毕 业 论 文

(小初号 楷体 居中)

(四号黑体)

|  |  |
| --- | --- |
| **题 目：** | springboot校园心声墙小程序 |
|  |  |
| **学 院：** |  |
| **专 业：** |  |
| **姓 名：** |  |
| **学 号：** |  |
| **指导教师：** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 年 |  | 月 |  | 日 |

# 摘 要

校园心声墙小程序的应用范围很广，很多学校都会建立一套具有自己学校风格的心声墙小程序，来方便学校学生之间的日常联系和交流心声的建立对学生之间的信息的传播，知识的交流提供了一个很好的平台，增加了学生课余生活的情趣。本校园心声墙是基于小程序开发的一套系统。校园心声墙小程序的主要功能就是方便同学之间的联系，促进同窗好友的感情，校园心声墙小程序系统功能齐全，含有留言，搜索，上传和后台管理等功能。

校园心声墙小程序的设计主要是对系统所要实现的功能进行详细考虑，确定所要实现的功能后进行界面的设计，在这中间还要考虑如何可以更好的将功能及页面进行很好的结合，方便用户可以很容易明了的找到自己所需要的信息，还有系统平台后期的可操作性，通过对信息内容的详细了解进行技术的开发。

校园心声墙小程序的开发利用现有的成熟技术参考，以源代码为模板，分析功能调整与校园心声墙小程序管理的实际需求相结合，讨论了基于校园心声墙小程序管理的使用。

**关键词：**校园心声墙小程序；微信小程序；MySQL

**Abstract**

The application scope of the campus voice wall widget is very wide. Many schools will establish a set of voice wall widget with their own school style to facilitate the daily contact and communication between school students. The establishment of voice of mind will spread information among students. The exchange of knowledge will provide a good platform and increase the interest of students' extracurricular life. The campus voice wall is a system developed based on small programs. The main function of the campus voice wall applet is to facilitate the contact between students and promote the feelings of classmates and friends. The campus voice wall applet system has complete functions, including message, search, upload and background management functions.

The design of the campus voice wall applet is mainly to consider the functions to be realized by the system in detail, and then to design the interface after determining the functions to be realized. In this process, we should also consider how to better combine the functions and pages, so that users can easily find the information they need, as well as the operability of the system platform in the later stage, Develop technology through detailed understanding of information content.

The development of the campus voice wall applet uses the existing mature technology reference, takes the source code as the template, analyzes the function adjustment and the actual needs of the campus voice wall applet management, and discusses the use of the campus voice wall applet management.

**Keywords：**Campus voice wall widget; WeChat applet; MySQL

# 目 录

[摘 要 1](#_Toc98315467)

[目 录 1](#_Toc98315468)

[1 引言 1](#_Toc98315469)

[1.1 课题研究背景 1](#_Toc98315470)

[1.2 研究现状 1](#_Toc98315471)

[1.3系统的特点 1](#_Toc98315472)

[1.4论文结构与章节安排 2](#_Toc98315473)

[2 系统需求分析 3](#_Toc98315474)

[2.1可行性分析 3](#_Toc98315475)

[2.1.1 技术可行性分析 3](#_Toc98315476)

[2.1.2 操作可行性分析 3](#_Toc98315477)

[2.1.3 经济可行性分析 3](#_Toc98315478)

[2.2功能需求分析 3](#_Toc98315479)

[3系统总体设计 6](#_Toc98315481)

[3.1 系统开发架构的选择 6](#_Toc98315482)

[3.1系统功能模块设计 6](#_Toc98315483)

[3.1.1 用户端页面设计 7](#_Toc98315484)

[3.1.2 用户模块设计 7](#_Toc98315485)

[3.1.3 评论管理模块设计 8](#_Toc98315486)

[3.1.4 热门话题模块设计 8](#_Toc98315487)

[3.2数据库设计 9](#_Toc98315488)

[3.2.1数据库概要设计 9](#_Toc98315489)

[3.2.2数据库逻辑设计 9](#_Toc98315490)

[4系统详细实现 14](#_Toc98315491)

[4.1用户端功能实现 14](#_Toc98315492)

[4.1.1网站主页功能实现 14](#_Toc98315493)

[4.1.2用户注册登录功能实现 14](#_Toc98315494)

[4.2.3 登录功能实现 15](#_Toc98315495)

[4.2.4心声墙功能实现 16](#_Toc98315496)

[4.2.5热门话题功能实现 17](#_Toc98315497)

[4.2.6我的功能实现 17](#_Toc98315498)

[4.2.7我的功能实现 18](#_Toc98315499)

[4.2管理端功能实现 19](#_Toc98315500)

[4.2.1管理端管理员登录功能实现 19](#_Toc98315502)

[4.2.2人员管理功能实现 20](#_Toc98315503)

[4.2.3网站管理功能实现 20](#_Toc98315504)

[4.2.4播图管理界面 21](#_Toc98315505)

[4.2.5公告消息界面 21](#_Toc98315506)

[4.2.6内容管理功能实现 22](#_Toc98315507)

[5系统测试 23](#_Toc98315509)

[5.1软件测试的重要性 23](#_Toc98315510)

[5.2测试实例的研究与选择 23](#_Toc98315511)

[5.3测试环境与测试条件 23](#_Toc98315512)

[5.4 测试总结 24](#_Toc98315513)

[总结 26](#_Toc98315514)

[参考文献 27](#_Toc98315515)

[致 谢 28](#_Toc98315516)

# 1 引言

## 1.1 课题研究背景

随着网络时代的到来，互联网的优势和普及时刻影响并改变着人们的生活方式。在信息技术迅速发展的今天，计算机技术已经遍及全球，使社会发生了巨大的变革。

为了不受时间和地点的限制，智能手机用户可以通过移动网络访问网站和处理各种业务和互联网，这是一个有效的将应用系统的功能扩展到手机终端的方法。现今各种智能手机层出不穷，各类基于手机平台的软件应运而生，其中，在众多交流软件中，微信备受人们青睐。近年来，微信发展规模越来越大，越来越多的人开始使用微信，目前随着智能手机系统的普及，人人手机上基本都有了微信。

所以，微信推出小程序广告支持公众号关注，而这就意味着小程序跟公众号之间的通道被彻底打通了。本论文校园心声墙小程序主要牵扯到的程序，数据库与计算机技术等。覆盖知识面大，可以大大的提高系统人员工作效率。

## 1.2 研究现状

随着社会的发展和科学技术的进步，互联网技术越来越受欢迎。网络传播的生活方式逐渐受到广大人民群众的喜爱。越来越多的互联网爱好者开始在互联网上满足他们的基本需求，同时逐渐进入各个用户的生活起居。互联网具有许多优点，例如便利性，速度，高效率和低成本。因此，类似于校园心声墙小程序，满足用户工作繁忙的需求，不仅是方便用户随时查看信息的途径，而且还能提高管理效率。

本文首先以校园心声墙小程序过程的基本问题作为研究对象。在开发系统之前，我们对现有状况进行了详细的调查和分析。最后，我们利用计算机技术开发了一套完整合适的微信小程序的校园心声墙。该系统的实现主要优势是：该系统主要采用计算机技术开发，它方便快捷；系统可以通过管理员界面查看用户的所有信息管理。

校园心声墙小程序软件是一款方便、快捷、实用的信息服务查询软件。随着智能网络在全球市场的不断普及以及各种智能平台的使用，系统的开发与人们的日常需求相关，作为中国主流智能的技术开发系统，自然需要这样的软件来满足更多用户的需求和体验。

## 1.3系统的特点

（1）校园心声墙小程序中的web管理端管理中的后端不再使用古老的jsp+javabean+servlet技术，而是使用当前主流的springboot框架，它减少java配置代码，简化编程代码，目前springboot框架也是很多企业选择的框架之一。

（2）校园心声墙小程序中的web管理端管理中的前端使用的是bootstrap框架，它配合ajax和jquery可以美化页面设计。

（3）校园心声墙小程序中数据库用的mysql5.7，它执行效率高。

## 1.4论文结构与章节安排

本文共分为六章，章节内容安排如下：

第一章：引言。第一章主要介绍了课题研究的背景意义，国内目前相关研究现状以及技术和本文的研究内容与主要工作。

第二章：系统需求分析。第二章主要从系统的可行性、功能等方面进行需求分析。

第三章：系统总体设计。第三章主要对系统功能模块、数据库进行功能设计。

第四章：系统详细设计。第四章主要介绍了系统各个用户的功能、系统界面的实现。

第五章：系统测试。第五章主要对系统的部分界面进行测试并对主要功能进行测试

# 2 系统需求分析

## 2.1可行性分析

通过以下的网站总体需求分析，进而分析是否拥有达成需求的技术、资金和开发人员等条件。那么就从以下几点进行简单的说明：

### 2.1.1 技术可行性分析

校园心声墙小程序存储所使用的mysql数据库以及开发中所使用的IDEA、Tomcat这些开发工具的使用，能够给我们的编写工作带来许多的便利。系统使用springboot框架进行开发，使系统的可扩展性和维护性更佳，减少java配置代码，简化编程代码，目前springboot框架也是很多企业选择的框架之一。

### 2.1.2 操作可行性分析

在日常生活中，随着小程序的快速推广和使用，越来越多人掌握小程序的使用方法，基于微信小程序校园心声墙小程序在这种条件背景下是很容易被人们所接受和熟悉的，所以在操作上没任何问题。

### 2.1.3 经济可行性分析

在开发校园心声墙小程序中所使用的开发软件像IDEA开发工具、Tomcat8.0服务器、MySQL5.7数据库、Photoshop图片处理软件等，这些都是开源免费的，并且校园心声墙小程序是自己设计并编码实现的，数据库是使用流行mysql进行数据的存储，开源的mysql等技术的使用，减少系统开发费用。

## 2.2功能需求分析

校园心声墙小程序中的web管理端管理端采用了IntelliJ IDEA 2019.3.4 开发工具，配合了java开发语言中springboot开发框架以及tomcat8.0服务器、jdk1.8，用户端采用了微信开发者工具稳定版1.05.2111300，页面使用wxss和wxml进行布局，用户端和web管理端管理端采用json接口通信。

校园心声墙小程序中的管理端管理员角色主要负责了如下功能操作。

（1）网站管理功能需求：网站管理包括了轮播图、公告信息两部分，管理员可以对这两部分进行添加，修改，删除和查询等。

（2）人员管理功能需求：人员管理部分包括了管理员，学生用户两个角色，管理员可以对这两个角色进行添加，修改，删除以及查询等。

（3）内容管理功能需求：内容管理主要实现了心声墙、心声分类、热门话题、话题分类的功能，管理员可以进行维护管理操作。



图2-1管理员角色用例图

校园心声墙小程序中的用户端用户角色主负责了如下功能操作。

（1）用户查看系统信息功能需求：用户可以通过系统首页查看校园心声墙小程序信息，包括首页，心声墙、热门话题、我的等模块。

（2）用户登录与注册功能需求：用户可以通过注册获取帐户。

（3）首页：可查看管理员发布所有公告消息；

（4）心声墙：可搜索查看心声墙所有帖子，对喜欢的帖子进行收藏、点赞评，也可发布帖子；

（5）热门话题：用户登录用户端以后，是可以对资讯信息进行查看、点赞、收藏、评论等操作；

（6）我的：用户可对个人信息、头像、密码、昵称进行修改等操作。



图2-2 用户端用户角色用例图

# 3系统总体设计

## 3.1 系统开发架构的选择

小程序可以通过浏览器来规避很多问题，依靠浏览器让项目的开发变得简单，不再担心项目的版本跟新带来的数据传输等一系列的问题。值得注意的是B/S架构的项目，浏览器与服务器的数据的交互是通过http协议进行的，同样，服务器与数据库的数据交互也是通过http协议，统一的格式让信息的交互更加的简便与快捷。下图是具体的b/s架构图：



图3-1 系统B/S结构

## 3.1系统功能模块设计

校园心声墙小程序的整体架构确定以后，再来看校园心声墙小程序的主要功能模块图。整体的功能模块包括用户端和管理端，用户端只要实现用户注册登录，主要的页面，包括首页，心声墙、热门话题等页面。管理端的模块主要实现了管理员用户，有网站管理、人员管理、内容管理等。确定了模块以后，再设计各个模块的功能特点，方便抽取出各个模块的公共部分来。

图3.1校园心声墙小程序功能模块图

### 3.1.1 用户端页面设计

本校园心声墙小程序的用户端页面所有的页面设计主要是上中下的结构，这样的好处是让所有的页面更加的整洁。主题的页面颜色采用淡色为主，为突出页面的美观。主页面上分为导航栏，导航栏的每个导航项目设计为一张jsp页面，除此之外，还有就是用户中心页面，管理端的页面了。管理端页面已简单的上下结构为主。

### 3.1.2 用户模块设计

（1）用户模块结构图

本网站的用户包括管理员，注册用户两模块。两种用户模块的功能基本是相同的，用户模块比管理员多了一个注册功能，所以以用户模块的结构图为例进行分析，如下图：



图3.2用户模块结构图

（2）各个结构的具体业务逻辑

a.查询用户信息：设计最基础的查询功能，主要就是根据用户id查询用户信息。该功能主要体现在管理员用户信息查询和用户自己查询信息的时候。

b.修改用户信息：用户可以根据情况，对自己的基本信息进行修改，包括名字修改、密码修改、头像修改等操作。

c.增加用户：用户可以通过注册来获得账号

d.删除用户信息：管理员没有权利进行删除信息的删除，最多把用户加入到黑名单中，禁止用户进行登录操作。

### 3.1.3 评论管理模块设计

（1）评论模块结构图

校园心声墙小程序是一个交流性质的公开网站，用户和用户之间可以对网站上的内容，比如心声墙、热门话题等进行点评，增加用户之间的互动性。但是同时也为了更好的规范评论的内容，给予管理员删除不合适的言论的功能，所以需要专门设计一个评论管理模块，具体的结构图如下：



图3-4评论模块结构图

（2）各个结构的具体业务逻辑

a.查询评论：设计最基础的查询功能，主要就是根据用户id查询该用户下的所有评论信息。

b.发布评论：用户可以根据情况，发布自己评论信息到网站上。

c.删除评论：管理员没有权限修改用户的评论信息，但是可以删除用户的整条评论。

### 3.1.4热门话题模块设计

校园心声墙小程序重要的一个功能就是热门话题，其模块功能结构，具体的结构图如下：



图3.4订单模块结构图

（2）各个结构的具体业务逻辑

a.添加话题：管理员可以对话题进行添加。

b.修改话题：管理员可以对话题进行修改。

c.删除话题：管理员可以对话题进行删除。

d.热门话题分类：管理员可以对热门话题进行分类管理。

## 3.2数据库设计

### 3.2.1数据库概要设计

校园心声墙小程序根据上面的E-R实体图，就再找到实体与实体之间的关系来，制作总的E-R图，实体之间的关系一共有4种，也就是一对一，多对一，一对多和多对多，总的E-R图是设计数据库表格的依据，如下图是网站的总E-R图。



图3.5系统总体E-R图

### 3.2.2数据库逻辑设计

根据上面的E-R图，继而设计数据库表格，注意的是数据库表格遵从三范式标准，这样以后操作也会更加简单一些。

系统的各个数据库表如下。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **access\_token** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| token\_id | int | 11 | 是 | 是 | 临时访问牌ID |
| token | varchar | 64 | 否 | 否 | 临时访问牌 |
| info | text | 0 | 否 | 否 |  |
| maxage | int | 2 | 是 | 否 | 最大寿命 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
| user\_id | int | 11 | 是 | 是 | 用户编号 |
|  |  |  |  |  |  |
| **article** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| article\_id | mediumint | 8 | 是 | 是 | 文章id |
| title | varchar | 125 | 是 | 否 | 标题 |
| type | varchar | 64 | 是 | 否 | 文章分类 |
| hits | int | 10 | 是 | 是 | 点击数 |
| praise\_len | int | 11 | 是 | 否 | 点赞数 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
| source | varchar | 255 | 否 | 否 | 来源 |
| url | varchar | 255 | 否 | 否 | 来源地址 |
| tag | varchar | 255 | 否 | 否 | 标签 |
| content | longtext | 0 | 否 | 否 | 正文 |
| img | varchar | 255 | 否 | 否 | 封面图 |
| description | text | 0 | 否 | 否 | 文章描述 |
|  |  |  |  |  |  |
| **article\_type** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| type\_id | smallint | 5 | 是 | 是 | 分类ID |
| display | smallint | 4 | 是 | 是 | 显示顺序 |
| name | varchar | 16 | 是 | 否 | 分类名称 |
| father\_id | smallint | 5 | 是 | 是 | 上级分类ID |
| description | varchar | 255 | 否 | 否 | 描述 |
| icon | text | 0 | 否 | 否 | 分类图标 |
| url | varchar | 255 | 否 | 否 | 外链地址 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
|  |  |  |  |  |  |
| **auth** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| auth\_id | int | 11 | 是 | 是 | 授权ID |
| user\_group | varchar | 64 | 否 | 否 | 用户组 |
| mod\_name | varchar | 64 | 否 | 否 | 模块名 |
| table\_name | varchar | 64 | 否 | 否 | 表名 |
| page\_title | varchar | 255 | 否 | 否 | 页面标题 |
| path | varchar | 255 | 否 | 否 | 路由路径 |
| position | varchar | 32 | 否 | 否 | 位置 |
| mode | varchar | 32 | 是 | 否 | 跳转方式 |
| add | tinyint | 1 | 是 | 是 | 是否可增加 |
| del | tinyint | 1 | 是 | 是 | 是否可删除 |
| set | tinyint | 1 | 是 | 是 | 是否可修改 |
| get | tinyint | 1 | 是 | 是 | 是否可查看 |
| field\_add | varchar | 500 | 否 | 否 | 添加字段 |
| field\_set | varchar | 500 | 否 | 否 | 修改字段 |
| field\_get | varchar | 500 | 否 | 否 | 查询字段 |
| table\_nav\_name | varchar | 500 | 否 | 否 | 跨表导航名称 |
| table\_nav | varchar | 500 | 否 | 否 | 跨表导航 |
| option | text | 0 | 否 | 否 | 配置 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
|  |  |  |  |  |  |
| **collect** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| collect\_id | int | 10 | 是 | 是 | 收藏ID |
| user\_id | int | 10 | 是 | 是 | 收藏人ID |
| source\_table | varchar | 255 | 否 | 否 | 来源表 |
| source\_field | varchar | 255 | 否 | 否 | 来源字段 |
| source\_id | int | 10 | 是 | 是 | 来源ID |
| title | varchar | 255 | 否 | 否 | 标题 |
| img | varchar | 255 | 否 | 否 | 封面 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
|  |  |  |  |  |  |
| **comment** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| comment\_id | int | 11 | 是 | 是 | 评论ID |
| user\_id | int | 11 | 是 | 是 | 评论人ID |
| reply\_to\_id | int | 11 | 是 | 是 | 回复评论ID |
| content | longtext | 0 | 否 | 否 | 内容 |
| nickname | varchar | 255 | 否 | 否 | 昵称 |
| avatar | varchar | 255 | 否 | 否 | 头像地址 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
| source\_table | varchar | 255 | 否 | 否 | 来源表 |
| source\_field | varchar | 255 | 否 | 否 | 来源字段 |
| source\_id | int | 10 | 是 | 是 | 来源ID |
|  |  |  |  |  |  |
| **forum** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| forum\_id | mediumint | 8 | 是 | 是 | 论坛id |
| display | smallint | 5 | 是 | 是 | 排序 |
| user\_id | mediumint | 8 | 是 | 是 | 用户ID |
| nickname | varchar | 16 | 否 | 否 | 昵称 |
| praise\_len | int | 10 | 否 | 否 | 点赞数 |
| hits | int | 10 | 是 | 是 | 访问数 |
| title | varchar | 125 | 是 | 否 | 标题 |
| keywords | varchar | 125 | 否 | 否 | 关键词 |
| description | varchar | 255 | 否 | 否 | 描述 |
| url | varchar | 255 | 否 | 否 | 来源地址 |
| tag | varchar | 255 | 否 | 否 | 标签 |
| img | text | 0 | 否 | 否 | 封面图 |
| content | longtext | 0 | 否 | 否 | 正文 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
| avatar | varchar | 255 | 否 | 否 | 发帖人头像 |
| type | varchar | 64 | 是 | 否 | 论坛分类 |
|  |  |  |  |  |  |
| **forum\_type** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| type\_id | smallint | 5 | 是 | 是 | 分类ID |
| name | varchar | 16 | 是 | 否 | 分类名称 |
| description | varchar | 255 | 否 | 否 | 描述 |
| url | varchar | 255 | 否 | 否 | 外链地址 |
| father\_id | smallint | 5 | 是 | 是 | 上级分类ID |
| icon | varchar | 255 | 否 | 否 | 分类图标 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
|  |  |  |  |  |  |
| **hits** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| hits\_id | int | 10 | 是 | 是 | 点赞ID |
| user\_id | int | 11 | 是 | 是 | 点赞人 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
| source\_table | varchar | 255 | 否 | 否 | 来源表 |
| source\_field | varchar | 255 | 否 | 否 | 来源字段 |
| source\_id | int | 10 | 是 | 是 | 来源ID |
|  |  |  |  |  |  |
| **information\_center** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| information\_center\_id | int | 11 | 是 | 否 | 信息中心ID |
| send\_user | int | 11 | 否 | 否 | 发送用户 |
| receiving\_user | int | 11 | 否 | 否 | 接收用户 |
| message\_header | varchar | 64 | 否 | 否 | 信息标题 |
| send\_date | date | 0 | 否 | 否 | 发送日期 |
| information\_content | text | 0 | 否 | 否 | 信息内容 |
| recommend | int | 11 | 是 | 否 | 智能推荐 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
|  |  |  |  |  |  |
| **notice** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| notice\_id | mediumint | 8 | 是 | 是 | 公告id |
| title | varchar | 125 | 是 | 否 | 标题 |
| content | longtext | 0 | 否 | 否 | 正文 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
|  |  |  |  |  |  |
| **praise** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| praise\_id | int | 10 | 是 | 是 | 点赞ID |
| user\_id | int | 11 | 是 | 是 | 点赞人 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
| source\_table | varchar | 255 | 否 | 否 | 来源表 |
| source\_field | varchar | 255 | 否 | 否 | 来源字段 |
| source\_id | int | 10 | 是 | 是 | 来源ID |
| status | tinyint | 1 | 是 | 否 | 点赞状态 |
|  |  |  |  |  |  |
| **slides** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| slides\_id | int | 10 | 是 | 是 | 轮播图ID |
| title | varchar | 64 | 否 | 否 | 标题 |
| content | varchar | 255 | 否 | 否 | 内容 |
| url | varchar | 255 | 否 | 否 | 链接 |
| img | varchar | 255 | 否 | 否 | 轮播图 |
| hits | int | 10 | 是 | 是 | 点击量 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
|  |  |  |  |  |  |
| **student\_users** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| student\_users\_id | int | 11 | 是 | 否 | 学生用户ID |
| student\_id | varchar | 64 | 否 | 否 | 学生学号 |
| student\_name | varchar | 64 | 否 | 否 | 学生姓名 |
| student\_gender | varchar | 64 | 否 | 否 | 学生性别 |
| student\_department | varchar | 64 | 否 | 否 | 学生院系 |
| major | varchar | 64 | 否 | 否 | 学生专业 |
| examine\_state | varchar | 16 | 是 | 否 | 审核状态 |
| recommend | int | 11 | 是 | 否 | 智能推荐 |
| user\_id | int | 11 | 是 | 否 | 用户ID |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |
|  |  |  |  |  |  |
| **upload** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| upload\_id | int | 11 | 是 | 否 | 上传ID |
| name | varchar | 64 | 否 | 否 | 文件名 |
| path | varchar | 255 | 否 | 否 | 访问路径 |
| file | varchar | 255 | 否 | 否 | 文件路径 |
| display | varchar | 255 | 否 | 否 | 显示顺序 |
| father\_id | int | 11 | 否 | 否 | 父级ID |
| dir | varchar | 255 | 否 | 否 | 文件夹 |
| type | varchar | 32 | 否 | 否 | 文件类型 |
|  |  |  |  |  |  |
| **user** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| user\_id | mediumint | 8 | 是 | 是 | 用户ID |
| state | smallint | 1 | 是 | 是 | 账户状态 |
| user\_group | varchar | 32 | 否 | 否 | 所在用户组 |
| login\_time | timestamp | 0 | 是 | 否 | 上次登录时间 |
| phone | varchar | 11 | 否 | 否 | 手机号码 |
| phone\_state | smallint | 1 | 是 | 是 | 手机认证 |
| username | varchar | 16 | 是 | 否 | 用户名 |
| nickname | varchar | 16 | 否 | 否 | 昵称 |
| password | varchar | 64 | 是 | 否 | 密码 |
| email | varchar | 64 | 否 | 否 | 邮箱 |
| email\_state | smallint | 1 | 是 | 是 | 邮箱认证 |
| avatar | varchar | 255 | 否 | 否 | 头像地址 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
|  |  |  |  |  |  |
| **user\_group** |  |  |  |  |  |
| **字段名称** | **类型** | **长度** | **不是null** | **主键** | **字段说明** |
| group\_id | mediumint | 8 | 是 | 是 | 用户组ID |
| display | smallint | 4 | 是 | 是 | 显示顺序 |
| name | varchar | 16 | 是 | 否 | 名称 |
| description | varchar | 255 | 否 | 否 | 描述 |
| source\_table | varchar | 255 | 否 | 否 | 来源表 |
| source\_field | varchar | 255 | 否 | 否 | 来源字段 |
| source\_id | int | 10 | 是 | 是 | 来源ID |
| register | smallint | 1 | 否 | 是 | 注册位置 |
| create\_time | timestamp | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |

# 4系统详细实现

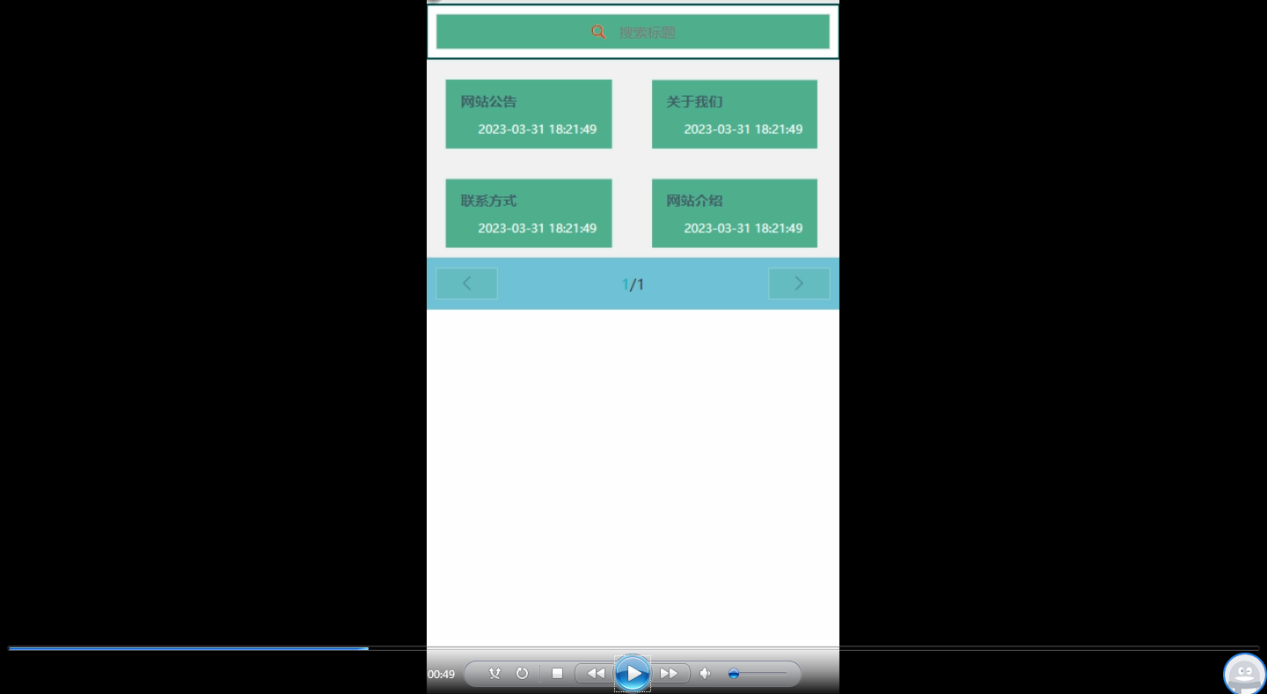
校园心声墙小程序的详细设计与实现主要是根据前面的校园心声墙小程序的需求分析和校园心声墙小程序的总体设计来设计页面并实现业务逻辑。主要从校园心声墙小程序界面实现、业务逻辑实现这两部分进行介绍。

## 4.1用户端功能实现

### 4.1.1 用户端主页功能实现

用户进入首页之后会显示如下的一个窗口。系统以上中下的布局进行展示，首先映入眼帘的是系统的导航栏，下面是网站公告，再往下是各种各样的资讯，其主界面展示如下图4.1所示。

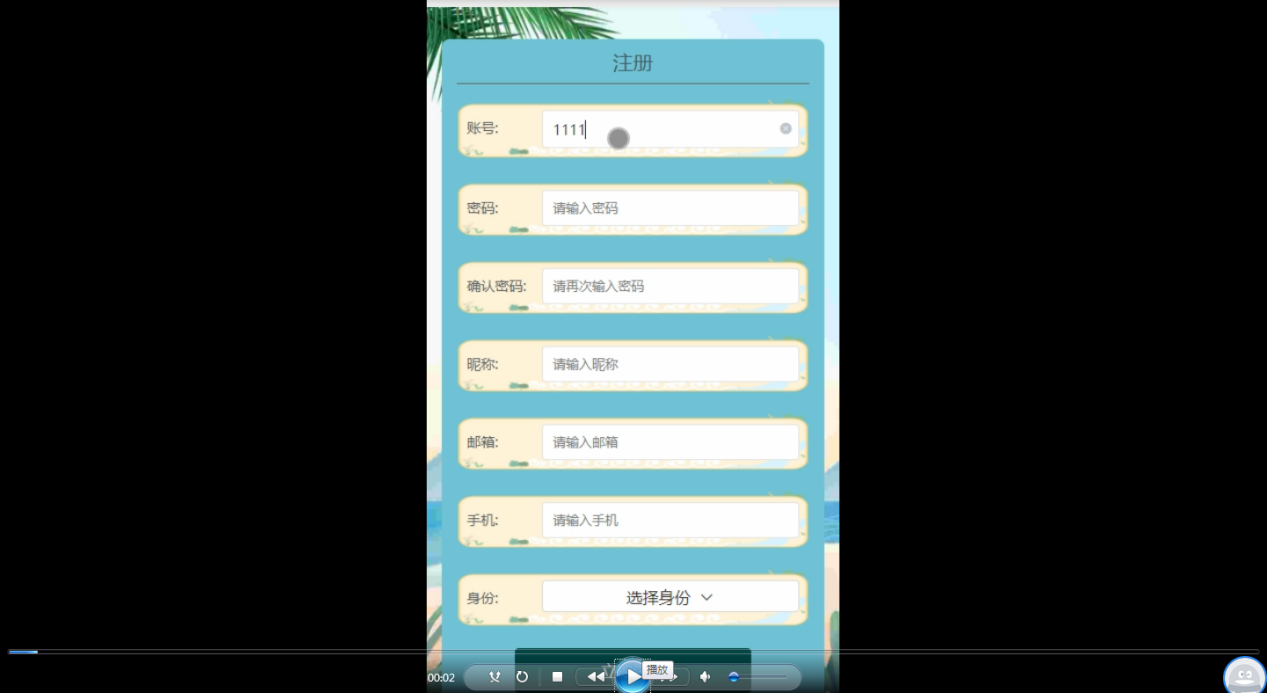
如下图：

图4.1校园心声墙小程序主界面

由于该项目只是用来测试，用户登录首页展示的图片等数据有的是从项目本地拿到的，有的则是从数据库中获取的。页面中的搜索部分，页头部分等都是属于公共样式，可以进行公告样式的抽取，然后再引入到jsp页面中。

### 4.1.2用户注册登录功能实现

不是校园心声墙小程序中正式用户的是可以在线进行注册的，当用户点击右上角的“注册”的时候，，当填写上自己的账号+密码+确认密码+昵称+邮箱+手机号+选择身份登信息后，后再点击“注册”按钮后将会先验证输入的有没有空数据，再次验证密码和确认密码是否是一样的，最后验证输入的账户名和数据库表中已经注册的账户名是否重复，只有都验证没问题后即可注册成功。用户注册界面展示如下图4.2所示。

图4.2校园心声墙小程序用户注册界面

系统用户注册关键代码如下。

/\*\*

\* 注册

\* @param user

\* @return

\*/

@PostMapping("register")

public Map<String, Object> signUp(@RequestBody User user) {

// 查询用户

Map<String, String> query = new HashMap<>();

query.put("username",user.getUsername());

List list = service.select(query, new HashMap<>()).getResultList();

if (list.size()>0){

return error(30000, "用户已存在");

}

user.setUserId(null);

user.setPassword(service.encryption(user.getPassword()));

service.save(user);

return success(1);

}

/\*\*

\* 用户ID：[0,8388607]用户获取其他与用户相关的数据

\*/

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "user\_id")

private Integer userId;

/\*\*

\* 账户状态：[0,10](1可用|2异常|3已冻结|4已注销)

\*/

@Basic

@Column(name = "state")

private Integer state;

/\*\*

\* 所在用户组：[0,32767]决定用户身份和权限

\*/

@Basic

@Column(name = "user\_group")

private String userGroup;

/\*\*

\* 上次登录时间：

\*/

@Basic

@Column(name = "login\_time")

private Timestamp loginTime;

/\*\*

\* 手机号码：[0,11]用户的手机号码，用于找回密码时或登录时

\*/

@Basic

@Column(name = "phone")

private String phone;

/\*\*

\* 手机认证：[0,1](0未认证|1审核中|2已认证)

\*/

@Basic

@Column(name = "phone\_state")

private Integer phoneState;

/\*\*

\* 用户名：[0,16]用户登录时所用的账户名称

\*/

@Basic

@Column(name = "username")

private String username;

/\*\*

\* 昵称：[0,16]

\*/

@Basic

@Column(name = "nickname")

private String nickname;

/\*\*

\* 密码：[0,32]用户登录所需的密码，由6-16位数字或英文组成

\*/

@Basic

@Column(name = "password")

private String password;

/\*\*

\* 邮箱：[0,64]用户的邮箱，用于找回密码时或登录时

\*/

@Basic

@Column(name = "email")

private String email;

/\*\*

\* 邮箱认证：[0,1](0未认证|1审核中|2已认证)

\*/

@Basic

@Column(name = "email\_state")

private Integer emailState;

/\*\*

\* 头像地址：[0,255]

\*/

@Basic

@Column(name = "avatar")

private String avatar;

/\*\*

\* 创建时间：

\*/

@Basic

@Column(name = "create\_time")

@JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss")

private Timestamp createTime;

@Basic

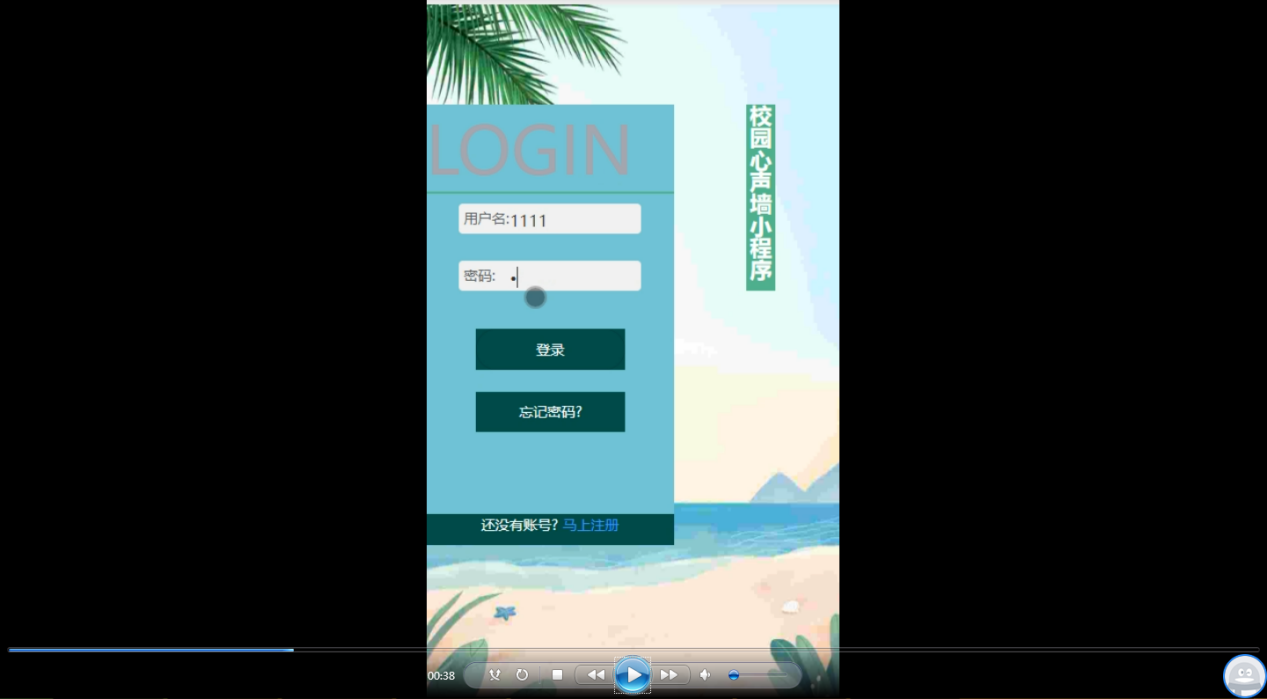
@Transient

private String code;

}

### 4.2.3 登录功能实现

校园心声墙小程序中的用户端上注册后的用户是可以通过自己的账户名和密码进行登录的，当用户输入完整的自己的账户名和密码信息并点击“登录”按钮后，将会首先验证输入的有没有空数据，再次验证输入的账户名+密码和数据库中当前保存的用户信息是否一致，只有在一致后将会登录成功并自动跳转到校园心声墙小程序的首页中；否则将会提示相应错误信息，用户登录界面如下图4.3所示。

图4.3校园心声墙小程序登录界面

登录关键代码如下。

/\*\*

\* 登录

\* @param data

\* @param httpServletRequest

\* @return

\*/

@PostMapping("login")

public Map<String, Object> login(@RequestBody Map<String, String> data, HttpServletRequest httpServletRequest) {

log.info("[执行登录接口]");

String username = data.get("username");

String email = data.get("email");

String phone = data.get("phone");

String password = data.get("password");

List resultList = null;

Map<String, String> map = new HashMap<>();

if(username != null && "".equals(username) == false){

map.put("username", username);

resultList = service.select(map, new HashMap<>()).getResultList();

}

else if(email != null && "".equals(email) == false){

map.put("email", email);

resultList = service.select(map, new HashMap<>()).getResultList();

}

else if(phone != null && "".equals(phone) == false){

map.put("phone", phone);

resultList = service.select(map, new HashMap<>()).getResultList();

}else{

return error(30000, "账号或密码不能为空");

}

if (resultList == null || password == null) {

return error(30000, "账号或密码不能为空");

}

//判断是否有这个用户

if (resultList.size()<=0){

return error(30000,"用户不存在");

}

User byUsername = (User) resultList.get(0);

Map<String, String> groupMap = new HashMap<>();

groupMap.put("name",byUsername.getUserGroup());

List groupList = userGroupService.select(groupMap, new HashMap<>()).getResultList();

if (groupList.size()<1){

return error(30000,"用户组不存在");

}

UserGroup userGroup = (UserGroup) groupList.get(0);

//查询用户审核状态

if (!StringUtils.isEmpty(userGroup.getSourceTable())){

String sql = "select examine\_state from "+ userGroup.getSourceTable() +" WHERE user\_id = " + byUsername.getUserId();

String res = String.valueOf(service.runCountSql(sql).getSingleResult());

if (res==null){

return error(30000,"用户不存在");

}

if (!res.equals("已通过")){

return error(30000,"该用户审核未通过");

}

}

//查询用户状态

if (byUsername.getState()!=1){

return error(30000,"用户非可用状态，不能登录");

}

String md5password = service.encryption(password);

if (byUsername.getPassword().equals(md5password)) {

// 存储Token到数据库

AccessToken accessToken = new AccessToken();

accessToken.setToken(UUID.randomUUID().toString().replaceAll("-", ""));

accessToken.setUser\_id(byUsername.getUserId());

tokenService.save(accessToken);

// 返回用户信息

JSONObject user = JSONObject.parseObject(JSONObject.toJSONString(byUsername));

user.put("token", accessToken.getToken());

JSONObject ret = new JSONObject();

ret.put("obj",user);

return success(ret);

} else {

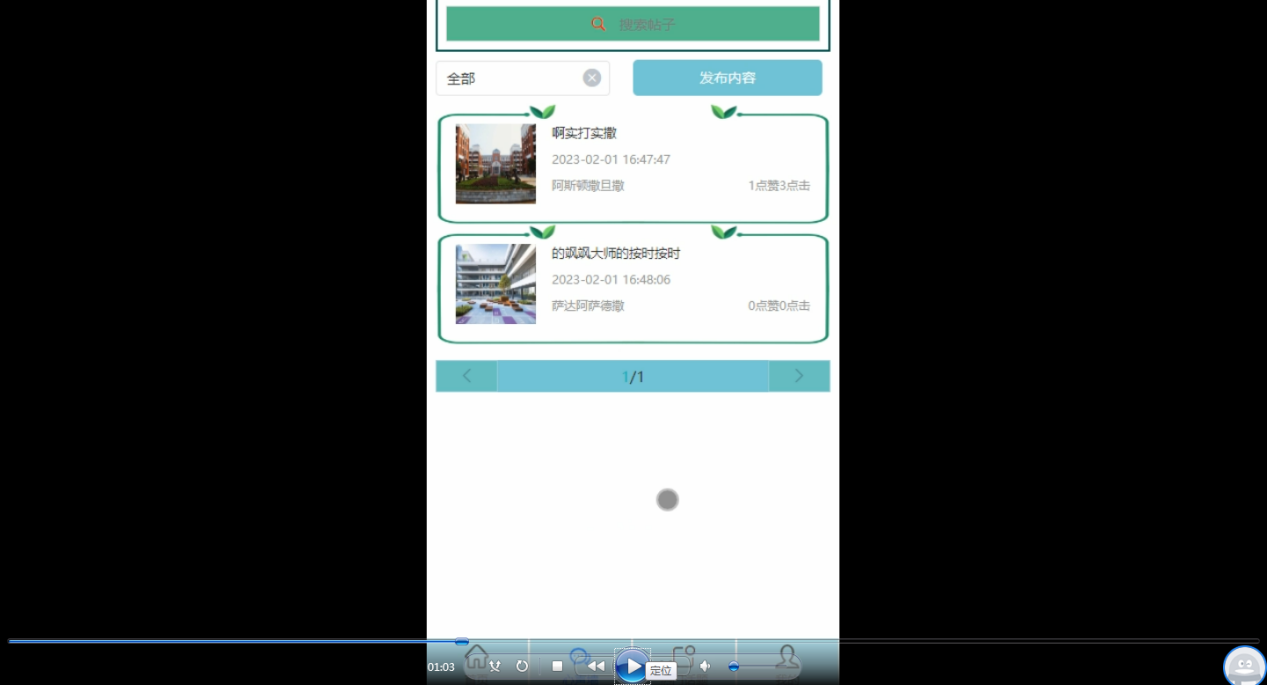
return error(30000, "账号或密码不正确");

}

}

### 4.2.4心声墙功能实现

当用户点击“心声墙”后，用户可以搜索查看心声墙所有帖子，对喜欢的帖子进行收藏、点赞评，也可发布帖子；如图4.4所示：

图4.4心声墙界面

心声墙关键代码如下所示。

@PostMapping("/add")

@Transactional

public Map<String, Object> add(HttpServletRequest request) throws IOException {

service.insert(service.readBody(request.getReader()));

return success(1);

}

@Transactional

public Map<String, Object> addMap(Map<String,Object> map){

service.insert(map);

return success(1);

}

public Map<String,Object> readBody(BufferedReader reader){

BufferedReader br = null;

StringBuilder sb = new StringBuilder("");

try{

br = reader;

String str;

while ((str = br.readLine()) != null){

sb.append(str);

}

br.close();

String json = sb.toString();

return JSONObject.parseObject(json, Map.class);

}catch (IOException e){

e.printStackTrace();

}finally{

if (null != br){

try{

br.close();

}catch (IOException e){

e.printStackTrace();

}

}

}

return null;

}

public void insert(Map<String,Object> body){

StringBuffer sql = new StringBuffer("INSERT INTO ");

sql.append("`").append(table).append("`").append(" (");

for (Map.Entry<String,Object> entry:body.entrySet()){

sql.append("`"+humpToLine(entry.getKey())+"`").append(",");

}

sql.deleteCharAt(sql.length()-1);

sql.append(") VALUES (");

for (Map.Entry<String,Object> entry:body.entrySet()){

Object value = entry.getValue();

if (value instanceof String){

sql.append("'").append(entry.getValue()).append("'").append(",");

}else {

sql.append(entry.getValue()).append(",");

}

}

sql.deleteCharAt(sql.length() - 1);

sql.append(")");

log.info("[{}] - 插入操作：{}",table,sql);

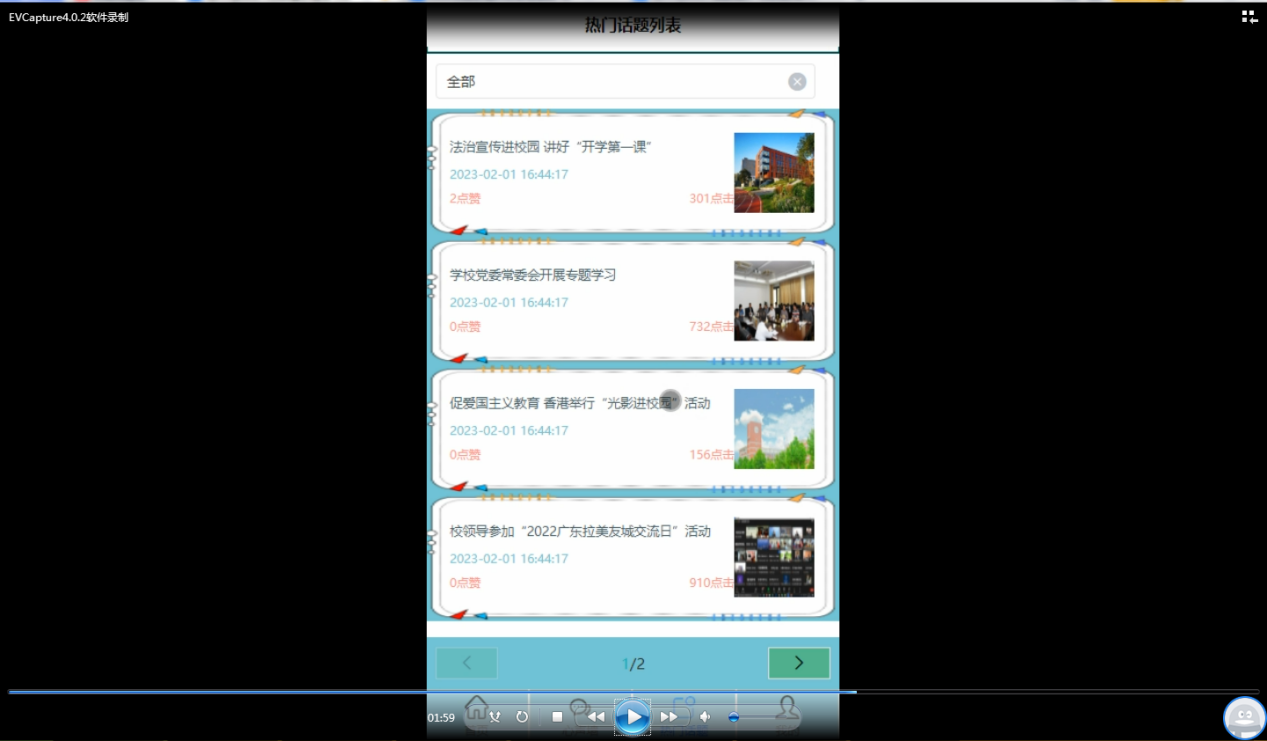
Query query = runCountSql(sql.toString());

query.executeUpdate();

}

### 4.2.5热门话题功能实现

当用户点击“热门话题”后，是可以对热门话题进行查看、点赞、收藏、评论等操作；具体的实现界面如下图4.5所示：

图4.5热门话题界面

热门话题关键代码如下所示。

@PostMapping("/upload")

public Map<String, Object> upload(@RequestParam("file") MultipartFile file) {

log.info("进入方法");

if (file.isEmpty()) {

return error(30000, "没有选择文件");

}

try {

//判断有没路径，没有则创建

String filePath = System.getProperty("user.dir") + "\\target\\classes\\static\\upload\\";

File targetDir = new File(filePath);

if (!targetDir.exists() && !targetDir.isDirectory()) {

if (targetDir.mkdirs()) {

log.info("创建目录成功");

} else {

log.error("创建目录失败");

}

}

String fileName = file.getOriginalFilename();

File dest = new File(filePath + fileName);

log.info("文件路径:{}", dest.getPath());

log.info("文件名:{}", dest.getName());

file.transferTo(dest);

JSONObject jsonObject = new JSONObject();

jsonObject.put("url", "/api/upload/" + fileName);

return success(jsonObject);

} catch (IOException e) {

log.info("上传失败：{}", e.getMessage());

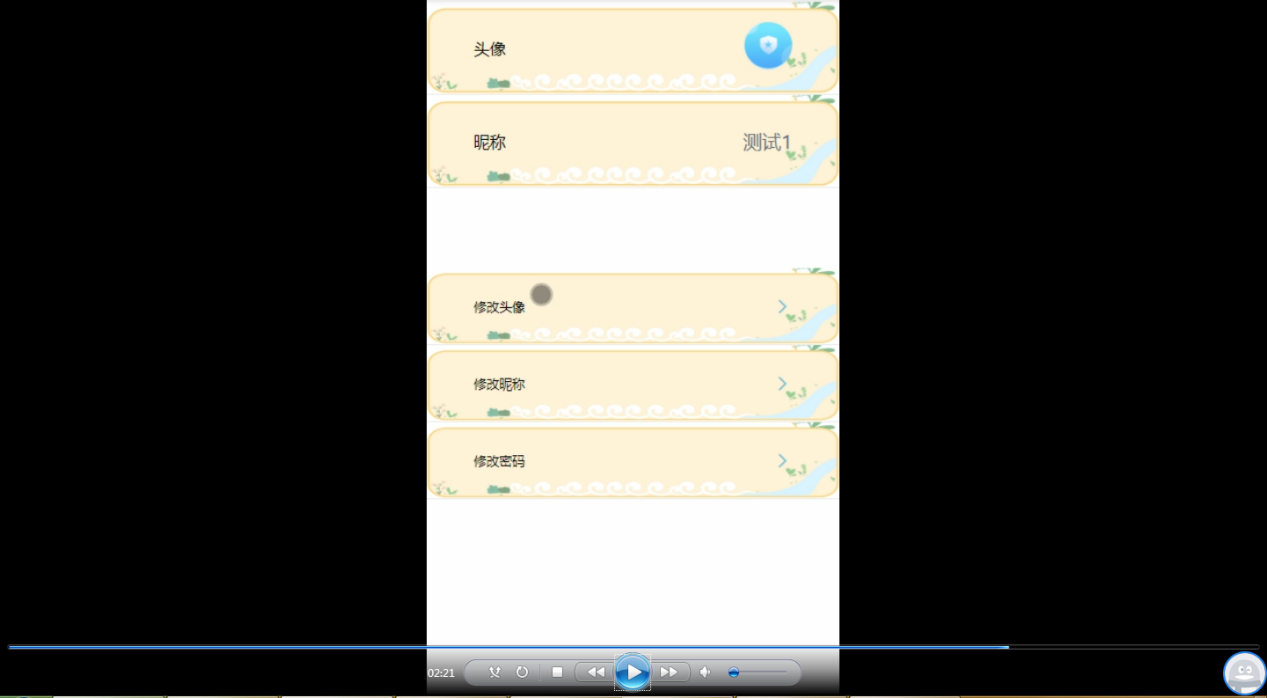
}

return error(30000, "上传失败");

}

### 4.2.6我的功能实现

从基于微信小程序校园心声墙小程序用户端底部的“我的”菜单进入后选择“基本信息”后可以对头像、昵称、密码等信息进行修书，界面如下图4.6所示。

图4.6我的界面

我的关键代码如下所示。

@RequestMapping("/get\_list")

public Map<String, Object> getList(HttpServletRequest request) {

Map<String, Object> map = service.selectToPage(service.readQuery(request), service.readConfig(request));

return success(map);

4.2 管理端功能实现

### 4.2.1管理端管理员登录功能实现

校园心声墙小程序中的管理端的管理员是可以通过自己的账户名和密码进行登录的，当管理员输入完整的自己的账户名和密码信息并点击“登录”按钮后，将会首先验证输入的有没有空数据，再次验证输入的账户名+密码和数据库中当前保存的用户信息是否一致，只有在一致后将会登录成功并自动跳转到校园心声墙小程序的首页中；否则将会提示相应错误信息，管理员登录界面如下图4.8所示。

图4.8管理端管理员登录界面

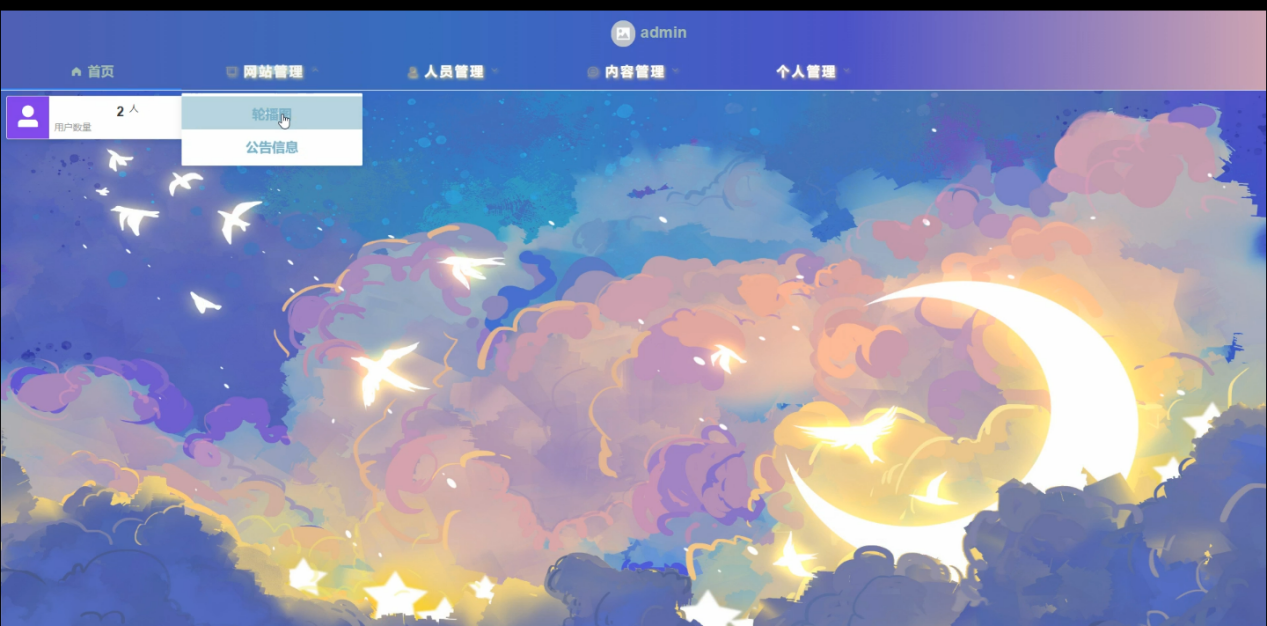
### 4.2.2人员管理功能实现

管理员在人员管理模块，可以对用户信息进行管理，人员管理模块由管理员，学生用户两部分组成，这里以学生用户管理为例，具体的实现界面如下图4.9所示：

图4.9人员管理界面

### 4.2.3网站管理功能实现

校园心声墙小程序中的管理人员在“网站管理”这一菜单中是可以对用户端显示的轮播图以及公告信息进行管控。界面如下图4.10所示：

图4.10网站管理界面

网站管理关键代码如下所示。

@RequestMapping("/get\_obj")

public Map<String, Object> obj(HttpServletRequest request) {

Query select = service.select(service.readQuery(request), service.readConfig(request));

List resultList = select.getResultList();

if (resultList.size() > 0) {

JSONObject jsonObject = new JSONObject();

jsonObject.put("obj",resultList.get(0));

return success(jsonObject);

} else {

return success(null);

}

}

public Query select(Map<String,String> query,Map<String,String> config){

StringBuffer sql = new StringBuffer("select ");

sql.append(config.get(FindConfig.FIELD) == null || "".equals(config.get(FindConfig.FIELD)) ? "\*" : config.get(FindConfig.FIELD)).append(" ");

sql.append("from ").append("`").append(table).append("`").append(toWhereSql(query, "0".equals(config.get(FindConfig.LIKE))));

if (config.get(FindConfig.GROUP\_BY) != null && !"".equals(config.get(FindConfig.GROUP\_BY))){

sql.append("group by ").append(config.get(FindConfig.GROUP\_BY)).append(" ");

}

if (config.get(FindConfig.ORDER\_BY) != null && !"".equals(config.get(FindConfig.ORDER\_BY))){

sql.append("order by ").append(config.get(FindConfig.ORDER\_BY)).append(" ");

}

if (config.get(FindConfig.PAGE) != null && !"".equals(config.get(FindConfig.PAGE))){

int page = config.get(FindConfig.PAGE) != null && !"".equals(config.get(FindConfig.PAGE)) ? Integer.parseInt(config.get(FindConfig.PAGE)) : 1;

int limit = config.get(FindConfig.SIZE) != null && !"".equals(config.get(FindConfig.SIZE)) ? Integer.parseInt(config.get(FindConfig.SIZE)) : 10;

sql.append(" limit ").append( (page-1)\*limit ).append(" , ").append(limit);

}

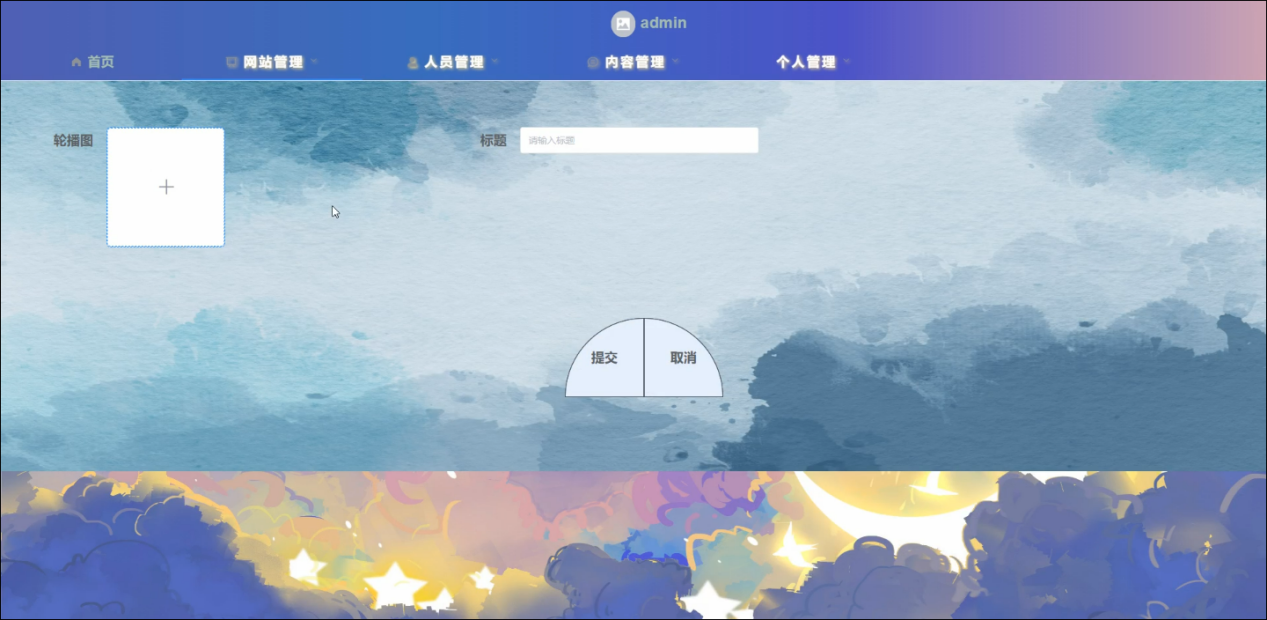
log.info("[{}] - 查询操作，sql: {}",table,sql);

return runEntitySql(sql.toString());

}

### 4.2.4轮播图管理界面

管理人员在“轮播图管理”这一菜单中是可以对前台显示的轮播图进行增删改查，具体的实现界面如下图4.11所示：

图4.11轮播图管理界面

轮播图管理关键代码如下所示。

@RequestMapping(value = {"/count\_group", "/count"})

public Map<String, Object> count(HttpServletRequest request) {

Query count = service.count(service.readQuery(request), service.readConfig(request));

return success(count.getResultList());

}

### 4.2.5公告消息界面

管理系统中的管理人员在“公告消息”这一菜单中是可以对校园心声墙小程序内的公告消息可以进行增删改查操作，具体的实现界面如下图4.12所示：

图4.12公告消息界面

公告消息管理关键代码如下所示。

@RequestMapping(value = {"/avg\_group", "/avg"})

public Map<String, Object> avg(HttpServletRequest request) {

Query count = service.avg(service.readQuery(request), service.readConfig(request));

return success(count.getResultList());

}

### 4.2.6内容管理功能实现

管理人员在“内容管理”这一菜单下是可以对心声墙、心声分类、热门话题、话题分类，这四个子菜单，管理员能够对这四个模块进行管控的，具体的实现界面如下图4.13所示：

图4.13内容管理界面

# 5系统测试

系统开发到了最后一个阶段那就是系统测试，系统测试对软件的开发其实是非常有必要的。因为没什么系统一经开发出来就可能会尽善尽美，再厉害的系统开发工程师也会在系统开发的时候出现纰漏，系统测试能够较好的改正一些bug，为后期系统的维护性提供很好的支持。通过系统测试，开发人员也可以建立自己对系统的信心，为后期的系统版本的跟新提供支持。

## 5.1软件测试的重要性

只有在运行和维护阶段之前经历大量的测试的软件，才能说明它的质量是经得起检验的。最近计算机业界也都一致认为，测试应该存在于软件设计的每个阶段，因为越早发现错误，修复起来就越容易。

实际上，对于一个软件应用，错误是必然存在的，无论使用何种技术或手段，都不可能绝对的排除软件漏洞。测试是随着软件开发一同诞生的，两者是共同发展进步的。实际上，测试可以大幅度的降低维护的成本，如果一个漏洞在开发的早期就被发现，那么修复它的成本远比上线后再修复的成本要低得多。

## 5.2测试实例的研究与选择

测试有白盒测试和黑盒测试两种方式。

其中，白盒测试是将软件看成一个透明的白盒子，按照程序的内部控制结构和处理技术逻辑来选定测试用例、软件系统测试的逻辑路径及过程需要进行管理测试，又称玻璃盒测试。因此白盒测试需要选择足够多的测试用例，覆盖尽可能多的代码来发现程序中的错误。

黑盒测试，也称为功能测试。它将需软件看作一个黑盒，像一个普通用户一样来模拟软件的使用流程。黑盒测试通过大量的输入边界值或错误数据，来检查是否可产生正确的输出。

本系统测试 主要选择黑盒测试，少量采用白盒测试。通过测试达到以下测试目的：

1.检查各大功能模块的运行，确保其能够正确运行，并检查各页面的完整性，保证页面完整。

2.检查各个接口是否可以正确地输入和输出，保证数据流通稳定可行。

3.检查数据结构，保证其和外部接口没有访问错误，访问顺利。

4.检查原计划的性能需求有没有完成，运行流畅。

本系统的测试用例（部分）：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **登录部分测试用例** | | | | | |
| **编号** | **对象** | **项目** | **操作** | **预期结果** | **结果** |
| 1 | 登录 | 登录提示 | 使用正确的账号密码登录 | 成功登录 | 预期结果 |
| 2 | 登录提示 | 使用正确的账号但错误的密码登录 | 提示密码错误 | 预期结果 |
| 3 | 登录提示 | 使用错误的账号登录 | 提示不存在账户 | 预期结果 |
| 4 | 登录提示 | 不输入账号，点击登录 | 提示输入账号 | 预期结果 |
| 5 | 登录提示 | 输入账号但不输入密码点击登录 | 提示输入密码 | 预期结果 |
| 6 | 登录入口 | 已登录账号，查看登录入口 | 不显示登录入口 | 预期结果 |

## 5.3测试环境与测试条件

处理器：Inter Core I7-4710MQ四核处理器

内存：4GB

硬盘：1T

操作系统：Windows 10

数据库：MySQL

## 5.4系统运行情况

全部测试用例都已通过（包括但不限于以上测试用例），且不存在漏洞，实现了论文开始时所作要求。本系统运行稳定，使用流畅，可以满足客户需求。

## 5.5系统评价

### 5.5.1系统功能评价

试运行后进行系统评估，可以认为该系统达到预定的目标要求，可以满足用户的需求，也满足了系统开发前所作目标。

### 5.5.2系统技术评价

系统在经过大量重复测试后运行十分稳定，安全实用，功能模块已经达到预定目标所需。

### 5.5.3系统经济评价

在规定的时间内实现系统的大部分功能，且满足要求，节省开发成本，有助于提高科学管理水平，符合本人经济情况。

# 6.总结

通过校园心声墙小程序的开发，本人巩固了之前学过的知识，如今将平时所学到的知识融合在设计中，在设计过程中，做了很多的准备，首先，在数据库系统的设计过程中，尤其是在数据库的工作原理、工作特点，对其深刻的讨论，与此同时，对于小型站点来说，最好服务器的选择，其次，利用所学的知识点分析所做的系统，并在此基础上设计。

目前本系统已经上线，正在试运行阶段，用户反馈良好，基本完成用户所需，试运行过程中没有出现阻断性问题，有一些不足和小问题也及时予以修正，系统上线后，为了保证数据的安全性，对系统进行了备份操作，系统备份是每两个月备份一次，数据库备份为每周备份一次，系统部署在租赁的云系统服务器中。

本次系统上线成功后，得到了用户的高度认可，但是在功能上和性能上还需做进一步的研究处理，使其有更高的性能和更好的用户体验。

系统在以后的升级过程中，需要解决一系列用户所提出的问题，例如打印过程中如何避免浏览器的兼容性问题，大量用户访问时，如何保持较高的响应速度，在系统今后的升级过程中将着重解决这些安全性问题。

# 参考文献

[1]陈少龙,张凡.实验室信息化管理微信小程序的设计[J].湖北理工学院学报,2022,38(06):6-10+43.

[2]姚雨辰,李瑜,王天航,蔡伊涵.基于微信小程序的校园资源回收的模式创新[J].数字技术与应用,2022,40(11):113-115.DOI:10.19695/j.cnki.cn12-1369.2022.11.34.

[3]李钊.基于微信小程序的基层社会治理积分制应用研究[J].现代信息科技,2022,6(22):97-100.DOI:10.19850/j.cnki.2096-4706.2022.22.024.

[4]李锦姬,李甜甜,崔佳伟,徐驰,蒙程. 基于微信小程序的患者能量摄入及出入量计算分析系统的设计与实现[C]//.中国医学装备大会暨2022医学装备展览会论文汇编（上册）.,2022:233-237.DOI:10.26914/c.cnkihy.2022.041897.

[5]何良超,冯国柱. 基于微信小程序的技术监督安全培训系统设计与实现[C]//.2022年电力行业技术监督工作交流会暨专业技术论坛论文集.,2022:1253-1258.DOI:10.26914/c.cnkihy.2022.044323.

[6]张笑月,杨海苓,陈园园,曹雁南.基于微信小程序的医院后勤报修平台设计与应用[J].中国医院建筑与装备,2022,23(11):64-67.

[7]张丽景,张文川.基于对比分析法的高职“MySQL数据库”课程设计——以兰州石化职业技术大学为例[J].现代信息科技,2022,6(21):188-191.DOI:10.19850/j.cnki.2096-4706.2022.21.046.

[8]周迪迪,程峰,李耀,朱俊,彭磊. 一种基于Springboot+vue框架的配置化报表开发方法[P]. 湖北省：CN115185520A,2022-10-14.

[9]谢霜.基于Java技术的网络管理体系结构的应用[J].网络安全技术与应用,2022(10):14-15.

[10]苏婉怡,揣小龙,赵国松,王煜尧.基于Java技术的考试系统关键点研究[J].无线互联科技,2022,19(18):58-60.

[11]焦宇,李民,王欢,余开朝.基于MySQL性能调优的推荐系统优化设计[J].软件导刊,2022,21(09):108-112.

[12]林佳一.混合式教学在高职MySQL数据库课程的实践研究[J].电脑知识与技术,2022,18(26):133-135.DOI:10.14004/j.cnki.ckt.2022.1737.

[13]王曼维,杨荻,李岩,及松洁.基于SpringBoot框架的智慧医疗问诊系统设计与实现[J].中国医学装备,2022,19(03):133-136.

[14]熊柏祥.基于Springboot和Vue框架的考试资源服务平台的设计与实现[J].信息与电脑(理论版),2022,34(01):97-99+103.

[15]曹明昊. 基于SpringBoot和Vue框架的邯郸市现代农业园区信息管理系统的研发[D].河北工程大学,2021.DOI:10.27104/d.cnki.ghbjy.2021.000671.

[16]He Xinbin,Bai Yongbin,Yue Lisen,Wang Haixiao,Liu Yi. Design and Implementation of Information System Based on Java Technology Platform[J]. Journal of Physics: Conference Series,2021,2033(1).

[17]Qu Xiaona. Application of Java Technology in Dynamic Web Database Technology[J]. Journal of Physics: Conference Series,2021,1744(4).

# 致 谢

大学生活在这个时候即将划上一个句号，但是对于我的人生道路来说，这仅仅是一个逗号，我将面对的是又一次征程的开始。

回忆过去，许许多多的事情浮现在脑海：刚上大学时欢乐心情和兴奋的场景还历历在目。一切都是那么新鲜，那么富有吸引力。有快乐也有艰辛，有收获也有失落。衷心感谢信息学院所有支持帮助过我的老师，谢谢你们多年来的关心和爱护。同窗的友情同样难忘，你们与我共同走过了人生中不平凡的道路，给我留下了值得珍藏的美好记忆。

最后，我要特别感谢指导论文设计的老师。本论文是在他的悉心指导和热情帮助下完成的，老师认真负责的工作态度，严谨的治学精神和精深的理论水平都使我受益匪浅。老师无论在理论上还是在实践中，都给予我很大的帮助，使我专业技能的应用水平得到很大提高，这对于我以后的工作和学习都有益处。值此论文完成之际，特别向老师表示衷心的感谢和崇高的敬意，谢谢他细心而又耐心地辅导，使得我得以顺利的完成毕业设计开发工作，同时也要感谢其他帮助过我的老师和同学，他们在我成长过程中给予了我很大的帮助，在此一并表示感谢。

由于本人水平有限，加上时间紧促，本文一定有不少缺点和不足，恳请各位老师给予帮助和指正。