

---

**Vanier College**  
**Faculty of Science and Technology**  
**System Development**  
**420-436-VA**

**Deliverable 4**

**Brown Team**

Team Leader: [Nabil Ramadan](#)

Sadaf Zakria

Hibba Qaraman

Peter Isaac Fishman

*In collaboration with **Justin Eberwein***

---

**Next Team Leader:** [Sadaf Zakria](#)

**We, the Brown Team, certify that this assignment is our own work**

I, Sadaf Zakria, [2151361], certify that I have contributed to this deliverable, S. Z.

I, Nabil Ramadan, [2195270], certify that I have contributed to this deliverable, N.R.

I, Hibba Qaraman, [1760010], certify that I have contributed to this deliverable, H.Q.

I, Peter Isaac Fishman, [1980427], certify that I have contributed to this deliverable, P.I.F.

## Table of Contents

---

<b>Executive Overview</b>	<b>4</b>
<b>Summary of the client</b>	<b>5</b>
<b>Business problem</b>	<b>6</b>
<b>Our way to resolve these problems:</b>	<b>6</b>
<b>Narrative description of system</b>	<b>7</b>
<b>Revised Flowchart</b>	<b>9</b>
<b>Appendix 1 &amp; 2: User Stories &amp; Acceptance Tests</b>	<b>10</b>
<b>User Stories Overview</b>	<b>10</b>
<b>Login/Register interface</b>	<b>10</b>
<b>Client interface</b>	<b>12</b>
<b>Admin interface</b>	<b>15</b>
<b>Appendix 3 - User Story Map</b>	<b>18</b>
<b>Client interface</b>	<b>19</b>
<b>Admin interface</b>	<b>19</b>
<b>Project Plan</b>	<b>20</b>

## Executive Overview

---

The following overview highlights the significant progress made in Deliverable 4, focusing on the development of user stories, test criteria, user map to represent how the client and admin will be using the web application to access different things.

We create these user stories/test criteria/user mapping based on the demand our client has told us so that it may be according to his liking.

Justin Eberwein, founder of "Just B Fitness," established the business in 2016. As a small organization, Justin has successfully managed a clientele of over 100 clients without a dedicated database management system. This has led to challenges in scheduling appointments efficiently.

The primary goal of this project is to upgrade the management of "Just B Fitness" by creating a booking web application. This application will ease the booking process for Justin and his clientele, offering a more efficient and convenient scheduling experience. Additionally, the platform will serve as a foundation for the expansion of services beyond fitness training sessions, and also potentially allow future trainers to book, and add their own clients.

## Summary of the client

---

Justin's journey in the fitness industry began in 2016 when he founded Just B Fitness. Back then, his online presence was limited to Facebook and Instagram, until 2020 when he launched his own website. Although the website was created by Silo, who is no longer part of the team, Justin can still rely on him for any necessary website modifications. Today, Justin operates independently but maintains valuable connections with former colleagues Francesca and Ian. He conducts his client sessions at Monster Gym, renting their space for his fitness endeavors. Notably, Justin has collaborated with fitness models throughout his career, as evidenced by the glowing testimonials on his webpage. Despite his lack of computer and programming skills, Justin manages a substantial clientele of over 100 individuals, without the help of a database. His work days span from 6:30 AM to 11:30 PM, Monday through Saturday, with Sundays dedicated to scheduling sessions with clients individually. To streamline his clients' fitness journeys, he relies on the FitLog mobile app, using it to monitor their health and fitness goals. Additionally, Justin collaborates with nutritionists to offer tailored diet plans through the FitLog app, ensuring a comprehensive approach to his client's well-being.

## Business problem

---

The new booking web application is poised to address critical business challenges that have been plaguing Justin Eberwein's "Just B Fitness." Firstly, the absence of a dedicated database and the reliance on a manual client management process has led to inefficiencies in scheduling and information management. Justin currently manages a substantial clientele of over 100 clients, and this manual approach has proven to be both time-consuming and prone to errors. **Without a centralized data repository, client information is scattered and challenging to access, update, and utilize effectively. This reduces the overall operational efficiency of the fitness business, increases the risk of scheduling conflicts, and can result in missed appointments and frustrated clients.**

Secondly, Justin's exhaustive working hours, which extend from 6:30 AM to 11:30 PM, Monday to Saturday, present a considerable business challenge. The current booking process consumes his entire Sundays as he schedules sessions individually with clients, and currently uses a third-party app "FitLog"<sup>1</sup> to provide dietary recommendations to his clients. This not only takes a toll on his personal well-being but also limits his capacity to focus on other vital aspects of growing and managing his fitness business. It is unsustainable in the long term, and Justin's dedication to his clients should be channeled into more strategic and growth-oriented activities.

### Our way to resolve these problems:

The new booking web application is expected to alleviate the strain on Justin's schedule, allowing him to optimize his time and energy for broader business development while ensuring a more efficient and client-friendly booking process.

The following application will be used by both the admin (Justin) and his clientele. The app will allow clients to create an account, create/cancel a booking based on the admin's (Justin) availability, and view their booking history. The admin Justin will be able to modify his availability, view all his previous and current bookings, cancel bookings, and view all client information.

---

<sup>1</sup> FitLog is a mobile app for managing coaching that lets businesses make personalized workout plans, meal schedules, supplement recommendations and other things.

## Narrative description of system

---

In this dynamic booking system, three primary roles have been identified, each playing a crucial role in the user experience.

The user role serves as the generic person, encompassing anyone (Viewer) engaging with the system. This includes individuals attempting to navigate the system without prior authentication, as well as those seeking to perform basic actions such as login, sign up, or log out. The actions available to an unprivileged user are deliberately kept minimal.. However, they lack the ability to make bookings or maintain profiles. Attempts to sign up or log in are contingent on whether they possess an existing account. During the sign-up process, a unique email is a requisite. This ensures a complete sign-up process. Concurrently, users must input their chosen password twice, with both entries being identical. Upon providing the correct credentials, a viewer transitions into a client. Although clients retain an unprivileged status, their capabilities surpass those of viewers.

Clients gain access to comprehensive website information, the ability to create and edit their profiles, and the authority to initiate bookings. This entails selecting a service, providing contact information, and specifying the preferred time and date. The time and dates available are setup by users with the administrator role. Clients also have the option to include additional notes during the booking process, such as the service they would like provided. The final step involves confirming the booking. Booking the reservation with the trainer (Admin) will then send a confirmation to both users. The client, as well as the trainer selected will both get an email confirmation that will validate the time, and date that was selected.

Admins have the authority to edit what is displayed on the website, and possess the ability to access important information such as client profiles, bookings, and services. Upon authentication, a user identified as an administrator officially assumes this role. Administrators are given critical permissions, allowing them to create, update, and delete services. This encompasses the ability to add, remove, or modify a service. The admin will be able to go to a dashboard that will show a calendar of all the working hours available. They will then be able to select available times that the clients will be able to select for their sessions. Administrators are also granted read permissions to access all client profiles and associated information. The Admin will be able to select any of his active clients, and view a history of their passed sessions, the clients contact information an more. A pivotal ability of an administrator is the rapid access to booking information. The system offers various tools for administrators to efficiently manage bookings. They can sort bookings alphabetically by client name or date, facilitating operations. In this board, they will also be able to see which time slots have been filled in, and by what client. Seeing the admin already has a

confirmation of this from an email they can choose to keep it, or if need be, select the time slot and cancel the session, which will then once again send a confirmation of the cancellation to both the client, and the admin.

Clients are individuals with restricted privileges, with accounts on the website and the capability to create bookings. While their access is limited compared to administrators, clients are able to set up bookings, providing them with a tailored experience. On the contrary pre-registered Admin accounts will be able to edit all detail including, time slots, as well as cancellations of bookings, and also be able to view all client accounts. Both these user roles will play hand in hand to make sure our booking application works as planned.



# Booking Web Application

```
graph TD
    START([START]) --> ClientAdmin{Client/Admin}
    ClientAdmin -- ADMIN --> LoginAdmin[Login]
    LoginAdmin --> VerifyEmailPassAdmin[/Verify Email + Password/]
    VerifyEmailPassAdmin --> IsEmailPassValidAdmin{Is Email + Password valid?}
    IsEmailPassValidAdmin -- YES --> ViewBookingsAdmin[View Bookings<br/>(previous + current)]
    IsEmailPassValidAdmin -- YES --> ModifyAvailability[Modify Availability]
    IsEmailPassValidAdmin -- YES --> CancelBooking[Cancel Booking(s)]
    IsEmailPassValidAdmin -- YES --> ViewClientInfo[View Client Info.]
    IsEmailPassValidAdmin -- YES --> LogoutAdmin[Logout]
    ViewBookingsAdmin --> END1([END])
    ModifyAvailability --> END1
    CancelBooking --> END1
    ViewClientInfo --> END1
    LogoutAdmin --> END1

    ClientAdmin -- CLIENT --> NewUser{New User?}
    NewUser -- YES --> Register[Register]
    Register --> EmailAlreadyInDB{Email already<br/>in database?}
    EmailAlreadyInDB -- NO --> SendUserInfo[/Send user info.<br/>to database/]
    SendUserInfo --> END2([END])
    EmailAlreadyInDB -- YES --> NewUser
    NewUser -- NO --> LoginClient[Login]
    LoginClient --> EnterEmailPass[/Enter Email +<br/>Password/]
    EnterEmailPass --> IsEmailPassValidClient{Is Email + Password valid?}
    IsEmailPassValidClient -- YES --> ViewBookingsClient[View Bookings<br/>(previous + current)]
    ViewBookingsClient --> END3([END])
    IsEmailPassValidClient -- NO --> ForgotPass{Forgot Password?}
    ForgotPass -- YES --> EnterPhoneEmailDOB[/Enter Phone<br/>Number, Email, and<br/>Date of Birth/]
    EnterPhoneEmailDOB --> IsEmailDOBValid{Is Email +<br/>DOB valid?}
    IsEmailDOBValid -- YES --> ResetPass[Reset Password,<br/>enter a new<br/>password]
    ResetPass --> UpdateUserInfo[/Update user info.<br/>in database/]
    UpdateUserInfo --> END4([END])
    IsEmailDOBValid -- NO --> ForgotPass
    ForgotPass -- NO --> LoginClient
```

The flowchart illustrates the logic for a Booking Web Application, starting from a 'START' terminal and branching into several user interaction paths. The main paths include: 1. \*\*Client/Admin Login Path\*\*: Starts with a decision 'Client/Admin'. If 'ADMIN', it proceeds to 'Login', then 'Verify Email + Password', and a decision 'Is Email + Password valid?'. If valid, it allows actions like 'View Bookings', 'Modify Availability', 'Cancel Booking(s)', 'View Client Info.', and 'Logout' before reaching 'END'. 2. \*\*New User Registration Path\*\*: If 'CLIENT', it asks 'New User?'. If 'YES', it goes to 'Register', then 'Email already in database?'. If not in the database, it 'Send user info. to database' and ends. If already in the database, it loops back to 'New User?'. If 'NO', it goes to 'Login'. 3. \*\*Forgot Password Path\*\*: From 'Login', it asks 'Forgot Password?'. If 'YES', it requires 'Enter Phone Number, Email, and Date of Birth', then checks 'Is Email + DOB valid?'. If valid, it allows 'Reset Password' and 'Update user info.' before ending. If 'NO', it loops back to 'Forgot Password?'. 4. \*\*Standard User Login Path\*\*: From 'Login', it asks 'Enter Email + Password', then checks 'Is Email + Password valid?'. If valid, it allows 'View Bookings' and ends. If not valid, it loops back to 'Forgot Password?'. 5. \*\*Booking Management Path\*\*: From 'View Bookings', it branches into 'Make a booking', 'Update login credentials', 'Cancel Booking', and 'Logout', each leading to a specific sequence of actions (like choosing options, confirming, or removing data) before reaching 'END'.

## Appendix 1 & 2: User Stories & Acceptance Tests

### User Stories Overview

To obtain user stories for our booking application, we based them on the client's needs and interview responses. These user stories cover a booking system with a admin dashboard equipped with essential functionalities such as the ability to view client bookings, modify availability, cancel bookings, and access client information. For clients, the user stories involve creating bookings, viewing past and future bookings, updating login credentials, and the ability to cancel bookings when necessary. The design format remains adaptable for the prototype.

### Login/Register interface

	As a	I want to	So that	Test criteria
1.	Client	Register	Client can access JustBFitness booking application.	<p><b>Accepted:</b>  <b>Given:</b> The Client wants to register.  <b>When:</b> The Client provides all required registration information accurately.  <b>Then:</b> The Client's registration is successfully processed, and they can access the booking application.</p> <p><b>Fails:</b>  <b>Given:</b> The Client wants to register.  <b>When:</b> The Client attempts to register but provides incorrect or incomplete registration information.  <b>Then:</b> The Client is prompted with a message indicating that their registration attempt was not successful.</p> <p><b>Fails:</b>  <b>Given:</b> The Client wants to register.  <b>When:</b> The Client attempts to register with registration information that matches an existing account in the system.</p>

				<p><b>Then:</b> The Client is prompted with a message indicating that their registration attempt was not successful because the credentials are already in use by another account.</p>
2.	Client	Login	Client can access their account.	<p><b>Accepted:</b>  <b>Given:</b> The Client wants to login.  <b>When:</b> The Client enters the correct credentials.  <b>Then:</b> The Client successfully logs in.</p> <p><b>Fails:</b>  <b>Given:</b> The Client wants to login.  <b>When:</b> The Client enters the wrong credentials.  <b>Then:</b> The Client is prompted with invalid credentials.</p>
3.	Client	Reset password	Client can access their account.	<p><b>Accepted:</b>  <b>Given:</b> The Client forgot their password.  <b>When:</b> The Client resets the password by entering the correct identity verification information.  <b>Then:</b> The Client successfully updates account's password.</p> <p><b>Fails:</b>  <b>Given:</b> The Client forgot their password.  <b>When:</b> The Client resets the password but enters incorrect information  <b>Then:</b> The Client is prompted with a message indicating that their attempt was not successful.</p>
4.	Client	Logout	Client can securely leave the application by signing out from their account.	<p><b>Accepted:</b>  <b>Given:</b> The Client is logged into their account.  <b>When:</b> The Client initiates the logout process by signing out.  <b>Then:</b> The Client successfully exits the application, and their session is securely ended.</p> <p><b>Fails:</b></p>

				<b>Given:</b> The Client is logged into their account. <b>When:</b> The Client attempts to sign out. <b>Then:</b> The Client is prompted with a message indicating that the logout attempt was not successful.
5.	Admin	Login	Admin can access the administrator account.	<b>Accepted:</b> <b>Given:</b> The Admin wants to login. <b>When:</b> The Admin enters the correct credentials. <b>Then:</b> The Admin successfully logs in.  <b>Fails:</b> <b>Given:</b> The Admin wants to login. <b>When:</b> The Admin enters the wrong credentials. <b>Then:</b> The Admin is prompted with invalid credentials.
6.	Admin	Logout	Admin can securely leave the application by signing out from their administrator account.	<b>Accepted:</b> <b>Given:</b> The Admin is logged into their account. <b>When:</b> The Admin initiates the logout process by signing out. <b>Then:</b> The Admin successfully exits the application, and their session is securely ended.  <b>Fails:</b> <b>Given:</b> The Admin is logged into their account. <b>When:</b> The Admin attempts to sign out. <b>Then:</b> The Admin is prompted with a message indicating that the logout attempt was not successful.

## Client interface

	As a	I want to	So that	Test criteria
1.	Client	Make a booking	Client can reserve a training session	<b>Accepted:</b> <b>Given:</b> The Client is logged into their account and is on the booking application page.

				<p><b>When:</b> The Client selects the desired service or product, specifies the details, and confirms the booking.</p> <p><b>Then:</b> The system processes the booking and provides a confirmation message. The booking details will be visible in the Client's account.</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Client is logged into their account and is on the booking application page.</p> <p><b>When:</b> The Client attempts to make a booking, but there is a technical issue or an error occurs during the process.</p> <p><b>Then:</b> The system displays an error message, indicating that the booking could not be completed.</p>
2.	Client	Confirm Booking	Client can ensure their reservation is valid and confirmed.	<p><b>Accepted:</b></p> <p><b>Given:</b> The Client has made a booking and is logged into their account.</p> <p><b>When:</b> The Client navigates to the booking section of their account.</p> <p><b>Then:</b> The system displays the booking details, including the booking status (Confirmed)</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Client has made a booking and is logged into their account.</p> <p><b>When:</b> The Client accesses the booking section of their account, but there is a system error or the booking details do not load.</p> <p><b>Then:</b> The system displays an error message, indicating that the booking information was not retrieved.</p>
3.	Client	Update Login Credentials	Client can ensure the security of their account.	<p><b>Accepted:</b></p> <p><b>Given:</b> The Client is logged into their account.</p>

				<p><b>When:</b> The Client selects the option to update login credentials (e.g., change password or email address).</p> <p><b>Then:</b> The system allows the Client to make the desired changes and confirm that the login credentials have been updated successfully.</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Client is logged into their account.</p> <p><b>When:</b> The Client attempts to update login credentials, but there is a validation error (e.g., password doesn't meet requirements) or a technical issue.</p> <p><b>Then:</b> The system displays an error message, explaining why the update could not be completed.</p>
4.	Client	View Bookings	Client can check the details of their existing reservations.	<p><b>Accepted:</b></p> <p><b>Given:</b> The Client is logged into their account.</p> <p><b>When:</b> The Client accesses the booking section of their account.</p> <p><b>Then:</b> The system displays a list of the Client's bookings, including details such as date, time, service/product, and booking status.</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Client is logged into their account.</p> <p><b>When:</b> The Client attempts to access the booking section of their account, but there is a system error or the booking information is not retrieved.</p> <p><b>Then:</b> The system displays an error message, indicating that the booking details could not be displayed.</p>
5.	Client	Cancel booking	Client can change their plans or preferences.	<p><b>Accepted:</b></p> <p><b>Given:</b> The Client is logged into their account and has an existing booking.</p>

				<p><b>When:</b> The Client navigates to the booking details and selects the option to cancel the booking.</p> <p><b>Then:</b> The system processes the cancellation request and provides a confirmation message. The booking status updates to "Canceled."</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Client is logged into their account and has an existing booking.</p> <p><b>When:</b> The Client attempts to cancel the booking, but the system encounters an issue or there are restrictions on the cancellation.</p> <p><b>Then:</b> The system displays an error message, explaining why the cancellation could not be completed. This might include reasons like a cancellation deadline has passed or technical errors.</p>
--	--	--	--	--

## Admin interface

	As a	I want to	So that	Test criteria
1.	Admin	View Bookings	Admin can check clients reservations	<p><b>Accepted:</b></p> <p><b>Given:</b> The Admin is logged into their account.</p> <p><b>When:</b> The Admin accesses the bookings section of the clients accounts.</p> <p><b>Then:</b> The system displays a list of the Client's bookings, including details such as date and time, as well as the type of service they requested.</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Admin is logged into their account.</p> <p><b>When:</b> The Admin attempts to access the booking section of the client account, but</p>

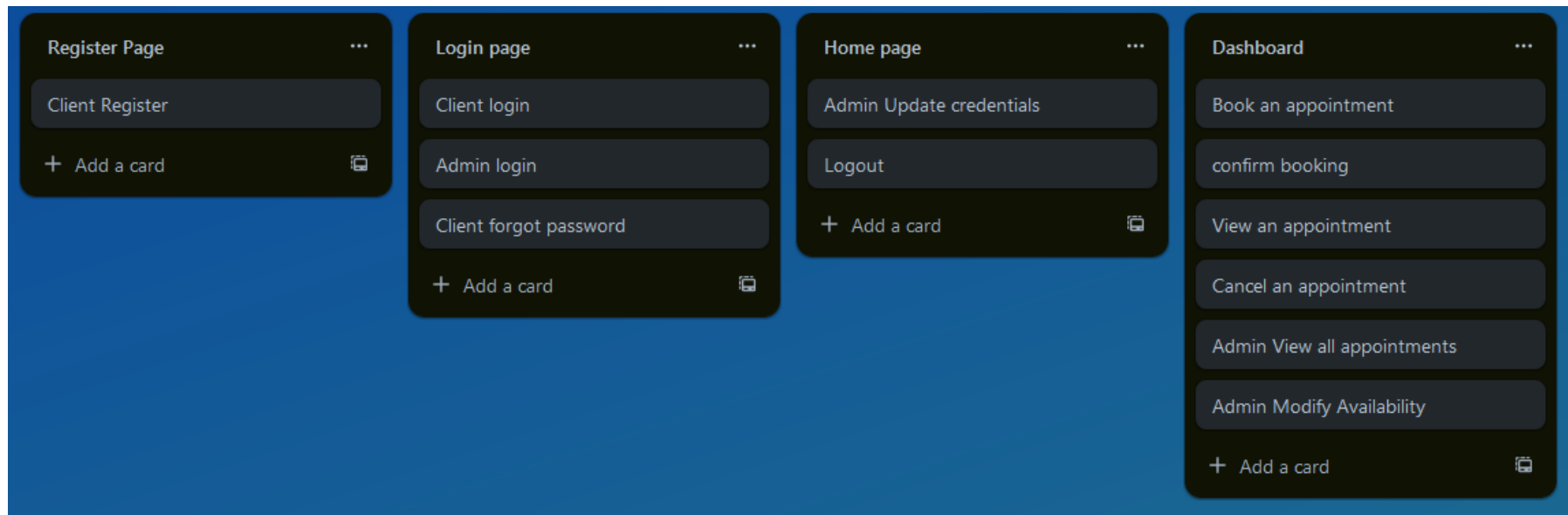
				<p>there is a system error or the booking information is not retrieved.</p> <p><b>Then:</b> The system displays an error message, indicating that the booking/reservation details could not be displayed.</p>
2.	Admin	Modify Availability	Admin can update their schedules for the clients to book reservations	<p><b>Accepted:</b></p> <p><b>Given:</b> The Admin is logged into their account.</p> <p><b>When:</b> The Admin accesses the bookings section of the clients accounts.</p> <p><b>Then:</b> The system displays a list of the Client's bookings, including details such as date and time, as well as the type of service they requested.</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Admin is logged into their account.</p> <p><b>When:</b> The Admin attempts to access the booking section of the client account, but there is a system error or the booking information is not retrieved.</p> <p><b>Then:</b> The system displays an error message, indicating that the booking/reservation details could not be displayed.</p>
3.	Admin	Cancel Bookings	Admin can cancel any client bookings	<p><b>Accepted:</b></p> <p><b>Given:</b> The Admin is logged into their account and the client has an existing booking.</p> <p><b>When:</b> The Admin navigates to the Client booking details and selects the option to cancel the booking.</p> <p><b>Then:</b> The system processes the cancellation request and provides a confirmation message. The booking status updates to "Canceled."</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Admin is logged into their account and the Client has an existing booking.</p>



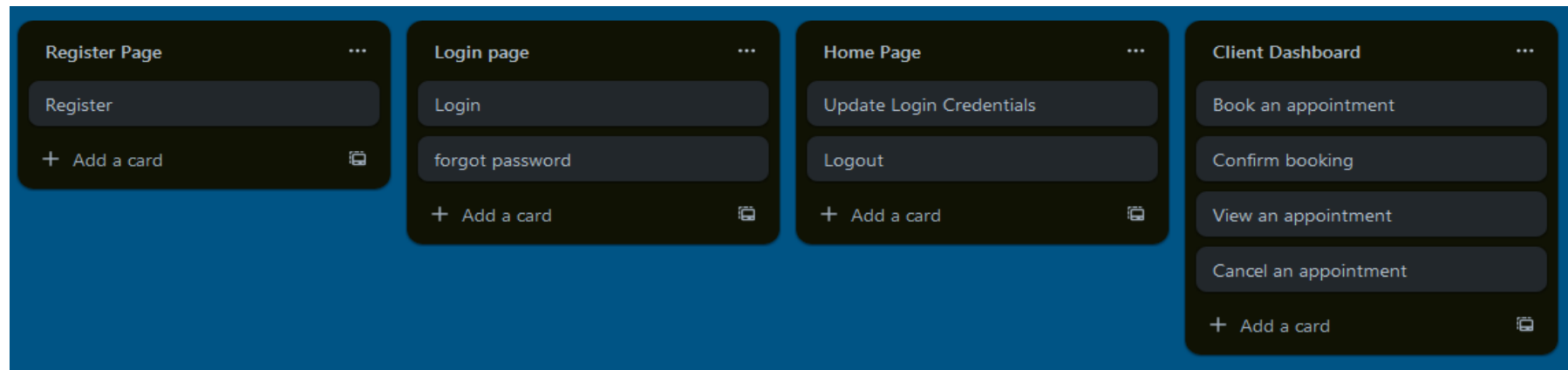
				<p><b>When:</b> The Admin attempts to cancel the Client booking, but the system encounters an issue or there are restrictions on the cancellation.</p> <p><b>Then:</b> The system displays an error message, explaining why the cancellation could not be completed. This might include reasons like a cancellation deadline has passed or technical errors.</p>
5.	Admin	View Client Info	Admin can view any clients information	<p><b>Accepted:</b></p> <p><b>Given:</b> The Admin is logged into their account.</p> <p><b>When:</b> The Admin selects a Client from a list and clicks on their account.</p> <p><b>Then:</b> The system allows the Admin to view all the client information, such as passed bookings, client names, emails and numbers....</p> <p><b>Fails:</b></p> <p><b>Given:</b> The Admin Is logged into their account.</p> <p><b>When:</b> The Admin attempts to view a Client account, but there is an error or a technical issue that prevents the Admin from viewing a client.</p> <p><b>Then:</b> The system displays an error message, explaining why the client's information could not be retrieved.</p>

## Appendix 3 - User Story Map

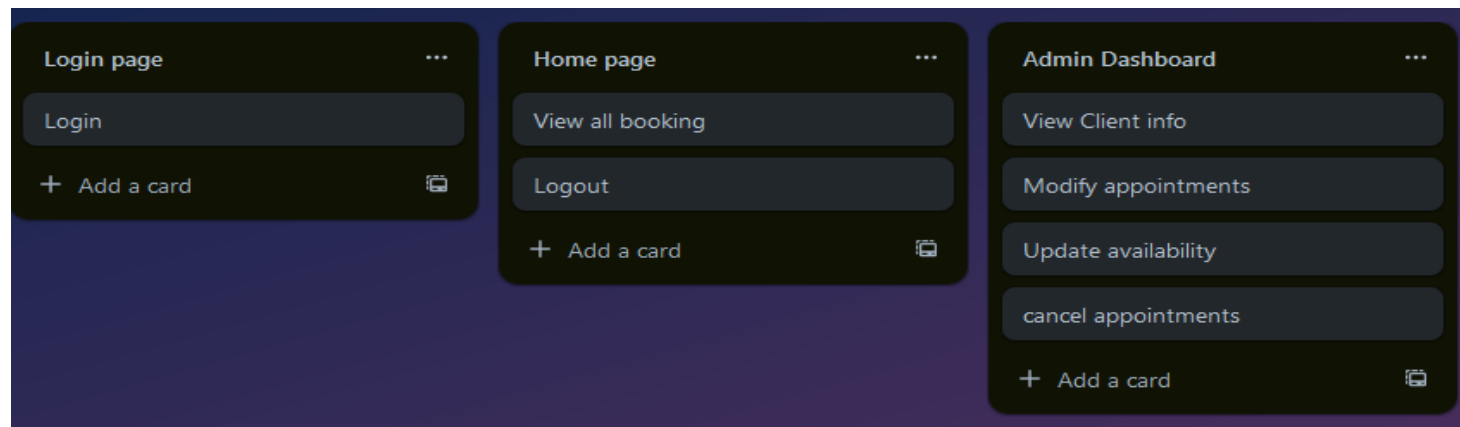
---



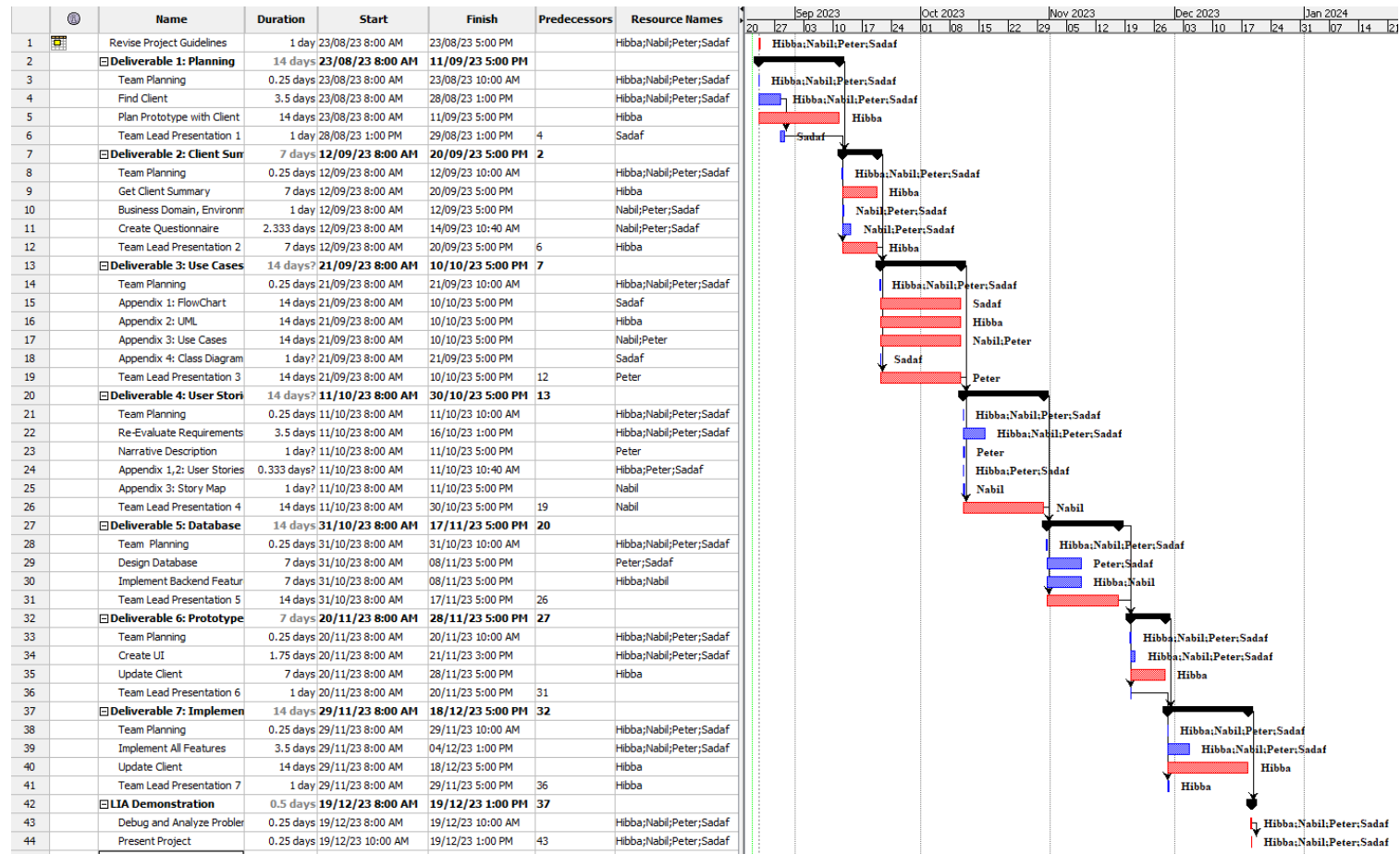
## Client interface



## Admin interface



# Project Plan



[Link to WBS and Gantt Chart & PDF Version](#)