武 汉 轻 工 大 学

数学与计算机学院

JAVA程序设计 课程设计报告

专 业： 软件工程

班 级： 1601

学 号： 1605110125

姓 名： 张 尹

目录

[1.课程设计概述 1](#_Toc353940)

[1.1 系统的主要目标 1](#_Toc353941)

[1.2 系统的开发环境及运行环境 1](#_Toc353942)

[2.系统的需求分析 1](#_Toc353945)

[2.1 系统可行性分析 1](#_Toc353946)

[2.2 系统功能描述 1](#_Toc353947)

[2.3 系统的数据流图 2](#_Toc353952)

[2.4 系统UML建模设计 2](#_Toc353953)

[2.5 系统的状态图 3](#_Toc353954)

[2.6 系统的UML类图 3](#_Toc353955)

[3.系统总体设计 4](#_Toc353956)

[3.1 系统结构方框图 4](#_Toc353957)

[3.2 各模块功能 4](#_Toc353958)

[3.3 详细的UML类图 5](#_Toc353963)

[3.4 设计数据管理子系统 9](#_Toc353968)

[3.5 系统E-R图 9](#_Toc353969)

[3.6 数据库表 10](#_Toc353970)

[3.7 数据库表之间的关系 10](#_Toc353971)

[3.8 数据库表结构 11](#_Toc353974)

[3.9 设计人机交互子系统 12](#_Toc353979)

[4.详细设计 17](#_Toc353983)

[4.1登陆注册模块流程图 17](#_Toc353984)

[4.2在线搜索模块流程图 17](#_Toc353985)

[4.3本地乐库模块流程图 17](#_Toc353986)

[4.4歌单管理模块流程图 18](#_Toc353987)

[5. 系统实现 18](#_Toc353988)

[6. 总结 19](#_Toc353989)

7. [参考文献： 19](#_Toc353990)

# 

# 课程设计概述

## 1.1系统的主要目标

基于javaEE的在线音乐系统。

## 1.2系统的开发环境及运行环境

### 开发环境:

Windows10、Eclipse、Tomcat 9.0、Mysql 8、JDK 10

### 运行环境:

Tomcat 9.0、Mysql 8 、Chrome 71

# 2.系统的需求分析

## 2.1系统可行性分析

本系统是采用Java Web技术的B-S架构的网站，用到的技术有Java EE 、CSS、JavaScript、Ajax。为了使用的效果以及开发的简便，在前端使用了CSS的Bootstrap及其UI组件框架，JS的Jquery库以及Ajax技术。在线音乐是获取的外部站点的API提供查询服务，并且每一次播放在线音乐会将其数据保存到本地数据库。本地音乐是从本地数据库中获取数据展现到用户界面。

## 2.2 系统功能描述

### 2.2.1 用户管理：

用户注册：用户可以点击注册按钮进入注册界面，注册属于自己的账号。

用户登录：用户使用自己的账号登录系统。

用户设置：可点击个人设置，修改个人信息，例如昵称，头像等。

### 2.2.2播放界面：

音乐播放[播放与暂停]：对在播放器中的音乐进行播放和暂停。

音量调节：对正在播放器中的音乐的音量大小进行调节。

### 2.2.3歌单与音乐管理：

歌单管理：对歌单的增删查改。

音乐管理：将音乐添加进歌单，将音乐从歌单中删除。

### 2.2.4音乐搜索：在线搜索、系统乐库

在线搜索：在网络上搜索获取音乐资源。

系统乐库：获取系统所在服务器上的音乐资源。

## 2.3系统的数据流图

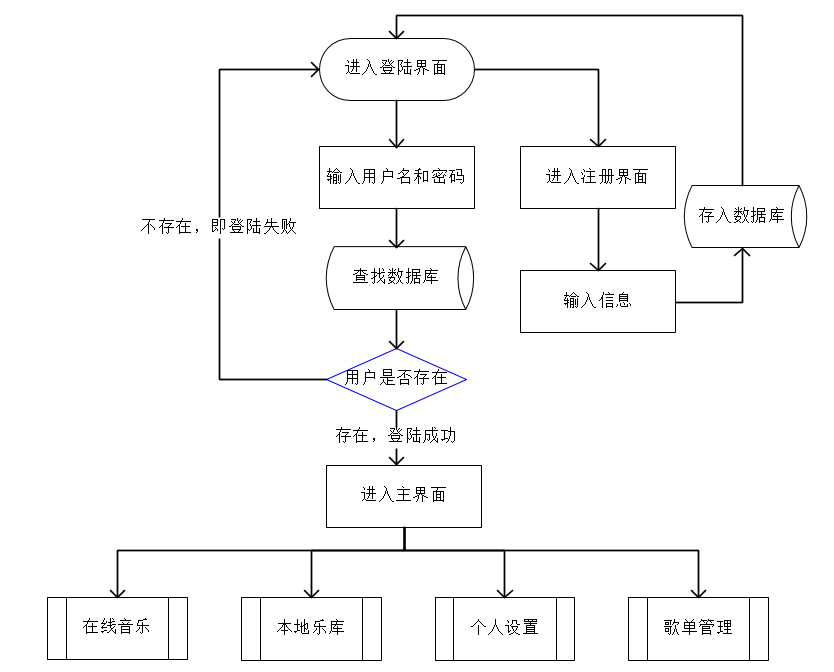


图2.3 数据流图

## 2.4系统UML建模设计

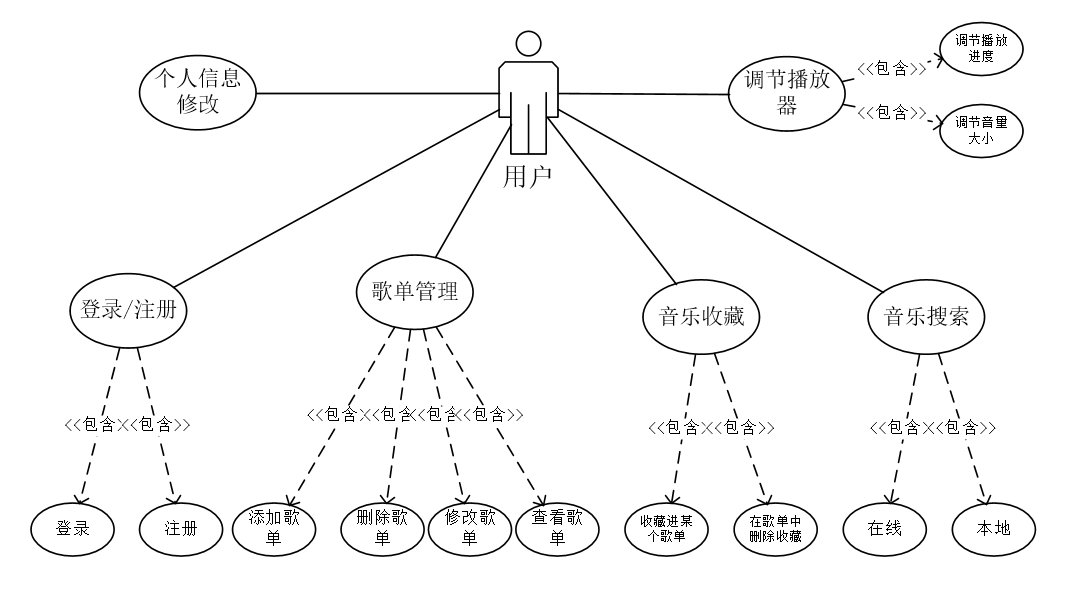


图2.4 UML用例图

## 2.5系统的状态图

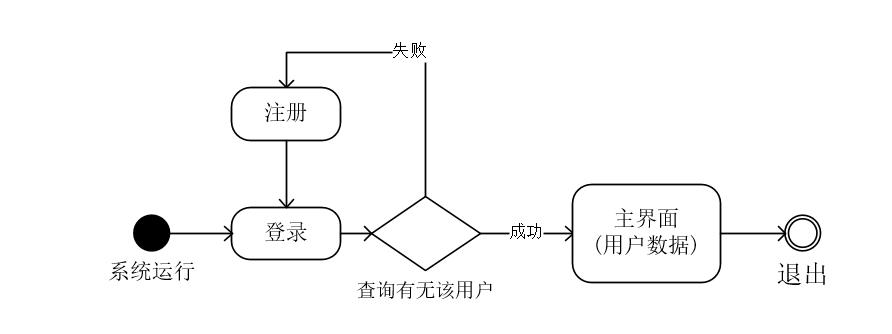


图2.5.1 登陆模块状态图

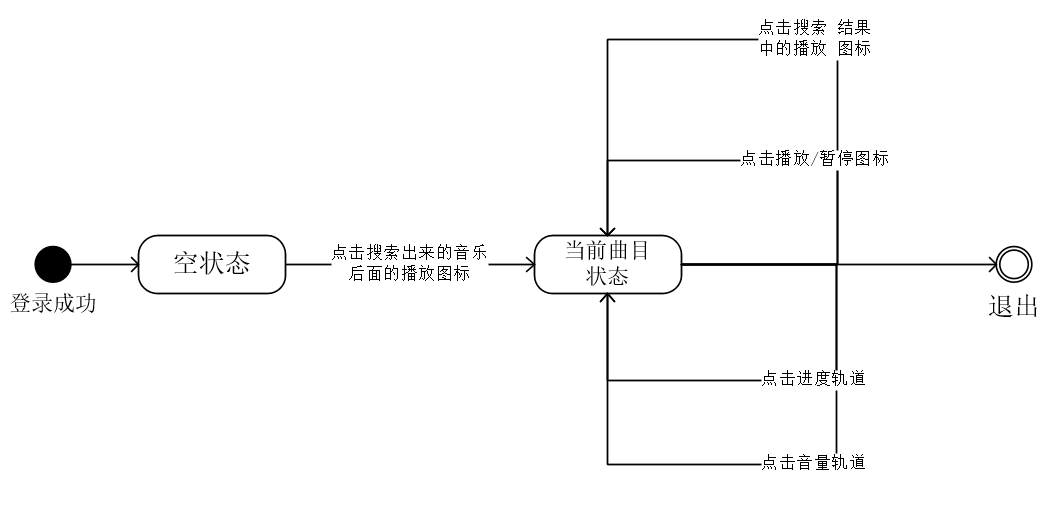


图2.5.2 播放器状态图

## 2.6系统的UML类图

本系统后台总共使用了四个包，其功能与关系如下。

utils:封装了连接数据库和关闭数据库的操作。

dto:包含封装了某个对象的所有信息的类。

dao:封装了对数据库的操作。

servlets:包含处理前端发送的各种信息的servlet。

以上四个包都位于cn.edu.whpu.music包下，它们的关系如图2.6所示。

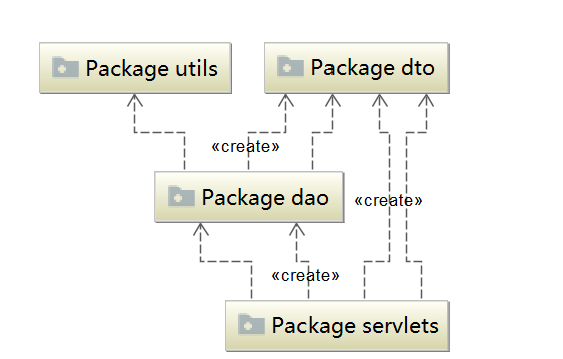


图2.6 系统所用的四个包之间的关系

# 3.系统总体设计

## 3.1 系统结构方框图

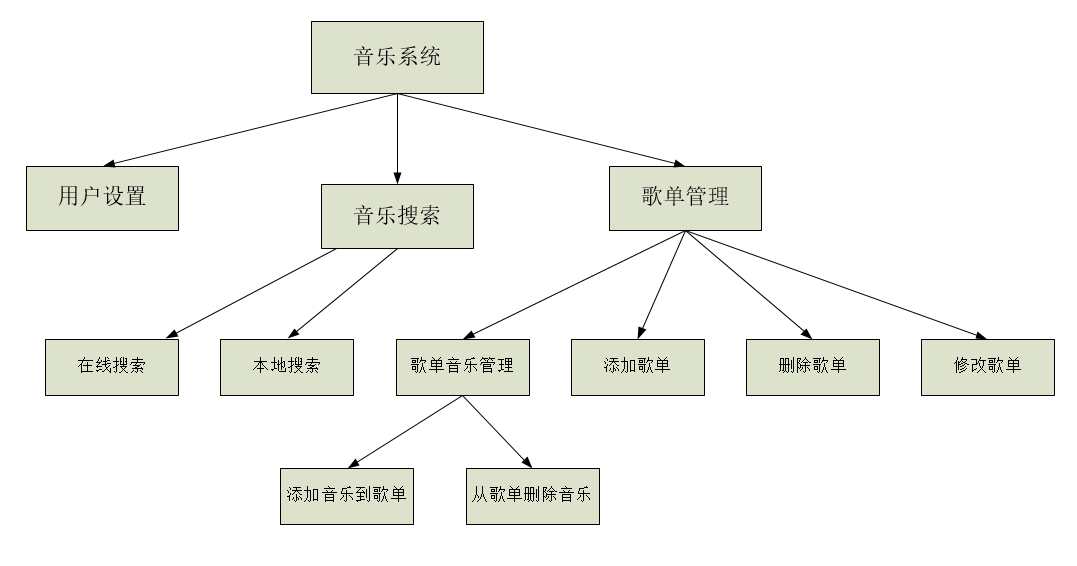


图3.1 系统结构方框图

## 3.2 各模块功能

### 用户信息模块：

包含功能有用户的注册，用户的登录，用户个人信息的修改和用户个人信息的展示。

### 搜索音乐模块：

包含功能有在线音乐搜索，本地乐库搜索。

### 歌单管理模块：

包含功能有添加新的歌单，修改歌单信息，删除歌单。

### 收藏歌曲模块：

包括添加音乐到指定歌单，从歌单中删除音乐。

## 3.3 详细的UML类图

### 用户信息模块:

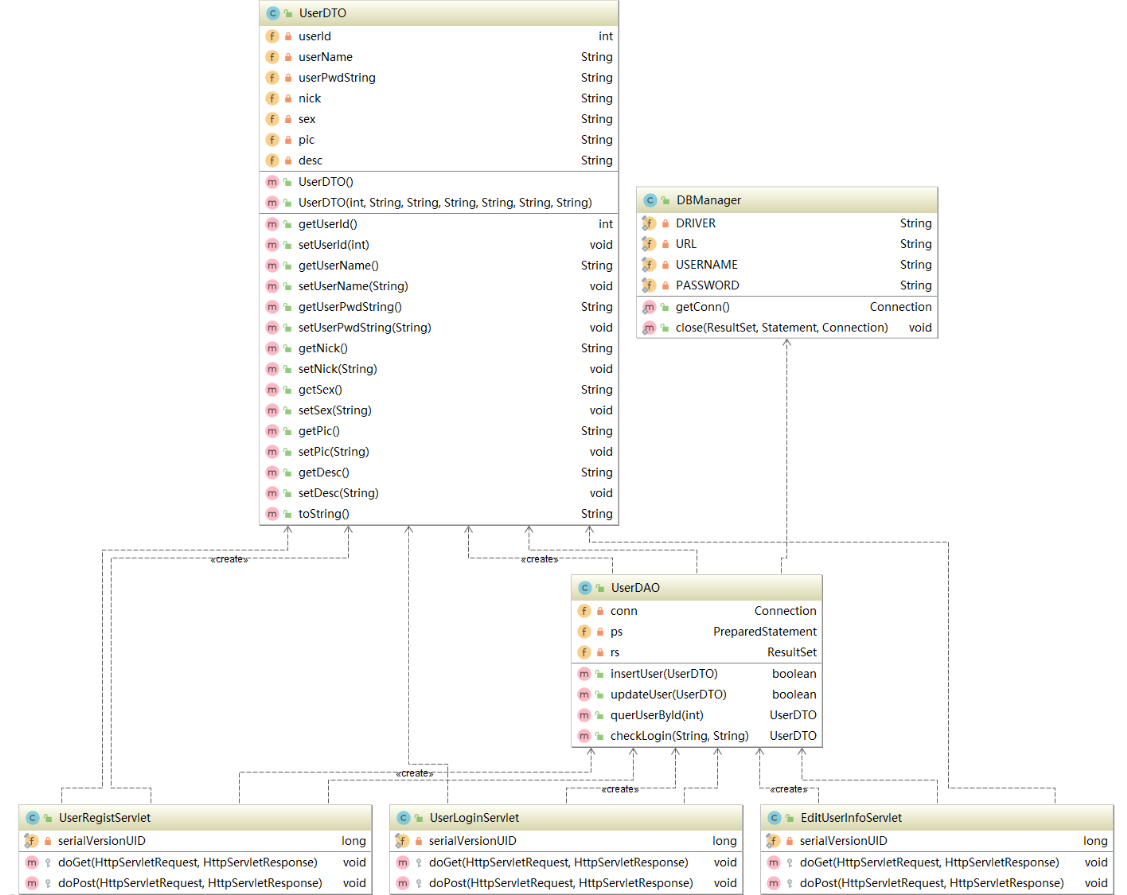


图3.3.1 用户信息模块类图

DBManager：封装了连接数据库和关闭数据库的操作。

UserDTO：封装了用户个人信息的类。

UserDAO：封装了操作用户个人信息的类。

UserRegistServlet：用户注册的servlet。

UserLoginServlet：用户登录的servlet。

EditUserInfoServlet：修改用户个人信息的servlet。

### 搜索音乐模块(本地乐库)：

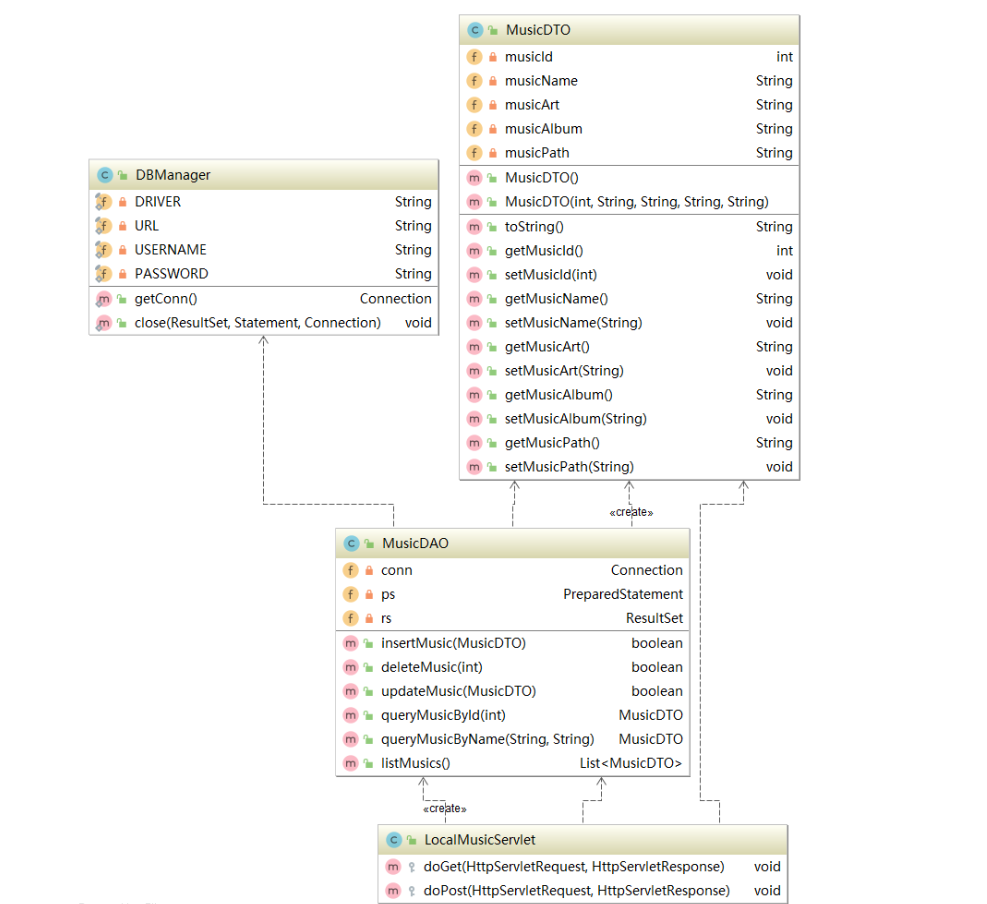


图3.3.2 搜索音乐模块类图

DBManager：封装了连接数据库和关闭数据库的操作。

MusicDTO：封装了音乐信息的类。

MusicDAO：封装了操作音乐信息的类。

LocalMusicServlet：处理点击本地乐库按钮后的请求并返回数据的servlet。

注：在线音乐搜索是用ajax请求的网络上的API接口获取数据并渲染到主页面，与该处的servlet无关。

### 歌单管理模块：

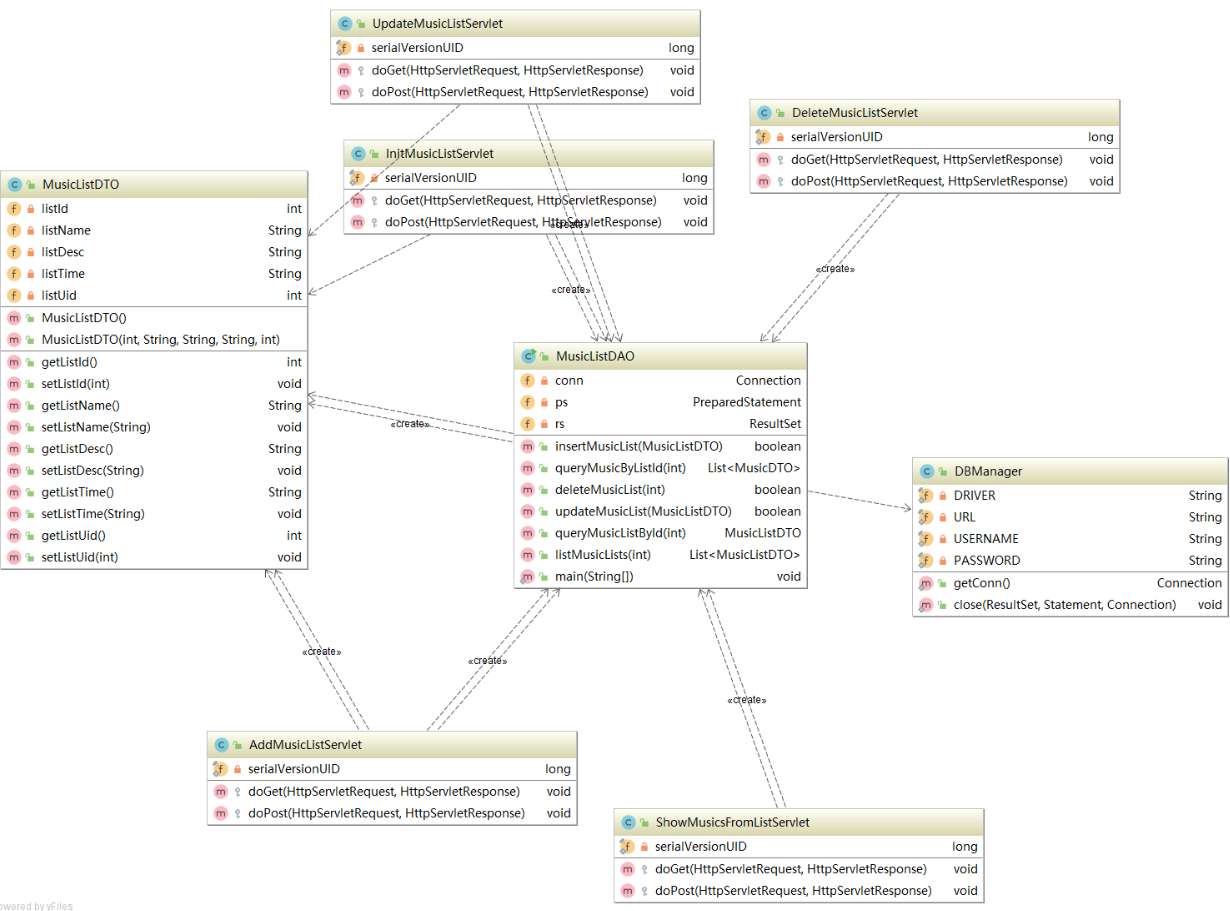


图3.3.3 歌单管理模块类图

DBManager：封装了连接数据库和关闭数据库的操作。

MusicListDTO：封装了歌单信息的类。

MusicListDAO：封装了操作歌单信息的类。

AddMusicListServlet：添加歌单信息的servlet。

InitMusicListServlet：通过用户ID查询并返回其对应的歌单的servlet。

DeleteMusicListServlet：删除歌单的servlet。

UpdateMusicListServlet：更新歌单的servlet。

ShowMusicsFromListServlet：通过用户ID和歌单ID查询并返回其对应的歌单的servlet。

### 收藏歌曲模块：

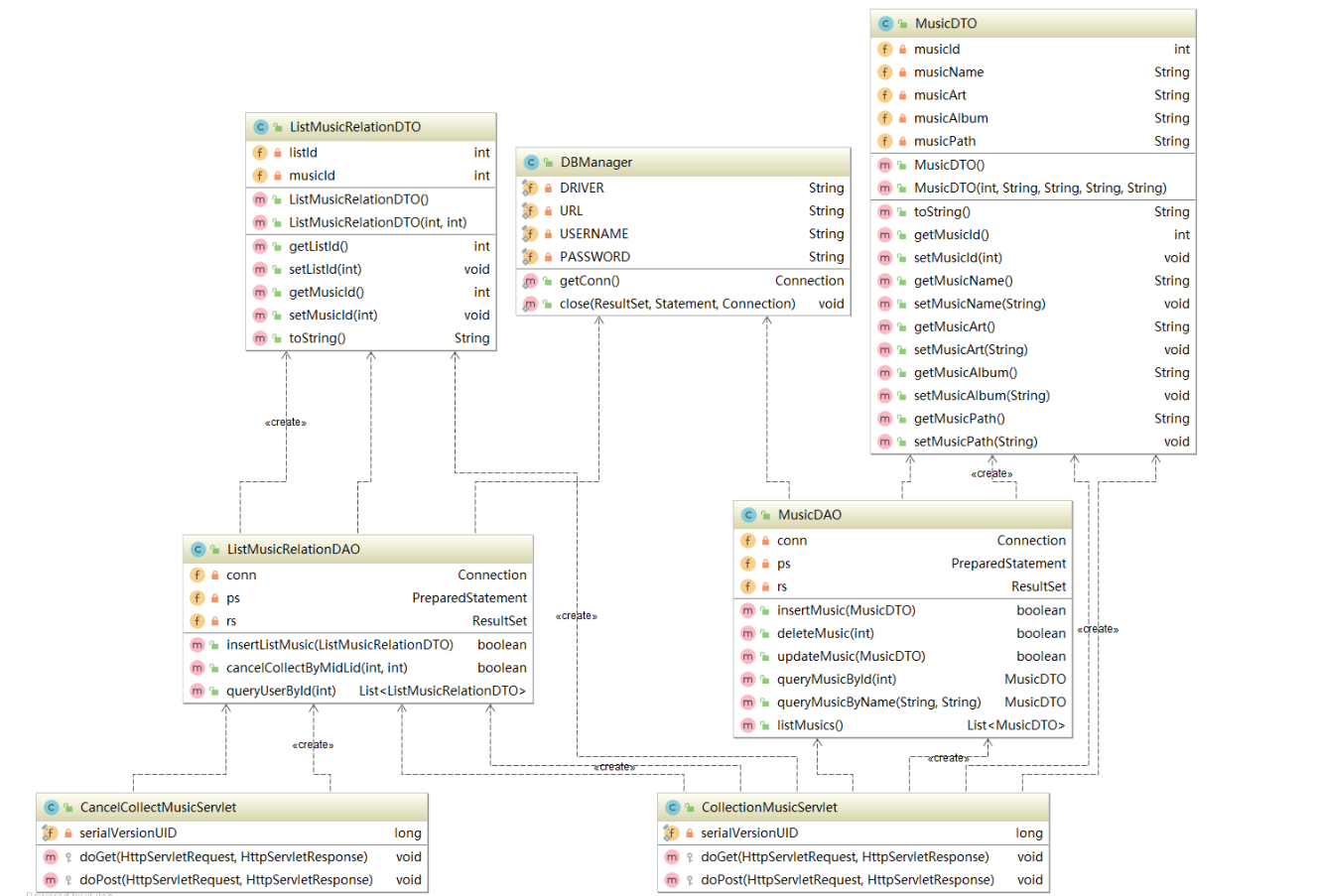


图3.3.4 收藏歌曲模块类图

DBManager：封装了连接数据库和关闭数据库的操作。

MusicDTO：封装了音乐信息的类。

MusicDAO：封装了操作音乐信息的类。

ListMusicRelationDTO：封装了音乐与歌单对应关系信息的类。

ListMusicRelationDAO：封装了操作音乐与歌单对应关系信息的类。

CollectionMusicServlet：收藏音乐进入指定歌单的servlet。

CancelCollectMusicServlet：从歌单取消某音乐收藏的servlet。

## 3.4设计数据管理子系统

因为使用java编写的并发量不是特别大的后台程序，所以使用了更为轻量级的mysql 8。其好处有如下几点：

1、普及性：MySQL在过去两年已经获得了25%的市场份额。相比其他的开源数据库和闭源数据库，越来越多的开发者将继续选择MySQL。MySQL在业界的流行所带来的另一个好处是，人们总可以很轻松地发现本行业的解决方案。

2、简单性：对于MySQL数据库，无论是在开发方面，还是支持方面，现在有大量强大的工具可以选择。每一个新手开发者可以轻松地使用MySQL数据库进行开发。甚至一个有经验的Windows管理者也可以轻松部署并开始学习它，不需投入一分钱来了解这个数据库。

3、低成本

MySQL数据库归MySQL AB公司所有，但是这个软件是开源的，有一个社区版可以免费下载。稍俱常识的新入门者都可以轻松实现在一个常见硬件上安装和配置MySQL。MySQL对硬件的较低要求是其最大的优势之一，不过需要注意的是：内存越多越好，因为所有的重要数据存储都在内存中完成。一个免费的数据库意味着，更多珍贵的资金可以用于其他业务的启动，诸如市场、广告或调研和开发等。

## 3.5 系统E-R图

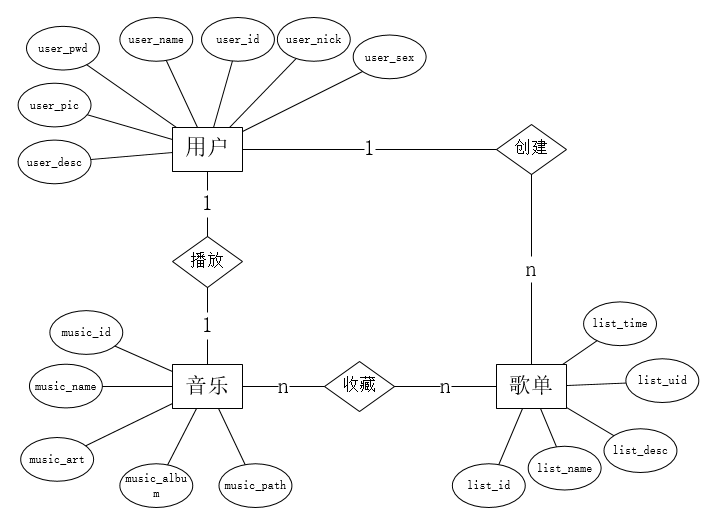


图3.5 系统E-R图

系统E-R图如图3.5所示。主要逻辑有：

1. 一个用户在某一时刻只能播放一首歌曲。
2. 一个用户可以创建n个歌单。
3. 一个音乐可以被n个歌单收藏，一个歌单也能收藏n首音乐。

## 3.6 数据库表

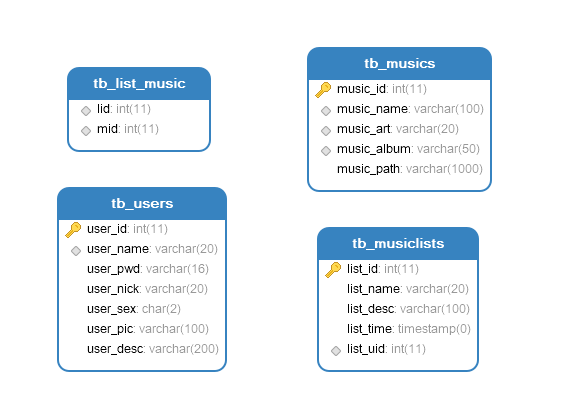


图3.6 数据库表

本系统该次设计了四张表：

tb\_users表记录用户信息，

tb\_musiclists表、

tb\_list\_music表、

tb\_musics表

## 3.7 数据库表之间的关系

### 关系1：

tb\_users表和tb\_musiclists表是一对多的关系。

其中tb\_musiclists中表的list\_uid字段必须遵照tb\_users表中的user\_id。

即一个用户可以有多个歌单，而一个歌单只能由一个用户。

其关系如图3.7.1所示。

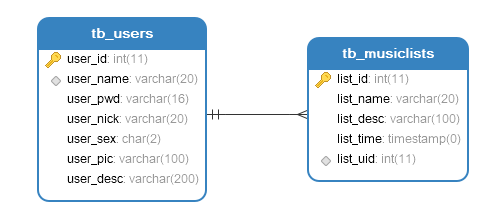


图3.7.1 tb\_users表和tb\_musiclists表的关系。

### 关系2：

tb\_musiclists表和tb\_musics表是多对多的关系，并且tb\_list\_music表记录了它们之间的对应关系。

tb\_list\_music表中lid参照tb\_musiclists中的list\_id字段，tb\_list\_music表中mid参照tb\_musics中的music\_id字段。并且lid与mid作为该表的联合主键。

即一个歌单可以收藏多首音乐，一个音乐也能被多个歌单收藏。

其关系如图3.7.2所示。

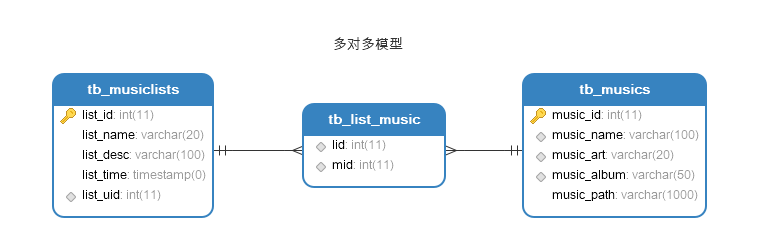


图3.7.2 tb\_musiclists表和tb\_musics表的关系

## 数据库表结构

### tb\_users表：

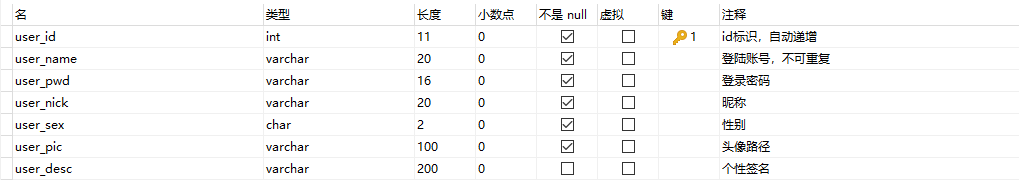


图3.8.1 tb\_users表详细设计

### tb\_musics表：



图3.8.2 tb\_musics表详细设计

### tb\_musiclists表：

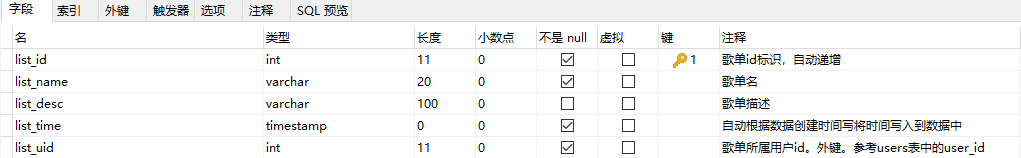


图3.8.3 tb\_musiclists表详细设计

### tb\_list\_music表：



图3.8.4 tb\_list\_music表详细设计

## 设计人机交互子系统

### 3.9.1登陆界面

登录界面引用了jquery.video库，将登录的背景设置为一个动态的循环播放的视频，极具科技感和新鲜感。

### 3.9.2注册界面

注册界面沿袭登陆界面的大体设计，并在注册信息上使用了jquery.validate库对于用户输入注册所需要的信息进行了初步的判断，避免录入冗余、错误的信息。

规则有：

用户名：必须输入，长度为6-20个字符。

密码：必须输入，长度为8-16个字符。

重复密码：必须输入，必须与密码一致。

用户昵称：必须输入，长度为20字符以内。

性别：必须选择。

头像：必须上传。

### 3.9.3.主界面

整体布局：使用一个界面避免用户进行跳转。采用bootstrap框架的栅栏式布局。

大致布局如图3.9.3.1所示。

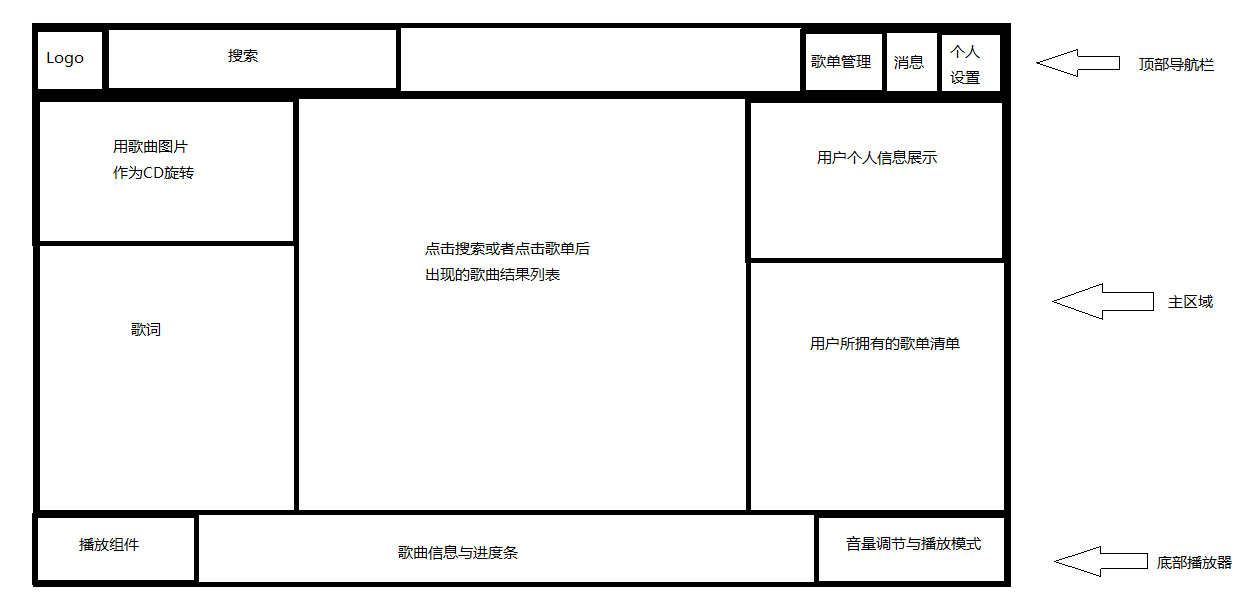


图3.9.3.1 主界面整体布局

顶部导航栏：

整体采用bootstrap的导航栏样式，如图3.9.3.2所示。

左侧依次为Logo区，在线搜索区和本地乐库区。当在在线搜索的输入框中输入内容，并点击搜索按钮后，会在主界面呈现出搜索内容。当点击本地乐库按钮后，会在主界面呈现出系统自带的音乐列表。

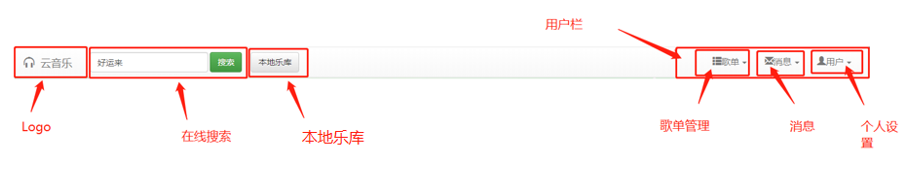


图3.9.3.2 顶部导航栏

右侧为用户栏，点击歌单按钮会弹出“添加歌单”、“管理歌单”功能按钮，如图3.9.3.3所示。点击“添加歌单”按钮会弹出模态框。点击管理歌单会在歌单栏出现编辑和删除按钮。

点击消息按钮会弹出“查看消息”功能按钮，如图3.9.3.4所示。

点击用户按钮会弹出“修改信息”和“退出”功能按钮，如图3.9.3.5所示。点击修改信息会弹出修改用户信息的模态框。详情在模态框中介绍。



图3.9.3.3 添加按钮下拉菜单



图3.9.3.4 消息按钮下拉菜单



图3.9.3.5 用户按钮下拉菜单

主区域：



图3.9.3.6 主区域



图3.9.3.7 歌单编辑模式

主区域静态区域和说明大体如3.9.3.6中所示。

主界面大致分为左、中、右三个区域。按照1：2：1的分配区域。

在点击导航栏中的“管理歌单按钮后”，歌单区域进入编辑模式，如图3.9.3.7所示。

点击主区域的播放按钮和收藏按钮，歌单区的编辑和删除，用户区的头像均会弹出模态框。详情在模态框中介绍。

唱片CD会在歌曲播放的时候旋转如图3.9.3.8所示



图3.9.3.8 CD唱片旋转

播放器区域：

播放器方面我们开发小组认为原生的AUDIO标签还有市面上找到的播放器插件不契合我们的样式，于是从5sing音乐网站的播放器部分借鉴了样式，并为静态的样式添加了拖动歌曲进度，实时显示已经播放时间，和调节音量大小的动态效果。其主要布局和功能如图3.9.3.9所示。

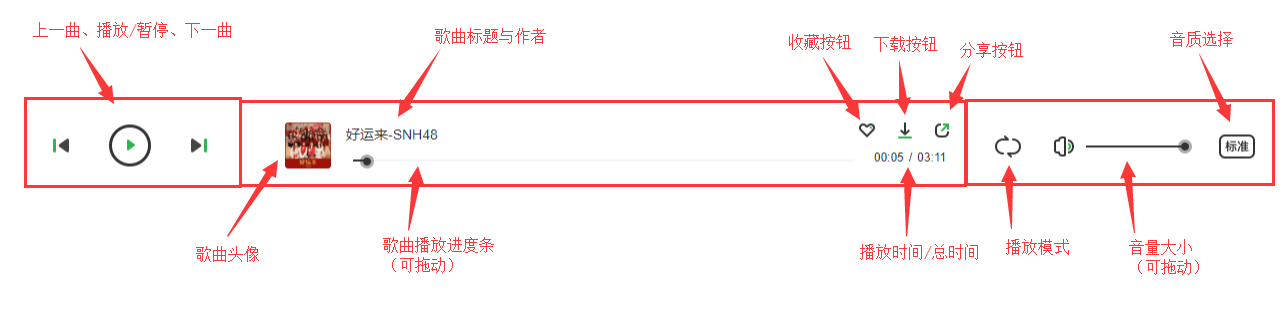


图3.9.3.9 播放器部分

\*模态框：

为了使界面不发生跳转，我们将主界面中所有的操作运用了bootstrap的模态框UI组件。在点击“歌单➡添加歌单”，“用户➡修改信息”，“歌曲➡收藏按钮”，“用户➡头像”，“歌单列表➡修改”，“歌单列表➡删除”时均会弹出相应的模态框组件。其效果如图3.9.3.10—图3.9.3.15所示。



图3.9.3.10 歌单➡添加歌单



图3.9.3.11 用户➡修改信息

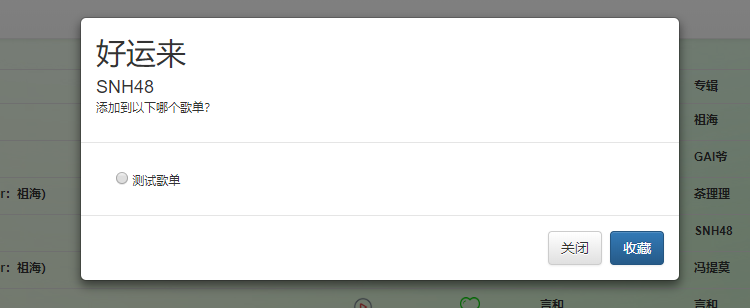


图3.9.3.12 歌曲➡收藏按钮

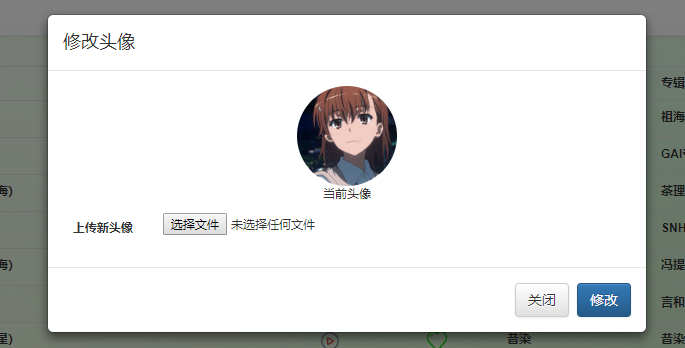


图3.9.3.13 用户➡头像



图3.9.3.14 歌单列表➡修改



图3.9.3.15歌单列表➡删除

整体效果如图3.9.3.16所示：



图3.9.3.16整体效果

# 4. 详细设计

## 4.1登陆注册模块流程图

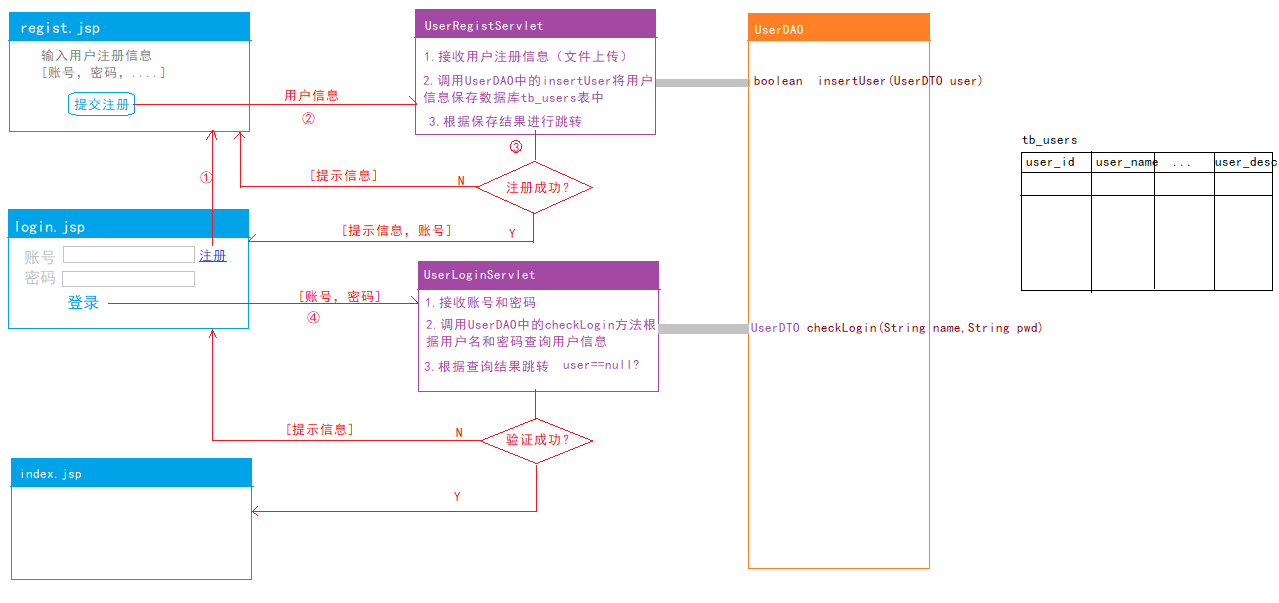


图 4.1登陆注册模块流程图

## 4.2在线搜索模块流程图

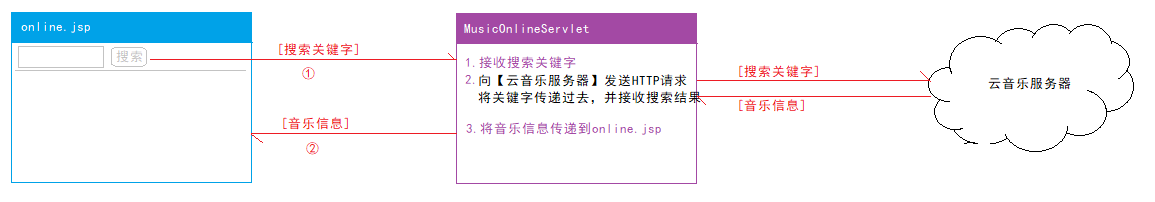


图 4.2在线搜索模块流程图

## 4.3本地乐库模块流程图

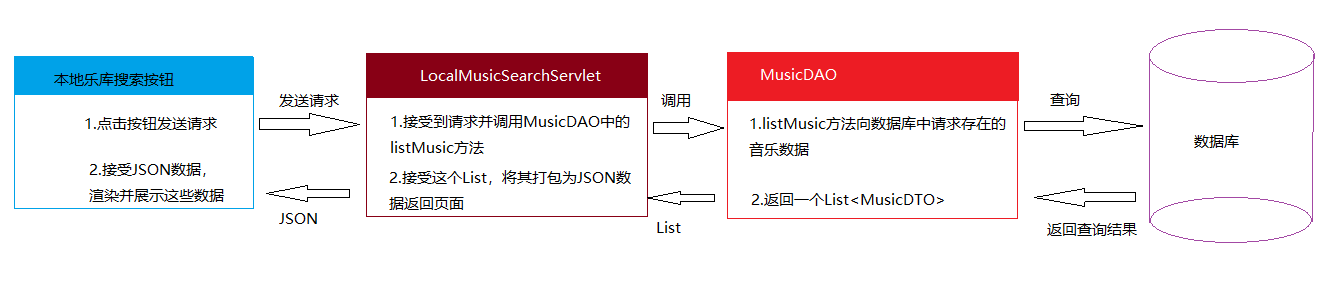


图 4.3本地乐库模块流程图

## 4.4歌单管理模块流程图

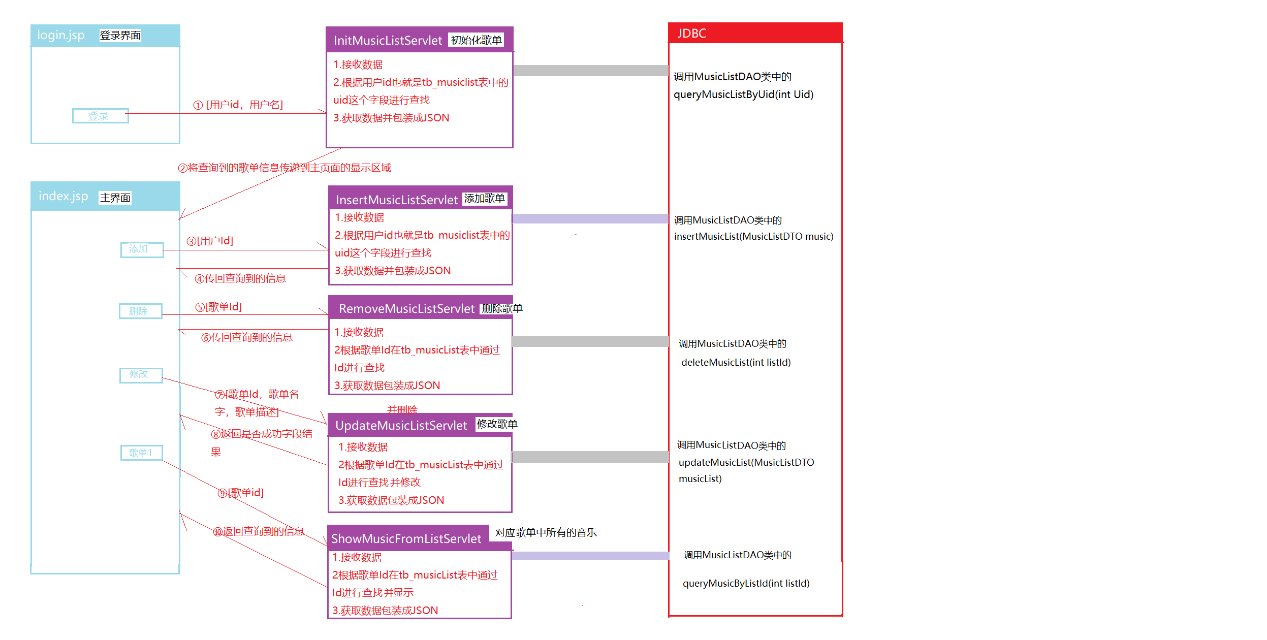
****

图 4.3歌单管理模块流程图

# 5. 系统实现

因篇幅限制，具体实现的源代码已经上传到github仓库，因此不在此赘述。

# 6. 总结

通过为期五天的课设，我收获颇多。最直观的感受就是在这次在线音乐网站的开发中能够系统性的运用从大一到现在所学到的知识。新学习到的Servlet的知识更是让我明白了为何之前写过的html只能叫做静态页面，还有一个网站的前端是如何与后台进行数据交互。同时这次也让我从平时的文件级别的代码编写上升到了工程级别的项目开发。在其中更好了理解了这个学期学过的关于软件开发的步骤。认识到了文档编写在软件项目开发中不可替代的重要性。我觉得收获最深的地方就是以前只觉得写代码才是最重要的部分，现在明白了项目管理、需求分析、文档编写才是项目开发中更重要的部分。让我把眼界从代码拓宽到了更高的层次。同时还深刻认识到自己在项目开发领域还存在着诸多的不足，明白了以后要学习的地方还有许多。

当然，在项目开发期间也遇到过许多问题：

1. 老师教授的是用后台修改动态数据，而我们想将前端和后台完全分离，这就存在了一个如何将后台数据打包成JSON发送到前端的问题。

**解决方案**：在老师的指导下使用了阿里巴巴公司的fastjson这个jar包来打包数据。同时在前端使用了axios来实现通信和局部页面刷新。

1. 因为一开始没有编写规范的文档，导致在开发途中想要增加功能的时候会发现不知道更改项目的哪个地方。他人接手编写的时候还要重新看一遍代码，效率低下。

**解决方案**：停下开发工作，转头编写规范的文档，统一了数据库和测试数据。并且集中到一起开发，使得遇到什么问题小组成员之间可以很快的交流并解决。

1. 每个人有自己的代码编写风格，而为了赶时间没有在文件中书写规范的注释。结果在互相测试小组成员单元时发现看不懂对方的代码，导致浪费极多时间。

**解决方案：**就算再赶时间也要书写规范的注释，写明类的用途，写明方法的用途以及接受参数和返回参数的类型、作用，这样在测试或者修改的时候会节省大量时间。

1. 在开发过程中想要对功能进行增加、删除、修改或者是想要复用一段代码时发现代码编写冗余没有规范。导致代码难以维护、复用程度低，重构成本极高。

**解决方案：**在将来的项目开发中一定要灵活运用设计模式，设计模式是前辈留给我们的解耦、复用代码的成功经验，虽然使用设计模式在编写的时候可能会造成困难，但是在维护、更新的时候会体现出设计模式的优越性。

# 7.参考文献：

1. BootStrap3文档 (<https://v3.bootcss.com/>)
2. Jquery文档(<https://api.jquery.com/>)
3. 5sing音乐网站(<http://5sing.kugou.com/index.html>)
4. Axios文档(<https://www.kancloud.cn/yunye/axios/234845>)

# 评分表单:

**《**Java程序设计**》课程设计评分表**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **学生姓名** | **张尹** | | | **班级** | | **软工1602** | | | **学号** | **1605110125** | |
| **评分项目** | **评分档次及分值** | | | | | | | | | | **评分** |
| **分值** | **A** | **B** | | **C** | | **D** | **E** | | |
| 出勤情况 | 20 | 20 | 18 | | 16 | | 14 | <12 | | |  |
| 分析与设计 | 30 | 30 | 27 | | 24 | | 21 | <18 | | |  |
| 代码质量 | 20 | 20 | 18 | | 16 | | 14 | <12 | | |  |
| 答辩水平 | 10 | 10 | 9 | | 8 | | 7 | <6 | | |  |
| 文档撰写 | 20 | 20 | 18 | | 16 | | 14 | <12 | | |  |
| **总 分** | | | | | | | | | | |  |

**请注意，以上评分表单独占用1页，一定放在报告最后一页， 否则没有分数.**