19103020128-李佳音-基于Django + React的疫情防控社区志愿者管理系统的设计与实现

【原文对照报告-大学生版】

报告编号：cefd8193e5f1e5a1　　检测时间：2023-04-07 11:04:01　　检测字符数：26872

作者姓名：李佳音　　所属单位：陕西国际商贸学院

陕西国际商贸学院

检测结论：

全文总相似比 **5.17%** = 复写率 4.12% + 他引率 1.05% + 自引率 0.0% + 专业术语 0.0%

其他指标：

自写率：94.83%

高频词：系统,用户,信息,可以,操作

典型相似文章：　无

疑似文字图片：0

指标说明：

复写率：相似或疑似重复内容占全文的比重　　他引率：引用他人的部分占全文的比重

自引率：引用自己已发表部分占全文的比重　　自写率：原创内容占全文的比重

典型相似性：相似或疑似重复内容占全文总相似比超过30%　　专业术语：公式定理、法律条文、行业用语等占全文的比重

相似片段：

总相似片段 23

期刊：1　　博硕：13　　综合：0

外文：0　　自建库：0　　互联网：9

检测范围：

中文科技期刊论文全文数据库　　中文主要报纸全文数据库　　中国专利特色数据库

博士/硕士学位论文全文数据库　　中国主要会议论文特色数据库　　港澳台文献资源

外文特色文献数据全库　　维普优先出版论文全文数据库　　互联网数据资源/互联网文档资源

高校自建资源库　　图书资源　　古籍文献资源

个人自建资源库　　年鉴资源　　IPUB原创作品

时间范围：1989-01-01至2023-04-07

VPCS 维普论文检测系统
VIP PAPER CHECK SYSTEM

原文对照

颜色标注说明：

- 自写片段
- 复写片段（相似或疑似重复）
- 引用片段（引用）
- 专业术语（公式定理、法律条文、行业用语等）

陕西国际商贸学院

学号 19103020128
陕西国际商贸学院本科毕业论文
基于Django+React的疫情防控社区志愿者管理系统的设计与实现
二 级 学 院：信息工程学院
专 业 名 称：计算机科学与技术
学 生 姓 名：李佳音
指 导 教 师：马小菊
二〇二三年五月
郑 重 声 明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。
本人签名： 日期：
摘 要

计算机技术在现代信息管理的应用中，是至关重要的工具，在某些方面能够高效便捷的解决管理员的工作，本系统采用计算机技术，实现对志愿者信息管理的自动化，提高效率。

本课题提出了一个基于Django和React搭建的志愿吧平台，即疫情防控社区志愿者管理系统，旨在帮助机构或组织更好地管理和运营志愿者队伍，提高组织的效率、减少管理成本。首先，本文对志愿者管理的相关背景和现状进行了分析和总结，指出了传统志愿者管理的弊端和不足。其次，本文介绍了课题中功能模块的设计与实现。其主要功能包括用户管理、活动管理、报名信息管理和友情链接管理等。最后，本文Postman测试工具对系统的接口进行了测试，根据结果分析可得出结论，本系统具有良好的用户体验和稳定性，可以满足志愿者管理的实际需求。

本文所提出的基于DRF框架和React框架的疫情防控社区志愿者管理系统具有一定的创新性和实用性，可为志愿公益活动管理提供新的思路和方法。系统功能模块齐全，实现了对志愿者以及疫情防控公益活动管理的系统化、科学化，既可以提高服务质量，又大大的促进了管理系统的发展。

关键词：志愿者管理系统；DRF框架；React框架

Abstract

Computer technology in the application of modern information management, is a vital tool, in some aspects can be efficient and convenient to solve the administrator's work, the system uses computer technology, to realize the automation of volunteer information management, improve efficiency.

VPCS 维普论文检测系统
VIP PAPER CHECK SYSTEM

This project proposes a volunteer bar platform based on Django and React, namely community volunteer management System for epidemic prevention and control , which aims to help organizations or organizations better manage and operate volunteer teams, improve organizational efficiency and reduce management costs. First of all, this paper analyzes and summarizes the background and current situation of volunteer management, and points out the drawbacks and shortcomings of traditional volunteer management. Secondly, this paper introduces the design and implementation of the functional module. Its main functions include user management, activity management, registration information management and friendship link management. Finally, Postman test tool tests the interface of the system. According to the analysis of the results, it can be concluded that the system has good user experience and stability, and can meet the actual needs of volunteer management.

The community volunteer management system for epidemic prevention and control based on DRF framework and React framework proposed in this paper is innovative and practical, and can provide new ideas and methods for the management of voluntary public welfare activities. With complete functional modules, the system has realized the systematic and scientific management of volunteers and public welfare activities of epidemic prevention and control, which can not only improve the quality of service, but also greatly promote the development of the management system.

Key words: Volunteer Management System; Django Restful Framework; React Framework

目 录

安阳国际商贸学院

VPCS 维普论文检测系统
VIP PAPER CHECK SYSTEM

武汉国际商贸学院

VPCS 维普论文检测系统
VIP PAPER CHECK SYSTEM

陕西国际商贸学院

1 绪 论

1.1 课题背景与意义

自2019年以来，新冠疫情形式严峻，各省市各级工作人员全民参与疫情防控工作，各小区处于封闭转态。根据疫情防控工作需要，为进一步组织和引导全省在校大学生志愿者和社区志愿者，积极参加公益志愿活动[1]，维护人民群众身体健康和生命安全。

当前，社区以及各单位对志愿者的管理无法统一，各自有各自的管理方式，耗费人力和时间。而对于疫情防控社区志愿者管理系统平台，它能够节省很多不必要的麻烦、减少人力资源和时间的消耗，更加精准地起到规范化的作用，更加高效地起到一站式服务的管理，对疫情防控带来极大的便利。

基于Django + React的疫情防控社区志愿者管理系统的设计与实现是采用Python、TypeScript XML等语言，Django框架、DRF框架、React框架、Ant Design 组件库、Umi、Ant Design Charts、Ant Design Pro、MYSQL数据库，基于B/S结构进行设计开发。系统主要包括三个用户角色，即管理员角色、普通用户角色、公益企业角色。系统功能齐全，实现了对志愿者以及疫情防控公益活动管理的系统化、科学化，既可以提高服务质量，又大大的促进了管理系统的发展。

1.2 国内外研究现状

目前，关于公益活动管理、公益企业组织管理、志愿者信息管理的相关资料数量少之又少。随着公益活动的日益发展，对其的管理问题将逐渐凸显。目前国内高校志愿者管理的形式大部分以院校志愿者组织团体为主，类似于院校的普通社团组织[2]。

我国志愿者管理是在上个世纪九十年代起步，由于科技方面、经济方面稍微落后以及各种其他因素，对公益活动、志愿者只能通过人工的方式进行管理，没有应用计算机管理系统来对其进行更好的管理。

国外在公益活动行业和志愿行业的研究在国际上乃至全球都产生了很大影响，是因为国外对此的研究相比国内开展的较早。当然，也离不开国外一些发达国家对其在经济层面、法律层面、政策层面的大力支持。更重要的是，很多西方国家比较开放，公民素质也相对于较高，所以他们的贡献精神相对强烈。随着国外一些志愿活动的开展、公益服务组织的完善、相关政策的扶持也促使着志愿服务体系走向成熟[3]。

1.3 本课题工作

本课题的基本工作包括：在本课题开始之前，对相关资料进行调研，对调研结果从不同的层面进行分析，最终确定本系统的总体功能模块。介绍整个系统的功能模块，并将总的系统拆分为多个子系统模块，降低系统的耦合度。对每个子系模块逐个分析其设计流程和开发过程。结合数据库的设计原则，从概念模型和逻辑结构层面对数据库进行设计。系统的前端开发方面，以TypeScript XML为主体开发语言，React为框架，实现SPA（Single page application），即我们常见的单页面应用。系统的后端开发方面，以Python语言为主体开发语言，Django为框架，方便实现Restful 风格的接口。数据库存储方面，选择最流行的关系型数据库，MySQL。以上技术栈在各自的领域都

有各自的优势，目前也有很多成功的开发案例，都属于比较成熟的技术。

2 系统开发环境

2.1 Python 技术

Python相比于其他的开发语言，在语法上占有更多的优势，简洁清晰是它的一大特色，而且，它适用于数据分析、人工智能、机器学习等多种领域。

Python提供了一系列强大的编程库，可以用于实现各种功能，如数据处理，机器学习，图形处理等。同时，还提供了一个交互式编程环境，可以让开发者快速开发和测试代码[4]。

Python的优点有很多，主要包含以下几点：易于学习，语法简洁明了，对编程初学者来说，更好上手。拥有丰富的库，能实现各种业务需求。支持多种编程范式，满足多样化的开发需求。可移植性强；Python拥有优秀的可读性，可以让程序员更容易理解代码。

2.2 MySQL数据库

MySQL在数据存储方面有很强大的功能，易于使用、安装和维护[5]；可以根据场景选择合适的存储引擎；可以使用多种语言来操作MySQL数据库；搭载各种不同的安全机制，可以有效地保护数据库的安全；支持多种复制机制，可以提高数据库的可用性和性能；支持多种分布式架构，可以满足不同的业务需求[6]。

2.3 DRF（Django REST Framework）框架

DRF(Django REST framework)是一个基于Django二次开发的后端框架，它可以让开发者轻松地构建和维护Restful API。它还提供了一系列的序列化器，可以帮助开发者将数据转换为JSON或XML格式，以便在Web浏览器和移动设备之间传输数据。

它提供了一系列的功能，包括路由，视图，序列化，认证，权限，调试、文档，测试。DRF框架的发展可以追溯到2010年，当时它是一个简单的Python库，用于构建RESTful API。随着时间的推移，DRF框架的生态变得越来越强大。现在，它已经成为一个完整的Web框架，可以帮助开发人员快速构建和部署RESTful API。

DRF框架的优势包括：

（1）易于使用

DRF框架具有易于使用的API，可以轻松地构建RESTful API。

（2）可扩展性

DRF框架具有可扩展性，可以根据需要轻松扩展功能。

（3）安全性

DRF框架具有强大的安全性，可以有效地防止CSRF攻击和XSS攻击。

（4）可读性

DRF框架具有可读性，可以轻松地调试和维护代码。

（5）可定制性

DRF框架具有可定制性，可以根据需要轻松定制功能。

2.4 React框架

React是前端三大主流框架之一，也是一个JavaScript库，具有高性能和易于理解的代码逻辑，用来开发前端界面UI。

React使用声明式编程，允许开发人员更轻松地创建动态的用户界面。React可以让开发人员更轻松地处理视图层，并且可以更快地渲染用户界面。它还可以帮助开发人员更轻松地管理状态，从而更轻松地创建可维护的应用程序[7]。

React拥有强大的功能和优点包括：使用JSX语法，可以更容易地构建用户界面。可以使用组件化的方式来构建应用，更容易维护和扩展。拥有虚拟DOM，可以更快地渲染页面。拥有强大的社区支持，可以获得更多的帮助。拥有一些实用的工具，可以更快地开发应用。

2.5 Ant Design Pro

众所周知，阿里巴巴拥有东半球最强大的前端团队。Ant Design Pro就是由阿里旗下的蚂蚁团队推出的。所以，可见Ant Design Pro的强大和知名度。

它是一套企业级模板，主要用于中台和后台产品的搭建，对程序员来说，是一个完美的解决方案和优秀的前端模板。它使用TSX和JSX为开发语言，结合umiJS和 Ant Design二次封装[8]，在不断更新更多功能的同时，不忘追求开发规范。同时，他还搭载了ProComponents页面级组件，整体遵循组件化开发，提高代码复用性。

该模板给程序员提供了很不错的开发体验，开箱即用是它的一大亮点。它还提供权限、网络请求和脚手架等。同时还提供了一系列模板组件，ProLayout，ProTable，ProList等都是开发中后台的优质资源，可以显著的减少样板代码[9]。

2.6 系统开发环境

操作系统：Windows10

Python运行环境：Python3.6

Node环境：Node -v16.13.2

数据库：MySQL 5.7

React运行环境：React17.0.0

2.7 系统开发工具

编译器：PyCharm，VSCode

数据库可视化工具：Navicat Premium

测试工具：Postman

接口文档工具：Swagger

浏览器：Google 浏览器和Edge浏览器

3 系统分析

3.1 可行性分析

疫情防控社区志愿者管理系统的主要功能是对公益活动信息和每个活动参加志愿者的管理。可行性分析要从系统的技术层和可操作性对系统进行全面分析，以下根据这三点对疫情防控社区志愿者管理系统进行全面的分析。

3.1.1 技术可行性

疫情防控社区志愿者管理系统主要采用Python语言和tsx语言，基于B/S结构进行开发[10]，DRF(Django REST Framework)框架完成接口的实现，React框架实现前端的设计与开发，MySQL数据库实现对数据的存储。以上技术为本系统的关键技术，可以支撑并完成对本系统的开发语实现，所以在技术可行性分析方面没有太大的问题。

3.1.2 操作可行性分析

现在已有的管理系统对使用者来说都是很简单方便，该系统也例外，根据使用者的角度，该系统在开发上致力于简化结构，使页面尽可能简洁，优化用户操作，使用者只需要在浏览器中打开系统并进行登录就可以进行使用操作系统。综上所述，该系统的操作简单，方便，可以进行开发。

3.2 系统流程分析

3.2.1 系统开发流程

关于系统的开发流程的分析如下：

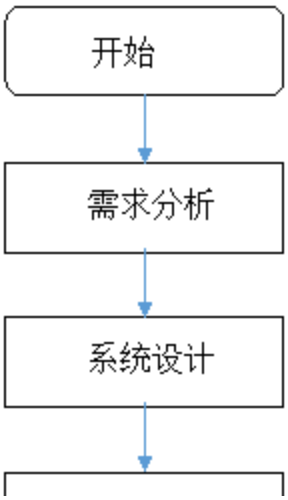
（1）需求分析阶段：确定系统功能模块，划分子功能模块。

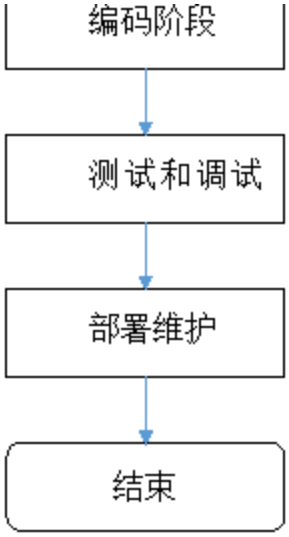
（2）系统设计阶段：设计系统的模块、组件、数据、接口等。

（3）开发阶段：按照开发计划，逐步开发独立的子模块，实现系统编码。

（4）测试和调试阶段：为确保本系统功能模块的可靠性，使用Postman工具，对系统接口进行测试。

（5）部署和维护阶段：部署系统到实际环境中，并对系统进行维护和升级。





系统开发的流程如图3.1所示。

图3.1 系统开发流程

3.2.2 用户注册流程

只有在注册账号后，普通用户和公益企业用户才能登入到系统内部。

用户注册流程分析：

（1）打开注册表单

用户进入登录页面，切换到注册表单。

（2）填写注册信息

根据表单提示，填写注册信息。

（3）校验信息

考虑到用户可能输入的信息有误，需要对数据进行检查，前端判断两次输入的密码是否一致，若不一致，提示用户，并且清空表单，让用户重新输入。若一致，前端请求注册接口，后端执行注册方法。

（4）返回结果

前端根据后端返回的响应结果，弹出提示框，提示用户注册结果。

（5）操作完成

用户根据提示信息进行操作，注册成功，则登录系统，注册失败，则根据失败的原因重新进行注册。

在设计系统注册流程时，需要考虑以下几个方面：

（1）界面设计

注册页面应该设计清晰、简单，便于用户填写相关信息。

（2）异常处理

系统需要捕获异常，例如用户填写错误信息、系统故障等情况。

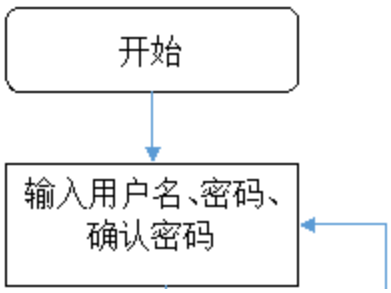
（3）用户体验

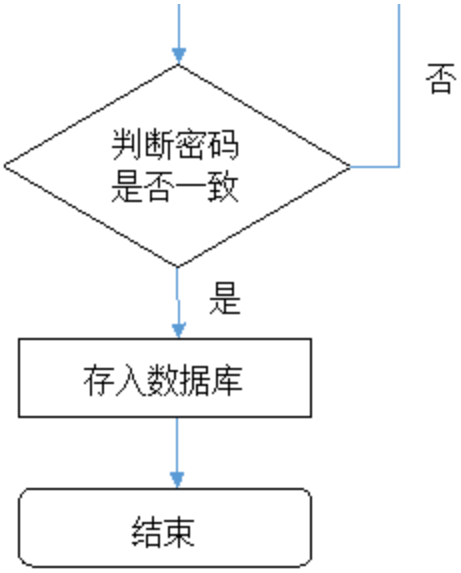
操作流程应该简单、易于理解，减少用户操作的复杂性和困难度。

（4）安全保障

系统应该采取相应的安全措施，确保用户的数据安全和隐私保护。

合理分析系统注册流程可以提高效率和用户体验，以及系统的可靠性和安全性。





注册流程如图3.2所示。

图 3.2 注册流程

3.2.3 用户登录流程

数据库的安全问题是非常重要的，数据库信息一旦泄露，便会给系统带来很大的危害[11]，例如网站可能会遭到恶意攻击，数据窃取等。为了防止此类情况发生，保证系统安全，session中只有监测到当前用户的信息，才能使用本系统，即只有在登录态下才可以操作系统。

由于不同的角色的权限不同，即功能不同[12]，所以用户在登录时，要选择角色，在点击登录时，后端要执行登录方法，进行响应。前端将响应的当前用户信息（currentUser）、角色（userType）、用户编号（user_id）存储到session中，然后根据session中存储的角色去匹配权限以及可访问的路由。

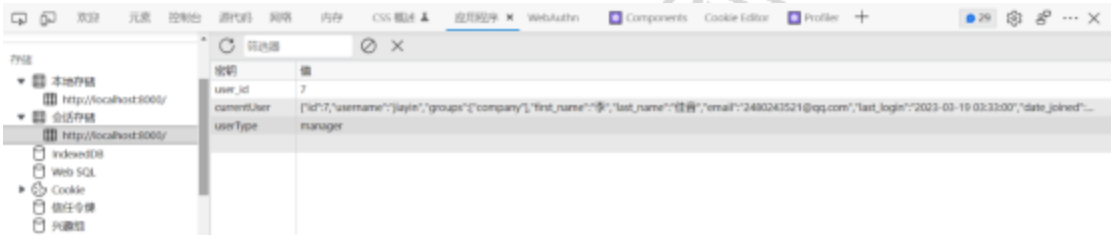
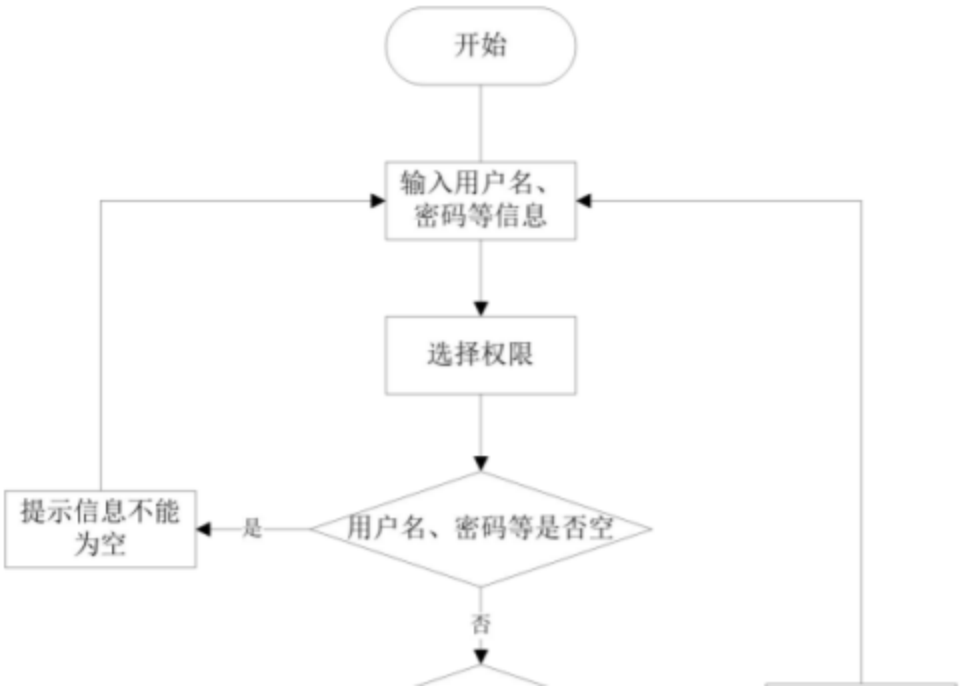
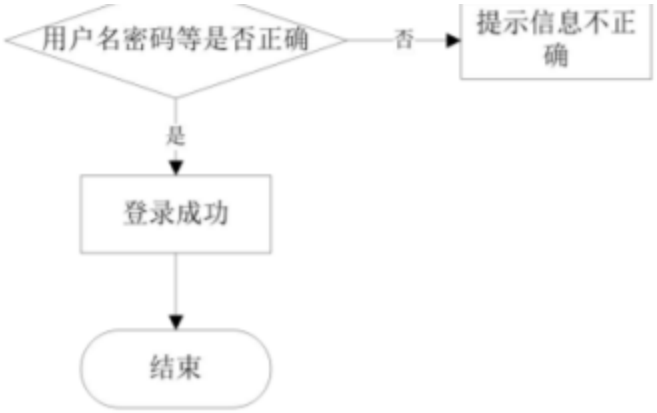


图3.3 session中的存储密钥登录成功后session中的存储的密钥如图3.3所示。





用户登录流程如图3.4所示。

图3.4 用户登录流程

3.2.4 系统操作流程

系统操作流程，是指用户在体验或使用系统时所需要遵循的操作步骤和流程，其主要目的是为了帮助系统用户完成相应的操作任务。系统操作流程包括以下几个方面：

（1）登录操作

用户需要输入登录信息，登录系统后进行后续的操作。

（2）界面操作

用户需要了解系统的界面布局和操作方式，包括菜单栏、工具栏、状态栏等，以便于快速定位和操作所需要的功能。

（3）功能操作

用户需要根据自己的需求，选择对应的功能模块进行操作，例如查询、新增、编辑、删除等操作。

（4）数据输入和输出

系统根据用户输入的查询条件输出查询结果。

（5）操作流程控制

在操作过程中，用户需要根据系统的提示信息和状态信息，控制操作流程，确保操作的正确性和完整性。

（6）操作结果处理

用户需要根据操作结果，进行相应的处理，例如保存数据、打印报表、导出数据等。

（7）系统退出操作

用户在完成操作后，需要进行系统退出操作，以保护系统和用户的数据安全。

在设计系统操作流程时，需要着重考虑用户体验，使得系统操作流程更加简单和易于使用。同时，需要将系统操作流程设计为可定制和可配置的，以便于满足不同用户的需求和使用习惯。

3.2.5 添加信息操作流程

系统添加信息操作流程包括以下几个步骤：

（1）打开添加信息对话框

用户需要打开系统中相应的功能模块，并选择添加信息的操作。

（2）填写信息

用户填写需要添加的信息字段，比如名称、描述、时间、地点等。

（3）检查信息

为确保添加的信息无误，用户在提交前需要仔细检查。

（4）提交表单

用户通过提交表单操作，将数据传给到系统。

（5）系统处理信息

后端执行POST新增方法，对信息进行检查，保存等操作。

（6）系统响应

处理完信息，系统返回响应结果。

（7）操作完成

前端根据系统后端返回的结果，进行对应的处理，例如保存信息、继续添加信息、返回信息列表页面、更新信息列表等。
在设计添加信息操作流程时，需要考虑以下几个方面：
（1）界面设计
添加信息界面应该设计清晰、简单。
（2）数据校验
考虑到用户可能输入的信息有误，前端需要有表单校验功能，后端需要有数据校验功能。
（3）异常处理
针对可能出现的不同错误情况，例如用户输入错误信息、系统故障等情况，系统需要捕获异常。
（4）用户体验
操作流程应该简单、易于理解，减少用户操作的复杂性和困难度。
（5）安全保障
系统应该采取相应的安全措施，确保用户的数据安全和隐私保护。
通过合理设计添加信息操作流程，可以提高用户的使用效率和用户体验，同时也可以提高系统的可靠性和安全性。
3.2.6 修改信息操作流程
系统修改信息操作流程包括以下几个步骤：
（1）选择要编辑的信息
选择目标信息，点击操作中的编辑按钮。
（2）打开信息编辑对话框
点击操作后，系统会自动打开对话框，显示其详细内容。
（3）修改信息
用户对信息进行修改，例如修改名称、描述、时间、地点等。
（4）检查信息
为确保修改的信息无误，用户需要仔细检查修改后的信息是否正确。
（5）提交表单
用户通过提交表单操作，将数据传给到系统。
（6）系统处理信息
后端执行PUT或PATCH更新方法，对信息进行检查，更新等操作。
（7）系统响应
处理完信息，系统返回响应结果。
（8）操作完成
前端根据后端返回的结果，进行对应的处理，例如保存修改后的信息、继续编辑信息、跳转回列表页面等。
在设计修改信息操作流程时，需要考虑以下几个方面：
（1）界面设计
为了方便用户对信息进行修改操作，信息编辑界面应该设计清晰、简单。
（2）数据校验
考虑到用户可能修改后的信息有误，前端需要有表单校验功能，后端需要有数据校验功能。
（3）异常处理
针对可能出现的不同错误情况，例如用户输入错误信息、系统故障等情况，系统需要捕获异常。
（4）用户体验
操作流程应该简单、易于理解，减少用户操作的复杂性和困难度。
（5）安全保障
系统应该采取相应的安全措施，确保用户的数据安全和隐私保护。
通过合理设计修改信息操作流程，可以提高用户的使用效率和用户体验，同时也可以提高系统的可靠性和安全性。
3.2.7 删除信息操作流程
系统删除信息操作流程包括以下几个步骤。
（1）选择要删除的信息

选择目标信息，点击删除操作按钮。
（2）确认删除
由于数据删除操作是不可逆的，删除之后无法恢复，用户需要对要删除的信息进行确认，确认之后，点击确认按钮，提交数据。
（3）系统处理信息
前端检测到用户提交的删除信息，会向后端发送请求，后端进行删除操作。
（4）系统响应
处理完信息，系统返回响应结果。
（5）操作完成
用户根据系统返回的结果，进行相应的处理，例如刷新信息列表、继续管理信息等。
在设计删除信息操作流程时，需要考虑以下几个方面：
（1）界面设计
信息管理界面应该设计简单明了，便于用户选择要删除的信息。
（2）数据校验
确保数据存在且准确无误，后端要查询删除的数据是否在数据库中存在。
（3）异常处理
系统需要对异常情况进行处理，例如用户选择错误信息、系统故障等情况。
（4）用户体验
操作流程应该简单、易于理解，减少用户操作的复杂性和困难度。
（5）安全保障
系统应该采取相应的安全措施，确保用户的数据安全和隐私保护。
通过合理设计删除信息操作流程，可以提高用户的使用效率和用户体验，同时也可以提高系统的可靠性和安全性。
3.3 系统用例分析
3.3.1 管理员用例
管理员角色可以管理整个系统的数据[13]。可以说是该角色可以操作所有模块和数据，该角色的用户一旦泄露，整个系统的数据都有被窃的风险。所以，该角色不可以自行注册，只能在数据库添加。
管理员用例分析：
（1）管理用户
为确保系统用户的账号能够正常使用，该角色可以进行添加用户、重置密码、修改用户、导出、批量注销等操作。
（2）管理信息
该角色可以管理全部的信息，对所有信息进行CRUD操作，以确保信息的准确性和完整性。
在管理员用例分析中，需要考虑以下几个方面：
（1）用户交互体验
管理员可以通过简单、直观的界面操作系统，提高用户的使用效率和用户体验。
（2）安全保障
系统应该采取相应的安全措施，例如密码安全策略、数据备份策略等，确保系统的安全性和可靠性。
（3）异常处理
系统需要对异常情况进行处理，例如系统故障、服务器异常等情况。
（4）数据管理
确保用户信息和系统信息的准确性和完整性，该角色需要对用户和信息进行管理[14]。
通过对该角色合理的用例分析，可以提高系统的管理效率，维护系统数据的稳定性。
管理员用例如图3.5所示。

长沙国际商贸学院

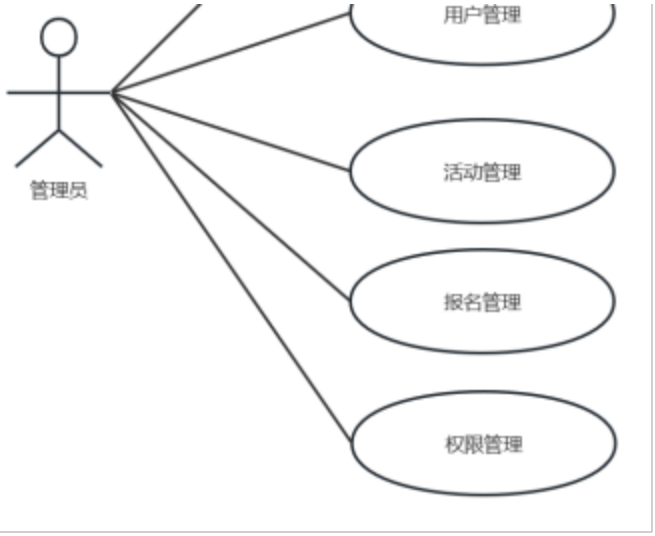


图3.5 管理员用例图

3.3.2 公益企业用例

公益企业用户可以进行活动的发布、编辑、删除、查看，以及对普通用户的报名信息进行审核[15]。

公益企业用例如图3.6所示。

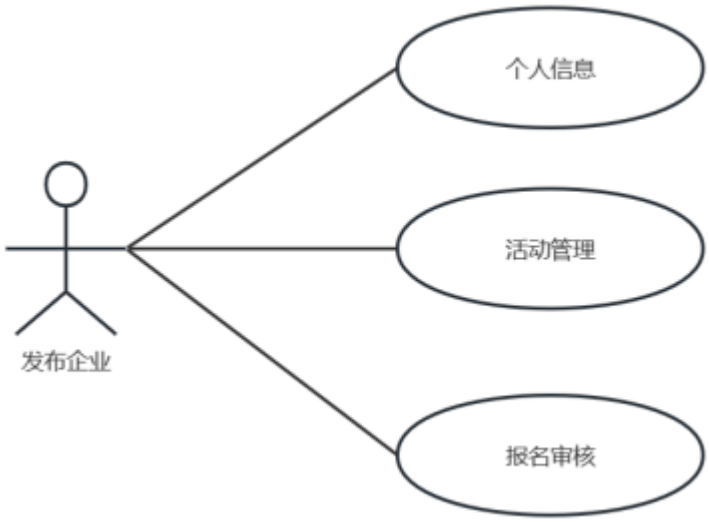


图3.6 发布企业用例

3.3.3 普通用户用例

普通用户在活动列表页面，可以查询活动，查看活动详情，报名自己感兴趣的公益活动，查看自己的报名状态。

普通用户用例如图3.7所示。

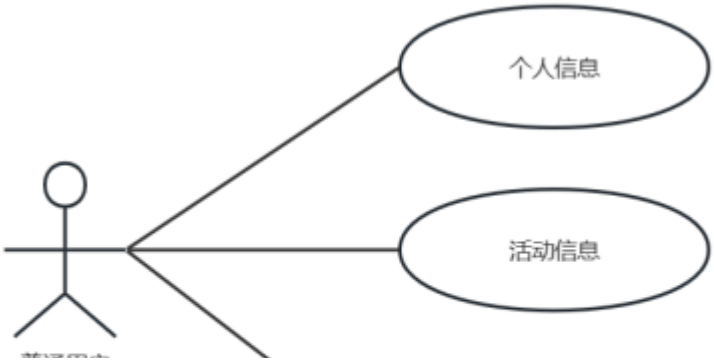


图3.7 普通用户用例

4 系统设计

4.1 系统结构设计

系统结构设计是指基于系统需求和系统功能，设计出系统的整体结构框架，包括系统的模块、组件、接口、数据流等。系统结构设计的目的是为了将系统划分为不同的部分，使得系统的开发、测试、维护和升级都更加容易和高效。

系统结构设计包括以下几个步骤：

（1）确定系统需求

为了方便后期的开发等工作，要确定好系统需求。

（2）划分子系统

将总系统分解为若干个独立的子系统[16]，确定每个子系统的功能模块，它们之间通过接口进行通信和数据交换。

（3）设计系统接口

确定不同模块之间的接口协议和数据格式，以确保数据的正确传输和解析。

（4）确定系统数据流

确定系统中各个模块之间的数据流动方式，以及数据流程的控制和管理方式。

（5）设计系统组件

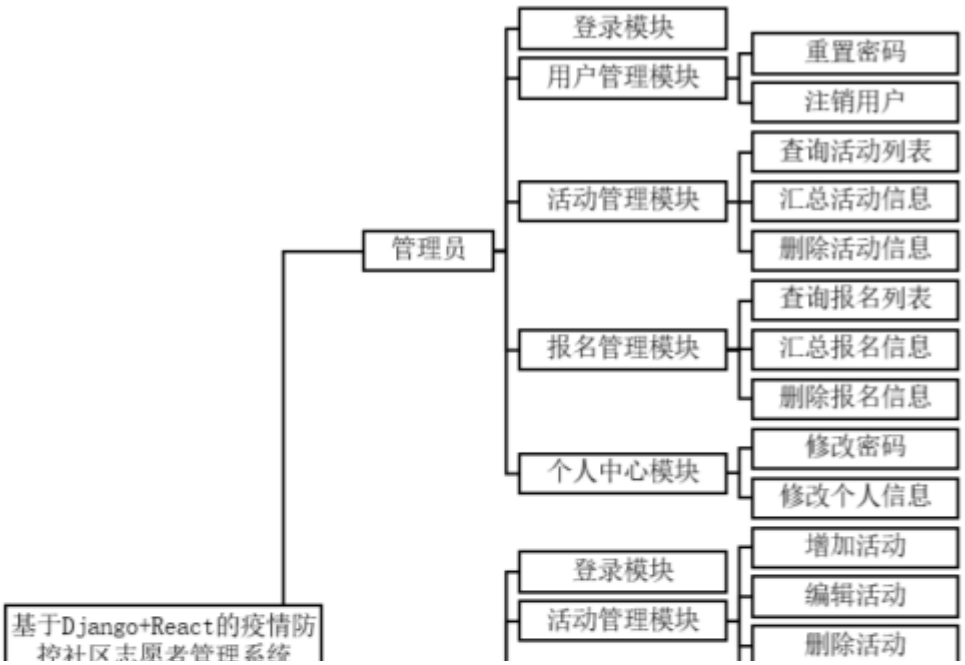
根据系统需求和模块划分结果，设计系统的组件，包括硬件组件、软件组件、网络组件等[17]。

（6）确定系统架构

根据系统模块、组件和接口的设计，确定系统的总体架构，包括层次结构、模块之间的依赖关系、系统的部署方式等。

系统结构设计要遵循“高内聚，低耦合”的原则，将总系统分解为若干个可以独立开发和测试的子模块。在设计过程中，需要考虑系统的可移植性、可复用性、可维护性、可靠性、可扩展性等方面的问题。最后，根据实际需求选择合适的设计模式和架构模式，以便于实现系统的功能和优化系统的性能。

系统功能模块如图4.1所示。



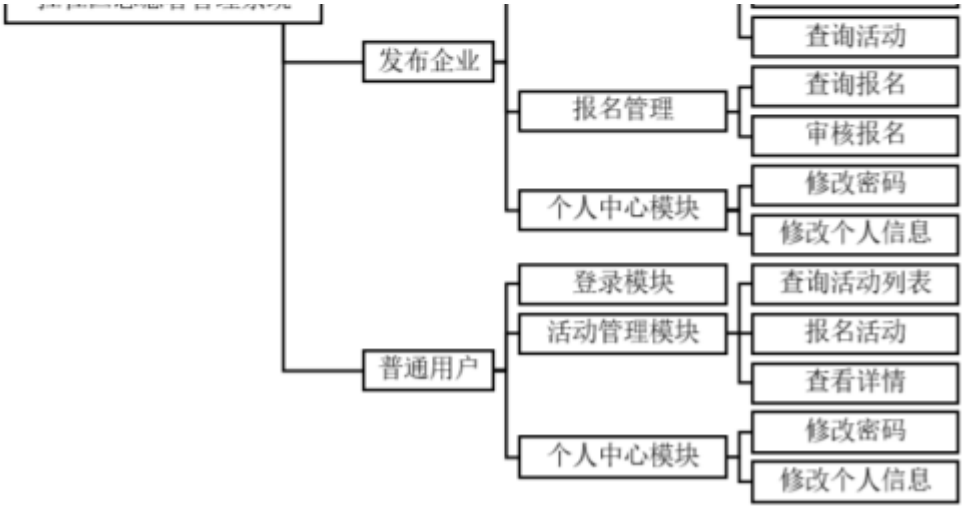


图 4.1 系统功能模块

4.2 数据库设计

数据库设计的目的是为了保存数据，在设计数据库和创建表时，不仅要设计合理，还要保证不脱离理论知识，这就要求我们通过不断地分析实体属性来确定每个字段的特征。

数据库的设计流程如下：

（1）需求分析

需求分析的核心是要确定数据特征[18]。在对数据库进行分析时，要对每个字段进行详细的解析，最终要确定该字段是某种类型、长度为多少、键码、能否为空值等。比如Id编号作为表中唯一标识的字段，它的类型是bigint，长度为11，为主键，在同一表中不能重复，不能为空，相邻数据行依次递增。

（2）概念设计

我们通过ER图来设计概念模型[19]。在设计概念模型时，我们要确定出ER图的实体有哪些[20]，每个实体有哪些属性，不同实体之间的对应关系是什么。比如公益企业实体与公益活动实体的关系为发布，即公益企业发布公益活动。一对多，一个公益企业可以发布多个活动。从实体、关系、属性等方面设计其的概念模型。

（3）逻辑设计

在逻辑模型的基础上，设计出物理模型，包括表空间、索引、分区等。

（4）物理设计

根据逻辑设计，设计出物理模型，包括表空间、索引、分区等。

（5）实现

根据对数据库的设计与分析，创建数据库和表。

在设计数据库时，不仅要保证数据不会泄露，且稳定性和兼容性良好，还要方便我们管理数据。这就要求在创建数据库时要遵循主键约束、完整性、范式化等原则。

4.2.1 数据库设计原则

数据库设计原则是指在设计数据库时需要遵循的基本原则，这些原则可以使我们创建出的数据库结构合理、稳定、安全、易于维护和扩展。在设计该系统的数据库时要遵循原则包括以下几点：

（1）范式化原则

应该遵循范式化原则，即将重复的数据分解到不同的表中，以避免冗余数据的存在，从而提高数据的一致性。

（2）实体完整性原则

确保每个表都包含主键字段，还要保证每一条数据都是一个完整的实体。

（3）关系完整性原则

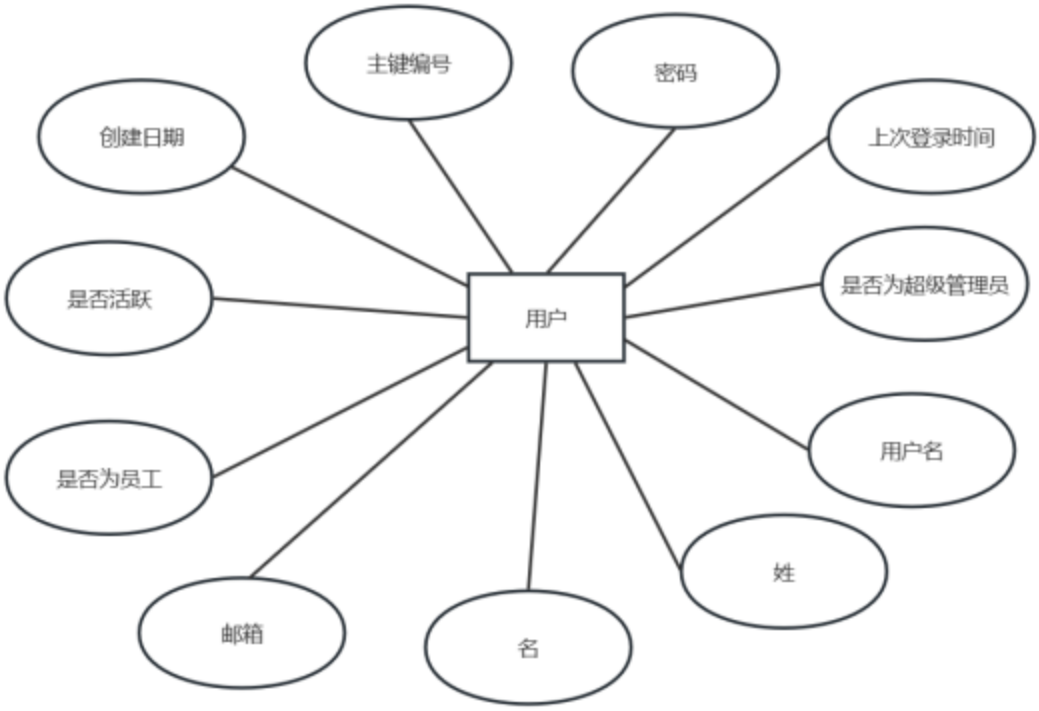
对于有关联关系的表，保证两个表能够正确对应。

（4）数据安全原则

为了防止数据库泄露，对数据库中的敏感字段进行加密，保证用户的信息安全。

4.2.2 概念模型设计

本系统有用户、报名信息、活动信息等多个实体，下面分别是各实体和整体的 E-R 图。

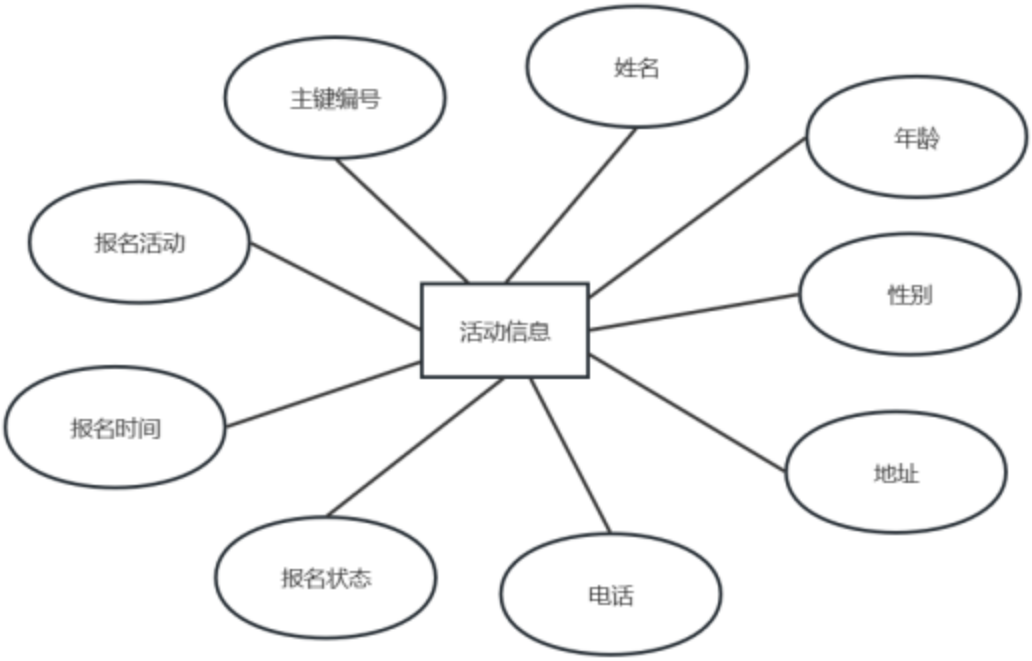


（1）用户实体如图4.2所示。

图4.2 用户实体

（2）活动信息实体如图4.3所示。

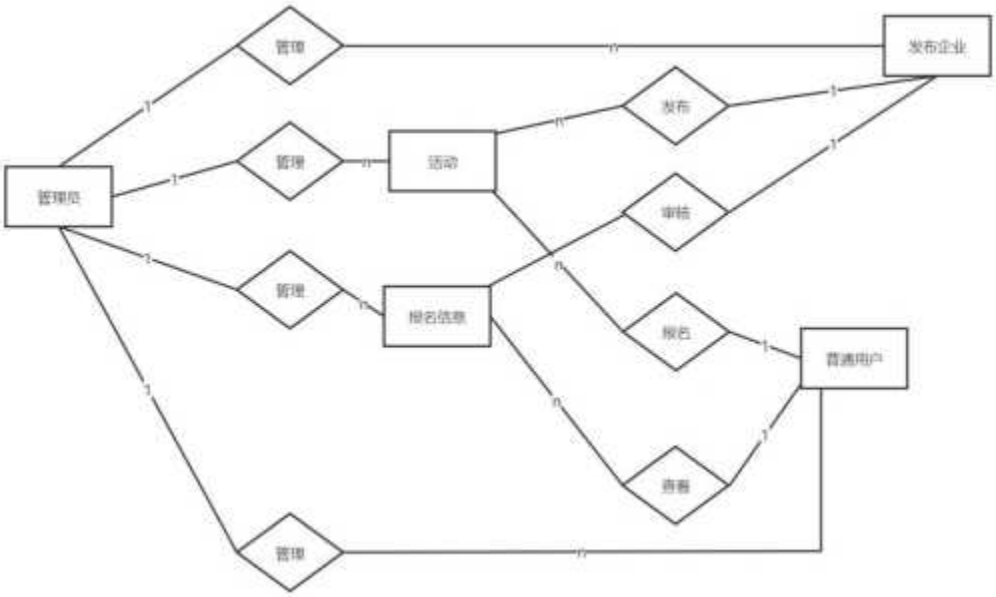
图4.3 活动信息实体



（3）报名信息的实体如图4.4所示。

图4.4 活动信息实体

根据以上分析，本系统存在了以下五个重要实体：活动信息、报名信息、管理员、普通用户、发布企业。



系统全局 E-R 如图4.5所示。

图4.5 全局E-R图

本系统各个实体关系，如表4.1所示。

表4.1 实体关系表

关系表	实体1	实体2	联系类型	系统需求
联系1	管理员	用户	1:m	一个管理员管理多个用户
	用户	管理员	m:n	多个用户可以被多个管理员管理
联系2	公益企业	活动	1:m	一个公益企业发布多个活动

	活动	公益企业	1:1	一个活动只能被一个公益企业发布
联系3	普通用户	活动	1:m	一个普通用户可以报名多个活动
	活动	普通用户	m:n	多个活动可以被多个用户报名
联系4	公益企业	报名信息	1:m	一个公益企业可以审核多个报名信息
	报名信息	公益企业	m:1	多个报名信息可以被同一个公益企业审核
联系5	管理员	友情链接	1:m	一个管理员可以管理多个友情链接
	友情链接	管理员	m:n	多个友情链接可以被多个管理员管理
联系6	管理员	活动信息	1:m	一个管理员可以管理多个活动信息
	活动信息	管理员	m:n	多个活动信息可以被多个管理员管理

表4.1 实体关系表（续）

联系7	管理员	报名信息	1:m	一个管理员可以管理多个报名信息
	报名信息	管理员	m:n	多个报名信息可以被多个管理员管理

4.2.3 数据库逻辑结构设计

用户信息表包括编号、用户名、等个人信息，如表4.2所示。

表4.2 用户信息表

字段名称	字段含义	类型	长度	键	是否为空
id	主键编号	int	11	主键	否
password	密码	varchar	128		否
last_login	上次登录时间	datetime	6		能
is_superuser	是否为超级管理员	tinyint	1		否
username	用户名	varchar	150		否
first_name	姓	varchar	30		能
last_name	名	varchar	150		能
email	邮箱	varchar	254		能
is_staff	是否为员工	tinyint	1		能
is_active	是否活跃	tinyint	1		能
date_joined	创建日期	datetime	6		否

活动信息表包含编号，名称等信息，如表4.3所示。

表4.3 公益活动表

字段名称	字段含义	类型	长度	键	是否为空
id	主键编号	int	11	主键	否
name	活动名称	varchar	50		否
desc	活动描述	varchar	255		否
publish_company_name	发布企业名称	varchar	50		否
address	活动地点	varchar	50		否
start_date	活动日期	date	0		否
start_time	活动时间	time	6		否
demand	活动要求	varchar	255		否

表4.3 公益活动表（续）

need_person_num	需要人数	int	11	否
apply_person_num	报名人数	int	11	否
pass_person_num	通过人数	int	11	否
create_time	创建时间	datetime	6	否

报名信息表如表4.4所示。
表4.4 报名信息表

字段名称	字段含义	类型	长度	键	是否为空
id	主键编号	int	11	主键	否
name	志愿者姓名	varchar	50		否
age	志愿者年龄	int	11		否
sex	志愿者性别	int	11		否
address	志愿者地址	varchar	50		否
tel	志愿者电话	varchar	255		否
apply_status	报名状态	int	11		否
apply_time	报名时间	datetime	6		否
belonging_activity_id	报名活动	int	11		否

分组信息表如表4.5所示。
表4.5 分组信息表

字段名称	字段含义	类型	长度	键	是否为空
id	主键编号	int	11	主键	否
name	名称	varchar	150		否

权限信息表如表4.6所示。
表4.6 权限信息表

字段名称	字段含义	类型	长度	键	是否为空
id	主键编号	int	11	主键	否
name	名称	varchar	255		否
content_type_id	类型id	int	111		否
codename	代码名称	varchar	00		否

用户权限关系对应表如表4.7所示。
表4.7 用户权限关系关联表

字段名称	字段含义	类型	长度	键	是否为空
id	主键编号	int	11	主键	否
user_id	用户编号	int	11		否

perssion_id	权限编号	int	111		否

用户分组关系对应表如表4.8所示。
表4.8 用户分组关系关联表

字段名称	字段含义	类型	长度	键	是否为空
id	主键编号	int	11	主键	否
user_id	用户编号	int	11		否
group_id	分组编号	int	111		否

分组权限关系对应表如表4.9所示。
表4.9 分组权限关系关联表

字段名称	字段含义	类型	长度	键	是否为空
id	主键编号	int	11	主键	否
group_id	分组编号	int	11		否
perssion_id	权限编号	int	111		否

5 系统的设计与实现

5.1 公共模块

5.1.1 登录模块

登录模块是一个重要的功能模块，每个用户在打开系统界面时，都要进行登录操作。用户只有在登录状态下，才可以操作系统。

进入登录页面，在登录表单填写并提交登录信息，前端将登录信息作为参数体，通过POST方法请求login接口，后端检测到前端发出的请求，执行登录方法，然后将结果响应给前端。

前端根据响应结果，在服务器端记录用户的登录状态，并将用户的ID、角色等信息存储在会话中，以便后续的操作。前端根据后台返回的登录状态，给出相应的提示信息，页面进行跳转。

登录页面如图5.1所示。
图 5.1 登录页面

5.1.2 个人中心模块

个人中心作为一个系统必须的模块，用户可以在该模块查看个人信息并对其进行修改操作，还可以进行修改密码的操作。在实现该模块时，需要对用户信息进行权限控制，保证个人信息的安全性、性能和用户体验。
个人中心页面如图5.2所示。

VPCS 维普论文检测系统
VIP PAPER CHECK SYSTEM

图5.2 个人中心页面

5.1.3 文件导出模块

用户可以下载导出活动信息表、报名信息表、用户信息表等。后端收到用户的下载请求，将数据库中的表导出为Excel文件，返回给前端文件编号，前端再去通过编号请求下载文件接口，请求成功，浏览器执行下载操作。本系统的下载分为两种方式：默认下载和批量下载，默认下载，即下载用户当前在页面所看到的全部数据。批量下载下载用户在页面表格中勾选的数据。后端在处理文件的导出时，采用了python中的xlwt包，将数据转换为Excel文件。导出的活动表文件如图5.3所示。

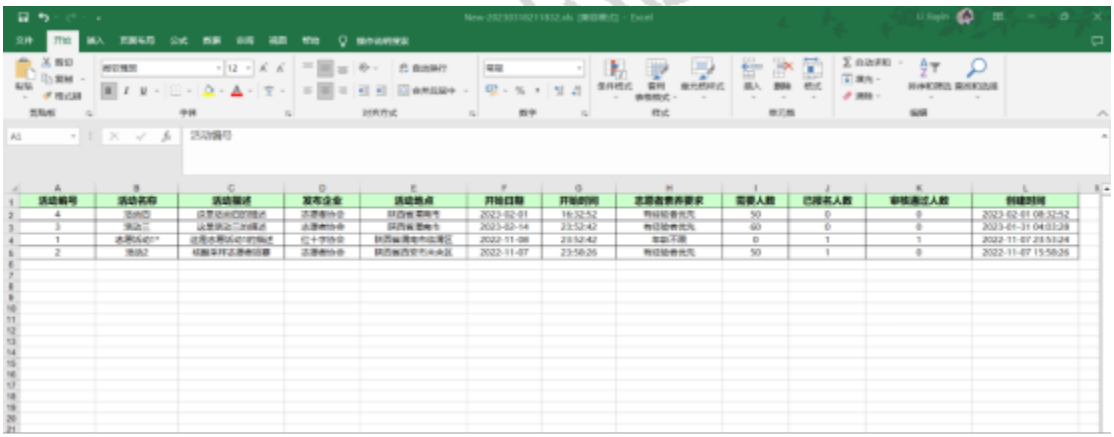


图5.3 活动表Excel文件

文件导出部分实现代码如下：

```
import xlwt, datetime
from xlwt import *
    def write_to_excel(n, head_data, records, download_url):
     timestr = datetime.datetime.now().strftime("%Y%m%d%H%M%S")
wbk = xlwt.Workbook()
     sheet1 = wbk.add_sheet('sheet1', cell_overwrite_ok=True)
for filed in range(0, len(head_data)):
     sheet1.write(0, filed, head_data[filed], excel_head_style())
for row in range(1, n + 1):
for col in range(0, len(head_data)):
     sheet1.write(row, col, records[row - 1][col], excel_record_style())
sheet1.col(col).width = 300 * 20
wbk.save(download_url + 'New-' + timestr + '.xls')
return timestr
```

5.1.4 接口文档模块

本系统接入了接口文档，接口文档模块是一份描述系统或应用程序接口的文档。本系统的接口接口文档模块的实现使用Django REST Swagger技术，Django REST Swagger是一个Django应用程序，它可以自动生成Web API的文档。它使用Swagger规范（也称为OpenAPI规范）来描述RESTful API，并提供交互式的文档页面，以便更容易地了解API的设计和使用方法。

接口文档页面如图5.4所示。

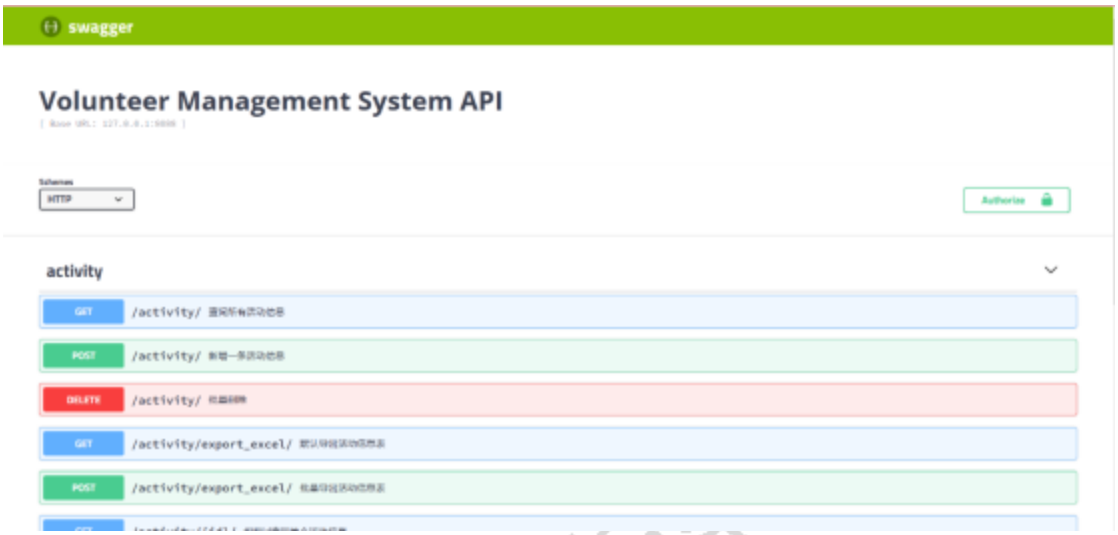


图5.4 接口文档页面

5.2 普通用户模块

普通用户登录系统可以在活动列表页面，查询自己感兴趣的活动，报名活动。在我的报名页面，可以查看相关邮件通知、自己的报名信息是否通过审核，在审核之前，用户还可以对自己的信息进行修改。

5.2.1 活动列表模块

进入活动列表模块，普通用户可以查看公益活动信息，查询自己感兴趣的公益活动，可以看到每个需要多少人数、已报名的人数、报名审核通过的人数，对自己想要参加的公益活动进行报名。

普通用户活动列表如图5.5 所示。

图5.5 活动列表页面

5.2.2 报名列表模块

普通用户在我的报名页面，可以查看自己的报名是否通过审核。审核之前，普通用户还有权限对其进行修改，一旦进入审核，系统将锁定用户的报名信息，不可以再对其进行编辑。报名完成后，报名信息等待公益企业审核，审核完毕后，普通用户可以查看自己的报名状态以及相关的通知。

我的报名页面如图5.6所示。

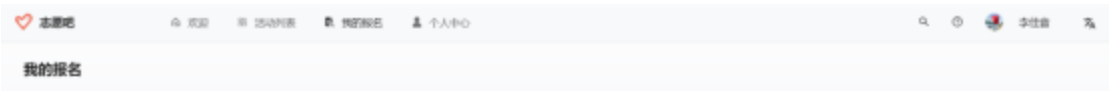


图5.6 我的报名页面

5.3 公益企业用户模块

公益企业用户的主要功能是发布活动和审核报名。公益企业依照自身情况，组织志愿活动，招募志愿者。活动发布后，普通用户可以进行报名，公益企业在报名管理模块对报名信息进行审核。

5.3.1 活动管理模块

包括活动发布、活动编辑、活动查看、活动删除、活动导出、批量删除、批量导出等功能，可以帮助公益企业发布新的志愿者活动信息，吸引志愿者参与活动，同时也方便公益企业机构管理和调度活动。同时还包含活动数据统计与分析，对活动的需求人数，报名人数、审核通过人数等进行统计分析，为公益企业用户提供数据支撑，帮助公益企业更好地了解和优化活动。公益活动数据统计分析，是由后端进行根据报名信息进行计算统计，然后返回给前端。前端页面进行展示。

公益企业的活动管理模块前端页面如图5.7所示。

图5.7 活动管理页面

活动管理部分实现代码如下。

5.3.2 报名管理模块

在该模块，公益企业负责对报名申请进行审核，审核状态分为三种状态，待审核、审核已通过、审核未通过。审核结束后，系统会自动给普通用户发送邮件通知，普通用户查看自己的报名状态，以及邮件通知。

报名管理页面如图5.8所示。

维普论文检测系统
VIP PAPER CHECK SYSTEM

图5.8 报名管理页面

5.4 管理员模块

管理员的功能除了包含以上功能模块外，还有用户管理模块。

5.4.1 用户管理模块

该模块包括新增用户、用户编辑、批量删除、批量导出、注销用户等操作。对于忘记密码的用户，管理员可以对其进行密码初始化。用户使用初始密码登录系统后，可自行修改密码。管理员可以注销违规或长时间未使用的用户。用户管理页面如图5.9所示。

图5.9 用户管理页面

5.4.2 友情链接管理

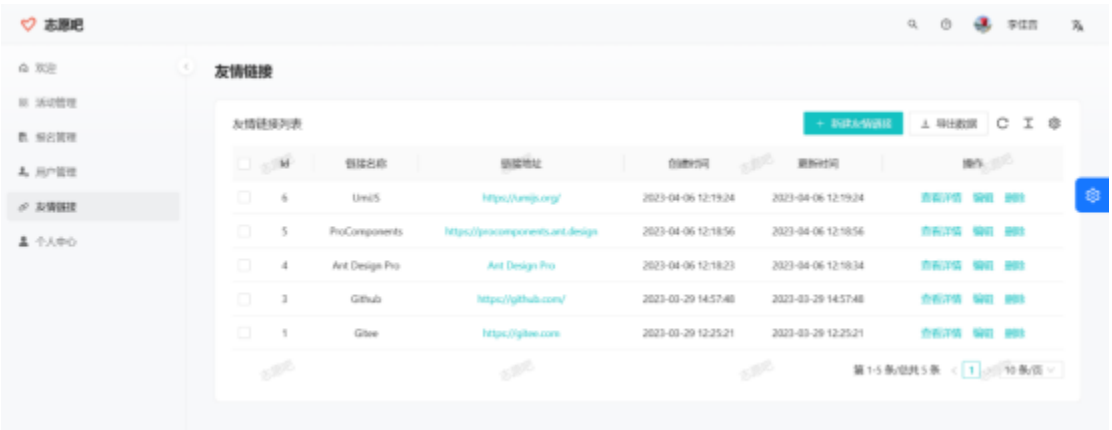


图5.10 友情链接管理页面该系统的实现离不开现有资源的技术支持，所以在系统中展示相关技术资源的友情链接，来表示对其的感谢。该模块由管理员对其进行管理，管理员上传本系统相应友情链接，系统对其进行展示。友情链接页面如图5.10所示。

6 系统测试

6.1 测试目的

验证系统符合预期的功能和性能要求；评估系统的稳定性和安全性，检查系统在异常情况下的表现和处理能力；从可操作性、可扩展性、实用性等多个方面对系统进行检查和评估；确保系统符合相关法律法规和行业标准，检查系统的合规性和安全性；检查前端页面在不同浏览器、不同缩放比例下的兼容性；发现并修复系统中的缺陷和问题，提高系统的质量和可靠性。

6.2 接口测试

传统的人工测试方法存在很多的弊端，耗时耗力，并且对于很多细节的问题容易遗漏。所以本课题使用Postman对应用程序接口进行测试，其优点在于可以进行快速、方便的测试和调试。Postman接口测试的流程如下：创建请求，输入URL，选择请求方法，添加参数。在发送请求后，Postman会处理请求并显示响应结果。请求结束后，可以直观的看到响应结果，响应结果包括状态码、响应头、响应体等信息。

测试查询报名列表接口过程如图6.1所示。

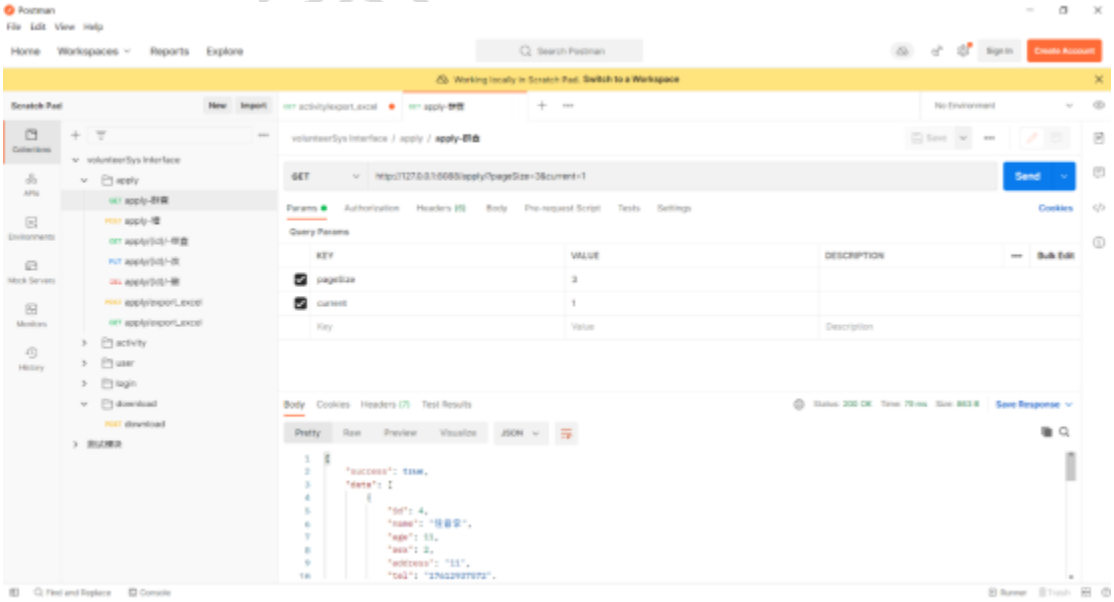
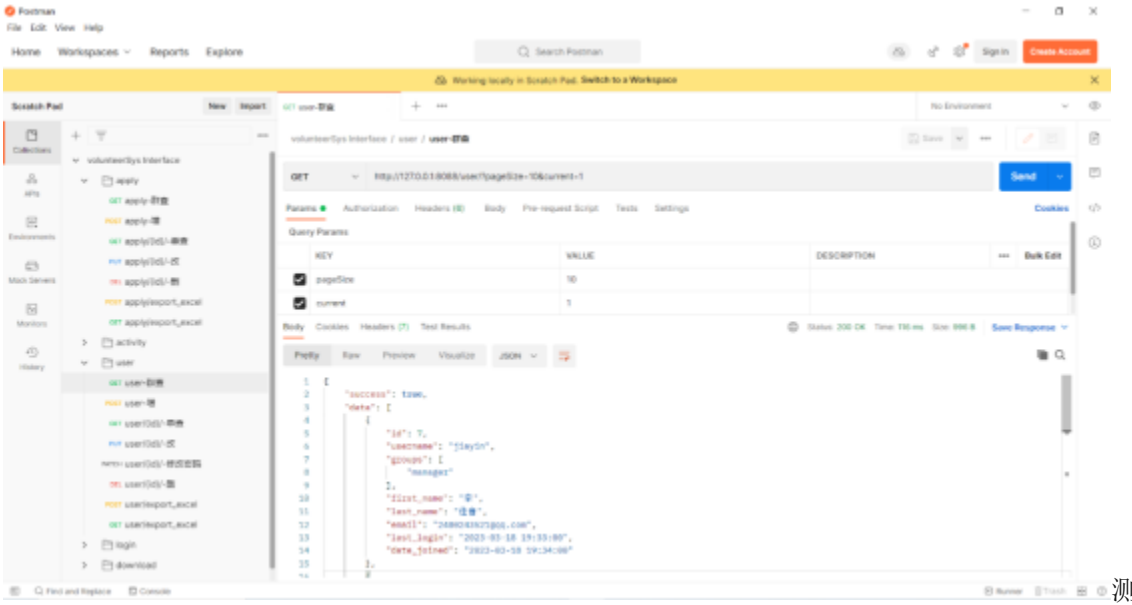


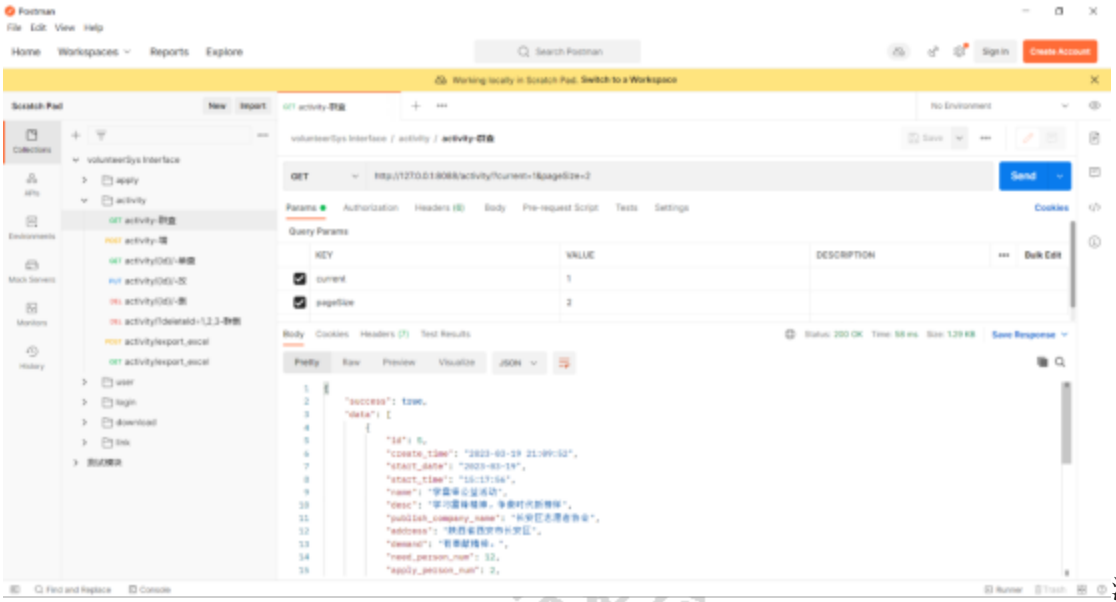
图6.1 查询报名列表接口测试

图6.2 查询用户列表接口测试



测试查询用户列表接口过

维普论文检测系统
VIP PAPER CHECK SYSTEM
程如图6.2所示。
图6.3 查询活动列表接口测试



测试查询活动列表接口过程如图6.3所示。
6.3 测试结果
经过多组测试数据对系统进行测试，该系统功能相对齐全，基本必要满足信息处理的需求，用户登录、注册、活动管理、报名管理等功能都能实现，符合预期，系统的性能完好，可靠性良好。
结 论
经过本论文的研究与分析，基于Django+React的疫情防控社区志愿者管理系统的设计实现取得了良好的效果，系统实现了疫情防控志愿者的招募、管理、统计、审核等功能，提高了志愿者的管理效率和任务完成率。在实际应用过程中，该系统得到了广泛的应用和认可，为疫情防控工作提供了有力的信息化支持。未来，可以在系统的基础上进一步完善功能和优化体验，为社区疫情防控工作提供更好的服务。
此外，本论文还对Django和React技术的特点和优势进行了分析和总结，对于开发类似系统的研究者具有一定的参考价值。其中，Django作为一种高效、易用、功能丰富的Web开发框架，能够快速开发出符合Restful风格的应用程序接口；React作为主流前端框架，能够提供高效、可重用、易维护的UI组件和极佳的用户体验。通过使用Django和React相结合的方式，可以更加高效地开发出功能丰富、易于维护和后期扩展的数字化Web应用管理系统。
总之，本论文的研究成果为疫情防控社区志愿者管理系统的开发提供了有力的支持和借鉴，对于推动信息化建设和社区防控工作具有一定的实际意义和价值。
在未来的研究中，还可以进一步完善该系统，例如增加更加精细化的权限控制、优化用户体验、增加更加实用的功能等。同时，我们也可以探索其他前后端技术的应用，例如Vue.js、Angular等，以提高Web应用的开发效率和性能。另外，我们可以将该系统应用到其他领域，例如社区服务、志愿者管理等，为社区的数字化建设和发展做出更多的贡献。最后，我们也可以将研究重心转移到其他领域，例如智慧城市、医疗健康等，以应用技术手段推动信息化建设的进步和发展。
参考文献
[1]王志伟.新冠疫情防控集中隔离管理系统的应用与实践[J].通信管理与技术,2022(02):27-31.
[2]曾丽娟,邱毅,段涛,李建水,唐啸龙,邓大炜.基于B/S架构医院志愿者管理系统的设计与实现[J].医学信息,2021,34(07):27-30+34.
[3]马英瑞,陈廉元,李娟,邹红,王冬梅. 基于Spring Boot的网上在线教育系统的设计与实现[J]. 电脑知识与技术,2021(08):55-60+70.
[4]刘振东. 威海志愿者管理系统的设计与实现[D].哈尔滨:哈尔滨工业大学,2019.
[5]杨仁波. 基于容器虚拟化技术的移动物联网仿真实验平台的设计与实现[D]. 内蒙古大学, 2021.
[6]蔡栋,金波.基于Fabric的区块链慈善业务管理系统设计[J].中国高新科技,2020(03):62-65+68.
[7]马春晓,叶青,吕明.志愿活动管理系统的设计与实现[J].工业控制计算机,2022,35(01):135-136+139.

[8]黄智霖.高校校园活动管理系统设计——以厦门华厦学院为例[J].信息技术与信息化,2021(12):72-74+78.
[9]李珊. 基于微信小程序的学生活动管理系统的设计与实现[D].广东:广东工业大学,2019.
[10]魏银平. 农业科技成果交易与服务云平台应用系统研究与开发[D]. 东南大学, 2018.
[11]刘新宇.疫情防控下高校志愿服务项目建设的路径探索[J].黑龙江教育(高教研究与评估),2022(10):90-92.
[12]金鑫,董耀众,张大伟,李伟良,肖磊,牟霄寒,孙建刚.基于移动应用的疫情防控管理系统的设计与构建——以电力企业为例[J].办公自动化,2022,27(18):6-9.
[13]金鑫,董耀众,张大伟,李伟良,肖磊,牟霄寒,孙建刚.基于移动应用的疫情防控管理系统的设计与构建——以电力企业为例[J].办公自动化,2022,27(18):6-9.
[14]张玉珠. 基于Vuejs框架的掌上零售系统的设计与实现[D]. 华中科技大学, 2019.
[15]Janhavi Desale,Kunal Gautama,Saish Khandare,Vedant Parikh,Dhanashree Toradmalle. NGO Support Software Solution: for effective reachability[J]. International Journal of Education and Management Engineering(IJEME),2020,10(6) 21-23+25.
[16]Noor Afiza Mat Razali,Nurjannatul Jannah Aqilah Md Saad,Hasmeda Erna Che Hamid,Muhammad Ramzul Abu Bakar,Khairul Khalil Ishak,Nor Asiakin Hasbullah,Norulzahrah Mohd Zainudin,Suzaimah Ramli, Norshahriah Wahab. Volunteer Management System for Disaster Management[J]. International Journal of Recent Technology and Engineering (IJRTE),2019,(7)51-54+57.
[17]. Deedia Inc.; Researchers Submit Patent Application, "Systems And Methods For Service Opportunity Management And Volunteer Management", for Approval (USPTO 20200142931)[J]. Politics & Government Week,2020(6)20-26+31.
[18]. InitLive; InitLive Donates Volunteer Management System To Aid In COVID-19 Relief[J]. Medical Letter on the CDC & FDA,2020(5)18-21+24.
[19]Jiang Qiwen,Zhu Xueyuan,Chen Lianghua,Zhao Ziyuan,Chen Yilong. Research on Time-Driven Activity-Based Management System of Public Hospitals [J]. Frontiers in Public Health,2022(9)20-26+31.
[20]Supun Wijekoon. University Activity Management System[J]. Journal of Information Technology & Software Engineering,2021,11(7)32-36+39.
致 谢
在本论文的完成过程中，我获得了许多人的支持和帮助，在此向他们致以最诚挚的谢意。
首先，我要感谢我的论文指导教师，她在整个研究过程中给予了我耐心的指导和关心，在研究方法、技术选型、论文撰写等方面提供了宝贵的意见和建议。
其次，我要感谢我的家人和朋友，在繁忙的学业生活中，他们一直支持我、鼓励我，给我以精神上的支持和帮助。我还要感谢我大学期间，计科教研室的老师们，给我传授了很多专业知识，才有了我今天的编程开发水平。
同时，我也要感谢我在实习期间公司的同事，在我毕业设计系统开发的过程中，给我提供了很大的帮助与支持，在我每次遇到技术难点和无法解决的bug时，他都会耐心的帮我解决，给我提供了很大的技术支持。
最后，我要感谢所有在疫情防控一线奋斗的医护人员和志愿者，是他们的无私奉献的精神，为我们创造了安全、舒适、健康的生活环境，让我们深刻认识到科技在疫情防控中的重要作用。
感谢以上所有人的支持和帮助，让我的论文能够顺利完成。
附 录
报名管理相关实现代码：
```
from django.http import HttpResponse
from rest_framework.generics import get_object_or_404
from rest_framework.views import APIView
from activity.models import Activity
from .models import Apply
from rest_framework.response import Response
from .serializers import ApplyModelSerializer
from utils.pagination import StandardPageNumberPagination
import sys, os
```

VPCS 维普论文检测系统
VIP PAPER CHECK SYSTEM

```
from utils.excel import *
class ApplyListAPIView(APIView):
queryset = Apply.objects.all()
serializer_class = ApplyModelSerializer
def get(self, request, *args, **kwargs):
response = {'success': True}
apply_list = Apply.objects.all()
total = apply_list.count()
apply_serializers = ApplyModelSerializer(apply_list, many=True, context={'request': request})
    pagination = StandardPageNumberPagination()
pg_data = pagination.paginate_queryset(queryset=apply_serializers.data, request=request, view=self)
response['data'] = pg_data
response['total'] = total
return Response(response)
def post(self, request):
data = request.data
    serializer = ApplyModelSerializer(data=data)
serializer.is_valid(raise_exception=True)
serializer.save()
response = {
'success': True,
'data': {
'message': 'success!'
}
}
return Response(response)
def delete(self, request, *args, **kwargs):
    delete_id = request.query_params.get('deleteId', None)
if not delete_id:
return Response({'message': '数据不存在!'})
for i in delete_id.split(','):
    get_object_or_404(Apply, pk=int(i)).delete()
response = {
'success': True,
'data': {
'message': '删除成功!'
},
}
return Response(response)
class ApplyDetailAPIView(APIView):
def get(self, request, pk):
try:
apply = Apply.objects.get(id=pk)
except Apply.DoesNotExist:
    return Response({'success': True, 'data': {'message': '数据不存在!'}})
    serializer = ApplyModelSerializer(instance=apply)
response = {
```

深圳国际商务学院

```
'success': True,
'data': serializer.data,
}
return Response(response)
def put(self, request, pk):
try:
apply = Apply.objects.get(id=pk)
except Apply.DoesNotExist:
return Response({'message': '数据不存在!'})
serializer = ApplyModelSerializer(instance=apply, data=request.data, partial=True)
serializer.is_valid(raise_exception=True)
serializer.save()
response = {
'success': True,
'data': {
'message': '更新成功'
},
}
return Response(response)
def delete(self, request, pk):
try:
apply = Apply.objects.get(id=pk)
except Apply.DoesNotExist:
return Response({'message': '数据不存在!'})
apply.delete()
response = {
'success': True,
'data': {
'message': '删除成功!'
},
}
return Response(response)
class ApplyExportExcelAPIView(APIView):
def post(self, request):
    apply_codes = request.data.get("apply_code")
n = len(apply_codes)
head_data = [u'报名编号', u'姓名', u'年龄', u'性别', u'地址', u'电话', u'报名活动', u'报名状态', u'申
请时间']
records = []
for apply_code in apply_codes:
if apply_code != "":
    apply_obj = Apply.objects.get(id=apply_code)
id = apply_obj.id
name = apply_obj.name
age = apply_obj.age
    sex = '未知' if apply_obj.sex == 0 else '男' if apply_obj.sex == 1 else '女'
address = apply_obj.address
```

浙江国际商贸学院

```
tel = apply_obj.tel
    apply_status = '待审核' if apply_obj.apply_status else '已审核' if apply_obj.apply_status == 1
else '未通过'
    apply_time = apply_obj.apply_time.strftime("%Y-%m-%d %H:%M:%S")
    belonging_activity_id = apply_obj.belonging_activity_id
    apply_activity = Activity.objects.get(id=belonging_activity_id).name
record = []
record.append(id)
record.append(name)
record.append(age)
record.append(sex)
record.append(address)
record.append(tel)
record.append(apply_activity)
record.append(apply_status)
record.append(str(apply_time))
records.append(record)
cur_path = os.path.abspath('.')
download_url = cur_path + '\\upload\\'
    ret = write_to_excel(n, head_data, records, download_url)
return HttpResponse(ret)
def get(self, request):
applys = Apply.objects.all()
n = len(applys)
head_data = [u'报名编号', u'姓名', u'年龄', u'性别', u'地址', u'电话', u'报名活动', u'报名状态', u'申
请时间']
records = []
for apply_obj in applys:
id = apply_obj.id
name = apply_obj.name
age = apply_obj.age
    sex = '未知' if apply_obj.sex == 0 else '男' if apply_obj.sex == 1 else '女'
address = apply_obj.address
tel = apply_obj.tel
    apply_status = '待审核' if apply_obj.apply_status else '已审核' if apply_obj.apply_status == 1
else '未通过'
    apply_time = apply_obj.apply_time.strftime("%Y-%m-%d %H:%M:%S")
    belonging_activity_id = apply_obj.belonging_activity_id
    apply_activity = Activity.objects.get(id=belonging_activity_id).name
record = []
record.append(id)
record.append(name)
record.append(age)
record.append(sex)
record.append(address)
record.append(tel)
record.append(apply_activity)
```

陕西国际商贸学院

VPCS 维普论文检测系统
VIP PAPER CHECK SYSTEM

```
record.append(apply_status)
record.append(str(apply_time))
records.append(record)
cur_path = os.path.abspath('.')
download_url = cur_path + '\\upload\\'
    ret = write_to_excel(n, head_data, records, download_url)
return HttpResponse(ret)
```

报名管理相关实现代码：

```
from .serializers import ActivityModelSerializer
from utils.pagination import StandardPageNumberPagination
import sys, os
from utils.excel import *
class ActivityListAPIView(APIView):
queryset = Activity.objects.all()
serializer_class = ActivityModelSerializer
def get(self, request, *args, **kwargs):
response = {'success': True}
activity_list = Activity.objects.all()
    paging_status = request.GET.get("pagingStatus")
for activity_item in activity_list:
    apply_activity = Activity.objects.get(id=activity_item.id)
    apply_person_num = apply_activity.apply_set.all().count()
    pass_person_num = apply_activity.apply_set.filter(apply_status=1).count()
    Activity.objects.filter(id=activity_item.id).update(
    apply_person_num=apply_person_num, pass_person_num=pass_person_num)
total = activity_list.count()
activity_serializers = ActivityModelSerializer(activity_list, many=True, context={'request':
request})
    pagination = StandardPageNumberPagination()
pg_data = pagination.paginate_queryset(queryset=activity_serializers.data, request=request,
view=self)
if paging_status == 'false':
response['data'] = activity_serializers.data
else:
response['data'] = pg_data
response['total'] = total
return Response(response)
def post(self, request):
data = request.data
    serializer = ActivityModelSerializer(data=data)
serializer.is_valid(raise_exception=True)
serializer.save()
return Response({
'success': True,
'data': {
'message': '创建成功!'
}
```

```
})
def delete(self, request, *args, **kwargs):
    delete_id = request.query_params.get('deleteId', None)
if not delete_id:
return Response({'message': '数据不存在！'})
for i in delete_id.split(','):
    get_object_or_404(Activity, pk=int(i)).delete()
response = {
'success': True,
'data': {
'message': '删除成功！'
},
}
return Response(response)
class ActivityDetailAPIView(APIView):
def get(self, request, pk):
try:
activity = Activity.objects.get(id=pk)
    apply_person_num = activity.apply_set.all().count()
pass_person_num = activity.apply_set.filter(apply_status=1).count() Activity.objects.filter
(id=activity.id).update(apply_person_num=apply_person_num, pass_person_num=pass_person_num)
except Activity.DoesNotExist:
    return Response({'success': True, 'data': {'message': '数据不存在！'}})
    serializer = ActivityModelSerializer(instance=activity)
response = {
'success': True,
'data': serializer.data,
}
return Response(response)
def put(self, request, pk):
try:
activity = Activity.objects.get(id=pk)
except Activity.DoesNotExist:
return Response({'message': '数据不存在！'})
serializer = ActivityModelSerializer(instance=activity, data=request.data, partial=True)
serializer.is_valid(raise_exception=True)
serializer.save()
response = {
'success': True,
'data': {
'message': '更新成功'
},
}
return Response(response)
def delete(self, request, pk):
try:
activity = Activity.objects.get(id=pk)
```

陕西国际商贸学院

```
except Activity.DoesNotExist:
return Response({'message': '数据不存在！'})
activity.delete()
response = {
'success': True,
'data': {
'message': '删除成功！'
},
}
return Response(response)
```

相似片段说明

相似片段中"综合"包括：《中文主要报纸全文数据库》《中国专利特色数据库》《中国主要会议论文特色数据库》《港澳台文献资源》《图书资源》《维普优先出版论文全文数据库》《年鉴资源》《古籍文献资源》《IPUB原创作品》

须知
1、报告编号系送检论文检测报告在本系统中的唯一编号。
2、本报告为维普论文检测系统算法自动生成，仅对您所选择比对资源范围内检验结果负责，仅供参考。

客服热线：400-607-5550、客服QQ：4006075550、客服邮箱：vpcs@fanyu.com
唯一官方网站：https://vpcs.fanyu.com

关注微信公众号