# 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?*

* Our client is DriverPass, a road test training service. The purpose of DriverPass is to provide people with training before their driving tests, because data shows that there are a lot of failures at their local DMV.
* They aim to accomplish this by offering online practice tests and road training with their fleet of cars and qualified trainers.
* In order to provide this service, their system will need to be able to handle online booking and transactions for their three tiers of services with varying amounts of training time and materials.

### 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 to help people practice for their driving exam to lower failure rates.
* The DriverPass system will be cloud based, to negate any worries of backup and security.
* The system needs to allow exports of any data so it can be worked on offline.
* Different staff members should have varying levels of permissions to edit appointments.
* DriverPass will be an online service that provides a dashboard to the user so they can see their upcoming appointments and training status.
* The owner should also be notified when there are any updates from the DMV so they can keep current with any changes.
* Registration should be possible over a phone call, as well as online.

### 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?*

* End user functions
  + Purchase training packages
  + Book appointments
  + View training status
  + View upcoming appointments
    - Length of session
    - Pickup location
    - Trainer ID
* Staff functions
  + Export data
  + Edit appointments
  + Edit trainee accounts
  + Register new users
  + Process payments

## 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?*

* DriverPass will be a web-based application that can be accessed from any device
* The system must be fast enough that the online assessment answers get submitted right away, and questions load right after the previous one.
* On the admin side, the owner should be able to access their data quickly so they can export into a spreadsheet. The system should be able to load data for exports reasonably fast, and this data should update whenever new information is sent from a user or an employee, such as lesson appointments, new accounts, and assessment scores.
* Since DriverPass will be connected to the DMV for policy updates, notifications must be received immediately.

#### 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?*

* DriverPass will run on Windows, Unix, Mac, and mobile, because it will be a web application that can run anywhere.
* The actual platform that the server will be running on will be a Unix based cloud server
* DriverPass will need to link to a database to store student information, package prices, assessments, scores, payment info, bookings, and all of the financial and traffic data stored by the website.

#### 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?*

* Different users will have unique identifiers to ensure that the correct match is identified when pairing students with trainers
* Inputs will be case sensitive when necessary, and any failed login attempts shall be logged in the database
* There will be three types of accounts, for admins, trainers, and students, and each will be identified differently.

#### 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 admin accounts will be able to edit all user roles and information, except for sensitive financial data
* All other account types (trainer and student) will be able to edit their own personal information, but nothing else
* When a user creates an account, it will be added to the system, but only one account per email address
* Updates to the system will be handled by adapting accounts in the database to any new format that is added, and since all data is stored in the backend in tables, it can be fitted to the new system before updates are rolled out to the public.

#### 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?*

* All accounts will be password protected and recommend two factor authentication
* Passwords will be encrypted and inputs will be hashed to verify a match with one in the system
* No more than 3 attempts can be made before the account is locked, at which point they will have to reset their password via the email address they signed up with This covers brute force protection, and is also how users will resolve forgotten passwords
* The admin account will receive a notification to their email if an incorrect password is attempted, due to it being a high privilege account type. For trainers and users, this will be optional.

### 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 allow users to book lessons with trainers at scheduled times
* The system shall provide practice assessments to users and display their scores
* The system shall display user training status
* The system shall allow admins to export data
* The system shall connect with local DMV policies

### 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 student home page will display the following:
  + Online test progress
  + Driver notes
  + Personal information (name, address, phone number, etc.)
  + Special needs
  + Driver photo
  + Trainer photo
* Trainers will have their own interface with a schedule for their training sessions
* Contact pages will be available for both staff and customers
* Admin accounts will have a basic data analytics page with ability to export data

### 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?*

* Students must have an internet connection to use the service
* Students must be able to afford at least one of the training packages
* Customers will not return once passing their driving test

### 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?*

* Target demographic will be restricted to people without a license, mostly young teens, and a lot of them only if they have parents willing to pay extra for training
* Budget will likely be a limitation as this will not be the most profitable business, initially only operating on a local level
* Data privacy will consume a lot of priority as most customer data will be from minors

### Gantt Chart

