**目 录**

[**1 绪 论** 1](#_Toc40358947)

[1.1 研究的背景及意义 1](#_Toc40358948)

[1.1.1 选题的背景 1](#_Toc40358949)

[1.1.2 国内外研究现状 1](#_Toc40358950)

[1.1.3 研究的意义 1](#_Toc40358951)

[1.2 系统目标 2](#_Toc40358952)

[**2 需求分析** 3](#_Toc40358953)

[2.1 业务需求 3](#_Toc40358954)

[2.2 功能需求 5](#_Toc40358955)

[2.2.1 角色分析 5](#_Toc40358956)

[2.2.2 业务功能 6](#_Toc40358957)

[2.3 非功能需求 14](#_Toc40358958)

[2.3.1环境需求 14](#_Toc40358959)

[2.3.2 性能需求 15](#_Toc40358960)

[2.3.3 安全需求 15](#_Toc40358961)

[**3 总体设计** 16](#_Toc40358962)

[3.1系统设计的原则 16](#_Toc40358963)

[3.2 系统体系结构设计 16](#_Toc40358964)

[3.3 系统功能结构设计 17](#_Toc40358965)

[3.3.1 综合应用支撑平台子系统功能结构设计 18](#_Toc40358966)

[3.3.2个人事务管理子系统功能结构设计 19](#_Toc40358967)

[**4 数据库设计** 20](#_Toc40358968)

[4.1 概念结构设计 20](#_Toc40358969)

[4.1.1 设计思路 20](#_Toc40358970)

[4.1.2 系统总体E-R图 20](#_Toc40358971)

[4.2 逻辑结构设计 20](#_Toc40358972)

[4.2.1 设计思路 20](#_Toc40358973)

[4.2.2 逻辑模型 21](#_Toc40358974)

[4.3 物理结构设计 22](#_Toc40358975)

[4.3.1 存取方式 22](#_Toc40358976)

[4.3.2 存储结构 22](#_Toc40358977)

[**5 界面设计** 23](#_Toc40358978)

[5.1 界面关系图或工作流图 23](#_Toc40358979)

[5.2 界面设计成果 24](#_Toc40358980)

[5.2.1 主界面 24](#_Toc40358981)

[5.2.2 子界面 24](#_Toc40358982)

[**6 详细设计** 27](#_Toc40358983)

[6.1 系统主要功能模块介绍 27](#_Toc40358984)

[6.2 统一身份认证和授权管理功能 27](#_Toc40358985)

[6.2.1功能结构设计 27](#_Toc40358986)

[6.2.2类图设计 28](#_Toc40358987)

[6.2.3顺序图设计 30](#_Toc40358988)

[6.2.4核心处理流程设计 31](#_Toc40358989)

[6.3工作流管理功能 32](#_Toc40358990)

[6.3.1功能结构设计 32](#_Toc40358991)

[6.3.2类图设计 32](#_Toc40358992)

[6.3.3顺序图设计 33](#_Toc40358993)

[6.3.4核心处理流程设计 33](#_Toc40358994)

[6.4电子表单管理功能 34](#_Toc40358995)

[6.4.1功能结构设计 34](#_Toc40358996)

[6.4.2类图设计 34](#_Toc40358997)

[6.4.3顺序图设计 35](#_Toc40358998)

[6.4.4核心处理流程设计 36](#_Toc40358999)

[6.5个人设置功能 36](#_Toc40359000)

[6.5.1功能结构设计 36](#_Toc40359001)

[6.5.2类图设计 37](#_Toc40359002)

[6.5.3顺序图设计 38](#_Toc40359003)

[6.5.4核心处理流程设计 39](#_Toc40359004)

[6.6通讯录管理功能 39](#_Toc40359005)

[6.6.1功能结构设计 39](#_Toc40359006)

[6.6.2类图设计 40](#_Toc40359007)

[6.6.3顺序图设计 41](#_Toc40359008)

[6.6.4核心处理流程设计 42](#_Toc40359009)

[6.7内部邮件管理功能 42](#_Toc40359010)

[6.7.1功能结构设计 42](#_Toc40359011)

[6.7.2类图设计 43](#_Toc40359012)

[6.7.3顺序图设计 44](#_Toc40359013)

[6.7.4核心处理流程设计 45](#_Toc40359014)

[**7 编码** 46](#_Toc40359015)

[7.1 代码实现与核心算法 46](#_Toc40359016)

[7.1.1 用户管理核心代码 46](#_Toc40359017)

[7.1.2 邮件管理核心代码 50](#_Toc40359018)

[7.1.3 通讯录管理核心代码 54](#_Toc40359019)

[7.1.4工作流管理核心代码 55](#_Toc40359020)

[7.2 代码优化分析 56](#_Toc40359021)

[**8 测试** 57](#_Toc40359022)

[8.1 测试方案设计 57](#_Toc40359023)

[8.1.1 测试策略 57](#_Toc40359024)

[8.1.2 测试进度安排 58](#_Toc40359025)

[8.1.3 测试资源 58](#_Toc40359026)

[8.1.4 关键测试点 58](#_Toc40359027)

[8.2 测试用例构建 59](#_Toc40359028)

[8.2.1 测试用例编写约定 59](#_Toc40359029)

[8.2.2 测试用例设计 59](#_Toc40359030)

[8.2.3 测试用例维护 62](#_Toc40359031)

[**9 总结与展望** 63](#_Toc40359032)

[9.1 设计工作总结 63](#_Toc40359033)

[9.2 未来工作展望 63](#_Toc40359034)

[**谢 辞 64**](#_Toc40359035)

[**参考文献 65**](#_Toc40359036)

[**附录A 外文翻译—原文部分 67**](#_Toc40359037)

[**附录B 外文翻译—译文部分 73**](#_Toc40359038)

[**附录C 主要源代码 78**](#_Toc40359039)

1 绪 论

## 1.1 研究的背景及意义

### 1.1.1 选题的背景

在社会高度发展的当下，有越来越多的人投入到创业的大潮中，这也使得有越来越多的新型企业开始在全国各地落地生根。与此同时，国家也顺应这种潮流，在各地设立高新开发区，为各种新企业提供政策和财政上的扶持。世界总是瞬息万变的，在经济上更是如此，如果还是按照往常的处理模式进行这类政务的处理，将会大大耽误企业的工作开展，同时也会大大增加政府人员的无关工作量。所以，一个拥有良好运行效率，能够让这类工作能够在网络上进行快速运转并实时进行操作的系统是所有人都迫切需要的。高新区企业综合应用支撑平台及个人事务管理系统能够很好的解决上述的问题。

### 1.1.2 国内外研究现状

改革开放以来，国家打出了以经济建设为中心的口号，为了进一步的促进各地方的发展，国家在许多地方设定了高新技术产业开发区。各个地方的高新技术产业区是为了促进高新技术的进一步发展，并为地方吸引到更多的高新企业和人才来到本地进行投资与就业，使得能够进一步促进经济的发展。而要达到这些目的，就要对企业制定一系列的扶持优惠政策。而在对企业实行政策的时候，过去受限于技术手段，大部分都是人工线下处理，这就造成了企业的办事不便以及手续复杂，又因纸质文件的传递流转不快，通常的事务处理效率常常不能令人满意。而随着时代与技术的发展，互联网技术开始发展并进入千家万户，将整个办公流程进行网络化也被搬上了日程。网络技术经过几十年的发展，已经可以初步完整的实现整个办公系统的在线化，人们已经能够利用现如今的网络技术与架构，实现整个办公流程通过网络进行流转。

本系统的功能属于办公自动化的一部分。办公自动化出现于上个世纪七十年代，得益于当时所爆发的信息革命，西方发达国家为了解决办公之中繁杂的问题，发展出了以个人计算机和办公软件为中心的第一代办公自动化。在经过数十年的发展之后，国外的办公自动化行业已经发展的比较成熟，形成了一整套完善的软硬件体系。而在国内，对办公自动化的研究起步于上世纪八十年代，并且受到外国相关行业的影响，也有越来越多的企业单位注意到了对办公自动化的需求，越来越多的人参与进办公自动化的发展之中。但与外国相比，国内的行业还是有着一定的差距，比如说技术含量相对而言比较不足，相关软件的规模也相差甚远等等。这些问题一方面是因为相关的技术积累不足，另一方面也是因为相较国外而言国内的企业还有很多并没有办公自动化的意识，，这都限制着办公自动化行业在国内的发展。

### 1.1.3 研究的意义

随着时代和技术的进一步发展，企业综合应用支撑平台以及个人事务管理系统将会得到进一步的发展，越来越多不断涌现的新技术也会使得这个系统更加的先进与完善。例如，可以将这个系统与当下火热的人工智能技术相结合，使得系统可以简单的处理一些事务，对不太复杂的事务进行自动的批复，更进一步减轻员工的工作量。又或者，将系统与大数据技术相结合，对所有企业提交的信息进行汇总分析，以帮助领导层更好的做出决策。随着技术的进步，企业个人事务管理系统也会越来越完善，这将大大方便企业的办公，大大推进这个管理系统的推广安装，有助于社会的进步，使中国在企业管理方面更加进步完善。

## 1.2 系统目标

本系统中，主要有综合应用支撑平台以及个人事务管理系统两个子系统。其中，综合应用支撑平台的功能主要包括统一身份认证和授权管理，工作流，电子表单三个模块。统一身份认证和授权管理功能模块会对所有登录系统的人员进行相应的登录认证。同时，可以供系统管理员使用，管理员可以使用此模块的功能对所有的系统用户的信息进行管理。此外，系统管理员还可以利用此模块对所有用户进行权限的分配与管理。而工作流和电子表单这两个功能模块，主要提供给系统的用户使用，在用户登录系统后，所有的事务都会以相应的条目呈现在用户的相应界面上，通过点击相应的条目，就会跳转进入相应的工作流或者是电子表单中，用户可以清楚的了解到每一项事务的所有流程步骤，同时可以利用提供的图形化表单编辑工具以及图形化工作流编辑工具，对事务进行相应的处理。

个人事务管理子系统主要包括个人设置，通讯录管理，内部邮件管理，三个功能模块。个人事务管理系统主要面向系统的用户，用户登录系统后，可以根据账户的权限选择相应的功能进行使用。个人设置模块，提供个人的基础设置功能，包括修改密码，个人信息设定等，用户可以使用此模块对密码进行修改，以及对个人的详细信息，例如姓名，年龄，联系方式等等信息进行设置。通讯录管理模块，可以展示用户对应账号下所存储的个人通讯录以及公共通讯录。内部邮件管理模块，可以帮助用户处理系统内部的邮件信息，包括收发邮件，或者是将账户中的邮件导出至本机硬盘进行存储。

2 需求分析

## 2.1 业务需求

（1）能够实现用户管理、身份认证、授权管理等功能。其中，用户管理提供用户数据增加、修改、删除等功能；身份认证模块提供登录验证、权限验证等功能；授权管理应提供根据组、角色的定义和授权。

（2）实现对用户的所有事项进行管理，已电子表单以及工作流的方式进行呈现，方便用户进行处理以及进行相应的校对和查阅。

（3）实现用户对自己的基础信息进行管理，包括修改密码，个人信息的设定。

（4）用户可以查看和编辑自己的个人通讯录和公共通讯录。

（5）用户可以对自己账号下的收件箱与发件箱进行相应的增删改查管理，并能将相应邮件导出至本地。

主要业务流程如下：

（1）登陆注册活动图展示了用户在进行登录和注册的过程中的活动流程，如图2-1：



图2-1 登录注册活动图

（2）事务处理活动图展示了用户在进行事务处理时的详细活动，如图2-2。在进入系统之后，用户可以选择两个不同的子模块，分别为工作流和电子表单，并对其进行相应的操作。



图2-2 事务处理活动图

（3）个人信息设置活动图展示了用户在进行个人信息设置的详细活动，如图2-3，用户在进入个人信息设置模块之后，可以选择不同的操作。

2-3 个人信息设置活动图

（4）通讯录管理活动图展示了用户在进行通讯录的相关管理的详细活动，如图2-4，

用户可以在登录系统之后，选择通信录管理，并进行不同的操作。



图2-4 通讯录管理活动图

（5）邮件管理活动图展示了用户在进行干邮件管理的过程中的详细活动，如图2-5，用户可以在登录系统之后，进入邮件管理，并进行不同的操作，例如收发邮件等等。

图2-5 邮件管理活动图

## 2.2 功能需求

### 2.2.1 角色分析

从系统的实际功能出发，得到了系列的系统角色，如表2-1：

表2-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 用户 | 使用系统处理事务 |
| 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

系统总共包括两个子系统，分别是个人事务管理子系统以及综合应用支撑平台子系统，系统总用例图如图2-6：



图2-6 系统总体用例图

（1）个人事务管理子系统

个人事务管理子系统包含三个功能模块，分别是通讯录管理，个人设置，以及邮件管理。子系统用例图如图2-7：

**

图2-7 个人事务管理用例图

1）用户登录用户描述详细描述了用户登录系统的过程，如表2-2：

表2-2 用户登录用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 登录 |
| 用例描述 | 登录是系统进行身份验证的功能模块，用户在提供有效的账号和密码后，发送系统后台进行身份验证，返回验证后的信息，根据返回信息判断身份验证是否通过。 | | |
| 执行者 | 系统用户 | | |
| 前置条件 | 用户打开登录界面且处于未登录状态 | | |
| 后置条件 | 登录成功，进入系统界面 | | |
| 涉众利益 | 1.使用者担心密码泄露  2.用户希望登录成功后能够记住密码 | | |
| 用例场景 | 系统登录界面 | | |
| 基本事件流 | **基本流程（Base Flow）**  1. 填写账号、密码  2. 系统验证账号、密码是否填写完整  3. 如果验证不完整，提示用户按要求填写必要的信息  4. 将有效的用户账号、密码提交到服务器验证  5. 系统根据服务器验证结果进行相应操作  6. 如果是账号或密码操作，直接提示账号或密码有误，不可以直接提示账号错误或密码错误  7. 如果验证通过，直接登录到系统主界面。  **扩展流程（Extend Flow）**  1a. 登录信息不充分：    1a1. 系统提示登录信息不充分  1a2. 用例结束  **字段列表（Filed List）**  1. 用户登录信息＝账号＋密码  **业务规则（Business role）**  1.账号不能为空，最长不能超过20个字符，必须是由字母或数字或下划线组成；  2.密码不能为空，最长不能超过20个字符；  3.密码必须是加密； | | |
| 特殊需求（Special Requirement） | 1.系统应能对用户的密码强弱进行一定的判断句，如果密码强度过低，建议用户修改密码。  2.系统反应快速，\*1-7应该在十秒内进行响应 | | |
| 编写人 |  | | |

2）邮件管理用例描述详细描述了用户对邮件进行管理的过程，如表2-3：

表2-3 邮件管理用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0002 | 用例名称 | 收发邮件 |
| 用例描述 | 收发邮件在用户成功登录系统后，可以选择邮件管理功能模块，查看该账号下的收到的邮件，并可以新建新的邮件进行发送。 | | |
| 执行者 | 系统用户 | | |
| 前置条件 | 用户成功登录系统 | | |
| 后置条件 | 查看邮件或发送邮件 | | |
| 涉众利益 | 1.邮件能否成功送达  2.是否会漏接邮件 | | |
| 用例场景 | 邮件管理系统 | | |
| 基本事件流 | **基本流程（Base Flow）**   1. 选择进入邮件模块   2. 查看收件箱中已收到的邮件  3. 对已收到的邮件进行直接回复  4. 查看发件箱中已发送的邮件  5. 新建新邮件进行编写与发送  6. 返回邮件发送成功的结果  7. 操作完成，返回系统主界面  **扩展流程（Extend Flow）**  1a. 接收邮箱名错误：    1a1. 系统返回发送失败的结果，提升接受邮箱名错误  1a2. 用例结束  **业务规则（Business role）**  1.收件邮箱名不能为空，必须是由字母或数字或下划线组成，最长不能超过20个字符； | | |
| 特殊需求（Special Requirement） | 1.系统能快速回应发送的结果，发送结果应在五秒内返回。  2.系统应反应快速，\*1-7应该在十秒内进行响应 | | |
| 编写人 |  | | |

3）个人设置用例描述详细描述了用户修改密码的过程，如表2-4：

表2-4 个人设置用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0004 | 用例名称 | 修改密码 |
| 用例描述 | 在登录系统后，用户可以选择个人设置功能模块，对自己的登录密码进行修改。 | | |
| 执行者 | 系统用户 | | |
| 前置条件 | 用户成功登录系统 | | |
| 后置条件 | 登录密码修改成功 | | |

续表2-4 个人设置用例描述

|  |  |
| --- | --- |
| 涉众利益 | 1.使用者担心新密码是否能成功应用  2.旧密码是否成功作废 |
| 用例场景 | 个人设置模块 |
| 基本事件流 | **基本流程（Base Flow）**  1. 进入个人设置模块  2. 选择修改密码，输入旧密码进行验证  3. 将用户密码提交到服务器验证  4. 如果密码有误，提示密码错误  5. 如果验证通过，要求用户设置新密码  6. 验证用户提交的新密码，验证通过后提示用户修改成功  **扩展流程（Extend Flow）**  1a. 新密码验证不通过：    1a1. 系统提示密码验证不通过，请用户进行修改  1a2. 用例结束  **业务规则（Business role）**  1.密码不能为空，最长不能超过20个字符； |
| 特殊需求（Special Requirement） | 1.系统应能对用户的密码强弱进行一定的判断，如果密码强度过低，建议用户修改密码。  2.系统应反应快速，\*1-7应该在十秒内进行响应 |
| 编写人 |  |

4）个人详细信息用例描述详细描述了用户如何修改自己信息的过程，如表2-5：

表2-5 个人详细信息设置用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0005 | 用例名称 | 详细信息设置 |
| 用例描述 | 在登录系统后，用户可以选择个人设置功能模块，对自己的个人详细信息进行设置。 | | |
| 执行者 | 系统用户 | | |
| 前置条件 | 用户成功登录系统 | | |
| 后置条件 | 个人信息设置成功 | | |
| 涉众利益 | 1.使用者担心个人信息的保存问题 | | |
| 用例场景 | 个人设置模块 | | |
| 基本事件流 | **基本流程（Base Flow）**  1. 进入个人设置模块  2. 选择个人信息设置  3. 进行个人信息界面进行相应的编辑与修改  4. 提交修改，系统返回修改成功的提示  **扩展流程（Extend Flow）**  1a. 个人信息不符合规范：    1a1. 系统提示不规范的具体信息，请用户进行修改  1a2. 用例结束  **业务规则（Business role）** | | |

续表2-5 个人详细信息设置用例描述

|  |  |
| --- | --- |
|  | 1.个人信息需在正确的区间 |
| 特殊需求（Special Requirement） | 1.系统能对用户输入的个人信息做出一定程度的判断  2.系统应对用户修改的结果进行快速响应 |
| 编写人 |  |

5）通讯录新增用例描述详细描述了用户新增通讯录的操作，如表2-6：

表2-6 通讯录新增用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0006 | 用例名称 | 通讯录新增操作 |
| 用例描述 | 在登录系统后，用户可以选择通讯录功能模块，对本账号下所对应的个人通讯录进行新增操作。 | | |
| 执行者 | 系统用户 | | |
| 前置条件 | 用户成功登录系统 | | |
| 后置条件 | 通讯录中新增用户成功 | | |
| 涉众利益 | 1.信息能否成功新增 | | |
| 用例场景 | 通讯录管理模块 | | |
| 基本事件流 | **基本流程（Base Flow）**  1. 进入通讯录管理模块  2. 选择新增联系人操作  3. 输入联系人信息  4. 点击提交进行新增  5. 系统返回新增成功提示  **扩展流程（Extend Flow）**  1a. 提交的信息不合规范：    1a1. 系统提示用户信息不合规范，请用户进行修改  1a2. 用例结束  **业务规则（Business role）**  1. 用户号码只能由十一位数字组成  2. 姓名只能有汉字或字母组成 | | |
| 特殊需求（Special Requirement） | 1.系统能对用户提交的信息进行简单的判断  2.系统反应快速 | | |
| 编写人 |  | | |

（2）综合应用支撑平台子系统包括三个功能模块，分别是流程表单管理，用户信息管理，以及事务管理。子系统用例图如图2-8：

****

图2-8 综合应用支撑平台子系统用例图

1）用户信息删除用例描述详细描述了管理员对用户信息进行删除的操作，如表2-7：

表2-7 用户信息新删除用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 007 | 用例名称 | 用户信息删除操作 |
| 用例描述 | 在登录系统后，管理员可以选择用户信息管理功能模块，对本系统下所选的用户信息进行删除操作。 | | |
| 执行者 | 系统管理员 | | |
| 前置条件 | 管理员成功登录系统 | | |
| 后置条件 | 删除用户信息成功 | | |
| 涉众利益 | 1.用户信息能否成功删除 | | |
| 用例场景 | 用户信息管理模块 | | |
| 基本事件流 | **基本流程（Base Flow）**  1. 进入用户信息管理模块  2. 选择一条或多条用户信息  3. 选择删除  4. 系统提示管理员是否确认删除，管理员进行确认  5. 系统返回删除成功提示  **扩展流程（Extend Flow）**  1a. 管理员取消删除：    1a1. 系统取消删除操作，返回用户信息界面  1a2. 用例结束 | | |

续表2-7 用户信息新删除用例描述

|  |  |
| --- | --- |
| 特殊需求 | 系统反应迅速，应在五秒内完成删除操作并返回相应的结果 |
| 编写人 |  |

2）用户权限设置用例描述详细描述了管理员对用户权限进行设置的过程，如表2-8：

表2-8 用户权限设置用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 008 | 用例名称 | 用户权限设置 |
| 用例描述 | 在登录系统后，系统管理员可以选择系统中的用户信息管理模块，为用户设置相应的权限。 | | |
| 执行者 | 系统管理员 | | |
| 前置条件 | 管理员成功登录系统 | | |
| 后置条件 | 用户权限设置成功 | | |
| 涉众利益 | 1. 用户的权限是否能成功生效  2. 设置的权限是否与对应的用户相符 | | |
| 用例场景 | 用户信息管理模块 | | |
| 基本事件流 | **基本流程（Base Flow）**  1. 进入用户信息管理模块  2. 选择一名用户  3. 设置相应的权限  4. 系统返回设置成功的提示  **扩展流程（Extend Flow）**  1a. 设置的权限与用户信息冲突：    1a1. 系统提示权限非法，请管理员修改  1a2. 用例结束  **业务规则（Business role）**  1. 用户权限需与用户对应的角色相符 | | |
| 特殊需求（Special Requirement） | 1.系统能对设置的权限进行简单的判断  2.系统反应快速 | | |

|  |  |
| --- | --- |
| 编写人 |  |

3）事务处理用例描述详细描述了用户进行事务处理的过程，如表2-9：

表2-9 事务处理用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 009 | 用例名称 | 处理事务 |
| 用例描述 | 在登录系统后，用户可以选择系统中的事务管理模块，进行事务管理。 | | |
| 执行者 | 用户 | | |
| 前置条件 | 用户成功登录系统 | | |
| 后置条件 | 用户处理完成事务 | | |
| 涉众利益 | 1. 是否完整列出所有事项 2. 事务是否可以方便的进行处理 | | |

续表2-9 事务处理用例描述

|  |  |
| --- | --- |
| 用例场景 | 事务管理模块 |
| 基本事件流 | **基本流程（Base Flow）**  1. 用户登录系统  2. 系统展示所有事务  3. 用户选择相应的事务  4. 系统以工作流的方式详细展示事务  5. 用户利用电子表单功能对事务进行处理  6. 事务处理完成，返回系统界面  **扩展流程（Extend Flow）**  1a. 事务处理过程中发生错误    1a1. 系统给出相应的错误提示  1a2. 用例结束 |
| 特殊需求（Special Requirement） | 1. 系统能对所有信息的流转以工作流的方式呈现   2. 系统反应快速 |

|  |  |
| --- | --- |
| 编写人 |  |

4）工作流设置用例描述详细描述了管理员对工作流进行设置的过程，如表2-10：

表2-10 工作流设置用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0010 | 用例名称 | 工作流设置 |
| 用例描述 | 在登录系统后，系统管理员可以选择系统中的流程表单管理模块，设置相应的工作流。 | | |
| 执行者 | 系统管理员 | | |
| 前置条件 | 管理员成功登录系统 | | |
| 后置条件 | 工作流设置成功 | | |
| 涉众利益 | 1. 工作流是否符合工作流程 2. 工作流是否能正确反映信息的流转 | | |
| 用例场景 | 流程表单管理模块 | | |
| 基本事件流 | **基本流程（Base Flow）**  1. 管理员选择流程表单管理模块  2. 选择工作流设置  3. 设置工作流的类型等  4. 系统返回设置结果  **扩展流程（Extend Flow）**  1a. 类型设置不符：    1a1. 系统提示类型不符，请管理员修改  1a2. 用例结束 | | |
| 特殊需求（Special Requirement） | 1. 设置的工作流应当尽可能契合工作流程  2. 工作流应当能进行实时更新 | | |

|  |  |
| --- | --- |
| 编写人 |  |

5）电子表单设置用例描述详细描述了管理员进行电子表单设置的过程，如表2-11：

表2-11 电子表单设置用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0011 | 用例名称 | 电子表单设置 |
| 用例描述 | 在登录系统后，系统管理员可以选择系统中的流程表单管理模块，设置相应的电子表单。 | | |
| 执行者 | 系统管理员 | | |
| 前置条件 | 管理员成功登录系统 | | |
| 后置条件 | 电子表单设置成功 | | |
| 涉众利益 | 1. 电子表单是否符合工作流程 2. 电子表单是否能实现信息的流转 | | |
| 用例场景 | 流程表单管理模块 | | |
| 基本事件流 | **基本流程（Base Flow）**  1. 管理员选择流程表单管理模块  2. 选择电子表单设置  3. 设置电子表单的类型等  4. 系统返回设置结果  **扩展流程（Extend Flow）**  1a. 类型设置不符：    1a1. 系统提示类型不符，请管理员修改  1a2. 用例结束 | | |
| 特殊需求（Special Requirement） | 1.设置的电子表单应当尽可能契合工作流程  2.电子表单应当能进行实时更新 | | |

|  |  |
| --- | --- |
| 编写人 |  |

## 2.3 非功能需求

### 2.3.1环境需求

服务器端的环境如表2-12：

表2-12 服务器端环境

|  |  |
| --- | --- |
| 系统所使用数据库 | MySQL5.7 |
| 操作系统 | Windows 10 |
| Web服务器 | Tomcat8.5 |

客户端的环境如表2-13：

表2-13 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） |
| 分辨率 | 最佳效果为1920×1080像素 |

### 2.3.2 性能需求

（1）响应时间：

必须要保证系统能够在4秒之内对用户的请求发出响应，若系统的负担较重，也应在不超过8秒的时间内完成响应。

（2）业务量：

系统每一天应当能够处理不少于500项的政府事务，并且应当能够提供不少于50人同时在线进行事务的相关处理。

### 2.3.3 安全需求

（1）系统保密性：若用户想对系统的信息进行修改与查看，必须要校验用户的权限，避免泄露系统的相关信息。

（2）系统完整性：用户想要使用系统的相关信息时，只能使用信息的原型，且需校验用户的权限。

（3）漏洞检测和安全风险评估：分析并仔细检查系统可能存在的漏洞，并需要对系统的安全风险进行仔细且完善的评估。

3 总体设计

## 3.1系统设计的原则

综合考虑该项目的实现背景以及应用情况，本项目将遵循以下的设计原则进行开发。

（1）实用性原则。本系统主要是服务于政府人员以及企业的工作人员的，所以实用性是开发过程中必须遵循的原则。

（2）先进性原则。技术的发展日新月异，先进的技术总有其特有的优势，在进行本系统的开发时，也要保证所使用系统的先进性。

（3）规范性原则。对本系统进行的设计一定要遵循相应的规范来进行，使得其有广泛使用的能力。

（4）安全性原则。本系统服务于企业与政府部门，对安全性有一定的标准。系统需能对不同的用户进行权限的校验，防止非法操作的产生，保证系统的安全性。

（5）稳定性原则。本系统服务于政府部门以及企业，对稳定性有很高的要求，系统需能在规定的时间范围内良好的工作，不至于影响客户的使用体验。

## 3.2 系统体系结构设计

在开发本系统的过程中，结合实际情况，我们选择了B/S体系结构，并且在完成系统的技术选择上，采用SSM框架完成系统的实现。

SSM框架是是目前较为流行的JAVA　EE框架，主要包括三个部分，分别是Spring，MyBatis,以及Spring MVC。SSM框架的系统架构如图3-1：

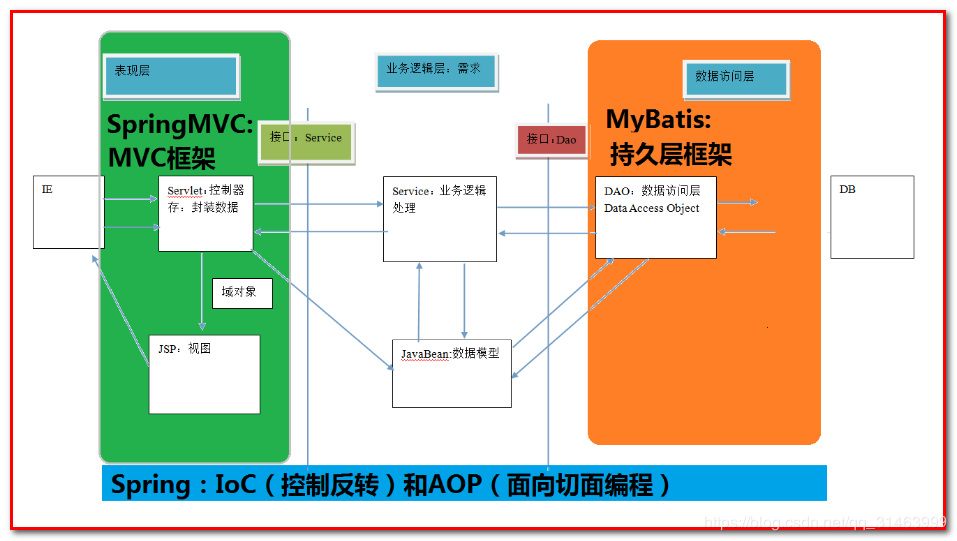


图3-1 SSM框架系统架构图

SpringMVC是一个表现层的框架。它能够监控和拦截前台的请求，解析请求中的参数，并将其传递给服务层。在本系统的开发过程之中，通过SpringMVC框架来进行前后台的数据传输。

Mybatis是一个持久层的框架，能够有效的较少对数据库进行操作的代码重复率，本项目使用该框架来进行对数据库数据的操作。

Spring框架是一个轻量型的Java开发框架， 能够有效的减少代码的耦合度，本系统使用该框架来对代码进行解耦。

## 3.3 系统功能结构设计

系统包含两个子系统，系统的总包图如图3-2：

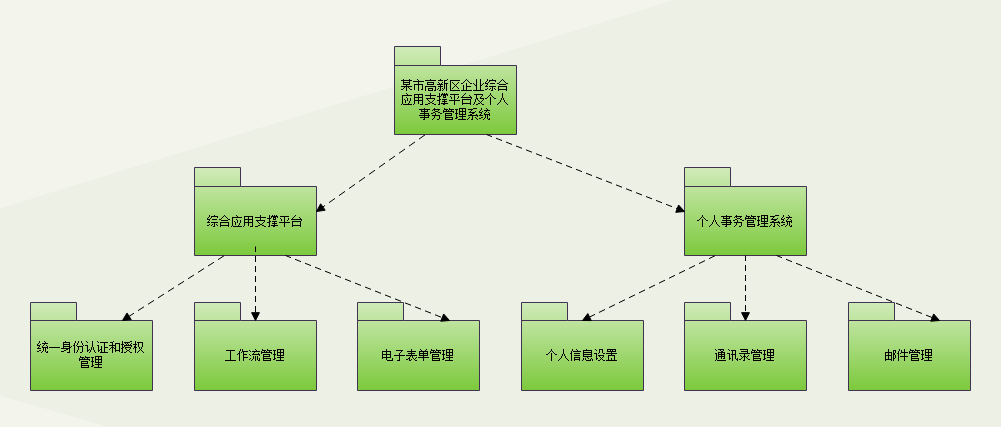


图3-2 系统总包图

如上图所示，系统一共分为两个子系统，综合应用支撑平台子系统以及个人事务管理子系统。

（1）综合应用支撑平台子系统

综合应用支撑平台的功能主要包括统一身份认证和授权管理，工作流，电子表单三个模块。统一身份认证和授权管理功能模块会对所有登录系统的人员进行相应的登录认证。同时，可以供系统管理员使用，管理员可以使用此模块的功能对所有的系统用户的信息进行管理，进行相应的增删改查操作。此外，系统管理员还可以利用此模块对所有用户进行权限的分配与管理。而工作流和电子表单这两个功能模块，主要提供给系统的用户使用，在用户登录系统后，所有的事务都会以相应的条目呈现在用户的相应界面上，通过点击相应的条目，就会跳转进入相应的工作流或者是电子表单中，用户可以清楚的了解到每一项事务的所有流程步骤，同时可以利用提供的图形化表单编辑工具以及图形化工作流编辑工具，对事务进行相应的处理。

（2）个人事务管理子系统

个人事务管理子系统主要包括个人设置，通讯录管理，内部邮件管理三个功能模块。个人事务管理系统主要面向系统的用户，用户登录系统后，可以根据账户的权限选择相应的功能进行使用。个人设置模块，提供个人的基础设置功能，包括修改密码，个人信息设定等，用户可以使用此模块对密码进行修改，以及对个人的详细信息，例如姓名，年龄，联系方式等等信息进行设置。通讯录管理模块，可以展示用户对应账号下所存储的个人通讯录以及公共通讯录。内部邮件管理模块，可以帮助用户处理系统内部的邮件信息，包括收发邮件，或者是将账户中的邮件导出至本机硬盘进行存储。

### 3.3.1 综合应用支撑平台子系统功能结构设计

综合应用支撑平台子系统包含三个功能模块，子系统包图如图3-3：

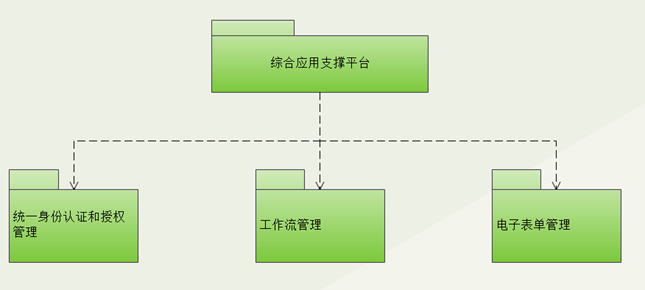


图3-3 综合应用支撑平台子系统包图

（1）统一身份认证和授权管理

此模块应包含用户管理、身份认证、授权管理等功能。其中，用户管理提供用户数据增加、修改、删除等功能；身份认证模块提供登录验证、权限验证等功能；授权管理应提供根据角色的定义和授权。

（2）工作流管理

本系统的使用者可以使用该模块的功能制定相关的工作流程，以应对不同的事务，完成相应的工作。

（3）电子表单管理

用户可以针对不同的事务设计不同的表单。

### 3.3.2个人事务管理子系统功能结构设计

个人事务管理子系统包含三个功能模块，子系统包图如图3-4：

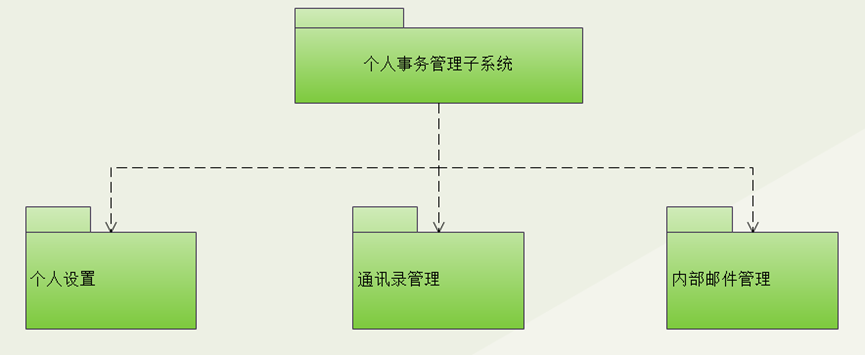


图3-4 个人事务管理子系统包图

（1）个人设置。提供个人的基础设置功能，包括修改密码，个人信息设定，个人快捷菜单设置等。

（2）通讯录管理。通讯录分公共通讯录和个人通讯录。

（3）内部邮件管理。支持主送、抄送等功能；可以直接给群组以及部门发送消息，且邮件需支持导出（另存）至本地计算机硬盘。

4 数据库设计

## 4.1 概念结构设计

### 4.1.1 设计思路

对于本系统关于数据库方面的设计，需要用到实体关系图(ER图)来进行辅助设计。针对统一身份认证和授权管理功能，需要有相应的用户信息表以及权限表，角色表。针对工作流，有相应的工作流信息表。针对个人设置，有用户信息表。针对通讯录管理功能，设计了相应的通讯录表。对于内部邮件管理，有邮件表。每个用户对应一个具体的角色，所以也对应一个具体的权限。同时，每个用户只有一个通讯录，但同时拥有多个邮件。

### 4.1.2 系统总体E-R图

在本系统的数据库中，总共包含五个实体。系统总体ER图如图4-1：



图4-1 系统总体ER图

## 4.2 逻辑结构设计

### 4.2.1 设计思路

关于本项目数据库的逻辑结构设计，是为了把本项目数据库的概念结构设计，更进一步的转变为与之对应的数据模型的过程。其中，在设计过程中，需要把项目的E-R图转变为相应的关系表，并遵照1:1联系，1:n联系，m:n联系等不同的转化规则，完成概念结构到逻辑结构的转变。

### 4.2.2 逻辑模型

通讯录表共有六个字段，如表4-1：

表4-1 通讯录表book

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 说明 |
| 1 | b\_id | 整型 | 11 | 通讯录信息id，主键 |
| 2 | b\_name | 字符型 | 255 | 联系人姓名 |
| 3 | b\_unit | 字符型 | 255 | 联系人岗位 |
| 4 | b\_phone | 字符型 | 255 | 联系人电话 |
| 5 | b\_email | 字符型 | 255 | 联系人邮箱 |
| 6 | b\_userid | 整型 | 11 | 联系人信息所属人id |

用户信息表共有七个字段，如表4-2：

表4-2 用户信息表t\_user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 说明 |
| 1 | id | 整型 | 11 | 用户信息id，主键 |
| 2 | username | 字符型 | 255 | 用户名 |
| 3 | password | 字符型 | 255 | 密码 |
| 4 | phone | 字符型 | 255 | 用户电话 |
| 5 | email | 字符型 | 255 | 用户邮箱 |
| 6 | unit | 字符型 | 255 | 用户工作单位 |
| 7 | role\_id | 整型 | 11 | 用户权限 |

邮件表共有七个字段，如表4-3：

表4-3 邮件表email

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 说明 |
| 1 | em\_id | 整型 | 11 | 邮件id，主键 |
| 2 | em\_get | 字符型 | 255 | 收件人 |
| 3 | em\_send | 字符型 | 255 | 发件人 |
| 4 | em\_time | 日期型 | 255 | 发送日期 |
| 5 | em\_head | 字符型 | 255 | 邮件标题 |
| 6 | em\_content | 字符型 | 255 | 邮件正文 |
| 7 | em\_url | 整型 | 11 | 邮件附件地址 |

角色表共有3个字段，如表4-4：

表4-4 角色表t\_role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 说明 |
| 1 | id | 整型 | 11 | 用户角色id，主键 |
| 2 | rolename | 字符型 | 255 | 角色名称 |
| 3 | remark | 字符型 | 255 | 注释 |

权限表共有3个字段，如表4-5：

表4-5 权限表t\_permission

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 说明 |
| 1 | id | 整型 | 11 | 用户权限id，主键 |
| 2 | permission\_name | 字符型 | 255 | 权限名称 |
| 3 | role\_id | 整型 | 11 | 角色id |

## 4.3 物理结构设计

### 4.3.1 存取方式

对于数据库的存取方式，通常由数据库管理系统(DBMS)来提供多种不同的选择，并由数据库管理系统根据系统所具有的数据的具体的存储方式，自行决定采用哪一种存取方式，这个过程用户一般不能进行干涉。

### 4.3.2 存储结构

本系统采用的数据库是MySQL数据库。MySQL数据库的存储结构主要分为两种，分别为B-tree和哈希索引。B-tree是一种顺序存储的结构，这种存储结构的主要使用范围是对精确匹配，范围匹配，最左匹配有所需求的系统。而哈希索引，是使用哈希值以及数据行指针进行数据项的查找与存储，是一种随机存储的结构。这种存储结构适用于对精确匹配有需求的系统，但对模糊匹配和范围匹配有需求的系统并不能完全适用。而由于本系统对于精确查询以及范围匹配都有较多的需求，所以本次系统的开发，数据库的存储结构采用B-tree的存储结构。

5 界面设计

## 5.1 界面关系图或工作流图

（1）用户工作流图

用户使用系统的详细工作流程，包括登录系统，注册账户，进入系统后进行通讯录信息的管理，进行邮件信息的相应操作以及管理，以及个人信息设置。如图5-1：



图5-1 用户工作流图

（2）管理员工作流图

管理员使用系统的详细流程，包括登录系统，对用户的信息，权限，角色等进行相应的管理，以及对系统的工作流和电子表单进行管理，如图5-2：



图5-2 管理员工作流图

## 5.2 界面设计成果

### 5.2.1 主界面

系统的主界面分为三个部分：顶部导航栏，左部导航栏以及右部的信息界面。其中顶部导航栏左上角为系统LOGO，右上角为用户的个人信息设置。左部导航栏为用户的功能选项。右部的信息界面展示了系统的总体信息。如图5-3：

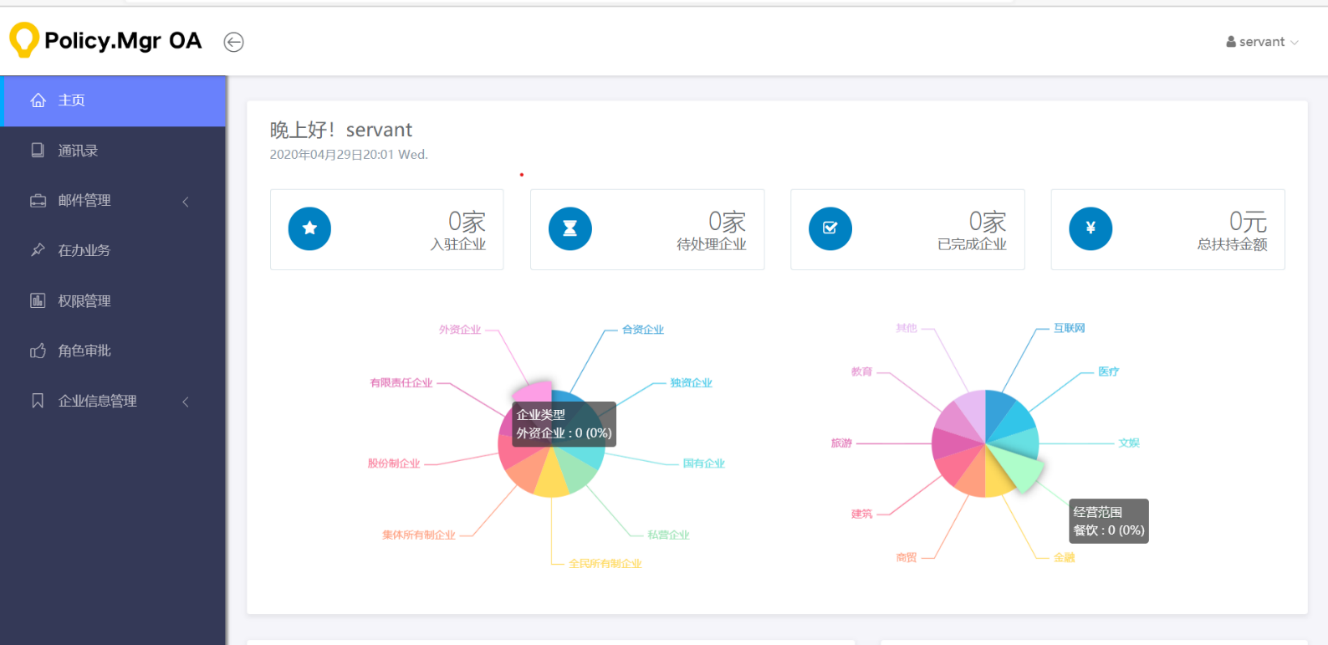


图5-3 系统主界面

### 5.2.2 子界面

用户登录子界面包含了用户输入登录信息的文本框以及登录的按钮。在界面底部还有一个“注册一个”的超链接文本，点击即可跳转至注册界面。如图5-4：

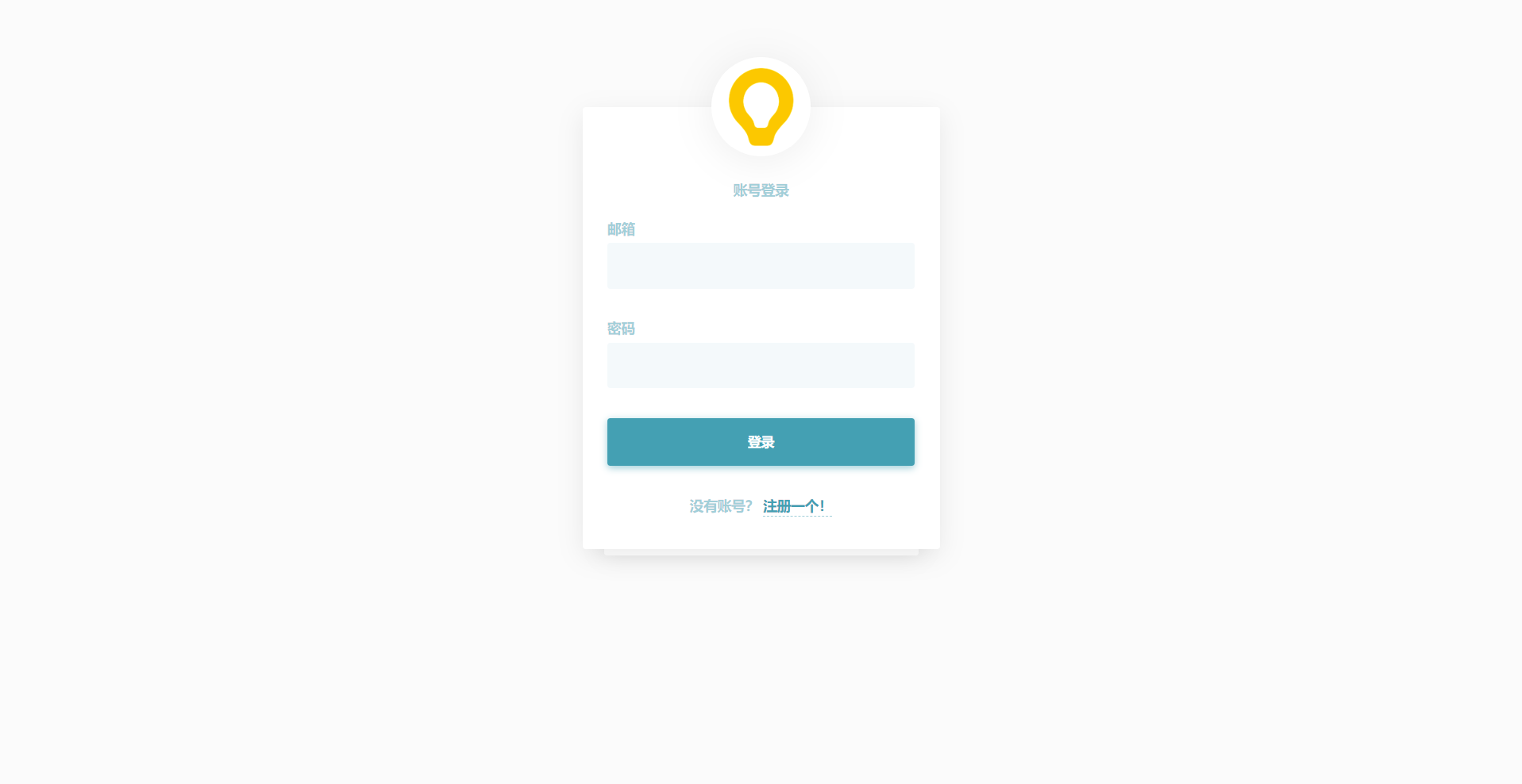


图5-4 登录子界面图

用户注册子界面包含输入用户注册信息的文本框以及注册按钮，在界面的最底部有一个“现在登录”的超链接文本，点击即可跳转至登录界面。如图5-5：

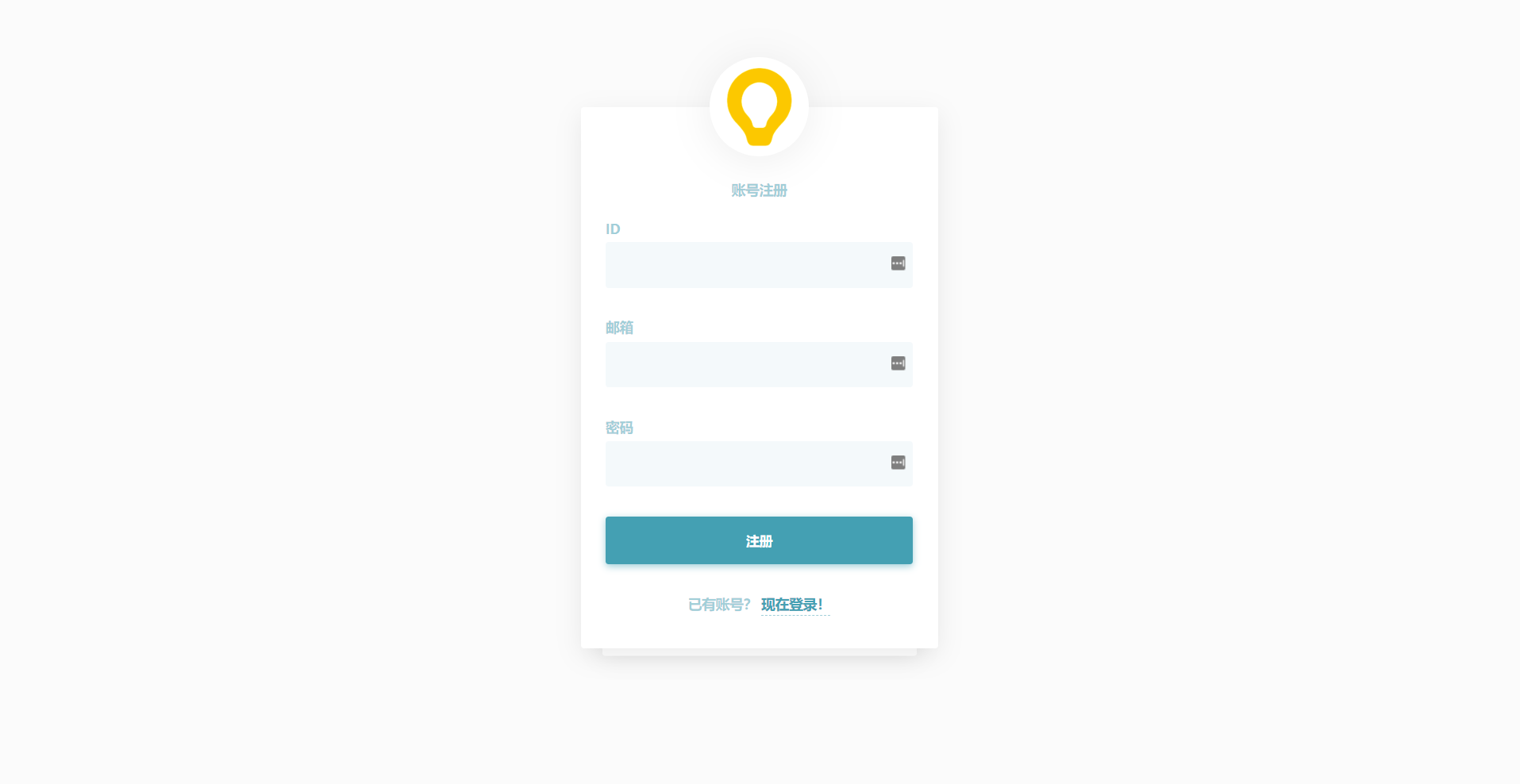


图5-5 注册子界面图

邮件写信子界面包含用户可填写的邮件相关信息的文本框，以及可选择的单发邮件或群发邮件按钮。同时，右边部分还可以选择是否进行抄送，界面最底部有发送邮件的按钮。如图5-6：

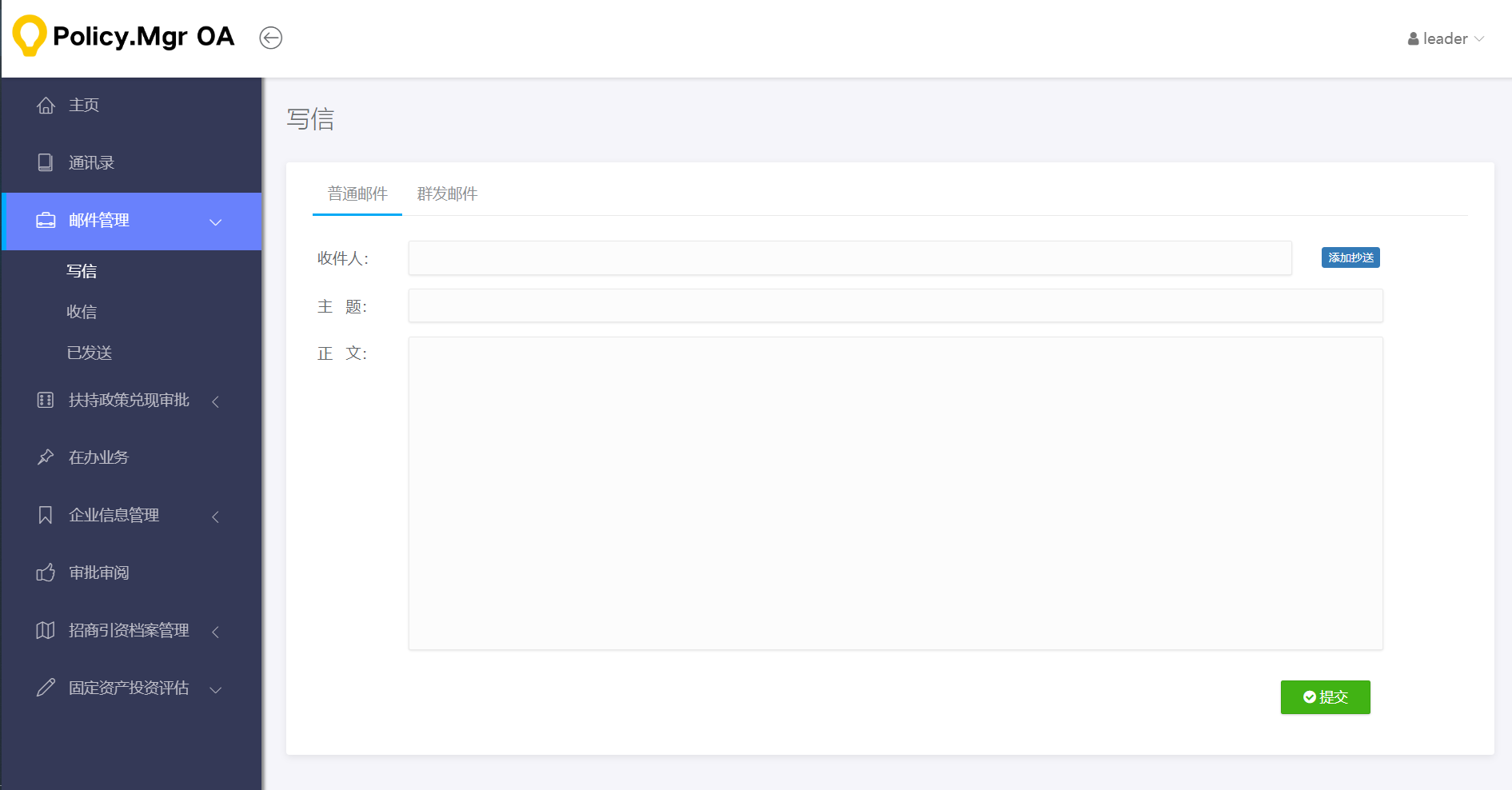


图5-6 写信子界面图

个人信息子界面展示出了用户的基本信息以及用户具有的相关权限。同时，还包含有不同的功能按钮，如查看代办的事项等等，如图5-7：

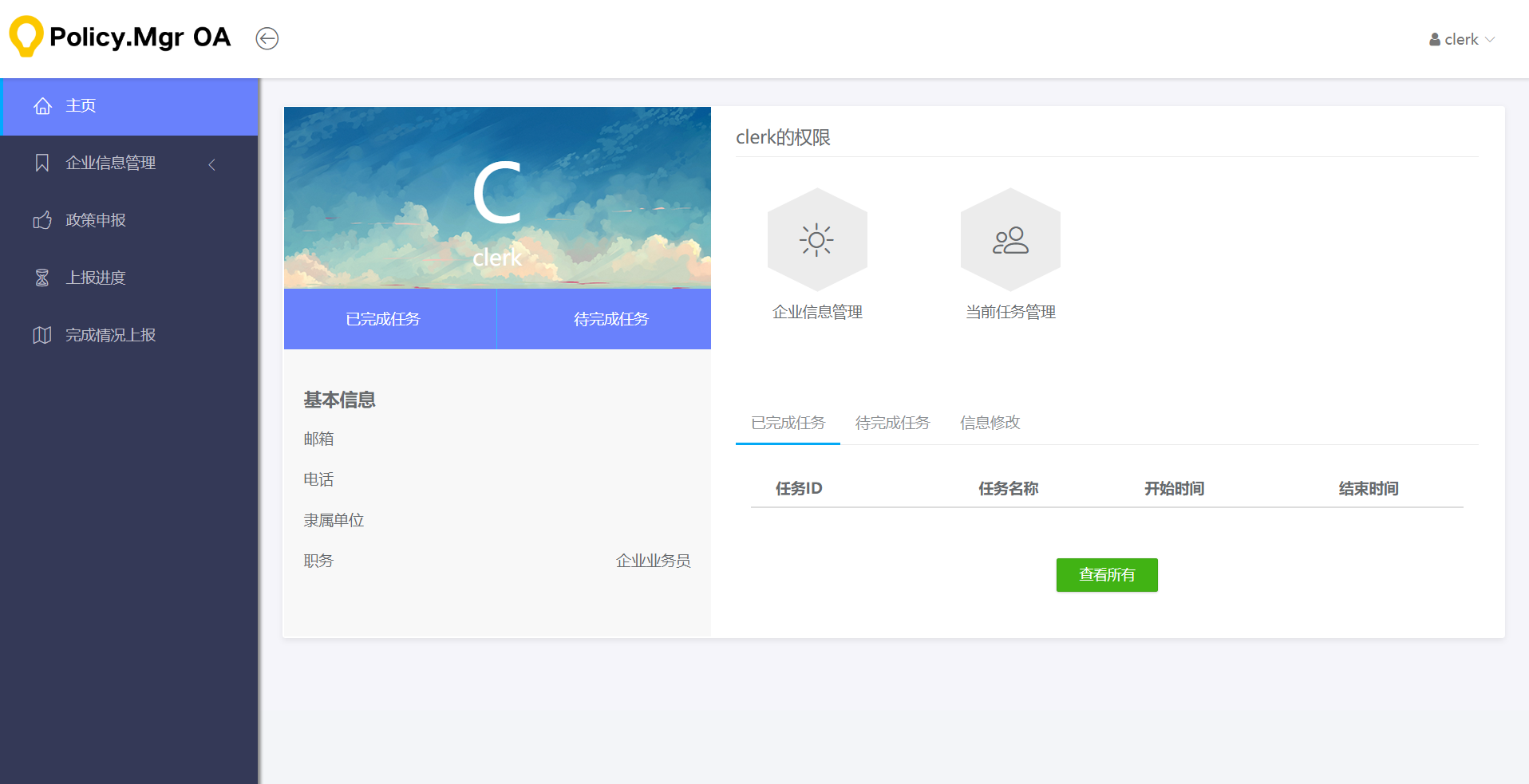


图5-7 个人信息界面图

通讯录管理子界面展示了用户所有的通讯录信息，在界面上部有“新建”按钮，用户可以新建通讯录信息。在每条通讯录信息上，用于可以选择修改或者删除操作。如图5-8：

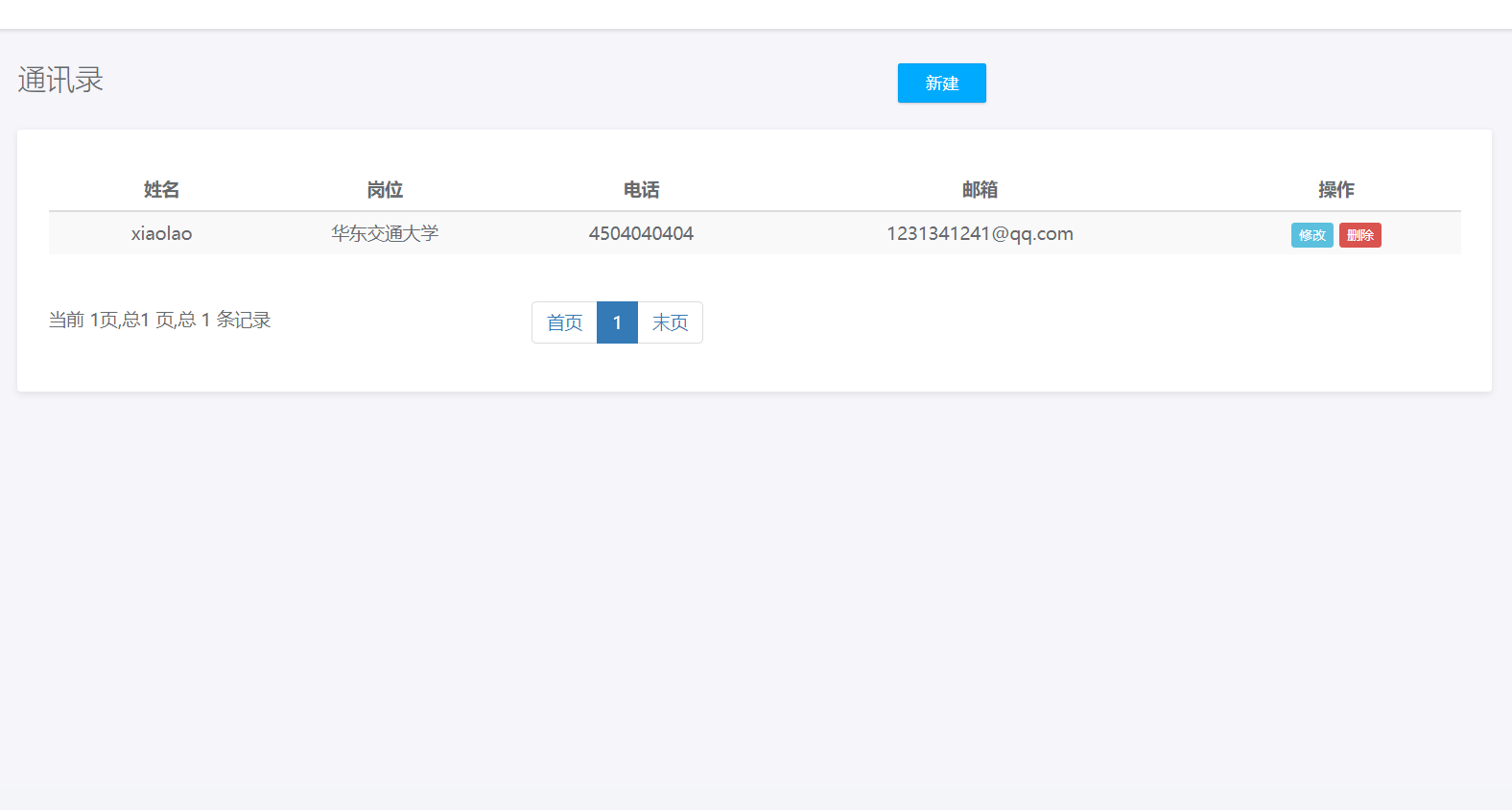


图5-8 通讯录管理界面图

6 详细设计

## 6.1 系统主要功能模块介绍

本系统分为综合应用支撑平台以及个人事务管理两个子系统。

综合应用支撑平台包含3个功能模块，分别是：

（1）统一身份认证和授权管理模块

（2）工作流管理模块

（3）电子表单管理模块

个人事务管理包含3个功能模块，分别是：

（1）个人设置功能模块

（2）通讯录管理模块

（3）电子邮件管理模块

## 6.2 统一身份认证和授权管理功能

### 6.2.1功能结构设计

统一身份认证和授权管理模块包含三个子功能模块，模块的包图如图6-1：

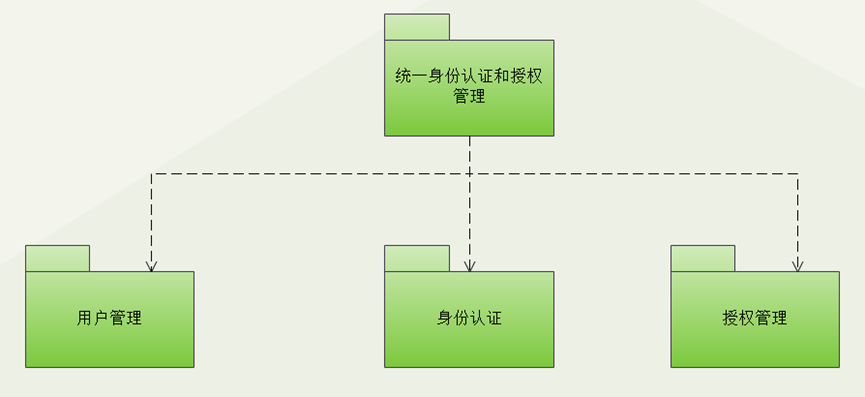


图6-1 统一身份认证和授权管理包图

用户管理：本系统的管理人员可以使用该模块来对系统的不同用户信息进行不同的操作。

身份认证：该子系统可以在用户进行登录时，对登录用户的身份进行相应的验证，根据验证的结果返回相应的登陆成功或者登录失败的信息。

授权管理：授权管理可以对所有的系统用户进行权限的授予以及对用户的权限进行相应的限定。

### 6.2.2类图设计

统一身份认证和授权管理模块的类图包含五个类，模块的类图如图6-2：

图6-2 统一身份认证和授权管理类图

统一身份认证和授权管理界面类包含三个方法，如表6-1：

表6-1 统一身份认证和授权管理界面类描述

|  |  |  |
| --- | --- | --- |
| 统一身份认证和授权管理界面类 | | |
| 类的描述：响应用户点击的不同按钮，打开不同的功能界面 | | |
| 操作描述 | Openusermanage() | 打开用户管理界面 |
| Openidentycheck() | 打开身份认证界面 |
| Openpowermanage() () | 打开权限管理界面 |

用户管理界面类包含四个方法，如表6-2：

表6-2 用户管理界面类描述

|  |
| --- |
| 用户管理界面类 |
| 类的描述：响应用户的请求，完成不同的功能 |

续表6-2 用户管理界面类描述

|  |  |  |
| --- | --- | --- |
| 操作描述 | Adduser() | 添加用户信息 |
| Updateuser() | 修改用户信息 |
| Deleteuser() | 删除用户信息 |
| Finduser() | 查找用户信息 |

身份认证界面类包含一个方法，如表6-3：

表6-3 身份认证界面类描述

|  |  |  |
| --- | --- | --- |
| 身份认证界面类 | | |
| 类的描述：实现用户的身份认证 | | |
| 操作描述 | Checkuser() | 验证用户信息 |

授权管理界面类包含三个方法，如表6-4：

表6-4 授权管理界面类描述

|  |  |  |
| --- | --- | --- |
| 授权管理界面类 | | |
| 类的描述：实现对用户权限的管理 | | |
| 操作描述 | Addpower() | 新增权限信息 |
| Deletepower() | 删除权限信息 |
| Findpower() | 查找权限信息 |

用户信息类包含七个属性以及四个方法，如表6-5：

表6-5 用户信息类描述

|  |  |  |
| --- | --- | --- |
| 用户信息类 | | |
| 类的描述：用户信息实体类 | | |
| 属性 | ID() | 用户Id |
| Name() | 用户名称 |
| 描 | Sex() | 用户性别 |
| 述 | Email() | 用户邮箱 |
|  | Telephone() | 用户号码 |
|  | Role() | 用户角色 |
|  | Address() | 用户地址 |

续表6-5 用户信息类描述

|  |  |  |
| --- | --- | --- |
| 操作描述 | Add() | 新增用户信息并保存于数据库中 |
| Delete() | 删除数据库中的用户信息 |
| Update() | 更新数据库中的用户信息 |
| Find() | 查找数据库中的用户信息 |

用户权限信息类包含三个属性以及四个方法，如表6-6：

表6-6 用户权限信息类描述

|  |  |  |
| --- | --- | --- |
| 用户权限信息类 | | |
| 类的描述：用户权限信息实体类 | | |
| 属性  描  述 | ID() | 用户Id |
| Name() | 用户姓名 |
| Power() | 用户权限 |
| 操作描述 | Add() | 新增用户权限信息并保存于数据库中 |
| Update() | 更新数据库中的用户权限信息 |
| Delete() | 删除数据库中的用户权限信息 |
| Find() | 查找本数据库中的用户权限信息 |

### 6.2.3顺序图设计

统一身份认证和授权管理顺序图详细的展示出了用户使用此模块相关功能的功能顺序，如图6-3：

图6-3 统一身份认证和授权管理顺序图

### 6.2.4核心处理流程设计

1）用户新增活动图详细展示了新增用户信息的活动过程，如图6-4：



图6-4 用户新增活动图

2）用户身份认证活动图详细展示了系统对用户的登录行为进行认证的活动过程，如图6-5：



图6-5 用户身份认证活动图

## 6.3工作流管理功能

### 6.3.1功能结构设计

工作流管理包图如图6-7：

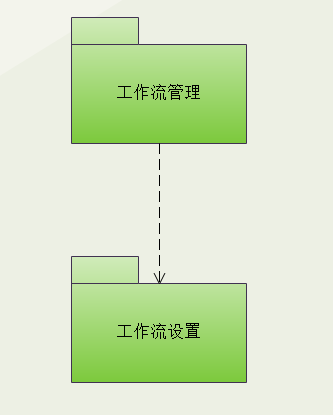


图6-7 工作流管理包图

工作流设置可以对各业务设置不同的工作流。

### 6.3.2类图设计

工作流管理功能模块共包含三个类，如图6-8：



图6-8 工作流管理类图

工作流界面类共包含一个类，如表6-7：

表6-7 工作流界面类描述

|  |  |  |
| --- | --- | --- |
| 工作流界面类 | | |
| 类的描述：展现工作流管理界面 | | |
| 操作描述 | FlowManage() | 工作流管理界面 |

工作流类共包含两个属性以及一个方法，如表6-8：

表6-8 工作流类描述

|  |
| --- |
| 工作流类 |
| 类的描述：工作流实体类 |

续表6-8 工作流类描述

|  |  |  |
| --- | --- | --- |
| 属性描述 | Flow\_name() | 工作流名称 |
| Flow\_prop() | 工作流属性 |
| 操作描述 | Create() | 工作流新建 |

工作流管理类共包含一个方法，如表6-9：

表6-9 工作流管理类描述

|  |  |  |
| --- | --- | --- |
| 工作流管理类 | | |
| 类的描述：对工作流进行管理 | | |
| 操作描述 | CreateFlow() | 新建工作流操作 |

### 6.3.3顺序图设计

工作流管理顺序图详细展示了用户使用此模块时的功能顺序，如图6-9：



图6-9 工作流管理顺序图

### 6.3.4核心处理流程设计

工作流新建活动图详细展示了新建工作流的活动流程，如图6-10：



图6-10 工作流新建活动图

## 6.4电子表单管理功能

### 6.4.1功能结构设计

电子表单管理包图如图6-11：

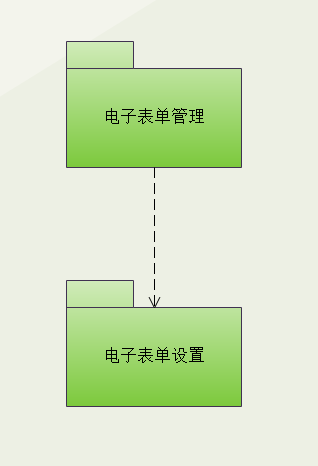


图6-11 电子表单管理包图

用户可以使用图形化的工具制作表单，定义各个控件的属性，制作流程中需要使用的各类表单。提供表单建模工具，降低表单设计的时间和难度。

### 6.4.2类图设计

电子表单管理功能模块共有三个类，如图6-12：



图6-12 电子表单管理类图

电子表单界面类包含一个方法，如表6-10：

表6-10 电子表单界面类类描述

|  |  |  |
| --- | --- | --- |
| 电子表单界面类 | | |
| 类的描述：展现电子表单管理界面 | | |
| 操作描述 | FromManage() | 电子表单管理界面 |

电子表单类包含两个属性以及一个方法，详细描述如表6-11：

表6-11 电子表单类描述

|  |  |  |
| --- | --- | --- |
| 电子表单类 | | |
| 类的描述：工作流实体类 | | |
| 属性  描述 | From\_name() | 电子表单名称 |
| From\_prop() | 电子表单属性 |
| 操作描述 | Create() | 电子表单新建 |

电子表单管理类包含一个方法，详细描述如表6-12：

表6-12 电子表单管理类描述

|  |  |  |
| --- | --- | --- |
| 电子表单管理类 | | |
| 类的描述：对电子表单进行管理 | | |
| 操作描述 | CreateFrom() | 新建电子表单操作 |

### 6.4.3顺序图设计

电子表单管理模块运行时的顺序如图6-15：



图6-13 电子表单管理顺序图

### 6.4.4核心处理流程设计

电子表单新建活动图详细展示了新建电子表单的活动流程，如图6-14：



图6-14 电子表单新建活动图

## 6.5个人设置功能

### 6.5.1功能结构设计

个人设置功能模块包含两个子模块，分别为信息查看模块以及信息修改模块。个人设置功能模块的包图如图6-15：

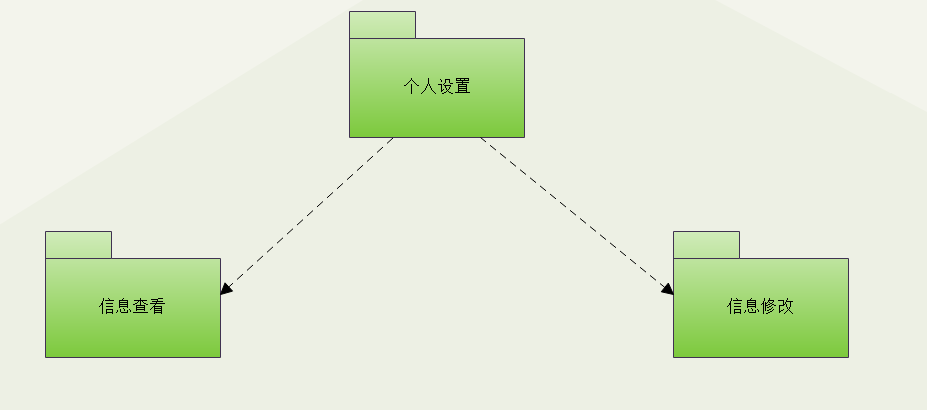


图6-15 个人设置功能包图

个人设置功能提供对个人的基础设置功能，用户可以使用此功能来查看个人的详细信息，并对相应的信息进行修改，如修改登录密码等。

### 6.5.2类图设计

个人设置功能模块包含三个类，如图6-16：

图6-16 个人设置类图

个人设置界面类包含两个方法，如表6-13：

表6-13 个人设置界面类描述

|  |  |  |
| --- | --- | --- |
| 个人设置界面类 | | |
| 类的描述：对个人信息进行查看和设置 | | |
| 操作描述 | CheckUserInfo() | 查看个人信息 |
| EditUserInfo() | 编辑个人信息 |

个人设置管理类包含两个方法，如表6-14：

表6-14 个人设置管理类描述

|  |  |  |
| --- | --- | --- |
| 个人设置管理类 | | |
| 类的描述：对个人信息进行操作 | | |
| 操作描述 | FindUserInfo() | 查找个人信息 |
| UpdateUserInfo() | 更新个人信息 |

个人信息类包含六个属性以及两个方法，如表6-15：

表6-15 个人信息类描述

|  |  |  |
| --- | --- | --- |
| 个人信息类 | | |
| 类的描述：个人信息实体类 | | |
| 属性  描  述 | ID() | 用户Id |
| Name() | 用户名称 |
|  | Sex() | 用户性别 |
|  | Email() | 用户邮箱 |
|  | Telephone() | 用户号码 |
|  | Address() | 用户地址 |
| 操作描述 | Update() | 更新数据库中的用户信息 |
| Find() | 查找数据库中的用户信息 |

### 6.5.3顺序图设计

用户使用个人设置功能模块时，功能顺序如图6-17：

图6-17 个人设置顺序图

### 6.5.4核心处理流程设计

用户在查看自己的个人信息时，相应的活动图如图6-18：



图6-18 查看个人信息活动图

## 6.6通讯录管理功能

### 6.6.1功能结构设计

通讯录管理功能模块包含四个功能子模块，模块包图如图6-20：

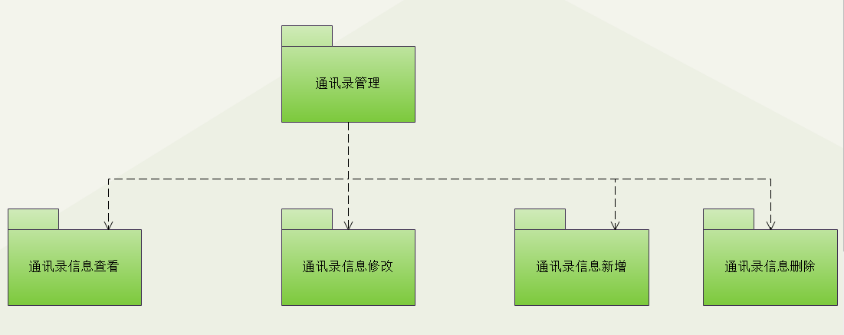


图6-20 通讯录管理功能包图

在通讯录管理模块中，用户可以对个人以及公共的通讯录进行操作，如可以查看详细的通讯录条目信息，可以对通讯录中的信息进行修改，可以在通讯录中新增详细的通讯信息，还可以删除相关的通讯录信息。

### 6.6.2类图设计

通讯录管理模块总共包含三个类，如图6-21：



图6-21 通讯录管理类图

通讯录管理界面类包含一个方法，如表6-15：

表6-15 通讯录管理界面类图描述

|  |  |  |
| --- | --- | --- |
| 通讯录管理界面类 | | |
| 类的描述：对通讯录信息进行管理 | | |
| 操作描述 | TeleManage() | 通讯录管理类 |

通讯录管理类包含四个方法，详细描述如表6-16：

表6-16 通讯录管理类描述

|  |  |  |
| --- | --- | --- |
| 通讯录管理类 | | |
| 类的描述：对通讯录信息进行增删改查 | | |
| 操作 | TeleAdd() | 新增通讯录信息 |
| TeleDelete() | 删除通讯录信息 |
| 描 | TeleUpdate() | 修改通讯录信息 |
| 述 | TeleFind() | 查找通讯录信息 |

通讯录信息类包含四个属性以及四个方法，详细描述如表6-17：

表6-17 通讯录信息类描述

|  |  |  |
| --- | --- | --- |
| 通讯录信息类 | | |
| 类的描述：通讯录信息实体类 | | |
| 属  性 | ID() | 用户Id |
| Tele\_Name() | 联系人姓名 |

续表6-17 通讯录信息类描述

|  |  |  |
| --- | --- | --- |
| 描 | Tele\_Email() | 联系人邮箱 |
| 述 | Tele\_Phone() | 联系人号码 |
| 操  作 | Update() | 更新数据库中的联系人信息 |
| Find() | 查找数据库中的联系人信息 |
| 描 | Add() | 新增数据库中的联系人信息 |
| 述 | Delete() | 删除数据库中的联系人信息 |

### 6.6.3顺序图设计

用户使用通讯录管理功能模块时，相应的功能模块使用顺序如图6-22：



图6-22 通讯录管理顺序图

### 6.6.4核心处理流程设计

用户修改通讯录的活动流程如图6-23：



图6-23 修改通讯录信息活动图

## 6.7内部邮件管理功能

### 6.7.1功能结构设计

内部邮件管理功能模块包含四个功能子模块，如图6-24：

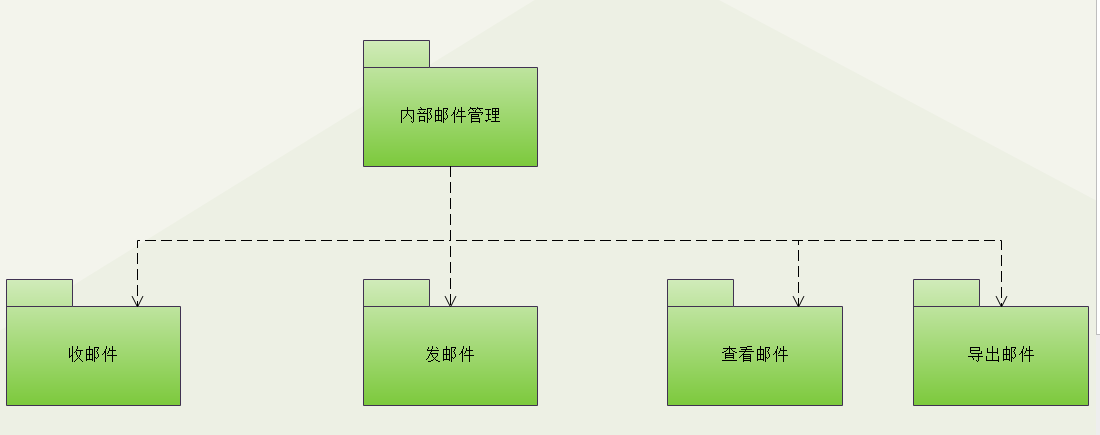


图6-24 内部邮件管理包图

1）收邮件：用户可以通过内部邮件管理系统，收到来自同一系统的用户发来的邮件。

2）发邮件：用户可以通过内部邮件管理系统，向同一系统的其他用户发送邮件。

3）查看邮件：用户可以通过内部邮件管理系统，查看自己账号下的已收邮件和已发邮件。

4）导出邮件：用户可以通过内部邮件管理系统，导出自己账号下的邮件至本地，以文本文件的形式保存在硬盘。

### 6.7.2类图设计

内部邮件管理功能模块共包含三个类，如图6-25：



图6-25 内部邮件管理类图

内部邮件管理界面类包含一个方法，详细描述如表6-18：

表6-18 内部邮件管理界面类图描述

|  |  |  |
| --- | --- | --- |
| 内部邮件管理界面类 | | |
| 类的描述：对内部邮件进行管理 | | |
| 操作描述 | EmailManage() | 内部邮件管理类 |

内部邮件管理类包含四个方法，详细描述如表6-19：

表6-19 内部邮件管理类描述

|  |  |  |
| --- | --- | --- |
| 内部邮件管理类 | | |
| 类的描述：对内部邮件进行接受，发送等操作 | | |
| 操作 | Email\_Send() | 发送邮件 |
| Email\_Receive() | 接受邮件 |
| 描 | Email\_Find() | 查看邮件 |
| 述 | Email\_Save() | 保存邮件 |

内部邮件信息类包含三个属性以及三个方法，详细描述如表6-20：

表6-20 内部邮件信息类描述

|  |  |  |
| --- | --- | --- |
| 内部邮件信息类 | | |
| 类的描述：内部邮件信息实体类 | | |
| 属  性  描 | Email\_Sname() | 邮件发件人 |
| Email\_Rname() | 邮件收件人 |
| 述 | Email\_Content | 邮件内容 |
| 操作 | Find() | 查找数据库中的邮件信息 |
| 描 | Add() | 新增数据库中的邮件信息 |
| 述 | Save() | 保存数据库中的邮件信息至本地 |

### 6.7.3顺序图设计

用户使用内部邮件管理模块的功能顺序图如图6-26：



图6-26 内部邮件管理顺序图

### 6.7.4核心处理流程设计

用户使用邮件管理功能模块发送邮件时的活动过程，如图6-27：



图6-27 邮件发送活动图

用户使用邮件管理功能模块保存邮件时的活动过程，如图6-28：



图6-28 邮件保存活动图

7 编码

## 7.1 代码实现与核心算法

### 7.1.1 用户管理核心代码

（1）用户登录核心代码

本段代码主要实现用户的登录功能，对用户输入的登录信息进行验证，并通过用户的ID查询用户是否进行角色申请，以决定用户是否需要跳转至角色申请界面。

|  |
| --- |
|  |

（2）用户注册核心代码

此段代码是用户进行注册的核心代码，可以通过用户输入的用户名判断是否已被注册，并在用户注册成功之后，跳转至系统主页。

|  |
| --- |
|  |

（3）删除用户信息

此段代码主要作用是删除用户的信息，并在完成操作之后跳转至用户信息界面。

|  |
| --- |
|  |

（4）修改用户信息核心代码

此段代码主要提供修改用户信息的功能，并携带修改后的信息跳转至信息编辑界面。

|  |
| --- |
|  |

（5）申请角色核心代码

此段代码可以在用户登录系统之后，若没有申请角色，为用户提供申请角色的功能。

|  |
| --- |
|  |

### 7.1.2 邮件管理核心代码

（1）获取已发邮件核心代码

此段代码可以根据当前登录的用户名查询到所有已发送的邮件。

|  |
| --- |
|  |

（2）获取已收邮件核心代码

此段代码可以通过获取当前登录用户的用户名，查询到所有已发送的邮件。

|  |
| --- |
|  |

（3）查看邮件内容核心代码

此段代码可以查看所有已收到的邮件的详细信息。

|  |
| --- |
|  |

（4）删除邮件核心代码

此段代码可以使用户通过发件人以及通过收件人来删除相关的邮件。

|  |
| --- |
|  |

（5）单独发送邮件核心代码

此段代码可以为用户提供单独向某个用户发送邮件的功能，并选择是否进行抄送。

|  |
| --- |
|  |

（6）群发邮件代码核心代码

此段代码为用户提供了群发邮件的功能。

|  |
| --- |
|  |

### 7.1.3 通讯录管理核心代码

（1）获取所有通讯录信息核心代码

此段代码为通过获取当前登录的用户的用户名信息，来查询该用户所有的通讯录信息。

|  |
| --- |
|  |

（2）管理通讯录信息核心代码

此段代码给用户提供了更新通讯录信息的操作。

|  |
| --- |
|  |

### 7.1.4工作流管理核心代码

此段代码进行的是工作流的相关部署，包括相关zip文件的上传，以及相应的文件的布置。

|  |
| --- |
|  |

## 7.2 代码优化分析

为了优化代码的结构，提高系统的运行效率，在编写代码时创建了相应的工具类，降低了代码的重复率。

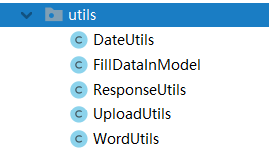


图7-1 utils包

如图7-1所示，包中的DateUtils类主要负责对不同的时间的格式，转换成标准统一的时间格式。它能被其他类和页面引用，有利于减少重复的代码。

图中的FillDateModel类主要的作用是导出相应的数据流为excel表格。

图中的ResponseUtils类负责对导出的文件进行下载的功能。

图中的UploadUtils主要负责对系统的数据进行上传。

8 测试

## 8.1 测试方案设计

### 8.1.1 测试策略

测试策略总体来说包含两个部分：功能测试，系统测试。

功能测试：测试在系统正常运行的情况下，各模块功能能否正常实现。

系统测试：在功能测试完成的基础上，检测对于系统的需求是否符合用户的标准和预期，以及对于系统的数据流处理和业务流处理是否存在不合理和错误的情况。

在对系统功能进行测试时，相关的功能测试策略如表8-1：

表8-1 功能测试策略表

|  |  |
| --- | --- |
| 测试目标 | 确保测试的功能正常，包括用户的登录注册，对用户信息的管理，对通讯录信息的管理，对邮件的管理，用户的个人设置以及对工作流，电子表单的管理。 |
| 测试范围 | 登录注册，对用户信息的管理，对通讯录信息的管理，对邮件的管理，用户的个人设置以及对工作流，电子表单的管理。 |
| 技术 | 分别设计一部分有效和另一部分无效的不同数据，以期获得如下的结果：  在使用有效数据进行相应的测试时，系统能够反馈出正确的结果。  在使用无效数据进行测试时，系统会返回错误信息。 |
| 开始标准 | 成功进行登录操作 |
| 完成标准 | 不存在相应的一级、二级缺陷，三级与四级缺陷要保证50%的修改并且重新测试通过。 |
| 测试重点和优先级 | 用户信息管理，邮件管理 |
| 需考虑的特殊事项 | 系统运行环境，网络问题 |

在对系统进行系统测试时，相应的系统测试策略如表8-2：

表8-2 系统测试策略表

|  |  |
| --- | --- |
| 测试目标 | 测试系统相关的业务流和数据流是否正确 |
| 测试范围 | 由几个不同的功能模块组合而成的一个大功能，或是系统具体的业务流程 |
| 技术 | 分别设计一部分有效和另一部分无效的不同数据，以期获得如下的结果：  在使用有效数据进行相应的测试时，系统能够反馈出正确的结果。  在使用无效数据进行测试时，系统会返回错误信息。 |
| 开始标准 | 在结束某一个集成测试的时候，必须达到通过的标准 |
| 完成标准 | 结束所有计划中的测试并且解决所有等级较高的缺陷 |
| 测试重点和优先级 | 用户信息管理，邮件管理 |
| 需考虑的特殊事项 | 系统运行环境，网络问题 |

### 8.1.2 测试进度安排

在进行测试时，相应的测试进度安排如表8-3：

表8-3 测试进度安排表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试阶段 | 测试任务 | 工作量估计 | 人员分配 | 起止时间 |
| 第一阶段  功能测试 | 1）登录注册功能  2）用户信息管理  3）通讯录管理  4）邮件管理  5）个人设置  6）工作流管理 | 4日 |  | 2020年4月11日至2020年4月14日 |
| 第二阶段  系统测试 | 1) 完成所模块的组合测 试  2) 确定所有业务流向和 数据都是正确的 | 3日 |  | 2020年4月15日至2020年4月17日 |
| 第三阶段  性能测试 | 在多用户访问，交替进行 负荷压迫测试 | 1日 |  | 2020年4月18日至2020年4月19日 |
| 第四阶段  安装部署测试 | 对与项目部署进行测 | 2日 |  | 2020年4月20日至2020年4月22日 |
| 第五阶段  兼容测试 | 软件在各个软件平台上 的运行情况 | 1日 |  | 2020年4月23日至2020年4月24日 |

### 8.1.3 测试资源

（1）测试环境

操作系统：Windows10家庭版

硬件：GTX940mx独立显卡，英特尔i5 6200u处理器，8G内存，256G SSD

服务器：Tomcat8.5

数据库：Mysql5.6

（2）测试管理工具

使用Google公司的Chrome浏览器进行测试。

（3）测试文档

系统相关测试用例，软件缺陷文档

### 8.1.4 关键测试点

在进行功能测试的时候，注意邮件的群发功能。

在对用户信息进行管理的时候，注意对用户角色和权限管理的测试。

## 8.2 测试用例构建

### 8.2.1 测试用例编写约定

用例编号：一个测试用例的唯一标识 用例名称：对需要测试的项目的简单的描述

前提条件：在运行相应的测试项目时预先准备的设置操作步骤：在运行相应的测试时需要完成的操作 预期结果：在完成测试后想要得到的结果实际结果：完成测试后系统实际输出的结果问题描述：执行用例之后系统展示的问题验证结果：此测试用例是否成功通过测试 BUG 编号：填写在测试过程中遇到的BUG编号 测试结果与结论：填写测试最终通过与否的结果测试执行者：执行该测试用例的人员姓名 。

### 8.2.2 测试用例设计

用户注册测试用例如表8-4：

表8-4 用户注册测试用例表

|  |  |
| --- | --- |
| 用例编号 | 1-1 |
| 用例名称 | 用户注册 |
| 前置条件 | 用户进入注册页面 |
| 操作步骤 | 1）邮箱为空，密码输入正确  2）邮箱输入正确，密码为空  3）邮箱格式不正确，密码正确  4）邮箱输入正确，密码小于6位或大于18位  5）邮箱输入正确，密码在6到18位之间但只有数字或字母  6）邮箱输入正确，密码输入正确 |
| 预期结果 | 1）提示邮箱不能为空  2）提示密码不能为空  3）提示邮箱格式不正确  4）提示密码必须在6到18位之间  5）提示密码必须包含字母数字  6）提示注册成功，跳转至登录界面 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |

|  |  |
| --- | --- |
| BUG描述 | 暂无 |
| 测试结果和结论 | 测试通过 |
| 测试执行者 |  |

用户登录测试用例如表8-5：

表8-5 用户登录测试用例表

|  |  |
| --- | --- |
| 用例编号 | 1-2 |
| 用例名称 | 用户登录 |
| 前置条件 | 用户进入系统首页 |
| 操作步骤 | 1）输入邮箱为空，密码正确  2）输入邮箱正确，密码为空  3）输入邮箱正确，密码错误  4）输入邮箱正确，密码正确 |

续表8-5 用户登录测试用例表

|  |  |
| --- | --- |
| 预期结果 | 1）提示邮箱不能为空  2）提示密码不能为空  3）提示密码错误  4）登录成功，进入系统主界面 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |
| BUG描述 | 暂无 |
| 测试结果和结论 | 测试通过 |
| 测试执行者 |  |

修改密码测试用例如表8-6：

表-8-6 修改密码测试用例表

|  |  |
| --- | --- |
| 用例编号 | 1-3 |
| 用例名称 | 修改密码 |
| 前置条件 | 用户进入个人设置，选择修改密码 |
| 操作步骤 | 1）新密码为空  2）新密码与 原密码相同  3）新密码小于6位或大于18位  4）新密码在6到18位之间但只有数字或字母  5）新密码输入正确 |
| 预期结果 | 1）提示密码不能为空  2）提示密码不能够与 原密码相同  3）提示密码必须在6到18位之间  4）提示密码必须包含字母数字  5）提示修改成功 |
| 实际结果 | 与预期结果相符 |

|  |  |
| --- | --- |
| 问题描述 | 暂无 |
| BUG描述 | 暂无 |
| 测试结果和结论 | 测试通过 |
| 测试执行者 |  |

个人信息修改测试用例如表8-7：

表8-7 个人信息修改测试用例表

|  |  |
| --- | --- |
| 用例编号 | 1-4 |
| 用例名称 | 修改个人信息 |
| 前置条件 | 用户进入个人设置，选择修改个人信息 |
| 操作步骤 | 1）存在信息为空  2）全部信息正确填入 |
| 预期结果 | 1）提示信息不能为空  2）提示修改成功 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |
| BUG描述 | 暂无 |

续表8-7 个人信息修改测试用例表

|  |  |
| --- | --- |
| 测试结果和结论 | 测试通过 |
| 测试执行者 |  |

通讯录信息录入测试用例如表8-8：

表8-8 通讯录信息录入测试用例表

|  |  |
| --- | --- |
| 用例编号 | 1-5 |
| 用例名称 | 通讯录信息录入 |
| 前置条件 | 用户进入通讯录管理，选择新增通讯录信息 |
| 操作步骤 | 1）姓名为空  2）姓名填写正确，邮箱为空  3）姓名填写正确，邮箱格式不正确、  4）姓名，邮箱填写正确，电话为空  5）姓名，邮箱，电话填写正确，岗位为空  6）所有信息填写正确 |
| 预期结果 | 1）提示姓名不能为空  2）提示邮箱不能为空  3）提示邮箱格式不正确  4）提示电话不能为空  5）提示岗位不能为空  6）提示操作成功 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |
| BUG描述 | 暂无 |
| 测试结果和结论 | 测试通过 |
| 测试执行者 |  |

发送邮件测试用例如表8-9：

表8-9 发送邮件测试用例表

|  |  |
| --- | --- |
| 用例编号 | 1-6 |
| 用例名称 | 发送邮件 |
| 前置条件 | 用户进入邮件管理，选择写信 |
| 操作步骤 | 1）收件人为空  2）收件人填写正确，主题为空  3）收件人，主题正确，正文为空 |
| 预期结果 | 1）提示收件人不存在  2）提示发送成功  3）提示发送成功 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |
| BUG描述 | 暂无 |
| 测试结果和结论 | 测试通过 |
| 测试执行者 |  |

### 8.2.3 测试用例维护

由于系统需求的变更以及系统版本的迭代，会存在部分用例过时或已经不再适用，甚至是产生部分冗余的用例等等情况，这时需要我们对相应的用例进行删除及维护。

（1）系统需求发生改变

由于用户对系统功能的需求发生改变导致系统的功能模块发生了相应的改变，需要对相应的测试用例进行修改，删除等操作。

（2）处理冗余的用例

在设计测试用例的时候，可能会存在冗余的情况，要及时处理这些冗余的用例。

（3）系统版本发生迭代

在系统的版本发生迭代时，代码的架构以及整个系统的功能体系都可能会发生变化，这时要及时处理过时的用例以及无效的用例，并设计相应的新的用例。

9 总结与展望

## 9.1 设计工作总结

经过努力以及在身边老师朋友们的帮助之下，最终完成了这个系统的设计与实现。在完成整个系统的过程中，大体上完成了如下的几项工作：

（1）在刚刚开始这个毕业设计的课题时，由于以前并没有接触相关的领域，所以前期对课题相关的情况进行了调研，了解了高新区政府的办公流程以及工作人员日常工作过程中处理的事务，对高新区办公流程之中涉及的不同角色和每个角色需要具备的相关权限进行了详细的分析，这些信息形成了对该课题所需系统的大体的框架。

（2）在系统的编码实现阶段，为了尽可能高效的实现系统的相关功能，在比较了不同的JAVA框架之后，最后选择了开发效率比较高的SSM框架来进行系统后端的开发。SSM框架由开源框架Spring和Mybatis整合而成，可以降低系统的耦合度，简化开发。而为了能有美观易用的前端界面，选择了Bootstrap框架来进行前端界面的编写。Bootstrap的响应式布局设计，可以使得系统可以适应不同分辨率的设备，并给用户提供更好的视觉使用体验。

（3）在完成系统的编码实现之后，为了进一步的完善系统的功能，对系统进行了一系列的测试，这些严密的测试帮助解决了系统的大部分问题，但是还是有些问题依然存在，这些问题需要在以后的工作之中逐步解决。

（4）毕业设计逐渐步入尾声，通过对毕业设计各个阶段的相关文档进行整合与仔细的修改完善，最终形成了本项目相关的毕业设计论文。

## 9.2 未来工作展望

虽然毕业设计已经告一段落了，但是我的工作还远远没有 。由于时间比较仓促以及受技术水平所限，本系统还有许多的问题需要解决，还有很多的地方需要完善。比如，相关的工作流管理在进行图形化定制工作流程时，提供的选择还不够丰富，造成这个问题的原因是因为对政府部门处理不同事物的工作流程了解的还不够多，需要在以后进一步深入了解政府部门的不同工作流程，扩展工作流管理的功能。而在对用户进行角色和权限的管理的相关功能中，由于对使用的shiro权限管理框架还不够熟练，造成这方面的功能偏简单，需要在后面的日子里，更深入的学习shiro框架的相关技术，完善这方面的相关功能。在以后的日子里，需要不断对系统进行完善和功能的扩展，争取为政府部门提供一个功能完善可靠的系统。

谢 辞

参考文献

[1]庞晓丽.基于JavaScript的微信理财产品销售系统的设计与实现[D].吉林大学,2016.

[2]黄卉.基于Web的宜春职业技术学院资产管理系统的设计与实现[D].电子科技大学,2013.

[3]张宇.办公自动化与移动办公系统的设计与实现[D].北京邮电大学,2010.

[4]青海省政府办公自动化系统[J].办公自动化,2003(01):2-5.

[5]杨雪梅.基于SSH2的轻博客系统的研究与实现[D].吉林大学,2016.

[6]林海琼.银基房屋租赁管理系统的设计与实现[D].电子科技大学,2014.

[7]崔晨.警察类高等院校学生管理系统的分析与设计[D].云南大学,2013.

[8]贺建峰,李雅娜,张俊,孟强,杨莹,谢颖夫.基于Bootstrap的脑卒中评分系统设计与实现[J].软件导刊,2018,17(3):93-96.

[9]王渊博,周树军.农村物流“最后一公里”互联网信息服务平台研究[J].现代电子技术,2018,41(22):34-37.

[10]王志文,钱松荣.大型预制构件预应力采集管理系统[J].微型电脑应用,2018,34(12):93-96.

[11]傅琦卉.省级电子政务办公系统设计与实现[D].湖南大学,2017.

[12]王梓,夏凯.基于SSM框架的水质监测数据管理系统研究[J].计算机时代,2018(7):25-27.

[13]蒙杰,杨生举,施韶亭.基于SSM轻量级复合框架的科技创新券管理系统研究与实现[J].兰州文理学院学报：自然科学版,2018,32(2):67-70.

[14]金文,姚凯学.基于物联网的农业墒情监测系统的设计与实现[J].计算机应用与软件,2018,35(3):84-88.

[15]毛莹,廖礼俊.基于Web的智能化软件售后服务管理系统[J].信息化研究,2018,44(2):50-56.

[16]李慧锋.基于CDIO计算机工程教育教学平台研究与实现[D].上海交通大学,2013.

[17]丁飞.基于Spring mvc户外广告管理系统访问控制实现[J].电脑与信息技术,2018,26(3):33-35.

[18]王鹤琴,张林静,朱珍元.基于Spring MVC的后台管理系统开发研究[J].黄山学院学报,2018,20(3):18-22.

[19]乔欢,曾勇(指导).基于云平台的备忘录系统设计[J].科技视界,2018(31):62-63.

[20]王先清.基于SSM架构博客平台设计与实现[J].电脑知识与技术：学术交流,2018,14(7):107-109.

[21]赵佳.基于Web的硕士教育虚拟教室平台的设计与实现[D].天津大学,2016.

[22]哀建宇.基于内容关联密钥技术的密钥管理服务器的设计与实现[D].华中科技大学,2018.

[23]李先耀. 基于SpringMVC房屋销售管理系统的设计与实现[D].电子科技大学,2019.

[24]Chenxi Li. Design and Implementation of University Student Organization Office Automation System[C]. Advanced Science and Industry Research Center.Proceedings of 2018 3rd International Conference on Control, Automation and Artificial Intelligence (CAAI 2018).Advanced Science and Industry Research Center:Science and Engineering Research Center,2018:44-48.

[25]Wu Wei. Application of Computer Technology in Office Automation[C]. International Information and Engineering Association.Proceedings of 2018 2nd International Conference on Systems,Computing,and Applications(SYSTCA 2018).International Information and Engineering Association:计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society),2018:178-181.

[26]Manan Shah,Rajbir Samra. Robotic process automation and its implications for programme management office[J]. Journal of Digital Banking,2019,4(3).

[27]Xian Hong. Application and Reform of Office System Automation in Higher Vocational Colleges[C]. Institute of Management Science and Industrial Engineering.Proceedings of 2019 International Seminar on Automation,Intelligence,Computing,and Networking(ISAICN 2019).Institute of Management Science and Industrial Engineering:计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society),2019:139-143.

[28]Jason Axelrod. Key automation practices government offices need to implement[J]. The American City &amp; County,2019.

[29]Zhe Wang. The realization of background management system based on SSM[C]. International Informatization and Engineering Associations.Proceedings of 2019 2nd International Conference on Information Science and Electronic Technology(ISET 2019).International Informatization and Engineering Associations:计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society),2019:410-414.

[30]Yuxiang Hou. The design and implementation of the framework for Spring+SpringMVC+MyBatis in the development of Web application[C]. Institute of Management Science and Industrial Engineering.Proceedings of 2019 4th International Industrial Informatics and Computer Engineering Conference(IIICEC 2019).Institute of Management Science and Industrial Engineering:计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society),2019:369-374.

[31]Bhaskar Kumar, Kumar Abhishek, Akshay Deepak, M.P.Singh. Implementation of Interactive Real Time Online Co-shopping Using Push AJAX[J].Procedia Computer Science，2016, 89: 473-482.

[32]Ying Xin Liu, Xi Yuan Li. Design and Implementation of a Business Platform System Based on Java[J].Procedia Computer Science，2020, 166: 150-153.

[33]Byung Cheol Lee,Jangwoon Park,Heejin Jeong,Jaehyun Park. Validation of Trade-Off in Human–Automation Interaction: An Empirical Study of Contrasting Office Automation Effects on Task Performance and Workload[J]. Applied Sciences,2020,10(4).

附录A 外文翻译—原文部分

*Institute of Management Science and Industrial Engineering:计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society),2019:369-374.*

The Design and Implementation of the Framework for Spring+SpringMVC+MyBatis In the Development of Web Application

**Keywords:** Web application; SpringMVC; Spring; Mybatis; Framework

**Abstract**

In the development and design of Web application, the choice of framework development is very important. A good framework development can speed up the development of Web application, reduce development costs, and reduce the workload of developers. At the same time, Web application has good expansibility and portability. The framework based on SpringMVC + Spring + Mybatis (SSM) has good performance and rapid development efficiency, gradually become a mainstream development framework of Web application.

**1. The Introduction of Spring, SpringMVC, MyBati**

**1.1 Spring**

Spring is an open source framework, and Spring is a lightweight Java development framework that emerged in 2003. It is created to solve the complexity of enterprise application development. One of the main strengths for the framework is its layered architecture, which allows users to choose which component to use, at the same time, provides integrated frameworks for J2EE application development. Spring uses the basic JavaBean to accomplish things that could have been done by EJB only before. However, the use of Spring is not limited to server-side development. From the point of view of simplicity, testability and loose coupling, any Java application can benefit from Spring. The core of Spring is Inversion of Control (IoC) and Aspect Oriented Programming (AOP). In simple terms, Spring is a layered JavaSE/EEfull-stack (one-stop) lightweight open source framework.

**1.2 SpringMVC**

SpringMVC belongs to follow-up products of Spring framework, has been integrated into Spring Web Flow. The Spring framework provides an full-function MVC module for building Web applications. SpringMVC decouples controllers, model objects, dispatchers, and the roles of handler objects, which made them easier to customize. SpringMVC is a powerful and flexible web framework provided by Spring. With annotations, SpringMVC provides almost POJO development patterns which made the development and testing of the controller easier. This kind of controllers normally does not process the requests directly, instead, delegates these requests to other beans in the Spring context, these beans are injected into the controller through the dependency injection function of Spring. SpringMVC is mainly composed of DispatcherServlet, processor mapping, processor (controller), view parser and view. It’s two cores as below: Processor mapping: choose which controllers to use to process the requests View parser: choose the results how to be rendered Through above two points, SpringMVC guarantees how to select, control and process the requests, and how to select views to show loose coupling between outputs.

**1.3 MyBatis**

MyBatis is an open source project iBatis of apache, this project was moved to Google code from apache software foundation in 2010, and renamed MyBatis.

MyBatis is an excellent persistence framework that supports general SQL queries, stored procedures and advanced mapping. MyBatis eliminates almost all JDBC code, the manual setup of parameters and the retrieval of result sets. MyBatis uses simple XML or annotations for configuration and original mapping, to map the interface and Java's POJOs (Plain Old Java Objects, normal Java objects) into the records of the database.

Each MyBatis application program is mainly used with the SqlSessionFactory instance, and a SqlSessionFactory instance can be obtained through the SqlSessionFactoryBuilder. The SqlSessionFactoryBuilder can be obtained from an xml configuration file or a predefined configuration instance.

Building an SqlSessionFactory instance with an xml file is a very simple thing. It is recommended to use the classpath resource in this configuration, but you can use any Reader instance, including instances created with file paths or url at the beginning of file://. MyBatis has a utility class----Resources, which has a number of methods, can easily load Resources from class paths and other locations.

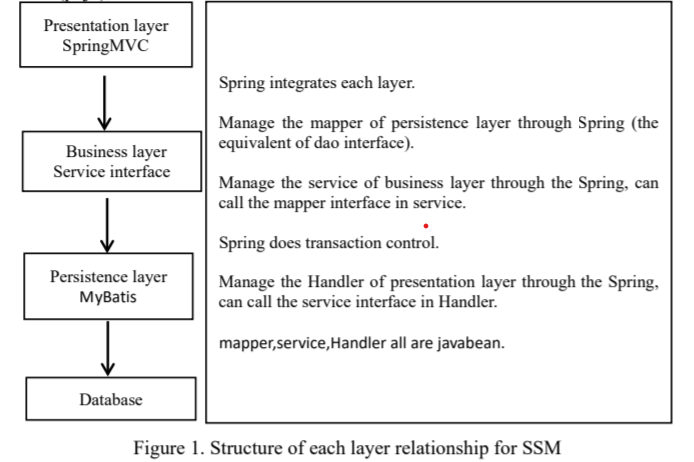
**2. The integration of SSM framework**

**2.1 SSM framework**

The SSM framework consists of three open source frameworks of Spring, SpringMVC and MyBatis, which are often used as a framework for web projects which the data source is simpler.

**2.2 Relationship between the various layers of SSM**

SpringMVC is a control layer, spring is used to manage the business logic layer. Mybatis is used for the dao layer. In the architecture MVC, m refers to the model, which contains the service, dao and javabean (pojo), v refers to the view.



The DAO layer (mapper) of persistent layer is invoked through MyBatis. DAO layer mainly do the work of data persistence layer, some of tasks responsible for contacting the database are encapsulated here, the design of the DAO layer first is to design the interface of DAO, and then to define implementation class of this interface in the configuration file of Spring, then can call this interface in the module to process the business of data, and don't need to care about the concrete implementation class of this interface is which class, structure seems to be very clear, the data source configuration of DAO layer and the parameters relating to database connections are configured in the configuration file of Spring.

The Service layer of Business layer: the Service layer is mainly responsible for the logical application design of business modules. First, design the interface, then design its implementation class, next, configure its implementation association in the configuration file of Spring. In this way, we can invoke the Service interface in the application to do business processing. Business implementation in the Service layer, in particular, need to call the defined interface in DAO layer, encapsulating the business logic of the Service layer is conducive to the independence and reuse of the general business logic, the procedure is shown very simple.

Controller layer (Handler layer): the Controller layer is responsible for the control of specific business module processes, in this layer, need to invoke the interface in Service layer to control the business process, the configuration of control is also processed in the configuration file of Spring, for the specific business process, there will be a different controller, in our specific design process, the process can be abstracted and summarized, to design the sub-unit process module which can be reused, this way not only makes the program structure clear, also greatly reduces the amount of code.

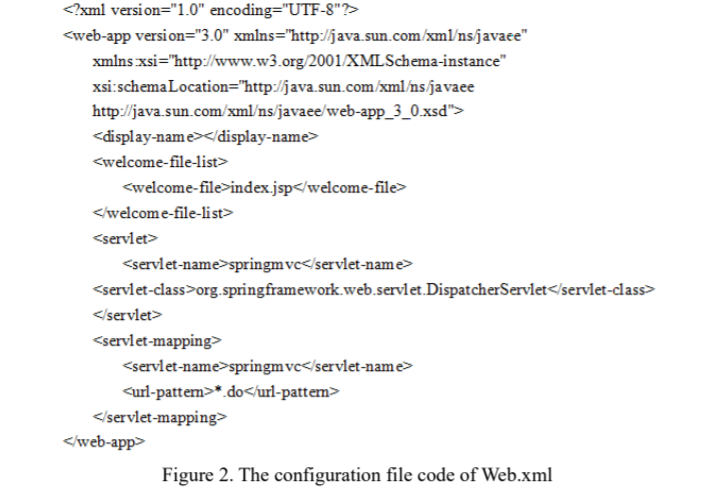
View layer: this layer is tightly coupled with the control layer, and requires the combination of the two to work together. The View layer is mainly responsible for the presentation of the foreground JSP page.

The links of each layer as follows:

The DAO layer and Service layer can be independently developed, their mutual coupling is very low, and can be carried out independently, such a pattern is especially advantageous in the development of large projects.

The Controller layer and View layer because of the high degree of coupling, so need to be developed together, but it can also be considered as a whole which is independent of the first two layers to be developed. In this way, we only need to know the definition of the interface before the layer and layer, and only need to call the interface, then can complete the necessary logic unit application, which is very clear and simple.

The design of Service logic layer is based on DAO layer, after set up the DAO layer, then can set up the Service layer, and Service layer is under the Controller layer, so Service layer not only need to invoke the interface of DAO layer, but also need to provide the interface for the class of Controller layer to make calls, it is just in a middle position. Each model has a Service interface, each interface encapsulates the respective business processing methods.



**2.3 The design and implementation of SSM framework**

The design and implementation of Spring+SpringMVC+MyBatis in the development of Web application, the design as follows:

application, the design as follows:

①In the development of web system, first to configure the file environment of web.xml, springmvc-servlet.xml.

The configuration file code of Web.xml is shown in Figure 2.

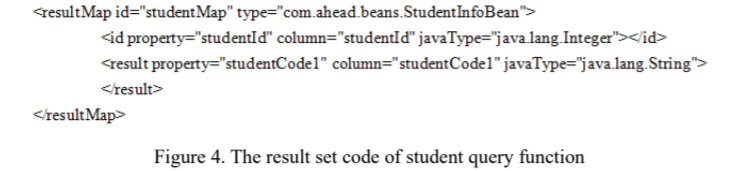
The configuration file of springmvc-servlet.xml is shown in Figure 3.



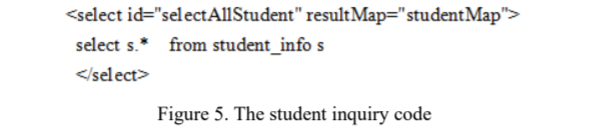
② In the development of web system, to design the entity class, to define the properties of object, properties has the methods of get and set (can refer to the field of the table in the database to set it, the database should be designed before all the coding starts), to create construction methods.

Under the directory of SRC, the design of configuration file for mysql.properties database which has four properties of jdbc.driver, jdbc.url, jdbc.usernamejdbc, password.

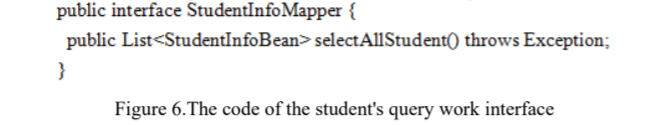
③ Then, to design the Mapper.xml, first to customize it to return to the result set, the id properties in various labels must be the same as the methods in the interface, and the id property value must be unique, and cannot be reused. Take the student to query the web page design as an example, the result set code of student query function is shown in Figure 4.



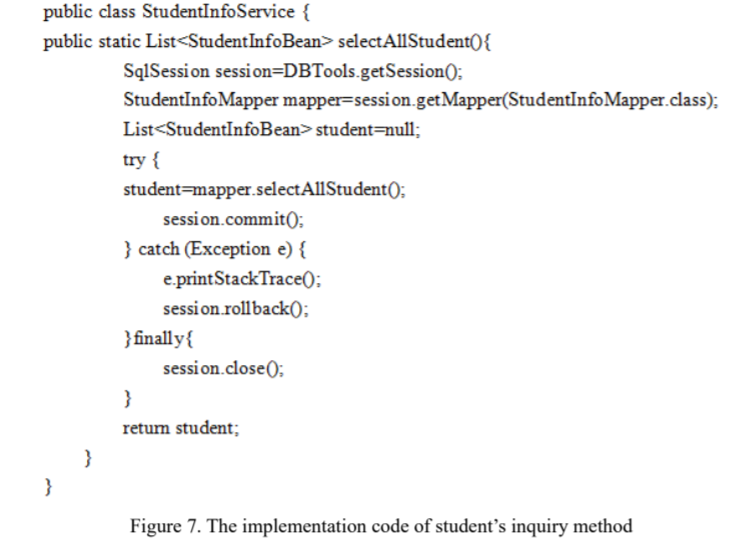
④ Then to define the function of the web system, to correspond to the operations that are performed on the database, such as insert, selectAll, selectByKey, delete, update, and so on. The student inquiry code is shown in Figure 5.



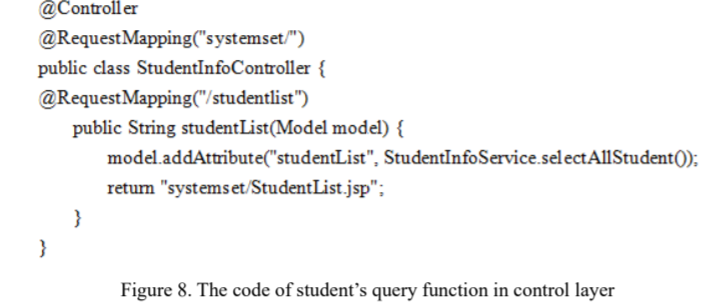
⑤ The design of Mapper.java, to map the operations in Mapper.xml to Java functions according to id. The code of the student's query work interface is shown in Figure 6.



⑥ The design of service.java, to provide services for the control layer, accept the parameters of 363 the control layer, complete the corresponding functions, and return to the control layer. The implementation code of student’s inquiry method is shown in Figure 7.

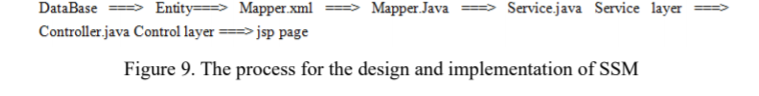


⑦ The design of Controller.java, to connect page requests and service layer, obtain the parameters of the page request, through the automatic assembly, to map a different URL to the corresponding processing function, and obtain parameters, process the parameters, and then pass it to the service layer. The code of student’s query function in control layer is shown in Figure 8.



⑧ Finally, to design the JSP page invoking, what parameters are requested and what data is required.

In a word, the process for the design and implementation of SSM is shown in Figure 9.



附录B 外文翻译—译文部分

计算机科学与电子技术国际学会, 2019年，第369-374页

Web应用程序开发中Spring + SpringMVC + MyBatis框架的设计与实现

**摘要**

在Web应用程序的开发和设计中，框架开发的选择非常重要。 一个好的框架开发可以加速Web应用程序的开发，降低开发成本，并减少开发人员的工作量。 同时，Web应用程序具有良好的可扩展性和可移植性。 基于SpringMVC + Spring + Mybatis（SSM）的框架具有良好的性能和快速的开发效率，逐渐成为Web应用程序的主流开发框架。

**1. Spring，SpringMVC，MyBati的介绍**

**1.1 Spring**

Spring是一个开源框架， 2003年出现的轻量级Java开发框架。它是为解决企业应用程序开发的复杂性而创建的。 该框架的主要优势之一是其分层体系结构，该体系结构允许用户选择要使用的组件，同时为J2EE应用程序开发提供集成的框架。 Spring使用基本的JavaBean来完成以前只有EJB才能完成的工作。 但是，Spring的使用不仅限于服务器端开发。 从简单性，可测试性和松散耦合的角度来看，任何Java应用程序都可以从Spring中受益。 Spring的核心是控制反转（IoC）和面向方面的编程（AOP）。 简单来说，Spring是一个分层的JavaSE / EEfull-stack（一站式）轻量级开源框架。

**1.2 SpringMVC**

SpringMVC属于Spring框架的后续产品，已集成到Spring Web Flow中。 Spring框架提供了用于构建Web应用程序的全功能MVC模块。 SpringMVC使控制器，模型对象，调度程序和处理程序对象的角色分离，这使它们更易于自定义。 SpringMVC是Spring提供的强大而灵活的Web框架。通过注释，SpringMVC提供了几乎POJO开发模式，这使得控制器的开发和测试更加容易。这种控制器通常不直接处理请求，而是将这些请求委托给Spring上下文中的其他bean，这些bean通过Spring的依赖项注入功能注入到控制器中。 SpringMVC主要由DispatcherServlet，处理器映射，处理器（控制器），视图解析器和视图组成。它是两个核心，如下所示：处理器映射：选择用于处理请求的控制器视图解析器：选择结果的呈现方式通过以上两点，SpringMVC保证了如何选择，控制和处理请求以及如何选择视图显示输出之间的松散耦合。

**1.3 MyBatis**

MyBatis是apache的开源项目iBatis，该项目于2010年从apache软件基金会移至Google代码，并更名为MyBatis。  
 MyBatis是一个出色的持久性框架，它支持常规SQL查询，存储过程和高级映射。 MyBatis消除了几乎所有的JDBC代码，参数的手动设置和结果集的检索。 MyBatis使用简单的XML或注释进行配置和原始映射，以将接口和Java的POJO（普通的Java对象，普通的Java对象）映射到数据库的记录中。  
 每个MyBatis应用程序主要与SqlSessionFactory实例一起使用，并且可以通过SqlSessionFactoryBuilder获得SqlSessionFactory实例。可以从xml配置文件或预定义的配置实例中获取SqlSessionFactoryBuilder。  
用xml文件构建SqlSessionFactory实例是一件非常简单的事情。建议在此配置中使用类路径资源，但是您可以使用任何Reader实例，包括使用file：//开头的文件路径或url创建的实例。 MyBatis有一个实用程序类-资源，它具有许多方法，可以轻松地从类路径和其他位置加载资源。

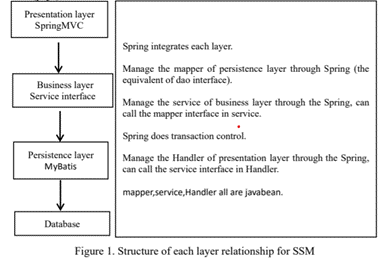
**2. SSM框架的整合**

**2.1 SSM 框架**

SSM框架由Spring，SpringMVC和MyBatis的三个开源框架组成，它们经常用作Web项目的框架，其数据源更简单。

**2.2 SSM各层之间的关系**

SpringMVC是控制层，spring用于管理业务逻辑层。 Mybatis用于dao层。 在MVC体系结构中，m表示包含服务的模型，dao和javabean（pojo），v表示视图。



持久层的DAO层（映射器）通过MyBatis调用。 DAO层主要负责数据持久层的工作，这里负责封装一些与数据库联系的任务，DAO层的设计首先是设计DAO的接口，然后在配置中定义该接口的实现类。 文件，然后可以在模块中调用此接口来处理数据业务，而不必关心该接口的具体实现类是哪个类，结构似乎很清楚，数据源的配置 在Spring的配置文件中配置DAO层和与数据库连接有关的参数。

业务层的服务层：服务层主要负责业务模块的逻辑应用程序设计。首先，设计接口，然后设计其实现类，其次，在Spring的配置文件中配置其实现关联。这样，我们可以在应用程序中调用Service接口来进行业务处理。特别是在Service层中的业务实现，需要在DAO层中调用已定义的接口，封装Service层的业务逻辑有利于通用业务逻辑的独立性和重用，程序显示非常简单。

控制器层（Handler layer）：控制器层负责控制特定业务模块流程，在这一层中，需要调用服务层中的接口来控制业务流程，控件的配置也在配置文件中进行处理Spring的特定业务流程中会有一个不同的控制器，在我们的特定设计流程中，可以对流程进行抽象和总结，以设计可重用的子单元流程模块，这种方式不仅使程序结构清晰，也大大减少了代码量。

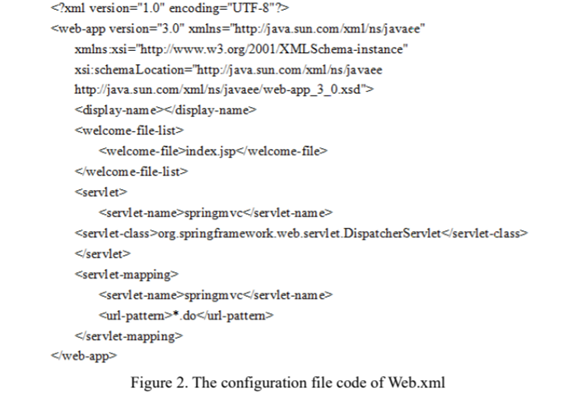
视图层：此层与控制层紧密耦合，并且需要两者结合才能一起工作。 View层主要负责前台JSP页面的呈现。

每层的链接如下：

DAO层和Service层可以独立开发，它们之间的相互耦合非常低，并且可以独立执行，这种模式在大型项目的开发中特别有利。

Controller层和View层由于高度耦合，因此需要一起开发，但也可以作为一个整体来考虑，它独立于要开发的前两层。这样，我们只需要在层和层之前知道接口的定义，并且只需要调用接口，就可以完成必要的逻辑单元应用程序，这非常清楚和简单。

服务逻辑层的设计是基于DAO层的，在建立DAO层之后，便可以建立Service层，而Service层在Controller层之下，因此Service层不仅需要调用DAO层的接口，但是还需要为Controller层的类提供接口以进行调用，它只是处于中间位置。每个模型都有一个Service接口，每个接口都封装了各自的业务处理方法。



**2.3 SSM框架的设计与实现**

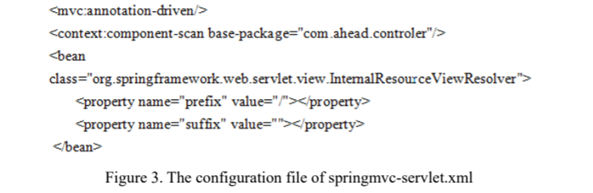
Spring + SpringMVC + MyBatis在Web应用程序开发中的设计与实现，设计如下：

应用程序的设计如下：

①在Web系统的开发中，首先要配置web.xml，springmvc-servlet.xml的文件环境。

Web.xml的配置文件代码如图2所示。

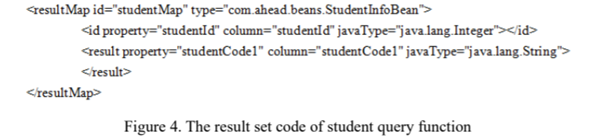
springmvc-servlet.xml的配置文件如图3所示。



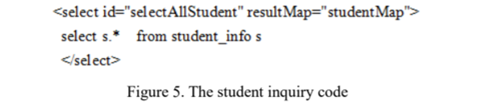
②在Web系统的开发中，要设计实体类，定义对象的属性，属性具有get和set的方法（可以参考数据库中表的字段进行设置，应设计数据库 在所有编码开始之前），以创建构造方法。

在SRC目录下，为mysql.properties数据库配置文件的设计，该文件具有jdbc.driver，jdbc.url，jdbc.usernamejdbc和password的四个属性。

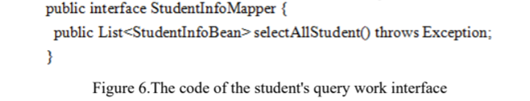
③然后，要设计Mapper.xml，首先要对其进行自定义以返回到结果集，各个标签中的id属性必须与接口中的方法相同，并且id属性值必须唯一，并且不能为 重用。 以学生查询网页设计为例，学生查询功能的结果集代码如图4所示。



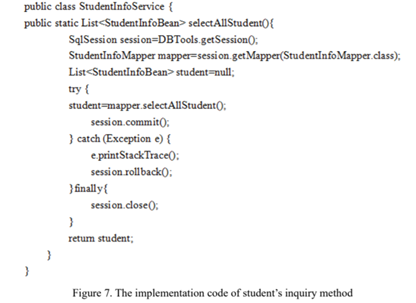
④然后定义Web系统的功能，使其与在数据库上执行的操作相对应，例如insert，selectAll，selectByKey，delete，update等。 学生查询代码如图5所示。



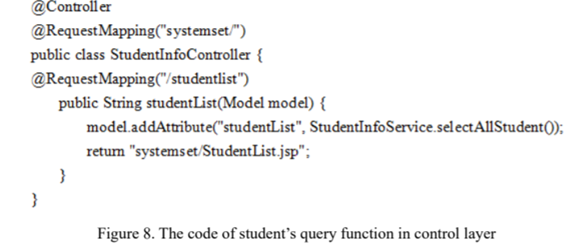
⑤Mapper.java的设计，根据id将Mapper.xml中的操作映射到Java函数。 学生的查询工作界面的代码如图6所示。



⑥service.java的设计，为控制层提供服务，接受363控制层的参数，完成相应的功能，并返回控制层。 学生查询方法的实现代码如图7所示。

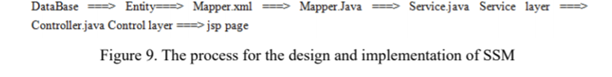


⑦ Controller.java的设计，用于连接页面请求和服务层，通过自动组装获得页面请求的参数，将不同的URL映射到相应的处理函数，并获取参数，处理参数，然后传递 它到服务层。 控制层中学生查询功能的代码如图8所示。



⑧ 最后，要设计JSP页面调用，需要请求什么参数和需要什么数据。

简而言之，SSM的设计和实现过程如图9所示。



附录C 主要源代码

（1）用户登录核心代码

|  |
| --- |
|  |

（2）用户注册核心代码

|  |
| --- |
|  |

（3）删除用户信息

|  |
| --- |
|  |

（4）修改用户信息核心代码

|  |
| --- |
|  |

（5）申请角色核心代码

|  |
| --- |
|  |

（6）获取已发邮件核心代码

|  |
| --- |
|  |

（7）获取已收邮件核心代码

|  |
| --- |
|  |

（8）查看邮件内容核心代码

|  |
| --- |
|  |

（9）删除邮件核心代码

|  |
| --- |
|  |

（10）单独发送邮件核心代码

|  |
| --- |
|  |

（11）群发邮件代码核心代码

|  |
| --- |
|  |

（12）获取所有通讯录信息核心代码

|  |
| --- |
|  |

（13）管理通讯录信息核心代码

|  |
| --- |
|  |

（14）工作流管理核心代码

|  |
| --- |
|  |