# 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 for this project is DriverPass. They are a newer startup looking to transform the way people obtain their driver’s license. Currently they believe too many people are failing their driving exams. They aim to remedy this by providing online curriculum for the written portion of the exam, and scheduling drivers training with their team of driving instructors for the practical portion.

### 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 has several needs for their new system these include

1. Data/Customer (CRM) entry.
2. Product and offering creation
3. Online booking
4. Online content delivery
5. Reporting
6. Checkout/credit card processing
7. API to DMV for updates on rules.

### 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?*

When completed the aforementioned modules need to be able to complete the following tasks

1. Administration
   1. User creation and administration for internal users
   2. Automated account creation for external users
2. Data/Customer (CRM) entry.
   1. Create customers with all relevant data
   2. Track customers progress in online and in person driving tests/education
   3. Track notes left by instructors.
3. Product and offering creation
   1. Initally setup three tiers of products
   2. Ability to turn product offerings on or off
   3. Possibly add ability to create new or custom products on the fly
4. Online booking
   1. Customers need to be able to book a driver or test when time slots are available
   2. Customers need to be able to check these reservations and mofiy them if needed
5. Online content delivery
   1. Educational documents and presumably videos need to be able to be delivered to customers who purchase certain product packages
6. Reporting
   1. Management needs to be able to see which customers are scheduled for when and -
      1. What they are scheduled for and with which driver
      2. Notes left by instructors
      3. All updates to reservation by user
7. Checkout/credit card processing
   1. Must take payments for different product offerings
8. API to DMV for updates on rules.
   1. The DMV needs to be able to send over updates when they make changes to policies and proecdures.

## 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 will run in the cloud. The db portion will run on AWS and the actual website will be hosted on a 3rd party web hosting service
* The system needs to run in real time
* The system will not need to be updated as it is mostly a simple CRUD architecture.

#### 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?*

* Yes the backend will run on a database in AWS and a Lambda server. The type does not matter, but for this application we will run SQL Server on RDS, and run python in Lambda.
* The AWS MsSQL database runs on Amazon Linux distributions. Since we are going through the cloud we are only exposed to the server endpoints and the operating system doesn’t matter.

#### 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 will have different permission sets depending on what type of user they are. This will all be determined in the user table.
* The input for user name will be case sensitive
* Error logging is important and we will have to setup a notification system for admins when the system encounters and issue. One of the big ones will be when AWS has any kind of downtime or maintence.

#### 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?*

* Yes users can be modified through direct db manipulation or setting up a trigger in lambda.
* There will not be platform updates generally as we are in the cloud. As long as we maintain good SQL standards we won’t face issues. We will pick a version of python when setting up Lambda and stick with that throughout the entire systems lifecycle.
* The IT admin team will just need access to the sQL server endpoints and the Lambda API.
* We can create a few scripts in python to do things like add/remove/modify users if there is an error or update an order etc.

#### 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?*

* Leveraging AWS will give us a huge advantage in security.
* The focus on security will switch from us to them, and they are industry leaders in security.
* We can setup our web cart to utilize https for extra security.
* All connections to either SQL or lambda endpoints we can setup to use tokens on both sides and log all connection attempts.
* If a user forgets their password they just click the forget password button on our website, this will trigger a lambda function to query that email from the db and send out an automated email for password recovery.

### 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 be able to create users
* The system shall validate user credentials when logging in
* The system will let users reset their passwords
* The system shall let internal users create customers
* The system shall let internal users create note about customers
* The system shall let customers book appointments
* The system shall let customers modify or cancel appointments
* The system shall deliver online content to customers
* The system shall be able to deliver reporting to internal users
* The system shall let a customer purchase a product package through the website via credit card
* The system shall ingest data from the DMV’s website and update content/copy based on ingested data

### 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 Ui will need to start with a log/sign up page
* From here customer users need to be able to purchase content packages, view that content and schedule driving lessons
* Internal users need to be able to see their scheduled lessons and make notes on a customer’s progress
* Admin users need to be able to run reports, and view logs.

### 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 assumed the customer is OK with using AWS and can afford it. We also assume that the customers are going to be able to maintain and administer their own website after we launch it. We are also assuming that the customers are computer literate enough to navigate to this page, sign up, make a payment and then book their own driving lessons.

### 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 system is limited in that it is built entirely in AWS. This is a great thing as it relieves us of a lot of back-end setup work. That said if the business ever decides to move to another system migration may be rather difficult and a lot of setup work would be required.

### 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.*

[Insert chart]