# 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 design and build an online and on-the-road training system for DriverPass, a company aiming to help individuals pass their driving tests at the DMV. The client, DriverPass, is represented by Liam, the company owner, and Ian, the IT officer. They want a system that allows customers to take online driving classes, practice tests, and schedule in-person driving lessons. The system should enable users to make, cancel, and modify reservations for driving lessons, with options for different training packages.
* Additionally, the system needs to support data access from various devices, ensure data security and tracking, manage multiple user roles, and enable integration with DMV updates to keep training material current. Liam also wants a user-friendly interface that includes test progress tracking, lesson scheduling, and driver notes. The project should be flexible to allow future adjustments in training packages and minimize technical overhead by operating on a secure, cloud-based platform.

### 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 a system to improve how people train for their DMV driving tests, tackling the problem of high failure rates due to lack of quality preparation. The idea is to offer a mix of online classes, practice tests, and in-person driving lessons, so customers feel ready for the test. By centralizing these services, DriverPass aims to create a one-stop solution for driving test prep.
* The system will have some key components. First, it needs secure role-based access for different users, like the admin, IT officer, secretary, and customers, ensuring that data stays safe, and accounts can be managed easily. The system will also offer an online learning platform, allowing customers to study, take tests, and track their progress so they know where they stand.
* Reservations and scheduling are also essential. Customers need to book, modify, or cancel lessons online or by calling in, with options for different training packages. The system will log these actions for accountability and allow access from any device. To keep training content up to date, the system should connect to DMV updates, notifying DriverPass of any new rules or policies.
* Finally, the interface will be web-based and user-friendly, with simple forms for student info, lesson details, and driver notes. Hosting on the cloud will handle security and backups, so DriverPass can focus on training without technical hassle. This setup will let DriverPass deliver effective training, keep data secure, and stay aligned with DMV standards.

### 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 this system is completed, it should provide an all-in-one solution for DriverPass to train students successfully for their DMV driving tests. It should allow customers to take online courses, access practice tests, and book in-person driving lessons, all from their computer or mobile device. The system also needs to be secure and accessible, enabling different users to manage accounts and reservations seamlessly. Tracking actions like bookings, cancellations, and changes should be easy so that DriverPass can keep a clear record of who did what.
* To make this happen, the system design needs to focus on some measurable tasks. First, it should include role-based access control so each user only has access to what they need, which will keep data safe and organized. The online learning platform must be able to track customer progress, displaying test statuses such as "not taken," "in progress," "failed," or "passed," so customers and staff can monitor readiness levels. Reservations and scheduling features should support real-time updates, allowing customers to book, cancel, or reschedule lessons, with each action logged for accountability.
* Another important task is integrating with DMV systems to receive updates on rules, policies, or sample questions, ensuring training content is always current. The system should be cloud-based, with secure data storage and automatic backups, so DriverPass won’t need to worry about technical maintenance. Finally, the interface should be designed to be user-friendly and web-based, featuring simple forms for customer info, scheduling details, and driver notes.

## 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 must run as a web-based platform hosted on the cloud. This ensures accessibility of any device with an internet connection, including mobile devices and desktop computers.
* The system should provide fast response times, with page loads and data retrieval occurring within 2 seconds for most operations.
* Updates to the system should happen during non-peak hours to minimize user disruption, with regular updates monthly for feature enhancements or as needed for critical patches or bug fixes.

#### 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 not be platform dependent and be accessible through all modern web browsers on operating systems such as Windows, macOS, Linux/Unix, iOS, and Android.
* The backend has several requirements such as relational database management system (RDBMS), for example MySQL, to manage user data, reservations, and activity logs.
* A cloud platform, for instance AWS, Azure, or Google Cloud, to ensure scalability, high availability, and secure data storage without much maintenance.
* Tools required include a web application framework like Spring Boot or Django for implementing business logic and integrating the user interface with the database.

#### 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?*

* Implementing a Role-based access control (RBAC) will differentiate users, e.g., admin, IT officer, secretary, customer, and define their permissions.
* We will follow the industry standard by keeping the username/emails case insensitive to avoid login issues and passwords case sensitive to improve security and ensure exact matching of characters.
* The system should inform the admin in the case of cloud service outages, server downtime, database connection issues, duplicate/corrupted data, repeated failed login attempts and a large surge in usage, which could indicate a DoS attack and other anomalies.

#### 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 driver pass system should allow IT admins to add, remove, or modify users through an admin dashboard linked to a relational database. This ensures updates can be made without altering the code.
* The system should use containerization for consistent operation across updates and implement automated testing to identify compatibility issues. Dependency management tools should keep third-party libraries up to date.
* The IT admin requires user management access, system logs, configuration controls and backup tools to keep the system secure and easy to maintain.

#### 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?*

* Users are required to log in using a unique username and a strong password that meets complexity standards. Additionally, they must perform multi-factor authentication, such as entering a code sent to their email.
* All data exchanges between the client and server should use HTTPS with TLS encryption. Sensitive data like passwords should be hashed, bcrypt, before storage in the database.
* In the event of a “brute force” attack, it is essential to lock the account after a certain number of failed login attempts. Implementing CAPTCHA after multiple unsuccessful attempts can improve security. In addition, notifying both the user and the administrator about any suspicious activity can help prevent and deter such attacks.

### 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 administrators to add, remove, or modify user accounts without requiring code changes.
* The system shall track, and log login attempts and lock accounts after a defined number of failed attempts.
* The system shall provide functionality for users to reset their password securely via email.
* The system shall support role-based access control, ensuring users and admins can only perform authorized actions.
* The system shall notify administrators of suspicious activity, such as multiple failed login attempts or potential data breaches.
* The system shall enable secure communication between the client and server using HTTPS with TLS encryption.
* The system shall generate, and display driver pass status and test results in real time for users and admins.
* The system shall provide a dashboard for administrators to monitor system activity and user performance.
* The system shall allow users to update personal information, such as email and contact details, while ensuring data validation.
* The system shall provide online access to data from any computer or mobile device with an internet connection.
* The system shall provide a user interface accessible through both mobile devices and web browsers to cater to different user needs.

### 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 shall display the user's online test progress on the left side and personal information on the right side. The Driver Pass logo should be located at the top. Special needs information should be shown below the personal information in a separate section. Driver and student photos should be displayed at the bottom, each in their own square. The driver's notes should be positioned in the bottom-left corner, below the online test progress section.
* The needs of the interface are clarity and simplicity, meaning the interface must be user-friendly and easy to navigate for drivers and admins. Interface must show up-to-date information such as online test progress, personal details, and special needs. Ability to modify user details and test information, depending on the user’s role. The interface must be accessible on multiple platforms, including mobile devices and browsers. The interface must ensure secure logins and data protection while allowing users to interact with the system.
* The customers will need to book and manage their driving lessons, track progress in online courses, create and manage accounts.
* The drivers will need to view assigned lessons and times. Leave comments on lessons and access the student’s information and driving history.
* Admins will need full access to reset passwords, modify user roles, and block access for former employees. Ability to monitor system activity and ensure smooth operation. Admins will also need the ability to disable or modify packages.
* Mobile access should allow users to interact with core functionalities like booking lessons, checking progress, and receiving notifications about updates.

### 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?*

* Users are assumed to have internet access for syncing data, though offline functionality is available with limited features.
* Users will have access to modern devices, smartphones, tablets, or computers, with browser or app compatibility.
* Users, including admins and customers, are expected to have basic knowledge of mobile and desktop applications.
* The system is designed to be scalable, but future features and growth are not fully identified.

### 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?*

* Cost constraints may limit the inclusion of advanced features such as AI-based scheduling or reporting tools.
* Real-time integration with DMV systems may face compatibility or latency challenges.
* Users may face a learning curve, especially for features like schedule management or driver notes.
* Limited ability for non-developers to modify or create new training packages without developer assistance.
* Only basic data can be accessed offline; updates and modifications require an internet connection.

### 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.*

