

南昌航空大學

## 課程設計說明書

課程設計名稱： Web 應用開發課程設計

課程設計題目： 學生信息管理系统

學院名稱： 信息工程学院

專 業： 计算机科学与技术 班 级： 180452

學 号： 18045214 姓 名： 陈明瀚

評 分： \_\_\_\_\_ 教 师： 胡建明、时招军

20 20 年 1 月 6 日

# 1. 引言

## 1.1 意义

学生信息档案的管理对于学校的管理者来说至关重要，学生信息是高等学校非常重要的一项数据资源，是一个教育单位不可缺少一部分。特别是近几年来，国家政策的调整，我国高等院校大规模的扩招，给高等院校的教学管理、学生管理、后勤管理等方面都带来不少的冲击。其包含的数据量大，涉及的人员面广，而且需要及时更新，故较为复杂，难以单纯地依靠人工管理，而且传统的人工管理方式既不易于规范化，管理效率也不高，我国各类高等院校中还有相当一部分学生档案管理还停留在纸介质的基础上，尤其是中、小学对学生档案的管理更是落后，这样的管理机制已经不能适应时代发展的要求，其管理方法将浪费许多人力和物力。随着科学技术的不断提高，计算机科学与技术日渐成熟，计算机应用的普及已进入人类社会生活的各个领域，并发挥着越来越重要的作用。这种传统的手工管理模式必然被以计算机为物质基础的信息管理方法所取代。

作为计算机应用的一部分，使用计算机对学生档案进行管理，有着手工管理所无法比拟的优点，如：检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点能够极大地提高学生档案管理的效率，也是学校向科学化、正规化管理发展的必要条件，更是各个高等院校与世界接轨的重要条件。

系统针对学校学生信息的特点以及管理中实际需要而设计，能够有效地实现学生信息管理的信息化，减轻管理人员的工作负担，高效率、规范化地管理大量的学生信息，并避免人为操作的错误和不规范行为。

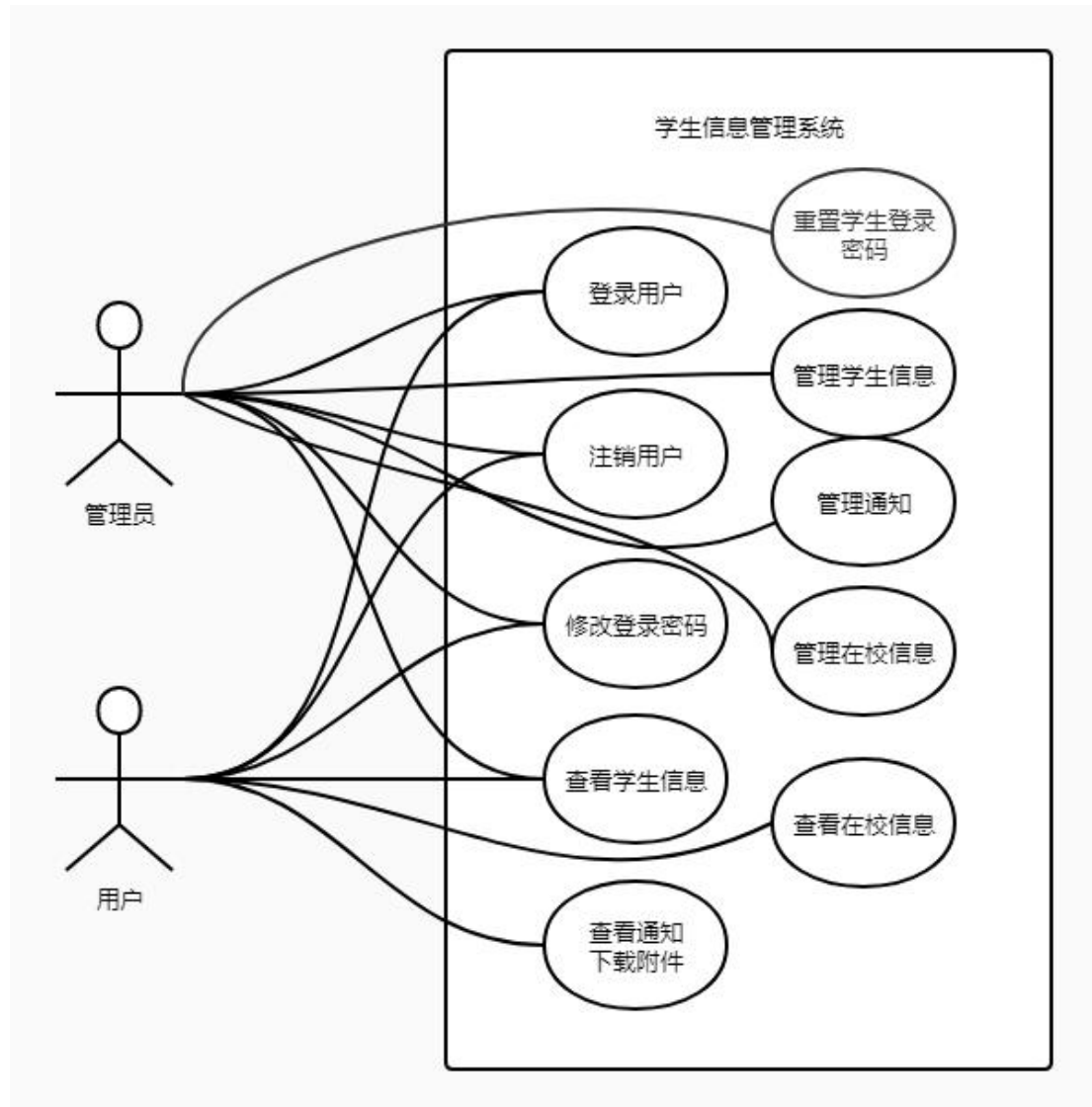
## 1.2 目标

由于现在的学校规模在逐渐的扩大，设置的专业类别、分支机构及老师、学生人数越来越多，对于过去的学生信息管理系统，不能满足当前学生信息管理的服务性能要求。本课程设计对于开发新的学生信息管理系统面临的问题及解决方案进行初步的设计与合理的安排，对用户需求进行了全面细致的分析，更清晰的理解学生信息管理系统业务需求，深入描述软件的功能和性能与界面，确定该软件设计的限制和定义软件的其他有效性需求，对开发计划进行了总体的规划确定开发的需求与面临困难的可行性分析。

我们希望通过本次课程设计可以建立一个学生信息管理系统以便方便对学生的学籍信息以及在校期间的记录信息进行查询、管理还有通知的发布和下载通知赴福建及其管理。学生信息管理系统主要任务是用计算机对学生各种信息进行日常管理，如查询、修改、增加、删除、统计，另外还考虑到学生查看和下载文件，针对这些要求设计了本次课程设计的任务——学生信息管理系统，利用本系统可以实现科学化、规范化和自动化管理学生信息。

## 2. 系统分析

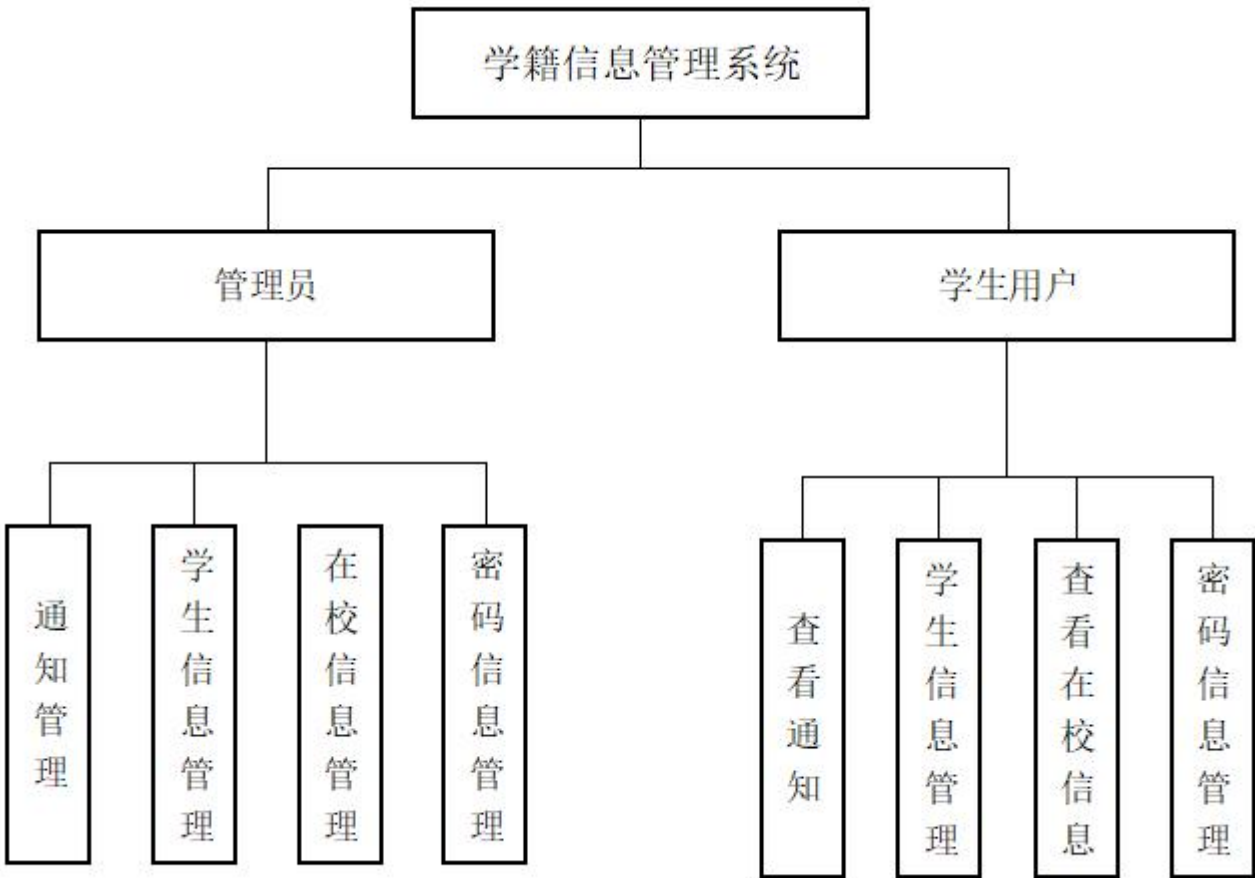
### 2.1 UML 用例图



### 3. 系统分析

#### 3.1 系统结构图

本次课程设计的学生信息管理系统结构中主要分管理员和学生用户两个角色，主要是围绕着通知和学生个人信息、学生在校信息以及密码信息进行的一系列的管理操作



#### 3.2 系统模块分析

本系统具有如下功能模块：

用户

1、登录模块：

选择学生登录入口，填写 帐号和 密码，并输入验证码进行登录。

2、查看通知模块：

登录后选择所要查询的通知内容的按钮直接进行下载文件进行查看更多内容。

3、查看个人信息模块：

当特殊权限允许时，登录后选择个人信息入口可对个人信息进行修改学生个人的部分信息。

4、查看在校信息模块：

登录后选择在校信息入口可对在校信息进行修改学生在校的部分信息。

5、登录密码管理模块

学生可以确认修改登录密码。

## 管理员

### 1、登录模块：

选择管理员登录入口，填写 帐号和 密码，并输入验证码进行登录。

### 2、通知管理模块：

登录后选择所要进行操作的入口进行增添、修改、删除，可以根据特定项模糊查询关键词。

同时可以选择添加通知附件提供用户下载。

### 3、学生个人信息管理模块：

登录后选择所要进行操作的入口进行增添、修改、删除，可以根据特定项模糊查询关键词。

在进行特定条件查询操作执行后可以对查询信息进行统计分析。

### 4、学生在校信息管理模块：

登录后选择所要进行操作的入口进行增添、修改、删除，可以根据特定项模糊查询关键词。

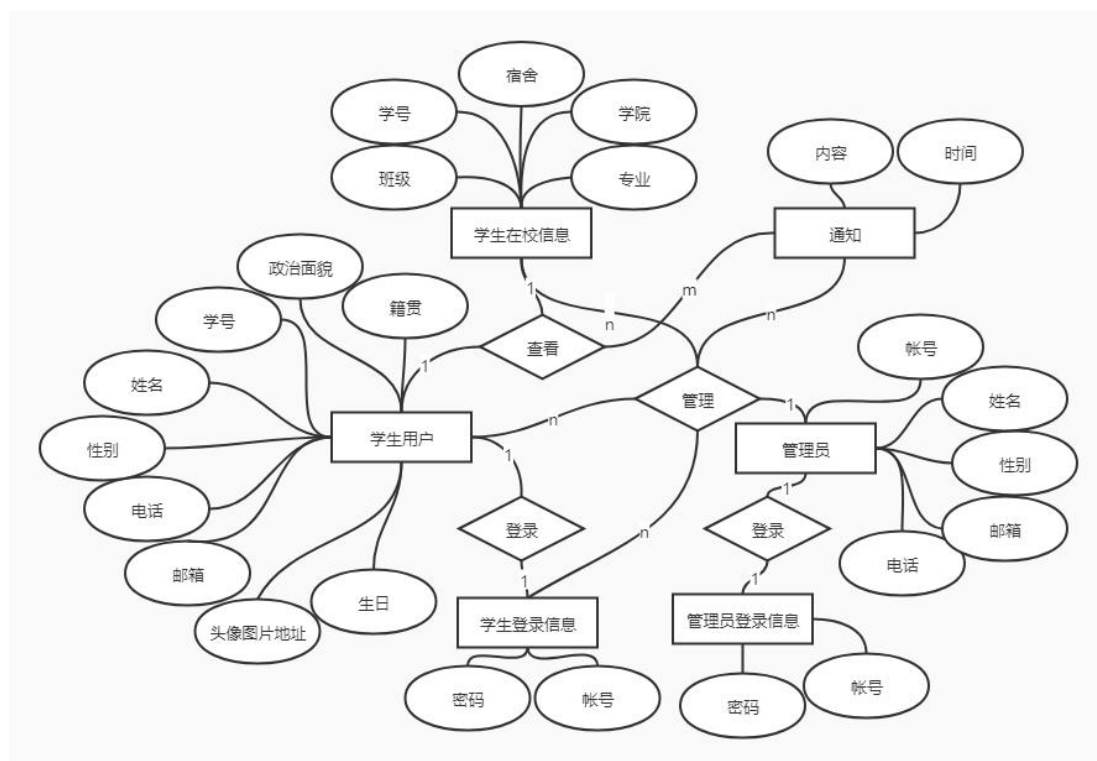
在进行特定条件查询操作执行后可以对查询信息进行统计分析。

### 5、登录密码管理模块：

登录后选择学生登录信息入口可进行增重置学生登录密码，防止学生忘记密码。

## 4. 数据库设计

### 4.1 E-R 图



## 4.2 数据库表的设计

数据库名:student

为本系统设计的表结构及其解释如下表表格:

表 2-1 数据表说明

表名	表的作用
manager_log	存放管理员的登录信息
notice	存放管理员发布的通知
stud_info	存放学生的个人信息
stud_school	存放学生在校信息
student_log	存放学生的登录信息

下面是各个表的详细结构:

1. manager\_log 表, 见表 2-2:

表 2-2 manager\_log 表

字段名称	数据类型	是否允许空	是否主键
mno	文本	否	是
password	文本	否	否

2.notice 表, 见表 2-3:

表 2-3 notice 表

字段名称	数据类型	是否允许空	是否主键
date	日期	否	是
info	文本	否	否

3.stud\_info 表, 见表 2-4:

表 2-4 stud\_info 表

字段名称	数据类型	是否允许空	是否主键
sno	文本	否	是
sname	文本	否	否
ssex	文本	否	否
birth	日期	否	否
native_place	文本	否	否
political_status	文本	否	否
phone	文本	否	否
email	文本	否	否
pic	文本	否	否

4.stud\_school 表，见表 2-5:

表 2-5 stud\_school 表

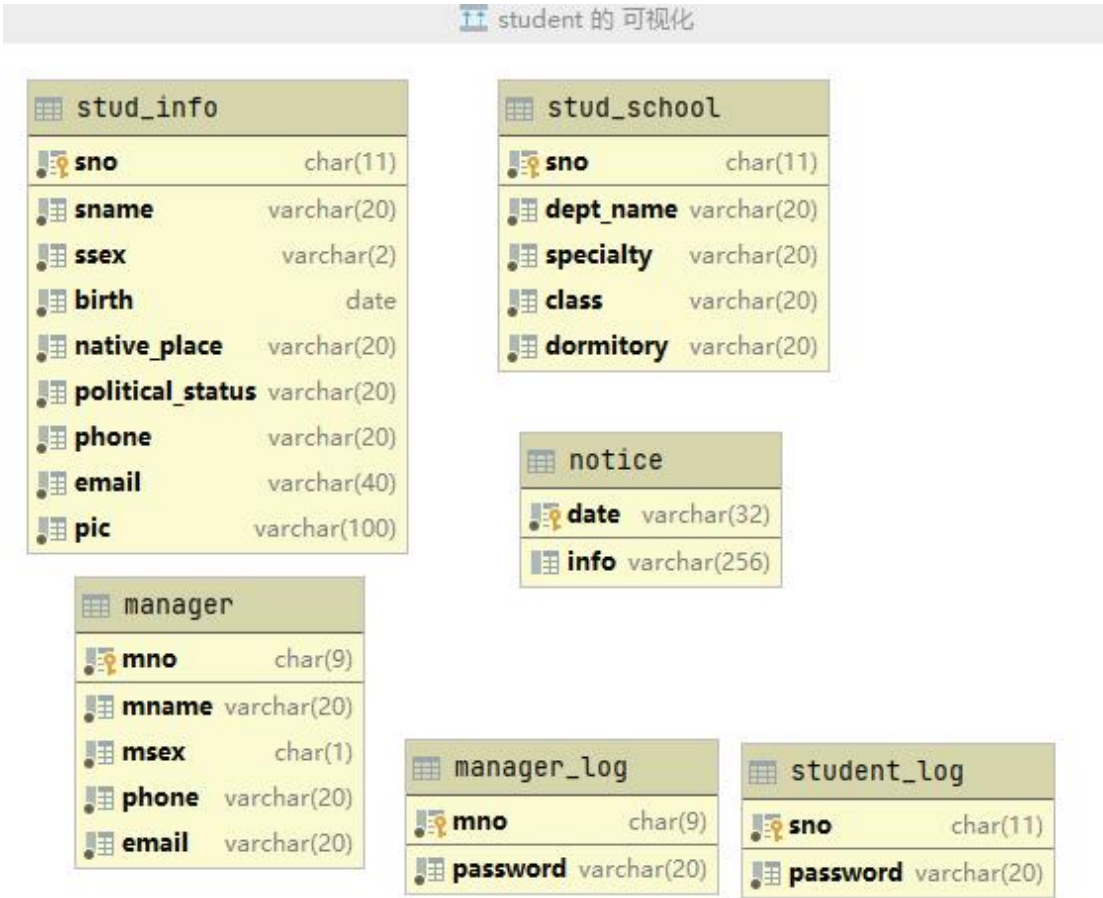
字段名称	数据类型	是否允许空	是否主键
sno	文本	否	是
dept_name	文本	否	否
specialty	文本	否	否
class	文本	否	否
dormitory	文本	否	否

5. stud\_log 表，见表 2-6:

表 2-6 stud\_log 表

字段名称	数据类型	是否允许空	是否主键
sno	文本	否	是
password	文本	否	否

4.3 数据关系图



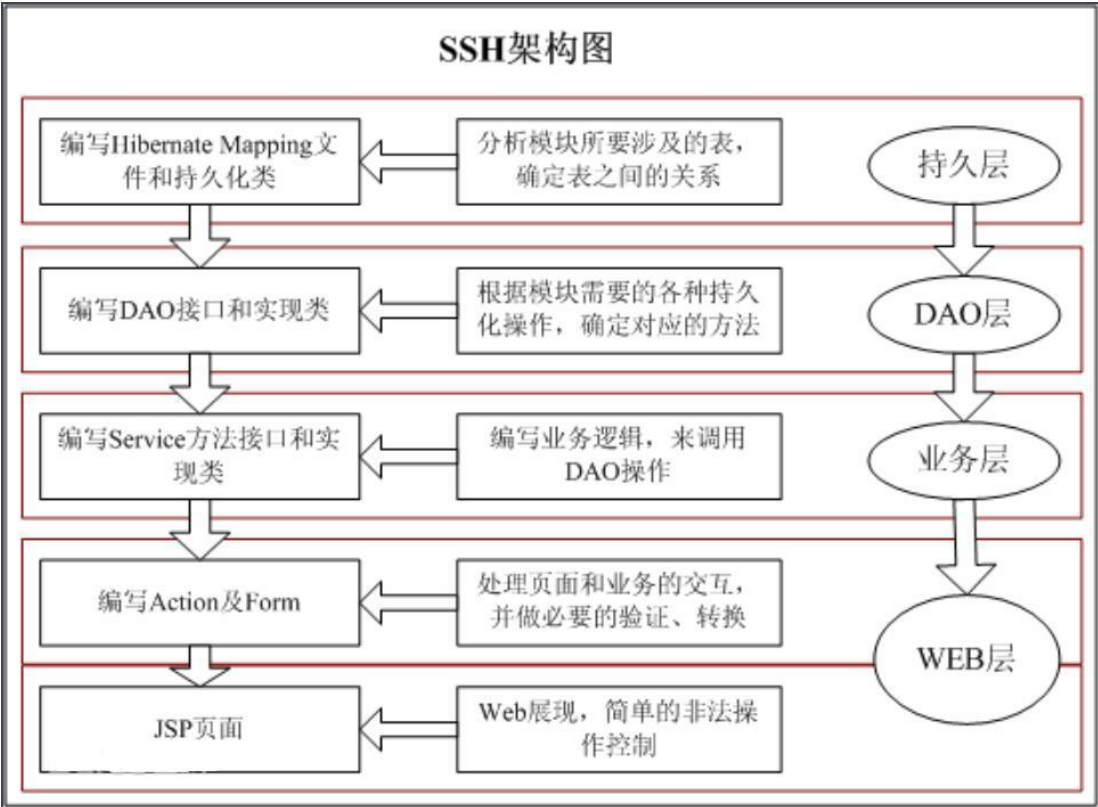
# 5. 系统实现

## 5.1 系统开发运行环境

操作系统: Windows 10  
Web 服务器: Tomcat 9.1  
IDE: IntelliJ IDEA 2020.3 (Ultimate Edition)  
Java 开发包: JDK 1.8.313  
数据库: MySQL5.7

## 5.2 总体技术

此系统采用 JSP+Struts+Hibernate+Spring 的开发模式，  
对象的调用流程如下图所示：



- 1.lib 包：用于支撑框架在服务器上运行。
- 2.Hibernate：是负责了跟数据库的交接。通过持久化数据对象，进行对象关系的映射，并以对象的角度来访问数据库。
- 3.Struts 技术基于 MVC 框架：Struts 的实现依赖 Servlet 和 JSP 实现。EJB 和 JavaBean 两个组件是 Struts 框架业务功能实现的基础部件；Action 和 ActionServlet 部件是框架实现控制力能的重要部件；视图部分则是由若干存在内在联系的 JSP 文件有机构成，来实现系统功能。
- 4.Spring，使用基本的实体 JavaBean 来完成以前只能用 EJB 完成的事情。而其核心则是控制反转（IOC）和面向切面编程（AOP）。

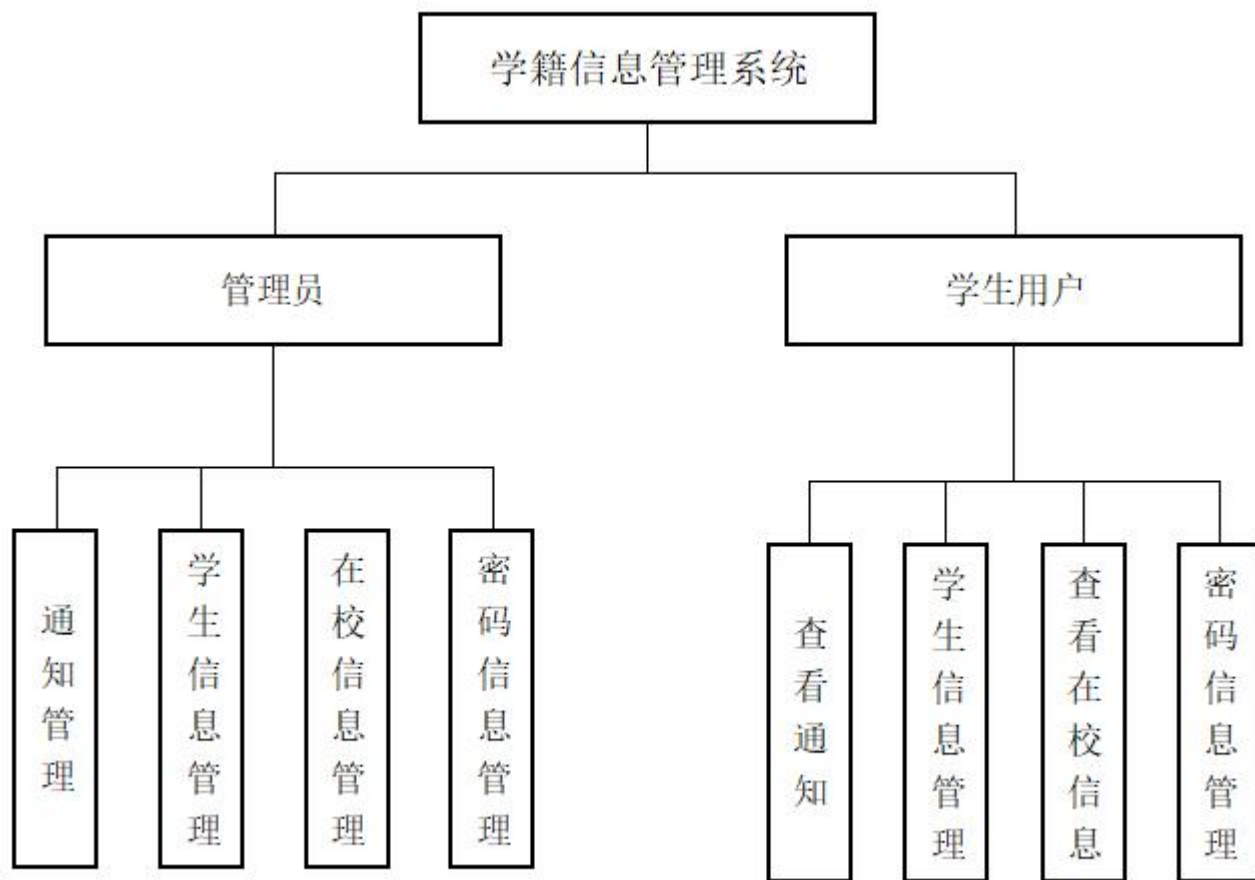


5. Dao 层:通过 Dao 层对数据库进行增删改查的操作,实现数据库与操作分类的目的。

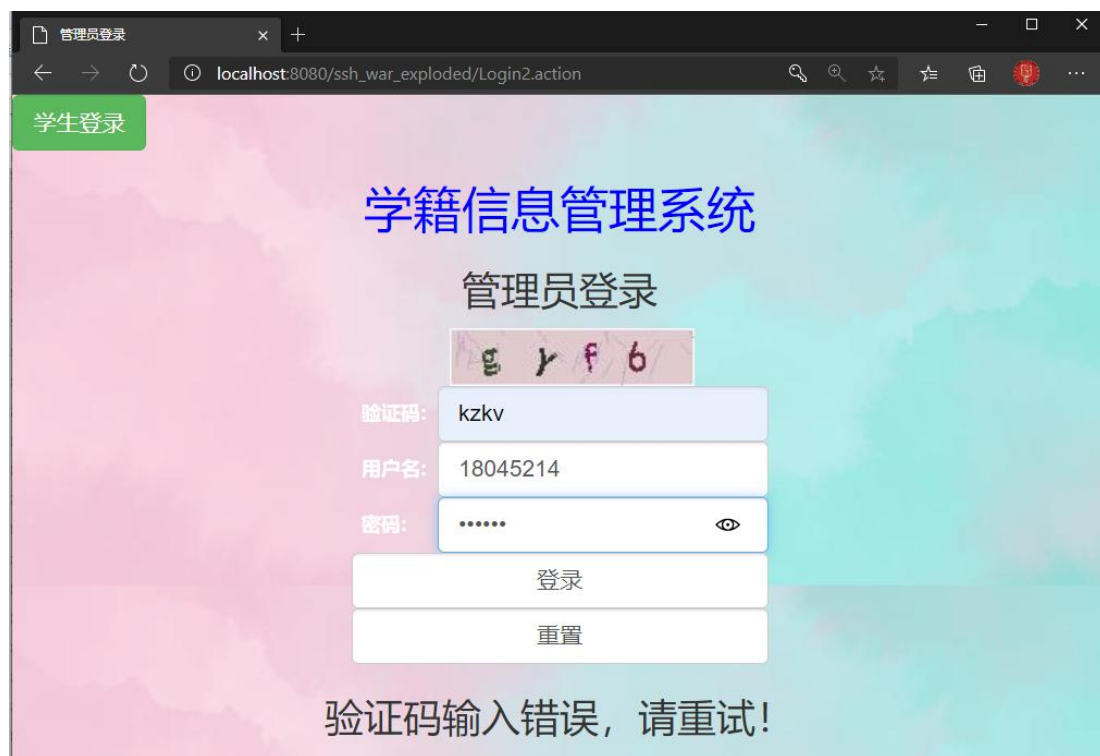
6. JSP 界面: 采用 Bootstrap 和 css 相结合对界面进行设计。

### 5.3 功能模块图

本次课程设计的学生信息管理系统主要分管理员和学生用户两个角色,主要是围绕着通知和学生个人信息、学生在校信息以及密码信息进行的一系列的管理操作



## 5.3 登录模块实现



管理员登录界面上图所示，登录前需要输入验证码，可以防止暴力破解登录信息。提交数据时验证用户数据的合法性和验证验证码的正误，同时本课程设计是利用服务端来实现验证码登录，区别于大多数利用网页内嵌 JavaScript 来实现，我们的验证必须经过服务端验证，不会被本地端暴力破解，核心代码如下：

```
<s:form action="Login2" method="post" enctype = "multipart/form-data">
    <s:textfield name="checkNum" label="验证码" class="form-control"/>
    <label for="checkImg">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</label>" onclick="change()" alt="验证码" title="点击更换验证码"
aria-label="验证码"/>
    <s:textfield name="user.sno" label="用户名" class="form-control"/>
    <s:password name="user.password" maxlength="16" label="密码" class="form-control"/>
    <s:submit value=" 登 录 " class="form-control"/><s:reset value=" 重 置 "
class="form-control"/>
</s:form>
<h3><s:property value="msg"/> </h3>
<script>
    function change(){ //切换验证码
        const img1 = document.getElementById("checkImg");
        img1.src="<s:url action="checkImg"/>?"+new Date().getTime(); //加时间戳防止缓存
    }
</script>
public String login2() throws Exception {
    msg="";
```

```

String checked =(String)
ServletContext.getRequest().getSession().getAttribute("checked");
if(checkNum.equalsIgnoreCase(checked)){
    ManagerLogEntity tmp=userDAO.select2(user.getSno());
    if(tmp==null){
        msg="用户帐号不存在！ ";
        return "error";
    }
    ManagerLogEntity tmp2 = new ManagerLogEntity();
    tmp2.setMno(user.getSno());
    tmp2.setPassword(user.getPassword());
    boolean judge=tmp.equals(tmp2);
    if(judge){
        Map<String, Object> session=ActionContext.getContext().getSession();
        session.put("user", user.getSno());
        return "success";
    }else {
        msg="密码输入错误，请重试！ ";
        return "error";
    }
} else{
    msg="验证码输入错误，请重试！ ";
    return "error";
}
}

```



验证码实现核心代码如下：

// 步骤一 绘制一张内存中图片

```

BufferedImage bufferedImage = new BufferedImage(width, height,
BufferedImage.TYPE_INT_RGB);

```

// 步骤二 图片绘制背景颜色 ---通过绘图对象

```

Graphics graphics = bufferedImage.getGraphics();// 得到画图对象 --- 画笔

```

// 绘制任何图形之前 都必须指定一个颜色

```

graphics.setColor(getRandColor(180, 240));

```

```

graphics.fillRect(0, 0, width, height);

```

// 步骤三 绘制边框

```

graphics.setColor(Color.WHITE);

```

```

graphics.drawRect(0, 0, width - 1, height - 1);

```

```
// 步骤四 四个随机数字
Graphics2D graphics2d = (Graphics2D) graphics;
// 设置输出字体
graphics2d.setFont(new Font("黑体", Font.BOLD, 25));
String words = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890";
Random random = new Random();// 生成随机数
// 将生成的字母存入到 session 中
ServletContext.getRequest().getSession().setAttribute("checked", sb.toString());
// 步骤五 绘制干扰线
然后利用 struct2 框架生成 URL 传到显示页面显示。
action name="checkImg" method="checkImg" class="checkImgAction"/>
<action name="Login" class="loginAction" method="login">
    <result name="success">/main.jsp</result>
    <result name="error">/index.jsp</result>
    <result name="input">/index.jsp</result>
</action>
```

## 5.4 通知管理模块实现

对通知信息进行管理，可以添加通知和修改、删除通知，同时还可以为通知添加附带详细文件方便用户下载文件查看了解更多的信息。





```
<table style="width: auto; margin:10px auto" class="table table-hover">
```

```
<tr>
```

```
<th align="center">通知时间</th>
```

```
<th align="center">通知内容</th>
```

```
<th align="center">操作</th>
```

```
</tr>
```

```
<s:set var="tmp" value="#session['notice2']"/>
```

```
<s:iterator value="tmp">
```

```
<tr>
```

```
<td><s:property value="date"/></td>
```

```
<td width="300" bgcolor="#f0f8ff"><div><s:property value="info"/></div></td>
```

```
<td>
```

```
<a href="changeNotice.action?id=<s:property value="date"/>">
```

```
<button>修改</button>
```

```
</a>
```

```
<a href="deleteNotice.action?id=<s:property value="date"/>">
```

```
<button onclick="test()">删除</button>
```

```
</a>
```

```
</td>
```

```
</tr>
```

```
</s:iterator>
```

```
</table>
```

```
public String findNotice() {
```

```
    msg = "";
```

```
    Map<String, Object> session = ActionContext.getContext().getSession();
```

```
    session.remove("notice2");
```

```
    List<NoticeEntity> tmp = manDAO.findNotice(condition);
```

```
    msg = condition + ",条件搜索结果如下";
```

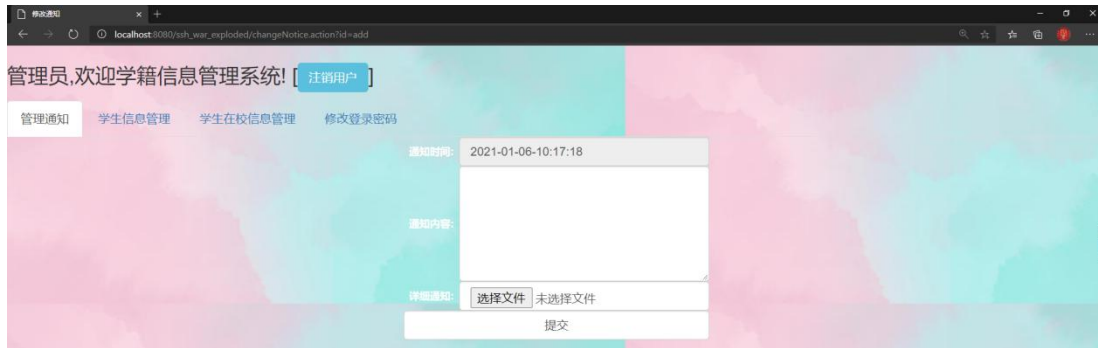
```
    session.put("notice2", tmp);
```

```
    System.out.println("搜索通知成功！");
```

```
    return "success";
```

```
}
```

添加和修改通知页面复用同一个页面，当进入添加页面时自动设置时间，修改自动获取要修改的信息填入方便修改。



```
public String changeNotice() {
    msg = "";
    ActionContext context = ActionContext.getContext();
    String id = context.getParameters().get("id").getValue();
    if (id == null) {
        if (manDAO.saveNotice(noticeEntity) != null) {
            if (file != null) {
                String destPath = "D:\\IdeaProjects\\ssh\\web\\upfile\\"; //你要保存的路
                径
                File destFile = new File(destPath, noticeEntity.getDate().hashCode() +
                ".pdf");

                try {
                    System.out.println(destFile.toString());
                    FileUtils.copyFile(file, destFile);
                    copyFileUsingFileChannels(file, destFile);
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            msg = "保存通知成功! ";
        } else {
            msg = "保存通知失败! ";
        }
    }
}
```



```

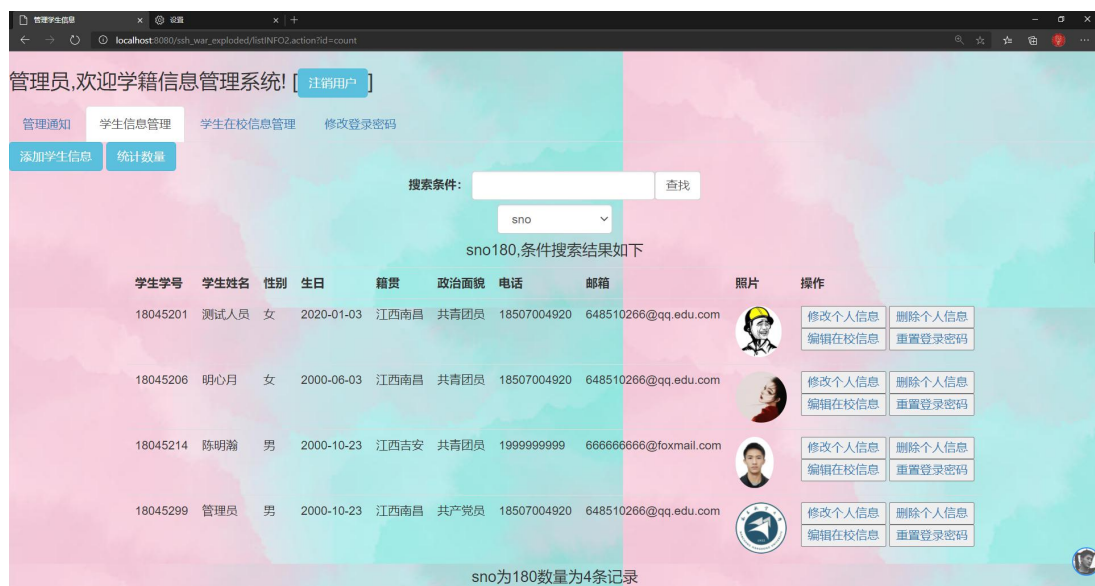
    } else if (id.trim().compareTo("add") == 0) {
        noticeEntity = new NoticeEntity();
        SimpleDateFormat ft = new SimpleDateFormat("yyyy-MM-dd-hh:mm:ss");
        noticeEntity.setDate(ft.format(new Date()));
    } else {
        noticeEntity = manDAO.selectOne(id);
        System.out.println("跳转到修改通知界面");
    }
    return "success";
}
}

```

本功能代码复用以方便扩展其它功能，只需修改在调用 action 时传不同的取值即可。

## 5.5 学生信息管理模块实现

对学生个人信息进行管理，可以添加学生信息和修改、删除学生信息，同时还可以为特定条件进行搜索后进行统计分析。同时还可以在操作栏对该学生进行更多操作，点击按钮即可快速编辑学生的在校信息或重置学生的登录密码方便管理员对于单个学生的其他信息管理，符合操作逻辑，方便使用。



实现的核心代码如下：

```
<a href="manINFOchange.jsp"><button class="btn btn-info">添加学生信息</button></a>
<a href="listINFO2.action?id=count"><button class="btn btn-info">统计数量</button></a>
<s:form action="listINFO2" method="POST" class="form-inline">
    <div class="form-group">
        <s:select name="kind"
list="{ 'sno','sname','ssex','birth','nativePlace','politicalStatus','phone','email'}"
class="form-control"/>
        <label>搜索条件:
            <input type="text" name="condition" class="form-control"/>
        </label>
        <input type="submit" value="查找" class="form-control"/>
    </div>
</s:form>
public String listINFO() {
    Map<String, Object> session = ActionContext.getContext().getSession();
    ActionContext context = ActionContext.getContext();
    String id = context.getParameters().get("id").getValue();
    if(id==null){
        msg = "";
        session.remove("stu2");
        List<StudInfoEntity> tmp = manDAO.listStu(kind, condition);
        session.put("stu2", tmp);
        if (condition != null) {
            msg = kind + condition + ",条件搜索结果如下";
        }
        System.out.println("查询全部学生个人信息成功！");
    }else if(id.trim().compareTo("count")==0){
        cntMsg="";
        System.out.println("count 操作传入");
        List<StudInfoEntity> tmp= (List<StudInfoEntity>) session.get("stu2");
        if(condition!=null)cntMsg=kind+"为"+condition+"数量为"+tmp.size()+"条记录";
        else cntMsg="数量为"+tmp.size()+"条记录";
        session.put("stu2", tmp);
        condition = null;
    }
    return "success";
}
```

18045206	明心月	女	2000-06-03	江西南昌	共青团员	18507004920	648510266@qq.edu.com		<a href="#">修改个人信息</a>	<a href="#">删除个人信息</a>
									<a href="#">编辑在校信息</a>	<a href="#">重置登录密码</a>
18045214	陈明瀚	男	2000-10-23	江西吉安	共青团员	19999999999	666666666@foxmail.com		<a href="#">修改个人信息</a>	<a href="#">删除个人信息</a>
									<a href="#">编辑在校信息</a>	<a href="#">重置登录密码</a>

快捷跳转操作传入学生的学号信息在服务器端进行查询后传到用户端后进行信息填充方便



查看和编辑，实现核心代码如下：

```
public String updateSchool2() {
    ActionContext context = ActionContext.getContext();
    msg = "";
    if (context.getParameters().get("id").getValue() == null) {
        System.out.println("修改在校信息页面信息传回");
        sch = new stuDAO().updateSchool(sch);
        if (sch != null) msg = "修改学生在校信息成功！";
        else msg = "修改在校信息失败！";
    } else {
        System.out.println("跳转到修改在校信息页面");
        msg = "";
        String id = context.getParameters().get("id").getValue();
        sch = new stuDAO().selectSchool(id);
        if (sch == null) {
            sch = new StudSchoolEntity();
            sch.setSno(id);
        }
    }
    return "success";
}

public String resetPass() {
    msg = "";
    ActionContext context = ActionContext.getContext();
    String id = context.getParameters().get("id").getValue();
    manDAO.saveLog(id, id);
    msg = id + "学号的登录密码重置成功！";
    return "success";
}
```

## 5.6 学生在校信息管理模块实现



对学生在校信息进行管理，可以添加学生在校信息和修改、删除学生在校信息，同时还可

以为特定条件进行搜索后进行统计分析。同时还可以在操作栏对该学生在校信息进行更多操作，点击按钮即可快速编辑学生的在校信息，方便管理员对于单个学生的其他信息管理，符合操作逻辑，方便使用。核心实现代码如下：

```
<a href="manSchoolChange.jsp"><button class="btn btn-info">添加学生在校信息</button></a>
<a href="findSchool.action?id=count"><button class="btn btn-info">统计数量</button></a>
<s:form action="findSchool" method="POST" class="form-inline">
  <div class="form-group">
    <s:select name="kind" list="{ 'sno','deptName','specialty','clazz','dormitory'}"
class="form-control"/>
    <label>搜索条件:
      <input type="text" name="condition" class="form-control"/>
    </label>
    <input type="submit" value="查找" class="form-control"/>
  </div>
</s:form>
<h3><s:property value="msg"/></h3>
```

快捷跳转操作传入学生的学号信息在服务器端进行查询后传到用户端后进行信息填充方便学生用户进行查看和编辑。



核心实现代码如下：

```
public String findSchool() {
    Map<String, Object> session = ActionContext.getContext().getSession();
    ActionContext context = ActionContext.getContext();
    String id = context.getParameters().get("id").getValue();
    if(id==null){
        System.out.println("无其余操作传入");
        msg = "";
        session.remove("school2");
        List<StudSchoolEntity> tmp = manDAO.listSchool(kind, condition);
        session.put("school2", tmp);
        if (condition != null) {
            msg = kind + ":" + condition + ",条件搜索结果如下";
        }
        System.out.println("查询全部学生在校信息成功！");
        condition = null;
    } else if(id.trim().compareTo("count")==0){
```

```

        System.out.println("count 操作传入");
        List<StudSchoolEntity> tmp= (List<StudSchoolEntity>) session.get("school2");
        msg="数量为"+tmp.size()+"条记录";
        session.put("school2", tmp);
        System.out.println("查询全部学生在校信息数量成功！");
    }
    return "success";
}

```

## 5.7 修改登录密码模块实现



修改登录密码界面如图所示，管理员可以修改登录密码，输入时要进行二次验证，增加了修改提示，确保了修改密码的安全性。核心实现代码如下：

```

public String changePass() {
    msg = "";
    Map<String, Object> session = ActionContext.getContext().getSession();
    String mno = (String) session.get("user");
    System.out.println("要修改密码的管理员号是: " + mno);
    int judge1 = password.trim().compareTo(new userDAO().select2(mno).getPassword());
    if (judge1 == 0) {
        if (password2.trim().compareTo(password3) == 0) {
            ManagerLogEntity tmp = new ManagerLogEntity();
            tmp.setMno(mno);
            tmp.setPassword(password3);
            tmp = new userDAO().saveManPass(tmp);
            if (tmp != null) msg = "修改管理员登录密码成功！";
            else msg = "修改管理员登录密码失败！";
        } else {
            msg = "两次输入的密码不一样！";
        }
    } else msg = "原来密码输入错误！";
    return "success";
}

```

## 6. 系统测试

### 6.1 登录模块测试

登录时会首先对验证码进行验证，然后是对于用户是否存在进行验证最后进行密码验证。



登录成功后会进入到管理员管理的首页面如后面图片所示。

## 6.2 通知管理模块测试

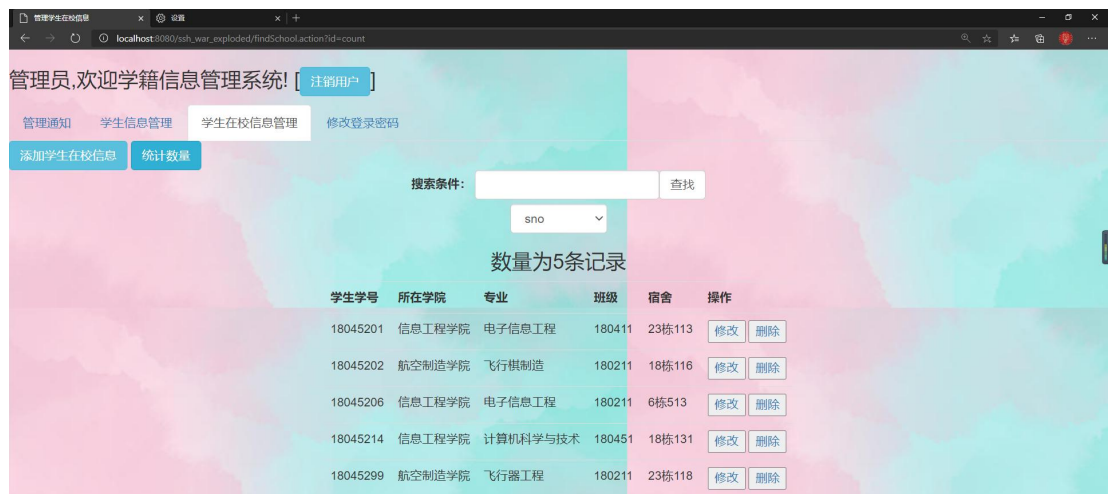




## 6.3 学生信息管理模块测试



## 6.4 学生信息管理模块测试



# 6.5 修改登录密码模块测试





## 7. 总结

课程设计就是简单的毕业设计，经过这次课设，给予我们实践的机会使我积累了必须的经验，为以后的学习打下了基础，也为毕业设计铺平了道路。经过这次的课程设计，使我学会了很多东西，包括书本上的理论知识及书本上学不到的知识。

最终，我对我们小组的成员同学表示衷心的感谢，因为我们是靠互帮互助才能完成课程设计的，我想其他同学也跟我有一样感受，同时也要感激我们的指导教师，教师严谨细致、一丝不苟的作风一向是我工作、学习中的榜样;教师循循善诱的教导和不拘一格的思路给予我无尽的启迪;这次模具设计的每个实验细节和每个数据，都离不开教师的细心指导。

学生信息档案的管理是非常复杂的，且数量十分庞大，管理起来需要消耗许多资源。随着信息技术的发展，各种信息技术的功能已经渗透到我们生活各个领域，发挥着巨大的作用。我们可以借助 web 移动网页这一先进的管理手段来提供一个电子化的学生档案管理平台，从而解决在管理上出现的问题。

通过本次课程设计，我们建立一个学生档案管理系统以便方便对学生的学籍信息以及在校期间的记录信息进行查询、管理，加深了我对于 Java EE 编程的理解能力，同时也加深了我对于 web 开发的理解深度。

本项目的可扩展性也非常的良好，方便移植改变成其它的人员管理系统，方便小型团体办公使用，其中通知功能可以方便人员快速获取信息，方便交流管理，同时对于人员信息管理系统有着非常大的参考意义。

## 8. 参考文献

- [1] 郭珍, 王国辉.JSP 程序设计教程(第 2 版)[M].北京:人民邮电出版社, 2012.10
- [2] 王小宁, 张广彬.JSP 课程设计案例精编(第 2 版)[M].北京:清华大学出版社, 2011.9
- [3] 李刚.轻量级 Java EE 企业应用实战(第 3 版):Struts 2+Spring 3+Hibernate 整合开发[M].北京: 电子工业出版社, 2012.
- [4] 张岫, 杨三成.关键技术: JSP 与 JDBC 应用详解[M].北京: 中国铁道出版社, 2010.11
- [5] 李兴华, 王月清. 名师讲坛:Java Web 开发实战经典基础篇(JSP, Servlet, Struts, Ajax)[M].北京: 清华大学出版社, 2010.
- [6] 张兴科. JSP 动态网站设计项目教程[M]. 北京: 中国人民大学出版社,2010.
- [7] 林信良.JSP&Servlet 学习笔记[M].北京: 清华大学出版社, 2012.5
- [8] 李兴华. Java 开发实战经典(名师讲坛)[M]. 北京: 清华大学出版社, 2009.
- [9] 孙卫琴.Tomcat 与 Java Web 开发技术详解(第 2 版)[M].北京: 电子工业出版社, 2009.01
- [10] 张银鹤.点石成金: JSP+Ajax 网站开发典型实例[M].北京: 电子工业出版社, 2009.01