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 purpose of this project is to aid our client, DriverPass, in building a system to provide online classes, practice tests, and scheduling for on-road driving lessons for young drivers who are learning.

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?

- The founder of the company saw a need for quality driving instruction because a lot of students fail their driving tests due to under-preparation.
- The system will have a front-end component which allows students and drivers to access to scheduling information, purchase lessons packages, and access study materials and tests. Employees should also be able to access the reservations system to allow for updates and to leave feedback from on-road driving lessons.
- The system will also have a back-end component which consists of databases housing login information, learning materials (lessons, tests, et cetera), reservation and schedule information, and secure credit card information from purchasing of lessons packages.

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 have a smooth front-end user interface that makes it easy for students to study, as well as schedule and receive feedback on their on-road driving lessons.
- To achieve that, the web interface should handle the ability to offer multiple lesson packages, as well as the employee ability to see and change reservations and purchases and update package offerings in the future.

- To ensure security, different roles will have different levels of access permissions in the system. A login authentication system will make sure that credentials are valid, and the system administrator will receive notifications on who makes what changes while they're logged in.
- Data can be downloaded for offline review by certain users (admin reports, et cetera), but data can only be edited online to minimize risk for redundancy.

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?

- On initial launch, the system will be web-based on the cloud.
- Performance should be quick, as the clients do not want to have to deal with too much of the tech. Backup and security should be a part of the hosting service the web app uses.
- Backups could occur every night as desired by the clients – again, automated to lower administrator work.
- The client would like some future functionality which will require updates, such as the ability to add, modify, and remove lessons modules.

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 intuitively run on most web browser such as Chrome, Safari, and Edge. If the website is launched on a mobile device, the website should automatically optimize for mobile browsing with a streamlined UI and the same functionality.
- The system should be cloud-based, and have built in security and backup abilities to minize administrative time on those tasks.
- The system will require at least one database for storage of learning materials, schedules, login credentials and permissions, and payment information.

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?

- There is a login system which has a username and password. These will be linked to in the database and will be validated upon login. Different users will have different permissions as determined by administrators.
- In case a user forgets their password, their should be an option to reset it, which will alert system administrators. They can then help the user reset their password.

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 user can edit their profile, payment information, and courses they are in.
- IT needs access to manage user logins and permissions, update courses and packages offered, and manage the information database(s).

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 will need a correct username and password to login.
- For added security, having a case-sensitive input would be good, as well as possibly implementing a SSO or 2FA process. Two-factor authentication would work especially well since the service requires the user's phone number for scheduling on-road driving lessons anyway.
- Making sure that the connection between user, the front-end webpage, and the database on the server is encrypted is a must for security.
- If there is a brute-force hacking attempt, we should limit the amount of login attempts a user can make. For example, perhaps they can only try three unsuccessful logins every fifteen minutes. There should be an option that the user can click to reset their password, which will let the admins know to help the user. Also, once a predetermined limit on total amount of unsuccessful logins is reached, the account should lock pending admin intervention.

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 allow users and administrators to schedule and change reservations for on-road driving lessons.
- The system shall not allow offline data editing.
- The system shall validate user credentials when logging in using case-sensitive usernames and password, as well as possibly using SSO or 2FA options.
- The system will show updates made by administrators in the front-end interface (i.e., modified lessons packages, schedules, et cetera).
- The system shall notify administrators when users need to reset their password.
- The system shall notify administrators when users make changes to online reservations.
- The system shall also allow employees to create user reservations and take payments from users who call to set up services.
- The system shall have some method of interface with the DMV to send alerts when official rules and regulations change from the DMV through some type of API.
- The system shall have a feature which allows driving instructors to leave feedback for their students' enrichment.

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 will be web-based and scale on both mobile and desktop environments.
- The users for the interface will be the students, the driving instructors, and the administrators.
- The users will be able to view online lessons and take practice tests, as well as schedule and pay for on-road driving lessons. There will be a small profile feature for them, as well as photos of both themselves and their instructor. They should be able to note any special needs that they have, and track their overall progress.
- The drivers should be able to view their reservations and student profiles, as well as leave driver notes after each on-road lesson so students can learn from the feedback.
- Administrators need to be able to view everything, as well as track all changes and get alerted by any modifications to packages and schedules as needed.

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?

- Some users may not desire a web-based experience, instead favoring a more traditional app experience on Android or iOS.
- Depending on where they live, internet may be unavailable or of poor quality. The website may need to scale depending on quality of internet connection.
- Interfacing with the DMV may be difficult, as I am unfamiliar with any APIs they offer. Building one from scratch may be beyond the initial scope of the project.

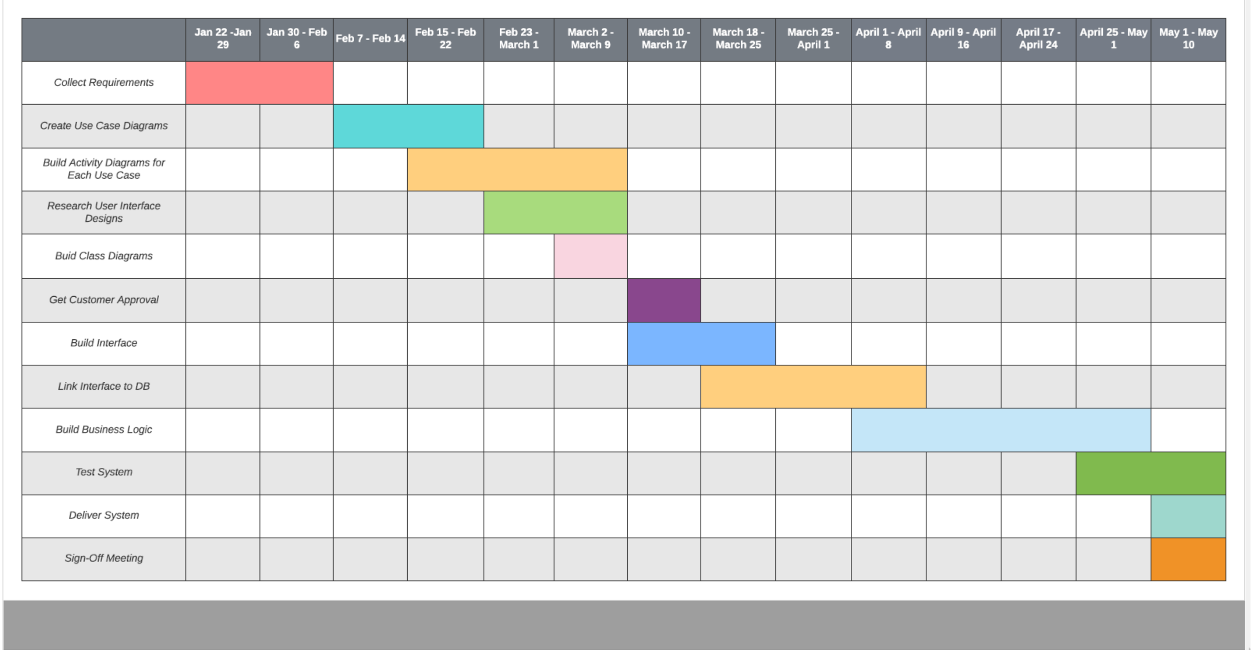
Limitations

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 major limitation will be ensure that the web-app runs the same in most browsers. Theis will require a lot of cross-testing, and the timeline of 15 weeks may be tight.
- Some features are easier than others, while the more difficult ones like the DMV API problem may take up considerable time.
- It also seems like we are working with a small team. If unexpected problems arise, we may not have the manpower to handle it.

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.



	Jan 22 -Jan 29	Jan 30 - Feb 6	Feb 7 - Feb 14	Feb 15 - Feb 22	Feb 23 - March 1	March 2 - March 9	March 10 - March 17	March 18 - March 25	March 25 - April 1	April 1 - April 8	April 9 - April 16	April 17 - April 24	April 25 - May 1	May 1 - May 10
Collect Requirements	■	■												
Create Use Case Diagrams			■	■										
Build Activity Diagrams for Each Use Case				■	■	■								
Research User Interface Designs					■	■								
Buid Class Diagrams						■								
Get Customer Approval							■							
Build Interface							■	■						
Link Interface to DB								■	■	■				
Build Business Logic										■	■	■	■	
Test System													■	■
Deliver System														■
Sign-Off Meeting														■