# 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?*

* Liam is the current of *DriverPass*, our client. Liam wants *DriverPass* to train students for their driving tests at their DMV. He wants to take advantage of the fact that there’s a void in this market.

### 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* wants to provide driving training for their users. The problem they want to fix is the percentage of people failing their driving test at the DMV. Some of the components needed for this system are taking online classes, taking practice tests, and on-the-road training.

### 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?*

* When the system is completed, the user should be able to register themselves to DriverPass. From there they make appointments to go out with a DriverPass driver and car for on-the-road training, modify the appointment, and take online classes. Measurable tasks would be utilizing UML diagrams to visualize the system. Then break up each component to a sprint and review functionality after each sprint is completed.

## 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?*

* This needs to be in a web-based environment of course and over the cloud but Liam also wants to access data from any mobile device as well. There should also be an accompanying mobile app also. We can something like Uber and Lyft.

#### 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?*

* Platforms we should focus on for are Windows, Linux, Mac and mobile. Yes, the back end requires multiple tools but for a database I suggest a SQL database. It’s an excellent database language and overall, the 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?*

* The users will create their own profiles that have them create a username, password and enter their email address. All those 3 are going to be case-sensitive, if a username is already taken a new user can’t use the same one. To log in the user must enter their username or email and their password. The system should flag an admin when there is any problem, but the problems can be put into categories going from minor to critical.

#### 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?*

* Yes, we can just have reusable functions for those features and for most of the program. This way when there is an update to the program it doesn’t disrupt the entire system. We can have a hierarchy system where an IT admin have access to system and user information. They can also have permissions to change the program.

#### 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?*

* To log in the user must enter their username or email and their password. We use a system like auth0 to secure data exchange between the client and server. We can also implement a two-factor verification system. If there is a “brute force” hacking attempt that immediately raise a critical flag to the admins so the team can resolve the as soon as possible. We will have a “forgot password” option for those users who forgot their password. Asking them to enter their email and validate other personal information.

### 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 absolutely validate user credentials. I’ll list some of the functional requirements, create unique users, allow users to sign up and book lessons, allow users to book full packages, have a hierarchy system in place to differentiate between users, instructors, and admins, cancel and modify appointments, have two-factor authentication, have a SQL database that keeps all system information.

### 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 different users for the system will be users, instructors, admins, root users. Users are the “customers”, they’ll be able to create a profile, book, modify and cancel appointments. Instructors will be able to accept lessons, modify and cancel appointments, end lessons, check out cars, and put down notes on each lesson/user. Admins will have higher permissions and access. They’ll be able to modify most of the code, alter any other users under them, and change the architecture of the system. Root users will have access to every part of the system without limits.

### 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?*

* That the client wants all the information we have stored linked to a database, client will assign hierarchy of anything above a user, and we’ll use AWS for cloud.

### 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 technology aspect of this project is doable, no limitations there, I see our budget as our limitation since we’re relying on a 3rd party for security and cloud storage. Also, physical resources will be an issue. If we have an influx of customers, we’re going to need an abundant amount of additional cars.

### 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.*

*Timeline, calendar

Description automatically generated*