# CS 255 Business Requirements Document Template

Kevin Tippin

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 Business Requirements Document (BRD) is to outline the key needs and specifications for the development of the DriverPass system.
* This platform aims to provide driving students with online practice exams, scheduling for on-road training, progress tracking, and communication tools.
* The goal is to enhance test preparation, reduce failure rates, and streamline administrative processes for instructors and administrators.

### 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 automate and secure driver credential management by providing a digital system for requesting, issuing, and tracking access passes. The main problem is manual processes that cause delays, errors, and security risks.
* To fix this, the system needs components like user management for driver data, pass management for issuing and renewing passes, secure authentication, an approval workflow, a database for records, notification alerts, reporting tools, and integration with other security systems. A user-friendly interface and strong security measures are also essential. Overall, it streamlines pass handling, improves security, and reduces administrative effort.

### 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?*

* Offer online practice exams: Students should be able to access practice tests that mimic real driving tests, review their answers, and track their progress. The system should allow students to take multiple exams and see their scores.
* Schedule and manage on-the-road training: Students need to be able to book, reschedule, or cancel driving lessons with qualified instructors through the platform. The system should handle instructor availability and booking conflicts.
* User account management: The system must allow different types of users—students, instructors, and administrators—to create secure accounts with appropriate permissions.
* Track progress and generate reports: Students and instructors should be able to view their completed tests, training hours, and overall progress. The system should produce reports to help monitor learner development.
* Ensure accessibility and ease of use: The platform should be simple to navigate on various devices, including smartphones and tablets, so users can access it anytime, anywhere.
* These objectives are designed to be measurable so we can assess whether the system meets DriverPass’s expectations. For example, aiming for at least 80% of students to complete practice exams or ensuring that 90% of scheduled training sessions are completed without cancellations.

## 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 designed to operate seamlessly across multiple environments, including web-based platforms accessible through standard browsers on desktops, laptops, and mobile devices, as well as dedicated mobile applications for iOS and Android to support on-the-go access. An optional desktop application could also be developed for organizations requiring more dedicated functionality.
* To ensure a positive user experience, the system should aim for response times of under 2 seconds for most interactions, capable of handling multiple concurrent users without significant delays.
* Regular updates are essential, with security patches and compliance fixes released monthly or quarterly, feature enhancements rolled out every 1-3 months based on user feedback, and data refreshes occurring either in real time or on scheduled daily intervals to maintain accurate driver and license information.
* Overall, the system should prioritize performance, accessibility, and security through consistent, timely updates and multi-platform support.

#### 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 is designed to run on platforms such as Windows and Unix/Linux, depending on the organization's infrastructure and deployment preferences.
* To support the application effectively, a backend database is generally required to store user information, credentials, access logs, and other relevant data; common choices include relational databases like MySQL, PostgreSQL, or SQL Server, or NoSQL options such as MongoDB.
* Supporting tools like web servers (e.g., Apache or Nginx) and application runtimes (such as Java, .NET, or Node.js) may be necessary to facilitate web interfaces and application processing. Overall, a robust backend infrastructure with an appropriate database and supporting tools is essential for the reliable operation of the DriverPass system across supported platforms.

#### 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 DriverPass system identifies users using unique identifiers like usernames or driver IDs stored in the database.
* Whether the input is case-sensitive depends on how the system is set up; many systems make usernames and passwords case-sensitive to improve security, but it can be adjusted if needed.
* The system should notify the admin when there are issues such as multiple failed login attempts, suspicious activity, database errors, or other problems that could affect security or operation. Quick alerts help the admin respond promptly and keep the system running smoothly.

#### 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?*

* Yes, in most well-designed systems, you can add, remove, or modify users without changing the code, typically through an administrative interface or management tool that interacts with the backend database.
* This allows for easier user management and reduces the need for code updates for routine changes.
* To adapt to platform updates, the system should be built with modular components and follow best practices for software maintenance, ensuring compatibility with new operating system versions or security patches through regular updates and testing.
* The IT admin generally needs access to the administrative dashboard or control panel, which includes permissions to manage users, view logs, configure settings, and perform maintenance tasks. This access should be secured with appropriate authentication and authorization measures to prevent unauthorized access.

#### 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, a user typically needs a valid username and password, which are authenticated against stored credentials in the system’s database.
* To secure the connection and data exchange between the client and server, implementing HTTPS with SSL/TLS encryption is essential; this encrypts all data transmitted, preventing eavesdropping or man-in-the-middle attacks.
* If there is a brute force hacking attempt—such as multiple failed login attempts—the system should lock the account temporarily or require additional verification steps to prevent unauthorized access.
* If a user forgets their password, the system should provide a secure password reset process, usually involving sending a verification code or link to the user’s registered email address, allowing them to create a new password 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 validate user credentials during the login process to ensure authorized access.
* The system shall encrypt data transmitted between the client and server using HTTPS with SSL/TLS protocols.
* The system shall lock a user’s account after a predefined number of consecutive failed login attempts to prevent brute-force attacks.
* The system shall allow users to reset their passwords securely through email verification or other secure methods.
* The system shall enable administrators to add, remove, or modify user accounts without changing the underlying code.
* The system shall provide different user roles with appropriate access permissions, such as admin, driver, and security personnel.
* The system shall log login attempts, account modifications, and security-related events for audit purposes.
* The system shall notify the admin of system errors or suspicious activities that may indicate security issues.
* The system shall provide an intuitive interface accessible via web browsers and mobile devices.
* The system shall verify user identity during login to prevent unauthorized access, potentially using multi-factor authentication.

### 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 for the DriverPass system needs to be user-friendly, secure, and accessible across various devices.
* Different users include system administrators, drivers, and possibly security personnel. The administrator requires access to manage user accounts, view logs, configure system settings, and monitor activity.
* Drivers need a simple way to authenticate, check their status, or update personal information. Security staff might need access to audit logs and system alerts.
* Interaction with the interface can be through a web browser on desktops or laptops, and potentially via mobile devices like smartphones or tablets for convenience. The design should ensure easy navigation, secure login, and clear access to the functions each user role requires.

### 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?*

* The design above does not specifically address details such as user authentication methods (e.g., two-factor authentication), accessibility features for users with disabilities, or how data privacy and security are managed beyond basic access controls.
* It also assumes that users have reliable internet access and compatible devices, such as browsers on desktops or mobile devices. Additionally, there is an assumption that users are familiar with standard web or app interfaces and that the system will be maintained and updated regularly to stay compatible with platform updates.
* These assumptions may not hold true in all environments, so further planning might be needed to accommodate offline access, enhanced security measures, or users with limited technical skills.

### 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?*

* Some limitations in the system design include potential security vulnerabilities if not properly implemented, such as data breaches or unauthorized access.
* The reliance on internet connectivity could also limit usability in areas with poor or no network access. Additionally, the system's scalability might be constrained if the infrastructure isn’t adequately planned, leading to performance issues as the user base grows.
* From a resource perspective, developing and maintaining the system requires time, skilled personnel, and financial investment, which may be limited.
* Budget constraints might restrict the choice of technology or limit the implementation of advanced features like two-factor authentication or extensive logging.
* Technologically, the system’s compatibility depends on the platforms and devices available to users; older devices or outdated browsers could pose challenges. Overall, these limitations highlight the importance of phased development, ongoing maintenance, and realistic planning based on available resources.

### 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.*

