# 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?*

* Client wants to create a web-based, cloud app that allows people who are preparing for their driver’s test to take online courses and tests before the practical.

### 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 says that they see many people failing their driver’s test at the DMV. DriverPass aims to allow users to take practice tests and online courses in preparation for their driver’s test.
* Components should include:
  + Registration and payment form
  + User Roles defined and access restrictions
  + Driving lesson scheduling
  + Progress tracking
  + DMV integration

### 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 registration database should include name, address, phone number, state, credit card number, expiration date, and security code.
* The app should be accessible anywhere
* User Roles need to be specified, Ian (IT) requires full access to reset passwords, remove, and block accounts
* Reservation tracking system, lessons are 2 hours long.
* Identify the driver customer is scheduled with
* Driving appointments scheduled, 10 cars, three training packages:
  + Package 1: 6 hours of driving
  + Package 2: 8 hours of driving + DMV Lesson
  + Package 3: 12 hours of driving + DMV Lesson, online courses and test
* Lesson history and test progress displayed
* Contact Us Page
* DMV integration component so that staff will be notified when DMV rules or questions are updated.

## 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.*

**Security** – Protecting personal data from intrusions

**Reliability** – The system needs to maintain a reliable connection in order for students to access their information and the system and not lose any progress.

**Scalability –** It is important that the system support an always increasing userbase and the ability to easily upgrade modules and components.

**Compatibility** – It is crucial for the application to work with different operating systems and platforms. These include but not limited to desktops, mobile phones, android OS, Apple IOS.

**Availability** – An application of this magnitude must be able to be accessed 99 to 100% of the time by users. Downtime should be minimized.

#### 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?*

* Apple IOS, Android OS, Windows. Web Browsers such as Edge, Safari, Chrome.

**Speed -** Ideally, a page should load in under 2 seconds. A response to users that a page is loading should appear within 100ms according to MDN Web Docs (n.d.).

**Updates –** It’s important for updates to the application be done regularly. The DMV interface needs to maintain updated information for the learning and testing materials to be accurate.

#### 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 able to run on Windows, macOS, Linux, and mobile devices such as Android and iOS. Since it will be a web-based application, the application will be built using a cloud-based server, which will support scalability. The backend will need APIs to connect to external interfaces such as the DMV; also, web-based databases, such as Supabase, should be considered.

#### 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 be required to enter a unique email address, and the input will be case sensitive. Two-factor authentication is also a good way to provide a more secure user connection.

**Administrators will be notified of a problem:**

* If the 2FA fails several times
* Failure to connect to the DMV interface
* If the system fails or crashes
* If there is an attempted breach or actual breach

#### 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 the user can be made without changing the code. This code should already be implemented during the initial development. This can also be done by accessing the user database on the cloud server, and because this is a cloud-based application, cloud-based deployments can be utilized to assist with these smooth updates.
* Platform updates should be smooth and not require any excessive downtime.
* The IT admin will need top-level permissions such as admin cloud panel access.

#### 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 Requirements:**

User email and password

**Secure Connection:**

A secure connection between the client and server can be established by using HTTPS/TLS and JSON Web Token.

**Brute Force Hacking Attempt:**

If a brute-force hacking attempt occurs, the user account should be locked, and the

administrator should be notified.

**User Forgets Password:**

The user can either click on the “forgot password” button, which will notify the system and

Administrator that the user needs to reset their password, or the user can fail to log in a certain

number of times, and the account will be locked until the user resets their password.

### 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 authenticate and validate user credentials
* The system shall allow users to sign up and enter their personal information
* The system shall allow users to select and sign up for different training packages
* The system shall allow users to cancel their training packages at any time
* The system shall allow users’ information to be entered by the secretary
* The system shall allow users to contact certain employees via online chat, email
* The system shall allow users to reset their passwords
* The system shall provide activity reports to track appointments
* The system shall track the progress of classes, tests, etc.
* The system shall notify the admin of DMV updates
* The system shall allow admin to disable packages
* The system shall allow admin to block those who are no longer employees.

### 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 needs to be responsive, easy to navigate, and work across different devices and platforms. Support secure user authentication and real-time updates and allow users to easily access the online learning and booking options.

**Types of Users:**

Students

Administrators

Instructors

Secretary

**Students:** I would expect most students to interact with the application via mobile, but I am sure some students will use laptops or desktops as well. Students will need to be able to sign up/login, schedule training lessons, access training and testing materials. Students should be able to contact faculty via chat or email.

**Administrators:** Administrators will need access via mobile and desktop. They will need to be able to disable packages, manage user access, receive update notifications from DMV, and block access if necessary.

**Instructors:**  Access should be on mobile and desktop. Instructors will need to check the scheduling, confirm scheduling, and be able to contact students. They should also be able to check and document student progress.

**Secretary: Access** will be mostly on a desktop/laptop. The secretary should have access to the scheduling of driving lessons and student progress.

### 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?*

* I am assuming that all students have internet access and access to an internet accessible device. There is no offline access; everything must be done while connected to the internet. I am also assuming that users will have the basic know-how of how to navigate the application and web competently.

### 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?*

* Cloud servers are great ways to provide security and storage but the more you scale the application the more expensive this can become.
* Accessing the application requires a stable internet connection
* Depending on the time frame specifications, it may not be possible to have all requested aspects of the application available at launch.

### Gantt Chart

A white board with many colored boxes

AI-generated content may be incorrect.