# 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?*

* DriverPass is a company that wants to offer online courses for drivers, practice driving tests, and on-the-road training if the customer wants it. This is provided through different packages on their website. They want their course to increase the rate at which newer drivers pass their tests at the DMV.

### 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 help new drivers. They see many new drivers fail their tests at the DMV and thought of a solution for this, having practice tests online, one-on-one training, and online courses. The different components needed for this system are the cloud to operate the website, administrators to uphold to new DMV standards, and instructors for on-the-road training.

### 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?*

* This system should be able to have interactive courses, practice tests with feedback, a tab that shows previously done courses/tests with a score and whether you passed or failed them, different packages the user chooses from, instructor feedback and a tracker for hours driven, and schedule appointments with instructors at a specific place and time.

## 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 run in both web-based and application environments to ensure accessibility online and offline.
* The server response time should not exceed 2 seconds.
* Once connected, page contents should load within 3 seconds, assuming the user has a stable internet connection.
* The application should be able to function offline, allowing users to access previously downloaded content and complete practice tests without an internet connection.
* The system should be capable of handling up to 10,000 concurrent users, considering the average user base for online learning platforms.
* Regular updates should be performed monthly to ensure the system meets the latest standards and requirements.

#### 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 web-based component should be accessible via any modern web browser, ensuring compatibility across all operating systems including Windows, macOS, Linux, iOS, and Android.
* The offline application should be available for download on Windows, macOS, iOS, and Android platforms.
* The back end requires a database such as SQL to store user information, appointments, and training records.
* The system should utilize cloud services for data storage and backup to ensure reliability and security.

#### 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 by unique usernames (not case sensitive) and passwords.
* Passwords should be case-sensitive to ensure security.
* The system should inform the admin of any suspected brute-force attacks or unusual login activity.
* User roles (Administrator, User, Secretary) will be clearly defined to ensure appropriate access and functionality.

#### 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 allow user management (add/remove/modify) through an admin interface without requiring code changes.
* The system should be designed to adapt to platform updates with minimal downtime and disruption. This could be achieved through methods such as continuous deployment.
* The IT admin needs full access to user management, system settings, data backups, and the ability to monitor system performance and security.

#### 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 need a username and password to log in.
* Secure the connection using SSL/TLS encryption to protect data exchange between the client and the server.
* Lock the account after multiple failed login attempts indicative of a brute-force attack and notify the user and admin.
* Provide a password recovery option via email verification to help users reset their passwords if forgotten.

### 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 validate user credentials when logging in.
* The system shall allow users to register and manage their personal information.
* The system shall provide different training packages for users to choose from.
* The system shall allow users to schedule, modify, and cancel appointments.
* The system shall track on-the-road training sessions and provide feedback.
* The system shall offer online practice tests and courses.
* The system shall allow administrators to manage the system and user data.
* The system shall ensure all functionalities comply with local authorities and DMV regulations.

### 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 be user-friendly, intuitive, and accessible.
* Different users: Administrator, User, Secretary.
* Administrators need to manage the system and user data, including adding/removing/modifying users and monitoring system performance.
* Users need to register, schedule appointments, access courses, and track their progress.
* Secretaries need to manage appointments, assist users, and handle customer inquiries.
* The interface should be accessible via web browsers on both desktop and mobile devices, as well as through dedicated applications for Windows, macOS, iOS, and Android.

### 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?*

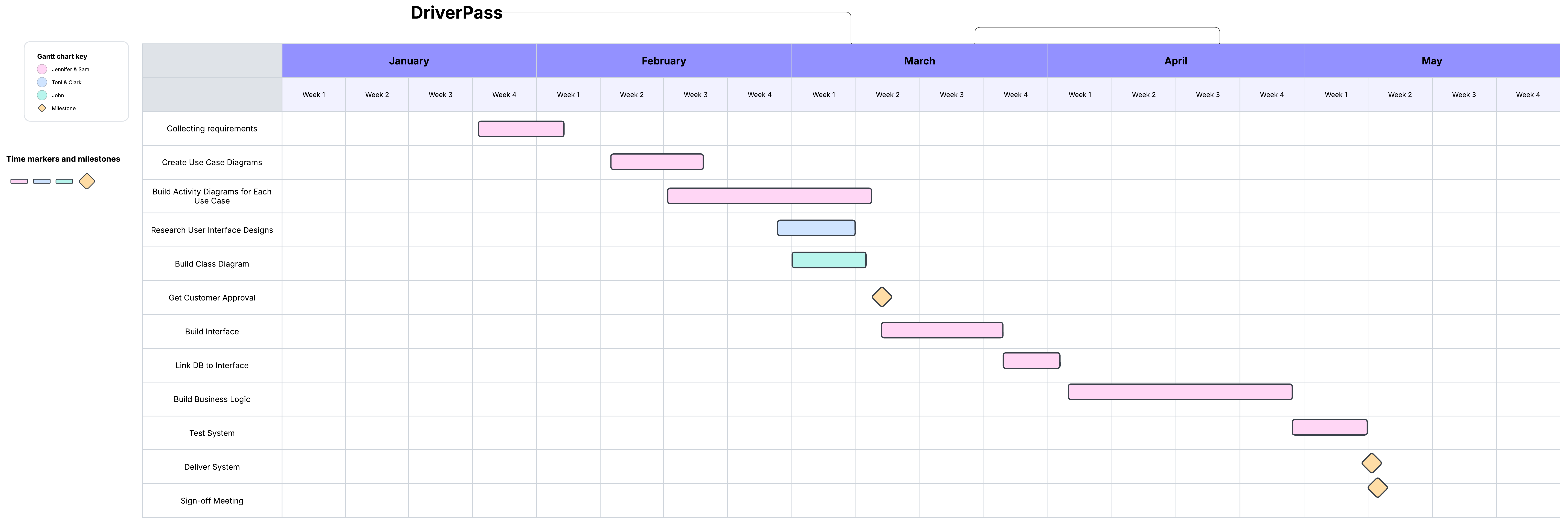
* Assumed users have basic internet access and a web browser.
* Assumed users have basic computer literacy.
* Assumed the system will be hosted on a reliable cloud service.
* Assumed users will have access to devices compatible with the offline application (Windows, macOS, iOS, Android).

### 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?*

* Limited by the budget for cloud services and development.
* Time constraints for developing and testing the system.
* Potential limitations in user internet connectivity and device compatibility.
* Limited resources for ongoing maintenance and support.
* The system may face scalability challenges as the user base grows.
* Changing the system or products offered, such as adding packages, may be complex and resource-intensive.

### Gantt Chart



The text isn’t exactly big, so here is a summary. Pink is Jennifer & Sam, blue is Toni & Clark, and green is John. The milestones are labeled with yellow, and the chart might not be exactly accurate but should be a pretty good representation of the tasks.