# 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?*

* This project’s goal is to build a system for DriverPass, a company that helps students get ready for their driving tests. They want a system where users can take practice exams online, schedule driving lessons, and manage their appointments. DriverPass needs this system to work on both computers and mobile devices and be able to access data online and offline.

### 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 wants to help people pass their driving tests. Many people fail because they don’t practice enough. DriverPass wants to fix this by giving driving lessons, DMV rule lessons, and practice tests online. The system should help schedule lessons, track them, and keep user info safe.

### 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 should let people sign up for training, schedule, change, or cancel lessons online.
* Different users need different access levels. The IT officer should manage all accounts.
* The system should show who made changes for tracking.
* It should send alerts when DMV test rules change.
* Users should be able to reset their passwords easily.
* The system should work on the web and use the cloud for security and backups.
* The interface should show test progress (test name, time, score, and status).

## 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 must work quickly on web browsers and mobile devices, with actions taking no more than 2 seconds. Updates should happen every few months to keep the system reliable and up to date.

#### 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 must work on Windows, macOS, iOS, and Android. It will use a cloud-based database to securely store and access data.

#### 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?*

* The system must clearly separate customer, IT, and management access with role based permissions. Passwords must be case-sensitive. If a user enters the wrong password three times, the system should alert the admin to possible issues.

#### 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?*

* Admins should be able to add, remove, or update users easily without needing to change the code. The system must adjust to platform updates to stay compatible.

#### 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?*

* Users will log in with secure passwords, and the system will use HTTPS to protect all data transfers. If a user enters the wrong password three times, their account will lock temporarily, and they can reset the password through email.

### 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 during login.
* The system shall allow users to schedule, modify, or cancel driving appointments.
* The system shall track practice exam results, showing test name, time, score, and status.
* The system shall send notifications for completed transactions and appointment confirmations.

### 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 interface will be accessible via web browsers and mobile applications.
  + **Customers**: Should be able to register, log in, book lessons, take practice exams, view progress, and receive notifications.
  + **IT Staff**: Should have tools to manage accounts, permissions, and system configurations.
  + **Management**: Should be able to view reports, track progress metrics, and update administrative settings.

### 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 probably have access to a reliable internet and modern browsers to access the system.
* The system will most likely rely on an external payment service for secure financial transactions.

### 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?*

* Budget restrictions may restrict advanced personalization features during the first rollout.
* The system might not work offline at launch.