**沈 阳 工 学 院**

**BS系统开发实训报告**

**题 目： 院校资源库系统的设计**

**学 院： 信息与控制学院**

**专 业： 计算机科学与技术**

**学 号： 1530110318**

**学生姓名： 周天祥**

**指导教师： 郭鲁**

**成 绩：**

**2017年12 月29日**

**任 务 书**

|  |
| --- |
| 题目：院校资源库系统的设计 |
| 设计内容及要求：  1.学生必须仔细阅读实训任务书，根据要求完成程序设计，有问题及时主动通过各种方式与教师联系沟通。  2.学生要发挥自主学习的能力，充分利用时间，安排好设计的时间计划，并在设计过程中不断检测自己的计划完成情况，及时向教师展示软件设计情况。  3.设计按照教学要求需要1周时间完成，需要20小时的上机来调试设计的程序。  4. 功能要求：要求利用MVC设计模式开发程序。 |
| 设计环境：  MyEclipse、Tomcat、MySQL（数据库可自选） |
| 实现目标：  最终完成相关网站或应用程序的开发，以及报告的书写。 |

# 1 系统分析与设计

## 需求分析

随着国家教育改革的不断深入，广大的高中应届毕业生对于志愿院校和专业的选择也越来越多，而如何了解并且确定自己的目标学校和想要修读的专业也逐渐成为广大学子的一个困扰。以往的传统手段只能是学生或家长通过上网一所学校一所学校的搜，一个专业一个专业的找并且以手写或电子版的形式记录下来，随后自己分析比对统计得来的数据选择一个最优的答案，但是这种方法不仅费时费力也容易出现纰漏极有可能让自己与自己心仪的院校和专业擦肩而过。

同时，现在也是“互联网+”和大数据的时代，在填报志愿时手里多一份资料，多一份数据就离实现梦想更近了一步。再依靠传统的信息搜集办法不仅耽误学子的填报也会因为人为因素造成最终的录取结果不如自己所愿。院校资源库则很好的解决了这个痛点，它搜集全网最权威最新的院校信息数据，统计介绍专业信息的进阶和学习路线让用户知道自己所想学的知识是否符合自己并且介绍毕业后可以从事的工作让广大学子不再迷茫。同时，院校资源库为广大学生家长、学子提供了和院校的交流接口，有疑问可以直接拨打相关院校招生办老师的电话了解相关情况，让应届毕业的学生在填报志愿时有更多的信息依靠。

院校资源库不仅包括了院校的相关信息也包含了专业的相关信息，搜集统计网络上的专业信息描述，并且罗列出开设该专业的相关院校让用户在浏览相关专业的同时可以看到开设该专业的相关院校方便用户选择。院校资源库在为广大莘莘学子服务的同时也为高校提供了一个面向社会透明的平台和直接与考生沟通的渠道，方便院校方面能招收到优质的生源也有利于考生和家长在填报志愿时多一个选择。

院校资源库同时将学校以学校类型分类汇总，这样对于那些对自己学力有一定程度认知和评估的考生更加迅速精准的定位报考区间，省时省力而且可以在选择冲高和保底的院校时多一份准备和选择。而且考虑到有的学子可能会选择去国外留学深造，院校资源库刻意添加了部分外国大学的资料并整理分类防止考生和家长上当受骗同时为其提供参考。

## 1.2 系统功能设计

### 1.2.1 前台

1. 前台登录

用户登录功能，根据用户输入的账号和密码来验证用户是否已经存在，若存在则登录系统，若失败则保留登录界面不动。

2. 专业信息浏览

该部分显示全部专业信息，如果用户对某一专业感兴趣可以点击查看进入该专业的专业信息描述页面。

3. 院校信息浏览

该部分显示全部院校信息，用户可以根据需求浏览自己心仪的院校同时点击查看即可以查看院校的详细信息。

4. 院校类型信息浏览

该部分显示全部院校类型信息，用户可以根据自己的学力水平挑选适合的院校同时可以查看该类型下的全部院校信息方便用户精准定位自己心仪的院校。

5. 前台注册

如果用户是第一次使用该系统则无法直接通过登录页面进入系统中，用户填写注册页面的相关信息后直接跳转至系统中。前台注册默认用户权限为普通用户。

### 1.2.2 后台

1. 后台登录

后台为管理员权限下进行操作的系统，提供后台登录功能让管理员进入系统对系统和系统中的信息进行维护和更新。

2. 专业信息管理

专业信息管理包含查看全部专业信息，添加专业信息，修改专业信息和删除专业信息等功能。其中，如果删除或修改专业信息会对院校管理造成一定的影响。

3. 院校信息管理

院校信息管理包含查看全部院校信息，添加院校信息，修改院校信息和删除院校信息等功能。其中，如果删除或修改院校信息会对专业信息管理和院校类型信息管理造成一定程度的影响。

4. 院校类型信息管理

院校类型信息管理包含查看全部院校类型信息，添加院校类型信息，修改院校类型信息和删除院校类型信息等功能。其中，如果修改或删除院校类型信息会对院校类型信息管理造成一定的影响。

院校资源库系统的功能模块图如图1.1所示。

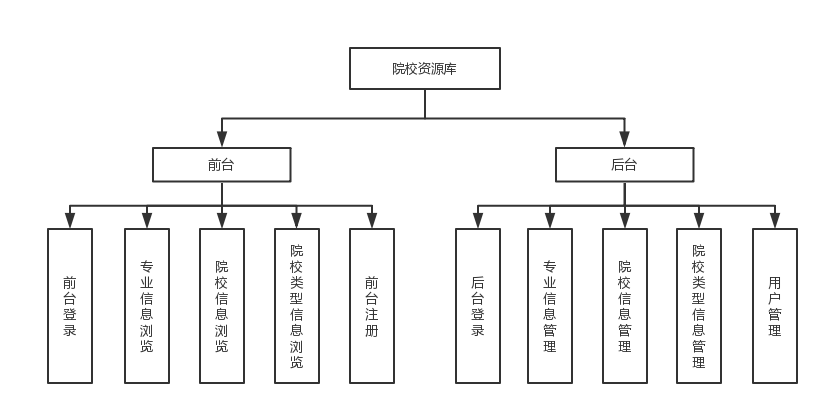


图1.1 功能模块图

# 2 系统实现

## 2.1 系统框架

系统框架利用MVC+DAO的设计模式，分为dbc包，vo包，dao包，impl包，factory包和Servlet包。其中dbc包负责连接数据库，vo包封装于数据库对应的表的字段，dao包包含对数据库进行操作的接口，impl包实现dao包中对应的方法，factory包含工厂类，降低程序耦合性，Servlet包用来相应客户端的请求。系统框架图如图2.1所示。

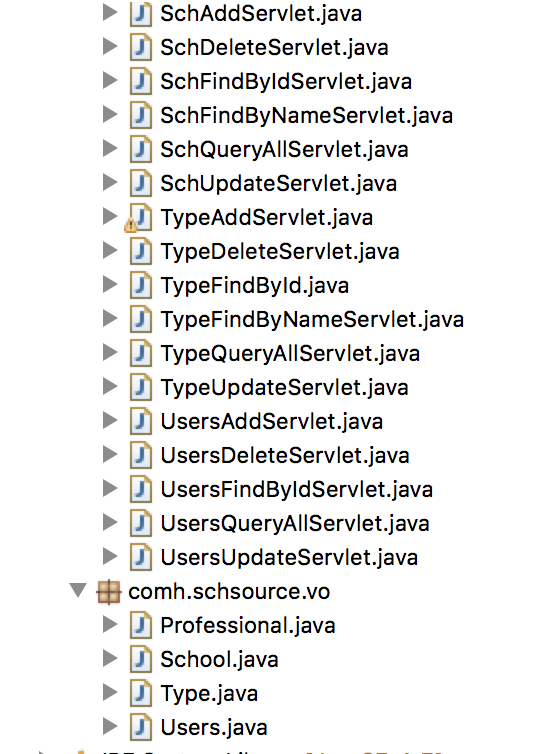
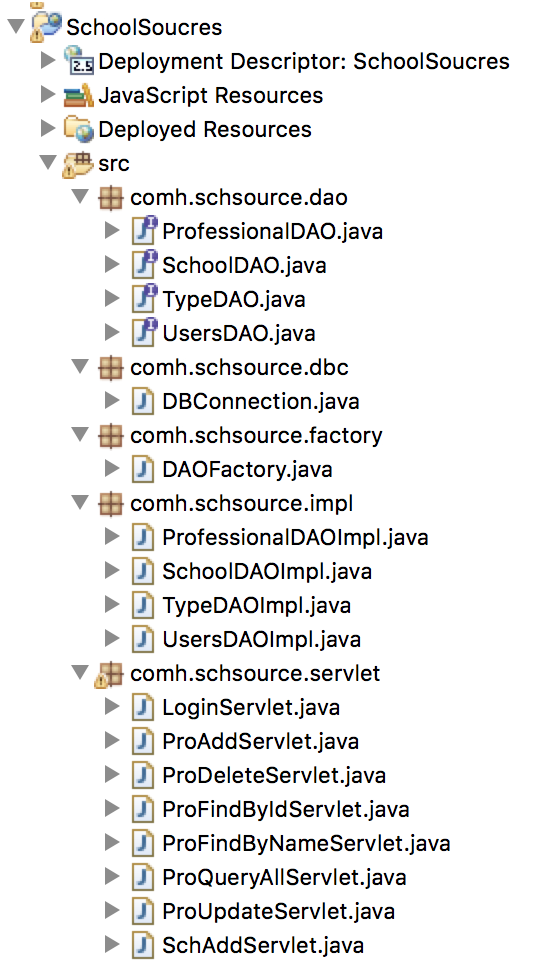


图2.1系统框架图

comh.schsource.dbc包为数据库连接用，comh.schsource.dbc包包含的文件介绍如表2.1所示。

表2.1 comh.schsource.dbc包文件介绍

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 作用 |
| comh.schsource.dbc | DBConnection | 连接数据库 |

comh.schsource.dao包包含专业，院校，院校类型和用户的接口，comh.schsource.dao包包含的文件介绍如表2.2所示

表2.2 comh.schsource.dao包文件介绍

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 作用 |
| comh.schsource.dao | ProfessionalDAO | 专业类操作方法接口 |
|  | SchoolDAO | 院校类操作方法接口 |
|  | TypeDAO | 院校类型类操作方法接口 |
|  | UsersDAO | 用户类操作方法接口 |

comh.schsource.factory包中包含工厂类，使用工厂的设计模式可以有效降低程序的耦合性，comh.schsource.factory包包含的文件介绍如表2.3所示。

表2.3 comh.schsource.factory包文件介绍

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 作用 |
| comh.schsource.factory | DAOFactory | 降低程序耦合性 |

comh.schsource.impl包中包含了之前由comh.schsource.dao包中接口定义的方法的具体实现类。comh.schsource.impl包包含文件介绍如表2.4所示。

表2.4 comh.schsource.impl包文件介绍

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 作用 |
| comh.schsource.impl | ProfessionalDAOImpl | 专业信息实现类 |
|  | SchoolDAOImpl | 院校信息实现类 |
|  | TypeDAOImpl | 院校类型信息实现类 |
|  | UsersDAOImpl | 用户信息实现类 |

comh.schource.servlet包包含了所有响应客户端请求的java类，调用之前封装定义好的方法，接口和实现类来执行相关操作。comh.schsource.servlet包包含文件简介如表2.5所示。

表2.5 comh.schsource.servlet包文件介绍

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 作用 |
| comh.schsource.servlet | LoginServlet | 登录事件响应 |
|  | ProAddServlet | 专业信息添加事件响应 |
|  | ProDeleteServlet | 专业信息删除事件响应 |

续表2.5 comh.schsource.servlet包文件介绍

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 作用 |
| comh.schsource.servlet | ProFindByIdServlet | 根据id查询专业信息事件响应 |
|  | ProFindByNameServlet | 根据name查询专业信息事件响应 |
|  | ProQueryAllServlet | 查询全部专业信息事件响应 |
|  | ProUpdateServlet | 修改专业信息事件响应 |
|  | SchAddServlet | 添加院校信息事件响应 |
|  | SchDeleteServlet | 删除院校信息事件响应 |
|  | SchFindByIdServlet | 根据id查询院校信息事件响应 |
|  | SchFindByNameServlet | 根据name查询院校信息事件响应 |
|  | SchQueryAllServlet | 查询全部院校信息事件响应 |
|  | SchUpdateServlet | 修改院校信息事件响应 |
|  | TypeAddServlet | 添加院校类型信息事件响应 |
|  | TypeDeleteServlet | 删除院校类型信息事件响应 |
|  | TypeFindByIdServlet | 根据id查询院校类型信息事件响应 |
|  | TypeFindByNameServlet | 根据name查询院校类型信息事件响应 |
|  | TypeQueryAllServlet | 查询全部院校类型信息事件响应 |
|  | TypeUpdateServlet | 修改院校类型信息事件响应 |
|  | UserAddServlet | 添加用户信息事件响应 |
|  | UsersDeleteServlet | 删除用户信息事件响应 |
|  | UsersFindByIdServlet | 用id查询用户信息事件响应 |
|  | UsersQueryAllServlet | 查询全部用户信息事件响应 |
|  | UsersUpdateServlet | 修改用户信息事件响应 |

comh.schsource.vo包中封装了所有基本数据的和其数据类型，防止有人在外部修改数据，保证程序的安全性。comh.schsource.vo包包含文件介绍如表2.6所示。

表2.6 comh.schsource.vo包文件介绍

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 作用 |
| comh.schsource.vo | Professional | 封装professional表中对应的字段 |
|  | School | 封装school表中对应的字段 |
|  | Type | 封装type表中对应的字段 |
|  | Users | 封装users表中对应的字段 |

专业表如表2.7所示。

表2.7 专业表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 是否为空 | 说明 |
| PID | int | NOT NULL | 专业编号(主键) |
| PNAME | varchar(50) | NOT NULL | 专业名称 |
| PDES | varchar(350) | NOT NULL | 专业描述 |

院校表如表2.8所示。

表2.8 院校表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 是否为空 | 说明 |
| SchID | int | NOT NULL | 院校编号(主键) |
| SchName | varchar(50) | NOT NULL | 院校名称 |
| PID | int | NOT NULL | 专业编号 |
| TID | int | NOT NULL | 类型编号 |
| SchADDRESS | varchar(50) | NOT NULL | 学校地址 |
| TEACHER | varchar(50) | NOT NULL | 联系教师 |
| TEL | varchar(50) | NOT NULL | 联系电话 |

院校类型表如表2.9所示。

表2.9 院校类型表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 是否为空 | 说明 |
| TID | int | NOT NULL | 院校类型编号(主键) |
| TNAME | varchar(50) | NOT NULL | 院校类型名称 |

用户表如表2.10所示。

表2.10 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 是否为空 | 说明 |
| UsersID | int | NOT NULL | 用户ID(主键) |
| UPWD | varchar(20) | NOT NULL | 用户密码 |
| UNAME | varchar(50) | NOT NULL | 用户名称 |
| UGENDER | varchar(20) | NOT NULL | 用户性别 |
| ULIMIT | int | NOT NULL | 用户权限 |

## 2.2 前台登录注册模块的实现

用户要想进入系统，进行浏览等操作，需要先登录，否则无法进入系统，有账号的用户即可在登录界面登录即可，没有账号的用户需要在注册界面注册后方可登录。登录注册界面如图2.2所示。



图 2.2 前台登录注册界面

功能描述：

用户在首页点击“登录”按钮，即可进入登录界面，用户输入正确的用户名和密码后，点击“登录”按钮。在服务器后台，提交表单，执行LoginServlet，获取用户输入的表单信息，接着查询数据库中的users表，查看是否存在该用户，如果不存在则提示错误信息，如果存在则返回该用户信息，并将该用户存在session域中，接着跳转至首界面。用户登录成功后，在首界面即可显示出用户的用户名信息。

当用户没有账号时，点击“注册”按钮，即可进入注册界面，用户输入正确的个人信息，点击“注册”按钮。在服务器后台，提交注册表单，执行UsersAddServlet，获取用户输入的表单信息，赋值至users对象中，传入到regist()方法，执行数据库的添加操作，注册失败则有提示错误信息，注册成功用户即可登录并直接跳转至主页面。

## 2.3 专业信息浏览模块的实现

用户登录成功后即可进入到院校资源库的主页，里面可以按自己的需求浏览信息点击查看全部专业信息即可浏览专业信息。专业信息浏览界面如图2.3所示。

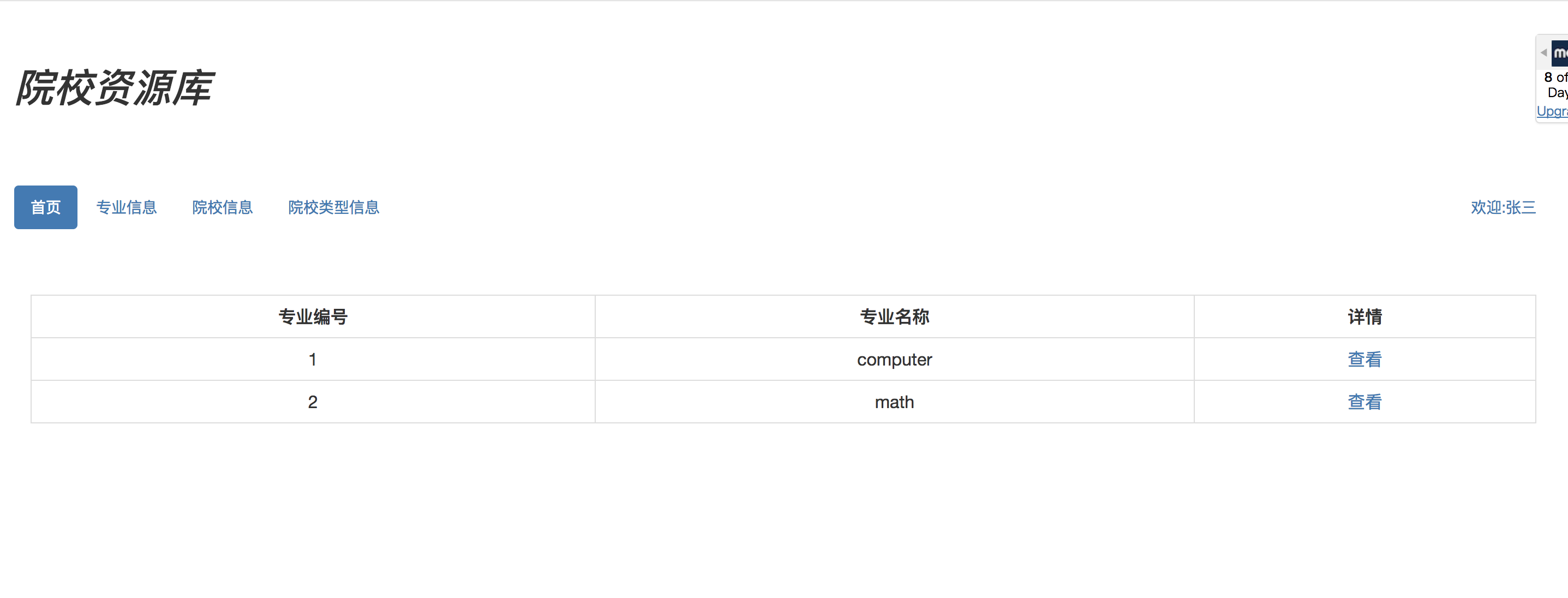


图2.3 专业信息浏览界面

功能描述：

当用户登录后，点击专业信息，服务器后台即刻执行ProQueryAllServlet中的findAllPro()方法，获取到数据库中所有的专业信息存到request域中，并跳转至pro\_queryall.jsp文件中将所有专业信息全部显示出来。

## 2.4 院校信息浏览模块的实现

用户登录成功后即可进入到院校资源库的主页，里面可以按自己的需求浏览信息点击查看院校信息即可浏览院校信息。院校信息浏览界面如图2.4所示。



图2.4 院校信息浏览界面

功能描述：

当用户登录后，点击院校信息，服务器后台即刻执行SchQueryAllServlet中的findAllSch()方法，获取到数据库中所有的院校信息存到request域中，并跳转至sch\_queryall.jsp文件中将所有院校信息全部显示出来。

## 2.5 院校类型信息浏览模块的实现

用户登录成功后即可进入到院校资源库的主页，里面可以按自己的需求浏览信息点击查看院校类型信息即可浏览院校信息。院校信息类型浏览界面如图2.5所示。

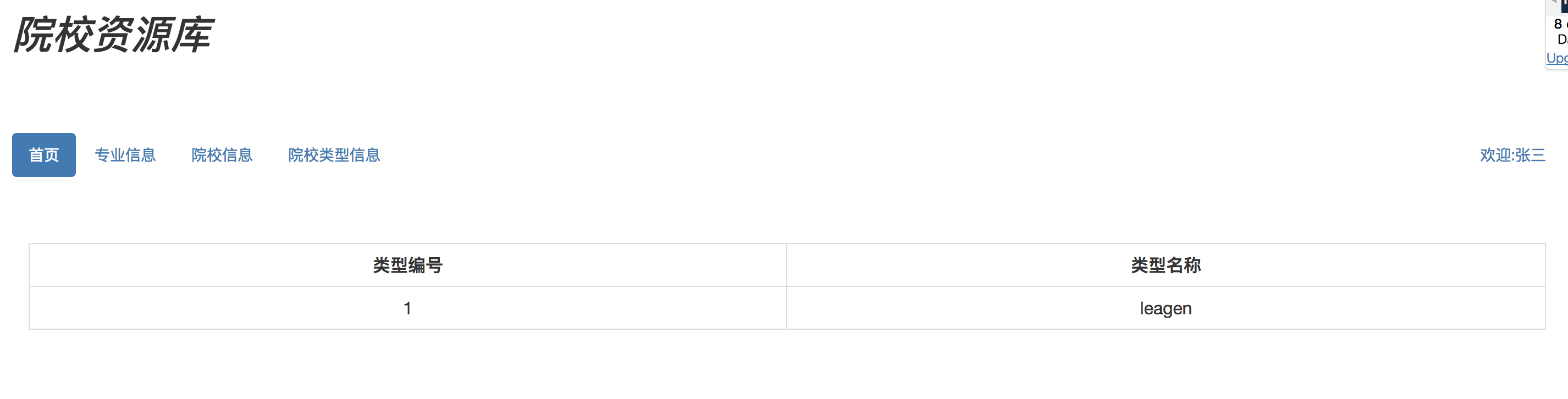


图2.5 院校类型浏览界面

功能描述：

当用户登录后，点击院校类型信息，服务器后台即刻执行TypeQueryAllServlet中的findAllType()方法，获取到数据库中所有的院校类型信息存到request域中，并跳转至type\_queryall.jsp文件中将所有院校类型信息全部显示出来。

## 2.6 后台登录模块的实现

管理员要想进入系统，进行浏览等操作，需要先登录，否则无法进入系统，由于管理员权限无法通过注册取得，所以没有注册页面。后台登录界面如图2.6所示。



图 2.6 前台登录注册界面

功能描述：

用户在首页点击“登录”按钮，即可进入登录界面，用户输入正确的用户名和密码后，点击“登录”按钮。在服务器后台，提交表单，执行LoginServlet，获取用户输入的表单信息，接着查询数据库中的users表，查看是否存在该用户并获取该用户权限，如果不存在或权限不是管理员则提示错误信息或进入前台首页，如果存在则返回该用户信息，并将该用户存在session域中，接着跳转至后台首页。用户登录成功后，在首界面即可显示出用户的用户名信息。

## 2.7 专业信息管理模块的实现

管理员点击“专业信息添加”、“专业信息管理”后，即可添加专业信息和查看当前已有的专业信息，并可以对专业信息执行相关操作。专业信息管理界面如图2.7所示。



图 2.7 专业信息管理界面

功能描述：

管理员点击“专业信息添加”，即可添加想要添加的专业信息，填写表单后，点击“添加”按钮，服务器后台提交表单至TypeAddServlet，执行addPro()方法，获取表单信息，执行实现类的addPro()方法，执行数据库语句，将专业信息添加至数据库中。后台成功添加专业信息后，前台界面的专业信息中也会对应显示出来。

管理员点击“专业信息管理”，即可查看数据库中已有的全部专业信息，执行TypeQueryAllServlett的findallPro()方法，并将数据库中信息保存至request域中，跳转至type\_queryall.jsp界面，查看全部。管理员想要对其中一个专业信息想要修改时，点击对应的超链接执行TypeFindByIdServlet中的findProById()方法通过获取的专业编号，执行数据库语句，查询到对应的专业信息，将其放到type\_updaye.jsp中，输入修改按钮后执行TypeUpdateServlet中的updatePro()方法，执行数据库语句，修改对应的专业信息；管理员想要对其中一个专业信息想要删除时，点击删除链接，执行TypeDeleteServlet中的deletePro()方法，通过获取的专业信息编号，执行数据库语句，即可删除专业信息。

## 2.8 院校信息管理模块的实现

管理员点击“院校信息添加”、“院校信息管理”后，即可添加院校信息和查看当前已有的院校信息，并可以对院校信息执行相关操作。院校信息管理界面如图2.8所示。

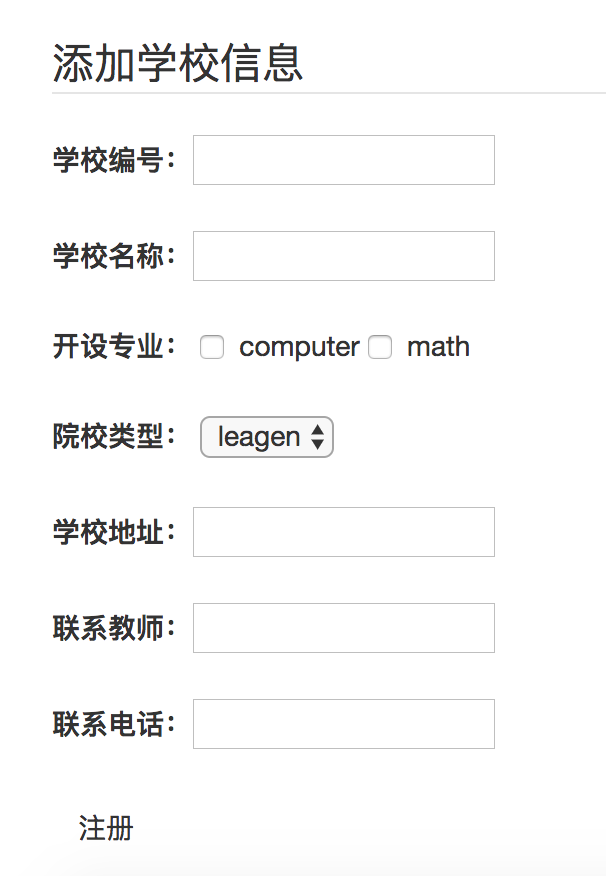


图 2.8 院校信息管理界面

功能描述：

管理员点击“院校信息添加”，即可添加想要添加的院校信息，填写表单后，点击“添加”按钮，服务器后台提交表单至SchAddServlet，执行addSch()方法，获取表单信息，执行实现类的addSch()方法，执行数据库语句，将院校信息添加至数据库中。后台成功添加院校信息后，前台界面的院校信息中也会对应显示出来。

管理员点击“院校信息管理”，即可查看数据库中已有的全部院校信息，执行Sch QueryAllServlett的findallSch ()方法，并将数据库中信息保存至request域中，跳转至sch\_queryall.jsp界面，查看全部。管理员想要对其中一个院校信息想要修改时，点击对应的超链接执行SchFindByIdServlet中的findSchById()方法通过获取的院校编号，执行数据库语句，查询到对应的院校信息，将其放到sch\_updaye.jsp中，输入修改信息点击按钮后执行SchUpdateServlet中的updateSch ()方法执行数据库语句，修改对应的院校信息；管理员想要对其中一个院校信息删除时，点击删除链接执行SchDeleteServl et中的deleteSch ()方法，通过获取的院校信息编号，执行数据库语句，即可删除院校信息。

## 2.9 院校类型信息管理模块的实现

管理员点击“院校类型信息添加”、“院校类型信息管理”后，即可添加院校类型信息和查看当前已有的院校类型信息，并可以对院校类型信息执行相关操作。院校类型信息管理界面如图2.9所示。

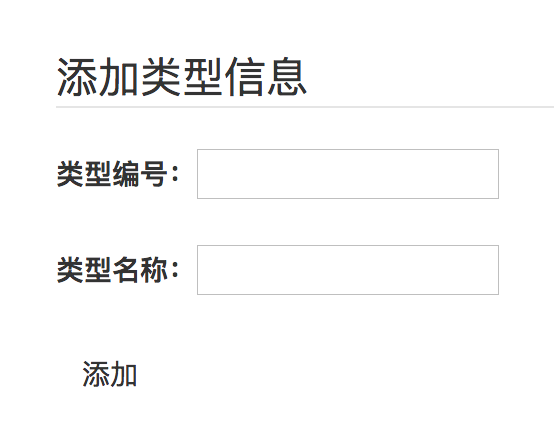


图 2.9 院校类型信息管理界面

功能描述：

管理员点击“院校类型信息添加”，即可添加想要添加的院校类型信息，填写表单后，点击“添加”按钮，服务器后台提交表单至TypeAddServlet，执行addType ()方法，获取表单信息，执行实现类的addType ()方法，执行数据库语句，将院校类型信息添加至数据库中。后台成功添加院校类型信息后，前台界面的院校类型信息中也会对应显示出来。

管理员点击“院校类型信息管理”，即可查看数据库中已有的全部院校类型信息，执行TypeQueryAllServlett的findallType ()方法，并将数据库中信息保存至request域中，跳转至type\_queryall.jsp界面，查看全部。管理员想要对其中一个院校类型信息想要修改时，点击对应的超链接执行TypeFindByIdServlet中的findTypeById()方法通过获取的院校类型编号，执行数据库语句，查询到对应的院校类型信息，将其放到type\_updaye.jsp中，输入修改信息点击按钮后执行TypeUpdateServlet中的updateType ()方法执行数据库语句，修改对应的院校类型信息；管理员想要对其中一个院校类型信息删除时，点击删除链接执行TypeDeleteServlet中的deleteType ()方法，通过获取的院校类型信息编号，执行数据库语句，即可删除院校类型信息。

## 2.10 用户信息管理模块的实现

管理员点击“用户信息添加”、“用户信息管理”后，即可添加用户信息和查看当前已有的用户信息，并可以对用户信息执行相关操作。用户信息管理界面如图2.10所示。

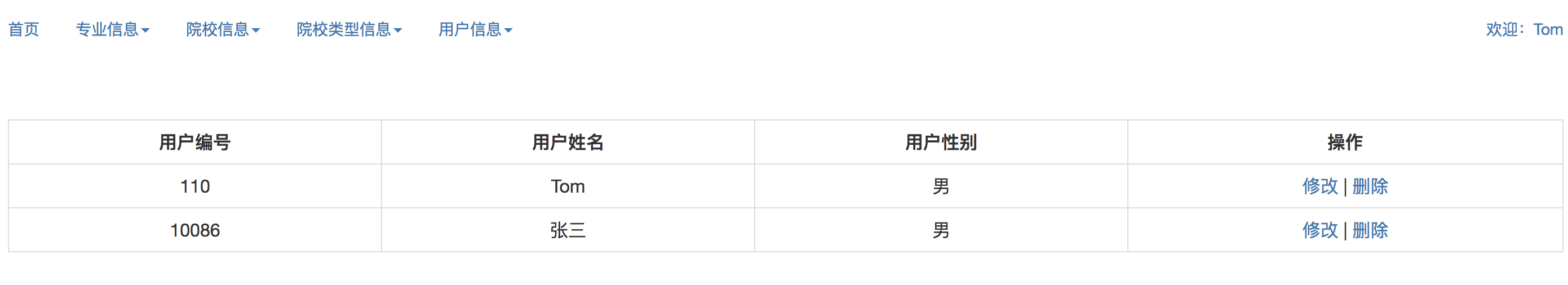


图 2.10 院校类型信息管理界面

功能描述：

管理员点击“用户信息添加”，即可添加想要添加的用户信息，填写表单后，点击“添加”按钮，服务器后台提交表单至UsersAddServlet，执行regist ()方法，获取表单信息，执行实现类的regist ()方法，执行数据库语句，将用户信息添加至数据库中。后台成功添加用户信息后，前台界面的用户信息中也会对应显示出来。

管理员点击“用户信息管理”，即可查看数据库中已有的全部用户信息，执行UsersQueryAllServlett的findallUsers ()方法，并将数据库中信息保存至request域中，跳转至users\_queryall.jsp界面，查看全部。管理员想要对其中一个用户信息想要修改时，点击对应的超链接执行UsersFindByIdServlet中的findUsersById()方法通过获取的用户编号，执行数据库语句，查询到对应的用户信息，将其放到users\_update.jsp中，输入修改信息按钮后执行UsersUpdateServlet中的updateUsers ()方法执行数据库语句，修改对应的用户信息；管理员想要对其中一个用户信息删除时，点击删除链接执行UsersDeleteServlet中的deleteUsers ()方法，通过获取的用户信息编号，执行数据库语句，即可删除用户信息。

# 总 结

院校资源库系统前台包含了用户登录注册，专业信息浏览，院校信息浏览和院校类型信息浏览，后台包含了后台登录，专业信息管理，院校信息管理，院校类型信息管理和用户管理。对系统中的用户进行了权限区分，同时环环相扣，对一种数据的操作很可能会导致另一种数据的变动。

待改进或不足之处：

1. 本系统功能较少，用法单一，虽然保留了日后可以此为基础大量扩展功能的条件，但就目前而言功能仍然较为单一和缺乏。
2. 系统中对于信息的查看只能选择查看全部这一种方式，应当提供一个搜索功能方便用户直接定位想要的查看的信息。

遇到的问题：

1. 由于开发中对于变量的命名不规范，经常容易造成java.lang.NullPointerException错误，后确定命名规范，格式名字统一，前端后台名字一致即可避免这种较为低级的错误发生。
2. 在进行院校信息添加功能开发时，出现了在添加后查看全部院校信息时院校类型信息和开设专业信息两者位置相反，定位问题发生在实现类的添加院校信息的sql语句时，二者赋值位置错误，造成错位，修改后问题解决
3. 运行程序时，通过jsp页面提交表单，跳转至servlet页面时，报错404。解决方法：查看web.xml文件中的servlet映射，查看url-pattern中的值，修改其中的值为自己的url地址即可解决。
4. 服务器启动时报错。原因：端口被占用。解决方法：打开控制面板，选择服务，关闭Apache Tomcat服务，重启Tomcat即可成功。
5. 期初数据库选用的是安装在虚拟机上的SQL Server2012数据库，但是连接会很慢而且容易造成断开连接导致程序出错，后将数据库转移到本机的MySQL数据库问题即得到解决，断定为虚拟机和物理机的兼容问题造成此问题。

本次实训对javaweb有了更深层次的理解，本次实训收获很多，对request，session，application有了更深入的理解。对EL表达式的使用有了新的认识，对乱码问题的解决可以设置全局的字符编码。也了解了一些开源的连接池，例如DBCP，c3p0连接池。接触了一些前端的如bootstrap框架和响应式布局等知识并将其运用到项目中来。