# CS 255 Business Requirements Document

Edwin Martinez

7/28/2025

## System Components and Design

### Purpose

*What is the purpose of this project? Who is the client and what do they want their system to be able to do?*

* The purpose of this project is to design a comprehensive training and appointment scheduling system for DriverPass, a company that helps students prepare for driving tests.
* DriverPass wants the system to support online practice exams, in-person driving lesson reservations, user data management, and test progress tracking.

### System Background

*What does DriverPass want the system to do? What is the problem they want to fix? What are the different components needed for this system?*

* The current problem is that many students fail the DMV driving test due to inadequate preparation.
* DriverPass aims to solve this by offering structured, accessible online and on-the-road training.
* Components needed include:
  + Web-based platform (cloud-hosted)
  + database for storing users and appointments
  + scheduling system
  + user role management
  + test progress tracker
  + notification system
  + DMV update integration.

### Objectives and Goals

*What should this system be able to do when it is completed? What measurable tasks need to be included in the system design to achieve this?*

* Provide users with the ability to create accounts, schedule/cancel/modify appointments, and access online practice exams.
* Track and display lesson history, driver comments, and test progress (status: not taken, in progress, failed, passed).
* Allow secretaries to register users, schedule lessons, and update user information.
* Allow IT/admin users to manage access, reset passwords, and generate activity reports.
* Allow future package customization and integration with DMV updates.

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

*What environments (web-based, application, etc.) does this system need to run in? How fast should the system run? How often should the system be updated?*

* The system must be web-based and cloud-hosted.
* Must load user dashboards and reports within 3 seconds.
* Reports should be exportable (Excel).
* Users should be able to access the platform from any device with internet access.

#### Platform Constraints

*What platforms (Windows, Unix, etc.) should the system run on? Does the back end require any tools, such as a database, to support this application?*

* Platform must be accessible via modern browsers (Chrome, Firefox, Edge).
* Cloud-hosted platform (AWS or Azure); no on-premise hosting.
* Back-end should use a secure database system (MySQL, PostgreSQL).

#### Accuracy and Precision

*How will you distinguish between different users?* *Is the input case-sensitive? When should the system inform the admin of a problem?*

* User roles must be clearly defined (admin, IT, secretary, student).
* Tracking must log which user made any changes to appointments.
* System must distinguish which student is assigned to which driver, time, and car.

#### Adaptability

*Can you make changes to the user (add/remove/modify) without changing code? How will the system adapt to platform updates? What type of access does the IT admin need?*

* Admin users should be able to disable packages without code changes.
* IT admin must be able to block or reset accounts.
* System should handle platform/browser updates without breaking functionality.

#### Security

*What is required for the user to log in? How can you secure the connection or the data exchange between the client and the server? What should happen to the account if there is a “brute force” hacking attempt? What happens if the user forgets their password?*

* Secure login required for all users; password reset via email.
* All data transfers must use HTTPS encryption.
* Role-based access control must be enforced.
* System should detect and lock out brute-force login attempts after 5 failures.

### Functional Requirements

*Using the information from the scenario, think about the different functions the system needs to provide. Each of your bullets should start with “The system shall . . .” For example, one functional requirement might be, “The system shall validate user credentials when logging in.”*

* The system shall allow customers to create and manage their account.
* The system shall allow students to schedule, cancel, and modify lessons.
* The system shall allow secretaries to register students and create appointments.
* The system shall track lesson history, start and end time, and driver notes.
* The system shall track practice test progress and scores.
* The system shall allow admin users to generate activity and audit reports.
* The system shall receive DMV updates and display policy/test changes.
* The system shall allow IT to manage user accounts, roles, and reset passwords.

### User Interface

*What are the needs of the interface? Who are the different users for this interface? What will each user need to be able to do through the interface? How will the user interact with the interface (mobile, browser, etc.)?*

* Web interface optimized for both desktop and mobile.
* Dashboard view for students showing test progress, appointments, and notes.
* Admin/IT interface to manage users and generate reports.
* Form-based interface for secretaries to input new student data.
* Login screen and password recovery workflow.

### Assumptions

*What things were not specifically addressed in your design above? What assumptions are you making in your design about the users or the technology they have?*

* Users have internet access and email.
* DMV will provide updates in a compatible format (JSON or XML).
* Students will use the platform as their primary method of interaction.

### Limitations

*Any system you build will naturally have limitations. What limitations do you see in your system design? What limitations do you have as far as resources, time, budget, or technology?*

* System cannot be updated while offline.
* System depends on third-party DMV data accuracy and frequency of updates.
* Non-technical users cannot add/remove new packages (only disable).

### Gantt Chart

