# 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 system design is to create a comprehensive, accessible platform for DriverPass, which will enhance the driver training experience and increase the success rate of individuals taking their driving tests. DriverPass requires a system that supports various training components, including scheduling, practice exams, and learning resources. The system should be accessible across devices and offer offline data access while ensuring data consistency and security.

**To achieve this purpose, the system will:**

* Enable clients to book driving practice sessions and access online training resources.
* Provide practice exams that simulate real DMV test conditions.
* Support tiered user access, where different levels of access allow administrative tasks, training scheduling, and resource management.
* Offer flexible configuration for training packages to support various learning needs.

### 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** aims to streamline the process for users preparing for their driving test, addressing the common challenge of limited access to quality practice sessions and resources.
* The problem is that clients often fail the driving test due to a lack of adequate preparation and quality practice resources, which DriverPass seeks to improve.

The system must include components for:

* Scheduling driving sessions with instructors.
* Accessing and taking practice exams.
* Managing educational resources for learning.
* Handling user and administrative access to maintain and update sessions, resources, and user accounts.

### 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?*

**System Capabilities**  
The system should be capable of:

* Enabling users to schedule and manage practice sessions with driving instructors.
* Providing users with access to practice exams that closely mimic the format and content of actual driving tests.
* Storing and managing tutorial materials that users can access to enhance their preparation for the driving test.
* Tracking user progress and offering analytics to assess performance on practice exams.

**Key Measurable Objectives**  
To ensure the system meets its goals, the following key measurable tasks must be achieved:

* Streamlining the session booking process to ensure users can complete it with a minimal number of steps.
* Delivering immediate feedback on practice exam results once completed.
* Ensuring 99% system uptime to guarantee uninterrupted access for users.

## 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 must support both web-based and mobile application interfaces for seamless user interaction across devices.
* The system should ensure a response time of no more than 2 seconds for all user interactions, providing a fast and efficient experience.
* Regular system updates should be conducted on a weekly basis to maintain performance, security, and feature enhancements.

#### 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 must be compatible with multiple platforms to ensure broad accessibility. It should support the following operating systems:
  + **Windows** (for desktop and laptop users)
  + **macOS** (for desktop and laptop users)
  + **Linux** (for users with a preference for open-source environments)
  + **Android** and **iOS** (for mobile users, ensuring compatibility with smartphones and tablets)
* The back end will require a **relational database management system (RDBMS)** such as **MySQL** or **PostgreSQL** for storing user data, session information, practice exams, and educational resources. Alternatively, a **NoSQL database** like **MongoDB** may be considered based on scalability and flexibility needs.
* The system's back end must also include integration with **RESTful APIs** for communication between the front end and database. The APIs will handle functions such as user authentication, session management, and retrieving practice exam data.
* For the development and deployment of the system, cloud-based services like **Amazon Web Services (AWS)** or **Microsoft Azure** may be utilized for hosting, ensuring scalability, data storage, and security.

#### 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?*

* **User Identification**: Users will be distinguished by unique identifiers, such as **usernames** and **email addresses**. Each user will be required to create a unique username and provide a valid email address during registration. These credentials will be used to track user activity and preferences within the system. For security and user differentiation, **user roles** (e.g., learner, instructor, administrator) will be assigned at registration or during account setup, and these roles will determine access permissions.
* **Case Sensitivity**: The system will enforce **case-sensitive** input for usernames, passwords, and email addresses. This ensures that credentials are entered correctly, enhancing security by preventing unauthorized access through case mismatches. However, to ensure ease of use, email addresses will be **case-insensitive** when validating registration and login credentials.
* **Error Handling and Admin Alerts**: The system will inform administrators in the event of critical issues, including but not limited to:
* **Failed login attempts**: If a user exceeds the maximum number of failed login attempts (e.g., five attempts), an alert will be sent to the system administrator for review.
* **Suspicious activities**: The system will monitor for patterns indicative of potential security threats (such as unauthorized attempts to access restricted areas) and notify administrators accordingly.
* **System errors**: If the system encounters database failures, session issues, or data inconsistencies, administrators will be promptly informed through real-time notifications to ensure timely resolution.

#### 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?*

* **User Management**: The system is designed to allow for seamless user management without the need to modify the underlying code. Administrators will have access to a **user management interface** where they can **add, remove, or modify** user accounts, roles, and permissions. Changes to user data, including profile information, access levels, and session history, can be made directly through the interface. This flexibility ensures that the system can evolve with changing user needs without requiring code changes or system downtime.
* **Platform Updates**: The system will be developed to be **platform-agnostic**, supporting multiple operating systems such as Windows, macOS, Android, and iOS. The backend architecture will be modular, allowing for easy integration with future platform updates or new versions of operating systems. Regular **system testing and compatibility checks** will be implemented to ensure that updates to the platform do not disrupt the functionality of the system. Additionally, any required updates will be incorporated through **incremental software patches** to minimize disruption.
* **IT Administrator Access**: The IT administrator will have **full administrative access** to the system for tasks such as user management, resource configuration, and security monitoring. Admins will have the ability to:
* **Add/modify user roles** and **adjust permissions** to ensure that users have the appropriate access levels.
* **Monitor system performance** and **perform maintenance tasks** such as updating system resources or resolving any technical issues.
* **Access logs** and receive **alerts** related to user activities or system errors to ensure smooth operation.

Additionally, IT admins will have the ability to **update system configurations**, such as adjusting settings for new platform requirements, security protocols, and software version upgrades.

#### 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?*

* **User Login Requirements**: To ensure that only authorized individuals can access the system, the login process will require **multi-factor authentication (MFA)**. Users will need to provide a valid **username** and **password**, and upon successful validation, they will be prompted to enter a **second factor** (e.g., a one-time password sent via email or an authentication app). This additional layer of security significantly reduces the likelihood of unauthorized access.
* **Securing Data Exchange**: The connection between the client (user interface) and the server will be secured using **SSL/TLS encryption**. This ensures that all data exchanged between the client and the server is **encrypted**, protecting sensitive information such as user credentials, personal data, and session information. The system will enforce **HTTPS** (secure HTTP) to prevent man-in-the-middle attacks and ensure that data integrity is maintained.
* **Brute Force Attack Protection**: In the event of a **brute force attack**, where an attacker attempts to gain access by systematically trying different password combinations, the system will implement the following safeguards:
  + After **five failed login attempts** within a short time frame, the system will temporarily **lock the account** for a specified duration (e.g., 30 minutes) to prevent further attempts.
  + Administrators will receive an **alert** when multiple failed login attempts are detected for a specific account, allowing them to investigate and take further action if necessary.
  + If an account is under continuous attack, the system will offer an option for the administrator to **permanently lock** the account or require additional authentication steps to regain access.
* **Password Recovery**: If a user forgets their password, they will be able to initiate a **password reset process**. The user will be asked to provide their registered **email address**. Upon submission, the system will send a **password reset link** to the registered email. The user will then be able to choose a new password, which will be encrypted and securely stored in the database. To further enhance security, the password reset link will be **time-sensitive**, expiring within a set period (e.g., 15 minutes). If the user does not reset the password within that time frame, they will need to initiate the process again.

### 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 allow users to create and manage their accounts by providing basic personal information (e.g., name, email, and password).
* The system shall authenticate users by validating their login credentials, ensuring secure access to their personal accounts.
* The system shall enable users to book and schedule driving practice sessions with certified instructors based on availability.
* The system shall allow users to access and take practice exams with questions and formats that closely resemble the actual driving test.
* The system shall track user progress on practice tests and display detailed analytics on performance, including areas of strength and improvement.
* The system shall provide users with access to tutorial content, which can be viewed and studied at their own pace to prepare for the driving exam.
* The system shall send users reminders and notifications about upcoming scheduled practice sessions, test results, and tutorial updates.
* The system shall allow users to reset their passwords securely if they forget their login credentials by sending a password reset link to their registered email.
* The system shall maintain an admin interface that allows system administrators to manage users, instructors, and content (e.g., practice exams, tutorial materials).
* The system shall allow administrators to monitor system performance and track usage analytics to ensure continuous improvement of the service.
* The system shall support multiple platforms, including web and mobile applications, to ensure accessibility for users on various devices.
* The system shall allow users to provide feedback after each session, exam, or tutorial to improve the quality of the system’s content and services.
* The system shall ensure that user data is securely stored, with sensitive information encrypted both in transit and at rest.

### 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 user interface (UI) of the DriverPass system must be intuitive, user-friendly, and accessible across multiple devices to ensure an optimal experience for all user groups. Below are the key interface requirements:

**Needs of the Interface**:

* + Consistency: The design should offer a consistent experience across all platforms (mobile and web), with similar navigation and functionality.
  + Responsiveness: The interface must adjust seamlessly to various screen sizes and orientations (e.g., smartphones, tablets, desktops) to provide users with a flexible experience.
  + Accessibility: The interface should be designed to accommodate users with varying levels of technical proficiency, ensuring ease of use for everyone.
  + Efficiency: The layout should minimize unnecessary steps and offer easy-to-understand options, ensuring users can perform tasks quickly and without confusion.
  + Aesthetic Appeal: The design should be modern, professional, and engaging to ensure users enjoy using the system.

**Different Users for the Interface**:

* + Learners (Primary Users**)**: Individuals seeking to prepare for their driving test by booking practice sessions, taking exams, and reviewing learning materials.
  + Instructors: Certified driving instructors who need to manage their schedules and interact with learners.
  + Administrators: System administrators responsible for managing users, instructors, resources, and monitoring system performance.

**User Needs and Actions**:

* + **Learners**:
    - Access to Dashboard: View upcoming sessions, track progress, and monitor exam results.
    - Booking and Scheduling: Book driving practice sessions with available instructors and reschedule if needed.
    - Practice Exams: Take practice tests and receive immediate feedback on performance.
    - Educational Content: Access a library of tutorials and educational resources to prepare for the test.
    - Feedback and Notifications: Provide feedback on practice sessions and exams and receive reminders for upcoming events.
  + **Instructors**:
    - Manage Schedule: View and update their availability for practice sessions.
    - Review Learner Progress: Access analytics of learners' practice test results and session performance.
    - Notifications: Receive updates about learner bookings and cancellations.
  + **Administrators**:
    - User Management: Add, remove, or update learner and instructor accounts.
    - Content Management: Upload, update, or remove practice exams and tutorial materials.
    - System Monitoring: Track system usage, performance metrics, and user feedback.
    - Security and Access Control: Set permissions and roles for users and instructors.

**User Interaction with the Interface**:

* + Mobile Interaction: Users will interact through a mobile application using touch gestures (tap, swipe, pinch) to navigate through screens, select options, and input data. This platform should be optimized for mobile screens, ensuring quick load times and responsive design.
  + Browser Interaction: On desktop or laptop computers, users will interact via a web browser using a mouse and keyboard. The web interface should support common browsers (Chrome, Firefox, Safari, Edge) and allow for a seamless experience.
  + Accessibility Features: Ensure that text is readable (consider font size, color contrast) and include features such as voice commands, screen readers, and keyboard navigation support for users with disabilities.

### 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?*

In the design of the DriverPass system, several aspects were not specifically addressed. The following assumptions are made regarding the users and the technology they have:

1. **User Technology Assumptions**:

**Internet Access**: It is assumed that all users (learners, instructors, and administrators) have access to the internet with sufficient bandwidth for smooth operation of web and mobile applications.

**Mobile Devices**: It is assumed that most learners will use smartphones or tablets to interact with the system, and that these devices will run on either Android or iOS platforms. The system is designed to be compatible with the latest versions of these operating systems.

**Browser Compatibility**: It is assumed that users accessing the system via desktop or laptop computers will use modern browsers (such as Chrome, Firefox, Safari, or Edge). Older browsers or devices may not provide optimal performance.

**Technology Literacy**: It is assumed that users (especially learners and instructors) have basic technological literacy to interact with the system, including knowledge of how to use mobile applications and navigate through web interfaces.

1. **System Functionality Assumptions**:

**System Performance**: It is assumed that the system will maintain sufficient performance standards, including response times under 2 seconds and minimal downtime. While the system will aim for a 99% uptime, any unexpected downtime due to external factors (e.g., server outages, maintenance) is not accounted for in this assumption.

**Data Storage**: The system assumes that users' data (such as personal information, bookings, test results, etc.) will be stored securely in a central database, and that this database will be able to handle the expected volume of users and transactions.

**Payment Integration**: If applicable, it is assumed that the system will integrate with a third-party payment processing service for booking and session payments. Security standards for payments are assumed to follow industry norms (e.g., PCI-DSS compliance).

**User Role Management**: It is assumed that the role-based access control (RBAC) system will work as intended, with administrators having full control over user accounts and content, and instructors having limited access to relevant learner data.

**User Input Handling**: It is assumed that input from users will generally be correct and consistent with the form fields (e.g., entering dates correctly, providing accurate contact information). The system will have some validation, but excessive or unusual user input may not be handled in every case.

1. **External Assumptions**:

**Compliance and Legal Regulations**: The system assumes that it complies with relevant data privacy regulations, such as GDPR or CCPA, based on the geographical location of its users. If operating in regions with specific legal requirements for data protection, it is assumed that the system will adapt as necessary.

**Platform Updates**: It is assumed that future updates to mobile operating systems (iOS and Android) or web browsers will not break the functionality of the system. The design is made to be adaptable, but continuous testing and updates may be necessary.

**Instructor Availability**: It is assumed that instructors will provide accurate availability data and adhere to the booking schedule. The system does not address potential issues like instructor cancellations, which will be handled manually by the instructor or through communication with learners.

### 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 design is robust and user-centric, but several limitations may impact its functionality and implementation:

1. **System Design Limitations**:

**Scalability**: While the system is designed for moderate scalability, sudden and unexpected surges in user traffic may result in slower performance or downtime if server capacity is exceeded.

**Cross-Platform Consistency**: Although efforts have been made to ensure uniform functionality across web and mobile platforms, slight variations in user experience due to device-specific constraints (e.g., screen size, performance) may persist.

**Data Syncing**: Real-time synchronization of data between multiple platforms and the back-end server may face delays during peak usage periods.

**Offline Access**: The system does not support offline functionality, which may inconvenience users in areas with unstable internet connectivity.

**Automation Limitations**: Certain processes, such as handling complex user disputes or providing detailed analytics beyond predefined metrics, require manual intervention or advanced machine learning, which is outside the current system scope.

1. **Resource Limitations**:

**Budget Constraints**: Developing, deploying, and maintaining a system of this scale requires significant financial resources. Constraints in budget may limit the ability to incorporate advanced features like AI-driven progress tracking or predictive analytics.

**Human Resources**: Availability of skilled developers, UX/UI designers, and testers may impact the timeline for achieving all design goals. Limited personnel may delay updates and maintenance.

**Infrastructure Costs**: The cost of robust infrastructure (e.g., high-performance servers, cloud services) to ensure 99% uptime may strain the budget, particularly during the initial deployment phase.

1. **Time Constraints**:

**Development Timeline**: The system may require more time for full development and testing than initially allocated, especially for complex functionalities like real-time progress tracking and secure payment integration.

**Frequent Updates**: Weekly system updates are planned, but testing and deployment of these updates within a tight schedule may introduce bugs or regressions.

1. **Technology Limitations**:

**Integration Challenges**: Seamlessly integrating with third-party tools, such as payment gateways or learning management systems, may pose compatibility issues or additional costs.

**Security Risks**: While measures are in place to secure user data and communication, evolving cybersecurity threats may require ongoing investments in security technologies.

**Platform Dependencies**: The system's reliance on third-party platforms (e.g., Android, iOS) means it is subject to constraints and updates from those platforms, potentially requiring frequent adjustments.

1. **User Limitations**:

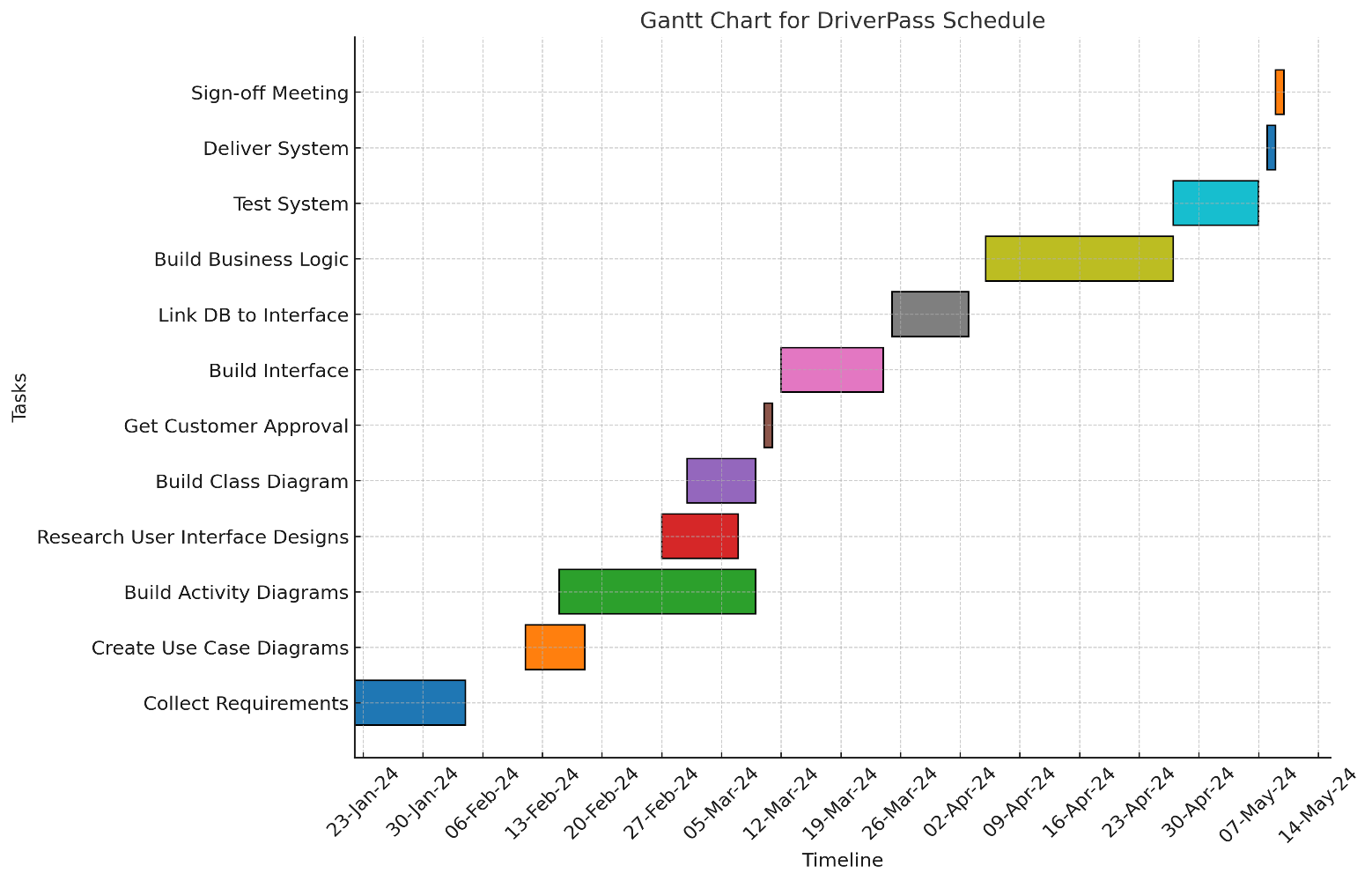
**Technology Proficiency**: Some users may lack the technological skills required to navigate the system effectively, necessitating additional resources for user support and training.

**Device Compatibility**: Older devices or unsupported browsers may not deliver the intended user experience, potentially alienating a segment of the user base.

These limitations highlight the challenges in implementing a comprehensive and efficient system while operating within practical constraints. Overcoming these limitations will require careful planning, prioritization, and iterative improvements based on user feedback and evolving requirements.

### 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.*

**