# CS 255 Business Requirements Document Template

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 goal of this project is to create a comprehensive and reliable system for DriverPass that allows students to access online practice exams, register for in-person driving lessons, and track their training progress. The client, DriverPass, wants to reduce failure rates for the DMV exam by offering both digital and physical support through a centralized platform. The system should work across desktop and mobile devices and be flexible for future updates.

### 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 has identified a lack of effective tools for students preparing for driving exams. Many students are failing because they rely solely on memorizing previous tests. DriverPass wants to create a system that combines practice tests and real-world driving lessons to better prepare students. The platform must be scalable, allow for various user roles like students and staff, manage schedules, store exam history, and connect to DMV rule updates. Key components include a student portal, appointment scheduler, package selection, instructor and vehicle assignment, and admin tools.

### 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 should let students create accounts, schedule driving lessons, take practice tests, and monitor their progress. Staff should be able to manage student accounts, lesson schedules, and generate performance reports. Admins need tools to oversee packages, reset passwords, and view user data securely. Each feature must be accessible on both desktop and mobile devices, with performance tracking and report generation included.

## 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 platform must be web-based and support high uptime availability. Response times should be under two seconds for most actions. The system should support frequent updates and allow quick deployment of new DMV rules and content.

#### 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 needs to be compatible with major operating systems like Windows and macOS. The back end will require a secure, cloud-based database such as PostgreSQL or MySQL to manage data efficiently. It should be scalable and work on both web browsers and mobile operating systems.

#### 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 identified via role-based login credentials. Input fields like username and password should be case-sensitive. Admins should be notified automatically if errors occur during scheduling, account creation, or test result processing.

#### 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 must allow IT staff to add, remove, or update users without modifying core code. It should be easy to push updates, and the platform must remain compatible with future device and software versions. IT admins need full access to manage data, users, and troubleshoot system performance.

#### 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?*

* User login should require a username and a secure password. Passwords must be encrypted in storage. Data transmission should be protected with HTTPS and possibly two-factor authentication. Accounts should be locked temporarily after multiple failed login attempts. The system must include password recovery options such as email verification or security questions.

### 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 register and create accounts.  
 The system shall validate user credentials during login.  
 The system shall allow students to take online practice tests.  
 The system shall track student test history including scores, duration, and status.  
 The system shall enable students to schedule, cancel, and reschedule driving lessons.  
 The system shall allow staff to manage appointments for walk-in or phone-in students.  
 The system shall assign instructors and vehicles to appointments.  
 The system shall log appointment changes for reporting.  
 The system shall generate downloadable reports in formats like CSV or Excel.  
 The system shall allow admins to activate or deactivate training packages.  
 The system shall restrict access to system components based on user roles.

### 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 system must be easy to use, clean, and accessible from both web browsers and mobile apps. Students will need to log in, take practice exams, view scores, and manage appointments. Staff and admins must have interfaces for scheduling, managing students, and viewing reports. IT staff will require access to user controls and system logs. The interface must include forms for contact and support and support visual cues for status 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?*

* We assume all users will have reliable internet access and a modern browser or smartphone. It is assumed that staff will be trained to use the system and follow security protocols. We also assume that students have basic digital literacy to navigate the system.

### 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?*

* Offline use is not supported due to the system's need for real-time data syncing. Customization of features outside the admin panel will likely require developer support. Budget constraints may limit the range of features in the initial version. Full DMV integration will depend on available APIs or manual content updates.

### Gantt Chart

