# 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?*

* Establish a better driver training to help students pass their driving tests at the local DMV (Department of Motor Vehicles).
* Liam-owner of DriverPass
* Develop a system for DriverPass
  + To offer online classes
  + Practice tests
  + On the road training
  + Manage driving lesson reservations

### 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 seeks to develop a robust system with several key objectives:

* **Customer Registrations:** The system should facilitate customer registrations, offering three distinct packages. Users should have the ability to add or remove packages as needed. To complete the registration, users are required to provide essential information such as their first name, last name, address, phone number, state, and credit card details, including expiration date and security code.
* **Appointment Management:** The system should manage appointments effectively, particularly 2-hour driving lessons. This includes specifying the day and time for lessons. Users should have the capability to make, cancel, or modify appointments. The system must also handle pickup and drop-off location details.
* **Tracking:** The system should track user/driver/car/time matchups for each driving lesson, ensuring accurate monitoring and record-keeping.
* **Website Requirements:** The website should have specific features, including the ability to automatically reset user passwords if forgotten. It should operate on a cloud-based infrastructure for scalability and accessibility. Additionally, the website should provide progress tracking for online tests, displaying customer data to indicate what's currently in progress and what's already completed.
* **Sync with DMV:** The system should synchronize with the Department of Motor Vehicles (DMV) to receive new updates regarding rules, policies, and sample questions relevant to driver training.

The problem that DriverPass aims to address is the lack of effective driver training resources for students preparing for their driving tests at the local DMV. To achieve these goals and functionalities effectively, the system requires several distinct components:

* **Administration:** This component is essential for administrators to control different accounts and manage permissions. It determines who can access specific functionalities within the system.
* **Secretary Account:** Users with secretary accounts should be able to manage appointments. This includes scheduling appointments received by phone or from walk-in customers.
* **User Account:** Students and customers need user accounts that enable them to schedule driving lessons according to their preferences.
* **Driver Accounts:** Drivers providing driving lessons should have dedicated accounts to log notes and details after each lesson.

### 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 is completed, its high-level objectives are:

* Automation of Scheduling: Implement an automated scheduling system for driving lessons.
* User Status Tracking: Develop a tracking mechanism to monitor user statuses within the system.
* Driver and Car Tracking: Create a system component for tracking drivers and cars used for driving lessons.

The goals are achieved if the system provides the following essential functionality:

* Scheduling Services: Offer online scheduling services for driving lessons.
* Online Classes: Deliver online classes with study materials and quizzes.
* Comprehensive Tracking: Implement a comprehensive tracking system that logs all operations performed within the system, including user actions such as additions and deletions, along with timestamps.
* Flexibility: Provide users with the flexibility to register for courses through both online and agent-assisted (secretary) methods.

## 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?*

* DriverPass aims to provide a user-friendly, web-based platform accessible from various devices, including desktop computers, laptops, tablets, and smartphones. The web-based environment ensures that users can easily access the system through standard web browsers, making it convenient for both students and administrators.
* In terms of system speed, the goal is to deliver a responsive and efficient experience. Given that the system involves scheduling appointments, online classes, and real-time tracking, it should prioritize swift response times. Actions such as scheduling driving lessons, accessing study materials, and logging driving lesson details should occur without noticeable delays to enhance the user experience.
* Regarding system updates, the approach is to maintain a regular schedule to address various aspects of the system. Updates aim to enhance functionality, improve security, and introduce new features. The frequency of updates will depend on several factors, including the complexity of changes, the introduction of new regulations or requirements from the DMV, and feedback from users. Generally, the system will undergo updates on a quarterly or semi-annual basis to ensure it remains current, secure, and efficient, aligning with industry best practices and user expectations.

#### 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 run on multiple platforms to ensure broad accessibility and compatibility. These platforms include:

* Windows: To cater to users who prefer Windows-based devices.
* Unix/Linux: To provide compatibility with Unix-based systems.
* macOS: To accommodate users with Apple devices such as MacBooks and iMacs.
* Web Browsers: The system should be accessible via popular web browsers such as Chrome, Firefox, Safari, and Edge, ensuring cross-platform support.

For the backend, the system will indeed require a database to support its application. The database is essential for storing and managing various types of data, including user profiles, scheduling information, and driving lesson records. To facilitate efficient data management and retrieval, a relational database management system (RDBMS) like MySQL, PostgreSQL, or Microsoft SQL Server would be suitable. The specific choice of database system would depend on factors such as scalability, data volume, and the development team's expertise. Additionally, the backend may require tools and technologies for web development, server hosting, and data security to ensure the system's reliability and performance.

#### 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?*

* To distinguish between different users, each user will register with an email address, username and password. I will implement user 2FactorAuthentication. 2FA adds an additional layer of verification beyond passwords, making it harder for unauthorized users to gain access. It typically involves something the user knows (password) and something the user has (a one-time code generated by a mobile app or sent to their registered email or phone number). This approach not only distinguishes between users but also significantly strengthens the system's security, reducing the risk of unauthorized access and potential breaches.
* The input for usernames can be designed as case-insensitive, ensuring that users can log in using uppercase or lowercase characters interchangeably. This user-friendly approach reduces the risk of login errors due to case sensitivity. However, passwords should remain case-sensitive for security reasons.
* The system should inform the admin of a problem in real-time or near-real-time for critical issues that impact system functionality, security, or user experience. For instance, security breaches, system failures, database errors, and low resource conditions should trigger immediate notifications to the admin. Additionally, a proactive approach to user feedback should be established, where the admin is informed of user-reported problems or concerns, allowing for swift resolution and continuous improvement of the system. Regular log reviews and scheduled reports can also help the admin stay informed about the system's performance and any potential issues that require attention.

#### 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?*

* To enable easy changes to users without altering the code, I would implement a robust user management module. This module should provide administrative access to authorized personnel, allowing them to add, remove, or modify user accounts through a user-friendly interface. It should support role-based access control (RBAC), granting different permissions to administrators based on their roles. This way, IT admins can handle user management without needing to access or modify the underlying code.
* In terms of adapting to platform updates, the system should follow best practices for version control and dependency management. Keeping the system components modular and well-documented will facilitate updates. Regularly checking for updates in the underlying platforms, libraries, and frameworks used in the system is essential. Additionally, a dedicated team or individual should be responsible for monitoring platform updates and ensuring compatibility.
* Regarding IT admin access, they should have privileged access to perform essential administrative tasks such as user management, system monitoring, and troubleshooting. However, their access should be strictly controlled and limited to prevent unauthorized changes or potential security breaches. Role-based access control can help define the specific permissions and responsibilities of IT admins, ensuring that they have the necessary access without compromising system integrity.

#### 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?*

* **User Authentication**: Users will need to log in with their unique username and password. They will also undergo a two-factor authentication.
* **Secure Connection**: To secure the connection between the client and the server, the system will use HTTPS (SSL/TLS). This encryption protocol ensures that data exchanged between the user's device and the server is encrypted and secure against eavesdropping.
* **Account Lockout**: In the event of a brute force hacking attempt (multiple failed login attempts within a short time), I will implement an account lockout policy. After a predefined number of unsuccessful login attempts, the user's account will be temporarily locked to prevent further unauthorized access.
* **Password Reset**: If a user forgets their password, the system will provide a secure password reset mechanism. This will involve sending a password reset link or code to the user's registered email address or mobile number. The user can then follow the link or enter the code to reset their password.
* **Security Questions**: As an additional security layer for password resets, users will be able to set up security questions and answers during the initial account setup. When they forget their password, they can answer these questions to verify their identity.
* **Account Recovery**: In some cases, if all else fails, users may need to go through an account recovery process, which will require them to contact customer support and providing additional information.

### 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 allow users to create accounts by providing their first name, last name, address, email address, phone number, state, and credit card information, including card number, expiration date, and security code.
* The system shall enforce validation rules for user-provided information, ensuring completeness and accuracy during registration.
* The system shall offer the option for users to select from three different packages for driving lessons.
* The system shall enable users to add and remove packages as needed.
* The system shall allow the owner to disable packages if no users register for them.
* The system shall enable customers to schedule 2-hour driving lessons.
* The system shall allow users to specify the day and time for their driving lessons.
* Users shall have the capability to make, cancel, or modify their driving lesson reservations.
* The system shall track reservations, recording details such as the date, time, assigned driver, and car.
* The system shall differentiate between various user roles, including customers, IT officers, and secretaries.
* Each user role shall have specific permissions and access rights within the system.
* The system shall provide user authentication mechanisms, including the use of usernames and passwords.
* It shall support password reset functionality in case a user forgets their password.
* The system's website interface shall automatically reset a user's password if forgotten.
* It shall run on a cloud infrastructure for scalability and accessibility.
* The website shall display online test progress for users.
* Users shall have access to their customer data, including information about ongoing processes and completed actions.
* The system shall be synced with the DMV (Department of Motor Vehicles) to receive updates regarding rules, policies, and sample questions.

### 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.)?*

* **Customer Interface**:
  + **Users**: Customers preparing for their driving tests.
  + **Needs**: Customers need an easy-to-use interface to access driving lesson services, schedule lessons, manage their accounts, and track their progress.
  + **Functionality**:
    - Customers should be able to log in securely using a web browser or a mobile app.
    - They should have access to a user-friendly dashboard to view and manage their account details.
    - The interface shall provide options for scheduling, modifying, or canceling driving lessons.
    - Customers should be able to browse study materials and take quizzes for online classes.
    - The interface shall display their driving lesson reservations and progress.
* **IT Officer Interface**:
  + **Users**: IT officers responsible for system maintenance and updates.
  + **Needs**: IT officers require tools for maintaining the system, monitoring user activities, and ensuring system integrity.
  + **Functionality**:
    - IT officers shall access the system through a secure web-based admin portal.
    - The interface shall provide tools for monitoring user activities, managing user accounts, and performing maintenance tasks.
    - IT officers should be able to review logs of system operations and user actions.
    - The interface shall facilitate system updates and data synchronization with external sources.
* **Secretary Interface**:
  + **Users**: Secretaries responsible for assisting customers with scheduling and registrations.
  + **Needs**: Secretaries need an interface to manage customer accounts, schedule driving lessons, and provide support.
  + **Functionality**:
    - Secretaries shall access the system through a web browser or a dedicated application.
    - The interface shall enable secretaries to assist customers in registering for driving lessons.
    - Secretaries should be able to schedule driving lessons on behalf of customers.
    - They shall have access to customer information and reservations to help.
* **Owner/Administrator Interface**:
  + **Users**: The owner or administrator overseeing the driving lesson service.
  + **Needs**: The owner requires a comprehensive interface for managing the entire system, including packages, finances, and overall performance.
  + **Functionality**:
    - The owner/administrator shall access a secure web-based admin portal.
    - The interface shall provide tools for managing packages, including adding, removing, and disabling them.
    - Financial data and reports should be accessible to monitor business performance.
    - The owner/administrator shall be able to view system logs and user activities.
    - Integration with external sources, such as the DMV, should be manageable through the interface.

### 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?*

* **User Language Proficiency**: It assumes that users are proficient in the language(s) used within the interface. There are no provisions for multilingual support or translation features.
* **Data Privacy and Security**: While it's implied that sensitive user data like credit card information will be handled securely, the design does not delve into specific security measures, encryption standards, or compliance with data protection regulations (e.g., GDPR or HIPAA).
* **User Training**: The design assumes that users, including customers, secretaries, and IT officers, are proficient in using the interface. It doesn't address potential user training needs or user onboarding processes.
* **User Support**: The design assumes that users will not face significant issues while interacting with the system. There's no mention of a comprehensive user support system, such as FAQs, chat support, or customer service contact details.
* **Scalability**: It's assumed that the system will handle a certain number of users and data. Scalability considerations for accommodating growth in users or data volumes are not outlined.
* **System Maintenance**: While IT officers are responsible for system maintenance, the design doesn't elaborate on the processes and tools they will use for routine maintenance and updates.
* **User Demographics**: Assumptions about user demographics, such as their level of technological literacy or familiarity with driving lesson bookings online, have been made without providing a breakdown.
* **Cost Considerations**: The design doesn't address the budget or cost considerations associated with developing, maintaining, and hosting 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?*

* **Geographical Scope**: The system's services are primarily accessible to users with internet access within a specific geographical area. It may not cater to users in regions with limited connectivity or in areas outside the defined service area.
* **Hardware and Software Compatibility**: The system assumes that users have compatible hardware and software to access the platform effectively. Older devices or non-standard configurations may not provide an optimal user experience.
* **Data Accuracy**: The accuracy of data provided by external sources, such as the DMV, is assumed but not entirely within the system's control. Inaccuracies or delays in external data could affect the system's performance.
* **Budget Constraints**: The system design operates within predefined budget constraints, potentially limiting the scope of development, features, or scalability.
* **Time Constraints**: Development and implementation are subject to time constraints. These limitations could affect the completeness and thoroughness of the system.
* **Human Resources**: Availability and expertise of human resources for system development and maintenance may be limited, impacting the speed and quality of implementation.
* **Hardware Resources**: The hardware infrastructure and hosting resources are budget-dependent and may restrict the system's scalability and responsiveness.
* **Software Resources**: Software licensing costs, particularly for third-party tools and services, could limit the incorporation of advanced features or functionalities.
* **Technological Advancements**: Rapid advancements in technology may render certain system components or software outdated or incompatible, requiring ongoing updates.
* **Integration Challenges**: Integration with external systems, such as DMV databases, may be subject to technological constraints and compatibility issues beyond the system's control.
* **Scalability**: The chosen technology stack may have limitations in terms of scalability, potentially affecting the ability to accommodate a growing user base.

### 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 screenshot of a calendar

Description automatically generated