# 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, DriverPass, wants a system designed around their current business model. DriverPass offers driver tests and practice to students who hope to get their driver's license from the DMV. The system that we're going to design will let them enlarge their business to let users access online reservations, scheduling, and payments. It will also grant DriverPass employees’ access to user data and schedules and handle the scheduling of drivers to users.]

### 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 has noticed a possible niche in the market: the drivers who go to the DMV to take their driver skills test are not prepared and end up not passing. DriverPass looks to better prepare drivers by providing them with higher quality practice tests and driver training so that the users are well-equipped to pass these tests. For that, the system would need to be able to store user information in accounts, to enable log-in, and to allow users to sign up for the test or driving practice. It will allow users to trace their progress and, upon ordering services, make relevant payments online. It will also help DriverPass trace their own employees and the cars they drive for training purposes, as well as the users with their data. The system is based in the cloud and accessible from any kind of device with an internet connection.]

### 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?*

* [

|  |  |
| --- | --- |
| Objective | Task |
| Housing of database of users | Create a function allowing users of the system to input personal information into a secure account they can log into. This will allow them to have access to their information and actively participate in the creation of better policies for themselves. This would involve creating secure data base which is housed in the cloud. |
| Intuitive User Interface | Get customer approval for overall design before user interface can be built. Design a UI that will give the client-user information about their account, including their personal details and progress tracking, along with the notes their driver has scheduled for them on services and their photo. Create a UI for the admin which will show the admin-user what options for scheduling and car choice were chosen, as well as user payments that were processed and other user client-user info. |
| Design a functionality that auto-matches cars/schedules for DriverPass employees. | Create a function that arranges employees and cars existing in DriverPass. Write a function that pairs available employees with available cars and books them to train drivers on days that were requested by the user. Allow the user to over-ride the automatic choice through a modification function. |
| DriverPass Employee roles | Add a parameter that identifies employee accounts from normal user accounts. Add a parameter for each kind of employee that will use the system along with the permissions associated with that kind. |
| Automatic system updates from DMV | Design a function that periodically checks for DMV updates through the DMV website/ database. Develop a notification service, which updates administrator accounts of DMV updates |
| Cloud access for all users | Design the system to be used in the cloud. Develop a function to compile specific data sets to be downloaded to a csv spreadsheet file. Write a function that uploads edited data to the cloud and updates cloud data with the newest edits. |
| User profile interactions for scheduling, pay, etc. | Write a function allowing users to view schedule of services purchased, edit when needed, cancel, or pay for services. Users will be allowed to purchase any of three different packages offered. Write a function to display user progress on a "dashboard" type of UI. Design a service that will hook into an online map repository to set pickup and drop-off locations whenever a customer purchases a service. |
| Link DB to interface | Build UI based on research provided by Toni and Clark. Prepare UI to be linked to DB. |
| Test System | The testing team will use both static and dynamic tests to assure that the system works properly. 10 days. |
| Final Deliverable | Product System testing with test-team shall be conducted before the product can be delivered. |

]

## 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 DriverPass system shall be run on both PC's/Mac's and smart phones. To this end, a web app should be developed to run in a typical PC web browser—like Firefox or Chrome—and a smartphone app that will handle mobile access. It will also be expected to run fast enough to support video streaming, as that is one component of the DriverPass business model. Ultimately, user bandwidth will determine the quality of the stream; however, the system should handle multiple users streaming driver's education videos. It will at least need updating at the same frequency as the web browsers through which it is accessed are updated, and whenever the mobile operating systems of its associated smartphones are updated. This means that when Google Chrome pushes an update to its users, DriverPass' web app should be tested on the new update; necessary fixes should be made, followed by a system update in support of the browser update.]

#### 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?*

* [Provided the DriverPass system is developed to work in web browsers using HTML as the delivery language, it will then support any web browser-supporting operating system by default. At a minimum, Android and iOS will have to be supported by the mobile application. Other niche mobile operating systems exist, but marketing should focus on having those users use the web application until a dedicated application can be developed—provided market research supports this. A database will have to be developed and connected to DriverPass' interface to store information about the users, clients, and administrators - drivers. A tool for backend development will also be necessary, depending on the preference of the developers, like ASP.NET Core or Node.Js. For the server to be able to make calls to a database, SQL is something standard.]

#### 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?*

* [Every user will have a unique username that shall be applied when log on to the Website. All these usernames are kept in a database such that whenever a new user is creating an account with a username that is already in use, it will immediately stop creation of the account and alert the user about this error. Usernames and passwords are case sensitive. Examples include "Bob123" not being the same as "bob123". Any error that is logged in the system should also be forwarded to admin as well as added to a log. Events to notify admin about include processes consuming more resources than expected, improper use of CPU resources, memory issues due to cache size, zombie processes, load average problems, or problems with disk read and write speeds.]

#### 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?*

* [Since users will be objects created within the system and held by the database, the ability to add/remove/modify users while keeping the system code intact will remain. Done properly, the interface will save the need to essentially refactor the entire system just to update one class. The IT team will need administrator rights to do so. They will have authorization to the username and some information of users, that is they can update certain user information, but they are unable to view the information that will be altered. For example, a customer may want toupdate payment information; IT support will be able to receive updated payment information andinsert itinto the current account of the user, but they will not be able to view the old payment information, nor will they view the updated information after he has itupdated in the system.]

#### 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?*

* [They will be prompted to create a unique User ID, with a "strong" password consisting of at least standard character length 10+, containing capitals, lowercase letters, numbers, and symbols. Transfer of all data over HTTPS using TLS to protect the data. Multiple attempts for a user to log into their account if they forget their password but after a number of attempts their account will lock and they will be required to contact IT to have their password reset either directly or via the "forgot password" link. The above policy will prevent brute force attack from outside the system via user log-in.]

### 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 at log-in.
* The system shall provide two-factor authentication for all users.
* The system shall track user progress in regard to testing.
* The system shall store user information in a private account, including driver notes, special needs, driver photo, address information, and payment information.
* The system shall permit user scheduling of a driver test and driving practice through one of three pre-loaded packages.
* The system shall allow users to make payments online.
* It will maintain the records of the business's employees.
* It will keep a record of when cars employees drive are scheduled for use.
* It will track the users' data and progress.]

### 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.)?*

* [It will be accessible through all types of devices which are connected to the internet, and which allow a user interface, be it a phone, a tablet, or a PC. It shall consist of the customer-user, the employee-user, and the administrator-user. For customer-users, the application's home screen shall feature a login through the input of credentials. The user shall check account status and progress, view the account balance, and schedule driver tests or practice. The user shall be capable of updating their account and profile, even including billing information. The employees can view their schedules and be able to take cars out. The employee can access his or her employee profile to make any necessary changes to the availability and everything else. Admins would be able to access most of the user and employee details. They would be able to reset passwords and unlock accounts, view schedule of employees, modify it, and add / remove employees and any other user from the system. Items related to driver-test issued by DMV would help keep the admin informed. The admin will, therefore, be able to schedule appointments for driver testing and practice for any other user. The interface will be touch- or click-based, with keyboard support for typing in names, and other pertinent information, and will be simple and user-friendly.]

### 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 all users will access the application via a modern device, either through a smartphone or PC, and through an updated web browser. It is not yet known in which coding language the application shall be written. What is yet to be known is the color scheme or overall aesthetic of the final site and settings that exist to make their experience better on their preferred device, such as color-blind friendly, slow-connection friendly, or high-contrast view. The videos of the driver tests have not been decided on for storage yet. If they are stored on the DriverPass server, then that will take up bandwidth and space. Being hosted on a free-use third-party site will reduce bandwidth but possibly bring on extra costs or other issues.]

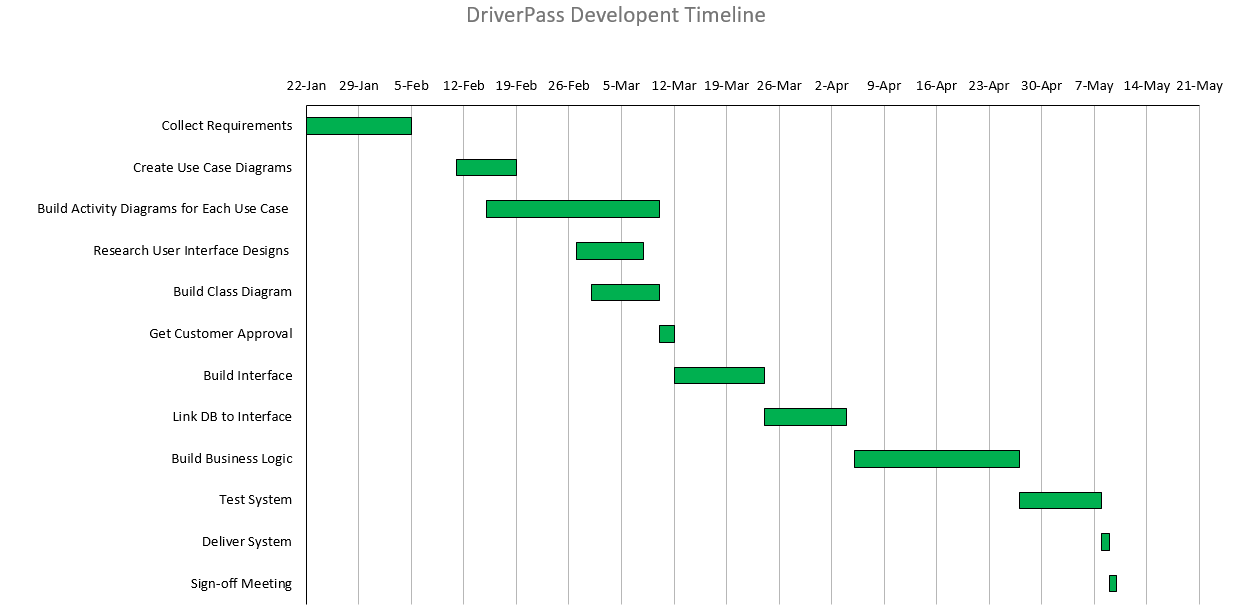
### Limitations

*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?*

* [Some of the limitations will be based on the potential time frame versus team size. If the development team is smaller, putting together an entire system in only six months could lead to some features being cut or minimized. Depending on the database host and their allotted bandwidth, user interface would need to be adjusted to include lower bandwidths with smaller pictures or less animations. Since the project budget has not yet been established at this point, a meeting regarding it with the DriverPass CEO could occur in case of an overspend so that the requirements may be adjusted or increased funding established.]

### 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.*

[]