# 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 is to develop a system for the driving training company DriverPass to improve their training methods. They are outlining a solution to provide a package system that includes features like bundled classes and on-the-road training. The company is seeking a system to streamline these things and provide centralized management for different customers and reservations. The system will include a reservation system, a registration module for personal information, and a compliance module to keep up with regulations. User access and management roles are also essential to control who has access to different parts of the system. The company also wants to have a cloud-based infrastructure to simplify the architecture and have a reliable and secure hosting service.

### 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 their system to address the inefficiencies in traditional training methods. They want to manage reservations and data seamlessly. The company would like to improve the lengthy process that is scheduling with drivers’ training and collecting data. This can be done by designing an application that is user-friendly, includes access rights, stores customer information, and includes an easy-to-use system. The systems’ network infrastructure needs to be simplified and built to utilize different cloud-based hosting services for reliability, security, and scalability.

### 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 system designed should be able to meet DriverPass’s needs. Customers should be able to seamlessly schedule lessons with the ability to choose from several package bundles. The system should also be able to securely store the customers’ information, like their payment method, safely and securely. Real-time notifications from the system will ensure the company stays up to date with different regulatory and compliance requirements. The user interface will be easy to use and to navigate. These new features will simplify the company’s process and operations and improve their bottom line.

## 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 application should have fast and responsive performance while executing different tasks within the system. Response times are critical to ensuring the user has a seamless experience while using the application. Frequent updates are mandatory, as the technologies we will be utilizing are consistently updated, addressing vulnerabilities and security threats. These updates will also extend to the addition of extra features and functionalities that will be implemented into the existing system.

#### 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 infrastructure of our system is going to have to ensure compatibility across other platforms. Multiple devices are going to be utilized when accessing our application, such as mobile phones and Windows computers. By making sure our system is accessible we can maximize the coverage and user reach of our application and ensure that our application can be accessed from anywhere. These requirements have led to the decision to utilize a cloud-based hosting service. By using this service, we can leverage the clouds scalability and cost efficiency. Our architecture and codebase will be fine-tuned to implement the cloud-based service for hosting the application.

#### 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?*

* When handling and authenticating data between different users and credentials, user authentication is required. Data in our system is sensitive, so certain requirements must be met. When implementing the logic of the application, data authentication methods like input validation and error-handling must be addressed. This also includes encrypting data with algorithms as well as implementing certain access controls within the system. This will help control the data that will flow throughout the system as well as authenticate the user that is executing these tasks.

#### 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 a process implemented that will allow for different users to have the ability to give and remove permissions of certain users, limiting what they can or can’t do inside the system. This includes administrative roles, giving the ability to IT professionals to configure different settings and analyze different functionalities within the system. The system will update, as all operating platforms and systems regularly do, so the system needs to be designed with this in mind. This means the structure and logic of our code needs to be able to support these different access control functionalities.

#### 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?*

* When it comes to security, one of the most important features to implement is a strong password requirement. This includes a set number of letters, numbers, and special characters. The system will lock you out after a set number of attempts, preventing forced access through excessive login attempts. Standard procedure would require the user to link to an active email account, for recovery and authentication purposes. As I mentioned, data encryption is critical and its implementation in our system logic is mandatory. This layer of security can be implemented with the utilizations of different algorithms that can help with the encryption of data.

### 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 the users to create an account to save their information securely.  
  -The system shall allow users to create and cancel reservations conveniently online.  
  -The system shall have role-based functionalities to ensure the appropriate tasks can be done by the appropriate user.  
  -The system shall conduct regular updates to ensure it is up to date with official requirements and DMV regulations.  
  -The system shall track all user activities to be used for reporting and system analysis purposes.

### 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.)?*

* All interfaces that will be utilized by customers need to be as user-friendly as possible. This provides a smooth and seamless experience, avoiding issues with users navigating our application. It must be smooth, responsive, and consistent. Modern interfaces have interactive elements as well, like generated metrics for your upcoming schedule dates and different timeframes to be accounted for. This helps create an experience that is perfectly tailored to the needs of the user.

### 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?*

* Our assumptions are logic-based through professional analysis. The cloud-based infrastructure will scale according to our operations and usage, as that’s one of the primary benefits of using the service. We also assume that our users can use a mobile phone as well as have a connection to the internet. We can also assume that federal and state regulations will be updated regularly with our DMV compliance standards and requirements.

### 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?*

* When it comes to project constraints, there are a few that stand out as critical, as well as found mutually among many other software projects. One constraint is time, as all projects have different deadlines. Resources can also serve as a limitation or constraint, as every team is limited to a certain number developers and other specialists. The technology stack you are working with directly influences the limitations of the system, as you must research and understand what different technologies can be utilized together and that are compatible.

### 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.*

The design phase should start right after requirements gathering is almost done, making initial sketches almost right away. Feedback phase generally starts as the design phase is over. The interface kicks off right after the approval and starts during feedback. Database is setup before the interface is done and is started along with the backend development so they can work in parallel. Testing should overlap with backend work. The goal is to start as early as possible on every task. Written documentation and graphical charts help represent and visualize these different phases of development. B  
  
  
  
  
  
-Requirements Gathering: 10 days  
-Design: 10 days  
-Feedback: 10 days  
-Interface: 14 days  
-Database: 5 days  
-Backend: 14 days  
-Testing: 10 days  
-Delivery: 3 days  
-Closure: 1 day