

Group Project Log

Group Name:	Group -10
--------------------	-----------

Group Members:	Alagu Swrnam Karrupiah, Aravind Jayanthi, Navya Jayapal, Prachi Kabtiyal, Samarth Jariwala, Soham Kansodaria
-----------------------	--

Deliverable:	Sportify Project Report
---------------------	-------------------------

Group Member Name	Work Done (%)
Alagu Swrnam Karrupiah	16.6%
Aravind Jayanthi	16.6%
Navya Jayapal	16.6%
Prachi Kabtiyal	16.6%
Samarth Jariwala	16.6%
Soham Kansodaria	16.6%
Total:	100%

TECHNICAL REPORT

GROUP 10

SPORTIFY: A SPORTS CLUB MANAGEMENT SYSTEM

Members and Contributors

Alagu Swrnam Karruppiyah
B00911810
al581093@dal.ca

Aravind Jayanthi
B00868943
ar687531@dal.ca

Navya Jayapal
B00886554
nv408879@dal.ca

Prachi Kabtiyal
B00862464
pr522601@dal.ca

Samarth Jariwala
B00899380
sm228153@dal.ca

Soham Kansodaria
B00865680
sh788512@dal.ca

Faculty of Computer Science
Dalhousie University

21 July 2022

ABSTRACT

Sportify is a one-stop web application offering numerous services a typical sports club provides. Our web application offers club members to purchase their membership online and book facilities (gym equipment, pool, badminton court) and events hosted at the club. Members can also blog about their fitness life in our application, empowering them to build a strong community. Admins can manage facilities, events, and products through the same intuitive, user-friendly web application. Thus, providing a unified solution for admins to drive a sports club seamlessly and members to schedule their workout and recreational session without coming to the club.

KEYWORDS

Admin, Blogs, Events, Facilities, Bookings, Members, Membership, Merchandise, Sports club, User-friendly, Web-application.

CONTENTS

1.	INTRODUCTION	1
1.1	Live Project URL.....	1
2.	BACKGROUND.....	1
2.1	Competitive Landscape	1
2.2	Problem and Approach	2
3.	APPLICATION DETAILS	2
3.1	Target User Insights.....	3
3.2	User-Centered Design Approach	3
3.2.1	Information Architecture	5
3.2.2	Design and Layout.....	6
4.	APPLICATION WORKFLOW	37
4.1	Interaction Design.....	39
4.1.1	User Management	39
4.1.2	Membership Management.....	48
4.1.3	Facility Management.....	53
4.1.4	Event Management.....	57
4.1.5	Payment Management.....	61
4.1.6	Merchandise Product Management.....	63
4.1.7	Search Function.....	68
4.1.8	Blogging Feature.....	71
4.1.9	Rewards System.....	75
4.2	Process and Service Workflow	77
5.	SECURITY	78
6.	CONCLUSION	78
7.	RECOMMENDATIONS	79
8.	REFERENCES	79

1. INTRODUCTION

Sports clubs are required to manage numerous administrative and management tasks and issues that become a big problem for long-term sustainability. Sportify is a sports club management system that provides a single-stop online portal to address the issues of managing a sports club, including managing existing customers and their memberships with a secure payment gateway and smoothening the customer onboarding process. In addition to providing a solution for club administrators, the system is designed to be a platform for club members to access services online provided by the club. We have also empowered the club members to write blogs related to fitness to build a reliable community. Overall, Sportify is a single web application for both sports club administrators/maintainers and members to utilize the online services of the club and streamline their work (for administrators) and fitness (for members) schedules.

1.1 Live Project URL

The URL of our developed application is provided below:

<https://sportify-prd.herokuapp.com/>

We have decided to create two Git Lab repositories for this project to maintain frontend and backend code separately. The details for the repositories are provided below:

Frontend: [5709-Group10](#)

Backend: [5709-g10-sportify-backend](#)

2. BACKGROUND

2.1 Competitive Landscape

Sportify offers a wide range of services to its user base, from membership to facility and event bookings. It has few competitors in the market, including the web portals of Good Life Sports [12], YMCA [13], and Dalplex [14]. It is crucial for us to distinguish the competitors from potential clients. For instance, the Dalplex system, which manages the services provided by the Dalplex club, is a competitor, but the Dalplex physical club itself is a potential client. Our competitors provide services for the users to book recreational facilities like swimming pool and badminton courts. Below is the list of features that distinguishes Sportify from its competitors:

1. Sportify provides both event management and equipment management for booking events and equipment's.
2. Sportify offers rewards which club members can utilize to reduce their membership price.
3. Customers can view membership plans and purchase or cancel memberships online.

4. Sportify has an added flavor of marketing by providing an advertisement platform for merchandise products.
5. Blogging feature in Sportify is a fresh concept designed to build a community for fitness enthusiasts.

2.2 Problem and Approach

Problem:

Customers need a unified online portal to access a sports club's facilities without going to the club physically. For example, a busy lawyer doesn't want to wait for the availability of the treadmill for his workout. The lawyer can book the treadmill equipment beforehand, ensuring the workout session will happen without any wait time.

A club administrator should have a system to manage club facilities like gym equipment, events, and products. For instance, if the club plans a summer yoga session next Monday, the administrator can add a new event on the web application for users to book tickets instead of announcing it on the notice board.

Approach:

The main project's goal is to improve user access to the sports club's resources and services and improve the efficiency of managing multiple facilities and events hosted by the club. Most of the target user-base of the application is the young population with a busy schedule from various backgrounds. Since the user base is from multiple backgrounds, we have decided to keep the website more accessible to navigate and user-friendly. We have also designed and developed the website UI responsive, which means the UI adapts to the screen size and provides an optimized and intuitive experience irrespective of the device type and size. We have also added up-to-date security features with password encryption on the client side. Club members can use the website to purchase/cancel their memberships, book facilities and events and browse through various merchandise products the club offers. Club administrators can easily manage facilities, events, and products that the club provides.

3. APPLICATION DETAILS

Designed and developed a website that allows sports clubs to open their own website and provide all information related to the club, offers and discounts, with the ability to book without the need to personally go to the club to register, in order to motivate them to participate in the appropriate events for them in an easy way and show the best equipment and machines in addition to innovative programs, cardio and strength halls, freestyle and varied classes. The MERN stack was used to create the proposed system. The site offers the customers and administrators an effective way to oversee the different activities of the sports club. Moreover, Sportify collects clients' personal information like name, date of birth, contact details, and payment details. It embeds and follows privacy protocols to safeguard sensitive information to ensure reliability and gain the public's trust. Sportify aims to provide a user-friendly and responsive interaction so that the

customers continue to use it frequently and repeatedly. We have designed a fully functional Event Manager thus the club can easily handle event users without any hassle and can make the event more comfortable. There is also auto ticket generate system that help the event organizer to make their event more secure and ensure the all participant.

3.1 Target User Insights

Sportify is designed to address the needs of two users: **club members** and **administrators**. Members and clients of the sports club will be able to use the club's facilities via our website. The club's personnel who oversee the club's facilities and operations are known as club administrators.

Club member: A typical club member is the one that is attempting to get in shape while juggling his busy schedule and other obligations. Members can use this website to manage their memberships, access numerous services, register for club activities, continue to blog about their experiences, and browse products. Overall, the website is user-friendly and intuitive to help users manage their time and work out efficiently.

Administrator: An administrator manages the club infrastructure and activities. The club administrators should be able to navigate and browse the internet more easily. They have complete control over users, events, and membership packages.

The website is set up so that any user can use it without difficulty. The design is also responsive, so the layout adapts based on the user's device and can be used on a desktop, tablet, or mobile phone. The uniformity of the pages cuts down on time it takes to explore the site. Users do not need to have any prior knowledge to browse the website.

3.2 User-Centered Design Approach

Since the target users of our website can have varied browsing experience. We designed the website as simple as possible by showing only the necessary content on the page. After through research, we decided to use the color palette shown in the **Figure 1** which make the website visually appealing and uniform across different pages and features.

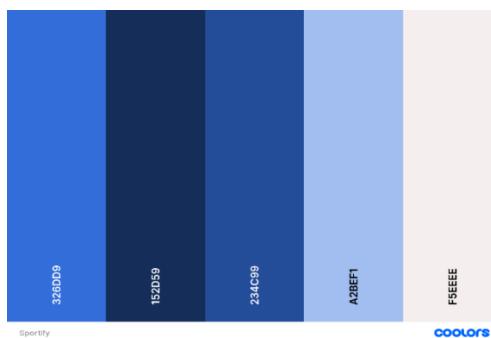


Figure 1: Color Palette used for the website.

We have followed the 7+2 rule for the app bar and 3-click rule to make the website features easily accessible and simpler to navigate for different types of users. Maintaining consistency on the website is critical to ensuring an optimal user experience. The amount

of time it takes for users to learn how to use a product is influenced by consistency. Our website is consistent in terms of forms, fonts, and layout. When designing the website, we aimed to provide the most readable dialogue possible to the user. This includes defining terminology, avoiding jargon, and displaying only information pertinent to the task at hand. Users expect a response to all their actions. This could include changing the appearance of the screen after completing an action. The website notifies the user of the outcome of each action. Some examples include reservations, events, membership purchases, and membership cancellations.

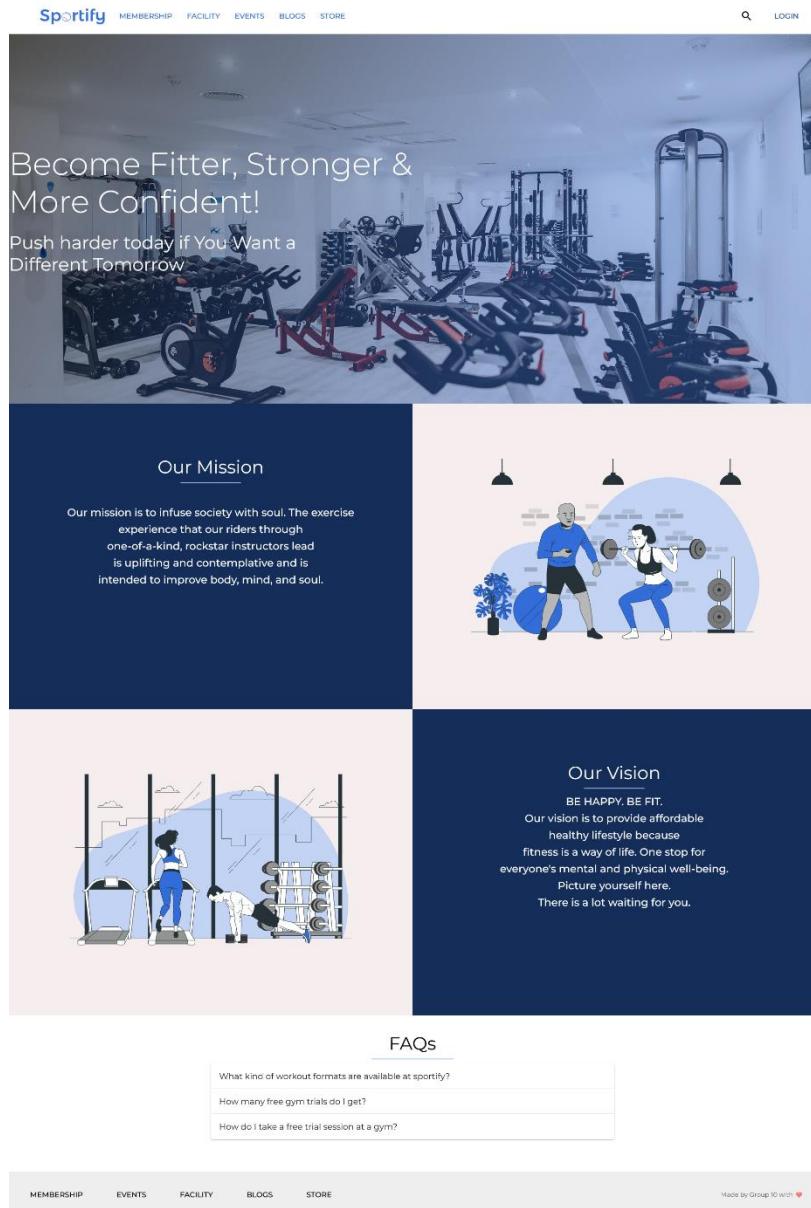


Figure 2: Sportify Home page.

3.2.1 Information Architecture

Proposed Sitemap

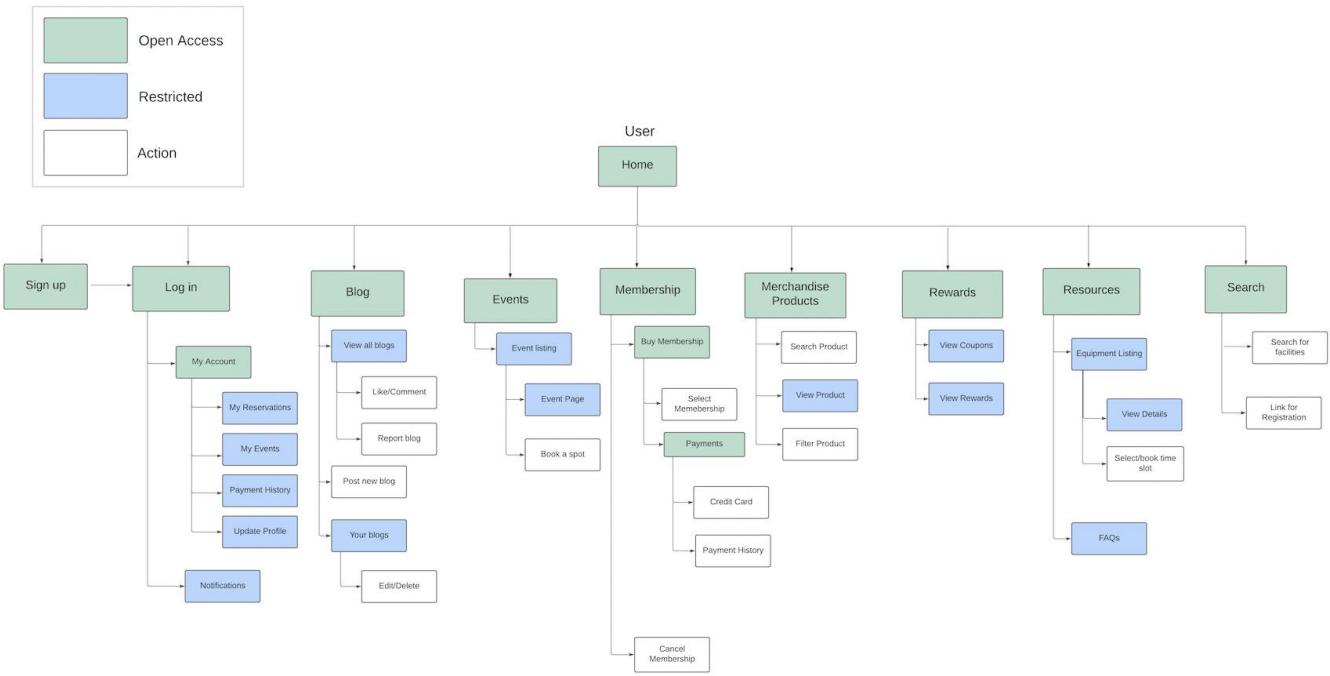


Figure 3: Sitemap for Sportify - User, created with Lucid chart [1].

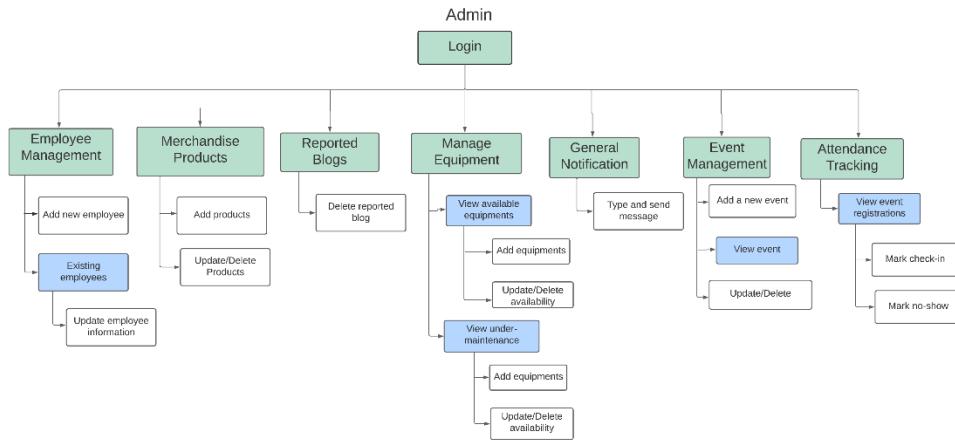


Figure 4: Sitemap for Sportify - Admin, created with Lucid chart [1].

3.2.2 Design and Layout

One of the key goals while analyzing the design and layout of the website is to keep it simple and easier to navigate. To achieve that goal, uniformity across website is needed which makes the website more cohesive and accessible to large user base with little to no learning curve.

To make the navigation simple, all the main features of the website are accessible through app bar as shown in **Figure 5**. We have hidden all the user specific details under profile icon to reduce clutter and at the same time making website accessible.



Figure 5: App bar of the website.

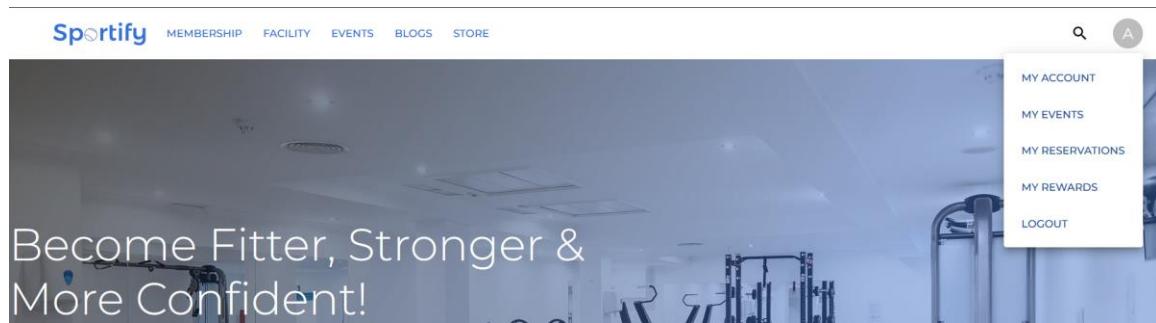


Figure 6: App bar with profile icon options.

Except during login, signup and forget password pages, app bar always stays on top of the page for user to easily navigate across website.

Admin can add new facilities, events, and products for users to browse through and utilize these services to their full extent. To make the admin's job easy we have designed the forms to add facilities, events, and products simple and straightforward with clear and appropriate prompts. As shown in **Figure 7**, **Figure 8**, and **Figure 9** the forms for adding the above mentioned entities are designed to make the task uniform across the site.

Add New Facility Details

Facility Name

Facility Location

Description

Category
Gym



BACK POST

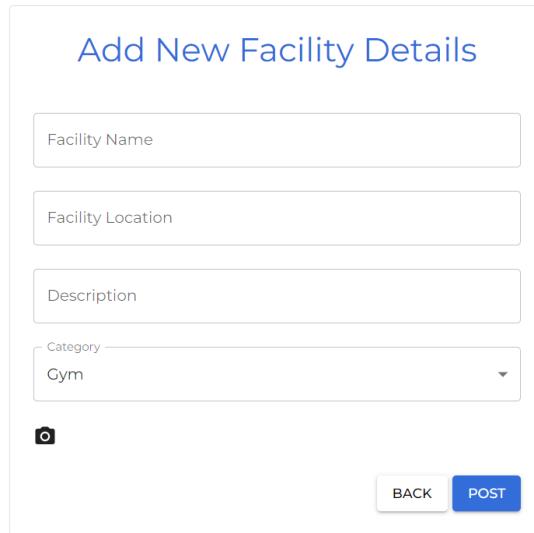


Figure 7: Add new facility form in admin panel.

Add New Event Details

Event Name

Event Location

Description

Choose Date 

Available Slots*
1



BACK POST

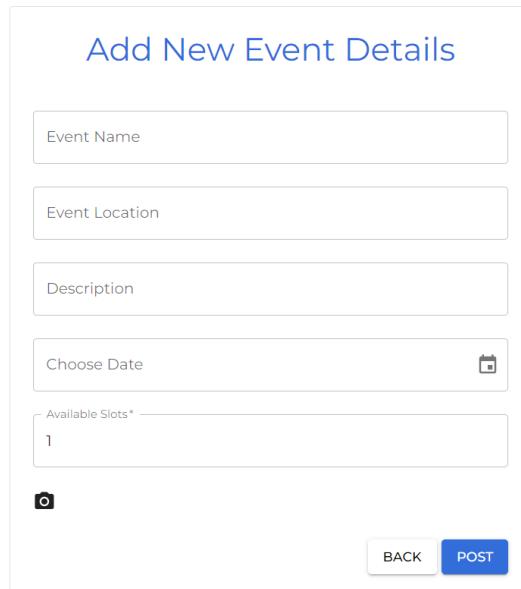


Figure 8: Add new event form like add new facility.

Add New Product Details

Product Name

Product Price

Description



BACK POST

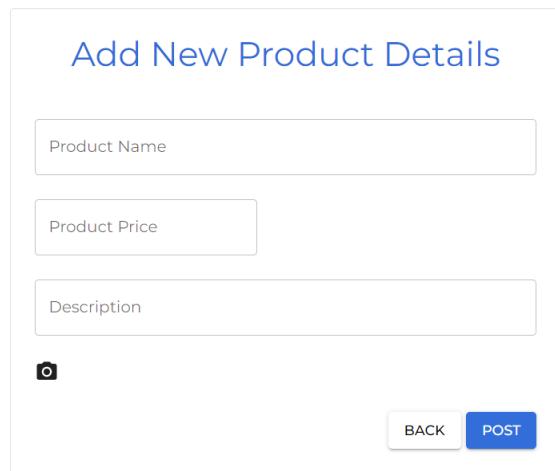
A wireframe of a mobile application screen titled "Add New Product Details". It contains three input fields: "Product Name", "Product Price", and "Description". Below the fields is a camera icon. At the bottom are "BACK" and "POST" buttons.

Figure 9: Form to add new products.

User Management:

It provides users ease of use to register and login in application. Also, it provides user information and user can also change it.

A Web Page
<https://sportify.com/login>

Brand

Signup

First Name Last Name

Email

Contact No

Password

Confirm Password

Submit

[Already a member? Signin](#)

[Back to home](#)

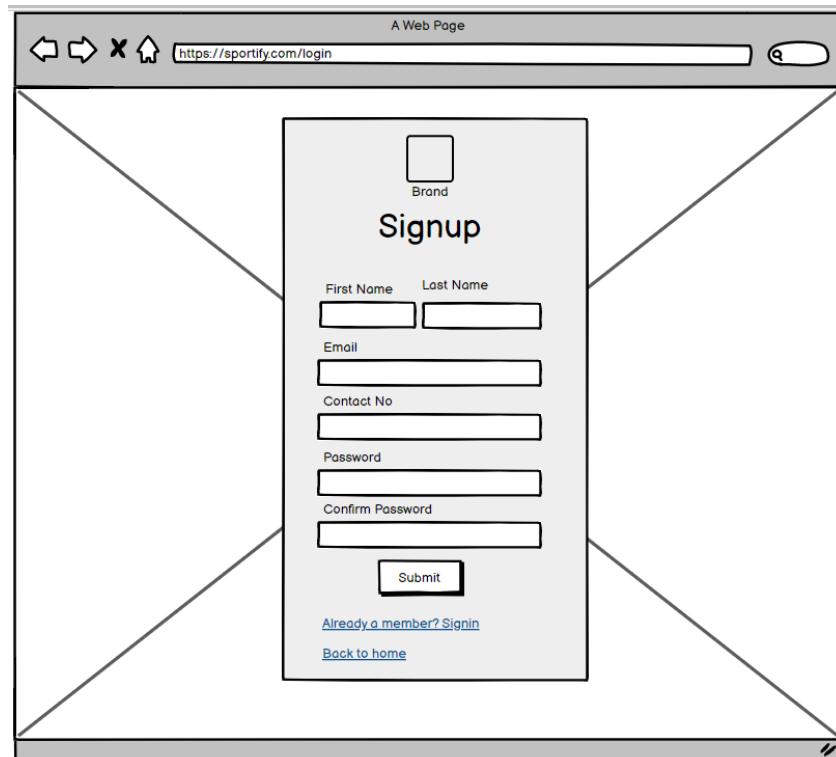
A wireframe of a web browser window showing a "Signup" form. The form includes fields for First Name, Last Name, Email, Contact No, Password, and Confirm Password, followed by a Submit button. There are links for "Already a member? Signin" and "Back to home". The browser header shows the URL "https://sportify.com/login".

Figure 10: Wireframe of "Signup" page using Balsamiq [3]

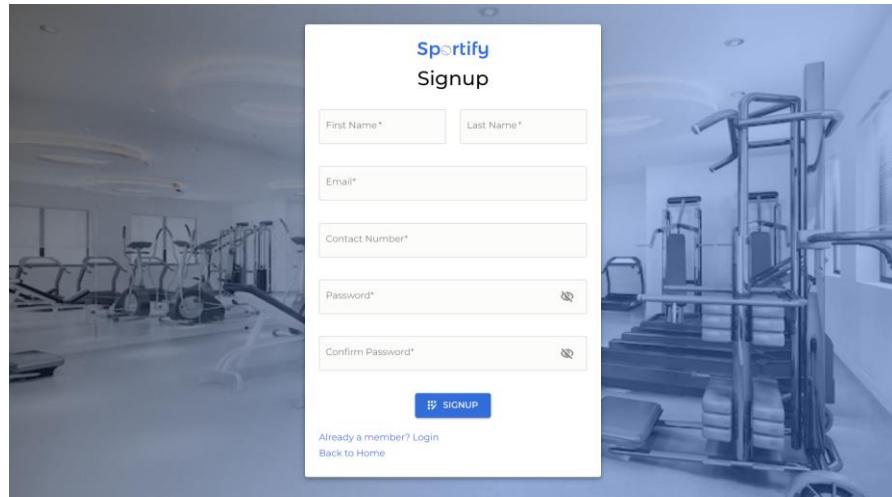


Figure 11: Design of “Signup” page

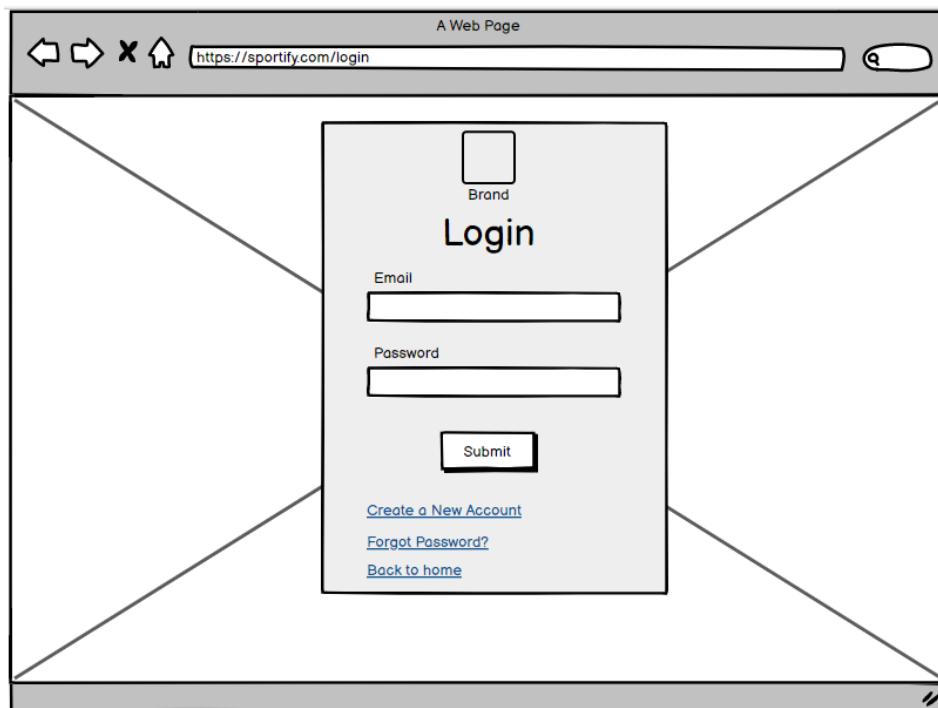


Figure 12: Wireframe of "Login" page using Balsamiq [3]

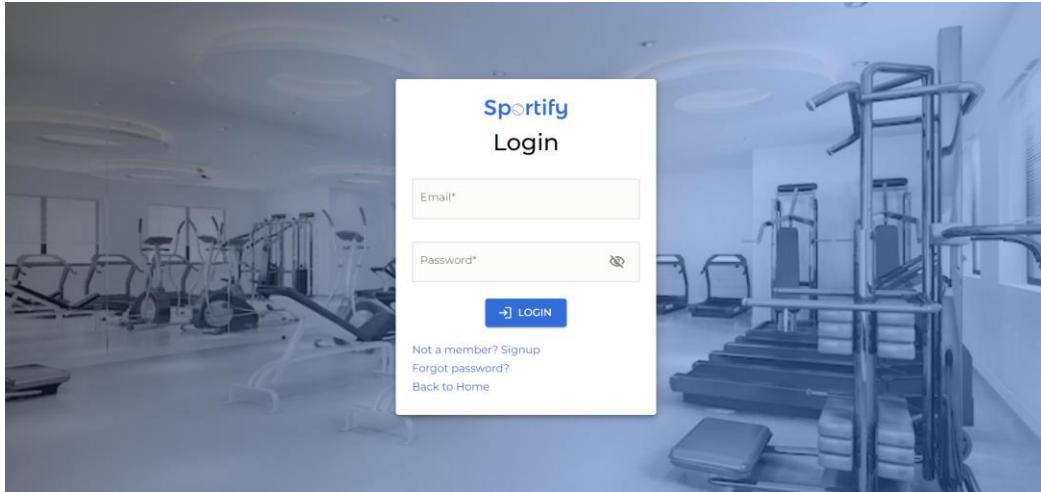


Figure 13: Design of "Login" page

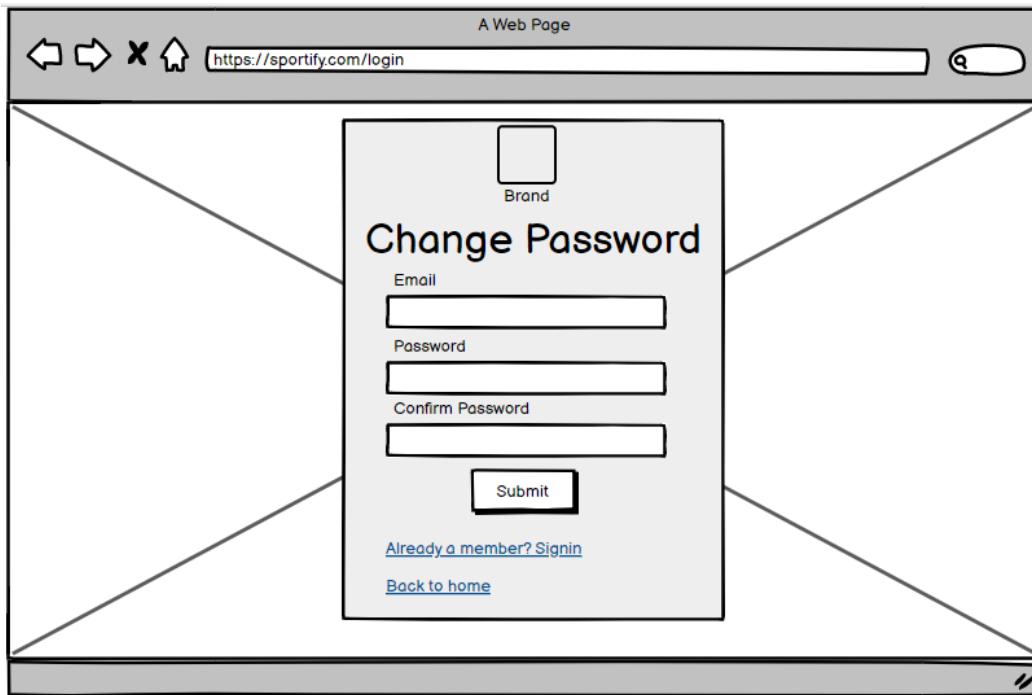


Figure 14: Wireframe of "Change Password" page using Balsamiq [3]

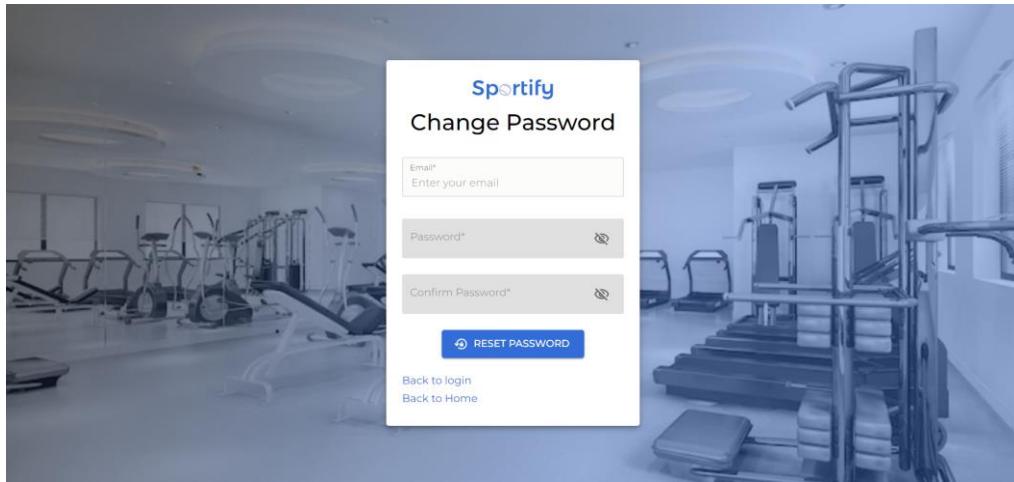


Figure 15: Design of "Change Password" page

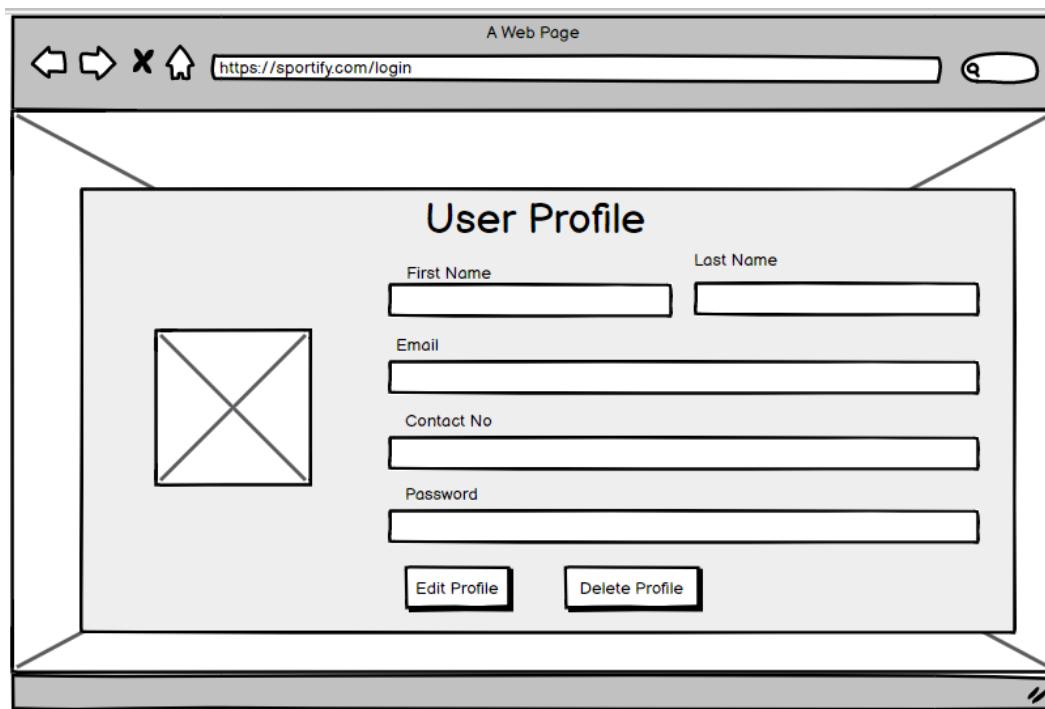


Figure 16: Wireframe of "User Profile " page using Balsamiq [3]

The screenshot shows the 'User Profile' section of the Sportify website. At the top, there is a placeholder for a user profile picture. Below it, there are four input fields: 'First Name' (John), 'Last Name' (Doe), 'Email' (john@gmail.com), and 'Address' (1333-South Park Street, Halifax, NS, Canada). There are also two buttons at the bottom: 'EDIT PROFILE' (blue) and 'DELETE PROFILE' (red).

Figure 17: Design of "User Profile" Page

The wireframe illustrates the 'User Update' interface. It features a central modal window titled 'User Update'. Inside, there are five input fields: 'First Name' and 'Last Name' (both with placeholder boxes), 'Email' (with a placeholder box), 'Contact No' (with a placeholder box), and 'Password' (with a placeholder box). At the bottom of the modal are two buttons: '< Update' on the left and '< Cancel' on the right. The entire modal is set against a background of a web browser window showing the URL 'https://sportify.com/login'.

Figure 18: Wireframe of "User Update" page using Balsamiq [3]

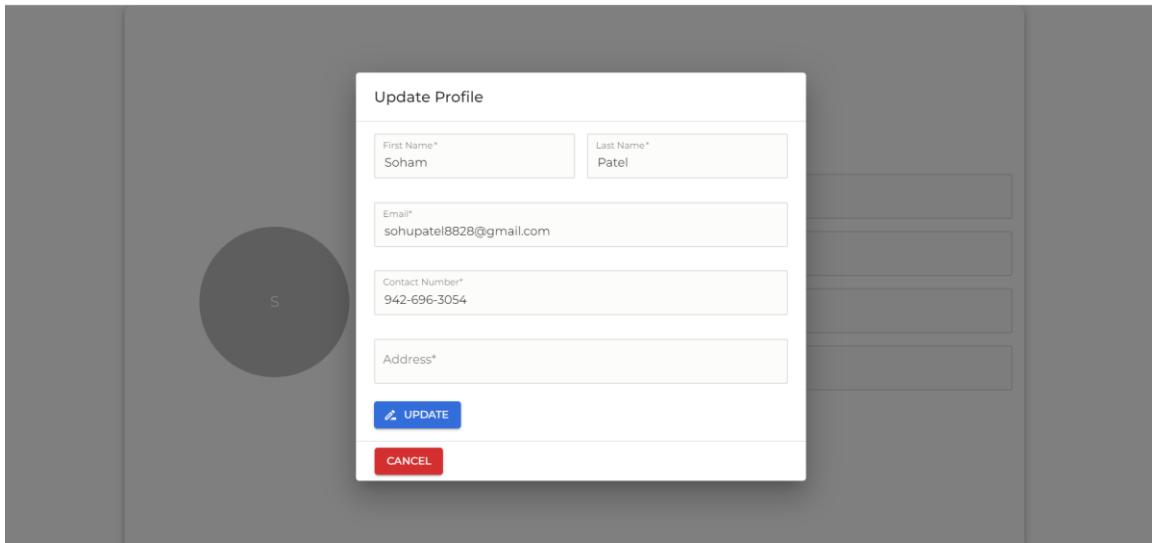


Figure 19: Design of "User Update" page

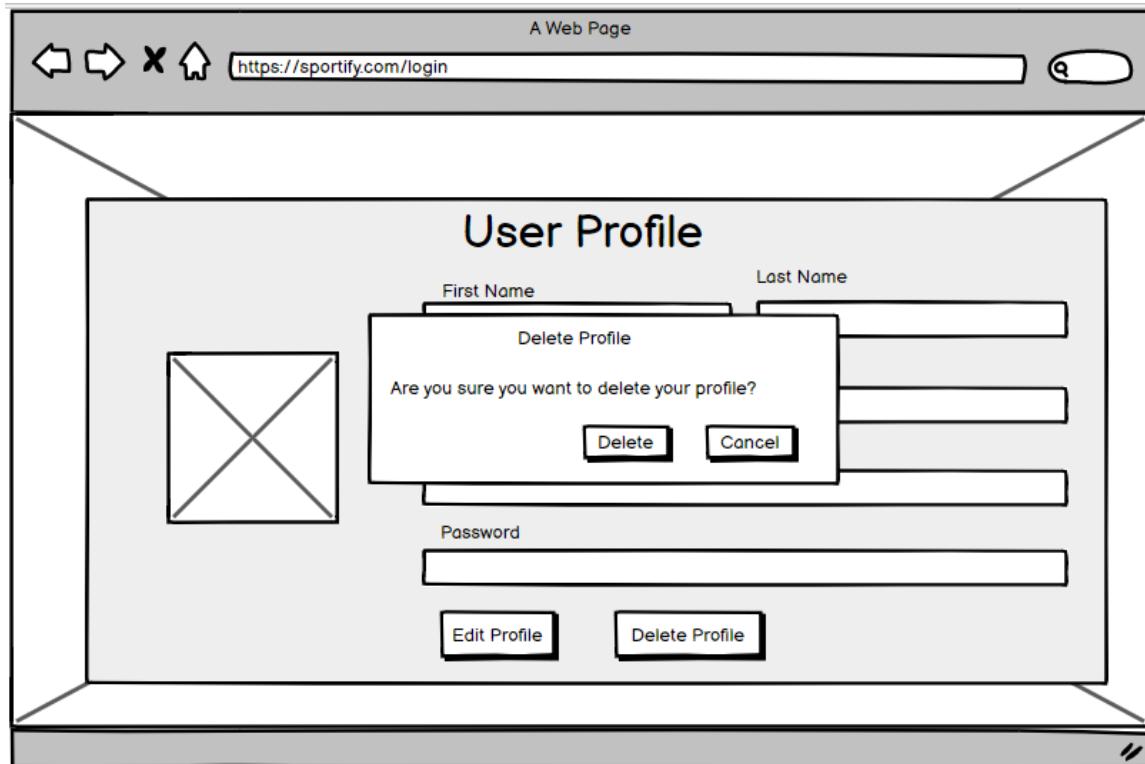


Figure 20: Wireframe of "Delete Profile" Page using Balsamiq [3]

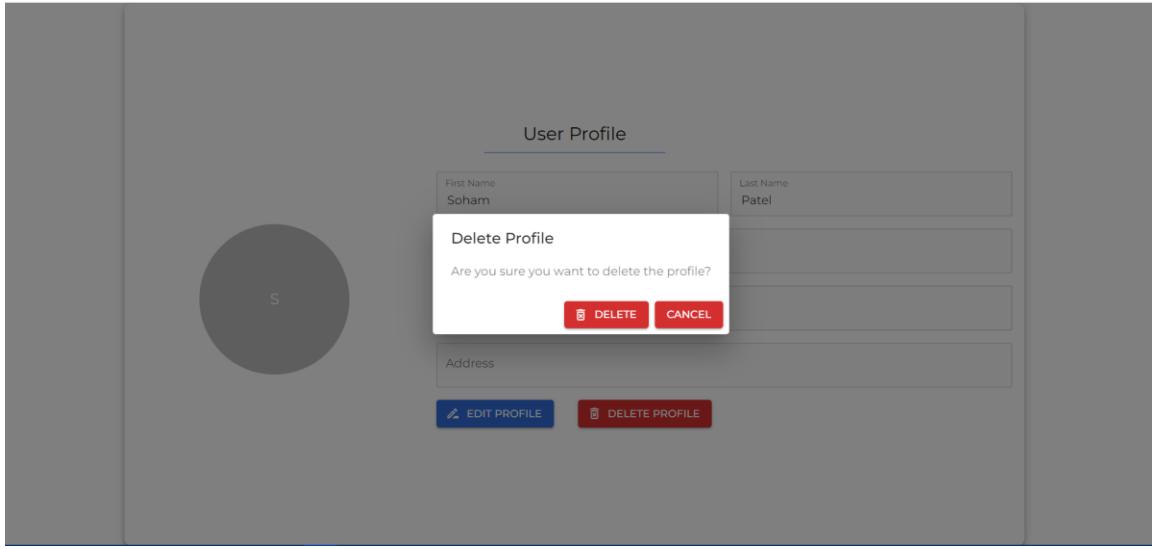


Figure 21: Design of "Delete Profile" page

Membership Management:

The Membership components begins with the cards showing the major categories of memberships offered by the sports club. The cards layout is minimalistic and easy to read so that users can make an informed decision based on the differences between the facilities offered in different memberships. After selecting a membership, user lands to a pre-filled form page (in case there is existing information). The user can update or add the billing details in the respective placeholder. The form has a simple and short design keeping in mind the user experience. Next, the user lands to review page that provides a summary of the purchase. After reviewing, user can proceed to payment. After successful payment, system redirects the user to a page showing the purchased membership. This page also provides an interface to cancel membership.

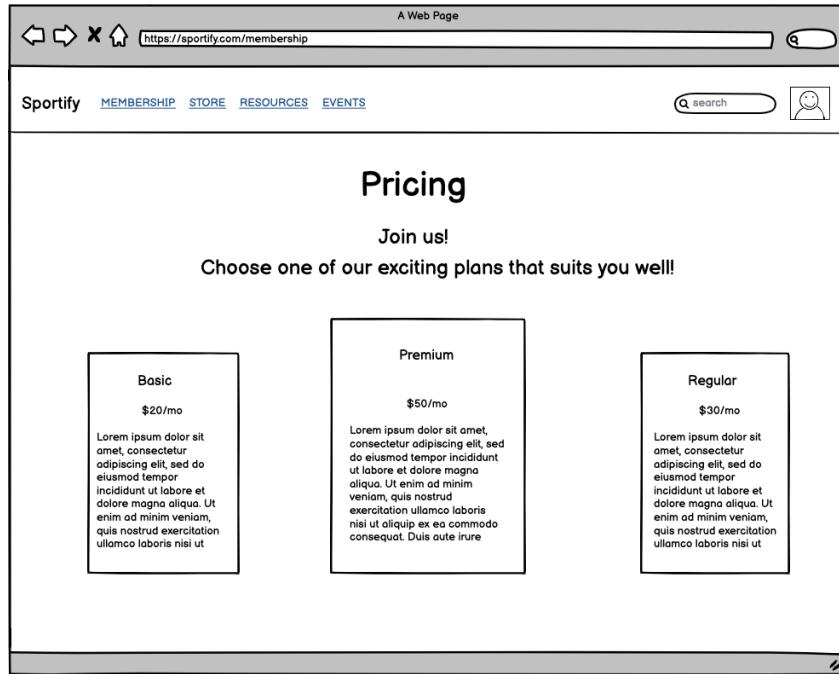


Figure 22: Wireframe of "View membership plans" using Balsamiq [3]

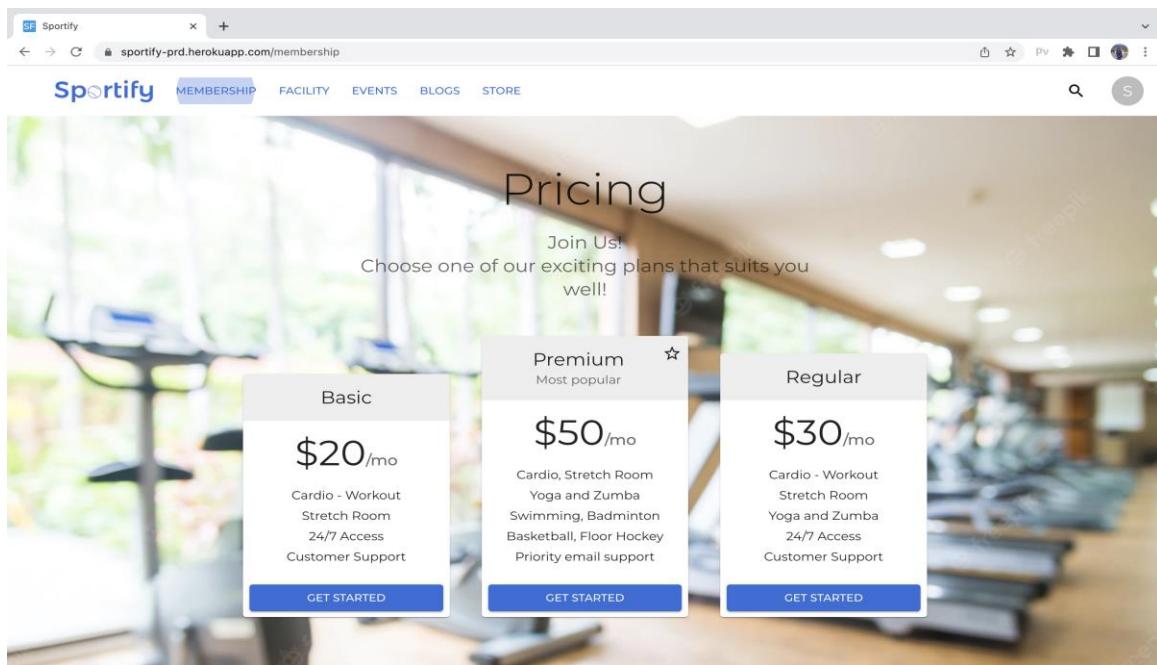


Figure 23: Design of "View membership plans"

A Web Page

<https://sportify.com/membership/checkout>

Libriya Study Room Booking Poster Request Fines Register/Login

Checkout

Billing Information Review Your Order

Billing Information

First Name *	Last Name *
Address Line 1 *	
City *	State/Province/Region
Zip/Postal Code *	Country *

Back **Proceed to review**

Figure 24: Wireframe of "Membership billing information form" using Balsamiq [3]

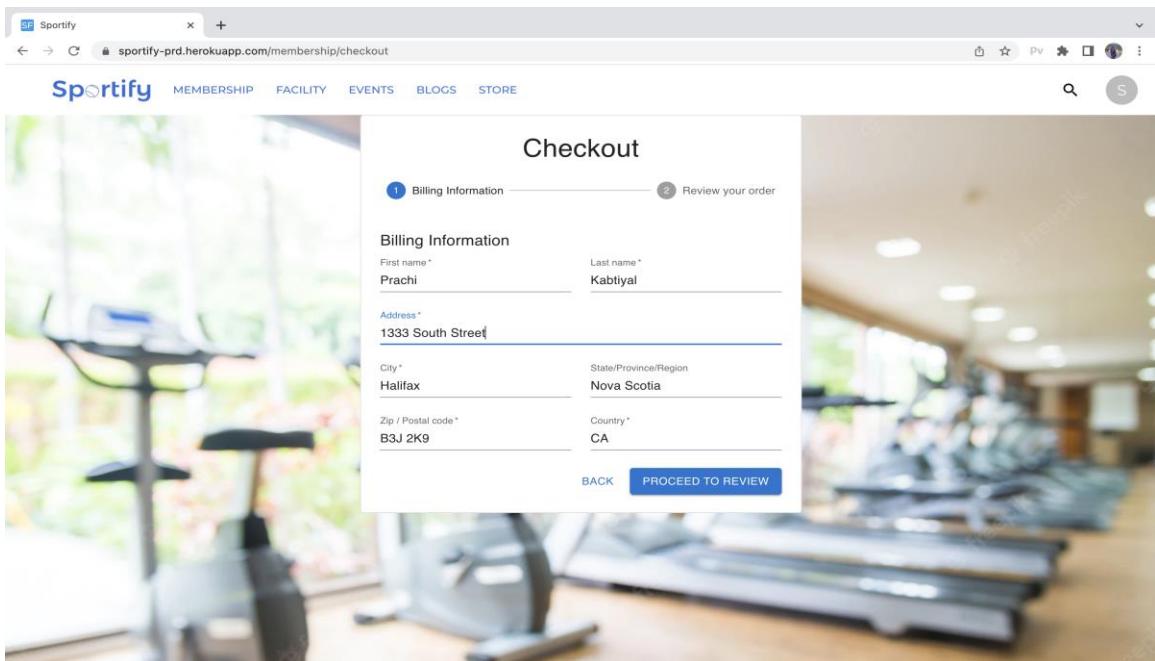


Figure 25: Design of "Membership billing information form"

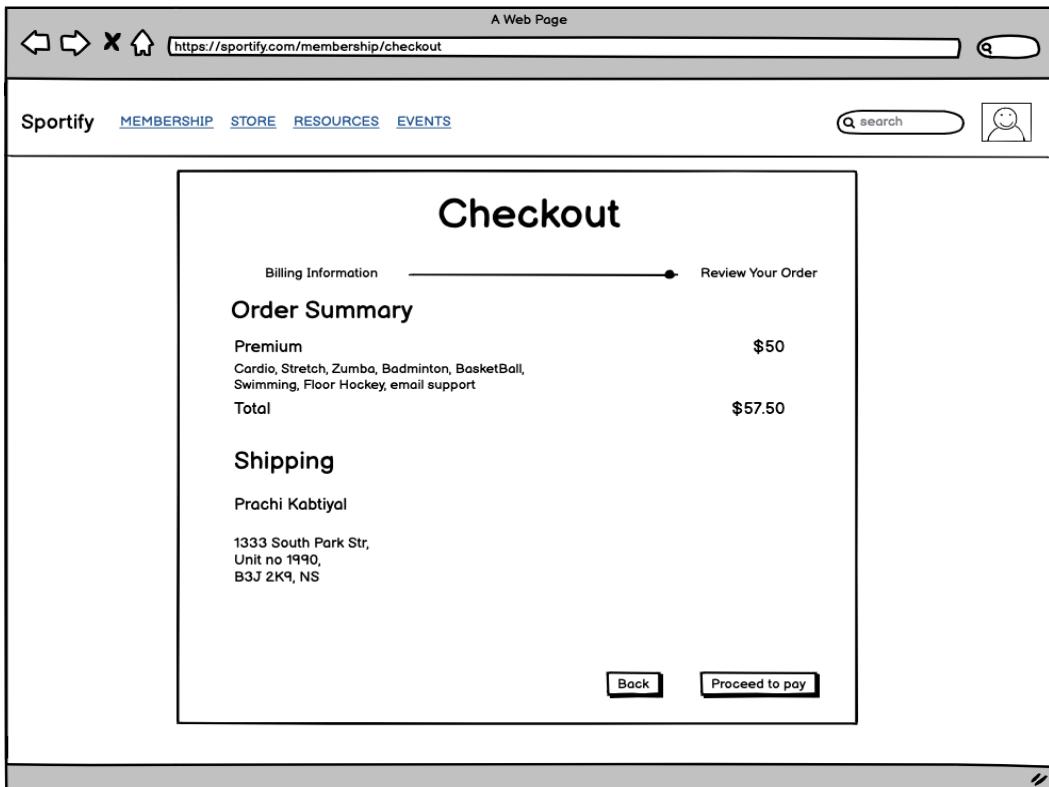


Figure 26: Wireframe of "Membership purchase review" using Balsamiq [3]

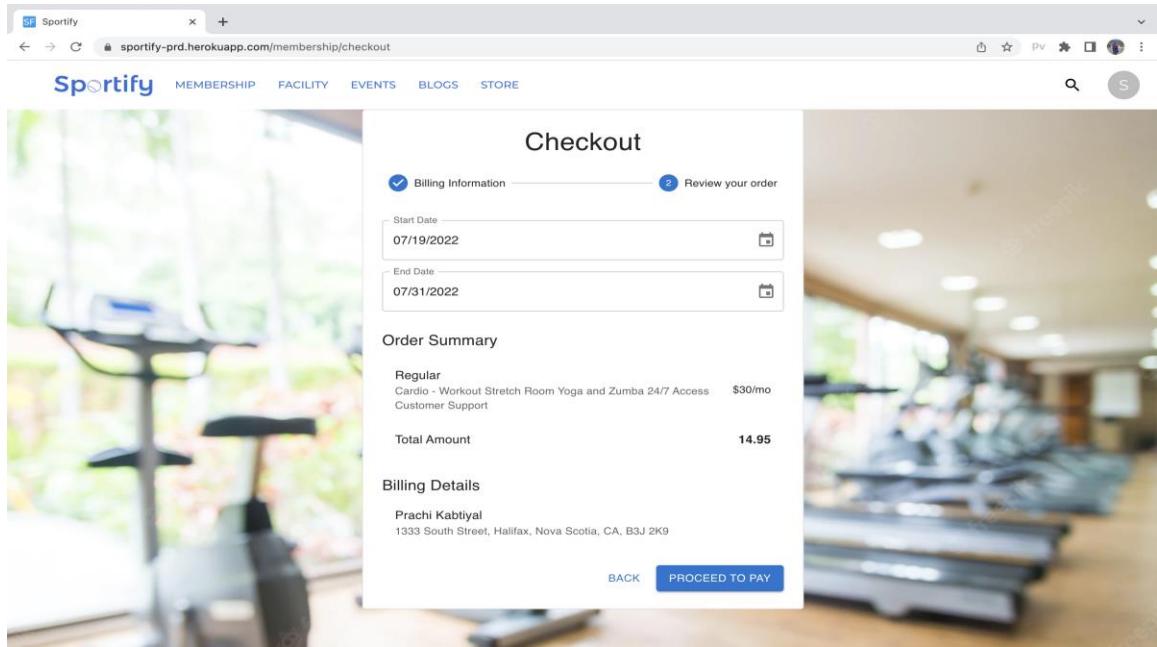


Figure 27: Design of "Membership purchase review"

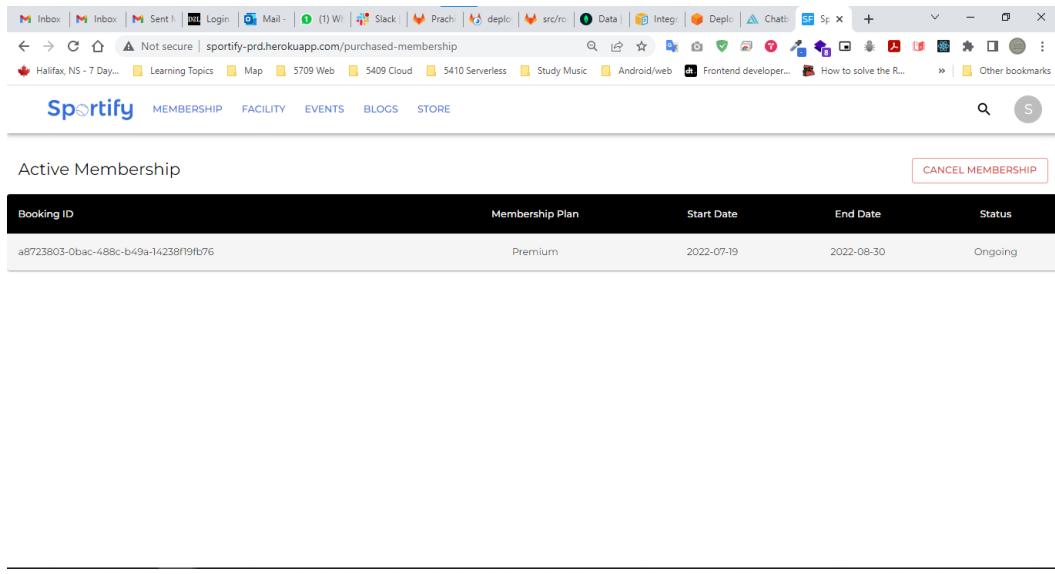


Figure 28: Design of "View active membership with cancellation option"

Facility and event Management:

The purpose of the facility reservation is to block a club's facility like swimming pool or a badminton court ahead of time so that users can plan their workout accordingly. Similarly, event booking also helps user to book tickets for an upcoming event based on the available capacity. In event booking, user can book more than one ticket. This makes the user plan the time ahead and the cancellation feature can help user to cancel any booking to a facility or an event if any unforeseen commitment occurs.

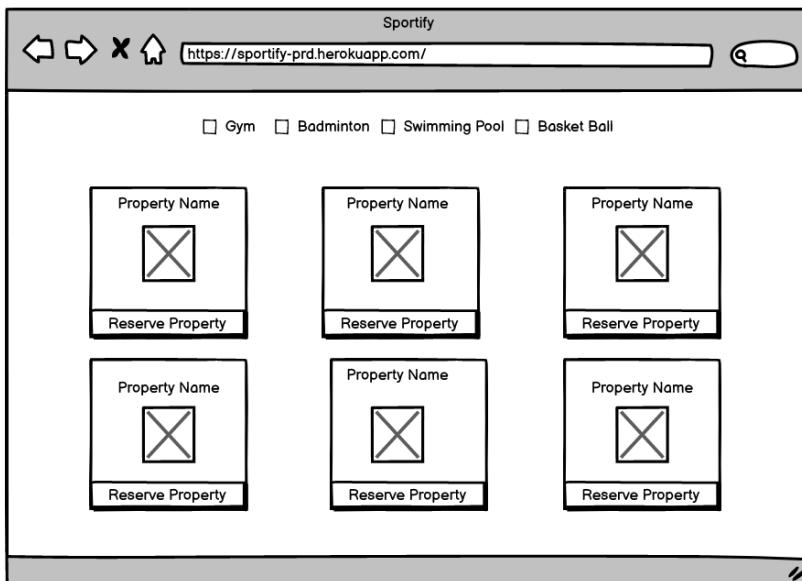


Figure 29: Wireframe for facility listing page using Balsamiq [3].

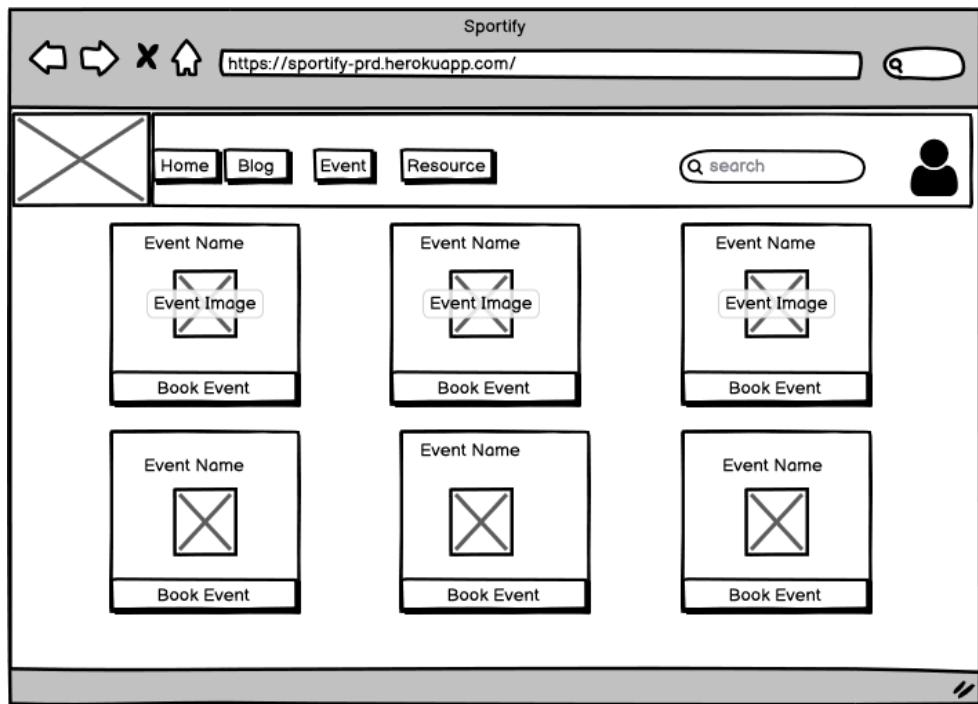


Figure 30: Wireframe for event listing page using Balsamiq [3] having similarity with facility listing page.

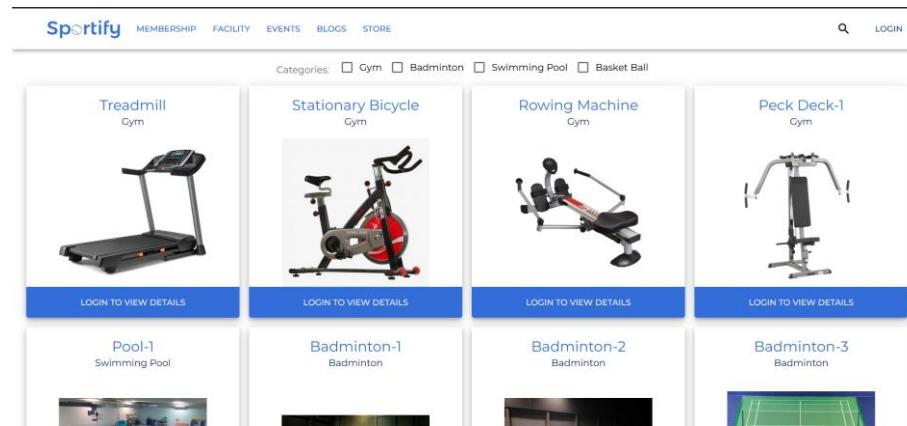


Figure 31: Facilities listing page.

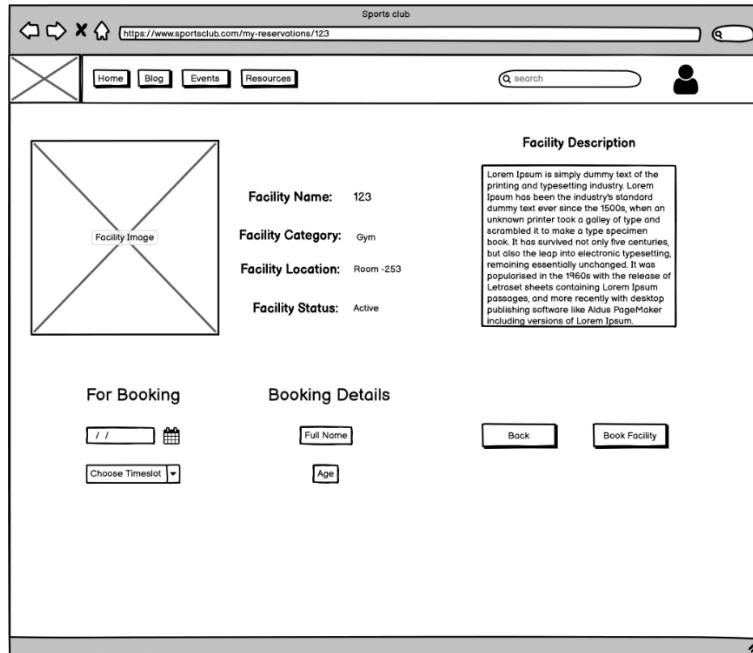


Figure 32: Wireframe of facility details page for reservation using Balsamiq [3].

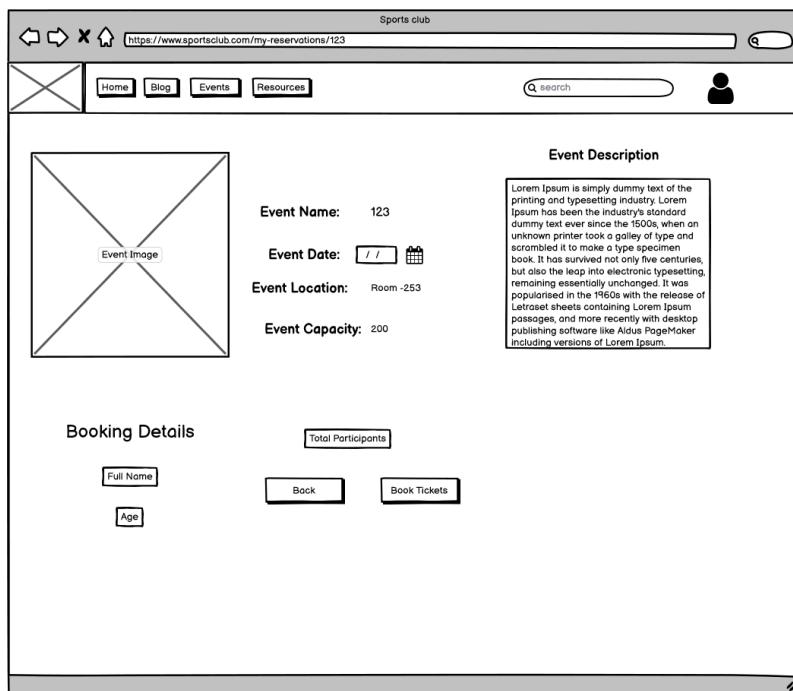


Figure 33: Wireframe of event details page used for event booking using Balsamiq [3] having similarities with facility details page.



Sportify MEMBERSHIP FACILITY EVENTS BLOGS STORE

Facility Details

Facility Name:	Stationary Bicycle
Facility Category:	Gym
Facility Location:	Room-254
Facility Status:	Active

Facility Description

A stationary bicycle (also known as exercise bicycle, exercise bike, spinning bike, or exercycle) is a device used as exercise equipment for indoor cycling. It includes a saddle, pedals, and some form of handlebars arranged as on a (stationary) bicycle.

For Booking

Choose Date	<input type="button" value="Calendar"/>
Choose Timeslot	<input type="button" value="▼"/>

Self Booking

Booking Details

Full Name*	Soham Patel
------------	-------------

BACK **BOOK FACILITY**

Figure 34: Facility details page displaying details and form for booking facilities.

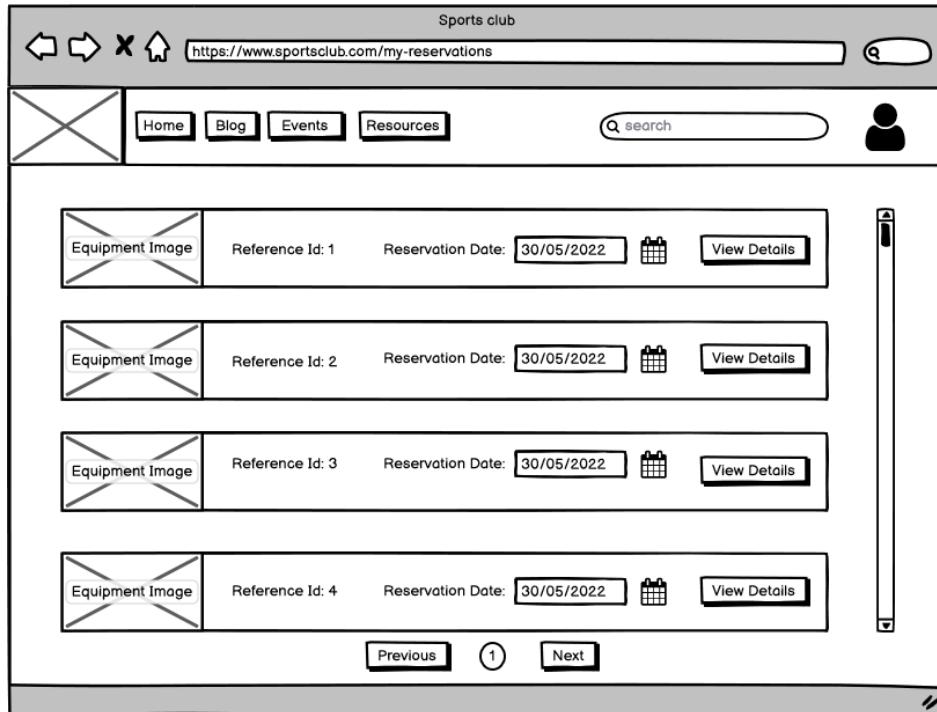


Figure 35: Wireframe of my reservations page using Balsamiq [3].

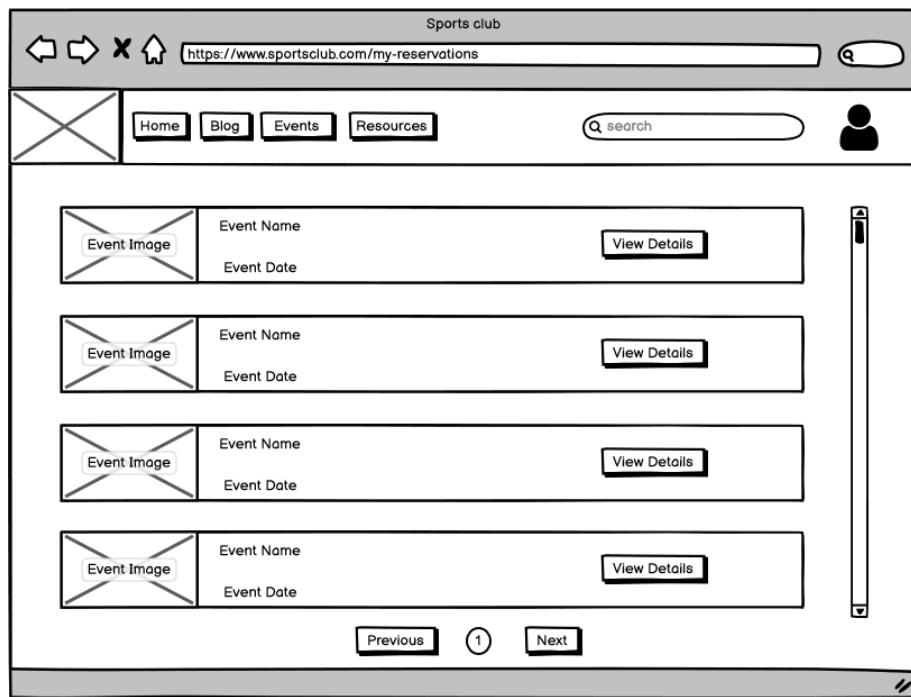


Figure 36: Wireframe of my events page using Balsamiq [3] having similarities with my reservations page.

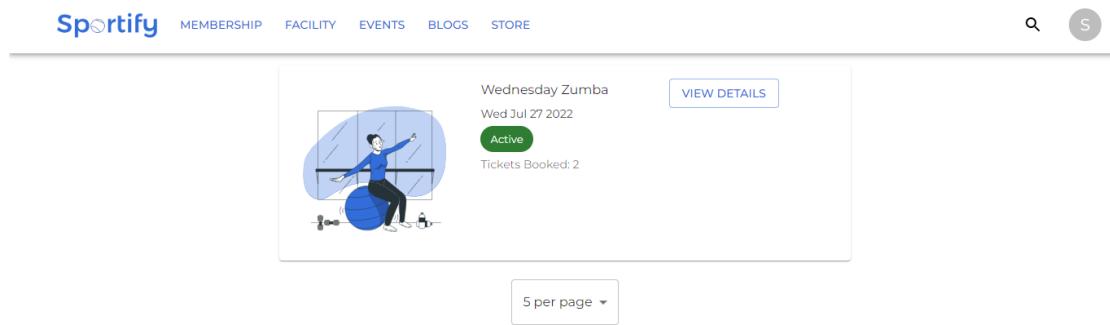


Figure 37: UI design of “my events” page having similarities with “my reservations” page.

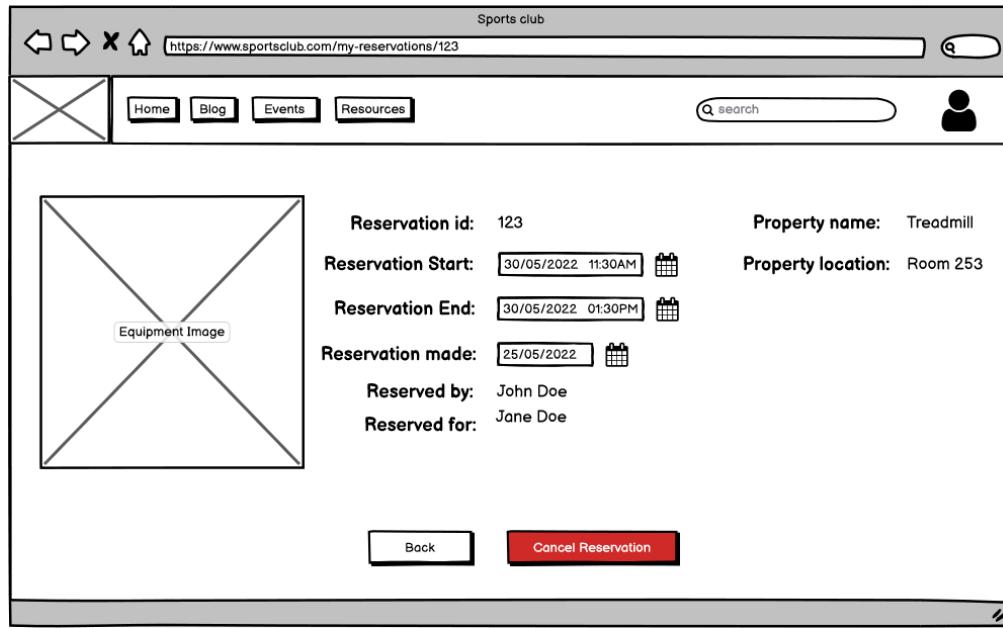


Figure 38: Wireframe of reservation details page using Balsamiq [3].

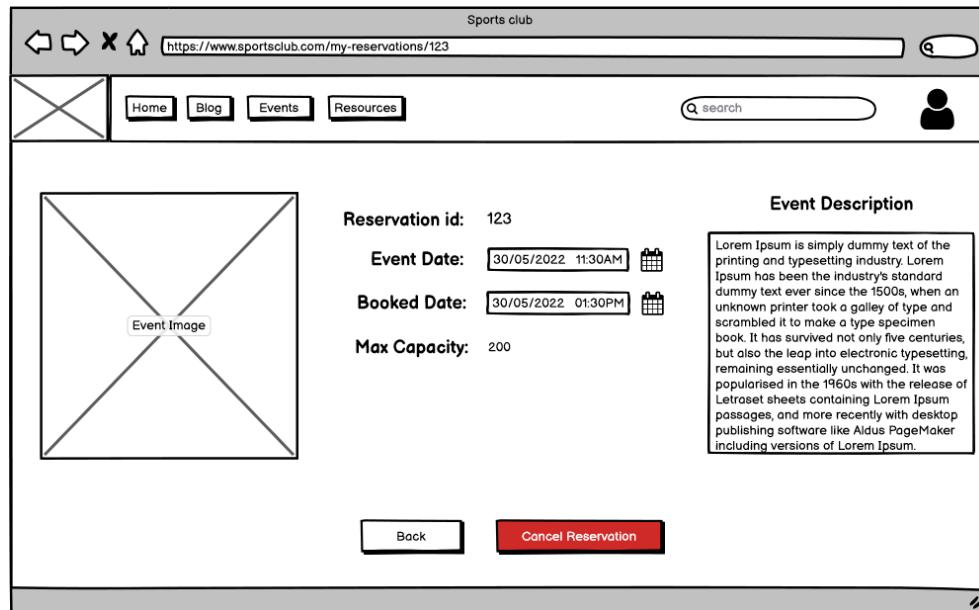


Figure 39: Wireframe of booked event page using Balsamiq [3].

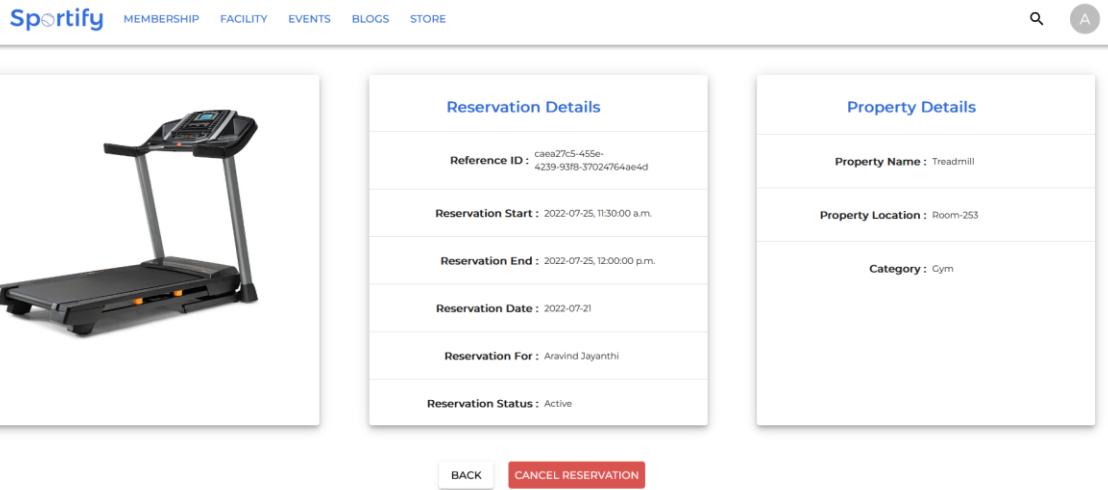


Figure 40: UI design of reservation details having similarities with event booking details page.

Payment Management:

This feature allows customers to pay for the membership plans they have signed up for. It is accomplished by using stripe.

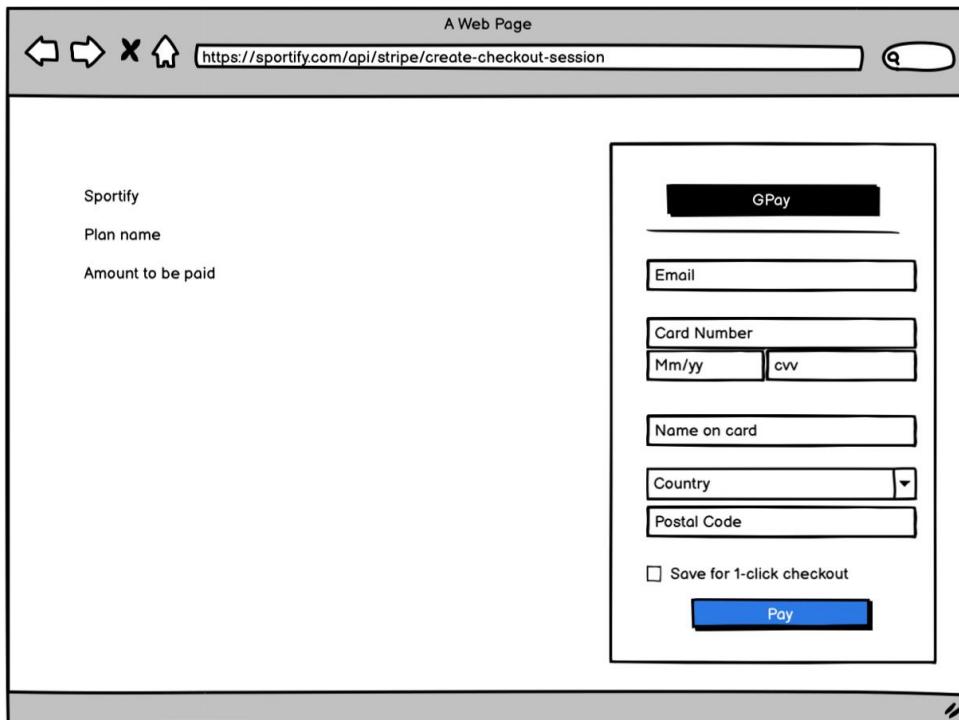


Figure 41 Wire Frame for Payment Feature with G-pay and Card

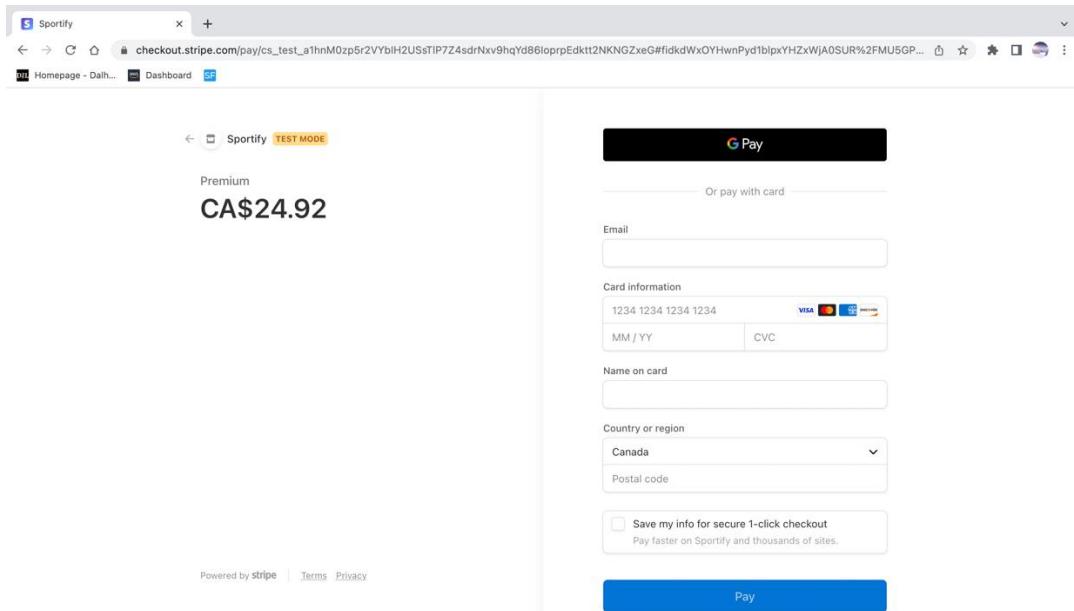


Figure 42 Snapshot of the payment feature

Merchandise Product Management:

This feature provides customers to view the club merchandise.



Figure 43 Wireframe for merchandise page

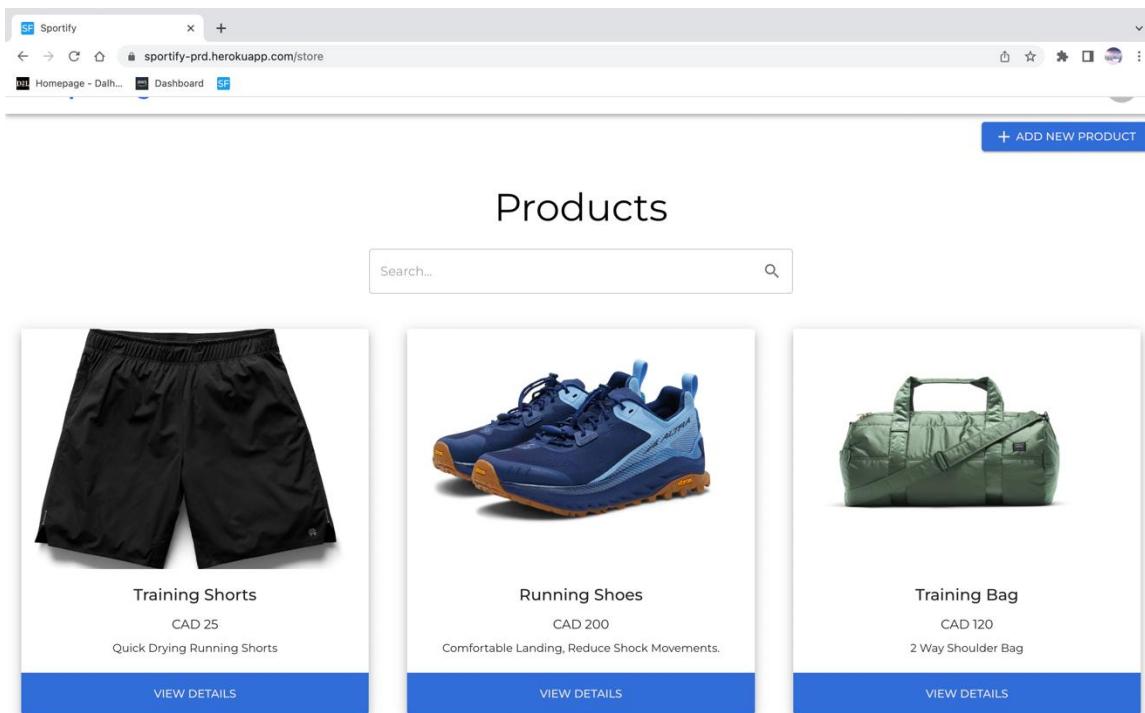


Figure 44: UI design of “store” page having similarities with “my events” page.

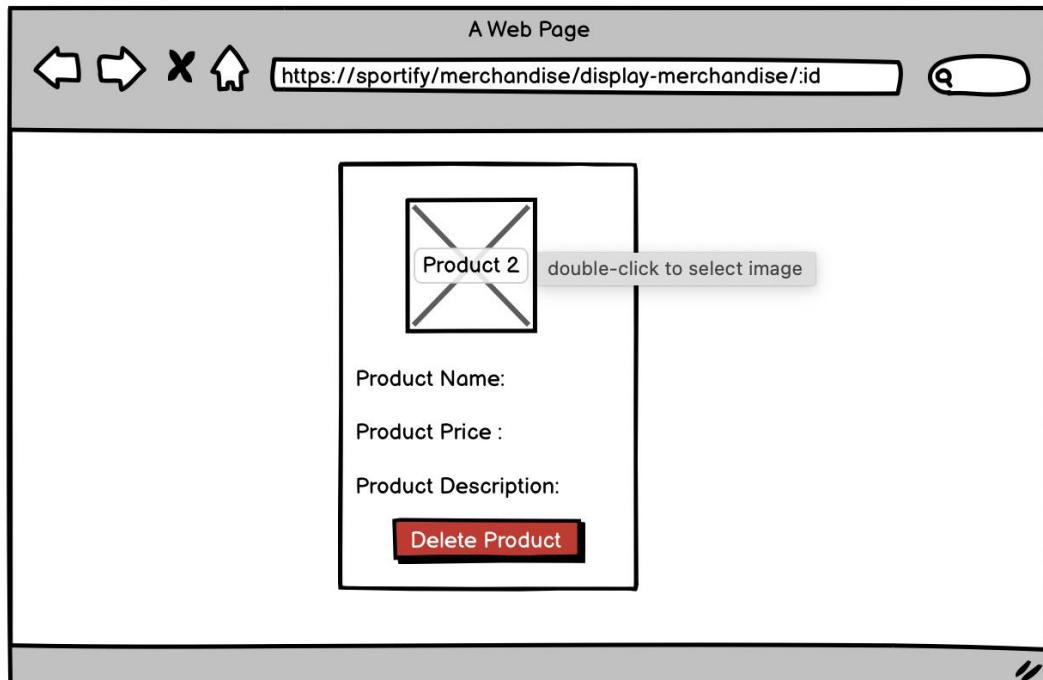


Figure 45: Wireframe for view product page

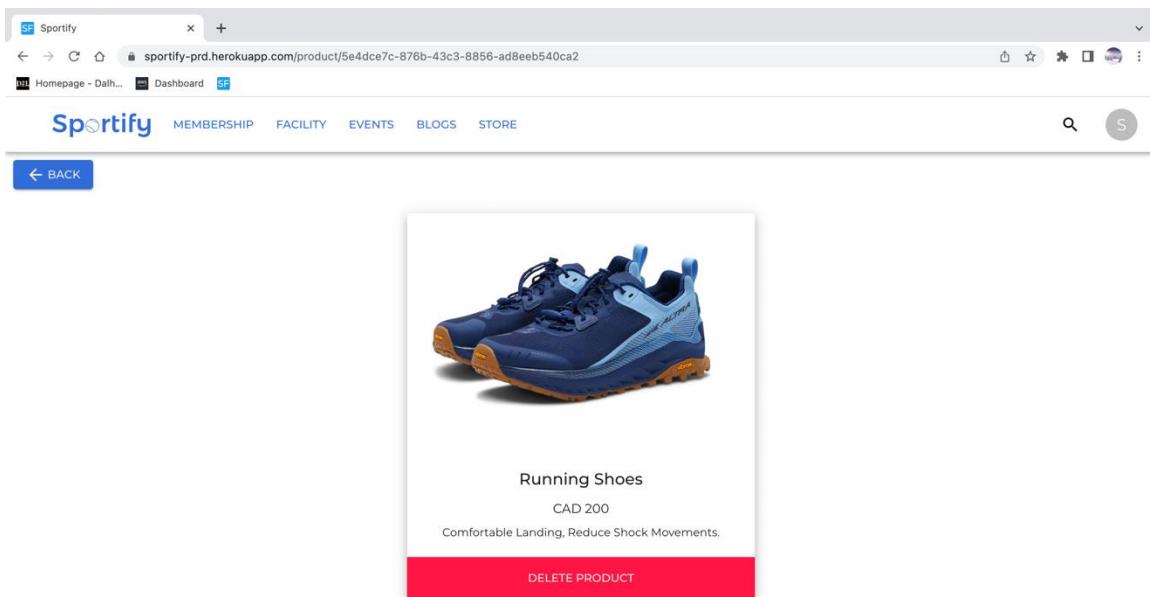


Figure 46: Snapshot of the view product page

Rewards System:

It provides users some perks like coupon code which will give them some discount for some brands and other things. User also has earned some points based on what money they spent on membership.

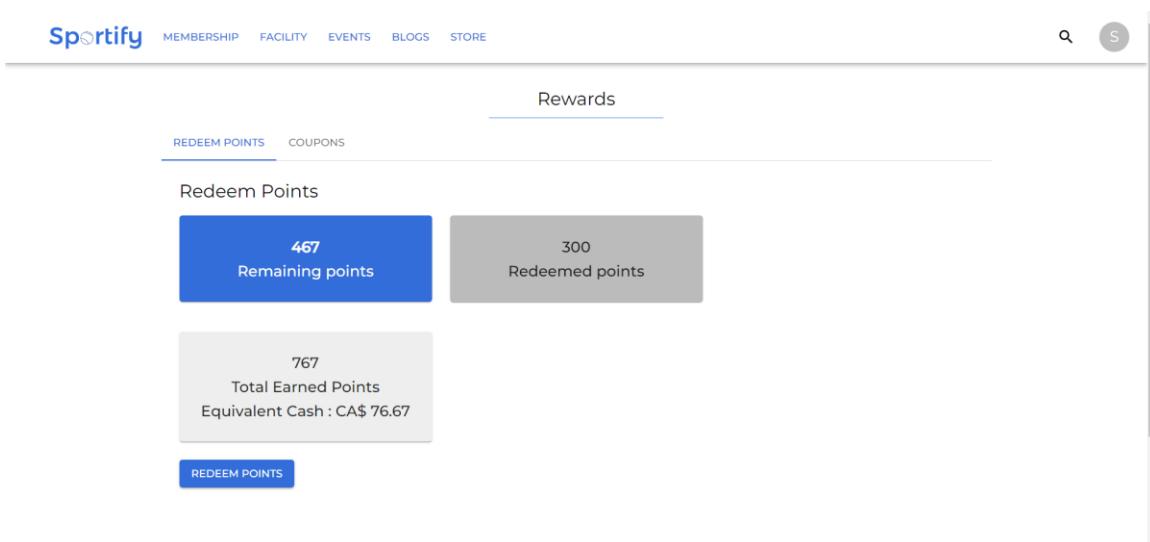


Figure 47: UI design for redeem reward points.

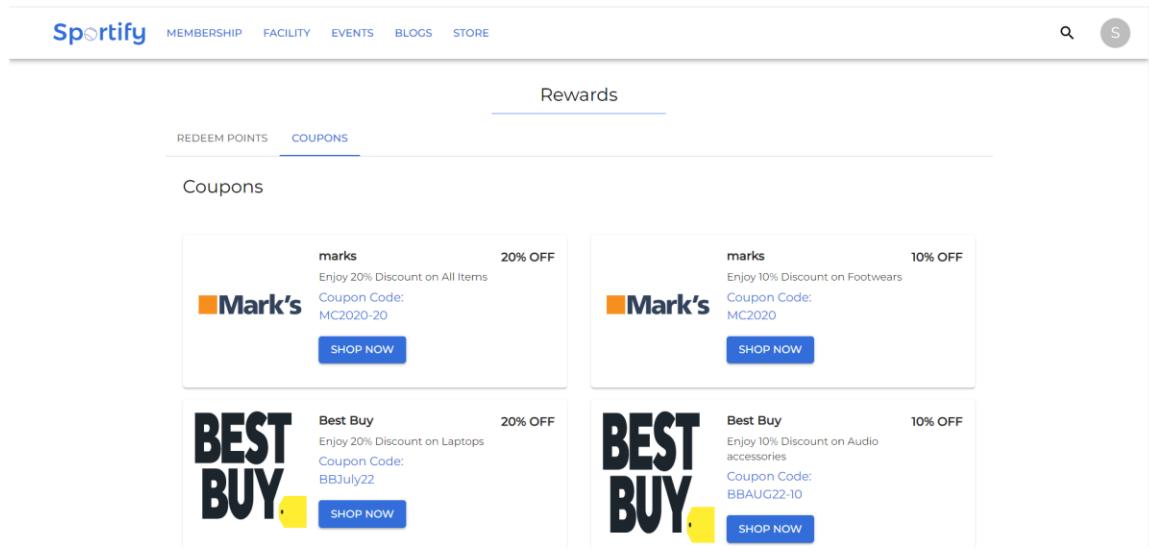


Figure 48: UI page for coupons.

Search Function:

The search function has a main page which provides quick search for the facilities. This is because facilities being the feature which will be most searched by users as it is the main part of sports management.

The main page provides option to user for what they want to search in the application. This simplifies the search for users as well as search results. The main aim for the search is to make sure user is getting what they are searching in the easiest way. The main page of search displays the options store, facility, event, and blogs which the user can select to proceed with search.

The search function has auto-complete feature which completes the search results of the user again simplifying the search for the user. The function provides pagination so that user does not find difficulty in looking at results and they don't have to waste their time looking at the results for their search. Moreover, the users get option to register for facility in facility search, option to register for events in event search, and option to view searched blogs in blog search. Thus, saving their time in proceeding with that activity.

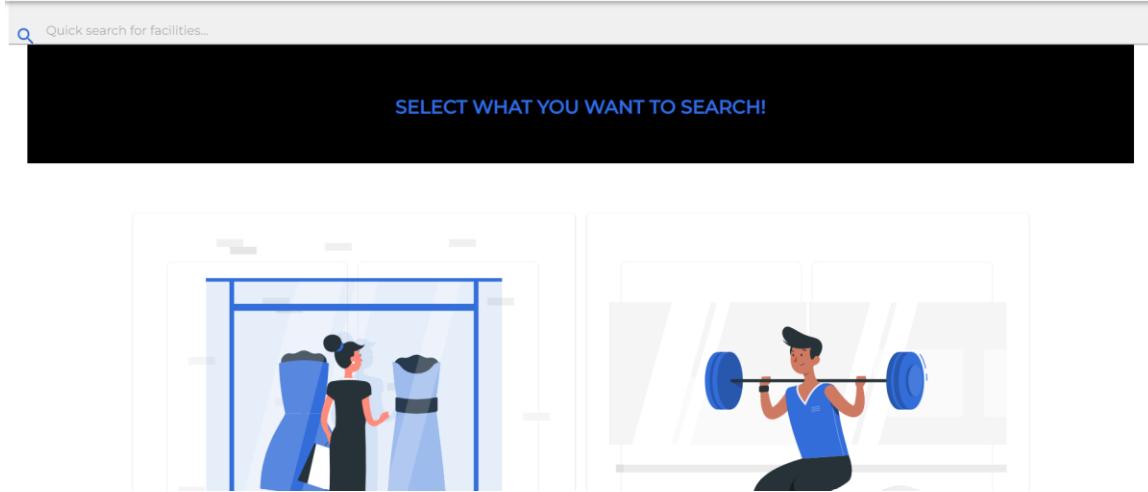
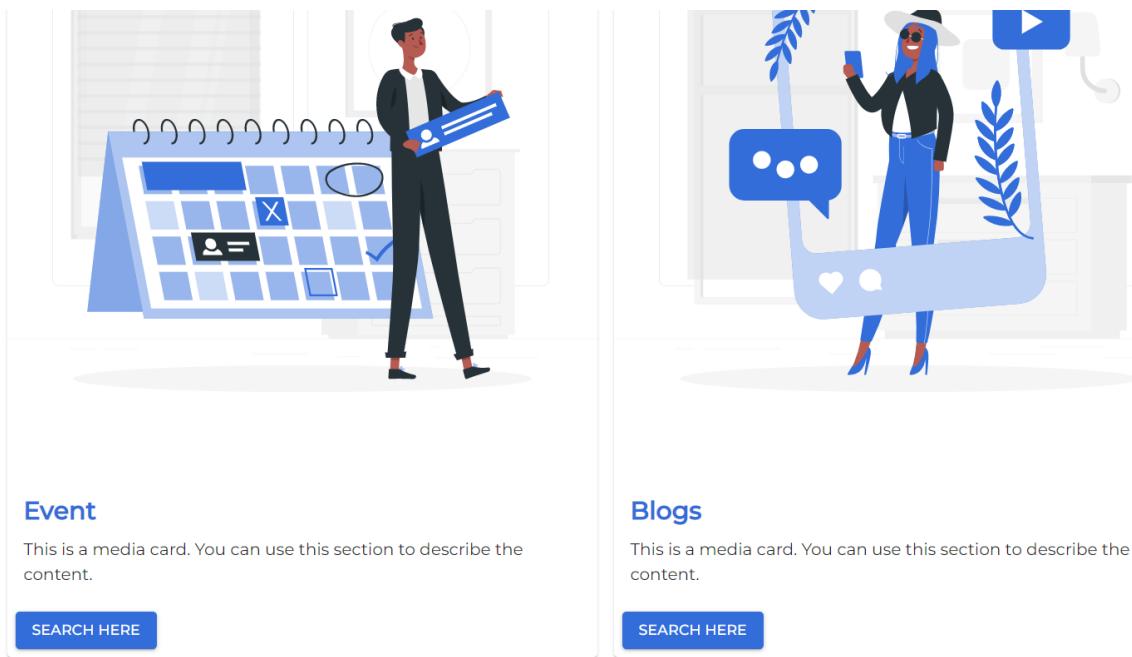


Figure 49: Main Page of search



The image displays two media cards for searching. On the left, under the heading "Event", there is an illustration of a person holding a blue ticket stub in front of a calendar. Below the illustration is the word "Event" and a brief description: "This is a media card. You can use this section to describe the content." A blue "SEARCH HERE" button is located at the bottom. On the right, under the heading "Blogs", there is an illustration of a person holding a smartphone. Below the illustration is the word "Blogs" and a brief description: "This is a media card. You can use this section to describe the content." A blue "SEARCH HERE" button is located at the bottom.

Figure 50: Main page of search showing options to search

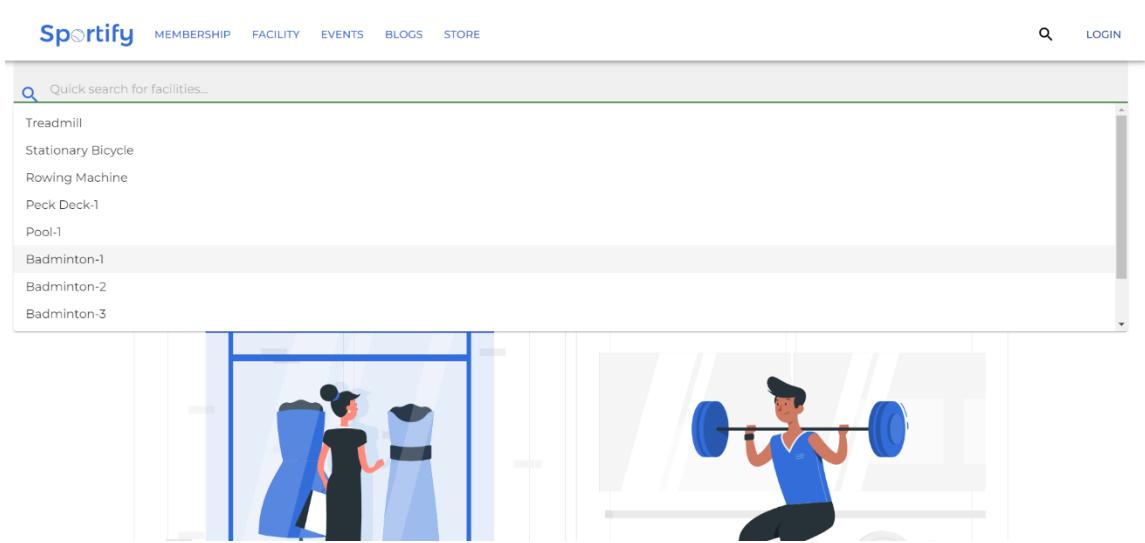


Figure 51: Main Page of search showing quick search for facilities

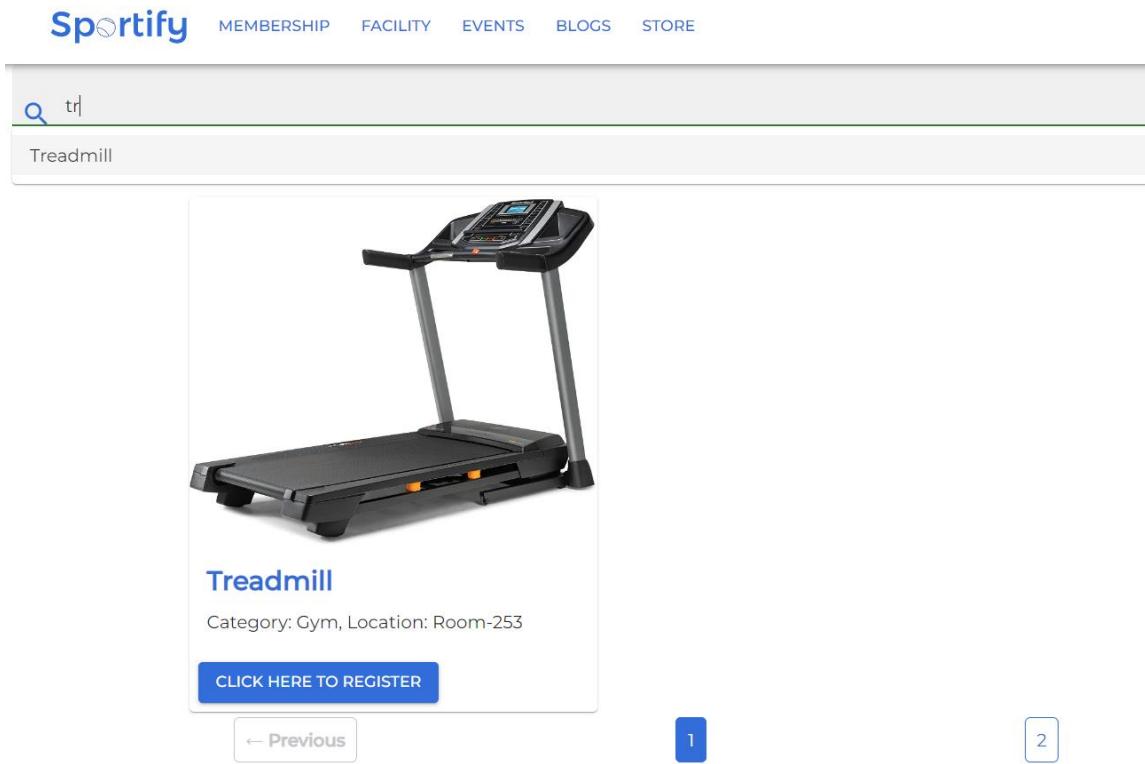


Figure 52: Searching with auto complete in main page

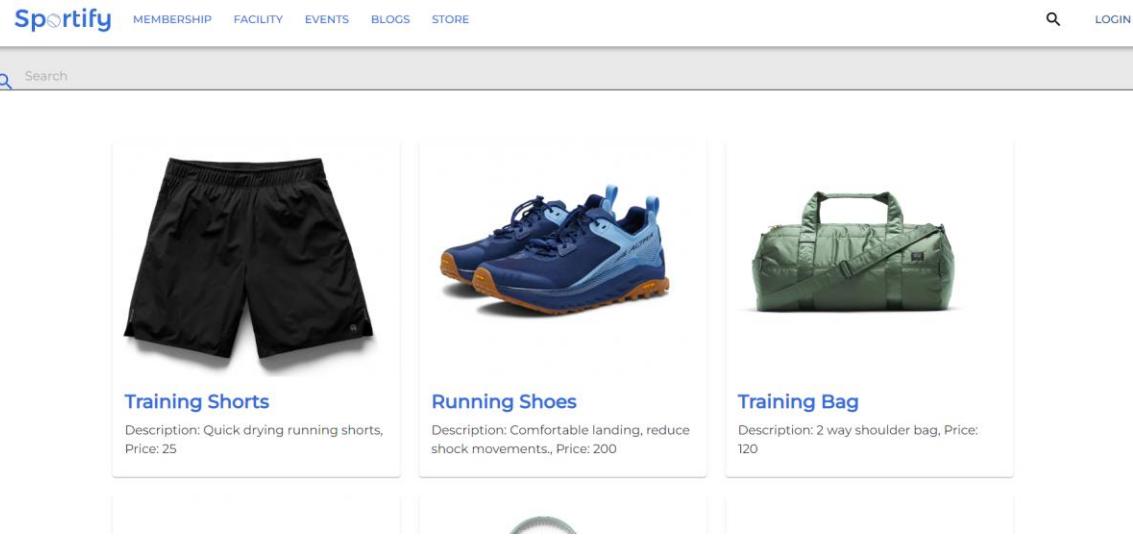


Figure 53: Search for merchandise

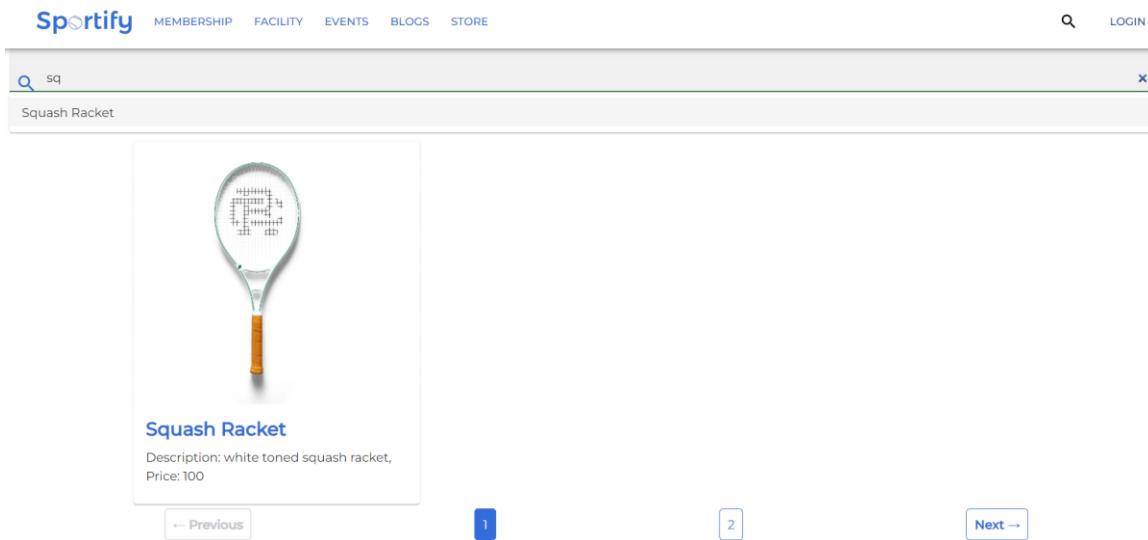


Figure 54: Search for Merchandise with auto complete and pagination

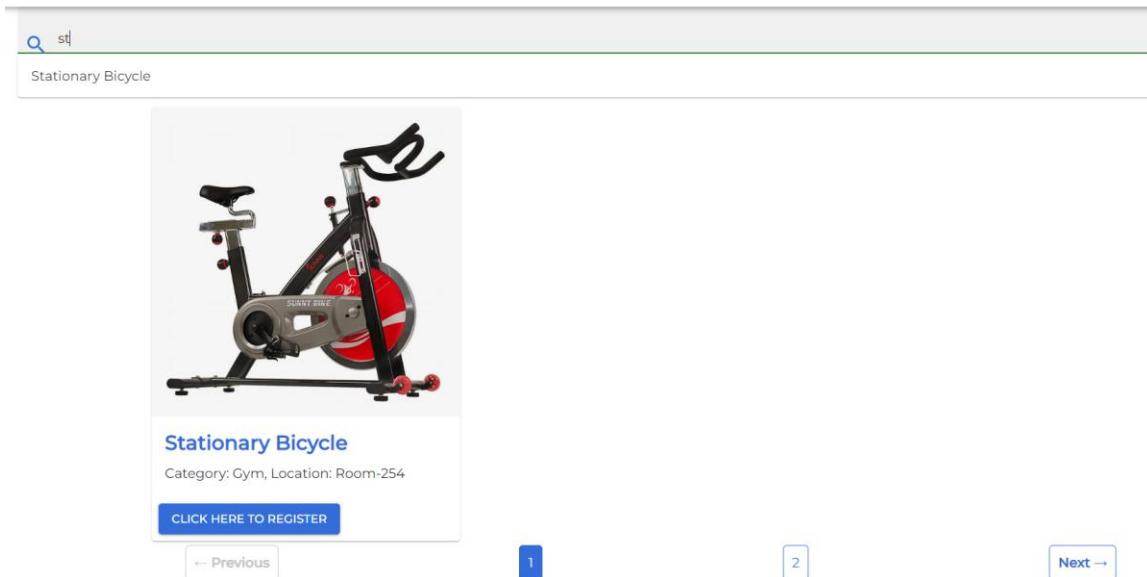


Figure 55: Search for Facilities with auto complete and pagination

Sportify [MEMBERSHIP](#) [FACILITY](#) [EVENTS](#) [BLOGS](#) [STORE](#) [LOGIN](#)

Search

- Sunday Yoga
- Monday Meditation
- Wednesday Zumba
- Thursday Yoga
- Tuesday Meditation
- Tuesday Meditation
- Tuesday Meditation
- Tuesday Meditation

Sunday Yoga

Location: Room - 234, Available: undefined

[CLICK HERE TO REGISTER](#)

Monday Meditation

Location: Room - 283, Available: undefined

[CLICK HERE TO REGISTER](#)

Wednesday Zumba

Location: Room - 200, Available: undefined

[CLICK HERE TO REGISTER](#)

Figure 56: Search for Events with auto complete and pagination

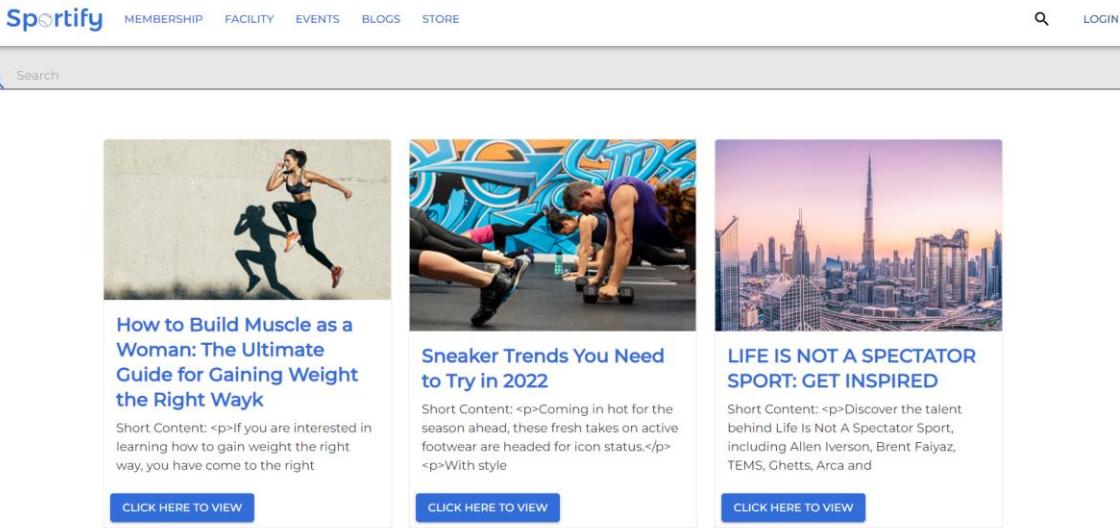


Figure 57: Search for blogs page

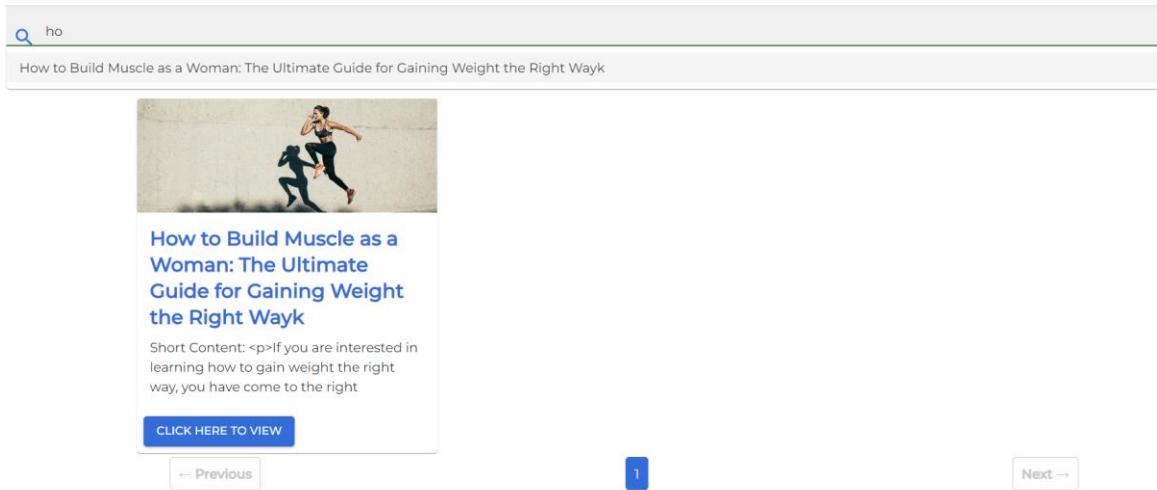


Figure 58: Search for Blogs with auto complete and pagination

Blogging

This feature allows customers to engage in a blogging environment. Only customers with membership plans can view and create blogs. The main goal is to create a healthy community where everyone can benefit from the other. The tasks associated with this feature are:

1. Create Blogs

2. View all Blogs
3. Edit/Delete Blogs
4. View your Blogs

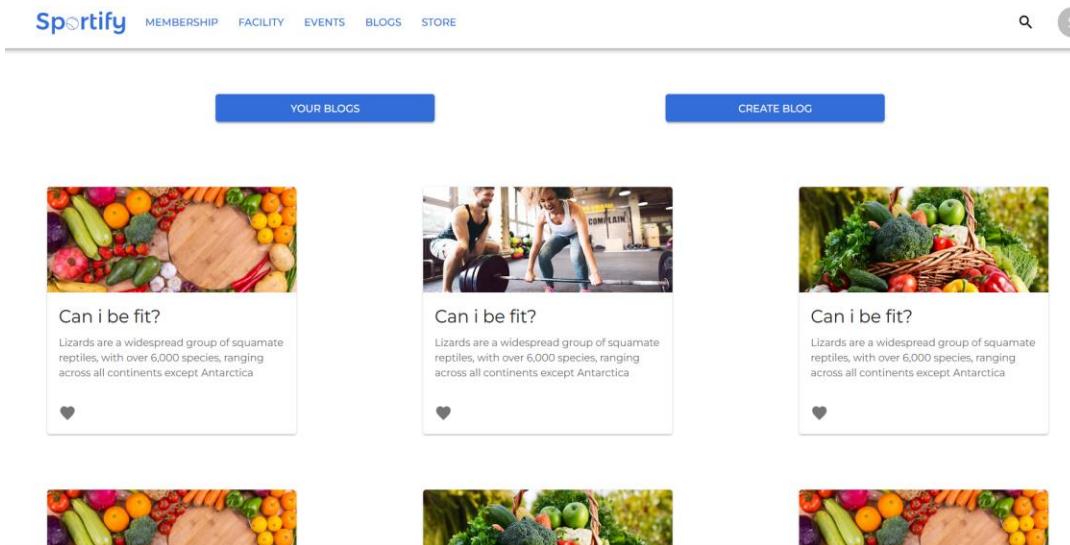


Figure 59: All Blogs Screenshot

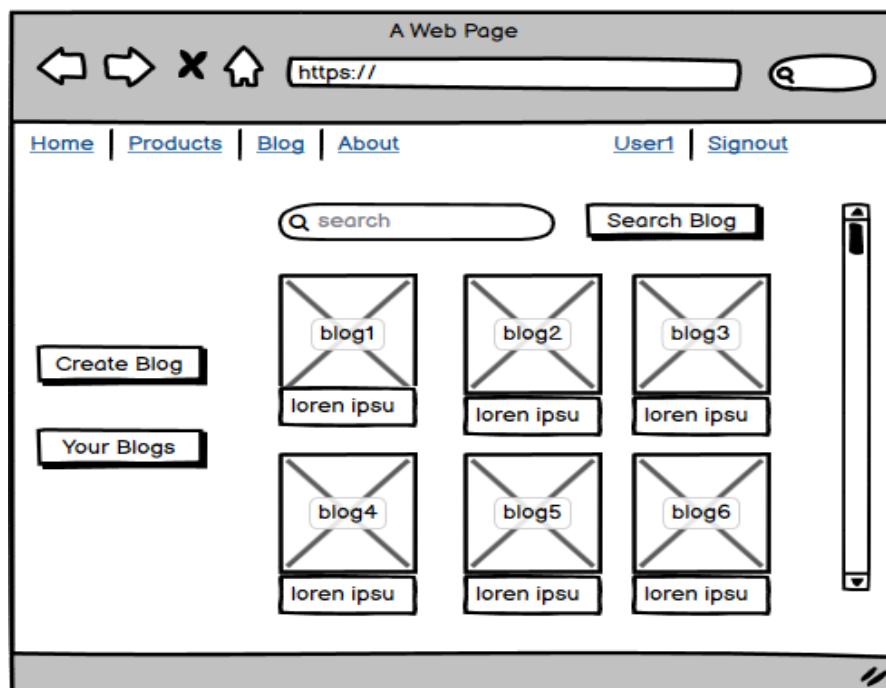


Figure 60: All Blogs Wireframe using Balsamiq [3]

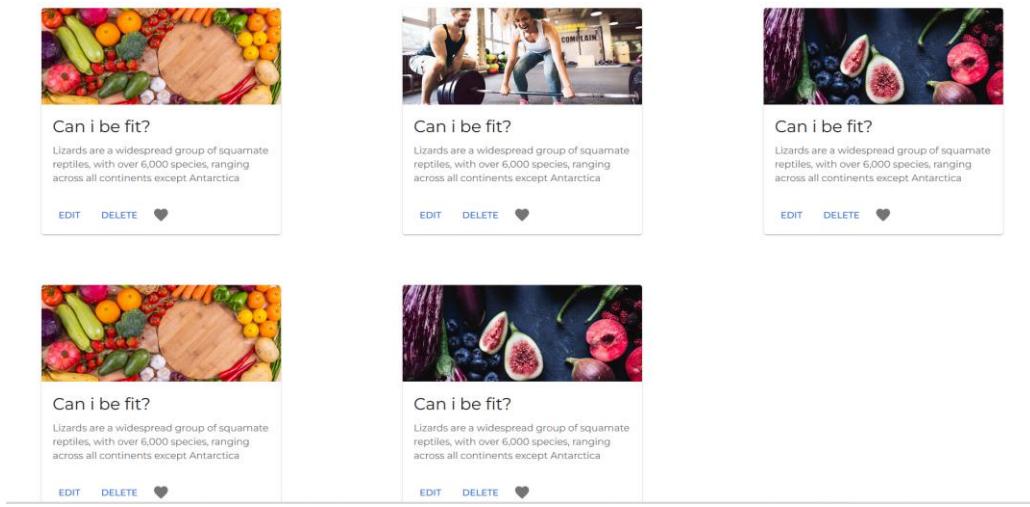


Figure 61: Your Blogs Screenshot

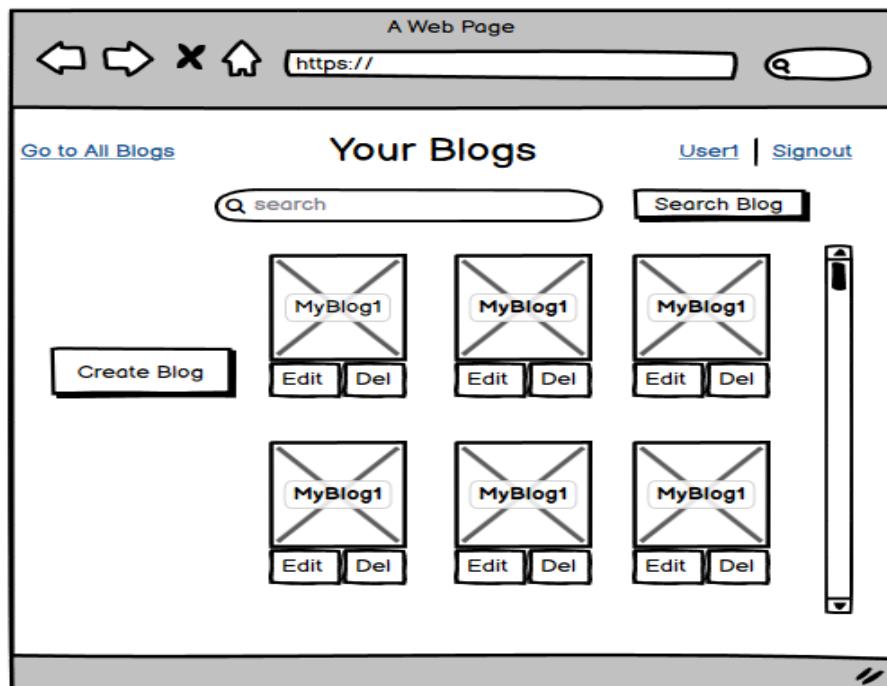


Figure 62: Your Blogs Wireframes

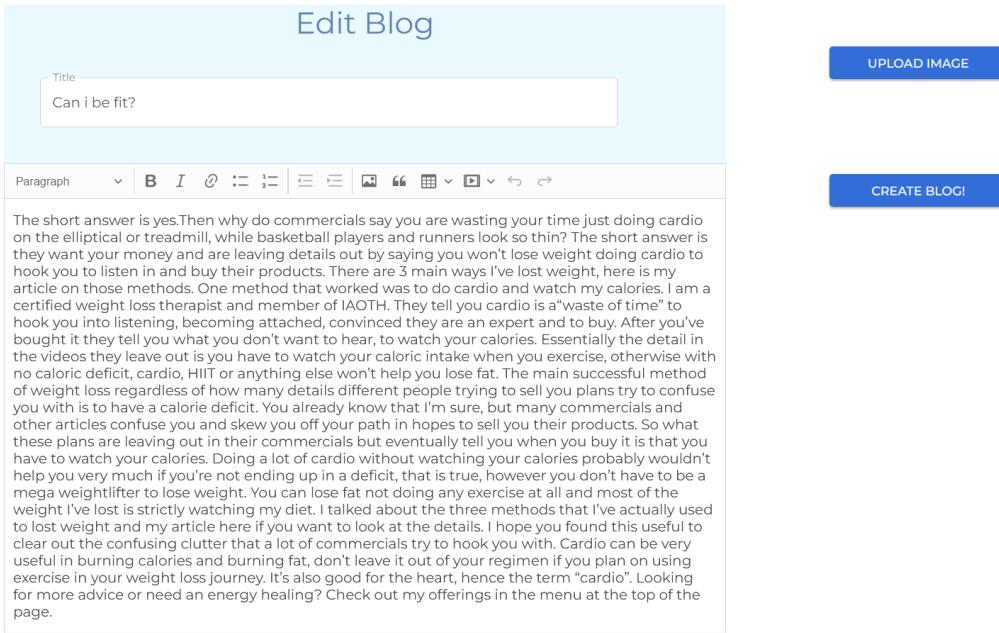


Figure 63: Edit Blog Screenshot

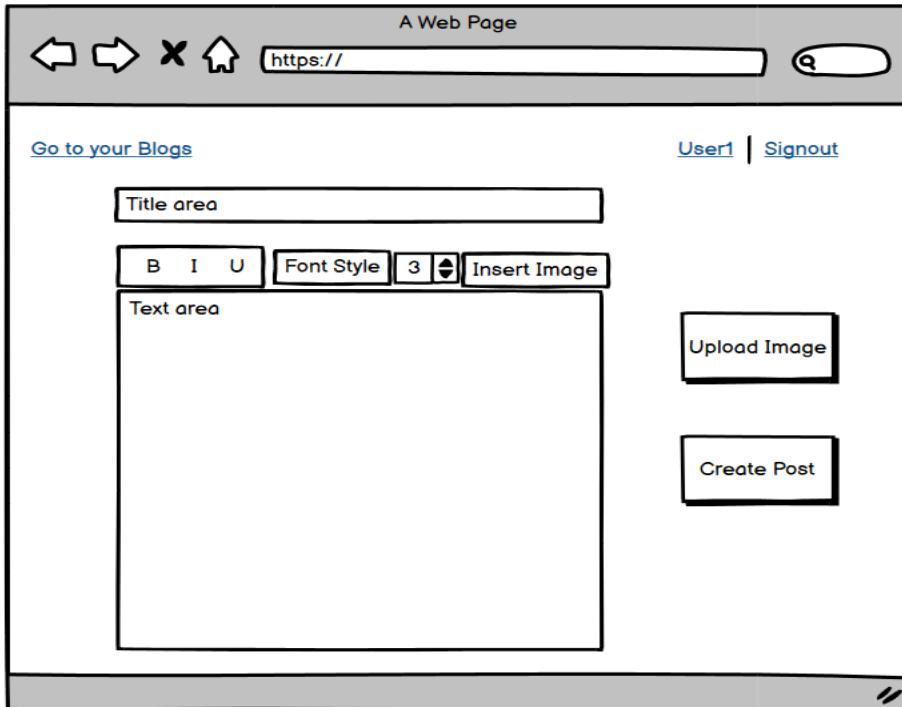


Figure 64: Edit Blog Wireframe

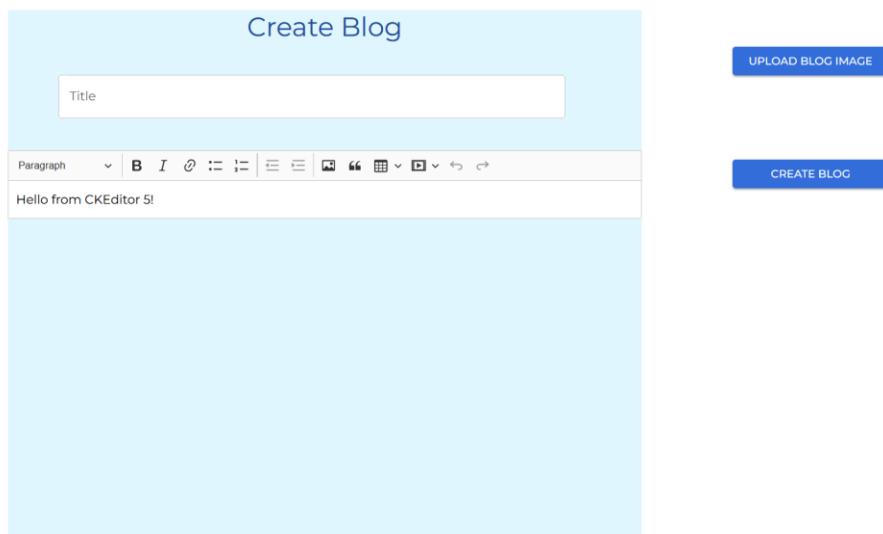


Figure 65: Create Blog Screenshot

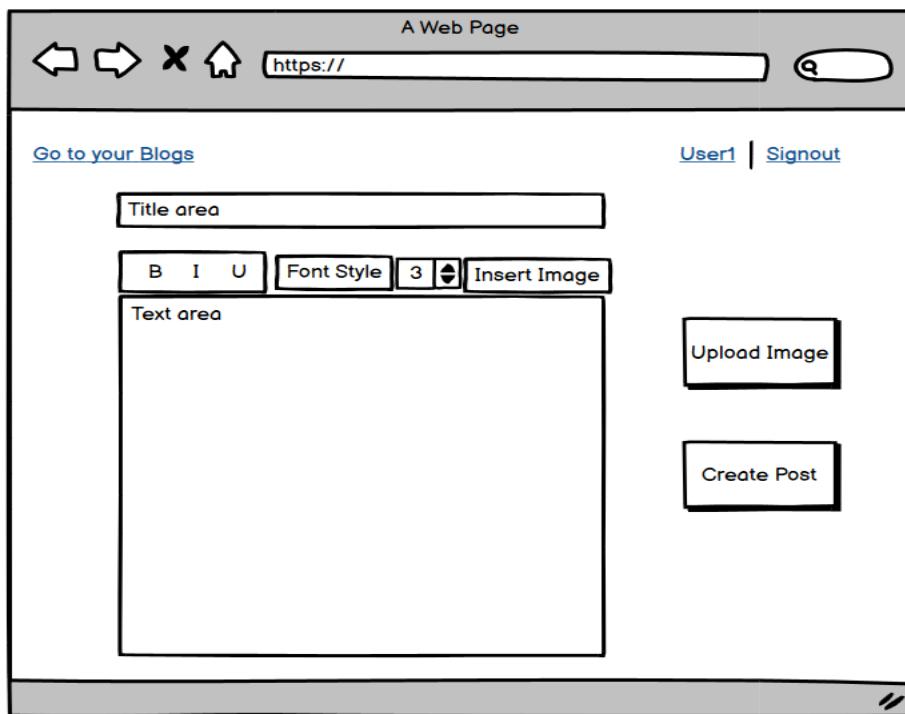


Figure 66: Create Blog Wireframe

4. APPLICATION WORKFLOW

The application workflow of the Sportify project is depicted in the application architecture shown in **Figure 67**. The application is mainly divided into three main components:

1. Frontend using ReactJS, React-Router and Material UI.
2. Backend using NodeJS, Express.
3. MongoDB database.

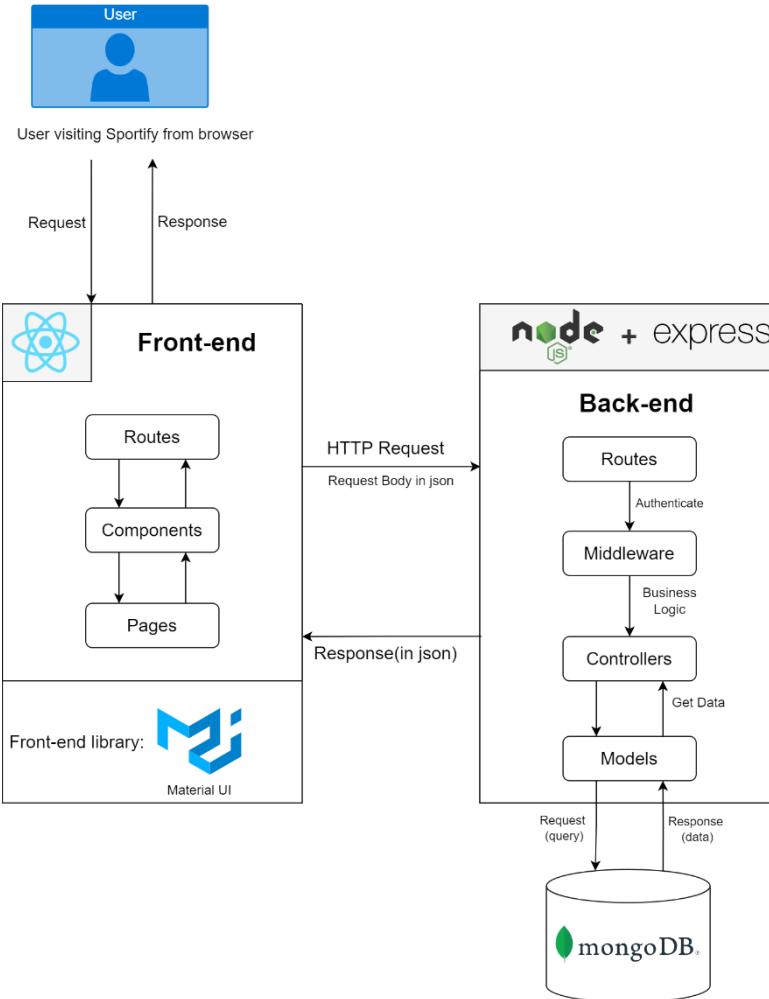


Figure 67: Spotify application architecture using draw.io [2][10].

The browser will download the minified code from the frontend server when the user enters the URL. The react-router will render the appropriate component(s) based on the URL route mentioned in the browser tab. The communication with the backend service will happen through REST APIs, and the data is transmitted to and from the frontend using JSON format.

The backend service for this project is developed with NodeJS and Express using layering architecture with routes, middleware, controllers, and models as the layers. All the REST API will be sent to the routing layer, which will route the request to appropriate business logic. Then the request is sent to middleware for authentication and authorization. If the request is valid and authenticated, the controller layer will execute the business logic by communicating with other entity controllers. All the queries to the

database go through the model layer that is built using Mongoose ODM (Object-Document Mapper).

4.1 Interaction Design

Figure shows the legends used for all the task flow diagrams that follows in this document.

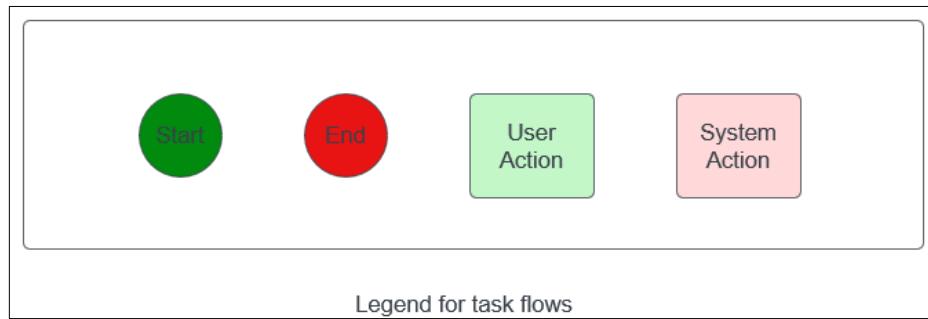


Figure 68: Legends used for all task flow diagrams

4.1.1 User Management

Task 1: User Signup

Scenario: A fitness enthusiast wants to join the fitness club to go to gym do some exercise and want some training from some trainer. So, he visits the website in laptop and finding some packages for training so now he wants to join the club (get membership) for that he must register in the system.

Use Case:

1. The user opens the browser and goes to the website's homepage.
2. The user clicks on the Signup button.
3. Application displays page with the registration/signup form
4. The user must enter details needed like the first name, last name, email address, password, confirm password, and other information and click on Submit button.
5. The application authenticates all the details.
 - 5.1. If any details are invalid, then show a message with some instruction
 - 5.2. The user enters details for invalid fields again.
6. All the details are valid, then the form will be submitted, and the user will be registered successfully.
7. The application sends a notification to the user that you are successfully registered and check your mail.
8. The user will get a verification email with a verification link in their inbox.

9. User clicks on the verification link in the mail to verify their account.
10. The application will verify the link.
 - 10.1. If it is invalid, show the notification that the link is broken.
 - 10.2. User must generate the link again and click on the link which they will get in the inbox.
11. After the valid link, it shows the message to the user that your account is successfully verified and redirects to the login page.

As you can see, the task flow diagram for the signup task is in Figure 69

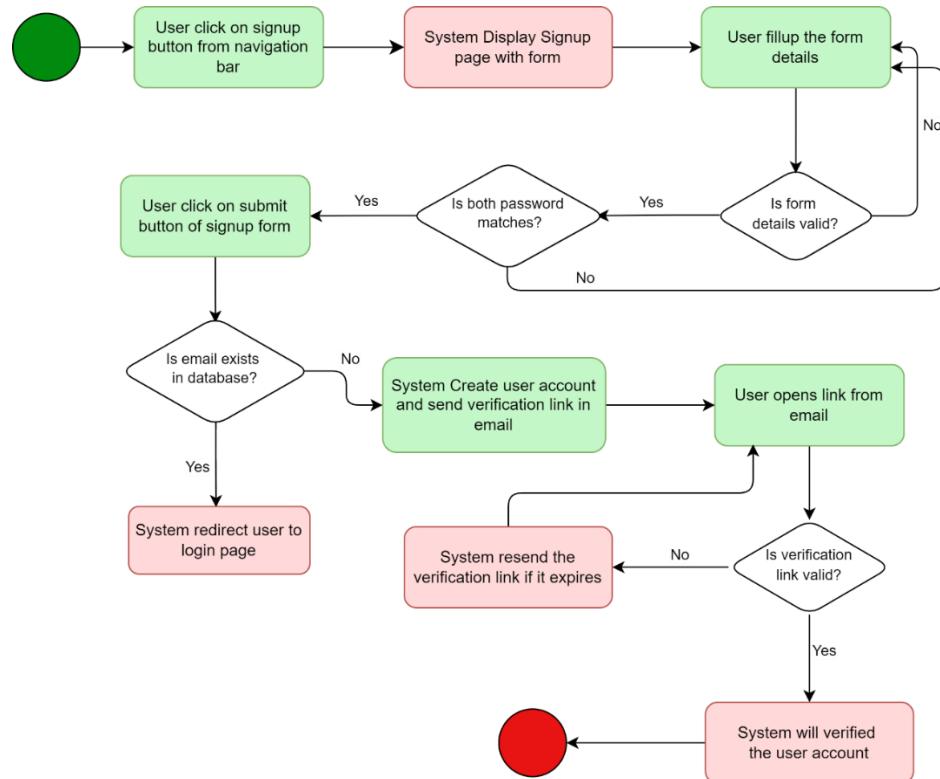


Figure 69: Task flow diagram of “Signup” using draw.io [2][10]

As you can see, the clickstream for the signup task is in Figure 70

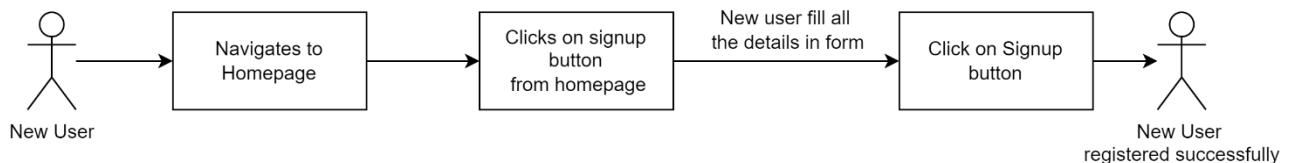


Figure 70: Clickstream of “Signup” using draw.io [2][10]

Task 2: User Login

Scenario: A customer already had an account and now he/she wants to see the profile details in their account.

Use Case:

1. The user opens the browser and goes to the homepage of the website
2. The user clicks on the Login button
3. The application displays the page with the login form
4. User must enter details which are needed, like email and password, and click on the Login(submit) button
5. The application authenticates all the details
 - 5.1. Users can't submit the form if any details are invalid.
 - 5.2. The application will show the instructions to the user if they enter the wrong
6. All the details are valid, then click on the Login(submit) button
 - 6.1. If the user enters the wrong credentials, then after four tries, they get suspicious activity mail to reset the password if they want
7. After successfully login application redirects the user to my account page

As you can see, the task flow diagram for the login task is in **Figure 71**

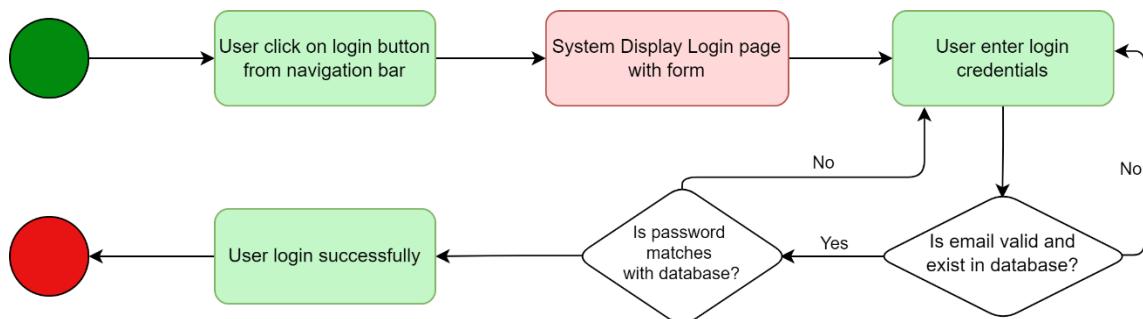


Figure 71: Task flow diagram of “Login” using draw.io [2][10]

As you can see, the clickstream for the login task is in Figure 72.

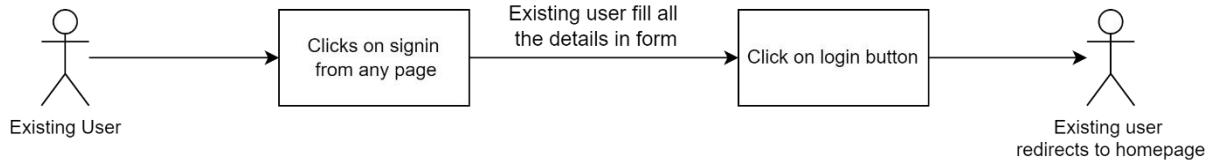


Figure 72: Clickstream of “Login” using draw.io [2][10]

Task 3: User Logout

Scenario: A customer wants to log out from application because their work is finish here on the website.

Use Case:

1. For a user visiting any page on the website, there is log out button on the top right corner
2. The user clicks on the Logout button
3. The application tries to logout
 - 3.1. If there is an error, the user stays on the same page they are visiting and receives some error message.
4. The user will get a notification that you successfully log out and then redirect to the homepage.

As you can see, the task flow diagram for the logout task is in Figure 73.

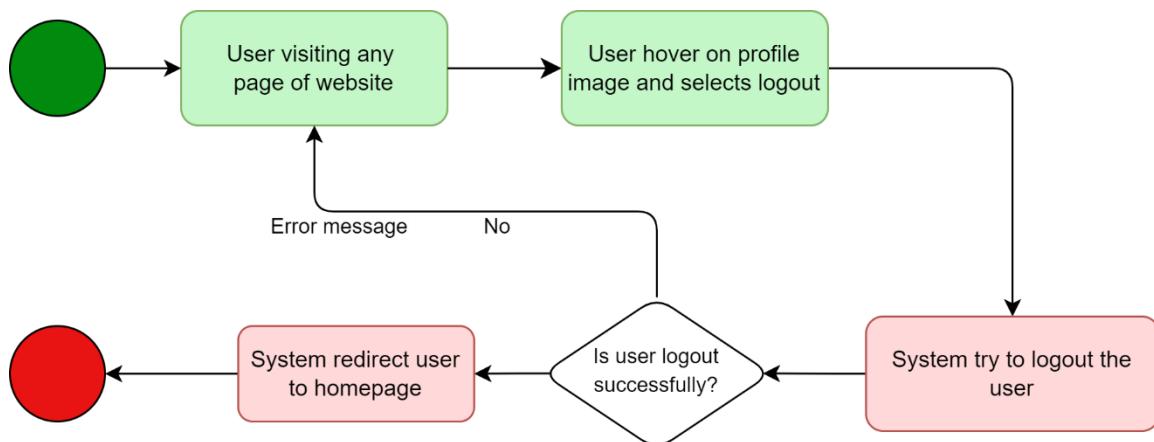


Figure 73: Task flow diagram of “Logout” using draw.io [2][10]

As you can see, the clickstream for the logout task is in Figure 74.

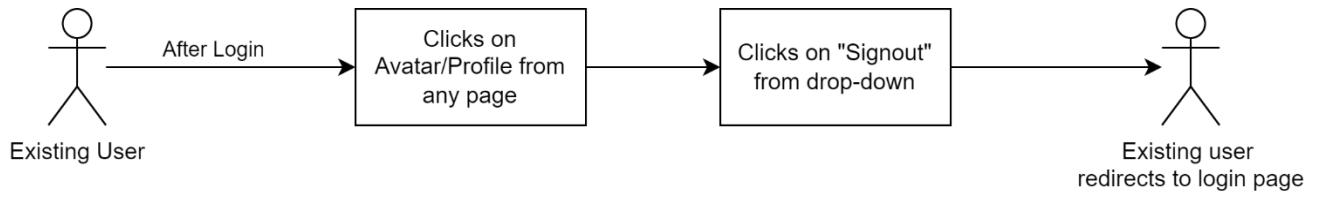


Figure 74: Clickstream of "Logout" using draw.io [2][10]

Task 4: User Profile Edit

Scenario: Any customer wants to see the details of themselves and wants to change some details because of some reason.

Use Case:

1. A user visiting any page on the website and hovering on the avatar with their name/image.
2. From the drop-down user selects my account.
3. The application displays user profile details on the page.
4. The user clicks on the edit button from the page.
5. Users must update their details in the form which is in modal.
6. Application check details entered by the user.
 - 6.1. The application will show instructions for invalid details.
7. If all the details filled by the user are valid, then a modal box will close by itself and give a notification that your profile is successfully updated.

As you can see, the task flow diagram for the user profile edit task is in Figure 75.

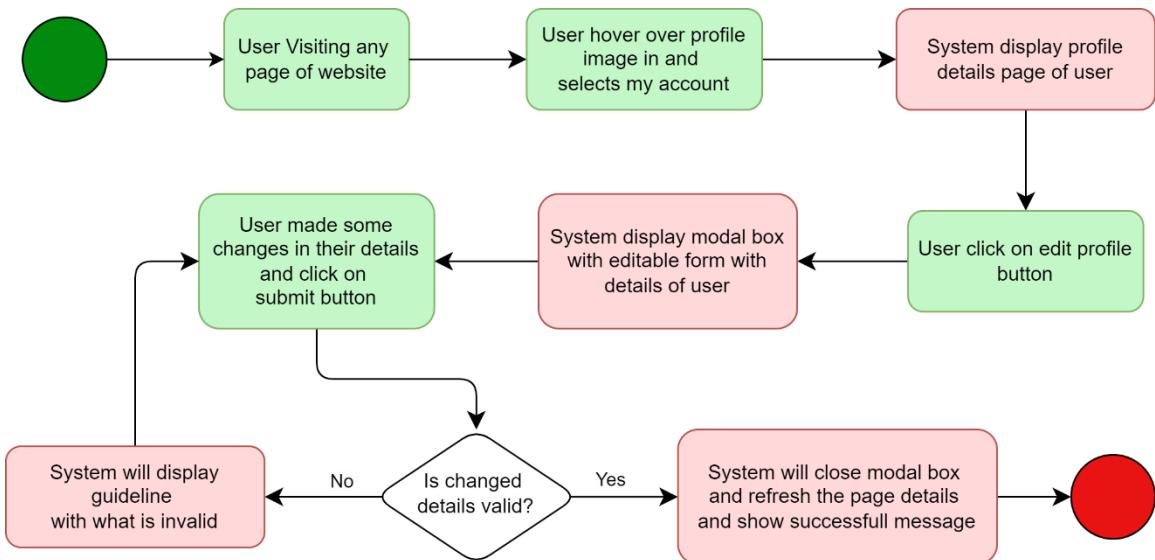


Figure 75: Task flow diagram of “User Profile Edit” using draw.io [2][10]

As you can see, the clickstream for the user profile edit task is in Figure 76.

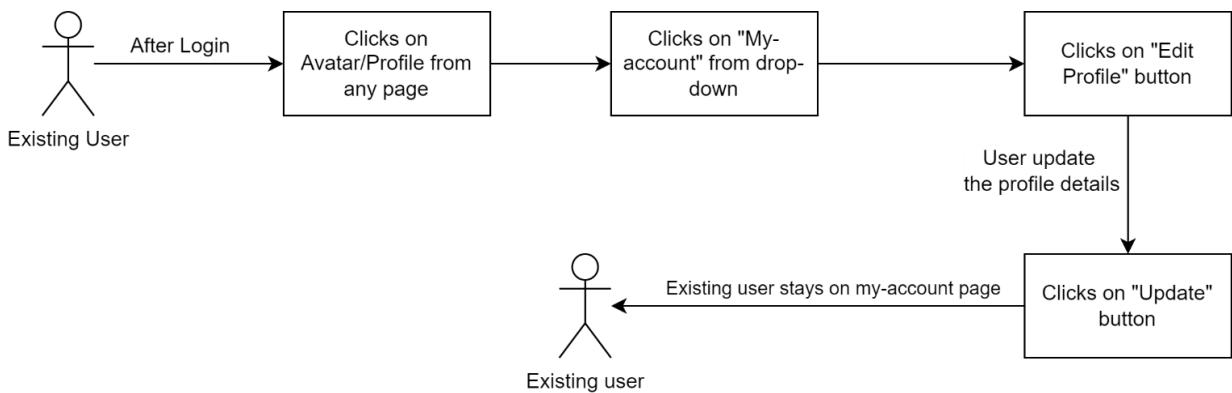


Figure 76: Clickstream of " User Profile Edit" using draw.io [2][10]

Task 5: User Profile delete

Scenario: Any client wants to delete profile because they are shifting in some other city so he/she don't require the equipment of club and membership.

Use Case:

1. The user visiting any page on the website and hovering on the avatar with their name/image.
2. From the drop-down user selects my account.
3. The application displays user profile details on the page.

4. The user clicks on the delete button from the page.
5. The application displays a modal box with a message to delete an account with “OK” and “Cancel” options.
6. User Click on "OK," then the data will be deleted, which is connected to their account.
 - 6.1. The user selects "Cancel," then the modal will be closed, and the user stays on the profile page.
7. The application will notify the user that your account has been successfully deleted.

As you can see, the task flow diagram for the user profile delete task is in Figure 77.

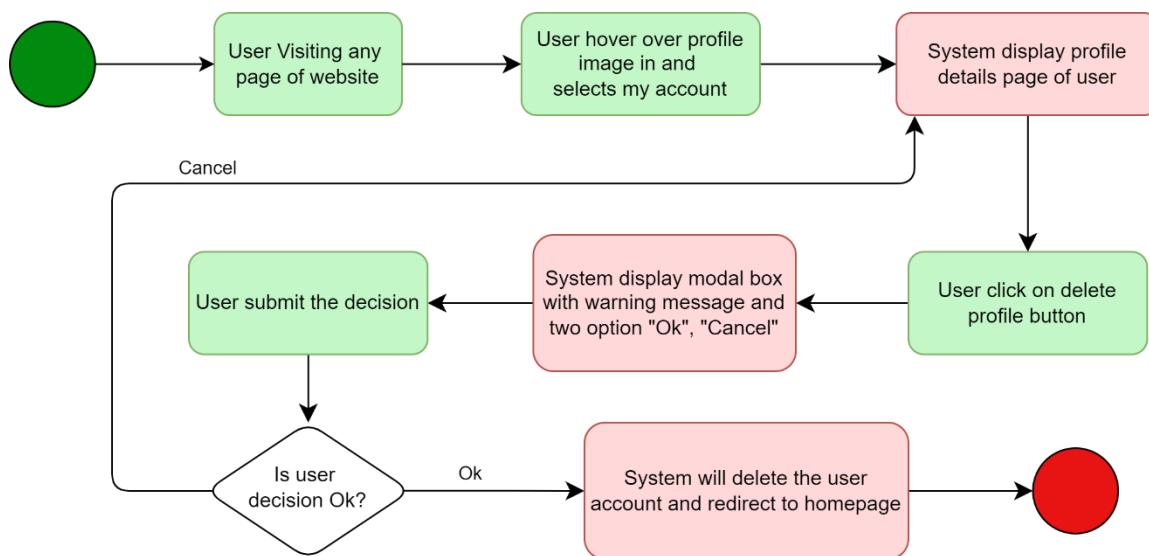


Figure 77: Task flow diagram of “User Profile Delete” using draw.io [2][10]

As you can see, the clickstream for the user profile delete task is in Figure 78.

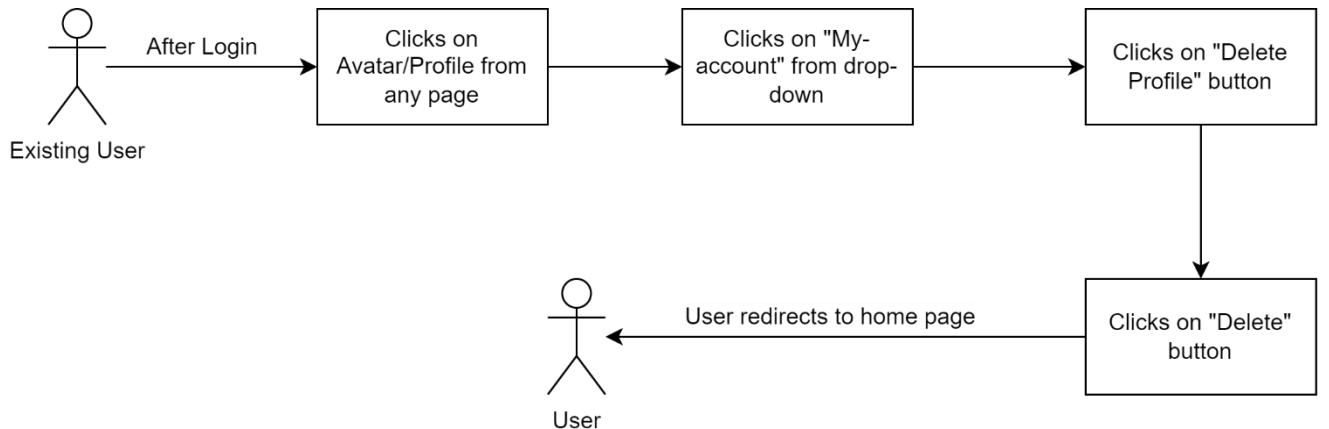


Figure 78: Clickstream of " User Profile Delete" using draw.io [2][10]

Task 6: Forgot Password

Scenario: A customer forgot the password because of some reason and now they want to change the password to access their account again.

Use Case:

1. The user opens the browser and goes to the webpage of our login page.
2. The application displays the login page.
3. The user clicks on forgot password link.
4. The application displays the forgot password page with the form.
5. The user must enter the email address which links to the account in the application.
6. Application check email for the association to any account
 - 6.1. If email is associated with the account, then forgot password link sent to the user mail
 - 6.2. If the email is not associated with any account, then nothing happened, and show the message to enter a valid email.
7. The user clicks on forgot password link to change the password.
8. The application checks the link for validation.
 - 8.1. If the link is valid, then redirect to the password change page.
 - 8.2. If the link is not valid due to any reason, then return the error message.
9. Users enter the new password and confirm the new password.

10. Application checks both passwords in real-time to give feedback. After both passwords match then, the submit button will activate to press.
11. The application sends a successful password change message and redirects to the login page.

As you can see, the task flow diagram for the forgot password task is in Figure 79.

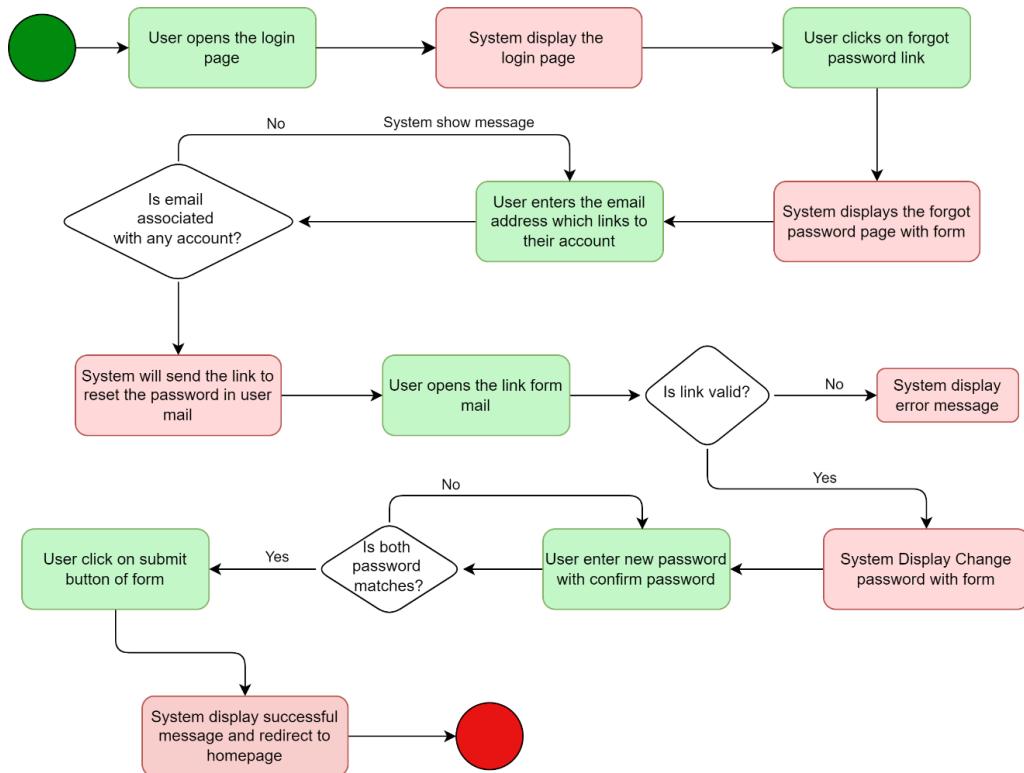


Figure 79: Task flow diagram of “Forgot Password” using draw.io [2][10]

As you can see, the clickstream for the forgot password task is in Figure 80.

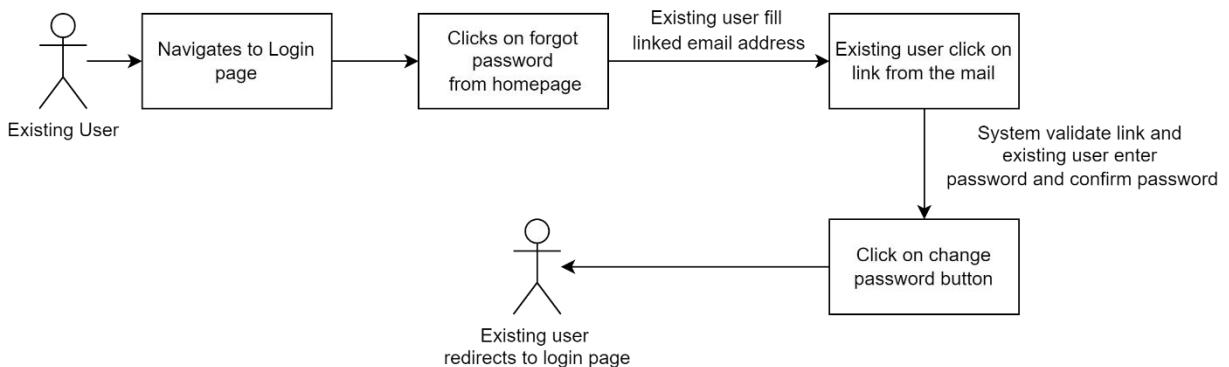


Figure 80: Clickstream of “Forgot Password” using draw.io [2][10]

4.1.2 Membership Management

Task 1: View membership plans

Scenario: A user (potential customer) is looking to enroll himself as a member of the club. The user needs to select an appropriate membership type that suits his requirements and know the monthly installments.

Assumption: The user has successfully signed up and logged in. The user is on the website's landing page.

Use Case:

1. User opens the web application.
2. User clicks on login.
3. System displays a login form.
4. User enters the credential and clicks on Login button.
5. System redirects user to the welcome page of the website.
6. User clicks on the membership tab present in the navigation bar.
7. The system prompts three membership plans:
 - 7.1. Basic membership – It offers cardio room and stretch room access.
 - 7.2. Regular membership – It offers cardio, stretch room, yoga, and Zumba classes.
 - 7.3. Premium membership – It offers cardio, stretch room, yoga, Zumba, swimming, badminton, basketball, table tennis and lawn tennis.
8. User clicks on Get Started button corresponding to a plan to select one of the types of membership.
9. The system redirects the user to the next page of billing information.

Task flow diagram: Figure shows task flow diagram to view the membership plans.

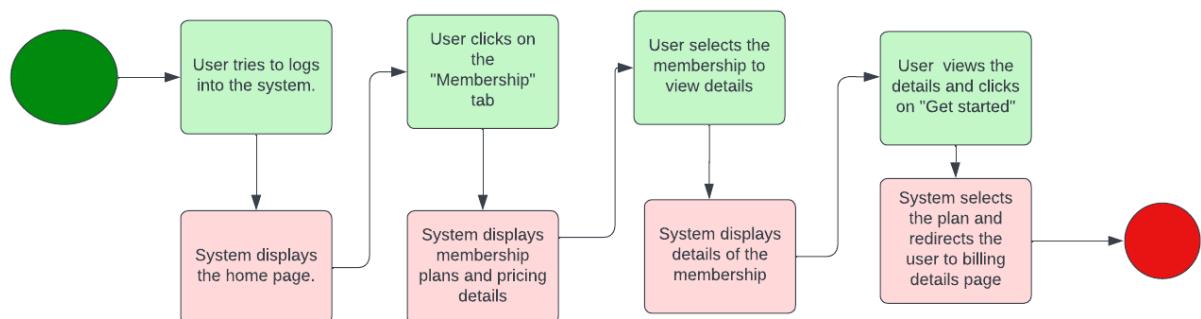


Figure 81: Task flow diagram of "Membership plan details and pricing" using draw.io
[2][11]

Clickstream: Figure shows clickstream diagram to view the membership plans.

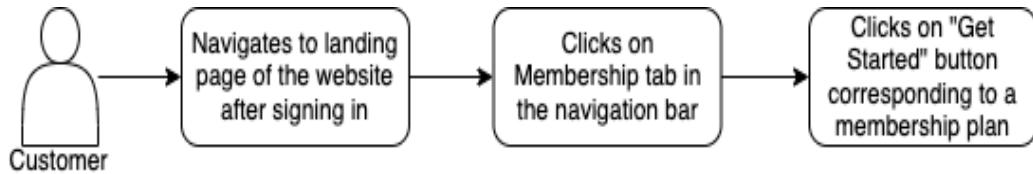


Figure 82: Click stream for "Membership plan details and pricing" [2] [11]

Task 2: Display the pre-saved billing information. Enter/edit the details.

Scenario: A customer “Gary Marin” is looking to enroll himself as a member of the Spotify club. He clicks on the membership page of the Spotify website to view the available membership plans. He selects a membership plan and clicks on “Get started” to proceed to purchase. The system redirects the user to a page to fill in the details about the billing of the purchase. After filling up the form, he clicks on “Next” button.

Use Cases:

1. User opens the web application.
2. User clicks on login.
3. System displays a login form.
4. User enters the credential and clicks on Login button.
5. System redirects user to the welcome page of the website.
6. User clicks on the membership tab present in the navigation bar.
7. The system prompts three membership plans:
 - 7.1. Basic membership – It offers cardio room and stretch room access.
 - 7.2. Regular membership – It offers cardio, stretch room, yoga, and Zumba classes.
 - 7.3. Premium membership – It offers cardio, stretch room, yoga, Zumba, swimming, badminton, basketball, table tennis and lawn tennis.
8. The user selects a plan they are interested in purchasing.
9. The system loads more information on the membership plan and displays a form to fill out the necessary billing details.
10. User enters first name, last name, and address.
 - 10.1. System displays invalid or missing value for first name, last name, or address.
 - 10.2. User enters the missing values or types in values in the correct format.
11. User clicks on Next button.
12. System redirects the user to the next page to review purchase details.

Task flow diagram: Figure shows task flow diagram to enter/edit billing information.

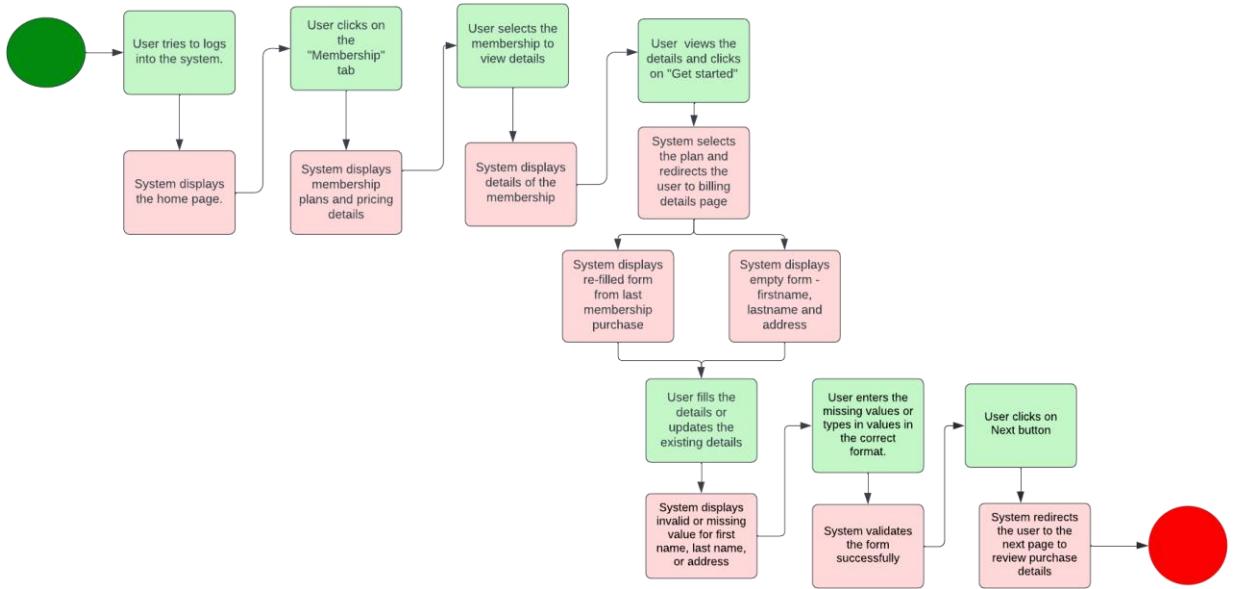


Figure 83: Task flow diagram of "Enter/edit billing information" using draw.io [2] [11]

Clickstream: Figure shows clickstream diagram to enter/edit billing information.

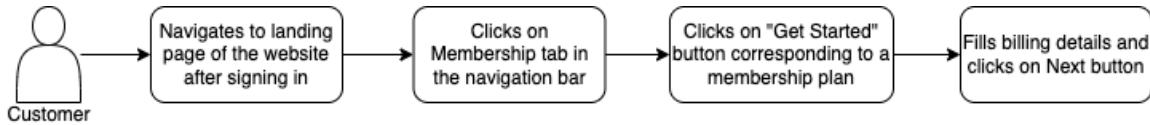


Figure 84: Clickstreams for "Enter/edit billing information" [2] [11]

Task 3: Review purchase details, pick the start and end date and proceed to payment.

Scenario: A customer “Gary Marin” is looking to enroll himself as a member of the Spotify club. He clicks on the membership page of the Spotify website to view the available membership plans. He selects a membership plan and clicks on “Get started” to proceed to purchase. The system redirects the user to a page to fill in the details about the billing of the purchase. After filling up the form, he clicks on “Next” button. System redirects the user to review page. User reviews the details, enters start and end date, and clicks on “Proceed to pay”.

Use case [1]:

1. User opens the web application.
2. User clicks on login.
3. System displays a login form.
4. User enters the credential and clicks on Login button.
5. System redirects user to the welcome page of the website.
6. User clicks on the membership tab present in the navigation bar.

7. The system prompts three membership plans:
 - 7.1. Basic membership – It offers cardio room and stretch room access.
 - 7.2. Regular membership – It offers cardio, stretch room, yoga, and Zumba classes.
 - 7.3. Premium membership – It offers cardio, stretch room, yoga, Zumba, swimming, badminton, basketball, table tennis and lawn tennis.
8. The user selects a plan they are interested in purchasing.
9. The system loads more information on the membership plan and displays a form to fill out the necessary billing details.
10. User enters first name, last name, and address.
 - 10.1. System displays invalid or missing value for first name, last name, or address.
 - 10.2. User enters the missing values or types in values in the correct format.
11. User clicks on Next button.
12. System redirects the user to the next page to review purchase details.
13. User reviews the purchase details, enters the start and end date, and clicks on Proceed to pay.
 - 13.1. System displays invalid or missing value for start and end date.
 - 13.2. User enters the missing values or types in values in the correct format.
14. User clicks on Proceed to pay button.

Task flow diagram: Figure shows task flow diagram to review membership and billing details.

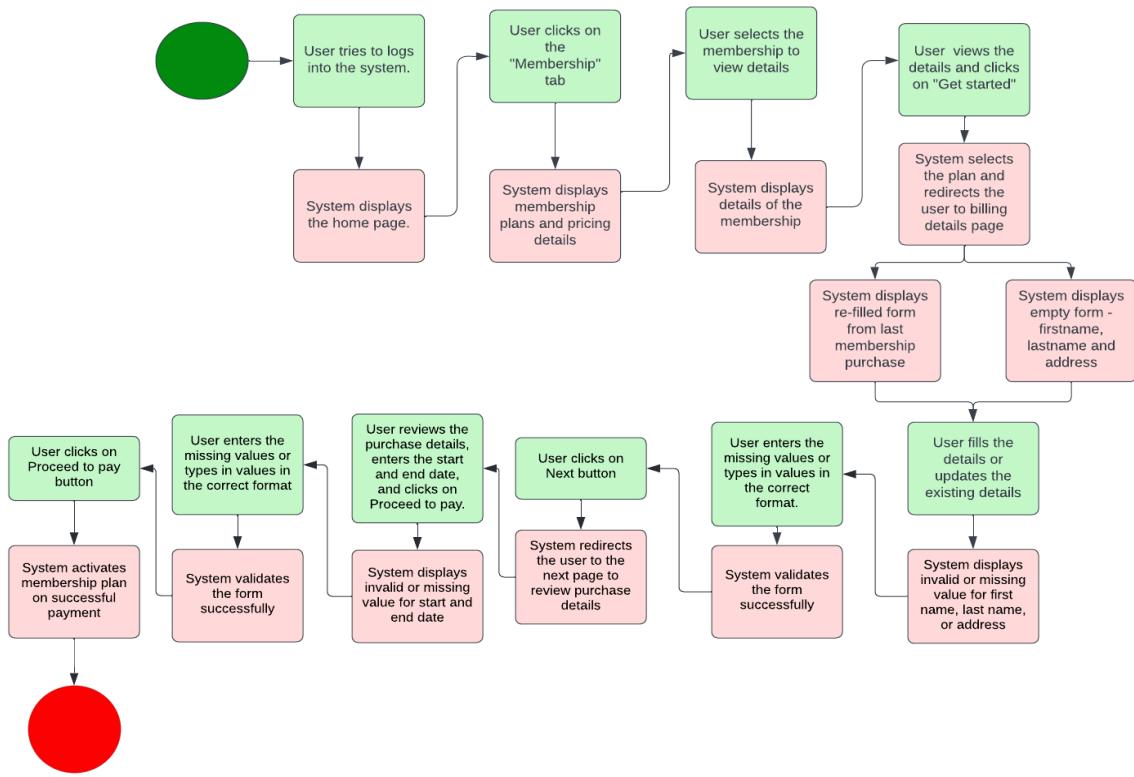


Figure 85: Task flow diagram of "Review membership purchase and payment" [1] [11]

Clickstream: Figure shows clickstream diagram to review membership and billing details.

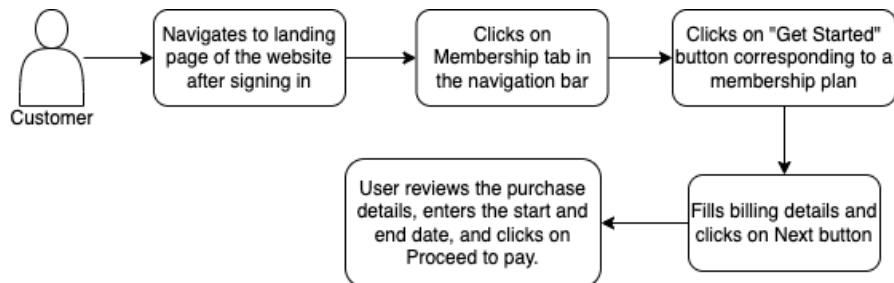


Figure 86: Clickstreams for "Review membership purchase and payment" [2] [11]

Task 4: Cancel membership

Scenario: A customer “Gary Martin” purchased a membership he no longer needs. He is looking to cancel his membership.

Use case [1]:

1. User opens the web application.
2. User clicks on login.

3. System displays a login form.
4. User enters the credential and clicks on Login button.
5. System redirects user to the welcome page of the website.
6. User clicks on the membership tab present in the navigation bar.
7. The system displays purchased membership plan, start date and end date.
8. User clicks on “Cancel Membership” button.
9. System updated the End date automatically to the last day of current month.
10. System cancels the membership plan from next month onwards.
11. The system notifies the user that the cancellation is processed, and they will receive due amount in the next 7 business days.

Task flow diagram: Figure shows task flow diagram to cancel membership.

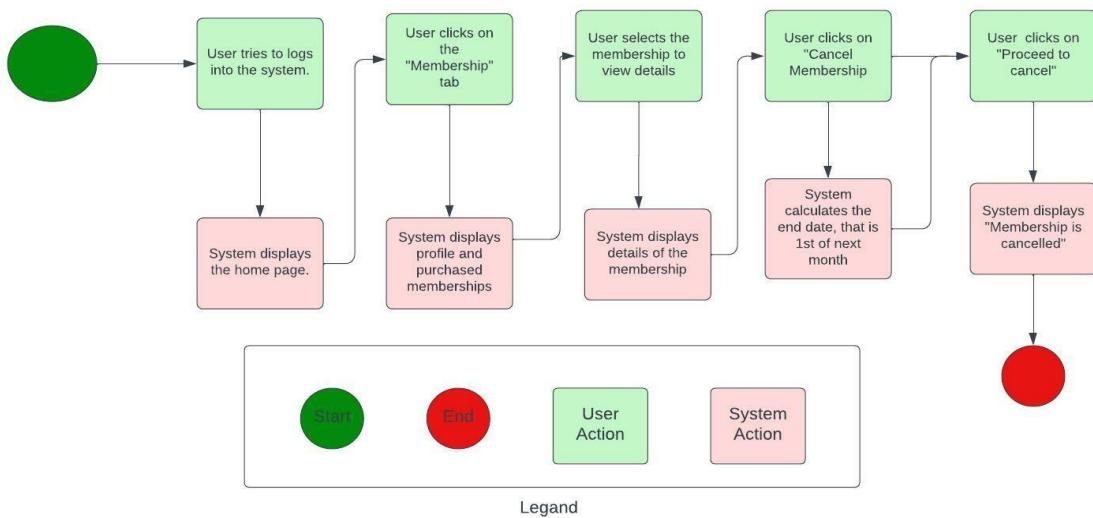


Figure 87: Task flow diagram of "Cancel membership" [1] [11]

Clickstream: Figure shows clickstream diagram to cancel membership.

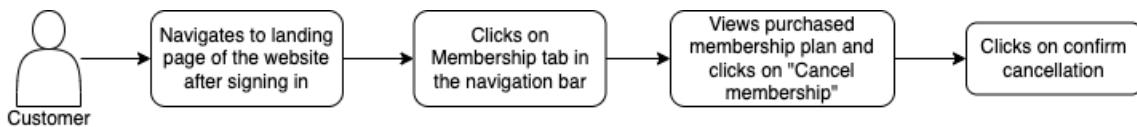


Figure 88: Clickstreams for "Cancel membership" [2] [11]

4.1.3 Facility Management

Task 1: View facility reservations:

Scenario: Bob is a planned individual; before closing for the day, he likes to check his tomorrow's workout schedule so that he could book the equipment if he forgot before. He goes to the website and then to the “my reservations” page to check his reservations.

Use Case:

1. User logs into the system.
2. System displays the home page and navigation buttons.
3. User clicks on the “profile” icon on the top-right of the page.
4. User clicks on the “My reservations” button.
5. System prompts with the list of reservations the user made.
6. User navigates through the page and selects the relevant reservation for viewing details.
7. System displays the reservation details like reservation date, date at which the reservation is made, reserved by, reserved for etc.

Figure 89 shows the task flow diagram to view facility reservations.

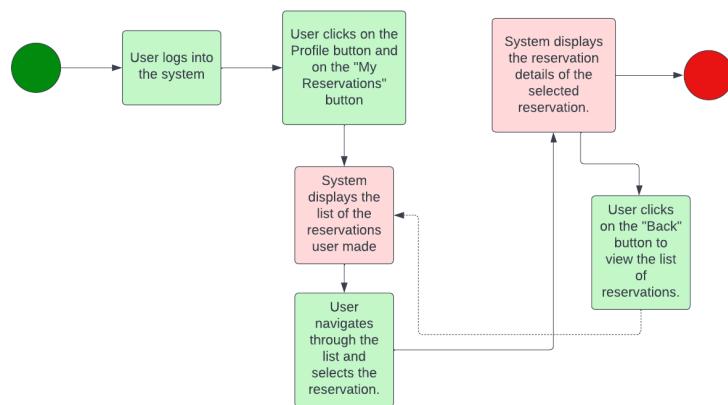


Figure 89: Task flow diagram of view reservation task [2][12].

Figure 90 shows the click stream diagram for viewing reservations.

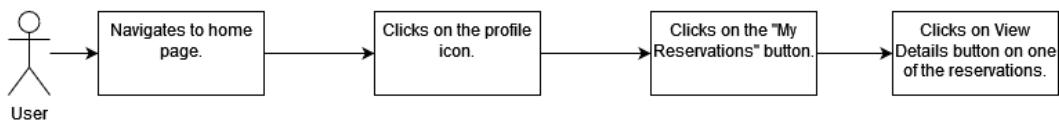


Figure 90: Click Stream for viewing reservations [2][12].

Task 2: Cancel facility reservation:

Scenario: Due to a sudden client meeting, Bob cannot attend his daily workout session and wants to cancel his booking so that other club members can use the equipment. He goes to the website and selects the reservation he intends to cancel.

Use Case:

1. User logs into the website.
2. System displays the home page and navigation buttons.
3. User clicks on the “profile” icon on the top-right of the page.
4. User clicks on the “My reservation” button.
5. System prompts the list of upcoming reservations the user booked.

6. User selects the booking that the user wants to cancel.
7. System displays the booking details along with the “cancel” button.
8. User clicks on the “cancel” button to cancel the selected booking.
9. System asks the user for confirmation. To make sure that the user is not mistakenly cancelling the booking.
 - 9.1. The user clicks on the “No” button.
 - 9.2. System clears the pop-up and displays the reservation details.
10. The user clicks on the “yes” button.
11. The system updates the reservation status from ‘Active’ to ‘Cancelled’.
12. The system displays to the user that reservation was successfully cancelled.
13. The system redirects the user to the “My Reservations” page.

Figure 91 shows the task flow diagram for cancelling a facility reservation.

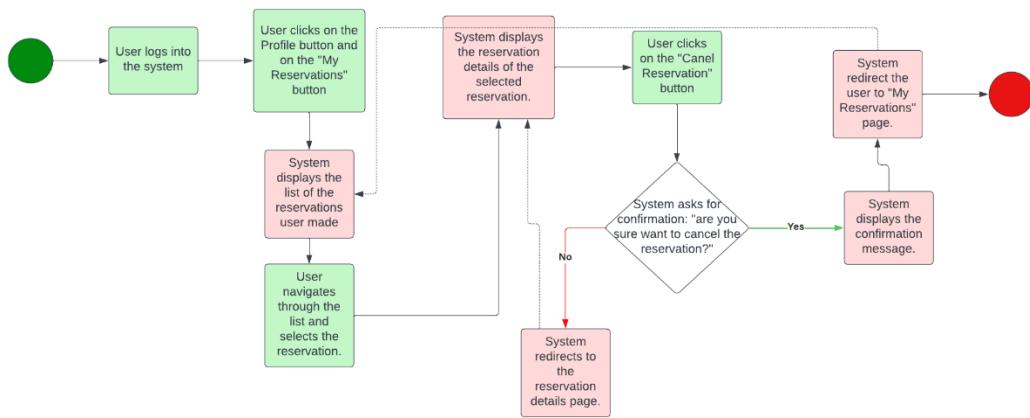


Figure 91: Task flow diagram for cancel reservation task [2][12].

Figure 92 shows the click stream diagram for cancelling a reservation.

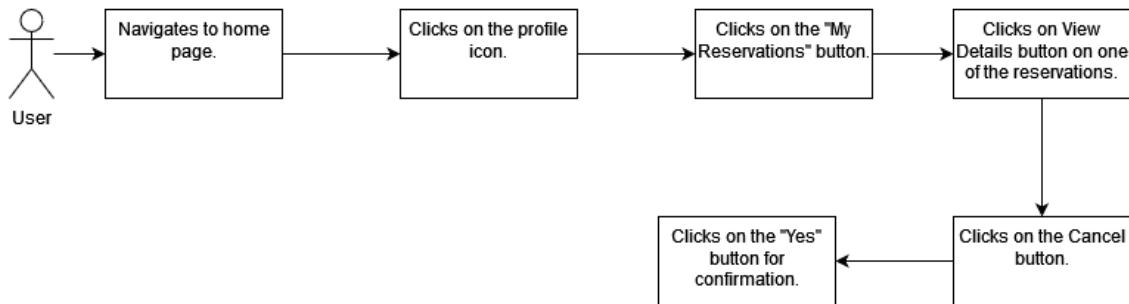


Figure 92: Click stream for cancelling a reservation [2][12].

Task 3: Reserve facility:

Scenario: To save time, Bob wants to plan his workout sessions and reserve the equipment needed for his workout. He goes to the website and chooses equipment and the available timeslot for his training to book the equipment.

Use Case:

1. User logs into the system.
2. System prompts the list of equipment available for booking.
3. User selects the equipment for tomorrow's workout.
4. System displays the equipment details like equipment name, location, available time slots for booking etc.
5. User checks the availability prompted by the system.
 - 5.1. If the user can't find a suitable timeslot for the booking and clicks on the back button.
 - 5.2. System redirects the user to the equipment list page.
6. User chooses an available timeslot for the booking.
7. System prompts the user with all the booking details like equipment name, timeslot, the user's name etc.
8. User clicks on the 'confirm' button.
9. System prompts the confirmation message that the booking has been made and redirects to the equipment list page.

Figure 93 shows the task flow diagram for reserving club's facility.

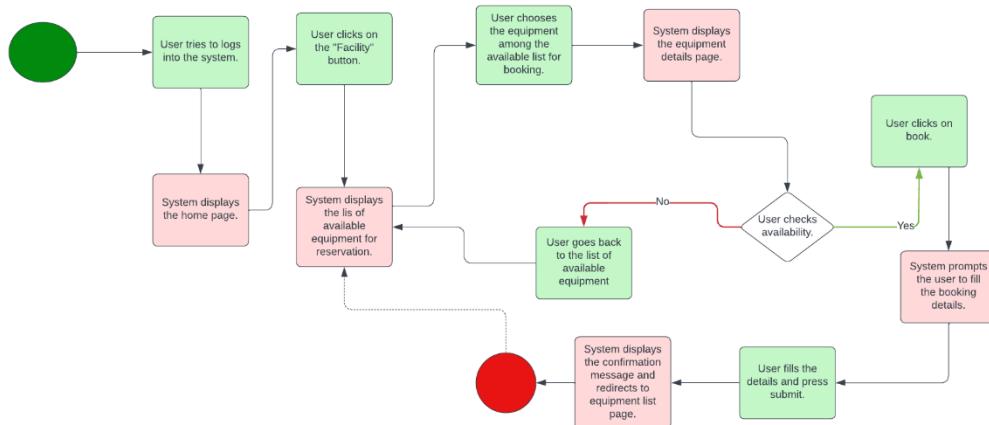


Figure 93: Task flow diagram for reserving club's facility [2][12].

Figure 94 shows the click stream diagram for facility reservation task.

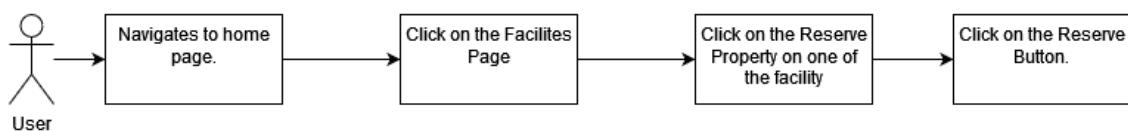


Figure 94: Click stream for the facility reservation task [2][12].

Task 4: Add new facility:

Scenario: Alice is an admin of the club and is responsible for managing the equipment in the club. Alice's responsibility is to make it available for the club members to book whenever the club purchases new equipment.

Use Case:

1. User logs into the system as admin.
2. System displays the home page.
3. User clicks on the "Facility" button.
4. System prompts with the list of the available equipment and a "Add new" button.
5. User clicks on the "Add new" button.
6. System asks the user to enter the new equipment details.
7. User adds the details of the new equipment like name, image, type, available from etc., and clicks submit button.
8. System sends the user's input to the backend processing and prompts the user with the confirmation that the new equipment is added to the available list of equipment.
9. System redirects the user to the page with the list of available equipment.

Figure 95 shows the task flow diagram for adding new facility for club members.

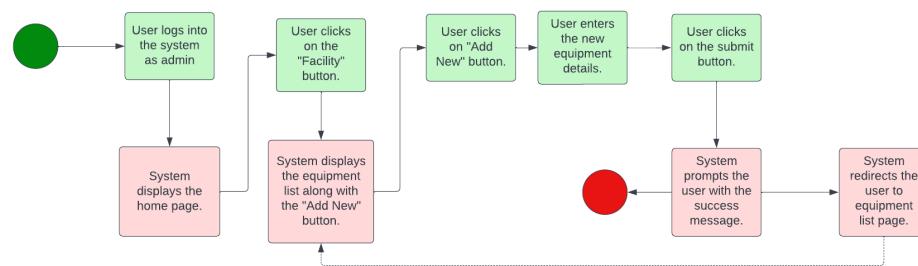


Figure 95: Task flow diagram for adding new facility [2][12].

Figure 96 shows the click stream diagram for adding new facility into the system.

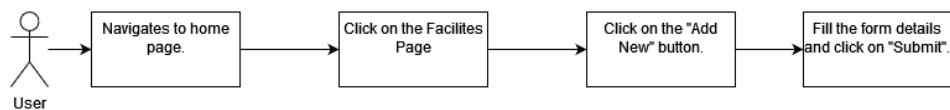


Figure 96: Click Stream for the add new facility task [2][12].

4.1.4 Event Management

Task 1: Register/Book for an event

Scenario: Brian is a club member who wants to register along with his four family members for the next available yoga event in the club from the club's website.

Use Case:

1. The user enters an email/username and password to log in to the system.
2. The system displays the home page.
3. The user clicks on the “Events” button.
4. The system redirects the user to the events list page.
5. The user navigates through the events and clicks on the interested event for registration.
6. The system redirects the user to the event’s details page.
7. The user checks the event details like event date, time, location, etc.
 - 7.1. The user clicks on the “Back” button.
 - 7.2. The system redirects the user to the events list page.
8. The user fills the registration form with appropriate details.
 - 8.1. The user clicks on the “Cancel” button.
 - 8.2. System pops up an alert asking the user, “Are you sure you want to cancel the update?” with “yes” and “No” buttons.
 - 8.2.1. The user clicks on the “No” button.
 - 8.2.2. System clears the pop-up and displays the registration form.
 - 8.3. The user clicks on the “Yes” button.
 - 8.4. System redirects the user to the event details page.
9. The user clicks on the “Register” button.
10. The system validates the user details.
 - 10.1. The system displays an error message because of invalid details in the form.
 - 10.2. The user updates the details and clicks the “Submit” button.
11. The system sends the request to the backend service for further processing and displays a success message to the user.
12. The system redirects the user to the events list page.

Figure 97 shows the task flow diagram for registering to an event.

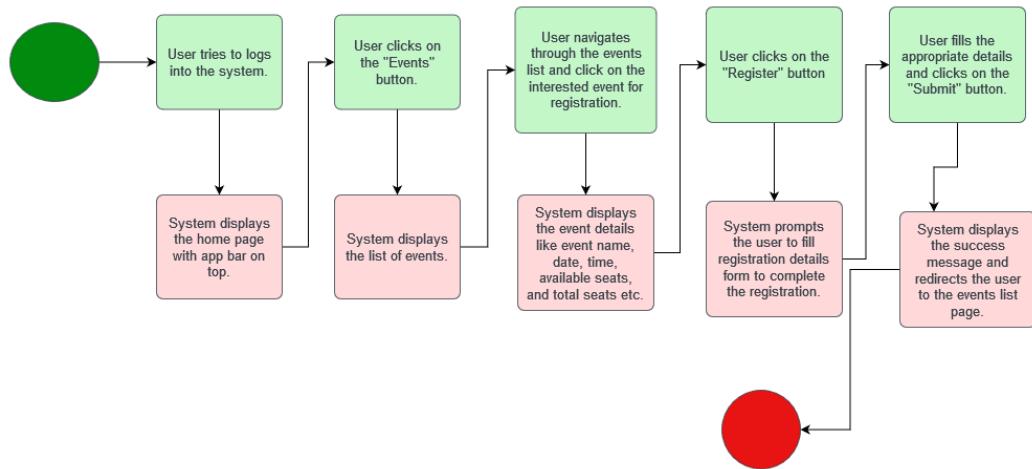


Figure 97: Task flow diagram for registering to an event [2][12].

Figure 98 shows the click stream diagram for registering to an event.

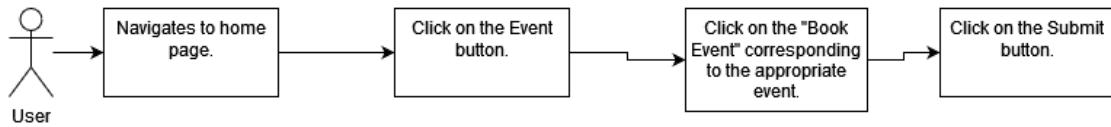


Figure 98: Click stream diagram for registering to an event [2][12].

Task 2: Cancel registration/booking for an event

Scenario: Brian is a club member who wants to cancel the booking of an upcoming event due to an unavoidable personal appointment scheduled at the same time.

Use Case:

1. The user enters an email/username and password to log in to the system.
2. The system displays the home page.
3. The user clicks on the profile page.
4. The system displays the drop-down menu for the user to select the appropriate page.
5. The user clicks on the “My Events” button.
6. The system redirects the user to the list of events page.
7. The user navigates through the events list that they registered.
8. The user selects the appropriate event.
9. The system displays the event details along with registration details.
10. The user clicks on the “Cancel Registration” button.
11. The system prompts the user with a pop-up; “Are you sure you want to cancel the update?” with “yes” and “No” buttons.
 - 11.1. The user clicks on the “No” button.
 - 11.2. System clears the pop-up and displays the registration details page.
12. The user clicks on the “yes” button.
13. The system sends the user’s request to the backend services for further processing and prompts with a success message that the registration is cancelled.
14. The system redirects the user to the “My Events” page.

Figure 99 shows the task flow diagram for cancelling a registration for an event.

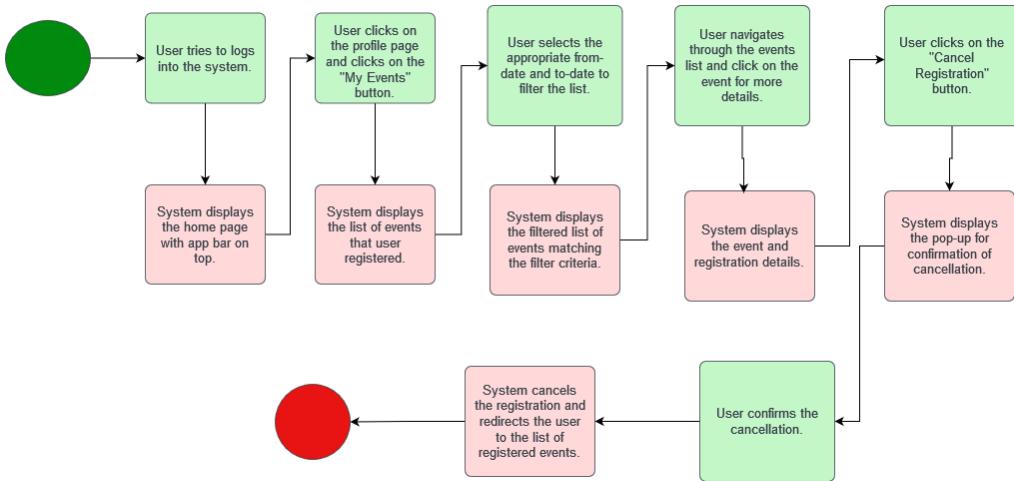


Figure 99: Task flow for cancelling registration for an event [2][12].

Figure 100 shows the click stream diagram for cancelling registration for an event.

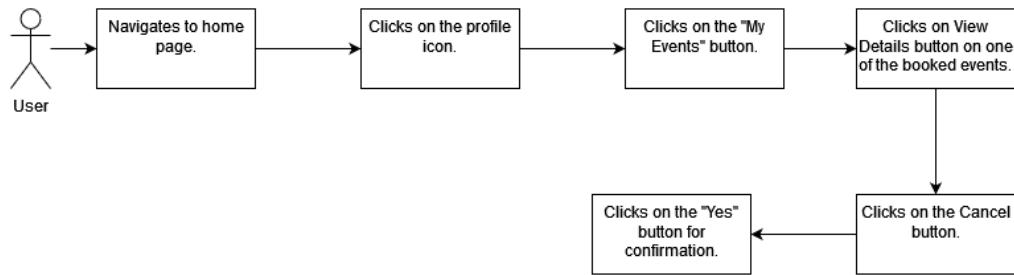


Figure 100: Click stream for cancelling a registration for an event [2][12].

Task 3: Add new event

Scenario: Henry is an admin for the sports club. As a part of his responsibility, he needs to create new events hosted in the club. The events generally have a participant's limit.

Use Case:

1. The user enters a username/email and password to log in as an admin.
2. System displays the home page along with the Event Management button.
3. The user clicks on the "Event Management" page.
4. System displays the list of events along with a "Add new" button.
5. The user clicks on the "Add new" button.
6. System displays a form with necessary details to fill for the new event the user wants to add.
7. The user fills in all the relevant information about the new event.
 - 7.1. The user clicks the "back" button.
 - 7.2. System pops up an alert asking the user, "Are you sure you want to cancel the update?" with "yes" and "No" buttons.
 - 7.2.1. The user clicks on the "No" button.

- 7.2.2. System clears the pop-up and displays the edit event page.
- 7.3. The user clicks on the “Yes” button.
- 7.4. System redirects back to the listing page.
8. The user clicks the “Submit” button.
- 8.1. System displays an error message because of the invalid information provided by the user.
- 8.2. The user updates the invalid information and clicks the “Submit” button.
9. System sends the event details to the backend service for further processing and displays a success message to the user.
10. System redirects back to the event listing page.

Figure 101 shows the task flow diagram for creating a new event.

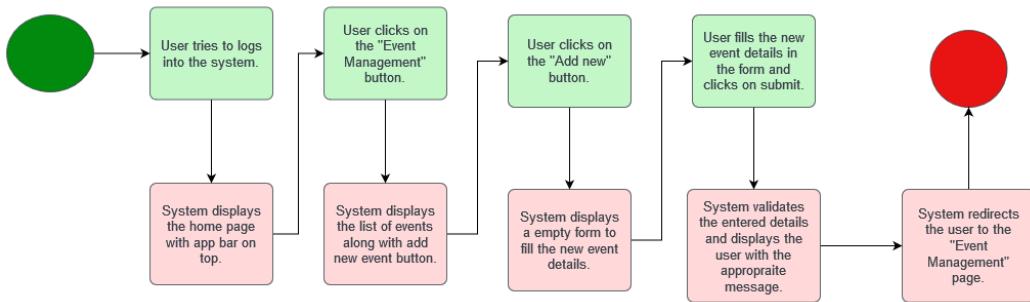


Figure 101: Task flow for creating new event [2][12].

Figure 102 shows the click stream diagram for creating new event.

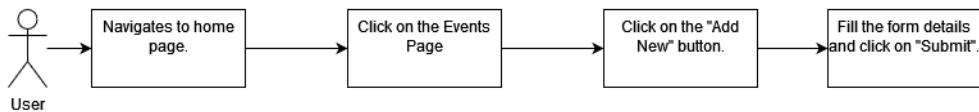


Figure 102: Click stream for adding new events [2][12].

4.1.5 Payment Management

Task 1: Establishing the payment system

Scenario: Dan, a customer is looking to enroll himself as a member of the club. He has created and selected his profile. He has selected the type of membership and knows the amount to pay. Now he must make the payment.

Use Case:

1. User opens the web application.
2. System displays the login page.
3. User clicks on login using their credentials.
4. System displays the homepage.
5. User clicks their profile.

6. System displays the user profile page.
7. User selects the membership tab.
8. System displays the various types of membership.
9. User selects membership type and views the amount to be paid and clicks on Proceed to Pay.
10. System redirects to the stripe payment gateway.
 - 10.1. On successful payment, the membership is sold successfully.
 - 10.2. If payment fails, shows payment failed message and redirects to step 9

Task Flow: The **Figure 103** depicts the task flow for the payment page. It will be redirected to the stripe gateway. Validation of credit cards are taken care by the Stripe gateway.

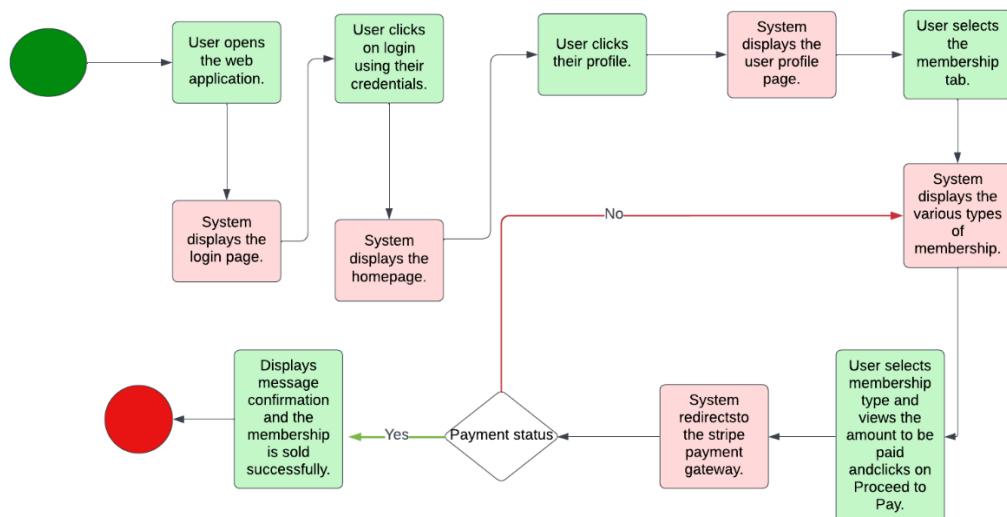


Figure 103: Task Flow diagram for Establishing the payment system using lucid chart
[1][16]

Click Stream: The click stream for the payment page is displayed in the **Figure 104**. It starts right from the start where the website loads.

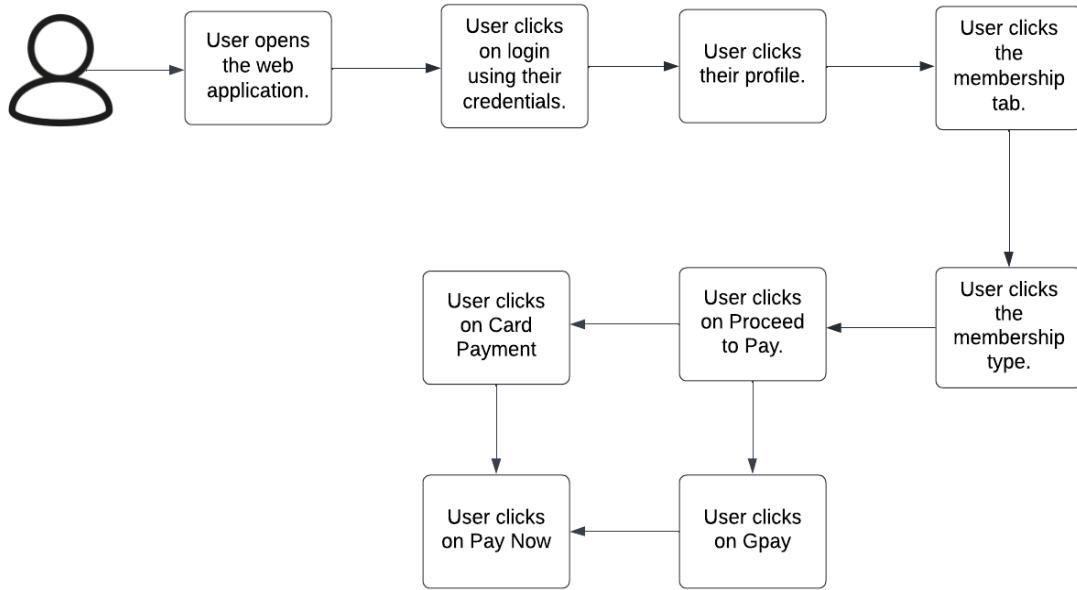


Figure 104: Click Stream for payment feature using draw.io [2][13]

4.1.6 Merchandise Product Management

Task 1: Display the products in the merchandise page

Scenario: Emily is a new club member who wants to purchase merchandise from the club's merchandise. She goes to the website to look at the various selections available.

Use Case:

1. User opens the web application.
2. System displays the login page.
3. User clicks on login using their credentials.
4. System displays the homepage.
5. User clicks on the merchandise tab.
6. System displays the products in the merchandise page.

Task Flow: The **Figure 105** displays the task flow diagram for display the products in the merchandise page

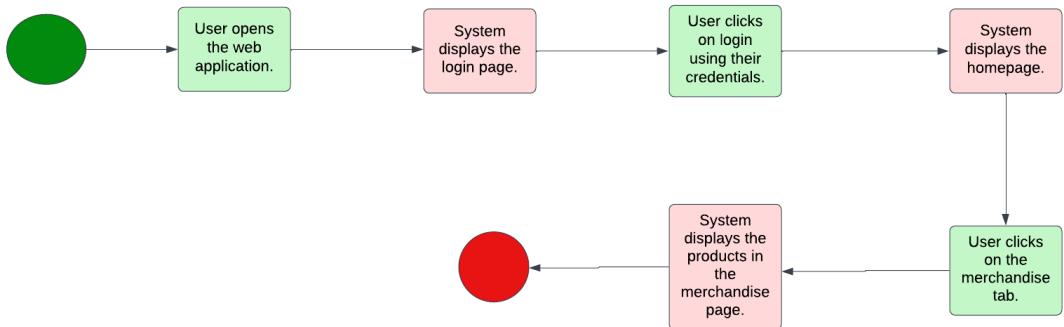


Figure 105: Task Flow diagram for display the products in the merchandise page using lucid chart [1][16]

Click Stream: The **Figure 106** displays the click stream for the merchandise page.

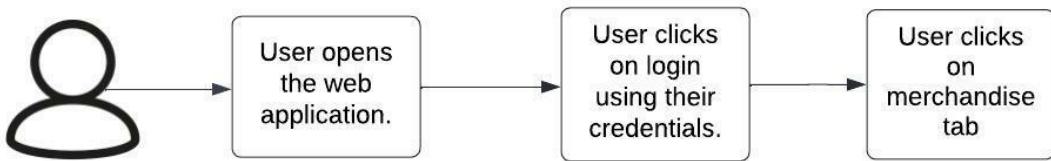


Figure 106: Click Stream for the merchandise page using draw.io [2][13]

Task 2: Add product to the Merchandise List

Scenario: Dan is the sports club's administrator. He wants to add the newly launched Hoodies that are available in store to the website in order to keep it up to date with what is available in store.

Use Case:

1. Admin opens the web application.
2. System displays the login page.
3. Admin clicks on login using their credentials.
4. System displays the homepage.
5. Admin clicks on the merchandise tab.
6. System displays the products in the merchandise page.
7. Admin clicks on the add product button
8. System displays the add product form for the user to fill in.
9. Admin fills the form and uploads the image as a requirement.
10. Admin clicks on add post button.
11. System adds the product to the merchandise list and gets redirected to the display merchandise page with the newly updated list.

Task Flow: The displays **Figure 107** task flow diagram for the add product to the merchandise page

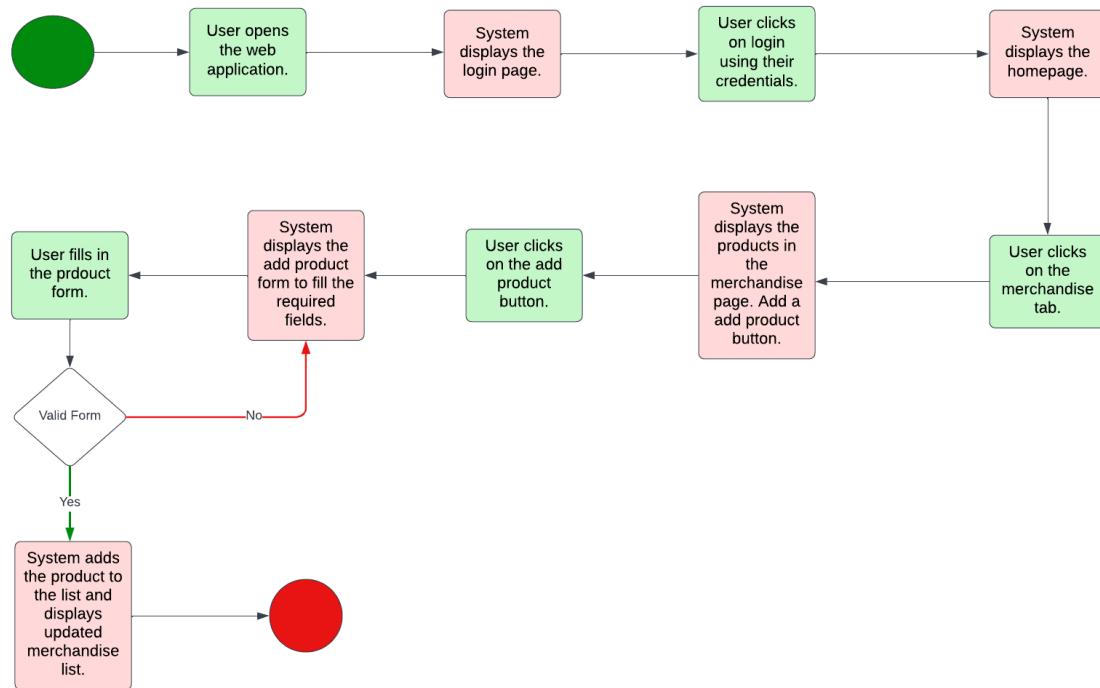


Figure 107: Task Flow diagram for the add product to the merchandise page using lucid chart [1][16]

Click Stream: The **Figure 108** displays the click Stream for add product to the merchandise page

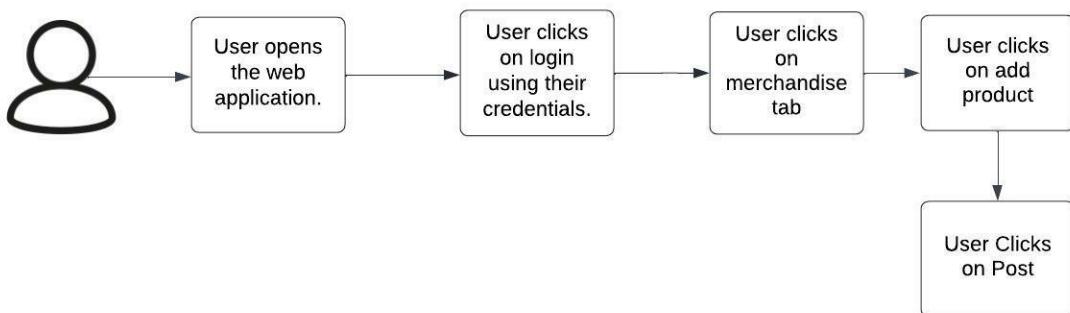


Figure 108: Click Stream for add product to the merchandise page using draw.io [2][13]

Task 3: View details of a product

Scenario: Dan, the administrator of the sports club, wants to know whether he can keep the product on the website or remove it. He decides to start by looking at the product's specifications.

Use case:

1. Admin opens the web application.
2. System displays the login page.
3. Admin clicks on login using their credentials.
4. System displays the homepage.
5. Admin clicks on the merchandise tab.
6. System displays the products in the merchandise page.
7. Admin clicks on the view product button of the product he wants to see.
8. System displays the details of that product with an option to delete it.

Task Flow: The **Figure 109** displays the task flow diagram for view details of a particular product

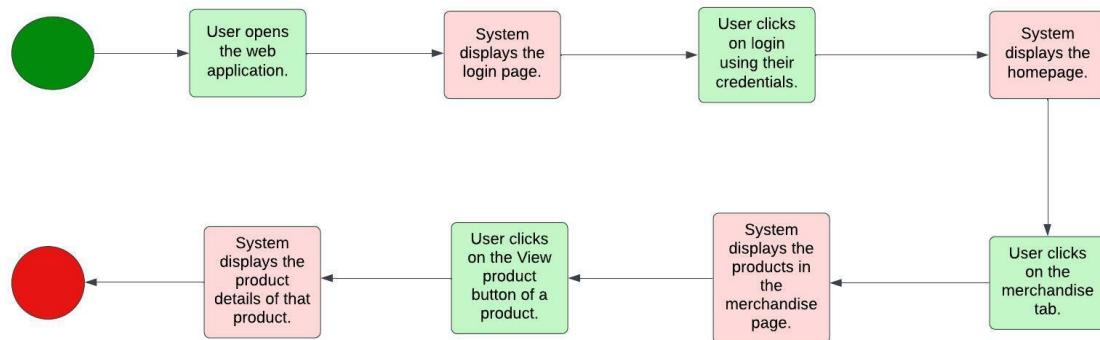


Figure 109: Task Flow diagram for view details of a particular product using lucid chart
[1][16]

Click Stream: The **Figure 110** displays the click stream for the view details of the product



Figure 110: Click Stream for the view details of the product using draw.io [2][13]

Task 4: Delete the product from the merchandise list

Scenario: Dan, the administrator of the sports club, wants to know whether he can keep the product on the website or remove it. He decides to start by looking at the product's specifications. On looking its specifications, he decides to remove the product.

Use Case:

1. Admin opens the web application.
2. System displays the login page.
3. Admin clicks on login using their credentials.
4. System displays the homepage.
5. Admin clicks on the merchandise tab.
6. System displays the products in the merchandise page.
7. Admin clicks on the view product button of the product he wants to see.
8. System displays the details of that product with an option to delete it
9. Admin clicks on delete product.
10. System deletes the product from the merchandise and displays the updated merchandise list.

Task Flow: The **Figure 111** displays the task flow diagram for delete product from merchandise list

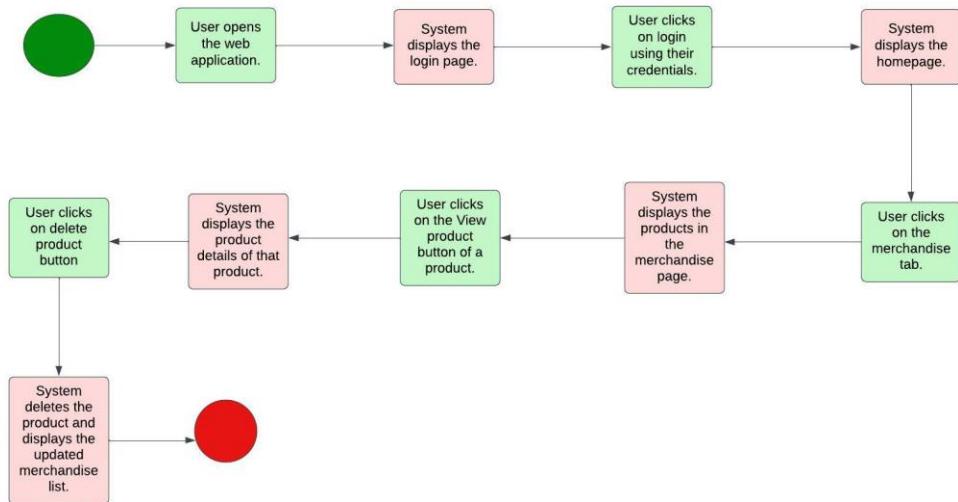


Figure 111: Task flow diagram for delete product from merchandise list using lucid chart [1][16]

Click Stream: The **Figure 112** displays the Click Stream for delete product from the merchandise list.

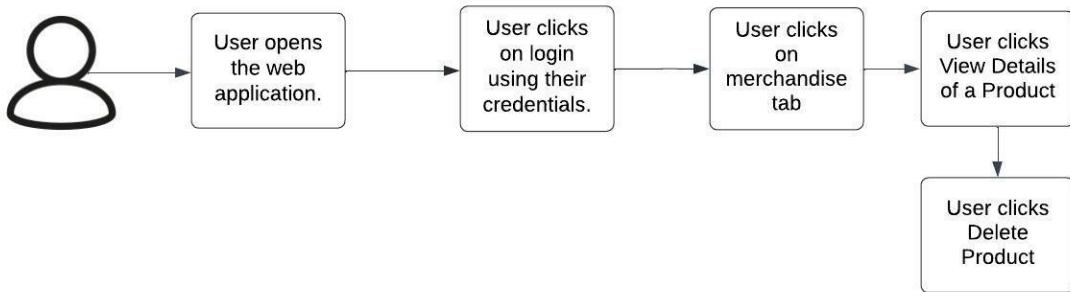


Figure 112: Click Stream for delete product from the merchandise list using draw.io [2][13]

4.1.7 Search Function

Task 1: Search the application facilities, event, merchandise, blogs with search auto completion

Scenario: A member of the club or any other user is searching for the facilities, event etc. the application is providing along with its details

Use Case:

1. User login to the application.
2. Users are searching whether the club is providing the facility.
3. User wants to get some detail on that facility.
4. User wants to visit the page of that facility.
5. Application displays the details of the facility.

Task Flow: The Figure displays the task flow diagram for search function.

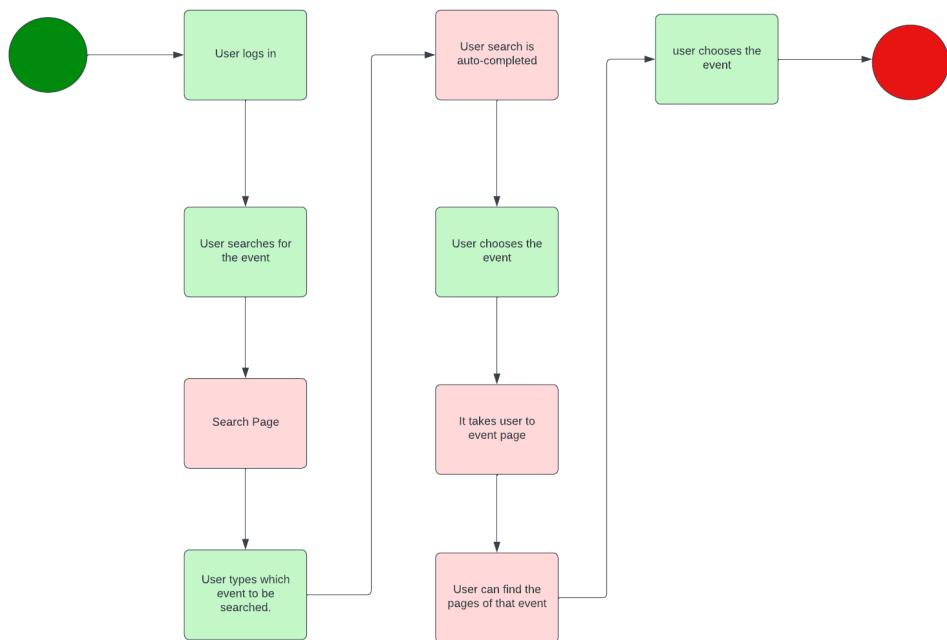


Figure 113: Task Flow diagram for search function using lucid chart [1][16]

The **Figure** below shows the click stream of Search feature where user can search for facilities in Spotify application.

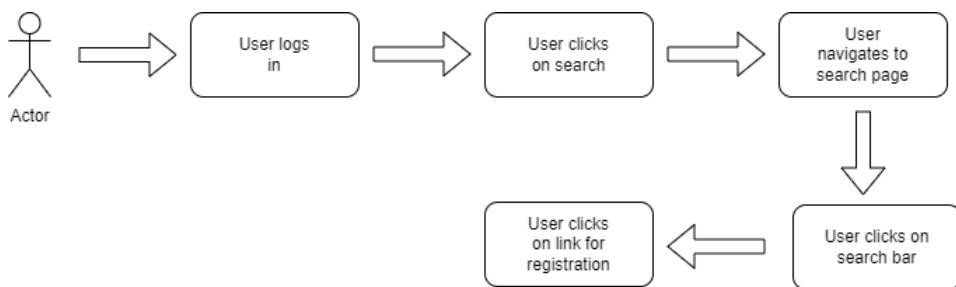


Figure 114: Click stream of search feature using draw.io [2][14]

Task 2: Proceed further with the feature user searched

Scenario: A member of the club is finding difficulty in searching for the navigation pane of registration of the facility, registration of the event, and in searching and viewing blogs.

Use Case:

1. User login to the application.
2. User is searching for the facility, event for the registration.
3. User is searching for the blog to view.
4. User is searching for the navigation pane.
5. The user is searching for the link of the facility registration, link of the event registration and page to view blogs.
6. Search function displays the above all functions...

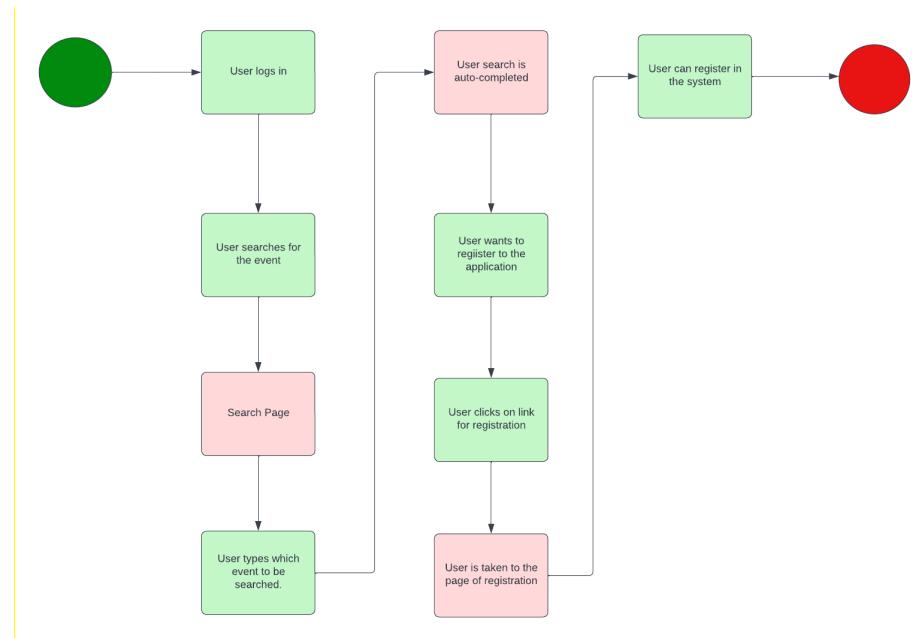


Figure 115: Task Flow diagram for link for registration using lucid chart [1][16]

The **Figure 116** below shows the click stream of link for registration where user can register for facilities in Spotify application. Similarly, user can register for the event and searched event, and view the searched blogs.

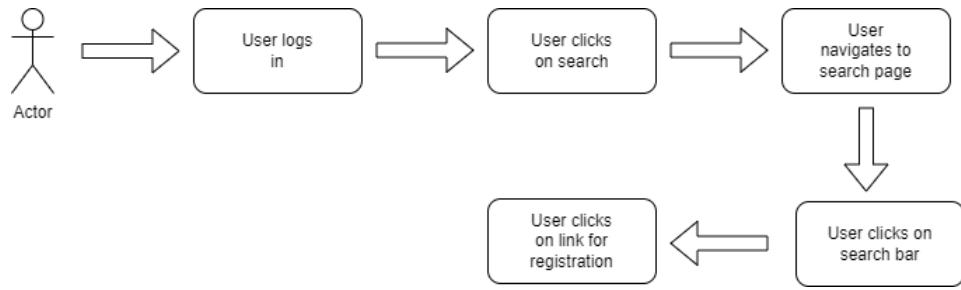


Figure 116: Click stream of link for registration using draw.io [2][14]

4.1.8 Blogging Feature

Task 1: Display All Blogs

Scenario: The user wants to read a blog posted by his gym friend on the blog page called fitness and me.

Use Case:

1. The user opens the home page and logs in using the credentials (if not logged in already.)
2. The user clicks on the blogs icon on the top center of the nav bar.
3. The user is redirected to the blogs homepage.
4. The user types in “Fitness and me” on the search bar present on the nav bar of all blogs page and clicks the search button.
5. The search results display the post under the title “Fitness and me”
6. The user clicks on the post and is redirected to the page which displays the content of the post.

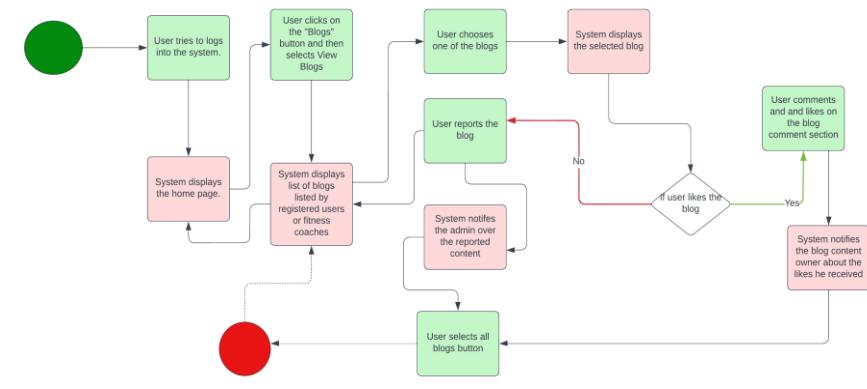


Figure 117: Task Flow diagram for All Blogs based on category using lucid chart [1][16]

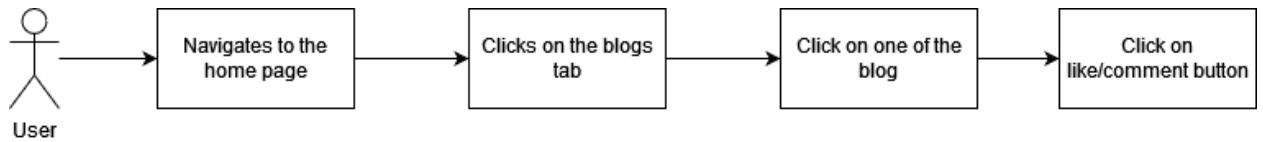


Figure 118: Click streams for all blogs using draw.io [2][15]

Task 2: Create a blog

Scenario: The user logs into the system and wants to create a blog to share his story about his road to fitness.

Use Case:

1. The user opens the home page and logs in using the credentials (if not logged in already.)
2. The user clicks on the blogs icon on the top center of the nav bar.
3. The user is redirected to the blogs homepage.
4. The user clicks on to the create blog icon on the dashboard which is present to the left of all blogs screen.
5. The user is redirected to the create blog page.
6. The user types the title of the blog title prompt.
7. The user types the content of the blog in the text box prompt.
8. The user can upload image from local system to the blog by clicking the upload image button present on the right side of the create blog page.
9. To create the blog, the user clicks on the create blog button on the right side of the create blog page.
10. The user is redirected to all blogs page where the user's blog is displayed on the screen based on the time stamp, he posted the blog. The more recent the time stamp the more top the post will be displayed on the page.

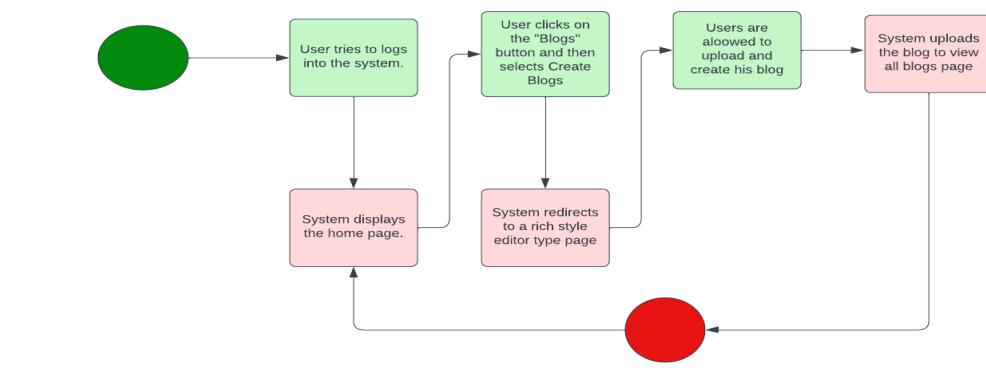


Figure 119: Task Flow diagram for Create Blog using lucid chart [1][16]

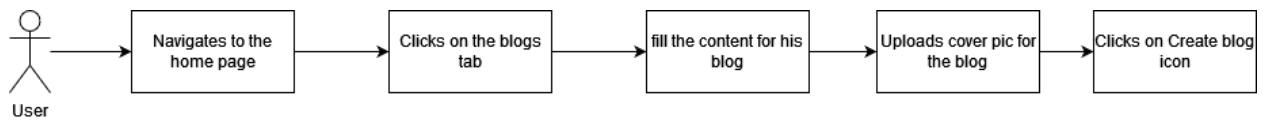


Figure 120: Click Stream for create blog using draw.io [2][15]

Task 3: Edit Blog

Scenario: The user logs into the system and wants to edit his blog he posted on the site

Use Case:

1. The user opens the home page and logs in using the credentials (if not logged in already.)
2. The user clicks on the blogs icon on the top center of the nav bar.
3. The user is redirected to the blogs homepage.
4. The user clicks on to your blogs icon on the dashboard which is present to the left of all blogs screen.
5. The user is redirected to your blogs page.
6. The user can view all the blogs he has written.
7. The user can select the edit button under the blog he wishes to edit.
8. The user is redirected to the edit blog page.
9. The user can edit the blog and click submit button once the edit is completed.
10. The user is redirected to all blogs screen, and he can search for his blog.
11. The user clicks on the blog and can see the edited blog.

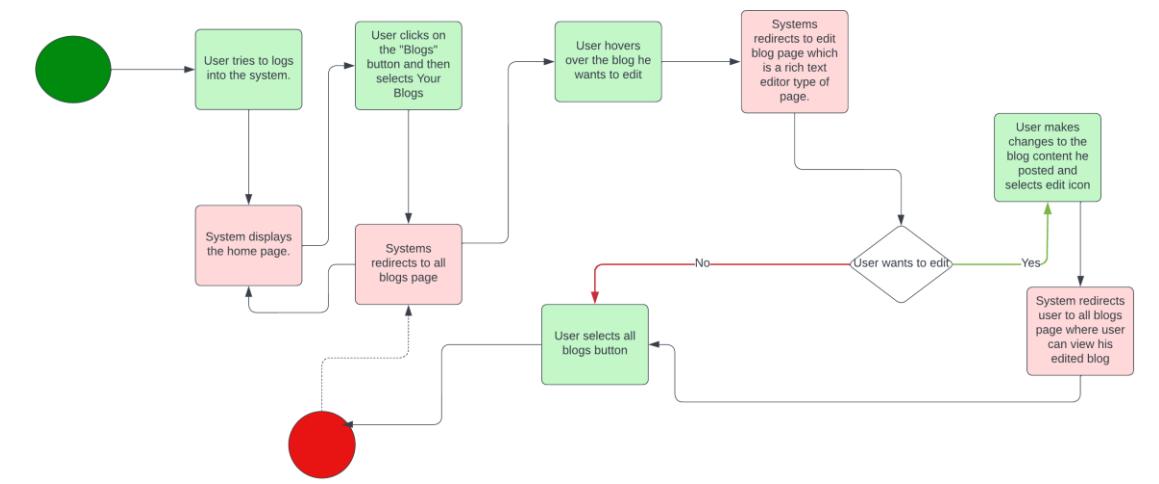


Figure 121: Task Flow diagram for Edit Blog using lucid chart [1][16]

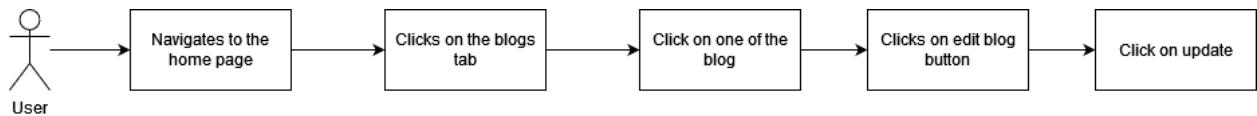


Figure 122: Click Stream for Edit Blog using draw.io [2][15]

Task 4: Delete Blog

Scenario: The user logs into the system and wants to edit his blog he posted on the site

Use Case:

1. The user opens the home page and logs in using the credentials (if not logged in already.)
2. The user clicks on the blogs icon on the top center of the nav bar.
3. The user is redirected to the blogs homepage.
4. The user clicks on to your blogs icon on the dashboard which is present to the left of all blogs screen.
5. The user is redirected to your blogs page.
6. The user can view all the blogs he has written.
7. The user can select the delete button under the blog he wishes to delete.
8. The user is redirected to the confirm delete pop up screen.
9. The user can confirm the delete process by selecting the delete button.

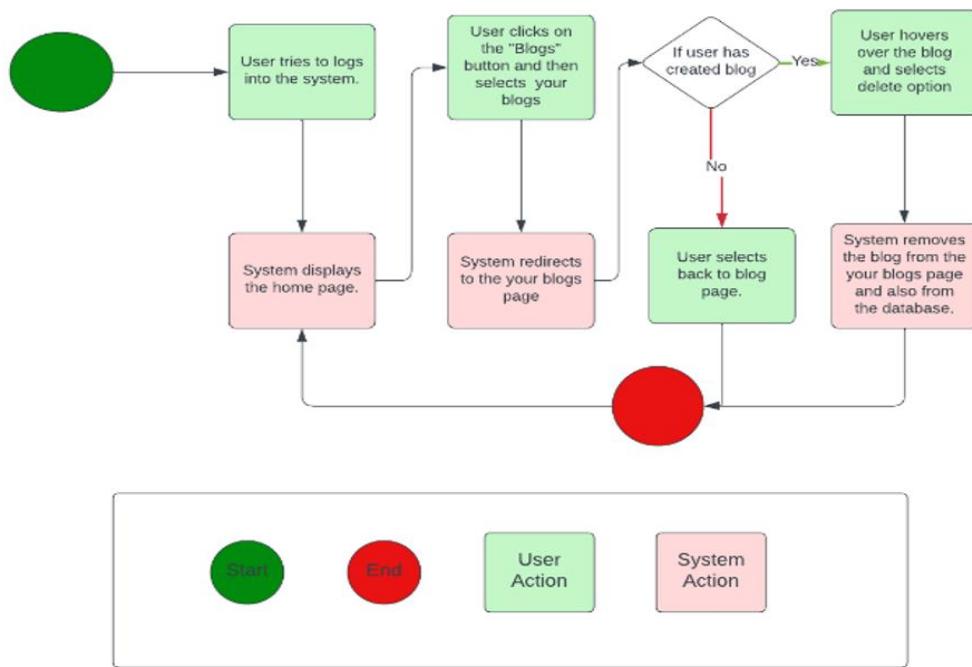


Figure 123: Task Flow for Delete Blog using Lucid chart [1][16]

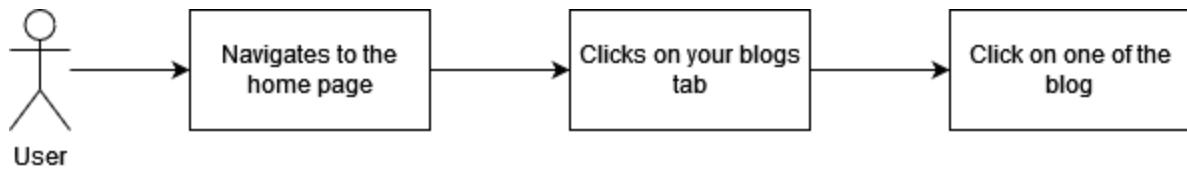


Figure 124: Click stream diagram for delete blog using Lucid chart [1][15]

4.1.9 Rewards System

Task 1: Reward connects with points

Scenario: A regular customer wants some benefits so for that they visit the rewards section where they find points to redeem with some brand's gift card or cash cards.

Use Case:

1. A user visiting any page on the website and hovering on the avatar with their name/image.
2. From the drop-down user selects my rewards.
3. The application displays a rewards page with two tabs
4. The user is on my points tabs by default.
5. The user selects redeem points button
6. The application display modal with a choice of drop-down with brand's
7. The user selects any brand from the drop-down
 - 7.1. If the user clicks on close in modal, then stays on the same page.
8. The user clicks on submit button.
9. The application will send the notification to the user that you successfully redeemed your points and that the gift card was sent to you in the mail.

As you can see, the task flow diagram for the rewards connected with points task is in **Figure 125**

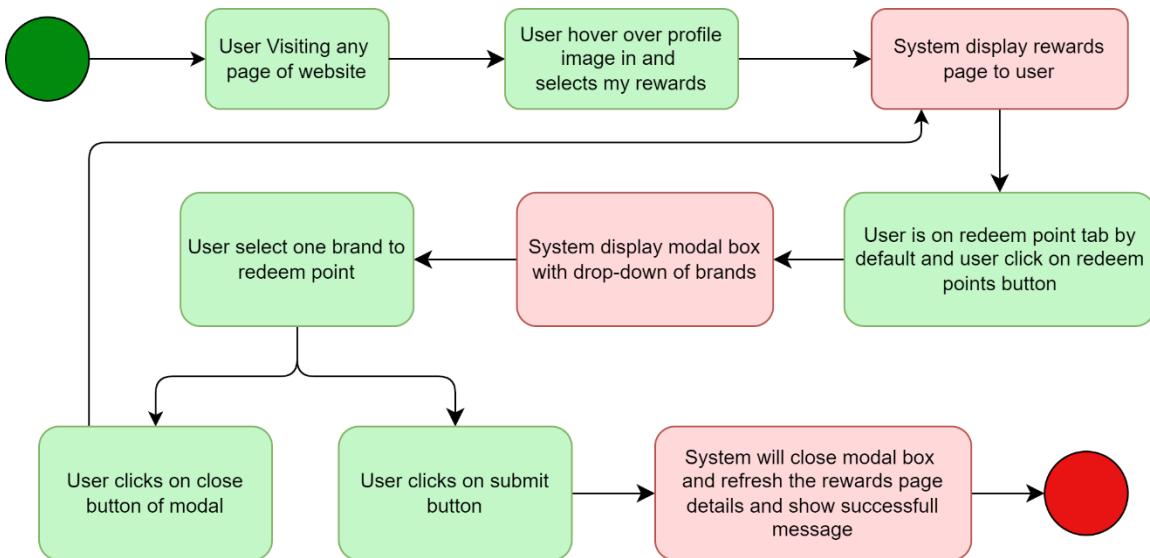


Figure 125: Task flow diagram of "Rewards connected with points" using draw.io [1][16]

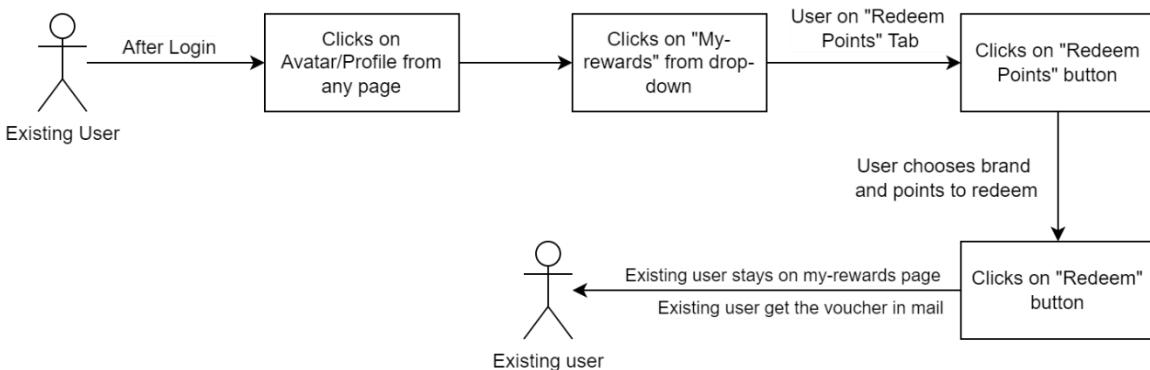


Figure 126: Clickstream of "Rewards connected with points" using draw.io [2][10]

Task 2: Different coupon codes for some brands:

Scenario: A regular customer wants some benefits so for that they visit the rewards section where they find points to redeem with some brand's gift card or cash cards.

User Case

1. A user visiting any page on the website and hovering on the avatar with their name/image.
2. From the drop-down user selects my rewards.
3. The application displays a rewards page with two tabs
4. The user clicks on my coupons.
5. The application shows different companies' coupons with codes and details
6. The user copied the code

As you can see, the task flow diagram for the signup task is in **Figure 127**

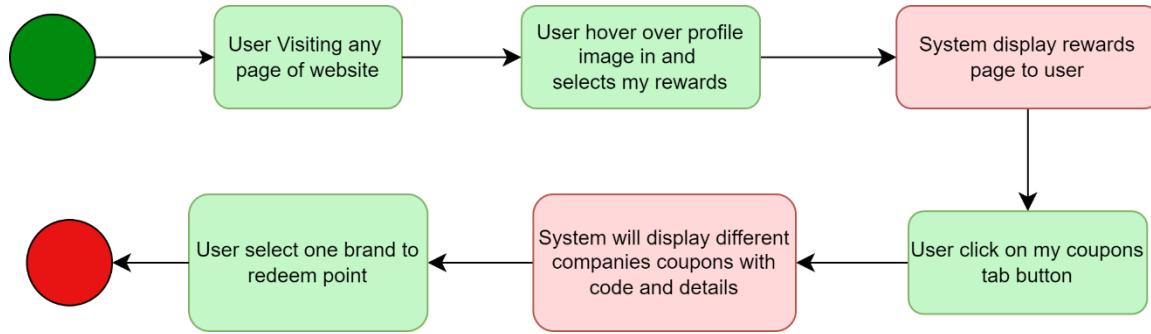


Figure 127: Task flow diagram of "different coupons" using draw.io [1][16]

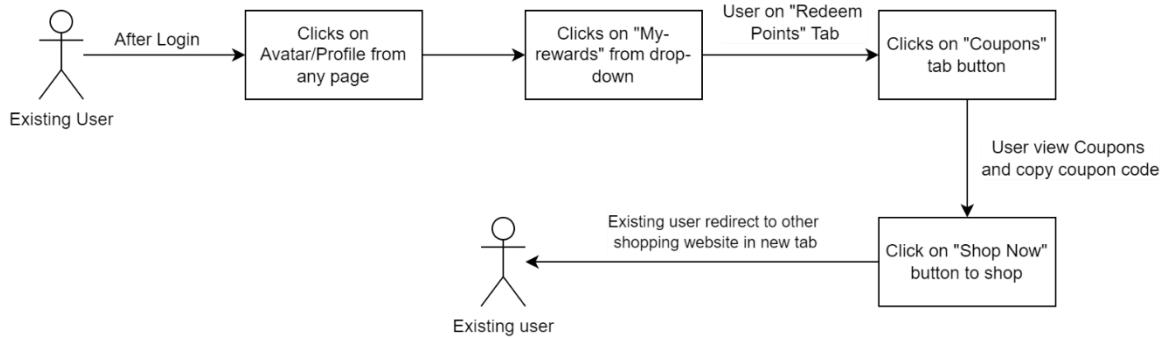


Figure 128: Clickstream of "different coupons" using draw.io [2][10]

4.2 Process and Service Workflow

As you can see, in **Figure 129**, the user visits any page of the user management feature, then in the front-end, it will display the respective form or other components, and after submitting any form, that request with data will go to the back end. After receiving the request from the front-end, it will check in queue for requests if the queue is empty, then it will go to the respective route for processing data and communicate with the database for any kind of operation and make a return response. Based on the return response, if a notification message is required, it will send a notification with a respective message to the user. [10]

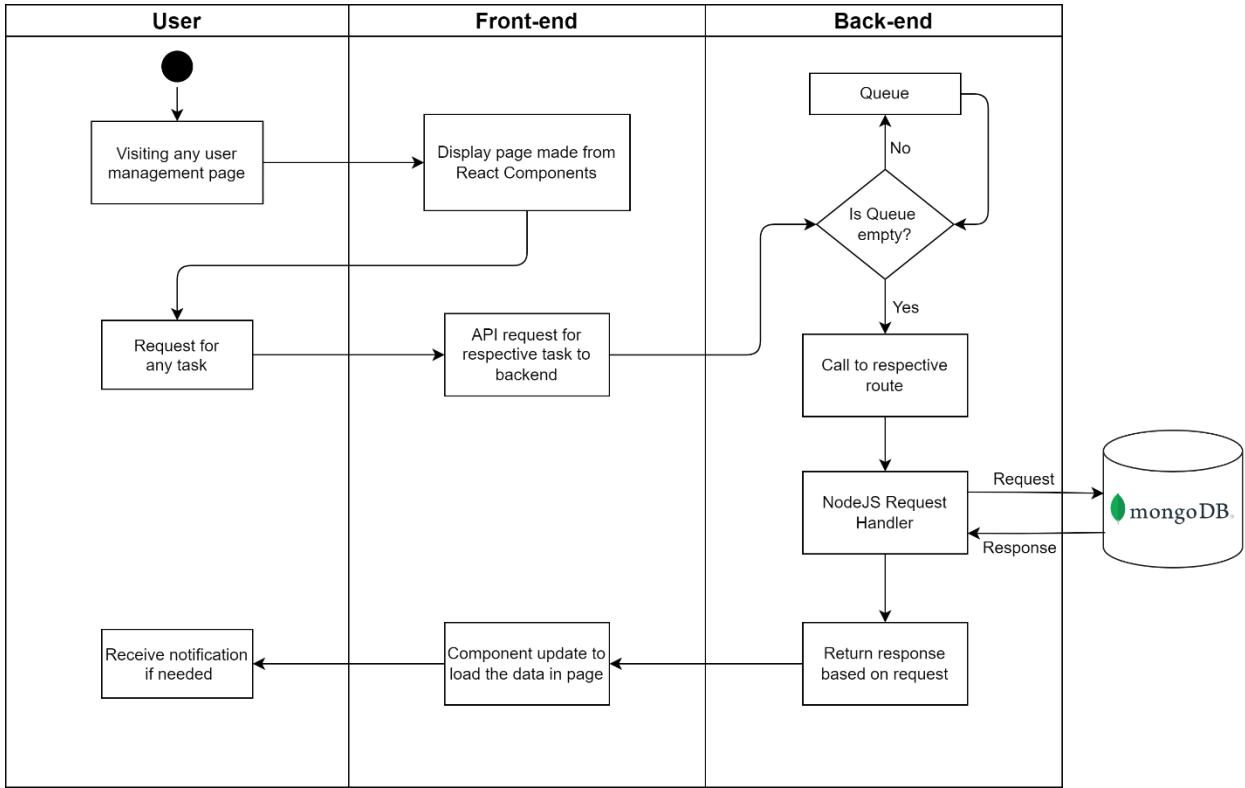


Figure 129. Process Workflow for User Management Feature using draw.io [2][10]

5. SECURITY

Security has been one of the core pillars for the Spotify project. We have properly authenticated APIs in the backend. To protect user's credentials, we have encrypted the password in both the client and server side. We have also followed recommended practices like making the token invalid and cleaning the cookies and local storage after user logout. We encrypted password using SHA-256 and b-crypt with secret key.

6. CONCLUSION

The web-application Spotify intends to bring convenience to administrators of a sports club and members (fitness enthusiasts) using the club facilities by transforming the daily activities of the club from a paper-based platform to a digital platform via a website. It provides an improved way of managing various business processes like selling membership, maintenance of authorized club members, record-keeping of financial transactions, club equipment, and in-house events like yoga. It also provides a platform for users with same interests to connect through blogging. The application also provides a medium to showcase and advertise merchandise products wherein the users can read and be aware of the product. Club admins manage these products. The features of the applications facilitate club members in performing operations like purchasing

memberships and booking events at the convenience of being at home instead of physically visiting the club for registration and inquiry purposes.

7. RECOMMENDATIONS

Due to time constraints, the project's defined scope is limited and can be extended to incorporate numerous features and integrations among features. One feature called broadcast notification is handy if club administrators want to announce some information/provide an update on the club activities. The broadcast notification feature can be integrated with event management and facility management to notify club members when a new facility/event is added to the club. The attendance tracking system is another feature that tracks attendance records using the system's facilities. The main advantage is that it keeps track of how many people use a particular facility and maintains its capacity. Another feature that can be added is Google map location, which allows users to find the nearest sports club centres by integrating it with Google Maps API.

On the other hand, we desired to incorporate a chatbot into the system. As a result, the administrator does not need to stay online to respond to the message. Our AI bot will respond and save the user data for future queries to the admin. As a result, the user can receive an immediate and helpful response from the bot.

For further improvements, merchandise feature can be improved by providing card functionality and integrating it with payment modules to purchase products through website. Rewards can also be integrated with merchandise by providing reward points as a token of gratitude for purchasing products.

8. REFERENCES

- [1] Lucid.app. *Outdated browser / Lucid*. [online] Available at: <https://lucid.app/documents#/dashboard> [Accessed 21 July 2022].
- [2] App.diagrams.net. *Flowchart Maker & Online Diagram Software*. [online] Available at: <https://app.diagrams.net/> [Accessed 21 July 2022].
- [3] “Balsamiq rapid, effective, and fun wireframing software: Balsamiq,” *Rapid, Effective and Fun Wireframing Software*. [Online]. Available: <https://Balsamiq.com/> [Accessed: 21 July 2022].
- [4] Mui.com. n.d. *MUI: The React component library you always wanted*. [online] Available at: <https://mui.com/> [Accessed 21 July 2022].
- [5] Npmjs.com. n.d. *npm*. [online] Available at: <https://www.npmjs.com/> [Accessed 21 July 2022].

- [6] Reactjs.org. n.d. *React – A JavaScript library for building user interfaces*. [online] Available at: <https://reactjs.org/> [Accessed 21 July 2022].
- [7] Ckeditor.com. n.d. *Content styles - CKEditor 5 Documentation*. [online] Available at: <https://ckeditor.com/docs/ckeditor5/latest/installation/advanced/content-styles.html> [Accessed 21 July 2022].
- [8] “Custom payment flow,” *Stripe.com*, 2022. [Online]. Available: <https://stripe.com/docs/payments/quickstart>. [Accessed: 21 July 2022]
- [9] M. Iqbal, “How to deploy a React + Node app to Heroku in 3 minutes without the command line,” *freeCodeCamp.org*, Sep. 07, 2019. [Online]. Available: <https://www.freecodecamp.org/news/deploy-a-react-node-app-to/> . [Accessed: 21 July 2022]
- [10] S. Kansodaria, “CSCI 5709 - Assignment 2,” dal.brightspace.com, 30-Jun-2022. [Online]. Available: https://dal.brightspace.com/d2l/common/viewFile.d2lfile/Database/MTI3MTQ2Mjg/A2_Kansodaria_Soham.pdf?ou=222418 . [Accessed: 21-July-2022].
- [11] P. Kabtiyal, “CSCI 5709 - Assignment 2,” dal.brightspace.com, 30-Jun-2022. [Online]. Available: https://dal.brightspace.com/d2l/lms/dropbox/user/folders_history.d2l?db=150462&grpid=0&isprv=0&bp=0&ou=222418 . [Accessed: 21-July-2022].
- [12] A. Jayanthi, “CSCI 5709 - Assignment 2,” dal.brightspace.com, 30-Jun-2022. [Online]. Available: https://dal.brightspace.com/d2l/common/viewFile.d2lfile/Database/MTI3MTM4NTQ/A2_Jayanthi_Aravind.pdf?ou=222418 . [Accessed: 21-July-2022].
- [13] A. Karruppiyah, “CSCI 5709 - Assignment 2,” dal.brightspace.com, 30-Jun-2022. [Online]. Available: https://dal.brightspace.com/d2l/common/viewFile.d2lfile/Database/MTI3MTQ0NjY/A2_Karruppiyah_AlaguSwrnam.pdf?ou=222418 . [Accessed: 21-July-2022].
- [14] S. Jariwala, “CSCI 5709 - Assignment 2,” dal.brightspace.com, 30-Jun-2022. [Online]. Available: https://dal.brightspace.com/d2l/common/viewFile.d2lfile/Database/MTI3MTQ2MDM/A2_Samarth_Jariwala.pdf?ou=222418 . [Accessed: 21-July-2022].
- [15] N. Jayapal, “CSCI 5709 - Assignment 2,” dal.brightspace.com, 30-Jun-2022. [Online]. Available: https://dal.brightspace.com/d2l/common/viewFile.d2lfile/Database/MTI3MTQ3NTg/A2_Jayapal_Navya.pdf?ou=222418 . [Accessed: 21-July-2022].

- [16] A. Karrupiah, A. Jayanthi, N. Jayapal, P. Kabtiyal, S. Jariwala, and S. Kansodaria, “CSCI 5709 - Project Proposal,” dal.brightspace.com, [Accessed: 21-July-2022]. [Online]. Available: https://dal.brightspace.com/d2l/common/viewFile.d2lfile/Database/MTI2Mzk3MzY/D1_Group10.pdf?ou=222418 . [Accessed: 21-July-2022].
- [17] “GoodLife Fitness: Gyms and Fitness Clubs: Fit and healthy good life,” *Goodlife Site.* [Online]. Available: <https://www.goodlifefitness.com/home.html> [Accessed: 21-Jul-2022].
- [18] YMCA of Greater Halifax Dartmouth. n.d. *YMCA of Greater Halifax / Dartmouth - Building Your Spirit, Mind and Body.* [online] Available at: <https://ymcahx.ca/> [Accessed 21 July 2022].
- [19] Dalsports.dal.ca. n.d. *Dalhousie Campus Recreation.* [online] Available at: <https://www.dalsports.dal.ca/> [Accessed 21 July 2022].