MUSIC PLAYLIST SRS

1. Introduction

1.1. Purpose

The purpose of this app is to make song 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. Song links are sent to the host who then downloads the song and plays the song for everyone.

2. System Requirements

2.1. Actors

- Host: Downloads youtube links as mp3s and plays them for everyone.
- User: Other individuals that are invited to connect to the Host and submit song links.

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. Users will have to connect to the host before being able to submit songs to the playlist

2.3. Uses Cases

2.3.1. Host Connection

2.3.1.1. User must enter the Server Name they want to connect to and the port number in order to join a group.

2.3.2. Main Menu

2.3.2.1. User song request window

2.3.2.1.1. User has to input the youtube link address to the song they would like added to the queue

2.3.2.2. Media player options

2.3.2.2.1. User can input the youtube url to play

2.3.2.2.2. Host window is able to view the song name playing

2.3.2.2.3. Host displays the song duration and the progress of the song through a progress slider.

2.3.2.2.4. Host: Can adjust volume, pause, resume, and skip the song

2.3.3. Playlist Queue Prioritization

2.3.3.1. FIFO queue

2.3.3.1.1. Songs that are submitted by the client window are put into the queue in a FIFO order

2.3.3.2. Number of songs a user has played determines their suggestions priority

2.3.3.2.1. More songs played := lower priority

2.3.3.3. If similar priority, a game determines the one who goes next

2.3.4. Adding Content to Playlist/ Privileges of Users

2.3.4.1. Client sends the song link to the mediator

2.3.4.1.1. Mediator checks if the song can be downloaded or if there is already a song being downloaded

2.3.4.1.1.1. If there is already a song playing, it will wait before starting the download.

2.3.4.1.2. Downloader will download the song and sends the file to the media player before going back to the mediator.

2.3.4.1.3. Media player adds the song to the playlist in FIFO order.

2.3.5. Song play

2.3.5.1. Once a song is downloaded, the host has to click on the play button to being playing the songs.

2.3.5.1.1. Song will play one after the other

2.3.5.2. In order to stop the songs playing, the host must click the pause button

2.3.6. Exiting the program

2.3.6.1. Both the client and the host can close out the program by terminating them through the windows state button or title bar buttons on a mac.