# CS 255 System Design Document Template

# DriverPass System

This document presents UML diagrams and technical requirements for DriverPass using the same practices in Module Six.

## UML Use Case Diagram

Actors: Student, Instructor, Admin;   
Externals: DMV System, Payment Gateway.   
Key actions include registration, login, purchasing packages, scheduling/rescheduling lessons, taking practice tests, viewing progress, managing availability/students, and receiving notifications.

A diagram of a diagram

AI-generated content may be incorrect.

## UML Activity Diagrams

### Activity #1: Schedule Lesson

Flow emphasizes checks similar to ATM logic: verify active package, handle slot availability and conflicts, policy checks, payment authorization (if needed), then commit the booking and notify parties.

A diagram of a flowchart

AI-generated content may be incorrect.

### Activity #2: Take Practice Test

Flow uses randomized questions, optional timer, scoring with feedback, and a retry loop to allow another attempt before storing results and recommending next steps.

A diagram of a student process

AI-generated content may be incorrect.

## UML Sequence Diagram: Schedule Lesson

Sequence shows Student → App → Scheduling Service → Payment Gateway → Instructor interactions: reserve slot (hold), authorize payment if needed, then confirm or release the hold. Notifications are sent on success.

A diagram of a diagram

AI-generated content may be incorrect.

## UML Class Diagram

Domain classes include users and roles, package purchases, payment transactions, lessons and bookings (with holds), vehicles, practice tests with questions, and notifications.

A black and white diagram

AI-generated content may be incorrect.

# Technical Requirements

Client: Responsive web app (React).   
Server: REST API (Java Spring Boot or Node.js/Express) with RBAC auth and JWT sessions.   
Database: Managed relational DB (PostgreSQL/MySQL) with encryption at rest and PITR backups.   
Integrations: Stripe (tokenized payments), DMV policy lookups (HTTPS).   
Scheduling: Availability, conflict detection, and holds for bookings with automatic expiry.   
Practice Tests: Question bank, randomized selection, scoring, and feedback storage.   
Notifications: Email/SMS providers (SendGrid/Twilio) with retry and audit logs.   
Hosting: Cloud (AWS/Azure/GCP) with load balancer/API gateway, containers or serverless, CDN for static assets.   
Security: TLS 1.2+, HSTS, CSRF/XSS/SQLi protections, least-privilege IAM, WAF/DDoS protections, admin MFA.   
Observability: Centralized logs, metrics, tracing (OpenTelemetry).   
Performance: ≥99.9% availability target, caching of read-mostly content, async queues for webhooks/notifications.   
Compliance & Privacy: Minimize PII, data retention policy, audit logs.   
CI/CD & Testing: Automated tests, contract tests for externals, blue/green or canary deploys.