# CS 255 System Design Document - Carolyn Klein

## UML Diagrams

### UML Use Case Diagram

*DriverPass Use Case Diagram:*

*A diagram of a person's work flow

AI-generated content may be incorrect.*

### UML Activity Diagrams

*DriverPass Login Activity Diagram:*

*A diagram of a computer program

AI-generated content may be incorrect.*

*DriverPass Purchase Packages Activity Diagram:*

*A diagram of a card

AI-generated content may be incorrect.*

### UML Sequence Diagram

*[You were asked to create a sequence diagram based on* ***one*** *of the use cases you chose. Please insert your sequence diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s needs.]*

### UML Class Diagram

A diagram of a work flow

AI-generated content may be incorrect.

This diagram visually represents the classes and their relationships. It shows:

* **Driver:** Has attributes like driverID, name, etc. A driver can schedule multiple appointments.
* **Test:** Has attributes like testID, date, etc. A test is included in multiple appointments (e.g., if re-taken).
* **Instructor:** Has attributes like instructorID, name, etc. An instructor conducts multiple appointments.
* **Appointment:** Has attributes like appointmentID, date, etc. An appointment is scheduled by one driver, conducted by one instructor, and includes one test.
* **Payment:** Has attributes like paymentID, amount, etc. Each appointment requires one payment.

The lines connecting the classes represent the relationships:

* Driver "1" -- "\*" Appointment: One driver can schedule many appointments.
* Instructor "1" -- "\*" Appointment: One instructor can conduct many appointments.
* Test "1" -- "\*" Appointment: One test can be part of many appointments (e.g., if a driver fails and retakes it).
* Appointment "1" -- "1" Payment: One appointment requires one payment.

## Technical Requirements

**3. Technical Requirements**

Based on the functional and non-functional requirements outlined in the Business Requirements Document, the following technical requirements are proposed for the DriverPass system:

**3.1 Hardware:**

* **Servers:** A server (or potentially multiple servers depending on scalability needs) to host the application and database. Specifications will depend on the expected load and traffic.
* **Client Devices:** Users will access the system through various devices, including desktops, laptops, tablets, and smartphones. The system should be responsive and compatible with different screen sizes and operating systems.

**3.2 Software:**

* **Operating System:** A stable and secure server operating system (e.g., Linux, Windows Server).
* **Database Management System (DBMS):** A robust database (e.g., MySQL, PostgreSQL, SQL Server) to store user data, test information, and other relevant data.
* **Web Application Server:** A platform (e.g., Apache Tomcat, JBoss) to deploy and manage the DriverPass application.
* **Programming Languages:** Languages suitable for web development (e.g., Java, Python, JavaScript) for the application's backend and frontend.
* **Web Framework:** A framework (e.g., Spring, Django, React) to streamline development and provide structure.

**3.3 Tools:**

* **CASE Tool:** Lucidchart (used for UML diagram creation).
* **Integrated Development Environment (IDE):** An IDE (e.g., Eclipse, IntelliJ IDEA, PyCharm) for developers to write and debug code.
* **Version Control System:** A system like Git for managing code changes and collaboration.
* **Testing Tools:** Tools for unit testing, integration testing, and user acceptance testing.

**3.4 Infrastructure:**

* **Network:** A reliable network infrastructure to ensure accessibility and performance. This might include load balancers and firewalls.
* **Security:** Measures to protect user data and prevent unauthorized access, including encryption, authentication, and authorization mechanisms.
* **Backup and Recovery:** A strategy for backing up data and restoring the system in case of failures.
* **Hosting:** A suitable hosting environment (e.g., cloud-based, on-premise) to host the application and database.

**4. Conclusion**

This system design document provides a blueprint for the DriverPass system. The UML diagrams illustrate the system's functionality and interactions, while the technical requirements outline the necessary hardware, software, tools, and infrastructure. This design aims to meet the client's needs for an efficient and user-friendly system for managing driving tests. The next step will be to work with the development team to implement this design.