

基于 Activiti 的低代码办公平台设计与开发^{*}

付明超 林宁

南宁学院信息工程学院, 南宁, 530200

摘要 低代码办公平台由两大模块组成, 分为低代码设计模块和基本信息模块。低代码设计模块为用户提供设计模板, 可以线上创建业务对象、表单、数据对象和流程模型等页面, 其中业务对象是通过 DDL 语言进行数据库操作, 流程是使用 Activiti 流程引擎进行管理, 而前端使用 Bpmn-js 进行流程模型设计, 表单和数据对象通过 Json 保存页面信息, 通过 Vue 框架的 Element UI 进行数据回显。基本信息模块主要是权限管理和系统管理两部分, 提供人员的信息管理、系统参数、系统监控信息、日志操作信息等内容。测试的结果表明, Activiti 低代码办公平台功能齐全, 都以傻瓜式操作进行配置, 设计页面的人员不将局限于专业人才, 减少了开发人才的投入。

关键字 流程, 表单, 业务对象, 数据对象

Design And Development of Low Code Office Platform Based on Activiti

Fu Mingchao Lin Ning

School of Information Engineering
Nanning University
Nanning 530200, China;

Abstract—The low code office platform is composed of two modules, which are divided into low code design module and basic information module. The low code design module provides users with design templates, which can create pages such as business objects, forms, data objects and process models online. The business objects are operated by the DDL language, the processes are managed by the activiti process engine, and the front end uses BPMN JS to design the process model. The forms and data objects save page information through JSON, and the data is echoed through the Vue framework's element UI. The basic information module is mainly composed of permission management and system management, providing personnel information management, system parameters, system monitoring information, log operation information, etc. The test results show that activiti low code office platform has complete functions and is configured by fool operation. The personnel who design the page will not be limited to professionals, which reduces the investment of development talents.

Key word—Process, form, business object, data object

1 引言

随着科技发展, 办公自动化进入了全新的时代。第七次全国人口普查显示, 总人口 1,443,497,378 人, 增长 5.38%, 办公基数随之增大, 同时政府、企业、高校内部交流越来越频繁, 传统的办公方式无法满足当今社会的需求。由于办公流程的复杂性、审批业务的耗时性以及历史文件的管理性, 成为了传统办公的一大痛点。

该平台将数据库、表单、数据对象和流程等管理使用低代码/无代码开发, 使系统具有开发简单, 少量代码即可完成复杂的办公业务, 该平台为中小型企业

部署办公系统节约开发和维护成本, 而且可以减少专业人才的投入需求, 建立自动化办公系统是顺应时代发展的要求。

2 相关技术研究

本办公系统是基于若依系统的基础上进行设计的, 系统环境为 Java EE 8、Servlet 3.0 和 Apache Maven, 主框架为 Spring Boot、Spring Framework 和 Spring Security, 持久层为 Apache MyBatis、Hibernate Validation 和 Alibaba Druid, 视图层使用的技术为 Vue、Element-ui 等技术, 数据采用 MYSQL 进行存储, 使用 HTTP 请求方式获取接口。

Spring Security 为基于 Java EE 的企业应用程序提

^{*} **基金资助:** 本文得到广西高校中青年教师科研基础能力提升项目 (2021KY1805) 资助。

通讯作者: 林宁, 副教授, bgy_2009@163.com

提供一个全面的解决方案,为应用程序提供一个有用的并且高度可配置的安全系^[3]。它是 Spring 一站式部署的安全框架,它提供了上下文配置,减少了配置复杂性的优点,充分利用了 Spring 的依赖注入和面向切片的功能,减少添加安全控制而导致编写大量重复代码的工作。

Maven 作为一个项目管理工具,主要服务于基于 Java 平台的项目构建、依赖管理和项目信息管理^[4]。它主要是通过中央信息片断来管理项目的构建、报告和文档的工具,它的编译、测试、打包、检查、安装和部署等功能大大提高了开发人员的工作效率,将开发人员的中心转移到业务需求上。开发人员可以使用命令进行构建项目,可以进行 Jar 包的依赖配置,它可以构建自己的远程 Maven 仓库,并与中央仓库进行连接,可以进行统一的错误报告管理。它具有项目信息的一致性站点、向后兼容性和并行构建等优点,深受企业和程序员的喜爱。

Apache MyBatis 是一个持久层框架,它拥有定制化 SQL 语句、存储过程以及映射等功能。它不仅简化了获取数据的检索过程,而且将持久层代码简单化,它可以通过 xml 的配置和 mapper 的注解进行数据映射。MyBatis 不仅具有灵活的操作空间,而且将代码进行了解耦操作,即解除 SQL 与代码的耦合度。该框架拥有 xml 标签和动态 SQL 的特点,深受中小企业和程序员的热爱。

MYSQL 是一个比较适合中小型网站开发的数据库,它是一个多线程、多用户、C/S 结构的关系型数据库系统,由于该系统拥有体积小、使用过程简单、开

源低成本、性能普遍偏高、查询数据灵活可靠和查询速度快等特点,将与 Oracle 和 SQL Server 相比肩,对于中小型公司已经绰绰有余。

Vue 是一套构建用户界面的渐进式框架。它采用自底向上增量开发的设计。Vue 的核心库侧重于视图层,并且非常容易学习和项目整合。Vue 拥有数据驱动视图和双向数据绑定的特性。MVVM 是 Vue 实现数据驱动视图和双向数据绑定的核心原理,其中 MVVM 指的是 Model、View 和 View Model,而 View Model 是 MVVM 的核心,是它把当前页面的数据源和页面的结构连接在了一起。

Activiti 是一个比较流行的工作流引擎,可以将复杂的业务问题进行图像化显示,将问题进行模块化抽象,使用 BPMN2.0 进行流程定义,业务流程将会按照预先定义好的流程进行执行,实现了流程系统管理功能。该工作流引擎不仅减少了流程变更升级带来的工作量,而且提高了该系统的健康性和生命周期。

3 系统设计

3.1 系统框架设计

Activiti 低代码办公平台是使用前后端分离的技术,它将增强该系统的代码可维护性,减轻后端服务器的请求压力,具体设计如下:

(1) 后端框架设计

该平台的后端使用 SSM 框架,其中代码结构分为 Controller 层、Service 层和 Mapper 层,具体内容见图 1 所示。

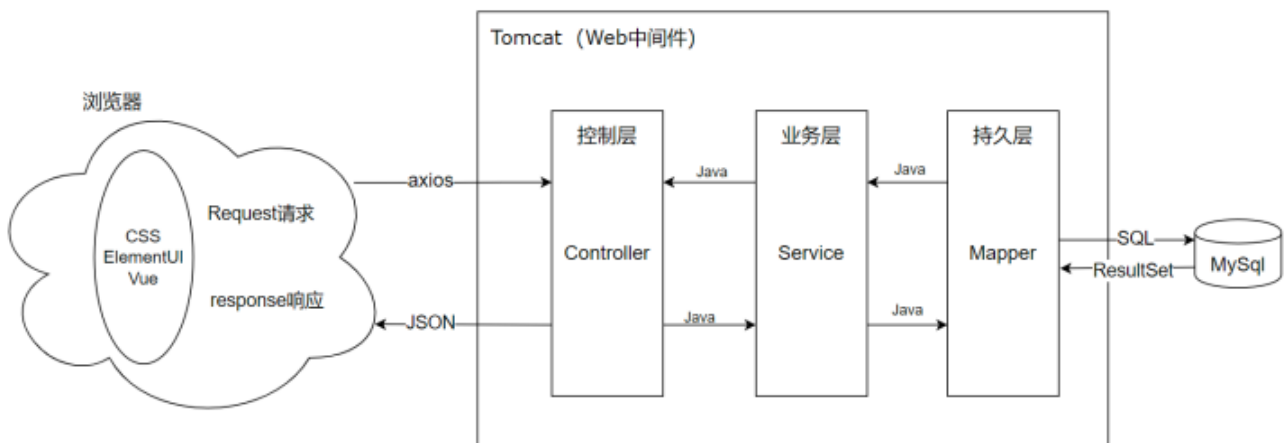


图 1 SSM 框架的架构图

该项目存在父工程和子模块的关系,其中 SleepFish 为父工程,子模块有 7 个,模块之间相互继

承和依赖,关系如图 2 所示,其中各个模块具有不同的作用,具体内容如下:

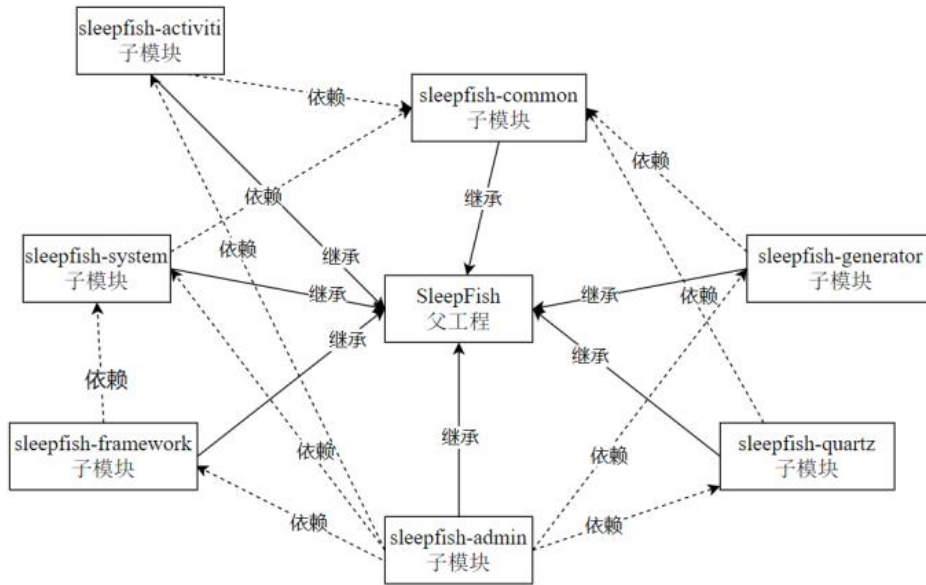


图 2 系统后端模块关系图

sleepfish-admin 主要负责系统管理和权限管理的 Controller 层，主要是负责相应的业务模块流程的控制。

sleepfish-system 主要存放 sleepfish-admin 所需的所有 Service 层接口。

sleepfish-common 是公共基础模块，提供一些公共的基础服务，如 web 层通用数据处理、XSS 过滤处理、异常处理、枚举以及一些通用的 java 工具。

sleepfish-framework 是框架核心模块，如注解实现、系统配置、拦截器、异步处理、权限控制等内容。

sleepfish-generator 和 sleepfish-quartz 分别是代码生成模块和定时任务模块。

(2) 前端框架设计

该系统前端使用目前主流的 MVVM 模式框架 Vue 进行开发，实现多个组件单页“web”应用程序，具体框架结构见图 3 所示。

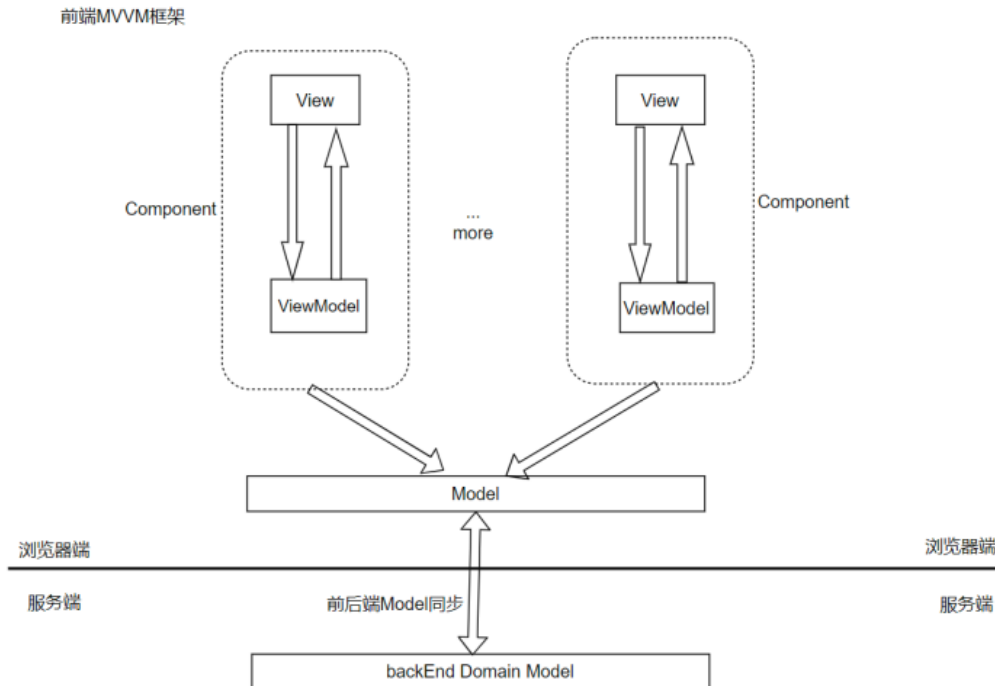


图 3 MVVM 框架图

该项目是由 Vue-cli 进行搭建, 结构分为 12 个模块, 具体内容见表 1 所示。

表 1 前端模块结构分析

文件名称	具体作用
api	接口请求方法
assets	静态资源存储
components	全局公共组件
directive	全局指令
icons	图片
layout	布局
router	路由
store	全局 Store 管理
styles	样式
utils	全局公共方法存储
views	Views 视图展示
plugins	插件

3.2 系统总体结构设计

低代码办公平台分为权限管理、数据管理、表单管理和流程管理等模块, 系统总体结构见图 4 所示。

该平台的模块与模块之间环环相扣, 以流程设计、表单管理和数据对象为例。此模块使用图形化界面, 通过封装的引擎, 将前端绘制的页面自动转化为相应的代码, 对数据库进行操作^[5]。其中平台枢纽在于数据库, 通过数据进行各个模块的连接与配合。表单设计需要进行数据库表绑定, 而数据对象和流程设计需要进行数据库表和表单的绑定, 具体联系和结构详情见图 5 所示。

3.3 系统主要功能设计

(1) 权限管理

权限管理功能分为用户、角色、部门和职位的管理, 其中功能协助紧密, 如用户管理中需要绑定角色、部门和职位, 角色管理也需要菜单权限的绑定, 而部门与部门之间也存在上下级关系, 具体内容见图 6 所示。

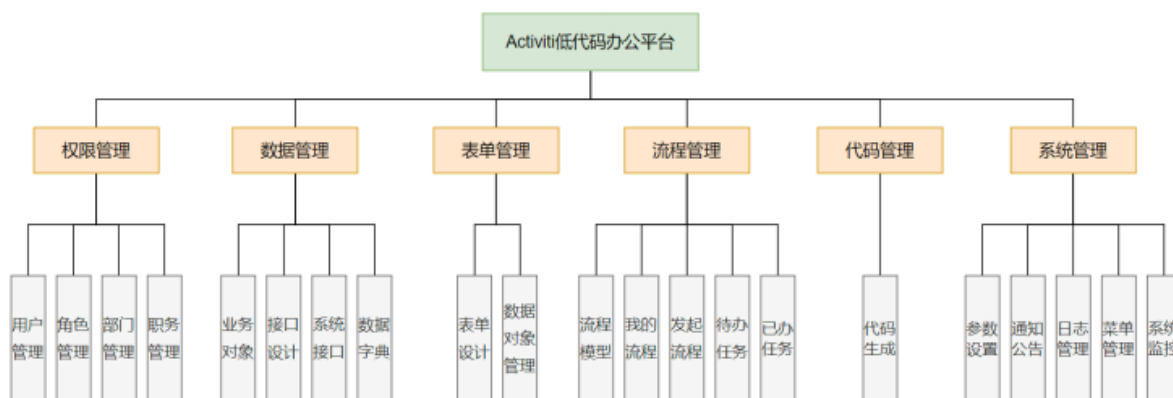


图 4 系统总体结构图

主要代码实现如下:

```

@ApiOperation(value = "获得用户的列表")
@PreAuthorize("@ss.hasPermi('power:user:list')")
@GetMapping("/list")
public TableDataInfo listUser(SysUser sysUser){
    startPage();
    List<SysUser> listUser=
    sysUserService.selectUserList(sysUser);
    return getDataTable(listUser);
}
@ApiOperation(value = "修改数据权限及保存")
@PreAuthorize("@ss.hasPermi('power:role:update')")
@PutMapping("/dataScope")
public AjaxResult dataScopekeep(@RequestBody SysRole sysRole)
{
    roleService.checkRoleAllowed(sysRole); //是否允许操作
    return toAjax(roleService.authDataScope(sysRole));
}

```

(2) 业务对象管理

业务对象与数据库进行桥接, 可以对数据库表进

行基本的 CRUD 操作, 其中新增数据库表时, 用户需要填写基本信息和字段信息, 字段信息可以根据需求进行修改。用户一旦提交成功, 数据库表是未生成状态, 其中基本信息将无法修改, 字段信息可以正常操作。当用户点击生成表时, 将会在数据库创建一个以 tb_开头的表, 详细流程信息见图 7 所示。

主要代码实现如下:

```

@ApiOperation(value = "修改 表-字段 (生成后)")
@PutMapping("/updateColTable")
public AjaxResult updateColTable(TableColumnTran
tableColumnTran){
    if(!tableColumnTran.getColumnName().endsWith("_")){
        tableColumnTran.setColumnName(
        tableColumnTran.getColumnName()+"_");
    }
    String sql=tableColumnTran.updateSql();
    try {
        genTableService.updateTable(sql);
        GenTableColumn genTableColumn=
        new GenTableColumn();
    }
}

```

```
tableColumnTran.setColumnTypes(tableColumnTran
.getColumnTypes()+"("+tableColumnTran.
getLen()+")");
```

```
BeanUtils.copyBeanProp(genTableColumn,
tableColumnTran);
genTableColumn.setJavaField(TextTransFormUtils.
underlineToCamel(tableColumnTran.
getColumnName()));
genTableColumnService.updateGenTableColumn(
```

```
genTableColumn);
return AjaxResult.success();
}
catch (Exception e){
logger.error(e.getMessage(), e);
return AjaxResult.error("修改表字段异常(生成后)["+
e.getMessage() + "]");
}}
```

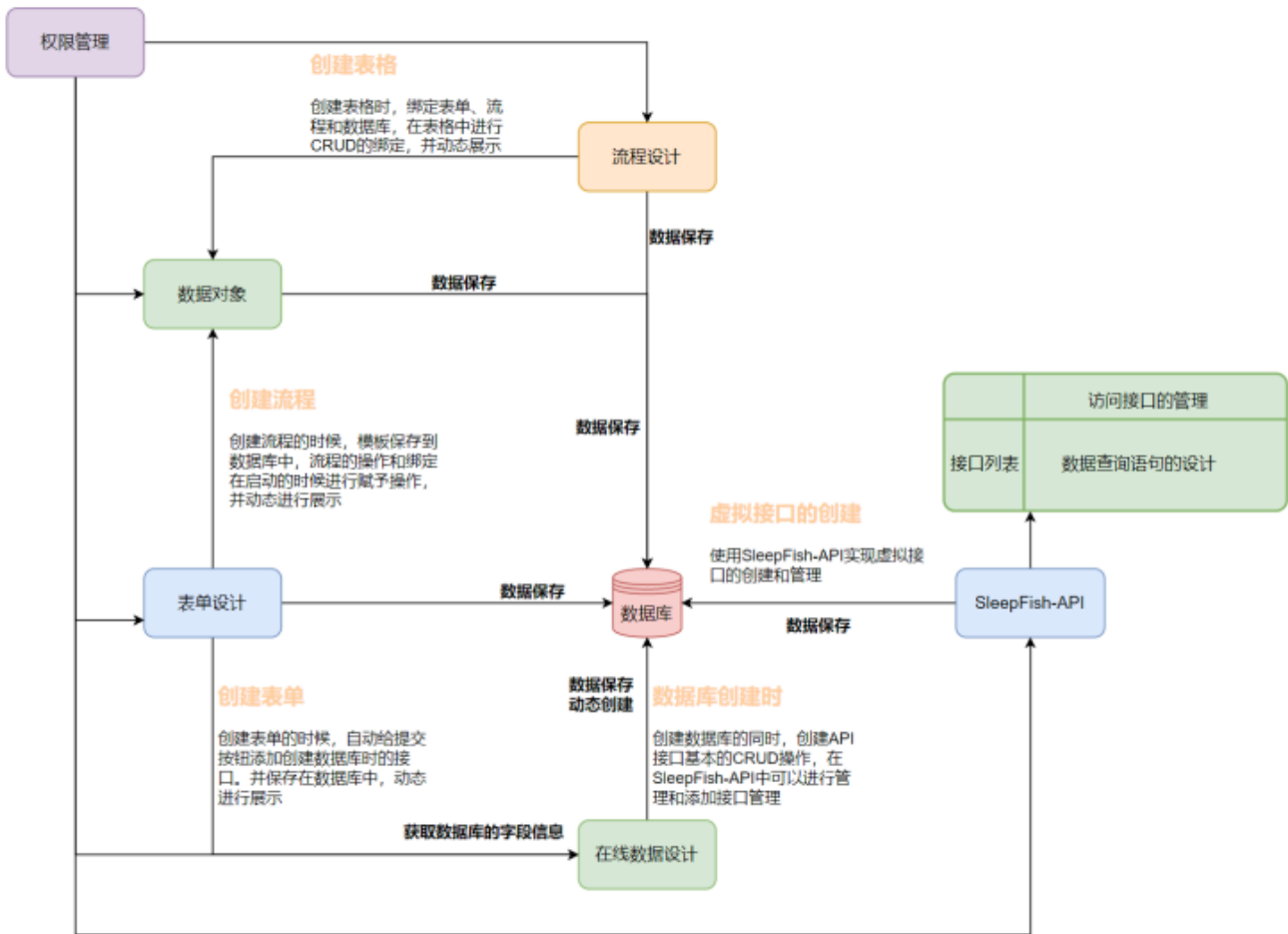


图 5 系统模块联系结构图

(3) 表单管理

表单设计主要负责表单页面的制作，新增页面操作中，用户需要操作表单属性、组件属性和表单控件，首先需要进行表单属性的编辑，如数据库表绑定、表单名的填写，不然用户将无法提交数据。用户可以根据自己的需求进行表单组件的选择，可以点击或者拖拽表单控件栏的组件到操作区，选择好对应的组件后，可以在组件属性栏中进行编辑，如字段名的选择、标题的填写、表单栅格宽度等设置，其中字段名是根据绑定的数据库表进行字段回显，具体详细信息见图 8 所示，用户也可以根据需求进行页面的 CRUD 和菜单添加操作等。

主要代码实现如下：

```
@PostMapping("/create")
@ApiOperation("创建动态表单")
@PreAuthorize("@ss.hasPermi('bpm:form:create')")
public AjaxResult createForm(@Valid @RequestBody
BpmFormCreateReqVO createReqVO) {
return AjaxResult.success(
formService.createDynamicForm(createReqVO));
}
@PostMapping("/update")
@ApiOperation("更新动态表单")
public AjaxResult updateForm(@Valid @RequestBody
BpmFormUpdateReqVO updateReqVO) {
formService.updateForm(updateReqVO);
return AjaxResult.success(true);
}
@Override
public Long createDynamicForm (BpmFormCreateReqVO
formCreateReqVO) {
//检查是否拥有重复的名称和字段
this.checkFields(formCreateReqVO.getFields());
BpmForm bpmForm=new BpmForm();
```

```

BeanUtils.copyBeanProp(bpmForm, formCreateReqVO);
List<BpmForm> listForm = formMapper.selectList();
for (BpmForm form : listForm) {
    if (form.getRemark().equals(bpmForm.getRemark())){
        throw new Error("表单已存在,请换个名称");
    }
}
return formMapper.insert(bpmForm);
}
}

```

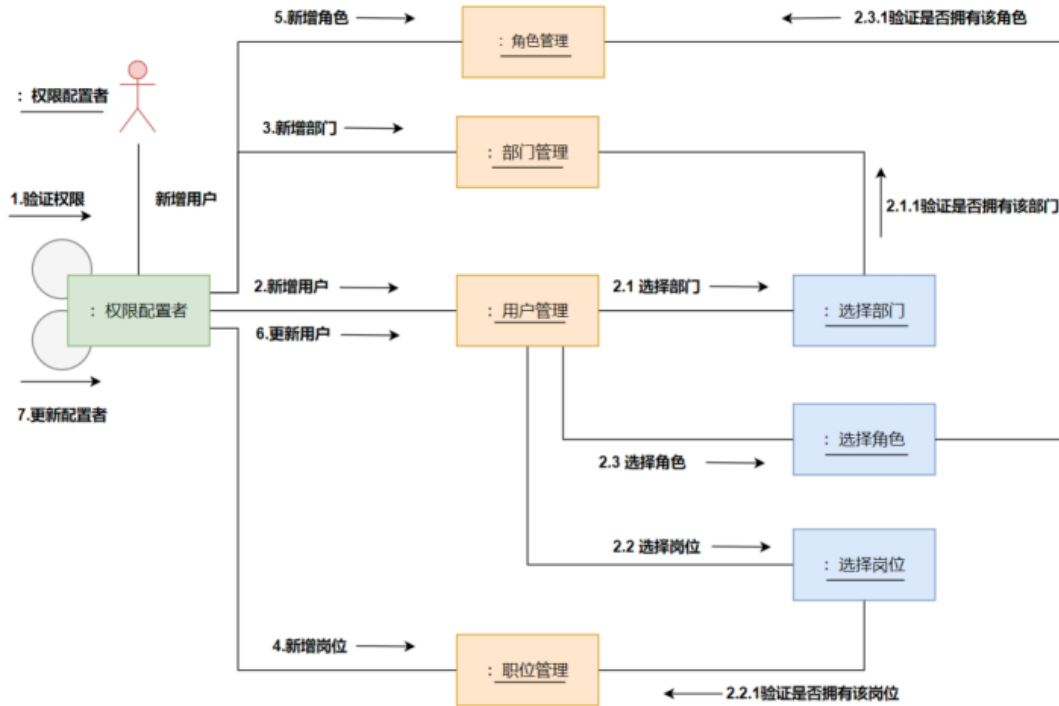


图 6 权限管理功能的协作图

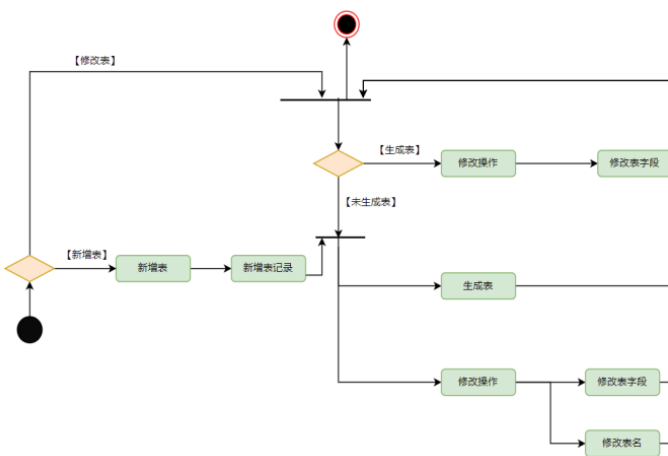


图 7 业务对象业务层活动图

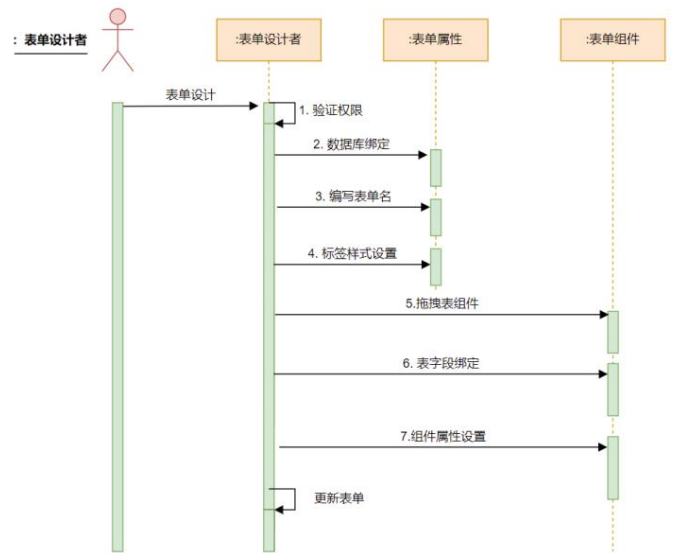


图 8 表单设计功能的时序图

(4) 数据对象管理

数据对象管理主要负责列表页面的制作，新增页面操作中，用户需要操作表格属性、组件属性和表格样式，首先需要进行表格属性的编辑，如数据库表绑定、表格名的填写，不然用户将无法提交数据。用户可以根据自己的需求进行表单的绑定、显示字段的选

择、查询字段的选择等，绑定表单是由数据库表决定其选择对象，其中多个表单页面可以同时绑定同一个表，而这些表单会回显到表单的选项列表里，具体内容见图 9 所示。用户可以根据需求进行页面的增删改查，也可以进行菜单添加操作。

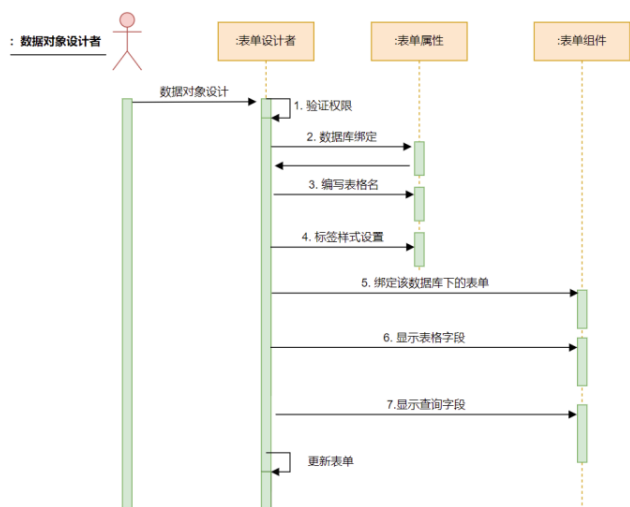


图 9 数据对象管理功能的时序图

主要代码实现如下：

```

@PutMapping("/update")
@ApiOperation("更新数据对象")
public AjaxResult updateForm(@Valid @RequestBody
    BpmFormUpdateReqVO updateReqVO) {
    TableService.updateForm(updateReqVO);
    return AjaxResult.success(true);
}
@GetMapping("/get")
@ApiOperation("通过ID获取数据对象")
public AjaxResult getTable(@RequestParam("id")
    Long tableId) {
    BpmForm table = TableService.getTable(tableId);
    BpmFormRespVO tableVo = new BpmFormRespVO();
    BeanUtils.copyBeanProp(tableVo, table);
    return AjaxResult.success(tableVo);
}
    
```

(5) 流程设计

流程设计功能分为流程模型和个人办公，此模块主要以流程设计为主导，其中流程模型主要负责流程模型的设计，个人办公主要负责流程的操作，如发起流程、审批流程、流程预览等内容，具体功能分析如下：

① 流程模型主要负责流程模板的设计，在新增流程模型中，用户可以选择新建流程模型或者导入流程模型、新建流程模型需要填写流程标识和名称，其次还需要绑定表单，然后进行设计流程，此过程用户需要使用 BPMN 符号进行建模，主要由事件、活动图、网关以及箭头流向组成。当设计流程结束后，用户可进行人员分配，规则可以根据角色、部门、岗位以及用户等进行分配。只有上述过程完成后，部署操作才可以执行，否则会提示完善以上操作，具体操作见图 10 所示。

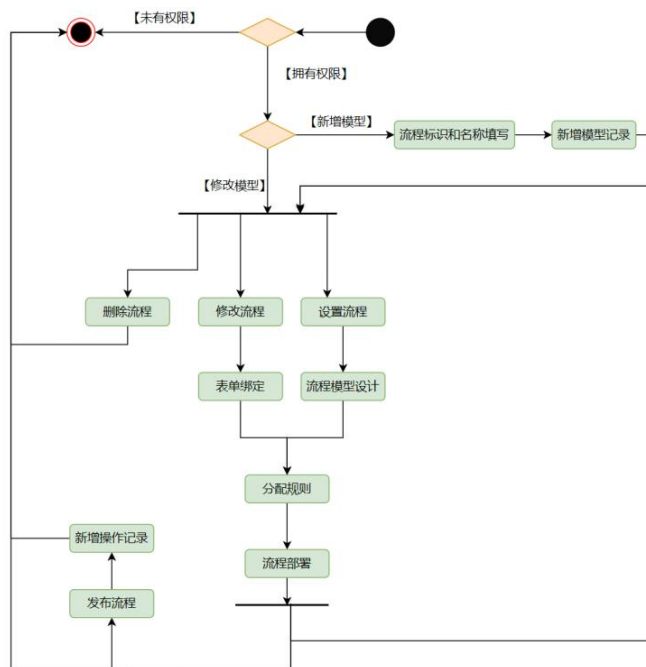


图 10 流程模型功能的活动图

② 个人办公主要负责流程的操作，用户可以根据需求进行流程发布和管理，发起流程的过程中，用户可以选择已有的流程模板进行操作，提交申请信息便可发布流程。如果该流程审批者是本人的话，可以进行干预审批，否则只有负责人才可以审批，具体内容见图 11 所示。

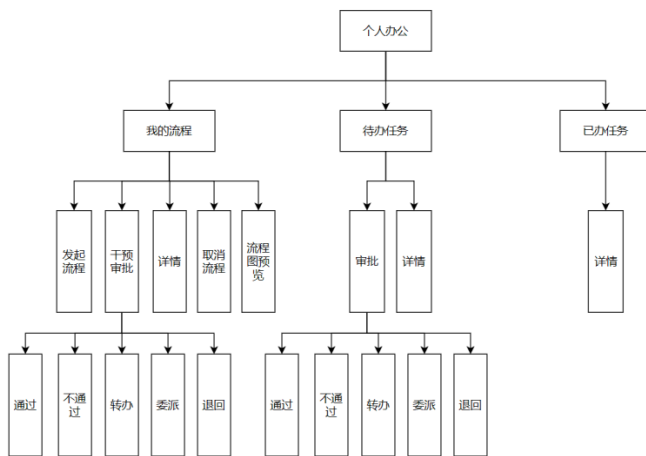


图 11 个人办公的功能图

主要代码实现如下：

```

@GetMapping("/generate-highlight-diagram")
@ApiOperation(value = "生成流程实例的高亮流程图")
@PreAuthorize("@ss.hasPermi('bpm:task:query')")
public void generateHighlightDiagram(
    @RequestParam("processInstanceId") String
    processInstanceId, HttpServletResponse response)
    throws IOException {
    byte[] bytes = activityService.
        generateHighlightDiagram(processInstanceId);
    ServletUtils.writeAttachment(response, StrUtil.format(
        "流程图-{}.svg", processInstanceId), bytes);
}
    
```

(6) 代码管理

代码管理只有一个代码生成功能，该功能主要负责代码的生成，方便程序员进行负责的页面设计，简化了设计过程。用户可以导入数据库表，根据数据库表内容进行基本代码的生成，可以进行压缩包下载或者自定义路径的导入，用户需要填写对应的生成信息，如生成模板、生成包路径以及上级菜单等，用户也可以进行代码的预览，具体功能见图 12 所示。

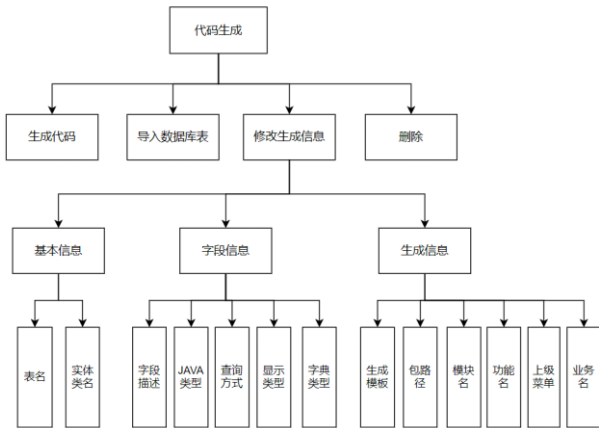


图 12 代码生成的功能图

主要代码实现如下：

```
@PreAuthorize("@ss.hasPermi('data:gent:code')")
@GetMapping("/download/{tableName}")
public void download(HttpServletRequest response,
@PathVariable("tableName") String tName)
throws IOException {
    byte[] dCode = genTableService.downloadCode(tName);
    genCode(response, dCode);
}
```

4 系统的实现

4.1 首页的实现

用户登录该平台首次看到的页面就是首页，为了给用户带来方便，将首页作为信息展示和跳转的直观页面，即展示给用户最重要的信息。该页面有工作台、信息通知、系统信息、我的任务、数据统计、系统参数和统计信息等。其中我的任务只展现四条信息，点击其中一条信息会跳转到办公页面，方便用户进行操作。系统信息分别展示当前该系统登录的人数、访问记录、操作次数和流程数等。首页运行页面如图 5-1 所示。

4.2 表单管理的实现

表单设计包含基本的增删改查、提交数据、详情和添加菜单操作等，其中增加和修改操作会进入表单设计台，它主要负责表单的设计操作，用户首先需要

进行表单属性的设计，如表单名和数据库名的编辑，其次可以进行表单控件的拖拽布局和参数设计，如字段名称、组件类型、栅格大小和标签宽度等，具体的页面展示见图 14 所示

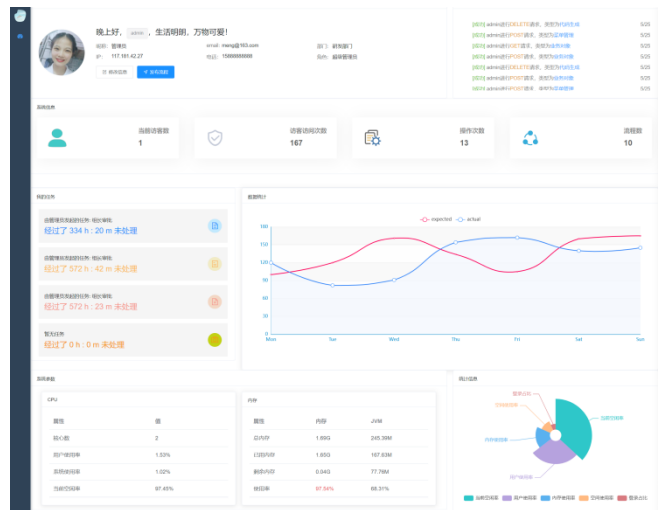


图 13 首页展示图



图 14 表单设计台的页面图

4.3 数据对象的实现

数据对象管理包含基本的 CRUD 操作，其中新增和修改操作会进入数据对象操作台，该操作台可以进行数据对象的设计，操作内容与表单设计类似，具体见图 15 所示。



图 15 数据对象操作台的页面图

4.3 流程设计的实现

流程模型包括基本的 CRUD 操作，还拥有导入、设计、分配规则和部署的功能，其中设计流程会跳转到流程模型设计台中，该流程设计台主要负责流程模型的设计，具体见图 16 所示。

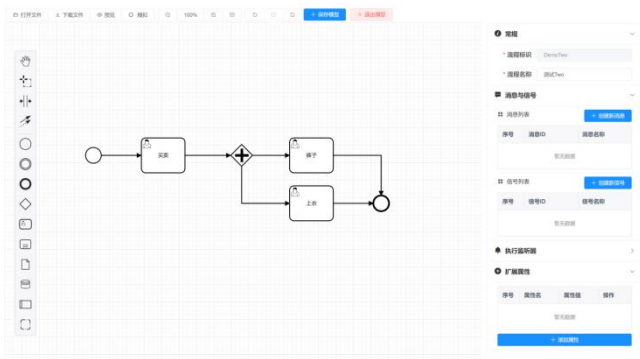


图 16 流程模型设计台的页面图

5 结束语

本文基于 Activiti 流程引擎进行办公平台的搭建，提供 Web 服务给予用户操作，采用低代码开发技术解决在线页面设计，使用数据流存储的形式进行页面参数保存，利用 Vue 的数据驱动视图优势进行页面回显。“低代码+办公”的功能覆盖了权限管理、数据库、表

单、数据对象、接口、流程模型和代码生成等操作，几乎满足基本页面编辑和数据存储功能。数据使用 mysql 进行存储，每 12 小时进行数据备份，有效防止数据丢失的问题，该平台还拥有一定程度的安全保障，有防恶意操作功能，对中小型企业搭建办公平台有一定的参考价值。

参考文献

- [1] 定义软件开发新模式 中国企业级无代码开发白皮书 2021 年[C]//艾瑞咨询系列研究报告(2021 年第 8 期). [出版者不详], 2021: 439-500.
- [2] 郭卫丹, 刘宗凡, 邱元阳, 倪俊杰, 杨磊. 深入评析低代码开发平台[J]. 中国信息技术教育, 2021(11): 80-86.
- [3] 朱运乔. 基于 Spring Security 认证与授权的 Web 应用与实现 [J]. 电脑编程技巧与维护, 2020(11): 14-16. DOI: 10.16184/j.cnki.comprg.2020.11.005.
- [4] 王崑. 浅谈 Java web 框架课程中 Maven 的教学内容和设计 [J]. 电脑知识与技术, 2021, 17(24): 85-86. DOI: 10.14004/j.cnki.ckt.2021.2292.
- [5] 车宇, 罗钊航. 办公自动化系统的开发引擎平台设计 [J]. 电子技术与软件工程, 2019(09): 167.