## Brandon Hobbs

**CS-255**

**May 16, 2022**

## 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?*

* Client: DriverPass (Liam, Owner and Ian, CIO)
* Client wants to create an online LMS (Learning Management System) that allows users to train for their driver’s test. The LMS should allow the user to create their own learning plan from multiple defined packages and track their progress. The LMS should also allow application owners to interact with users through direct feedback or to update lesson plans if the source material changes

### 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?*

* The system is targeted to fill the “void in the market for training drivers at the DMV”. And address the issue of “so many people failing at the DMV”. The system will allow users to enroll in enough physical and/or virtual training until they can pass their tests.  
    
  The system will also allow DriverPass to better understand their users by generating marketing intelligence, available online and offline, and provide the most up-to-date information by connecting to the DMV directly.  
    
  The system also needs to be web-based and have some offline-report capabilities.

### 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 goal of the application to allow users to learn and be proficient enough to pass their exam at the DMV. The application will also allow DriverPass to understand their business better by generating marketing intelligence from user data and actions.
* To achieve these goals these components need to be derived:  
    
  Components breakdown for DriverPass:
  + Entitlement’s service: used to administrate user role
  + Report manager: used to deliver data for offline/online analysis
  + Asset management: used to assign users to instructors and instructors to cars
  + Subscription management: used to syndicate to DMV for changes in policy
  + Lesson management: used to administrate packages, including cancelling a package
  + Driver feedback: used to allow trainers to communicate to driver
  + Proxy service: used by certain roles, e.g., secretary, to enroll users as a proxy
  + Admin portal: used for admin tasks, e.g., password reset, account deletion
  + Dashboard: used to view any Key Performance Indicators (KPIs), e.g., driver booking percentage

Components breakdown for user:

* + Account management: used to create and administrate, e.g., change password, account
  + Training reservation: used to reserve time driving time
  + Testing service: used by the user to take tests
  + Lesson delivery: user access to the online lessons
  + Dashboard: used to view pertinent info., e.g., upcoming reservations, lessons remaining

## 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?*

* System is to be web-based with Chrome as the primary platform; PWA should also be implemented
* The underlying system should scale with user-base
* KPIs should be generated with a delay no greater than 24 hours
* Scheduling sub-system needs to be close to real-time to prevent duplication or “double booking”

#### 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-compatible; with Chrome as the main platform
* Mobile friendly with Chrome for Android as the main target and iPhone as the secondary; PWA enabled
* System to be based on a Linux back-end
* System will employ REST for any microservices

#### 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?*

* The system shall enforce case sensitivity on passwords but not on usernames. Usernames will be the users valid email address which should be unique, anyway.
* The system should have a “heartbeat monitor” alerting admins if the system goes offline

#### 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 system shall be based on an extensible platform, e.g., adding more disk storage, memory, etc.
* The system shall allow users and their roles to be curated within the platform
* The system shall allow for periodic updates with minimal downtime, e.g., server security updates

#### 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?*

* System shall employ account control via roles (RBAC)
* User will be required to register a valid email as their username
* User should be able to recover an account if username or password is forgotten
* System needs to be served through HTTPS and not HTTP
* System will have a mandatory “time out” period on account creation from the same IP address
* System will have a mandatory “time out” period for sign-on attempts from the same IP address
* System will lock an account after too many failed sign-on attempts; only an Admin may remove the lock

### 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 attempt to validate a user upon sign-on
* The system shall enforce the security policy upon successful or failed sign-on
* The system shall allow a user to review any current course subscriptions
* The system shall allow a user to reserve any available driving slots
* The system shall allow a user to alter (edit, cancel, etc.) any previously scheduled driver reservation
* The scheduling sub-system shall allow a user to view any available driver slots
* The system shall allow driver trainers to review their schedule
* The system shall allow instructors to interact with course takers, i.e., post grades, post messages, etc.
* The system shall allow content to be maintained, i.e., updates to training material following changes from DMV
* The system shall have a reporting function
* The reporting shall allow for offline exploration, e.g., download data as a CSV
* The system shall collect data in such a way to permit Key Performance Indicators (KPI) to be generated
* The KPIs shall be calculated automatically and viewable online to certain roles
* The system shall allow some roles to act as proxies for other accounts
* The system shall have an entitlements system to administer the RBAC
* The scheduling sub-system shall be able to operate across multiple time zones

### 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.)?*

* User shall be greeted by their dashboard based upon the client sketch
* User shall be able to review current subscriptions
* User shall be able to review current course work
* User shall be able to access the driver reservation system
* User shall be able to administrate their own accounts
* User shall be able to perform any test and review their performance on current or past tests
* Admin-roles shall be able to see all students’ and drivers’ schedules but not any Personal Identifiable Information (PII)
* Users with the proxy role shall see the schedule and subscriptions of the user they are a proxying

### 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 will be English speaking
* Users will have a valid email address
* Engineering staff will have access to business owners to answer any design questions

### 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?*

* System only can update while connect on-line
* Reservation system only works within the USA
* System will not be ADA complaint in first version

### 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.*Chart

Description automatically generated