**Software Implementation and Testing Document**

**For**

**Group <jingle>**

Version 1.0

**Authors**:

Dante Coupet

Jack Splaine

John Barden

Gary Bowen

Kohin Khandwalla

# Programming Languages (5 points)

Framework: Django. This makes up the foundation of our project. We chose Django because it was the easiest for us to learn and implement since none of us have any experience building web apps.

Languages: Python. We are using python to develop the project because that is what our framework uses. Python is also very easy when messing with lots of data. Program is small so doesn’t need to use a super-efficient language

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

APIs: Spotify API; Genius API, YouTube API;

The Spotify API uses the song name that is searched and returns:

1. song name

2. artist name

3. album name

4. song duration

5. genre

A function returns the top 10 results.

The Genius API scrapes lyrics from Genius to get whichever song is searched.

The YouTube API is used to get the most related videos from YouTube about that song

Spotify API shows top popular songs as well as their info.

Database: SQLite for storing the most searched songs.

We are using HTTP requests to get user input.

Generated a Top 50 page.

Implemented the About page.

Feedback page.

# Execution-based Functional Testing (10 points)

We would test functions or classes as they were added to the code, making sure they didn’t interfere with any code that has already checked. This usually included back end testing of components without linking them to the front end to make debugging much simpler

# Execution-based Non-Functional Testing (10 points)

Non function requirements were mainly the looks and feel of website. We tested these by trying out a variety of colors and designs to find what looked best. We then would test the user components like buttons and search bars by linking them to simple back end components to ensure their functionality

# Non-Execution-based Testing (10 points)

No code reviews. But walkthroughs each time a segment was added to the main code, explaining how it functions both on front and backend in its simplest form so all members know its functionality