# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram

*A screenshot of a computer screen

Description automatically generated*

### UML Activity Diagrams

*A flowchart of a card

Description automatically generatedA flowchart of a cashier

Description automatically generated*

### UML Sequence Diagram

*A diagram of a bank

Description automatically generated*

### UML Class Diagram

*A diagram of a cash flow

Description automatically generated*

## Technical Requirements

The DriverPass system aims to enhance students' preparedness for driving tests by providing online practice exams, on-the-road training, and efficient management of reservations. This document outlines the prioritized technical requirements to ensure a successful development and implementation of the DriverPass system.

**Prioritized Requirements**

Essential:

* Server infrastructure capable of handling the system's expected load.
* High-speed and reliable internet connectivity.
* Secure network configurations for data protection.
* Secure web application built with a robust framework.
* Integration with the DMV's database for accurate information exchange.
* User-friendly and accessible web interface for online practice exams and reservation management.
* Secure authentication and authorization mechanisms for user access control.
* High-performance web application with responsive design.

Desirable:

* Mobile applications for on-the-go access to practice exams and reservation management.
* Scalable architecture to accommodate future growth and user base expansion.
* Real-time feedback on driving lessons (e.g., through instructor feedback or driving simulators).
* Integration with additional resources like e-books or video tutorials.
* Advanced reporting and analytics features for instructors and administrators.
* Multi-language support for accessibility across diverse user populations.

Future Considerations:

* Development of a driving test prediction model based on user performance data.
* Integration with smart driving glasses or other augmented reality technologies for enhanced learning experience.
* Integration with connected car technologies for real-time driving data analysis and feedback.

**System Architecture**

The DriverPass system will be implemented using a microservices architecture for scalability. Key components include:

* Web application: Built with a framework like Django or Express.js, providing online practice exams, reservation management, and user account functionalities.
* API gateway: Handles API calls from the web application and directs them to the appropriate microservices.
* Database: PostgreSQL or MySQL will store user data, practice exam content, and reservation information.
* DMV database integration: Secure connection established through RESTful APIs or GraphQL for real-time data exchange.
* Authentication and authorization service: Handles user login, token generation, and access control.

**Performance Metrics**

* Web application response times should not exceed 2 seconds.
* System uptime should be at least 99.5%.
* Practice exam loading times should not exceed 5 seconds.
* Reservation confirmation time should not exceed 10 seconds.

**Deployment Strategy**

The DriverPass system will be deployed in a staged manner:

* Development phase: System developed and tested in a controlled environment.
* QA testing: System undergoes rigorous testing by a dedicated QA team.
* Beta testing: System released to a limited group of users for real-world feedback.
* Production launch: System rolled out to all users upon successful beta testing.

**Security Considerations**

* Data encryption: All sensitive data will be encrypted at rest and in transit.
* User authentication: Two-factor authentication will be implemented for enhanced security.
* Vulnerability management: Regular system scans and updates will be conducted to mitigate security risks.
* Secure coding practices: Developers will follow secure coding practices to avoid vulnerabilities.

**Backup and Recovery Plan**

* Regular backups of the database and application data will be performed.
* Backups will be stored offsite in a secure location.
* Disaster recovery plan will be established to ensure swift system restoration in case of failures.

**UI/UX Design**

* The UI/UX will be user-friendly and intuitive, catering to users with varying technical expertise.
* The design will be responsive and accessible across various devices and browsers.
* WCAG guidelines will be followed to ensure accessibility for users with disabilities.

**Version Control and CI/CD Practices**

* Git will be used for version control.
* CI/CD pipeline will be implemented for automated code testing, integration, and deployment.
* Jenkins will be used for CI/CD automation.

**Testing Strategy**

* Unit testing will be performed on individual modules.
* Integration testing will ensure smooth interaction between components.
* Functional testing will verify system functionality against requirements.
* User acceptance testing will involve real users to evaluate system usability.

**Monitoring and Logging**

* System performance will be monitored for metrics like response times and resource utilization.
* User activity will be logged for troubleshooting and analysis.
* Monitoring tools like Grafana or Prometheus will be used for visualization and analysis.

**Conclusion**

The DriverPass System Technical Requirements Document provides a comprehensive roadmap for the development and implementation of a robust and user-friendly driving education platform. By prioritizing essential functionalities, outlining a scalable architecture, and addressing key considerations like performance, security, and user experience, the document sets a solid foundation for project success. Implementing the recommendations outlined in this document, along with continuous monitoring, adaptation, and stakeholder engagement, will ensure the DriverPass system delivers a transformative learning experience for aspiring drivers.