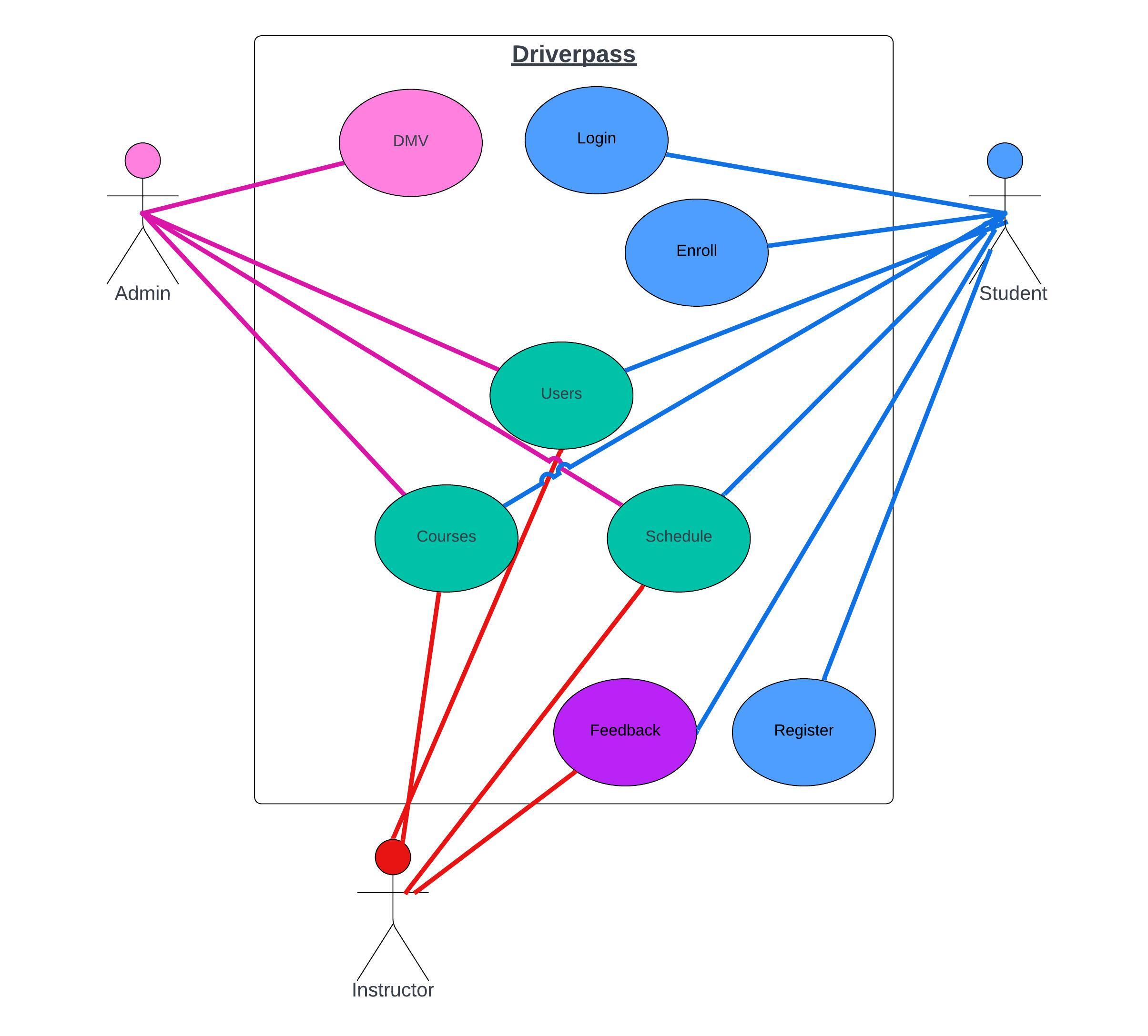
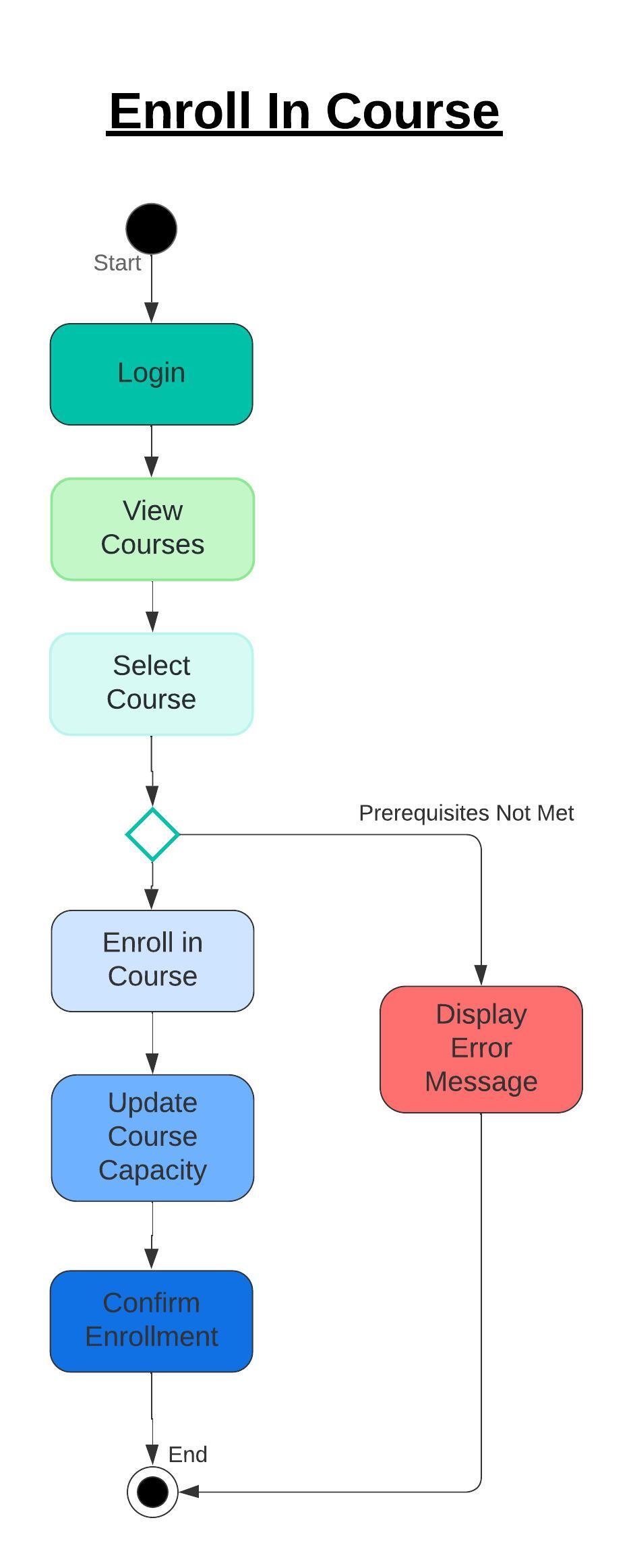
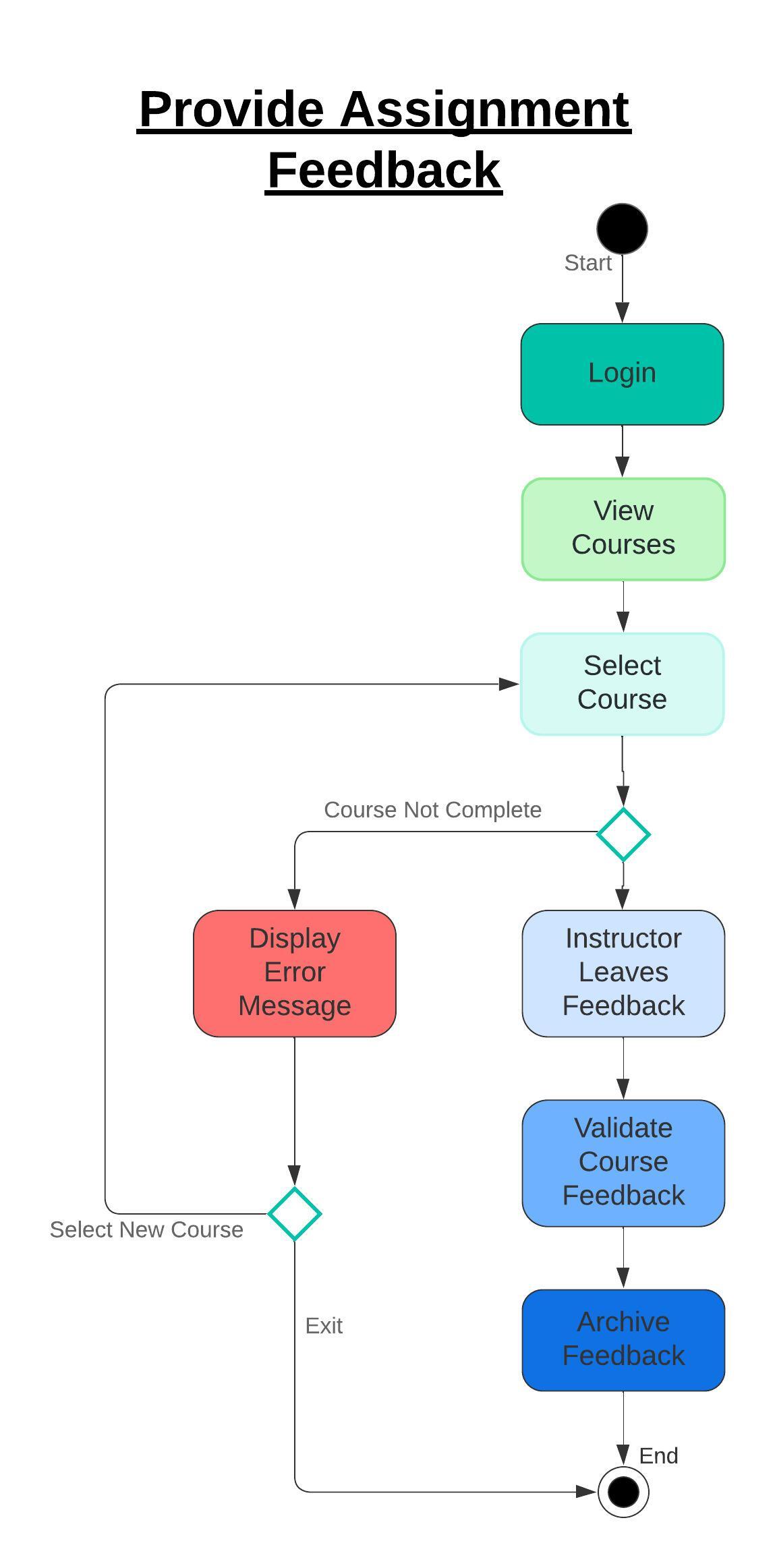
# CS 255 System Design Document

## UML Diagrams

### UML Use Case Diagram

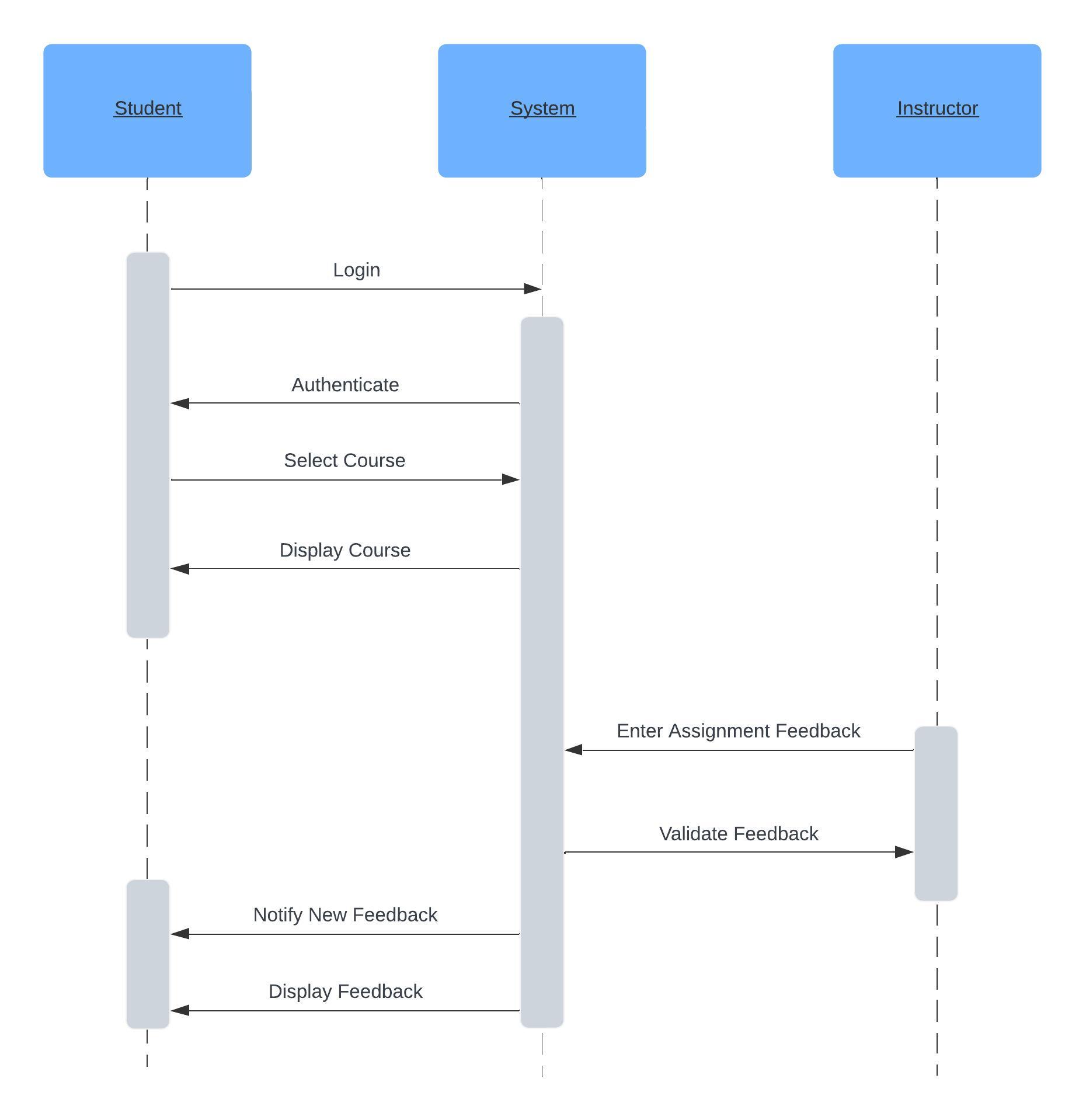
**

### UML Activity Diagrams

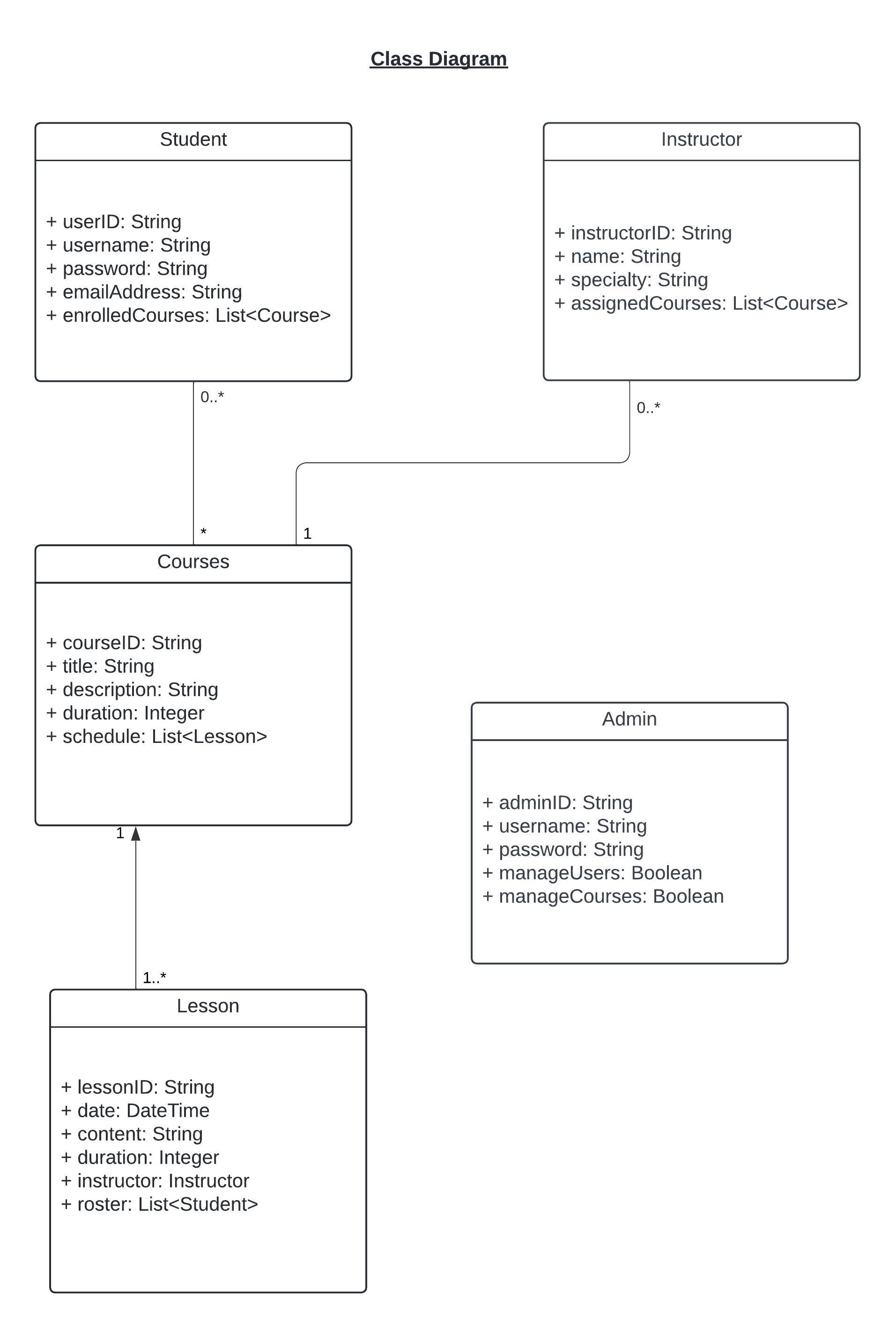
** 

### UML Sequence Diagram

Provide Assignment Feedback

**

### UML Class Diagram

**

## Technical Requirements

**Hardware Requirements:**

To accommodate the operational demands of driving schools, DriverPass requires a scalable and secure server environment. The use of cloud service providers such as AWS or Azure is specified to ensure this, offering the flexibility to scale up resources in response to the platform's activity. Additionally, these providers must supply the necessary networking hardware—like firewalls and load balancers—to facilitate efficient data flow and provide a secure and performant system. Furthermore, robust storage solutions are required to manage and safeguard the extensive educational content, user information, and transaction data.

**Software Requirements:**

DriverPass's software architecture must be robust and flexible to meet the system's dynamic needs. Backend development will depend on Java with the Spring framework or Node.js, providing the necessary capabilities for handling complex operations and real-time data interactions. For the frontend, the software must include HTML5, CSS3, and JavaScript to ensure a user-friendly and intuitive interface. Additionally, the implementation of frameworks like React or Angular is needed to guarantee a responsive experience across various devices, including desktops and mobile phones.

**Tooling Requirements:**

The development process for DriverPass calls for sophisticated tooling. Version control via Git is necessary for effective source code management and team collaboration. For project tracking and management, tools such as JIRA are required to provide visibility over development timelines and to keep tasks in alignment with the project's strategic goals. CI/CD pipelines are also a requirement to enable automated testing and deployment, supporting a development cycle that is both fast-paced and quality-assured.

**Infrastructure Requirements:**

The DriverPass platform's infrastructure must be designed for scalability to support anticipated growth. It must include cloud-based solutions that can dynamically adjust to user demand and system usage patterns. API integration with DMV systems is a crucial requirement to ensure the platform's content remains current with regulatory changes and testing standards. The choice of a central database—either SQL or NoSQL—is essential and must be determined based on its reliability, performance, and ability to manage the vast data requirements effectively.