# 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 Client, DriverPass, wants a system to capture a niche market of new driver training in the United States. The client desires a system that will allow for online, up-to-date training on all required DMV testing material. Additionally, the client wants the ability to schedule in-person driver’s training for these users that will allow them to have all of their drivers permit and driver’s license needs met in one easy to use application/website.

### 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?*

* **Up-to-date study materials –** The client wants up-to-date study materials to be available for all students. This will be done through a system that keeps DriverPass up-to-date on all updates by the DMV in relation to their testing information and procedures.
* **Drive training scheduling –** The client wants to have a reservation system that allows users to book in-person training with a licensed trainer for 2-hour periods. This portion of the application should be able to display a variety of packages selected by DriverPass and allow users to sign up for or cancel these packages at will. DriverPass would like this to be fluid and have the scheduling to be easy to read and understand by employees.
* **All in one –** The client wants all of these features to be given in a custom designed UI that allows the user to study, take practice tests, see their progress, and take driver training all from one easy to use nexus.

### 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?*

* **Allow users to take online tests –** The first objective is to allow these tests to be taken and studied by users. The application should have a database of practice tests and study information that is accessible from the main page.
* **DMV updates –** The next objective is to connect this application to a DMV database that updates the client whenever updates are made so that they can be added to the study material
* **Driver scheduling –** An easy-to-use scheduling system should be created that stores user requests in 2-hour blocks in a calendar database. This database should be accessible to the client. This database should show client times and teacher assigned.
* **Cloud compliance –** Application should work in tandem with secure cloud storage that can access and store user information, to include schedule, teacher assigned to, personal information, payment information, and package information.

## 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?*

* **Environment:** This system will need to be run on a large variety of environments in order to be able to capture all environments the wide variety of clients will be using to learn their driver’s training or schedule appointments. First, this system needs to support **web browsers** like Chrome, Firefox, Safari and Edge. Second, it will need to support Android and iOS capabilities for mobile users if a designated native app is desired. Third, it will need a scalable cloud infrastructure like AWS or Azure to ensure constant availability. Finally, if a native computer app is desired through Microsoft or apple stores, it will need to be compatible with macOS and Windows.
* **Speed:** It will be beneficial for this system to be quick and responsive to compete with industry standards or user expectations for most applications and websites available today. More specifically, when taking tests and studying, it will need to be responsive (under 2 seconds) so as to not disrupt the flow and enjoyability of the user. Also, things like scheduling should be updated in real-time so as not have multiple bookings or unexpected interactions with the system when appointments are being booked by a large number of users.
* **Updates:** This application requires a few things to be consistently updated in order to function properly and efficiently. First, the testing and studying materials should be updated at minimum quarterly, or even more frequently based on the updates released by the DMV. These times can fluctuate greatly depending on local DMV updates and laws. Second, the system itself should have weekly updates within the first year to capture any system issues, bugs, or maintenance issues, then monthly once the system has been stabilized. However, updates for these systems should have zero downtime for users, as this could affect their ability to test and prepare for such an important event. So, things like rolling updates should be used to facilitate this.

#### 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?*

* **Platforms:** As stated above, front end platforms like Chrome, Safari, and Edge should be used, as well as mobile platforms like iOS and Android. This ensures that the application is available for a wide majority of users across the many popular platforms used today. It would be best to capture the two most popular desktop operating systems like macOS and Windows, eventually branching to Linux if desire is seen.
* **Tools:** A programming language will be needed for all of these different platforms. For example, node or python for restful APIs, react for front end integration, and MySQL for databases. Other tools like OAUTH 2.0 can be used for authentication. When it comes to hosting, Kubernetes or Jenkins could be helpful. Other tools like payment gateways (stripe and paypal), DMV data integration through APIs or DMV partnerships, and things like Firebase Cloud push notifications could be used for mobile to ensure users are always up to date.

#### 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:** To keep things simple, users can be distinguished through a unique user ID that is generated through account creation, and user accounts can be distinguished through unique email addresses and phone number combinations, also done through account creation and verification. There will also be different roles, such as student, instructor, or admin, implemented.
* **Input:** For best security practices, usernames, emails, and passwords should all be case-sensitive. Things like DMV learning content can also support case-sensitive searches or Boolean logic to enhance usability.
* **Admin:** The admin should be informed when major issues are detected in the system. Things like internal server errors, or any extended periods of downtime. Things like usage thresholds over 90% or similar issues caused by DDOS attacks should also notify the admin. If any serious synchronization errors occur during scheduled DMV information updates, the admin should also be notified to remedy the issue.

#### 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?*

* **Changes:** Specific features like a user management dashboard can be implemented that focuses entirely on updating user roles, resetting passwords, or modifying accounts can be implemented to ensure modifications can be made without modifying other code.
* **Updates:** Some tools exist like Renovate to allow the system to track and apply updates to libraries and frameworks. Additionally, it will be smart to implement backwards compatibility to ensure no updates completely break the system, and rollbacks are possible.
* **Access:** In order to best maintain the system, admins should have access to system monitoring, user management, content update tools, scheduling tools, and payment records.

#### 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?*

* **Login:** Users will be able to log in using their email/username and password combination, as well as use MFA or 2FA features like SMS or an authenticator application for better security. Password standards should be enforced, such as requiring symbols, numbers, upper and lowercase, and preventing duplicates or heavily repeated letters and numbers.
* **Secure connection:** It will be necessary to enforce HTTPS for all client-server communication to encrypt data in transit. Additionally, passwords should be encrypted using strong algorithms like bcrypt.
* **Brute force:** The simplest method for dealing with brute force entry is an account lockout after multiple attempts. The numbers can increase with attempts, (5 minutes, 15 minutes, 1 hour, etc), and the user can be notified through email or SMS to give them the opportunity to secure their account. Captchas like reCAPTCHA can also be used to slow attempts like this or prevent the method from being abused by bots. Additionally, IP addresses that exhibit brute-force behavior should be blocked.
* **Forgot password:** Password reset should be simple, using something like an SMS or authentication application to allow the user to reset their password. The password reset request can then be sent to the user’s email address to ensure this request is being sent by the user with access to the SMS and email of the account. New password standards should be enforced, with restrictions on using old passwords.

### 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 provide up-to-date DMV training materials including DMV practice tests and video tutorials
* The system shall support multi-user roles
* The system shall provide features for user profile management
* The system shall allow users to retake tests and quizzes
* The system shall allow users to search the database to quick reference DMV content in the library
* The system shall allow user to book in-person drivers training based on instructor availability
* The system shall send automated reminders through SMS or email to remind users of upcoming appointments and sessions
* The system shall use location tracking or zip-code input to ensure users are matched to the closest local instructors
* The system shall enable users to purchase training packages or pay for individual sessions using credit cards and bank integration
* The system shall offer fun promotions and discounts to attract new or existing customers that have not yet purchased
* The system shall send receipts to customers
* The system shall track user progress to show them how well they are doing before taking tests
* The system shall maintain records of all student performance for legal or regulatory purposes
* The system shall issue certificates of completion or graduation for online portions
* The system shall allow admins to manage accounts, modify content, and approve or deny refund requests
* The system shall monitor analytics and reporting such as user engagement and instructor performance
* The system shall allow users to rate instructors and leave feedback
* The system shall allow instructors to leave feedback on users and the system
* The system shall allow instructors to view and manage their own schedules
* The system shall allow instructors to securely message their students
* The system shall validate user credentials when logging in
* The system shall send real-time updates to users when DMV material is updated
* The system shall sync with local DMV databases

### 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.)?*

* **Needs:** The user interface should be simple and intuitive. It should be user friendly with a very low learning curve that caters to all levels of technical knowledge. All information should be easily accessible from one page, preventing the need to search for key features of the system through many different pages. The system should be responsive and work seamlessly through all environments and platforms. The system needs to adhere to WCAG guidelines to ensure accessibility to those with disabilities. It should also allow users to customize their dashboard with relevant progress to them.
* **Users:** The system should be prepared for a variety of users such as new drivers, instructors, administrators, and IT support.
* **Specific usage:** Students will need to be able to view data and modules, take tests, view progress, schedule and modify appointments, and search for information. Instructors will need to be able to view all students, a calendar or schedule for all appointments, modify appointments, and securely message students. IT and Admin will need all of the above, with additional access and ability to view chat logs, receipts, and user information on one dashboard.
* **Interaction:** Users will interact with the interface in a variety of ways as well. Some may use their desktop to study and take tests, while others might want to use their mobile phones to do so on the go.

### 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?*

* **Users:** I am assuming users will be ages 16-25. This is a younger generation that may be comfortable with technology but will still need guidance for navigation and payment methods. Instructors will most likely have some sort of technical proficiency, but they could also need guidance if they have not used systems like this and are used to over-the-phone or in-person scheduling. Most users will have some sort of mobile phone that they will most likely want to access the application through.
* **Internet:** I am also assuming that users will have reliable internet access when using this system, however it could be useful to allow downloadable modules and SMS updates for scheduling if the user is unable to access the internet.
* **Responsibility:** It is also assumed that users will be responsible for all account information and security of that information, and instructors will be responsible for maintaining their schedules and account information as well.

### 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?*

* **Offline:** Offline functionality will be a concern. If users are unable to gain internet access to use the secure messaging system or modify appointments, conflicts could occur.
* **Growth:** It is unknown what the eventual user-base will be, and it is difficult to assume how many users will eventually use the system, so scalability will always need to be understood, and user statistics monitored to prepare for possible growth.
* **DMV compliance:** DMV updates will require state-specific partnerships that could be difficult to acquire and could create a lack of uniformity in some areas.
* **Availability:** Instructor availability in many regions is unknown, and the system could be essentially useless to some if there are regions or areas where instructors are not available for hire.

### 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 project

Description automatically generated*