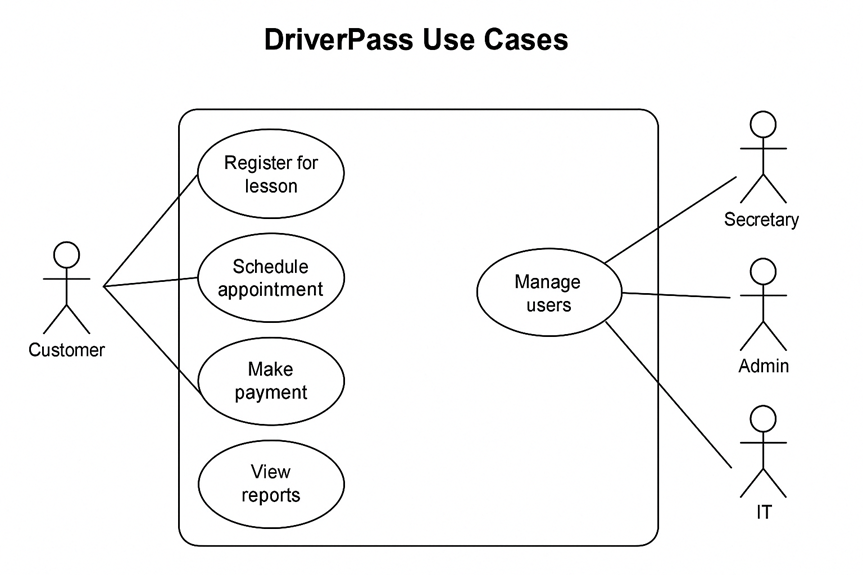
DriverPass System Design Document

# UML Use Case Diagram

The use case diagram below outlines the interactions between different types of users (Customer, Secretary, Admin, and IT) and the DriverPass system. This includes actions such as registering for driving lessons, scheduling appointments, making payments, and managing user accounts.



# UML Activity Diagrams

Two activity diagrams were created to represent the workflows for:  
1. Registering for a driving lesson  
2. Making a payment  
  
These diagrams provide a step-by-step visualization of how users interact with the system to complete these tasks.

A diagram of a payment process

AI-generated content may be incorrect.

# UML Sequence Diagram

The sequence diagram below details the process and message flow when a customer registers for a driving lesson, including system validations, schedule confirmation, and user feedback.

A diagram of a system

AI-generated content may be incorrect.

# UML Class Diagram

This class diagram outlines the main entities in the DriverPass system, including User, Lesson, Appointment, Instructor, TestResult, and TrainingPackage. Each class includes relevant attributes, and relationships are shown to define associations, such as users being linked to appointments or instructors.

A diagram of a user

AI-generated content may be incorrect.

# Technical Requirements

Based on the system’s needs and the diagrams provided, the technical requirements are as follows:  
  
\*\*Hardware Requirements\*\*  
- Cloud-hosted environment with scalability and backup support  
- Reliable internet connection for web application access  
  
\*\*Software Requirements\*\*  
- Web server technologies (e.g., Node.js, Python Flask, or Java Spring Boot)  
- Frontend in HTML5/CSS/JavaScript (React or Angular optional)  
- Relational database (PostgreSQL or MySQL)  
- Lucidchart or draw.io for UML diagram creation  
  
\*\*Tools\*\*  
- Git for version control  
- IDE (Visual Studio Code, IntelliJ, etc.)  
- CI/CD pipeline tools for deployment (GitHub Actions, Jenkins)  
  
\*\*Infrastructure\*\*  
- Hosting via AWS, Azure, or Google Cloud  
- Secure login mechanisms and database encryption  
  
\*\*Security\*\*  
- Encrypted password storage and authentication via HTTPS  
- Role-based access controls (RBAC)  
- Account lockout after multiple failed logins  
- Admin tools for account recovery and session logging