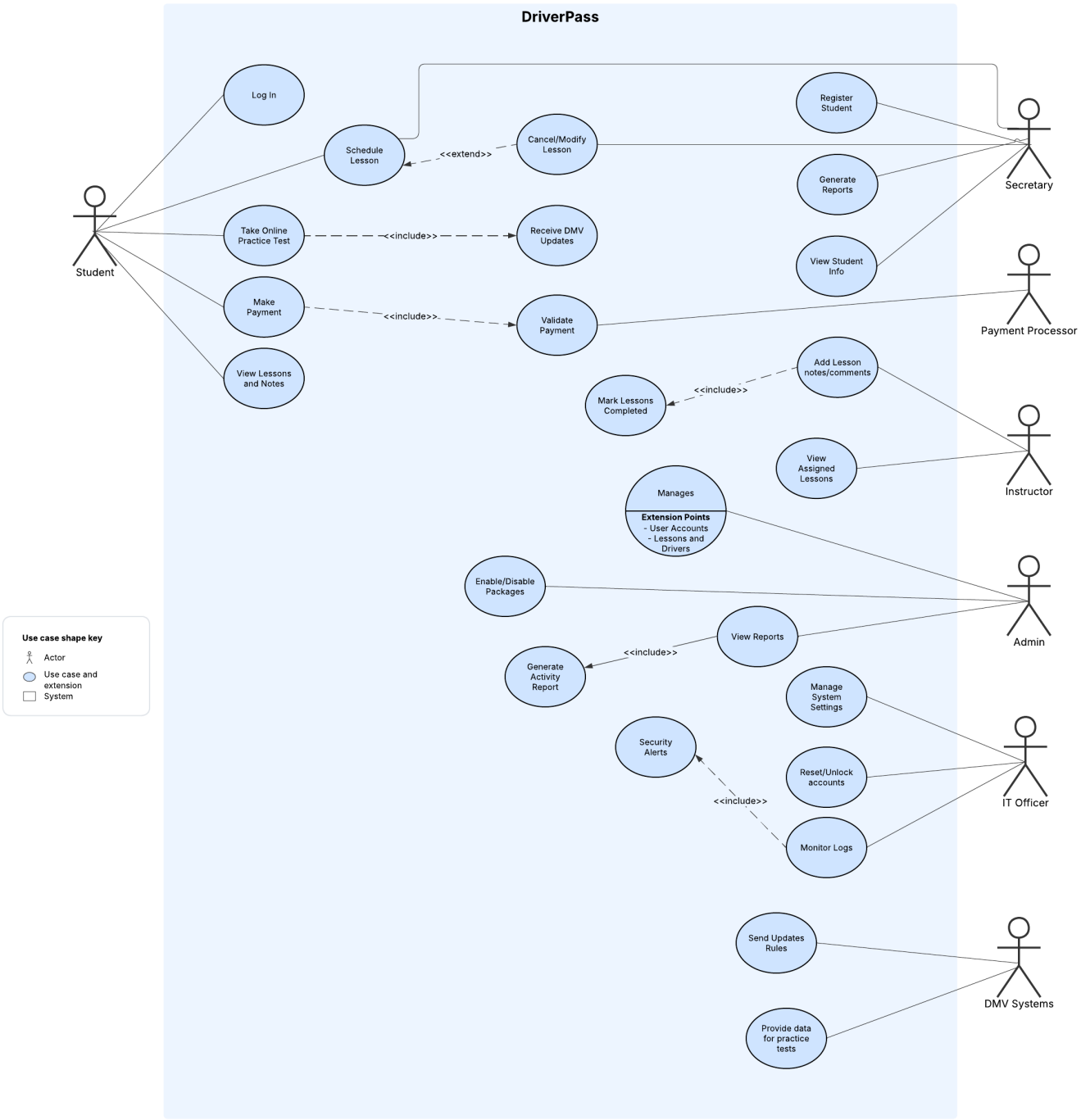
Southern New Hampshire University

CS 255 System Design Document Template

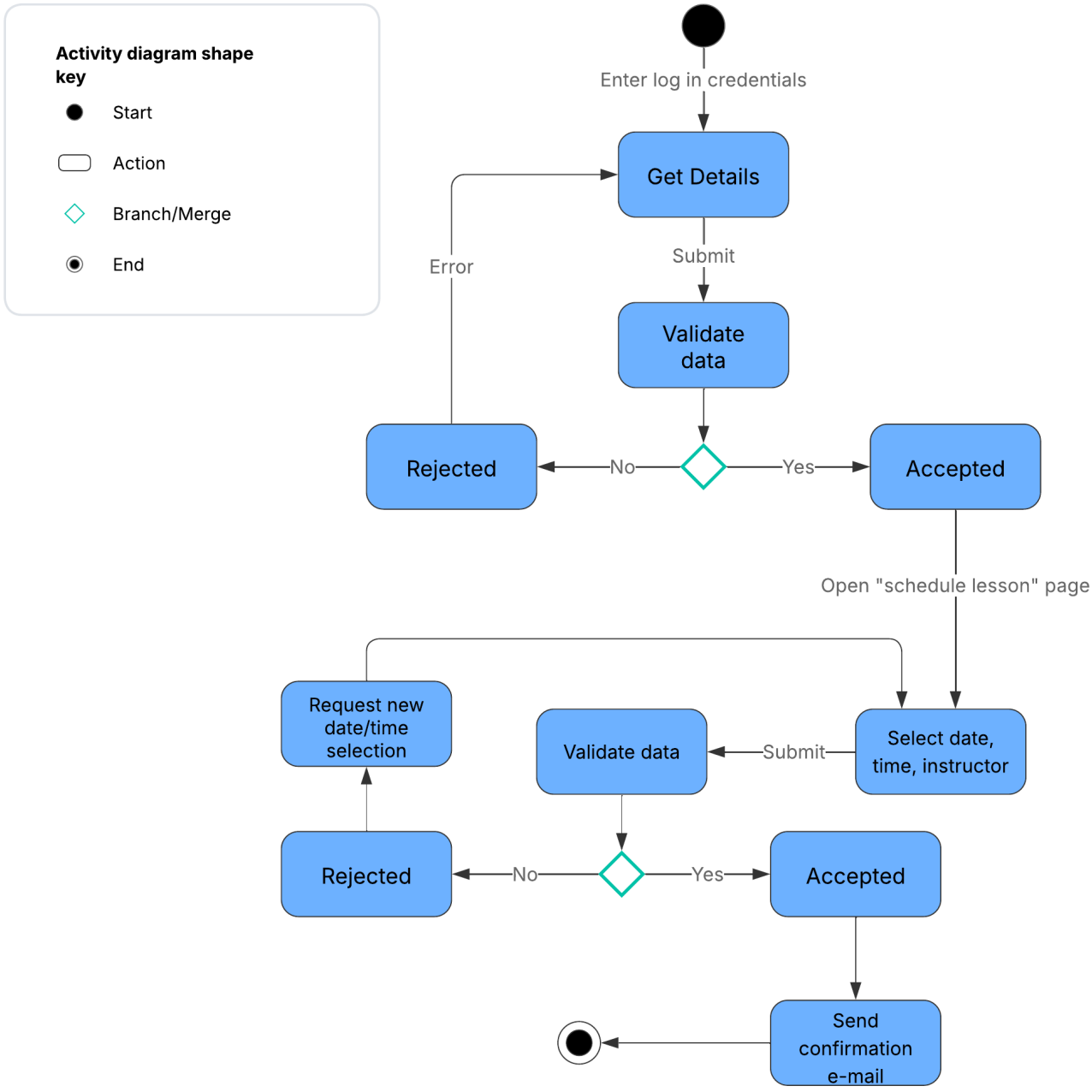
Jorge Fernando Moreno Jacob

UML Diagrams

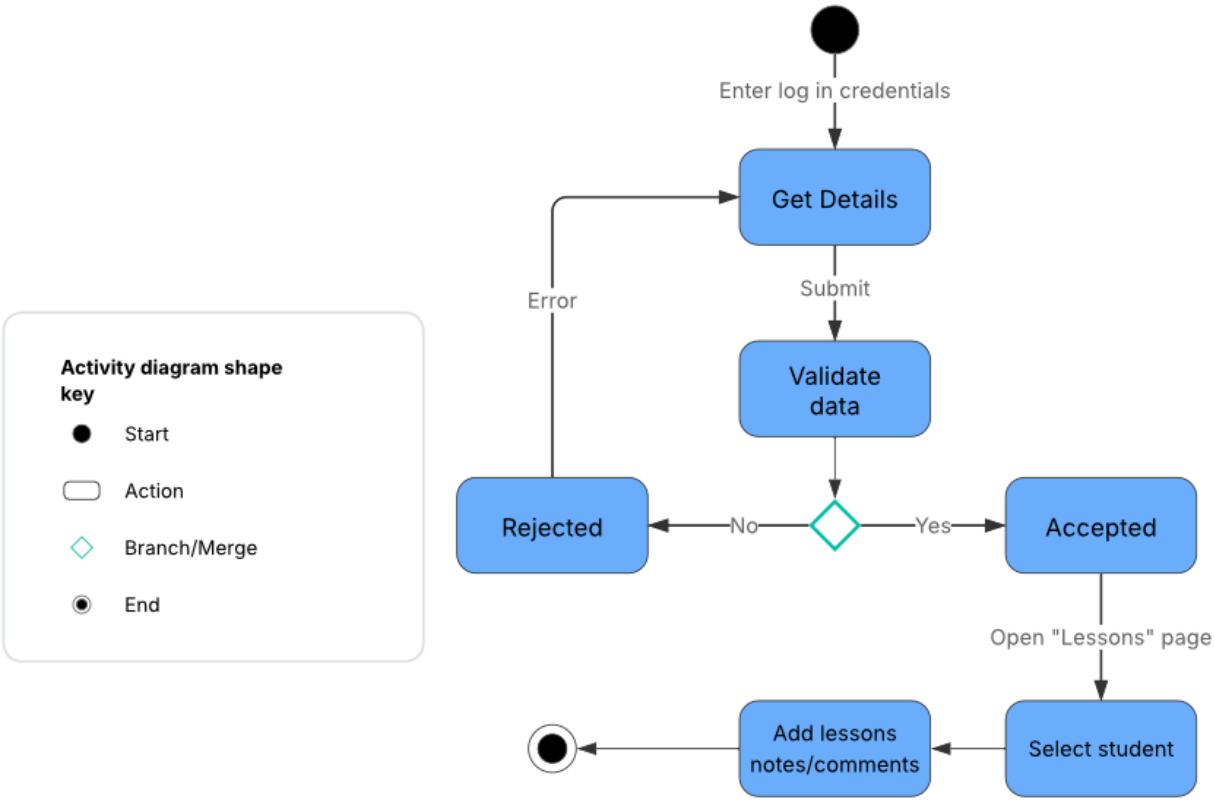
UML Use Case Diagram



UML Activity Diagrams

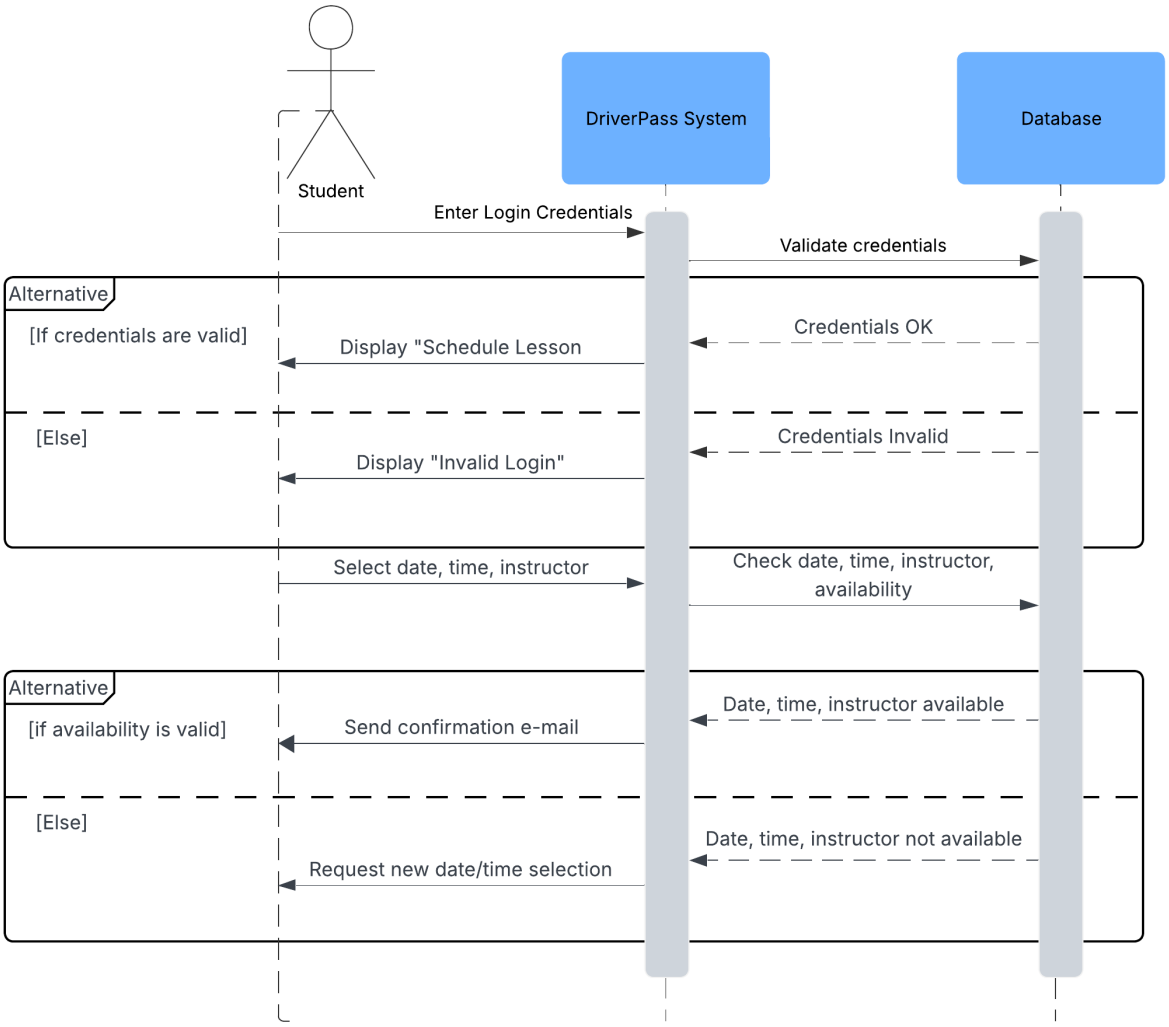


Student: Schedule Lesson UML Activity Diagram

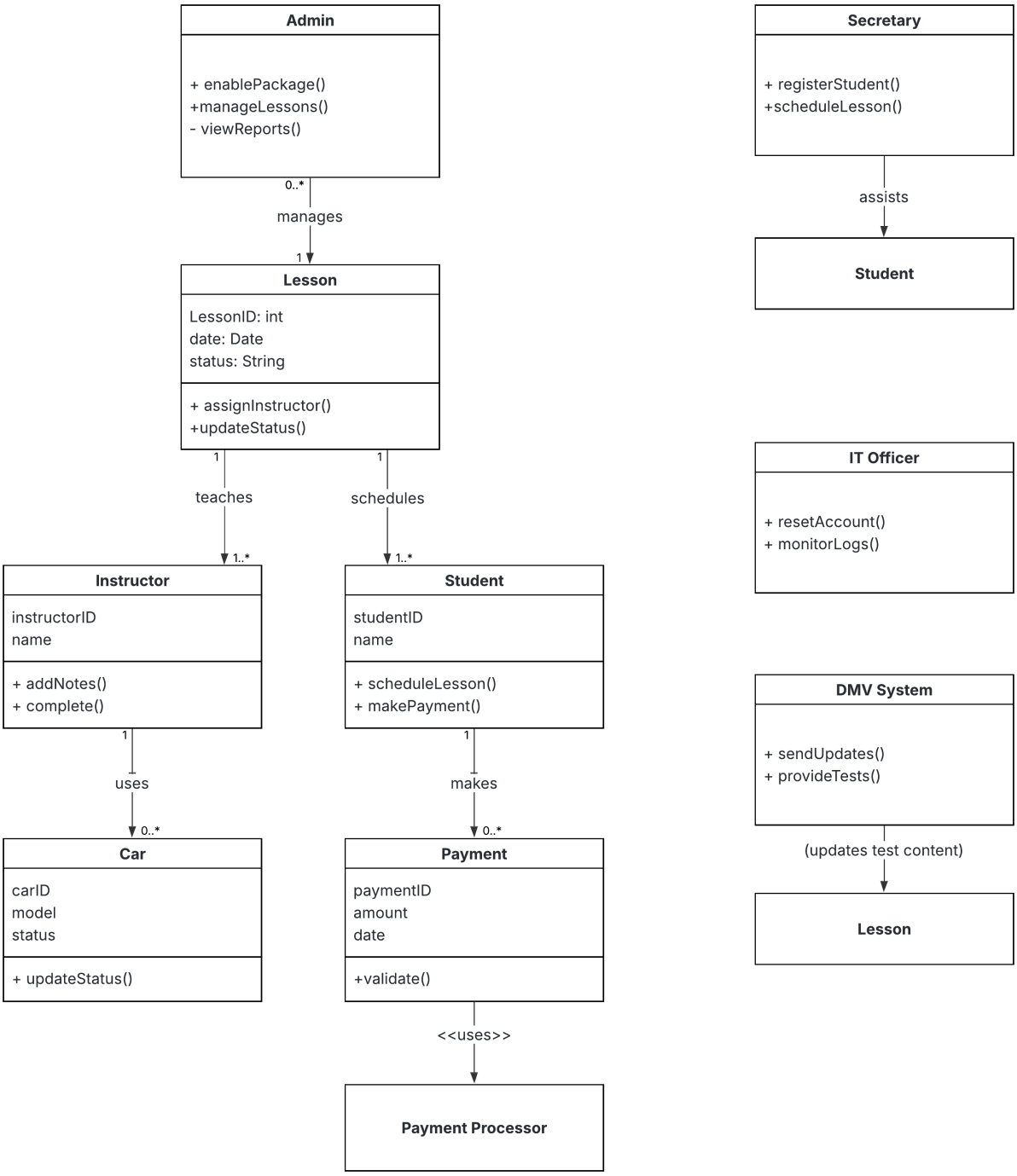


Instructor: Add Lesson notes/comments UML Activity Diagram

UML Sequence Diagram



UML Class Diagram



Technical Requirements

The DriverPass system will operate as a secure, cloud-based web application that supports online scheduling, lesson tracking, and testing for students, instructors, and administrators. The system must integrate with external services, including a payment processor and the DMV database, while ensuring reliability, scalability, and data protection.

Hardware Requirements

DriverPass will be hosted on a cloud-based server infrastructure (such as AWS or Microsoft Azure) with scalable resources to accommodate multiple concurrent users. The server will require at least an 8-core CPU, 16 GB RAM, and 500 GB of SSD storage to support the application, database, and backup services. Local users such as administrators, secretaries, and instructors will access the system using standard computers or mobile devices with stable internet connections and updated web browsers. Network reliability is essential to ensure lesson scheduling, payment processing, and DMV data synchronization.

Software Requirements

The system will be implemented as a web-based platform using a modern multi-tier architecture:

- **Frontend:** HTML5, CSS3, and Angular or React for dynamic and responsive user interfaces.
- **Backend:** Java with Spring Boot or Python with Django for processing logic and API handling.

- **Database:** MySQL or PostgreSQL relational database to store user information, lesson details, and payment transactions.
- **Operating System:** The servers will run on **Linux** or **Windows Server**, depending on hosting configuration.
- **APIs:** Secure RESTful APIs will facilitate communication between DriverPass and external systems such as the DMV and payment processor.

Tools and Development Environment

The development process will rely on industry-standard tools. Lucidchart will be used to design UML diagrams including use case, class, and sequence diagrams. Visual Studio Code or IntelliJ IDEA will serve as the primary IDEs for coding. Version control will be managed through GitHub or GitLab repositories. Testing will be conducted using JUnit for backend components and Selenium for automated UI testing. Continuous integration and deployment (CI/CD) pipelines will ensure frequent, stable updates during development.

Infrastructure and Security

DriverPass will be deployed in a secure cloud environment to ensure data availability and resilience. HTTPS encryption will be used for all communications between clients and servers. The system will implement role-based access control (RBAC) to restrict permissions based on user roles (Student, Instructor, Secretary, Administrator, and IT Officer). Sensitive information such as passwords and payment details will be encrypted both in transit and at rest. Automated system logs will record all user actions and system events for auditing and error tracking. Regular database backups and server snapshots will be maintained for disaster recovery. The IT

Officer will monitor security alerts and system logs to ensure compliance with data protection standards.