# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram

*A diagram of a company

Description automatically generated*

### UML Activity Diagrams

***Two*** *use cases and* ***two*** *activity diagrams, one for each use case.*

*A diagram of a test

Description automatically generated* A diagram of a work flow

Description automatically generated

### UML Sequence Diagram

A screenshot of a computer screen

Description automatically generated

### UML Class Diagram*A diagram of a computer Description automatically generated*

## Technical Requirements

*[Based on the diagrams here are the technical requirements of the system. These requirements address the required hardware, software, tools, and infrastructure necessary for the system design.]*

Hardware Requirements:

* Computers for administrative and development tasks with sufficient processing power, RAM, and storage.
* Mobile Devices: Smartphones or tablets for users to access the system remotely, requiring compatibility with iOS and Android platforms.

Software Requirements:

* Database Management System (DBMS): A cloud-compatible relational database such as Amazon RDS or Google Cloud SQL, which offers scalability and managed services.
* Application Software: Server-side technologies like Node.js or ASP.NET for backend services and React or Angular for the frontend to provide a responsive user interface.
* Security Software: SSL certificates for secure data transmission, along with antivirus and anti-malware software to protect the system.

Tools and Technologies:

* Development Tools: Integrated Development Environments (IDEs) like Visual Studio for coding, debugging, and testing.
* Version Control: Tools such as Git for source code management, hosted on platforms like GitHub to facilitate collaboration among developers.
* Project Management Tools: Applications like Jira or Trello to manage tasks, track bugs, and monitor project progress.
* Continuous Integration/Continuous Deployment (CI/CD) Tools: Jenkins or GitHub Actions to automate the testing and deployment of code changes.

Cloud Infrastructure Requirements:

* Cloud Hosting: Utilization of a third-party cloud service provider like AWS, Google Cloud, or Azure to host the entire application stack, ensuring scalability, reliability, and geographical redundancy. This includes hosting web servers, application servers, and databases.
* Backup Solutions: Cloud-based backup solutions provided by the cloud host to ensure data integrity and facilitate easy recovery in case of data loss.
* APIs: Integration with third-party APIs for additional functionalities such as DMV updates or payment processing (e.g., Stripe, PayPal) managed through secure cloud connections.
* Security Protocols: Implementation of HTTPS, OAuth for authentication, and role-based access control (RBAC) in the cloud environment to secure access to system resources.