# CS 255 Business Requirements Document

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?*

This project aims to develop a comprehensive system for DriverPass, serving Liam, the owner, as the client. The system's objectives are to:

* Offer driver training services, including online classes, practice tests, and on-the-road training.
* Ensure accessible and secure data management, online and offline.
* Implement user role-based security and activity tracking.
* Enable customer reservations for driving lessons, including various packages.
* Support online appointment management and user registration.
* Ensure DMV compliance and provide real-time updates.
* Host the system on the web or cloud with a specific interface design.

### 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 seeks to address the issue of inadequate driver training by creating a comprehensive system. Key components include:

* User Management
* Data Access and Security
* Reservation System
* Compliance and Updates
* User Interface
* Database Management
* Business Logic
* Testing and Quality Assurance
* Documentation

### 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?*

Upon completion, the system should:

* Deliver high-quality driver training services.
* Allow seamless data access across devices and locations.
* Ensure data security and user privacy.
* Enable user action tracking and accountability.
* Streamline customer reservations and appointments.
* Maintain DMV compliance and timely updates.
* Ensure system reliability and user-friendly interface.

To achieve these goals, the system design will encompass user management, security measures, reservation systems, compliance monitoring, interface design, database management, business logic, testing, and documentation.

## 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?*

* **Environment**: The system needs to run in a web-based environment, allowing users to access it via web browsers and mobile devices.
* **Speed**: The system should provide a responsive user experience, with page load times ideally under 2 seconds for standard operations.
* **Update Frequency**: The system should receive updates regularly, with critical updates deployed immediately, and non-critical updates scheduled for quarterly releases.

#### 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?*

* **Supported Platforms**: The system should run on multiple platforms, including Windows, macOS, and various Unix-based systems.
* **Backend Tools**: The system relies on a robust database system for data storage and retrieval, with PostgreSQL selected as the preferred database management system.

#### 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?*

* **User Identification**: User identification should be case-insensitive, ensuring that "JohnDoe" and "johndoe" are treated as the same user.
* **Error Notifications**: The system should notify the admin immediately when a critical error occurs, ensuring prompt resolution.

#### 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?*

* **User Customization**: The system should allow administrators to make changes to user roles, permissions, and profiles without requiring changes to the underlying code.
* **Platform Updates**: The system should be designed to adapt to platform updates, with a quarterly review of compatibility with new platform versions.
* **IT Admin Access**: IT administrators should have full access to system configuration, user management, and the ability to modify system behavior.

#### 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 login requires a combination of username and password, with password complexity requirements.
* **Data Security**: Data exchanged between the client and server should be encrypted using industry-standard SSL/TLS protocols.
* **Brute Force Protection**: The system should have mechanisms in place to detect and respond to brute force hacking attempts, including temporary account lockouts.
* **Password Reset**: Users who forget their password should have the option to reset it through a secure email-based process.

### 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 schedule driving lessons online or through phone calls.
* The system shall provide different lesson packages for customers to choose from.
* The system shall match customers with available instructors, cars, and lesson times.
* The system shall track the progress of each customer's driving lessons.
* The system shall generate notifications for scheduled lessons, changes, and updates.
* The system shall allow instructors to provide feedback and notes on each customer's lesson.
* The system shall support customer inquiries and communication through the interface.
* The system shall maintain a secure database of customer and instructor information.

### 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.)?*

* **Interface Needs**: The user interface should cater to customers, instructors, and administrative staff.
* **Customer Interaction**: Customers should be able to schedule lessons, view progress, and communicate with instructors.
* **Instructor Interaction**: Instructors need to access schedules, provide feedback, and communicate with customers.
* **Administrative Access**: Administrative staff should have full access to manage users, lessons, and system configuration.
* **Multi-Platform Support**: The interface should be accessible through web browsers on desktops and mobile devices.

### 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?*

* **Technology Proficiency**: Users are assumed to have basic proficiency in using web-based interfaces.
* **Internet Connection**: Users are assumed to have reliable internet access to use the system.
* **Data Security Awareness**: Users are expected to understand the importance of data security and password protection.
* **Legal Compliance**: Assumption that the system will adhere to relevant data protection and privacy regulations.

### 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?*

* **Resource Constraints**: The project is subject to budget and time constraints, limiting the scope of features and testing.
* **Technology Limitations**: The system's functionality may be limited by the capabilities of the selected technologies and tools.
* **Scalability**: The initial design may not fully account for future scalability requirements, which may necessitate adjustments in subsequent phases.
* **User Dependency**: The system's effectiveness relies on users providing accurate and timely information.

### 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.*

A graph with a line

Description automatically generated with medium confidence