**Intelligent song recommendation application based on expression recognition**

**Project Test Plan**

Project Name: Intelligent song recommendation application based on expression recognition

Team Member： Yujie Yao 201830310206

小组成员： Jiaying Zhu 201830310230

小组成员： Huifang Dong 201830310236

小组成员： Lingling Zhou 201830310256

April 25, 2019

**目录**

[1.Introduction 3](#_Toc7359583)

[1.1 Purpose 3](#_Toc7359584)

[1.2 Background 3](#_Toc7359585)

[1.3 Scope 4](#_Toc7359586)

[2. Test reference documentation and test submission documentation 5](#_Toc7359587)

[2.1 Test reference documentation 5](#_Toc7359588)

[2.2 Test submission documentation 6](#_Toc7359589)

[3. Terms and Definitions 6](#_Toc7359590)

[4. Test arrangement and process 6](#_Toc7359591)

[4.1 Resource requirements and timing for each test phase 6](#_Toc7359592)

[4.2 Project milestone 7](#_Toc7359593)

[5. System risk、priority 7](#_Toc7359594)

[5.1Priority 7](#_Toc7359595)

[5.2 Possible impact or risk 8](#_Toc7359596)

[6.Test strategy 8](#_Toc7359597)

[6.1 Data and database integrity testing 8](#_Toc7359598)

[6.2 Integration Testing 9](#_Toc7359599)

[6.3 Function test 10](#_Toc7359600)

[6.4 User interface test 10](#_Toc7359601)

[6.5 Performance Testing 11](#_Toc7359602)

[6.6 Security and access control testing 12](#_Toc7359603)

[6.7 Failover and recovery testing 13](#_Toc7359604)

[6.8 Installation test 15](#_Toc7359605)

[7.Test items 16](#_Toc7359606)

[7.1 Emotion recognition module 16](#_Toc7359607)

[7.2 Song recommendation module 16](#_Toc7359608)

[7.3 Test case 16](#_Toc7359609)

[8. Problem severity description 19](#_Toc7359610)

# 1.Introduction

## 1.1 Purpose

This document is mainly for the purpose of providing test-related information for "smart song recommendation based on expression recognition".

The purpose of the test plan is to test whether the various functional modules of the system meet the user requirements, and test whether there are bugs. It is expected to achieve rapid system improvement and system improvement. In order to use the software system when it is used, it can detect system errors as much as possible while improving system performance.

This Test Plan document will help achieve the following objectives:

1. The system intelligently recommends songs by recognizing the user's facial expressions. The main functional modules are model training, expression recognition and song playback. These three modules are the parts of the project that need to be tested.

2. The test requirements are functional requirements and user requirements. The functional requirements mean that the recommendation system is to be able to recognize the expression and play the song. The user requirements mean that the user who uses the recommendation system can accurately recognize the expression and obtain the recommendation of the relevant song.

3. The corresponding test strategies used in this document include data and database integrity testing, interface testing, functional testing, user interface testing and other related strategies. Specific test strategy related information is described in detail in this document.

4. This test does not require additional resources. It only needs to be able to run the system and perform related tests. The workload of the test is estimated, and the division of labor can be achieved.

5. The test items need to be delivered are test related documents, such as feasibility analysis report, software requirements analysis, test plan, etc.

## 1.2 Background

The main test of this project is that the intelligent song recommendation based on expression recognition is to provide the user with a system for recommending and mood-related songs according to the user's mood at that time, which can meet the atmosphere required by the user's mood at that time.

Identifying user emotions requires a certain amount of samples to be trained to obtain a relevant predictive model. Therefore, whether the trained sample model can accurately recognize the facial expression needs to be tested. Again, the song is recommended in the module. Whether the song database can match the expression accurately, whether the interface of the expression recognition module and the song playing module is accurate, whether the song can be successfully played, etc. need to be tested.

## 1.3 Scope

Whether the function of the main test software meets the needs of the user, whether the performance is superior and the system has problems.

Perform detailed tests on the expression recognition module and song playback module of the system, and record the test results. Perform a detailed analysis of the results of the test. During the test, each functional module of the system is split and tested, and each module is tested. The expression recognition module and the song playing module of the system are tested in detail, and the test results are recorded. Then, the results of the test are carefully analyzed. During the test, each functional module of the system is split and tested, and each module is tested.

The expression recognition module is split into three parts:

First, the image path to identify the expression is open correctly;

Second, whether the image to be recognized is normalized and whether the face can be detected after being converted into a grayscale image;

Third, whether the expression predicted by the face grayscale image is accurate according to the model that has been trained.

What the song playback module wants to test is whether the player can play, pause, stop, and switch songs.

We test all possible outcomes, as well as the problems that exist during the testing, and then submit a record of the tests. Finally, a comprehensive analysis of the software problems and performance tests is given and recorded.

# 2. Test reference documentation and test submission documentation

## 2.1 Test reference documentation

The following table lists the documentation used to develop the test plan and indicates the availability of each document:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document | Created or available | Accepted or Reviewed | Author or source | Remarks |
| Feasibility analysis report | Yes□　No□ | Yes□　No□ |  |  |
| Software requirements definition | Yes□　No□ | Yes□　No□ |  |  |
| Software System Analysis (STD, DFD, CFD, DD) | Yes□　No□ | Yes□　No□ |  |  |
| Software summary design | Yes□　No□ | Yes□　No□ |  |  |
| Software detailed design | Yes□　No□ | Yes□　No□ |  |  |
| Software testing requirements | Yes□　No□ | Yes□　No□ |  |  |
| Hardware feasibility analysis report | Yes□　No□ | Yes□　No□ |  |  |
| Module development manual | Yes□　No□ | Yes□　No□ |  |  |
| Test schedule and staffing | Yes□　No□ | Yes□　No□ |  |  |
| Test Plan | Yes□　No□ | Yes□　No□ |  |  |
| Test program | Yes□　No□ | Yes□　No□ |  |  |
| testing report | Yes□　No□ | Yes□　No□ |  |  |
| Test analysis report | Yes□　No□ | Yes□　No□ |  |  |
| User operation manual | Yes□　No□ | Yes□　No□ |  |  |
| Installation guide | Yes□　No□ | Yes□　No□ |  |  |

## 2.2 Test submission documentation

The following is a list of all documents that can be submitted after the end of the test phase:

(1) Test cases: Through the preparation before the test and the summary after the test, the test cases are continuously improved, and the priorities are classified according to the modules.

(2) Test log: The tester summarizes daily or phased, records what has been done, including the part where the problem is not found and the evaluation of the current status of the module and system. The above records are saved in the Notes library.

(3) Defect Report: Use Bugfree records and VSS to manage defect reports. After the test is completed, the VSS is organized and analyzed.

(4) Summary: After the acceptance test is completed, fill in the acceptance test summary report as required, and summarize the entire test process.

# 3. Terms and Definitions

（1）Bugfree: Drawing on Microsoft's R&D process and Bug management philosophy, a bug management system written independently using PHP+MySQL

（2）VSS: Visual Source Safe. The main task is to manage the project files, manage the different versions of the source code and documentation in the software development, occupy small space and facilitate the acquisition of each version of the code and documents, and effectively coordinate the access to the source code in the development team.

# 4. Test arrangement and process

## 4.1 Resource requirements and timing for each test phase

|  |  |  |
| --- | --- | --- |
| Test activity | Person | Time schedule |
| Develop a test plan | Yujie Yao、Lingling Zhou | 2019.04.24 - 2019.04.25 |
| Design test | Huifang Dong | 2019.04.25- 2019.04.26 |
| Integration Testing | Jiaying Zhu、Huifang Dong、Yujie Yao、Lingling Zhou | 2019.04.26 - 2019.04.27 |
| System test | Jiaying Zhu、Huifang Dong、Yujie Yao、Lingling Zhou | 2019.04.27 - 2019.04.28 |
| Acceptance test | Huifang Dong | 2019.05.04 - 2019.05.10 |

## 4.2 Project milestone

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone mission | Workload | start date | End date |
| Develop a test plan | 2 people / day | 2019.04.24 | 2019.04.25 |
| Develop a test plan | 4 people / day | 2019.04.25 | 2019.04.26 |
| Design test | 1 person / day | 2019.04.26 | 2019.04.27 |
| Integration Testing | 4 people / day | 2019.04.28 | 2019.04.29 |
| System test | 4 people / day | 2019.05.04 | 2019.05.05 |
| Acceptance Test | 1 person/week | 2019.05.05 | 2019.05.10 |

# 5. System risk、priority

## 5.1Priority

The following table lists system risk management and priority descriptions.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Functional testing phase | Installation test phase | Document test |
| Correctness | H | H | H |
| File integrity | H | H | H |
| Processing continuity | M | M | M |
| Access control | M | M | M |
| Compliance | H | H | H |
| reliability | H | H | H |
| Easy to operate | H | H | H |
| Maintainability | H | H | H |
| portability | H | H | H |

## 5.2 Possible impact or risk

* The test time is not enough, mainly because the system test time after the function freeze may not be enough.
* Test resources are not available in time (mainly personnel issues)
* Change in development schedule, change in requirements or design

# 6.Test strategy

The test strategy provides a recommended method for testing test objects.

For each test, test instructions should be provided and the reasons for their implementation should be explained.

The main considerations when developing a test strategy are: the technology to be used and the criteria for determining when the test is completed.

The following is a list of things to consider when doing each test. In addition, tests should only be performed using a known, controlled database in a secure environment.

Note: If you do not implement a test, you should explain it in one sentence and state the reason. For example, "This test will not be implemented. This test is not applicable."

## 6.1 Data and database integrity testing

In the intelligent music recommendation system based on face recognition, the database and database processes should be tested as a subsystem. When testing these subsystems, the test object's user interface should not be used as an interface to data. For database management systems (DBMS), in-depth research is needed to identify tools and techniques that can support the following tests.

|  |  |
| --- | --- |
| Test target: | Ensure that database access methods and processes are functioning properly, data is not corrupted, and data consistency is ensured. |
| Test Range: | All functional modules of intelligent music recommendation system based on face recognition |
| Technology: | Invoke individual database access methods and processes and populate them with valid and invalid data (or requests for data).Check the database to ensure that the data has been populated as expected, and that all database events have occurred normally; or to check the data returned to ensure that the correct reason has retrieved the correct data |
| Starting standard: | The database is up and running, the test version has been submitted for testing |
| Completion criteria: | All database access methods and processes are run as designed and the data is not corrupted.  The planned tests have all been implemented and the defects found have been fully resolved. |
| Special considerations: | Testing may require the DBMS development environment or driver to directly enter or modify data in the database.  The process should be called manually.  Small or minimal databases (limited number of records) should be used to make all unacceptable events more visible. |

## 6.2 Integration Testing

Integration Test - The main purpose is to detect whether the system meets the requirements of whether the processing of business processes and data flows meets the standards. The detection system has logic rigor and errors in the processing of business flows, and whether there are unreasonable standards and requirements for detecting requirements. This phase tests based on functional completion.

|  |  |
| --- | --- |
| Test target: | Detect the business process in the demand, the correctness of the data flow |
| Test Range: | A clear business process in the requirements, or a combination of different functional modules to form a large function. |
| Technology: | Use valid and invalid data to execute individual use cases, use case flows, or features to verify the following:  Get the expected results when using valid data.  Displays an appropriate error or warning message when invalid data is used.  Every business rule has been applied correctly. |
| Starting standard: | Must meet the standard when completing an integration test |
| Completion criteria: | The planned tests have all been implemented.  The defects found have all been resolved. |
| Special considerations: | Identify or account for issues or factors that will affect the implementation and execution of functional testing (internal or external) |

## 6.3 Function test

Functional testing of test objects should focus on all testing requirements that can be traced directly to use cases or business functions and business rules. The goal of this test is to verify that the data is accepted, processed, and retrieved correctly, and that the implementation of business rules is appropriate. This type of testing is based on black box technology, which interacts with the application through a graphical user interface (GUI) and analyzes the output or results of the interaction to verify the application and its internal processes. The recommended test summary is listed below for various applications:

|  |  |
| --- | --- |
| Test target: | Ensure that the system's functionality meets the requirements defined in the requirements specification, including data entry, processing, and retrieval. |
| Test Range: | All modules of intelligent music recommendation system based on face recognition |
| Technology: | Use valid and invalid data to execute individual use cases, use case flows, or features to verify the following:  Get the expected results when using valid data.  Displays an appropriate error or warning message when invalid data is used.  Every business rule has been applied correctly. |
| Starting standard: | Code completion, submit test |
| Completion criteria: | Defect repair is greater than 90% |
| Test focus and priority: | Consistent with the requirements in the requirements specification |
| Special considerations: | Identify or account for issues or factors that will affect the implementation and execution of functional testing (internal or external)Defect repair rate calculation rule: defect repair rate = number of defects closed by verification / total number of defects |

## 6.4 User interface test

User interface (UI) tests are used to verify the interaction between the user and the software. The goal of user interface testing is to ensure that the user interface provides the user with access or browsing capabilities through the functionality of the test object. In addition, UI testing ensures that objects in the UI operate as expected and meet company or industry standards.

|  |  |
| --- | --- |
| Test target: | Verify the following:  Browsing through testing correctly reflects the functionality and needs of the business, including browsing between windows and windows, between fields and fields, and the use of various access methods (Tab, mouse, and shortcuts).  For example: whether the interface is beautiful, whether the interface is intuitive, the operation is friendly, whether it is humanized, and the ease of operation is good.  The objects and features of the window (for example, menu, size, position, status, and center) are all compliant. |
| Test Range: | All functional modules of intelligent music recommendation system based on face recognition |
| Technology: | Create or modify tests for each window to verify that individual application windows and objects are properly viewed and in a normal object state. |
| Starting standard: | System interface design completed and passed the review |
| Completion criteria: | Consistent with UI requirements in the requirements specification or in compliance with acceptable standards |
| Test focus and priority: | Consistent with the requirements in the requirements specification |
| Special considerations: | Not all features of custom or third-party objects are accessible. |

## 6.5 Performance Testing

Performance tests measure and evaluate response times, transaction rates, and other time-related requirements. The goal of performance evaluation is to verify that performance requirements are met. The purpose of implementing and performing performance reviews is to evaluate and fine tune the performance behavior of test objects as a function of conditions such as workload or hardware configuration.

Note: The following transactions refer to "logical business transactions." This transaction is defined as the addition or modification of a given contract by a particular use case that will be performed by an Actor of the system using the test object.

|  |  |
| --- | --- |
| Test target: | Verify the performance behavior of the specified transaction or business function under the following conditions:  Normal expected workload  Expected most heavy workload |
| Test Range: | Queue message, topic message (concurrent access) |
| Technology: | Use a test process developed for functional or business cycle testing.  Increase the number of transactions by modifying the data file, or increase the number of iterations per transaction by modifying the script.  Scripts should run on a single computer (preferably on a single user basis, on a single transaction basis) and on multiple clients (virtual or actual clients, see "Special Considerations" below) repeat. |
| Starting standard: | Functional test completed |
| Completion criteria: | Single transaction or single user: The test script was successfully completed within the expected time frame of each transaction without any failure.  Multiple transactions or multiple users: The test script was successfully completed within an acceptable time frame without any failure. |
| Test focus and priority: | Consistent with the requirements in the requirements specification |

## 6.6 Security and access control testing

Security and access control testing focus on two key aspects of security:

Application-level security, including access to data or business functions.

System level security, including login or remote access to the system.

Application-level security ensures that Actors can only access specific features or use cases or have limited access to limited data under the expected security conditions. For example, you might allow everyone to enter data and create a new account, but only an administrator can delete the data or account. If there is data-level security, the test ensures that User Type One can see all customer messages, while User Two sees the same customer's statistics.

System-level security ensures that only users with system access can access the application and can only access it through the appropriate gateway.

|  |  |
| --- | --- |
| Test target: | Application-level security: Verify that an Actor can only access those features or data that the user type to which it belongs is authorized to access.  System level security: [Verify that only Actors with system and application access can access systems and applications. |
| Test Range: | Intelligent music recommendation system based on face recognition |
| Technology: | Application-level security: Identify and list each user type and the features or data that it is authorized to access.  Create tests for each user type and verify their permissions by creating transactions specific to each user type.  Modify the user type and rerun the test for the same user. For each user type, ensure that these additional features or data are properly provided or rejected.  System level access: See "Special considerations" below. |
| Starting standard: | Submitted by intelligent music recommendation system module based on face recognition |
| Completion criteria: | Various known Actor types have access to the corresponding function or data, and all transactions run as expected and all transactions are run in previous application functional tests. |
| Test focus and priority: | Consistent with the requirements in the requirements specification |
| Special considerations: | System access rights must be checked and discussed with the appropriate network or system administrator. Since this test may be a system managed function of network management, you may not need to perform this test. |

## 6.7 Failover and recovery testing

Failover and recovery testing Cocoa ensures that test subjects successfully complete the transfer and recover from various hardware and software network failures that result in unexpected data loss or data integrity breaches.

Failover testing ensures that, for systems that must continue to operate, in the event of a failure, the standby system will “replace” the failed system without loss of time to avoid losing any data or transactions.

Recovery testing is a confrontational testing process. In this test, the application or system will be placed under extreme conditions (or under extreme conditions of the simulation) to generate faults (such as device input/output (I/O) faults or invalid database pointers and keys. word). Then call the recovery process and monitor and check the application and system to verify that the application or system and data have been properly restored.

|  |  |
| --- | --- |
| Test target: | Ensure that the recovery process (either manually or automatically) properly restores the database, applications, and systems to the expected known state.  The test will include the following:  Client power off  Server power down  Communication interruption generated by the web server  DASD and/or DASD controllers are interrupted, powered down, or interrupted communication with DASD and/or DASD controllers  The cycle is not completed (the data filtering process is interrupted and the data synchronization process is interrupted).  Database pointer or keyword is invalid  Invalid or corrupted data elements in the database |
| Test Range: | All functional modules of intelligent music recommendation system based on face recognition |
| Technology: | You should create a series of transactions using tests created for functional and business cycle testing. Once the expected test starting point is reached, the following actions should be performed or simulated separately:  Client power off: Turn off the power of the PC.  Server power off: simulates or starts the server's power down process.  Interruption generated by the network server: Interruption of communication that simulates or starts the network (actually disconnects the communication line or turns off the power of the network server or router). The DASD and DASD controllers are interrupted, powered down, or interrupted communication with the DASD and DASD controllers: simulating communication with one or more DASD controllers or devices, or actually canceling such communications.  Once the above situation (or simulation) is achieved, other transactions should be performed. And once the second test point state is reached, the recovery process should be called. When testing an incomplete cycle, the technique used is the same as above, except that the database process itself should be terminated abnormally or prematurely.  Testing for the following situations requires reaching a known database state. When destroying several database fields, pointers, and keywords, it should be done directly in the database (via the database tool). Other transactions should be performed by using tests in Application Functional Testing and Business Cycle Testing, and a full cycle should be performed. |
| Starting standard: | Functional test of intelligent music recommendation system based on face recognition has ended |
| Completion criteria: | In all of the above cases, the application, database, and system should immediately return to a known, expected state when the recovery process is complete. This status includes data corruptions that are limited to fields that are known to be corrupted, pointers or keywords, and reports that indicate that the process or transaction was not completed due to a section. |
| Test focus and priority: | Consistent with the requirements in the requirements specification |
| Special considerations: | Recovery testing can cause a lot of trouble for other operations. The method of disconnecting the cable (analog power outage or communication interruption) may not be desirable or feasible. Therefore, other methods, such as diagnostic software tools, may be needed.  Requires resources in the system (or computer operations), databases, and network groups.  These tests should be run outside of working hours or on a separate computer. |

## 6.8 Installation test

Installation testing has two purposes. The first purpose is to ensure that the software can be installed under normal conditions and under different conditions, for example, for the first time installation, upgrade, full or custom installation. Abnormalities include insufficient disk space, lack of directory creation permissions, and so on. The second purpose is to verify that the software is up and running as soon as it is installed. This usually means running a large number of tests developed for functional testing.

|  |  |
| --- | --- |
| Test target: | Verify that the test object is properly installed into the various required hardware configurations under the following conditions:  Install for the first time. New computer that has never been installed before  Update. A computer that has previously installed the same version of the system  Update. An earlier version of the computer that has previously installed the system |
| Test Range: | Installation program of intelligent music recommendation system based on face recognition |
| Technology: | Start or perform the installation.  Run the transaction using a subset of the predefined functional test scripts. |
| Starting standard: | A complete installation package for intelligent music recommendation system based on face recognition has been submitted |
| Completion criteria: | The intelligent music recommendation system transaction based on face recognition was successfully executed without any failure. |
| Test focus and priority: | Consistent with the requirements in the requirements specification |
| Special considerations: | Which of the system's transactions should be selected to accurately test that the system application has been successfully installed, and that the main software components are not missing. |

# 7.Test items

## 7.1 Emotion recognition module

Main features: photo upload, image capture, expression recognition

Test content:

1. It is correct to open the picture path of the expression to be recognized;

2. Whether the image to be recognized is normalized and whether the face can be detected after being converted into a grayscale image;

3. Whether the expression predicted by the grayscale image of the face is accurate according to the model that has been trained.

4. The user clicks the “Emotion Recognition” button to check whether photos can be taken normally or uploaded.

## 7.2 Song recommendation module

Main features: playback controls, music library, song recommendations (including category recommendations and random recommendations)

Test content:

1. Open and close the player

2. Play, pause, stop the player

3. Previous video, next video

4. Volume, mute

5. Maximize, minimize

6. Add, delete, view playlists

7. Playlist play order, single loop, multi-loop, sequential play, then play

## 7.3 Test case

|  |  |  |
| --- | --- | --- |
| Function A Description | Get all the music in your local music library | |
| Use purpose | Test if you can successfully get all the music in your local music library | |
| Precondition | None | |
| Input/Action | Expected output | The actual situation |
| Absolute path to the music library | Get all the ".mp3" files in the folder, store the song file names in a list, and store their corresponding song information (such as song title, singer, etc.), and finally display them in the WEB music list. |  |
| Function B Description | Music playback function | |
| Use purpose | Test whether the music player can perform normal operations (such as switching songs, starting pause, volume control, song progress control, etc.) | |
| Precondition | Successfully got all the music in the local music library | |
| Input/Action | Expected output | The actual situation |
| Click the "Previous" button | Correctly capture and display the song information of the previous song and automatically play the previous song |  |
| Click the "Next" button | Get and display the song information of the next song correctly, and automatically play the next song |  |
| Click the "Play" button | Start playing the current song |  |
| Click the "Pause" button | Pause the currently playing song |  |
| Drag the volume control progress bar | The volume changes correctly with the drag control |  |
| Drag the song progress bar | Song progress can be switched correctly |  |
| Click play mode icon | Switch playback mode (three modes: sequential playback, random playback, single loop), able to play music correctly according to the current playback mode |  |
| Click on a song in the music list | Switch to the song correctly |  |
| Function C Description | Search for songs | |
| Use purpose | Test if you can search for music in your local music library; if the song is not in your local music library, test if you can search for it on multiple music sites | |
| Precondition | Successfully got all the music in the local music library | |
| Input/Action | Expected output | The actual situation |
| Enter the title of the song in the text box (the song already in the music library) and click the "search" button. | Locate the song's location in the web music list and highlight it |  |
| Enter the title of the song in the text box (the song not in the music library) and click the "search" button. | Aggregate searches for the song on multiple music sites and display a list of search results. |  |
| Function D Description | Download songs | |
| Use purpose | Test if you can download music from your website to your local music library | |
| Precondition | Successfully aggregated information about the song on multiple music sites | |
| Input/Action | Expected output | The actual situation |
| Click on one of the aggregated search results lists | Successfully download music to your local music library and display it in the web music list |  |
| Function E Description | Upload face expressions | |
| Use purpose | Test whether you can successfully upload facial expression images and display them on the web. | |
| Precondition | None | |
| Input/Action | Expected output | The actual situation |
| Click Upload Image to select image file (including single face) | Successfully detected the face and displayed the image on the web |  |
| Click Upload Image to select image file (without face) | No face is detected, an error message is displayed, and the user is allowed to reselect the picture. |  |
| Click Upload Image to select image file (including multiple faces) | Multiple faces detected, an error message is displayed, allowing the user to reselect the image |  |
| Function F Description | Expression recognition | |
| Use purpose | Test whether the face expression in the picture can be correctly recognized | |
| Precondition | Success on the flyer personal face expression picture | |
| Input/Action | Expected output | The actual situation |
| Absolute path of face image | Successfully identified the correct expression (happy, sad, etc.) |  |
| Function G Description | Recommended song | |
| Use purpose | Test whether the corresponding song can be recommended based on the recognized expression | |
| Precondition | Successfully on the flyer personal face expression picture and correctly recognize the expression | |
| Input/Action | Expected output | The actual situation |
| Click the "Recommended Songs" button | According to the recognized expression, find the song corresponding to the label in the database, and load the song into the music list of the web. |  |

# 8. Problem severity description

|  |  |
| --- | --- |
| Problem severity | Description |
| High | 1. The crash caused by the program, illegally quit  2. Software error message is sent out  3. The software does not respond  4. Program interruption due to incorrect operation5. The main function is lost or the function is seriously wrong. |
| Mid | 1. Program error  2. Program interface error |
| Low | 1. Operation interface error (including the definition of column names in the data window, whether the meaning is consistent)  2. Simple input restrictions are not placed in the foreground for control |