NeuMiu

Executive Summary:

A music player that sorts music by the user's choice in to multiple different types of playlists, keeps track of how many times a song has been played, allows the user to upload their own music, allows a user to assign an image to a song and rename their files, and can sort by name, genre, artist, colour?, and song length. (possible lyrics page)

Specific Terms:

* MIDI - Musical Instrument Digital Interface
* MP3 - MPEG (Motion Picture Experts Group) Layer-3
* OGG - Jargon for Ogging
* M4A - Different form of compression for MP3’s
* FLAC - Free Lossless Audio Codec
* WAV - Waveform Audio File
* EQ - Equalization

Requirements Map:

The Menu Bar in the GUI will contain drop down methods that will show other Buttons that run methods that allow a user to Add/Edit/Remove Songs, and Create/Edit/Remove Playlists.

The “ADD” Button under the Songs tab will create a new GUI Window that will contain TextField(s) where the user will input information for a file, in the term of Strings. (EX: Title, Artist, Genre, IMG File Path for the Cover Art, File Path for the Song). When the user clicks the saveSong Button at the bottom of the new GUI Window, the song will be added to a List of Songs in the default Playlist: “(none)”.

The “EDIT” Button under the Songs tab will create a new GUI Window that will contain the same TextField(s) as the add Button, and load the currently saved information from the previous process. Each of the TextField(s) will be editable so that the user can edit the information needed, and saveChanges Button at the bottom of the GUI Window will save the changes, and take the user back to the main GUI.

The “REMOVE” Button under the Songs tab will allow the user to select a song from the default playlist and remove its data from said playlist, thus removing the song from the entire application.

The “CREATE” Button under the Playlists tab will open a new GUI Window that will have a TextField prompting for the Name of the new playlist. Once the name is defined, the user can click on the “DONE” Button, and the Playlist will show up under the playlist File Tree on the main GUI Window.

The “EDIT” Button under the Playlists tab will open a new GUI Window that will allow the user to select the Playlist they would like to modify, and also allow the user to take all the selected songs, and store them into the Playlist they defined via CheckBoxes in a new GUI Window.

The “REMOVE” Button under the Playlists tab will open a new GUI Window similar to the one opened by the “CREATE” Button, but will offer a DropDown or other Listing type mechanic, and allow the user to select a playlist that they would like to delete, or remove from the application.

There are two volume controls in the Settings tab. One is powered by a Vertical Slider, and the other powered by a CheckBox. The slider will control the volume of the application on a 1:1 scale, and the CheckBox will mute all audio from the program. The user can modify the slider by clicking on the marker, and dragging it up or down.

On the Face of the Player, there are two boxes. One is a File Tree meant to display all of the available Playlists that were added and modified. There is to be one default playlist will all of the added music, so that the display on the right side of the application can serve its job to List all of the songs inside of a playlist.

Specifically to the bottom left hand side of the GUI there is a ImageView for the Album Cover, that of which the Path was defined when creating a song. That can be updated/changed when EDITing a song, using the EDIT Button under the Songs tab.

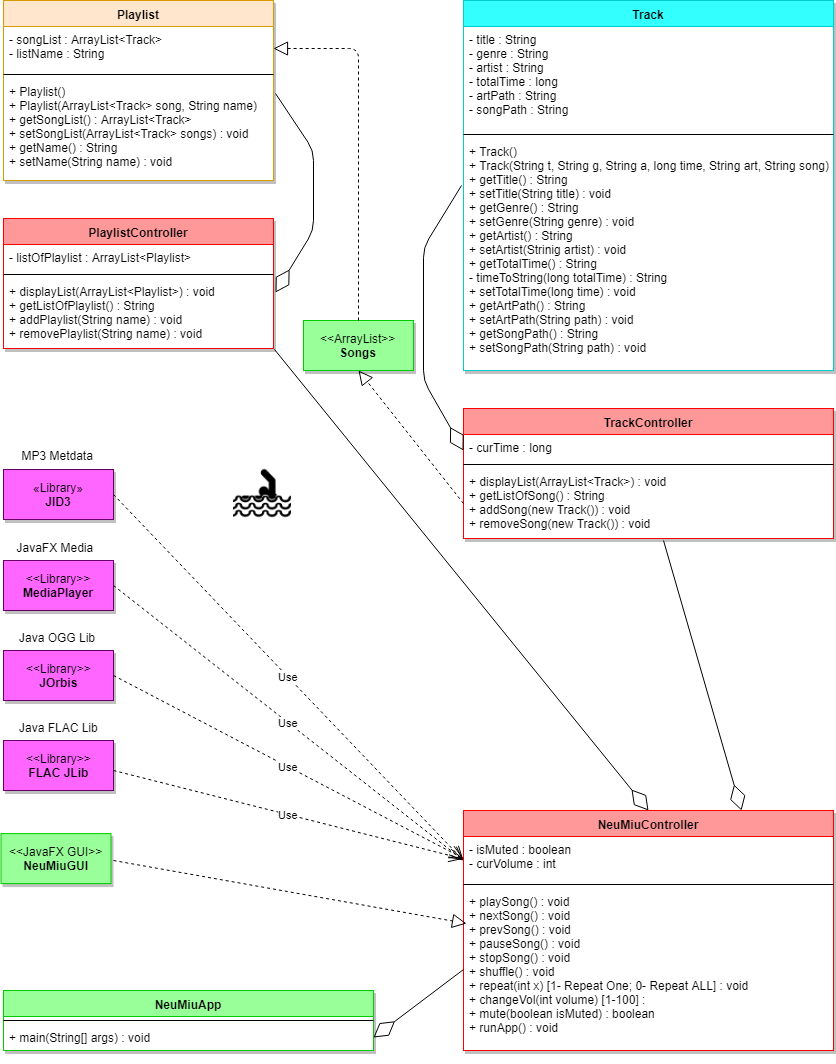
There is a Seekbar which is powered by a Horizontal Slider underneath the Songs File List box. The seekbar will follow the current timestamps of the Song that is being played. It has two labels that will change as a song is selected. Current Time to tell what time the song is currently at while playing, and the other telling the Total Time that is available in said song.

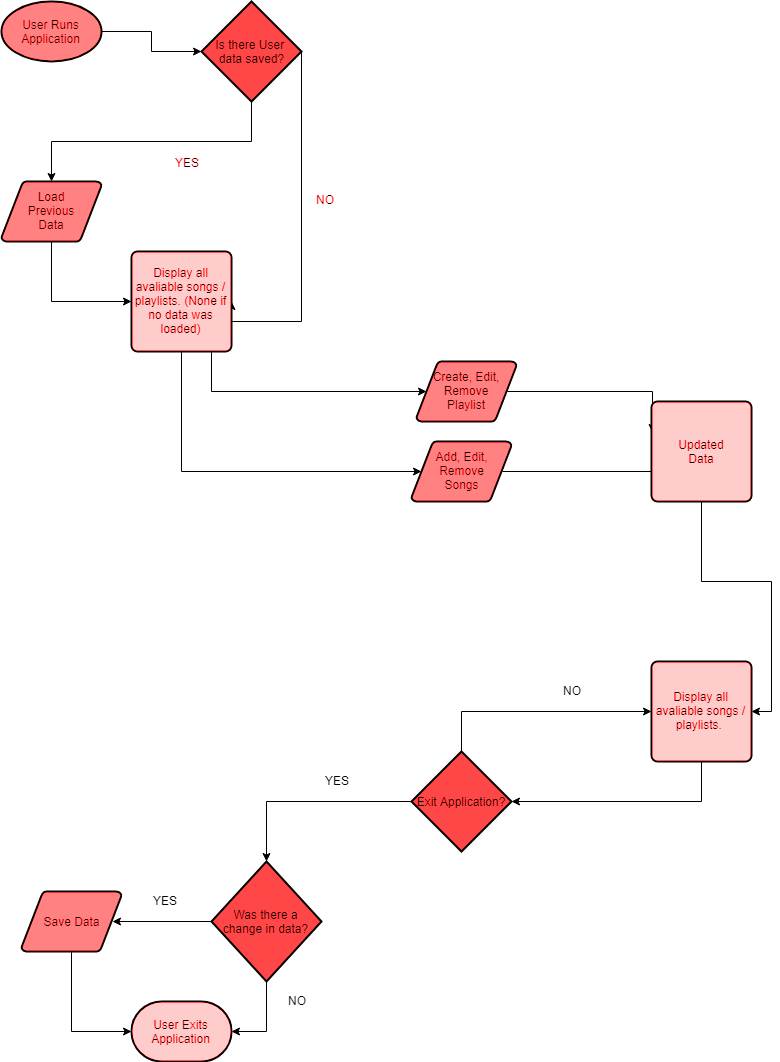
Directly under the Seekbar are the Control Buttons. Each button is to work on its own task. Specifically speaking, The Button labeled PLAY will resume playing if the song is currently paused, or start playing music if nothing was previously playing. The STOP button will stop the song/track completely, and the PAUSE button will simply pause the track, allowing it to be resumed in the future by the PLAY button.

There are Two audio seek buttons. One, known as the SKIP button will skip ahead to the next song, and the BACK button will return to the one that was playing prior to skipping. Along with the seek buttons, there are REWIND and FASTFORWARD buttons as well, which will allow a user to seek through the song without touching the Horizontal Slider.

Finally, there will be Repeat or Shuffle options, with the SHUFFLE option being controlled by a CheckBox. While checked it will shuffle the music in said Playlist before the next track is played, and it will put a song on at random after the shuffling process finishes. Like the shuffle option, the REPEAT functions will support the options of ONE or ALL, in which it will continuously play one track, or the entire playlist once all other songs in the list get played.

UML:



Process(Data) Flow:

* The user runs the application
* The application attempts to load previous data
* If data was found, it displays all song/playlist data, else it shows blank for no data being loaded (Normal on first time runs).
* All Modifications to the default Song List or Playlist(s) will be updated and showed like the start time loaded data
* Upon application exit, it saves all changed data if there was a difference from the originally loaded file
* The application finishes saving, and then closes.

Mock Up’s:



Test Case:

* The user wants to download music into their music player
  + They open up NeuMiu
  + They hover over the ‘Songs’ dropdown menu
  + They select add music
  + They select the files they want to upload from their files
  + They select ok
  + The music is added to their library
* The user downloaded music with no information, yet they want to sort their music by genre
  + The user goes to the songs dropdown
  + They select edit song info
  + They choose what song they want to edit
  + They enter the information they want
  + They hit save
  + They can then go to settings and sort and sort songs by the genere’s they entered
* The user wants the music to play louder
  + They find the volume slider
  + They adjust up to make it louder
  + They adjust down to make it softer
* The user wants to jump to a certain point in the song
  + The user hovers over the seek bar
  + They click the point in the song they want
  + The song jumps to that point
* The user wants to create different types of playlists for different moods
  + User goes to add a playlist
  + The user names their playlist
  + The user can then add any other information they want about their playlist, such as colours or genres
  + User then go to the add song button and select what songs they want in their playlist
  + User hits save
  + User has a new playlist
* A user downloaded a song and the one song is really three in one
  + The user finds where the first song stops and the second song starts with the seek bar
  + The user would then mark that place with the flag button
  + When the user wants to skip to the ‘next song’, they’ll click on the flag button to skip to their next time flag.
* A user wants to crank the bass up on one of his songs so much so the original song is distorted
  + They go to the songs drop down menu
  + They choose the EQ tab and options of different EQ settings are displayed to the user
  + The user chooses to maximize the bass
  + The song becomes heavily distorted with bass
* A user wants to add a song from youtube into his music library
  + The user copies the video url
  + The user pasts the video url into the youtube-mp3 tab in NeuMiu
  + They name the song
  + Hit add song
  + The song gets added to the users music library
* A user wants to shuffle his playlist
  + He selects the *shuffle playlists* button
  + The songs are shuffled using unique algorithms
* A user wants to sort his songs by their titles
  + She goes to the songs tab
  + She clicks the sort button
  + She chooses the playlist she wants to sort
  + The songs are sorted by their title