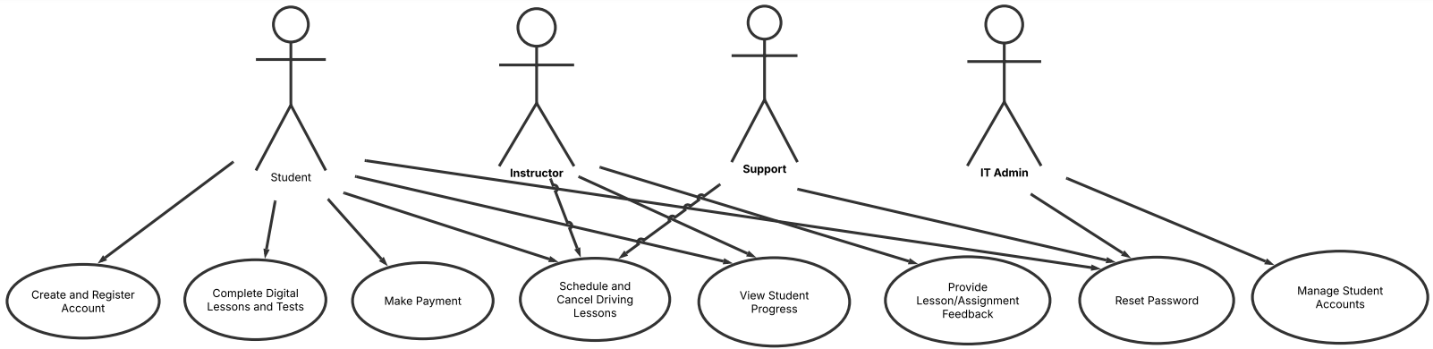
# CS 255 System Design Document Template

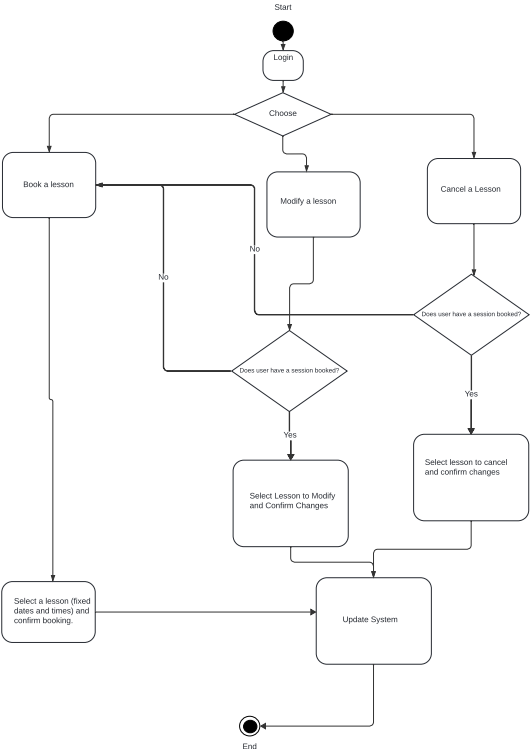
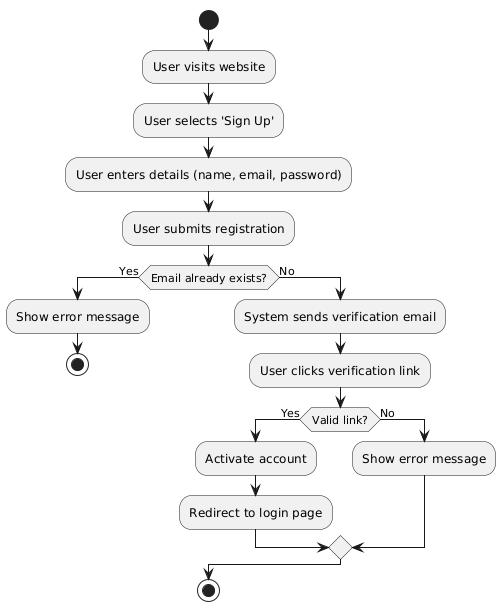
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length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

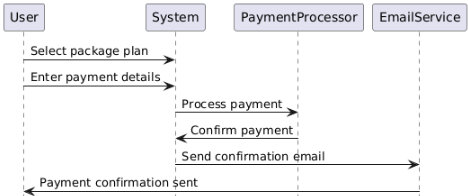
### UML Use Case Diagram



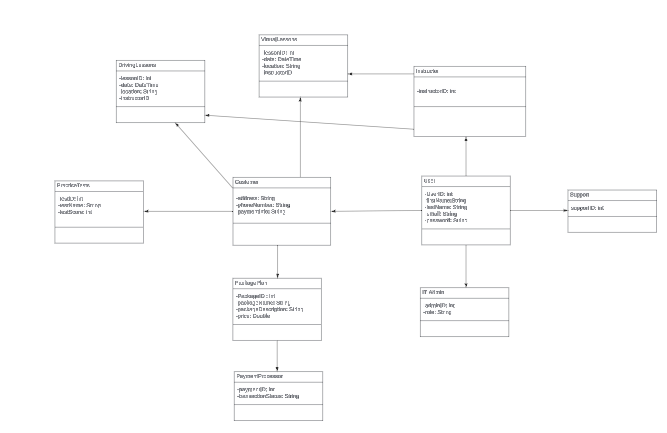
### UML Activity Diagrams



### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

Regarding the UML diagrams and the system, I created, there are many requirements to consider. I think the system should run on a Linux-based operating system like Ubuntu Server or CentOS, ensuring stability and security. For managing data, it will use MySQL as the database. The backend will be built using Python to handle system operations, while the frontend will use React.js, Vue.js, or Angular to create an easy-to-use interface. If a mobile app is needed, Flutter will be used for cross-platform development, while Kotlin and Swift will be used for Android and iOS apps. Payments will be securely processed using Stripe or PayPal, and notifications will be sent through services like SendGrid, AWS SES, Firebase Cloud Messaging, or OneSignal. To make development and deployment smoother, the system will use Git for version control, with coding done in IntelliJ IDEA or PyCharm. Automated processes will be set up using Jenkins or GitHub Actions, and project management tools like Jira, Trello, or Asana will help keep track of progress. The system will also use Docker and Kubernetes to ensure scalability and efficient deployment. Hosting will be on cloud platforms such as AWS, Google Cloud, or Microsoft Azure, with databases managed through Amazon RDS or Firebase. Security will be a key focus, with SSL/TLS encryption, OAuth 2.0 or JWT for authentication, and daily backups to prevent data loss. To protect against cyber threats, services like Cloudflare or AWS Shield will help prevent DDoS attacks. With this setup, the system will be reliable, secure, and scalable. The chosen technologies I have chosen are all reliable and up-to-date software. They will all ensure that the platform can grow over time while maintaining high performance and security standards.