# 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?*

* For this project, the client, driverPass, is looking to capitalize on a niche within the community of people interested in getting their driver’s license. They are looking to construct a system where users can receive driving instruction and feedback to in theory raise their odds of doing well on their driver’s exam.

### 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 to be constructed should be able to first and foremost schedule users with instructors to receive feedback on their driving. driverPass is trying to fill in a hole in the market where future driver’s license test takers are not receiving formal instruction prior to their exam. This will require an application which has a web component which the users will interact with. A scheduling/reservation component where users can schedule appointments. User components where they can create profiles, login, and other functionalities like recovering/resetting user information. There also needs to be a comments and rating components for rides taken so the instructors can leave comments. Finally, a component for contacting driverPass is also a need for development.

### 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 completed product for driverPass will be able to schedule users with instructors and receive feedback on their sessions. This product will also have basic user function in terms of creating and updating accounts. The measurable tasks for this system will be in the collecting requirements, creating use case diagrams, creating activity diagrams, researching user UI designs, building class diagrams, getting customer approval, building the interface, linking the database to the interface, building the business logic, testing the system, delivering the system, and conducting the sign-off meeting.

## 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 should be able to run efficiently on all modern-day web browsers and should be accessible on both mobile and desktop technologies.
* The system should run as efficiently as possible with minimal wait times and an appropriate use of progressively loading of the web application page. This means loading of static elements rapidly and show loading markers for dynamic information which may take a little longer.
* The system should be updated on a per-sprint basis for new features/updates (assuming an agile work environment – if not, anywhere between 1-3 months for fully new features). While security features should be rolled out as soon as possible as to minimize risk from attacks and data leaks.

#### 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?*

* This web-based application should run on all major platforms, such as, windows, Unix, macOS, Linux, and other relevant platforms.
* Any system that has user login information will need a back end to store user information. Beyond this, scheduling information will also need to be stored. To attain something like this, a MySQL or MongoDB database paired with something to help facilitate the transfer of information to and from the database, such as Node.js. Data validation and cleaning can also be done on the back end to ensure that only acceptable data is making it into the database. On top of this, certain security tools can be applied to the back end to help prevent attacks. One tool could be using a library to prevent DOS attacks.

#### 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?*

* Users will be distinguished by a unique username that will indeed be case-sensitive. Users will also have an admin flag which will be a Boolean value which helps the system determine what version of the website to render and provide necessary restrictions where necessary.
* There should be a system in place that notifies the IT team whenever there is an issue with the application. It is critical that their issues are dealt with as soon as possible so a real time error messaging system is necessary.

#### 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?*

* The system should have specialized functions that deal with the creation, updating, and mutating of user information already in place. These functions will make up the user management systems. Thus, when needing to perform any CRUD operations on user accounts, nothing in the codebase will need to be altered, just a request will need to be made to the back end server.
* Through the practice of breaking the application down into separate components, updates can be made without changing more code than needed. Also, testing will need to be performed on the added code as to ensure that any updates do not compromise the production application.
* Due to the nature of IT, the IT admin will need full access to the back end components of the application. They will need to be able to access the server which the application is being hosted on, as well as the server which hosts the back end code/database.

#### 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?*

* A username and password will be required for a login attempt. If deemed necessary, two factor authentication can be implemented to further secure the application.
* Data encryption should be used when trying to match user input to a created user in the database. This way if there is a man in the middle attack they will not be able to easily decrypt and analyze the information in the request.
* The application should only allow a certain number of login attempts before it blocks a user from further attempts. This could also be monitored by only allowing a certain number of requests to be put into a back end server from a single user, this could look like rate-limiting on Api requests.
* There should be user account recovery pathways for users which have forgotten login information. Normally executed through a recovery email/phone number pathway and authentication codes.

### 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 be accessible to modern day web browsers on mobile and laptop
* The system shall have a report downloading system for offline digestion of information.
* The system shall be compliant with DMV rules and regulations.
* The system shall be able to create, update, and delete users in the systems.
* The system should have an account recovery system for forgotten passwords and usernames.
* The system should be able to manage the creation, updating, and deleting of reservations within their calendar system.
* The system must be able to facilitate tests and show progress on them.
* The system must be able to store user information on tests and display them.
* The system must be able to have a reporting system for instructors.
* The system shall have a robust security system which informs the IT team when errors or breaches in security happen.

### 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 has options to route to online tests, present driver notes, have an accessibility route, a section for both a driver photo and a student photo, user information banner, and a logo which will route back to the home page when clicked. The interface will be used by instructors, students, and admin users. Users will interact with the interface either through a cursor mechanism if on browser, or touchscreen mechanism if on mobile.

### 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?*

* I am assuming users have an internet connection.
* I am assuming a baseline knowledge of navigating web applications.
* I am assuming compliance for government regulations on dealing with user data.
* I am assuming the use of cloud technologies to host the web app and storage of user data.

### 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?*

* For a system which could potentially house thousands of users, scalability could certainly be a limitation if the application is not designed appropriately.
* Another limitation is the dependence on internet connection for almost all functionalities.
* A resource limitation is time due to tight deadlines and even overlapping tasks.
* A resource limitation could be budgetary in not being able to afford the correct cloud services plan that serves the application best.

### 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.*

A screenshot of a project

Description automatically generated