# CS 255 – System Design Document

# Project Two

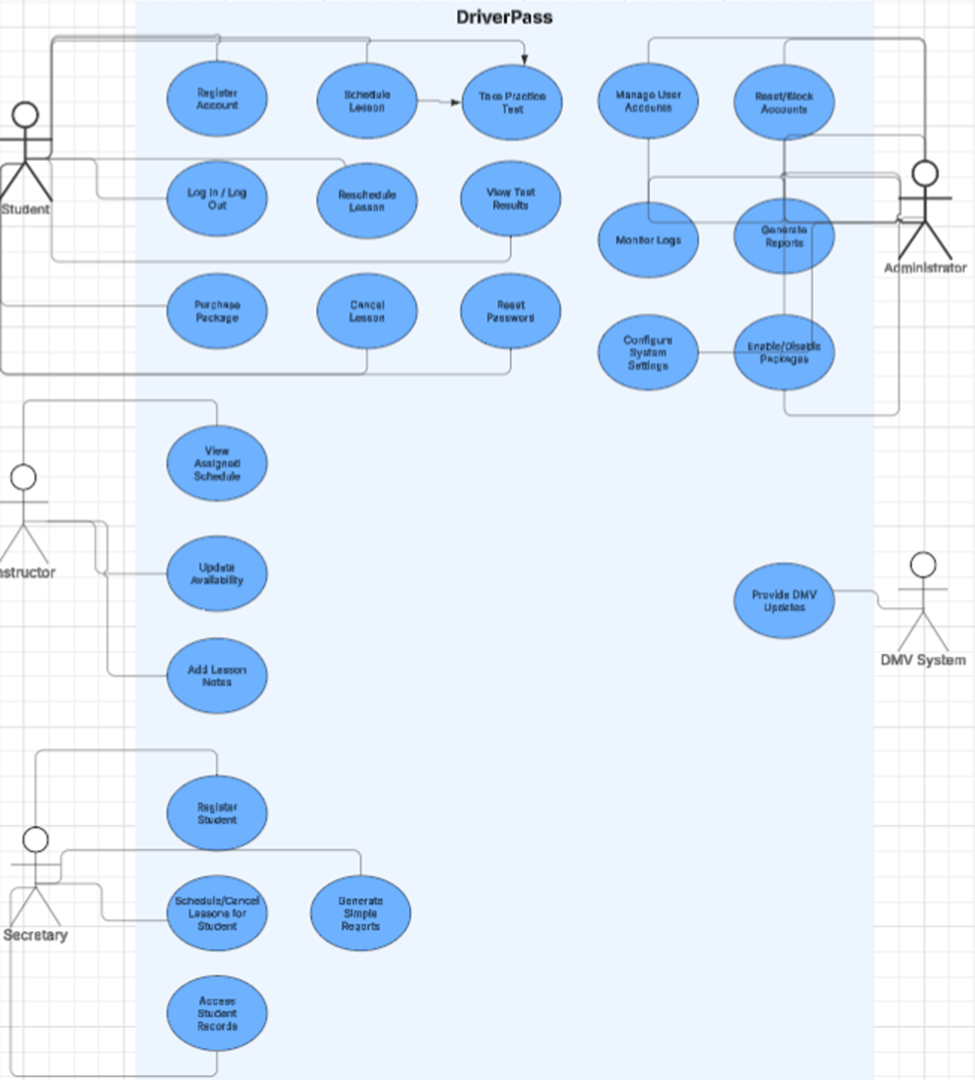
# Ian Brown

# Southern New Hampshire University

# 10/18/2025

## UML Diagrams

### UML Use Case Diagram



### UML Activity Diagrams

**Activity Diagram 1: Schedule Lesson, Activity Diagram 2: Take Practice Test**

A diagram of a diagram

AI-generated content may be incorrect.**A diagram of a company

AI-generated content may be incorrect.**

### UML Sequence Diagram

A white sheet with black text

AI-generated content may be incorrect.

### UML Class DiagramA screenshot of a computer AI-generated content may be incorrect.

## Technical Requirements

**Hardware and Infrastructure**

* **Cloud Hosting:** The system will be deployed on cloud-based servers (Windows or Linux) to ensure scalability, reliability, and reduced on-site IT maintenance.
* **Availability:** Target uptime of 99% with redundancy across multiple availability zones.
* **Storage and Backup:** Relational database storage for structured data (users, lessons, payments, exams) and object storage for media (video, audio). Automated nightly backups with monthly restore testing.
* **Networking:** Load balancer and firewall protection, with SSL/TLS termination for secure traffic.

**Software and Tools**

* **Client Application:** Web-based application accessible via modern browsers (Chrome, Edge, Safari, Firefox). Responsive design ensures compatibility with desktops, tablets, and smartphones.
* **Server Application:** Backend services built with a scalable web framework, exposing RESTful APIs for authentication, scheduling, payments, and reporting.
* **Database:** Relational Database Management System (MySQL, PostgreSQL, or SQL Server) to support ACID transactions for reservations, payments, and exam results.
* **Payment Integration:** Secure integration with third-party payment gateways (e.g., Stripe, PayPal) for processing transactions.
* **Notification Services:** Email and SMS integration for confirmations, reminders, and alerts.
* **Development Tools:** UML modeling with Lucidchart, version control with Git, and CI/CD pipelines for automated testing and deployment.

**Security**

* **Authentication:** Unique username/email with case-sensitive passwords.
* **Password Storage:** Secure hashing and salting of credentials.
* **Encryption:** SSL/TLS for all client–server communication; encryption at rest for sensitive data.
* **Access Control:** Role-based access (Student, Instructor, Secretary, Administrator) with least-privilege permissions.
* **Brute Force Protection:** Account lockout after five failed login attempts, with administrator alerts.
* **Account Recovery:** Self-service password reset via email verification, with audit logging.

**Performance and Reliability**

* **Response Time:** Core actions (login, scheduling, test submission) must complete in under 2 seconds under normal load.
* **Concurrency:** System must support multiple concurrent users (students, instructors, administrators) without degradation.
* **Maintenance:** Monthly updates for features and bug fixes; emergency patches applied as needed. Scheduled maintenance will occur during off-peak hours.

**Monitoring and Reporting**

* **System Monitoring:** Centralized logging and alerting for performance, errors, and security events.
* **Reporting:** Administrative dashboards and exportable reports (e.g., Excel) for reservations, cancellations, payments, and user activity.
* **Audit Trails:** All user actions (create, modify, cancel) logged with timestamps and user IDs for traceability.