# CS 255 System Design Document

## UML Diagrams

### UML Use Case Diagram

*A diagram of a group of people

AI-generated content may be incorrect.*

### UML Activity Diagrams

A diagram of a company

AI-generated content may be incorrect.

*A diagram of a company

AI-generated content may be incorrect.*

### UML Sequence Diagram

A white sheet of paper with arrows

AI-generated content may be incorrect.

### UML Class Diagram

A diagram of a company

AI-generated content may be incorrect.

## Technical Requirements

For the server infrastructure, a cloud hosted server will run the system, needing a CPU like the Intel Xeon, a RAM of 32GB, storage of 1 TD SSD, and high-speed internet connection. All of this to ensure fast database transactions, API requests handled fast, and real-time updates for schedules and notifications.

Users will be able to access through Windows, macOS, and Linux, if using desktops or laptops. However, users will also be able to use mobile devices, where Android and iOS smartphones and tablets will be compatible with the system. These devices must have webcams for identity verification. Printers will be needed to print reports, student records, and invoices.

To ensure scalability, security, and maintainability, the system will be developed and hosted using Linux, as it is widely used for cloud deployments due to its security, stability, and cost-effectiveness.

Data will be stored on MySQL, as we will use a relational database to store all the records. A relational database will be used due to its security, as well as the structured and well-defined relationships between data, and as speed server is important, SQL databases are optimized for structured queries.

For the backend of the system, we will use Python with Django, as it handles API routing, authentication, and database interactions efficiently, allowing for fast development and readability, making prototype and test new features easier. To authenticate users, we will use OAuth 2.0 as it enables secure third-party logins (like Google authentication), and JSON Web Tokens, to ensure stateless authentication, improving system scalability.

The frontend of the system will be developed using JavaScript with React.js, as this framework is component-based structured, which makes UI updates easier and faster, allowing for fast, interactive, and dynamic user experience. However, React.js can only be used for the website that will be used on desktops and laptops, to develop the application that will be used on mobile devices and tablets, we will use React Native, as it allows us to write one codebase and deploy for both iOS and Android, which will reduce development time and costs. Besides, React Native allows us to re-use components across platforms (unlike Swift or Kotlin), saving time and money.

To be able to know the students’ locations, they will input it on the system. However, the system can be integrated with Google Maps API to enable location tracking, not only for the students but also for the instructors. Payments will be made using PayPal, as they will handle the security. We will use SMS and Email APIs to send notifications to keep students and instructors up to date on lessons, cancellations, and progress updates.

The project will be developed using Git and GitHub for control version, as it enables collaborative development, version tracking, and rollback capabilities. For testing, we will use PyTest and Postman, to test APIs endpoints and responses, ensuring the system runs correctly and preventing bugs and performance issues.