# CS 255 Business Requirements Document

## System Components and Design

### Purpose

DriverPass wants to create a web-based scheduling and training system for new drivers. The system should be able to test users, schedule tests at the DMV, allow users to take notes, and present learning modules. The system should be able to store local information, have an intuitive UI, and still have database support.

### System Background

DriverPass is looking to break into the market of online training for new drivers. The system will allow training of these drivers, and scheduling of appointments with the local DMV. DriverPass believes that tapping into this market will help thousands of students learn to drive. To do this, DriverPass is asking us to create a web based application with a UI, database, and user profile support.

### Objectives and Goals

The system should:

* be able to schedule appointments at the DMV
* have learning modules
* an ability to store notes
* an ability to store user profiles
* have an intuitive UI

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

* The system must be a web-based application, available on any OS that supports a web browser
* The system would be updated as new features or content are introduced, the browser or OS regular updates should not affect the application.
* The system should be responsive, user updates to their profile should take no longer than 2 seconds for the server to save

#### Platform Constraints

* The system should be available on any system that supports a web browser, this includes mobile devices, desktops, or game consoles.
* The back end should run on “serverless” cloud - based technologies offered by a company like AWS, IBM, or Microsoft.
* Server updates, security, as well as resources will be managed by the server company, whatever company that may be.

#### Accuracy and Precision

* To distinguish users, each user will select their own username.
  + The system should check the name against the database of current names, and reject duplicates.
* The usernames can consist of lowercase, capital letters, and numbers.
* The system should inform admins of error reports, the report should include the user and circumstances of the error(such as browser type, system parameters, user inputs prior to crash).
* The system should also inform Admins of server overuse, for example: a certain IP is driving up server costs in an attempt to undermine the company, therefore the Admin should IP ban the user.

#### Adaptability

* The IT admin will have access to prune inactive users from the database, modify inappropriate usernames, and have the ability to send password reset links manually if needed.
* The server processing power should scale with the amount of users active online, it is a benefit of the “Serverless” web based application model.
* The system should run as usual on any web browser after new platform updates.
* The database should scale with the amount of users on the platform.

#### Security

* Username and password are required for all users to login
* Client/ server communication would be handled by HTTPS
* Multiple failed login attempts trigger an email to the user and a CAPTCHA
* Users can request password reset via email

### Functional Requirements

* The system shall schedule appointments
* The system shall store notes
* The system shall have user profiles
* The system shall have a UI
* The system shall store and provide learning content

### User Interface

* The user should interact with the interface via a web browser
* The user should be able to view test progress and notes
* Users could include drivers, students, or secretaries
* The UI should work on any browser, including mobile.

### Assumptions

* The user has a reliable internet connection
* The user is using the most updated browser

### Limitations

* Only a few months of development
* Cannot support indefinite users

### Gantt Chart

