南方IT学院

**学 生 毕 业 论 文**

**题 目 基于SSM的音乐播放系统的设计与实现**

**学 号：** 197370827100

**专 业：** 软件工程

**学生姓名：** 谢敏

**指导教师：**

年 月

**诚 信 声 明**

我声明，所呈交的毕业论文是本人在老师指导下进行的研究工作及取得的研究成果。据我查证，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得其他教育机构的学位或证书而使用过的材料。我承诺，论文中的所有内容均真实、可信。

毕业论文作者签名：谢敏 签名日期： 年 月 日

**论文摘要**

随着科技与社会的迅速发展，人们的工作和生活的压力不断增大，许多人因为压力过大而得不到缓解影响生活与工作，而音乐是一种简单、方便又成本低的大众喜爱的缓解压力的方式。随着互联网的迅速发展，音乐网站逐渐渗透到互联网中，并且普及都人们的日常生活当中,为了带给用户有一种方便、易操作更好的缓解压力的方式，本系统便应运而生。

本系统作为一款娱乐休闲的网站，主要目的是放松人们的心情、缓解压力而设计的，使用HTML5、javascript、jquery作为网页前端开发语言和java作为后台服务器开发语言，前端为了美观部分页面使用了layui、bootstrap等前辈们开发的前端优秀流行框架，使用数据库是mySql,前端与后端开发工具都为IDEA,主要实现在线听歌、观看mv、浏览最新歌曲咨询、将歌曲收藏到歌单、歌曲评论、浏览歌曲排行榜等基本功能。基本上满足了用户对听音乐的需求，给广大用户带来一种简单的娱乐方式。

**关键词：**音乐播放；MV播放；用户自定义歌单；歌曲评论；歌曲管理；权限管理

# 目录

[1 引言 1](#_Toc34833100)

[1.1背景 1](#_Toc34833101)

[1.2 研究现状与发展趋势 1](#_Toc34833102)

[1.3 现实意义 1](#_Toc34833103)

[1.4 系统概述 1](#_Toc34833104)

[1.5 开发环境介绍 2](#_Toc34833105)

[1.5.1 运行环境 2](#_Toc34833106)

[1.5.2 开发工具 2](#_Toc34833107)

[2 可行性分析 3](#_Toc34833108)

[2.1 问题定义 3](#_Toc34833109)

[2.2 技术可行性分析 3](#_Toc34833110)

[2.3 经济可行性分析 3](#_Toc34833111)

[2.4 操作可行性分析 3](#_Toc34833112)

[3 需求分析 4](#_Toc34833113)

[3.1 功能性需求 4](#_Toc34833114)

[3.1.1 用户手机验证码注册功能 4](#_Toc34833115)

[3.1.2 登录功能 4](#_Toc34833116)

[3.1.3 注销功能 4](#_Toc34833117)

[3.1.4 用户个人信息管理功能 4](#_Toc34833118)

[3.1.5 模糊查询功能 4](#_Toc34833119)

[3.1.6图片轮播显示最新歌曲 4](#_Toc34833120)

[3.1.7 用户在线听音乐 4](#_Toc34833121)

[3.1.8 在线播放mv 5](#_Toc34833122)

[3.1.9 用户收藏歌曲 5](#_Toc34833123)

[3.1.10 评论功能 5](#_Toc34833124)

[3.1.11 评论点赞功能 5](#_Toc34833125)

[3.1.12 热歌排行榜 5](#_Toc34833126)

[3.13 用户自定义歌单 5](#_Toc34833127)

[3.1.14 添加歌曲到自定义歌单 5](#_Toc34833128)

[3.1.15 用户收藏管理 5](#_Toc34833129)

[3.1.16 评论评论功能 5](#_Toc34833130)

[3.1.17 歌曲管理 5](#_Toc34833131)

[3.1.18 mv管理 5](#_Toc34833132)

[3.1.19 歌手管理 5](#_Toc34833133)

[3.1.20 用户权限管理 5](#_Toc34833134)

[3.2 非功能性需求 5](#_Toc34833135)

[3.2.1 界面需求 6](#_Toc34833136)

[3.2.2 安全性需求 6](#_Toc34833137)

[3.2.3 数据库需求 6](#_Toc34833138)

[4 概要设计 7](#_Toc34833139)

[4.1 概述 7](#_Toc34833140)

[4.2 技术采用 7](#_Toc34833141)

[4.3 系统总体设计 7](#_Toc34833142)

[5 数据库设计 8](#_Toc34833143)

[5.1 数据库概述 8](#_Toc34833144)

[5.2 数据表介绍 8](#_Toc34833145)

[5.3 数据字典 9](#_Toc34833146)

[6 系统实现 14](#_Toc34833147)

[6.1 用户注册 14](#_Toc34833148)

[6.2 用户首页 14](#_Toc34833149)

[6.4 登陆 15](#_Toc34833150)

[6.5 用户注销 16](#_Toc34833151)

[6.6 歌手页面 16](#_Toc34833152)

[6.7 排行榜 17](#_Toc34833153)

[6.8 用户个人信息管理功能 18](#_Toc34833154)

[6.9 评论功能 19](#_Toc34833155)

[6.10 搜索功能 21](#_Toc34833156)

[6.11 在线听音乐 21](#_Toc34833157)

[6.12 收藏（取消收藏）歌曲功能 22](#_Toc34833158)

[6.13 用户“我的音乐“的“我喜欢”模块 22](#_Toc34833159)

[6.14 用户“我的音乐“的“歌单管理”模块 23](#_Toc34833160)

[6.15 用户下载歌曲、MV 23](#_Toc34833161)

[6.16 管理员主界面 24](#_Toc34833162)

[6.17 歌曲管理 24](#_Toc34833163)

[6.18 mv管理 24](#_Toc34833164)

[6.19 歌手管理 25](#_Toc34833165)

[6.20 用户权限管理 25](#_Toc34833166)

[7 系统测试 27](#_Toc34833167)

[7.1 测试计划 27](#_Toc34833168)

[7.1.1 测试目标 27](#_Toc34833169)

[7.1.2 测试环境 27](#_Toc34833170)

[7.2 测试方法 27](#_Toc34833171)

[7.3 测试用例 27](#_Toc34833172)

[7.3.1 功能测试 27](#_Toc34833173)

[7.3.2 界面测试 29](#_Toc34833174)

[7.4 测试结论 29](#_Toc34833175)

[8 总结 30](#_Toc34833176)

[参考文献 31](#_Toc34833177)

[论文评定表 32](#_Toc34833178)

# 1 引言

在这计算机网络的蓬勃发展、信息高度发达的现代社会，互联网在当今社会的传播速度正以指数的方式增长，通过网络，我们可以收听自己喜欢的音乐，与相隔千里的其他用户进行评论互动，互联网在我们生活中扮演者不可缺失的重要角色，本章对用户听音乐的需求进行了初步的探讨，同时介绍了系统开发的目标、运行环境以及主要意义在于提高用户的音乐体验。

## 背景

在当今时代，互联网的发展迅速逐渐体现在人们的日常生活中，足不出户就能了解外面的信息极大的方便了人们的日常生活，但随着社会的迅速发展也带来的巨大的生活压力和工作压力，音乐恰好是一个简单而且又成本小受人们欢迎的一种排遣方式，互联网相比于传统的下载歌曲到mp3等设备中才能听歌的方式具有很大优势，只需要进入网站就能在线听歌。因此，网页播放器就有了巨大的前景。

## 1.2 研究现状与发展趋势

随着科技与互联网的发展，音乐播放器在国内外都有了不少的例子，计算机互联网技术的迅速发展，网站也因此贴近人们的日常生活，时间、可操作性和成本在网站上体现可巨大的优势，互联网作为一种对于客户端基本零负担的形式，用户不需要下载到设备中才可以进行听歌，打开网站就可以收听自己喜欢的歌曲，与相隔千里的其他用户进行互动，因此，在线的网页音乐播放器就有巨大的发挥空间。

## 1.3 现实意义

随着科技与互联网的迅速发展，电脑已经成为人们日常工作生活的常见工具，此音乐播放器能方便用户更简单的了解音乐信息和听歌曲，愉悦心情。

## 1.4 系统概述

主要针对的是广大用户和需要发布音乐信息的管理员的设计的，基本符合他们所要用到的功能。目前，本设计应解决的主要的功能有三个:

1. 用户在线音乐功能模块

用户能够随时了解音乐和歌手的信息、在线听歌、观看mv、用户自定义歌单

1. 歌曲的管理

管理员可以对歌曲、mv、歌手等进行添加、删除、修改等操作

1. 用户权限的管理

管理员可以修改用户的登录身份、如将“用户”修改为级别更高的“管理员”，这样该用户就可以登录到后台进行歌曲的管理等高级功能。

## 1.5 开发与运行环境介绍

### 1.5.1 运行环境

硬件环境：

CPU:2.30GHZ或者更高

内存：4G或更多

显示器分辨率：1366\*768或者更高

系统类型：64位操作系统

软件环境：

操作系统：windows 10

### 1.5.2 开发工具

后台：idea 前端：idea、HBuilder 数据库：mysql Java环境：JDK1.8

# 2 可行性分析

## 2.1 问题定义

系统面向的使用者是广大群众，而管理者面向的是任何想发布音乐、管理、推广的公司或个人，因此系统的使用和维护都不能过于复杂，造成用户使用困难或者管理员系统维护困难的问题，应该符合实际要求，开发出操作简单、界面较美观、可维护的系统。

## 2.2 技术可行性分析

系统采用前后端分离的模式进行设计的，这样是系统后期维护工作变得更简单，前端使用javascript、jquery等语言，后端使用ssm进行架构，使用DruidDataSource框架作为与mysql数据库连接的管理工具，使用的技术都是较为基础的，使得系统的维护是简单可行的。

## 2.3 经济可行性分析

本系统的运行只需要一台性能一般的电脑就可以运行了，系统环境也是符合大众的windows界面，数据库也是不需要成本的Mysql，无论是一般的公司还是个人，都是能负担的起的。

## 2.4 操作可行性分析

操作界面简单易懂，用户可以很轻松的掌握，而设计简单也让管理员后期维护系统变得简单可行。

# 3 需求分析

在系统的开发中需求分析是必不可少的一项工作，系统开发的最终目的是满足用户的需求，如果开发出来的软件脱离实际，不符合逻辑，或者根本不满足用户需求，那软件的开发就没有了实际意义。所以需求分析是必不可少的。

## 3.1 功能性需求

由于该软件作为一个娱乐休闲软件，无论年龄、职业、性别限制都可以使用，为满足娱乐休闲的目的，需要针对用户需求实现相应的功能。功能按登陆类型进行了区分。具体如下：

### 3.1.1 用户手机验证码注册功能

为了防止有些用户恶意注册，注册使用了按手机号注册用户的策略，每个手机号只能对应注册一个用户，另外，防止用户使用非本人手机注册，采用了短信验证码的方法。

系统有两种身份，一种是用户，一种是管理员，在登陆界面输入账号和密码，登陆时系统根据账号密码识别该账号的身份类型，进入对应的主界面。同时还设定有注销功能，可以实现用户的切换。

### 3.1.2 登录功能

登陆功能是为了区分用户和管理员两种身份，方便对系统的管理和运行，同时每个用户都有独立的账号方便实现如“我的歌单”和“对歌曲进行评论”等这样的需要区分独立每个用户的功能。管理员则对歌曲、mv、歌手等信息进行管理

### 3.1.3 注销功能

用户登录后在主页右上方显示用户的头像，鼠标悬停显示一个列表，点击“注选项，可以完成登陆注销。

### 3.1.4 用户个人信息管理功能

每个用户对应有关的部分个人信息可以自行修改，如昵称、性别等这种权限较低级的信息

### 3.1.5 模糊查询功能

当数据很庞大时，用户手动一个一个找是很不现实的，所以查询功能是必不可少的，在首页上房的搜索框中输入文本，无论是歌曲、mv、还是歌手含有输入的文本都可以被查询出来，如输入 大约在冬季 就能查找歌曲名、mv、歌手包含“大约在冬季“的数据。

### 3.1.6图片轮播显示最新歌曲

进入网站首页，有一个图片轮播模块，显示前5个发布时间最接近的歌曲，方便用户了解最新歌曲信息。也可以点击播放该歌曲。

### 3.1.7 用户在线听音乐

用户不需要进行登录就可以在线听音乐及浏览歌曲、歌手等信息。

### 3.1.8 在线播放mv

用户不需要登陆就可以在线播放mv及浏览mv信息。

### 3.1.9 用户收藏歌曲

用户听歌时可以点击❤收藏歌曲，将歌曲添加到“我喜欢”模块中。

### 3.1.10 评论功能

当用户听歌曲的时候，在播放器右端有一个评论区，用户可以点击进入，浏览其他用户的评论或者自己进行评论。

### 3.1.11 评论点赞功能

用户可以对其他用户的评论进行点赞，通过点赞数优先显示热门评论，这样好的评论便可以让更多用户便捷的浏览。

### 3.1.12 热歌排行榜

这里可以查看按播放次数降序的歌曲榜单。

### 3.13 用户自定义歌单

用户可以添加自定义歌单，将歌曲添加到自定义歌单中，也可以删除、修改歌单信息。

### 3.1.14 添加歌曲到自定义歌单

用户除了可以收藏歌曲，也可以添加曲到自定义歌单中去。

### 3.1.15 用户收藏管理

用户登录后才可进入“我的音乐”模块，“我喜欢“部分为用户收藏的歌曲、专辑、歌单、mv，可以点击右端的删除按钮，移除收藏

### 3.1.16 评论评论功能

用户可以浏览评论，那就必不可少的有对其他用户的评论进行评论的功能，让用户之间有更多的交流。

### 3.1.17 歌曲管理

管理员具有上传歌曲、下架歌曲、修改歌曲信息等权限较高的功能权限。

### 3.1.18 mv管理

管理员具有为歌曲添加mv、下架mv、修改mv信息等功能权限

### 3.1.19 歌手管理

管理员具有添加歌手、删除歌手、修改歌手信息等权限较高的功能权限。

### 3.1.20 用户权限管理

管理员具有可以改变用户的登录身份，升级为管理员或改变为用户。

## 3.2 非功能性需求

作为一款休闲娱乐的软件，除了具备应该有的功能外，还应该有较简洁易懂的操作流程，界面较友好等这些能影响用户体验的非功能因素。

**3.2.1 界面需求**

因为系统面向的使用者是广大群众，不限年龄，性别，因此对计算机的了解程度会差异很大，如果操作过于复杂，一些没有计算机基础的用户就可能使用困难，这样就不满足作为一款娱乐软件的宗旨了。因此界面必须要简介明了、美观，用户一看就明白，同时不会因为界面丑陋而影响用户体验，同时功能应该操作简单，让用户轻松就能掌握上手。

**3.2.2 安全性需求**

系统应该有良好的权限管理机制，如分用户和管理员两种登陆类型，防止有些用户对网站数据进行恶意操作，使系统不能正常运作。

**3.2.3 数据库需求**

任何系统的运作的核心就是数据，而数据库就是存储数据的仓库，数据库的使用是必须的，系统能否正常运转很大部分是依赖于数据库的，所以选择一个较合适的数据库是很重要的，本系统选择mysql数据库，该数据库作为一个较流行的数据库，功能方面也叫完善，并且也不收费，适用于一般公司或个人作为开发系统的选择。

数据设计是根据不同的实体进行设计的。如：

用户信息：姓名、出生日期、性别等。

歌曲信息：歌曲名、歌手、时长、歌曲曲风等

歌手信息：姓名、所属地等

歌单信息：歌单名、所属用户等

Mv信息：mv名字、mv时长等

通过这些庞大的实体，构建相对应的联系，让业务更精准的实现和维护系统。

# 4 概要设计

## 4.1 概述

本系统使用前后端分离模式开发，面向对象的体系结构设计，其主要用途是为广大群众提供在线听音乐的便利和娱乐休闲的享受，给有需求的公司、企业、团队等需要向外发布音乐的单位提供渠道。主要功能包括权限登录、在线听音乐、播放mv、评论、评论点赞、收藏歌曲、歌曲管理、用户管理等功能。

## 4.2 技术采用

**开发语言：**

**Java**:java是一门面向对象编程语言，不仅吸收了c++语言的各种有点，还摒弃了c++里难以理解的多继承、指针等概念，具有功能强大和简单易用的两个特征。

**Sql**：结构化查询语言(Structured Query Language)简称Sql，是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系型数据库。

**Html**:超文本标记语言，是一种标识性的语言。

**js**: JavaScript是一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言

**jquery**: jQuery是一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的JavaScript代码库

**框架：**

**Mybatis**: MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。

**Spring**: Spring框架是一个开放源代码的J2EE应用程序框架，提供了功能强大IOC、ADP及Web MVC等功能。

## 4.3 系统总体设计

本系统主要由以下几个功能模块组成;

登录功能：用户、管理员可以通过登陆分别进入歌曲首页和歌曲管理页面。

在线听音乐功能：用户可以在线听音乐。

评论功能： 用户可以在评论区进行评论，点赞，和评论其他用户的评论。

播放mv功能：用户可以观看mv。

歌曲收藏功能：用户可以在听音乐时点击收藏添加到“我喜欢”的歌单中。

歌单管理功能：用户可以创建歌单、删除歌单、添加歌曲到歌单等。

歌曲管理功能：管理员可以对歌曲歌曲进行管理。

用户管理功能：管理员可以对用户进行管理。

# 

# 5 数据库设计

根据以上功能需求，本系统的数据库应当符合上述需求，满足用户和管理员的信息要求和处理要求。

## 5.1 数据库概述

本系统使用的是mysql数据库，是系统设计时最常用的一种开放源代码的关系型数据库，是一种可以处理成千上万条数据记录的大型数据库，它因为速度性、可靠性和适应性而备受欢迎。其次它提供了多线程的运行模式， 完全支持Java、php、python等编程语言的接口使用且提供了多种数据库的连接方式如JDBC、ODBC、TCP/IP等，更重要的是他优化了SQL的查询算法，支持数据库的范式（1NF（第一范式），2NF（第二范式），3NF（第三范式）和BCNF），有效的提高了查询速度和效果。

优点：

1.mysql可运行在不同的操作系统下，几乎是免费的，支持SQL语句对数据的各种详细的查询

2.稳定性好，足以支撑一个超大规模的数据库。

3.他有一个灵活且非常安全的权限和口令系统，客户与服务器之间的连接，所有口令都是被加密的。

缺点:他不完全支持不熟悉的关键词，并且缺乏一些存储程序的功能，更重要的是使用缺省的ip端口，有时候这些ip也会被一些黑客闯入。

## 5.2 数据表介绍

根据分析所设计的数据库所有表的名称，如表5.2.1所示。

表5. 2.1 数据库表总表

|  |  |
| --- | --- |
| 用户信息表 | user |
| 登陆信息表 | login |
| 我喜欢信息表 | ilike\_info |
| 歌单信息表 | song\_sheet |
| 歌手信息表 | singer |
| 专辑信息表 | album |
| 歌曲信息表 | song |
| 歌单歌曲关系表 | sheet\_song\_relation |
| mv信息表 | mv |
| 歌曲我喜欢关系表 | song\_ilike\_relation |
| 歌单我喜欢关系表 | sheet\_ilike\_relation |
| 专辑我喜欢关系表 | album\_ilike\_relation |
| mv我喜欢关系表 | mv\_ilike\_relation |
| 歌曲评论信息表 | song\_comment |
| 评论点赞信息表 | iike\_comment |

## 5.3 数据字典

表5.3.1 用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| user\_id | int | NOT NULL | 用户编号 |
| user\_name | varchar(20) | NOT NULL | 姓名 |
| user\_sex | char(2) | NOT NULL | 用户性别 |
| user\_phone | char(11) | NOT NULL | 电话号码 |
| user\_picture | varchar(200) | 用户头像 | 用户头像 |

用户信息表：此表是用来记录用户的基本信息，相应字段如表5.3.1所示。

表5.3.2登录信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| login\_id | int | NOT NULL | 编号 |
| user\_id | int | NOT NULL | 用户编号 |
| login\_name | varchar(8) | NOT NULL | 登录用户名 |
| password | varchar(16) | NOT NULL | 密码 |
| login\_type | varchar(6) | NOT NULL | 登陆类型 |

登陆信息表：此表是用来记录用户的登陆账号和身份类型，相应字段如表5.3.2所示。

表5.3.3我喜欢信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| id | int | NOT NULL | 主键、自增 |
| user\_id | int | NOT NULL | 外键user(user\_id) |

表5.3.4歌单信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| sheet\_id | int | NOT NULL | 歌单编号 |
| sheet\_name | varchar(20) | NOT NULL | 歌单名字 |
| user\_id | int | NOT NULL | 所属用户 |
| language | varchar(50) |  | 语种标签 |
| style | varchar(50) |  | 风格标签 |
| mood | varchar(50) |  | 心情标签 |

歌单信息表：此表是用来记录歌手的基本信息，相应字段如表5.3.4所示。

表5.3.5歌手信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| singer\_id | int | NOT NULL | 歌手编号 |
| singer\_name | varchar(20) | NOT NULL | 歌手名字 |
| singer\_sex | char(2) | NOT NULL | 歌手性别 |
| singer\_area | varchar(20) | NOT NULL | 出生地 |
| singer\_birth | date | NOT NULL | 出生日期 |
| singer\_country | varchar(50) | NOT NULL | 国籍 |
| singer\_height | int | NOT NULL | 身高 |
| singer\_weight | int | NOT NULL | 体重 |
| blood\_type | varchar(10) | NOT NULL | 血型 |
| constellation | varchar(6) | NOT NULL | 星座 |
| synopsis | varchar(1000) | NOT NULL | 简介 |
| singer\_picture | varchar(200) | NOT NULL | 歌手头像 |
| fans | int | NOT NULL | 粉丝数量 |
| singer\_name\_initials | char(2) | NOT NULL | 歌手名字首字母 |

表5.3.6专辑信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| album\_id | int | NOT NULL | 专辑编号 |
| album\_name | varchar(20) | NOT NULL | 专辑名字 |
| singer\_id | int | NOT NULL | 专辑所属歌手编号 |
| publish\_date | date | NOT NULL | 专辑发行时间 |

歌手信息表：此表是用来记录歌曲的基本信息，相应字段如表5.3.6所示。

表5.3.7歌曲信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| song\_id | int | NOT NULL | 歌曲编号 |
| song\_name | varchar(20) | NOT NULL | 歌曲名字 |
| singer\_id | int | NOT NULL | 所属歌手 |
| song\_language | varchar(20) | NOT NULL | 歌曲语种 |
| publish\_date | date | NOT NULL | 歌曲发布时间 |
| song\_click | int | NOT NULL | 歌曲热度(点击量) |
| song\_src | varchar(200) | NOT NULL | 歌曲路径 |
| picture\_src | varchar(200) | NOT NULL | 歌曲图片 |
| song\_length | int | NOT NULL | 歌曲时长 |
| album\_id | int | NOT NULL | 歌曲所属专辑编号 |
| lylic\_src | varchar(200) | NOT NULL | 歌词路径 |

歌曲信息表：此表是用来记录歌曲的基本信息，相应字段如表5.3.7所示。

表5.3.8歌单歌曲关系信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| id | int | NOT NULL | 关系编号 |
| song\_sheet\_id | int | NOT NULL | 歌单编号 |
| song\_id | int | NOT NULL | 歌曲编号 |

表5.3.9 mv信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 备注 |
| mv\_id | int | NOT NULL | mv编号 |
| mv\_picture | varchar(200) | NOT NULL | mv图片 |
| mv\_src | varchar(200) | NOT NULL | mv路径 |
| mv\_click | int | NOT NULL | mv热度(点击量) |
| song\_id | int | NOT NULL | mv所属歌曲 |
| mv\_length | int | NOT NULL | mv时长 |
| mv\_name | varcbar(20) | NOT NULL | mv名字 |

表5.3.10 歌曲我喜欢信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| id | int | NOT NULL | 关系编号 |
| song\_id | int | NOT NULL | 歌曲编号 |
| ilike\_id | int | NOT NULL | 我喜欢编号 |

表5.3.11歌单我喜欢关系信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| id | int | NOT NULL | 关系编号 |
| sheet\_id | int | NOT NULL | 歌单编号 |
| ilike\_id | int | NOT NULL | 我喜欢编号 |

表5.3.12专辑我喜欢关系表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| id | int | NOT NULL | 关系编号 |
| album\_id | int | NOT NULL | 歌曲编号 |
| ilike\_id | int | NOT NULL | 我喜欢编号 |

表5.3.13mv我喜欢关系表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| id | int | NOT NULL | 关系编号 |
| mv\_id | int | NOT NULL | mv编号 |
| ilike\_id | int | NOT NULL | 我喜欢编号 |

表5.3.14歌曲评论信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 备注 |
| scomment\_id | int | NOT NULL | 评论编号 |
| song\_id | int | NOT NULL | 歌曲编号 |
| user\_id | int | NOT NULL | 用户编号 |
| scomment\_click | int | NOT NULL | 点赞数 |
| recomment\_id | int | NULL | 回复的评论id |
| content | varchar(200) | NOT NULL | 评论内容 |
| comment\_time | dateTime | NOT NULL | 评论时间 |

表5.3.15评论点赞信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可否为空 | 备注 |
| id | int | NOT NULL | 编号 |
| user\_id | int | NOT NULL | 用户编号 |
| scomment\_id | int | NOT NULL | 被点赞的评论编号 |

# 6 系统实现

## 6.1 用户注册

这个功能是由用户自行注册，注册得到的登陆身份只能是“用户”，为防止恶意注册，注册需使用手机短信验证码的方式进行注册，并且手机验证码发送完后必须60s后才能再次发送，这种登陆类型的账号登陆只能进入用户主界面。



图6.1 用户注册

## 6.2 用户首页

这个界面是不需要登陆就直接进入的页面，但并不是所有功能都能在未登陆的情况下使用，但有些功能是可以使用的如查看歌曲、歌手、排行榜、播放mv不需要进行登录就可以使用，但也有一些功能是需要登陆的，例如“我的音乐模块”，

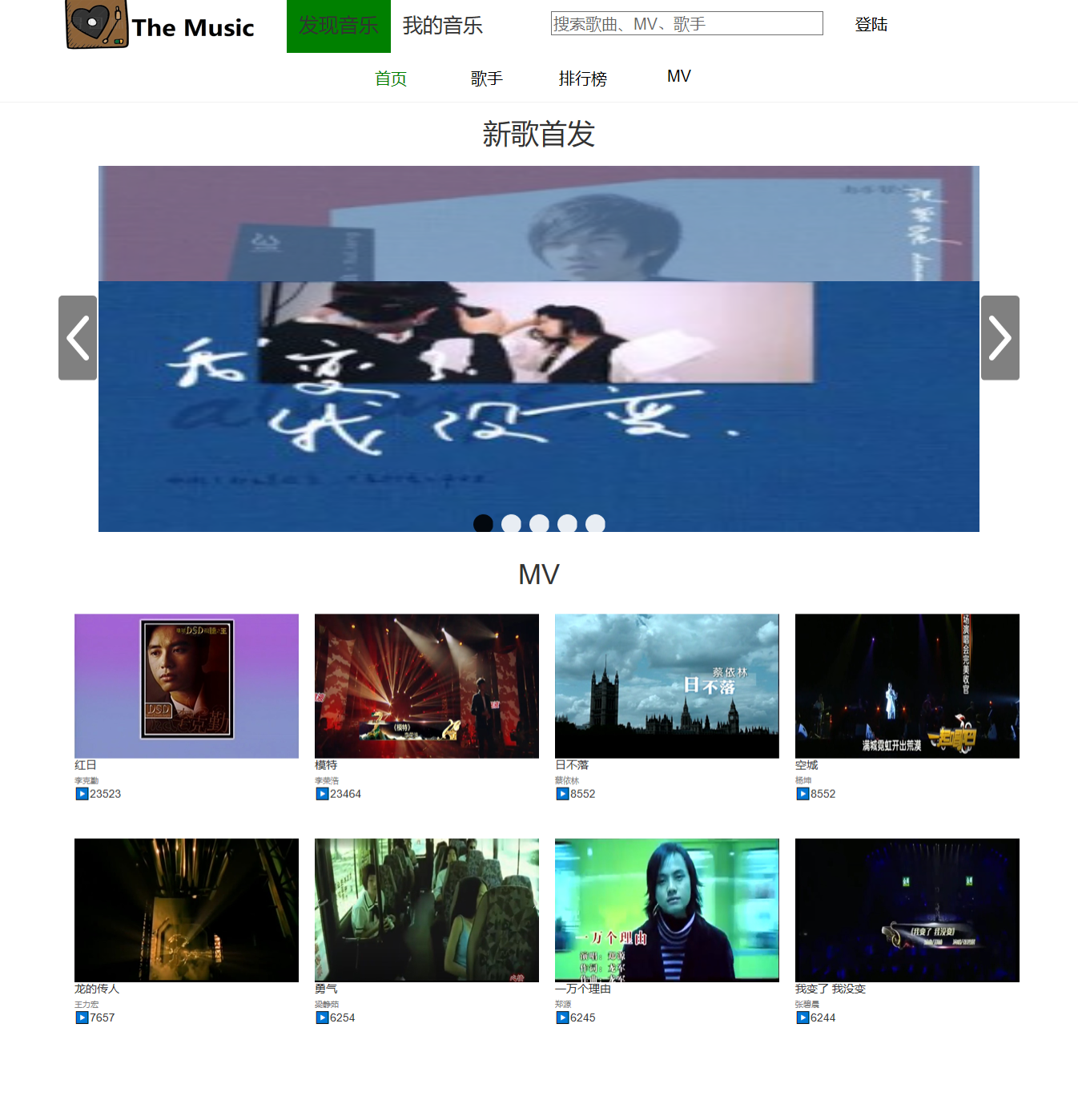


图6.2 用户首页图

## 6.3 登陆

如果想使用“我的音乐”模块，或者直接点击右上角的“登陆”，弹出登陆的提示框，点击登陆按钮后会通过jquery把表单数据传递到后台进行校验，识别该用户身份（用户或管理员）进入到对应主界面。同时在主界面右上角显示当前登录用户的名字



图6.3 登录界面图

## 6.4 用户注销

登录后在主页面右上角显示当前登录用户，悬浮在头像上在下拉列表中点击“注销”可以进行登陆注销。



图6.4 用户注销图

## 6.5 歌手页面

这个页面可以根据条件筛选歌手，点击歌手图片进入歌手主页，可以查看歌手的基本信息、歌曲和mv

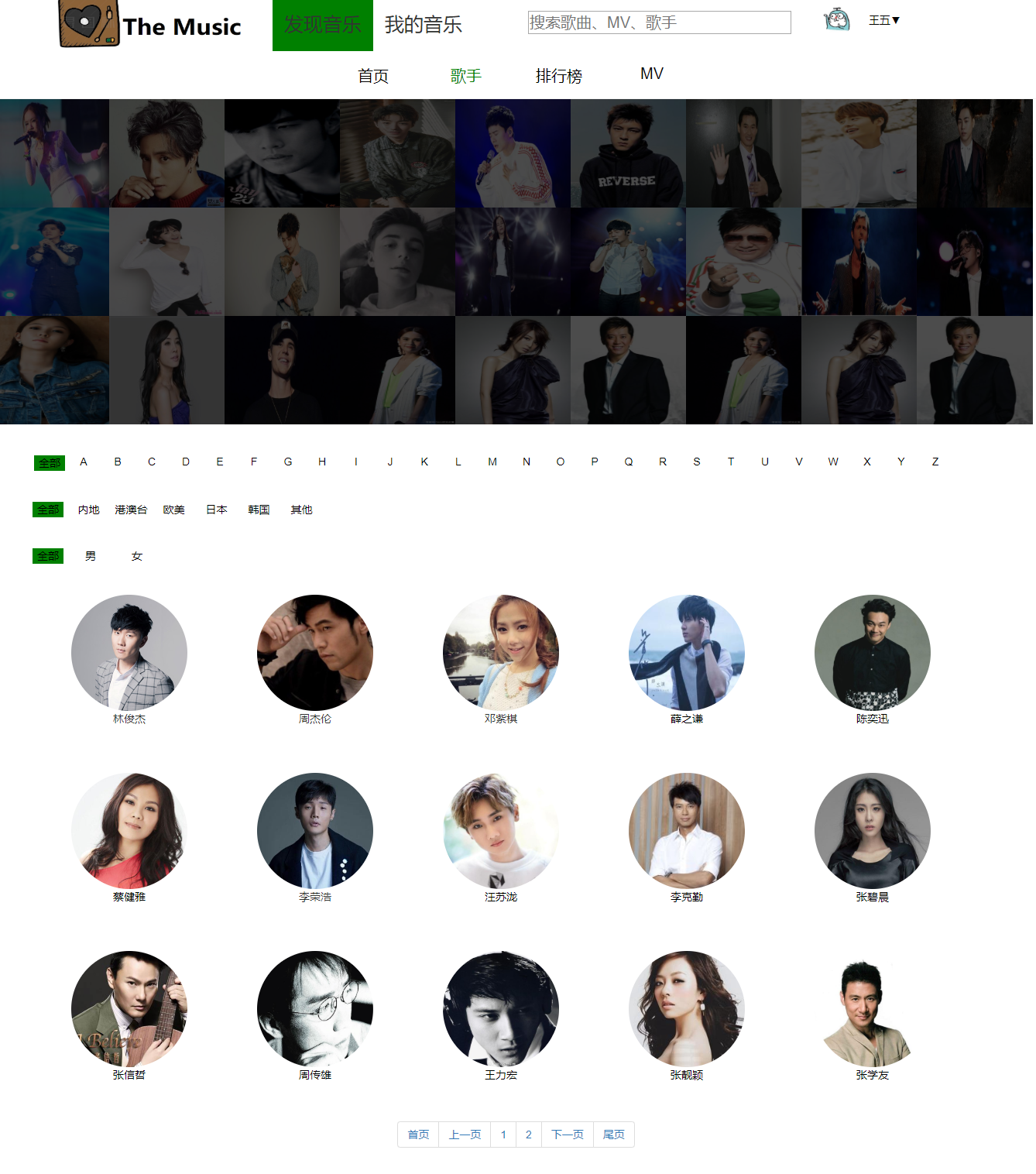


图6.5 歌手页面图

## 6.6 排行榜

这里显示歌曲按播放次数降序的歌曲。



图6.6 排行榜页面图

## 6.7 用户个人信息管理功能

每个用户对应有关的部分个人信息可以自行修改，如昵称、、性别等这种权限较低级的信息

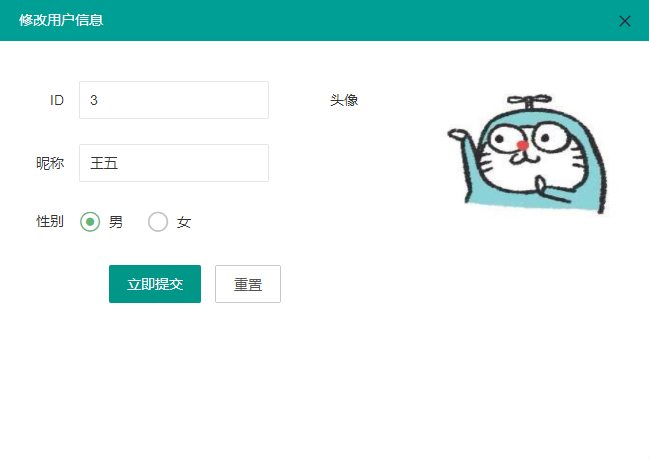


图6.7 修改个人信息图

## 6.8 歌曲主页

这个页面是显示歌曲的一些信息，包括评论。



图6.8 歌曲主页图

## 6.9 评论功能

用户在听歌曲的时候在播放页面的下方有一个评论按钮，点击进入歌曲主页，在歌曲下方有一个评论区，显示这首歌曲的用户评论，分为“精彩评论”和“最新评论”两部分，分别按照评论的点赞数降序和评论时间降序显示，用户可以在评论区进行评论、或者浏览其他用户评论，给评论点赞。



点击可以回复评论



图6.9 歌曲评论图

## 6.10 搜索功能

这个功能包含歌曲、歌单、mv的搜索,让用户更加快速的寻找到想要的资源。





图6.10 搜索页面图

## 6.11 在线听音乐

用户在其他页面点击播放歌曲会将歌曲添加到播放器列表中，进入到播放器页面进行播放歌曲，左边列表显示播放器中的歌曲队列，下方显示歌曲的播放进度，可以拖动进行快进（倒退），可以在列表中双击进行切换歌曲，也可以在右下方（上一曲、下一曲）等播放控制按钮进行控制。

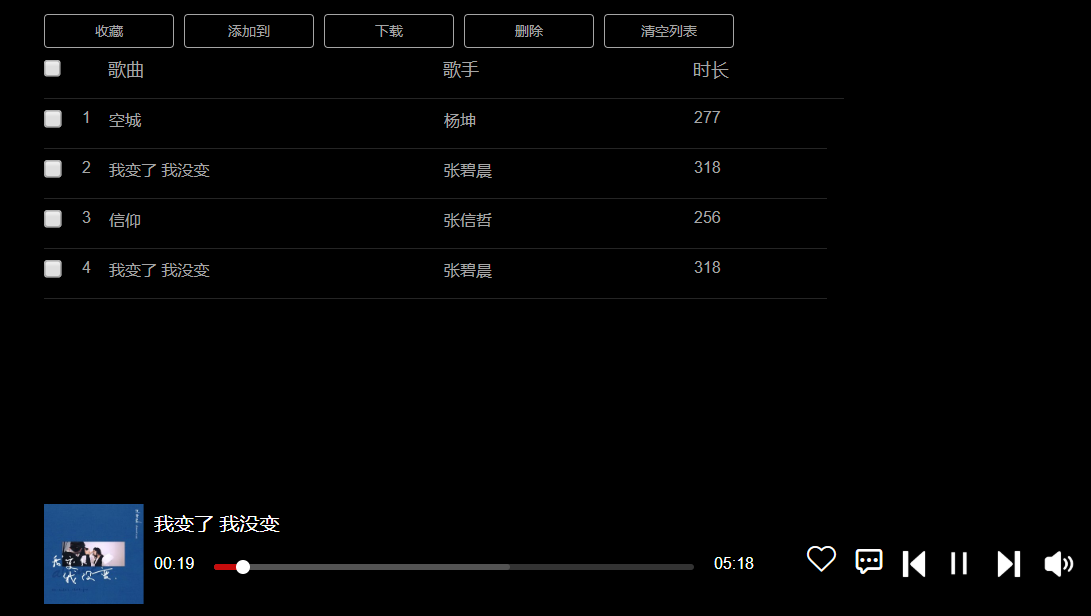


图6.11 音乐播放图

## 6.12 收藏（取消收藏）歌曲功能

再播放器页面的下方有一个❤（收藏/取消收藏）按钮，点亮可以把当前歌曲添加到用户的默认歌单------“我喜欢”的歌单中。已经点亮的歌曲也可以再一次点击取消收藏。



图6.12 收藏歌曲图

## 6.13 用户“我的音乐“的“我喜欢”模块

这个模块可以查看到用户收藏的单曲、专辑、歌单、MV，用户可以再这四个选项卡中浏览信息，在每一项的后面还有一个删除按钮，点击删除就可以将单曲（专辑、歌单、MV）取消收藏。

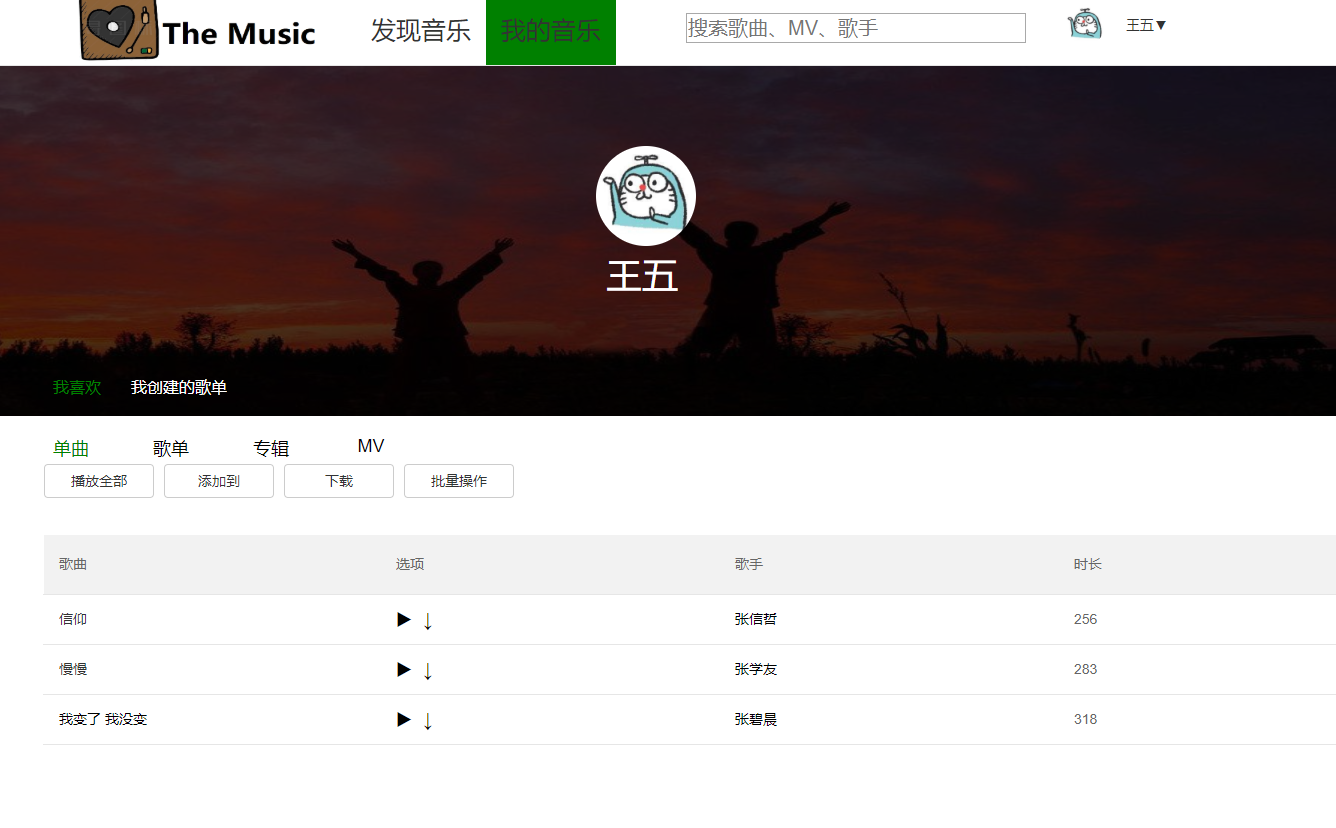


图6.13 我的音乐的我喜欢模块图

## 6.14 用户“我的音乐“的“歌单管理”模块

这个模块显示用户的所有歌单，包含歌单的名字、歌单中的歌曲数量，这个歌单不允许删除在上方有一个新建歌单按钮，点击可以创建一个用户自己的歌单，如果想删除歌单，歌单后方有一个删除按钮，不仅会删除歌单，还会删除歌单中的歌曲。其中系统默认歌单“我喜欢”只允许向歌单中添加（移除）歌曲，而不允许被删除。

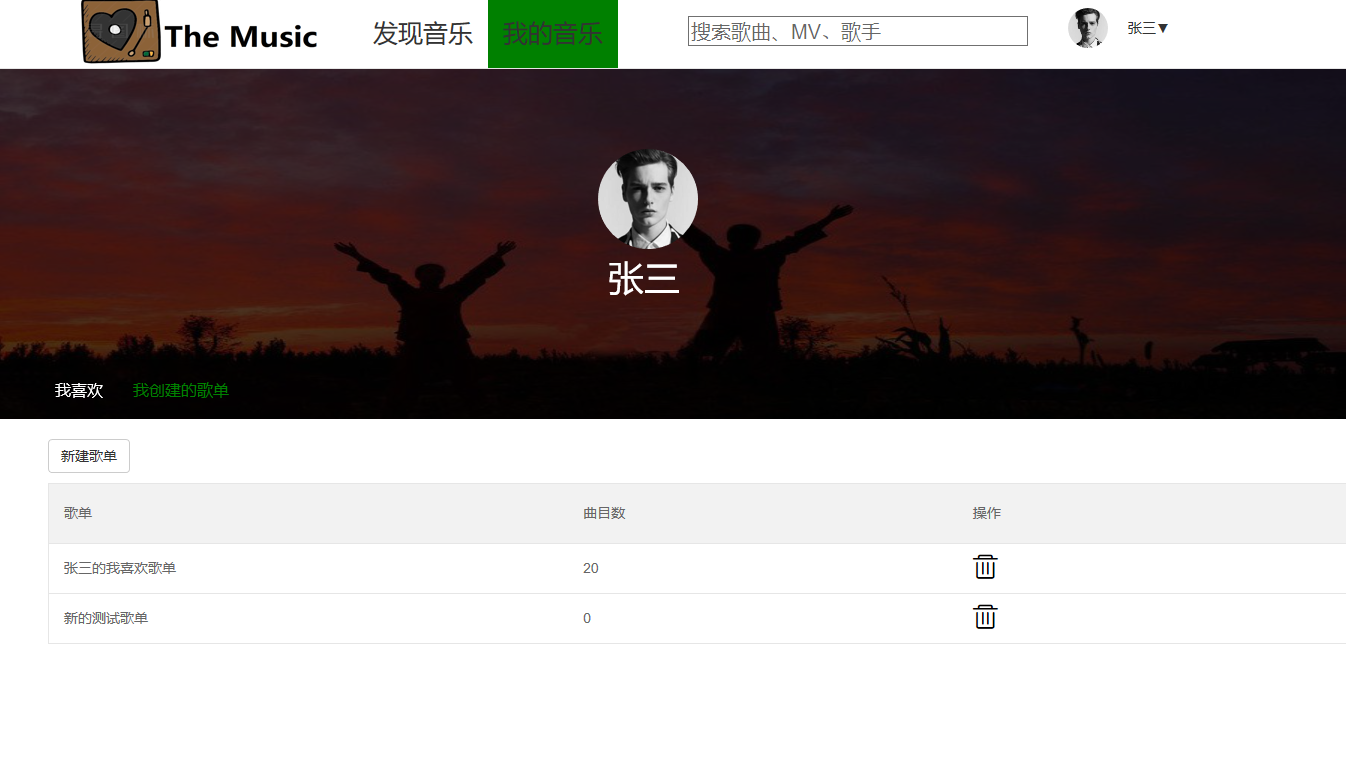


图6.14 我创建的歌单页面图

## 6.15 用户下载歌曲、MV

用户可以将喜欢的歌曲或者MV下载到本地，下载免费。



图6.15下载页面图

## 6.16 管理员主界面

这个界面是需要进行登陆并且登陆身份是管理员才能进入的界面，这个界面的主要功能是对歌曲、mv、员工进行管理。

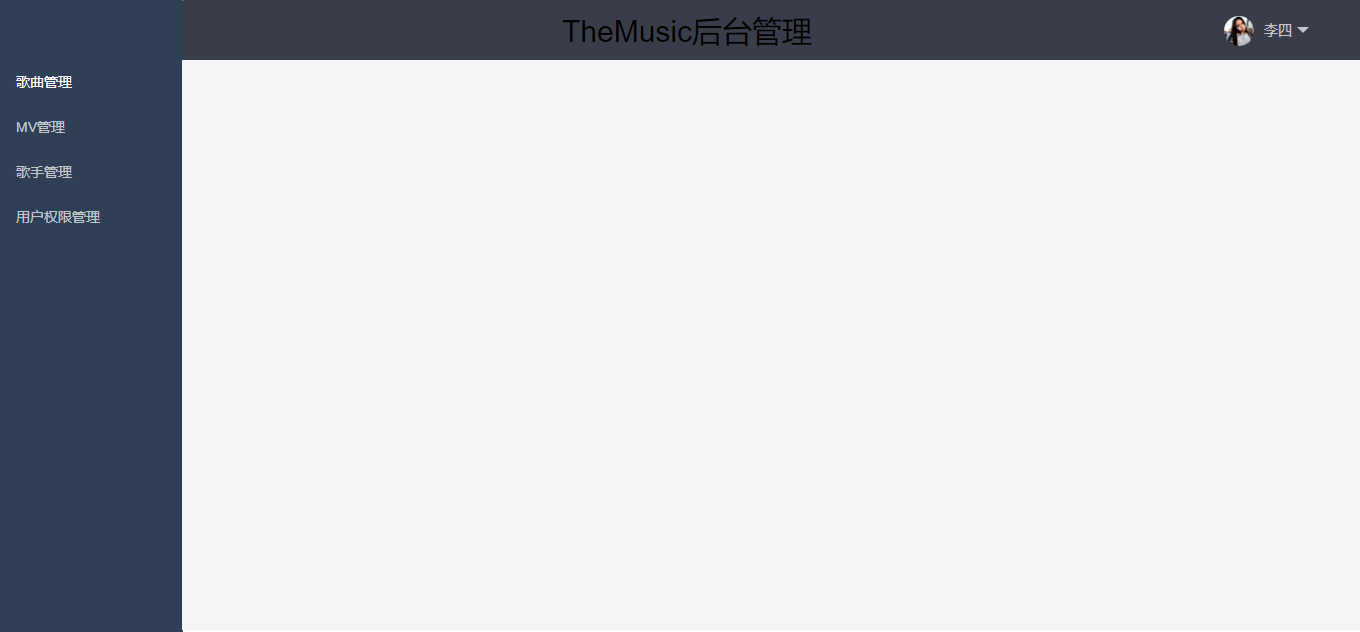


图6.16 管理员主页图

## 6.17 歌曲管理

歌曲管理是管理员主界面的一个模块，主要对歌曲进行管理，主要功能例如：添加歌曲、下架歌曲、修改歌曲信息等功能



图6.17 歌曲管理页面图

## 6.18 mv管理

mv管理是管理员主界面的一个模块，主要对mv进行管理，主要功能有例如： 上传mv、下架mv、修改mv信息等功能。



图6.18 mv管理页面图

## 6.19 歌手管理

这个功能是管理员对歌手进行一些操作，如添加、删除、修改、查询。



图6.19 歌手管理页面图

## 6.20 用户权限管理

这个功能是管理员可以修改用户的登陆身份为管理员，或者将管理员权限改为用户



图6.20 权限管理页面图

# 7 系统测试

系统测试是将经过集成的软件在实际的运行环境下，对系统进行一系列有效严格的测试，发现系统存在的问题并保证系统的正常运行。主要包裹功能功能测试、健壮性测试、安全性测试、压力测试等。目的在于，对最终的系统进行全面的测试检查，确保系统满足用户的要求，不会出现大问题，这是确保软件质量的关键，同时也是对功能说明文档和代码的最后审查。

## 7.1 测试计划

真正的测试是在和系统开发同时进行的，从最小最细微的单元测试开始到集成测试最后到系统测试。这里所有的测试都是依据需求规格说明书、设计原型以及规定的测试文档模板等相关文档进行测试的。

### 7.1.1 测试目标

确保系统的每一个功能都能正常运行并且符合系统功能规格说明书上，所写的需求，同时也要保证系统的容错率。于此同时尽可能多的发现软件中的错误，保证软件的质量，给用户一个满意的软件。参考人员主要为用户、开发人员、项目管理者。测试过程中，测试的人必须对业务需求的有清晰地了解，并对测试的结果进行详细的分析。

### 7.1.2 测试环境

硬件：

PC机（戴尔2.30GHz CPU，4G内存，独立显卡，258G硬盘）

软件：

操作系统：Windows 10

数据库: Mysql数据库

IDE工具：IDEA

Java环境：JDK 1.8

## 7.2 测试方法

本系统的测试方法主要使用了两种分别为黑盒测试、白盒测试。黑盒测试是只关心输入的信息和输出的结果是否符合预期，而白盒测试是细致到去探究源代码和程序结构，从根源出发去检验功能的实现，界面的实现等是否符合预期所想。

黑盒测试又分为功能测试和性能测试，本系统主要以黑盒测试为主白盒测试为辅而进行的综合性测试。

## 7.3 测试用例

### 7.3.1 功能测试

主要目的是检查软件的实际功能是否符合用户的需求，这部分的测试分成两个部分去执行，首先输入正确的部分，确保能按照预想的那样实现相应的功能，在输入错误的信息，看是否能按照相应的错误，系统自动给出错误提示。测试用例表如表7.1所示。

表7.1 用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能名称 | 测试用例 | 应有结果 | 实际结果 | 结论 |
| 用户登录 | 用户名user  密码：user | 进入用户界面 | 进入用户界面 | 正确 |
| 用户登录 | 用户名admin  密码：admin | 进入管理员界面 | 进入管理员界面 | 正确 |
| 用户注册 | 使用错误的手机号码或者填写错误的验证码 | 注册失败 | 注册失败 | 正确 |
| 用户注册 | 使用正确的手机号码并填写正确验证码 | 注册成功 | 注册成功 | 正确 |
| 收藏歌曲 | 点击未收藏歌曲的❤按钮，进行收藏 | 收藏成功 | 收藏成功 | 正确 |
| 取消收藏 | 点击已收藏歌曲的❤按钮，取消收藏 | 取消成功 | 取消成功 | 正确 |
| 修改歌曲、mv信息 | 填写要修改的正确信息 | 成功 | 成功 | 正确 |
| 修改歌曲、mv信息 | 填写要修改的错误信息 | 失败 | 失败 | 正确 |
| 上传歌曲、mv | 选择正确的文件上传 | 成功 | 成功 | 正确 |
| 权限分配 | 点击修改登陆身份为管理员开关 | 修改成功 | 修改成功 | 正确 |

### 7.3.2 界面测试

主要为了测试界面的美观性、易浏览性是否符合用户的所掌握的技能和知识，以免用户在使用中感到厌恶和操作困难，界面测试用例如表7.2所示。

表7.2 界面用例表

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试过程 | 结果 |
| 界面美观、风格一致性 | 检查界面的结构如登录界面、主界面的导航、菜单、字体等是否美观及风格是否一致 | 界面美观性符合用户需求、风格一致 |
| 易浏览性 | 检查功能的名称是否直观  输入输出的过程是否规范、结果是否简、洁易读  是否符合一般的操作习惯 | 符合预期所想 |

## 7.4 测试结论

通过对上述系统和模块进行的相关测试，虽然基本实现了用户的需求，系统也是可以使用并满足客户基本的日常需求的，虽然没有发现错误，但是这仅仅是表面的，还有更多的潜在的问题等待发现，产品想要有广阔的市场就必须符合用户的需求，而需求是不断在变化的。目前已实现了基本的部分，而且经过测试这些部分也基本满足需求，但是系统的系统还有很多不足的地方，除了在系统本身的功能还不是很全面，设计不够理想，还需要加强系统功能更加全面的设计。在许多方面的设计考虑的还不是很细致。界面也不够美观。所以系统还存在许许多多的功能问题有待完善，期望在后续的研究中能进一步的优化这个系统，让它更加的完善。

# 8 总结

经过数星期的研究和开发，网页音乐已经基本实现了音乐播放器网页版的所需要的功能，本系统根据网页播放器的实际情况，从界面的设计到使用都是从用户的角度出发，目的是带给用户良好的使用体验，简单易懂，让用户很容易就能掌握使用。主要功能包括在线听音乐、播放MV、歌曲评论、收藏(取消收藏)歌曲、将歌曲添加到我的歌单、歌单管理、用户管理、音乐管理、mv管理、歌曲排行榜、歌单管理等功能。

通过这次毕业设计，我通过实际出发锻炼了自己的动手能力。虽然中途遇到了很多困难，而且数据库设计也不太合理，经过后面的不断改进，也基本能满足网页音乐播放器的需求。不足之处是界面设计方面还需要进一步修饰和美化。

能基本完成这次毕业设计，给了我很大的信心，让我理解理论与实际结合的重要性，认识了自己的不足之处，我也会更加努力完善自己。

# 致谢

在完成这次的毕业论文的过程中，首先我要感谢我的指导老师，我是在老师的指导下完成这次的论文的。在这完成的过程中，我的老师对我的论文提出了许多意见，并督促我完成论文和耐心指导我，给了我巨大的帮助，无论是从选题到定稿，老师都倾注了大量的时间。老师的许多经验，在我遇到困难时给了我解决的思路，更重要的是老师总是督促我，让我有一种时间的紧迫感，我觉得这种感觉很好，因为我太懒散了，如果没有老师的督促我可能很难集中注意力去写论文，所以感谢老师在这次论文中给予的帮助。其次我要感谢我的父母，在这段时间的关怀和帮助，是他们给了我这样安逸的环境去写论文让我不用为我的衣食起居而担忧，从而让我顺利的完成论文的设计。同时我要感谢学院给我这样一次机会，毕业设计不单单只是毕业设计，它是一个很好的锻炼机会，它也是我们大学学习的一个验证，它证明了我们大学学习生活并没有白白的浪费掉。同时在完成的过程中我们能发现很多问题，然后用我的知识去解决问题就是一个提高的过程，为我们将来迈向社会提供了很好的帮助。

最后，我也不得不感谢在我学习和生活中，给了我很大帮助的温顺豪同学，我很感谢他们在这个过程中给予的帮助和支持，当我因为压力想放弃时，是他开导我、引导我，让我重拾信心，谢谢你的帮助和鼓励。

总之，快要毕业了，可是我不会就此放下，继续学习，努力拼搏和加油！在此，祝你们身体健康、事业有成！

# 参考文献

[1]唐权. SSM框架在JavaEE教学中的应用与实践[J]. 福建电脑,2017,33(12):93-94+61.  
 [2]李杉,贾彦平,达虎. Mybatis逆向工程在JavaEE中的应用[J]. 通讯世界,2017(24):342.  
 [3]李宏利,李梅,鱼晓. JavaEE技术课程资源建设方法研究[J]. 黑龙江教育(高教研究与评估),2018(02):60-61.［2］张耀春，黄轶．《Vue.js权威指南》．北京：电子工业出版社，2016，0-512.

［4］刘增辉．《MyBatis 从入门到精通》．北京：电子工业出版社，2017，0-312.

［5］杨开振．《深入浅出Spring Boot 2.x》．北京：人民邮电出版社，2018，0-298.

［6］杜波依斯.MySQL技术内幕[M].第4版.人民邮电出版社,2011年07月.

［7］戴声，王波。基于图数据库的大规模配置管理数据库关联查询[J].计算机与现代化，2017,（01）：51-56.

[8]黄统奎,张艳红. 基于JavaEE高校毕业设计管理系统的设计与实现[J]. 电脑知识与技术,2014,10(19):4384-4386.

［8］黄俊.Java程序设计与应用开发.机械工业出版社.2014，0-298.

［9］石正喜.MySQL数据库实用教程.北京：北京师范大学出版社.2014，0-237.

［10］崔洋.MySQL数据库应用从入门到精通.北京：中国铁道出版社.2013，0-235.

# 论文评定表

|  |  |
| --- | --- |
| 指  导  教  师  意  见 | 评 语：  数据库设计较完善，功能比较符合逻辑，项目贴近现实生活，操作简单，界面美观，总体来说不错。  评定分数 80 （签章）  年 月 日 |
| 学  院  负  责  人  意  见 | （签章）  年 月 日 |