# 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?*

* The purpose of this project is to design and develop a better system for DriverPass, a company that is addressing the needs for improved driver training. DriverPass intends to offer an integrated platform the provides online practice exams and road training to enhance the preparation of students for drivers’ tests. The client Liam is in hopes that we will fulfill his vision by creating a system that facilitates access to online recourses, scheduling of driving lessons, and tracking user progress in compliance with DMV regulations.

### 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 has identified a significant issue in society regarding inadequate driver training, resulting in a high rate of test failures at the DMV. To address this Drivers pass wants to create a system that can run online, and off the web using the cloud. This lifts the burden of dealing with backup and security so they can focus on the business without technical issues. The solution comprises of offering training serves in the form of online classes, practices test, and on road driving lessons. Using and leveraging technology, DriverPass intends to increase driver education and improve the pass rate of individuals taking driver tests.

### 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?*

* When completed, the system for DriverPass should offer a platform for driver training, providing users with access to online classes, practice tests, and scheduling for on the road driving lessons. Users should be able to seamlessly schedule appointments and be able to track their progress. In addition, the system should ensure data security, scalability, and reliability to support the companies’ operations.

Measurable tasks:

1. Develop user authentication system to validate credentials.
2. Design and implement online classes and practice tests.
3. Create the functionality the allows users to schedule, modify, and cancel driving appointments.
4. Establish feature that allows user to track their activity and progress.
5. Implement a cloud-based architecture to ensure scalability and reliability.
6. Conduct testing to ensure functionality and performance are running as intended.
7. Deploy the prototype system for testing and feedback to stakeholders.

## 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 should be able to run in a web-based environment to ensure accessibility from any device that can connect to the internet.
* The system should employ a seamless experience to users, aiming for loading times to be under the margin of 3-5 seconds.
* Updates should be a regular occurrence; updates should occur with new features and/or bug fixes implemented every 3 or 4 months.

#### 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 be able to run on any operating system such as Windows, Linux, and MacOS.
* The backend requires a database management system to store and retrieve data efficiently.
* The system should be resized to fit the screen of mobile devices.
* The system should be able to run on all internet browsers such as Microsoft Edge, Google Chrome, Firefox, 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?*

* User identification will be distinguished by unique credentials of the email address and password.
* The input will be case sensitive to prevent user authentication issues.
* The system will notify the admin in the case of multiple failed attempts of user authentication.

#### 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 changes to users to be added, removed, and modified roles and permissions without changing the underlying code.
* It should adapt to platform updates seamlessly with minimal disruptions to service.
* The IT administrator should have full access to system configurations, this includes user management, security settings, and system updates.

#### 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?*

* User login will be done using a username and password, or email and password.
* Data exchange regarding the client and server should be encrypted using a protocol such as HTTPS.
* Users should be able to reset their passwords in the event they would like to change it or have forgotten their password through a verification process. This can be done using their email or a SMS verification text that brings them to a link to change their password.

### 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 validate user credentials during login.
* The system shall confirm the user selection of the package selected.
* The system shall provide online classes and practice tests for users.
* The system shall allow users to schedule, modify, and cancel driving appointments online.
* The system shall track user activity.
* The system shall have a cloud-based architecture for scalability and reliability.

### 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.)?*

* User interface should be based off the roles of the user.
* Users should see:

Home page

Exams

Courses

Grades

Account information

Register (new users)

Inbox (for communication with instructors)

Scheduling (for in person driving appointments)

* Administrators should be able to manage user accounts, view activity logs, and configure the system settings.
* Instructors should have access to scheduling, student profiles, and lesson feedback sections.
* The interface should be accessible through web browsers on mobile devices and desktops.

### 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?*

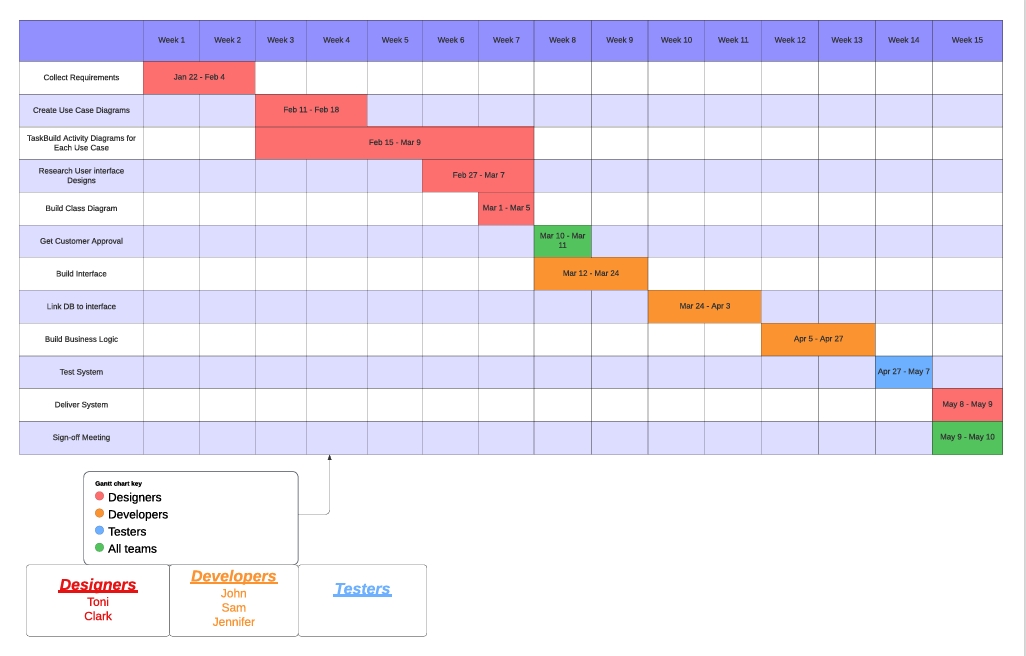
* It is assumed the user must have internet access and access to web browsers.
* We can assume the users of the platform will be DMV students.
* Users are expected to have an email address for account registration.
* The system assumes users will comply with the security protocols and maintain confidentiality of their login credentials.

### 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?*

* Limited recourses can affect the speed of the development of the system and feature improvements.
* Time constraints can affect the testing phases and the quality of the system.
* Budget constraints can impact the scope of the project, this can lead to prioritization of essential features over total functionalities.
* The client only has 10 cars, so scheduling needs to be updated regularly to avoid overbooking.

### Gantt Chart

**