# CS 255 Business Requirements Document Template

## 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 client is DriverPass. They want us to create a system that allows students to take online classes and practice tests. They also want students to be able to schedule in-person 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?*

* The client noticed a high failure rate with students taking the driving tests. They want students to be better prepared. They would like a system that allows students to be able to take online classes and practice tests. Finally, they would like to let students set up appointments for in-person on-the road driving tests.

### 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 be able to do the following:
* Gives employees access to certain functions. User Access Control
* Allow students to create user accounts and reset passwords.
* Allow students to schedule in-person training and choose different training packages.
* Allow students to track their progress in the user interface.
* Allow students to track their reservations.
* Connect to DMV to stay current with new rules, policies, and sample questions..

## 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 will have support for multiple browsers (e.g., Chrome, Safari, Firefox).
* The system will be supported by multiple OS’s (e.g., Windows, macOS, and Linux)
* The system will run on the cloud (e.g., Azure, AWS)
* Load times will be less than 5 seconds
* The system will be able to handle up to 100 users at a single time.
* The system will be optimized for performance and scalability.
* The system will be updated as needed.

#### 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 compatible with multiple operating platforms (e.g., Windows, macOs, Linux) System should also be able to run on mobile devices (e.g., IOS and Android).
* The system would require a database to store user and system information.

#### 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 would have separate login sections for customers and employees. Usernames and passwords would both be case-sensitive.
* The system should notify the admin as soon as a problem is identified so that the admin can fix the issue with minimal user impact.

#### 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?*

* Changes to users would not require changing code. Changes can be made from the Administrator Dashboard, or the Content Management System. Users would be able to modify their own profiles via Account Settings.
* The system should once an update is detected check for compatibility. The system should also have a rollback feature to be able to revert to stable versions.
* The IT admin needs access to the web-server, cloud-server, database, and file servers.

#### 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?*

* A complex password consisting of a minimum of 12 characters (letters, numbers, and symbols) would be required to log in. As an added security measure multi-factor authentication (MFA) would be required for both customers and employees to confirm identity.
* A secure encrypted connection would be used. (HTTPS as opposed to HTTP) to connect the user to the server.
* The users account would be locked out after 3 unsuccessful login attempts. Would use reCAPTCHA as an added measure to deter against brute force attacks.
* The system will allow users to reset their passwords if they forget them. User would not be able to reuse the past two passwords.

### 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 create an account.
* The system shall validate user credentials.
* The system shall allow users to reset password if forgotten.
* The system shall allow users to select and purchase a package.
* The system shall allow admins to disable package if it’s full.
* The system shall be accessible from mobile and PC devices.
* The system shall allow users to track their progress.
* The system shall display driver notes.
* The system shall connect to local DMV to be kept up to date.

### 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 interface should be easy to navigate and user-friendly.
* The user interface should be accessible for users with disabilities.
* The interface will be accessible for new users, instructors, and administrators.
* Users will be able to access course materials, track their progress, communicate with instructors, set-up in-person driving tests.
* Instructors will be able to create and manage courses, communicate with students, and track their progress.
* Administrators will be able to manage user accounts, disable packages that are full, troubleshoot technical issues, and generate reports.
* User will be able to interact via web browsers and mobile devices.

### 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?*

* Assuming that users will understand how to navigate website
* Assuming that user will have a newer smartphone, or personal computer.
* Assuming that user will have 24/7 access to the content

### 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?*

* Scheduling limitations. Only 10 vehicles available. Would need to be able to work around this issue. Limited number of packages due to limited cars.
* Would need to be completed within the 16-week timeframe.

### 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 screenshot of a computer

Description automatically generated