# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram

*A screenshot of a computer screen

AI-generated content may be incorrect.*

### UML Activity Diagrams

A screenshot of a diagram

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

### UML Sequence Diagram

*A diagram of a driving application

AI-generated content may be incorrect.*

### UML Class Diagram

*A diagram of a computer

AI-generated content may be incorrect.*

## Technical Requirements

The technical requirements of the DriverPass system are derived from the system’s functional demands, user roles, and operational expectations as articulated in the business requirements and validated through multiple UML design artifacts. These requirements encompass the software, hardware, tools, and infrastructure necessary to implement a secure, scalable, and user-friendly solution that serves customers, instructors, secretaries, IT staff, and administrators.

**Software Requirements**

To fulfill the system behaviors illustrated in the use case and activity diagrams, such as scheduling appointments, taking practice tests, and managing lesson packages—the system must operate on a stack of reliable software platforms.

* Backend Platform: The application logic will be developed using server-side language such as Java or Python, offering support for modular architectures and secure session handling.
* Frontend Framework: A responsive web interface should be implemented using HTML5, CSS3, and JavaScript, with React.js recommended for its component-based structure and strong integration with RESTful services.
* Database Management System: The system will use MySQL, a widely supported relational database, to manage structured data for students, appointments, vehicles, instructors, and test results. The schema will reflect the class relationships defined in the UML class diagram, such as one-to-many mappings between students and appointments or students and practice tests.
* Authentication & Security Libraries: Secure login, session management, and password encryption will be enforced using OAuth 2.0, JWT (JSON Web Tokens), and bcrypt hashing algorithms to comply with nonfunctional requirements related to system security.
* Cloud Integration Tools: The backend will integrate with AWS SES or similar tools for email confirmation notifications and utilize AWS S3 for optional cloud file storage if downloadable reports are needed.

**Hardware Requirements**

While DriverPass will operate as a cloud-hosted system with minimal physical hardware dependencies, certain endpoints and back-end specifications must still be supported.

* Client Devices: The system must support desktop and mobile devices running:  
    
  + Windows 10 or later
  + macOS Monterey or later
  + iOS 14+
  + Android 10+
* Minimum Server Configuration (for cloud deployment):  
    
  + 4 vCPU cores
  + 8 GB RAM
  + 100 GB SSD storage
  + Hosted through Amazon Web Services (EC2) or Microsoft Azure Virtual Machines

These specifications ensure that multi-user concurrency (e.g., over 100 users), high availability, and responsiveness goals outlined in the performance requirements are consistently met.

**Infrastructure Requirements**

The DriverPass system will operate on a cloud-first infrastructure to ensure scalability, maintainability, and disaster recovery.

* Cloud Hosting: The system will be deployed using a Platform-as-a-Service (PaaS) model, such as AWS Elastic Beanstalk or Heroku, to minimize manual server configuration and ensure automated load balancing.
* Database Hosting: A managed Relational Database Service instance (e.g., Amazon RDS for MySQL) will ensure secure and performant database access with automatic backups and encryption at rest.
* Network Security:  
    
  + All communication will occur over HTTPS.
  + The infrastructure will use firewall rules, role-based access control, and intrusion detection systems for administrative and IT staff interactions.
* Backup and Recovery: Daily snapshots and off-site backups will be enforced with automated recovery protocols.
* Monitoring and Logging:  
  + System activity and audit logs (e.g., user login, appointment creation, package changes) will be maintained using AWS CloudWatch, Splunk, or an equivalent centralized logging system.

**Development and Modeling Tools**

While planning and building the system, the following tools are used:

* Lucidchart: For creating UML diagrams (use case, activity, sequence, and class) to visualize system behavior and structure
* Visual Studio Code / IntelliJ IDEA: As the recommended integrated development environments (IDEs) for full-stack development
* Postman: For testing RESTful API endpoints
* Git & GitHub: For source control, versioning, and collaborative development

**Summary**

These technical requirements are aligned with the functional and nonfunctional system goals presented in the Project One documentation and supported through comprehensive UML modeling. By adopting a cloud-native, modular architecture with robust user interface design and secure backend infrastructure, the DriverPass system will be equipped to support all user roles effectively while remaining scalable and adaptable for future enhancements.