分类号：TP311.5 UDC：D10621-408-(2019)2403-0

密　级：公开编号：**2015081147**

成都信息工程大学

学位论文

ACM小程序的设计与实现

|  |  |
| --- | --- |
| 论文作者姓名： | 袁曦焜 |
| 申请学位专业： | 软件工程 |
| 申请学位类别： | 工学学士 |
| 指导教师姓名（职称）： | 杜晓宇（讲师） |
| 论文提交日期： | 2019年5月20日 |

ACM小程序的设计与实现

**摘　要**

随着我校ACM竞赛的不断发展，队员数量的不断增多，传统的消息传递方式已经无法满足快速增长的信息数量。为了解决上述问题，将设计实现一款ACM微信小程序，使同学们通过微信小程序即能实时获取ACM校队的信息，能及时参加校队活动和比赛，能更多了解ACM信息，激发竞赛的兴趣，积极参与到竞赛中。

本文将详细介绍ACM微信小程序的设计与实现，用户可以使用该程序阅读校队最新的新闻，及时了解校队发布的公告，报名校赛，也可以对校队成员贴上标签。本系统后端使用Spring+SpringMVC+MyBatis的架构，管理员后台管理界面使用react框架，用户前台使用微信小程序的框架，前后端分离的开发模式；使用Mybatis连接Mysql数据库，实现对数据的存储。本系统分为未授权用户、正式用户和管理员，根据不同的权限具有不同的功能。用户主要实现登陆注册、公告查看、新闻查看、校赛查看和报名、签到、印象添加点赞和值日功能，管理员主要实现用户管理、公告管理、新闻管理、校赛管理、值日管理和印象管理功能。

**关键词：** react；SSM；Mysql；ACM；微信小程序；

The Design and Implementation of ACM WeChat Mini Program

**Abstract**

With the continuous development of the ACM competition in our school, the number of players has increased, and the traditional way of messaging has been unable to meet the rapidly growing amount of information. In order to solve the above problems, we will design and implement an ACM WeChat applet, so that students can get the information of ACM school team in real time through WeChat applet, and can participate in school team activities and competitions in time, can learn more about ACM information and stimulate competition. Interest in actively participating in the competition.

This article will introduce the design and implementation of ACM WeChat applet in detail. Users can use this program to read the latest news of the school team, keep abreast of the announcements issued by the school team, sign up for the school, and label the team members. The backend of the system uses Spring, SpringMVC and MyBatis architecture, the administrator background management interface uses the react framework, the user foreground uses the framework of the WeChat applet, and the development mode is separated from the front and the back; using Mybatis to connect to the Mysql database to realize the storage of data. The system is divided into unauthorized users, official users and administrators, with different functions according to different permissions. The user mainly implements login registration, announcement viewing, news viewing, school competition viewing and registration, sign-in, impression add-on and value date functions, and the administrator mainly implements user management, announcement management, news management, school competition management, and value management. Impression management function.

**Key words:** react; SSM; Mysql; ACM; WeChat Mini Program

**目 录**

论文总页数：30页

[1 引言 1](#_Toc10064782)

[1.1 课题背景 1](#_Toc10064783)

[1.2 国内外现状 1](#_Toc10064784)

[1.3 本课题研究的意义 1](#_Toc10064785)

[1.4 本课题的研究方法 2](#_Toc10064786)

[2 需求分析及项目方案 2](#_Toc10064787)

[2.1 需求分析 2](#_Toc10064788)

[2.2 项目开发工具分析及选择 5](#_Toc10064789)

[2.2.1 开发工具及技术选择 5](#_Toc10064790)

[2.2.2 开发技术具体介绍 5](#_Toc10064791)

[3 系统总体设计 6](#_Toc10064792)

[3.1 系统功能模块设计 6](#_Toc10064793)

[3.1.1 用户管理 6](#_Toc10064794)

[3.1.2 公告管理 7](#_Toc10064795)

[3.1.3 新闻管理 8](#_Toc10064796)

[3.1.4 校赛管理 9](#_Toc10064797)

[3.1.5 值日管理 10](#_Toc10064798)

[3.1.6 签到管理 10](#_Toc10064799)

[3.1.7 印象管理 10](#_Toc10064800)

[3.2 系统数据库设计 11](#_Toc10064801)

[3.2.1 数据库概念设计 11](#_Toc10064802)

[3.2.2 数据库逻辑设计 11](#_Toc10064803)

[3.2.3 数据库表结构详细说明 12](#_Toc10064804)

[4 系统模块功能实现 17](#_Toc10064805)

[4.1用户管理 17](#_Toc10064806)

[4.1.1 注册 17](#_Toc10064807)

[4.1.2 登录 18](#_Toc10064808)

[4.2 新闻管理 18](#_Toc10064809)

[4.2.1 查看新闻 18](#_Toc10064810)

[4.2.2 分类管理 18](#_Toc10064811)

[4.2.3 新闻管理 19](#_Toc10064812)

[4.3 公告管理 20](#_Toc10064813)

[4.3.1 查看公告 20](#_Toc10064814)

[4.3.2 管理公告 20](#_Toc10064815)

[4.3.3 公告置顶 20](#_Toc10064816)

[4.4 校赛管理 20](#_Toc10064817)

[4.4.1 用户操作 20](#_Toc10064818)

[4.4.2 管理校赛 21](#_Toc10064819)

[4.5 值日管理 21](#_Toc10064820)

[4.5.1 查看值日信息 21](#_Toc10064821)

[4.5.2 管理值日信息 21](#_Toc10064822)

[4.6 印象管理 21](#_Toc10064823)

[4.6.1 用户操作 21](#_Toc10064824)

[4.6.2 印象管理 22](#_Toc10064825)

[4.7 签到模块 22](#_Toc10064826)

[5 系统测试 23](#_Toc10064827)

[5.1 系统测试环境 23](#_Toc10064828)

[5.2 系统功能测试 23](#_Toc10064829)

[5.3 本章小结 26](#_Toc10064830)

[结 束 语 27](#_Toc10064831)

[参考文献 28](#_Toc10064832)

[致　　谢 29](#_Toc10064833)

[声　　明 30](#_Toc10064834)

# 引言

## 课题背景

近年来，网络信息不断膨胀，移动互联网技术不断发展，单纯的从PC端获取信息已经无法赶上信息的增长速率，需与移动设备相结合传播信息[1]。但移动APP并不方便，无法满足用户方便快捷的需求，为此微信推出了一款基于微信，具有轻量快捷、硬件要求低等特点的微信小程序[2]。

我校ACM竞赛不断发展，队员数量不断上升，但是却没有一个较官方的平台为同学们提供实时的信息和了解ACM校队。鉴于上述问题，笔者将设计开发一款ACM微信小程序让同学们通过小程序即能实时的获取ACM校队的信息，能及时参加校队活动和比赛。能够让更多的同学了解到ACM的信息，参与到竞赛中来。

ACM微信小程序的使用，可以提高同学们对竞赛的了解，提高对竞赛的认知，促进同学们的交流，提高学习竞赛的热情和兴趣。

## 国内外现状

据微信公开数据显示，当前已上线小程序总数超过100万，开发者超过150万，而用户方面，小程序日活已达2.8亿，人均使用20个小程序，人均使用时长超过10分钟[3]。截至2018年底，微信小程序用户规模超6亿，经常使用用户仅占34%，微信小程序已覆盖超过200个细分行业，服务超过1000亿人次用户[4]。微信对小程序的宣传也十分到位，经常开展小程序开发竞赛，“小程序化”已经成为一种必然。

对国内而言，各高校ACM校队虽拥有自己的Web端官网，但信息更新及传递并不及时，同时由于智能机等移动设备更为普及，用户更愿意从移动设备中获取信息，因此大部分ACM校队的信息传递方式仍停留在传统社交软件中。但通过社交软件获取信息会使用户获取信息不及时，无法从大量的聊天记录中获取有用信息，从而导致信息传递阻塞。

对国外而言，高校在ACM宣传方面更多使用官网及国外社交软件，使用较多是Instagram、Facebook、Twitter等的社交平台，但也存在国内相似的信息阻塞问题。

为了解决上述问题，需一个轻量级程序去承载这一需求，由于微信的普遍性和小程序的便携性，本文选择在微信上开发设计开发一款ACM小程序去传递ACM相关信息。

## 本课题研究的意义

数据结构与算法是计算机程序的灵魂，是朴素的基础，学好数据结构有助于理解各种上层抽象的结构[5] [6]。ACM竞赛是锻炼和学习数据结构与算法的途径，因此通过竞赛的学习可以帮助同学们很好的掌握数据结构的学习和使用。

本着促进同学们学习竞赛的兴趣，提高校队管理相关事务的效率。通过本系统，同学们可以方便快捷的了解校队新闻，获得的奖项，最新的公告信息和校赛信息，也可以贴上对大神的印象。提高对竞赛的认知，能够促进同学们的交流，提高学习竞赛的热情和兴趣。校赛报名系统、值日系统和签到系统也可以方便校队管理相关事务。

## 本课题的研究方法

本系统使用IntelliJ IDEA和VSCode**开发工具进行开发，Tomcat8.0作为服务器。**利用MySQL+Redis进行数据存储，**使用Spring+SpringMVC+MyBatis作为后端框架进行开发，管理员后台管理界面使用React架构，用户展现页面使用微信小程序架构，前后端分离的开发模式[7][8]。使用**Mybatis对jdbc的封装，让数据库底层操作变的透明[9]。使得系统易于维护和拓展。

# 需求分析及项目方案

## 需求分析

本系统的用户分为未通过审批的用户、正式用户和管理员。通过本系统，管理员对用户信息、公告、新闻、校赛、值日、印象几大模块进行管理；用户可以通过小程序获取公告、新闻、校赛、值日、印象的信息，并对校赛进行报名。

未通过审批用户端用例图如图 2‑1所示：



图 2‑1未通过审批用户端用例图

对于未通过审批用户进行详细说明：

（1）注册登录：用户在首次进入系统时，系统会强制要求注册，注册后成为未通过审批用户，在管理员通过审批后才能成为正式用户。

（2）查看新闻：未通过审批用户在小程序端可以在小程序中查看新闻列表和每条新闻的详情内容。

（3）查看公告：未通过审批用户可以在小程序端查看公告列表和详情。

（4）查看校赛：未通过审批用户可以在小程序端查看校赛列表和校赛详情。

（5）查看值日：未通过审批用户可以在小程序端查看当天值日信息和所有的值日信息。

（7）查看印象：未通过审批用户可以在小程序查看所有用户被贴上的印象。

正式用户端用例图如图 2‑2所示：



图 2‑2正式用户端用例图

对于正式用户功能进行详细说明：

（1）注册登录：用户在首次进入系统时，系统强制要求注册，注册后成为未通过审批用户，在管理员通过审批后成为正式用户。

（2）新闻管理：用户在小程序端可以在小程序中查看新闻列表和详情。

（3）公告管理：用户可以在小程序端查看公告列表和详情。

（4）校赛管理：用户可以在小程序端查看校赛列表和校赛详情，报名参赛和取消报名。

（5）值日管理：用户可以在小程序端查看当天值日信息和所有的值日信息。

（6）签到管理：用户可以在小程序端进行签到。

（7）印象管理：用户可以在小程序查看所有用户的被贴上的印象，添加印象和对印象进行点赞。

管理员端用例图如图 2‑3：



图 2‑3管理员端用例图

对于管理员功能进行详细说明：

（1）用户管理：在此模块中，管理员在管理页面可以对注册的用户进行审核和修改信息。

（2）新闻管理：在此模块中，管理员在管理页面可以添加新闻、修改新闻和删除新闻。

（3）公告管理：此模块中，管理员可以在管理页面添加公告、修改公告、删除公告和对公告进行置顶和取消。

（4）校赛管理：在此模块中，管理员可以在管理页面添加校赛、修改校赛、删除校赛，管理报名校赛的用户并将名单导出。

（5）值日管理：在此模块中，管理员可以在管理页面修改值日人员。

（6）印象管理：在此模块中，管理员可以在管理页面对印象进行修改和删除。

## 项目开发工具分析及选择

### 开发工具及技术选择

本系统使用IntelliJ IDEA和VSCode进行开发，利用MySQL和Redis进行数据存储，Spring、SpringMVC和MyBatis相结合作为后端框架进行开发，管理员后台管理界面使用react架构，用户展示界面使用微信小程序架构。

### 开发技术具体介绍

（1）IntelliJ IDEA

IntelliJ IDEA是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等功能超常[10]。

（2）微信小程序

微信小程序是2016年微信团队推出的一种基于微信的新型应用号，小程序的运行环境分成渲染层和逻辑层，其中 WXML 模板和 WXSS 样式工作在渲染层，JS 脚本工作在逻辑层[11]。小程序的通信模型图如图 2‑4所示：



图 2‑4 微信小程序通信图

渲染曾使用WebView对页面进行渲染，逻辑层运用JsCore县城运行JS脚本，微信客户端（Native）作为开发者服务器和微信小程序的中转站，将对应的用户请求发送给第三方服务器[12]。

（4）MySQL和Redis

MySQL 是一个高性能且相对简单的数据库系统，与其他更大系统的设置和管理相比，其复杂程度较低，优化的SQL查询算法，有效地提高查询速度[13]。Redis 是一款完全开源免费的key-value内存数据库， 通常被认为是一款数据结构服务器，使用标准C语言编写，所有数据和计算都在内存中完成，具有速度快的特点[14]。

（5）SSM框架

SSM框架是Spring、SpringMVC和MyBatis相结合，是目前主流的Java后端框架，大部分Java开发工程师都会选择SSM进行开发。Spring框架是 [Java](https://zh.wikipedia.org/wiki/Java) 平台的一款[开源](https://zh.wikipedia.org/wiki/%E5%BC%80%E6%94%BE%E6%BA%90%E4%BB%A3%E7%A0%81)的全栈[应用程序框架](https://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6%E6%A1%86%E6%9E%B6)和[控制反转](https://zh.wikipedia.org/wiki/%E6%8E%A7%E5%88%B6%E5%8F%8D%E8%BD%AC)容器实现[15]。MyBatis是一款[Java](https://zh.wikipedia.org/wiki/Java)[持久化框架](https://zh.wikipedia.org/wiki/%E6%8C%81%E4%B9%85%E5%8C%96%E6%A1%86%E6%9E%B6)，通过[XML](https://zh.wikipedia.org/wiki/XML)描述符或注解把[对象](https://zh.wikipedia.org/wiki/%E5%AF%B9%E8%B1%A1_(%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%A7%91%E5%AD%A6))与[存储过程](https://zh.wikipedia.org/wiki/%E5%AD%98%E5%82%A8%E8%BF%87%E7%A8%8B)或[SQL](https://zh.wikipedia.org/wiki/SQL)语句关联起来，对数据进行操作[16]。SpringMVC 框架提供了模型-视图-控制的体系结构和可以用来开发灵活、松散耦合的 web 应用程序的组件[17]。三个框架结合起来，能够快速实现项目的开发和迭代。

（6）React框架

React 是一款全新思路的前端 UI 框架，它完全接管了 UI 开发中最为复杂的局部更新部分，擅长在在复杂场景下保证高性能[18]；同时，引入了基于组件的开发思想，从另一个角度来重新审视 UI 的构成。通过这种方法，不仅能够提高开发效率，而且可以让代码更容易理解，维护和测试[19]。

# 系统总体设计

## 系统功能模块设计

本系统由注册登录、新闻管理、公告管理、签到、值日管理、校赛管理、印象管理几大模块组成，分析可得如图 3‑1所示的系统功能结构图：



图 3‑1系统功能结构图

### 用户管理

（1）注册

用户的微信号必须绑定系统账号才能进入系统，登录的时候会检测该微信号是否绑定系统账号，如果绑定了则自动跳转到系统首页。若微信号未绑定系统账号，则跳转到注册页面，输入注册信息，系统会检测信息是否符合规范，如不符合规范，会提示错误，注册成功后会自动跳转到系统首页。

注册流程图如图 3‑2所示：



图 3‑2 注册流程图

（2）登录

用户进入小程序首页会根据从微信服务器获取的用户编号进行登录，如果没有注册过微信小程序的账号会跳转至注册页面，注册过即自动登录，进入系统首页。

登录流程图如图 3‑3所示：



图 3‑3 登录流程图

（3）用户审核

管理员在后台管理页面可以查看注册了但还未审批通过的用户的信息，核对用户信息无误后可使用户通过审核。

### 公告管理

（1）查看公告

用户在进入系统后，可以点击公告模块浏览公告列表，也可以点击某一个公告浏览详情。

（2）管理公告

管理员进入对应管理页面后可对公告信息进行管理。可以进行添加公告、修改公告、公告置顶或取消置顶及删除公告。

添加公告：管理员可以发布近期公告，让用户能获取近期最新通知。

修改公告：如若公告信息有所变动，管理员可在相应修改页面修改公告信息。

删除公告：若管理员觉得某公告已过期或不需要，管理员可以删除该条公告，减少用户查看的数据量，更加简洁。

公告置顶：管理员可将重要消息置于顶部，便于用户查看。

取消置顶：管理员可以取消已置顶的公告信息。

因此，公告管理流程图如图 3‑4所示：



图 3‑4 公告管理流程图

### 新闻管理

管理员对新闻相应信息进行管理，主要分为分类管理及新闻管理。

（1）查看新闻

用户在进入系统后，可以点击新闻模块浏览新闻列表，也可以点击某一个新闻浏览详情。流程图如图 3‑5所示：



图 3‑5 查看新闻流程图

（2）分类管理

每篇新闻会有对应的分类属性，添加新闻时会要求选择对应的新闻分类信息，同时管理员可对新闻的分类进行管理。具体分为添加分类、修改分类和删除分类。

添加分类：管理员可以添加对应的分类信息。

删除分类：管理员可以删除某些不想要的分类信息。

（3）新闻管理

管理员登录后进入新闻管理页面，对新闻相关信息进行管理。具体分为添加新闻、修改新闻、删除新闻。详情如下：

添加新闻：管理员可以发布最新ACM相关新闻，让用户及时获取信息。

修改新闻：管理员可以修改已发布的新闻信息。

删除新闻：管理员可以删除废弃的新闻信息。

新闻管理流程图如图 3‑6所示：



图 3‑6 新闻管理流程图

### 校赛管理

（1）管理校赛

管理员进入校赛管理页面，可对校赛信息进行管理。具体分为添加校赛、修改校赛、删除校赛、查看报名人员、导出报名信息。

添加校赛：管理员添加即将开展的校赛信息，用户可查看对应的校赛信息。

修改校赛：管理员可修改已发布的校赛信息。

删除校赛：管理员可删除发布错误或者已过时的校赛。

查看报名人员：管理员可以查看某场校赛对应的用户报名信息。

导出校赛报名名单：管理员可以将某场校赛的报名人员的用户信息导出excel。

校赛管理流程图如图 3‑7所示：



图 3‑7 校赛管理流程图

（2）查看校赛

用户在进入系统后，可以点击校赛模块浏览校赛列表，也可以点击某一个校赛浏览详情。

（3）报名校赛

用户在校赛详情中可以报名想参加的校赛，也可以取消报名。

### 值日管理

（1）查看值日

用户在进入系统后，可以在系统首页查看今日的值日信息，也可以点击值日查看所有日期的值日信息。

（2）管理值日

管理员在进入系统后，可以对值日的信息进行修改。点击修改按钮，即可修改指定日期的值日信息。

### 签到管理

用户在进入小程序首页后有对应的签到按钮，可以点击签到进行签到，系统会根据上次签到时间判断今日是否已经签到，如若签到成功会显示该用户连续签到的时间，如若已签到会显示对应的失败信息。

### 印象管理

1. 查看印象

用户在进入系统后，可以点击印象进入印象模块查看所有用户被添加的点赞数最高的三个印象，点击用户可以查看该用户的所有印象。流程图如图 3‑8所示：



图 3‑8 查看印象流程图

1. 增加印象

在用户的详细印象界面中，用户可以对该用户增加新的印象。用户输入对该用户的印象并提交，该印象即会出现在改名用户的印象列表中，此时点赞数为1。

1. 印象点赞

在用户的详细印象界面中，用户并可以对该用户被添加的印象进行点赞。用户对赞同的印象点击即可使该印象的点赞数增加1。

1. 管理印象

管理员在进入系统后，可以查看所有用户的所有印象信息，同时页可以对用户的印象进行删除和点赞数修改的操作。

## 系统数据库设计

### 数据库概念设计

根据需求分析和系统设计，可以画出本系统中使用的数据库E-R图如图 3‑9所示：



图 3‑9数据库E-R图

对图 3‑9进行说明：

（1）1个用户可以管理多个新闻。

（2）1个用户可以管理多个公告。

（3）1个用户可以管理多个校赛。

（4）1个用户可以管理多个校赛人员。

（5）1个用户可以报名多个校赛。

（6）1个用户可以添加多个印象。

（7）1个用户可以点赞多个印象。

（8）1个用户可以管理多个值日。

### 数据库逻辑设计

数据库具体的关系表如图 3‑10所示：



图 3‑10数据库具体关系表

本系统中数据库的关系表如下：user是用户信息表，competition是校赛信息表，applycompetition是报名校赛信息表，news是新闻信息表，newclasss是新闻属性信息表，classification是新闻分类信息表，announcement是公告信息表，attendance是签到信息表，impression是印象信息表，dayduty是值日信息表。

### 数据库表结构详细说明

该系统设计的数据库包括：用户信息表、校赛信息表、报名校赛信息表、新闻信息表、新闻属性信息表、新闻分类信息表、公告信息表、签到信息表、印象信息表、值日信息表。各个数据库表的简单解释说明如下：

（1）用户信息表(user)：用于存储用户的具体信息，用户包括用户名、密码、手机号、学号、头像、该用户在微信小程序的编号和权限信息，将用户编号设为逐渐，用户进行操作时，记录操作用户的用户编号，方便查询创建人信息，同时还需存储用户是否被删除。用户信息表如表 3‑1所示：

表 3‑1用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 规则 | 说明 |
| userId | int | pk,not null | 用户编号，自增 |
| username | varchar(50) | not null | 用户昵称 |
| password | varchar(50) | not null | 用户密码 |
| mobile | varchar(50) | not null | 手机号 |
| studentId | varchar(50) |  | 学号 |
| createDay | datetime |  | 创建时间 |
| image | varchar(50) |  | 用户头像上传后的访问url |
| auth | int | not null | 用户权限，0-未授权，1-授权用户，4-管理员 |
| unionid | varchar(50) | not null | 微信小程序对该用户的编号 |
| isEffective | int | not null | 是否被删除，0-删除，1-未删除 |

（2）新闻信息表(news)：用于存储新闻的具体内容，新闻信息包括新闻标题、新闻内容、创建信息、修改信息，同时还需要记录创建人和该新闻是否公开的信息、它的存活状态。因为在管理员添加新闻页面，管理员添加成功后不会立即跳转至详情页面，再次点击提交进行的是更新操作，因此新闻的编号不为自增型，而是一个18位的随机数。新闻信息表如表 3‑2所示：

表 3‑2新闻信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| newsId | bigint | pk,not null | 新闻编号，随机生成的bigint |
| newsTitle | varchar(100) | not null | 新闻标题 |
| newsBody | text | not null | 新闻内容 |
| createUser | bigint | not null | 创建用户编号 |
| createDate | datatime | not null | 创建日期 |
| updateUser | bigint | not null | 修改用户编号 |
| updateDate | datetime | not null | 修改时间 |
| isPublic | int | not null | 是否公开，0-未公开，1-公开 |
| isEffective | int | not null | 是否被删除，0-删除，1-未删除 |

（3）新闻分类表(newclass)：用于存储新闻与分类的对应关系，存储包括新闻的编号、指向的分类的编号和创建时间，同时记录创建用户的编号，此外还需记录它的存活状态。通过该表可以获取信息和分类的对应编号信息，通过mybatis查询，可以获取单独新闻的对应信息。

新闻的分类表如表 3‑3所示：

表 3‑3新闻分类表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| newsClassId | bigint | pk,not null | 新闻分类编号，自增 |
| newsId | bigint | fk,not null | 新闻信心编号 |
| classId | bigint | fk,not null | 分类信息编号 |
| createUser | bigint | not null | 创建用户编号 |
| createDate | datetime |  | 创建日期 |
| updateUser | bigint | not null | 修改用户编号 |
| updateDate | datetime | not null | 修改时间 |
| isEffective | int | not null | 是否被删除，0-删除，1-未删除 |

（4）新闻分类信息表(classification)：用于存储新闻分类的信息，包括分类的编号、分类的名称、创建时间以及创建人等信息，还有修改人和修改的时间信息，同时记录它的存活状态。因为分类不会在创建的同时被使用，因此分类的编号可以是自增型。

新闻分类信息表如表 3‑4所示：

表 3‑4新闻分类信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| classId | bigint | pk,not null | 新闻分类编号，自增 |
| classname | bigint | fk,not null | 新闻分类的名称 |
| createUser | bigint | not null | 创建用户编号 |
| createDate | datetime |  | 创建日期 |
| updateUser | bigint | not null | 修改用户编号 |
| updateDate | datetime | not null | 修改时间 |
| isEffective | int | not null | 是否被删除，0-删除，1-未删除 |

1. 公告信息表（announcement）：用于存储公告的具体内容，公告包括公告的标题、内容和创建用户等信息，以及该公告是否被公开，同时记录它的存活状态。因为在管理员添加公告页面，管理员添加成功后不会立即跳转至公告详情页面，再次点击提交进行的是更新公告操作，因此公告的编号不为自增型，而是一个18位的随机数。

公告信息表结构如表 3‑5所示：

表 3‑5公告信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| announced | bigint | pk,not null | 公告编号，随机生成的bigint |
| announceTitle | varchar(100) | not null | 公告标题 |
| announceBody | text | not null | 公告内容 |
| announcecreateUser | bigint | not null | 创建用户编号 |
| announcecreateDate | datatime | not null | 创建日期 |
| announceupdateUser | bigint | not null | 修改用户编号 |
| announceupdateDate | datetime | not null | 修改时间 |
| isPublic | int | not null | 0-未公开，1-公开 |
| isEffective | int | not null | 是否被删除，0-删除，1-未删除 |
| isFirst | int | not null | 是否置顶，0-未置顶，1-置顶 |

1. 校赛信息表（competition）：用于存储校赛的具体内容，包括校赛的标题、描述、比赛时间等信息，同时需记录创建用户的编号和修改用户的编号及时间，此外还需记录它的存活状态。

校赛信息表结构如表 3‑6所示：

表 3‑6校赛信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| competitionId | bigint | pk,not null | 比赛编号,随机生成的bigint |
| competitionTitle | varchar(100) | not null | 比赛标题 |
| competitionBody | text | not null | 比赛描述 |
| createUser | bigint | not null | 创建用户编号 |
| createDate | datatime | not null | 比赛日期 |
| updateUser | bigint | not null | 修改用户编号 |
| updateDate | datetime | not null | 修改时间 |
| isEffective | int | not null | 是否被删除，0-删除，1-未删除 |

1. 校赛报名表（announcement）：用于存储报名信息，包括报名参加校赛的编号、用户的编号和报名时间等信息，同时记录它的存活状态。

校赛报名表结构如表 3‑7所示：

表 3‑7校赛报名表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| applyCompetitionId | bigint | pk,not null | 报名校赛编号，自增 |
| competitionId | bigintbigint | fk,not null | 对应的校赛编号 |
| joinUser | bigint | not null | 报名用户编号 |
| createDate | datetime | not null | 报名时间 |
| isEffective | int | not null | 是否删除，0-删除，1-未删除 |

1. 签到信息表：用于存储签到的信息，包括签到用户的编号、时间和连续签到天数，同时记录它的存活状态。

签到信息表结构如表 3‑8所示：

表 3‑8签到信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| attendanceId | bigint | pk,not null | 签到编号，编号自增 |
| createUser | bigint | not null | 签到用户编号 |
| createDate | datatime | not null | 签到日期 |
| sumDay | int | not null | 连续签到天数 |
| isEffective | int | not null | 是否删除，0-删除，1-未删除 |

1. 印象信息表：用于存储用户印象的具体内容，包括被添加印象的用户编号、创建时间和添加印象的用户编号，同时记录它的存活状态。

印象信息表结构如表 3‑9所示：

表 3‑9印象信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| impressionId | bigint | pk,not null | 印象编号,编号自增 |
| impressionTitle | Varchar(100) | not null | 印象标题 |
| imUser | bigint | not null | 被添加印象的用户编号 |
| createDate | datatime | not null | 创建日期 |
| createUser | bigint | not null | 修改用户编号 |
| isEffective | int | not null | 是否被删除，0-删除，1-未删除 |

1. 值日信息表：用于存储值日的信息，包括值日的日期，值日人员信息、修改用户编号和时间等信息，同时记录它的存活状态。因为一周只有7天，因此该表只应有7条数据分别存储周一至周五的值日信息。

值日信息表结构如表 3‑10所示：

表 3‑10值日信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **规则** | **说明** |
| dayDutyId | bigint | pk,not null | 值日编号，只有1-7 |
| dayName | Varchar(50) | not null | 值日日期，星期一-星期天 |
| dayUserNames | Varchar(100) | not null | 值日人姓名 |
| createUser | bigint | not null | 创建用户编号 |
| createDate | datatime | not null | 创建日期 |
| updateUser | bigint | not null | 修改用户编号 |
| updateDate | datetime | not null | 修改时间 |
| isEffective | int | not null | 是否被删除，0-删除，1-未删除 |

# 系统模块功能实现

本系统主要包含注册登录、新闻、公告、签到、值日、校赛、印象几个模块的功能，以下分别对这七个模块进行说明。

## 4.1用户管理

本模块由注册和登录两个功能组成。用户在第一次使用该系统时需注册账号，在注册账号后会与微信绑定后即可进入本系统。

### 注册

用户在注册界面需要输入手机号、学号、真实姓名、入学年份的信息，系统会对输入的信息进行校验，并提示对应的错误信息；若是信息正确，执行registerUser接口，向数据库中添加对应的用户信息，将信息和微信号绑定，并跳转到系统首页。

注册页面如图 4‑1所示：



图 4‑1注册页面

### 登录

用户在进入小程序时，系统会根据微信服务器返回的微信编号判断该微信号是否已完成注册，如未注册过会跳转到注册页面；如用户已经注册过则自动跳转到小程序首页。

## 新闻管理

本模块主要是由管理员在后台对新闻进行管理和用户在小程序端查看新闻组成。

### 查看新闻

用户可以点击新闻进入新闻模块，点击对应的新闻标题，获取对应新闻编号，系统会调用detailNews接口，首先校验是否存在该编号的新闻信息，其次查看该新闻的存活状态，获取新闻数据，返回至小程序端，通过小程序将对应数据依次渲染在对应的位置上。

代码如下所示：

public ResultBean detail(User user, long newsId){

try {

List<News> news = newsService.findNewsListByNewsId(newsId);

if (news.size()==0) { //判断是否存在id未newsId的新闻

LOG.info("该id新闻不存在");

return new ResultBean(ResultCode.PARAM\_ERROR, "不存在该新闻");

}

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

map.put("newsId", newsId);

map.put("isEffective", 1);

//获取新闻对应的分类

List<Classification> classifications=classificationService.findNewsClasfication(map);

//将新闻内容和新闻分类组合，返回

NewsView newsView = new NewsView(news.get(0), classifications);

return new ResultBean(ResultCode.SUCCESS, newsView);

} catch (Exception e) {//异常

e.printStackTrace();

LOG.error(e.getMessage());

return new ResultBean(ResultCode.SYSTEM\_FAILED);

}

}

### 分类管理

（1）添加分类：管理员进入分类管理页面后可在左侧添加对应的分类名称，点击添加调用addClass接口，后端会判断该名称的分类是否已存在，否则添加成功。

（2）删除分类：管理员进入分类管理页面后可在右侧查看已存在的分类信息，用户可以点击删除按钮，删除不需要的分类信息。

效果图如图 4‑2所示：



图 4‑2 分类效果图

### 新闻管理

管理员在登录进后台管理页面，可以点击新闻模块，对新闻进行管理。

1. 添加新闻：点击添加按钮，会跳转到添加新闻的页面，管理员在输入标题、内容和分类之后，点击提交，前端会直接会对数据进行校验，如果正确则会传到后端进行数据库操作。

添加新闻页面如图 4‑3所示：

图片包含 屏幕截图

描述已自动生成

图 4‑3添加新闻页面

1. 删除新闻：点击删除按钮可以删除需要删除的新闻，后端的接口会进行校验，判断新闻是否存在，并且将该条新闻的状态设置为删除，然后把结果返回到前端。
2. 修改新闻内容：点击修改按钮，会跳转到修改新闻的页面，管理员在输入修改后的标题、内容和分类之后，点击提交，前端会直接会对数据进行校验，如果正确则会传到后端进行数据库操作。

## 公告管理

本模块主要是由管理员在后台对公告进行管理和用户在小程序端查看公告组成。

### 查看公告

用户可以点击公告进入公告模块，系统会调用后端接口，获取公告数据，返回到前端显示。点击公告标题会进入到该条公告的详情，此时前端会将该条公告编号传给后端，后端对编号进行校验，返回对应的公告数据。

### 管理公告

管理员在登录进后台，可以点击公告模块，对公告进行管理。

1. 添加公告：点击添加按钮，会跳转到添加公告的页面，管理员在输入标题、内容之后，点击提交，前端会直接会对数据进行校验，如果正确则会传到后端进行数据库操作。
2. 删除公告：点击删除按钮可以删除需要删除的公告，后端的接口会进行校验，判断公告是否存在，并且将该条公告的状态设置为删除，然后把结果返回到前端。
3. 修改公告内容：点击修改按钮，会跳转到修改公告的页面，管理员在输入修改后的标题、内容之后，点击提交，前端会直接会对数据进行校验，如果正确则会传到后端进行数据库操作。

### 公告置顶

管理员在公告管理列表可以对公告选择置顶或者取消，此时前端会将操作用户和公告的编号传给后端，后端对数据进行校验后，对数据库进行操作并返回给前端结果。

## 校赛管理

本模块主要是由管理员在后台对校赛及报名人员进行管理和用户在小程序端查看校赛和报名校赛组成。

### 用户操作

1. 查看校赛：用户可以点击校赛进入校赛模块，系统会调用后端接口，获取校赛数据，返回到前端显示。点击校赛标题会进入到该场校赛的详情，此时前端会将该条校赛编号传给后端，后端对编号进行校验，返回对应的结果。
2. 报名和取消报名校赛：在进入校赛详情之后，可以点击报名或取消报名，前端会将用户编号、校赛编号传入后端，后端会对校赛编号、用户、时间和报名状态进行校验，通过之后进行数据库操作并将结果返回到前端。

### 管理校赛

管理员在登录进后台，可以点击校赛模块，对校赛进行管理。

1. 添加校赛：点击添加按钮，会跳转到添加校赛的页面，管理员在输入标题、内容、比赛时间之后，点击提交，前端会直接会对数据进行校验，如果正确则会传到后端进行数据库操作。
2. 删除校赛：点击删除按钮可以删除需要删除的校赛，后端的接口会进行校验，判断校赛是否存在，并且将该场校赛的状态设置为删除，然后把结果返回到前端。
3. 修改校赛内容：点击修改按钮，会跳转到修改校赛的页面，管理员在输入修改后的标题、内容和比赛时间之后，点击提交，前端会直接会对数据进行校验，如果正确则会传到后端进行数据库操作。
4. 导出校赛报名信息：点击导出信息按钮，会执行export接口，后端会

## 值日管理

本模块主要是由管理员后台对值日信息的修改和用户在小程序端查看值日信息组成。

### 查看值日信息

用户可以进入小程序首页后，会前端会将当前时间传给后端，后端会对数据进行校验，如果通过则会调用数据库查询并返回给前端对应的信息。点击进入值日模块，后端会查询数据库把所有的值日信息返回给前端。

### 管理值日信息

（1）修改值日信息：点击修改按钮，会跳转到修改值日信息的页面，管理员在输入修改后值日信息之后，点击提交，前端会直接会对数据进行校验，如果正确则会传到后端进行数据库操作。

## 印象管理

本模块主要由用户查看用户印象、添加印象、点赞印象和管理员管理印象组成。

### 用户操作

（1）查看印象：用户点击印象模块，前端会将当前页码传给后端，后端对数据进行验证，无误后将该页码的印象信息返回给前端。点击用户，前端会将该用户的编号传给后端，后端校验无误后，进行数据库操作，将该用户的印象数据返回给前端。

查看印象的界面如图 4‑4所示：

图片包含 屏幕截图

描述已自动生成

图 4‑4印象界面图

（2）添加印象：在进入用户的印象页面后，可以对该用户添加印象,在输入印象并点击添加按钮之后，前端会对长度进行校验，并将内容和添加用户的编号和被添加用户的编号传给后端，进行数据校验，校验完成后返回对应的结果。

（3）点赞印象：用户在进入其他用户的印象页面后，对已有的印象进行点赞，前端会将印象的编号和操作用户的编号传给后端校验，校验成功后进行数据库操作并向前端返回结果。

### 印象管理

（1）查看所用用户印象：管理员进入对应管理页面可以查看所有用户的印象信息，通过调用selectImpression后端接口，后端将对应的信息返回前端，再将信息显示在页面上。

（2）删除印象：管理员进入印象管理页面，可以删除不适当的印象信息，调用deleteImpression接口，首先会判断印象是否存在，如若存在即可正常删除，否则返回提示信息。

## 签到模块

签到用户可以在小程序的首页点击签到进行签到，前端会将用户的编号传给后端，后端会判断该用户上次签到时间和当前时间是否在同一天，如果两次签到时间在同一天，会提示今天已签到，如果不在同一天，会记录连续签到天数并显示。代码如下所示：

·public ResultBean addAttendance(User user) {

Attendance attendance = attendances.get(0);

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

String date1 = sdf.format(attendance.getCreateDate()); //获取最近签到的日期

String date2 = sdf.format(new Date());//获取当前日期

if (date1.equals(date2)) { //是同一天

return new ResultBean(ResultCode.SYSTEM\_FAILED, "今天已签到");

}

Date date = new Date();

int dif = DateUtils.differentDays(attendance.getCreateDate(), date); //获取相差的天数

int day = 1; //总共差的天数

if (dif==1) { day = attendance.getSumDay()+1; }

Attendance add = new Attendance();

add.setCreateUser(user.getUserId().longValue());

add.setCreateDate(new Date());

add.setSumDay(day);

attendanceService.addAttendance(add); //存入数据库

return new ResultBean(ResultCode.SUCCESS, "连续签到"+day+"天");

}

# 系统测试

## 系统测试环境

为保证系统能够正常工作，对系统的各项功能进行了测试。后台的系统测试环境是Deepin 15.9操作系统和Chrome浏览器。

## 系统功能测试

该测试是针对模块的具体功能进行测试，为了确保所有的功能都能够正常运行。该模块主要针对的模块有用户信息管理、新闻、公告、校赛、印象、签到、值日管理。

1. 签到模块管理测试如表 5‑1所示：

表 5‑1签到模块管理测试

| 序号 | 输入 | 预期结果 | 是否正常 |
| --- | --- | --- | --- |
| 1 | 点击签到 | 1. 当天第一次点击签到，提示连续签到天数； 2. 当前第二次及以上次点击签到，提示今天已签到。 | 是 |

1. 用户信息管理模块测试如表 5‑2所示：

表 5‑2用户管理模块测试

| 序号 | 输入 | 预期结果 | 是否正常 |
| --- | --- | --- | --- |
| 1 | 管理员登录界面输入用户名、密码 | 1. 输入账号为空时，提示不能为空； 2. 输入密码为空，提示不能为空； 3. 账号和密码不匹配，提示密码输入错误或不存在该账号； 4. 点击登录，并且账号和密码信息输入均正确时，跳转至后台首页。 | 是 |
| 2 | 用户通过小程序登录 | 1. 如果该微信号已注册账号自动登录后跳转至首页； 2. 如果该微信号还未注册账号则跳转到注册页面。 | 是 |
| 3 | 注册界面输入必填信息 | 1. 若输入对应信息为空时，提示对应项为必填项； 2. 如输入信息格式错误，提示对应的错误信息； 3. 若手机号或学号已注册，则提示对应信息已存在； 4. 若填入的所有信息均没有问题，点击注册按钮，提示注册成功并且跳转到系统的首页。 | 是 |

1. 新闻管理模块测试如表 5‑3所示：

表 5‑3新闻管理模块测试

| 序号 | 输入 | 预期结果 | 是否正常 |
| --- | --- | --- | --- |
| 1 | 添加新闻输入标题、内容和类别 | 1. 若输入的内容或标题为空，提示对应项不能为空； 2. 若标题输入过长，提示长度过长； 3. 若提交时未选择新闻分类，提示未分类； 4. 内容输入完整，点击提交，添加成功并发布； 5. 内容输入完整，点击保存为草稿，保存为草稿。 | 是 |
| 2 | 修改新闻标题、内容和类别 | 1. 能够正确显示对应的新闻； 2. 输入修改的内容； 3. 点击修改，修改成功。 | 是 |
| 3 | 删除对应的新闻 | 1. 要被删除的新闻是正确存在的； 2. 点击删除，提示是否确认删除，防止误操作； 3. 点击确认删除，新闻删除成功。 | 是 |

1. 公告管理模块测试如表 5‑4所示：

表 5‑4公告管理模块测试

| 序号 | 输入 | 预期结果 | 是否正常 |
| --- | --- | --- | --- |
| 1 | 添加公告输入标题、内容 | 1. 若输入的内容或标题为空，提示对应项不能为空； 2. 若标题输入过长，提示长度过长； 3. 内容输入完整，点击提交，添加成功并发布。 | 是 |
| 2 | 修改公告标题、内容和类别 | 1. 能够正确显示对应的新闻； 2. 输入修改的内容； 3. 点击修改，修改成功。 | 是 |
| 3 | 删除对应的公告 | 1. 要被删除的公告是正确存在的； 2. 点击删除，提示是否确认删除； 3. 点击确认删除，公告删除成功。 | 是 |
| 4 | 公告置顶 | 1. 点击对应公告的置顶按钮，该公告提升到未置顶的公告之前。 2. 点击公告的取消置顶按钮，回到原本的位置。 | 是 |

1. 印象管理模块测试如表 5‑5所示：

表 5‑5印象管理模块测试

| 序号 | 输入 | 预期结果 | 是否正常 |
| --- | --- | --- | --- |
| 1 | 添加印象 | 1. 输入印象为空，提示不能为空； 2. 输入印象过长，提示内容长度过长； 3. 正确输入后，点击添加，提示添加成功。 | 是 |
| 2 | 对印象点赞 | 1.点击印象，印象的点赞数加一。 | 是 |
| 3 | 管理员删除印象 | 1. 要被删除的印象是正确存在的； 2. 点击删除对应的印象，提示是否确定删除，防止误操作； 3. 点击确认删除，印象删除成功 | 是 |

1. 值日模块管理测试如表 5‑6所示：

表 5‑6值日模块管理测试

| 序号 | 输入 | 预期结果 | 是否正常 |
| --- | --- | --- | --- |
| 1 | 修改值日信息 | 1. 若修改的输入信息为空，提示不能为空； 2. 若修改的输入信息过长，提示输入长度过长； 3. 点击修改按钮，提示修改成功。 | 是 |

1. 校赛管理模块测试如表 5‑7所示：

表 5‑7校赛管理模块测试

| 序号 | 输入 | 预期结果 | 是否正常 |
| --- | --- | --- | --- |
| 1 | 添加校赛输入标题、内容、时间 | 1. 若输入标题或内容为空，会提示不能为空； 2. 若标题输入信息过长，会提示长度过长； 3. 时间格式设置正确； 4. 添加成功。 | 是 |
| 2 | 修改校赛标题、内容时间 | 1. 能够正确显示对应的校赛； 2. 输入修改的内容； 3. 点击修改，修改成功。 | 是 |
| 3 | 删除对应的校赛 | 1. 要被删除的校赛是正确存在的； 2. 点击删除，提示是否确认删除，防止误操作； 3. 点击确认删除，校赛删除成功。 | 是 |
| 4 | 报名校赛和取消报名校赛 | 1. 点击报名校赛，提示成功报名校赛； 2. 校赛报名成功后，点击取消报名校赛，提示取消报名校赛成功。 | 是 |

## 本章小结

经过测试，ACM微信小程序的主要功能模块基本实现，且实际运行结果与预期基本一致。因此，该系统经测试符合要求。

结 束 语

本系统根据我校ACM校队的实际需求设计并实现了ACM微信小程序，基本实现了对用户信息、新闻、公告、校赛、用户印象、值日、签到的管理过程。为同学们提高竞赛的认知、提高学习竞赛的热情和兴趣提供了一个平台。同时也可以方便了校队管理相关事务。

经测试，该系统基本满足用户的基本的业务需求，具有较高的稳定性，同时用户界面也体现了良好的实用性、交互性和友好性。目前系统正常运行，已实现预期功能。

通过这次毕业设计的实现和论文的编写，使得我的代码实现能力和学习能力有了进一步的提升，对**Spring+SpringMVC+MyBatis**架构和React框架也更加熟悉，对数据库的运用更加熟练。也让我对一个完整的软件开发流程有了更进一步的认识，使得我具备了独立开发工程项目的能力，增强了我的代码力。

参考文献

1. 陈强. 移动互联网时代视觉化一致性的品牌营销研究[D]. 2017.
2. 微信小程序[OL]. 搜狐. http://www.sohu.com/a/235614901\_100126244, 2018-06-31.
3. 溪风. 微信小程序数量超过100万个，开发者超过150万人[OL]. IT之家. https://www.ithome.com/html/it/369819.htm. 2018-7-11
4. 微信小程序发展现状及发展趋势[OL]. 新浪博客. http://blog.sina.com.cn/s/blog\_19ffd89370102yoso.html. 2019-022-19.
5. 数据结构-一个程序员优秀与否的评判标准. 搜狐. http://www.sohu.com/a/282831121\_825834, 2018-12-18.
6. 宋杨, 贺薇. 试论“专业竞赛”在设计类专业学风建设中的作用--以南京理工大学设计艺术与传媒学院为例[J]. 文教资料, 2014(20):111-112.
7. 曾艳丽, 李诺. 针对SSM框架Web系统的相关思考[J]. 信息与电脑(理论版),2019(03):116-117.
8. 张妮.基于微信小程序的移动学习平台的设计开发研究[J].电脑知识与技术,2019,15(04):94-96.
9. 王诚. 研究JDBC连接MySQL数据库技术[J]. 网络安全和信息化,2018(12):68-69.
10. 陈立兵. IntelliJ IDEA——开发人员利器[J]. 程序员,2009(01):118-120.
11. 王婷婷. 微信小程序开发[J]. 信息技术与信息化,2018(12):62-63.
12. 刘刚. 微信小程序开发图解案例教程[M]. 人民邮电出版社, 2019.
13. 李莹.MySQL索引背后的数据结构及算法原理[J].现代交际,2015(08):117.
14. Jeremy Nelson. 深入理解Redis[M]. 电子工业出版社, 2017.
15. 温立辉. Spring框架在模型层的应用及原理[J]. 福建电脑,2017,33(05):147-148.
16. 杨开振. 深入浅出MyBatis技术原理与实战[M]. 电子工业出版社, 2016.
17. 向春会. 基于SpringMVC的web工程通用模板设计[D].北京邮电大学,2018.
18. 王沛. 颠覆式前端 UI 开发框架：React[OL]. InfoQ, https://www.infoq.cn/article/subversion-front-end-ui-development-framework-react. 2014-12-14.
19. Stoyan Stefanov. React快速上手开发[M]. 人民邮电出版社, 2017.

致　　谢

本文是在杜晓宇老师的指导下完成的，感谢他在百忙之中对我的论文进行指导和给出修改建议。他渊博的学识和严谨的治学态度使我受益匪浅，对本课题的顺利完成起到了极大的作用。同时也感谢学院老师的教导，为我打下了牢靠的基础，让我能独立完成本课题。

最后向百忙之中评审本文的各位专家、老师表示衷心的感谢！

声　　明

本论文的工作是2018年10月至2019年5月在成都信息工程大学软件工程学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

（1）学校有权保管并向有关部门递交学位论文的原件与复印件。

（2）学校可以采用影印、缩印或其他复制方式保存学位论文。

（3）学校可以学术交流为目的复制、赠送和交换学位论文。

（4）学校可允许学位论文被查阅或借阅。

（5）学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

除非另有科研合同和其他法律文书的制约，本论文的科研成果属于成都信息工程大学。

特此声明！

作者签名：

年 月 日