# 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?*

* This project is to develop a system for the company DriverPass that allows them to connect with users so they may provide services.
* The client, DriverPass, aims for this system to assist users with passing their driving tests at the DMV by providing practice tests and training with driving instructors.
* The system should allow the company to keep records regarding the users and services.

### 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 client wants the system to allow customers to easily sign up for classes to aid them in passing their driving test.
* For the customer facing functions of the system, the company requires its users to be able to self-serve.
  + First, the user should be able to create an account and purchase a package entitling them to services with DriverPass.
    - This should be able to be done online by the user, through a call to the company, or in person.
    - At registration, the system should store the user’s personal information associated with the account which includes:
      * first and last name,
      * address,
      * phone number,
      * state they live in,
      * credit card number,
      * credit card expiration, and
      * credit card security code.
  + The user should be able to make, modify, and cancel appointments with a driving instructor or for in person lesson.
    - The appointments should be specified as two-hour increments and they should be able to specify a pick-up location for the appointment.
    - It should be noted to the user that the pick-up location will also be the drop-off location when the lesson has concluded.
  + The system should include an online class environment that provides up to date practice testing and is accessible online by users that have purchased the required package.
* For the users on the company’s side, the system should provide user account control for each type of employee.
  + Administrator level privileges would allow for complete control over the data stored in the system and the ability to disable sales to certain packages.
  + Non-admin users should have the ability to register new users, sell packages, and make, modify, and cancel appointments, but not alter user or system data.
* The reporting data should be accessible by admins in an on and offline capacity, however the offline capacity will be read-only.
  + Changes can only be made while online.
  + All data backups and security should be handled through the cloud.
* Finally, the system needs to obtain real time reporting data from the DMV indicating when testing questions, policies, or rules change.
  + There should be an alert sent to necessary individuals in order to update procedures.

### 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 it should be able to:
  + Add and store users and their info
    - Measurable goal: Modifiable database that contains all required info
  + Enable users to purchase packages and certain users to disable package sales
    - Measurable goal: Associate user in the database with the purchased package and its benefits
  + Enable users to make, modify, and cancel appointments
    - Measurable goal: UAC for customers
  + Enable employee users to have permissions associated with their role
    - Measurable goal: UAC for each type of employee
  + Scheduling system to ensure appointments do not overlap
    - Measurable goal: Database with drivers, cars, and appointment times. Build a schema that guarantees times and drivers do not have scheduling conflicts
  + Enable cloud storage and encryption for user data
    - Measurable goal: Secure the databases
  + Enable online access for databases and the ability for certain users to download and print reporting
    - Measurable goal: Ensure databases can be stored as a local copy that is overwritten when online access returns.
  + Sync with DMV to receive real time updates when changes to relevant policies, procedures, or testing information are made.
    - Measurable goal: Alert system that notifies certain users whenever changes are made.

## 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 program will be web based and integrated with an internal server to store user data and appointment information. The server will be cloud based and built with NoSQL for easy integration with the website. The site will provide and receive data from the server in real time. Other than scheduled maintenance, the server will be accessible at all times. Customer information such as appointment times, personal data, and account security details, will be updated automatically when changed. User data should be reviewed quarterly to purge inactive accounts. Updates to the site will be pushed on a quarterly schedule, unless there is an imminent failure in the system that needs to be corrected.

#### 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 run on all current internet browsers. Specifically, the system will be optimized for use on Chrome, Firefox, Edge, and Safari. The operating system should not be a prohibitive factor for the system and should run regardless of the OS being used. It should also function on smartphones in a similar manner to a computer. On the backend, a MongoDB database will be used to store customer data and other business critical data. The server will be cloud based, likely utilizing AWS. Additionally, connectivity and back end security will be handled by CloudFlare.

#### 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 in the database by a unique ID that they do not create. It will be randomly assigned through the database when their account is created. Users will identify their account by a unique username and password that is associated with their email address, name, address, and their phone number. Passwords will be case-sensitive, but usernames will not be. The system should reject any attempt to create a duplicate username. Should there be a system issue, an alert should be immediately sent to the IT officer to determine the severity and response time.

#### 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 be built to allow updates to all parts of the user data without changing any code. The only exception will be the unique ID provided by the database, this will be immutable. Users should be able to be added or removed without any structural changes to the code. Any platform updates should not be utilized until the integrity of the system is confirmed in a testing environment. The IT admin should have access to view the entire system to ensure any issues can be resolved. They should be able to change permissions, reset passwords, and update information. They should not have access to edit the source code for the application unless it is determined the admin will handle updates rather than the developer. In addition, the owner should have access to the entire system but should not make any software changes.

#### 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 a user will need to input their username and password. They will also need to verify they are not a bot using a system such as reCAPTCHA. To protect the integrity of the user accounts, usernames and passwords will be encrypted and use a hash such as SHA-256. To protect data during transmission, the site will utilize Secure Sockets Layer certificates and they will be kept current by IT. If there is a brute force attempt on a user account, the account should lock after 5 attempts and require a password reset in order to unlock. The password reset will require the user to receive a communication at an external location, such as their email address or phone number. This will send a secure code and route them to the password change page for the site. If a user forgets their password, they can follow the same password reset option.

### 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 allow users to create an account secured by a password that is encrypted.
  + The system shall store the user’s first and last name, username, email, password, address, state, phone number, credit card number, expiration date, and security code.
* The system shall allow users to purchase packages through the site or by telephone.
* The system shall allow users to schedule appointments.
  + It will allow the users to create new appoints, modify existing appointments, or cancel appointments.
* The system shall maintain user account control and allows specific actions based on the user’s level of access.
* The system shall maintain a database containing scheduled appointments and confirm appointments do not overlap.
  + The system shall remove appointment slots as they are reserved.
* The system shall allow users to take practice tests through the website.
  + The system shall present relevant data related to the tests including the current test name, test enrollment status, score, and the length of time taken on the test.
* The system shall be able to print reports of the data relating to each class reservation or other user data.
* The system shall ensure driver availability for reservation times.
* The system shall allow present a landing page with the user’s test progress, personal information, notes from the driver, the photos of the student and the driver, and any special needs or notes regarding the student.
* The system shall maintain real-time updates from the DMV regarding any relevant changes to driving test regulations.
* The system should allow the user to schedule a pick-up and drop off location for their driving reservation. These shall be the same location.
* The system shall process payments for services.
* The system shall provide drivers the ability to review customer pick up and drop off location data, review reservations, and leave customer notes.

### 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 needs to be organized and easy to maneuver. The users for the interface will be customers, IT, the secretary, the drivers, and the owner. Customers will need to be able to sign up for packages, schedule tests, take practice tests and review related data, reserve driving time, set their personal information, and schedule and manage their driving reservations. The secretary will use the system to schedule and manage tests, process payments, and update customer payment information. IT will manage the system and resolve any technical issues. The drivers will need to access their schedule, customer notes, and the customer information regarding the reservation. The owner will access every aspect of the system and be able to aggregate and print reports of system data. All users will be able to access the system through a website, both on mobile and computer browsers.

### 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 above information does not address individuals who do not have internet connectivity. This document assumes the user will utilize the features through the internet. There will be the option to schedule classes by phone, but the majority of the functionality will not be available to a user who cannot access the internet.

### 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 limitations to the project would be access to resources and potentially a budget issue. The system does not have a dedicated software developer so any changes to the code would require additional development time. This would involve hiring another developer, or the one who builds the initial system, and spending more time and money adding features.

### 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.*

