

成绩：

# 江西科技师范大学

## 课程设计（论文）

题目（中文）：\_\_\_\_\_ 基于 SSM 的教务管理系统 \_\_\_\_\_

（外文）：The Enterprise Personnel Management System Based on SSM

院（系）：\_\_\_\_\_ 元宇宙产业学院 \_\_\_\_\_

专 业：\_\_\_\_\_ 计算机科学与技术 \_\_\_\_\_

学生姓名：\_\_\_\_\_ 蔡欣彤 \_\_\_\_\_

学 号：\_\_\_\_\_ 20193016 \_\_\_\_\_

指导教师：\_\_\_\_\_ 朱书彪，付淇 \_\_\_\_\_

2024 年 6 月 18 日

# 目录

一、引言 .....	1
1.1 研究背景 .....	1
1.2 课题目的和意义 .....	2
1.3 研究内容 .....	2
二、需求分析 .....	2
2.1 系统可行性分析 .....	2
2.2 系统功能分析 .....	3
2.3 系统模块需求分析 .....	3
2.2.1 用户信息管理 .....	4
2.2.2 部门信息管理 .....	4
2.2.3 职位信息管理 .....	5
2.2.4 员工信息管理 .....	5
2.2.5 公告信息管理 .....	6
2.2.6 文档信息管理 .....	6
2.4 数据流图 .....	7
三、概要设计 .....	9
3.1 E-R 图设计 .....	9
3.2 实体关系设计 .....	11
3.3 数据库设计 .....	12
3.3.1 管理员表设计 .....	12
3.3.2 员工表设计 .....	13
3.3.3 部门表设计 .....	14
3.3.4 职位表设计 .....	14
3.3.5 公告表设计 .....	15
3.3.6 文档表设计 .....	15
四、详细设计 .....	16
4.1 系统流程图 .....	16
4.1.1 简易的登录流程图 .....	16
4.1.2 普通员工登录流程图 .....	16
4.1.3 系统管理员登录流程图 .....	17
五、系统实现 .....	18
5.1 系统框架 .....	18
5.2 系统登录功能的实现 .....	18
5.3 系统管理功能的实现 .....	21
5.3.1 管理功能的实现代码 .....	21
5.3.2 管理功能模块的页面图 .....	22
六、系统测试 .....	27
6.1 测试目的和意义 .....	27
6.2 实例测试 .....	27
6.2.1 登录系统测试 .....	27
6.2.2 新增管理员测试 .....	28
6.2.3 查找管理员测试 .....	28

6.2.4 新增员工测试 .....	29
6.2.5 新增职位测试 .....	29
6.2.6 新增员工测试 .....	30
6.2.7 新增文档测试 .....	30
6.3 测试问题的处理方式 .....	30
6.3.1 系统中打日志 .....	31
6.3.2 使用编辑器提供的 Debug .....	31
6.3.3 查找相关博客 .....	31
七、总结和展望 .....	31
7.1 结论 .....	31
7.2 心得体会和展望 .....	31
参 考 文 献 .....	32

# 企业人事管理系统

**摘要:** 随着计算机技术的飞速发展,计算机在各种单位机构管理中应用的普及,管理信息系统的开发在强调管理、强调信息的现代社会中也显得越来越重要。因此,利用计算机高效率地完成人事管理的日常事务,是适应现代各种单位机构制度要求、推动各种单位机构人事管理走向科学化、规化的必要条件。本系统的数据库服务器使用 MySQL,后台采用 SpringMVC、Spring 和 MyBatis 流行框架进行整合开发,采用 Java 语言对系统进行设计和实现。本系统是对事业单位中的用户、部门、职位、员工、公告等信息进行管理的界面友好、操作简便与可靠性高的数据处理软件,有利于提高人事信息管理的效率、安全性与可靠性,使人事管理人员的工作变得轻松简单。

**关键词:** javaee; ssm 框架; 数据库设计; 系统实现; 系统测试

## 一、引言

### 1.1 研究背景

人力伴随着社会经济的不断进步和高速发展逐渐成为重要的社会前进动力的宝贵来源,由此针对人力资源管理的方式已经从传统的原始手动记录逐渐演变成劳力和人力资源方面的竞争<sup>[7]</sup>。社会对人力资源的需求不断加大,从而带动对人力管理能力持续增强,逐渐挖掘更深层次的管理方式,这无形中反应出社会竞争日渐强烈。同样,企业在此方面也持续性提升管理水平,高效科学的人力管理系统已经成为社会经济高速发展的重要影响因素<sup>[3]</sup>。

就企业本身而言,由于公司规模持续性扩张,公司员工持续性增加,人力管理成为难题,突显出管理系统的重要地位,企业急切需要针对人事管理的成套管理系统,需要不断提高对人力资源的管理水平,优化人员管理模式,对于提升公司人事管理工作效率具有不可替代的作用<sup>[3]</sup>。针对上述背景,将课题的焦点集中在企业的人事管理体系设计和实现方面,希望能辅助此方面的管理机制,不断增强人事管理能力,为管理效率的提升奠定基础。

企业人事管理信息系统是一个现代化的企业在竞争中立于不败之地的必要部分,它能够为查询人事信息提供充足的信息和快捷的查询手段<sup>[1]</sup>。使用计算机对企业人事信息进行管理,具有手工管理所无法比拟的优点。例如:检索迅速、查找方便、可靠性高、存储量大、寿命长、成本低等。这些优点能够提高企业人事管理的效率,也是企业的科学化、正规化管理与世界接轨的重要条件<sup>[2]</sup>。

## 1.2 课题目的和意义

通过本文的深入探究，希望能针对性地提升企业在人力管理方面的科学含量。不断增强其信息化水平，摒弃原始的人工管理模式，从而杜绝因为认为操作导致的各种信息沟通不畅的不足，最大程度降低人工劳动量，进而增强企业在人事管理方面的能力。不断更新企业人事资源管理机制是优化管理模式的必经之路，与此同时可以增强工作的便捷性，提升实现战略目标的效率。

人事管理体系应该具备安全、经济且实用的特性，同时要能易于拓展功能，方便后续的管理，能保证数据的严密性。在保证系统开放的前提上，不断满足企业的人事管理需求,同时兼备相对方管理模块性能,基于现阶段的功能模块研发，且能实现与其他管理流程或者部门职能等相连接的功能接口,完成公司信息资源共享的要求，同时不断加快信息融合的进程<sup>[4]</sup>。

## 1.3 研究内容

针对传统手工管理人事信息存在的问题，设计并实现了基于 SSM 的人事管理系统。本系统的数据库服务器使用 MySQL，后台采用 SpringMVC、Spring 和 MyBatis 流行框架进行整合开发，采用 Java 语言对系统进行设计和实现<sup>[8]</sup>。本系统是对事业单位中的用户、部门、职位、员工、公告等信息进行管理的界面友好、操作简便与可靠性高的数据处理软件，有利于提高人事信息管理的效率、安全性与可靠性，使人事管理人员的工作变得轻松简单。

# 二、需求分析

## 2.1 系统可行性分析

人事管理系统的主要任务是对人事档案进行整理，可以对用户、部门、职位、员工、公告、文档等信息进行增删改查。通过该系统，使企业的人事管理工作系统化、规范化、自动化，从而提高企业人事管理的效率。企业的人事管理应该以提升职工工作效率，实现管理的信息化，以及管理的无纸化，不断推动企业效益增长，降低生产成本为目的。基于对企业信息的透彻把握，才能将人事资源体系

设计的更加合理,只有在设计系统前认真调研企业实际需求，才能满足公司发展需求，才能使得系统功能解决企业现存问题<sup>[5]</sup>。

## 2.2 系统功能分析

根据系统设计要求，系统对管理员信息、员工信息、文档信息、公告信息、职位信息以及部门信息分别实现以下功能：

（1）数据录入:管理员信息、员工信息、文档信息、公告信息、职位信息以及部门信息的新建；

（2）数据修改:当管理员信息、员工信息、文档信息、公告信息、职位信息、部门信息出错时，可以实现管理员信息、员工信息、文档信息、公告信息、职位信息以及部门信息的修改；

（3）数据删除:当出现管理员离职、员工离职、文档过期、公告过期、职位删除、部门合并等情况时，需要对管理员信息、员工信息、文档信息、公告信息、职位信息以及部门信息进行删除操作；

（4）数据查询：实现管理员员工登录以及管理员信息、员工信息、文档信息、公告信息、职位信息以及部门信息的列表查询。条件查询等查询操作。

具体的企业人事管理系统功能模块图 2.1 所示。

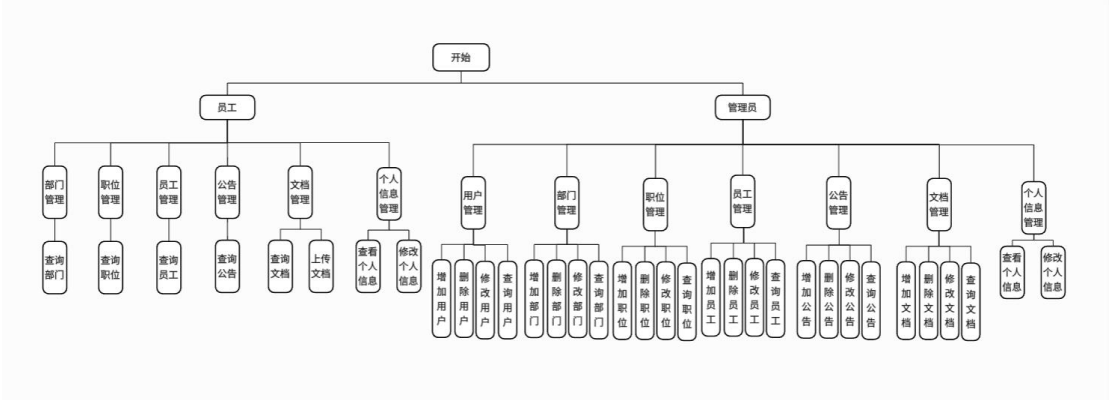


图 2.1 系统功能模块图

## 2.3 系统模块需求分析

人事管理系统按照功能分为：管理员模块、部门模块、职位模块、员工模

块、公告模块、文档模块六大模块。各模块之间又通过相互调用及资源共享使整个系统形成一个统一的有机整体，如图 2.8 所示。

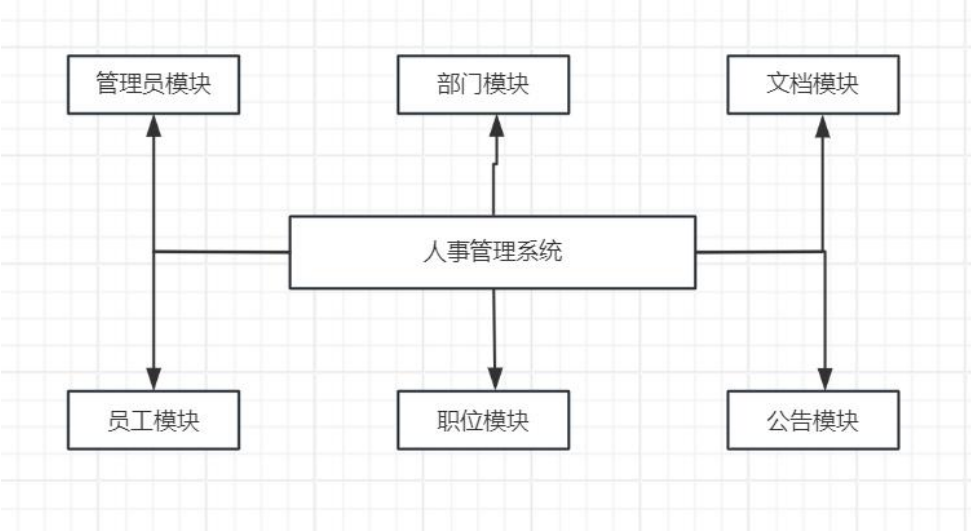


图 2.2 系统模块图

2.2.1 用户信息管理

用户登录后可对用户进行增删改查，其功能如图 2.3 所示。

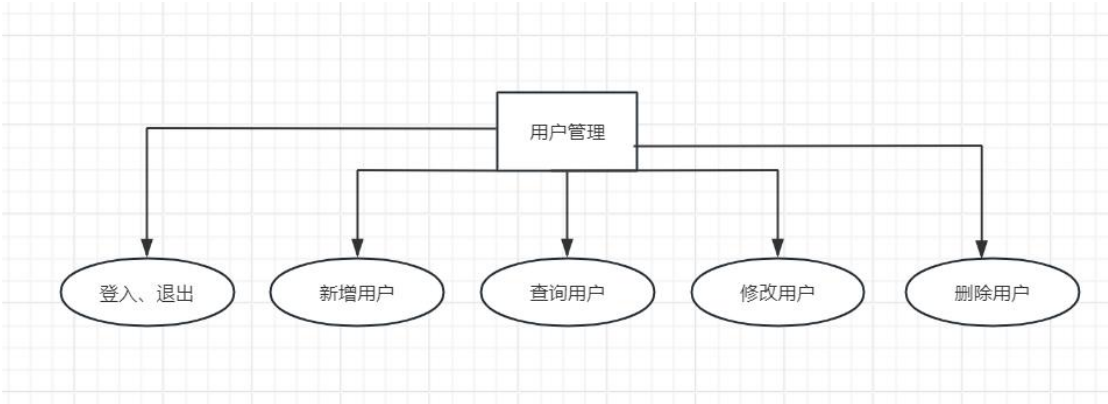


图 2.3 用户管理需求图

2.2.2 部门信息管理

部门信息管理主要涵盖了部门机构信息的修改、相关信息单的删除以及不同部门的信息单新增等工作，这些工作均由相关的管理专员负责，该部分是人事管理体系的重要构成之一，能实现部门职能信息和部门信息相关的增减修整操作，具体需求如图 2.4 所示。

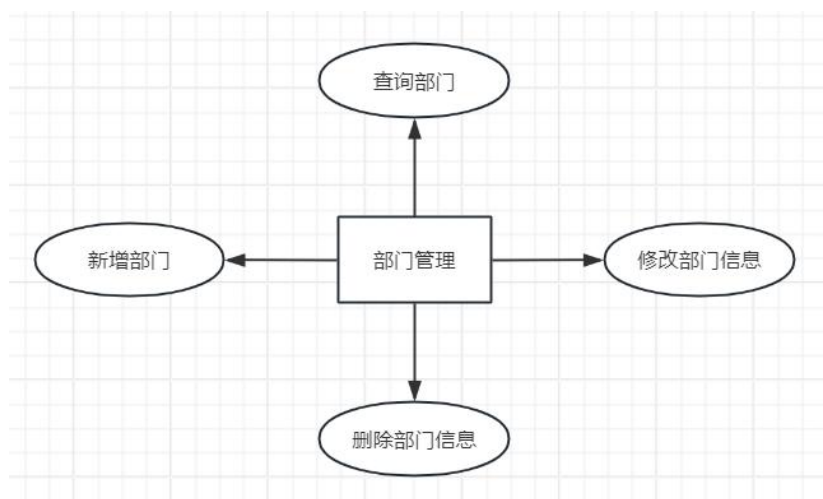


图 2.4 部门管理需求图

### 2.2.3 职位信息管理

职位信息管理需对员工所属部门和所担任的职务都要有所记录，需要具有职位信息的修改、相关信息单的删除以及不同职位的信息新增等功能。具体需求如图 2.5 所示。

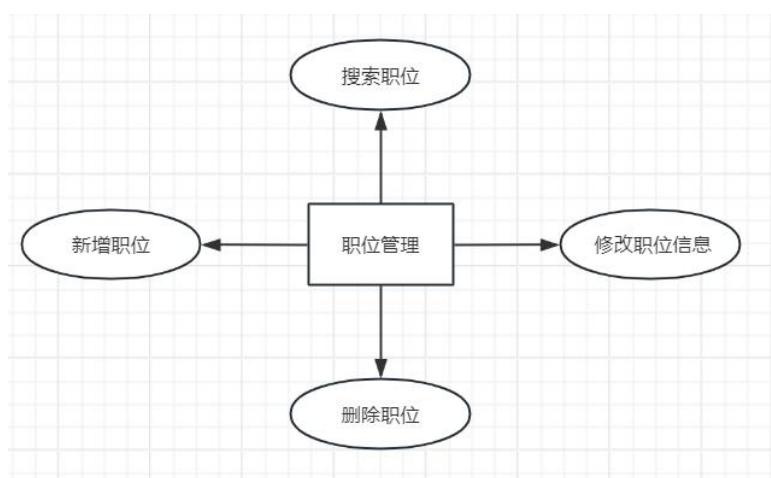


图 2.5 职位管理需求图

### 2.2.4 员工信息管理

员工信息管理为系统设计的基础部分，初步完成人事信息管理方面的信息化，通过对基本信息的掌握进一步延伸实现对绩效考核等方面的信息共享管理模式，进而完成无缝传输数据功能，完成信息管理的统一化发展。具体需求如图 2.6 所示。



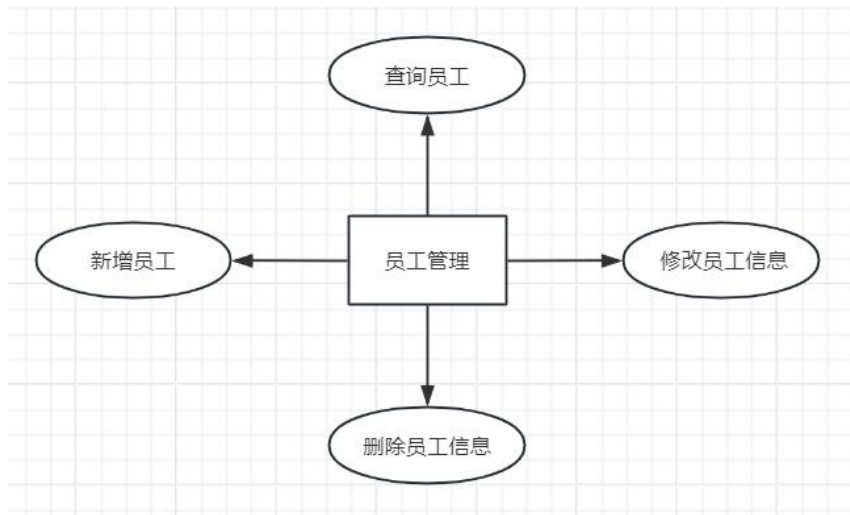


图 2.6 员工管理需求图

### 2.2.5 公告信息管理

公告信息管理主要功能即发布、修改、删除、以及查询功能，其功能需求如图 2.7 所示。

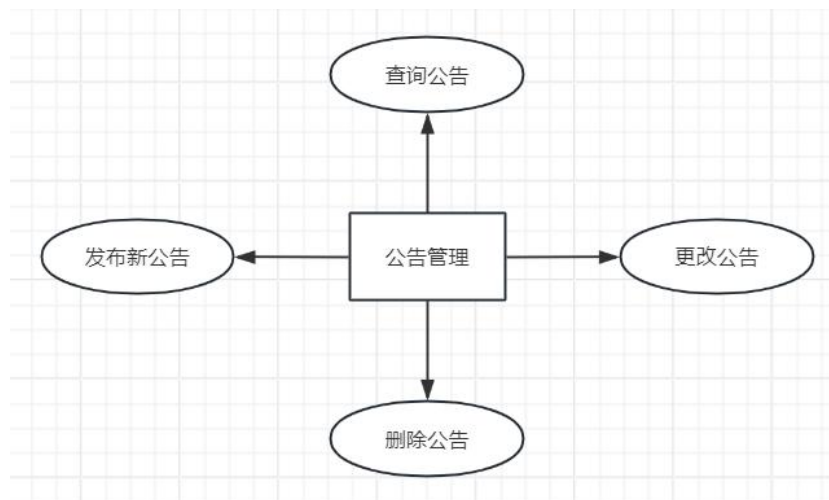


图 2.7 公告管理需求图

### 2.2.6 文档信息管理

文档信息管理主要功能即发布、修改、删除、以及查询功能，其功能需求如图 2.8 所示。

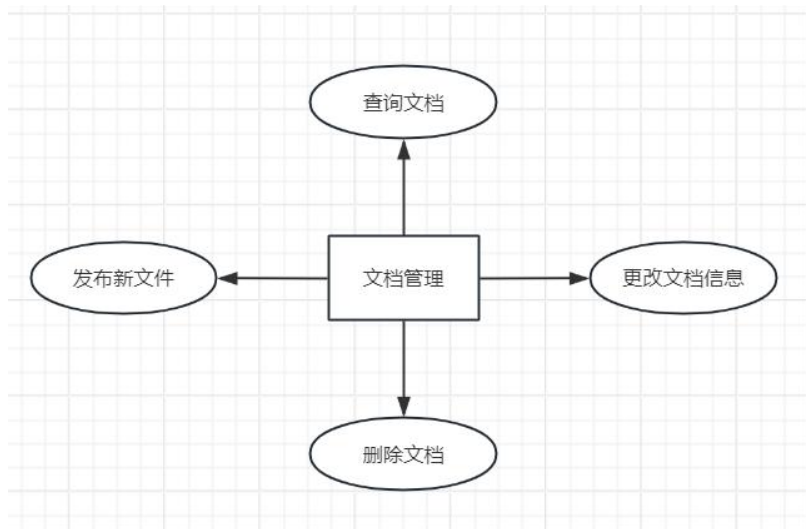


图 2.8 文档管理需求图

## 2.4 数据流图

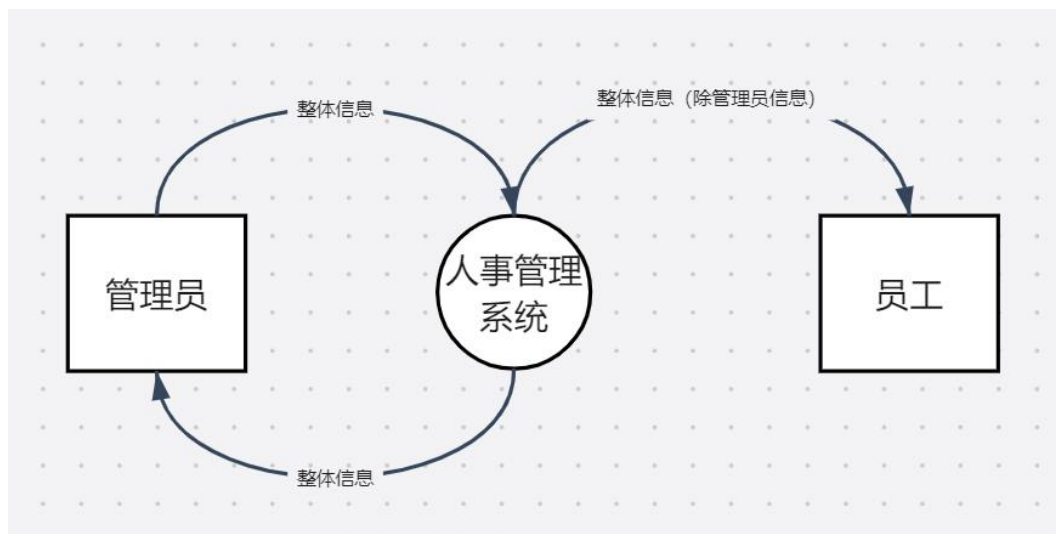


图 2.9 顶层数据流图

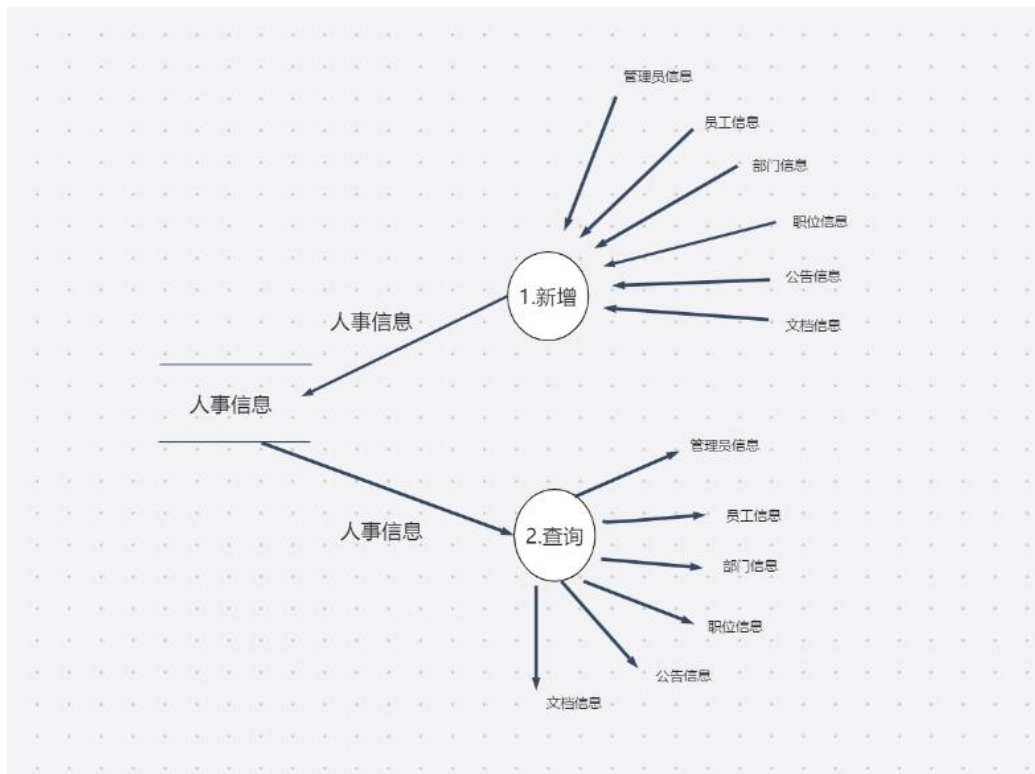


图 2.10 层数据流图

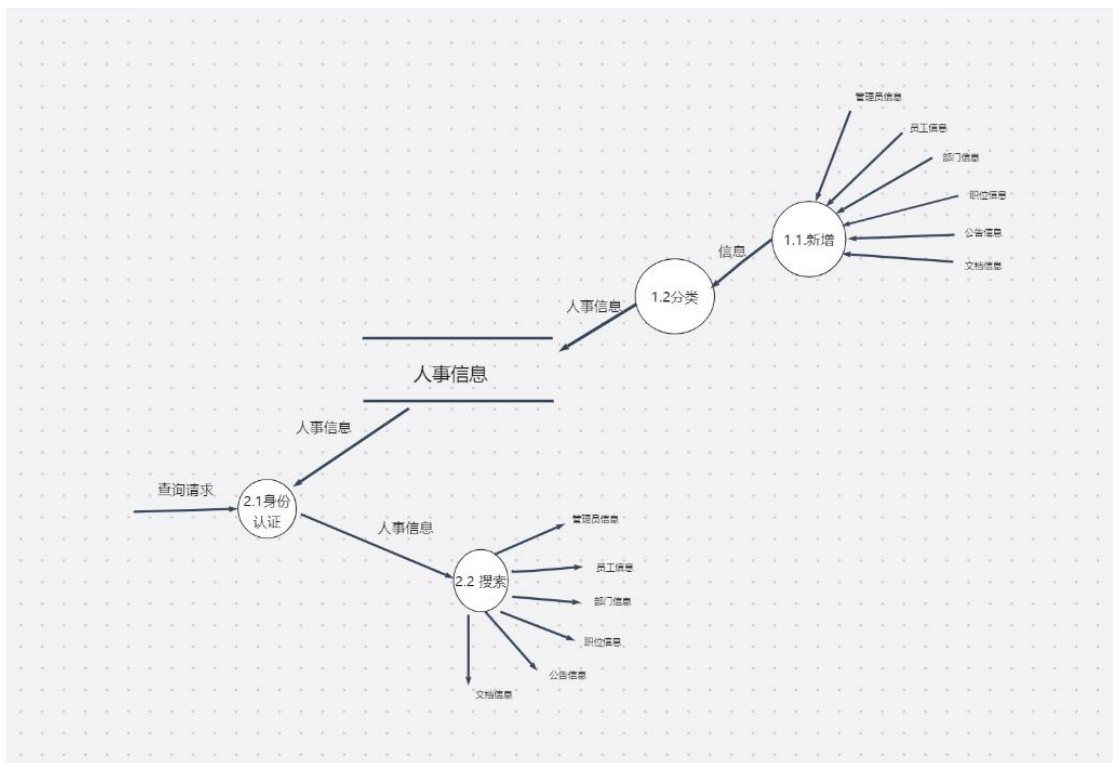


图 2.11 层数据流图

### 三、 概要设计

#### 3.1 E-R 图设计

在设计数据库表之前，我们需要根据功能需求设计出数据库实体，以及实体对象直接的关系即（E-R 图）该系统的所有 E-R 图如下：

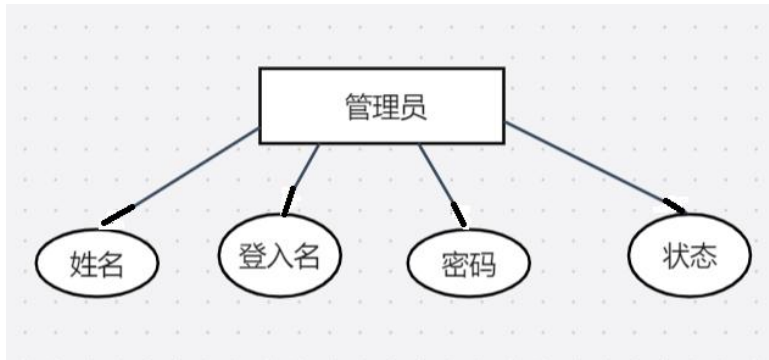


图 3.1 管理员 E-R 图

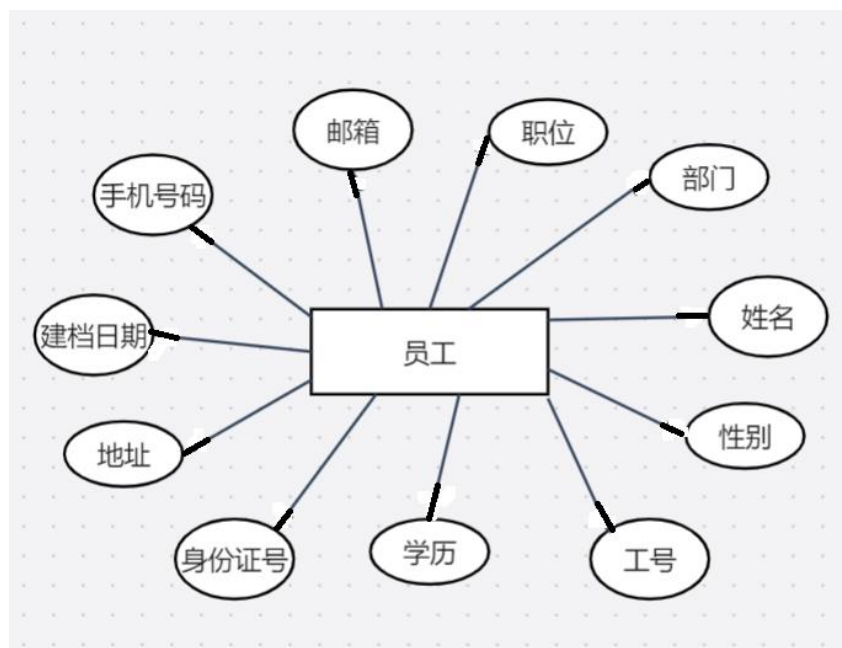


图 3.2 员工 E-R 图

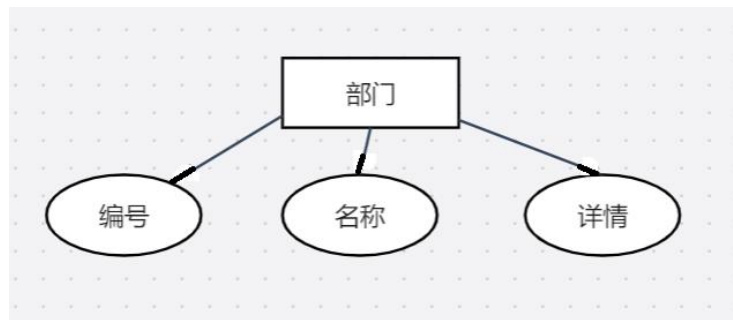


图 3.3 部门 E-R 图

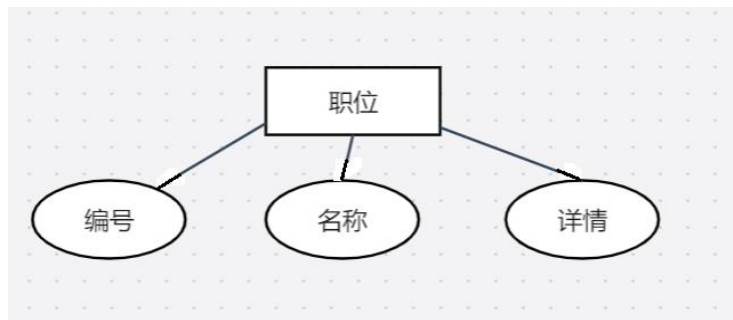


图 3.4 职位 E-R 图

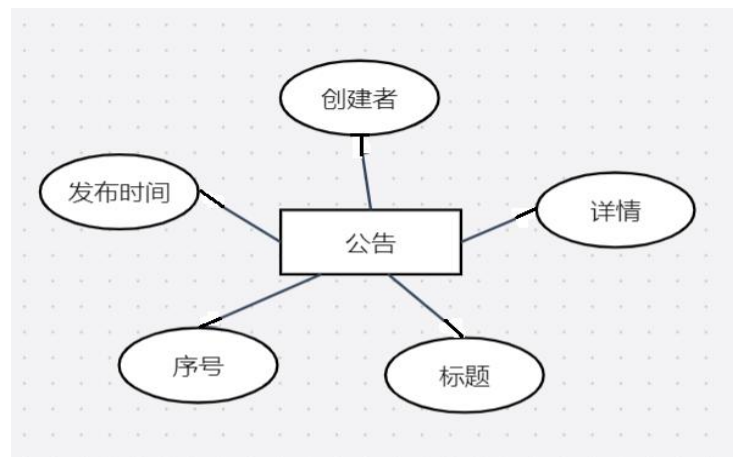


图 3.5 公告 E-R 图

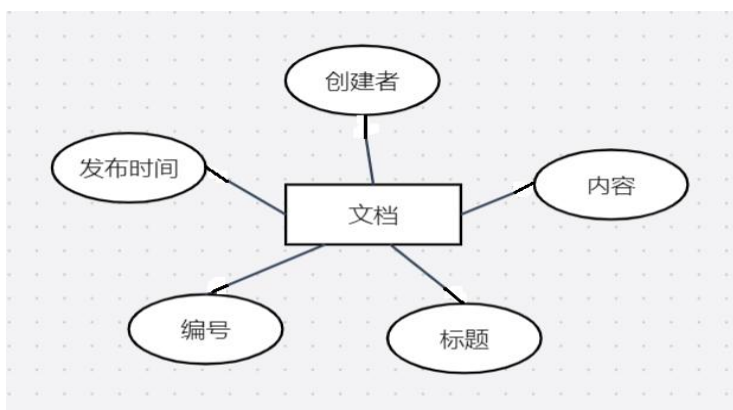


图 3.6 文档 E-R 图

### 3.2 实体关系设计

系统中的实体对应的属性会存在对应关系，从而将实体关联起来。本系统中的实体之间的对应关系如下图所示：

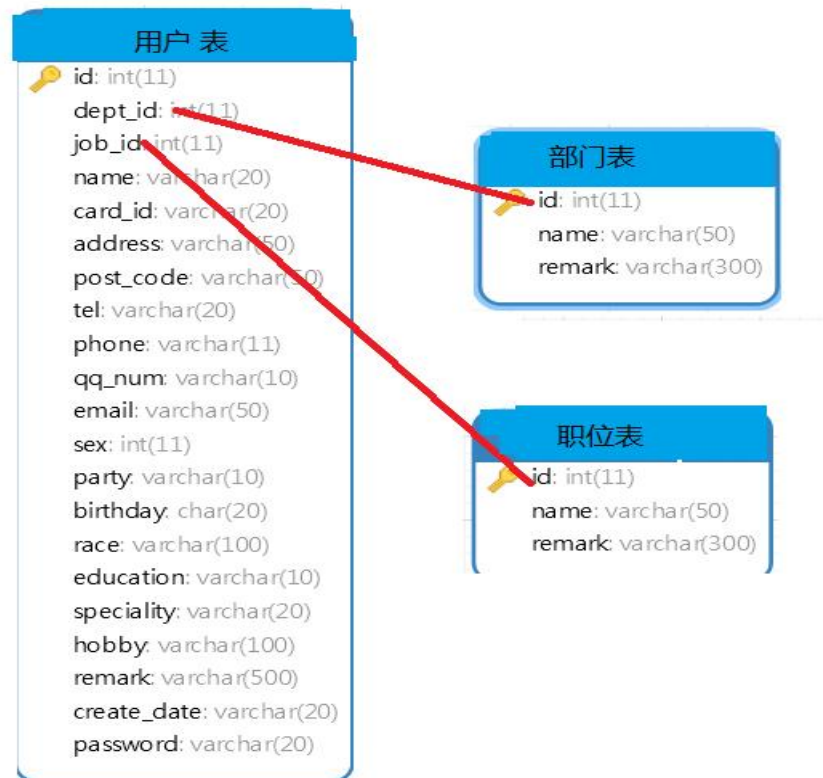


图 3.7 用户、部门、职位关系图

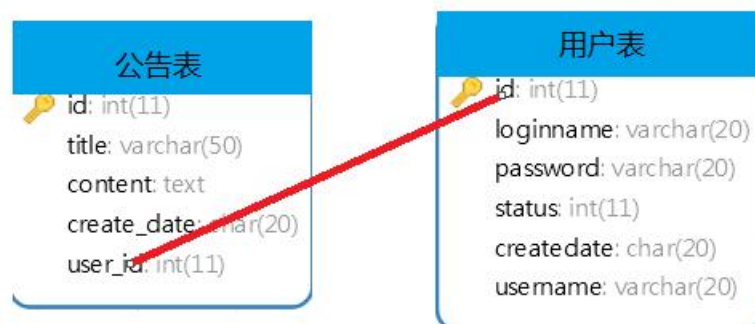


图 3.8 用户、公告关系图



图 3.9 用户、文档关系图

### 3.3 数据库设计

设计完实体对象以后，我们可以很快的设计出数据库表，数据库表一共六张，如下图所示：

#### 3.3.1 管理员表设计

表 3-3-1 管理员表 (user\_inf)

字段名称	类型（长度）	允许空	主键	外键
id	int（11）	否	是	否
loginname	varchar（20）	是	否	否
password	varchar（20）	是	否	否
status	int（11）	是	否	否
createdate	char（20）	是	否	否
username	varchar（20）	是	否	否

类型	长度	小数点	不是 null	虚拟	键	注释
int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1	
▶ varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
int	11		<input type="checkbox"/>	<input type="checkbox"/>		
char	20		<input type="checkbox"/>	<input type="checkbox"/>		
varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		

图 3.10 管理员表

### 3.3.2 员工表设计

表 3-3-2 员工表 (employee\_inf)

字段名称	类型 (长度)	允许空	主键	外键
id	int (11)	否	是	否
dept_id	int (11)	是	否	否
job_id	int (11)	是	否	否
name	varchar (20)	是	否	否
card_id	varchar (20)	是	否	否
address	varchar (50)	是	否	否
post_code	varchar (50)	是	否	否
tel	varchar (20)	是	否	否
phone	varchar (11)	是	否	否
qq_num	varchar (10)	是	否	否
email	varchar (50)	是	否	否
sex	int (11)	是	否	否
party	varchar (10)	是	否	否
birthday	char (20)	是	否	否
race	varchar (100)	是	否	否
education	varchar (10)	是	否	否
speciality	varchar (20)	是	否	否
hobby	varchar (100)	是	否	否
remark	varchar (500)	是	否	否
create_date	varchar (20)	是	否	否
password	varchar (20)	是	否	否



名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
dept_id	int	11		<input type="checkbox"/>	<input type="checkbox"/>		
job_id	int	11		<input type="checkbox"/>	<input type="checkbox"/>		
name	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
card_id	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
address	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		
post_code	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		
tel	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
phone	varchar	11		<input type="checkbox"/>	<input type="checkbox"/>		
qq_num	varchar	10		<input type="checkbox"/>	<input type="checkbox"/>		
email	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		
sex	int	11		<input type="checkbox"/>	<input type="checkbox"/>		
party	varchar	10		<input type="checkbox"/>	<input type="checkbox"/>		
birthday	char	20		<input type="checkbox"/>	<input type="checkbox"/>		
race	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>		
education	varchar	10		<input type="checkbox"/>	<input type="checkbox"/>		
speciality	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
hobby	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>		
remark	varchar	500		<input type="checkbox"/>	<input type="checkbox"/>		
create_date	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
password	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		

图 3.11 员工表

### 3.3.3 部门表设计

表 3-3-3 部门表 (dept\_inf)

字段名称	类型（长度）	允许空	主键	外键
id	int（11）	否	是	否
name	varchar（50）	是	否	否
remark	varchar（300）	是	否	否

字段	索引	外键	触发器	选项	注释	SQL 预览
名						
id						
name						
remark						

图 3.12 部门表

### 3.3.4 职位表设计

表 3-3-4 职位表 (job\_inf)

字段名称	类型（长度）	允许空	主键	外键
id	int（11）	否	是	否

name	varchar (50)	是	否	否
remark	varchar (300)	是	否	否

字段	索引	外键	触发器	选项	注释	SQL 预览						
名						类型	长度	小数点	不是 null	虚拟	键	注释
▶ id						int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	
name						varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		
remark						varchar	300		<input type="checkbox"/>	<input type="checkbox"/>		

图 3.13 职位表

### 3.3.5 公告表设计

表 3-3-5 公告表 (notice\_inf)

字段名称	类型 (长度)	允许空	主键	外键
id	int (11)	否	是	否
title	varchar (50)	是	否	否
content	text	是	否	否
create_date	char (20)	是	否	否
user_id	int (11)	是	否	否

名	类型	长度	小数点	不是 null	虚拟	键	注释
▶ id	int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	
title	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		
content	text			<input type="checkbox"/>	<input type="checkbox"/>		
create_date	char	20		<input type="checkbox"/>	<input type="checkbox"/>		
user_id	int	11		<input type="checkbox"/>	<input type="checkbox"/>		

图 3.14 公告表

### 3.3.6 文档表设计

表 3-3-6 文档表 (document\_inf)

字段名称	类型 (长度)	允许空	主键	外键
id	int (11)	否	是	否
title	varchar (50)	是	否	否
filename	varchar (300)	是	否	否
remark	varchar (300)	是	否	否
create_date	varchar (20)	是	否	否
user_id	int (11)	是	否	否



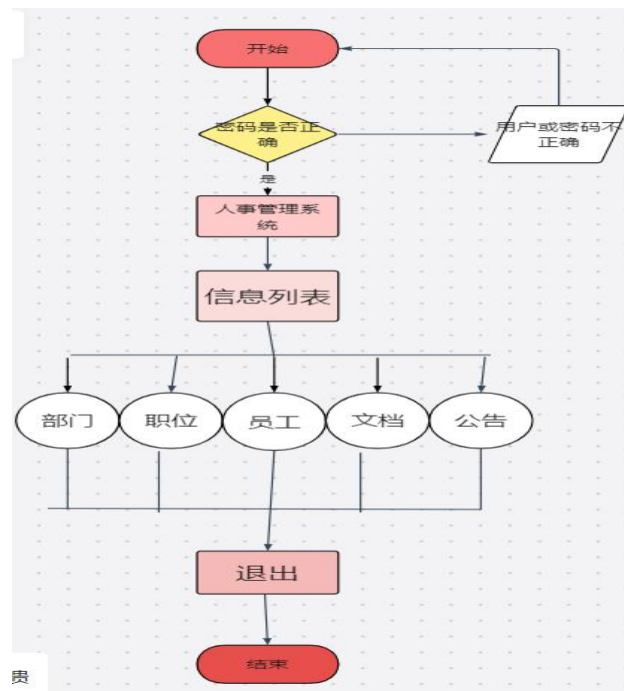


图 4.2 用户登录流程

### 4.1.3 系统管理员登录流程图

系统管理员登录后进入到管理系统，可对管理员信息、部门信息、职位信息、员工信息、文档信息、公告信息进行增删查改，具体流程如图 2.10 所示。

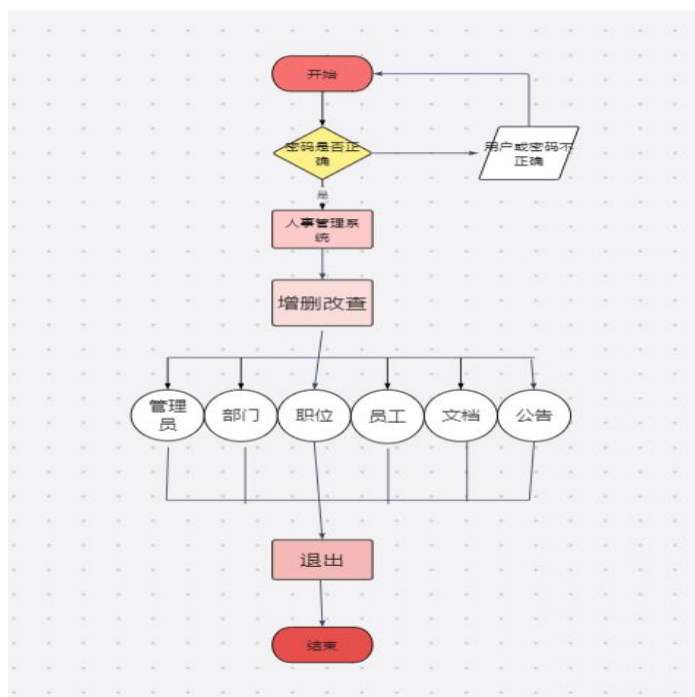


图 4.3 管理员登录流程

## 五、系统实现

### 5.1 系统框架

SSM 框架是一种轻量级系统开发框架，主要由 Spring，SpringMVC，和 MyBatis 三个组件框架组合而成，在 SSM 框架中，Spring MVC 主要负责整体系统的 view，module，和 control 三个层次分离。而 MyBatis 是由阿里开源的数据持久化框架。Spring 则是一个不错的容器，和创建空间之间的依赖。SSM 相比与传统的 Web 框架 SSH(Structs、Spring、Hibernate)，更为灵活和简易<sup>[6]</sup>。

系统具有三大层面，其中表示层为是 web 浏览器,能实现客户端的展示功能，由于其寄托于网络的持续发展进程，受到不同使用者和研发人员的青睐<sup>[10]</sup>。该浏览器功能不仅可以实现使用者与系统的交互操作，还能对用户信息进行处理操作，方便用户信息的传输。举例如下，当使用者要完成信息发送操作时，可借助该信息窗口的不同按钮和文本框进行操作。一旦事件在界面端提交就会触发功能层的不同处理机制，完成信息的传输<sup>[9]</sup>。

其次为业务逻辑层，也可视为功能层，其功能是完成数据传输操作，具有一定的桥梁作用，该层可以针对数据进行管理操作，逻辑层收到应用层的任务请求会进行相应数据访问与事件响应，并作用在数据库，由此对数据进行把控与监管，通过该层的功能，终可以处理不同的数据信息，从而完成数据交互等操作。

后一层是数据层，亦可以称作持久层，该层具有访问数据的功能，为数据资源的管理功能模块，当中间层向数据库发送一定指令时，该数据层可以接受上述指令，从而对数据库的内容进行删除、更新、浏览、修正等操作，后将操作结构提送到功能层部分<sup>[5]</sup>。

### 5.2 系统登录功能的实现

系统登录模块，为确保系统安全性，系统操作员只有在登录界面输入正确的管理员名、密码和角色，单击“登录”按钮后才能够进入本系统的主界面。登录界面如图 5.1 所示：

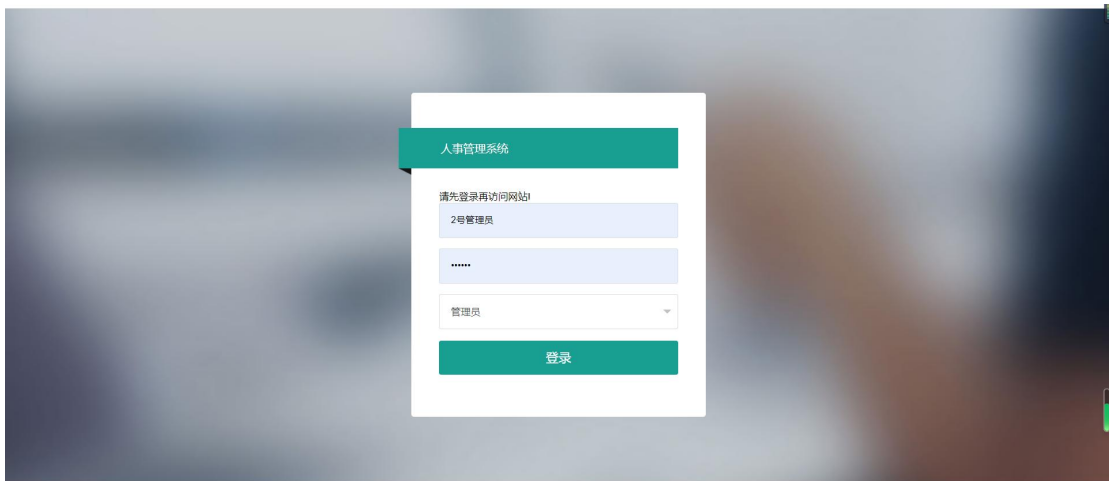


图 5.1 登录界面

(1) 实现登录页面的代码如图 5.2 所示：

```
<div class="login lay-anim lay-anim-up">
  <div class="message">人事管理系统</div>
  <div id="darkeners"></div>
  ${requestScope.message}
  <form method="post" class="lay-form" action="${ctx}/login">
    <input name="loginname" placeholder="用户名" value="${loginname}" type="text" lay-
    <hr class="hr15">
    <input name="password" lay-verify="required" placeholder="密码" type="password" cla
    <hr class="hr15">

    <select class="lay-input" name="tip" >
      <option value="1">管理员</option>
      <option value="2">员工</option>
    </select>
    <br>
    <input value="登录" lay-submit lay-filter="login" style="..." type="submit">
    <hr class="hr20" >
  </form>
</div>
```

图 5.2 登录界面代码

(2) 实现管理员登录功能的主要代码设计如图 5.3 所示：

```

public ModelAndView login(@RequestParam("loginname") String loginname,
    @RequestParam("password") String password, @RequestParam("tip") String tip,
    HttpSession session,
    ModelAndView mv){
    // 调用业务逻辑组件判断用户是否可以登录
    boolean flag = false;
    if("1".equals(tip)) {
        User user = rainservice.login(loginname, password);
        if(user!=null){
            // 将用户保存到HttpSession当中
            System.out.println("HttpSession");
            session.setAttribute(Constants.USER_SESSION, user);
            session.setAttribute(s: "tip", o: "1");
            // 客户端跳转到main页面
            mv.setViewName("redirect:/index");
        }else{
            // 设置登录失败提示信息
            System.out.println("设置登录失败提示信息");
            mv.addObject( attributeName: "message", attributeValue: "登录名或密码错误!请重新输入");
            // 服务器内部跳转到登录页面
            mv.setViewName("forward:/loginForm");
        }
    }
}

```

图 5.3 管理员登录功能代码

(3) 实现普通员工登录功能的主要代码设计如图 5.3 所示:

```

}else {
    Employee user = rainservice.login2(loginname, password);
    if(user!=null){
        // 将用户保存到HttpSession当中
        System.out.println("HttpSession");
        session.setAttribute(Constants.USER_SESSION, user);
        session.setAttribute(s: "tip", o: "2");
        // 客户端跳转到main页面
        mv.setViewName("redirect:/indexcustomer/");
    }else{
        // 设置登录失败提示信息
        System.out.println("设置登录失败提示信息");
        mv.addObject( attributeName: "message", attributeValue: "登录名或密码错误!请重新输入");
        // 服务器内部跳转到登录页面
        mv.setViewName("forward:/loginForm");
    }
}
return mv;
}

```

图 5.4 员工登录功能代码

## 5.3 系统管理功能的实现

### 5.3.1 管理功能的实现代码

(1) 用户管理功能:

这里使用添加管理员用户功能的代码做范例,实现此功能的主要代码设计如图 5.5 所示:

```
@RequestMapping(value="/user/add",method=RequestMethod.GET)
public String add(Model model,Integer id){
    if(id!=null){
        User job = rainservice.getUserInfo(id);
        model.addAttribute("job",job);
    }
    return "/user/add";
}
```

图 5.5 添加用户功能代码

(2) 职位管理功能:

这里使用删除职位功能的代码做范例,实现删除职位功能的主要代码设计如图 5.6 所示:

```
@RequestMapping(value="/job/delete",method=RequestMethod.GET)
public void delete(Integer id){
    System.out.println(id);
    if(id!=null){
        rainservice.deleteJobInfo(id);
    }
}
```

图 5.6 删除职位功能代码

(3) 员工管理功能:

这里使用修改员工信息功能的代码做范例,实现此功能的主要代码设计如图 5.7 所示:



```

@RequestMapping(value="/employee/add",method=RequestMethod.POST)
public ModelAndView add(ModelAndView mv,@ModelAttribute Employee job ,Integer id){
    System.out.println(id);
    if(id!=null){
        rainservice.update_EmployeeInfo(job);
    }else{
        rainservice.insert_EmployeeInfo(job);
    }
    mv.setViewName("redirect:/employee/list");
    return mv;
}

```

图 5.7 添加员工功能代码

#### (4) 公告管理功能:

这里使用查询公告功能的代码做范例,实现查询公告功能的主要代码设计如图 5.8 所示:

```

// 如果在目录下输入任何不存在的参数,则跳转到list
@RequestMapping(value="/notice/{formName}")
public String index2(@PathVariable String formName){
    String blank = "/notice/list";
    return blank;
}

@RequestMapping(value="/notice/list",method=RequestMethod.GET)
public String index(Model model,String content){
    List<Notice> job_list = rainservice.get_NoticeList();
    if (content!=null){
        job_list = rainservice.get_NoticeLikeList(content);
    }
    model.addAttribute("list",job_list);
    return "notice/list";
}

```

图 5.8 查询公告功能代码

### 5.3.2 管理功能模块的页面图

(1) 管理员登录进入企业人事管理系统可以查看管理员用户信息、部门信息、职位信息、员工信息、公告信息、以及文档信息进行相应操作,如图 5.9 所示。

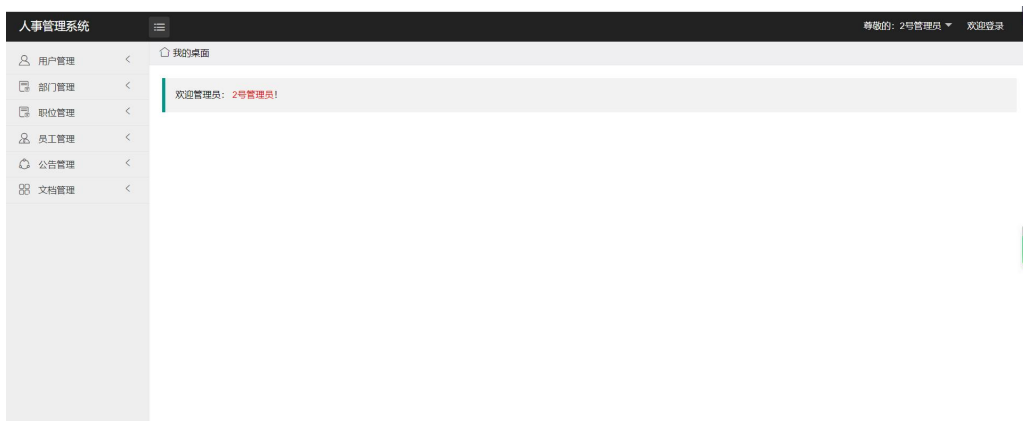


图 5.9 管理员功能页面图

(2) 用户管理，在用户管理页面中可以对部登录名，用户名，注册日期进行修改或删除用户，和添加管理员用户等详细操作，如图 5.10 所示。



图 5.10 用户管理员功能页面图

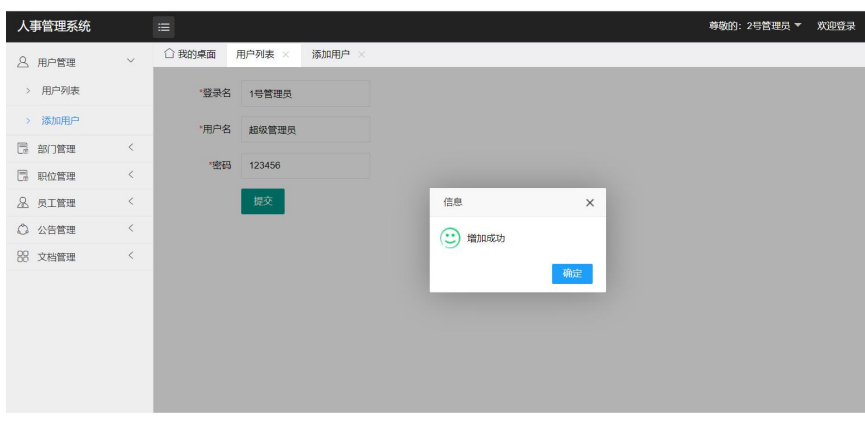


图 5.11 增加用户管理员成功页面图

(3) 部门管理，在部门管理页面中可以对部门名称，详细信息进行修改或删除部门，和添加部门等详细操作，如图 5.11 所示。

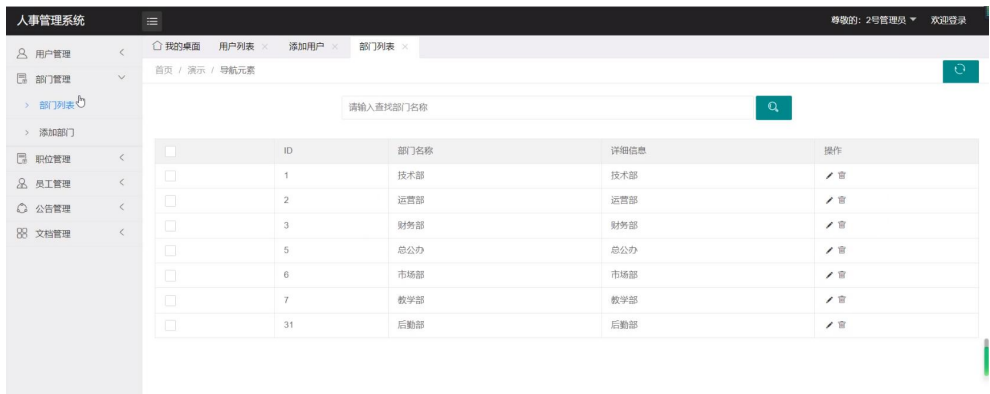


图 5.12 部门管理功能页面图

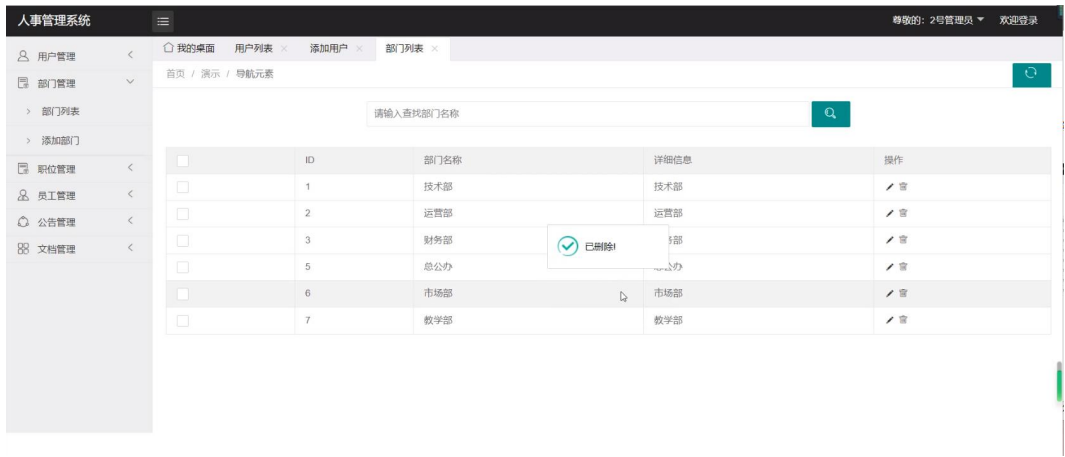


图 5.13 删除后勤部部门成功页面图

(4) 职位管理，在职位管理页面中可以对职位、职位信息进行修改或删除职位，和添加职位等详细操作，如图 5.14 所示。

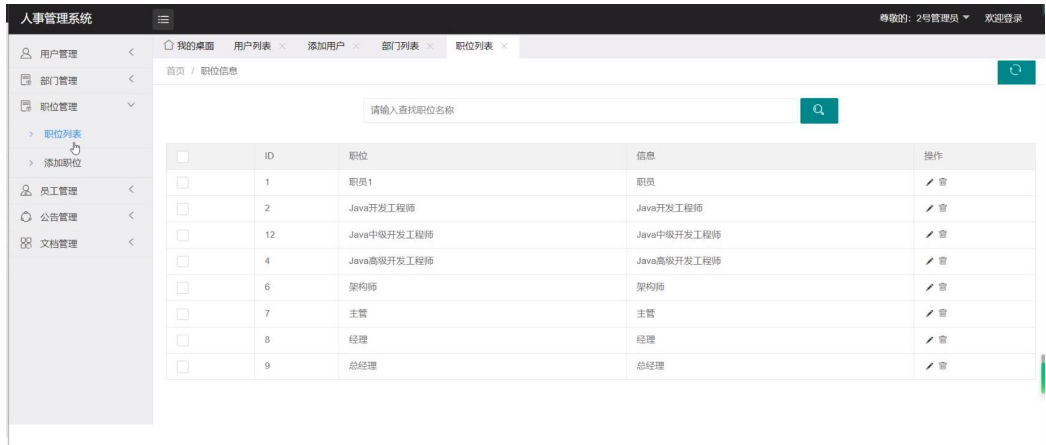


图 5.15 职位管理页面图

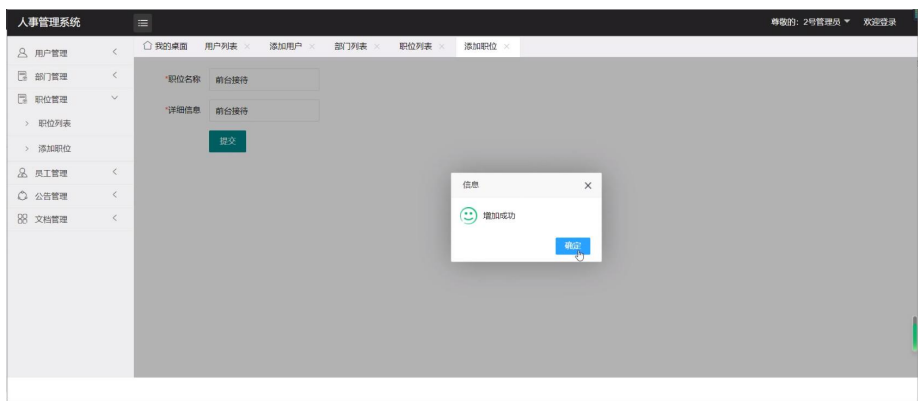


图 5.16 添加职位成功页面



图 5.17 添加职位成功页面图

(5) 员工管理，在员工管理页面中可以对姓名、性别、手机号码、邮箱、职位、学历、身份证号、部门、联系住址、建档日期进行修改或删除，和添加员工等详细操作，如图 5.18 所示。

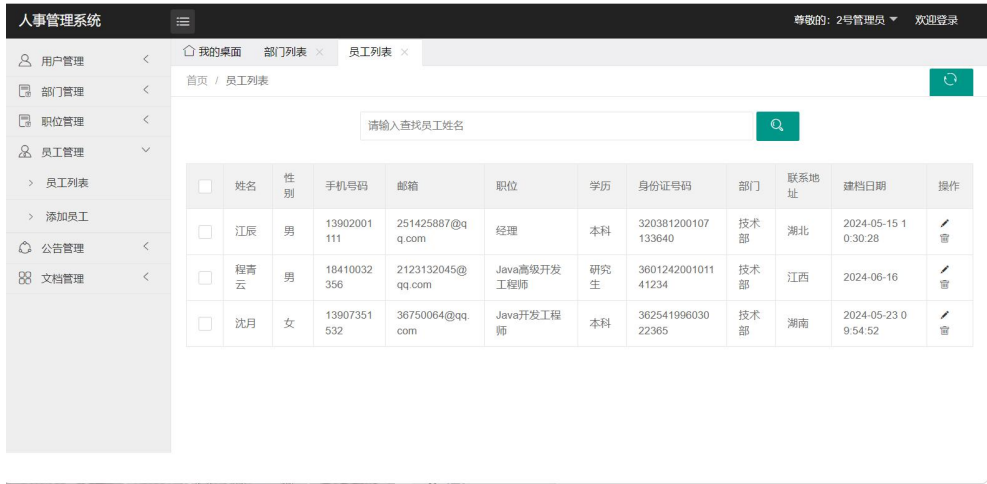


图 5.19 员工管理页面图

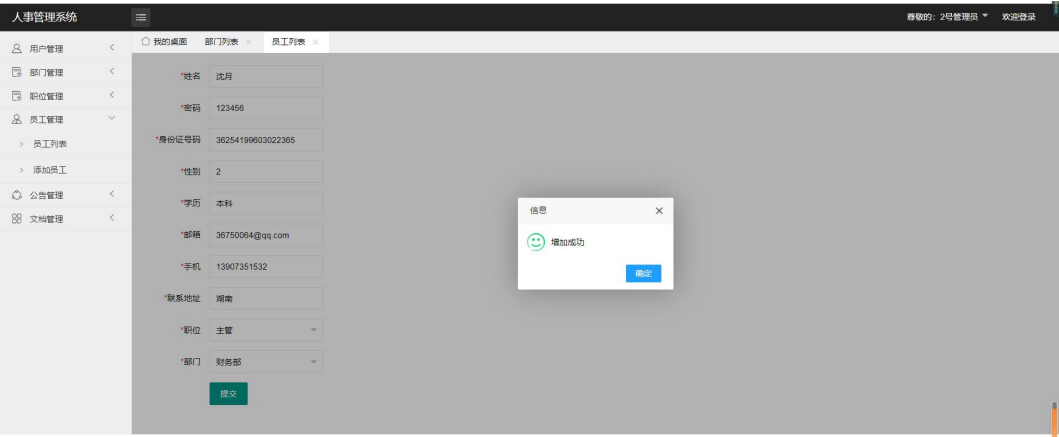


图 5.20 修改员工信息成功页面图

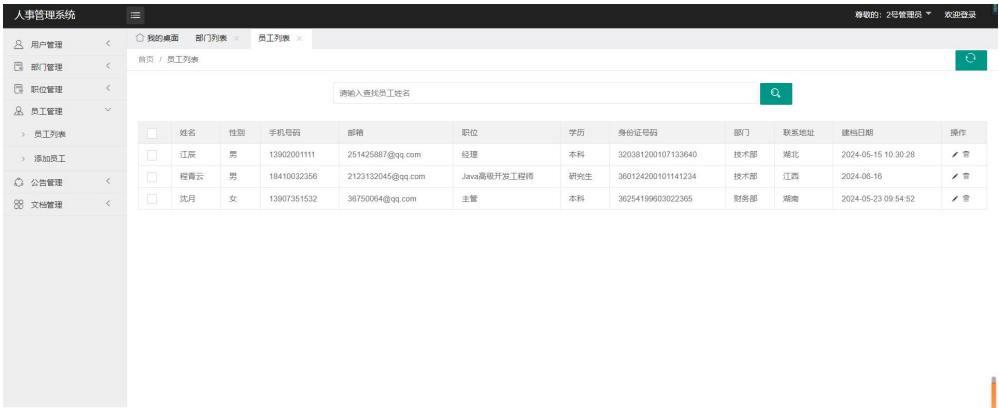


图 5.21 修改员工信息成功页面图

（6）公告管理，在公告理页面中可以对标题，内容，发布日期，发布用户进行修改或删除公告，和添加公告等详细操作，如图 5.22 所示。



图 5.22 公告管理页面图

（7）文档管理，在文档管理页面中可以对标题，描述，发布日期，发布用户进行修改或删除等详细操作，如图 5.23 所示。



图 5.23 文档管理页面图

## 六、系统测试

### 6.1 测试目的和意义

系统测试的目的是为了发现错误而执行程序的过程，成功的测试是发现了至今尚未发现的错误的测试。测试是希望能以最少的人力和时间发现潜在的各种错误和缺陷。根据开发各阶段的需求、设计等文档或程序的内部结构精心设计测试用例，并利用这些实例来运行程序，以便发现错误。

### 6.2 实例测试

本系统根据不同的功能设计了对应的测试用例。下面列举了系统中不同模块的添加以及搜索功能用例。

#### 6.2.1 登录系统测试

项目名称	人事管理系统		
系统模块	管理员/员工	测试编号	TestUserCase-001
测试功能	系统登入		
测试方法	黑盒测试法		
测试步骤	测试数据	预期结果	实际结果
输入正确的用户名和密码	用户名: admin 密码: 123456	登入成功, 进入系统	登入成功, 进入系统
输入用户名 不是如密码	用户名: admin 密码:	提示必选项不能为空	提示必选项不能为空
输入密码不输入用户名	用户名: 密码: 123456	提示必选项不能为空	提示必选项不能为空
输入的用户名或密码错误	用户名: admin 密码: 1	清空保单数据, 重新登入	清空保单数据, 重新登入

### 6.2.2 新增管理员测试

表 6-2-2 新增管理员用例表

项目名称	人事管理系统		
系统模块	管理员模块	测试编号	TestUserCase-002
测试功能	新增管理员		
测试方法	黑盒测试法		
测试步骤	测试数据	预期结果	实际结果
输入登入名、用户名、密码	登入名: admin 用户名: 管理员 密码: 123456	提示添加成功	提示添加成功
不输入登入名、密码、用户名中的一项或几项	登入名: admin 用户名: 密码: 123456	提示必选项不能为空	提示必选项不能为空

### 6.2.3 查找管理员测试

表 6-2-3 查找管理员用例表

项目名称	人事管理系统		
系统模块	管理员模块	测试编号	TestUserCase-003
测试功能	查找管理员		
测试方法	黑盒测试法		
测试步骤	测试数据	预期结果	实际结果
输入存在登入名	登入名: admin	得到管理员信息	得到管理员信息
输入不存在登入名	登入名: adm	得不到管理员信息	得不到管理员信息
不输入登入名		返回所有管理员信息	返回所有管理员信息

#### 6.2.4 新增员工测试

表 6-2-4 新增员工用例表

项目名称	人事管理系统		
系统模块	部门模块	测试编号	TestUserCase-005
测试功能	新增部门		
测试方法	黑盒测试法		
测试步骤	测试数据	预期结果	实际结果
输入部门和部门详情	部门名: 技术部 部门详情: 技术支持	提示添加成功	提示添加成功
只输入部门或部门详情	部门名: 技术部 部门详情:	提示必选项不能为空	提示必选项不能为空

#### 6.2.5 新增职位测试

表 6-2-5 新增职位用例表



项目名称	人事管理系统		
系统模块	职位模块	测试编号	TestUserCase-006
测试功能	新增职位		
测试方法	黑盒测试法		
测试步骤	测试数据	预期结果	实际结果
输入职位名称和职位详情	职位名称: Java开发 详情: 开发工程师	提示添加成功	提示添加成功
只输入职位名称或详情	职位名称: Java开发 详情:	提示必选项不能为空	提示必选项不能为空

## 6.2.6 新增员工测试

表 6-2-6 新增公告用例

项目名称	人事管理系统		
系统模块	公告模块	测试编号	TestUserCase-007
测试功能	新增公告		
测试方法	黑盒测试法		
测试步骤	测试数据	预期结果	实际结果
输入标题和内容	标题: 公告 内容: 这是一条公告	提示添加成功	提示添加成功
只输入标题或内容	标题: 公告 内容:	提示必选项不能为空	提示必选项不能为空

## 6.2.7 新增文档测试

表 6-2-7 新增文档用例表

项目名称	人事管理系统		
系统模块	文档模块	测试编号	TestUserCase-008
测试功能	新增文档		
测试方法	黑盒测试法		
测试步骤	测试数据	预期结果	实际结果
输入标题、描述	标题: 标题 描述: 这是一个文档	提示添加成功	提示添加成功
只输入标题、描述	标题: 标题 描述:	提示必选项不能为空	提示必选项不能为空

## 6.3 测试问题的处理方式

系统开发的过程中，出现 bug 是不可避免的，我们需要通过不断的测试将 bug 处理掉。在本系统的开发过程中，处理方式有以下几种：

### 6.3.1 系统中打日志

在系统中打日志可以快速查看用户输入的参数，解决参数格式不一致，空参导致系统无法运行等问题。

### 6.3.2 使用编辑器提供的 Debug

当系统运行没问题，但是逻辑出现问题的时候，debug 是一种和好的解决方式。通过对目标代码块打断点，一步一步调试，分析问题的原因即可解决。

### 6.3.3 查找相关博客

当系统抛出异常，导致系统无法运行，异常信息我们看不明白的时候，我们可以在网上搜索异常相关的关键词。通过相关的案例，对自己的系统排查从而达到解决问题的目的。

## 七、总结和展望

### 7.1 结论

本文基于对企业人力管理模式的探究，设计出适合企业人事管理的信息体系，基于三层架构的系统设计对每个功能的实现进行详细探究，能充分填补人工管理的不足之处，更方便后续系统的维护操作。整个设计过程兼顾现阶段企业的人力管理工作特征，不断满足业务功能需求，能很好扮演信息管理的重要角色，极大地提升了职工的工作效率。经过测试对其管理功能与流程进行分析，实际的系统运营过程，需要持续性完善系统的功能，并不断增强其性能，不断推动人力资源管理不断朝着信息化和智能化发展。

### 7.2 心得体会和展望

本次设计的系统，虽然不够完美，但是在做的过程中让我学到了很多有用的东西，例如 SSM 框架，mysql 数据库使用等，经过本次实践，相信在今后的开发中也会让我受益匪浅，并且以后的学习生活中，我更应该多加学习，掌握更多的专业知识，使自己的专业能力得到提升，开发效率也更加的快。

## 参 考 文 献

- [1] 黄梯云. 管理信息系统(第四版)[M]. 北京: 高等教育出版社, 2009.7
- [2] 黎连业. 管理信息系统设计与实施. 北京: 清华大学出版社, 1998, 10
- [3] 蒋波, 汪永超, 姚必强. 企业人力资源信息管理系统中的评价体系[J]. 机械, 2007, 33(11): 55-57
- [4] 容会. 基于 C/S 模式的昆明冶金高等专科学校党员信息管理系统的分析与设计.[D] 云南: 云南大学, 2010: 04-11.
- [5] 李燕荔, 姚国清. 企业人力资源管理系统[J]. 电脑编程技巧与维护. 2008, (6): 19-26.
- [6] 刘红霞. 企业人事管理系统的设计与实现[J]. 科学与财富, 2015, 03(6): 50-50, 51.
- [7] 杨文慧. 人力资源管理信息系统的设计与实现. 电脑知识与技术, 2013, 9(16), 140-141.
- [8] 方调元, 王超, 范松巍. 人力资源管理系统设计与实现. 计算机应用, 2010, 30(8), 2188-2192.
- [9] 陈学东, 张红, 殷晓勇. 基于 WEB 的人力资源管理系统的设计与实现. 计算机应用与软件, 2010, (7), 138-140.
- [10] 徐建军, 吴希兵. 基于 WEB 的人力资源管理系统的设计与实现. 计算机工程与应用, 2008, 44(9), 221-224.

## The Enterprise Personnel Management System Base on SSM

Abstract: With the rapid development of computer technology, the popularization of the application of computer in the management of various units and organizations, the development of management information system is becoming more and more important in the modern society that emphasizes management and information. Therefore, using the computer to efficiently complete the daily affairs of personnel management is a necessary condition to adapt to the requirements of modern institutions and institutions, and promote the personnel management of various institutions to be scientific and programmed. The database server of this system uses MySQL, the background uses SpringMVC, Spring and MyBatis popular frameworks for integrated development, and uses Java language to design and implement the system. This system is a user-friendly, easy to operate and reliable data processing software for the management of users, departments, positions, employees, announcements and other information in public institutions, which is conducive to improving the efficiency, security and reliability of personnel information management, making the work of personnel management personnel easy and simple.

Key words: javaee; ssm framework; Database design; System implementation; System test