# CS 255 Business Requirements Document

Michael Watkins

12/01/2023

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 the DriverPass project is to fill a market gap in training students for their driving tests at local Department of Motor Vehicles (DMV) offices. The client, DriverPass, led by Liam, aims to provide improved driver training, including online classes, practice tests, and on-the-road training. They require a system that allows them to access data both online and offline, and manage various aspects of their training services​

### 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 wants their system to provide online classes, practice tests, on-the-road training, and the ability to access data both online and offline. They aim to address the high failure rate of driving tests at the DMV. The system components needed include:

1. Online appointment booking, cancellation, and modification.
2. Flexible package options for driving lessons.
3. Registration process for customers' personal and payment information.
4. Scheduling system to match customers with specific drivers and cars.
5. Security features for different employee roles and rights.
6. Tracking system to monitor reservations, cancellations, and modifications

### 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 completed, the DriverPass system should be able to:

1. Offer online classes and practice tests.
2. Schedule on-the-road training.
3. Allow users to access data both online and offline, with the ability to download reports and information.
4. Enable users to make, cancel, and modify appointments online.
5. Include different packages for driving lessons, with the flexibility to customize them in the future.
6. Feature security controls for different employee roles and access rights.
7. Track changes made in the system, like reservations, cancellations, and modifications.
8. Schedule lessons and match customers with specific drivers and cars.
9. Process customer registrations with personal and payment information.
10. Stay compliant with DMV requirements, receiving updates on rules and policies.

Measurable tasks for the system design to achieve these capabilities include creating a user-friendly interface, integrating a secure and flexible appointment scheduling system, developing a robust data management and tracking system, and ensuring compliance with DMV updates and requirements​.

## 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?*

This system should operate in both web-based and mobile application environments to ensure accessibility and convenience. It should run with minimal latency, aiming for response times under a few seconds for user interactions. Regular updates, ideally on a bi-monthly basis or as needed, should be scheduled to ensure security, introduce new features, and improve user experience.

#### 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 should be compatible with major platforms like Windows, macOS, and Unix/Linux for broader accessibility. For mobile access, it should support iOS and Android. The backend requires a robust database system (such as MySQL, PostgreSQL, or MongoDB) for data storage and management, and may also benefit from server-side technologies like Node.js or .NET for efficient application support.

#### 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?*

Different users can be distinguished through unique identifiers like usernames or email addresses, along with secure password authentication. The input, especially for login credentials, should ideally be case-sensitive to enhance security. The system should inform the admin of a problem in scenarios like repeated failed login attempts, unusual activity patterns indicating potential security breaches, or system errors and malfunctions.

#### 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, changes to the user (add/remove/modify) should be possible through an administrative interface without changing the underlying code. The system should be designed with a modular architecture to adapt seamlessly to platform updates, ensuring compatibility with new versions of operating systems and browsers. The IT admin needs comprehensive access, including user management, system settings control, and the ability to update or modify system configurations as required.

#### 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?*

For user login, a combination of a unique username (or email) and a strong password is required. To secure data exchange between the client and server, use HTTPS with SSL/TLS encryption. In case of a brute force attack, the account should be temporarily locked after a predefined number of failed attempts, and the user and admin should be notified. If a user forgets their password, they should be able to reset it via a secure process involving email verification or security questions.

### 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 when logging in.
* The system shall allow users to view, book, and cancel reservations.
* The system shall enable administrators to add or modify available slots and manage user accounts.
* The system shall provide real-time updates on slot availability to users.
* The system shall securely process and store payment information for transactions.
* The system shall generate and send confirmation and cancellation notifications to users.
* The system shall maintain a transaction history for both users and administrators.
* The system shall offer a password reset feature for users who forget their login credentials.

### 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 needs to be user-friendly, intuitive, and responsive to cater to different user types. The primary users include customers, administrators, and instructors. Customers need to view and book slots, check their reservations, and make payments. Administrators should manage slots, user accounts, and view transaction histories. Instructors might need to view and manage their schedules. Interaction with the interface should be versatile, accommodating both mobile and web browser access, ensuring ease of use across various devices.

### 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?*

Not specifically addressed in the design are detailed security protocols beyond basic login, specific accessibility features for users with disabilities, and integration with external systems like banking APIs for payments. The design assumes that users have access to basic internet-enabled devices and a moderate level of technological proficiency to navigate the interface. It also presumes the availability of stable internet connectivity for uninterrupted access to the system.

### 

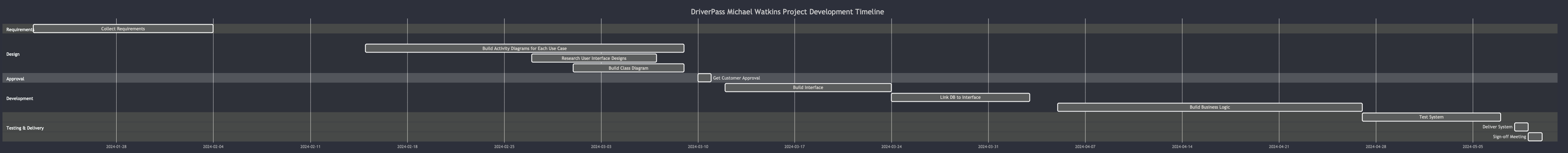
### Limitations

*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 limitations in the system design include potential scalability challenges under high user load, limited advanced features due to current technology constraints, and potential compatibility issues with older devices or browsers. Resource-wise, there are constraints in terms of budget, which may limit the use of premium tools or technologies, and time, which could affect the depth of testing and refinement. Additionally, the current technology stack might limit the implementation of cutting-edge features or integrations until further advancements or budget increases.

### 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.*



Gantt chart too wide – exported it as a png and included it in files.