**数据库工程课程设计**

**说明书**

目录

[系统使用说明 2](#_Toc470811340)

[配置 2](#_Toc470811341)

[相关路径 2](#_Toc470811342)

[运行方法 3](#_Toc470811343)

[其他说明 3](#_Toc470811344)

[软件基本功能介绍 3](#_Toc470811345)

[模块(1) 登陆注册模块 3](#_Toc470811346)

[注册功能： 3](#_Toc470811347)

[登陆功能： 4](#_Toc470811348)

[模块(2) 首页推送模块 5](#_Toc470811349)

[新闻推送功能： 5](#_Toc470811350)

[音乐排名功能 5](#_Toc470811351)

[模块(3) 音乐模块 6](#_Toc470811352)

[音乐播放功能 6](#_Toc470811353)

[音乐收藏功能 6](#_Toc470811354)

[模块(4) 用户模块 7](#_Toc470811355)

[数据库设计 7](#_Toc470811356)

[功能模块使用的主要技术 10](#_Toc470811357)

[模块(1) 登陆注册模块 10](#_Toc470811358)

[模块(2) 首页推送模块 10](#_Toc470811359)

[模块(3) 音乐模块 10](#_Toc470811360)

[模块(4) 用户模块 10](#_Toc470811361)

[系统测试报告 10](#_Toc470811362)

[系统存的不足 11](#_Toc470811363)

# 系统使用说明

## 配置

开发环境：Eclipse Java EE IDE for Web Developers 4.3.2

开发语言：Java 1.8

MVC框架：Spring 4.3.3

web服务器：Tomcat

开发系统：OSX/LINUX

数据库： MySQL

## 相关路径

controller： CloudMusic/src/controller

model: CloudMusic/src/model

view: CloudMusic/WebContent/WEB-INF/view

## 运行方法

1、需要提前配置好mysql数据库，sql文件位于根目录 (DDL.sql DML.sql)

2、设置数据库参数 配置文件位于

apache-tomcat/webapps/CloudMusic/WEB-INF/classes/dbConfig.properties

3、终端启动服务器

cd apache-tomcat

./bin/catalina.sh run

运行成功后，浏览器打开 http://localhost:8080/CloudMusic 可访问

## 其他说明

1、若运行.sh文件 报permisison denied， 则在终端运行 chmod +x catalina.sh 修改权限

2、本项目github地址：<https://github.com/Hareric/Spring-MVC-CloudMusic>

3、本项目为对已经制作好前端界面功能的项目进行MVC重构 原项目github地址：<https://github.com/Zenaro/CloudMusic-for-normal>

# 软件基本功能介绍

## 模块(1) 登陆注册模块

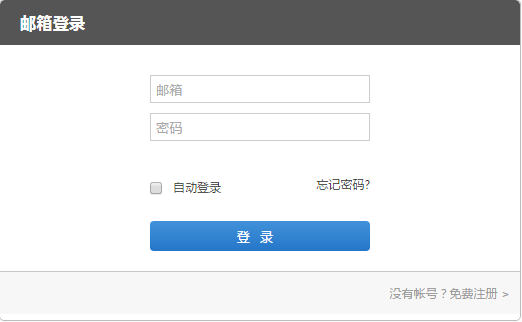
### 注册功能



功能：

1. 判断邮箱是否注册过
2. 判断输入的邮箱是否为有效邮箱
3. 确认密码与登陆密码无误
4. 注册成功，相关信息并随机生成不重复的用户id存入数据库
5. 注册成功后，自动登录并生成cookie并跳转回首页

### 登陆功能



功能：

1. 判断邮箱是否注册过
2. 从数据库查找相关信息，确认密码是否正确
3. 成功登陆后，生成cookie并跳转回首页

## 模块(2) 首页推送模块

### 新闻推送功能



功能：

1. 从数据库调用相关新闻信息后展示在首页

### 音乐排名功能



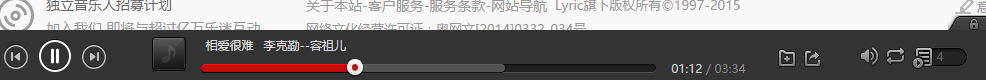
功能：

1. 连接数据库，按照类型从数据库中抽取最新的10首显示

## 模块(3) 音乐模块

### 音乐播放功能



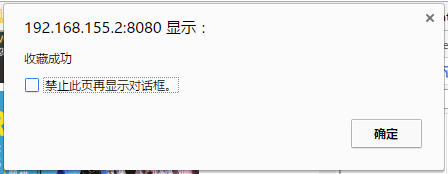


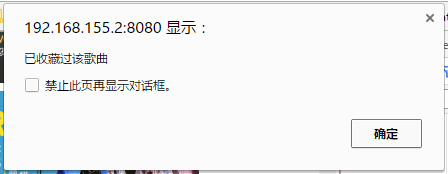
功能：

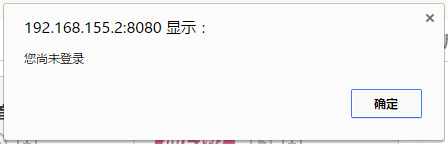
1. 音乐选择播放（第一个按钮），从数据库根据音乐id查询音乐相关信息，根据音乐的url下载播放
2. 音乐选择加入歌单（第二个按钮），播放器会接着播放。

### 音乐收藏功能









功能：

1. 音乐收藏（第三个按钮），从cookie中获得用户id，再根据音乐id存入数据库中。
2. 若没有登陆则无法收藏
3. 也无法重复收藏同样的音乐

## 模块(4) 用户模块



功能：

1. 获取用户信息并显示在个人页面。
2. 获取个人收藏的音乐
3. 在个人页面可使用小播放器播放音乐，功能同上
4. 若没有登陆，将无法成功进入该页面

# 数据库设计

**1. 用户表 （普通用户+管理员）（app\_user）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| email | 登录邮箱 | Varchar | 16 |  |  | 否 |
| pwd | 密码 | Char | 34 |  |  | 否 |
| id | 用户id | Int | 11 |  |  | 否 |
| regDate | 注册时间 | date | ～ |  |  | 否 |
| root | 身份 | Int | 11 |  |  | 否 |

**2. 新闻推送表 （app\_news）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| title | 推送标题 | Text | ～ |  |  | 否 |
| content | 推送内容 | Text | ～ |  |  | 否 |
| id | Id | Int | 11 |  |  | 否 |
| type | 类型 | Varchar | 16 |  |  | 否 |
| pubDate | 发表日期 | date |  |  |  | 否 |

**3. 用户信息表（app\_info）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| User\_id | 用户id | Int | 10 |  |  | 否 |
| Name | 昵称 | varchar | 16 |  |  | 否 |
| image | 头像路径 | varchar | 34 |  |  | 否 |

**4.歌曲表（app\_music）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| Name | 曲名 | text |  |  |  | 否 |
| src | 歌曲路径 | text |  |  |  | 否 |
| lyric | 歌词 | text |  |  |  | 否 |
| Music\_id | 歌曲id | int | 10 |  |  | 否 |
| Listener | 收听人数 | int | 10 |  |  | 否 |

**5.歌曲类型 表（app\_class）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| class\_id | 类型id | Int | 11 |  |  | 否 |
| Class\_name | 类型名 | Varchar | 80 |  |  | 否 |

**6.歌曲—类型 表（app\_musicRclass）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| music\_id | 歌曲id | Int | 11 |  |  | 否 |
| Class\_id | 类型\_id | Int | 11 |  |  | 否 |

**7. 歌手表（app\_singer）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| singer\_id | 歌手id | Int | 11 |  |  | 否 |
| Singer\_name | 歌手名字 | Varchar | 80 |  |  | 否 |
| country | 国籍 | Varchar | 80 |  |  | 否 |
| birthday | 生日 | Date | ～ |  |  | 否 |

**8. 歌手-歌曲表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| music\_id | 歌曲id | Int | 11 |  |  | 否 |
| singer\_id | 歌手\_id | Int | 11 |  |  | 否 |

**11.用户-歌曲收藏表(app\_collection)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 说明 | 类型 | 长度 | 精度 | 默认值 | 允许空值 |
| user\_id | 用户id | Int | 11 |  |  | 否 |
| Music\_id | 歌曲\_id | Int | 11 |  |  | 否 |
| colDate | 收藏的日期 | Date | ～ |  |  | 否 |
| Id | 字段id | Int | 10 |  |  | 否 |

# 功能模块使用的主要技术

## 模块(1) 登陆注册模块

1. 登陆注册功能都使用了cookie功能，用来保存用户的id，以支持用户的收藏功能。
2. 注册功能中，需要验证邮箱是否合法，通过正则匹配方法来匹配是否有有效邮箱。

正则表达式：[\\w+@(\\w+.)+[a-z]{2,3}](file:///\\\\w+@(\\\\w+.)+%5ba-z%5d%7b2,3%7d)

## 模块(2) 首页推送模块

1. 主要功能的实现都是通过SQL语句从数据库获取相关数据，并传给前端进行展示。
2. 从数据库获取的数据为ResultSet格式需要引入org.json.jar包并将ResultSet格式转换为json格式。

## 模块(3) 音乐模块

1. 数据库存取了音乐的url播放地址，通过音乐id获取。再通过网络获取相关资源后实现播放功能。
2. 收藏音乐前，会先检查是否有有效cookie从而确认是否登陆，之后再在数据库确认是否在用户-音乐收藏表中存在该记录，确认无误后，再将数据存入数据库。

## 模块(4) 用户模块

1. 检查本地是否有有效cookie，若无则跳转回。
2. 通过SQL语句从数据库获取相关数据，并传给前端进行展示。

# 系统测试报告

**问题一、环境配置复杂**

Spring框架是基于java开发的框架，相比于ThinkPHP框架，单从文件大小来看就相差几十M，配置起来也是非常麻烦。

Spring框架并没有像XAMPP一样的集成环境的软件可以选用，因此数据库和服务器都必须独立安装。Spring框架相比于HTML文件更加支持jsp文件，因此服务器的选择上选择了Tomcat进行开发。数据库需要本地安装并且配置好Mysql。

Eclipse有专门针对java web开发的版本，因此选用Eclipse Java EE IDE for Web Developers 4.3.2

**问题二、编码问题**

Spring框架对编码问题的解决还是比较方便，在前端的jsp页面上申明 pageEncoding=”UTF-8”，基本都可以解决页面中文乱码的问题。

从Mysql获取的数据常常都为ISO-8859-1编码格式，controller中RequestMapping可以事先申明编码类型 charset=utf-8即可解决编码问题。

**问题三、数据库查询复杂**

由于数据设计较为复杂，以获取音乐相关信息为例，音乐的链接、音乐名；音乐的作者；音乐的类型；分别在多个表中，在设计查询的SQL时需要用到natural join 关联多个表获取信息。

如获取音乐相关属性SQL语句：

SELECT name, music\_id, singer\_name, src FROM app\_singerRmusic NATURE JOIN app\_Singer NATURAL JOIN app\_Music LIMIT 20

**问题四、本地缓存影响调试**

由于本地缓存的功能，常常出现更改的后台相关文件，却无法看到更改后的效果，因此在调试的时候，时常需要清除本地缓存，并重启浏览器

**问题五、Spring框架中，Request的地址中不能获取带有’/’的参数**

由于链接通常是带有多个’/’标点，所以无法使用 /getSrc/{url} 的url参数获取音乐的链接，因此改用 /getSrc/\*\* 匹配所有该样式的访问，再通过request.getRequestURL()获取完整的链接，然后经过简单的字符串处理后获取所需要的参数url.

## 系统存的不足

1. 功能未完善，由于时间原因，原项目的还具有增加好友、音乐评论等功能尚未实现。
2. 用户密码存入数据库并没有进行加密，而是直接存入数据库。