# 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 a system designed to help students prepare for their driving tests by providing online practice exams and on-the-road training. The system needs to enable users (customers, secretaries, and IT staff) to perform different tasks, such as scheduling driving lessons, tracking user activity, and accessing data online.

### 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?*

* DriverPass aims to solve the issue of students failing driving tests due to inadequate preparation. The system will provide online tools and resources to prepare users, including online exams and the ability to book on-road training. Key components of the system include user management, online scheduling, data reporting, and security features.

### 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?*

* The system will allow customers to book and modify driving lessons online.
* It will provide access to training materials, including online practice tests.
* The system will track user actions, including reservations, cancellations, and modifications.
* IT staff will be able to manage accounts and permissions.
* The system will generate reports showing user activity and progress.

## 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 needs to run on the web, supporting mobile and desktop access. It should be fast, with load times under 3 seconds, and updated weekly to ensure continuous improvement.

#### 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 be accessible via common operating systems like Windows, macOS, and mobile platforms (iOS, Android). The backend will use a cloud-based database to support scalability.

#### 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?*

* Users will have unique identifiers, and the system will differentiate users by role (e.g., customer, IT staff). All inputs should be validated to avoid errors, and the system will notify the admin of issues, such as 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?*

* The system will allow modifications to user access without requiring code changes. It should adapt to changes in platform requirements and offer full access to IT administrators for managing roles and permissions.

#### 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 authentication will be required, with passwords stored securely and encrypted. If multiple failed login attempts occur, the system will lock the account and notify the user. Password recovery will be available for users who forget their login credentials.

### 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 validate user credentials when logging in.
* The system shall allow users to schedule, modify, and cancel appointments.
* The system shall track user activity, including appointment scheduling and modifications.
* The system shall generate reports showing activity logs and driving lesson details.

### 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’s interface will cater to different user roles:

* **Customers** will schedule appointments and take online exams via a web-based interface.
* **Secretaries** will manage appointments through the system.
* **IT Staff** will manage system permissions and security through an admin interface.

### 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?*

* The system assumes users have basic technical proficiency, such as navigating web interfaces and managing accounts. It also assumes the use of up-to-date browsers for optimal performance.

### 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?*

* The system will not be able to perform offline data updates, as all major modifications must occur online. Additionally, it will depend on cloud services for scalability, and any failure in cloud hosting could limit system functionality.

### 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.*

