Tremeloo: Spotify Music Suggestions

CS 4850

Spring Semester

INDY-7

Team Members:

Vincent Green, Leiko Niwano, Mark Walker, Khemrind Ung

Advisor: Sharon Perry

Tremeloo

(Draft)

Project Overview

Our project, Tremeloo, is a mobile application implemented in JavaScript and TypeScript utilizing React Native components and Spotify API endpoints. The app receives data from a user's Spotify account and displays their playlists. The app is then able to parse Spotify's database of music and suggest music tracks based off the content in each individual playlist.

Project Background

Executive Summary

Table of Contents

Project Planning and Management

General group interaction and file sharing were performed through Microsoft Teams, as well as deadline notification and schedule management. Team meetings were held on the a weekly basis and recorded and transcribed via Teams functionality.

Project Requirements

1. Gain access to user's Spotify Account.
2. Load and display a list of the user's playlists.
    a. Send GET request for list of the user's playlists.
    b. Display list of playlists.
        i. Display playlist cover image.

        1. Send get playlist cover image request.
    ii. Display playlist name.
3. Access each of the user's playlist
    a. Send GET playlist request.
5. Output a list of song recommendations for each unique playlist.
    a. Pass MD into recommendation algorithm.
    b. Display returned list of song names.
6. Add songs from the recommendation list to the user's playlist.
    a. Send POST request for chosen song by user using MD.
7. Recommendation Algorithm:
    a. Access MD from each song within a given playlist.
    b. Access Spotify song DB
    c. Compare MD of user's playlist songs to MD of Spotify database
    d. Generate list of songs matching MD criteria
9. Certain actions are restricted for Spotify free users. In these cases, Spotify *recommends* displaying the message depending on usage of either Android or iOS as follows:
    a. Android premium message: "Spotify Premium lets you play any track, ad-free and with better audio quality. Go to spotify.com/premium to try it for free."
    b. iOS premium message. "Spotify Premium lets you play any track, ad-free and with better audio quality." Link to quote.

Architecture

Design

Tools

Adobe XD and Adobe Illustrator were the main developer tools utilized in the production of the application. Adobe XD was used to produce mockups of each app screen. As well as demonstrating the overall flow of the app. Adobe Illustrator was utilized in the development of the various logos and images of the app. The app logo was created as scalable vector graphic to ensure image sharpness at any zoom magnification.

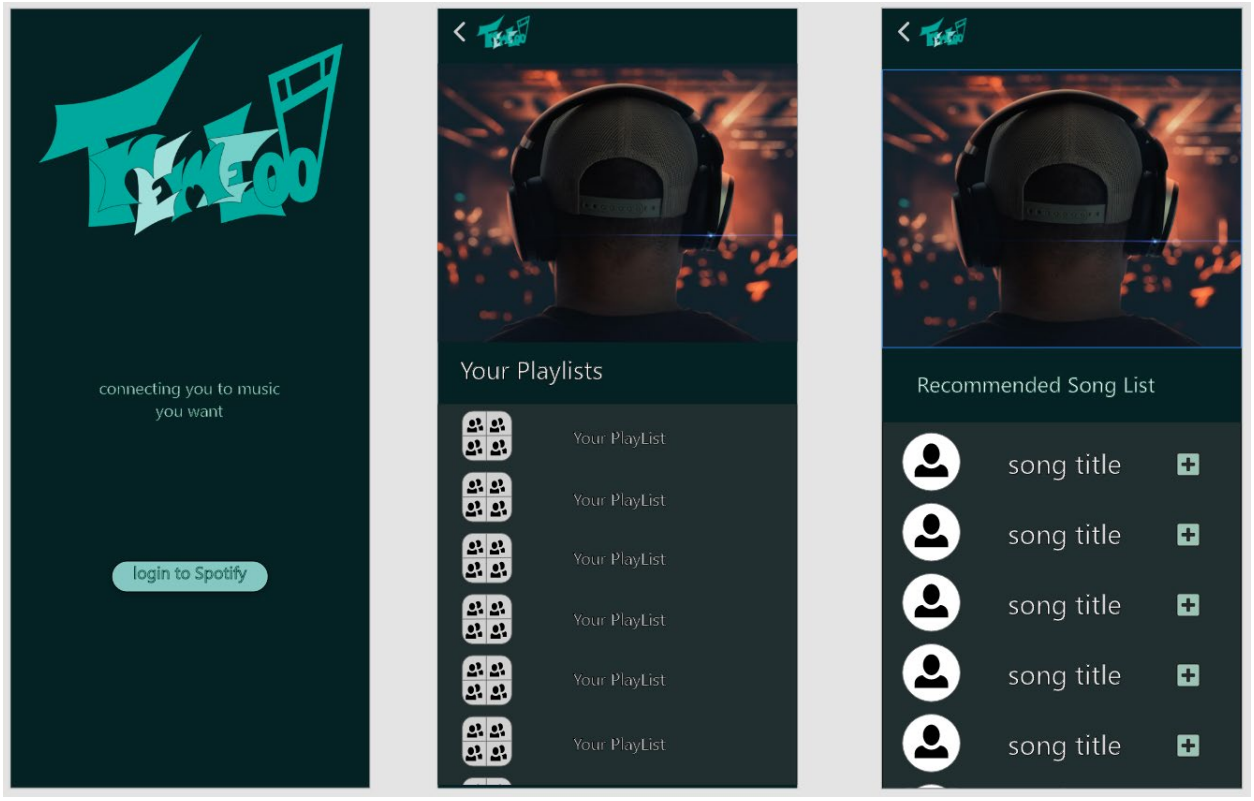

Figure 1. Early UI Mockup

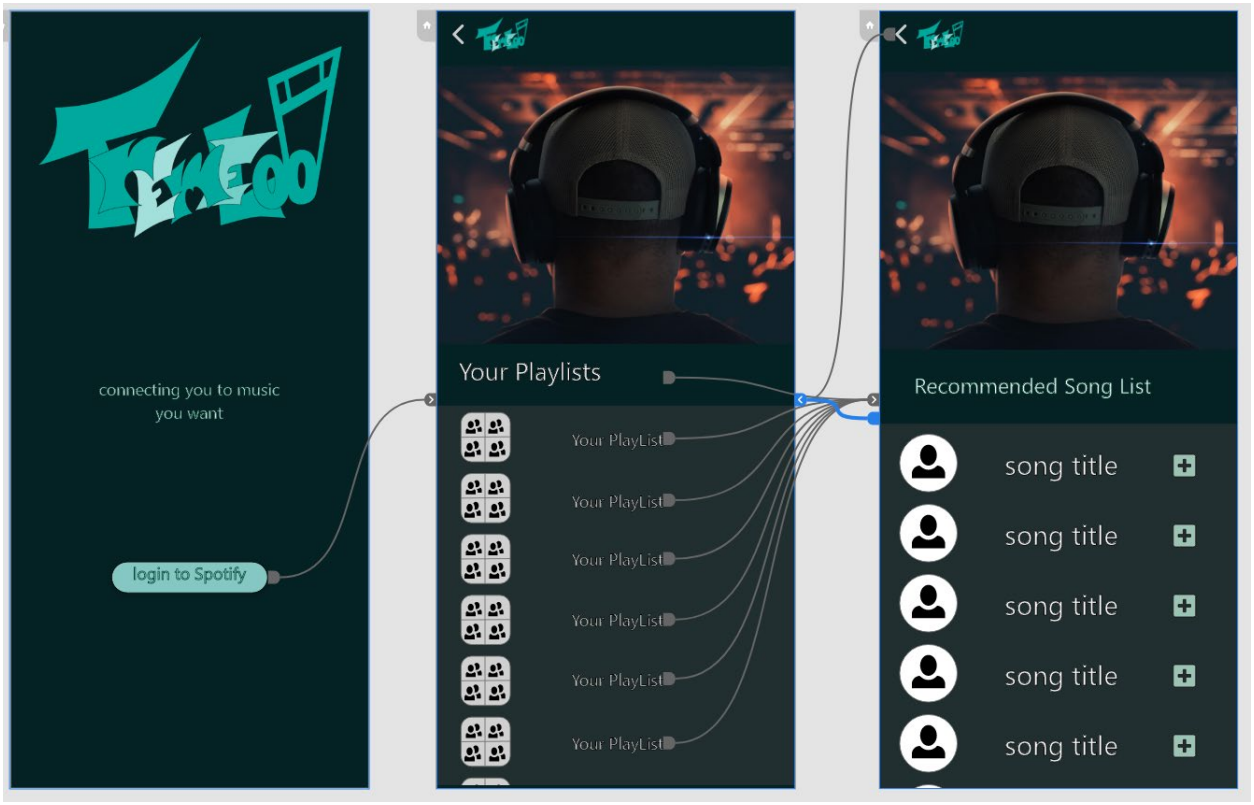

Figure 2: UI Design Flow

Analysis

The design mockup details the overall structure of the applications 3 main pages: home, playlists, and song recommendations. The home page consists of the app logo, a brief text slogan and a button for logging into the Spotify authentication process. The playlist page contains the navigation menu on top, a banner image, A playlist header for the list of playlists, and a scroll view element containing the user's playlists. The song recommendation page is designed in the same manner as the playlist page, with alteration to the scroll view header string and a displayed list of recommended songs.

Figure 2 is a UI mockup of the design flow for the app. The arrows are indicators for screen transitions. Clicking the login to Spotify button transitions the user from the home page screen to the playlist screen. On the playlist screen, clicking a playlist name transitions the user to the song recommendation screen. The user is so capable of navigating back to the previous screen by clicking either the back button in the upper left corner or swiping right on the screen.

Implementation

Tools

The implementation of the project is conducted using React-Native integrating with the Spotify API. React-Native -> uses the Metro bundler to compile various JavaScript files into a single consolidated file to be run

React-Navigation

Spotify API

Node.js

Analysis

The UI is implemented in JavaScript with React, React-Native, and React-Navigation libraries. The main app.js file imports the react-navigation library which is utilized to build a stack navigator for each of the various screens. Each screen is a separate JS file located in the pages folder within the component directory. Each page contains its own CSS styles sheet. All logos and images are located in the assets directory. The backend is implemented in TypeScript with Expo, React, and Spotify API libraries. The backend file returns an object consisting of 3 main functions (Authenticate, RefreshRecommendations, and AddTracks), a Boolean variable (isAuthenticated) and an array (Playlists). The Authenticate function grants access to the user's Spotify account and grants the app permission to access and edit the user's playlists. The RefreshRecommendations functions performs an API call to Spotify to retrieve a newly updated list of songs displayed for recommendation to the user. The AddTracks function makes an API call to add the selected recommended song to user's currently selected playlist. The isAuthenticated variable is a simple Boolean indicating whether a valid access token has been retrieved from the Spotify API. The Playlist array is a list of each individual playlist located within the user's Spotify library.

Execution

Tools

The project was executed using Expo CLI and Expo Go – open-source command line interfacing and framework that enable the development and testing of React Native applications in both iOS and Android environments.

Analysis

Software Test Specification

When the user opens the app, they are directed to the homepage component. Upon clicking the login to Spotify button, a call to the authenticate function is made and the user is directed to a web browser to authenticate the app. Upon authentication, the user is directed to the YourPlaylist component. A scroll view box containing a list of each of the user's playlists is displayed from the playlist array. Clicking a playlist with the in the scroll view navigates the user to the SongRecs component. The Song Recs contains a scroll view box displaying a unique list of recommended songs generated for each playlist. Clicking the add to playlist button on the right side of the screen calls the AddTrack function, adding the song the current playlist. A new song is then added to the recommended song list.

Conclusion

Appendix

A – Gantt Chart

Project Name:	Spotify Music Suggestions																			
Report Date:	4/6/2023																			
			Milestone #1									*Milestone #2*								C-Day
Deliverable	**Tasks**	**Assigned To**	01/12	01/19	01/26	02/02	02/09	02/16	02/23	03/02	03/09	03/16	03/23	03/30	04/06	04/13	04/20	04/27		
Requirements	Meet with group	All	8	8																
	Assign roles	All				2														
	Define requirements	Mark, Vincent			2	2	2													
	Review requirements with group	All				3	1													
Research	React-Native documentation	All			4	4														
	Spotify API documentation	All			4	4														
Project Design	Define tech required*	All				2	2													
	Discussion of the flow of the app (ERD concept design)	Khemrind, Vincent				2	4													
	Create UI concept of the flow of the app	Leiko, Vincent				2	4													
	Access Spotify Database	Khemrind, Leiko					4	2												
	Integrate with user / user data	Khemrind, Leiko						6	8											
	Develop music recommendation algorithm	All						5	10	10										
	Update playlist following recommendations	Khemrind							10	10										
	Develop working prototype	Khemrind, Leiko							15	15	5									
	Test prototype	All								4	4	2								
Development	Review prototype design	All									2	2	2							
	Rework requirements	All											6	6	6	6				
	Document changes	Mark														3	4			
	QA Testing	All													5	5	5			
Final report	Presentation preparation	Mark, Vincent															4	4	3	
	Website preparation	Mark, Vincent												4			2			
	final report draft	Mark, Vincent													8					
	Final report submission to D2L	Mark, Vincent															1	1		
		Total work hours	249	8	8	10	21	17	13	43	39	11	4	8	10	19	18	16	4	

Legend	
Planned	
Delayed	
Number	Work: man hours

B – Phase 2 requirements

- Display a list of songs within the user's playlist (PHASE 2).
    - IMPORTANT
        - All MD displayed from Spotify MUST link back to the Spotify app.
        - If the user does not have the Spotify app installed, links must divert them to the app store to download the app.
    - Display song name from GET playlist request.
    - Display artist image from GET playlist request.

8. All songs must link to Spotify app. (PHASE 2)
    a. Display song name and album artwork of selected song at the top of the page.
    b. Newly displayed song links to the Spotify app.
10. Play preview of selected song. (PHASE 2).
    a. Get preview URL from returned JSON object.