# CS 255 Business Requirements

## 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 the client and is a company that provides students with online practice exams and on-the-road training, helping them prepare better for driving tests.

### 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 is a company that enables students to take practice exams for DMV driving tests both online and offline. The system requires a computer, cellphone, tablet, or iPad.

### 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?*

* Offer on-the-road training sessions.
* Enable both online and offline access to personal data.
* Access data from any computer or mobile device via the internet.
* Download reports and specific information using applications like Excel.
* Implement a system to identify the scheduled driver for each customer, considering the variety of drivers and cars. This involves tracking the user matched with a specific driver, time, and car.
* Allow customers to make reservations online using their accounts.

## 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 application should connect to an online server while retaining access to documents offline if they are downloaded.
* The application must feature straightforward navigation, enabling users to easily access various pages and videos, with fast page-to-page loading times.
* Updates to the system should be implemented promptly to reflect changes in DMV policies and practices.

#### 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?*

* Windows and Apple operating systems are among the most widely used platforms.
* A database is required to store user records.
* Springboot

#### 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?*

* Various users will have unique login credentials consisting of a username and password.
* The system must verify that there are no duplicate login names already registered.
* User roles include IT officer, customers, secretary, and owner.
* Input fields should be case sensitive.
* Admin privileges are required to monitor reservations, including who creates, cancels, or modifies them.
* In case of password resets, users must be able to update their information accordingly.

#### 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?*

* Additionally, functionalities such as resetting passwords and updating specific user information will be incorporated.
* The system will support the addition or removal of modules for future releases.
* Built-in capabilities include adding, removing, or modifying features that impact users.

#### 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?*

* Various levels of access will be granted to different employees based on their roles.
* All employees will have unique usernames and passwords for authentication.
* The system will enable authorized personnel to access accounts and implement blocking or restriction measures for others as needed.

### 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.”*

* User credentials must undergo validation upon logging into the system.
* Users should be able to modify personal information such as addresses and appointments as required.
* The system should provide information pertinent to practice exams.
* Regular updates are necessary to ensure that the system remains compliant with DMV requirements.

### 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 user is the head boss, Liam
* Ian is the IT officer
* The user should have the option to book, cancel, and change appointments online
* The registration process involves a phone call, where the customer provides their information. This information consists of their first name, last name, address, phone number, state, and their credit card details

### 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?*

* Diverse language options
* Colorblind-friendly feature
* Visually attractive and user-friendly interface
* Dependent on electricity for operation

### 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?*

* Incorporating or eliminating modules for a forthcoming release
* Resources will need to be sourced from the existing assets of the development team
* The timeline spans from January 22nd to May 10th.

### Gantt Chart

A screenshot of a calendar

Description automatically generated