Design Document Template

**September 4th 20XX**

**OBJECTIVE**

To add more complex functionality that users might come to expect from an application like Jammming, by incorporating a small suite of features including:

* Search on Enter
* Calling a list of the user’s saved playlists on Spotify upon opening the application and saving playlist’s to Spotify
* Adding the ability to Sample (Via – Spotify), or to Play (Via – Last FM), tracks from search results prior to adding them to the playlist.
* Adding Dynamic Behavior to the Background image so that upon searching a term, a random image associated with the search will replace the default background image.

This suite of features will increase Jammming’s ability to compete with similar applications.

**BACKGROUND**

A more thorough explanation of why this feature is important to implement and all of the essential functionality of the feature. This section could contain research findings, user stories, examples of similar features, or anything else that explains why this feature has been prioritized and why the technical design laid out below implements the pieces of functionality that it does.

**TECHNICAL DESIGN**

**On enter:**

render: function () {

return <input type="text" onKeyPress={this.\_handleKeyPress} />;

},

\_handleKeyPress: function(e) {

if (e.key === 'Enter') {

console.log('do validate');

}

}

});

Call Playlist:

1. Get Current User:

Documentation: <https://developer.spotify.com/web-api/get-current-users-profile/>

Endpoint: GET https://api.spotify.com/v1/me

1. Get Current User’s Playlists.

Documentation: <https://developer.spotify.com/web-api/get-list-users-playlists/>

Endpoint: GET https://api.spotify.com/v1/users/{user\_id}/playlists

Play Sample:

1. Search (Already functional)
2. Get Track Id from Search for specific Track (already Functional - Partially)
3. Get Sample – When Grabbing Track info during search add attribute for track\_preview = track.preview\_url,
4. Create a button for play sample which uses track.preview\_url.

Play Track:

1. See Trackster project for implementation of Last FM for playing songs. (May or may not be possible based on information incompatibility between spotify and last FM.

Change Background:

1. Search (Already functional)
2. Get Track Info from Search for specific Track (already Functional - Partially)
3. Get Album Image – When Grabbing Track info during search add attribute for album-ImageUrl = track.album.images[0].url,
4. Add document.getElementById(‘root’).style.background-image=

This section should lay out all of the information needed to implement this feature. In the context of React, this could include new components and their functionality, existing component updates, how to address edge cases, and any other information an engineer would need before implementing this feature.

For front-end changes, this section will often include design mocks or wireframes to specify how the design of the application needs to be updated.

**CAVEATS**

This section is used to lay out alternative solutions and their respective drawbacks, as well as potential drawbacks to the proposed solution above. This is used to make it clear why the technical implementation detailed previously was chosen instead of alternatives. It additionally allows stakeholders or other developers to consider those drawbacks and choose one of the alternate solutions if they prefer it. This may occur if they feel the benefits or drawbacks of that solution are more desirable than the current solution, or if they can identify other benefits and drawbacks not currently listed.