logs.*
* ***Programming Languages:*** *The system can be implemented in a modern language that supports object-oriented paradigms such as Python, Java, or Javascript.*
* ***Notification System:*** *A real-time notification system must be implemented or purchased and integrated to support system level notifications. Cloud solutions do exist for this service.*

***Tools***

* ***Integrated Development Environment:*** *A standard IDE for developers to code in such as Visual Studio Code or Eclipse will be needed for coding and debugging.*
* ***Version Control System:*** *A version control system such as Git will be needed to manage source code.*
* ***Testing Framework:*** *A testing framework must be selected upon selection of a programming language. This will be used for unity, integration, and end-to-end testing to ensure proper functionality of the DriverPass system.*

**Infrastructure**

* **Cloud Provider:** A cloud platform such as Amazon Web Services, Microsoft Azure, or Google Cloud should be selected as a host for our DriverPass system server. This will allow for scalable hosting, storage, and database solutions that allow the system to adapt to new growth.
* **Security:** The system network should use SSL/TLS encryption for secure data transmission and firewalls to prevent network intruders**.** The system components, including all security components, should be subject to regular security audits to ensure system integrity.
* **Backup Solutions:** Regular backup of sensitive user data should be performed regularly. Disaster recovery plans should be established to prevent data loss in cases of system failure.
* **Monitoring and Logging:** System and network level tools should be implemented to provide detailed logs of both system activity and network traffic. These logs can allow the IT Officer and administrator to analyze system and network behavior identifying any suspicious activity.

The system requires robust hardware, a reliable network, and scalable cloud infrastructure to ensure all user interactions and data is processed efficiently. The software stack provides secure payment integration, a set of comprehensive development tools, and ensures a secure and seamless user experience. Implementing these technical requirements will ensure that the system is performant and capable of handling the described functionalities in the diagrams and interactions described in this document.