Playlister

Product Specification

Author: **Richard McKenna**
Fall 2022

Table of Contents

1. Introduction

This document describes the specification for Playlister, an application for creating and playing playlists of YouTube music videos. The site will allow users to create, edit, and play playlists as well as share playlists so that others may then play and comment on them. The site will include a built-in YouTube player and will allow users to find others' playlists via multiple search criteria.

Platform

The application will be a full-stack MERN Web application, so it will use Mongo, Express, React, and Node as well as the YouTube API and various Node libraries. The existing Playlister prototype (i.e. HW 4) should serve as the basis for this project. Note, this document provides the application's requirements, the next development step should be application design documents, which should serve as the code blueprints to be followed during implementation.

2. Use Cases

Below is a list of all the ways our users will interact with our system. Note that once a playlist is "published" it cannot be edited (since other users can then Like/Dislike it). It can be deleted and duplicated (a duplicated playlist can be edited). One user cannot have two lists with the same name. Once one publishes a list it is findable and viewable by other users.

Actors - the Use Case diagrams below refer to the following types of users:

- **Guest** - a user navigating the site who does not have an existing account
- **Registered User** - a user who has already created an account
- **Logged-In User** - a user who has an existing account and is logged in

Use Case Number	Use Case
2.1	Create Account
2.2	Login to Account
2.3	Logout of Account
2.4	Use Site as Guest
2.5	Create Playlist
2.6	Rename Playlist
2.7	Edit Playlist
2.8	Publish Playlist
2.9	Duplicate Playlist
2.10	Delete Playlist
2.11	View Owned Playlists
2.12	Search by Playlist
2.13	Search by User
2.14	Play Playlist
2.15	Comment on Playlist

Use-Case Number:	2.1
Use-Case Name:	Create Account
Actors:	Guest
Preconditions:	The user has a working Internet connection, has loaded the application page, and is not currently logged in.
Postconditions:	A new account is created for the user, unique to their email address, with a secure password.
Story:	The user arrives at the splash screen and would like to start making playlists and so clicks on the “Create Account” button. The user can then enter their user name, first name, last name, email, and password in the provided text fields and press the Create Account button. This will add the account to the database but will not log the user in. The user will then need to do that separately.
Scenario:	Joe Shmo is a user who wishes to make a playlist. Joe goes to the site’s splash screen and clicks on “Create New Account”. This brings up the Create Account dialog where he enters “Joe Shmo” in the name field, joe@shmo.com in the email field, and JoeShmo123 in the password fields. He then clicks on the Create Account button and a new account is created and he is then taken to the login screen where he may now login.
Exceptions:	No two users can create an account using the same email address. Should one try to create an account using an email that is already in use the application should provide appropriate, nicely styled feedback. The same is true if improper passwords are provided. The same is true of user name, All user names must be unique.

Use-Case Number:	2.2
Use-Case Name:	Login to Account
Actors:	Registered User
Preconditions:	The user has
Postconditions:	The user is logged in and forwarded to their home screen, where they may view their owned playlists.
Story:	The user navigates to the login page, either through the splash screen or via the drop-down menu. There, the user enters a registered email address and proper password and clicks on the login button. The user is then logged in and bright to their home screen, where they may view their owned playlists.
Scenario:	Joe Shmo is on the site splash screen and clicks on the Login button. Joe Shmo enters joe@shmo.com in the email text field and JoeShmo123 in the password field. Since that email/password combo is correct, it is verified and Joe Shmo is logged in, forwarding him to his home page, where he can view the 5 playlists he’s already created.
Exceptions:	Should the user provide an incorrect email or password, the application must give suitable feedback to the user.

Use-Case Number:	2.3
Use-Case Name:	Logout of Account
Actors:	Logged-In User
Preconditions:	The user is logged-in and on any application screen.
Postconditions:	The user is logged-out and returned to the splash screen.
Story:	The user is logged in and making use of site services. The user moves the mouse to the top-right and clicks on the account circle, which opens the account drop-down menu. The user then clicks on the Logout menu item. This logs the user out of the site and returns them to the splash screen.
Scenario:	Joe Shmo is logged into the site and is playing one of his playlists. Joe Shmo decides to leave the site and wants to logout. He moves the mouse up to the top right and clicks on the account circle reading “JS”. This opens a drop-down menu which includes a “Logout” menu item. Joe Shmo clicks on the Logout menu item and is logged out. He is then automatically forwarded to the site splash screen.
Exceptions:	None. If logged-in, this should be accessible at all times from all pages.

Use-Case Number:	2.4
Use-Case Name:	Use Site as Guest
Actors:	Guest
Preconditions:	The user has a working Internet connection, has loaded the application page, and is not currently logged in.
Postconditions:	The user navigates the site and makes use of some provided services.
Story:	The user lands on the splash screen and clicks on the button reading “Continue as Guest”. This takes the user to the application’s All Lists screen, where the user can search for lists made by others by name and then open a found playlist, view the list, view the comments, and play the playlist in the app player. The user can also navigate to the Users screen, which lets them search for playlists by user name. The user cannot navigate to the home screen, so that button is disabled, as the user is a guest and owns no lists.
Scenario:	Jane arrives on the splash screen and wants to use the site but has no interest in making an account. Jane clicks on the “Continue as Guest” button, which brings Jane to the All Lists screen. There, Jane types “Dance Party” in the text field and hits enter, which brings up a list of playlists that contain the text “Dance Party”. Jane selects “Dance Party 101” and plays it, which begins playing in the app’s player. Jane reads through the comments, but cannot add her own as she is not logged in. After a time, Jane clicks on the Users button and it takes her to the Users screen where she enters “JoeShmo” and it pulls up all the playlists from users with login names containing “JoeShmo”.
Exceptions:	Foolproof design should be employed to make it clear that a guest user cannot go home, this button must be visibly and functionally disabled.

Use-Case Number:	2.5
Use-Case Name:	Create Playlist
Actors:	Logged-In User
Preconditions:	User is logged-in to a registered account.
Postconditions:	User has created a new playlist.
Story:	User is logged-in to the site. User clicks on Home button to navigate to Home Screen. User then clicks on New Playlist button, which creates a new playlist and opens it for editing. Note the playlist is given a default “Untitled ?” name, where ? is a numeric value that counts up starting a 0. Note, no two playlists owned by one user should have the same name. So, if a playlist owned by the user is titled “Untitled 0”, that title can never be assigned again, even through renaming a list.
Scenario:	Joe Shmo is logged into the site and clicks on the Home button, which takes him to his home screen. There, his owned playlists are listed in the order selected by the SORT BY control. Joe clicks on the New Playlist button, which creates a playlist titled “Untitled ?”, where ? is a number that ensures it is uniquely named for Joe Shmo’s collection. The playlist is opened for editing.
Exceptions:	Site must make sure the generated playlist name is unique for that user.

Use-Case Number:	2.6
Use-Case Name:	Rename Playlist
Actors:	Logged-In User
Preconditions:	User is logged-in to a registered account and has at least one owned playlist.
Postconditions:	User has renamed an existing playlist.
Story:	The user is logged in and clicks on the Home button to navigate to the home screen. The user wishes to change the name of a list and so double clicks on one of the playlist cards. This turns the playlist name text into a textfield, initially with the name of the playlist. The user can then type in any name and hit enter to change the name.
Scenario:	Joe Shmo has a registered account and is logged-in. Joe clicks on the Home button, which takes him to the home screen, where all the playlists he owns are listed, ordered according to the selected SORT BY criteria. Joe changes the sorting to Sort By Name, which reorders them. Joe sees one list titled “Untitled 5”. Joe double clicks on that card and a text field appears in place of the text. The text field contains “Untitled 5”. Joe Shmo changes the text field text to read “My Workout Songs” and hits enter. The playlist is updated in the database and in the display and the lists are resorted accordingly. The text field should then be closed.
Exceptions:	Should the user enter a name that is already in use by this user, a popup dialog should open warning the user and the name should not change, the text field should remain open.

Use-Case Number:	2.7
Use-Case Name:	Edit Playlist
Actors:	Logged-In User
Preconditions:	The user is logged-in and has at least one existing playlist.
Postconditions:	The user has updated the songs in the playlist.
Story:	The user navigates to their Home Screen, where they find the playlists they own. The user selects an unpublished playlist, which opens it for editing. During editing, the user clicks on the Add Song button, which adds songs, double clicks on the song, which opens a dialog where the user may change song title, artist, and YouTubeId values. The user can also click on the remove button on each song to remove it from the playlist. The user can also drag and drop the songs to change their order. Finally, the user can press the undo and redo buttons to undo mistakes and redo to reaffirm them during editing.
Scenario:	Joe Shmo goes to his Home Screen and clicks on the “My Workout Songs” playlist, which already has 10 songs. Joe Shmo drags the third song to the first one, to change order. Joe then clicks on the remove song button for the 8th song, and via a modal, approves the action. Joe double-clicks on the 2nd song, opening a modal, and changes the title, artist name, and youTubeId and clicks approve. Joe continues to do such actions until his playlist is complete.
Exceptions:	Note that Undo and Redo must be available during editing and the transaction stack must be reset each time a playlist is selected.

Use-Case Number:	2.8
Use-Case Name:	Publish Playlist
Actors:	Logged-In User
Preconditions:	The user is logged-in.
Postconditions:	The user has a playlist that is published and therefore findable and playable by other users.
Story:	The user navigates to their Home Screen, where the make a new playlist, and then edit it. Once all the editing is complete (per the Edit Playlist use case), the user wishes to make it publicly available. To do this, the user presses the “Publish” button. This permanently publishes the playlist, which cannot be undone.
Scenario:	Joe Shmo has a playlist called “My Workout Songs” that he made and edited and is happy with it and wishes to share it with the world. Joe Shmo goes to his home screen and finds this playlist, which is currently unpublished, and clicks on it to open it. Joe Shmo reviews the list one last time and then clicks the Publish button, which changes the appearance of the playlist to denote it has been published, and removes the Publish button. It is now a playlist others can play.
Exceptions:	A published playlist can be deleted. Also not it can be forked as well, so an owner can duplicate a playlist before deleting it if they are unhappy with their published work.

Use-Case Number:	2.9
Use-Case Name:	Duplicate Playlist
Actors:	Logged-In User
Preconditions:	User has logged into an existing account
Postconditions:	User has created a new playlist that can be edited with all the content of an existing playlist
Story:	The user is logged in and navigates to either their own Home screen or the All Lists or Users screen. There, the user finds a playlist that they wish to duplicate and edit. They select the playlist of interest and click on the duplicate button. This adds a new playlist to the user's own list with the same name and all the same content as the original.
Scenario:	Joe Shmo is looking at lists made by others on the All Lists screen and finds a list titled "Sad Songs", he likes but is missing a few songs he'd like to add. Joe clicks on the duplicate button on the playlist and it makes a new list in his own home screen also titled "Sad Songs" as he had no list with that name. This new list is unpublished so Joe can then edit it and publish it when it is completed if he wishes others to find and use it.
Exceptions:	Should the user duplicating a list already have a list with that name, a number should be added to the list so that it remains uniquely named.

Use-Case Number:	2.10
Use-Case Name:	Delete Playlist
Actors:	Logged-In User
Preconditions:	The user is logged in and has at least one owned playlist
Postconditions:	The affected playlist is removed from the database and the user interface is updated so it can no longer be found.
Story:	The user is logged in and navigates to their Home screen where they find a list of all their owned playlists. The user wishes to delete one of them and so clicks on the delete playlist button. This opens a modal that asks to confirm or cancel this action. The user clicks to confirm and the list is deleted from the database and the user interface is updated to reflect this change.
Scenario:	Joe Shmo is logged in and on his home screen. There he finds his “Stupid Songs” playlist, which he realizes was a mistake. Joe clicks on the delete playlist button, this opens a dialog asking to confirm this action. Joe clicks on the confirm button, which deletes the playlist from the database, and is returned to his home screen, where the playlist is no longer listed.
Exceptions:	The user can press cancel to close the dialog and not delete the list. Note, once a list is deleted, it is permanent, it cannot be undone.

Use-Case Number:	2.11
Use-Case Name:	View Owned Playlists
Actors:	Logged-In User
Preconditions:	User has an account and is logged in
Postconditions:	User gets to view the playlists they own
Story:	User is logged in and navigating the site. User clicks on the Home button and is brought to their Home screen, where they may see all their owned playlists. The user clicks on the SORT BY button to select sorting criteria for listing the playlists. The choices are “By Creation Date (Old-New)”, “By Last Edit Date (New-Old)”, and “By Name (A-Z)”. The user selects a sort criteria and finds a list of interest. The user then selects the list and can play it.
Scenario:	Joe Shmo is logged in and goes to his home page. He sorts his lists by name and clicks on the “My Workout Songs” list to open it. He can then do things like play or view comments.
Exceptions:	The user may not own any playlists and so the home screen may be empty. Note, a user can access and use their own unpublished playlists.

Use-Case Number:	2.12
Use-Case Name:	Search by Playlist
Actors:	Registered User or Guest
Preconditions:	The user has an internet connection and has navigated to the site.
Postconditions:	The user finds playlists published by others and can use them.
Story:	User is on the site and navigates to the All Lists screen. On this screen the user enters text in the search text field and hits enter. This loads all the published playlists made by any user that contain the text entered in the textfield. The user can then select and play those lists.
Scenario:	Jane Doe has an account and logs-in. She then goes to the All Lists screen. In the search text field she types “Workout” and hits enter. A list of many playlists is then displayed, all containing the text “Workout”. Jane scrolls down and finds one titled “My Workout Songs”, owned by JoeShmo. Jane opens it up for use.
Exceptions:	Should one type-in text that is not found in any playlists, the list of playlists on the All Lists screen should simply be empty. Note, if one does not type any text and hits enter, no playlists should appear.

Use-Case Number:	2.13
Use-Case Name:	Search by User
Actors:	Registered User or Guest
Preconditions:	The user has an internet connection and has navigated to the site.
Postconditions:	The user finds playlists published by others and can use them.
Story:	User is on the site and navigates to the Users screen. On this screen the user enters text in the search text field and hits enter. This loads all the published playlists made by any user whose user name contains the text entered in the textfield. The user can then select and play those lists.
Scenario:	Jane Doe has an account and logs-in. She then goes to the Users screen. In the search text field she types “Shmo” and hits enter. A list of many playlists is then displayed, all owned by users whose user name contains the text “Shmo”. Jane scrolls down and finds one titled “My Workout Songs”, owned by JoeShmo. Jane opens it up for use.
Exceptions:	Should one type-in text that is not found in any user name, the list of playlists on the Users screen should simply be empty. Note, if one does not type any text and hits enter, no playlists should appear.

Use-Case Number:	2.14
Use-Case Name:	Play Playlist
Actors:	Registered User or Guest
Preconditions:	The user has found a playlist they wish to listen to.
Postconditions:	The user plays the playlist in the manner they desire.
Story:	The user is on the site and has searched for a playlist using one of the three screens. Upon finding the desired playlist they click on it, which highlights it and loads it into the player. They can then press the play button which will play the first song. They can press pause to pause the song, they can press next or prev to navigate to the song in the playlist they desire. Note that the song being played is highlighted in the playlist view.
Scenario:	Joe Shmo has found his own “My Workout Songs” playlist. Joe clicks on the playlist, which loads it into the player. Joe then clicks the Play button to play it but then decides he wishes to hear another song in the playlist, so presses Pause, and then Next and Previous until finding the song he wishes to hear. He then presses Play and the playlist then plays continuously, one song after the other until the end of the playlist.
Exceptions:	All published playlists should be playable by all users. Unpublished playlists can only be played by their owners, as they are the only ones who can find them.

Use-Case Number:	2.15
Use-Case Name:	Comment on Playlist
Actors:	Logged-In User
Preconditions:	The user is logged-in and has found a playlist they wish to interact with.
Postconditions:	The user has liked/disliked a published playlist and has left comments.
Story:	The user is on the site and has searched for a playlist using one of the three screens. Upon finding the desired playlist they click on it, which highlights it and loads it into the player. The user then clicks on the Comments tab, which shows the comments for that video. The user can press Like/Dislike and can also add their own comment by entering text in the comment text field and hitting enter.
Scenario:	Jane Doe has found Joe Shmo's "My Workout Songs" playlist. Jane clicks on the playlist, which loads it into the player. Jane then clicks the Comments tab and reads through the comments. Jane likes the playlist so she presses the "Like" button, which increments the Like number. Jane then leaves a comment "My Favorite!", which updates the comments list for all those viewing the playlist.
Exceptions:	Comments on a playlist should update everyone's comments viewing that playlist, but not it only need be loaded upon demand. In other words, if a user is viewing comments and someone else adds a comment, it would update the next time the user requested the comments, i.e. the next time they load the playlist, or if they enter their own comments and thus retrieve the comments again.

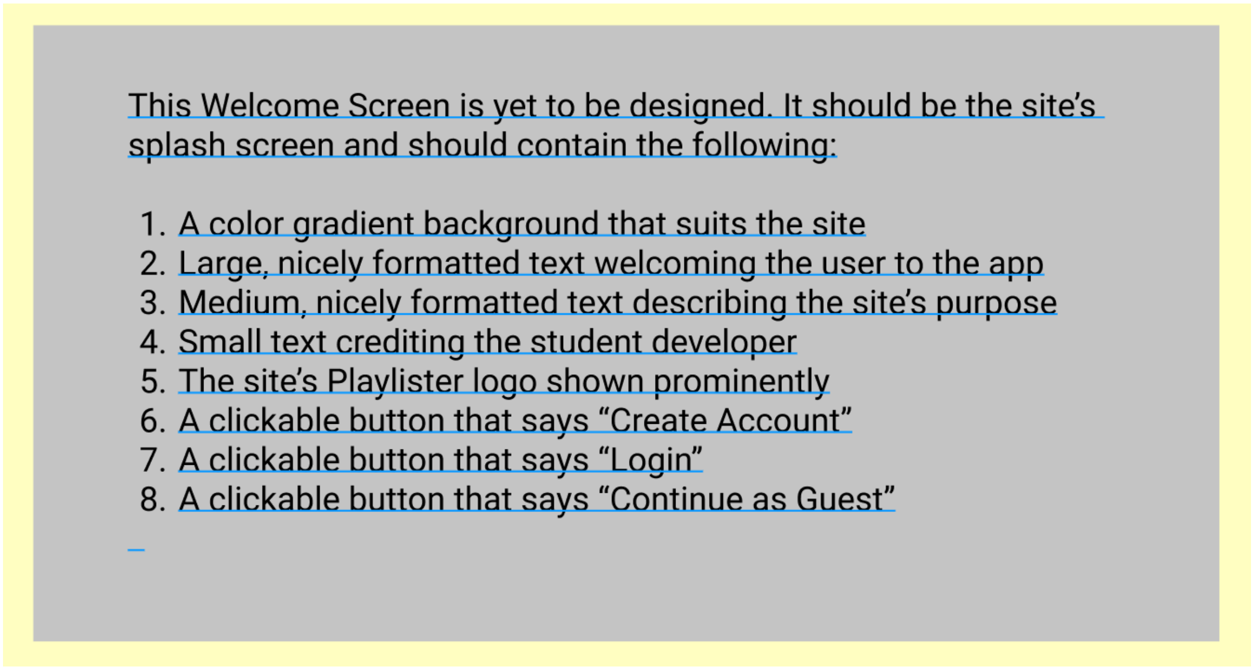
3. User Interface Mockup Diagrams

The application will have the seven user interface contexts/views and should be made to look approximately so. Note there is one exception, it is up to the developer to decide two things: how a landmark may be edited on the region viewer screen and how an existing parent region should be changed to another existing region. Here are the mockups:

- **Home Button** - when selected, searching should find all of the logged in user's lists that have list names that start with the search text
- **All Lists Button** - when selected, searching should find all of the lists created by any user that have list names the same as the search text
- **Users Button** - when selected, searching should find all of the lists created by a user with the name searched in the search text

View Number	Description
3.1	Welcome Screen View
3.2	Home Screen View Playing a Playlist
3.3	Home Screen viewing comments
3.4	Home Screen editing a playlist and verifying a song is usable
3.5	Edit Song Dialog
3.6	Remove Song Dialog
3.7	Delete List Dialog
3.8	All Lists View with Playlist Playing
3.9	User's Lists View With None Open
3.10	SORT BY Menu View
3.11	Account Menu View
3.12	Logout Menu View
3.13	Create Account View
3.14	Login View

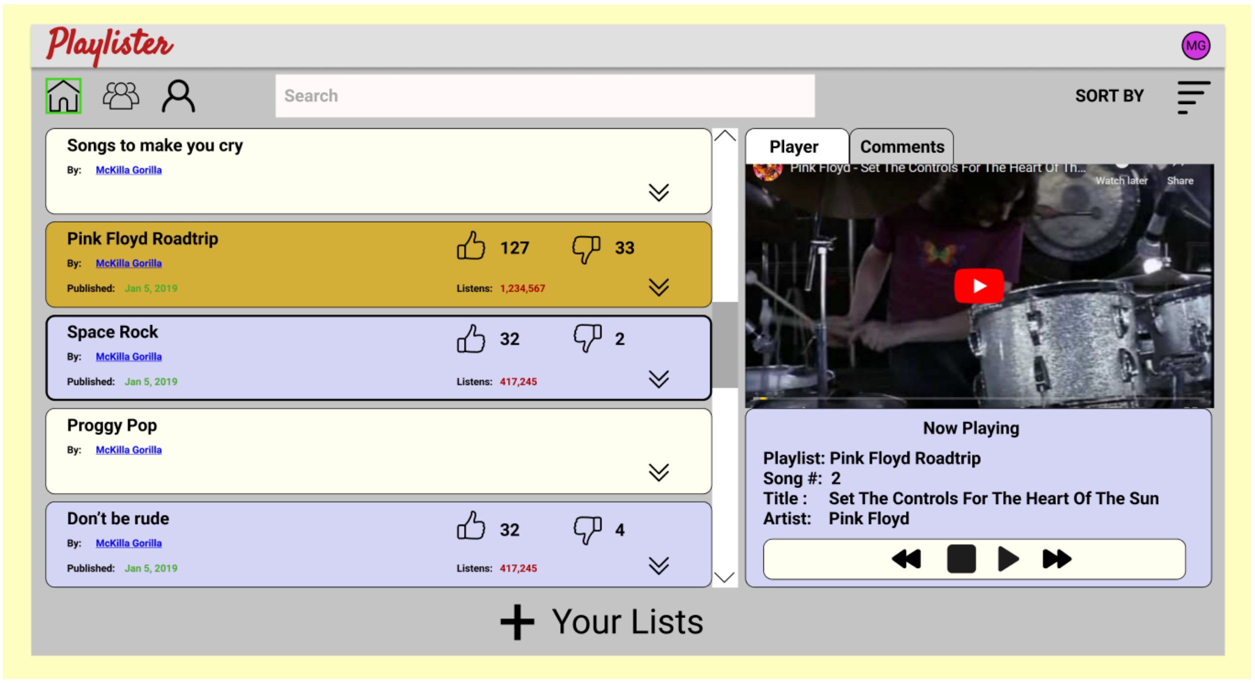
Figure 3.1: Welcome Screen View



This Welcome Screen is yet to be designed. It should be the site's splash screen and should contain the following:

1. A color gradient background that suits the site
2. Large, nicely formatted text welcoming the user to the app
3. Medium, nicely formatted text describing the site's purpose
4. Small text crediting the student developer
5. The site's Playlister logo shown prominently
6. A clickable button that says "Create Account"
7. A clickable button that says "Login"
8. A clickable button that says "Continue as Guest"

Figure 3.2: Home Screen View Playing a Playlist



Songs to make you cry
By: McKilla Gorilla

Pink Floyd Roadtrip 127 33
By: McKilla Gorilla
Published: Jan 5, 2019 Listens: 1,234,567

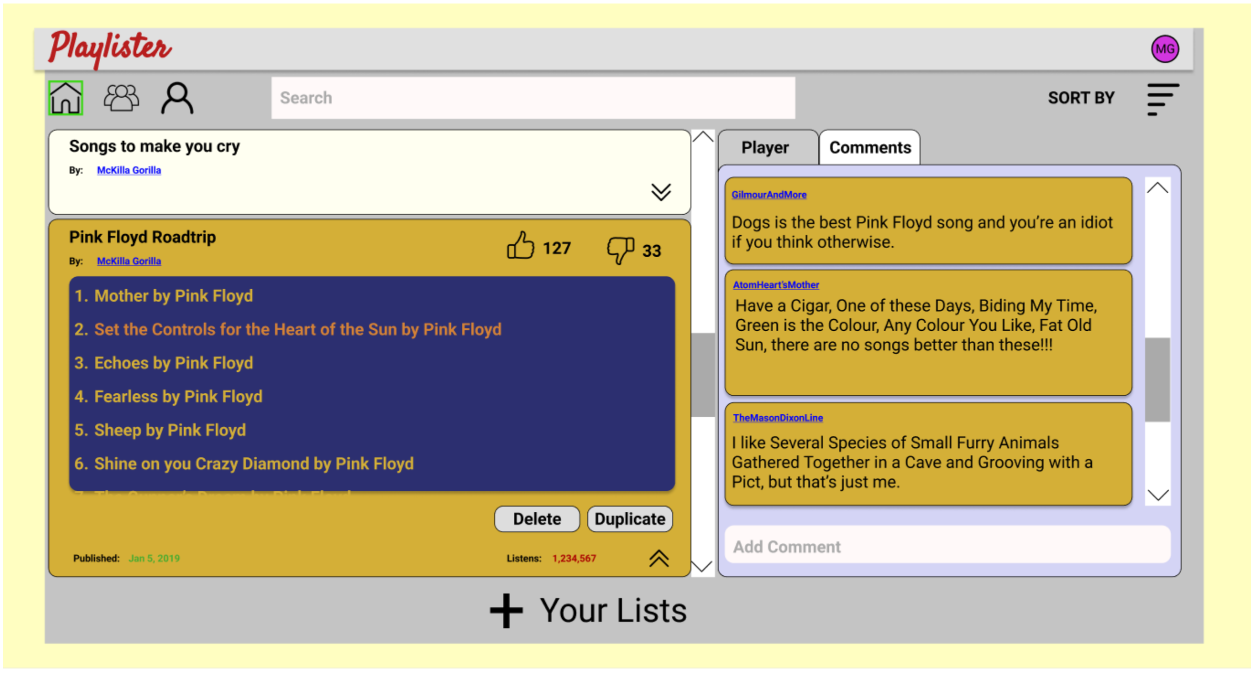
Space Rock 32 2
By: McKilla Gorilla
Published: Jan 5, 2019 Listens: 417,245

Proggy Pop
By: McKilla Gorilla

Don't be rude 32 4
By: McKilla Gorilla
Published: Jan 5, 2019 Listens: 417,245

Now Playing
Playlist: Pink Floyd Roadtrip
Song #: 2
Title : Set The Controls For The Heart Of The Sun
Artist: Pink Floyd

Figure 3.3: Home Screen viewing comments



Songs to make you cry
By: McKilla Gorilla

Pink Floyd Roadtrip
By: McKilla Gorilla
127 33

1. Mother by Pink Floyd
2. Set the Controls for the Heart of the Sun by Pink Floyd
3. Echoes by Pink Floyd
4. Fearless by Pink Floyd
5. Sheep by Pink Floyd
6. Shine on you Crazy Diamond by Pink Floyd

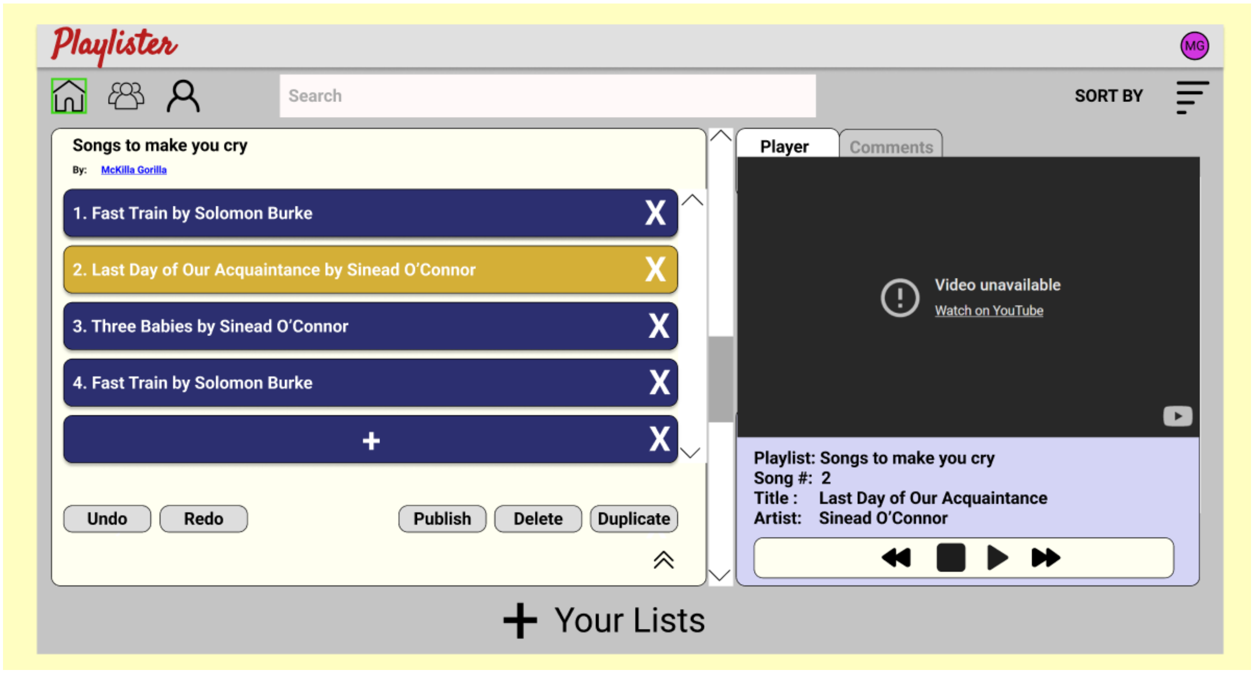
Published: Jan 5, 2019
Listens: 1,234,567

GilmourAndMore
Dogs is the best Pink Floyd song and you're an idiot if you think otherwise.

AtomHeart'sMother
Have a Cigar, One of these Days, Biding My Time, Green is the Colour, Any Colour You Like, Fat Old Sun, there are no songs better than these!!!

TheMasonDixonLine
I like Several Species of Small Furry Animals Gathered Together in a Cave and Grooving with a Pict, but that's just me.

Figure 3.4: Home Screen editing a playlist and verifying a song is usable

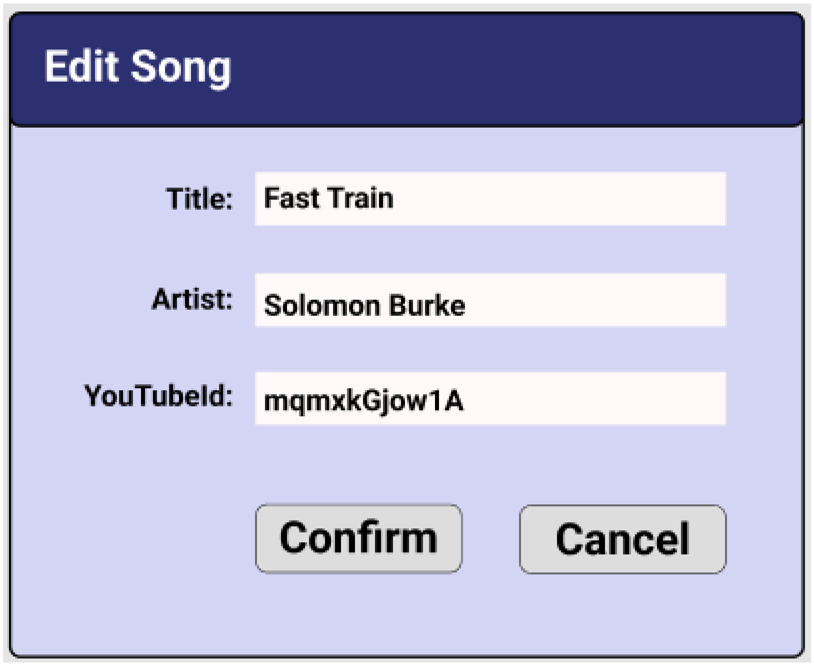


Songs to make you cry
By: McKilla Gorilla

1. Fast Train by Solomon Burke
2. Last Day of Our Acquaintance by Sinead O'Connor
3. Three Babies by Sinead O'Connor
4. Fast Train by Solomon Burke

Playlist: Songs to make you cry
Song #: 2
Title : Last Day of Our Acquaintance
Artist: Sinead O'Connor

Figure 3.5: Edit Song Dialog



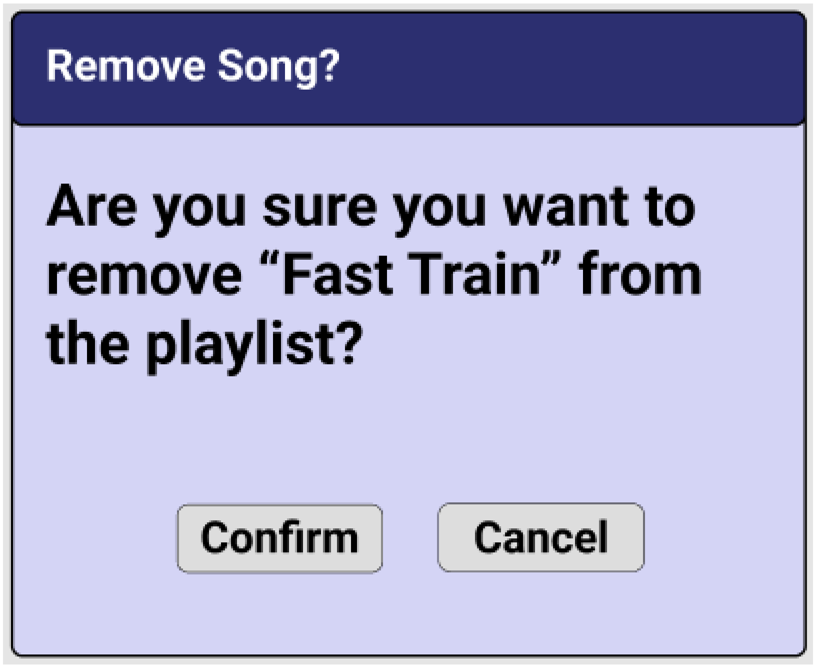
Edit Song

Title: Fast Train

Artist: Solomon Burke

YouTubeId: mqmxkGjow1A

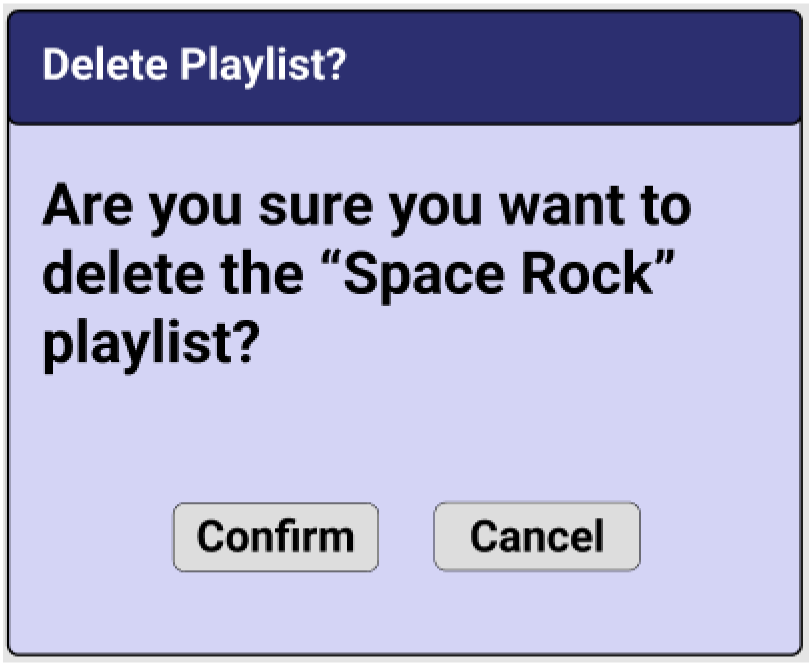
Figure 3.6: Remove Song Dialog



Remove Song?

Are you sure you want to remove “Fast Train” from the playlist?

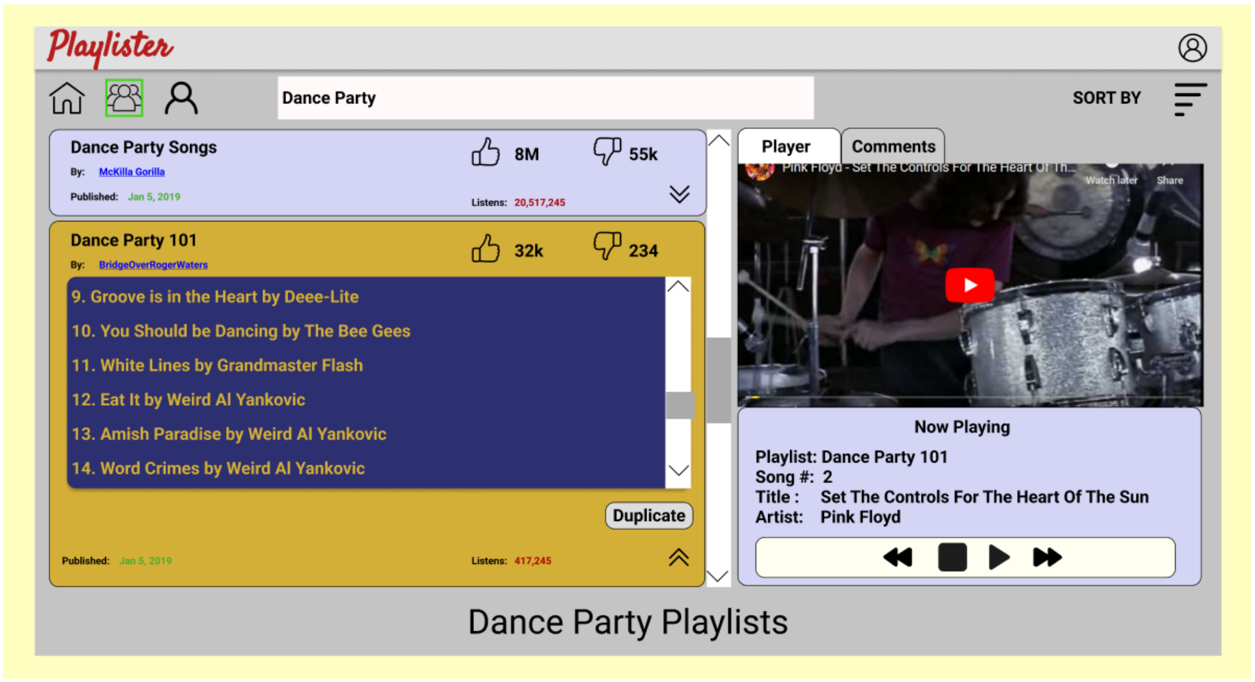
Figure 3.7: Delete List Dialog



Delete Playlist?

Are you sure you want to delete the “Space Rock” playlist?

Figure 3.8: All Lists View with Playlist Playing



Dance Party Songs
By: McKilla Gorilla
Published: Jan 5, 2019
8M
55k
Listens: 20,517,245

Dance Party 101
By: BridgeOverRogerWaters
32k
234

9. Groove is in the Heart by Deee-Lite
10. You Should be Dancing by The Bee Gees
11. White Lines by Grandmaster Flash
12. Eat It by Weird Al Yankovic
13. Amish Paradise by Weird Al Yankovic
14. Word Crimes by Weird Al Yankovic

Published: Jan 5, 2019
Listens: 417,245

Pink Floyd - Set The Controls For The Heart Of Th...

Now Playing
Playlist: Dance Party 101
Song #: 2
Title : Set The Controls For The Heart Of The Sun
Artist: Pink Floyd

Dance Party Playlists

Figure 3.9: User's Lists View With None Open

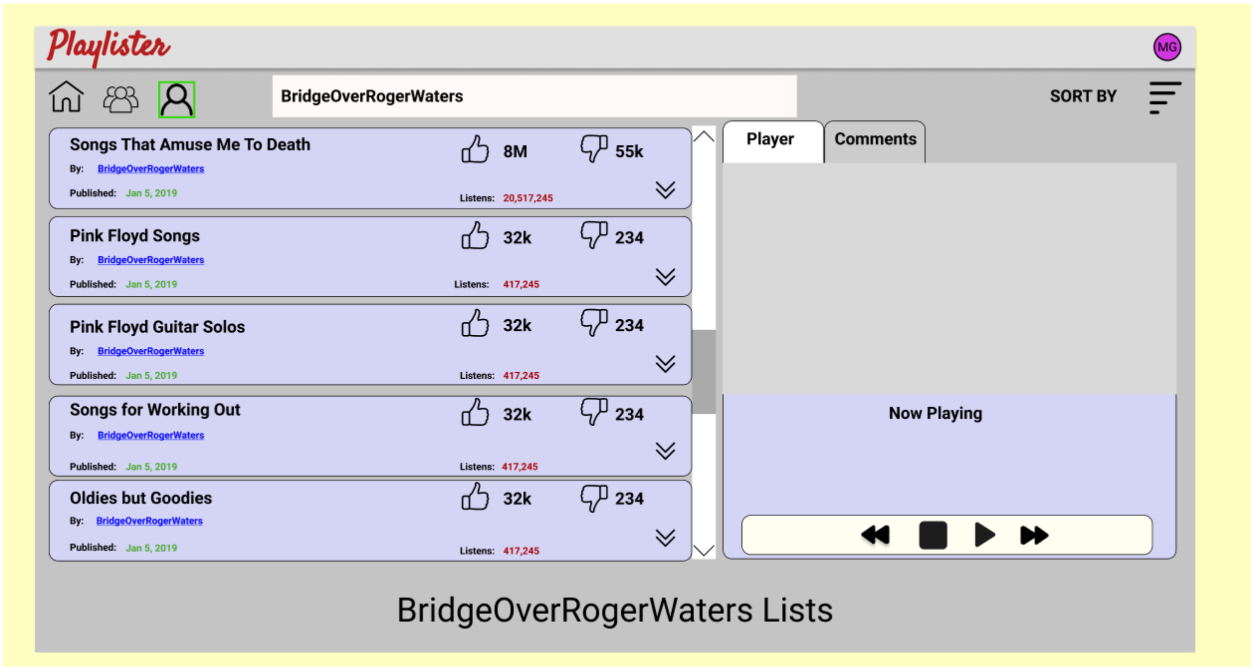


Figure 3.10: SORT BY Menu View

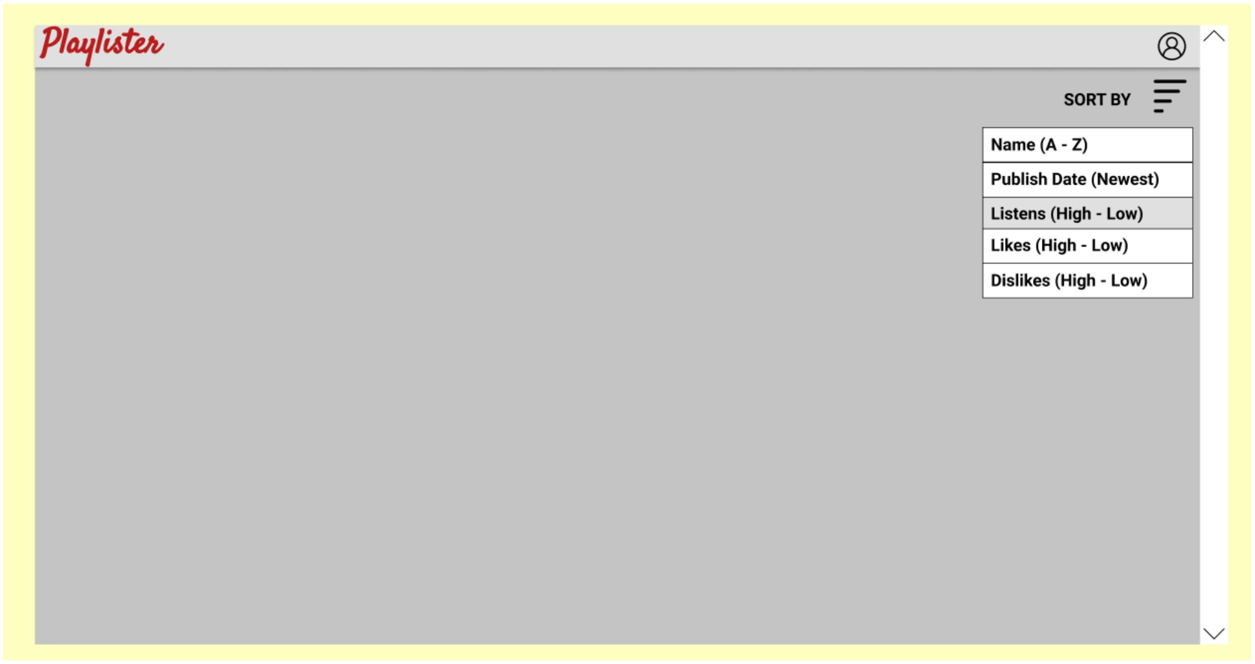


Figure 3.11: Account Menu View

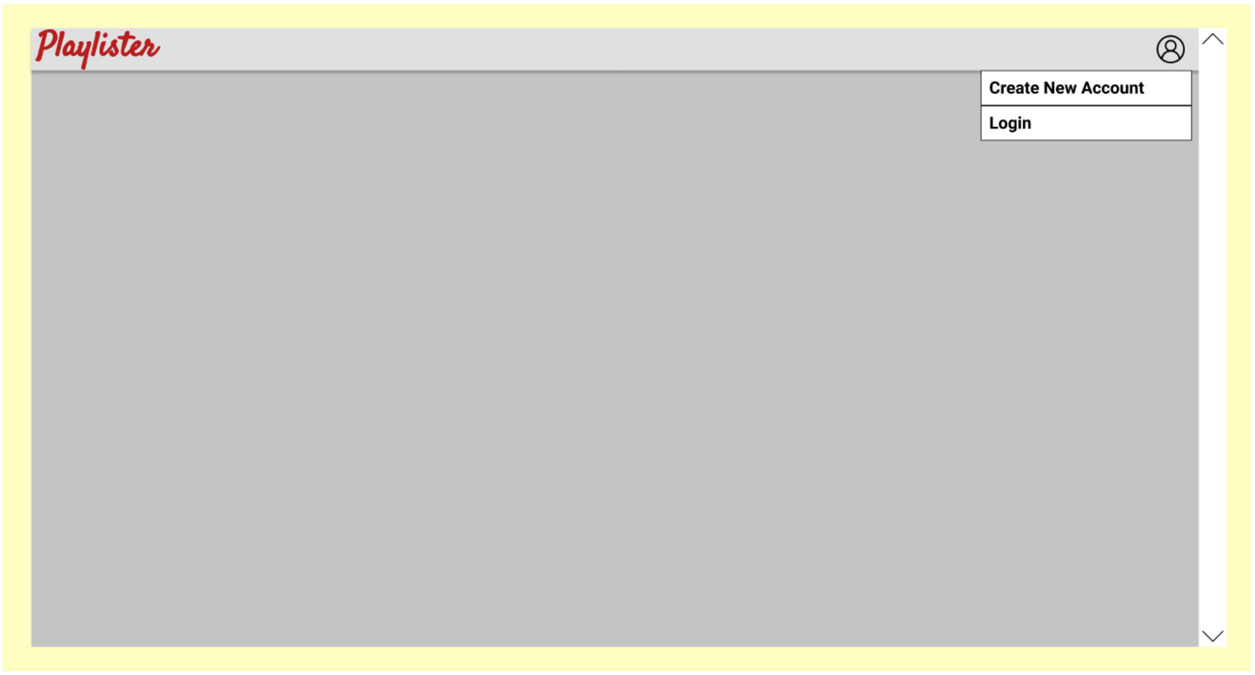


Figure 3.12: Logout Menu View

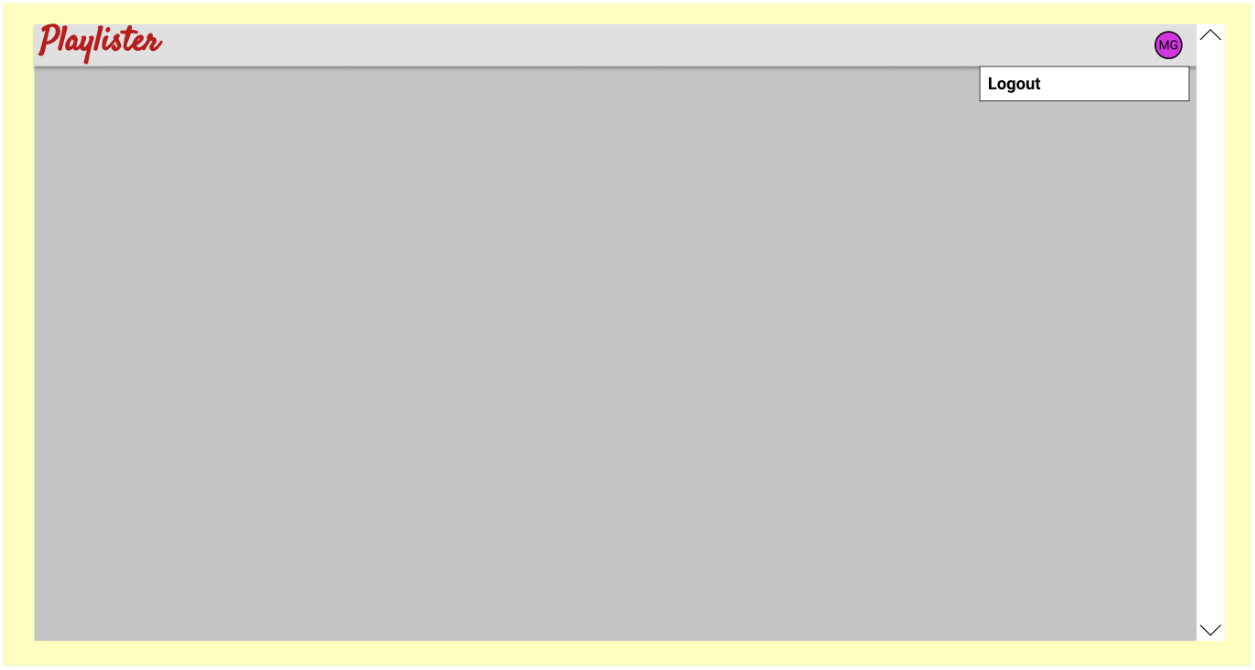


Figure 3.13: Create Account View

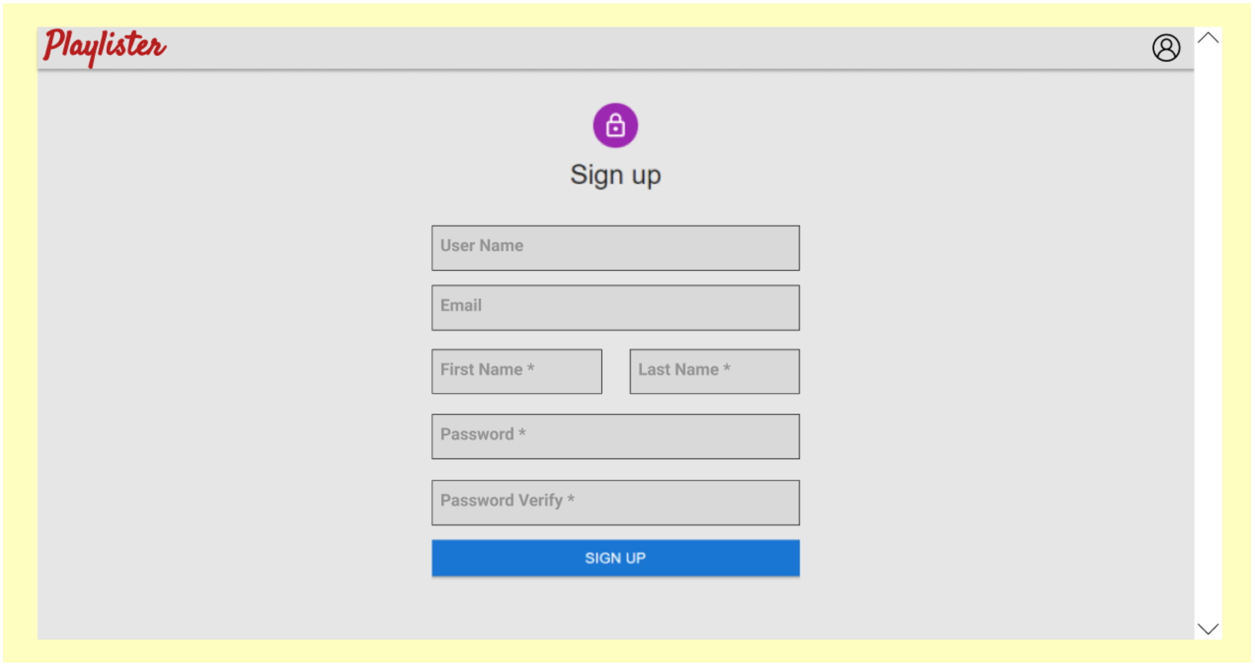
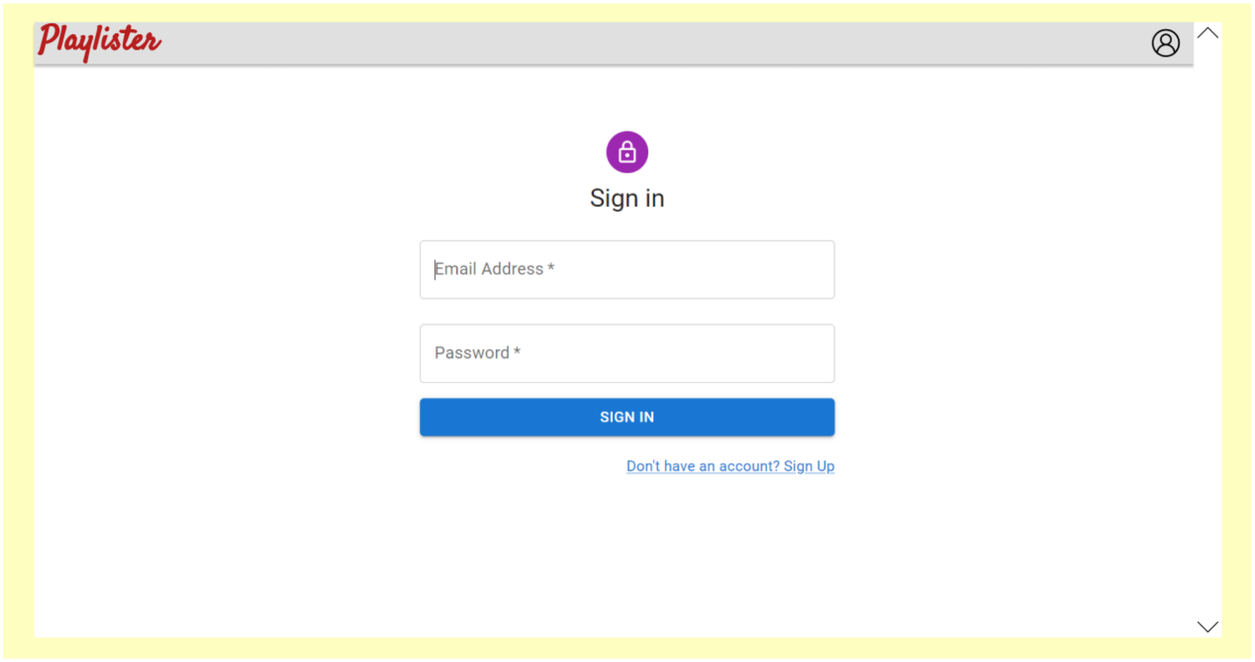


Figure 3.14: Login View


Playlister
Sign in
Email Address *
Password *
SIGN IN
Don't have an account? Sign Up