# 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 this project is to build an application that can help train students for their driving test.
* The client is DriverPass a company that wants to fill the void in the market of training students to drive and prepare for their road 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 provide online training and tests for student drivers.
* They want to fix the problem that is students failing their driving tests because they are underprepared.
* For administration driverpass wants to be able to access this data anywhere online and offline.

### 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 it is completed, the system should be able to communicate with a server to store data on clients.
* The Data should be accessible from any computer or phone if it is online.
* Store data on appointments and allow students to schedule appointments using their accounts.
* Some of the tasks to be included in the system design to achieve this are building an administrative back end to allow the staff to view data that the regular customer can’t see.
* Allowing the system to interact with the server to update the data in real time will make it so that if a customer cancels and appointment the staff would know, same with if they accept an appointment.

## 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?*

* The system should be web-based
* The system should run fast considering it needs to share data at a constant rate
* The system should be updated often to ensure that there are no bugs or problems with security

#### 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 platforms should run on Unix based operating systems, it is very light weight and companies save a lot of money using this over other options.
* The back end requires a database to support this application because of the saving and transferring of customer data.
* As much as I don’t like applications on mobile phones that just bring up the company's website it is an easy and cheaper alternative to building an entire application for iPhone and android and so on so the website should at least be able to be scaled down to fit on a phone screen.

#### 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?*

* Distinguishing between different users would be easy if you go by usernames or by email addresses or even names on a profile created by the user.
* The only case-sensitive information should be the password the user creates when making their profile.
* If there is an attempted security breach on another user's account, for example if I enter my password wrong 3 times it should inform an administrator.

#### 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?*

* Using POST requests and controller handling you can make changes to the user without editing the code at all.
* When updating the platform, you want the system to not be disrupted at all, this way students could still have access to the application.
* The IT admin should have access to everything, they should be able to remove accounts, send change password requests, remove employees if they get terminated and so on.

#### 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?*

* Log in with email and password
* SSL certificates ensure a secure connection between the client and the server
* If there is a brute force attempt the user should be put on a “cool down” period where they must contact the company and explain what happened to their account just in case, it was hacked or something bad happened.
* If the user forgets their password they should be able to request a password change through the email associated with their account.

### 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 the user to change their password.
* The system shall be completely online and only accessible online to ensure that there is no duplicated information given to the server.
* The system shall confirm whatever package that the customer selects based on the three packages available by DriverPass.
* The system shall allow the user to disable the package if they so choose to do that.
* The system shall be updated based on any problems with the dmv, (closed for holidays and such).
* The system shall confirm the user's information when they attempt to login.

### 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 user should be able to see on their profile what they are registered for or any upcoming available spots for a test
* The user should be able to see a part of the website to practice exams
* Home page
* The user should be able to see their information just in case anything changes

### 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?*

* We can assume that the users will comply with the rules of the DMV
* The system will stay updated with all the DMV rules in case they for some reason change.
* The system should be available to be always accessed

### 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?*

* Internet access
* Costs of hosting a website and using a database
* Can only schedule 10 driving tests at a time

### Gantt Chart

