# CS 255 Business Requirements Document

## 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 create a system for a client named DriverPass, which aims to address the need for better driver training and develop a scheduling system to provide students better driver training.

The client, Liam, is the owner of DriverPass. DriverPass is a company that seeks to provide driver training services to help individuals prepare for their driving tests at the local Department of Motor Vehicles (DMV).

### 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?*

The client, Liam, is the owner of DriverPass and wants the system to improve driver training. The system should offer online classes and practice tests. Additionally, the system should support students with on-road training, scheduled at their convenience.

The problem that DriverPass aims to fix is the need for improved driver training and appointment scheduling services. The client, Liam, mentioned that there is a trend in the market of high failure rates in driving tests and wanted to take advantage of this void.

DriverPass requires several components to fulfill its functionality. These include accessibility and mobility, an appointment scheduling system, reporting and activity tracking, cloud-based infrastructure, and to development of a user-friendly, visually appealing interface that aligns with the client's requirements.

### 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?*

When the system for DriverPass is completed, it should be able to perform a range of functions and meet specific objectives. These objectives and measurable tasks are:

Appointment Scheduling, create a user-friendly interface for customers to book appointments.

Reporting and Activity Tracking, develop a logging and auditing system to record user activities within the system. Create a reporting feature that generates reports for tracking user progress and system usage.

User Interface, design a user interface that meets client-specified design preferences. Create dashboards that display a user's progress, including tests taken, scores, and driving lesson information. Develop design data entry forms for user registration and appointment scheduling.

Accessibility and Data Mobility, enable online and offline access to data, with offline access limited to viewing and downloading reports.

## 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 specific performance requirements for a system can vary based on the nature of the application and its intended use. If the system is web-based, it needs to be compatible with the newest and most popular web browsers. Response times should be acceptable for different operations within the system. The system should respond to user input and several transactions or operations within a certain number of seconds.

#### 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 choice of platforms and backend tools for a system depends on various factors, including the application's requirements, target audience, and development preferences. Windows is suitable if the target audience primarily uses Windows-based devices. The back end requires a database and tools for monitoring application performance and logging errors. Choose a database management system (DBMS) based on data requirements.

#### 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?*

For user identification use a combination of unique identifiers such as usernames, email addresses, or user IDs to distinguish between different users. Clearly define whether the input is case-sensitive or case-insensitive. The system should inform the admin of certain situations, such as in critical errors, security breaches, performance issues, and downtimes.

#### Adaptability

*Can you make changes to the user (add/remove/modify) without changing the code? How will the system adapt to platform updates? What type of access does the IT admin need?*

Provide a user Interface for admin that is an intuitive and secure interface that allows IT admins to add, remove, or modify user accounts without requiring direct code changes. Design the system using standardized APIs and adhere to industry standards for interoperability. Use version control systems to manage code changes and role-based access control to define different levels of access for IT admins based on their roles and responsibilities.

#### 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?*

Users typically need a combination of user login and credentials, a username/email, or a password to log in. Consider supporting alternative authentication methods like multi-factor authentication (MFA) for added security. Securing the connection and data exchange, use HTTPS to encrypt data exchanged between the client and the server. Obtaining and regularly renewing SSL/TLS certificates to ensure the security of the connection. Handling Brute Force Attempts with account lockouts. Implement account lockout mechanisms after a certain number of failed login attempts. Specify a lockout duration, after which the account is automatically unlocked. Introduce CAPTCHA or challenge-response mechanisms to thwart automated bots attempting brute force attacks.

### 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.”*

* [Insert text]

### 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 should have an intuitive and user-friendly design to facilitate easy navigation and interaction. Ensuring that the interface is accessible to users with diverse needs, including those with disabilities. Users should be able to create accounts, log in securely, and manage their profiles. Users should perform actions and tasks specific to the application's purpose. Many users and administrators, may access the system through a web-based interface using standard web browsers. Consider developing mobile applications for users who prefer to interact with the system through smartphones or tablets. Ensure a responsive design for mobile access.

### 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 did not explicitly detail specific measures related to data privacy, or compliance with regulations. The design did not delve into details regarding secure payment processing and transaction management, DriverPass applications involving transactions or payments. The design assumes that users have reliable internet connectivity, especially in the context of web-based interfaces and mobile applications that rely on server interactions. There is also an assumption that users will engage with the system in a manner consistent with the application's intended purpose.

### 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 design might face challenges in handling rapid scalability, especially if the user base or data volume grows significantly. The design did not explicitly address the unique regulatory compliance requirements of specific industries or regions, such as healthcare, finance, or international markets. Budget constraints and limited availability of skilled personnel, development resources, or IT infrastructure could impact the successful implementation and maintenance of the system.

### 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 calendar with a colorful pattern

Description automatically generated with medium confidence