LyricsByU

Software Requirements Specification

SFWR ENG 2XB3

March 31, 2017

Group#: 8

Team Name: Lyricist

McMaster University

Domain

Our idea is to allow our users to make new lyrics in an interesting way. The program is intended to help people who are inspired by something but not creative enough to write their own song, or performers looking to quickly create a new and interesting piece. We will use a dataset which contains more than 380000 lyrics from thousands of different artists. The program will use lyrics from the dataset to make a new combination of lyrics based on the keyword/s which the user wants it to contain. The algorithms we plan to use for this program are mergesort, binary search, and a string matching algorithm. We plan to use Java to implement this application which will be used on a personal computer.

As stakeholders in this project, our goal of this project is to learn the professial and practical process of creating an application. It will also help us to make some critical thinking about the dataset.

Our expectation is to fulfill every requirment from each step to create a correct and robust program.

Other stakeholders include our professor Reza Samavi and our TA Terin Dhadda who expect us to work together to meet our milestones and goals throughout the development process and to produce an algorithmically challenging program at the end of our development term which solves a need of its users.

The clients are another major stakeholder for this project. Our users will be people who will want to create new and interesting musical lyrics based off of the sucesses of thousands of artists, and they will want the program to produce reliable results in a short period of time.

Functional Requirements

Req#	Requirement	Priority	Date Reviewed
1	The program should ask the user for keywords to help find lyrics from the dataset.	1	21/03/2017
2	For each keyword entered by the user, the program should find a line from the lyrics dataset containing a matching word.	1	27/03/2017
3	If the program finds multiple lines that contain the keyword, the program should choose one line at random.	3	21/03/2017
4	The program should allow the user to specify a genre, year, or artist to find lyrics from.	3	21/03/2017
5	The program should continue to find lines from the lyrics dataset until the user tells it to stop.	2	21/03/2017
6	The program should create a song for the user by putting the lines that it has found together.	1	21/03/2017
7	The program should display the song it has created for the user.	1	21/03/2017
8	After the program has created a song, it should ask the user if it wants to create another song.	2	21/03/2017
9	All input entered by the user(keywords, specification of genre/year/artist, whether or not the user wants to stop the program, and whether or not the user wants to create another song), should be read from the console.	1	27/03/2017

Non-Functional Requirements

Reliability(availability, integrity, security, safety)

User errors should be accounted for to avoid a crash. Expected input errors will be handled and the user will be asked to input a correct string.

Security

N/A

Safety

If the crash happens, it should never damage the system or hardware.

Accuracy of the Result

The lyrics output should always contain the desired keyword(s) that the user entered.

Performance

When searching the lyrics according to a keyword, the time should be less than three seconds. Generally, a desirable lyrics should be created in less than two minutes.

Human-Computer Interface Issues

The program will prompt the user and specify what input is required when necessary. Overall, at least 95% of people should immediately how to use the program.

Physical Constaints

This program is only for personal computer users.

Portability

The program is designed to run on Windows, MacOS, and Linux environments.

Requirements On the Development and Maintenance Process

Corrective Maintenance

- According to the specification, the program is going to make a symbol table when initialized. The keys are more than 10000 common English words and the values are Song objects containing the data for the song such as the artist, genre, and lyrics.

    1. Checking that the key contains almost every word and no repetitive ones.
    2. Randomly selecting some words in the key, checking that their corresponding song contains that word. Checking whether there are other lines containing that word but not included in the values.
    3. Randomly selecting some values in the values, checking whether the song contains the lyric line.
    4. Checking whether the key word is in order.
    5. Removing some words with no corresponding lyric line.

- Input/Output from csv file:

    1. Checking that the file location is correct.
    2. Checking that the input data is correctly put in the ArrayList or array.
    3. Checking that the output is correct.

- Searching for the input keyword in the dictionary:

    1. Checking that binary search is operating correctly.
    2. If the keywords exist in multiple lines, checking that the output lyric line was chosen randomly.
    3. If the client input multiple keywords, checking that the output lyrics line contains all keywords.
    4. If the client inputs a word not in the dictionary, tell the client to input again.

System Test Procedures

- Formal unit testing with JUnit will be used to test the program. Testing will take place at the end of each sprint.

- All classes will have a minimum of one test.
- Tests for classes with strings will test for cases with empty string, string that shouldn't exist, and a normal case.
- Tests for classes with lists will test that list sizes match expected values and values at specific indicies match expected values.
- All expected errors will be tested for exception handling.

Adaptive Maintenance

- New function 1 : When the client inputs a word, the program can return a main lyric line(the line containing the exact keyword) and some similar lyrics for the client to choose.
- New function 2 : When the client inputs multiple words at once, the program can return a whole song according to certain algorithm.

Perfective Maintenance

1. Develop an interface based on a website, so everyone around the world has access to our program.
2. Use a database like Mysql to store our datasets.