**Summary report**

# Member:

Jianan Wen 001678266 ,

Wenbo Sun 001994516,

Kaiying Zhang 001222657,

Yipeng Hong 001228971

# Project name:

ListentMusic

# Summary:

Have no idea what to listen? Tired to select music yourself?

Now, no matter where you are, you will be recommended a list of songs by just taking photo with whatever you want. After entering a photo that you take, the application will find the songs from the database by analyzing the features of the image and then recommend a list of songs to user.

# Use cases:

1. When you are travelling, you also want to enjoy the view of the landscape with some background music. But do you have any idea what kind of music is suitable for the view that can make you more enjoyable? By taking a photo of the view, the application will recommend a list of songs to you and these songs are suitable for the background.
2. People want some songs to be played in a wonderful party! Right?! But most of people have no idea what kinds of music are good for the party. Do not worry, by taking a simple picture of the party, a list of songs will be recommended and played in the party that it seems everything is perfect.
3. Are you lazy to decide what kind of music to be played? Especially you need different kinds of songs to played when you are doing different things. Now, by taking a simple photo of your current circumstance, a list of songs will be played automatically by the application according to the photo.

# Challenges:

1. According to the requirement of the application, we need our own dataset that we can analyze certain conditions. So, how to build up our own dataset?
2. After taking a photo, how to analyze these images and get specific features from it?
3. To recommend a precise list of songs in a short time, what kind of format of database we need?

# Research:

1. Crawler Technology: By wring python program and knowing the format of html of website, a specific dataset can be built will specific features.
2. R-Data analysis: We need to analyze the dataset to get more specific useful features from the original dataset. And we also want to analyze the emotions of the songs. R can analyze the dataset by using several libraries.
3. Database selection (MongoDB): The database should be used in a convenient and easy way, so we need a kind of database in a format that can easily be queried in a short time. And what we know is that NoSQL is good way for us.

# Approaches:

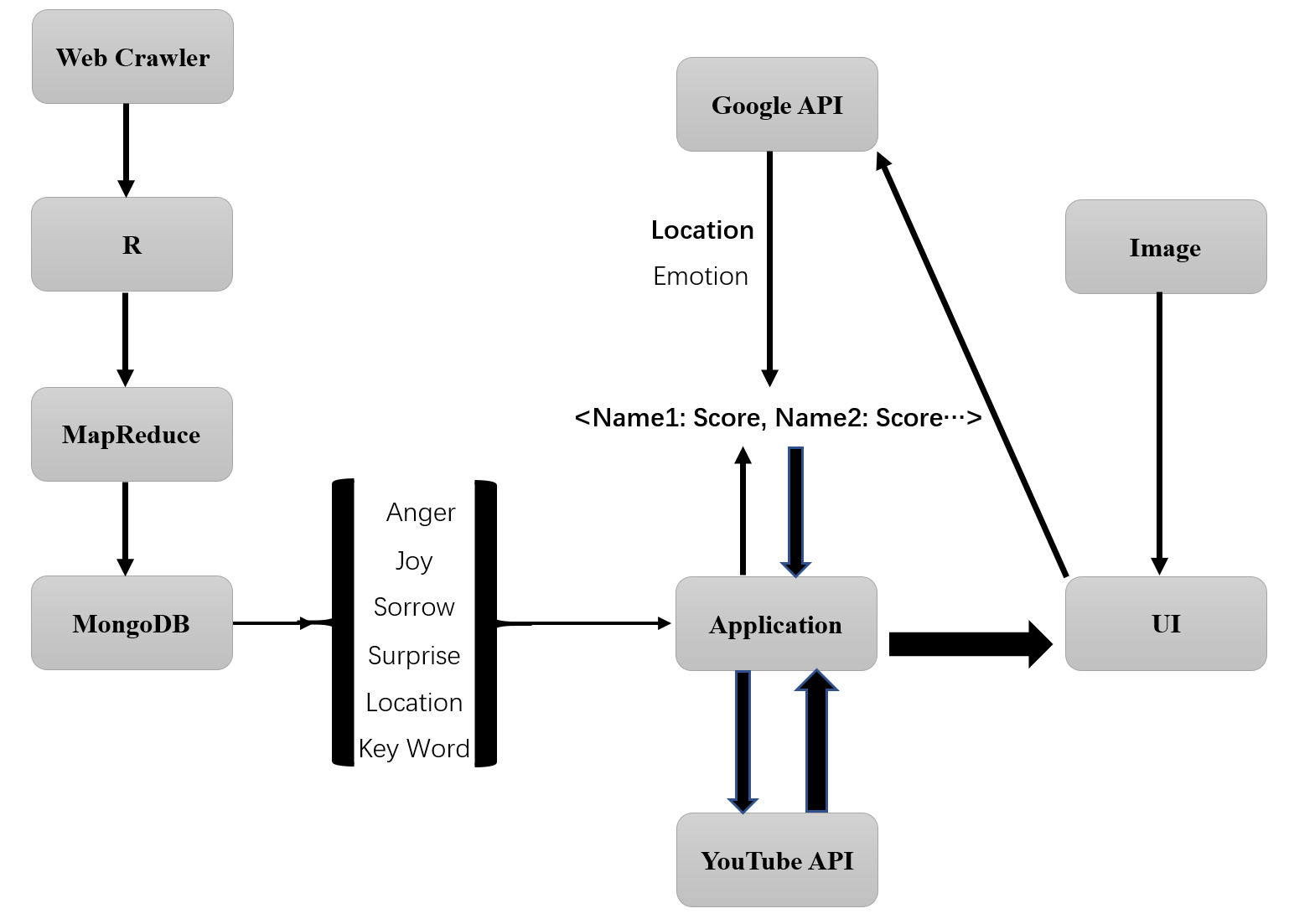
1. For building our own dataset, we can use Web Crawler which can copy specific data from specified website without some useless information.
2. By using R with specific libraries, we can analyze dataset and get the emotions or other useful features from it.
3. For analyzing images, Google API is a good choice that can get some precise data of an image even the emotion of the person of the image.
4. Before using the data, we still need to transfer it to some format that can be easily used. At same time, a best way is required to deal with these big and complicated data. So, we use MapReduce to deal with the dataset.

# Future scope:

1. We want to expand the size of the database, and update it periodically. So, the application can recommend more exhaustive list of songs.
2. For the recommendation, we use our own algorithm which is not precise enough to recommend songs to user. We want to optimize it, so it can run faster and get a more exhaustive recommendation.
3. We want to add a function that can record user’s history. So, when the old user come back, the application will recommend the songs depending on not only the analysis of the image but also the history of the user.

# Process

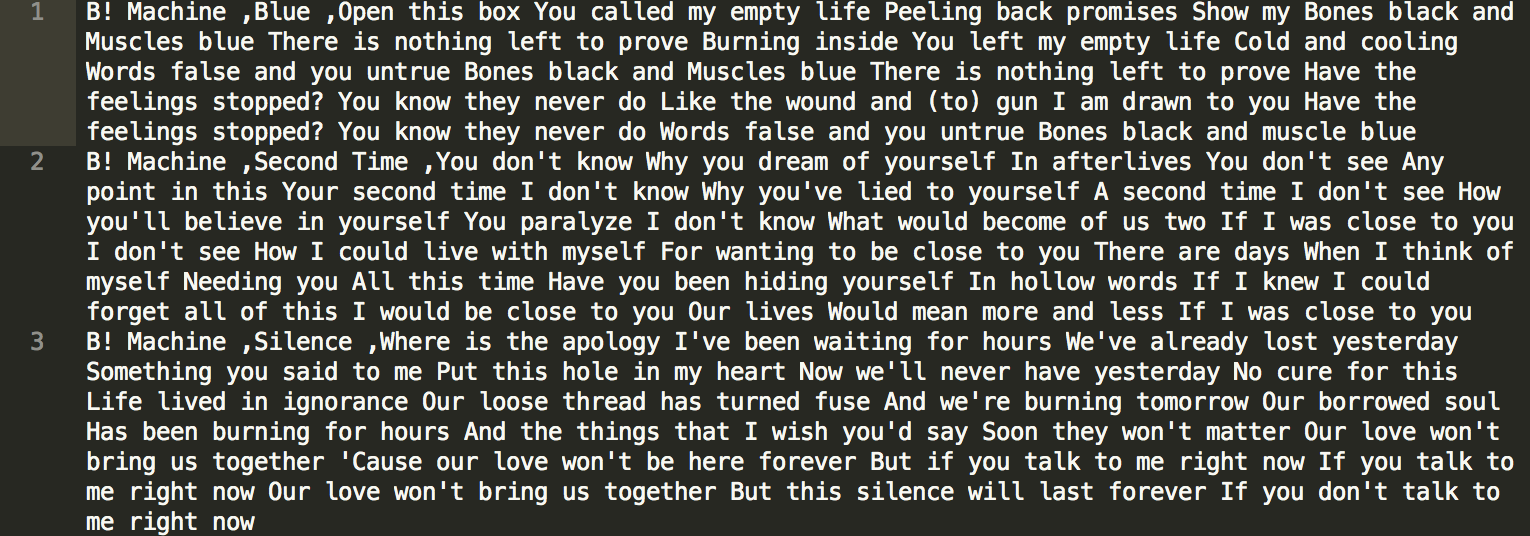
# Structure



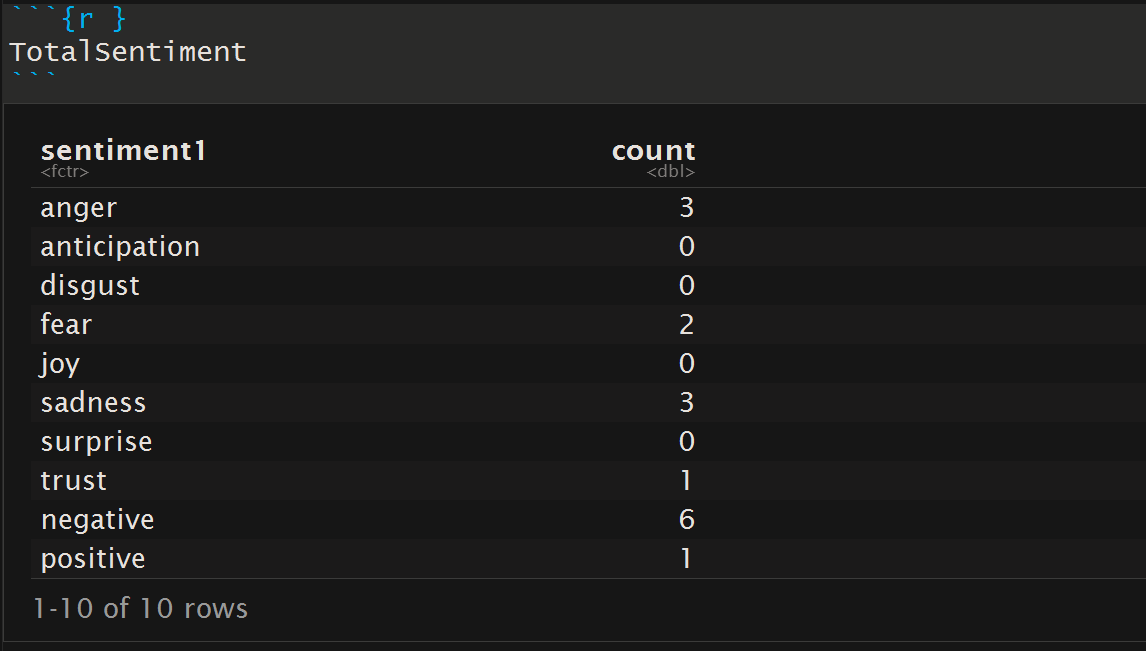
# Web Crawler for building dataset

A Web crawler, sometimes called a spider, is an Internet bot that systematically browses the World Wide Web. We use Web Crawler to copy all data from the website <http://lyrics.wikia.com/wiki/Lyrics_Wiki> . For our project, we need to build our own dataset including name of songs, artist of songs, lyrics of songs by using python. For mac, we need to install lxml in terminal and write python program so we can use format of Crawler program. For copying data from the website, we need to use the xpath of the web page and the format of the html. Then, you can capture data from the website through certain order. We create a 1.13G dataset to be used. If you want to have a bigger dataset, you can use Crawler Technology to capture more data.

The lyrics of the songs will be used to be analyzed by RStudio, and this dataset will be transferred to another kind of format that can be easily used.

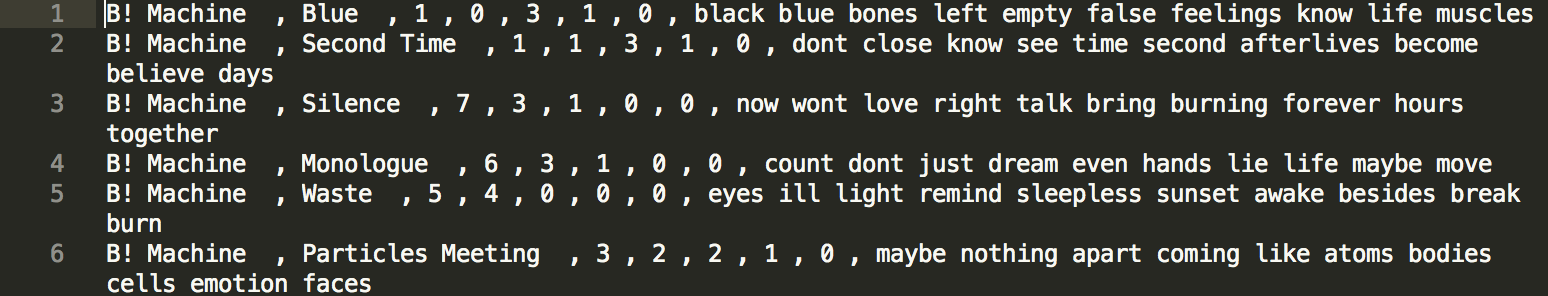


**R-Data mining**

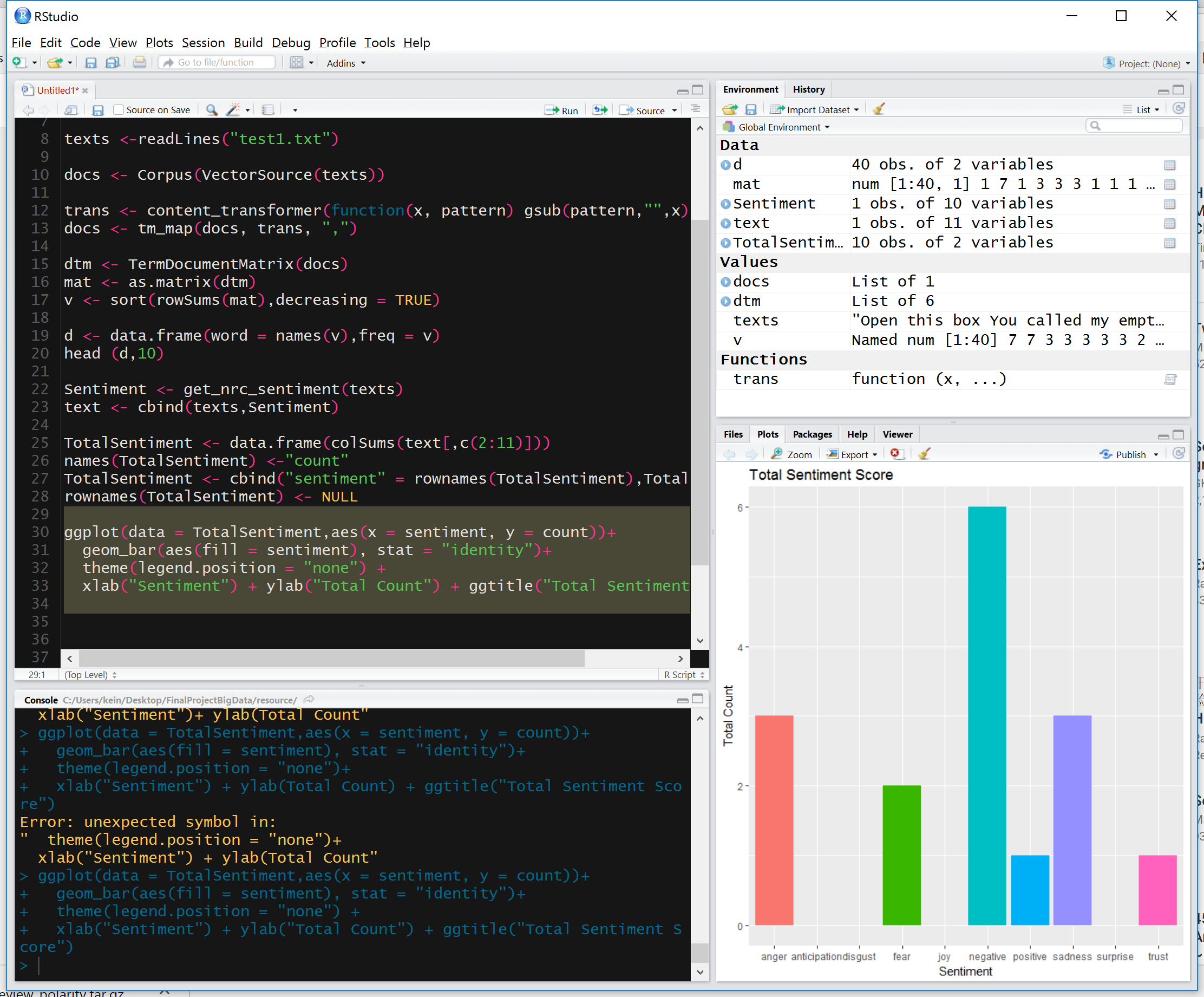


Input format: The output of the Web Crawler will be the input of R.

Output format: Including 8 columns. First is the artist of the song, second is the name of the song, the next 5 columns are the scores (1-10) of the emotion (Joy, Anger, Sorrow, Surprise, Positive & Negative), the last column is the collection of the song’s key words.



By using R Sentiment analysis with specific libraries, we can get the sentiment from the lyrics analysis with specific scores. We set the scope of scores from 0 to 10. The sentiment become stronger as the score increasing. And the key words come from the lyrics with highest rate of occurrence.



# MapReduce

The input’s every line is in format name of song, name of artist, four column scores (0-10) of the emotions (Joy, Sorrow, Anger, Surprise), ten key words of the lyrics of the songs. Each score and the key word will be the key word and the connection of artist and name of song will be the value.

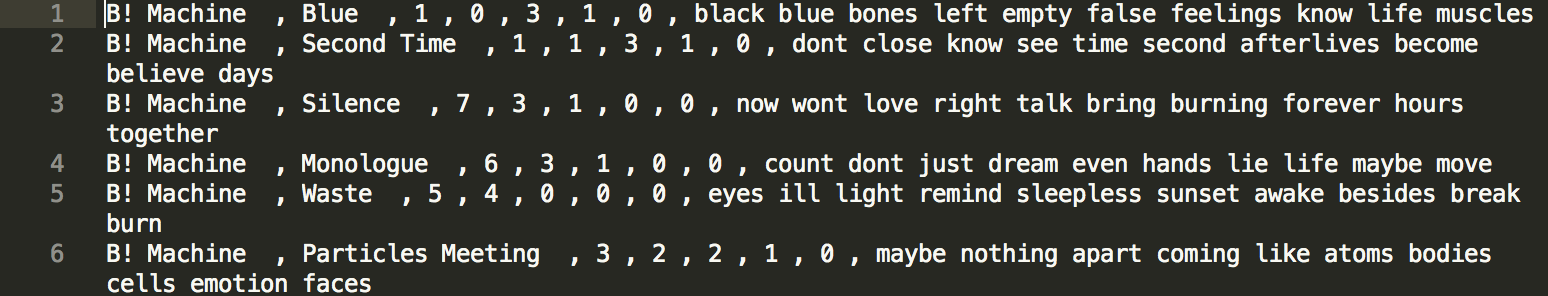
For the next use, the output’s format will be label, list of connection of name and artist. In order to let these labels be distinguished, a string ‘SooooooooogName>’ is added in front of the key.

The final output of the MapReduce will be imported to mongodb (json) and used by the application.

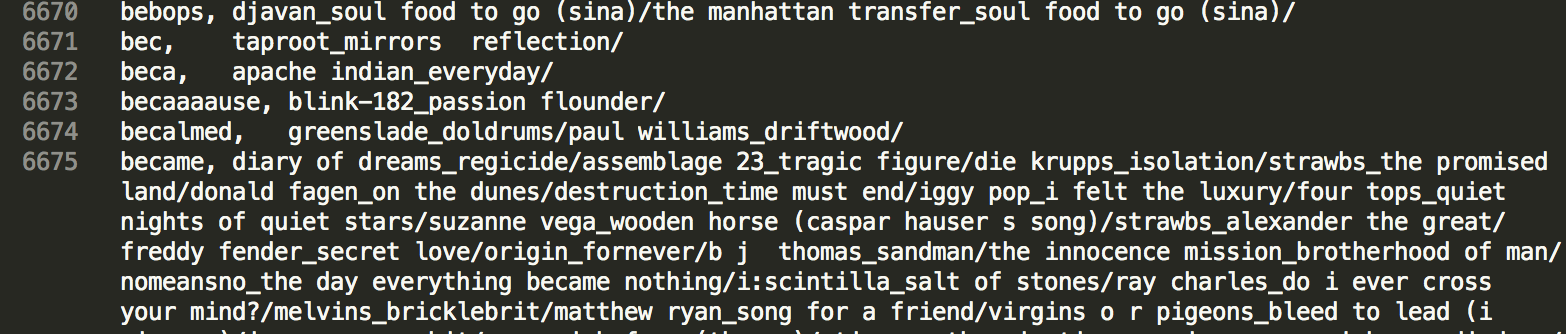
Map: Use each line’s score and key word as key, and the connection of the name and the artist will be the value. At same time, every word of the name of the song will also be the key and the artist will be the value.

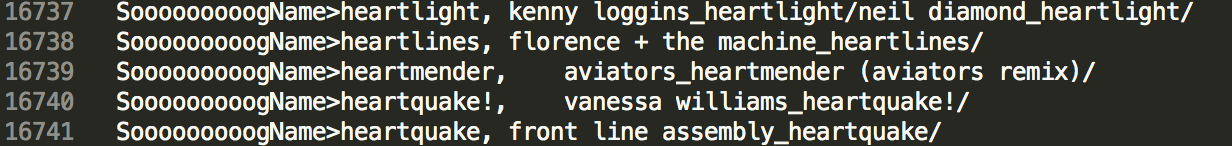
Reduce: Connect each value with same key and eliminate unnecessary symbol.

Input: ---73M



output: ---103M





# Photo analysis

We use Google vision API to analyze the input picture.(https://cloud.google.com/vision)

This API can analyze picture and get a lot of data base on the characteristics of the picture.

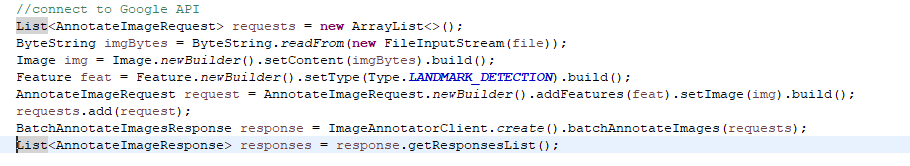
To use the Google API, you need to:

1. Install the Google SDK, which connect your code to Google server.(https://cloud.google.com/sdk/)
2. Go to you Google cloud console, create a project and enable the Google vision API on that project.
3. Go to API manger-Credentials and create a services account key. You will get a Json file. Keep it somewhere, you will need it latter.
4. Open a Terminal, type: gcloud auth application-default login

You will be asked to login your google account.

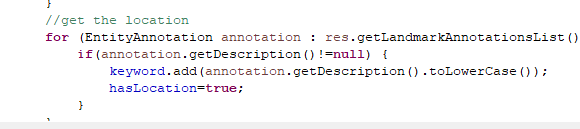
1. After login, you will see the default Credentials has been save in some folder. Find that folder and copy the previous Credentials into the folder replace the old one.

Now, you have the environment all set to run Google vision API.



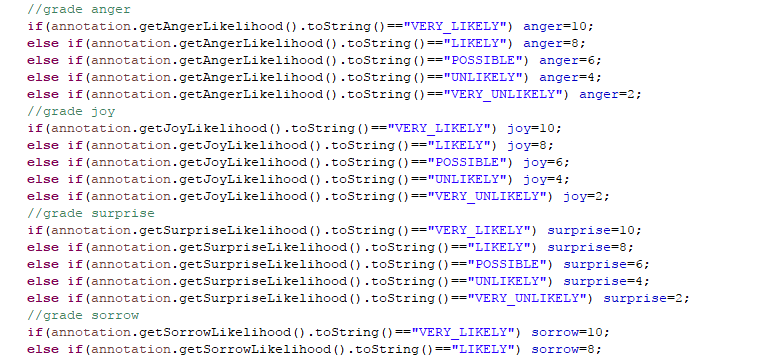
I use this code to connect to Google vision API.

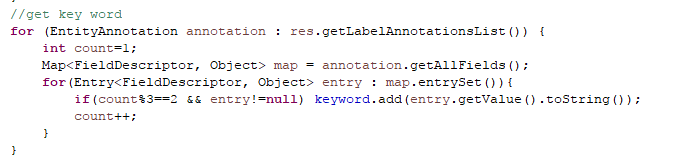
However, there so many information from Google vision API, we need to pick up what we need.

So I get four method to get emotion, location, keyword, and color.

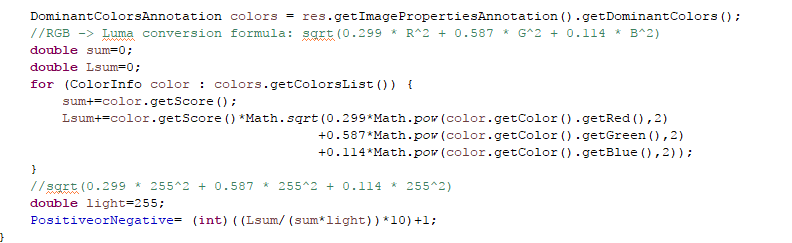
When the picture is a human face, we will grade the emotions.

We assign a score (out of 10) to each emotion(joy, surprise, sorrow, and anger)

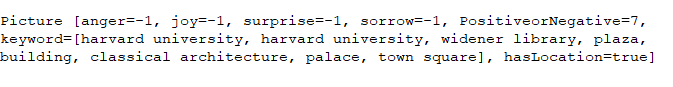




When the picture is not a human face, we convert the RGB to brightness to decide to the sentiment.



Here is the result of picture processing :

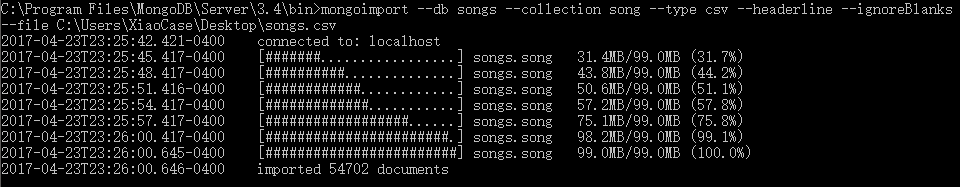


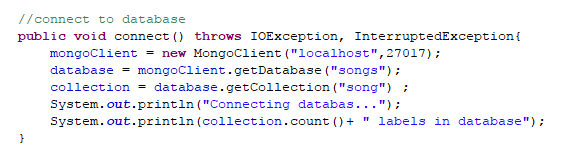
Emotion is -1 because it is not a human face.

# Build Database

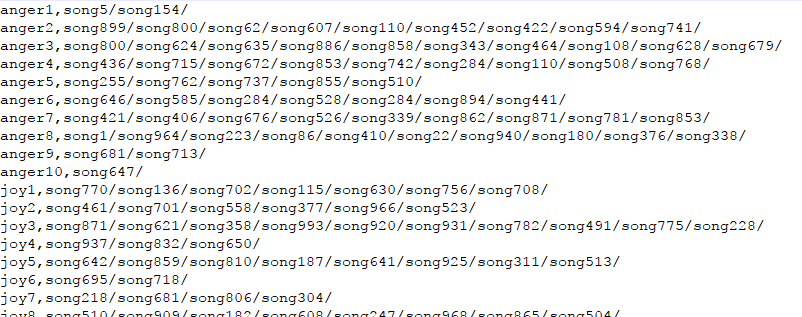
We use MongoDB as database.

MongoDB can directly import a csv file into database:



The we use this code to connect to database:

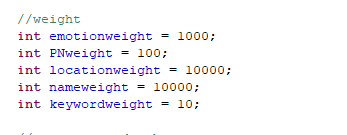
The database looks like this:



(This is a simulate data)

**Grade Songs**

We assign each emotion and keyword with different weight:



This weight make sure the song name is the most important concern.

Location is the second.

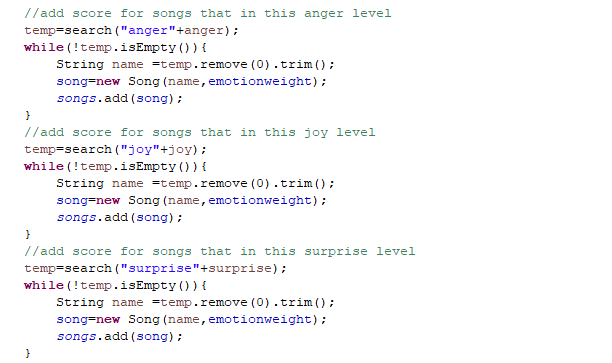
Emotion is the third.

For example,

if the picture’s joy score is 7, we will search for label “joy7”, the give all songs in this label 1000 point.

If the picture has location information ”boston” , we will give all songs in “boston” lable 10000 point.

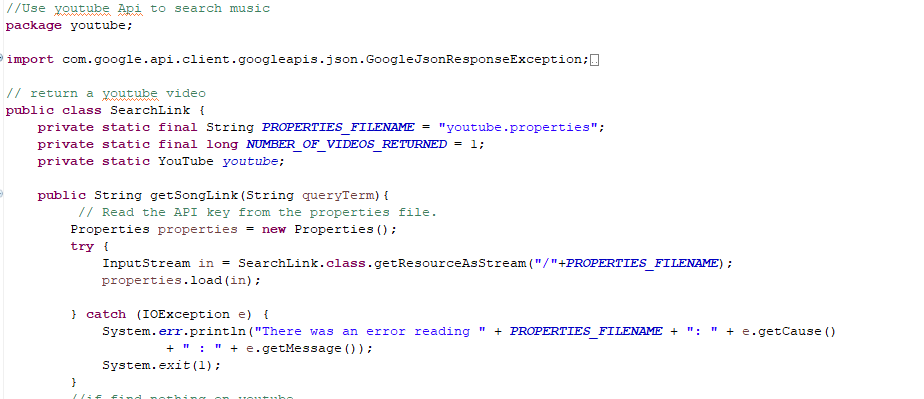
If the picture’s first three keyword has “sky”, we will give all songs in “sky” 10000 point.



At the end we will recommend top five songs.

Youtube API

We used Youtube API to play those Songs



Servlet UI to present the video