# 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 design and implement a system for Liam to provide comprehensive driver training, including online classes, practice tests, and on-the-road training, for his company. The client, wants the system to manage user reservations for driving lessons, handle administrative and security functions internally, and allow for flexibility in accessing and managing data both online and offline. The system should support different user roles, track changes to records, integrate with DMV updates, and be adaptable to future modifications, such as customizing training packages.

### 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 address the high failure rates in driving tests by offering a structured training program that includes online classes, practice tests, and physical driving lessons. The system needs to support online and offline access to data, handle reservations for driving lessons, and track modifications to records to ensure accountability. It should also provide a secure, role-based access control for different users, such as the IT officer and secretaries, and integrate real-time updates from the DMV. Different components of the system include user account management, reservation scheduling, data tracking, security features, and a web-based interface that supports both administrative tasks and customer interactions. The system must also be flexible enough to allow future modifications to training packages and adhere to updated DMV regulations. All together they will require a relational database that scales well. Something like MySQL for the backend and SQLite for the client-side/offline data. Alternatively they could use a NoSQL database and standard JSON for the offline data.

### 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?*

On completion of the system, DriverPass should be able to facilitate online and in-person reservations for driving lessons, manage user accounts, allow users to modify or cancel appointments, and enable seamless access and management of data both online and offline. It must include measurable tasks such as user authentication, online and offline data synchronization to prevent redundancy, role-based access controls, activity logging for audit purposes, and compliance with DMV updates. Additionally, the system should support customizable driving lesson packages and generate various reports, while ensuring the interface is user-friendly and operates efficiently in a cloud environment. It will need to be abstract enough to allow for on the fly customization but solid enough to withstand growth. This requires building a strong, type safe backend, simple user interface, and a reliable database integration, all while maintaining high security, encryption, and compliance standards.

## 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 DriverPass system should run in a web-based environment, accessible from any computer or mobile device to facilitate online classes, practice tests, and appointment scheduling. It should also have an offline capability for downloading personal reports and information to be worked on using applications like Excel. The system should operate efficiently, with minimal latency to ensure a smooth user experience, particularly for real-time scheduling and data access. Depending on the external dependencies, updates to the system should be conducted regularly, ideally on a monthly basis, to incorporate new features, security patches, and compliance updates from the DMV to ensure that the training content remains current and accurate.

#### 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 platform that the system should run on can be whatever the development team is most comfortable with. I choose to work in exclusively Linux environments, due to the low cost of setup and the wide range of tools available for development. The Linux platform is useful for taking advantage of its stability, security, and flexibility, but these same features can be found in Windows or MacOS. At the end of the day, if you have a team that works better with the Windows or Mac ecosystem, then they will probably be able to develop safer than in a new environment.

The backend will require tools like a relational database, PostgreSQL or MySQL are my gotos. This will be to support data storage and management for user accounts, lesson scheduling, and tracking user progress. Additionally, depending on the size of the network a web server such as Apache or Nginx might be needed to handle HTTP routing, and a development framework like Boot-Strap, Node.js, Django, or Cargo-Leptos can be used to streamline the development of the application logic.

#### 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?*

We will distinguish between different users by implementing role-based access controls, where each user is assigned a specific role such as Developer, Admin, User Manager, Instructor, Student, or Secretary. These roles will determine the permissions and access levels within the system. User authentication will be managed through unique usernames, passwords and possibly an OAuth system. I’ve been a fan of even the username being case-sensitive to enhance security, not not everyone has hopped on that band-wagon yet.

The system should inform the admin of a problem in real-time or as soon as an issue is detected. This includes system errors, security breaches, failed multiple login attempts, data inconsistencies/corruptions, or any failures in booking and scheduling functions. Notifications can be sent via email, SMS, or through push-notification like a dedicated admin dashboard alert system to ensure prompt attention and 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?*

Absolutely, modifying users will exclusively be done outside of code. There should never be a hard-coded user in this system. The system will adapt to platform updates by implementing a good update management process that includes regular monitoring of platform updates, thorough testing in a staging environment, and automated deployment tools to ensure smooth transitions without disrupting service. An IT admin will need full administrative access to the system, which includes the ability to manage user accounts, reset passwords, configure system settings, and perform maintenance tasks. Additionally, the IT admin should have access to system logs, performance metrics, and security alerts to monitor the system's health and respond to issues promptly.

#### 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?*

To log in, users must provide a unique username and a strong password. We should also implement a password strength test like most good websites have. To secure the connection and data exchange between the client and the server, the system will implement HTTPS with SSL/TLS encryption, making sure that all data is protected going between the client and server. Additionally, OAuth can be added for extra security. In the event of a "brute force" hacking attempt, the system will detect multiple failed login attempts and temporarily lock the account, notifying the user that they have been locked out until they can reach an Admin or User Manager. Furthermore, if a user opts for OAuth, we won’t have to worry about “brute force” attacks as most OAuth providers, like Microsoft, Google and GitHub already utilize their own features.

If a user forgets their password, the system will offer a secure password recovery process, or have them contact an admin. This will involve sending a password reset link to the user's registered email address or mobile phone number. The link will direct the user to a secure page where they can set a new password after verifying their identity through security questions. This ensures that only authorized users can reset their passwords, maintaining the security of the system.

### 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 users to create new accounts and manage their profiles.
* The system shall allow users to use an OAuth provider.
* The system shall enable users to schedule, modify, and cancel driving lessons online.
* The system shall provide an interface for administrators to manage user roles and permissions.
* The system shall track and log user activities, including reservations, modifications, and cancellations.
* The system shall generate and allow for downloadable reports that can be viewed offline using applications like Excel.
* The system shall send notifications to users and administrators for important events, such as appointment confirmations and security alerts.
* The system shall allow the admin or user manager to reset passwords and manage system configurations.
* The system shall support different training packages and allow customization of these packages by authorized users.
* The system shall ensure secure data transmission using HTTPS with SSL/TLS encryption.
* The system shall provide a password recovery process for users who forget their passwords.
* The system shall integrate with the DMV to receive updates on rules, policies, and sample questions.
* The system shall display test progress, scores, and status for users taking online practice tests.
* The system shall allow instructors to leave comments and notes about driving lessons.
* The system shall automatically lock user accounts after multiple failed login attempts to prevent brute force attacks.
* The system shall enable the admin to monitor system health, performance metrics, and security alerts.
* The system shall support ~real-time data access and updates while preventing data redundancy.

### 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 User Interface for the DriverPass system needs to be intuitive and user-friendly, able to be used by different users such as students, instructors, administrators ...etc. Students need to be able to easily schedule, modify, and cancel lessons, access online classes, and view their progress and test results. Instructors require functionality to manage their schedules, view student information, and leave lesson notes. Administrators need tools for managing user roles, permissions, and generating reports. The admin needs access to system configurations, user management, and security monitoring. The UI should be accessible via web browsers on both desktop and mobile devices, creating a seamless interaction across different platforms, with responsive design principles to accommodate various screen sizes and 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?*

There were a few aspects that were not specifically addressed in the design above, that would make this application better. We should implement detailed error handling procedures, and integration with third-party payment gateways for processing transactions. Integration with a third-party payment gateway is a big one, because making a payment system that complies with all state and national laws would take a lot of time, considering companies like PayPal have already taken these into consideration. The design assumes that users have access to reliable internet connections and modern web browsers or mobile devices capable of running the web-based application smoothly. It also assumes that users are somewhat tech-savvy and can navigate an online interface with minimal assistance.

### 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?*

I totally agree that any system we build will have limitations. The system design limitations would include reliance on stable internet connections, which for now, can be a barrier for users with limited connectivity, and the complexity of implementing role-based access controls as well as creating a team dedicated to continuous DMV integration. These could lead to higher costs and longer development times. Ensuring strong security for sensitive information is somewhat challenging but fairly standard, and resource constraints may limit the availability of skilled developers for maintenance.

### 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.*

