# CS 255 System Design Document

## UML Diagrams

### UML Use Case Diagram

A diagram of a driver pass system

AI-generated content may be incorrect.

### UML Activity Diagrams

A diagram of a program

AI-generated content may be incorrect.A diagram of a payment process

AI-generated content may be incorrect.

### UML Sequence Diagram



### UML Class Diagram

A diagram of a user

AI-generated content may be incorrect.

## Technical Requirements

The DriverPass system requires a combination of hardware, software, and network infrastructure to support its user accounts, reservations, payment processing, and data management features. On the hardware side, users, including customers, administrators, and secretaries—can access the system using any modern desktop or mobile device with a web browser and internet connection. The backend infrastructure can be hosted on cloud-based servers to provide scalability, security, and continuous uptime without needing physical servers on-site.

The system’s software stack includes a web-based front end built with a responsive framework such as React, Angular, or Vue to allow users to register, log in, purchase packages, and schedule appointments. The backend operates through a secure API developed in Node.js, Java Spring, or Python, which handles business logic, authentication, and data transactions. The database layer, using PostgreSQL or MySQL, stores all user, reservation, payment, and report data, reflecting the relationships defined in the class diagram. A caching service like Redis can be used to improve performance for frequent queries and login sessions.

To support its payment and reporting features, the system integrates with third-party services such as Stripe or Braintree for secure payment processing, and email services like SendGrid or Amazon SES for notifications and confirmations. DMV updates can be handled through an API connection that securely exchanges data between the DriverPass system and the DMV’s system. All data transmissions use encryption (HTTPS/TLS), and user information is protected through role-based access control, password hashing, and optional multi-factor authentication.

The system’s infrastructure is deployed through cloud services such as AWS, Azure, or Google Cloud, using load balancers for traffic management and automated backups for disaster recovery. Continuous integration and deployment (CI/CD) tools ensure that updates can be tested and released efficiently, while monitoring and logging tools like CloudWatch or ELK Stack track performance and system errors. Overall, the combination of scalable cloud resources, secure integrations, and strong data management ensures that DriverPass can handle user activity, payments, and reporting reliably while maintaining high availability and compliance with security standards.