# CS 255 Business Requirements Document Template

## 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?*

* Client : DriverPass
* Schedule appointments for driving training
* Take online tests/record progress
* Keep dmv standards up to date

### 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?*

* Void in the market when it comes to training students for driving test
* Need for better training/ reduce test failures
* Requires
  + Completed websites
  + User accounts
  + Varying levels of permissions
* Operating platform and programming language
* Cloud Server Host

### 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?*

* Cloud hosted website
* Reservation class with objects
  + User
  + Date
  + Time
  + Vehicle
  + Instructor
* Administrators can generate reports
* Connection to the dmv, update new rules policies, sample questions, notification on updates
* Measurable tasks
  + Cloud website
  + User accounts
  + Admin can review logs/ handle password resets/edit records
  + Users can create reservations
  + Website fits the customer template

## 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 web-based
* The system needs to be updated on a per request basis from the client, or monthly to account for DMV rule changes
* The system needs to be fast enough to allow for smooth loading for both the client and users

#### 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 run on Windows
* The system should run on Azure Cloud to account for backend/database requirements

#### 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 should have unique, case sensitive account names to distinguish between users
* These accounts should have case sensitive passwords meeting industry standards to ensure security.
* Admin should be alerted if any issues arise, regardless of the size

#### 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?*

* The system should allow edits to users without changing code
* The IT admin should have access to the entire system so that they can properly administrate
* Updates should seamlessly apply to the system, ensuring that the database and user accounts aren’t affected by the update

#### 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?*

* There should be a case sensitive username and password combination that is properly hashed according to industry standards
* The user’s account should lock after 3 attempts, requiring either a time delay or admin assistance to reset. This should prevent brute force attempts.
* If the user forgets their password, they should be able to receive an email allowing them to reset it on their own
* The data exchange should occur over the cloud, ensuring that security is handled by the service provider

### 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 check the username and password combination and verify it is correct before allowing a login
* The system shall allow the user to make minor changes to their account, ie: updating payment information, address, etc
* The system shall allow the user to book reservations
* The system shall allow the user to edit or cancel reservations
* The system shall provide the user with reservation information, such as time, location, vehicle and instructor
* The system shall allow the user to take practice tests
* The system shall include a history of what the customer has accomplished, to allow for a review
* The system shall ensure that there are 3 packages available to the customer and allow them to select them
* The system shall ensure that the 10 available vehicles do not get overbooked
* The system shall allow admin to disable packages
* The system shall allow varying levels of access, from customers, to instructors, to admin

### 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 should allow non-technical users to easily login and navigate the site
* Locating user account information, as well as finding different pages, such as booking a reservation and taking practice exams should be easily found on the home page
* The different users, with escalating levels of access, should be: customers, instructors/employees, admin
* Customers should be able to view their information, take practice tests, and book appointments
* Instructors should be able to review/administrate tests, and view appointments
* Admin should be able to edit account information, cancel tests, cancel appointments, and contact users
* There should be a desktop and mobile version of the website, to allow users to interact with the interface via mobile and desktop browser.

### 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?*

* One major assumption is that the users will be familiar with how websites generally function, such as how to login, how websites are laid out, etc.
* We are assuming that users have a stable, high speed internet connection and a computing device(desktop/laptop/tablet/mobile) to properly interact with the site
* How IT will continue to maintain the website via the cloud server, and the associated costs wasn’t really discussed.
* It’s not clear on whether or not our company will be handing this project off to the client or will continue to be responsible for updates down the line.

### 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 budget was not discussed
* The cloud is necessary, but upkeep costs should be kept in mind
* We are limited by our number of team members, and time
* We have a time frame of Late January to Early May, or about 3.5 to 4 months

### 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.*

