# 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?*

* The client is DriverPass, working with Liam the company owner and Ian the IT officer. The client wants to create a system where people seeking to learn how to drive and pass the driving test at their local DMV can access driving lessons and, optionally, practice the written driving exam to prepare using online classes and study materials along with practice tests. The idea is to have resources for students available from anywhere by having the resources online as well as a download option to access them offline. Users are able to schedule driving lessons via different packages offered by the client with varying number of sessions, each two hours long.

### 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?*

* The client is trying to provide a system where people can learn to drive and pass the driving exam at their local DMV. Essentially a driver training service. According to the client there is a lack of resources for this in the market and they believe they can provide the solution. The first, and main, component is the ability for students to schedule driving lessons with an instructor. The student should be able to select a day, time slot, each session lasting two hours, as well as the pickup and drop off addresses. Another component is in regards to the online study material and classes provided online. Should the student choose the package that provides study materials as well as practice tests, they should be able to access this information from anywhere while online or have the option to download it for use offline. Users should also be able to reset a password if needed without needing to call customer support.

### 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?*

* To evaluate when the system is complete, there are two sides that need to be addressed which are the administrative side and the customer side:
* First there are the administrative requirements. First, DriverPass must be able to register new students so they can use the system. Since there are varying roles within the system as well as sensitive data being stored, security needs to be addressed. As far as sensitive customer data being stored, database access has to be restricted to certain users with authorization. This can be addressed using a cloud solution as requested by the client. This will include roles and the ability to view who has updated anything in the database. Ian must have full access to all accounts in case he needs to access any user information, help with password resets, or remove accounts of employees that have left the company. In addition, packages can be disabled at any time. There needs to be an interface that shows the student’s progress in terms of the tests completed and the ones in progress. It will contain data such as test name,time taken, whether it was completed, and a status such as passed, failed, etc. There must be a section with the customer’s information such as name, address, etc. This information will be pulled from a form filled out by the user or secretary. A picture of the student and instructor. A section with any special needs required. Finally there is a “Driver Notes” section that contains the lesson times and any comments left by the instructor. Additionally, there will be a page with contact information for both customer support and the customer. For compliance, DriverPass must offer students up to date policies for their respective local DMV.
* From the customer side the main thing they must be able to do is schedule driving lessons with an instructor. They can select one of three packages that offer a different number of sessions, each two hours long. The user must be able to provide a day, time, pickup and drop off address, not necessarily the same, and can reschedule, if necessary, online or by calling the secretary. It is important that rescheduling must be done online or through customer support. However, If they choose the third package which offers online classes with study materials and practice tests, these must also be available offline. In addition to being able to log into their account, they can also reset their password without having to call in for customer support.

## 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 application needs to run both on web and mobile. The system should run quickly, a magnitude of milliseconds for the most part to ensure excellent user experience. The system should be updated for significant changes every four to six weeks based on the use of the Agile methodology. Changes concerning the user’s local DMV should occur as soon as a change is detected by the system since it will communicate with the DMV often.

#### 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?*

* The system will run on both Windows and Unix/Unix-like systems. The backend will require a relational database, in this case MySQL, to store important data used by the application be it user details, driving lessons schedules, etc.

#### 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?*

* The system will use a login with two factor authentication to ensure user identity. The password will be case sensitive and will be hashed with a salt to be compared to a saved hash in the database. Actual passwords will not be stored for security reasons. After three unsuccessful logins, the user will be locked out and admins will be notified of the event via an event log in the database for admin use.

#### 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 can modify their own information in their profiles without any issues. As far as adding and removing users, admins will be able to do that via an admin dashboard. During significant updates, the system may be down for a few hours, or may have limited functionality depending on the update. The effect on users can be limited by performing updates during hours of low traffic. IT should be able to access AWS via their logins to make any necessary updates to the system since this is a serverless application.

#### 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?*

* A user will need a username, password, and a two factor authentication application, in this case Authy. The password will be hashed and combined with a salt to compare against the password hash in the database. If the password hash matches, then the two factor authentication app will be used to finalize the login process. The connection will be secured using HTTPS. Should there be a brute force attack, after three unsuccessful logins, the user will be locked out and an admin will receive a notification and an event log in the database will record that a login issue has occurred. The user will be notified that they need to contact customer support to unlock their account. If a user needs to reset a password, they can simply follow a link that will ask them to reset that password which will require the use of two factor authentication again.

### 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 will validate user credentials.
* The system will allow users to reset their password.
* The system will allow students to access courses if they are included in the package both online and offline.
* The system will allow students to book driving lessons.
* The system will allow instructors to write notes about a student’s session as well as include details of progress and lessons remaining.
* The system will allow admins to add, modify, and remove users.
* The system will allow admins to access sensitive data stored in the database that no other user can access.
* The system will allow IT special privileges to perform system updates through AWS.

### 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 interface should present them with a GUI where they can modify their account details. They must also be able to interact with the site to schedule driving lessons as well as access course materials.
* The instructors must be able to interact with the site to record updates on a student’s driving lessons, including their progress and how many lessons remain.
* Admins will have full access to the site and can interact with any interface provided to both the student and the instructor. It is important that an event log is maintained to keep track of which user has made changes during an event be it an admin, student, etc.
* IT will not make much use of the interfaces of the site, if at all, since any updates to the application will be done through the AWS portal.

### 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?*

* As far as users, the main assumption is that the typical user is fairly tech savvy and can navigate the site or application without much, if any, help.
* The application will be secure by leveraging robust third party services such as Auth0 to make sure everything is up to industry standards.
* The system will leverage AWS to make sure the system, and all of its services, are available 99.99% of the time, and the scalability will not be an issue since AWS can handle this as well.

### 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?*

* The main constraint will be the budget. Although AWS can help with cost effective solutions, it can be costly if not properly set up. Since the budget must be kept in mind, it is not guaranteed that the services used by the application can be configured to scale optimally during high periods of traffic, which could negatively impact user experience. Although this should not be an issue most of the time, it must be taken into account.
* Another limitation concerns users. Although most users should be fairly tech savvy, not all users will be and the system can only be so intuitive. Setting up tutorials on how to use the application within the application for users can help mitigate this, but it will not fix the issue completely.

### 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.*

