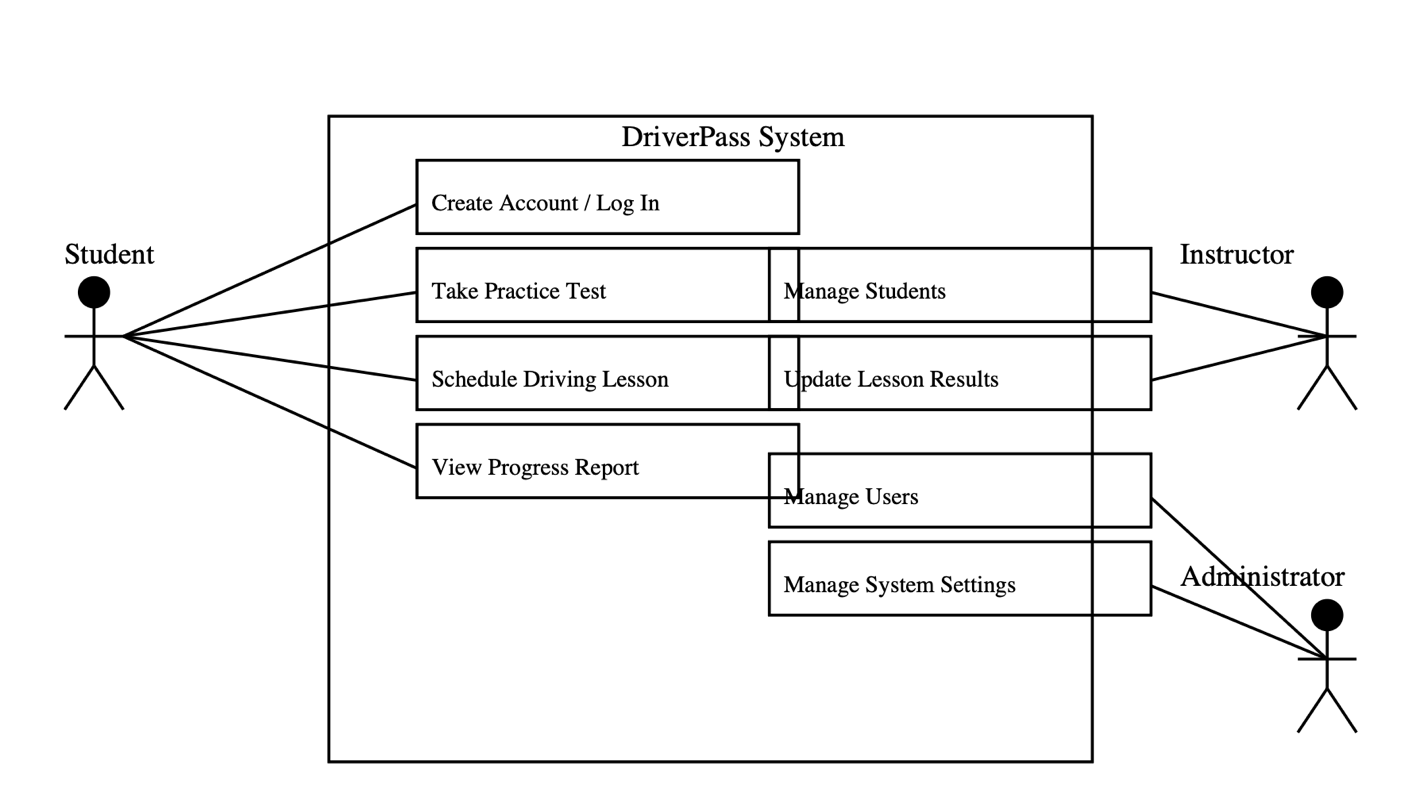
# CS 255 System Design Document Template

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

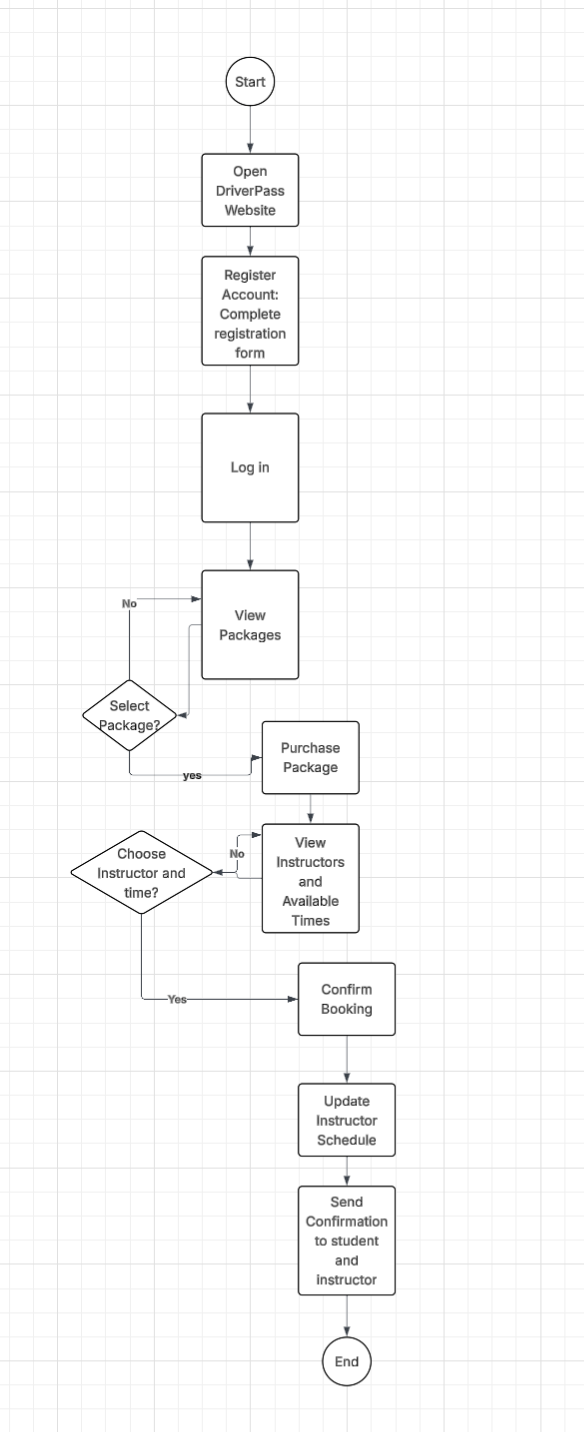
### UML Use Case Diagram



DriverPass use case diagram identifies the primary users of the system and the activity that each one can perform. The primary users are instructor, student, and administrator. Students can register for an account, purchase lessons packages, practice tests online, schedule lessons, and view test results. Instructors can inspect and manage their schedule, accept or decline booking requests, and update lesson completion status and student progress. Administrators manage user accounts, payment processing, and update content and system settings. The diagram groups these tasks around the three actors so the client is immediately aware of what each of them does.

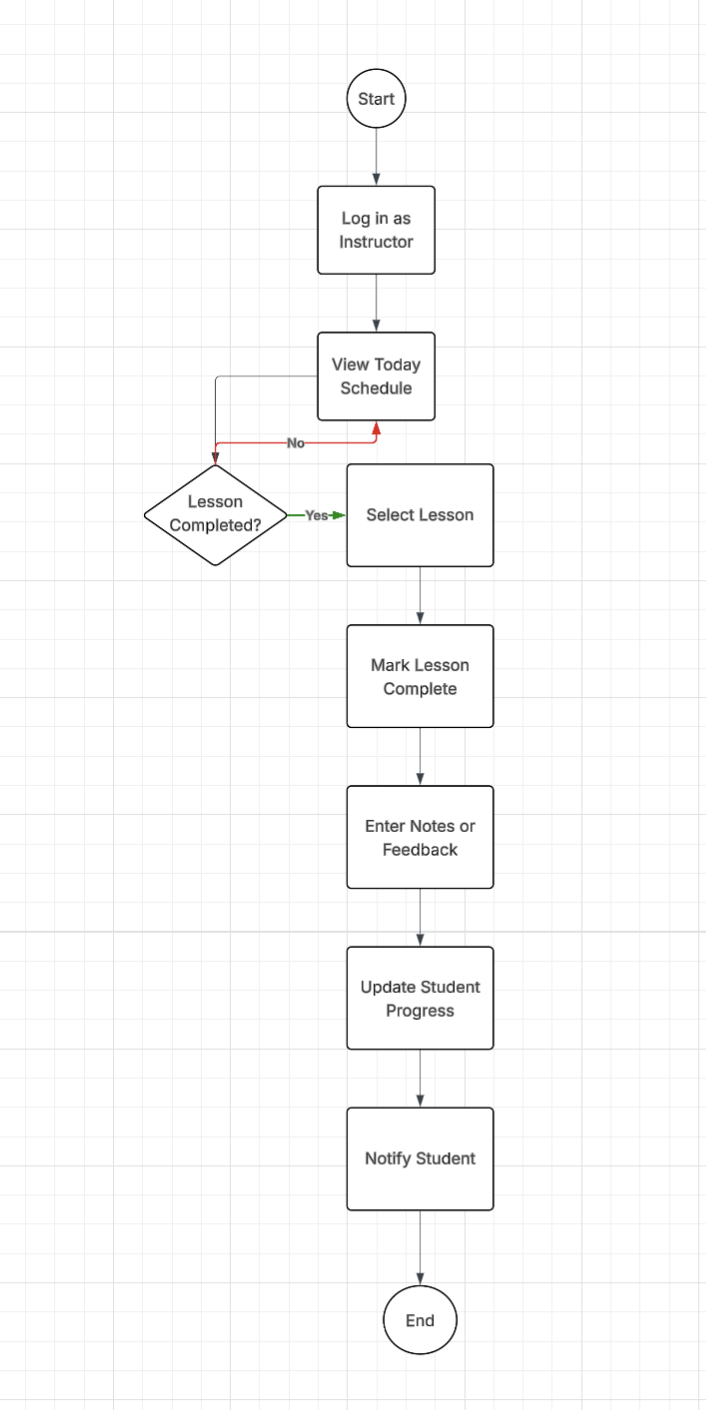
### UML Activity Diagrams

Activity 1:



The first activity diagram provides for the flow when a student registers and schedules a lesson. The student goes to the website, completes a registration form with his/her personal details and login details, and then logs in. After logging in, the student views available lesson packages and selects one to purchase. Once the purchase is completed, the student searches for available instructors and timings, selects a desired instructor and timing, and makes a booking. The system updates the instructor schedule and sends instructor and student booking confirmation messages. This activity diagram describes step by step interaction from registration to booking in such a way that it presents the visible activities a student takes and system reactions in turn.

Activity 2:



The second activity diagram describes the process of how an instructor marks a lesson as completed after completion. The instructor signs in to the system and enters their day view of schedule. The instructor selects the lesson completed, marks it as done, and includes any feedback or comments for the student. The system keeps the lesson status and instructor comments, updates the student progress record, and optionally notifies the student that the lesson is completed. This flow focuses on the instructor end of the process and describes how a completed lesson is added into the student's record.

### UML Sequence Diagram

A diagram of a system

AI-generated content may be incorrect.

The lesson scheduling sequence diagram explains the communication between the student user interface, the system logic, the schedule database, and the instructor notification component. When scheduling is requested by the student, the request is made by the user interface to the system logic that queries the schedule database for schedules. The available schedules are sent back to the student by the system logic who selects a time. The system books into the schedule database and notifies the instructor. Finally, the system notifies the student by sending a booking confirmation. This diagram serves to identify which part of the system does what and the order that the messages and updates are sent.

### UML Class Diagram

*A diagram of a diagram

AI-generated content may be incorrect.*

Class diagram shows the main data structures and how they relate to each other in the DriverPass system. There are some main classes such as Student, Instructor, Lesson, Package, Payment, and Admin. Student holds identification, contact details, and package or progress information. Instructor holds identification and schedule information. Lesson holds date, time, location, status, and a reference to a single student and a single instructor. Package holds the package type, number of lessons, and price. Payment amount, date, and type and is associated with the student and package it covers. Admin is the staff member who can administer accounts and system settings. The relationships indicate that students purchase packages, packages contain lessons, each lesson is associated with one teacher, and payments are associated with purchases.

## Technical Requirements

The system needs a web server and a database server on a reliable cloud platform with daily backup and SSL encryption for establishing secure connections. Server specifications must be sized for expected users, but a reasonable starting point is a server with at least 8 gigabytes of RAM, a quad core processor, and sufficient disk space for media and user data. The software stack can use a common web server such as Apache or Nginx, a back-end language such as Java or Python, a relational database such as MySQL or PostgreSQL, and front-end technologies such as HTML, CSS, and JavaScript. The development team can use Lucidchart for modeling, an integrated development environment such as Visual Studio Code, and version control such as Git. Finally, the environment must possess user authentication, role-based access for administrators, instructors, and students, automatic backup, and monitoring for security or performance issues.