|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | |  | 版 本： | V1.0 | |  |
|  | |  | |  | 密 级： | 内部 | |  |
|  | |  | |  | 总页数： | 25 | |  |
|  | | | | | | | | |
| **大数据应用工程实践课程设计文档**  文档控制号：  在线音乐播放网站软件  概要设计说明 | | | | | | | | |
|  | | | | | | | | |
|  | **单 位** | | 20及数据科学与大数据技术班 | | | |  | |
|  | **参与人** | | 张金聪 李光华 耿晓雪 刘光蕊  郭彦泽 毛欣竹 徐艳华 | | | |  | |
|  |  | |
|  | **校 对** | |  | | | |  | |
|  | **审 核** | |  | | | |  | |
|  | **批 准** | |  | | | |  | |
|  | | | | | | | | |
|  | | | | | | | | |
|  | **楚雄师范学院数计学院** | | | | | |  | |
|  | | | | | | | | |

目录

[一． 项目简介 3](#_Toc25754)

[（一）项目简介 3](#_Toc30043)

[（二）访问yin音乐网的状态： 3](#_Toc598)

[二． 项目分析 3](#_Toc21596)

[（一）功能实现 3](#_Toc3929)

[三． 总体设计 4](#_Toc14186)

[3.1音乐网站组成 4](#_Toc12174)

[3.2音乐网站技术架构 5](#_Toc17353)

[四． 概要设计 6](#_Toc26638)

[4.1分层（总体）设计 6](#_Toc5009)

[4.2数据库设计 7](#_Toc22150)

[4.3实体类设计 12](#_Toc20957)

[4.4项目的配置文件 14](#_Toc5411)

[4.5前后端交互 15](#_Toc26022)

[五． 项目结果 20](#_Toc13963)

# 项目简介

（一）项目简介

Yin音乐网站设计主要是为了更好的直接的更改音乐网站里面的信息，修改歌手歌曲信息等。它的特点是前端和后端可以相互交互，更改歌曲信息歌手信息的同时数据库也会更新。

该音乐网站的设计范围主要包括前端和后端的交互，创建属于自己的仓库，并把数据、文件等传入自己所建的仓库保存。Yin音乐网站设计可以通过npm run serve命令代开音乐网站，并展示yin音乐的具体信息。

（二）访问yin音乐网的状态：

1.管理员：

（1）未登录状态：只显示一个登录页面，不能查看任何信息。

（2）登录状态：可以实现未登录状态下的所有操作，同时可以有更多的操作：查看歌单歌手及用户的详细信息，还可以查看用户数量，歌曲数量，歌手数量及歌单数量，用户和歌手的性别比例，歌曲类型和歌手国际等，还可以直接对歌曲普通用户及歌手进行编辑，修改歌单照片等。

2.普通用户：

（1）未登录状态：可以查看歌单、歌手的详细信息及分类，还可以搜索歌曲、歌手。

（2）登录状态：可以收藏歌曲、评论歌曲等。

# 项目分析

（一）功能实现

1.用户注册和登录功能：

（1）提供用户注册页面，收集用户基本信息，并进行验证和存储。

（2）提供用户登录页面，验证用户身份，以便后续操作和个性化功能。

2.音乐搜索和播放功能：

（1）提供搜索框和搜索按钮，允许用户输入关键词搜索音乐。

（2）根据用户输入的关键词，在音乐库中进行检索，并显示相匹配的音乐结果。

（3）用户点击播放按钮后，开始播放所选音乐，并显示音乐播放进度和控制按钮。

3.歌曲上传和管理功能：

（1）提供上传页面，管理员可以上传音乐文件和相关信息。

（2）在后台进行音乐文件的存储和管理，并为每首歌曲分配唯一的标识符。

（3）提供管理界面，允许管理员编辑和删除已上传的歌曲信息。

4.创建和管理歌单功能：

（1）允许用户创建自己的歌单，将喜欢的音乐添加到歌单中。

（2）提供管理界面，允许用户对歌单进行编辑和删除。

5.评论功能：

（1）允许用户对歌曲进行评论，表达自己的感受和观点。

6、用户个性化设置：

（1）允许用户自定义个人资料，包括头像、昵称和个性签名等。

7、音乐排行：

（1）提供热门音乐排行榜，展示当前热门的歌曲和艺术家。

8、歌词展示：

（1）对于支持歌词的音乐文件，提供歌词展示功能，帮助用户更好地理解和欣赏音乐。

# 总体设计

## 3.1音乐网站组成

1. 组成部分

1.前端（Front-end）：

用户界面（UI）：包括网站的主页、播放器、搜索功能、歌曲排行榜等。

用户体验（UX）：设计用户友好的界面，确保良好的用户交互和体验。

HTML/CSS/JavaScript：构建和呈现前端页面的技术基础。

2.后端（Back-end）：

服务器端技术：处理请求、处理逻辑、与数据库交互等。

数据库管理系统：用于存储和管理音乐数据、用户信息以及其他相关数据的数据库系统。

API（Application Programming Interface）：用于与前端页面交互和数据传输的接口。

3.用户管理系统：

用户注册和登录功能：让用户创建账户并登录，以便于他们收藏、评论和与其他用户互动。

用户个人资料管理：用户个人信息、收藏的歌曲、歌单等。

4.音乐库和数据管理：

音乐上传和存储：允许音乐供应商、艺术家或用户上传音乐，并在服务器上存储。

元数据管理：用于上传和关联音乐文件的元数据，如歌曲标题、艺术家、专辑、封面等信息。

音乐分类和标签：对音乐进行分类、标记和组织，以方便用户浏览和发现音乐。

5.播放：

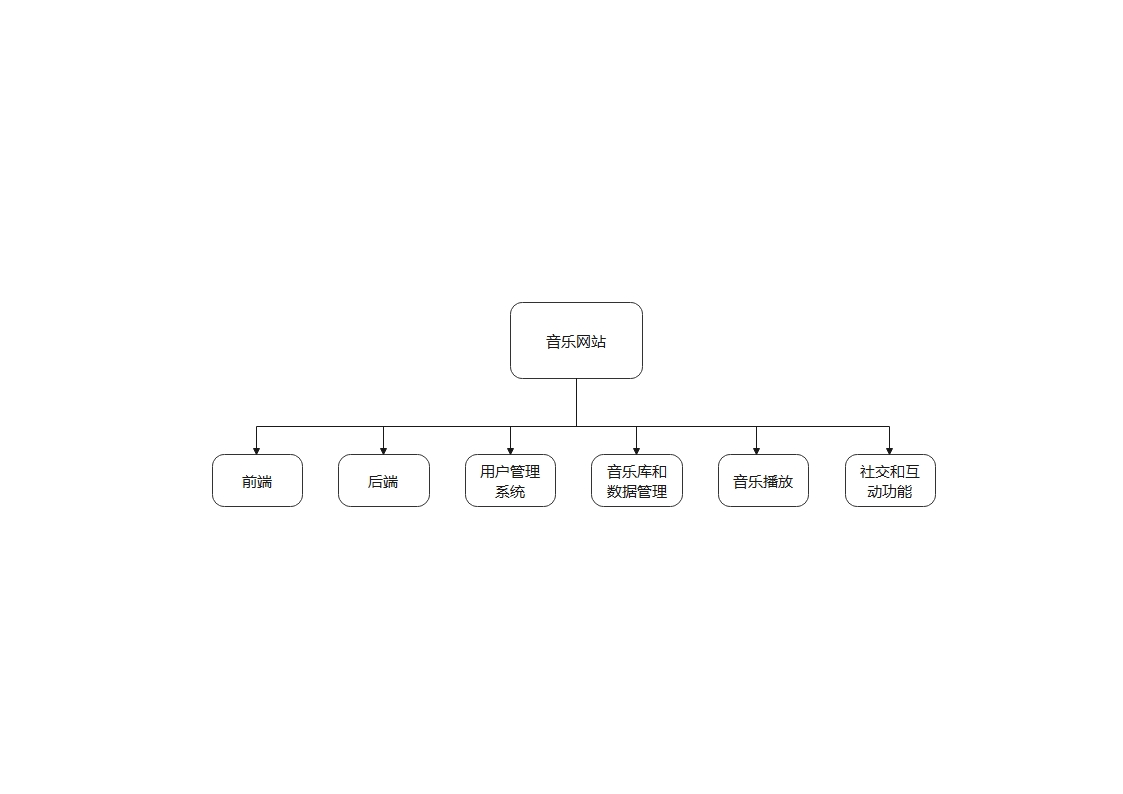
音乐播放器：提供基本的音乐播放功能，如播放、暂停、上一首、下一首等。

6.社交和互动功能：

评论和评分：用户可以在音乐上留下评论和评分，与其他用户交流。

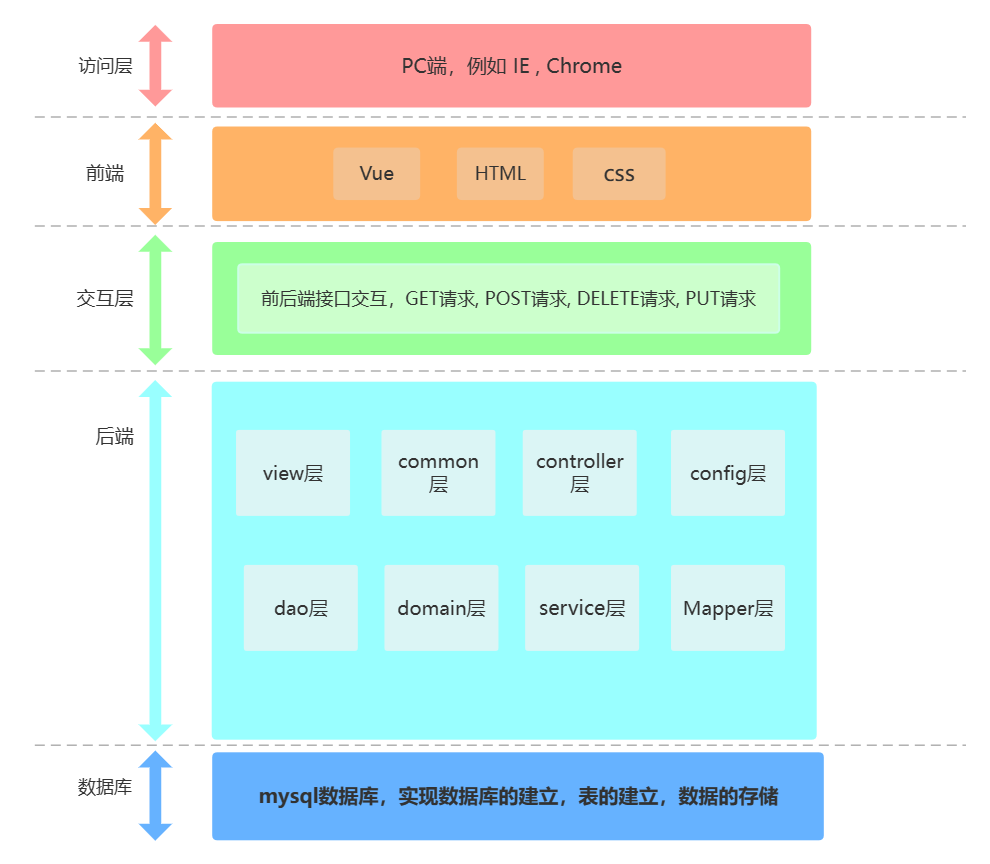
分享和推荐：用户可以与好友分享自己喜欢的歌曲、歌单或推荐给其他用户。

社交集成：集成第三方社交媒体平台，让用户更方便地与其他社交圈子分享音乐。

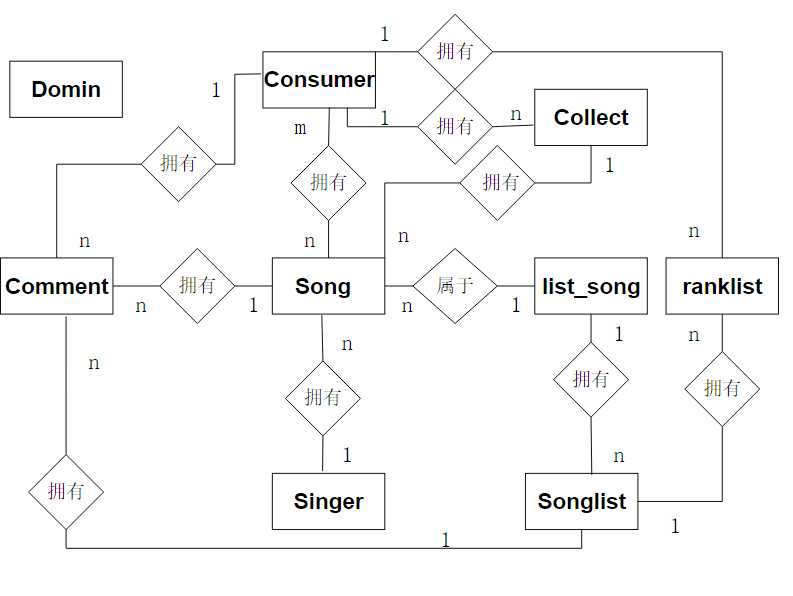


## 3.2音乐网站技术架构

（一）架构图展示



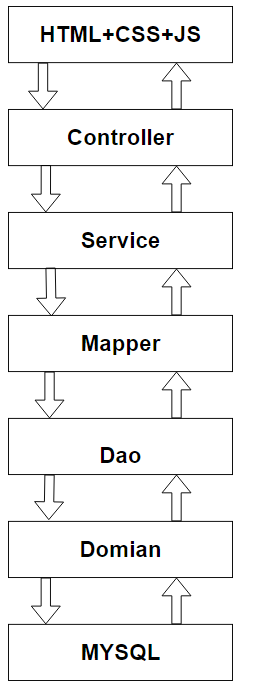
1. ER图展示



# 概要设计

## 4.1分层（总体）设计

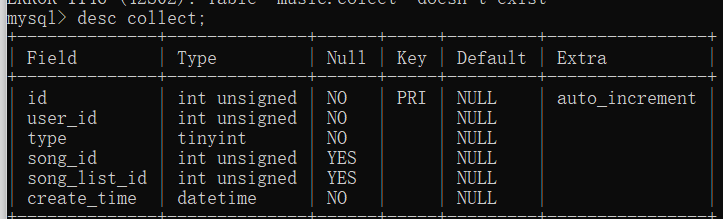
（一）先编写domain 层定义数据库中的属性，先编写domain 层,然后在dao层写实现方法的mysql语句，然后实现service接口，继续写service的实现类，然后在controller写实现方法，然后配置前后端交互文件，然后在前端实现并返回数据库实现交互。

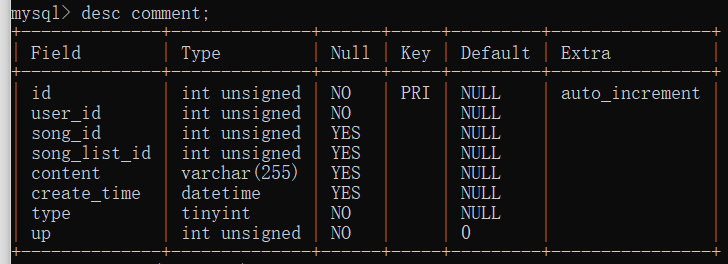


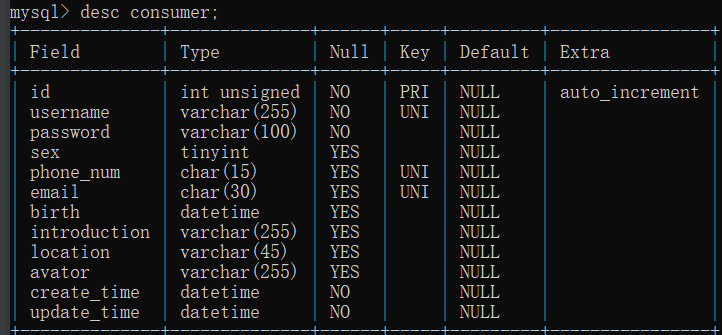
## 4.2数据库设计

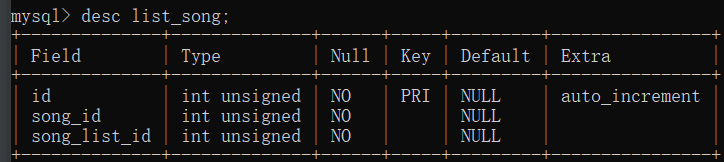
（一）表结构设计

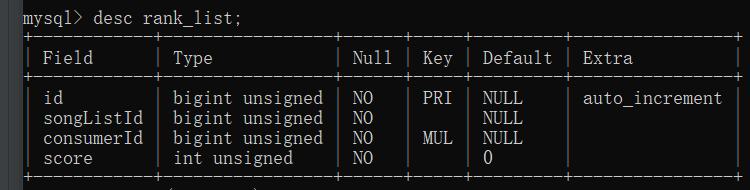
1.

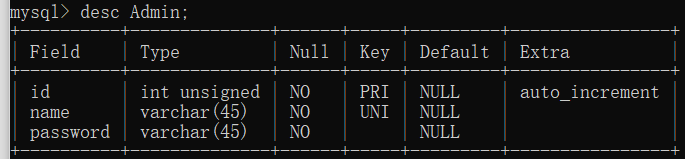


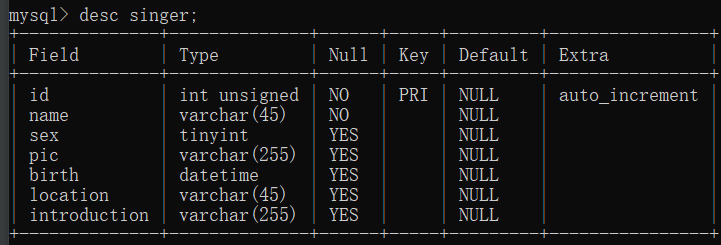


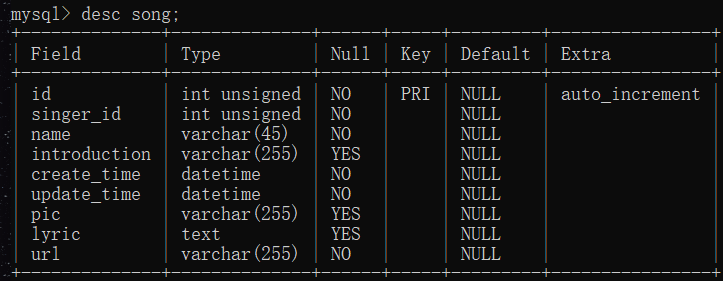


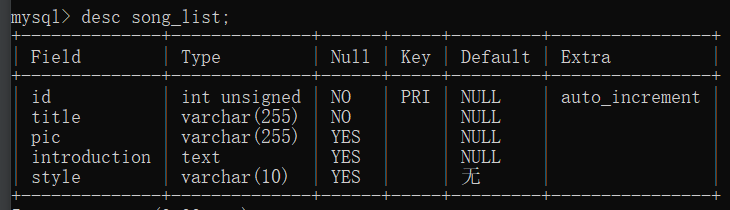












（二）类图

为了更好的描述系统中的类、接口、属性和方法等元素之间的关系。可以帮助我们更好地理解系统的架构和设计，类图中的元素之间可以存在多种关系，如继承、实现、关联、聚合等。继承表示一个类从另一个类继承属性和方法；实现表示一个类实现了多个接口；关联表示两个类之间存在某种联系；聚合表示多个类组合成一个整体。

1.图形基础

将整个系统划分为View层，Controller层，Service层，DAO层四层，使用Spring MVC负责请求的转发和视图管理，Spring实现业务对象管理，Mybatis作为数据对象的持久化引擎。

（1）后端：

**Controller层**：Controller层负责具体的业务模块流程的控制。

**DAO层：**DAO层主要是做数据持久层的工作，负责与数据库进行联络的一些任务都封装在此。

**Domain层:**Domain层负责定义私有属性和公有方法，链接数据库。

**Service层：**Service层主要负责业务模块的逻辑应用设计。

**mapper：**存放XML文件,有关数据交换和配置文件 。

**application.properties** ：用于存放程序的各种依赖模块的配置信息，比如 服务端口，数据库连接配置等。

**gitignore：**使用版本控制工具 git 的时候，设置一些忽略提交的内容。

**YinMusicApplication.java：**SpringBoot 程序执行的入口，执行该程序中的 main 方法，启动当前SpringBoot项目。

（2）前端：

**View层：**View层与控制层结合比较紧密，需要二者结合起来协同工发。View层主要负责前台jsp页面的表示。

**dist和assets：**存放静态资源。如图片、CSS、JavaScript 等 。

2.图形解释

（1）类图是静态图的一种，我们为yin音乐网站设计了九个类，分别为admin,comment,ranklist,song,songlist,comsumer，singer,list\_song,collect。其中song与comment，songlist与comment,comsumer与songlist,singer与song为一对多的关系，song与list\_song，songlist与list\_song为多对一的关系，ranklist依赖于song,collect依赖于list\_song。

（2）-： private +： public

冒号前是方法名/变量名（根据有无括号区分），冒号后是返回参数/变量类型（根据有无括号区分），如果没有冒号的话表示方法返回空（也有人通过：void表示返空）。

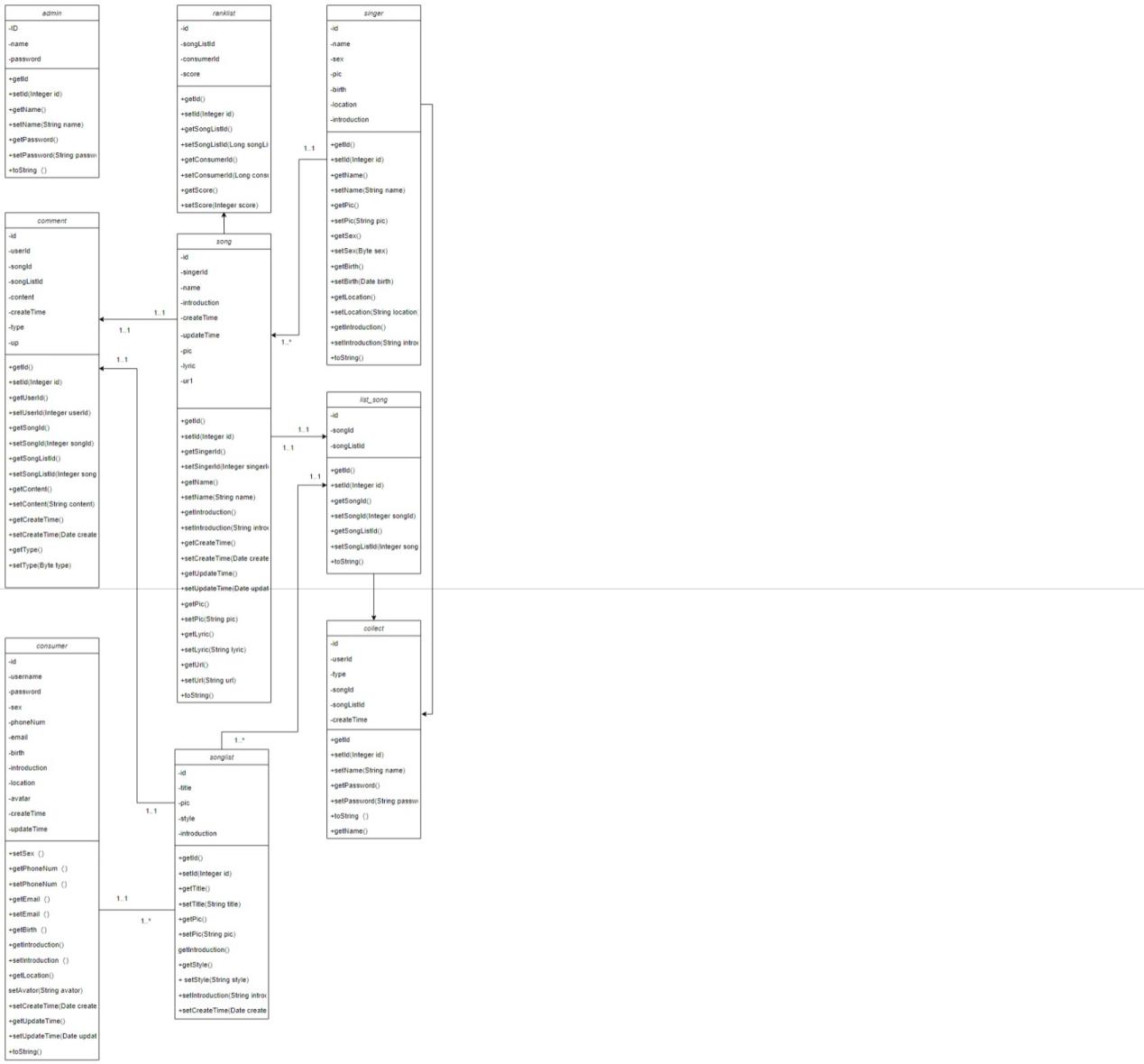
（3）1:表示一个集合中的一个对象对应另一个集合中一个对象。

0..\*:表示一个集合中的一个对象对应另一个集合中的0个或多个对象。(可以不对应)

1..:表示一个集合中的一个对象对应另一个集合中的一个或多个对象。(至少对应一个)

\*:表示一个集合中的一个对象对应另一个集合中的多个的对象。

类图如图所示：



## 4.3实体类设计

（一） 显示实体类的属性。

1.Admin实体类

private Integer id; //定义管理员ID

private String name; //定义管理员的名字

private String password; //定义管理员的密码

2.Collect实体类

id（Integer类型）： //表示收藏记录的唯一标识符。

userId（Integer类型）： //表示收藏记录所属的用户的唯一标识符。

type（Byte类型）： //表示收藏的类型。

songId（Integer类型）： //表示被收藏的歌曲的唯一标识符。

songListId（Integer类型）：//表示被收藏的歌单的唯一标识符。

createTime（Date类型）： //表示收藏记录的创建时间。

3.Commentt实体类

private Integer id; //评论的唯一标识符（主键）

private Integer userId; // 评论的发表者的用户ID。

private Integer songId; // 评论所针对的歌曲的ID

private Integer songListId; // 评论所属的歌单的ID

private String content; // 评论的内容

private Date createTime; // 评论的创建时间

private Byte type; // 评论的类型，

private Integer up; // 评论的类型，可能是一个表示评论类型的字节值

4.Consumer实体类

private Integer id; // 普通用户的唯一标识符（主键）

private String username; // 普通用户的用户名

private String password; // 普通用户的密码

private Byte sex; // 普通用户的性别，

private String phoneNum; // 普通用户的手机号码

private String email; // 普通用户的电子邮件地址

private Date birth; // 普通用户的出生日期

private String introduction; // 普通用户的简介或个人介绍

private String location; // 普通用户的所在地

private String avator;···· // 普通用户的头像文件路径或URL

private Date createTime; // 普通用户的创建时间

private Date updateTime; // 普通用户的更新时间

5.ListSong实体类

private Integer id; // 歌单的唯一标识符（主键）

private Integer songId; // 歌曲的唯一标识符

private Integer songListId; // 关联的歌单的ID。

6.RankList实体类

private Long id; // RankList 的唯一标识符（主键），

private Long songListId; //歌单的ID

private Long consumerId; //普通用户的ID

private Integer score; //普通用户的评分

7.Singer实体类

private Integer id; // 歌手的唯一标识符（主键）

private String name; // 歌手的姓名

private Byte sex; // 歌手的性别

private String pic; // 歌手的照片路径或URL，

private Date birth; // 歌手的出生日期

private String location; // 歌手的所在地

private String introduction; // 歌手的简介或介绍

8.Song实体类

private Integer id; // 歌曲的唯一标识符（主键）

private Integer singerId; // 歌曲所属的歌手的ID。

private String name; // 歌曲的名称

private String introduction ;// 歌曲的简介或介绍

private Date createTime; // 歌曲的创建时间

private Date updateTime; // 歌曲的更新时间

private String pic; // 歌曲的封面图片路径或URL

private String lyric; // 歌曲的歌词

private String url; // 歌曲的音频文件路径或URL

9.SongList实体类

private Integer id; // SongList 的唯一标识符（主键）

private String title; // 歌单的标题或名称

private String pic; // 歌单的封面图片路径或URL

private String style; // 歌单的风格或类型

private String introduction; // 歌单的简介或介绍

## 4.4项目的配置文件

（一）配置文件

spring.datasource.url=jdbc:mysql://localhost:3306/music?serverTimezone=Asia/Shanghai&characterEncoding=utf8

spring.datasource.username=root

spring.datasource.password=password

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

mybatis.typeAliasesPackage=com.example.yin.domain

mybatis.mapperLocations=classpath:mapper/\*.xml

spring.jackson.date-format=yyyy-MM-dd HH:mm:ss

spring.jackson.time-zone=GMT+8

（二）端口号

server.port=8888

（三）热部署生效

spring.devtools.restart.enabled=true

（四）设置重启的目录

spring.devtools.restart.additional-paths=src/main/java

#classpath目录下的WEB-INF文件夹内容修改不重启

spring.devtools.restart.exclude=WEB-INF/\*\*

# 关闭CONDITIONS EVALUATION REPORT及自动配置内容向控制台的输出

logging.level.org.springframework.boot.autoconfigure=ERROR

logging.level.org.apache.tomcat=ERROR

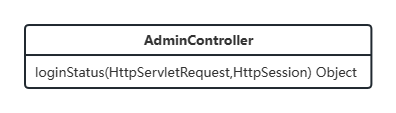
logging.level.org.apache.catalina=ERROR

## 4.5前后端交互

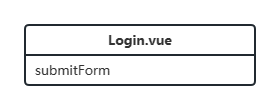
（一）管理员系统和后端交互

1.管理员登录

登录管理员用户，验证登录，根据用户输入得到的用户名和密码查询数据库，验证该用户是否存在与数据库中。

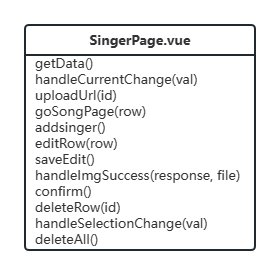
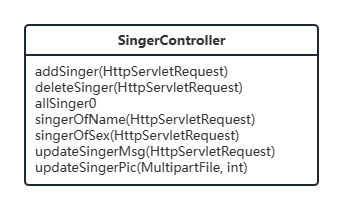


在Login.vue中使用submitForm提交表单的方法，包括发送登录请求、



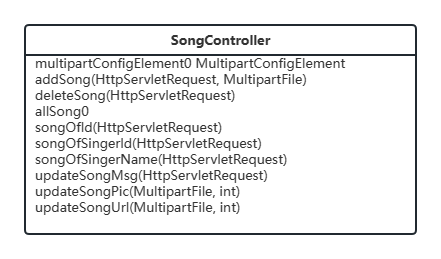
2.歌手管理

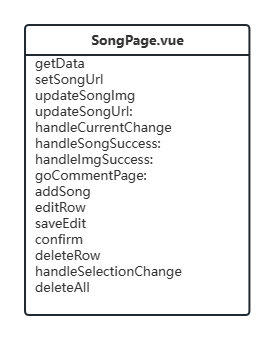
在SingerService里面定义查找歌手信息、添加歌手信息，更新歌手信息，更新头像，根据歌手ID删除歌手信息。然后在SingerPage.vue中实现，并能与数据库交互，实时更新数据库中的信息和获取数据库中的信息。



3.歌曲管理

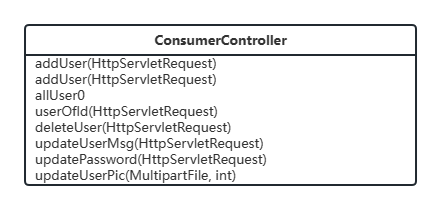
在SongController里面可以查找歌曲信息，添加歌曲，更新歌曲信息，更新图片，更新歌曲文件，删除歌曲信息。然后在SongerPage.vue中实现，并能与数据库交互，实时更新数据库中的信息和获取数据库中的信息。

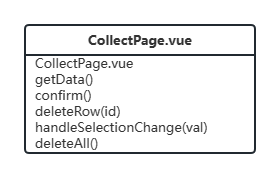


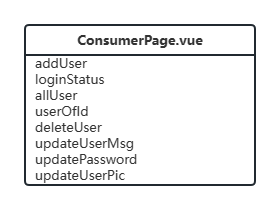


4.用户管理

通过COnsumerController里面课更新用户信息，查找用户信息，添加用户信息，更新用户头像，更新用户信息，删除用户信息，收藏歌曲信息。然后在CollectPage.vue和ConsumerController中实现，并能与数据库交互，实时更新数据库中的信息和获取数据库中的信息。

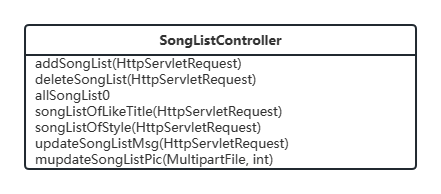


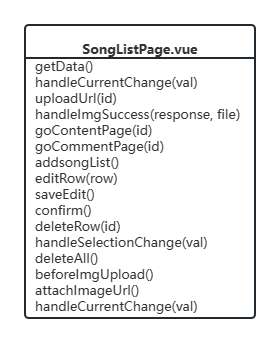




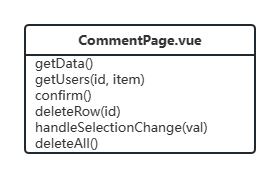
5.歌单管理

在SongListController中查找歌单信息，依据查找到的歌单信息添加歌单信息，更新歌单头像，更新歌单信息，删除歌单信息。然后在CollectPage.vue和ConsumerController中实现，并能与数据库交互，实时更新数据库中的信息和获取数据库中的信息。



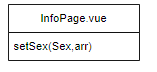


歌单评论在CommentPage.vue中实现可以获取评论数据，删除评论。



6.系统首页

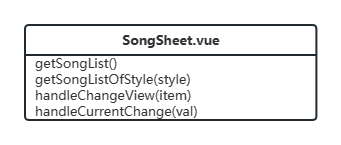
在InfoPage.vue中统计了用户数、歌曲数、歌手数、歌单数。



（二）普通用户和后端交互

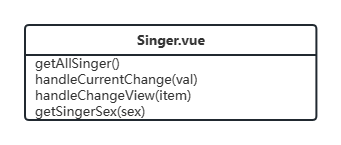
1.查看歌单信息

在SongSheet.vue中查看全部歌单，或者查看某类歌单信息。

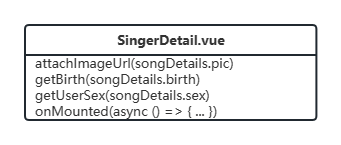


2.歌手信息

在Singer.vue中实现查看所有歌手信息、查看某类歌手信息



在SingerDetail.vue中实现获取歌手信息：头像、生日、性别…



3.排行榜

在RankListController中查看歌单信息，查看歌曲信息，歌单评分，在数据库中查询，查看是否交互成功。



4.用户操作：播放歌曲，在song.ts中实现循环播放、上一首、暂停、下一首、音量、收藏、下载，点击歌曲头像查看歌词。



5.收藏操作，在controller中收藏歌曲，取消收藏，是否收藏，更新收藏的歌曲，在数据库中查看是否取消收藏成功。



6.评论操作，在CommentController中显示所有的评论信息，添加评论，删除自己的评论



# 项目结果

（一）结果展示



