# CS 255 Business Requirements Document Template

## System Components and Design

### Purpose

* The purpose of this project is to build a system for our client DriverPass,
* DriverPass wants to take advantage of the void in the market for Driver Test training
* System should be able to host online classes
* System should be able to give practice tests

### System Background

* Data from the system should be accessible and modifiable remotely
* They want extensive security: roles, rights, and access control
* Actions in the system should be tracked and recorded, and should be easily viewable/retrievable
* The system should allow reservations to be made online or by DriverPass themselves
* Reservations should be put into a schedule with the corresponding Driver, Time, and Car
* There are three beginning packages that can be reserved, and they need the ability to disable a package if need be
* For registration, the system requires First and Last Name, Address, Phone Number, State, Credit Card Number, Expiration Date, and the Security Code
* System should allow password resetting
* System should link to the DMV and provide updates to DriverPass when rules, policies, and sample questions are created/removed/updated
* Backend should be cloud based

### Objectives and Goals

* A mock design was provided and should be used to create the basic functioning webpage for testing
* The system being cloud based means no server equipment is needed, but the correct cloud service should be selected for the project’s needs
* A database must be set up to keep track of client information, tests, and scheduling
* Individual reports should be pullable about clients, test scores, schedules, finances, etc. Each report needs a template that will be filled with the appropriate information when retrieved
* DriverPass’s drivers need to provide availability before scheduling functionality is implemented
* DriverPass employees should be assigned the correct permissions, and the IT Officer needs full access to remove accounts/reset accounts/revoke privileges
* Registration should be possible online by clients and in person by DriverPass, and the information is put into the database
* Must inquire with the DMV if they have an API or something similar to obtain new rules and such, otherwise a unique solution will need to be implemented to provide notifications to DriverPass
* Finished product should be clean and functional with the cloud on the backend working with the database(s)

## Requirements

### Nonfunctional Requirements

#### 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?*

* This will be a web-based system accessible through a browser on desktop and mobile devices
* Security and cloud services are outsourced (Suggested: AWS for top-of-the-line service)
* The front-end should be extremely responsive as it should be lightweight in build with the usage of JavaScript and/or other front-end languages
* Retrieving information from the back end should also be very fast as it will be built in the cloud
* Depending on the language and database, the system could need to be updated anywhere from weekly to a few months, as methods and functions are updated or removed somewhat commonly

#### 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?*

* There should be little in the way of platform constraints as the front-end operates in a browser, the back end is in the cloud, and popular databases like MySQL and MongoDB have platform-specific distributions
* The back end will need to link to a database to store and retrieve customer data, scheduling information, and test questions and answers
* MongoDB is suggested for the database as it is noSQL and can be cloud based, which should improve security and speed

#### 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 will have unique roles in the system to differentiate between employees, customers, and admins
* JavaScript can easily parse input, checking for validity of names and passwords, as well as simple things like case-sensitivity
* The system should push a notification to the admin if a user registers a support ticket
* The system should notify the admin if a user attempts to login incorrectly too many times
* The system should notify the admin if there are security breaches (mass login attempts, DDOS attacks, etc.)

#### 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?*

* User accounts should be easily added/removed/modified with methods communicating those specific purposes through the back end to the database, which would then make the change
* The system should be mostly unbothered by platform updates as it is built in the cloud and accessed in a browser, meaning the system would likely only need to adapt to new browser specifications or new database/cloud methods
* The IT admin should have overarching access across all accounts with the ability to manual reset passwords or restrict account access
* The IT admin’s access should be tied to their account, which has a unique role that grants them their abilities in the system

#### 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 should ensure that the username and password being used are in the database and are linked together, only then should a user be allowed to sign in
* The data exchange between client and server should always be encrypted to prevent data loss or theft, and this can be done with JSON Web Encryption or something similar
* The IT admin should be notified if the system catches brute force hacking attempts (mass inquiries or duplicate packets), who can then lock the account and take measures against the attempt
* If a user wants to reset their password, they can go through a link that will prompt them to verify their identity through email or text authorization, and if verified, they will an email or text that gives a password reset link
* If an employee forgets their password, the IT admin should be able to manually reset their password for them

### 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 receive front and back-end functionality
* The system shall display appropriate login portals for customers and employees
* The system shall prompt the user to login with their credentials on each trip to the site
* The system shall allow users to reset their username and password through email or text verification
* The system shall display appropriate landing pages and content depending on the user’s role
* The system shall provide students a platform to access their learning information and results
* The system shall provide tests and practice materials relevant to the local DMV
* The system shall allow users to update their personal information once logged in
* The system shall allow students to register for available courses
* The system shall display content relevant to the courses the user is registered for
* The system shall notify users of maintenance times with at least 24-hour notice
* The system shall have scalable storage capability
* The system shall allow employees with correct permissions to generate and download reports
* The system shall allow employee drivers to manually update their schedules
* The system will provide easily viewable privacy clauses, legality statements, and other relevant company information
* The system shall include accommodations for disabled users

### 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 to responsive and provide specific information and content depending on the role of the current user
* The interface is different for students and employees
* Students should be able to access their coursework, register for courses, see their accommodations, view and update their profile, view their progress, see their driver, and see notes made by their driver
* Employees should be able to access their student’s progress, leave students notes, view and update their own profiles, view their schedules, and review their course materials
* The owner should have access to all forms of data, including financials
* The IT admin should be able to generate site data reports as well as reset passwords and restrict access for employees
* The website should be responsive and adaptive, meaning it should work efficiently and effectively on desktop, phones, and tablets

### 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 are assumed to be accessing the website on a device capable of using a modern web browser, be it on the desktop or on mobile
* Hardware specifications are not defined, but modern systems should be able to access the website without problems stemming from doing so
* Overall system costs are currently unknown but will be presented to DriverPass during the customer approval phase
* Current roles inside the system are assumed to be “Student”, “Driver”, “IT Administrator”, and “Owner”
* The system is designed under the assumption that the DMV is willing and capable to provide new information and tests through the backend (preferably an API of some form)

### 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?*

* Storage space is limited by DriverPass’s willingness to spend
* Production time could be limited if not enough experienced engineers in building cloud-based systems can be employed
* Internet speed can severely limit responsiveness
* Time to deliver system is tentatively limited to under 5 months
* Budget is currently unknown but will need to contain cloud storage, database, and web hosting

### Gantt Chart

Timeline

Description automatically generated