# CS 255 Business Requirements Document Template

## 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?*

* Liam the owner of Driver Pass and Ian his technical lead
* Allow liam to keep track of all of his drivers and appointments
* Provide learning drivers a place to study and book trainings
* Data needs to be accessible anywhere online or read only offline
* Up to date DMV practice questions

### 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 hope to assist users in driver test prep providing practice tests lessons and behind the wheel appointments.
* Liam wants to be able to manage and track his fleet of vehicles, trainers, and customers.
* Data needs to be accessible from anywhere so work can be done in and out of the office.
* Database needed to store the data for user accounts driver locations and appointments.
* Need admin controls to reset passwords enable/disable accounts etc...
* Receptionist and users need ability to book appointments.

### 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?*

* Users need to be able to create edit and delete appointments for behind the wheel training
* Access to data anywhere online and off
* Multiple levels of access for different users Liam needs to be able to deactivate accounts as necessary
* Customizable packages in the future but for this project the ability for Liam to disable packages himself with no dev intervention.
* Activity tracking with Printable reports.

## 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 will need a web based front end
* System will need a database to store all of the user information and appointments
* System will need to have a backend architecture since its web-based Mabey something like Node.js or a similar backend service.
* The system ideally will run fast enough to handle all of the sites load and still allow users to make changes, check information and update appointments in real time.
* The system should be updated as necessary particularly when new security features are implemented.
* Database and website should also have regular backups taken to allow for recovery as necessary

#### 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?*

* OS would depend on the web dev software selected MacOS for ruby and ROR windows for C#/PHP etc...
* Database will be critical to this systems operation I would recommend a SQL product like Oracle or mySQL
* Would also recommend a software like graphql on the backend to allow for easy API integration as needed.

#### 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?*

* Admins should receive alerts for any problem with the system either via email or text message allowing the system to maintain as much uptime as possible.
* Usernames should be unique meaning no two users can have the same user name and usernames will not be case sensitive.
* Passwords will be case sensitive and will comply with the company's password policy to set a strong password.
* Passwords should be stored encrypted and never sent in plaintext.
* SQL for accounts should be explicitly defined so it cannot be injected and return user information

#### 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?*

* IT admin should have full control over user accounts without making code changes
* Build-in plans to allow for other site modifications like pricing and such without code changes but not necessary right away.
* Also, will need a special account for the receptionist as well so they can make appointments on people's accounts but should not be able to delete create or change any other settings on the user's account.

#### 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 will need to contact an admin to reset passwords if it is forgotten.
* User accounts should only allow for a limited number of failed logins to prevent a brute force attack 5 try's or so should be plenty for most people to successfully login.
* As mentioned above user data should be stored encrypted so that even if it is exposed it is unusable.
* User accounts will use secure tokens more than likely JSON style to prevent their accounts from being spoofed.

### 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 allow users to book an appointment
* The system shall allow users to access learning materials
* The system shall allow for multiple price point options
* The system shall be readable in offline mode anywhere anytime

### 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.)?*

* Regular user's admins and the receptionists will all have different interfaces.
* Admins need full account control with no code changes.
* Receptionists should have read-only access to all accounts to monitor appointments and answer questions as well as the modify right to set appointments for any account.
* Interface should be scalable so that it can be used on any device mobile or desktop.
* Interface should support mouse and keyboard as well as touchscreen interfaces.

### 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?*

* Assuming the customer will purchase licenses and equipment to support all the software
* Assuming that a cloud storage or possibly Haas system will be what the customer uses instead of maintaining their own equipment.
* User has no hard specifications for site design.

### 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 size and capacity will be limited by the equipment Drivers pass can purchase and maintain.
* System will need to be deployed quickly so that driverpass can get started but can plan for future integrations and upgrades as necessary.
* Money required to implement different features may vary and determine what features are available at release.

### 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.*

