

**高等学历继续教育本科生毕业论文（设计）**



**学习类型 🗹网络教育 🞎成人教育 🞎自学考试**

**学习中心（站点）**奥鹏远程教育重庆学习中心(直属)[2016]

**题 目** 基于SpringBoot的脚手架管理系统设计与实现

**办学学院** 计算机学院

**专 业** 计算机科学与技术

**学生姓名** 付册

**学 号** aDD12022f004 **年级** 2009

**指导教师** 张罡

**成人继续教育学院教学服务部制表**

**二Ο二二 年 八 月 一 日**

基于SpringBoot的脚手架管理系统设计与实现

专业：计算机科学与技术

学生：付册 指导老师：张罡

**摘 要**

随着互联网行业的飞速发展，给人类生活创造了新的空间，给社会生产带来了新变化，也给软件应用开发行业注入了活性剂。为了满足人类生活中日益提高的生理和心理需求，应运而生的各类软件应用层出不穷。在此背景下，本文提出了一个研究主题：基于SpringBoot的脚手架管理系统的设计与实现。首先经过对现在主流的软件应用架构的分析与研究，应用软件主要分为CS架构和BS架构。BS架构为主的软件应用主要为各类软件应用的管理端，在各类管理端应用中有40%的功能模块大同小异，且页面与功能相似度极高。结合当前的主流系统要求与当前的主流技术与价格成本，通过该脚手架系统的建设集成了主流的权限控制、系统管理、用户管理、系统工具、代码生成模块。在需求阶段对系统的主要模块功能进行功能性与非功能性的定义分析；在程序的初步设计阶段从整体的系统架构进行分析设计，然后再从系统的层面、业务层面、数据库层面根据核心功能进行分类设计；在系统实现阶段对每个核心功能模块的功能实现进行了介绍，展示了部分程序界面以及代码。文章最后，对主要工作进行分析总结，并对下一步工作改进工作进行了瞻望。

**主题词：**BS Java SpringBoot MVC Mybatis

**Design and implementation of scaffold management system based on springboot**

**Major: Computer science and technology**

**Student: FuCe Supervisor: ZhangGang**

**Abstract**

With the rapid development of the Internet industry, new space has been created for human life, new changes have been brought to social production, and an activator has been injected into the software application development industry. In order to meet the increasing physiological and psychological needs of human life, various software applications emerge one after another. In this context, this paper proposes a research topic: the design and implementation of scaffold management system based on SpringBoot. First of all, after the analysis and research of the current mainstream software application architecture, the application software is mainly divided into CS architecture and BS architecture. The software application based on BS architecture is mainly the management end of various software applications. In various management end applications, 40% of the functional modules are very similar, and the page and function are very similar[1]. Combining the current mainstream system requirements with the current mainstream technology and price costs, the scaffolding system integrates mainstream authority control, system management, user management, system tools, and code generation modules. In the demand stage, the functional and non functional definitions of the main module functions of the system are analyzed; In the preliminary design stage of the program, the overall system architecture is analyzed and designed, and then the core functions are classified and designed from the system level, business level, and database level; In the system implementation stage, the function implementation of each core function module is introduced, and part of the program interface and code are displayed. At the end of the article, the main work is analyzed and summarized, and the next step of work improvement is looked forward to.

**Key words**：BS Java SpringBoot MVC Mybatis

**目 录**

[1导 论 6](#_Toc111368800)

[1.1研究背景 6](#_Toc111368801)

[1.2研究意义 6](#_Toc111368802)

[1.3国内外研究现状 7](#_Toc111368803)

[2系统相关技术与开发环境概述 8](#_Toc111368804)

[2.1系统的相关技术 8](#_Toc111368805)

[2.1.1 B/S模式 8](#_Toc111368806)

[2.1.2 MVC模式 8](#_Toc111368807)

[2.1.3 MySql数据库 8](#_Toc111368808)

[2.1.4 系统开发工具 9](#_Toc111368809)

[2.2系统的开发环境 9](#_Toc111368810)

[3系统需求分析 11](#_Toc111368811)

[3.1系统管理模块分析 11](#_Toc111368812)

[3.1.1 文件管理 11](#_Toc111368813)

[3.1.2 电子邮件管理 11](#_Toc111368814)

[3.1.3 日志管理 11](#_Toc111368815)

[3.1.4 公告管理 11](#_Toc111368816)

[3.1.5 字典管理 11](#_Toc111368817)

[3.1.6 定时任务 11](#_Toc111368818)

[3.1.7 定时任务日志 12](#_Toc111368819)

[3.2 代码生成模块分析 12](#_Toc111368820)

[3.2.1 代码生成 12](#_Toc111368821)

[3.3 用户管理模块分析 12](#_Toc111368822)

[3.3.1 用户管理 12](#_Toc111368823)

[3.3.2 角色管理 12](#_Toc111368824)

[3.3.3 权限展示 12](#_Toc111368825)

[3.4 系统工具模块分析 12](#_Toc111368826)

[3.4.1 表单构建 12](#_Toc111368827)

[3.4.2 Druid监控 13](#_Toc111368828)

[3.4.3 系统监控 13](#_Toc111368829)

[3.4.4 API文档 13](#_Toc111368830)

[3.4.5 URL拦截 13](#_Toc111368831)

[4系统设计 14](#_Toc111368832)

[4.1 总体架构设计 14](#_Toc111368833)

[4.2 系统核心交换设计 14](#_Toc111368834)

[4.3 系统数据流程设计 15](#_Toc111368835)

[4.4 模块功能设计 16](#_Toc111368836)

[4.4.1 系统管理模块设计 16](#_Toc111368837)

[4.4.2 代码生成模块设计 17](#_Toc111368838)

[4.4.3 用户管理模块设计 17](#_Toc111368839)

[4.4.4 系统工具模块设计 17](#_Toc111368840)

[4.5 数据库设计 18](#_Toc111368841)

[5系统实现 22](#_Toc111368842)

[5.1 系统管理模块 22](#_Toc111368843)

[5.1.1 文件管理 22](#_Toc111368844)

[5.1.2 电子邮件管理 23](#_Toc111368845)

[5.1.3 日志管理 24](#_Toc111368846)

[5.1.4 公告管理 25](#_Toc111368847)

[5.1.5 字典管理 26](#_Toc111368848)

[5.1.6 定时任务 27](#_Toc111368849)

[5.1.7 定时任务日志 28](#_Toc111368850)

[5.2 代码生成模块 29](#_Toc111368851)

[5.2.1 代码生成 29](#_Toc111368852)

[5.3 用户管理模块 31](#_Toc111368853)

[5.3.1 用户管理 31](#_Toc111368854)

[5.3.2 角色管理 32](#_Toc111368855)

[5.3.3 权限展示 34](#_Toc111368856)

[5.4 系统工具模块 35](#_Toc111368857)

[5.4.1 表单构建 35](#_Toc111368858)

[5.4.2 Druid监控 36](#_Toc111368859)

[5.4.3 系统监控 37](#_Toc111368860)

[5.4.4 API文档 39](#_Toc111368861)

[5.4.5 URL拦截 40](#_Toc111368862)

[6系统测试用例 42](#_Toc111368863)

[6.1 系统测试方法 42](#_Toc111368864)

[6.2 系统功能测试 42](#_Toc111368865)

[总结 44](#_Toc111368866)

[参考文献 45](#_Toc111368867)

[致谢 46](#_Toc111368868)

1导 论

1.1研究背景

互联网促进了我国社会主义经济的提升。这一新兴的产业正在对不同的产业和行业产生巨大的影响，产业隔阂日益模糊甚至产生交融，各类服务模和新兴经产业的加速崛起，衍生了全新的行业生态。该行业不管是对社会经济的构成体系和发展转变方式等方面都起着不可厚非作用。整个产业也在整个社会的不可或缺的工具，对人类社会的各产业个层次的发展产生着深刻影响。为了满足人类生活中日益提高的生理和心理需求，应运而生的各类软件应用层出不穷。

据《中国互联网状况》报报导大型企业电子商务正在从网上信息发布、采购、销售等基础性应用向上下游企业间网上设计、制造、计划管理等全方位协同方向发展。中小企业电子商务应用意识普遍提高。中小应用电子商务的Internet的企业家数量保持较高的增长速度。新零售模式的星期促使该模式规模增长速度巨大，饭圈市场规则逐渐得到规范。据调查，电子商务专业化服务体系正在形成，数字认证、电子支付、物流配送等电子商务应用支撑体系正在逐步形成。上到大型企业下到中小型企业，在电子商务的建设上80%采用BS架构设计，且核心功能模块大同小异。

1.2研究意义

随着国家政策的支持，在各类软件应用的开发中，开发工程师遵循先搭建程序的公共模块，再开始从事特色业务模块的开发。从程序员角度讲这样对于公共模块的反复开发不利于自身技术的发展与积累，在程序员圈子中广为流传的“CV大法”也是对这种重复工作的调侃；从企业角度来讲这样重复的工作只会增加公司的人力成本支出，因此一套兼具包含公众模块管理以及可以“自动生成代码”的系统越来越成为各大企业的迫切需求。

在实际的开发生产中，互联网大厂已经形成了一套自己的研发体系以及编程规范，己经是将整个研发体系提升了一个档次，但是在互联网高速发展的今天，更多的中小型公司还没有能力去形成了自己的研发体系，且重复“造轮子”的模式己经不能满足中小企业的实际需求，当下的开发思维和模式给中小企业带来了新的瓶颈。基于SpringBoot的脚扌架管理系统快速地搭建类管理系统，是本课题的目的。

1.3国内外研究现状

由于国内政府对互联网应用的鼓励，使互联网应用得到了常足的发展，通过大力构建调整该发展不但有利于方便人类社会的互联互通应用，且通过大力构建系统结构合理、行业发展均衡的互联网应用形式，提高互联网应用的整体发展和技术水平。因此为促进经济增长正大力发展商务类网站发展，提高各公共服务类的水平，加大政务网络建设，积极推进电子政务应用的建设，支持网络直播、短视频平台等新兴媒体，倡议内容丰富、形式多样化的公众类服务信息，以满足社会消费形态信息多样化、层次化的需求。

在国内相关领域当中只有个别的互联网大厂以及传统软件行业的头部公司开始积极参与开源的架构以及类似脚手架的架构开发。但是存在的问题在于这种头部公司以及互联网大公司所创建的开源架构都是根据自身的业务需求以及技术能力体现的，这样的方式反而有点鸡肋。对于有这种业务和体量的公司都会采用自己研发的框架或者架构，不会用第三方公司的架构。但是对于没有这种业务体量的小公司而言，也用不上各种“高大上”的框架，毕竟架构技术难度高对于人力成本就会提升，小公司从根本上就用不到某些东西，例如小公司就一台服务器，如果架构还采用所谓的微服务这类的，那只会增加维护成本，还不如常规的单体架构。

在国外也有程序员开发类似的脚手架应用，但是出于各种原因，所以在国内用到的微乎其微，能流传到国内的都是技术新且被大厂利用的，对于大多数中小型公司也是不适用。所以结合国内外关于脚手架软件的研发现状，针对中小型公司开发一套技术适用成本低廉的脚手架管理系统非常有前景。

2系统相关技术与开发环境概述

2.1系统的相关技术

2.1.1 B/S模式

B/S结构（Browser / Server 结构）结构即浏览器和服务器结构[2]。它是随着移动互联网技术的兴起，这是C/S基础结构的改进以适应多种用户操作系统环境的结构。在B/S结构下，用户通过INTERNET各类浏览器来启动用户操作界面，极少部分的业务以及逻辑在Browser（前端）实现，但是主要业务和处理事务逻辑在通过INTERNET调用Server（服务器端）实现，构成前端、服务端、数据存储的三层B-树结构。这样极大简化了用户的电脑负荷，用户操作系统的难度，减少了系统整体维护与能提升该升级的时间成本和TCA成本，提升了用户和系统运维人员的的整体满意度。以目前的技术看，通过互联网与云服务结合的模式下数据库应用，相对易于把握、成本也是较低的。它是一次性到位的开发，能实现不同的用户，从不同的低于，以多种多样的接入方式（比如5G, WAN, Internet/Intranet等）访问和操作共同的数据库；它出色的权限管理能有效地保护平台以及数据，服务器与数据库独立也很安全。特别是在JAVA这样的跨平台语言出现之后，以B/S架构研发的管理软件更是操作简便、响应速度快、整体效果优良。

2.1.2 MVC模式

MVC作为一种典型的三层模型，是现代软件架构中优良的也是使用最多的架构模式。该模型把整个系统分为三个部分。对开发人员它使开发操作更加清晰，方便协调各层接口与接口之间的调用，并使负责不同业务分支的开发人员能够并行开发，从而提高开发效率和开发进度。三层模型结构的基本要求是，每一层代码相对独立在修改时不影响其他层的操作性，有利于各层逻辑代码的的复用和公共方法的抽取。修改层间接口调用应该尽量减少代码修改，以降低各层之间的耦合性。通过其核心观念的表达，我们可以明白，完美的结构层次完全符合低耦合的设计目的，从而保证了系统良好的可扩展性和可复用性的最大可能性。

2.1.3 MySql数据库

作者在开发此系统中选用了现在比较流行的关系型数据库MySQL数据库。该数据库是由Oracle公司提供的一个免费的开源数据库（RDBMS），也是如今中小企业使用最多最广泛的关系型数据库。本系统选择MySQL的主要原因有两点。一它是一个免费的开源关系数据库，而且支持多方面的操作，我们可以单独修改基础代码以适应我们的实际需求[3]。而且，无需为像SqlServer这样的数据库付费使用。并且它具有高度自主性和易用性，在一些数据量不大并发少的项目中，可以使用单体MySQL，在数据体量大并发大的项目中可以使用MySQL集群的方式来同步存储数据，且能通过第三方组件能有效地管理MySQL。

2.1.4 系统开发工具

系统开发工具这里选用了Eclipse作为开发工具。首先它是免费且开源的，可以根据自身开发特段进行调整组件，自行下载或者编写需要的组件。首先，它作为一款功能强大的IDE可用编写代码胜任开发工作。另外它可以为其他系统如SHEL.等的编译工具打开它。这些程序可以在eclipse上更改编译后集成到其他程序。对比Myeclipse而言Eclipse有运行速度快，电脑资源使用低，及时适用等优点。最重要的是Eclipse是开源免费使用，不需要像Myeclipse和IEDA那样收费。且下载方便，安装简单，绿色版解压即可用并且安装插件也非常容易。

2.2系统的开发环境

本系统采用Java语言开发，JDK版本8.0以上，数据库版本MySql5.7以上；主要技术要点详见表1。

表2-1详细技术选型

|  |  |  |  |
| --- | --- | --- | --- |
| **技术** | **名称** | **技术** | **名称** |
| springboot | springboot框架 | jQuery | 函式库 |
| Apache Shiro | 权限框架 | bootstrap | 前端页面框架 |
| MyBatis Generator | 代码生成 | Font-awesome | 字体图标 |
| PageHelper | MyBatis物理分页插件 | jquery.validate | jquery验证插件 |
| hikari | 数据库连接池 | vue | 渐进式框架 |
| Thymeleaf | 模板引擎 | ladda.min.js | 按钮加载js |

续表2-1详细技术选型

|  |  |  |  |
| --- | --- | --- | --- |
| **技术** | **名称** | **技术** | **名称** |
| Log4J | 日志组件 | bootstrap-table | 表格组件 |
| Swagger2 | 接口测试框架 | layer.js | 弹窗组件 |
| Maven | 项目构建管理 | jquery.blockUI.js | 遮蔽层组件 |
| Websocket | websocket消息通知 | Layui.js | 前台导出组件 |
| GSON | 谷歌json | bootstrap-wysihtml5 | 富文本 |
| druid | 阿里连接池 | bootstrap-switch | 开关按钮 |
| quartz | 定时框架 | LayEdit | 富文本 |

3系统需求分析

3.1系统管理模块分析

3.1.1 文件管理

文件管理系统是系统管理模块中的一个子模块，整个模块包含了文件的新增、编辑、查看与查询。用户在此模块下可对整个系统的上传文件进行实时动态管理。系统根据上传的文件类型进行自动归类展示，方便维护人员管理。

3.1.2 电子邮件管理

用户可以在此模块下进行直接发送邮件的操作，也能够在此处查看有系统直接向用户发送的邮件，且支持邮件的导出。让用户可以直观地查看由系统发送的各类邮件，也为后续其他统计分析提供数据支撑。

3.1.3 日志管理

日志管理主要展示系统内部的日志信息；本系统中日志信息主要分为三种：一种是用户操作日志，一种是系统运行产生的错误日志，一种是系统定时任务产生的日志。在本模块下可以实时展示日志情况，支持导出与统计。后期开发人员可以根据日志信息针对系统进行调优改错。

3.1.4 公告管理

系统管理员在此模块下对发布的各类公告进行管理，在线增删改查，前端进行同步更新在线用户可查收到系统广播，离线用户在登录后可在公告模块下查看发布的公告。发布对象可以看到指定发布的信息，系统管理员拥有数据与功能的最高权限。

3.1.5 字典管理

字典管理可对系统中各类基础字典表数据，进行统一的维护管理。管理员登录系统后，用户可对数据字典中各类数据类别进行统一管理。可对数据字典的数据类别目录进行添加、修改、删除操作。添加的数据类别信息包括上级名称、类别编号、是否启用、类别名称、备注信息等。添加后的数据类别以下拉目录结构进行显示，用户可直接点击查看。

3.1.6 定时任务

定时任务管理模块是对整个系统中需要使用到定时任务的业务进行统一的管理与维护，用户可根据实际需求通过cron的操作实现对定时任务的执行时间、执行策略等进行配置。

3.1.7 定时任务日志

具有相应权限的用户在定时任务日志模块下可以查看所有执行定时任务的日志信息，包括执行的开始时间、结束时间、执行状态、调用任务、详细日志等信息，用户可以点击详情查看更相信的日志信息。

3.2 代码生成模块分析

3.2.1 代码生成

在此模块下用户可以根据系统自动扫描出的数据库表，选择对应的表进行自动生成代码的操作；用户可以自定义生成的包路径，以及直接选择菜单权限。支持自动生成后端代码（java）和前端代码（html），且支持预览前端界面。

3.3 用户管理模块分析

3.3.1 用户管理

用户登录系统后，为系统使用主体按机构定义机构属性，用户有所属机构的，为其定义角色属性，角色属性继承其机构属性。无所属机构的用户，直接定义其角色属性。通过设置用户功能权限，可以有效规避用户非法操作、隔绝病毒传播、提高机密信息安全性。用户可对已有的用户信息进增删改查。可对用户信息进行添加操作，添加的用户信息包括所属机构、登录名、真实姓名、性别、登录密码、确认密码、移动电话、固定电话、电子邮件。用户还可对已经添加的用户信息进行修改、删除的操作。用户也可对已经添加的用户进行角色分配。

3.3.2 角色管理

角色管理主要实现角色的增删改查，按条件查询角色。可根据按角色名查询角色，新增角色，修改角色信息，删除角色。为用户分配不同的角色，角色与用户逻辑上是多对多的关系，并对不同角色进行相对应的权限授权。角色管理包括：创建角色，授予用户角色，修改用户角色。

3.3.3 权限展示

用户权限管理模型是现实中的用户、用户的职责和各机构之间的关系在系统中的抽象，它是系统实现统一认证、授权和权限管理的基础[4]。用户功能权限和用户权限两个部分组成了用户权限的核心功能。

3.4 系统工具模块分析

3.4.1 表单构建

表单构建主要实现了表单页的快速创建，可以让使用人员实现点击组件，在页面上预览效果同时实时的把页面html代码和js代码输入到旁边展示。生成的代码可以直接复制拿到开发工具中使用。

3.4.2 Druid监控

Druid监控模块核心功能主要是监控Mysql数据库的连接池使用情况，Druid综合了C3P0、DBCP等数据库连接池的优点，同时加入了日志监控。使管理员可以很好地掌握数据库池连接状态和SQL的执行效率情况，方便管理员对数据库连接池状态以及SQL的执行情况进行追踪与维护。

3.4.3 系统监控

系统监控模块主要是将整个系统的运行状态以图表的方式进行动态展示，主要包含系统运行时长、Java版本、服务器基本信息、流量统计、访问量、系统公告等几大模块。可以让管理员直接在系统中查看这些信息，而不是需要再去其他统计系统中查看。

3.4.4 API文档

对于系统开发而言一份完整的简洁的API文档对于整个系统研发的工作开展非常有必要。因此本模块下会根据系统的配置自动生成API文档。自动生成的文档包含接口描述、请求参数、响应参数、响应状态等信息，方便直观地展示整个接口所需要的信息。可自动生成word下载查看也可以在线网页查看。

3.4.5 URL拦截

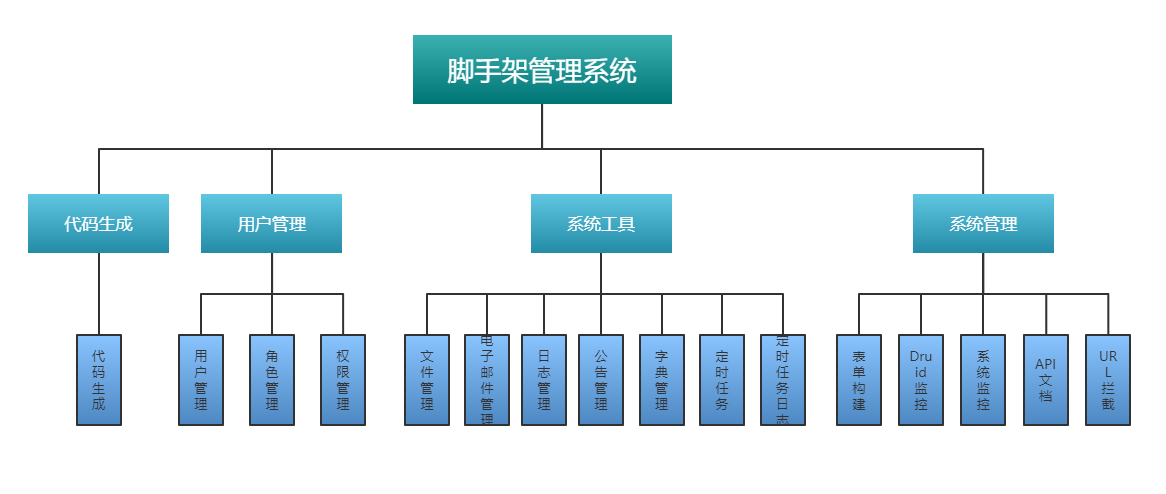
传统的URL拦截要不是写在系统的配置文件要不就是写在静态代码块里面，这种方式死板且不利于管理。在本系统中对这种需要拦截的URL进行动态的管理，灵活且高效，动态的管理拦截的URL。

4系统设计

4.1 总体架构设计

就本系统而言遵循J2EE标准采用Java语言开发，该平台具有良好的扩展性，系统整体数据资源的保密性、应用自身权限高可用性。根据系统的特点与设计要求。需要满足横向扩宽和纵向深入的技术要求。这就要平台各个公共模块和具有相似性的模块提取整个引入系统，并且每个模块的需求须清晰与明确。

脚手架系统主要包括：系统管理模块、代码生成模块、用户管理模块、系统工具模块，整体架构如图4-1。

图4-1 系统架构图

4.2 系统核心交换设计

本文定义了所有Controllers的公共基类（BaseControllers）。所有具体的Controllers都继承于公共基类BasControllers，并定义实现私有的请求方法，从而实现View（视图层）的核心功能。系统视图层的核心类图如图4-2所示。

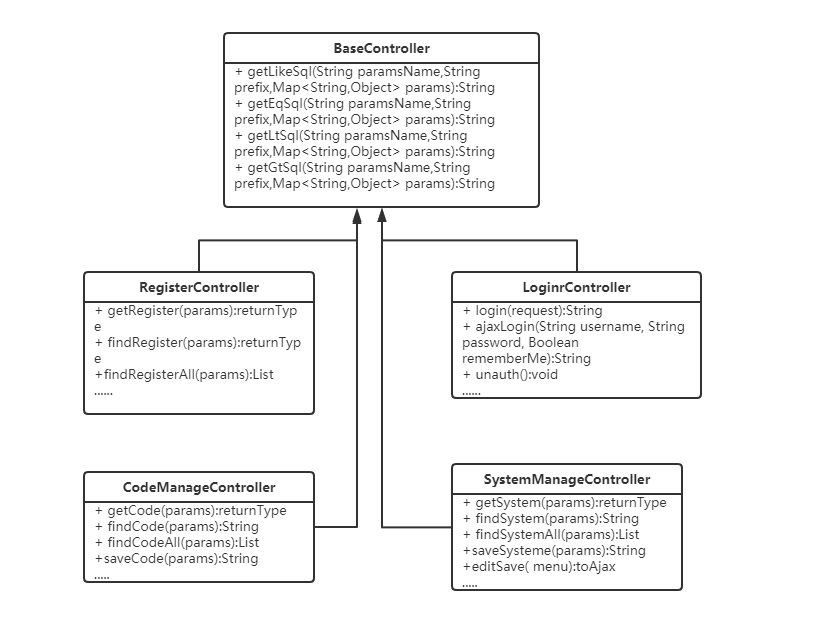


图4-2 交换控制类图

在交互层控制类图中，我们采用Spring注解MVC[5]。在实际开发当中，可应对多种情况的Controllers比只能处理单一功能的Controllers更受喜爱。BaseControllers为整个系统的公共基类，RegisterController为系统注册中心，LoginrController为用户瞪陆状态核心控制器，CodeManageController为核心模块代码生成管理的控制器，SystemManageController为系统管理核心控制类，其基本功能包括增加系统功能、编辑调整系统信息、删除（物理、逻辑）系统信息等。

4.3 系统数据流程设计

持久层为数据库提供了多种操作模式如：新增、删除、编辑、查询等操作，其中，数据层对数据库的新增、删除、编辑、查询等操作的业务流程图如下所示。

在系统数据层中，通过DAO类增加数据信息操作的流程图如下图4-3所示:

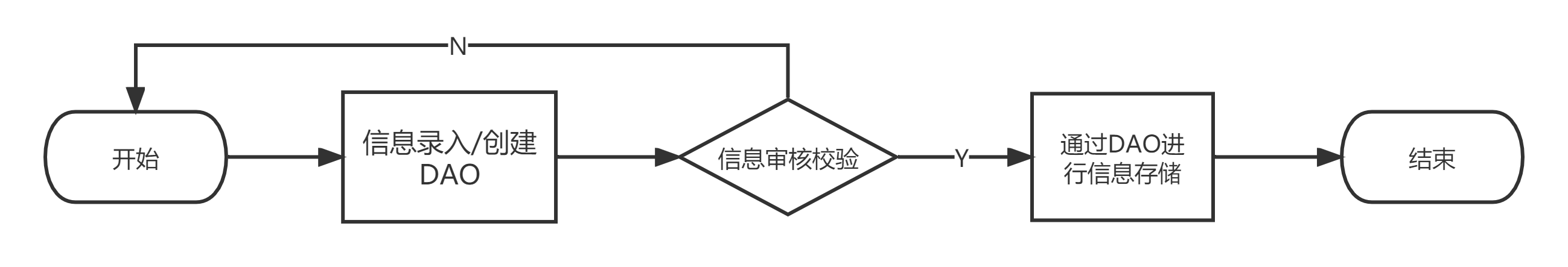


图4-3 增加数据流程图

在系统数据层中，通过DAO类删除数据信息操作的流程图如下图4-4所示:

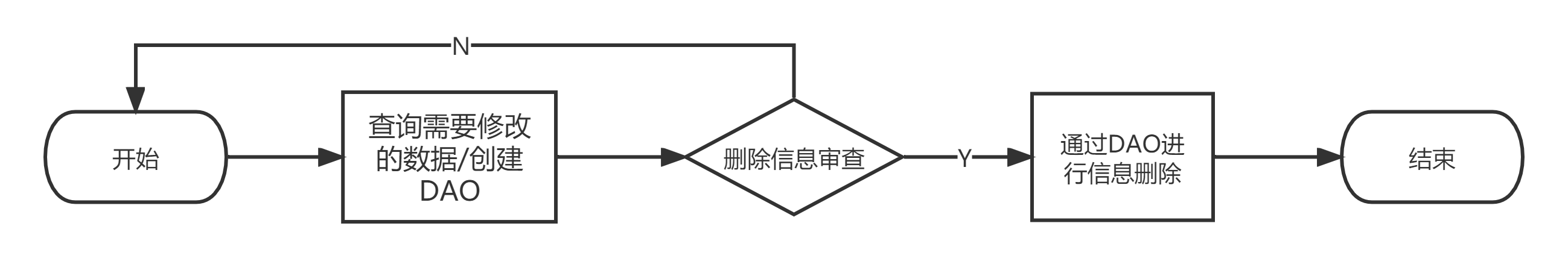


图4-4 删除数据流程图

在系统数据层中，通过DAO类修改数据信息操作的流程图如下图4-5所示:

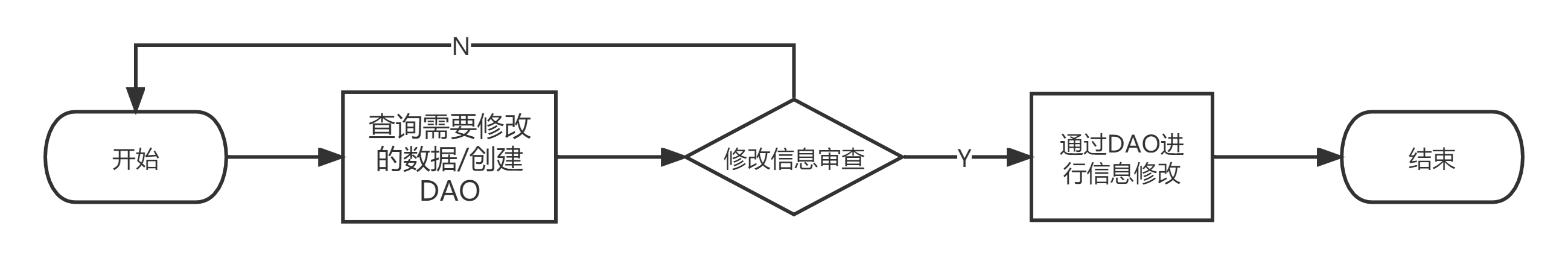


图4-5 修改数据流程图

通过继承实现BaseDao<T>中的方法后，我们可以通过BaseDao<T>的子类，与实体类的映射对数据进行DML操作，在系统数据层中，我们对数据库数据的insert、delete 、update、select等操作都是通过映射类进行完成的。其中操作的流程图如上四个图所示。

4.4 模块功能设计

4.4.1 系统管理模块设计

用户在此模块下可以对系统进行一系列的管理操作。能够实现对电子邮件、日志信息、公告信息等信息进行实时管理，为管理员对系统管理提供便捷的操作。

系统管理模块包括7个子模块：文件管理，电子邮件管理，日志管理，字典管理，定时任务，定时任务日志：

（1）文件管理：管理整个系统的文件上传，包含新增、编辑、查询、查看、删除功能；

（2）电子邮件管理：对系统平台的有电子邮件服务的进行管理，包含邮件发送、查询、查看、删除功能；

（3）日志管理：有权限的用户可以在此功能下对各个模块或者用户的日志进行查看以及查询、导出功能；

（4）公告管理：管理员可以在此模块下发布公告，包含发布、查询、查看、删除（撤销）功能；

（5）字典管理：管理员对整个系统的字典进行管理，包含新增、查询、查看、删除功能；

（6）定时任务管理：管理员在此模块下配置定时任务，包含先、查询、查看、删除功能；

（7）定时任务日志：管理员可以在此模块下查看定时任务的运行情况，包含查看、查询、导出功能；

4.4.2 代码生成模块设计

在此模块下用户可以根据系统自动扫描出的数据库表，对选中的数据库表进行操作，点击数据库表格可以查看表注释以及各字段的详细信息，然后通过自定义设置路径生产Java代码以及前端代码。生成文件压缩包然后根据自定义地址直接存放到本机对应路径下。

4.4.3 用户管理模块设计

用户分为管理用户和业务操作用户两大类。一类是根据自身业务的不同按照等级分为不同级别和权限的管理员，如SAdmin系统超级管理员、某模块子管理员等。能够实现对各类用户的新增、修改、查询，按条件查询用户。可且可以在该模块下对用户进行多维度查询，增加系统用户，编辑用户信息，修改密匙和启停用状态。并提供各部门用户的管理，对各部门进行用户的添加、删除、修改。

用户管理包括如下子模块：用户管理，角色管理，权限展示：

（1）用户管理：通过系统管理员建立账户，管理账户基本信息；

（2）角色管理：通过系统管理员建立角色，管理角色基本信息；

（3）权限展示：管理员可以在此模块下对各个角色以及用户进行权限展示管理。

用户具有以下主要属性：

（1）UserId：在系统中唯一性；

（2）RoleId：在系统中唯一；

（3）Notes：描述或者补充的信息。

4.4.4 系统工具模块设计

本模块的功能主要是提供给开发人员或者运维人员对系统进行检查和维护升级提供帮助的辅助模块。

系统工具模块包括5个子模块：表单构建，Druid监控，系统监控，API文档，URL拦截：

（1）表单构建：为开发人员提供页面表单快速构建，用组件拖拽的方式实现快速开发；

（2）Druid监控：管理整个系统的数据库连接池以及SQL运行状态，包含查询、查看、导出功能；

（3）系统监控：管理员可以在此模块查看整个系统的运行情况，包含流量查询、访问统计、运行时长查询；

（4）API文档：管理员可以在此模块下查看系统生成的API文档信息，包含查询、查看、导出功能；

（5）URL拦截：管理员对需要系统层面上拦截的访问地址进行管理，包含新增、编辑、查询、查看、删除功能；

4.5 数据库设计

系统中的数据，按照预测的访问评率和数据量，单位时间内数据存量，单位时间内数据增量，数据存储时长等综合考虑设计，可以如下分为三类：

（1）用户操作数据，含用户认证信息；

（2）业务数据，包含对9项基础数据对操作记录；

（3）文件数据；

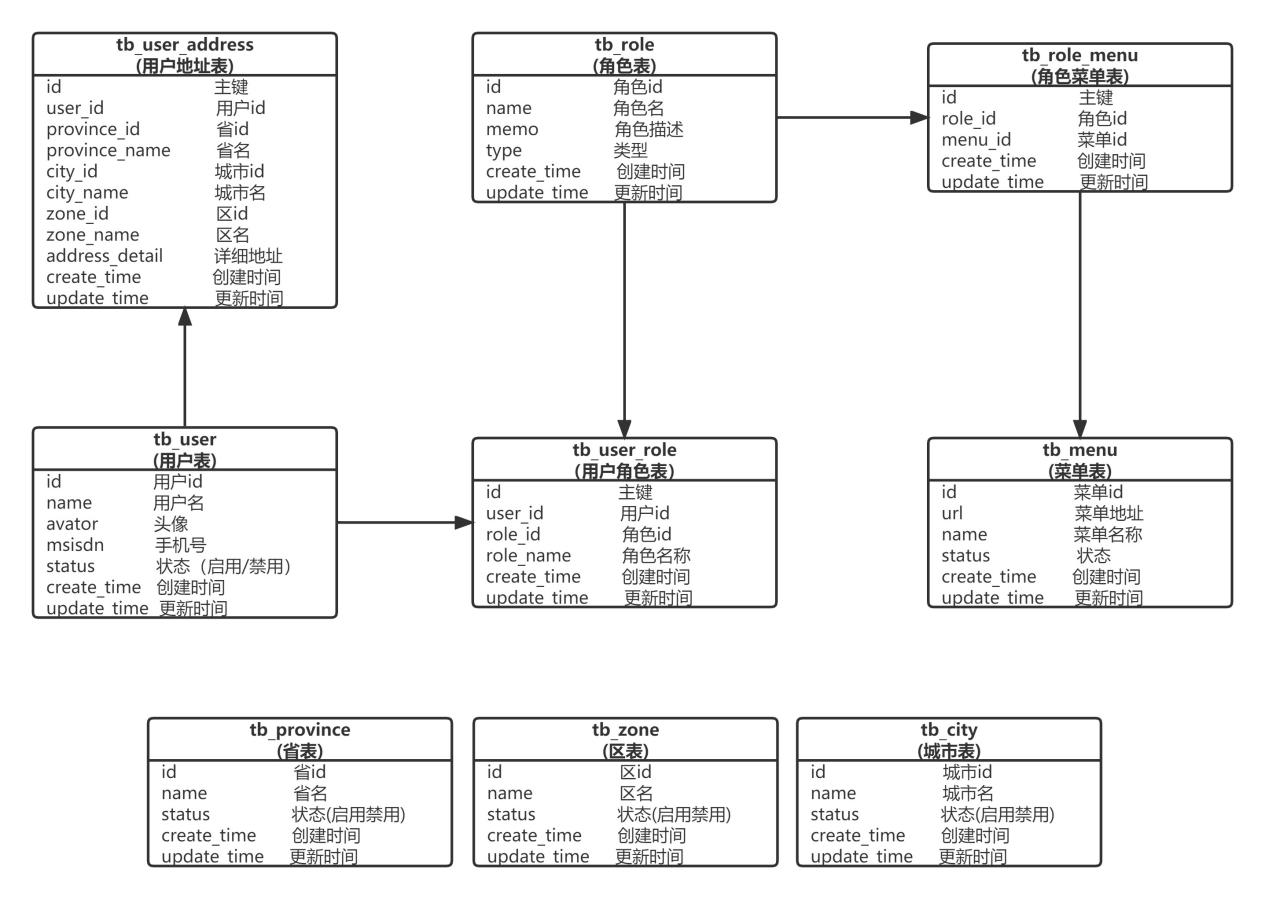


图4-6 用户ER图

如下是部分系统的详细数据库表的详细结构。

表4-1用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Primary | Foreign Key |
| 主键 | id | varchar | 255 | TRUE | FALSE |
| 用户账号 | username | varchar | 255 | FALSE | FALSE |
| 用户密码 | password | varchar | 255 | FALSE | FALSE |
| 昵称 | nickname | varchar | 255 | FALSE | FALSE |
| 部门id | dep\_id | int | 11 | FALSE | FALSE |
| 岗位id | pos\_id | varchar | 255 | FALSE | FALSE |
| 主键 | id | varchar | 255 | TRUE | FALSE |

表4-2部门信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Primary | Foreign Key |
| id | id | int | 11 | TRUE | FALSE |
| 父id | parent\_id | varchar | 255 | FALSE | FALSE |
| 部门名称 | dept\_name | varchar | 255 | FALSE | FALSE |
| 部门负责人 | leader | varchar | 255 | FALSE | FALSE |
| 电话 | phone | varchar | 255 | FALSE | FALSE |
| 邮箱 | email | varchar | 255 | FALSE | FALSE |
| 状态 | status | varchar | 255 | FALSE | FALSE |
| 排序 | order\_num | int | 11 | FALSE | FALSE |

表4-3角色信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Primary | Foreign Key |
| id | id | varchar | 255 | TRUE | FALSE |
| 角色名称 | name | varchar | 255 | FALSE | FALSE |

表4-4角色用户中间表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Primary | Foreign Key |
| 主键 | id | varchar | 255 | TRUE | FALSE |
| 用户id | sys\_user\_id | varchar | 255 | FALSE | FALSE |
| 角色id | sys\_role\_id | varchar | 255 | FALSE | FALSE |

表4-5定时任务调度表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Primary | Foreign Key |
| 日志id | id | varchar | 255 | TRUE | FALSE |
| 任务名称 | job\_name | varchar | 255 | FALSE | FALSE |
| 任务组名 | job\_group | varchar | 255 | FALSE | FALSE |
| 调用目标字符串 | invoke\_target | varchar | 255 | FALSE | FALSE |
| cron执行表达式 | cron\_expression | varchar | 255 | FALSE | FALSE |
| cron计划策略 | misfire\_policy | varchar | 255 | FALSE | FALSE |
| 是否并发执行 | concurrent | varchar | 255 | FALSE | FALSE |
| 任务状态 | status | int | 11 | FALSE | FALSE |

表4-6定时任务调度日志表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Primary | Foreign Key |
| 主键 | id | varchar | 255 | TRUE | FALSE |
| 任务名称 | job\_name | varchar | 255 | FALSE | FALSE |
| 任务组名 | job\_group | varchar | 255 | FALSE | FALSE |
| 调用目标字符串 | invoke\_tar | varchar | 255 | FALSE | FALSE |
| 执行状态 | status | int | 11 | FALSE | FALSE |
| 异常信息 | exception | varchar | 255 | FALSE | FALSE |
| 开始时间 | start\_time | datetime |  | FALSE | FALSE |
| 结束时间 | end\_time | datetime |  | FALSE | FALSE |

表4-7岗位信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Primary | Foreign Key |
| 主键 | id | varchar | 255 | TRUE | FALSE |
| 岗位名称 | post\_name | varchar | 255 | FALSE | FALSE |
| 排序 | order\_num | int | 11 | FALSE | FALSE |
| 状态 | status | int | 11 | FALSE | FALSE |

表4-8权限信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Primary | Foreign Key |
| id | id | varchar | 255 | TRUE | FALSE |
| 权限名称 | name | varchar | 255 | FALSE | FALSE |
| 权限描述 | descripion | varchar | 255 | FALSE | FALSE |
| 授权链接 | url | varchar | 255 | FALSE | FALSE |
| 是否跳转 | is\_blank | int | 255 | FALSE | FALSE |
| 父节点id | pid | varchar | 255 | FALSE | FALSE |
| 权限标识 | perms | varchar | 255 | FALSE | FALSE |
| 目录类型 | type | int | 11 | FALSE | FALSE |
| 菜单图标 | icon | varchar | 255 | FALSE | FALSE |
| 排序 | order\_num | int | 11 | FALSE | FALSE |

5系统实现

5.1 系统管理模块

5.1.1 文件管理

本模块的功能主要是提供文件查询和上传,主要包含增、删、改、查功能,，以下是文件管理模块的部分代码：

|  |
| --- |
| @ApiOperation(value = "分页跳转", notes = "分页跳转")  @GetMapping("/view")  @SaCheckPermission("system:file:view")  public String view(ModelMap model) {  model.put("bucketURL",template.getOssProperties().getEndpoint()+"/"+template.getOssProperties().getBucketName());  return prefix + "/list";  }  /\*\*  \* 修改文件  \*/  @GetMapping("/edit/{id}")  public String edit(@PathVariable("id") String id, ModelMap map)  {  map.put("BucketName",template.getOssProperties().getBucketName());  map.put("sysFile", sysFileService.selectByPrimaryKey(id));  return prefix + "/edit";  } |

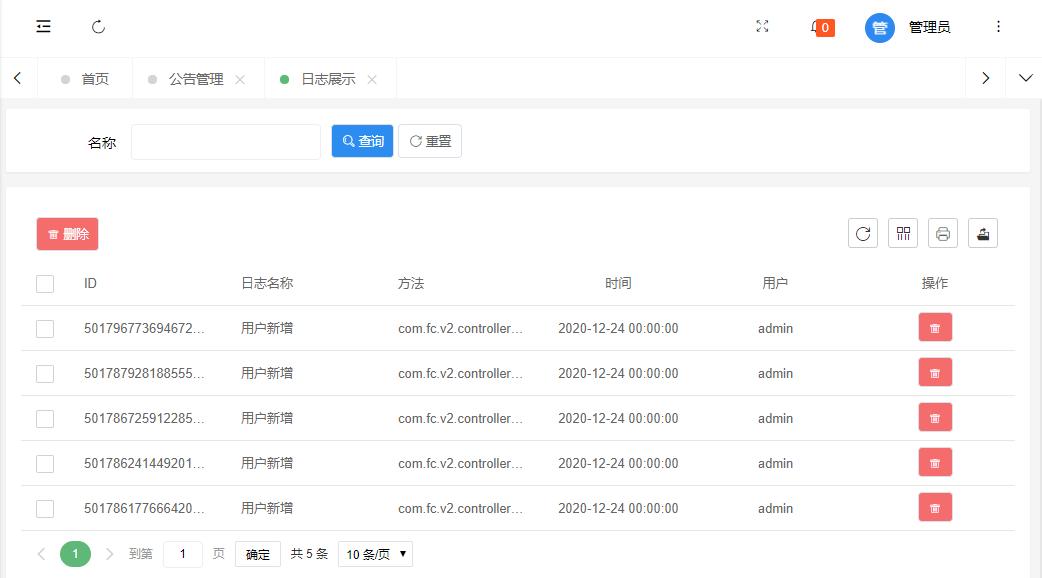
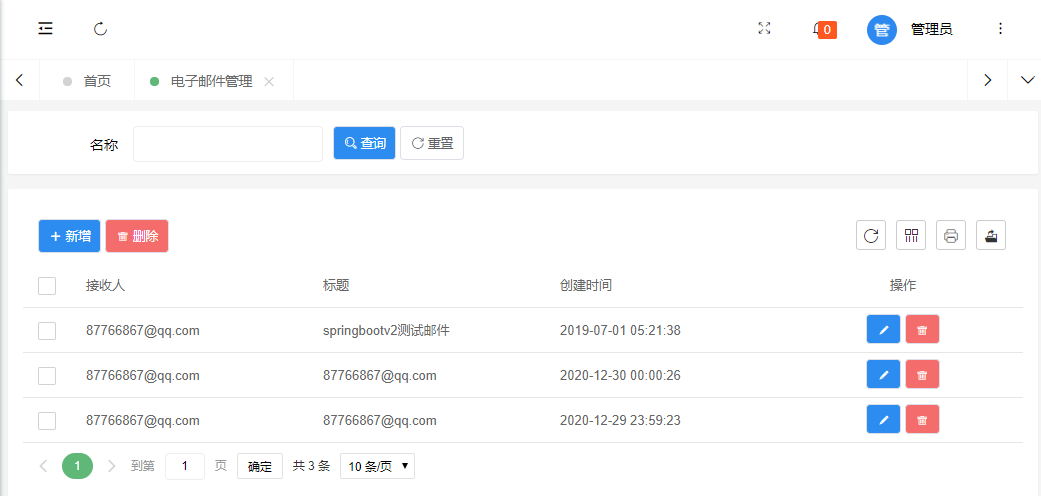


图5-1文件管理查询界面图

5.1.2 电子邮件管理

管理整个系统的电子邮件管理，包含邮件发送、查询、查看、删除功能。在列表页用户可以直观地查看由系统发送的各类邮件的统计情况，为后续其他统计分析提供数据支撑，以下是电子邮件管理的部分代码：

|  |
| --- |
| /\*\*  \* 分页list页面  \* @param tablepar  \* @param searchText  \* @return  \*/  @ApiOperation(value = "分页查询", notes = "分页查询")  @GetMapping("/list")  @SaCheckPermission("system:email:list")  @ResponseBody  public ResultTable list(Tablepar tablepar, String searchText){  PageInfo<TSysEmail> page=tSysEmailService.list(tablepar,searchText) ;  return pageTable(page.getList(),page.getTotal());  }  //@Log(title = "新增邮件", action = "1")  @ApiOperation(value = "新增", notes = "新增")  @PostMapping("/add")  @SaCheckPermission("system:email:add")  @ResponseBody  public AjaxResult add(@RequestBody TSysEmail tSysEmail,Model model) throws Exception{  int b=tSysEmailService.insertSelective(tSysEmail);  if(b>0){  //发送邮件  SimpleEmailUtil.sendEmail(tSysEmail);  return success();  }else{  return error();  }  }  //@Log(title = "删除邮件", action = "1")  @ApiOperation(value = "删除", notes = "删除")  @DeleteMapping("/remove")  @SaCheckPermission("system:email:remove")  @ResponseBody  public AjaxResult remove(String ids){  int b=tSysEmailService.deleteByPrimaryKey(ids);  if(b>0){  return success();  }else{  return error();  }  } |

图5-2电子邮件管理查询界面图

5.1.3 日志管理

日志管理主要展示系统内部的日志信息；管理员可以在此模块下对各个模块的日志进行查看以及查询、导出，方便开发人员可以根据日志信息针对系统进行调优改错，以下是日志管理的部分代码：

|  |
| --- |
| @ApiOperation(value = "分页查询", notes = "分页查询")  @GetMapping("/list")  public Object list(Tablepar tablepar,String searchText){  PageInfo<SysFile> page=sysFileService.list(tablepar,searchText) ;  return pageTable(page.getList(),page.getTotal());  }  @DeleteMapping("/remove")  public AjaxResult remove(String ids){  int b=sysFileService.deleteByPrimaryKey(ids);  if(b>0){  return success();  }else{  return error();  }  } |

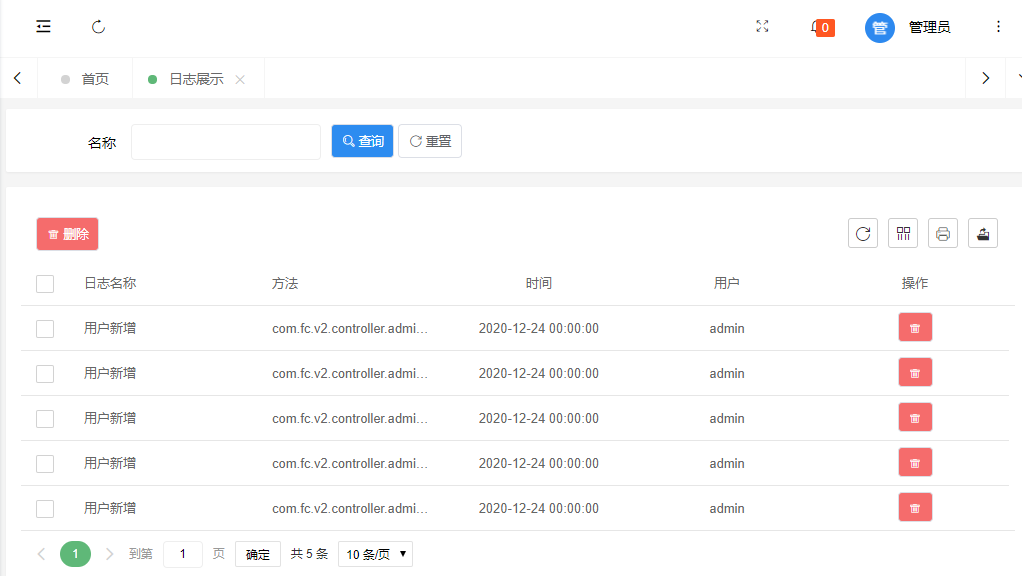
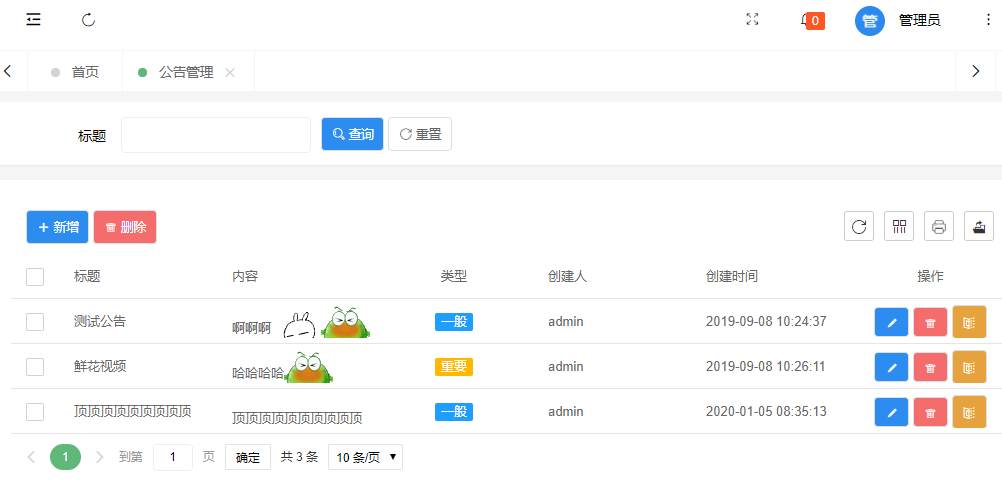


图5-3日志管理界面图

5.1.4 公告管理

系统管理员在此模块下对发布的各类公告进行管理，包含发布公告，包含发布、查询、查看、删除（撤销）功能在线增删改查，前端进行同步更新在线用户可查收到系统广播，离线用户在登录后可在公告模块下查看发布的公告。以下是公告管理的部分代码：

|  |
| --- |
| //@Log(title = "公告集合查询", action = "111")  @ApiOperation(value = "分页查询", notes = "分页查询")  @GetMapping("/list")  @SaCheckPermission("gen:sysNotice:list")  @ResponseBody  public ResultTable list(Tablepar tablepar, String searchText){  PageInfo<SysNotice> page=sysNoticeService.list(tablepar,searchText) ;  return pageTable(page.getList(),page.getTotal());  }  \*/  //@Log(title = "公告新增", action = "111")  @ApiOperation(value = "新增", notes = "新增")  @PostMapping("/add")  @SaCheckPermission("gen:sysNotice:add")  @ResponseBody  public AjaxResult add(SysNotice sysNotice){  int b=sysNoticeService.insertSelective(sysNotice);  if(b>0){  return success();  }else{  return error();  }  } |

图5-4公告管理界面图

5.1.5 字典管理

字典管理模块是对整个系统中各类基础字典表数据，进行统一的维护管理。管理员登录系统后，用户可对数据字典中各类数据类别进行统一管理。可对数据字典的数据类别目录进行添加、修改、删除操作，以下是字典管理的部分代码。

|  |
| --- |
| @ApiOperation(value = "分页查询", notes = "分页查询")  @GetMapping("/list")  @SaCheckPermission("system:dictData:list")  @ResponseBody  public Object list(Tablepar tablepar,String searchText,String dictId){  PageInfo<TSysDictData> page=tSysDictDataService.list(tablepar,searchText,dictId) ;  return pageTable(page.getList(),page.getTotal());  }  @ApiOperation(value = "新增", notes = "新增")  @PostMapping("/add")  @SaCheckPermission("system:dictData:add")  @ResponseBody  public AjaxResult add(TSysDictData tSysDictData, Model model){  int b=tSysDictDataService.insertSelective(tSysDictData);  if(b>0){  return success();  }else{  return error();  }  } |



图5-5字典管理界面图

5.1.6 定时任务

管理员在此模块下可根据实际需求通过cron的操作实现对定时任务的执行时间、执行策略等进行配置。本模块包含添加、修改、删除操作，以下是定时任务管理的部分代码。

|  |
| --- |
| @ApiOperation(value = "定时任务调度list", notes = "定时任务调度list")  @GetMapping("/list")  @SaCheckPermission("gen:sysQuartzJob:list")  @ResponseBody  public ResultTable list(Tablepar tablepar, String searchText){  PageInfo<SysQuartzJob> page=sysQuartzJobService.list(tablepar,searchText) ;  return pageTable(page.getList(),page.getTotal());  }  @ApiOperation(value = "删除", notes = "删除")  @DeleteMapping("/remove")  @SaCheckPermission("gen:sysQuartzJob:remove")  @ResponseBody  public AjaxResult remove(String ids){  int b=sysQuartzJobService.deleteByPrimaryKey(ids);  if(b>0){  return success();  }else{  return error();  }  } |



图5-6定时任务管理界面图

5.1.7 定时任务日志

此模块主要是对定时任务执行时产生的日志信息进行动态展示。本模块包含查询、查看、导出操作，方便对定时任务的执行情况进行追踪管理，以下是定时任务日志的部分代码。

|  |
| --- |
| @ApiOperation(value = "分页查询", notes = "分页查询")  @GetMapping("/list")  @SaCheckPermission("gen:sysQuartzJobLog:list")  @ResponseBody  public ResultTable list(Tablepar tablepar, String searchText){  PageInfo<SysQuartzJobLog> page=sysQuartzJobLogService.list(tablepar,searchText) ;  return pageTable(page.getList(),page.getTotal());  }    @ApiOperation(value = "查看详情", notes = "查看详情")  @GetMapping("/detail/{id}")  public String detail(@PathVariable("id") String id,ModelMap modelMap)  {  SysQuartzJobLog log= sysQuartzJobLogService.selectByPrimaryKey(id);  modelMap.put("SysQuartzJobLog", log);  return prefix + "/detail";  }    @ApiOperation(value = "定时任务日志删除", notes = "定时任务日志删除")  @DeleteMapping("/remove")  @SaCheckPermission("gen:sysQuartzJobLog:remove")  @ResponseBody  public AjaxResult remove(String ids){  int b=sysQuartzJobLogService.deleteByPrimaryKey(ids);  if(b>0){  return success();  }else{  return error();  }  } |

图5-7定时任务日志管理界面图

5.2 代码生成模块

5.2.1 代码生成

此模块是本系统的核心模块，用户可以根据系统自动扫描出的数据库表，选择对应的表进行自动生成代码的操作；用户可以自定义生成的包路径，以及直接选择菜单权限。支持自动生成后端代码（java）和前端代码（html），且支持预览前端界面。用户将系统自动生成的文件放入开发工具中可以直接编译使用，方便开发人员快速开发功能模块，以下是核心实现代码。

|  |
| --- |
| @ApiOperation(value = " 代码自动生成全局配置", notes = "代码自动生成全局配置")  @GetMapping("/global")  @SaCheckPermission("system:autocode:global")  public String global(ModelMap modelMap) {  modelMap.put("author", AutoCodeConfig.getConfig().getProperty("author"));  modelMap.put("email", AutoCodeConfig.getConfig().getProperty("author"));  modelMap.put("parentPath", AutoCodeConfig.getConfig().getProperty("parentPath"));  modelMap.put("dictTypes",tSysDictTypeService.selectByExample(null));  return prefix + "/global";  }  @GetMapping("/selectTables")  @ResponseBody  public ResuTree selectTables() {  List<TsysTables> list = generatorService.queryList(null);  List<TsysTablesVo> TreeList = new ArrayList<TsysTablesVo>();  for (int i = 0; i < list.size(); i++) {  TsysTablesVo tablesVo = new TsysTablesVo(i + 1, -1, list.get(i).getTableName(), list.get(i).getEngine(),  list.get(i).getTableComment(), list.get(i).getTableModel(), list.get(i).getCreateTime(),  list.get(i).getTableName() + " > " + list.get(i).getTableComment());  TreeList.add(tablesVo);  }  TsysTablesVo tables = new TsysTablesVo();  tables.setTableModel("all");  tables.setTableAndName("所有表");  tables.setParentId(0);  tables.setId(-1);  TreeList.add(tables);  return dataTree(TreeList);  }  @ApiOperation(value = "根据表查询表字段详情", notes = "根据表查询表字段详情")  @GetMapping("/queryTableInfo")  @ResponseBody  public ResultTable queryTableInfo(String tableName) {  List<BeanColumn> list = generatorService.queryColumns2(tableName);  return pageTable(list, list.size());  } |



图5-8代码生成界面图

5.3 用户管理模块

5.3.1 用户管理

用户分为管理员用户和业务操作员用户。管理员用户又可根据不同的属性分为不同级别管理员，如系统管理员、业务管理员等。能够实现对各类数据的新增、修改、查询，按条件查询等。可根据按用户名，真实姓名查询用户，新增用户，修改用户信息，修改密码和删除用户。可对用户信息进行添加操作，添加的用户信息包括所属机构、登录名、真实姓名、性别、登录密码、确认密码、移动电话、固定电话、电子邮件。用户还可对已经添加的用户信息进行修改、删除的操作。用户也可对已经添加的用户进行角色分配。以下是用户管理部分代码：

|  |
| --- |
| //@Log(title = "分页查询", action = "1")  @ApiOperation(value = "分页查询", notes = "分页查询")  @GetMapping("/list")  @SaCheckPermission("system:user:list")  @ResponseBody  public ResultTable list(Tablepar tablepar){  PageInfo<TsysUser> page=sysUserService.list(tablepar) ;  return pageTable(page.getList(),page.getTotal());  }  @ApiOperation(value = "新增跳转", notes = "新增跳转")  @GetMapping("/add")  public String add(ModelMap modelMap)  {  //添加角色列表  List<TsysRole> tsysRoleList=sysRoleService.queryList();  //部门列表  List<SysDepartment> departments=departmentService.selectByExample(new SysDepartmentExample());  //岗位列表  List<SysPosition> sysPositions=positionService.selectByExample(new SysPositionExample());  //角色  modelMap.put("tsysRoleList",tsysRoleList);  //部门  modelMap.put("departmentsList",departments);  //岗位  modelMap.put("sysPositionsList",sysPositions);  return prefix + "/add";  } |



图5-9用户管理界面图

5.3.2 角色管理

角色管理主要实现角色的增删改查，按条件查询角色。可根据按角色名查询角色，新增角色，修改角色信息，删除角色。为用户分配不同的角色，角色与用户逻辑上是多对多的关系，并对不同角色进行相对应的权限授权。以下是角色管理部分代码：

|  |
| --- |
| //@Log(title = "角色添加", action = "1")  @ApiOperation(value = "新增", notes = "新增")  @PostMapping("/add")  @SaCheckPermission("system:role:add")  @ResponseBody  public AjaxResult add(@RequestBody TsysRole role){  int b=sysRoleService.insertSelective(role);  if(b>0){  return success();  }else{  return error();  }  }  //@Log(title = "删除角色", action = "1")  @ApiOperation(value = "删除", notes = "删除")  @DeleteMapping("/remove")  @SaCheckPermission("system:role:remove")  @ResponseBody  public AjaxResult remove(String ids){  int b=sysRoleService.deleteByPrimaryKey(ids);  if(b>0){  return success();  }else{  return error();  }  } |



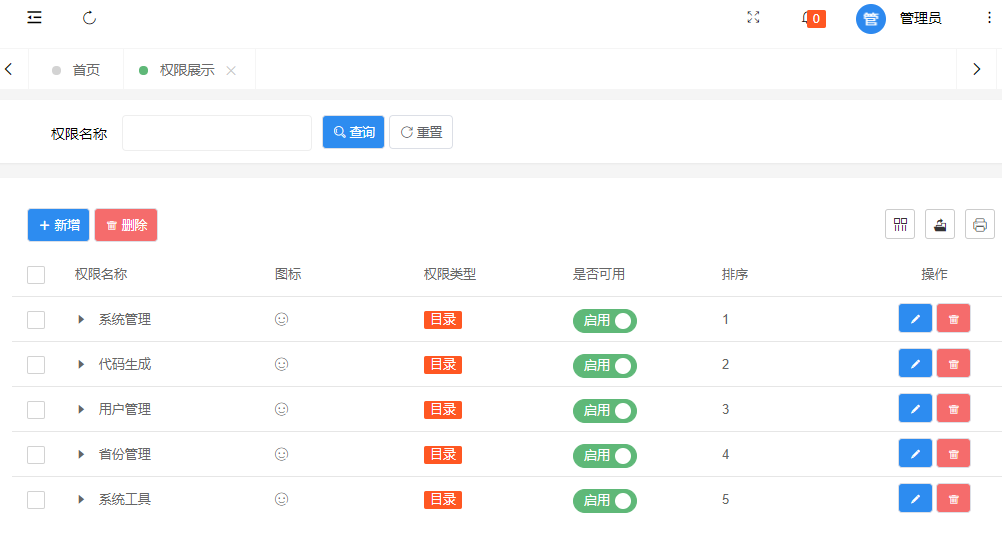
图5-10角色管理界面图

5.3.3 权限展示

用户权限管理模型是现实中的用户、用户的职责和各机构之间的关系在系统中的抽象，它是系统实现统一认证、授权和权限管理的基础[6]。

管理员登录系统后，可进行权限管理。用户可根据查询条件查询、查看已经设置的权限角色信息。用户还可以对用户权限进行添加、修改、删除操作，其中，添加、修改、删除的用户权限信息包括角色名称、角色描述。对于已经添加的用户权限，用户还可以选择后进行权限分配操作。以下是角色管理部分代码：

|  |
| --- |
| //@Log(title = "权限添加", action = "1")  @ApiOperation(value = "新增", notes = "新增")  @PostMapping("/add")  @SaCheckPermission("system:permission:add")  @ResponseBody  public AjaxResult add(@RequestBody TsysPermission tsysPermission){  int b= sysPermissionService.insertSelective(tsysPermission);  if(b>0){  return success();  }else{  return error();  }  }  //@Log(title = "删除权限", action = "1")  @ApiOperation(value = "删除", notes = "删除")  @DeleteMapping("/remove")  @SaCheckPermission("system:permission:remove")  @ResponseBody  public AjaxResult remove(String ids){  int b= sysPermissionService.deleteByPrimaryKey(ids);  if(b==1){  return success("删除成功");  }else if(b==1){  return error("有权限，请解除绑定");  }else if(b==2){  return error("已有角色请解绑");  }else {  return error();  }  } |

图5-11权限展示界面图

5.4 系统工具模块

5.4.1 表单构建

表单构建主要是为了方便开发人员快速地对需要的表单进行组件拖拽开发，实现了表单页的快速创建，、在页面上预览效果同时实时的把页面html代码和js代码输入到旁边展示。生成的代码可以直接复制拿到开发工具中使用。以下是表单构件的部分代码：

|  |
| --- |
| <body>  <div class="layui-fluid">  <div class="layui-row layui-col-space10">  <div class="layui-col-md1">  <div class="layui-card nav">  <div class="layui-card-header">短</div>  <div class="layui-card-body">  <button class="pear-btn pear-btn-sm" plain data-size="inline" data-type="text">输入框</button>  <button class="pear-btn pear-btn-sm" plain data-size="inline" data-type="password">密码框</button>  <button class="pear-btn pear-btn-sm" plain data-size="inline" data-type="select">选择框</button>  <button class="pear-btn pear-btn-sm" plain data-size="inline" data-type="checkbox\_a">复选框</button>  <button class="pear-btn pear-btn-sm" plain data-size="inline" data-type="checkbox\_b">开关</button>  <button class="pear-btn pear-btn-sm" plain data-size="inline" data-type="radio">单选框</button>  <button class="pear-btn pear-btn-sm" plain data-size="inline" data-type="textarea">文本域</button>  <button class="pear-btn pear-btn-sm" plain data-size="block" data-type="submit">提交</button>  </div>  </div>  </div>  <div class="layui-col-md6">  <div class="layui-card r-code-js">  <div class="layui-card-header">code</div>  <div class="layui-card-body">  <textarea name="" class="layui-textarea js-show"></textarea>  </div>  </div>  </div>  </div>  </div>  </body> |

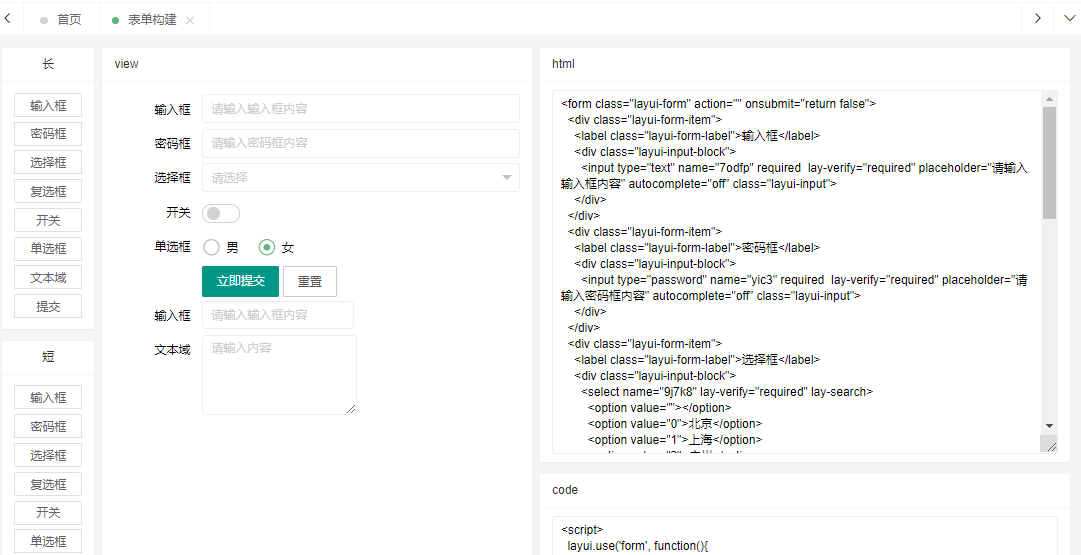
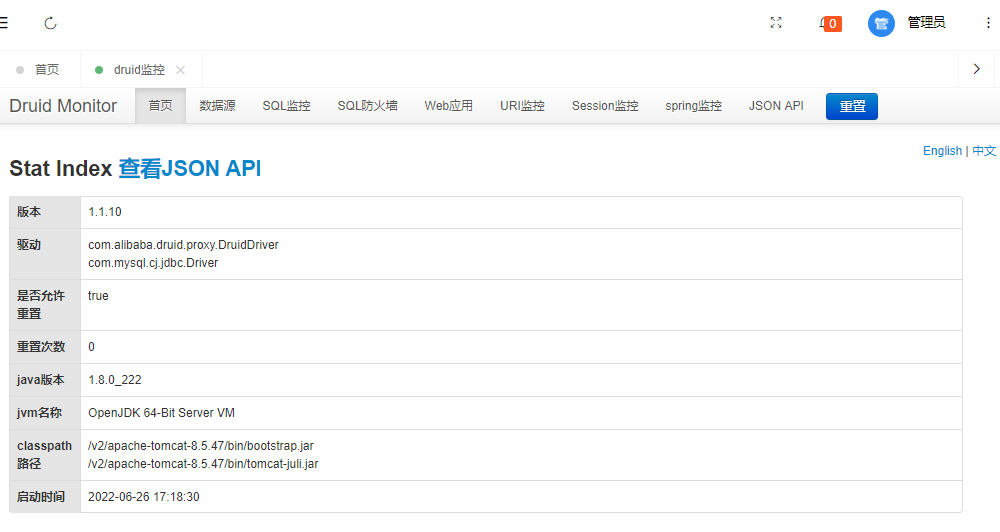


图5-12表单构件界面图

5.4.2 Druid监控

主要是为了监控数据库池连接和 SQL 的执行情况，将监控情况以图表的方式动态展示在界面上，方便管理员对数据库连接池以及SQL的执行情况进行追踪与维护。以下是Druid监控部分代码：

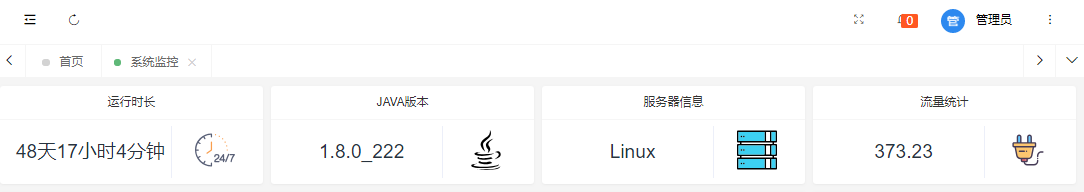
|  |
| --- |
| /\*\*\*\*\*\*\*\*\*Spring Boot 配置Druid关联URI监控\*\*\*\*\*\*/  @Bean  public FilterRegistrationBean<WebStatFilter> webStatFilter(){  FilterRegistrationBean<WebStatFilter> bean = new FilterRegistrationBean<WebStatFilter>();  bean.setFilter(new WebStatFilter());  //exclusions：设置哪些请求进行过滤排除掉，从而不进行统计  Map<String, String> initParams = new HashMap<>(16);  initParams.put("exclusions", "\*.js,\*.css,/druid/\*,bc/\*");  bean.setInitParameters(initParams);  //"/\*" 表示过滤所有请求  bean.setUrlPatterns(Collections.singletonList("/\*"));  return bean;    }  /\*\*\*\*\*\*\*\*\*Spring Boot 配置Druid关联spring监控\*\*\*\*\*\*/  @Bean  public JdkRegexpMethodPointcut druidStatPointcut() {  JdkRegexpMethodPointcut pointcut = new JdkRegexpMethodPointcut();  pointcut.setPattern("com.fc.v2.mapper.\*");  return pointcut;  } |

图5-13 Druid监控界面图

5.4.3 系统监控

系统监控模块主要是将整个系统的运行状态以图表的方式进行动态展示，主要包含系统运行时长、Java版本、服务器基本信息、流量统计、访问量、系统公告等几大模块。可以让管理员直接在系统中查看这些信息，而不是需要再去其他统计系统中查看。以下是系统监控部分代码：

|  |
| --- |
| /\*\*  \* 展示页面  \*  \* @param model  \* @return  \* @author fuce  \*/  @ApiOperation(value = "展示页面", notes = "展示页面")  @GetMapping("/view")  @SaCheckPermission("system:service:view")  public String view(ModelMap model) {  List<SysNotice> sysNotices = sysNoticeService.getNEW();  List<TsysOperLog> sysOperLog = sysOperLogService.getNEW();  if (sysNotices == null || sysNotices.size() <= 0) {  SysNotice sysNotice = new SysNotice();  sysNotice.setId("0");  sysNotice.setTitle("暂无公告");  sysNotice.setCreateTime(new Date());  sysNotices.add(sysNotice);  }  long nd = 1000 \* 24 \* 60 \* 60;  long nh = 1000 \* 60 \* 60;  long nm = 1000 \* 60;  long ns = 1000;  // 获取时间差  long diff = oshiMonitor.getJvmInfo().getUptime();  // 计算差多少天  long day = diff / nd;  // 计算差多少小时  long hour = diff % nd / nh;  // 计算差多少分钟  long min = diff % nd % nh / nm;  // 计算差多少秒//输出结果  //long sec = diff % nd % nh % nm / ns;  model.addAttribute("runtime", day + "天" + hour + "小时" + min + "分钟");  model.addAttribute("service", oshiMonitor);  model.addAttribute("sysNotices", sysNotices);  model.addAttribute("sysOperLog", sysOperLog);  return prefix + "/list";  } |

 图5-14 系统监控界面图

5.4.4 API文档

在系统的各个接口处添加对应的注解，通过注解自动生成的接口的相应信息包含接口描述、请求参数、响应参数、响应状态等信息，方便直观地展示整个接口所需要的信息。可自动生成word下载查看也可以在线网页查看。以下是API文档实现部分代码：

|  |
| --- |
| @Bean(value = "defaultApi2")  public Docket defaultApi2() {  Contact contact=new Contact("v2","https://gitee.com/bdj/SpringBoot\_v2", "87766867@qq.com");  Docket docket=new Docket(DocumentationType.SWAGGER\_2)  .apiInfo(new ApiInfoBuilder()  //.title("swagger-bootstrap-ui-demo RESTful APIs")  .description("SpringBoot\_v2项目是努力打造springboot框架的极致细腻的脚手架")  //服务条款Url  .termsOfServiceUrl("https://gitee.com/bdj/SpringBoot\_v2/blob/master/LICENSE")  .contact(contact)  .version(v2Config.getVersion())  .build())  //分组名称  .groupName("v2")  .select()  //这里指定Controller扫描包路径  .apis(RequestHandlerSelectors.basePackage("com.fc.v2.controller"))  .paths(PathSelectors.any())  .build();  return docket;  }  } |

图5-15 API文档界面图

5.4.5 URL拦截

在粗模块下可以根据自己的需求在此模块下对URL拦截校验进行管理包含新增、删除、编辑、查看、查询功能。以下是URL拦截部分代码：

|  |
| --- |
| @ApiOperation(value = "新增", notes = "新增")  @PostMapping("/add")  @SaCheckPermission("gen:sysInterUrl:add")  @ResponseBody  public AjaxResult add(SysInterUrl sysInterUrl){  int b=sysInterUrlService.insertSelective(sysInterUrl);  if(b>0){  return success();  }else{  return error();  }  }    /\*\*  \* 修改保存  \*/  @ApiOperation(value = "修改保存", notes = "修改保存")  @SaCheckPermission("gen:sysInterUrl:edit")  @PostMapping("/edit")  @ResponseBody  public AjaxResult editSave(SysInterUrl record)  {  return toAjax(sysInterUrlService.updateByPrimaryKeySelective(record));  } |

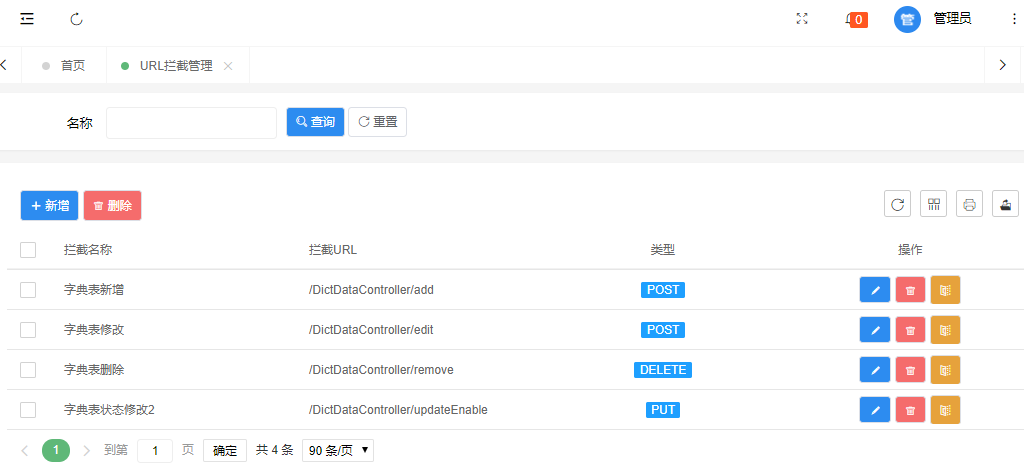


图5-16 URL拦截界面图

6系统测试用例

6.1 系统测试方法

在经历了上述的程序设计以及功能开发过程后，接下来是系统的测试工作，这是软件开发过程中最重要的也是最后一步，是必须认真对待的重要步骤。因此，为规避这类风险，可以在软件开发的所有阶段对已经完成开发的功能模块进行测试。

系统通过各种测试，以确定系统软件是否满足原始用户提供的各种功能[7]。这是最重要的系统要求，可以通过专业的测试方法如黑匣子测试之类的方法来完成。并且要求系统具有良好的可扩展性。

通常，标准系统测试的步骤如下：

（1）测试系统的用例，通过输入仿真数据来测试是否已验证了试用条件的使用，包括详细记录测试和结果；

（2）与表单填报的有关信息，不仅包含必填信息性数据，而且还参与一些有效数据的边界测试确保系统可以检测到该数据边界，自动并显著提高系统性能；

（3）使用代码审查的结果，首先了解执行代码的流程，同时检查输入数据的合法性，以确保代码无重大bug。

6.2 系统功能测试

在分三个层次测试：功能全覆盖测试

表6-1测试信息表1

|  |  |
| --- | --- |
| 内容1 | 业务流测试 |
| 测试目标 | 确保整个系统的基本业务的路径能正确执行。 |
| 测试技术和方法 | 用例设计用基本路径覆盖方法，执行是手工执行。 |
| 完成标准 | * 严重级别为1级的缺陷修复率达到100%； * 严重级别为2级的缺陷修复率不少于95%； * 严重级别为3级的缺陷修复率不少于80%； * 严重级别为4级的缺陷修复率不少于70%。 |
| 需考虑的特殊事项 | 无 |

表6-2测试信息表2

|  |  |
| --- | --- |
| 内容2 | 性能测试 |
| 测试目标 | 测试系统的并发用户/在线用户数、系统查看页面响应、业务办理页面响应、查询统计操作等响应时间。 |
| 测试技术和方法 | 支持200 人的并发用户访问。  查看页面响应时间不超过4秒。  系统页面响应时间不超过6秒。  的查询统计操作的响应时间不超过10秒。  业务报表生成时间不超过10秒。  业务报表生成时间不超过30秒。 |
| 完成标准 | * 严重级别为1级的缺陷修复率达到100%； * 严重级别为2级的缺陷修复率不少于95%； * 严重级别为3级的缺陷修复率不少于80%。 * 严重级别为4级的缺陷修复率不少于70%。 |
| 需考虑的特殊事项 | 无 |

表6-3测试信息表3

|  |  |
| --- | --- |
| 内容3 | 功能点之间的接口测试以及风险点的再测。 |
| 测试目标 | 确保整个系统的所有功能点之间的接口的正确性。 |
| 测试技术和方法 | 同功能点测试 |
| 完成标准 | * 严重级别为1级的缺陷修复率达到100%； * 严重级别为2级的缺陷修复率不少于95%； * 严重级别为3级的缺陷修复率不少于80%。 * 严重级别为4级的缺陷修复率不少于70%。 |
| 需考虑的特殊事项 | 无 |

总结

本文使用软件工程理论来增强系统的业务运作，为广大基层开发人员提供方便快捷搭建系统功能完善的脚手架系统，对管理类系统的开发起到重要作用。在所需的步骤中，使用各类图表设计系统，并从各个维度，各个方面和不同的层次对系统进行分析充分发挥系统的优点。在设计过程中，首先考虑整个系统架构的设计是否合理满足易用性、安全性等条件，然后从系统前端展示级别，结合业务模块和产品设计来设计主要的系统业务流程以及功能模块内容。主要的系统功能包括：系统管理模块、代码生成模块、用户管理模块、系统工具模块。本文以逐章的形式概述了系统的结构，并详细介绍了总体架构设计、模块功能设计、数据库设计等。

由于只是兼职开发这个项目精力有限，因此需要完整的完成整个系统的研发工作还需要在以下方面进行改进工作：

（1）完善系统设计

本文使用基于内容的方法，最大程度地降低了操作，维护和验证系统规范的难度，尽管通过多维度（论坛、贴吧、QQ群）的和广大程序员交流，收集了主要的重复开发工作，但是这只是一小部分想要更完善的系统功能将要后续不断的迭代。

（2）开发一套全面的服务

随着整个互联网产业的发展以及硬件技术的不断突破。对整个系统不管是硬件技术还是软件技术大部分都得到解决，从程序开发者的角度进行审视后，对于某些用户提出的建议不仅是切实可行的，而且提出的建议也确实是是能给实际工作带来便捷的，而且从技术层面上来讲完全可以实现。所以本人将会在下一个运营阶段将与系统用户和广大程序员进行进一步讨论，专注用户体验和系统功能性的建全而努力。

参考文献

1. M.J.Taylor,J.McWilliam.H.Forsyth,S.Wade.Methodologies and website development:asurvey of practice[J].Information and Software Technology,2002,1(44):381-391.
2. 汤阳,田欣.基于B/S结构的信息数据库设计与实现[J].现代情报,2006(08):73-74.
3. 苗雪兰.数据库系统原理及应用（第三版）[M].北京：机械工业出版社，2007

1. [Apache Shiro安全框架在技术转移服务系统中的应用](https://www.zhangqiaokeyan.com/academic-journal-cn_computer-modernization_thesis/0201241682293.html) [J] [. 杨运平](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E6%9D%A8%E8%BF%90%E5%B9%B3&option=202) [,吴智俊](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E5%90%B4%E6%99%BA%E4%BF%8A&option=202) [. 计算机与现代化](https://www.zhangqiaokeyan.com/journal-cn-8911/). 2014,003

1. [基于SpringBoot和Vue框架的数据库原理网站设计与实现](https://www.zhangqiaokeyan.com/academic-journal-cn_digital-community-smart-home_thesis/0201291476855.html) [J] [. 单树倩](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E5%8D%95%E6%A0%91%E5%80%A9&option=202) [,任佳勋](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E4%BB%BB%E4%BD%B3%E5%8B%8B&option=202) [. 电脑知识与技术](https://www.zhangqiaokeyan.com/journal-cn-8877/). 2021,030
2. 何鼎权,胡辉,严家成.基RBA的通用权限管理系统[J].电脑知识与技术,2020,16(33):97-102.

1. [SeC‐SDWSN: Secure cluster‐based SDWSN environment for QoS guaranteed routing in three‐tier architecture](https://www.zhangqiaokeyan.com/academic-journal-foreign_international-journal-communication-systems_thesis/0204119781415.html) [J] . [Vishnu V. M.,](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=Vishnu%20V.%20M.&option=202) [Manjunath P.](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=Manjunath%20P.&option=202)[International journal of communication systems .](https://www.zhangqiaokeyan.com/journal-foreign-1453/) 2019,14

致谢

首先，要感谢我的恩师们，在您们的谆谆教诲中，我深刻地体会到了知识的重要性，以及人生的意义;她教会了我如何去发现知识，了解知识，掌握知识。也让我明白了，什么是真正的学无止境；其次我要感谢我的家人，是他们的支持与鼓励让我能完成学业。在今后的道路中，我会更加努力，去证明自己超越自己，力争不做 最好制作更好！最后，我要感谢我的同窗好友，是你们给了我生活的乐趣，也是你们坚定了我前进的道路，在未来的日子里，我会把你们铭刻在心，每当我遇到挫折的时候，我就会回想我们在一起的美好，那些时光，总是充满着欢声笑语。