# CS 255 Business Requirements Document

## 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 system they want to build is to branch into an industry, lacking schools and training because numbers show that many people fail their exam the first time.

### 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 do both in person and online trainings it should be able to allow users to create and modify appointments. It should also give access to online trainings to certain users. It should also be up-to-date and send notifications when the DMV updates for regulations. They also want security to be done.
* This will need users and user roles; this would include access hierarchy as a security measure.

### 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 have user road functionality and an appointment calendar that can be accessed on the cloud and off-line. The webpage should look as suggested by the client. Work should be split into user stories and broken between continuous sprints.

## 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 needs to be web-based as it needs to be a website accessible to clients to be able to book lesson appointments as well as allow the owners and moderators to fetch the most current data.

#### 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 System should be able to run on any platform. As a website should be able to be opened the same on Windows and a Mac. Should be accessible from cellphones as well.

#### 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?*

* Each student should have their information given, such as first name, last name, address, ect. The system should alert the head of IT if any issues so he may be able to resolve those.

#### 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?*

* IT admin would need superior access to the system to have the ability to add/remove/modify and make changes.

#### 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?*

* The system would need some kind of username, email, or phone number and a password to log into an account. They should be able to reset their password on their own but accounts should also have blocked access after several failed attempts to discourage brute force attacks.

### 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.”*

### 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 shall be web-based, therefore browser friendly on all devices (computer and mobile)

### 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 am assuming that the system will require users to make a username. I am also assuming that since they want security taken care of from a development standpoint, we will be using JWTs.

### 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?*

* Offline access is very limited because it requires a fetch feature every time you are connected ot the internet. Meaning if you have not connected and made updated then when you connect, your changes may not push due to changes already in server not accounted for in your version. The safe way of this is simply to not allow offline changes and let offline be view only. This limitation will protect and keep the server up to date without conflicts but limits the user from making offline changes.

### Gantt Chart

