# 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?*

This project deals with the designing of the system that would better help students prepare for driving tests. The objective of this work is to fill in the existing market gap by providing online practice exams and on-road training.

DriverPass is the client, owned by Liam, a driving education company which is set out to reduce the rates at which driver's licenses exams fail.

Client Needs:

* Allow customers to book, amend, and cancel driving lessons online or over the phone.
* Offer three training packages with different features, including on-road training and practice tests.
* Track customer progress on online tests and provide activity reports. Provide role-based access to employees for security and tracking of data. Integrate with DMV systems for real-time updates to rules and practice tests. Be accessible with both web and mobile devices, and data modification can be availed only online.

### 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?*

What DriverPass wants the System to Do:

* Provides online practice exams and on-the-road driving lesson scheduling.
* Tracks customer progress, including test results and lesson completion.
* Allows role-based access for employees performing different functions.
* Allows flexibility in scheduling an appointment, modifying it, or canceling it over the internet or by telephone.
* Integrates with DMV systems to stay current on rules and test content.

The Problem They Want to Solve

DriverPass aims to solve the problem of a high failure rate, 65%, of applicants for driving tests. The problem is that students rely on outdated or insufficient test preparation tools. DriverPass focuses on providing effective resources and personalized training to raise success rates.

Components Needed for the System

1. Web-based User Portal:

* Customer booking of lessons, practice test capabilities, and tracking their progress online.

1. Scheduling System :

* Manages driving lesson bookings, trainers, cars, and timeslots.

1. Role-Based Access System:

* Provides different levels of access for the owner, IT officer, secretary, and customers.

1. DMV Integration:

* Keeps the system updated with DMV changes in practice tests and training.

1. Reporting Tools:

* Tracks user activities like reservations, modifications, and cancellations.

1. Scalable Design:

* Allows package customization and possibly even future expansion.

### 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?*

When implemented, the DriverPass system shall:

1. Provide Comprehensive Training:

* Offer online practice exams with tracking of progress, scores, and completion status.
* Support on-the-road training with lesson scheduling and driver feedback.

1. Streamline Appointment Management:

* Allow the customers to book, change, or cancel driving lessons online or by calling.
* Assign drivers, cars, and timeslots for each lesson.

1. Support Business Operations:

* It enables the role-based access for employees, which includes the system management of the IT officer.
* Track and log user actions for auditing and reporting purposes.
* Allow the owner to view and download business reports.

1. Ensure Security and Compliance:

* Protect user data using strong authentication and automatic password recovery.
* Keep DMV-related rules and test content current in real time.

1. Improve User Experience:

* Create intuitive web-based interfaces that can reside on computers and mobile smartphones.
* Easy access to practice test progress and driver feedback.

1. Allow for Future Scalability:

* Disable or edit training packages on an as-needed basis.
* Integrate modular design for adding features later.

Measurable tasks of System Design

1. User Features:

* Design and test an appointment scheduling system with two-hour session slots.
* Create an online class module, including practice tests and progress tracking.

1. Employee Tools:

* Apply role-based access and security layers.
* Create a reporting system for change tracking and logging.

1. Infrastructure:

* Design a cloud-based accessibility and backup platform.
* Enable real-time DMV syncing.

1. User Interface:

* Design forms by taking user's registration input and appointment details.
* Include a dashboard on practice test progress and lessons feedback.

## 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?*

1. Environment:

* The system must also be web-based and accessible with any form of modern browser from both desktop and mobile devices.
* It needs to be on a cloud-based platform for availability, scalability, and minimum downtime.
* Compatibility with common operating systems, such as Windows, macOS, Android, and iOS, is very important.

1. Speed and Responsiveness:

* Pages should load in under 2 seconds for users on standard broadband internet.
* Operations like login, appointment scheduling, and retrieval of test results should be done in less than 3 seconds.
* Average queries should generate reports and logs in about 5 seconds.

1. Update Frequency:

* Perform the security patching and system maintenance monthly, or in that regard, when needed. DMV content updates shall include updates of rules, policies, and practice questions synchronously whenever available. And scheduling major upgrades quarterly-including new features or user interface improvements-reduces disruptions.

#### 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?*

1. Platforms:

* It should run on Windows, macOS, Android, and iOS to support a wide range of users.
* It should be compatible with all web browsers, such as Chrome, Firefox, Safari, and Edge, for easy access without extra installations.
* It should be hosted on a cloud platform like AWS, Google Cloud, or Azure to ensure reliability and scalability.

1. Back-End Tools

* A relational database like MySQL or PostgreSQL is required, for the storage of user information and appointment schedules and test results.
* The web server, like Apache or Nginx, handles the job of request dispatching and serves content to users.
* A programming language like Python (using Django or Flask) or JavaScript (Node.js) should be used for server-side processing.

1. Other Misc. Support Tools:

* Integration of email service for password recovery and notifications.
* Real-time synchronization of DMV updates using APIs or similar technology.

#### 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?*

1. Distinguishing Users:

* Users will be divided into categories based on their roles, such as customers, secretaries, IT officers, and admins.
* Different levels of access, such as view, edit, or manage, when considering data, will be granted based on user roles.
* Differentiation will rely on unique usernames or email addresses used during login.

1. Input sensitivity:

* All input, with the exception of things like usernames and passwords, will be case-sensitive for security reasons.
* Other fields, like names and addresses, will not be case-sensitive to be more user-friendly in interaction.

1. Admin Notifications: The system should notify the admin in case of:

* Security breaches: A number of login failures or suspicious access patterns.
* Data Issues: Duplicate, missing, corrupted data on the database.
* System Failures: Down time of servers, or non-responsive features, or update failures.
* Customer Facing Issues: Payment failures, for instance, or recurring issues of some sort on the user accounts.

#### 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?*

1. User Management without Code Change:

* The system will provide an admin dashboard.
* Admins are allowed to add, remove, or modify users through the dashboard forms.
* Role permissions, and account statuses may be changed without coding.

1. Adapting to Platform Updates:

* The system will be based on standardized APIs and frameworks.
* Automatic updates in cloud hosting handle the changes of the platform.
* Regular maintenance will test compatibility with new operating system versions.

1. IT Admin Access:

* The IT administrator needs full control over the user accounts and their permissions.
* They need access to system logs for monitoring and troubleshooting.
* Security settings and role-based controls must be manageable by the admin.

#### 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?*

1. Login Requirements:

* It requires a unique username or email address and a decent password.
* Passwords are required to meet a certain strength criteria: minimum length, a mix of characters.

1. Securing Connections and Data Exchange

* Use HTTPS to encrypt all communications between client and server.
* Store passwords using a one-way hashing algorithm such as bcrypt.
* Implement a Secure Socket Layer (SSL) for additional connection security.

1. Brute-Force Attempts Handling:

* Temporarily lock accounts after numerous failed login attempts, say 5 tries.
* Notify users of suspect login activity via email.
* Log IP for repeated failed attempts for security review.

1. Password Recovery:

* Allowing the user to reset their passwords via a secure, timed email link.
* Verify identity by requiring the input of security questions or a two-factor authentication code.
* Prevent the previous password from being reused when resetting.

### 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 verify users' credentials at the time of login.
* The system shall provide a means for users to reset their password through a secure, email-based process.
* The system should allow administrators to create, delete, or edit user accounts via the dashboard.
* The system should allow customers to book, reschedule, and cancel driving lessons online.
* The system should assign drivers, cars, and timeslots for scheduled lessons.
* It shall provide online practice exams tracking with scores and status.
* The system should log changes made by the users through activity logs.
* It should notify the administrator when there have been several failed attempts at login.
* The system shall integrate with DMV systems to refresh practice questions and rules in real-time.
* The system will allow the use of reporting tools that permit administrators to download data associated with the business.
* It should show driving lesson information such as time, location, and driver feedback.
* The system temporarily will lock the accounts upon multiple failures to login into it.
* The system shall do password resets for IT personnel to reset passwords and grant rights to employees.
* The system should ensure sensitive data is encrypted during its storage and transmission.
* It enables administrators to temporarily disable certain training packages in case of need.

### 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:

* It needs to be user-friendly with easy navigation and clear instructions.
* It should be able to support mobile devices, desktops, and laptops through most web browsers.
* It will be necessary to provide forms from which the input design will possess, the dashboards where one can view data, and means through which users may take some actions like scheduling or reporting.

Different Users, Different Needs:

1. Customers:

* Book, reschedule, or cancel driving lessons.
* View driving packages and purchase options.
* Track progress of online practice exams and view scores.
* Recover passwords securely if forgotten.

1. Secretary:

* Schedule, change, or cancel appointments for clients.
* Manage customer profiles, including contact and payment information.
* Assign drivers and cars for lessons.

1. IT Officer:

* Reset user passwords, manage account access.
* Monitor activity logs and troubleshoot system issues.
* Set up security settings and permissions.

1. Admin [Owner]:

* Access and download business reports.
* Enable/Disable training packages.
* Track activities of employees and changes made in the system.

Interaction with the Interface

* The interface will be web-based and browser-compatible, including Chrome, Firefox, Safari, and Edge. This will also allow for mobile responsiveness to be used seamlessly on smartphones and tablets. The user is then guided through the process of efficiently performing tasks with simple input forms, clickable menus, and dashboards.

### 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?*

Unaddressed Design Elements

1. Accessibility Features:

* The design does not take note of accessibility for disabled persons, such as compatibility with screen readers or keyboard navigation.

1. Notifications and Alerts:

* Specifics about how and when users are notified via email or push notification-for example, when a lesson is due-are not explained.

1. Error Handling:

* The procedure for system error management or system downtime has not been developed.

1. Payment System Integration:

* While payment information is mentioned, it is not specified to integrate with secure payment gateways, such as PayPal or Stripe.

1. Scalability for Growth:

* The way the system would handle the increase in traffic or load of data over time isn't discussed.

Assumptions in Design

1. User Assumptions:

* Users will have reliable internet connections for online features.
* Customers know how to navigate basic websites and fill in forms.
* IT personnel are competent enough technically to perform all account management and problem-solving.

1. Technology assumptions:

* Users will be using modern devices and browsers that support HTML5 standards.
* Mobile users will access the system via responsive web design, not a native app.
* The cloud-hosted environment will handle data storage and backups, with no local interventions being required.

### 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?*

System Design Limitations

1. User Specific Constraints:

* Limited access by users not having reliable internet.
* Dependency on the fact that the end users will have modern devices or browsers that are compatible with the system.

1. Scalability:

* Initial design may not support sudden surge in users or data volume without upgrades.
* That means any addition of new features later on could require serious changes to the already existing system.

1. Security Challenges:

* The system uses encryption and appropriate protocols but is still attacked by emerging threats.
* Full observance of evolving regulations, such as GDPR and CCPA, may be needed with continuing updates.

1. Customization Limitations:

* It is not easy for users to add or modify a training package themselves without support from IT.

Resource Limitations

1. Time:

* Development deadlines sometimes can hardly allow time for extensive testing and refinement.
* The DMV integration in real time and updating the system may take more time than initially estimated.

1. Budget:

* However, some advanced tools, cloud storages, or additional features may be restricted because of budget constraints.
* High-quality security and scalability support may require expensive resources.

1. Technology:

* This tool relies on cloud hosting and, therefore, is subject to periodic outages or limitations.
* Integration with DMV systems assumes their API is reliable and compatible.

Mitigation Strategies

* It includes regular updates and scalable design to meet the present challenges of growth and security.
* Prioritize essential features to deliver a functional MVP within time and budget constraints.
* Monitor user feedback for guiding improvements and adaptations in the future.

### 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.*

