Southern New Hampshire University

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 provide people with better driver training. Liam has seen many people fail their driving test at the DMV and wants to create this system to provide a more comprehensive education to driving students that will help them pass their tests. The majority of the educational resources will be provided online and will provide on-the-road training on a per-need basis.

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?

- Reporting - DriverPass needs the system to be able to download reports online or offline for quick access to business data.
- Security - DriverPass needs a way to modify privileges for all different positions in the company. This will allow IT to cut off access to people who have been let go or modify privileges for people starting new positions.
- Compliance - DrivePass will need to connect with the DMV to keep their tests up to date with DMV standards.
- Scheduling - DrivePass needs a system that allows students to sign up for an appointment the same way that they do over the phone to help manage call volume.
- Tracking - DrivePass will need the system to be able to track where each of their 10 cars is and when they are scheduled in the future. They also must be able to see which driver is scheduled which student and when.
- Packages - DriverPass wants to be able to disable a package on their system. They'd also like to add, remove, or modify packages in the future (future release).
- Interface - The system should contain a visual progress dashboard that contains the student's online test progress, notes from the driver, basic information, special needs, student photo, and driver photo.

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?

- Liam requires the system to allow him to access the driver training data online from any computer or mobile device. He needs to be able to download this data into a formatted CSV.
- Liam mentioned having the data available offline as well. However, this data will be view-only from offline devices.
- Ian notes that there are many different roles in the company that has different rights and roles. He will need to have full control over these accounts to manage privileges.
- The system will need to track when a person creates, removes, or modifies a driving reservation and print out an activity report.
- Customers need to be able to reserve a driving lesson through their online accounts. The information needed from the customer includes the first name, last name, address, phone number, state, pickup location (same as drop-off location), and credit card number, expiration date, and security code. This will allow customers to make appointments the same way that they do over the phone.
- The system also needs to be able to track which driver the customer is scheduled with including the time and the car. This will help DriverPass keep track of all of its drivers and students.
- The system should be run on a cloud-based architecture over the web and outsource backup and security measures to focus solely on the business.
- The system needs to connect to the DMV so that DriverPass gets a notification each time the DMV modifies one of its standards. This will help them keep their tests current with DMV standards.
- The system should contain a visual progress dashboard that contains the student's online test progress, notes from the driver, basic information, special needs, student photo, and driver photo.
- The system will also include a way for the student to contact DrivePass and vice versa.

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 will be a web-based application with a mobile-friendly interface.
- The site supporting 5 thousand users per hour must load within 10 seconds on mobile and desktop devices in Chrome, Safari, and Firefox browsers.
- All text and images must be loaded on the site within 10 seconds.
- The system will implement any security or minor version updates once a week on the lowest traffic day at 2 a.m. EST.
- Version updates on the system will be evaluated and implemented if necessary at the end of every month by the development team.

- The system will be developed in a cloud-based microservices architecture using Docker containers for each microservice.

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 will be able to run on the three major operating systems: Linux, macOS, and Windows.
- The system will be hosted on a cloud-based Ubuntu (Linux) server through AWS.
- The system will require a SQL database to store user data.
- The SQL database will be backed up every day at 2 a.m. EST.

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?

- Each user will receive a user id that corresponds to the id of their row of data in the database.
- Input is not case-sensitive.
- There will be a range of answers for each fill-in-the-blank question.
- The system should inform the admin of a problem immediately via email.

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?

- We will build an internal API that will connect to the database and allow for modifications to user data without changing the code.
- The Docker containers will allow our system to perform optimally regardless of platform updates.
- The IT admin will have full access to all user data and the ability to add, remove, or edit any user in the system's database.

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 system's database must adhere to HIPPA standards.
- The system will require a username and password to log in
- Passwords must be at least 15 characters long and must include 1 letter, 1 number, and 1 special character.
- The passwords will be encrypted before they are saved in the database. The only way to unlock the user's account is to have the user's secret key. Database admins will only be able to see encrypted hashes as passwords.
- If a user incorrectly submits a password 10 times in a row, the account will be locked and the user must call IT to support to unlock the account.

- If the user forgets their username, the user can input an email address and the system will send an email with the correct username.
- If the user forgets their password, the user can input an email address and the system will send an email with a link to reset their password.

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 store multiple quizzes and training exercises
- The system shall store information on multiple different users
- The system shall display information based on the user that has logged in
- The system shall display the student's training statistics on a personalized dashboard
- The system shall allow for admin to download driver training data into a formatted CSV
- The system shall allow admins to view driver data while offline
- The system shall have one super admin who can control the system privileges of all employees at DriverPass
- The system shall track when a person creates, removes, or modifies a driving reservation
- The system shall be able to print out an activity report
- The system shall be able to reserve a driving lesson through their online accounts using the same information gathered during phone inquiries
- The system shall track a driver if they are currently with a student and which car they are in
- The system shall show the drivers' upcoming appointments and which car they will be using
- The system shall be able to display a car and driver that is available for a time slot that is gathered from student input
- The system must send emails to DriverPass employees when the DMV modifies its standards
- The system shall contain a visual progress dashboard that contains the student's online test progress, notes from the driver, basic information, special needs, student photo, and driver photo
- The system shall include a way for the student to contact DrivePass and vice versa
- The system shall store special needs for each student
- The system shall store online test progress to the student
- The system shall be able to store driver notes about a student
- The system shall be able to handle image uploads

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.)?

- For students, the system shall have a dashboard on the homepage that displays the DriverPass logo, student information, driver notes (including lesson time, start hour, end hour, and driver comments), special needs (if applicable to the student), a photo of the student's driver, and a photo of the student

- For students, the system shall have a form where the student can fill out their s first name, last name, address, etc
- For students, their personal information shall be editable in a settings form
- For students, the system shall display a contact page that gives instructions for how they can contact DriverPass
- For students, the system shall have an interface that shows them a concise view of all previous drives
- For drivers, the system shall display the drives that they are scheduled to take including the student and car information, appointment start time, appointment end time, start location, and end location
- For drivers, the system shall email a link to a form after each drive to collect the lesson time, start hour, end hour, and comments
- For drivers, the system shall have an interface where they can see all of their past drives and have the ability to edit the comments with a form
- For drivers, the system shall have an interactive calendar where they can enter their work availability
- For admins, the system shall display all drives including driver name, student name, start location, end location, car, lesson time, start hour, and end hour
- For admins, the system shall display general contact information on each student
- For admins, the system shall be able to filter past drives by the student, driver, car, and/or time period
- For admins, the system shall contain data visualizations that show student progress on both online and driving tests
- For admins, the system shall have forms to add/remove/edit drivers and students
- The system shall be a web-based application with a mobile-friendly interface

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 server has an internet connection at all time
- The users and drivers shall have access to the internet
- Importing existing user information is out of scope
- All accounts will be accessible 24/7
- There is no existing codebase
- The development team can pick the best programming languages and tools based on the project's requirements

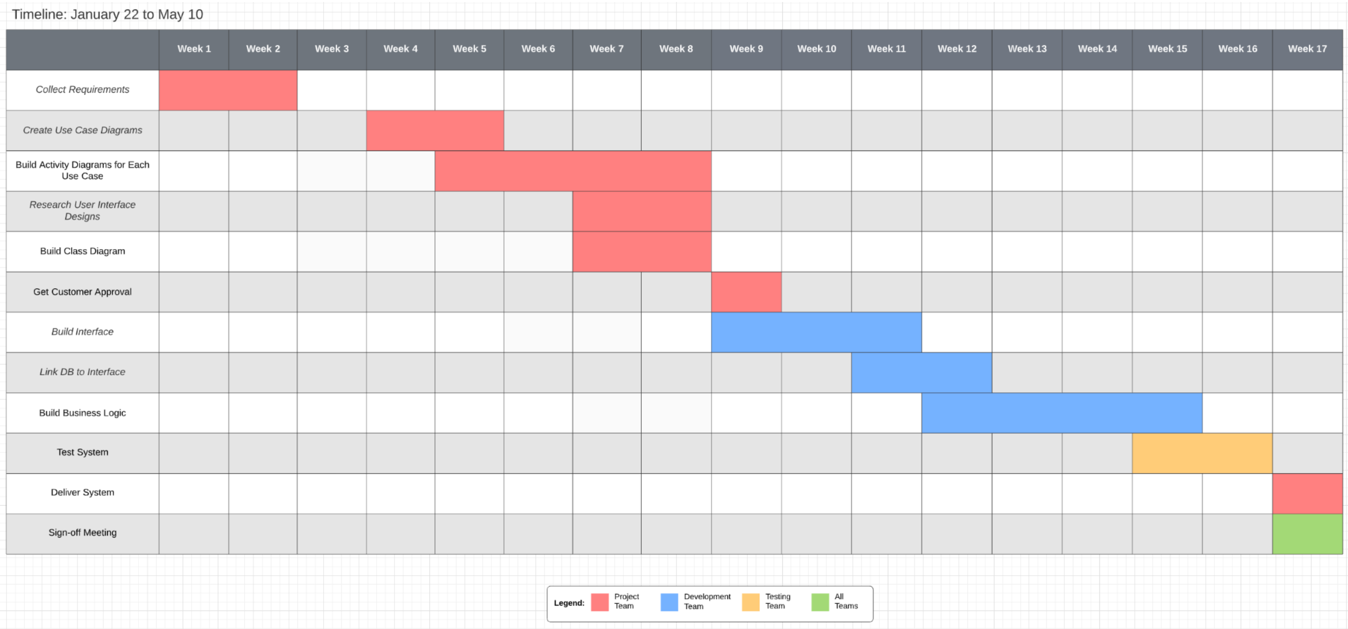
Limitations

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?

- Development time is an estimate, meaning that it could take more time than allotted
- A delay in customer approval on user cases and class diagrams may delay the final timeline
- The timeline may vary depending on how many developers we have available during the development timeframe
- The budget will determine how fast of a server we can host this project on

- If a developer on the team isn't familiar with the chosen tech stack, they will need additional time to learn

Gantt Chart



Timeline: January 22 to May 10

	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	Week 17
Collect Requirements	🟥	🟥															
Create Use Case Diagrams				🟥	🟥												
Build Activity Diagrams for Each Use Case					🟥	🟥	🟥	🟥									
Research User Interface Designs							🟥	🟥									
Build Class Diagram							🟥	🟥									
Get Customer Approval									🟥								
Build Interface									🟦	🟦	🟦						
Link DB to Interface											🟦	🟦					
Build Business Logic												🟦	🟦	🟦	🟦		
Test System															🟨	🟨	
Deliver System																	🟥
Sign-off Meeting																	🟩

Legend: 🟥 Project Team 🟦 Development Team 🟨 Testing Team 🟩 All Teams