# 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?*

* Fill the void in the market for drivers training apps
* Better Driver’s training
* Create the ability for online classes, testing, and practice

### 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?*

* System needs to be accessible (downloadable) from anywhere, even offline
* Needs to have different system privileges and access
* Need to track users’ actions including changes, updates, sign-ins, etc.
* Customer’s need to be able to make reservations for driving lessons.

### 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?*

* System should be able to communicate reservations to both the customer and to the DriverPass team when reservations are made
* Drivers, customers, and cars need to be able to be reserved out independently so that there is no overlapping with scheduling.
* Customers should be able to schedule drives from one of three packages that are offered without double booking a car or a trainer. Each drive is 2 hours long and the packages differ between the # of drives offered
* Ability to disable a package as an offer in the future
* 1 Main page
  + Show the customer their progress of the package they have purchased
  + Show any of the drivers notes that were taken during drive times
  + Show the user information on the home screen
* 1 Student/User settings page
  + Student can fill in information such as first name, last name, address, etc.
* 1 Page for contacting DrivePass and for a way to contact the students

## 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?*

* System needs to run off the web, preferably over the cloud.
* Minimal updates, downtime, technical problems
* The system needs to be relatively quick so that the students and trainers can get access to the info that they need quickly
* System does not need to be updated regularly, maybe only for bug fixes or additional features in the distant future.

#### 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?*

* Application should run on all platforms through the cloud and will be served up through a web browser. Coverage over the most popular web browsers (google chrome, internet explorer, firefox, safari, etc.) should be easy enough to build in one shot and deploy over a website.
* There needs to be a database to store the information about the customers/users and the trainers as well as all of the static information that will be served up on the site.

#### 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?*

* Unique IDs can be generated upon completion of an application and can be tied to a SSN or other unique ID so that there are no duplicate users.
* Input for the application will not be case sensitive and will be standard formatted before being saved to the database.
* The system should be able to inform the admins of a problem upon the user’s request, so there will be a help me button with a form so they can briefly explain their problem and put their contact information so they can keep in touch.

#### 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?*

* Any changes will always technically change the code, however the functionality of everything will always remain the same.
* Admins need direct access to the entire database, whereas the users will only have direct access to the information within their logged in account.
* The system should never really be changed too much after the initial build and good programming principles would make it so that there is only functionality being added and never changed too much.

#### 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 needs a verified email/username and associated password to successfully login to their account
* Brute force hacking can be mitigated by setting a limit to the # of attempts that can be made to logging to one’s account in a given amount of time
* If the user forgets their password, they can retrieve a password change form to their email after a successful 2FA to their phone or 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 validate user credentials when logging in
* The system shall load the logged in user’s information and display it to the page
* The system shall respond to any of the user’s input appropriately
* The system shall save the user’s information to the database upon a successful application
* The system shall notify administrators of unsuccessful login attempts
* The system shall save any updates the user makes to the database
* The system shall log out and clear the browsers cache upon a logout event
* The system shall send a verification email to the user upon application to 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.)?*

* User – This is the base level user that has no privileges or associated accounts. This could be a first-time user looking to apply for an account or a user that has not logged into their account. They should see the marketing and login/create an account section and nothing more that that. This is where every single new user should start
* Customer – After a user successfully creates an account, they become a user where they can log in and out of their account as well as see all the relevant personal information on their dashboard. From here they can update their personal information and schedule drives as well as see their progress on their package.
* Admin – The admin is a user with special prideless that gives an easy user interface directly tied to the entire back end so they can make edits and adjustments to any of the customers within the system. There should be extra security on these accounts because they have a lot of power over the entire system and could cause a lot of trouble if a bad actor were to gain access.

### 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 type of technology that the system should be accessible from was not specified but inferred as any modern web browser since we know the application should be hosted on the web. Knowing this we should make an application that works on both desktop and mobile browsers, one that can update the formatting automatically.

### 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 number of customers/admins will always have a hard limit. Although the system may never reach this limit, this will always be a limitation since information and data take up space and we can’t be sure that there is an infinite amount of space in the universe.
* Another limitation will the speed and the storage of the cloud service that the budget can afford. Nearly unlimited bandwidth and storage are available from some of the top cloud service providers, but the real limitation will be the budget of the company.
* Finally, there will be a time limit to the amount of features that can be added to this application.

### Gantt Chart

