# Meisner 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?*

* Help schedule driving tests
* Help schedule on the road training
* Offer better driving training
* Help educate customers online with different classes
* Help customers take practice tests
* DriverPass, the Owner wants to be able to access his data from anywhere online and be able to download them for offline use that will integrate with excel.
* DriverPass also wants all user actions to be trackable and stored in clear case documentation so they are able to address issues in the case there is a problem and they need to find out who is responsible.
* DriverPass wants to be able to disable packages in the case they do not want customers to register for it.
* DriverPass wants to be connected to the DMV and get notifications when there are updates to the rules
* DriverPasses IT officer needs to have full access over all accounts for resetting passwords or blocking people they do not want using their system.
* DriverPass wants to give the customers the ability to easily reset their own password in the case they need to.
* DriverPass wants the system to run over the cloud and does not want to deal with backup and security but need it to be taken care of.

### 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 improve the educational system they have in order to increase the amount of people that pass their driving tests.
* The different components needed for this system
  + Package Allocation (Customer options)
  + Car and Driver Counter
  + Object oriented architecture of the car, driver, and customer
  + Registration System
  + Online class system
  + Path optimization for drivers to customer locations
* Measurable Tasks

The measurable tasks that need to be included for this system to be implemented consist of:

* Collecting Requirements
* Creating Use Case Diagrams
* Building Activity Diagrams for each Use Case
* Researching User Interface Designs and building Class Diagrams
* Building the interface and linking it to the database
* Building Business Logic
* Testing the system and delivering the product

### 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 system should be able to integrate both the employees and customers on the server so they can schedule different events.
* Some of the measurable tasks that need to be included in the system design
  + Number of Cars
  + Number of Drivers
  + Allotted time per drive
  + Allotted time per test
  + Allotted time per test drive
  + Optimization of scheduling
  + Pathway tracking for drivers to safely and efficiently drive to the customer
  + Customer grade
  + Different training packages

## 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 environment needed to run the software will be integrated into the cloud, so they do not have to deal with backup and security. There may also be a cause to suggest the creation of a web application to act as an interface between the cloud so that DriverPass can locally download and use the data to access it offline.
* When it comes to speed, this system needs to run fast enough to be able to process the number of requests made by users based daily. That means that there always needs to be an amount of computing power made available for their system, and this is dependent on the size of their business. To be more concise, it is always good practice to have more computing power and storage available than needed so your system does not crash. There are also many types of models that allow you to have unlimited amounts of this at a higher price.
* Given that this is a cloud platform and that DriverPass does not want to deal with backup and security they are giving up the ability in my opinion to choose when this does and does not get updated. There are responsible ways to update a system at a time that does not take away from the utilization of the system, so the real question here is when will they not be using their system? Certain platforms choose certain days to do updates on to handle massive amounts of changes to a hosted application, and other companies may choose to do it later at night when nobody else is working. The one issue here though is that in addition to what I mentioned prior about backup and security, they also want to be updated with a notification every time the DMV updates as well, which means that they will have to be reliant on the DMV as they will have to in some way coordinate with the DMV to centralize their updates allowing the latest laws and updates to be correct.

#### 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 platform should run off some type of windows system, the reason I say this is because if their customers are going to be interacting with the system, Windows is the most used, and user-friendly platform out there to choose from. The backend will require tools as well, given that there is a need for developers and system analysts to add and remove modules, Ian, their IT officer needs to be able to help users and employees in the case there is a problem, in addition to blocking users from accessing their server, and definitely a database, as Liam had said he needs to be able to track actions of users and employees, in addition to being able to back data up on the cloud servers which likely helps them make informed decisions about the functions their business offer and how successful they have been as a company. This means collecting user information, whether they passed or failed, class metrics, etc.

#### 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?*

* Distinguishing between different users can be like filling out an input form. This form includes first name, last name, address etc. The importance of this is that we will be able to reduce the redundancy of users using important information like this. Specific ID’s can be made per customer that are allocated to the number of users they are, or even a different indicator that can be decided later such as geo-location, or even a random serial number that is given for the account.
* The input should absolutely be case sensitive, if the input is not case sensitive then you will be causing many security problems because you will not be able to account for a constant when inputting information into your application. This is bad because when your application is intaking input, you want it to be predictable.
* Lastly, this system should inform the admin of a problem if there is any type of collision within the database, for example if two users have the same exact name, or if there are multiples of one account. I think overall looking for irregularity within the patterns of the system is what is most important for informing the admin of any issue but given that he does not want to deal with security, this maybe outside the scope of his concern, so it primarily should only be some technical issues, in addition to user issues.

**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?*

* Yes, you can make changes to the user without changing code, the question is whether the function will be implemented at the delivery date of the application. In addition to this, it cannot be so easy where a non-developer would be able to make these changes, so in the case there is a utilization of code in some way, it will have to be light scripting, or even inputting commands into a terminal application to update the server directly using pre-made functions that take a specified type of input.
* The system itself can adapt to platform updates by utilizing cloud, this means that we can interact directly with the database, for example if you are using a tool such as SQL, the commands you input will directly be interacting with the backend, so you are able to seamlessly update information without having to directly interact with the server and you will not have to deal with any runtime issues.
* Lastly, the IT admin should have the ability to do pretty much anything within the system, but primarily maintaining and modifying the system. The reason Ian needs to have full access is primarily because he can test and fix issues. In order to do this, a common tactic is to isolate the application and run tests on the specific function separate from the running application, perhaps in a different server somewhere so you can make changes without crashing the entire system in the case you make a mistake.

#### 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?*

* Given the circumstances of the application, the primary things needed for a user to log in would be a username and a password. In addition to this, the primary way to secure a login is with email as well but in this case, we could require other information as well in the case there is reason to believe in authenticity per user due to the type of services we are offering.
* The way that we secure the connection and data exchange between the client and the server is through the utilization of something called an SSL, or a Secure Socket Layer. This not only encrypts the data being sent, allowing us to store and communicate sensitive information like credit card numbers or social security numbers, but also utilizes public and private keys to assure that the domains and web browsers that are interacted with are safe to interact with and held to a specified standard.
* As a fun fact, as I had mentioned earlier when it comes to case sensitivity, if the input is case sensitive, this helps protect against brute force attacks. In addition to this, when I mentioned earlier about looking for an irregularity of patterns within a system, this is considerably one of those instances. There are many ways to protect against these types of attacks but one of the best ways is through the education of your users about changing and protecting your passwords. This will help DriverPass communicate efficiently with their customers in case they need help using the system, or if there happens to be a problem with their password. A common way to go about stopping this are with ReCAPTCHA images, in the case the brute force attack is done by some type of bot. In addition to this, if it is an IP address that is not normally the IP the user uses, we could email them to verify their identity, or even have some type of authentication application as a third way of user authentication. As a last resort, we can lock the users account after a certain number of attempts making them have to reset the password over the phone.
* Lastly, and this is one of the most important points, is that we need to educate employees about cyber security, in the case someone attempts to social engineer their way into resetting their password. When it comes to a user forgetting their password, this is when the users access to the system is most vulnerable, mainly because the system put in place in order to reset it can be easily abused if not set up correctly. One way that we can attempt to solve this issue is through the utilization of a personalized key, or code given to the user to hold onto that is unique to them so they may say it over the phone in addition with other security questions such as their mother's maiden name, or their childhood school name. The only issue with this is that the information or the answers to the questions must not be easily accessible online, and it cannot be totally public. There is a huge tradeoff here mainly because if the user forgets these questions, we also have a huge problem too.
* When it comes to users resetting their own passwords, we can set allotted times for time they have to reset their password, in addition to this, we must have a strong password security policy that is consistent and cannot just be a random string of letters and numbers.

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 provide training using online classes and practice tests*
* *The system shall only allow the modification of data while online*
* *The system shall allow downloading of reports to employees while working offline*
* *The system shall allow the access of data from multiple computers and even cellular devices*
* *The system shall allow the IT officer to have full access over accounts to reset passwords or block users*
* *The system shall track user actions such as who made a reservation, canceled a reservation, modified a reservation and allow printing of activity reports*
* *The system shall allow customers to make online reservations for driving lessons, and appointments. In addition to letting them change or alter appointment data pertinent to themselves.*
* *The system will allow three packages for users to choose from, in addition to letting the IT officer, and owner customize, add, and remove or disable them.*
* *The system shall allow users to be able to reset their passwords if needed.*
* *The system shall notify the IT Officer of updates from the DMV when updates are required.*
* *The system shall allow the IT officer to maintain and take care of the system in addition to the identification of any issues flagged.*

### 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.)?*

* It is not determined fully yet what type of interface will be able to be initially interacted with, but for the purposes of the initial creation of the application it will be likely both a browser interface, and an application interface. Browser because of cloud utilization, and application wise for the potential of personal use as I described earlier. Long term, there maybe need for an additional mobile interface as Liam had stated he wanted to be able to access the system with his phone.
* What each user will be able to do with the interface is dependent on the job it is that they have in reference to the system, for example, Ian needs access to everything, Liam needs access to most things, which prompts the creation of a management tool. In addition to this, we have the secretary and other employees such as a developer or a systems analyst that will need to modify future modules. Lastly, we have the customers who need to be able to access and manipulate the data in the way they would like in order to set schedules, register for classes, etc.
* Lastly, the interface is what provides overall functionality to the different users of this system, but what is most important to note is that this may change dependent on your role design wise to optimize it for your specified use-case. Cloud for example will have a different interface opposed to a customer given the access to the types of tools they have.]

### 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?*

Something that was not specifically addressed were the creation of these different tools for different users, or the organization of the differing interfaces for each user. This also is an assumption I am making because logically it would make more sense to have different pages only accessible by these specified users for security purposes so that a customer could not accidentally stumble onto a login for an admin. Another assumption is that every user will have a computer when we initially create this application, it has targets set on being an application that’s usable on your phone too, and that’s doable, but it would also require the users have internet, or are technically literate in some way. The accountability is there for these nontechnical users through the capability of calling in with the secretary. But realistically this may prove challenging in the long term in the case this system becomes necessary for online training and scheduling.

### 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 main limitation within the system design is the utilization of cloud. Usually, I am the #1 person that vouches for cloud and its benefits, but it also has a lot of drawbacks as well. One of these drawbacks is the lack of control we will have on their end, which may affect the system in a massive way in the case that the job done isn’t correctly working or is compromised in some way. We will have to communicate with them and be on their time and that may not be up to standard, especially if they cannot fix the issue right away. In addition to this, there was no mention of any of the drivers having accounts as well or tracking of any kind on their performance or where it is they drive for security purposes. The main point to focus on here is the utilization of inventory tracker. A database was mentioned yes, but we have no way of paying employees with this system, or logging assets such as cars. There maybe need for a financial aspect in this system for this reason to centralize ease of the application in the grand scheme of the business.

### 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.*

