# 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, a company that provides driving students with both online education and practice problems to help them pass the licensure test and offline one-on-one driving lessons to help them pass the road test. The main purpose of this project is to design a web-based system that allows interactions between DriverPass and its customers as well as features that allow leadership roles to manage data and access.

### 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?*

* For the backend (company-side interface), the system should be running “off the web, preferably over the cloud”. It should also provide administrators offline access to review and store necessary information. Database modification privileges should be available for the IT officer to allow password or information update for customers.
* For the frontend (customer-side interface), the system should allow new users to register account, including their detailed information like name, address, phone number, and payment information. It should also allow the registered users to choose their desired package. From the interview, there are 3 packages provided by the client, and they all should have the ability to be nullified in case the client no longer wants any new users to sign up for the package. The system should also be able to connect the user with a dedicated driver for in-person lessons, and include a webpage with an UI that looks similar to the one DriverPass provides that allows users to visualize their current progress with online tests and notes given from their drivers. Lastly, the system should be connected to the local DMV’s server to receive any updates on rules and regulations.
* The main problem DriverPass wants to fix is that many students are suffering from failing the licensure exam and not able to find appropriate study guides. DriverPass provides these students with easy access of online education content and practice tests that are update to date with the local regulations, as well as in-person lessons to help them pass the road test.

### 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?*

* The system should be able to:
  + Allow administrators to download necessary information for offline use
  + Allow IT officer to manage user access and update user information
  + Allow students to create account and choose desired package
  + Packages should have the ability to be nullified in case DriverPass no longer wants any students to enroll in a specific package
  + Provide a platform to contain online educational content including practice tests
  + Provide UI that allows students to easily visualize their current progress
  + Connect students with their dedicated drivers and make/update/cancel appointments
  + Any appointments made, updated, or cancelled should be easily tracked by the company
  + Connect to DMV’s server to receive up-to-date information about policies and regulations
* Some measurable tasks would be:
  + Whether the system allows downloads of its current database
  + Whether the system allows modifications of its current database
  + Whether the system and the UI provide direct access to transform user input into entries that are sent to the central database
  + Whether the hyperlinks for the package choices could be removed, or whether the UI for it could be updated to show “not available at this moment”
  + Whether the system can run an online educational system stably and transform user progress into trackable data
  + Whether the system allows transformation of user input into calendar events, and whether the system allows multiple traffics accessing the same calendar events and only execute one of them and return error code for the others
  + Whether the system establishes a connection with the DMV’s website to tail updates on policies and regulations and pushes any updates as notifications to the DriverPass IT department

## 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?*

* All functionalities should be available under different environments (PC, Apple, Android, webpage, etc.). This will allow users to access the system easily on any device.
* System should be running fast enough to avoid delays and waiting periods on the user side. To achieve this, minimal cache should be preserved
* System should also be able to handle multiple requests sent at the same time and appropriate responses should be returned. This ensures the system is capable of handling large number of requests and returning proper responses to improve the stability and availability
* The system should be updated periodically—preferably scheduled for a monthly basis, and could add urgent patches or make ad hoc changes. This is to ensure that the system administrator reviews all errors and bugs and provides constant updates to the system to consolidate the usability and stability

#### 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 platform should be designed under Unix environment since it is the most common environment used in the development phase, but the final product should be able to run on multiple platforms.
* Preferably, if there is no financial issues, cloud relational database would be the preferred choice to have constant backup of data in case a catastrophic event happens and allows multiple users interact with the database at the same time

#### 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 should be assigned with different roles. With the help of relational database, this could be easily achieved. With SQL language, roles could be easily assigned by the administrator to give specific privileges. For example, a developer could have most accesses to the database but cannot modify any privileges of other people; on the other hand, a user should only be given a read-only access
* Whenever any roles have a tendency to carry out , or have already carried out actions that do not fall under their umbrella, the admin should immediately be involved and update the system to patch the glitch and revoke the unwanted access for that role

#### 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?*

* Most of the privilege and access changes can be achieved on the developer side UI, but eventually these are all changing the code at the backend. Proper logs will be documented to have tracking information of any changes
* Since the system is doing a routine health check at least monthly, the system can easily adapt to platform updates by adding immediate patches
* The IT admin will need to have the access to grant or revoke privilege of most users including developers, tester, and users. The IT admin is also responsible for tracking down the logs and find the root cause of any privilege violations due to system glitch

#### 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?*

* Just like other major website, credentials including username and password would be required for the user to log in. a two-factor authentication could also be issued based on user preference to enhance security
* Client’s data can be transmitted to an on-premise database first. Such data will be verified on the on-premise database and then sent to the cloud database. This is to create a double security in user data and avoid any user violations happening
* An account should be suspended and notifications should be sent out if hacking attempt happens. An automatic system check should also be carried out when such event happens to ensure the integrity of other data
* If the user forgets their password, then a password retrieval tool can be used to reset the user’s password. User will need to key in the answers corresponding to the security questions they picked upon account registration and if a two-factor authentication is disabled, the user will be prompted to enable TFA to avoid future burden

### 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 enable users to login with correct credentials and reset password if forgotten. This is to ensure the correct user is logged in to match with the correct profile.
* The system shall deliver the most updated progress of the user as of where they are standing with their online and in-person training
* The system shall allow users to choose the package they think is the best fit for them.
* The system shall allow users to schedule in-person driving classes with trainers

### 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 should at least include the following: company’s logo, login/sign in window, forget password hyperlink. Once logged in: user name, current package chosen, current progress with online course, calendar to schedule in-person driving class, test scores.
* The main users for this interface are those who are seeking more convenient means of learning how to drive. When glitches happen, developers and testers could also be the users of this interface
* For learners: they should be able to choose the package they like, add payment method, pay for their package, check on the progress of their online courses, schedule in-person driving classes, and view their test scores
* For developers and testers: should be able to regenerate bugs detected and look at the backend log to fix the bug
* For PC software or web version of the system, the users should be able to click on the highlighted hyperlinks or textboxes to proceed with the next step. For mobile apps, the users should be able to tap on the appropriate area to gain access to the next step

### 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 UI is set up with a preliminary system and a stable database
* Resources available for such UI design is infinite and any revisions could be performed impromptu

### 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?*

* Main UI design is heavily based on hyperlinks towards other sites
* Graphics and sounds are yet to be considered

### 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.*

