# 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?*

### DriverPass seeks a custom system tailored to their existing business model. They currently provide driving tests and practice sessions for students preparing to obtain their driver's licenses through the DMV. The proposed system will support their growth by enabling online reservations, scheduling, and payment processing for users. Additionally, it will grant DriverPass staff access to user profiles and schedules, while efficiently managing driver assignments for student sessions.

### 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 has identified a promising gap in the market: many individuals preparing for their DMV driving skills test are underprepared and often fail. To address this need, DriverPass aims to offer enhanced practice exams and comprehensive driver training. Supporting this initiative requires a robust system capable of managing user accounts, enabling scheduling for driving tests and practice sessions, and allowing users to monitor their progress and make online payments for services. Additionally, the system will facilitate internal operations by tracking DriverPass employees, the training vehicles they use, and user-related data. The entire platform will be cloud-based, ensuring seamless access from any internet-connected device.

### 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?*

Objective Task

House a database of users

Develop a function that enables users to securely enter personal information into their account, accessible through a login system. Implement a cloud-based database with robust security measures to store this data.

Intuitive U/I

Obtain customer approval for the overall design before proceeding with user interface development. Design a client-facing UI that displays account details, including personal information, progress tracking, driver notes, scheduled services, and a profile photo. Develop an admin interface that provides tools for scheduling, vehicle assignment, payment processing, and access to client account information.

Automatic schedule matching for employees

Develop a function to manage and organize the current roster of employees and vehicles at DriverPass. Implement a scheduling system that pairs available employees with suitable cars to conduct driver training sessions on user-specified dates. Include a modification feature that enables users to manually adjust or override the automatically generated assignments.

Roles for employees

Define a parameter to differentiate employee accounts from general user accounts. Additionally, establish a parameter for each employee role within the system, specifying the associated permissions for each role.

Automatic updates from DMV

Develop a function that periodically monitors the DMV website or database for updates. Implement a notification system that alerts administrator accounts whenever new DMV information becomes available.

Cloud access for all users

Create a cloud-based system architecture. Implement a function to aggregate targeted data sets and export them as CSV spreadsheet files. Create a function to upload revised data back to the cloud, ensuring the cloud database reflects the latest updates.

User profile interactions

Develop a function that enables users to view and manage their service schedule, including options to modify, cancel, or complete payments. Users should also be able to select from three available service packages. Implement a function that presents their progress and activity through a user-friendly dashboard interface.

Test System

Testing team will use static and dynamic tests to ensure the system performs correctly.

Deliver a finished product

System testing with the test-team must be finished before the product can be delivered.

## 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?*

Cross-Platform Compatibility: The DriverPass system must be accessible on both desktop platforms (PCs and Macs) and smartphones. To achieve this, a web application should be developed that functions seamlessly in standard desktop browsers like Firefox and Chrome, while a dedicated mobile app should be created to support smartphone users.

Performance Requirements: The system must deliver sufficient speed and responsiveness to support video streaming, a core component of the DriverPass business model. Although streaming quality will depend on individual users' bandwidth, the platform should be capable of handling multiple concurrent users streaming driver education videos without performance degradation.

Ongoing Maintenance and Updates: The system must be maintained in alignment with updates to the browsers and mobile operating systems it relies on. For example, when Google Chrome releases a new version, the DriverPass web app should be tested for compatibility, any necessary adjustments should be implemented, and the system should be updated accordingly to ensure continued functionality.

#### 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?*

If DriverPass is built as a web-based system using HTML, it will inherently be compatible with any operating system that supports modern web browsers. For mobile access, the application should be available on both Android and iOS platforms. While alternative mobile operating systems do exist, marketing efforts should encourage users on those platforms to utilize the web version until a native app is developed assuming market research justifies investment. To manage user data including clients, administrators, and drivers a database must be implemented and integrated with the DriverPass interface. Developers will also need to select a backend framework, such as ASP.NET Core or Node.js, based on their preferences. SQL remains the standard choice for handling server-side database operations.

#### 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?*

#### Each user will be assigned a unique username for site access, which will be stored in a centralized database. If a new user attempts to register with a username that already exists, the system will block the registration and display an error message. Both usernames and passwords are case-sensitive meaning “Bob123” and “bob123” are treated as distinct accounts.

#### Whenever an error occurs within the system, it should be both logged and immediately reported to the administrator. Examples of issues that warrant admin notification include excessive resource consumption by processes, improper CPU usage, memory problems related to cache size, zombie processes, abnormal load averages, and disk read/write performance anomalies.

#### 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?*

Since users are treated as objects within the system and stored in the database, it will be possible to add, remove, or modify user records without altering the underlying system code. When properly implemented, the interface will allow updates to individual classes without requiring a full system refactor.

To maintain the system effectively, IT personnel will require administrator-level access. They will be able to view usernames and certain user details and will have permission to update specific information. However, they will not be able to view the actual data being modified. For instance, if a user requests a change to their payment method, IT support can input the new payment details into the user’s account, but they will neither see the previous payment information nor have visibility into the new data once it has been submitted.

#### 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?*

Users must create a unique user ID and a strong password, which includes a minimum of 10 characters, a mix of uppercase and lowercase letters, numbers, and special symbols.

All data transmissions will be secured using HTTPS over TLS, ensuring the confidentiality and integrity of user information.

In case of a forgotten password, users will have several attempts to log in. After exceeding the allowed number of attempts, the account will be locked, requiring users to contact IT support for a password reset—either directly or through the "Forgot Password" feature.

This security protocol is designed to prevent brute-force login attempts from external sources.

### 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 will authenticate user credentials during login.

It will implement two-factor authentication for all users.

It will monitor and record each user's testing progress.

User data including driver notes, special needs, photos, address details, and payment information will be securely stored in private accounts.

Users will be able to schedule driver tests and practice sessions by selecting from three predefined packages.

Online payment functionality will be available to all users.

The system will maintain records of business employees.

It will manage scheduling for vehicles assigned to employees.

User data and progress will be continuously tracked and updated.

### 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 system will be accessible from any internet-enabled device that supports user interaction, including smartphones, tablets, and PCs. The application will serve three types of users: customers, employees, and administrators.

Customers will be able to log in using their credentials to view account status and progress, check balances, and schedule driving tests and practice sessions. They will also have the ability to update their profile, account details, and billing information.

Employees will have access to their work schedules and the schedules of vehicles assigned to them. They can view and update their employee profiles, including setting their availability.

Administrators will have broad access to user and employee data. They can reset passwords, unlock accounts, manage employee and user schedules, and add or remove users from the system. Admins will also receive updates from the DMV regarding driving tests and can schedule appointments for users.

The user interface will support both touch and click interactions, with keyboard input available for entering names and other relevant information. It will be designed to be intuitive, streamlined, and optimized for 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?*

All users are expected to access the application using a modern device, such as a smartphone or personal computer, via an up-to-date web browser.

The programming language for developing the application has not yet been determined.

The final design elements including the color scheme and overall aesthetic remain undecided. Additionally, it is unclear what accessibility or customization settings will be available to users, such as options for color-blind modes, low-bandwidth optimization, or high-contrast viewing.

The hosting solution for the driver-test videos is still under consideration. If hosted on the DriverPass server, it may consume significant bandwidth and storage. Alternatively, using a free third-party platform like YouTube could alleviate bandwidth concerns but might introduce costs or complications, such as advertisements that conflict with DriverPass’s business model.

### 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?*

Time and Team Constraints: The scope of development may be limited by the available timeframe and the size of the team. A smaller team working under tight deadlines might need to scale back or simplify certain features to meet delivery goals.

Bandwidth Considerations: The performance of the user interface will depend on the database host’s bandwidth capacity. If bandwidth is limited, the UI may need to be optimized with smaller images and reduced animation to ensure smooth functionality.

Budget Uncertainty: The project budget has yet to be defined. Should expenses exceed expectations, a meeting with the DriverPass CEO will be necessary to reassess project requirements or secure additional funding.

### 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 orange dots

AI-generated content may be incorrect.