Southern New Hampshire University

CS 255 System Design Document Template

UML Diagrams

UML Use Case Diagram

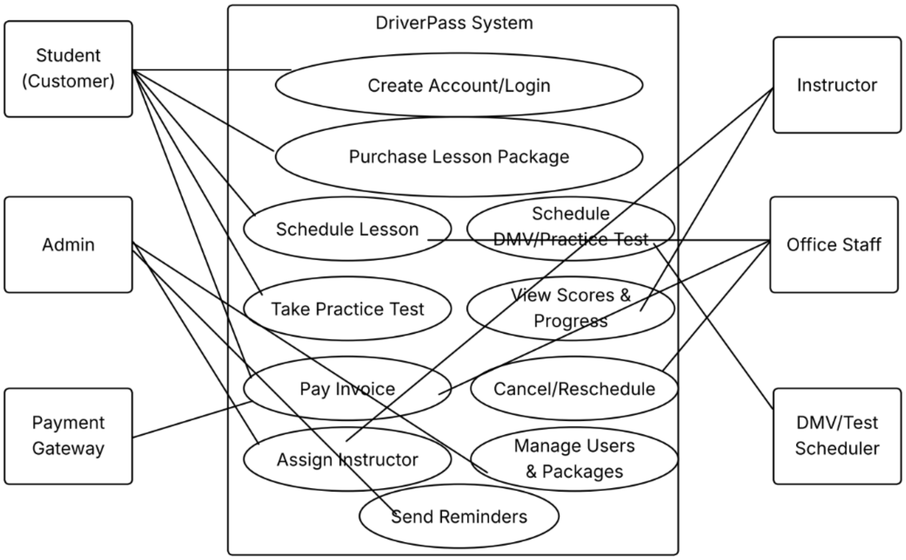
For the DriverPass system, I created a UML use case diagram that outlines how the main users interact with the system based on both the interview transcript and the business requirements document from Project One. The goal of this diagram is to show the core system functions that support customer training, scheduling, payments, and reporting, while also highlighting the external systems that interact with DriverPass.

The primary actors in the system include:

- *Student(Customer): Registers an account, purchases lesson packages, schedules or cancels lessons, takes practice tests, views progress, and makes payments online.*
- *Instructor: Reviews assigned students, logs lesson times, and records notes or feedback about each students progress.*
- *Office Staff: Manages scheduling, modifies or cancels customer appointments, assigns instructors, and oversees scheduling across all vehicles and trainers.*
- *Admin/IT Staff: Maintains user accounts, manages packages, assigns instructors, controls system access, and handles system updates and notifications.*

External Systems:

- *Payment Gateway: Processes all customer payments securely.*
- *DMV/Test Scheduler: Sends policy updates and test information to keep DriverPass compliant with DMV requirements.*



Use Cases

The main use cases for the DriverPass system are:

Create Account/Login, Purchase Lesson Package, Schedule Lesson, Schedule DMV/Practice Test, Take Practice Tests, View Scores and Progress, Pay Invoice, Cancel or Reschedule Lessons, Assign Instructor, Manage Users and Packages, and Send Reminders.

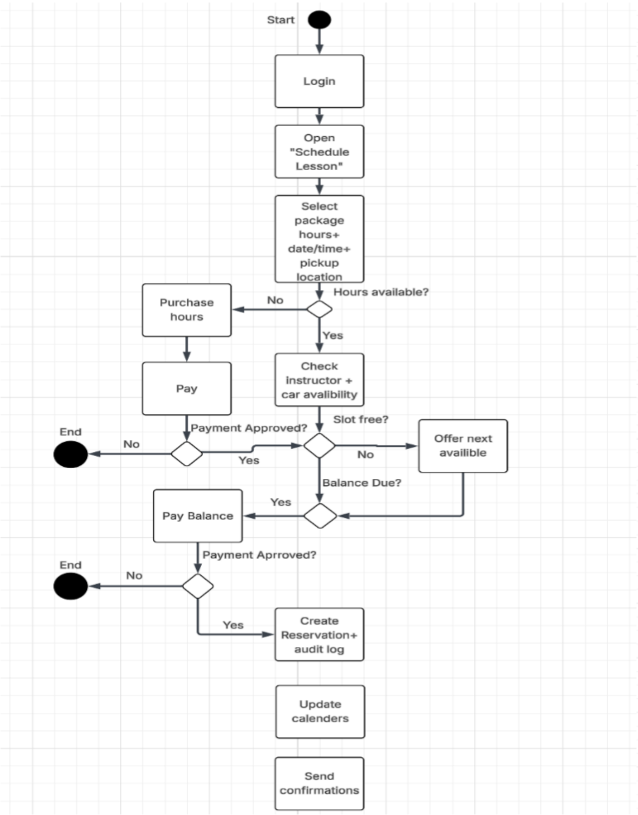
Each of these use cases was identified directly from the client's needs described by Liam in the interview. For example, Liam wanted the ability for customers to book and modify lessons, take online practice tests, and access their data securely from any device. He also emphasized the need for reporting and tracking so that DriverPass can see progress for both students and instructors.

The diagram captures how each user interacts with system in a simple, organized way. Students mainly handle their own scheduling and test preparation. Instructors focus on lesson tracking and notes. The office staff and admin users control scheduling, system access, and customer management, while the DMV and payment systems work externally to keep the system up to date and secure.

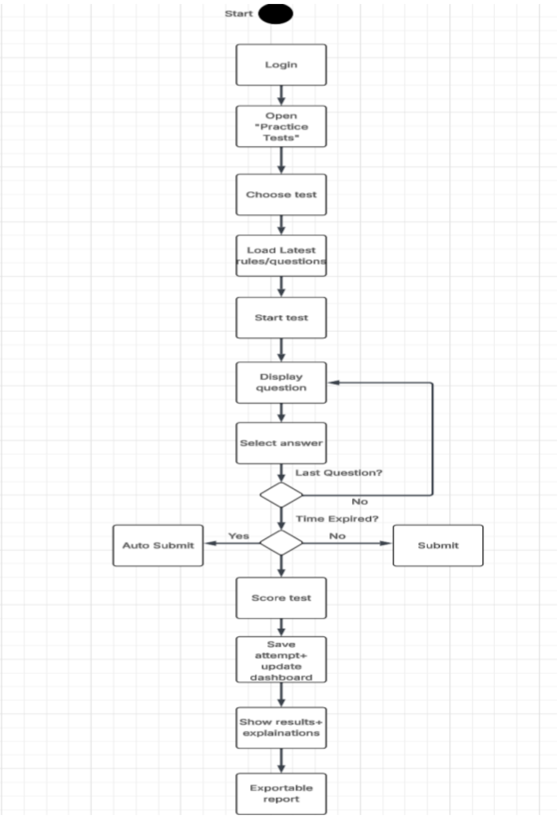
Overall, this diagram shows how DriverPass connects users, staff, and external systems in one platform that supports flexible scheduling, online learning, secure payments, and detailed progress reporting, all of which meet the client's vision of improving driving test pass rates and training efficiency.

UML Activity Diagrams

Schedule a driving lesson

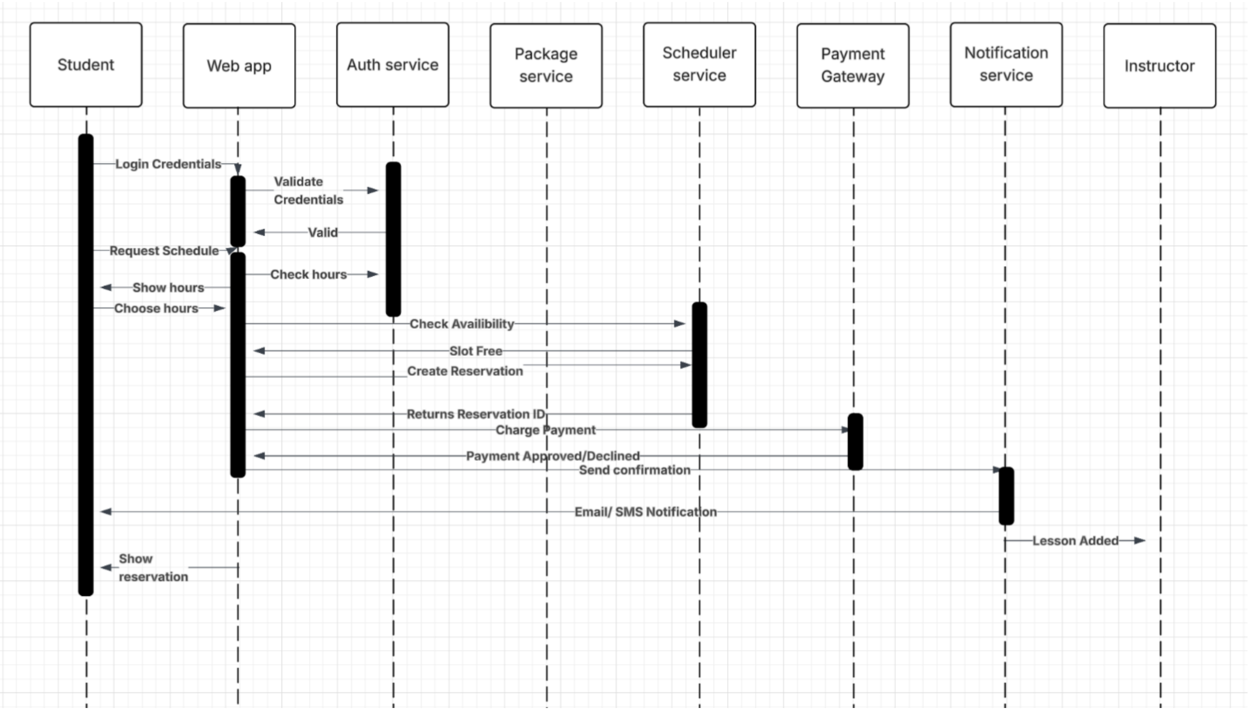


Take a practice test

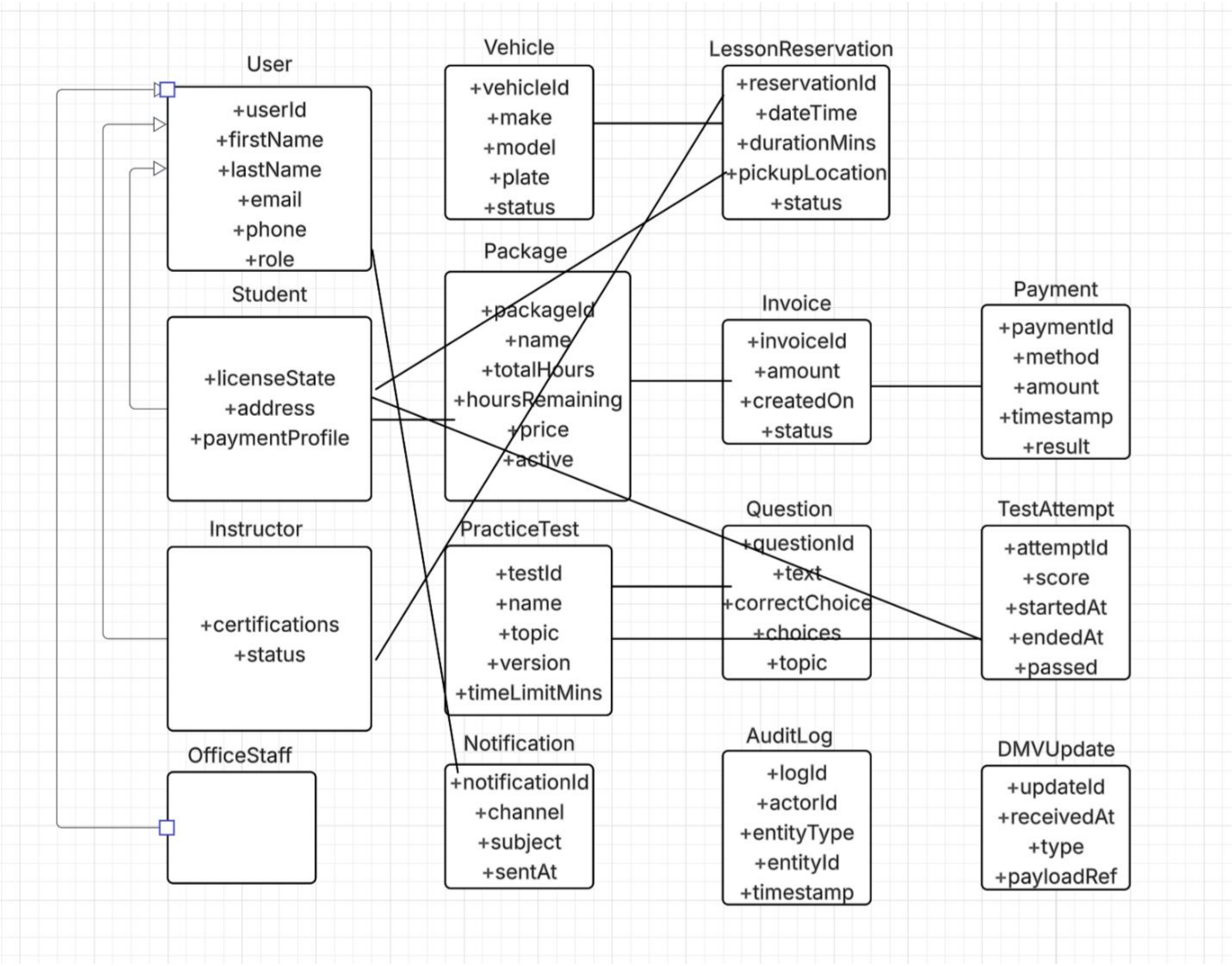


UML Sequence Diagram

Schedule a driving lesson



UML Class Diagram



Technical Requirements

The DriverPass system will be a secure, cloud-based web application that students, instructors, and staff can access from any device. It will help users schedule lessons, take practice tests, make payments, and get updates all in one place.

System Setup

DriverPass will run on a web-based client and server setup. The system will be hosted on a cloud platform such as Amazon Web Services or Microsoft Azure so it can handle many users at once and stay online with minimal downtime.

Users will only need an internet connection and a browser on their computer, tablet, or phone.

The front end will be built with standard web tools like TML, CSS, and JavaScript for easy use on different devices. The back end will use a framework like Java Spring Boot or Python Flask to handle login, scheduling, payments, and DMV connections.

Hardware

Because it's cloud-based, DriverPass doesn't need special equipment. The hosting service will handle storage, servers, and system backups.

Each user just needs an internet ready device such as a computer, tablet, or smartphone. Servers will have enough memory and processing power to support hundreds of users at the same time and automatically scale if more users join.

Software

- *Operating Systems: Works on Windows, MacOS, iOS, and Android.*
- *Web browsers: Chrome, Edge, Direfox, and Safari.*
- *Database: MySQL or PstgreSQL to store user data, lessons, invoices, and test results.*
- *Server Framework: Java Spring Boot or Python Flask for handling all system functions.*

External Services:

  - *Payment Gateway for secure online payments.*
  - *Email/Text Notification Service for reminders and confirmations.*
  - *DMV Integration to keep lesson and test information updated automatically.*

Security

Security is a top priority since the system stores payment and personal information. All data will be protected with SSL encryption and secure login features.

- *User roles will control who can access certain features.*
- *Passwords will be encrypted and stored securely.*
- *The system will log important actions to track changes and prevent misuse.*
- *Regular backups and multi-factor authentication will protect admin accounts.*

Performance and Growth

The system will support many users at once without slowing down. Cloud hosting will let it add resources automatically during bust times and scale as the business grows. Caching and load balancing will keep performance fast and reliable.

Maintenance

The DriverPass system will be easy to maintain because of it modular design. New features like additional training tools or reports can be added without interrupting current users. The DMV data and practice tests will update automatically through the connected API.

Summary

These technical requirements make sure the DriverPass system is secure, reliable, and scalable. By using cloud hosting, modern web tools, and strong data protection, the system will run smoothly for all users while staying easy to update and expand in the future.