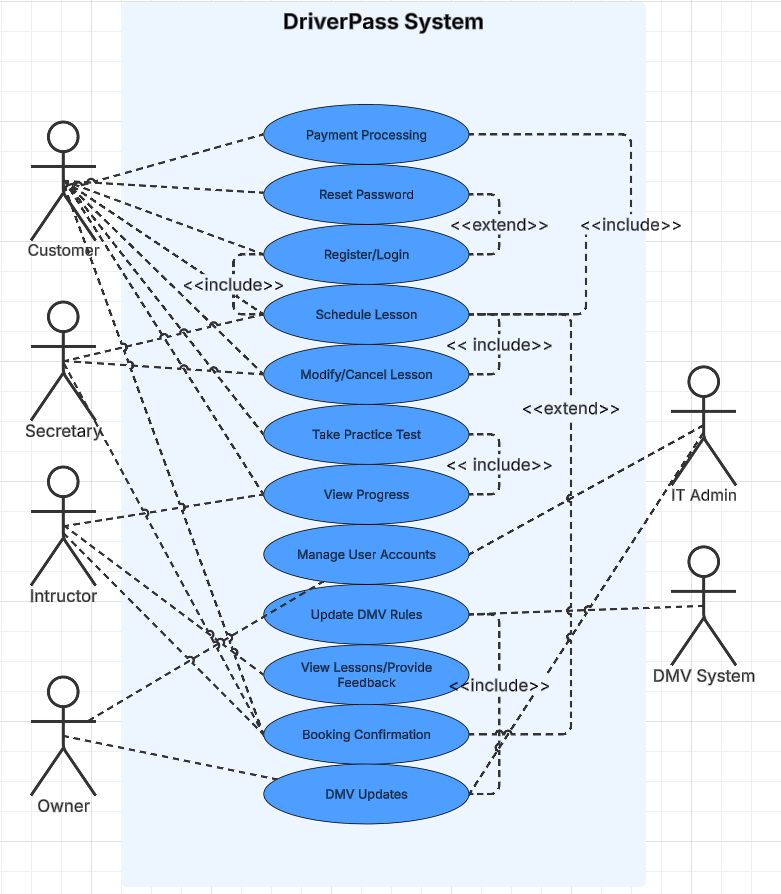
# CS 255 System Design Document Template

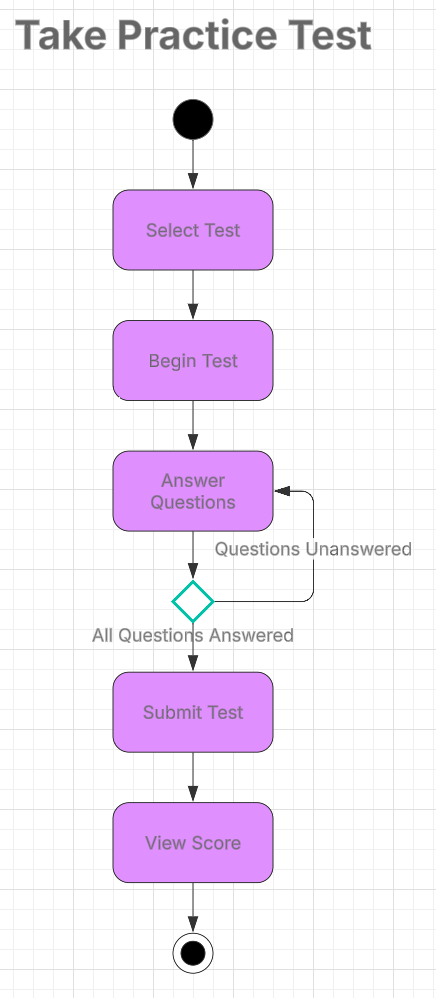
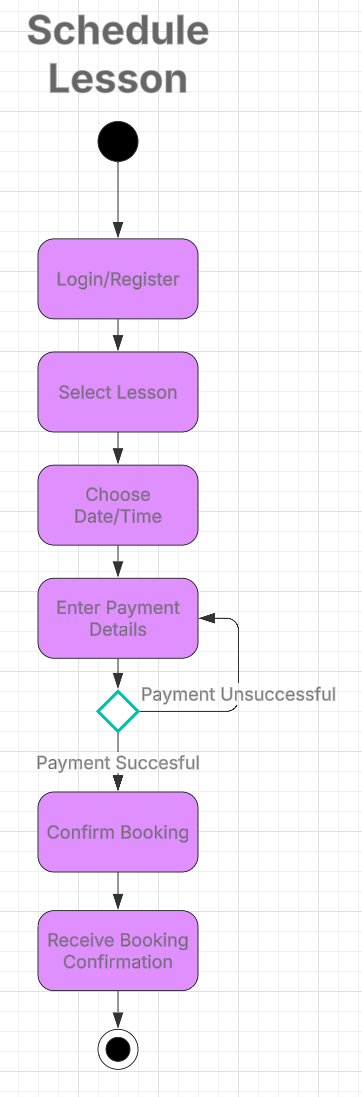
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length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

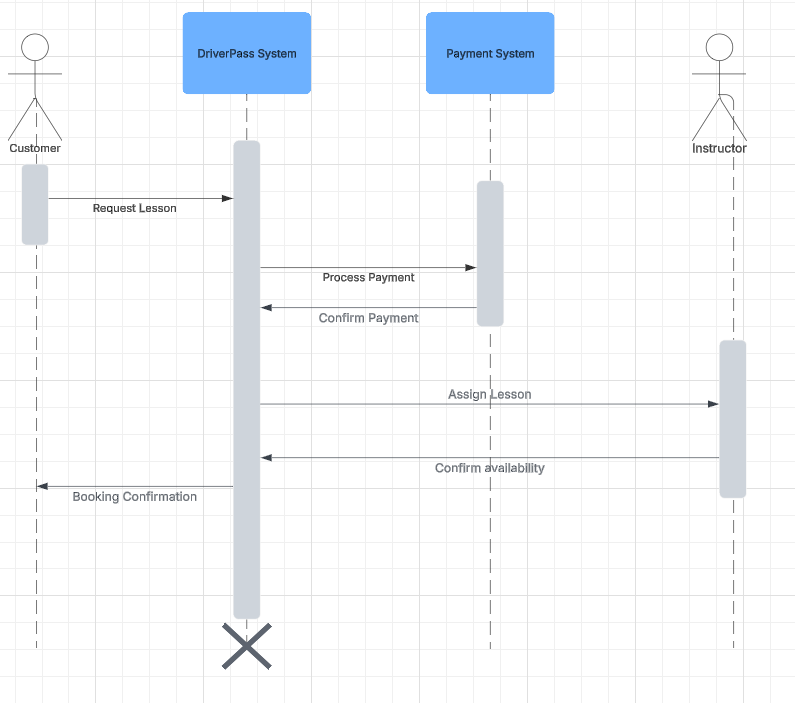
### UML Use Case Diagram



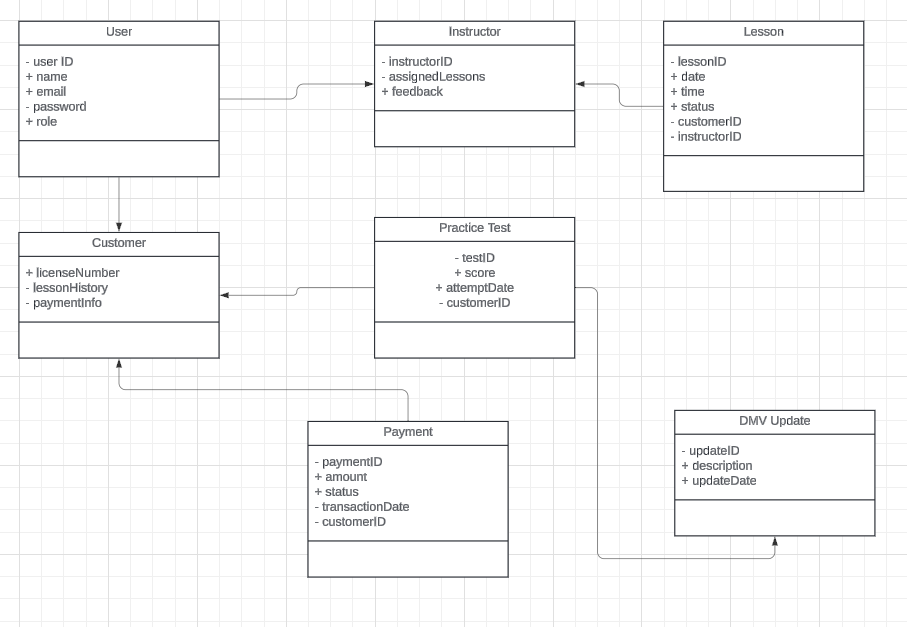
### UML Activity Diagrams



### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

The system-based and cloud-hosted, ensuring accessibility across desktop, tablet, and mobile devices through modern web browsers. To achieve this, the hardware infrastructure must include cloud-based servers to host the application and database. A relational database management system like MySQL, PostgreSQL, or Microsoft SQL Server will be required to store suer accounts, lesson bookings, payments, and test results. The application backend will be developed using scalable framework, such as Node.js while the frontend will leverage HTML, CSS and JavaScript for an intuitive user interface.

For security and compliance, the system must implement SSL/TLS encryption for secure data exchange, along with role-based access controls to restrict access based on user roles. The payment processing system will integrate with third party services like Stripe or PayPal to securely handle transactions. Additionally, the system must support automatic DMV updates, requiring an API connection to secure data integration to retrieve regulatory changes. Regular automated backups and a disaster recovery plan will be essential to ensure data integrity and business continuity.

Development and maintenance of the system will require integrated development environments such as Visual Studio Code, along with version control using GitHub. The testing environment must include automated unit testing, integration testing, and load testing tools like JUnit to validate functionality and performance. The system must also be optimized for scalability, allowing for future expansion if DriverPass decides to add more lesson types, locations, or users. This infrastructure ensures a secure, scalable, and efficient system that meets DriverPass’s business needs while maintaining high availability and reliability.