# CS 255 Business Requirements Document Template Pittman

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 DriverPass project aims to create a comprehensive system to support driving test preparation for students at the DMV. The client, DriverPass, is focused on helping more individuals pass their driving tests by providing online classes, practice exams, and on-the-road training options. They want a system that enables seamless access to data from any location, whether online or offline, to manage the entire training process efficiently. This includes scheduling driving lessons, providing up-to-date practice resources, and tracking the progress of each customer. The system is also expected to support various user roles with differing access levels, ensuring secure, role-based access to data, while tracking user activities and generating reports as needed.

### 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’s vision includes a flexible, cloud-based system that minimizes technical maintenance for the business and allows for easy updates in response to DMV regulations. The key objectives of this system are to streamline the booking and tracking of driving lessons, deliver online learning resources, and ensure regulatory compliance. Core components required to achieve this include user management with defined access roles, a booking interface, tracking capabilities, and reporting tools.

### 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 allow customers to book, modify, or cancel lessons online, track their progress in the training, and provide DriverPass with a secure, scalable platform to support its mission. The system must include key measurable tasks such as implementing user management with defined roles, developing booking, tracking, and reporting functionalities, and ensuring DMV compliance through regular updates and notifications.

## 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 must operate on a web-based platform accessible via both desktop and mobile devices, leveraging a cloud-hosted infrastructure to ensure minimal maintenance and high availability. The system is expected to respond to user actions, such as booking lessons or accessing reports, within two seconds to maintain a seamless user experience. It should also support at least 100 concurrent users without performance degradation. Updates to the system must be deployed quarterly or as needed to comply with DMV regulation changes, with provisions for urgent hotfixes to be implemented within 24 hours. This ensures the system remains reliable, efficient, and aligned with regulatory requirements.

#### 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 DriverPass system must be compatible with common operating platforms, including Windows, macOS, and major Linux distributions, to ensure accessibility for all users. The system's back end requires a robust relational database, such as MySQL or PostgreSQL, to manage user data, lesson scheduling, and tracking efficiently. Additionally, the application should integrate with cloud services for storage and scalability, leveraging tools like Amazon Web Services (AWS) or Microsoft Azure for hosting and database management to minimize on-premise infrastructure needs.

#### 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 DriverPass system will distinguish between users by assigning unique identifiers to each account, such as email addresses or user IDs, ensuring role-based access for administrators, secretaries, and customers. All user input, such as usernames and passwords, will be case-sensitive to enhance security. The system will inform the administrator immediately if critical issues occur, such as failed login attempts exceeding three attempts (indicating potential brute force attacks), database connectivity issues, or errors that prevent booking or modifying lessons. Notifications will be sent via email or displayed on the admin dashboard to ensure prompt resolution of any problems.

#### 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?*

The DriverPass system will allow changes to users, such as adding, removing, or modifying accounts, through a user-friendly administrative interface without requiring code changes. This flexibility ensures that user management tasks can be performed efficiently by authorized personnel. The system will adapt to platform updates by leveraging cloud-based infrastructure and scalable frameworks that accommodate operating system and browser upgrades seamlessly. Regular compatibility testing will ensure smooth operation across updated platforms. IT administrators will have full access to manage user roles, monitor system activities, and implement updates. This includes rights to reset passwords, disable accounts, and manage system settings to maintain security and functionality.

#### 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 valid username and password, with optional two-factor authentication (2FA) for enhanced security. The connection and data exchange between the client and server will be secured using HTTPS with TLS encryption to protect against interception and eavesdropping. All sensitive data, such as passwords, will be stored securely using hashed and salted encryption in the database.

In the event of a brute force hacking attempt, the system will lock the targeted account after three consecutive failed login attempts and notify the administrator immediately. The user will be required to complete additional verification steps to unlock their account, such as confirming their identity via email or answering security questions. If a user forgets their password, they can initiate a password recovery process by entering their email address. The system will send a secure link to reset the password, ensuring no sensitive information is exposed during the 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 during login to ensure secure access.
* The system shall allow customers to schedule, modify, and cancel driving lessons online.
* The system shall enable administrators to manage user accounts, including adding, modifying, and disabling users.
* The system shall provide role-based access, ensuring users can only access features relevant to their roles (e.g., admin, secretary, customer).
* The system shall track and log all user actions, such as reservations, cancellations, and modifications, for audit purposes.
* The system shall allow customers to select and purchase driving packages based on their needs.
* The system shall notify users of upcoming lessons via email or SMS.
* The system shall generate reports for administrators, including activity logs and customer progress summaries.
* The system shall provide practice tests and online learning materials for customers enrolled in specific packages.
* The system shall connect with the DMV database to receive updates on rules, policies, and test requirements.
* The system shall allow customers to reset their passwords securely through an automated recovery process.
* The system shall integrate with cloud storage for secure data backup and retrieval.

### 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 DriverPass system's user interface must be intuitive, user-friendly, and accessible across various devices, including desktops, laptops, tablets, and mobile phones, through a browser-based platform optimized for responsive design. Administrators will use the interface to manage user accounts, including adding, modifying, and disabling users, while also accessing activity logs and generating reports. They will monitor system notifications for critical issues, such as failed login attempts or system errors. Secretaries will rely on the interface to schedule, modify, and cancel driving lessons for customers, input customer details for phone or in-person bookings, and manage driver schedules. Customers will use the interface to create accounts, log in, reset passwords, schedule and modify lessons, track progress on practice tests, and access learning materials and practice exams included in their selected packages.

The interface will be accessible primarily through a web browser, with a responsive design ensuring compatibility across different screen sizes and devices. Mobile-friendly features will enhance usability for customers and secretaries, while administrators will have access to desktop-specific tools for comprehensive management tasks. Key features include simple navigation with clear labels and action buttons, visual progress tracking for customers, calendar integration for lesson scheduling, and secure login with optional two-factor authentication to protect user accounts.

### 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?*

The design assumes that all users will have reliable internet access to interact with the web-based system for features such as scheduling lessons and accessing learning materials. It also assumes that users will have access to modern devices, such as smartphones, tablets, or computers, capable of running a compatible web browser.

For administrative and secretarial tasks, it is assumed that staff members are familiar with basic computer operations and can navigate the interface without extensive training. The design presumes that the system's cloud-based infrastructure will handle scalability and performance needs without requiring on-premise servers. Additionally, it assumes that the DMV will provide timely updates regarding rules, policies, and test requirements in a format compatible with system integration.

Finally, it is assumed that any changes to customer packages, such as disabling or modifying offerings, will not require substantial reconfiguration of the system beyond administrative inputs through the provided interface.

### 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?*

The DriverPass system has several inherent limitations. From a resource perspective, the initial design may not include advanced features such as full automation of DMV updates or the ability for non-developers to add or remove system modules without assistance. Budget constraints may limit the scope of initial development, prioritizing core functionalities like scheduling, tracking, and security while postponing advanced analytics or additional integrations for future updates.

Time constraints may also restrict the implementation of non-essential features, such as highly customizable user dashboards or advanced reporting tools, until subsequent development phases. From a technology standpoint, reliance on cloud-based infrastructure assumes consistent internet connectivity, meaning offline functionality will be limited to specific scenarios like downloading reports. Compatibility with future platforms and technologies may also require ongoing updates and maintenance.

Lastly, as a web-based system, the design may not fully meet the needs of users in areas with limited or unreliable internet access, potentially requiring additional solutions to ensure universal usability.

### 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.*

