# 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?*

* Liam, the owner of DriverPass, wants to create a system to provide better driver training services.
* The system should offer online classes and practice tests.
* It should also provide on-the-road training if desired.

### 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?*

* Data Accessibility: The system should allow data access online and offline, with data synchronization to prevent redundancy.
* Security: Different roles should have varying levels of access, with IT having full access.
* Tracking: The system should track user activities, such as reservations and modifications.
* Reservations: Customers should be able to schedule driving lessons online or through phone calls.
* Customizable Packages: There are three driving lesson packages, and Liam wants the system to be flexible for future customization.
* Registration: Collect customer information, including personal details and payment information.
* Compliance: Stay up-to-date with DMV regulations and receive notifications of updates.
* Cloud-Based: The system should run on the cloud for easy maintenance and security.
* User Interface: Liam has a specific design in mind for the interface, including test progress and driver notes.

### 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 completed DriverPass system should be able to perform the following functions:

1. User Registration and Management:
   * Allow users to register with their personal information.
   * Store user data securely.
   * Enable users to reset their passwords if forgotten.
2. Driving Lesson Reservations:
   * Allow users (students) to schedule driving lessons.
   * Provide options for selecting the desired package (e.g., Package One, Package Two, Package Three).
   * Assign available drivers and cars to scheduled lessons.
   * Track reservations, modifications, and cancellations.
3. User Roles and Permissions:
   * Define roles (e.g., administrator, IT officer, secretary, student, driver) with specific permissions.
   * Administer access control for various system functionalities.
4. Data Accessibility:
   * Ensure data can be accessed online and offline.
   * Synchronize data to prevent duplication and maintain consistency.
5. Security and Authentication:
   * Implement user authentication and authorization mechanisms.
   * Safeguard sensitive user data, including credit card information.
6. Activity Tracking and Reporting:
   * Log and track user activities, such as reservations, cancellations, and modifications.
   * Generate activity reports to identify responsible users.
7. Compliance with DMV Regulations:
   * Establish a connection with the DMV to receive updates on rules, policies, and sample questions.
   * Ensure that training materials and practices align with DMV requirements.
8. Cloud-Based System:
   * Host the system on the cloud for scalability, reliability, and easy maintenance.
   * Manage backups and security in the cloud environment.
9. User-Friendly Interface:
   * Design an intuitive and visually appealing user interface (UI).
   * Include features like test progress tracking and driver notes.
10. Flexibility for Package Customization:
    * Allow for the customization of driving lesson packages as per client's requirements.

To achieve these goals, the system design should include the following measurable tasks:

1. System Requirements Specification:
   * Document detailed system requirements based on client interviews and needs.
2. Database Design and Implementation:
   * Design the database schema to store user data, reservations, and other relevant information.
   * Implement the database structure and ensure data integrity.
3. User Registration Module:
   * Develop a user registration module that collects and stores user information securely.
   * Implement password reset functionality.
4. Driving Lesson Reservation Module:
   * Create a module that allows users to schedule, modify, and cancel driving lessons.
   * Implement the assignment of drivers and cars to lessons.
5. Role-Based Access Control:
   * Define user roles and permissions.
   * Implement access control mechanisms based on user roles.
6. Data Synchronization and Offline Access:
   * Design data synchronization processes to ensure data consistency across online and offline modes.
7. Security and Authentication System:
   * Implement user authentication and authorization methods.
   * Ensure the encryption of sensitive data.
8. Activity Logging and Reporting:
   * Develop an activity logging system.
   * Create reports for tracking user activities.
9. DMV Compliance Integration:
   * Establish a connection with the DMV for updates and notifications.
   * Implement mechanisms to incorporate DMV-mandated changes.
10. Cloud Hosting and Backup:
    * Deploy the system on a cloud platform.
    * Set up automated backup and security measures.
11. User Interface Development:
    * Design and develop a user-friendly interface as per client specifications.
12. Package Customization Module:
    * Create a module to customize driving lesson packages.
13. Testing and Quality Assurance:
    * Conduct comprehensive testing to ensure system functionality, security, and user experience.
14. System Deployment:
    * Deploy the system on the cloud and configure it for production use.
15. Client Training and Handover:
    * Provide training to the client's team for system usage.
    * Hand over the completed system to the client.
16. Documentation and Support:
    * Prepare user documentation.
    * Provide ongoing technical support as needed.

These measurable tasks will guide the development and implementation of the DriverPass system, ensuring that it meets the client's requirements and delivers the desired functionality.

## 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?*

* [Insert text]

#### 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?*

* [Insert text]

#### 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?*

* [Insert text]

#### 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?*

* [Insert text]

#### 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?*

* [Insert text]

### 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.”*

* [Insert text]

### 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.)?*

* [Insert text]

### 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]