# 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?*

* The client is DriverPass.
* The purpose of the project is to provide better driver training.

### 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 system should be able to accomplish all the goals stated in the “Objectives and Goals” section.
* The system aims to provide better driver training to customers through online classes, practice tests, and on-the-road training through different packages.
* The system’s backup and security needs to be taken care of for DriverPass.

### 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 access their data from anywhere.
* The company has different employees with different rights and roles. As such, the system should have a role management system with varying levels of privileges.
  + The owner should be able to have full access over all accounts so they can reset them in case of a forgotten password or blocking someone’s account access if they’re let go from the company.
  + The IT officer should be able to maintain the system, be able to modify it, etc.
  + The secretary should be able to make appointments.
  + Users should be able to make appointments, cancel, and modify appointments online.
* There should be three packages to start off with. A goal is to allow the company to easily modify packages and add more. Each driving lesson is two hours long.
  + Package One includes six hours in a car with a trainer.
  + Package Two includes eight hours in a car with a trainer and an in-person lesson where DriverPass explains the DMV rules and policies.
  + Package Three includes twelve hours in a car with a trainer, an in-person lesson where DriverPass explains the DMV rules and policies, plus access to their online class with all the content and material. The online class also includes practice tests.
* Registration happens through a phone call where the customer gives information such as their first name, last name, address, phone number, state, as well as their credit card number, expiration date, and security code. It should also include the pickup location from where the customer wants to be picked up, as well as a drop-off location, which should be the same as the pick-up location.
* Have a notification system tied to the DMV so that any changes made to rules, policies, or sample questions would notify DriverPass so that they can stay compliant.

## 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 application is web-based, meaning it can run from the web browser.
* If the app is a progressive web app, then it can be run offline in the web browser, even if not connected to a network.
* The system should be updated when new features or security patches become available from the developers—there is no need to delay updates except for testing to make sure that they do not break the existing system.

#### 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 all modern web browsers, including Google Chrome, Firefox, Opera, Safari, and modern mobile web-browsers.
* The back end requires a database to store all the custom information in the system, including customers, users, management, etc.
* The system should be hosted from a Linux server as the licensing is free compared to Windows, and extremely resource optimized. The system will only use what is needed, with a negligible overhead.

#### 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?*

* Different users have different information, such as their phone number, name, address, billing information, their role—if they work at DriverPass, and appointments.
* The only case sensitive input that’s needed is the user’s password, but everything else (name, address, billing information) can be case insensitive.
* The system should inform the administrator(s) of a problem when there’s a back-end error, such as a customer not being charged properly, appointments not being assigned, etc.

#### 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?*

* Users can be added, removed, and modified with the system, as in the back end, it should just be a matter of performing simple database queries.
* The system will adapt to platform updates easily, so long as the code is written with cross-platform capabilities in mind (not using platform-specific libraries and code; sticking to standard libraries) and the database is transferred with the same configuration as before.
* The IT admin should have access to the server hosting the database as well as the site code. This is a high level of trust; they have access to all the live production data of DriverPass, but it is required if they need to fix an aspect of it or debug for issues to pass along to developers.

#### 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?*

* The user can sign in with their phone number and a password, as registration occurs via phone.
* The connection between the host and client can be secured using TLS v1.2, otherwise known as HTTPS. This means a connection is completely secure from the client to the host, and vice versa.
* If there is a brute force hacking attempt, the system can automatically add a delay to signing in, or if the brute force attempts are high, block connection IPs for a certain amount of time, like a few hours up, a day, a week, or permanently. In cases like that, if it’s a false-positive (for whatever reason) support should be able to unblock the IPs.
* If the user forgets their password, they should be able to reset it by phone. If that is not viable, they should be able to provide their personal information including the billing information used. If all else fails, they can contact support to assist them.

### 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 when logging in.
* The system shall allow users to manage their appointments.
* The system shall allow administrators to add, remove, and modify users and their appointments.

### 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 will be used by customers and administrators.
* The user needs to be able to sign in, reset their password, modify their own information, make reservations for appointments, pay, etc.
* The interface should be responsive and adapt for all types of screen sizes. For example, if the screen is below a certain width (600px being a common breakpoint width), switch to a mobile layout of the same page with the same information, so that the system is completely accessible from any device.
* The interface should be able to run on all modern browsers, but if support for older browsers (such as Internet Explorer) is feasible without making it harder for developers to maintain, good! This will have to be determined during the development process, but supporting Internet Explorer is not a requirement as it has been deprecated for a long time.

### 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?*

* The hardware each user has varies from person to person, so it is assumed that they have a device capable of running a web browser.
* As the design assumes that the platform for the back end will be Linux (though it is certainly adaptable for Windows, macOS and other environments), it should have enough resources to compile the front end (unless handled during the CI workflow).
* The user will have to set their password at some point—it’s not addressed here. Maybe they can do it on the phone when registering, or maybe in the interface there can be a page that allows the user to input a password and receive an automated call to verify? If not that, then add support for email addresses and authenticate that way.

### 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?*

* If the front end is built using JavaScript and React, the CI server needs to be able to build it, potentially taking up hundreds of megabytes of memory.
* A Linux server should barely dent the budget—a VPS in New York from a company costs $6.50/mo. Choosing Windows would add additional fees for the licensing.

### 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.*

GANTT Chart
