# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram

*A diagram of a driver pass

Description automatically generated*

### UML Activity Diagrams

A screenshot of a computer

Description automatically generated

### UML Sequence Diagram

*A diagram of a payment system

Description automatically generated*

### UML Class Diagram

*A diagram of a software application

Description automatically generated with medium confidence*

## Technical Requirements

**Technical Requirements for DriverPass System**

**Hardware Requirements:**

The DriverPass system will require robust hardware to handle user traffic and application processing efficiently. The server should feature a multi-core processor, such as an Intel Xeon or AMD Ryzen, with a minimum of 16 GB of RAM, ensuring smooth multitasking and efficient data handling. Additionally, at least 500 GB of SSD storage will be necessary for rapid data access and storage of user applications, documents, and logs. To accommodate anticipated growth, scalability options should be considered, potentially involving additional storage or server instances.

**Software Requirements:**

For software, the system will operate on a Linux-based server environment (such as Ubuntu Server or CentOS) to provide a secure and stable foundation. The application will be developed using a backend technology like Node.js, Python (Django/Flask), or Java (Spring), while the frontend will utilize HTML, CSS, and JavaScript frameworks such as React or Angular to ensure a responsive and user-friendly interface. The backend will also require a relational database management system (DBMS) such as MySQL or PostgreSQL to manage user data and application records, with automated backup solutions for data protection. Additionally, the system will integrate with secure payment processing APIs like Stripe or PayPal for handling transactions.

**Tools:**

To develop and maintain the DriverPass system, various development tools will be utilized. Integrated Development Environments (IDEs) such as Visual Studio Code or IntelliJ IDEA will be used for coding and debugging. Version control systems like Git will manage source code, and platforms like GitHub or GitLab will host the repositories. For design and diagramming, tools like Lucidchart or Draw.io will assist in creating UML diagrams, while wireframing tools such as Figma will aid in designing the user interface. Automated testing frameworks, including Selenium for web applications and Postman for API testing, will ensure quality assurance.

**Infrastructure Requirements:**

The infrastructure must include a load balancer, such as HAProxy or AWS Elastic Load Balancing, to manage incoming user requests efficiently, ensuring availability and performance during peak times. A secure firewall should be deployed to protect the server from unauthorized access, while SSL/TLS certificates will secure data transmission between users and the server. Additionally, cloud services like Amazon Web Services (AWS) or Google Cloud can provide scalability and flexibility, offering resources like Database as a Service (DBaaS) for easy database management and cloud storage for user uploads.

**Summary:**

These technical requirements outline a comprehensive framework for building the DriverPass system, addressing the needs for hardware, software, tools, and infrastructure to ensure a reliable, efficient, and secure online platform for managing driver's license applications and renewals. The design aims to meet the client's objectives of reducing wait times, improving processing efficiency, and enhancing user accessibility, ultimately contributing to a more streamlined user experience.