# 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 diagram of a person's life cycle

Description automatically generated*

### A diagram of a schedule Description automatically generatedA diagram of a flowchart Description automatically generatedUML Activity Diagrams

### UML Sequence Diagram

A diagram of a company

Description automatically generated

### UML Class Diagram

A diagram of a computer flowchart

Description automatically generated

## Technical Requirements

* A reliable ​network connection​ will be needed to access the system. This doesn’t have to be super-fast, but it should be reliable enough to handle tasks like watching videos, scheduling appointments, or making payments without any issues.
* To ensure no data is lost, regular backups should be set up. If something goes wrong, these backups will allow us to restore the system to a recent working state.
* Ensuring the safety of user data is our highest priority. We will implement encryption to safeguard information, which means scrambling the data so that only authorized individuals can read it. Also, we will establish secure login procedures to restrict system access to only those permitted.
* The system will be hosted on a cloud platform. This allows it to be accessible from anywhere and scales up if suddenly a lot of people start using it at once.
* To keep an eye on how the system is running, tools like Google Analytics can be used. These tools let us know if there’s a problem, like if the system is running slow or if there’s an unexpected issue.
* The platform should be accessible on all major web browsers like Chrome, Firefox, Safari. This way users won’t need to download anything extra to use the system, they can just log in from any browser.
* We’ll use something like Node.js for building the back-end of the system. This handles all the “behind-the-scenes” work, like processing login requests, updating profiles, and managing payments.
* To keep track of all the information—like user details, appointments, courses, and payments—we’ll need a database. MySQL is a good choice because it’s dependable and can handle a lot of data without getting overwhelmed.
* The system should work smoothly on common operating systems like Windows for PCs, macOS for Apple users, and Linux. For mobile users, it should support both iOS and Android.
* The system will be accessible to students, instructors, administrators, and other users across a range of devices, including desktops, laptops, smartphones, and tablets. As long as these devices can connect to the internet and support a web browser, users will be able to access the system seamlessly.
* The system needs a dependable server to host the web application. It should have enough storage and processing power to handle multiple users at once, especially during busy times, like when a lot of students are scheduling appointments or taking tests.