**南京航空航天大学金城学院**

**信息工程学院**

**音乐网站系统设计报告**

实习项目/岗位 音乐歌曲网站

实习单位/地点 南航金城

学生姓名 汤亚东

学 号 2018027421

专 业 软件工程

联系方式 15261176327

# 需求分析

网站架构：ssm

Java版本：JDK8

前端：使用jsp，html配合css，js

数据库：mysql

**基本功能：**

1. 音乐播放：进入音乐网站，点击歌曲进入音乐播放界面，打开音乐播放键，歌曲开始播放；

2. 音乐上传：用户个人中心中，进行音乐的添加，输入音乐名称，类型以及图片等信息之后便可以上传音乐文件；

3. 音乐查询：通过音乐名称，音乐分类，音乐所唱的语言进行音乐查询；

4. 音乐收藏：选中收藏按键，音乐收藏成功；

5. 用户注册登录：输入用户名、姓名、出生日期等信息并且提交，进行用户注册；

6. 账号管理： 管理员成功登录后，可以查看账号信息、增添账号、查看已经注册的用户、添加新的用户以及对管理员账号的密码进行相应的修改等；

7. 音乐分类管理：对音乐的类别来进行增添和查询。添加分类时输入分类名称并提交，可在首页添加新的音乐分类，音乐分类查询可以查看已经拥有的音乐分类；

8. 音乐管理：对音乐进行增删改查操作，添加音乐时将音乐名称，音乐类型，音乐语言等信息一并输入，即可进行音乐的增添操作。

# 总体设计

## 使用技术概述

整体架构：ssm+jsp+js

数据库：Mysql

后端：Spring Framework,Spring MVC, ,Jakson,Servlet, commons-fileupload,JDK8

数据库连接：Mybatis, Mybatis,Mybatis-Generator,PageHelper

前端：Jsp,html,Jsp,Javascript

接口调试工具：Postman

版本管理：Git+Gitee

包管理：Maven

服务器：Tomcat

日志：Log4j

测试工具：Junit

单独搭建一个文件上传服务器，用来接收上传的音乐与图片数据

## 架构预览

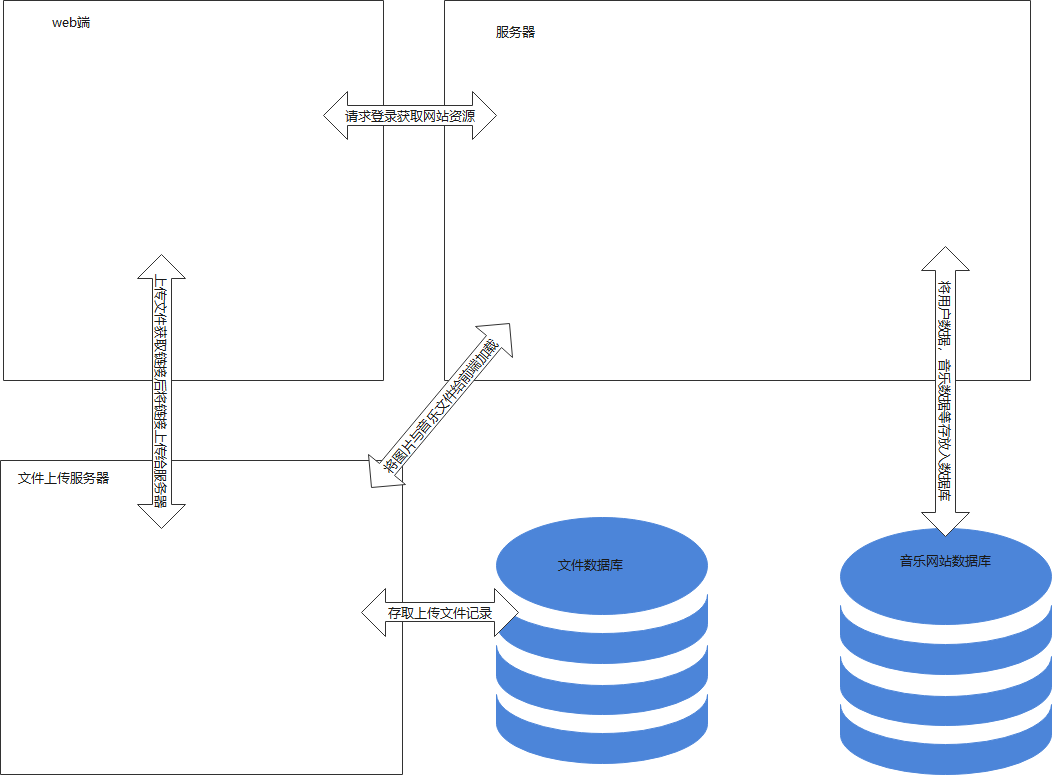


图 1 总体运行流程

# 数据库设计

## 数据库说明

数据库使用了mysql,使用可视化管理工具对数据库进行管理，后端使用mybatis与数据库进行交互

## ER图

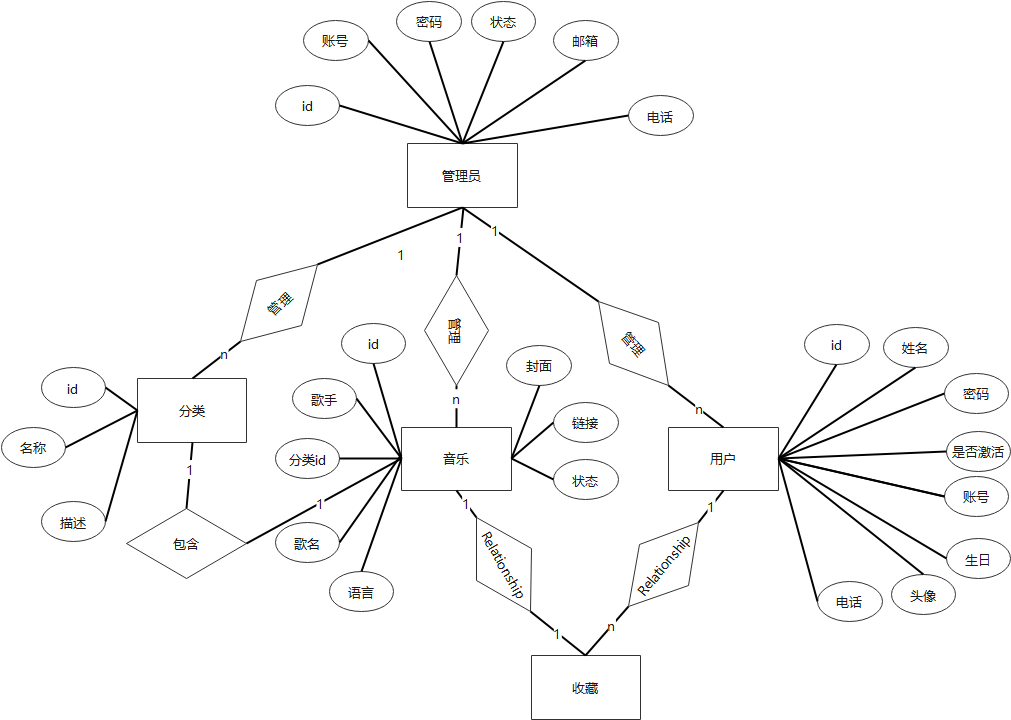


图 2 数据库ER图

## 总体设计

|  |  |
| --- | --- |
| 表名称 | 说明 |
| admin | 管理员信息表 |
| user | 用户信息表 |
| genre | 各区分类 |
| song | 歌曲信息表 |
| collect | 收藏表 |
| comment | 评论表 |

表 1 总体设计

## 管理员表

表名：admin

登录：用户名+密码

表 2 管理员表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 说明 |
| id | int | 非空唯一 | 主键 |
| username | varchar(30) | 非空唯一 | 用户名 |
| password | varcha(50) | 非空 | 密码 |
| mobile | varchar(11) |  | 手机号 |
| email | varchar(25) |  | 邮箱 |
| enable | varchar(1) | 非空 | Y：启用 N：禁用 默认启用 |

## 用户表

用户名：user

登录：邮箱+密码

表 3 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 说明 |
| id | int | 非空唯一 | 主键 |
| mobile | varchar(11) | 唯一 |  |
| mail | varchar(25) | 非空唯一 | 邮箱最为用户登录账号 |
| password | varchar(50) | 非空 | 密码 |
| nickname | varchar(20) |  | 昵称（可以使用昵称登录） |
| avatar | varchar(300) | 非空 | 用户头像 |
| locked | char(1) | 非空 | Y：锁定 N：正常 默认正常 |

## 音乐表

表名：song

表 4 音乐表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 说明 |
| id | bigint | 非空唯一 | 主键 |
| genre\_id | int |  | 外键 分类ID |
| admin\_id | int |  | 外键 上传歌曲管理员ID |
| title | varchar(255) | 非空 | 歌曲名称 |
| source | varchar(300) | 非空 | 音乐链接 |
| poster | varchar(300) | 非空 | 音乐图片 |
| author | varchar(20) |  | 音乐作者，使用“,”间隔 |
| collections | bigint | 非空 | 音乐收藏量 |
| status | int | 非空 | 发布状态 0:未发布 1:已发布 2:已下线 |
| publish\_time | timestamp |  | 发布时间 |
| offline\_time | timestamp |  | 下线时间 |
| create\_time | timestamp | 非空 | 创建时间 |

## 分类表

表名：genre

表 5 分类表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 说明 |
| id | int | 非空唯一 | 主键 |
| name | varchar(20) | 非空唯一 | 名称 |
| desc | varchar(50) |  | 分类描述 |

## 收藏表

表名：collect

一个用户收藏多首音乐，一首歌可以被多个用户收藏

表 6 收藏表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 说明 |
| id | int | 非空唯一 | 主键 |
| song\_id | int | 非空 | 歌曲ID |
| user\_id | int | 非空 | 用户ID |

## 文件表（单独数据库）

表名：fileinfo

表 7 文件表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 说明 |
| id | int | 非空唯一 | 主键 |
| name | varchar(255) | 非空 | 文件名 |
| type | varchar(20) |  | 文件类型 |
| size | double |  | 大小 |
| uuid | varchar(32) |  | 自动生成的uuid，防止重复 |
| MD5 | varchar(255) |  | 防止重复上传 |
| path | varchar(255) |  | 文件路径 |
| createtime | date |  | 创建时间 |
| url | varchar(255) |  | 链接 |

# 后端

## Maven

使用maven自动生成webapp项目，同时使用Maven管理依赖，生成打包项目等Pom文件相关依赖等

* Junit：测试
* Log4j: 日志
* Spring与Spring MVC相关依赖
* Mybatis与Mybatis++：数据库管理
* Mysql驱动
* Servlet：处理请求，前端控制器
* Pagehelper：查询分页

## Dao层

使用Mybatis-Generator自动生成javabean和mapper文件，提供与数据库通信的接口。

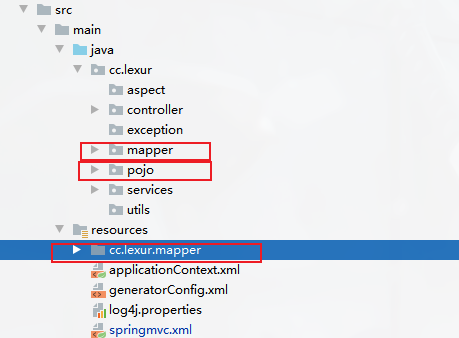


图 3 Dao层

## service层

### AdminService

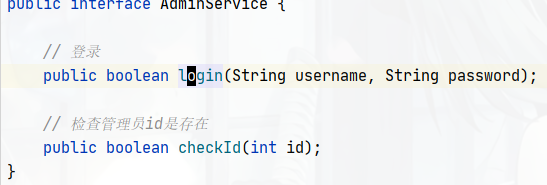


图 4 AdminService

### CollectService

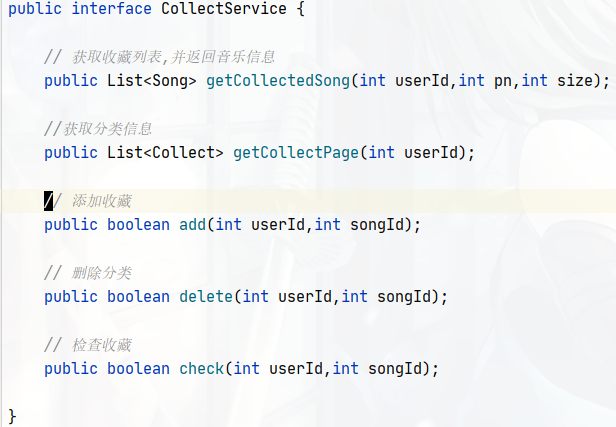


图 5 CollectService

### GenreService

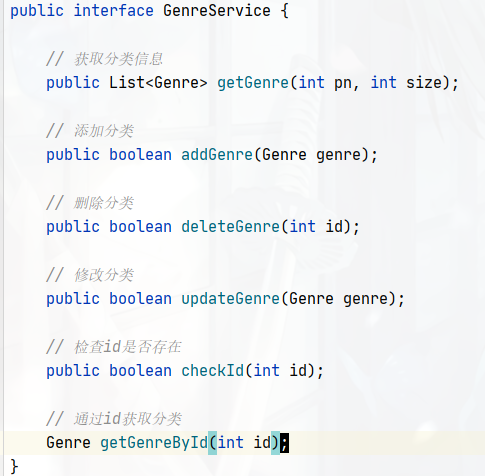


图 6 GenreService

### SongService



图 7 SongService

### UserService



图 8 UserService

## controller层

### 通用返回类

使用了jackson自动将类转换为json格式数据发送给前端

通用Msg返回类传递信息

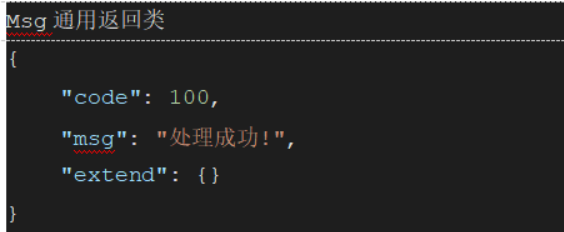


图 9 Msg传输数据

* code状态码：code=100成功，code=200为失败
* msg：存放消息
* extend：传输信息

# 接口设计

## 管理员

### 登录

请求参数

请求网址：localhost:2001/admin/login

方法：POST

表 8 登录

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| username | String | 是 | 用户名 |
| password | String | 是 | 密码 |

响应元素

说明：返回为html页面

登录失败：返回error.jsp页面，过5面后自动跳转错误页面

登录成功：返回musicweb-admin.jsp页面

### 用户信息

请求参数

请求网址：localhost:2001/admin/page

方法：GET

表 9 用户信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| pn | int | 是 | 页码 |
| size | int | 否 | 页面大小，默认8 |

响应元素

说明：返回为json数据

### 添加用户

请求参数

请求网址：localhost:2001/admin/addUser

方法：POST

表 10 2.响应元素

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| mail | String | 是 | 邮箱 |
| password | String | 是 | 密码 |
| nickname | String | 否 | 昵称 |
| phone | String | 否 | 电话 |
| avatar | String | 否 | 头像 |
| birth | Date | 否 | 生日 |

响应元素

说明：返回为json数据

【1】添加成功:code=100

【2】添加失败:code=200

### 删除用户

请求参数

请求网址：localhost:2001/admin/deleteUser

方法：GET/POST

表 11 删除用户

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| id | int | 是 | id |

响应元素

说明：返回为json数据

【1】删除成功:code=100

【2】删除失败:code=200

### 修改用户

请求参数

请求网址：localhost:2001/admin/updateUser

方法：POST

表 12 修改用户

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| id | int | 是 | id |
| mail | String | 否 | 邮箱 |
| password | String | 否 | 密码 |
| nickname | String | 否 | 昵称 |
| phone | String | 否 | 电话 |
| avatar | String | 否 | 头像 |
| birth | Date | 否 | 生日 |
| locked | Boolean | 否 | 状态 |

响应元素

说明：返回为json数据

【1】更新成功:code=100

【2】更新失败:code=200

## 分类

### 分类添加

请求参数

请求网址：localhost:2001/genre/addGenre

方法：POST

表 13 分类添加

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| name | String | 是 | 分类名称 |
| description | String | 否 | 描述 |

注意如果descrption使用缩写desc会与关键字冲突！

响应元素

说明：返回为json数据

【1】添加成功:code=100

【2】添加失败:code=200

### 分类查询

请求参数

请求网址：localhost:2001/genre/page

方法：GET

表 14 分类查询

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| pn | int | 是 | 页码 |
| size | int | 否 | 页面大小 |

响应元素

说明：返回为json数据

【1】成功:code=100

### 分类删除

请求参数

请求网址：localhost:2001/genre/deleteGenre

方法：GET

表 15 分类删除

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| id | int | 是 | id |

响应元素

说明：返回为json数据

【1】添加成功:code=100

【2】添加失败:code=200

### 分类修改

请求参数

请求网址：localhost:2001/genre/updateGenre

方法：POST表 16 分类修改

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| id | int | 是 | id |
| name | String | 否 | 分类名称 |
| description | String | 否 | 描述 |

响应元素

说明：返回为json数据

【1】添加成功:code=100

【2】添加失败:code=200

## 音乐管理

### 音乐信息

请求参数

请求网址：localhost:2001/song/page

方法：POST

表 17 音乐信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| pn | int | 否 | 页码 |
| size | int | 否 | 页面大小 |

响应元素

说明：返回为json数据

### 添加音乐

请求参数

请求网址：localhost:2001/song/addSong

方法：POST

表 18 添加音乐

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| genre\_id | int | 是 | 分类id |
| admin\_id | int | 是 | 上传人员id |
| title | String | 是 | 标题 |
| language | String | 否 | 语言 |
| source | String | 是 | 歌曲链接 |
| poster | String | 是 | 歌曲封面 |
| author | String | 否 | 歌曲作者 |

响应元素

说明：返回为json数据

### 删除音乐

请求参数

请求网址：localhost:2001/song/deleteSong

方法：POST

表 19 删除音乐

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| id | int | 是 | id |

响应元素

说明：返回为json数据

### 修改音乐

请求参数

请求网址：localhost:2001/song/updateSong

方法：POST

表 20 修改音乐

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| id | int | 是 | 修改音乐id |
| genre\_id | int | 否 | 分类id |
| title | String | 否 | 标题 |
| language | String | 否 | 语言 |
| source | String | 否 | 歌曲链接 |
| poster | String | 否 | 歌曲封面 |
| author | String | 否 | 歌曲作者 |
| status | int | 否 | 歌曲状态 |

响应元素

说明：返回为json数据

### 按名称查询

请求参数

请求网址：localhost:2001/song/title

方法：GET

表 21 按名称查询

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| title | String | 是 | 音乐名称 |

响应元素

说明：返回为json数据

### 按分类查询

请求参数

请求网址：localhost:2001/song/genre

方法：GET

表 22 按分类查询

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| id | int | 是 | 分类名称 |

响应元素

说明：返回为json数据

## 用户

### 登录

请求参数

请求网址：localhost:2001/user/login

方法：POST

表 23 按分类查询

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| mail | String | 是 | 邮箱 |
| password | String | 是 | 密码 |

响应元素

说明：返回为jsp页面

【1】登录成功:music.jsp

【2】登录失败:error.jsp，5秒后返回登录页面

### 注册

请求参数

请求网址：localhost:2001/user/register

方法：POST表 24 注册

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| mail | String | 是 | 邮箱 |
| password | String | 是 | 密码 |
| nickname | String | 否 | 昵称 |
| phone | String | 否 | 电话 |
| avatar | String | 否 | 头像 |
| birth | Date | 否 | 生日 |

### 检测邮箱重复

请求参数

请求网址：localhost:2001/user/checkMail

方法：POST表 25 检测邮箱重复

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| mail | String | 是 | 邮箱 |

响应元素

说明：返回为json

## 收藏/取消收藏

### 取消收藏

请求参数

请求网址：localhost:2001/collect/delete

方法：GET表 26 取消收藏

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| songId | String | 是 | 音乐id |
| userId | String | 是 | 用户id |

响应元素

说明：返回为json数据

### 收藏

请求参数

请求网址：localhost:2001/collect/add

方法：GET

表 27 收藏

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| songId | String | 是 | 音乐id |
| userId | String | 是 | 用户id |

响应元素

说明：返回为json数据

### 展示收藏

请求参数

请求网址：localhost:2001/collect/show

方法：GET

表 28 展示收藏

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| userId | String | 是 | 用户id |

响应元素

说明：返回为json数据

# 前端

## 原理

网页端向服务器返送ajax请求获取json数据后渲染页面

## 使用技术

前端搭建使用了jsp,html,css,bootstrap快速搭建

使用js与jquery对网页的内容进行动态加载

## 界面展示

### 用户登录

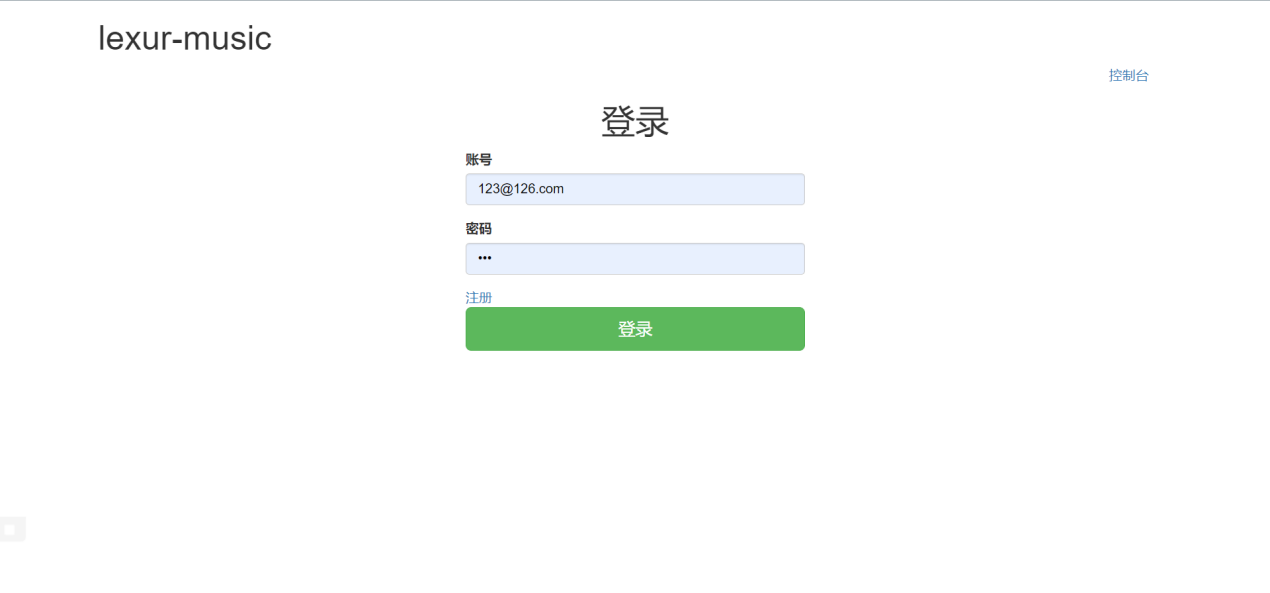


图 10 用户登录

### 用户注册

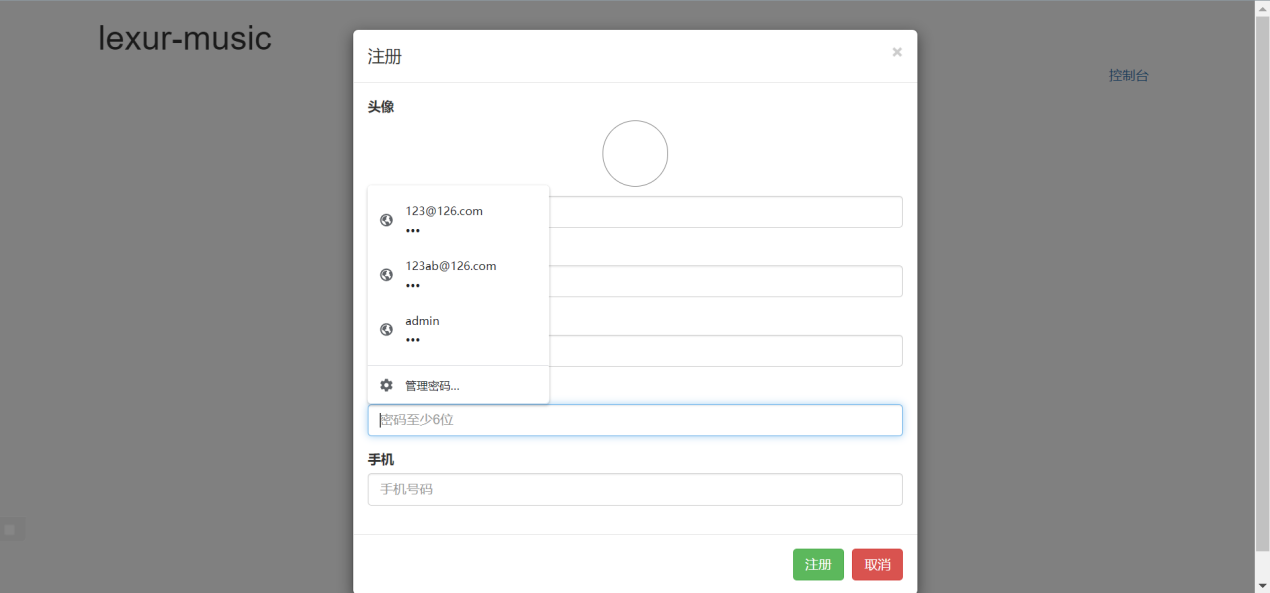


图 11 用户注册

### 音乐网站主页

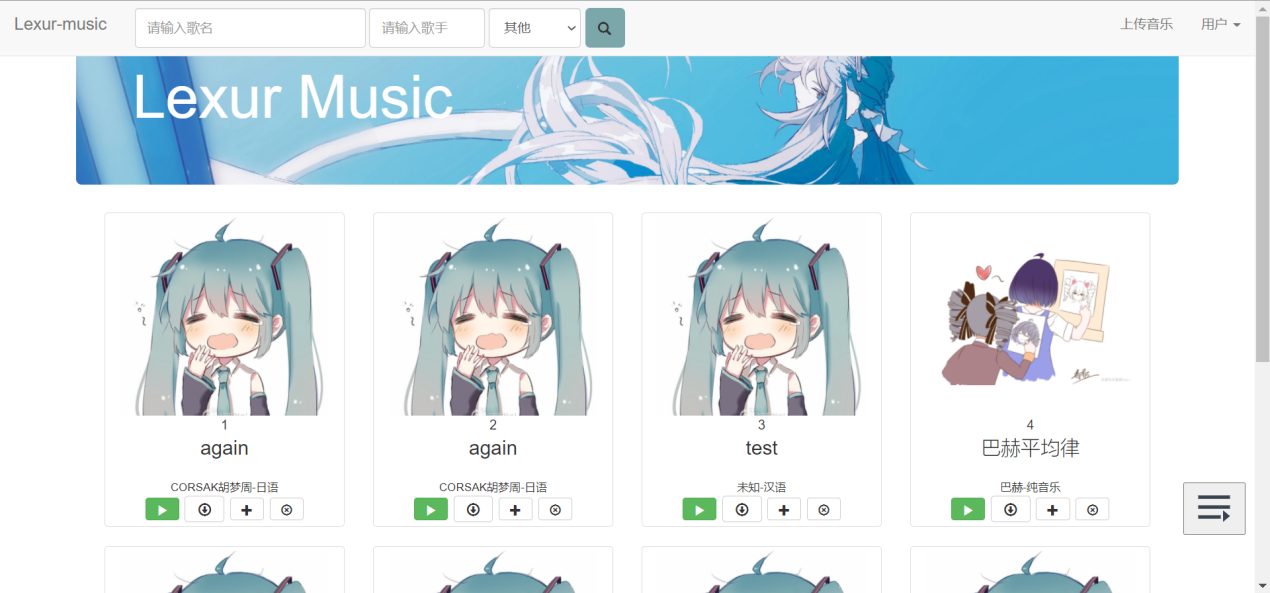


图 12 音乐网站主页

### 音乐播放器

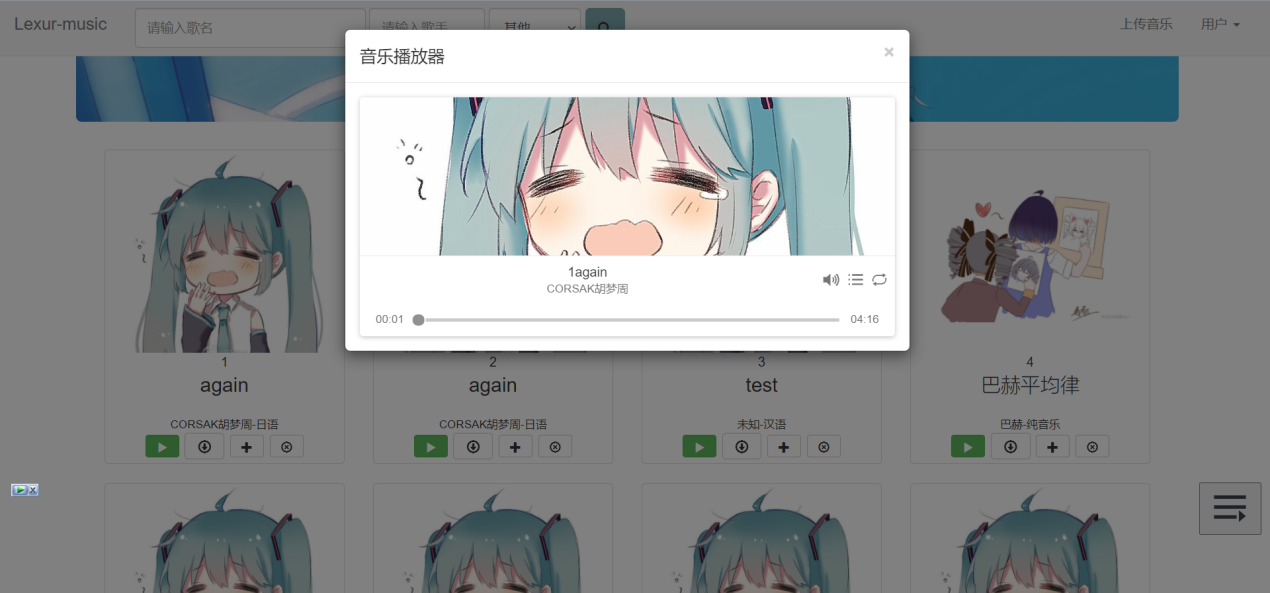


图 13 音乐播放器

### 上传音乐



图 14 上传音乐

### 搜索音乐

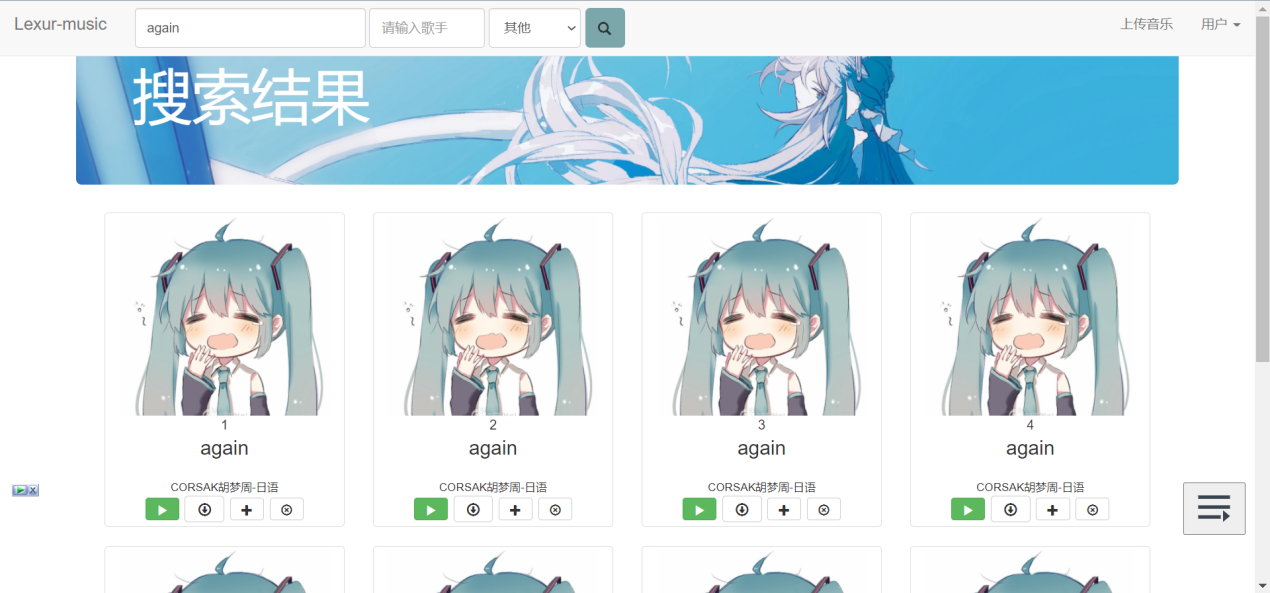


图 15 搜索音乐

### 管理员登录界面



图 16 管理员登录界面

### 管理用户界面



图 17 管理用户界面

### 管理音乐界面

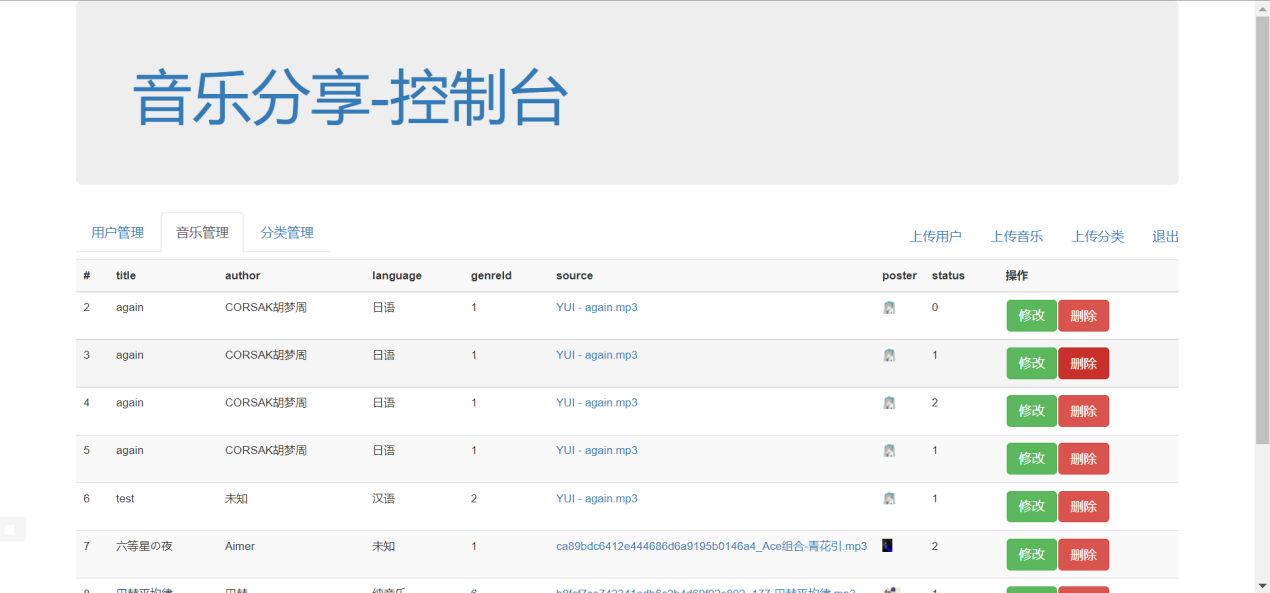


图 18 管理音乐界面

### 管理分类界面



图 19 管理分类界面

### 管理员添加用户、音乐、分类

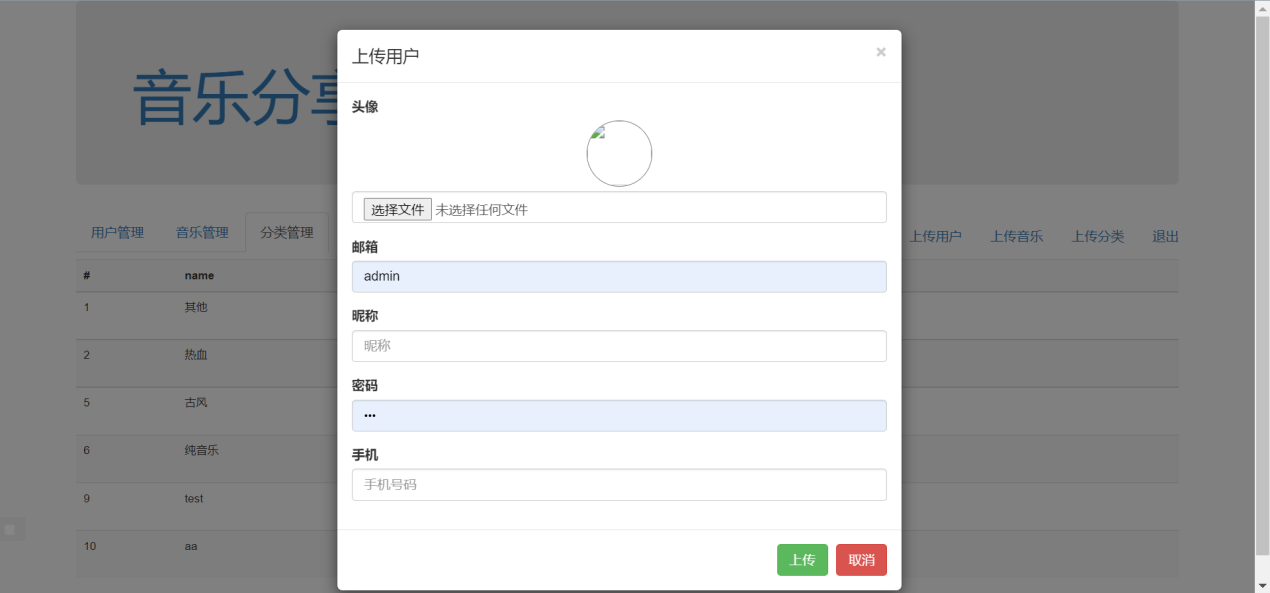


图 20 添加用户



图 21 添加音乐

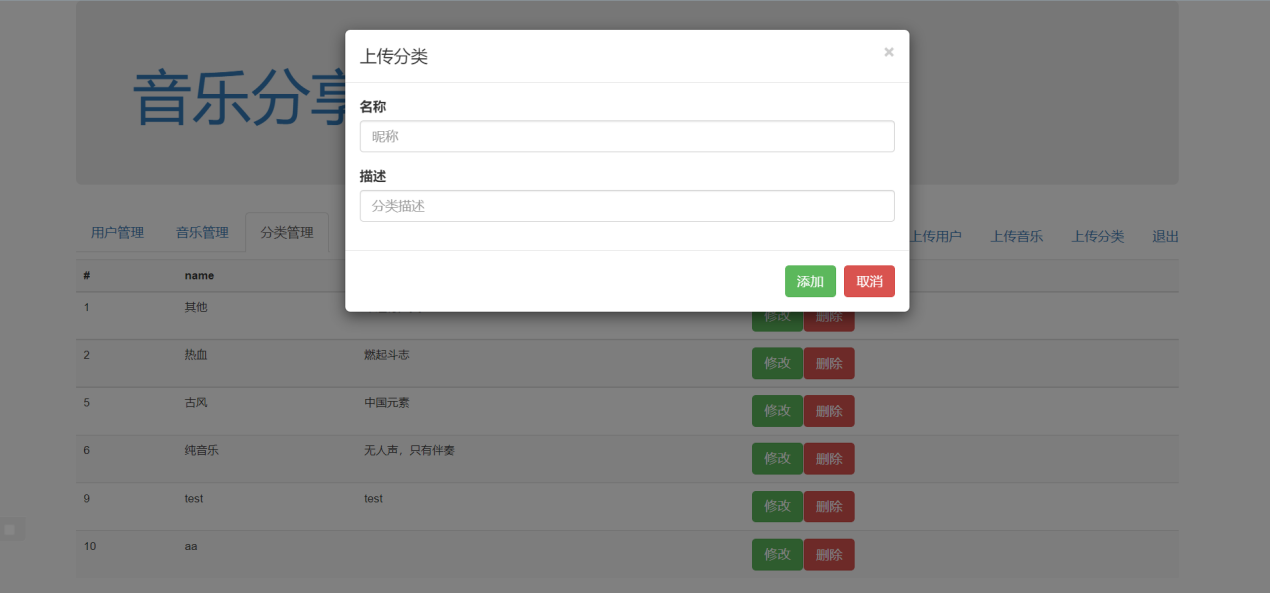


图 22 添加分类

# 未解决问题

* 未能将war包部署到服务器，服务器的数据库无法通过Navicat管理
* 未能实现评论功能
* 未对传输密码与文件加密
* 链接格式不够统一
* 可以将网站做成前后端分离，这样可以减轻主服务器的压力

# 附录

JQuery中文文档：<https://jquery.cuishifeng.cn/>

MyBatis-Plus：<https://baomidou.com/>

CPlyaer：<https://cplayer.js.org/>

尚硅谷SSM：<https://www.bilibili.com/video/BV17W411g7zP>

SSM框架合集：<https://www.bilibili.com/video/BV1mE411X7yp>