Southern New Hampshire University

# CS 255 Business Requirements Document

## 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?*

- Liam, the owner of DriverPass has noticed people are failing their driving test since there isn't a good service in place to educate and accommodate these future drivers. Students can make reservations for an online session or in-person session. Additionally, the system will need to have administrative accounts so that can access reports and other important information.

### 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?*

- Provide appropriate up-to-date driving education
- Access data online and offline
- Online classroom
- Online test taking capability
- Personal training session
    - Ability to pick a package
- Administrative account
    - Reset passwords
    - Block accounts
    - User report
- Reservation system
- Reservation Report system
- User's object

### 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 student should be able to make a reservation for an online class or in-person driving trainer. Additionally, the student should be able to take online test and have up-to-date information from the DMV. The student will also have access to their profile and should be able to track progress.
- An administrator should be able to have access to the system online or offline. Additionally, the admin will need to be able to manage user's password resets, reservations, and personalize training data.
- There are two essential measurable tasks. First, a user should be able to create an account, make a reservation, and attend that reservation. This task is primarily focused on the user's experience. Lastly, the administrator task needs to have access to the system, download reports, reset passwords, and handle reservations.

## 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?*

- This system is a web-based system.
    - Mobile apps for iOS, Android, and Linux devices should be very easy since it will bring up the web-mobile version of the site within the app.
    - The website needs a strong mobile design that can accommodate screen with notches, and punch hole cameras.
- Once a reservation has been created the website needs to reflect that updated information for all users as soon as possible.
- The information from the DMV also needs to be updated as soon as possible.

**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 will support all platforms since it is a web-app
- Yes, a database is required for populating data on the website. A database for users, and a database for reservations

**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 have a unique online username. The username is case-sensitive, as to ensure that all usernames are unique.
- If a user attempts to user a username that is already taken than it will alert the user automatically that the username is already taken.
- The system should notify the administrators if there are any duplicate usernames since that shouldn't be allowed.

**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, there should be code put with administrator object to allow them to already add, remove, and modify users' data.

- Updating the web-app allows for an always updated experience. Since users wouldn't need to download an update since the content is dynamically changing and being reflected on the website.

- Ian is going to maintain the system, have access to modify it, and updating it.

**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 will need an authentication system for verifying users. Additionally, the system will be using HTTPS, which is a secure version of the HTTP protocol that specifically uses the SSL/TLS protocols for encryption and authentication.
- The system will have a minimum required characters, special characters, numbers, and symbols. This password standard should mitigate any brute force attack since the passwords will have to be to long to attempt to brute force.
- The user will be able to reset their password by submitting their email and answer security questions based on their initial profile setup.

**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 the user's credentials when logging in.
- The system shall allow users to update their package planned.
- The system shall allow users to take practice exams.
- The system shall allow users to update their information.
- The system shall allow users to reset their password.
- The system shall allow users to view their progress.
  The system shall allow administrators to download reports.
- The system shall allow administrators to modify data on the system.
- The system shall allow administrators to download/update reservations

**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 have three main layouts. The first is a desktop version of the website, then a tablet version, lastly, a mobile version of the website. These three layouts will encompass all users. Certain considerations should be for the mobile layout of the website. Some phone displays have notches and punch-hole cameras that obstruct content on the upper part of the display. The mobile layout should take that into consideration.

- The user should be able to create an account, reset password, update personal information, pick a packaged plan, make a reservation, modify an existing reservation, take a test, and view study material.

**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 user knows how to use the internet.
- The user should know how to download the app from the mobile store.

**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?*

- For the mobile layout, I can see the screen size being a limitation. Since know the design teams needs to figure out the best way to deliver content without over cluttering the screen with information.
- Additional time for testing and debugging.

**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.*

Southern New Hampshire University

Legend: Design Analysis | Development | Testing Approval

# Gantt chart

Sergio H. Passos | September 22, 2021

| | January | February | March | April | May |
|---|---|---|---|---|---|
| Collect Requirements | Start: Jan 22 | Finish: Feb 4 | | | |
| Create Use Case Diagrams | | Start: Feb 11<br>Finish: Feb 18 | | | |
| Build Activity Diagrams for Each Use Case | | Start: Feb 15 | Finish: Mar 9 | | |
| Research User Interface Designs*Task 4* | | Start: Feb 27 | Finish: Mar 7 | | |
| Build Class Diagram | | | Start: March 1<br>Finish: March 9 | | |
| Get Customer Approval | | | Start: March 10<br>Finish: March 11 | | |
| Build Interface | | | Start: March 12<br>Finish: March 24 | | |
| Link DB to Interface | | | Start: March 24 | Finish: April 3 | |
| Build Business Logic | | | | Start: April 5<br>Finish: April 27 | |
| Test System | | | | Start: April 27 | Finish: May 7 |
| Deliver System | | | | | Start: May 8<br>Finish: May 9 |
| Sign-off Meeting | | | | | Start: May 9<br>Finish: May 10 |