# Michael Reynolds - 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 Client is “DriverPass”
* The Client envisions an application to fill a void in driver training per locality.
* Utilizing this application to solve an issue of outdated or unavailable driver training material/opportunity.

### 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 following to solve the problem of non-pipelined driver training per locality.
* Provide Online classes/tests and schedulable road training.
* User signup and online payments for reservation submission/cancellation/modification.
* Secretary reservation modification permissions.
* Online dashboard for Student test tracking.
* Run from cloud.
* Provide database/admin records across multiple platforms (Desktop/Mobile) to authorized users.
* Package plans for more exclusive/gradually beneficial appointments.
* Driver access to notes and test progress per student.
* Appointment details including driver assigned, tracking reports for modification history etc.

### 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 following will determine the system is complete.
* Online reservations and manual secretary assigned reservations.
* Dashboard to provide student progress, admin dashboard for reports and data/user modification.
* Run from cloud.
* Provide database/admin records across multiple platforms (Desktop/Mobile) to authorized users.
* Package plans for more exclusive/gradually beneficial appointments.
* Driver access to notes and test progress per student.

## 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?*

* Multi-Platform, Providing the application on both desktop, mobile, and web-based to provide for access from multiple device types.
* The system should run with constant updates, using web sockets to provide updated info at all times.
* The system should constantly be attempting updates with DMV standards, requiring routine maintenance to ensure the application and its resources are up to date.

#### 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 will need to be able to run on Windows, Unix, Android, iOS in order to be accessible on all platforms. The application should also be web-based to provide access through web browsers.
* The back-end will require a database and need to be run in the cloud, likely with Linux server distribution.
* The database will require middleware and a thorough API for providing database CRUD operations.

#### 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 must be stored on user objects held in documents on the database. These objects will include screen names, credentials, ID etc.
* User credentials will be case-sensitive and information will be parsed for security, the application will need to be aware of locality and accommodate language barriers.
* The system should have thorough logging, and auto-generate IT tickets upon any errors thrown outside of secure error handling.

#### 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?*

* Providing a role specific admin dashboard for modifying user data and modifying environment variables to customize the application will allow for controlled modifications allowing the administrators to manage the application without compromising code.
* Using a framework, we can ensure that the application can undergo a Continuous Integration pattern. With rolling updates, and platform dependent update methods.
* The IT Admin will have direct CRUD access to the database, credentials for access to the back-end cloud, and access to change permissions for users in the application and administrative users in the back-end.

#### 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 credentials required for logging in are a case-sensitive username and password, following average password length and special character guidelines.
* Using SSL and HTTPS we can secure the endpoint access from the client to the server on multiple platforms.
* If there is a brute force hacking attempt on an account, the account will be automatically locked and the end-user will be notified by email that they require either 2FA or manual authentication through a company support email.
* If the user forgets their password, they must request a password reset to their email or have administrators reset the password for them through support.

### 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 credentials when logging in.
* The system shall provide users with test/road grading and completion.
* The system shall provide secure access to user account management
* The system shall provide access from multiple devices and web-browsers.
* The system shall provide scheduling appointments and road tests.
* The system shall provide payment processing for online package purchases.
* The system shall provide records of user actions and modifications for administrative purposes.
* The system shall provide a separate dashboard for instructors and admins.
* The system shall provide an administrator product package plan add/delete interface

### 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.)?*

* A user interface for online reservations will be used both by end-users and for manual secretary assigned reservations. It should provide student/driver/location/transportation needs/date-time/payment status.
* A user interface for login and authentication, will be used by all front-end users alike, in order to log in using password/username input fields. A forgot password option and registration section.
* Dashboard to provide student progress will be available to end-users, showing them their credential specific progress, previous road tests etc. This dashboard will also be available to administrative users, where they can specify which user to view and view all student’s progress.
* An Admin specific dashboard for reports and data/user modification, providing a section for admin logs and a section for user querying and CRUD operations on the user documents.

### 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?*

* It’s assumed that the driver instructors will be provided paper grading criteria as they will not have a system for inputting live grading for road tests.
* It’s assumed that local DMVs will not differ should the application be used across multiple locations.
* It’s assumed that there is no need for access to any sort of digital products for the paid packages.
* It’s assumed that administrative users will have no need for access to the student portal/dashboard.

### 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?*

* The limitation of multi-platform standardization may conclude that the mobile-end will differ in available functionality to the website/desktop application.
* The limitation of budget for processing power on the cloud may prove challenging for min-maxing processing usage across devices.
* The limitation of time for Jan-22 till May-9 will prove to be an issue as standardization across platforms can have lots of platform-dependent bugs and programming language barriers.

### 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.*

Timeline

Description automatically generated