

System Design Document

1.Introduction

1.1 purpose

The goal of this project is to develop a practical and easy-to-use music recommendation system. Users can quickly and accurately find songs related to their favorite songs that may be of interest. According to the design requirements of the recommended system, this document provides an overall framework and design direction, and also defines some requirements of the system for users to confirm the function and performance of the system.

1.2 background

In today's information explosion, it is difficult for people to find what they want from the huge amount of data on their own things. In the field of music, how to quickly and accurately find the songs that you may be interested in is a necessary research direction, which can improve the efficiency, provide more entertainment space for people, and also facilitate the managers to organize and manage the relevant resources and information reasonably and efficiently.

2. project overview

2.1 objectives

There are two user stories, as follows:

User story 1: I'm a user of the recommendation system, I already have some favorite songs, I want the system to push `recommend me some songs that I might be interested in.

User story 2: I'm a user of the recommendation system. I've created some song lists. I want more song list, hoping the system can recommend some other people's song list to me.

The design goal of this music recommendation system is: for ordinary users who like music or professional music producer recommend more songs or lists that they might be interested in.

2.2 product functions

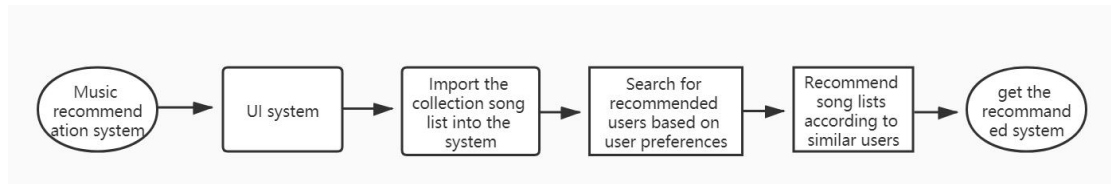
Users first need to log in to the system, and then the user will record and collect the songs on a certain music platform.

Import songs or song lists into the system. The system will search for other users with similar preference style according to the user's preference information, and recommend some songs or song lists of these users to the user. Finally, the user can get some songs or song lists that he may be interested in from the system.

Through the system users can achieve the following functions:

- A. Users register the system
- B. Users login the system
- C. Users logout the system
- D. Users change the information of the own account.
- E. Users import the song list
- F. Users get the recommended song list.

2.3 flow chart of product system



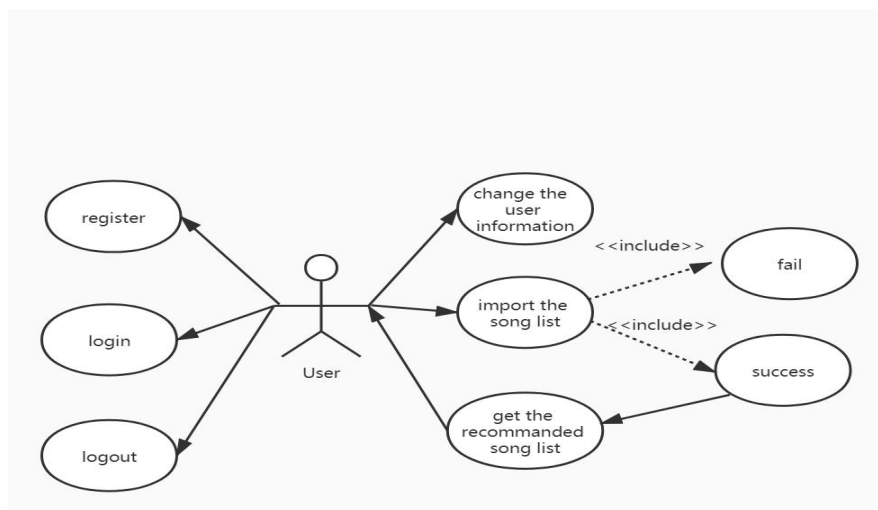
2.4 user characteristics

The service group of this system is all ordinary users who love music and professional users such as music producers,

Ordinary users can search for songs or song lists they are interested in, and music producers can search for music works of the same style according to their creative styles.

3. Specific task requirements

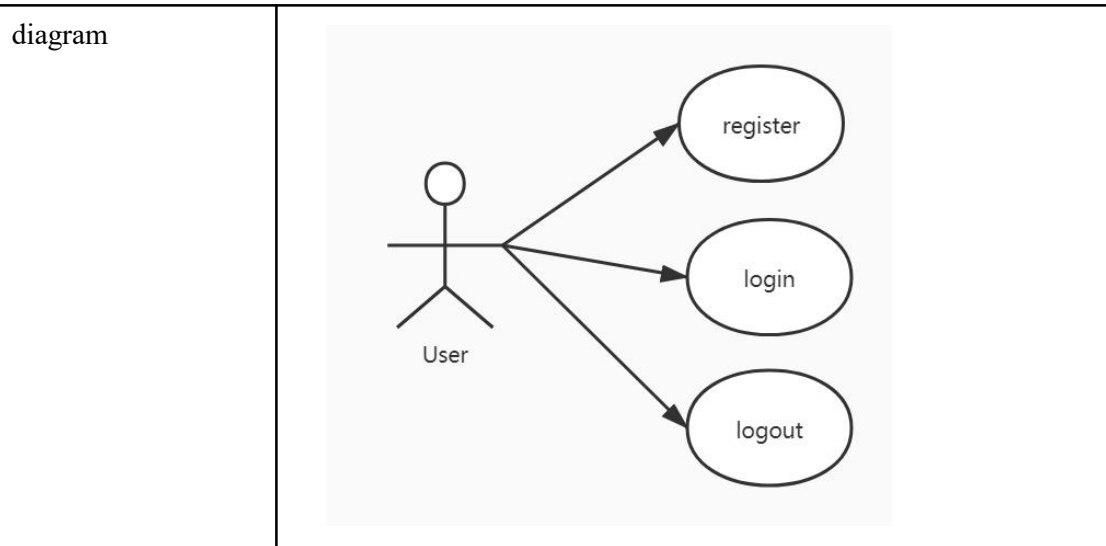
3.1 functional requirements



Use case detail

1.

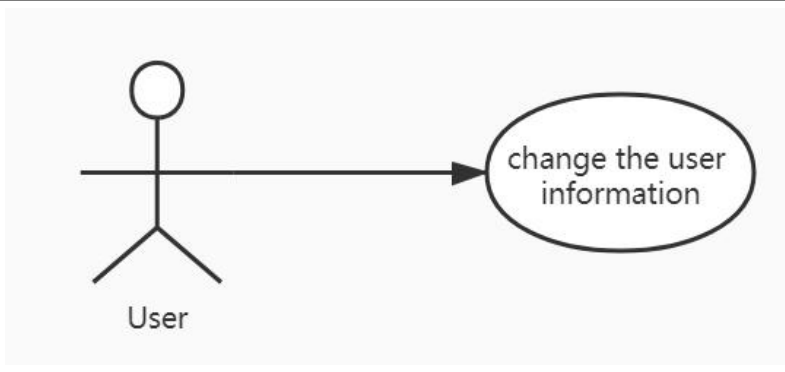
Use Case	Case1: login ,logout,register
goal	User can login ,logout and register the system form the UI
detail	Users can click the registration, login and logout buttons on the user interface to perform these operations. The premise of login and logout is to register an account.
actor	User



Basic flow	actor	system
1.	Users click to register	
2.	The user enters the necessary information into the system	
3.		The system sends the verification code to the mailbox.
4.	The user fills in the verification code	
5.		Display successful registration information
1.	User click to login	
2.	User input account and password	
3.		System confirmation information, such as correct display login success
1.	User click to logout	
2.		The system displays a confirmation log out message
3.	The user clicks OK and logs out	

	successfully	
--	--------------	--

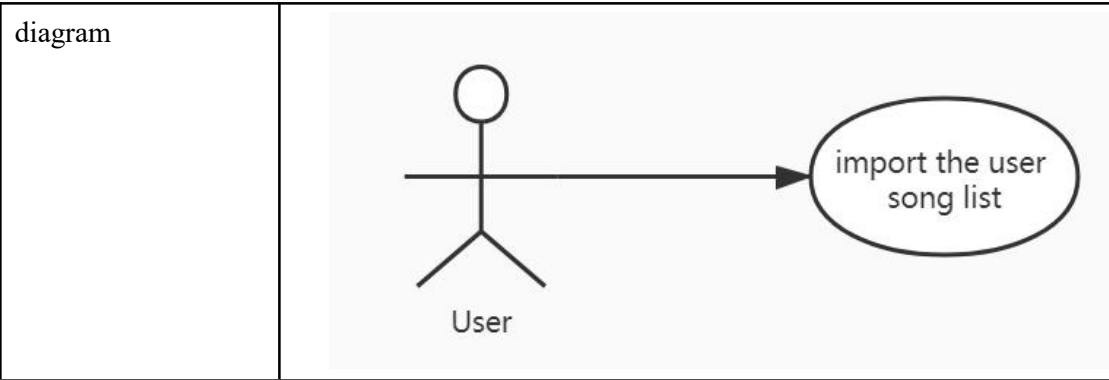
2.

Use Case	Case2: Users change account information
goal	The user can change the account information. For example, they want to change the password
detail	Users click the settings button on the user interface to change their personal information
actor	User
diagram	 <pre> graph LR User((User)) --> UC((change the user information)) </pre>

Basic flow	actor	system
1.	User click to change personal information	
2.		The system displays the original information
3.	The user enters the new information and clicks OK	

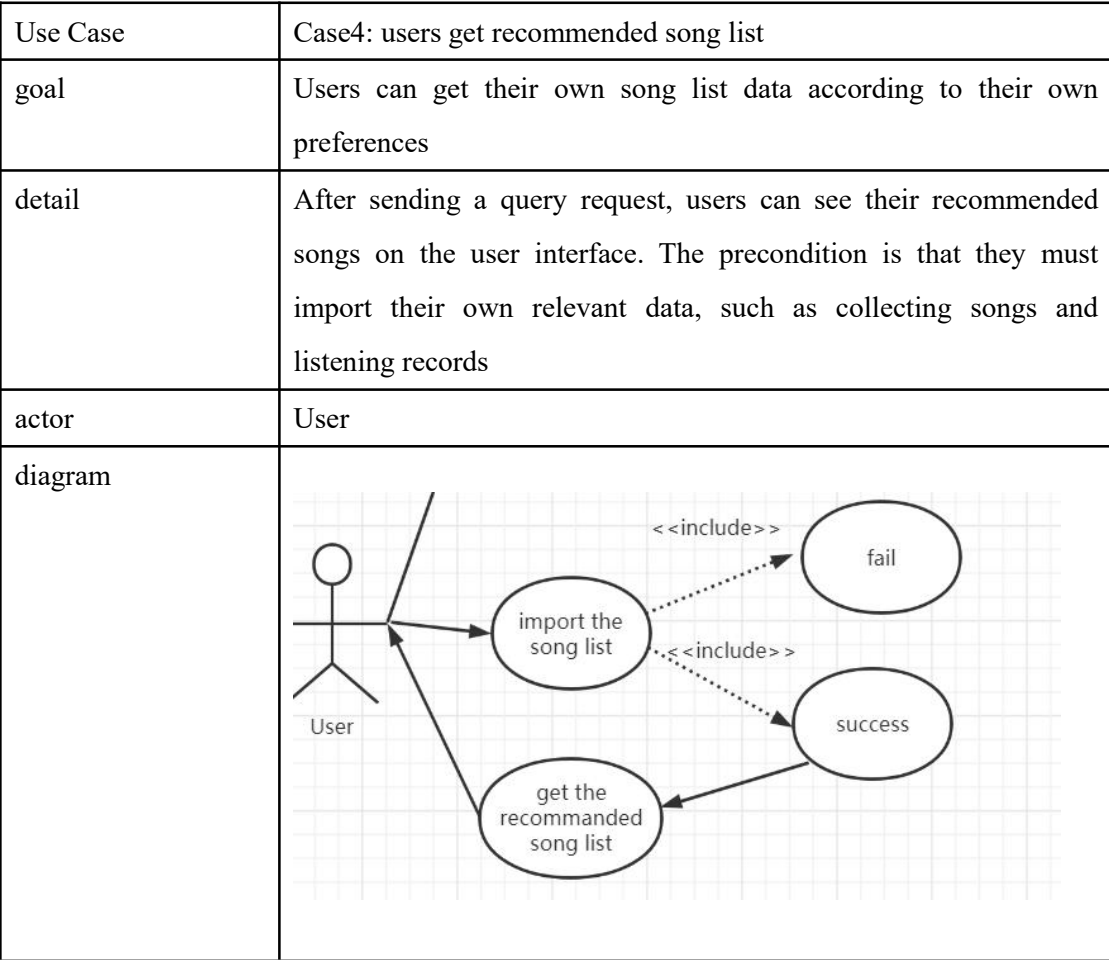
3.

Use Case	Case3: user importing song list data
goal	Users can import their own collection of songs and listen to songs
detail	The user can click the phase import record button on the user interface to import data
actor	User



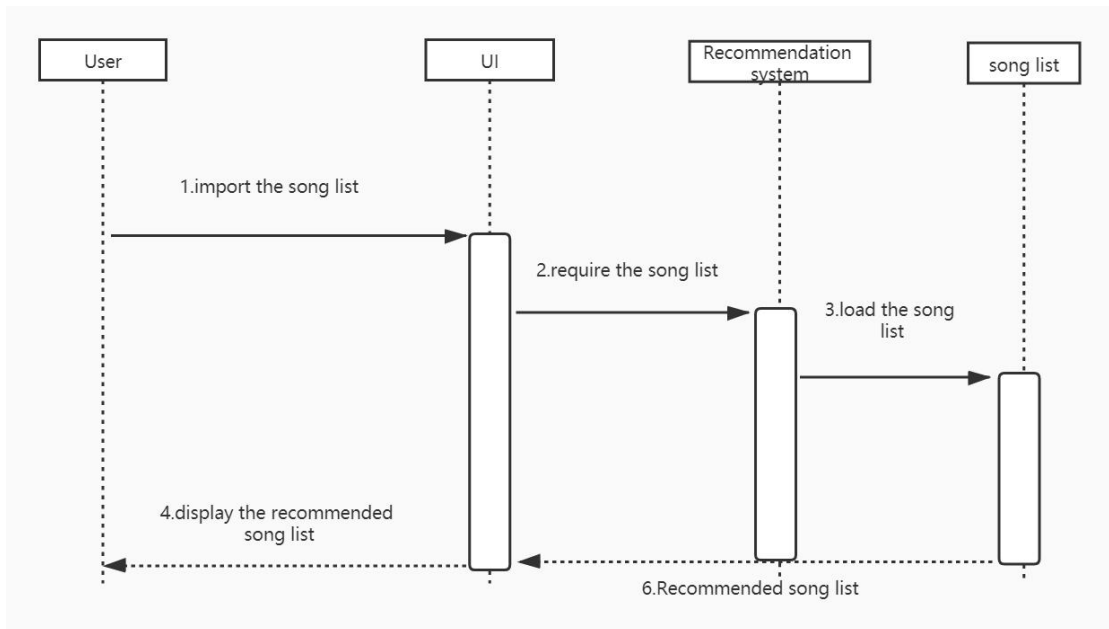
Basic flow	actor	system
1.	User clicks to import	
2.		The system can import local records and other data

4.

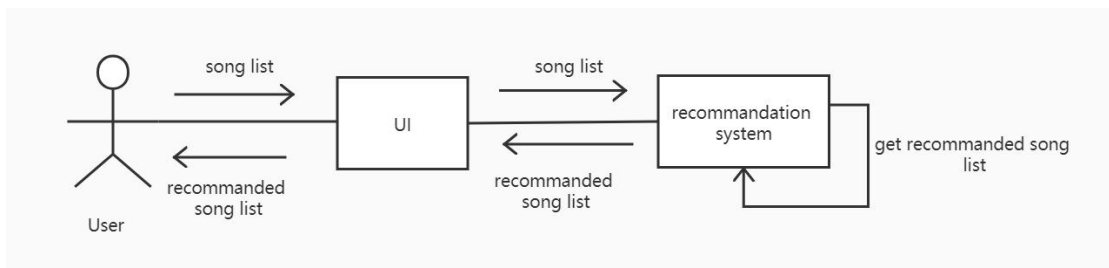


Basic flow	actor	system
1.	User click to import song list	
2.		System import recommended song list
3.		The system obtains the recommended song list through the algorithm
4.		The recommended song list is displayed on the interface

UML Sequence diagram



UML collaboration diagram



3.2 reliability requirements

For functions with high frequency and high criticality, such as login, the system shall be fault free and low response time. For functions with high resource consumption, such as query of

interested songs and song lists, it should be able to meet the concurrent operation of a large number of users.

3.3 operation environment requirements

The system plans to deploy the back-end service and database on the server temporarily, and the client is extended by browser

The system should meet the requirements of cross platform operation.

3.4 user interface requirements

The user interface is simple, efficient and beautiful. The user first enters the login interface and then enters the system.

In the main interface, you can enter the account management interface to manage account information in the main interface, and you can also enter the import interface in the main interface to import music preferences. There should be a query panel in the main interface to display the songs or song lists that users are interested in.