# 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?*

* To design a system for DriverPass that supports online driver education, practice tests, and on-the-road training scheduling.
* The client, DriverPass, wants an accessible, web-based platform for managing customer registrations, training packages, scheduling, and testing.

### 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 identified a lack of effective tools to train students for DMV driving tests.
* The new system must support lesson scheduling, test-taking, user data management, payment processing, and real-time reporting.
* Components needed: web-based platform, cloud database, user interfaces for different roles, activity tracking, and secure login system.

### 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?*

* Provide customers with an online portal to register, schedule, and take practice tests.
* Track user activities such as reservations, cancellations, and modifications.
* Provide access to reports, schedules, and feedback.
* Support multiple training packages and allow future modifications.
* Enable role-based access for admin, IT, and secretary users.

## 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 must run as a web application hosted in the cloud.
* It should support concurrent users with minimal load time.
* Regular updates must be scheduled quarterly.

#### 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?*

* Compatible across Windows, macOS, iOS, and Android.
* Cloud-based storage and backup.
* Uses a relational database.

#### 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?*

* Role-based authentication with unique user credentials.
* Case-sensitive inputs where applicable.
* Admin notified for unauthorized access or errors.

#### 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?*

* IT admin should modify user accounts through admin panel.
* Future updates can be deployed with minimal disruption.
* Packages can be enabled/disabled by admin without code changes.

#### 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?*

* Login required with multi-factor authentication.
* SSL encryption for all data exchanges.
* Auto account lockout after five failed login attempts.
* Password reset via email verification.

### 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 users to register for accounts.
* The system shall validate user credentials during login.
* The system shall allow users to schedule, cancel, and reschedule driving appointments.
* The system shall support three training packages with predefined lesson hours and content.
* The system shall log activity for reservations and changes.
* The system shall display test history with status.
* The system shall track instructor notes per session.
* The system shall allow admin to access reports and download data.
* The system shall allow users to update their personal and payment information securely.
* The system shall automatically sync with DMV updates.

### 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.)?*

* Web-based UI accessible via desktop and mobile browsers.
* User roles: Admin, IT Officer, Secretary, Customer.
* Customers: register, schedule training, take tests, view history.
* Secretary: input customer data, make appointments.
* Admin/IT: manage users, view reports, oversee activity logs.
* Dashboard includes test progress, lesson schedule, driver comments, contact forms.

### 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 the internet and modern browsers.
* Users have basic technical literacy.
* DMV data integration is possible via API or periodic update.

### 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?*

* Future package customization requires developer intervention.
* System depends on external DMV updates.
* Limited support for offline usage.
* Initial budget and timeline restrict scope to current features onlyl.

### 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.*

**