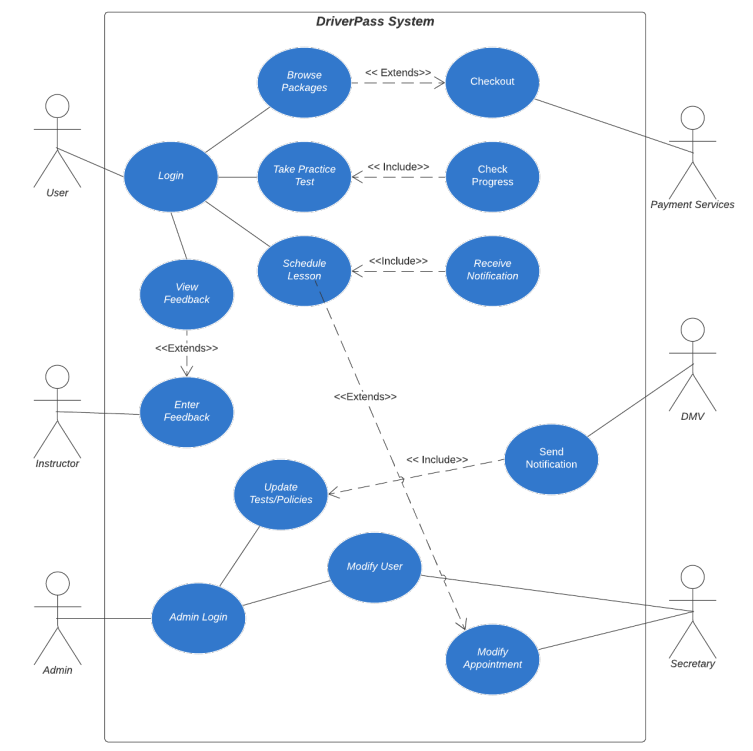
# DriverPass System Design

## UML Diagrams

### UML Use Case Diagram



### UML Activity Diagrams

A diagram of a package

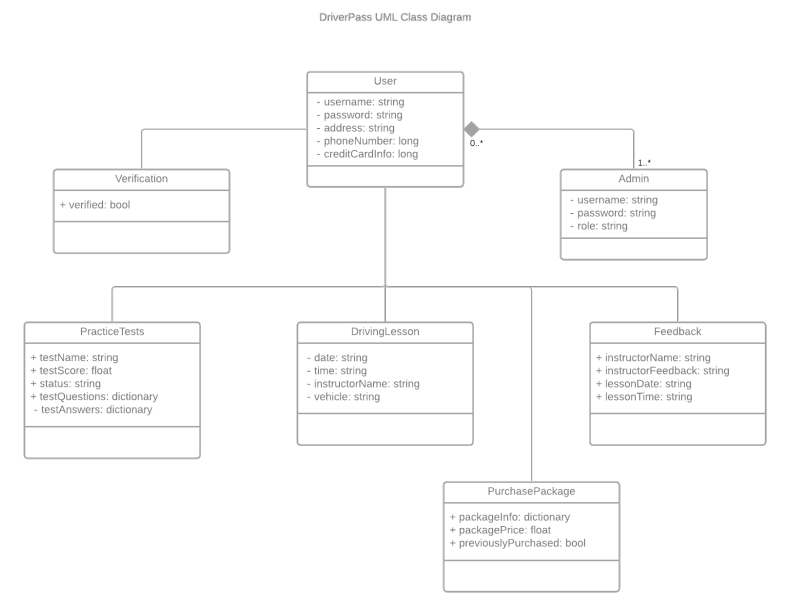
Description automatically generated

### UML Sequence Diagram

A diagram of a user login sequence diagram

Description automatically generated

### UML Class Diagram



## Technical Requirements. Hardware Requirements

## The system requires dedicated server space for hosting the DriverPass web application and its databases. Cloud services such as AWS, Google Cloud, or Microsoft Azure can be used for scalability and reliability. This infrastructure should support high availability and quick scaling. The system must be accessible on various devices, including desktop computers, tablets, and smartphones. Both Android and iOS devices should be supported, requiring adaptive design and cross-platform compatibility. The system should run on commonly used operating systems, including Windows, macOS, iOS, and Android. A robust relational database (e.g., MySQL or PostgreSQL) will store user information, scheduling data, and test results. This database must be secured to handle sensitive user data and optimized for efficient querying.

## The DriverPass system will be built using modern web development frameworks such as React (for the frontend) and Node.js (for the backend). This ensures fast performance and smooth user experience. A firewall, antivirus programs, and encryption tools will be used to protect sensitive data, including user credentials and driving test results. SSL encryption will be mandatory for secure data transfer. Integrated development environments (IDEs) like Visual Studio Code, version control systems like Git, and project management tools such as Jira or Trello will be used by developers to manage the system's development. Automated testing tools, such as Selenium and JUnit, will ensure the system functions as expected across various devices and browsers. Lucidchart or a similar tool will be used for creating and maintaining UML diagrams during system design.

## A reliable cloud hosting provider (AWS, Google Cloud, or Microsoft Azure) will ensure the system's uptime and allow for data backups and recovery. High-speed internet access is required for seamless data transfer between users and the server. Bandwidth must support concurrent user access, especially during peak times. Two-factor authentication (2FA) will ensure that only authorized users can access the system. Passwords must be encrypted, and there should be periodic security audits. Sensitive data, such as user information and test results, will be encrypted both in transit and at rest. Role-based access control (RBAC) will be implemented to restrict access to different parts of the system based on user roles.