# 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 client for this project is DriverPass. The purpose of this project is to create a website that allows students to easily take online classes, practice exams, and on the road training.

### 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 wants to make driver training easier and more accessible to everyone. Driver Pass believes that it’s too difficult to pass the driving test in its current state and believes it’s due to poor driver training.

*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 system needs to be able to handle massive amounts of traffic, without falling apart. So, basically it needs to be able to scale correctly. It also needs to handle some sort of scheduling system. This would be for scheduling when the student would do either a test, or training on the road.

The system will need to have a login system, and probably a permission system as well. This way Admins could easily remove lower-level members of the team if they’re let go or leave for whatever reason.

The system also needs activity monitoring for each user, or in other words, decent logging with a front end to view it.

There are a few other requirements listed in the interview, but ultimately I believe the system should be created 1 step at a time with constant updates to the client. Business requirements change all the time, so taking everything at face value is a mistake, because the result will end up much different than the initial idea.

## 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 be run on any web browser, and if it needs to be an application then it would need to have some sort of cross platform solution. Something like React would do the trick here. The system should be updated often and should be designed with that in mind. Speed is very important as well, and it should run just as fast as any of the fastest websites online. If it ran slowly, no one would want to use 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?*

* This system should be running on a Unix-based system. Realistically speaking, using the cloud for this will be better. This is still going to be Unix, but scaling problems will already be solved with this. Cloud solutions usually offer database support of any kind as well. I think the database used here is optional, but MongoDB is a solid choice.

#### 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 way to distinguish between 2 different users is always based on each user having their own unique identifier. This can be an integer, a UUID, or even just an email or name. Because multiple people can have the same name, that’s not an option. But every email is unique, so just the email alone would be enough. Case sensitive checks wouldn’t be necessary.

#### 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?*

* If the system is designed to allow user updates without updating code, then yes that’s completely possible. The code would just need to be designed to allow the front end to be configurable. An IT admin would probably just need an ADMIN account, or maybe some kind of SUDO account which allows access to literally everything.

#### 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 user would need to enter their email/password to login and then pass a basic 2FA question. This could be through a text message, email, or even some sort of security app. Security would be handled through basic 2FA, which just means if someone somehow knew their email and password, they still wouldn’t be able to login with just that. And if the user forgets their password, there just needs to be a simple “forgot password” button at the bottom, which would send a new password update option to their email.

### 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 validate further through 2FA.
* The system shall have different account types with different permissions. Example -> Student, Teacher, Admin
* The system shall have log auditing upon every interaction for basic tracking measures.

### 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 system needs of this interface is to ultimately provide a system for teaching and learning. The different users would be a Student, Teacher, Admin, and maybe even something else. Each user would simply have different things to access. For example, the teacher might be able to create a discussion board and might be able to set up curriculum/grading. Whereas the student would only have access to view/use those systems, but not create or update them. And the Admin user would be able to do everything and more, for updating/maintenance reasons.

### 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?*

* The assumptions being made are that the people using this system are either teachers or students. And the people paying for the system are likely a school/college. The entire system is being built around that assumption, as it’s trying to reach a specific target audience.

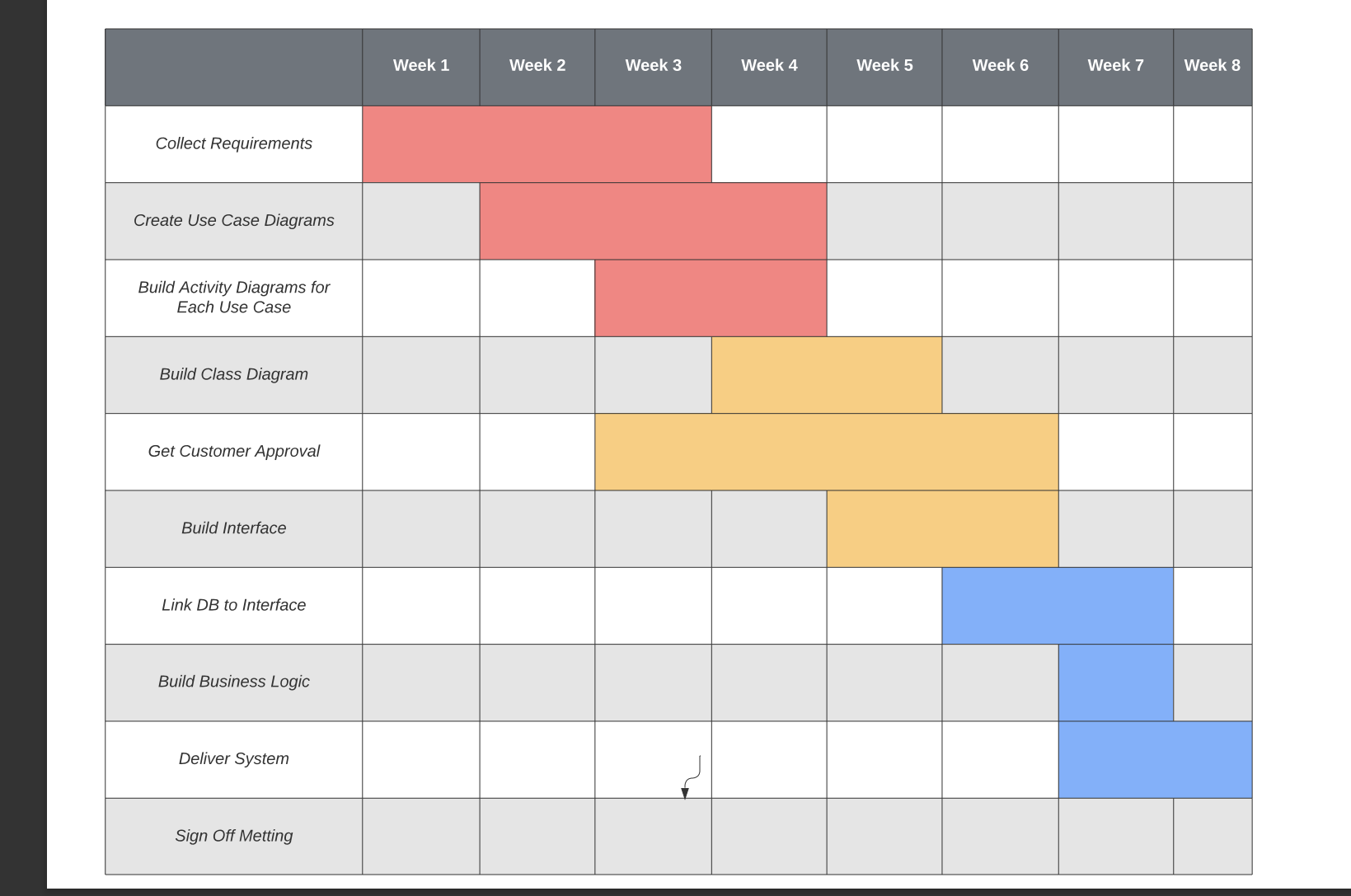
### Limitations

*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?*

* I think the main limitation would be the people running the system itself. Scalability issues can be solved through the cloud, and things like internet connection are out of the control of this system. The main thing that can be controlled is the maintenance and new features. The limitation here will basically come down to the funding of the company.

### 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.*

**