

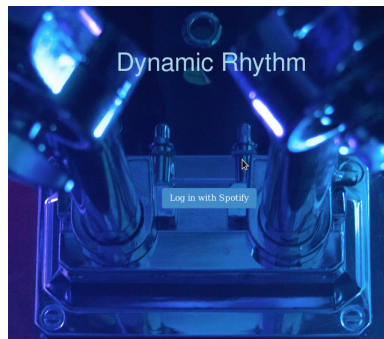
Dynamic Rhythm - Milestone 5: Testing

Sofie
Jake
Sricharan
Xingyu
Wylie

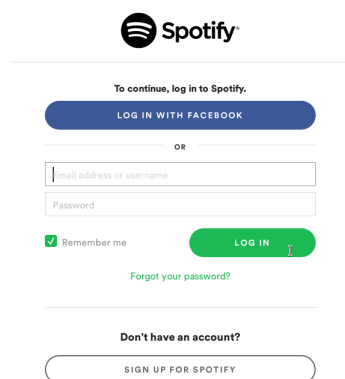
December 9th 2018

1 User Acceptance Tests

1.1 ‘Log in with Spotify’ Test



The first test is inclusive to logging in to Spotify via the application's JavaScript login call to Spotify. This is done manually by clicking the login button on the website.

A screenshot of the Spotify login page. At the top is the Spotify logo. Below it, the text "To continue, log in to Spotify." is displayed. There are two main login options: "LOG IN WITH FACEBOOK" and "LOG IN WITH SPOTIFY". The "LOG IN WITH SPOTIFY" option is highlighted. Below this, there are input fields for "Email address or username" and "Password". A "Remember me" checkbox is checked. A green "LOG IN" button is visible. Below the login fields, there is a link for "Forgot your password?". At the bottom, there is a link for "Don't have an account?" and a "SIGN UP FOR SPOTIFY" button.

Once the button has been clicked the user has to login with Spotify and fill out their account information as shown to the left. They then fill out their account information and click login. They can either login with Facebook and use Spotify and Facebook's authentication, or just login with Spotify and only using Spotify's authentication. Then the user should be sent to the main page of the website.

Once the user logs we can actually check for the information that we fetch from their account. For an example, here is Sri's account information below. We each have to verify this for ourselves. For individuals who login through Facebook, their display name is a number.

```
{
  "country": "US",
  "display_name": "kawmaster",
  "email": "sricharanvarra@gmail.com",
  "external_urls": {
    "spotify": "https://open.spotify.com/user/kawmaster"
  },
  "followers": {
    "href": null,
    "total": 2
  },
  "href": "https://api.spotify.com/v1/users/kawmaster",
  "id": "kawmaster",
  "images": [
    {
      "height": null,
      "url": "https://profile-images.scdn.co/images/userprofile/default/74198f751fbleebf67a78f6f5f55cfb41b066fb",
      "width": null
    }
  ],
  "product": "premium",
  "type": "user",
  "uri": "spotify:user:kawmaster"
}
```

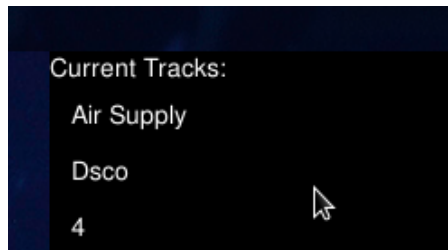
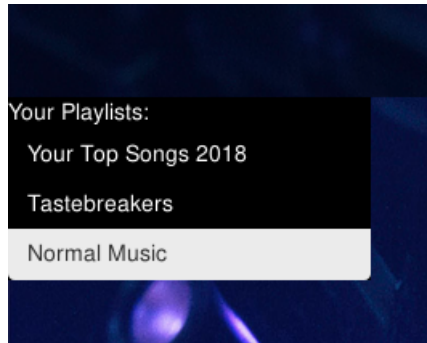
Furthermore, whenever anyone logs into our website they should be added into the database. If they are already in the database, they should not be added again. Below you can see that after logging in, I am able to see all of my own info, and the console notifies me that I am already in the database. We tested this with all group members and the function successfully added all of us.

```
Listening on 8888
{ country: 'US',
  display_name: 'Sofia Lange',
  email: 'sofie.lange.98@gmail.com',
  external_urls: { spotify: 'https://open.spotify.com/user/12136968179' },
  followers: { href: null, total: 7 },
  href: 'https://api.spotify.com/v1/users/12136968179',
  id: '12136968179',
  images:
    [ { height: null,
        url: 'https://scontent.xx.fbcdn.net/v/t1.0-1/c0.0.200.200a/p200x200/36316805_10215521724543833_7376957736514224128_n.jpg?_nc_cat=103&_nc_ht=scontent.xx&h=fbe9fc79910201927c81891973f54d11&oe=5C973503',
        width: null } ],
  product: 'premium',
  type: 'user',
  uri: 'spotify:user:12136968179' }
sofie.lange.98@gmail.com
12136968179
[ { count: '1' } ]
1
User already in database
```

id	email	name	country	uri
12136968179	sofie.lange.98@gmail.com	Sofia Lange	US	spotify:user:12136968179
test	test	test	test	test
bfgrih4n4ap4mrksubd4dig1	xizh8339@colorado.edu	Applet	US	spotify:user:bfgrih4n4ap4mrksubd4dig1
1285650559	maloney.jake.p@gmail.com	Jake Maloney	US	spotify:user:1285650559
1249055928	mrsnuffles.will@gmail.com	Will Wells	US	spotify:user:1249055928
kawmaster	sricharanvarra@gmail.com	kawmaster	US	spotify:user:kawmaster

(6 rows)

1.2 Playlist Control



The next step is for users to be able to control their playlists by double clicking on them. All playlists should be double clickable which allows them to be played via the Spotify playback SDK. For every playlist that is double clicked it should start playing the first track in the song list.

For example for Sri's 'Normal Music' playlist the first track should be Air Supply. The verification for this is pretty straightforward, as users more or less should know songs in their own playlists that they create.

1.3 Track Control



Within a playlist that users select they should be able to double click on track names on the right and get certain songs with their playlist to play. Once again the verification of this is having users play on songs that they know and listen if those are the correct ones that come up. For example when Run Into Flowers is double clicked, the box is shaded white, and the text turns black. Not every song is loaded by the API, as the maximum amount of tracks that a user can get is a bit above 200.

In addition to this the next and previous track buttons should do as expected, and in addition the user can press the left and right arrow keys to accomplish the same task. This can be trivially checked by the tester along with the pause and play buttons. For the pause and play button, the user can also opt to use the 'space' key on their keyboard that accomplishes both pause and play by alternating states.



1.4 Progress Bar and Track Seeking



Track Seeking is another feature that all modern music players out there have. The both act as a progress bar and allow the user to click on certain areas of the song, if they wish to go back or forward to a specific moment in time of the song. This test can be conducted by measuring the latency between initiating the seek protocol and for the bar to actually seek. Verification of the progress bar can be done attempting a seek somewhere there and verifying that the location in the track sounds correct. In addition to have a better form of testing, it is possible to measure