# CS 255 System Design Document

## UML Diagrams

### UML Use Case Diagram

*A screenshot of a computer

Description automatically generated with medium confidence*

### UML Activity Diagrams

*A diagram of a process

Description automatically generated with low confidence*

*A picture containing text, screenshot, diagram, font

Description automatically generated*

### UML Sequence Diagram

*A diagram of a project

Description automatically generated with low confidence*

### UML Class Diagram

*A picture containing text, diagram, plan, parallel

Description automatically generated*

## Technical Requirements

These technical requirements address the hardware, software, tools, and infrastructure needed to support the system design, ensuring its reliability, scalability, security, and performance. It is essential to choose appropriate technologies and infrastructure based on the anticipated user load, budget constraints, and business needs. Based on the functional and nonfunctional requirements discussed, the technical requirements for this system design include the following:

Hardware Requirements

* Server Infrastructure: Sufficient hardware resources to host the web-hosted application, including processing power, memory, and storage.
* Network Infrastructure: Reliable and high-speed network connectivity to ensure seamless communication between clients and the server.

Software Requirements

* Web Application Framework: A suitable web application framework to develop the web-hosted application, such as Django, Ruby on Rails, or ASP.NET.
* Database Management System: A robust database management system (DBMS) to store and manage user data, appointment information, instructor details, and practice test data. Examples include MySQL, PostgreSQL, or MongoDB.
* Operating System: A compatible operating system for hosting the web application, such as Windows Server, Linux distributions like Ubuntu or CentOS, or cloud-based platforms like Amazon Web Services or Microsoft Azure.

Tools and Technologies

* Development Tools: Integrated Development Environment such as Visual Studio Code, PyCharm, or Eclipse for writing and debugging code.
* Version Control: A version control system like Git to manage source code and facilitate collaboration among developers.
* Automated Testing Framework: Testing frameworks like Selenium or PHPUnit to automate testing processes and ensure software quality.
* Payment Gateway Integration: Integration with a reliable payment gateway service provider to facilitate secure and smooth online payments.
* Encryption and Security Tools: Tools and libraries to implement secure data transmission (HTTPS) and data encryption to protect sensitive user information.
* Cloud Services: Utilize cloud-based platforms like AWS or Microsoft Azure for hosting the application, ensuring scalability, high availability, and automated backup and security features.

Infrastructure

* Server Infrastructure: Adequate server infrastructure to handle the expected user load, with scalable options for handling increased traffic during peak times.
* Data Backup and Disaster Recovery: Regular automated backups of the application and data to prevent data loss, along with a disaster recovery plan.
* Security Measures: Implementation of robust security measures, including firewalls, intrusion detection systems, and access control mechanisms, to protect against unauthorized access and attacks.