# 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 is to create an online platform to host classes and practice tests to help new Drivers pass the driving test. This includes scheduling/handling appointments for in-person trainings.
* DriverPass is the client
* This will be a web platform, preferably cloud based, with online classes/tests and scheduling for in-person appointments. They are also looking for record keeping/downloading, ability to track changes, accounts with the ability to manage it, and multiple webpages.

### 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 is looking to solve the problem of many new Drivers failing the driving test.
* As such, they want the system to help them facilitate improving the pass-fail ratio of the test through testing and classes.
* The system will need to have webpages with multiple UI elements connected to at least one database. This database(s) will handle accounts, storing pre-made courses, appointments, and user results. It will also have a security system and ability to make changes to the database information and the appointments/modules.

### 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 will be able to:
  + Create and edit in-person appointments
  + This includes the ability to update and add/remove/hide modules and packages
  + Store, access, and update classes
  + Create, store, access, take, and update tests
  + Store and access results, including downloading them
  + Create, access, control, update, and maintain accounts
    - Must include the ability to automatically reset passwords if the user requests
  + Multiple access levels for accounts to increase security
  + Track changes, specifically appointments
  + Connection to the DMV rules/policy system
    - Notifications for staff when there is an update
  + Website on a cloud-based platform
    - Multiple UI elements based on the design drawings
    - Page with contact information for the team and student

## 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 will be required to run on web with an application available.
* The system should be updated every day unless certain actions happen
  + A class is changed
  + A user updates their profile or completes a class/test
  + An appointment is made or updated
* The system won’t need many resources or be extremely fast. It will need to be fast enough to run classes and test though.

#### 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?*

* This system will be able to run on any platform that supports web applications
* It will require a back-end database to hold the various tests, accounts, classes, appointments, etc.
* There will also need to be a server to run the website and services.
* Additionally, there will need to be an API set to communicate between the different components and the DMV.
* Security service should also be running on the backend to ensure the data and system is safe.
  + This should either include or be paired with a secured/encrypted section of the database that holds the password.

#### 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?*

* An account system will be the easiest way to distinguish between users.
  + A username/email/ID number would help id users.
  + Passwords would be stored encrypted can be used to help validate users.
* Input won’t be case sensitive except for inputs like passwords.
  + This can be beneficial to help fight input error that users may make and ease their frustrations if something is hard to type.
* Admins should be informed:
  + Any time an error or dangerous warning is detected by the system
  + When the security system detects an attack, threat, or access violation attempt
  + Multiple incorrect password attempts
  + Changes or viewing of accounts made by admins. This is to ensure customer data is secured in the event of an admin account is breached.
  + Multiple quick changes to classes/tests/appointments by one user

#### 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?*

* Making changes to a user is possible. It is important for admins and users to be able to make changes to their account information.
* The best way to update a system is to compartmentalize different parts of it.
  + This will make it so the system can stay online while the updates are made
  + It will also make it easier for the dev team to update pieces of the system without making extensive changes to other parts of the system.
* The system should update outside of peak hours with the exception of critical updates.
* The IT admin will need to be able to access almost all parts of the system.
  + This will include logs, customer accounts, database, force resetting passwords, and security service
  + The IT admin shouldn’t need access to customer PII, plain text passwords, the code base, or payment information.

#### 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?*

* A user will be required to provide a username/email and a password to access their account.
  + I would also recommend multi-factor authentication, but it isn’t required
* Using TLS encryption, checksums, user authentication (depending on the action), and ensuring connections are closed can help ensure connections are secured.
* Attacks on user accounts should always be considered. In this event, the system will:
  + Send a notification to the admin and IT team
  + Lock the user account
  + Force a password reset
* If a user forgets their password, they will have the option to recover their password through email.
  + If MFA is used, they will also be required to use it during the reset process to help authenticate the user
  + A user’s account will be locked and force a password reset if they enter an incorrect password too many times

### 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 allow users to view and make appointments
* The system shall allow teachers to view, create, and edit appointments
* The system shall allow users to enroll in, take, and view classes and tests
* The system shall allow teachers to view, create, and edit classes and tests
* The system shall allow users to change their passwords
* The system shall allow users and admins to update user information
* The system shall allow admins to update access controls
* The system shall monitor access control and errors and notify admins about issues
* The system shall provide secure connections between the user, the system, and back-end services
* The system shall provide admins and teachers to remove, disable, and update packages
* The system shall log events for follow-up
* The system shall allow offline access
* The system shall prevent offline edits to ensure data collisions don’t happen
* The system shall provide account security to ensure users and admins are secured

### 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 show:
  + Test Progress
  + User information
  + Special needs
  + Driver Notes
  + Driver’s photo
  + Student’s photo
* It will also need to have the ability to show and make appointments, view and edit the user profile
* Admins will also need to have an interface to view backend components, logs, website information, and user accounts
* There will need to be different user groups: Admins, IT, Teachers, and normal User
  + Users will be able to edit their profiles, access tests and classes, view their progress, makes appointments, and change/reset their passwords
  + Teachers will have all normal user abilities, but will be able to update, create, and remove classes, tests, and package information.
  + IT will have all the above but won’t be able to change other user’s information through the UI. They will be able to view logs, edit website information, edit some backend information, and reset user passwords.
  + Admins will have the ability to view, edit, and create/remove everything through the UI

### 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 will have the ability to encrypt user passwords, PII, and any payment information
* The system will have support through IT and admins
* The system will be scalable to accommodate increased user and content load
* The system will have an application to enable offline use of the system
* The system will have a similar UI experience across the website to help the user experience
* The system will be easy to use
* The system will be able to be expanded to accommodate future expansions and topics.
  + This will allow DriverPass to pivot when new topics or areas of failure become known
* The system and services will need to be lightweight to preserve resources and improve performance

### 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?*

* With our system, we could have a limitation with the application and website combination.
  + Websites are generally not available offline. As such, there will need to be an offline component like an application. This could require additional time and resources to design the interface and functionality of an application.
  + Applications are much different in terms of user experience, systems, and security requirements.
  + This application would need to cover multiple Operating Systems.
* Interfacing with the DMV could prove to be a limitation.
  + This would require communications and negotiations.
  + States have different requirements for their DMVs
  + State laws can differ from each other, as such requirements would need to be understood for each state
  + State governments may not allow interfacing with their internal systems
  + There would need to be 50 different reading algorithms to retrieve updates to the DMV driving requirements.

### 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 drawing of a graph

AI-generated content may be incorrect.

