SE 3XA3: Test Report
Title of Project

Team , Team Name
Student 1 name and macid
Student 2 name and macid
Student 3 name and macid

December 7, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

This document ...

1 Functional Requirements Evaluation

2 Nonfunctional Requirements Evaluation

2.1 Usability

2.2 Performance

2.3 etc.

3 Comparison to Existing Implementation

This section will not be appropriate for every project.

4 Unit Testing

The specific modules used for Unit testing can be found in the test folder which is in the src folder. The results for these tests can also be found in the same folder which is also linked here.

5 Changes Due to Testing

6 Automated Testing

Automated Testing was done through a combination of Mocha.JS (for unit testing) and Selenium-Webdriver (for system-wide testing).

Mocha.JS tested various pure functions throughout the codebase, based on a predefined set of input and output test vectors.

Selenium-Webdriver was used to produce a firefox instance, simulate a connection to the server, simulate user interaction, and analyze the HTML output to ensure the server is producing the correct data, and that the web client is receiving and parsing the data correctly.

6.1 Specific System Tests

Reads songs from music folder	
Initial State:	Library module called to read songs from a folder
Input:	Folder with songs
Output:	List of all the songs in the folder

Reads metadata from a song	
Initial State:	Metadata module called to read the metadata from a music file
Input:	Music file
Output:	Correct metadata information extracted from file

Voting System returns highest voted item	
Initial State:	Multiple users cast their votes, voter module is called
Input:	List of votes
Output:	Returns highest rated item

Voting System handles an empty songs array	
Initial State:	Voter module is called
Input:	Empty array of songs, non-empty array of votes
Output:	Returns empty string

Voting System handles an empty votes array	
Initial State:	Voter module is called
Input:	Empty array of votes, non-empty array of songs
Output:	Returns empty string

Webpage Title is Loaded	
Initial State:	Server is running, browser directed to webpage
Input:	N/A
Output:	Correct title of browser window is displayed

Loads the first button	
Initial State:	Server is running, browser directed to webpage
Input:	N/A
Output:	First button has the correct name

Loads the second button	
Initial State:	Server is running, browser directed to webpage
Input:	N/A
Output:	Second button has the correct name

Loads the third button	
Initial State:	Server is running, browser directed to webpage
Input:	N/A
Output:	Third button has the correct name

Initially sets first vote to zero	
Initial State:	Server is running, browser directed to webpage
Input:	N/A
Output:	First vote-count element has a value of 0

Initially sets second vote to zero	
Initial State:	Server is running, browser directed to webpage
Input:	N/A
Output:	Second vote-count element has a value of 0

Initially sets third vote to zero	
Initial State:	Server is running, browser directed to webpage
Input:	N/A
Output:	Third vote-count element has a value of 0

Votes for an item when a user clicks a button	
Initial State:	Server is running, browser directed to webpage, a vote button is clicked
Input:	N/A
Output:	Vote count for the corresponding button has a value of 1

7 Trace to Requirements

8 Trace to Modules

9 Code Coverage Metrics