目录

[1. 背景简介 2](#_Toc43582118)

[1.1. 项目背景 2](#_Toc43582119)

[2. 可行性分析 2](#_Toc43582120)

[2.1. 经济可行性 2](#_Toc43582121)

[2.2. 社会可行性 3](#_Toc43582122)

[2.3. 技术可行性 3](#_Toc43582123)

[3. 需求分析 4](#_Toc43582124)

[3.1 学生需求分析 4](#_Toc43582125)

[3.2 导师需求分析 5](#_Toc43582126)

[3.3 学院级别管理员需求分析 5](#_Toc43582127)

[3.4 研究室管理员需求分析 6](#_Toc43582128)

[4. 系统平台及软件 6](#_Toc43582129)

[4.1 开发硬件操作系统 6](#_Toc43582130)

[4.2 IDE 7](#_Toc43582131)

[4.3 软件平台 7](#_Toc43582132)

[5. 用例图 7](#_Toc43582133)

[4.1 参与者 7](#_Toc43582134)

[4.2 用例图 8](#_Toc43582135)

[4.2.1 学生用例图 8](#_Toc43582136)

[4.2.2 教师用例图 8](#_Toc43582137)

[4.2.3 学院管理员用例图 9](#_Toc43582138)

[4.2.4 研究室管理员用例图 9](#_Toc43582139)

[6. 包图 10](#_Toc43582140)

[6.1 详细包图 10](#_Toc43582141)

[7. 数据库图表 10](#_Toc43582142)

[7.1. ER图 10](#_Toc43582143)

[7.2 数据库关系表 13](#_Toc43582144)

[7.3 数据字典 14](#_Toc43582145)

[9．系统功能介绍 15](#_Toc43582146)

[10. 核心模块介绍 29](#_Toc43582147)

[10.1 登录模块 29](#_Toc43582148)

[10.2 上传模块 30](#_Toc43582149)

[11.附录(源代码, 见压缩包) 30](#_Toc43582150)

## 背景简介

#### 1.1. 项目背景

1.1.1.项目背景

自互联网发展，关于研究生和导师双选的需求量越来越大，学校与在校研究生之间一直用邮件进行交互，无法有一个较好的平台提供进行双方合作。授予在校研究生和导师互相联络的限制，导致硕士无法正确与导师有一个好好的沟通平台，有时候会选错专业，误选专业，甚至造成研究生两年都从事一个不太喜欢的工作。

据《搜狐网》2018年06月14日《考研大数据，看完我默默哭了一会儿》文章所示，考研人群女生占比超过6成。从就读院校性质上看，非双一流高校占比超过8成，一方面根非“双一流”高校学生总量有关，另一方面，普通高校毕业生读研意愿更强。从数据上看，超过半数读研是为了学校背景出生。全国有近3成的考研学生是选择了读研转换专业，而在换专业时，选择了错误的导师不在少数。在知乎问答中，不少人在寻求考研选错导师，成为了研究生两年的遗憾。

在现实层面上，国内没有一个统一的平台能够提供研究生导师和学生进行双选的平台。导师与学生往往不能互相了解信息。常有的学生抱怨选错了导师，导师抱怨选错了学生，这样的结果造成了一年又一年的悲剧发生。

1.2.2项目概述

“研究生导师双选系统“的目标是开发出信息双选平台，涵盖校外准研究生和校内导师的各个信息情况，同时也包含了登录、导入、以及校内院管理、校内研究室管理的多权限的一个管理平台。市场分析定位在校内的研究生。

“研究生导师双选系统“通过多方面的收集资源和从校内向多方面推广，来实现初步的信息双选。采用HTML5、CSS3、JavaScript、JSP、Servlet、JSTL等技术，设计一款能满足PC端的一个响应式设计网页。

## 可行性分析

#### 经济可行性

人力成本：该项目从调研、分析、设计中可计算出人力成本，约元人月。

硬件软件成本：产品生产及上线后需要购买软件不需要购买，均为开源的。MySQL为开源数据库，java是一门开源的编程语言，jsp和jstl，html5、css3、JavaScript均无版权问题需要交费。而硬件上需要通过阿里云提供技术支持，提供一个centos平台进行单机部署，该产品会产生硬件成本，约元人月。开发工具用到了Jetbrains IDEA，在财务上可能会消耗1k人年，服务器数量需要通过实际情况进行调整。其他路由器网络成本均在可调节的预算范围内。

如：产品用到的数据库、开发工具、第三方软件、服务器数量、路由器、网络等成本。

#### 社会可行性

该项目遵守法律法规。

该产品的社会层面问题的解决：目的是为了解决学校的导师和学生之前的选课障碍，产品符合道德标准和大众审美，无负能量传播。社会影响力，通过产品的推广，产品将会传播到各大高校，增加产品的实用程度。

社会影响力的关键，通过产品的推广，产品会给公司带来一定的社会效益，在高校内外，使用该产品会带来更高的选课成功率，这些会给市场资源在高校的环境下带来一定的社会效益。

#### 技术可行性

该产品用到如下html5、css3、JavaScript、jsp、jstl、el等相关技术，无太多技术难题，软件成功率较大。

HTML称为超文本标记语言，是一种标识性的语言。它包括一系列标签．通过这些标签可以将网络上的文档格式统一，使分散的Internet资源连接为一个逻辑整体。HTML文本是由HTML命令组成的描述性文本，HTML命令可以说明文字，图形、动画、声音、表格、链接等。

CSS层叠样式表(英文全称：Cascading Style Sheets)是一种用来表现HTML（标准通用标记语言的一个应用）或XML（标准通用标记语言的一个子集）等文件样式的计算机语言。CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。 能够对网页中元素位置的排版进行像素级精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力。

JavaScript（简称“JS”） 是一种具有函数优先的轻量级，解释型或即时编译型的高级编程语言。虽然它是作为开发Web页面的脚本语言而出名的，但是它也被用到了很多非浏览器环境中，JavaScript 基于原型编程、多范式的动态脚本语言，并且支持面向对象、命令式和声明式（如函数式编程）风格。

JSP（全称JavaServer Pages）是由Sun Microsystems公司主导创建的一种动态网页技术标准。JSP部署于网络服务器上，可以响应客户端发送的请求，并根据请求内容动态地生成HTML、XML或其他格式文档的Web网页，然后返回给请求者。JSP技术以Java语言作为脚本语言，为用户的HTTP请求提供服务，并能与服务器上的其它Java程序共同处理复杂的业务需求。

JSTL（Java server pages standarded tag library，即JSP标准标签库）是由JCP（Java community Proces）所制定的标准规范，它主要提供给Java Web开发人员一个标准通用的标签库，并由Apache的Jakarta小组来维护。开发人员可以利用这些标签取代JSP页面上的Java代码，从而提高程序的可读性，降低程序的维护难度。

EL（Expression Language） 是为了使JSP写起来更加简单。表达式语言的灵感来自于 ECMAScript 和 XPath 表达式语言，它提供了在 JSP 中简化表达式的方法，让Jsp的代码更加简化。

该技术程序员的技术水平在一般即可，无较高难度的技术。在软件意义上，成功率较大，可制作成一个较为稳定的环境。

## 需求分析

系统功能是在实际开发设计过程中经过调研、分析用户需求，和用户一起共同确定下来，是系统满足用户需求所应完成的功能。本课程设计模拟一个研究生导师双选系统。研究生导师双选业务活动包括：三个角色的登录登录模块，研究生入学时需要选择自己的导师，导师也需要选择学生；管理员（各个学院相应专业）也需要管理学生，导师数据，最终确定学生导师。每个研究生选导师有三个志愿；导师所带学生上限由管理员设定；

#### 3.1 学生需求分析

学生人员包含学号、姓名、密码、所报考专业。功能模块包括登录功能、选择导师、查看导师的选择情况、最终导师情况、个人信息维护、密码修改等功能。

3.1.1 登录模块

本系统包含登录模块，登录功能包含4个角色，研究室管理员，学院管理员，教师和学生。

用户首页点击角色进入登录界面，进入登录界面后，进入学生端首页。

3.1.2 查看导师

登录成功可进行查看导师选项，查看各个专业导师（导师与自身专业无关，因为包含跨考学生，所以我们提供该学校的老师，供以学生选择。

3.1.3 选择导师

在查看导师页面可以选择3个导师，我们这个仅仅是志愿，志愿选择后可以对其进行修改，我们选择导师后，志愿可以对其进行功能性质的修改。

3.1.4 查看导师选择情况，最终导师

选择后，我们学生进行等待，等待导师的选择，导师选择完毕后，学生端可进行查看导师的选择情况，并对其进行回应。

3.1.5 个人信息维护

学生可以对自己的信息进行维护，不限于修改一些无关因素，不可修改名字，学号等关键字段。但是可以对其密码进行修改。

#### 3.2 导师需求分析

导师人员包含以下功能：登录，依据上限，选定学生（已经选择了该导师的学生）需要专业页面。

3.2.1 登录模块

教师端登录都进入一个教师端首页。

3.2.2 选择学生

教师端登录成功后可以看到学生的志愿，因为学生人数较多，所以我们显示的是已经选择（本导师）的学生，关于选择，这边导师查看完毕学生的个人资料后，可以做出相应的回应，是同意或者拒绝。

3.2.3 个人信息维护

教师也可以对自己的信息进行维护，不限于修改一下无关因素，不可修改名字，学号等关键字段。但是可以对其密码进行修改。

#### 3.3 学院级别管理员需求分析

学院级别管理员包含以下功能：登录、学生基本信息批量导入（单个录入），维护；导师数据批量导入（单个录入），维护；专业，设定导师上限；查看学生选择导师情况，导师选择学生情况，最终确定学生导师；（只限于管理本院的招生专业）导出志愿表，选择导师最终结果（只限于管理本院的招生专业）。

3.3.1 登录模块

学院级别管理员和研究室管理员公用一个登陆的界面，但是，登录跳转不一样，用户首页点击角色进入登录界面，进入登录界面后，进入管理端首页。

3.3.2 学生基本信息批量导入（单个维护）

管理员可以进行处理一个excel表格，进行填写学生信息，之后选择系统的批量导入功能，进行导入excel表格，对excel进行处理维护后，导入，直接导入数据库。也可以对其单个学生进行信息维护，对学生具有增加、删除、修改、查询功能。

3.3.3 教师基本信息批量导入（单个维护）

管理员可以进行处理一个excel表格，进行填写教师信息，之后选择系统的批量导入功能，进行导入excel表格，对excel进行处理维护后，导入，直接导入数据库。也可以对其单个教师进行信息维护，对学生具有增加、删除、修改、查询功能。

3.3.4 专业上限修改

专业会在导入教师信息的时候自动插入，但是管理员可以对专业人员进行把控，控制每个专业的人数，便于进行调试进行专业的信息修改。

研究室级别管理员包含以下功能：登录、学生基本信息批量导入（单个录入），维护；导师数据批量导入（单个录入），维护；专业，设定导师上限；查看学生选择导师情况，导师选择学生情况，最终确定学生导师；（只限于管理本院的招生专业）导出志愿表，选择导师最终结果（只限于管理本院的招生专业）。对学院级别管理员进行相关维护。

#### 3.4 研究室管理员需求分析

3.4.1 登录模块

学院级别管理员和研究室管理员公用一个登陆的界面，但是，登录跳转不一样，用户首页点击角色进入登录界面，进入登录界面后，进入管理端首页。

3.4.2 学生基本信息批量导入（单个维护）

管理员可以进行处理一个excel表格，进行填写学生信息，之后选择系统的批量导入功能，进行导入excel表格，对excel进行处理维护后，导入，直接导入数据库。也可以对其单个学生进行信息维护，对学生具有增加、删除、修改、查询功能。

3.4.3 教师基本信息批量导入（单个维护）

管理员可以进行处理一个excel表格，进行填写教师信息，之后选择系统的批量导入功能，进行导入excel表格，对excel进行处理维护后，导入，直接导入数据库。也可以对其单个教师进行信息维护，对学生具有增加、删除、修改、查询功能。

3.4.4 专业上限修改

专业会在导入教师信息的时候自动插入，但是管理员可以对专业人员进行把控，控制每个专业的人数，便于进行调试进行专业的信息修改。

3.4.5 学院级别管理员维护

学院级别管理员因为人数较少，可对其进行单独的增加修改，删除等功能。

## 系统平台及软件

#### 4.1 开发硬件操作系统

4.1.1 硬件平台：MacBook Pro

4.1.2 操作系统：MacOS Catalina v10.15.4

#### 4.2 IDE

4.2.1 Java开发：JetBrains IntelJ IDEA 2020.1

#### 4.3 软件平台

4.3.1 开发语言：Java、HTML、CSS、JS

4.3.2 后端框架：jsp、servlet

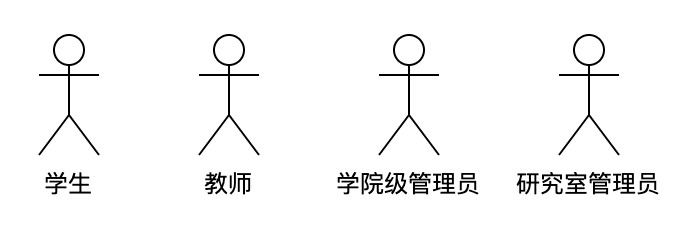
4.3.3 前端框架：bootstrap

4.3.4 服务应用：tomcat

4.3.5 数据库：mysql

## 用例图

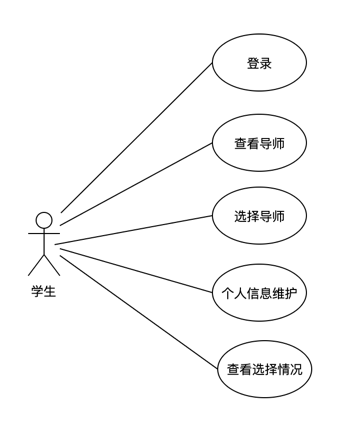
#### 4.1 参与者



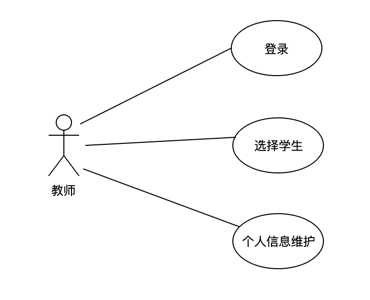
参与者分为：学生、教师、学院级管理员、研究室管理员

#### 4.2 用例图

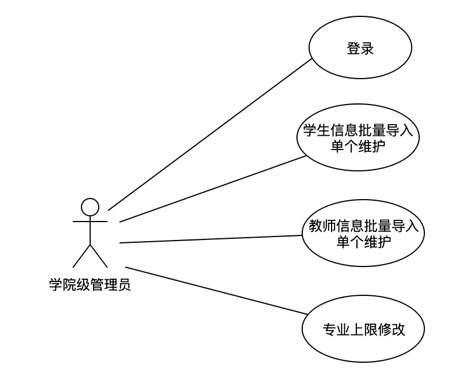
##### 4.2.1 学生用例图



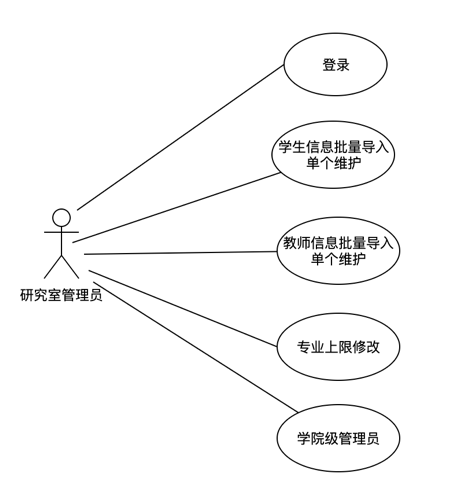
##### 4.2.2 教师用例图



##### 4.2.3 学院管理员用例图



##### 4.2.4 研究室管理员用例图



## 包图

#### 6.1 详细包图

该系统使用MVC的开发模式。开发包图如下。

com.littlepage.Entity：实体层，属于MVC的Model（模型），表示应用程序核心（比如数据库记录列表）。

com.littlepage.listener：easy excel的实体，里面有部分业务逻辑，用于读取和写入excel表的逻辑

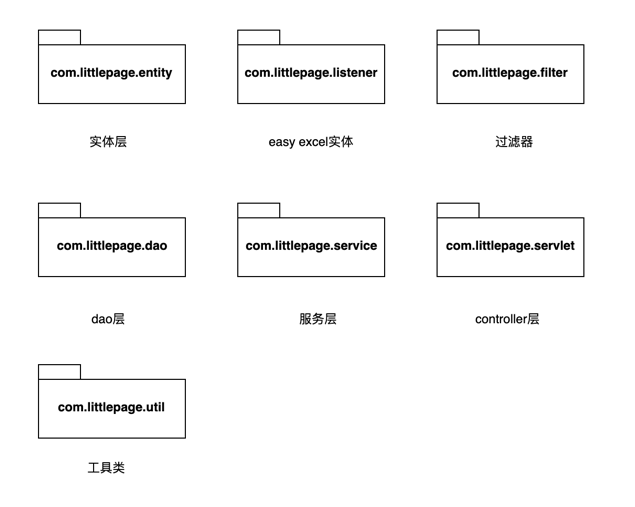
com.littlepage.filter：过滤器，用于权限管理和路由管理

com.littlepage.dao：数据库接口层，用于映射数据库的实体的一个关系表

com.littlepage.service：业务层，用于处理复杂的业务

com.littlepage.servlet：属于MVC的C，Controller层，处理输入

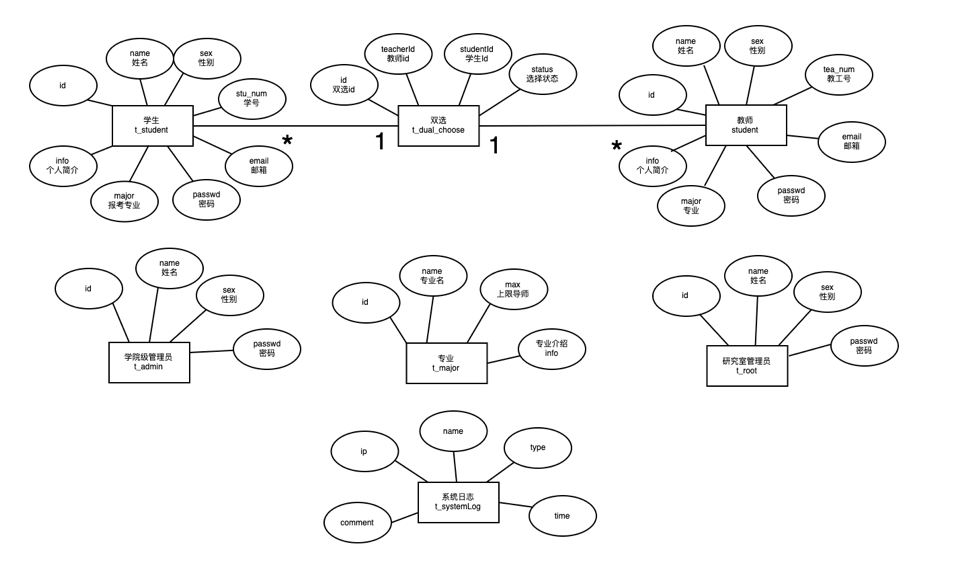
com.littlepage.util: 工具包，富有很多作为工具的文件



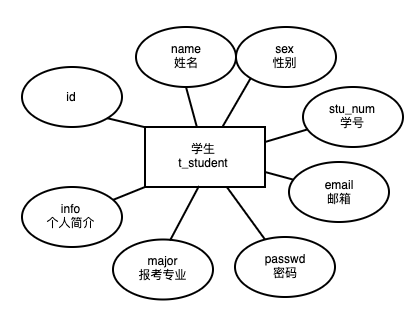
## 数据库图表

#### 7.1. ER图

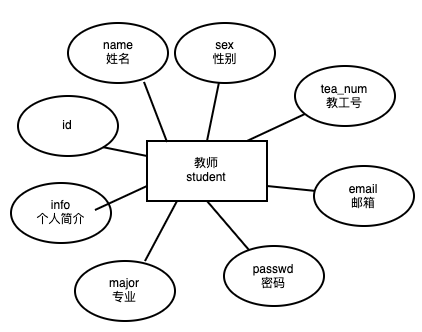
7.1.1 ER图概览



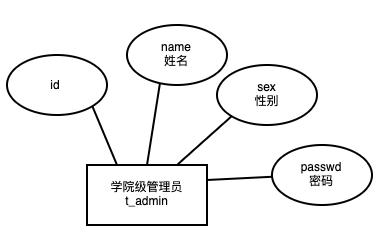
7.1.1学生



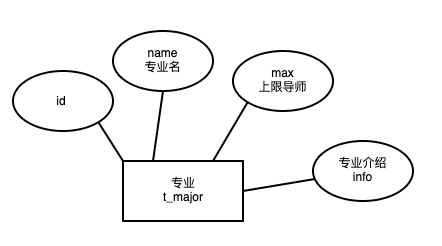
7.1.2 教师



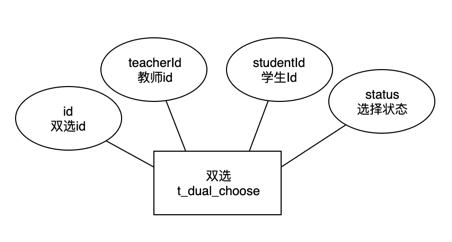
7.1.3 学院级管理员



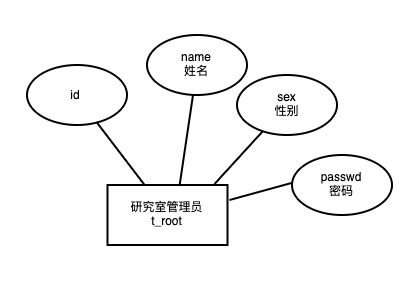
7.1.4 专业



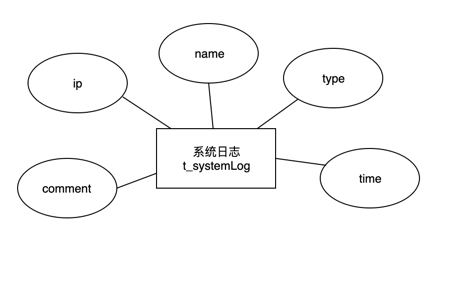
7.1.5 双选



7.1.6 研究室管理员



7.1.7 系统日志



#### 7.2 数据库关系表

学生(id，姓名，性别，学号，邮箱，密码，报考专业，个人简介)

教师(id，姓名，性别，教工号，邮箱，密码，专业，个人简介)

双选(teacherId，studentId，选择状态)

学院级别管理员(id，姓名，性别，密码)

专业(id，姓名，上限导师，专业介绍)

研究室管理员(id，姓名，上限导师，密码)

系统日志(ip，名字，类型，时间，备注)

#### 7.3 数据字典

研究室管理员表(t\_root)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 | 备注 |
| id | int | 否 |  | 自增 | 是 |  | Id |
| name | varchar(50) | 是 |  |  |  |  | 姓名 |
| sex | char(2) | 是 |  |  |  |  | 性别 |
| passwd | varchar(200) | 是 |  |  |  |  | 密码 |

院级管理员表(t\_admin)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 | 备注 |
| id | int | 否 |  | 自增 | 是 |  | Id |
| name | varchar(50) | 是 |  |  |  |  | 姓名 |
| sex | char(2) | 是 |  |  |  |  | 性别 |
| passwd | varchar(200) | 是 |  |  |  |  | 密码 |

学生表(t\_student)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 | 备注 |
| id | int | 否 |  | 自增 | 是 |  | Id |
| name | varchar(50) | 是 |  |  |  |  | 姓名 |
| sex | char(2) | 是 |  |  |  |  | 性别 |
| stuNum | varchar(50) | 是 |  |  |  |  | 学号 |
| email | varchar(50) | 是 |  |  |  |  | 邮箱 |
| major | varchar(50) | 是 |  |  |  |  | 专业 |
| info | varchar(2000) | 是 |  |  |  |  | 信息 |
| passwd | varchar(200) | 是 |  |  |  |  | 密码 |

老师表(t\_teacher)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 | 备注 |
| id | int | 否 |  | 自增 | 是 |  | Id |
| name | varchar(50) | 是 |  |  |  |  | 姓名 |
| sex | char(2) | 是 |  |  |  |  | 性别 |
| teaNum | varchar(50) | 是 |  |  |  |  | 教工号 |
| email | varchar(50) | 是 |  |  |  |  | 邮箱 |
| major | varchar(50) | 是 |  |  |  |  | 专业 |
| info | varchar(2000) | 是 |  |  |  |  | 信息 |
| passwd | varchar(200) | 是 |  |  |  |  | 密码 |

专业表(t\_major)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 | 备注 |
| id | int | 否 |  | 自增 | 是 |  | Id |
| name | varchar(50) | 是 |  |  |  |  | 专业名 |
| max | int | 是 |  |  |  |  | 最大数量 |
| info | varchar(2000) | 是 |  |  |  |  | 专业信息 |

双选(t\_dual\_choose)

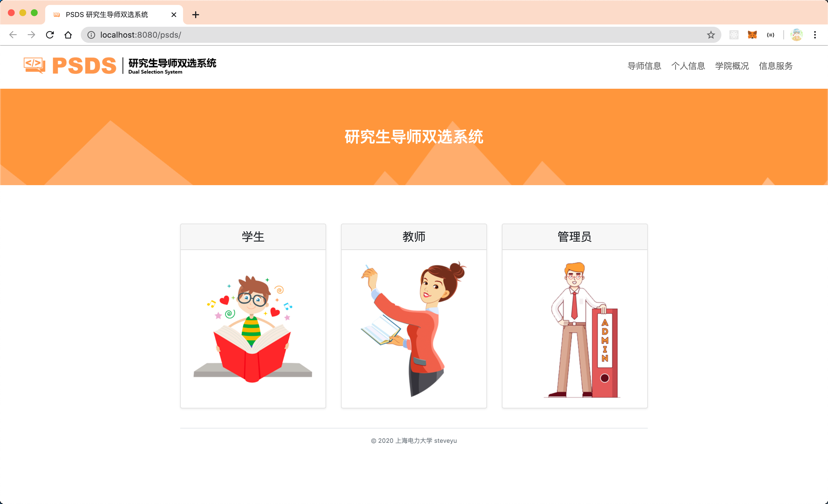
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 | 备注 |
| teacherId | int | 否 |  |  | 是 | 是 | 教师id |
| studentId | int | 否 |  |  | 是 | 是 | 学生id |
| status | int | 是 |  |  |  |  | 状态 |

系统日志(t\_systemLog)

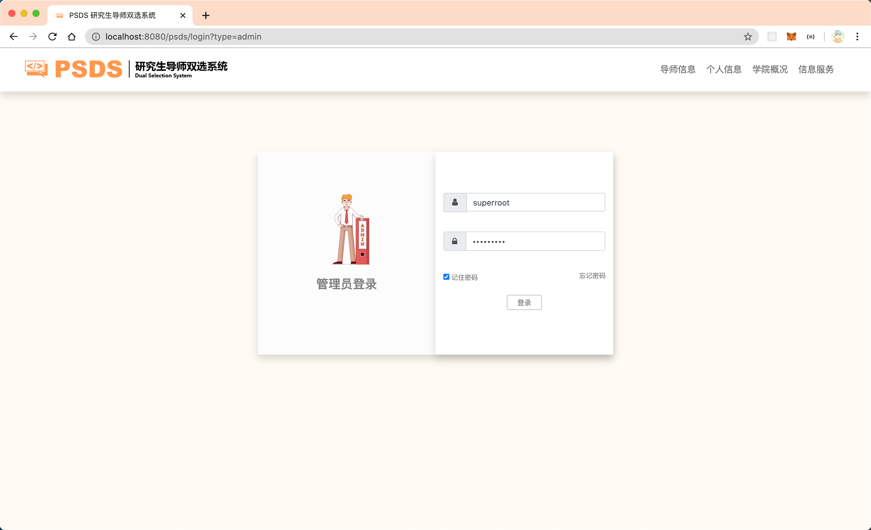
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 | 备注 |
| ip | varchar(20) | 是 |  |  |  |  | ip |
| name | varchar(20) | 是 |  |  |  |  | 姓名 |
| type | varchar(20) | 是 |  |  |  |  | 类型 |
| time | varchar(20) | 是 |  |  |  |  | 时间 |
| comment | varchar(200) | 是 |  |  |  |  | 备注 |

## 9．系统功能介绍

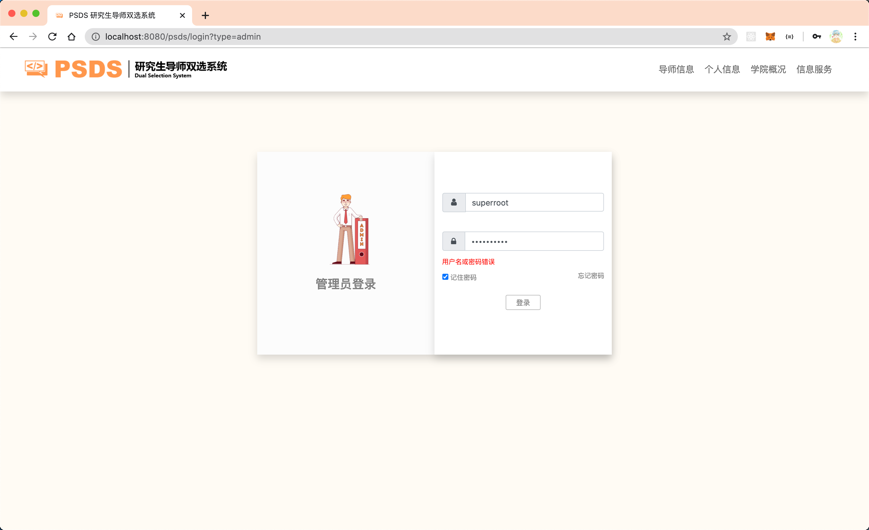
**1.打开系统，访问8080端口进入主页。**



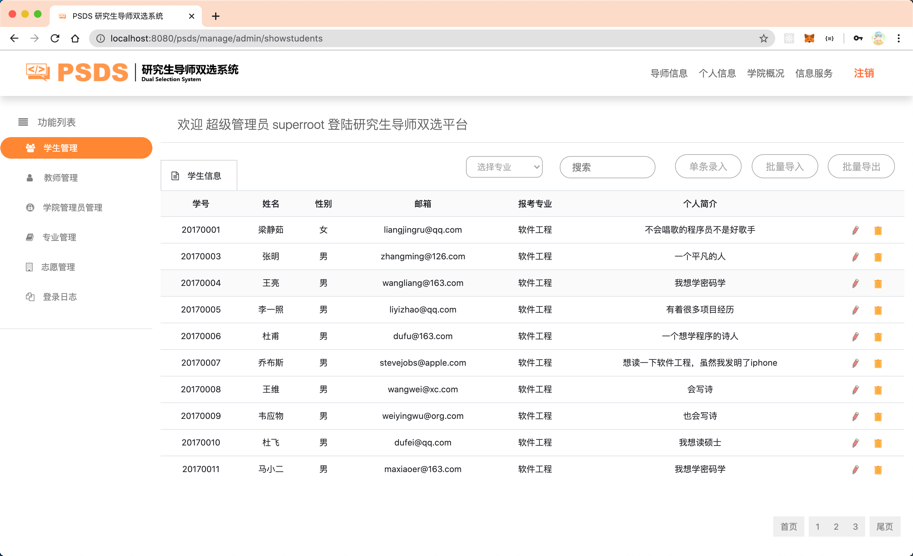
**2.点击管理员，进入管理员页面。**



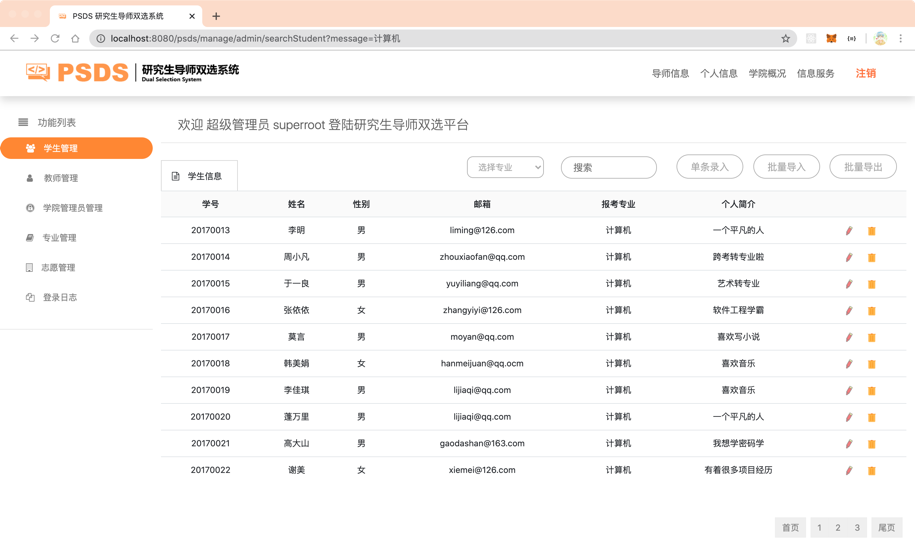
1. **输入用户名，密码，密码错误显示密码错误，输入为空，会报出为空。**



1. 输入用户名，密码，登录成功后，进入管理员首页， 管理员登录端可以登录校方管理员和院方管理员。

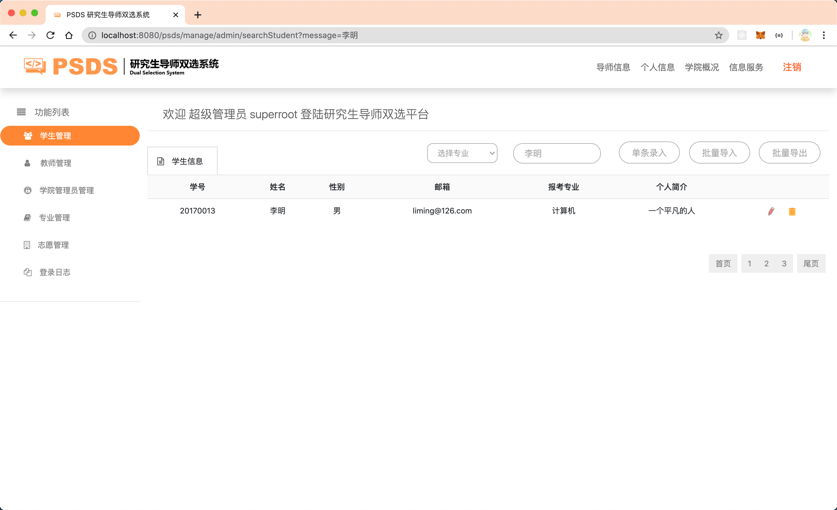


1. 选择专业可以进行筛选相应专业的，比如，我们选择计算机专业，可以筛选出计算机专业的人。

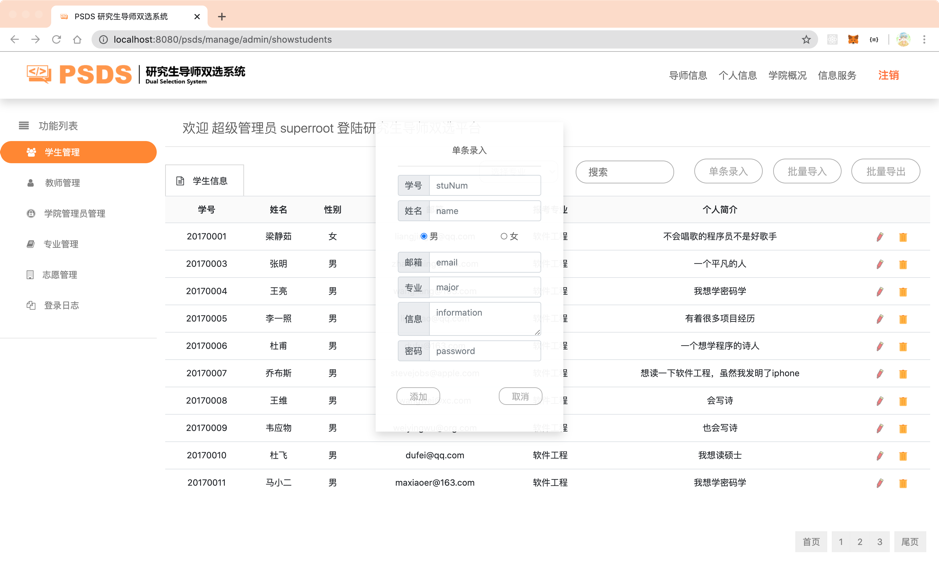


1. 搜索专业可以搜索各种关键词，进行筛选相应的人员。

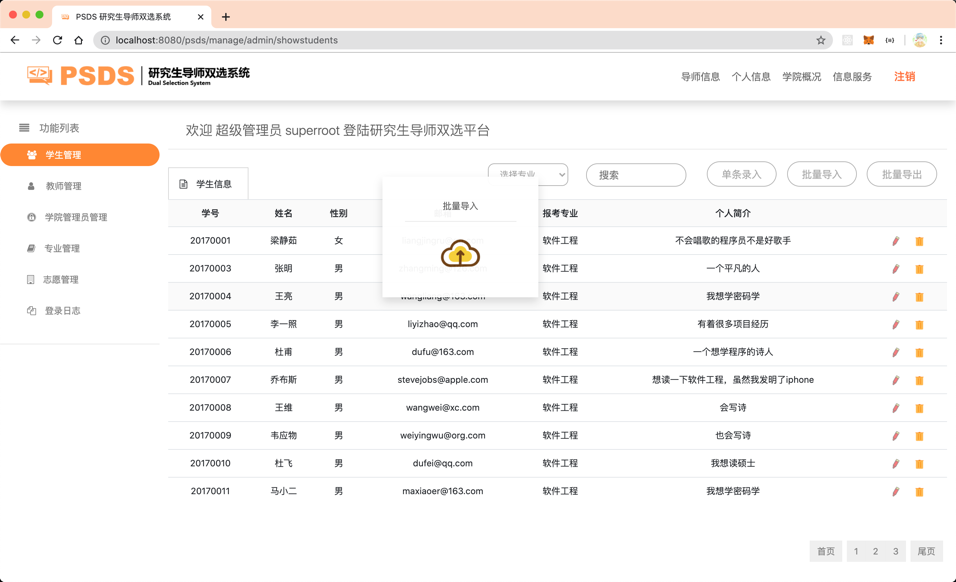
比如：我们搜索李明



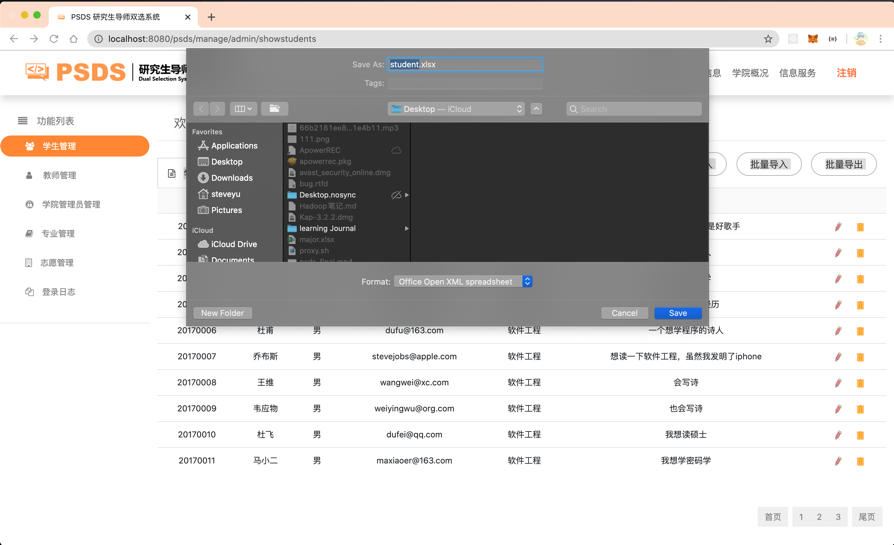
1. 点击单条录入，可以录入单条信息。点击添加，可以添加一条记录。点击取消，可以取消添加。



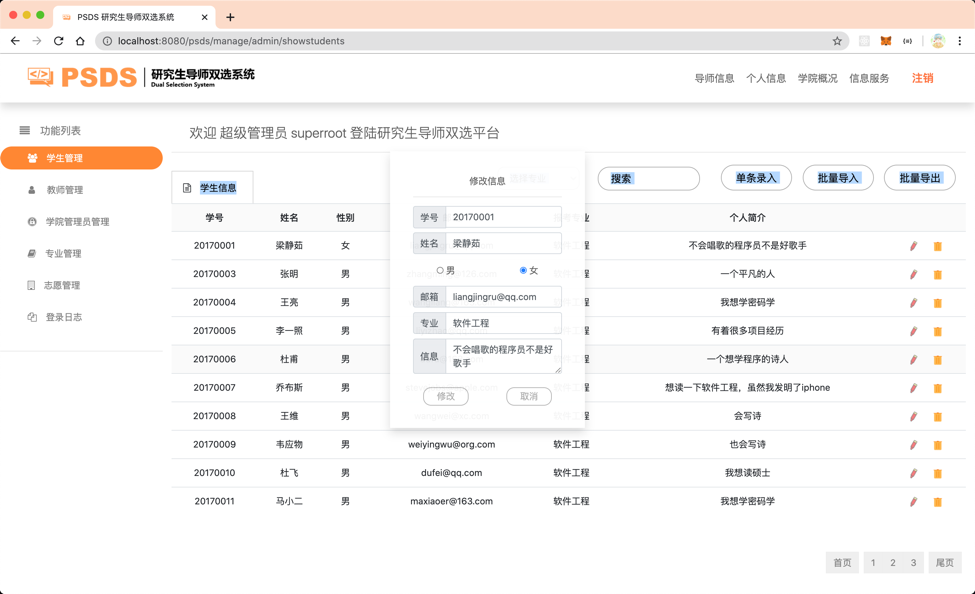
1. 点击批量导入，可以显示导入图标。点击导入，可以导入一个xml文件，后台会进行xml解析，解析完毕会将学生批量加入数据库。点击旁边



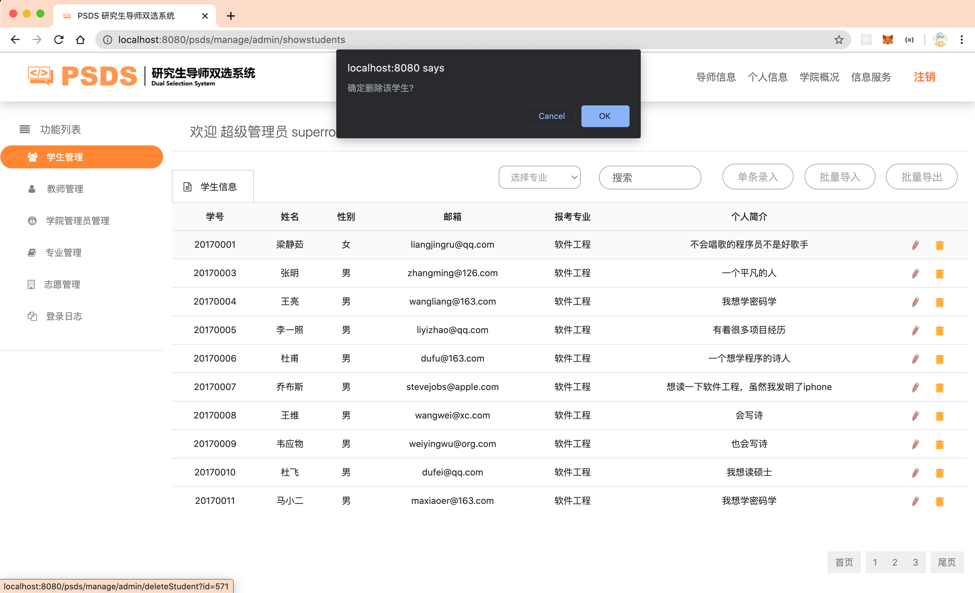
1. 点击批量导出，可以进行导出学生信息。



1. 点击修改可以进行修改学生信息。



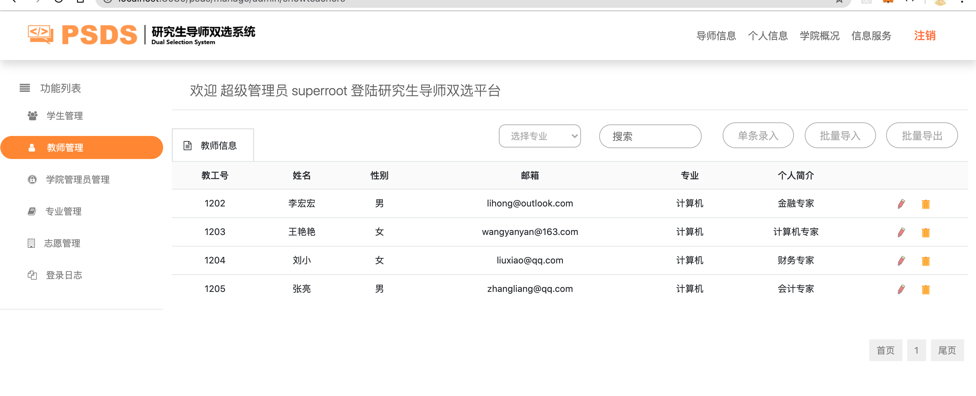
1. 点击删除可以删除学生信息。



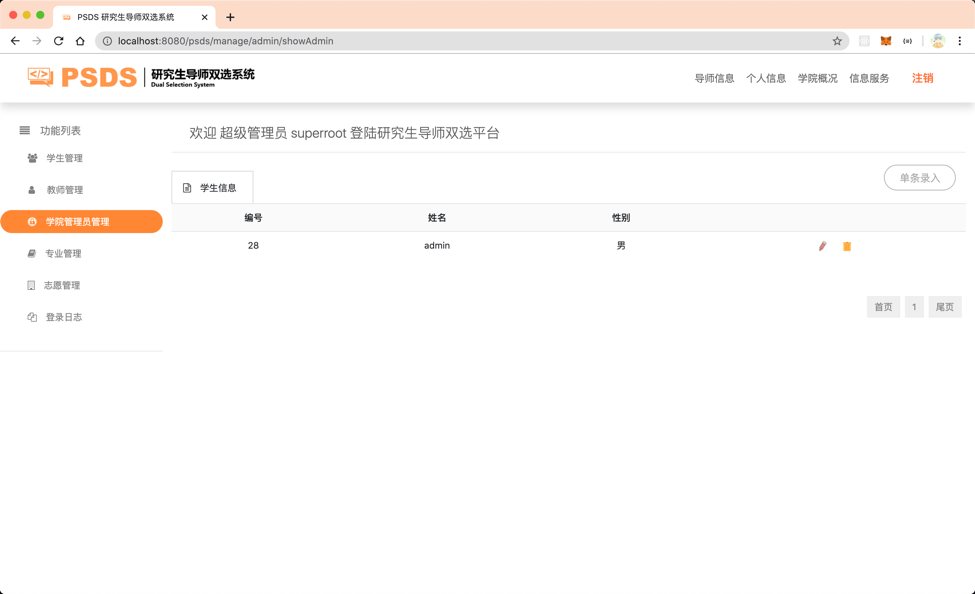
1. 点击页码可以进行翻页



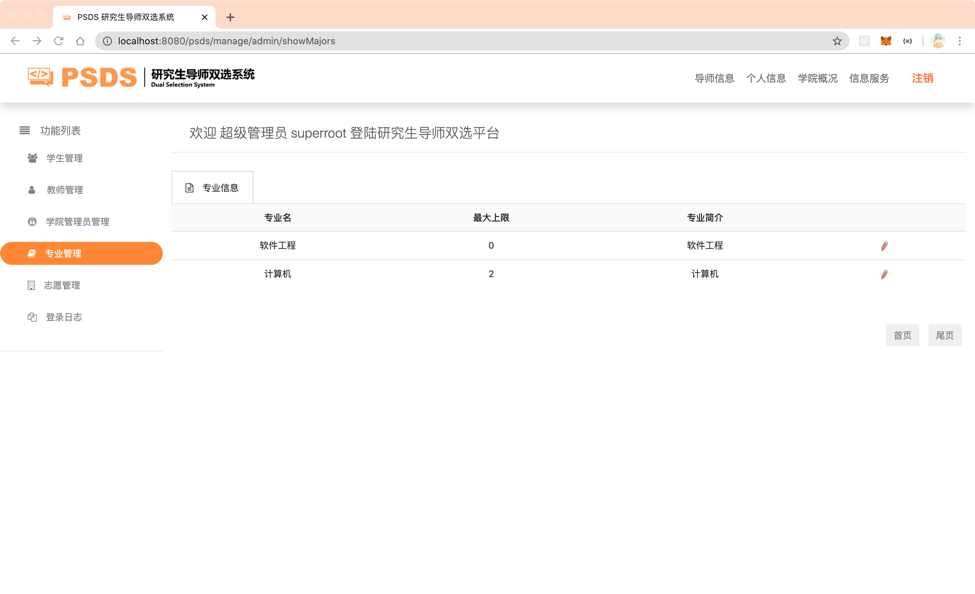
1. 在教室管理页面可以进行和学生管理相同的操作，这边不加以赘述



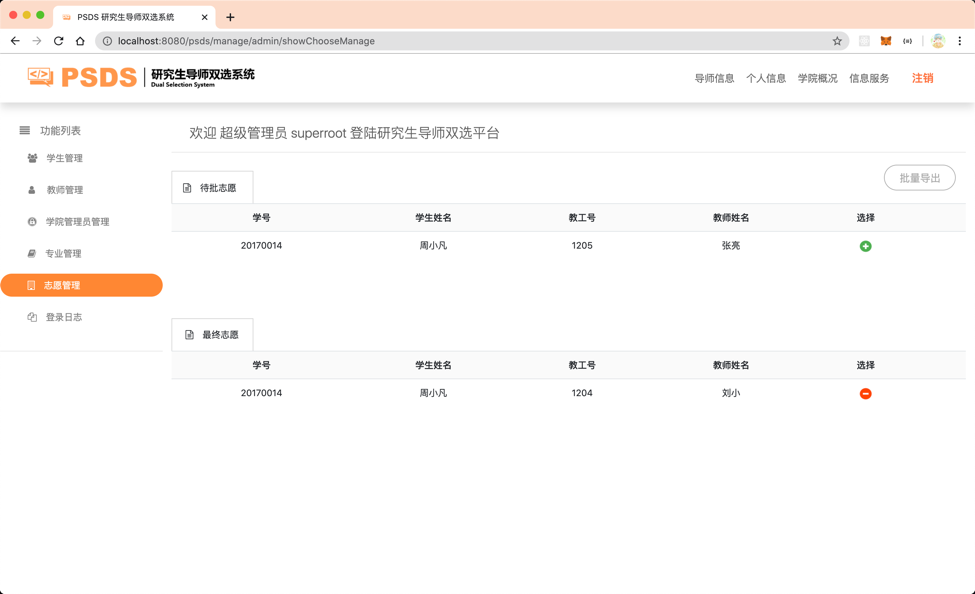
1. 在学院管理员页面，可以进行对学院管理员进行管理。



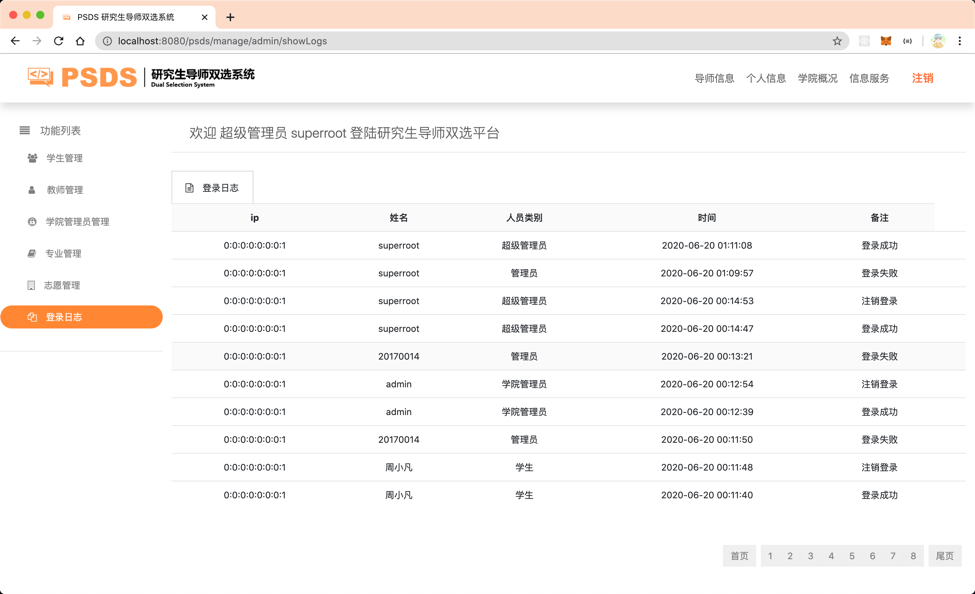
1. 专业管理。在专业管理页面可以设置专业上限，以及进行修改专业相关信息。



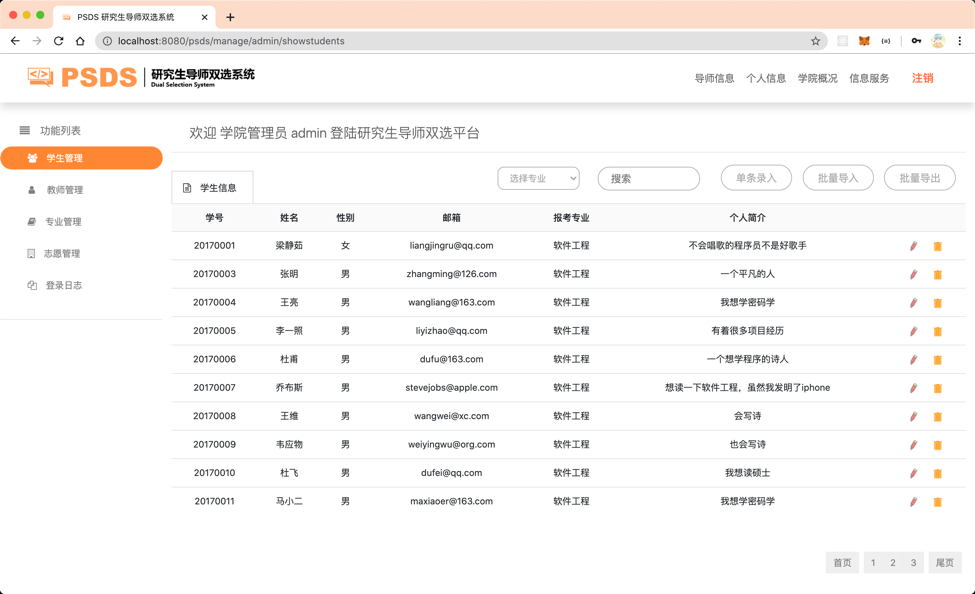
1. 志愿管理。在志愿管理界面，可以批准老师和学生都双选的结果，可以通过，也可以取消通过。这边如果专业人数都达到上限，或者学生已经有了导师，这边会进行系统提示。



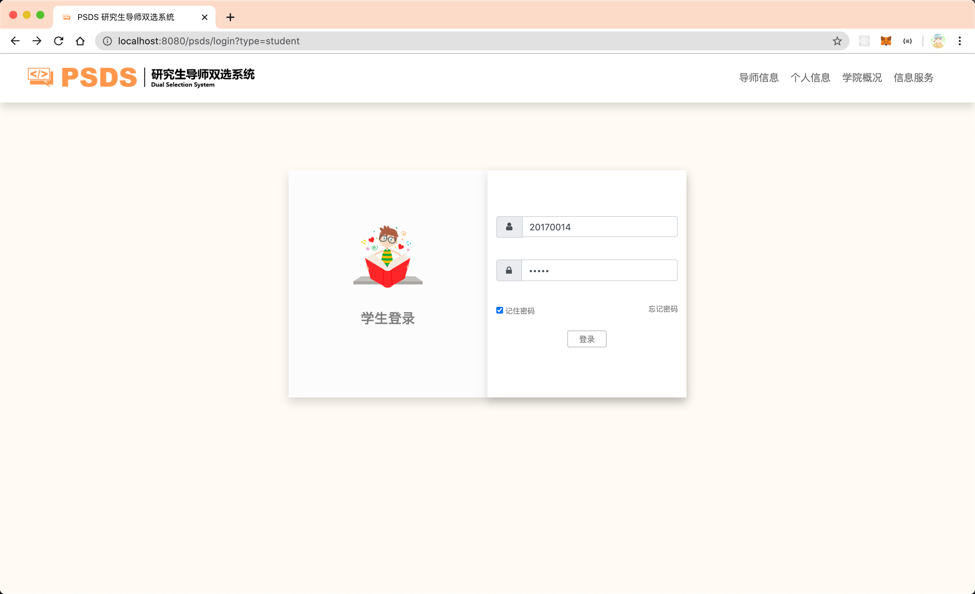
1. 登录日志。登录日志版块可以对登录和注销时间进行存储，方便管理员的查看。



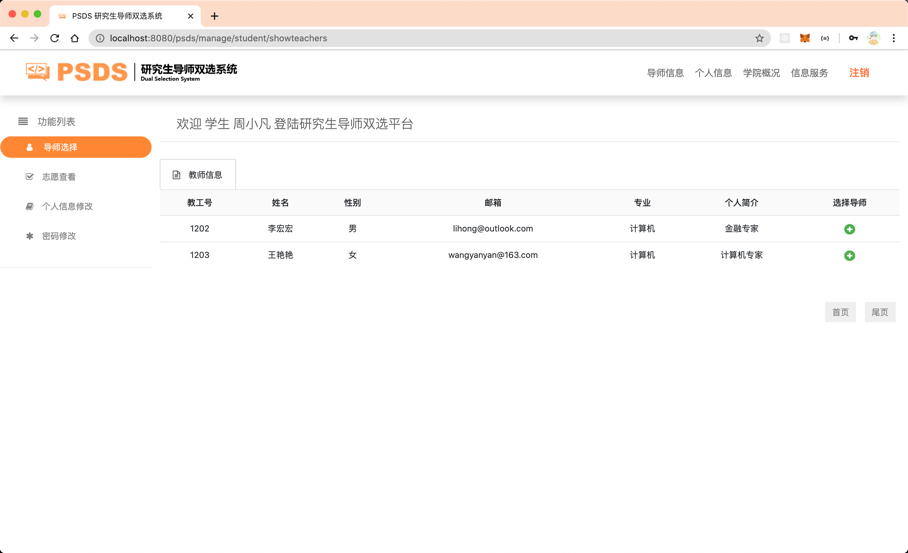
1. 在院级管理员下，我们可以没有校级管理的管理员管理功能。（我们也不可以进行输入地址访问，系统会自动判别你是院级管理员，不给与你权限。



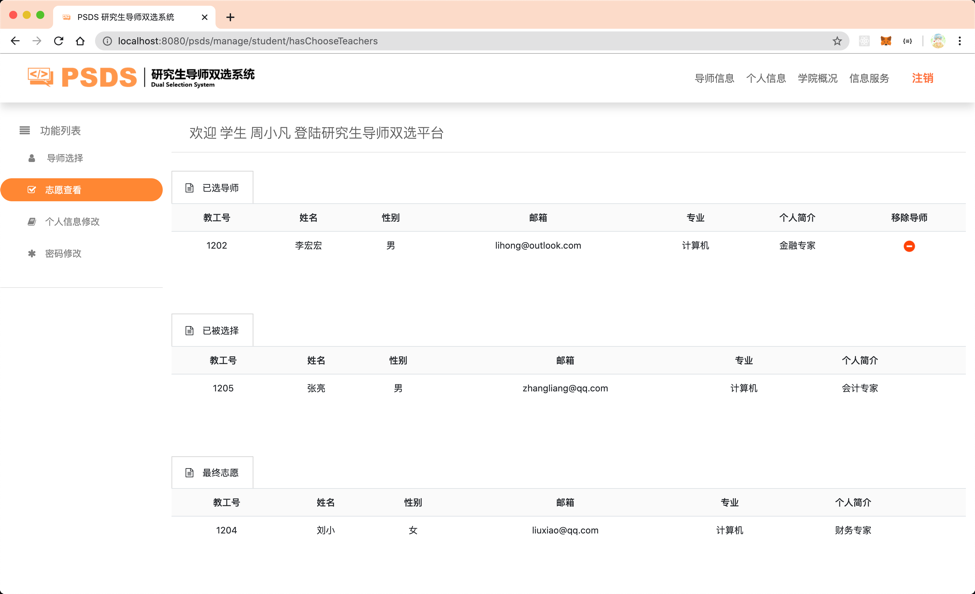
1. 学生登录。进入学生登录页面，输入用户名和密码，即可进行登录。



1. 登录成功后可以进行导师选择（本专业）。我们可以在该页面选择导师。选择超过3个，系统会给出提示。



1. 在志愿查看页面，我们可以查看到选择导师的界面。已被选择是我们正在选择的导师，我们可以对自己志愿进行溢出。已被选择的是老师也选择了你的。最终志愿是管理员确认的最终结果。



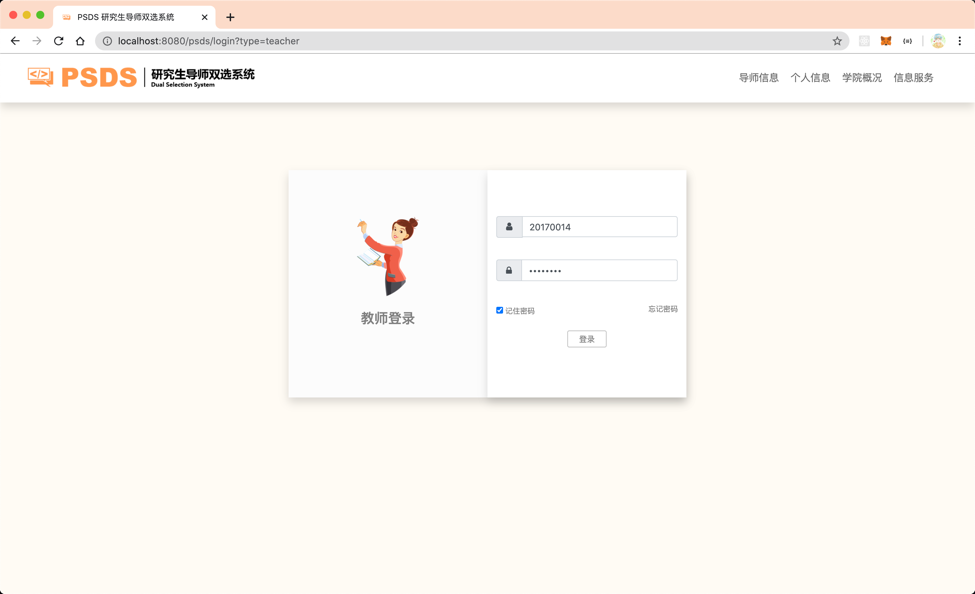
1. 个人信息修改页面，可以修改个人的信息。



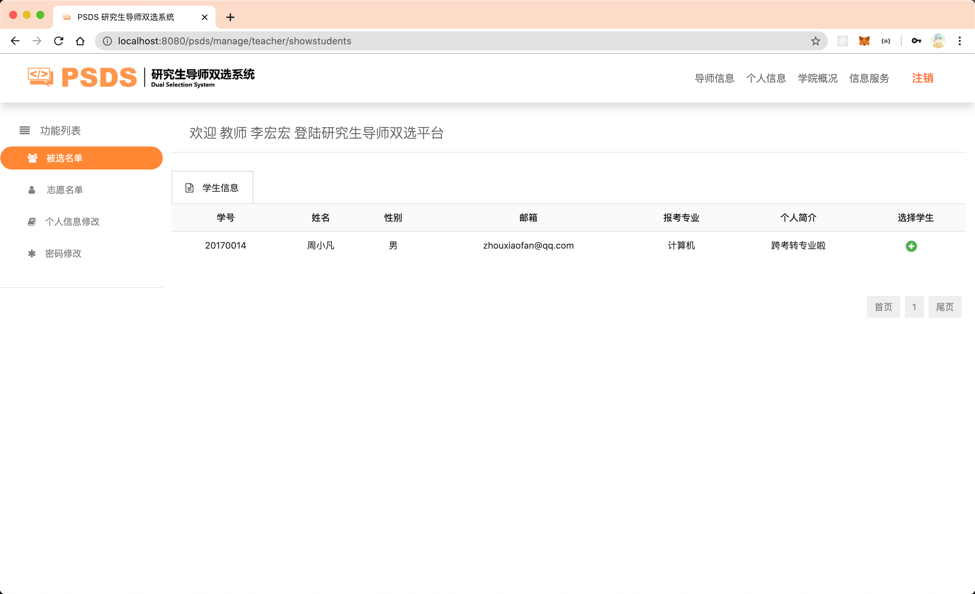
1. 密码修改页面可以进行修改密码。



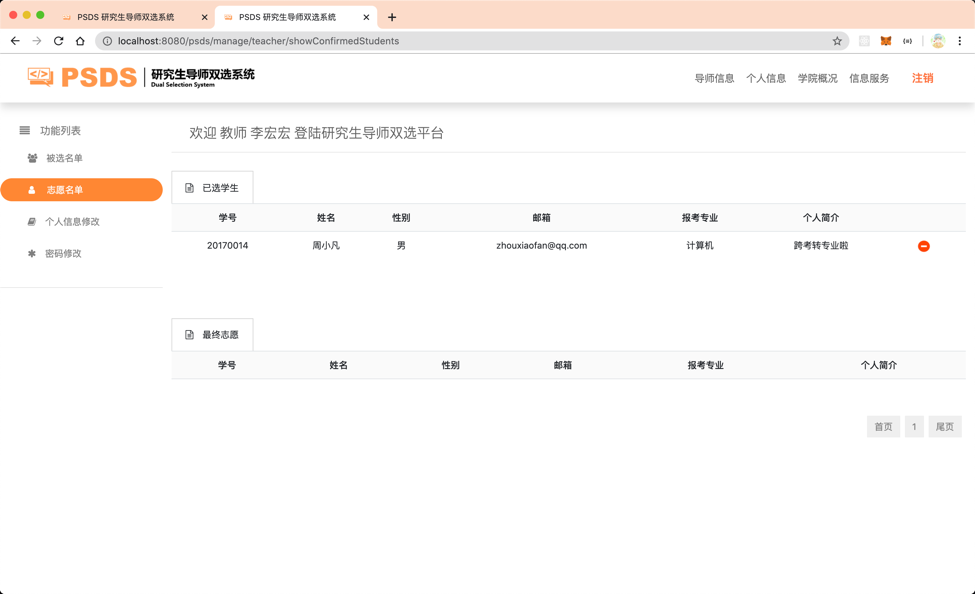
1. 教师登录，输入用户名密码，即可进行教师登录。



1. 教师登录成功后，可以进行展示学生信息，这边展示的是已经选择的学生。



1. 志愿名单中，可以进行查看已选择学生，和最终自己的学生名单。



1. 个人信息修改，可以进行修改自己的个人信息。



1. 密码修改界面，可以进行修改密码。



## 10. 核心模块介绍

### 10.1 登录模块

获取了界面的用户名和密码，并且如果记住密码，则把用户名密码加入session，下次进入的时候就可以进行从session中取出。

否则先判断类型。是管理员，则进入管理员逻辑，是学生，则进入学生逻辑，是教师，则进入教师逻辑。

String loginName = req.getParameter(**"loginName"**);  
 String password = req.getParameter(**"password"**);  
 String type = req.getSession().getAttribute(**"loginType"**) == **null** ? **"student"**: req.getSession().getAttribute(**"loginType"**).toString();  
 String remember = req.getParameter(**"remember"**);  
 *// 记住密码* recitePassword(loginName, password, type, remember, req, resp);  
 *// 三端登陆* **if**(type.equals(**"admin"**)) {  
 *// 最高权限管理员登录* **boolean** flag = rootAdmin(loginName, password, req, resp);  
 *// 学院级别管理员登录* **if**(flag == **false**){  
 academicAdminLogin(loginName, password, req, resp);  
 }  
 } **else if**(type.equals(**"student"**)) {  
 *// 学生登录* studentLogin(loginName, password, req, resp);  
 } **else if**(type.equals(**"teacher"**)) {  
 *// 老师登录* teacherLogin(loginName, password, req, resp);  
 }

### 10.2 上传模块

1．模态框

上传模块制作了一个模态框，用来进行导入xml

2.servlet层，调用easyExcel模块，进行处理Excel的数据。

List<FileItem> formItems = upload.parseRequest(req);  
 **if** (formItems != **null** && formItems.size() > 0) {  
 *// 迭代表单数据* **for** (FileItem item : formItems) {  
 *// 处理不在表单中的字段* **if** (!item.isFormField()) {  
 String fileName = **new** File(item.getName()).getName();  
 String filePath = uploadPath + File.***separator*** + fileName;  
 File storeFile = **new** File(filePath);  
 *// 在控制台输出文件的上传路径* System.***out***.println(filePath);  
*// logger.info(filePath);  
 // 保存文件到硬盘* item.write(storeFile);  
 EasyExcel.*read*(storeFile, Student.**class**, **new** StudentListener()).sheet().doRead();  
 File delFile = **new** File(filePath);  
 delFile.delete();  
 req.getSession().setAttribute(**"message"**, **"批量导入成功"**);  
 }  
 }  
 }

## 11.附录(源代码, 见压缩包)