Final Project Progress Report

Report 1

Xingbang Liu, Matt Jones, Travis Thomas

April 2, 2018

As we discussed and brainstormed our "Music Suggestions & Making Software" we decided the simplest playlist storing data structure, an ArrayList Stack, provides us with methods useful to the ones we would call in our program (Play, Next, etc.). Because of this, we will be using many different stacks to store and sort our data. We will also be using an ArrayList for its useful indexing of elements, and because it works with the "contains" method to make sure our playlist doesn't contain repeats. We also will be using the "file reading software" our team has developed in previous labs, to store all the initial data, and continue to discuss whether we should store our final output in a csv file as well.

For sorting technique, we did not plan much for now, but We will use our previous bubblesort algorithm to sort the final output/playlist alphabetically.

Basic Code Structure

By conbining the code we had in the previous lab and the code we found online. In this siurce, we found how to develop effecient algorithm to deal with large amount of data. We can use 2-dementional ArrayList to store our music cloud list. Factors by rows, and items by columns.

For the example of our previous code of reading CSV file:

```
//Rank and Year
SortInt RY = new SortInt();
//Song and Genre
SortString SAG = new SortString();

BufferedReader br = null;
String line = "";
String cvsSplitBy = ",";

System.out.println("Here is your playlist: "); //add to class

  try {

  br = new BufferedReader(new FileReader(csvFile));
  System.out.println("Rank   Song   Artist   Year   Genre");
  while ((line = br.readLine()) != null) {

      // use comma as separator
      String[] Songs = line.split(cvsSplitBy);

      System.out.println(Songs[0] + "   " + Songs[1] + "   " + Songs[2] +
      +"   " + Songs[3] + "   " + Songs[4]);
    }

  } catch (FileNotFoundException e) {
      e.printStackTrace();
```

```
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (br != null) {
            try {
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
```

Stack DailyMix (stack contains songs feeded to users which picked according to user liked list);

Stack userLikeList(stack contains user liked songs);

In this list, we would cotain anoher column which marks like or not. "like" represents like, "dislike" represents don't like. So if user marks a song "like", program will first search through the userLikeList, if there are no repeated songs, then liked song will be added to userLikeList.

while(hasNext())

do commends;

This means user could kept prompting commands until user prompted quit.

Main Code Methods

To show playlist, the program will print out the whole list. Then, user can get specific factors, such as artist, song, genre, by typing it.

```
//Play:
if(userInput = play){
  System.out.println("Playing DailyMix.")
  Use the "next" method to play from dailymix stack;
}


//View:
if(userInput = view){
  System.out.println("Pulling up your Favorites...");
  System.out.prinln(Arrays.toString(Favorites.toArray));
  }

//Exit:
if(userInput = exit){
  exit;
}
```

Conlusion

This is the work we have for now. Next, we would like to finishing most of the code we planned here, and think of more functions we could add.