# CS 255 System Design Document Kastigar

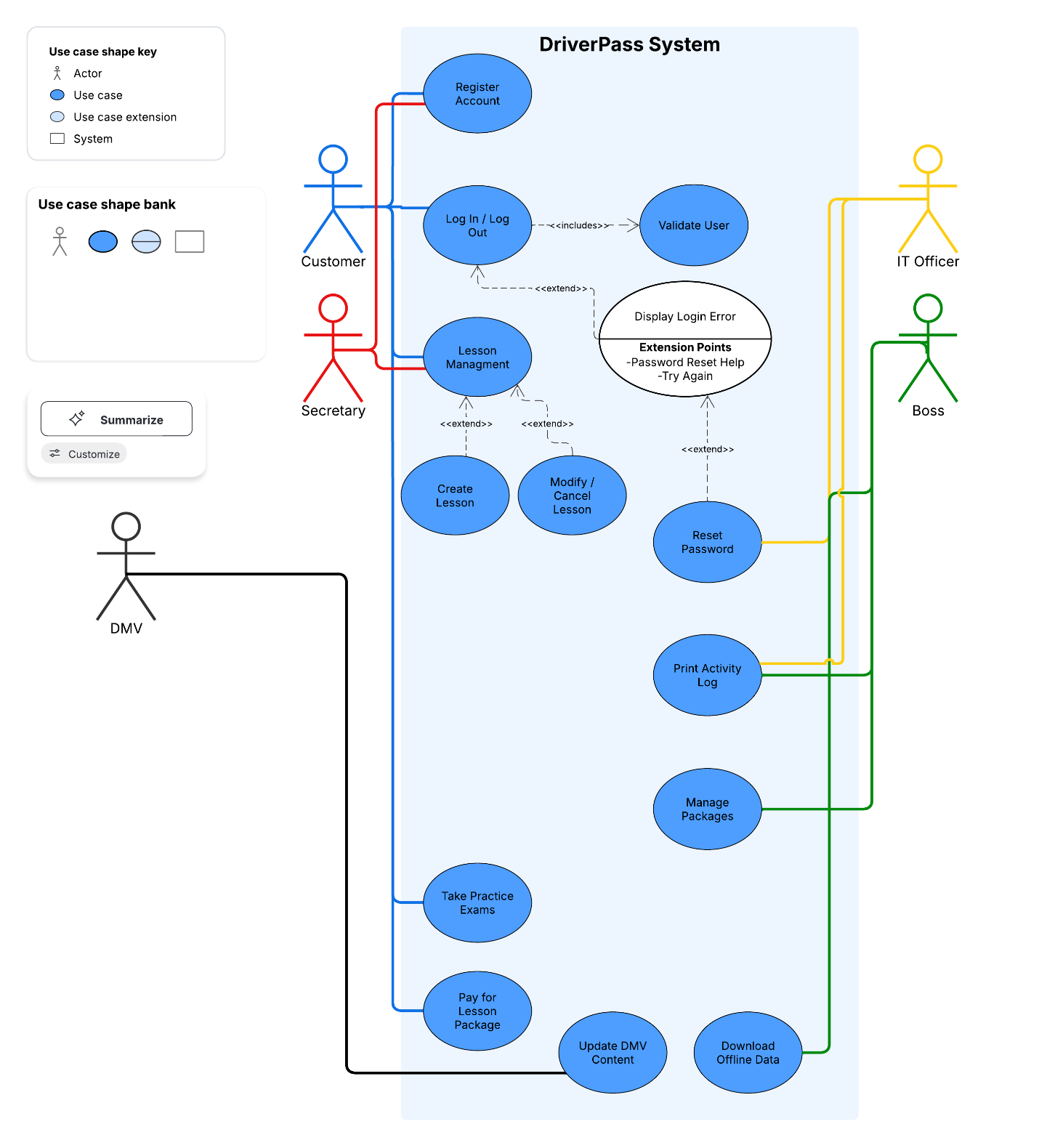
Alex Kastigar

4/20/2025

CS-255 System Analysis and Design

## UML Diagrams

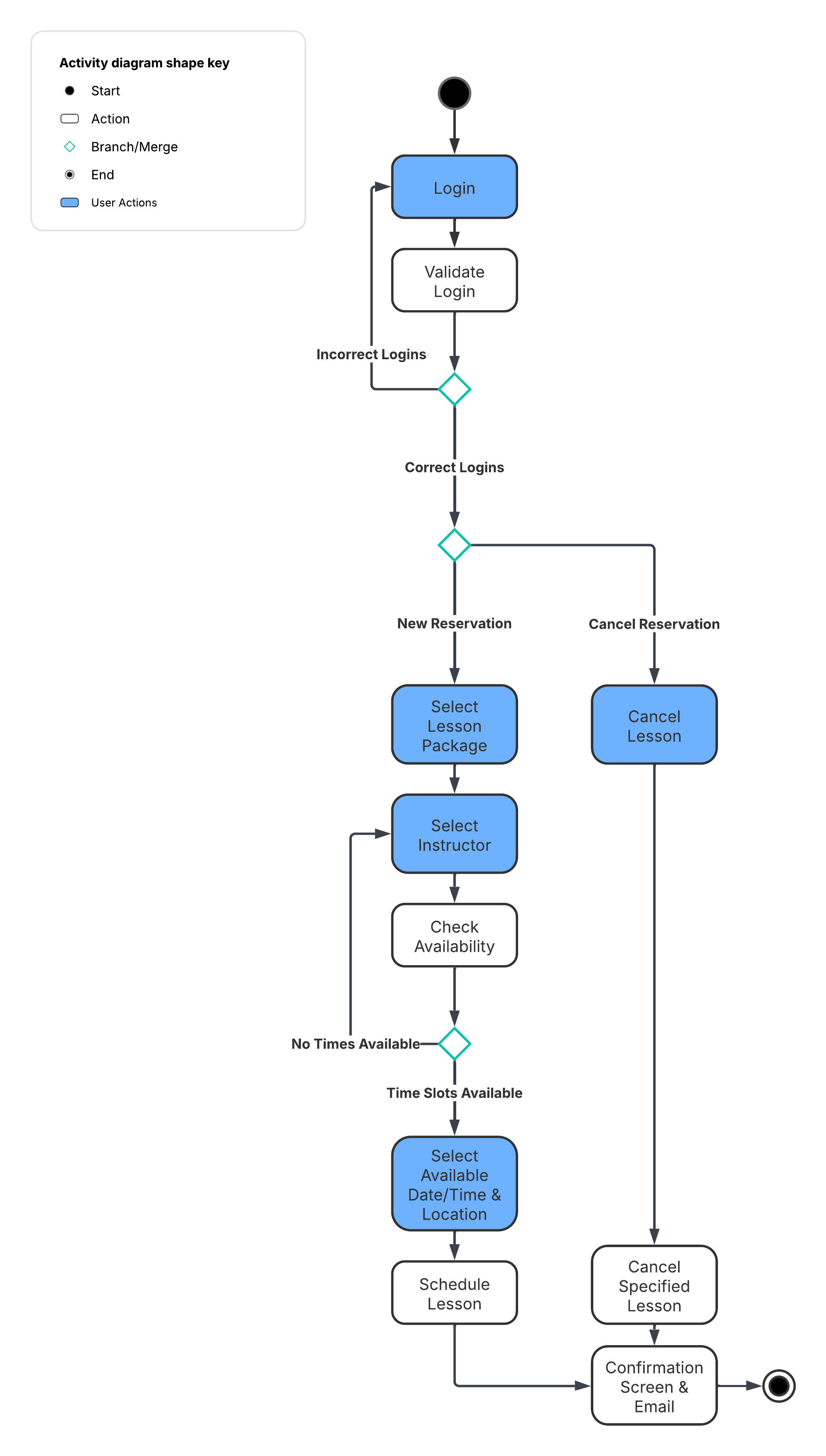
### UML Use Case Diagram DriverPass System

**

This diagram shows how the main users of the DriverPass system interact with its key features. Students can register, schedule lessons, take practice tests, and view feedback. Secretaries manage appointments on behalf of students. The IT Admin can reset passwords and manage account access, while the Boss can view reports and enable or disable lesson packages. This diagram helps outline the system’s major functions and who performs them.

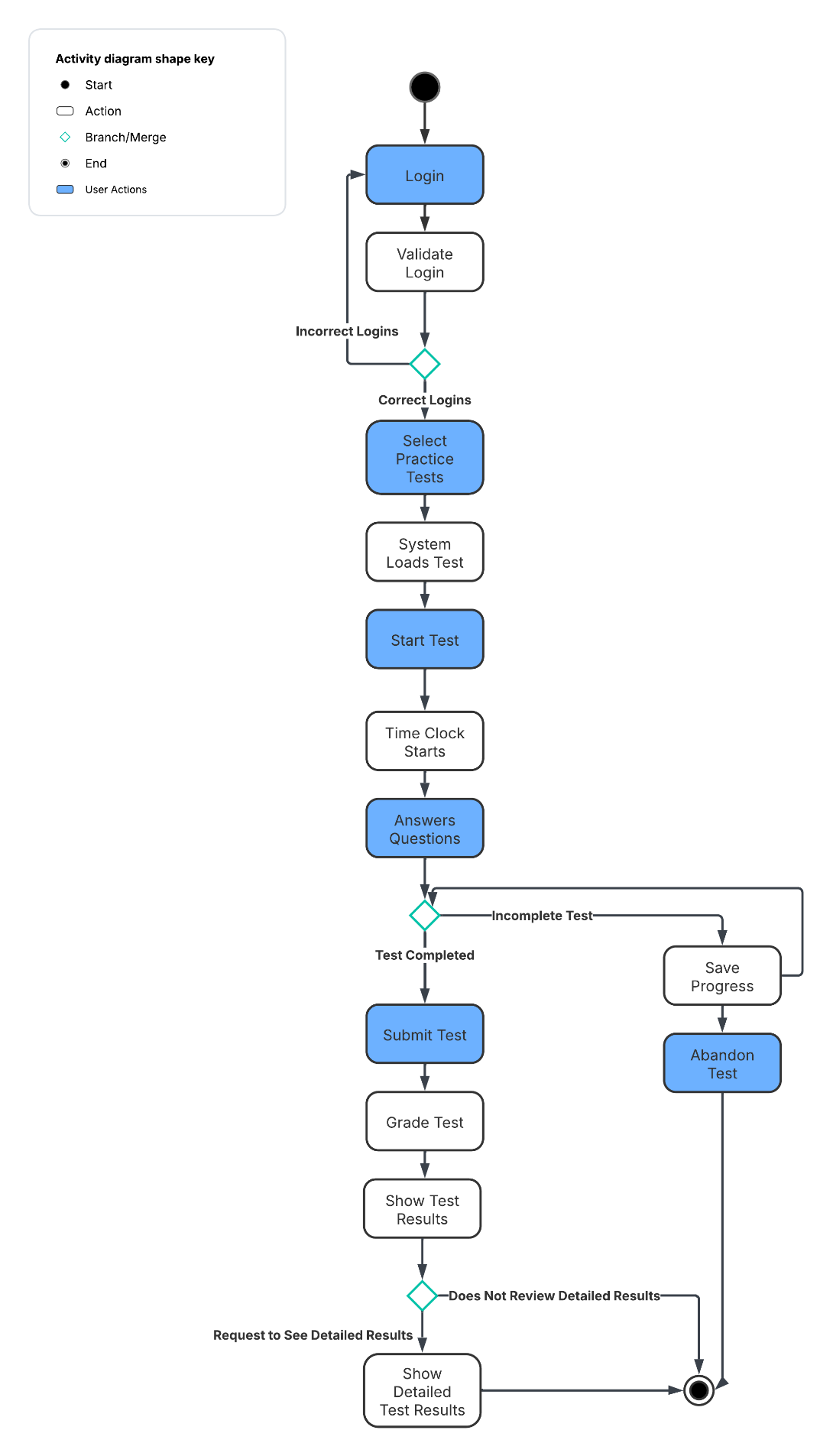
### UML Activity Diagrams

Activity Diagram 1: Schedule Lesson



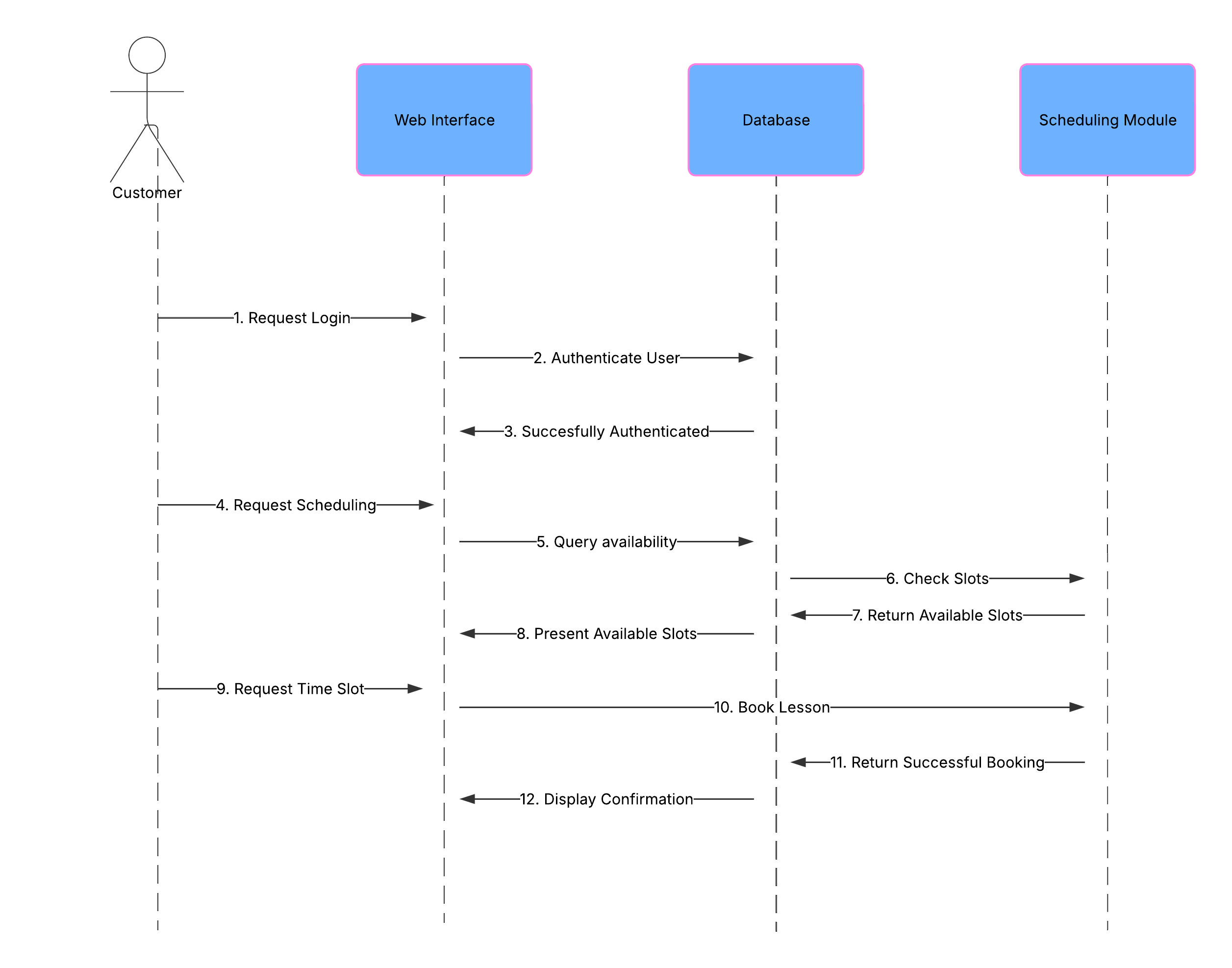
This diagram shows the steps a customer follows to schedule a driving lesson. It begins with the customer logging in and selecting a lesson package. The system then checks for available time slots and instructors. If a match is found, the appointment is confirmed and stored. If no availability is found, the customer is prompted to pick a different instructor. The diagram also allows for optional exits like canceling. User actions and system processes are clearly separated to highlight responsibility at each step.

Activity Diagram 2: Take Practice Test



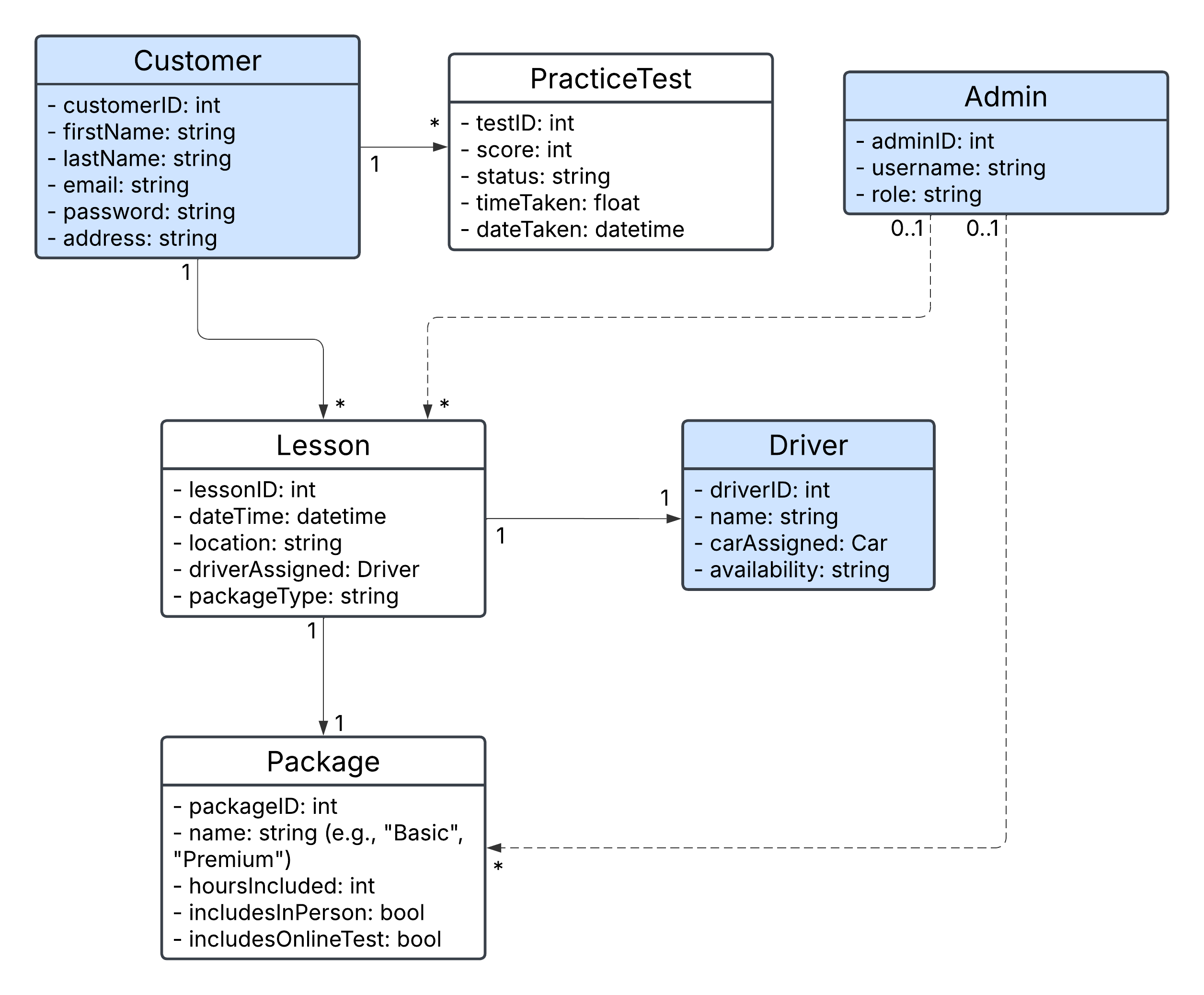
This activity diagram outlines how a customer takes an online practice exam. After logging in and choosing a test, the system loads the content and starts tracking time. The customer answers questions and submits the test, triggering automatic scoring. If the test is not completed, progress can be saved or flagged as incomplete. Additional paths include reviewing a failed test or abandoning the attempt. The diagram shows a smooth user experience while also handling alternative scenarios.

### UML Sequence Diagram Schedule Lesson

**

This sequence diagram details how the system components interact when a customer books a lesson. It starts with the user logging in, then moves through the web interface to the database and scheduling module. There are checks for availability and, if a slot is found, stores the appointment. A confirmation message is then returned to the customer. The diagram emphasizes the step-by-step communication between different parts of the system and the user, ensuring the flow is clear and understandable.

### UML Class Diagram: DriverPass System

**

This UML class diagram outlines the core components of the DriverPass system. The primary classes include Customer, Lesson, Package, Driver, PracticeTest, and Admin. Customers can schedule multiple lessons and take multiple practice tests. Each lesson is tied to a specific package and is assigned one driver. Admins have the ability to manage lessons and packages. Attributes for each class reflect the key data stored in the system, such as lesson time, package type, and customer contact information. The relationships between classes help define how different parts of the system interact and support the scheduling, training, and testing experience for users.

## Technical Requirements

To support the functionality and reliability of the DriverPass system, a set of technical requirements must be in place. These requirements cover the necessary hardware, software, hosting environment, and infrastructure to ensure the system operates smoothly for both customers and internal staff.

The system will be deployed on a cloud-hosted platform, minimizing the need for on-site infrastructure. However, employees such as the secretary or IT administrator will require standard desktop or laptop computers with internet access. These devices should have at least 8 GB of RAM and run a modern operating system such as Windows 10 or macOS Monterey.

The front-end interface will be web-based, accessible via major browsers including Chrome, Firefox, Safari, and Edge. The backend system will be built using scalable server-side technologies, such as Node.js or Python frameworks, with a secure and modular architecture. A SQL-based relational database (e.g., PostgreSQL or MySQL) will be used to store user data, lesson details, driver assignments, and test results.

To ensure reliability, the system will be hosted on a cloud platform such as Amazon Web Services (AWS) or Microsoft Azure. This will allow for high availability, automatic backups, and scalable performance as the number of users increases. Load balancing and distributed servers will be configured to manage concurrent users, especially during peak times when multiple students may be taking practice tests or scheduling lessons.

All data exchanged between users and the system will be encrypted using HTTPS/TLS protocols. Role-based access controls will be implemented to restrict access based on user type (e.g., customer, admin, IT). Passwords will be securely hashed and stored, and self-service password reset functionality will be included for user convenience. The system will also lock accounts after repeated failed login attempts to prevent brute-force attacks.

The IT administrator will have the tools to manage users, update lesson packages, and monitor system activity through a secure admin dashboard. System updates will be deployed during off-peak hours and will follow a version-controlled deployment strategy. Automated notifications will alert the admin when DMV test requirements are updated, allowing for timely content revisions.

The system will be fully responsive, allowing users to access their accounts from both desktop and mobile devices. This ensures customers can schedule or review lessons and take practice tests from anywhere, supporting a more flexible user experience.