

# 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: 199eb03333h19484

检测时间: 2023-04-04 20:00:26

检测字符数: 15324

作者姓名: 李佳音

所属单位: 陕西国际商贸学院

检测结论: 全文总相似比 = 复写率 + 他引率 + 自引率 + 专业术语  
**22.41%** = **22.41%** + **0.0%** + **0.0%** + **0.0%**

其他指标: 自写率: 77.59%

高频词: 系统, 用户, 信息, 进行, 可以

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 118  
期刊: 5 博硕: 59 综合: 0  
外文: 0 自建库: 8 互联网: 46

检测范围: 中文科技期刊论文全文数据库  
博士/硕士学位论文全文数据库  
外文特色文献数据全库  
高校自建资源库  
个人自建资源库

中文主要报纸全文数据库  
中国主要会议论文全文数据库  
维普优先出版论文全文数据库  
图书资源  
年鉴资源

中国专利特色数据库  
港澳台文献资源  
互联网数据资源/互联网文档资源  
古籍文献资源  
IPUB原创作品

时间范围: 1989-01-01至2023-04-04

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）

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

二〇二三年三月

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。

本文介绍了一种基于DRF框架和React框架的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, 本文提出了一种基于DRF框架和React框架的疫情防控社区志愿者管理系统的设计方案, 并详细介绍了系统的功能模块及其实现方法。系统的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, 本文对系统中的所有功能模块进行了系统测试、功能测试和接口测试, 根据测试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.

This paper introduces a community volunteer management system for epidemic prevention and control based on DRF framework and React framework, 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 proposes a design scheme of community volunteer management system for epidemic prevention and control based on DRF framework and React framework, and introduces the functional modules and implementation methods of the system in detail. The main functions of the system include user management, activity management, registration information management and friendship link management. Finally, this paper conducts system test, function test and interface test for all functional modules in the system. According to the analysis of test results, it can be concluded that this 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

## 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,科学有序助力社区疫情防控工作,携手筑牢疫情防线,巩固疫情防控成果,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。

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和志愿意识比较强烈。

但就当时的经济水平和计算机发展状况而言,没有互联网服务的志愿者管理要想有高效率是十分局限的。志愿活动组织者很难将自己活动及时宣传让大众都知道,也难以第一时间通知志愿者参与活动。并且,当时计算机十分昂贵,普遍设计的管理系统功能匮乏,只能简单地记录活动信息,实现并普及志愿服务管理系统是比较困难的事情。随着国外志愿服务组织的发展和大型志愿活动等的开展,国外志愿者相关政策的发展也促使着志愿服务的条件

日趋成熟，志愿服务管理系统设计的渐渐完善。

### 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。对每个子系统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。

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。拥有很多第三方库，可以实现各种业务需求和功能。Python支持多种编程范式，可以满足不同的开发需求。Python拥有强大的可移植性，可以在多种操作系统上运行；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，易于安装和维护；支持多种存储引擎，可以根据不同的应用场景选择合适的存储引擎；支持多种编程语言，可以使用多种编程语言来操作 MySQL 数据库；支持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。

### 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：

易于使用：DRF框架具有易于使用的API，可以轻松地构建RESTful API。

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。

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。

可读性：DRF框架具有可读性，可以轻松地调试和维护代码。

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。

React拥有强大的功能和优点：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

Ant Design Pro是由蚂蚁开发团队推出的一套企业级前端设计解决方案，主要用于中台和后台产品的搭建，采

用umiJS和 Ant Design二次封装，在追求开发规范和组件设计的基础上，不断构建。

Ant Design Pro致力于给开发人员提供开箱即用的体验，提供国际化配置、用户权限、mock数据、网络请求和完整的脚手架，为中后台开发人员提供了最佳实践来减少学习和开发成本。同时为了提供更加高效的开发体验，Ant Design Pro提供了一系列模板组件，ProLayout, ProTable, ProList 都是开发中后台的好帮手，可以显著的减少样板代码。

## 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经济层、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，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

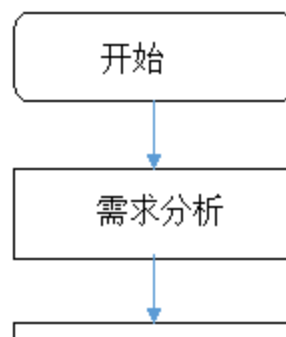
现在已有的管理系统对使用者来说都是很简单方便，该系统也例外，根据使用者的角度，该系统在开发上致力于简化结构，使页面尽可能简洁，优化用户操作，使用者只需要在浏览器中打开系统并进行登录就可以进行使用操作系统。综上所述，该系统的操作简单，方便，可以进行开发。

## 3.2 系统流程分析

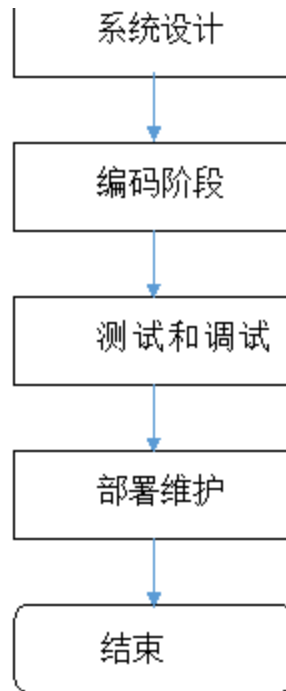
### 3.2.1 系统开发流程

系统的开发流程从以下几个阶段进行分析：

1. 需求分析阶段：确定系统的功能模块，以及子功能模块的划分。
2. 系统设计阶段：设计系统的前后端开发框架、子功能模块、数据结构、接口设计风格和等。
3. 编码阶段：根据开发计划进行编码实现。
4. 测试和调试阶段：使用Postman测试工具，对系统的所有接口进行自动化测试，确保本系统所有功能模块的可靠性。
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。







系统开发的流程图如图3.1所示：

图 3.1 系统开发流程图

### 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。

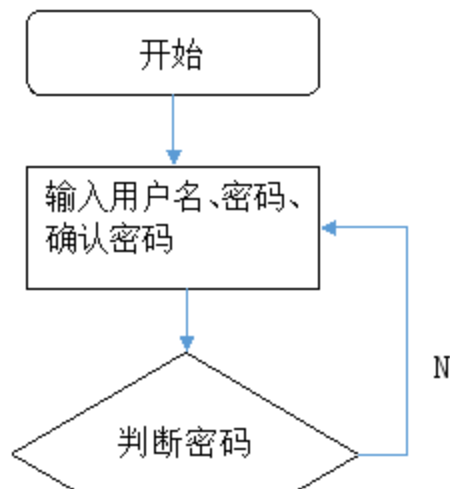
用户注册流程分析：

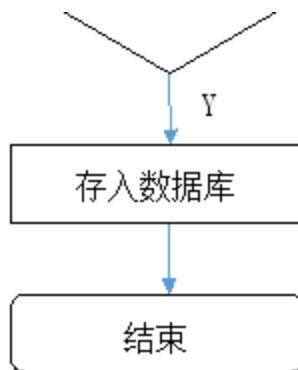
1. 在登录页面切换到注册表单：用户进入登录页面，切换到注册表单。
2. 填写注册信息：根据表单提示，填写用户名、密码、确认密码等，确保两次密码一致。
3. 校验信息：前端进行表单校验，后端进行数据校验，确保信息的正确性和完整性。
4. 返回结果：前端根据后端返回的响应结果，弹出提示框，告知用户是否注册成功。
5. 操作完成：用户根据提示信息进行操作，注册成功，则登录系统，注册失败，则根据失败的原因重新进行注册。

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. 数据校验：前、后端要对注册信息进行表单校验和数据校验，确保数据的正确性和完整性。
3. 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。
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。

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。





注册的流程图如图3.2所示：

图 3.2 注册流程图

### 3.2.3 用户登录流程

数据库的安全问题是非常重要的，关系到用户的隐私。为了防止用户信息数据泄露，数据库对用户的敏感信息进行加密，用户要进行登录才能使用系统。用户登录的流程图如图3.3所示。

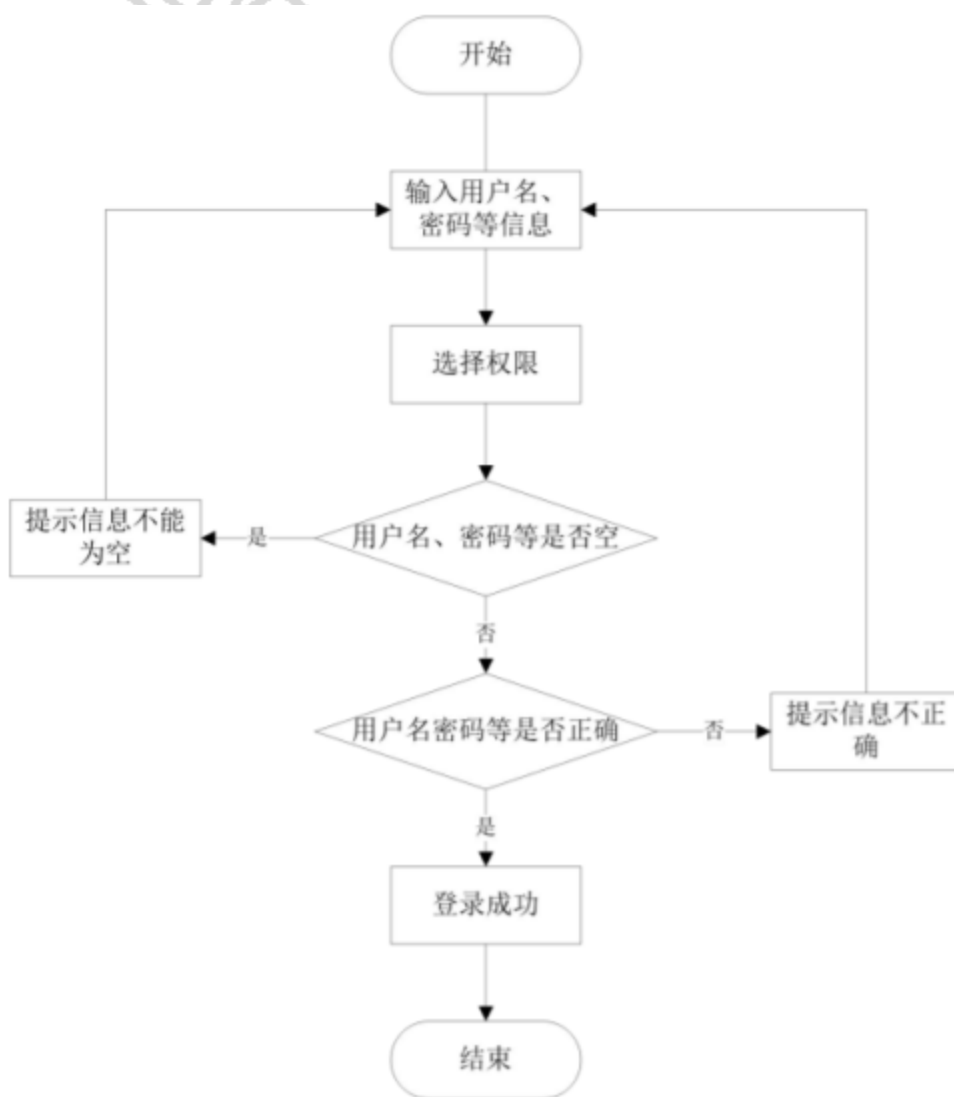


图 3.3 用户登录流程图

### 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：用户需要输入账号和密码等信息，登录系统后才能进行后续的操作。
2. 界面操作：为了方便用户快速定位和操作所需要的功能，应该熟悉系统的操作方式和布局，包括状态栏、弹框、菜单等。

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。用户查看返回的数据信息，以满足自己的业务需求。

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：系统收到用户提交的信息后，会进行相应的处理，包括校验数据、保存数据等操作。
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：系统收到用户提交的修改信息后，会进行相应的处理，包括校验数据、保存数据等操作。
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：由于数据一旦删除就无法恢复，系统会弹出确认删除对话框，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。

4. 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。

5. 系统响应：处理完信息，系统返回响应结果。

6. 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。

管理员用例分析：

1. 登录系统：

管理员角色拥有系统的所有的权限，需要登录后才可以对后台进行管理。

2. 管理用户：

管理员角色可以管理系统用户，为确保用户账号可以正常使用，管理员角色可以进行添加用户、重置密码、修改用户、注销用户等操作。

3. 管理信息：

管理员角色可以管理全部功能模块中的信息，可以进行添加、修改、删除信息等操作，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。

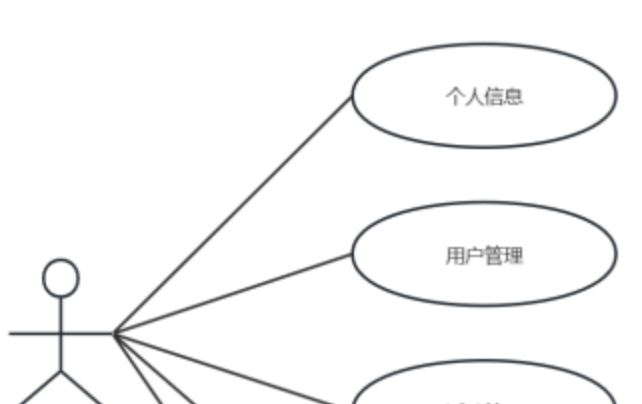
2. 安全保障：系统应该采取相应的安全措施，例如密码安全策略、数据备份策略等，确保系统的安全性和可靠性。

3. 异常处理：系统需要对异常情况进行处理，例如系统故障、数据错误等情况。

4. 数据管理：确保用户信息和系统信息的准确性和完整性，管理员需要对用户和信息进行管理。

通过合理的管理员用例分析，可以提高系统的管理效率和安全性，满足用户的需求和要求。

管理员用例如图 3.4 所示：



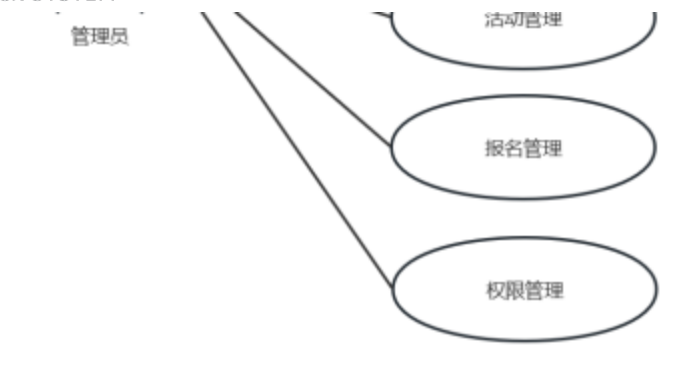


图 3.4 管理员用例图

### 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。

发布企业用例如图 3.5 所示：

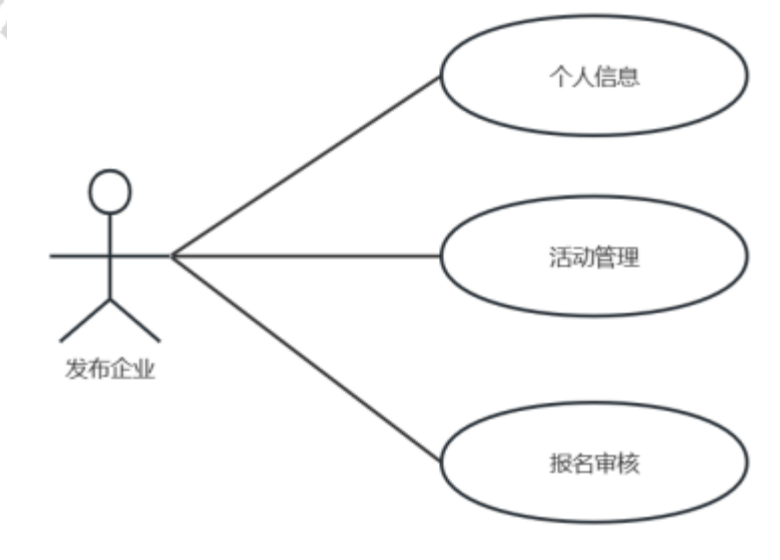


图 3.5 发布企业用例图

### 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3.6所示：

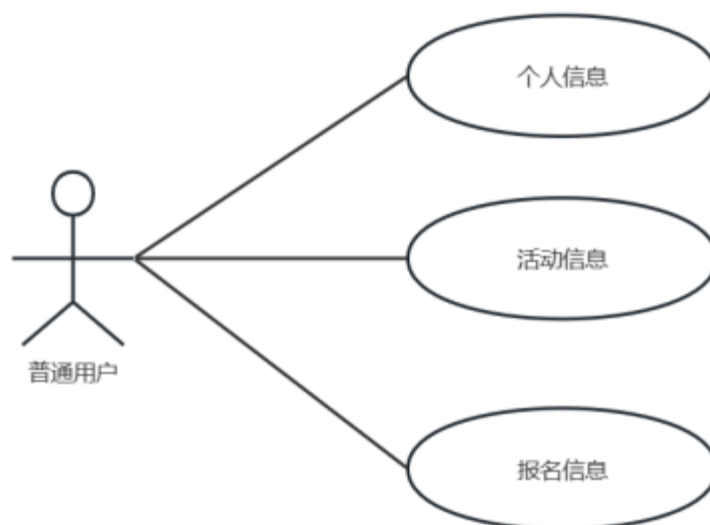


图 3.6 普通用户用例图

#### 4 系统设计

##### 4.1 系统结构设计

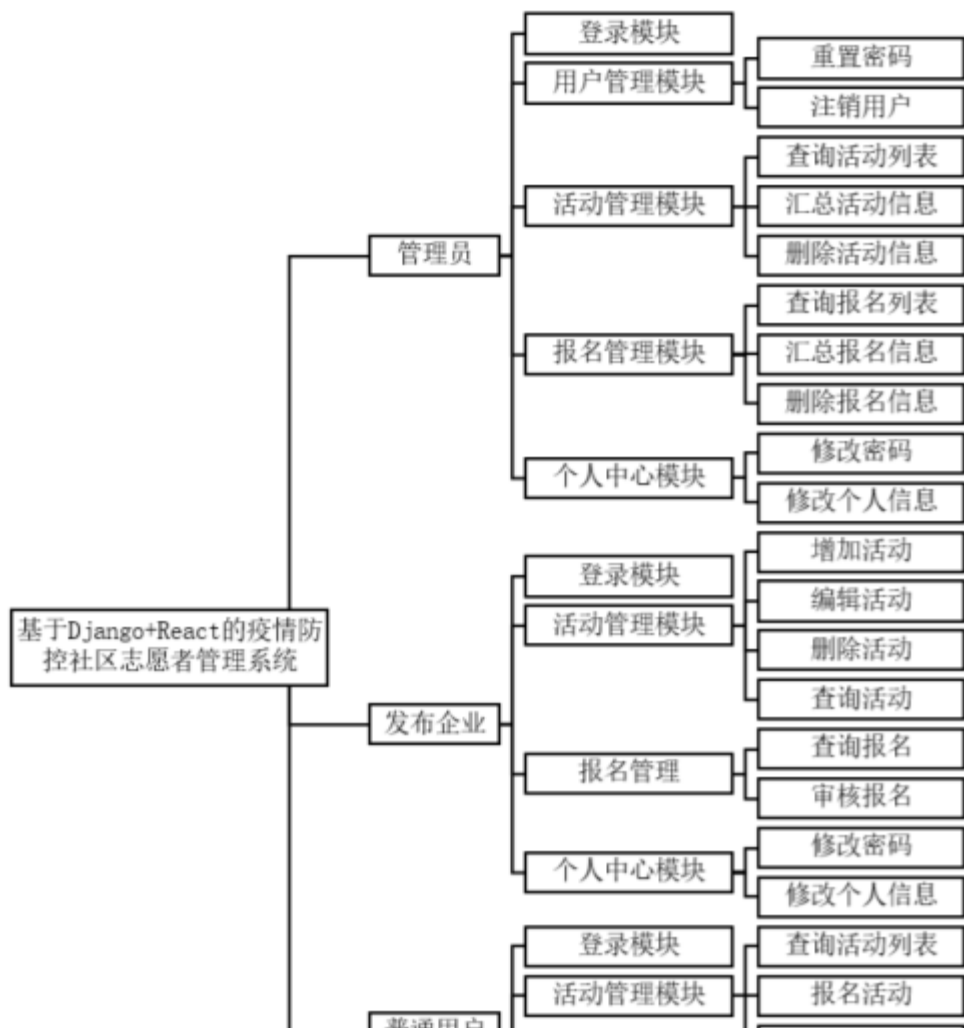
系统结构设计是指基于系统需求和系统功能，设计出系统的整体结构框架，包括系统的模块、组件、接口、数据流等。系统结构设计的目的是为了将系统划分为不同的部分，使得系统的开发、测试、维护和升级都更加容易和高效。

系统结构设计包括以下几个步骤：

1. 确定系统需求：确定系统需求，以便于后续的设计。
2. 划分子系统：对系统的功能模块进行划分，划分为多个独立的子系统，确定每个子系统的功能模块，子系统之间要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。
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。
7. 进行系统测试：根据系统结构设计，进行测试，确保系统能够达到预期的结果。

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：



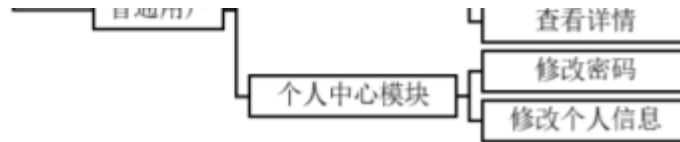


图 4.1 系统功能模块

## 4.2 数据库设计

数据库设计是指在遵循数据库理论的基础上，根据业务需求和数据特点，设计出合理的数据结构和数据关系，以便于数据的存储。数据库的设计，从以下几个流程进行：

1. 需求分析：从数据类型、关联关系、数据结构等多个方面确定数据库的数据特征。
2. 概念设计：根据数据库的需求分析，从实体、关系、属性等方面设计其的概念模型。
3. 逻辑设计：根据数据库概念模型，从表结构、字段约束、关联关系、属性、等方面设计其逻辑模型。
4. 物理设计：根据数据库的逻辑设计，设计出物理模型，包括表空间、索引、分区等。
5. 实现：根据物理模型，创建数据库和表。

在设计数据库时，要确保数据安全，以防造成数据泄露。同时，需要遵循数据库设计的一些原则，如实体完整性、关系完整性、范式化等，以便于数据的管理和使用。

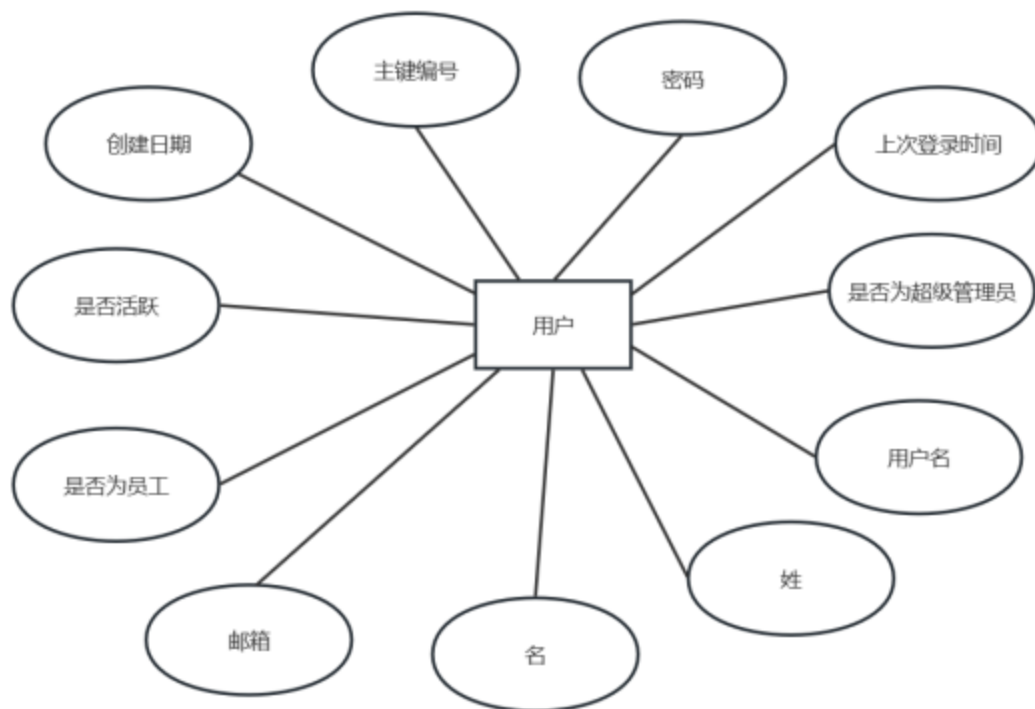
### 4.2.1 数据库设计原则

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，确定表与表之间的关系，确保两个表的数据能够正确对应。
4. 数据访问安全原则：为了防止数据库重要信息泄露，对数据库中的敏感字段，如用户密码等，进行加密，保证用户的信息安全。
7. 数据库备份与恢复原则：为了防止数据库数据丢失等隐患，对系统的数据库进行备份。

### 4.2.2 概念模型设计

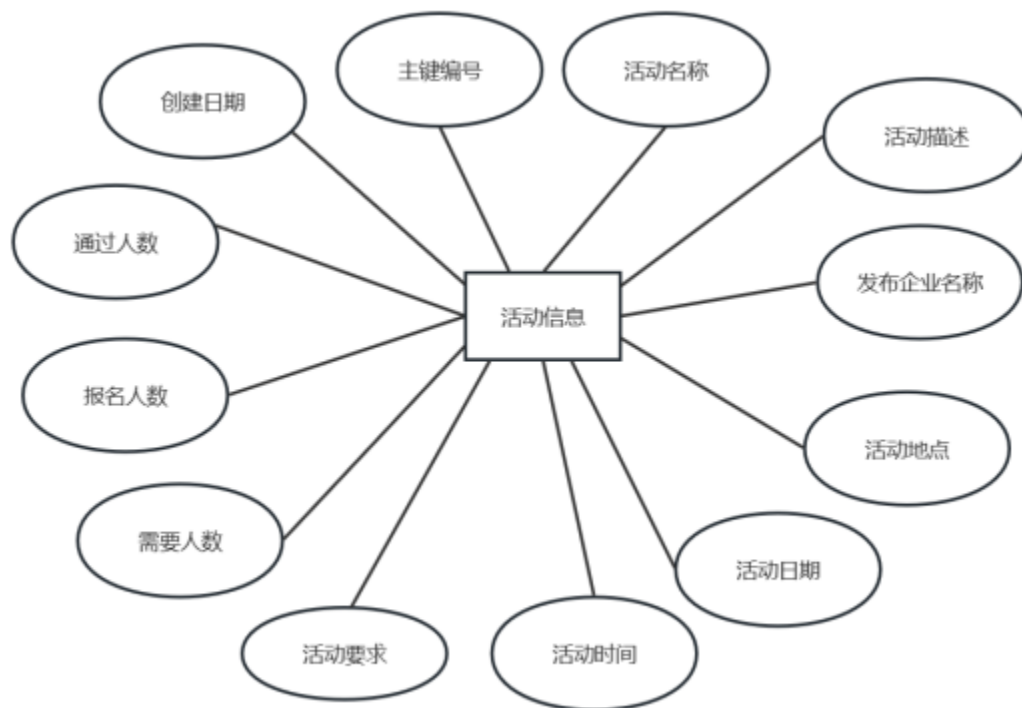
本系统有用户、报名信息、活动信息等多个实体，通过 E-R图来体现数据库的概念模型设计，下面分别是各实体和整体的 E-R 图。



(1) 用户实体图如图4.2

所示:

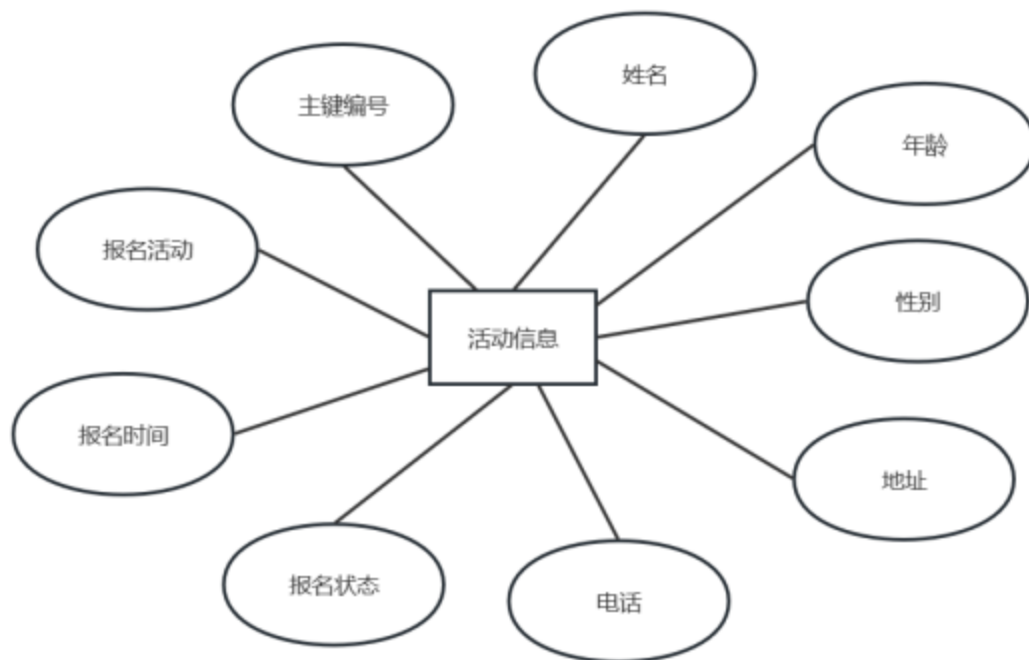
图 4.2 用户ER图



(2) 活动信息实体图如图

4.3所示:

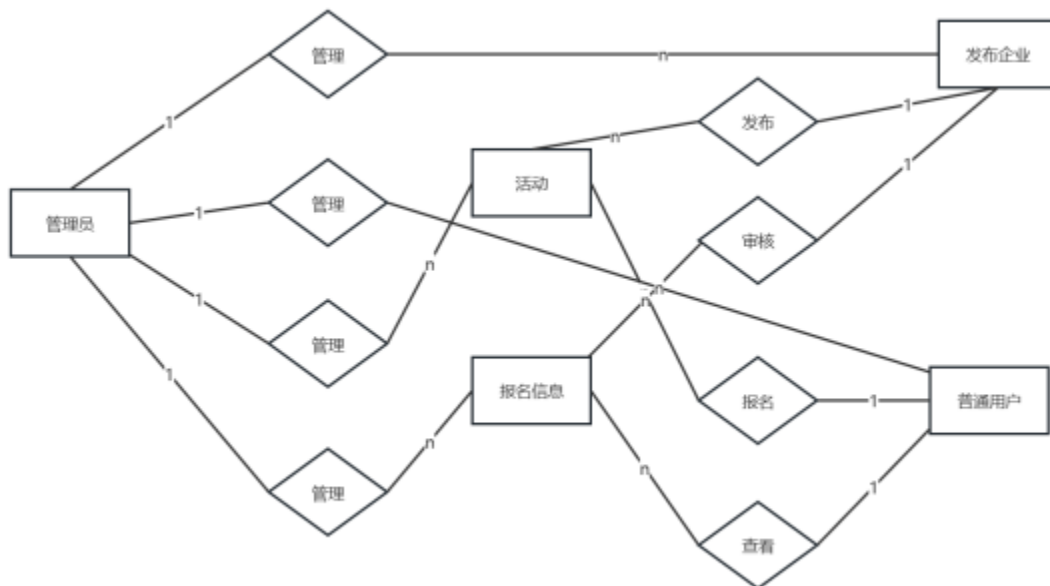
图 4.3 活动信息ER图



(3) 报名信息的实体图

如图4.4所示:

图 4.4 活动信息ER图



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图如图4.5所示:

图 4.5 全局E-R图

#### 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4.1所示:

表 4.1 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4.2所示：

表 4.2 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
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4.3所示：

表 4.3 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4.4所示：

表 4.4 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4.5所示:

表 4.5 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4.6所示:

表 4.6 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4.7所示:

表 4.7 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4.8所示:

表 4.8 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,在登录表单根据表单提示填写并提交登录信息,前端向后端发送请求,将用户的登录信息传给后端,后端接受前端发送的数据,并根据数据库中的用户信息表进行校验,然后将结果响应给前端。

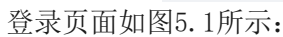


图 5.1 登录页面

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:




### 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：



文件导出部分实现代码如下：

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的设计和的使用方法。


 swagger

# Volunteer Management System API

[ Base URL: [http://127.0.0.1:3000](#) ]

Schemes

HTTP

Authorize 

activity

GET/activity/ 查看所有活动信息

POST/activity/ 新增一条活动信息

DELETE/activity/ 删除活动

GET/activity/export\_excel/ 导出活动信息表

POST/activity/export\_excel/ 批量导出活动信息表

POST/activity/delete/ 批量删除活动信息

## 5.2 普通用户模块

### 5.2.1 活动列表模块

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模块有两个，活动管理和报名管理。公益企业用户根据自身需求，组织公益活动，招募志愿者。公益活动发布后，普通用户可以进行报名，公益企业在报名管理界面对志愿者的报名信息进行审核。

### 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：

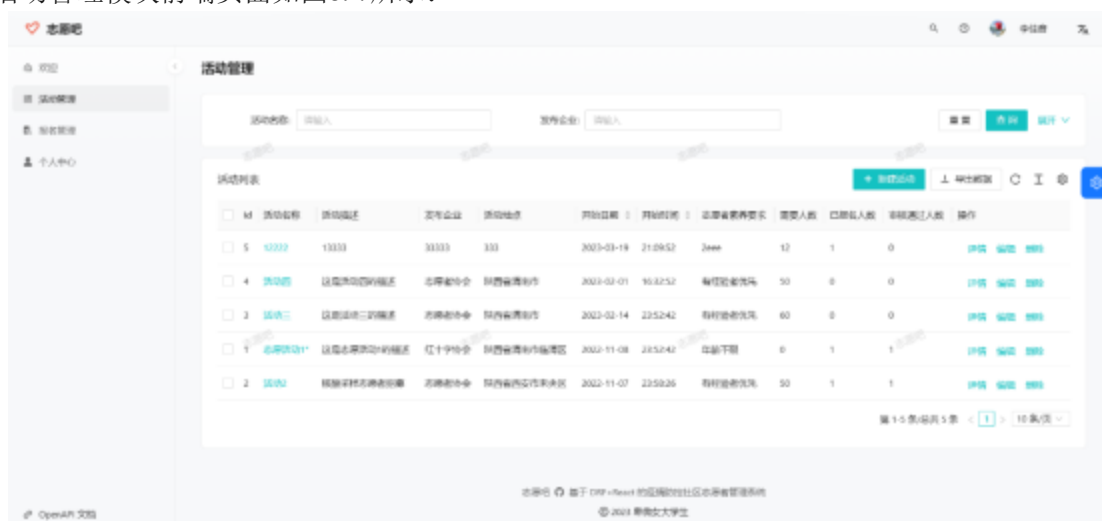


图 5.7 活动管理页面

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：



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

管理员能够对系统所有用户进行管理，包括添加用户、查看用户列表、修改用户、注销用户、批量删除、批量导出等操作。对于忘记密码的用户，用户联系管理员后，管理员可以对其进行密码初始化，用户密码重置为123456。用户使用初始密码登录系统后，可自行修改密码。管理员可以注销违反规则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

## 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

通过常用的API测试工具Postman对本系统的所有API进行测试。Postman的优点在于可以对API进行快速、方便的测试和调试。在Postman中，创建一个新的请求，输入请求URL，选择请求方法，添加请求参数。在发送请求后，Postman会自动处理请求并显示响应结果。请求结束后，通过Postman的界面来查看响应结果，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：

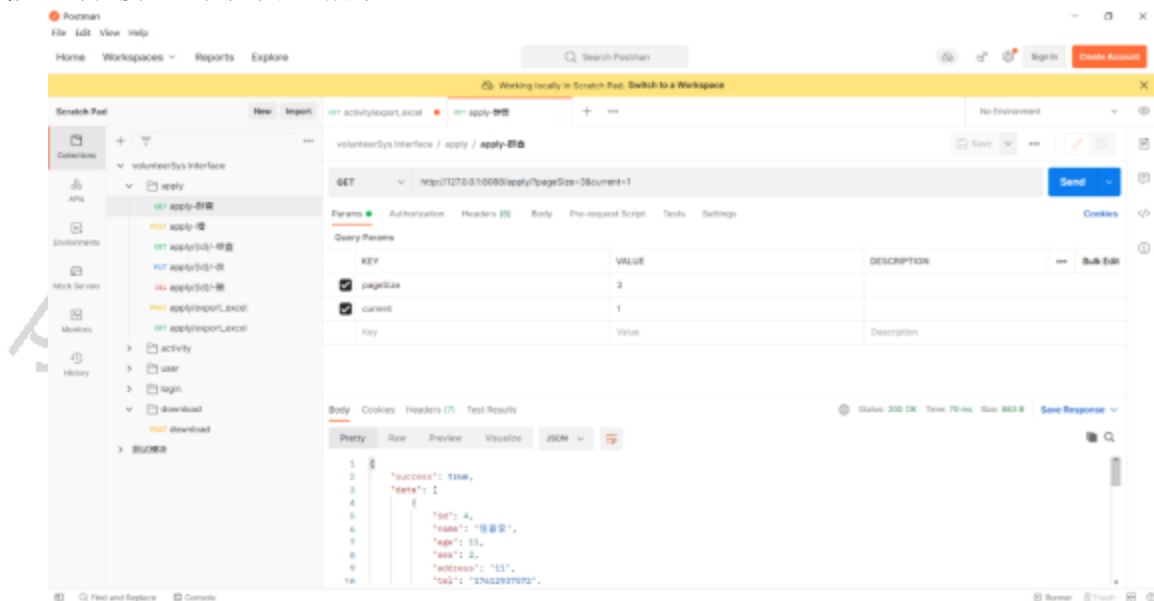
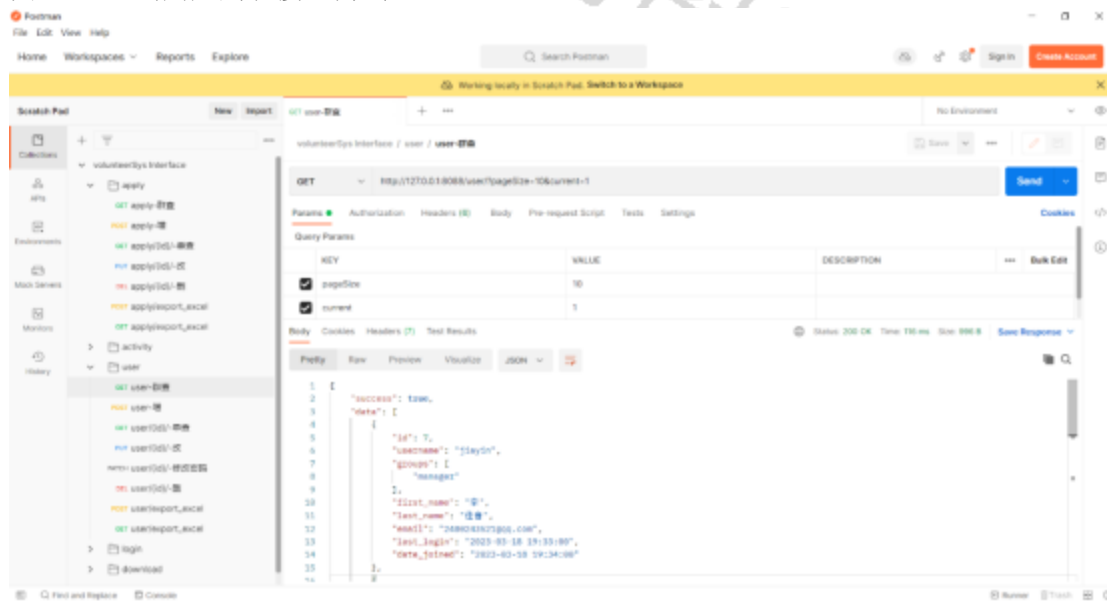


图 6.1 查询报名列表接口测试

图 6.2 查询用户列表接口测试



测试查询用户列表接口过

程如图6.2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。

### 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