**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 believes it is their duty to help users all over the world struggling with the driver test which is required to obtain their drivers license. They'd like their program to offer a support system to drivers in need or struggling with preparation for said test.

**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 offer training packages to their users to help with preparing for the driver's test. They would like to fix the worldwide issue of beginner drivers failing to pass the driver orientation test which is required to obtain their drivers license. The different components needed for this program to succeed would be the program the user utilizes to book training sessions and the program the owner and staff use to remove users and add/remove driver training packages at will.

**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?*

* A user should be able to buy a package, and then utilize this package by scheduling sessions with a trainer, having access to online classes, and practice tests. The staff should be able to monitor movements of the users such as making, modifying, or canceling a session. For this we will need a login component, user/staff roles, notification component so staff can track user movements, online class and practice tests components, and reservation scheduling, modifying, and canceling components.

**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?*

* This program will most likely be web-based as it'll make it easier for the user to access the program from both their phone and computer. The system should run quite fast as all it'll need to handle is classes and scheduling appointments. How often the system will need to be updated will be entirely up to the DriverPass team as he stated he would like the ability to add and remove packages meaning it'll be updated whenever he deems necessary.

**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?*

* Windows would probably be easiest as it is most well known, and the backend will require a database as we will need to store user login credentials as we need to be able to know which users scheduled which appointments and what classes said user is taking or has passed.

**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?*

* Yes, usernames and passwords should be case-sensitive to distinguish between different users. Adding user roles (Admin, Staff, or User) will also help to tell different users apart and what that user should be able to do on the platform. The admin should be informed of an issue if a user schedules multiple appointments on the same day or if a user has bought multiple packages as it might be duplicate or fraud purchases.

**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, adding, changing, and/or removing users shouldn't require any changes to the code as I imagine they would be dealt with in the original program and would only require changes to the database and not the program. If the system was designed from the start to leave room for improvement it should have no issues handling updates but if it was poorly designed from the start it might have some issues adapting. The IT admin will most likely fill the user role Admin as it'll have the most capabilities in the program.

**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?*

* For initial sign up a user will moist likely be required to provide first and last name, email, username, and password. After that, the user will only need to use the username and password to sign in. You'd secure the data between client and server by serializing it in a way where even if someone got access they wouldn't understand it without the key. If someone tries to hack into a user's account they'll most likely be notified through the provided email to change their password. Same for if a user forgets their password there will most likely be a "forgot password?" option at sign-in to handle cases like this. Clicking this feature would result in an email being sent to help the user change their password.

**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 allow users to create an account
* The system shall validate user credentials when logging in
* The system shall allow the user to modify or delete their account
* The system shall allow the user to attend online classes to help study for the driving test
* The system shall allow the user to take online practice tests
* The system shall allow the user to purchase a driving lesson package from various options
* The system shall allow the user to set up driving lessons with a driver
* The system shall allow the user to modify or cancel said sessions with the driver

**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 requirements for the user interface should be to be able to easily navigate and understand its controls to make the user experience as easy and carefree as possible. As of now I know of two different types of users and that would be customer-user and admin-user, the customer-user would have the widespread UI that lets you do things the program is intended for and the admin-user will have permissions allowing them to be notified when customer-users schedule, modify, or cancel appointments. Users will interact with this program through a web browser meaning they could either do it from the computer or a phone.

**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 left out all the mechanics/features a customer-user has as we have mentioned them multiple times before. I'm assuming the user has a computer and/or phone in order to access the program.

**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?*

* If it is only Sam and Jennifer working on the program our limitation might be time as that is a lot of work for two people to accomplish. Another limitation might be that cars and drivers are willing to offer lessons. As the owner stated they only have 10 cars meaning they can only take so many appointments at once and that's excluding the fact that you'll need drivers willing to offer lessons in those 10 cars.

**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.*