# 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 screenshot of a computer

Description automatically generated

This diagram shows the different actors of the system. These are the people and things that interact with the system, and the ways that that they interact with the system. There is the Student actor, who can create a profile, purchase driving packages, study online materials, take practice tests, schedule and edit driving lessons, and view their student information. The Reception actor can do many of the same activities, assisting the Students. The Driver can also post notes on the driving lessons. The Payment processor must be involved to process payments, and the DMV must be able to update information for the system. The IT and Admin have full access to the system.

### UML Activity Diagrams

A black and white text on a white background

Description automatically generated

This diagram shows the use case, or activity, of creating a Student profile by the Student. The Student starts on the DriverPass website, and clicks “Create Profile”. The student then enters their personal information, a username, and password. The Student is then prompted to purchase a driving package, although it is not required and can be purchased later. Once the confirmation message is displayed, the profile has been created.

A diagram of a workflow

Description automatically generated

This diagram shows the activity of a Student scheduling or editing a driving lesson. If the Student has not purchased driving lessons, or has already completed all their purchased lessons, they cannot schedule or edit. If the Student has purchased lessons, and has lesson hours available, they can schedule a new lesson, or edit the date or time for a lesson that has already been scheduled. The Student can also cancel a lesson. Once the confirmation message has been displayed, the lesson has been booked.

### UML Sequence Diagram

A white sheet of paper with black text

Description automatically generated

This diagram shows the sequence of events that happen when a Student wants to schedule or edit a lesson. It is the same activity of scheduling a lesson, but this diagram shows how the order of events go. This diagram also shows the different objects that are involved in scheduling a lesson, such as the Student, the LessonScheduler, the Student database, and the LessonAppointments database. You can see how they interact with each other through sending messages, such as requesting dates and times from the database and relaying the information to the Student.

### UML Class Diagram

A diagram of a company

Description automatically generated

This diagram is a class diagram that shows the different components of the system and how they relate to one another. The system will use these components to process the different activities of the system. At the top, we have a generic User object, that has different attributes such as username and password. Everyone who uses the system will be a User. Then we have specific types of Users that add their own attributes, such as a Student will have credit card information and driving lessons. There is also a LessonScheduler that is used to create and manage Lessons, this is the only way to create Lessons, and there is only one LessonScheduler that can have many different lessons.

## Technical Requirements

This system will need the following technical requirements:

* A cloud service to host the system for users to visit
* An appropriate operating system and architecture for the system, such as Linux and x86\_64
* Databases to hold user and system information, also part of a cloud service
* API’s for authorization and authentication needs and DMV updates
* CD/CI tools for platform updates
* Secure HTTPS protocols when sending and receiving data over the internet