Group Project Log

Group Name:	MusicO

Group Members:	Anuj Upadhyay, Fenil Nikeshkumar Shah, Neehar Parupalli Ramakrishna, Simar Saggu, Utkarsh Nileshbhai Patel, Venkata Kanakayya Prashant Vadlamani

Deliverable:	MusicO Project Technical Report

Group Member Name	Work Done %
Anuj Upadhyay	16.66%
Fenil Nikeshkumar Shah	16.66%
Neehar Parupalli Ramakrishna	16.66%
Simar Saggu	16.66%
Utkarsh Nileshbhai Patel	16.66%
Venkata Kanakayya Prashant Vadlamani	16.66%
Total:	**100%**

TECHNICAL REPORT

Project Group 1

MusicO

Members and Contributors

Utkarsh Patel

B00883322

ut436516@dal.ca

Fenil Nikeshkumar Shah

B00871894

fn777214@dal.ca

Venkata Kanakayya Prashant Vadlamani

B00883901

vn855760@dal.ca

Neehar Parupalli Ramakrishna

B00883889

nh238106@dal.ca

Anuj Upadhyay

B00862821

an584590@dal.ca

Simar Saggu

B00883509

sm905783@dal.ca

Dalhousie University

01, August 2021

ABSTRACT

MusicO is a music streaming website that lets users listen to music without the hassle of downloading it. MusicO contains various features like creating and importing playlists, following other users, searching for artists and albums, rating, and shuffling albums. The application's uses ReactJS and Bootstrap for client-side, SpringBoot framework for server-side and MySQL as the database technology. The client-side interacts with the server-side by using REST API calls. About 9 out of the 12 proposed features have been completed and this report documents their interaction design and drawbacks.

KEYWORDS

MusicO | Music streaming application | Playlists | React Application | REST API | SpringBoot | UX approaches | Web Design | Workflow Architecture | Web Services

Table of Contents

1 INTRODUCTION

The purpose of MusicO web application is to provide an easy and flawless music streaming platform to save them from the hassle of searching for their favourite songs, artists, or albums. MusicO provides a unique feature that allows the user to search for the specific users, allows them to connect with their friends and view their playlists as well as import it to their own account. The user-interface of the MusicO web application provides easy access to all the functionalities to their users which enables them to use the application while they are multi-tasking.

MusicO aims to act as the stress buster in the hectic life of the users. MusicO can be accessed from anywhere and anytime using the internet. MusicO caters the music for the users with any taste with the large database with the music for people of different age groups and ethnicities [16].

1.1 Live Project URL

Our GitHub code for front-end can be found at:
https://github.com/sm905783/group1-musico.git

Our GitHub code for back-end can be found at: **https://github.com/neeh97/group1-musico-backend**

Our application can be accessed at: **https://group1-musico.herokuapp.com/**

2 BACKGROUND

2.1 Competitive Landscape

MusicO is an online music streaming web application with a vast database of the songs from all the genres and from older generation to the currently trending songs. Existing competitors of our web application are Spotify.com, Apple Music, Amazon Music.

2.2 Problem and Approach

We are providing ad free music in our application without any premium subscription. We have also added a feature to import the playlists from the friends a user follows.

3 APPLICATION DETAILS

3.1 Target User Insights

This website is a universal application which can be used by people of different age groups and ethnicities. However, this application targets the user base based on certain demographics. This application intends to target three main groups of individuals: Students, Corporate employees, and Common users. The students will mostly be using this application to listen to soft music while studying and the trending

songs. The corporate employees can listen to music of their taste while working or even while they are on short breaks. This web application has a vast database that caters the music of all types and generations right from the old 70's to the currently trending songs. Hence, the application can be used even by the common users to listen to music for relaxation.

MusicO is designed to be as user-friendly as possible, so that the users belonging to different generations can easily navigate through the application with the basic knowledge of computers and who are familiar with the general structure of the other websites. The simple and elegant design of the application allows the corporate users to use the application with very few clicks to perform any specific operation to use the application even while they are working. The continually updating music database of the application enables the students userbase to get access to trending songs sooner after their release in the market [16].

3.2 User-Centred Design Approach

3.2.1 Information Architecture

Figure 1 shows the sitemap of the application. This figure shows all the different web pages and the way to navigate to them along with the different buttons and actions the users can perform on the application.

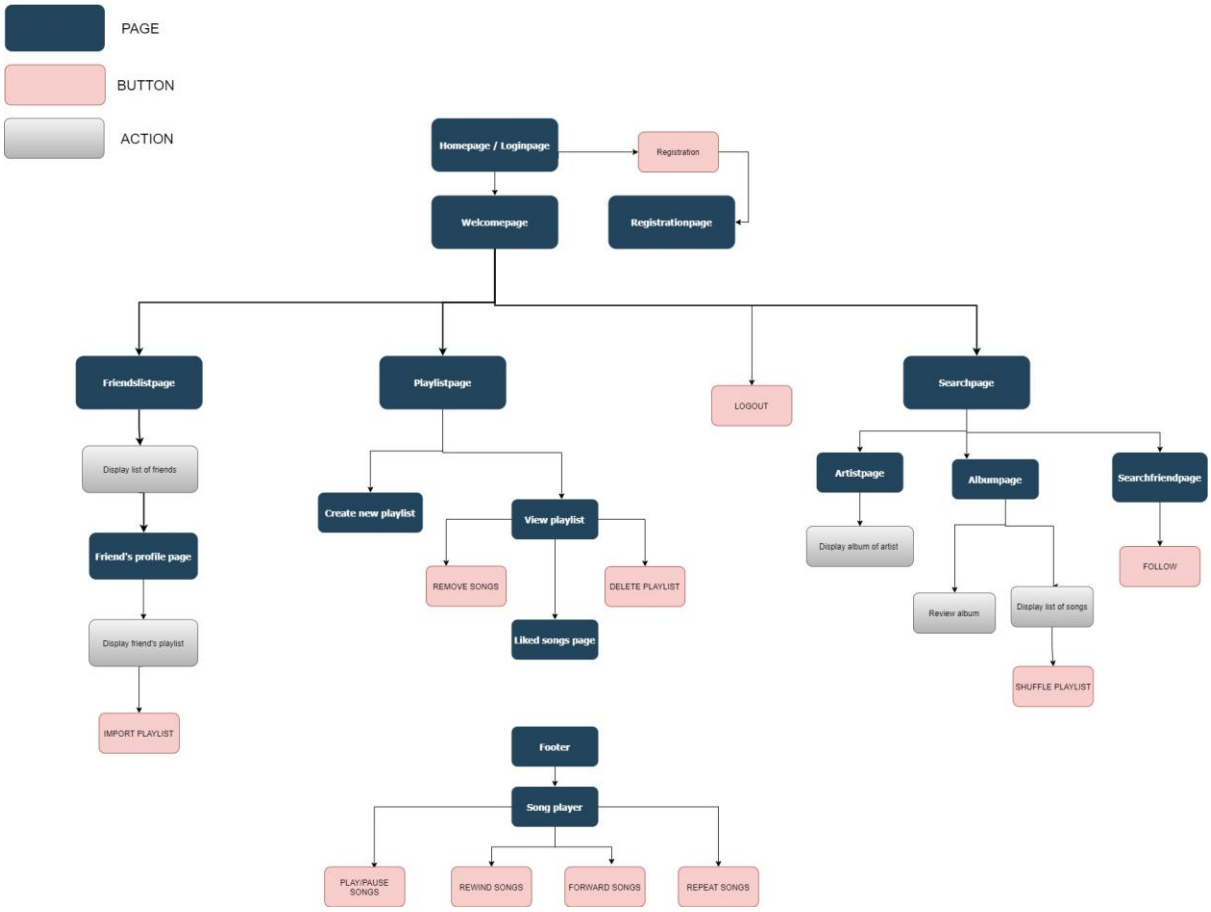

Figure 1: Sitemap for MusicO [2,16]

3.2.2 Design and Layout

Figure 2 illustrates the login page where the user logs into the application. When the user enters valid credentials, the user is navigated to the welcome page.

Figure 2: Wireframe for Homepage [4,16]

Figure 3 shows the registration page of our website "MusicO." To use "MusicO," a user needs to register first. Users need to enter all the required fields like full name, phone number, email Id, password, confirm password to register. Users can also upload their profile pictures.

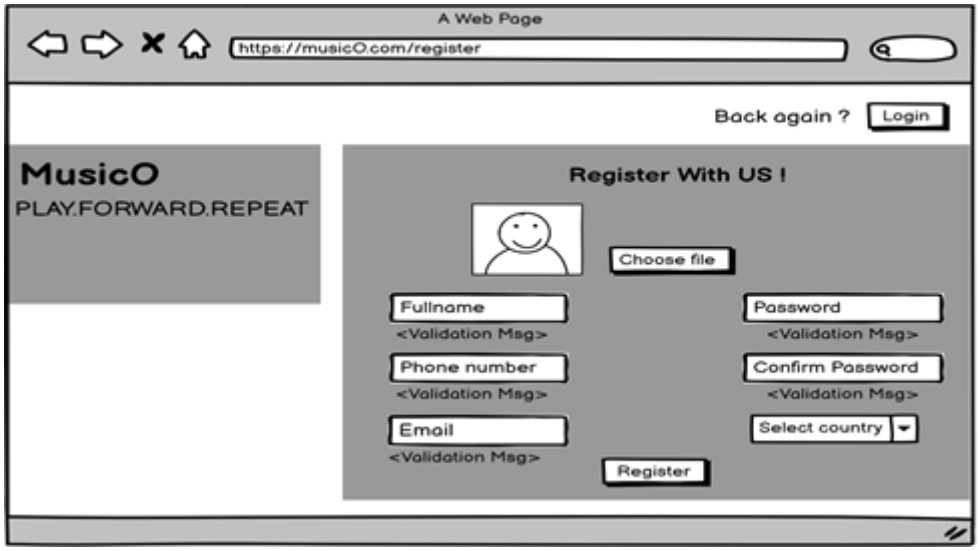

Figure 3: Wireframe for Registration Page [4,16]

Figure 4 shows the profile page of our website "MusicO". The profile displays user-specific data like followers, following, user playlist etc. A profile page allows users to view and edit their profiles. It also allows a user to share their profile with others. Users can edit all the fields except the phone number and email Id.

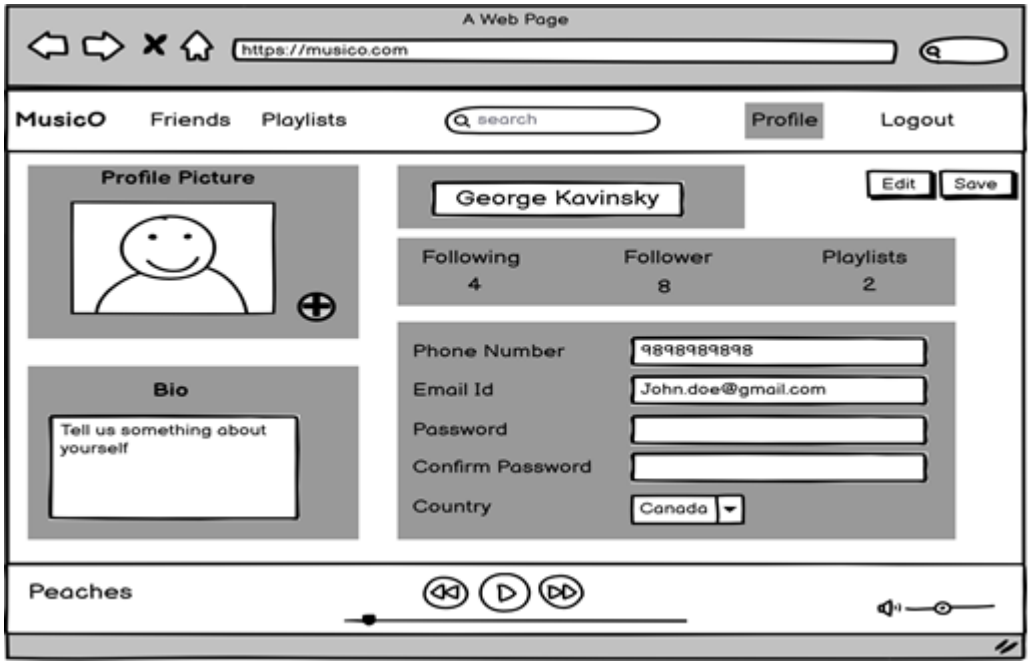

Figure 4: Wireframe for Profile Page [4,16]

Figure 5 illustrates the welcome page of the application which shows the user's history of the recently visited artists / album's page and the recently played songs. From here users can revisit the listed pages or replay/like the listed songs.

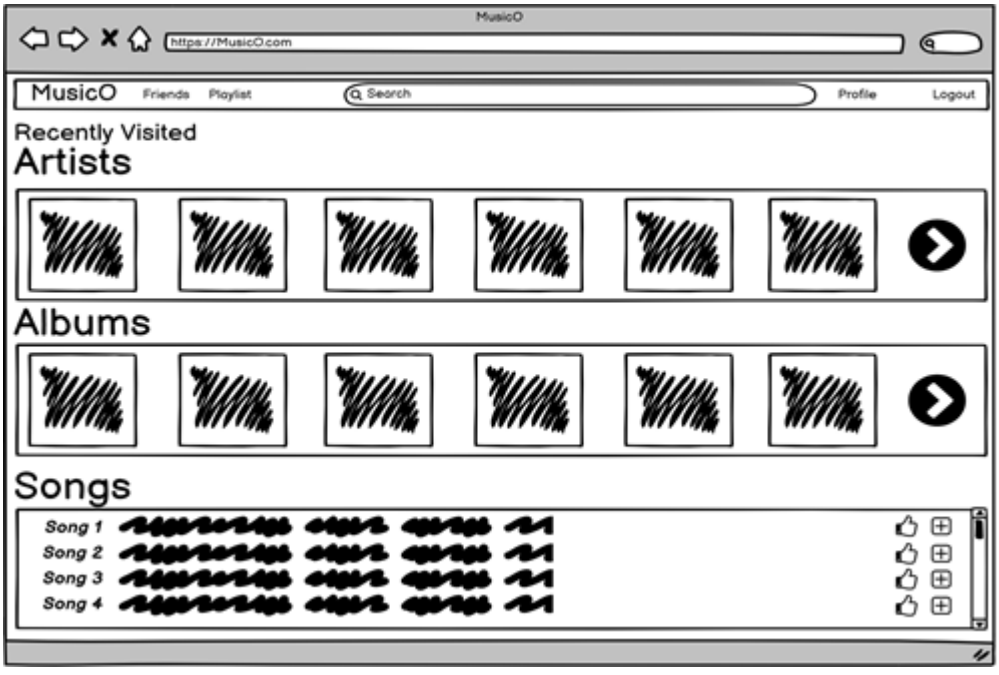

Figure 5: Wireframe for Welcome Page [4,16]

Figure 6 illustrates a wireframe of a user playlist page where the user can create a different playlist of his favourite songs, that he can listen to. User is given an option to create new playlist, delete playlist and delete songs from the playlist.

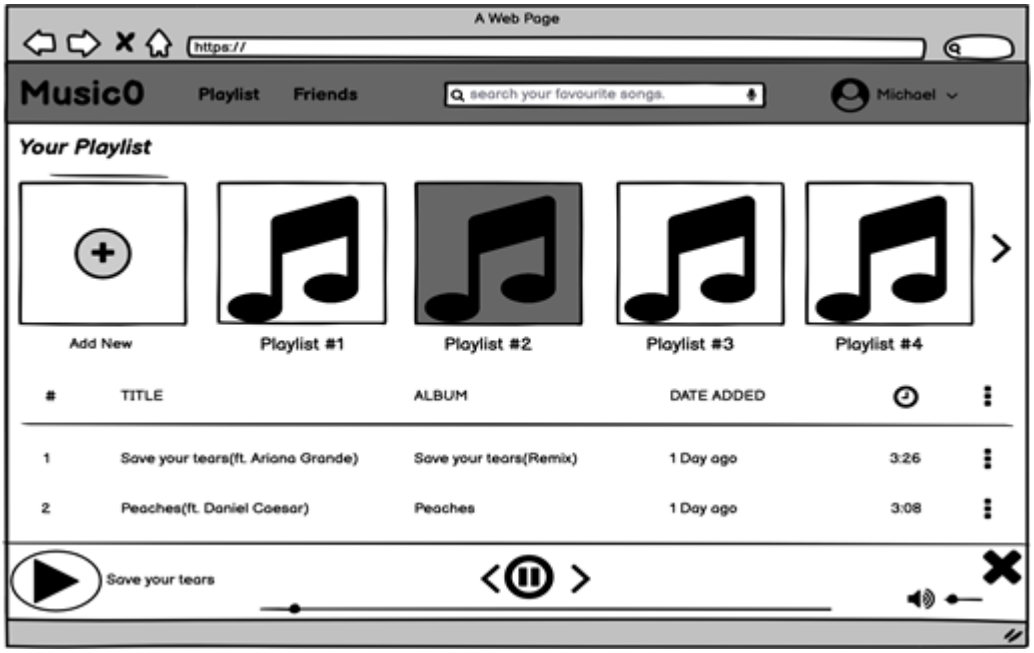

Figure 6: Wireframe for user playlist page [4,16]

Figure 7 illustrates the wireframe of the search results page of the application which shows the search results of the keyword entered by the user. This page shows the link to artists and albums page and a list of songs that matched with the keyword searched. From here users can visit the listed artist/albums page or play/like the listed songs.

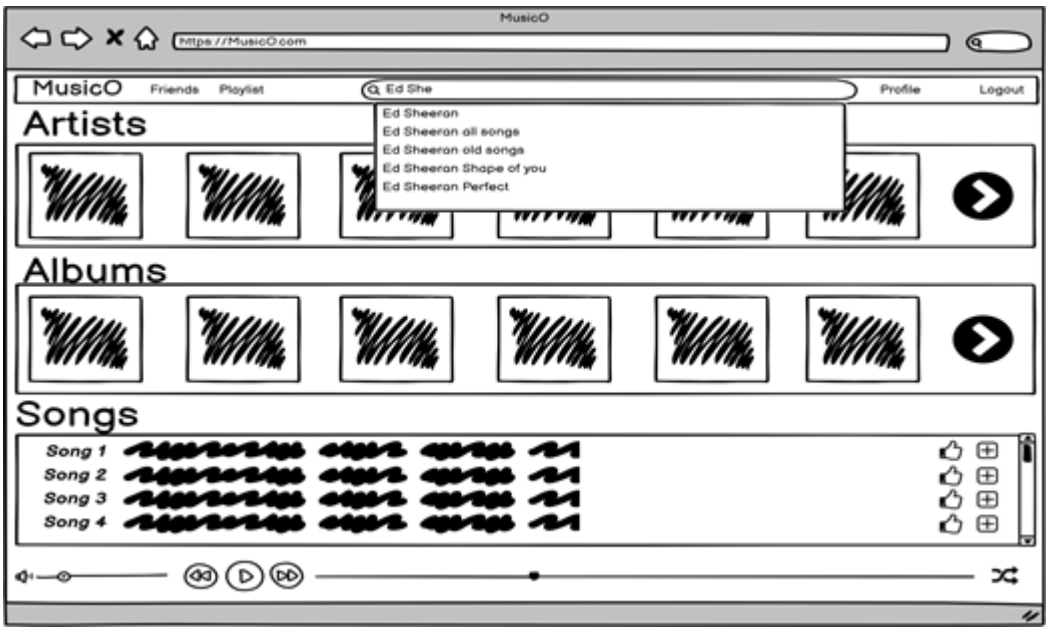

Figure 7: Wireframe for Search results page [4,16]

Figure 8 shows the friends page of our website "MusicO". This page shows the list of all the friends of a particular user. The user can view their profile from here or unfollow them.

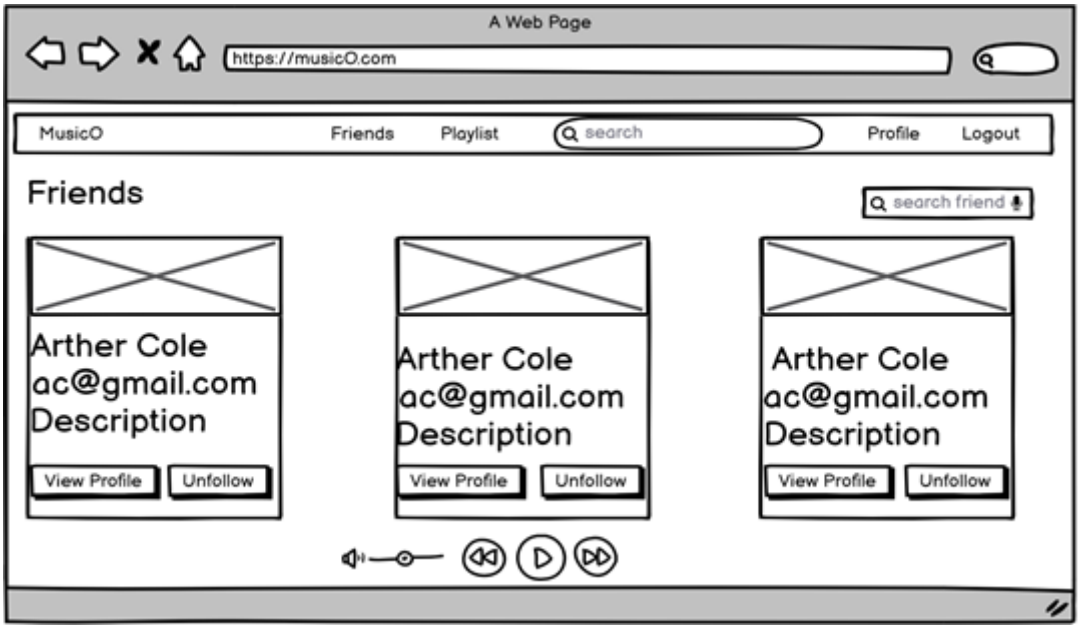

Figure 8: Wireframe for friend's page [4,16]

Figure 9 illustrates a wireframe of a friend's playlist page where a user can view all his friend's playlist. User is also provided with a feature to import his friend's playlist into his playlist section.

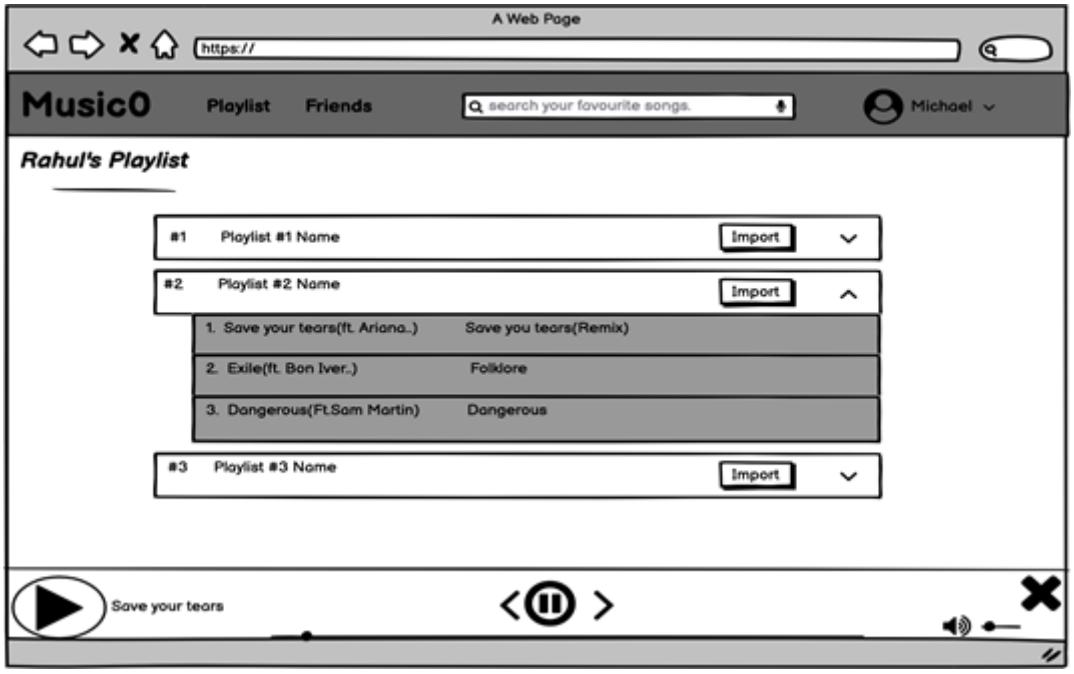

Figure 9: Wireframe for Friend's playlist page [4,16]

Figure 10 is the wireframe for the artist page. The artist page displays the entire discography or the list of the albums the artist has put out. This list is sorted by the score of the album which is rated by the user. Clicking on the link of the album redirects to the album page which contains the tracklist of the album.

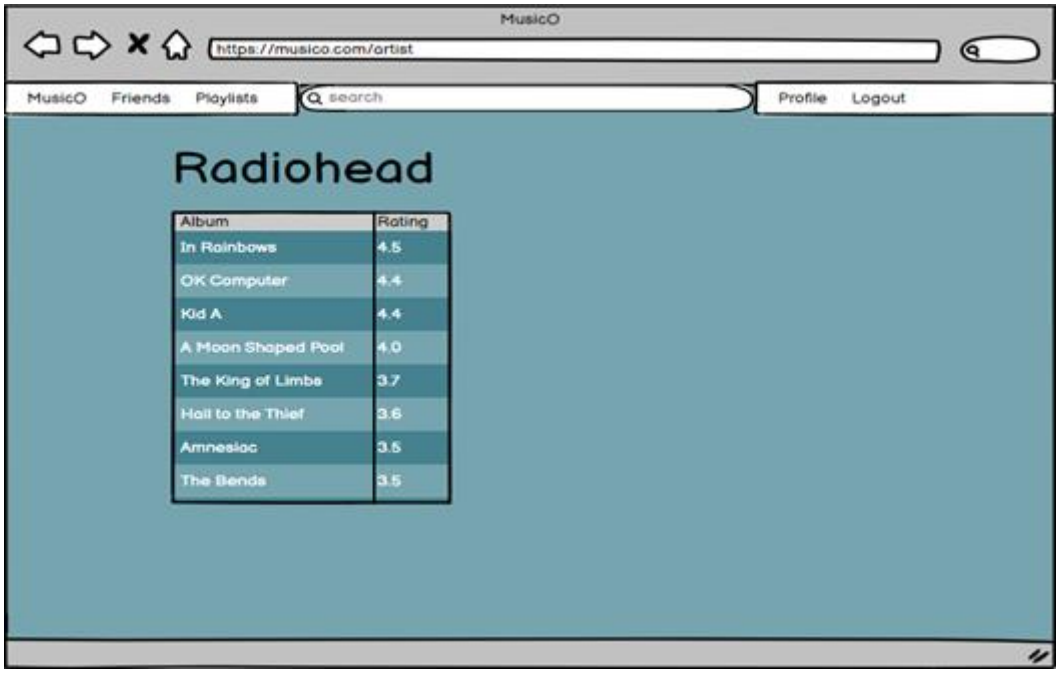

Figure 10: Wireframe for the artist page [4,16]

Figure 11 shows the album page where users can play the songs from an album and can also rate the album.

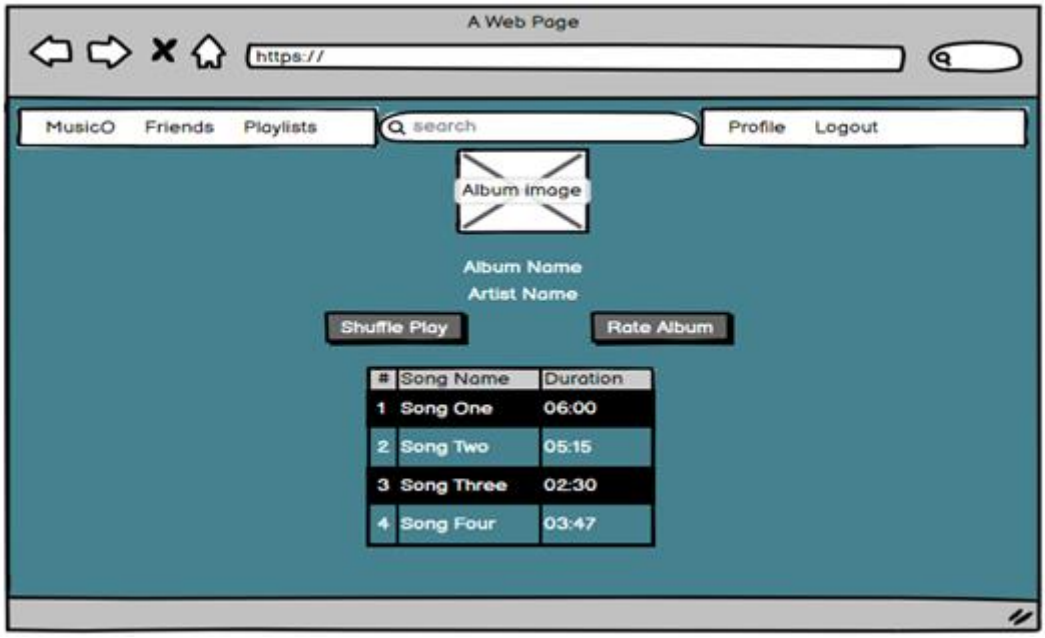

Figure 11: Wireframe for the album page. [4,16]

4 APPLICATION WORKFLOW

4.1 Interaction Design

Figure below shows the user persona for George Kavinsky. George is a 45-year-old Motion Graphic Designer who is very busy and does not like spending time on finding different music. His persona embodies the person who wants quick access to different music without much hassle.

Figure 12: User Persona [1,5,16]

Figure below shows the user persona for Michael Townley. Michael is a 27-year-old web developer. He is a music enthusiast who wants a service that is easy to use. Michael embodies the corporate worker who wants to listen to songs but does not have the time to download them.

Figure 13: User Persona [1,16]

Feature: User Registration, Login Authentication & User Profile Management

At 10, on a Monday morning, your boss comes and asks you to create a 30 second commercial for a headphone launch and gives you a day's timeline to complete this task. Your profession expects you to create ads with songs that boost the marketability of the product. So, you start looking for unique music that you can use while creating this short commercial. You are a new user who discovered our website "MusicO" while finding the music. To use our website, you are required to register yourself with us. Therefore, you decide to register to discover our website.

User Registration

1. User opens the home page/ login page of "MusicO" [user action].
2. User clicks on the Register button to open the Registration page [user action].
3. The System displays the Registration page requesting the user to enter name, phone number, email Id, country and upload a profile picture [system action].
4. The System highlights some of the fields as required which include name, phone number and email Id [system action].
5. User enters a name in the Name field [user action].
    1. System displays an error message stating that the name field cannot contain special characters or numbers [system action].
    2. User enters the name again [user action].
6. User enters email id in the Email-Id field [user action].
    a. System displays an error message stating that the email format is incorrect [system action].
    b. User enters the new email Id [user action].

7. User enters a phone number in the Phone Number field [user action].
    a. An error message is displayed saying that length cannot be more than 10. [system action].
    b. User enters the phone number again [user action].
8. User enters the password in the password and confirms the password field [user action].
    a. An error is displayed that both passwords do not match [system action].
    b. User enters the password and confirms the password again [user action].
9. Now, the user clicks on the submit button [user action].
10. System checks if the email/phone number values exist in the system. [system action].
    a. System displays an error message with the message as the phone number already exists in our system. And suggests choosing a different number [system action].
    b. User enters a different phone number this time [user action].
11. System successfully registers the new user and sends a confirmation alert message [system action].
12. User is taken to the login page again [user action].

Task Flow:

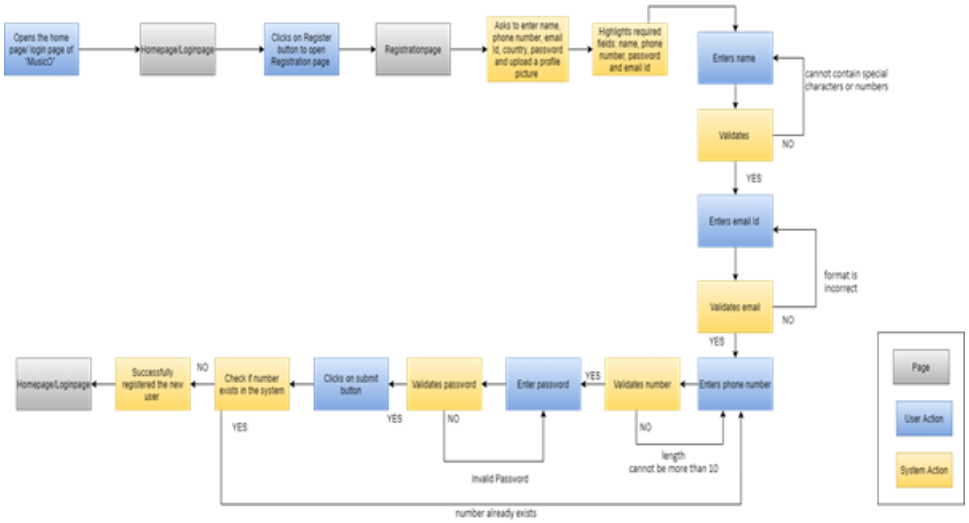

Figure 14: Task Flow Diagram - User Registration [2,10]

Clickstream:

The flow diagram below shows the click stream for user registration.

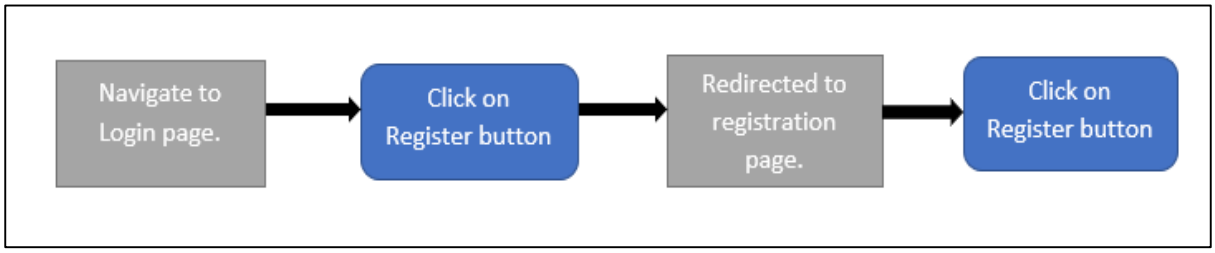

Figure 15: User Click Stream: User Registration [2,10]

Login Authentication

1. User opens the home page/ login page of “MusicO” **[user action]**.

2. System prompts users to enter registered email / phone number and password **[system action]**.
3. User enters a phone number in the Phone Number field **[user action]**.
4. System displays that the number entered is more than 10 numbers **[system action]**.
5. User enters the phone number again **[user action]**.
6. User enters a password in the Password field **[user action]**.
7. User clicks on the Login button **[user action]**.
8. System validates the field values entered by the user **[system action].**
    a. System displays an error message saying that please enter the registered phone number/ email Id or password **[system action]**.
    b. User enters phone number and password again **[user action]**.
9. User is authenticated successfully and is redirected to the welcome page **[system action].**

Task Flow:

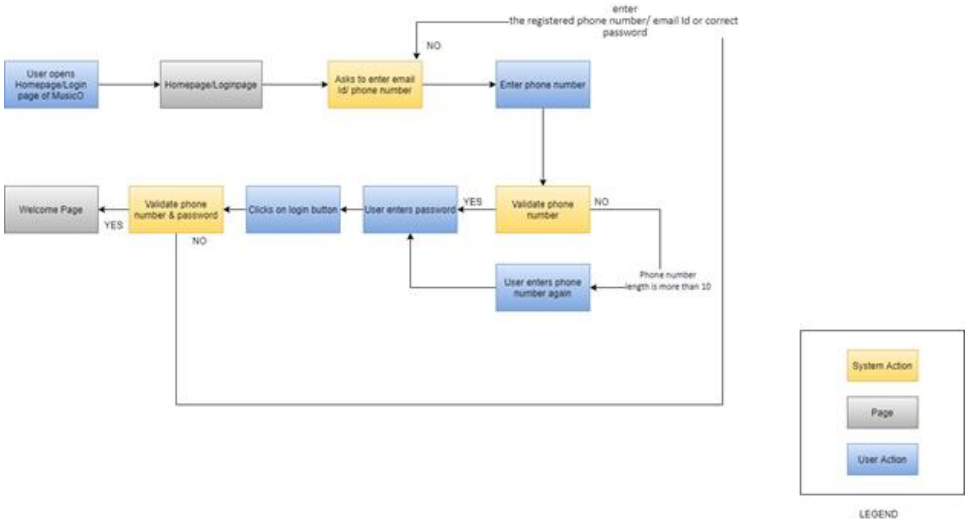

Figure 16: Task Flow Diagram - Login Authentication [2,10]

Clickstream:

The flow diagram below shows the click stream for user login.

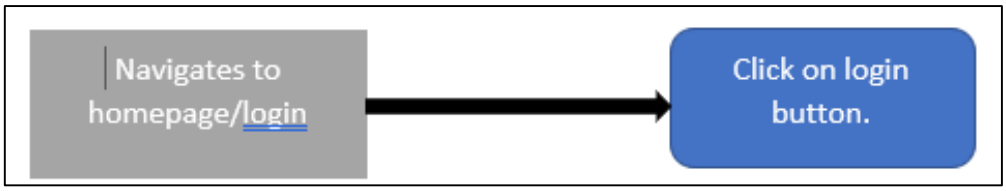

Figure 17: User Click Stream: User Login [2,10]

User Profile Management

1. User is registered and logged into “MusicO” **[user action]**.
2. User is on the welcome page **[system action]**.

3. User clicks on the profile icon present on the navigation bar to open the Profile page **[user action]**.
4. System displays the Profile page with some already filled fields **[system action].**
5. User clicks on the camera icon present in the upload image section to upload the picture **[user action].**
    a. System asks the user to choose an image from his computer to upload **[system action].**
    b. User clicks on the upload picture button **[user action].**
    c. User selects a picture from the gallery **[user action].**
    d. User clicks on the upload button **[user action].**
        i. System displays an error message saying the size of the upload is too large. And suggests uploading an image smaller than 1mb **[system action].**
        ii. User uploads another image **[user action]**.
6. System uploads the image in the image section **[system action].**
7. System displays the profile page with updated profile picture **[system action].**

Task Flow:

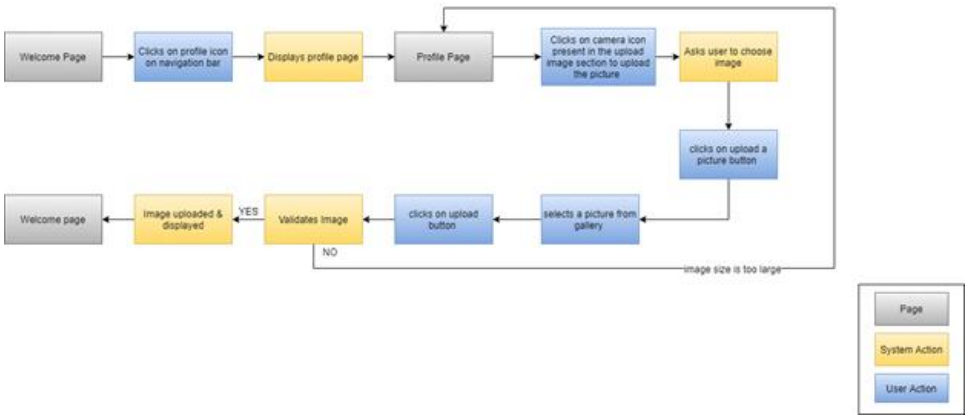

Figure 18: Task Flow Diagram - User Profile Management [2,10]

Clickstream:

The flow diagram below shows the click stream for user profile update.

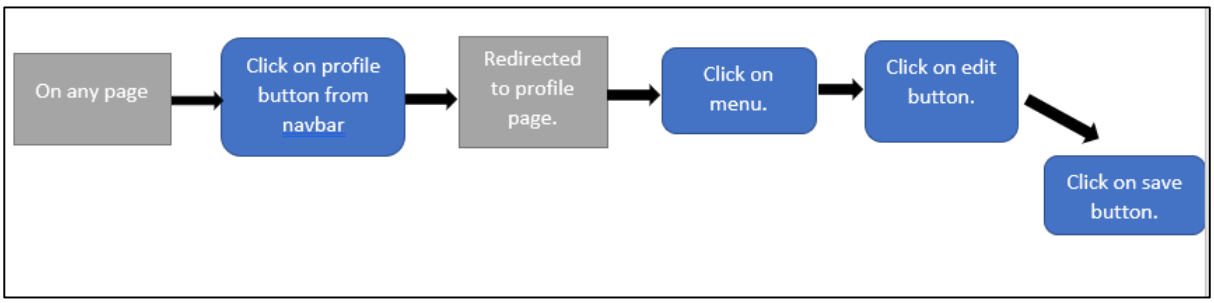

Figure 19: User Click Stream: Update User Profile [2,10]

The flow diagram below shows the click stream for view user profile.

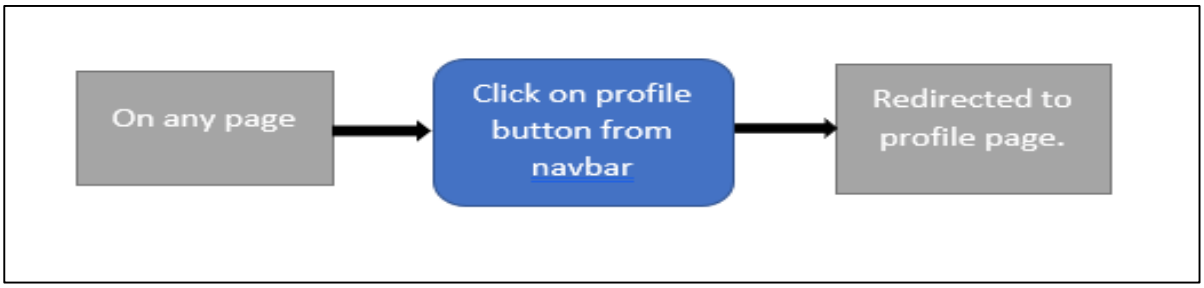

Figure 20: User Click Stream: View User Profile [2,10]

The flow diagram below shows the click stream for share user profile.

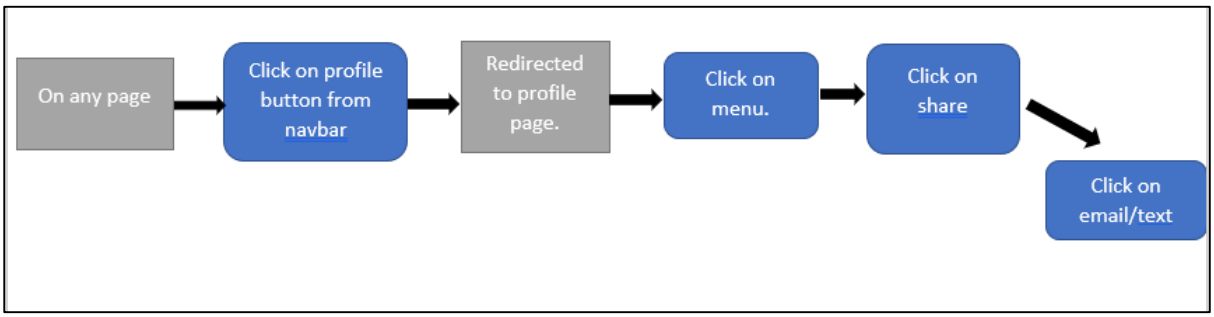

Figure 21: User Click Stream: Share User Profile [2,10]

User Logout:

The flow diagram below shows the click stream for logout task.

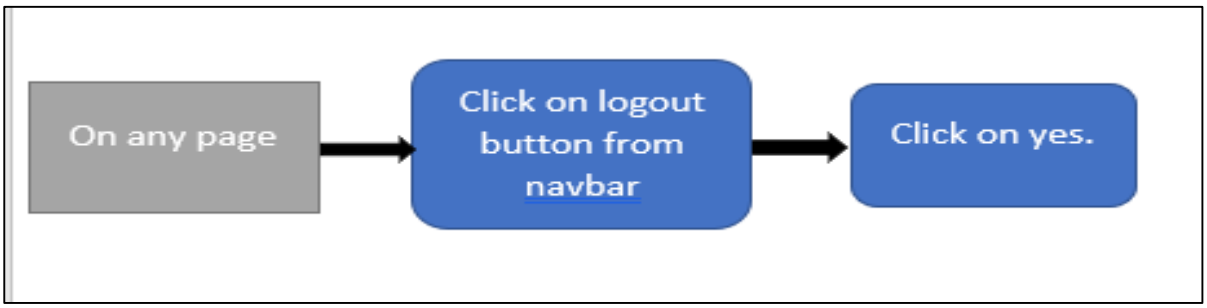

Figure 22: User Click Stream: User Logout [2,10]

Forgot Password:

The flow diagram below shows the click stream for logout task.

Figure 23: User Click Stream: Forget Password [2,10]

Delete User Account:

The flow diagram below shows the click stream for how user can delete their account.

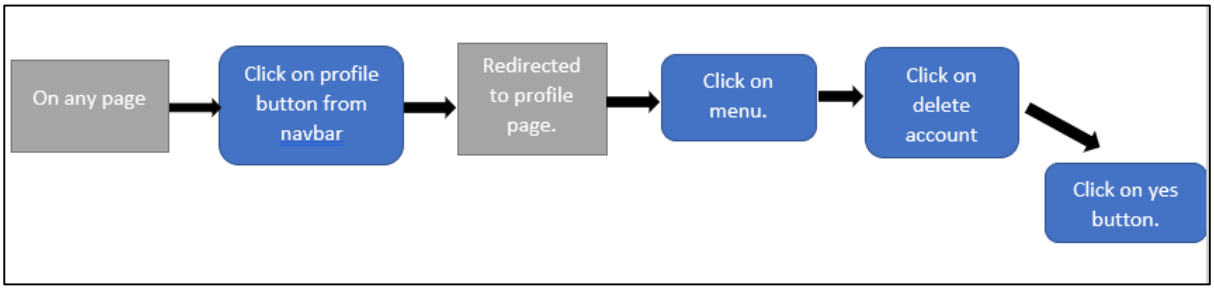

Figure 24: User Click Stream: Delete User Account [2,10]

Feature: Music Player Controls

You love listening to music while you work, but you are often approached by other people for guidance and enquiry. To address them, you often are required to remove your headphones. Our application's music player feature lets you pause the songs as you see fit and play them from the point you paused the song. You can also rewind to listen to a certain part of the song if you missed it.

Use Case:

1. User logs in to MusicO **[user action]**
2. User selects a song to play **[user action]**
3. User clicks on pause button **[user action]**
    a. System pauses the playing song **[system action]**
4. User clicks on play button **[user action]**
    a. System playing the song **[system action]**
5. User clicks on the skip button **[user action]**
    a. System skips the current song and plays the next song in the queue **[system action]**
6. User clicks on the rewind button **[user action]**
    a. System stops playing the current song and plays the previous song in the queue **[system action]**

Task Flow:

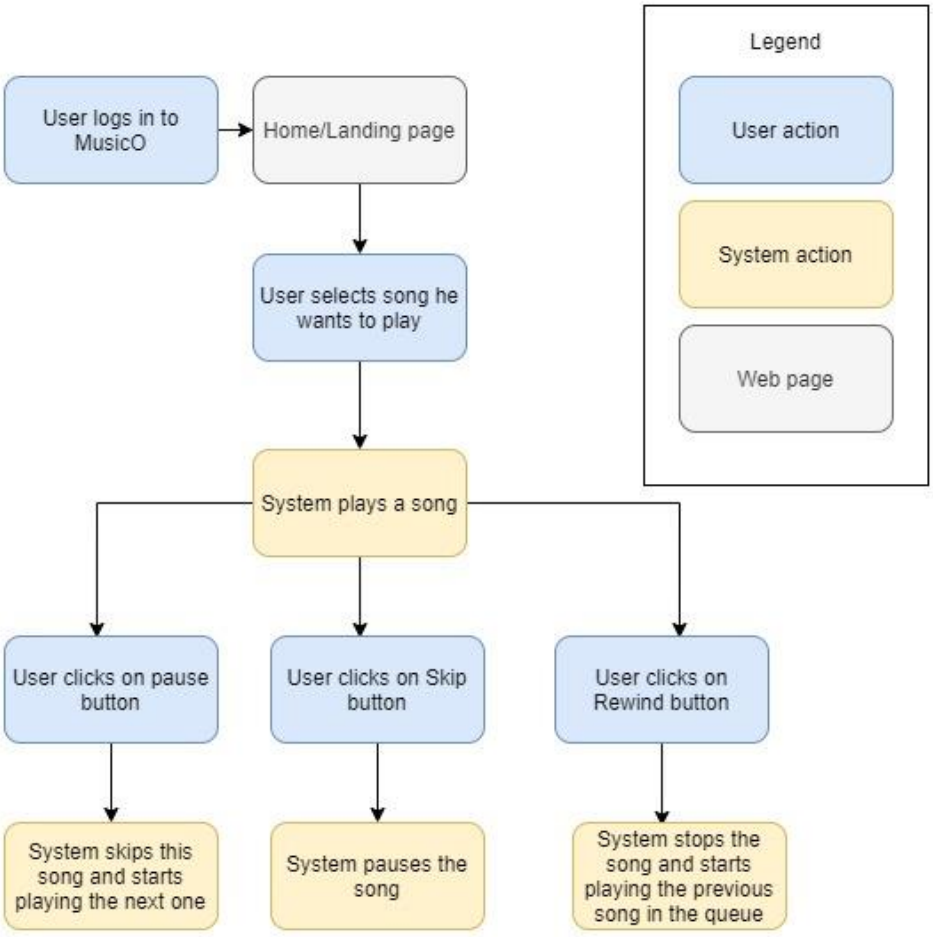

Figure 25: Task Flow Diagram - Music Player Controls [2]

Feature: Artist Page

You grew interested in David Bowie's music from the day our application recommended you Ziggy Stardust. You like the album and would like to explore the rest of Bowie's discography. However, the wide range of albums Bowie's put out over the years makes it very intimidating for you to pick a starting point. Our artist page displays all the albums David Bowie has put out in a descending rating order. It is now easy for you to listen to his discography from Bowie's best to worst.

Artist Page:

1. User logs in to the system using their credentials **[user action]**
2. User searches for an artist **[user action]**
3. System displays a list of relevant results for the keyword the user has searched for. **[system action]**
4. User clicks on the artist they searched for **[user action]**

5. System displays a list of albums released by the artist sorted by their score in a descending order. **[system action]**
6. User clicks on a certain album to open its track list. **[user action]**

Task Flow:

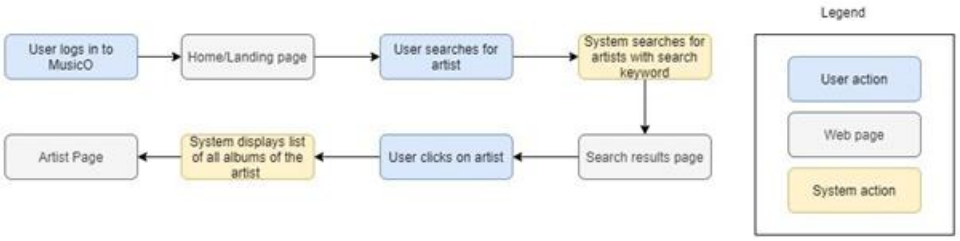

Figure 26: Task Flow Diagram - Artist page [2]

Clickstream:

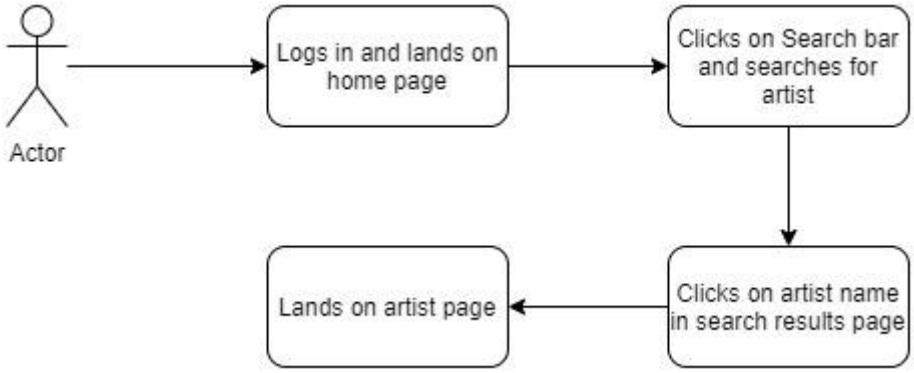

Figure 27: Clickstream – Artist page [2,11]

Feature: Album Page

You are a new user who is always looking to listen to new songs. Our application, MusicO, has recommended an album that has caught your attention. After listening to the entire album, you liked it enough to rate the album with the rating feature in the album page. You started listening to the same album daily and used the shuffle play option in the album page to randomise the play order of the songs.

Album Page

1. User logs into MusicO application with valid credentials [user action]
2. In the welcome page, the user searches for an album from the search bar. [user action]
3. The system displays the results of user search in searchResults page. [system action]
4. User finds the album, and navigates to album page by clicking on the album [user action]
5. In the album page, the system displays all songs of the album [system action]

6. User can play the songs by clicking on the song [user action]
7. User can also play songs from the album randomly by clicking on Shuffle play button [user action]
8. Users can rate the album by clicking on the rate album button which is present in the album page. [user action].

Task Flow:

Figure 4 shows the task flow diagram for the task to play a song in the album page.

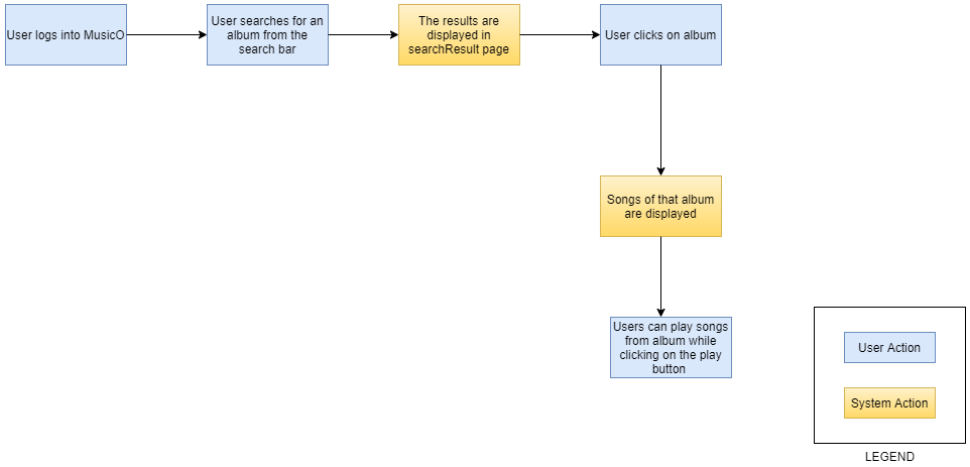

Figure 28: Task flow diagram: Play a song [2,12]

Clickstream:

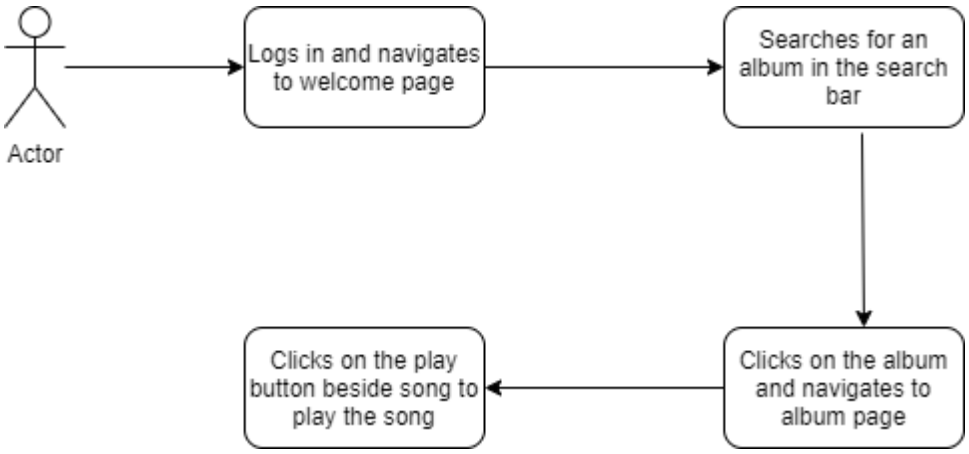

Figure 29: shows the click stream for the task to play a song in the album page [2,12]

Task Flow:

Figure 6 shows the task flow diagram for the task to rate an album in the album page.

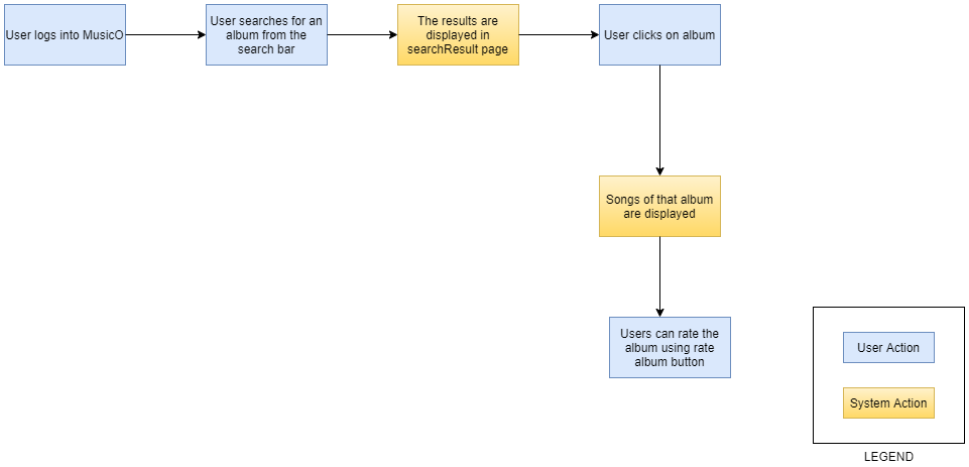

Figure 30: Task flow diagram: Rate Album [2,12]

Clickstream:

Figure 7 shows the click stream for the task to rate an album in the album page

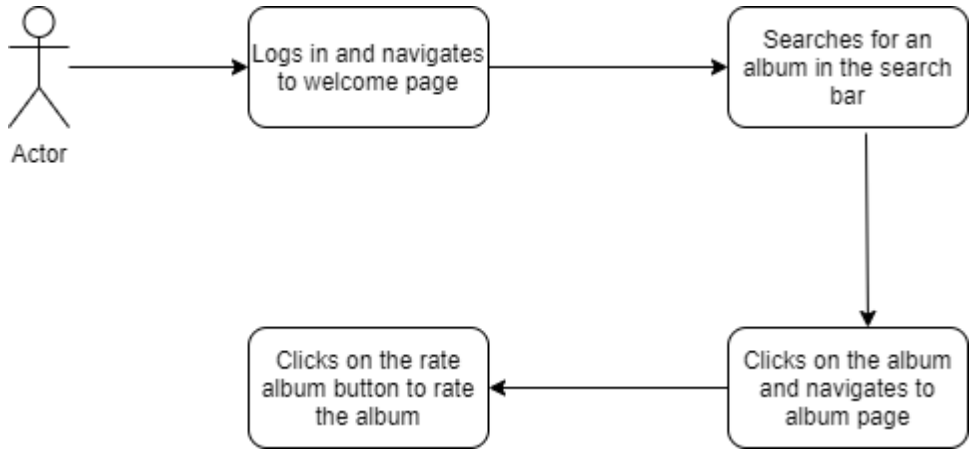

Figure 31: Click stream diagram to rate an album [2,12]

Task Flow:

Figure 8 shows the task flow diagram for the task to shuffle play entire album in the album page.

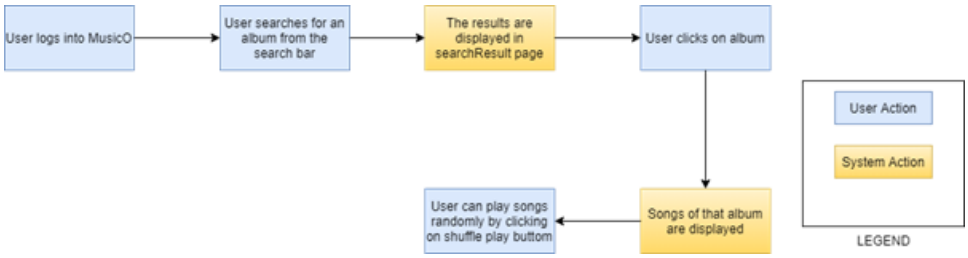

Figure 32: Task flow diagram: Shuffle play [2,12]

Clickstream:

Figure 9 shows the click stream for the task to shuffle play entire album in the album page.

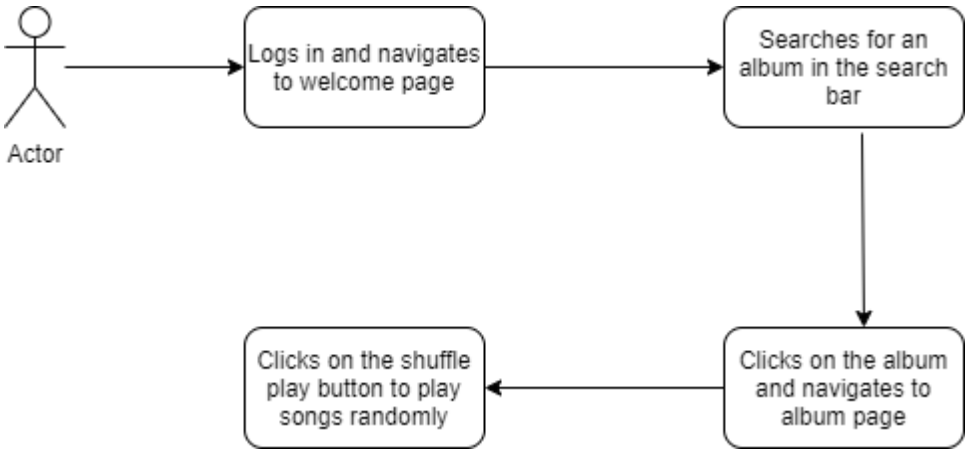

Figure 33: Click stream diagram to shuffle play an album [2,12]

Feature: Searching Songs/Artists/Albums

You find searching the song a tedious task. Our application MusicO, with its searching feature, makes it simpler and easier to search for the songs, albums, or artists with its autocomplete feature. Now, you just have to type in a few words and the search feature will come up with the related suggestions of songs, artists, and albums helping you to find the correct song you are looking for. Further from here, you can easily add the songs to the currently playing queue or like the songs or visit the album's page / artist's page best matching to the keyword you entered.

Search Songs/Artists/Albums

Assumption: User is successfully logged in to the MusicO application from the browser

1. User clicks on the search box on the Welcome Page. **[user action]**
2. User types in some letters for the song/album/artist to be searched for. **[user action]**
3. While the user is typing, the System suggests related suggestions with autocompletion of the search item. **[system action]**
    a. System displays "Not Found". **[system action]**
    b. System displays some random recommendations of artists, albums, and songs. **[system action]**
4. System displays a page with the songs related to the keyword entered into the search box, list of artists and albums best matching to the keyword. **[system action]**
5. User can now select the song and add to the queue or visit the list album/artist page. **[user action]**

Task Flow

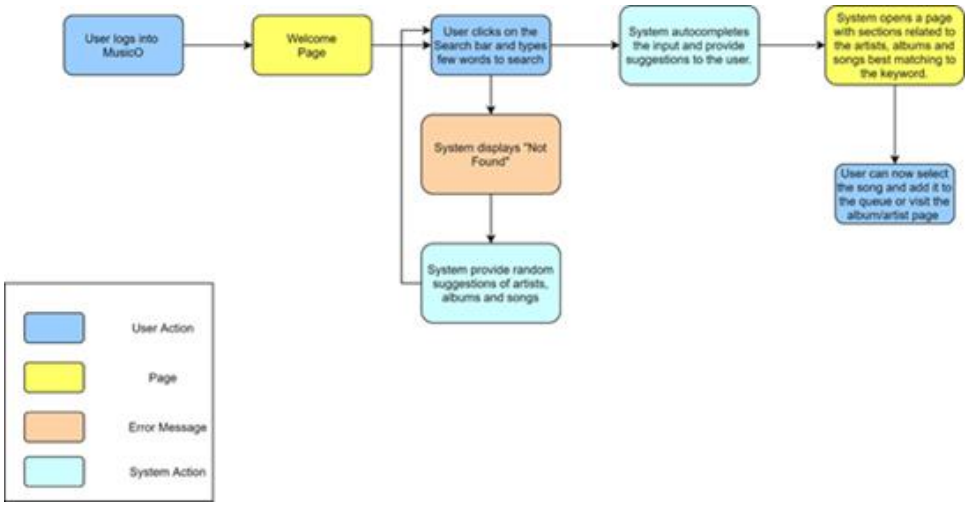

Figure 34: Task Flow Diagram - Search page [2,13].

Clickstream:

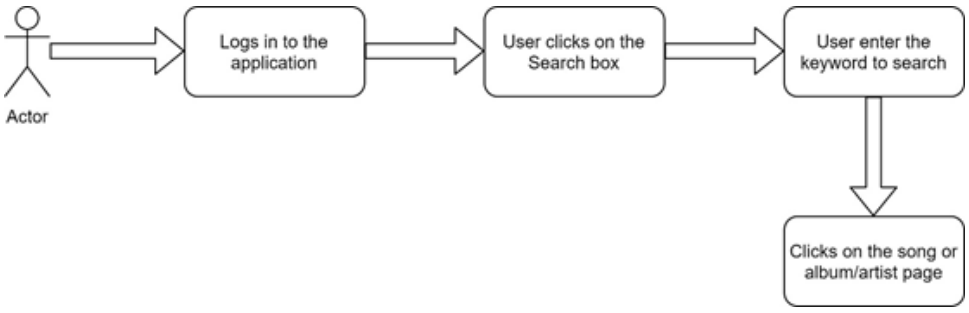

Fig 35: Search Page Click Stream [2,13].

Feature: Viewing History

You spend several hours on his laptop for work, and love to listen to songs while working. MusicO provides a wide range of songs and easy access to all the features of the application. MusicO enables Michael to easily start listening to previously played songs right from the homepage of the application after logging in. MusicO has a consistent design with a navigation bar at the top allowing easy access to other features of the application and the music player at the bottom of each page making it easy to access while working.

Assumption:

1. The user logs into the MusicO application with valid credentials **[user action]**.
2. The user will be navigated to the welcome page [**user action]**.
3. The system displays the user history of recently visited album/artist's page along with the user's history of the recently played songs [**system action]**.

Task Flow:

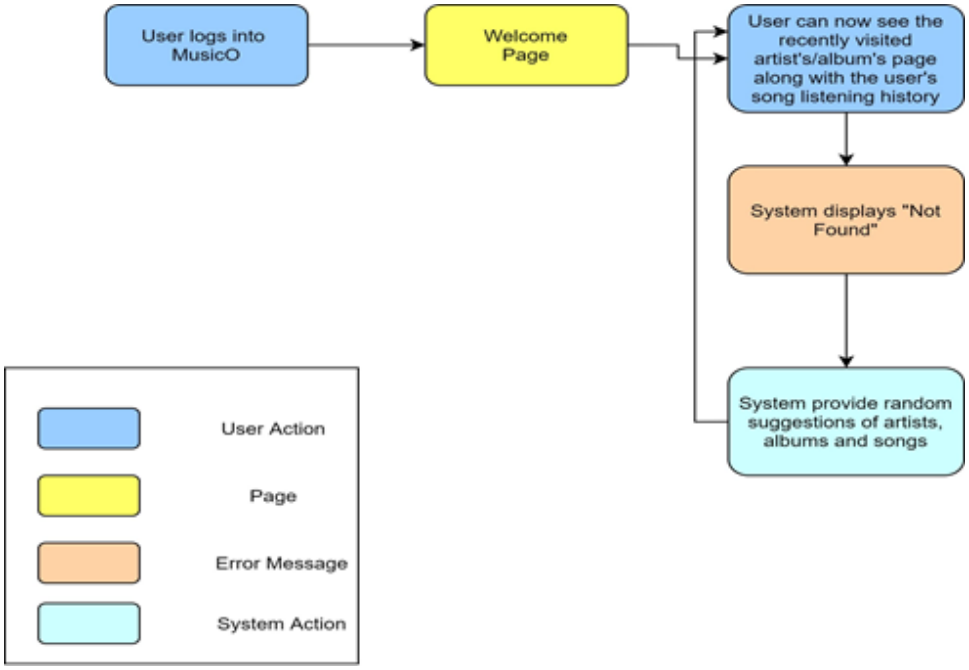

Figure 36: Task Flow Diagram - Viewing listening history of user [2,13].

Clickstream:

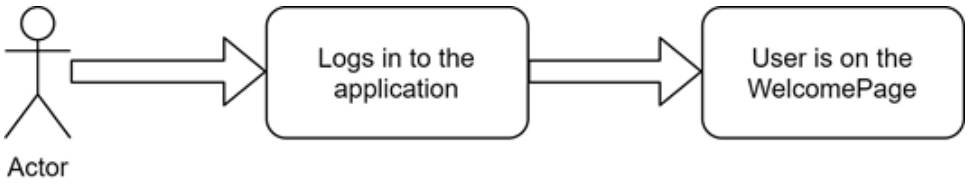

Fig 37: Welcome Page Click Stream [2,13].

Feature: **Create new Playlist**

You are a fitness person who enjoys working out and loves listening to encouraging music while working out. However, as a song ends, you must pause your workout to find and play the next good music. Lately, you got to know about this application called MusicO, which offers a feature that allows you to construct a playlist of your favourite songs that will play one after the other. You can also manage the playlist by adding or removing songs. With MusicO, you may now listen to his favourite music without having to take a break from his workout.

Use Case: Creating new playlist

1. User logs into the MusicO application with valid credentials. **[user action]**
2. System displays the welcome page. **[system action]**
3. Users click on the playlist button available on the navigation bar. **[user action]**
4. System displays the playlist page. **[system action]**
5. User clicks on create new playlist button. **[user action]**
6. User enters the name of the playlist. **[user action]**
    a. System shows playlist already exists. **[system action]**
    b. System prompts the field to enter the name again. **[system action]**
    c. User enters the name of the playlist. **[user action]**
7. System displays ‘new playlist created.’ **[system action]**

Task Flow:

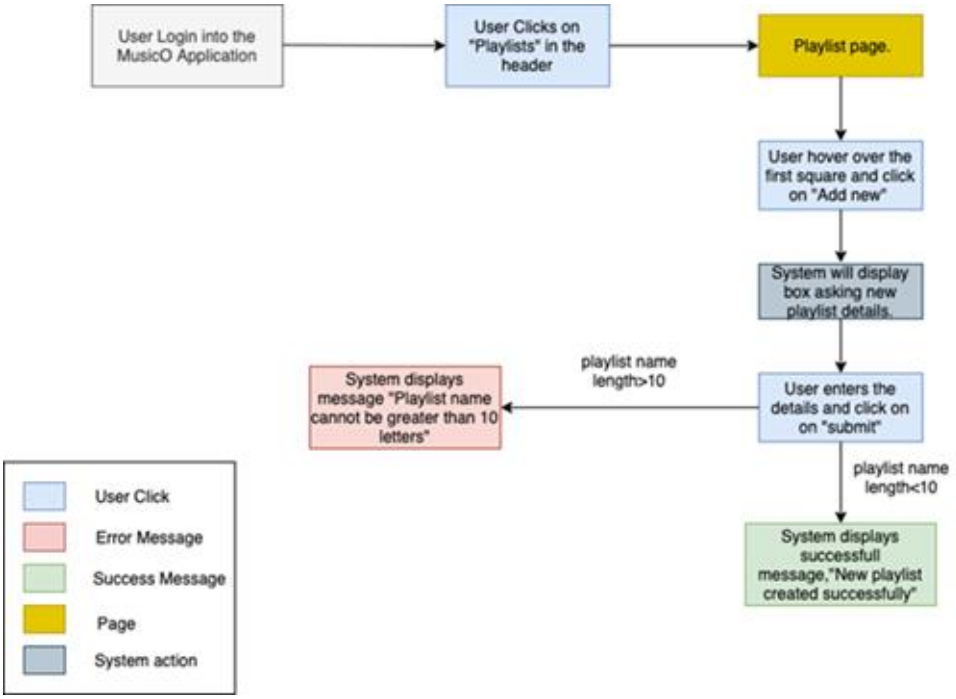

Figure 38: Task Flow Diagram - Creating playlist [2,14]

Use Case: Add song to playlist

1. Application displays a welcome page once Michael is logged in. **[System Action]**
2. Michael searches for the song in the search bar provided in the navigation bar. **[User Action]**
3. Application shows all the songs starting with text entered by Michael. **[System Action]**
4. Michael clicks on the “Add to playlist” option in front of the song. [User Action]
    a. Application displays the message “No playlist exists, please create one from the Playlist page.” **[System Action]**
    b. Michael navigates to the Playlist page and clicks on the “create new playlist” button. **[User Action]**
    c. Michael provides the name of the playlist and clicks the “ok” button. **[User Action]**
    d. Application displays the message “New playlist created successfully.”
    e. Michael navigates back to the search page and again searches for the song. **[User Action]**
5. Application displays the list of different playlists that Michael created. **[System Action]**
6. Michael selects the playlist in which he wants to add the song. **[User Action]**
7. If the songs are already present, the application will display the message” the song is already present in the playlist.” **[System Action]**

8. Application displays the message "Successfully added to playlist." **[System Action]**

Remove song from a playlist

1. User logs into the MusicO application with valid credentials. **[user action]**
2. System displays the welcome page. **[system action]**
3. User clicks on the playlist button available on the navigation bar. **[user action]**
4. System displays the playlist page. **[system action]**
5. User selects the playlist they want to view. **[user action]**
6. System shows the playlist to the user. **[system action]**
7. User clicks the remove song button given besides each song. **[user action]**
8. System removes that song. **[system action]**

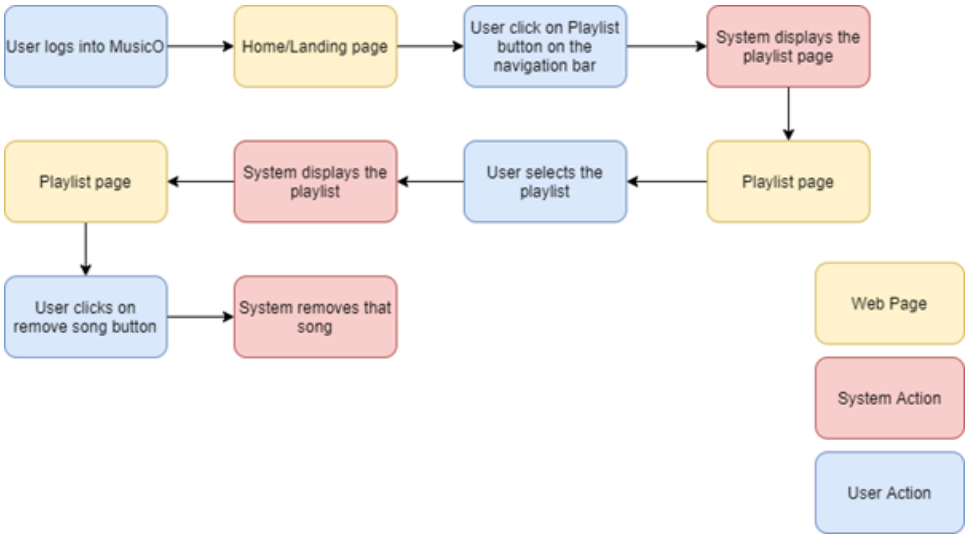

Figure 39: Task Flow Diagram - Remove song from playlist [2,14]

Delete a playlist

1. User logs into the MusicO application with valid credentials. **[user action]**
2. System displays the welcome page. **[system action]**
3. User clicks on the playlist button available on the navigation bar. **[user action]**
4. System displays the playlist page. **[system action]**
5. User selects the playlist they want to view. **[user action]**
6. System shows the playlist to the user. **[system action]**
7. User clicks on the delete playlist button. **[user action]**
8. System deletes the playlist. **[system action]**

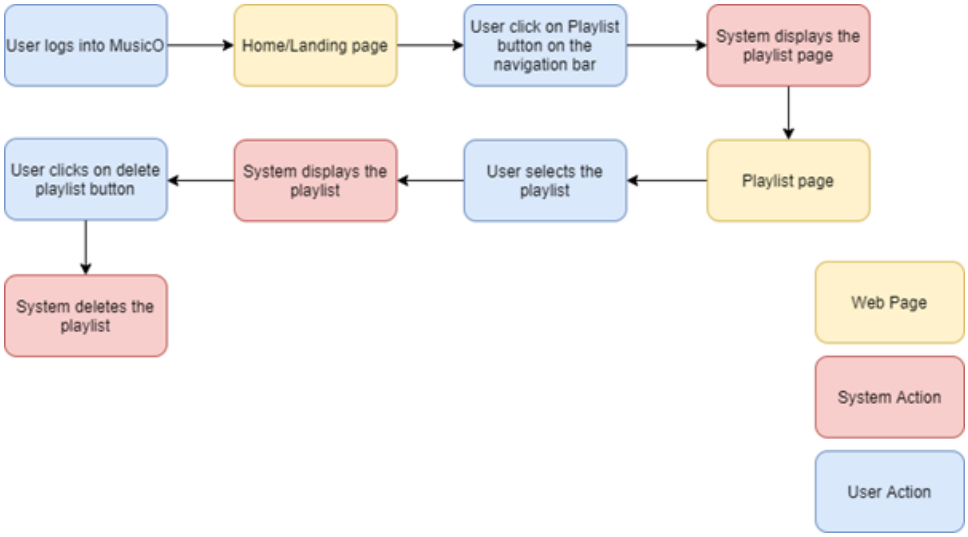

Figure 40: Task Flow Diagram - Delete a playlist [2,14]

View playlist

1. User logs into the MusicO application with valid credentials. **[user action]**
2. System displays the welcome page. **[system action]**
3. User clicks on the playlist button available on the navigation bar. **[user action]**
4. System displays the playlist page. **[system action]**
5. User selects the playlist they want to view. **[user action]**
6. System shows the playlist to the user. **[system action]**

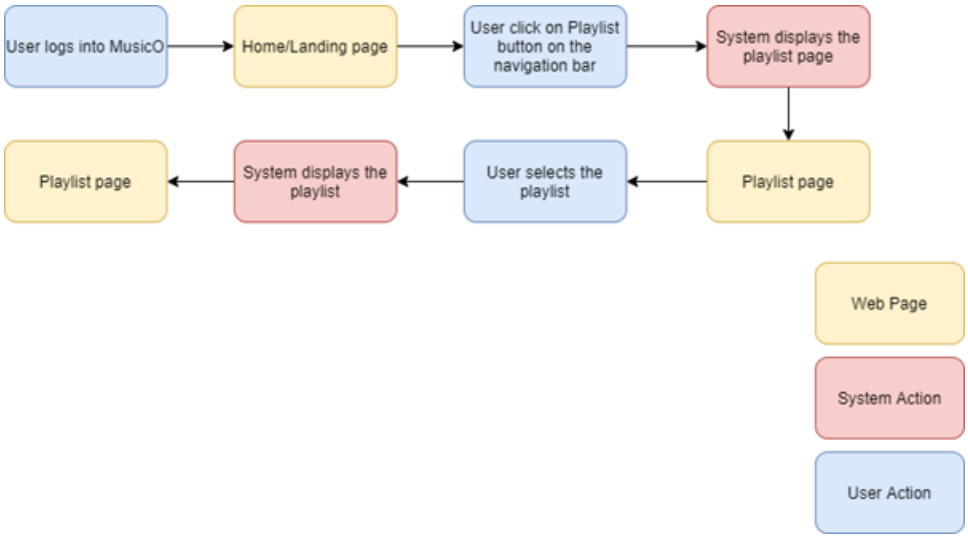

Figure 41: Task Flow Diagram - View playlist [2,14]

View liked songs page

1. User logs into the MusicO application with valid credentials. **[user action]**
2. System displays the welcome page. **[system action]**
3. Users click on the playlist button available on the navigation bar. **[user action]**

4. System displays the playlist page. **[system action]**
5. User clicks on the liked songs button. **[user action]**
6. System displays the list of liked songs by the user. **[system action]**

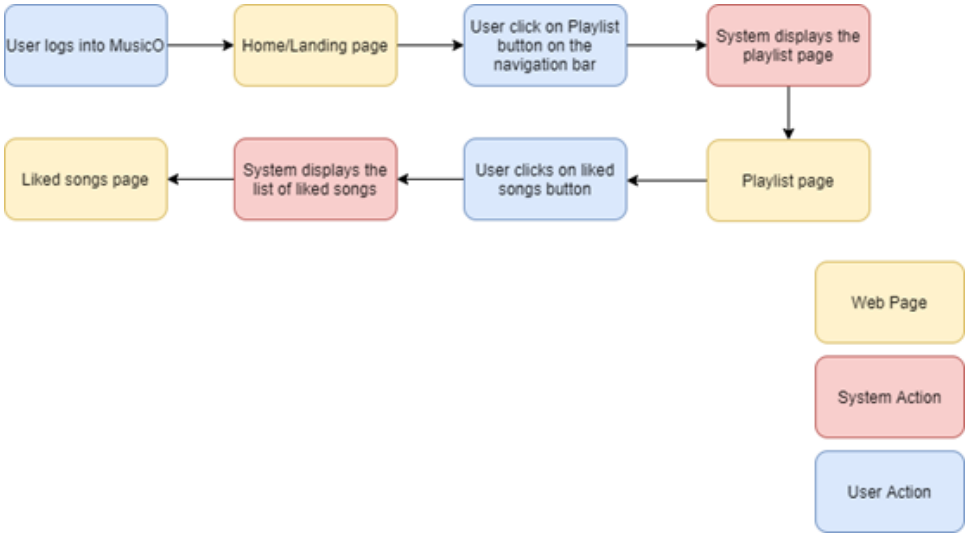

Figure 42: Task Flow Diagram - Liked song

Feature: Importing a friend's playlist

Michael, 27, a web developer at Microsoft often stays back at the office and works after the office hours. He is mostly accompanied by his colleague Mathew who works with him on the same project. After most of the staff has left the place, they like to code while listening to nice music. Michael is always fascinated by the songs that Mathew plays. But going through Mathew's entire playlist and adding all those songs in his own playlist is an exhausting task. MusicO offers the feature to directly import any of your friend's playlist without any inconvenience. Michael can now listen to as many songs as he wants from his friend's playlist without the hassle of manually finding those songs and adding them to his playlist.

Use case: Import friend's playlist

1. User opens the musicO.
2. The system displays the homepage/login page.
3. User enters their username and password and hits the login button.
    a. System displays an "incorrect username/password" message.
    b. System prompts the user to try again.
    c. User enters the username and password again and hits the login button.
    d. System displays an "incorrect username/password" message.
    e. System prompts users to reset their password and ask for their email address.
    f. User types the email address and hits enter.
    g. System prompts the user to enter the authentication code they received via email.

h. User types the authentication code and hits enter.
i. System prompts the user to enter a new password.
j. User enters the new password and hits enter.
k. System updates the password and re-directs them to the login page.

4. User logs in into the music app with correct credentials.
5. System displays the Welcome page.
6. User clicks on Friends.
7. System displays the user's friend list.
8. User sees his/her friends.
9. User clicks on a drop-down option besides the friend's name.
10. User selects the view profile option.
11. System opens his friend's profile page.
12. User sees the profile of his/her friend.
13. System shows all the playlists of that friend.
14. User sees all the playlists of that friend.
15. User clicks on the import playlist button located beside each playlist.
16. System imports that specific playlist to the user's playlist.

Task Flow:

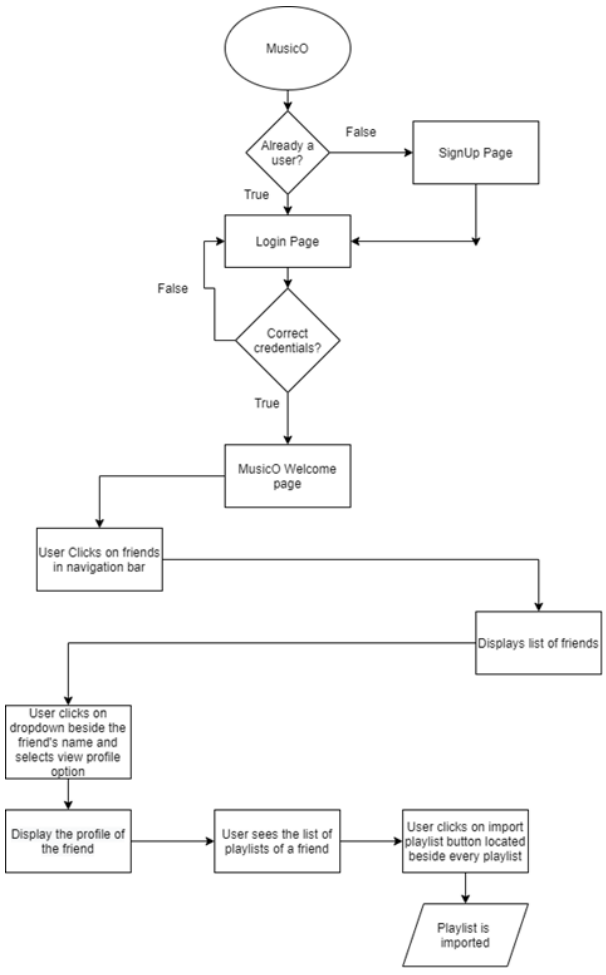

Figure 43: Task Flow Diagram - Import friends' playlist

View list of friends of a user

1. User opens the MusicO.
2. The system displays the homepage/login page.
3. User enters their username and password and hits the login button.
    a. System displays an “incorrect username/password” message.
    b. System prompts the user to try again.
    c. User enters the username and password again and hits the login button.
    d. System displays an “incorrect username/password” message.
    e. System prompts the user to reset their password and ask for their email address.
    f. User types the email address and hits enter.
    g. System prompts the user to enter the authentication code they received via email.
    h. User types the authentication code and hits enter.
    i. System prompts the user to enter a new password.
    j. User enters the new password and hits enter.
    k. System updates the password and re-directs them to the login page.
4. User logs in into the music app with correct credentials.
5. System displays the Welcome page.
6. User clicks on Friends.
7. System displays the user’s friend list.
8. User sees his/her friends.

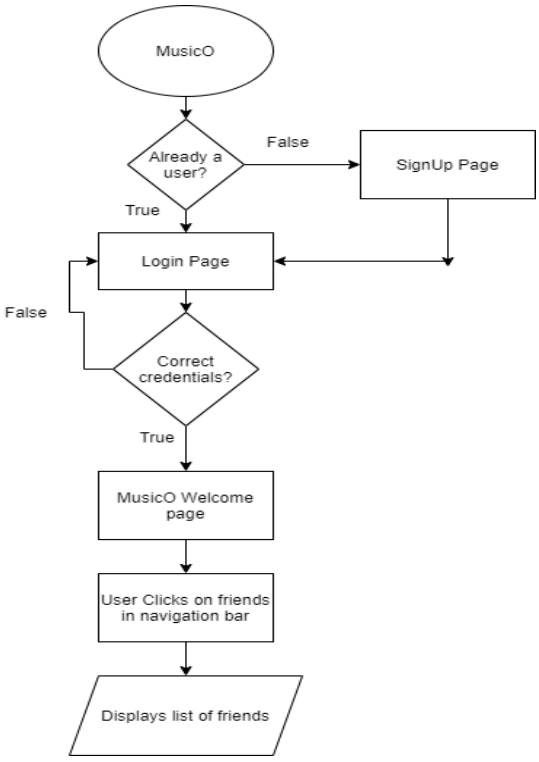

Figure 44: Task Flow Diagram - View list of friends of a user

Clickstream:

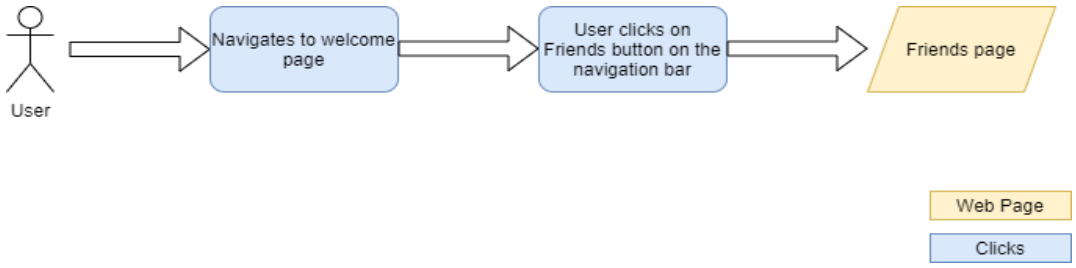

Figure 45: Clickstream Diagram - View list of friends of a user

Visit a friend's profile

1. User opens the MusicO.
2. The system displays the homepage/Login page.
3. User enters their username and password and hits the login button.
    a. System displays an "incorrect username/password" message.
    b. System prompts the user to try again.
    c. Users enter the username and password again and hit the login button.
    d. System displays a "Incorrect username/password" message.
    e. System prompts the user to reset their password and ask for their email address.
    f. User types the email address and hits enter.
    g. System prompts the user to enter the authentication code they received via email.
    h. User types the authentication code and hits enter.
    i. System prompts the user to enter a new password.
    j. User enters the new password and hits enter.
    k. System updates the password and re-directs them to the login page.
4. User logs in into the music app with correct credentials.
5. System displays the Welcome page.
6. User clicks on Friends.
7. System displays the user's friend list.
8. User sees his/her friends.
9. User clicks on a drop-down option besides the friend's name.
10. User selects the view profile option.
11. System opens his friend's profile page.
12. User sees the profile of his/her friend.

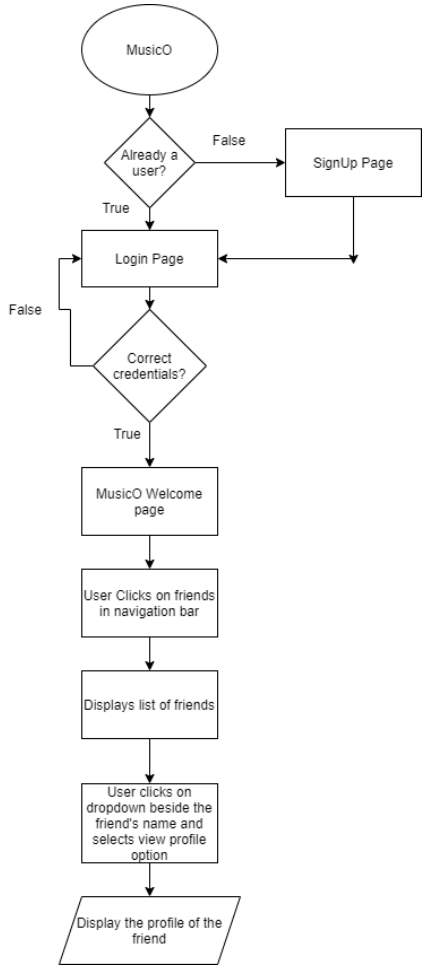

Figure 46: Task Flow Diagram – Visit a friend's profile

Clickstream:

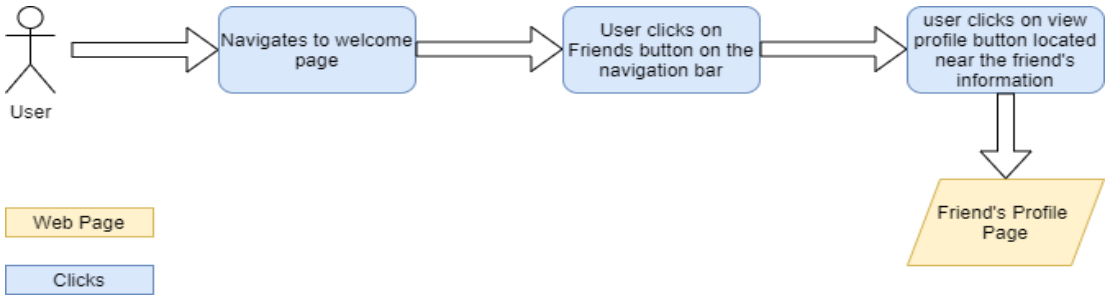

Figure 47: Clickstream Diagram – Visit a friend's profile

Display friend's playlist

1. User opens the MusicO.
2. The system displays the homepage/login page.
3. User enters their username and password and hits the login button.
    a. System displays an "incorrect username/password" message.
    b. System prompts the user to try again.

c. Users enter the username and password again and hit the login button.
d. System displays an “incorrect username/password” message.
e. System prompts the user to reset their password and ask for their email address.
f. User types the email address and hits enter.
g. System prompts the user to enter the authentication code they received via email.
h. User types the authentication code and hits enter.
i. System prompts the user to enter a new password.
j. User enters the new password and hits enter.
k. System updates the password and re-directs them to the login page.

4. User logs in into the music app with correct credentials.
5. System displays the Welcome page.
6. User clicks on Friends.
7. System displays the user’s friend list.
8. User sees his/her friends.
9. User clicks on a dropdown option besides the friend's name.
10. User selects the view profile option.
11. System opens his friend’s profile page.
12. User sees the profile of his/her friend.
13. System shows all the playlists of that friend.
14. User sees all the playlists of that particular friend.

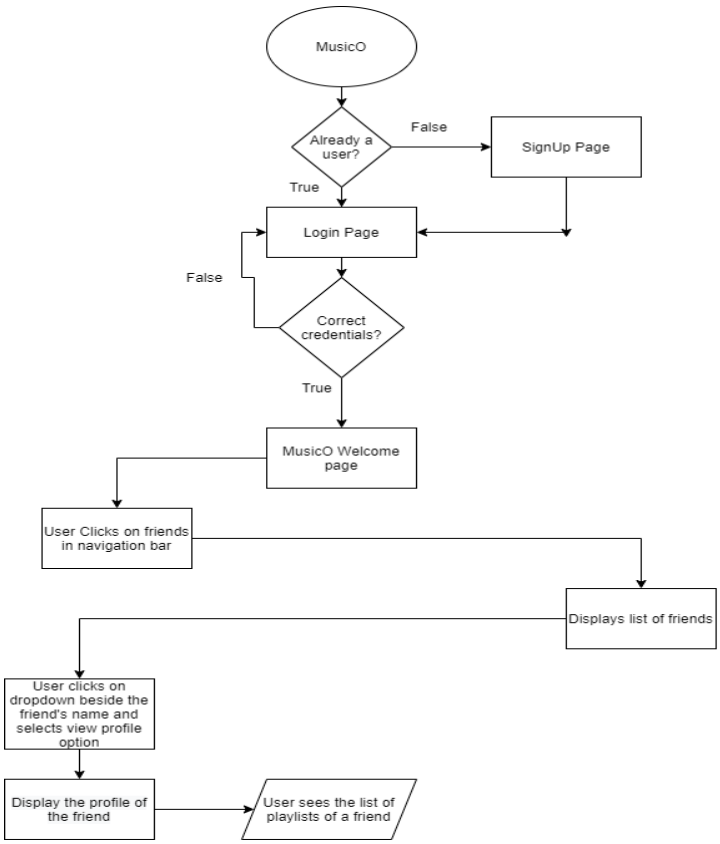

Figure 48: Task Flow Diagram - View friend’s playlist

Follow a friend

1. User opens the musicO.
2. The system displays the homepage/login page.
3. User enters their username and password and hits the login button.
    a. System displays an “incorrect username/password” message.
    b. System prompts the user to try again.
    c. User enters the username and password again and hits the login button.
    d. System displays an “incorrect username/password” message.
    e. System prompts users to reset their password and ask for their email address.
    f. User types the email address and hits enter.
    g. System prompts the user to enter the authentication code they received via email.
    h. User types the authentication code and hits enter.
    i. System prompts the user to enter a new password.
    j. User enters the new password and hits enter.
    k. System updates the password and re-directs them to the login page.
4. User logs in into the music app with correct credentials.
5. System displays the Welcome page.
6. User clicks on the search button.
7. System prompts a field for input.
8. User types friend’s name.
9. System searches returns a list of names matching the keyword.
10. User finds their friends and clicks on the button besides their friend’s name to follow them.

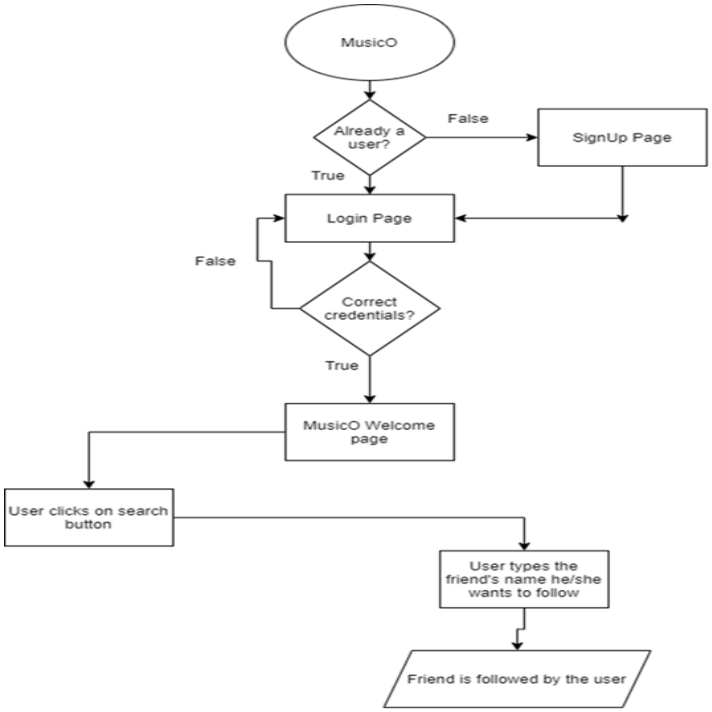

Figure 49: Task Flow Diagram - Follow a friend

Unfollow a friend

1. User opens the MusicO.
2. The system displays the homepage/login page.
3. User enters their username and password and hits the login button.
    a. System displays an “incorrect username/password” message.
    b. System prompts the user to try again.
    c. User enters the username and password again and hits the login button.
    d. System displays an “incorrect username/password” message.
    e. System prompts users to reset their password and ask for their email address.
    f. User types the email address and hits enter.
    g. System prompts the user to enter the authentication code they received via email.
    h. User types the authentication code and hits enter.
    i. System prompts the user to enter a new password.
    j. User enters the new password and hits enter.
    k. System updates the password and re-directs them to the login page.
4. User logs in into the music app with correct credentials.
5. System displays the Welcome page.
6. User clicks on Friends.
7. System displays the user’s friend list.
8. User sees his/her friends.
9. User clicks on a drop-down list besides the friend's name.
10. System gives a drop-down option of view profile and unfollow.
11. User selects the unfollow option.
12. System unfollows the friend for that user.

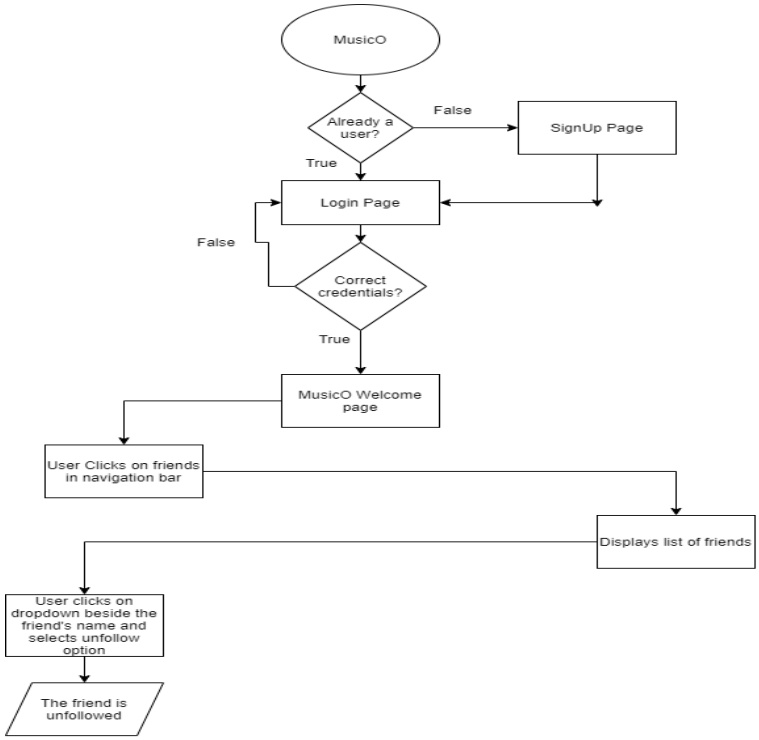

Figure 50: Task Flow Diagram - Unfollow a friend

4.2 Process and Service Workflow

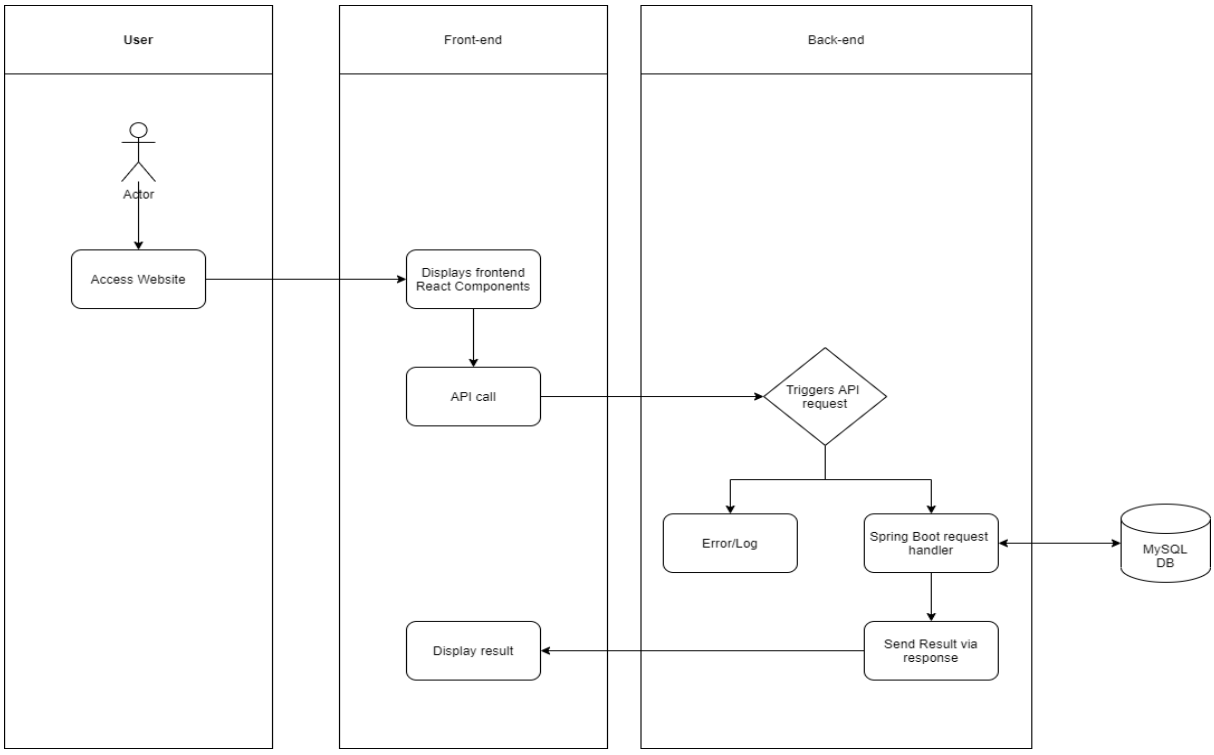

Figure 51: Workflow Diagram

5 FUTURE WORK

In the future, we would use React Redux or any similar state management library which would help in maintaining the state of the music player across the website. We would also like to complete the three missing features in our application right now, which are, displaying lyrics, recommending songs, and letting users upload their own tracks. To display lyrics, we would need to look for a viable dataset. To let users, upload their own songs, we would need to provide a form to upload artist and album details, upload the mp3 file which is stored in an S3 bucket and a place for users to upload their own lyrics. This would mean that not every track might contain lyrics. To recommend songs to the user, we would capture users' listening and browsing history and train a model to recommend similar songs and albums.

6 CONCLUSION

We have developed an application which streams music for free. Users can search for songs, albums, artists or other users in the application. Users can add any song on the application to a newly created or an existing playlist. Users can also follow other users and view their playlists. These playlists can also be imported to the user's own account. All these features are available as soon as the user registers on the application.

7 RECOMMENDATIONS

We would recommend having knowledge of ReactJS and SpringBoot when working on this application in the future. Getting familiar with the react-mp3-player library is also crucial as this library is used in most of the pages in the application [9]. Most of the application's CSS is built using Bootstrap library, so having knowledge of Bootstrap is recommended [8]. While building a new webpage, we would recommend reusing the CSS files for maintaining a uniform design across the application.

8 REFERENCES

[1] "Adobe Spark | Easily Create Graphics, Web Pages, Videos & More", Adobe.com, 2021. [Online]. Available: https://www.adobe.com/in/products/spark.html.[Accessed: 01- Aug- 2021].

[2] "Flowchart Maker & Online Diagram Software", App.diagrams.net, 2021. [Online]. Available: https://app.diagrams.net/. [Accessed: 01- Aug- 2021].

[3] "UXPin - UI Design and Prototyping Tool", Uxpin.com, 2021. [Online]. Available: https://www.uxpin.com/. [Accessed: 01- Aug- 2021].

[4] "Balsamiq Wireframes - Industry Standard Low-Fidelity Wireframing Software | Balsamiq", Balsamiq.com, 2021. [Online]. Available: https://balsamiq.com/wireframes/. [Accessed: 01- Aug-2021].

[5] "Photo by nappy from Pexels " [Online]. Available: https://www.pexels.com/photo/man-wearing-blue-blazer-936043/. [Accessed: 01- Aug- 2021].

[6] "React-Bootstrap", React-bootstrap.github.io, 2021. [Online]. Available: https://react-bootstrap.github.io. [Accessed: 01-Aug-2021].

[7] Material-ui.com. 2021. React Grid component - Material-UI. [Online] Available at: https://material-ui.com/components/grid/. [Accessed: 01-Aug-2021].

[8] "Bootstrap 3 Tutorial", W3schools.com, 2021. [Online]. Available: https://www.w3schools.com/bootstrap/. [Accessed: 01-Aug-2021].

[9] "react-mp3-player", npm, 2021. [Online]. Available: https://www.npmjs.com/package/react-mp3-player. [Accessed: 01-Aug-2021].

[10] Simar Saggu,"CSCI5709ASSIGNMENT2."Dalhousie University, [Online document],2021. [Accessed: 01-Aug-2021].

[11] Neehar Parupalli Ramakrishna,"CSCI5709ASSIGNMENT2."Dalhousie University, [Online document],2021. [Accessed: 01-Aug-2021].

[12] Venkata Kanakayya Prashant Vadlamani,"CSCI5709ASSIGNMENT2."Dalhousie University, [Online document],2021. [Accessed: 01-Aug-2021].

[13] Fenil Nikeshkumar Shah,"CSCI5709ASSIGNMENT2."Dalhousie University, [Online document],2021. [Accessed: 01-Aug-2021].

[14] Anuj Upadhyay,"CSCI5709ASSIGNMENT2."Dalhousie University, [Online document],2021. [Accessed: 01-Aug-2021].

[15] Utkarsh Nileshbhai Patel,"CSCI5709ASSIGNMENT2."Dalhousie University, [Online document],2021. [Accessed: 01-Aug-2021].

[16] MusicO,"CSCI5709GROUP1PROEJCTPROPOSAL."Dalhousie University, [Online document],2021. [Accessed: 01-Aug-2021].