# CS 255 Business Requirements Document Template

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

### The purpose of the project is to design and develop a DriverPass system for a transportation company that will enable them to manage their drivers and ensure compliance with regulatory requirements. The client is the transportation company, and they want the system to be able to track driver information, manage driver certifications and qualifications, schedule driver training, and generate reports to ensure compliance with regulations. Additionally, the system should provide a way for drivers to easily access their own information and manage their certifications and training requirements.

### System Background

### DriverPass wants to create a system that can manage the drivers and vehicles for their transportation business. The problem they want to fix is the lack of an efficient system for managing their operations. They want to streamline their operations by having a centralized system that can handle driver and vehicle management, trip scheduling, customer communication, and reporting.

### The different components needed for this system include a user interface for managing driver and vehicle information, a database to store the information, algorithms for trip scheduling and route optimization, a communication system for interacting with customers, and a reporting system for generating various reports related to the business operations.

### Objectives and Goals

## The DriverPass system should be able to perform the following tasks when completed:

## Allow drivers to create and manage their profiles, including adding and removing vehicles, updating their contact information, and setting their availability.

## Allow fleet managers to create and manage their fleets, including adding and removing drivers, assigning vehicles, and setting driver availability.

## Provide a user-friendly interface for both drivers and fleet managers to use, with clear instructions and minimal learning curve.

## Allow drivers to view and accept available jobs, as well as view their past job history.

## Allow fleet managers to view and manage job assignments, as well as view past job history for their drivers.

## Provide real-time updates and notifications for job assignments and changes.

## Ensure the security and privacy of user data, with proper encryption and access control mechanisms in place.

## Handle a large number of users and transactions simultaneously, with minimal downtime and fast response times.

## Support multiple platforms and devices, including desktop and mobile platforms.

## To achieve these tasks, the following measurable tasks need to be included in the system design:

## Develop a user registration and login system that validates user credentials securely and efficiently.

## Develop a user profile management system that allows users to add, update, and delete their profiles, and allows fleet managers to manage their fleets.

## Develop a job management system that allows drivers to view and accept jobs, and allows fleet managers to manage job assignments.

## Implement real-time updates and notifications for job assignments and changes.

## Develop a robust security and privacy system that ensures the confidentiality and integrity of user data.

## Develop a scalable and fault-tolerant architecture that can handle a large number of users and transactions simultaneously.

## Test and optimize the system for fast response times and minimal downtime.

## Develop a responsive user interface that is intuitive and easy to use.

## Ensure compatibility with multiple platforms and devices, including desktop and mobile platforms.

## Requirements

### Nonfunctional Requirements

Here are some non-functional requirements for the DriverPass system:

* Security: The system should be secure and protect sensitive user data. All communication between the client and server should be encrypted. The system should have measures to prevent unauthorized access and mitigate the risk of attacks like SQL injection or cross-site scripting
* Performance: The system should be fast and responsive to user requests. The system should be able to handle a large number of users and transactions without significant slowdowns.
* Scalability: The system should be scalable to support a growing number of users and data. The system should be able to handle increases in traffic and transaction volume without significant degradation in performance.
* Availability: The system should be available 24/7 with minimal downtime for maintenance or updates. The system should have a failover mechanism to ensure continuous operation in case of hardware or software failures.
* Usability: The system should be easy to use and intuitive for all users. The system should have clear and concise error messages to help users resolve issues. The system should be accessible to users with disabilities.
* Compatibility: The system should be compatible with various platforms and devices, including desktop and mobile operating systems and web browsers.
* Maintainability: The system should be easy to maintain, update, and enhance. The system should have modular architecture, clear code structure, and adequate documentation.
* Regulatory compliance: The system should adhere to applicable regulations, such as data privacy laws, and industry standards, such as PCI DSS (Payment Card Industry Data Security Standard) for credit card transactions.

#### Platform Constraints

The platform requirements for the system would depend on the technology stack chosen for development. However, in general, the system should be designed to be platform-independent and able to run on a variety of operating systems, such as Windows, Linux, and macOS.

The back end of the system would require a database to store user information, driver information, and other data related to the system's functionality. The choice of the database management system would also depend on the technology stack chosen for development, but common options include MySQL, PostgreSQL, Oracle, and Microsoft SQL Server. Additionally, the back end may require other tools such as a web server, application server, or middleware to support the application.

#### Accuracy and Precision

To distinguish between different users, the system can utilize a unique identifier such as a username or email address, along with a password or other form of authentication. The input may be case-sensitive or not, depending on the specific requirements of the system. If the input is case-sensitive, the system should ensure that the user input matches the expected case.

The system should inform the admin of a problem as soon as it is detected. This could include errors in user input, system failures, or security breaches. The admin should be notified in a timely manner to minimize the impact of the problem on the system and its users. Additionally, the system should provide appropriate error messages to users to inform them of any issues with their input or actions. These error messages should be clear and concise, providing enough information to help the user understand the problem and how to resolve it.

#### Adaptability

Yes, it's possible to make changes to the user (add/remove/modify) without changing the code. The system can provide an interface for administrators to manage users, such as adding new users, updating user information, and removing users. This can be done through a user management dashboard or API.

To ensure that the system can adapt to platform updates, it's important to follow best practices in software development, including modular and extensible code design, automated testing, and version control. Any changes to the underlying platform or technology can be addressed through updates to the relevant code modules and configurations.

The IT admin may need different types of access depending on their role and responsibilities. For example, they may need access to the user management dashboard, the ability to modify system configurations, and access to system logs and analytics. The level of access can be controlled through user roles and permissions, which can be managed by a super admin or another authorized user.

#### Security

To log in, the user will need to provide their email and password, which will be authenticated by the server. To secure the connection and data exchange between the client and the server, the system should use HTTPS protocol, which encrypts all data exchanged between the client and the server.

To prevent brute force hacking attempts, the system can implement measures such as rate-limiting login attempts, CAPTCHA verification, and two-factor authentication. If a brute force attack is detected, the system should lock the user's account temporarily and notify the IT admin for investigation.

If the user forgets their password, they should be able to reset it through the system. The system can send a password reset link to the user's email, which will allow them to set a new password. Additionally, the system can implement security questions or two-factor authentication to verify the user's identity before resetting the password.

### Functional Requirements

Functional requirements for the DriverPass system based on the scenario:

### The system shall allow users to create a DriverPass account with a valid email address and password.

### The system shall allow users to log in to their DriverPass account with their email address and password.

### The system shall allow users to update their account information, including their name, email address, and password.

### The system shall allow users to upload and manage their driver's license information, including the license number, expiration date, and a photo of the license.

### The system shall allow users to upload and manage their vehicle registration information, including the make and model of the vehicle, the license plate number, and a photo of the registration.

### The system shall allow users to view and manage their driving history, including the dates and times of their trips and the routes taken.

### The system shall allow users to submit feedback or report issues related to their trips or the DriverPass system.

### The system shall validate user credentials when logging in and prevent unauthorized access to user accounts.

### The system shall use encryption to secure the connection and data exchange between the client and the server to prevent unauthorized access or interception of user data.

### The system shall temporarily suspend or lock a user's account if there is a "brute force" hacking attempt to prevent further attempts and alert the user of the incident.

### The system shall allow users to reset their password if they forget it by sending a password reset link to their registered email address.

### User Interface

### General features and elements that could be included in the user interface of a DriverPass system:

### Login Page: The login page would prompt the user to enter their username and password in order to access their account.

### Dashboard: The dashboard would be the main page that appears after the user logs in. It could include a summary of the user's upcoming driving lessons, progress towards completing their driving package, and links to any other relevant pages.

### Appointment Booking: This page would allow users to schedule and modify their driving lessons. Users could select their preferred date, time, and location for the lesson, as well as the package they have selected.

### Course Materials: This page would provide access to the online class materials, including videos, practice tests, and written course materials. Users could track their progress through the course and access any materials they need to review.

### Account Management: This page would allow users to manage their account information, including their personal details and payment information. Users could also change their password or email address if necessary.

### Reporting and Analytics: This page would provide detailed reports and analytics on user activity and performance, including lesson attendance, test scores, and progress towards completing their driving package.

### Help and Support: This page would provide links to support resources, such as FAQs, troubleshooting guides, and contact information for customer support.

### Overall, the user interface should be intuitive and easy to use, with clear navigation and visual design that reflects the brand identity of DriverPass.

### Assumptions

In the design provided above, there are a few things that were not specifically addressed:

* Integration with existing driver databases: It is assumed that the DriverPass system will be a standalone system, without any integration with existing driver databases. However, if such integration is required, it will need to be designed and implemented separately.
* User authentication: While it is assumed that users will be required to register and log in to use the system, the specific method of user authentication (e.g., username/password, biometric, two-factor authentication) has not been addressed.
* Payment processing: While the design includes a payment system for users to pay for their DriverPass subscription, the specifics of payment processing (e.g., payment gateway, transaction fees) have not been addressed.
* Data privacy and security: While the design includes measures to protect user data (e.g., SSL encryption, secure storage), the specific policies and procedures for ensuring data privacy and security have not been addressed.

As for assumptions about users and technology, the design assumes that users have access to a computer or mobile device with an internet connection and a web browser. It also assumes that users have a basic understanding of how to use a web-based system and are comfortable with online payment processing.

### Limitations

There are several limitations in the system design that I have proposed. Some of the limitations are:

* Accuracy: The accuracy of the facial recognition technology and the GPS system used in the system can affect the accuracy of the system.
* Data privacy and security: The system deals with sensitive personal information and requires robust security measures to protect the data from unauthorized access or cyber-attacks.
* User adoption: The system relies heavily on user adoption and may not be effective if the users are not willing to use it or if they find it cumbersome.
* System scalability: The system may face challenges in scaling up to handle large volumes of data and users, which may require additional infrastructure and resources.

In terms of resources, time, budget, and technology, there may be limitations that could affect the implementation and deployment of the system. For example, the development of the facial recognition technology and the GPS system may require significant resources and time, which may affect the budget and timeline of the project. Additionally, the deployment of the system may require additional hardware and software infrastructure, which may require additional budget and resources.

### Gantt Chart

