

浙江科技学院

## 2018 届毕业设计(论文)

论文题目： 基于 Java Web 的在线音乐网站

学 院： 信息与工程学院

专 业： 软件工程

班 级： 142 班

学 号： 1140299345

学生姓名： 沈列飞

指导教师： 孙晓勇

二〇一八年 五月

## 摘要

随着当今中国近几年经济实力的不断发展，科学技术的不断发展，互联网科学技术在近几年有了飞速的发展。随着互联网的飞速发展，并且物联网的发展所带来的影响已经渗透到生活的方方面面，改善了人们的生活方式。音乐作为人们生活中陶冶情操的一种精神生活的方式，越来越需要互联网的支持。为了顺应此趋势，涌现出了很多在线流媒体音乐网站，如 QQ 音乐、酷狗音乐、酷我音乐、虾米音乐等音乐网站及其相应的客户端。

基于 Java Web 的在线音乐网站的基本功能就是实现用户能够在该网站上在线收听音乐的功能。当今各种网上的流媒体音乐网站随着版本的迭代，界面上各种更新后，界面上来说操作越来越繁琐。相比之下，基于 Java Web 的在线音乐网站实现了操作和界面上的极简化操作，可以根据自己的喜好创建自己的歌单，收藏其他用户创建的歌单。同时可以搜索音乐、专辑、歌手、歌单或其他用户，查看其他用户的信息和歌单，查看我创建的歌单内的歌曲，对歌单内的歌曲做个性化管理。查看我收藏后的别人的歌单。同时可以查看歌手、专辑和专辑内歌曲的单曲的功能。

本文介绍了基于 Java Web 的在线音乐网站的开发环境和开发技术，介绍了本文采用的 B/S 架构，即浏览器/服务端架构，介绍了此网站的设计与开发和各个模块的功能，包括主模块（发现音乐模块、我的音乐模块、我的收藏模块和歌曲模块）、登录流程（注册、登录和忘记密码），同时也涉及到了 Web 前端和服务端所提到的新的技术。

**关键词：**互联网，流媒体音乐，B/S 架构

## ABSTRACT

With the development of Chinese economic power, the continuous development of science and technology in recent years. Internet science and technology has developed rapidly also. With the development of the Internet, the impact of the development of the Internet has penetrated into all aspects of life and improved people's way of life. Music, as a kind of spiritual life in people's life, needs the support of the Internet more and more. In order to adapt to this trend, a number of online streaming media music websites have sprung up, such as QQ music, Kugou music, Kuwo music, Xiami music and so on with their corresponding clients.

The basic function of music based on Java Web is to enable users to listen to music online. With the iteration of the version and various updates on the interface, the operation of music website is becoming more and more complicated. Based on Java Web online music website, by contrast, has realized the operation and a simplified operating interface, can according to favorite themselves to create their own playlists, collect other users to create playlists. At the same time can search music, album, artist, song list or other users, and single view other user's information, check out my created within a single song, song within a single song do personalized management. Check out my favorites list. You can also check the performance of vocals, albums and songs on the album.

This paper introduces the Java Web based online music website development environment and development technology, this paper adopts B/S structure is introduced, namely the browser/server architecture, this paper introduces the design and development of this website and the function of each module, including main module (found music modules, my music module, my collection module and music module), the login process (registered, log in and forgot password), it also involves a Web front end and the server mentioned new technology.

**Keywords:** Internet, Streaming music, Browser/Server framework

## 目录

摘要 .....	I
ABSTRACT .....	II
第 1 章 引言 .....	1
1.1 背景 .....	1
1.2 研究内容 .....	1
第 2 章 系统分析 .....	3
2.1 系统可行性分析 .....	3
2.1.1 技术可行性 .....	3
2.1.2 经济可行性 .....	3
2.1.3 易操作性 .....	3
2.2 系统需求分析 .....	3
2.3 功能需求分析 .....	4
2.3.1 用户模块 .....	4
2.3.2 发现音乐模块 .....	4
2.3.3 用户创建歌单模块 .....	5
2.3.4 用户收藏歌单模块 .....	5
2.3.5 其他用户模块 .....	5
2.3.6 音乐播放模块 .....	6
第 3 章 开发技术 .....	7
3.1 使用的开发技术 .....	7
3.2 开发技术 .....	7
3.2.1 NodeJS 技术 .....	7
3.2.2 Web 前端技术 .....	7
3.2.3 Java 语言 .....	8
3.2.4 Spring & Spring Boot & Spring Cloud 技术 .....	8
3.2.5 MyBatis 框架 .....	9
3.2.6 Redis 高速缓存技术 .....	9
3.2.7 Nginx 高速反向代理技术 .....	9

3.2.8 MySQL 数据库 .....	10
第 4 章 系统设计 .....	11
4.1 系统体系结构.....	11
4.2 系统模块设计.....	11
4.2.1 用户模块.....	11
4.2.2 发现音乐模块.....	12
4.2.3 用户创建的歌单模块.....	13
4.2.4 用户收藏的歌单模块.....	14
4.2.5 其他用户模块.....	15
4.2.6 音乐播放模块.....	15
第 5 章 数据库设计与系统实现 .....	17
5.1 系统数据库设计.....	17
5.2 系统功能实现.....	23
5.2.1 用户模块实现.....	23
5.2.2 发现音乐模块实现 .....	25
5.2.3 用户创建的歌单模块实现.....	28
5.2.4 用户收藏的歌单模块实现.....	30
5.2.5 其他用户模块实现 .....	31
5.2.6 音乐播放模块功能实现.....	32
5.3 系统接口实现.....	32
5.3.1 邮件发送验证服务接口实现.....	32
5.3.2 音乐网站主服务接口实现.....	33
第 6 章 系统测试与维护.....	38
6.1 系统的测试.....	38
6.1.1 系统的黑盒测试 .....	38
6.1.2 系统的单元测试 .....	39
6.1.3 系统的系统测试 .....	39
6.2 系统的维护.....	40
第 7 章 结束语 .....	41
致谢 .....	42
参考文献.....	43

# 第 1 章 引言

## 1.1 背景

随着近几年计算机科学与技术的飞速发展，特别是互联网技术，它已经运用到了生活的方方面面。流媒体音乐也渐渐和互联网技术关联在一起了。

互联网技术的时代下，加速了各个行业的模式的转型。借助互联网技术，可以运用到生活的各个方面。听音乐对于人们的精神生活来说是不可获取的一部分，从互联网技术发展的过程中，和互联网技术达成了紧密的联系，也随之产生了很多音乐媒体的平台。

现下很多音乐媒体的平台大多以 C/S 架构为主，以客户端的方式呈现，这有一个弊端，如果用户没有安装客户端，那就必须去网站上下载并安装客户端，对于不是很懂的人来说安装可能是个比较繁琐的操作，中间安装步骤可能比较复杂。而 B/S 架构下的流媒体音乐网站所带来的好处是，用户只需要在浏览器上访问对应的流媒体音乐的域名即可，访问之后就可以在页面上进行操作，节省了下载和安装软件的时间和精力，方便了人们进行操作。

由于个人比较喜欢听音乐，所以产生了设计并实现基于 Java Web 的在线音乐网站的初衷。

本应用带来的好处是人们可也随时随地访问浏览器网址并收听自己喜欢的音乐。

## 1.2 研究内容

本 Web APP 研究的主要内容是设计与实现基于 Java Web 的在线音乐网站中的每个模块的内容，并达到需求上的要求。最终能够达到能够让用户能够很快上手此网站的各种操作，以及达到良好的用户体验，并能够使用户感受到良好的听觉和视觉冲击。

本 Web APP 主要涉及的范围有发现与搜索音乐模块、我创建的歌单模块、我收藏的歌单、用户个人中心、歌曲模块（其中包括艺人、专辑与歌曲三大小模块）、其他用户模块与其他用户的歌单模块和音乐播放模块等内容。

本应用在以上几个范围的基础之上，需要解决的问题有以下几个方面：一、

在注册时注册邮件和重置密码时重置密码邮件发送的问题；二、在歌曲模块中艺人、专辑与歌曲三大小模块的关系联系的问题；三、在我创建的歌单模块中，我创建的歌单与音乐之间关联的问题；四、在我收藏的歌单模块中，用户与其他用户的歌单进行关联的问题；五、在音乐播放模块中音乐的播放模式以及进度条的实时调控的功能。

在第二章中，将会介绍到系统可行性分析与需求分析。

在第三章中，将会介绍到系统使用的开发技术。

在第四章中，将会介绍到系统的设计。

在第五章中，将会介绍到系统的数据库设计与系统的功能实现。

在第六章中，将会介绍到系统的测试与系统的维护。

在第七章中，将会讲到结束语。

最后，写明自己的致谢以结束此篇毕业设计论文。

## 第 2 章 系统分析

### 2.1 系统可行性分析

#### 2.1.1 技术可行性

本 Web APP 采用 Java、HTML5+CSS3+JavaScript 作为基本的开发语言。Spring Boot、Spring Cloud 和 MyBatis 作为服务端主要的框架，MySQL 作为数据库。使用 Spring 的 Restful API 作为与前端之间接口交互的技术。在前端 Web 方面，采用 NodeJS 作为服务端的一个架子，以 jQuery 作为 JavaScript 的集成框架，以 AJAX 异步刷新的方式提供良好的网络交互，增强可维护性。以 Redis 作为高速缓存存储用户的登录信息，使用 Nginx 作为高速反向代理。使用 git 作为代码托管工具。

#### 2.1.2 经济可行性

本应用的开发语言都是开源免费的，成本比较低。而使用的 IDE 是 IntelliJ IDEA 和 WebStorm，这两个的价格比较高，但是用学生邮箱可以免费使用。而且使用本应用只需要一个浏览器即可进行访问，不需要安装其他的软件，比较省时省力。用户只需要在浏览器上访问本应用的域名即可进行访问和使用本 Web APP，比较方便。

#### 2.1.3 易操作性

本应用采用的设计架构是 B/S 架构，即 Browser/Server 架构（浏览器/服务端架构）。适合各大平台，在 macOS、Windows OS、Linux OS 等发行版的平台上均可使用。用户只需要会使用浏览器就可以非常容易驾驭本 Web APP。

### 2.2 系统需求分析

基于 Java Web 的在线音乐网站的设计与实现旨在给喜欢听音乐的用户提供一个良好的听觉与视觉体验的平台。能够让用户随时都能在线享受音乐带来的听觉盛宴。

需求需要实现用户模块、音乐模块、用户创建和收藏歌单模块、发现音乐模



块和播放音乐模块的功能。

## 2.3 功能需求分析

此次毕业设计的课题目的是开发一个基于 Java Web 的在线音乐网站，基于上述系统需求分析后具有以下基本功能：

### 2.3.1 用户模块

用户模块主要包含的子模块有注册、登录、忘记密码和个人中心的信息设置。

在注册的子模块中，用户可以根据自己的个人信息进行注册，需要提供邮箱进行注册的验证，然后之后就是设置密码和一些其他的基本信息的操作。

在登录子模块中，用户可以用自己的邮箱进行登录，登录后即可跳转到首页进行各种操作。

在忘记密码子模块中，用户先用自己的邮箱进行重置密码验证，验证通过后即可重新设置自己的密码。

在个人中心的子模块中，用户可以看到自己的个人信息，有自己的昵称、出生日期、所在地区等信息。用户在这里可以进行对自己的个人信息进行设置。也可以点选密码设置进行对密码进行重设。在个人信息中心，有头像的显示，然后可以点选头像下方的更换头像按钮跳转到头像设置的页面进行头像的设置。

### 2.3.2 发现音乐模块

在发现音乐模块主要包含的是提供搜索关键字进行对各种信息的搜索功能。

提供一个关键字，在导航栏中的搜索框进行搜索，即可跳转到搜索音乐的页面。

在搜索音乐的页面中，点选单曲后，系统根据提供的关键字，进行对音乐进行搜索并提供搜索出来的和单曲有关联的歌曲记录。点选专辑后，会出来与专辑有关的专辑记录。点选艺人后，会出来与艺人有关系的艺人记录。点选歌单页面，会出来用户创建的歌单记录。点选用户页面，会出来搜索到的用户的记录。

在单曲选项中，点击对应的信息，会跳转到对应的页面，点击收藏按钮，即可实现对歌曲进行收藏。

在专辑选项中，点击专辑的信息，会跳转到专辑信息的页面。

在艺人选项中，点击艺人信息，即可进入艺人页面。在艺人页面中，有艺人的专辑信息和艺人信息。点击专辑，即可进入专辑页面。在专辑页面中，有这个专辑所包含的歌曲。点击每首歌曲，即可进入单曲页面。在单曲页面，可以看到单曲的信息，可以对单曲进行收藏操。在单曲页面下，可以对单曲进行评论，对于自己已经评论的内容，如果觉得不是很好的话也可以对评论进行删除操作，对于自己喜欢的评论，用户可以对其进行点赞或者取消点赞。

在用户歌单选项中，点击用户歌单，几个进入每个歌单的页面。在此页面中，可以播放用户歌单。可以收藏用户歌单内的歌曲。

在用户选项中，点击进入用户，可以进入其他用户的页面。

### 2.3.3 用户创建歌单模块

在用户创建的歌单中，具有以下几个功能：创建歌单、删除歌单、编辑歌单信息、对歌单内的歌曲进行管理和播放歌单功能。

创建歌单的时候，用户可以给歌单取名后进行创建歌单操作。

如果用户觉得歌单自己不满意或者有其他的原因，可以进行删除歌单操作。

如果用户觉得歌单信息有误，可以对歌单信息进行编辑。

在每个歌单中，用户可以对自己的歌单内的歌曲进行管理，可以收藏到另外的歌单中，也可以删除。

用户也可以播放自己创建的歌单。

### 2.3.4 用户收藏歌单模块

在用户收藏歌单模块，可以看到用户收藏的其他的用户的歌单。

对歌单可以进行一定的管理，涉及对收藏歌单进行取消收藏操作、对收藏歌单内的歌曲进行收藏操作以及播放用户收藏的歌单的操作。

### 2.3.5 其他用户模块

在其他用户模块中，可以看到某个其他用户的一些基本信息、该用户创建和收藏的歌单的数量以及该用户创建的歌单和收藏的歌单的信息。

点击该用户创建的歌单的某个歌单即可进入该用户创建的某个歌单的页面，并可以播放创建的歌单。同时可以对歌单内的歌曲进行收藏操作。点击相应的链接也可跳转到对应的页面进行操作。

点击该用户收藏的歌单的某个歌单即可进入该用户收藏的某个歌单的页面，并可以播放收藏的歌单。与上述歌单相似，也可以进行收藏操作。点击相应的链接可跳转到对应的页面进行一定的操作。

### **2.3.6 音乐播放模块**

在音乐播放模块中，页面中显示的是之前点击的播放的某个歌单中的歌曲列表并对这个歌曲列表进行播放行为。

在音乐播放中，主要包含有音乐播放列表、正在播放的音乐的信息、播放条等信息的显示。

音乐列表中有被播放的歌单的歌曲列表，可以播放其中自己选定的一首歌曲，也可以对其中的歌曲进行查看和收藏的操作等。

正在播放的音乐的信息中，有歌曲所属的专辑图片、歌曲和艺人信息的显示。

在播放条中，有播放的音乐的信息的显示。可以点击播放暂停按钮来控制音乐的播放状态。可以点击上一曲和下一曲来控制音乐的播放选择。可以点击进度条来控制音乐的播放进度，也可以进行滑动调整音乐的播放进度。对音量可以进行音量的调整以及音乐的播放模式的选择。音乐播放模式有单曲循环、循环播放和随机播放这三个播放模式。

## 第 3 章 开发技术

### 3.1 使用的开发技术

前端开发技术：NodeJS、HTML5+CSS3+JavaScript（jQuery）、AJAX

后端开发技术：Spring、Spring Boot、Spring Cloud、Nginx、Redis、MyBatis

数据库开发技术：MySQL

源代码管理技术：Git

### 3.2 开发技术

#### 3.2.1 NodeJS 技术

本应用使用 NodeJS 作为前端的服务启动程序。

NodeJS 是一个偏向于服务端的 JavaScript 运行环境，由于其是一个基于事件驱动 I/O 和 Google V8 引擎，执行 JavaScript 的性能非常好。

依托于 NodeJS 的 Express 框架，可以快速地构建一个的网站。在 Express 框架中可以导入自己需要的中间件和开源框架，使得应用变得更简单快速。在功能上，Express 扩展了 NodeJS 所没有的功能。

本应用使用 Express 中的 HTML 格式的页面作为主要的模板引擎。

#### 3.2.2 Web 前端技术

本应用使用 HTML5、CSS3、JavaScript（jQuery）和 AJAX 作为前端开发的语言。

Web 前端是从网页设计和开发演变而来。在最初的前端网页设计和开发中使用的是 HTML 和一些 CSS 实现的，其中大多数网站以静态网页为主。而随着时代的变迁和 Web 前端技术的发展，特别是随着 HTML5 和 CSS3 的发展，并配合 JavaScript 和基于 JavaScript 的 jQuery 库迭代的使用，能够实现网页上的各种炫酷的效果，使得现下绝大多数网站以动态设计的网页为主。

借助于 JavaScript 和 jQuery 实现前端的一些效果，并辅之以 AJAX 异步无刷新网页改善用户体验。

### 3.2.3 Java 语言

本 Web APP 使用 Java 作为服务端的语言。

Java 是一门面向对象的语言，在吸收了 C++ 的弊端的基础之上改进，特别是摒弃了在 C++ 中的指针，而直接采用对象的引用。Java 具有继承、封装、多态、重载、多线程和可移植性等特性。<sup>[1]</sup>

由于 Java 具有高度的跨平台特性，使得其具有很好的 Web 服务端的兼容性。在多个平台可用，不管是在 Linux 上，还是在 macOS 和 Windows 上，都可以使用 Java 来实现 Web 服务端语言的编译和部署。<sup>[8]</sup>

### 3.2.4 Spring & Spring Boot & Spring Cloud 技术

本 Web APP 使用 Spring、Spring Boot 和 Spring Cloud 作为本 Web APP 的服务端的核心框架。

Spring 是一个将 Java 语言集成开发后的一个开源的框架，它提供了 J2EE 的业务方面的一站式解决方案。Spring 具有 IOC（控制反转）/DI（依赖注入）的特性，使得 Spring 将所有创建的 Bean 放入到一个容器中，需要哪个 Bean 就直接从容器中使用。<sup>[3]</sup>Spring 还具有 AOP（面向切面编程）的特性，提供了业务逻辑这个面向切面编程的特性，采用动态代理特性，使得 AOP 在 Spring 中具有举足轻重的作用。<sup>[4]</sup>

Spring Boot 是基于 Spring 集成开发后的一个框架，它简化了 Spring 的代码配置，使开发人员专注于 Web 服务端的业务开发，从而减轻了开发人员配置压力。采用使用 JavaBean 基于注解的方式来实现 Spring 的一些基础配置。<sup>[2]</sup>

Spring Cloud 是基于 Spring Boot 继承后的一个微服务框架，其中具有服务治理 Eureka、服务调用 Feign 和消息总线 Bus 等特性。Spring Cloud 的优势在于，原本庞大的服务端，可以经过 Spring Cloud 拆分后得到若干微服务，使得服务端模块化。这些被拆分后的微服务各司其职，服务端的耦合程度被降低了，每个微服务内部的内聚成都被提高了，达到了软件设计的低耦合、高内聚的目的。也方便了开发运维人员的服务端问题定位。但是问题是随着微服务数量的增加，维护成本也提升了。

### 3.2.5 MyBatis 框架

本网站应用使用 MyBatis 作为数据库访问框架。

MyBatis 由 Mapper XML、Mapper Java Interface 接口和 POJO 数据库表对象组成。其中在 Mapper XML 中主要负责数据库方面的编码，在 Mapper Java Interface 接口主要负责连接 Mapper XML 操作数据库，在 POJO 数据库表对象主要是一个 Model 的模型角色，负责存储和给前端展示数据使用。

而且 MyBatis 相比于 Hibernate 来说具有得天独厚的动态 SQL 灵活性，SQL 维护成本低，Java 与 SQL 分离的特征，这使得相比于 Hibernate JPA 来说根由使用上的灵活性和易维护性。<sup>[10]</sup>

### 3.2.6 Redis 高速缓存技术

本网站使用 Redis 作为高速缓存。

Redis 作为高速缓存可以缓解数据库的压力。如果还是按照以往的访问数据库的方式，那么数据库在高并发的情况下可能会崩溃或者数据库服务器宕机。采用 Redis 缓存的时候，可以将从数据库访问到的数据放在缓存中，当下次访问的时候可以先从缓存访问数据，然后如果缓存没有就访问数据库。当无需这些数据的时候，可以清空缓存。

### 3.2.7 Nginx 高速反向代理技术

本网站使用 Nginx 来实现网站的某些资源的高速反向代理。

Nginx 的高速反向代理可以在用户发起资源访问的时候可以快速访问到资源所在位置，并将访问的资源快速返回给前端展示。

使用 Nginx 可以实现无需自己写繁杂的服务端代码，直接配置资源的访问路径以及资源访问的路径所在的绝对路径即可。在配置完后直接使用 Nginx 命令将服务重新加载即可。

而且安装及其方便，在 Linux 中可以使用包管理工具下载安装，在 macOS 下可以借助 Homebrew 第三方包管理工具即可安装使用。

### **3.2.8 MySQL 数据库**

本应用采用 MySQL 作为数据库。

MySQL 是 Oracle 提供的开源数据库。支持数据库绝大多数特性。支持事务的 ACID（原子性、一致性、隔离性和持久性）的特性。支持数据库索引的特性，而且开源免费，成本比较低。

## 第 4 章 系统设计

### 4.1 系统体系结构

本系统采用 B/S 结构开发，即 Browser/Server（浏览器/服务端）结构，是一种伴随着互联网技术的发展而兴起的一种结构。在这种结构下，应用分为浏览器端和服务端两部分。浏览器端部分是每个用户拥有，服务端则是众多用户的信息存储的中台，负责处理用户的各种请求和业务逻辑的处理。而浏览器端只是负责一些简单的功能的展示，服务端则执行后续任务。

B/S 架构和 C/S 架构是在互联网技术不断发展的时代下兴起的一种架构。C/S 架构就目前而言主要有 PC 端和移动端两个方面。但是 C/S 端的不足之处是用户需要从浏览器上访问官网下载软件并安装，而且有些软件安装及其麻烦，对于不是很懂软件安装的用户来说，并不是很方便。而且带来的问题是比较占用电脑空间。对于移动端而言，软件上架需要指定的系统提供商进行一段时间的审核。

相比于 C/S 架构而言，B/S 架构只需要 PC 上一个浏览器即可，而且系统更新迭代和发布不会受到第三方系统提供商的影响，有发现问题可以及时修复并更新上线。省去了审核的时间。

### 4.2 系统模块设计

在设计与实现基于 Java Web 的在线音乐网站时，考虑到良好的用户体验，将系统分为如下几个模块的内容：（1）用户模块；（2）发现音乐模块；（3）用户创建的歌单模块；（4）用户收藏的歌单模块；（5）其他用户模块；（6）音乐播放模块。

概括来说整个系统包括如上六个模块的内容。

#### 4.2.1 用户模块

用户模块主要包括用户的注册、登录、忘记密码和个人中心设置四个子模块。

其中四个模块中：注册包括新用户的注册；忘记密码主要是老用户登录的时候忘记密码；登录即为老用户登录；个人中心涵盖了用户登陆进去之后对自己的个人信息进行设置的功能等操作。



用户模块涉及的内容如图 4-1 所示：

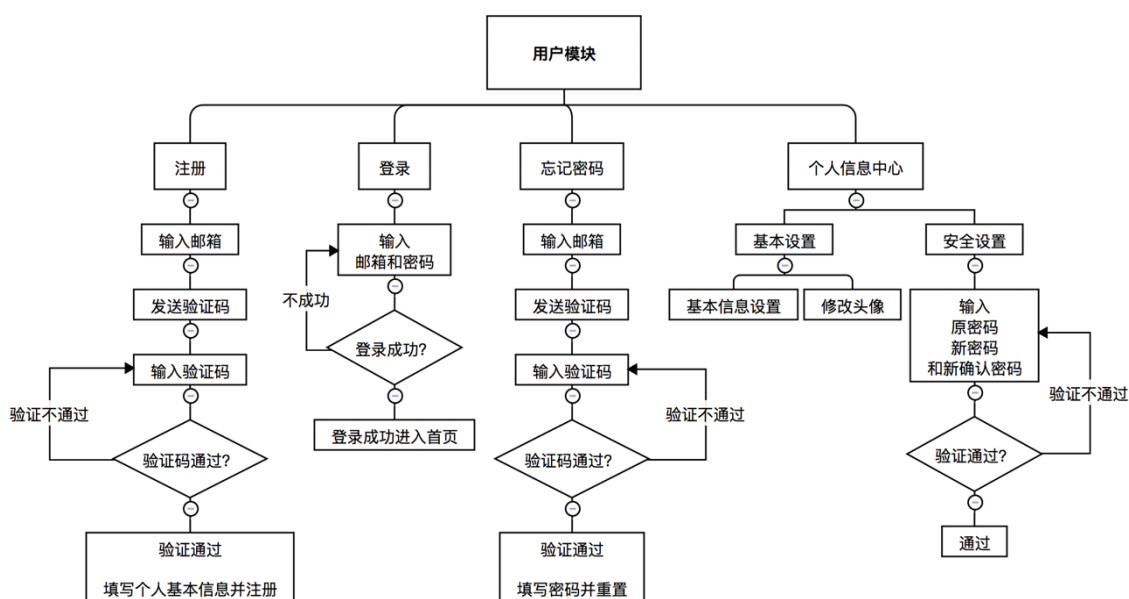


图 4-1 用户模块的模块图

### 4.2.2 发现音乐模块

在发现音乐模块中，主要包括搜索单曲、歌手、专辑、歌单和用户五大子模块。

在五大子模块中：搜索单曲主要是反馈单曲列表信息，并可以对每首单曲进行收藏操作；搜索歌手中，显示的是与歌手有关的歌手信息列表；搜索专辑，显示的是与专辑有关的专辑列表信息；搜索歌单，显示的是与歌单有关的歌单信息；搜索用户，显示的则是于用户有关的用户列表信息。

发新音乐模块涉及的内容如图 4-2 所示：

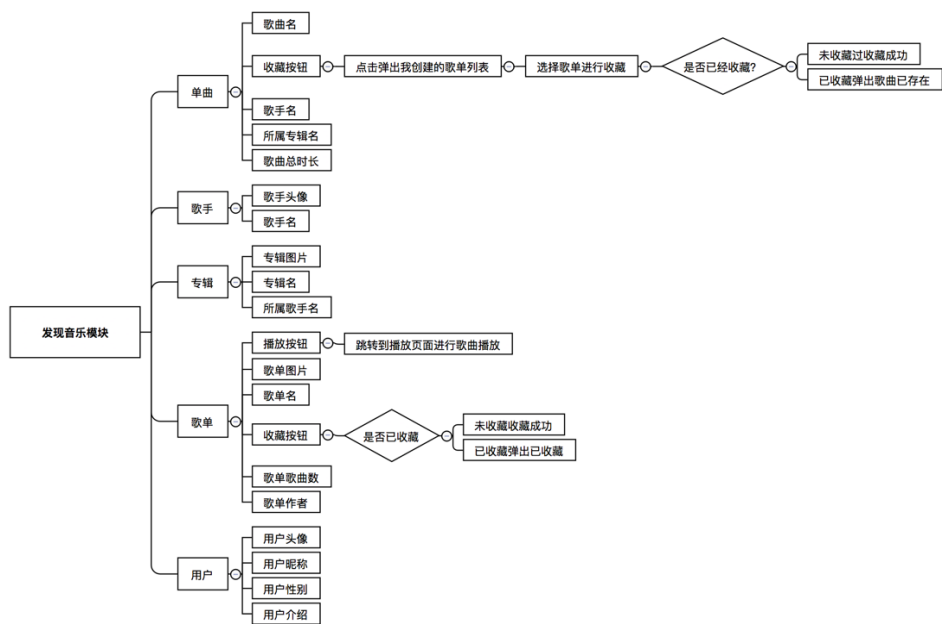


图 4-2 发现音乐模块总图

其中发现音乐模块中的艺人、专辑和单曲子模块如图 4-3 所示：

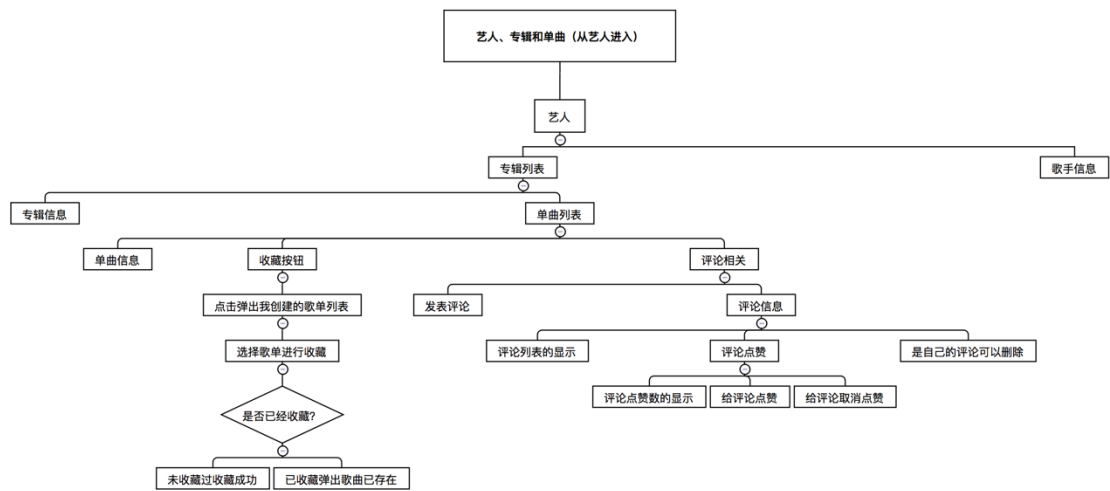


图 4-3 艺人、专辑和单曲子模块设计图

歌单和用户模块将会在其他用户模块中介绍到。

### 4.2.3 用户创建的歌单模块

用户创建的歌单模块主要包含新建歌单、编辑歌单、删除歌单、对歌单内的歌曲进行管理以及播放歌单这五个子模块的内容。

在这五个子模块中：用户可以根据自己的爱好创建歌单；用户可以进行编辑

自己的歌单；用户可以删除自己的歌单；用户可以对歌单内的歌曲列表进行管理；用户可以选中一个歌单进行播放。

用户创建歌单模块涉及的内容如图 4-4 所示：

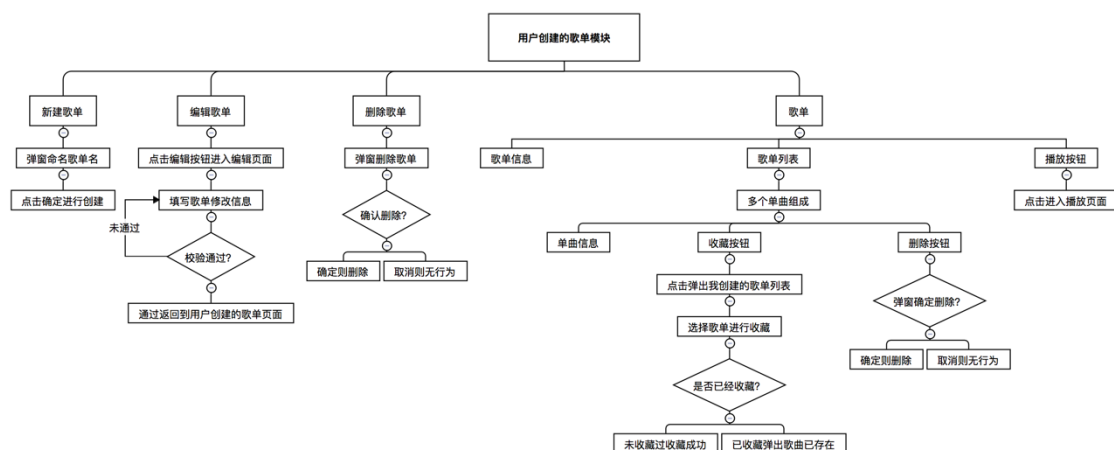


图 4-4 用户创建的歌单模块图

#### 4.2.4 用户收藏的歌单模块

用户收藏的歌单主要包含取消收藏歌单、对收藏的歌单内的歌曲进行收藏以及播放歌单这三个子模块的内容。

在这三个子模块中：用户可以取消收藏某一个已经收藏的歌单；用户可以对歌单内的歌曲进行收藏；用户可以选中一个收藏的歌单进行播放。

用户收藏的歌单模块的内容如图 4-5 所示：

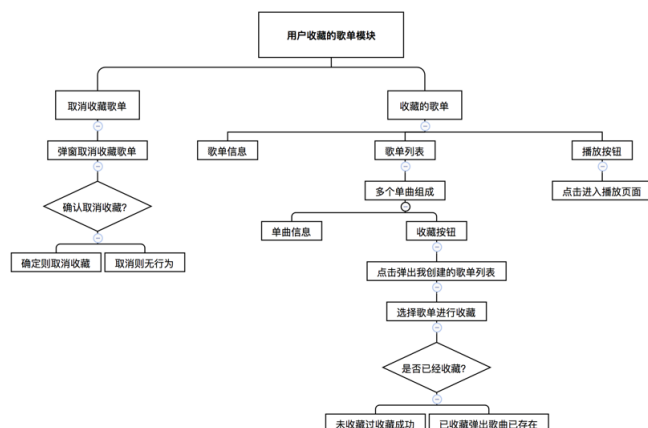


图 4-5 用户收藏的歌单模块图

### 4.2.5 其他用户模块

其他用户模块中主要包含其他用户的信息的显示、其他用户创建的歌单的数量和歌单列表信息以及他用户收藏的歌单的数量和歌单列表信息这三个子模块的内容。

在这三个子模块中：用户可以看到其他用户的基本信息的显示，包括用户头像、用户昵称、用户性别、用户个人介绍和用户所在地区的信息；查看其他用户创建的歌单的数目和创建的歌单列表，在其他用户创建的歌单中，也可以查看歌单的信息，播放歌单，收藏歌单和查看歌单内的歌曲信息并进行收藏音乐操作；查看其他用户收藏的歌单的数目和收藏的歌单列表，在其他用户收藏的歌单中，也同样可以查看歌单的信息，播放歌单，收藏歌单和查看歌单内的歌曲信息并进行收藏音乐操作。

其他用户模块的内容如图 4-6 所示：

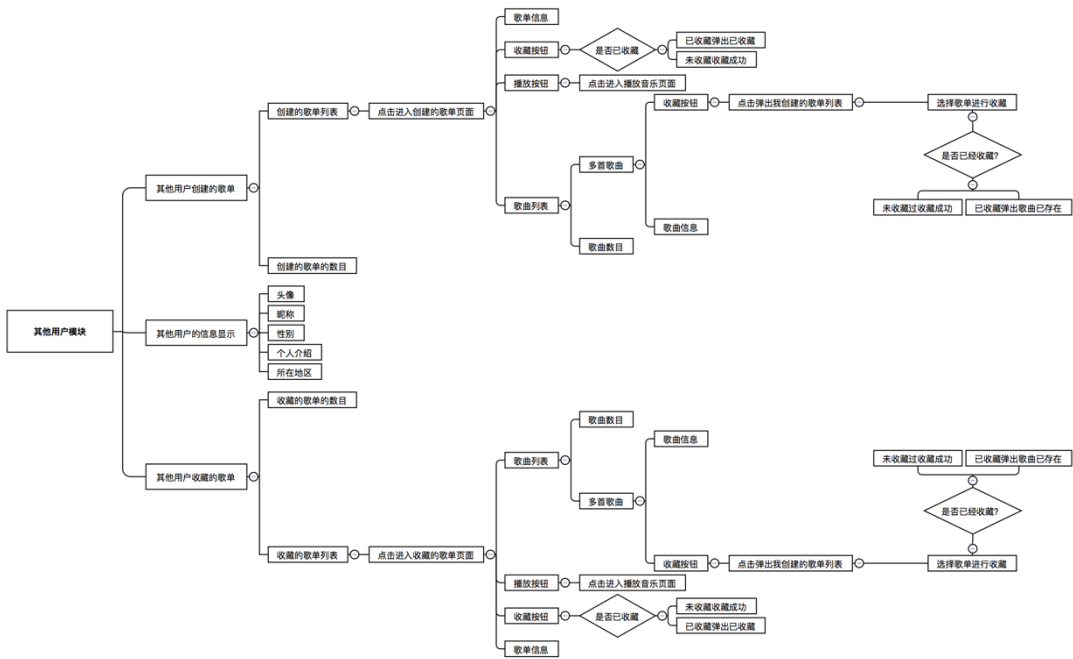


图 4-6 其他用户模块图

### 4.2.6 音乐播放模块

音乐播放模块中，主要由播放列表、歌曲概要信息和播放条这三个子模块组成。

在这三个子模块中：用户可以查看播放列表，可以点击其中的各种链接就可以跳转到对应的页面，然后点击收藏按钮，可以将歌曲添加到另一个歌单中，点击播放按钮就可以播放对应的单曲；歌曲概要信息中，有显示歌曲的专辑图片、歌曲名、歌手和专辑信息，点击对应的链接会跳转到对应的页面；在播放条中，有上一曲、下一曲、暂停/播放、歌曲名、歌手、进度条（播放进度）、音量调节和播放模式这几个控件，点击不同的按钮会触发不同的播放行为。

音乐播放模块的内容如图 4-7 所示：

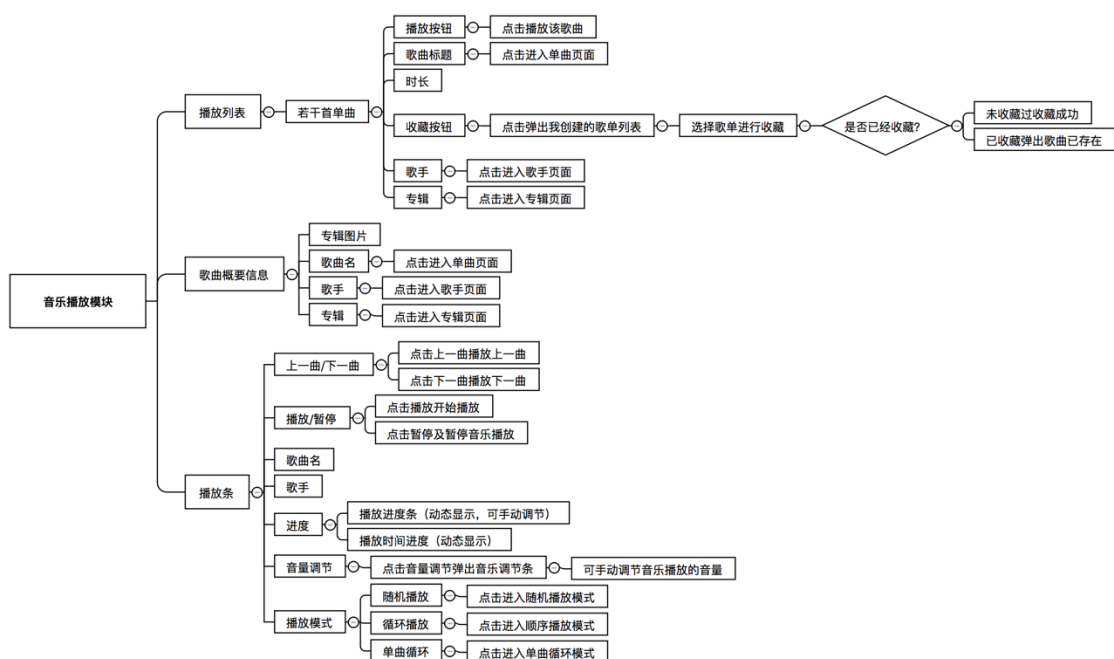


图 4-7 音乐播放模块图

## 第 5 章 数据库设计与系统实现

### 5.1 系统数据库设计

本系统总共由音乐服务和邮件服务这两个数据库组成，相互之间独立。

在音乐服务中，由用户、艺人、专辑、歌曲、用户歌单、用户歌单内歌曲、歌单收藏、歌曲评论和点赞歌曲评论这 9 张数据库表组成。

在邮件服务中由邮件验证码这个数据库表组成。

音乐服务系统数据库设计的 ER 图如图 5-1 所示：

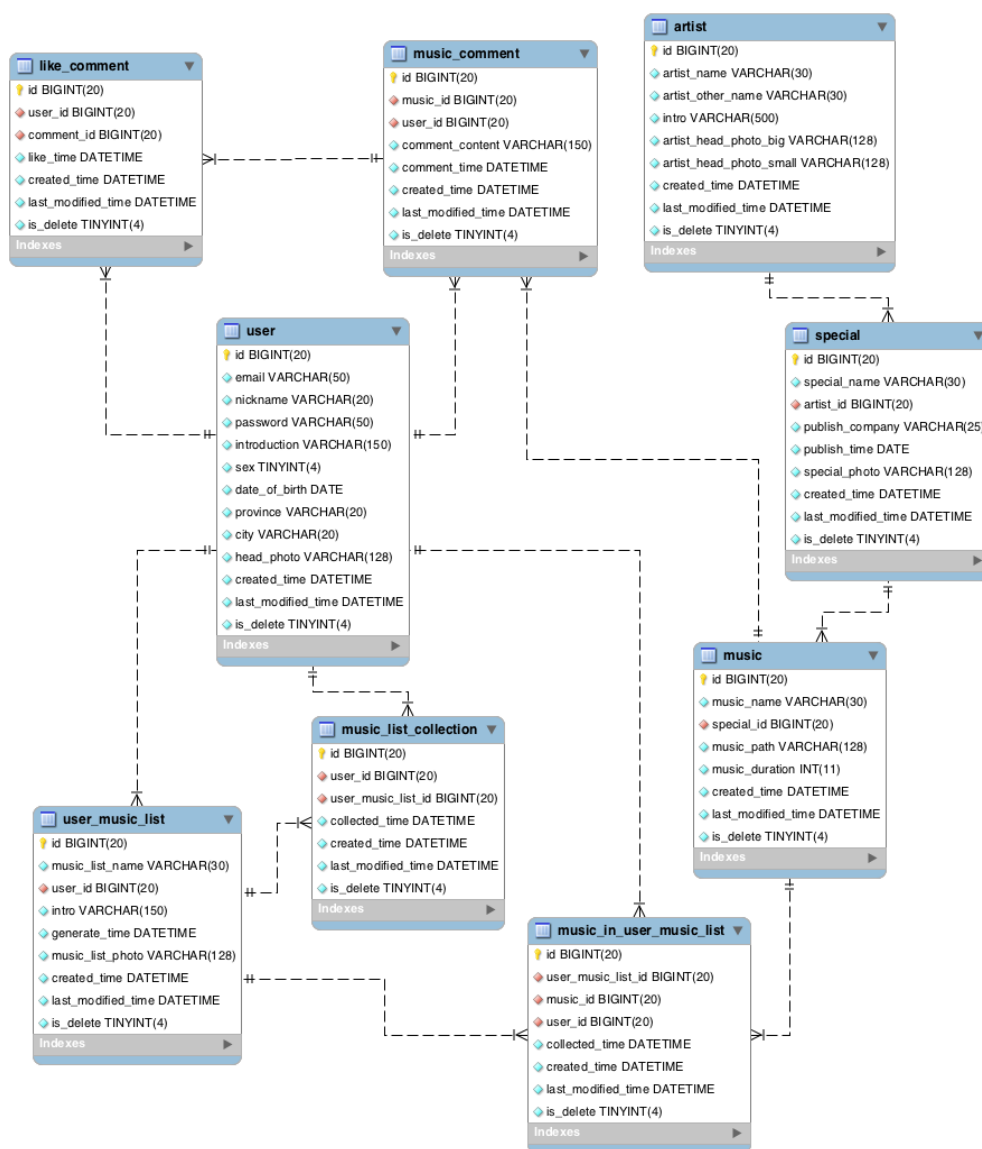


图 5-1 音乐服务的数据库 ER 图

邮箱服务系统数据库设计的 ER 图如图 5-2 所示：

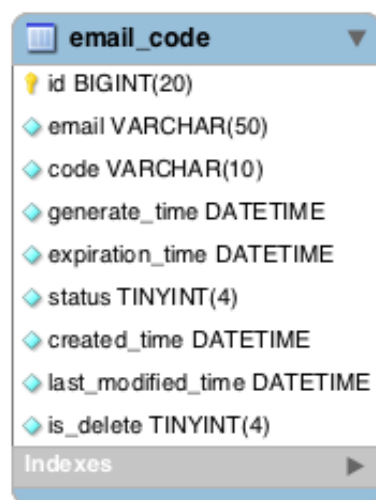


图 5-2 邮件服务的数据库 ER 图

由上述两个分数据库的 10 个数据库表组成了整个系统。

其中，用户表表结构如表 5-1 所示：

表 5-1 用户表

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	用户 id
email	varchar	50	是	否	用户邮箱
nickname	varchar	20	是	否	用户昵称
password	varchar	50	是	否	密码
introduction	varchar	150	是	否	介绍
sex	tinyint	4	是	否	性别
date_of_birth	date	0	是	否	出生日期
province	varchar	20	是	否	所在省
city	varchar	20	是	否	所在市
head_photo	varchar	128	是	否	头像
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

艺人表表结构如表 5-2 所示：

**表 5-2 艺人表**

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	艺人 id
artist_name	varchar	30	是	否	艺人姓名
artist_other_name	varchar	30	是	否	艺人别名
intro	varchar	500	是	否	艺人简介
artist_head_photo_big	varchar	128	是	否	艺人大头像
artist_head_photo_small	varchar	128	是	否	艺人小头像
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

专辑表表结构如表 5-3 所示：

**表 5-3 专辑表**

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	专辑 id
special_name	varchar	30	是	否	专辑名
artist_id	bigint	20	是	否	艺人 id
publish_company	varchar	25	是	否	发行公司
publish_time	date	0	是	否	发行时间
special_photo	varchar	128	是	否	专辑图片
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除



歌曲表表结构如表 5-4 所示：

**表 5-4 歌曲表**

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	歌曲 id
music_name	varchar	30	是	否	歌曲名
special_id	bigint	20	是	否	专辑 id
music_path	varchar	128	是	否	歌曲路径
music_duration	int	11	是	否	歌曲时长
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

用户歌单表表结构如表 5-5 所示：

**表 5-5 用户歌单表**

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	用户歌单表 id
music_list_name	varchar	30	是	否	歌单名
user_id	bigint	20	是	否	用户 id
intro	varchar	150	是	否	简介
generate_time	datetime	0	是	否	歌单创建时间
music_list_photo	varchar	128	是	否	歌单图片
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

用户歌单内歌曲表表结构如表 5-6 所示：

**表 5-6 用户歌单内歌曲表**

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	用户歌单内歌曲表 id
user_music_list_id	bigint	20	是	否	用户歌单 id
music_id	bigint	20	是	否	歌曲 id
user_id	bigint	20	是	否	用户 id
collected_time	datetime	0	是	否	收藏时间
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

歌单收藏表表结构如表 5-7 所示：

**表 5-7 歌单收藏表**

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	收藏表 id
user_id	bigint	20	是	否	用户 id
user_music_list_id	bigint	20	是	否	用户歌单 id
collected_time	datetime	0	是	否	收藏时间
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

歌曲评论表表结构如表 5-8 所示：

**表 5-8 歌曲评论表**

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	歌曲评论 id
music_id	bigint	20	是	否	歌曲 id
user_id	bigint	20	是	否	用户 id
comment_content	varchar	150	是	否	评论内容
comment_time	datetime	0	是	否	评论时间
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

点赞歌曲评论表表结构如表 5-9 所示：

**表 5-9 点赞歌曲评论表**

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	点赞表 id
user_id	bigint	20	是	否	用户 id
comment_id	bigint	20	是	否	评论 id
like_time	datetime	0	是	否	评论时间
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

邮件验证码表结构如表 5-10 所示：

表 5-10 邮件验证码表

列名	数据类型	长度	非空	主键	注释
id	bigint	20	是	是	邮件验证码表 id
email	varchar	50	是	否	邮箱号
code	varchar	10	是	否	邮箱验证码
generate_time	datetime	0	是	否	生成时间
expiration_time	datetime	0	是	否	过期时间
status	tinyint	4	是	否	状态
created_time	datetime	0	是	否	创建时间
last_modified_time	datetime	0	是	否	最近修改时间
is_delete	tinyint	4	是	否	是否删除

5.2 系统功能实现

5.2.1 用户模块实现

本模块的主要功能是：（1）提供给新用户注册账号的功能，让新用户能够注册使用此系统；（2）提供给老用户忘记密码的功能，让老用户能够找回并充值账号密码；（3）提供给用户个人设置的功能，让用户能够进行个性化设置信息；（4）提供给用户登录的功能，能够让用户登录使用此系统。

注册页面如图 5-3 所示：

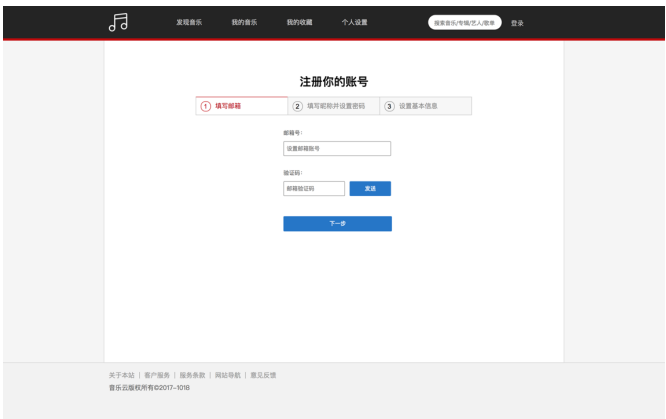


图 5-3 注册页面

忘记密码页面如图 5-4 所示：

重置你的密码

1 验证邮箱 2 重置密码

邮箱号:  
邮箱号

验证码:  
邮箱验证码 发送

下一步

关于本站 | 客户服务 | 服务条款 | 网站导航 | 意见反馈  
音乐云版权所有©2017-1018

图 5-4 忘记密码页面

用户个人设置页面如图 5-5 所示：

个人设置

基本设置 安全设置

昵称: sam199510

介绍:

性别: ☐ 男 ☐ 女 ☒ 保密

生日: 1995 年 10 月 20 日

地区: 浙江 杭州

保存

关于本站 | 客户服务 | 服务条款 | 网站导航 | 意见反馈  
音乐云版权所有©2017-1018

图 5-5 用户个人设置页面

登录页面如图 5-6 所示：

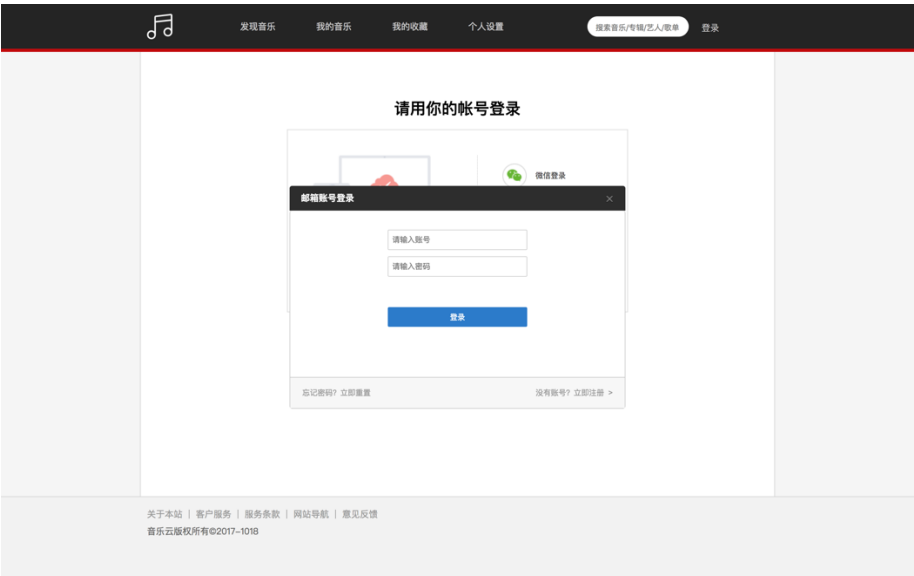


图 5-6 登录页面

### 5.2.2 发现音乐模块实现

本模块的主要功能是：（1）给用户提供输入关键词，可按照单曲、歌手、专辑、歌单和用户这五个分类搜索的功能；（2）给用户提供点击各个分类的每个模块，进入各个不同的页面，实现用户不同需求的功能；（3）艺人、专辑和单曲三块内容的展示；（4）给用户提供给单曲的评论和点赞的功能。

按单曲搜索的实现如图 5-7 所示：

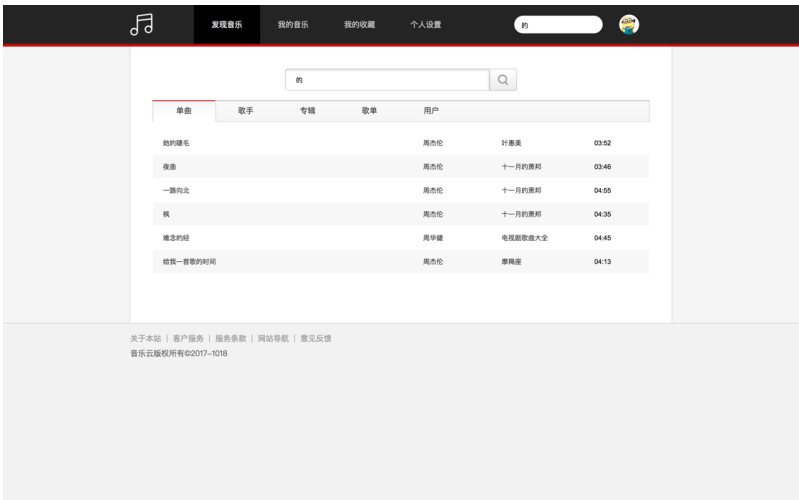


图 5-7 按单曲搜索

按歌手搜索的实现如图 5-8 所示：

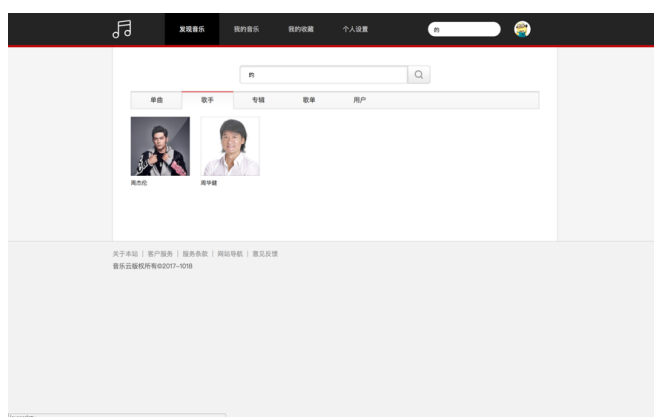


图 5-8 按歌手搜索

按专辑搜索的实现如图 5-9 所示：

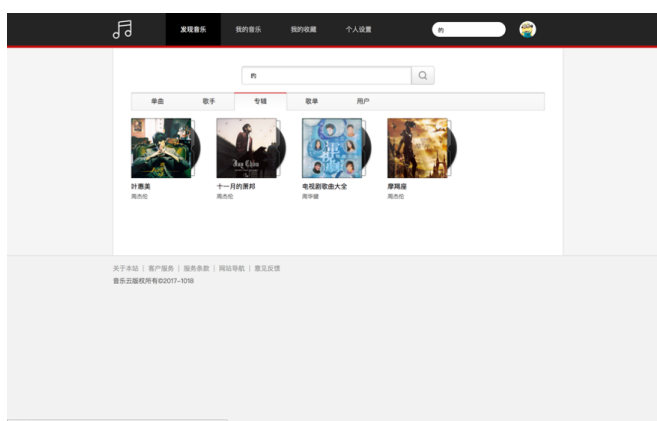


图 5-9 按专辑搜索

按歌单搜索的实现如图 5-10 所示：

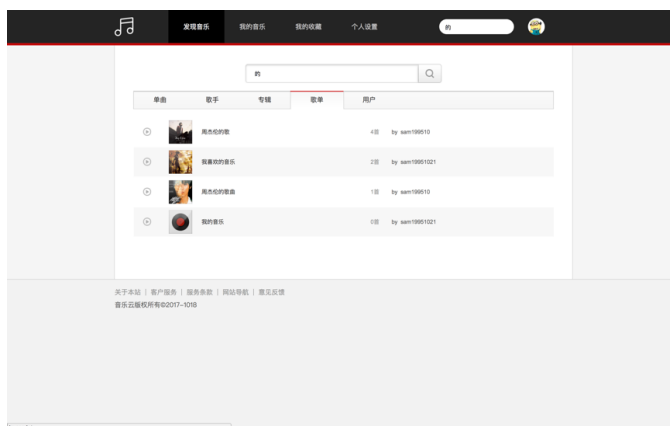


图 5-10 按歌单搜索

按用户搜索的实现如图 5-11 所示：

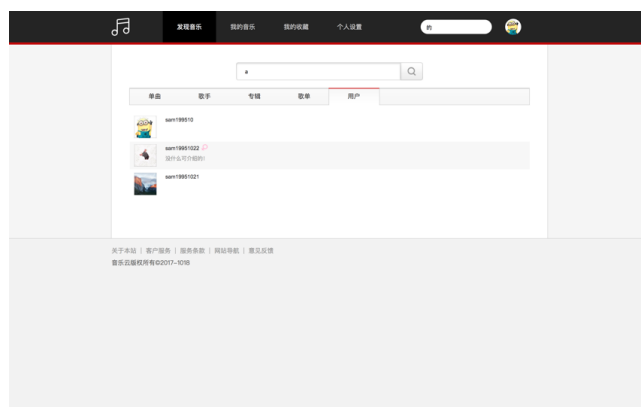


图 5-11 按用户搜索

艺人页面的实现如图 5-12 所示：

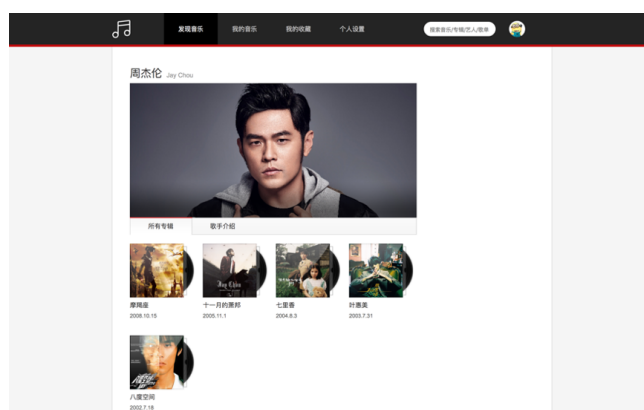


图 5-12 艺人页面

专辑页面的实现如图 5-13 所示：

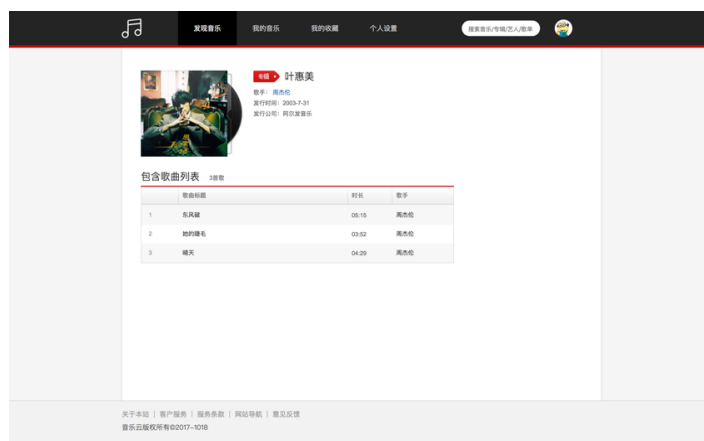


图 5-13 专辑页面



单曲页面的实现如图 5-14 所示：

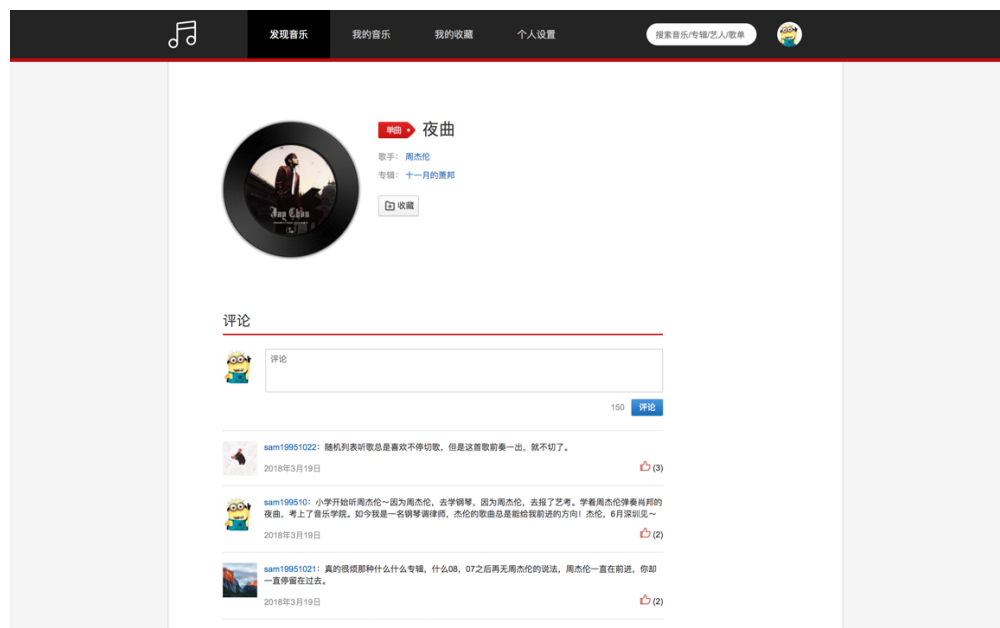


图 5-14 单曲页面

### 5.2.3 用户创建的歌单模块实现

本模块的主要功能是：（1）给用户新建歌单的功能；（2）给用户编辑歌单的功能；（3）给用户删除歌单的功能；（4）给用户管理歌单的功能；（5）给用户播放歌单按钮的功能。

创建歌单的页面如图 5-15 所示：

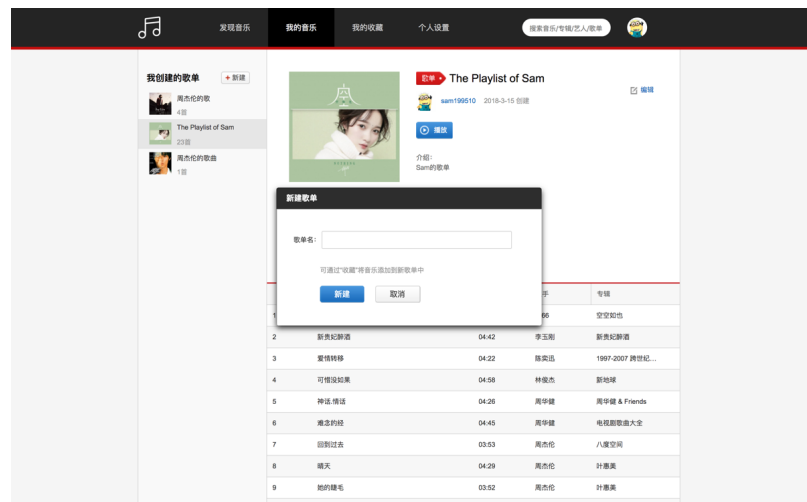


图 5-15 创建歌单

编辑歌单的页面如图 5-16 所示：

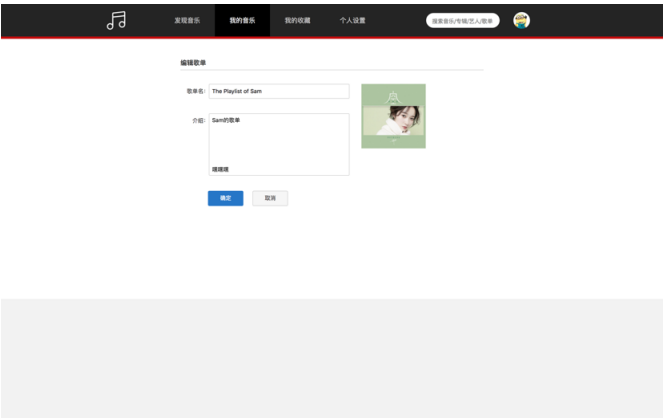


图 5-16 编辑歌单

删除歌单的页面如图 5-17 所示：

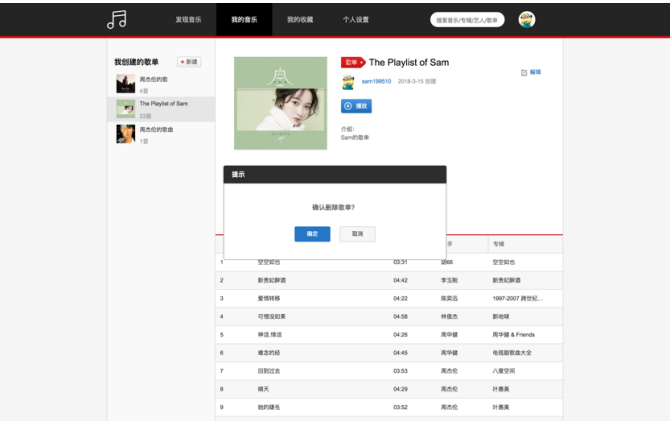


图 5-17 删除歌单

管理歌单的页面如图 5-18 所示：

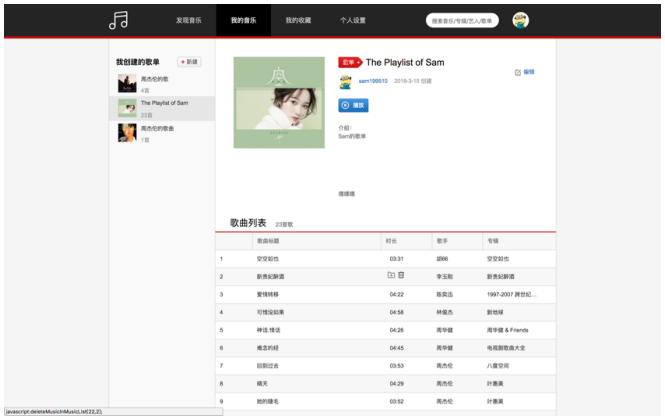


图 5-18 管理歌单

### 5.2.4 用户收藏的歌单模块实现

本模块的主要功能是：（1）给用户提供弹窗取消收藏歌单的功能；（2）给用户提供查看歌单内歌曲的功能；（3）给用户提供播放歌单的按钮的功能。

取消收藏歌单的功能如图 5-19 所示：

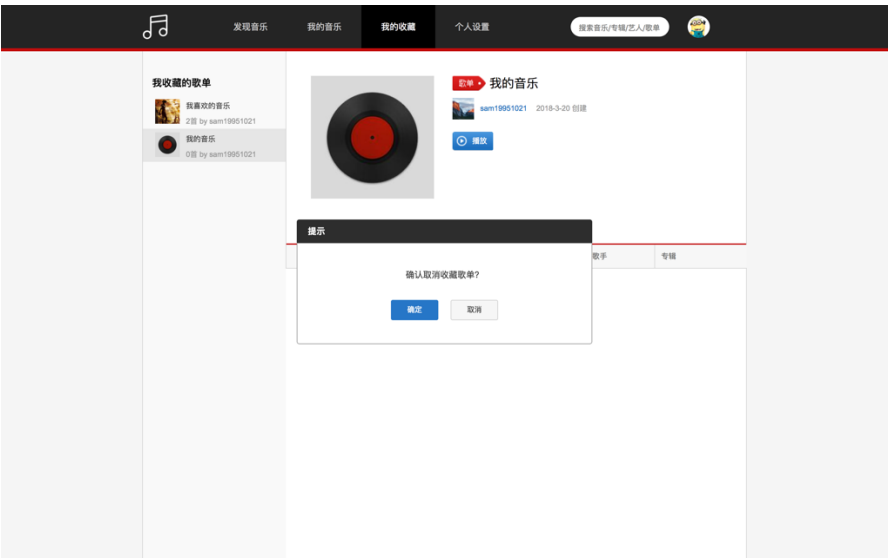


图 5-19 取消收藏歌单

查看收藏歌单的功能如图 5-20 所示：

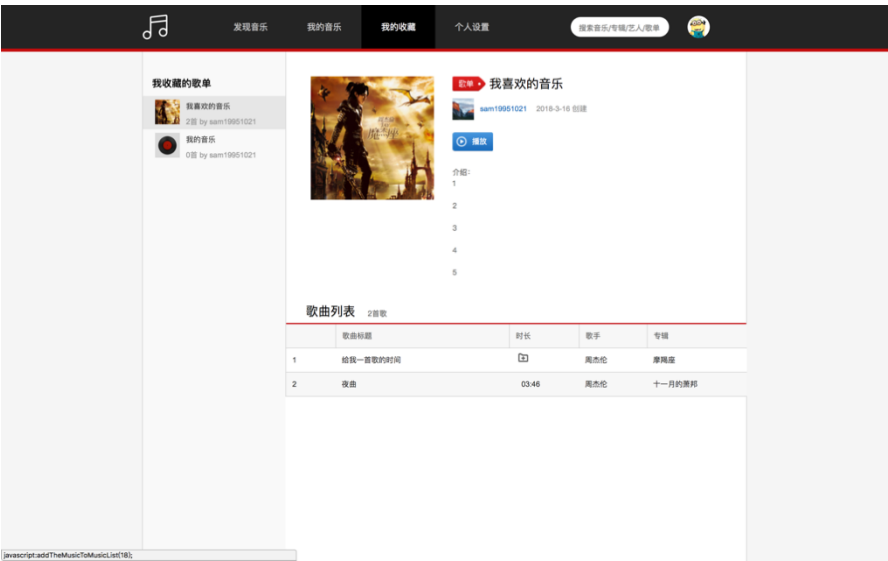


图 5-20 查看收藏歌单

### 5.2.5 其他用户模块实现

本模块的主要功能是：（1）给用户提供查看其他用户的信息、创建的歌单和收藏的歌单的信息的功能；（2）给用户提供查看其他用户创建或收藏的某个歌单的歌单信息和歌单内歌曲内容的功能。（3）给用户提供播放歌单的按钮的功能；（4）给用户提供收藏歌单的按钮的功能。

其他用户的页面如图 5-21 所示：

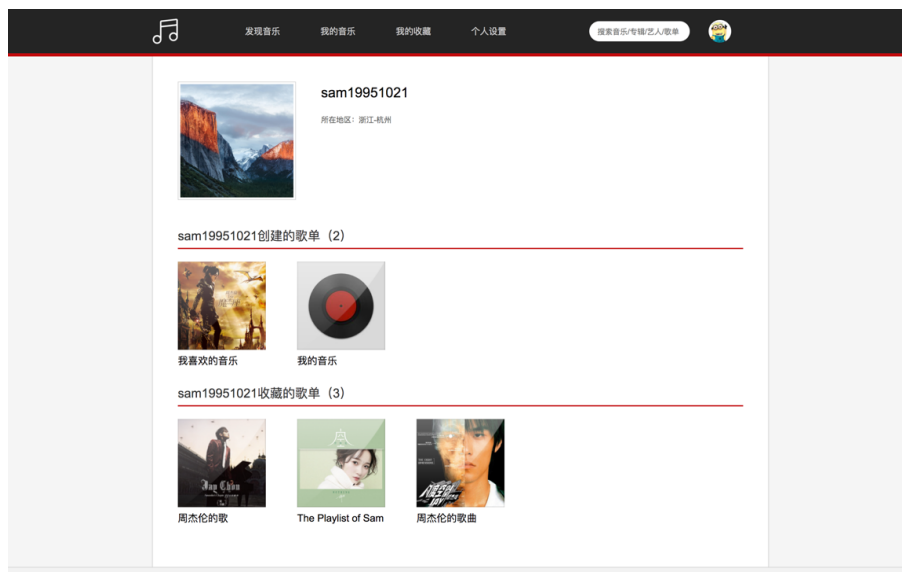


图 5-21 其他用户页面

其他用户创建的歌单的页面如图 5-22 所示：

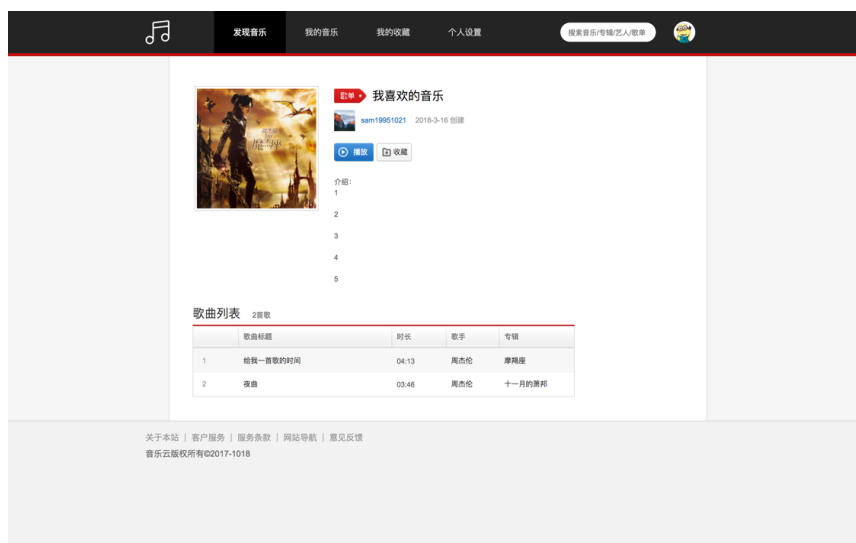


图 5-22 其他用户创建的歌单的页面

### 5.2.6 音乐播放模块功能实现

本模块主要给用户提供的功能是：（1）给用户查看和播放歌单内歌曲以及查看正在播放的歌曲的概要信息的功能；（2）给用户播放音乐播放条的功能（其中播放条有上一曲/下一曲、播放/暂停、进度条与播放进度、音量和播放模式的功能）。

播放音乐的页面实现如图 5-23 所示：

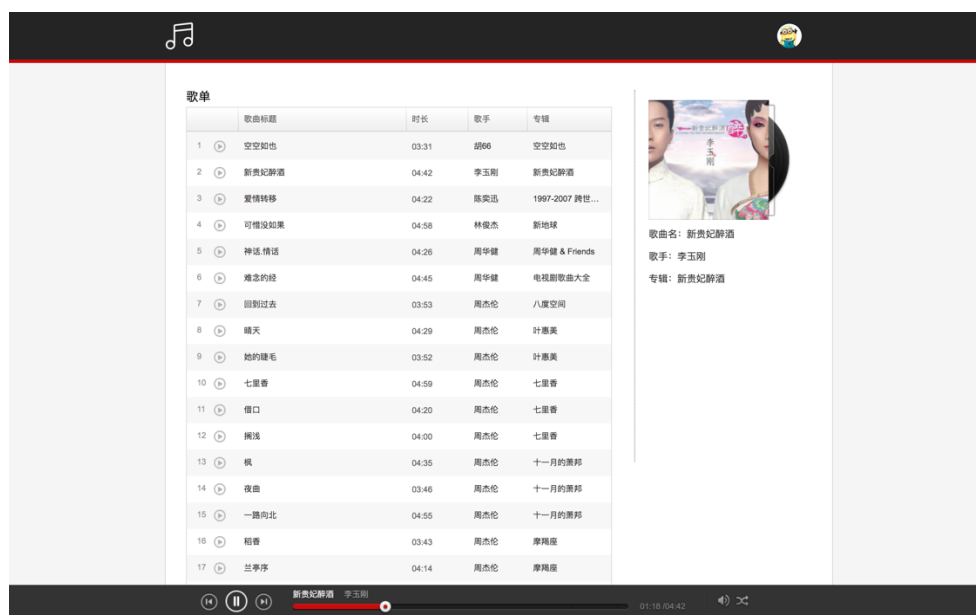


图 5-23 音乐播放页面

## 5.3 系统接口实现

本系统在服务上采用了 Spring Cloud，所以服务分开治理，本系统由邮件发送验证服务和音乐网站主服务组成。

### 5.3.1 邮件发送验证服务接口实现

邮箱验证服务主要由邮箱发送验证一个接口控制器组成。

本系统的邮件发送验证服务接口组成如表 5-11 所示：

**表 5-11 邮件发送验证服务接口**

接口所属	接口及用途
邮箱验证码发送及验证类	/gdemailserver/email/code/@check 邮件验证码验证
	/gdemailserver/email/code/@send 注册邮件验证码发送
	/gdemailserver/reset/password/email/code/@send 重置密码邮件验证码发送

### 5.3.2 音乐网站主服务接口实现

音乐网站主服务由收藏音乐相关、用户歌单内歌曲相关、用户歌单相关、用户相关、艺人和专辑与音乐相关、艺人专辑相关、艺人相关、邮箱验证码相关、音乐播放列表相关、音乐相关以及音乐评论相关接口等十一个接口控制器组成。

本系统的音乐网站主服务接口组成如表 5-12 所示：

**表 5-12 音乐网站主服务接口**

接口所属	接口及用途
收藏音乐相关	/gdmusicserver/collect/other/user/music/list/@collect 收藏别的用户创建的歌单的接口
	/gdmusicserver/show/collected/other/user/music/list/by/user/id/@query 显示收藏别的用户创建的歌单的接口
	/gdmusicserver/un/collect/other/user/music/list/@uncollect 取消收藏别的用户创建的歌单的接口
用	/gdmusicserver/get/music/in/user/music/list/@query

户 歌 单 内 歌 曲 相 关	获取我的歌单的单个歌单的歌曲列表接口
用 户 歌 单 相 关	/gdmusicserver/collect/music/into/user/music/list/@collect 添加歌曲接口
	/gdmusicserver/find/like/user/music/list/name/@query 根据用户歌单名查询歌单接口
	/gdmusicserver/find/user/music/list/by/music/list/id/@query 根据用户歌单名查询歌单接口
	/gdmusicserver/find/user/music/list/by/user/id/@query 根据用户 id 查找创建的歌单的信息
	/gdmusicserver/show/a/user/music/list/by/user/music/list/id/@query 显示一个歌单的具体信息的接口
	/gdmusicserver/un/collect/music/into/user/music/list/@uncollect 删除歌单内某一首歌的接口
	/gdmusicserver/update/user/music/list/@update 歌单信息更新接口
	/gdmusicserver/user/music/list/@create 用户创建歌单
	/gdmusicserver/user/music/list/@delete 删除用户歌单接口
用 户 相 关	/gdmusicserver/show/user/page/info/and/num/of/created/and/collected/@query 根据用户 id 获取用户页面的基本信息和歌单数量信息的接口

关	/gdmusicserver/user/service/@login 登录接口
	/gdmusicserver/user/service/@register 注册接口
	/gdmusicserver/user/service/basic/info/@change 个人基础信息更改接口
	/gdmusicserver/user/service/find/like/nickname/@query 根据用户名模糊搜索接口
	/gdmusicserver/user/service/head/photo/@change 更改头像接口
	/gdmusicserver/user/service/info/@get 获取个人信息接口
	/gdmusicserver/user/service/password/@forget 重置密码接口
	/gdmusicserver/user/service/password/@reset 个人密码修改接口
艺 人、 专 辑 和 音 乐 相 关	/gdmusicserver/artist/special/music/find/like/artist/name/@query 根据歌手名模糊搜索信息接口
	/gdmusicserver/artist/special/music/find/like/music/name/@query 根据歌曲名模糊搜索信息接口
	/gdmusicserver/artist/special/music/find/like/special/name/@query 根据专辑名模糊搜索信息接口
	/gdmusicserver/find/music/by/music/id/@query 根据歌曲 id 获取音乐信息接口
艺 人、 专 辑 和	/gdmusicserver/find/music/in/special/by/special/id/@query 根据专辑 id 获取专辑内歌曲列表的接口
	/gdmusicserver/find/special/by/special/id/@query 根据专辑 id 获取专辑信息接口
	/gdmusicserver/find/specials/by/artist/id/@query



音乐相关	根据艺人 id 查找专辑列表接口
艺人相关	/gdmusicserver/find/artist/by/artist/id/@query 根据艺人 id 获取艺人信息接口
邮箱验证码相关	/gdmusicserver/email/@check 邮箱验证码验证接口
	/gdmusicserver/email/@get 注册邮件验证码获取接口
	/gdmusicserver/reset/password/email/@get 忘记密码邮件验证码发送接口
音乐播放列表相关	/gdmusicserver/get/music/play/list/@query 获取音乐播放列表接口相关
音乐评论相关	/gdmusicserver/comment/music/@comment 音乐评论接口
	/gdmusicserver/delete/music/comment/@delete 删除音乐评论接口
	/gdmusicserver/get/music/comment/by/music/id/or/user/id/@query 音乐评论显示相关接口
	/gdmusicserver/like/music/comment/@like

	音乐评论点赞接口
	/gdmusicserver/un/like/music/comment/@unlike
	音乐评论取消点赞接口

## 第 6 章 系统测试与维护

### 6.1 系统的测试

本系统采用从黑盒测试到单元测试再到系统测试进行对整个系统各个功能的测试。对每个步骤的测试逐一通过后再执行下一阶段的测试，对于各个阶段的测试归结如下。

#### 6.1.1 系统的黑盒测试

在软件测试中，一开始是将一个软件看做是一个不知道内部是如何运作的黑盒，它完全不用我们考虑软件内部是如何运作的。我们只需要设计黑盒测试的测试用例来实现对软件的黑盒化测试。

对于系统中的用户创建的歌单模块，采用了黑盒化的测试方案，得到了如下表所示的测试用例。

表 6-1 用户创建的歌单模块的黑盒测试

测试项	输入	预测输出	真实结果
新建歌单功能	点击新建按钮，输入歌单名，点击新建按钮	歌单列表中新增了刚刚创建的歌单	歌单列表中新增了之前创建的歌单了
编辑歌单功能	点击编辑歌单按钮，输入歌单信息，点击确定按钮	歌单信息被更新	歌单信息被更新了
删除歌单功能	点击删除歌单按钮，点击确定按钮	歌单被删除	歌单被删除了
播放歌单功能	点击播放按钮	跳转到播放音乐页面并开始播放音乐	跳转到了音乐播放页面并开始播放音乐
切换歌单功能	点击需要切换的歌单	跳转到需要切换到的歌单	切换到了需要切换的歌单

管理歌单功能	点击添加按钮，选择并点击需要添加到的歌单	歌曲被添加到想要添加到的歌单，如果已存在，则会提示“歌曲已存在”。如果未存在，则会提示“收藏成功”	如果歌曲已存在，提示“歌曲已存在”，不允许添加。如果歌曲未被添加，提示“歌曲添加成功”
	点击删除按钮，点击确定按钮	歌单中的该歌曲被删除	歌单中想删掉的歌曲被删除

### 6.1.2 系统的单元测试

单元测试主要是针对于软件层面的最小单元的代码层面的测试。在本系统中，主要针对的是对 service 业务层的接口方法测试。

这里以测试获取用户歌单内歌曲的接口为测试用例进行测试。单元测试代码如图 6-1 所示：

```

19 @SpringBootTest(classes = GdMusicServerApplication.class)
20 @RunWith(SpringJUnit4ClassRunner.class)
21 public class UserMusicListTest {
22
23     @Autowired
24     MusicInUserMusicListService musicInUserMusicListService;
25
26     @Test
27     public void testGetUserMusicList() {
28         Long userMusicListId = 2L;
29         List<ArtistSpecialMusicDto> artistSpecialMusicDtos = this.musicInUserMusicListService.findMusicByUserMusicListId(userMusicListId);
30         System.out.println(JSON.toJSONString(artistSpecialMusicDtos, prettyFormat: true));
31     }
32 }

```

图 6-1 获取用户歌单内歌曲的单元测试代码

得到如图 6-2 所示的单元测试结果：

图 6-2 获取用户歌单内歌曲的单元测试结果

### 6.1.3 系统的系统测试

系统测试主要针对的是在单元测试通过的基础之上对于整个系统的各个单元进行整合后的测试。在系统测试中，主要包括功能测试和健壮性测试这两方面的

测试内容。功能性测试中主要是测试功能是否能按照预期的结果进行，而健壮性测试则表现在容错能力测试和恢复能力测试方面。

在本系统的系统测试中，功能测试上，通过对各个系统模块进行测试，功能上达到了系统需求的目的。健壮性测试上，通过对各个模块的测试，测试下来的结果达到了需求的目的，系统中对于用户的错误操作能够有一定的容错能力。

## 6.2 系统的维护

本系统采用前后端分离的设计架构。在系统维护方面，比较容易，如果再执行过程中，出现问题时，如果是前端的责任则可以找前端来维护，如果是服务端的问题则可以找服务端人员进行维护。

在本系统的后期，还需要考虑到的是日后不断增长的用户需求下，系统的并发能力和数据库并发事务能力等方面的问题需要得到解决。

因为目前音乐存在各种版权的问题，所以本系统后续还需要解决音乐版权管理的问题。

## 第 7 章 结束语

本文主要讲述的是基于 Java Web 的在线音乐网站的整个开发过程。从可行性分析与需求分析到开发技术介绍，再到系统模块的概要设计与详细设计，接着到系统的数据库实现与页面上的具体实现，最后到系统的测试与系统的维护的整个软件开发的过程。

其中，基本上实现了需求分析中所需要达到的功能，包括用户、用户创建的歌单、用户收藏的歌单、发现音乐、其他用户模块和音乐播放模块的功能。

回首整个系统的开发过程，总结出来的结论有：

（1）前端方面在编写代码的时候需要提高界面的友好程度，使用户在使用时具有良好的用户体验。

（2）服务端方面在编写代码的时候系统模块之间的耦合度不能过高，系统模块内的内聚程度不能太低，否在系统的功能模块就达不到需求。

（3）数据库方面，设计数据库表格的时候需要将表结构设计得合理些，否则关系就比较混乱。

（4）业务方面，在设计与实现系统的功能时，需要仔细去分析系统的功能实现的逻辑，想好了再下手，尽量避免业务出错。

在整个设计与实现的过程中，遇到了很多问题，不过在指导老师、同学的帮助下，同时也经过查阅大量的资料和文献，使问题被各个击破，难题也迎刃而解了。

与此同时，在设计的过程中，自己的设计与开发软件的能力得到了一定的提升。但是骄兵必败，软件系统中也必定会存在着一定的不足之处，所以需要提升自己的软件维护和后续升级的能力。在设计与开发软件程序时，需要时时刻刻以用户体验为第一准则，这样最后的软件才能达到客户的效果。

## 致谢

写到此，即代表毕业设计以及毕业设计论文的即将完结。同时，他也宣告了我即将进入人生的另一个阶段。毕业设计是对于大学四年学业的总结，也是奠定大学四年之后的另一个阶段的一个基础。

在毕业设计完成的同时，也要感谢周边很多对象给予我的帮助。

首先我要感谢的对象是我的毕业导师孙晓勇老师，感谢他对我毕业设计的帮助。在毕业设计的开题阶段、中期检查、设计过程等各个阶段给予我的帮助，感谢他在这几个阶段中给我提的要求和建议，给予我的帮助和技术支持。也同时感谢大学四年中教育过我的所有教师，从他们所教的课程中和他们身上，我学习到了去适应社会的能力和对人对事的高贵品质。

其次我要感谢的对象是我的家人，感谢他们当初支持我选择这个专业，能够让我在软件工程专业这个平台上拥有自己的发展。

接着我要感谢的对象是我身边的同学和同事，感谢他们给我提供的各种帮助和技术支持，让我在毕业设计的技术中提升了一定的技术能力。

同时，我也要感谢浙江科技学院这个大学，感谢她给我提供了一个能够让我在大学四年中表现自己的一个平台。同时她也培养了我沟通、适应和技术等各方面的能力，能够让我在未来的人生中有立足之地。

大学四年这趟拥有四年的班车即将到达终点站，看似很长，但是他相比于人生这条漫长的道路来说，也只是其中的一小节。虽然中间遇到过意外，但是凭着自己的努力，开完了这趟班车，到达了学士学位这个终点站，也宣告我们即将转到另一辆班车中。最后，感谢大学四年这趟班车，因为有你，才有了山青水绿的大学旅途。

## 参考文献

- [1] Salvador Martínez, Valerio Cosentino, Jordi Cabot. Model-based analysis of Java EE web security misconfigurations[J]. Computer Languages, Systems & Structures, 2017, 49:36-61.
- [2] Fisher M, Partner J, Bogoevici M, et al. Spring Integration in Action[J]. Annals of the Rheumatic Diseases, 2015, 74(Suppl 2):229-229.
- [3] 徐金虎, 宋斌, 丁锐. Spring MVC+Hibernate+jQuery 模式开发框架应用研究[J]. 自动化技术与应用, 2015, 34(3).
- [4] 杨慧. 基于 SpringMVC 与 iBATIS 框架的企业工单管理系统的设计与实现[D]. 中国科学院大学(工程管理与信息技术学院), 2015.
- [5] 王佳康, 李明, 买尔夏提·尔肯. 基于 Spring MVC 架构的旅游攻略网站前后台的设计与实现[J]. 电脑知识与技术, 2015(8).
- [6] 宋涛, 王洪鑫, 徐庆增. J2EE 平台标准下的 SPRING 3.0 轻量级框架技术概述[J]. 通讯世界, 2015(12):306-307.
- [7] 孙莹, 许俊华, 张毅,等. MVC 编程模型在 Web 程序中的应用及 Java 实现[J]. 计算机工程与应用, 2001, 37(17):160-163.
- [8] 李文杰, 姜淑娟, 钱俊彦,等. 基于对象引用关系的 Java 程序内存行为分析方法[J]. 电子学报, 2015, 43(7):1336-1343.
- [9] 王正玉, 李斌. 基于 DAO 模式的 Hibernate 框架在 Java Web 开发中的应用[J]. 微型机与应用, 2015(11):14-17.
- [10] 郑强. 基于 J2EE 技术的 Hibernate 架构优化策略探讨[J]. 中国科技博览, 2015(5):238-238.
- [11] 田丽华, 孙颖馨, 王金甫. Spring 框架下 IoC 容器的在线考试系统设计[J]. 重庆理工大学学报:自然科学版, 2015, 29(1):82-84.
- [12] (美)施瓦茨(Schwartz, B.), (美)扎伊采夫(Zaitsev,等. 高性能 MySQL[M]. 电子工业出版社, 2013.
- [13] 丁宋涛. JavaEE 案例教程[M].北京: 北京大学出版社, 2015.
- [14] 张海藩.软件工程导论[M].北京: 清华大学出版社, 2013.
- [15] 郭克华. JavaEE 程序设计与应用开发[M].北京: 清华大学出版社, 2011.