# CS 255 Business Requirements Document Template

Zach DeLong

CS 255

6OCT2024

Professor Stefanelli

## 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 develop an online platform that helps students prepare for both the theoretical and practical components of driving tests.
* To provide students with access to realistic practice driving exams and on-the-road training sessions.
* To create a seamless system for scheduling driving lessons with instructors and receiving feedback.
* To improve the overall pass rate of students taking their driving tests by offering comprehensive training beyond memorizing past exams.
* To support multiple user roles (students, instructors, admins) with a secure, role-based access control system.
* To provide DriverPass with a system that integrates with their existing IT infrastructure for monitoring, managing, and updating content.

### 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?*

* Online practice exams that simulate the real driving test.
* On-the-road training with instructors through scheduling and feedback mechanisms.
* A system that supports both desktop and mobile interfaces.
* Admin capabilities to manage users, content, and system settings.

### 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?*

* Providing an intuitive user interface for students to sign up, take practice exams, and schedule on-the-road training sessions.
* Ensuring that students receive feedback after each practice test, allowing them to track their progress.
* Offering a seamless process for scheduling driving lessons with instructors, including appointment reminders.
* Managing users, courses, tests, and feedback data.
* Facilitating admin and IT staff with monitoring, managing, and updating system components, user accounts, and course material.

## 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 handle up to 10,000 concurrent users without a noticeable degradation in response time.
* Load times for any page must not exceed 2 seconds under normal operating conditions.
* The system must process user inputs (e.g., quiz submissions, scheduling requests) within 1 second.
* The backend must efficiently handle large volumes of data, particularly during peak registration periods.

#### 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 must be compatible with all modern web browsers (Chrome, Firefox, Safari, and Edge).
* The platform must be responsive and accessible on mobile devices, including smartphones and tablets.
* It should be compatible with Windows, macOS, Linux, Android, and iOS operating systems.
* Integration with third-party scheduling and payment processing APIs, such as Google Calendar and PayPal, is required.

#### 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 must provide highly accurate scheduling, ensuring no overlap in instructor-student bookings.
* Practice exam results must be precise and based on official standards to ensure accurate student feedback.
* All payment transactions must be accurately recorded to avoid discrepancies in billing or service provision.
* Data integrity must be maintained, with regular checks on databases to ensure that no information is lost or corrupted.

#### 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 must be scalable to accommodate future growth, including the addition of new users, courses, and locations.
* It should support modular updates, allowing for the easy addition of new features, such as practice exams for commercial driving licenses.
* The platform must be able to adapt to regulatory changes without requiring a complete redesign.
* It must be designed in a way that can easily integrate with future tools or services, such as enhanced reporting features or machine-learning-based performance feedback.

#### 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 must comply with relevant data privacy regulations, such as GDPR, to ensure student data is protected.
* Role-based access control must be implemented, ensuring that only authorized personnel (admins, instructors) can access sensitive information.
* All payment information must be encrypted and securely stored to prevent unauthorized access.
* Regular security audits and penetration testing must be conducted to ensure the system is free from vulnerabilities.
* The system must offer two-factor authentication (2FA) for added security, especially for administrative access.

### 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 when logging in.
* The system shall provide students with access to practice tests and driving lesson scheduling.
* The system shall track user progress, including test scores and completed lessons.
* The system shall allow instructors to log feedback on student driving sessions.
* The system shall provide administrators with the ability to manage content, users, and system settings.
* The system shall provide reminders for scheduled lessons via email or SMS notifications.

### 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.)?*

* **Students:** They will need to be able to sign up, log in, take practice tests, view feedback, and schedule driving lessons. The interface must work well on both desktop and mobile browsers.
* **Instructors:** They will log in to view their schedule and student feedback, as well as enter session results and observations.
* **Admins:** They will need to monitor system performance, update course content, manage users, and review reports on system usage.

### 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 will have basic access to the internet and the ability to use standard browsers such as Chrome, Firefox, or Safari.
* Instructors and admins are assumed to have some technical proficiency for interacting with the back-end system.
* The system will integrate smoothly with DriverPass's existing infrastructure and IT environment.

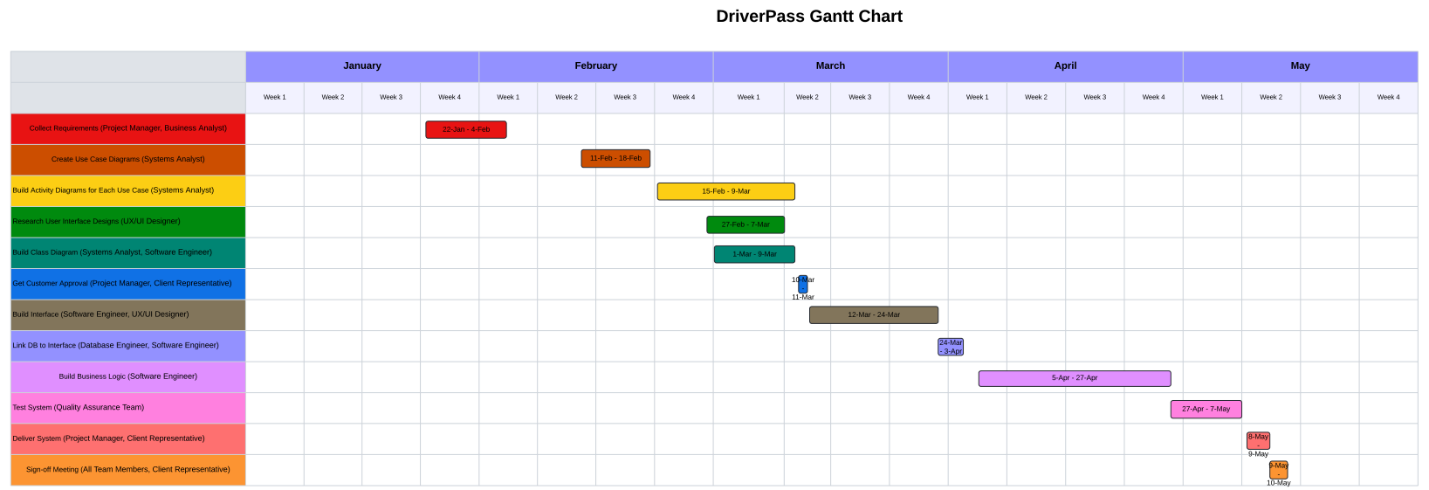
### Limitations

*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 system must be built with a limited budget, which could constrain certain advanced features.
* Time constraints may limit the scope of functionalities available in the initial launch.
* Compatibility with future browser versions may require ongoing updates post-launch.

### 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.*

**