# CS 255 Business Requirements Document

Matthew Tyson

Southern New Hampshire University

CS 255: System Analysis and Design

Dr. Terry House

June 5, 2023

# CS 255 Business Requirements Document

## System Components and Design

### Purpose

* The purpose of this project is to build a system for DriverPass, a company that provides driver training for customers who want to take their driving test at their local Department of Motor Vehicles (DMV).
* The client, Liam, wants the system to handle online classes, practice tests, and on-the-road training.
* The system should help him access data online and offline, and he wants to be able to download reports and information to work on using Excel. Additionally, Liam wants the system to have different employee roles and rights, be secure, and have tracking capabilities. The system should allow customers to make appointments, cancel, and modify them online, as well as in person. There are different packages that customers can choose from, each with a varying number of hours in a car with a trainer and access to an online class with content and practice tests. The system should be flexible to accommodate future changes in the packages offered by DriverPass.

### System Background

* DriverPass wants the system to create an automated process for driver background checks and qualification management. The problem they want to fix is the inefficiency and errors that can occur when manually processing driver qualifications and background checks.
  + Database Management: A system to manage the driver information and qualification data.
  + Automated Background Check: An automated background check process that can collect and analyze relevant information about drivers such as their driving record, criminal history, and drug tests.
  + Qualification Management: A system to manage driver qualification documents such as licenses, medical certificates, and training records.
  + Notifications: A system to notify drivers and managers of expiring qualifications and upcoming renewals.
  + Integration with HR Systems: The system should integrate with HR systems to manage employee information, compliance, and compensation.
  + Reporting and Analytics: A system to generate reports and analytics on driver qualification and compliance data to improve decision-making and risk management.

### Objectives and Goals

* When completed, the DriverPass system should be able to provide a comprehensive solution for managing driver qualifications and compliance requirements for fleet owners and operators. It should be able to automate the entire process of managing driver documentation and qualification requirements, thereby reducing administrative workload and ensuring compliance with regulatory requirements.
  + Store and manage driver qualification documents - The system should be able to store and manage all driver qualification documents, including driver licenses, medical certifications, training certificates, and other required documents.
  + Monitor document expiration dates - The system should be able to monitor the expiration dates of driver qualification documents and notify the appropriate personnel when documents are about to expire.
  + Automate the renewal process - The system should be able to automate the renewal process for driver qualifications, including scheduling medical exams, training sessions, and other required activities.
  + Ensure compliance with regulatory requirements - The system should be able to ensure compliance with all regulatory requirements related to driver qualifications and documentation.
  + Generate reports - The system should be able to generate reports on driver qualification status, compliance, and other relevant metrics to help fleet owners and operators manage their operations more efficiently.

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

* The DriverPass system needs to be a web-based application that can be accessed by both drivers and fleet managers. The system should be responsive, user-friendly, and capable of handling a large volume of data. As for the speed of the system, it should be fast enough to allow users to perform tasks without experiencing significant delays or downtime.
* In terms of updates, the system should be updated regularly to ensure that it is functioning correctly and that any bugs or vulnerabilities are addressed promptly. The frequency of updates will depend on the specific needs of the system, but in general, updates should be done at least once a month or as needed to maintain optimal performance and security.

#### Platform Constraints

* The system will require a back-end infrastructure that includes a database to store user information and transaction data, as well as a web server to handle requests from clients. Windows, Unix and Linux are all viable options to be used for a backend platform. The front-end components of the system, such as the web interface and mobile application, will likely require support for multiple platforms, including iOS and Android for the mobile app, and commonly used web browsers for the web interface.
* As for the specific tools and platforms needed, that will depend on the programming languages and frameworks used in the system design. For example, the back-end infrastructure could be built on a combination of programming languages such as Python, Java, or Node.js, with a database system like MySQL, PostgreSQL, or MongoDB. The web interface could be built with HTML, CSS, and JavaScript, while the mobile application could be developed using tools like React Native or Flutter.

#### Accuracy and Precision

* The distinction between different users can be achieved through user authentication, where each user is required to provide a unique username and password combination. The system can also assign different levels of access privileges to different user roles, such as regular users and administrators, to control what actions they are authorized to perform.
* Whether or not the input is case-sensitive depends on the specific requirements of the system. Generally, it is a good practice to treat input as case-insensitive to avoid confusion and errors.
* The system should inform the admin of a problem as soon as it is detected. For example, if there is an error in the database connection, the system should notify the admin immediately. Additionally, the system can be designed to send regular reports to the admin about the overall performance of the system and any issues that have been detected.

#### Adaptability

* The system should allow for easy changes to user accounts without requiring any changes to the underlying code. This can be achieved by implementing a user management system that allows for user accounts to be created, modified, and deleted through a graphical user interface or API.
* To ensure the system can adapt to platform updates, it's essential to follow best practices for software development and infrastructure management. This includes regularly updating the system's dependencies, monitoring for any breaking changes or security vulnerabilities, and conducting rigorous testing to ensure the system continues to function as expected.
* The IT admin should have full access to the system, including the ability to modify system settings, access logs, and manage user accounts. However, it's essential to implement proper access controls and security measures to prevent unauthorized access to sensitive information and system resources.

#### Security

* To log in, the user will need to enter their username and password. The system can secure the connection by using SSL/TLS encryption to encrypt the data exchange between the client and the server.
* To prevent "brute force" hacking attempts, the system can implement security measures such as account lockout policies or captcha verification after a certain number of login attempts.
* If a user forgets their password, the system should provide a password reset option that sends a password reset link to the user's registered email address. The link should be time-limited and expire after a certain amount of time to prevent unauthorized access.

### Functional Requirements

* The system shall allow users to create an account with a unique username and password.
* The system shall allow users to log in using their username and password.
* The system shall validate user credentials when logging in.
* The system shall provide a dashboard for users to view their driving records and scores.
* The system shall allow users to take driving courses and assessments to improve their scores.
* The system shall allow users to schedule in-person driving assessments.
* The system shall allow users to dispute driving records or assessments.
* The system shall allow administrators to manage user accounts and records.
* The system shall encrypt user data to ensure security during transmission.
* The system shall temporarily lock user accounts after a specified number of failed login attempts.
* The system shall allow users to reset their passwords via email verification.
* The system shall provide error messages and notifications to users and administrators as appropriate.
* The system shall generate reports on user activity and system performance as needed.

### User Interface

* The system shall provide an interface for drivers to log in, view their scores, and see feedback.
* The system shall provide an interface for managers to view driver scores, assign training modules, and see reports.
* The system shall provide an interface for IT admins to add and remove users, and manage system settings.
* The driver interface shall be accessible from a mobile device or a desktop browser.
* The manager and IT admin interfaces shall be accessible from a desktop browser.
* The driver interface shall be intuitive and easy to navigate.
* The manager interface shall provide easy access to reports and be customizable.
* The IT admin interface shall provide easy access to user management tools and system settings.

### Assumptions

* The specific implementation details of the database, the exact authentication mechanism to be used, the specific user roles and permissions, and the exact design and layout of the user interface were not detailed.
* The design assumes that users have a basic level of technical competency and are familiar with standard web application interfaces. It also assumes that the users have access to a device with an internet connection, a web browser, and can complete the necessary steps to register and log in to the system. Additionally, the design assumes that the necessary hardware and software infrastructure is in place to support the system, such as servers, network equipment, and databases.

### Limitations

* The system may not be able to handle a large number of users, which could result in slower response times or even system crashes.
* The security of the system may be compromised if there are vulnerabilities in the software or hardware used in the system, or if users are not following secure login and password practices.
* The system may not be able to integrate with all existing hardware and software systems that the client currently uses, which could result in additional costs or limited functionality.
* The system design assumes that the users have a basic level of technical expertise and access to compatible devices and internet connectivity.
* In addition to these limitations, there may be limitations in terms of resources, time, budget, or technology that could impact the system design. For example, limitations in resources could affect the development timeline or the ability to implement certain features. Limitations in technology could restrict the type of hardware or software that can be used, which could impact the functionality or performance of the system. Ultimately, it is important to consider all of these limitations when designing and implementing any system to ensure that the final product meets the client's needs while staying within the defined constraints.

### Gantt Chart

