Angel Silva

November 27, 2025

CS-255 | 5-2 Project

System Components and Design
Purpose

- DriverPass seeks to develop an online system that allows customers to access practice exams, online course materials, and schedule on-the-road training sessions.

- The purpose of the system is to improve student success rates on DMV driving tests by providing high-quality digital and in-person training resources.

- The client, **DriverPass**, wants a system that:

    - Allows customers to take online classes and tests.

    - Allows customers to schedule, modify, and cancel appointments.

    - Tracks drivers, cars, lessons, and customer progress.

    - Provides administrators with full access to user accounts, reports, and system data.

    - Enables secure access from any device.

System Background

- DriverPass identified a gap in driver-training tools: **65% of students fail their DMV exam** due to inadequate preparation.

- The system must give customers:

    - Online practice tests

    - Course content

    - Package registration

    - Driving lesson scheduling
- The system must support employees:
    - Secretaries making appointments
    - IT admins managing roles, rights, and account resets
    - Owner access to reporting
- Key components:
    - Web-based user interface (cloud hosted)
    - Customer account management
    - Online practice test delivery
    - Scheduling/reservation system
    - Driver/car assignment module
    - Reporting and activity logging system
    - DMV integration for updated rules, questions, and policies

Objectives and Goals

- Provide user-friendly access to online courses and driving tests.
- Enable customers to:
    - Create accounts
    - Purchase packages
    - Schedule lessons
    - Take online tests
    - View scores and driver notes

- Enable admin and staff to:
    - Manage users
    - Reset passwords
    - Track who created/modified each reservation
    - Print activity reports
- Ensure accurate tracking of drivers, cars, lessons, and customer progress.
- Maintain up-to-date DMV compliance through notifications and updates.

Requirements
Nonfunctional Requirements

Performance Requirements

- The system will run as a cloud-based web application accessible from any computer or mobile browser.
- The system must load core pages (login, scheduling, tests) within **2–3 seconds** under normal load.
- Updates to system content (DMV updates, course modules, tests) should be deployable without service interruption.
- System availability goal: **99% uptime**.

Platform Constraints

- The system must run in a **web browser** on Windows, macOS, iOS, Android, and Linux.
- The backend requires:
    - A secure cloud server
    - A relational database (e.g., MySQL, PostgreSQL)

    - Automatic cloud backup services

- No local installation required.

Accuracy and Precision

- The system must distinguish users by unique credentials (username + password or email + password).
- All input (addresses, names, card info) must be validated for accuracy.
- System will track:
    - Who made a reservation
    - Who canceled it
    - Who last modified it
- Admins will be notified of:
    - Failed login attempts
    - Password reset requests
    - DMV update notifications

Adaptability

- Admins should be able to:
    - Add/remove/disable user accounts
    - Disable customer package options
- The system must be flexible for future updates (adding new packages or modules).
- The system must adapt to platform updates (browser updates, OS changes) with minimal user impact.
- IT admin requires full access to:

    - Modify roles
    - Reset accounts
    - Disable employees
    - Oversee system logs

Security

- The system requires secure login with encrypted passwords.
- Enforce secure HTTPS communication for all data transfers.
- After 5 failed login attempts, system should temporarily lock the account.
- Password reset workflow available for both admins and customers.
- Sensitive data (credit card, personal info) must use encryption at rest and in transit.
- Activity logs must record all system actions.

Functional Requirements

(Each bullet begins with "The system shall…")

- The system shall allow users to create accounts and log in securely.
- The system shall validate user credentials during login.
- The system shall allow customers to purchase training packages.
- The system shall allow customers to schedule, modify, and cancel driving lesson reservations.
- The system shall assign each reservation to a driver, car, date, and time.
- The system shall track which user created, modified, or canceled each reservation.
- The system shall provide online practice tests with scoring.

- The system shall store and display customer test history, including status (not taken, in progress, passed, failed).
- The system shall allow drivers to leave notes for each lesson.
- The system shall allow administrators to reset user passwords.
- The system shall allow administrators to enable, disable, or modify user accounts.
- The system shall send notifications when DMV updates rules or test questions.
- The system shall support generating printable activity reports.
- The system shall store customer registration info (name, address, phone, payment info).
- The system shall allow customers or secretaries to enter registration data.
- The system shall provide contact forms for communication between DriverPass staff and customers.

User Interface

- Interfaces needed:
    - **Customer portal**
        - View packages
        - Schedule/cancel lessons
        - Take online tests
        - View scores, progress, and driver notes
    - **Secretary portal**
        - Register customers
        - Enter scheduling information
        - Update reservations by phone or in person

        - **Admin portal**
            - Reset passwords
            - Manage users
            - Track system changes
            - View reports
        - **Owner portal**
            - Access all reports
            - View business activity
- Accessible via:
    - Web browsers on desktop and mobile.
- UI should be modern, simple, cloud-based, and based on Liam's sketches:
    - Progress tables for tests
    - Driver notes table
    - Input forms for registration

Assumptions

- Users will have access to the internet and a modern browser.
- DMV provides API or automated notifications for policy/test updates.
- Credit card processing will be handled through a secure third-party payment service.
- Secretaries and admins will be trained prior to system rollout.
- All drivers have fixed schedules that can be assigned to reservations.

Limitations

- The system requires the internet to modify or update data.
- Customers cannot modify packages; only admins can enable/disable them.
- Resources, budget, and time may limit the number of advanced features in Version 1.
- Future enhancements (adding new packages/modules) may require developer involvement.
- DMV integration depends on external system availability and compatibility.

Gantt Chart

Here is a clean Gantt schedule recreated exactly from the transcript:

Task	Start Date	End Date
Collect Requirements	Jan 22	Feb 4
Create Use Case Diagrams	Feb 11	Feb 18
Build Activity Diagrams	Feb 15	Mar 9
Research UI Designs	Feb 27	Mar 7
Build Class Diagram	Mar 1	Mar 9
Customer Approval	Mar 10	Mar 11
Build Interface	Mar 12	Mar 24
Link DB to Interface	Mar 24	Apr 3
Build Business Logic	Apr 5	Apr 27
Test System	Apr 27	May 7
Deliver System	May 8	May 9
Sign-off Meeting	May 9	May 10