# 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?*

* This project is focused on developing a system for DriverPass. This company offers driver training to individuals preparing for their driving tests at local Department of Motor Vehicles (DMV) offices. The system will enable customers to access online classes, take practice tests, and schedule on-the-road training sessions. The client, DriverPass, wants the system to be accessible online and offline from any computer or mobile device. They also want the system to be secure, with different employees having different rights and roles. The system should track who made, canceled, or modified reservations for driving lessons. Customers should be able to make reservations for two-hour driving lessons online or by phone. The system should also track which driver and car each customer will go out with. DriverPass offers three training packages, and the system should be flexible enough to allow for customization of these packages in the future. The system should also be able to collect customer information, including credit card details, and allow customers to reset their passwords automatically. The system should be connected to the DMV to update DriverPass with new rules, policies, or sample questions.

### 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 system must be secure, with distinct employee roles and access rights. It should also track who creates, cancels, or modifies driving lesson reservations. Customers should be able to make reservations for two-hour driving lessons online or by phone. The system should also track which driver and car each customer will go out with. DriverPass offers three training packages, and the system should be flexible enough to allow for customization of these packages in the future. The system should also be able to collect customer information, including credit card details, and allow customers to reset their passwords automatically. The system must be connected to the DMV to ensure that DriverPass can be updated with the latest rules, policies, and sample questions.
* Furthermore, there is a need to enhance the currently inadequate driver training options available to customers, as this has contributed to a high failure rate on DMV driving tests. DriverPass aims to tackle this issue by offering a comprehensive training program that includes online classes, practice tests, and on-the-road training sessions. To ensure the successful implementation of this product, the following systems must be established:
  + **User Accounts and Roles**: The system must support different user roles with specific permissions and access levels. The various roles outlined include:
  + Administrator: Full access to all accounts and system functionalities (e.g., Liam, the owner).
  + IT Officer: Responsible for system maintenance, modifications, and user management (e.g., Ian).
  + Secretary: Can schedule, modify, and cancel appointments.
  + Customer: Can create an account, register for packages, and schedule, modify, or cancel appointments. They can also access online classes and practice tests.
  + Driver: Can view assigned appointments, provide feedback, and track working hours.

In addition, they will need an online class and practice test system to take online courses and practice tests. The system will provide online classes and practice tests for the driver's license exam, featuring content that the DMV can update.

An appointment scheduling system will enable customers and secretaries to schedule, modify, and cancel driving appointments.

A Driver Management System is essential for managing driver profiles, including availability, assigned appointments, and performance feedback. A Package Management System will also be implemented to oversee various training packages. This system will allow customers to register for the packages and enable the administrator to customize or deactivate them.

The Reporting and Tracking system generates reports on various activities, such as user activity, appointment history, and driver performance. It should also track who made specific changes to the system.

Along with reporting and tracking, a communication system must facilitate communication between DriverPass and its customers. For example, it should send notifications about appointments or DMV updates and allow customers to contact DriverPass.

Of course, a security system must ensure data security and protect sensitive information, such as customer credit card details. It should also support features like password resets and access control based on user roles.

Lastly, the Offline Access system allows certain users, like the owner, to access and download data for offline viewing and analysis.

DriverPass will need the UI to create a user-friendly interface accessible from various devices, including computers and mobile devices. The interface should display information clearly and intuitively, as the owner has sketched.

### Objectives and Goals

*What should this system be able to do when it is completed? To achieve this, what measurable tasks need to be included in the system design?*

* When completed, the DriverPass system should be able to perform the following tasks:

- Create and manage user accounts with different roles and permissions, including administrator, IT officer, secretary, customer, and driver.

- Host and deliver online classes and practice tests with content the DMV can update.

- Allow customers and secretaries to schedule, modify, and cancel driving appointments, including selecting the desired date, time, and driver.

- Manage driver profiles, including their availability, assigned appointments, and performance feedback.

- Manage various training packages, enabling customers to register while allowing the administrator to customize or disable them.

- Generate reports on user activity, appointment history, and driver performance.

- Track changes made to the system and identify the user responsible for each change.

- Facilitate communication between DriverPass and its customers, such as sending notifications about appointments or updates from the DMV.

- Ensure data security and protect sensitive information, including customer credit card details.

- Provide offline access to specific users for viewing and analyzing data.

- Display a user-friendly interface accessible from various devices, including computers and mobile devices.

In addition, the following will be the measurable tasks needed:

1. Develop and test user accounts for various roles, ensuring appropriate permissions are assigned.

2. Create and upload online classes and practice tests, allowing the DMV to update the content as needed.

3. Implement and test appointment scheduling functionality that allows users to create, modify, and cancel appointments.

4. Develop and test driver management features, including the creation of profiles, availability tracking, and feedback mechanisms.

5. Create and test package management features that enable customer registration and administrator customization.

6. Develop and test reporting and tracking features to ensure accurate data capture and presentation.

7. Implement and test communication features like notifications and contact forms.

8. Implement and test security measures, including data encryption and access control.

9. Develop and test offline access functionality to ensure data integrity and security.

10. Design and test the user interface across different devices to ensure accessibility and usability.

## Requirements

### Nonfunctional Requirements

*This section will detail the different nonfunctional requirements for the DriverPass system. You must consider what the system needs to function correctly.*

#### 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 system must be accessible from **any computer or mobile device**, both **online and offline**. [cite: 23, 24, 25, 26, 27, 28] This suggests a **web-based application** is the most suitable solution. This allows platform independence and accessibility from any device with an internet connection. The offline requirement implies that certain functionalities, possibly for administrative tasks, should be available through a **desktop application** or a **mobile application with offline capabilities**.

**How fast should the system run?**

While the transcript doesn't explicitly state performance requirements, it implies that the system should be **responsive and efficient**. Customers should be able to schedule driving lessons quickly and easily, and administrators should be able to access and manage data without significant delays. A reasonable target would be to ensure that all core functions (e.g., scheduling an appointment, and accessing online classes) execute within a few seconds.

**How often should the system be updated?**

The system should be updated regularly to ensure it remains secure, functional, and up to date with any changes from the DMV. Here's a breakdown of potential update frequencies:

* **Security Updates:** As needed, whenever vulnerabilities are discovered or security patches are released. This could be as frequent as weekly or monthly.
* **Content Updates (online classes, practice tests):** As needed, based on changes in DMV regulations or test content. This could be quarterly or annually.
* **Feature Updates and Bug Fixes** are made regularly, monthly or quarterly, based on user feedback and identified issues.

A robust testing process is also essential to ensure that updates do not introduce new bugs or negatively impact system performance.

#### 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 must be accessible online and offline from any computer or mobile device, so it's likely to be a web-based application. This means the system should be platform-independent and accessible from any device with an internet connection.

However, the transcript also mentions that the owner, Liam, wants to be able to access and download data for offline viewing and analysis. This suggests that a desktop or mobile application with offline capabilities might also be required, particularly for administrative tasks.

Therefore, the system should ideally be able to run on multiple platforms, including:

* **Windows:** To cater to desktop and laptop computers' most common operating system.
* **macOS:** To support Apple users.
* **Android and iOS:** To enable access from mobile devices like smartphones and tablets.

Regarding the back end, a database will be required to support the application. The database will be used to store and manage various types of data, including:

* **User data:** Information about customers, drivers, secretaries, and administrators, such as their names, contact details, login credentials, and roles.
* **Appointment data:** Details about scheduled driving lessons, including date, time, driver, customer, and car assigned.
* **Package data:** Information about the different training packages offered by DriverPass, including their prices and features.
* **Online class and practice test data:** The DMV can update the content of the online courses and practice tests.
* **Reporting and tracking data:** User activity, appointment history, and driver performance data.

The specific type of database (e.g., relational or NoSQL) will depend on the technical architecture and the application's particular requirements. However, choosing a reliable, scalable, and secure database is crucial to ensuring the integrity and availability of the data.

#### 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 in the DriverPass system will be distinguished primarily through their **usernames** and **roles**.

* **Usernames:** Each user will have a unique username that is their primary identifier and will be used to log in to the system.
* **Roles:** Each user will be assigned a specific role (e.g., administrator, IT officer, secretary, customer, driver), determining their permissions and access levels within the system. [cite: 29, 30, 31, 32, 33, 34]

**Is the input case-sensitive?**

Generally, it's best practice to **avoid case sensitivity** for user inputs like usernames and passwords. This improves user experience by reducing the chance of login errors due to capitalization mistakes. However, the specific implementation will depend on the chosen programming language and authentication system.

**When should the system inform the admin of a problem?**

The system should inform the administrator (Liam) of a problem in the following situations:

* **Security breaches or attempted breaches:** Any suspicious activity, such as failed login attempts or unauthorized access attempts, should be immediately reported to the administrator.
* **System errors or failures:** The administrator should be notified if the system encounters critical errors that affect its functionality or availability. This includes database errors, application crashes, or server issues.
* **Unusual or suspicious user activity:** Any unusual patterns of user behavior, such as a sudden increase in failed login attempts or many data modifications, should be flagged to the administrator.
* **Scheduled maintenance or updates:** The administrator should be informed about planned maintenance activities or system updates that might temporarily affect the system's availability.
* **Data backup and recovery issues:** The administrator should be notified of any problems during data backup or recovery processes.

The system can inform the administrator through various channels, such as:

* **Email notifications:** Sending alerts to the administrator's email address.
* **In-app notifications:** Displaying notifications within the application's interface.
* **SMS messages:** Sending text message alerts to the administrator's phone.

The specific notification method and the level of detail provided will depend on the severity and urgency of the problem.

#### Adaptability

*Can you change 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?*

The system should be designed to allow for user management without requiring code changes. This is typically achieved through an administrative interface within the application itself. This interface should allow authorized users (like the administrator or IT officer) to:

* **Add new users:** Create new user accounts with appropriate roles and permissions.
* **Remove existing users:** Deactivate or delete user accounts.
* **Modify user information:** Update user details such as names, contact information, roles, and passwords.

This administrative interface would interact with the database to make the necessary changes, eliminating the need for direct code modifications for basic user management tasks.

**How will the system adapt to platform updates?**

The system should be designed with platform independence in mind to adapt to platform updates (e.g., new versions of Windows, iOS, and Android). This means:

* **Using cross-platform technologies:** Employing programming languages and frameworks across different operating systems.
* **Following platform-specific guidelines:** Adhering to design and development guidelines provided by each platform to ensure compatibility.
* **Regular testing:** Conduct thorough testing on different platforms and operating system versions after each update to identify and address compatibility issues.
* **Modular design:** Creating a modular architecture that allows easy updates and modifications to specific components without affecting the entire system.

**What type of access does the IT admin need?**

The IT admin (Ian, in this case) needs a high level of access to manage and maintain the DriverPass system effectively. This includes:

* **System-level access:** Access to the server infrastructure, databases, and application files to perform maintenance tasks, troubleshoot issues, and apply updates.
* **User management:** Ability to add, remove, and modify user accounts and their permissions.
* **Security management:** Access to security settings and logs to monitor system activity, identify potential threats, and implement security measures.
* **Data management:** Ability to access and manage the database, including performing backups, restoring data, and optimizing performance.
* **Application monitoring:** Access to tools and logs to monitor the application's performance, identify errors, and troubleshoot issues.

While the IT admin needs extensive access, proper security measures must be implemented to prevent unauthorized access and protect sensitive data. These measures include strong passwords, two-factor authentication, and access control lists to restrict access to specific functions and data.

#### 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 a “brute force” hacking attempt exists? What happens if the user forgets their password?*

At a minimum, the DriverPass system should require the following for user login:

* **Username:** A unique identifier for each user.
* **Password:** A secret combination of characters known only to the user.

To enhance security, consider implementing:

* **Password complexity requirements:** Enforce rules for password creation, such as minimum length, character types (uppercase, lowercase, numbers, symbols), and the disallowance of common passwords.
* **Two-factor authentication:** Add an extra layer of security by requiring a second verification form, such as a code sent to the user's email or phone.

**What should happen to the account if a "brute force" hacking attempt exists?**

* **Account lockout:** Temporarily lock the account after a certain number of failed login attempts to prevent brute-force attacks.
* **IP address blocking:** Block IP addresses that exhibit suspicious activity, such as multiple failed login attempts from the same IP address.
* **Notification to the administrator:** Alert the administrator of suspected brute-force attacks.

**What happens if the user forgets their password?**

* **Password reset:** Allow users to reset their passwords through a secure process, such as sending a password reset link to their registered email address.
* **Security questions:** Optionally, offer security questions as an alternative method for password recovery.

### 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.”*

* Here is a small sampling:

 The system shall allow customers to create and manage their accounts, including registering for training packages and scheduling driving lessons. [cite: 18, 19, 20, 21, 22]

 The system shall provide access to online classes and practice tests for the driver's license exam. [cite: 46, 47, 48, 49, 50]

 The system shall allow customers to schedule, modify, and cancel driving appointments online or by phone. [cite: 36, 37, 38, 39, 40, 41]

 The system shall allow secretaries to manage appointments, including scheduling, modifying, and canceling customer appointments. [cite: 42, 43, 44, 45]

 The system shall assign drivers to appointments and track their availability. [cite: 42, 43, 44, 45]

 The system shall track the car assigned to each driving lesson. [cite: 42, 43, 44, 45]

 The system shall manage different training packages and allow for customization of these packages in the future. [cite: 46, 47, 48, 49, 50]

 The system shall collect customer information, including credit card details, securely. [cite: 55, 56, 57, 58, 59, 60]

 The system shall allow customers to reset their passwords automatically. [cite: 55, 56, 57, 58, 59, 60]

### User Interface

*What are the needs of the interface? Who are its different users? What will each user need to be able to do through the interface? How will the user interact with it (mobile, browser, etc.)?*

**What are the needs of the interface?**

The DriverPass system interface needs to be:

* **User-friendly:** Easy to navigate and understand for users with varying levels of computer literacy.
* **Accessible:** Usable on different devices (computers, tablets, smartphones) and with various browsers.
* **Informative:** Present information about training packages, schedules, and account details.
* **Secure:** Protect user data and prevent unauthorized access.
* **Efficient:** Allow users to complete tasks quickly and easily.
* **Visually appealing:** Have a clean, modern design that aligns with DriverPass's branding.

**Who are the different users of this interface?**

The DriverPass system will have different user roles, each with specific needs and functionalities:

* **Customer:** Access online classes, practice tests, schedule driving lessons, manage their account.
* **Secretary:** Manage customer appointments and assign drivers to lessons.
* **Driver:** View assigned appointments and provide feedback on lessons.
* **IT Officer:** Manage system settings, user accounts, and security.
* **Administrator (Liam):** Full access to all system functionalities, including reports and data analysis.

### Assumptions

*What things were not explicitly addressed in your design above? What assumptions are you making in your design about the users or the technology they have?*

While I've tried to be comprehensive, here are some aspects that might need further clarification or consideration:

* **Scalability:** How will the system handle a significant increase in users or data? Will it need to be deployed on multiple servers or use cloud-based infrastructure?
* **Third-party integrations:** Will the system need to integrate with any third-party services, such as payment gateways, mapping services, or the DMV's database?
* **Specific technologies:** What programming languages, frameworks, and databases will be used to build the system?
* **Accessibility features:** What specific accessibility features will be implemented to support users with disabilities?

### Limitations

*Any system you build will naturally have limitations. What limitations do you see in your system design? What are your resources, time, budget, or technology limitations?*

**System design limitations:**

* **Offline functionality:** While offline access is mentioned, the offline functionalities might be limited.
* **Real-time updates:** The system might not provide real-time updates for all data, such as driver availability or DMV changes.
* **Integration challenges:** Depending on its technology and data-sharing policies, integrating with the DMV's systems could pose challenges.

**Resource, time, budget, or technology limitations are always in play.**

### Gantt Chart

*Please include a screenshot of the GANTT chart that you created with Lucidchart. Ensure that it meets the plan described by the characters in the interview.*

