

**上海海事大学**

SHANGHAI MARITIME

UNIVERSITY

**听歌识曲系统**

**Audio Recognizer System**

**课程名称： 软件项目管理**

**组 名： 第 13 组**

**指导教师： 刘晋**

**日 期： 2019年5月**

**小组成员： 何鹏程 朱云龙 梁骏杰 陈柏山**

**Contents**

[1 Introduction 5](#_Toc9086495)

[1.1 Writing purpose 5](#_Toc9086496)

[1.2 Background 5](#_Toc9086497)

[2. Description of project 5](#_Toc9086498)

[2.1 Project objectives 5](#_Toc9086499)

[2.2 Demand analysis 6](#_Toc9086500)

[2.3 Task decomposition 6](#_Toc9086501)

[2.4 System structure drawing 7](#_Toc9086502)

[2.5 Detailed module design 7](#_Toc9086503)

[3. Job instructions (SOW) 8](#_Toc9086504)

[3.1 scope of project 8](#_Toc9086505)

[3.2 organizational structure 8](#_Toc9086506)

[3.3 Personnel allocation 9](#_Toc9086507)

[3.4 Project schedule 10](#_Toc9086508)

[3.5 Collaboration and communication 10](#_Toc9086509)

[3.5.1 Internal collaboration 10](#_Toc9086510)

[3.5.2 Project communication skills 11](#_Toc9086511)

[4 Related knowledge Learning 12](#_Toc9086512)

[5.Project required interfaces 13](#_Toc9086513)

[6.Test Plan 14](#_Toc9086514)

[6.1Test schedule 14](#_Toc9086515)

[6.2 Writing purpose 15](#_Toc9086516)

[6.3 reference 15](#_Toc9086517)

[6.4 Test risk assessment 15](#_Toc9086518)

[6.4.2 emergency measures 15](#_Toc9086519)

[6.5 Unit Test Plan 16](#_Toc9086520)

[6.5.1 purpose 16](#_Toc9086521)

[6.5.2 Test content 16](#_Toc9086522)

[6.5.3 Unit testing strategy 16](#_Toc9086523)

[6.5.4 Test tools 17](#_Toc9086524)

[6.5.5 Test module 17](#_Toc9086525)

[6.5.6 Test activity schedule 20](#_Toc9086526)

[6.5.7 Test case 20](#_Toc9086527)

[6.5.8 Test script code 21](#_Toc9086528)

[6.5.9 problem summary 24](#_Toc9086529)

[6.6 Performance Test 24](#_Toc9086530)

[6.6.1 Test tool 24](#_Toc9086531)

[6.6.2 Test process 25](#_Toc9086532)

[7.Milestones 26](#_Toc9086533)

[8.The risk plan 27](#_Toc9086534)

[9 detailed design 29](#_Toc9086535)

[9.1 writing purpose 29](#_Toc9086536)

[9.2 reference 29](#_Toc9086537)

[9.3 Terminology definition 29](#_Toc9086538)

[9.4 Key technologies and solutions 30](#_Toc9086539)

[9.5 profile design 30](#_Toc9086540)

[9.5.1 Demand profile 30](#_Toc9086541)

[9.5.2 System development environment 31](#_Toc9086542)

[9.5.3 Functional chart 31](#_Toc9086543)

[9.5.4 System flow chart 32](#_Toc9086544)

[9.6 Detailed module design 33](#_Toc9086545)

[9.6.1 System module design 33](#_Toc9086546)

[9.6.2 Interface design 33](#_Toc9086547)

[9.7 UML 35](#_Toc9086548)

[9.8 Database module design 36](#_Toc9086549)

[9.9 System error handing design 37](#_Toc9086550)

[10 summarize 38](#_Toc9086551)

[10.1 final product demo 38](#_Toc9086552)

[10.2 Project execution 38](#_Toc9086553)

[10.3 Lessons learnt 39](#_Toc9086554)

[10.4 conclusion 40](#_Toc9086555)

[10.5 Future plan 40](#_Toc9086556)

[11 Appendix 41](#_Toc9086557)

[11.1 Project code Structure 41](#_Toc9086558)

[11.2 Code 42](#_Toc9086559)

[11.2.1 The back-end code 42](#_Toc9086560)

[11.2.2 The front-end code 52](#_Toc9086561)

# 1 Introduction

## 1.1 Writing purpose

In order to ensure that the project team completes the project objectives on time with good quality, easy to project team members to better understand the project situation, a reasonable and orderly to project work to each process, it is necessary to file the form, put in the project life cycle within the scope of work, each work task decomposition, the project team structure, team members have to work responsibility, team communication inside and outside collaboration mode, progress, project content such as internal and external communication collaboration mode, progress and describe in writing a way out. So that each person in the project development process can play their strengths and role, so that the whole project development process can be smooth and orderly, and provide effective and powerful progress reference for the project development.

## 1.2 Background

With the rapid keyword retrieval technology based on massive audio content becoming more and more mature, it plays an important role in people's spirit, life and other aspects. When we overhear a song or a piece of music, but we need to know the name of the song in order to record, but there is no way to start. In order to increase the user experience, this product is a simple and easy to use music recognition application, the user can use this function to identify the current environment is playing the song information, give people in the break time to bring new surprises and experience. This is also a way to find good music, after all, a good song, worth recommending to more listeners. With it, you can explore more new music tracks with your friends and share them around the world.

# 2. Description of project

## 2.1 Project objectives

Player can quickly and accurately identify the current environment to play the song information.

## 2.2 Demand analysis

1. Do a good job of the current research on music player. Through the research on the current function of listening to and recognizing songs of music player, we can help the project more clearly conceive the general structure and appearance of the system developed by ourselves.

2.Search for technical knowledge and prepare tools and environment for the project.

3.describe system functions and make test plans.

4.Determine the interface style, coding language.

## 2.3 Task decomposition

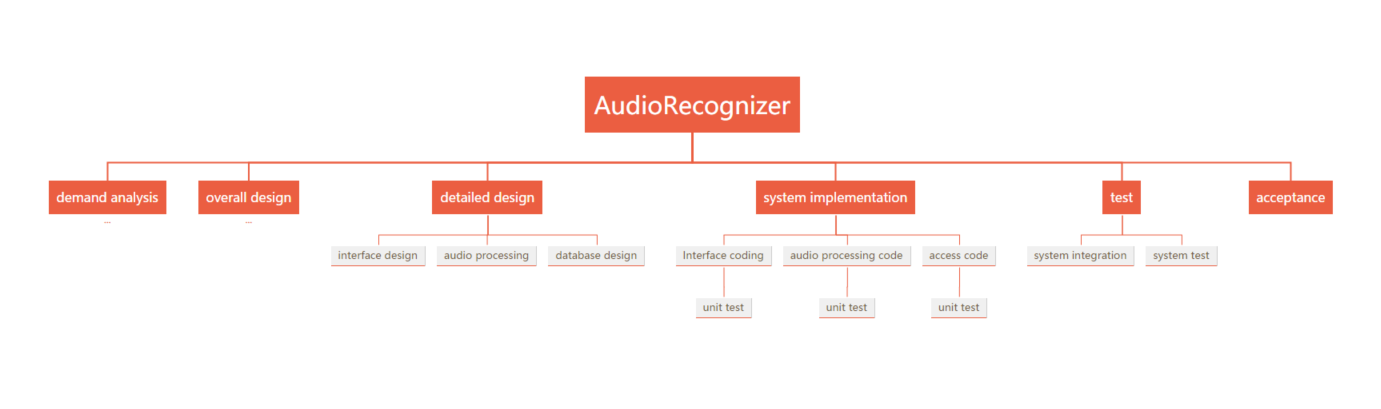
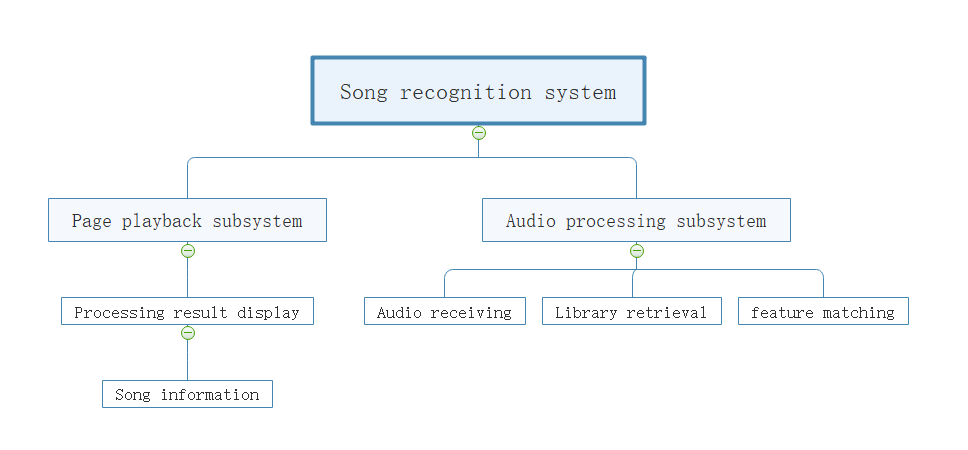


Diagram 2-3 Work breakdown structure diagram

## 2.4 System structure drawing



Digram 2-4 system structure drawing

## 2.5 Detailed module design

1. Page module design: this page is divided into three parts: the song recognition button, the information display part of the retrieved song returned in the background, and the control button of the returned song playing.

2. Audio processing module: Fourier algorithm is used to convert audio time-domain information into frequency domain information, and audio fingerprint idea introduced by shazam is used to extract audio characteristic values;

3. Database module: the database processing module is divided into database storage design and access operation. The database adopts hash index storage method. The retrieval USES a series of methods such as matching fingerprint to find the music with the highest matching degree in the database and then returns the information such as ID.

# 3. Job instructions (SOW)

## 3.1 scope of project

(1) project development module

|  |  |
| --- | --- |
| **order** | **module** |
| 1 | interface presentation |
| 2 | receiving voice |
| 3 | audio processing |
| 4 | data store |
| 5 | library retrieval |
| 6 | music play |

(2) Major deliverables

|  |
| --- |
| **work products** |
| Requirements specification |
| Detailed design specification |
| Interface specification document |
| operation manual |
| Project change requirements documentation |
| Test documentation |

## 3.2 organizational structure

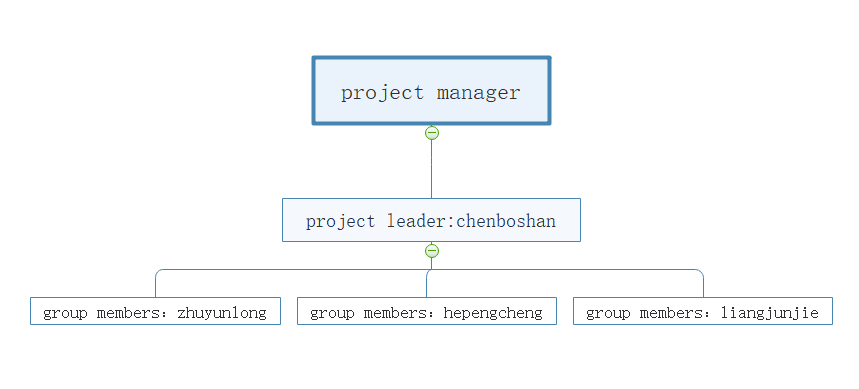


Table 3-2 [organization](javascript:;) [chart](javascript:;)

## 3.3 Personnel allocation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order | [job](javascript:;) [title](javascript:;) | specific personnel | Start time | End time | Average daily workload（Working hours） |
| 1 | Demand analysis | The whole team | 2019.2.26 | 2019.3.5 | 3 |
| 2 | Overall design | The whole team | 2019.3.6 | 2019.3.12 | 4 |
| 3 | Detailed interface design | 朱云龙，何鹏程 | 2019.3.13 | 2019.3.24 | 2 |
| 4 | Detailed database design | 陈柏山 | 2019.3.13 | 2019.3.24 | 2 |
| 5 | Audio processing design | 梁骏杰 | 2019.3.13 | 2019.3.24 | 4 |
| 6 | Interface code | 朱云龙，何鹏程 | 2019.3.25 | 2019.4.30 | 4 |
| 7 | Database design  coding | 陈柏山 | 2019.3.25 | 2019.4.30 | 2 |
| 8 | Audio processing coding | 梁骏杰 | 2019.3.25 | 2019.4.26 | 2 |
| 9 | Interface unit  testing | 朱云龙，何鹏程 | 2019.4.28 | 2019.4.30 | 3 |
| 10 | Database unit  test | 陈柏山 | 2019.4.28 | 2019.4.30 | 2 |
| 11 | Audio processing unit test | 梁骏杰 | 2019.4.28 | 2019.4.30 | 4 |
| 12 | Integration testing | The whole team | 2019.5.1 | 2019.5.4 | 2 |
| system test | The whole team | 2019.5.4 | 2019.5.8 | 2 |

Table 3-2 Personnel allocation

## 3.4 Project schedule

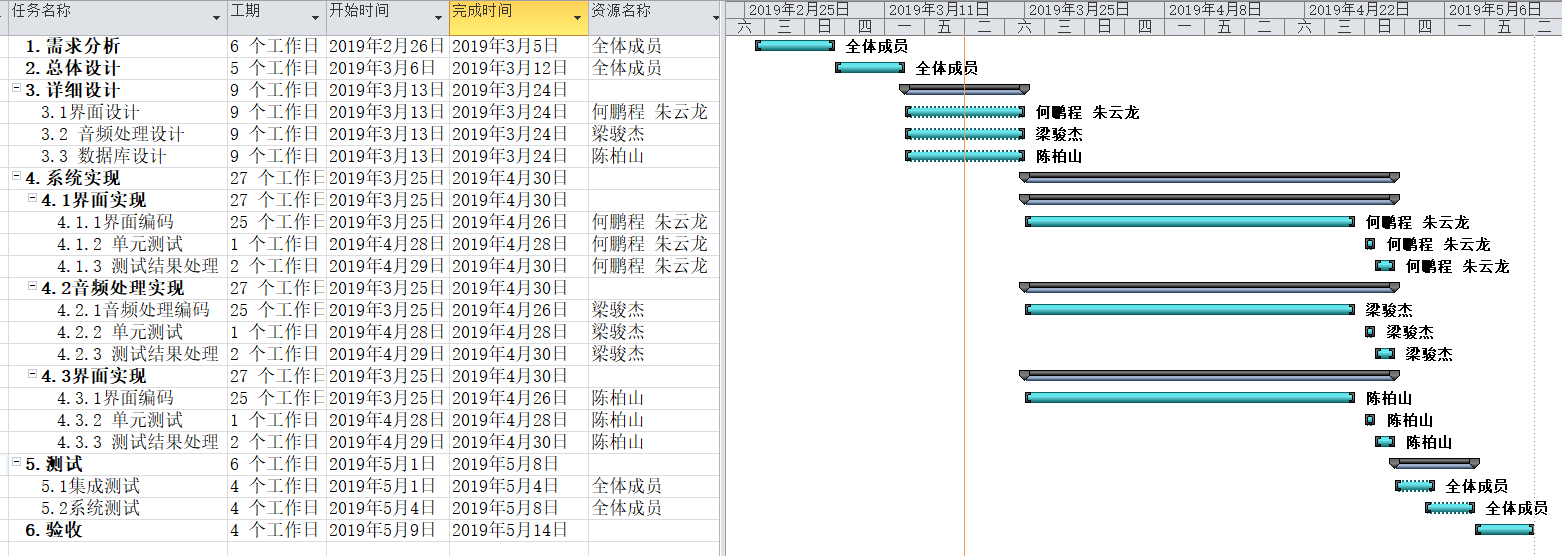


Diagram 3-4 project schedule

## 3.5 Collaboration and communication

Project communication management includes the processes required to ensure timely and reasonable generation, collection, distribution, storage, extraction and final utilization of project information. The objective is to ensure that all project stakeholders (including the project team, stakeholders, customers, and sponsors) receive and respond to timely information. Project communication runs through almost every link of the project. Effective communication management contributes to the success of the project. All stakeholders should understand the impact of communication on the project.

### 3.5.1 Internal collaboration

In order to ensure the smooth progress of the project development process and effective communication of information, the following communication plan is required:

1. Members of the project team shall have oral communication from 18:00 to 19:00 every day;

2. Submit weekly report before 14:00 every Friday, and make format template;

3. Held weekly project meetings from 16:00 to 17:00 every Friday, and issued meeting minutes to relevant project personnel after the meeting, which explained the progress and existing problems of the project;

Timely submit the problem report, the problem can be submitted through the network, the project leader will timely access to problem information.

### 3.5.2 Project communication skills

1. Use correct expressions. Communication must be purposeful. Before exchanging information, senders should consider their intentions and be concise. Express yourself in plain words.

2.Improve your listening skills. Communication is not just talking, but talking and listening. Listening is not only an important way for us to get first-hand information about others and get to know them correctly, but also the best way for us to show respect to others.

3.Avoid endless arguments. The result of this endless debate is inconclusiveness, which not only leaves the problem unresolved, but also delays its resolution.

4.Keep communication channels open. Communication is of course important, but without smooth communication channels, the organization will inevitably show spontaneous unorganized state, and it will be impossible to obtain the real information needed, and the operation efficiency of the entire organization will decline.

5.Use effective communication tools. Within the project organization, the relevant mature project management software, E-mail system, office automation system and other tools are usually used to support the generation, transmission and storage of various information of the project. The use of these tools greatly improves the efficiency of communication, pulls in the distance between the two sides, and reduces unnecessary interviews and meetings.

6. Grasp the principle of communication. One is the difference between internal and external communication. That is to ask the team as a whole to external consensus, a team to use a voice; Second, informal communication contributes to harmonious relations. Third, the other party can accept the communication style; Fourth, the principle of communication upgrade, namely the first step, and the other side of communication; The second step is to communicate with the superior of the other party. Step 3: communicate with your superiors. Step four: communicate with your superior and your counterpart's superior. Fifth, remove communication barriers.

# 4 Related knowledge Learning

* **Algorithm**

The algorithm to learn: the Fourier algorithm

Idea used: audio fingerprint feature matching

This project needs to learn the Fourier algorithm (FFT) to convert audio time-domain information into frequency domain information, and the audio fingerprint idea introduced by Shazam. This is a knowledge field that we have never been exposed to and understood before, so our team members decided to spend a week to learn and master FFT and audio fingerprint before the project started coding.

# 5.Project required interfaces

1. Recording interface: click the front-end song recognition button to trigger the recording event, and call this interface to limit the sampling size, frequency channel number and other parameters of recording and record;

2. I/O interface: call this interface to receive and read the audio stream;

3. Fourier transform interface: after clicking the song recognition button, the background calls this interface to transform the Fourier transform of the transmitted audio and extract the frequency domain information;

4. Fingerprint extraction interface: after receiving the frequency domain information, this interface is called to extract the audio fingerprint characteristic values;

5. Database operation interface: call this interface to access audio information;

6. Player interface: when the front-end song play button is clicked, the play music event is triggered, and this interface is called to play the returned music.

# 6.[Test](javascript:;) Plan

## 6.1Test schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [type](javascript:;) [of](javascript:;) [test](javascript:;) | [test](javascript:;) [content](javascript:;) | submission | Test time | head |
| Unit testing | Whether the front-end page UI effect can be achieved, and whether the recording, playing and information display functions can achieve the desired effect | Test results, bugs, solutions | 2019.4.28 | 朱云龙、何鹏程 |
| Whether the audio processing module decomposes the audio into frequency domain signals as expected and extracts fingerprint information with high recognition degree | 2019.4.28 | 梁骏杰 |
| Whether the database design module can establish an efficient hash access method and realize the storage and fast reading of fingerprint information | 2019.4.28 | 陈柏山 |
| Integration testing | Whether the audio recorded in the front end can be correctly transmitted to the background for processing, and whether the processed fingerprint information can be retrieved in the database and accurately matched and returned to the front end for playing | 2019.5.4 | [all](javascript:;) [members](javascript:;) |
| The system test | Package the system and imitate the user's operation process to see if the desired effects can be achieved correctly | 2019.5. 8 | [all](javascript:;) [members](javascript:;) |

Table 6-1

## 6.2 Writing purpose

* + The purpose of writing this document is to describe in detail the testing process of "music recognition system for listening to songs", and this test plan is to make sure whether the music recognition module is consistent with the function described in the system specification. By testing whether each functional module meets the requirements of users, storing errors and possible risks in the process, the feasible plan is made and the risks that may occur in the plan are effectively prevented to ensure that the test activities are carried out smoothly within the planned scope. The main purpose is to develop all kinds of necessary guidelines and specifications for the test work to be carried out, and to reasonably organize and manage the test work on the basis of the agreement of relevant parties.

## 6.3 reference

* + Detailed design and Project plan of the Project.
  + Li lianye, wang hua, li shuchun. Software testing and testing technology [M]. Beijing: tsinghua university press,2009:296-344.
  + Wang fasheng et al. Practical software testing course [M]. Beijing: tsinghua university press,2011:262-269.
  + IEEE 829-1998 software test documentation

## 6.4 Test risk assessment

6.4.1Risk:

* + resources: Does our team have sufficient programming development knowledge and capability for automated testing? If not, do we have the foundation to quickly master the technology? Are you going to build a good automation team?
  + Cost: Problems that plague the team: people, time, resources.
  + 100%automation? In the areas of performance testing, regression testing, and load stress testing, there are opportunities to achieve "100%."
  + Tool selection
  + Stability: Is the application stable enough to support automated testing?

## 6.4.2 emergency measures

* + (1) delete function (in special cases).
  + (2) some risks are unavoidable and we try to reduce them. For example, "no defects found in the program" is always present. We need to reduce risk by increasing test case coverage (for example, 99.9%).
  + (3) since the project features are relatively simple, you can choose a lightweight testing tool, which is easy to learn how to use.
  + (4) establish document standards and a mechanism for timely filing of documents.
  + (5) mutual review of all work and timely discovery of problems, including the exchange of different testers on different test modules.
  + (7) daily tracking of all processes, timely finding signs of risks and avoiding risks.

## 6.5 Unit Test Plan

### 6.5.1 purpose

* + Unit testing is the most important tool for code correctness verification and also the most important link in system testing. It is also the only test method that requires writing code to test. In standard development, unit test code is just as important as actual program code. Each unit test is used to test whether the data of the corresponding unit is correct. Typically, a unit test is used to determine the behavior of a particular function under a particular condition (or scenario).

### 6.5.2 Test content

* + The purpose of this unit test is to verify whether the newly added or modified modules meet the requirements of correct input and output, to ensure that the temporarily stored data is completely correct and logically correct in the process of program execution, so as to conduct static analysis and dynamic analysis.

### 6.5.3 Unit testing strategy

* + The unit test strategy adopts static analysis and dynamic analysis, which are specifically applied as follows:
  + Static analysis is a test method that does not actually run the software under test but finds possible errors in the software under test by means of test tools or manual inspection .This method is applied to key modules and adopts the code walkthrough technique in static analysis. Dynamic analysis refers to the process of actually running the oftware under test, input the corresponding test data, and check whether the actual output results are consistent with the expected results.

### 6.5.4 Test tools

|  |  |  |  |
| --- | --- | --- | --- |
| **工具名称** | **版本** | **生产厂商** | **说明** |
| Testbed | 9.4.0 | LDRA | * Basic static analysis and dynamic coverage analysis |
| TBvision | 9.4.0 | LDRA | Static software analysis |
| Selenium |  |  | * [business](javascript:;) [logic](javascript:;) [layer](javascript:;) |
| Tbrun | 9.4.0 | LDRA | Dynamic analysis and testing |
| QUnit | 2.9.2 | jQuery | JavaScript Unit test framework |
| Junit |  |  |  |
| 人工检查 | N/A | N/A | * It is mainly applied to the code rule checking of FPGA in static analysis |

### 6.5.5 Test module

1. Interface display module

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * Test categories | | * functional test | | Module to be tested | Interface module | |
| * Output results after test execution | | | | | | |
| * Number | * Functions to be tested | Expected output | Test output | test mode | | Defect severity |
| * 1 | * index() | The program launches with an HTML pa | * Start the program, out of the home index.html | * Pass | |  |
| * 2 | * Recorder() | * Can realize web page call system microphone | * Call the microphone to record | * Pass | |  |
| * 3 | * Getfullwavdata() | * Convert recorded audio in PCM format to wav format | * Successfully converted to wav format | * Pass | |  |
| * 4 | getPureWavData() | * Convert wav audio to blob format | * Successful transformation | * Pass | |  |
| * 5 | Trans() | * The blob format is transmitted to the backend for identification | * The resulting blob format was transferred successfully | * Pass | |  |

1. Recognition module

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * Test categories | | * functional test | | Module to be tested | Interface module | |
| * Output results after test execution | | | | | | |
| * Number | * Functions to be tested | * Expected output | * Test output | * test mode | | Defect severity |
| * 1 | * makeFFT() | * Apply FFT transform to the audio | * Block FFT conversion of a two-dimensional array | * Pass | |  |
| * 2 | * determineFingerprints() | * Fingerprint is extracted from the information of tone frequency domain, and matched with the information in the database, and the matching result is output | * The feature fingerprint was successfully obtained and the song information was returned | * Pass | |  |
| * 3 | * addFingerprint() | * Add song fingerprint to fingerprint library | Successfully adding | * Fail | | * 一般 |
| * 4 | * getIndex() | * An index that outputs audio of a frequency band | Successful outcome | * Pass | |  |
| * 5 | * shiBieWenJianLiu() | * Identify file streams and output string song names | * String song name | * Pass | |  |
| * 6 | * getFormat() | * Gets the audio data for the specified parameters | * Audio message successfully obtained | * Pass | |  |

1. **Playback module**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test categories | | functional test | | Module to be tested | Interface module | |
| * Output results after test execution | | | | | | |
| Number | Functions to be tested | Expected output | Test output | test mode | | Functions to be tested |
| 1 | StartRecording() | * Start recording function to receive audio message | * Successful recording | Pass | |  |
| 2 | stopRecording() | * Stop recording function, send and receive audio information to the background | * Audio message sent and received successfully | Fail | | [**tiny**](javascript:;) |
| 3 | PlayRecording() | * The identified audio message will be played successfully | * Play audio successful | Pass | |  |

### 6.5.6 Test activity schedule

|  |  |  |
| --- | --- | --- |
| **测试活动** | **测试时间** | **测试人员** |
| * Develop a software unit test plan | 2019.05.07-2019.05.09 | 何鹏程、梁骏杰 |
| * Design software unit test cases | * 2019.05.07-2019.05.09 | * 朱云龙、何鹏程 |
| * Review the software unit test plan | * 2019.05.09-2019.05.11 | * 小组全体成员 |
| * Execute software unit test cases | * 2019.05.09-2019.05.11 | * 陈伯山，梁骏杰 |
| * Verify the software unit test report | * 2019.05.12 | * 小组全体成员 |

### 6.5.7 Test case

|  |  |
| --- | --- |
| tester:何鹏程，朱云龙 | |
| * Number | * [test](javascript:;) [procedures](javascript:;) |
| * 001 | |
| * Test description: display of buttons at all levels. | |
| Function point test | * 1 Function button order, icon display |
| 2 Function button function |
| 1.1 | * Open the music recognition application, enter the music recognition interface, and click the function options. |
| 1.2 | * Enter the list of listening history records. |
| 1.3 | * In the standby interface, repeatedly press the start recognition button. |

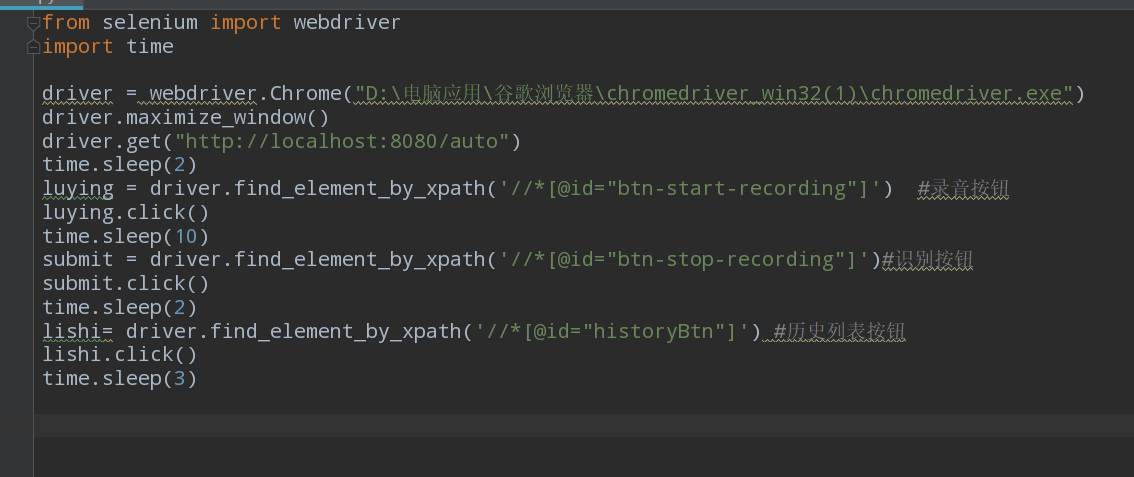
|  |  |
| --- | --- |
| tester:陈伯山，梁骏杰 | |
| * Number | * [test](javascript:;) [procedures](javascript:;) |
| * 002 | |
| * Test description: display of buttons at all levels. | |
| Function point test | * 1 Recording function verification. Start, pause and stop the recording. |
| * 2Enter the list of historical recordings for simple operation. |
| 2.1 | * Click the start recognition button to enter the recording interface. |
| 2.2 | Click the pause button. |
| 2.3 | Click the continue button. |

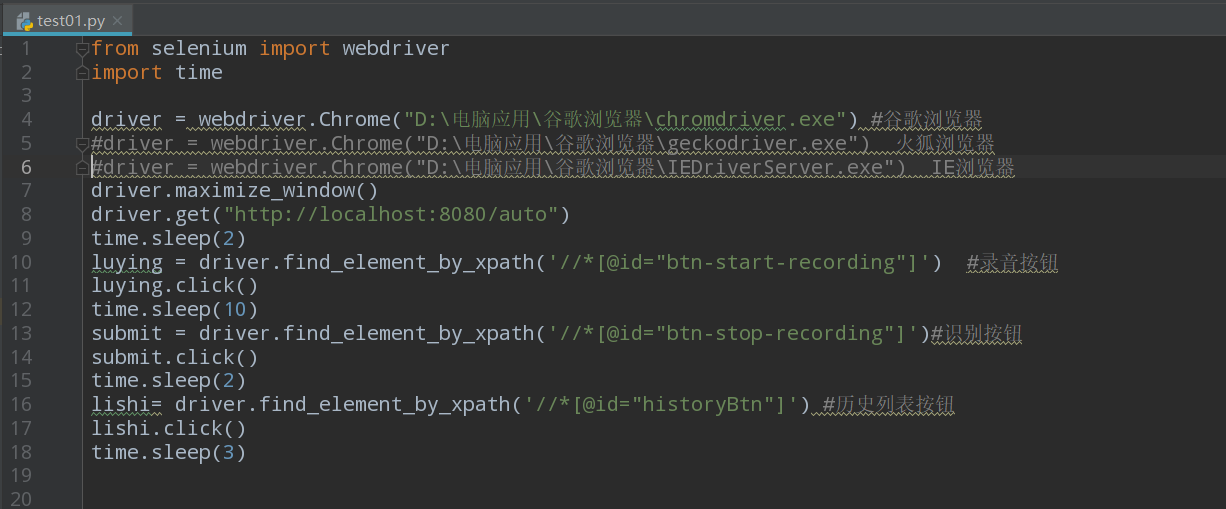
### 6.5.8 Test script code

* **//Junit test**

**package** com.it;  
  
**import** org.junit.Before;  
**import** org.junit.Test;  
**import** javax.sound.sampled.LineUnavailableException;  
**import static** org.junit.Assert.*assertEquals*;  
  
**public class** AudioRecognizerTest {  
  
 AudioRecognizer **testAR**;  
  
 @Before  
 **public void** setUp() **throws** Exception {  
 **testAR**=**new** AudioRecognizer();  
 }  
  
 @Test  
 **public void** getIndex() {  
 *assertEquals*(0,**testAR**.*getIndex*(38));  
 *assertEquals*(1,**testAR**.*getIndex*(50));  
 *assertEquals*(2,**testAR**.*getIndex*(100));  
 *assertEquals*(3,**testAR**.*getIndex*(150));  
 *assertEquals*(4,**testAR**.*getIndex*(200));  
 }  
  
 @Test  
 **public void** addSongInfo(){  
 **testAR**.addSongInfo();  
 }  
  
 @Test  
 **public void** listenSound() **throws** LineUnavailableException {  
 **testAR**.listenSound();  
 }  
  
}

* **package** com.it;  
    
  **import** org.junit.Before;  
  **import** org.junit.Test;  
  **import** javax.sound.sampled.LineUnavailableException;  
  **import** javax.sound.sampled.UnsupportedAudioFileException;  
  **import** java.io.IOException;  
    
  **public class** WenJianToFingerprintTest {  
    
   WenJianToFingerprint **wjtfp**;  
    
   @Before  
   **public void** setUp(){  
   **wjtfp** = **new** WenJianToFingerprint();  
   }  
    
   @Test  
   **public void** shiBieWenJianLiu() **throws** UnsupportedAudioFileException, IOException, LineUnavailableException {  
   **wjtfp**.shiBieWenJianLiu();  
   }  
  }
* Automated test





### 6.5.9 problem summary

All functions have been implemented;Basic stability;Performance up to the requirements;

But there are still the following problems;

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Problems | Cause analysis, | solution |
| 1 | After running the system, song recognition can only be performed once. | When uploading mp3 files in the background, the original intention was to create only one file to save the recording, which could be overwritten later.But while the server is running, the recording file is occupied, causing the following program to fail to run | Name the file with an argument. |
| 2 | When the background gets the return value of the song name, the foreground value is often unable to get the value itself due to the string concatenation within quotation marks. |  | It needs to be checked carefully. |

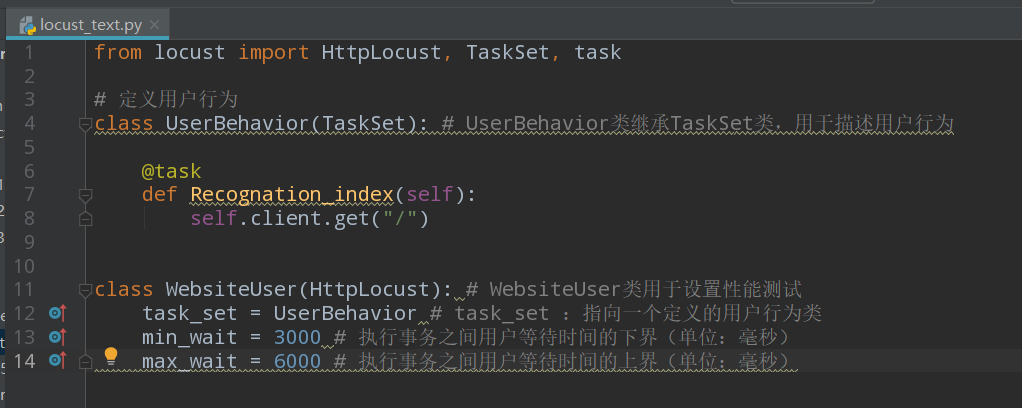
## 6.6 Performance Test

### 6.6.1 Test tool

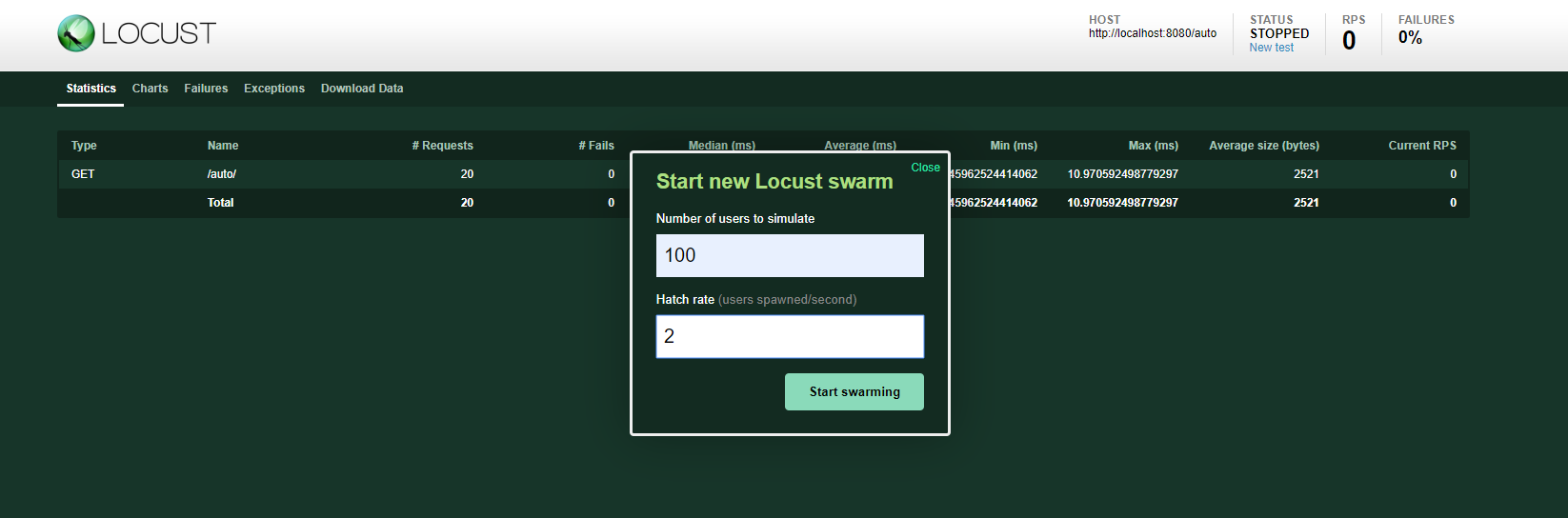
Locust was used as the performance testing tool for this performance test. Locust's concurrency mechanism abandons processes and threads in favor of a gevent mechanism. Coroutines avoid system - level resource scheduling and can greatly improve the concurrency of a single machine. Locust is web-oriented and can be used to test almost any system. Just write a client that you can test. Locust has a simple HTML + JS user interface that displays relevant test details in real time .Because it's web-based, it's cross-platform and easy to extend.

### 6.6.2 Test process

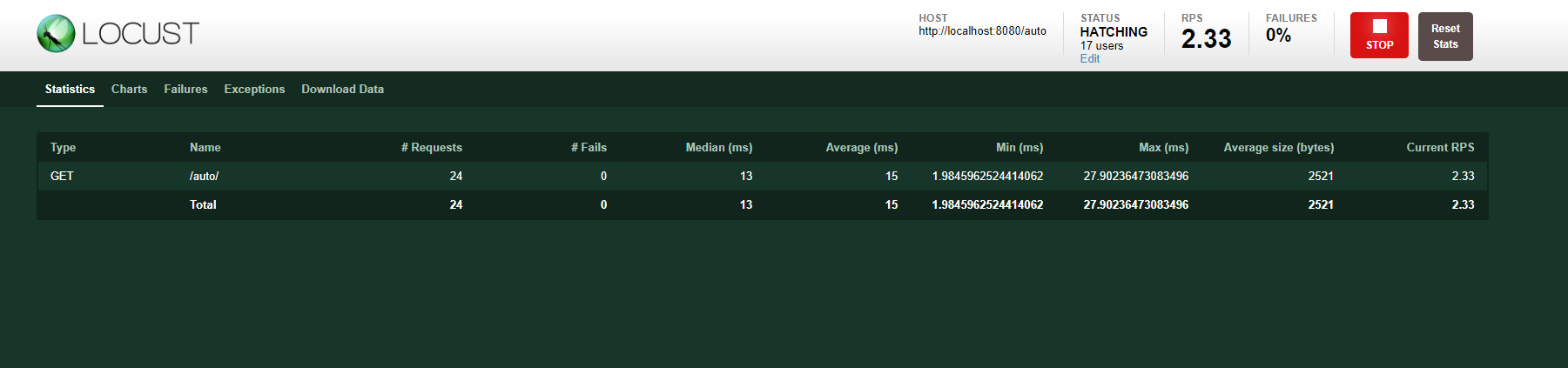
1. Test script



1. Set test parameters



1. Test status



# 7.Milestones

The system adopts modular development. There are 8 milestones in the development process:

Table 7-1

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone name | Date of completion | Deliverables |  |
| M1 Demand analysis milestone | 2019.3.5 | Demand specification | [all](javascript:;) [members](javascript:;) |
| M2 Detailed design milestone | 2019.3.19 | Detailed design specification | [all](javascript:;) [members](javascript:;) |
| Coding milestone | 2019.4.26 | Code status, code walk record | Responsible person of each module |
| Interface test milestone | 2019.4.30 | Test results, bugs, solutions | 朱云龙、何鹏程 |
| Audio processing test milestone | 2019.4.30 | 梁骏杰 |
| Database test milestone | 2019.4.30 | 陈柏山 |
| Integration test milestone | 2019.5.04 | [all](javascript:;) [members](javascript:;) |
| System test milestone | 2019.5.08 | [all](javascript:;) [members](javascript:;) |
| End of project milestone | 2019.5.14 | Requirements specification;  Detailed design specification;  Interface specification document;  operation manual;  Project change requirements documentation;  Test documentation | [all](javascript:;) [members](javascript:;) |

# 8.The risk plan

Risk refers to the events that may occur during the course of the project, which will have a significant impact on the completion of the project according to the expected time, resources and budget. The goal of risk analysis is to identify these events, try to avoid them and formulate measures to deal with them once they occur.

The following risk list is a risk identification, risk assessment, risk response, and finally the project risk plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project risk management | | | | | |
| Judgment criteria of risk occurrence probability: | | | | | |
| High risk: >60% chance of risk: 30-60% chance of risk low risk: <30% chance of defense | | | | | |
| Order | risk description | probability of occurrence | incidence | risk grade | Risk response plan |
| 1 | Due to time constraints, it could not be finished on time | [low](javascript:;) | maximum | high | Full consideration of various potential factors, appropriate margin; Detailed task breakdown for easy assessment; In case of necessary delay, the group leader shall communicate with the relevant person in charge in time and apply for extension time. |
| 2 | The system does not have enough test time | high | Big | high | Continuous monitoring, project progress control with the progress of the project, to ensure that there is enough time for each link. |
| 3 | There are some problems in developing software structure system, which make the completed software products fail to achieve the predetermined goal of the project | High | High | [middle](javascript:;) | Choose legal software development |
| 4 | Developers have no project development experience and lack of technical expertise | High | High | High | Carry on the related knowledge study, consult the related technical domain expert, solve the problem in the actual combat. |
| 5 | In the process of software project development and implementation, the necessary  management tools, development tools and testing tools are not in place in time | Low | Big | High | Identify the source of the tools or possible alternatives at the start of the project, and track and implement the tools in place before they are needed. |
| 6 | Team members are unable to participate in the design due to accidents | Middle | Middle | High | Make a personnel replenishment plan in advance, and invite foreign aid at critical moments. Or do a character breakdown plan for unexpected people. |
| 7 | Overestimate their own ability，not estimated the difficulty of the  Project | High | Middle | High | Ask experienced project personal to check，for difficult to overcome the technical help of relevant experts，can not directly achieve the idea can be replaced by the analogy method. |

Table 8-1

# 9 detailed design

## 9.1 writing purpose

Writing the detailed design specification is an essential part of the software development process. The purpose is to enable the developer to complete the design work of each module specified in the outline design on the basis of completing the outline design specification.

Based on the function and performance of Audio Recognizer system, the paper expounds the realization details of the system's function module.

Expected readers of this document include:

* Design developer
* Project management personnel
* Software Tester
* User

## 9.2 reference

[1] software project management, China railway publishing house

[2] HTML5+CSS3+JavaScript from entry to mastery (standard edition). China water resources and hydropower press

[3] practical HTML5, people's posts and telecommunications press

[4] Fourier transform algorithm and application, China machine press

## 9.3 Terminology definition

|  |  |  |
| --- | --- | --- |
| **Number** | **Term** | Descriptive definition |
| 1 | MySql | * A database relational system (DBMS) used by a system server. |
| 2 | B/S | Browser/server structure, it is with the rise of Internet technology, the C/S structure of a change or improvement of the structure. |
| 3 | SQL | A language for accessing query databases |
| 4 | UML | Unified modeling language (uml) is a set of standard modeling language to design software blueprints. |
| 5 | HTML5 | HTML 5 technology officially formed in 2014, it is stronger than any previous HTML version, a more interactive, will include multimedia, and also provides an application program interface. It can handle text, audio and video, images, etc., have a good human-computer interaction. HTML can also cross-platform, all can use in the different system platforms. You can also across devices, can be different to automatically adjust the size of the screen web browser. |
| 6 | shazam | Is an audio fingerprinting system already in use that allows a mobile device to identify a playing song in one of several ways and return information about the song being played to the user.The fingerprint of the algorithm is based on the matching of peak energy points of the spectrogram. |

## 9.4 Key technologies and solutions

This system adopts B/S structure, development tool is Intellij IDEA, background database is MySql, software requirements: Windows 10 operating system, client Chrome, hardware requirements: Inter i5 or higher, memory requirements of 2G or higher.

This subject needs to learn the Fourier algorithm (FFT) to convert audio time-domain information into frequency domain information, and the audio fingerprint idea introduced by Shazam.This was an area of knowledge we had never been exposed to and understood, so our team members decided to spend a week learning and mastering FFT and audio fingerprinting before the project started coding.

## 9.5 profile design

### 9.5.1 Demand profile

With the rapid keyword retrieval technology based on massive audio content becoming more and more mature, it plays an important role in people's spiritual life. When we overhear a song or a piece of music, but we need to know the name of the song to record, but there is no way to start. In order to increase the user experience, this product is a simple and easy to use music recognition application, the user can use this function to identify the current environment is playing the song information, give people in the break time to bring new surprises and experience. It's also a way to find a good music, after all, a good song is worth recommending to more listeners. With it, you can explore more new music with your friends and share it around the world.

The system is required to be effective, fast, reliable and error-free to complete the song recognition operation. The system interface is required to be simple and easy to operate.

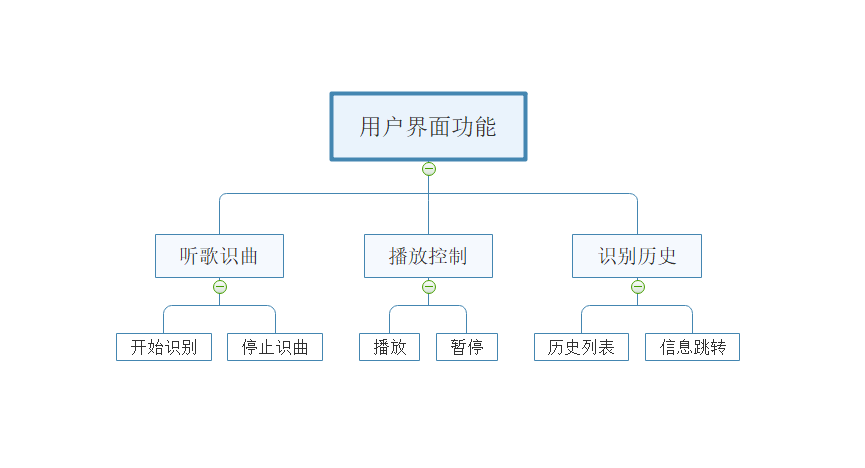
### 9.5.2 System development environment

Configuration required for product development and maintenance

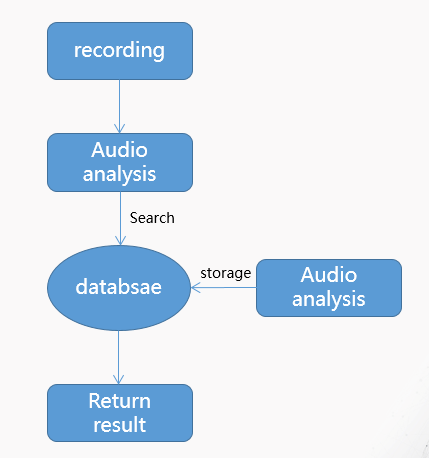
：

|  |  |  |
| --- | --- | --- |
| [classify](javascript:;) | [name](javascript:;) | Version |
| [operating](javascript:;) [system](javascript:;) | Windows | Windows 10 |
| The database | Mysql | 5.7 |
| Programming language front end | H5 |  |
| The programming language back end | Java | 10.0.2 |
| Development environment IDE | Intellij IDEA | 2018.03 |

### 9.5.3 Functional chart



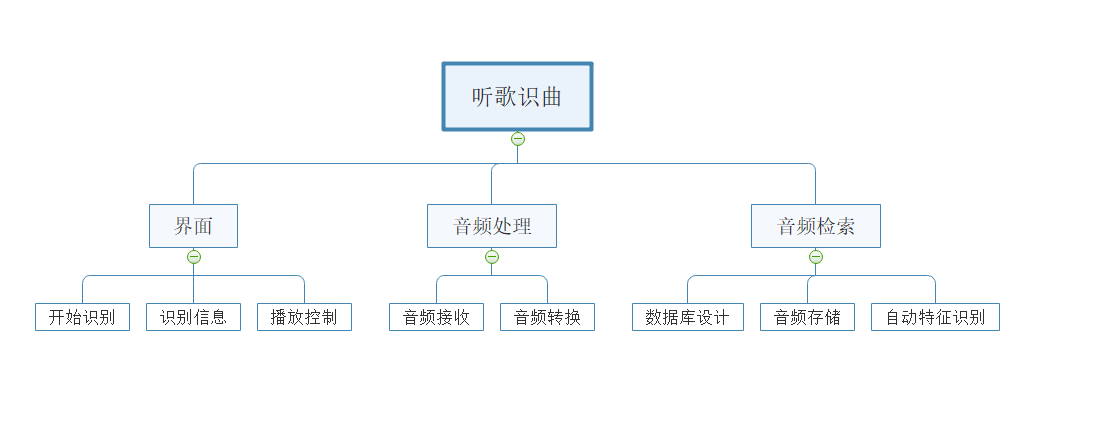
### 9.5.4 System flow chart



First, the user clicks the button of "listen to music", the system calls the microphone of the device to record, and transmits the audio information to the background server.The background receives the audio information from the front end, performs Fourier transform on it, extracts audio fingerprint and so on, and then compares and retrieves the processed fingerprint information in the database, and returns the retrieval result (song information) after the retrieval is completed. Finally, the user can click the play button in the song information display section to play to see whether the displayed song is the song played in the current environment.

## 9.6 Detailed module design

### 9.6.1 System module design



Module description:

1. Page module design: this page is divided into three parts: the song recognition button, the information display part of the song retrieval returned by the background, and the control button of returning the song playing.

2. Audio processing module: the audio time-domain information is converted into frequency-domain information by Fourier algorithm, and the audio fingerprint idea introduced by shazam is used to extract audio characteristic values;

3. Database module: the database processing module is divided into database storage design and access operation.The database USES hash index storage method.The retrieval adopts a series of methods such as matching fingerprint to find the music with the highest matching degree in the database, and then returns the information such as ID.

### 9.6.2 Interface design

1. Recording interface: click the front-end song recognition button to trigger the recording event, and call this interface to limit the sampling size, frequency channel number and other parameters of recording and recording;

2.I/O interface: call this interface to receive and read audio stream;

3. Fourier transform interface: after clicking the song recognition button, the background calls this interface to carry out Fourier transform on the transmitted audio and extract the frequency domain information;

4. Fingerprint extraction interface: after receiving the frequency domain information, the interface is called to extract the audio fingerprint characteristic values;

5. Database operation interface: call this interface to access audio information;

6. Player interface: click the front-end song play button to trigger the play music event and call the interface to play the returned music.

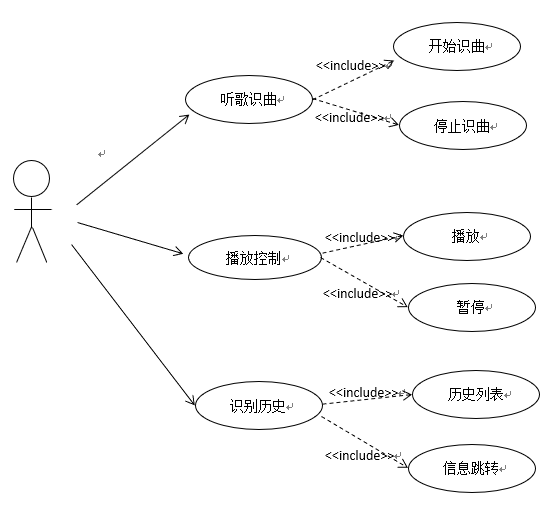
UI design

The diagram is simply an operational prototype interface, which is then evaluated to verify that the user's requirements are met, modified as necessary, and then modified again until the user is satisfied.

Prototype of song recognition interface:



## 9.7 UML

System use case diagram:

**Use case description:**

(1) use case name: start music recognition

Participant: user

Objective: to enable the user to record using the microphone of the current device and identify the songs playing in the current environment

Precondition: the system is running

Basic event flow: 1. The user clicks the "start music recognition" button on the interface

2. The system will recognize the songs played in the current environment

(2) use case name: stop music recognition

Participant: user

Goal: to enable the user to stop the system from identifying songs

Precondition: the system is recognizing the song played in the current environment (no result has been returned)

Basic event flow: 1. The user clicks the "stop music recognition" button

2. The system will stop identifying songs

(3) use case name: play

Participant: user

Goal: enable the user to play the song returned by background recognition

Precondition: the system recognizes and returns the song information recorded by the microphone

Basic event flow: 1. The user clicks the "play" button

Play the identified song

(4) use case name: pause

Participant: user

Goal: to enable users to pause playing songs

Precondition: the identified song is playing

Basic event flow: 1. The user clicks the "pause" button

2. Pause the currently playing song

(5) use case name: identify history

Participant: user

Goal: enables the user to view the history that the system has identified on this device

Precondition: the system is in a running state

Basic event flow: 1. The user clicks the "play history" button

2. The system will display the list of identified historical songs

(6) use case name: song information display

Participant: user

Goal: enables the user to view specific information about the song in the recognition history form

Precondition: identify songs in the history form

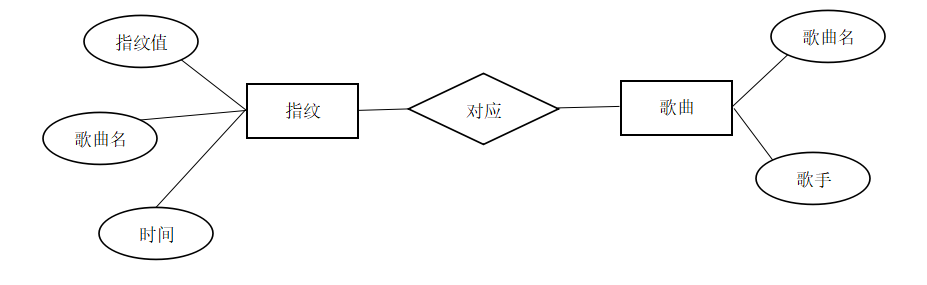
Basic event flow: 1. The user clicks the list song

2. The system will jump to the song information display page

## 9.8 Database module design

Database conceptual structure design: e-r diagram

:

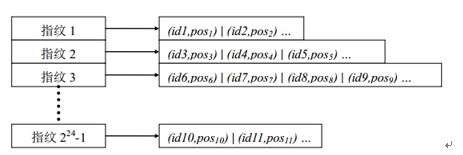


* Relationship mode corresponding to e-r diagram:

Song fingerprint (fingerprint value, song name, time)

Song (song title, singer)

* Database physical structure design: design the physical structure of the database, according to the logical structure of the database to choose the DBMS, and design and implement the data base storage structure, access, etc.
* After the song generates the fingerprint, the songs to be retrieved can be returned by searching the fingerprint database.The design of this system puts the generated fingerprint into the hash table, and the fingerprint is used to generate the hash key. Each table item is the music data corresponding to the same fingerprint.Music data includes the id of music and the time when the fingerprint appears in music。



## 9.9 System error handing design

* The main error in this system is that the audio provided by the user is not correctly matched. In order to solve this problem, aiming at the possible causes of this error, the fundamental method is to improve the rigor of the code in the system design and coding stage, improve the selection method and number of fingerprints, in order to improve the system's audio recognition rate and noise resistance to the environment.

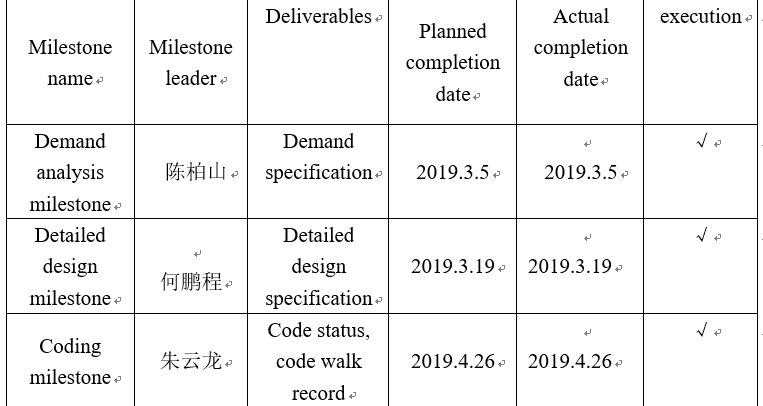
|  |  |  |
| --- | --- | --- |
| Error messages | Cause | [solution](javascript:;) |
| Program stop | Program interrupt | * Restart the application/clear the cache |
| Music doesn't play | * File corrupt or incorrect playback path | Place the music files in the specified folder |
| Recognition error/no response | * User record audio too short/noisy environment | Adjust admission time/choose a quiet recording place. |

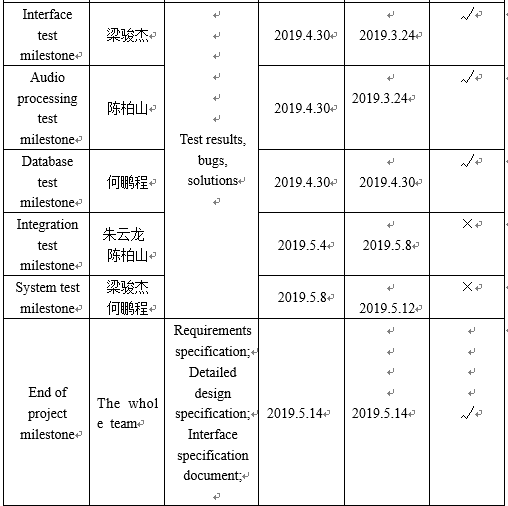
# 10 summarize

## 10.1 final product demo



## 10.2 Project execution





## 10.3 Lessons learnt

* Mentality problems: The change of people's mentality runs through the whole project and actively adjusts the state of the team, which is the key to the smooth progress of the project.
* Team collaboration problems: Internal sometimes due to the different roles in different stages of the project team or with phase division of responsibility between different role is not clear and cause mutual shuffle, the phenomenon of mutual shirk responsibility, sometimes between different roles or stage with different roles in different stages of the division of responsibility more clear, but each team member to complete her that part of the task, not willing to collaborate with others. These phenomena more or less cause the depletion of internal resources of the project team, thus affecting the progress of the project.
* Communication problems: Some important information in the project is not fully and effectively communicated. Insufficient communication with relevant personnel in making plans, feedback, situation reports, technical problems or achievements, etc., resulting in repeated work and even unnecessary losses; There is no clear communication medium to keep up with the latest information.
* Risk management issues: Failure to identify risks effectively and plan for risk response.
* Personnel quality problems: The quality of project members is uneven, and the reasonable evaluation of the level of members is the key to the smooth progress of the project, which also involves the division of labor.
* Project planning awareness: Always be vigilant and follow the project plan to the letter.
* Technical thinking problems: We should be flexible in dealing with all kinds of problems and difficulties that cannot be solved by technology. We should learn to change our thinking, jump out of the technological trap and see problems from other perspectives.

## 10.4 conclusion

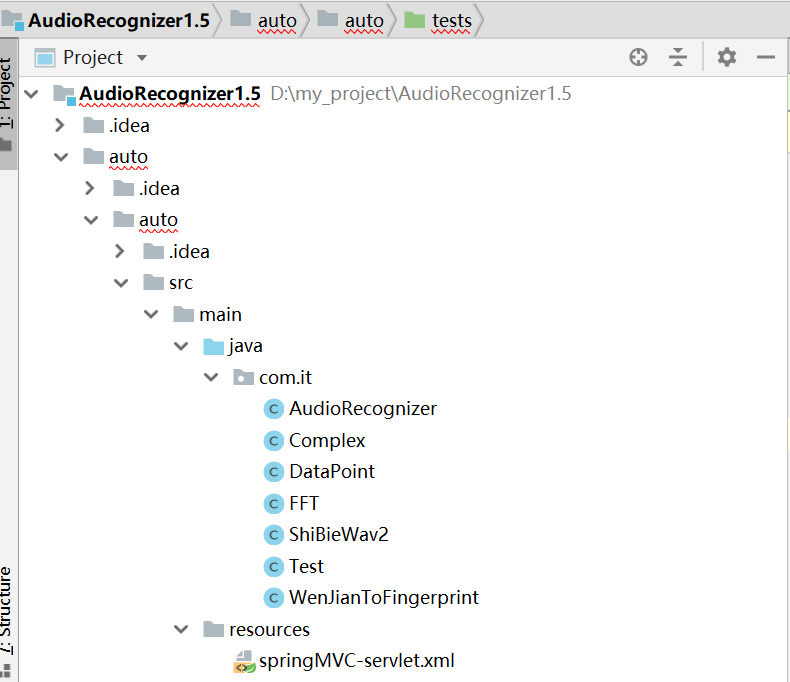
* Completing a software project is more difficult than you might think. Software projects require more than just writing code.
* It is impossible to do things alone, teamwork is the first element, no communication team is scattered.
* learning by doing.
* Practice is the of all instructors.
* Do not rely too much on the management of people, but to manage the process of good norms; Instead of following the project manager's instructions, project members follow process specifications.
* Management mentality.

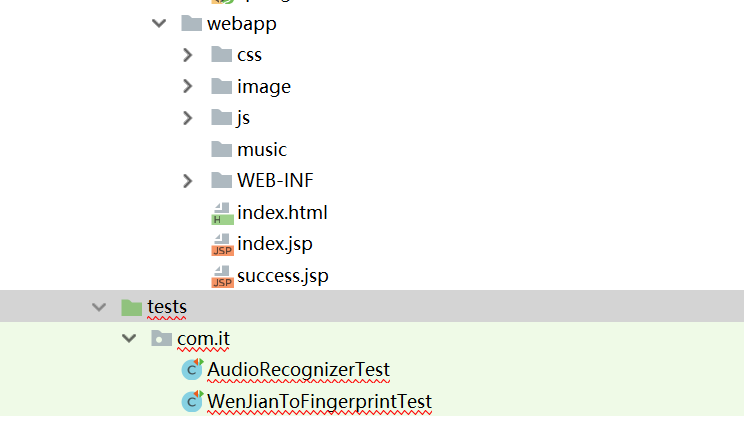
## 10.5 Future plan

1. Improve product function and standardize all project documents.
2. Use software project management tools, Zentao.
3. Optimization of product UI.
4. The algorithm is improved to improve the recognition speed and accuracy.
5. Refactoring project code.
6. Determine the product logo, development of mobile terminal, WeChat small program.
7. Focus on recognition, develop new areas of recognition, and add audio social features.

# 11 Appendix

## 11.1 Project code Structure





## 11.2 Code

### 11.2.1 The back-end code

1. AudioRecongnizer

**package** com.it;  
  
**import** javax.sound.sampled.\*;  
**import** java.io.\*;  
**import** java.sql.\*;  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
**import** java.util.List;  
**import** java.util.Map;  
  
*/\*  
 \*录音然后进行频域转换，再提取指纹构建哈希表  
 \*  
 \*/***public class** AudioRecognizer {  
 Boolean **running** = **false**;  
 **double highscores**[][];  
 **long points**[][];  
 Map<Long, List<DataPoint>> **hashMap**; *// Map<哈希值，对应的歌曲名和指纹出现时间的信息>* Map<Integer, Map<Integer, Integer>> **matchMap**; *// Map<SongId, Map<Offset,Count>> Map<歌曲名，Map<时间差，时间差计数>>  
  
 // 录音的参数* **public** AudioFormat getFormat() {  
 **float** sampleRate = 44100; *// 采样频率* **int** sampleSizeInBits = 8; *// 1,2 信道数（单声道为 1，立体声为 2，等等）* **int** channels = 1; *// 单声道 1,2 信道数（单声道为 1，立体声为 2，等等）* **boolean** signed = **true**;  
 **boolean** bigEndian = **true**;  
 **return new** AudioFormat(sampleRate, sampleSizeInBits, channels, signed, bigEndian); *// 构造具有线性 PCM 编码和给定参数的 AudioFormat* }  
  
 *// TargetDataLine是声音的输入(麦克风),而SourceDataLine是输出(音响,耳机)。* **public void** listenSound() **throws** LineUnavailableException {  
 AudioFormat format = getFormat();  
  
 *// lineClass - 该信息对象所描述的数据行的类  
 // format - 所需的格式* DataLine.Info info = **new** DataLine.Info(TargetDataLine.**class**, format);  
 **final** TargetDataLine line = (TargetDataLine) AudioSystem.getLine(info);  
  
 *// 启动麦克风？* line.open(format);  
 line.start();  
  
 *// 读取TargetDataLine中的数据* Thread listenThread = **new** Thread(**new** Runnable() {  
  
 @Override  
 **public void** run() {  
  
 *// 字节数组输出流在内存中创建一个字节数组缓冲区，所有发送到输出流的数据保存在该字节数组缓冲区中  
 // 字节数组输出流不需要目标源，因为是输出到内存中  
 // 创建一个32字节（默认大小）的缓冲区* ByteArrayOutputStream out = **new** ByteArrayOutputStream();  
 **running** = **true**;  
 **int** n = 0;  
 **byte**[] buffer = **new byte**[(**int**) 1024];  
 **try** {  
 System.out.println(**"正在聆听。。。"**);  
 **while** (**running**) {  
 n++;  
 **if** (n > 1000) {  
 **break**;  
 }  
 **int** count = line.read(buffer, 0, buffer.**length**);  
 **if** (count > 0) {  
 out.write(buffer, 0, count);  
 }  
 }  
 determineFingerprints(makeFFT(out));  
 out.close();  
 line.close();  
 } **catch** (IOException e) {  
 System.err.println(**"I/O problems: "** + e);  
 System.exit(-1);  
 }  
 }  
 });  
 listenThread.start();  
 }  
  
 *// 对音频进行FFT变换* **public** Complex[][] makeFFT(ByteArrayOutputStream out) {  
  
 *// toByteArray():创建一个新分配的字节数组。数组的大小和当前输出流的大小，内容是当前输出流的拷贝* **byte** audio[] = out.toByteArray();  
 **final int** totalSize = audio.**length**;  
 **int** chunkSize = 4096;*//以4096byte(4KB)为一个数据块  
  
 // 采样的块数* **int** amountPossible = totalSize / chunkSize;  
 Complex[][] results = **new** Complex[amountPossible][];  
  
 *// 代码的内层循环将采样数据放入一个复数数组中（虚部为0），外层循环遍历每一块数据，并进行FFT变换  
 // j是行，i是列* **for** (**int** j = 0; j < amountPossible; j++) {  
  
 *// 每块数据块大小的复数数组* Complex[] complex = **new** Complex[chunkSize];  
 **for** (**int** i = 0; i < chunkSize; i++) {  
  
 *// 将时域数据放入具有虚数的复数中  
 // 虚部为零* complex[i] = **new** Complex(audio[(j \* chunkSize) + i], 0);  
 }  
  
 *// 对块进行FFT转换 二维数组可以省略行？* results[j] = FFT.*fft*(complex);  
 }  
 **return** results;  
 }  
  
 **public static final int *UPPER\_LIMIT*** = 300;  
 **public static final int *LOWER\_LIMIT*** = 30;  
 **public static final int**[] ***RANGE*** = **new int**[]{40, 80, 120, 180, ***UPPER\_LIMIT*** + 1};  
  
 *// 找出频率所属范围* **public static int** getIndex(**int** freq) {  
 **int** i = 0;  
 **while** (***RANGE***[i] < freq)  
 i++;  
 **return** i;  
 }  
  
 *// 对音频频域信息提取指纹* **public** String determineFingerprints(Complex[][] results) {  
 String info = **null**;  
 System.out.println(**"正在匹配。。。"**);

……………………………………………省略………………………………………………………………………………..

1. Complex

**package** com.it;  
  
*/\*  
\* 定义复数，以及一些复数的加减乘除等操作  
\*  
\*/***public class** Complex {  
 **private final double re**; *// 实部* **private final double im**; *// 虚部  
  
 // 使用给定的实部和虚部创建一个新对象* **public** Complex(**double** real, **double** imag) {  
 **re** = real;  
 **im** = imag;  
 }  
  
 *// 返回调用Complex对象的字符串表示形式* **public** String toString() {  
 **if** (**im** == 0)  
 **return re** + **""**;  
 **if** (**re** == 0)  
 **return im** + **"i"**;  
 **if** (**im** < 0)  
 **return re** + **" - "** + (-**im**) + **"i"**;  
 **return re** + **" + "** + **im** + **"i"**;  
 }  
  
 *// 返回绝对值/模数/幅度和角度/相位/参数* **public double** abs() {  
 **return** Math.hypot(**re**, **im**);  
 } *// Math.sqrt(re\*re + im\*im)* **public double** phase() {  
 **return** Math.atan2(**im**, **re**);  
 } *// between -pi and pi  
  
 // return a new Complex object whose value is (this + b)* **public** Complex plus(Complex b) {  
 Complex a = **this**; *// invoking object* **double** real = a.**re** + b.**re**;  
 **double** imag = a.**im** + b.**im**;  
 **return new** Complex(real, imag);  
 }

……………………………………………省略………………………………………………………………………………..

1. DataPoint

**package** com.it;  
  
**public class** DataPoint {  
  
 **private int time**;  
 **private int songId**;  
  
 **public** DataPoint(**int** songId, **int** time) {  
 **this**.**songId** = songId;  
 **this**.**time** = time;  
 }  
  
 **public int** getTime() {  
 **return time**;  
 }  
  
 **public int** getSongId() {  
 **return songId**;  
 }  
}

1. FFT

**package** com.it;  
  
*/\*  
\* 定义FFT（快速傅里叶转换）操作，利用递归方法  
\*  
\*/***public class** FFT {  
  
 *// compute the FFT of x[], assuming its length is a power of 2  
 //建立一个数组，数组长度是2的整数幂，方便二分* **public static** Complex[] fft(Complex[] x) {  
 **int** N = x.**length**;  
  
 *// base case* **if** (N == 1)  
 **return new** Complex[] { x[0] };  
  
 *// radix 2 Cooley-Tukey FFT* **if** (N % 2 != 0) {  
 **throw new** RuntimeException(**"N is not a power of 2"**);  
 }  
  
 *// 偶数项* Complex[] even = **new** Complex[N / 2];  
 **for** (**int** k = 0; k < N / 2; k++) {  
 even[k] = x[2 \* k];  
 }  
 Complex[] q = *fft*(even);  
  
 *// 奇数项* Complex[] odd = even; *// reuse the array 为了节省空间* **for** (**int** k = 0; k < N / 2; k++) {  
 odd[k] = x[2 \* k + 1];  
 }  
 Complex[] r = *fft*(odd);  
  
 *// combine，底层FFT操作* Complex[] y = **new** Complex[N];  
 **for** (**int** k = 0; k < N / 2; k++) {  
  
 *// 使用欧拉公式e^(-i\*2pi\*k/N) = cos(-2pi\*k/N) + i\*sin(-2pi\*k/N)* **double** kth = -2 \* k \* Math.PI / N;  
 Complex wk = **new** Complex(Math.cos(kth), Math.sin(kth));  
 y[k] = q[k].plus(wk.times(r[k]));  
 y[k + N / 2] = q[k].minus(wk.times(r[k]));  
 }  
 **return** y;  
 }  
}

1. ShiBieWav2:

**import** org.tritonus.sampled.convert.PCM2PCMConversionProvider;  
**import** javax.sound.sampled.\*;  
**import** java.io.ByteArrayOutputStream;  
**import** java.io.File;  
**import** java.io.IOException;  
  
*/\*  
 \* 识别音频文件信息，并转换成音频指纹  
 \*  
 \* \*/***public class** ShiBieWav2 {  
  
 **static int** *j* = 0;  
 AudioInputStream **in**;  
 String **filePath** = **"D:\\"** + (*j*++) + **".mp3"**;  
 File **file** = **new** File(**filePath**);  
 **boolean running** = **false**;  
  
 **public** String shiBieWenJianLiu() **throws** IOException, UnsupportedAudioFileException, LineUnavailableException {  
 String jieguo = **null**;  
 AudioInputStream din = **null**;  
 AudioInputStream outDin = **null**;  
 **in** = AudioSystem.getAudioInputStream(**file**);  
 PCM2PCMConversionProvider conversionProvider = **new** PCM2PCMConversionProvider();  
 AudioFormat formatTmp = **null**;  
 TargetDataLine lineTmp = **null**;  
 **final** AudioRecognizer ar = **new** AudioRecognizer();  
  
 *//获得in的文件格式* AudioFormat baseFormat = **in**.getFormat();  
 System.out.println(baseFormat.toString());  
  
 *//转换文件编码* AudioFormat decodedFormat = **new** AudioFormat(  
 AudioFormat.Encoding.PCM\_SIGNED,  
 baseFormat.getSampleRate(), 16, baseFormat.getChannels(),  
 baseFormat.getChannels() \* 2, baseFormat.getSampleRate(),  
 **false**);  
  
 *//把in流的音频格式转换成decodeFormat格式* din = AudioSystem.getAudioInputStream(decodedFormat, **in**);  
 **if** (!conversionProvider.isConversionSupported(ar.getFormat(), decodedFormat)) {  
 System.out.println(**"Conversion is not supported"**);  
 }  
 System.out.println(decodedFormat.toString());  
 outDin = conversionProvider.getAudioInputStream(ar.getFormat(), din);  
 System.out.println(ar.getFormat().toString());  
  
 **final** AudioInputStream outDinSound = outDin;  
  
 ByteArrayOutputStream out = **new** ByteArrayOutputStream();  
 *// running = true;* **int** n = 0;  
 **byte**[] buffer = **new byte**[(**int**) 1024];  
 **try** {  
 **while** ((n = outDinSound.read(buffer)) != -1) {  
 out.write(buffer, 0, buffer.**length**);  
 }  
 Complex[][] results = ar.makeFFT(out);  
 *//System.out.println(ar.determineFingerprints(results)+" "+"1");* jieguo = ar.determineFingerprints(results);  
 System.out.println(**"jieguo:"** + jieguo);  
  
 out.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **return** jieguo;  
 }  
  
 **public static void** main(String[] args) **throws** IOException, UnsupportedAudioFileException, LineUnavailableException {  
 ShiBieWav2 wjtfp = **new** ShiBieWav2();  
 wjtfp.shiBieWenJianLiu();  
 }  
}

1. WenJianToFingerprint:
2. */\*  
    \* 识别音频文件信息，并转换成音频指纹  
    \*  
    \* \*/***public class** WenJianToFingerprint {  
     
    AudioInputStream **in**;  
    String **filePath** = **"D:\\auto\\auto\\src\\main\\webapp\\music\\33 - Gonna Fly Now.mp3"**;  
    File **file** = **new** File(**filePath**);  
    **boolean running** = **false**;  
     
    **public void** shiBieWenJianLiu() **throws** IOException, UnsupportedAudioFileException, LineUnavailableException {  
     
    AudioInputStream din = **null**;  
    AudioInputStream outDin = **null**;  
    **in** = AudioSystem.getAudioInputStream(**file**);  
    PCM2PCMConversionProvider conversionProvider = **new** PCM2PCMConversionProvider();  
    AudioFormat formatTmp = **null**;  
    *//AudioFormat format2=getFormat();* TargetDataLine lineTmp = **null**;  
    **final** AudioRecognizer ar = **new** AudioRecognizer();  
    *//获得in的文件格式* AudioFormat baseFormat = **in**.getFormat();  
    System.out.println(baseFormat.toString());  
     
    *//转换文件编码* AudioFormat decodedFormat = **new** AudioFormat(  
    AudioFormat.Encoding.PCM\_SIGNED,  
    baseFormat.getSampleRate(), 16, baseFormat.getChannels(),  
    baseFormat.getChannels() \* 2, baseFormat.getSampleRate(),  
    **false**);  
     
    *//把in流的音频格式转换成decodeFormat格式* din = AudioSystem.getAudioInputStream(decodedFormat, **in**);  
    **if** (!conversionProvider.isConversionSupported(ar.getFormat(), decodedFormat)) {  
    System.out.println(**"Conversion is not supported"**);  
    }  
    System.out.println(decodedFormat.toString());  
    outDin = conversionProvider.getAudioInputStream(ar.getFormat(), din);  
    System.out.println (ar.getFormat().toString());  
     
    *//formatTmp = decodedFormat;  
    //DataLine.Info info = new DataLine.Info(TargetDataLine.class, ar.getFormat());  
    //lineTmp = (TargetDataLine) AudioSystem.getLine(info);* **final** AudioInputStream outDinSound = outDin;  
     
    Thread shiBieThread = **new** Thread(**new** Runnable() {  
    @Override  
    **public void** run() {  
    ByteArrayOutputStream out = **new** ByteArrayOutputStream();  
    *// running = true;* **int** n = 0;  
    **byte**[] buffer = **new byte**[(**int**) 1024];  
    **try** {  
    **while** ((n = outDinSound.read(buffer)) != -1) {  
    out.write(buffer, 0, buffer.**length**);  
    }  
     
    Complex[][] results = ar.makeFFT(out);  
    ar.addFingerprint(results, 33);  
     
    out.close();  
    } **catch** (IOException e) {  
    e.printStackTrace();  
    }  
     
    }  
    });  
    shiBieThread.start();  
    }  
     
    **public static void** main(String[] args) **throws** IOException, UnsupportedAudioFileException, LineUnavailableException {  
    WenJianToFingerprint wjtfp = **new** WenJianToFingerprint();  
    wjtfp.shiBieWenJianLiu();  
    }  
   }

### 11.2.2 The front-end code

* Index.html

<!DOCTYPE **html**>  
<**html xmlns="http://www.w3.org/1999/xhtml"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"**>  
 <**meta name="apple-mobile-web-capable" content="yes"**>  
 <**title**>语音转写</**title**>  
 <**link rel="stylesheet" type="text/css" href="css/style.css"**/>  
  
  
</**head**>  
<**body**>  
<**div id="container"**>  
 <**div id="player"**>  
 <**h1**>听歌识曲</**h1**></**br**>  
 <**button id="btn-start-recording" onclick="***startRecording*();**"**>录音</**button**>  
 <**button id="btn-stop-recording" disabled onclick="***stopRecording*();**"**>识别</**button**>  
 *<!--<a id="btn-start-palying" disabled href="">播放</a>-->* <**div id="inbo"**>  
 <**div id="change"**></**div**>  
 </**div**>  
 <**input type="hidden" id="audiolength"**>  
 <**hr**>  
 <**audio id="audioSave" controls** ></**audio**>  
 *<!--<audio src="music/27-Eversleeping.mp3" controls></audio>-->* <**textarea id="btn-text-content" class="text-content"**></**textarea**>  
 <**button id="historyBtn"**>历史记录</**button**>  
 <**div id="history" style="display**: **none**; **"**>  
 <**p**>识别历史</**p**>  
 <**ol**>  
 *<!--<li><a href="#" id="music/27-Eversleeping.mp3" class="liShi">27-Eversleeping.mp3</a></li>-->  
 <!--<li><a href="#" id="music/08-周杰伦-夜曲.mp3" class="liShi">08-周杰伦-夜曲.mp3</a></li>-->* </**ol**>  
 </**div**>  
 </**div**>  
</**div**>  
<**script type="text/javascript" src="js/jquery-1.11.0.js"**></**script**>  
<**script type="text/javascript" src="js/HZRecorder.js"**></**script**>  
<**script src="js/main.js"**></**script**>  
<**script type="text/javascript"**>  
 **$**(***document***).**ready**(**function** () {  
 **$**(**"#historyBtn"**).click(**function** () {  
 **$**(**"#history"**).toggle(1000);  
 })  
  
 *// $("a.liShi").click(function () {  
 // //console.log($(this).attr('id'));  
 // $("#audioSave").attr('src',"music/"+$(this).text());  
 // // console.log("music/"+$(this).text());  
 // var gequ1\_audio=$('#audioSave')[0];  
 // gequ1\_audio.load();  
 // gequ1\_audio.play();  
 // })* **$**(**"body"**).on(**"click"**,**"a.liShi"**,**function** (){  
 **$**(**"#audioSave"**).attr(**'src'**,**"music/"**+**$**(**this**).**text**());  
 **var** gequ1\_audio=**$**(**'#audioSave'**)[0];  
 gequ1\_audio.**load**();  
 gequ1\_audio.play();  
 } )  
 })  
</**script**>  
</**body**>  
</**html**>