# 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 screenshot of a diagram

Description automatically generated

### UML Activity Diagrams

A screenshot of a diagram

Description automatically generated

A screenshot of a diagram

Description automatically generated

### UML Sequence DiagramA gantt chart with a diagram Description automatically generated

A diagram of a flowchart

Description automatically generated

### UML Class Diagram

This class diagram represents the core components of the DriverPass system, their attributes, and relationships. The User class serves as a parent to Customer and Employee. Lessons, reservations, and payments are associated with customers and drivers, ensuring an organized and efficient system design

*A white square with black text

Description automatically generated*

## Technical Requirements

***Technical Requirements***

***1. Hardware Requirements***

*The system requires reliable hardware for cloud hosting, data management, and end-user devices.*

* ***Servers****:*
  + *Cloud-based servers to host the system (e.g., Amazon Web Services (AWS), Microsoft Azure, or Google Cloud).*
  + *High availability and redundancy to prevent downtime.*
* ***End-User Devices****:*
  + *Standard devices such as desktops, laptops, smartphones, and tablets.*
  + *Devices must support modern web browsers.*
* ***Storage****:*
  + *Sufficient cloud storage to manage user data, lesson reservations, driver schedules, and payment records.*

***2. Software Requirements***

*The system will utilize modern web-based technologies for front-end, back-end, and database management.*

* ***Operating System****:*
  + *Cloud-hosted OS such as* ***Linux*** *for servers.*
* ***Web Development Tools****:*
  + ***Front-End****: React.js, Angular, or Vue.js for creating responsive web interfaces.*
  + ***Back-End****: Node.js, Python (Django/Flask), or Java (Spring Boot) for server-side logic.*
* ***Database Management System****:*
  + ***MySQL*** *or* ***PostgreSQL*** *for structured data management (e.g., customer accounts, reservations, and payments).*
  + *Support for* ***SQL queries*** *to ensure data accuracy and retrieval efficiency.*
* ***Browser Compatibility****:*
  + *The system must work on modern web browsers (Google Chrome, Firefox, Safari, Microsoft Edge).*

***3. Tools***

*The following tools will be used for system design, development, and testing:*

* ***System Design Tools****:*
  + ***Lucidchart****: For creating UML diagrams (use case, activity, sequence, and class diagrams).*
* ***Development Tools****:*
  + ***Visual Studio Code****: IDE for coding and debugging.*
  + ***GitHub****: Version control for collaboration and tracking changes.*
* ***Testing Tools****:*
  + ***Selenium****: Automated testing for front-end functionality.*
  + ***JUnit*** *or* ***pytest****: Unit testing for back-end logic.*
* ***Security Tools****:*
  + ***SSL/TLS Certificates****: To ensure secure data transfer.*
  + ***Role-Based Access Control****: Implemented via back-end logic.*
  + ***Password Encryption****: Hashing algorithms like bcrypt for secure password management.*

***4. Infrastructure Requirements***

*To ensure reliability, security, and performance, the following infrastructure is required:*

* ***Cloud Hosting****:*
  + *The system will be hosted on a cloud provider (e.g., AWS, Azure, or GCP) for scalability and minimal downtime.*
  + *Key services include compute instances, managed databases, and storage services.*
* ***Networking****:*
  + ***Internet Connectivity****: Reliable internet access for users to access the web application.*
  + ***Firewalls and VPNs****: To secure connections and protect against unauthorized access.*
* ***Backup and Recovery****:*
  + *Automated backup of data to ensure recovery in case of failure or data loss.*
* ***Notifications****:*
  + *Integration with an email or SMS API (e.g., Twilio, SendGrid) to notify customers of lesson confirmations and password resets.*

***Summary***

*The system leverages a* ***cloud-based architecture*** *to ensure flexibility, security, and scalability. It uses* ***modern web technologies*** *for front-end and back-end development, a* ***relational database*** *for structured data storage, and robust tools for design and testing. This infrastructure will meet DriverPass's functional requirements for reservation management, role-based security, and system performance.*