Southern New Hampshire University

Clayton Cooper

CS-255

2/18/2026

CS 255 System Design Document Template

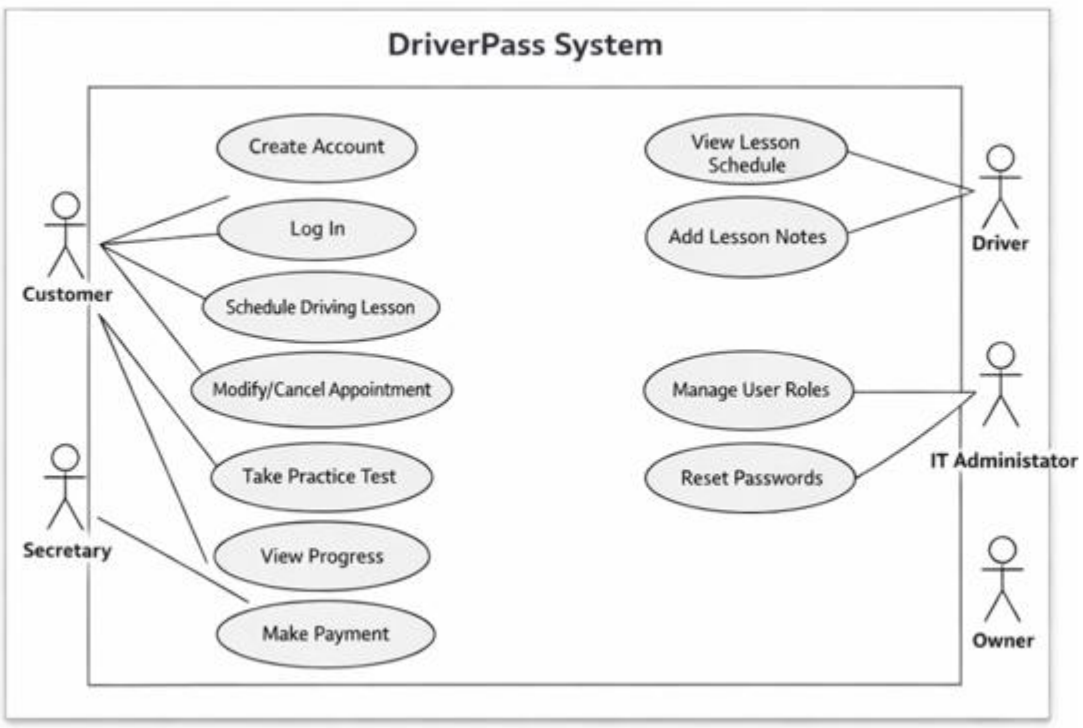
UML Diagrams

UML Use Case Diagram

The use case diagram shows how various users engage with the DriverPass system to accomplish learning and business-related tasks. By offering online scheduling, driving instruction administration, and progress tracking capabilities, the system assists clients, administrative personnel, teachers, and system administrators.

The system is used by users to register, log in, book driving lessons, change or cancel appointments, take practice exams, pay, and monitor their progress. Secretaries assign teachers and cars and oversee customer appointments. The system is used by drivers to check lesson plans and take notes on student performance. By controlling user roles, changing passwords, and guaranteeing system operation, IT administrators keep the system in good working order. The system is used by the business owner to keep an eye on activities and provide reports.

The use case illustrates how the solution helps DriverPass achieve its objectives of raising student achievement rates while enhancing administrative effectiveness and appointment scheduling.

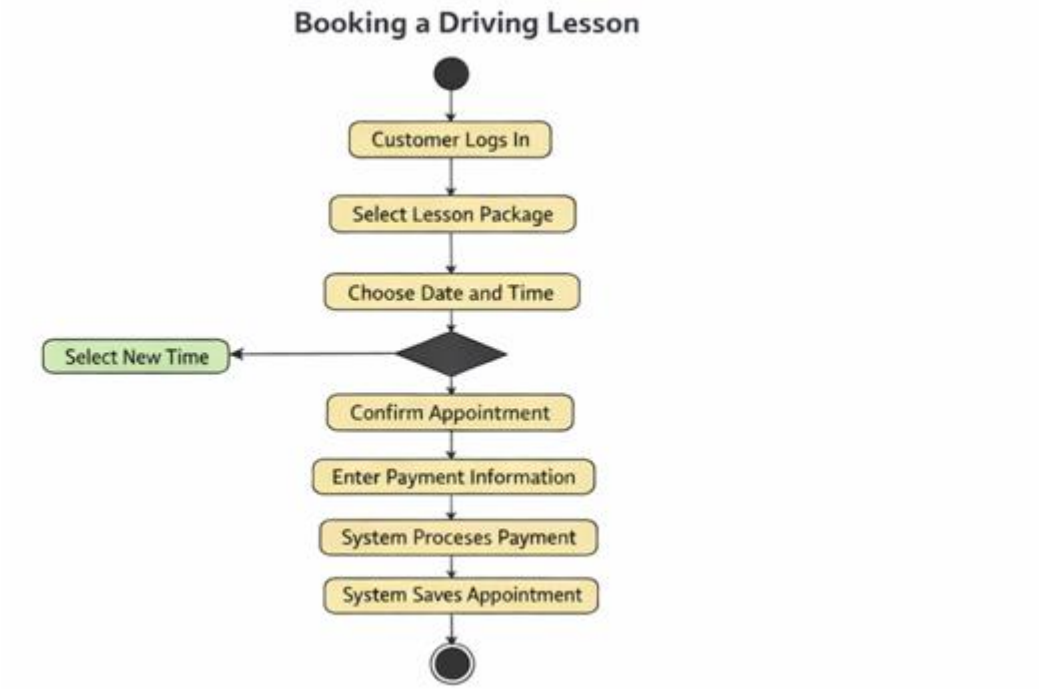


UML Activity Diagrams

Activity Diagram 1: Booking a Driving Lesson

The steps clients take to arrange driving lessons are shown in this activity diagram. After choosing a lesson package and logging into the system, the process starts. After that, the client chooses a time and date that works for the lesson. Before confirming the appointment, the system verifies the availability of the teacher and the car. The consumer must choose a different time if the one they have chosen is not available. The system saves the class schedule and handles payment after the appointment is confirmed. The customer is subsequently shown a confirmation message.

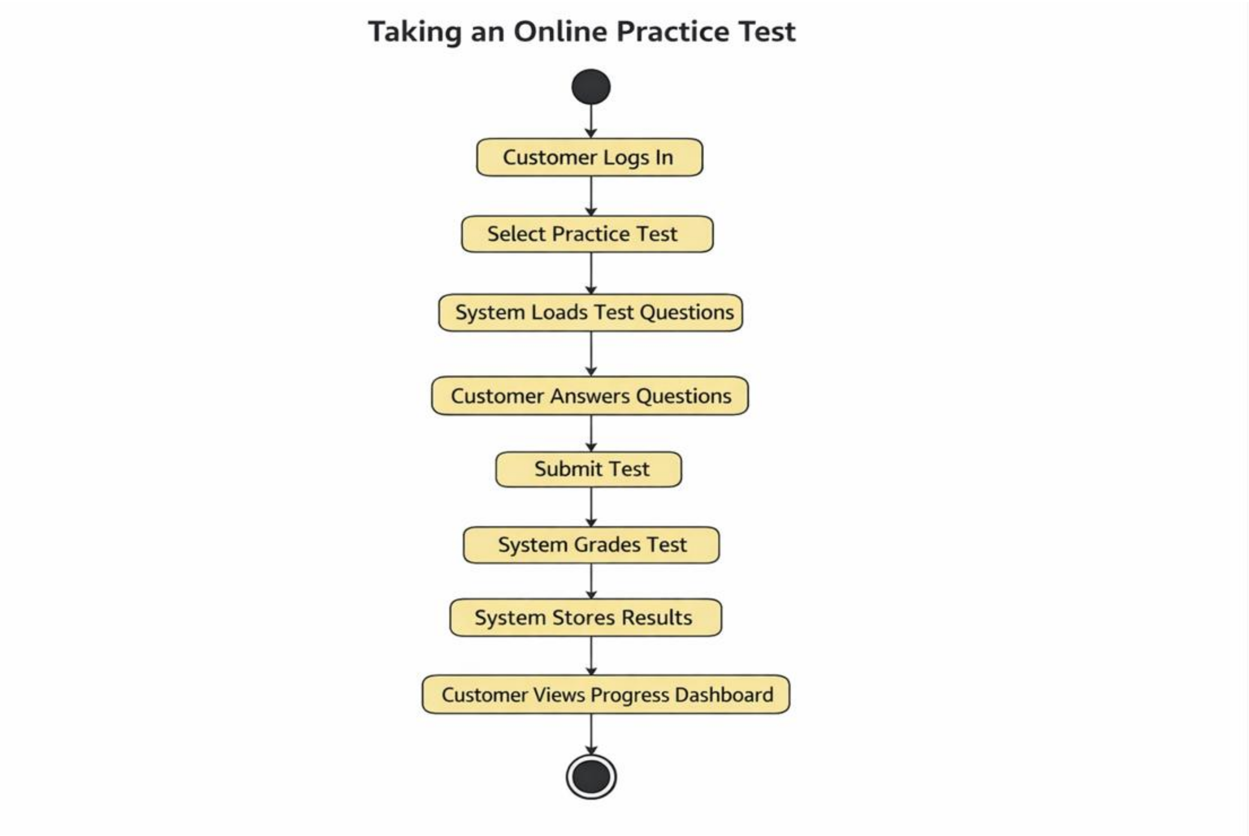
By eliminating scheduling conflicts and enabling online lesson booking, this procedure increases efficiency.



Activity Diagram 2: Taking an Online Practice Test

This activity map shows how users navigate the system to finish practice exams. After choosing a practice test and logging into the system, the customer starts the process. The customer can finish the exam after the system loads the test questions. The system automatically grades the test after it is turned in, and the results are saved in the customer's progress record. After then, the client can examine test results and monitor progress over time.

DriverPass's objective of increasing test success rates and assisting students in preparing for license examinations is supported by this procedure.

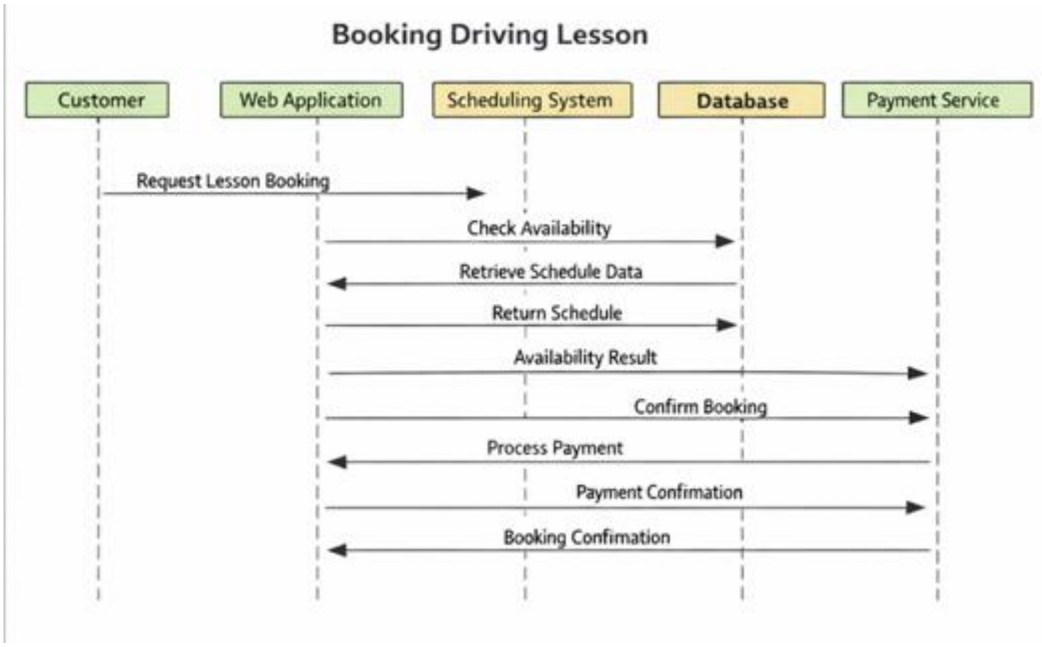

Taking an Online Practice Test
Customer Logs In
Select Practice Test
System Loads Test Questions
Customer Answers Questions
Submit Test
System Grades Test
System Stores Results
Customer Views Progress Dashboard

UML Sequence Diagram

The system's interactions during the scheduling of driving lessons are depicted in the sequence diagram. Communication between the client, web application, database, scheduling system, and payment processing service is depicted in the diagram.

When a customer uses the web application to request a lesson, the procedure starts. To verify the availability of lessons, the program interacts with the scheduling system. The web application receives availability results from the scheduling system once it has retrieved schedule data from the database. The buyer provides payment details when availability is verified. Payment information is transmitted from the web application to the payment service for processing. The system sends the consumer a confirmation message and enters the appointment details into the database upon receiving payment confirmation.

This figure shows how the many parts of the system cooperate to finish lesson scheduling in a safe and effective manner.



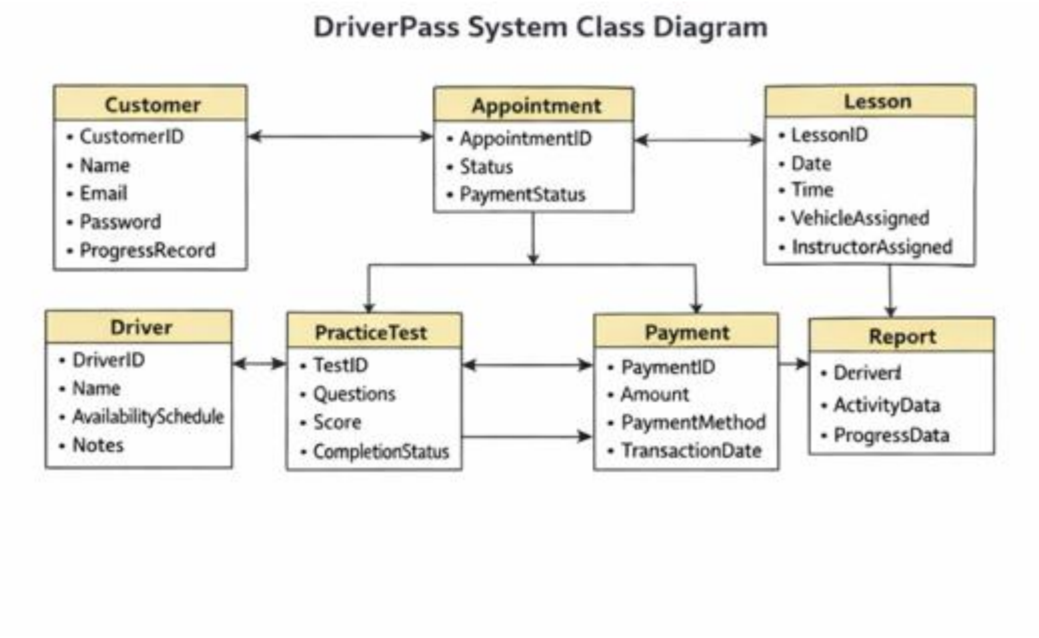
UML Class Diagram

The DriverPass system's structure and the connections between its many parts are depicted in the class diagram. The basic classes shown in the diagram are as follows:

Identification information, contact data, login passwords, and learning progress records are all stored in the Customer class. Lesson availability schedules, instructor notes, and identity information are all stored in the Driver class. Details about the lesson, such as the date, time, instructor, and vehicle allocated, are stored in the Lesson class. Information about scheduled lessons, such as appointment and payment status, is stored in the Appointment class.

Information about online tests, such as test identity, test questions, completion status, and scores, is stored in the Practice Test class. Payment details, such as payment identifier, payment method, transaction amount, and payment date, are stored in the Payment class. Data used to create performance and business activity reports is stored in the Report class.

The class diagram also shows the connections between the various parts of the system. Clients are able to make several payments, take multiple practice exams, and arrange multiple appointments. Drivers are allocated lessons, and appointments are linked to lessons. Reports facilitate business monitoring by compiling data from various system components.



Technical Requirements

For the DriverPass system to support system functioning, security, performance, and dependability, certain technical components are needed.

Hardware Requirements

For the web application and database storage to be hosted, the system needs cloud-based servers. Devices with internet connection, such as laptops, desktop computers, tablets, and smartphones, must be available to both employees and customers. Workstation computers are necessary for instructors and administrative staff to observe course information, manage timetables, and track student progress. Systems for backup storage are necessary to safeguard and maintain system data.

Software Requirements

Modern web browsers can access the DriverPass system as a web-based application. User accounts, lesson plans, payment details, and student progress records must all be stored in a database management system. Integrating secure payment processing software is necessary to safeguard client financial data. To protect login authentication procedures and encrypt data communications, security software is necessary.

Development Tools

Lucidchart was used to accomplish system modeling and diagram construction. Web development frameworks to support application functionality and testing tools to confirm system security and performance are examples of additional development tools.

Infrastructure Requirements

For clients and staff to utilize the system's functionality, dependable internet access is necessary. Scalability, dependability, and safe data storage are all dependent on cloud hosting infrastructure. To securely execute financial transactions, the system needs to be integrated with a secure payment gateway. Automated backup and recovery systems are required to prevent data loss and support system continuity.