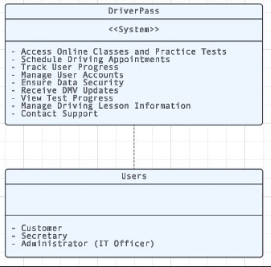
# CS 255 System Design Document Template

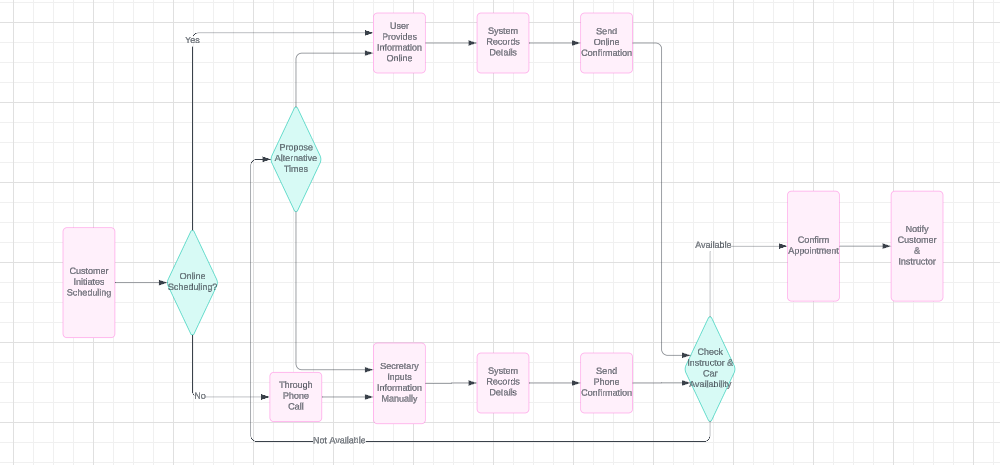
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length 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

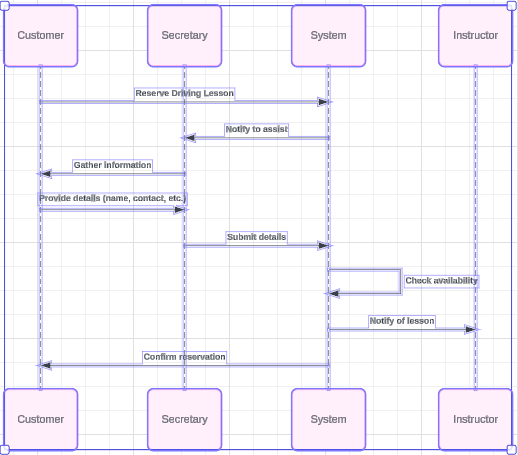
**

### UML Activity Diagrams

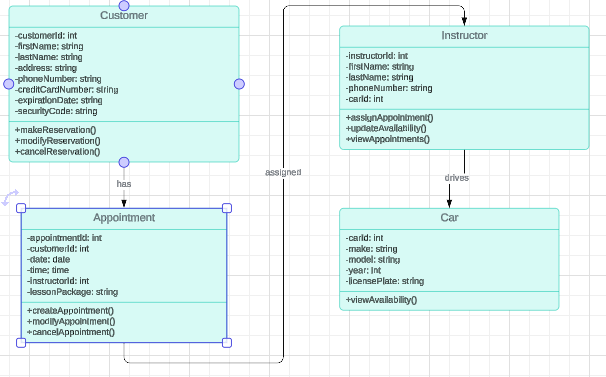
**

### 

### UML Sequence Diagram

**

### UML Class Diagram

**

## Technical Requirements

1. Hardware Requirements:
   * Server infrastructure to host the system components.
   * Sufficient computing resources (CPU, memory, storage) to handle concurrent user interactions and database transactions.
   * Networking equipment to ensure reliable communication between client devices and the server.
2. Software Requirements:
   * Operating System: The server environment should support a suitable operating system for hosting web applications, such as Linux (e.g., Ubuntu, CentOS) or Windows Server.
   * Web Server: A web server software (e.g., Apache, Nginx) to serve web pages and handle HTTP requests from clients.
   * Database Management System (DBMS): A relational database management system (e.g., MySQL, PostgreSQL) to store and manage data related to customers, appointments, instructors, and cars.
   * Programming Languages: Technologies such as HTML, CSS, JavaScript, and server-side scripting languages (e.g., PHP, Python, Node.js) for building the web-based user interface and backend logic.
   * Development Frameworks and Libraries: Frameworks and libraries (e.g., React.js, Angular, Bootstrap) to expedite frontend development and ensure a responsive and user-friendly interface.
   * Security Software: Implementation of security measures such as SSL/TLS certificates, firewalls, and intrusion detection/prevention systems to protect against unauthorized access and data breaches.
3. Tools:
   * Integrated Development Environment (IDE): Development environments like Visual Studio Code, IntelliJ IDEA, or Eclipse for coding, debugging, and version control.
   * UML Modeling Tool: Utilization of UML modeling tools like Lucidchart, draw.io, or Visual Paradigm for designing and documenting system architecture and diagrams.
   * Collaboration and Communication Tools: Platforms such as Slack, Microsoft Teams, or Zoom for team collaboration, communication, and project management.
4. Infrastructure Requirements:
   * Cloud Services: Utilization of cloud computing platforms (e.g., AWS, Azure, Google Cloud) for hosting the application, providing scalability, reliability, and easy maintenance.
   * Database Hosting: Deployment of the database on a cloud-based or self-managed server for data storage and management.
   * Backup and Disaster Recovery: Implementation of backup mechanisms and disaster recovery plans to ensure data integrity and availability in case of system failures or outages.