# CS 255 Business Requirements Document

Parker Harb

## 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 the new company Driver Pass, represented by Liam, the owner, and Ian the IT Officer.
* They need a system to manage their driving education program, as well as provide services such as online testing.
* They also need information management for users.
* There is a requirement for three different class packages, as well as the ability to disable them without a developer's assistance.
* There will need to be different user roles, big boss, IT Officer, Secretary/internal user, and public users.

### 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 need a system that can handle an arbitrary number of users, with their own personal data. The owner Liam identified a lack of driving testing assistance in his area, and came up with this system to address said issue. For this, they need a number of features.
* Notably they’ll need both a front facing web app as well as a backend to manage data. They also prefer this to be done over the cloud using some kind of IaaS system, with minimal requirements for them to get their own hands dirty managing it.
* The whole thing will have complex security needs, as it will store a great deal of sensitive user info for DMV compliance.

### 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?*

* A user needs to be able to create an account.
* The owner needs a way to update user passwords
* Users should be able to schedule appointments through the online system.
* A user needs to be able to choose a package.
* A user needs to be able to store their personal information.
* Packages need to be able to be disabled.
* There needs to be an online test system to gauge users progress.
* These are the minimum tasks the system will need to have to complete, though more features such as being able to edit packages without a developer are nice to have. I’ve also listed them with the more important and system critical ones at the top, before anything else users need to be able to create accounts and register appointments, as that’s the primary business model. Even if there’s an emergency and the software needs to be released early, with those features, the business isn’t blocked from operating.
* Number one objective is to create working software as fast as possible. A good way to measure how far along the team is could be to compare the app to the drawing the owner provided, see what’s still missing, what might need to be cut.

## 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?*

* The system needs to be on the web, with a complete front end, back end, and cloud based server. The system should not have nearly zero lag and should run very well, it’s not doing anything complicated. The system will need some to be updated regularly with new content, and admins should be able to edit some or all of the data.

#### 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?*

* It needs to run on all major platforms, which shouldn’t be an issue if it’s a web app. As for tools on the backend, a cloud based virtual server running some kind of relational database would be ideal in both performance and cost efficiency.

#### 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 need to have user profiles which store their personal data, these need to be secure on a server. Input will sometimes be case sensitive and sometimes not. It depends on the particular context. In a perfect world the system would have a logger that automatically takes in users bug reports.

#### 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?*

* Users need to have mutable data, yes. It’s rather easy to include added functionality, user roles and placeholder variables. The system will be a web app on the front end, it should run pretty consistently, provided no unnecessary or poorly understood technologies are placed on top of it. The admin needs to be able to edit user account information at a minimum, resetting passwords and the like. It would be nice if they can edit web content as well.

#### 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?*

* If you want to make security a priority, a password and username, as well a multi factor authentication will be decently secure. Something like a one time code sent to the users email or phone. The system can also keep track of which devices have accessed it before. For securing the data transfer, no system is ever truly secure, but you can encrypt the data which helps a lot. For brute force hacking attempts, you can have the system check if someone accesses it, and see how often they access it, and block them based on that. If the user forgets their password, you can have the system email them a one time link to reset it.

### 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 store user data for future use.
* The system shall keep track of what packages a user has chosen
* The system shall allow the admin to edit packages
* The system shall authenticate purchases
* The system shall provide a testing environment for users
* The system shall allow scheduling of packages through various methods
* The system shall allow users to upload a driver photo
* The system shall be able to take in user input through the web

### 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 interface needs are fairly complex. There will need to be a way for the user to interact with the packages. There will also need to be a way to upload images or various users. There will also need to be an online testing system. There also needs to be a separate interface that only admins and users with specific roles can access. As for roles, there needs to be general users, or customers, drivers, system admins, and the owner with a root account that has 100% access. The user interface will probably be navigated mostly through a mouse to make setting up mobile interaction easy. The web app will need to be responsive and change based on what size the user's screen is.

### 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?*

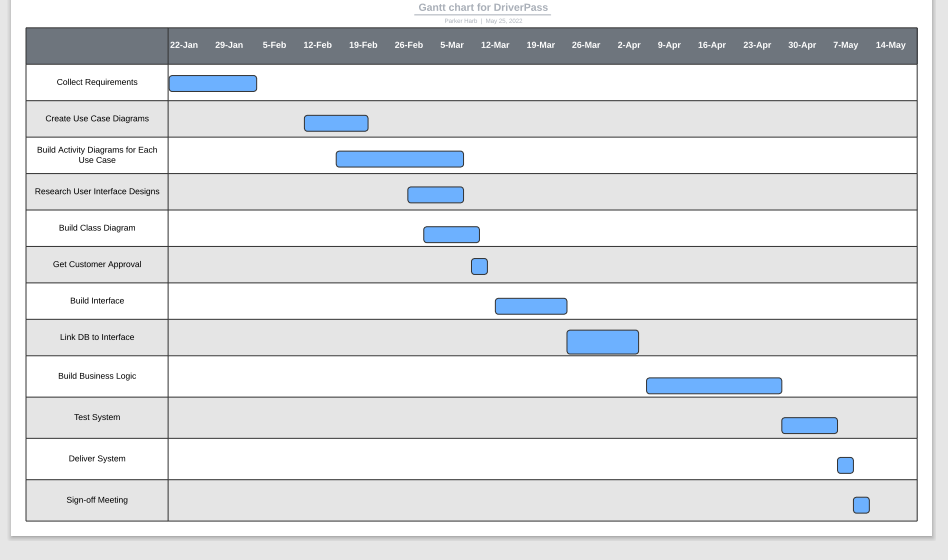
* I didn’t take accessibility into account whatsoever. It also wasn’t something listed a priority in the interview. That being said I believe that it’s something that needs to be designed for in every application. I pretty much wrote the above statement assuming every user would be able to navigate using a keyboard, and also didn’t mention the importance of using color schemes that can be viewed by colorblind people.

### 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?*

* Number one limitation will be our ability to build a system that can be edited by the admins. It takes a long time to make a web app like that, and we clearly have a limited budget and time. There will also likely be limitations on how much of the user interface will be built. We may not have the time to include things like online testing. As for technology, many important features are only available post Internet Explorer 9, so we have to choose between not supporting old versions of that browser and important new features to make our lives easier.

### Gantt Chart

**