# 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?*

* DriverPass will be a service where clients should be able to take online classes and practice tests. The course materials will prepare students for the test at the local DMV.
* Their company is owned by Liam, who has an IT officer named Ian.
* DriverPass will also offer on-the-road training.
* Liam and Ian want us to build a system that can offer these services to their customers.

### 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?*

* There is a current need for better driver education offerings. Many students are currently failing their drivers tests which prevents them from being able to get their license.
* The solution that DriverPass wants to offer is more resources to prepare these students for their tests, at a price. An important part of these offerings is that they are readily available for users wherever they are.
* Test should prepare students for passing the actual driver’s education tests.

### 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?*

* Customers should be able to access their data from any location, so it should be stored on a secure server that always runs and is always connected to the internet.
* They should also be able to access their data both online and offline, meaning they need to be able to download reports.
* DriverPass needs to work on both desktop and mobile, meaning we may need to prepare a mobile version of the site.
* Security involves offering administrative rights to the DriverPass owner, and some extended access to the driver pass employees. Customers would be restricted to accessing their data only. Liam, the owner, would control permissions for all users including employees. The DriverPass employees would be able to assist customers with password resets and support issues, but not control each other’s permissions.
* The system will have a history log of any changes made to the reservation database to help offer visibility to the owner and hold employees accountable.
* Customers should be able to create, edit, and cancel calendar reservations for a day and time to take two hour driving lessons. They should be able to make a reservation online or in our offices. If they come into our offices, their employees will need to enter the reservations for the customer which is why tracking is important.
* They have drivers who go with the students and instruct them. The system needs to track which of our driving instructors is matched up with what car, student, and reservation.
* The system may need to include a phone line for the secretary to take phone calls.
* There should be three packages that users can select from that increase the amount of training offered and increase in price. The DriverPass employees should be able to modify these packages, add new ones, and remove them in the future if they like. The starter packages are:
  + 6hrs w/ trainer
  + 8 hrs w/ trainer, in-person lesson
  + 12 hrs w/ trainer, in-person lesson, access to online class, content, and practice tests.
* The system should be able to store several fields of customer data such as name, address, phone number, payment info, drop off/pickup location, etc.
* There needs to be a password reset option for customers.
* There should be a link to the DMV website so that DriverPass is made aware of any updates that need to make to their curriculum.
* They would like the system to be cloud based so that the DriverPass employees don’t have to deal with server maintenance or worry about data loss.
* The software should have a user interface that has already been sketched out by Liam. This includes sections of the page that are meant to display information such as online test progress, user info, driving instructor notes, driving instructor photo, student photo, and special needs. There should be a nicely centered logo at the top. The driving instructors' notes should include a table with columns for lesson time, start hour, end hour, and driver comments.
* There should be input forms for when the students are first registering so that they can enter all their information including a way for us to contact them.
* There should be a contact us page so that customers can get support and 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?*

* DriverPass should be accessible by all internet connected devices both desktop and mobile. The best way to do this would be to make it a web-based service hosted on a cloud server like Microsoft Azure.
* This would allow users to access DriverPass on a browser of their choice at any time. DriverPass would always run at peak speeds because cloud services increase allocated resources automatically as they are needed.
* It also saves us the cost of having to build, maintain, and update our own servers. Cloud servers reduce the risk of a server outage due to natural disasters because the cloud servers themselves are dispersed in several different low risk locations.

#### 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 be able to run on any platform that the user has if we make it web based. If we host the service in the cloud, we will likely end up using something like Microsoft Azure, Amazon Web Services, or Google Cloud. Most of these cloud services use some form of Linux, so designing the system to be Linux friendly is a good start. Most cloud servers offer the ability to create a virtual machine using the operating system of your choice to run a server. For example, Azure allows you to use Linux or Windows Server.
* A database will be needed to store data such as customer information, appointment times, and driver information. This database will need to be able to display data to customer devices securely via our web application.

#### 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?*

* Users will have an internal ID number that is stored within an instance of a user object. Whenever a user is created, a new instance of the user object will be created with a unique identifier that is only visible to administrators and the database. Users will establish a unique, case sensitive username on profile creation, as well as a case sensitive password with some minimum requirements. Users will need to provide an email address for password reset on account creation. All of these things are stored within their instance of a user object. When a user logs in, their username will tell the system which user profile to pull from the database.
* Administrators should be informed whenever a profile is subject to repeated failed login attempts. Typically, a user will need assistance getting logged in when this happens, or they could be subject to malicious login attempts.

#### 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 have special user objects that have administrator access to the system. These administrators will be able to access any user profile and change any information needed. Each user object will come with built in setters and getters that will allow both the access and modification of data contained within that instance of the user object without any code change.
* The system will adapt to platform updates as needed. There may need to be minor modifications in code, but typically a major platform update to something like Windows Server that would affect services running on that server would be communicated to us ahead of time.
* IT would need full admin access to user profiles, and full access to the database and cloud server. IT will need full visibility on the system to ensure there aren’t any issues in the infrastructure or backend.

#### 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?*

* Users will need to provide a unique case sensitive username when logging in. They will also need a case sensitive password. These should be established on user profile creation by the user.
* The REST API would be an excellent tool to handle transfer of login credentials securely from the user to the service running in the cloud. REST API supports basic authentication over HTTPS.
* Every system should come with a lockout feature that locks a profile for a certain amount of time after a certain amount of failed login attempts. This is how most software companies prevent their users from being subject to a brute force attempt.
* If a user cannot remember their password, they should be able to reset it using the password reset email that they provide during account creation. The email should be required to create a profile.

### 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 ask users if they have already created a profile or not. If not, it will collect a new username, password, and password reset from them. A new user profile will be created from this information, and access to the system will be then granted.
* The system shall collect a username and password from all users accessing the system and validate them through the REST API with what is currently stored in the database.
* The system shall allow the download of reports so users can view them offline in a spreadsheet.
* The system shall allow administrators special access to view and change any user profile or appointment.
* The system shall create a log that shows the history of any changes made to customer reservations and present them in a report.
* The system shall allow standard users to create, modify, and delete calendar reservations for DriverPass Lessons and select from a variety of lesson packages.
* The system shall allow users to select which instructor and vehicles they would like to use for their reservations.
* The system shall allow for creating, modifying, and deleting package offerings.
* The system shall have the ability for users to modify their user profiles and add drop off/pickup locations to them. Also, the ability for them to change their own password and correct any other information that may be outdated.
* The system should pull instructional documentation directly from the DMV website to ensure the most up to date information.

### 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 interface needs to display a logo for DriverPass near the top. It should also display modules that show details for online test progress, user profile information, driving instructor notes, special needs, driving instructor photo, and student photo. The instructor notes module needs a table that shows lesson time, start hour, end hour, and driver comments for each session.
* Each user will need to be able to select packages and schedule dates on their calendar for sessions. They should be able to modify and delete these through the interface as well. Users should be able to modify their profile information as well and add their own comments to the special needs section.
* Users will be presented with the interface after logging in through a web browser, either via desktop or mobile. We may want to have a different layout for mobile devices; however, it should work on both regardless.

### 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?*

* I am assuming that all DriverPass users will have devices and a stable internet connection. Without those things users will not be able to access our services unless they call the offices for everything they need.
* I am assuming that DriverPass will have an IT employee who can handle the cloud server and database.

### 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?*

* One limitation that we have is time, as we are giving ourselves a little over three months to establish an entire system. Things can go wrong, and we need to be able to adapt quickly to any unforeseen challenges.
* We need technicians and developers that are familiar with cloud computing services.
* They need to be able to afford the cost of something like AWS or Microsoft Azure. They are becoming more and more affordable, and they scale with your resource needs.
* Users must provide us with an email address to use our online service.

### 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.*

