# 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?*

Customer DriverPass company wants us to make an application for driving exam preparation. The aim here is for people to easily prepare for the exam online and offline.

To enable them to attend online classes and receive driving training.

If we detail this system, it is a system where users can prepare for the driving license exam with the basic lines that the client wants.

We can define the entire system that the user wants as follows:

* Preparation documents
* Preparation Tests
* Online courses
* Driving lessons
* Preparation lesson

In short, the system should present these contents and features to users as the customer wants.

### 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 thinks that many people cannot prepare for the driving exams properly and wants to offer a flexible solution for this. It creates a solution to this problem by creating a web-based training platform and providing the necessary training, documents and information to prepare for this exam within certain packages. When the users of the system are examined

It is predicted that there will be three types of users in the system

* Standard user
* Administrators
* Drivers

**Standard user:** A person who is a member of the system and will benefit from the content and courses according to the package.

**Administrators**: Persons who will carry out operations such as creating packages, editing, deactivating, activating and deleting, operations related to standard users and drivers, exam and content management, regulatory operations, etc. (here, the rights of each administrator can be restricted with the role system)

**Drivers:** people who will give driving lessons and evaluate the person taking the lessons.

What is missing in the interview is that separate components should be created for these three users and they should perform their operations from their own parts, so the customer should take these components into consideration.

### 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?*

In this context, I will examine the success of the system in two parts: from a project management perspective and specifically for the project itself. On a project management level, the success criteria selected will be quality-scope, and time, meaning the project will be completed within the specified time and with the required quality, thus a corresponding budget will be established. From a project management perspective, an application that successfully delivers on time and meets all specified features will be considered successful.

On a project-specific level, success will be determined by achieving the defined objectives. These objectives will be set as functional components of the project:

1. Defining the features of the administrator, standard user, and driver systems.
2. Designing the interface.
3. Developing the administrator system:
   * + - Developing the role system for administrators.
       - Creating a content management system: creating, adding, updating, and editing content, and associating this content with DMV.
       - Completing package operations module: creating, adding, updating, and editing packages, and associating them with content.
       - Developing the user management system: manually updating, adding, deleting user information (password, name, address, subscribed packages, etc.), user activation and deactivation, etc.
       - Financial systems: package pricing, sales, payment processing, and related transactions.
4. User operations: providing access to content based on membership type, package purchase, renewal, cancellation, payment processing, progress monitoring, reservation operations, etc.
5. Developing the driver section:
   * Setting schedules, approving or canceling incoming reservation requests, rescheduling, entering and updating student evaluations and grades, contacting students, etc.

These components will be defined as criteria, and when successfully completed, the project can be considered successful.

## 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?*

* We can examine the system in two ways: as a web application and as a mobile application that communicates with a web service for data exchange. The mobile application will have the capability to download information and operate offline. However, in both cases, the system's performance, including internet speed and responsiveness, should be smooth and satisfying for the user. Different parts of the system need to be updated at different times. For example, changes related to exams should be updated when the local DMV makes changes, while processes such as course registration and payment should be updated in real-time. Therefore, the update plans for different parts of the system should be managed separately.

#### 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 cross-platform. For the web pages and web services, there will be no restrictions on the user side, as they will operate over the HTTP protocol and can be interpreted by any processor in the same way. However, mobile applications need to be platform-native, meaning separate development for Android and Apple. On the server side, programming languages and systems that work across platforms, such as .NET Core, Java, and PHP, should be used, along with open-source databases like MySQL and MariaDB that work on all platforms. This ensures platform independence on the server side and allows the customer to manage the server based on their own experience. Additionally, for automatic updates from the DMV, the network should have open access to the relevant servers, IP addresses, and ports.

#### 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?*

* Three types of users are defined in the system, differentiated by a role-based access control mechanism. Each user type will have access only to the parts related to their roles, utilizing the appropriate navigation systems to access these areas. The role system must be integrated throughout the entire system and carefully implemented, as a well-executed role system can successfully distinguish between different users. The alert mechanism can operate at different levels for different processes. For instance, errors that affect the system's operation should trigger immediate alerts, while processes that do not affect the system's operation but require data integrity adjustments can be addressed with an end-of-day report, allowing the administrator to make all necessary corrections at once. This way, the system administrator's time is managed and utilized efficiently.

#### 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?*

* We can perform all changes (Add/Delete/Edit) without repeatedly modifying the code because the system will store all information in the database, and the data in the database will be automatically updated from the relevant panel. Since the system is designed to process this data, all changes can be made easily without modifying the code. An administrator module will be developed, allowing the IT admin to access the management platform where they can view, edit, and delete this information. By accessing this module, the admin will be able to perform all types of operations.

#### 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?*

* Ensuring user security involves multiple steps, starting even before the user registers on the system. The system should be designed to compel users to choose passwords that are difficult to guess and complex enough to protect against brute force attacks. Secondly, after a certain number of failed login attempts, the user can be locked out for a specific period or until they reactivate their account via their email address or phone number. This approach can render brute force attacks ineffective. When storing user passwords in the database, they should be encrypted using hash algorithms that cannot be reversed (such as SHA-2 or MD5). This ensures that even if a data breach occurs, the user's login information remains uncompromised. Additionally, the data transmission between the client and the server should be encrypted using HTTPS protocol and TLS certificates to prevent Man-In-The-Middle (MITM) attacks from exposing user credentials. Finally, all session IDs generated by the system should be unique to each user, incorporating user-specific information when creating these IDs. This way, if a session ID is stolen, the mismatch of data from a different computer will prevent session hijacking.

### 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 should allow users to register online.
* The system should allow users to register via phone or office visits.
* The system should allow users to take online classes and practice tests.
* The system should allow users to make reservations for driving lessons.
* The system should record the date, time, and instructor information for each scheduled driving lesson.
* The system should track and report reservation, cancellation, and modification activities
* The system should provide access control with different user roles and rights.
* The system should match each customer with a specific vehicle and instructor for tracking purposes.
* The system should allow users to select package options and make reservations.
* The system should manage the active/inactive status of packages.
* The system should allow users to automatically reset their passwords.
* The system should receive and notify DMV updates.
* The system should allow users to download reports and work on them in programs like Excel.
* The system should collect users' pickup and drop-off location information.
* The system should allow users to modify package contents.
* The system should provide full account access to IT admins via an admin module.

### 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 ensure that users can access the system easily and effectively. Students should be able to register online, take courses and practice tests, make reservations, and manage those reservations. They should also be able to reset their passwords and receive DMV updates. Admins should have full access to all accounts to maintain the system, download and review reports, and manage user accounts. Instructors should be able to manage students' lessons and reservations. The interface should be accessible via mobile devices and web browsers, offering a user-friendly experience. It should be aesthetically pleasing, modern, and straightforward while meeting data display requirements.

### 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?*

In my design, I did not define whether users and administrators know how to use such systems, and it is not known how the backend resources should be in the production environment and whether the local DMV has a structure for automatic updates, so the following assumptions have been made accordingly

**Assumption of whether users and administrators have sufficient knowledge.**

While the system was being designed, it was prepared with the assumption that users and administrators have sufficient knowledge to use this system.

**Sufficient resource assumption**

When the system is taken to the production environment, it is assumed that the infrastructure is sufficient and at a level that will protect the density and functionality of the system.

**DMV data sharing assumption**

It is assumed that there is a system where DMV shares data for automatic updates and that the necessary data will be pulled from here and automatic updates will be made.

### 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?*

* In the meeting data, since there is no budget mentioned, we can naturally consider the budget as a limitation. Technology requirements do not impose a limitation during the development phase, but they might during the development stage. When the budget is sufficient, time cannot be considered a limitation because hiring expert personnel can eliminate time constraints. However, when the budget is insufficient, time can also be considered an additional limitation.

### 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 management

Description automatically generated*