# Submission 5: Dynamic UI Mockups

SoundRoom Team

November 17, 2016

## 1 Outstanding Bugs

- Search feature for tracks hasn't been implemented yet

    - We have to implement the SoundCloud API (getting the access tokens, ids, figuring out how to search and stream tracks)
    - We need to figure out where in the page (Room Component) we want to populate the search results and how many results we want to display

- We haven't figured out how to update a user's profile

    - We need to use $writeDocument()$ method to write new json into our "users" collection.

- On the playlists page, we don't have an embedded SoundCloud player to play tracks

    - As of now, we have mocked up some songs in our database, but in the future these songs will come directly from the SoundCloud API. We will store the SoundCloud song track number in our database so we can pull information easily from the API

- We haven't implemented the like function for songs in the room

    - The playlist queue is based on the number of likes a certain song has
        * The song with the highest likes will be played next
    - A user can only like one song, so we need a way to check that using the database and the server

- Our CreateRoom component makes the same room everytime with the same ID.

    - One of our members has implemented a hash function that returns a unique code everytime you create a room, but it has not been implemented in our component as of yet. We will do this as soon as we have a real mongodb database
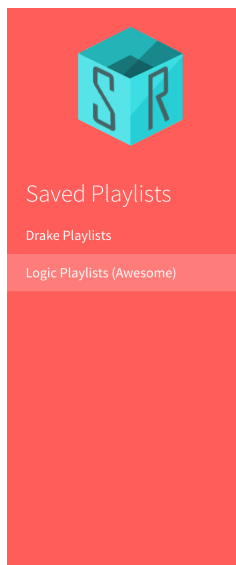
- After joining the room, it does not show the room id anywhere in the page. We will implement this by passing it a room id prop telling the RoomComponent what the id of the room is, thus we will be able to display this id in the component

- The room component does not have a SoundCloud embedded player at this moment. This feature will be implemented once we actually use the SoundCloud API to access data/songs.
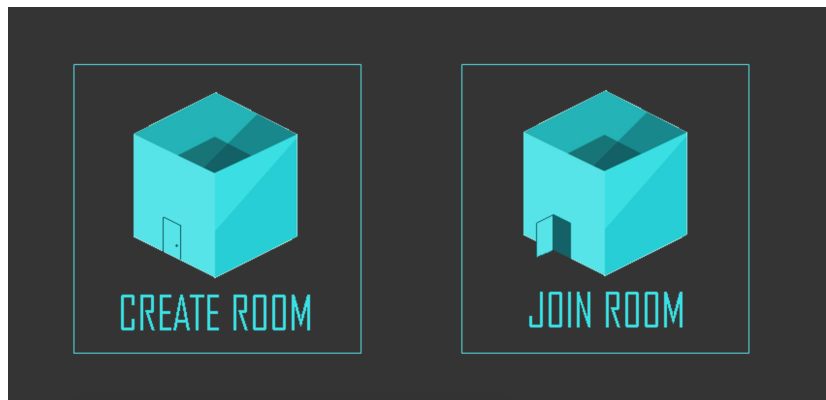
# 2   Description of Key SoundRoom React Components

- **Navbar Component**: Contains the HTML code for the Navbar for the webpage. It contains the "Welcome to SoundRoom" text and also contains links to the Account Info and Signout Pages. It also contains the Profile Picture of the User logged in.



- **Sidebar Component**: The Sidebar Component is a fixed component that contains a list of all the saved playlists for the user logged in. Essentially, these are links to another Playlist component where the user can actually play their songs in their saved playlist
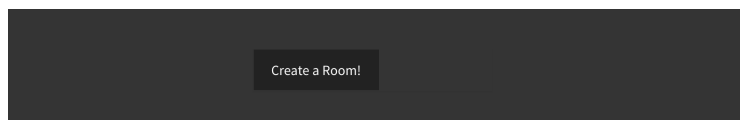
- **Homepage Component**: The Homepage component is the designed landing page when the user logs in into their account. The Homepage component contains links to creating a room and joining a room.
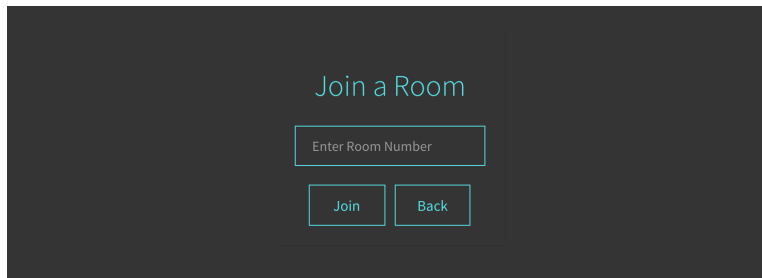


- **Playlist Component**: The Playlist component let users listen to their songs on their saved playlists. It component will directly connect to the SoundCloud API and embed the SoundCloud stream player to easily play songs



Drake Playlists

| Song Name | Artist | Album |
|---|---|---|
| Pt. 2 | Kayne West | Life of Pablo |
| Welcome to heartbreak | Kayne West | 808s Hearbreak |
| Starboy | The Weeknd | Starboy |
| Now and forever | Drake | If you are reading this |
| Soul Food | Logic | Underpressure |

- **CreateRoom Component**: The CreateRoom component lets users create a room or a session where other users can join in and enjoy a social playlist. The CreateRoom component gives a user a unique hash code for the room which they can give it to their friends so they can join the room
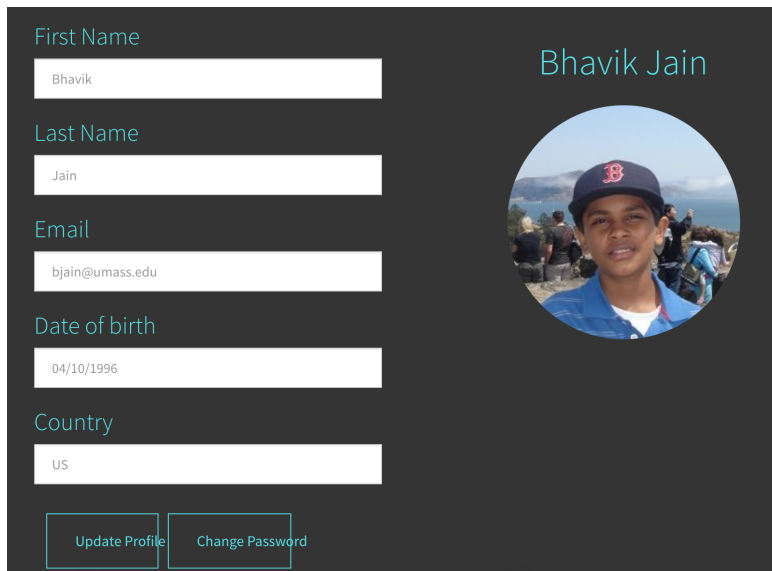


Create a Room!

- **JoinRoom Component**: The JoinRoom Component lets users join a room or session in which they have the access code for. They receive this access code through the host. Once the access code has been validated, the component takes the user into the Room where they can enjoy and listen to music



- **AccountInfo Component**: The AccountInfo component contains information about the user logged in such as their name, country, date of birth and etc. The component also lets the user update their account information.
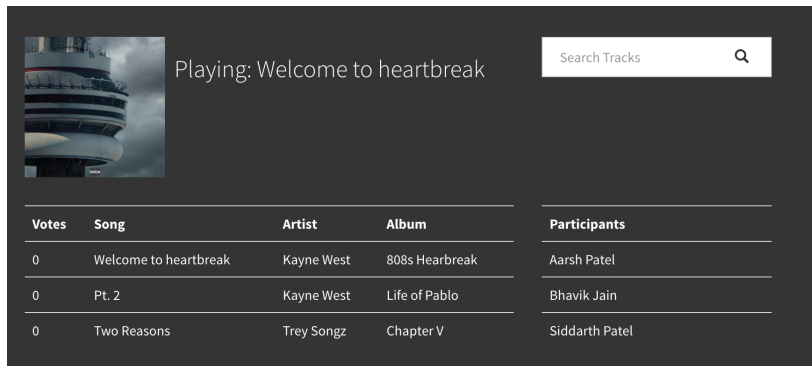
- **Room Component**: The Room Component is the main component of SoundRoom. It is essentially an online room that users can join using an access token (given by the host). In the room, the user can add songs to a playlist and upvote songs in order to queue up what song plays next. The Room Component keeps track of what songs are in the current playlist and also the participants. It also lets the participants save the playlist that's part of the room to their saved playlists, which would be displayed in the Sidebar. It also allows users to search songs using the SoundCloud API and add that song to the current playlist. The Room component needs to use socket.io (event based communication) to allow the room to communicate in real-time with all participants in the room.

# 3   Startup Founders Contribution

- **Aarsh Patel**

  - Made the Sidebar Component which displays the saved playlists for the user. He mocked up the users, playlists and the songs in the database and made sure everything made sense in the database. He also connected each of the user's playlists to a Playlist Component where the user can see the songs saved in that playlist.
  - Made the Navbar Component which is essentially a static component that contains links to the Account Info and the Signout (not implemented yet) pages.
  - Made the Homepage Component, which is also another static component which contains links to the Create Room and Join Room Components. Made sure the Homepage component is the first thing that get displayed when the user "logins" to SoundRoom.
  - Helped team members out with their issues with React/Javascript
  - With the help of Bhavik, assigned tasks to each member and made sure that these tasks were done on time

- **Bhavik Jain**

  - Made the Playlist Component. The Playlist Component conatains the user's saved songs for that playlist. Used mock data from the database to inject it into our playlist view. Essentially, the component takes the id of the playlist from the sidebar and using a function on the server, it will display the name of the playlist and the songs contained within that playlist along with the artist and the album.
  - Help design the linking between all pages using the React Router. He was the main person to go to our group for any React Router questions. Played a big role in designing the app.js file which instantiates our ReactRouter.
  - With the help of Aarsh, assigned task to each member and made sure that these tasks were done on time.
  - Helped all team members with their questions with the database/server

- **Ronit Arora**

  - Made the Create Room Component. Also created a unique alpha-numeric hash-code for the room and redirects the host to the room calling the Room Component with relevant props

- **Siddarth Patel**

– Made the JoinRoom Component which lets the user join an existing room via the roomId which is generated when a user creates a room. Made a validation function that checks to see if the roomID is currently in session (room is running) and takes them to the room if it.

- **Justin Martinelli**

  – Made the Account Info Component, which lets the user sees his/her account information. At the moment, input boxes have placeholders that displays the current information of the user that is saved in the database. The avatar on the top right takes the id of the user and dynamically displays the profile photo saved to that user on the database.

  – Designed the logos for the CreateRoom and JoinRoom buttons in the Homepage Component

  – Designed the colors for SoundRoom and helped make some important design decisions for the team

  – Helped all team members with their design questions or general CSS questions

- **Lynn Samson**

  – Made the dynamic Room Component. It contains the implementation of a particular room, which contains the playlist for the room, the list of participants in the room as well as the search component that lets the User add songs to the playlist. At the moment, playlist and participant information is being dynamically pulled from the mock database. I am still working on functionality to like a song, as well as implementing the search functionality. I spent a lot of time refactoring and modularizing the code, for example, by splitting the playlist and participants information into separate components.

# 4 Updated ER Diagram