**目 录**

[**1 绪 论 1**](#_Toc41480613)

[1.1 研究的背景及意义 1](#_Toc41480614)

[1.1.1 选题的背景 1](#_Toc41480615)

[1.1.2 国内外研究现状 1](#_Toc41480616)

[1.1.3 研究的意义 1](#_Toc41480617)

[1.2 系统目标 2](#_Toc41480618)

[**2 需求分析 3**](#_Toc41480619)

[2.1 业务需求 3](#_Toc41480620)

[2.1.1主要业务流程 3](#_Toc41480621)

[2.2 功能需求 6](#_Toc41480622)

[2.2.1 角色分析 6](#_Toc41480623)

[2.2.2 业务功能 6](#_Toc41480624)

[2.3 非功能需求 9](#_Toc41480625)

[2.3.1环境需求 9](#_Toc41480626)

[2.3.2 性能需求 9](#_Toc41480627)

[2.3.3 安全需求 9](#_Toc41480628)

[**3 总体设计 10**](#_Toc41480629)

[3.1系统设计的原则 10](#_Toc41480630)

[3.2 系统体系结构设计 10](#_Toc41480631)

[3.3 系统功能结构设计 10](#_Toc41480632)

[**4 数据库设计 12**](#_Toc41480633)

[4.1 概念结构设计 12](#_Toc41480634)

[4.1.1 设计思路 12](#_Toc41480635)

[4.1.2 E-R图 12](#_Toc41480636)

[4.2 逻辑结构设计 16](#_Toc41480637)

[4.2.1 设计思路 16](#_Toc41480638)

[4.2.2 逻辑模型 16](#_Toc41480639)

[4.3 物理结构设计 17](#_Toc41480640)

[4.3.1 存取方式 17](#_Toc41480641)

[4.3.2 存储结构 17](#_Toc41480642)

[**5 界面设计 18**](#_Toc41480643)

[5.1 界面关系图或工作流图 18](#_Toc41480644)

[5.2 界面设计成果 18](#_Toc41480645)

[5.2.1 主界面 18](#_Toc41480646)

[5.2.2 子界面 19](#_Toc41480647)

[**6 详细设计 22**](#_Toc41480648)

[6.1 系统主要功能模块介绍 22](#_Toc41480649)

[6.2新闻管理模块 22](#_Toc41480650)

[6.2.1 功能结构设计 22](#_Toc41480651)

[6.2.2 类图设计 23](#_Toc41480652)

[6.2.3 顺序图设计 24](#_Toc41480653)

[6.2.4 核心处理流程设计 24](#_Toc41480654)

[6.3补助申请模块 25](#_Toc41480655)

[6.3.1功能结构设计 25](#_Toc41480656)

[6.3.2类图设计 25](#_Toc41480657)

[6.3.3顺序图设计 26](#_Toc41480658)

[6.3.4核心处理流程设计 26](#_Toc41480659)

[6.4邮件信管理模块 27](#_Toc41480660)

[6.4.1功能结构设计 27](#_Toc41480661)

[6.4.2类图设计 27](#_Toc41480662)

[6.4.3顺序图设计 28](#_Toc41480663)

[6.4.4核心处理流程设计 28](#_Toc41480664)

[6.5留言板管理模块 29](#_Toc41480665)

[6.5.1功能结构设计 29](#_Toc41480666)

[6.5.2类图设计 29](#_Toc41480667)

[6.5.3顺序图设计 30](#_Toc41480668)

[6.5.4核心处理流程设计 31](#_Toc41480669)

[6.6人员信息管理模块 31](#_Toc41480670)

[6.6.1功能结构设计 31](#_Toc41480671)

[6.6.2类图设计 31](#_Toc41480672)

[6.6.3顺序图设计 32](#_Toc41480673)

[6.6.4核心处理流程设计 32](#_Toc41480674)

[**7 编码 34**](#_Toc41480675)

[7.1 代码实现与核心算法 34](#_Toc41480676)

[7.2 代码优化分析 37](#_Toc41480677)

[**8 测试 38**](#_Toc41480678)

[8.1 测试方案设计 38](#_Toc41480679)

[8.1.1 测试策略 38](#_Toc41480680)

[8.1.2 测试进度安排 38](#_Toc41480681)

[8.1.3 测试资源 38](#_Toc41480682)

[8.1.4 关键测试点 38](#_Toc41480683)

[8.2 测试用例构建 38](#_Toc41480684)

[8.2.1 测试用例编写约定 38](#_Toc41480685)

[8.2.2 测试用例设计 39](#_Toc41480686)

[8.2.3 关键测试用例 39](#_Toc41480687)

[8.2.4 测试用例维护 40](#_Toc41480688)

[**9 总结与展望 41**](#_Toc41480689)

[9.1 设计工作总结 41](#_Toc41480690)

[9.2 未来工作展望 41](#_Toc41480691)

[**谢 辞 42**](#_Toc41480692)

[**参考文献 43**](#_Toc41480693)

[附录A 外文翻译—原文部分 45](#_Toc41480694)

[附录B 外文翻译—译文部分 50](#_Toc41480695)

[附录C 软件使用说明书 53](#_Toc41480696)

[附录D 主要源代码 56](#_Toc41480697)

1 绪 论

## 1.1 研究的背景及意义

### 1.1.1选题的背景

国内现在有许多的离休老干部，而这些老干部最好需要形成一个群体，方便让其他人来了解离休老干部的生活是怎么样的，以及离休老干部关注的地方，那么如何让大家都能看到这一现象呢，这时我们可以做一个关于老干部管理的门户网站，用此网站来了解老干部的退休生活，以及老干部之间可以通过此网站实现线上发件来发送各方所需要的东西，同时通过SSM搭建框架来开发此平台。开发此平台有很多好处，从社会性角度来说，离休老干部是相当于在自己原岗位退休，开始过起了退休的生活，而此网站的建设成功一定能为他们的退休生活增添一份色彩，并且可以让那些充满好奇的人可以去了解离休老干部的生活，针对这几点，我选择了此题目来设计出一个适合的老干部门户网站。

### 1.1.2 国内外研究现状

老年人的社会参与是实现生产性老龄化的重要途径。在“互联网+”的背景下要确保老干部活动经费的投入，让老干部享受悠闲又有意义的退休生活，在我国，科学发展观的核心是以人为本，就是“人人为我，我为人人”一切围绕老干部而建设此网站，一切为了老干部，一切服务老干部。对老干部的信息进行管理，了解老干部的住址，婚姻情况，对老干部的用户进行科学规范细致的管理和满意并且及时的服务，保证老干部都能够健健康康，做到晚年能够不寂寞，有人陪伴。

老干部离休之后也成为了一位普通的老百姓，但是又与其他老板姓不同，他们以往贡献大、功劳多、职务高，为了方便管理他们，也要像他们对待以前的职务一样来对待他们，做到尽量满足他们的需求。而国外也对他们的退休高官给予了一定的退休补偿，同时他们也做了类似老干部门户网站之类的网站来记录管理离休老干部的生活，这都是我值得借鉴的，可以看出国内外都挺重视离休老干部的管理。

### 1.1.3 研究的意义

随着互联网技术的高速度的发展和信息化技术的不断更新提高，我们对于离休干部信息的管理的需求也越来越高。需求虽然很高，但是都是很基本的东西，此项研究设计只需让普通老百姓来认真了解老干部的退休生活以及老干部需要了解的信息，通过老干部之间的相互联系，让老干部们的退休生活不再孤单，同时也给我们反应出一个良好的信号，虽然现在我国处于人口老龄化的时代，但是我们一样做到老有所养，有始善终，不光对老干部们好，同时也对社会也是个积极的信号，让社会的老干部联系在一起，变得热闹起来，不会让他们觉得老年生活是冷清的。  
 所以在这种背景下，开发一款关于离休干部的网站很重要。也为政府的服务提供了一个重要的帮助，帮助减轻了政府工作人员的工作量，也更好地管理了离休干部的基本相关的信息。

## 1.2 系统目标

（1）查阅资料和相关的文档，确定所需要的功能如何实现。

（2）开发环境的搭建:包括JAVA环境变量设置，JDK,TOMCAT的下载安装及配置。

（3）进行系统的可行性分析，完成系统的需求分析。  
（4）对系统进行概要设计，画出系统的总结构图，理清各个模块之间的关系，并根据这些关系设计好数据库。  
（5）对系统进行详细设计，确定各模块，包括软件系统的登录注册界面，还有后台系统的功能模块页面，主要包括：新闻管理，邮件信管理，困难补助管理，人员信息管理。  
（6）进行软件编码，实现其各项基本功能。  
（7）系统完成后进行真实的检测，确保数据的准确性。

做到能够对离退休人员基本信息的管理，实现对其的增删改查操作;管理员可以有权限对用户的权限进行审核和撤销;实现数据库信息的导入，导出，备份和恢复;离退休人员的个人信息模块由个人信息修改，邮件信息管理，补助管理组成。

# 2 需求分析

## 2.1 业务需求

（1）能够实现对老干局网站用户的信息管理（对信息的增删改查）。

（2）具有新闻发布功能，点开此网站的人能够浏览到新闻新消息。

（3）用户可以发布补助通知，并经过管理员审核之后发布成功并能在首页显示供浏览者查看。

（4）用户与用户之间可以互相发送邮件，并且邮件管理具有增删改查的特性。

（5）管理员可以授予和撤销用户的权力。

（6）数据库具有恢复和备份的功能。

### 2.1.1主要业务流程

（1）用户登录

用户登录活动图如下图2-1所示。

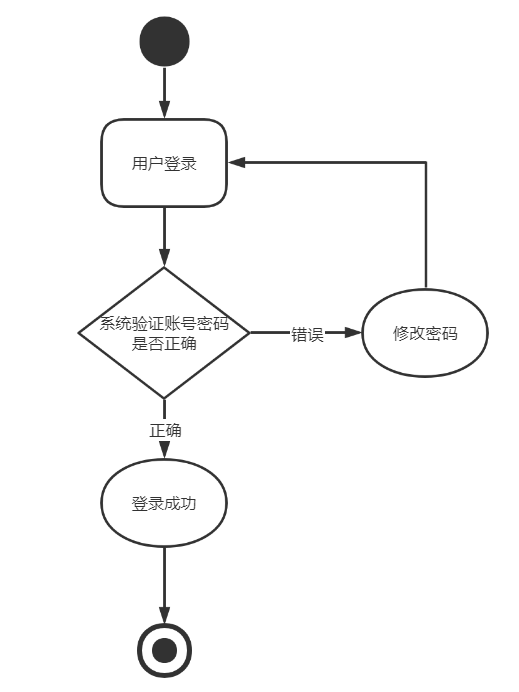


图2-1用户登录活动图

（2）人员信息管理活动图

人员信息管理活动图如下图2-2所示。

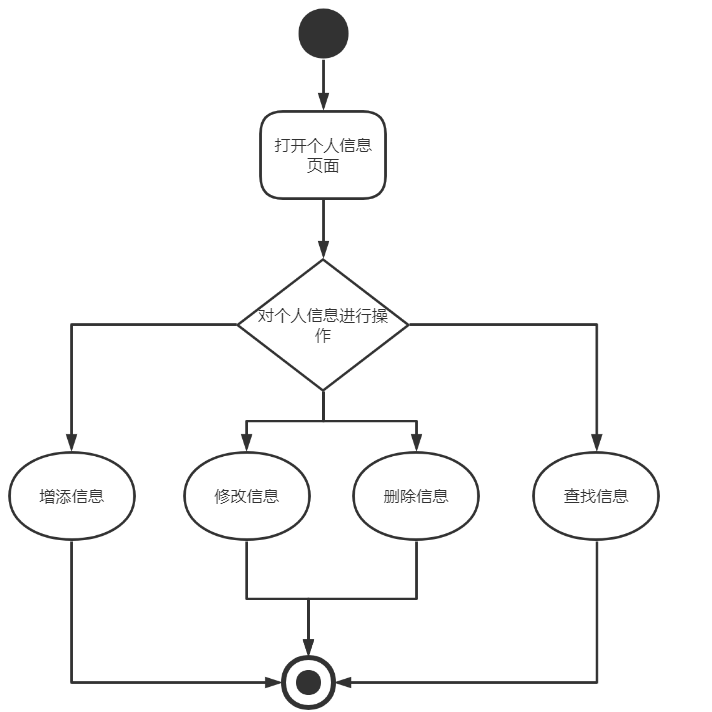


图2-2人员信息管理活动图

（3）邮件信管理活动图

邮件信管理活动图如下图2-3所示。

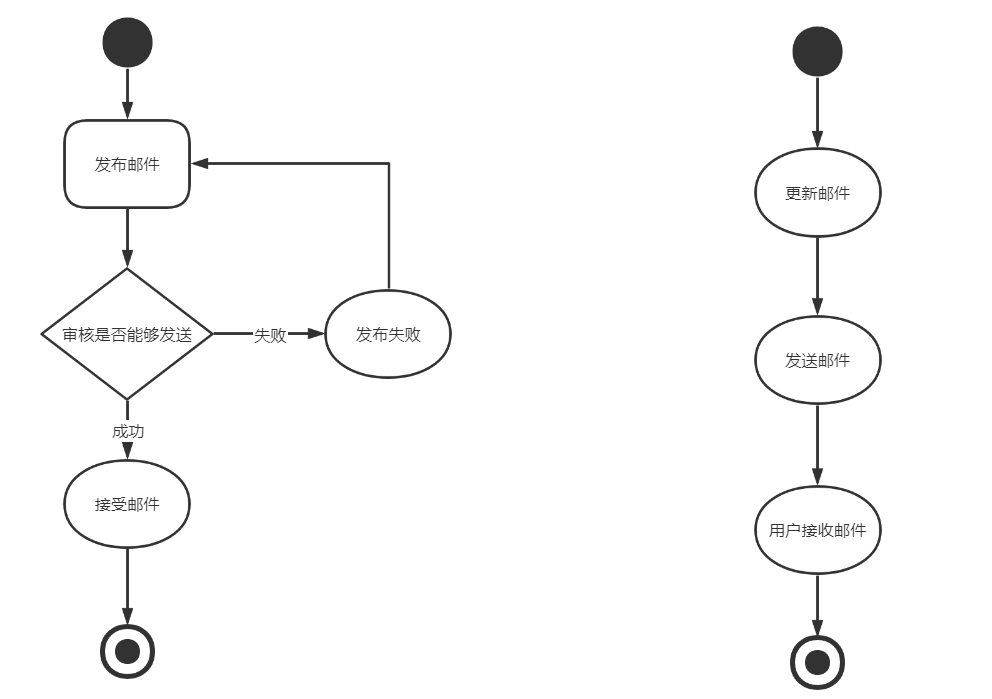


图2-3邮件信管理活动图

（4）新闻通知活动图

新闻通知活动图如下图2-4所示。

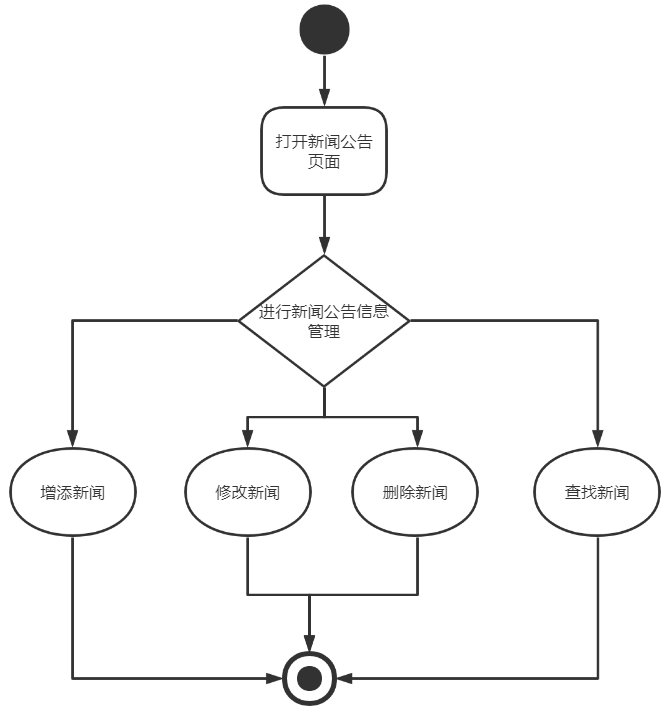


图2-4新闻通知活动图

（5）补助管理活动图

补助管理活动图如下图2-5所示。

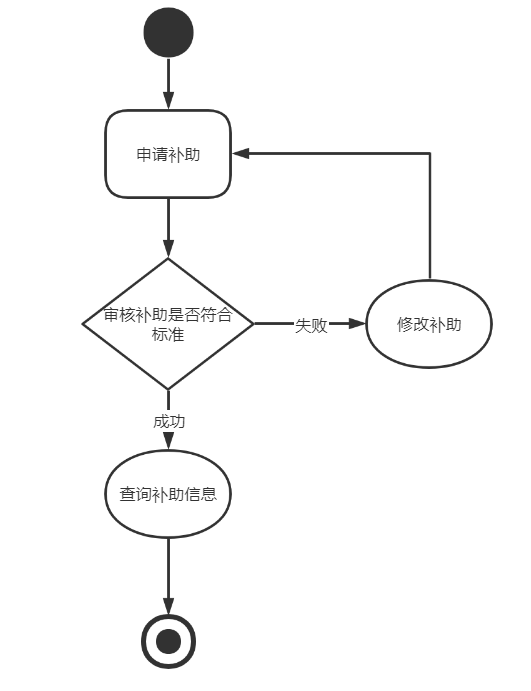


图2-5补助管理活动图

## 2.2 功能需求

### 2.2.1 角色分析

以下从老干局门户网站的实际需求分析，系统涉及到以下角色：

表3-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 一般用户 | 能够发布补助公告给管理员审核，并且能够互相发送邮件 |
| 系统管理员 | 管理和维护整个系统的用户组织结构，负责对用户、角色、用户级别的增、删、改、查等管理。 |

### 2.2.2 业务功能

以下从业务角度出发，给出了系统的总体用例图，包含系统登录，发送邮件，发布补助，接收邮件，审核补助，发布新闻权限管理，查看留言，留言板留言。如下图3-6所示：

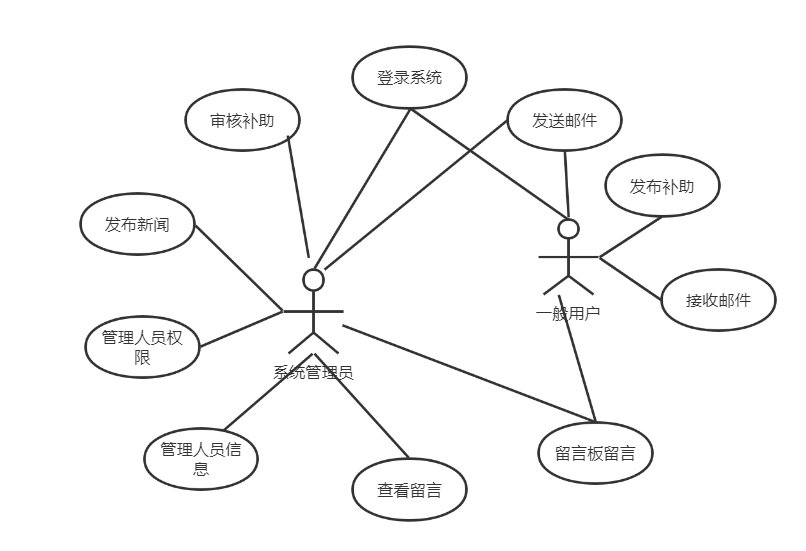


图3-6系统总体用例图

2.2.2.1补助管理

补助管理用例图如下图3-7所示。

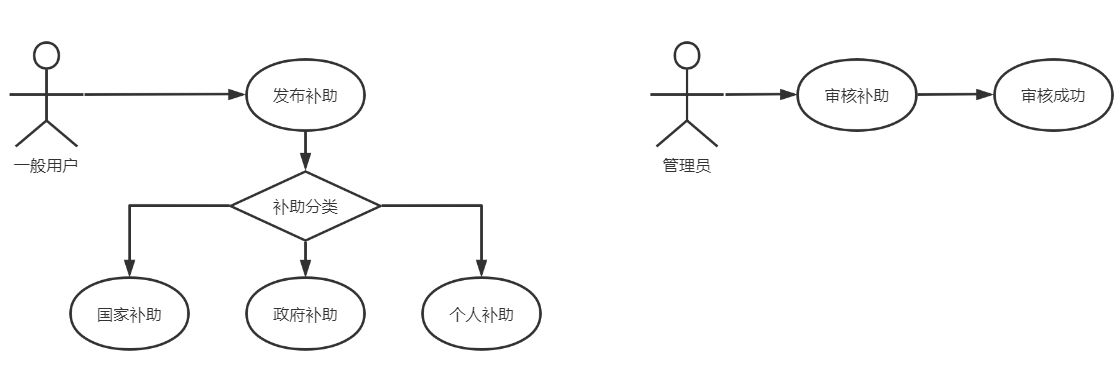


图3-7补助管理用例图

根据补助管理的用例图列出如下表3-1所示的用例描述。

表3-1补助管理用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 发布补助 |
| **标识符\*** | YL01 |
| **用例描述** | 描述了一般用户如何发布补助介绍了补助的分类并提交给后台 |
| **参与者表** | 一般用户 |
| **优先级** | 1 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录系统 |
| **后置条件** | 系统给出操作成功提示 |
| **基本操作流** | 1.一般用户在申请补助页面上写出自己申请的补助并填写补助的金额；  2.选定补助的分类，个人补助，国家补助，政府补助；  3.确定补助分类后并写完了正文点击提交补助即可； |
| **被泛化用例表** | 该用例的特化用例列表 |

根据审核补助用例图列出如下图3-2所示的用例描述。

表3-2审核补助用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 审核补助 |
| **标识符\*** | YL02 |
| **用例描述** | 描述了管理员审核用户补助的过程 |
| **参与者表** | 系统管理员 |
| **优先级** | 2 |
| **状态\*** | 进行中 |

续表3-2

|  |  |
| --- | --- |
| **前置条件** | 用户已登录系统 |
| **后置条件** | 系统管理员能够正常行使管理审核权力 |

2.2.2.2新闻发布管理

新闻发布管理用例图如下图3-8所示。

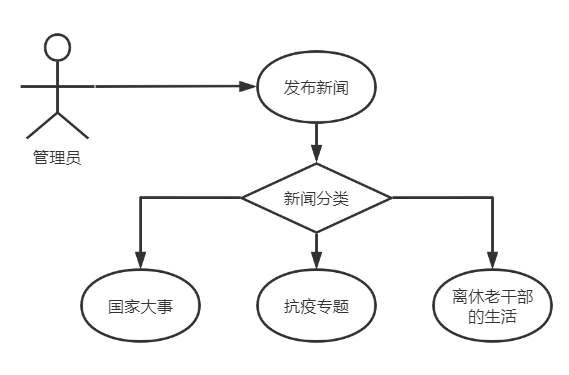


图3-8新闻发布管理用例图

根据新闻发布管理用例图列出如下表3-3所示的用例描述。

表3-3新闻发布管理用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 新闻发布 |
| **标识符\*** | YL03 |
| **用例描述** | 描述了管理员如何在后台发布新闻 |
| **参与者表** | 管理员 |
| **优先级** | 1 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录系统 |
| **后置条件** | 系统给出操作成功提示 |
| **基本操作流** | 1.管理员在后台的新闻发布中添加新闻；  2.选定新闻的分类，有很多分类我这里就例举了3个；  3.确定新闻分类后并写完了正文点击提交新闻即可； |
| **被泛化用例表** | 该用例的特化用例列表 |

## 2.3 非功能需求

### 2.3.1环境需求

表 3-1 服务器端环境

|  |  |
| --- | --- |
| 系统所使用数据库 | My SQL 8.0 |
| 操作系统 | Windows 7 |
| Web服务器 | Cisco web服务器 |
| 数据库 | My SQL 8.0 |

表 3-2 客户端环境

|  |  |
| --- | --- |
| 操作系统 | Windows XP 或其以上版本操作系统 |
| 浏览器 | 火狐浏览器，QQ浏览器 |
| 分辨率 | 最佳效果为1920×1080像素 |

### 2.3.2 性能需求

（1）系统操作流畅，不崩溃。

（2）系统响应速度在2000毫秒以下。

（3）保证2000并发量。

### 2.3.3 安全需求

（1）日志系统，对于每一个接口调用，每一个错误出现都要有详细到参数和时间的日志系统，便于日后维护。

（2）全局统一异常处理，不在内层代码捕捉处理单个异常，在最外层统一处理。

（3）参数校验系统，请求进来到后台的重要参数，都需要校验一遍。

3 总体设计

## 3.1系统设计的原则

在充分考虑到本系统的总体规划目标，以及系统的应用环境前提下，本系统的设计将严格遵循以下的设计原则：

（1）系统性原则：系统是一个有机整体。所以，设计时要满足高度统一的这一原则性，从整体系统来看，使系统有统一的数据计算方法，统一的设计标准，统一的信息条件代码，来规范并提高系统的设计质量。

（2）经济性原则：在系统完整的情况下，做到简便，能够减少不需要的支出来满足这个经济性的原则。

（3）可靠性原则：此原则是做系统的基本，只有做出来的系统是安全可靠的，人们才会去使用，系统也才能够发挥出自己的作用。

## 3.2 系统体系结构设计

本系统采用了SSM框架，技术层面用了MAVEN，MYSQL，HTML，CSS，JSP以及JQUERY的技术首先SSM框架是由Spring+Spring MVC+MyBatis的缩写，是现在比较主流的框架结构，比如我的页面跳转跟页面访问限制主要就由SSM框架做成的，通过最大上传值跟最小上传值的限制来限制用户的访问。

Spring通过控制反转的依赖注入模式将很多系统的功能类有效的黏合在一起，并且在同一时间让他们保持独立。Spring MVC分离了控制器、分派器、模型对象，使他们更加容易进行定制。而MyBatis则是用来帮助我管理模块中增删改查的一个框架。

## 3.3 系统功能结构设计

系统一共分为两个子系统，前台老干局门户网站子系统和后台管理子系统。其中，在线老干局门户网站子系统又分为新闻查看、补助查看、留言板留言、新闻分类、补助分类共5个模块。后台管理子系统又分为用户信息管理、补助信息管理、留言板管理和用户权限管理、新闻消息管理、邮件信管理共6个模块。系统总包图如下图3-3所示。

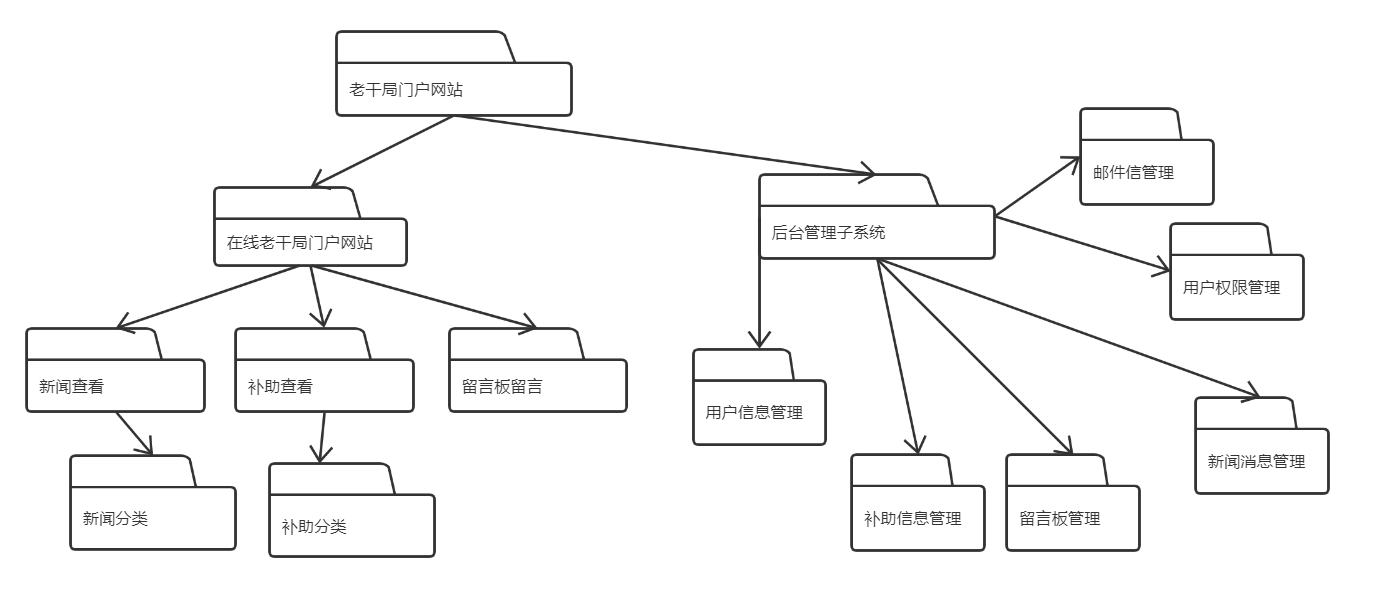


图3-3系统总包图

# 4 数据库设计

## 4.1 概念结构设计

### 4.1.1 设计思路

根据具体功能需求分析可得，其中涉及实体为以下几个：一般用户、管理员、新闻、发件箱、收件箱、新闻日期、补助公告、新闻分类；对应关系如下：

（1）一般用户与管理员对应关系为多对一。

（2）一般用户与补助公告的对应关系为一对多。

（3）一般用户与发件箱的对应关系为一对一。

（4）一般用户与收件箱的对应关系为一对一。

（5）管理员与新闻的对应关系为一对多。

（6）管理员与发件箱的对应关系为一对一。

（7）管理员与收件箱的对应关系为一对一。

（8）新闻与新闻日期的对应关系为多对多。

（9）新闻与新闻分类的对应关系为多对一。

（10）发件箱与收件箱的对应关系为多对多。

（11）新闻日期与新闻分类的对应关系为多对多。

其中具体业务关系为：一般用户发送邮件，发布补助公告；管理员发布新闻，添加新闻分类与新闻日期。

### 4.1.2 E-R图

实体关系的E-R图描述了整个系统的实体关系，如下图4-1所示。

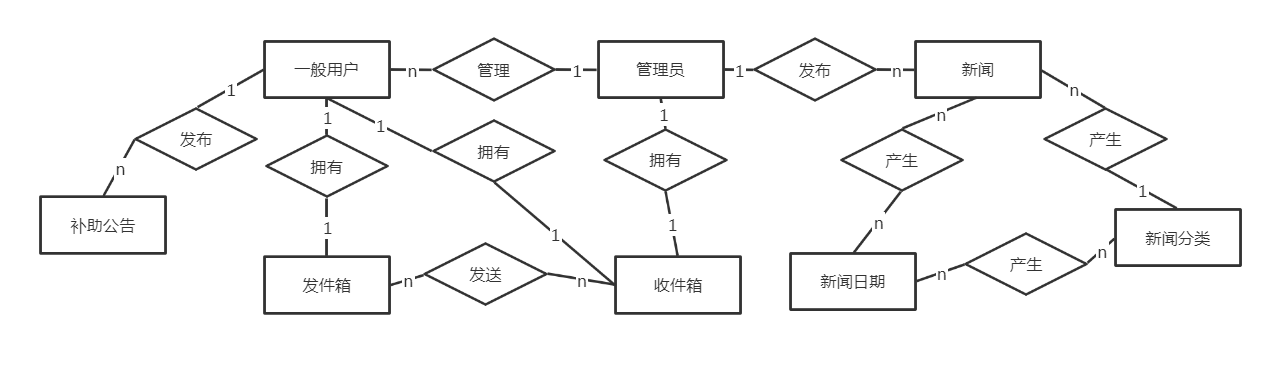


图4-1 实体关系E-R图

一般用户的实体关系包括登录的id，登录密码，姓名，电话号码，性别，婚姻情况，地址，如下图4-2所示。

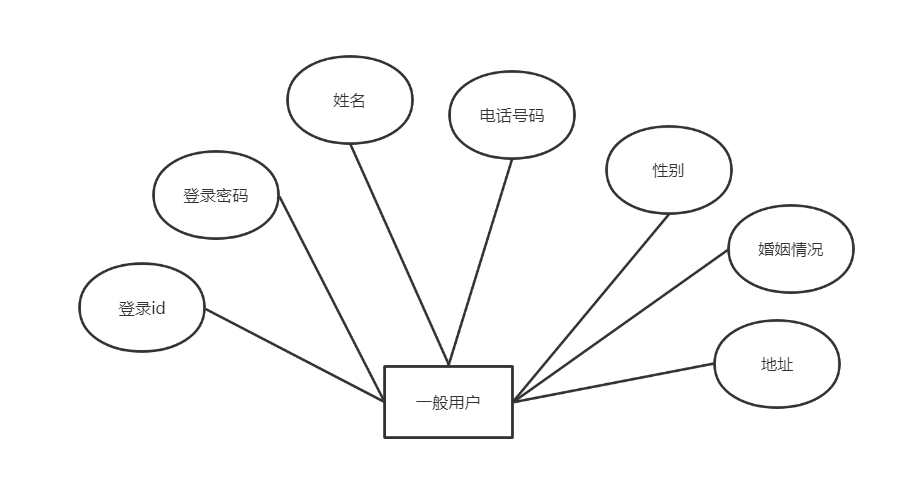


图4-2 一般用户实体关系E-R图

管理员的实体关系包括登录id，登录密码，姓名，电话号码，性别，婚姻情况，地址，如下图4-3所示。

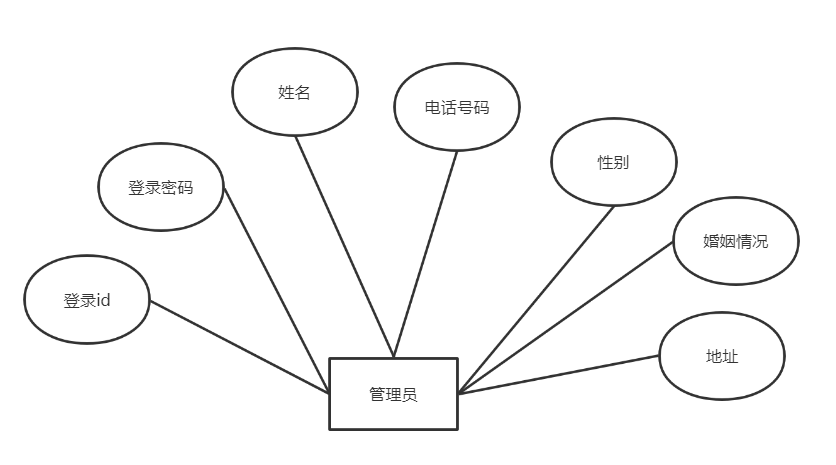


图4-3 管理员实体关系E-R图

新闻的实体关系主要包括新闻标题，新闻来源，新闻编号，新闻时间，如下图4-4所示。

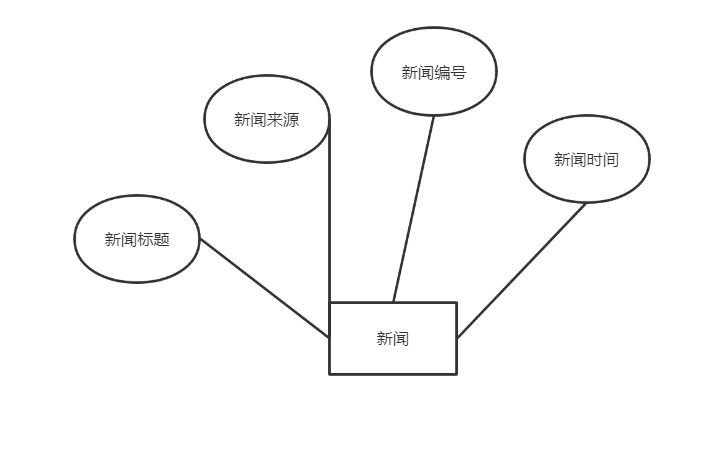


图4-4 新闻实体关系E-R图

发件箱的实体关系主要包括邮件标题，邮件内容，邮件目标以及邮件发送时间，如下图4-5所示。

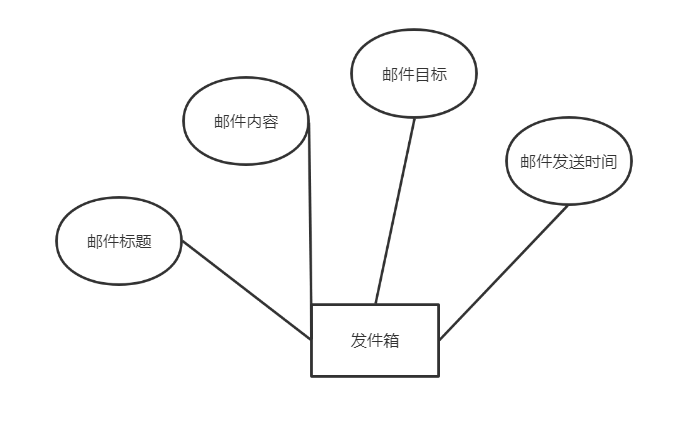


图4-5 发件箱实体关系E-R图

收件箱的实体关系主要包括邮件标题，邮件内容，邮件目标以及邮件发送时间，如下图4-6所示。

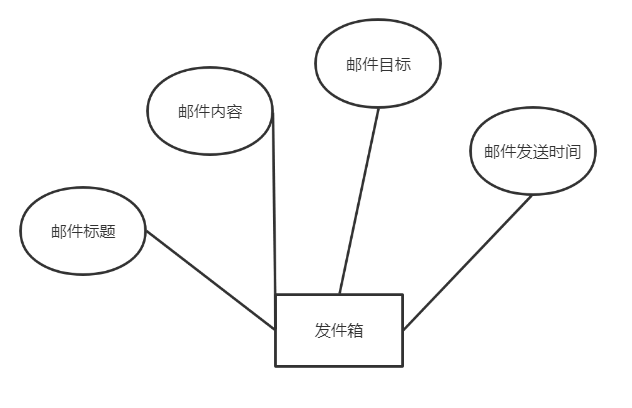


图4-6 收件箱实体关系E-R图

补助公告的实体关系主要包括补助来源，补助金额，补助内容，补助提交时间，如下图4-7所示。

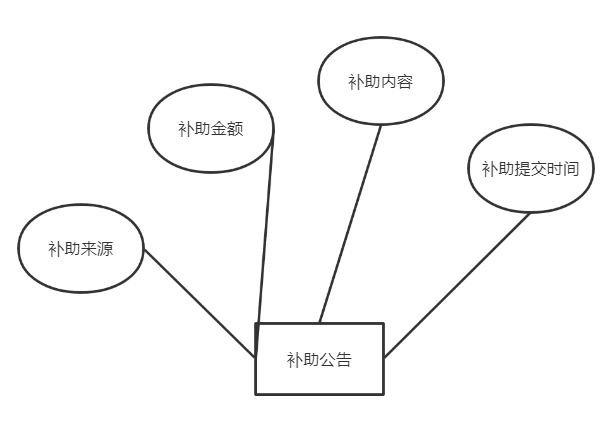


图4-7 补助公告实体关系E-R图

## 4.2 逻辑结构设计

### 4.2.1 设计思路

在此数据库中，每个实体类都增加了属于自己的id，例如以一般用户为主要实体，其属性“用户id”与其紧密相连，相对的，属性“管理员id”对应着管理员的实体；具体属性如下：

（1）一般用户（登录ID，登录密码，姓名，电话号码，性别，婚姻情况，家庭住址）

（2）管理员（登录ID，登录密码，姓名，电话号码，性别，婚姻情况，家庭住址）

（3）新闻（新闻标题，新闻来源，新闻编号）

（4）发件箱（邮件内容，邮件标题，邮件目标）

（5）收件箱（邮件内容，邮件标题，邮件目标）

（6）补助公告（补助来源，补助金额，补助内容，补助提交时间）

### 4.2.2 逻辑模型

一般用户基本信息表内包括有主键id，登录密码，姓名，电话号码，性别，婚姻情况，家庭住址，如下表4-1所示。

表4-1 一般用户基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 默认值 | 描述 |
| id | 整型 | 11 |  | 主键id |
| password | 字符型 | 32 |  | 登录密码 |
| name | 字符型 | 50 |  | 姓名 |
| phone | 字符型 | 11 |  | 电话号码 |
| gender | 字符型 | 1 | 1 | 性别 |
| marriage | 字符型 | 4 |  | 婚姻情况 |
| address | 字符型 | 100 |  | 家庭住址 |

管理员基本信息表内包括有主键id，登录密码，姓名，电话号码，性别，婚姻情况，家庭住址，如下表4-2所示。

表4-2 管理员基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 默认值 | 描述 |
| id | 整型 | 11 |  | 主键id |
| password | 字符型 | 32 |  | 登录密码 |
| name | 字符型 | 50 |  | 姓名 |
| phone | 字符型 | 11 |  | 电话号码 |
| gender | 整型 | 1 | 1 | 性别 |
| marriage | 字符型 | 4 |  | 婚姻情况 |
| address | 字符型 | 100 |  | 家庭住址 |

新闻基本信息表内包括的主要有新闻标题，新闻来源，新闻编号，如下表4-3所示。

表4-3 新闻基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 默认值 | 描述 |
| title | 整型 | 16 |  | 新闻标题 |
| source | 整型 | 100 |  | 新闻来源 |
| number | 字符型 | 16 |  | 新闻编号 |

发件箱基本信息表内包括的主要有邮件标题，邮件内容，邮件目标，如下表4-4所示。

表4-4 发件箱基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 默认值 | 描述 |
| title | 整型 | 16 |  | 邮件标题 |
| content | 整型 | 100 |  | 邮件内容 |
| target | 字符型 | 16 |  | 邮件目标 |

收件箱基本信息表内包括的主要有邮件标题，邮件内容，邮件目标，如下表4-5所示。

表4-5 收件箱基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 默认值 | 描述 |
| title | 整型 | 16 |  | 邮件标题 |
| content | 整型 | 100 |  | 邮件内容 |
| target | 字符型 | 16 |  | 邮件目标 |

补助公告基本信息表内包括的主要有补助来源，补助金额补助内容，补助时间，如下表4-6所示。

表4-6 补助公告基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 默认值 | 描述 |
| source | 字符型 | 4 |  | 补助来源 |
| amount | 字符型 | 10000 |  | 补助金额 |
| content | 字符型 | 1000 |  | 补助内容 |
| time | 字符型 | 100 |  | 补助时间 |

## 4.3 物理结构设计

### 4.3.1 存取方式

为各个实体的常用字段设置索引，比如新闻实体的分类字段，为常用字段，设置为索引，降低查找消耗。同时查询时，避免查询所有字段，只将需要的字段取出来即可。

### 4.3.2 存储结构

顺序存储结构，把逻辑上相邻的那些结点都储存在物理位置上相邻的存储单元里，结点之间的逻辑关系由存储单元的邻接关系来体现的索引储存方法，然后索引表建立完成，通过关键字来存储查询信息。

5 界面设计

## 5.1 界面关系图或工作流图

5.1.1一般用户客户端界面关系图

通过系统设计的描述，编写了了一般用户的客户端界面的关系，一般用户先在登录页面登录，然后进入网站首页可以查看网站首页的新闻页面与补助页面，同样进入自己的个人中心页面也可以在后台去查看自己的个人信息页面，如下图5-1所示。

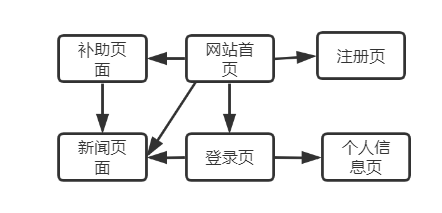


图5-1一般用户客户端界面关系图

5.1.2管理员客户端界面关系图

同样也通过系统设计的描述，编写了管理员用户的客户端的界面关系，管理员在登录页面登录以后，管理员不光可以在网站首页查看首页的新闻与补助页面，同时也可以进入个人中心页面在后台管理新闻，补助，用户信息页面，如下图5-2所示。

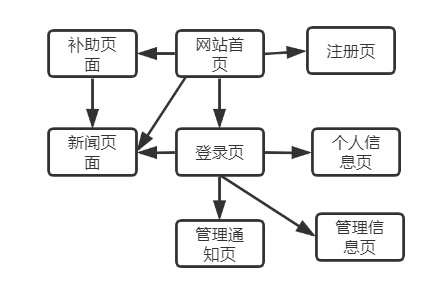


图5-2管理员客户端界面关系图

## 5.2 界面设计成果

### 5.2.1 主界面

主界面的效果展示如下图5-3所示。



图5-3游客客户端主界面效果图

### 5.2.2 子界面

一般用户客户端的个人信息页面展示，如下图5-4所示。



图5-4一般用户客户端个人信息页面效果图

管理员客户端的个人信息页面展示，如下图5-5所示。



图5-5管理员客户端个人信息页面效果图

邮件发送页面展示如下图5-6所示。

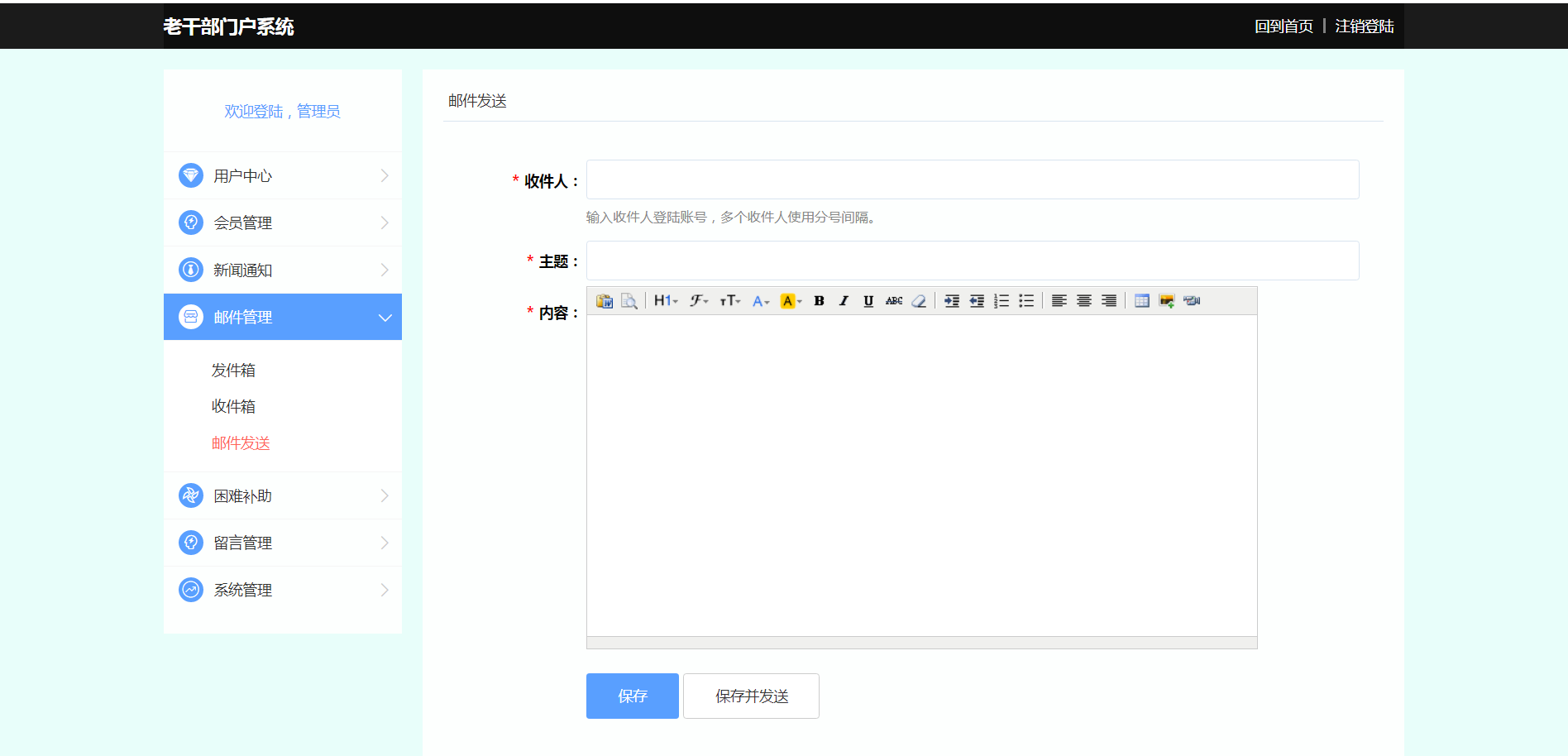


图5-6邮件发送页面效果图

申请补助页面展示如下图5-7所示。



图5-7申请补助页面效果图

新闻添加页面展示如下图5-8所示。



图5-8新闻添加页面效果图

补助首页展示如下图5-9所示。



图5-9补助首页页面效果图

留言板页面展示如下图5-10所示。



图5-10留言板页面效果图

登录页面展示如下图5-11所示。



图5-11登录页面效果图

6 详细设计

## 6.1 系统主要功能模块介绍

（1）新闻管理模块

新闻管理模块具有如下功能，管理员能够在此页面添加与删除新闻，并发布新闻或者公告在首页供用户和游客浏览，新闻有分类供用户分开查看消息。

（2）困难补助模块

困难补助模块是供用户申请困难补助然后由管理员审核补助是否符合自己的标准，若符合则申请补助成功，申请成功的补助可以在首页中的补助分类中查看到自己申请的补助，若不符合则申请补助失败，补助分3个分类供申请人来申请。

（3）邮件信管理模块

每个用户都具有发送与收到邮件的功能，用户之间可以互相发送邮件，也可以收到管理员发送的邮件通知。

（4）留言板管理模块

必须是本网站的用户才能够在留言板中留言，留言可以在管理员的留言管理板块中找到留言方便听从用户的建议进行改善。

1. 人员信息管理模块

管理员可以通过后台来管理用户的个人信息，包括对人员信息的增加，删除，修改，查看，方便直接去管理用户。

## 6.2新闻管理模块

### 6.2.1 功能结构设计

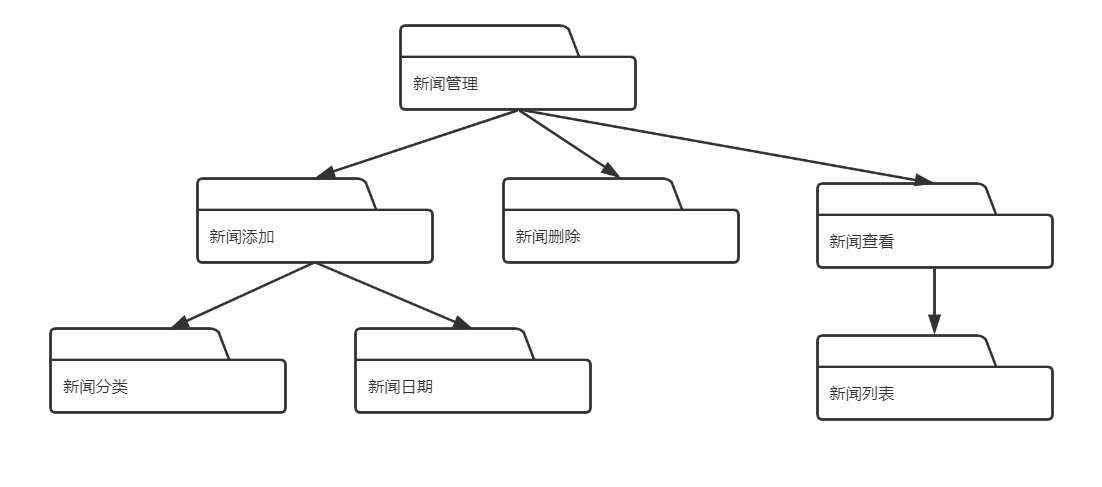


图6-1新闻管理包图

“新闻管理”功能主要目的是通过管理员在后台添加新闻供用户在前端网页中查看，添加新闻时要选择新闻分类跟新闻日期才能够添加，也可以将过时的新闻删除，在新闻列表中查看新闻，便可以通过自己喜欢的分类来浏览新闻。

### 6.2.2 类图设计

新闻管理类的类图包括新闻添加类，新闻删除类，新闻查看类，新闻列表类，如下图6-2所示。

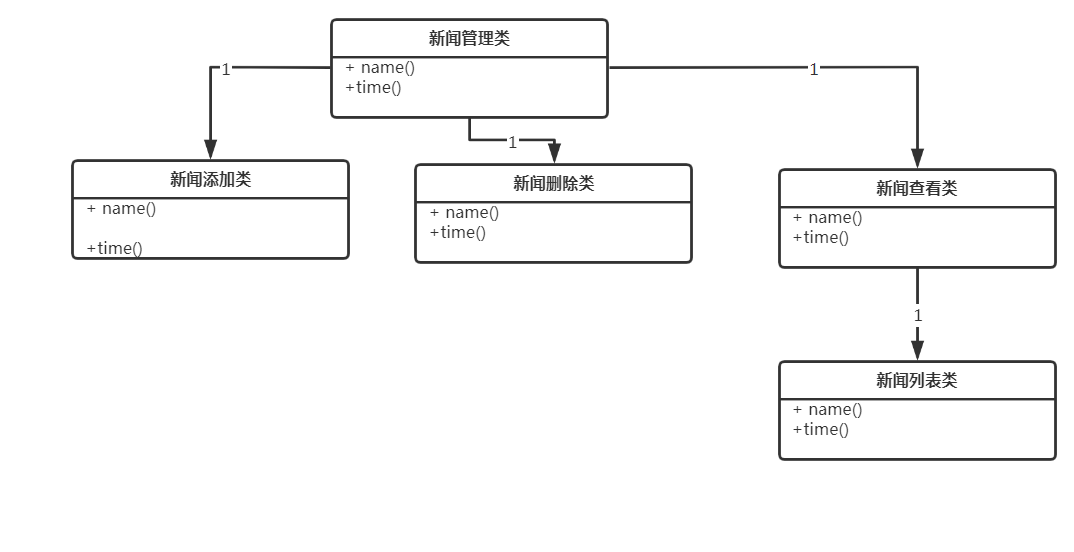


图6-2新闻管理类图

新闻添加类的描述如下表6-2所示。

表6-2新闻管理界面类描述

|  |  |  |
| --- | --- | --- |
| 新闻添加界面类 | | |
| 类的描述：负责添加新闻 | | |
| 操作描述 | Addnewsname() | 添加新闻标题 |
| Addnewstime() | 添加新闻时间 |
| Addnewscontent() | 添加新闻内容 |
| Addnewslist() | 添加新闻分类 |

新闻删除类的描述如下表6-3所示。

表6-3新闻删除类描述

|  |
| --- |
| 新闻删除类 |
| 类的描述：删除掉过时的新闻 |

续表6-3

|  |  |  |
| --- | --- | --- |
| 操作描述 | Deletenewsname() | 删除新闻标题 |
| Deletenewstime() | 删除新闻时间 |
| Deletenewscontent() | 删除新闻内容 |
| Deletenewslist() | 删除新闻分类 |

新闻查看类的描述如下表6-4所示。

表6-4新闻查看类描述

|  |  |  |
| --- | --- | --- |
| 新闻查看类 | | |
| 类的描述：通过首页查看新闻 | | |
| 操作描述 | Searchnewslist() | 根据分类查看新闻 |
| Getnewslist() | 获取所有分类的新闻 |

### 6.2.3 顺序图设计

新闻管理的顺序图如下图6-5所示。

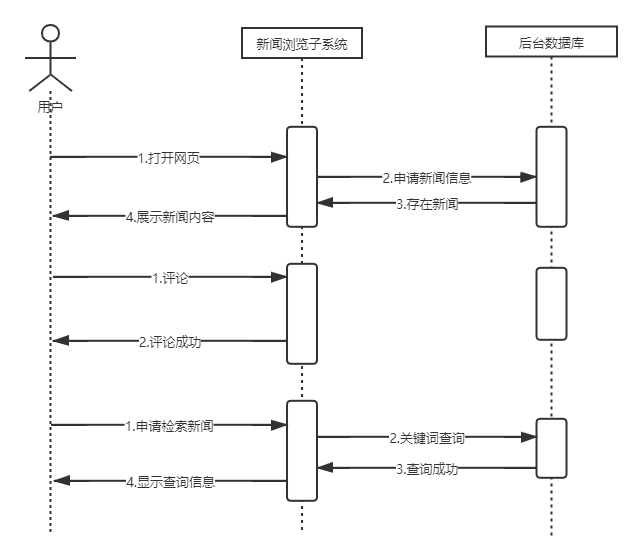


图6-5新闻管理顺序图

### 6.2.4 核心处理流程设计

新闻管理活动图如下图6-6所示。

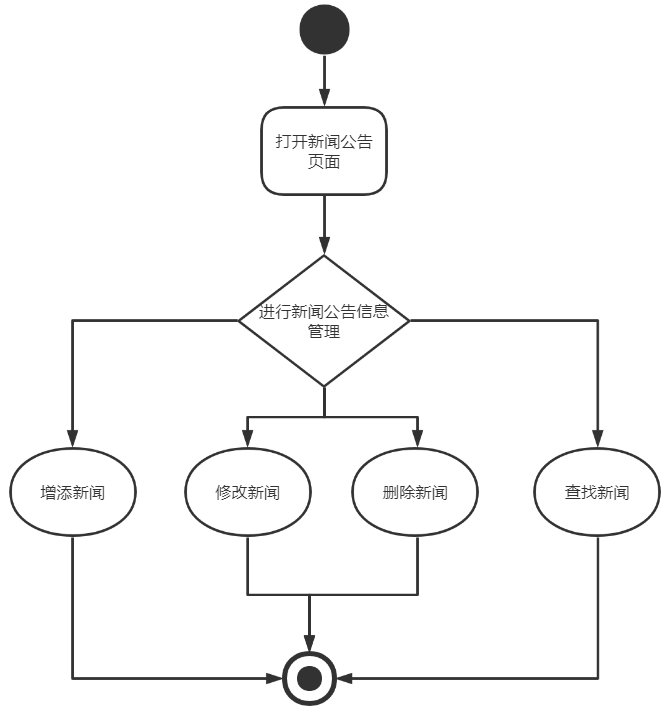


图6-6新闻管理活动图

## 6.3补助申请模块

### 6.3.1功能结构设计

补助管理包括对补助的增删改查以及对补助的审核，如下图6-7所示。

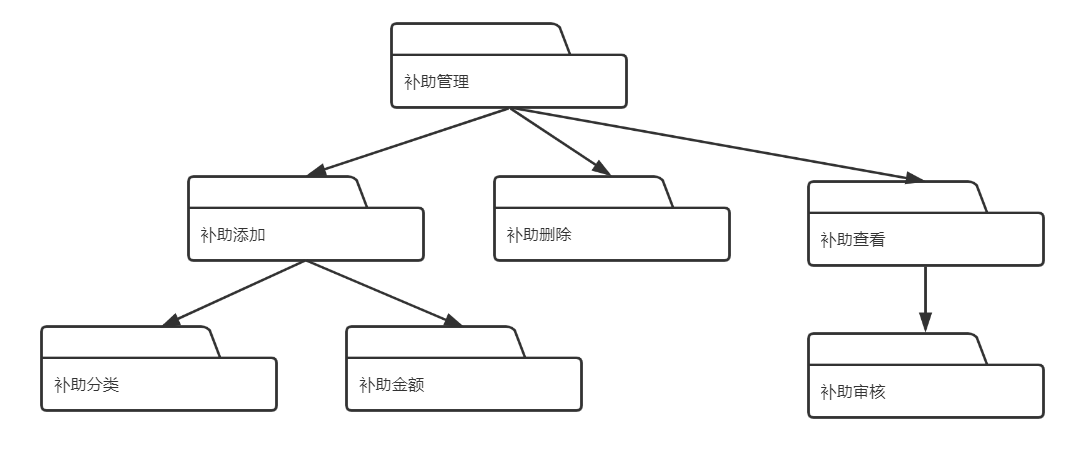


图6-7补助管理包图

### 6.3.2类图设计

补助管理类图如下图6-8所示。

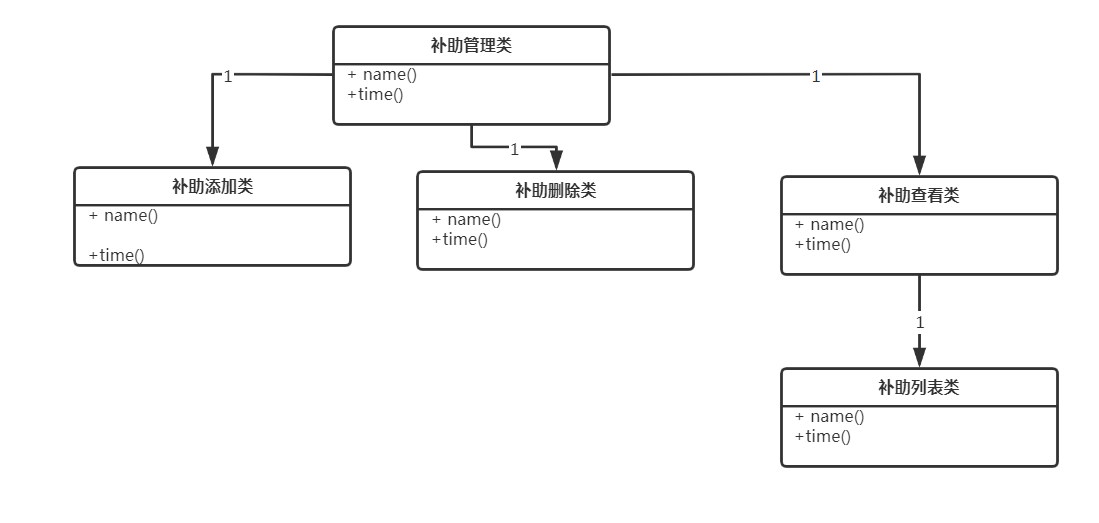


图6-8补助管理类图

### 6.3.3顺序图设计

补助管理顺序图介绍了用户发布补助到审核补助到发布成功的顺序，如下图6-9所示。

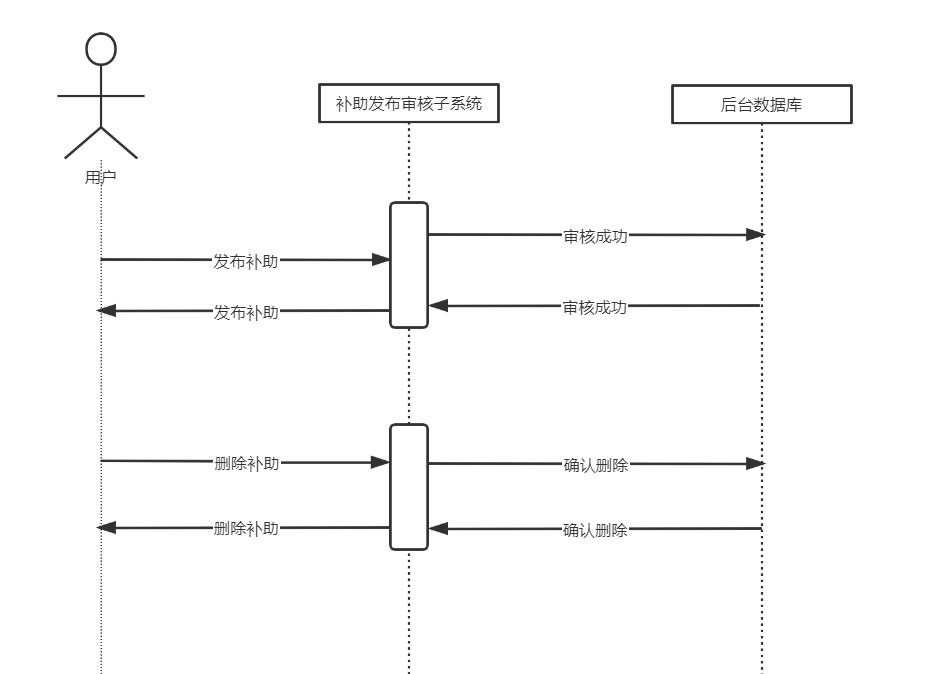


图6-9补助管理顺序图

### 6.3.4核心处理流程设计

补助管理的流程从申请补助到申请成功的流程图如下图6-10所示。

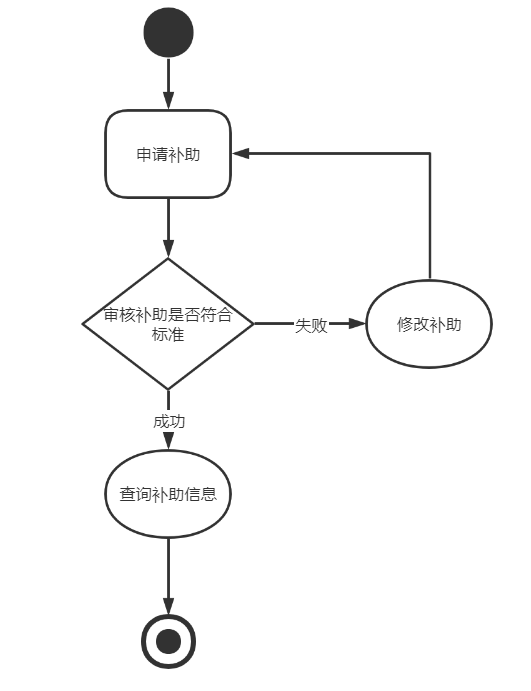


图6-10补助管理流程图

## 6.4邮件信管理模块

### 6.4.1功能结构设计

邮件信管理包括邮件的发送以及对邮件的接收，如下图6-11所示。

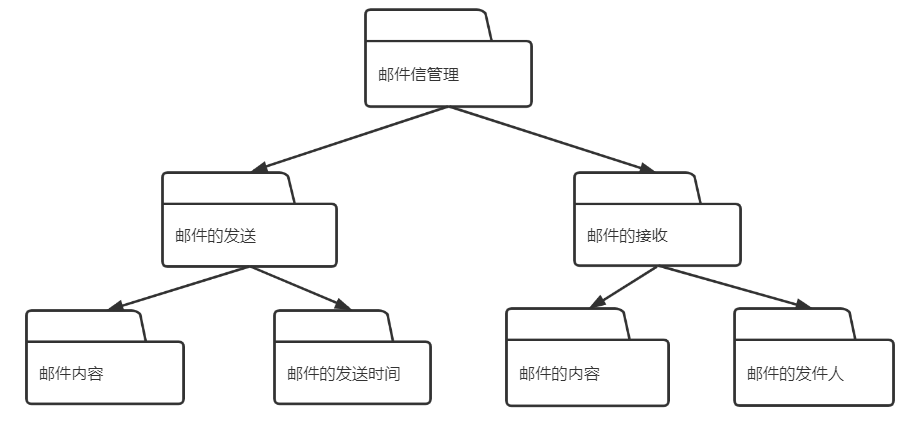


图6-11邮件信管理包图

### 6.4.2类图设计

邮件信管理类图如下图6-12所示。

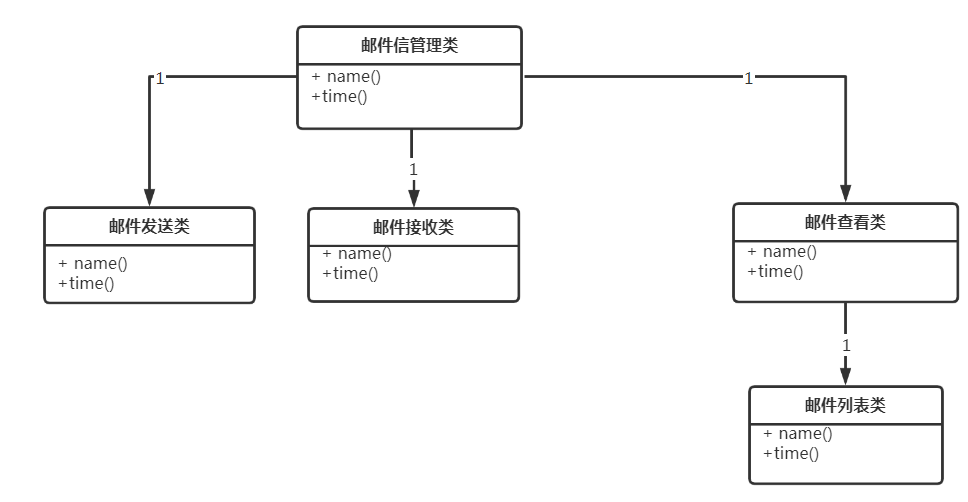


图6-12邮件信管理类图

### 6.4.3顺序图设计

邮件信管理顺序图明确地表述了邮件是如何发送以及接收的，如下图6-13所示。

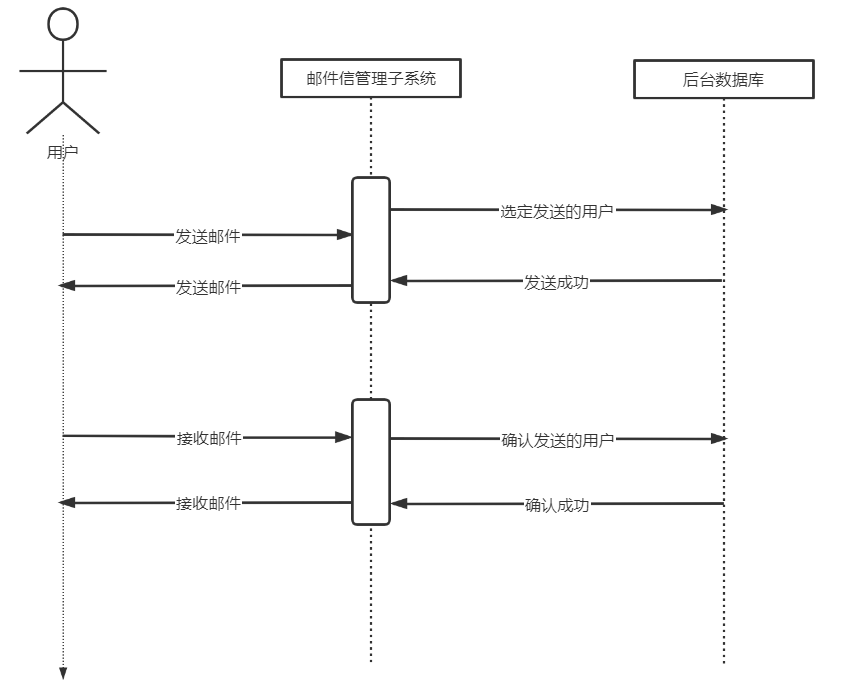


图6-13邮件信管理顺序图

### 6.4.4核心处理流程设计

邮件信管理的流程从发送邮件到接收邮件的流程图，如下图6-14所示。

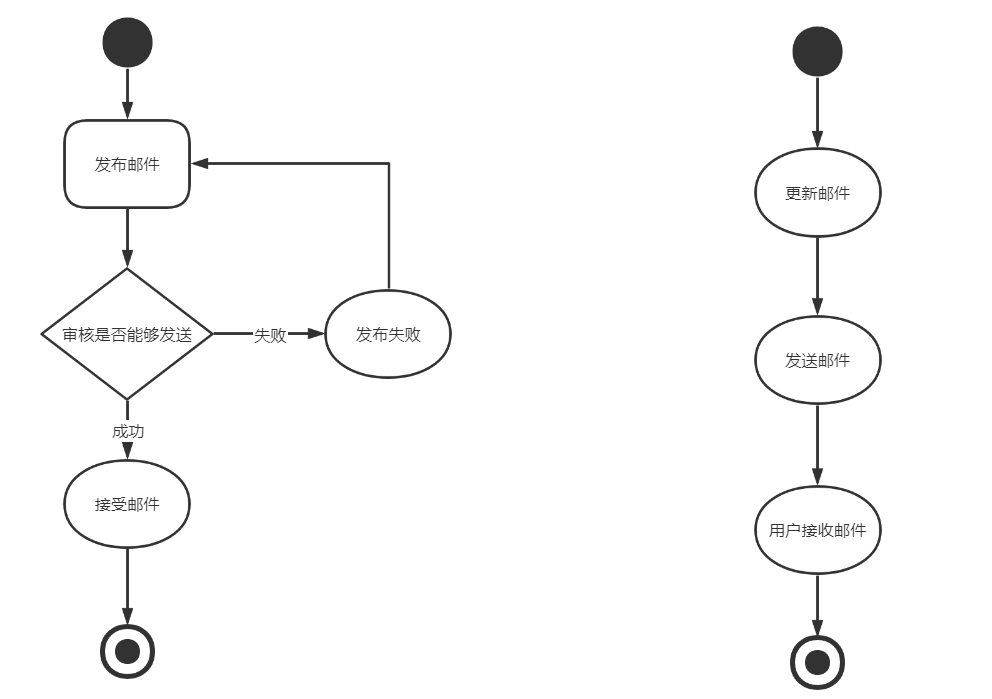


图6-14邮件信管理流程图

## 6.5留言板管理模块

### 6.5.1功能结构设计

留言板包括用户的留言以及管理员在后台的查看，如下图6-15所示。

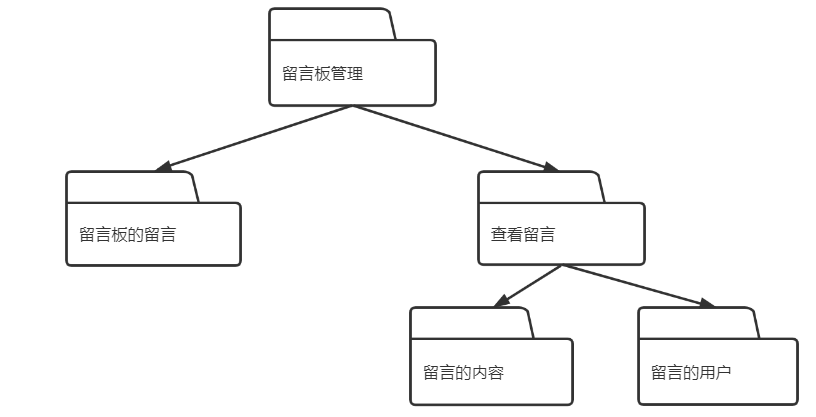


图6-15留言板管理包图

### 6.5.2类图设计

留言板管理类图如下图6-16所示。

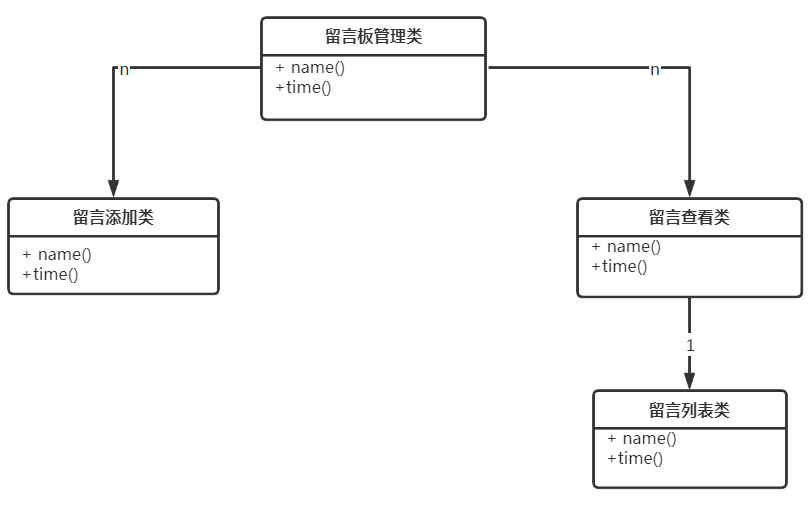


图6-16留言板管理类图

### 6.5.3顺序图设计

留言板管理顺序图表述了用户如何发送留言以及管理员如何查看留言，如下图6-17所示。

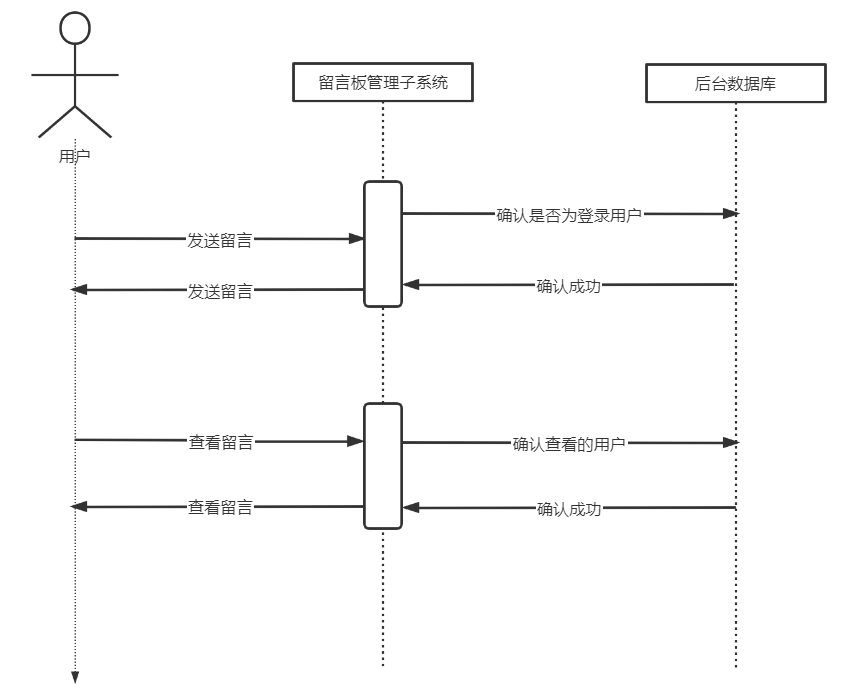


图6-17留言板管理顺序图

### 6.5.4核心处理流程设计

留言板管理的流程是发送留言的流程图，如下图6-18所示。

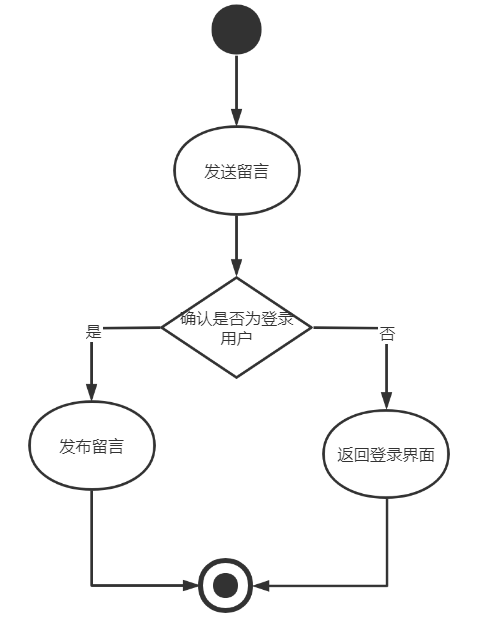


图6-18留言板管理流程图

## 6.6人员信息管理模块

### 6.6.1功能结构设计

人员信息管理模块包括对人员信息的增删改查，如下图6-19所示。

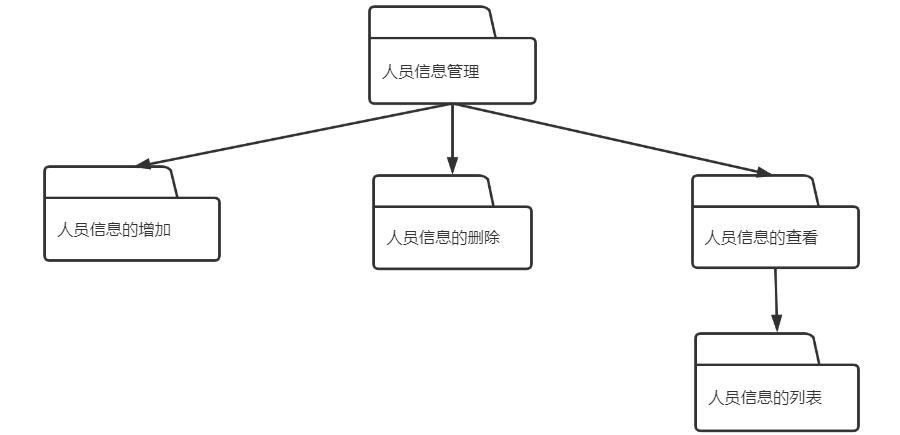


图6-19人员信息管理包图

### 6.6.2类图设计

人员信息管理类图如下图6-20所示。

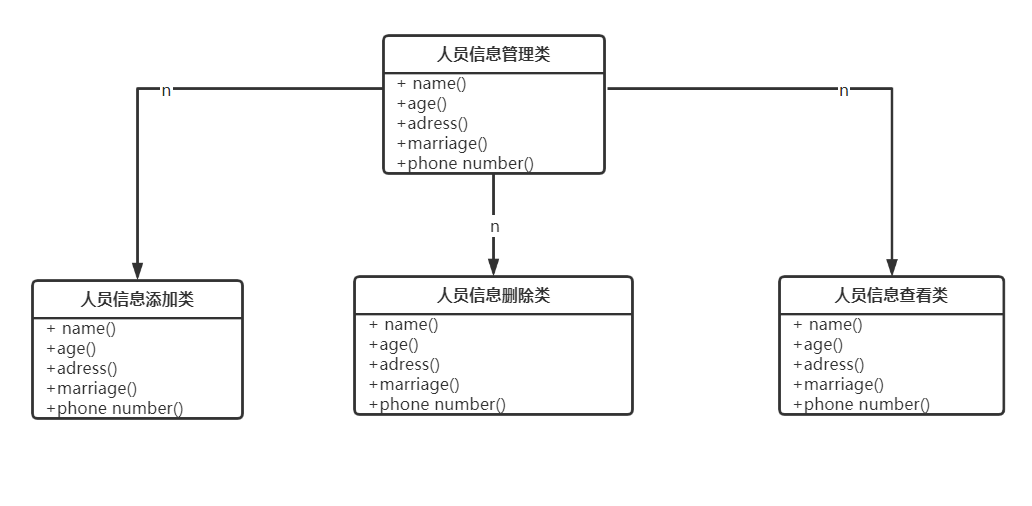


图6-20人员信息管理类图

### 6.6.3顺序图设计

人员信息管理顺序图表述了管理员如何修改用户的个人信息，如下图6-21所示。

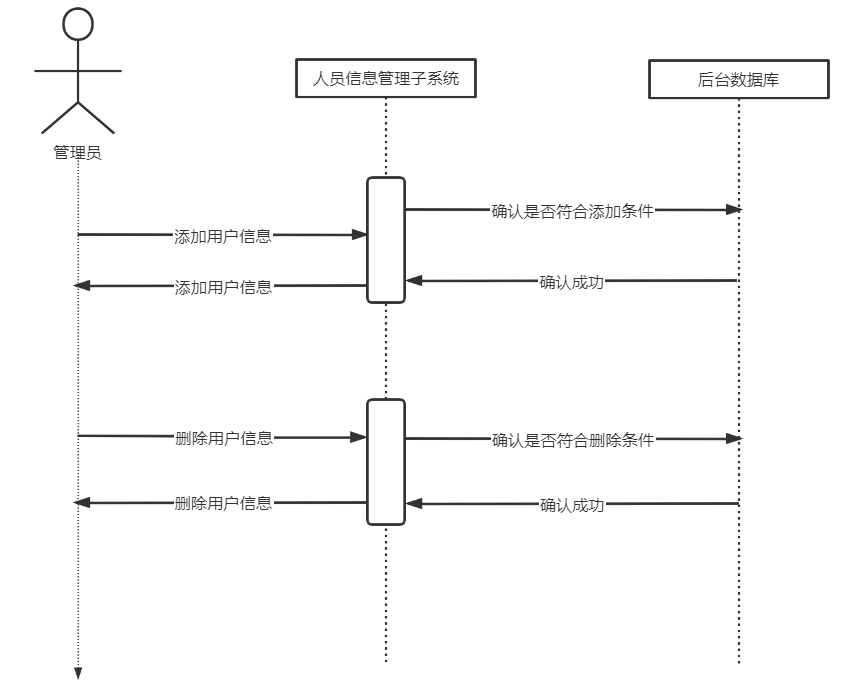


图6-22人员信息管理顺序图

### 6.6.4核心处理流程设计

人员信息管理的流程是对人员信息的增删改查的流程图，如下图6-23所示。

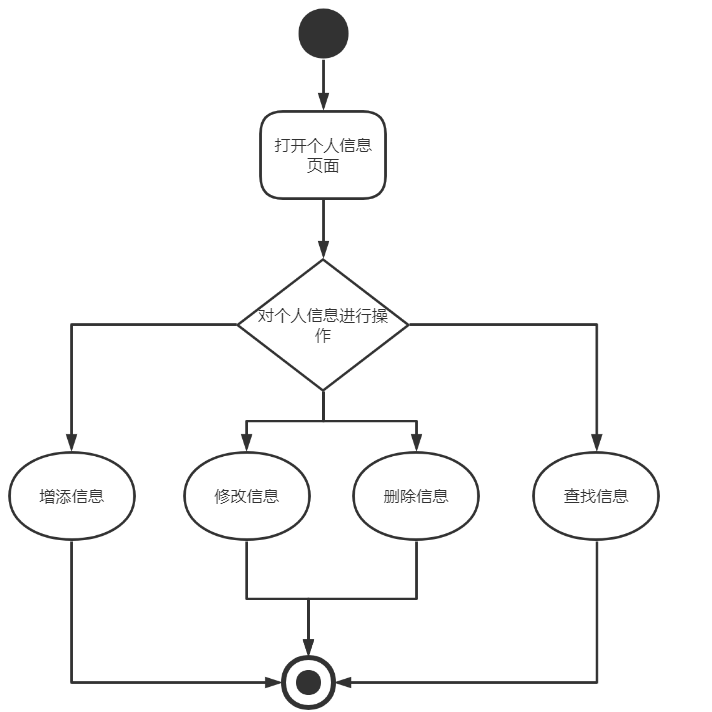


图6-23人员信息管理流程图

# **7 编码**

## 7.1 代码实现与核心算法

文件存放位置为云服务器，故文件上传采用的是服务器提供的接口，代码实现与核心算法如下：

主要是新闻通知管理的代码实现



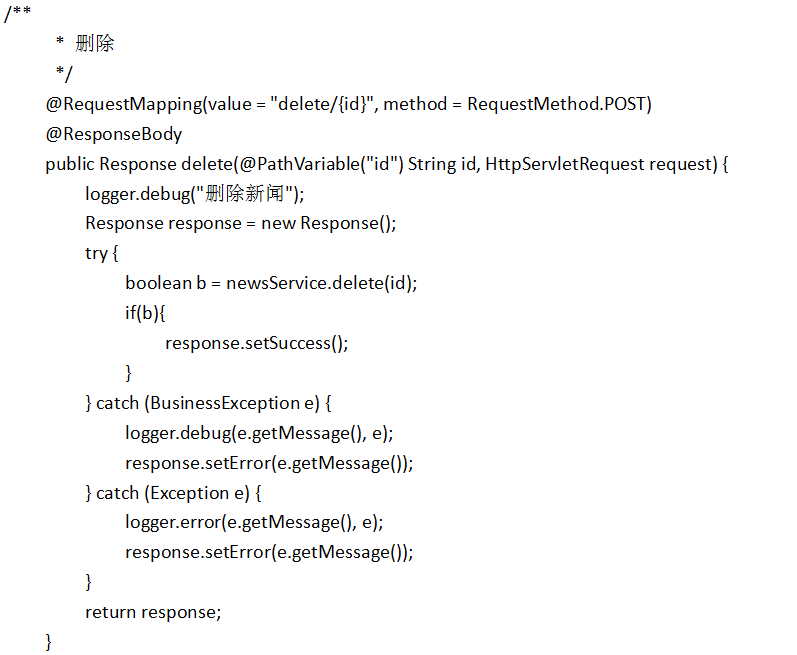
添加新闻的时候先在后台向前端发送添加的新闻内容，如果标题，内容，来源其中有一个为空，则不能添加新闻，如果标题，内容，来源均不为空，则添加新闻成功，并将数据保存在数据库中。



修改新闻的时候先在后台向前端发送要修改的新闻内容，如果标题，内容，来源其中有一个为空，则不能修改新闻，如果标题，内容，来源均不为空，则修改新闻成功，并将数据保存在数据库中。



删除新闻的时候先在后台向前端发送删除的新闻内容，向后台发送请求删除新闻的数据，便可以直接删除新闻。



## 7.2 代码优化分析

（1）代码模块化与接口式开发后，代码规范不少，同时注释也模板化，请求入参、响应出参都继承实现相应的基类，给人以整洁与专业化的感觉，但同时代码重复也增多，后期优化的话应该从增加基类，父类，接口的角度出发，进一步增加代码的复用率，同时前端也应该抽离出公共请求方法，免得在每个请求上重写只有请求接口，请求参数不一样的请求方法。

（2）应提高代码的注释量与注释的表述方式的专业程度。

（3）公共参数与标识等固定值应抽离出来，放在公共模块，用枚举或常量的方式呈现，避免写错或下次引用的时候不清楚其具体属性。

8 测试

## 8.1 测试方案设计

### 8.1.1 测试策略

进行系统测试，主要为功能测试与验收测试，同时进行UI测试。测试的时候尽量所有的测试可能性全部都要测试到，以免出现误差导致不与测试结果相匹配。

### 8.1.2 测试进度安排

在每个功能点完成时进行功能测试，每个模块完成时进行功能测试与UI测试，项目完成后进行验收测试。从最基础的登录注册模块开始进行测试，多测试几个不同的数据，包括对的数据与错的数据都需要进行测试，然后再是新闻管理模块，补助管理模块，邮件信管理模块，留言板模块，都要一个接着一个去测试，并写出它们的测试结果。

### 8.1.3 测试资源

测试人力资源为本开发人员，测试工具为Google Chrome，Firefox。

### 8.1.4 关键测试点

（1）一般用户登录测试

（2）补助添加测试

（3）新闻添加测试

（4）邮件发送测试

（5）留言板功能测试

## 8.2 测试用例构建

### 8.2.1 测试用例编写约定

（1）能够用最直接最简单最清楚地表达出一个测试用例，此测试用例只能用来实现一个功能点，让读者能够清楚地知道我这个用例是用来干什么的，不用让他们太复杂地去思考。

（2）每一个测试用例都应该有一个类似于代号一样简单的代码，让读者能够直观地看出这个代号代表的用例是干什么的，增强理解。

（3）所有的测试用例的流程一定要清楚仔细简单明了，让读者一看就知道该如何一步一步地操作，增强它的可操作性。

（4）所有的测试用例的数据要准确无误，输入的执行前提条件跟工作流程要保证万无一失，测试用例在测试的时候才能体现出很高的可操作性，读者也能很明白测试的结果如何，也能验证出结果是否正确。

（5）每一个测试用例需要有自己的独立性，就是每个功能不重复，各自完成各自的功能，每个流程不会去互相包含，没有重复与冗余，各自满足各自的功能条件。

（6）用例描述要清楚直接，运用特定的术语，不要描述得模模糊糊，让人不知道你在讲什么东西，不知道该如何去操作，给人一种云里雾里的感觉，应避免人们对此系统的描述产生这种感觉。

（7）测试用例中也需要一些操作错误的数据，让读者明白这样操作是不合理的，使他们在使用程序时能够少走弯路，考虑周全。

（8）每一个测试用例都应该保证能够完成整个系统设计对应的每个功能模块的所有功能，考虑到每个数据的不同导致结果的不同要进行多次测试，写出多次测试的多次不同的结果，让读者明白各个用例各个功能的不同测试结果。

（9）有一些无输入的操作更应该详细地描述操作过程，并写出他们的结果，做到认真负责。

所有的测试用例都应当要保证数据的准确性，以及操作的多样性跟正确性，不要出现不符合常理的操作，做测试时首先要保证被测试的用例的数据正确然后才能够进行下一步，预期的输出结果应当与测试过程得出的结果相吻合，同样，错误的操作也应当与程序响应的结果相吻合。

### 8.2.2 测试用例设计

测试用例编写包括测试用例编号、测试标题、测试的重要级别、测试输入、操作步骤、预期结果这几项，此次测试围绕如下几个方面：用户登录、新闻添加、补助添加、邮件发送、留言板功能。

### 8.2.3 关键测试用例

表 8-1 关键测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 测试标题 | 级别 | 操作步骤 | 预期结果 |
| XHEDU-F-01 | 测试一般用户登录时，不输入用户名密码，网页响应情况 | 高 | 1.进入客户端登录界面  2.点击登录按钮 | 网页提示：请输入用户名密码 |
| XHEDU-F-02 | 测试一般用户登录时，输入错误用户名密码，网页响应情况 | 高 | 1.进入客户端登录界面  2.输入错误的用户名和密码  3.点击登录按钮 | 网页提示：用户名或密码错误 |
| XHEDU-F-03 | 测试一般用户登录时，输入正确用户名密码，网页响应情况 | 高 | 1.进入客户端登录界面  2.输入正确的用户名和密码  3.点击登录按钮 | 网页跳转到客户端首页 |
| XHEDU-F-04 | 测试一般用户登录时，管理员审核不通过，网页响应情况 | 高 | 1.进入客户端登录界面  2.输入正确的用户名和密码  3.点击登录按钮 | 网页提示：您提交的注册信息未通过注册，请重新注册 |

续表8-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| XHEDU-F-05 | 测试新闻添加时不添加标题时，网页的响应情况 | 高 | 系统正常运行，管理员已登录进入管理页面 | 1.进入管理员的新闻添加页面  2.添加新闻 | 网页提示：请输入标题 |
| XHEDU-F-06 | 测试添加新闻时不选择分类，网页的响应情况 | 高 | 系统正常运行，管理员登录进入管理页面 | 1.进入管理员的新闻添加页面  2.输入标题  3.点击发布 | 网页提示：请选择分类 |
| XHEDU-F-07 | 测试发布困难补助审核不通过时网页的响应情况 | 高 | 系统正常运行，一般用户已提交补助申请 | 1.进入管理员的补助列表界面  2.选择审核列表  3.点击拒绝审核 | 网页提示：补助的状态是审核拒绝 |
| XHEDU-F-08 | 测试发布困难补助审核通过时网页的响应情况 | 高 | 系统正常运行，一般用户已提交补助申请 | 1.进入管理员的补助列表界面  2.选择审核列表  3.点击同意审核 | 网页提示：补助的状态是审核成功 |
| XHEDU-F-09 | 测试邮件发送时不输入收件人，网页的响应情况 | 高 | 系统正常运行，用户进入邮件发送页面 | 1.进入邮件发送页面  2.点击发送邮件 | 网页提示：请输入发件人 |
| XHEDU-F-10 | 测试邮件发送时不输入邮件主题，网页的响应情况 | 高 | 系统正常运行，用户进入邮件发送页面 | 1.进入邮件发送页面  2.输入收件人  3.点击发送邮件 | 网页提示：请输入邮件主题 |
| XHEDU-F-11 | 测试邮件发送时输入了邮件主题输入了收件人与邮件内容，网页的响应情况 | 高 | 系统正常运行，用户进入邮件发送页面 | 1.进入邮件发送页面  2.输入收件人  3.输入邮件主题  4.编辑好邮件内容  5.点击发送 | 网页提示：邮件发送成功 |

### 8.2.4 测试用例维护

（1）若后期增加或修改了相应功能点，则应该新增或修改测试用例；

（2）删除因新功能增加而多余的用例；

9 总结与展望

## 9.1 设计工作总结

本次毕业设计，算是竭尽我所能来完成的，由于疫情关系已经差不多3个月没与老师同学见面，要是能够去企业公司那边上课经过老师的指导再与同学们一起讨论的话我的进度一定能快不少，这时候孤立无援的情况下自己只能硬着头皮去自学，找同学要了视频来看，然后不懂的地方去向前辈或者已经是从事软件工作的人去询问如何完成，花了不少金钱与精力，当然老师也是非常关心我们一周会询问我们的进度情况不懂的问题可以问老师，虽然我不怎么习惯这种远距离的聊天方式，不过还是感谢老师的一番好意，在设计系统的时候其实是挺简单的，通过自己慢慢思考先把系统设计设计好，系统设计当时想着是要有个新闻管理，补助管理，邮件信管理，人员信息管理与人员权限管理，后来由于前端过于简陋才加上了一个留言板管理模块，由于对SSM框架还不怎熟悉所以写起来代码还是存在着一些困难，比如刚设计的时候还不知道一些语句的意思，不能够灵活运用。在插入富文本的时候也想了挺久，最后在Git上找了个模板直接套用，至少效果还是立竿见影的。

在设计此系统的最后其实也不是很圆满，还有一些应该完成的东西没有完成，比如我觉得这个补助申请应该更详细一点，再设置一个超级管理员专门来审核补助，以及再设计一个可以提交补助申请文件的富文本，总的来说，功能比较齐全但是抠起细节来看的话每个模块做的也不是很完美。过程是坎坷的，算是不圆满的完成了这次毕设吧，有很多自己不怎么满意的地方，比如界面问题，但是功能方面我还是挺满意的，想不到自己也能做到这些，在技术上，知识上，心态上都得到了提升跟锻炼，收获还是挺大的。

## 9.2 未来工作展望

一开始通过老师的指导意见告诉我这个网站要是游客查看会怎么办，是不是会有一些限制查看的，我针对老师给的这个建议开始对老干局门户网站游客用户使用用户功能时会跳出什么页面开始花心思去想，后来通过同学跟老师的帮助，我使用了SSM框架的限制最小与最大输出值的方式，将游客用户的访问权限限制了，使用用户功能时直接跳转登录注册页面，这算是一个解决方式吧。同样还有其他功能板块也是通过老师与同学的指导才能完成，在设计前台页面时由于不熟前台的设计所以做的页面不是很美观。

通过设计此门户网站大部分功能还是比较齐全的，就是一些细节值得商榷，比如对老干局的理解不是很透彻，应当再设置一个管理员用来专门管理补助管理，这样的话更符合实际，同样补助的分类也有点小问题，补助的来源不是很正确，这些都是以后制作网站时值得注意的点。

谢 辞

参考文献

[1]刘昕林，罗伟峰. 基于Java平台的Web应用系统业务性能监测分析[J]. 信息与电脑（理论版），2019.

[2]倪海顺. 计算机软件开发的Java编程语言应用探讨[J]. 信息与电脑（理论版），2019.

[3]罗超，龙侃. 基于java的企业进销存系统的设计和实现[J]. 信息记录材料，2019.

[4]刘涛，闵鹏瑾，肖汉. 基于JAVA的小学数学四则运算教学系统的设计与实现[J]. 计算机与数字工程，2018.

[5] Craig Walls. Spring Boot实战[M].丁雪丰，译.北京：人民邮电出版社，2016.

[6]王红伟，李会凯. 计算机软件开发的Java编程语言探究[J]. 无线互联科技，2018.

[7]Abdelhak Mesbah,Jean-Louis Lanet,Mohamed Mezghiche. Reverse engineering a Java Card memory management algorithm[J]. Computers &amp; Security,2017.

[8]范国婷，曹飞虎，王秀友，杨颖.基于Java Web的进销存管理系统设计[J].廊坊师范学院学报（自然科学版），2018.

[9]Ronald A. Olsson,Todd Williamson. RJ: a Java package providing JR‐like concurrent programming[J]. Softw. Pract. Exper.,2016.

[10] 郑阿奇. Java EE实用教程[M]. 北京： 电子工业出版社，2018.

[11] 苟雨晴，兰全祥.基于Java的拼图游戏设计与实现[J].电脑知识与技术，2019.

[12] 陈晓勇. MySQL DBA修炼之道[M]. 北京：机械工业出版社，2017.

[13] 关东升. Java从小白到大牛[M]. 北京：清华大学出版社，2018.

[14] 王祥顺. 软件工程的安全检测和维护[J].电子技术与软件工程，2017 ,（19）：36.

[15] 何为, 迟文恒. Java服务端研发知识图谱[M]. 北京：机械工业出版社 ，2019.

[16] 袁琛. 基于Web技术的民办高校教学管理系统的设计与实现[D].湖南大学,2017.

[17] 张耀春. Vue.js权威指南[M]. 北京： 电子工业出版社，2016.

[18] 杨开振，周吉文，梁华辉，谭茂华. Java EE互联网轻量级框架整合开发 SSM框架（Spring MVC+Spring+MyBatis）和Redis实现[M]. 北京：电子工业出版社，2017.

[19] 阳波. JavaScript核心技术开发解密[M]. 北京：电子工业出版社，2018.

[20] 肖美婷. 移动Web开发技术在商务网站的应用及发展趋势探究[J].中国新通信, 2018, 20 (17):125.

[21] 于博文. 基于计算机软件工程的数据库编程技术[J].中国高新区，2017,（24）：182.

[22] 张甦. MySQL 王者晋级之路[M]. 北京：电子工业出版社，2018.

[23] Craig Walls. Spring Boot实战[M].丁雪丰，译. 北京：人民邮电出版社，2016.

[24] 梁灏. Vue.js实战[M]. 北京：清华大学出版社，2017.

[25] 王福强. Spring Boot揭秘:快速构建微服务体系[M]. 北京：机械工业出版社，2016.

[26] 姜桂洪. MySQL数据库应用与开发[M]. 北京：清华大学出版社，2018.

[27] 贺春旸. MYSQL管理之道:性能调优、高可用与监控[M]. 北京：机械工业出版社，2016.

[28] 张玉宏. JAVA从入门到精通:精粹版[M]. 北京：人民邮电出版社，2018.

[29] 阳波. JavaScript核心技术开发解密[M]. 北京：电子工业出版社，2018.

[30] 刘增杰. MySQL5.7从入门到精通[M]. 北京：清华大学出版社，2016.

[31]Ying Xin Liu,Xi Yuan Li. Design and Implementation of a Business Platform System Based on Java[J]. Procedia Computer Science,2020,166.

[32]Andrea Capiluppi,Nemitari Ajienka. Lexical content as a cooperation aide: A study based on Java software[J]. The Journal of Systems & Software,2020,164.

附录A 外文翻译—原文部分

文章来源：《Lecture on Java and education》—— Samatha Sudarshanam

Distance Education has drawn a great deal of attention in the field of distance and continuing education.Distance Education has proved to enhance learning and teaching environments. The types of available technologies used in distance learning are divided into two groups: synchronous and asynchronous. One of the main advantages of distance leaning is that web-based courses can be accessed at any time or place feasible.

This project is to establish a Web-based distance learning system, where instructors and students

participate in learning activities while geographically separated from each other. This study is aimed to design and implement a distance learning system and use internet as the delivery mode. The motivation of this study is from Hawkins remarks a cheap and easy-to-use technology must be used in schools to allow greater use among students and teachers”.

This report discusses in detail background research on distance education and different modes of delivery technologies used. It also discusses advantages and disadvantages of distance learning. It also gives tips for successful distance learning and who should opt distance learning, as it is not suited for everyone.

This report presents the design and implementation of this new and inexpensive distance le-

arning system. It was developed using Java-based software. It handles three different kinds of us-

ers: student, instructor and admin. Various tools have been developed like course tool,message b-

oard tool:discussion board tool, exam tool, grade tool etc. Based on the role of the users the func-

tionality of the tools changes. Like the instructor has add/edit privilege in creating course materia

and the student has view-only functionality. Admin use does not have access to all the tools. They can add/edit/delete course information and can add/edit/delete user profile.

This report discusses pros and cons of this newly developed distance learning system. These observationsare based on tests performed by ten people. This report concludes the results of the system and suggests future work to be done to the system.

Technology is changing the way teacher teaches and learner learns. Difficulty inobtaining higher education increases for a number of reasons. It could be their possible limit of capabilities with an educational environment or disabilities that limit their access to educational institutions. There are also a number of people that live agreat distance from the educational institutions. The need to commute becomes anencumbrance to achieving higher education. Also, the job markets are so competitivethat employees finđ that they must obtain higher education for further employment success. With synchronous schedules in present educational institutions, individuals arerequired to resign from their current employment to further their education. Typically,individuals are not willing to surrender their income due to the need to support afamily or their current lifestyle. Some form of educational reform must take place toaddress the problems.

Distance education or Distance learning programs are the solution to the aboveproblems. The greatest appeal of distance learning is that one can study without having to leave home or a job to obtain higher education. These programs made itpossible for students to complete their education without having to sacrifice theircareer and family time.

With the introduction of new media and technology for learners and teachers, universities have introduced distance learning/distance education. With the above technologies in mind the objective of this project is to develop a system using this internet as one of the delivery mediums. The objective of this report/project is to design and implement a web-based system that allows interaction between instructors and students. This involves developing an intuitive user interface for both instructor and student. Instructors and students are the external entities to the system who can log into the system and use the functionality provided by the system. The instructors and the students enter the system through a login tool component. The objective of this system is to develop non-expensive interactive tools like message board tool,discussion group tool and chat tool to provide interaction between the student and the instructor. The additional objective of this project is to design a system with reusable components, feasibility and provision for system expansion without compromising system performance.

Distance education dates back to at least as early as 1728, when "an advertisement in the Boston Gazette... 'Caleb Phillips, Teacher of the new method of Short Hand" was seeking students for lessons to be sent weekly. Modern distance education has been practiced at least since Isaac Pitman taught shorthand in Great Britain via correspondence in the 1840s. The development of the postal service in the 19th century leads to the growth of commercial correspondence colleges with nation-wide reach.

In the twentieth century, radio, television, and the Internet have all been used to further distance education. Computers and the Internet have made distance learning distribution easier and faster.

In 2006 the Sloan Consortium reported that more than 96 percent of the largest colleges and universities in the United States offered online courses and that almost 3.2 million U.S. students were taking at least one online course during the fall 2005 term.

The learner can complete most of the classes at their convenience. Most of the classes are a-

synchronous, which means leaner does not have to attend a lecture at a particular time and place. Review of the assignments and doing homework can be done during off-hours or from home.

Live anywhere, study from anywhere while pursuing the education of choice. Learners don’t have to live in the same city or the same country to attend the learning institution of choice. Learner can study wherever they have access to a computer and Internet connection.

Gain extra knowledge. Learner can transfer the computer and Internet skills that they will gain in the process of your distance learning experience to other facets of life.

Self-paced learning. For slow and quick learners. This reduces stress and increases satisfaction.

Accessibility. Online classes address physical accessibility issues that some people with limited mobility encounter when taking traditional classes. Learner can use their comfortable furniture while enjoying free movement and a chance to further education.

Costly and complex technology. Despite the many opportunities of distance education, there are inevitable accompanying costs. Live video communication for example, requires careful planning of the equipment and facilities. For online learning, learner must own a computer (possibly with access to the Internet) or have access to one. Thus required technology is not always available. Some learners may also be afraid of technology.

Advance planning. Both the instructors and students involved in distance learning may need to make sacrifices at times to get things done in time.

Hidden costs. If learner works for the military for example, and they are on the ship, how do

they get the required materials? They may need to be mailed in advance incurring extra shipping and handling costs.

Distance learning does not offer immediate feedback. In a traditional classroom setting, a st-

udent's performance can be immediately assessed through questions and informal testing. With distance learning, a student has to wait for feedback until the instructor has reviewed their work and responded to it.

Distance learning does not always offer all the necessary courses online. Students pursuing a

specific certificate or degree program may not have all the necessary courses available through distance learning so it is not suited for all subjects. While student can study a history lesson completely online but cannot perform nursing clinical online. Thus physical classroom attendance will be mandatory to complete the course.

Distance learning may not be acknowledged by all employers. Although most employers do acknowledge distance learning, certain employers do not. Students who want to work for a specific employer upon graduation should be sure of that employer's perspective about online education.

Distance learning does not give students the opportunity to work on oral communication skills. Students in distance learning courses do not get the practice of verbal interaction with professors and other students.

Social isolation. Most often learner will be studying alone. Distance learners may feel isolated or miss that social physical interaction that comes with attending a traditional classroom. However this impersonality has been lessening with advances and use of communication technologies such as bulletin boards, threaded discussions, chats, email and conferencing.

The software design of this system focuses on the Java-based, client/server type software model. The system is implemented using Java/J2EE technologies. It includes windows XP as the operating system, MySQL as the backend.

Java/J2EE: Java (1.4) is an object oriented programming language (OOPs). With Java complete applications can be built, featuring accelerated 3D graphics and other multimedia features to strong cryptography and network connectivity. On the web, Java can be used on the client side to create applets and on the server side to create dynamic web pages using Servlets and Java Server Pages (JSP).

BEA Weblogic Server: It is Java application server. It is a light weight development server. Fast cycle times with rapid develop/deploy/debug cycles.

MYSQL: The database is designed and implemented using MySQL v4.1. MySQL is free and scaleable.

Struts Framework (MVC model 2): Model-View-Controller is a design pattern. MVC decouples interface from business logic and data. Struts is an MVC implementation that uses Servlets 2.2 and JSP 1.1 tags, from the J2EE specifications, as part of the implementation. Struts helps control change in web project and promote specialization. Struts is a set of cooperating classes, Servlets, and JSP tags that make up a reusable MVC 2 design. This definition implies that Struts is a framework, rather than a library, but Struts also contains an extensive tag library and utility classes that work independently of the framework.

Client browser：An HTTP request from the client browser creates an event. The Web container will respond with an HTTP response.

Controller：The Controller receives the request from the browser, and makes the decision where to send the request. With Struts, the Controller is a command design pattern implemented as a Servlet. The struts- config.xml file configures the Controller.

Business logic：The business logic updates the state of the model and helps control the flow of the application. With Struts, this is done with an Action class as a thin wrapper to the actual business logic.

Model state：The model represents the state of the application. The business objects update the application state. ActionForm bean represents the Model state at a session or request level, and not at a persistent level. The JSP file reads information from the ActionForm bean using JSP tags.

View：The view is simply a JSP file. There is no flow logic, no business logic, and no model information -- just tags. Tags are makes Struts unique compared to other frameworks like Velocity.

附录B 外文翻译—译文部分

文章来源：《Java与教育联系类演讲》——萨玛莎·苏达沙南

远程教育在远程教育和继续教育领域引起了极大的关注。远程教育已经证明可以改善学习和教学环境。远程学习中使用的可用技术类型分为两组：同步和异步。远程学习的主要优点之一是可以在任何时间或地点访问基于网络的课程。

这个项目是建立一个基于网络的远程学习系统，教师和学生参与学习活动，同时在地理上彼此分开。本研究旨在设计和实施远程学习系统，并使用互联网作为交付模式。这项研究的动机来自霍金斯的评论，学校必须使用便宜且易于使用的技术，以便更好地利用学生和教师。

本报告详细讨论了远程教育和使用的不同交付技术模式的背景研究。它还讨论了远程学习的优缺点。它还提供了成功远程学习的技巧，以及谁应该选择远程学习，因为它并不适合所有人。

本报告介绍了这种新的廉价距离的设计和实施学习系统。它是使用基于Java的软件开发的。它处理三种不同的类型用户：学生，教师和管理员。已经开发了各种工具，如课程工具，留言板工具，讨论板工具，考试工具，等级工具等。基于角色用户工具的功能发生变化。就像教师在创建时添加/编辑权限一样课程材料，学生只有查看功能。管理员使用无法访问所有工具。他们可以添加/编辑/删除课程信息，并可以添加/编辑/删除用户轮廓。

本报告讨论了这一新开发的远程学习系统的优缺点。这些观察结果基于十个人进行的测试。该报告总结了系统的结果，并建议将来要对系统进行的工作。

科技正在改变教师的教学和学习者的学习方式。由于种种原因，接受高等教育的困难增.加了。这可能是由于他们受教育环境或残疾限制了他们进入教育机构的能力。还有一些人住在离教育机构很远的地方。通勤成为实现高等教育的障碍。此外，就业市场竞争如此激烈，员工发现，他们必须获得更高的教育，以进一步就业成功。由于目前教育机构的时间表是同步的，个人必须辞去目前的工作，继续接受教育。I 般来说， 个人不愿意放弃他们的收入，因为需要支持一一个家庭或他们目前的生活方式。必须进行某种形,式的教育改革来解决这些问题。

远程教育或远程学习项目是上述问题的解决方案。远程教育最大的吸引力是一个人可以不用离开家或工作来获得高等教育。这些项目使学生能够在不牺牲职业和家庭时间的情况下完成学业。

通过引入新媒体和技术，为学生和教师，各个大学相继推出远程教育/远程教育。考虑到上述技术这一项目的目标是开发使用该网络系统作为传递媒介之一。本报告/项目的目标是设计并实现了一个基于Web的系统，让教师和学生之间的互动。这包括开发既教师和学生直观的用户界面。教师和学生是外部实体谁可以登录到系统并且使用系统提供的功能系统。教官和同学通过登录工具组件进入系统。该系统的目的是开发类似留言板工具非昂贵的交互式工具，讨论组工具 和聊天工具，以提供学生和教师之间的互动。该项目的另一个目标是设计一个系统，可重用的组件， 可行性及提供系统扩充而不影响系统性能。

远程教育可以追溯到至少早在1728，当“波士顿公报的广告......“迦勒菲利普斯，短手的新方法教师”正在寻求学生将每周发送教训。现代远程教育至少已经实行自艾萨克·皮特曼 经由对应于19世纪40年代教导速记在大不列颠。的发展邮政服务 在19世纪导致商业函授学院与全国性范围的增长。

在二十世纪，广播，电视，和 互联网 都被用来进一步距离教育。电脑和互联网取得了远程教育的分布更加方便快捷。

2006年斯隆联盟报道称，在最大的高校超过96％ 美国 提供的在线课程和近320万学生2005年秋季学期至少服用一个在线课程。

远程学习不需要上下班。这样可以节省金钱和时间斯图登T ON旅行花费，否则来回学校。学习者可以安排在他们的个人和职业生活的其他方面的学习。

学习者可以在他们方便的完成大部分课程。大多数类是异步的，这意味着精简不必参加在特定添了演讲e和地点。该任务的审查和做功课可以在非工作时间或在家完成。

同时追求选择的教育从任何地方的任何地方生活，学习。学习者不必住在同一城市或同一国家参加财的学府CE。学习者可以学习的地方，他们有机会获得一台电脑和互联网连接。

获得额外的知识。学习者可以将电脑和互联网技能的Ÿ将获得你的远程学习体验到生活的其他方面的进程。

自学。缓慢和快速的学习者。这减少压力和增加的满意度。

可访问性。 在线课程解决物理可访问性问题，有些人有限制采取传统的类时流动性遭遇。学习者可以利用他们的舒适 家具同时享受自由移动，并有机会深造。

昂贵和复杂的技术。尽管远程教育的机会很多，也有必然伴随成本。实时视频通信，例如，需要的设备和设施的精心策划。对于网上学习，学习者必须拥有一台电脑（可能与接入互联网），或有一个访问。因此，所需的技术并不总是可用的。有些学生也可能是怕技术。

提前规划。无论是教师和学生参与远程学习机唉需要在次做出牺牲，把事情的时间内完成。

隐性成本。例如，他们在船上，怎么办他们得到所需材料？他们可能需要提前招致内线被邮寄RA运输和处理成本。

远程学习不提供即时反馈。在传统的课堂设置，随着距离学生的成绩可以通过提问和非正式的测试来评估马上。学习，学生必须等待，直到反馈导师审查他们的WORK，并回答了它。

远程学习并不总是在线提供所有必需的课程。攻读具体证书或学位课程可能不具有所有可用通过远程学习必要的课程，所以它不是适合所有的科目。虽然学生可以完全在线学习历史教训，但不能执行临床护理在线。因此物理课堂出勤率会强制完成课程。

远程学习可能不会被所有雇主承认。虽然大多数雇主承认做远程教育，某些雇主没有。谁愿意为毕业后的特定的雇主工作的学生必须确保雇主的有关网络教育的视角。

远程学习不给学生有机会在口头沟通技巧的工作。远程学习课程的学生没有得到与教授和其他学生口头互动的做法。

社会孤立。最常见的学习者将独自学习。远程学习者S可感到被孤立或错过附带参加传统的课堂教学，社会物理相互作用。然而，这客观已经减轻与进步，并使用通信技术如公告栏，主题讨论，聊天，电子邮件和会议的。

该系统的软件设计的重点是基于Java的客户端/服务器类型的软件模型。该系统是使用Java / J2EE技术实现的。它包括Windows XP作为操作系统，MySQL作为后端。

在Java / J2EE： 爪哇（1.4）是一种面向对象的编程语言（OOPS）。在Java应用程序完全可以建成，担纲加速3D图形和其他多媒体功能强大的加密和网络连接。在网络上，Java可以在客户端可以用来创建applet和服务器端创建使用servlet和Java Server Pages（JSP）的动态网页。

BEA的Weblogic服务器：它是Java应用程序服务器。它是一种重量轻developm耳鼻喉科服务器。快速开发/部署/调试周期短的循环时间。

MYSQL： 数据库设计和实现使用MySQL 4.1版。MySQL是免费的 和可扩展性。

Struts框架（MVC模型2）：模型 - 视图 - 控制器是一种设计模式。MVC减弱了业务逻辑和数据接口。Struts是一个使用Servlet的2.2和JSP 1.1标签，从J2EE规范，作为执行工作的一部分的MVC实现。Struts的帮助Web项目的控制变化和促进专业化。Struts是一组协作类，Servlet和JSP标签组成一个可重用的MVC 2设计。这个定义意味着，Struts是一个框架，而不是一个库，但Struts的还包含框架的独立工作了广泛的标记库和实用程序类。

客户端浏览器：从客户端浏览器的HTTP请求创建一个事件。Web容器将重新用HTTP响应spond。

调节器：控制器接收来自浏览器的请求，并且使得该决定向何处发送该请求。用struts，控制器被作为一个Servlet实现的命令的设计模式。在struts- config.xml文件配置控制器。

商业逻辑：业务逻辑更新模型的状态，并有助于控制的流 应用。用struts，这是通过一个Action类完成的瘦包装到实际的业务逻辑。

模型状态：该模型代表了应用程序的状态。业务对象更新AP折叠状态。ActionForm Bean的代表模型的状态在会话或请求级别，而不是在一个持续的水平。JSP文件中读取使用JSP标签的ActionForm Bean的信息。

视图：视图是一个简单的JSP文件。没有流程逻辑，没有业务逻辑，也没有模式l信息 只是标签。标签是相对于如Velocity其他框架的Struts品牌独特。

附录C 软件使用说明书

用户在此页面进行登录注册，自己是一般用户。如图7-1所示。



图7-1登录页面图

只有注册有账号的用户才能在留言板处留言，如图7-2所示。

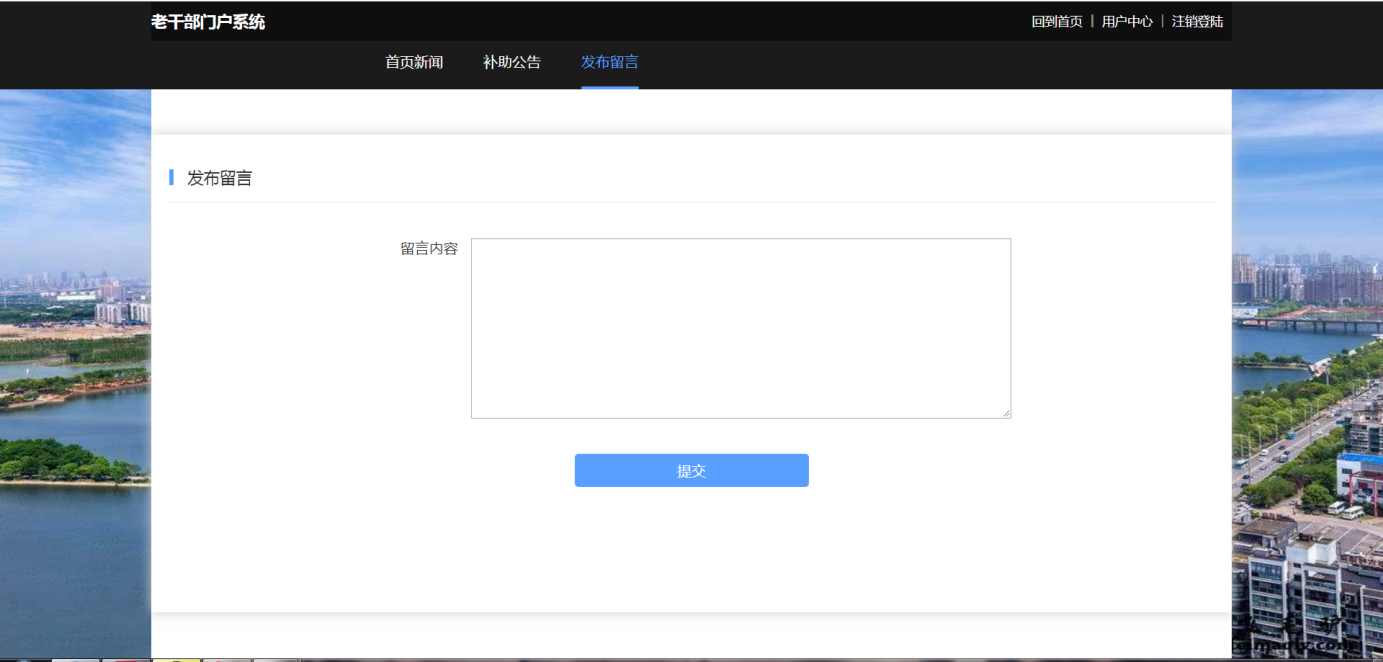


图7-2留言板页面图

管理员在添加新闻的时候要注意标题内容不能为空。如图7-3所示。

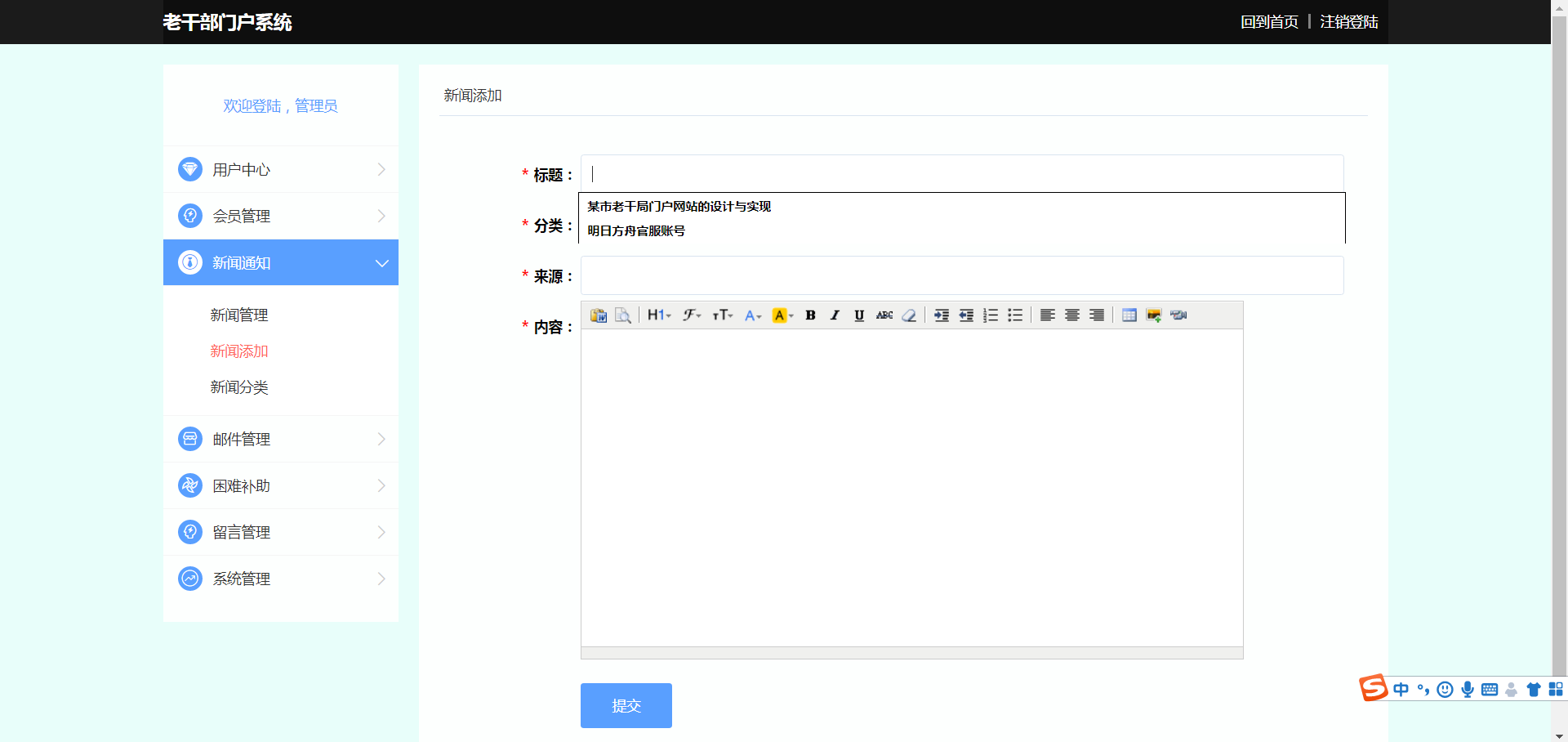


图7-3新闻添加页面

管理员在添加补助时也要符合补助添加规定，即标题内容不为空，如图7-4所示。

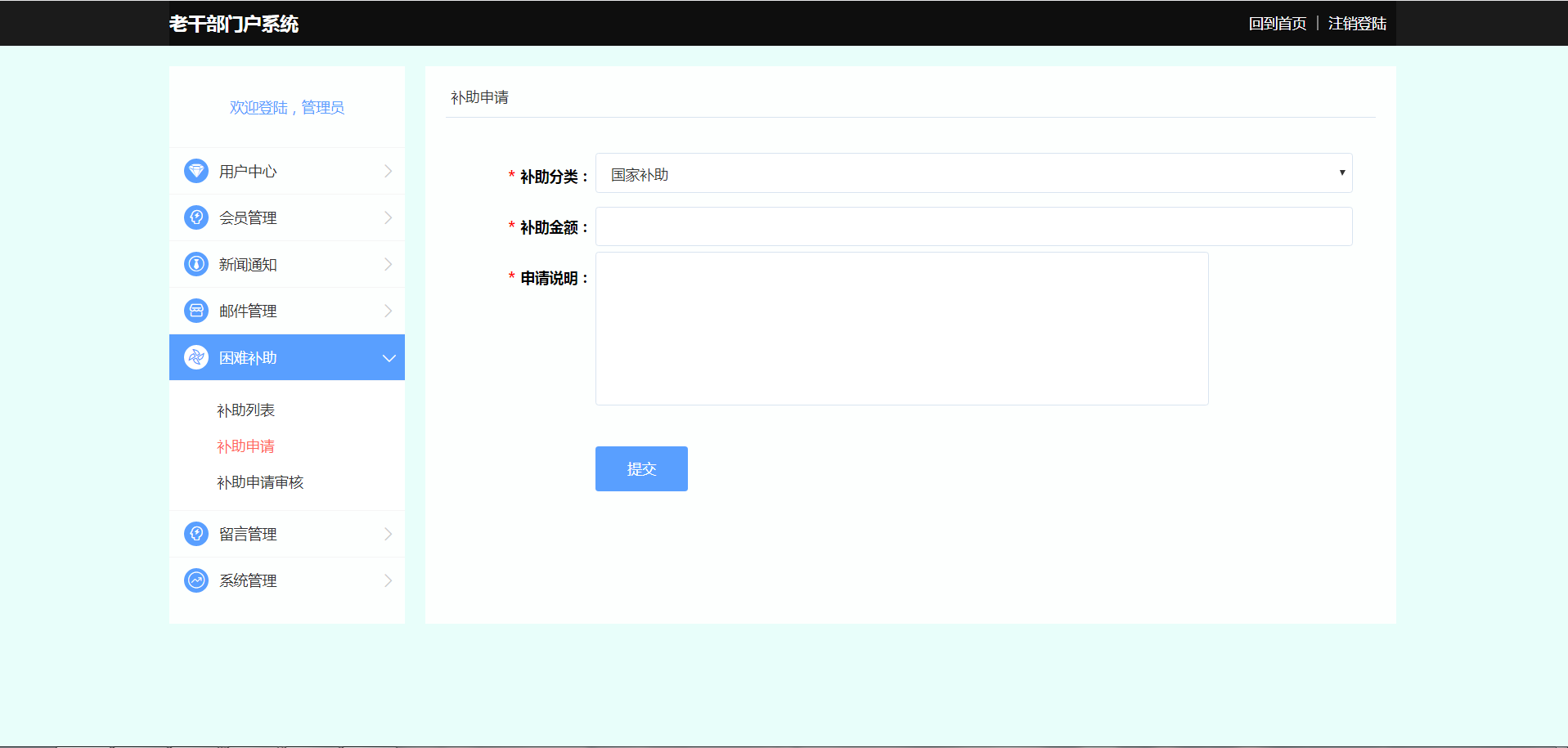


图7-4补助申请页面

发送邮件的时候要注意邮件的标题内容不能为空，且用户id正确，如图7-5所示。

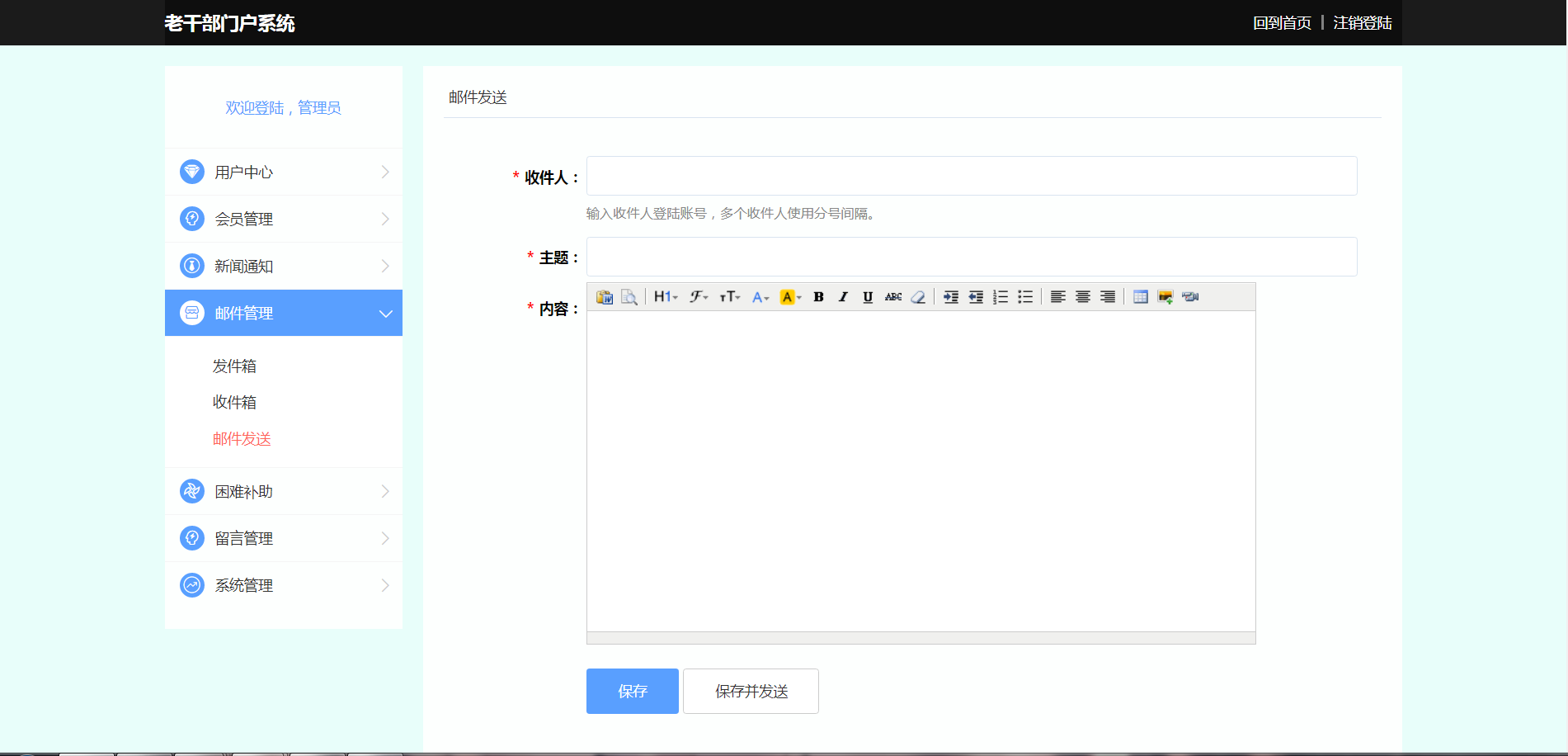


图7-5邮件发送页面

附录D 主要源代码

/\*\*

\* 公告管理

\* @author Oven

\* @date 2020/3/12 22:03

\*/

@Controller

@RequestMapping("portal/news")

public class NewsController {

private static final Logger logger = LoggerFactory.getLogger(NewsController.class);

@Autowired

private NewsService newsService;

@Autowired

private CategoryService categoryService;

@RequestMapping(value = {"", "list"})

public String main(Integer pageNo, HttpServletRequest request) {

pageNo = ValidateHelper.isNullOrEmpty(pageNo) ? 1: pageNo;

Page page = new Page(pageNo, 20);

Map<String, String> params = new HashMap<>();

List<News> list = newsService.queryPage(params, page);

request.setAttribute("pageNo", pageNo);

request.setAttribute("totalPages", page.getPageCount());

request.setAttribute("list", list);

return "news/list";

}

@RequestMapping("data")

public String data(Integer pageNo, HttpServletRequest request) {

pageNo = ValidateHelper.isNullOrEmpty(pageNo) ? 1: pageNo;

Page page = new Page(pageNo, 20);

Map<String, String> params = new HashMap<>();

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

params.put("userId", user.getId());

List<News> list = newsService.queryPage(params, page);

request.setAttribute("pageNo", pageNo);

request.setAttribute("totalPages", page.getPageCount());

request.setAttribute("list", list);

return "news/data";

}

@RequestMapping("add")

public String add(Integer pageNo, HttpServletRequest request) {

request.setAttribute("categoryList", categoryService.queryList(Collections.emptyMap()));

return "news/add";

}

/\*\*

\* 新增新闻

\*/

@RequestMapping(value = "insert", method = RequestMethod.POST)

@ResponseBody

public Response insert(News news, HttpServletRequest request) {

logger.debug("新增保存新闻");

Response response = new Response();

try {

AssertUtil.isNotNull(request, "title", "标题不能为空");

AssertUtil.isNotNull(request, "source", "来源不能为空");

AssertUtil.isNotNull(request, "content", "内容不能为空");

boolean b = newsService.insert(news);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 进入编辑页面

\*/

@RequestMapping(value = "edit/{id}", method = RequestMethod.GET)

public String edit(@PathVariable("id") String id, HttpServletRequest request) {

request.setAttribute("news", newsService.getById(id));

request.setAttribute("categoryList", categoryService.queryList(Collections.emptyMap()));

return "news/edit";

}

/\*\*

\* 修改新闻

\*/

@RequestMapping(value = "update", method = RequestMethod.POST)

@ResponseBody

public Response update(News news, HttpServletRequest request) {

logger.debug("修改保存新闻");

Response response = new Response();

try {

AssertUtil.isNotNull(request, "title", "标题不能为空");

AssertUtil.isNotNull(request, "source", "来源不能为空");

AssertUtil.isNotNull(request, "content", "内容不能为空");

boolean b = newsService.update(news);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 删除

\*/

@RequestMapping(value = "delete/{id}", method = RequestMethod.POST)

@ResponseBody

public Response delete(@PathVariable("id") String id, HttpServletRequest request) {

logger.debug("删除新闻");

Response response = new Response();

try {

boolean b = newsService.delete(id);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 标记已读

\*/

@RequestMapping(value = "updateReaded", method = RequestMethod.POST)

@ResponseBody

public Response updateReaded(String newsId, HttpServletRequest request) {

logger.debug("标记已读新闻，新闻ID为[{}]", newsId);

Response response = new Response();

try {

AssertUtil.isNotNull(request, "newsId", "新闻ID不能为空");

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

boolean b = newsService.updateReaded(user.getId(), newsId);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 进入详情页面

\*/

@RequestMapping(value = "detail/{id}", method = RequestMethod.GET)

public String detail(@PathVariable("id") String id, HttpServletRequest request) {

request.setAttribute("news", newsService.getById(id));

try {

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

newsService.updateReaded(user.getId(), id);

} catch (Exception ignored) {

}

return "news/detail";

}

}

/\*\*

\* 邮件管理

\* @author Oven

\* @date 2020/3/12 22:03

\*/

@Controller

@RequestMapping("portal/mail")

public class MailController {

private static final Logger logger = LoggerFactory.getLogger(MailController.class);

@Autowired

private MailService mailService;

@RequestMapping(value = {"", "list"})

public String main(Integer pageNo, HttpServletRequest request) {

pageNo = ValidateHelper.isNullOrEmpty(pageNo) ? 1: pageNo;

Page page = new Page(pageNo, 20);

Map<String, String> params = new HashMap<>();

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

params.put("userId", user.getId());

List<Mail> list = mailService.queryPage(params, page);

request.setAttribute("pageNo", pageNo);

request.setAttribute("totalPages", page.getPageCount());

request.setAttribute("list", list);

return "mail/list";

}

@RequestMapping("data")

public String data(Integer pageNo, HttpServletRequest request) {

pageNo = ValidateHelper.isNullOrEmpty(pageNo) ? 1: pageNo;

Page page = new Page(pageNo, 20);

Map<String, String> params = new HashMap<>();

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

params.put("targetId", user.getId());

List<Mail> list = mailService.queryPage(params, page);

request.setAttribute("pageNo", pageNo);

request.setAttribute("totalPages", page.getPageCount());

request.setAttribute("list", list);

return "mail/data";

}

@RequestMapping("add")

public String add(Integer pageNo, HttpServletRequest request) {

return "mail/add";

}

/\*\*

\* 新增邮件

\*/

@RequestMapping(value = "insert", method = RequestMethod.POST)

@ResponseBody

public Response insert(Mail mail, HttpServletRequest request) {

logger.debug("新增保存邮件");

Response response = new Response();

try {

AssertUtil.isNotNull(request, "target", "收件人不能为空");

AssertUtil.isNotNull(request, "title", "标题不能为空");

AssertUtil.isNotNull(request, "content", "内容不能为空");

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

if (checkContains(mail.getTarget(), user.getLoginId())) {

throw new BusinessException("不能给自己发邮件");

}

mail.setUserId(user.getId());

mail.setState(0);

boolean b = mailService.insert(mail);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 新增并发送邮件

\*/

@RequestMapping(value = "insertAndSend", method = RequestMethod.POST)

@ResponseBody

public Response insertAndSend(Mail mail, HttpServletRequest request) {

logger.debug("新增保存并发送邮件");

Response response = new Response();

try {

AssertUtil.isNotNull(request, "target", "收件人不能为空");

AssertUtil.isNotNull(request, "title", "标题不能为空");

AssertUtil.isNotNull(request, "content", "内容不能为空");

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

if (checkContains(mail.getTarget(), user.getLoginId())) {

throw new BusinessException("不能给自己发邮件");

}

mail.setUserId(user.getId());

mail.setState(0);

boolean b = mailService.insertAndSend(mail);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

private boolean checkContains(String target, String loginId) {

String[] userArr = target.replace("；", ";").split(";");

return Arrays.asList(userArr).contains(loginId);

}

/\*\*

\* 新增邮件

\*/

@RequestMapping(value = "send", method = RequestMethod.POST)

@ResponseBody

public Response send(String id, HttpServletRequest request) {

logger.debug("发送邮件，ID为{}", id);

Response response = new Response();

try {

AssertUtil.isNotNull(request, "id", "邮件ID不能为空");

boolean b = mailService.updateSended(id);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 进入编辑页面

\*/

@RequestMapping(value = "edit/{id}", method = RequestMethod.GET)

public String edit(@PathVariable("id") String id, HttpServletRequest request) {

request.setAttribute("mail", mailService.getById(id));

return "mail/edit";

}

/\*\*

\* 修改邮件

\*/

@RequestMapping(value = "update", method = RequestMethod.POST)

@ResponseBody

public Response update(Mail mail, HttpServletRequest request) {

logger.debug("修改保存邮件");

Response response = new Response();

try {

AssertUtil.isNotNull(request, "id", "邮件ID不能为空");

AssertUtil.isNotNull(request, "target", "收件人不能为空");

AssertUtil.isNotNull(request, "title", "标题不能为空");

AssertUtil.isNotNull(request, "content", "内容不能为空");

boolean b = mailService.update(mail);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 修改并发送邮件

\*/

@RequestMapping(value = "updateAndSend", method = RequestMethod.POST)

@ResponseBody

public Response updateAndSend(Mail mail, HttpServletRequest request) {

logger.debug("修改保存并发送邮件");

Response response = new Response();

try {

AssertUtil.isNotNull(request, "id", "邮件ID不能为空");

AssertUtil.isNotNull(request, "target", "收件人不能为空");

AssertUtil.isNotNull(request, "title", "标题不能为空");

AssertUtil.isNotNull(request, "content", "内容不能为空");

boolean b = mailService.updateAndSend(mail);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 删除

\*/

@RequestMapping(value = "delete/{id}", method = RequestMethod.POST)

@ResponseBody

public Response delete(@PathVariable("id") String id, HttpServletRequest request) {

logger.debug("删除邮件");

Response response = new Response();

try {

AssertUtil.isNotNull(request, "id", "邮件ID不能为空");

boolean b = mailService.delete(id);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 标记已读

\*/

@RequestMapping(value = "updateReaded", method = RequestMethod.POST)

@ResponseBody

public Response updateReaded(String mailId, HttpServletRequest request) {

logger.debug("标记已读邮件，邮件ID为[{}]", mailId);

Response response = new Response();

try {

AssertUtil.isNotNull(request, "mailId", "邮件ID不能为空");

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

boolean b = mailService.updateReaded(user.getId(), mailId);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

/\*\*

\* 移除收件箱

\*/

@RequestMapping(value = "deleteMail", method = RequestMethod.POST)

@ResponseBody

public Response deleteMail(String mailId, HttpServletRequest request) {

logger.debug("删除收件箱邮件，邮件ID为[{}]", mailId);

Response response = new Response();

try {

AssertUtil.isNotNull(request, "mailId", "邮件ID不能为空");

User user = (User) request.getSession().getAttribute(SystemConstant.USER);

boolean b = mailService.deleteMail(user.getId(), mailId);

if(b){

response.setSuccess();

}

} catch (BusinessException e) {

logger.debug(e.getMessage(), e);

response.setError(e.getMessage());

} catch (Exception e) {

logger.error(e.getMessage(), e);

response.setError(e.getMessage());

}

return response;

}

}