MUSIC PLAYLIST SRS

1. Introduction

1.1. Purpose

The purpose of this app is to make playlists collaborative and flexible. It allows friends to connect with one another to share content in an easily digestible, fair, and fun way. It also eliminates the frustration of having people's content over a multitude of devices, which typically has made the exchange of music awkward. The aim is also to share the burden of streaming music among everyone on the receiving end, preserving people's data and avoiding costly data overages.

1.2. Background

This application will allow multiple users to make collaborative playlists as well as implement a point system in a game-like fashion. Points will be acquired through various activities such as answering trivia questions or getting "likes" on your song choice.

2. System Requirements

2.1. Actors

Host: Spotify users who has a premium Spotify account and act as the host phone.
User: Other individuals that are invited to connect to the Host.

2.2. High Level Flow

To use the application, the host will initiate a playlist upon which the users will join and begin adding songs. The devices will ideally be within a certain proximity to the host to be a valid member of the playlist.

2.3. Uses Cases

2.3.1. Account Creation
- 2.3.1.1. Log in
  - 2.3.1.1.1. Enter email/username
  - 2.3.1.1.2. Enter password
- 2.3.1.2. Create an account
  - 2.3.1.2.1. Enter email
  - 2.3.1.2.2. Enter password
  - 2.3.1.2.3. Enter other account attributes
- 2.3.1.3. No Account option
  - 2.3.1.3.1. Would have limited features, functionality

2.3.2. Main Menu
- 2.3.2.1. Can choose to be either a Host or User
  - 2.3.2.1.1. Host: Create a playlist and connect to Spotify
  - 2.3.2.1.2. User: Connect based on invitation from User
- 2.3.2.2. List of different options to choose from
  - 2.3.2.2.1. Add songs #8
  - 2.3.2.2.2. Viewing current playlist and current song playing

2.3.3. Playlist Creation
- 2.3.3.1. Host names playlist
- 2.3.3.2. Host sets playlist attributes

2.3.4. Joining the playlist
- 2.3.4.1. Host invites friends to join the playlist
- 2.3.4.2. Invites directly through the app
- 2.3.4.3. Users join the group by accepting invite

2.3.5. Content Sourcing
- 2.3.5.1. Local Storage
  - 2.3.5.1.1. Ask permission to access stored library
- 2.3.5.2. Spotify authentication
  - 2.3.5.2.1. Use spotify android sdk (or IOS) to access their spotify account
- 2.3.5.3. Other
  - 2.3.5.3.1. Youtube, Soundcloud, etc.

2.3.6. Content Sharing
- 2.3.6.1. Local wifi network
  - 2.3.6.1.1. Initiated through app by host
- 2.3.6.2. Bluetooth
  - 2.3.6.2.1. Limited on connectivity to roughly 6-7 devices
- 2.3.6.3. Web server specific to the app
- 2.3.6.4. Other Services
  - 2.3.6.4.1. i.e. Play through app on Youtube, Soundcloud, etc…

2.3.7. Playlist Prioritization
- 2.3.7.1. FIFO queue
- 2.3.7.2. Number of songs a user has played determines their suggestions priority
  - 2.3.7.2.1. More songs played := lower priority
- 2.3.7.3. Result of games played, winner gets priority points
- 2.3.7.4. Thumbs up or down (like/dislike)

2.3.8. Searching for Songs
- 2.3.8.1. Access to database of songs either through spotify or host's available song
- 2.3.8.2. Ability to look through based on genre, artist, album, or song

2.3.9. Adding Content to Playlist/ Privileges of Users
- 2.3.9.1. All Users (Host + Others)
  - 2.3.9.1.1. Permission to source songs under any method from section #4
- 2.3.9.2. Host User
  - 2.3.9.2.1. Top Priority- respect to the driver
  - 2.3.9.2.2. Authority to remove any song from playlist
  - 2.3.9.2.3. Post to spotify or share with friends options
- 2.3.9.3. Other users
- 2.3.9.4. Search for available songs from section #7(Searching for Songs)
- 2.3.9.5. Choose available songs and add into the playlist
  - 2.3.9.5.1. Ranking determined by methods from section #6

2.3.10. History/Log of Songs played

2.3.10.1. Saves a list of songs recently heard for later referencing

2.3.10.1.1. Cross references songs in playlist with your library to recommend songs you don't have

2.3.10.1.2. On songs explicitly liked, the user can see a diff between their library and the new songs

2.3.10.1.3. Filter where the user can choose to see only songs they liked or just the basic difference between the playlist and their library

2.3.10.2. Song and artist recommendations

2.3.11. Load Balancing

2.3.11.1. Distribute the amount of data used between devices evenly

Features we can implement

- Point system to determine road trip winner
  - Likes on recommended songs that give the suggester of said song points. points can be used to trump other people's suggestions aka move your song to top priority
  - Lose points when you request a song that someone already requested.
- Trivia about the artist (true false) that gives priority points
  - Guess the song/artist playing
- Equal priority showdown (mini game to determine whose song is played next)
- Google maps (to determine estimated time of arrival)
- Implement some type of karaoke(they will be able to sing their favorite song).
  - Display Lyrics
- Be able to share what they are listening to the social media(e.g. facebook..)
- Basic: Being able to add songs to spotify playlist through the app
- Advanced: Having the app organize the songs that are added and choosing them based on point based system