

Software Implementation and Testing Document

For

Group 10

Version 1.0

Authors:

Krenn Intharaksa
Josiah Walters
Aditi Anandjiwala
Jacob Holleman
Philip Reece Sablan

1. Programming Languages (5 points)

List the programming languages used in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).

- Javascript
 - React
 - Used to build the interface of our web app
 - Will be used throughout the entire app
 - Works very well with the other languages/frameworks we are using (CSS,HTML,Node)
 - Node
 - Will be used to add features such as chat, authentication, and live user data
 - works well in tandem with frameworks like React
- CSS
 - Used to style the interface that is built by React
 - Works very well together with HTML and React
- HTML
 - Used for adding text onto the pages
 - Works very well together with CSS and React
- SQL
 - Used to communicate with our database

2. Platforms, APIs, Databases, and other technologies used (5 points)

List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).

- MongoDB
 - Our database of choice to hold all of our data
- Spotify API
 - Will be used to search for songs, albums, or artists through built in methods
 - Will be used in combination with Node

3. Execution-based Functional Testing (10 points)

Describe how/if you performed functional testing for your project (i.e., tested for the **functional requirements** listed in your RD).

- We have tested account creation and loading the account information into a database
- We can login to the account if the information provided is present in the database (proper authentication required still)
- *Many of the other requirements are yet to be implemented

4. Execution-based Non-Functional Testing (10 points)

Describe how/if you performed non-functional testing for your project (i.e., tested for the **non-functional requirements** listed in your RD).

- So far one of the only non-functional testing requirements we have been able to focus on is the intuitive ease-of-use of the website for new users. The search bar and sign in options are always listed at the top of the page and clearly readable as well as the JukeBoxd self-titled link to the homepage at the tab at the top.
- We have also focused on keeping code and documentation neat and organized so it is clear to see what we have accomplished so far.

5. Non-Execution-based Testing (10 points)

Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).

- We have not performed many reviews, inspections, or walkthroughs. There is not a huge chunk of code for us to review at this point, however we did review a good bit of code Aditi provided for the website and made sure it would work for our project and what we want to accomplish.