密级：



**NANCHANG UNIVERSITY**

**学士学位论文**

**THESIS OF BACHELOR**

（2015—2019年）



题目 合同管理系统的设计与实现

学院：软件学院系软件工程

专业班级：Java(卓越千峰)153班

学生姓名：张志勇学号：8000115210

指导教师：曾长清职称：副研究员

起讫日期：

**南昌大学**

**学士学位论文原创性申明**

本人郑重申明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式表明。本人完全意识到本申明的法律后果由本人承担。

作者签名：日期：

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权南昌大学可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密□，在年解密后适用本授权书。

本学位论文属于

不保密□。

（请在以上相应方框内打“√”）

作者签名：日期：

导师签名：日期：

# **摘要**

专业：软件工程学号：8000115210

姓名：张志勇指导教师：曾长清

本系统解决企业实际业务管理中的问题，专注于企业合同管理的动态化、智能化、规范化，用计算机技术动态解决传统合同管理效率低下、费时费力等问题，为企业提供安全、稳定、高效的合同管理功能,包括合同的在线起草、签订、进度查询和执行监测服务,可以基于合同关键条款对合同履行情况进行监控和预警。本系统可以在线编辑合同、实现合同的申报、审批、执行、可以上传合同相关附件并进行管理、可以按合同的类型查询合同、可以对合同的关键信息进行结构化管理。

本系统基于SSM框架进行开发，其中用户权限控制通过shiro框架实现，用户在线预览编辑word文档通过pageoffice实现，用户动态导出word模板通过freemarker实现。前端采用目前较为流行的Bootstrap UI，表单验证采用bootstrapValidator，数据库选择MySQL数据库。

**关键词：**eclipse;mysql;shiro;pageoffice;springmvc;spring;mybatis;合同管理；

Abstract

This system to solve problems in enterprise's actual business management,focus onthe dynamic,intelligent,standardization of enterprise contract management,using computer technology to solve the traditional dynamic problems such as inefficient and time-consuming to contract management,for the enterprise to provide a safe,stable and efficient contract management functions,including online drafting,signing contract,schedule query and perform monitoring service,can be based on the key terms of the contract performance of the condition monitoring and early warning.This system can edit the contract online,and realize the application,approval and execution of the contract.It can upload the contract attachments and manage them.It can query the contract according to the type of the contract,and it can conduct structured management of the key information of the contract.

This system is developed based on SSM framework,in which user rights control is realized through shiro framework,user online preview and edit word document is realized through pageoffice,and user dynamic export word template is realized through freemarker.The current popular Bootstrap UI is adopted for the front end,the bootstrapValidator is adopted for the form validation,and the MySQL database is selected for the database.

Keywords:eclipse;Mysql;Shiro.Pageoffice;springmvc;spring;Mybatis;Contract management;

# 目录

摘要 I

目录 III

第一章绪论 1

1.1选题依据及意义 1

1.1.1选题依据 1

1.2国内外现状及发展趋势 2

1.3论文章节安排 3

第二章需求分析 4

2.1功能需求 4

2.1.1系统简介 4

2.1.2系统角色 4

2.2运行环境需求 5

第三章系统总体设计 6

3.1系统功能结构 6

3.2关键技术选择 6

（1）前端 6

（2）后台 7

第四章数据库设计 11

4.1概念结构设计 11

4.2数据库逻辑设计 12

第五章系统功能设计 16

5.1登录模块 16

5.2合同发起模块 17

5.3合同审批模块 18

5.4进度查询模块 19

5.5已办事项查询模块 20

5.6用户管理模块 20

5.7修改个人信息模块 21

5.8附件管理模块 21

第六章系统实现 23

6.1登录模块 23

6.2合同发起模块 26

6.3合同审批模块 29

6.4进度查询模块 32

6.5已办事项查询模块 34

6.6用户管理模块 35

6.7附件管理模块 38

第七章总结与展望 40

参考文献 41

致谢 42

# 第一章绪论

## 1.1选题依据及意义

1.1.1选题依据

市场经济在某种意义上说是法制经济，再贴近些说，是契约经济。企业的经济往来，主要是通过合同形式进行的。所以，一个企业的经营成败和合同及合同管理有密切关系。因此，必须十分重视合同及合同管理。

合同早在古罗马时代就受到了法学家的关注，在当代对企业也至关重要。合同是企业对外开展业务的主要依据凭证，无处不在，合同管理水平是企业综合管理水准的集中体现。合同管理不是简单的要约、承诺，而且是全方位、全过程的动态管理，合同行为贯穿了企业管理活动的始终，如何管好合同，经常是困扰管理者的难题。

合同的执行过程可以划分为四个阶段，即：合同准备、合同签署、合同管理和合同终止。合同管理必须是全过程的、系统性的、动态性的。全过程就是由洽谈、草拟、签订、生效开始，直至合同失效为止。我们不仅要重视签订前的管理，更要重视签订后的管理。系统性就是凡涉及合同条款内容的各部门都要一起来管理。动态性就是注重履约全过程的情况变化。

1.1.2选题意义

采用手工管理合同，由于涉及的部门众多，需要管理的合同要素也各不相同，因此造成信息不集中，实时性不强，导致各部门协作，业务流程组建，监控制度执行方面效率不高，费时费力等问题。具体表现有文档管理困难、进度控制困难、信息汇总困难、缺少预警机制。

完善的合同管理是公司健康运作的一个重要标志。然而，完善的合同管理需要公司许多的资源，如何简化合同的管理而不失其完整、科学是许多公司头痛的问题。并且，结合中国公司的实际情况，将客户管理和项目管理也融合到合同管理中，真正体现优化公司资源的目的。自动化的合同结算提醒有效的保护了公司的资金运营安全。统计功能使拖放字段的方式即可实现不同的统计，真正简化了烦琐的统计过程，而且非常高效。

合同管理系统可以为企业管理带来很多价值：规范基础数据管理、提高管理效率、实现标准化管理和个性化管理的有机结合、为领导决策提供准确及时广泛的信息、为用户提供全面合同管理解决方案。总之，通过合同管理系统的应用和实施，可以为合同管理信息化和合同管理能力提升提供一个最佳的结合点，切实打造企业核心竞争力。

## 1.2国内外现状及发展趋势

西方发达国家合同管理在理论和实践上已经远远超越中国，形成一种制度准则和形式规范。美国在1959年就成立了国家合同管理协会NCMA[[[1]](#footnote-0)]，有些州甚至提供专门的合同管理指南供企业参考。例如美国科罗拉多州人事行政部门提供的合同管理指南就详细讲述了合同管理的七部分内容：“①合同的执行：合同准备、合同分析、合同类型选择、合同付款、合同执行文件的建立、合同执行计划；②日常合同执行：合同执行的基本责任、工作计划和步骤、合同效力检查、监控和接受、国家所有权管理、债权管理和财务管理；③合同变更：变更基础、变更案例、采购规则和变更批准、其他变更、价格变更、合同主体名称变更和合同主体变更协议；④转包合同和权利义务分配；⑤纠纷解决：纠纷解决的责任、谈判

、其他纠纷解决方式、纠纷解决的执行；⑥合同效力的补救和终止；⑦合同结束。”[[[2]](#footnote-1)]

石振武在《道路经济与管理》中提到，合同管理是当事人双方或数方确定各自权利和义务关系的协议，虽不等于法律，但依法成立的合同具有法律约束力，工程合同属于经济合同的范畴，受经济和刑法法则的约束，合同管理主要是指项目管理人员根据合同进行工程项目的监督和言理，是法学、经济学理论和管理科学在组织实施合同中的具体运用[[[3]](#footnote-2)]。

目前我国的技术合同管理制度包括技术合同的认定登记制度。技术保密制度、技术市场管理制度、鉴证制度、公证制度以及合同管理机关对危害国家利益、社会公共利益的管理制度等。在上述制度中，技术保密制度、对利用合同危害国家利益、社会公共利益的违法行为监督管理制度是强制性的制度，其他制度都是引导性的制度，引导技术合同行为符合市场经济的要求，促进市场经济的发展[[[4]](#footnote-3)]。

国内对合同管理的研究目前主要集中在建筑、工程、招投标、劳动合同、项目合同的合同管理方面，如尹贻林在《合同法与工程合同管理》中以合同法和招投标的法律知识为背景，分析了所涉及到的勘察设计合同、施工合同、监理合同等合同的管理，并提出了关于工程合同的风险管理[[[5]](#footnote-4)]。如梁世林、裴恩才在《工程项目管理学》中从工程管理角度，研究了工程项目投资前期、投资建设期直至工程项目投产整个过程的计划、组织、指挥、决策、控制及协调的理论、方法与手段[[[6]](#footnote-5)]。如陈祥锋在《供应链合同管理模型与应用研究》对企业供应链中的合同管理模型和应用问题进行了系统的研究。[[[7]](#footnote-6)]国内目前对其他行业的合同管理研究尚很欠缺，并且国内也没有专门的政策法规对企业合同管理进行指导和规范。未来我国的合同管理研究将逐渐向更大范围的企业合同管理扩展，并将逐步开始研究如何规范操作流程，如何利用定量研究方法来精确设计合同。随着我国对合同管理的日益重视，我国企业的合同管理制度也将得到进一步的完善。

## 1.3论文章节安排

本论文会分成六章的内容去详述项目协作管理系统

第一章：绪论部分。详细介绍了改选题的依据及意义，说明选题的背景是什么，还有国内外现状及发展趋势。并在第三小节介绍了系统的架构及项目应用的技术，原因以及好处。

第二章：需求分析部分。简要介绍系统开发背景。详细分析了该系统的功能需求、运行环境以及主要开发技术。

第三章：系统设计部分。详细介绍系统的功能模块。

第四章：数据库设计实现部分。详细介绍该系统的数据库结构如字段，字段含义等

第五章：系统实现部分。根据需求及系统设计完成系统功能的实现。

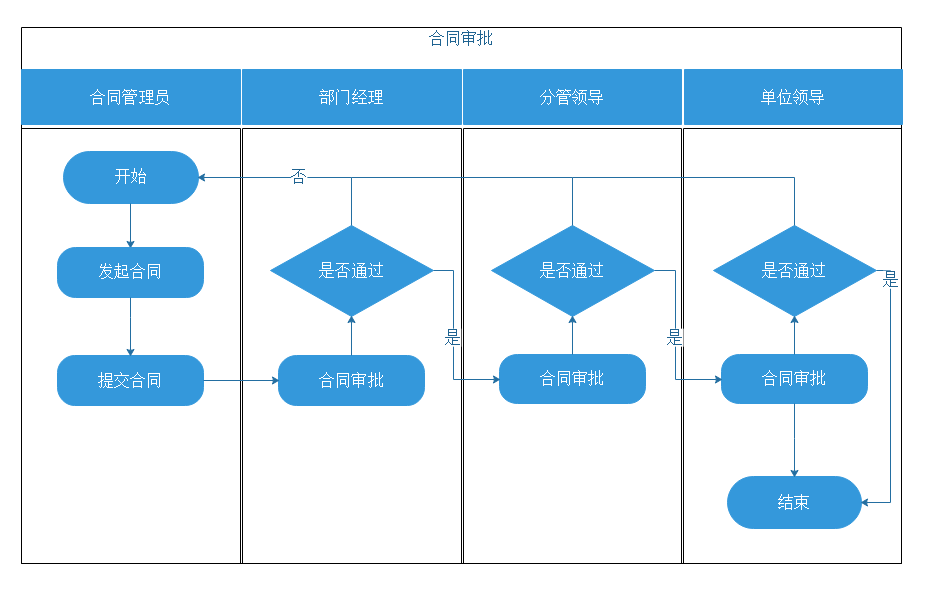
第六章：结论部分。总结自己在实现系统功能的过程中所遇到的问题以及解决思路，获得的收获以及完成整个项目的自我评价

# 第二章需求分析

## 2.1功能需求

### 2.1.1系统主要功能描述

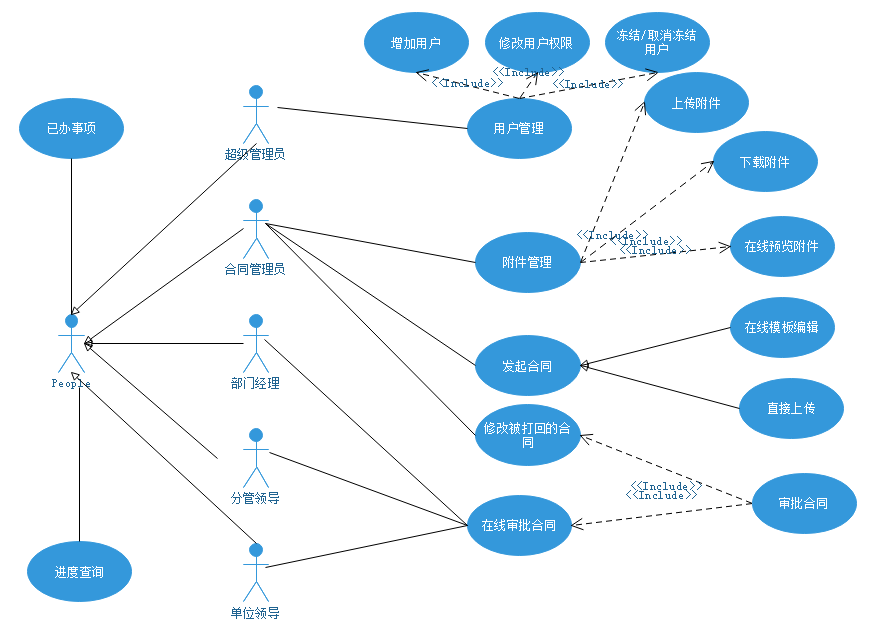
本系统解决企业实际业务管理中的问题包括合同的在线起草、签订、进度查询和执行监测服务,可以基于合同关键条款对合同履行情况进行监控和预警。本系统可以在线编辑合同（在线模板编辑、上传合同）、实现合同的申报、审批(三级)、执行、可以上传合同相关附件并进行管理、可以按合同的类型、合同号、合同名称查询合同、可以对合同的关键信息进行结构化管理。用户可以登录、找回密码。同时也包括用户的管理，包括用户的注册、用户权限的修改、用户的冻结、解除冻结。



**图2.1合同审批泳道图**

### 2.1.2系统角色

该合同管理系统主要角色分为合同管理员、部门经理、分管领导、单位领导、超级管理员5个角色。合同管理员可以在线发起合同、查询合同进度状态、处理被打回的合同、修改个人信息、管理合同附件。部门经理可以审批由合同管理员发起的合同、查询合同进度状态、修改个人信息分管领导、单位领导的权限基本跟部门经理的权限一样，只不过单位领导作为最高级别的审批人，审批通过后，该合同的审批流程立即结束（即该合同可以下载）。超级管理员可以对用户、角色、权限进行管理，实现对用户表、角色表、权限表进行增删改查。

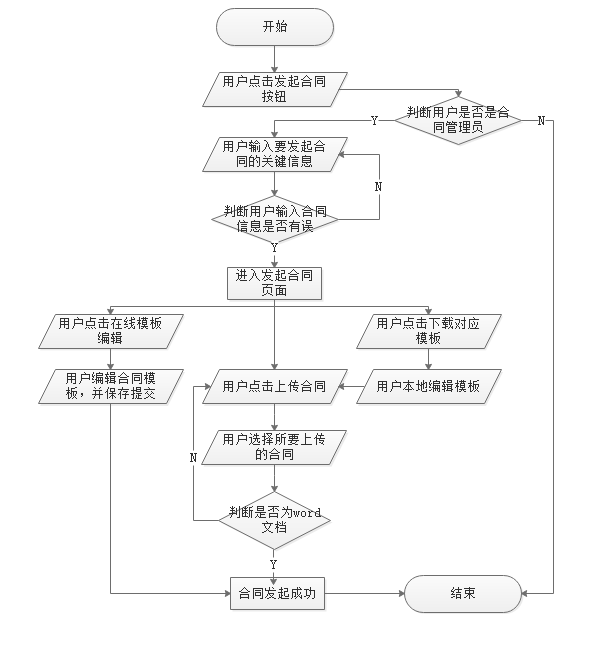


**图2.2系统用例图**

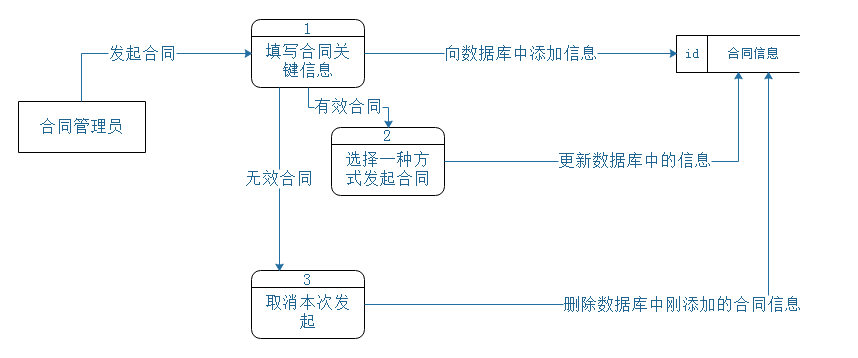
## 2.2系统主要业务逻辑

2.2.1合同发起模块

该模块只能由合同管理员进行操作，用户登录成功后，点击合同发起按钮，如果是合同管理员，则进入合同发起界面，如果是其他角色，则提示没有该权限。在进入合同发起界面前，合同管理员需要输入一些合同的关键信息（比如合同名、合同编号、合同类别、甲方公司名、合同金额）进行提交，提交成功后方可进入合同发起页面，提交失败，则会弹出提交失败的信息。图2.3展示了发起合同的流程据流。图2.4展示了发起合同的数据流图。



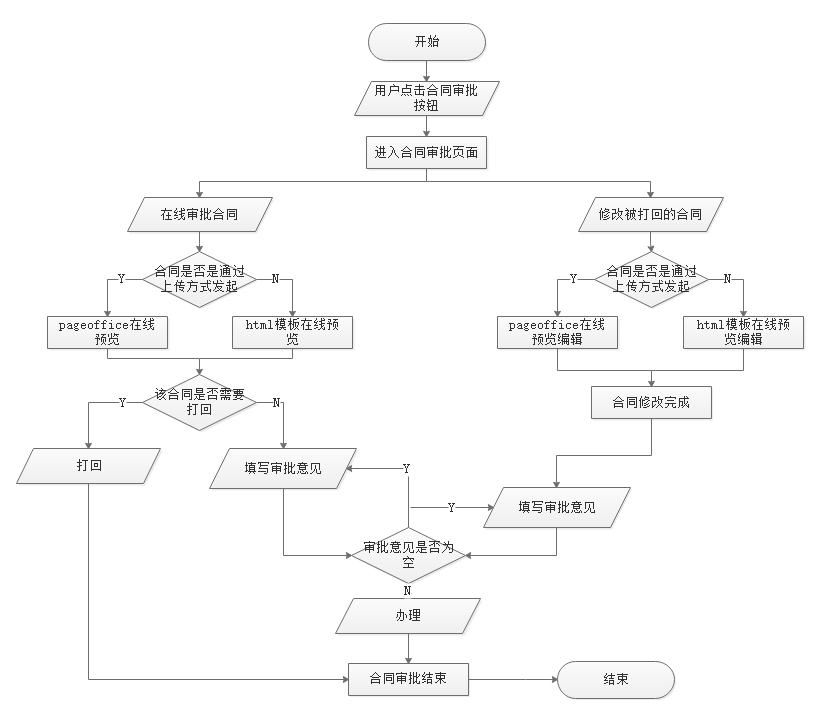
**图2.3发起合同流程图**



**图2.4发起合同数据流图**

2.2.2合同审批模块

该模块会通过用户的身份进行判断该合同是否可以编辑。合同管理员可以在线编辑被打回的合同，而其他角色只能在线预览，不能编辑。而且在合同审批模块中，要根据上传合同的方式，来进行合同的在线预览。如果合同管理员是以上传的方式发起的合同，那么其他审批人员通过pageoffice只读模式进行在线预览，如果合同管理员是以在线模板编辑的方式发起的合同，那么其他审批人员通过html模板的只读模式进行在线预览。图2.4展示了合同审批的流程图。

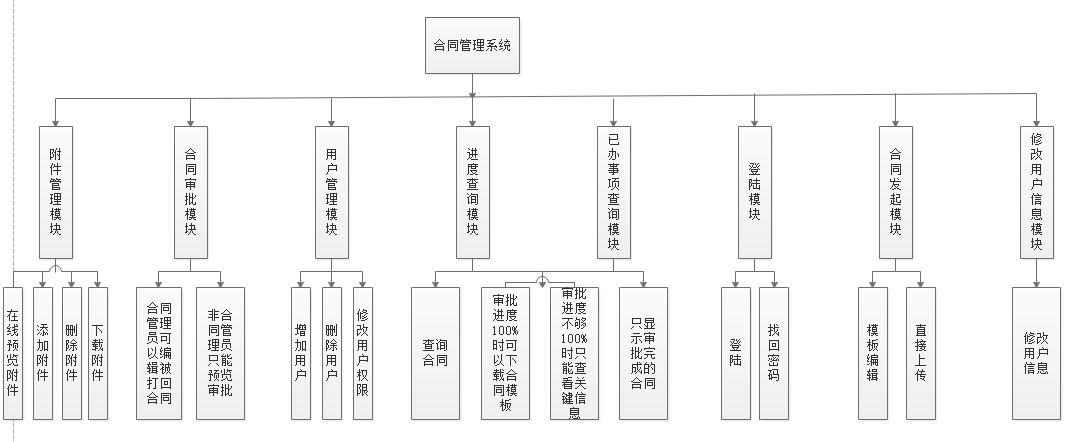


**图2.4合同审批流程图**

# 第三章系统总体设计

## 3.1系统功能结构

本系统中存在合同管理员、部门经理、分管领导、单位领导、超级管理员五个角色对系统进行操作。系统主要分为登陆模块、合同发起模块、合同审批模块、进度查询模块、已办事项查询模块、用户管理模块、修改用户信息模块、附件管理模块这八大模块。其中登陆模块包括用户找回密码，合同发起模块又可以细分为上传合同和合同模板在线编辑，合同审批模块根据用户角色分为只能预览审批（除合同管理员外的所有角色）和可以修改合同文档（只有合同管理员拥有该权限），进度查询模块跟已办事项查询模块则需要根据模板审批进度来决定该合同可不可以下载，用户管理模块则可以对用户进行角色的修改，从而修改用户的权限，也可以对用户进行增删改查。合同管理员在附件管理模块可以上传合同附件，同时也可以删除附件。



**图3.1系统总体模块图示**

## 3.2关键技术选择

### （1）前端

**jQuery**：jQuery是一个JavaScript函数库。它的核心理念是"Write Less，Do More"。它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互。jQuery的核心特性可以总结为：具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的css选择器，并且可对CSS选择器进行扩展；拥有便捷的插件扩展机制和丰富的插件。jQuery兼容各种主流浏览器。所以在项目前端中运用该技术可以极大的减少JavaScript的开发量，但是却可以完成更多的效果。

**Bootstrap**：Bootstrap，来自Twitter，是目前最受欢迎的前端框架。Bootstrap是基于HTML、CSS、JAVASCRIPT的，它简洁灵活，使得Web开发更加快捷。Bootstrap提供了一个带有网格系统、链接样式、背景的基本结构。它自带全局的CSS设置、定义基本的HTML元素样式、可扩展的class。它还包含了十几个可重用的组件，用于创建图像、下拉菜单、导航、警告框、弹出框等等。同时Bootstrap还包含了十几个自定义的jQuery插件。

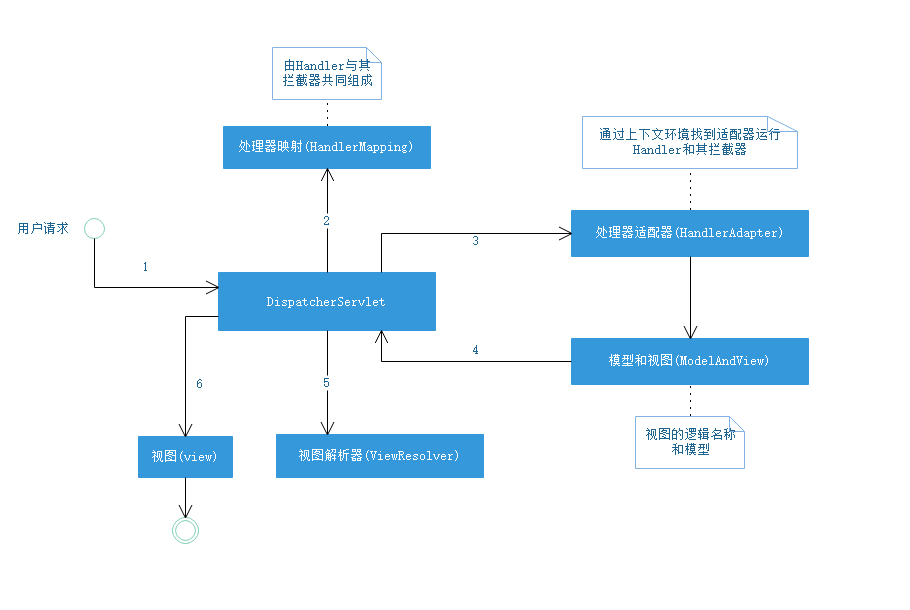
### （2）后台

**SSM框架**：SSM框架是spring MVC，spring和mybatis框架的整合，是标准的MVC模式，将整个系统划分为表现层，controller层，service层，DAO层四层。在SSM中spring Ioc承担了一个资源管理、整合、即插即拔的功能，Spring MVC用于把模型、视图和控制器分层，组成一个有机灵活的系统，Mybatis提供了一个数据库访问的持久层。

**Spring**：spring的成功来自于他的理念，并不是技术本身。spring的理念包括IoC（**Inversion of Control,控制反转**）和AOP（Aspect Oriented Programming，面向切面编程）。其中IoC是一个容器，用来管理bean及他们之间的关系。只是Spring IoC管理对象及他们之间的依赖关系不是人为主动创建的，而是由spring Ioc通过描述创建的。

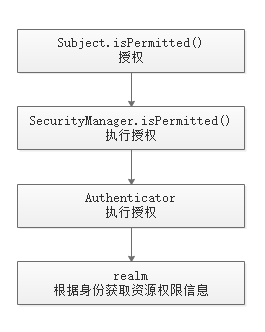
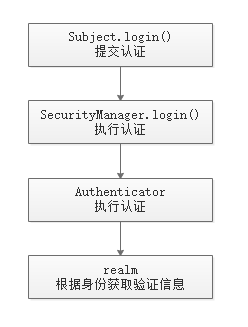
**Mybatis**：MyBatis是一个支持普通SQL查询，存储过程和高级映射的优秀**持久层框架**。MyBatis消除了几乎所有的JDBC代码和参数的手工设置以及对结果集的检索封装。MyBatis可以使用简单的**XML或注解**用于配置和原始映射，将接口和Java的POJO（Plain Old Java Objects，普通的Java对象）映射成数据库中的记录。Mybatis作为数据访问层DAO（Data Access Object）是不需要实现类的，它只需要一个接口和XML（或者注解）。Mybatis提供自动映射、动态SQL、级联、缓存、注解、代码和SQL分离等特性，同时也对SQL进行优化。

**Spring MVC**：SpringMVC代表了Model(模型)View(视图)Controller(控制)接收外部请求,进行分发和处理。



当一个请求到来时，DispatcherServlet首先通过请求和事先解析好的HandlerMapping配置，找到对应的处理器（Handler），然后调用处理器适配器（HanderAdapter）执行Handler。处理器执行完成后，会返回模型和视图给DispatcherServlet，然后DispatcherServlet就会把对应的视图信息传递给视图解析器（ViewResolver）。这一步取决于是否使用逻辑视图，如果是逻辑视图，name视图解析器就会解析它，然后把模型渲染到视图里。如果不是，则会直接通过视图渲染模型。

**Shiro框架**：Apache Shiro是Java的一个安全框架。Shiro可以帮助我们完成：认证、授权、加密、会话管理、与Web集成、缓存等。Shiro的主要功能是管理应用程序中与安全相关的全部，同时尽可能支持多种实现方法。Shiro是建立在完善的接口驱动设计和面向对象原则之上的，支持各种自定义行为。Shiro的四大基石：身份验证，授权，会话管理和加密。

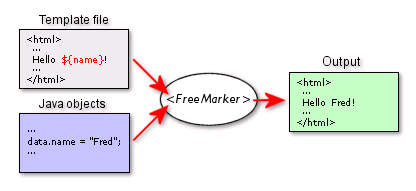


登录验证是调用subject的login()方法,调用login()方法时Shiro会自动调用自定义的MyRealm类中的doGetAuthenticationInfo()方法进行验证的，验证逻辑是先根据用户名查询用户，如果查询到的话再将查询到的用户名和密码放到SimpleAuthenticationInfo对象中，Shiro会自动根据用户输入的密码和查询到的密码进行匹配，如果匹配不上就会抛出异常，匹配上之后就会执行doGetAuthorizationInfo()进行相应的权限验证。

doGetAuthorizationInfo()方法会根据用户名获取到他所拥有的角色以及权限，然后赋值到SimpleAuthorizationInfo对象中即可，Shiro就会按照我们

配置的XX角色对应XX权限来进行判断断。

**Freemarker**：FreeMarker是一款用java语言编写的模版引擎，它虽然不是web应用框架，但它很合适作为web应用框架的一个组件。它是一个轻量级模版引擎，不需要Servlet环境就可以很轻松的嵌入到应用程序中，而且能生成各种文本，如html，xml，java，等。它入门简单，用java编写的，很多语法和java相似

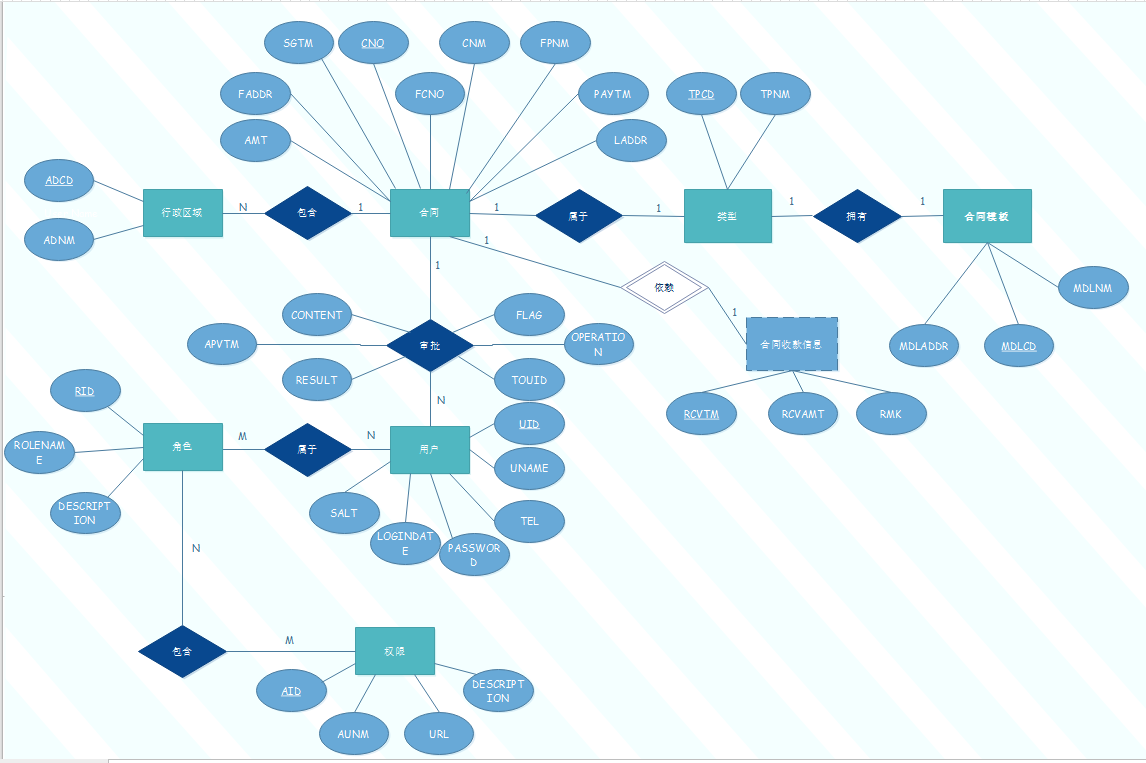


**PageOffice**：PageOffice是用来提供在线编辑Office文档、动态填充、动态提交功能的通用组件库。集成PageOffice不但能够实现在线编辑、保存Office格式的文档、强制痕迹保留，手写批注，圈阅签字，手写签名，电子印章、动态模板套红等在线办公的功能，而且还可以轻松实现Word/Excel动态数据填充，Word/Excel用户输入提交，Word/Excel/PowerPoint等Office文档的在线打开、编辑、保存，权限控制，只读控制，编辑区域控制等高级复杂的功能。

# 第四章数据库设计

## 4.1概念结构设计

系统的总体E-R图设计，如图4.1所示



**图4.1系统总体E-R图**

合同作为本系统的核心，其中合同中包含行政区域，是一对一的关系，即一个合同中包含一个行政区域。一个合同属于一个类型，所以合同跟合同类型也是一对一的关系，而一个合同类型对应一个合同模板，故合同类型跟合同模板也是一对一的关系。

一个合同可以被多个用户进行审批，所以合同跟用户是一对多的关系，在本系统中每个用户只能拥有一个角色，而一个角色可能对应多个用户，所以用户跟角色也是多对一的关系。一个角色可能拥有多个权限，一个权限也可能有多个角色适用，所以角色跟权限是多对多的关系。

## 4.2数据库逻辑设计

1. 用户表（USER\_INFO）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| UID | 用户id | C(8) | PK | N |  |
| UNAME | 用户昵称 | NVC(50) |  |  |  |
| TEL | 用户电话 | C(11) |  |  |  |
| PASSWORD | 用户登录密码 | C(32) |  |  |  |
| SALT | 盐值 | C(50) |  |  |  |
| LOGINDATE | 登录日期 | DATE |  |  |  |
| EMAIL | 用户邮箱 | C(50) |  |  |  |

说明：用户密码通过MD5加密，盐值为随机生成的字符串

1. 角色表（ROLES）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| RID | 角色编号 | C(8) | PK | N |  |
| ROLENAME | 角色名 | NVC(50) |  |  |  |
| DESCRIPTION | 角色描述 | NVC(100) |  |  |  |

1. 权限表（AUTHORITY）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| AID | 权限编号 | C(8) | PK | N |  |
| AUNM | 权限名 | NVC(50) |  |  |  |
| URL | Url | C(50) |  |  | 通过url控制权限 |
| DESCRIPTION | 权限描述 | NVC(100) |  |  |  |

1. 用户角色表（USERS\_ROLES）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| ID | 编号 | int | PK | N |  |
| RID | 角色编号 | C(8) |  |  |  |
| UID | 用户id | C(8) |  |  |  |

1. 角色权限表（ROLES\_AUTHORITY）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| ID | 编号 | int | PK | N |  |
| RID | 角色编号 | C(8) |  |  |  |
| AID | 权限编号 | C(8) |  |  |  |

1. 合同类型表（CNT\_TP）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| TPCD | 合同类型代码 | C(2) | PK |  |  |
| TPNM | 合同类型名称 | NVC(20) |  |  |  |
| MDLCD | 合同模板代码 | C(2) | FK |  | 该合同类型对应的合同模板 |

说明：

（1）TPCD，第1位为经营分类，S为销售合同；B为采购/购买合同，如：

S1-软件开发、S2-技术服务、S3-技术咨询、S4-产品销售、S5-系统集成

B1-设备采购、B2-服务采购

1. 合同模板表（CNT\_MDL）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| MDLCD | 合同模板代码 | C(2) | PK |  |  |
| MDLNM | 合同模板名称 | NVC(20) |  |  |  |
| MDLADDR | 合同模板地址 | VC(255) |  |  | 合同模板地址 |

说明：管理合同模板，如采购项目合同、开发项目合同、技术服务合同等，模板不同；合同模板供下载使用

1. 行政区划表（ADDV）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| ADCD | 行政区划代码 | C(6) | PK |  |  |
| ADNM | 行政区划名称 | NVC(50) |  |  |  |

1. 合同基本信息表（CNT\_INFO）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| CNO | 合同编号 | C(8) | PK | N | YYYYMMNN |
| TPCD | 合同类型 | C(2) | FK | N |  |
| FCNO | 销售合同编号 | C(8) | FK |  |  |
| CNM | 合同名称 | NVC(20) |  | N |  |
| FPNM | 甲方名称 | VC(100) |  |  |  |
| SGTM | 签订时间 | DATETIME |  |  |  |
| AMT | 合同金额 | N(11,2) |  |  |  |
| ADCD | 甲方所属行政区划 | C(6) | FK |  |  |
| PAYTM | 付款条件 | VC(500) |  |  | 所有的付款时间、条件及金额 |
| FADDR | 合同初稿地址 | VC(255) |  |  | 审批的版本地址 |
| LADDR | 合同终稿地址 | VC(255) |  |  | 签订后扫描版本的地址 |

说明：

（1）FCNO是对应的销售合同：如果当前签订的是销售合同，则本字段为空；如果当前签订的是采购合同，则应该有对应的销售合同；这个字段引用本表的CNO字段；

（2）如果是采购合同，则FPNM在页面上显示为“乙方名称”

（3）如果没有经过审批，则合同初稿不能下载；如果审批被退回修改，合同初稿需要提供在线编辑的功能，也可以再次上传覆盖

1. 合同审批信息表（CNT\_APV）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| CNO | 合同编号 | C(8) | PK/FK | N |  |
| UID | 用户id | C(8) | PK/FK | N |  |
| CONTENT | 审批内容 | NVC(255) |  |  |  |
| RESULT | 审批结果 | C(2) |  |  | 1为审批通过；0位审批未通过 |
| APVTM | 审批时间 | DATETIME |  |  |  |
| FLAG | 审批标志位，打回审批时生效 | C(2) |  |  |  |
| OPERATION | 记录给谁审批 | NVC(255) |  |  |  |
| TOUID | 给谁审批 | C(8) |  |  |  |

说明：合同需要经过部门审批、分管领导、单位领导三级审批

1. 合同收款信息表（CNT\_APV）

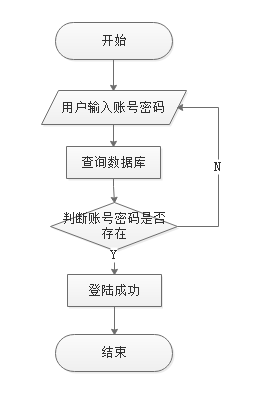
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 主/外键 | 空值 | 备注 |
| CNO | 合同编号 | C(8) | PK/FK | N |  |
| RCVTM | 收款时间 | DATETIME | PK |  |  |
| RCVAMT | 收款金额 | N(7,2) |  |  |  |
| RMK | 备注 |  |  |  | 备注本次收款的金额 |

# 第五章系统功能设计

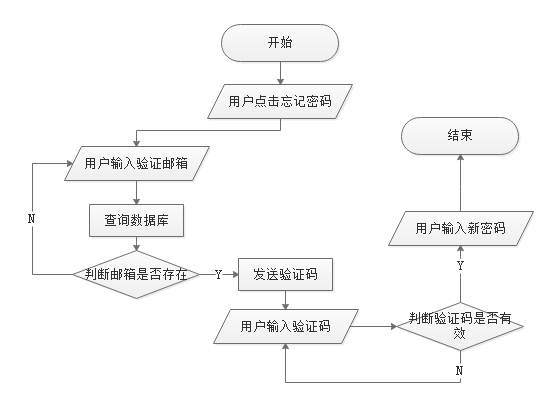
## 5.1登录模块

登陆：用户输入账号密码后，点击登陆按钮，后台会根据用户提交的信息与数据库中存储的信息进行对比（具体的认证过程是通过shiro的Authentication进行认证），匹配成功后则成功登陆系统，进入主页。图5.2.1展示了登陆的流程图。

忘记密码：用户点击忘记密码后进入密码找回页面。在该页面用户需要输入自己的邮箱进行密码找回，输入邮箱后，用户点击获取验证码按钮，此时系统会查找数据库来判断这个邮箱有没有注册，若没有注册，会返回提示信息，若注册过，则会发送验证码，验证码匹配成功后，则可以输入新密码，然后用户密码修改成功。图5.2.2展示了忘记密码的流程图。



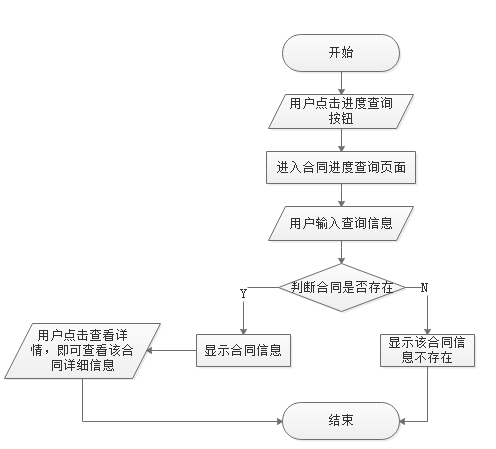
**图5.2.1登陆流程图**



**图5.2.2忘记密码流程图**

## 5.2进度查询模块

该模块将显示系统中所有合同的信息，包括合同号、合同名称、审批进度等，用户可以点击查看详情进入相关合同的详情页面，该页面会显示该合同的部分关键信息，跟该合同的审批流程。当合同的审批进度到达100%时，用户可以下载该合同的word文档。用户也可以在该模块查询合同，可以通过输入合同号、合同名、发起时间段、合同类别进行查询。图5.2.5展示了进度查询的流程图。



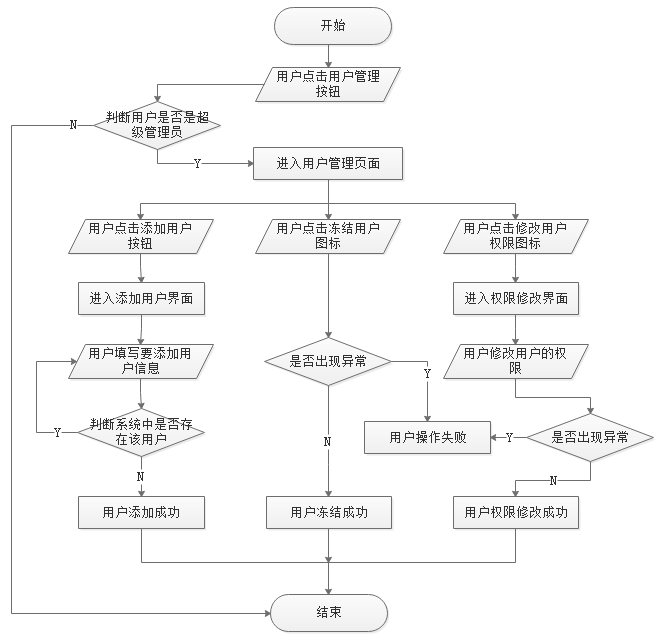
**图5.2.5合同审批流程图**

## 5.3已办事项查询模块

该模块会显示所有审批完成的合同，点击查看详情按钮，则会显示该合同的审批流程，跟该合同的关键信息。此外，该模块中的所有合同都可以进行下载。

## 5.4用户管理模块

该模块只有系统管理员可以进行操作，在该页面，系统管理员可以增加用户，冻结用户，查询用户也可以修改用户权限。图5.2.6展示了用户管理模块的流程图。



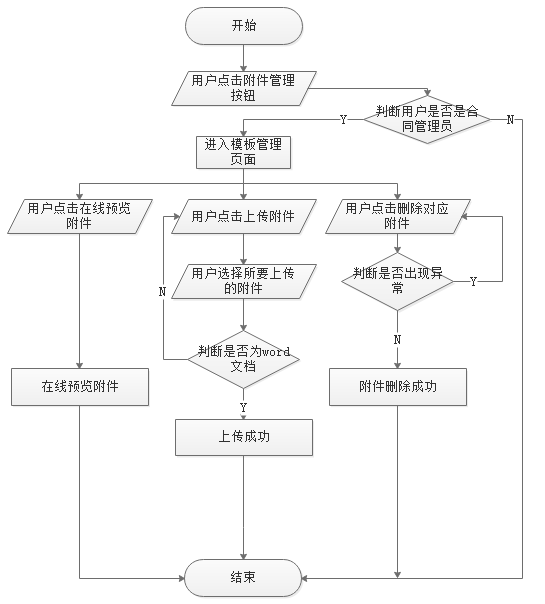
**图5.2.6用户管理模块程图**

## 5.5修改个人信息模块

该模块中用户可以修改个人信息。

## 5.6附件管理模块

该模块只有合同管理员可以进行操作，合同管理员点击附件管理，则会显示所有合同的附件信息，点击详情按钮，则可以添加、删除附件，并且可以在线预览。图3.2.7展示了附件管理模块的流程图。



**图5.2.7附件管理模块图**

# 第六章系统实现

## 6.1登录模块

启动Tomcat，系统会先进入登录界面。此时输入其他页面url，系统不会进入，只有登录成功后，才能访问，此处是通过配置shiro的过滤器来实现。

用户登录进入登录页面，需要输入正确的账号密码才能进入，如果账号、密码错误或不存在，则会给与相应的提示，如果用户被超级管理员冻结，则也不可以进入。如果用户忘记密码，则可以通过邮箱验证来重新设置密码。如果系统中没有该邮箱的注册信息，则会提示该邮箱尚未注册，如果该邮箱被注册，则点击获取邮箱验证码时会获取验证码，用户输入获取到的验证码，然后系统通过ajax技术与后台进行校验，失败则会返回验证信息。验证通过后，用户输入两次新密码，两次输入一致，则找回密码成功。找回密码的表单验证通过bootstrapvalidator进行验证。





关键代码如下所示：

Shiro过滤器设置：

|  |
| --- |
| <bean id="shiroFilter"class="org.apache.shiro.spring.web.ShiroFilterFactoryBean">  <property name="securityManager"ref="securityManager"></property>  <property name="loginUrl"value="/login"></property>  <property name="filterChainDefinitions">  <value>  /bootstrap-3.3.7-dist/\*\*=anon  /images/\*\*=anon  /js/\*\*=anon  /css/\*\*=anon  /my/logout=logout  /my/\*\*=anon  /\*\*=user  </value>  </property>  </bean> |

bootstrapvalidator表单验证(部分展示):

|  |
| --- |
| $(function(){/\*文档加载，执行一个函数\*/  $('.alert').fadeOut(5000);  var eMailCode;  $('#defaultForm').bootstrapValidator({  message:'This value is not valid',  feedbackIcons:{/\*input状态样式图片\*/  valid:'glyphicon glyphicon-ok',  invalid:'glyphicon glyphicon-remove',  validating:'glyphicon glyphicon-refresh'  },  fields:{/\*验证：规则\*/  email:{//验证input项：验证规则  message:'The username is not valid',  validators:{  notEmpty:{//非空验证：提示消息  message:'邮箱不能为空'  },  emailAddress:{  message:'邮箱地址格式有误'  }  }  },  password:{  message:'密码无效',  validators:{  notEmpty:{  message:'密码不能为空'  },  stringLength:{  min:6,  max:30,  message:'密码长度必须在6到30之间'  },  identical:{//相同  field:'password',//需要进行比较的input name值  message:'两次密码不一致'  },  different:{//不能和用户名相同  field:'username',//需要进行比较的input name值  message:'不能和用户名相同'  },  regexp:{  regexp:/^[a-zA-Z0-9\_\.]+$/,  message:'密码只能包含数字，字母，下划线'  }  }  },  validate:{  message:'验证码',  validators:{  notEmpty:{  message:'验证码不能为空'  },  threshold:4,  remote:{//ajax验证。  url:'my/valid',//验证地址  message:'验证码错误',//提示消息  delay:2000,  type:'POST  }  }  }  }  }) |

登陆模块代码展示：

|  |
| --- |
| String error=null;  User user1=userService.getUserByUid(uid);  if(user1==null){  error="该用户尚未注册";  mv.addObject("error",error);  mv.setViewName("login");  return mv;  }  if(user1.getStatus()==-1){  error="该用户被冻结，如有疑问请联系超级管理员";  mv.addObject("error",error);  mv.setViewName("login");  return mv;  }  UsernamePasswordToken token=new UsernamePasswordToken(uid,password);  rememberMe=rememberMe==false?false:rememberMe;  token.setRememberMe(rememberMe);  Subject subject=SecurityUtils.getSubject();  Collection<Session>sessions=sessionDao.getActiveSessions();  for(Session session:sessions){  if(session.getAttribute(DefaultSubjectContext.PRINCIPALS\_SESSION\_KEY)!=null){  SimplePrincipalCollection collection=(SimplePrincipalCollection)session.getAttribute(DefaultSubjectContext.PRINCIPALS\_SESSION\_KEY);  User user=(User)collection.getPrimaryPrincipal();  if(uid.equals(user.getUid())){  mv.setViewName("login");  mv.addObject("error","该用户已登录");  return mv;  }  }  }  try{  subject.login(token);  }catch(UnknownAccountException e){  error="用户名/密码错误";  }catch(IncorrectCredentialsException e){  error="用户名/密码错误";  }catch(AuthenticationException e){  e.printStackTrace();  error="用户尚未注册/出现其他错误";  } |

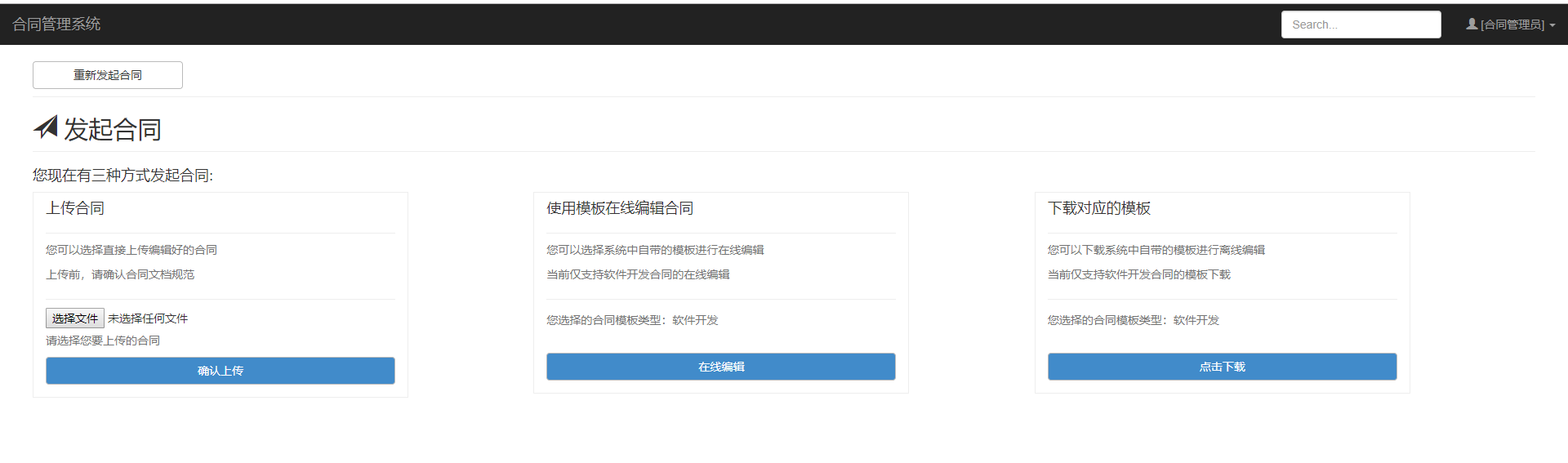
## 6.2合同发起模块

该模块只有合同管理员可以进入。如果其他成员点击该按钮则会显示没有改下权限，合同管理员进入后，需要先填写要发起合同的关键信息，提交成功后，合同管理员才可以正式发起合同。合同管理员可以通过三种方式发起合同：直接上传合同、在线模板编辑、下载合同模板，编辑完成后再进行上传。

直接上传合同：如果用户上传文件为空或者不是word文档则不能上传，系统会给与提示信息。在线模板编辑：只能在线编辑软件开发合同。合同模板下载也是只能下载软件开发的模板，如果第一步选择了其他合同类型，则按钮不可被点击。

如果合同管理员在第一步合同信息填写错误，也可以重新发起合同。





关键代码如下所示：

填写发起合同的关键信息的关键代码如下所示：

|  |
| --- |
| @RequestMapping("/insertCrt")  public ModelAndView insertCrt(ModelAndView mv,String cno,String tpcd,String fpnm,String adcd,String amt,String cnm){  Double amt1=Double.parseDouble(amt);  String adcd1=adcd.substring(0,6);  Contract contract=new Contract();  contract.setCno(cno);  contract.setTpcd(tpcd);  contract.setFpnm(fpnm);  contract.setAdcd(adcd1);  contract.setAmt(amt1);  contract.setCnm(cnm);  try{  int n=contractService.insertCrt(contract);  if(n>0){  mv.setViewName("contract/crtStartDetil");  mv.addObject("success","提交成功");  Session session=SecurityUtils.getSubject().getSession();  session.setAttribute("cno",cno);  Contract contract1=contractService.getApproves(cno);  String crtType=contract1.getConType().getTpNm();  session.setAttribute("contract",contract1);  mv.addObject("crtType",crtType);  mv.addObject("tpcd",contract.getTpcd());  }else{  mv.setViewName("contract/crtStart");  mv.addObject("error","提交失败，请重试");  }  }catch(Exception e){  mv.setViewName("contract/crtStartNew");  mv.addObject("error","系统中可能存在该合同信息...请在确认后重试");  }  return mv;  } |

填写完合同信息后，将合同管理员填写的数据插入到数据库中，同时在session中写入合同的信息，供后续发起合同使用，同时将合同插入数据库的结果写入ModelAndView中，并返回给前台。

合同上传的关键代码如下所示：

|  |
| --- |
| **if**(fileName==""||fileName.equals("")||fileName.length()<=0){  mv.setViewName("contract/crtStartDetil");  mv.addObject("error","上传文件不能为空");  **return** mv;  }  String rootPath="D:\\项目\\毕设\\springMVC\\src\\main\\webapp\\upload";  String prefixName=FilenameUtils.*getExtension*(fileName);  **if**(prefixName.equalsIgnoreCase("doc")||prefixName.equalsIgnoreCase("docx")){  SimpleDateFormat format=**new** SimpleDateFormat("yyyy-MM-dd");  String date=format.format(**new** Date());  String newFileName=rootPath+"\\"+cno+date+fileName.substring(fileName.lastIndexOf("."));  File targitFile=**new** File(newFileName);  **try**{  file.transferTo(targitFile);  mv.setViewName("contract/crtStartNew");  mv.addObject("success","上传成功");  contractService.updateCrtFaddr(cno,newFileName); |

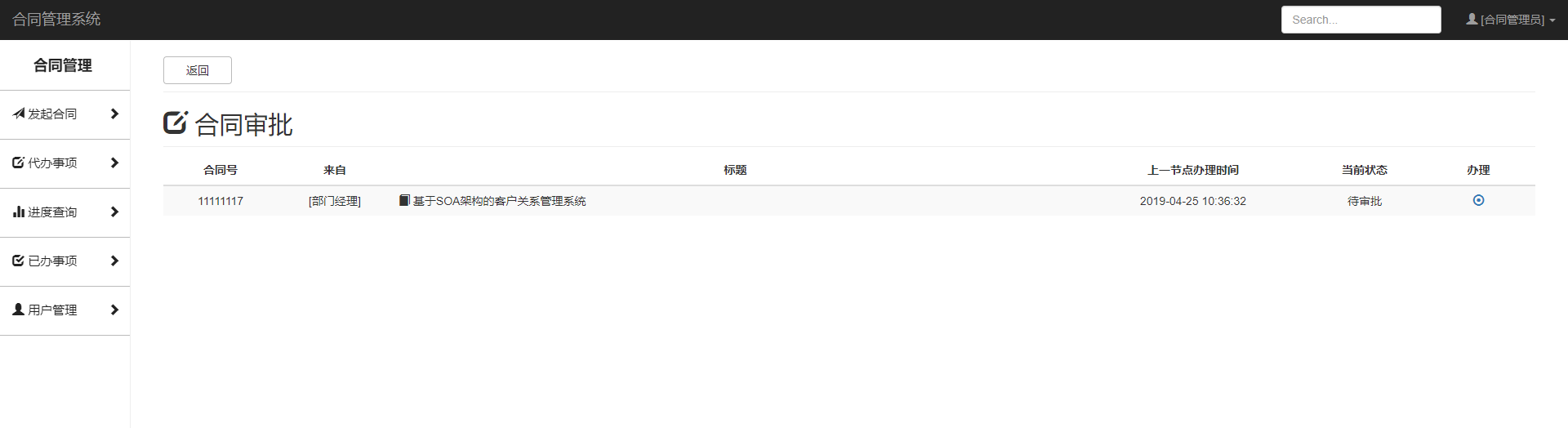
用户上传通过springMVC中自带的上传功能进行上传，上传前先判断上传文件是否为空，然后再通过文件名后缀来判断上传文件是不是word文档，如果不是则返回提示信息。上传成功后，会清除session中的合同信息，以此判断该合同已经上传成功。

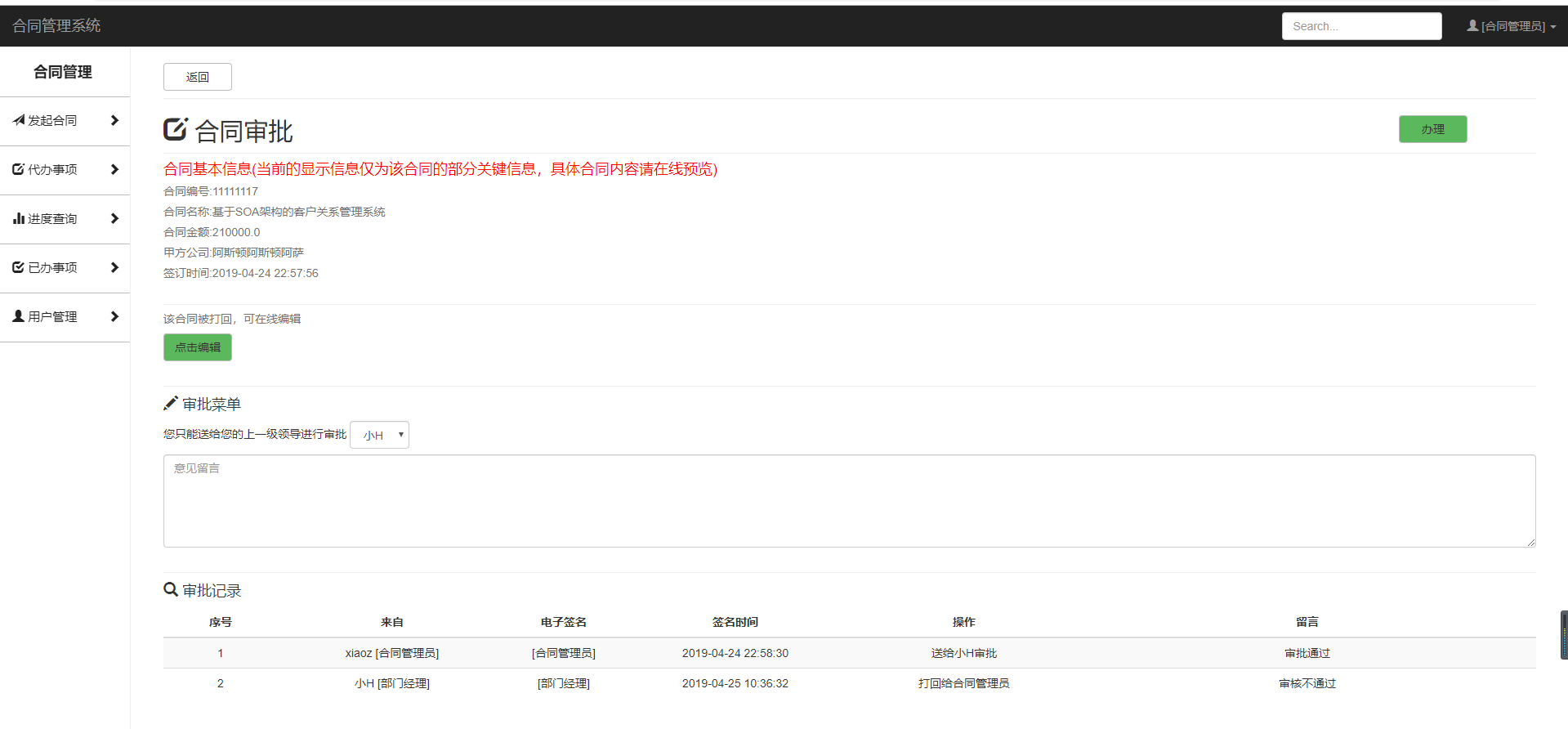
合同模板下载的关键代码如下所示

|  |
| --- |
| //转码，免得文件名中文乱码  String name=URLEncoder.*encode*(conTemplate.getMdlNm(),"UTF-8");  response.addHeader("Content-Disposition","attachment;filename="+name+".doc");  //1.设置文件ContentType类型，这样设置，会自动判断下载文件类型  response.setContentType("application/msword");  //读取要下载的文件，保存到文件输入流  FileInputStream in=**new** FileInputStream(path);  //创建输出流  OutputStream out=response.getOutputStream();  //创建缓冲区  **byte** buffer[]=**new** **byte**[1024];  **int** len=0;  //循环将输入流中的内容读取到缓冲区当中  **while**((len=in.read(buffer))>0){  out.write(buffer,0,len);  }  //关闭文件输入流  in.close();  //关闭输出流  out.close(); |

## 6.3合同审批模块

在该模块中，用户审批成功后会自动传给用户选择的下一级审批人。该模块分为两种情况，合同管理员审批可以编辑合同，而非合同管理员只能在线预览合同。编辑合同跟预览合同也分为两种方式，通过上传方式发起的合同用pageoffice在线预览、编辑合同，通过在线模板发起的合同则通过html模板在线预览、编辑合同。合同管理员只能审批不能打回，而其他审批成员则可以打回合同。同时在该模块下方会显示该合同所有的审批记录。





关键代码如下所示：

审批代码：

|  |
| --- |
| @RequestMapping("/submit")  @Transactional(propagation=Propagation.***REQUIRED***,isolation=Isolation.***DEFAULT***)  **public** String ConAprSubmit(Model model,String cno,String touid,String content,String operation){  **int** n=conApproveService.cryApvSumbitUpdate(cno);  **if**(n<0){  model.addAttribute("approveError","审批失败");  }**else**{  ConApprove conApprove=**new** ConApprove();  conApprove.setCno(cno);  conApprove.setContent(content);  conApprove.setOperation(operation);  conApprove.setTouid(touid);  User loginUser=(User)SecurityUtils.*getSubject*().getPrincipal();  String uid=loginUser.getUid();  conApprove.setUid(uid);  conApprove.setFlag("1");  String rid=(String)SecurityUtils.*getSubject*().getSession().getAttribute("rid");  conApprove.setResult(rid);  **int** m=conApproveService.cryApvSumbitInsert(conApprove);  **if**(m<0){  model.addAttribute("approveError","审批失败");  }  model.addAttribute("approveSuccess","审批成功");  }  **return**"redirect:../crt/conApprove";  } |

用户点击办理按钮时，执行本块代码。本块代码的逻辑是先将合同审批表中该合同的所有flag字段置为0，然后再插入本条审批记录，本条审批记录的flag置为1。

打回代码：

|  |
| --- |
| @RequestMapping("/back")  @Transactional(propagation=Propagation.***REQUIRED***,isolation=Isolation.***DEFAULT***)  **public** String ConAprBack(Model model,String cno,String content,String operation){  System.***out***.println(cno);  **int** n=conApproveService.cryApvBackUpdate(cno);  **if**(n<0){  model.addAttribute("approveError","打回失败");  }**else**{  ConApprove conApprove=**new** ConApprove();  conApprove.setCno(cno);  conApprove.setContent(content);  conApprove.setOperation(operation);  conApprove.setTouid("8000");  User loginUser=(User)SecurityUtils.*getSubject*().getPrincipal();  String uid=loginUser.getUid();  conApprove.setUid(uid);  conApprove.setFlag("1");  conApprove.setResult("-1");  **int** m=conApproveService.cryApvSumbitInsert(conApprove);  **if**(m<0){  model.addAttribute("approveError","打回失败");  }  model.addAttribute("approveSuccess","打回成功");  }  **return**"redirect:../crt/conApprove";  } |

用户点击打回按钮时，执行本块代码。本块代码的逻辑是现将合同审批表中该合同的所有touid字段置删除，然后插入一条打回记录，result置为-1，flag置为1。

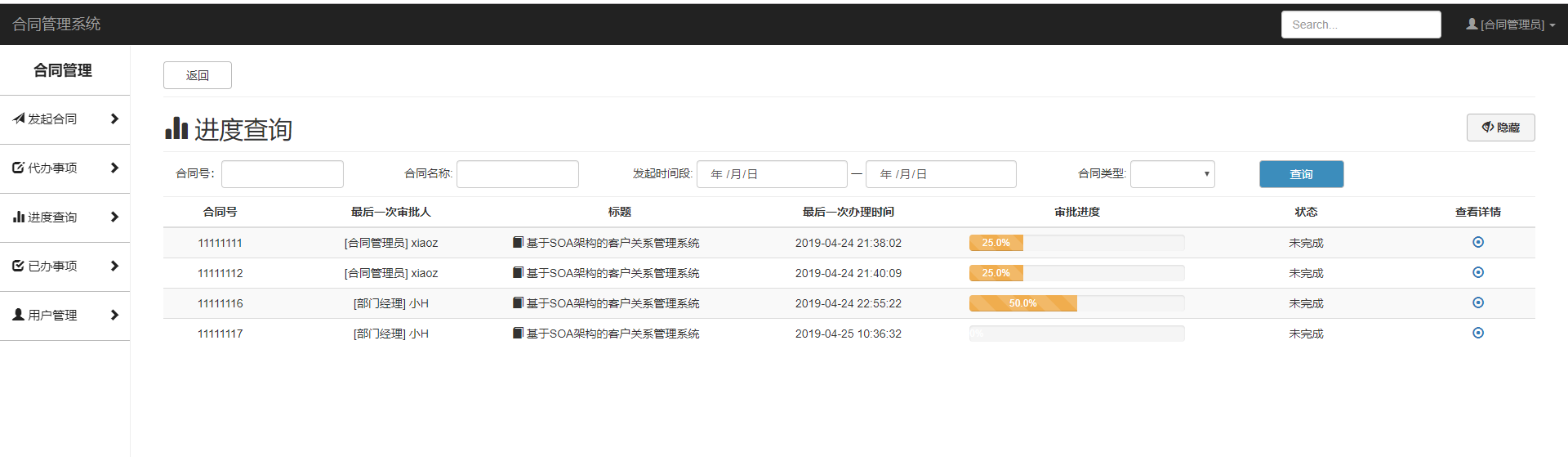
代办事项提示代码:

|  |
| --- |
| @RequestMapping(value="/conApprove")  **public** ModelAndView conApprove(ModelAndView mv,String approveSuccess,String approveError){  System.***out***.println("进入conApprove Controller");  **if**(approveSuccess!=""||!approveSuccess.equals("")){  mv.addObject("approveSuccess",approveSuccess);  }  **if**(approveError!=""||!approveError.equals("")){  mv.addObject("approveError",approveError);  }  User loginUser=(User)SecurityUtils.*getSubject*().getPrincipal();  List<Contract>contractList=contractService.getContractByUser(loginUser.getUid());  **for**(Contract contract:contractList){  List<ConApprove>conApproveList=contract.getConApprove();  **if**(conApproveList.size()!=0){  **for**(**int** i=0;i<conApproveList.size();i++){  ConApprove conApprove=conApproveList.get(i);  String uid1=conApprove.getUid();  User user=UserService.getUserByUid(uid1);  Set<Role>roleSet=user.getRoleList();  Set<String>roleName=**new** HashSet<String>();  **if**(roleSet.size()>=1){  **for**(Role role:roleSet){  roleName.add(role.getRoleName());  }  }  conApprove.setUid(roleName+"");  conApproveList.set(i,conApprove);  }  }  }  mv.addObject("contractList",contractList);  mv.setViewName("contract/conApprove");  **return** mv;  } |

该代码的逻辑是通过当前登录用户id来查找合同审批表中第一条数据（按时间降序排序，取第一条）。

## 6.4进度查询模块

该模块可以直观的查看所有合同的进度，可以按条件查询合同。该页面查询到的合同，通过点击查看详情按钮，则可以查看合同的部分关键信息，跟审批记录，进度为百分之百的合同可以进行下载。





关键代码如下所示:

进度查询关键代码:

|  |
| --- |
| @RequestMapping(value="/processFind")  **public** ModelAndView processFind(ModelAndView mv){  List<Contract>contractList=contractService.getContract("","","","","");  **if**(contractList.size()!=0){  **for**(Contract contract:contractList){  List<ConApprove>conApproveList=contract.getConApprove();  **if**(conApproveList.size()!=0){  **for**(**int** i=0;i<conApproveList.size();i++){  ConApprove conApprove=conApproveList.get(i);  String uid=conApprove.getUid();  User user=UserService.getUserByUid(uid);  Set<Role>roleSet=user.getRoleList();  Set<String>roleName=**new** HashSet<String>();  **if**(roleSet.size()>=1){  **for**(Role role:roleSet){  roleName.add(role.getRoleName());  }  }  conApprove.setUid(roleName+""+user.getuName());  conApproveList.set(i,conApprove);  }  }  }  mv.addObject("contractList",contractList);  }**else**{  mv.addObject("contractList","");  }  mv.setViewName("contract/processFind");  **return** mv;  } |

该模块会查询所有合同，也是取合同审批表中第一条数据（按时间降序排序）。通过mybatis级联，可以获取到用户的角色信息。

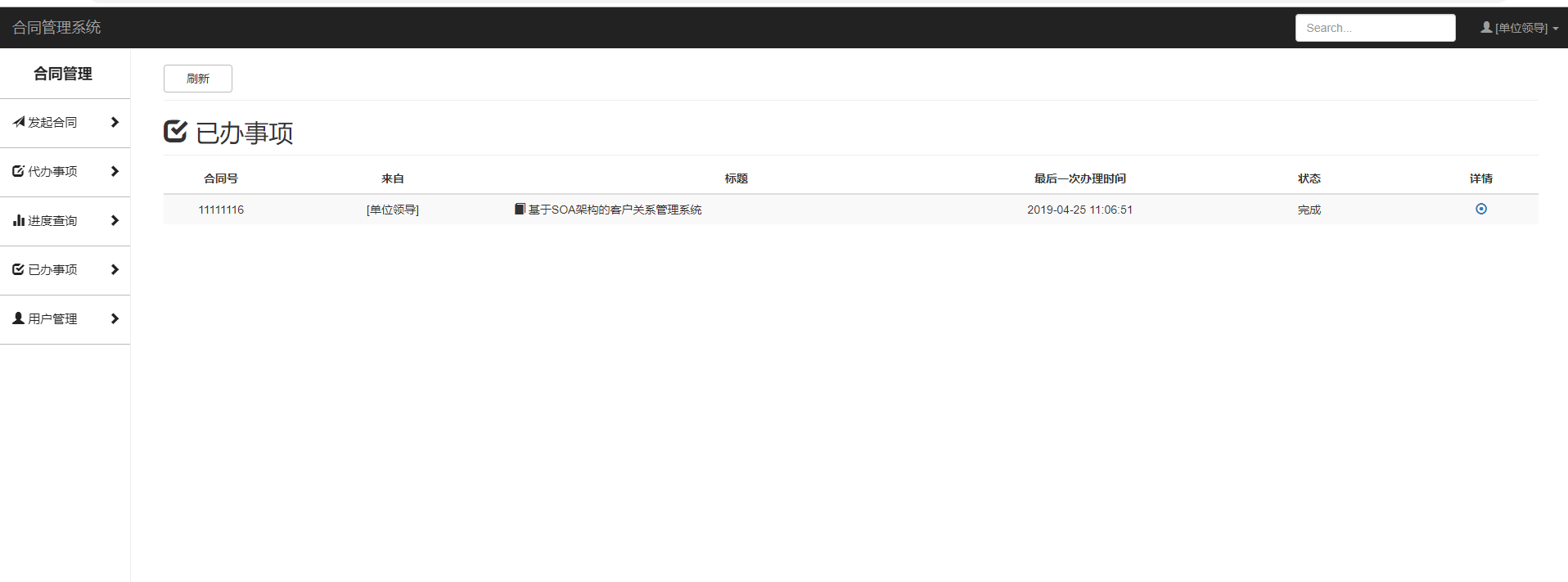
合同查询关键代码：

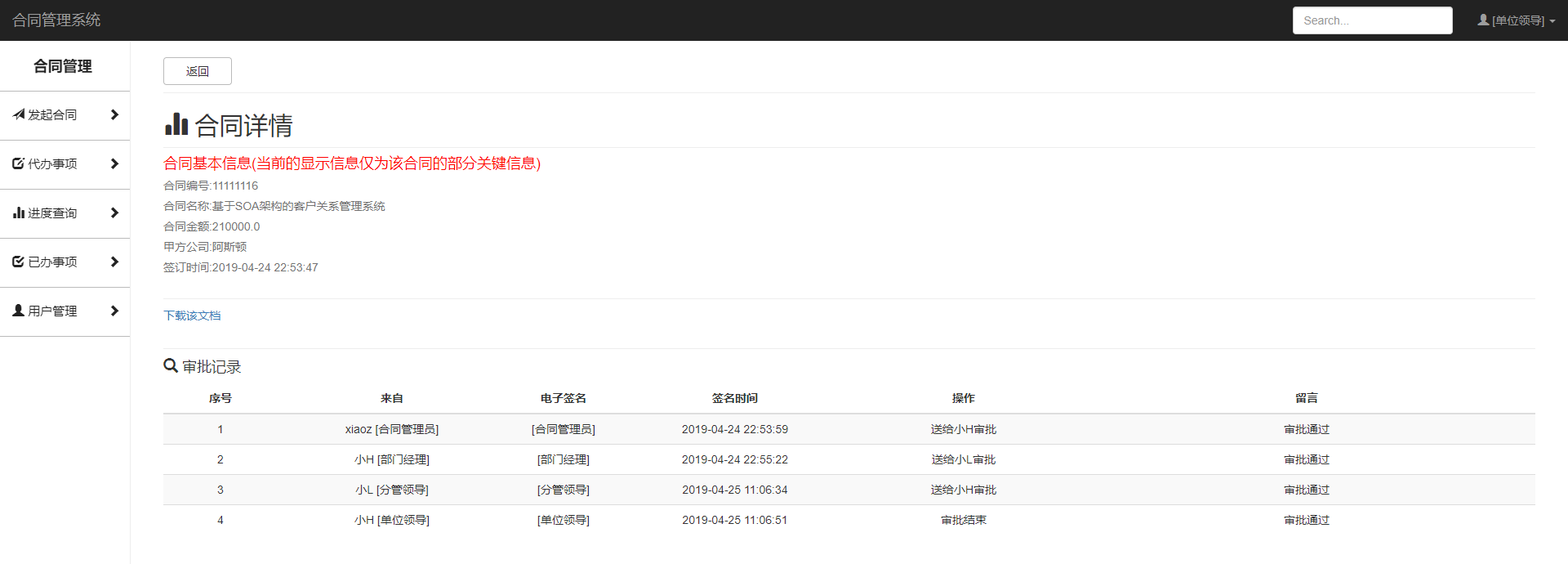
|  |
| --- |
| <select id=*"getContract"*resultMap=*"contractMapper"*>  select cno,tpcd,fcno,cnm,fpnm,sgtm,amt,adcd,paytm,faddr,laddr from cnt\_info  where 1=1  <choose>  <when test=*"cno!=null and cno!=''"*>  AND cno=#{cno}  </when>  <when test=*"cnm!=null and cnm!=''"*>  AND cnm=#{cnm}  </when>  <when test=*"tpcd!=null and tpcd!=''"*>  AND tpcd=#{tpcd}  </when>  <when test=*"startDate!=null and startDate!=''"*>  AND SGTM&gt;#{startDate}  </when>  <when test=*"endDate!=null and endDate!=''"*>  AND SGTM&lt;#{endDate}  </when>  <otherwise>  AND 1=1  </otherwise>  </choose>  </select> |

该模块通过动态SQL来进行合同的查询，并显示相关的信息

## 6.5已办事项查询模块

该模块可以直观的查看审批记录为百分之百的合同信息。通过点击查看详情按钮，则可以查看合同的部分关键信息，跟审批记录，并且可以进行下载。



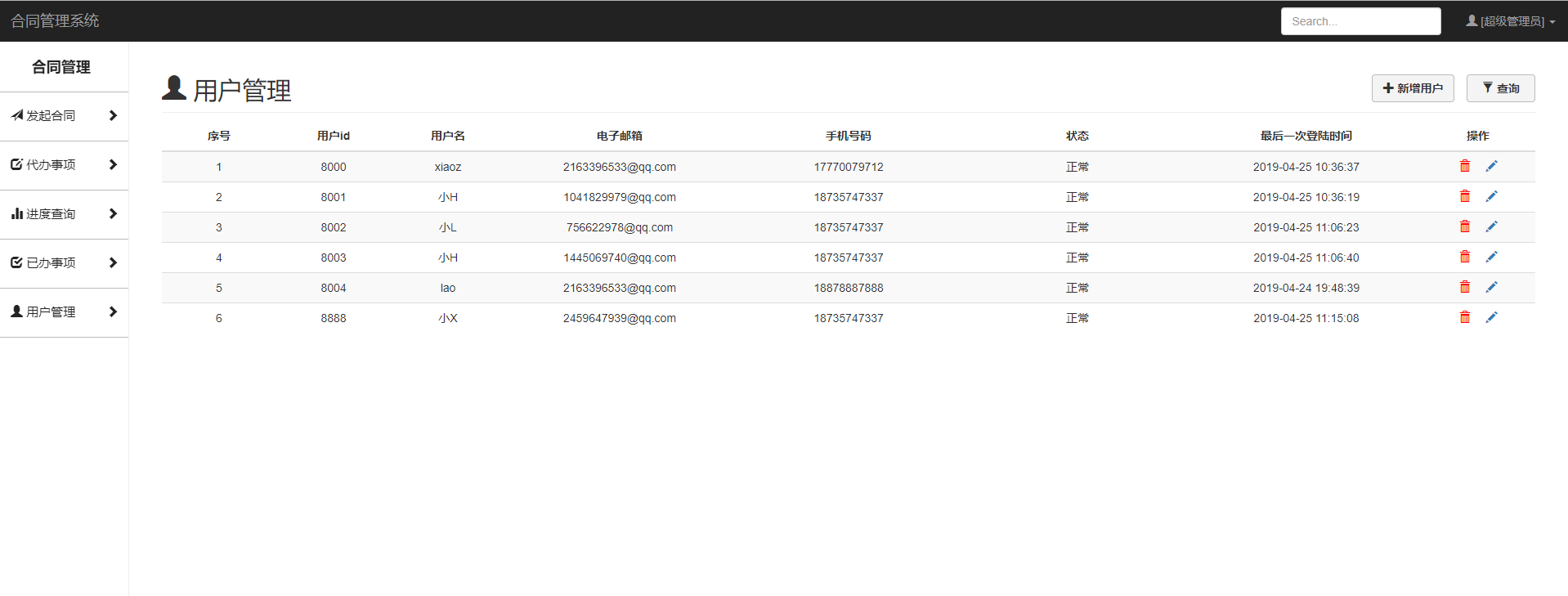


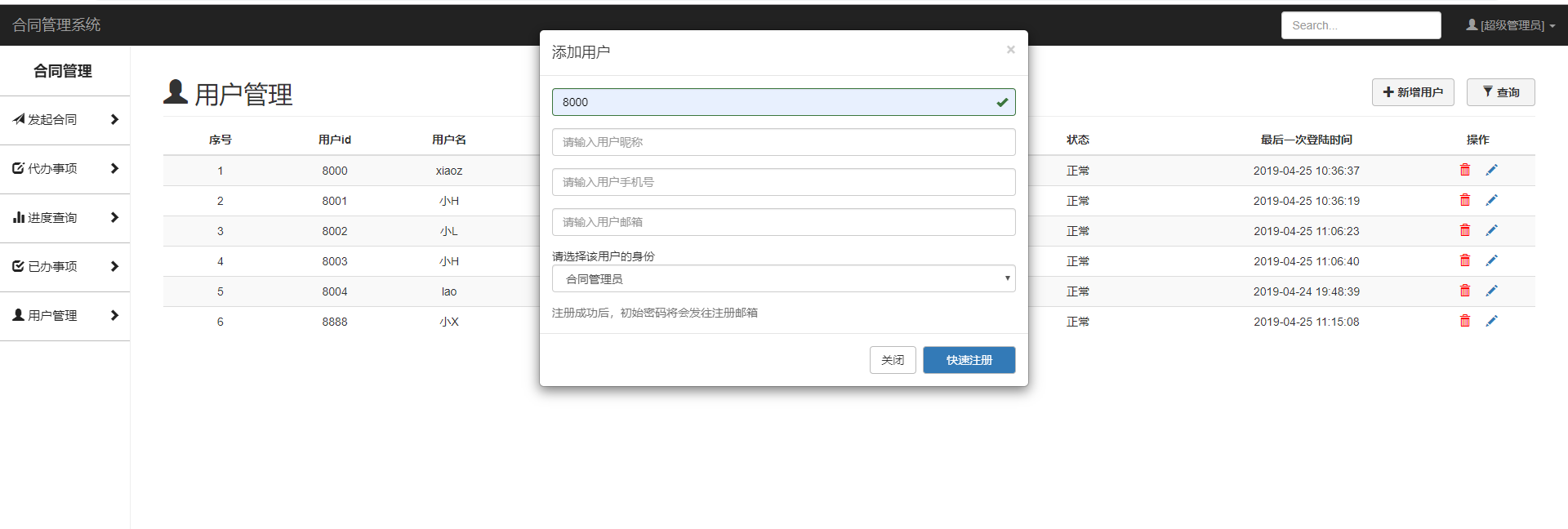
关键代码如下所示:

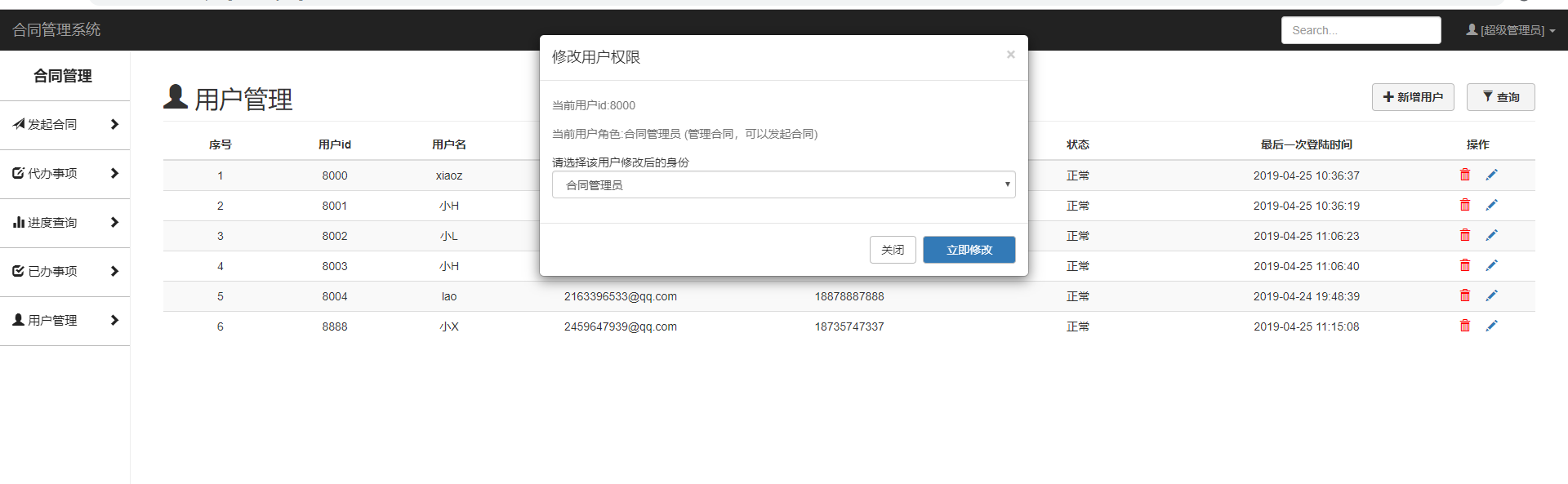
|  |
| --- |
| @RequestMapping(value="/conOk")  **public** ModelAndView conOk(ModelAndView mv){  List<Contract>contractList=contractService.getContractOver();  **for**(Contract contract:contractList){  List<ConApprove>conApproveList=contract.getConApprove();  **if**(conApproveList.size()!=0){  **for**(**int** i=0;i<conApproveList.size();i++){  ConApprove conApprove=conApproveList.get(i);  String uid1=conApprove.getUid();  User user=UserService.getUserByUid(uid1);  Set<Role>roleSet=user.getRoleList();  Set<String>roleName=**new** HashSet<String>();  **if**(roleSet.size()>=1){  **for**(Role role:roleSet){  roleName.add(role.getRoleName());  }  }  conApprove.setUid(roleName+"");  conApproveList.set(i,conApprove);    }  }  }  mv.setViewName("contract/conOK");  mv.addObject("contractList",contractList);  **return** mv;  } |

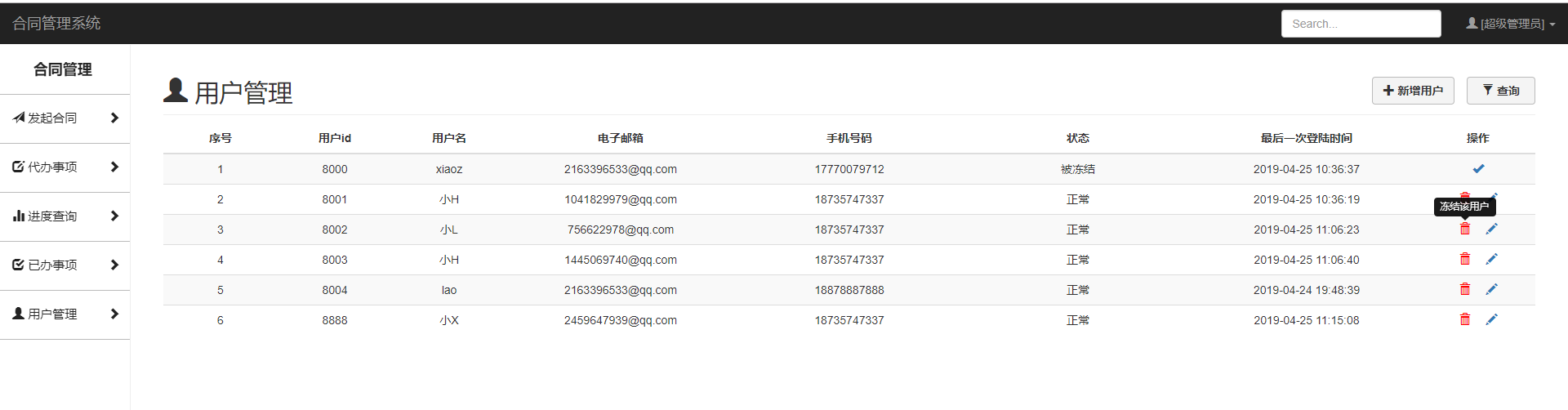
## 6.6用户管理模块

该模块只有超级管理员才有权限访问。在用户管理模块中，超级管理员可以直观的看到所有用户的信息，并可以进行冻结跟权限修改。超级管理员也可以按用户某些信息来查询用户。被冻结的用户不能登录该系统。该模块也可以添加用户。









关键代码如下所示：

用户查询关键代码:

|  |
| --- |
| @RequestMapping("/userFindDetil")  **public** ModelAndView userFindDetil(ModelAndView mv,String uid,String uName,String eMail,String tel){  List<User>userList=userService.getUsers(uid,uName,eMail,tel);  **if**(userList.size()!=0){  mv.addObject("userList",userList);  }  mv.setViewName("/user/userMan");  **return** mv;  } |

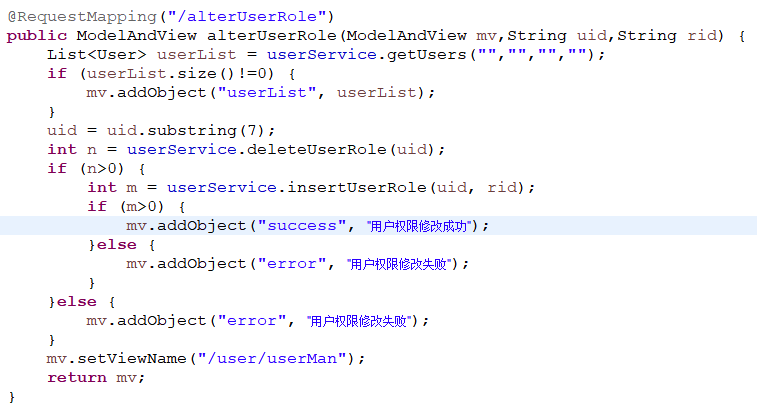
通过动态SQL进行用户查询。

用户冻结关键代码:

|  |
| --- |
| @RequestMapping("/deleteUser")  **public** ModelAndView deleteUser(ModelAndView mv,String uid){  System.***out***.println(uid);  **int** n=userService.freezeUser(uid,-1);  List<User>userList=userService.getUsers("","","","");  **if**(userList.size()!=0){  mv.addObject("userList",userList);  }  mv.setViewName("/user/userMan");  **if**(n>0){  mv.addObject("success","用户"+uid+"冻结成功");  }**else**{  mv.addObject("error","用户"+uid+"冻结失败");  }  **return** mv;  } |

超级管理员点击用户冻结按钮，则会将用户表中的标志位置为-1，登陆时根据标志位来判断用户账号的状态。

修改用户权限关键代码:



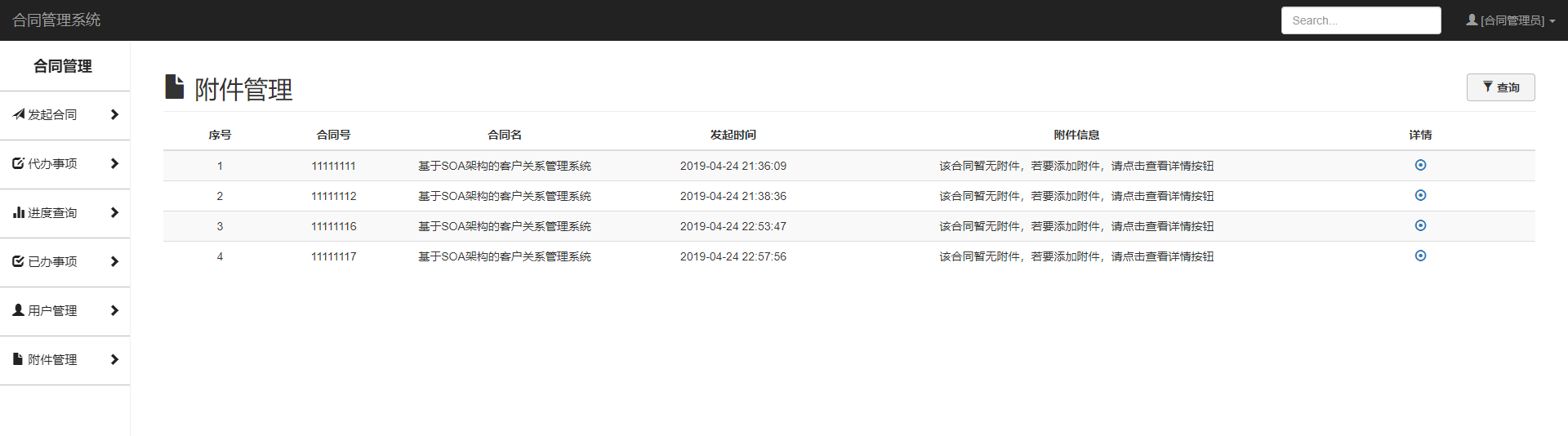
添加用户关键代码:

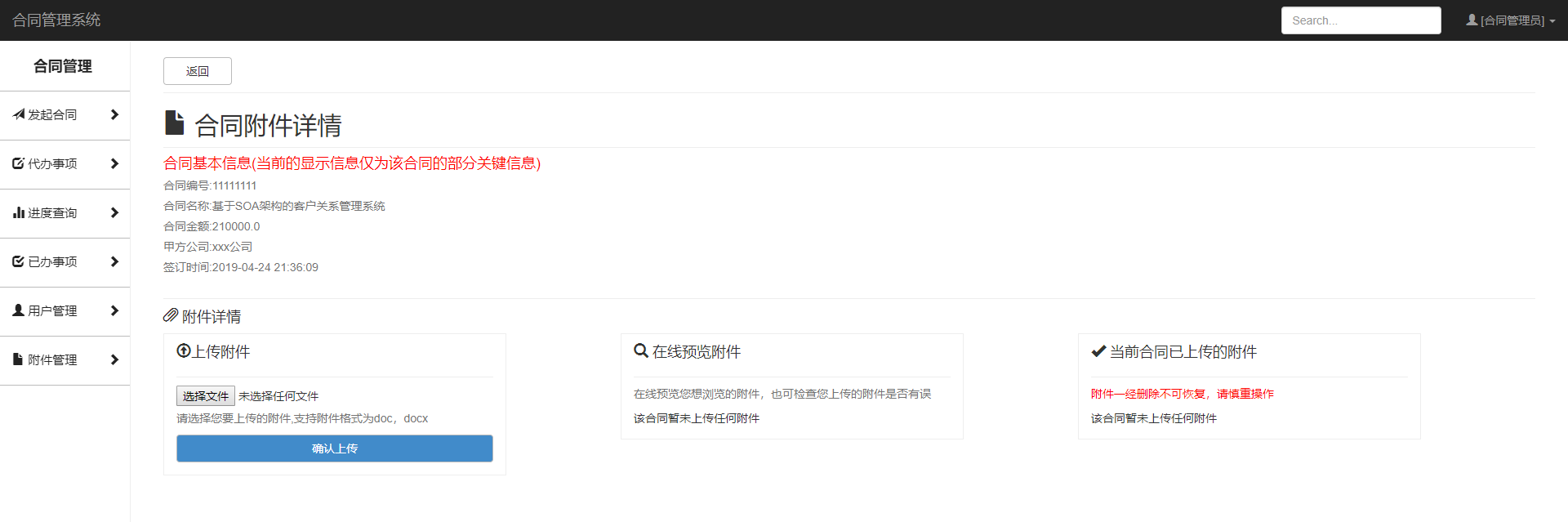
|  |
| --- |
| @RequestMapping("/insertUser")  **public** ModelAndView insertUser(ModelAndView mv,String uid,String uName,String tel,String eMail,String rid){  System.***out***.println(eMail);  User user=**new** User();  user.setUid(uid);  user.setuName(uName);  user.setTel(tel);  user.seteMail(eMail);  String password="000000";  String salt=ShiroUtil.*createSalt*();  password=ShiroUtil.*getSalt*(password,salt);  user.setSalt(salt);  user.setPassword(password);  user.setCreateDate(**new** Date());  **int** n=userService.insert(user);  **if**(n>0){  userService.deleteUserRole(uid);  **int** m=userService.insertUserRole(uid,rid);  **if**(m>0){  mv.addObject("success","用户添加成功");  }**else**{  mv.addObject("error","用户添加失败");  }  }**else**{  mv.addObject("error","用户添加失败");  }  List<User>userList=userService.getUsers("","","","");  **if**(userList.size()!=0){  mv.addObject("userList",userList);  }  mv.setViewName("/user/userMan");  MailUtil.*sendMail*(eMail,"000000");  **return** mv;  } |

将用户提交的表单信息插入输入库，初始密码为000000

## 6.7附件管理模块

该模块只有合同管理员才能进行操作，合同管理员可以随时增加删除附件，也可以在线预览附件。合同管理员也可以按合同号进行查询合同来对合同附件进行管理。





关键代码如下所示:

|  |
| --- |
| String prefixName=FilenameUtils.*getExtension*(fileName);  **if**(prefixName.equalsIgnoreCase("doc")||prefixName.equalsIgnoreCase("docx")){  String newFileName=rootPath+"\\"+fileName;  File targetFile=**new** File(newFileName);  **try**{  file.transferTo(targetFile);  }**catch**(Exception e){  //**TODO**:handle exception  System.***out***.println(e.toString());  mv.addObject("error","上传失败");  **return** mv;  }  //判断是否上传过同名文件  List<ConAppendix>conAppendixList=contract.getConAppendix();  **for**(ConAppendix conAppendix:conAppendixList){  **if**(conAppendix.getApxName().equals(fileName)){  String addr=conAppendix.getAddr();  Date date=conAppendix.getDate();  **int** n=conAppendixService.updateApx(addr,cno,date);  **if**(n>0){  mv.addObject("success","上传成功");  contract=contractService.getApproves(cno);  mv.addObject("contract",contract);  }**else**{  mv.addObject("error","上传失败");  }  **return** mv;  }  }  **int** n=conAppendixService.insertApx(cno,newFileName,fileName);  **if**(n>0){  contract=contractService.getApproves(cno);  mv.addObject("contract",contract);  mv.addObject("success","上传成功");  }**else**{  mv.addObject("error","上传失败");  **return** mv;  } |

每个合同都会创建一个自己的文件夹来管理自己的附件。该上传通过ajax实现。向后台传的参数为formData参数。

# 第七章总结与展望

本系统基于java web实现，主要是为了提高企业合同管理的效率，减少人为劳动，同时可以为企业管理带来很多价值：规范基础数据管理、提高管理效率、实现标准化管理和个性化管理的有机结合、为领导决策提供准确及时广泛的信息、为用户提供全面合同管理解决方案。

这篇论文从选题意义、国内外现状及发展趋势开始对合同管理系统进行了解，然后通过对系统功能、需求，系统各模块进行详细的分析来一步步介绍系统的功能。本系统基于SSM框架进行开发。其中spring作为业务层（service层）、springMVC作为控制层/表现层、mybatis作为持久层。用户的认证跟权限控制是通过shiro框架来进行控制。Shiro框架只需要自定义remal,重写doGetAuthorizationInfo方法和doGetAuthenticationInfo方法便可实现用户自定义的授权与认证。freemarker自定义模板，将动态填充数据后模板下载成word文档。Pageoffice则提供在线编辑功能。前端框架则使用bootstrap UI，表单校验也是使用bootstrapValidator。最终完成了合同管理系统的开发。

通过本次系统设计与实现，提高我实践动手能力的同时，我也学到不少新的东西。从前端界面到后台实现，都是自己一个个设计跟实现的，在系统实现的过程中，也碰到了许多大大小小的问题，不过通过老师的指导、自己的钻研，大体上都有解决。由于个人思维不够严谨，系统功能上也可能会有一些小小的纰漏，代码逻辑也可能不够严谨，希望各位能够指出，我会尽力改正。

# 参考文献

[1]Contract Management Guide[R],Colorado State Dept.of Personnel and Administration

[2]石振武.《道路经济与管理》[M],华中科技大学出版社，2007

[3]臧广州.科研单位管理规章制度全集（中册）[R].银声音像出版社，2004

[4]尹贻林等.《合同法与工程合同管理》[M]，天津大学出版社，2005

[5]梁世林、裴恩才.《工程项目管理学》[M]，2002

[6]陈祥锋.《供应链合同管理模型与应用研究》[M]，2002

[7]唐俊开.《HTML5移动Web开发指南》[M].电子工业出版社，2012

[8]候鸿儒.《响应式Web设计实践》[M].人民邮电出版社，2013

[9]安东尼.《Ajax权威指南》[M].机械工业出版社，2009

[10]未来科技.《Bootstrap实战从入门到精通》[M].中国水利水电出版社，2017

[11]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].电子工业出版社，2017

[12][美]Paul，Deck著；林仪明译.《Spring MVC学习指南第2版》[M].人民邮电出版社，2017

# 致谢

光阴如梭，不知不觉间大学生涯还有学生时代都要结束了。在这四年中，不仅仅是作为学生学习了很多知识，在生活上也明白了很多道理，也算为以后步入社会做了一些功课。这四年，将会是难忘的回忆。

在这里，首先要感谢我的指导老师曾长清，在编写系统的过程中，向老师请教的问题，老师都会给予解决，而且会针对我们个人需要开发的系统，提出一些功能上的建议跟改进的方向，也会给我们一些文献作为参考。真的很感谢老师！同时也要感谢昌大清科的胡坤小哥，他帮我解决了不少的技术问题，每次请教他都会耐心的向我解答，因此这段时间内我学习了很多新的东西，受益匪浅。

最后，再次感谢帮助过我的人，感谢一直陪伴我四年的老师同学，谢谢你们，正是因为你们，我的大学生涯才会如此丰富多彩。

1. [] NCMA：National Contract Management Association 国家合同管理协会 [↑](#footnote-ref-0)
2. [] Contract Management Guide[R],Colorado State Dept.of Personnel and Administration，1959 [↑](#footnote-ref-1)
3. [] 石振武.《道路经济与管理》[M],华中科技大学出版社，2007 [↑](#footnote-ref-2)
4. [] 臧广州.科研单位管理规章制度全集 （中册）[R].银声音像出版社，2004 [↑](#footnote-ref-3)
5. [] 尹贻林等.《合同法与工程合同管理》[M]，天津大学出版社，2005 [↑](#footnote-ref-4)
6. [] 梁世林、裴恩才.《工程项目管理学》[M]，2002 [↑](#footnote-ref-5)
7. [] 陈祥锋.《供应链合同管理模型与应用研究》[M]，2002 [↑](#footnote-ref-6)