# 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 is DriverPass. They want their system to be able to provide users with access to online driving classes and practice tests, as well as provide a way for both DriverPass and users to make reservations for in-person driving lessons.

### 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 solve the issue of drivers performing poorly on their driving tests. They believe that there is a need to provide better driver training, before drivers take their actual tests at the DMV. There are a handful of components that are needed for this system, but the main components are: user management, a reservation system, a portal for online tests and classes, a way for customers to register, descriptions of their training offerings, DMV integration, and a way to generate reports and action tracking.

### 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 system should be able to handle information provided from the customer. This information will include their first name, last name, address, phone number, state, and credit card information. The system should also be able to take in the customer’s pickup location from where the customer wants to be picked up and dropped off. Additionally, the user should also be able to schedule their own appointments over the internet. The customer should also be able to reset their password automatically if they forget their login information. System also needs to be connected with the DMV so that the tests and policies can be up to date with information provided from the DMV, the system should provide a notification when the DMV has an update.
  + The measurable tasks that need to be included in this system:
    - User management – Users need to be given roles with specific permissions, password resetting, and activity logging for user actions
    - Reservation system – There needs to be a creation of some sort of interface that allows users to make and modify reservations, reservations should be stored in some sort of database, drivers need to be able to be matched with customers
    - Training packages – Each package should include descriptions of the features of the package and a setting that can enable/disable packages so that customers cannot select them when disabled.
    - Registration – Customers should be able to register with their information securely, payment processing and validation via credit cards, customer data storage in a secured location.
    - Online classes and tests – The system should be able to create tests that users can take to practice and a place to host online classes for users.
    - DMV integration – system should be able to receive notifications when the DMV provides an update, system needs to be synced with current DMV regulations.
    - Reports and action tracking – Some users should be able to generate downloadable reports, reports should be for things like user activity and reservation changes/updates.

## 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 be web-based and be able to run on any operating system, including mobile phones.
* The system should run quickly and smoothly, allowing for minimal interruptions for users.
* The system should be updated whenever there is a new update from the DMV regarding any regulation changes.
* Additional updates may be necessary depending on any sort of Package offering changes from DriverPass (IE: adding new packages to the site or removing any packages)
* Additional updates may be necessary for the online driving class platform, but those changes are TBD.

#### 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?*

* The system should be able to run on every platform, including mobile devices.
* A database will be necessary to manage different aspects and information for the system to be able to run.
  + The database will need to store various information regarding clients and drivers for driving lessons.
  + The database will also need to store information regarding online driving courses and packages being offered from DriverPass.
  + The system will also need a reservation system to function. Some of the functions of this reservation system will be things such as: booking packages for clients, allowing clients to book their own packages, assigning drivers to clients, setting time blocks for clients, assigning cars to clients and drivers, the ability to make changes/cancel the reservation, etc.
  + The system also needs tracking information (modifications to reservations/bookings, changes to the system, etc.) to track who made changes and when.
  + The client also wants the system to be cloud-based so that they don’t need to worry about backups and security, those need to be built-in to the system.

#### 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?*

* There should be at least 3 different levels of users:
  + Admin – this would be for the boss and the IT officer. This role would have complete control over the system, allowing them to make changes to the system in the future and allow them to edit things like package offerings and enable/disable packages, and etc. This role would have essentially total control over the system.
  + User/Employee – This role would be for the secretary(s) that are responsible for booking appointments when they receive phone calls. This role would be able to add/modify/remove bookings, create/update customer information, etc.
  + User/Client – This role would be for customers that are booking appointments with DriverPass. The customer should be able to update things like their personal information (name, address, etc.), add/modify/cancel their *own* bookings, set up pickup locations for said bookings, etc. This role should only have access to their own information. This role would also need access to the online class platform if applicable. This user role should be auto-generated via something like a “create account” button on the website.
  + Passwords, customer information (names, addresses, etc) should all be case-sensitive.
  + The system should inform the admin of a problem as soon as it arises. These problems can be things like system outages, issues with bookings, over-bookings on certain days/packages, etc. The admin should be informed via a pop-up on their end of the system or via email, as this would allow for the promptest response from the admin.

#### 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, changes to the user should be able to be made without changing the code for the application/system. The user information should be store within a database. The users of the system should be able to interface with this system via some sort of GUI or user-interface that functions like a program rather than executing things like SQL queries, as this will allow for the most user-friendly experience when updating user information.
* The system should adapt to platform updates by alerting admins whenever an update comes from the DMV, this will allow them to make any necessary updates to the system based on the information they receive from the DMV.
* Additionally, the client has stated that they want the system to be “flexible” so that they can customize certain packages, add new ones, remove old ones. I’ve outlined that these should be options available to the Admin role in previous sections of this paper, but they will still need to bring on a developer to add or remove any of these modules, as there isn’t a way for us to build it to where a non-dev could make these changes easily. This should be noted however as it is stated that we will talk about this feature in a future release.
* The IT admin needs full access to the system, as he is responsible for maintaining it and making any changes to it in the future.

#### 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 to the system, the user should be prompted to enter a username or email address connected to their account, and then a password. Passwords should be case-sensitive and left up to the user to generate, but passwords should be strong as this helps prevent outside access to personal information on user accounts.
* The connection/data exchange between the client and server can be established via HTTPS to encrypt the data as it is exchanged between the client and server.
* If there is a brute force hacking attempt:
  + There should be a limit on failed log in attempts on user accounts to prevent unwanted access.
    - After users hit this limit, they must consult with IT to unlock their account or provide proof of identity (two factor authentication, etc.) to unlock their account.
  + The account should also be locked after a certain number of attempts to help protect accounts from brute force hacking.
  + A CAPTCHA would also help prevent brute force hacking attacks by requiring the user to fill out a CAPTCHA to prevent bots from accessing the account.
* If a user forgets their password:
  + A user should be able to reset their password via a “forgot password” option on the website. This would then send them an email with a link that will allow them to create a new password for their account.
  + In an effort to combat any unwanted access, it might be worth requiring the user to enter in security questions to confirm their identity.
  + The client wants their secretary to be able to reset passwords for users, but I would personally be wary of this, as someone could easily try to impersonate users over the phone to gain access to their accounts. I would personally recommend that users follow the “password reset” option on the website. I would still leave this option available per client request, but highly discourage it without verifying the identity of the user beforehand.

### 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 provide different access levels based on the role the user is assigned to when creating their account in the system.
* The system shall allow new users to create their account and log in once registration is complete using user provided usernames and passwords.
* The system shall provide password recovery and password reset functions.
* The system shall accept and store customer information such as first name, last name, address, phone number, state, and credit card details.
* The system shall provide users with an appointment scheduling system that allows users to book, modify, and cancel appointments.
* The system shall offer customers different training packages based on the offerings established by the client.
* The system shall offer online classes and tests to customers.
* The system shall generate reports and provide tracking to Admin level users of the system.
* The system shall be updated via integration with local DMVs so that it may update to reflect any updates to DMV policies and procedures.
* The system shall be cloud-based with automatic backup and data recovery.

### 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 have a “homepage” or dashboard that provides and overview of system status and offers quick access to things that are used frequently like scheduling and user management. This dashboard should change to reflect the access level of the user. IE: admins should have access to user management and scheduling, customers should only have access to classes, scheduling, and updating their own information.
* The users of this interface are Admin, Employee, and Customer. Each user will have access to different aspects of the interface based on their role. When a user doesn’t have access to certain aspects of the interface, those aspects should be hidden from view. Admins will need to be able to manage users and make changes to the system, as well as have full access to every other aspect of the system. Admins should also be able to generate reports and access a “compliance” interface that allows them to check for updates from the DMV. Employees should be able to update customer information and schedule/register customers for appointments and provide customer information. Customers should be able to book their own appointments, update their information, access their online courses and be able to track their progress. Customers should also be able to access a “contact us” page where they can contact the client.
* Users should be able to interact with this interface via web browser.

### 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’m assuming that users will have internet connectivity, as users will have to have internet access in order to access the system.
* Not specifically addressed, but data security is to be assumed as the system will handle sensitive information.
* Basic tech-savvy – it is assumed that users are comfortable navigating web pages and moving comfortably through different parts of a website. For example, if a user forgets their password, we’re assuming that they’ll think to look for a password reset option.

### 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?*

* Offline Functionality
  + The Client stated that they wanted to be able to download and access the information in the system offline, but this can create data redundancy. They might be able to view information when it is downloaded at a certain point in time, but they won’t be able to make any changes to the system until they are online again.
* Customization
  + Any updates after we create the system will require a developer to come in and make changes. This limits how customized the system will be, as they won’t be able to easily make any changes once we’re finished with our final iteration of the product.
* DMV Integration
  + This might be one of the more difficult parts of building this system, as we’re assuming that we will be able to establish some sort of online connection to DMV resources that will allow us to make quick updates when the DMV has an update.
* Usage
  + Users might face a small learning curve depending on how complex the system is designed. This requires the interfaces to be as simplistic as possible to ensure that users can learn the system quickly and feel comfortable when using the system.
* Cloud Dependency
  + The system has the potential to be inaccessible for extended periods of time should the cloud provider experience an outage. The system’s performance and security are also completely dependent on the cloud service provider.
* Budget
  + More reliable and secure cloud service providers will most likely be more expensive. The client may want to save money in certain areas, such as this, that will potentially be detrimental to the system as a whole, as we aren’t hosting the system on site, but relying on a cloud to host for us.

### 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.*



