# 目 录

1. **绪 论** 1
   1. 研究的背景及意义 1
      1. 选题的背景 1
      2. 国内外研究现状 1
      3. 研究的意义 2
   2. 系统目标 3
2. **需求分析** 4
   1. 业务需求 4
      1. 主要业务流程 4
   2. 功能需求 6
      1. 角色分析 6
      2. 功能需求 6
   3. 非功能需求 17
      1. 环境需求 17

2.3.2 性能需求 18

2.3.3 安全需求 18

1. **总体设计** 19
   1. 系统设计的原则 19
   2. 系统体系结构设计 19
   3. 系统功能结构设计 20
2. **数据库设计** 22
   1. 概念结构设计 22

4.1.1 设计思路 22

4.1.2 E-R 图 22

* 1. 逻辑结构设计 27

4.2.1 设计思路 27

4.2.2 逻辑模型 27

* 1. 物理结构设计 31

4.3.1 存取方式 31

4.3.2 存储结构 31

1. **界面设计** 32
   1. 界面关系图或工作流图 32
   2. 界面设计成果 33

5.2.1 主界面 33

5.2.2 子界面 35

1. **详细设计** 38
   1. 系统主要功能模块介绍 38
   2. 用户管理 39
      1. 功能结构设计 39

6.2.2 类图设计 39

* + 1. 顺序图设计 40
    2. 流程图设计 41
  1. 新闻资讯管理 42
     1. 功能结构设计 42

6.3.2 类图设计 42

* + 1. 顺序图设计 44
    2. 流程图设计 44
  1. 政策文件管理 45
     1. 功能结构设计 45

6.4.2 类图设计 46

* + 1. 顺序图设计 47
    2. 流程图设计 47
  1. 活动发布管理 48
     1. 功能结构设计 48

6.5.2 类图设计 48

* + 1. 顺序图设计 50
    2. 流程图设计 50
  1. 活动审批管理 50
     1. 功能结构设计 50

6.6.2 类图设计 51

* + 1. 顺序图设计 52
    2. 流程图设计 52
  1. 活动报名管理 53
     1. 功能结构设计 53

6.7.2 类图设计 53

* + 1. 顺序图设计 54
    2. 流程图设计 55
  1. 建议反馈管理 55
     1. 功能结构设计 55

6.8.2 类图设计 56

* + 1. 顺序图设计 57
    2. 流程图设计 58
  1. 爱心项目管理 58
     1. 功能结构设计 58

6.9.2 类图设计 58

* + 1. 顺序图设计 60
    2. 流程图设计 60

**7 编码** 61

* 1. 代码实现与核心算法 61
  2. 代码优化分析 68

[8 测试 69](#_TOC_250007)

* 1. 测试方案设计 69

8.1.1 测试策略 69

8.1.2 测试进度安排 69

8.1.3 测试资源 70

8.1.4 关键测试点 70

* 1. 测试用例构建 70
     1. 测试用例编写约定 70
     2. 测试用例设计 70
     3. 关键测试用例 72
     4. 测试用例维护 72

1. [总结与展望 73](#_TOC_250006)
   1. 设计工作总结 73
   2. 未来工作展望 73

[谢 辞 74](#_TOC_250005)

[参考文献 75](#_TOC_250004)

[附录 A 外文翻译—原文部分 77](#_TOC_250003)

[附录 B 外文翻译—译文部分 84](#_TOC_250002)

[附录 C 软件使用说明书 90](#_TOC_250001)

[附录 D 主要源代码 92](#_TOC_250000)

# 1 绪 论

### 研究的背景及意义

* + 1. 选题的背景

随着我国经济的快速发展、社会不断地进步，志愿服务活动也逐渐蓬勃发展。青年一代的重要组成部分包括大学生，大学生也在志愿服务中扮演着越来越重要的角色，承担着越来越重要的责任。从个人活动、小规模的志愿服务到重大会议、赛事文化交流活动等不同规模、不同领域的各项志愿服务，现在也进一步朝着各年龄段、各行业群体积极参与的社会活动，包括赛事文艺活动的志愿服务、社区居民生活的志愿服务、应急救援的志愿服务到各行各业、特色鲜明的志愿服务等，推进志愿服务的建设对于我国社会和谐稳定、良好风气的形成有着十分重要的作用。志愿服务活动所涉及的行业、领域不断扩展以及志愿活动对所需志愿者的不断增长，负责人管理志愿服务的相关事务也不是一件简单的事，志愿服务的特殊和复杂性在实施流程中往往会面临着很大的压力，同时， 各种各样的问题也随时可能出现在志愿服务相关管理的任何过程中。志愿服务行业越来越多的人群和力量加入其中，灵活、方便也日益成为时代下的主流步伐、节奏，让爱心随时掌握在人们手中，通过志愿服务平台让形式多样的志愿项目活动、资讯等宣传到位、信息畅通、顺利开展可以发展社会事业、解决社会矛盾。

大学生作为青年志愿服务的重要主体，在志愿者队伍中发挥着关键的作用，同时，

高校作为青年志愿者服务的主要载体，应建立规范、高效、功能实用的志愿服务系统和符合行业特点的志愿服务体系，以进一步发挥高校青年志愿者服务的重要作用。大学生志愿者可以通过参加各项的服务活动中更加深入了解国情、社区和学校的情况，进一步培养大学生热爱社会、热爱祖国的高尚情感。当今社会对高校大学生在某些重要方面的要求和评价能够从他们参与志愿服务中体现，不断思考以及正确评判自己的优势和不足， 在参与志愿服务的实践中学习和探索，进而提升自己。学以致用，在不断的参加各种志愿服务中提高自己的实际应用能力，同时在志愿服务中学习各个方面的知识，提高志愿者的思维能力，表达技巧，动手操作的应用、协调配合、团结合作和其他各个方面的能力。大学生志愿者参加志愿服务是一项重要的实践活动，有利于学生学习、成长，锻炼实际动手能力，活动的多样性和活动空间范围的变化性，以及志愿者服务队伍规模庞大， 对志愿者工作的管理和志愿服务的实施正变得越来越复杂，建立志愿服务平台能够加快服务体系的建设和提高管理效率。

* + 1. 国内外研究现状

当前，我国志愿服务事业呈蓬勃发展、稳步向前的趋势，志愿服务类型的多样、涉及领域的广泛以及主体的多元性，志愿服务也逐渐地融入更多群体的日常生活之中，志愿服务也在一座文明城市的建设中充当着不可或缺的组成部分，志愿服务是衡量一座城

市文明、美丽的一项重要指标，依托高校搭建的志愿服务平台一方面有利于处理学校内的项目活动，另一方面也为城市的发展贡献力量，志愿服务工作也会在以后发挥更加关键和重要的作用。为更好地管理志愿服务工作需要建立更加友好的志愿服务体系并通过志愿服务平台宣传、展示，以确保志愿服务的顺利开展。国家加强对公民的道德培养以及维护社会稳定的有效形式之一是开展各行各业的志愿服务活动，大力宣传志愿者活动， 吸引高校志愿者以更加积极的态度参加志愿活动、公益项目也都需要志愿服务平台的建设和完善。现在国内志愿服务产业无论是在志愿者的人数上，还是志愿服务所涉及的领域数量上都取得了快速的发展，志愿服务事业迅速发展的一个重要力量是越来越多的大学生志愿者积极参加各项志愿服务，大学生群体作为主要力量承担志愿服务事业进一步发展的责任也越来越重要。

同时，我国的志愿活动流行的比较晚，经验相对不足，受各种因素的影响和制约， 目前在很多高校的志愿者服务的管理和实施上还存在着一些不足之处以及面临着一些困难：大学生群体对志愿服务的很多方面认识不够充分，了解程度不够深入；高校志愿服务机制和管理工作不够健全，需要进一步完善，缺乏一定的规范化，特色化，需要加强在很多方面制度化的严格管理和建立符合高校特点的志愿服务系统。

国外的大学和相关机构很早就开始对志愿服务活动、志愿服务事业进行探索和研究， 通过长期的发展积累了丰富的经验和比较健全的机制，外国学者、研究人员对大学生这一重要志愿者群体进行了全面和系统的研究。如研究志愿者群体和志愿者活动，志愿者动机的研究，高校志愿服务机制和服务的研究。国外志愿服务活动的开展和形式都非常普遍、发达，进行得十分活跃，广泛群众参与的普及和社会信誉良好的支撑，使得志愿服务已进入比较成熟的规范化，系统化管理，也形成了比较规范、完整的管理方法和运作机制。加强公民道德的修养，促进社会进步的有效形式是这些国家通过开展志愿服务来得以具体实现，广大公民的自觉行动包括参加社区、学校等各个方面、行业的志愿服务活动，其中，大多数公民都乐意接受并付诸行动来积极参加志愿服务，越来越多的国家政府高度重视志愿服务所带来的社会发展和效益。

* + 1. 研究的意义

随着志愿服务活动种类越来越丰富，规模越来越大，都需要招募大量的志愿者，同时志愿者也是促进城市文明、健康的重要群体和高校举办赛事活动、文化交流的不可或缺的群体，也是政府、高校、行业等充分调度资源、凝聚人心力量的有效形式，为方便志愿服务建立的志愿服务平台提供系统化、集中化管理，方便志愿者、志愿服务发布者、管理员能及时、高效的获得相应的服务、管理，让更多人更方便地了解志愿服务事业， 更加积极参加活动、传递爱心。不同人群在参与志愿服务行业中也需要一个平台来对志愿者群体、志愿服务活动等进行规范、统一的操作管理。整合资源，促进志愿服务体系建设，志愿服务信息及时分享，志愿服务涉及行业、领域的广泛性，特色的高校志愿服

务体系建设和调度、共享志愿服务相关的资源，让学生能更全面地了解志愿服务的内容和形式。在志愿服务活动的组织开展过程中，高校大学生之间交流学习、团队合作，共同完成志愿服务，在实践中提高思想政治素质和道德修养。同时能够实现高校学生参与志愿服务活动的实际情况，结合学生专业特色，给予相关方面的引导，更好地通过志愿服务平台来了解专业方面的知识和实践能力。

志愿服务平台为志愿者提供一个良好的平台以及对所在地区志愿者的管理，为赛事活动、公益事业等进行服务，同时搭建志愿者、志愿团体、求助者之间的信息沟通桥梁。通过这个渠道，更大力度宣传志愿服务活动，充分调动志愿者群体资源配置，确保志愿服务的顺利开展，更好地为志愿服务活动组织方提供优质的服务，更好的给大学生志愿者提供便捷的服务，将活动参与和服务质量相结合，建立志愿服务平台对于推进志愿服务活动的开展是一件意义重大的事情。

### 系统目标

志愿者服务平台搭建志愿者、志愿服务发布者、管理员之间的信息交流枢纽，对志愿者能够方便地获取志愿服务信息，对志愿服务发布者提供及时、高质量服务，对管理员高效的管理志愿服务。志愿者通过志愿服务平台浏览了解志愿服务相关新闻资讯、下载学习志愿服务相关的政策文件，报名参加各项志愿服务，为爱心项目捐献个人爱心积分，提交建议反馈信息。志愿服务发布者通过志愿服务平台发布志愿活动，审核志愿者报名申请，发布志愿服务行业有关的新闻动态，设置人员标准。管理员通过志愿服务平台审批志愿服务发布者发布的志愿活动，上传志愿服务相关的政策文件资料供用户下载学习，查看用户提交的建议反馈信息，新增爱心项目及相关的管理，添加学员管理员账户、创建学院和添加学生志愿者身份类别，审批学院志愿者和设置志愿者身份类别。志愿服务系统主要实现以下目标：

1. 展示高校好形象，宣传高校与志愿服务活动相关的各种资源、特色服务、平台建设、服务政策等信息。
2. 公布学校建立的志愿服务体系、服务详情、服务文化等全面的服务信息，方便学

生、老师、领导了解学校的志愿服务内容。

1. 依托学习、奉献搭建的志愿服务平台，大力开展公益事业，为公益服务做出贡献。
2. 采集和了解学生、志愿者、社团、老师、志愿服务发布者等群体的问题、建议、意见，为高校志愿者服务的改善提供参考依据。
3. 实现简便、高效的志愿服务相关管理工作，在各项服务中灵活使用，适用于当前高校开展志愿服务的实际需求，进一步减少志愿服务活动的复杂性，减轻志愿服务活动管理人员的工作负担。
   1. **业务需求**

# 2 需求分析

志愿者：高校学生群体。查看、修改个人账号状态、个人相关信息（所属学院、志愿服务时长、爱心积分等），报名参加志愿活动，捐献个人爱心积分，查看已报名的志愿服务活动列表信息，查看爱心积分捐献明细。

志愿服务发布者：校内外志愿活动、公益项目负责人。查看、修改个人相关信息， 查看所有人员标准（性别、年龄、标准名、可选择学院、可选择身份），新建、修改、删除人员标准，发布、修改、删除志愿服务相关的新闻资讯，查看已发布的新闻资讯列表信息，发布志愿服务活动，查看已发布的志愿服务活动列表信息，修改志愿服务活动信息，撤销志愿服务活动，查看某志愿服务活动所有已通过报名者，删除已提交报名申请的志愿者。

管理员：院校两级负责人。审批志愿服务发布者发布的志愿服务，上传、修改、删除志愿服务相关的政策文件，查看已上传的政策文件列表信息，查看用户提交的建议反馈信息，添加爱心项目、修改爱心项目相关信息、删除爱心项目，查看已添加的爱心项目列表信息，添加学院管理员账户、修改、删除学院管理员，查看已添加的学院管理员列表信息，添加、修改、删除学院，查看已添加的学院列表信息，添加、修改、删除身份类别信息，查看已添加的身份类别列表信息。

* + 1. 主要业务流程

（一）新闻资讯发布活动图如图 2-1 所示：



N

志愿服务发布者 填写新闻发布信息

添加新闻资讯

(标题、内容等)

确认发布

Y

新闻资讯发布



图 2-1 新闻资讯发布活动图

（二）政策文件上传活动图如图 2-2 所示：

N

管理员上传政 填写上传文件

策文件

信息

选择上传文件

确认上传 Y

图 2-2 政策文件上传活动图



政策文件上传

（三）志愿服务发布活动图如图 2-3 所示：

N

志愿服务发布者添

加志愿服务活动

填写活动发布信息(标

题、时间、地点等)

确认发布

Y

图 2-3 志愿服务发布活动图



志愿服务发布

（四）志愿服务审批活动图如图 2-4 所示：



通知活动发布

者未能通过

N

管理员查看志愿

服务活动列表

审核志愿服务

活动

创建通过活

动列表

审核通过

Y

活动审核通过

通知活动发

布者通过

图 2-4 志愿服务审批活动图

（五）志愿服务报名活动图如图 2-5 所示：

注册志愿者

N

查看已发布的

志愿活动信息

志愿者已注

册

Y



审核未通过

审核已通过

报名申请

报名完成

图 2-5 志愿服务报名活动图

（六）建议反馈提交活动图如图 2-6 所示：

N

提交建议完成



 填写建议反馈

内容

确认提交 Y

图 2-6 建议反馈提交活动图

（七）爱心项目添加活动图如图 2-7 所示：

Y



N

管理员添加爱

心项目

填写爱心项目

信息

选择上传图片

确认添加



爱心项目添加完成

图 2-7 积分捐献活动图

### 功能需求

* + 1. 角色分析

以下从志愿服务平台的实际需求分析，系统涉及到以下角色如表 2-1 所示：

表 2-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 志愿者 | 报名参加志愿服务活动，捐献个人爱心积分 |
| 志愿活动发布者 | 发布志愿服务相关新闻资讯，发布志愿活动、审批志愿者报名申请  和组织安排志愿活动的相关管理。 |
| 系统管理员 | 审批志愿服务发布者发布的志愿活动，上传志愿服务相关的政策文  件，查看建议反馈信息，添加爱心项目，添加身份类别、人员标准， 审批志愿者，对用户、管理员级别的一系列相关管理。 |

志愿服务平台是针对高校开展志愿服务搭建的系统，角色分为：志愿者、志愿服务发布者、管理员。

志愿者：在校学生进入志愿服务平台网站注册成为志愿者。志愿者浏览学校在志愿服务平台中发布、通知志愿服务活动并选择其中的项目参加。填写、上传志愿服务相关的建议反馈给管理员，将自己参加志愿服务活动所获得的爱心积分捐献给公益组织或爱心捐助项目。

志愿服务发布者：高校社团、学生会部门，院级、校级志愿服务协会，校外单位或组织等。招募已注册成为志愿者并符合条件的在校大学生参加审批通过的志愿服务活动， 以及志愿服务活动过程中的一系列管理，管理志愿服务活动中志愿者、服务活动的相关事务，发布志愿服务新闻资讯及其相关的各项管理。

管理员：社团、部门、协会负责人和老师。对志愿者申请、审批、人员、信息等进行管理，添加身份类别、人员标准，审批志愿者，对用户、管理员级别的一系列相关管理，对志愿服务发布者发布的志愿活动审批等信息进行管理，对游客、志愿者、志愿服务发布者提交的建议反馈等信息进行管理，对志愿者的爱心积分捐献、公益组织、爱心捐助项目的相关事务进行统筹安排。

* + 1. 功能需求

志愿服务平台总体用例图包括：系统管理、新闻资讯管理、政策文件管理、志愿活动管理、建议反馈管理、爱心项目管理。其中系统管理子系统包括：用户管理、人员标准管理、身份类别管理、学院管理四个模块；志愿活动管理子系统包括：活动发布管理、活动审批管理、活动报名管理三个模块；爱心项目管理子系统包括：爱心积分管理模块。如下图 2-8 所示：



用户管理

系统管理

<<包含>>

<<包含>> 学院管理

<<包含>>

<<包含>> 身份类别管理

新闻资讯管理

人员标准管理

政策文件管理

活动发布管理

志愿服务发布者

志愿活动管理

管理员

<<包含>>

<<包含>> 活动审批管理

<<包含>>

活动报名管理

建议反馈管理

爱心项目管理 <<包含>> 爱心积分管理

志愿者

1. 用户管理

图 2-8 系统总体用例图

用户管理包括：系统管理员添加用户、浏览用户的相关信息、修改用户相关信息和删除用户，用户管理用例图如图 2-9 所示：



<<

<<包含>>

<<包含>>

<<

管理员

删除用户

修改用户

包含>>

查看用户

包含>>

添加用户

图 2-9 用户管理用例图

添加用户的用例描述如表 2-2 所示：

表 2-2 添加用户管理用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 添加用户 |
| 标识符**\*** | YH01 |
| 用例描述 | 描述了管理员对用户管理进行添加用户的整个相关过程 |
| 参与者表 | 管理员 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 管理员进入用户管理 |
| 后置条件 | 管理员完成添加操作 |
| 基本操作流 | 1. 管理员点击用户管理进入用户管理界面 2. 点击“添加”进入添加用户界面 3. 填写添加用户的相关信息 4. 点击“确认”完成添加用户 |

1. 新闻资讯管理

新闻资讯管理包括：志愿服务发布者添加新闻资讯、修改新闻资讯、删除新闻资讯、查看新闻资讯详细信息；志愿者、管理员查看新闻资讯相关内容。新闻资讯管理用例图如图 2-10 所示：



添加新闻资讯

<<包含>>

<<包含>>

查看新闻资讯

管理员

<<包含>>

<<包含>> <<包含>>

<<包含>>

修改新闻资讯

志愿活动发布者

删除新闻资讯

志愿者

图 2-10 新闻资讯管理用例图

新闻资讯管理的用例描述如表 2-3、表 2-4、表 2-5 所示：

表 2-3 志愿服务发布者新闻资讯管理用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 新闻资讯管理 |
| 标识符**\*** | XW01 |
| 用例描述 | 描述了志愿服务发布者登录志愿服务平台，进入新闻资讯管理页面， 填写需要发布的新闻资讯内容，添加新闻资讯。 |
| 参与者表 | 志愿服务发布者 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 志愿服务发布者进入新闻资讯管理页面 |
| 后置条件 | 志愿服务发布者完成新闻资讯管理操作 |
| 基本操作流 | 1. 志愿服务发布者进入新闻资讯管理页面 2. 点击“添加”进入添加新闻资讯界面 3. 填写新闻资讯的相关信息 4. 点击发布完成发布新闻资讯 |

表 2-4 管理员查看新闻资讯用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 新闻资讯管理 |
| 标识符**\*** | XW02 |
| 用例描述 | 描述了管理员进入新闻资讯页面，查看管理员已发布的新闻资讯内容。 |
| 参与者表 | 管理员 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 管理员进入新闻资讯页面 |
| 后置条件 | 管理员完成查看新闻资讯操作 |
| 基本操作流 | 1. 管理员进入新闻资讯页面 2. 点击新闻列表查看新闻资讯内容 |

表 2-5 志愿者查看新闻资讯用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 新闻资讯管理 |
| 标识符**\*** | XW03 |
| 用例描述 | 描述了志愿者进入新闻资讯页面，查看管理员已发布的新闻资讯内容。 |
| 参与者表 | 志愿者 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 志愿者进入新闻资讯页面 |
| 后置条件 | 志愿者完成查看新闻资讯操作 |
| 基本操作流 | 1. 志愿者进入新闻资讯页面 2. 点击新闻列表查看新闻资讯内容 |

1. 政策文件管理

政策文件管理包括：管理员上传政策文件、查看政策文件、删除政策文件、下载政策文件；志愿服务发布者、普通用户（志愿者、游客）查看、下载管理员已上传的政策文件。政策文件管理用例图如图 2-11 所示：



上传政策文件

<<包含>>

<<包含>>

志愿活动发布者

<<包含>>

<<包含>>

查看政策文件

<<包含>>

<<包含>>

管理员

<<包含>>

删除政策文件

<<包含>>

下载政策文件

志愿者

图 2-11 政策文件管理用例图

政策文件的用例描述如表 2-6、表 2-7、表 2-8 所示：

表 2-6 管理员上传政策文件管理用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 政策文件管理 |
| 标识符**\*** | ZC01 |
| 用例描述 | 描述了管理员登录志愿服务平台，进入政策文件上传页面，选择需要上传的政策文件，上传政策文件。 |
| 参与者表 | 管理员 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 管理员进入政策文件管理页面 |
| 后置条件 | 管理员完成政策文件管理操作 |
| 基本操作流 | 1. 系统管理员进入政策文件管理页面 2. 填写政策文件相关信息   3 选择需要上传的政策文件完成政策文件添加  4.点击提交完成政策文件上传 |

表 2-7 志愿服务发布者查看政策文件用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 政策文件查看 |
| 标识符**\*** | ZC02 |
| 用例描述 | 描述了志愿服务发布者进入政策文件页面，查看政策文件，下载政策文件 |
| 参与者表 | 管理员 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 志愿服务发布者进入政策文件页面 |
| 后置条件 | 志愿服务发布者完成下载政策文件操作 |

续表 2-7

|  |  |
| --- | --- |
| 基本操作流 | 1. 志愿服务发布者进入政策文件页面 2. 查看政策文件相关列表信息 |

表 2-8 志愿者下载政策文件用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 政策文件下载 |
| 标识符**\*** | ZC03 |
| 用例描述 | 描述了志愿者进入政策文件页面，查看政策文件，下载政策文件 |
| 参与者表 | 管理员 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 志愿者进入政策文件页面 |
| 后置条件 | 志愿者完成下载政策文件操作 |
| 基本操作流 | 1. 志愿者进入政策文件页面 2. 查看政策文件相关列表信息 3. 点击下载政策文件 |

1. 建议反馈管理

建议反馈管理包括：志愿者添加建议反馈；管理员查看建议反馈信息、删除建议反馈。建议反馈管理用例图如图 2-12 所示：



<<包含>>

添加建议反馈

志愿者

查看建议反馈

<<包含>>

<<包含>>

删除建议反馈

管理员

图 2-12 建议反馈管理用例图

建议反馈管理的用例描述如表 2-9、表 2-10 所示：

表 2-9 志愿者添加建议反馈用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 建议反馈添加 |
| 标识符**\*** | FK01 |
| 用例描述 | 描述了志愿者登录志愿服务平台，进入建议反馈页面，填写建议反馈内容，提交建议反馈。 |
| 参与者表 | 志愿者 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 志愿者进入建议反馈页面 |
| 后置条件 | 志愿者完成添加建议反馈操作 |
| 基本操作流 | 1. 志愿者进入建议反馈页面 2. 填写相关信息完成建议反馈添加 3. 点击提交完成建议反馈 |

表 2-10 管理员查看建议反馈用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 查看建议反馈 |
| 标识符**\*** | FK02 |
| 用例描述 | 描述了管理员登录志愿服务平台，进入建议反馈页面，查看建议反馈信息。 |
| 参与者表 | 管理员 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 管理员进入建议反馈管理页面 |
| 后置条件 | 管理员完成查看建议反馈操作 |
| 基本操作流 | 1. 管理员进入建议反馈管理页面， 2. 浏览志愿者提交的建议反馈详细信息 |

1. 志愿活动管理

志愿活动管理包括：志愿活动发布者添加志愿活动、查看志愿活动、修改志愿活动、删除志愿活动；管理员查看志愿活动、审批志愿活动、查看审批通过活动；志愿者查看管理员审批通过活动、添加报名申请、查看报名信息。志愿活动管理用例图如图 2-13 所示：



修改活动信息

<<包含>>

<<包含>>

<<包含>>

<<包含>>

志愿活动发布者

<<

管理员

<<包含>>

审批志愿活动

查看审批通过活动

提交报名申请

<<包含>>

<<包含>>

<<包含>>

取消报名申请

志愿者

发布志愿活动

包含>>

<<包含>>

删除活动信息

<<包含>>

浏览活动信息

添加活动信息

图 2-13 志愿活动管理用例图

志愿活动管理的用例描述如表 2-11、表 2-12、表 2-13 所示：

表 2-11 志愿服务发布者志愿活动管理用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 管理员志愿活动管理 |
| 标识符**\*** | HD01 |
| 用例描述 | 描述了管理员对志愿服务发布者已发布的志愿服务活动查看、审批相关志愿服务活动的一系列流程。 |
| 参与者表 | 管理员 |
| 优先级 | 1 |

续表 2-11

|  |  |
| --- | --- |
| 状态**\*** | 进行中 |
| 前置条件 | 管理员进入志愿活动管理页面 |
| 后置条件 | 管理员完成志愿活动管理操作 |
| 基本操作流 | 1. 管理员进入政策文件管理页面 2. 浏览志愿服务发布者提交的志愿活动详细信息 3. 审核志愿活动完成审批通过 |

表 2-12 志愿服务发布者志愿活动管理用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 志愿服务发布者志愿活动管理 |
| 标识符**\*** | HD02 |
| 用例描述 | 描述了志愿服务发布者添加、修改、删除志愿服务活动信息，发布志愿服务活动和查看已发布的志愿服务活动相关的一系列流程。 |
| 参与者表 | 志愿服务发布者 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 志愿服务发布者进入志愿活动管理页面 |
| 后置条件 | 志愿服务发布者完成志愿活动管理操作 |
| 基本操作流 | 1. 志愿服务发布者进入志愿活动管理页面 2. 填写志愿活动发布信息完成添加 3. 点击发布完成志愿服务活动发布 |

表 2-13 志愿者志愿活动管理用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 志愿者志愿活动管理 |
| 标识符**\*** | HD03 |
| 用例描述 | 描述了志愿者查看志愿服务活动信息，报名申请参加志愿活动相关的一系列流程。 |
| 参与者表 | 志愿者 |

续表 2-13

|  |  |
| --- | --- |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 志愿者进入志愿活动详情页面 |
| 后置条件 | 志愿者完成志愿活动相关操作 |
| 基本操作流 | 1. 志愿者查看已审批通过的志愿活动详细信息 2. 选择报名参加的志愿活动提交报名申请 3. 完成报名参加志愿服务活动 |

1. 爱心项目管理

爱心项目管理包括：管理员添加爱心项目、修改爱心项目、查看爱心项目，删除爱心项目；志愿者查看爱心项目相关内容。爱心项目管理用例图如图 2-14 所示：

添加爱心项目



管理员

<<包含>>

<<包含>>

<<包含>>

<<包

含>>

查看爱心项目

修改爱心项目

<<包含>>



志愿者

删除爱心项目

图 2-14 爱心积分管理用例图

爱心项目管理的用例描述如表 2-14、表 2-15 所示：

表 2-14 志愿者查看爱心项目用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 爱心项目查看 |
| 标识符**\*** | AX01 |
| 用例描述 | 描述了志愿者查看爱心项目相关信息操作的过程 |
| 参与者表 | 志愿者 |

续表 2-14

|  |  |
| --- | --- |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 志愿者进入爱心项目页面 |
| 后置条件 | 志愿者完成查看爱心项目信息 |
| 基本操作流 | 1. 志愿者进入爱心项目详情页面 2. 查看已发布的爱心项目信息 |

表 2-15 管理员爱心项目管理用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 爱心项目管理 |
| 标识符**\*** | AX01 |
| 用例描述 | 描述了管理员对爱心项目相关管理的整个过程 |
| 参与者表 | 管理员 |
| 优先级 | 1 |
| 状态**\*** | 进行中 |
| 前置条件 | 管理员进入爱心项目管理页面 |
| 后置条件 | 管理员完成爱心项目管理操作 |
| 基本操作流 | 1. 管理员进入爱心项目管理页面 2. 填写爱心项目详细信息完成添加 3. 点击发布完成爱心项目发布 |

### 非功能需求

* + 1. 环境需求

服务器端环境如表 2-16 所示：

表 2-16 服务器端环境

|  |  |
| --- | --- |
| 系统所使用数据库 | MySQL |
| 操作系统 | Windows 10 |
| Web 服务器 | Tomcat 8 |
| 数据库 | MySQL 5.7 |

客户端环境如表 2-17 所示：

表 2-17 客户端环境

|  |  |
| --- | --- |
| 操作系统 | Windows XP 或其以上版本操作系统 |
| 浏览器 | IE 6.0 以及其以上版本（IE7.0、IE8.0），Firefox 浏览器  等主流浏览器 |
| 分辨率 | 最佳效果为 1028×960 像素 |

* + 1. 性能需求

1. 可靠性

在系统发生出错的情况下，不会影响用户的行为操作与数据，系统故障发生率要保持在一定的水平下。

1. 安全性

对系统中各项操作的数据采取加密保护措施，确保数据在各种使用过程中不被修改、窃取，并且根据各个用户的相应权限控制该用户访问的数据、进行的操作合理化。

1. 可维护性

系统可以在数据遭到恶意操作、数据不合理损坏等情况下查询到相应的记录并对数据进行恢复。

1. 易用性

系统中各个界面简洁、美观，风格统一，各项功能实用，易于各用户群体操作使用， 符合用户实际需要，方便管理和高效处理实际应用问题。

* + 1. 安全需求

1. 权限控制

根据不同的用户角色设置该用户相应权限，所有用户可以浏览前台展示页面信息， 包括：新闻资讯、志愿服务活动、政策文件、志愿统计数据。志愿者、志愿服务发布者、管理员登录账号，并且对个人信息及其他相关业务进行管理，志愿者可以报名参加志愿活动、爱心积分捐献等操作，志愿服务发布者可以发布志愿活动、新闻资讯等操作，管理员可以进行后台数据的相关管理，审批志愿者、添加学院、上传政策文件等操作。

1. 日志记录

系统能够记录用户关键性操作信息，同时记录系统运行时发生的所有错误。以便后续系统出现故障能够回到系统故障之前的状态。

1. 用户安全

对密码信息进行密码隐藏保护，对网站上用户登录信息的获取采用 post 方法以保证信息的安全传输。

# 3 总体设计

### 系统设计的原则

在充分考虑到本系统的总体规划设定，以及系统的应用环境前提下，本系统的设计将严格按照以下所涉及的设计原则：

1. 稳定性原则。系统经过长时间的运行之后，保持稳定的服务，各项功能正常，不会

出现系统响应时间较大增长和反应迟缓的相关现象，响应时间合理，能及时的在规定的时间内完成相应的操作处理。

1. 合理性原则。充分保障整个系统的运行流畅，数据信息准确，确保在实际使用过程中的每一个环节上都能符合需求的相应处理，使得志愿服务管理最大程度上满足用户的实际需求。
2. 实用性原则。系统充分考虑志愿者、志愿服务发布者、管理员三者之间的关系，详细分析用户的实际需求，各类用户在处理志愿服务相关事务的正常逻辑操作，确保准确、真实进行操作，着力解决不及时、效率低等问题，搭建一个便捷、高效管理的志愿服务平台。
3. 安全性原则。系统中不同的用户对系统进行操作时确保操作合法，不同用户相应的权限划分合理，保证志愿者、志愿服务发布者、管理员之间不会出现权限不合理的相关操作，避免合法用户进行非法操作，阻止非法人员对系统进行操作，数据库建立完备的数据备份处理。
4. 可扩展性原则。系统中的各个功能都具有扩展接口，方便系统功能进一步扩展的需要，充分考虑本系统功能结构上的一些扩展和调整处理，保持系统拥有良好的扩展性，方便系统的进一步开发。
5. 易维护原则。系统在实际的使用过程中发现某些故障或问题时可以较准确、快速地找到这个问题所在之处并较为顺畅的处理解决。在设计志愿服务系统时保证系统结构的合理性，系统功能模块组成清晰，对相应的部分进行修改、调整能快速实现， 易于维护。

### 系统体系结构设计

志愿服务平台的设计采用的体系结构是 B/S 结构，利用了日渐成熟的浏览器技术， 用户界面也完全通过浏览器实现，极大方便用户使用，在部署和维护方面也及其高效， 可拓展性强，便于进行软件开发以及节约软件成本，共享性、交互性强。

系统采用SSM 框架进行开发，数据库采用 Mysql，Maven 仓库用于项目的依赖管理，基础环境的搭建需要创建一个 maven 工程，引入项目依赖的 jar 包，包括：spring、spring mvc、mybatis、数据库连接池、驱动包等其他相关配置文件，Spring 是一个管理 bean 的

大容器，方便解耦，极大方便软件的开发，所有对象创建和依赖关系维护都可以全部让Spring 进行相关的管理，Spring 提供面向切面编程，在具体的实现上可以十分方便地对程序进行权限拦截、运行监控等操作，同时也对集成各种优秀框架非常简便，对很多优秀的框架都可以提供直接支持。Spring MVC 是一个 web 层、控制层框架，主要用来负责与客户端交互，业务逻辑的调用，SpringMVC 是 Spring 中的一大组件，Spring 整合SpringMVC 可以做到无缝集成。MyBatis 是一个优秀的持久层框架，通用 Mapper 是一款用于单表增删改查的Mybatis 插件，开发人员可以省去编写 sql 语句和在 DAO 层编写任何方法，在使用通用 Mapper 时，dao 层接口会继承通用 Mapper 的相关接口，通过在运行期间反射获取所需的实体类信息，并由此构造出对应的SQL。

网络层包括的网络基础设施有存储系统、数据库系统、操作系统、服务器等系统所需要相关设施。Dao 层继承通用 Mapper 接口，获取常用的相关方法，包括增、删、改、查等相关方法，完成对应 SQL 的调用和数据库查询，为业务逻辑层提供数据服务。业务逻辑层是对接口的实现业务的逻辑层，主要是对具体问题的一系列相关操作，对数据业务进行相关的逻辑处理。控制层是处理请求的跳转和处理前端发来的请求，对用户的请求接受，以及数据的返回，为客户端提供应用程序的访问。用户界面通过 HTML、CSS、Bootstrap 等技术实现界面间跳转和界面相关内容的展示。

### 系统功能结构设计

系统管理子系统：包括用户管理、人员标准管理、身份类别管理、学院管理四个模块。身份类别管理包括系统管理员对不同用户群体设定该用户相应的身份类别，查看该用户对应的身份类别，修改和删除身份类别。人员标准管理包括志愿服务发布者对发布志愿活动进行招募符合一定条件的志愿者的管理，新建人员标准以对特定的志愿服务活动进行招募志愿者，查看人员标准信息，删除人员标准。学院管理包括系统管理员对学院管理员、学院名称相关信息的操作管理，创建学院管理员账户，新增学院名称相关信息，删除学院相关信息。

新闻资讯管理子系统：志愿服务发布者添加、修改新闻资讯，查看新闻资讯详细信息，删除新闻资讯，已发布的新闻资讯相关内容在前台网站展示；志愿者、游客浏览志愿服务发布者已发布的新闻资讯相关内容。

政策文件管理子系统：管理员添加需要上传的政策文件、删除政策文件、查看政策文件详细内容，政策文件相关的内容在前台网站展示；志愿者和游客浏览、下载管理员已上传的政策文件相关内容。

志愿活动管理子系统：包括活动发布管理、活动审批管理、活动报名管理三个模块。活动发布管理包括志愿服务发布者添加发布志愿服务活动，查看进行发布的志愿服务活动的详细信息，删除志愿服务活动。活动审批管理包括系统管理员查看志愿服务发布者

已发布的志愿服务活动列表相关信息，审批单个或批量审批志愿服务活动是否通过，志愿服务活动未能审批通过需填写拒绝理由，已审批通过的志愿服务活动相关内容在前台网站展示。活动报名管理包括志愿者浏览查看管理员已审批通过的志愿服务活动相关内容，选择志愿活动提交报名申请参加志愿服务活动，可在志愿者后台查看个人已报名的志愿活动相关信息，志愿服务发布者可查看已报名参加该活动的志愿者信息和删除已报名的志愿者。

建议反馈管理子系统：游客、用户对志愿服务平台管理、志愿服务活动管理等相关的事件提交建议反馈信息。管理员查看游客、用户已提交的建议反馈内容，管理员删除游客、用户提交的建议反馈。

爱心项目管理子系统：包括爱心项目、爱心积分管理模块。系统管理员添加设置爱心项目相关信息，查看爱心项目详细信息，删除爱心项目，捐献个人爱心积分，查看爱心积分捐献明细。

系统功能结构设计图如图 3-3 所示：

图 3-3 系统总包图

# 4 数据库设计

### 概念结构设计

* + 1. 设计思路

首先，对需求分析过程中的相关数据进行分类、组织，确定实体、实体的属性以及实体之间的关系类型，在形成过程中解决各分 E-R 图之间的冲突，将分 E-R 图合并起来生成初步E-R 图，在各 E-R 图之间的冲突和冗余等需要将这些问题合理消除，最终形成系统总 E-R 图。各个实体之间的关系为：

管理员与新闻的关系为一对多，管理员与政策文件的关系为一对多。

志愿服务发布者与志愿服务的关系为一对多，志愿服务发布者与新闻的关系为一对多，志愿服务发布者与人员标准的关系为一对多。

志愿者与志愿服务的关系为多对多，志愿者与爱心项目的关系为多对多，志愿者与

身份的关系为多对多，志愿者与建议反馈的关系为一对多，志愿者与学院的关系为多对一，志愿服务与留言的关系为一对多。

人员标准与学院的关系为多对多，人员标准与身份的关系为多对多，人员标准与志愿服务的关系为一对多。

* + 1. E-R 图

志愿服务平台总E-R 图如图 4-1 所示：



1

上传

n 政策文件

包含

n

n

1

属于

n

包含

1

n

1

创建 n

1

n

1

限制

包含

1

包含

1

n

n

n

1

1

1

1

1

1

n

报名

属于

捐赠

n

n

n

n

n 拥有 1

志愿者

学院

学院-人员标准

人员标准

人员标准- 身份

身份

爱心项目

建议反馈

志愿服务- 志愿者

志愿者-爱心项目

管理员

n

发布

1

志愿服务发布者

新闻

|  |  |
| --- | --- |
|  | 1 |
| 志愿者-身份 | |

属于

1

1

报名

n

发布

n

志愿服务

拥有

留言

图 4-1 系统总 E-R 图

1、新闻实体 E-R 图如图 4-2 所示： 实体：新闻

属性：newsId，标题，发布时间，内容

标题

发布时间

内容

newsId

新闻

图 4-2 新闻实体图

2、管理员实体 E-R 图如图 4-3 所示： 实体：管理员

属性：mid，手机号，级别

级别（1,学校

/2,学院）

手机号

mid

志愿服务管理员

(校/院 青协会长)

图 4-3 管理员实体图

3、志愿服务发布者实体E-R 图如图 4-4 所示： 实体：志愿服务发布者

属性：qid，性别，年龄，姓名，手机号

年龄

姓名

性别

手机号

志愿服务发布者

（社区管理员等）

图 4-4 志愿服务发布者实体图

qid

4、建议反馈实体 E-R 图如图 4-5 所示： 实体：建议反馈

属性：id，称呼，电话，时间，内容

称呼

电话

内容

时间

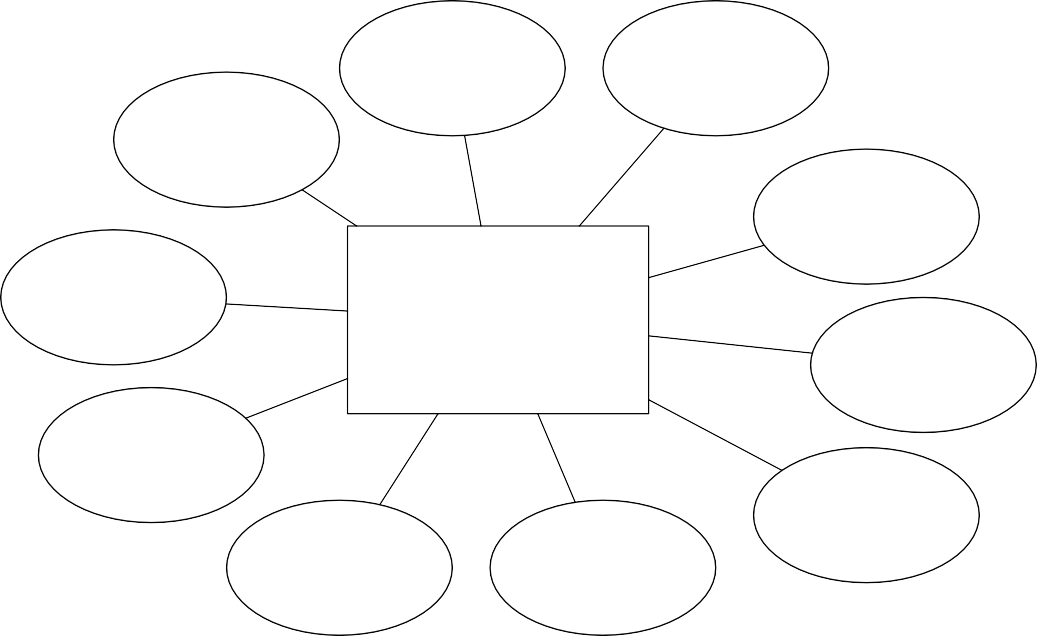
反馈

图 4-5 建议反馈实体图

id

5、志愿服务实体E-R 图如图 4-6 所示： 实体：志愿服务

属性：sid，人数需求，标题，群聊二维码，详情简介，相关文件，当前状态，时间， 地点，物资需求



标题

群聊二维码

人数需求

详情简介

sid

志愿服务

相关文件

物资需求

当前状态

地点

时间

图 4-6 志愿服务实体图

6、政策文件实体 E-R 图如图 4-7 所示： 实体：建议反馈

属性：文件uuid，文件名，时间，下载量

文件uuid

时间

政策文件

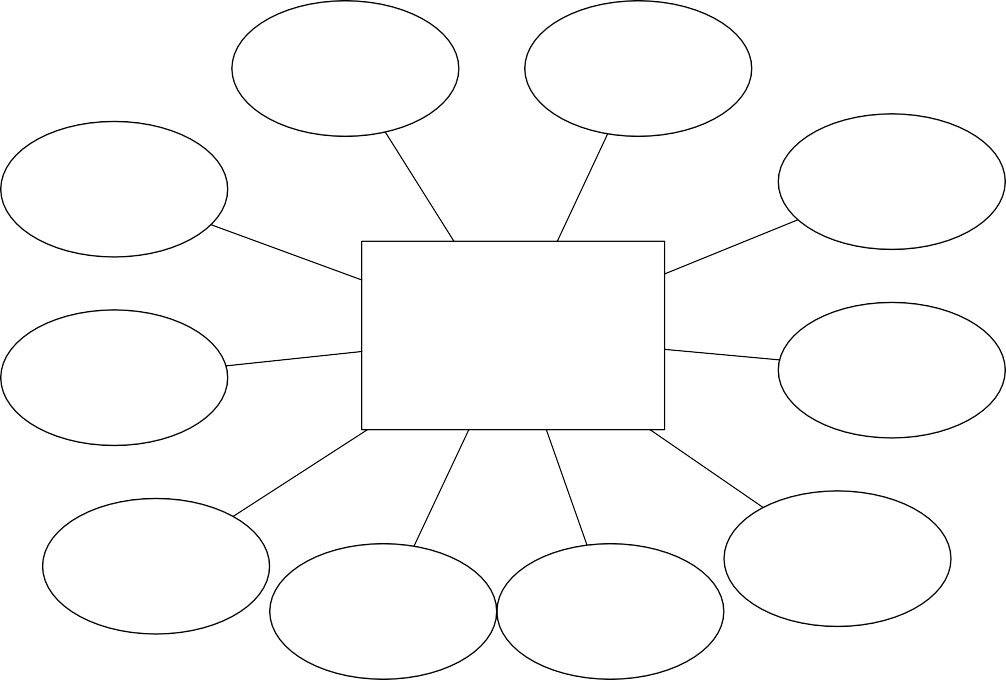
图 4-7 政策文件实体图

文件名

下载量

7、志愿者实体 E-R 图如图 4-8 所示： 实体：志愿者

属性：vid，状态，服务时长，年龄，学号，姓名，手机号，服务积分，性别，积分明细



状态

服务时长

vid

年龄

志愿者

积分明细

学号

性别

姓名

服务积分 手机号

图 4-8 志愿者实体图

8、人员标准实体E-R 图如图 4-9 示： 实体：人员标准

属性：pid，性别，年龄，标准名

性别

pid

人员标准

图 4-9 人员标准实体图

年龄

标准名

9、身份实体 E-R 图如图 4-10 所示： 实体：身份

属性：iid，名称

iid

身份

名称

图 4-10 身份实体图

10、学院实体 E-R 图如图 4-11 所示： 实体：学院

属性：did，名称

did

名称

学院

图 4-11 学院实体图

11、留言实体 E-R 图如图 4-12 所示： 实体：留言

属性：sid，称呼，类型，时间，内容

内容

留言

图 4-12 留言实体图

时间

称呼

类型

sid

12、爱心项目实体E-R 图如图 4-13 所示： 实体：爱心项目

属性：aid，标题，图片，总需求爱心，目前爱心数，目前献爱心人次

图片

标题

总需求爱心

目前爱心数

aid

爱心项目

图 4-13 爱心项目实体图

目前献爱心人次

### 逻辑结构设计

* + 1. 设计思路

首先将概念结构设计中系统总E-R 图转换为一般的关系模型，再将关系模型转化为数据库系统所支持的实际数据模型，最后进行数据模型的优化。

* + 1. 逻辑模型

1、新闻表（new）如表 4-1 所示：

表 4-1 新闻表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| newsId | 新闻 Id | 整型 | 10 | 非空 | PK 自增 |
| title | 标题 | 字符型 | 40 | 非空 |  |
| content | 内容 | 文本型 |  | 非空 |  |
| fabuTime | 发布时间 | 日期型 |  | 非空 |  |
| qid | 发布者 Id | 整型 | 10 | 非空 |  |
| mid | 管理员 Id | 整型 | 10 | 非空 |  |

2、志愿服务管理员表（mAdmin）如表 4-2 所示：

表 4-2 志愿服务管理员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| mid | 管理员 Id | 整型 | 10 | 非空 | PK 自增 |
| phone | 手机号 | 字符型 | 11 | 非空 |  |
| rank | 级别 | 整型 | 4 | 非空 |  |
| did | 学院 Id | 整型 | 10 | 非空 |  |

3、反馈表（feedback）如表 4-3 所示：

表 4-3 反馈表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| id | Id | 整型 | 10 | 非空 | PK 自增 |
| time | 时间 | 日期型 |  | 非空 |  |
| call | 称呼 | 字符型 | 10 | 非空 |  |
| phone | 电话 | 字符型 | 11 | 非空 |  |
| content | 内容 | 文本型 |  | 非空 |  |

4、政策文件表（document）如表 4-4 所示：

表 4-4 政策文件表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| uuid | Id | 字符型 | 30 | 非空 | PK |
| name | 文件名 | 字符型 | 30 | 非空 |  |
| dTime | 时间 | 日期型 |  | 非空 |  |
| downloads | 下载量 | 整型 | 10 | 非空 |  |

5、志愿服务者发布表（announce）如表 4-5 所示：

表 4-5 志愿服务者发布表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| qid | 发布者 Id | 整型 | 10 | 非空 | PK 自增 |
| sex | 性别 | 字符型 | 2 | 非空 |  |
| age | 年龄 | 整型 | 10 | 非空 |  |
| name | 姓名 | 字符型 | 20 | 非空 |  |
| phone | 手机号 | 字符型 | 11 | 非空 |  |

6、留言表（message）如表 4-6 所示：

表 4-6 留言表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| msgid | 留言 id | 整型 | 10 | 非空 | PK 自增 |
| content | 内容 | 字符型 | 200 | 非空 |  |
| call | 称呼 | 字符型 | 20 | 非空 |  |
| type | 类型 | 整型 | 10 | 非空 |  |
| time | 时间 | 日期型 |  | 非空 |  |
| sid | 志愿服务 Id | 整型 | 10 | 非空 |  |

7、志愿服务表（volService）如表 4-7 所示：

表 4-7 志愿服务表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| sid | 志愿服务 Id | 整型 | 10 | 非空 | PK 自增 |
| qid | 发布者 Id | 整型 | 10 | 非空 |  |
| pid | 人员标准 Id | 整型 | 10 | 非空 |  |
| title | 标题 | 字符型 | 40 | 非空 |  |
| content | 详情简介 | 文本型 |  | 非空 |  |
| vdocumnet | 相关文件 | 字符型 | 250 |  |  |
| state | 当前状态 | 整型 | 10 | 非空 |  |
| time | 时间 | 日期型 |  | 非空 |  |
| place | 地点 | 字符型 | 50 | 非空 |  |
| matDemand | 物资需求 | 字符型 | 50 |  |  |
| numDemand | 人数需求 | 整型 | 10 |  |  |
| qrcode | 群聊二维码 | 字符型 | 30 |  |  |

8、学院表（college）如表 4-8 所示：

表 4-8 学院表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| did | 学院 Id | 整型 | 10 | 非空 | PK 自增 |
| name | 名称 | 字符型 | 40 | 非空 |  |

9、人员标准表（peoStandard）如表 4-9 所示：

表 4-9 人员标准表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| pid | 人员标准 Id | 整型 | 10 | 非空 | PK 自增 |
| qid | 发布者 Id | 整型 | 10 | 非空 |  |
| sex | 性别 | 字符型 | 2 | 非空 |  |
| age | 年龄 | 字符型 | 10 | 非空 |  |
| staName | 标准名 | 字符型 | 20 | 非空 |  |

10、身份表（identity）如表 4-10 所示：

表 4-10 身份表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| iid | 身份 Id | 整型 | 10 | 非空 | PK 自增 |
| name | 名称 | 字符型 | 20 | 非空 |  |

11、（志愿者-爱心项目）表（volAip）如表 4-11 所示：

表 4-11 志愿者爱心项目表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| vid | 志愿者 Id | 整型 | 10 | 非空 | PK |
| aid | 爱心项目 Id | 整型 | 10 | 非空 | PK |
| donNumber | 捐献分值 | 整型 | 10 |  |  |

12、志愿者表（volunteer）如表 4-12 所示：

表 4-12 志愿者表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| vid | 志愿者 Id | 整型 | 10 | 非空 | PK 自增 |
| did | 学院 Id | 整型 | 10 | 非空 |  |
| name | 姓名 | 字符型 | 20 | 非空 |  |
| stuNumber | 学号 | 字符型 | 20 | 非空 |  |
| sex | 性别 | 字符型 | 2 | 非空 |  |
| age | 年龄 | 整型 | 10 | 非空 |  |
| phone | 手机号 | 字符型 | 11 | 非空 |  |
| serLeng | 服务时长 | 整型 | 10 | 非空 |  |
| serScore | 服务积分 | 整型 | 10 | 非空 |  |
| state | 状态 | 整型 | 10 | 非空 |  |
| serScoreRe | 积分明细 | 文本型 |  |  |  |

13、爱心项目表（aiProject）如表 4-13 所示：

表 4-13 爱心项目表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| aid | 项目 Id | 整型 | 10 | 非空 | PK 自增 |
| title | 标题 | 字符型 | 40 | 非空 |  |
| photo | 图片 | 字符型 | 30 | 非空 |  |
| aiNumber | 总需求爱心 | 整型 | 10 | 非空 |  |
| preNumber | 目前爱心数 | 整型 | 10 |  |  |
| prePeople | 献爱心人次 | 整型 | 10 |  |  |

14、（人员标准-身份）表（staide）如表 4-14 所示：

表 4-14 人员标准身份表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| pid | 人员标准 Id | 整型 | 10 | 非空 | PK |
| iid | 身份 Id | 整型 | 10 | 非空 | PK |

15、（人员标准-学院）表（staDep）如表 4-15 所示：

表 4-15 学院人员标准表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| pid | 人员标准 Id | 整型 | 10 | 非空 | PK |
| did | 学院 Id | 整型 | 10 | 非空 | PK |

16、（志愿者-身份）表（voide）如表 4-16 所示：

表 4-16 志愿者身份表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| vid | 志愿者 Id | 整型 | 10 | 非空 | PK |
| iid | 身份 Id | 整型 | 10 | 非空 | PK |

17、（志愿者-志愿服务）报名表（enroll）如表 4-17 所示：

表 4-17 志愿活动报名表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 数据类型 | 长度 | 空/非空 | 约束条件 |
| sid | 志愿服务 Id | 整型 | 10 | 非空 | PK |
| vid | 志愿者 Id | 整型 | 10 | 非空 | PK |
| time | 报名时间 | 日期型 |  | 非空 |  |
| msg | 删除理由 | 字符型 | 20 |  |  |

### 物理结构设计

* + 1. 存取方式

数据库类型：Mysql

* + 1. 存储结构

数据库采用Mysql 关系型数据库。按照系统所选择的数据库特点对相应的表结构进行设计，定义数据库、表及字段的命名规范，为表中的字段选择合适的数据类型，数据存储结构中整型：INT，字符型：VARCHAR，其中手机号使用 CHAR 类型，定长设定为 11，日期型：DATETIME，其格式是按照 YYYY-MM-DD HH:MM:SS[.fraction]格式存储日期时间，文本型：TEXT。

# 界面设计

### 界面关系图或工作流图

系统前台主界面包括：登录窗口、爱心项目展板、志愿统计数据。点击注册进入志愿者、志愿服务发布者注册界面，点击爱心项目展板图片或标题进入爱心项目详情界面， 点击志愿服务进入志愿服务列表和详情界面，点击新闻资讯进入新闻资讯列表界面，点击政策文件进入政策文件列表界面，点击建议反馈进入建议反馈详情界面，在登录窗口中选择用户类别，填写用户手机号、密码正确无误后，登录志愿服务平台，点击对应的用户后台进入相应的用户后台管理界面。

志愿者后台主界面包括志愿者个人详细信息和志愿者的账号状态，主页面左侧列表包括：已报名活动列表、积分捐献明细，点击进入对应的详细内容界面。志愿服务发布者主页面左侧列表包括：志愿服务管理、活动报名审批管理、新闻资讯管理、人员标准管理，点击进入对应的管理操作界面。学院管理员主页面左侧列表包括：志愿者管理、身份类别管理，点击进入对应的管理操作界面。学校管理员主页面左侧内容包括：志愿服务管理、政策文件管理、爱心项目管理、学院管理、身份类别管理，点击进入对应的管理操作界面。

前台网站界面关系图如图 5-1 所示：

跳转链接

主页内容

前台主界面

爱心项目

统计信息

用户登录

注册

爱心项目详情

志愿服务列表

新闻资讯列表

后台页面

建议反馈页面

政策文件列表

图 5-1 前台网站界面关系图

后台管理界面关系图如图 5-2 所示：

志愿服务发布者

内容

个人信息

学校管理员

学院管理员

志愿者后台

后台管理页面

志愿者个人信息

已报名活动列表

积分捐献明细

内容

人员标准详情

已发布志愿服务

活动报名审批

新闻发布

跳转链接

图 5-2 后台管理界面关系图

内容

已通过志愿者信息

待审批志愿者信息

身份类别设置

跳转链接

内容

学院管理员账户

所有学院列表志愿服务审批设置爱心项目政策文件上传

建议反馈详情页

跳转链接

### 界面设计成果

* + 1. 主界面

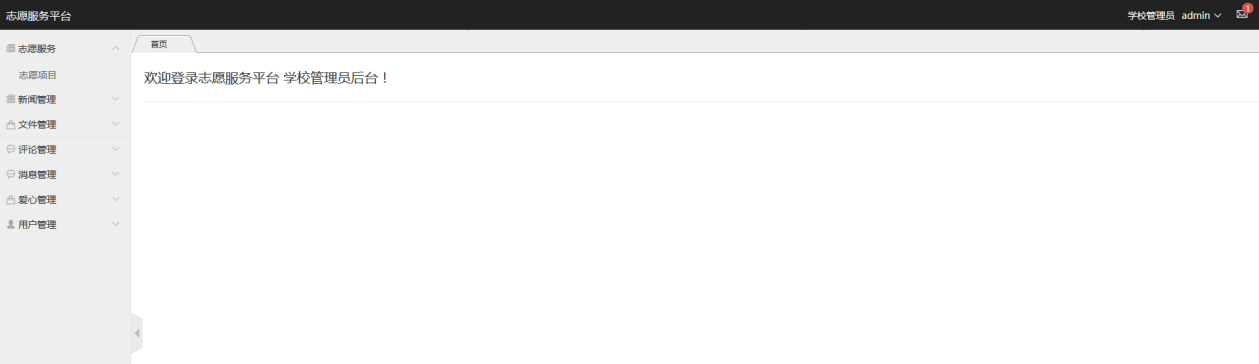
学校管理员后台管理主页面，首页界面左侧列表分别是：志愿服务、新闻资讯管理、政策文件管理、建议反馈管理爱心管理、用户管理，可进入相应管理的子页面进行操作， 如图 5-3 所示：

图 5-3 管理员后台管理主界面

志愿服务发布者后台管理页面，首页界面内容是志愿服务发布者个人相关信息，左侧列表分别是：志愿服务、新闻资讯管理，人员标准管理，可进入相应管理的子页面进行操作，如图 5-4 所示：



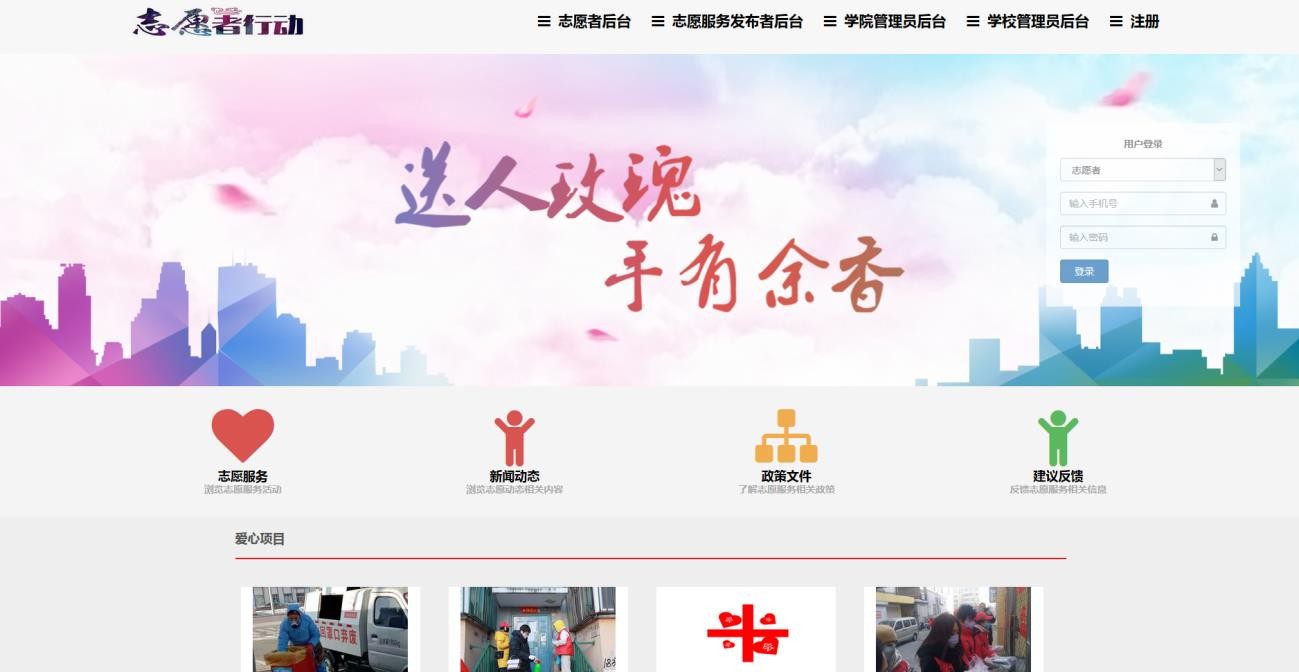
图 5-4 志愿服务发布者后台管理主界面

志愿者后台管理主页界面设有个人信息、已报名活动、积分捐献明细，个人信息包括：个人账号状态、姓名、学号、所属学院、手机、性别、年龄、身份类别、志愿时长、爱心积分，已报名活动可查看志愿者报名参加活动的详细信息，积分捐献明细可查看志愿者捐献积分的详细信息，如图 5-5 所示：



图 5-5 志愿者后台管理主界面

志愿服务平台门户网站主页界面最上方导航栏设有志愿者后台、志愿服务发布者后台、学员管理员后台、学校管理员后台、注册跳转按钮。主页界面设有用户登录、志愿服务、新闻动态、政策文件、建议反馈、爱心项目，各用户可查看相应的详细信息，前台主界面如图 5-6 所示：



* + 1. 子界面

图 5-6 志愿服务平台门户网站主界面

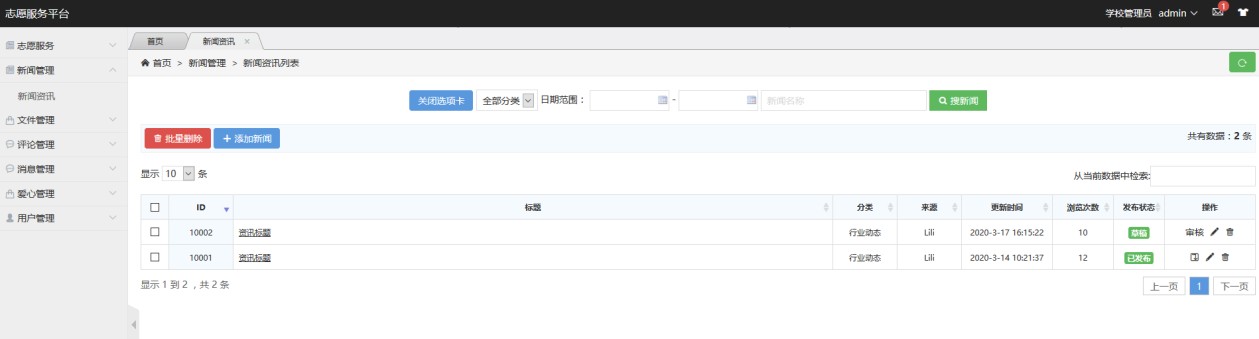
新闻资讯后台管理页面是管理员、志愿服务发布者对添加新闻资讯信息、修改新闻资讯信息、查看已添加的新闻资讯信息、发布新闻资讯、单个删除或批量删除新闻资讯的操作，如图 5-7 所示：

图 5-7 新闻资讯后台管理界面

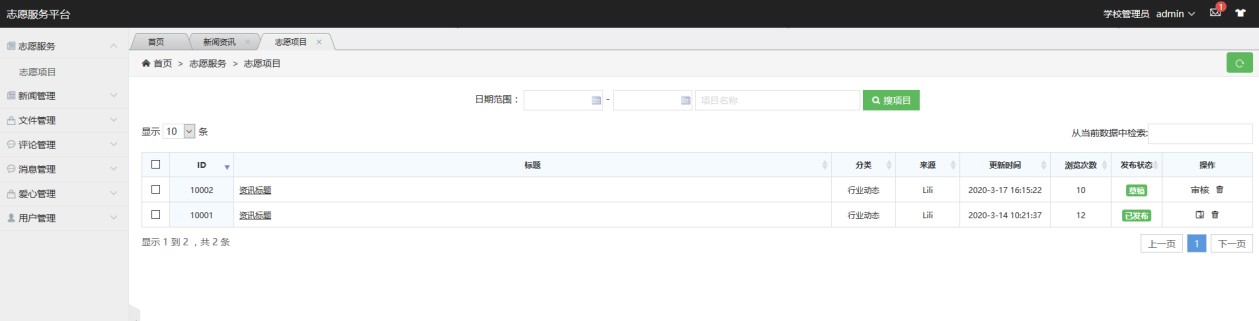
志愿服务活动审批界面是管理员对志愿服务发布者提交的志愿活动进行审核，管理员可查看相应的志愿活动详细信息并对该志愿活动进行审批操作，如图 5-8 所示：

图 5-8 志愿服务活动审批界面

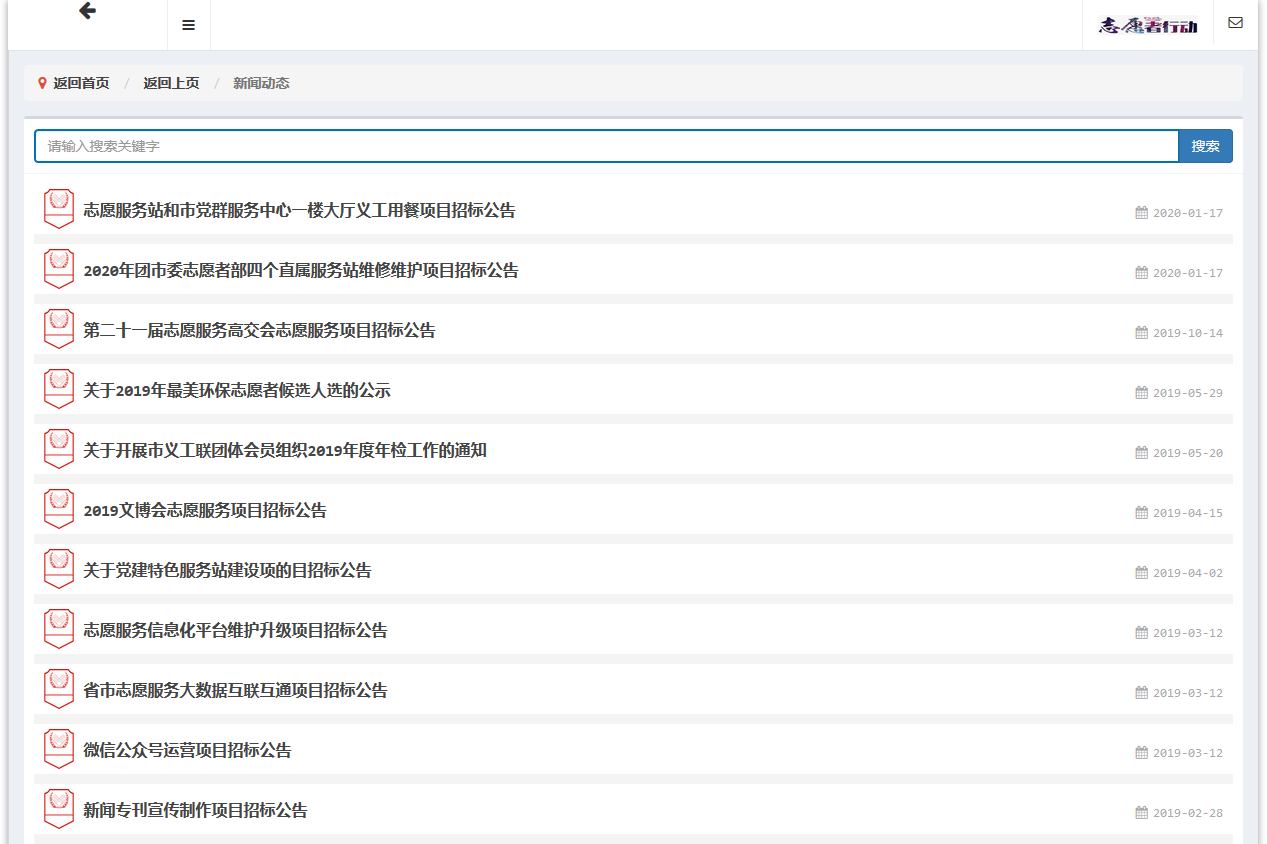
新闻资讯前台展示界面是管理员、志愿服务发布者已发布的新闻资讯列表信息，包括：搜索框、新闻资讯标题、发布时间等相关信息，点击新闻资讯标题可进入相应新闻详细信息界面，如图 5-9 所示：

图 5-9 新闻资讯前台展示界面

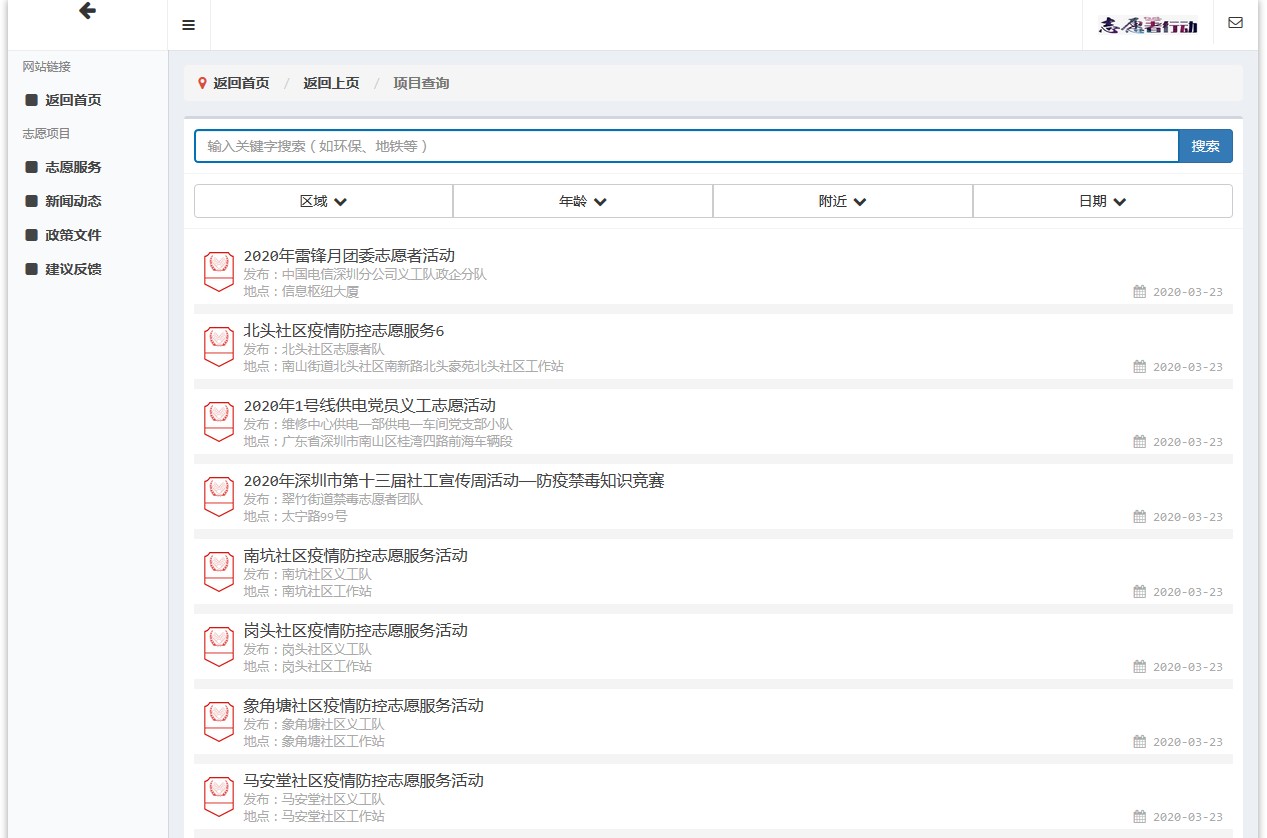
志愿服务活动前台展示界面是志愿服务发布者发布并且管理员审批通过的的志愿服务活动列表信息，包括：搜索框、按条件查询、志愿服务活动标题、发布时间等相关信息，点击志愿服务活动标题可进入相应志愿服务活动详细信息界面。左侧可点击志愿项目相关栏目进入志愿项目子项目界面，如图 5-10 所示：

图 5-10 志愿服务活动前台展示界面

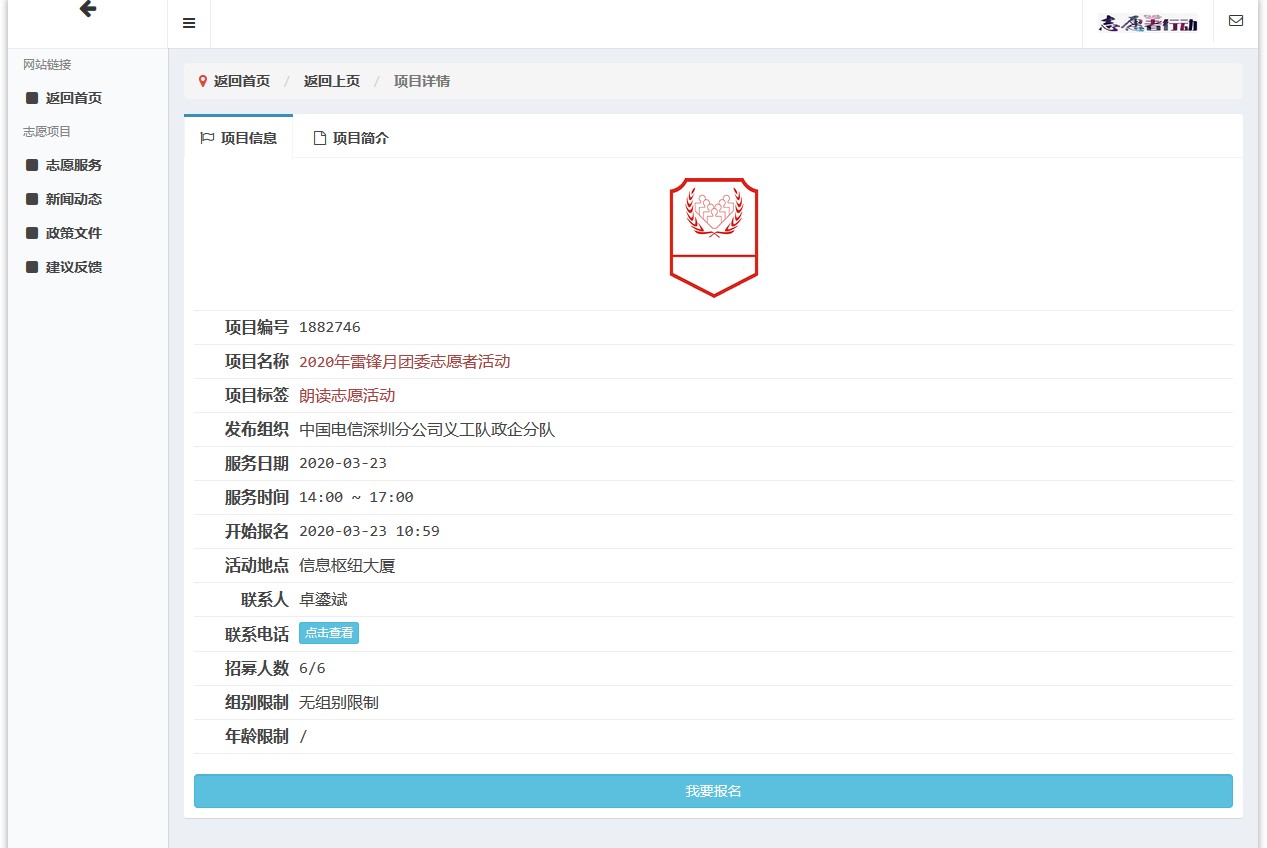
志愿服务活动报名界面是志愿者对管理员已审批通过的志愿服务活动提交报名申请，志愿者可查看所有志愿活动的详细信息，选择其中志愿活动进行报名。左侧可点击志愿项目相关栏目进入志愿项目子项目界面，如图 5-11 所示：

图 5-11 志愿服务活动报名界面

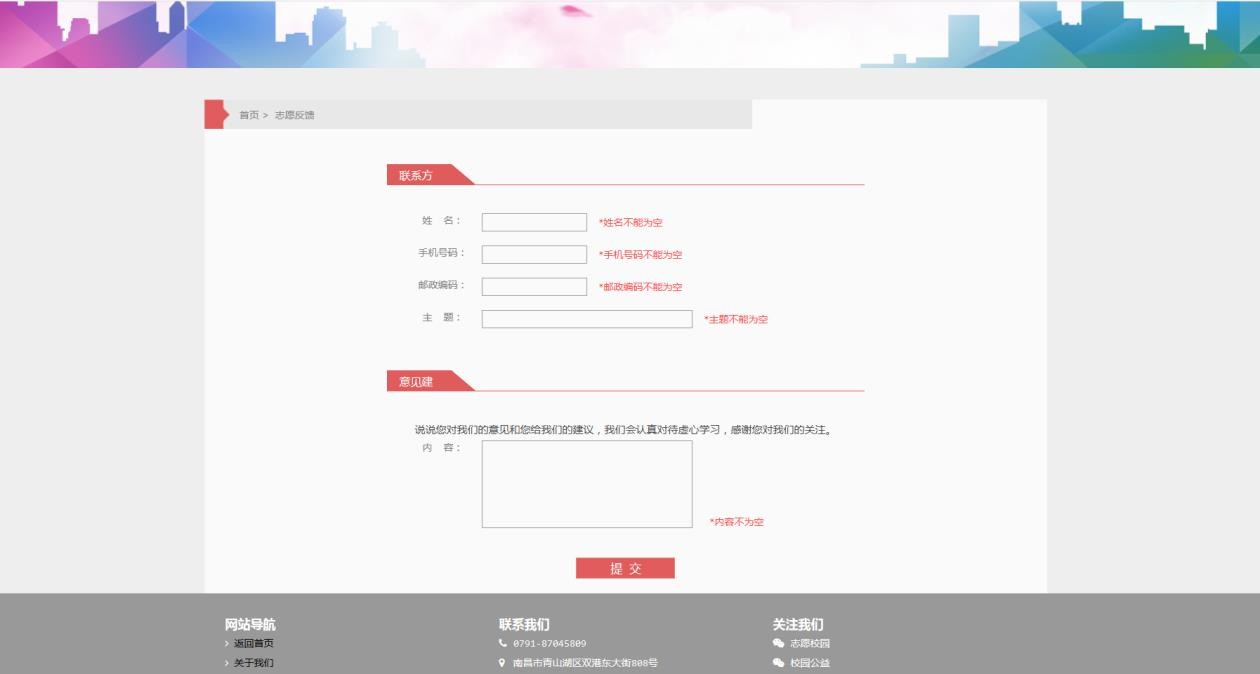
建议反馈界面是各用户填写建议反馈信息并提交，管理员可在后台查看建议反馈相关内容，如图 5-13 所示：

图 5-13 建议反馈界面

# 详细设计

### 系统主要功能模块介绍

身份类别管理：系统管理员对不同用户群体设定该用户相应的身份类别，管理员可对新加入志愿群体的用户添加该用户对应的身份类别，对用户与其身份类别不一致、错误的进行修改和删除身份类别。

人员标准管理：志愿服务发布者对发布志愿活动进行招募符合一定条件的志愿者的管理，新建人员标准以对特定的志愿服务活动进行招募志愿者，删除无用、不符合要求的人员标准，修改人员标准。

学院管理：系统（学校）管理员对学院管理员、学院名称相关信息的操作管理，创建学院管理员账户，学院管理员可对所在学院的所有志愿者进行统一管理，新增学院名称相关信息，修改有误的学院名称相关信息，删除不存在、更换学院名称的学院相关信息。

新闻资讯管理：志愿服务发布者添加、修改新闻资讯，查看新闻资讯详细信息，删除新闻资讯，已发布的新闻资讯相关内容在前台网站展示；志愿者、游客浏览志愿服务发布者已发布的新闻资讯相关内容。

政策文件管理：管理员添加需要上传的政策文件、删除政策文件、查看政策文件详细内容，政策文件相关的内容在前台网站展示；志愿者和游客浏览、下载管理员已上传的政策文件相关内容。

活动发布管理：志愿服务发布者添加发布志愿服务活动、修改、查看进行发布的志愿服务活动的详细信息，删除志愿服务活动，发布的志愿服务活动交由管理员审批，已审批通过的志愿服务活动相关内容在前台网站展示；志愿者、游客浏览已审批通过的志愿服务活动相关内容。

活动审批管理：系统管理员查看志愿服务发布者已发布的志愿服务活动列表相关信息，审批单个或批量审批志愿服务活动是否通过，志愿服务活动未能审批通过需填写拒绝理由。

活动报名管理：志愿者浏览查看管理员已审批通过的志愿服务活动相关内容，选择志愿活动提交报名申请参加志愿服务活动，可在志愿者后台查看个人已报名的志愿活动相关信息，志愿服务发布者可查看已报名参加该活动的志愿者信息和删除已报名的志愿者。

游客、用户对志愿服务平台管理、志愿服务活动管理等相关的事件提交建议反馈信息。管理员查看游客、用户已提交的建议反馈内容，管理员删除游客、用户提交的建议反馈。

爱心项目管理：系统管理员添加设置爱心项目相关信息，浏览已设置完成的所有爱

心项目详细信息，修改已添加设置完成的爱心项目相关信息，删除无用、过期等不符合要求的爱心项目。

### 用户管理

* + 1. 功能结构设计

用户管理主要是管理员添加用户，查看用户相关信息，修改用户相关信息和删除用户。用户管理包图如图 6-1 所示：

图 6-1 用户管理包图

* + 1. 类图设计

用户管理类图如图 6-2 所示：

Madmin

-feedbackMapper:

+addMadmin():

+getAllMadmin():

+updateMadmin():

+deleteMadmin():

-madminAllManage:

+addMadmin():

+getAllMadmin():

+updateMadmin():

+deleteMadmin():

MadminAllManage

MadminHandler

MadminMapper

+mid

+phone

+rank

+did

图 6-2 用户管理类图

用户管理类描述如表 6-1、表 6-2、表 6-3 所示：

表 6-1 用户管理控制层类描述

|  |  |  |
| --- | --- | --- |
| 用户管理控制层类 | | |
| 类的描述：用户管理控制层类 | | |
| 操作描述 | addMadmin() | 添加用户 |
| getAllMadmin() | 查看用户 |
| updateMadmin() | 修改用户 |
| deleteMadmin() | 删除用户 |

表 6-2 用户管理服务层类描述

|  |  |  |
| --- | --- | --- |
| 用户管理服务层类 | | |
| 类的描述：用户管理服务层类 | | |
| 操作描述 | addMadmin() | 添加用户 |
| getAllMadmin() | 查看用户 |
| updateMadmin() | 修改用户 |
| deleteMadmin() | 删除用户 |

表 6-3 用户类描述

|  |  |  |
| --- | --- | --- |
| 用户类 | | |
| 类的描述：用户实体类 | | |
| 属性描述 | mid | 管理员 Id |
| phone | 手机号 |
| rank | 级别 |
| did | 学院 Id |
| 操作 | Get() | 获取数据库中用户信息 |
| Set() | 设置数据库中用户信息 |

* + 1. 顺序图设计

删除用户顺序图如图 6-3 所示：



用户管理界面

用户管理控制层

用户管理业务层

用户管理数据访问层

1：删除用户

2：向业务对象进行删除操作

3：向数据对象进行删除操作

6：向界面返回操作结果

5：向控制层返回删除结果

4：向业务层返回删除结果

* + 1. 流程图设计

图 6-3 删除用户顺序图

删除用户活动图如图 6-4 所示：

点击选中删除

删除用户

是

保存数据库

是否继续删除

否



图 6-4 删除用户活动图

### 新闻资讯管理

* + 1. 功能结构设计

新闻资讯管理主要是志愿服务发布者添加、修改、查看新闻资讯详细信息，删除新闻资讯，已发布的新闻资讯相关内容在前台网站展示；管理员、志愿者浏览志愿服务发布者已发布的新闻资讯相关内容。新闻资讯管理包图如图 6-5 所示：

* + 1. 类图设计

图 6-5 新闻资讯管理包图

新闻资讯管理类图如图 6-6 所示：

NewslistMapper

+getAllNewList():

+getOneNew():

+pushNew():

+getMyNewList():

+deleteNews():

+allNews():

+oneNews():

+myNews():

+addOneNews():

+deleteNewsList():

-newsManage:NewsManage

-newMapper:NewMapper

-newslistMapper:NewslistMapper

NewsHandler

NewsManage

NewMapper

图 6-6 新闻资讯管理类图

+newsid

+title

+fabutime

+qid

+content

News

+qid

+newsid

+title

+fabutime

+name

+phone

Newslist

新闻资讯管理类描述如表 6-4、表 6-5、表 6-6、表 6-7 所示：

表 6-4 新闻资讯控制层类描述

|  |  |  |
| --- | --- | --- |
| 新闻资讯控制层类 | | |
| 类的描述：新闻资讯管理控制层类 | | |
| 操作描述 | addOneNews() | 添加新闻资讯 |
| updateNewsList() | 修改新闻资讯 |
| myNews() | 查看本人已发布的新闻资讯 |
| deleteNewsList() | 删除新闻资讯 |
| allNews() | 查看所有新闻列表 |
| oneNews() | 查看新闻详情 |

表 6-5 新闻资讯服务层类描述

|  |  |  |
| --- | --- | --- |
| 新闻资讯服务层类 | | |
| 类的描述：新闻资讯管理服务层类 | | |
| 操作描述 | getAllNewList() | 查看新闻列表 |
| getOneNew() | 查看新闻资讯详情 |
| pushNew() | 发布新闻资讯 |
| getMyNewList() | 查看本人发布的新闻资讯列表 |
| updateNews() | 修改新闻资讯 |
| deleteNews() | 删除新闻资讯 |

表 6-6 新闻资讯类描述

|  |  |  |
| --- | --- | --- |
| 新闻资讯类 | | |
| 类的描述：新闻资讯实体类 | | |
| 属性描述 | newsid | 新闻 Id |
| title | 标题 |
| fabutime | 发布时间 |
| content | 内容 |
| qid | 发布者 Id |
| 操作 | Get() | 获取数据库中新闻资讯信息 |
| Set() | 设置数据库中新闻资讯信息 |

表 6-7 新闻资讯列表类描述

|  |  |  |
| --- | --- | --- |
| 新闻资讯列表类 | | |
| 类的描述：新闻资讯列表实体类 | | |
| 属性描述 | newsid | 新闻 Id |
| title | 标题 |
| fabutime | 发布时间 |
| name | 姓名 |
| phone | 手机号 |
| qid | 发布者 Id |
| 操作 | Get() | 获取数据库中新闻资讯列表信息 |
| Set() | 设置数据库中新闻资讯列表信息 |

* + 1. 顺序图设计

查看新闻资讯顺序图如图 6-7 所示：



新闻资讯管理

界面

新闻资讯管理控制

层

新闻资讯管理业务

层

新闻资讯管理数据访

问层

1：请求查看新闻资讯

2：向业务层提交查找新闻资讯

3：向数据访问层提交查找新闻资讯

6：向界面返回操作结果

5：向控制层返回查询结果

4：向业务层返回查询结果

* + 1. 流程图设计

图 6-7 查看新闻资讯顺序图

查看新闻资讯活动图如图 6-8 所示：



输入关键字

在数据库中查询

否

是否存在符合的记录是

是

返回符合的新闻资讯

提示未能找到

是否继续查找否

图 6-8 查看新闻资讯活动图

### 政策文件管理

* + 1. 功能结构设计

政策文件管理主要是管理员添加、删除、查看政策文件详细内容，选择上传的政策文件，政策文件相关的内容在前台网站展示；志愿者、游客浏览、下载管理员已上传的政策文件相关内容。政策文件管理包图如图 6-9 所示：

图 6-9 政策文件管理包图

* + 1. 类图设计

政策文件管理类图如图 6-10 所示：

DocumentMapper

Document

-documentAllManage:

+getAllDocument():

+deleteDocument():

+addDocument():

-documentMapper:

+getAllDocument():

+addOneDocument():

+deleteOneDocument():

DocumentAllManage

DocumentHandler

+uuid

+name

+dtime

+downloads

图 6-10 政策文件管理类图

政策文件管理类描述如表 6-8、表 6-9、表 6-10 所示：

表 6-8 政策文件控制层类描述

|  |  |  |
| --- | --- | --- |
| 政策文件控制层类 | | |
| 类的描述：政策文件管理控制层类 | | |
| 操作描述 | addDocument() | 添加政策文件 |
| getAllDocument() | 查看政策文件列表 |
| deleteDocument() | 删除政策文件 |
| downloadDocument () | 下载政策文件 |

表 6-9 政策文件服务层类描述

|  |  |  |
| --- | --- | --- |
| 政策文件服务层类 | | |
| 类的描述：政策文件管理服务层类 | | |
| 操作描述 | addDocument() | 添加政策文件 |
| getAllDocument() | 查看政策文件列表 |
| deleteDocument() | 删除政策文件 |
| downloadDocument () | 下载政策文件 |

表 6-10 政策文件类描述

|  |
| --- |
| 政策文件类 |
| 类的描述：政策文件实体类 |

续表 6-10

|  |  |  |
| --- | --- | --- |
| 属性描述 | uuid | 政策文件 Id |
| name | 文件名 |
| dtime | 时间 |
| downloads | 下载量 |
| 操作 | Get() | 获取数据库中政策文件信息 |
| Set() | 设置数据库中政策文件信息 |

* + 1. 顺序图设计

添加政策文件顺序图如图 6-11 所示：



政策文件管理

界面

政策文件管理控制

层

1：请求添加上传文件

3：向数据访问层提交添加信息

4：向业务层返回添加结果

果

政策文件管理数据访

问层

政策文件管理业务

层

2：向业务层提交添加信息

6：向界面返回操作结果

* + 1. 流程图设计

5：向控制层返回添加结

图 6-11 添加政策文件顺序图

添加政策文件活动图如图 6-12 所示：

点击选中添加政策文件



添加政策文件

是

保存数据库

是否继续添加

否



图 6-12 添加政策文件活动图

### 活动发布管理

* + 1. 功能结构设计

活动发布管理主要是志愿服务发布者添加志愿服务活动、修改志愿服务活动、查看志愿服务活动的详细信息，删除志愿服务活动。活动发布包图如图 6-13 所示：

* + 1. 类图设计

图 6-13 活动发布管理包图

活动发布管理类图如图 6-14 所示：

volservicelistinfoMapper

+addService():

+getAllAnnouncedService():

+cancelAnnouncedService():

+updateServiceInfo():

+addOneNewService():

+approveServiceList():

+getMyAnnService():

+deleteMyAnnService():

-volservicelistinfoMapper:

-volserviceMapper:

-announceVolService:

-myAnnouncedServiceManage:

VolserviceMapper

MyAnnouncedServiceManage

VolServiceHandler

图 6-14 活动发布管理类图

+qid

+pid

+title

+place

+time

+state

+qrcode

Volservice

活动发布管理类描述如表 6-11、表 6-12、表 6-13 所示：

表 6-11 活动发布控制层类描述

|  |
| --- |
| 活动发布控制层类 |
| 类的描述：活动发布管理控制操作类，与前台界面交互的数据进行操作 |

续表 6-11

|  |  |  |
| --- | --- | --- |
| 操作描述 | addOneNewService() | 添加志愿活动 |
| allService() | 查看志愿活动 |
| updateMyAnnService() | 修改志愿活动 |
| deleteMyAnnService() | 删除志愿活动 |

表 6-12 活动发布服务层类描述

|  |  |  |
| --- | --- | --- |
| 活动发布服务层类 | | |
| 类的描述：包含活动发布管理各个功能的方法，用于实现前台数据和后台进行的相关交互操  作 ，与控制层交互的数据进行操作 | | |
| 操作描述 | addOneNewService() | 添加志愿活动 |
| allService() | 查看志愿活动 |
| updateMyAnnService() | 修改志愿活动 |
| deleteMyAnnService() | 删除志愿活动 |

表 6-13 志愿活动类描述

|  |  |  |
| --- | --- | --- |
| 志愿活动发布类 | | |
| 类的描述：志愿活动实体类 | | |
| 属性描述 | sid | 志愿服务 Id |
| title | 标题 |
| dtime | 时间 |
| vdocument | 相关文件 |
| state | 当前状态 |
| time | 时间 |
| place | 地点 |
| matdemand | 物资需求 |
| numdemand | 人数需求 |
| qrcode | 群聊二维码 |
| content | 详情简介 |
| qid | 发布者 Id |
| 操作 | Get() | 获取数据库中志愿活动信息 |
| Set() | 设置数据库中志愿活动信息 |

* + 1. 顺序图设计

添加志愿活动顺序图如图 6-15 所示：



活动发布管理业务层

1：请求添加志愿活动信息

6：向界面返回操作结果

活动发布管理控制

层

活动发布管理

界面

活动发布管理数据访问层

2：向业务层提交添加信息

3：向数据访问层提交添加信息

4：向业务层返回添加结果

5：向控制层返回添加结果

* + 1. 流程图设计

图 6-15 添加志愿活动顺序图

添加志愿活动活动图如图 6-16 所示：

点击选中添加志愿活动

添加志愿活动信息

是

保存数据库

是否继续添加否



图 6-16 添加志愿活动活动图

### 活动审批管理

* + 1. 功能结构设计

活动审批管理主要是系统管理员查看志愿服务发布者已发布的志愿服务活动信息，审

批单个或批量审批志愿服务活动是否通过，志愿服务活动未能审批通过拒绝。活动审批管理包图如图 6-17 所示：

* + 1. 类图设计

图 6-17 活动审批管理包图

活动审批管理类图如图 6-18 所示：

volservicelistinfoMapper

+getNeedapproveserviceList():

+approveService():

+refuse():

+getAllNeedApproveServiceInfoList():

+approveServiceList():

+unapproveServiceList():

-needapproveserviceMapper:

-VolserviceMapper:

-volServiceManage:

VolserviceMapper

VolServiceManage

VolServiceHandler

图 6-18 活动审批管理类图

+qid

+pid

+title

+place

+time

+state

+qrcode

Volservice

活动审批管理类描述如表 6-14、表 6-15 所示：

表 6-14 活动审批控制层类描述

|  |  |  |
| --- | --- | --- |
| 政策文件控制层类 | | |
| 类的描述：政策文件管理控制层类 | | |
| 操作描述 | getAllNeedApproveServiceInfoList() | 查看待审批志愿活动 |
| approveServiceList() | 审批通过志愿活动 |
| unapproveServiceList() | 审批拒绝志愿活动 |
| deleteServiceList() | 删除志愿活动 |

表 6-15 活动审批服务层类描述

|  |  |  |
| --- | --- | --- |
| 政策文件服务层类 | | |
| 类的描述：政策文件管理服务层类 | | |
| 操作描述 | getAllNeedApproveServiceInfoList() | 查看待审批志愿活动 |
| approveServiceList() | 审批通过志愿活动 |
| unapproveServiceList() | 审批拒绝志愿活动 |
| deleteServiceList() | 删除志愿活动 |

* + 1. 顺序图设计

删除志愿活动顺序图如图 6-19 所示：



活动审批管理

界面

活动审批管理控制

层

活动审批管理业务

层

活动审批管理数据访

问层

1：删除志愿活动

2：向业务对象进行删除操作

3：向数据对象进行删除操作

6：向界面返回操作结果

* + 1. 流程图设计

5：向控制层返回删除结果

图 6-19 删除志愿活动顺序图

4：向业务层返回删除结果

删除志愿活动活动图如图 6-20 所示：

点击选中删除



删除志愿活动

是

保存数据库

是否继续删除

否



图 6-20 删除志愿活动活动图

### 活动报名管理

* + 1. 功能结构设计

活动报名管理主要是志愿者选择添加报名申请信息参加志愿服务活动，查看个人报名相关信息，删除报名申请。活动报名管理包图如图 6-21 所示：

* + 1. 类图设计

图 6-21 活动报名管理包图

活动报名管理类图如图 6-22 所示：

EnrollVolInfo

+applyService():

+allService():

+deleteEnroll():

+applyService():

+allService():

+deleteEnroll():

-enrollMapper:

-enrollManage:

EnrollMapper

EnrollManage

VolServiceHandler

+vid

+name

+sex

+age

+phone

+time

+collegename

+sid

图 6-22 活动报名管理类图

活动报名管理类描述如表 6-16、表 6-17、表 6-18 所示：

表 6-16 活动报名控制层类描述

|  |
| --- |
| 活动报名控制层类 |
| 类的描述：活动报名管理控制层类 |

续表 6-16

|  |  |  |
| --- | --- | --- |
| 操作描述 | applyService() | 添加报名申请 |
| allService() | 查看报名信息 |
| deleteEnroll() | 删除报名申请 |

表 6-17 活动报名服务层类描述

|  |  |  |
| --- | --- | --- |
| 活动报名服务层类 | | |
| 类的描述：活动报名管理服务层类 | | |
| 操作描述 | applyService() | 添加报名申请 |
| allService() | 查看报名信息 |
| deleteEnroll() | 删除报名申请 |

表 6-18 报名类描述

|  |  |  |
| --- | --- | --- |
| 报名类 | | |
| 类的描述：报名实体类 | | |
| 属性描述 | vid | 志愿者 Id |
| name | 姓名 |
| stunumber | 学号 |
| sex | 性别 |
| age | 年龄 |
| phone | 手机号 |
| time | 时间 |
| collegeName | 学院名称 |
| sid | 志愿服务 Id |
| 操作 | Get() | 获取数据库中报名信息 |
| Set() | 设置数据库中报名信息 |

* + 1. 顺序图设计

添加活动报名顺序图如图 6-23 所示：



活动报名管理

界面

活动报名管理控制

层

活动报名管理业务

层

活动报名管理数据访

问层

1：请求添加报名申请

2：向业务层添加报名申请

3：向数据访问层添加报名申请

6：向界面返回添加结果

5：向控制层返回添加结果

4：向业务层返回添加结果

* + 1. 流程图设计

图 6-23 添加活动报名顺序图

添加报名申请活动图如图 6-24 所示：

点击选中添加报名申请



添加报名申请

是

保存数据库

是否继续添加否



图 6-24 添加报名申请活动图

### 建议反馈管理

* + 1. 功能结构设计

建议反馈管理主要是各用户对志愿服务平台管理、志愿服务活动管理等相关的事件提交建议反馈信息。用户添加建议反馈，志愿者查看本人已提交的建议反馈内容，管理员查

看用户已提交的建议反馈内容。建议反馈管理包图如图 6-25 所示：

* + 1. 类图设计

图 6-25 建议反馈管理包图

建议反馈管理类图如图 6-26 所示：

Feedback

-feedbackMapper:

+submitFeedback():

+getAllFeedback():

-feedbackAllManage:

+addMsg():

+getAllFeedback():

FeedbackAllManage

FeedbackHandler

FeedbackMapper

+id

+time

+call

+phone

+content

图 6-26 建议反馈管理类图

建议反馈管理类描述如表 6-19、表 6-20、表 6-21 所示：

表 6-19 建议反馈控制层类描述

|  |  |  |
| --- | --- | --- |
| 建议反馈控制层类 | | |
| 类的描述：建议反馈管理控制层类 | | |
| 操作描述 | addFeedback() | 添加建议反馈 |
| getAllFeedback () | 查看建议反馈 |
| deleteFeedback() | 删除建议反馈 |

表 6-20 建议反馈服务层类描述

|  |  |  |
| --- | --- | --- |
| 建议反馈服务层类 | | |
| 类的描述：建议反馈管理服务层 | | |
| 操作描述 | addFeedback() | 添加建议反馈 |
| getAllFeedback () | 查看建议反馈 |
| deleteFeedback() | 删除建议反馈 |

表 6-21 建议反馈类描述

|  |  |  |
| --- | --- | --- |
| 建议反馈类 | | |
| 类的描述：建议反馈实体类 | | |
| 属性描述 | id | Id |
| time | 时间 |
| call | 称呼 |
| phone | 电话 |
| content | 内容 |
| 操作 | Get() | 获取数据库中建议反馈信息 |
| Set() | 设置数据库中建议反馈信息 |

* + 1. 顺序图设计

查看建议反馈顺序图如图 6-27 所示：



建议反馈管理业务

建议反馈管理控制

层

建议反馈管理

界面

层

建议反馈管理数据访

问层

1：请求查看反馈信息

2：向业务层查找反馈信息

3：向数据访问层查找反馈信息

6：向界面返回操作结果

5：向控制层返回查询结果

4：向业务层返回查询结果

图 6-27 查看建议反馈顺序图

* + 1. 流程图设计

查看建议反馈活动图如图 6-28 所示：

输入关键字



在数据库中查询

否

是否存在符合的记录是

是 返回符合的建议反馈信息

提示未能找到

是否继续查找

否



图 6-28 查看建议反馈活动图

### 爱心项目管理

* + 1. 功能结构设计

爱心项目管理主要是系统管理员添加设置爱心项目，查看已添加的所有爱心项目详细信息，修改爱心项目相关信息，删除爱心项目。爱心项目管理包图如图 6-29 所示：

* + 1. 类图设计

图 6-29 爱心项目管理包图

爱心项目管理类图如图 6-30 所示：

AiprojectMapper

+aid

+title

+photo

+ainumber

图 6-30 爱心项目管理类图

爱心项目管理类描述如表 6-22、表 6-23、表 6-24 所示：

Aiproject

-aiprojectMapper:

+deleteOneAi():

+getAllAiProject():

+addAiProject():

+updateAiProject():

-aiProjectAllManage:

+getaiProInfo():

+getAiList():

+deleteAi

+addAi

AiProjectAllManage

AiProjectHandler

表 6-22 爱心项目控制层类描述

|  |  |  |
| --- | --- | --- |
| 爱心项目控制层类 | | |
| 类的描述：爱心项目管理控制层类 | | |
| 操作描述 | addAi() | 添加爱心项目 |
| getAiList() | 查看爱心项目 |
| updateAi() | 修改爱心项目 |
| deleteAi() | 删除爱心项目 |

表 6-23 爱心项目服务层类描述

|  |  |  |
| --- | --- | --- |
| 爱心项目服务层类 | | |
| 类的描述：爱心项目管理服务层类 | | |
| 操作描述 | addAi() | 添加爱心项目 |
| getAiList() | 查看爱心项目 |
| updateAi() | 修改爱心项目 |
| deleteAi() | 删除爱心项目 |

表 6-24 爱心项目类描述

|  |  |  |
| --- | --- | --- |
| 爱心项目类 | | |
| 类的描述：爱心项目实体类 | | |
|  | aid | 爱心项目 Id |

续表 6-24

|  |  |  |
| --- | --- | --- |
| 属性描述 | title | 标题 |
| photo | 图片 |
| aiNumber | 总需求爱心 |
| preNumber | 目前爱心数 |
| 操作 | Get() | 获取数据库中爱心项目信息 |
| Set() | 设置数据库中爱心项目信息 |

* + 1. 顺序图设计

删除爱心项目顺序图如图 6-31 所示：



爱心项目管理控制

爱心项目管理

界面

层

爱心项目管理业务

层

爱心项目管理数据访

问层

1：删除爱心项目

2：向业务对象进行删除操作

3：向数据对象进行删除操作

6：向界面返回操作结果

5：向控制层返回删除结果

4：向业务层返回删除结果

* + 1. 流程图设计

图 6-31 删除爱心项目顺序图

删除爱心项目活动图如图 6-32 所示：

点击选中删除



删除爱心项目

是

保存数据库

是否继续删除否

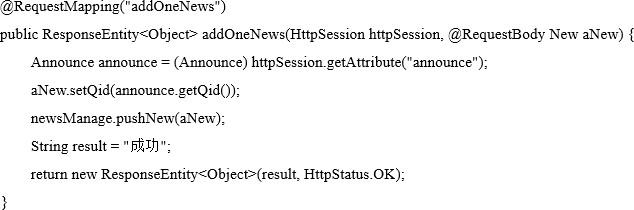


图 6-32 删除爱心项目活动图

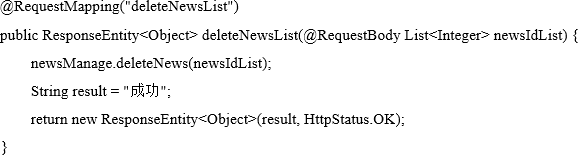
# 编码

### 代码实现与核心算法

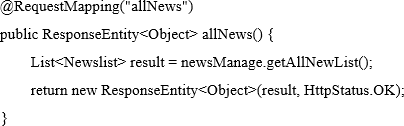
1. 添加新闻资讯：



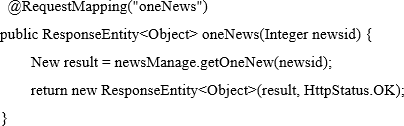
1. 删除新闻资讯：



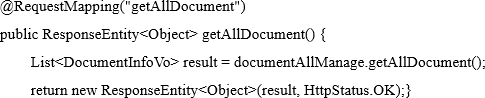
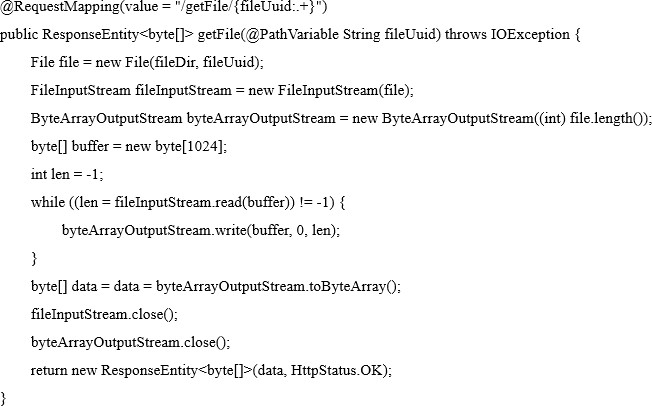
1. 查看所有新闻资讯列表：



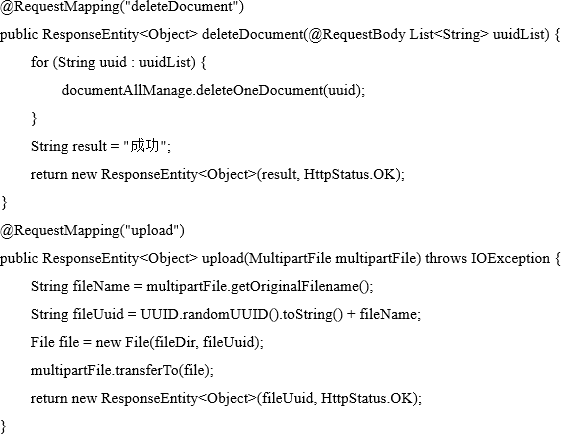
1. 查看新闻资讯详情：



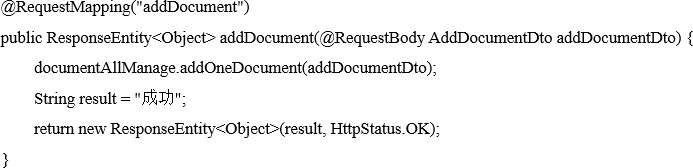
1. 查看政策文件列表：



1. 删除政策文件：



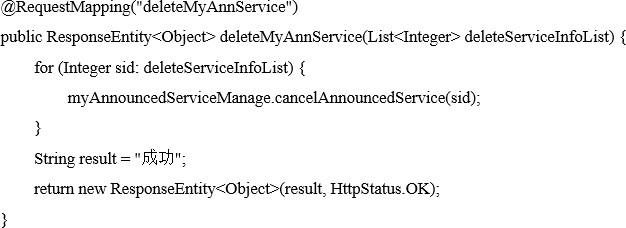
1. 上传政策文件：
2. 添加政策文件：



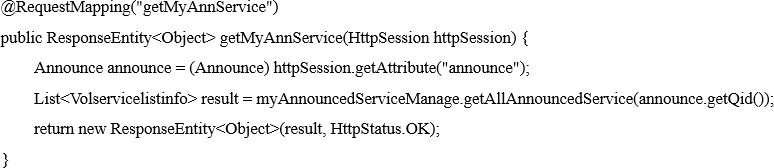
1. 发布志愿活动：



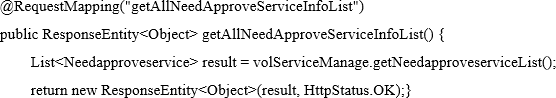
1. 删除志愿活动：



1. 查看自己发布的志愿活动：



1. 查看待审批的志愿活动：



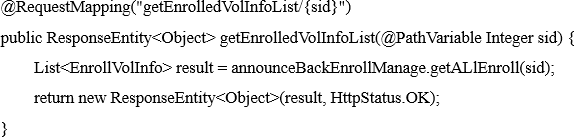
1. 审批通过志愿活动：



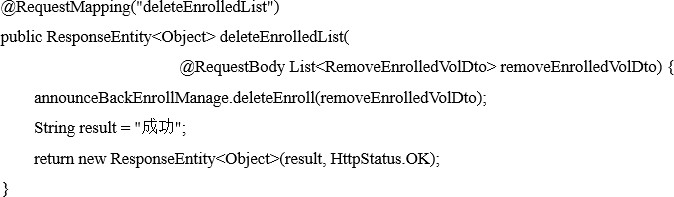
1. 审批拒绝志愿活动：



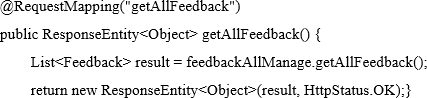
1. 查看志愿者报名列表：



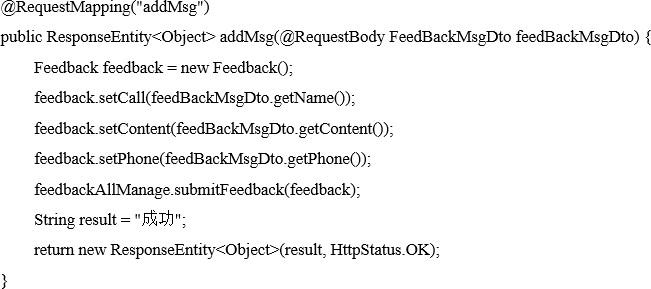
1. 删除报名志愿者：



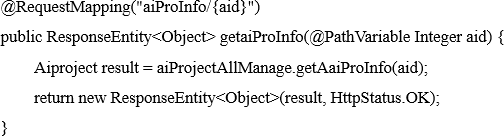
1. 查看建议反馈：



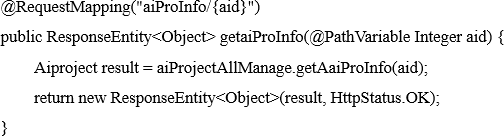
1. 添加建议反馈：



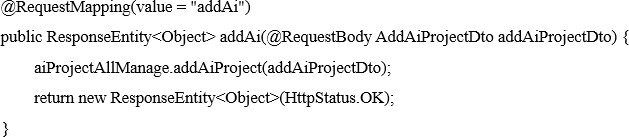
1. 查看爱心项目：



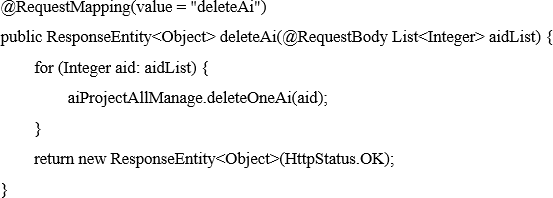
1. 查看爱心项目列表：



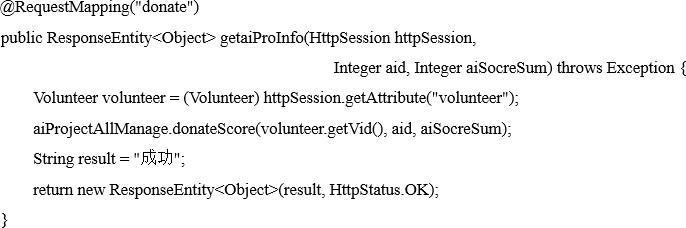
1. 添加爱心项目：



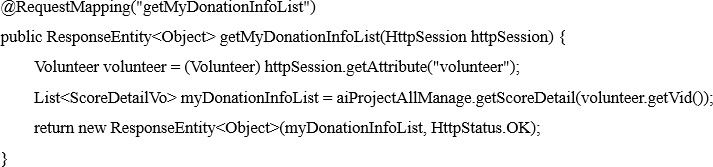
1. 删除爱心项目：



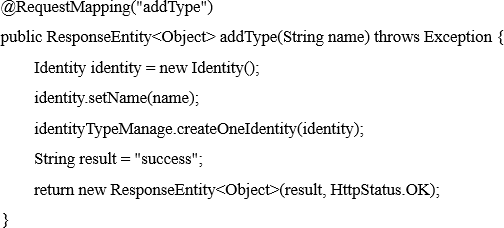
1. 捐献爱心积分：



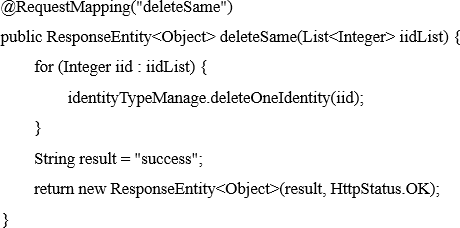
1. 查看捐献积分列表：



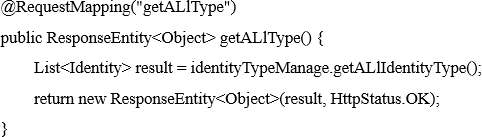
1. 添加身份类别：



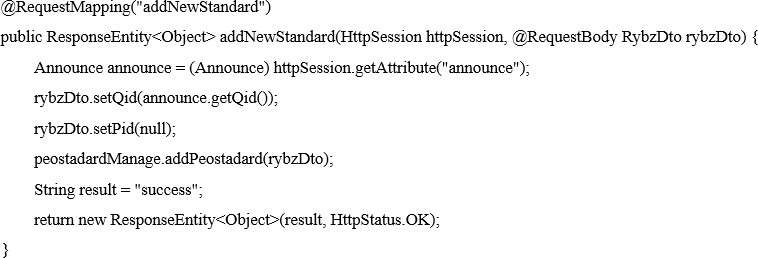
1. 删除身份类别：



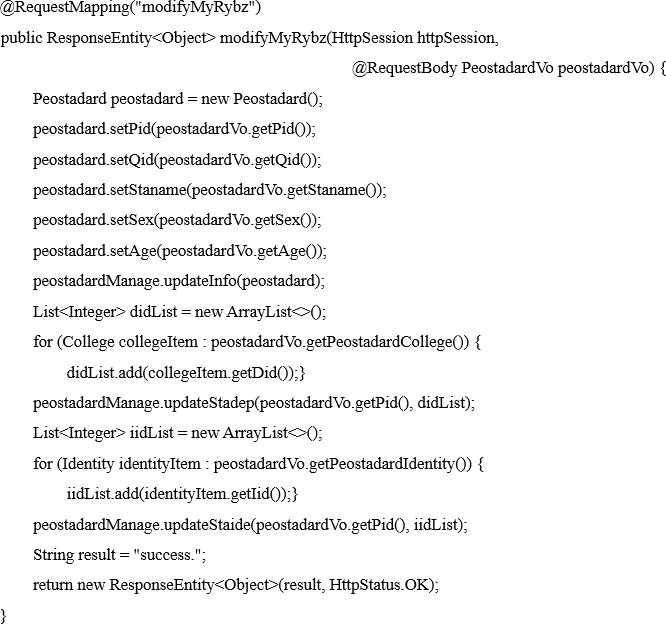
1. 查看身份类别：



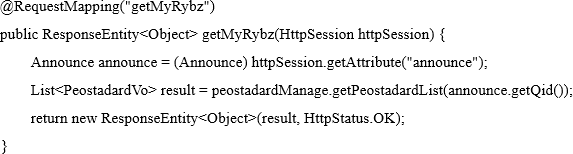
1. 添加人员标准：



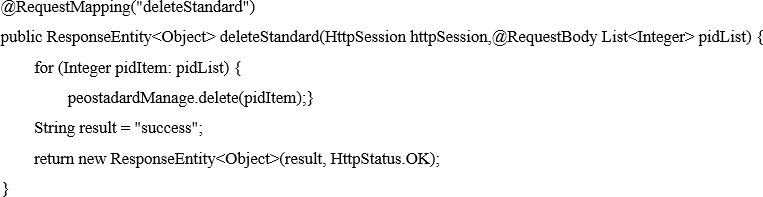
1. 修改人员标准：



1. 查看人员标准：



1. 删除人员标准：



### 代码优化分析

系统采用 SSM 框架进行开发，根据 Spring 的 IOC 和 AOP 这些特性，使得 Spring 统一管理各个对象间的依赖关系，方便解耦，同时在软件开发的过程中也更加简便，在集中和控制的各个方面对重复模块进行管理，提供了对其他优秀开源框架的集成支持。SpringMVC 对 web 层进行解耦，与 Spring 无缝衔接并且在软件开发的过程中更加简洁， 对数据验证处理的灵活化，数据绑定机制 MyBatis 中采用便捷的 XML 或注解来处理在原始映射和相关的配置上，并用接口和 Java 的对象映射成数据库中相对应的记录。

时间效率优化，对志愿服务平台中消息通知管理的数据频繁更新，采用将数据存放在内存中，加快数据处理速度。系统架构部署采用前后端分离技术，在数据处理工作上，前端 JS 发挥很大的作用，可以使得服务器压力减轻以及加快用户打开的速度。

# 测试

### 测试方案设计

* + 1. 测试策略

功能测试的测试目的是对志愿服务平台的核心功能进行测试，在功能测试过程中对志愿服务平台所涉及的功能进行详细分析，对不同的模块选择合适的测试方法进行测试，在测试过程中遇到的问题妥善解决，进而保障志愿服务平台的正常运行。如表 8-1 所示：

表 8-1 功能测试策略表

|  |  |
| --- | --- |
| 测试目的 | 对志愿服务平台的用户管理、新闻资讯管理、政策文件管理、志愿活动管理（包括志愿活动发布、志愿活动审批、志愿活动报名）、建议反馈  管理进行测试，保证各个功能测试正常。 |
| 测试范围 | 用户管理、新闻资讯管理、政策文件管理、志愿活动管理（包括志愿活  动发布、志愿活动审批、志愿活动报名）、建议反馈管理 |
| 开始标准 | 各个用户登录志愿服务平台 |
| 结束标准 | 执行的各个功能全部完成测试  处理解决发现的缺陷 |
| 测试方法 | 对各个功能都采用有效、无效两类数据进行测试： 输入有效的数据将得到预期结果。  输入无效的数据显示相应的提示信息。 |

系统测试的测试目的是对系统的业务流程进行测试，测试其是否合乎志愿服务的实际需求，测试其是否存在不合乎逻辑、不合理或错误的流程。如表 8-2 所示：

表 8-2 系统测试策略表

|  |  |
| --- | --- |
| 测试目的 | 对系统的业务流程进行测试 |
| 测试范围 | 系统中已明确的业务流程 |
| 开始标准 | 集成测试时已完成所要求达到的标准 |
| 结束标准 | 执行的测试全部完成  处理解决发现的缺陷 |
| 测试方法 | 对各个功能都采用有效、无效两类数据进行测试： 输入有效的数据将得到预期结果。  输入无效的数据显示相应的提示信息。 |

* + 1. 测试进度安排

测试进度安排表如表 8-3 所示：

表 8-3 测试进度安排表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试阶段 | 测试范围 | 开始时间 | 结束时间 |
| 功能测试 | 用户管理  新闻资讯管理政策文件管理  志愿活动管理 | 2020 年 4 月 12 日 | 2020 年 4 月 17 日 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 建议反馈管理 |  |  |

续表 8-3

|  |  |  |  |
| --- | --- | --- | --- |
| 测试阶段 | 测试范围 | 开始时间 | 结束时间 |
| 系统测试 | 系统业务流程合乎逻辑 | 2020 年 4 月 18 日 | 2020 年 4 月 19 日 |
| 性能测试 | 多用户同时访问平台并进行  相应的操作 | 2020 年 4 月 20 日 | 2020 年 4 月 20 日 |
| 兼容测试 | 系统在各个软件平台中运行  无冲突 | 2020 年 4 月 21 日 | 2020 年 4 月 21 日 |

* + 1. 测试资源

1、测试环境：

硬件环境：windows10，软件环境：Tomcat，mysql，Firefox Browser 2、测试工具：

Mozilla Firefox 是一个方便、开放源代码的网页浏览器，排版引擎使用的是 Gecko，并且支持多种操作系统，如Windows、Mac OS X 及 GNU/Linux 等。

* + 1. 关键测试点

志愿活动管理过程中对志愿活动发布、志愿活动审批、志愿活动报名的相应操作测试。

### 测试用例构建

* + 1. 测试用例编写约定

用例编号：每个测试用例拥有唯一标识用例名称：测试用例的简要描述

前置条件：测试用例所需的前提条件测试步骤：测试用例所需的各个操作

预期结果：用例测试完成后预期的显示结果输出结果：输出的显示结果

测试结果：用例测试正常通过、未通过

* + 1. 测试用例设计

查看用户测试用例表如表 8-4 所示：

表 8-4 查看用户测试用例表

|  |  |
| --- | --- |
| 用例编号 | YH02 |
| 用例名称 | 查看用户 |
| 前置条件 | 管理员进入查看用户界面 |
| 测试步骤 | 点击用户管理界面中“查看”按钮 |
| 预期结果 | 弹出用户详情界面，显示用户信息 |
| 输出结果 | 和预期结果一致 |

|  |  |
| --- | --- |
|  |  |
| 测试结果 | 通过 |

删除新闻资讯测试用例表如表 8-5 所示：

表 8-5 删除新闻资讯测试用例表

|  |  |
| --- | --- |
| 用例编号 | XW01 |
| 用例名称 | 删除新闻资讯 |
| 前置条件 | 志愿服务发布者进入新闻管理界面 |
| 测试步骤 | 点击新闻管理界面中“删除”按钮 |
| 预期结果 | 成功删除 |
| 输出结果 | 和预期结果一致 |
| 测试结果 | 通过 |

上传政策文件测试用例表如表 8-6 所示：

表 8-6 上传政策文件测试用例表

|  |  |
| --- | --- |
| 用例编号 | ZC01 |
| 用例名称 | 上传政策文件 |
| 前置条件 | 管理员进入添加文件界面 |
| 测试步骤 | 1、文件名为空  2、上传时间为空  3、选择文件为空 |
| 预期结果 | 1、提示文件名不能为空  2、提示上传时间不能为空  3、提示选择文件不能为空 |
| 输出结果 | 和预期结果一致 |
| 测试结果 | 通过 |

审批志愿活动测试用例表如表 8-7 所示：

表 8-7 审批志愿活动测试用例表

|  |  |
| --- | --- |
| 用例编号 | HD03 |
| 用例名称 | 审批志愿活动 |
| 前置条件 | 管理员进入待审批志愿活动界面 |
| 测试步骤 | 点击待审批志愿活动详情界面审批操作（通过/拒绝）按钮 |
| 预期结果 | 弹出审批操作结果 |
| 输出结果 | 和预期结果一致 |
| 测试结果 | 通过 |

查看建议反馈测试用例表如表 8-8 所示：

表 8-8 查看建议反馈测试用例表

|  |  |
| --- | --- |
| 用例编号 | FK02 |

|  |  |
| --- | --- |
| 用例名称 | 查看建议反馈信息 |
| 用例编号 | FK02 |

续表 8-8

|  |  |
| --- | --- |
| 前置条件 | 管理员进入查看建议反馈界面 |
| 测试步骤 | 点击建议反馈管理界面中“查看”按钮 |
| 预期结果 | 弹出建议反馈详情界面，显示建议反馈信息 |
| 输出结果 | 和预期结果一致 |
| 测试结果 | 通过 |

* + 1. 关键测试用例

发布志愿活动测试用例表如表 8-9 所示：

表 8-9 发布志愿活动测试用例表

|  |  |
| --- | --- |
| 用例编号 | HD01 |
| 用例名称 | 发布志愿活动 |
| 前置条件 | 志愿服务发布者进入发布志愿活动界面 |
| 测试步骤 | 1、活动标题为空  2、活动时间为空  3、活动地点为空  4、详情简介为空 |
| 预期结果 | 1、活动标题不能为空  2、活动时间不能为空  3、活动地点不能为空  4、详情简介不能为空 |
| 输出结果 | 和预期结果一致 |
| 测试结果 | 通过 |

* + 1. 测试用例维护

（一）及时删除冗余的测试用例

在设计测试过程时，测试用例可能会有两个或两个以上相同的测试用例的内容，为减少回归测试的效率，所以定期组织测试用例及时消除冗余。

（二）增加新的测试用例

随着需求变化以及版本使用情况的发展，测试用例遗漏或缺陷的因素使得原来的测试并不能完全涵盖软件的需求，因此需要添加新的测试用例。

（三）改进测试用例

在测试用例增加的某些情况下系统输入和当前状态的变化而变得不再适用，这些用例

都难以再利用，影响回归测试的效率，需要改善，使得可以重复使用以及可以控制。

# 总结与展望

### 设计工作总结

本文基于 SSM 框架的基础上完成了志愿服务平台的设计与实现。主要以高校学生群体为志愿者、志愿服务发布者、院校两级负责人为管理员三种用户角色的志愿服务活动管理系统，在志愿服务平台的相关设计与实现期间，在前期的广泛的查阅相关资料、文献， 参考高校志愿服务平台目前现状，从大学生青年团体进行志愿服务的出发角度对志愿服务平台进行各个阶段详细的工作，包括需求分析、系统设计、系统实现、系统测试等。

志愿服务平台主要有以下几个模块：系统管理模块、新闻资讯管理、政策文件管理、志愿活动管理、志愿者管理、建议反馈管理、爱心项目管理。系统管理主要是对用户、身份类别、人员标准、学院的相关操作，志愿活动管理主要是对活动发布、活动审批、活动报名的相关操作，爱心项目管理主要是爱心项目、爱心积分的相关操作。

志愿服务平台的搭建能够方便管理员、志愿服务发布者的管理工作，减轻任务量，提供更加便捷的服务，同时也对志愿服务相关的新闻资讯、政策文件、志愿活动、爱心项目进行统一管理，使得志愿者、普通游客可以及时、高效的获取信息，有利于提高实施志愿服务活动的效率，有利于进一步推动高校志愿服务管理工作的蓬勃发展以及弘扬真善美、传播正能量的具体实施情况。

### 未来工作展望

志愿服务平台的建设将为志愿者、志愿服务发布者、管理员三者之间的信息沟通、服务管理搭建了一个桥梁，方便志愿服务活动、志愿者的相关管理，也促进高校志愿服务发布、动态的展示，提升高校志愿服务资源的协调、分配能力，推动高校志愿服务工作的进一步发展。

志愿服务平台也还存在很多不足以及改进之处，如在志愿服务统计、志愿者培训等方面需要实现和完善，志愿服务活动类型、志愿者调度都需要进行进一步探索和拓展，让更多的大学生志愿者充分地参加志愿服务活动，并积极参与志愿服务平台的建设，使得志愿服务平台能为用户提供更加优质的服务。

# 谢 辞

# 参考文献

[1]籍慧文. Web 应用开发中 JAVA 编程语言的应用探讨[J].科技创新与应用，2017,07:90. [2]肖成金，吕冬梅. Java 程序开发数据库与框架应用[J].科技展望，2017,05:19.

[3]周宝亮，王敏. 计算机软件开发中 JAVA 编程语言的应用[J].电子技术与软件工程，2017,03:61-62. [4]杜丁超. 计算机软件 Java 编程特点及其技术分析[J].电脑知识与技术,2017,13(36):215-216.

[5]蔡创. 计算机软件开发中 JAVA 编程语言的分析和思考[J].信息技术与信息化,2017(12):80-81. [6]于宝洋. 计算机软件 Java 编程特点及其技术的思考[J].南方农机,2017,48(23):123+127.

[7]邢俊鑫. 基于 Java 编程语言的计算机软件开发技术[J].电子技术与软件工程,2017(23):47-48. [8]蒋银. 计算机软件 Java 编程特点及其技术研究[J].通讯世界,2017(21):51-52.

1. 杨兰. 计算机软件开发的 JAVA 编程语言及其实际应用[J].电子设计工程,2017,25(21):49-52+56.
2. 成孝俊. 计算机软件开发中 JAVA 编程语言的应用分析与研究[J].数字技术与应用,2017(10):166-168. [11]陆敏. 基于高校志愿服务活动常态化建设的思考[J].产业与科技论坛，2017，16(24):241-243
3. 谭建光. 中国新时代青年志愿服务的发展分析[J]. 青年学报, 2017, 01.
4. 王宏序. “志愿者之家”公益平台的网络志愿者的管理机制研究[D].东华大学,2017: [14]孙伟. 高校志愿服务活动长效机制研究[J].黑河学院学报,2018,9(6):80-81.
5. 康健，孟杰. 浅析在青年学生中开展志愿服务活动的意义[J].文化创新比较研究. 2019(33)
6. 张新颖. 大学生志愿服务发展历程回顾与展望[J].高校辅导员学刊. 2019(05)
7. 周国华. 基于 SSM 框架学习平台的设计与实现[J].信息与电脑,2017,0(24):138-139.
8. 邹红霆. 基于SSM 框架的Web 系统研究与应用[J].湖南理工学院学报：自然科学版,2017,30(1):39-43. [19]贺雪梅. Web 应用开发中的 SSM 框架设计[J].电子世界,2019,0(1):206-206.
9. 刘义忠,张伟. 基于 SSM 框架的后台管理系统设计与实现[J].软件导刊,2019,18(2):68-71.
10. Kim T. Briggs,Baoguo Zhou,Gerhard W. Dueck. Cold object identification in the Java virtual machine[J]. Softw. Pract. Exper.,2017,47（1）：.

[22]S. Jayaraman,B. Jayaraman,D. Lessa. Compact visualization of Java program execution[J]. Softw. Pract.

Exper.,2017,47（2）：.

1. Geoffrey Fox. Java Technologies for Real‐Time and Embedded Systems （JTRES2013）[J]. Concurrency Computat.: Pract. Exper.,2017,29（6）：.
2. Bambang Riadi,Ahmad Budiman Suriadi. Identification of paddy field using Landsat image in Karawang Regency, West Java[J]. IOP Conference Series: Earth and Environmental Science,2017,54（1）：.

[25]J T Nugroho,Zylshal,G A Chulafak,D Kushardono. Performance of LAPAN-A2 satellite data to classify

land cover/land use in Semarang, Central Java[J]. IOP Conference Series: Earth and Environmental Science,2017,54（1）：.

1. Abdi Sukmono,Ardiansyah. Identification of rice field using Multi-Temporal NDVI and PCA method on

Landsat 8 （Case Study: Demak, Central Java）[J]. IOP Conference Series: Earth and Environmental Science,2017,54（1）：.

1. Kreshna Dwi Nugroho,Singgih Pebrianto,Muhammad Arif Fatoni,Alvin Fatikhunnada,Liyantono,Yudi Setiawan. Comparison between wavelet transform and moving average as filter method of MODIS imagery to recognize paddy cropping pattern in West Java[J]. IOP Conference Series: Earth and Environmental Science,2017,54（1）：.
2. Abdelhak Mesbah,Jean-Louis Lanet,Mohamed Mezghiche. Reverse engineering a Java Card memory

management algorithm[J]. Computers &amp; Security,2017,66:.

1. Volunteer Involvement and Organizational Performance: The Use of Volunteer Officers in Public Safety[J]. Seong C. Kang. Public Performance & Management Review. 2019(3)
2. Police volunteers in the Netherlands: a study on policy and practice[J]. Ronald van Steden,Shanna M. Mehlbaum. Policing and Society. 2019(4)
3. Voluntary sensemaking: the identity formation process of volunteers in sport organizations[J]. Christine E. Wegner,Gareth J. Jones,Jeremy S. Jordan. European Sport Management Quarterly. 2019(5)

## 附录A 外文翻译—原文部分

Introduction to Java autor：Martin Ngobye.

source：Computing Static Slice for Java Programs

Java is designed to meet the challenges of application development in the context of heterogeneous, network-wide distributed environments. Paramount among these challenges is secure delivery of applications that consume the minimum of system resources, can run on any hardware and software platform, and can be extended dynamically.

Java originated as part of a research project to develop advanced software for a wide variety of network devices and embedded systems. The goal was to develop a small, reliable, portable, distributed, real-time operating platform. When the project started, C++ was the language of choice. But over time the difficulties encountered with C++ grew to the point where the problems could best be addressed by creating an entirely new language platform. Design and architecture decisions drew from a variety of languages such as Eiffel, SmallTalk, Objective C, and Cedar/Mesa. The result is a language platform that has proven ideal for developing secure, distributed, network based end-user applications in environments ranging from network-embedded devices to the World-Wide Web and the desktop.

The design requirements of Java are driven by the nature of the computing environments in which software must be deployed.

The massive growth of the Internet and the World-Wide Web leads us to a completely new way of looking at development and distribution of software. To live in the world of electronic commerce and distribution, Java must enable the development of secure, high performance, and highly robust applications on multiple platforms in heterogeneous, distributed networks.

Operating on multiple platforms in heterogeneous networks invalidates the traditional schemes of binary distribution, release, upgrade, patch, and so on. To survive in this jungle, Java must be architecture neutral, portable, and dynamically adaptable.

The Java system that emerged to meet these needs is simple, so it can be easily programmed by most developers; familiar, so that current developers can easily learn Java; object oriented, to take advantage of modern software development methodologies and to fit into distributed client-server applications; multithreaded, for high performance in applications that need to perform multiple concurrent activities, such as multimedia; and interpreted, for maximum portability and dynamic capabilities.

Together, the above requirements comprise quite a collection of buzzwords, so let’s examine some of them and their respective benefits before going on.

What’s completely new is the manner in which Java and its run-time system have combined

them to produce a flexible and powerful programming system..

Developing your applications using Java results in software that is portable across multiple machine architectures, operating systems, and graphical user interfaces, secure, and high performance, With Java, your job as a software developer is much easier—you focus your full attention on the end goal of shipping innovative products on time, based on the solid foundation of Java. The better way to develop software is here, now, brought to you by the Java language platform.

Very dynamic languages like Lisp, TCL, and SmallTalk are often used for prototyping. One of the reasons for their success at this is that they are very robust—you don’t have to worry about freeing or corrupting memory.

Similarly, programmers can be relatively fearless about dealing with memory when programming in Java, The garbage collection system makes the programmer’s job vastly easier; with the burden of memory management taken off the programmer’s shoulders, storage allocation errors go away. Another reason commonly given that languages like Lisp, TCL, and SmallTalk are good for prototyping is that they don’t require you to pin down decisions early on—these languages are semantically rich.

Java has exactly the opposite property: it forces you to make explicit choices. Along with these choices come a lot of assistance—you can write method invocations and, if you get something wrong, you get told about it at compile time. You don’t have to worry about method invocation error.

The Java beginner must grasp 30 basic concepts, Basic concept.

* 1. In OOP the only relations is what the object’s interface, such as the computer seller he leaves alone internal structure of electrical source, but he is only concerned about that whether the power will be supplied to you, also so long as know can or not but is not how and why.All procedures are make up of certain attribute and the behavior object, the different object visit invokes through the function invocation, between the object all intercommunion are invoke through the method invocation, By object data encapsulation, enhances reuse rate very much..
  2. In OOP the most important thought is class, the class is the template ,is a blueprint, construct an object from a class, namely construct an instance of the class.
  3. Encapsulation: is that combines the data and the behavior in a package) and hides the data the realization process to the object user, in an object data is called its instance field.
  4. Through expands a class to obtain a new class is called inheritance, but all classes are constructed by the object super root class of expansion, super root class of as follows can make the introduction.
  5. Object 3 principal characteristics

Behavior--- explained this object can make what.

Tate--- when the object exerts the method object reflection.

Dentity--- and other similar behavior objects discrimination symbols.

Each object has only indentity and among three characteristics they affect mutually.

* 1. Relations among classes:

Use-a:Dependent relation

Has-a:Polymerization relation

Is-a: inheritor relation -- example:A class has inherited B class, this time A class not only has B class of method, but also has its own method(Individuality exists in general character) 7.Structure object use structure:Structure proposing, the structure is one special method, the

structure object and to its initialization.

Example:A Data class of structure calls Data

New Data () --- structure a new object, also initialize current time.

Data happyday=new Data () --- an object evaluates an variable happyday, thus enables this object to be used many times, here be stated the cause variable and the object variable are different.

New returns the value is a quotation.

Constructor characteristic:The constructor may have 0, one or many parameters

The constructor and the class have the same name

A class may have many constructor. The constructor has not returned value

The constructor always be together used with the new operator

1. Over loading: When many methods have the same name when includes the different parameter, then has the over loading Which method does the compiler have to choose invokes.
2. Package : Java allow one or many classes to become together as group, is called package, to organizing duty easily, the standard Java storehouse divides into many packages Java.Lang java.Util java, net and so on, the package is layered and all java packages are in java and in a javax package.
3. Extendable thought: permit to construct new class on existing classes , when you extend the class which already existed, then you reuse this class of method and the field, at the same time you might add the new method and the field in the new class.
4. Expandable class:The expandable class fully manifested is-a to extend the relations The form is:Class (subclass) extends (base class).
5. Multi-modality: In java, the object variable is multi-modality But in java does not support multiple extend.
6. Dynamic combine: the mechanism of invoking object method mechanism.

1) compiler examines object statement type and method name.

2) the compiler examines that method invokes parameter type.

3) static combine: If the method type is priavte static the final ,compiler can accurately know which method should invoke.

4) when the procedure runs and uses dynamic combine to invoke a method, the method edition which then hypothesized machine must invoke x the object actual type which aims at to match.

5) dynamic combine: is a very important characteristic, it can cause the procedure to change again may expand but does not need to translate has saved the code.

1. Final class:In order to prevent other people derive the new class from yours class, this class is cannot expanded.
2. The dynamic invocation spend longer time than the static invocation expenditure. 16.Abstract class:Stipulated or many abstract methods class of itself must define is abstract.

Example: Public abstract string getDescripition

1. In Java each class is be extended by object class.
2. equal and toString method in object class .

Equal uses in testing an object is whether equal with another object.

ToString returns to represent this object the string of character, each class can nearly over loading this method, in order to returns to the current condition the correct expression.

(The toString method is a very important method)

1. General programming:Any class of type all values all may replace with a object class of variable.
2. The array tabulates: The ArrayList dynamic array tabulates, is a class of storehouse, defines in java.In uitl package, but automatic control array size.
3. in class and class of object ,getclass method returns to the class type an example, when the procedure start contains can increase in the main method class, hypothesized confidential increase all classes which he needs, each increase class all must increase the class which it needs. 22.The class: class might dynamic operate the java code for the compilation the procedure

to provide the formidable function reflection, this function was JavaBeans is specially useful, the use reflected Java to be able to support the VB programmer to be familiar with the use the tool.

1) when run analysis class ability.

2) when run searches observes a class of object.

3) realizes the general array operation code.

4) provides the method object.

But this mechanism mainly aims at the tool but not the application and the procedure.

In the reflection mechanism most important part is that permits class that you inspect structure. With to API includes:

Method pointer: Java does not have the method pointer, makes a method address another method, may invoke it in behind, but the interface is the better solution.

1. interface: should showing class could do what but not to assign how to do, a class may realize one or many interfaces.
2. The interface is not a class, but is to conforms to a interface request class of set of standard.

If realizes a interface to need 2 steps:

1) the statement class needs to realize assigns the interface.

2) provides in the interface all methods definition.

Stated a class realizes a interface to need to use the implements key words

Class actionB implements Comparable its actionb needs to provide the CompareTo method, the interface is not the class, cannot use a new example interface.

25.A class only then a super class, but a class can realize many interfaces. In a Java important interface: Cloneable

1. The interface and call-back :to programs a commonly used pattern is call-back,in the pattern, you may refer when this class of pattern settled specific time occurs returns to adjusts on the object the method.

Example:ActionListener interface monitor.

Similar API includes:

1. Object clone:The clone method is a object protection method, this meant your code cannot simple invoke it.
2. Inner class an inner class definition is the definition in another class.

The reason is:

* 1. an inner class object can visit founds its object realization, including private data.

2) about other classes in the same package in that, inner class can be hided.

3) the anonymous inner class may the very convenient definition accent.

In 4) uses the category to be possible the extremely convenient compilation event driver.

1. Agent class (proxy):

1) appointing all codes that interfaces request

2) all methods (toString equals) that object class define

1. Data type:Java is kind of emphasizing type language, each variable all must be declared its types at first, in java altogether has 8 basic types . four kinds are the long, two kinds are the

float, one is the char, being used in the Unicode code char, Boolean.

1. java is simple

Java and C + + are very similar, but much simpler. All the high-level programming language features, is not absolutely necessary have been deleted. For example, Java does not overload operator, the title of the document, pre-processing, computing pointer, the structure of the joint, multi-dimensional array, templates and implicit type conversion. If you know a little C, C + + or Pascal, you will soon master Java. Here is a simple procedure JavaHelloWorld:

publicclassHelloInternet ( publicstaticvoidmain (Stringargv []) ( System. out. println ( "HelloIn-ternet!")

)

)

1. Java is object-oriented

Java is an object-oriented programming language. In addition to the simple types, such as digital and Boolean operators in addition, Java is an object of most. As with any object-oriented languages, Java code also organized by category. Each category provides a definition of the object behavior. Another type of succession can be a kind of behavior. In the category of the root level, often the target category. Java support for the single type of inherited hierarchy. This means that each category can only inherit one other category. Some of the language to allow multiple inheritance, but it may cause confusion and unnecessarily complicated language. For example, imagine that an object would inherit two completely different category. Java also supports the kind of summary of the interface. This allows programmers to define the interface methods, and do not have to rush immediately to determine the methods to achieve. A type of interface can be a number of purposes in order to truly multi-inheritance of a number of advantages. The implementation of an object can be any number of interfaces. IDL interface and Java interface very similar. Very easy to set up IDLJ compiler. In other words, Java can be used to create a CORBA object system distributed object system. In the view of many computer systems use IDL interface and CORBA objects, such compatibility is important.

1. Java is a type of static

In a Java program, it is essential to the definition used by the target (number of characters, such as an array) type. This will help programmers quickly found because the procedure when the compiler can detect the type of error. However, Java System objects are also dynamic types. A requirement for the type of dynamic is often possible, so programmers can write the procedures for different types of objects to do different things.

1. Is a Java-based compiler

When running Java programs, which were first compiled into byte code. Byte code is very similar to the machine instructions, so Java program is very efficient. However, the byte code does not specifically for a particular machine, so no need to recompile Java program can be in many different computer implementation. Java source code files were compiled into a category, which is equivalent to process byte code performance. In a Java class file, and an example for all of the variables are in the light of, and for the first time in the implementation of the code be resolved. This makes the code more common and more easily subject to revision, but still high.

1. Java is architecture neutral

Java language is the same for each computer. For example, simple types are the same: 32-bit integer always, always 64-bit long integers. It is strange, such as C and C + + programming language, and so fashionable it is not the case. As a result of these languages so the definition of freedom, each of the compiler and development environment will be different, so that this process nuisance become a transplant. Java programs can easily gain access to transplants, and there is no need to re-compile.

1. Java is a sound

Java program can not be caused by the collapse of the computer. Java careful testing of the system memory of each visit, make sure it is legitimate and will not cause any problems. However, even if the Java program may also be wrong. If there is some kind of unexpec ted things, the process will not collapse, and to abandon the exception. Procedures for such an exception would be found to address them. Traditional computer programs can access the full memory. May (unconsciously) to amend any of the value of memory, which will cause problems. Java program can only access memory to allow them access to those parts of the Java program can not modify it does not seek to change the value.

1. Java is a compact

As the Java is designed to run on a small computer, as a programming language for the system is relatively small. It effectively in more than 4MB of RAM to run on PC machine. Java translator occupied by only a few hundred KB. This translator for the Java platform independence and portability is reliable. Due to Java is very small, it is a very small computer memory, such as the Java-based PC, as well as television sets, ovens, telephone and home computer, and so on, it is ideal.

## 附录B 外文翻译—译文部分

JAVA 介绍

作者：Martin Ngobye.

出处：Computing Static Slice for Java Programs

Java 是被设计用来解决在上下文分布式的异构网络中应用程序开发的问题。在这些复杂问题中，首要的是安全发布应用程序、消耗最小的系统资源、可以在任何硬件和软件平台上运行以及被动态地扩展。

Java 最初是作为研发方案的一部分而开发的，这个研究方案是为了给广阔多样的网络设备和嵌入式系统开发高级的软件。目标是开发一个小巧的、可靠性高的、可移植的、分布式的、实时的操作系统平台。当这个方案开始的时候，C++是精选的语言，但是过了一段时间，用 C++语言遇到的困难也随之增长，解决这些问题最好的办法是创造一个全新的 语言平台。设计和体系结构的决定汲取了多种语言，例如Eiffel、 SmallTalk,、Objective C 以及 Cedar/Mesa。结果证明这个语言平台是理想的，对于开发一个可靠性高的、可移植的、分布式的、基于终端用户网络的应用程序在多种环境中广泛搜索从嵌入网络的设备到万维网和桌上型电脑。

Java 的需求设计通过自然状态的，在软件必须被配置的处理环境中被推动的。

因特网和万维网的巨大发展导致我们完全地以新的情形着眼于开发和发行软件。对于全球电子化的商业和销售发行的存在，Java 必须使安全的，高性能的和非常健壮的应用程序在多平台异构的，分布式的网络中开发。

在异构的网络中，多平台上运行会使传统配置的二进制发送、释放、更新、补丁等等无效。为了在这些问题中幸存下来，Java 必须是体系结构中立的、可移植的以及可以被动态的修改。

为满足以上的需求而开发的Java 系统是简单的，所以它能够被绝大多数的程序员进行编程；熟悉的，所以当前的程序员学习Java 语言是简单的；面向对象的，为了利用现代的软件开发方法学和适合分布式的客户机/服务器应用程序；多线程的，为了高性能在应用程序中需要执行多个并发的活动,例如多媒体；解释的，为了最大极限的可移植和动态的容量。

同时，以上的需求完全是由收集的时髦词语组成的，所以在开始以前让我们来查看它们中的一些和它们的各自的优点。

什么是完全新的，样式风格在Java 语言和它的运行系统结合它们来产生一个灵通的且强大的编程系统。

在软件中，用Java 语言开发的你的应用程序可以在多种机器总体结构、操作系统和图形用户接口之间移植。运用 Java 语言，诸如软件开发者的工作就会变得简单的多——你会把你的全部精力都集中在最终的目标：准时运送革新的产品、基于可靠的构造的 Java 语言。较好的开发软件的方法在这里，现在，Java 语言平台会带给你这些。

许多动态的语言像 Lisp、TCL 和 SmallTalk 通常被应用于原型构造技术它们成功原因

中的一个在于它们非常健壮——你没有必要去担心关于解除或破坏存储器。

同样的，程序员可以相对大胆的处理存储器，当在 Java 语言里面编程的时候。垃圾回收系统使程序员的工作大为容易，卸下了程序员肩上存储管理之负担，存储分配错误不再发生。

一般认为像Lisp, TCL, 和 SmallTalk 这样一些语言十分适合于原型法的另一个理由， 是因为它们不要求你受早期决定的约束——这些语言的语义是很丰富的。

Java 语言有着严密的对应的属性：它促使你做出清楚的选择。伴随着这些选择带来了许多的协助——你可以写一个方法调用，而且如果你在编写时有一些错误的时候，在编译的时候将会被告知。你没有必要去担心关于方法调用的错误。

Java 初学者需掌握的 30 个基本概念。基本概念：

1.OOP 中唯一关系是什么是对象的接口，就像计算机的销售商不管电源内部结构是怎样的，它只关心能否给你提供电，也就是只要知道能或不能而不是知道怎样和为什么会有电。所有的程序是由一定的属性和行为对象组成的，不同的对象的访问通过函数调用来完成，对象间所有的交流都是通过方法调用，通过封装对象数据，很大限度上提高复用率。2.OOP 中最重要的思想是类，类是模板是蓝图，从类中构造一个对象，即创建了这个

类的一个实例(instance)。

1. 封装：就是把数据和行为结合起在一个包中，并对对象使用者隐藏数据的实现过程， 一个对象中的数据叫他的实例字段(instance field)。
2. 通过扩展一个类来获得一个新类叫继承(inheritance)，而所有的类都是由 Object 根超类扩展而得，根超类下文会做介绍。
3. 对象的 3 个主要特性：

Behavior——说明这个对象能做什么。Tate——当对象施加方法时对象的反映。Dentity——与其他相似行为对象的区分标志。

每个对象有唯一的indentity 而这 3 者之间相互影响。

1. 类之间的关系： use-a ：依赖关系has-a ：聚合关系

is-a ：继承关系——例:A 类继承了 B 类，此时 A 类不仅有了 B 类的方法，还有其自己的方法(个性存在于共性中)。

1. 构造对象使用构造器:构造器的提出，构造器是一种特殊的方法，构造对象并对其初始化。

例:Data 类的构造器叫Data。

New Data()——构造一个新对象，且初始化当前时间。

Data happyday=new Data()——把一个对象赋值给一个变量happyday，从而使该对象能够多次使用，此处要声明的使变量与对象变量二者是不同的。

new 返回的值是一个引用。

构造器特点:构造器可以有 0 个，一个或多个参数。构造器和类有相同的名字。

一个类可以有多个构造器,构造器没有返回值,构造器总是和new 运算符一起使用。

1. 重载:当多个方法具有相同的名字而含有不同的参数时，便发生重载.编译器必须挑选出调用哪个方法。
2. 包(package)Java 允许把一个或多个类收集在一起成为一组，称作包，以便于组织任务， 标准Java 库分为许多包Java.lang java.util java，net 等，包是分层次的所有的 java 包都在 java和 javax 包层次内。
3. 继承思想：允许在已经存在的类的基础上构建新的类，当你继承一个已经存在的类时，那么你就复用了这个类的方法和字段，同时你可以在新类中添加新的方法和字段。
4. 扩展类：扩展类充分体现了 is-a 的继承关系，形式为：class (子类) extends (基类)。12.多态：在 java 中，对象变量是多态的，而java 中不支持多重继承。
5. 动态绑定：调用对象方法的机制。
6. 编译器检查对象声明的类型和方法名。
7. 编译器检查方法调用的参数类型。
8. 静态绑定：若方法类型为priavte static final 编译器会准确知道该调用哪个方法。
9. 当程序运行并且使用动态绑定来调用一个方法时，那么虚拟机必须调用 x 所指向的对象的实际类型相匹配的方法版本。
10. 动态绑定：是很重要的特性，它能使程序变得可扩展而不需要重编译已存代码。
11. final 类：为防止他人从你的类上派生新类，此类是不可扩展的。
12. 动态调用比静态调用花费的时间要长。
13. 抽象类：规定一个或多个抽象方法的类本身必须定义为abstract。例:：public abstract string getDescripition
14. Java 中的每一个类都是从Object 类扩展而来的。
15. object 类中的 equal 和 toString 方法。

equal 用于测试一个对象是否同另一个对象相等。

toString 返回一个代表该对象的字符串，几乎每一个类都会重载该方法，以便返回当前状态的正确表示(toString 方法是一个很重要的方法)。

1. 通用编程：任何类类型的所有值都可以同object 类性的变量来代替。
2. 数组列表：ArrayList 动态数组列表，是一个类库，定义在 java.uitl 包中，可自动调节数组的大小。
3. class 类 object 类中的 getclass 方法返回 class 类型的一个实例，程序启动时包含在

main 方法的类会被加载，虚拟机要加载他需要的所有类，每一个加载的类都要加载它需要的类。

1. class 类为编写可动态操纵 java 代码的程序提供了强大的功能反射，这项功能为

JavaBeans 特别有用，使用反射Java 能支持 VB 程序员习惯使用的工具。

1. 在运行时分析类的能力。
2. 在运行时探察类的对象。
3. 实现通用数组操纵代码。
4. 提供方法对象。

而此机制主要针对是工具者而不是应用及程序。

反射机制中的最重要的部分是允许你检查类的结构.用到的 API 有：

方法指针:java 没有方法指针，把一个方法的地址传给另一个方法，可以在后面调用它， 而接口是更好的解决方案。

1. 接口(Interface)说明类该做什么而不指定如何去做，一个类可以实现一个或多个

interface。

1. 接口不是一个类，而是对符合接口要求的类的一套规范。若实现一个接口需要 2 个步骤： 1）声明类需要实现的指定接口。

2）提供接口中的所有方法的定义。

声明一个类实现一个接口需要使用implements 关键字。

class actionB implements Comparable 其actionb 需要提供CompareTo 方法，接口不是类， 不能用 new 实例化一个接口。

1. 一个类只有一个超类，但一个类能实现多个接口。Java 中的一个重要接口：Cloneable。
2. 接口和回调.编程一个常用的模式是回调模式，在这种模式中你可以指定当一个特定时间发生时回调对象上的方法。

例:ActionListener 接口监听。类似的 API 有：

1. 对象 clone:clone 方法是 object 一个保护方法，这意味着你的代码不能简单的调用它。
2. 内部类：一个内部类的定义是定义在另一个内部的类。原因是：
3. 一个内部类的对象能够访问创建它的对象的实现，包括私有数据。
4. 对于同一个包中的其他类来说，内部类能够隐藏起来。
5. 匿名内部类可以很方便的定义回调。
6. 使用内部类可以非常方便的编写事件驱动程序。29.代理类(proxy)：
7. 指定接口要求所有代码。
8. object 类定义的所有的方法(toString equals)。

30.数据类型：Java 是强调类型的语言，每个变量都必须先申明它的类型，java 中总共有 8 个基本类型：4 种是整型，2 种是浮点型，一种是字符型，被用于 Unicode 编码中的字符，布尔型。

1. Java 是简单的

Java 与 C++极为相似，但却简单得多。高级编程语言的所有特性中，不是绝对需要的都已删去了。例如，Java 没有算符过载、标题文件、预处理、指针运算、结构、联合、多维数组、模板及隐式类型变换。如果你知道一点C、C++或 Pascal，你很快就会驾驭 Java。这里是一个简单的JavaHelloWorld 程序：

publicclassHelloInternet{

publicstaticvoidmain(Stringargv[]){

System。out。println(“HelloIn-ternet!”)

}

}

1. Java 是面向对象的

Java 是一种面向对象的编程语言。除了简单的类型，如数字和布尔算子之外，Java 中的大部分都是对象。正如任何面向对象的语言一样，Java 代码也按类组织。每个类定义一组规定对象行为的方法。一个类可以继承另一个类的行为。在类的根层次上，通常是类对象。Java 支持单继承类层次结构。这就是说，每个类一次只能继承一个别的类。有些语言允许多继承性，但这可能造成混乱，使语言不必要地复杂化。例如，难以想像，一个对象会继承两个完全不同的类的行为。Java 还支持摘要类的接口。这允许编程人员先定义接口的方法，而不必急于马上确定方法的实现。一个类可以施行多个接口，从而具有真正多继承性的许多优点。一个对象也可实施任意数量的接口。Java 接口与 IDL 接口极其相似。很容易建立IDLJ 编译器。这就是说，Java 可用于 CORBA 对象系统来建立分布式对象系统。鉴于在许多计算机系统中都采用IDL 接口和 CORBA 对象系统，这种兼容性是重要的。

1. Java 是静态类型的

在一个 Java 程序中，必须定义所用对象(数字、字符、数组等)的类型。这有助于编程人员很快发现问题，因为当程序编译时可以检测类型错误。不过，Java 系统中的对象也有动态类型。要求一个对象为动态类型往往是可能的，所以编程人员写的程序可以对不同类型的对象做不同的事。

1. Java 是编译型的

当运行 Java 程序时，它首先被编译成字节代码。字节代码非常类似于机器指令，所以Java 程序非常高效。然而，字节代码并不专对一种特定的机器，所以 Java 程序无需重新编译便可在众多不同的计算机上执行。Java 源程序被编译成类文件，它相当于程序的字节代码表现。在一个 Java 类文件中，所有对方法及实例变量的参照均按名进行，并在第一次执

行代码时加以分辨。这使得代码更通用，更不易受修改的影响，而仍具高效。

1. Java 是体系结构中立的

Java 语言对每种计算机都一样。比如，简单的类型都是不变的：整数总是 32 位，长整数总是 64 位。令人奇怪的是，诸如 C 及 C++等时髦的编程语言却不是这样。由于这些语言定义如此自由，每种编译器及开发环境便各有不同了，这使程序的移植成为讨厌的问题。Java 程序的移植却很容易，而且不需要进行重新编译。

1. Java 是健全的

Java 程序不可能造成计算机崩溃。Java 系统仔细检测对内存的每次访问，确认它是合法的，而且不致引起任何问题。不过，即使Java 程序也可能有错误。如果出现某种出乎意料之事，程序不会崩溃，而把该例外抛弃。程序会发现这类例外，并加以处理。传统的程序可以访问计算机的全部内存。程序可能(无意识地)修改内存中的任何值，这就会造成问题。Java 程序只能访问内存中允许它们访问的那些部分，所以 Java 程序不可能修改不拟修改的值。

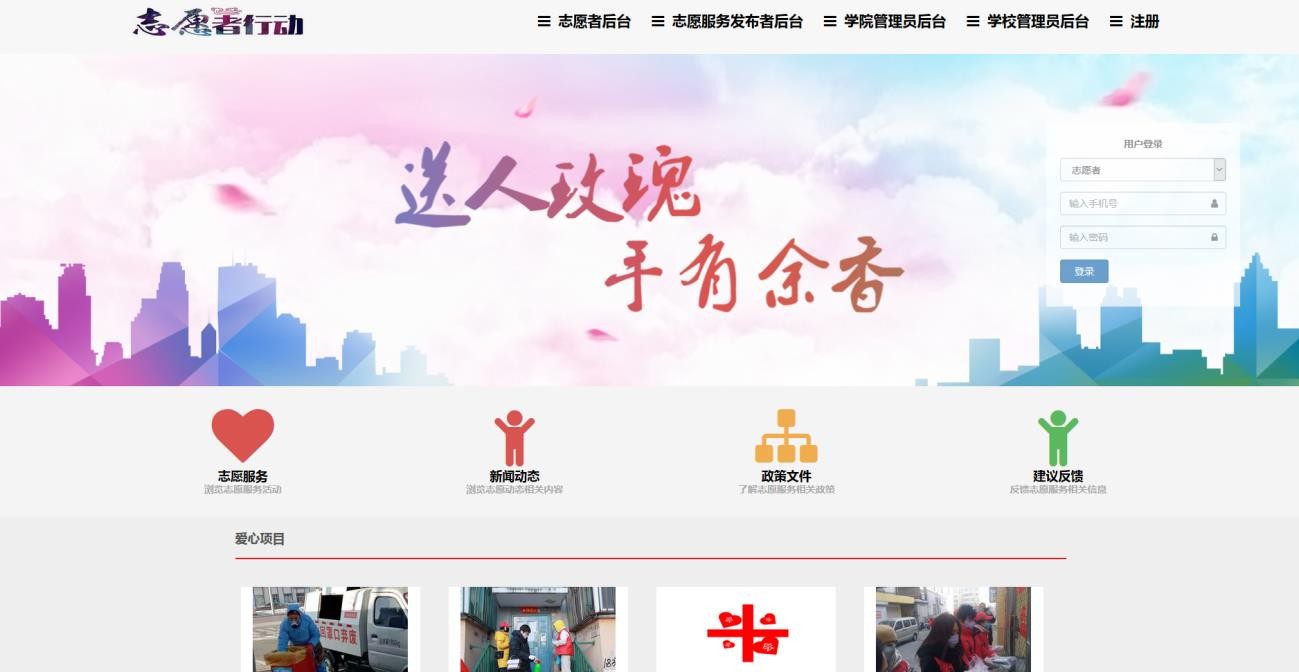
1. Java 是小巧的

由于 Java 的设计是要在小的计算机上运行，作为一种编程语言来说其系统是相对较小的。它能有效地在 4MB 以上 RAM 的 PC 机上运行。Java 翻译器只占用几百KB。这种翻译器对 Java 的平台无关性和可移植性是可靠的。由于Java 很小，它对内存很小的计算机， 如基于 Java 的 PC 机，以及电视机、烤箱、电话机及家用计算机等，是很理想的。

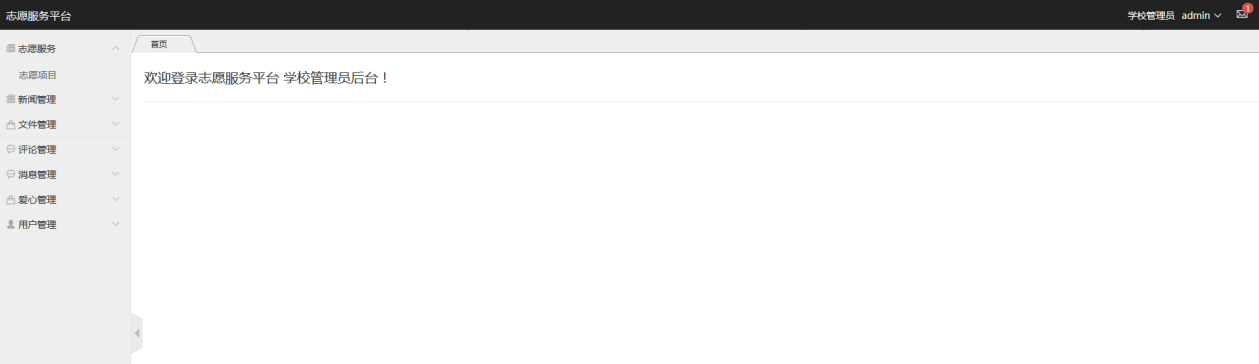
## 附录C 软件使用说明书

志愿服务平台门户网站主页界面最上方导航栏设有志愿者后台、志愿服务发布者后台、学员管理员后台、学校管理员后台、注册跳转按钮，点击可进入到相应的界面，非法用户不能对后台进行任何操作。

用户登录处首先选择用户类型，共四类：志愿者、志愿服务发布者、学院管理员、学校管理员，再输入手机号、密码，点击登录按钮进入对应的用户后台管理页面。

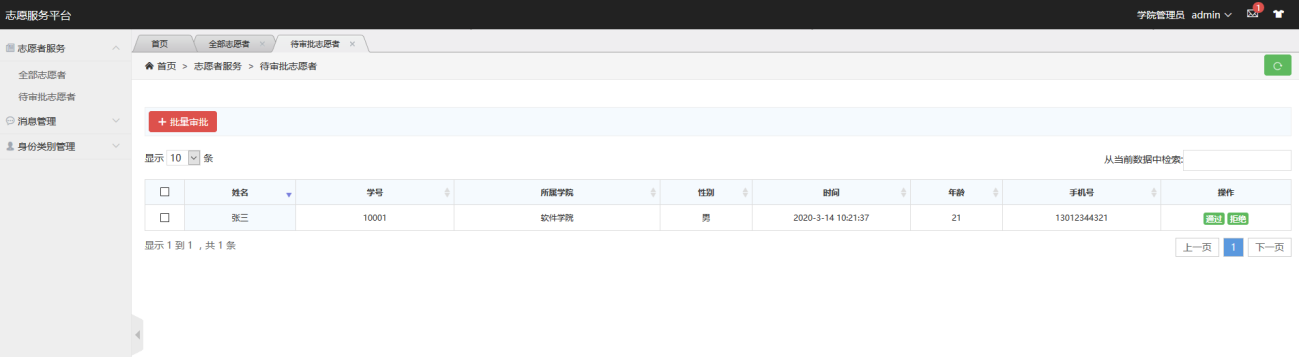
游客、用户可点击志愿服务进入新闻列表界面，在志愿服务列表界面中可点击志愿服务标题处进入志愿服务详情界面。游客、用户可点击新闻动态进入新闻列表界面，在新闻列表界面中可点击新闻标题处进入新闻详情界面。游客、用户可点击政策文件进入政策文件列表界面，在政策文件列表界面中可点击政策文件标题下载该政策文件。游客、用户可点击建议反馈进入建议反馈详情界面，在建议反馈详情界面中可填写个人相关信息和反馈内容提交建议反馈。

学校管理员后台管理主页面中左侧列表栏分别是：志愿服务、新闻管理、文件管理、评论管理、消息管理、爱心管理、用户管理，可进入相应的子页面进行对应的各项操作。

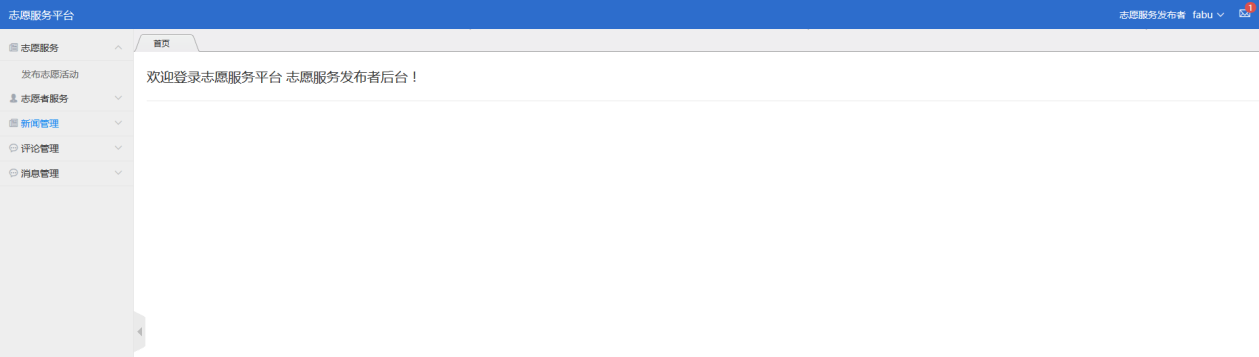


学院管理员后台管理主页面中左侧列表栏分别是：志愿者服务管理、身份类别管理、

消息管理、 爱心管理、 用户管理， 可进入相应的子页面进行对应的各项操作。



志愿服务发布者后台管理页面，首页界面左侧列表分别是：志愿服务、志愿者服务、新闻管理、评论管理、消息管理，可进入相应管理的子页面进行各项操作。



志愿者后台管理主页界面设有个人信息、已报名活动、积分捐献明细，个人信息包括： 个人账号状态、姓名、学号、所属学院、手机、性别、年龄、身份类别、志愿时长、爱心积分，已报名活动可查看志愿者报名参加活动的详细信息，积分捐献明细可查看志愿者捐献积分的详细信息。

## 附录D 主要源代码

