第六组

系统设计文档

编 撰 人：邱亢迪、文灿玥、毛弘宇、
詹维璐、和靖东、唐靓
审 核 人：文灿玥
批 准 人：
批准日期：
文档版本：v1.0

文档信息

标题: 基于 QQ 音乐的大数据分析和音乐推荐平台
作者: 邱亢迪、文灿玥、毛弘宇、詹维璐、和靖东、唐靓
创建日期: 2023-4-15
上次更新日期: 2023-4-18
版本: version 2.0
部门名称: 第六组

修订文档历史记录

日期	版本	说明	作者
2023-4-16	1.0	创建文档初版	邱亢迪、文灿玥、唐靓
2023-4-18	2.0	根据实际情况进行部分内容修改	詹维璐、毛弘宇、和靖东

目 录

1. 简介

本文档用于对整个系统的软件构架进行初步的简要描述。

1.1 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面做出的重要决策。

1.2 范围

作用于整个分析设计、实施、测试阶段，将影响与上述活动相关的角色。

1.3 定义、首字母缩写词和缩略语

无。

1.4 参考资料

《需求规格说明书》、《项目开发计划》

1.5 概述

略

2. 构架表示方式

本文档采用 UML 分析设计语言对软件备选构架进行描述，使用 StarUML 工具生成软件构架的用例视图、逻辑视图和部署视图。对于进程视图和实施视图，由于在本软件备选构架中作用不明显，因而略去。

3. 构架目标和约束

用户操作简单，系统稳定性高、可扩展性高，构件重用性好，语言版本可扩展，采用面向对象进行分析与设计，采用 Java、Python、HTML、JavaScript 编程。

4. 关键用例视图

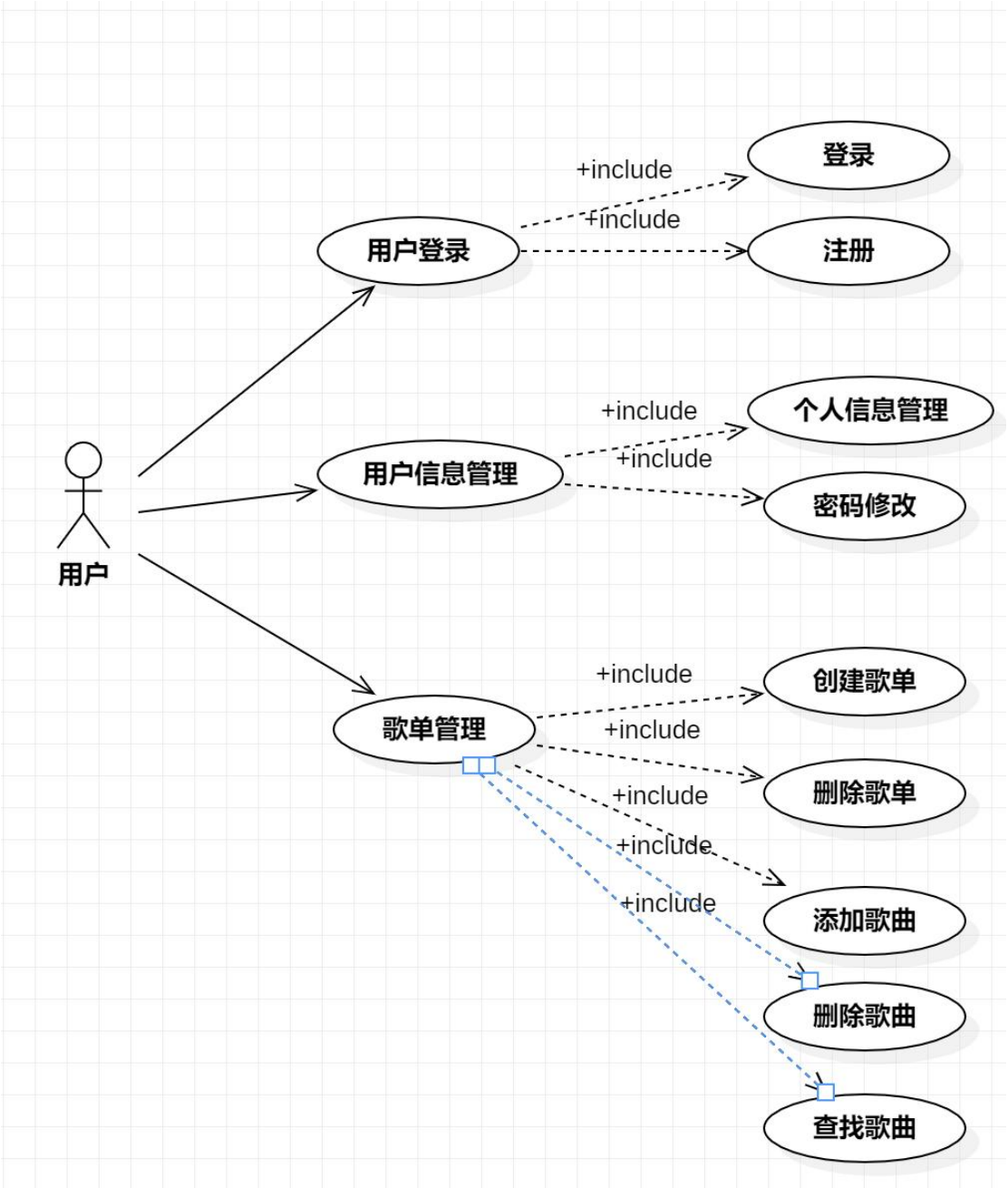
登录
+include
+include
用户登录
注册
+include
个人信息管理
用户信息管理
+include
密码修改
用户
+include
创建歌单
歌单管理
+include
删除歌单
+include
+include
添加歌曲
+include
删除歌曲
查找歌曲

图 1

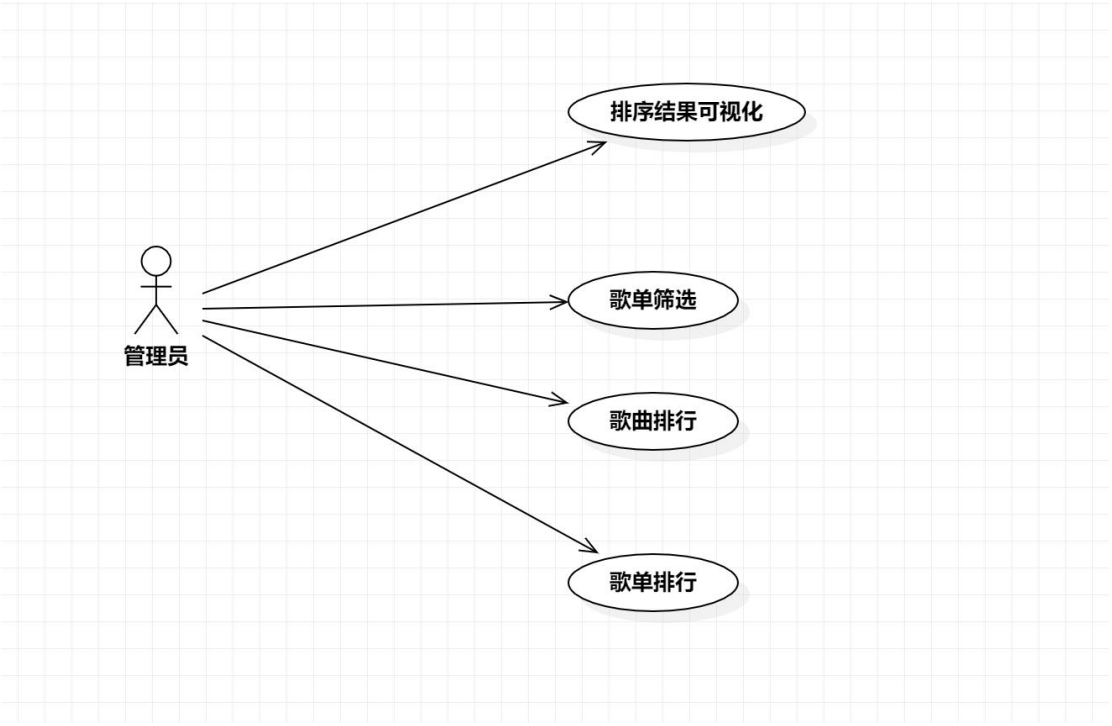

图 2

4.1 用户登录

用户进入首页，在登录框中输入自己的用户名、密码，点击按钮进入后台进行验证，如果验证通过，进入平台用户界面，如果未通过验证，则转至登录框，提示用户错误的原因，如此直到用户信息通过验证，再执行通过验证的后续操作，直到结束。

4.2 用户信息管理

个人信息编辑、密码修改

4.3 歌单管理

创建歌单、删除歌单、添加歌单至目标歌单、从目标歌单移除歌曲、歌曲搜索

4.4 可视化结果

要求进行排序结果可视化，通过播放量、收藏量等参考标准进行排序并最终以可视化图表展现与用户、视图更新

4.5 歌单排行推荐

歌单筛选：根据用户所选的标签在数据库中进行查询筛选，根据歌曲关联度、歌曲所有标签来进行歌曲推荐

歌单推荐：根据热度、标签等进行歌单推荐

歌曲推荐：根据已经收藏的歌曲，通过相关度、标签相似度等进行歌曲推荐，便于用户找到更多喜爱的歌曲

5. 层次结构

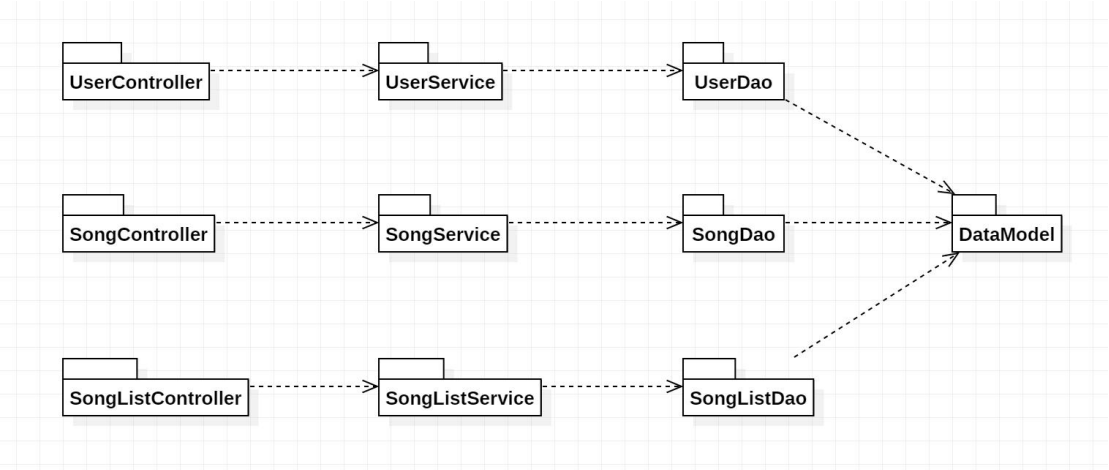

图 3

6. 逻辑视图

6.1 概述

说明构架的逻辑视图。该视图说明了最为重要的类、它们在服务包和子系统中的组织形式以及将这些子系统组织为层的方式。另外还说明了最为重要的用例实现（例如，构架的动态方面）。为了说明在构架方面具有重要意义的类、子系统、包和层的相互关系，可能会在逻辑视图中包含类图。

基于 QQ 音乐的大数据分析和音乐推荐系统的逻辑视图由四层组成。

- **用户服务层(Controller)**
    - 用户与系统交互的层面，通常包含用例分析中产生的边界类。
- **业务逻辑层（Service）**
    - 响应用户操作，组织和管理系统的正常运行，通常包含在用例分析中产生的控制类。
- **数据服务层（Dao）**
    - 内部数据结构、外部数据存取。
- **数据实体层**
    - 内部数据结构、外部数据结构，系统采用 Hibernate ORM 设计实施数据模型。

6.2 用户服务层

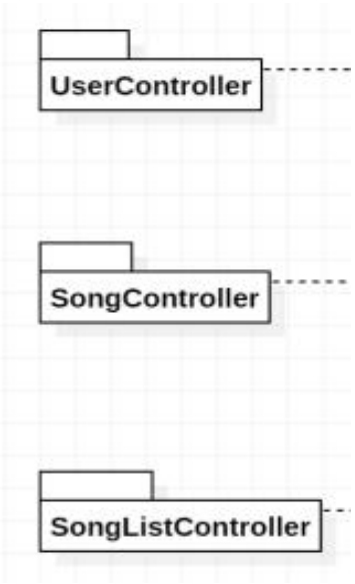

图 4

UserController：主要用于实现用户信息管理、用户登录、用户注册等功能逻辑。（见 6.2.1）

SongController:主要用于实现歌曲的增删查改等基础功能.（见 6.2.2）

SongListController:主要用于实现歌曲推荐列表、搜索列表功能。（见 6.2.3）

6.2.1 Package UserController

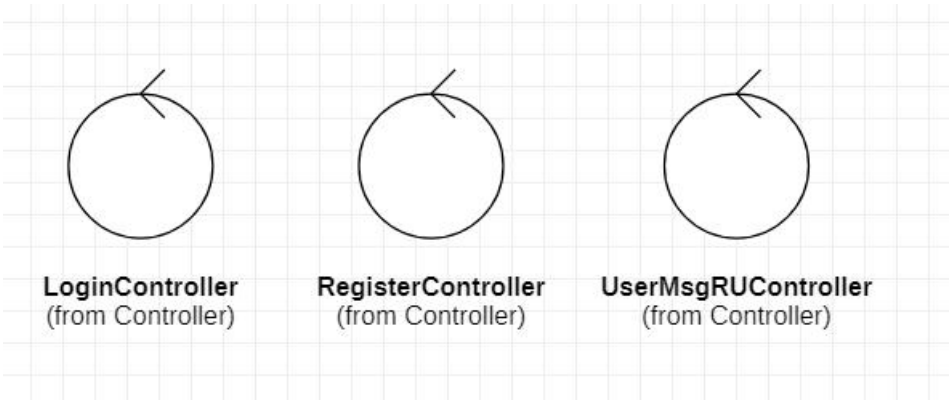

图 5

6.2.2 Package SongController

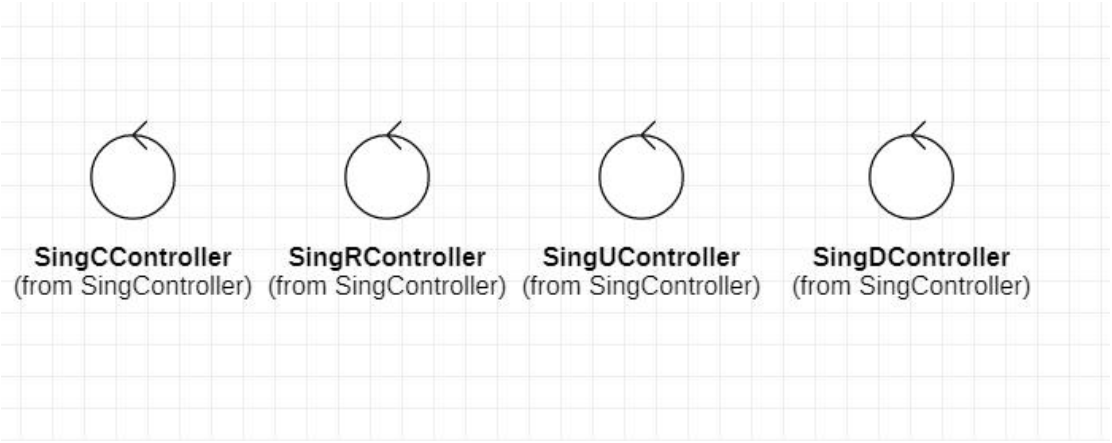

图 6

6.2.3 Package SongListController

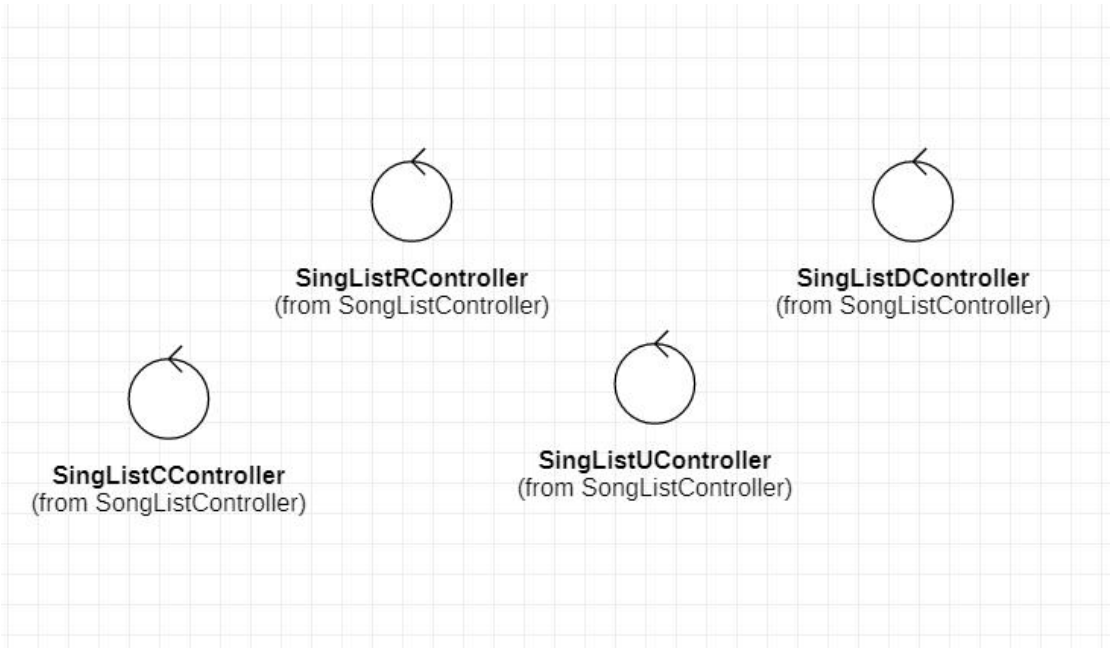

图 7

6.3 业务逻辑层

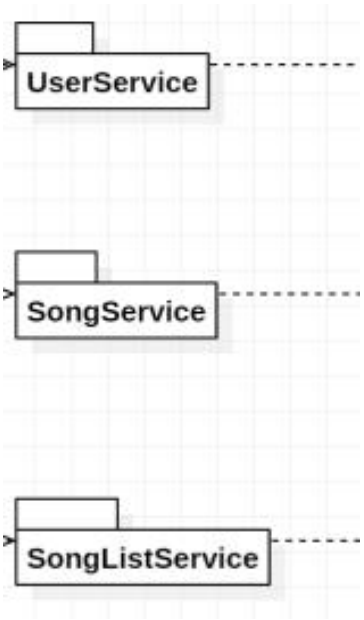

图 8

UserService:主要用于实现用户信息管理、用户登录、用户注册功能。

SongService:主要用于实现歌曲的增删查改等基础功能.

SongListService:主要用于实现歌曲推荐列表、搜索列表功能。

6.3.1 Package UserService

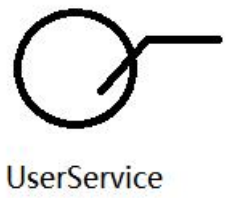

图 9

UserService 层是业务逻辑层，接收的数据来自于 Dao 层，Dao 层与数据库进行信息交互，通过数据库语言查询或筛选需要的信息，UserService 对其进行进一步的业务处理，以实现功能，用户注册要将用户信息添加至数据库，并且在此之前要对输入的数据进行筛选，这项工作在 Service 完成。

6.3.2 Package SongService

图 10

将 Dao 层从数据库中获取的信息生成对象，传入 Controller 层中便于前端获取并展示。SongService 主要用于实现歌曲的增删查改功能。

6.3.3 Package SongListService

图 11

将 Dao 层从数据库中获取的信息生成对象，传入 Controller 层中便于前端获取并展示。SongListService 主要用于实现歌曲推荐列表、搜索列表功能。

6.4 数据服务层

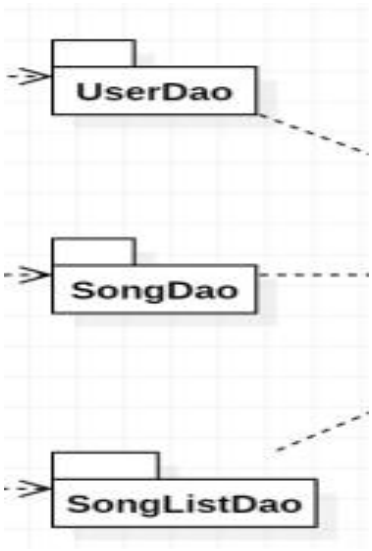

图 12

6.4.1 Package UserDao

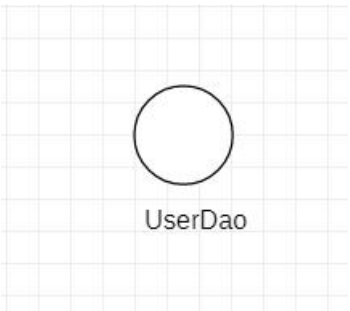

图 13

Dao 层从数据库获取信息或将数据存储到数据库，UserDao 层主要是用户信息、用户账号与密码等信息的增删查改。

6.4.2 Package SongDao

图 14

SongDao 层用于歌曲信息包括歌名、歌手、发行时间、播放量等基础信息的增删查改。

6.4.3 Package SongListDao

图 15

i

SongListDao 层用于搜索或推荐时查询歌曲的相关详情信息。

7. 性能设计

4.1. 响应时间特性

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 0.1 秒之内；
项目采用 HiKari 连接池，HiKariCP 是数据库连接池的一个后起之秀，号称性能最好，可以完美地 PK
掉其他连接池，是一个高性能的 JDBC 连接池，基于 BoneCP 做了不少的改进和优化。
2) Web 用户浏览时，页面响应时间要求在 1 秒之内。
a)对于大量数据的传输，例如列表的数据传输，采取分页的方式，一次加载少量的数据，提高页面
的响应时间。
b)在页面渲染的生命周期前进行 ajax 请求，提高页面加载效率。
c)对于需要重复使用的数据，例如用户登录信息，使用 Vuex 进行存储，避免重复的数据请求。

4.2. 并发特性

1）数据库并发

数据库支持超过 500 个用户的并发访问能力。
采用 mysql 数据库，默认支持高并发
2）访问并发
平台具备不少于 10000 个访问并发的能力。
采用 Tomcat 服务器，默认支持高并发
3）传输并发
系统业务功能包括附件和图片的传输的时候，需提供稳定快速的传输效率，以及支持多附件多图片并发
上传和下载的能力。
采用阿里云服务器，以达到稳定网络状态

8. 五性设计保证

7.1. 可靠性

1） 后端部署在云服务器上，带宽至少为 1M，保证数据传输的可靠性。
2） 数据库进行冗余处理，防止因其中一个数据库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。

7.2. 可维护性

系统采用前后端分离的方式，同时将系统划分成逐个小模块，降低了系统的耦合度。系统某一个模块发
生的错误通常不会影响到其他模块的正常使用。源码的编写风格需要符合编写规范，具有一致性，并且
注释和方法、变量的命令要清楚有意义。以此保证项目易于修改和维护。招生宣传报名系统
系统设计文档
第五组
48

7.3. 安全性

要求系统通过一定的措施防范通过浏览器对系统的破坏活动，包括：
1）为了系统数据的安全性，会定时对数据库的数据进行人工备份。
2）系统会保护用户的隐私信息，没有相应权限的用户是不能访问没有权限的信息。
3）为了个人密码的安全，系统会对个人的密码进行加密。
4)对于不符合要求的数据，系统提醒会用户或者进行过滤等操作，防止不良数据进入系统。

7.4. 测试性

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。
错误输出易于识别，通过日志自动分析或者界面高亮显示的方式，能有助于发现。
2） 增加输出参数、减少变量重用，包括打印内部信息、将局部变量作为输出、增加断言、增加局部变
量等。

3） 采用模块化设计，各模块支持独立测试，对于每个相对独立的模块设计专用的测试驱动和测试桩，
模块异常时不影响其他模块的测试。
4） 后端需要打印日志输出，至少保存 15 日。
5） 方法和代码提供适当的注释，方便测试。

7.5. 保障性

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，当访问不到数据库是，系统需要在一定时间内
尝试自动重连。
3）提供用户错误反馈入口，以便及时发现和处理系统出现的问题。

9. 接口设计

9.1 内部接口设计

以下为内部接口的详细描述：

表格 7.1- 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	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

		签信息		
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/{songName}	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

10.部署视图

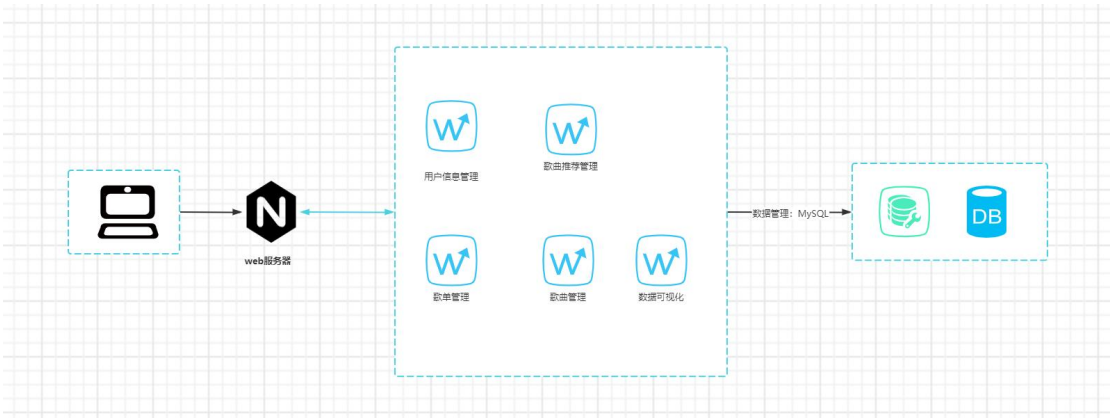

图 16