## 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 is the client, consisting of Liam and Ian.
* They want DriverPass to fill a void in the market, training students for their driving tests.
* Ultimately, they would like DriverPass to allow students to take online classes and practice tests, along with actual on the road 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?*

* They want the system to help them access their data on and offline, as well as the before mentioned driving test assistance.
* They want to fix the low quality of driving training, to help more people pass their driving tests.
* Security is one of the larger components, as well as offline access and cloud storage.

### 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?*

* They system should have…
  + Offline access
  + Visible notes from the drivers
  + Tracking of information such as appointments, cancellations, driver and teacher status, etc.

## Requirements

### Nonfunctional Requirements

#### 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* The system should be updated frequently, as there could be massive issues if the security goes out of date, or DMV guidelines.
* The system should run both on the web and on a mobile app for ease of access to important information.
* The system speed should be quick in order for a smooth process when trying to complete all its functions, complete requests, quizzes and give lessons to students, etc.

#### 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 application would run well on any choice of internet browser with any platform.
* A database will be needed to store all of this information.

#### 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 need to enter personal credentials in order to distinguish between others, an email and username and password would work well.
* Case sensitivity is important for both keeping variety available and security.
* As soon as an issue arises the admin should be notified.

#### 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, you should be able to make changes without adjusting the code.
* The system can adapt via programmers.
* IT would have complete access to nearly all functions in order to keep things running smoothly.

#### 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 would need either an email/password combination or a username/password combination.
* HTTP would be a good solution for a solid client-server connection.
* The admin should be notified about something like excessive failed login attempts, and if it continues the admin can block access after X amount of failed attempts or the system can do it automatically.
* If a user forgets their password, they could have it sent to the email they used to sign up.

### 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 confirm login information.
* The system shall confirm user package preferences.
* The system shall update frequently and accordingly.
* The system shall confirm sensitive customer details.
* The system shall be fully available online, partially available offline.

### 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 needs to fit any device that can access the internet, phone, laptop, etc.
* The users are the admins, developers, and customers.
* Users will need to be able to book or cancel packages, lessons, and tests.

### 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 under the assumption that those who need access to the driving material have stable internet connection and a device to access the platform.

### 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 without the technology or internet requirements are limited.
* We are on a timeline as far as time constraints go, however we do not have a set budget.