# 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 purpose of this project is to develop a system for the client, DriverPass, that provides driving training for individuals who are learning how to drive.
* The client wants their system to aim for filling a market void by offering online classes, practice tests, and on the road training sessions.

### 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 the system to provide an efficient platform for driver training, enable easy scheduling and tracking of the driving lessons, ensure secure and role-based access control, maintain up-to-date compliance with DMV regulations, and operate on a web-based platform for accessibility and ease of management.
* The problem DriverPass wants to fix is the high failure rate of driving tests.
* There are several different components needed for this system. User management for different roles with specific levels of access. A reservation component for scheduling, canceling and modifying lesson dates. A tracking system for logging changes and creating reports. Training Modules for the classes and tests. A web interface that can be accessible on multiple devices and platforms. And finally, a layer of security for data protection and secure access.

### 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 fully designed and completed, it should be able to allow for customers to schedule, cancel, and modify lessons, track all changes made to records with logs, provide online training resources and practical tests, stay updated on the latest DMV rules and requirements in the area of the user, and operate under a web-based application that operates on a cloud system module. Lastly, the system should have access to data both online and offline.
* The measurable tasks needed for the system to achieve its desired functionality are the following: creation of user roles and access control mechanisms, development of a reservation system with online schedule, implementation of a logging and reporting system that features a changelog, the development of training modules for the classes and practical tests, the setting up of a compliance system that updates with the current DMV regulations that’s user locale specific, the development and deployment of a web-based application that integrates a cloud-bases operation model, and the implementation of data security design.

## 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 system should operate under a web-based application environment.
* Ideally, the system should have a fast response time, meaning, under a second or two.
* The system should be updated regularly. Overall, the system should be updated quarterly or every half of a year. For the DMV rulesets, this system should trigger an update when it notices a change in the DMV rules.

#### 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?*

* Since the system is a web-application, it should run on any platform with access to the internet. This includes mobile devices as well as desktop environments.
* The backend does require a database management system of some sort. This can be MySQL, PostgreSQL, or MariaDB. This is needed for user information, reservation, lesson data, logs, etc.

#### 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 logging into the system will be distinguished based on their roles. These roles include the owners, IT officer, customer, and secretary.
* In many systems, the username is not case sensitive. In this system it should follow suit. However, passwords should be case sensitive.
* The system should inform the admin in cases where data integrity issues arise, unauthorized access attempts, or any errors in the reservation system.

#### 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?*

* No change to the code is needed to add new users or to modify existing users. This can be done through an admin control panel interface with user management capabilities.
* The system should use a standardized web stack like React or Angular, as these are well maintained with a large community behind them for support. The backend should use a common framework like NODE.JS, as it is also easy to update and maintain, and there is also a large community around it. Using commonly used and tests frameworks in a stack can help eliminate any potential bugs that may occur when updating the entire platform.
* The IT admin needs full access to the system, including, but not limited to, the ability to access all user data and system performance and security monitoring.

#### 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 must provide a username and password.
* When connection forms between a client and a server, HTTPs with SSL/TLS encryption should be formed to ensure total security.
* When the event of brute forcing happens, the system should temporarily lock the account down after a certain amount of failed login attempts. This event should trigger a notification to the user and the admin.
* In the event where a user forgets their password, the system should allow them to reset it. This can be done through sending a password reset link to the email address registered under the user’s account.

### 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 make, cancel, and modify reservations.
* The system shall log all changes made to records and generate reports.
* The system shall provide different access levels for different user roles.
* The system shall update with the latest DMV rules and requirements specific to the user locale.
* The system shall operate on a web-based application through any device, mobile or desktop, that has access to the internet.
* The system shall store user and reservation data in a database.
* The system shall notify the admin in case of failed login attempts, data integrity issues, and unauthorized access attempts.
* The system shall lock user accounts down temporarily after multiple failed attempts to login.
* The system shall allow the admin to manage user accounts through a web-interface admin control panel.
* The system shall encrypt all traffic between the client and the server through HTTP SSL/TLS encryption.

### 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 interface needs to be user-friendly, meaning it needs to be easy to follow and navigate through while also being responsive. It should be updated in real time to reflect any changes made to user data, reservation data, DMV rulesets, etc. Lastly, the interface needs to have a log in portal for security for the users.
* The different users and their available actions are the following: Owner, Liam, which has full access to all the data and system configurations, user account data, and the reports and system logs. IT Officer, Ian, who has full access to maintenance features, the ability to manage security settings, access to system performance and system logs, and user data. The secretary, who has the ability to manage reservations and customer information and view and modify lesson schedules. Ideally, the secretary should have a control panel of their own, but this one should not be as detailed as the IT Officer one. And lastly, the customers, who have the ability to create and manage their accounts, schedule, modify, or cancel their lessons, and view class progress and class history.
* Users will interact with the web interface through a web browser on devices that range from mobile environments to desktop environments. This means they need a device that is connected to the internet and updated on all of the standard web packages. This usually comes with the web browser when installing it.

### 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 general user experience was something that was not addressed in the design. Error handling and performance metrics were also not addressed as well. Lastly, third-party integration for payment processing was not specifically addressed.
* The system assumed that the users are technically aligned and are effective at using and navigating through web-applications. It also assumed that users have modern web browsers with reliable internet connections.

### 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 system has limitations in user error. When users enter in the wrong information, there are forces to reach out to an admin to get the information corrected. An error handling system can be built to fix this issue. Security is also limitations are security should be continuously updated to match the modern security standards on the internet. Lastly, the DMV ruleset system relies on the DMV to update their standards, assuming, on their own website. This can be a flawed approach because not every local DMV is going to have a website nor does every DMV keep their online rulesets fully up to date.
* Resource limitations can include time, where the project pipeline might be constrained by the availability of the team members, budgets constraints, which can limit the ability to implement features, and lastly, human resources, which is the availability of skilled developers. Lacking developers can impact the project’s scope and timeline.

### 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 calendar

Description automatically generated