

基于 QQ 音乐的大数据分析与推荐平台
(Big data analysis and recommendation platform based on QQ Music)

需求规格说明书
(System Requirements Specifications)

编 撰 人：全组人员

审 核 人：邱亢迪

文档版本：V1

[illegible]

目录

基于 QQ 音乐的大数据分析与推荐平台	1
需求规格说明书	1
(System Requirements Specifications)	1
1. 引言	3
1.1. 目的	3
1.2. 文档约定	4
1.3. 预期的读者和阅读建议	4
1.4. 产品的范围	5
1.5. 分析依据	5
2. 功能需求	5
2.1. 用户管理	6
2.2. 大数据展示	17
2.3. 歌单排行推荐	21
3. 接口需求	31
3.1. 【SRS-012】内部接口	31
4. 其他需求	32
4.1. 需求	32
4.2. 约束	33
5. 项目要求	34
5.1. 用户界面	34
5.2. 运行环境	35
5.3. 用户文档	35
5.4. 设计和实现上的限制	35
5.5. 假设和依赖	35

1. 引言

1.1. 目的

音乐是最令人愉悦的艺术, 音乐是人类的创造, 自然与人有着密不可分的联系, 音乐在人类文明的进程中, 以其特有的方式在社会发展、个体发展以及教育发展中发挥着十分重要的作用, 它以人的生存意识为中心, 反映人的本质, 展示人的心态, 塑造人的品格, 更是影响着人的长远发展。因此本项目基于 QQ 音乐, 进行大数据分析和音乐推荐, 让音乐融入日常生活。

现如今, 人们的生活已经离不开音乐, 但当前市面上主流的音乐平台各种功能让人眼花缭乱, 初次接触的用户不可避免会感到繁琐和迷茫, 2022 年, QQ 音乐在各在线音乐软件的使用时长占第四位, QQ 音乐作为在线音乐软件的三大巨头之一, 月活跃人数已达 23000 万人。因此本项目基于 QQ 音乐, 进行大数据分析和音乐推荐, 针对不同的用户群体提供相应的歌曲歌单推荐服务, 实现用户科学管理歌曲和歌单。

1.1.1. 直观展示数据

当前音乐创作层出不穷，在大量的音乐面前，人们无法方便地获取当前音乐的数据，基于 QQ 音乐的大数据分析和音乐推荐系统（以下简称 QMABG），为直观展示当前歌手、歌曲、歌单、音乐风格、歌曲标签、歌曲评论量、最常一起出现的歌曲标签、评论所处的时段的排名以及评论中不同 IP 地址出现的次数，将所有数据采取可视化界面直观展示出来，让用户更加方便地了解当前乐坛数据，给予用户直观详细的信息展示。

1.1.2. 科学管理歌曲和歌单

目前用户在使用音乐软件时可管理的信息过多，而 QMABG 仅提供和歌曲歌单管理，将功能简洁化，真正成为一个易于上手，简洁而不繁琐的音乐平台。

1.1.3. 实现个性化歌曲推荐

当前主流的音乐软件因各种因素导致主页所展示的推荐信息无法真正迎合每个用户，而又缺少让用户选取自己感兴趣的歌曲类型进行个性化音乐定制的功能，因此 QMABG 将提供给用户个性化歌曲推荐服务，本系统通过两种方式进行个性化歌曲推荐，提升用户的使用体验。

1.2. 文档约定

该文档采用国家标准《计算机软件产品开发文件编制指南》。

下面详细介绍文件的编写目的与内容要求根据以上的调整，本文档按以下要求和约定进行书写：

页面的左边距为 2.5cm，右边距为 2.0cm，装订线靠左。

正文字体为宋体小四号，采用 1.5 倍行间距。无特殊情况下，字体颜色均采用黑色。

1.3. 预期的读者和阅读建议

本软件产品需求分析报告的预期读者包括：

普通用户、项目经理、开发人员、测试人员、文档编写人员、系统维护人员。

普通用户——从用户使用的便捷性和有效性对软件进行了解，软件的功能是否能满足日常工作的业务流程。

项目经理——从系统功能的角度对系统进行分析和建模，并拟定项目的开发周期，做好实

施项目的各种准备。

开发人员——对系统功能进行分析，系统功能是否能满足用户的要求，系统是否可行。

测试人员——从测试的角度对系统的功能进行分析，为以后系统的功能测试和集成测试等做准备。文档编写人员——作为开发人员和用户沟通的桥梁，站在开发人员的角度对问题进行描述，用通俗的语言描述给用户，方便用户的理解和沟通。

系统维护人员——利用此需求文档进行系统的维护工作。

1.4. 产品的范围

QMABG 从听歌人群、类别、地区、歌手热度等多维度分析对比数据，将最终得到的数据通过可视化展示出来，让数据更加直观可靠，同时，QMABG 将展现用户独特的音乐品味定制用户专属歌单，有效改善了因数据过多导致系统缺乏信息条理性和逻辑性，因功能过多导致初次使用的用户陷入迷茫的现状。

1.5. 分析依据

参考数据：

QQ 音乐歌曲播放量、QQ 音乐歌单播放量、QQ 音乐标签之间的联系、QQ 音乐歌曲评论量、QQ 音乐歌手粉丝数、QQ 音乐用户发表评论的 IP 地址、QQ 音乐用户发表评论的时间段、QQ 音乐歌曲评论数

2. 功能需求

QMABG 系统包含用户管理、大数据展示和歌曲推荐三大部分的内容，具体实现包含以下部件以及功能：

功能点			
模块	功能	用例编号	测试方法
用户管理	用户注册	SRS-001	Test
	用户登录	SRS-002	Test
	用户信息管理	SRS-003	Test
	歌曲收藏	SRS-004	Test

大数据展示	排序结果可视化	SRS-005	Test
	视图更新	SRS-006	Test
歌单排行推荐	歌单筛选	SRS-007	Test
	歌曲筛选	SRS-008	Test
	歌单排行	SRS-009	Test
	歌曲排行	SRS-010	Test
	个性化推荐	SRS-011	Test
其他需求	内部接口需求	SRS-012	Analyze
	可靠性需求	SRS-013	Analyze
	可维护性需求	SRS-014	Analyze
	可测试性需求	SRS-015	Destination
	保障性需求	SRS-016	Analyze
	性能需求	SRS-017	Test

图 2-1 用例描述

2.1. 用户管理

注册与登录、用户信息管理、歌单收藏、歌曲收藏。

2.1.1. 【SRS-001】用户注册

描述：用户通过输入账户名和密码进行账号注册，注册成功后即可解锁该系统所有功能。

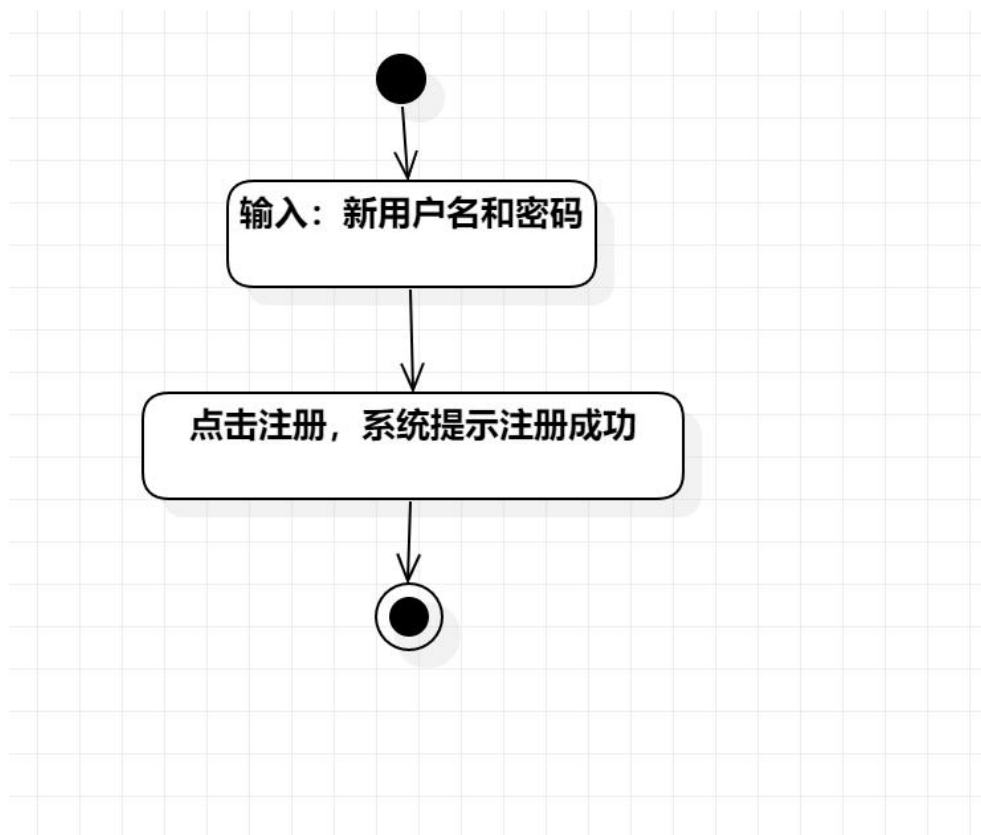
输入：账户名和密码

处理：

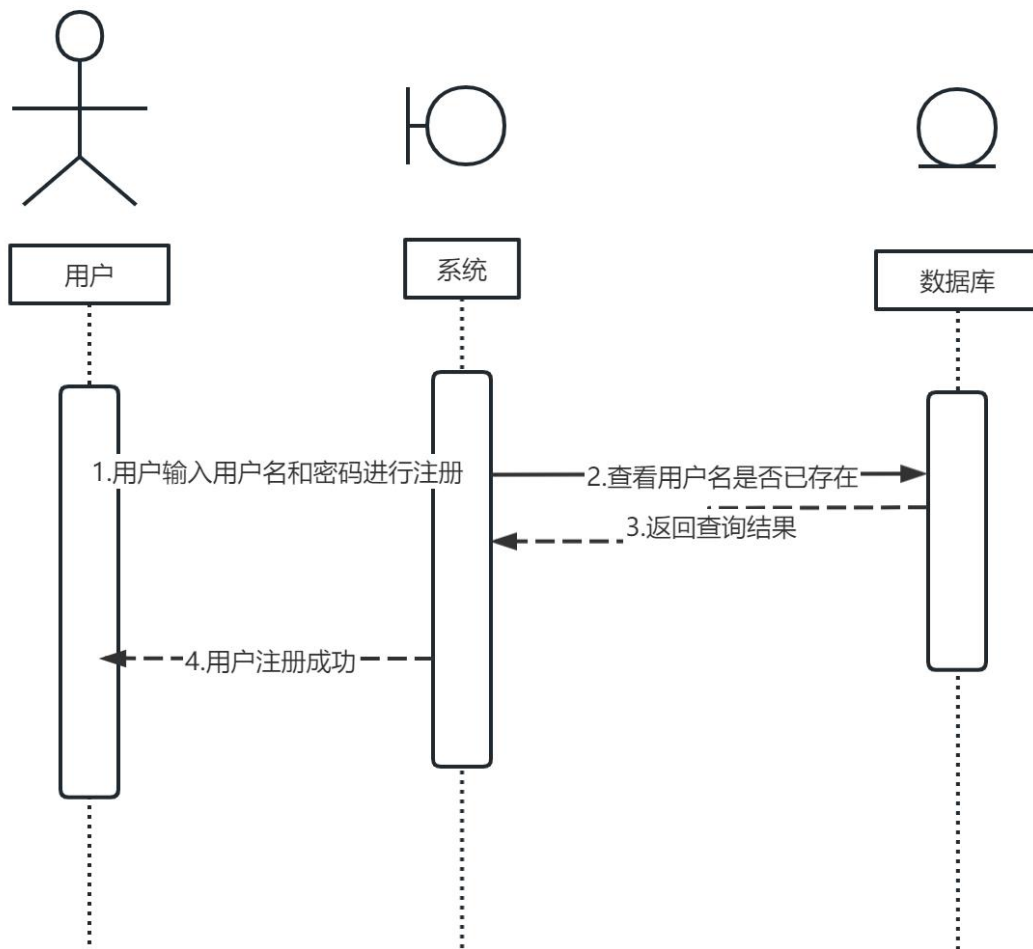
- 1) 用户输入要注册的账户名。
- 2) 用户输入要设置的密码。
- 3) 再次输入要设置的密码。
- 4) 点击提交完成注册

输出：用户注册成功。

活动图：



时序图：



2.1.2. 【SRS-002】用户登录

描述：用户通过输入账户名和密码，系统进行登录验证，若登录验证成功就可以进入个人页面，若登录验证失败则提示登录失败，需重新登录。

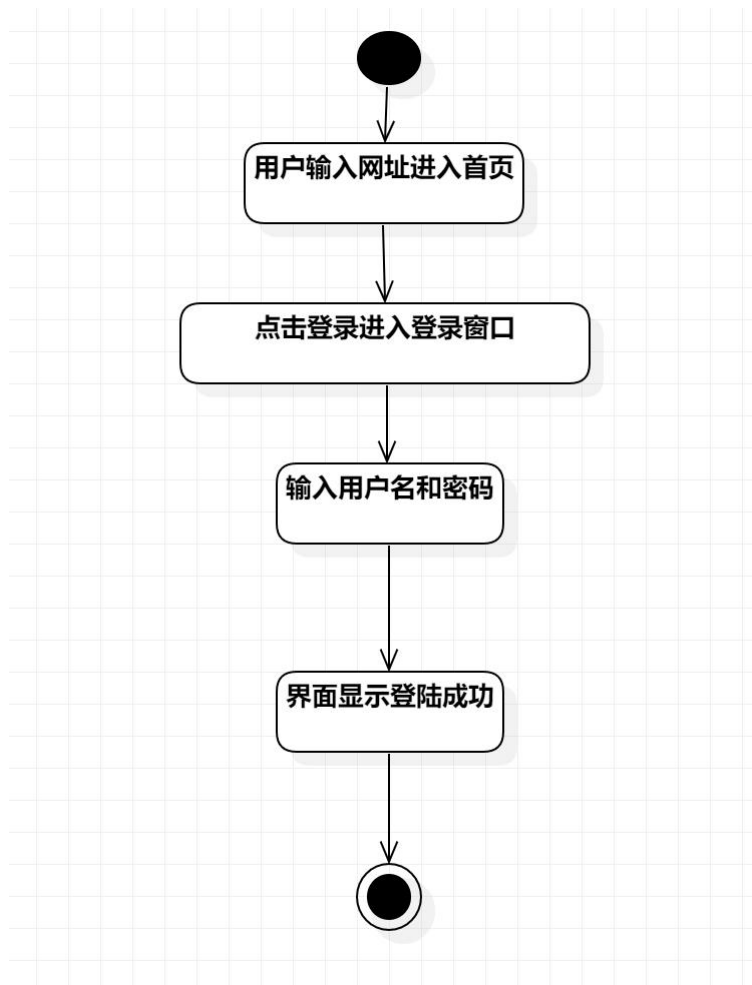
输入：账户名和密码

处理：

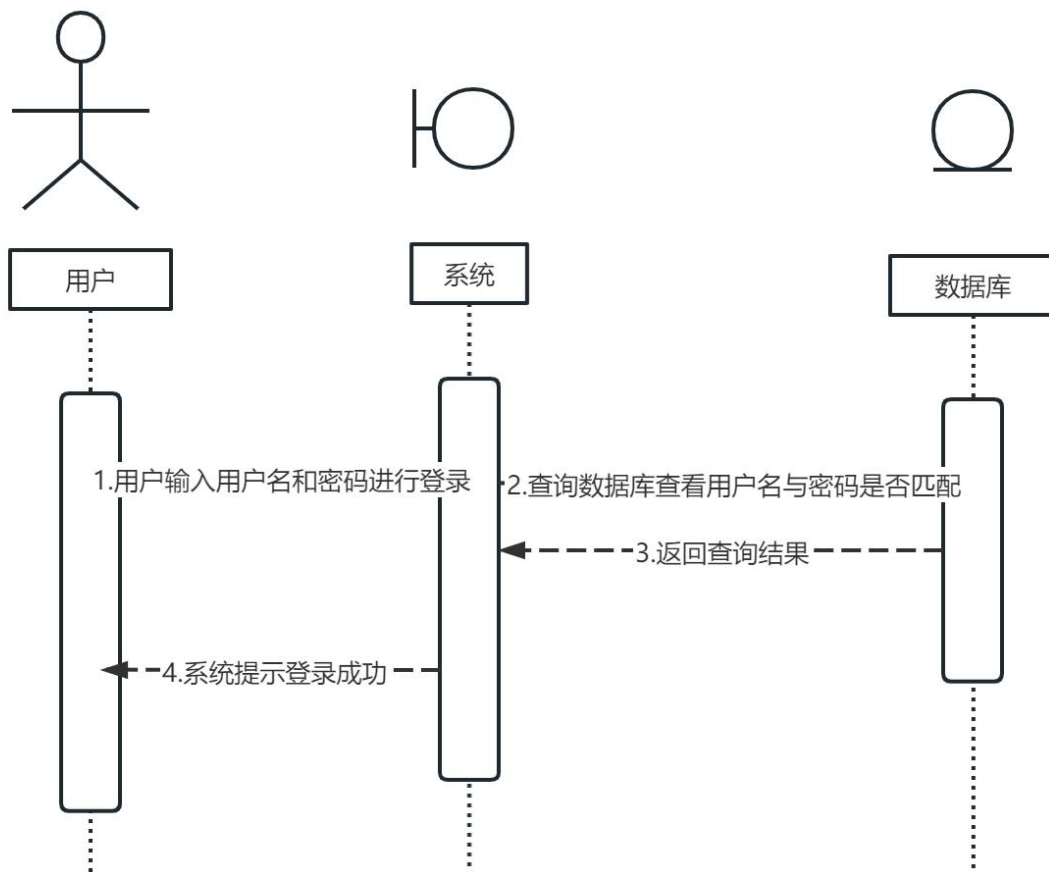
- 1) 用户输入账户名。
- 2) 用户输入账户名对应的密码。

输出：登陆成功，能够进入个人页面。

活动图：



时序图：



2.1.3. 【SRS-003】用户信息管理

描述：用户登录成功后可以进入个人信息管理界面进行昵称、性别、年龄和地址的修改和重置。

输入：用户自主选择修改编辑昵称、性别、年龄、地址。

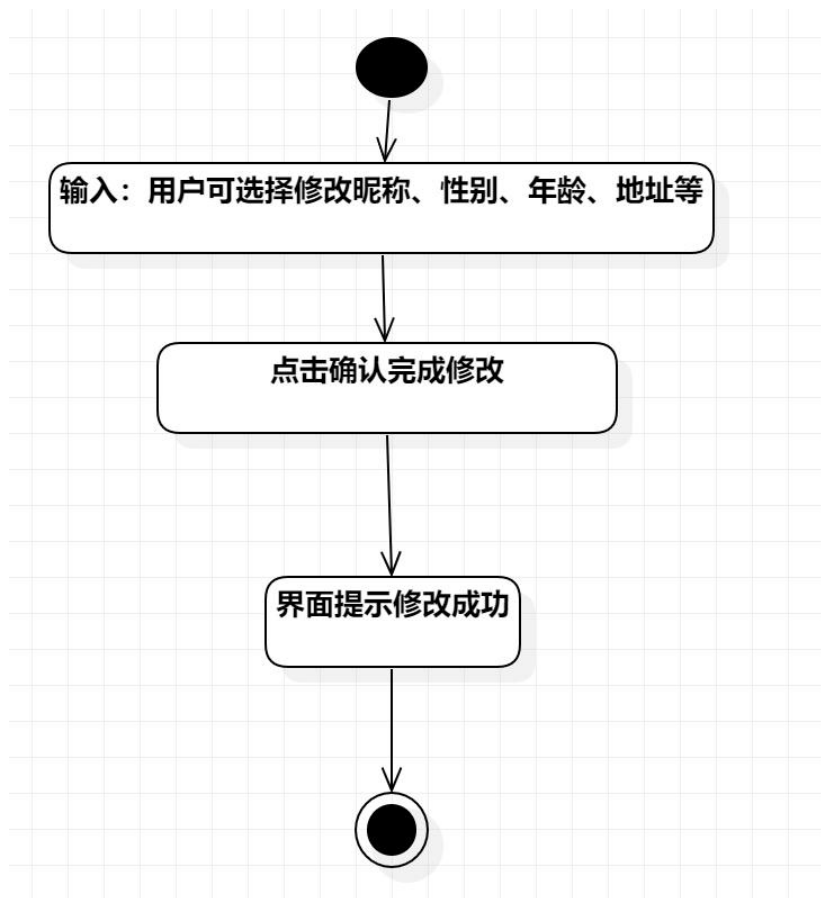
处理：

- 1) 用户可选择修改昵称
- 2) 用户可选择修改性别
- 3) 用户可选择修改年龄
- 4) 用户可选择修改地址
- 5) 点击确认完成修改

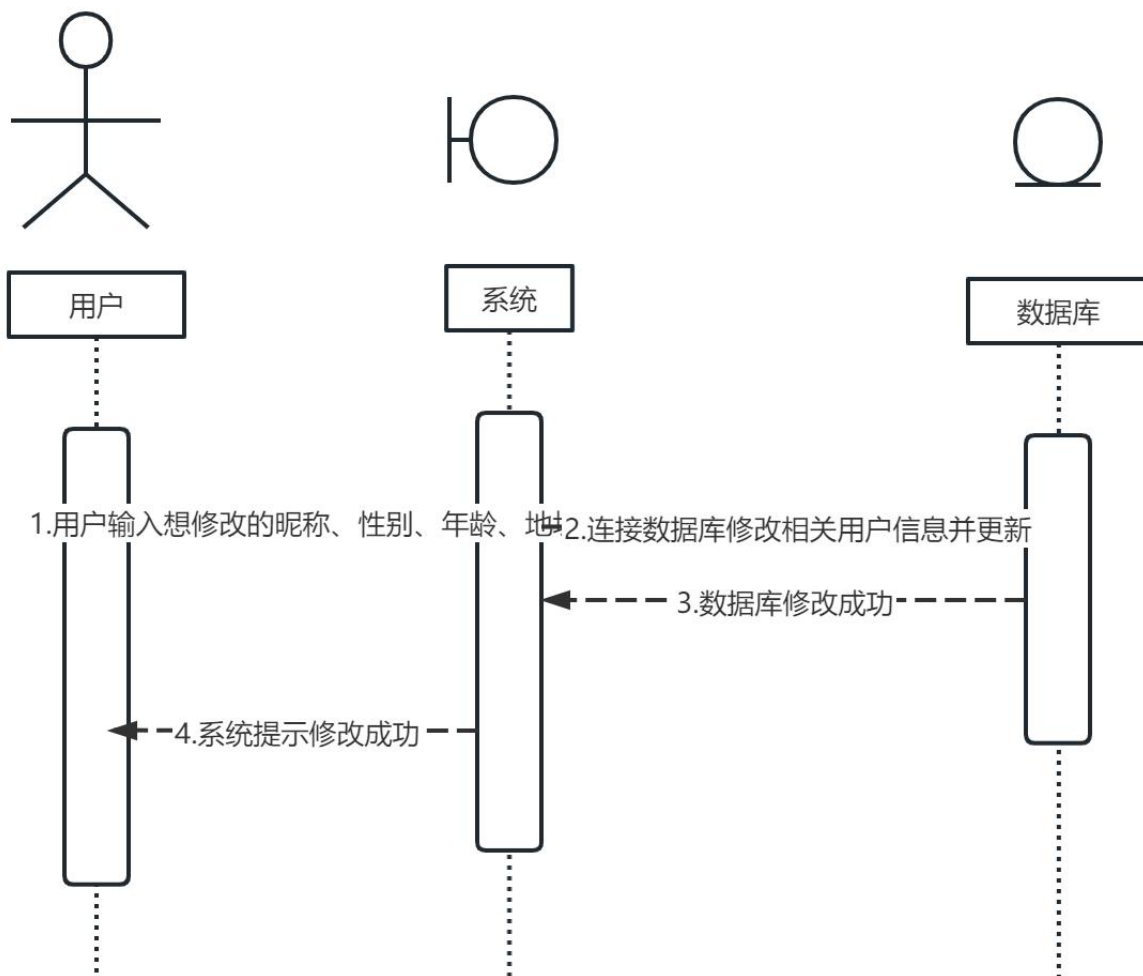
输出：

- 1) 用户未编辑前只有昵称，其余信息为空
- 2) 成功修改后变为修改后的个人信息
- 3) 可进行多次编辑

活动图：



时序图：



2.1.4. 【SRS-004】歌曲收藏

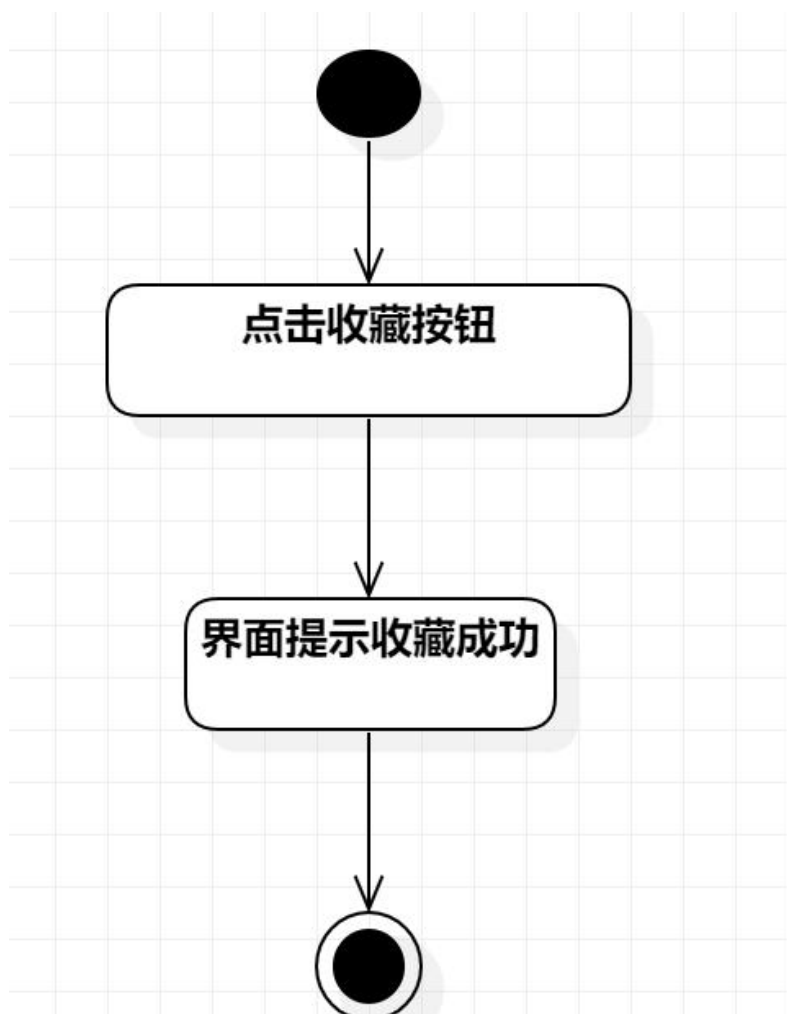
描述：歌曲收藏提供给用户收藏歌曲、取消收藏歌曲、点击跳转链接跳转到 QQ 音乐对应的歌曲界面、展示相关歌曲信息等功能

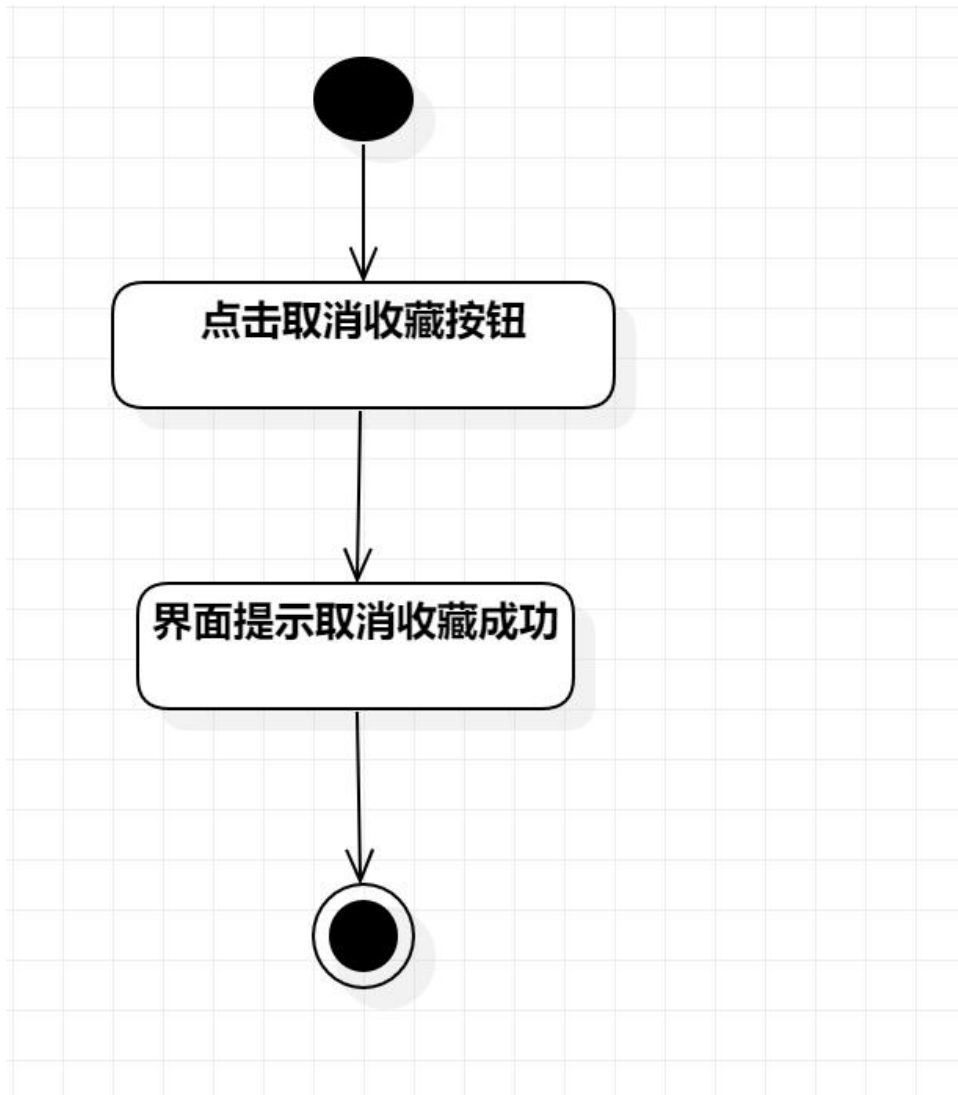
输入：用户通过点击完成收藏、取消收藏、跳转 QQ 音乐详情界面等操作。

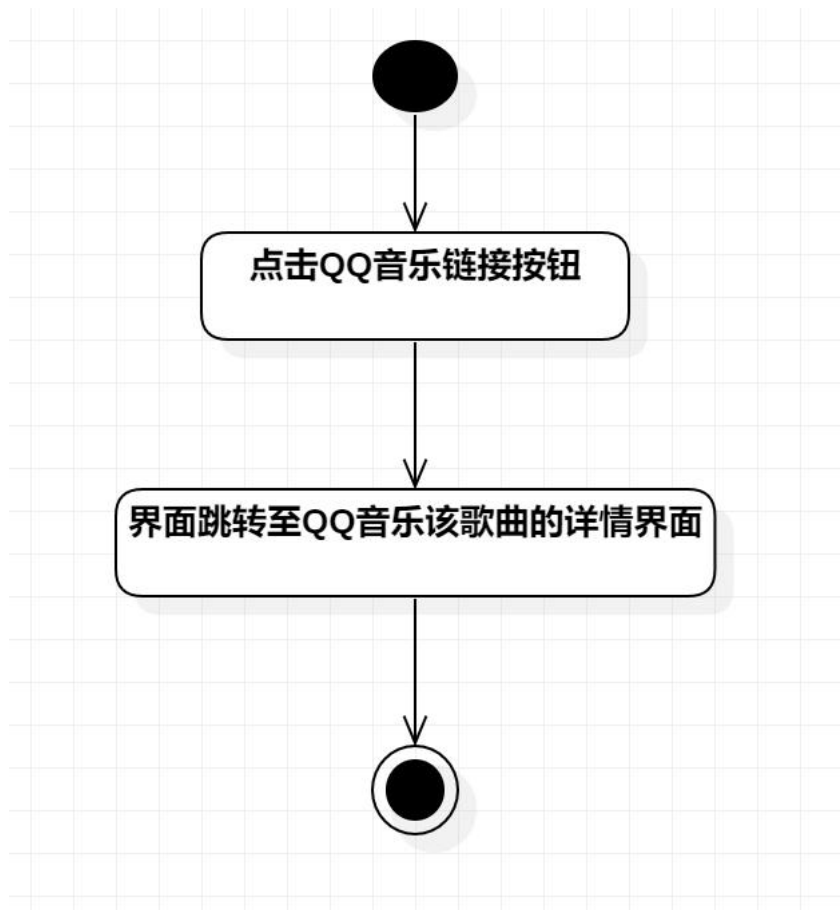
处理：

- 1) 用户点击收藏按钮完成收藏歌曲操作
- 2) 用户点击取消收藏按钮完成取消收藏操作

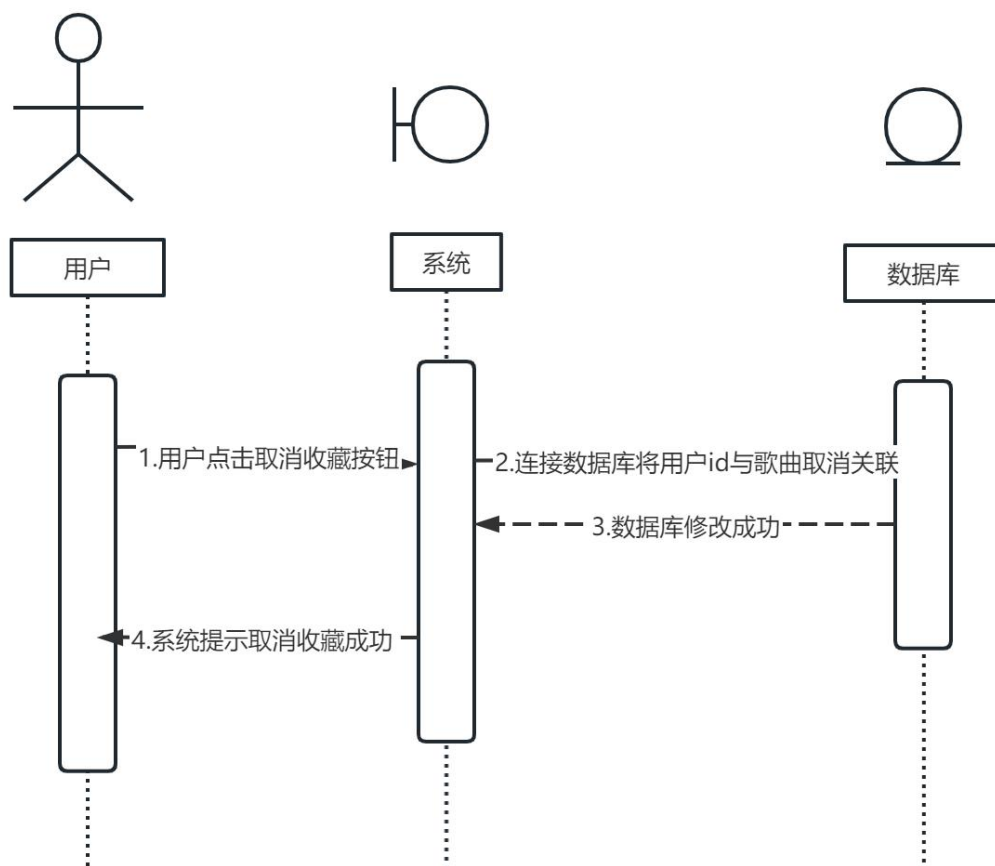
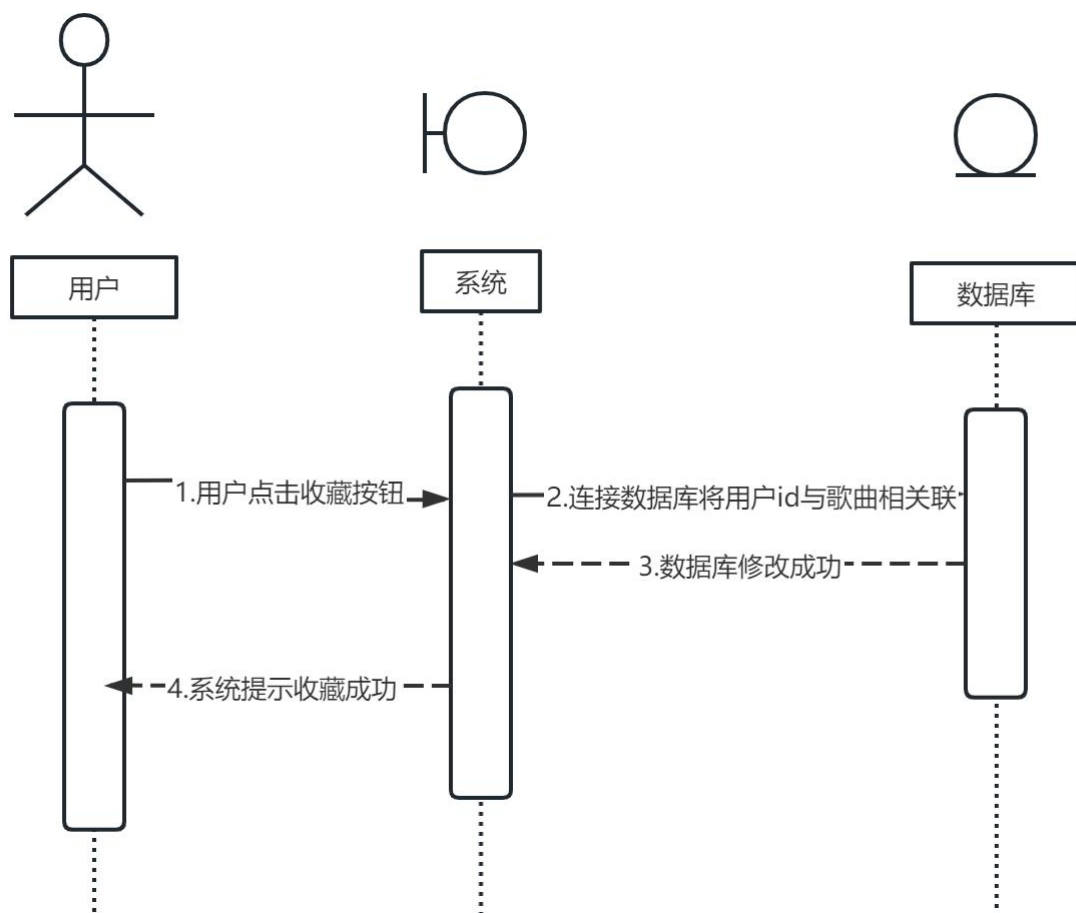
活动图：

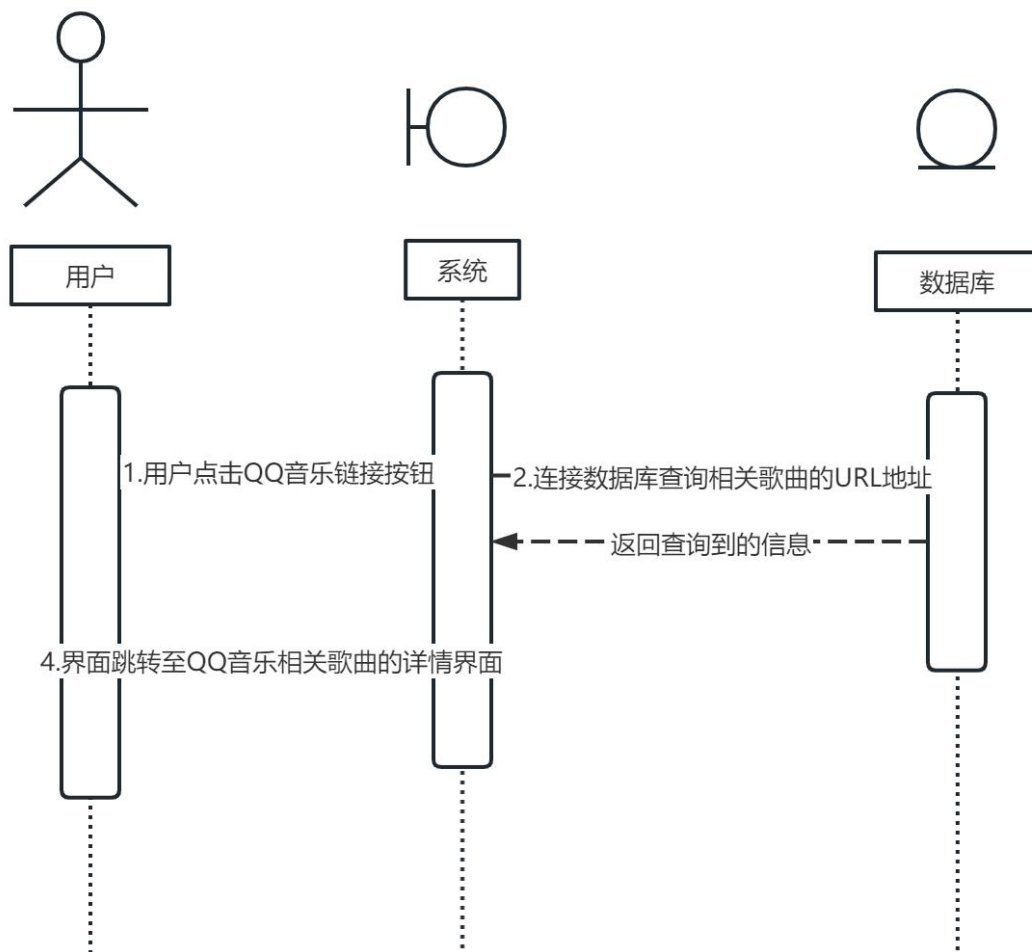






时序图:





2.2. 大数据展示

大数据展示包括按要求进行排序结果可视化、视图更新。

2.2.1. 【SRS】排序结果可视化

描述：要求进行排序结果可视化是平台通过播放量、收藏量等多方面考量对大量的歌单或歌曲进行排序，并最终以可视化图表等形式展示给平台用户，以便用户直观、准确的接收信息。

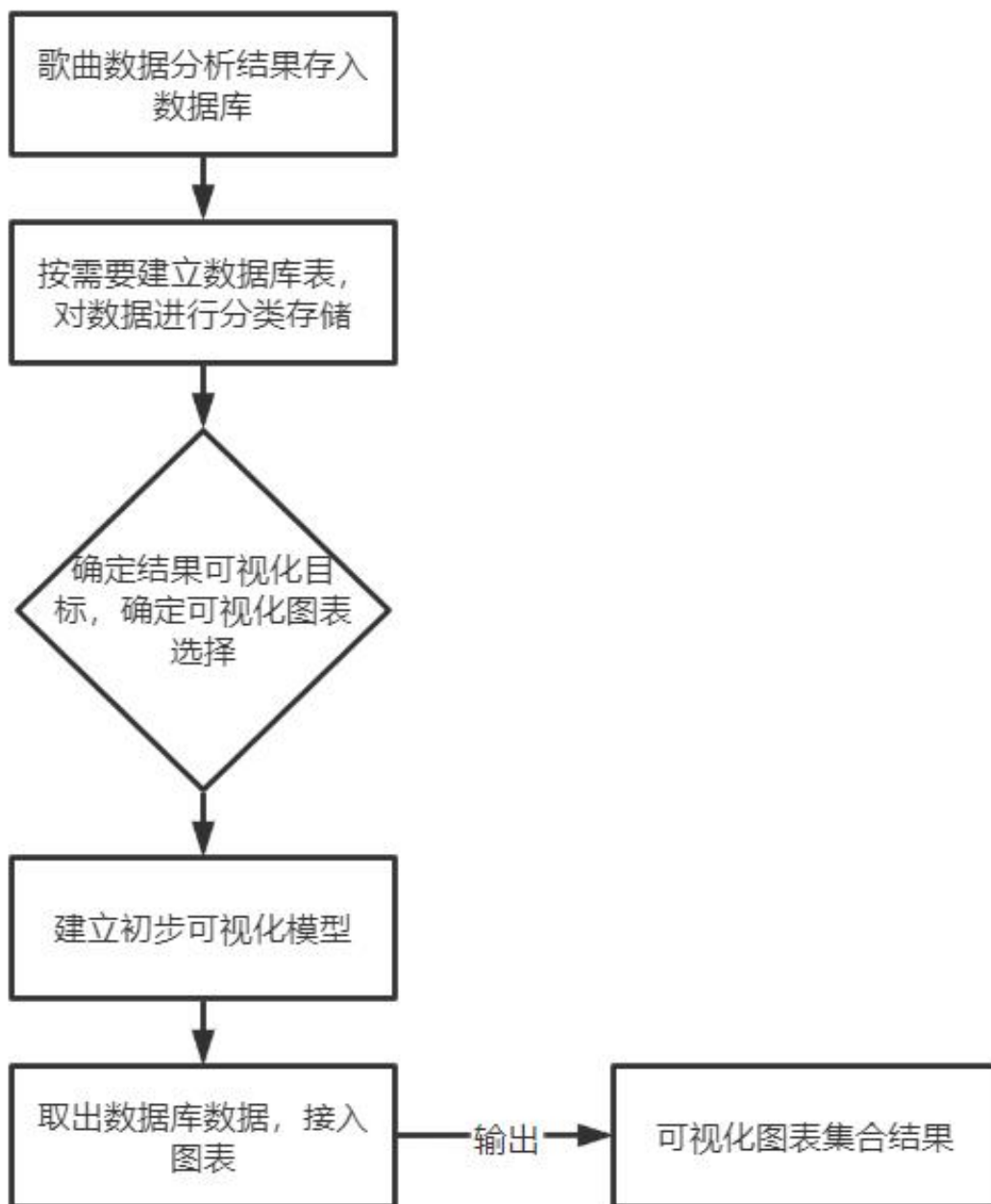
输入：大数据分析后的文件结果，包括歌曲收藏量，标签关联度，歌手热度等相关数据信息。

处理：

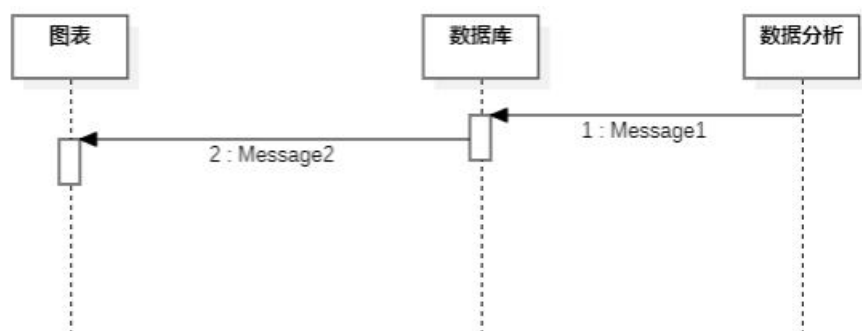
- 1) 将歌曲数据分析结果存入数据库，并按需要建立数据库表，对数据进行分类存储。
- 2) 确定结果可视化目标，确定可视化图表选择。
- 3) 调用 VUE 接口，选择合适图表，建立初步可视化模型
- 4) 取出数据库数据，接入图表

输出：可视化图表集合结果

活动图：



时序图：



2.2.2. 【SRS】视图更新

描述：视图更新是平台根据时间的变化，不断更新展示给用户的热门歌单以及热门歌曲，做到与时俱进，对数据信息不断进行处理。

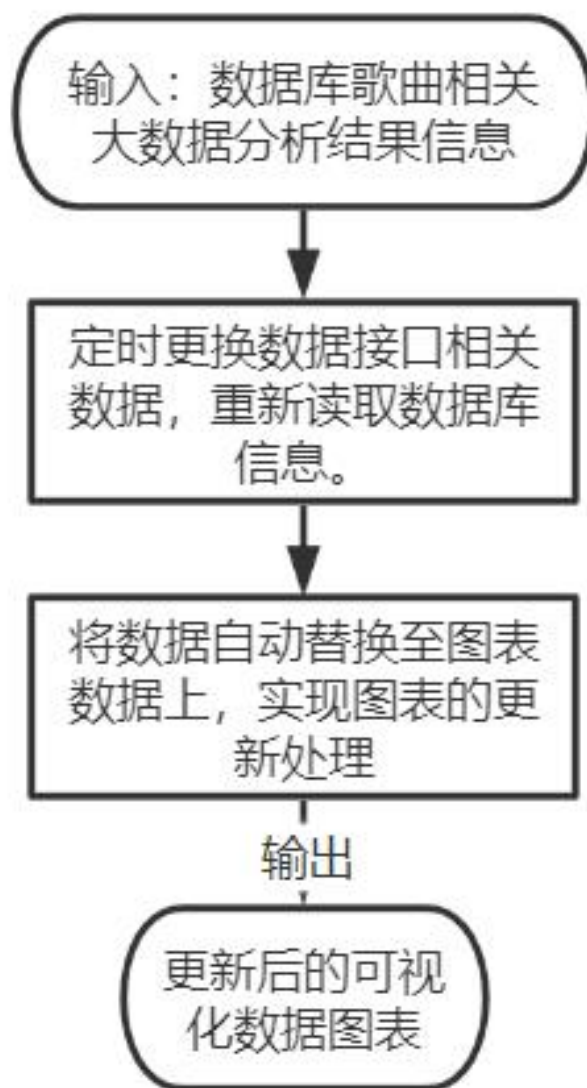
输入：数据库歌曲相关大数据分析结果信息

处理：

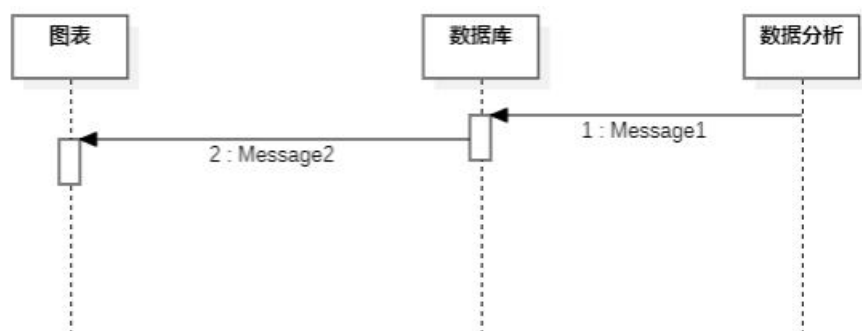
- 1) 定时更换数据接口相关数据，重新读取数据库信息。
- 2) 将数据自动替换至图表数据上，实现图表的更新处理

输出：更新后的可视化数据图表

活动图：



时序图：



2.3. 歌单排行推荐

歌单排行推荐功能模块主要包括推荐歌单筛选、歌曲筛选、歌曲排行、歌单排行、个性化推荐等功能模块。

2.3.1. 【SRS】歌单筛选

描述：歌单筛选是平台通过用户所在地、年龄、性别等信息通过大数据分析相似性对用户进行歌单筛选后向用户推荐歌单

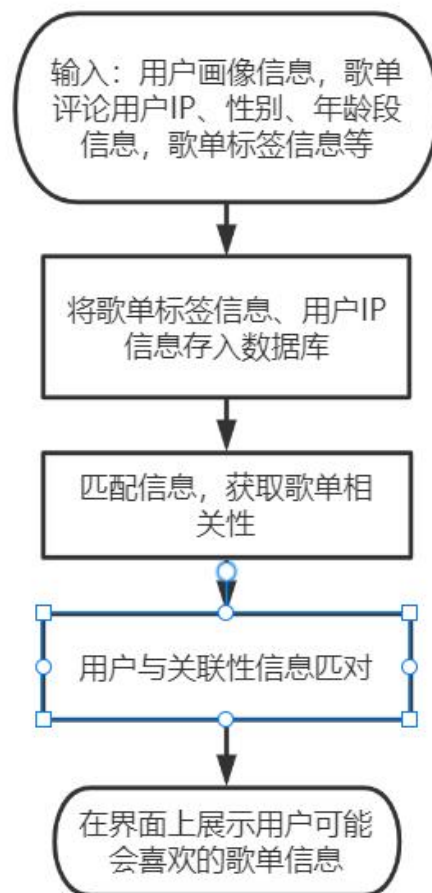
输入：用户画像信息，歌单评论用户 IP、性别、年龄段信息，歌单标签信息等

处理：

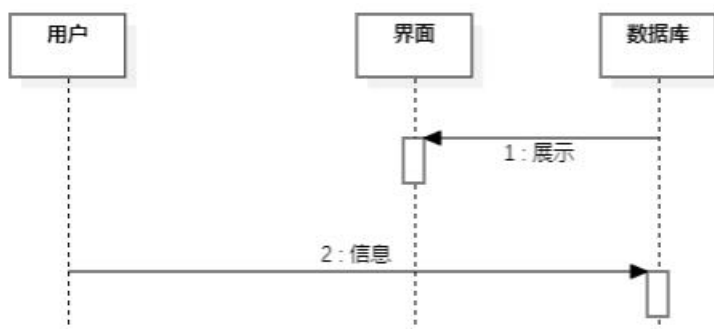
- 1) 将歌单标签信息、用户 IP 信息存入数据库
- 2) 将标签信息与用户 IP、年龄段、性别等信息进行关联分析，获得用户所在地区、用户年龄段、用户性别与歌单倾向的相关性
- 3) 获得用户画像信息，与数据库中的所存储的关联性信息进行比对
- 4) 将该类用户可能喜欢的歌单展现在用户推荐界面上

输出：在界面上展示用户可能会喜欢的歌单信息

活动图：



时序图：



2.3.2. 【SRS】歌曲筛选

描述：歌曲筛选是平台通过用户所在地、年龄、性别等信息通过大数据分析相似性对用户进行歌曲筛选后向用户推荐歌曲

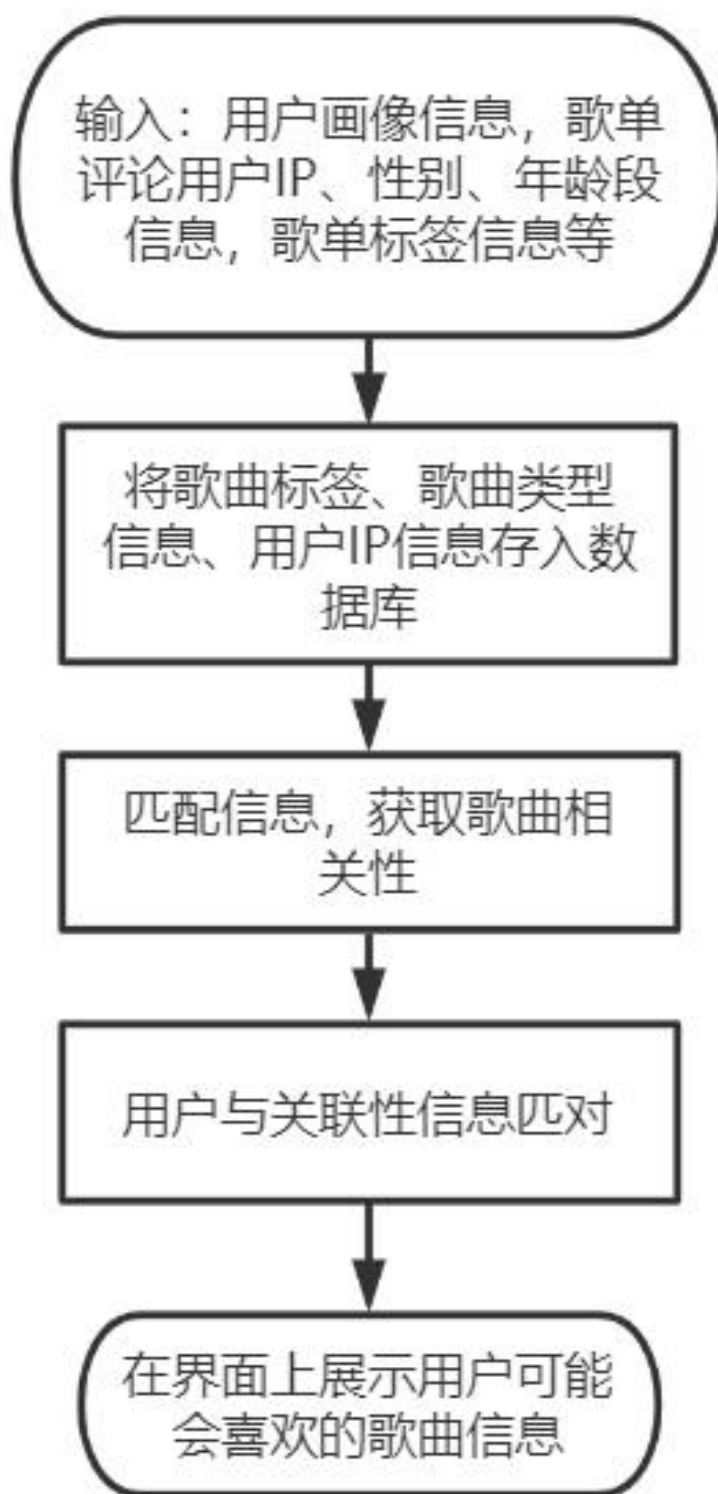
输入：用户画像信息，歌曲评论用户 IP、性别、年龄段信息，歌曲标签、类型信息等

处理：

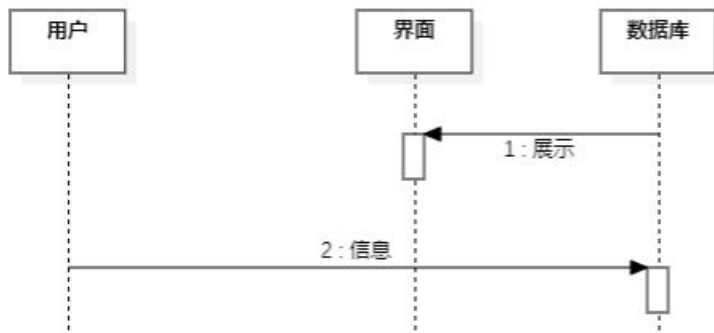
- 1) 将歌曲标签、歌曲类型信息、用户 IP 信息存入数据库
- 2) 将标签信息与用户 IP、年龄段、性别等信息进行关联分析，获得用户所在地区、用户年龄段、用户性别与歌曲倾向的相关性
- 3) 获得用户画像信息，与数据库中的所存储的关联性信息进行匹配
- 4) 将该类用户可能喜欢的歌曲展现在用户推荐界面上

输出：在界面上展示用户可能会喜欢的歌曲信息

活动图：



时序图：



2.3.3. 【SRS】歌单排行

描述：歌单排行是通过爬取 QQ 音乐热门歌单信息进行统计后，在主页面中展示热门歌单排行

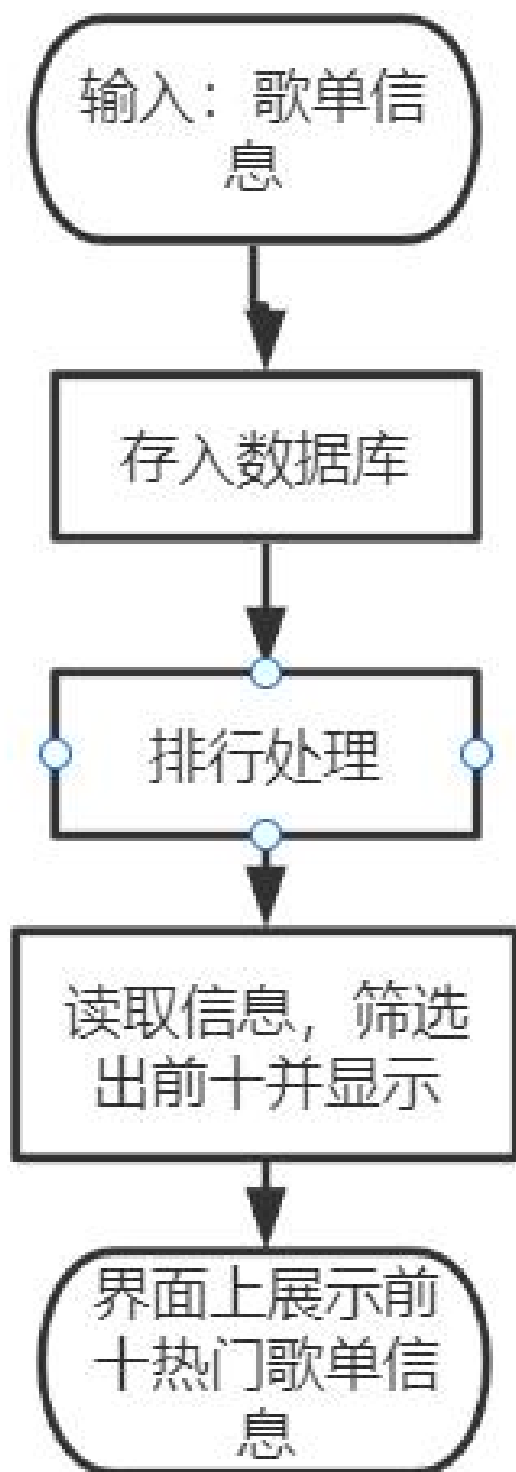
输入：歌单名称、歌单标签、歌单收藏量、歌单播放量、歌单评论数信息

处理：

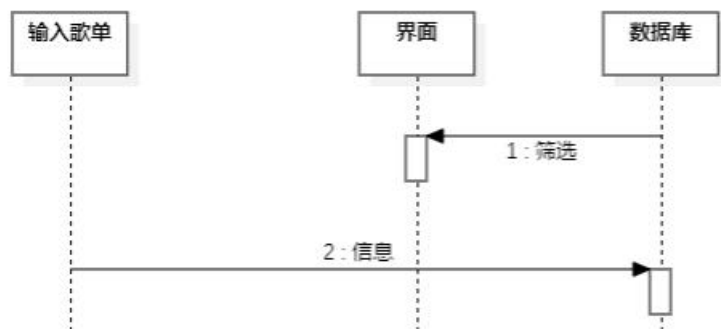
- 1) 将歌单名称、歌单标签、歌单收藏量、歌单播放量、歌单评论数信息存入数据库中
- 2) 将数据库中数据进行排行处理
- 3) 读取数据库信息，筛选前十歌单，显示在前端界面上

输出：在界面上展示前十热门歌单信息

活动图：



时序图：



2.3.4. 【SRS】歌曲排行

描述：歌曲排行是通过爬取 QQ 音乐热门歌曲信息进行统计后，在主页面中展示热门歌曲排行

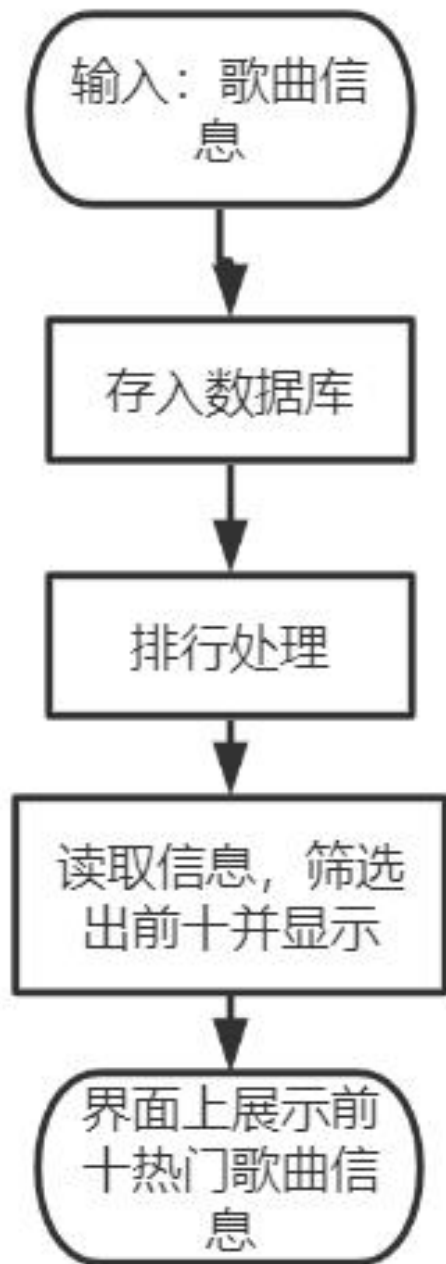
输入：歌曲名称、歌曲标签、歌曲收藏量、歌曲播放量、歌曲评论数、歌手信息、专辑信息、歌曲发行时间等信息

处理：

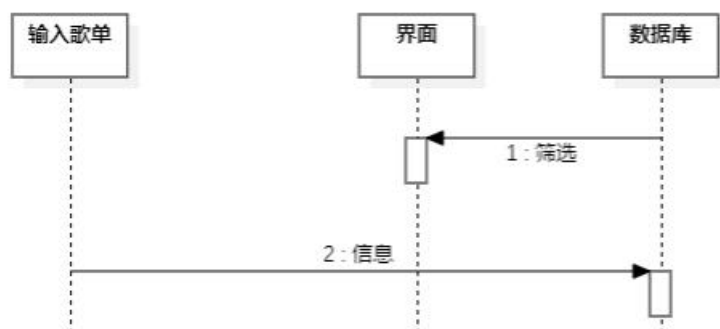
- 1) 将歌曲名称、歌曲标签、歌曲收藏量、歌曲播放量、歌曲评论数、歌手信息、专辑信息、歌曲发行时间等信息进行分析后存入数据库中
- 2) 将数据库中数据进行排行处理
- 3) 读取数据库信息，根据播放量、收藏量、评论数筛选前十歌曲，显示在前端界面上

输出：在界面上展示前十热门歌曲信息

活动图：



时序图：



2.3.5. 【SRS】个性化推荐

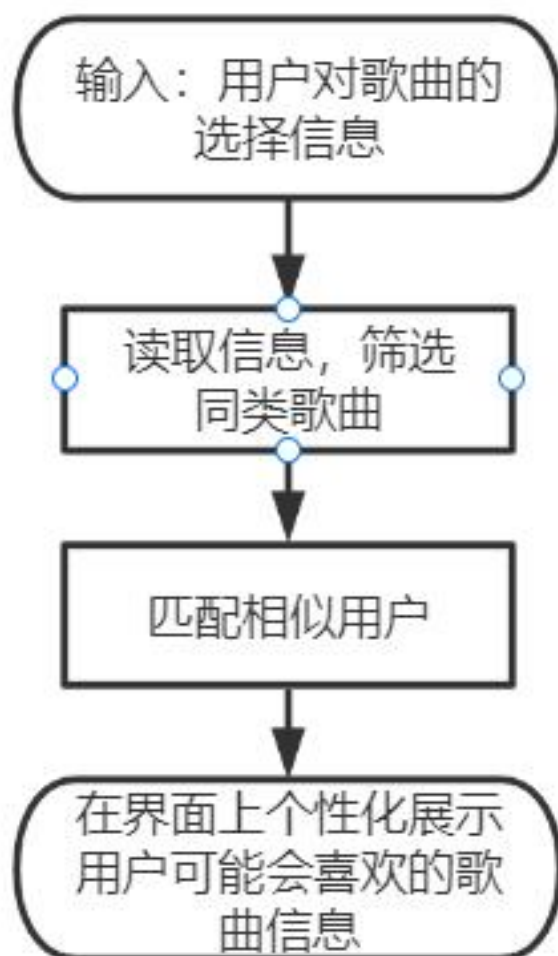
描述：个性化推荐是通过对用户个人喜好的分析进行歌曲推荐的模块，主要依据为用户个人标签选择以及用户所收藏的歌曲。

输入：用户对歌曲标签的选择、用户收藏的歌曲信息

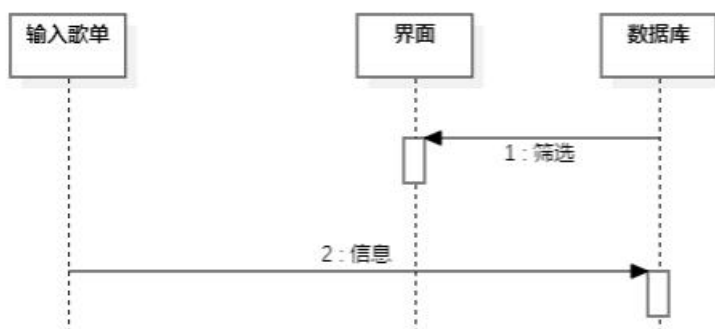
处理：

- 1) 读取用户对歌曲标签的选择，将标签信息与数据库中同标签歌曲信息进行了对比，筛选同标签歌曲以及与此标签关联度最高的相关歌曲
- 2) 在数据库中读取用户收藏的歌曲标签、类型、喜欢这首歌曲的用户的 IP 地址、性别、年龄段等信息，通过大数据分析存储在数据库中的关联信息库获取关联性，筛选同标签、相同或者相似用户画像下用户可能会喜欢的歌曲。
- 3) 将歌曲展示在界面上

活动图：



时序图：



3. 接口需求

3.1. 【SRS-012】内部接口

以下为内部接口的详细描述：

表格 4.1- 1 内部接口

接口	前端传入值	功能	后端传回值	类型
musicUser				
register	用户名 username, 密码 password	用户注册	根据页面提交的用户名 username 查询数据库, 如果查询到则返回用户名已存在, 注册失败信息; 否则注册成功, 将用户 id 和密码存入并返回注册成功结果。	post
login	用户名 username, 密码 password	用户登录	密码比对, 如果不一致则返回登录失败结果; 否则登录成功, 将用户 id 存入 Session 并返回登录成功结果。	post
logout		用户退出	清理 Session 中保存的当前登录员工的 id, 返回退出成功结果。	post
getById	用户账号 id Long id	根据 id 查询用户信息	查找成功返回用户实体对象; 否则返回没有查询到对应用户信息结果。	
update	用户名 username, 更新信息	编辑个人信息	根据页面提交的用户名 username 查询数据库, 如果查询到则返回更新失败结果; 否则返回用户信息修改成功结果。	post
collect				
collect/{id}	用户 id Long id	根据用户 id 查询收藏歌曲信息和对应的标签信息	根据页面提交的 id 查询数据库, 如果查询到则返回所有收藏歌曲信息; 失败返回 null。	get
collect	歌曲 id songId, 用户 id userId	取消收藏	根据对应歌曲 id 和用户 id 删除记录, 删除成功返回取消收藏成功结果; 失败返回失败结果。	delete
save	歌曲 id songId, 用户 id userId	收藏歌曲	根据对应歌曲 id 和用户 id 增加记录, 增加成功返回收藏成功结果; 失败返回失败结果。	post
recommend				

hot		热门推荐	返回热门歌曲信息，共 5 条	get
search/{song Name}	搜索歌曲名 songName	歌曲搜索	成功返回模糊查询搜索结果；失败返回 null	get
labels/page	当前页 page，每页 记录条数 pagesize, 标签信息 List<String> labels	分页查询通 过标签推荐 歌曲信息	成功返回根据标签信息推荐歌曲结果；失败返回 null	get
labels/collect	用户 id userId	个人收藏推荐	成功返回根据个人收藏信息推荐歌曲结果；失败返回 null	get

4. 其他需求

4.1. 需求

4.1.1. 【SRS-013】可靠性需求

- 1) 保证数据传输的可靠性
- 2) 防止因其中一个数据库文件出现错误导致整个系统不可用；
- 3) 对于逻辑紧密相关的数据库操作，应采用事务方式访问，以便在异常状态下也能保持数据的一致性。

4.1.2. 【SRS-014】可维护性需求

- 1) 提升软件的可理解性，模块化、详细地设计文档，结构化设计源代码内部文档，建议软件维护的难度；
- 2) 制定合适的软件设计原理，把握耦合、内聚、局部化、控制域的关系；

4.1.3. 【SRS-015】可测试性需求

- 1) 任何一项操作或输入都应该有预期的、明确的响应或输出，不管是正确的还是错误的甚至是异常的，这样软件才是可测的；

4.1.4. 【SRS-016】保障性需求

- 1) 交互层面，前端要对用户操作提供充足的反馈。例如，出现异常时，对用户提示出现异常的可能的原因和相应的解决方法，提高系统的可用度。
- 2) 系统出现错误时要有一定的自我恢复能力。例如当访问不到数据库时，系统需要在一定

时间内尝试自动重连。

4.1.5. 【SRS-017】性能需求

4.1.5.1. 数据精准度

- 1) 在执行数据的增加、删除、修改操作时，不允许因为程序原因导致操作失败；
- 2) 在执行数据增加时，不逊于发生多增加或重复增加的情况；
- 3) 在执行数据删除时，不允许发生多删除数据；
- 4) 在执行数据删除时，对有关联的数据要求删除完全；
- 5) 在执行数据修改时，要求保持对应的准确性。

4.1.5.2. 时间特性

- 1) 在管理员执行增加删除等操作时，数据库响应时间要求在 2s 之内；
- 2) Web 用户浏览网页时，页面响应时间要求在 4s 之内；
- 3) Web 用户进行网页端操作时，页面响应时间要求在 4s 之内；
- 4) 更新处理时间小于等于 2s；
- 5) 数据传送时间小于等于 1s；
- 6) 数据计算时间小于等于 20s；

4.1.5.3. 灵活性

- 1) 操作方式：项目对所有功能都提供取消操作，用户可以改变想法；
- 2) 有效时限：当响应时间、更新时间等超过限制时，项目可以进行“重新刷新”，从而重新发送请求，等待响应。

4.1.5.4. 客户端软件

- 1) 操作系统：Windows
- 2) 浏览器：IE、Chrome

4.1.5.5. 服务器端软件

- 1) 操作系统：Windows
- 2) 数据库：Mysql8.0

4.2. 约束

4.2.1. 业务环境约束

- 1) 本系统的开发、测试、部署的时间大约 2 个月左右，项目需要在结课之前完成开发和测试；

- 2) 了解爬虫相关的法律法规和关注项目是否构成侵权行为;
- 3) 无法使用付费开发平台及工具;

4.2.2. 使用环境需求

- 1) 项目面向各年龄、各阶层所有热爱音乐的人;
- 2) 项目使用 IE 和 Chrome 浏览器进行页面访问。

4.2.3. 技术约束

- 1) 框架: Spring Boot;
- 2) 数据库管理: Mysql、Navicat Premium;
- 3) 前端: Vue、axios、router
- 4) 后端: Mybatis-Plus
- 5) 开发模式: 瀑布模式

4.2.4. 人员约束

开发人员 6 人, 分别负责项目管理、前端设计及开发实现、后端设计及开发实现、前后端协作及测试。

5. 项目要求

5.1. 用户界面

5.1.1. 界面设计风格

设计风格为扁平简约风格, 渐变的蓝色为主题色突出重要元素, 能够有效地区分视觉层级根据功能的优先级来确定元素的视觉层级的高低。扁平化的元素能减少视觉噪点, 使用户专注于任务本身而不分散用户的注意力。

5.1.2. 美观性要求

系统用户操作界面应设计简洁美观, 利于用户操作。

5.1.3. 显示格式规定

输入框中文字选择左对齐, 提示文字为灰色, 输入的文字为黑色; 日期格式统一为 yyyy-MM-dd, 具体到天; 输入框中存在不符合规定的内容时, 给出错误提示。

5.1.4. 错误提示

系统错误需要给出错误提示，采取在屏幕上端弹出提示文字的方式进行错误提示。

5.2. 运行环境

以下是系统的软件环境。为使系统达到预期的性能，必须根据业务规模对硬件配置进行论证。

1) 客户端操作系统：

Windows98/Windows2000/Windows XP/Windows vista/ Windows7/Windows 10。

数据库访问：MyBatis

服务器的通讯：TCP/IP 协议；

2) 数据库：

管理系统：MySQL

管理工具：Navicat

3) 后端编程：

Java 运行环境：Jdk1.8；

IDE 工具：IntelliJ IDE

5.3. 用户文档

同本软件一起发行的用户文档包括：

- (1) 用户手册：Word 格式文件。
- (2) 在线帮助：HTML Help 格式文件，联机式。

5.4. 设计和实现上的限制

5.5. 假设和依赖

假设：

- 1) 使用 Spring Boot、Vue 框架； 界面简洁友好、易于使用；
- 2) 本系统的用户主要是非计算机专业人员，主要是通过网页的表单提交数据，主要的交换窗口是浏览器，方便使用；
- 3) 本系统客户端不需要安装运行环境，只通过浏览器进行数据的录入和查询；

约束：

- 1) 工期约束：本系统的开发、测试、部署的时间大约 2 个月左右；