# CS 255 Business Requirements Document Template

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 this project is to design and develop a comprehensive system for DriverPass, a company that seeks to provide effective driver training for customers. The client for this project is DriverPass

Here is a list of some key functions needed for DriverPass:

1. Enable customers to take online classes and practice tests.
2. Facilitate on-the-road training through reservation of driving lessons.
3. Allow access to the data from anywhere, both online and offline, with the capability to download reports for offline use.
4. Handle different access levels and roles for different employees, with an emphasis on security.
5. Track changes and modifications to reservations and user records, and provide a clear audit trail.
6. Facilitate online appointment scheduling by customers.
7. Manage different driving lesson packages, with the flexibility to customize these packages in the future.

Driver pass wants their system to handle all aspects of their business from course registration to appointment booking, while ensuring strong data management and user-friendly interactions.

### 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 address the issue of high failure rates in driving tests at the Department of Motor Vehicles by providing a comprehensive driver training program. They want the system to facilitate this solution by offering an end-to-end service that includes online classes, practice tests, and on-the-road training.

Some different components needed for this system are:

1. **Education and Assessment**: Provide online classes and practice tests for customers preparing for driving tests.
2. **Reservations Management**: Enable customers to book, change, and cancel driving lesson reservations.
3. **Data Management and Security**: Offer secure data access and management, with role-based access levels and change tracking.
4. **Customer Service Features**: Handle customer registration, password resets, and lesson packages customization.
5. **DMV Integration**: Stay updated with DMV policies, rules, and sample questions to provide the most current training.

### 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?***

Upon completion, this system should be able to execute the following tasks:

1. **Online Training and Assessment**: The system should support online classes and practice tests. Progress tracking and test scores should be visible to the user.
2. **Lesson Reservation Management**: It should be capable of scheduling, modifying, and cancelling lesson reservations. Each reservation should specify the date, time, and associated driver.
3. **Customer Registration**: The system needs to support the collection of customer details such as name, address, phone number, credit card details, and pickup and drop-off locations.
4. **Package Management**: It should allow for the selection of lesson packages, and enable the disabling of certain packages if necessary.
5. **Integration with DMV**: The system must be able to receive and process updates from the DMV regarding changes in rules, policies, or sample questions.
6. **User Interface Design**: A clear and intuitive interface that displays information like test progress and driver notes. The design should be based on the client's sketch and meet their specifications.

All of these tasks are measurable in the sense that their implementation and functionality can be tested and verified against the requirements set out by the client.

## 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?*

1. **Environments**: The DriverPass system should function as a web-based platform, making it accessible from different devices (computers, tablets, smartphones) and operating systems (Windows, MacOS, iOS, Android, etc.) to cater to a wide range of users.
2. **System Speed**: The system must be quick and responsive to provide a seamless user experience. Load times for different functionalities (logging in, accessing course materials, scheduling lessons, etc.) should not exceed 2-3 seconds on a stable and standard internet connection.
3. **Updates**: Regular system updates should occur to keep the system secure, optimized, and up-to-date with the latest industry standards and user needs. Minor updates and patches should be implemented as needed, whereas major updates introducing new features or significant changes should be scheduled for off-peak hours and communicated to the users in advance. Updates should be scheduled at least quarterly, or more frequently as required.
4. **Scalability**: The system should be designed to handle a significant increase in user load without degradation in performance. This is especially important during peak registration periods or when new course material is released.
5. **Reliability**: The system should have high availability and minimum downtime. Regular backups and redundancy plans should be in place to recover from any potential data loss or system failure.
6. **Interoperability**: The system should be able to interact seamlessly with external systems like the DMV, payment gateways, and potentially other driving schools or instructor databases in the future.

These are some requirements to make sure that DriverPass operates with maximum efficiency and can reach the most amount of clients without performance issues.

#### 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?*

1. **Platforms**: The DriverPass system is a web-based application and should be designed to run on various platforms to ensure wide accessibility. This includes operating systems like Windows, MacOS for desktop users, and iOS, Android for mobile users.
2. **Browsers**: The system should be compatible with all modern web browsers, such as Google Chrome, Firefox, Microsoft Edge, Safari, and others, ensuring a consistent user experience across different browsers.
3. **Backend Requirements**:
   * **Database**: A robust relational database such as PostgreSQL or MySQL is required to efficiently manage and store user data, course materials, scheduling information, and payment details.
   * **Server**: The system should be hosted on reliable server platforms that provide scalability, like AWS, Google Cloud Platform, or Linode.
   * **Backend Language**: The backend could be built using languages like Python (with Django or Flask frameworks) or javascript (with Node.js) depending on the development team's expertise.
4. **APIs**: The system will need to integrate with various APIs for functionalities like secure payment processing (Stripe, PayPal), DMV database access, and potentially mapping services for route planning and tracking.

These platform constraints and requirements ensure that the DriverPass system will work effectively across different devices, browsers, and operating systems, providing a seamless user experience regardless of their chosen platform.

#### 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?*

1. **User Differentiation**: We will distinguish between different users using unique user credentials. These will include an email address (or username) and a password. Additionally, user profiles will be categorized based on their roles (such as student, instructor, or admin), which would allow us to manage permissions and accessibility according to their role requirements.
2. **Case Sensitivity**: To maintain consistency and avoid confusion, email addresses used for login should not be case-sensitive, as it is standard practice for web applications. However, password fields should remain case-sensitive to ensure security.
3. **Notification to Admin**: The system should be designed to inform the admin of a problem under the following circumstances:
   * Suspicious login activities or multiple failed login attempts
   * Issues related to payment processing
   * Technical errors in system functionality or server issues
   * Student or instructor complaints or feedback requiring attention
   * Any changes in DMV regulations or policies
4. **Data Validation**: To maintain accuracy and precision, the system should use data validation rules to ensure correct data input. For instance, email addresses should follow the correct format, credit card information should be validated for correctness, etc.
5. **Error Handling**: Proper error handling should be in place to prevent any inaccuracies in data or system malfunctions. Clear, informative error messages should be shown to users, and system-level errors should trigger notifications to the admins or the technical support team.

By focusing on these aspects of accuracy and precision, the DriverPass system can maintain the integrity of user data, ensure smooth operations, and swiftly address any issues or problems that arise.

#### 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?*

1. **User Modifications**: Yes, modifications to the user (add/remove/modify) should be achievable without changing the code. The system will include an administrative dashboard or panel that will allow for user management, including adding new users, removing users, or modifying user details and permissions.
2. **Platform Updates Adaptation**: The system should be designed to adapt to platform updates without disrupting service. This will be achieved by using best practice coding standards, ensuring forward compatibility, and testing thoroughly when updates to the operating system or platform occur. It will also utilize APIs that are robust and are likely to remain supported in future versions of the platform.
3. **IT Admin Access**: The IT admin should have top-level access to all system features. This will include but not be limited to, user management, system configuration, data backup and restore, application updates, monitoring system health, and access to error and activity logs. The IT admin should also have the ability to grant and revoke permissions to other system users as needed. Security measures will be implemented to ensure that this high-level access is protected, such as multi-factor authentication and strict password requirements.

This adaptability allows the DriverPass system to remain flexible and effective in a dynamic technical environment, ensuring the best service possible to its users regardless of external changes.

#### 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?*

1. **User Login**: To login, the user will be required to provide a unique username and a secure password. Additionally, for enhanced security, two-factor authentication could be implemented. This could involve the use of a one-time code sent to the user's registered email or phone number.
2. **Data Exchange Security**: To secure the connection or data exchange between the client and the server, we will implement Secure Sockets Layer (SSL) encryption. This method will encrypt the data during transmission, making it unreadable to anyone except the intended recipient.
3. **Brute Force Attack Protection**: In case of a suspected "brute force" hacking attempt, account lockout mechanisms will be implemented. After a defined number of unsuccessful login attempts (for example, 5), the account will be temporarily locked for a certain period (for example, 15 minutes). If repeated lockouts occur, the account could be locked until manual unlocking by an administrator. This will deter brute force attacks by making them time-consuming and ineffective.
4. **Forgotten Password**: If a user forgets their password, they will be able to use a 'Forgot Password?' function. This function will send a password reset link to the user's registered email address. To ensure security, this link will be time-limited (for example, 24 hours). If the user does not reset their password within this period, they will need to request another reset.

These security measures will help to ensure the integrity and confidentiality of user data and provide a safe and secure environment for the DriverPass system users.

### 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.”*

1. **The system shall** validate user credentials when logging in to ensure that only authorized users gain access to their respective accounts.
2. **The system shall** provide a mechanism to add, modify, or remove users, thereby enabling efficient management of user accounts by authorized administrators.
3. **The system shall** enable users to easily and efficiently schedule, modify, or cancel driving lessons.
4. **The system shall** facilitate automatic allocation of instructors to students based on availability, ensuring optimal utilization of resources.
5. **The system shall** send automated notifications to both instructors and students for any upcoming lessons or changes to scheduled lessons, ensuring effective communication.
6. **The system shall** maintain a log of past driving lessons for each user, allowing for easy tracking and reference.
7. **The system shall** support the functionality for users to leave feedback and ratings for completed lessons, promoting continuous improvement and quality assurance.
8. **The system shall** provide an intuitive and user-friendly interface, facilitating easy navigation and operation by users of varying technical expertise.
9. **The system shall** generate reports and analytics for the administrators, assisting in decision making and continuous improvement.
10. **The system shall** implement robust security measures to protect user data and prevent unauthorized access.

### 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.)?*

1. **Students:**
   * Ability to create and manage an account.
   * Ability to schedule, modify, or cancel driving lessons.
   * Option to view history of past lessons.
   * Capability to provide feedback and ratings for the instructors and the lessons.
   * Availability of a help or support feature for any queries or concerns.
2. **Instructors:**
   * Capability to view their schedule and any updates or changes to it.
   * Ability to accept or decline the assigned lessons based on their availability.
   * Option to view feedback and ratings given by students.
   * Availability of a help or support feature for any queries or concerns.
3. **Administrators:**
   * Ability to manage user accounts, i.e., add, modify, or remove users.
   * Capability to assign instructors to students.
   * Option to generate and view reports and analytics.
   * Ability to handle and resolve support tickets or queries raised by students or instructors.

**User Interaction with the Interface:**

Considering the evolving digital age, it would be ideal for the users to be able to interact with the interface through multiple platforms. Therefore, the system should be accessible through a browser for desktop users and should also have a mobile application for both Android and iOS users, providing convenience and accessibility.

### 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?*

1. **User Tech-Savviness:** It is assumed that the users of the system (students, instructors, and administrators) are familiar with using web-based applications and mobile apps. They understand the basics of how to navigate through the interface and use the different functionalities provided.
2. **Internet Connectivity:** The design assumes that users have consistent and reliable access to the internet. As the system is web-based and may also feature mobile apps, it is critical that users can connect to the internet to use the system.
3. **Device Compatibility:** It is assumed that the users have devices (computers, smartphones, tablets) that are compatible with the latest software and can run the system smoothly.
4. **Tech Support:** It is assumed that there will be a dedicated IT support team available to address any technical issues that arise.
5. **User Training:** It is assumed that the necessary training will be provided to all users, especially administrators and instructors, to efficiently use the system.

These assumptions have a significant impact on the system design and functionality and need to be validated to ensure the system's successful implementation and use.

### 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?*

1. **Technical Compatibility:** As the system is designed to be web-based and potentially include a mobile application, users may experience issues with older devices or outdated software. Ensuring compatibility with all devices and software versions could be challenging.
2. **Internet Dependence:** The system's functionality heavily depends on internet connectivity. In regions or scenarios with poor or no internet access, the system usage may be restricted.
3. **Time and Budget Constraints:** Given project timeline and budget, it might be difficult to incorporate all desired features in the initial version of the system. Some features might need to be rolled out in phases.
4. **Technical Expertise:** Building a sophisticated system like this requires a team of experienced developers and IT professionals. Resource limitations in this area might impact the development process and timelines.
5. **Scalability:** While the system will be designed to accommodate a certain number of users, a sudden increase in users might strain the system's resources and affect its performance.

These limitations are inherent to the process of system design and should be factored into project planning and resource allocation. It's important to manage stakeholder expectations accordingly.

### 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.*

