# CS 255 DriverPass 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 assist drivers in training with preparing and eventually passing their driving tests at their local DMV.
* The client is Drivepass, they want their system to allow end users to take online classes and practice tests. They also want the end user to be able to schedule on the road training if they wish.

### 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 wants the system to provide online driving classes as well as practice tests. Driverpass also wants a scheduling system implemented where end users can schedule an appointment/lesson.
* The problem that needs fixed is the rate at which drivers fail their drivers tests. Drivers need better training before heading to the DMV to attempt to pass their drivers test.
* Different Components:  
   - Scheduling System (create,cancel, modify appointments)  
   - Provide online practice Tests  
   - Reservation tracking  
   - User Permissions system  
   - Cloud based architecture

### 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?*

* Once completed the system should be able to:  
  - Provide drivers with online materials/ practices tests/ and courses  
  - Allow users to schedule an in person practice session (or cancel/modify)  
  - Allow Driverpass (Liam) to access the data anywhere and edit it (online only)  
  - Allow Driverpass to download reports and info   
  - Assign roles to different users across the company  
  - Allow admin users the ability to reset passwords and revoke access  
  - Tack who made a reservation, who canceled, and who last modified  
  - Stay up to date with DMV requirements  
  - Remain cloud based (only editable online to avoid redundancy)
* If all the above is implemented, the system should function as a scheduler (that allows ussr to make, cancel, or modify appointments), an online learning resource that allows users to take practice exams, and a database that Driverpass can use to pull 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?*

* The system needs to be web based (a website)
* The system should be responsive and not overly bloated to allow it to work on most hardware
* The system should be well maintained, and updated as new features or defects arise

#### 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 in the cloud
* The databases required for the back end will also be housed in the cloud and will not need built out locally.

#### 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?*

* When creating the account, the user will create a username and password (case sensitive). This will be used to authenticate.
* The admin should be alerted if too many attempts have been made with an incorrect password (say 10)
* After too many incorrect attempts, the user will be asked to reset their password

#### 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?*

* The system will allow modifications to be made without needing to change code
* The system will allow users to update personal information, and update their credentials
* The system will apply updates globally when new features are implemented or bugs are fixed
* The IT admin will need full access to the system (users personal information should be encrypted and not accessible)
* The IT admin should be able to make system changes and perform password resets and lock/unlock accounts

#### 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 user will need to enter their username and password to login to the system
* These credentials will authenticate over the cloud
* If an account has too many failed login attempts, the account will lock
* The only way to get an account unlocked will be to get an admin to unlock it (via call)
* If the user forgets their password, there should be a reset password link that sends a password reset email to the email they have on file

### 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 validate user credentials when logging in
* The system shall provide practice quizzes and tests
* The system shall allow the user to schedule an appointment
* The system shall allow teacher/employees to confirm appointments

### 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.)?*

### 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?*

* We assume that the user has access to a computer/ smartphone
* We assume that the user has access to internet
* We assume that this build will not go over budget (no budget was listed)

### 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?*

* The limitation I see here is not having a budget listed, it may be difficult to allocate resources appropriately without a specific budget in mind
* We do have a timeline, so abiding by the timeline is considered a limitation
* Since we want this to be based online, that is another limitation – as we must make sure the system is light enough to be ran in all browsers

### 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.*

