# 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 create a comprehensive system for **DriverPass**, a company that provides driver training and online resources for individuals preparing for their driving test.
* **DriverPass** (owned by Liam, with Ian as the IT officer) wants their system to:
  + Offer online classes, practice tests, and on-the-road driving lessons.
  + Allow customers to make, modify, and cancel driving lesson reservations.
  + Enable administrators and employees (secretary, IT officer, etc.) to manage user accounts, monitor reservations, and track driving lesson schedules.

### 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?*

* **What** **DriverPass Wants the System to Do:**
  + Provide better driver training resources to reduce DMV test failures.
  + Include an online portal that users can access from anywhere.
  + Allow for reservations for driving lessons, each of which is two hours.
  + Support multiple packages (ex: 6-hour, 8-hour, 12-hour training plans).
  + Track reservations, modifications, and cancellations to identify who made changes.
  + Integrate with DMV updates for policy/rule changes and practice test questions.
* **Problem to Fix:**
  + Existing driver training options are inadequate or fragmented; DriverPass wants a centralized, flexible system.
  + Need a secure system that can handle multiple user roles and track all lesson bookings without confusion.
  + Must allow offline download of some data, with the understanding that **modifications** only happen when online to avoid data redundancy.
* **Components Needed:**
  + **Web Based Application** for customers, secretary, and admin.
  + **Database** to store user information, reservations, packages, and driving test data.
  + **Security and Access Controls** to differentiate between user roles.
  + **Connection to DMV** for receiving updates on test rules, policies, and sample questions.
  + **Reporting/Tracking Mechanism** to identify who made or changed a reservation.

### 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?*

* **Overall Objectives:**
  + Provide a way for customers to sign up for lessons, packages, and practice tests via an online portal.
  + Ensure administrators can manage user accounts, reset passwords, and track data changes.
  + Keep accurate records of lesson reservations, packages purchased, payments, and user activities.
  + Provide an interface for both customers and employees to operate with minimal technical barriers.
* **Measurable Tasks/Goals:**
  + **User Account Management**: The system shall allow new user registration, login validation, password resets, and role-based access.
  + **Reservation System**: The system shall enable scheduling, modifying, and canceling lessons with automatic tracking of changes.
  + **Package Management**: The system shall allow for enabling/disabling training packages without requiring direct code changes.
  + **Reporting**: The system shall generate activity reports that identify which user performed each action, when, and for which record.
  + **DMV Integration**: The system shall have a mechanism to receive and apply updates for rules, policies, or practice questions.
  + **Online Class/Practice Tests**: The system shall show test progress, time taken, score, and status (ex: not taken, in progress, failed, or passed).

## 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 browsers.
* Average/Normal usage performance goal: Pages should load and respond within 2 seconds for standard/basic actions.
* The system shall be updated periodically to ensure the latest driver training material is available:
* DMV updates: On demand or as soon as updates are received.
* System maintenance/patches: At least once per quarter or as needed for security updates.

#### 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 shall be operating system–agnostic, meaning it can run on Windows, Mac, or Linux, as long as a modern web browser is available.
* The back end requires a database (ex: MySQL, PostgreSQL, or SQLLite but not csv form) to store user, reservation, and package data.
* A cloud hosting platform is preferred, so routine backups and infrastructure security are handled off-site.

#### 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?*

* The system must clearly distinguish between user roles to ensure appropriate data access.
* Case sensitivity: Usernames could be case-insensitive, but passwords must be case-sensitive to maintain security.
* The system shall notify the admin (IT) whenever a critical error occurs.

#### 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?*

* Admin should be able to add, remove, or modify users and their roles without changing code.
* The system shall accommodate platform updates (e.g., browser updates, minor server OS updates) with minimal downtime.
* The IT admin needs full permissions to reset passwords, deactivate accounts (employee or customer), and handle data management tasks.

#### 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 Authentication: User must log in with a valid username and password. Password resets should be self-service or handled by IT.
* Secure Connection: All data exchange should occur over HTTPS/TLS encryption.
* Brute Force Protection: If there are multiple failed login attempts, The system shall temporarily lock the account and alert an admin.
* Forgot Password: Users must be able to reset their password via a secure password reset link or email validation flow.

### 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.”*  
  
*Breaking them down into sections:*

* **User Management**
  + The system shall allow users to create accounts with personal information such as first name, last name, address, phone, and state.
  + The system shall validate user credentials at login.
  + The system shall allow admins to reset or block a user’s access.
  + The system shall allow users to reset their own password if forgotten.
* **Reservation Management**
  + The system shall allow customers to schedule driving lessons via an online portal or by calling or visiting the office.
  + The system shall keep track of which customer is assigned to which driver, car, date, and time.
  + The system shall log changes to reservations including creation, modification, and cancellation, along with the user who made the change.
  + The system shall allow customers to cancel or modify reservations online if necessary.
* **Package Management**
  + The system shall offer three initial packages consisting of 6, 8, or 12 hours of in-car training plus additional services.
  + The system shall allow an admin to enable or disable packages as needed without code changes.
* **DMV Integration**
  + The system shall connect to the DMV’s resources to receive any updated rules, policies, or sample questions.
  + The system shall notify the admin when new updates are received from the DMV.
* **Online Classes and Practice Tests**
  + The system shall display practice tests, track progress, record scores, and note pass or fail status.
  + The system shall store test history for each customer including test name, time taken, score, and status.
* **Reporting**
  + The system shall generate activity reports indicating which user performed which action at what time.
  + The system shall allow for exporting data to Excel for offline analysis.

### 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:**
  + **Customers:** Need to register, log in, schedule or cancel or modify lessons, take online tests, view progress, and update personal details.
  + **Secretary:** Needs to create or update customer records, schedule lessons by phone or in person, run simple reports, and handle inquiries.
  + **IT or Admin:** Needs full control to manage accounts, view all reservations, generate advanced reports, and reset or remove user access.
  + **Owner Liam:** Has privileges similar to IT or admin and can view all data, track usage, and review reports.
* **Interface Requirements:**
  + Must be web-based and mobile-friendly, enabling customer and employee access from any device with an internet connection.
  + Provide clear navigation to key functions such as scheduling, packages, practice tests, and profile management.
  + Include easy forms for registration and scheduling, with minimal steps to reduce user error.
  + Show relevant data in tables such as lesson times, driver comments, and test statuses with sorting and filtering options.

### 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?*

* [Insert text]

### 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?*

* [Insert text]

### 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.*

[Insert chart]