# CS 255 Business Requirements Document

## Nick Allen

6/3/2023

## 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?*

Going through the transcript, the client wants DriverPass to:

* Handle online classes and practice tests
* Allow the students to be able to make, cancel, and modify driving lesson reservations
* Track reservations and the reservation’s information
* To be able to sell 3 lesson packages with the ability to add more later
* To store the customers data and create customer accounts
* To be able to download reportsoffline
* To connect with the DMV to stay up to date on policies and questions
* To show the students progress in the course
* To display comments left by the driver and time stamps for the lessons
* To be accessible from any computer or mobile device using the cloud

Including each of these functions would make a easily useable software for students to manage their driving lessons, along with a good way for each instructor to monitors the students progress.

### 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 a system to integrate with their existing company to make a comprehensive online driving training service. The clients believe this may fix the high failure rate for drivers tests at the DMV. To achieve this, they need a learning management system to handle the classes and practice tests, a reservation system for the in-person lessons with instructors, a customer database to hold everyone’s information, an access management system to handle different roles in the software, cloud accessibility, an intuitive interface and connection to the DMV to stay up to date with the DMV policies and regulations.

### 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 system should easily facilitate better communication between customers and instructors while also giving a well-organized learning experience. The tasks to be completed are the same as above:

1. Design a learning management system.
2. Develop a reservation module.
3. Implement a package management system.
4. Create a secure database.
5. Make user access management functionalities.
6. Make connectivity with the DMV.
7. Create an easy-to-use UI.
8. Implement the system on the cloud.

## 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?*

* Based on the request for this system to run on the cloud, it needs to be web-based for the most part. It’s not mentioned, but possibly in the future it could shift to app based should the company decide to upgrade.

In my opinion, there’s little downside to making the system run as fast as possible other than financial or infrastructure limitations, so I suppose as quickly as possible would be the logical answer.

As far as update frequency, the system may want to update on a daily, or weekly time scale depending on how frequent updates to DMV policy are.

#### 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?*

* If we are going with a cloud hosted, web based system, there’s no real requirement to run on any specific type of platform as the system is platform independent.

In terms of tools, apart from a Database Management System (DBMS), there may also be a need for tools related to web server management, cloud service management, security protocols, and possibly an integration tool for DMV updates.

#### 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?*

* So, how we separate users is straightforward. They'll all have unique usernames and passwords. Depending on who they are—like Liam the boss, Ian the IT guy, the secretary, or one of the customers—the system gives them different rights and access. As for case sensitivity, usually, usernames aren't case-sensitive because it's just easier for people to log in. Passwords, on the other hand, are usually case-sensitive for extra security.

As for when the system should nudge the admin, it's pretty much any time something isn't quite right. If someone tries to log in and can't, or if the system is having trouble connecting to the DMV for updates, or if it looks like someone is trying to mess with the system, the admin should get a heads-up. Basically, anything that could disrupt smooth operations should trigger an alert to the admin.

#### 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?*

* We could build a user management interface, and this is where the IT admin comes in. The IT admin, like Ian, would need full access to manage all the user accounts, so they can do stuff like resetting passwords or blocking access if someone leaves the company.

As for platform updates, we should make sure that the system is built to be as future-proof as possible. That means using widely supported technologies and frameworks. We should also do regular testing and have a version control system. With these, when platform updates roll out, we can adapt to the changes without affecting the system's functioning or causing downtime.

#### 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?*

* To log in, each user should have their unique username and a secure password. In addition, we could also add multi-factor authentication to increase security, particularly for users with higher-level access like the IT admin.

When it comes to securing the connection or data exchange between the client and the server, we'd use encrypted connections like HTTPS with SSL/TLS.

In the case of a "brute force" hacking attempt, where someone tries to guess a password by trying a lot of possibilities, we can set up the system to lock the account after a certain number of failed login attempts. This temporary lock would prevent the hacker from continually guessing the password. The real user, or the admin, can then unlock it with some verification steps.

If a user forgets their password, we'd have a secure password reset mechanism in place where the user needs to verify their identity to reset the password. Ian, as the IT admin, should also have the ability to reset passwords if users get stuck.

### 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 register, providing their personal details, and set up an account.
* The system shall validate user credentials when logging in.
* The system shall provide a mechanism for users to reset their forgotten password.
* The system shall allow customers to make, modify, or cancel reservations for driving lessons.
* The system shall support scheduling of lessons, allowing customers to select a preferred day and time.
* The system shall provide access to online classes and practice tests for customers.
* The system shall track and display a user's progress through the online tests, including scores and statuses of the tests.
* The system shall allow for selection of different packages for driving lessons and services.
* The system shall maintain records of each driving lesson, including the driver, time, car, and any driver comments.
* The system shall provide administrative access to Ian (IT Officer), to manage all accounts, and also allow for blocking or resetting access when necessary.
* The system shall track all changes to a record in the system, noting which user made each change.
* The system shall provide notifications of updates from the DMV to ensure current information.
* The system shall allow Liam (the owner) to disable a package from being chosen by new customers.
* The system shall offer offline access to reports and certain information for Liam.
* The system shall record and handle sensitive customer information securely, such as credit card details and addresses for pick-up and drop-off.
* The system shall be hosted over the cloud, taking care of backup and security issues.

### 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.)?*

* Its primary role is going to be to allow customers to easily make, modify, or cancel reservations for driving lessons. It also needs to provide a clear pathway for customers to access online classes and practice tests. It's also important that it displays progress through these online tests, including scores and statuses, and facilitates the selection of different driving lesson packages. The interface should allow users to securely input and update their profile information and to see visual data on driver notes, including the times for lessons and any comments left by the driver. Also, it should offer a way for customers to set their pick-up and drop-off locations. The interface needs to be built with administrative functions for different roles within the company.

For the UI, customers need to be able to register, log in, choose driving packages, schedule and manage lessons, access online classes and practice tests, and view their progress. They will interact with the interface primarily through a web browser, either on a computer or a mobile device. The owner, Liam, needs to be able to access data from anywhere, download reports, disable a package, and possibly perform more administrative tasks. He'll likely interact with the interface via both a computer and a mobile device. Ian, the IT Officer, needs full administrative access to all accounts to manage, block, or reset them as needed. He will also likely interact via a computer and possibly a mobile device. The secretary needs to manage appointments, including making, modifying, and cancelling them on behalf of customers who call or visit the office. This role will likely interact primarily with a computer at the office location.

### 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?*

* Scalability isn't explicitly addressed - the system needs to be built to accommodate business growth, which means an increase in traffic and user data. Secondly, the necessity for customer support functionality is overlooked. An ideal design would consider a feature where users can submit queries or report issues.

The system design also doesn't discuss localization and accessibility to cater to different regional users or those with specific accessibility needs, which could be important in future expansion. The payment processing aspect of package selection isn't specified either, which will be important for functionality.

Some assumptions I’ve made are the customers access to reliable internet, their technical literacy, and access to modern devices that support the interface 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?*

* Scalability comes into play again as a limitation. If there’s suddenly a lot of users, there may be performance issues. Cloud systems are generally reliable during surges, but large amounts of users could slow things down.

In regard to time, there’s a lot of different aspects of this software that need to be implemented, meaning the development may take quite a while. In regard to budget and resources, ultimately, we may be limited by the clients monetary means, as the developers and other team members working on this software will need to be compensated for their time and expertise.

### 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.*

*A screenshot of a computer

Description automatically generated with medium confidence*