# CS 255 Business Requirements Document

## 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?*

* DriverPass is addressing a need in the market for more comprehensive driving test preparation tools.
* The system will offer both online practice exams and on-the-road training.
* The system will help DriverPass customers pass their DMV driving tests by providing lessons, test preparation, and scheduling tools.
* Users will be able to access lessons, practice exams, and book driving lessons through the system.
* The system will allow for role-based access for different employees (e.g., IT admin, secretaries, etc.) and provide comprehensive tracking of user actions.

### 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?*

* Problem to be addressed: A large percentage of students fail their DMV driving exams due to inadequate preparation. DriverPass wants to address this by offering both online and in-person training resources.

**Key system components**:

* Online class modules with practice exams based on DMV tests.
* A reservation system for booking driving lessons.
* Reporting and tracking tools for user actions, reservations, and system modifications.
* Role-based access control to ensure that different user roles (e.g., IT admin, secretary, customers) have appropriate levels of access.
* Integration with DMV databases for real-time updates on driving test rules and policies.

### 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?*

* Lesson Scheduling: Customers should be able to book and cancel lessons both online and via phone.
* Flexible Packages: Three pre-configured packages of driving lessons should be available, with future options for customization or disabling of certain packages.
* Progress Tracking: Students should be able to view their progress, including completed lessons, test scores, and instructor feedback.
* Data Reporting: The system should allow admins to track and generate reports on user actions such as bookings, cancellations, and modifications.
* Secure Login: Users should be able to reset passwords securely, and the system must protect against security threats like brute-force attacks.

## Requirements

### Nonfunctional Requirements

*In this section, you will detail the different nonfunctional requirements for the DriverPass system. You will need to think about the different things that the system needs to function properly.*

#### 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 should be web-based and responsive on both mobile and desktop platforms.
* System updates should be applied seamlessly without downtime.
* The system should be available 24/7 with minimal downtime for updates or maintenance.
* It should load within two seconds on average and support up to 500 simultaneous users.

#### 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?*

* The system should run on common platforms like Windows, iOS, and Android.
* It will require a database to support user accounts, scheduling, and package tracking.
* The system must be compatible with major browsers (e.g., Chrome, Firefox, Safari) and mobile devices (iOS, Android).
* A cloud-based infrastructure is preferred to avoid local server maintenance.

#### 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?*

* All user actions should be accurately logged and the system should validate inputs like usernames and passwords (case-sensitive).
* The system must notify administrators of any irregularities or errors in user data.
* User inputs (e.g., reservation details, passwords) must be case-sensitive, and all changes to records must be logged with timestamps.
* Admins should be notified of any potential issues, such as booking conflicts or failed login attempts.

#### 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?*

* Administrators should be able to add, remove, or modify user accounts without coding.
* The system must be adaptable to future updates to platforms and user needs.
* The system must adapt to changes in external APIs with minimal disruption.

#### 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?*

* User login must require secure credentials, and the system must protect against brute-force attacks.
* Password reset options must be secure, and accounts should be locked after multiple failed login attempts.
* All user data must be encrypted during transmission and storage, with two-factor authentication required for admin accounts.
* Users should have the ability to reset their passwords via email verification.

### 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 users to register for driving lessons online or through a secretary.
* The system shall track selected driving packages and manage lesson schedules.
* The system shall allow users to reset their passwords securely.
* The system shall track and report changes to reservations, cancellations, and modifications.

### 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.)?*

* The system will provide a web-based user interface that allows customers to register for lessons, view their progress, and manage their accounts.
* Administrators will have an interface to manage user accounts, track reservations, and generate activity reports.

### 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 will have access to internet-capable devices for scheduling lessons and accessing training materials online.
* The system will be integrated with DMV databases for updates, though specific methods of integration will be handled by the IT team.

### 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?*

* Package modifications require developer input and cannot be done by end-users.
* System performance depends on third-party integrations, such as the DMV databases.

### Gantt Chart

*Please include a screenshot of the GANTT chart that you created with Lucidchart. Be sure to check that it meets the plan described by the characters in the interview.*

A graph of a driver pass project

Description automatically generated