# 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?*

* DriverPass is a startup company that wants to provide driving training for people who are going to take their driver’s test at the DMV.
* The new system needs to provide online classes and practice tests for driving students.
* The new system needs to allow students to book on-the-road training sessions with a trainer.

### 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?*

* Many people fail their driver’s test at the DMV and DriverPass wants to help more people pass.
* DriverPass wants to mitigate test fails by better preparing drivers for the test.
* DriverPass wants to offer paid online classes and practice tests, as well as paid on-the-road driving lessons.
* For the system to work successfully, it will need several components: administration, secretary account, user accounts, driver accounts, a server, clients, backend API, frontend responsiveness, a database, and connectivity.

### 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 be accessible at any time. This means the system should work in a client-server architecture that provides web-based distributed resources to clients at any time.
* The system should be accessible to any computer or mobile device. This means the system should support traditional desktop operating systems using mainstream web browsers, as well as iOS and Android.
* Admins should have the ability to download a report of DriverPass information
* DriverPass does not want the hassle of an on-premise server. A cloud-based system is preferred.
* The system will have several different types of users. The system needs a role-based security implementation to allow different employees distinct system privileges.
* The system needs to be able to track user changes and provide an activity report of recorded changes.
* Each customer needs to be able to make their own account on the system. Data recorded on each customer will be first name, last name, address, phone number, state, and credit card number, expiration date, and security code, as well as pickup/drop-off location.
* Each customer needs the ability to change their password if they forget it. The system will need more than just a password for authentication, such as email verification.
* The system should allow reservations to be made online for driving lessons, as well as cancel or modify such reservations.
* The system must identify the driver that the customer is scheduled to go out with for driving lessons. For each reservation, the system will track which user is matched up with a certain driver, time, and car.
* The driving lessons reserving feature must give the customer a choice between three package options.
* DriverPass wants to be able to disable/enable these driver lesson packages at will.
* The system needs the ability to get a notification whenever the DMV makes updates to their drivers test curriculum.
* The system needs to provide the user with their test(s) progress and outcomes.
* The system needs to record driver notes for on-road lessons, as well as the lesson time information. These will be displayed on the customer’s account page.
* The system will need an account creation page for the customer or secretary making the account.
* The system will need a “contact us” page and a way to contact the student, such as a chat feature or an official company support email.
* The system will need to allow students to take online classes and practice tests

Other measurable tasks include:

* Collect all requirements
* Create use case diagrams
* Build activity diagrams for each use case
* Research user interface designs
* Build class diagram
* Build interface
* Link DB to interface
* Build business logic
* Test system
* Deliver system

## 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.*

#### Operational Requirements

*What environments (web-based, application, etc.) does this system need to run in?*

* The services must be available 24/7
* The application will be a web-based distributed environment based on a client-server architecture
* The system should be able to work with any web browser

#### Performance Requirements

*How fast should the system run? How often should the system be updated?*

* Desktop and laptop clients should be able to load resources within 2 seconds
* Mobile devices should be able to load resources within 4 seconds
* The system should be updated monthly to handle any technical changes in the industry, or faults that might be found in the system

#### 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 should support traditional desktop platforms such as popular web browsers on Windows, Linux, and macOS
* The system should support mobile platforms such as iOS and Android web browsers
* The back end will require a RESTful API
* The back end will require a database
* Front end will require responsiveness
* The system will be hosted by a cloud server

#### 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 input will be case-sensitive for user login credentials
* Users will be distinguished by their account username and will be authenticated with a password
* The system will inform admins automatically when it loses connectivity or functionality
* The system will inform admins automatically when it is compromised by a threat
* The system will be able to track user changes and provide an activity report of recorded changes

#### 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?*

* Admins will be able to add, remove, or modify users if they have the right role/authority to do so
* The system design will follow programming standards that allow browser compatibility. The system will be updated when there is a change to these standards. This will allow seamless compatibility between the application and browsers even when there are innovations released in browsers
* The admins will be ready when there are OS updates. If incompatibility issues are reported after Windows releases an update, the system will be addressed accordingly

#### 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?*

* The system will have a role-based authorization scheme and username/password authentication to account for user types and privileges
* To combat “brute force” hacking attempts, the system will require a Captcha verification when a client attempts to login with a wrong password 3 times in a row within a certain time frame. The user will be locked out of the account after failing password attempts too many times or failing the Captcha too many times
* The system will use email verification and phone verification to allow authentication for password resets
* The system will use a RESTful API that encrypts all data transmissions. The encryption algorithm will be a proven-secure open source algorithm
* The API will issue a temporary token out to clients when they establish a connection and are authenticated. The client will have access to resources until the token expires, in which case the client will need to acquire a new token

### Functional Requirements

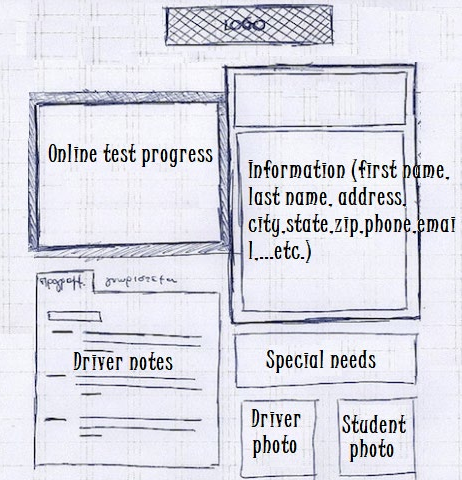
*Using the information from the scenario, think about the different functions the system needs to provide. Each of your bullets should start with “The system shall . . .” For example, one functional requirement might be, “The system shall validate user credentials when logging in.”*

* The system shall allow student users to take online classes
* The system shall allow student users to take practice tests
* The system shall provide the student user with their test(s) progress and outcomes
* The system shall allow student users or secretary to make reservations online for driving lessons
* The system shall allow students and admins to cancel or modify driving lessons reservations
* The system shall track which student is matched up with which driver, which car, and at what time
* The system shall have three driving lesson packages that a user can choose from
* The system shall allow DriverPass admins to disable/enable these driver lesson packages at will
* The system shall get notifications when the DMV makes updates to their drivers test curriculum
* The system shall record driver notes for on-road lessons, as well as the lesson time information, all of which will be displayed to the student
* The system shall allow students or the secretary to make a new account
* The system shall display contact information for the student to reach out to DriverPass, such as an official customer support DriverPass email address
* The system shall allow admins to download a report of DriverPass information

### 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 UI should follow the eight golden rules of interface design (Shneiderman, n.d)
* The user will interact with the UI via a browser on a traditional desktop or a mobile device
* The users of the UI will include the owner, IT staff, secretary, and students/customers
* The student users should be able to input information such as first name, last name, address, et cetera
* There should be a “contact us” page that allows the student to reach out to DriverPass
* There should be a page for taking online classes
* There should be a page for taking practice tests
* There should be a page for registering for driver’s lessons
* The UI should have an account page that shows the student’s test progress, account information, driver notes, special needs, and the driver and student photos, as shown below:



### 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?*

* Users have access to a computer or mobile device that has a popular web browser
* Users have access to the internet
* Enough marketing was done so that there are users who know about the website

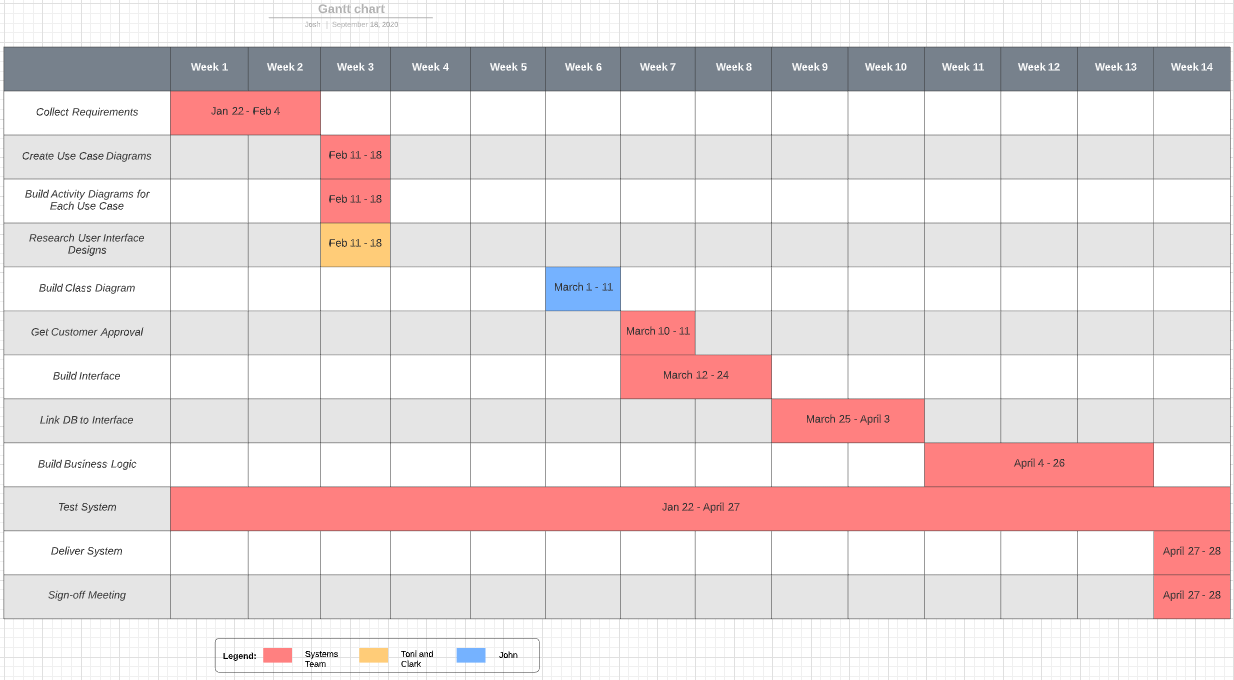
### Limitations

*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?*

* We are not making a native app for iOS and Android, which could impact the convenience of the customer
* The server and clients must be connected to the internet
* The success of the application is dependent on the content of the classes, tests, and the quality of driving lessons
* The project should be signed off on May 10th which will give us about three and a half months
* The project will have a limited budget based on the agreed sum on the sign-off sheet
* The system should be able to work with limited maintenance, as the owner of DriverPass does not want a lot of technical hassle
* The system cannot be an on-premise, and will be an offsite server in the cloud

### 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.*



**References**

Shneiderman, Ben. N.d. The Eight Golden Rules of Interface Design. University of Maryland. Retrieved from https://www.cs.umd.edu/users/ben/goldenrules.html