**Sheehan 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 of the project is to design an all in one driving test assistance portal. The client wants to make a portal where people can take online driving classes, register for real life road training with a teacher, all while being compliant with the DMV. The client wants their portal to be an easy place to do all facits of drivers training.

**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 to be the most convenient way people can do their drivers training. They want to fix the problem of a lack of online resources for road tests. To do this they have multiple components that will need to meet DMV standards to be accreditied. The portal wants to include real life scheduling, curated online trainigs, and specials modules to equip new drivers to more conveniently pass their road 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?*

* When the system is completed drivers should be able to purchase training packages, scheudle real life driving training, view and practice web driving modules to learn and practice their knowledge. The measured tasks of being able to do this in system design should be a boolean value of 'does user have access' I envision the UI as a 'unlock more when you pay, or upgrade your plan to view these now.' This 'conversion' monetization could also be a useful metric for measuring the success of the system we create.

**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 cloud based and usable from a web-based application. This way we will be able to ensure real time access our users, inluding booking lessons, paying, and pushing new features in terms of lesson models. With cloud based models it will be an ondemand service keeping latency at a minimum even at scale, with cloud based models we can also frequently update our application at odd hours to reduce customer inconvenience.

**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?*

* Since we want to use a web based cloud application the service should run on all platforms to maximize profits. We will need a relational database to facilitate this as well as an api to make communicating between different platforms and various databases easier. The database will need to store completed training models, user data, reservations, and payment history.

**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 login credentials not visible to them. We will have rolls like admin, teacher, student, to validate all users sign ons we will use modern best practices. This includes hashing secure values, and case sensitive inputs mattering to increase incyption. Admins should be informed of suspicious logins, sercuity concerns, and also bugs that are impacting buisness operations. Teachers another class of users should be notified of scheduling issues, payment issues, lesson creation issues.

**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 code should support adding, removing, and modifying features without acutal code changes, this should be features users can intuitvely do through the UI. The system will handle platfrom updates by having the update pushed during odd hours, and scheudled maitenence to reduce the impact of downtime on service. IT and admin level rolls will have full access to managing user's data, configuring packages and payment options, system logs, and API data to better allow their troubleshooting and increase of features.

**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 have case sensitive data as previously discussed, MFA will also be required to increase security and make brute forcing user info more difficult. We will use industry standard encryption in SSL or TLS to protect our users data and the exchange of data on our systems. On mulitple failed login attempts, a consequence of brute force attacks, the users account will be locked and their contact methods will be notificed of failed logins, as well as the IT team. If a user forgets their password they can have their primary contact method recieve a link to reseting it, by answering a security question, and veryifying their username users can reset 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 validate user's credentials when logging in, and requrie MFA to complete sign on
* The system shall allow student class users to schedule, reschedule and cancel driving lessons
* The system shall allow users to view real time updates, and courses for drivers training.
* The system shall enable instructors to view and edit schedules, update lesson plans, and provide feedback to students.
* The system shall allow student to track their progress through completed lessons, and instructor feedback.
* The system shall provide admins and IT to manage user roles, lesson plans, and pricing.
* The system shall allow users and instructors to generate lesson history, user activities, and financial transactions.
* The system shall enable automated messaged to students reminding them of incoming lessons, and allow them to edit those lessons.
* The system shall allow secure payment transactions.
* The system shall generate log info to allow for better troubleshooting.

**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 needs to have multiple user classes. Students, instructors, and admin, all users should have different dispalys of the UI.
* Students need to be able to schedule lessons, view their progress within their course, acess the learning material they are paying for, and make payments.
* Instructors will ened to be able to assign, update, and view lessons. They also will need to be able to view their students and provide feedback.
* Admin's interface should include managment of user feature's, logs of various data, and financial transaction data.
* The UI should be supported accross all web based desktop and mobile browsers.

**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?*

* Users have internet access, if they do not they can't access material real time.
* Users have basic technological literacy.
* The system can integrate wih an API to include SMS/ email automation for MFA, and scheudling.
* Payments are processed through trusted, secured, third party tools.
* The platform will only have English at launch.
* The development team is not responsible for the integrity of the information, we assume Driver's pass has passed and validated accredited DMV trainings and instructors.

**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?*

* Users need to have internet to view information, and update their lesson progress to ensure validated data.
* The design team doesn't have the time and budget to validate all information being provided, we are simply including the specifications and trust the people hiring us to validate their infomration.
* If we can't afford liscensing of specific API's we might have to change or limit potential initial features. Things like automatctic texting for appointments, and MFA.

**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.*

