# CS 255 Business Requirements Document Template

**Jess Dowd**

**Maureen Kinkela**

**CS 255 System Analysis and Design**

**07 August 2024**

## 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?*

* **Project Purpose**: Develop a comprehensive educational system for DriverPass that enables students to access online practice exams and schedule driving lessons, aiming to improve their preparation and success rates for DMV driving tests.
* **Client**: DriverPass, an innovative company dedicated to improving driver training outcomes by providing integrated access to online educational tools and practical on-road training sessions, aimed at better preparing students for their driving tests.
* **Client Objective**: Provide a robust platform that allows students to prepare for driving tests by accessing online practice exams and scheduling driving lessons, enhancing their chances of passing DMV tests successfully.
* **System Functionalities Desired**: DriverPass wants the system to facilitate seamless access to educational content online and to manage the logistics of scheduling and tracking driving lessons efficiently. Liam, the owner of Driverpass specifically mentioned the need for a system that allows for online and possibly offline access to data, easy management of reservations, and clear tracking of user activities such as lesson bookings and cancellations
* **Vision for Impact**: By integrating these educational services, DriverPass seeks to establish a new standard in driver education, making it more accessible, effective, and adaptable to student needs

### 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?*

* **System Functionality**: DriverPass wants the system to provide an all-encompassing platform where students can easily access online practice exams, schedule driving lessons, and track their progress, with the ability to manage their learning experience both online and offline efficiently.
* **Identified Problem**: DriverPass has recognized that a significant number of potential drivers fail their DMV tests due to insufficient preparation and training, a gap they aim to fill with their comprehensive training approach.
* **Proposed Solution**: DriverPass aims to fill this gap by providing comprehensive training options that combine online educational tools and practical driving lessons, aiming to significantly improve pass rates (Transcript, p. 1).
* **Required System Components**:
  + **Online Platform**: A user-friendly interface for students to access and take practice tests, view educational materials, and monitor their progress
  + **Scheduling System**: An efficient module for booking, rescheduling, and cancelling driving lessons, accessible both online and via traditional methods such as phone or in-person visits
  + **Data Management**: A secure backend system to manage user data, lesson schedules, and educational content, ensuring data consistency and reliability
  + **Reporting Tools**: Capabilities for generating detailed activity reports to track reservations, cancellations, and modifications, as required by Liam for operational oversight
  + **Security Features**: Robust authentication and authorization protocols to manage different user roles and access rights, addressing the needs described by Ian for administrative control over user accounts

### 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?*

* **Facilitate Access to Online Educational Tools**: The system should provide a user-friendly interface where students can log in to access and take practice tests. This objective is driven by DriverPass’s requirement for a system that allows students to access data online from any device, which they highlighted as essential for the system's success.
* **Streamline Scheduling of Driving Lessons**: According to DriverPass, customers should be able to make reservations for driving lessons online or via phone and in person. The system must support flexible scheduling options while managing the logistics of time slots, driving instructors, and vehicle availability.
* **Implement Comprehensive Tracking and Reporting Capabilities**: Tracking user activities such as reservations, cancellations, and the last modifications made by users is crucial, as emphasized by DriverPass. They expressed the need to print activity reports to ascertain responsibility in case of discrepancies.
* **Ensure System Flexibility for Future Modifications**: DriverPass expressed a desire to possibly modify the range of training packages offered in the future. While the initial system setup does not require the capability for DriverPass to add or remove packages themself, it should be designed to allow easy updates and modifications by our team, reflecting DriverPass’s vision for a flexible, scalable solution.
* **Maintain Regulatory Compliance**: DriverPass also stressed the importance of staying current with DMV regulations and requirements. The system should have capabilities to receive updates from the DMV and integrate them into the training content, ensuring all provided material is up-to-date and compliant.

## 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 operate as a web-based platform and also a mobile application to have more accessibility.

#### Response times for user interactions should not be more than two seconds in normal conditions.

#### System updates should be scheduled biannually to add feature and update security.

#### 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 platform-independent and run on Windows and Unix systems and other major operating systems.
* The backend will need a database management system to store data, retrieve it, and backup. This will make sure that the system remains flexible and can use third-party services and APIs.

#### 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 be distinguished based on their login credentials that will include a username and password.

#### The input fields should not be case-sensitive to prevent any entry errors.

#### Administrators should get informed immediately for any critical system errors, and a report of minor issues should be sent daily.

#### 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 will let administrators add, remove, or modify the user accounts through a interface without any direct code modifications.

#### It will automatically adjust to software updates to make sure that is compatible and is working well.

#### IT administrators will need full access to system settings and management tools.

#### 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?*

* Users will need to give a username and a password to login.
* Secure connections will be managed through TLS/SSL encryption for all the data exchanges between the client and the server.
* To prevent brute force hacking after five consecutive failed login attempts the account will be temporarily locked, and an alert should be sent to administrators.
* If a user forgot their password they can reset it through a secure process that would include answering some personal security questions.

### 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 let students sign up, log in, and book driving lessons.
* It shall allow instructors to upload lesson materials and handle their schedules.
* It shall have a way for students and instructors to chat in real time.
* It shall keep detailed records of all activities, like bookings and cancellations.
* It shall make changing or canceling bookings as easy as possible for students.

### 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 be easy to use and accessible for everyone including students, instructors, and admins.
* Students should be able to book, reschedule, and cancel lessons. They should also be able to use learning material and communicate with the instructors.
* Instructors should be able to upload content, manage their schedules, and interact with students.
* Administrators should be able to keep and eye on and manage the system operations and user activities.
* Everyone should be able to access the system easily whether they are on their phone or with a computer using a web 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?*

**Internet Accessibility**: Assumes all users have access to high-speed internet.

* **Rationale**: The system's performance and user experience depends on reliable internet connectivity.

**Technical Proficiency**: Assumes users has basic digital literacy skills.

* **Rationale**: The system design is made for users who are familiar with basic digital operations.

### 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?*

**Budget Constraints**: Limited funding could restrict some of the features at launch.

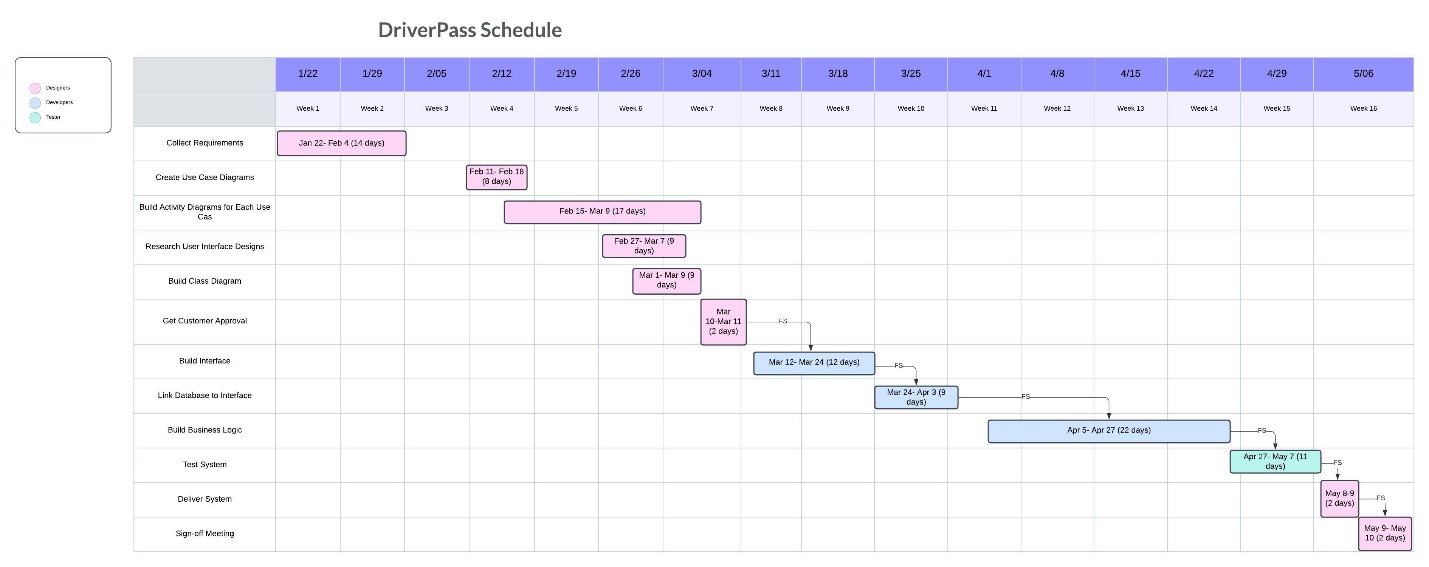
* **Rationale**: Financial limitations can affect the scope of the system features, which could affect the development timeline and quality.

**Technology Dependence**: The system’s performance is relying on third-party service reliability (e.g., hosting services).

* **Rationale**: Dependencies on service providers could have risks out of our control, like downtime or service disruptions.

### 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.*



Reference:

*CS 255 DriverPass Interview Transcript.* Southern New Hampshire University.

<https://learn.snhu.edu/content/enforced/1644084-CS-255-11450.202456-1/course_documents/CS%20255%20DriverPass%20Interview%20Transcript.pdf?ou=1644084>

Dennis, A., Wixom, B. H., & Tegarden, D. (n.d.). *O’Reilly Media - Technology and Business training*. O’Reilly Online Learning. <https://learning.oreilly.com/library/view/systems-analysis-and/9781118037423/08_chapter003.html#ch003>

Pmp, F. U. (2023, March 18). Assumptions and Constraints in Project Management | PM Study Circle. *PM Study Circle*. <https://pmstudycircle.com/assumptions-and-constraints-in-project-management/>

*Tips for writing business requirements documents*. (2020, April 29). Lucidchart. <https://www.lucidchart.com/blog/tips-for-a-perfect-business-requirements-document>