# CS 255 Business Requirements Document Template

Complete this template by replacing the bracketed text with the relevant information.

This template lays out all the different sections that you need to complete for Project One. Each section has guiding questions to prompt your thinking. These questions are meant to guide your initial responses to each area. You are encouraged to go beyond these questions using what you have learned in your readings. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead, the goal is to complete each section based on your client’s needs.

**Tip:** You should respond in a bulleted list for each section. This will make your thoughts easier to reference when you move into the design phase for Project Two. One starter bullet has been provided for you in each section, but you will need to add more.

## 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 – DriverPass. DriverPass wants a system that can provide various driving lessons packages for a designated price. The system should be a website where customer can take online test and track their progress. The system has multiple different capabilities which include but are not limited to creating appointments, looking at the data offline (this could be exportable data), and updating the data if deem necessary.

### 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 bridge the gap between new drivers and their learning experience as they go through lessons to get their driver license.
* DriverPass pass notice there was a lack in proper driving test preparation.
* DriverPass needs multiple different components such as a Web Interface or Frontend, Database, Hosting Service, etc.

### 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?*

* Once the system is completed it should be able to let users create accounts, appointments and see their progress. The system will also have the functionality to let DriverPass see online information with the ability to download the information in excel format. The DriverPass team should have the ability to give/denied access, to user/employees. DriverPass can also adjust, create, or delate appointments.

## 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 should be able to run mainly on the web. Also, the system will branch out to mobile applications such as IOS and Android.
* The system should be very fast, ideally using asynchronous request and AJAX to be able to provide the user with instant feedback as they go through lessons.
* The system should be updated under the following circumstances.
  + DriverPass request of new feature
  + Security Update
  + Bug Fixes
  + User Interface Changes

#### 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?*

* Since the system will be mainly web based it will be able to run on any browser including but not limited to Safari, Chrome, Firefox, and Edge.
* The system should be built with responsiveness in mind, so it works the perfectly on any device including mobiles.
* The backend usually requires somewhere to store the information. A database such as PostgreSQL or MongoDB should suffice. We can the connect this database through a REST endpoint to any other interface.
* As the backend we can opt to use WebSocket’s and regular HTTPS request to have a live dashboard that user can constantly see without refreshing the browser/app and avoiding multiple requests to the backend.

#### 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?*

* Each user will have the username and password. Username must be unique that way we can avoid trouble, and data mixing.
* The system should have a logging system. The logging system will oversee watching all activity within the system. If there is an issue, we can have the system send an email to the admin with the log files, so the admin has a good understanding of what caused the 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?*

* Changes can be made to the user without changing code. There will be an interface to perform CRUD functions and mostly all elements of the system.
* Usually, the platform has deprecation warnings that can be used to ensure the system is compliant with future versions of the platforms.
* The IT Admin should have a high-level access. The reason for this is so the IT admin can give and remove permission to other employees/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?*

* There are multiple ways to avoid brute force attacks. An example will be to force the user to reset their password after a set number of incorrect tries.
* 2FA can be enabled as an extra security layer.
* To protect the data exchange is important that we validate all request, implement authentication and authorization.
* Request must have a bearer token that is tied to the current logged users, if the token is not valid the request will not go through. We can also encrypt the data to avoid issues. A common usage for account logins is JWT tokens.
* If the user forgets their password, they would be able to reset the password through the link sent to their 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 create a JWT token with the user’s credentials for safety purposes.
* The system should lock the user account after a set number of incorrect tries.
* The system shall log all activity.
* The system shall send email with a password reset link if the user requested it or the account is locked by too many invalid attempts.
* The system shall track all information and their progress from the user.
* The system shall notify the IT Admin when suspicious action happens in the system.

### 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 must display all the functionality requested by DriverPass.
* It must meet the accessibility rules and be easy to use.
* User will be able to access on any browser and/or device.
* User will be able to see all their information, progress, create appointment and learn through the system.
* Admin should be able to see all the information related to the users and employees. Admins will also be able to see the activity and logs of the applications.
* The users will be interacting through the web interface which will have a very simple tutorial on how to navigate the 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?*

* It is assumed that the users have an internet connection.
* It is assumed that the user has email and a phone number.
* It is assumed that the users know the basic of using a device that support web browsing.

### 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 limitation might be on the speed a mobile device can represent the application due to the limited real state the screen has.
* Other limitation could depend on breaking changes of the browser.
* Incompatibility with different or older browser. (Yes Internet Explorer, I’m looking at you)

### 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.*

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence