# 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.

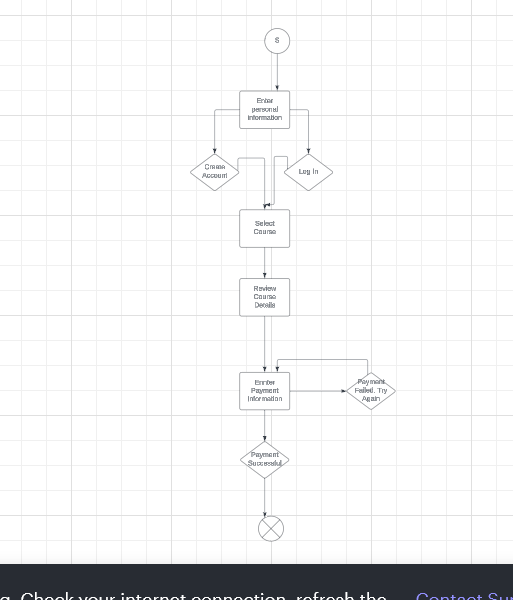
## UML Diagrams

### UML Use Case Diagram

A diagram of a person's process

Description automatically generated

### UML Activity Diagrams

**

*A diagram of a lesson

Description automatically generated*

### UML Sequence Diagram

### UML Class Diagram

A diagram of a program

Description automatically generated

A diagram of a lesson

Description automatically generated

## Technical Requirements

The DriverPass system requires robust technical infrastructure to ensure its functionality, scalability, and security. The system must be accessible through modern web browsers, supported by a secure web-based application hosted on a reliable server infrastructure. This can be achieved either through cloud hosting (such as AWS or Azure) or an on-premises solution, both of which must provide high availability and failover mechanisms. Key components like a relational database (e.g., MySQL or PostgreSQL) will store user data, including student profiles, lesson schedules, and instructor information. To safeguard this sensitive information, encryption protocols (such as SSL for secure login and data transmission) must be implemented. The system also needs reliable internet connectivity, firewall protection, and intrusion detection systems to prevent cyber threats. In addition, backup storage is essential for regular data redundancy and recovery.

From a development standpoint, tools such as Visual Studio Code or IntelliJ IDEA will be used to build and maintain the codebase, supported by a version control system like Git for collaboration. The system’s functional requirements include secure user authentication, real-time lesson scheduling, and administrator access to manage courses and schedules. Non-functional requirements ensure that the system remains scalable, user-friendly, and performant, with minimal delays in request processing. Key features like email or SMS notifications for students and data privacy protections will further enhance the usability and security of the platform, meeting the client’s need for a reliable, automated solution for managing driving lessons and student progress.