# CS 255 Business Requirements Document Template

Josh Hall

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?*

* Our client, *DriverPass,* sees a need for better driver training.
* *DriverPass* would like us to develop a system to offer online training and practice tests, as well as scheduling in-person training.

### 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?*

* Online classes and practice tests for customers preparing for driving tests at the DMV.
* On-the-road training and scheduling sub-system.
* The ability to download reports for offline work.
* Security features are crucial, with different roles and access levels for employees.
* Tracking functionality is desired to monitor user activities and changes in the system, particularly for reservations.

### 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?*

– User Registration and Account Management:

* Task: Users (customers, Liam, Ian, secretary) should be able to register accounts.
* Measure: Successful account creation with required user details and appropriate access levels.

– Online Classes and Practice Tests:

* Task: Provide online classes and practice tests for customers.
* Measure: Users can access and complete online classes and practice tests. Progress tracking for completed tests.

– On-the-Road Training Scheduling:

* Task: Schedule on-the-road training sessions based on user preferences.
* Measure: System successfully schedules driving lessons, considering available time slots, drivers, and cars.

– Data Accessibility:

* Task: Enable online and offline access to data for Liam and other authorized users.
* Measure: Data access from any device, with the ability to download reports for offline use.

– Tracking System Activities:

* Task: Implement tracking functionalities to monitor changes in the system.
* Measure: Activity logs accurately record user actions, including reservations, modifications, and cancellations.

– Compliance with DMV Rules:

* Task: Connect to the DMV for updates on rules, policies, and sample questions.
* Measure: System receives timely notifications and updates from the DMV. Compliance checks are performed.

– Cloud-Based System:

* Task: Implement a cloud-based system to ensure seamless operation.
* Measure: System runs efficiently on the web with minimal technical issues. Backup and security are managed effectively.

– Interface Design:

* Task: Design an interface with specified features, including test progress tracking and driver notes.
* Measure: Interface matches the provided sketch. Test progress and driver notes are displayed accurately.

– Future Feature Considerations:

* Task: Design the system to allow for future feature additions, such as customizing driving lesson packages.
* Measure: System architecture is flexible enough to accommodate future modifications and additions.

## 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 system should be accessible over the internet, preferably through cloud-based infrastructure.
* The system should be responsive and provide quick access to information and functionality.
* Users should not experience significant delays when accessing the system or performing tasks such as scheduling appointments or taking online classes.
* The system should be updated regularly to address bugs, security vulnerabilities, and incorporate new features or enhancements based on customer feedback and evolving business needs.

#### 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 system should run on multiple platforms, including Windows, Unix/Linux, and macOS.
* Since the system is accessed over the internet, users can interact with it using web browsers on various platforms without the need for platform-specific installations or configurations.
* The backend of the DriverPass system requires a database system to store and manage user data, scheduling information, training materials, and other relevant information.
* A web server is necessary to host the web-based application and serve content to users over the internet.
* Given the sensitive nature of user data and financial transactions, the backend infrastructure should incorporate security tools and protocols to protect against unauthorized access, data breaches, and other security threats.

#### 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 should have a unique username and password combination to access the system.
* User authentication can be implemented using standard protocols such as username/password authentication, OAuth, or OpenID Connect.
* The case sensitivity of input during the user authentication process to ensure users understand how to correctly enter their credentials.
* The system can enforce strong password policies, such as minimum length requirements, inclusion of alphanumeric characters, and regular password expiration to enhance security.
* DriverPass could consider implementing multi-factor authentication (MFA) for an added layer of security.
* The system should be equipped with monitoring and alerting mechanisms to detect and notify administrators of any problems or issues that may arise.
* Additionally, DriverPass may establish incident response procedures and escalation protocols to ensure timely resolution of issues and minimize service disruptions for users.

#### 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?*

* DriverPass can develop an intuitive user management interface within the application, which could then be accessible to authorized administrators.
* This interface should provide functionalities for adding new users, removing existing users, and modifying user attributes such as roles, permissions, and contact information.
* User-related data, such as usernames, passwords, roles, and permissions, can be stored in configuration files or a database separate from the application code.
* DriverPass should follow best practices for version control, dependency management, and modular design to facilitate a system of updating and expanding.
* By using version control systems like Git and package managers like npm or Maven, DriverPass can manage code changes and updates efficiently while maintaining backward compatibility with existing functionalities.
* The IT admin may need access to backend infrastructure components, such as servers, databases, and networking devices, to ensure smooth operation and address infrastructure-related challenges effectively.

#### 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, the user needs to provide their username and password.
* DriverPass can enhance security by implementing strong password policies, such as minimum length requirements, complexity rules (including a mix of alphanumeric characters, symbols, and uppercase/lowercase letters), and regular password expiration.
* DriverPass should use secure communication protocols such as HTTPS to encrypt data exchanged between the client and the server.
* DriverPass can implement brute force protection mechanisms to mitigate the risk of unauthorized access due to repeated login attempts.
* If a user forgets their password, DriverPass should provide a password recovery mechanism to help them regain access to their account securely.

### 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 provide online classes for driver training.
* The system shall offer practice tests for users preparing for driving exams.
* The system shall allow users to make appointments for driving lessons through their accounts.
* The system shall support multiple packages for driving training, including options for different durations and in-person instruction.
* The system shall enable users to customize their driving training packages.
* The system shall allow users to cancel or modify their scheduled driving appointments online.
* The system shall track and manage the availability of driving instructors and vehicles.
* The system shall match users with available driving instructors based on their preferences and scheduling constraints.
* The system shall provide a secure login mechanism for users to access their accounts by utilizing strong password policies and multi-factor authentication.
* The system shall encrypt data transmitted between the client and the server using secure communication protocols.
* The system shall provide a password recovery mechanism for users who forget their passwords, including verification procedures to confirm their identity.
* The system shall notify administrators of any security incidents or unauthorized access attempts.
* The system shall comply with regulatory requirements and standards for data privacy and security in the driver training industry.
* The system shall integrate with external systems or databases to stay updated on driving regulations and requirements set by the Department of Motor Vehicles (DMV).
* The system shall provide reporting and auditing capabilities to track user activity, reservations, and system usage.
* The system shall be accessible from different platforms and devices, including desktop computers, laptops, and mobile devices.
* The system shall maintain data integrity and consistency across all user interactions and transactions.
* The system shall be scalable and able to accommodate growth in the number of users and data volume over time.
* The system shall provide a user-friendly interface with intuitive navigation and accessibility features for users with disabilities.
* The system shall undergo regular maintenance and updates to address bugs, security vulnerabilities, and performance issues.

### 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.)?*

* Administrators: Manage user accounts, configure system settings, view reports on system usage and activity, monitor reservations and instructor availability.
* Instructors: View their assigned driving appointments, provide feedback on driving lessons, update availability, and access training materials or resources.
* Secretaries: Manage appointments, answer customer inquiries, update customer information, and assist with administrative tasks such as generating reports or processing payments.
* Customers: Schedule, modify, or cancel driving appointments, access online classes and practice tests, view progress on completed tests, update personal information, and make payments for services.
* The interface should be accessible via both desktop browsers and mobile devices to accommodate users accessing the system from different platforms.
* Administrators, instructors, and secretaries may primarily interact with the interface through desktop browsers, as they often require access to advanced features and functionalities.
* Customers, on the other hand, may prefer to interact with the interface through mobile devices for convenience and flexibility, especially when scheduling or modifying driving appointments on the go.
* For mobile users, the interface should be optimized for smaller screens and touch-based interactions, with responsive design elements that adapt to various screen sizes and orientations.
* The interface should be responsive and user-friendly, with intuitive navigation and accessibility features to accommodate users with different levels of technical proficiency.
* Features such as dropdown menus, buttons, and input fields should be easily accessible and clearly labeled to facilitate seamless interaction with the system.

### 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?*

* User Interface Design: The design did not delve into specific details of the user interface layout, visual elements, or interactions.
* Assumption: Users have basic familiarity with common user interface patterns and conventions, allowing them to navigate and interact with the system intuitively.
* Accessibility Features: The design did not explicitly mention accessibility features to accommodate users with disabilities, such as screen readers, keyboard navigation, or alternative text for images.
* Assumption: Users with disabilities have access to assistive technologies and require features that comply with accessibility standards to ensure equal access to system functionalities.
* Performance Optimization: The design did not discuss performance optimization techniques to ensure fast loading times and smooth interaction, especially for users with slower internet connections or older devices.
* Assumption: Users may access the system from various devices and network conditions, requiring optimization strategies such as lazy loading, caching, and image compression.
* Error Handling and Feedback: The design did not detail error handling mechanisms or user feedback features to guide users in case of input errors, system failures, or unexpected behavior.
* Assumption: Users may encounter errors or issues while using the system, necessitating clear error messages, tooltips, and feedback prompts to assist them in troubleshooting.
* Security: The design briefly mentioned security measures such as password policies and encryption but did not delve into detailed security features such as role-based access controls, audit trails, or data encryption at rest.
* Assumption: Users expect their data to be secure and protected against unauthorized access, requiring robust security measures to safeguard sensitive information.

### 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: The system design outlined above may not fully capture all the nuanced requirements and complexities of the DriverPass business model and operation. Certain functionalities or edge cases may not have been adequately addressed, leading to potential gaps in system functionality or user experience.
* Resource Limitations: Adequate resources for user research, requirement gathering, and stakeholder engagement may be lacking, affecting the accuracy and completeness of the system design. The availability of infrastructure resources, such as servers, databases, and networking equipment, may impose constraints on system performance, reliability, and scalability.
* Time Constraints: Time limitations may impact the development timeline and project delivery schedule, potentially leading to rushed implementations or compromises in quality. Dependencies on external factors, such as regulatory approvals, vendor deliverables, or stakeholder availability, may introduce delays and disrupt project timelines.
* Budget Constraints: Budgetary constraints may limit the allocation of resources for system development, testing, deployment, and ongoing maintenance. Cost considerations may force trade-offs between system features, scalability, security, and user experience, potentially compromising the overall value proposition of the system.
* Technology Constraints: Technological limitations, such as compatibility issues, platform dependencies, or legacy system integrations, may hinder the adoption of cutting-edge technologies or best practices. Availability of technology expertise and specialized skills may influence the selection of development tools, frameworks, and platforms, impacting system capabilities and performance.

### 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.*

