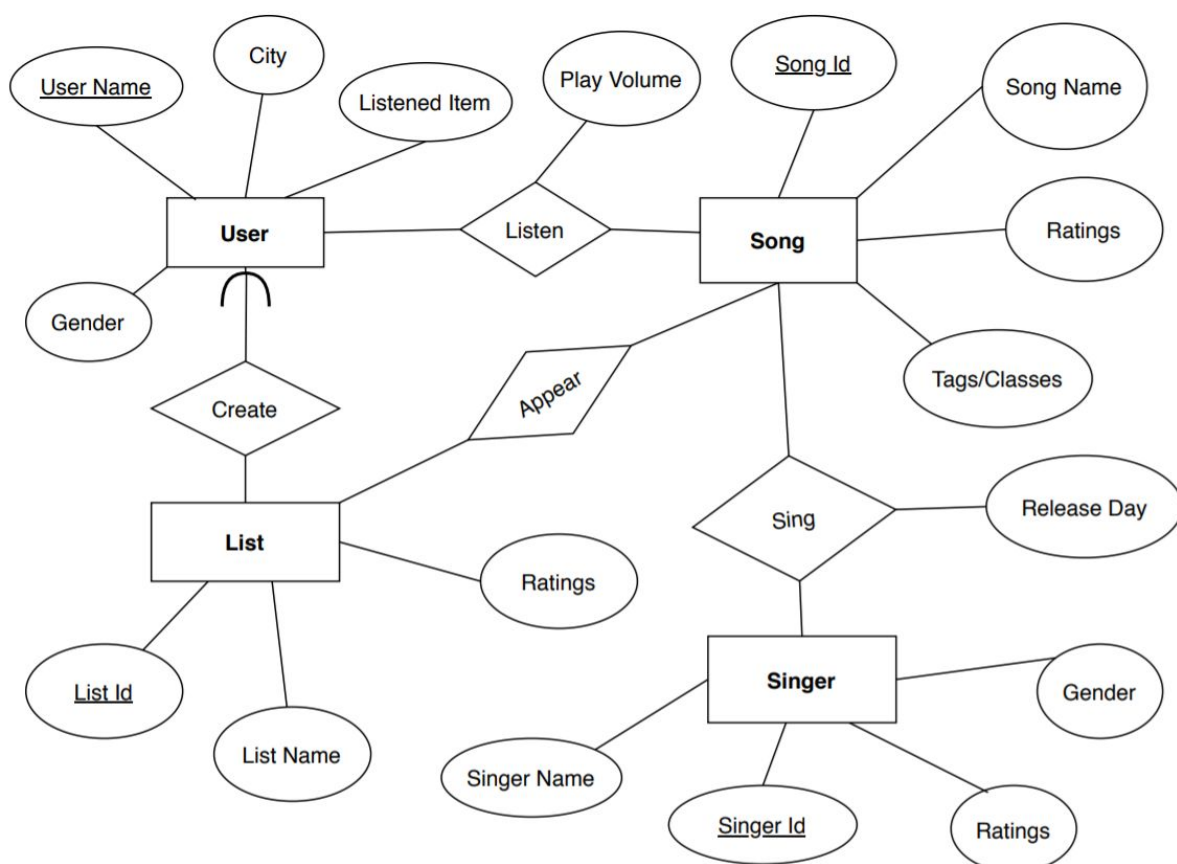


1. The website called “Songs leading friends” helps users search for their interested songs and singers, also recommends songs and other users as potential friends to them according to their preferences.
2. The website allows users to find their interested songs and helps them expand their interested music field by recommending songs and friends sharing common interests with them. There are some similar applications, like some music playing apps, to the project we tend to make. The unique point of our project is that users could set their own matrices to give ratings to songs, and the website will recommend songs based on the users’ preferences.
3. Our database has five tables. One of them is the song table. The data are derived from a csv file of top 2000 songs in Spotify. Another table, weather_genre shows the corresponding relationship between weather and genre. The data in other tables, rating, adding and users, are generated by users in the website.
- 4.



5. The data in our song table are derived from a csv file (from Kaggle) containing the information of top 2000 songs in Spotify. We used python to do some data processing: we get all songs which are in genre having more than 10 songs and exported the csv file as sql file and import it into phpmyadmin.
6.
 - 1> users can query songs according to its singer, its songname, its genre and the range of its release year;
 - 2> users can rate whatever songs they would like to rate, including inserting new ratings, updating the original ratings and cancel the original ratings; users can rate multiple songs at the same time or they can rate songs individually in the page specifically for each song;
 - 3> users can add songs to their song list and also remove songs from song list if they want;

4> users can view the statistics of each songs (average raings, max and min ratings, the amount of people rate and add to the song list), and they can also view the statistics concerning their own music preference(favorite songs, favorite singer and favorite genre)

5> users can use recommend friends function to know some friends who shared the similar music preference with them. Users can also view their friends' song list and music preference and add their friends song to their song list

6>users can use recommend songs function based on local weather. They will get songs with different BeatsPerMinute and genre.

7. I will explain the query function here.If users put in SingerName and Genre in the search page, then the corresponding songs will appear in the following page, with the information of songs with joining rating table.

```
8. $sql = "SELECT * FROM song LEFT JOIN (SELECT * FROM rating WHERE
userid=$searchuser) AS temp ON song.SongID=temp.songid WHERE Singer LIKE
'%'$singername%' AND SongName LIKE '%'$songname%' AND ReleaseDate>='$year1' AND
Genre LIKE '%'$genre%' LIMIT 900";
```

```
$sql = "SELECT * FROM song LEFT JOIN (SELECT * FROM rating WHERE
userid=$searchuser) AS temp ON song.SongID=temp.songid WHERE Singer LIKE
'%'$singername%' AND SongName LIKE '%'$songname%' AND ReleaseDate>='$year1' AND
ReleaseDate<='$year2' AND Genre LIKE '%'$genre%' LIMIT 900";
```

```
$updatesql = "UPDATE rating SET ratings=$NewRating WHERE songid=$currentsong AND
userid=$searchuser;";
```

```
$insertsql="INSERT rating
(userid,songid,ratings)VALUES('$searchuser','$currentsong','$NewRating');";
```

```
$deletesql=DELETE FROM rating WHERE songid=$currentsong AND userid=$searchuser;
```

```
$get_name_sql="SELECT SongName FROM song WHERE SongID = $song;";
```

```
$count_user_sql="SELECT count(userid) FROM rating WHERE songid=$song;";
```

```
$count_add_sql="SELECT count(userid) FROM adding WHERE songid=$song;";
```

```
$max_rating_sql="SELECT max(ratings) FROM rating WHERE songid=$song;";
```

```
$min_rating_sql="SELECT min(ratings) FROM rating WHERE songid=$song;";
```

9.Firstly,we sign up a new account and insert a new record in users table:

If we search song table by singer “Jay Chou” and genre “Pop”

[Show All Songs](#)

Hi, **liju2**
[Log Out](#)

Location: Shanghai
6th August, 2020
 37.78°C / 32°C
Weather Description: overcast clouds,

[Recommend Songs to You](#)
[Recommend New Friends to You](#)
[Add Songs That Have Not Been Included](#)

Search By Singer:

Search By Song Name:

Search By Song Genre:

Search By Release Year:
From To

So the result shows in the following page:

[Hi, liju2](#)
[Log Out](#)

Song Information and Rating System

List	Individual	SongName	Singer	Year	Genre	Previous	RatingInput	NewRating	AvgRating
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Mine Mine	Jay Chou	2011	Electronic Music, Pop, R&B		<input type="text"/>	No Input	0
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Mojito	Jay Chou	2020	Rumba, classical, Pop		<input type="text"/>	No Input	0

If you click show all songs

[Hi, liju2](#)
[Log Out](#)

Song Information and Rating System

List	Individual	SongName	Singer	Year	Genre	Previous	RatingInput	NewRating	AvgRating
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Sunrise	Norah Jones	2004	adult standards	9	<input type="text"/>	Canceled	5.7
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Love Me Tender	Elvis Presley	2002	adult standards	6.9	<input type="text"/>	No Input	5.81
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Dont Know Why	Norah Jones	2002	adult standards	6	<input type="text"/>	No Input	4.13
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	"You're The First, The Last, My Everything"	Barry White	2009	adult standards	9	<input type="text"/>	7.5	8.01
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Papa Was A Rolling Stone	The Temptations	2001	adult standards	8.9	<input type="text"/>	No Input	6.31
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Home	Michael Bubl�	2005	adult standards	9.9	<input type="text"/>	No Input	8.41
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	A Little Less Conversation - JXL Radio Edit Remix	Elvis Presley	2002	adult standards		<input type="text"/>	8	5.97
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Everything	Michael Bubl�	2007	adult standards	8.6	<input type="text"/>	No Input	6.33
<input type="button" value="Add"/>	<input type="button" value="To Rate"/>	Heaven Must Be Missing An Angel - Pt. 1	Tavares	2004	adult standards	9	<input type="text"/>	No Input	7.49

This is the personal page for the current user, you can see the user's song list and user's music preference statistic. You can also update profile email and location:

FrienZongs

Hi, Iijju2

Log Out

My Song List

SongName	SingerName	Remove
Sunrise	Norah Jones	Remove
I Just Cant Help Believin	Elvis Presley	Remove
A Whiter Shade of Pale - Original Single Version	Procol Harum	Remove
Lola Montez	Volbeat	Remove
Lemon Tree	Fools Garden	Remove
Now You See Me	Jay Chou	Remove
Mojito	Jay Chou	Remove

Personal Information

Information	Information content
Current Email	wenboye2@illinois.edu
Current Location	Chicago
Highest Rated Song	Spicks And Specks
Highest Rated Singer	Bee Gees
Highest Rated Genre	german pop rock

Update Profile

Please fill in your information to get real friends.

Email

Location

submit

For the recommend friends, you need to put four parameters for weight:

FrienZongs

Hi, Iijju2

Log Out

Singer Weight:

Genre Weight:

BPM Weight:

Danceability Weight:

submit

Friends List

Friend	Score	Email	Location	His Song List
user4	19.803	user4ever@illinois.edu	Tokoyo	View
user12	18.826	xzhu15913@gmail.com	NYC	View
user11	16.317	1single1@hku.edu	Hong Kong University	View

If you click the view you can see your friends' song list, you can add your friend's songs:

FrienZongs

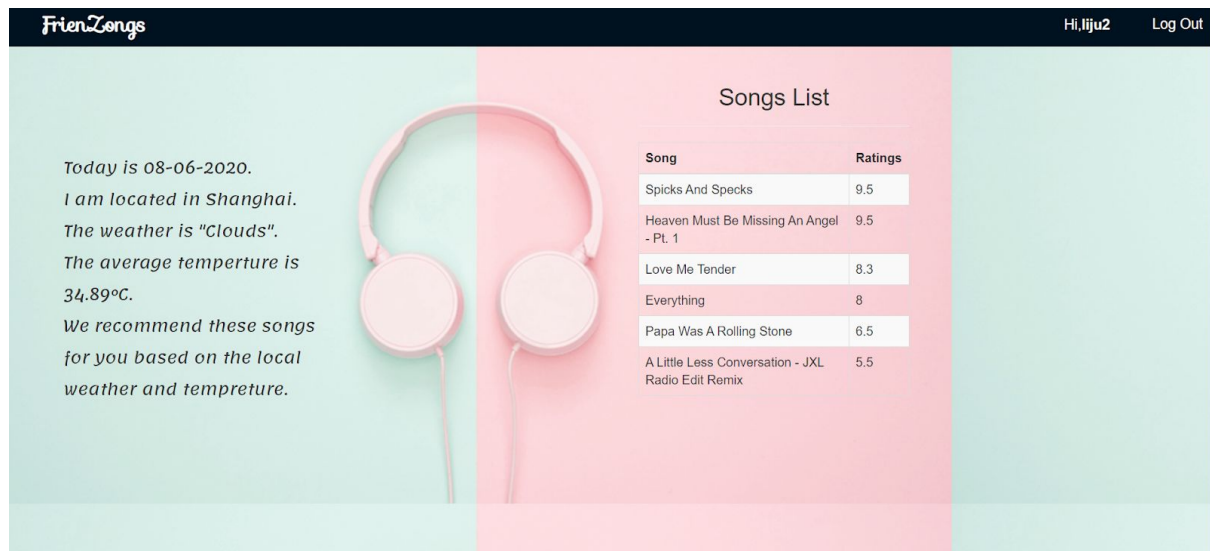
Hi, Iijju2

Log Out

user4's Song List

SongName	Singer	Year	Genre	Add
Sunrise	Norah Jones	2004	adult standards	Add
"Youre The First, The Last, My Everything"	Barry White	2009	adult standards	Add
"Oh, Pretty Woman"	Roy Orbison	1962	adult standards	Add
The Man Who Sold the World - 2015 Remaster	David Bowie	1970	album rock	Add
I Want To Hold Your Hand - Remastered 2015	The Beatles	2000	british invasion	Add

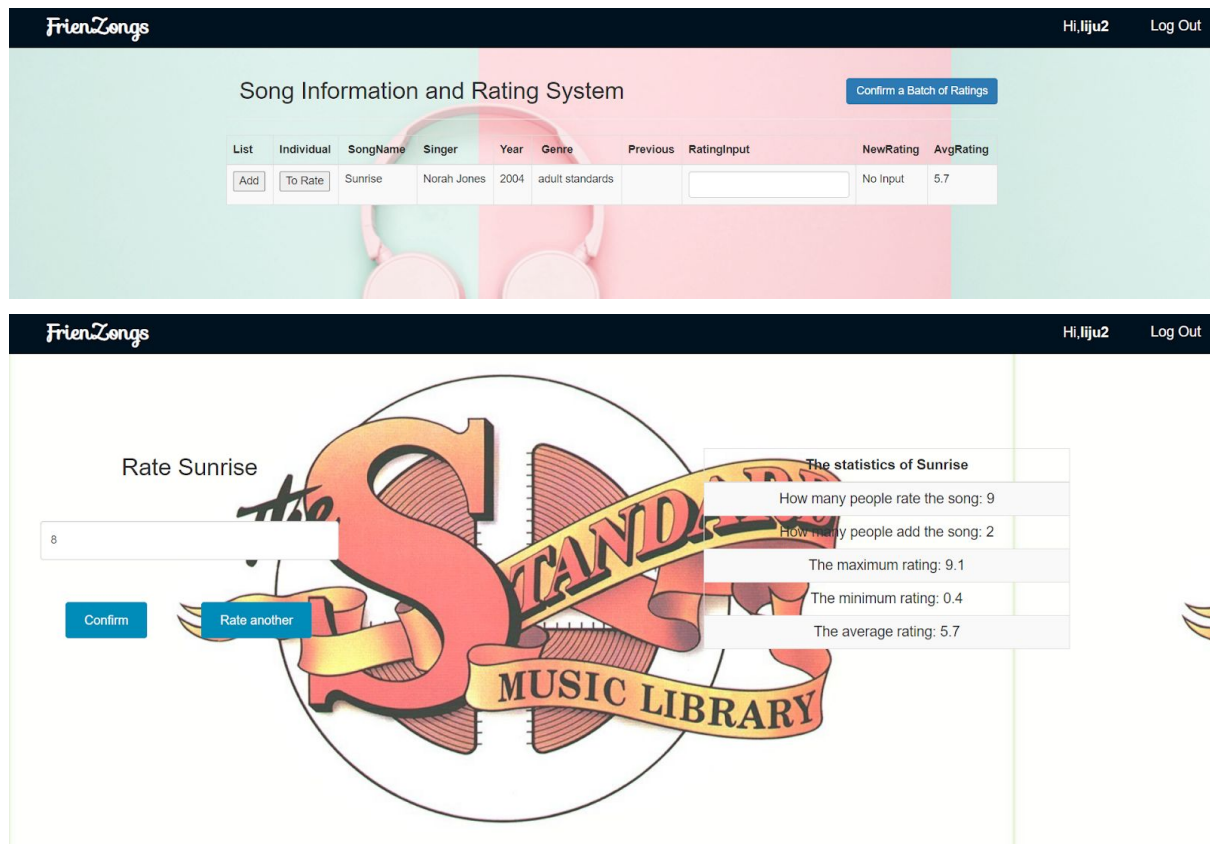
This is the recommend songs function based on weather:



11. The technical challenge we met is about how to conceive a reasonable algorithm when we were developing the algorithm of advanced function. For the first part of advanced function, recommending friends, we need to calculate the difference of ratings between the current user and other users. To arrange our rating data in a decent way, we designed four matrices, whose row dimension comprised of all users, and whose column dimension comprised of all singers, or all genres, or all BeatsPerMinute or all Danceability according to the recommend standard(parameter) which users put the the input boxes. In each grid of the above four matrices, we put in average ratings of each user for each genre, or each singer, or each kind of BeatsPerMinute, or each kind of danceability. Once we acquired the above four matrices, we can calculate the square difference of average ratings between the current users and each other user. Then we can get the summation of $\text{weight}_i \times \text{square difference}(i)$ (from 1 to 4, the parameters we put for each recommending standard), and based on that summation we calculate a score for each other users(who are not the current use) to show the similarity extent. Once we got the score, we use `array_multisort` function to rearrange friends based on their score in descending order and finally print out the friends table. Another difficulty we met is how to make the website memorize which song we click "To Rate" or "Add" button. For "To Rate" button, we need to go to another page specifically for the clicked song. We have designed a special button with the following structure:

```
<td><form action='test_following.php' method='post'>
<input type='hidden' name='id' value=$currentsong>
<button type='submit' name='sub' value='subed'>To&nbsp;Rate</button>
</form></td>";
```

to deliver the values of "\$currentsong" to the "test_following.php" page and in that page, we store it as a session variable to make sure "\$currentsong" is accessible to all pages.



13. Wenbo Ye, Li Ju and Mengxuan Yu are responsible for the basic function work. Wenbo Ye developed the query function, and Wenbo Ye, Li Ju and Mengxuan Yu developed insert, update and delete functions together. Wenbo Ye, Li Ju and Yuhang Chen cooperated on the advanced function development. Yuhang Chen designed algorithms for the functions of recommending friends and recommending songs. Yuhang Chen, Li Ju Wenbo Ye worked on the back-end work for the advanced functions. Wenbo Ye did the front-end development work. Yuhang Chen and Mengxuan Yu cooperated to get the real data. We teammates organized daily discussion to help teammates debug and resolve some problems met in the website developing process. In the daily discussion, we also

10. We make two advanced functions in this project.

1) For the advanced function of recommending friends to user:

Initially we want to recommend friends to user with similar preference on music. So we learn some algorithm about recommendation and finally we choose to use **collaborative filtering algorithm**. And we use **weight coefficients** (range in float (0-1)) to represent the weight percentage of 4 parameters to this recommendation. In this function, the parameters include average ratings on the particular same singers, average ratings on the particular genres, average ratings on the particular range of danceability and average ratings on the particular range of danceability. Moreover, we give users free choice on the weight of each parameter that counts on the algorithm. We implement this algorithm by calculating the

similarity difference and store the difference result in 4 matrices then use feature matching method to get the **recommended scores** (range in (0- 100) of other users compared to the user that login in the websites. Then We make a table in html to show the scores desc with the relative username.

In this way we succeed to recommend friends to the user. And we also add buttons in each row (exclude the table head) to view the **personal page** of other recommended users. Therefore, the user can get the basic information including contact information like email and the songlist of his/her recommended friends. And the user can add some songs in the songlist of their friends to his/her personal song list.

2) For the advanced function of recommending songs to user:

Initially we want to recommend friends to the current user based on mood. We implement this by getting the **local real-time information** of the user because the environment that the user stays has a large effect on their mood. The first thing we do is to get the ip address of the current user by connecting to an api called ipstack. Then we use the ip to get the actual information (including location, the weather description, average temperature) by another api pot called openweathermap.

We divide the algorithm of this recommendation on three aspects.

First we **use \$weather to select the genre** of the recommended songs (i.e. According to some literature people show more preference on Jazz and Rock on a rainy day).

Second, we build **a mathematical model of linear regression between \$temperature and bpm of the songs**, calculate the relative coefficients (gradient k and vertical intercept b, depending on the seasons). Then we select out the bpm range of the recommended songs.

Third, we build **a model based on a regular time schedule**. We select the songs with supposed range of danceability. (i.e. If the time of the current user is between 0:00a.m. to 5:00a.m. it means he/she is staying up late so we recommend he/she songs with high danceability to excite the user and sweep out the sleepy feeling)

Then we combine these three aspects together to form our algorithm of recommending songs.

12. Up to now the websites we make work match a lot with our initial development plan.

For build of the database: We succeed to build our database with a large amount or data by transforming .csv files to .sql files using python and insert them to the local database. Initially we intend to use python scrapy. However, in real procedure we find that use python to "crawl" the data from some music app is not as efficient as the way we directly use the data(.csv) provided by some website, because it can provide more accurate information (like Danceability, "BeatsPerMinute") about the songs that contributes to the implementation of our functions.

Our desire on the basic function has been totally implemented including Add/Search/Insert/Update/Delete Func. By combine the web front end(html + css) and the web backend(PHP) we can do basic operation to our mian tables(rating table and song table) our database.

Our desire on the advanced function has been approximately implemented and is updated with more new fancy idea each day before the final demo. We use weight choice and collaborative filtering algorithm to build our function of recommending friends to users. We use api pots to get the users' information and recommend songs to users according to the weather and time. During these procedure we make progress on the accuracy and rationality

by search on the internet and comparing the result with relative literature (like How can weather influence people' preference on choice about songs). Two advanced funtions strengthen the practicability of our website.

The website may work as 60% of our initial plan. And the left 40% parts are totally new, which makes our project more interesting.