答辩稿

# 老师同学们大家好，

我的答辩题目是《基于Spring Boot框架的人事管理系统设计与实现》

我的论文是在张宏智、孔祥兵两位老师的悉心指点下完成的，在这里我向我的导师表示深深的谢意，

下面我将本毕业设计论主要内容向各位老师作一汇，报恳请各位老师批评指导；

# 首先简单描述一下本论文:

本文主要描述人事管理系统的设计及功能,人事管理在政府机关等事业单位和私有企业的日常行政中占有极其重要的地位。

各部门岗位之间协作和工效的提高,是人事管理的基本要求,

通过分析微型的人事管理系统适应当下需求.本项目就是在这样背景下提出的.

本项目旨在实现微型人事管理信息系统,

功能主要由人事在线通信,员工资料,人事信息管理,薪资调整分配,统计分析管理和系统设置管理系统构成.

实现员工信息编辑,人事信息入库展示分析,工资套帐处理及系统的设置,点对点在线讯息等功能.

本项目在技术方面采用了较前沿的技术,

应用端基于javaEE体系中开源应用框架Spring Boot(集成IOC,MVC,ORM),Spring Sercurity(安全框架),内嵌Tomcat服务器,

前端基于javaScript的渐进式框架Vue.js,页面设计基于ElementUI库, 路由处理基于 vue-router.

数据库端使用Mysql数据库,

系统架构基于当前主流的MVVM(Model-view-viewModel)模式前后端分离的SPA(single page web application),即单页富应用程序.

# 下面我以这五个部分对论文进行阐述：

# 系统概述：主要描述立项背景，开发架构、模式

目前,人事管理的系统大都是CS架构的大型系统,很少有面向机关,事业单位内部的基于BS微型人事系统,

因此.有了该人事信息管理系统.在人事信息系统中,功能需简单清晰,可操作性强,

其次安全和权限是关键点.对人事信息的保密涉及到安全处理,对不同的操作员角色根据保密级别给予不同的权限,

是当前信息系统的基本需求.项目基于MVVM的前后端分离开发模式进行开发

MVVM即模型(Model)-视图(View)-视图模型(View Model)，实现了数据视图的双向绑定.相对于MVC和MVP来说，MVVM是一个比较新的开发架构，它是一种将MVP模式与WPF相结合应用方式发展演变而成的新型开发架构.**mvp的全称为Model-View-Presenter，Model提供数据，View负责显示，Controller/Presenter负责逻辑的处理。MVP与MVC的区别：在MVP中View并不直接使用Model，它们之间的通信是通过Presenter (MVC中的Controller)来进行的，所有的交互都发生在Presenter内部，而在MVC中View会直接从Model中读取数据而不是通过 Controller。**

**WPF（Windows Presentation Foundation）是微软推出的基于Windows Vista的用户界面框架，属于.NET Framework 3.0的一部分。可以对软件UI层进行加细节化、可定制化。即指设计中，你需要一个为View量身定制的model，这个model就是ViewModel。ViewModel包含所有由UI特定的接口和属性，并由一个 ViewModel 的视图的绑定属性，并可获得二者之间的松散耦合，所以需要在ViewModel 直接更新视图中编写相应代码。**

前后端分离是指将前端和后端从之前的全部有后端负责中分离开来,不再共用一个Server,前端作为一个独立Serve存在.前后端通过接口使用HTTp协议交互,

本项目使用vue-router做前端路由处理.页面跳转不在由后端处理,前后端只是数据的交互.

前后端分离的好处在降低了前后端的耦合性，当面对不同的硬件场景时，需要构建不同的界面，前后端分离之后，只需要扩展前端项目即可,不需要修改后端服务。

# 系统分析：可行性分析、流程分析

下面我们看。技术可行性

本系统中技术使用,后端技术涉及到Spring Boot,Spring Security, MyBatis,MySQL等,前端技术设计到Vue, ElementUI, axios,webSocket等.系统架构基于B/S架构.

Spring Boot是当前流行的JavaEE体系框架,Spring Boot提供了一种新的编程范式,能在小的阻力下开发Spring应用程序,可以更加敏捷地开发Spring应用程序,专注于应用程序的功能,不用在Spring的配置上多花功夫,甚至完全不用配置[4].Spring Boot遵循约定大于配置的理念,在父子工程理念上极大程度上解决了依赖冲突的问题.

Spring Security是Spring家族中的安全框架,Spring Boot对于Spring Security提供了自动化配置方案,可以零配置使用 Spring Security.Spring Security借助过滤器技术对Web应用进行保护,包括认证和授权两个过程[5].

Mybatis,Mysql是持久化技术,都是开源轻量,Mybatis是ORM解决方案,Mysql是数据库系统.

Vue 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用，Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。[6]

axios是基于Promise 用于浏览器和 node.js 的 HTTP 客户端的异步编程解决方案,可以从浏览器中创建XMLHttpRequests,从node.js中创建http请求,支持Promise API,拦截请求和响应,转换请求数据和响应数据,客户端支持防御XSRF等.[6]

vue-router在项目中用于前端动态加载路由管理。SPA应用来讲,其实就是在前后端分离的基础上,加一层前端路由,即是由前端来维护路由规则,实现有两种，一种是利用url的hash,另一种就是HTML5的History模式.[7]这里我们使用vue-router的hash方式.

B/S架构是对C/S架构的一种改进,软件应用的业务逻辑完全由服务端实现[8],依托于浏览器,相对于C/S更轻量,用户体验要差一些,但是基于B/S的SPA解决了这个问题.

下面我们看。经济可行性

实现人事管理的自动化,HR在减少失误的同时精简许多费用开支,在硬件消耗上依托于互联网.同时开发使用技术工具基本为开源产品技术。而且，对于Web版的微型人事管理系统很少，未来有良好市场。下面我们看。运营可行性本系统在打包之后整体大小只有38MB,硬件内存需求相当于一个移动端的普通App大小,运行内存基本上普通的单核CPU,2G RAM的ESC就可以满足需求。系统硬件需求低.访问兼容大部分浏览器。用户体验处理上,由前端视图操作通知，结合后端逻辑处理消息构成.响应用户的每一步操作.构建了良好的用户体验.**访问流程分析**

访问流程的实现基于RBAC0模型.由之相对应的数据实体构成.由用户表,角色映射表,角色表,权限表,权限映射表.

本系统有管理员或操作员经由,自定义验证,表单验证通过之后,会进行请求路径分析,

Spring Security会进行鉴权处理,确认当前登录用户的角色,匹配放行响应的资源.

之后进行对员工信息进行增删改查,都是经过鉴权处理的,

管理员对系统的操作都将被记录在日志里.前端会对所有请求拦截进行统一处理.分析需要返回数据还是讯息.

业务比较简单..

# 系统设计实现 功能模块设计及具体实现

功能模块设计

系统安全访问实现

用户登录访问后由Spring Security的WebSecurityConfigurerAdapter类实现认证和授权校验.

登录成功后跳转主页,之后的一次访问都会通过校验用户,角色,基于Ant格式的url权限的匹配.权限处理的逻辑由后端处理,前端为了良好的用户体验.对没有权限的菜单项做了隐藏.

公共模块

主页

主页在UI设计上使用时间线和无限滚动框组件,日志显示当前系统用户的操作信息,系统公告显示管理员添加的公告信息.逻辑处理上默认加载最新的操作日志信息及公告信息,数据展示通过Spring定时任务方式，Service定时执行SQL，以排序和截取方式获取最新入库数据，动态更新数据.

在线聊天

在线聊天实现点对点的在线通讯.前端基于全双工通信WebSocket实现, WebSocket由HTTP建立连接,由TCP进行会话,Web服务器与客户端之间建立起WebSocket协议的连接,之后的协议都依靠专用协议进行,通信过程中可以互发送JSON,XML,HTML或图片等任意格式的数据,由于是建立在HTTP基础的协议上,即发起方仍是客户端,一旦确立WebSocket通信连接,不论服务器还是客户端,任意一方都可以直接向对方发送报文.[7] 在处理逻辑上，由键值对构建消息对象。

后端基于Spring 框架提供的 WebSocket 的 STOMP 支持, STOMP 是一个简单的可互操作的协议, 通常被用于通过中间服务器在客户端之间进行异步消息传递.

员工资料

基本资料

实现对员工基本资料的增删改查.实现员工基本资料表的Excel导入导出,多条件搜索.添加员工由数据库自动生成工号,其他信息为必填项.修改时,根据id进行数据更新.

高级资料

高级资料主要实习展示查询功能，由以下三部分信息构成:

(1).培训资料:培训日期,培训内容.

(2).考评资料:考评日期,考评内容,考评结果.

(3).工资信息:账套名称,基本工资.

人事管理

主要用于对员工奖惩,培训,考评,调动信息的管理.

员工奖惩.

对员工奖惩模块的增删改查的实现.在UI设计上使用穿梭框和tab标签,逻辑处理上员工实体和奖惩信息是一对多的的关系,即对同一员工在相同或者不同时间都可以添加多个奖惩,奖惩管理页面显示有奖惩信息的全部员工,可以查看奖惩详情.

员工培训

对员工培训模块的增删改查的实现.员工培训由添加培训和培训管理连个子模块构成，在UI设计设计上加了进度条.逻辑处理上是一对一的关系,即同一员工在一个时间段只能参加一个培训,培训完成删除当前培训后可以添加新的培训,支持批量添加培训,培训进度修改.

员工考评

由添加考评和考评管理两个子模块构成，实现对员工好评的增删改查.UI设计上结合评分条体现,支持批量添加评价.逻辑处理上是一对一的关系.

员工调动

对员工调动模块的增删改查的实现.UI设计上由table内嵌table构成,逻辑处理上是一对多的关系.

工资管理

实现对员工工资账套管理,员工账套设置,工资表管理,查询一系列对员工工资管理的功能.

工资账套管理

对员工奖金,基本工资,提成等各项的增刪改查. 设置当前套账信息.

员工账套设置

根据公司账套查看相应的员工的工资信息,并实现员工套账的其增删功能.

工资表管理

集中对员工工资,进行搜索查看的功能.添加筛选过滤功能，可以根据不同的职称部门进行表内筛选，工资套账详情查看。

统计管理

对员工资料的动态分析统计,包括员工积分统计,人事信息统计,人事记录统计.在设计方面结合图表处理VeCharts,由服务端对数据进行了整合,由VeCharts展示.代码方面,结合java8的Stream API及数据库端聚合函数实现.数据动态展示基于Spring定时任务.

员工积分统计

根据工号查找员工的积分情况.以条形图显示当前有积分员工的积分分布情况.积分由奖惩产生。

人事信息统计

对员工人数信息进行的分析统计,饼图的方式展示，以高校，职称，职位，党派，民族，学历，部门为维度的人员分布图,用于人事统计。

人事记录统计

对员工离职信息的统计,以部门，职称，职位为维度,对员工离职率,离职工龄,离职年龄进行分析，以柱状图结合折线图的方式统计.

系统管理

对系统,管理员等-些操作,包括基础信息设置,系统管理,操作员管理,授权处理.

基础信息设置

基础信息设置主要对部门,职位,职称,奖惩规则和权限组的设置增删查改功能管理,实现部门树,权限树预览.职称等级设置，奖惩描述自定义及使用模糊提示等。

部门管理，主要以多叉树的方式实现部门信息展示，及添加删除部门的操作，展示的实现由在Server端基于递归SQL的方式遍历部门树，删除操作当存在子部门，不能删除父部门，只能从叶子部门进行删除操作。添加不受限制。

员工职位管理即对企业职位的管理，由职位展示查询及自定义职位添加删除更改职位状态构成。

员工奖惩管理主要用于自定义员工奖惩，由添加奖惩类型，分值，描述及删除修改编辑构成。自定义奖惩描述可以使用以有描述，也可以自己定义。

员工职称管理于职位管理类似，由职称展示查询及自定义职称添加删除更改职称状态构成。不同在于职称有对应等级，添加时需要选择职称等级。

用户权限管理由权限树预览，角色添加，角色权限设置构成。权限树在数据处理上以二叉树的结构展示，因为只有两级菜单，对于不同角色通过权限树的设置，赋予不同菜单权限。权限树预览是基于VeCharts树形图表的展示。在UI设计上，以折叠面板和Tree树型组件方式实现。

系统管理

系统管理即对系统本身的一些设置.主要由公告管理和公告编辑构成，公告编辑主要结合VueQuillEditor富文本第三方扩展实现,有标题和具体公告内容构成.公告管理包括公告修改,公告预览,公告删除，可以动态修改所有公告信息并实现预览。

操作日志管理

操作日志管理主要展示登录用户对系统的操作信息.记录具体人事操作，便于以后的信息检索，功能上实现由日志类型,日期间隔的的日志查询.

操作员管理

操作员管理在UI设计上使用卡片组件,弹性布局处理.功能包括基本信息展示，刪改功能。具体由管理员角色权限的设置,是否启用的修改,删除管理员操作构成，在权限修改上以相互赋予的方式实现，即登录用户不能自己给自己赋值。只能通过其他用户赋予权限

# 总结回顾

通过对基于Spring Boot框架的人事系统的开发,我对系统开发有一个更清楚的了解,系统的开发过程中,不但熟悉了人事系统业务需求,同时拓宽了我对spring Boot,Linux的认知.整理开发遇到的问题，书写相关博文。也对前后端分离的SPA有了新的认识,尤其是前后端分离的权限控制,前端更多的是为了良好的用户体验,正真处理权限的是后端的处理.对于RBAC权限管理模型有了新的认识.相比于其他的大型的e-HR系统，比如钉钉之类商业软件，本系统显得相形见绌，但是我们面向的需求不同，相对于钉钉而言，功能众多，操作略显复杂，面向企业办公，而本系统功能简单轻量，属于BS架构Web版,面向基本的人事需求，即主要的使用人员是企业单位的HR。在人事业务方面，本系统涉及业务逻辑还是较浅，关于人事关联的考勤，工资，日常审批的业务，本系统都没有涉及到。在系统架构方面，本系统使用使用基本单模块，单服务，功能之间耦合性强。在此基础上，可以随着业务的深入进行模块服务解耦，实现多服务，接口交互架构。在多服务的基础上还可以进行云原生微服务架构。在使用场景方面，除了PC端，还可以开发面向微信的流应用，即微信小程序。所以，本系统的有很大的迭代升级空间。

# 参考文献

**我的参考文献主要分四部分构成，学位论文，电子文献，专著，以及期刊文章，**

**其中学位论文，期刊文章主要 通过上海科创数据中心获取，集中于万方数据库。**

**其中电子文献 ： 通过github开源库及技术官网获得。**

**专著部分主要来源 实习的及毕设开发 技术学习的相关书籍**

# 以上就是我的，毕业设计论文汇报，恳请各位老师批评指导。