# 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 software process

Description automatically generated

### UML Activity Diagrams

*[A diagram of a flowchart

Description automatically generated A diagram of a process

Description automatically generated*

### UML Sequence Diagram

*A diagram of a login system

Description automatically generated*

### UML Class Diagram

*A diagram of a software application

Description automatically generated*

## Technical Requirements

*[Based on the diagrams you have created, describe the technical requirements of your system. These requirements should address the required hardware, software, tools, and infrastructure necessary for your system design.]*

The system will be hosted on Amazon Web Services (AWS), with utilizes the cloud provider’s services. The frontend of the application will be developed using web technologies, including HTML, CSS, and JavaScript, with the backend powered by Node.js. The user interface will be crafted using the React library for a responsive and dynamic user experience. To ensure scalability, the system will utilize AWS's auto-scaling features for adaptability. The architecture will use Amazon RDS for a scalable and managed database solution. The system will feature an appointment booking system, allowing users to schedule, modify, and cancel appointments. Backup and recovery mechanisms will be used to safeguard user data and system configurations. Documentation will be maintained, covering system architecture, APIs, and deployment process, facilitating development, maintenance, and troubleshooting processes throughout the systems lifetime.