# 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 develop an online system for DriverPass, a company that provides driver training. The client, Liam (Owner), wants a system that allows customers to take online practice tests, schedule driving lessons, and a progress tracking system. The system must also support role-based access, secure data handling, and compliance with DMV regulations while being accessible via web and mobile devices.

### 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 provide a comprehensive platform for driver training, addressing the issue that over 65% of students fail their driving test due to inadequate preparation. The system should allow users to take online practice tests, book on-the-road driving lessons, and track their training progress. Additionally, it must support role-based access control for different users and ensure secure data management.
* The primary problem DriverPass wants to fix is the lack of effective training resources beyond studying past tests. Many students are unprepared for real-world driving scenarios and need structured hands-on training combined with theoretical lessons. DriverPass also needs a flexible system where administrators can track reservations, manage lesson packages, and stay updated with DMV rule changes.
* Key System Components Include:
  + User Accounts and Role Based Access
  + Online Practice Tests
  + Lesson Scheduling System
  + Instructor and Vehicle Assignment
  + Payment and Registration System
  + Compliance and DMV Integration
  + Security and Data Tracking
  + Cloud-Based Infrastructure

### 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?*

* The system should enable students to take online practice tests, schedule driving lessons, and track their progress. It must also support role-based access, ensuring that administrators, IT officers, and secretaries can manage accounts, appointments, and security settings. Additionally, it should log user activity, integrate DMV updates, and allow secure payment processing.
* Measurable Tasks Include:
  + User Authentication and Role Based Access – Implement login functionality with different user roles.
  + Online Practice Test Module – Track progress, show grades, and mark test statuses.
  + Lesson Booking System – Customer can schedule, modify, or cancel lessons online or through the secretary.
  + Instructor and Vehicle Assignment – Assign instructors and cars to scheduled lessons, ensuring there are no conflicts.
  + Payment Processing and Registration – Collect customer details, payment info, and secure transactions.
  + DMV Compliance Integration – System must receive updates from the DMV and notify admins of changes.
  + Activity Logging and Reports – Track modifications and generate reports.
  + Cloud-Based Infrastructure – Ensure system accessibility, security and data backups.

## 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 be web-based and cloud-hosted, accessible from desktops, laptops, tablets, and mobile devices to ensure flexibility for both students and administrators. It must be fast and responsive, with page loads under 2 seconds and booking transactions within 5 seconds, while the online test module should process and save answers in real-time. The system requires monthly maintenance updates for security and performance, quarterly feature updates, and immediate updates for any DMV rule or test changes to ensure compliance.

#### 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 should be platform-independent, running on Windows, masOS, Linux and mobile operating systems since it will be a web-based, cloud-hosted application accessible via modern web browsers. The back end will require a relational database such as MySQL to store user accounts, test results, lesson bookings and transaction records. Additionally, the system may use a cloud service like AWS for hosting, data storage and automated backups to ensure reliability and scalability.

#### 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 system will distinguish between users by implementing role-based authentication, where each user has unique login credentials and different levels of access. User roles will be stored in the database and enforced through access control mechanisms. Input will be case-sensitive for passwords to enhance security but not for usernames or email addresses to improve user experience. The system should notify the admin immediately when there are security breaches, failed login attempts exceeding a set limit, payment processing failures, system downtime, or scheduling conflicts. Additionally, the admin should receive alerts for critical updates, system errors and DMV rule changes that require action.

#### 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?*

* Users can be added, removed, or modified without changing code through an admin dashboard where authorized personnel can manage user accounts. The system will include role-based access controls, allowing admins to assign or revoke permissions dynamically without modifying the source code.

The system will be cloud-based, ensuring compatibility with platform updates by relying on modern web standards. It will be tested against major browser and OS updates, and backend dependencies will be updated through regular maintenance cycles to prevent compatibility issues.

The IT admin will require full system access, including user management, security settings, database access, system logs, and software updates. They should be able to reset passwords, deactivate accounts, troubleshoot issues, and monitor system performance while restricting access to critical backend configurations to prevent unintended modifications.

#### 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 enter their registered email or username and password. Multi-factors authentication can be added for extra security, requiring a one-time code sent via email or SMS.

To secure the connection and data exchange, the system will use SSL/TLS encryption to protect user credentials and sensitive information. Passwords will be hashed and salted in the database, and OAuth or token-based authentication will be sued for secure session management.

In the event of a brute-force attack, the system will lock the account temporarily after multiple failed login attempts and notify the user via email. IT admin will also receive alerts for excessive failed attempts from the same IP address.

If a user forgets their password, they can request a password reset, triggering an email with a secure time-limited rest link. IT admins will have the ability to manually reset passwords if needed while maintaining security protocols.

### 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 and enforce role-based access control.
* The system shall allow customers to register, book, modify, and cancel driving lessons online.
* The system shall assign instructors and vehicles to scheduled lessons, preventing conflicts.
* The system shall store and track customer progress on online practice tests, displaying scores and statuses.
* The system shall securely process payments and store customer billing information.
* The system shall allow users to reset passwords via a secure, time-limited email link.
* The system shall encrypt all sensitive data, including passwords and payment details.
* The system shall allow the owner to enable or disable lesson packages as needed.
* The system shall ensure uptime and data integrity with automated cloud backups.

### 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 interface must be user-friendly, responsive, and accessible on both desktop and mobile devices. It should provide a clear dashboard with relevant features for each user role, ensuring smooth navigation, secure authentication, and quick access to key functionalities.
* Secretaries
  + Manually book, modify or cancel lessons for customers.
  + View customer schedules and instructor assignments.
  + Manage customer profiles and payment records.
  + Generate reports on lesson bookings.
* IT Admins
  + Manage system users.
  + Monitor security logs and system activity.
  + Receive alerts for security threats or system errors.
  + Oversee system updates and DMV data integration.
* Owner
  + Monitor business performance and system usage.
  + Enable or disable lesson packages as needed.
  + Access financial reports and customer statistics.
  + Approve major system changes and updates.
* Customers
  + Register, log in, and manage profile.
  + Book, modify, or cancel lessons
  + Access online practice tests and view progress
  + View upcoming lessons and receive notifications.

### 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?*

* Instructor Interface: While the system track instructor assignments, their direct interaction with the platform detailed.
* Customer support integration: No mention of how users will request help if they encounter issues.
* Multi-Language Support: The design assumes English as the primary language without addressing potential multi-language functionality.
* Scalability and Future Expansion: The system’s ability to support multiple locations or franchise expansion isn’t explicitly covered.
* Payment System Details: The design assume secure payment processing but doesn’t specify whether refunds, installment payments, or different payments gateways will be supported.
* Regulatory Compliance Beyond the DMV: While DMV updates are mentioned, broader compliance with data protection laws aren’t detailed.
* Assumptions About Users and Technology:
  + Users have internet access and a modern device to interact with the system.
  + Customers are familiar with online booking systems and can navigate a web-based interface with minimal support.
  + Admins and secretaries have computer proficiency to manage reservations and report.
  + The system will be cloud-hosted, eliminating the need for local installation or on-premise servers.
  + The DMV provides an API or some form of automated data sharing for regulatory updates.

### 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 Limitations:
  + Dependence on Internet connection.
  + Limited Customization of packages.
  + Scalability constrains.
  + DMV Integration challenges.
  + User tech literacy.
* Other Limitations:
  + Development timeline – system must be completed in 4 months which can limit feature complexity.
  + Budget Constraints – DriverPass is a small business, so funding for high-end features may not be available.
  + Hosting and Maintenance Costs - Cloud hosting is required for accessibility, but ongoing costs for server space, security, and backups must be managed.
  + Testing and Security – While best practices must be followed, comprehensive penetration testing and regulatory compliance may be limited by time and budget.

### 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 gantt chart with pink rectangular shapes

Description automatically generated