# CS 255 System Design Document

Shane Holbrook

CS255

Dr.Stefanelli

## UML Diagrams

### UML Use Case Diagram

A diagram of a company

Description automatically generated

### UML Activity Diagrams

*A diagram of a software company

Description automatically generatedA diagram of a flowchart

Description automatically generated*

### UML Sequence Diagram

A diagram of a person's work flow

Description automatically generated

### UML Class Diagram

A diagram of a computer

Description automatically generated

## Technical Requirements

Hardware Requirements:

1. Server Infrastructure:

- Multiple servers for distributing load (e.g., web server, database server).

- Adequate processing power and memory to handle concurrent user requests.

2. Database System:

- Database server with sufficient storage capacity.

- Backup and recovery mechanisms to ensure data integrity.

3. Networking:

- High-speed internet connectivity to facilitate real-time interactions.

- Secure network protocols to protect data during transmission.

4. Client Devices:

- Compatibility with a range of devices (PCs, laptops, tablets, smartphones).

- Minimum system requirements for client devices to run the application smoothly.

Software Requirements:

5. Operating System:

- Compatibility with popular operating systems (Windows, macOS, Linux).

- Server operating system (e.g., Linux distributions) for hosting backend components.

6. Web Server:

- Use of a reliable web server (e.g., Apache, Nginx) to serve web pages.

- Support for HTTPS to ensure secure communication.

7. Database Management System:

- Integration with a robust relational database management system (e.g., MySQL, PostgreSQL).

- Database schema design to efficiently store and retrieve data.

8. Programming Languages:

- Selection of programming languages for backend (e.g., Python, Java) and frontend (e.g., JavaScript, HTML, CSS) development.

- Compatibility with frameworks and libraries used for development.

9. Version Control:

- Implementation of version control systems (e.g., Git) for tracking and managing code changes.

10. Security Measures:

- Encryption of sensitive data (both in transit and at rest).

- Secure user authentication and authorization mechanisms.

- Regular security audits and updates to address vulnerabilities.

Performance Requirements:

11. Response Time:

- Defined response time for critical user actions (e.g., login, reservation).

- Minimization of latency for real-time interactions.

12. Scalability:

- Scalable architecture to handle a growing user base.

- Load balancing mechanisms to distribute traffic efficiently.

Functional Requirements:

13. User Interfaces:

- Intuitive and user-friendly interfaces for both administrators and end users.

- Compatibility with different web browsers.

14. Functionality:

- Implementation of all specified features (e.g., account management, lesson reservation, testing).

- Error handling and graceful degradation in case of system failures.

Integration Requirements:

15. Third-Party Integrations:

- Integration with external services (e.g., DMV systems) for data synchronization.

- Compatibility with industry standards for seamless integration.

Compliance and Reporting:

16. Compliance:

- Adherence to relevant industry standards and regulations (e.g., data protection regulations).

- Generation of compliance reports.

17. Logging and Auditing:

- Logging of system activities for troubleshooting and auditing purposes.

- Generation of detailed reports for system administrators.

These technical requirements provide a foundation for the development and deployment of the DriverPass system, ensuring that it meets performance expectations, security standards, and user needs. Adjustments may be made based on the specific technology stack chosen for implementation and any unique project constraints.