武汉纺织大学

课 设 报 告

|  |  |
| --- | --- |
| 课程名称： | Javaweb |
| 专 业： | 软件工程 |
| 班 级： | 软工11808 |
| 学 号： | 1804230823 |
| 学生姓名： | 余明豪 |
| 指导老师： | 黄俊杰 |

2021年6月24日

目录

[一、课设目的 3](#_Toc75098548)

[二、开发环境 4](#_Toc75098549)

[三、课设内容 5](#_Toc75098550)

[3.1具体要求 7](#_Toc75098551)

[四、需求分析 7](#_Toc75098552)

[4.1系统功能需求分析 7](#_Toc75098553)

[4.2系统性能需求分析 7](#_Toc75098554)

[4.3用例图 8](#_Toc75098555)

[五、设计过程 9](#_Toc75098556)

[5.1数据库设计 9](#_Toc75098557)

[5.1.1数据库表的建立 9](#_Toc75098558)

[5.2登录 10](#_Toc75098559)

[5.3下载文件 11](#_Toc75098560)

[5.4上传文件 13](#_Toc75098561)

[5.5人员信息管理 18](#_Toc75098562)

[5.5.1添加人员 18](#_Toc75098563)

[5.5.2修改人员信息 18](#_Toc75098564)

[5.5.3查询人员信息 19](#_Toc75098565)

[5.5.4删除人员信息 19](#_Toc75098566)

[六、测试 19](#_Toc75098567)

[6.1学生登录测试 19](#_Toc75098568)

[6.2教师登录测试 20](#_Toc75098569)

[6.3管理员登录测试 21](#_Toc75098570)

[6.4下载文档测试 22](#_Toc75098571)

[6.5下载视频测试 22](#_Toc75098572)

[6.6添加教师测试 23](#_Toc75098573)

[6.7教师信息管理测试 24](#_Toc75098574)

[6.8学生信息管理测试 25](#_Toc75098575)

[6.9上传测试 26](#_Toc75098576)

[6.10注册页面测试 27](#_Toc75098577)

[七、调试过程 27](#_Toc75098578)

[7.1遇到的问题 27](#_Toc75098579)

[7.2问题的解决 28](#_Toc75098580)

[八、小结 29](#_Toc75098581)

[九、参考文献 30](#_Toc75098582)

# 课设目的

1.复习、巩固JavaWeb的基础知识，进一步加深对JavaWeb技术的理解和掌握。

2.课程设计为学生提供了一个既动手又动脑，独立实践的机会，将课本上的理论知识和实际有机的结合起来，锻炼学生的分析解决实际问题的能力。

3.培养学生在项目开发中团队合作精神、创新意识及能力。

# 二、开发环境

操作系统：Windows

开发工具：MyEclipse

MyEclipse 是一个十分优秀的用于开发Java, J2EE的 Eclipse 插件集合，MyEclipse的功能非常强大，支持也十分广泛，尤其是对各种开源产品的支持十分不错。MyEclipse可以支持Java Servlet，AJAX，JSP，JSF，Struts，Spring，Hibernate，EJB3，JDBC数据库链接工具等多项功能。可以说MyEclipse是几乎囊括了目前所有主流开源产品的专属eclipse开发工具。

Java中间件服务器：Tomcat

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好Apache 服务器，可利用它响应HTML（标准通用标记语言下的一个应用）页面的访问请求。实际上Tomcat是Apache 服务器的扩展，但运行时它是独立运行的，所以当你运行tomcat 时，它实际上作为一个与Apache 独立的进程单独运行的。

数据库： MySQL

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

# 三、课设内容

师生交流系统是一个教学系统，主要是帮助学生和教师更好进行交流，更好地将若干课程更加紧密地结合起来，更好地达到学以致用；达到教、学结合的目的；让学生能够更加完整而系统地掌握相关知识和技能。

根据师生交流系统的基本需求，本系统需要完成以下任务：

|  |  |  |
| --- | --- | --- |
| 角色 | 操作 | 操作详情 |
| 学生 | 登录师生交流系统 | 学生输入账户、密码、验证码之后，通过后台验证。登录进师生交流系统学生端。 |
| 退出学生交流系统 | 学生完成目标操作后退出师生交流系统。 |
| 下载视频 | 学生选择需要下载的视频进行下载。 |
| 下载文档 | 学生选择需要下载的文档进行下载。 |
| 教师 | 登录师生交流系统 | 教师输入账户、密码、验证码之后，通过后台验证。登录进师生交流系统学生端。 |
| 退出学生交流系统 | 教师完成目标操作后退出师生交流系统。 |
| 上传资料 | 教师选择本地资料进行上传。 |
| 资料信息管理 | 教师选择具体资料，进行删除等操作。 |
| 上传视频 | 教师选择本地视频进行上传。 |
| 视频信息管理 | 教师选择具体视频进行删除等相关操作。 |
| 管理员 | 登录师生交流系统 | 管理员输入账户、密码、验证码之后，通过后台验证。登录进师生交流系统学生端。 |
| 退出学生交流系统 | 管理员完成目标操作后退出师生交流系统。 |
| 添加教师 | 管理员填写教师的工号、密码、姓名等信息后，上传到系统后台。 |
| 教师信息管理 | 管理员可以对教师信息进行浏览、查询、修改、删除等操作。 |
| 添加学生 | 管理员填写学生的学号、姓名、密码等信息后，上传到系统后台。 |
| 学生信息管理 | 管理员可以对学生信息进行浏览、查询、修改、删除等相关操作。 |

表（1）—角色功能

## 3.1具体要求

1.使用三层框架，每个功能要尽可能的贴近实际，合理全面。

2.对系统进行功能模块分析、控制模块分析正确，符合课题要求，实现相应功能；可以加以其他功能或修饰，使程序更加完善、合理；

3.系统设计要实用，采用模块化程序设计方法，编程简练、可用，功能全面；

说明书、流程图要清楚；

4.记录设计情况（备查，也为编写设计说明书作好准备）

# 四、需求分析

## 4.1系统功能需求分析

师生交流系统在增进师生交流中占有重要的地位，它为教师办公提供便捷、为学生学习提供有力支持。对于学校来讲，师生交流系统是网络化教学不可缺少的组成部分。

本系统根据实际要求，结合学生学习和教师办公的实际流程，需实现以下功能：

1.学生能够对资料、视频进行下载。

2.教师能够上传资料、视频，并对资料、视频信息进行管理。

3.管理员能够添加教师、学生，并对教师、学生信息进行管理。

## 4.2系统性能需求分析

为了保证系统能够长期、安全、稳定、可靠、高效的运行，学生信息管理系统应该满足以下的性能需求：

① 系统处理的准确性和及时性

系统处理的准确性和及时性是系统的必要性能。在系统设计和开发过程中，要充分考虑系统当前和将来可能承受的工作量，使系统的处理能力和响应时间能够满足学校对信息处理的需求。

② 系统的开放性和系统的可扩充性

学生信息管理系统在开发过程中，应该充分考虑以后的可扩充性。而要实现这一点，应通过系统的开放性来完成，既系统应是一个开放系统，只要符合一定的规范，可以简单的加入和减少系统的模块，配置系统的硬件。通过软件的修补、替换完成系统的升级和更新换代。

③ 系统的易用性和易维护性

要求系统能够提供良好的用户接口，易用的人机交互界面。要实现这一点，就要求系统应该尽量使用用户熟悉的术语和中文信息的界面；针对用户可能出现的使用问题，要提供足够的帮助，缩短用户对系统熟悉的过程。

## 4.3用例图



# 五、设计过程

## 5.1数据库设计

### 5.1.1数据库表的建立

1.t-admain：管理员账户信息表

2.t-stu：学生账户信息表

3.t-tea：视频信息

4.t-mv：视频信息管理

5.t-doc：文档信息管理

| 字段名 | 字段描述 | 数据类型 | 主键 | 外键 | 非空 | 唯一 | 自增 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| adld | 账号 | INT(4) | 是 | 否 | 是 | 是 | 否 |
| adname | 姓名 | CHAR(10) | 否 | 否 | 是 | 是 | 否 |
| adPwd | 密码 | VARCHAR(9) | 否 | 否 | 是 | 否 | 否 |

表（2）t-admain表结构

| 字段名 | 字段描述 | 数据类型 | 主键 | 外键 | 非空 | 唯一 | 自增 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| stuld | ID | INT(8) | 是 | 否 | 是 | 是 | 否 |
| stupwd | 密码 | INT(4) | 否 | 否 | 是 | 否 | 否 |
| stuname | 姓名 | CHAR(10) | 否 | 否 | 是 | 否 | 否 |
| Stusex | 性别 | VARCHAR(8) | 否 | 否 | 是 | 否 | 否 |
| stuatus | 审核状态 | INT(2) | 否 | 否 | 是 | 否 | 否 |

表（3）t-stu表结构

| 字段名 | 字段描述 | 数据类型 | 主键 | 外键 | 非空 | 唯一 | 自增 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| teald | ID | INT(8) | 是 | 否 | 是 | 是 | 否 |
| teapwd | 密码 | INT(4) | 否 | 否 | 是 | 否 | 否 |
| teaname | 姓名 | CHAR(10) | 否 | 否 | 是 | 否 | 否 |
| teasex | 性别 | VARCHAR(8) | 否 | 否 | 是 | 否 | 否 |
| teaatus | 审核状态 | INT(2) | 否 | 否 | 是 | 否 | 否 |

表（4）t-tea表结构

| 字段名 | 字段描述 | 数据类型 | 主键 | 外键 | 非空 | 唯一 | 自增 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| mvld | ID | INT(8) | 是 | 否 | 是 | 是 | 否 |
| mvname | 名字 | CHAR(15) | 否 | 否 | 是 | 否 | 否 |
| mvpath | 路径 | CHAR(10) | 否 | 否 | 是 | 否 | 否 |
| upload | 上传时间 | VARCHAR(8) | 否 | 否 | 是 | 否 | 否 |

表（5）t-mv表结构

| 字段名 | 字段描述 | 数据类型 | 主键 | 外键 | 非空 | 唯一 | 自增 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| docld | ID | INT(8) | 是 | 否 | 是 | 是 | 否 |
| docname | 名字 | CHAR(15) | 否 | 否 | 是 | 否 | 否 |
| docpath | 路径 | CHAR(10) | 否 | 否 | 是 | 否 | 否 |
| upload | 上传时间 | VARCHAR(8) | 否 | 否 | 是 | 否 | 否 |

表（6）t-doc表结构

## 5.2登录

1.用户输入用户信息和验证码后，系统验证账号信息是否合法。

2.首先获取验证码，验证验证码是否输入正确。验证码正确继续验证账户信息，向下进行，如果验证码错则要求用户重新输入。

3.然后获取表单信息，验证用户名和密码。

4.成功就保存用户信息到session，并重定向succ1.jsp。

5. 如果失败，保存错误信息到requset，转发到login.jsp。 (服务器内部跳转只有一个request请求，重定向会有新的request，就不能获取错误信息)

## 5.3下载文件

以下是下载文件的核心代码和注解：

String action = request.getParameter("action");  
 if ("findAllFile".equals(action) || "findAllVideo".equals(action)) {  
 try {  
 findAllFile(request, response);  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 } else if ("downloadfile".equals(action)) {  
 //得到要下载的文件名  
 String fileName = request.getParameter("filename"); //23239283-92489-阿凡达.avi  
 fileName = new String(fileName.getBytes("iso8859-1"),"UTF-8");  
 /\*int index = fileName.lastIndexOf("\\");  
 if (index != -1) {//如果文件名中有\，则表示有路径，需截取  
 fileName = fileName.substring(index + 1);  
 }\*/  
 //上传的文件都是保存在/WEB-INF/upload目录下的子目录当中  
 String fileSaveRootPath=this.getServletContext().getRealPath("uploadFile");  
 // String fileSaveRootPath=this.getServletContext().getRealPath("WEB-INF/upload");  
 //通过文件名找出文件的所在目录  
 String path = findFileSavePathByFileName(fileName,fileSaveRootPath);  
 //得到要下载的文件  
 File file = new File(path + "\\" + fileName);  
 //如果文件不存在  
 if(!file.exists()){  
 request.setAttribute("message", "您要下载的资源已被删除！！");  
 request.getRequestDispatcher("/message.jsp").forward(request, response);  
 return;  
 }  
 //处理文件名  
 String realname = fileName.substring(fileName.indexOf("\_")+1);  
 //设置响应头，控制浏览器下载该文件  
 response.setHeader("content-disposition", "attachment;filename=" + URLEncoder.*encode*(realname, "UTF-8"));  
 //读取要下载的文件，保存到文件输入流  
 FileInputStream in = new FileInputStream(path + "\\" + fileName);  
 //创建输出流  
 OutputStream out = response.getOutputStream();  
 //创建缓冲区  
 byte buffer[] = new byte[1024];  
 int len = 0;  
 //循环将输入流中的内容读取到缓冲区当中  
 while((len=in.read(buffer))>0){  
 //输出缓冲区的内容到浏览器，实现文件下载  
 out.write(buffer, 0, len);  
 }  
 //关闭文件输入流  
 in.close();  
 //关闭输出流  
 out.close();  
 }  
}

## 5.4上传文件

1.查询所有文件信息，调用Service中的findAllFile或findAllVideo方法

2.将文件放在request中

3.请求转发给InfoManage.jsp或VideoInfoManage.jsp

\*创建工厂对象

\*使用工厂对象创建解析器

\*通过解析器解析request请求，获取FileItem集合

\*通过循环遍历依次取出表单项，如果是文件则存入磁盘。

核心代码：

private void downloadFile(HttpServletRequest request, HttpServletResponse response) throws SQLException, ServletException, IOException {  
 //查询所有文件信息，调用Service中的findAllFile方法  
 List<Map<String, Object>> list = FileService.*findAllFile*();  
 System.*out*.println(list);  
 //将文件放在request中  
 request.getSession().setAttribute("list", list);  
 //请求转发给InfoManage.jsp  
 request.getRequestDispatcher("/stuDownloadInfo.jsp").forward(request, response);}  
private void findAllFile(HttpServletRequest request, HttpServletResponse response) throws SQLException, ServletException, IOException {  
 //查询所有文件信息，调用Service中的findAllFile方法  
 List<Map<String, Object>> list = FileService.*findAllFile*();  
 System.*out*.println(list);  
 //将文件放在request中  
 request.getSession().setAttribute("list", list);  
 //请求转发给InfoManage.jsp  
 request.getRequestDispatcher("/InfoManage.jsp").forward(request, response);  
}  
private void addFile(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException, SQLException {  
 request.setCharacterEncoding("utf-8");  
 response.setContentType("text/html;charset=utf-8");  
 //得到上传文件的保存目录，将上传的文件存放于WEB-INF目录下，不允许外界直接访问，保证上传文件的安全  
 //String savePath = this.getServletContext().getRealPath("/WEB-INF/upload");  
 //上传时生成的临时文件保存目录  
 String tempPath = this.getServletContext().getRealPath("/WEB-INF/temp");  
 java.io.File file = new java.io.File(tempPath);  
 if (!file.exists() && !file.isDirectory()) {  
 file.mkdir();  
 }  
 //消息提示  
 String message = "";  
 try {  
 //使用Apache文件上传组件处理文件上传步骤：  
 //1、创建一个DiskFileItemFactory工厂  
 DiskFileItemFactory diskFileItemFactory = new DiskFileItemFactory();  
 //设置工厂的缓冲区的大小，当上传的文件大小超过缓冲区的大小时，就会生成一个临时文件存放到指定的临时目录当中。  
 diskFileItemFactory.setSizeThreshold(1024 \* 100);  
 //设置上传时生成的临时文件的保存目录  
 diskFileItemFactory.setRepository(file);  
 //2、创建一个文件上传解析器  
 ServletFileUpload fileUpload = new ServletFileUpload(diskFileItemFactory);  
 //解决上传文件名的中文乱码  
 fileUpload.setHeaderEncoding("UTF-8");  
 //监听文件上传进度  
 fileUpload.setProgressListener(new ProgressListener() {  
 public void update(long pBytesRead, long pContentLength, int arg2) {  
 //System.out.println("文件大小为：" + pContentLength + ",当前已处理：" + pBytesRead);  
 }  
 });  
 //3、判断提交上来的数据是否是上传表单的数据  
 if (!fileUpload.*isMultipartContent*(request)) {  
 //按照传统方式获取数据  
 return;  
 }  
 //设置上传单个文件的大小的最大值，目前是设置为1024\*1024字节，也就是1MB  
 fileUpload.setFileSizeMax(1024 \* 1024\*10);  
 //设置上传文件总量的最大值，最大值=同时上传的多个文件的大小的最大值的和，目前设置为10MB  
 fileUpload.setSizeMax(1024 \* 1024 \* 20);  
 //4、使用ServletFileUpload解析器解析上传数据，解析结果返回的是一个List<FileItem>集合，每一个FileItem对应一个Form表单的输入项  
 List<FileItem> list = fileUpload.parseRequest(request);  
 for (FileItem item : list) {  
 //如果fileItem中封装的是普通输入项的数据  
 if (item.isFormField()) {  
 String name = item.getFieldName();  
 //解决普通输入项的数据的中文乱码问题  
 String value = item.getString("UTF-8");  
 String value1 = new String(name.getBytes("iso8859-1"), StandardCharsets.*UTF\_8*);  
 System.*out*.println(name + " " + value);  
 System.*out*.println(name + " " + value1);  
 } else {  
 //如果fileItem中封装的是上传文件，得到上传的文件名称，  
 String fileName = item.getName();  
 //file1.setTitle(fileName);  
 if (fileName == null || fileName.trim().equals("")) {  
 continue;  
 }  
 //注意：不同的浏览器提交的文件名是不一样的，有些浏览器提交上来的文件名是带有路径的，如： c:\a\b\1.txt，而有些只是单纯的文件名，如：1.txt  
 //处理获取到的上传文件的文件名的路径部分，只保留文件名部分  
 fileName = fileName.substring(fileName.lastIndexOf(java.io.File.*separator*) + 1);  
 //得到上传文件的扩展名  
 String fileExtName = fileName.substring(fileName.lastIndexOf(".") + 1);  
 if ("zip".equals(fileExtName) || "rar".equals(fileExtName) || "tar".equals(fileExtName) || "jar".equals(fileExtName)) {  
 request.setAttribute("message", "上传文件的类型不符合！！！");  
 request.getRequestDispatcher("/teaUploadInfo.jsp").forward(request, response);  
 return;  
 }  
 //如果需要限制上传的文件类型，那么可以通过文件的扩展名来判断上传的文件类型是否合法  
 //System.out.println("上传文件的扩展名为:" + fileExtName);  
 //获取item中的上传文件的输入流  
 InputStream is = item.getInputStream();  
 //得到文件保存的名称  
 fileName = mkFileName(fileName);  
 //得到文件保存的路径  
 String savePathStr = mkFilePath("savePath", fileName);  
 System.*out*.println("保存路径为:" + savePathStr);  
 //创建一个文件输出流  
 FileOutputStream fos = new FileOutputStream(savePathStr + java.io.File.*separator* + fileName);  
 //创建一个缓冲区  
 byte buffer[] = new byte[1024];  
 //判断输入流中的数据是否已经读完的标识  
 int length = 0;  
 //循环将输入流读入到缓冲区当中，(len=in.read(buffer))>0就表示in里面还有数据  
 while ((length = is.read(buffer)) > 0) {  
 //使用FileOutputStream输出流将缓冲区的数据写入到指定的目录(savePath + "\\" + filename)当中  
 fos.write(buffer, 0, length);  
 }  
 //关闭输入流  
 is.close();  
 //关闭输出流  
 fos.close();  
 //删除处理文件上传时生成的临时文件  
 item.delete();  
 message = "文件上传成功";  
 }  
 }  
 } catch (FileUploadBase.FileSizeLimitExceededException e) {  
 e.printStackTrace();  
 request.setAttribute("message", "单个文件超出最大值！！！");  
 request.getRequestDispatcher("/uploadFile.jsp").forward(request, response);  
 return;  
 } catch (FileUploadBase.SizeLimitExceededException e) {  
 e.printStackTrace();  
 request.setAttribute("message", "上传文件的总的大小超出限制的最大值！！！");  
 request.getRequestDispatcher("/uploadFile.jsp").forward(request, response);  
 return;  
 } catch (FileUploadException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 message = "文件上传失败";  
 }  
 //file1.setTitle(name);  
 *file1*.setAuthor(request.getParameter("author"));  
 *file1*.setPath("savePath");  
 *file1*.setTime(new Date());  
 FileService.*addFile*(*file1*);  
 request.setAttribute("message", message);  
 request.getRequestDispatcher("/teaUploadInfo.jsp").forward(request, response);  
}

## 5.5人员信息管理

**Ps：**学生和教师信息管理基本相同，故统称为人员。

### 5.5.1添加人员

1.获取表单信息

2.将表单信息封装到user对象中

### 5.5.2修改人员信息

1.获取要修改人员的信息

2.将数据封装到user对象中

3.调用service中的update方法进行修改

4.调用finall方法显示修改后的人员信息列表

### 5.5.3查询人员信息

1.查询所有人员信息，调用service中的findall方法

2.将人员信息放在request中

3.请求转发给stuinfomanage.jsp

### 5.5.4删除人员信息

1.获取人员id

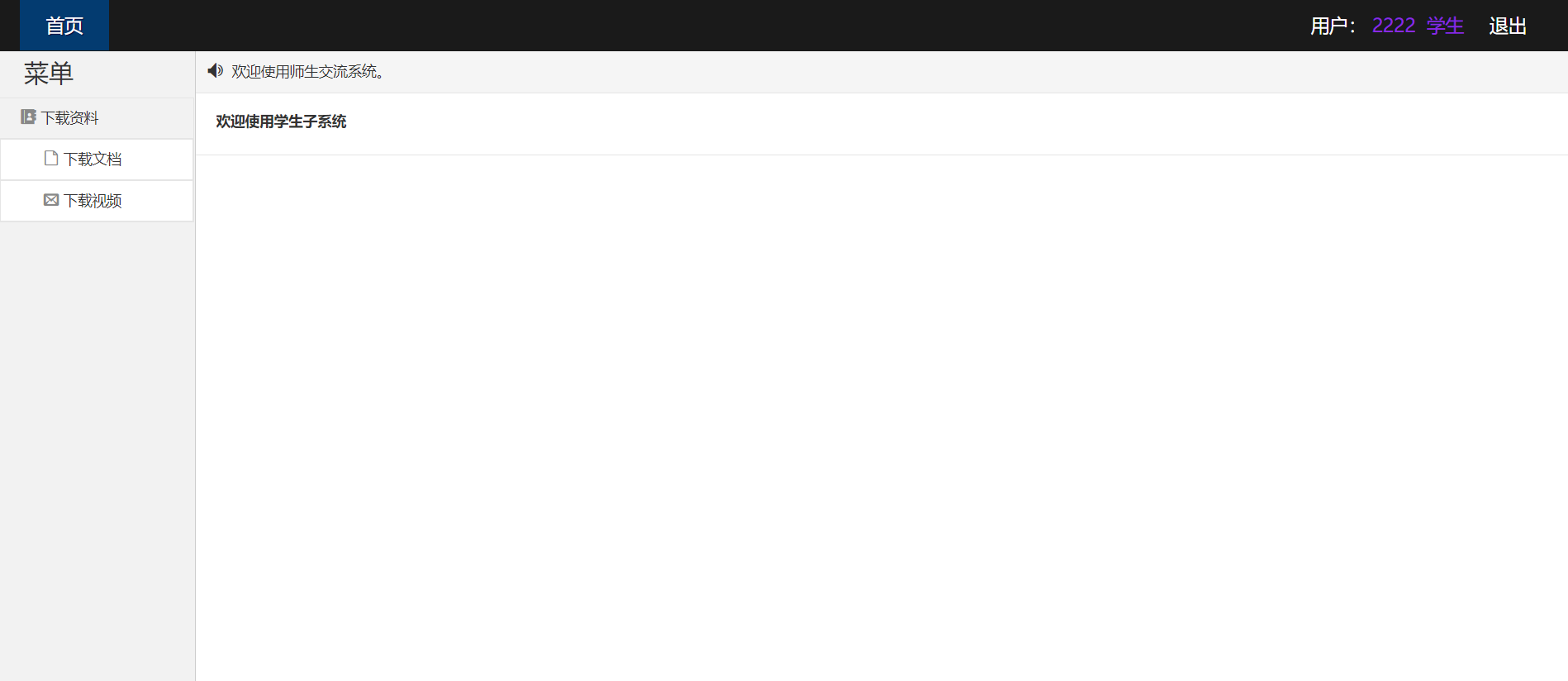
2.删除该人员的信息记录

# 六、测试

## 6.1学生登录测试



图一学生登录界面

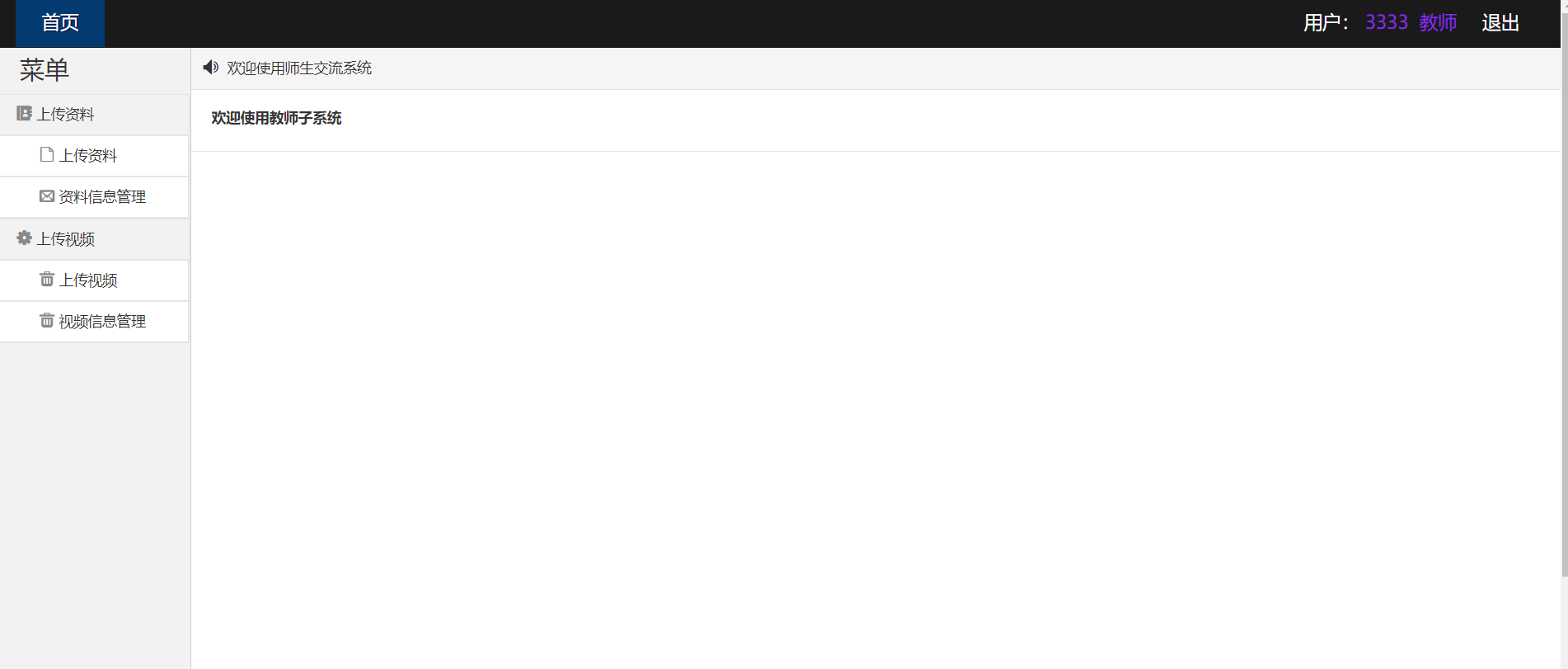


图二学生登录成功界面

## 6.2教师登录测试



图三教师登录界面

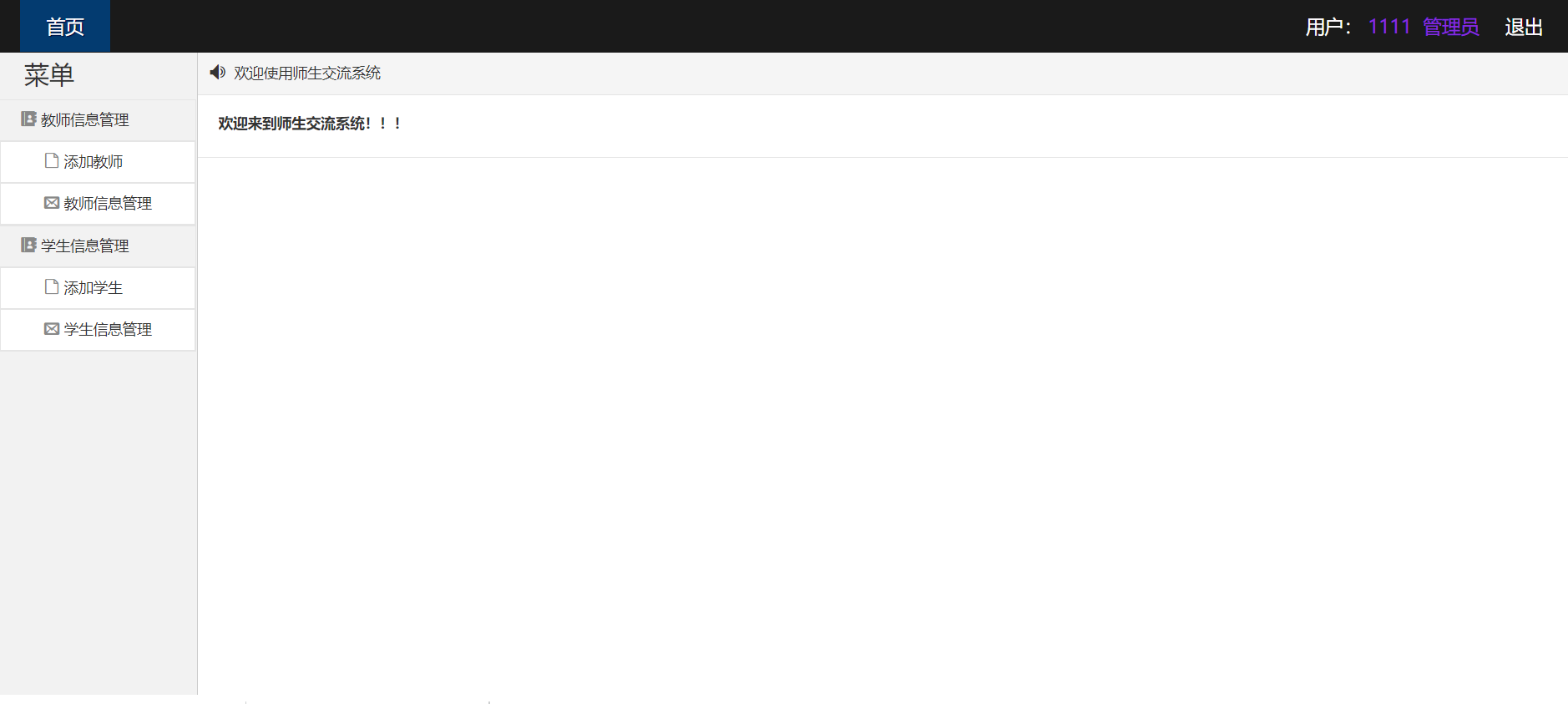


图四教师登录成功界面

## 6.3管理员登录测试

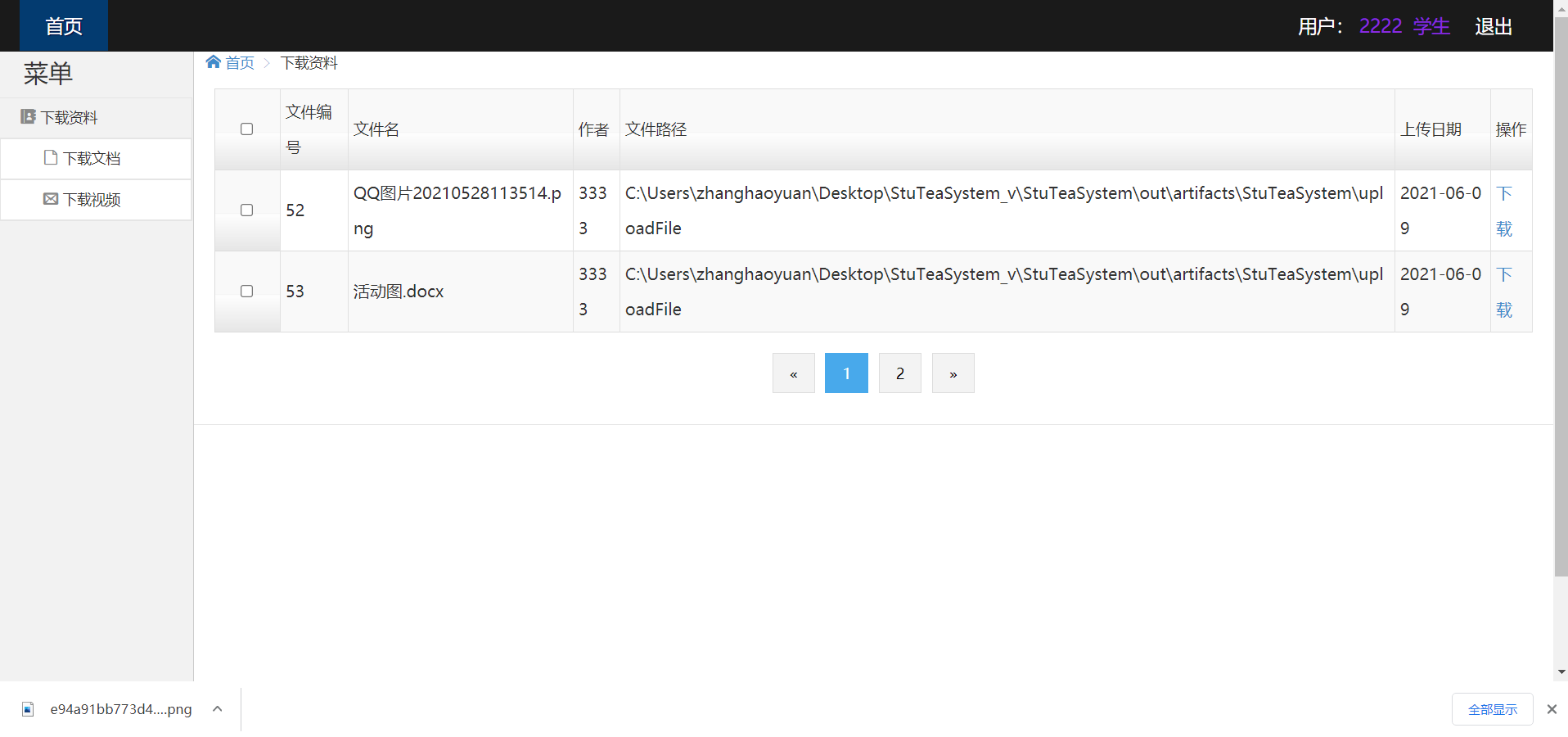


图五管理员登录界面



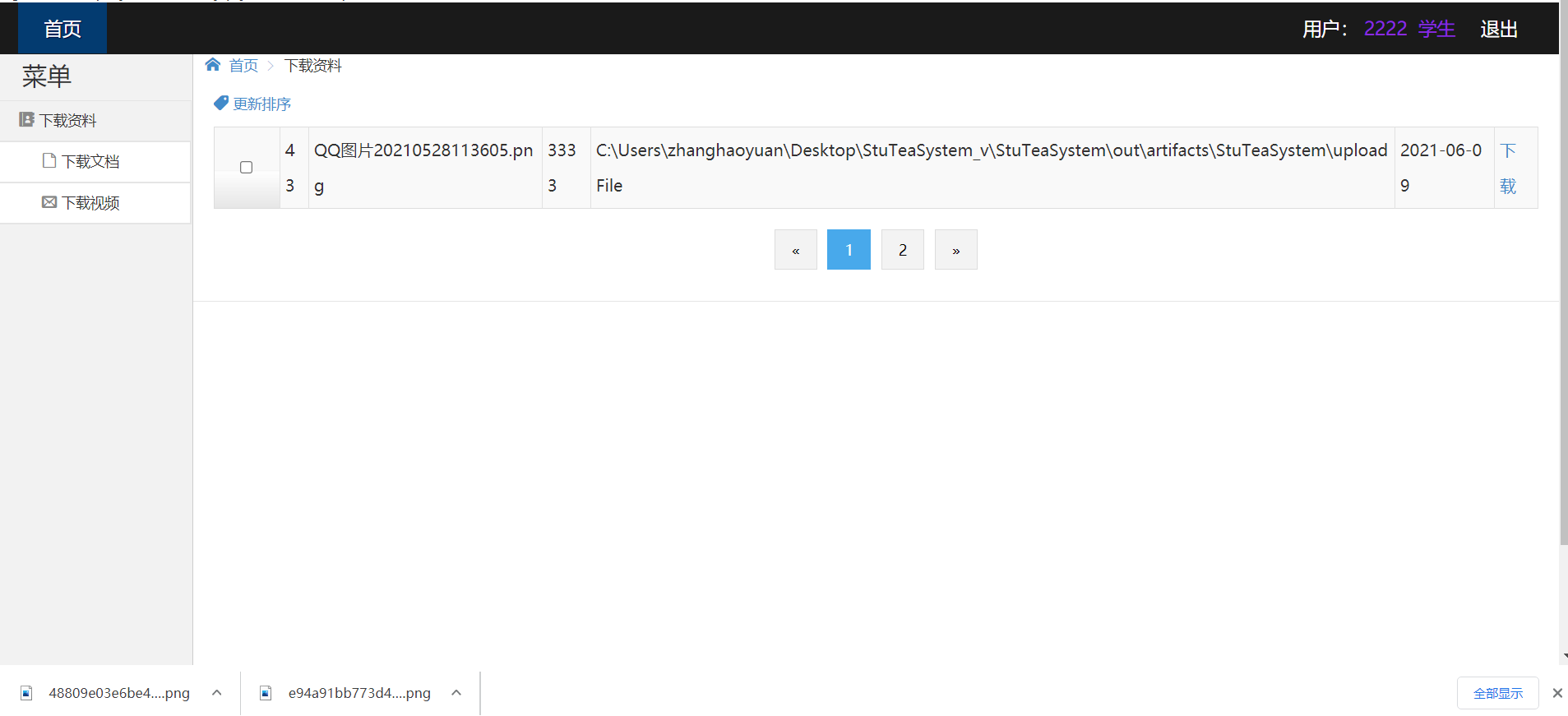
图六管理员登录成功界面

## 6.4下载文档测试



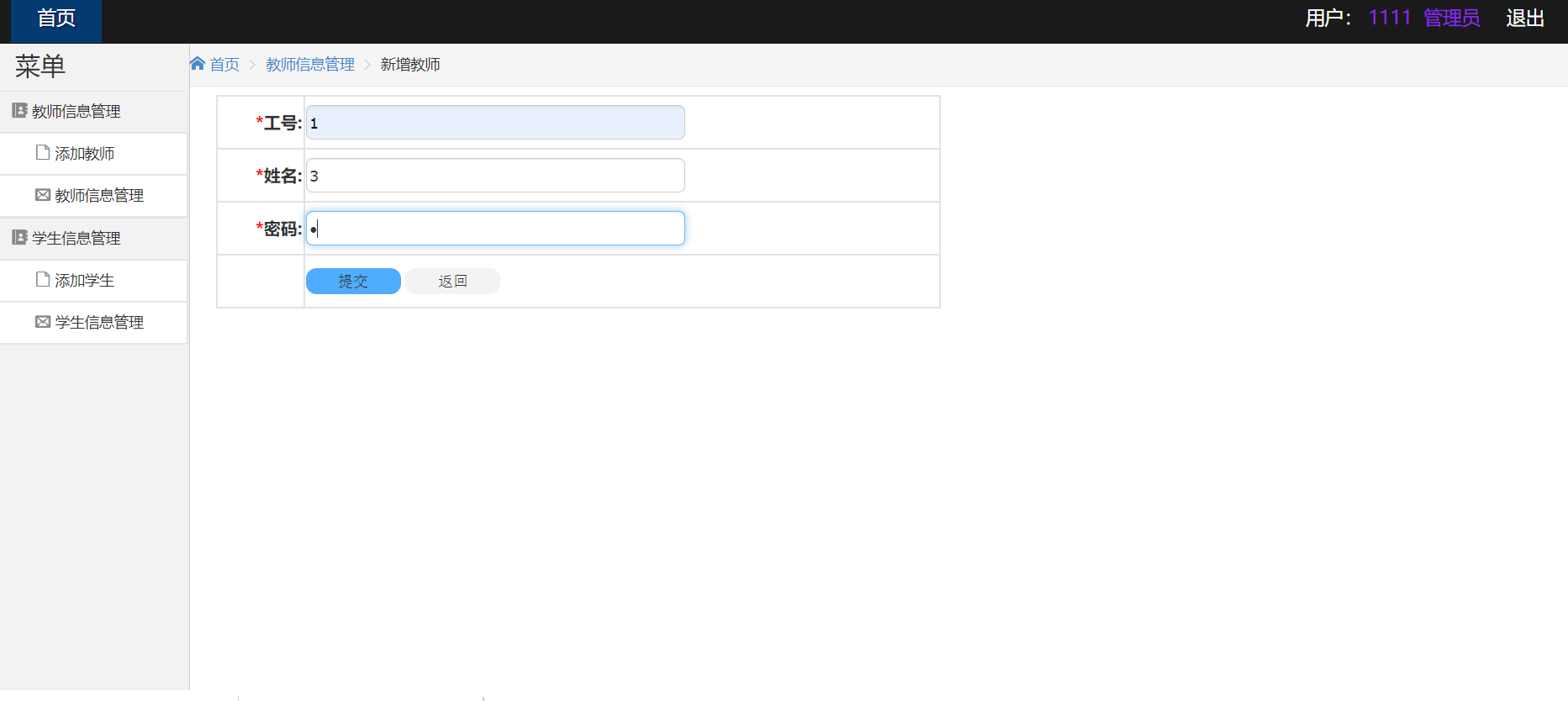
图七下载文档成功界面

## 6.5下载视频测试

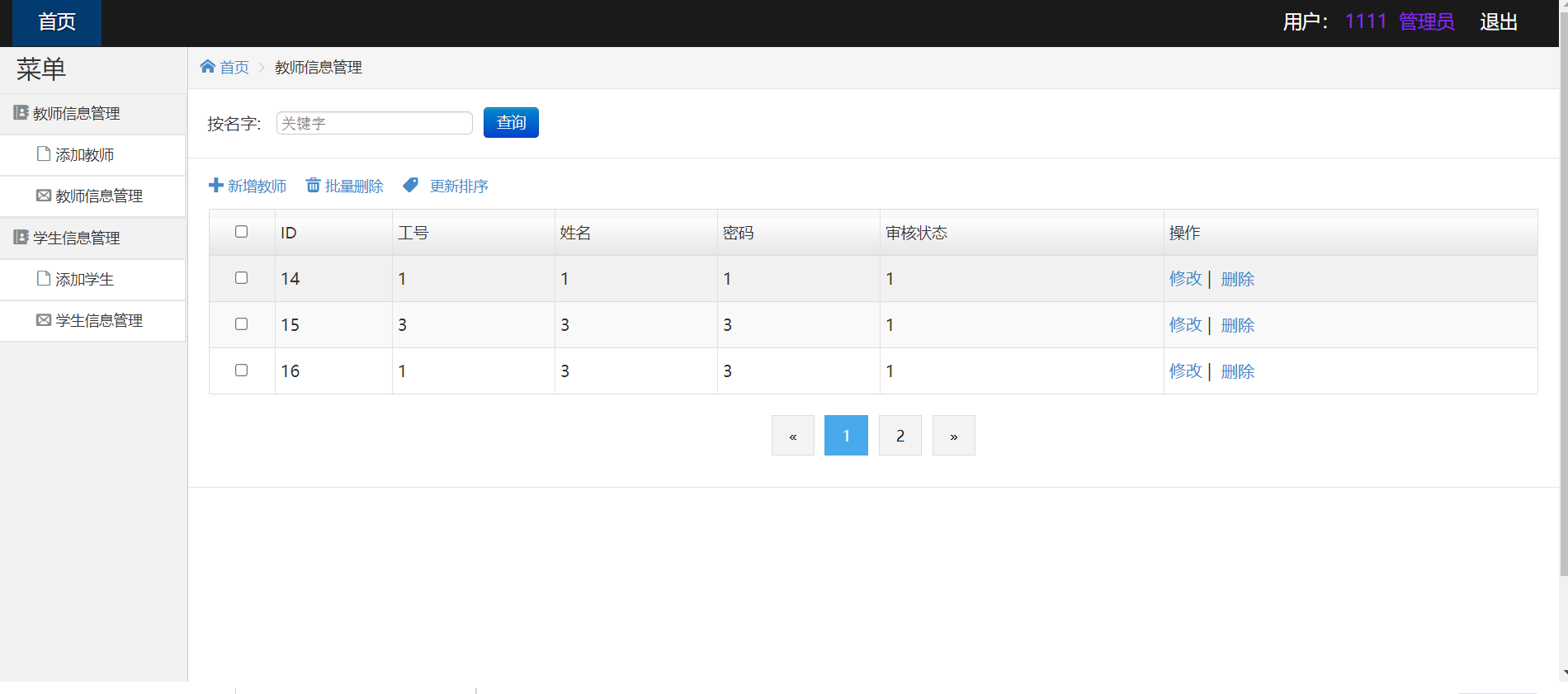


图八下载视频成功界面

## 6.6添加教师测试

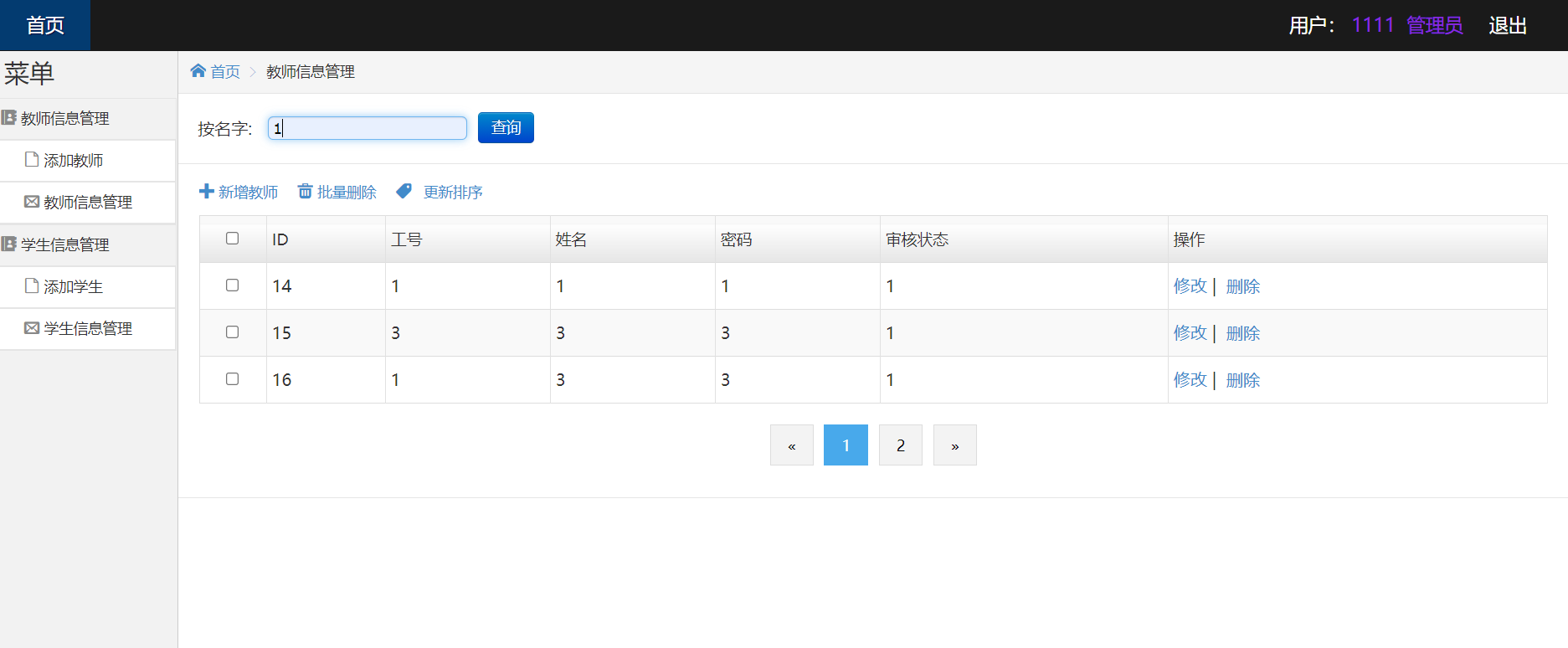


图九输入新增教师信息界面

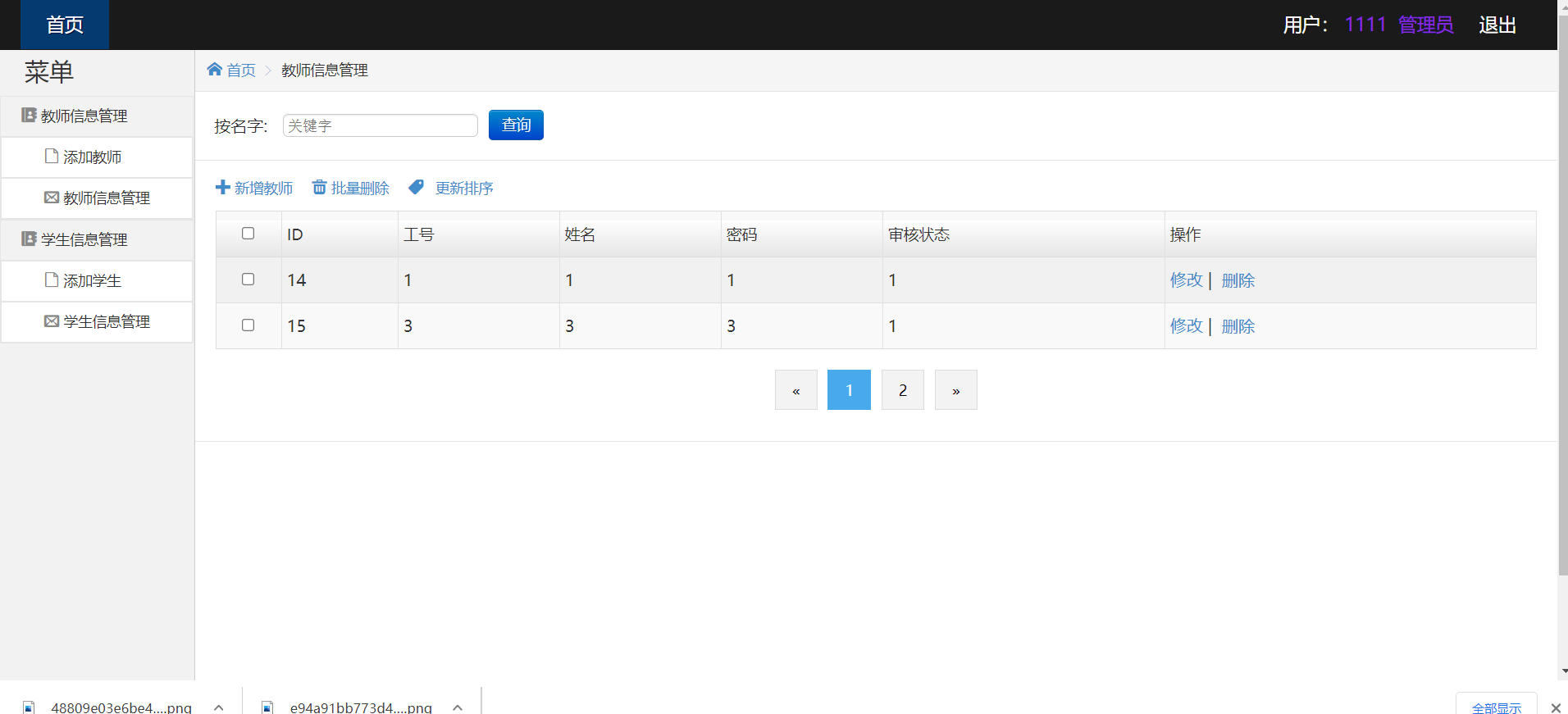


图十添加教师成功界面

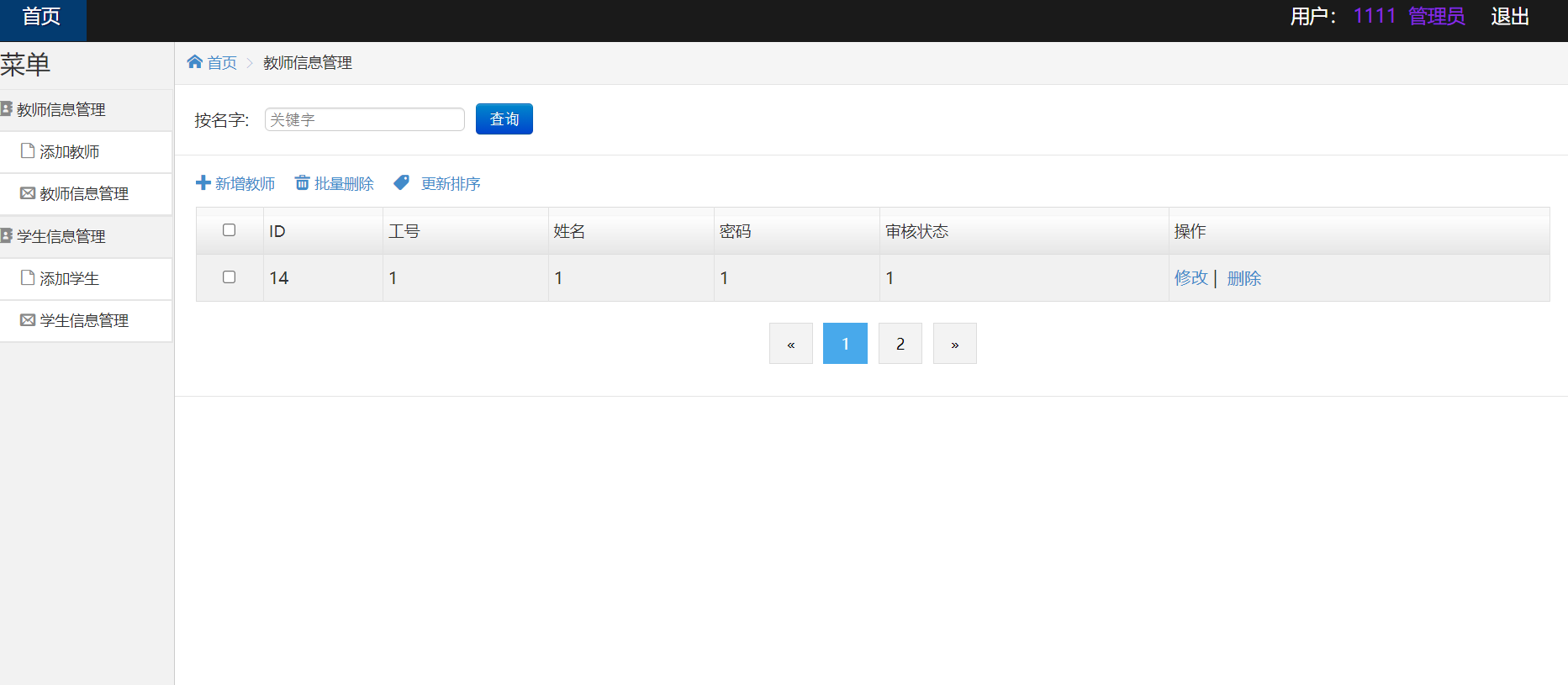
## 6.7教师信息管理测试



图十一教师信息管理初始界面

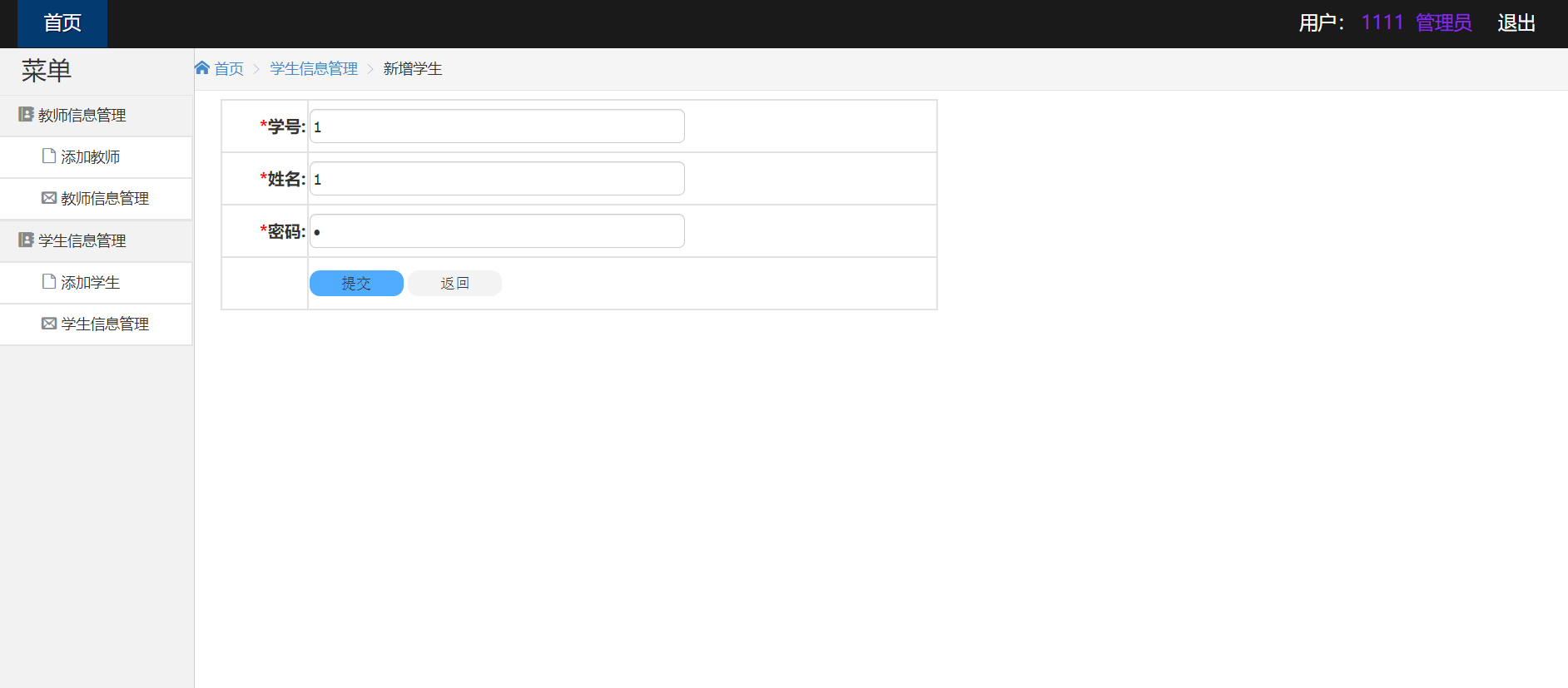


图十二教师信息删除界面

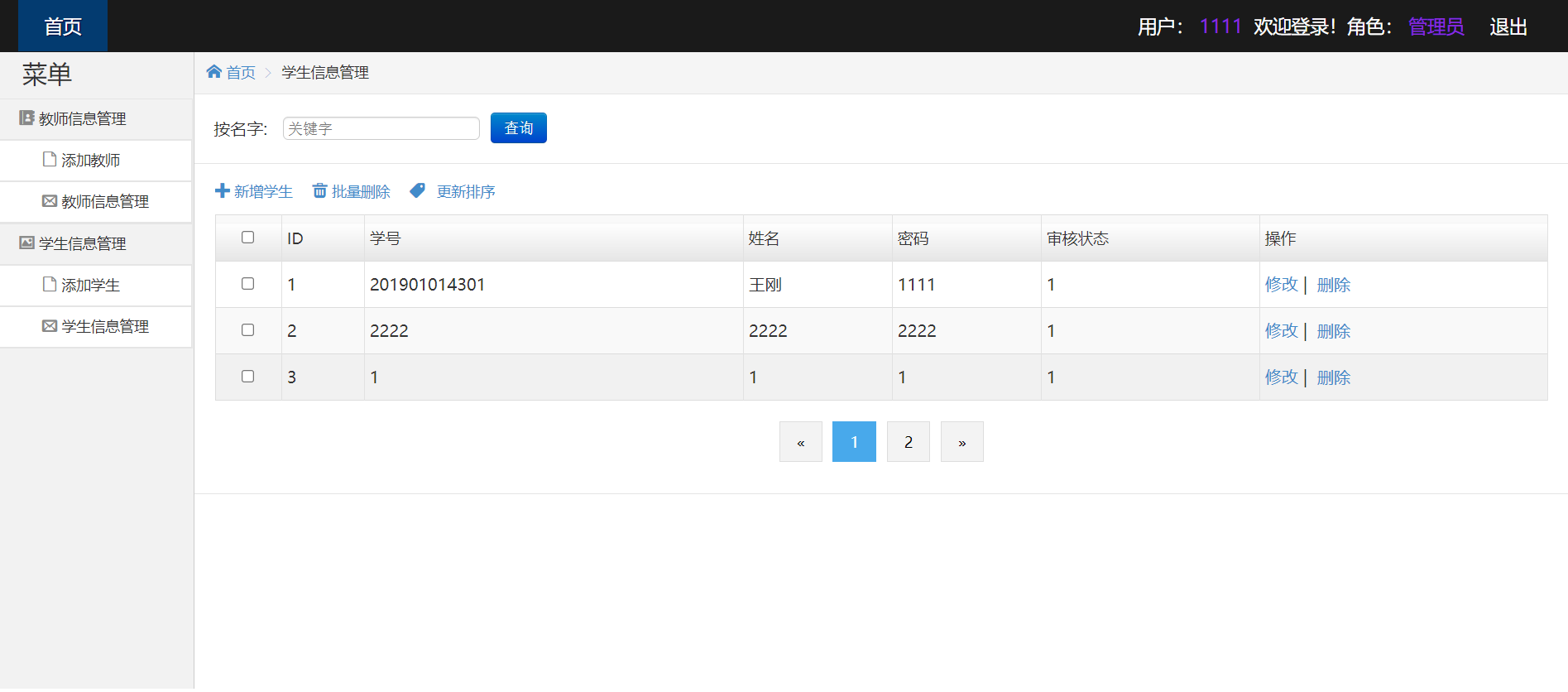


图十三教师信息查询界面

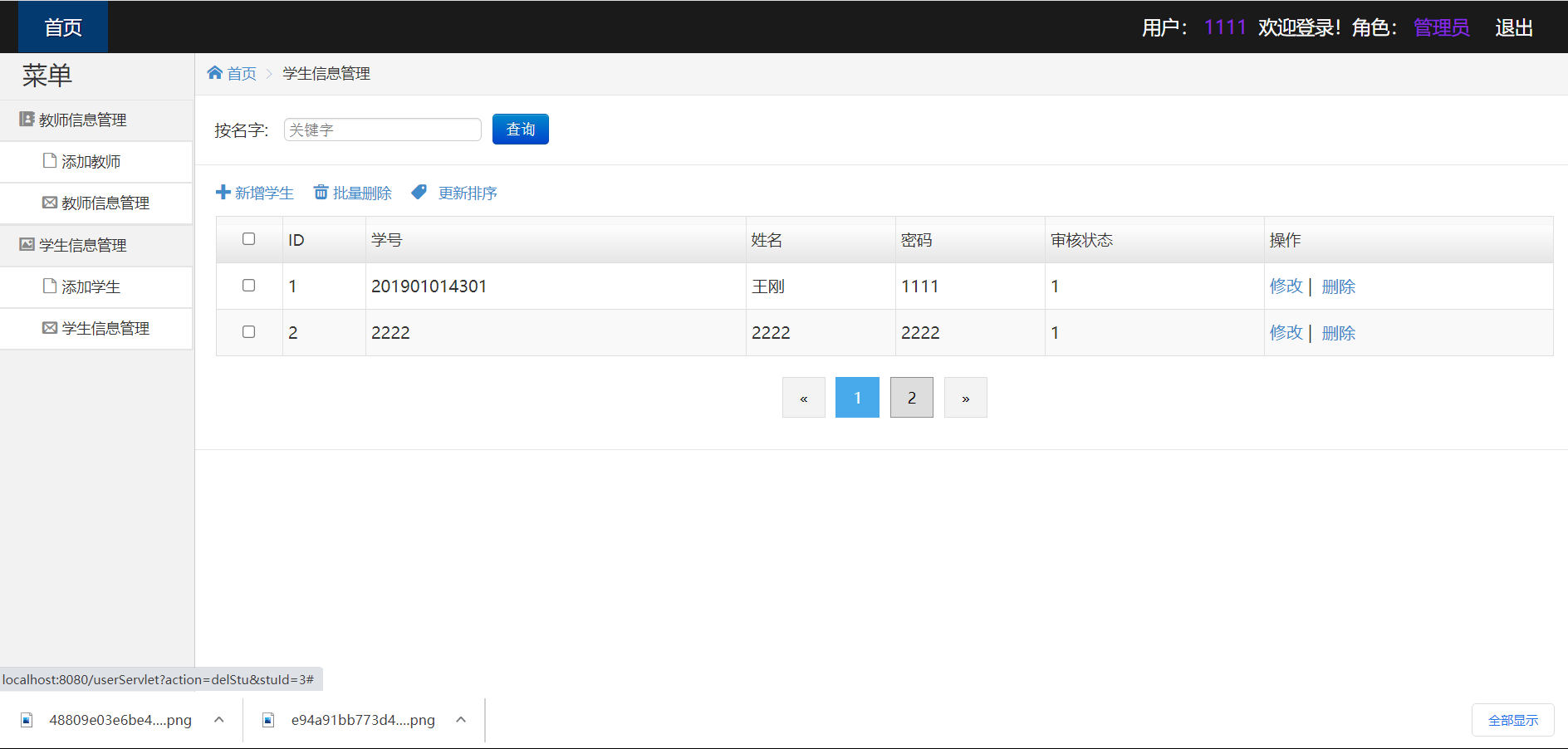
## 6.8学生信息管理测试



图十四输入新增学生信息界面

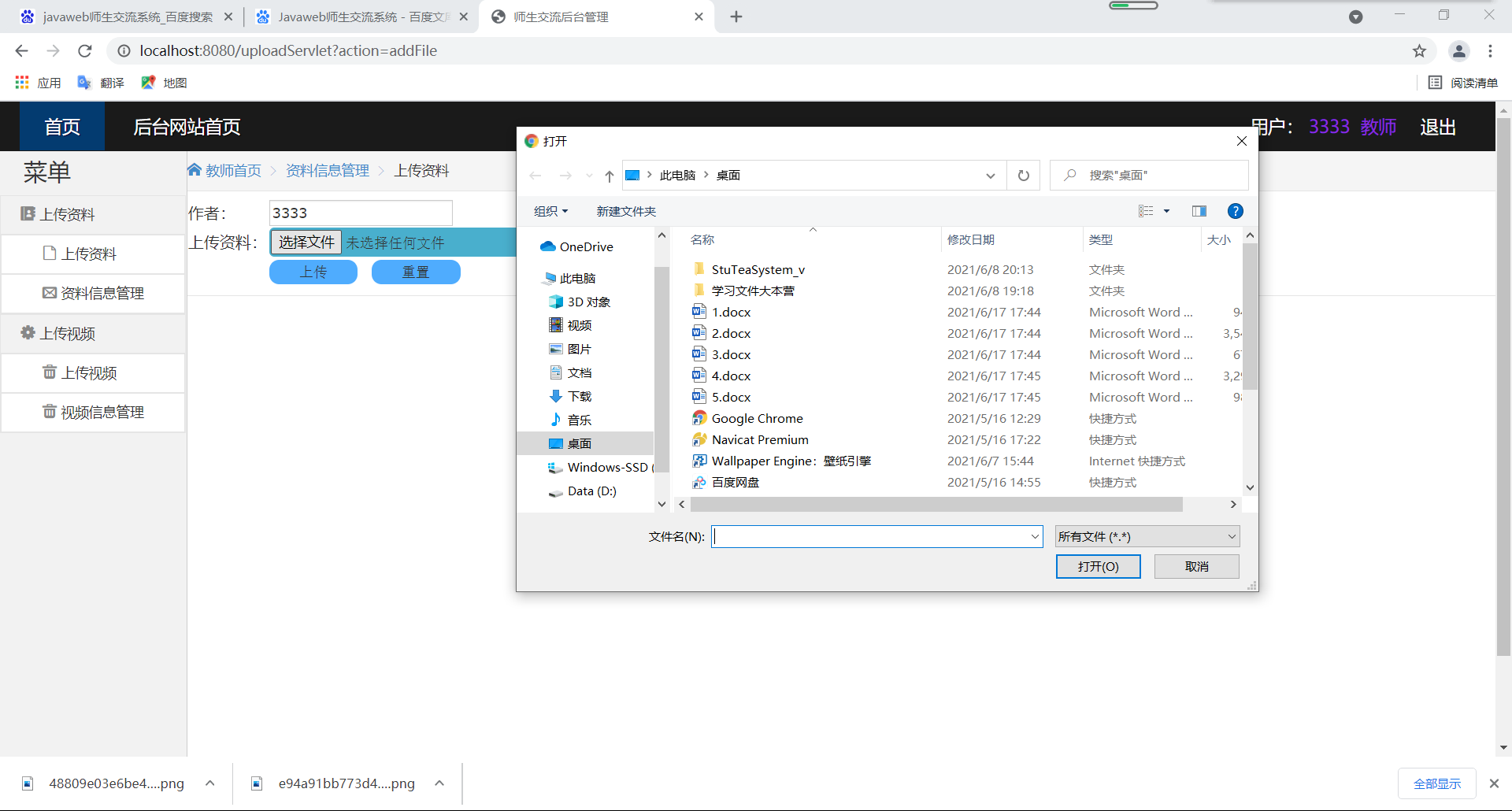


图十五添加学生信息成功界面



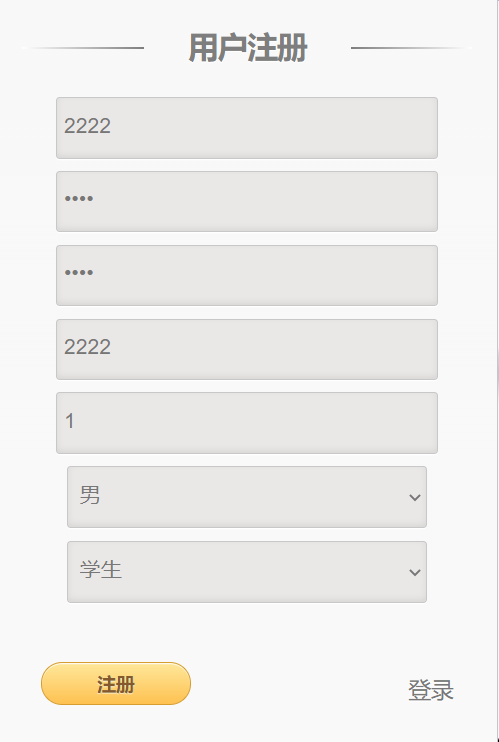
图十六删除学生信息成功界面

## 6.9上传测试



图十六上传界面

## 6.10注册页面测试



# 七、调试过程

## 7.1遇到的问题

1.登录时输入错误，不提示登录失败，而是网页直接报错。

2.提交表单时出现乱码。

3.rs.next()不能正确获取数据库中的数据。

4.不知道什么时候使用post提交，什么时候使用get提交。

## 7.2问题的解决

（1）登录判断页面（login.jsp）中，登录信息session的保存没有放在判断是否为空的判断语句（if（list！=null））中，导致登录失败也会保存session，这时就无法正确获取登录信息，导致网页报错。将session放在判断语句中就可以解决。

（2）表单提交时以get的请求发送到tomcat服务器后又会以默认的（ISO8859-1）解码，这时会出现中文乱码，改为post提交方式就解决了。

（3）ResultSet是一个结果集，即ResultSet存有一个表（通过执行查询数据库的语句生成），该表的当前行可以访问（当前行的初始位置是null），可以使用rs.next()方法移动到下一行（即表的第一行），如果新的当前行有效（即有值）就返回true，然后就可以使用get方法获取该行各列的值了，如果不存在下一行，则返回 false；因为这个方法会抛出一个SQLException，故要捕捉（try-catch）。直接使用rs.next()不能获取，前面要加上判断语句（if(rs.next)或while(rs.next)）。

（4）get方法传输数据会在地址栏显示，并且数据不能过长，但会优先读取本地缓存，很多时候会加速用户访问。一般获取数据用get，进行数据更改用post。

# 八、小结

收获和感受：

本次课设我的收获就是自己写代码的能力有了很大的提升，以前无从下手的问题现在都能游刃有余的解决。自我感觉最大的收获就是培养了自己对写代码的兴趣，每次解决一个问题的时候内心的喜悦和满足感难以言表，特别是在经历过无数次的修改，不明白为什么对，为什么不对最后改对时。

本次课设还让我明白，自己的专业知识还比较欠缺，面对很多问题自己还不能独立解决，书本上的知识已经不能完全满足自己，自己需要学习的还有很多，学习是一个积累的过程，需要自己持之以恒的坚持。

# 九、参考文献

[1] 郭克华. JavaWeb程序设计第三版.清华大学出版社  
[2] 王珊，萨师煊.数据库系统概论（第5版）.高等教育出版社  
[3] Cay S. Horstmann / Gary Cornell. JAVA核心技术（卷1） 机械工业出版社  
[4] [美] Bruce Eckel. Java编程思想 [第4版] 机械工业出版社  
[5] [美] Cay S. Horstmann/ Gary Cornell  JAVA核心技术（卷1）机械工业出版社  
[6]《Java 实时编程》  
[7]《Java加密与解密的艺术》  
[8]《Eclipse插件开发(原书第3版)(畅销插件开发指南新版)》  
[9]《Java语言程序设计:进阶篇(原书第8版)》  
[10]《Maven实战》