# CS 255 Business Requirements Document

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 build a system that enables customers to obtain a driving education outside of the normal DMV channels, via several driving sessions (depending on package selection).
* The client is DriverPass, a company founded/owned by a person named Liam. Liam wants his system to be able to schedule classes for drivers that are interested in obtaining driving classes, presumably for licensure, but not explicitly so- and practice driving tests.

### 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?*

* Customers should be able to schedule their own appointments/reservations for lessons.
* A User Interface is required, in the style provided by the owner, with several different pages, including at the very least one provided, and a contact information page for the company.
* The owner wants the system cloud based.
* The owner wants clients to be able to take classes and practice tests completely online if they wish to do so, with the option for practical/on-the-road training.
* Need a way to keep requirements/information current with DMV policies/regulations.

### 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?*

* Lessons should be 2-hour blocks, using the following packages, tentatively:
* Package One: Six hours in a car with a trainer
* Package Two: Eight hours in a car with a trainer and an in-person lesson where we explain the DMV rules and policies.
* Package Three: Twelve hours in a car with a trainer, an in-person lesson where we explain the DMV rules and policies—plus access to our online class with all the content and material. The online class also includes practice tests.
* The owner wants the ability to modify, or at least disable packages in real time without our support, and possibly pre-design other packages that can be swapped in/out at his behest.
* The owner wants a hierarchy-based security system with himself as the top tiered administrator, and tentatively- three more roles. One role for the IT officer, Ian, one role for the secretary, and one role for the end-users.
* Customers should be able to schedule their own appointments/reservations for lessons.
* Customer information gathered should be:
* First name
* Last name
* Address
* Phone Number
* State
* Credit Card #, Exp Date, CVV Code
* Customers should be able to schedule their own appointments/reservations for lessons.
* The owner wants a permanent data log that tracks all changes made to all records in the system. The log needs to be formatted in a manner such that the owner can understand the contents if printed for investigation purposes.
* Access to the DMV/DOT API may be required for updates.
* Customer’s drop off location should be the same as their pickup location.
* The owner provided a wireframe that he wants the user interface to resemble, including the following:
* Online test progress: (not taken, in progress, passed, or failed)
* Driver Notes
* Customer Information
* Customer and Driver Photos

## 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?*

* The system should be web-based, as the customer has expressed the desire to be able to access the system from multiple locations and/or devices.
* There is not a real-time function required (such as interaction with other entities in other locations in real time via the internet, much as one might find in a first-person shooter, for instance), therefore the latency of the server, while important in all cases for a good user experience, does not have to be extremely low. The server can likely be purchased with budget in mind, instead of performance, allowing the customer a plethora of options around the globe at different price points.
* While latency is not a major concern, scalability is. Virtually every country on the globe (with few exceptions) has a driving test for licensing, potentially making this product able to be sold globally, with modifications for language, etc. This makes scalability a key factor in developing the product, as it has the potential to be used in the entire state, but the entire world.
* Along with scalability, as mentioned above, reliability goes hand in hand. As the program grows, it will become more difficult and labor intensive to maintain. There will be a need for updates as driving laws and regulations change, which will differ between states, countries, and regions. Some will need to be updated manually, while others may be able to pull information using an API.

#### 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?*

* System should be multi-platform, as Liam expressed the desire for the system to be accessible from multiple locations and devices, portability is a must.
* System needs to have a database that is scalable on extremely large levels to keep track of user information such as personal info, reservations, etc., and licensing/test information for several regions/countries.
* System needs to be on a cloud-based platform that handles backups and security, the customer does not want to be responsible for those aspects.

#### 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?*

* Usernames will be the user’s email address to maximize simplicity. Input for login shall not be case sensitive, as email addresses are not case sensitive, this will make for ease of use for those less familiar with computers, such as the elderly.
* Tentatively (requires input from customer), the system should inform the admin of a problem anytime there is an error internal to the system that cannot be resolved by the user. There should be a separate support system for users that encounter issues such as forgotten passwords, logins, or technical support help.

#### 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?*

* Users shall be able to make, cancel, and modify their own appointments online.
* The secretary role shall be able to perform the same functions as the users, but for all users.
* IT officer (admin) role shall have full access to all aspects of the system.
* Changes made to user information shall not require code modification except in rare circumstances involving platform updates, as the system shall be designed for maximum portability pre-launch (multi-platform engine).

#### 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?*

* The customer has explicitly expressed that they do not want to handle any aspects of security, and thus the cloud computing company they decide to work with shall negotiate all aspects of security with the customer. That being said, subjectively, it would be ideal to use email verification and 2FA for login, with an IP address ban after X amount of failed login attempts. There should be a “forgot password” button, which routes the customer through a series of information verification.

### 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 scheduling, modification, and cancellation of appointments.
* The system shall provide practice tests for users, which shall be available online.
* The system shall provide classes for users, which shall be available online.
* The system shall allow users to login once their account is validated.
* The system shall allow users to reset their password when necessary.
* The system shall keep a log (length/time TBD, must consult customer) of all changes made in the system to user information, or reservation information by all entities.

### 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 best option for user interface, given multi-platform portability, would be a website that supports several different browsers (Brave, Chrome, Edge, Firefox, Safari, etc.).
* The website should allow the user to download the tests and/or classes directly onto their device for offline use once they have paid (must consult with customer whether test and classes will be subscription-based, pay-per-use, or one payment for all access). This will reduce overall system traffic by a large margin, resulting in lowered costs for the customer, and therefore likely the users.

### 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?*

* The system assumes that users have a valid email address and phone number, or the ability to obtain one of the two, in order to eventually obtain an email address used for login.
* The system assumes that users have access to a computer, whether PC, laptop, tablet, or mobile device.
* The system assumes that users are not blind. It is highly unlikely if not impossible that people without eyesight will need access to a system designed for training drivers for licensing, and therefore the system does not need to have accessibility for the blind.
* The system assumes that users have internet access.

### 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?*

* Limitations will require further consultation with the customer, and are as follows:
* The system will require scalability on some level, and the rate at which it scales will be limited by budget, staff, and time. These items must be discussed and fleshed out with the customer.
* The system can have built-in redundancy providing zero-downtime maintenance for maximum user access, or it can have scheduled outages for system maintenance. This decision will also be limited by budget, and/or the necessity for system access at any given point depending on the scale of the system, and the user volume during peak hours and off-hours.
* The system may be limited regarding interaction with organizations such as the DMV, as they may have confidential information that they’re not willing to risk using an API for automated updates, as this could potentially cause system vulnerabilities.

### Gantt Chart

*Please include a screenshot of the GANTT chart that you created with Lucid chart. Be sure to check that it meets the plan described by the characters in the interview.*A chart with multiple colored rectangles

Description automatically generated