# 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 the DriverPass system is to provide students with access to online driving practice exams, DMV rule tutorials, and on-the-road driver training appointments.
* The client is DriverPass, represented by its owner Liam and IT officer Ian.
* The system will enable online class access, lesson reservations, appointment management, data tracking, and report generation.

### 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 the system to provide comprehensive driver training by allowing students to take online practice exams, attend virtual and in-person lessons, and schedule driving sessions. The problem they aim to fix is the high failure rate of students on DMV exams due to inadequate preparation tools. To address this, the system needs several components: a web-based interface, a reservation and scheduling system, online test and course management, role-based user access, administrative tools for tracking and reporting, and integration with DMV updates.

### 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 allow:
  + Customers to register, schedule lessons, take practice exams.
  + Admins to track all changes and generate activity reports.
  + IT staff to manage user roles and reset passwords.
  + Secretaries to book lessons on behalf of customers.
* Key goals:
  + Flexible package management (disable/enable packages).
  + Secure user login and access control.
  + Real-time notification system for DMV updates.
  + Mobile and desktop accessibility.

## 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 be web-based and accessible via desktop and mobile.
* It must support real-time updates for schedules and test progress.
* The system should be highly available with 99.9% uptime.
* Daily backups and periodic maintenance are required.

#### 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 run on a cloud-based platform that supports modern web browsers across Windows, macOS, and mobile operating systems. The back end will require a relational database, such as MySQL or PostgreSQL, to manage user data, reservations, test results, and system logs efficiently.

#### 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?*

* Different users will be distinguished by their unique login credentials and assigned roles (e.g., admin, IT, secretary, student). The system will enforce role-based access control to ensure each user only accesses functions relevant to their role. Input such as usernames will not be case-sensitive to improve usability, but passwords will be case-sensitive for security. The system should automatically inform the admin of problems such as failed login attempts, scheduling conflicts, unauthorized access attempts, or failed data updates.

#### 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?*

* Yes, the system will include an admin dashboard that allows authorized personnel (like the IT admin) to add, remove, or modify users without changing the code. It will be designed with configuration-based user management to support this flexibility. The system will be built using modern frameworks and APIs to ensure compatibility with platform updates, and cloud infrastructure will manage patches and security updates automatically. The IT admin will need full access, including user management, account resets, permission control, and system monitoring tools.

#### 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 will be required to log in using a valid email and password combination. To secure the connection and data exchange, the system will use HTTPS with SSL/TLS encryption, and all sensitive data (e.g., passwords, credit card details) will be securely hashed or encrypted in storage. If a brute force attack is detected such as multiple failed login attempts, the system will temporarily lock the account and notify the admin. For password recovery, users will be able to reset their password through a secure email-based token link.

### 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 customers to register, log in, and manage their profiles.
* The system shall allow customers to schedule, cancel, and modify driving lesson appointments.
* The system shall allow secretaries to book appointments on behalf of customers.
* The system shall track and display driving session history, including driver notes and lesson times.
* The system shall offer different training packages and manage availability based on active or disabled status.
* The system shall provide access to online course materials and practice tests for eligible customers.
* The system shall track user progress on online tests and display status (not taken, in progress, failed, or passed).
* The system shall generate reports detailing user activity, including who created, canceled, or modified appointments.
* The system shall allow IT admins to reset passwords and manage user access rights.
* The system shall notify staff of DMV updates, such as changes in rules, policies, or test questions.
* The system shall securely process and store customer payment and contact information.
* The system shall allow customers to reset forgotten passwords via secure token-based email verification.

### 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.)?*

**Users & Needs:**

* **Students:** Register, manage profiles, schedule lessons, take tests, view progress.
* **Secretaries:** Create/update student profiles, manage appointments.
* **IT/Admins:** Manage users, reset passwords, control access, generate reports.
* **Owner:** Oversee activity, manage packages, download reports.

### 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?*

Some specifics like payment gateway integration details, exact DMV update mechanism, and mobile app support were not fully addressed.

Assumptions made include:

* Users have access to reliable internet and modern web browsers.
* The DMV provides updates via an accessible API or feed.
* Customers are comfortable using online forms and email-based password recovery.
* Payment processing will be handled through a third-party service (e.g., Stripe or PayPal).

### 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?*

* **Customization:** Training packages can't be fully created or modified by non-technical users; changes may require developer support.
* **Offline Functionality:** Limited offline capabilities—most features require an internet connection.
* **Scalability:** Initial deployment may be optimized for small to mid-sized user volume; scaling may require future infrastructure upgrades.
* **Resource Constraints:** Time and budget may limit advanced features like AI-driven test recommendations or native mobile apps.
* **Dependency:** Relies on timely DMV updates and third-party services (e.g., for payments and cloud hosting).

### 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.*

**