

分类号: TP311.5

UDC: D10621-408-(2020)2715-0

密 级: 公开

编号: 2016081187

成 都 信 息 工 程 大 学

学 位 论 文

CUIT-ACM 微信公众号的设计与实现

论 文 作 者 姓 名 :	谢仁义
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):	王嘉昀(讲师)
论 文 提 交 日 期 :	2020 年 6 月 12 日

CUIT-ACM 微信公众号的设计与实现

摘 要

随着微信的普及，微信公众号也成为了日常生活中最常见的程序。微信公众号依托于手机，操作方便，访问快捷，目前微信公众号开发已逐渐发展成一类标准。此外，我校越来越重视 ACM 竞赛，但是却缺乏一个用来宣传该类竞赛的相关公众号，为了扩大这一类竞赛的影响力，让我校更多学生能了解和加入，也让实验室的日常工作能更好的被记录和管理，CUIT-ACM 公众号应运而生。

本公众号依托于微信，通过微信端服务器与第三方服务器进行数据交换。后端基于 Spring Boot 的 SSM 框架，使用 Redis, MySQL 作为数据库，前端使用目前业界最流行的三大前端框架之一的 React 构建，利用 React 框架的组件化、单项数据流、增量更新等特性可以更好更快的开发前端页面。公众号根据不同的角色提供不同的操作权利，管理员权限包括新闻管理，公告管理，值日管理，友链管理，讨论区管理，评论管理，举报管理，反馈管理，用户管理等。而普通用户主要权限是讨论区管理，评论管理，举报管理，反馈管理。CUIT-ACM 微信公众号不仅可以让本校学生更好的了解和加入到该项竞赛中，还能提高实验室管理效率。

关键词：竞赛；微信公众号；ACM；Spring Boot；React

Design and Implementation of CUIT-ACM WeChat Official Account

Abstract

With the popularity of WeChat, the WeChat public account has become the most common program in daily life. Relying on mobile phones, the WeChat public account is easy to operate and quick to access. At present, the development of the WeChat public account has gradually developed into a class of standards. On the other hand, our school pays more and more attention to the ACM competition ^[1], but it lacks a relevant public account to promote this kind of competition. In order to expand the influence of this kind of competition, more students of our school can understand and Joining, also allows the daily work of the laboratory to be better recorded and managed, CUIT-ACM public number came into being.

This public account relies on WeChat to exchange data with third-party servers through WeChat server. The back-end is based on Spring Boot's SSM framework, using Redis and MySQL as the database, and the front-end is built using React, one of the three most popular front-end frameworks in the industry. The features of the React framework include componentization, single data flow, and incremental updates. Better and faster development of front-end pages. The public account provides different operation rights according to different roles. Administrator rights include news management, announcement management, duty management, friend chain management, discussion board management, comment management, report management, feedback management, user management, etc. The main permissions of ordinary users are discussion board management, comment management, report management, and feedback management. The CUIT-ACM WeChat public account not only allows our students to better understand and join the competition, but also improves laboratory management efficiency.

Key words: Contest; WeChat official Account; ACM; Spring Boot; React;

目 录

论文总页数：40 页

1	引言	1
1.1	课题背景与意义	1
1.2	国内外研究现状	1
1.3	本课题的研究方法	2
2	需求分析	2
2.1	系统需求概述	2
2.2	系统角色分析	2
2.3	系统功能性分析	4
2.4	可行性分析	5
3	系统总体设计	5
3.1	后端总体架构	5
3.2	微信公众号开发者模式	6
3.3	系统数据库设计	6
3.3.1	数据库概念设计	6
3.3.2	数据库物理结构模型	8
3.3.3	数据库表结构设计	8
3.4	系统功能设计	14
3.4.1	用户管理模块	15
3.4.2	新闻管理模块	16
3.4.3	公告管理模块	16
3.4.4	讨论区管理模块	18
3.4.5	值班管理模块	19
3.4.6	友链模块	19
3.4.7	其余模块	19
3.5	系统技术难点及关键技术	20
4	系统实现	20
4.1	系统开发环境	20
4.2	用户模块的实现	21
4.2.1	用户注册	21

4.2.2	用户登录	21
4.2.3	关注用户	22
4.2.4	用户管理	23
4.3	新闻管理模块的实现	24
4.4	公告管理模块的实现	25
4.4.1	添加公告	25
4.4.2	删除公告	25
4.4.3	修改公告	25
4.4.4	报名管理	25
4.4.5	类别管理	27
4.5	讨论区管理模块的实现	27
4.5.1	添加帖子	27
4.5.2	删除帖子	27
4.5.3	修改帖子	28
4.5.4	标签管理	28
4.5.5	评论管理	28
4.6	值班管理模块的实现	29
4.7	友链管理模块的实现	30
4.8	其余管理模块的实现	32
4.8.1	反馈管理	32
4.8.2	举报管理	32
5	系统测试	33
5.1	测试环境	33
5.2	测试用例及过程	34
5.2.1	用户管理模块测试	34
5.2.2	公告管理模块测试	34
5.2.3	讨论区管理模块测试	34
5.2.4	值班管理模块测试	35
5.2.5	友链管理模块测试	35
5.2.6	举报管理模块测试	36
5.3	测试结果分析	36
结 束 语		37
参考文献		38

致 谢	39
声 明	40

1 引言

1.1 课题背景与意义

随着微信公众号^[1]的快速发展，它已经是日常生活中最常见的程序了，在腾讯的 2016 年的总结报告中就给出了目前微信公众号的日活就已经突破了 5 亿，微信公众号的易于访问与操作便捷带来了巨大的流量，只需要一个能运行微信的手机就能操作。目前微信公众号开发逐渐发展成一类标准，再结合目前在高校中被越来越重视的 ACM 竞赛^[2]，为了扩大这一类竞赛的影响力，让我校更多学生能了解和加入进来以及让实验室的日常工作能更好的被记录和管理，同时开发微信公众号所需的金钱成本以及人工成本不高，结合微信公众号操作的便捷等特点，设计一款关于 ACM 的微信公众号是非常有必要的事情。

CUIT-ACM 微信公众号是基于微信服务器上的轻量 WEB 页面^[3]，它可以方便的让本校学生更好的了解和加入到该项竞赛中以及可以更好的管理实验室的日常工作，减轻老师和学生的负担。

1.2 国内外研究现状

在国内，微信的使用用户和日活都在不断的上涨，在 2019 年末微信用户甚至达到了 10 亿多，而微信公众号的数量也达到了近 3000 万个。目前微信公众号还在不断完善中，已经渗透到我们的日常生活中^[6]，各行各业都在使用着微信公众号进行传递信息，如平时我们手机微信中收到的各种公众号的消息推送。另一方面，虽然国内的大部分高校宣传 ACM 竞赛都是通过官方提供好的微信公众号，这种官方提供的公众号目的是方便使用，并且高度定制可修改的不多，所以在功能上非常局限，常见的就是用于发一些简单消息通知等，并没有提供更多额外的功能。而在自定义开发公众号上，由于国内的公众号流量巨大，开发者也非常的多。微信公众号的基础是一款基于手机的网页应用，从刚开始普通的电脑端网页适配到手机端。到后面随着前端框架的不断更新和迭代，现在的公众号是可以基于 Vue 开发的，并且越来越多的第三方工具的开发也使得公众号的开发形式越来越多样化。公众号开发的一些优良特性比如可以内网穿透方便 Debug，同时依托于微信给出的强大文档支持，使得开发公众号非常容易入手。

而在国外，根据众多因素的影响，国内的微信在国外受众并不大，而国外也有像微信的社交平台比如 Facebook、Twitter 等聊天软件，因此微信在国外的流量并不大，在国外的影响力远不如国内，流量也远不及国内。而国外高校对 ACM 竞赛的宣传更多是通过国外比较流行的平台去宣传，比如上面所说的 Facebook 等社交账号。也是因为国外流量的影响，导致国外的公众号开发者非常的少，技术上也停留在比较前的版本。国外更多的还是基于电脑端网页的开发。

1.3 本课题的研究方法

平台基于 Spring Boot^[7]的 SSM 框架开发, 后台使用 Java 作为开发语言, 前端页面使用目前最流行的三大前端框架之一的 React^[8]进行构建与开发, 使用 Fetch 网络 API 进行数据传递, 前端组件样式选用阿里巴巴官方组件库 Ant Design, 使用 wangEditor 插件作为富文本编辑器。

2 需求分析

2.1 系统需求概述

本微信公众号系统用户分为普通用户, 普通管理员和超级管理员。

公众号的功能需要同时兼顾普通用户和管理员, 主要需求分析如下:

1. 新闻管理: 用于发布关于 ACM 竞赛的一些相关新闻和我校竞赛获奖新闻。
2. 公告管理: 管理员发布一些比赛或者会议通知等, 便于用户能更快更好的收到消息, 管理员还可以导出相关比赛的报名人员名单。
3. 值日管理: 管理员添加实验室的值班安排, 方便实验室人员查看。
4. 友链功能: 提供一些相关的友情链接。
5. 讨论区管理, 普通用户可以在这里发布、查看和评论帖子, 方便大家分享感想和问题, 同时也能让实验室的人员能了解到目前大家的状况, 促进交流。
6. 举报功能: 普通用户可以在微信公众号内举报相关不当的帖子、回复以及用户。
7. 反馈功能: 用户可以将自己的建议或者 bug 反馈给后台。
8. 用户管理功能: 管理员负责管理好公众号的使用人员。

2.2 系统角色分析

系统主要角色分为普通用户, 普通管理员和超级管理员。普通用户可以查看新闻、公告、值日、友链, 可以任意发布、查看和评论帖子, 对帖子进行举报和反馈。普通管理员除了具有普通用户的功能, 还包括审核用户, 而超级管理员除了有普通管理员的权利外还有一个权限, 就是能将用户设置和取消普通管理员身份。

根据用户角色分析, 得出如下角色用例图。普通用户用例图如图 2-1 所示, 普通管理员用例图如图 2-2 所示, 超级管理员用例图如图 2-3 所示。

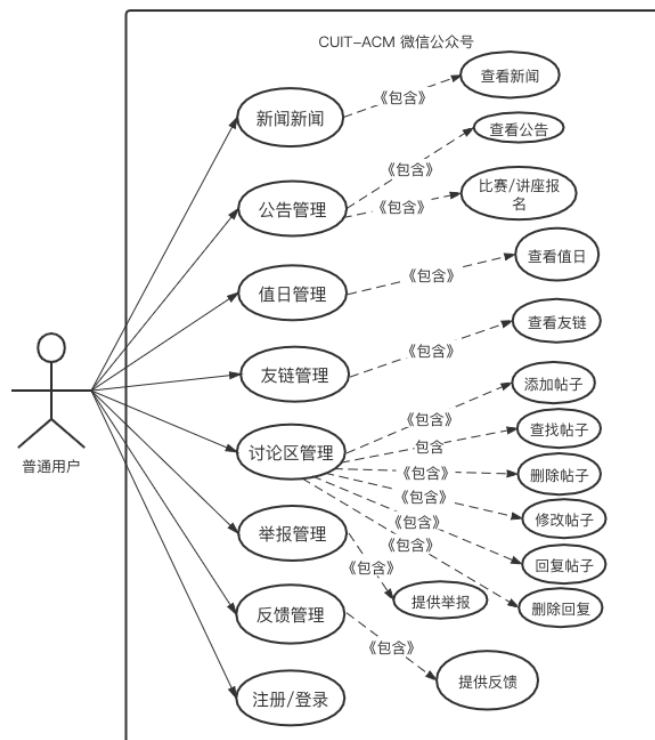


图 2-1 普通用户用例图

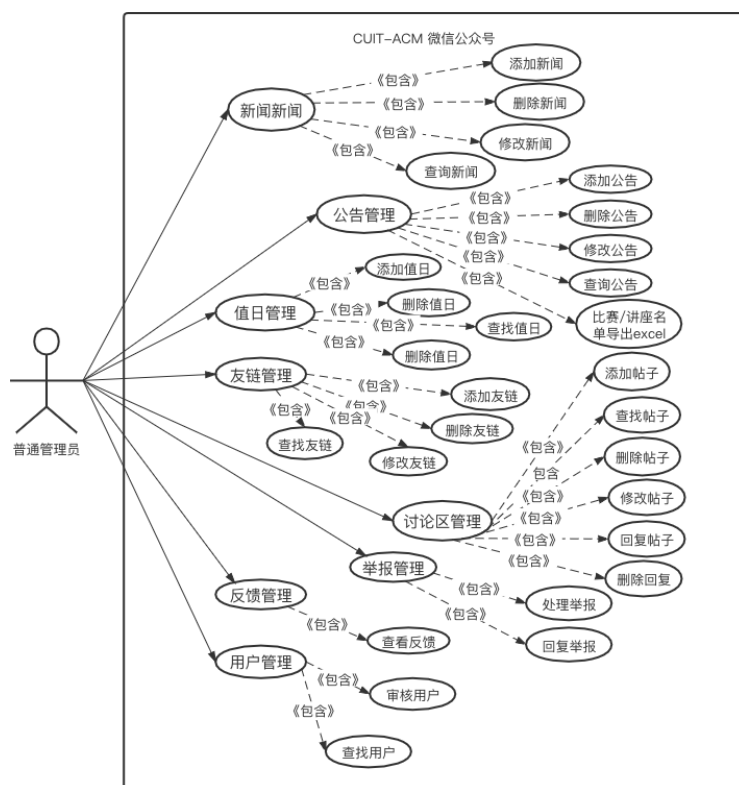


图 2-2 普通管理员用例图

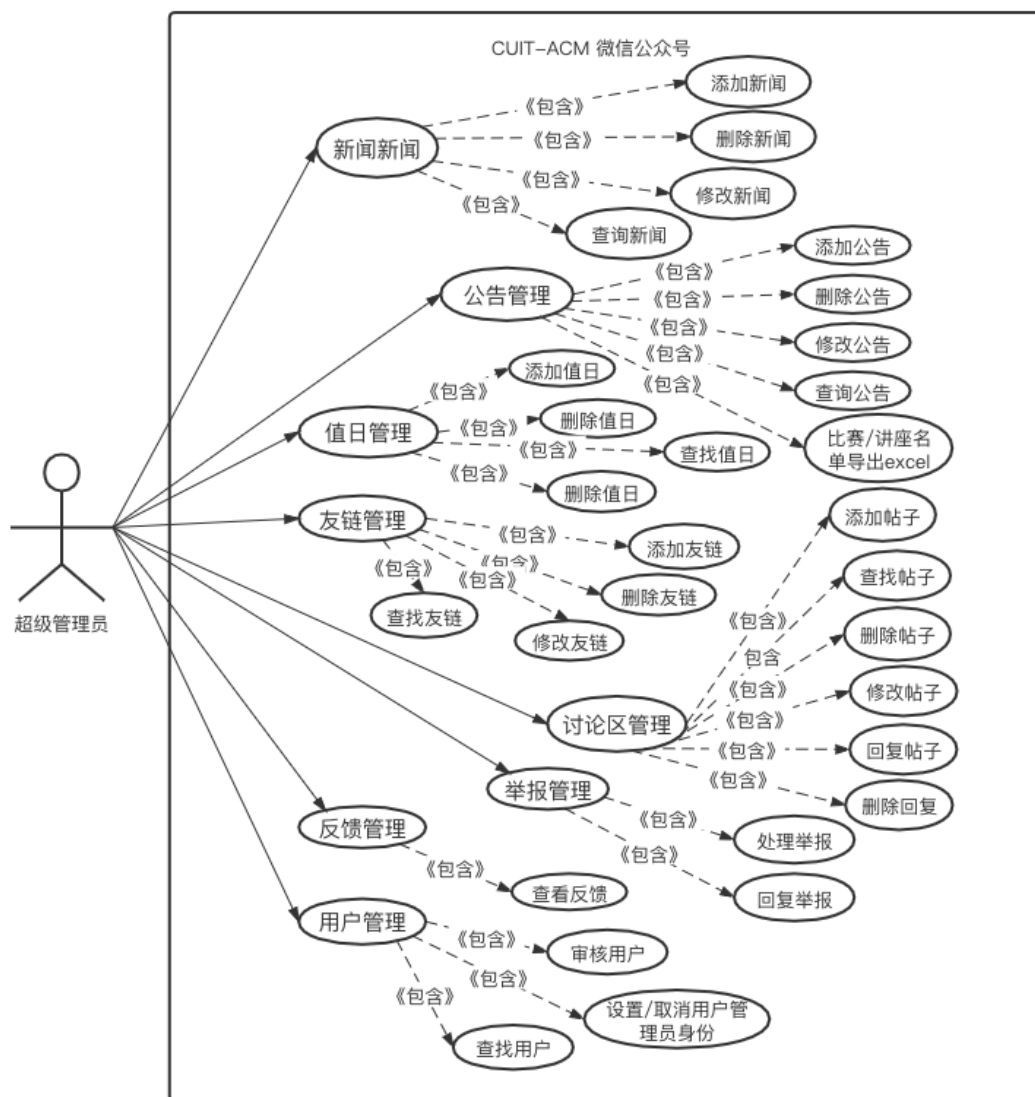


图 2-3 超级管理员用例图

2.3 系统功能性分析

系统主要功能划分包含新闻管理，公告管理，值日管理，友链管理，讨论区管理，举报管理，反馈管理，用户管理。

(1) 新闻管理

普通管理员发布与竞赛相关的新闻，普通用户进行查看。管理员进行管理。

(2) 公告管理

普通管理员发布相关比赛或者会议通知，普通用户进行报名和查看。管理员进行管理。

(3) 值日管理

普通管理员发布和安排实验室的值日，实验室人员可以随时随地的在微信公众号上查看相关值日安排。管理员进行管理。

(4) 友链管理

普通管理员添加一定的友情链接作为学习链接。管理员进行管理。

(5) 讨论区管理

用户可以在讨论区中发布帖子、浏览帖子、评论帖子。普通用户与管理员可对帖子进行管理。

(6) 举报功能

普通用户对讨论区中不当的帖子和评论进行举报。管理员进行管理。

(7) 反馈管理

普通用户对后台提供一定的使用建议和 bug 反馈,方便开发人员更好的改善公众号。

(8) 用户管理

普通用户第一次登录微信公众号需要进行注册,由普通管理员进行审核后方可进入微信公众号使用相关功能,只有超级管理有权利设置和取消用户的管理员身份。

2.4 可行性分析

在着手开发系统前,为了避免资源的浪费,必须先做好全面的可行性分析。而本公众号的可行性分析主要包含以下两个方面:

1.经济可行性

本公众号开发所使用的框架、插件和软件全部为免费或者开源的。整个项目的计划与实现也都由本人独立完成,无额外的开销。平台也只在本地计算机上运行,暂时不需要额外购买服务器搭建平台,综上分析,开发该公众号的成本极低,故经济上可行。

2.技术可行性

后端开发技术采用当前最流行的 SSM 框架,同时使用 Spring 社区很火的 Spring Boot 来更快的搭建后端框架。前端采用目前流行的 3 大框架之一的 React,目前 React 也在不断的更新,同时又有很多适应于 React 的 UI 以及工具。前端部分要分为移动端和后端页面。而这些页面都是会仿照一定的网站和 App 去设计。使用的 Ant Design UI 样式、相关插件以及 React 的生态技术等都是在前期时间中已经学习掌握,故技术方面是可行的。

3 系统总体设计

3.1 后端总体架构

该微信公众号后端总体框架采用的基于 Spring Boot 并整合了 Spring MVC, MyBatis 的 SSM 框架。由 Model 层负责视图的数据传递,再由 Controller 层进行逻辑判断,其中 Controller 层调用 Service 层接口进行业务逻辑处理,最后由

Service 层调用 Mapper 层进行数据持久化处理^[5]。

后端总体架构图如图 3-1 所示：

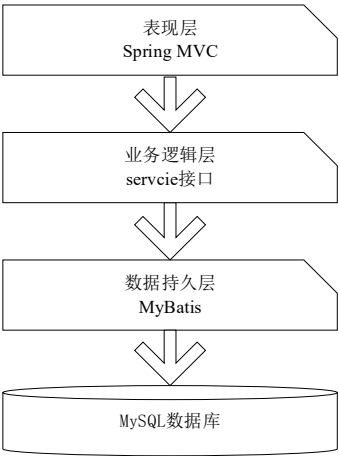


图 3-1 后端总体架构图

3.2 微信公众号开发者模式

微信公众号开发者模式是微信官方推出的为自由开发者提供一种自定义微信公众号的模式，流程为当用户在微信客户端进行相关操作时，微信公众号服务器会将相应的 HTTP^[16]请求发送给第三方服务器，从而自定义操作的处理。开发者可以在开发者服务器上开发相应的功能模块^[6]。该模式的交互关系如图 3-2 所示。



图 3-2 微信公众号开发者模式图

3.3 系统数据库设计

3.3.1 数据库概念设计

根据需求分析，一共设计了 13 个实体^[12]：

（1）用户信息（User）：用户编号、用户名、密码、身份、头像、学号、学号、年级、注册时间、电话、真实姓名、性别、微信用户唯一编号等。

（2）公告（Announcement）：公告编号、公告标题、公告内容、公告类别编号、发布人编号、发布时间、是否注册、注册开始时间、注册结束时间、开始时间、持续时间、是否发布、浏览量等。

（3）公告类别（AnnouncementTag）：公告类别编号、公告类别名称、公告

类别颜色、发布人编号、发布时间、是否需要开始时间等。

(4) 新闻 (News): 新闻编号、新闻标题、新闻内容、新闻类别编号、发布人编号、发布时间、是否发布、浏览量、信息来源等。

(5) 新闻类别 (NewsTag): 新闻类别编号、新闻类别名称、新闻类别颜色、发布人、发布时间等。

(6) 值日管理 (OnDuty): 值日编号、值班人员用户编号、值班人用户名、值班人员电话、值班周期开始时间、值班周期结束时间、发布人、发布时间等。

(7) 友链管理 (FriendUrl): 友链编号、友链名称、友链类别、友链地址、发布人编号、发布时间等

(8) 帖子管理 (Post): 帖子编号、帖子标题、帖子内容、帖子标签、点赞数、浏览量、是否加精、是否加热、是否置顶、发布人、发布时间等。

(9) 讨论帖标签 (Label): 标签编号、标签名称、标签颜色、标签标识、发布人、发布时间等。

(10) 评论 (Comment): 评论编号、评论内容、回复帖子编号、点赞数、评论人、评论时间等。

(11) 人员回复 (ForumTotalReply): 人员回复编号、回复的用户编号、类型、类型对应的编号、回复的内容、发起人、发起时间等。

(12) 反馈 (Feedback): 反馈编号、反馈人编号、反馈内容、反馈时间等。

(13) 举报 (Report): 举报编号、类型、对应类型的编号、举报说明、举报理由、举报人、举报时间、处理人、处理时间、处理结果、是否处理等。

根据实体间的联系，得出本微信公众号的数据库整体 E-R 图如图 3-3 所示。

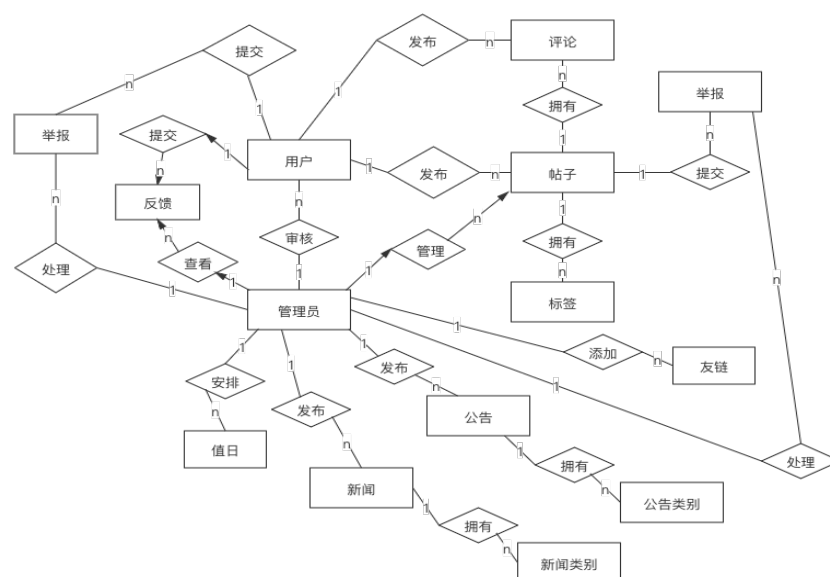


图 3-3 系统整体 E-R 图

3.3.2 数据库物理结构模型

本系统一共设计了 14 张数据库表，数据库物理结构如图 3-4 所示：

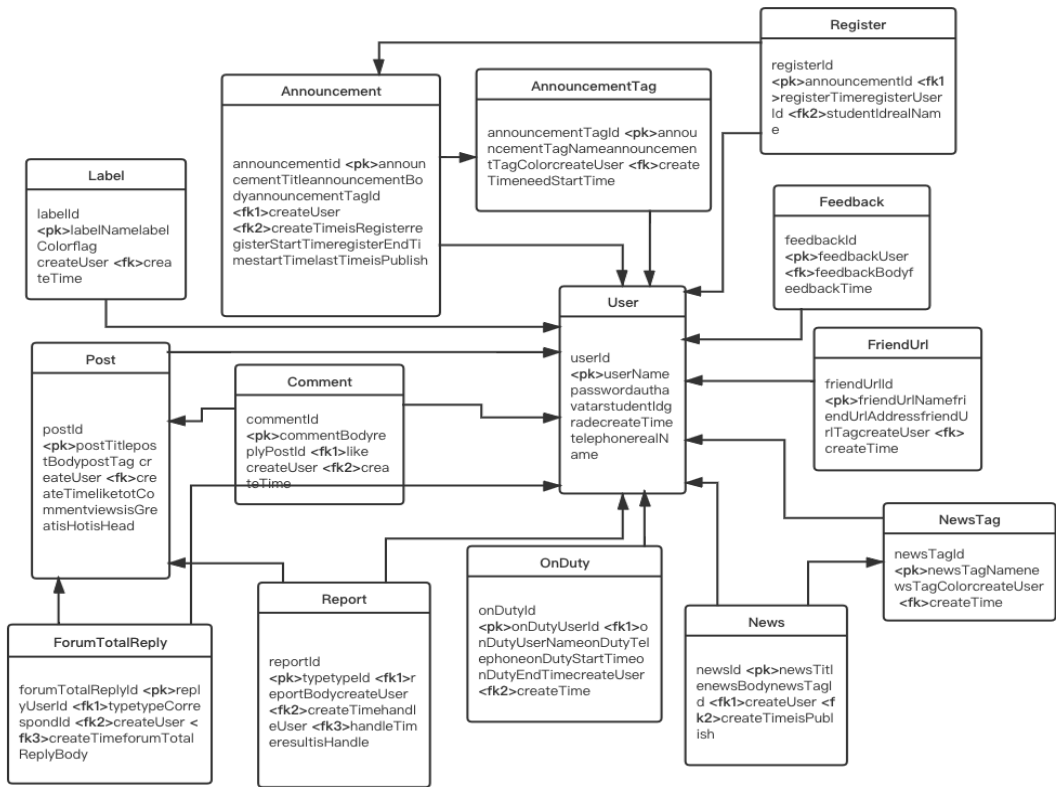


图 3-4 数据库物理结构图

3.3.3 数据库表结构设计

本系统使用 MySQL^[9]持久化数据库，一共有 14 张数据库表。各表信息如下：

(1) 用户信息表 (User)

存放用户信息，如头像、账号密码、姓名等等，用户信息表 (User) 的设计如表 3-1 所示：

表 3-1 用户信息表 (User)

序号	中文名称	列名	数据类型	主键	非空	外键
1	用户 id	USERID	BIGINT	√	√	×
2	用户名	USERNAME	VARCHAR	×	√	×
3	密码	PASSWORD	VARCHAR	×	√	×
4	身份	AUTH	INT	×	√	×
5	头像	AVATAR	VARCHAR	×	√	×
6	学号	STUDENTID	VARCHAR	×	√	×
7	年级	GRADE	INT	×	√	×

续表 3-1

8	注册时间	CREATETIME	DATETIME	×	√	×
9	电话	TELEPHONE	VARCHAR	×	√	×
10	真实姓名	REALNAME	VARCHAR	×	√	×
11	性别	SEX	INT	×	√	×
12	微信用户编号	OPENID	VARCHAR	×	√	×

(2) 公告表 (Announcement)

用于发布相关的公告信息，包括讲座，比赛，会议等类型，公告表 (Announcement) 的设计如表 3-2 所示:

表 3-2 公告表 (Announcement)

序号	中文名称	列名	数据类型	主键	非空	外键
1	公告编号	ANNOUNCEMENTID	BIGINT	√	√	×
2	公告标题	ANNOUNCEMENTTITLE	VARCHAR	×	√	×
3	公告内容	ANNOUNCEMENTBODY	TEXT	×	√	×
4	公告类别编号	ANNOUNCEMENTTAGID	BIGINT	×	√	√
5	发布人编号	CREATEUSER	BIGINT	×	√	√
6	发布时间	CREATETIME	DATETIME	×	√	×
7	是否注册	ISREGISTER	INT	×	√	×
8	注册开始时间	REGISTERTARTTIME	DATETIME	×	×	×
9	注册结束时间	REGISTERENDTIME	DATETIME	×	×	×
10	开始时间	STARTTIME	DATETIME	×	×	×
11	持续时间	LASTTIME	VARCHAR	×	×	×
12	是否发布	ISPUBLISH	INT	×	√	×
13	浏览量	VIEW	INT	×	√	×

(3) 公告类别表 (AnnouncementTag)

用于表示公告的种类，公告类别表 (AnnouncementTag) 的设计如表 3-3 所示:

表 3-3 公告类别表 (AnnouncementTag)

序号	中文名称	列名	数据类型	主键	非空	外键
1	公告类别编号	ANNOUNCEMENTTAGID	BIGINT	√	√	×

续表 3-3

2	公告类别名称	ANNOUNCEMENTTAGNAME	VARCHAR	×	√	×
3	公告类别颜色	ANNOUNCEMENTTAGCOLOR	VARCHAR	×	√	×
4	发布人编号	CREATEUSER	BIGINT	×	√	√
5	发布时间	CREATETIME	DATETIME	×	√	×
6	是否需要开始时间	NEEDSTARTTIME	INT	×		×

(4) 新闻表 (News)

用于发布相关的新闻, 包括一系列的类型 (见 NewsTag 表), 新闻表 (News) 的设计如下表 3-4 所示:

表 3-4 新闻表 (News)

序号	中文名称	列名	数据类型	主键	非空	外键
1	新闻编号	NEWSID	BIGINT	√	√	×
2	新闻标题	NEWTITLE	VARCHAR	×	√	×
3	新闻内容	NEWSBODY	TEXT	×	√	×
4	新闻类别编号	NNEWSTAGID	BIGINT	×	√	√
5	发布人编号	CREATEUSER	BIGINT	×	√	√
6	发布时间	CREATETIME	DATETIME	×	√	×
7	是否发布	ISPUBLISH	INT	×	×	×
8	浏览量	VIEW	INT	×	×	×
9	信息来源	FROMWHERE	VARCHAR	×	×	×

(5) 新闻类别表 (NewsTag)

用于表示新闻的类别, 新闻类别表 (NewsTag) 的设计如表 3-5 所示:

表 3-5 新闻类别表 (NewsTag)

序号	中文名称	列名	数据类型	主键	非空	外键
1	新闻类别编号	NEWSTAGID	BIGINT	√	√	×
2	新闻类别名称	NEWSTAGNAME	VARCHAR	×	√	×
3	新闻类别颜色	NEWSTAGCOLOR	VARCHAR	×	√	×
4	发布人	CREATEUSER	BIGINT	×	√	√
5	发布时间	CREATETIME	DATETIME	×	√	×

(6) 值日表 (OnDuty)

记录了值日的相关信息，包括值日周期、值日人等信息，值日表（OnDuty）的设计如表 3-6 所示：

表 3-6 值日表（OnDuty）

序号	中文名称	列名	数据类型	主键	非空	外键
1	值日编号	ONDUTYID	BIGINT	√	√	×
2	值班人员用户编号	ONDUTYUSERID	BIGINT	×	√	√
3	值班人用户名	ONDUTYUSERNAME	VARCHAR	×	√	×
4	值班人员电话	ONDUTYTELEPHONE	VARCHAR	×	√	×
5	值班周期开始时间	ONDUTYSTARTTIME	DATETIME	×	√	×
6	值班周期结束时间	ONDUTYENDTIME	DATETIME	×	√	×
7	发布人	CREATEUSER	BIGINT	×	√	√
8	发布时间	CREATETIME	DATETIME	×	√	×

（7） 友情链接表（FriendUrl）

用于记录友链的信息，包含名称，地址等信息，友情链接表（FriendUrl）的设计如表 3-7 所示：

表 3-7 友情链接表（FriendUrl）

序号	中文名称	列名	数据类型	主键	非空	外键
1	友链编号	FRIENDURLID	BIGINT	√	√	×
2	友链名称	FRIENDNAME	VARCHAR	×	√	×
3	友链地址	FRIENDADDRESS	VARCHAR	×	√	×
4	友链类别	FRIENDURLTAG	VARCHAR	×	√	×
5	发布人编号	CREATEUSER	BIGINT	×	√	√
6	发布时间	CREATETIME	DATETIME	×	√	×

（8） 帖子表（Post）

主要用于记录讨论区中发布的帖子信息，包含标题，内容，浏览量等信息，帖子表（Post）的设计如表 3-8 所示：

表 3-8 帖子表（Post）

序号	中文名称	列名	数据类型	主键	非空	外键
1	帖子编号	POSTID	BIGINT	√	√	×
2	帖子标题	POSTTITLE	VARCHAR	×	√	×

续表 3-8

3	帖子内容	POSTBODY	TEXT	×	√	×
4	帖子标签	POSTTAG	BIGINT	×	√	×
5	点赞数	LIKE	INT	×	√	×
6	浏览量	VIEWS	INT	×	√	×
7	是否加精	ISGREATE	INT	×	√	×
8	是否加热	ISHOT	INT	×	√	×
9	是否置顶	ISHEAD	INT	×	√	×
10	发布人	CREATEUSER	BIGINT	×	√	√
11	发布时间	CREATETIME	DATETIME	×	√	×

(9) 讨论区标签表 (Label)

主要用于表示帖子的标签信息，讨论区标签表 (Label) 的设计如表 3-9 所示：

表 3-9 讨论区标签表 (Label)

序号	中文名称	列名	数据类型	主键	非空	外键
1	标签编号	LABELID	NUMBER	√	√	×
2	标签名称	LABELNAME	VARCHAR	×	√	×
3	标签颜色	LABELCOLOR	NUMBER	×	√	×
4	标签标识	FLAG	BIGINT	×	√	×
5	发布人	CREATEUSER	BIGINT	×	√	√
6	发布时间	CREATETIME	DATETIME	×	√	×

(10) 评论表 (Comment)

主要用于记录在讨论区中的帖子下面的评论信息，包含内容，点赞数等信息，评论表 (Comment) 的设计如表 3-10 所示：

表 3-10 评论表 (Comment)

序号	中文名称	列名	数据类型	主键	非空	外键
1	评论编号	COMMENTID	BIGINT	√	√	×
2	评论内容	COMMENTBODY	TEXT	×	√	×
3	回复帖子编号	REPLYPOSTID	BIGINT	×	√	√
4	点赞数	LIKE	INT	×	√	×

续表 3-10

5	评论人	CREATEUSER	BIGINT	×	√	×
6	评论时间	CREATETIME	DATETIME	×	√	×

(11) 人员回复表 (ForumTotalReply)

主要用于讨论区模块中用户在帖子之间的回复或者评论之间的回复, 包含回复内容, 回复类型等信息, 人员回复表 (ForumTotalReply) 的设计如表 3-11 所示:

表 3-11 人员回复表 (ForumTotalReply)

序号	中文名称	列名	数据类型	主键	非空	外键
1	人员回复编号	FORUMTOTALREPLYID	BIGINT	√	√	×
2	回复的用户编号	REPLYUSERID	BIGINT	×	√	√
3	类型	TYPE	INT	×	√	×
4	类型对应的编号	TYPECORRESPONDID	BIGINT	×	√	√
5	回复的内容	FORUMTOTALREPLYBODY	TEXT	×	√	×
6	发起人	CREATEUSER	BIGINT	×	√	×
7	发起时间	CREATETIME	DATETIME	×	√	×

(12) 反馈表 (Feedback)

主要用于记录用户提供的反馈信息, 反馈表 (Feedback) 的设计如表 3-12 所示:

表 3-12 反馈表 (Feedback)

序号	中文名称	列名	数据类型	主键	非空	外键
1	反馈编号	FEEDBACKID	BIGINT	√	√	×
2	反馈人编号	FEEDBACKUSER	BIGINT	×	√	√
3	反馈内容	FEEDBACKBODY	TEXT	×	√	×
4	反馈时间	FEEDBACKTIME	DATETIME	×	√	×

(13) 报名信息表 (Register)

主要用于记录报名比赛的参赛人员信息, 方便导出数据等, 报名信息表 (Register) 的设计如表 3-13 所示:

表 3-13 报名信息表 (Register)

序号	中文名称	列名	数据类型	主键	非空	外键
1	报名编号	REGISTERID	BIGINT	√	√	×

续表 3-13

2	对应的公告编号	ANNOUNCEMENTID	BIGINT	×	√	√
3	报名时间	REGISTERTIME	DATETIME	×	√	×
4	报名用户编号	REGISTERUSERID	BIGINT	×	√	√
5	学号	STUDENTID	BIGINT	×	√	×
6	真实姓名	RELANAME	VARCHAR	×	√	×

(14) 举报表 (Report)

主要用于记录用户提供的举报信息以及后台管理员的处理信息等，举报表 (Report) 的设计如表 3-14 所示：

表 3-14 举报表 (Report)

序号	中文名称	列名	数据类型	主键	非空	外键
1	举报编号	REPORTID	BIGINT	√	√	×
2	类型	TYPE	INT	×	√	×
3	对应类型的编号	TYPEID	BIGINT	×	√	√
4	举报说明	REPORTBODY	VARCHAR	×	√	×
5	举报理由	REASON	VARCHAR	×	√	×
6	举报人	CREATEUSER	BIGINT	√	√	√
7	举报时间	CREATETIME	DATETIME	×	√	×
8	处理人	HANDLEUSER	BIGINT	×	×	√
9	处理时间	HANDLETIME	DATETIME	×	×	×
10	处理结果	RESULT	VARCHAR	×	×	×
11	是否处理	ISHANDLE	INT	×	×	×

3.4 系统功能设计

本系统主要包含用户管理、新闻管理、公告管理、讨论区管理、值班管理、友链管理、其余管理七大模块。系统功能结构图如图 3-5 所示。

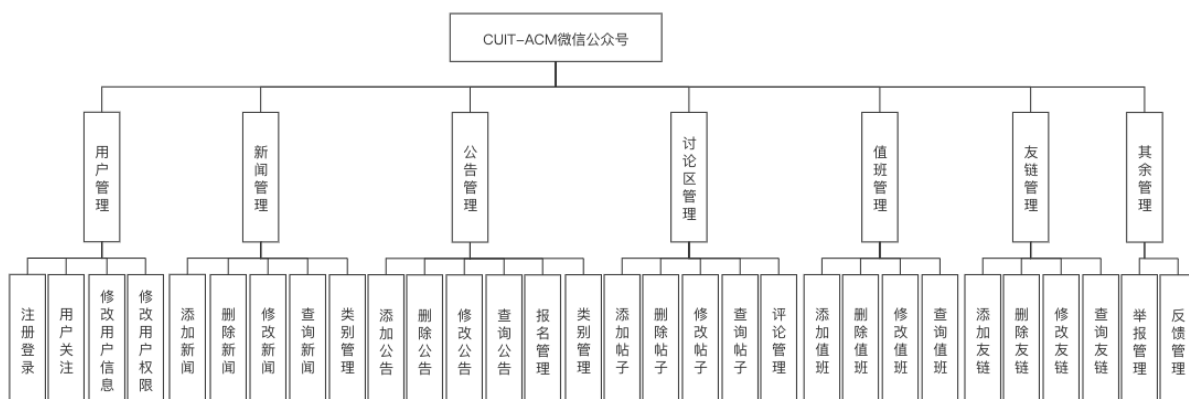


图 3-5 系统总体功能模块图

3.4.1 用户管理模块

(1) 用户注册

用户需要登录进入公众号，若用户是第一次进入公众号则会要求先注册，然后自动登录进公众号。在填写注册要求数据时，若注册信息无误且手机号、用户名及学号格式正确且唯一即注册成功，随后跳转至公众号首页。若用户填写的信息有误或格式不规范，则系统会显示相应的提示信息。用户注册流程图如图 3-所示。

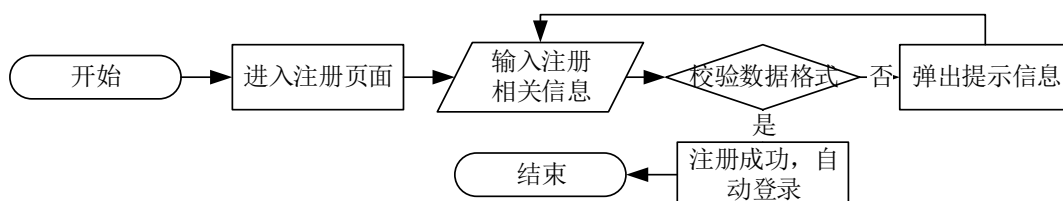


图 3-6 用户注册流程图

(2) 用户登录

用户进入公众号前会根据用户本人的微信号进行自动登录，第一次登录时会先强制要求注册，而后自动登录，若已经是注册过的账号则会直接进入公众号首页。用户登录流程如图 3-7 所示。

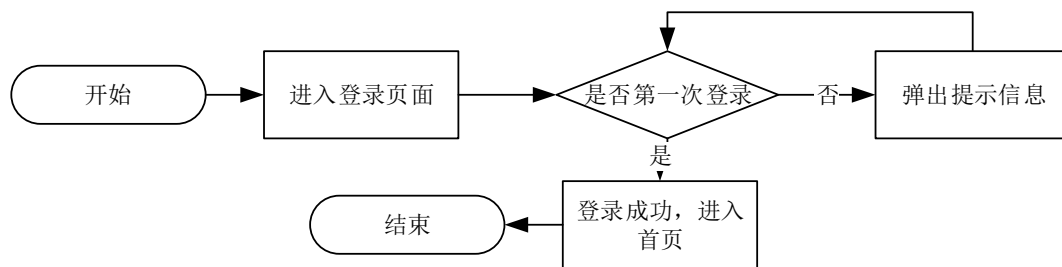


图 3-7 用户登录流程图

(3) 关注用户

用户通过关注一些其它用户，可以获取到关注用户的发帖纪录和回复纪录。

(4) 用户管理

后台管理员对用户信息和权限的管理，具体操作如表 3-15 所示

表 3-15 用户管理说明表

操作	说明
审核用户	对于注册用户，管理员查看用户对应信息是否正确后给出审核结果，正确则通过审核。
删除用户	管理员可以关闭一些多次违规的账号和清除废弃账号。
修改用户信息	用户自身可以修改自己的信息，管理员也可以修改用户的部分信息，如年级等。
权限管理	只有超级管理员才有资格对用户设置或者取消管理员。

3.4.2 新闻管理模块

后台管理员登录到后台系统后，可对新闻模块进行一定的管理。

(1) 添加新闻：管理员发布一些时下与 ACM 竞赛相关的新闻或者是实验室得奖等新闻。

(2) 删除新闻：管理员删除掉错误的新闻或者陈旧新闻。

(3) 修改新闻：管理员修改发布新闻中的不当的地方。

(4) 查询新闻：管理员可通过一定的查询条件查询特定的新闻。

(5) 类别管理：管理员管理好相关的新闻类别，提供不同的颜色去表示不同的新闻类别。

3.4.3 公告管理模块

(1) 添加公告

管理员发布相应的实验室公告，常见的公告类别有比赛、会议、讲座和重要

通知等，其中大部分类别是需要设置一定的属性的，比如是否需要报名，报名周期，开始时间，持续时间等，需要和类别处有一定的联系。添加公告的流程如图 3-8 所示：

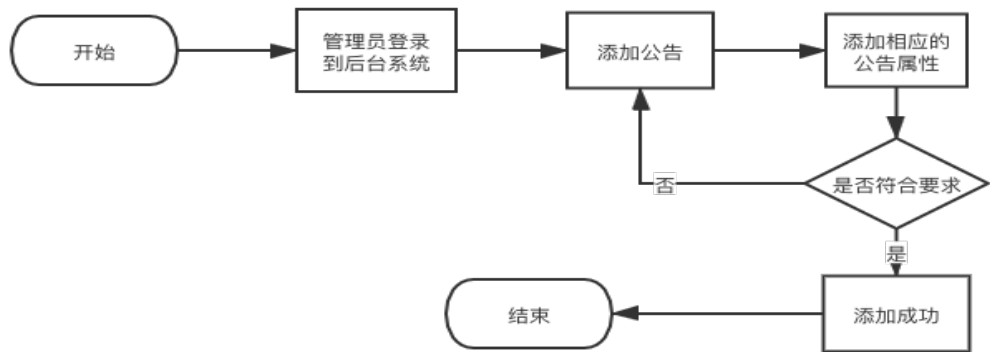


图 3-8 添加公告流程图

- (2) 删除公告：管理员删除掉错误的公告或者陈旧公告。
- (3) 修改公告：管理员修改发布出去的公告中的不当的地方。
- (4) 查询公告：管理员可通过一定的查询条件查询特定的公告。
- (5) 报名管理

因为有可能某次的比赛或者讲座需要打印名单，那么此时用户直接在公众号上报名相应的比赛或者讲座，管理员可在后台中导出相应的名单到 Excel 表中，这样就可以非常简单的得到名单。并且也不会存在重复报名等，在公众号上报名时禁止多次报名的情况。报名的流程如图 3-9 所示：

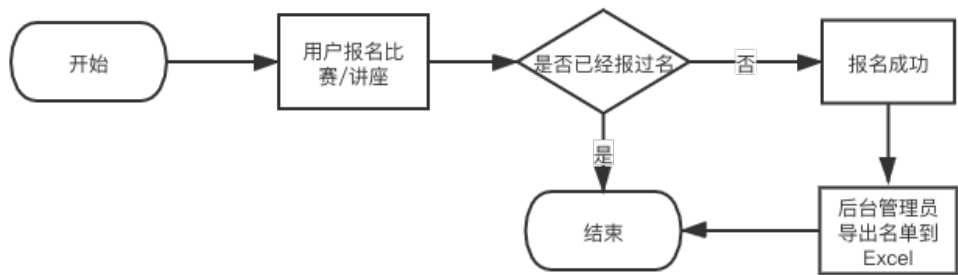


图 3-9 报名流程图

- (6) 类别管理：管理员负责管理好相关的公告类别，提供不同的颜色去表示不同的类别，并且需要确定好该项类别是否需要报名。

3.4.4 讨论区管理模块

(1) 添加帖子：

用户可以在讨论区处自由发布帖子。输入基本的帖子标题和帖子内容，在符合发帖的要求后可成功发出，如果不符合则会给与一定的提示信息。相应的流程图如图 3-10 所示。

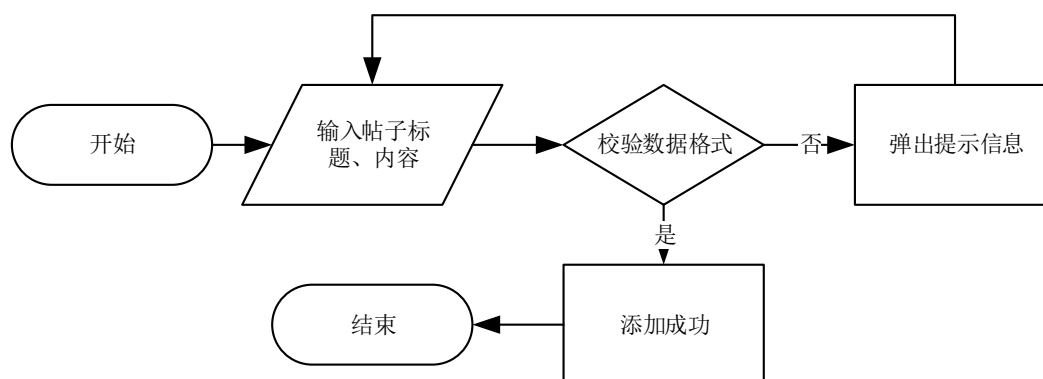


图 3-10 添加帖子流程图

(2) 删除帖子

分为用户和管理员删除帖子。用户可以自由删除自己发布的帖子。管理员则是会将那些被举报并且情况属实的帖子删除掉。

(3) 修改帖子

用户发表帖子后，在觉得发布的帖子中内容有误则可进行修改，同样的会对用户修改后的的帖子信息进行验证，如若符合要求则修改成功，若不符合会给予相应的提示信息。修改帖子的流程图如错误!未找到引用源。。

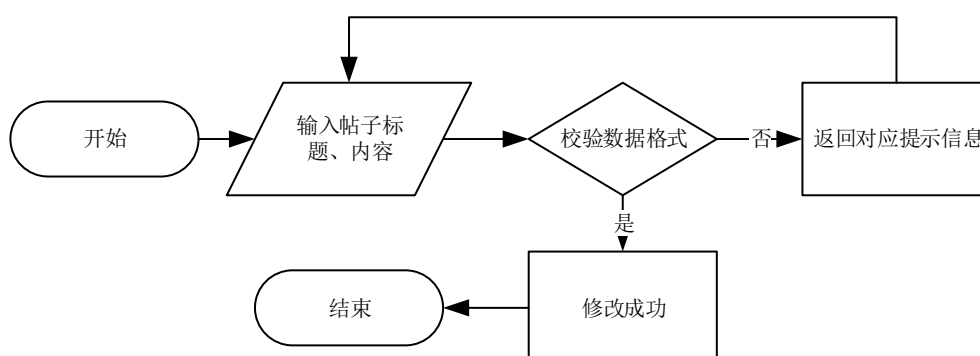


图 3-11 修改帖子流程图

(4) 查询帖子

用户可以查询自己想要的帖子，管理员也可检索帖子信息。

(5) 标签管理

帖子的标签可以有多个，用户也是可以自行修改自己的帖子标签，管理员也可以删除或者修改部分不合理的标签。

(6) 评论管理

用户可自由在任意帖子下面进行评论，也可以继续回复帖子下面的相关评论或者评论的回复。

3.4.5 值班管理模块

后台管理员登录到后台系统后，可对值班模块进行一定的管理。

(1) 添加值班：管理员添加值班信息。

(2) 删除值班：管理员删除掉错误的值班日期安排。

(3) 修改值班：在安排的值日中经常遇到事情冲突，故管理员在后台中修改一些值班的属性即可。

(4) 查询值班：管理员可通过一定的检索条件查询值班情况。

3.4.6 友链模块

后台管理员登录到后台系统后，可对友链模块进行一定的管理。

(1) 添加友链：管理员添加相应的有意思友情链接。后台也会对添加的链接进行链接测试，看是否为一条正确的链接。添加友链的流程图如图 3-12 所示。

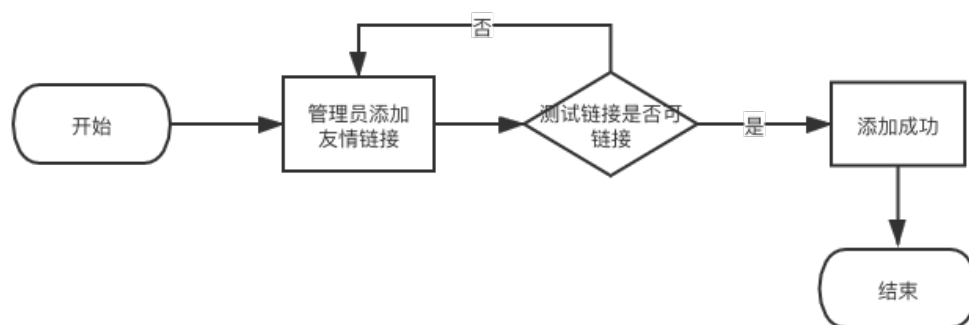


图 3-12 添加友链流程图

(2) 删除友链：管理员删除失效的友链。

(3) 修改友链：管理员修改友链信息。

(4) 查询友链：管理员可查询友链。

3.4.7 其余模块

其余模块是对系统的补充点，包括反馈功能和举报功能，功能介绍如下。

(1) 反馈管理

用户在微信公众号上提供建议或者 bug 反馈，后台管理员看到后联系开发人员，来让这个微信公众号系统做得更好。

(2) 举报管理

用户在讨论区逛帖子时，可能会遇到一些不正当的帖子或者评论、还有就是不正当的用户，此时用户可对相应的帖子、评论和用户进行举报，再提交到后台后，管理员会对这些举报进行验证如果情况属实，那么会由管理员将向相应的帖子和评论删除掉，将相关用户进行拉黑处理等，最后将结果返回给提供举报的用户。流程图如 3-13 所示。

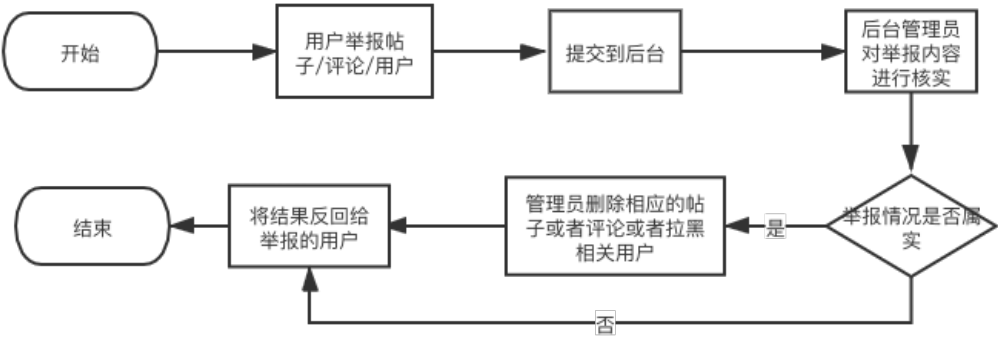


图 3-13 举报管理流程图

3.5 系统技术难点及关键技术

本系统的关键技术难点在于讨论区的评论展示和消息通知两个方面。

评论展示：在公众号的讨论区中帖子是具有评论功能，那么好的评论系统展示是非常重要的，同样为了做好体验，在后台 PC 端和公众号手机端上都想要好的评论展示效果，在网上搜索了一些效果后，最终决定只嵌套一层的方式展示评论，即帖子下面的一层评论，然后评论下面的回复是平级关系，不再嵌套，并且实现了一些动画效果利用 CSS3,而这些评论样式也都是模仿多种风格做的，多开了几张表去实现的这个功能。另外在消息通知功能中，如何在别人评论了你的帖子或者回复了你的评论后得到通知呢？因为是公众号，不像 App 有消息机制，故最后也是采用的数据库表的形式，每当有相应的回复，会在数据库表中插入一条相应回复数据，前端在刷新页面时就会重新取数据，此时有人回复了那么就会有相应通知出现。

4 系统实现

4.1 系统开发环境

本平台在 macOS Catalina10.15 开发环境下进行，编辑器使用了当下最流行

的 Java 编辑器之一的 IntelliJ IDEA 2019。前端代码则是用的 Visual Studio Code 编辑器编写。这两款编辑器简单好用，许多公司也都以这两款开发软件作为公司相应的前后端中的第一编辑器。

为了快速的进行项目搭建，公众号使用了 Spring Boot 框架，Spring Boot 是一款十分优秀的可以让开发人员快速构建项目的开源框架，该框架省去了 Spring、Spring MVC 框架所需的复杂配置，仅需少量的配置文件就能搞定。同时，Spring Boot 集成了 Tomcat 与 MyBatis 持久层框架，所以开发人员只需配置数据库信息就能立即进行开发调试。同时，还使用了 Maven 对项目中繁多的 jar 包进行管理，Maven 的使用大大提高了构建项目的效率，也更加利于开发人员对项目中 jar 包的使用。

而选择前端 React 框架之一的原因就是它是热更新的，也就是每次有改变，网页都会立即重新刷新，不用重启服务器之类的，快速看到修改后的页面效果，节省了大量时间。

系统总体开发环境清单如下表 4-1 所示：

表 4-1 系统开发环境清单

硬件环境	处理器	3.1 GHZ 双核 Intel Core i5
	内存	8GB
	硬盘	512GB
	屏幕分辨率	2560x1600
软件环境	操作系统	macOS Catalina 10.15
	浏览器	Chrome ,微信浏览器
	开发工具	IntelliJ IDEA 2019, Visual Studio Code
	项目管理工具	Maven
	服务器	Tomcat
	数据库	MySQL,Redis
	JDK 版本	1.8

4.2 用户模块的实现

4.2.1 用户注册

用户在首次登录公众号时，会被强制性注册，在用户填写完需要的数据并通过校验，此时通过异步请求访问后端接口，进行数据上的校验，防止某些信息重复。然后该注册信息就会存在数据库中，待后台管理员通过后即可进行登录。

4.2.2 用户登录

用户的权限不同，登录的方式和界面也有所不同。

（1）普通用户登录：在微信方点击进入公众号时，微信服务器会传递相应

登录用户的 flag 值，后端服务器再接收到这个 flag 值后，再在微信服务器中获取相应用户的 openId 值，最后再查看数据库中是否有对应的数据。

(2) 后台管理员登录：在 PC 端网页处进行管理员后台页面。在登录页面输入相应的用户名和密码后，调用后端 login 接口，将输入的用户名(userName)和密码(password)传递给后端，后端会相应的查询是否正确，正确则成功登录到后台页面，否则给出错误提示。后台管理员登录的时序图如图 4-1 所示。

在用户登录后，会将用户对应的登录信息存储在 Redis 中，会将对应的生成的 token 值返回给前端，而前端的每次请求都会在 HTTP 头部中放入 token 值，后端则可以通过这个 token 值判断用户当前的登录用户。

通过 token 值获取登录用户信息的代码如下：

```
/**
 * 获取当前的操作用户信息
 * @param request
 */
public User getUserIdFromSession(HttpServletRequest request) {
    if (request.getHeader(SysConst.TOKEN) == null) {
        return null;
    }
    if (redisComponent.get(request.getHeader(SysConst.TOKEN)) == null) {
        return null;
    }
    int userId = Integer.parseInt(redisComponent.get(request.getHeader(SysConst.TOKEN)));
    Map<String, Object> map = new HashMap<>();
    map.put("userId", userId);
    map.put("isEffective", SysConst.USE);
    List<User> user = userService.findUserListByQueryMap(map);
    if (user == null) {
        return null;
    }
    return user.get(0);
}
```

4.2.3 关注用户

公众号用户可以在点击其它用户的头像进行关注用户，进入其它用户界面时就会判断是否已经关注过该用户，或者是此用户能否被关注，在符合关注的条件后可以成功关注相应用户。对应的关注列表是存在 Redis 中的，因为是个一对多的关系，Redis 的集合类型可以很方便的处理。

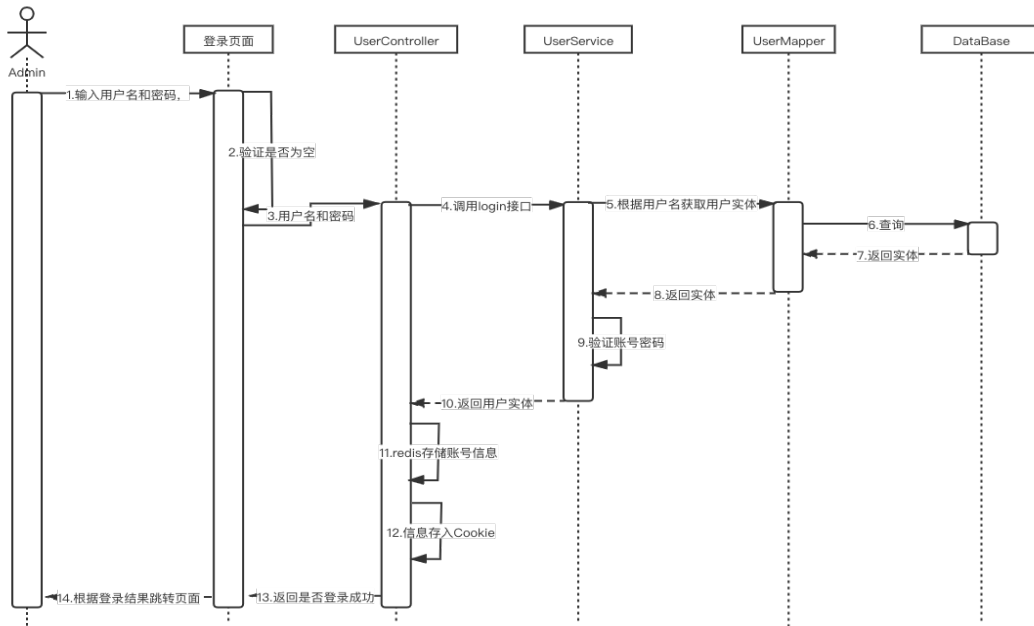


图 4-1 后台管理员登录的时序图

4.2.4 用户管理

(1) 用户审核

在用户注册提交信息到后台，管理员会对提交的注册信息进行检查，比如年级和学号等是否正确，如果正确，即可审核成功。

(2) 权限管理

需要登录唯一的超级管理员账号在对应的用户后面设置相应的权限即可，后台会修改该用户的身份信息。

(3) 删除用户

管理员将删除不必要的用户，点击删除按钮后，后端执行 `deleteUser` 接口，然后就会将用户的状态设置为黑户。并且不能直接拉黑管理员身份，需要先修改权限信息后才能拉黑。效果图如图 4-2 所示。

用户管理						
用户名	真实姓名	学号				
用户名	学号	年级	联系电话	真实姓名	注册时间	操作
xixi	2019011198	2019	18299830012	路人	2020-05-31	取消拉黑
陶博	2017091116	2017	134803128901	陶博	2020-04-09	拉黑 取消管理员 修改
杨浩	2017091115	2017	15689102651	杨浩	2020-04-09	拉黑 设为管理员 修改

图 4-2 用户管理效果图

(4) 修改用户信息：管理员可以在后台修改部分用户数据，用户自身也可

在公众号端自行修改数据，公众端的效果如图 4-3 所示。



图 4-3 公众号端修改效果图

4.3 新闻管理模块的实现

（1）添加新闻：管理员点进后端添加新闻页面，在输入相应的新闻标题和新闻内容后，前端会检验标题长度，再符合要求后调用后端 `addNews` 结构，将新闻信息存入到后端数据库中。

（2）删除新闻：管理员会删除掉错误的新闻或者陈旧新闻，后端调用 `deleteNews` 接口，在确认新闻存在后，改变该新闻的存在状态。

（3）修改新闻：在新闻的列表界面处，点击修改按钮，后端调用 `updateNews` 接口，前端在检验好修改后标题长度后，修改完对应的新闻信息，存入到后台数据库中。

（4）查询新闻：管理员在新闻列表页面可通过一定的查询条件查询特定的新闻。

（5）类别管理：管理员管理好相关的新闻类别，对不同的新闻类别分别用不同的颜色去表示，同时如果要删除一个类别，必须保证没有此类别的新闻存在了。类别的界面如图 4-4 所示。

公告类别

类别名称	添加	类别名称	色值	
类别名称	类别颜色	最近一次更新时间	更新人	操作
测试专用		2020-03-15 21:16:48	谢仁义	修改 删除
学校相关		2020-03-15 21:34:10	谢仁义	修改 删除
时事新闻		2020-01-12 07:15:28	超级管理员	修改 删除
获奖		2019-12-26 03:25:02	超级管理员	修改 删除
竞赛		2020-12-11 22:15:28	谢仁义	修改 删除

图 4-4 新闻类别效果图

4.4 公告管理模块的实现

4.4.1 添加公告

管理员进入到后端系统中，进入到添加公告页面，输入对应的公告标题和内容，然后设置公告的属性，比如是否需要报名，需要的话报名周期是多少，多久开始等属性，在设置完这些后，前端会校验是否符合发布要求，后端调用 `addAnnouncement` 接口存储到后端数据库中。

4.4.2 删除公告

在公告的展示列表中，管理员可删除不需要的公告，后端调用 `deleteAnnouncement` 接口，在校验公告存在后便可成功删除。

4.4.3 修改公告

对于发布有误的公告，管理员可在选怎对应的公告后，进行修改，在输入完修改后的公告标题和公告内容后，前端校验长度是否符合要求，符合后发送给后台，后台调用 `updateAnnouncement` 接口更新后台数据库从而更新公告。

4.4.4 报名管理

在需要报名的公告中，设置好报名周期后，公众号端用户点击进入相应的公告点击报名，在后台管理员可以得知报名的详细情况，待报名结束后可以在前端利用好 `React` 的插件执行导出操作，前端便可直接导出到本地 `Excel` 中。公众号端报名界面如图 4-5 所示。后台管理界面如图 4-6 所示。导出到 `Excel` 的效果如图 4-7 所示。



图 4-5 公告端报名效果图

学号

Q

导出报名表

学号	姓名	操作
2016051227	林胡龙	修改 删除
2017081175	闫国龙	修改 删除
2016081187	谢仁义	修改 删除
2017091116	陶博	修改 删除

图 4-6 报名后台管理效果图

	A	B	C	D
1	学号	姓名		
2	2016051227	林胡龙		
3	2017081175	闫国龙		
4	2016081187	谢仁义		
5	2017091116	陶博		
6	2017091115	杨浩		
7	2017091114	彭峰云		
8	2017091113	左一凡		
9				
10				

图 4-7 导出到 Excel 的效果图

4.4.5 类别管理

这个是需要设置公告的属性，一样的指定颜色区分类别，然后需要再新增公告类别时就要指定好改类别在被选择后是否需要报名。后端调用相应的接口去处理。

4.5 讨论区管理模块的实现

4.5.1 添加帖子

用户在公众号端的讨论区模块，点击添加帖子，在输入完帖子标题和帖子内容后点击下一步，会弹出标签选择框，在选择完标签后，前端会校验帖子的标题是否过长或者为空，标签是否为空等，在通过校验后将数据传递给后端，后端调用 `addPost` 接口将数据存入到数据库中，发表帖子成功后，会回到帖子列表页面，添加帖子的界面如图 4-8 所示。



图 4-8 公众号端发表帖子效果图

4.5.2 删除帖子

(1) 用户删除

用户可以删除自己发布的帖子，在帖子详情页面便可点击删除，在确定好删

除后，便可成功删除。

（2）管理员删除

对于用户举报过的帖子，管理员在核实后便会在后台删除相应的帖子。

上面两种删除都是调用的后端 `deletePost` 接口，后端会校验帖子是否存在，通过校验后便会将帖子的状态设置为死亡，即删除帖子。

4.5.3 修改帖子

用户可以修改自己的发布的帖子，对应的变动帖子标题、内容和帖子标签后，前端会校验，通过校验后将数据发送给后端，后端调用 `updatePost` 接口修改数据中的帖子的数据。

4.5.4 标签管理

标签为帖子的标签，这里一般的帖子都是具有多种标签的，所以处理方式上比新闻和公告麻烦点。这里是利用了二进制的思想去处理多标签的情况，这样二进制的位数不能太多，所以每次新增标签时，需要取到最小未使用的数字作为新标签的标识。相关的代码段如下。

```
// 获取全部有效标签
List<Label> list = labelService.findLabelListByLabelId(null);
Collections.sort(list, new Comparator<Label>() {
    @Override
    public int compare(Label a, Label b) {
        Long ans = a.getFlag() - b.getFlag();
        return ans.intValue();
    }
}); //从小到大排序
long flag = list.size();
for (int i = 0 ; i < list.size() ; ++ i) {
    if (list.get(i).getFlag() != i) {
        flag = i;
        break;
    }
} // 取最小未使用的标志位
```

4.5.5 评论管理

（1）评论帖子：公众号用户在评论区对任意帖子可进行评论。

（2）回复评论：用户在帖子下面的评论中还可以相互回复。

这两种情况调用同一个后端接口，不同的是传递参数，参数不同表示的情况也不同，从而将对应的数据存入数据库中。评论管理的时序图如图 4-9 所示。

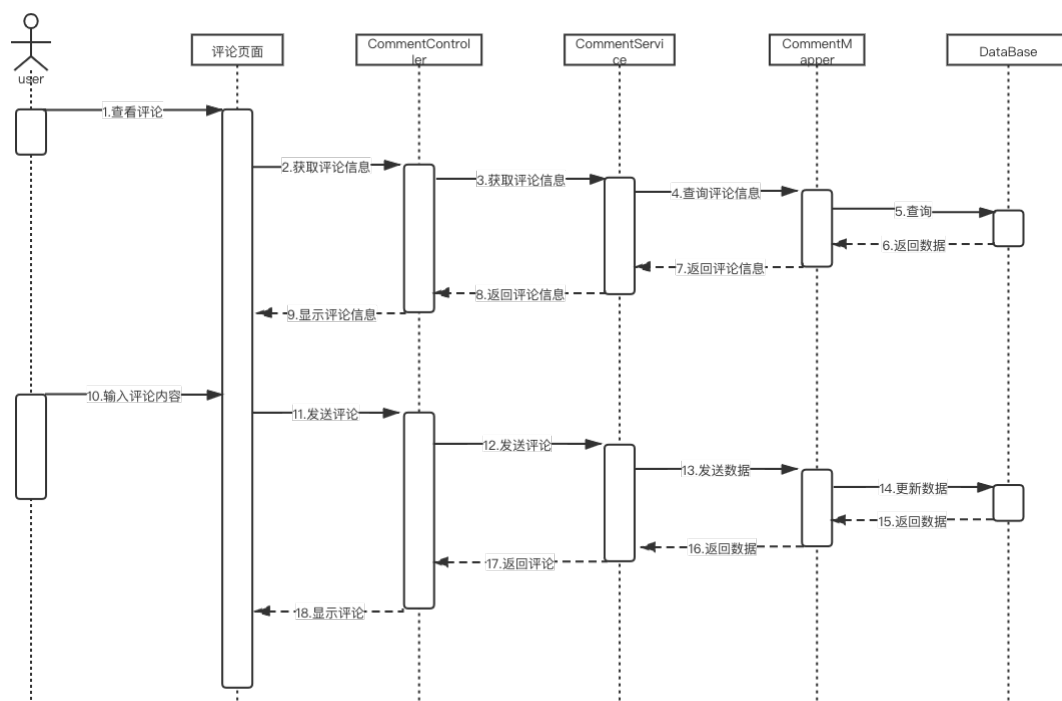


图 4-9 评论管理的时序图

4.6 值班管理模块的实现

(1) 添加值班：管理员点进后端添加值班页面，安排对应的值班人员和值班周期，后端调用 `addOnDuty` 接口处理加入到数据库中。

(2) 删除值班：管理员会删除错误的值班信息情况。

(3) 修改值班：管理员在后台页面修改相应的值班信息，比如值班周期或者是值班人员的变动，后台调用 `updateOnDuty` 处理。

(4) 查询值班：管理员通过一定的筛选条件可查询值班信息，公众号端用户自身也可以查看自己的值班情况，后台调用 `selectOnDuty` 处理。查询值班的时序图如图 4-10 所示。

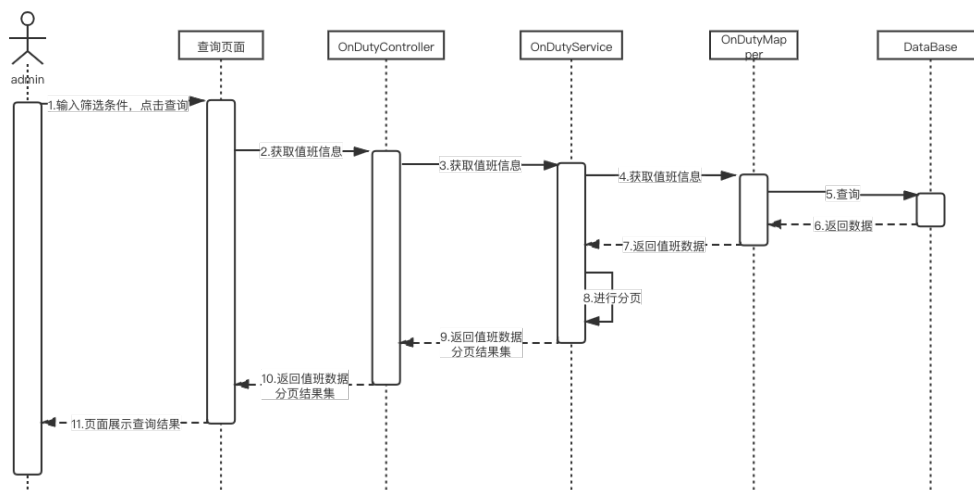


图 4-10 查询值班的时序图

4.7 友链管理模块的实现

(1) 添加友链：管理员进入友链页面，点击添加友链按钮，会弹出一个输入框要求输入对应的友链信息，在输入后，前端会将数据传递给后端处理，后端会先校验该链接是否正确，即能否正确链接，在可以通过校验后，调用 `addFriendUrl` 接口添加到数据中去。添加友链的时序图如图 4-11 所示。添加友链的界面如图 4-12 所示。

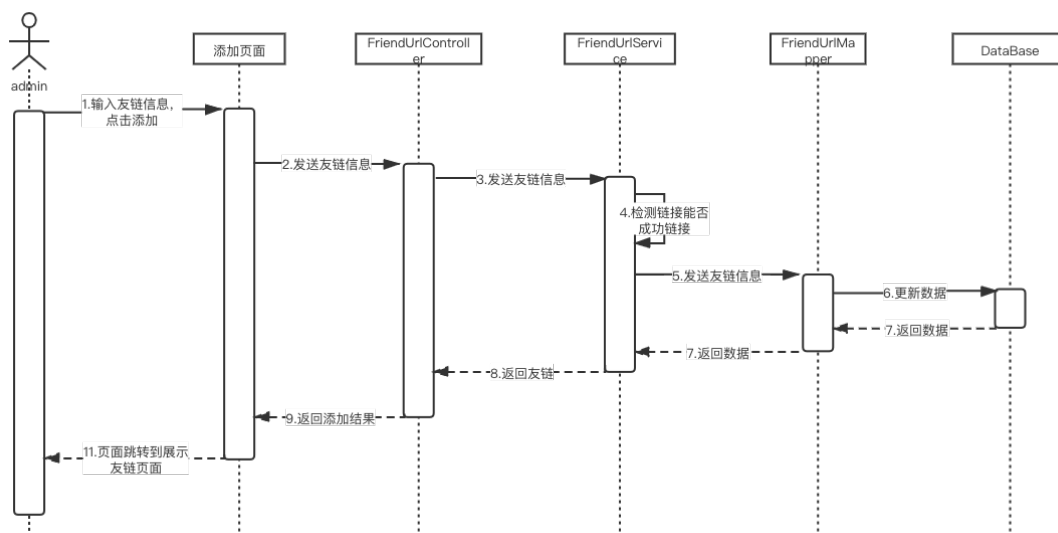


图 4-11 添加友链的时序图

后端校验链接是否可用的代码段如下：

```

/**
 * 检查 URL 是否可访问
 * @param url          访问 URL
 * @param timeOutMilliseconds 超时时间
 * @param checkType     检查类型 https|http|http or https 默认传 2
 * @return
 */
public static boolean isAvailable(String url, int timeOutMilliseconds, int checkType) {
    long lo = System.currentTimeMillis();
    boolean flag = false;
    if(url.indexOf(URL_PREFIX_SECURITY)==0||
url.indexOf(URL_PREFIX_INSECURITY) == 0) {
        flag = isAvailable(url, timeOutMilliseconds); // 有连接协议的
    } else {
        if (checkType == 0) {
            //加 https 是否可用
            flag = isAvailable(url, timeOutMilliseconds, true);
        } else if (checkType == 1) {
            //加 http 是否可用
            flag = isAvailable(url, timeOutMilliseconds, false);
        } else if (checkType == 2) {
            //加 http 或 https 是否可用
            boolean availableSecurity = isAvailable(url, timeOutMilliseconds, true);
            boolean availableInSecurity = isAvailable(url, timeOutMilliseconds, false);
            if (availableSecurity || availableInSecurity) {
                flag = true;
            } else {
                flag = false;
            }
        }
    }
    return flag;
}

```

(2) 删除友链：管理员会删除掉部分陈旧的友链，后端调用 `deleteFriendUrl` 接口，在确认友链存在后，从数据中删除掉对应友链。

(3) 修改友链：管理员可修改部分改变了的友链，包括地址和名称，后端再重新校验链接可用后调用后端接口 `updateFriendUrl` 修改数据库中的友链。

(4) 查询友链：管理员在友链界面，通过部分条件筛选查询对应的友链。



图 4-12 添加友链的界面效果图

4.8 其余管理模块的实现

4.8.1 反馈管理

用户在公众号端的个人页面中，提供反馈建议或者 bug，后端管理员查看相应的反馈即可。后端调用的是 `addFeedback` 接口处理。管理员则可以在后端反馈页面看到相应的用户提供的反馈。

4.8.2 举报管理

(1) 举报帖子或者回复

用户在讨论区处看帖子，如果遇到不正当的帖子或者回复，可以在右上角进行举报，举报后，后端调用 `addReport` 接口处理，后台管理员可以后台系统中看到相应的举报，在核实后会对相应的帖子或者回复进行处理。

(2) 举报用户

公众号端用户可以举报其它用户，同样的接口处理，同样的合适情况后会对相关用户进行处理。

以上两种情况的举报，都会记录在用户的举报记录中，用户可以在个人页面中看到自己提供的举报，同时也可以查看举报的处理结果。

公众号端举报帖子或者回复的效果图如图 4-13 所示。

<举报帖子(回复)

举报类型

广告及垃圾信息

抄袭或涉嫌侵权

不友善/色情/低俗内容

政治敏感

其他

举报说明

请输入内容

提交

图 4-13 公众号端举报帖子或者回复页面效果图

5 系统测试

在代码完成后对系统需要进行了一系列测试，需要对各个功能模块进行相应的测试，从而找出功能是否正常完成，以保证能够在系统上线运行时的稳定性。本次测试主要使用黑盒测试^[13]方法对各个功能模块进行功能验证。

5.1 测试环境

本系统的部署及测试环境如表 5-1 所示。

表 5-1 测试环境

硬件	处理器	3.1 GHZ 双核 Intel Core i5
	内存	8GB
	硬盘	512GB
	屏幕分辨率	2560x1600
软件	操作系统	macOS Catalina 10.15
	浏览器	Chrome ,微信浏览器
	服务器	Tomcat
	数据库	MySQL，Redis

5.2 测试用例及过程

5.2.1 用户管理模块测试

用户管理测试内容如表 5-2 所示。

表 5-2 用户管理测试内容

测试用例编号	Test001	测试用例名称	用户管理模块
测试描述	测试用户的登录、注册和关注等功能		
前驱条件	无		
测试步骤	1.注册用户，输入注册要求信息后等待结果 2.用户登录微信公众号以及管理员登录后台系统 3.关注或者取关另一个用户		
期待输出结果	1.用户注册成功 2.用户成功登录公众号和管理登录后台系统 3.关注或者取关用户成功		
测试结果	通过		

5.2.2 公告管理模块测试

公告管理测试内容如表 5-3 所示。

表 5-3 公告测试内容

测试用例编号	Test002	测试用例名称	公告管理模块
测试描述	测试公告的增删改查以及报名的管理		
前驱条件	账号已登录		
测试步骤	1.添加相应的公告，设置属性 2.公众号端点击报名 3.后台导出名单 4.修改和删除公告 5.查询公告		
期待输出结果	1.添加公告成功 2.公众号用户可成功报名，后台有相应的记录 3.成功导出名单 4.修改和删除公告成功 5.查询出正确公告信息		
测试结果	通过		

5.2.3 讨论区管理模块测试

讨论区管理测试内容如表 5-4 所示。

表 5-4 讨论区管理测试内容

测试用例编号	Test003	测试用例名称	讨论区管理模块
测试描述	验证帖子的增加、删除、修改、查询和评论的回复与显示		
前驱条件	账号已登录		
测试步骤	1.添加帖子，查看是否成功添加 2.修改帖子，查看是否成功修改 3.查询相应的帖子，查看是否成功查询并显示 4.在帖子下面评论，查看是否成功评论 5.回复帖子下面的评论，查看是否回复成功 6.删除帖子，查看是否成功删除		
期待输出结果	1.添加帖子成功 2.修改帖子成功 3.正确查询出帖子并显示 4.评论成功 5.回复成功 6.删除帖子成功		
测试结果	通过		

5.2.4 值班管理模块测试

值班管理测试内容如表 5-5 所示。

表 5-5 值班管理测试内容

测试用例编号	Test004	测试用例名称	值班管理模块
测试描述	测试值班情况的添加、修改、查询		
前驱条件	账号已登录		
测试步骤	1.添加值班情况，查看是否成功 2.修改值班情况，查看是否成功 3.删除值班情况，查看是否成功		
期待输出结果	1.值班添加成功 2.值班修改成功 3.值班删除成功		
测试结果	通过		

5.2.5 友链管理模块测试

友链管理测试内容如表 5-6 所示。

表 5-6 友链管理测试内容

测试用例编号	Test005	测试用例名称	友链管理模块
测试描述	测试友链的添加、修改、删除		
前驱条件	账号已登录		
测试步骤	1.添加友链，查看是否成功 2.修改友链，查看是否成功 3.删除友链，查看是否成功		
期待输出结果	1.友链添加成功 2.友链修改成功 3.友链删除成功		
测试结果	通过		

5.2.6 举报管理模块测试

举报管理测试内容如表 5-7 所示。

表 5-7 文章发布测试内容

测试用例编号	Test006	测试用例名称	举报管理模块
测试描述	测试是否能举报帖子、回复和用户以及后端处理		
前驱条件	账号已登录		
测试步骤	1.举报帖子，查看是否举报成功 2.举报回复，查看是否举报成功 3.举报用户，查看是否举报成功 4.在后台处理举报		
期待输出结果	1.举报帖子成功 2.举报回复成功 3.举报帖子成功 4.后台处理成功，并且成功将结果返回给公众号端		
测试结果	通过		

5.3 测试结果分析

经测试，各个功能模块的功能正常运行，前端页面在测试的不同浏览器下的适配效果不错，展示效果基本一致，符合用户的基本习惯要求。综上所述，本系统功能符合要求。

结 束 语

本系统主要是根据我校 ACM 实验室目前一些比较欠缺并且需要的点出发的，收集了大部分实验室人员和学生们的要求。基本实现了新闻、公告、报名、讨论区、值班等大部分功能。为我校广大对 ACM 感兴趣的学生提供了一个交流平台，同时也方便了实验室这边更好的管理。

在开发中，为了贴合目前业界的开发方式，主动选择了前后端分离开发，降低模块之间的耦合，选择了目前主流的前端框架和后端框架以及技术。

通过本次对系统的设计与实现，不仅增强了自己对现在技术的掌握，更是对一个完成工程的流程和文档有了很清晰的认识。最重要的是个人独立开发时遇到的问题都有了一定的解决方法去搞定，为我以后在公司中遇到问题打下了基础，同时更重要的是学习了目前市场上主流的技术，也可以让我在未来更好的去找和适应工作。

参考文献

- [1] 百度百科. 微信公众号. [EB/OL]. (2012-12-26)[2020-03-10].
<https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1%E5%85%AC%E4%BC%97%E5%8F%B7/4916400>
- [2] 孙大烈, 车万翔. ACM 竞赛与学生综合实践能力培养[J]. 计算机教育, 2009(19):14-16.
- [3] 张蓓, 窦天芳, 张成昱, 等. 开发模式下图书馆微信公众平台服务的设计与实现[J]. 现代图书情报技术(01):91-95.
- [4] 周功建. 微信公众号订餐平台系统设计与实现[J]. 软件导刊, 015(8):74-76.
- [5] 周歧波, 崔忠伟. 面向校园宣传的微信公众平台设计与实现[J]. 电子技术与软件工程, 000(011):P.65.
- [6] 黄楚新, 王丹. 微信公众号的现状、类型及发展趋势[J]. 新闻与写作, 2015, 000(007):4-9.
- [7] 王永和, 张劲松, 邓安明, 等. Spring Boot 研究和应用[J]. 信息通信, 2016(10):91-94.
- [8] 张丰麒, 王飞. ReactJS 的新特性在 Web 开发中的应用[J]. 移动信息, 2015(10):00071-00072.
- [9] 殷丽, 徐海华, 吴海涛. MySQL 查询优化技术——索引[C]// 工业仪表与自动化学术会议. 2007.
- [10] 洪珂, 柳阳彬, 陈影旺, 等. 一种 Redis Key 管理方法及系统.
- [11] 毛炎, 任福, 王功存, 等. 基于新型 Web 脚本样式框架构建城市规划编制信息平台——以 ReactJS 和 Ant Design 为例[J]. 测绘与空间地理信息(8).
- [12] 徐枫. 浅析论坛管理系统的数据库设计[J]. 电子世界, 2016(18):91-91.
- [13] 赵宸. 浅析黑盒测试与白盒测试[J]. 硅谷, 2010, 000(011):39-39.
- [14] 谢宝良, 习聪玲. 微信公众号开发与使用技术研究[J]. 电脑编程技巧与维护, 2015, 000(009):53,76.
- [15] 李原, 李建军, 杨海成. Web 技术与产品数据管理研究[J]. 制造业自动化, 2000(03):37-40.
- [16] 龚玲张云涛. 计算机网络与 Web 技术[M]. 机械工业出版社, 2005.

致 谢

在论文的完成过程中，得到了我的指导老师王嘉昀老师的巨大帮助。感谢她在百忙之中对我的毕业论文和系统设计上的指导以及文档上的修正。同时还要感谢大学里教过我的各位老师，是他们给了我所有的基础，让我能独立成功的完成此毕业设计。

再次对以上老师表示我的衷心感谢！

最后向在百忙之中评审本文的各位专家、老师表示衷心的感谢！

作者简介：

姓 名：	谢仁义	性别：	男
出生年月：	1997 年 8 月 8 日	民族：	汉
E-mail:	xdadaxie@qq.com		

声 明

本论文的工作是 2019 年 10 月至 2020 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：谢仁义

2020 年 6 月 12 日