**Dat Nguyen**

**CS-255: System Analysis and Design**

**Module 5**

**Christopher Eppenger**

**6 October 2024**

# CS 255 Business Requirements Document Template

Complete this template by replacing the bracketed text with the relevant information.

This template lays out all the different sections that you need to complete for Project One. Each section has guiding questions to prompt your thinking. These questions are meant to guide your initial responses to each area. You are encouraged to go beyond these questions using what you have learned in your readings. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead, the goal is to complete each section based on your client’s needs.

**Tip:** You should respond in a bulleted list for each section. This will make your thoughts easier to reference when you move into the design phase for Project Two. One starter bullet has been provided for you in each section, but you will need to add more.

## 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 DriverPass system aims to develop a comprehensive online platform to help students pass their driving tests by offering practice exams, study materials, and on-the-road driving lesson scheduling. DriverPass seeks to address the gap in the market by providing more than just study materials; it focuses on practical learning and real-life driving experiences. The system will serve customers, administrators, and instructors with different needs for accessing and managing lessons, tests, and progress.

### 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 aims to provide better training for people preparing for their driving license exams. More than 65% of students fail the exam because they rely solely on studying past tests. DriverPass wants to improve this by offering online practice tests, instructional materials, and real-world driving lessons. The system will need to support online registration, payment processing, appointment scheduling, practice tests, and the generation of user progress reports.

The fundamental problem they are trying to solve is offering a system that integrates all aspects of driver training, from scheduling lessons to tracking customer progress, while providing easy access for customers and staff.

Several critical components are essential to achieving the goals set by DriverPass:

* The system must include user registration and login capabilities for different roles, such as customers, administrators, and instructors. It should also provide online practice tests with progress tracking specifically for customers.
* The system requires a scheduling feature that accommodates driving lessons, with flexibility for handling multiple drivers and vehicles. Administrators will also need access to a report generation tool to track system activity and monitor user progress.
* Role-based access control will ensure appropriate access and functionalities are available to different users.

### 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?*

The DriverPass system is built with several key goals, all designed to enhance the experience for customers and staff. First and foremost, the system will make it easy for customers to create accounts, book driving lessons, take online practice tests, and track their progress. This gives users complete control over their learning journey, allowing them to manage their schedules and monitor how well they prepare for the driving exam. The system will provide administrators with the tools to manage user roles, monitor lesson schedules, and generate performance reports, making oversight smooth and efficient. Instructors will also benefit from the system, as it will help them manage their schedules, assign lessons, and leave feedback for students, making it easier to stay organized. Security is a top priority, so role-based access control ensures that only authorized users can access sensitive functions. Lastly, the system will be flexible enough to accommodate future changes, like adding new training packages as DriverPass grows and evolves.

A few critical tasks must be implemented to ensure these goals are met. One of the most important is user account management, where customers can quickly sign up, log in, and manage their accounts. The scheduling system will be another crucial feature, allowing users to book, cancel, or change lessons, with the system automatically updating lesson availability and tracking assigned drivers. Administrators can export data, such as user details and lesson schedules, into external applications like Excel for easy reporting and analysis. Additionally, the system will offer practice tests that track a user's progress by displaying test names, scores, and completion statuses, giving customers a clear view of their actions.

## 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 DriverPass system will be a web-based platform designed to work seamlessly on desktop and mobile devices. This makes it easy for users to access the system, whether at home on their computer or the go with their phone. The system needs to handle many users at once without slowing down. Ideally, it should respond to user actions in less than five seconds, so everything runs smoothly without frustrating delays, whether someone is booking a lesson or taking a practice test.

In addition to speed, keeping the system updated regularly is critical. As DMV rules and driving test requirements change, the system must be refreshed to reflect those updates, mainly when new practice questions or regulations are introduced. This will ensure that users are always prepared with the most current information available, helping them stay ahead of any changes in driving requirements. Regular updates and fast, responsive performance will keep the DriverPass system running efficiently and effectively for everyone who uses it.

#### 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 DriverPass system is built to be flexible and work across multiple platforms, including Windows, Mac, and Unix. This ensures that no matter what operating system users are on, they'll have no trouble accessing and using the system. Whether someone prefers working on a Windows PC, a Mac, or a Unix-based machine, they can navigate the system smoothly without compatibility issues.

The system will rely on cloud-based data storage and management on the back end. This setup provides the advantage of scalability and secure access, making it easier for the system to grow as DriverPass expands. Storing data in the cloud also enhances security. It minimizes the need for complex physical server setups, allowing the company to focus more on its core services rather than worrying about managing servers.

Regarding accessibility, the system must be fully compatible with popular web browsers like Chrome, Firefox, and Safari. Ensuring the system works well on these browsers makes it more user-friendly, allowing people to log in and use it from their preferred browser and device. This combination of cross-platform support, cloud-based infrastructure, and browser compatibility ensures a smooth experience for all users.

#### 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?*

In the DriverPass system, it is essential to differentiate between the various types of users—customers, administrators, and instructors. Each group will only have access to the specific parts of the system relevant to them. For example, customers will interact with features like booking lessons and taking practice tests, while instructors will need access to schedules and the ability to leave feedback on lessons. Administrators will have complete control over managing users, generating reports, and overseeing the system. Keeping each role limited to its appropriate areas ensures security and smooth functionality.

The system will also track and log everything users do, especially changes like rescheduling lessons or updating test results. This records who did what, which is essential if any issues or disputes arise later. Whether a customer modifies a booking, or an instructor updates a student's progress, these actions will be logged to provide transparency and accountability.

Additionally, the system will keep administrators in the loop if something goes wrong. If a user fails to log in after multiple attempts or there is an error with the scheduling system, the admin will be notified immediately. This allows them to quickly step in and fix the problem, ensuring the system stays up and running smoothly for everyone.

#### 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 DriverPass system needs to be flexible enough for administrators to make changes without involving a developer. For example, admins should be able to easily modify, add, or remove lesson packages through the system's interface. This makes it simple to adjust offerings like driving lessons or online test options as needed without the hassle of changing the system's underlying code. This flexibility means the system can quickly respond to customer needs or business changes without requiring major technical work, keeping everything running smoothly.

The system must also handle updates to its platform and any third-party tools it connects with, such as online payment systems. Whether it is integrating a new payment method or adapting to updates in external tools, the system should be able to incorporate these changes without causing interruptions for users. This adaptability ensures that DriverPass can continue operating efficiently as technology evolves, allowing the business to stay current without disruptions. By incorporating this flexibility, the system will be well-prepared to grow and change alongside the company's needs.

#### 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?*

Security is a top priority for the DriverPass system, and several safeguards will be in place to protect user data and ensure safe access. One crucial feature is role-based access control, which ensures that users—an admin, instructor, or customer—only have access to the system's relevant parts. This keeps sensitive areas of the system secure and ensures that users only interact with the tools they need.

The system uses multi-factor authentication (MFA) and HTTPS encryption to keep accounts secure when logging in. This means that even if someone gets a hold of a password, they will still need to provide an additional verification form, like a code sent to their phone. HTTPS encryption will ensure that all data shared between the user and the system stays protected, especially when logging in or handling sensitive information.

In the case of a brute force attack, where someone attempts to break into an account by repeatedly guessing the password, the system will lock the account after several failed attempts. This will trigger an alert for the admin so they can immediately secure the account. Additionally, if a user forgets their password, the system will provide a straightforward way to reset it securely so they can regain access without any hassle.

Together, these security measures ensure that the DriverPass system remains safe for all users, providing robust protection for personal data while keeping the system accessible and easy to use.

### 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 should include the following functional features:

* User Registration:
  + The system shall allow customers, admins, and instructors to create accounts.
  + The system shall validate user credentials when logging in.
* Lesson Scheduling:
  + The system shall allow users to schedule, cancel, and reschedule lessons.
  + The system shall track lesson availability based on drivers' and car schedules.
* Practice Tests:
  + The system shall provide online practice tests that track customer performance, including test scores, completion time, and status (passed, failed, in progress).
* Progress Tracking:
  + The system shall allow instructors to leave feedback on driving lessons and track customer performance.
  + The system shall generate reports that detail customer progress, including completed lessons and tests.
* Data Export:
  + The system shall allow admin users to export user data and reports in formats like Excel.

### 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 DriverPass system needs a simple and intuitive user interface that works well for three main groups: customers, instructors, and administrators. The interface should be easy to navigate for customers, allowing them to create accounts, schedule lessons, take practice tests, and track their progress without hassle. Since people will be accessing the system from different devices, the interface must work seamlessly on both desktops and mobile phones so customers can use it whenever and wherever they need to.

Instructors will need a distinct set of tools within the interface. They should be able to easily view their lesson schedules and track how their students' progress. The interface should also give them a way to leave notes and feedback on lessons, making it simple to keep everything organized and ensure students get the support they need.

The interface will need to be more robust for administrators. They will use it to manage user accounts, update lesson packages, and generate reports. Admins must also have access to system logs and user activity to track what is happening within the system. Overall, the interface must be user-friendly for all three groups, ensuring everyone can complete their tasks efficiently using a computer or a mobile device.

### 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?*

In designing the DriverPass system, a few assumptions have been made about the users and the technology they will be using. First, the system assumes all users will have reliable internet access. Since the platform is web-based, it is essential that users have a stable internet connection to book lessons, take practice tests, manage schedules, and handle other necessary tasks.

We also assume that users will be familiar with fundamental online interactions. This means things like creating an account, booking appointments, and navigating a dashboard will be familiar to them. While the system is designed to be user-friendly, we expect users to already have a general understanding of how these processes work.

Lastly, the system's roles—customers, instructors, and admins—must stay consistent throughout development. The system is built with these specific roles in mind, and we do nott anticipate any significant changes to how they function or are managed. These assumptions help guide the system's design and ensure it works smoothly for everyone involved.

### 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?*

Like any system, the DriverPass platform has limitations. One of the main restrictions is around offline functionality. While users can access specific data, like downloadable reports, when they are offline, they will not be able to make any changes to lessons, user accounts, or practice test results without an internet connection. This is to ensure there are not any conflicts or data issues that arise from making offline changes that don't sync properly when the user is back online.

Another system limitation is tied to the project's budget and scope. Due to financial constraints, some advanced features, such as video conferencing for driving lessons or automatic updates from the DMV, won't be included in the first phase of the system. While these features could enhance the platform, they must be considered for future updates if the budget allows.

Finally, although admins will have control overturning training packages on or off, more complex changes—like adding or removing new packages—will still require a developer's help. So, while the system is flexible to an extent, some more extensive adjustments will still need technical support. These limitations are something to be aware of as the system is developed, but they also leave room for improvement as the platform grows.

### 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.*

