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 help students prepare for the driving test at their local DMV. The client is Liam, the owner of DriverPass. Liam wants the system to be able to help students study for the written driving test as well as offer behind the wheel lessons the students can book through the website.

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?

- Liam, the owner of DriverPass, wants the system to help him access his data online from any computer or mobile device so he is able to work from anywhere. He also needs to have full access to all accounts to reset passwords or remove people's access to the system. Liam would also like to be able to track reservations and print activity reports in case there's an incident. The problem Liam would like to fix with DriverPass is the fail rate of students who are taking drivers tests at the DMV. He has noticed a need for better drivers' education. The different components needed for this system are password reset, online booking, ability to add and remove packages, online learning, appointment tracking, and a security system.

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?

- When it is complete, the system should allow students to create an account, study, take practice tests, purchase different training packages, and book behind the wheel lessons. Some measurable tasks that need to be included to achieve this are encryption for payment processing, so sensitive credit card information is secure when students purchase packages, login and password reset features, online course access, and tracking of student data.

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 should be accessible from any device, and reports should be downloadable from any location. The system should also be cloud based to avoid DriverPass having to deal with backup and security. The system should be updated often to reflect the DMV's current policies.

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?

- Because the system needs to be cloud based, it needs to be able to run on any platform, like Windows or Mac, and any device, like mobile devices, laptops, or desktops. The backend will require a database to manage data relating to payments, reservations, clients, and tests.

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 needs to be access control, so different employees have different rights and roles, so they are distinguishable based on what they have permission to access in the system. The IT officer should have access to all accounts so he can revoke access to the system and reset case-sensitive passwords if employees forget or they get locked out after entering their password incorrectly too many times.

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 owner would like to be able to easily remove packages without needing to change the code. Because the system will be cloud-based, platform updates should be automatic. The IT admin needs complete access to be able to revoke access from former users and manage user accounts and passwords in the event someone gets locked out or forgets their password.

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?

- For the user to be able to log in, the user will need to enter a username and password, and they will have access to things based on permissions set for their role. Having HTTPS encryption, multi-factor authentication, and role-based access is a great way to make sure the connection between the client and the server is secure. In the event of a brute force hacking attempt, an account should be locked temporarily, and admin should be notified. If a user forgets their password, they should be able to reset it automatically.

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 be able to access data from anywhere, online, and offline.
- The system shall be compatible with any device.
- The system shall have reservation tracking so admin can see when users make appointments, change appointment times, and cancel appointments.
- The system shall allow users to reset passwords automatically.
- The system shall allow the admin to print reports in case something goes wrong.
- The system shall allow users to make reservations online.
- The system shall be flexible so the admin can remove, add, or modify different packages.
- The system shall update automatically to comply with DMV policies.

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 should be accessible through both mobile and desktop. The different users of this interface are the owner of DriverPass, the IT officer, the secretary, and the driving students. Both the owner and the IT officer need full access to the whole system in order to manage accounts and download reports. The secretary will need to be able to make and modify reservations and view and select different packages according to the students' needs. Students should also be able to view and select different packages as well as be able to make and modify reservations online. The system should be scalable so users can access it through mobile screens like they would a destop.

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?

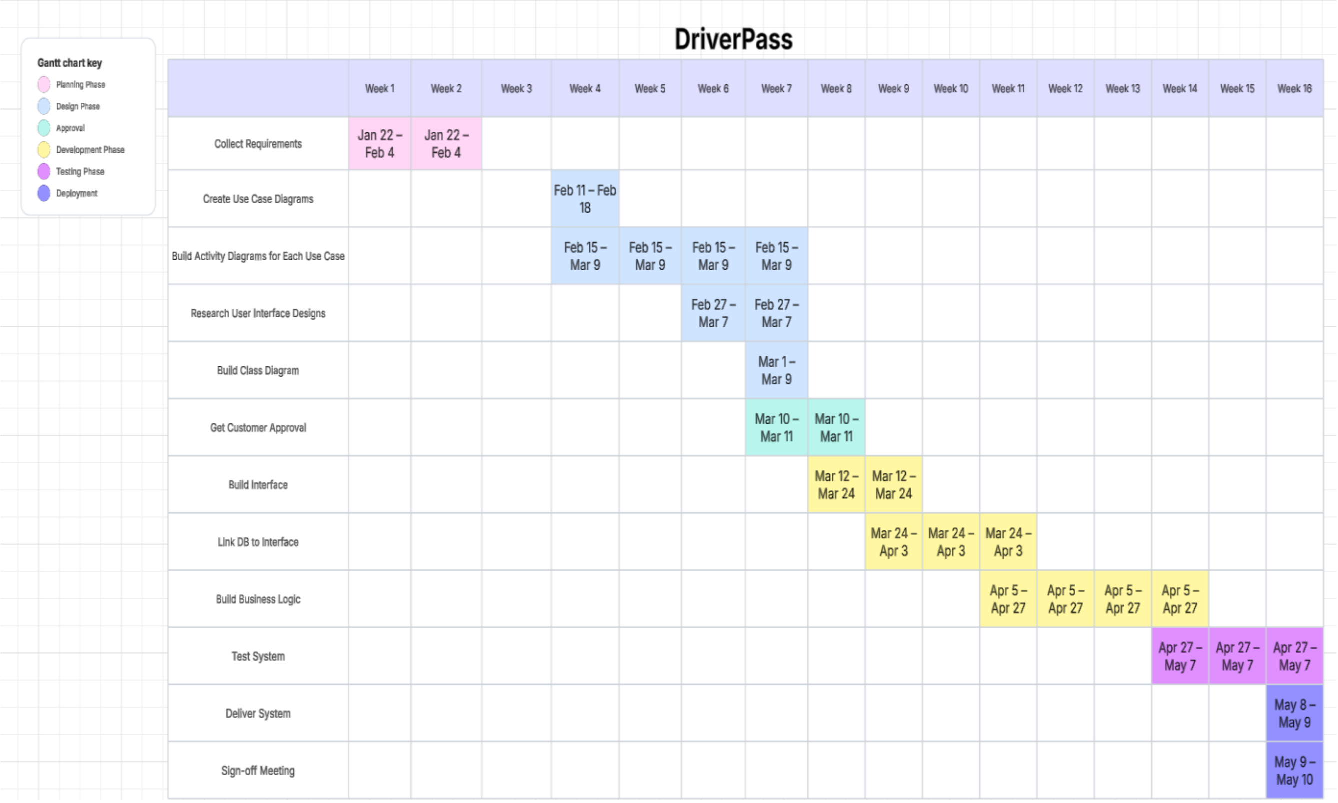
- Even though it is not explicitly stated, I am assuming that all users are able to access the system anytime they need if they are able to connect to the internet. I am also assuming that users will be able to reset their own passwords through their email addresses. I am assuming that all users have a device they can access the system through, phone or desktop.

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?

- A limitation I see in the current system design is that while administrators will be able to enable or disable packages, without a system analyst, they won't be able to fully add or remove packages from the system completely. So, at the start of the project the flexibility of the system will be greatly limited.

Gantt Chart



DriverPass

Gantt chart key
- ● Planning Phase
- ● Design Phase
- ● Approval
- ● Development Phase
- ● Testing Phase
- ● Deployment

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16
Collect Requirements	Jan 22 – Feb 4	Jan 22 – Feb 4														
Create Use Case Diagrams				Feb 11 – Feb 18												
Build Activity Diagrams for Each Use Case				Feb 15 – Mar 9	Feb 15 – Mar 9	Feb 15 – Mar 9	Feb 15 – Mar 9									
Research User Interface Designs						Feb 27 – Mar 7	Feb 27 – Mar 7									
Build Class Diagram							Mar 1 – Mar 9									
Get Customer Approval							Mar 10 – Mar 11	Mar 10 – Mar 11								
Build Interface								Mar 12 – Mar 24	Mar 12 – Mar 24							
Link DB to Interface									Mar 24 – Apr 3	Mar 24 – Apr 3	Mar 24 – Apr 3					
Build Business Logic											Apr 5 – Apr 27	Apr 5 – Apr 27	Apr 5 – Apr 27	Apr 5 – Apr 27		
Test System														Apr 27 – May 7	Apr 27 – May 7	Apr 27 – May 7
Deliver System																May 8 – May 9
Sign-off Meeting																May 9 – May 10