Workflow

High Level

**Spotify Authentication**

* Spotify api verifies application
* User allows app to access spotify

**Retrieve Playlists**

* Send request to see playlists user has access to/
* Or user searches for playlist
* User selects playlist to load
* Display all songs within selected playlist
* For every song, see if song id is found in db, if it is get totals of up/down votes
  + If not found in db, no votes have been made on song

**Voting**

* User upvotes or downvotes individual song
* Vote is sent as request to backend
* When a vote is placed, create a db entry
  + Db entry will contain song id, and increment value in either up or down vote column

**Extras**

* Specific number of down votes will remove a song from playlist
* To keep track of users we could use local storage in browser

Backend Design

**API Calls**

*Spotify Project*

getAuthUrl

-called if user is not already authenticated

-This will send user to a url to authenticate spotify

redirectUrl

-Called once users authorizes app to access their spotify

-Gets access token, refresh token

-Store tokens in db with user id

isAuthenticated

-call to check if user has been authenticated already

-search db filter on userid, see if token exists

getPlaylists

-can get a specific playlist

-can get users playlists

-get playlist cover image

-delete item from playlist

*API*

getPlaylists

voteSubmitted

**Db Models**