COMP3032J 2021-2022 Online Music Playing System
Group4 : Atom

Chunyu Li 18206177
Yunkai Li 18206372
Haotian Li 18206169
Zhongqi Kang 18206175
Xuhui Shi 18206370

SUPERVISOR: Dr.Mooney Catherine, Dr.Becker Brett, Dr.Shen Wang, Dr.Ruihai Dong, Dr.Henry McLaughlin

Abstract

The project background and requirements analysis will be carried out in the Introduction section. The technical solutions and challenges will be in the Technical Realization part. Then we will talk about how we carry out efficient teamwork in the Group Work section. Finally, we will summarize in the conclusion section.

The project is based on Python-Flask and Vue. The following functions have been realized.

Users can access our website using URL (http://csi420-01-vm3.ucd.ie). In addition, you can visit our Gitee repository (https://gitee.com/atom23/music-website) to view all source code.

Contents

1 Introduction

This project aims to create a music website. Users can find almost all kinds of music and discover more favorite music here. We will also implement some complex functions for solving specific problems and providing a better user experience.

1.1 Background

Music is a kind of abstract art. It has greatly enriched people's spiritual lives and gradually become an essential part of their daily entertainment. At present, the Internet is widely used by ordinary people, and the intelligent digital music service platform built on the Internet has an extensive market. Therefore, we decided to design a web-based streaming media music platform that can intelligently recommend music to users, including advanced functions such as music recognition, to bring a better auditory experience to the majority of music lovers.

1.2 Requirement Analysis and System Design

According to the requirements for a music website, to boost the efficiency of the system, our team proposes to address the following issues, which can be divided into two different parts:

1.2.1 Functional requirements

- Users will be allowed to visit the website by registering a new account first.
- Users will be allowed to listen to music, either get the recommended music by the system or search for the music they like.
- Users will be allowed to edit their personal information.
- Users will be allowed to upload or record a piece of music to identify the name of the music.
- Users will be allowed to get visualized data about listening to music using this website.

1.2.2 Non-functional requirements

This website can be accessed anywhere on the earth, whether using computers or mobile devices if connected to the Internet. This project has been deployed on the UCD Virtual Machine.

- Performance
  - The response will be very fast, with a time-out period of no more than two seconds in common.
  - The website will support more than 50,000 users online simultaneously.
- Reliability
  - The website should be robust to potential attacks without making errors.
- Security
  - The users' account password will be encrypted in SHA-256.
  - Token is used to ensure security.
- Compatibility
  - The website should be compatible and adaptive on both computers and mobile devices. The structure and interaction can be correctly displayed on mobile devices.
- Usability
  - The features should be easy to access and operate by users.
  - The web pages should have a user-friendly design and are easy to understand and use.
  - The information related to the users and music resources (for recognition) should be structured and stored in proper databases,
  - The database of music resources (for recognition) should be extended to identify music as much as possible.
  - The system should try to prevent users from listening to music for too long, which could potentially harm ears.

1.3 System Overview

This system is developed as a cloud-based music website to provide music services for users. Based on the requirements, the project mainly consists of two sub-systems: the user sub-system and the music sub-system.

1.3.1 The user sub-system

For each user account, login, logout, and registration functions are provided.
After login, users are allowed to use the functions of listening to music by keyword searching or recommendation from the system.
Moreover, users are also allowed to edit their personal information, of which some are not required in registration.

1.3.2 The music sub-system

All the music resources are acquired from the Net-Ease Cloud Music API.
The music recommendation function is based on deep learning, depending on the users' preference.
The music recognition allows users to uploading or recording a piece of music to identify the name of the music.
All the data related to listening to music from this website for each user will be recorded and visualized directly.

2 Groupwork

In terms of teamwork, our group has adopted a clear personal division of labor and group responsibility distribution system. Establish a standardized and assessable responsibility system by clearly delineating each member's team role, tasks to be completed, and the final implementation effect. The specific content is introduced as follows.

2.1 Labor Division

This development adopts the agile development model. The project is divided into a shorter product interaction cycle so that the project development can adapt to the customer needs change and team communication to understand the original business and needs of the product. In addition, testing should be done from the beginning of the project into the production process, rather than at the end when all features are complete. This dramatically reduces the impact of changes on the plan.

According to the Agile team role *(Reprinted from Journal of System Architecture, 52, Dubinsky Y, Hazzan O. Using a role scheme 5to derive software project quality, 693-699, Copyright (2006))*, Our team is divided into four groups, which are Leading group, customer group, code group, and maintenance group. According to the number of team members, their specialties, and abilities, they are divided as follows:

Name	Role	Duties and Responsibilities	Contribution
Cunyu Li	Leader Group	As the core team leader, based on the overall project to layout the whole team division of labor and process allocation. Major responsibilities include but not limited to: coordinating and solving team problems, leading and directing development meetings; Seek methodological solutions to major problems and decisions.	20%
Haotian Li	Customer Group	As a bridge between the development team and customers, the customer group first needs to communicate with customers in various aspects, take various ways to obtain as many user needs and potential intentions as possible, and correctly convey them to the development team. At the same time, throughout the development process, should continue to follow up with the customer to deal with any contingencies or any questions. Adoption of a user-centric approach, work with customers to define and develop acceptance tests, stimulate the most highly driven development process, and conduct continuous user evaluation of the product.	20%
Zhongqi Kang	Code Group (Front)	The code team, as the product producer, directly contacts and participates in the whole process of development of the whole project. The task should be to maintain the current design, simplify the design, search for refactoring tasks, and ensure that they are executed correctly. The front-end code group should complete the front-end page design of the web page, meet the needs of users and be able to interact normally, and provide sufficient interfaces for the back-end.	20%
Xuhui Shi	Code Group (Back)	The code team, as the product producer, directly contacts and participates in the whole process of development of the whole project. The task should be to maintain the current design, simplify the design, search for refactoring tasks, and ensure that they are executed correctly. The back-end code group focuses primarily on the implementation of functionality. The interface of the front end is used to obtain the user data and interact with the database to complete the system operation and feedback.	20%
Yunkai Li	Maintenance Group	The maintenance group is responsible for planning and organizing the iteration/release demos, demos, and personas; We make timely adjustments and modifications to the group's tasks through weekly discussion, and change our time for new function development. And then we plan and organize project documentation such as processing documentation, user guides, and installation instructions plan. We also ensure the development of automated installation suites, maintain collaborative workspace infrastructure.	20%

2.2 RACI Matrix: Individual Responsibilities

RACI Chart	**Person**				
Activity	Cunyu Li	Haotian Li	Zongqi Kang	Xuhui Shi	Yunkai Li
Communication With User	A	R	I	I	I
Create UI Sketches	I	C	R	I	I
Create Web Pages	I	I	R	C	I
Function Development	A	C	I	R	I
Database Design	A	I	I	C	R
UI Test	I	I	C	I	I
Function Test	I	I	I	C	C
Development Documentation	R	I	A	I	I
Confirm With User	I	R	C	C	I

R = Responsible
A = Accountable
C = Consult
I = Inform

3 System Development

3.1 Development Approach

Considering the system contains many functions and short development duration, we decided to adopt the agile development method. This iterative development approach of the agile process can help the team better meet user needs and make timely updates to the development plan when there are new requirements for the system improvement. Moreover, during the development process, we had three time nodes to present to our mentors to track the progress of our project completion.

3.2 Development Schedule

This project started on 21st February and is due on 15th May 2022. The milestones decided to be achieved as follows:

Description	Forecast Date	Gate / Approval
Complete requirements acquisition	2022-02-25	2022-02-25
Division of labor and formulation of project development plan	2022-03-01	2022-03-01
Create project architecture and warehouse	2022-03-09	2022-03-08
Complete the basic framework construction	2022-03-20	2022-03-18
Produce an operable demo version	2022-04-03	2022-04-01
Complete system function development	2022-04-24	2022-04-20
Deploy the project on the Virtual Machine	2022-04-24	2022-04-21
Integration, test and improvement	2022-05-05	2022-05-01
Complete documentation	2022-05-15	2022-05-15

3.3 Risk Management

Since the function of music recognition may be difficult to implement, we planned to leave two or three weeks to finish it based on the NLP technique. If we don't complete innovative functions before week 9, we will remove the problematic function and put more effort into increasing the quality and integrity of the project.
Finally, the function of music recognition has been completed, and it works well. However, instead of using NLP as we planned, we chose to use the Shazam algorithm to implement this function.

4 Technical Implementation

4.1 Justification

In addition to the design of the system architecture and technology, the project is small, the development time is short, and the new deep learning algorithm is included, so we decided to user FLASK, a lightweight framework based on the python language, which is used to adapt to more database operations, form validation, etc. On the front end, our team also adopted the lightweight Vue framework to focus on the simple view layer and implement the two-way data binding, which provides a faster speed of operation for the site. The database is used for medium data storage, which does not require any installation and configuration.

4.1.1 Framework

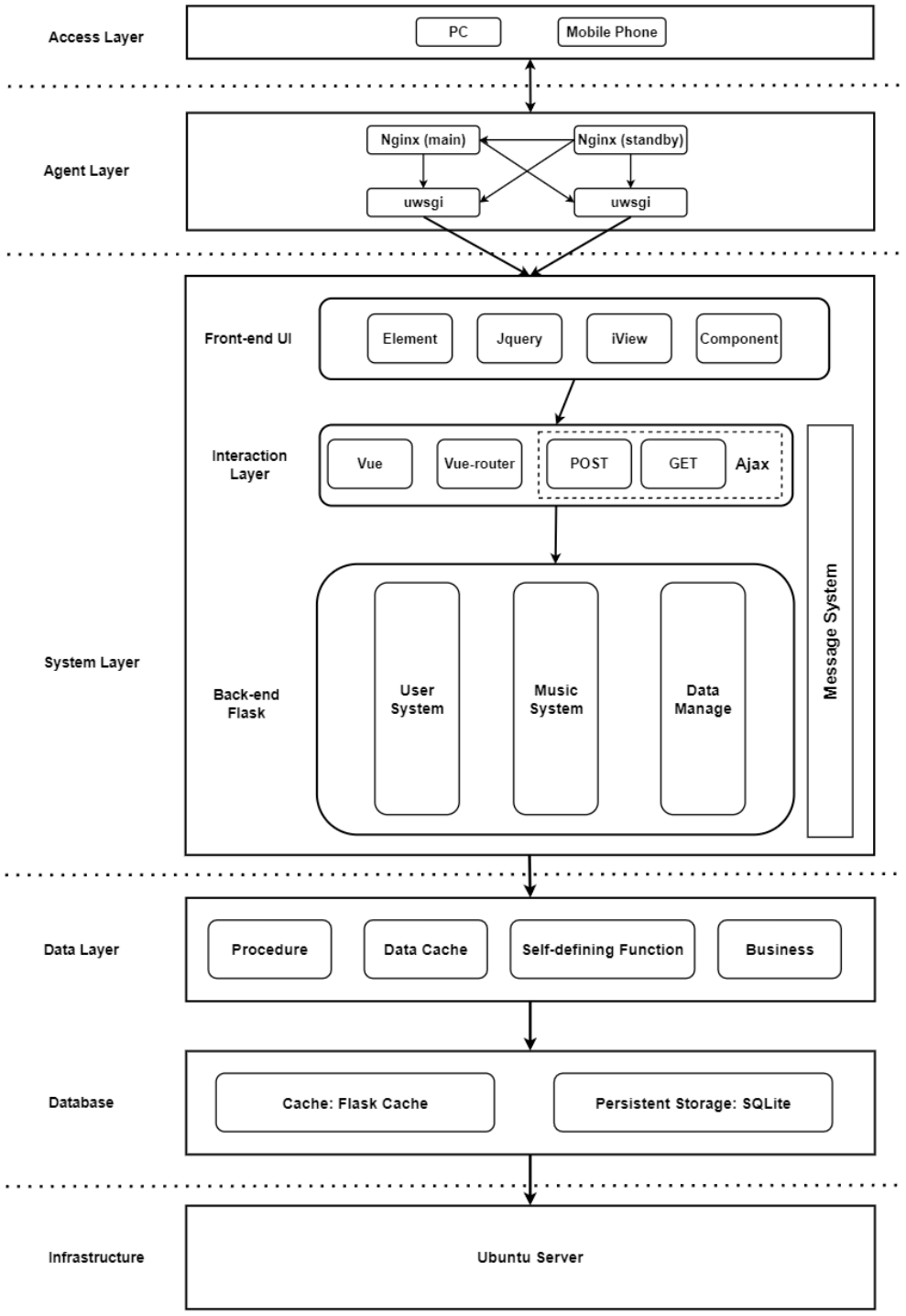


Figure 1: System Framework

4.1.2 Cross domain

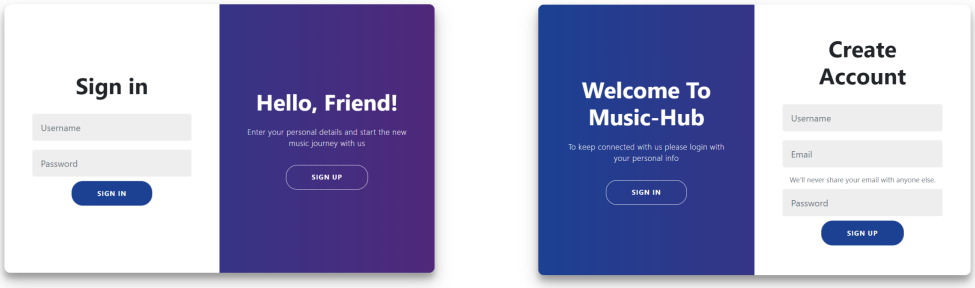
The back and front end are separated, so we use Axios to implement interaction between them. Meanwhile, CORS is also needed to implement cross-domain access due to the back and front ends starting in two different ports.

4.2 The Front-end

Based on the componentized design concept of Vue, the front-end page uses the Element Plus plug-in to help the design of components and page adaptation. Website page design is divided into four big page parts. They are the "Login and Registration Page", "Music Page", "User Information Page" and "Recognize Page".

- **Login and Registration Page**

  The login and registry interface is the portal page of the site, so we designed a separate page to show it. Given the integration and user-friendliness of the page, we combine the card elements to merge the login and registration and allow users to display different content according to different options.



- **Music Page**

  Music as the central part of the site, we provide users with two options: "listen to songs randomly" and "listen to songs carefully", corresponding to our home page and song details page, respectively. On the home page, the user can hear the current song's audio and switch songs through the mouse wheel to achieve the effect of extensive access to songs. The sliding operation can reduce the user's operation burden and increase the user's sense of participation and interactive experience. Page with user scrolling operation using infinite fill list control, and set to full screen to achieve the effect of the wheel.

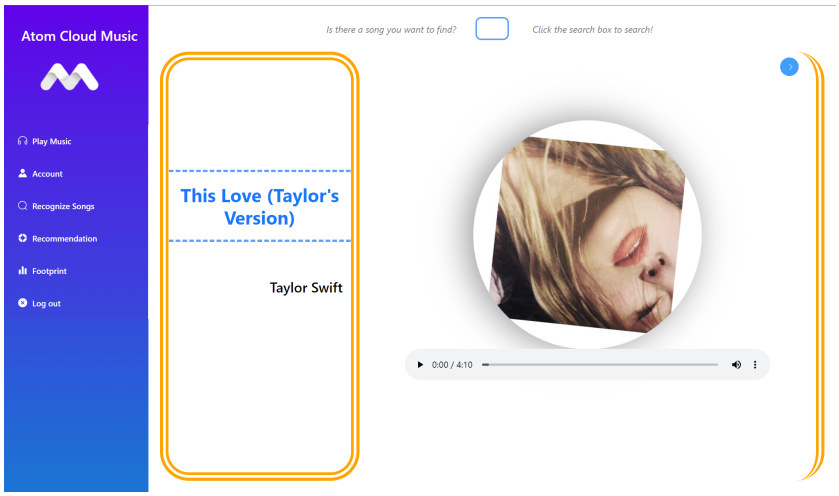


Figure 2: Home Page

When the user is interested in the random song, they can click the button in the upper right corner to enter the song details interface. In the song Details interface, we provide a player-like page with lyrics and song covers that scroll in real-time. Similarly, the user can control where the song is played in the progress bar below. In addition, if the user is interested in the music, they can click on the little red heart to mark it as a favorite song.

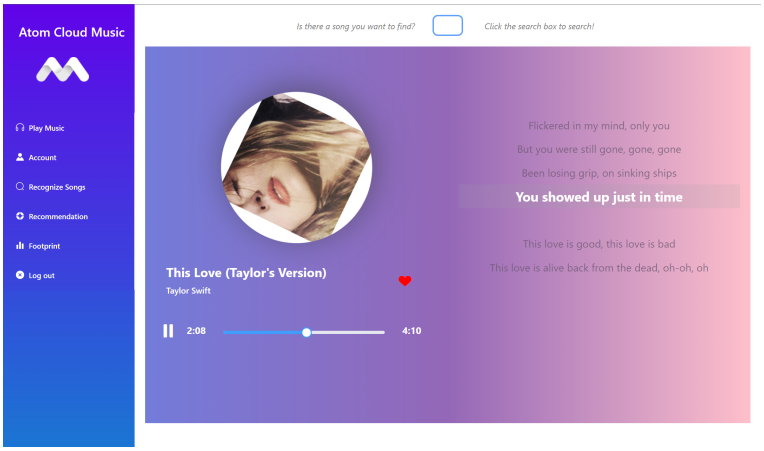


Figure 3: Music Detail Page

The system will record how long users listen to songs to ensure that users do not overindulge during a single use of the listening function to ensure the user's ear health. If the user listens to the song for a long time, the system displays a warning window. (**Note: The current system sets the reminding time as 30 minutes.**)[4]).

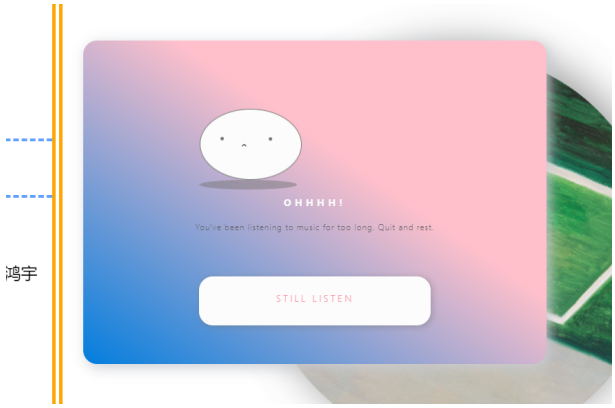


Figure 4: The Hearing Protection Window

- **User Information Page**

  The user information interface is the main interface of the system user management module. This is where you display all of the user's personal information, such as profile picture, user name, password, address, etc. Users can also edit the information themselves to improve the data. The page also adopts the card design idea, providing users with brief identity cards and details cards.

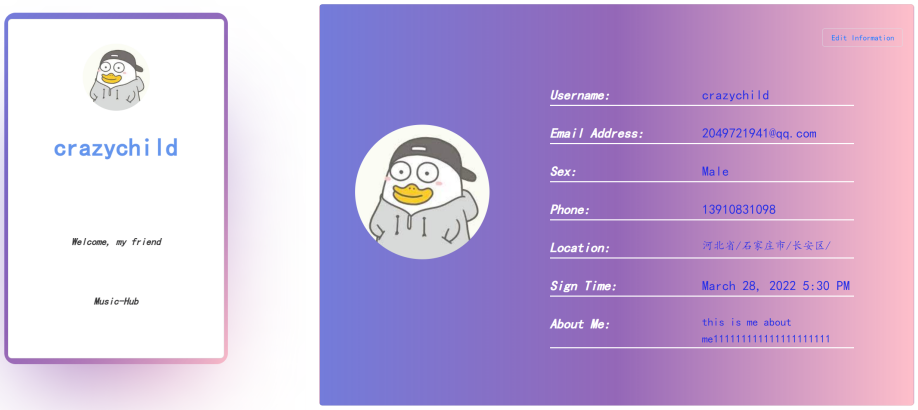


Figure 5: card(left) detail(right)

When the user information changes, the user can enter the editing interface to select the content to be edited according to the blocks and make changes.

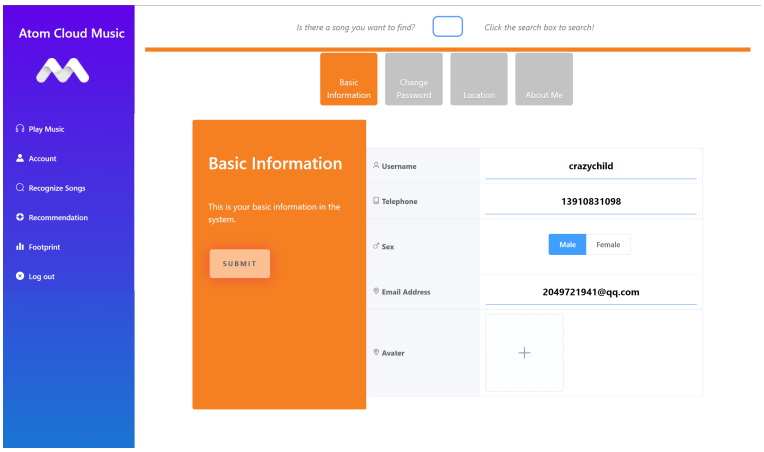


Figure 6: User Info edit

- **Recognize Page**

  In the song recognition interface, the system provides users with two ways to upload audio and record computer audio. Switch by clicking the Toggle button in the upper left corner of the page.

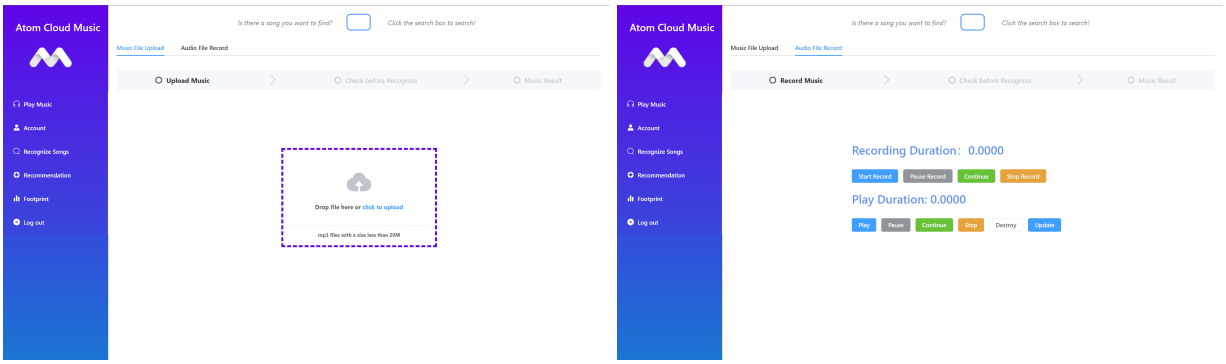


Figure 7: Recognize Page

4.3 The Back-end

4.3.1 Basic functions

Flask architecture is adopted in the back-end of the system. Data is transferred between Axios and the front end, and SQLAlchemy and SQLite are used to complete data processing and storage. The following is a brief introduction based on the functions.

1. **Login and Register**

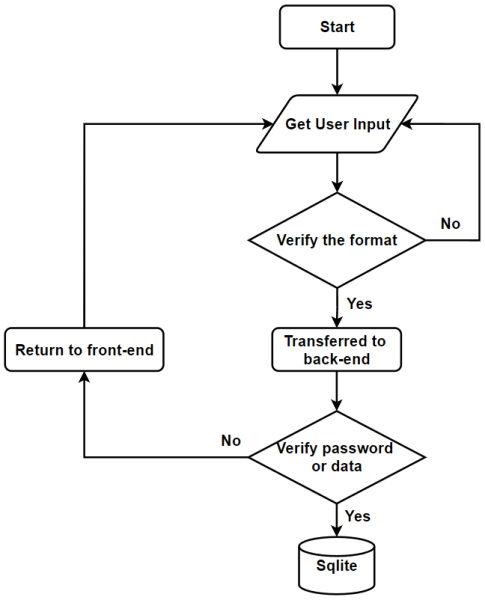


Figure 8: Flow of Register/Login

After the system receives the user input, the login and registration functions verify the basic data format and type in the front end, ensure that the user input meets the basic requirements, and report the results to the user. The information is then sent using the Axios front end to the back end, which selects a different route based on the login or registration chosen by the user, enters a different function execution, and returns the result.

Similarly, in the login function, after verifying the password, the system generates a login token for the user.

After login or registration is completed, the system redirects the user to the main screen.

2. **User Information Management**

User information system mainly records and displays user personal information data. This process especially involves two key technologies: data interaction and data display.

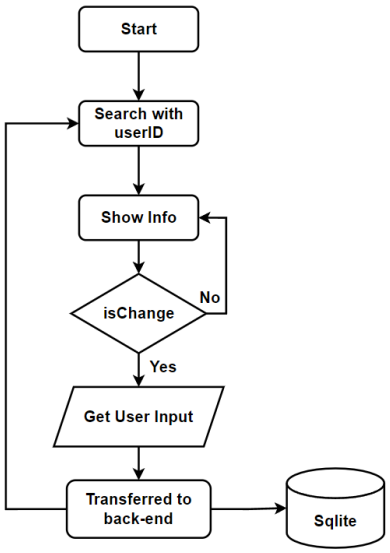


Figure 9: Information Management

As shown in the flowchart, the system obtains the user ID, queries the specific information about the user, and returns the information to the front end for display. When the user chooses to change the information,

the system takes the user's input, converts it to JSON format, and sends it to the back end. The back-end system retrieves the data and makes changes to the database.

1	id	INTEGER
2	username	VARCHAR (64)
3	email	VARCHAR (120)
4	password_hash	VARCHAR (128)
5	sex	VARCHAR (64)
6	phone	INTEGER
7	location	VARCHAR (64)
8	about_me	TEXT
9	create_time	DATETIME
10	avatar	VARCHAR
11	last_login_time	DATETIME
12	accumulate_duration	int

Figure 10: User Table

3. **User Data Virtualization**

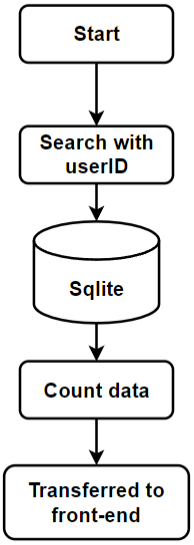


Figure 11: Data Virtualization

Different from the simple operation of adding, deleting, changing, and checking the basic information of users in the database, data visualization requires the collation of user history records and other data and the combination of "Vue-chartjs" and "echarts" plug-ins to achieve data visualization and display in the form of graphics. In this project, we use bar charts, pie charts and broken lines to represent the "historical ranking of listening to songs", "ratio of favorite songs to the number of listening to songs" and "Daily trend of listening to songs" respectively.

4. **Music Show**

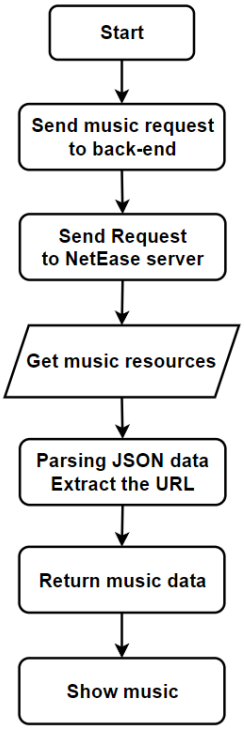


Figure 12: Get Music Flow

The music display function relies on NetEase Cloud music open-source API. during system page initialization, requests for a song list from the NetEase cloud Music server are created. The OBTAINED JSON data is analyzed in the background and sorted into a list containing URLs, song names, images, and other content, which is returned to the front-end for loading and display.

The most important part of this process is using the Requests function to send requests to third-party servers and collate JSON data.

4.3.2 Music Recognition

The music recognition feature is mainly implemented by the Shazam algorithm.

4.3.2.1 Why Choose Shazam?

There are many music recognition algorithms, and finally, we chose shazam. The core reason is that its implementation is simple, and it has a low number of false positives while having a high recognition rate.

4.3.2.2 Algorithm Details

Shazam has three main components: Robust Constellations, Fast Combinatorial Hashing, and Searching.

1. Robust Constellations: The first thing we need to know is that the core of song recognition is to convert the sound signal into a spectrogram and compare its content. However, in order to address the problem of robust identification in the presence of highly significant noise and distortion, the spectrogram peaks are selected. Thus, a complicated spectrogram, as illustrated in Figure 13 may be reduced to a sparse set of coordinates, as illustrated in Figure 14.

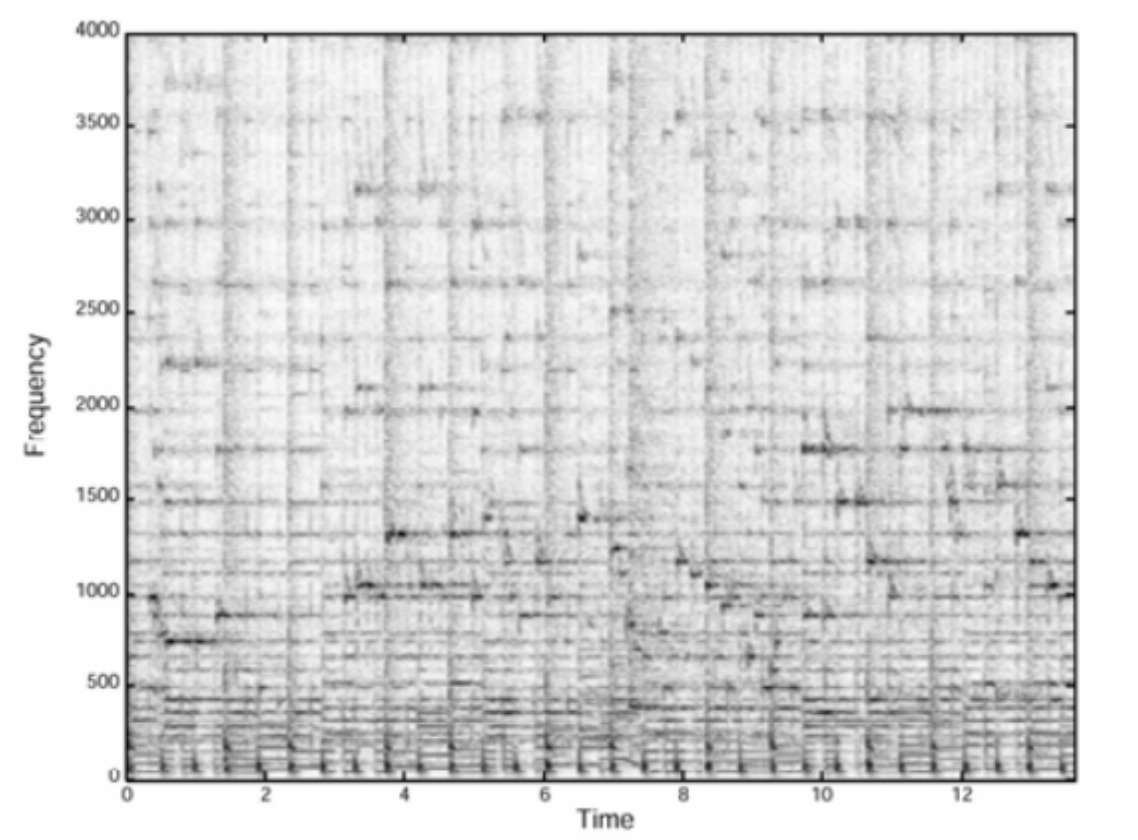


Figure 13: Spectrogram

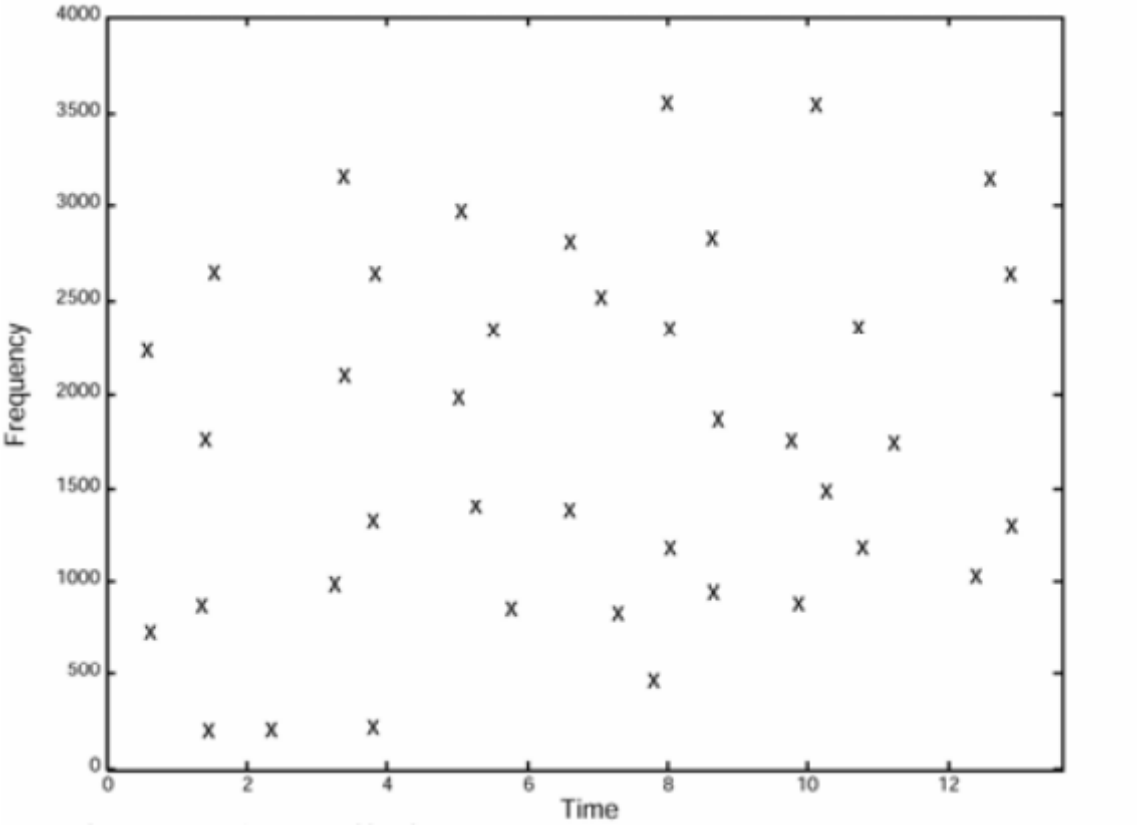


Figure 14: Constellation

The sparse coordinate lists are termed “constellation maps” since the coordinate scatter plots often resemble

a starfield.

2. Fast Combinatorial Hashing: Finding the correct registration offset directly from constellation maps can be relatively slow, so, next, we need Fingerprint hashes. The fingerprint hashes are formed from the constellation map, in which pairs of time-frequency points are combinatorially associated.

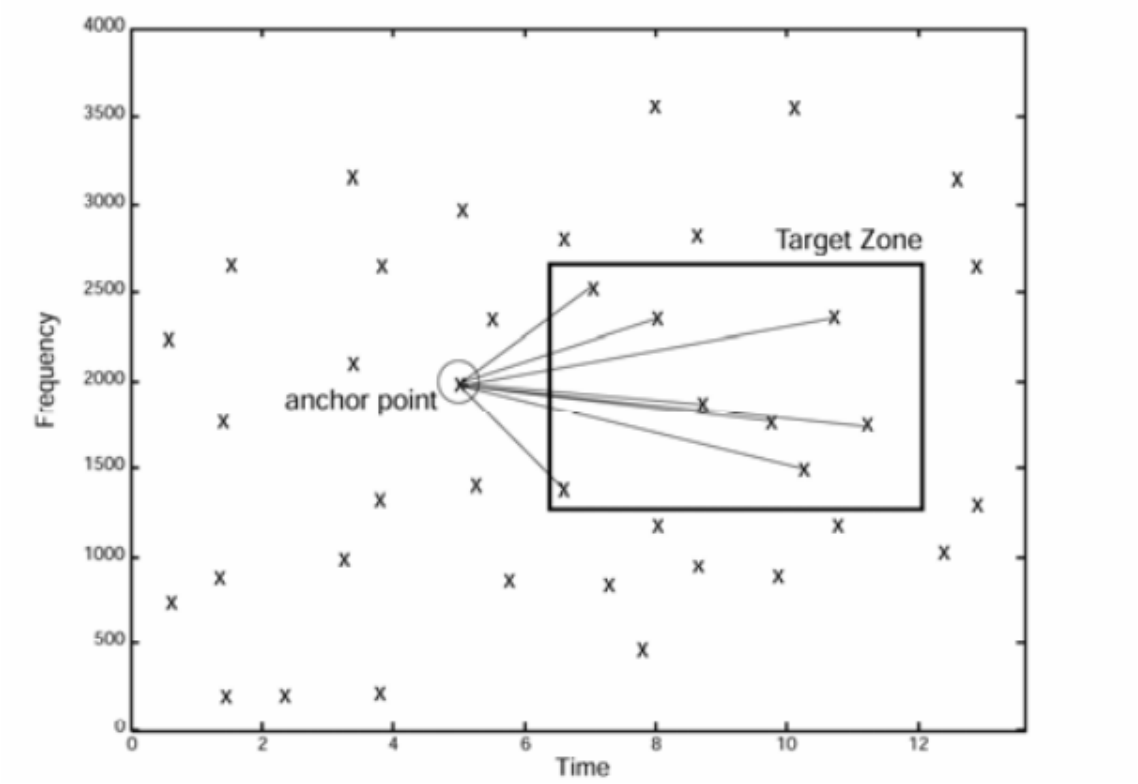


Figure 15: Generate Fingerprints

As shown in Figure 15, anchor points are chosen, each anchor point having a target zone associated with it. Each anchor point is sequentially paired with points within its target zone, each pair yielding two frequency components plus the time difference between the points. Then we get a fingerprint.

3. Searching: Last, we need to get the fingerprints from the sample music file before we take searching. The music in the database that produces the most matches with the sample fingerprints results from the recognition.

4. Multi-threading: Because step 2 and step 3 can be calculated in parallel, we applied multi-threading to speed up these processes. Finally, we were pleasantly surprised to find that the whole process of music recognition can be completed in less than a minute.

4.3.3 Similar Music Recommendation

Unlike the traditional recommendation system, our function does not require user reviews. The recommendation process is based on the inherent features of music, such as frequency, which can be considered a kind of unsupervised learning.

4.3.3.1 Basic idea of implementation

1. Prepare a large number of songs as our dataset.
2. Convert these songs from MP3 format into image format.
3. Use a pre-trained CNN (Convolutional Neural Network) model to extract feature vectors from images.
4. Store these feature vectors into a npy file.
5. When there is a song required for the recommendation, we first extract the feature vector of the song through steps similar to previous steps.

6. Then, we calculate the cosine similarity between the given song and each song in our prepared dataset.
7. Finally, we return 4 top songs after ranking.

4.3.3.2 Implementation Techniques

- **The conversion of audio file to image file**

  There is a cross-platform (Windows, Linux, MacOS X, etc.) command-line utility called SoX that can convert different formats of computer audio files into other formats. We use this tool to implement the conversion of an audio file to an image file.

- **Feature extraction**

  For this process, we applied a pre-trained model found on the website. The following image illustrates the structure of our CNN model.

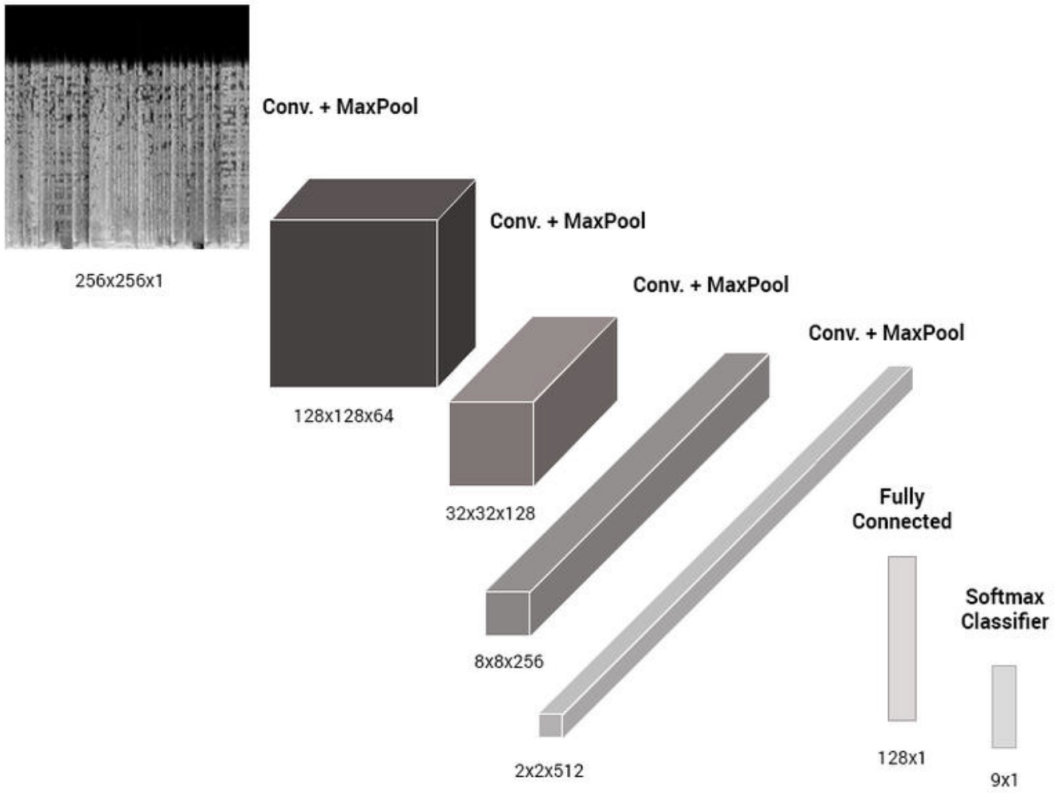


Figure 16: CNN Model Structure

You can see this is a complete CNN model structure, but we don't need the softmax classifier since the classification task is not required in our function. After the fully connected layer, a vector of 128 x 1 size will be output, and it can be considered the representation of the song features.

In most situations, the length of song is different, to solve this problem we just slice the long spectrogram to multiple small squares and extract feature vectors of each square. Then we calculate the average vector among multiple vectors, and consider the average vector as the final feature vector.

The model has been trained previously based on 1000 songs, so the parameters in the layers have been tuned well.

As for the deep learning framework, we use TensorFlow, which provides an easy Python API with supports GPU acceleration ability. The detailed steps of using TensorFlow are:

1. Download the CNN model, which is a file in HDF5 format(Hierarchical Data Format).
2. Load the model into memory.
3. Read image files into memory using image processing API in Keras library.
4. Run the model to get the feature vector.

We can store the vectors on disk in npy or pkl format to prepare the song dataset. They can be easily read by NumPy API when needed.

- **Similarity calculation**

  Next, we calculate the cosine similarity between the given song and each song in our dataset. In this process, we use sklearn library since there is a convenient API - cosine_similarity().

  Then we rank the songs in descent order and select 4 top songs as the most similar songs returned to users.

4.4 Implementation of non-functional requirements

4.4.1 Compatibility

The adaptive front-end development ensures the compatibility of multiple devices while maintaining user-friendly and beautiful pages. An adaptive website is also called a responsive website, and adaptive is a technology that means the web page can automatically determine whether the device is a computer or a mobile phone and display the appropriate screen size of the website. We use the adaptive framework Bootstrap for development to ensure the consistency of the page template and the integrity of the project. At the same time, the adaptive site has a better user experience and is easier to maintain.

4.4.2 Capacity

When the capacity is insufficient, only the popular music can be uploaded preferentially. For minority users, their preferences cannot be determined. Since the application of larger server capacity, it can upload more majority and minority music while reducing the long tail effect.

4.4.3 Security

When a user wants to start interacting with the API, it must authenticate with a username and password to get a temporary token. As long as the Token is valid, the user can make an API request with the Token to obtain permission to visit each function. Once the Token expires, the system would automatically log out, and the user needs to request a new token to log back in.

5 Software Testing

5.1 Web Test Criteria

5.1.1 Web UI Test

	Complete interface display, no dislocation or data loss	**Can display normally in different devices**	**The interface is easy to read, and the user can quickly become familiar with the operation**	**Page can provide user feedback correctly**
Score (1-10)	10	7	10	9

5.1.2 Web Function Test

	User can complete account registration and login	**System can provide recommended music and play it according to user data**	**System can search correctly based on user input and return actionable results**
Score (1-10)	10	8	10
	System can correctly record and display user information	**System can correctly display user information and usage data**	System can accurately identify the songs uploaded by users and return reasonable results
Score (1-10)	10	9	8

5.1.3 Web Offline Test

1. When a Website is deployed to a server, it should maintain the same functionality and interface as it does locally;
2. When a user is accessing, the response time should not exceed 5s;
3. It can be accessed anywhere in the world and at any time;
4. Not losing data in the cloud;

6 Trouble Shooting

1. **Error:** The back end could not start correctly after deploying the project on the server

   Solution: Adding a new location component in the nginx configuration to listen the port of 8082, which is for the back end

2. **Inefficiency:** In deep learning approach, it took long time to extract feature vectors of prepared songs, normally a couple of hours.

   Solution: We use cloud computing approach to accelerate this process. More specifically, we use CUDA on GPU server to perform matrix multiplication in parallel, so the elapsed time of passing through convolution layers, pooling layers, etc in the CNN model will be reduced a lot.

3. **Error:** The server capacity did not have sufficient space to store thousands of music

   Solution: Sending request to UCD CStech for larger capacity

4. **Error:** After logged in, the users could not know their ID, so the users could not access their personal account information.

   Solution: Use JWT to implement the function of token, instead of create Token from the back end and store into the database.

7 Conclusion

7.1 Epilogue

At this point in the text, all should be finished. This project ends at the end of the semester. Looking back on the whole development process, from the initial requirement, the completion of the primary function on the deadline, to the completion of the overall function of the project today, our team has paid too much for it. I would like to express my gratitude to every member of the group **(Chunyu Li, Yunkai Li, Haotian Li, Zhongqi Kang, Xuhui Shi)** for the excellent cooperation and the spirit of duty and responsibility during the whole semester. The spirit of each person has become the premise and foundation for our team to create miracles. In addition, our teaching assistant, **Lan Wei**, has also provided great help and support for our project. All the team members would like to express their deep gratitude and respect to the teaching assistant. The teaching assistant's selfless help is like a beacon on the sea, pointing out the direction of development for our team and urging on the progress. Also, our team would like to thank **the teachers of this course and any students who have helped with this project**. Thank you all for contributing to any part of the project.

This project is a cloud music-playing website that aims to realize users' browsing and sharing functions through the Internet and Web platform. Although there is a very mature technology, the development process is still a big challenge for our team. We encountered various problems in cooperation and technology throughout the process, but we all solved them through patience and exploration.

Learning to solve problems and learn new skills or attitudes from issues may be the purpose of this project development.

Finally, if you have any questions about this project, don't hesitate to get in touch with the team at Group 4: Atom or email us.

7.2 Project Prospect

Due to the limitation of project development time and other reasons, the system has only completed the main functions and collected and recorded a large amount of user data for further system function expansion. In future improvement and development, the system will further develop data information based on main functions and provide users with a further music ecosystem and communication system.

In the case of sufficient time, we will add the following functions to the system in the subsequent development:

- Matching mechanism and chat function for users with the same music preferences;
- A more accurate recommendation system based on user listening records;
- Music - based video playback;
- More music collections are associated with user listening records;

The purpose of adding functions is to improve the social function and more personalized user experience of the song website, form a closed ecological loop, and build a platform for communication and sharing based on music.