

**本科生毕业设计**

[基于不同课程类别的高校学生评教系统的设计与实现](http://lwgl.xcu.edu.cn/student.shtm" \l "!/javascript:void(0))

**学 院　信息工程学院**

**专 业　计算机科学与技术**

**班 级　2014级14计科1班**

**学 号　5001140010**

**学 生 姓 名　郭延明**

**联 系 方 式　17133790332**

**指 导 教 师　王亚 　职称：讲师**

2018年5月

**独 创 性 声 明**

本人郑重声明：所呈交的毕业论文（设计）是本人在指导老师指导下取得的研究成果。除了文中特别加以注释和致谢的地方外，论文（设计）中不包含其他人已经发表的研究成果。与本研究成果相关的所有人所做出的任何贡献均已在论文（设计）中作了明确的说明并表示了谢意。

签名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

　　　　　　　　　　　　　　　　　　　\_\_\_\_\_\_\_\_年\_\_\_\_\_\_月\_\_\_\_日

**授权声明**

本人完全了解许昌学院有关保留、使用本科生毕业论文（设计）的规定，即：有权保留并向国家有关部门或机构送交毕业论文（设计）的复印件和磁盘，允许毕业论文（设计）被查阅和借阅。本人授权许昌学院可以将毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编论文（设计）。

本人论文（设计）中有原创性数据需要保密的部分为（如没有，请填写“无”）：

学生签名：

　　　　　年　　月　　日

指导教师签名：

　　　　　　　　　　　　　　　　　 年　　月　　日

说明:独创声明和授权声明不编页码，单独成页。

基于不同课程类别的高校学生评教系统的设计与实现

摘 要

教学质量是高等学校的生命线，提高教师课堂教学质量是高校的毕生追求。而学生评教是进行教学质量监控的重要手段。高校的课程分类多样，高校学生评教对所有课程都采用相同的一套评教内容，这不能准确的反应教师的教学水平。针对不同的教学内容应当设计针对性的评教标准，这样才能更好的体现出教师的教学水平。

系统针对此需求进行开发出基于不同课程类别的高校学生评教系统，系统采用目前流行的MVC设计模式进行设计，开发工具为[IntelliJ IDEA](https://www.jetbrains.com/idea/) 2017，数据库采用MariaDB 5.5.56 ，web服务端采用Spring，MyBatis技术，安全框架采用Apache Shiro，前端使用Html、JavaScript、Css技术。

关键词： JavaWeb；Spring；Mvc；Shiro;Mysql;教学质量;教师评教

**Design and Implementation of College Student Evaluation System Based on Different Courses**

**Summary**

The quality of teaching is the lifeline of colleges and universities, and improving the quality of teachers' classroom teaching is the lifelong pursuit of colleges and universities. The student evaluation of teaching is an important means of monitoring the quality of teaching. Colleges and universities have a variety of courses, and college students' evaluation of teaching uses the same set of evaluation content for all courses. This does not accurately reflect the teaching level of teachers. Different teaching content should be designed for targeted assessment of teaching standards, so as to better reflect the teaching level of teachers.

The system has developed a student evaluation system for colleges and universities that is based on different curriculum categories based on this requirement. The system adopts the current popular MVC design pattern design, development tools for IntelliJ IDEA 2017, database using MariaDB 5.5.56, web server using Spring, MyBatis technology, security framework using Apache Shiro, front-end use Html, JavaScript, Css technology.

**Keywords**: JavaWeb; Spring; Mvc; Shiro; Mysql; Educational administration system; Teacher evaluation

目　录

[1. 绪 论 1](#_Toc29088)

[1.1 研究背景与现状 1](#_Toc28230)

[1.1.1 研究背景 1](#_Toc29525)

[1.1.2 选题研究现状 1](#_Toc31507)

[1.2 系统功能简介 1](#_Toc18244)

[1.2.1 系统的功能及意义 1](#_Toc32738)

[1.2.2 系统的技术路线与研究方法 1](#_Toc26668)

[1.3 系统开发环境和开发工具 1](#_Toc31697)

[2. 系统分析 3](#_Toc4742)

[2.1 可行性分析 3](#_Toc29676)

[2.2 需求分析 3](#_Toc20445)

[2.2.1 业务目标 3](#_Toc10641)

[2.2.2 业务用例（用例图) 4](#_Toc18958)

[2.2.3 系统实体关系图（E-R图） 7](#_Toc15678)

[2.3 、系统拟解决的关键问题 9](#_Toc3435)

[2.4 性能分析 9](#_Toc26778)

[3. 系统设计 11](#_Toc22807)

[3.1 概要设计 11](#_Toc4596)

[3.1.1 总体结构设计 11](#_Toc22288)

[3.1.2 代码设计原则 11](#_Toc19600)

[3.1.3 功能要求 11](#_Toc8645)

[3.2 详细设计 12](#_Toc13428)

[3.2.1 功能模块详细设计 12](#_Toc27439)

[3.2.2 数据库设计 15](#_Toc18841)

[4. 系统实现 21](#_Toc31161)

[4.1 系统实现概述 21](#_Toc19884)

[4.2 系统模块实现 21](#_Toc4490)

[4.2.1 登录模块 21](#_Toc191)

[4.2.2 管理员模块 25](#_Toc21732)

[4.2.3 教师端 29](#_Toc15515)

[4.2.4 学生端 31](#_Toc15541)

[5. 系统测试 35](#_Toc11158)

[5.1 概述 35](#_Toc4760)

[5.2 测试方法 35](#_Toc32531)

[5.3 测试用例 35](#_Toc24523)

[5.3.1 系统的登录模块测试用例 35](#_Toc27422)

[5.3.2 系统教学评教模块测试 35](#_Toc8421)

[5.3.3 查询功能测试 36](#_Toc6077)

[5.3.4 表单检验测试 36](#_Toc19551)

[5.3.5 用户维护测试 36](#_Toc32234)

[参考文献 38](#_Toc27170)

[附 录 39](#_Toc5469)

[致 谢 55](#_Toc11445)

# 绪 论

## 研究背景与现状

### 研究背景

一所高校能否持续稳定的发展，教学质量是至关重要的。教师授课质量是高校的命脉，每所高校也将提高教师质量视为终身追求的目标。学校按期对教师教学质量进行评估，通过收集高校学生对一个教师，一门课程的反馈信息进行总结以此来掌握各教师的教学能力，并参考这些数据对课程和教师进行更加合理的安排组织。

教师评教开始于美国20世纪20年代，60年后才开始逐渐出现在我国。目前全国高校已经相当普及，成为高校的一个常规性工作。随着互联网的发展教学评教也很快实现了网络化，教师评教网络系统也成为高校教务系统中的重要模块。

### 选题研究现状

早一批的教学评教系统随着时间的推进，最初的设计开始暴露出许多弊端，这些教学评教系统对所有的老师和课程都采用同一套的评教内容，这是有缺陷的。以此收集的数据对于教师和课程也不具备针对性，评教的价值也将不能充分的体现。本系统将针对不同类型的课程设计不同的评价内容，使评价内容更有针对性地反映不同学科教师的教学水平，并据此制定新的评价体系。

## 系统功能简介

### 系统的功能及意义

系统要实现的主要功能包括教师，学生的选课系统和学生评教系统。系统首先实现双向选课的功能来关联教师、课程及学生之间的关系。本系统的课程具有艺术实践类，理论类，理论+实践类属性。在学生评教系统模块中学生对课程及教师评教时根据此属性系统将提供不同的问卷，以此来提高评教内容和课程的契合度。使得评价内容更有价值。

### 系统的技术路线与研究方法

系统的开发遵循：需求分析-概要设计-详细设计-编码实现与系统测试5个环节。

结合面向对象技术和快速原型方法对系统进行设计；

使用构建工具组织jar包,系统采用MVC分层设计；

## 系统开发环境和开发工具

本系统采用Java语言作为后台开发语言，使用JavaScript、CSS、HTML和部分前端插件开发前台。使用MariaDB作为后台数据库，系统开发遵循MVC的设计模式。WEB服务器使用Tomcat，系统开发工具使用[IntelliJ IDEA](https://www.jetbrains.com/idea/)。

具体环境如下：

1. Java环境：JDK1.8.0\_152-release
2. Tomcat版本：Tomcat8.5.30
3. MariaDB版本：MariaDB 10.2
4. [IntelliJ IDEA](https://www.jetbrains.com/idea/)版本：[IntelliJ IDEA](https://www.jetbrains.com/idea/)2017.3
5. Maven构建工具：Maven 3.5.3

# 系统分析

## 可行性分析

（1）经济可行性

本系统的开发采用java语言开发，开发工具是开源免费的[IntelliJ IDEA](https://www.jetbrains.com/idea/) 2017社区版本，使用的数据库也是开源免费的MariaDB；系统的开发仅需投入开发的时间精力成本。本系统是基于B/S结构的系统，用户只需拥有一个浏览器无需使用特定的软硬件。

通过简单的分析，该系统满足经济可行性。

1. 技术可行性

本系统是B/S结构的系统。服务端采用安全稳健的java语言开发，java语言发展历史悠久，拥有足够的前人经验和众多的基层服务支持；框架采用成熟的spring框架来架构，分层的设计使得后端逻辑功能高内聚低耦合便于维护；数据持久层使用Mybatis能够灵活的操纵数据的读写；数据库使用开源的MariaDB，其前身为Mysql数据库，多次的版本迭代已经使得这个数据库十分成熟。

客户端工作界面是通过WWW浏览器来实现，这种模式统一了客户端，简化了系统的开发、维护和使用。前端技术已经发展多年，拥有足够的文献和经验可供学习和参考

通过简单的分析，该系统基于此的设计和实现是可行的

（3）操作可行性

该系统对于用户的要求仅仅是用户安装一个浏览器，无论你是使用的Chrome，Opera，Foxfix还是其他的浏览器你随时都能够使用本系统，对于设备也是跨平台的你可以在个人电脑上、安卓手机、苹果手机上轻松访问。跨平台的特性使用户能够很轻松的获取服务，在系统易用性和可操作性上完全满足。

## 需求分析

### 业务目标

本系统是基于不同课程类别设计不同的评教内容以此对教师进行更具针对性的评教。本系统要实现简单操作性强的用户交互界面，高效的教学评教工作，在性能上保证系统运行流畅安全，严格区分角色权限。在此基础上实现针对课程类别来对教师进行评教的功能。

### 业务用例（用例图)

基于不同课程类别的高校学生评教系统的用户对象为高校的管理者，教师和学生用户，不同的角色登录系统拥有不同的权限及功能。

图2-1为系统总例图，体现该系统的用户为高校管理者，教师和学生用户。

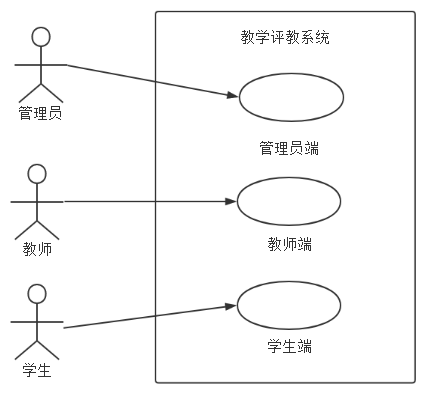


图 2-1 用例总图

管理员用户登录系统后，能够对系统的数据字典进行维护、管理用户、管理课程、管理评教内容、管理评教记录。管理员用例图如图 2-2 管理员用户业务用例所示。

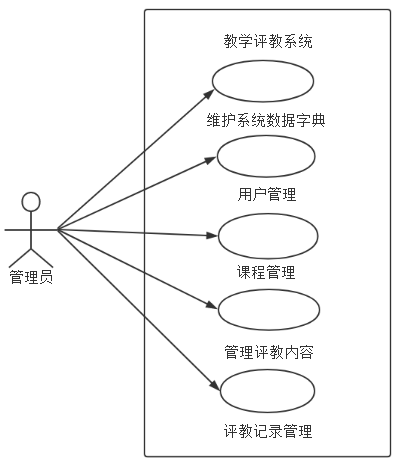


图 2-2 管理员用户业务用例

教师用户登录系统后，可以对自己的信息进行维护，查看课程列表并进行选课以关联自己与课程的关系，可以查看自己所带的学生信息。教师用户业务用例图如图 2-3所示。

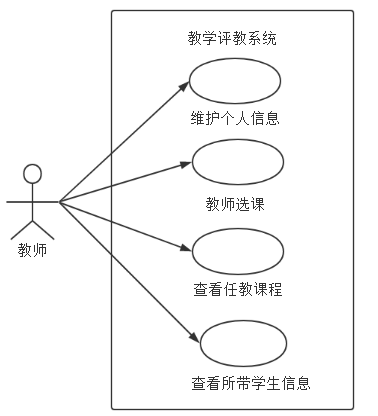


图 2-3 教师用户业务用例

学生用户登录系统可以维护自己的个人信息，可以在系统中进行选课，可以对所选课程进行评教。学生用户业务用例图如图 2-4所示。

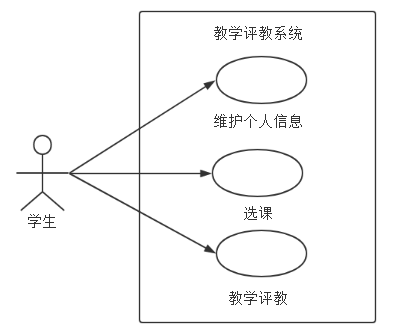


图 2-4 学生用户业务用例

### 系统实体关系图（E-R图）

系统实体间的关系对系统实现时能否更好的对数据进行关联操作至关重要，系统实体的合理设计将影响系统的实现方式。这里对本系统涉及的实体进行分析和设计。

系统的用户实体包含管理员，教师和学生实体，教师，管理员，学生的主键作为外键来关联用户，其E-R图如图 2-5所示：

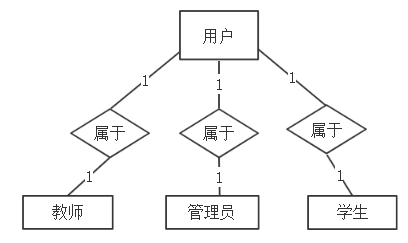


图 2-5 用户ER关系图

用户E-R关系图中的实体属性如下：

学生（学号、姓名、性别、班级、电话号、身份证号、邮箱、地址）；

教师（教职工号、姓名、性别、职称、电话号、身份证号、邮箱、地址）；

用户（用户ID,学生/教师/管理员ID，密码）。

课程实体可以分别与教师和学生建立关联关系，教师任教多门课程，多名学生参与多门课程。

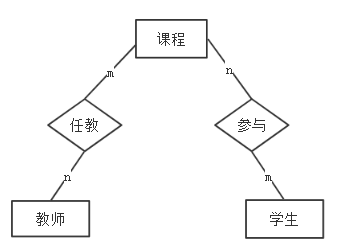


图 2-6 课程ER关系图

课程ER关系图中的实体属性如下：

学生（学号、姓名、性别、班级、电话号、身份证号、邮箱、地址）；

教师（教职工号、姓名、性别、职称、电话号、身份证号、邮箱、地址）；

课程（课程编号、课程名、开设时间、课时、学分、地点、课程类型、上课人数）

教师、学生、评教内容及课程都由管理员来管理。管理员系统E-R图如图 2-7所示：

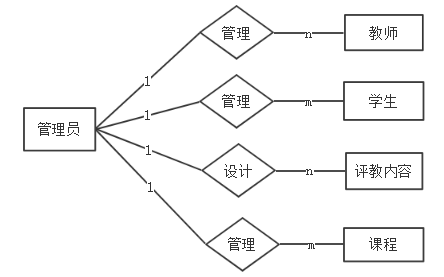


图 2-7 管理员系统ER图

管理员系统ER图关系图中的实体属性如下：

评教内容（课程类型、评教类型、分值、内容）；

学生（学号、姓名、性别、班级、电话号、身份证号、邮箱、地址）；

教师（教职工号、姓名、性别、职称、电话号、身份证号、邮箱、地址）；

课程（课程编号、课程名、开设时间、课时、学分、地点、课程类型、上课人数）。

集成的E-R图如下图 2-8所示：

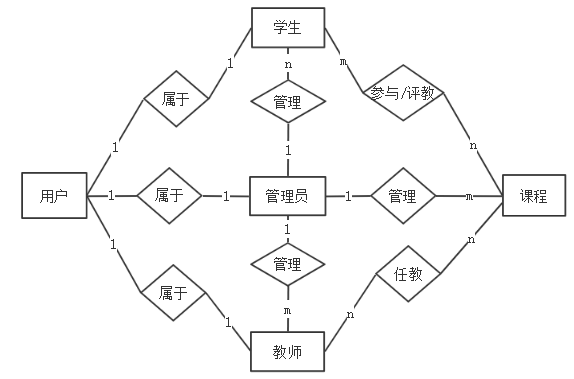


图 2-8 系统E-R集成图

系统E-R集成图涉及的关系模型如下：

课程学生关系模型（学生、课程）；

教师课程关系模型（教师、课程）。

## 、系统拟解决的关键问题

1.处理和设计好教师、课程、学生、课程类型等对象之间的关系模型；

2.完成学生选课模块建立学生、课程、教师的关系模型；

3.根据课程类别设计多元的评教内容；

4.对评教后的数据进行数据挖掘形成图表报告，多方位，多维度的分析既有数据，为教师、高校提供可参考的数据支持。

## 性能分析

1. 系统使用Java语言作为后台开发语言，Java是目前使用最为广泛的网络编程语言之一。它具有简单，面向对象，稳定，与平台无关，解释型，多线程，动态等特点[1] 。使用java开发本系统稳定健壮且易于维护。
2. 本系统使用的数据库MariaDB目前也是许多互联网公司都在采用的数据库，以此来作为本系统的数据库完全能够应对。
3. 系统是属于B/S结构的，用户使用浏览器便可以使用本系统，系统的前台页面遵循CMAScript 6标准，随着近年来Google公司对于浏览器的探索，更多快速方便的浏览器的推出，使得前台技术突飞猛进，使用网页也能体验到丰富的功能。在此基础上本系统使用了目前流行的几款前台插件，以此来提高用户体验。

# 系统设计

## 概要设计

### 总体结构设计

经过对基于不同课程类别的高校学生评教系统的认真剖析，结合实际需要，本系统设计管理员端、教师端和学生端。系统结构如下图所示：

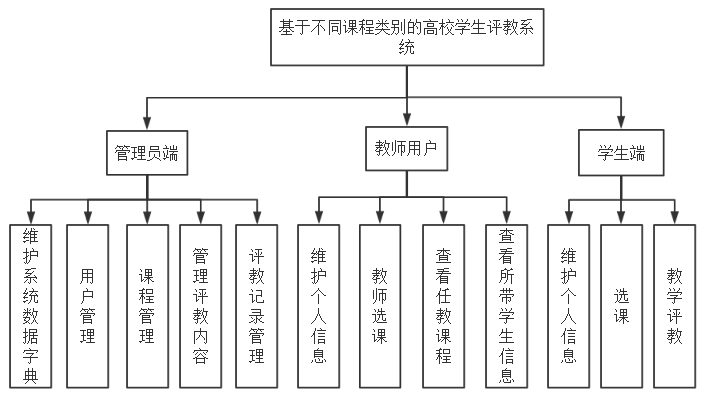


图 3-1系统结构图

通过该系统管理员可以对字典进行维护，对用户进行管理，对课程进行管理，对评教内容和评教记录进行管理。

教师通过教师端可以维护个人信息、进行选课、查看任教课程和查看选课的学生信息。

学生通过学生端可以进行个人信息的维护、选课和教学评教。

### 代码设计原则

良好规范的代码设计原则可以提高系统的可维护性，有助于快速方便的找到系统出现问题的原因[2]，本系统在开发过程中坚持以下代码设计原则：

（1）对每个方法内的重要代码行进行科学标注，使之易读易懂[2]。

（2）每个功能模块都应该高内聚低耦合[3]。

（3）每个模块的功能应该单一化，避免一个模块实现多个功能[3]。

### 功能要求

（1）系统应操作简单，上手容易，拥有友好的UI界面。

（2）完善规范的基础信息的管理维护。增加系统的而灵活性可扩展性。

（3）满足数据库设计的三大范式[4]。

（4）支持多线程多人登录使用，有权限过滤功能，回话管理功能[5]。

（5）为了方便用户使用，要求系统支持模糊查询功能。用户可根据条件选择多个查询方式。

## 详细设计

### 功能模块详细设计

每个具有权限控制的系统都具有登录认证功能，该系统也实现了登录认证的模块。登录模块流程图见图 3-2 登录模块：

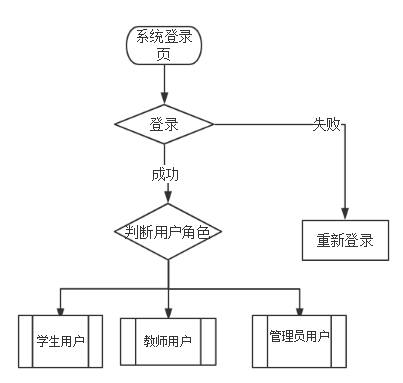


图 3-2 登录模块

系统的管理工作，初始化工作及维护功能全由管理员操作。系统也设计了管理员模块。管理员子模块具有的功能及操作流程如图 3-3所示：

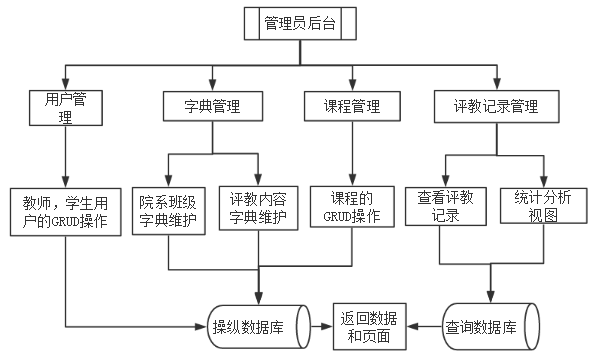


图 3-3 管理员模块

系统设计了教师端模块，教师拥有的功能及工作流程如图 3-4所示。

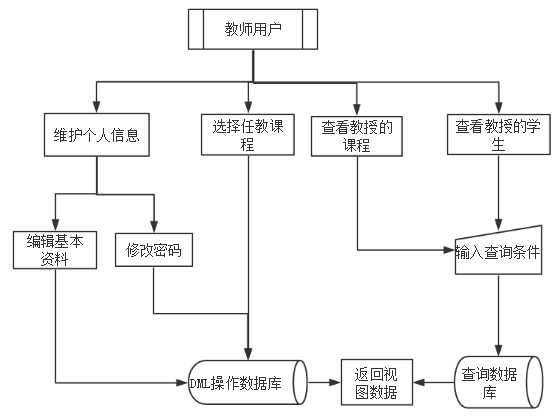


图 3-4 教师模块

系统的学生用户子模块具有的功能及流程如所示。

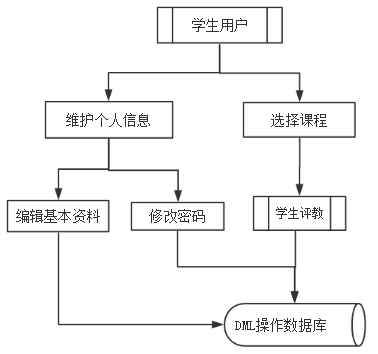


图 3-5 学生模块

系统的评教功能流程图如图 3-6所示。

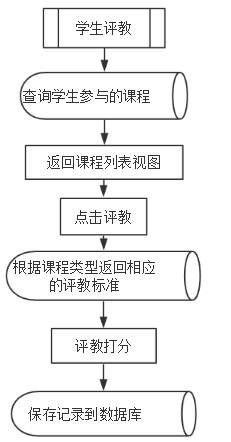


图 3-6 评教模块

### 数据库设计

根据系统设计需要，结合ER图，设计数据库表结构。共11个表，设计的表包括tb\_user、dic\_classroom、dic\_coursetype、dic\_scope、gx\_stucourse、gx\_teacourse、tb\_course、tb\_pjcontent、tb\_pjrecord、tb\_student、tb\_teacher，表结构如表3-1~表3-11所示：

表 3-1 tb\_user表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | ID |
| uid | varchar | 50 | 否 | student,teacher的ID |
| role | int | 2 | 否 | 1，管理员；2，教师；3，学生 |
| userName | varchar | 30 | 否 | 用户账号； |
| userPwd | varchar | 30 | 否 | 密码 |
| block | varchar | 2 | 否 | 用户锁定 1，锁定 0解锁 |
| isadmin | int | 1 | 否 | 是否有管理员权限 |
| isstudent | int | 1 | 否 | 是否是学生用户 |
| isteacher | int | 1 | 否 | 是否有教师权限 |
| remark | varchar | 255 | 否 | 备注 |

表 3-2 tb\_teacher表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | 教师工号 |
| name | varchar | 50 | 否 | 教师姓名 |
| teaLevel | varchar | 50 | 否 | 教师职称 |
| yuanxi | varchar | 255 | 否 | 院系 |
| telNum | varchar | 15 | 否 | 教师电话 |
| sex | varchar | 5 | 否 | 性别 |
| address | varchar | 255 | 否 | 地址 |
| mail | varchar | 50 | 否 | 邮箱 |
| Reamrk | Varchar | 255 | 否 | 备注 |

表 3-3 tb\_student 学生表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | 学生 学号作为id |
| classid | varchar | 255 | 否 | 班级编号 |
| name | varchar | 255 | 否 | 姓名 |
| sex | varchar | 5 | 否 | 性别 |
| telnum | varchar | 255 | 否 | 电话号 |
| mail | varchar | 255 | 否 | 邮箱 |
| address | varchar | 255 | 否 | 家庭住址 |
| idcardNum | varchar | 11 | 否 | 身份证号码 |
| remark | varchar | 255 | 否 | 备注 |

表 3-4 tb\_course课程表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | 课程id |
| courseName | varchar | 255 | 否 | 课程名 |
| courseHours | varchar | 255 | 否 | 课时 |
| address | varchar | 255 | 否 | 上课地址 |
| courseScore | int | 2 | 否 | 学分 |
| year | varchar | 255 | 否 | 年度 |
| term | varchar | 255 | 否 | 上下学期 |
| courseType | varchar | 255 | 否 | 课程类型 |
| remark | varchar | 255 | 否 | 备注 |
| timeplan | varchar | 255 | 否 | 课程时间安排 |
| commonType | varchar | 255 | 否 | 公共选修 |
| countStu | varchar | 500 | 否 | 上课人数 |

表 3-5 tb\_pjContent 评教内容表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | id |
| courseType | varchar | 255 | 否 | 课程类型 |
| pjType | varchar | 255 | 否 | 评教类型 |
| maxValue | varchar | 20 | 否 | 最高分 |
| block | varchar | 2 | 否 | 是否启用 |
| sort | varchar | 50 | 否 | 排序 |
| contentStr | varchar | 255 | 否 | 评教内容 |
| remark | varchar | 255 | 否 | 备注 |

表 3-6 gx\_teaCourse 教师课程关系表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| teaID | varchar | 50 | 是 | 教师ID |
| courseID | varchar | 50 | 否 | 课程id |
| remark | varchar | 255 | 否 | 备注 |

表 3-7 gx\_stuCourse 学生课程关系表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| StuID | varchar | 50 | 是 | 学生ID |
| courseID | varchar | 50 | 否 | 课程id |
| remark | varchar | 255 | 否 | 备注 |

表 3-8 dic\_scope 院系班级字典表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | 班级id |
| yuanxi | varchar | 50 | 否 | 院系 |
| zhuanye | varchar | 50 | 否 | 专业 |
| nianji | varchar | 50 | 否 | 年级 |
| banji | varchar | 50 | 否 | 班级 |

表 3-9 dic\_courseType 课程类型字典表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | 主键 |
| courseType | varchar | 50 | 否 | 课程类型 |
| remark | varchar | 255 | 否 | 备注 |

表 3-10 dic\_classroom 教室属性字典表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | 主键ID |
| address | varchar | 255 | 否 | 地点 |
| floorPie | varchar | 255 | 否 | 楼号 |
| seatCount | varchar | 50 | 否 | 座位总数 |
| remark | varchar | 255 | 否 | 备注 |

表 3-11 tb\_pjRecord 评教内容表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键 | 备注 |
| id | varchar | 50 | 是 | 主键Id |
| stuNum | varchar | 50 | 否 | 学号 |

续表 3-11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| courseId | varchar | | 50 | 否 | 课程ID |
| contextId | varchar | | 50 | 否 | 评教ID |
| marking | | varchar | 50 | 否 | 打分 |
| reamrk | | varchar | 255 | 否 | 备注 |

# 系统实现

## 系统实现概述

本系统的用户为高校管理者，教师和学生，系统设计了登录模块，当用户登录时进行身份验证和授权，当用户为管理者时自动跳转至管理者首页，为教师时跳转教师用户，为学生时跳转学生用户。

系统的管理用户负责管理系统的字典数据，包括课程、班级、课程类别、评教类别，负责初始化评教内容，维护学生和教师用户信息。可对用户进行新增、编辑和删除等操作。在统计分析模块能够查看学生对教师的评价信息；

系统的教师用户可以维护自己的基本信息、选择自己教授的课程、查看课程下参与的学生信息。

系统的学生用户可以维护自己的基本信息、修改密码、选择课程及对课程进行评教。

## 系统模块实现

### 登录模块

系统的安全性在软件设计中至关重要，本系统的登录认证模块集成了Apache Shiro。后台通过继承和实现Shiro的AuthorizingRealm抽象类对用户进行身份验证。登录页如图 4-1 登录页所示。



图 4-1 登录页

输入的用户名密码和验证码到了后台，首先对验证码进行验证，如果错误直接返回，验证码通过再将用户名和密码封装为Shiro的UsernamePasswordToken调用Subject的login方法，由Shiro对用户进行认证如果用户名不存在返回账号不存在，密码错误返回密码输入错误。

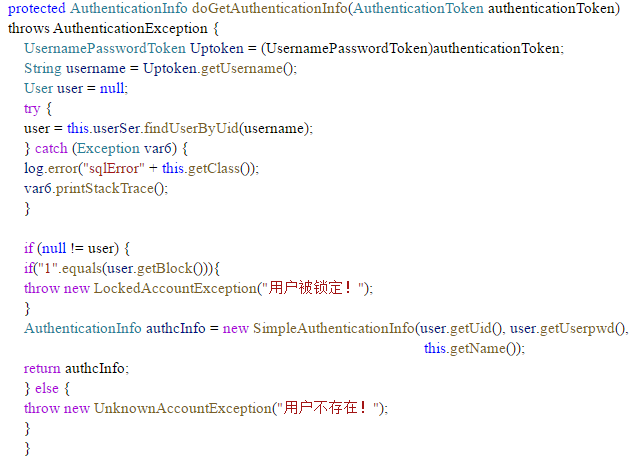
登录认证模块的后台实现代码如下所示:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_LoginController.java:**



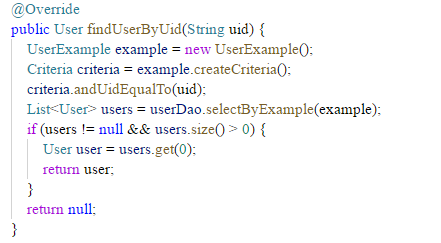
Shiro AuthorizingRealm的关键代码实现如下所示:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MyRealm.java:**



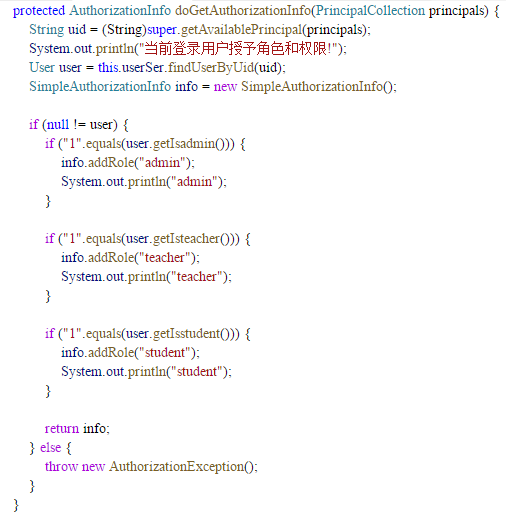
findUserByUid的实现

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_UserSerImpl.java:**



当认证通过后，根据用户的角色权限进入相应的后台首页

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MyRealm.java:**



跳转

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_LoginController.java:**



### 管理员模块

管理员后台分为：用户管理、课程管理、评教管理、信息查询菜单，菜单下细分具体模块。

用户管理页面如图 4-2所示：

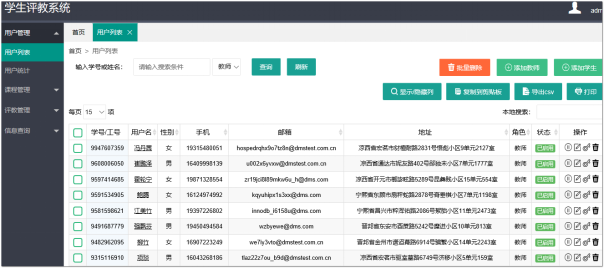


图 4-2 用户列表

（1）用户管理模块：

添加用户：

点击添加教师或点击添加学生用户，页面弹出添加用户的窗口，添加相关信息，点击提交按钮，保存新用户。如图4-3所示：

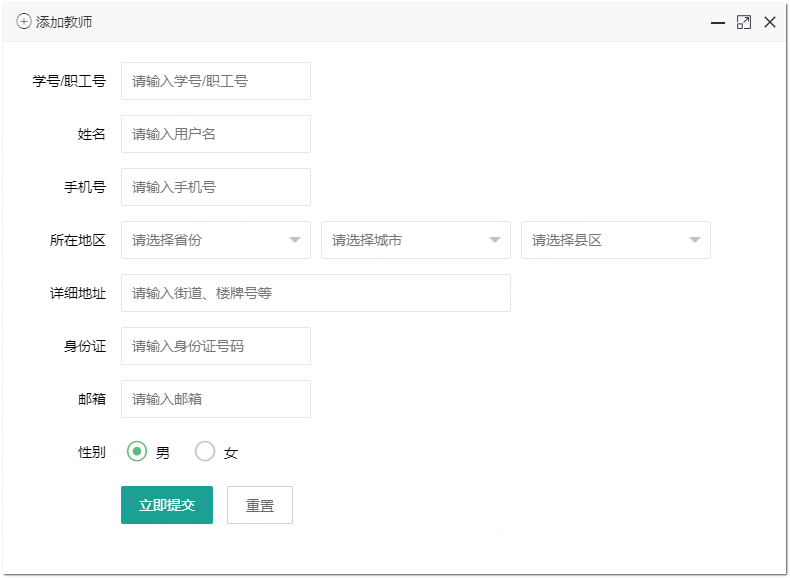


图 4-3 添加用户页面

账户管理：

在用户列表末端的操作列可对用户进行账户的启用和停用，修改密码，编辑删除操作

账户锁定与启动，点击操作列的按钮弹出提示框。如 图 4-4 停启用账户：

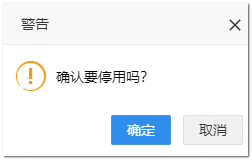


图 4-4 停启用账户

点击确定:用户列表将刷新修改用户锁定状态。如图 4-5 删除用户：

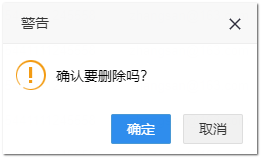


图 4-5 删除用户

点击确定:用户列表将刷新该用户将被删除。

1. 课程管理模块

课程管理页面可以对开设的课程进行维护，可以删除和添加课程。系统首先列出已有的课程可以对其编辑删除操作，通过点击添加课程按钮，添加新的课程。

课程分为艺术实践类，理论类，理论+实践类，在添加课程时课程将按此进行分类。课程将带有此属性。在学生评教模块将根据课程的此属性来生成不同的评教问卷。

课程管理列表如图 4-6所示：



图 4-6 课程管理页面

添加课程功能如图 4-7所示：

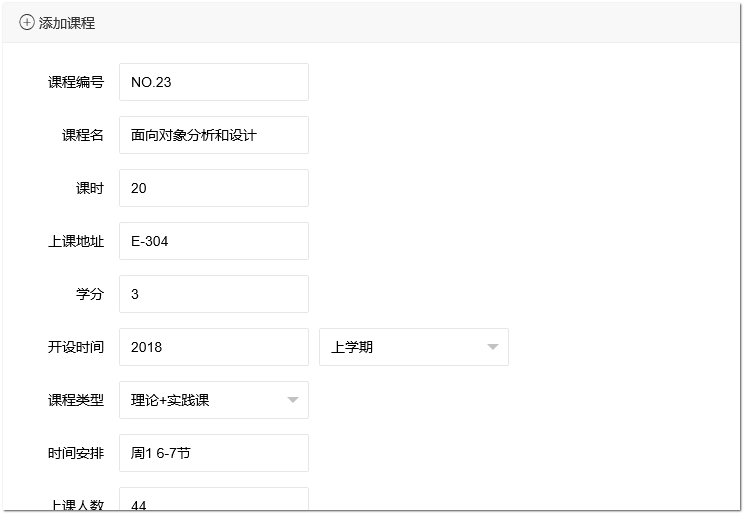


图 4-7 添加课程

1. 评教管理模块

该模块管理员可以对预设的评教内容进行维护可以新增和修改评价内容。添加的评教内容也将围绕实践，理论，理论+实践进行分类的评教内容。评教列表管理页如图 4-8所示：

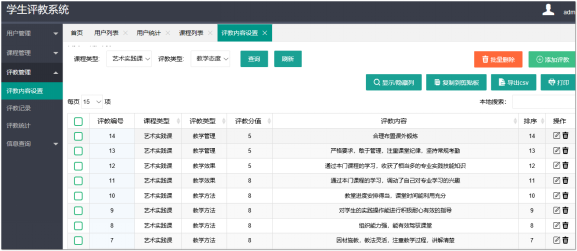


图 4-8 评教管理页面

点击添加评教按钮，弹出添加评教表单回话框。如图 4-9所示：



图 4-9 添加评教题目

管理员可以查看学生的评教记录，可通过多种图表多维度了解评教工作的情况如图 4-10 评教数据分析：

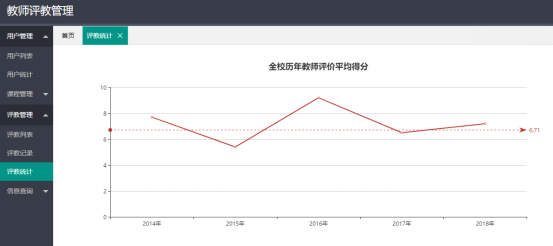




图 4-10 评教数据分析

### 教师端

1. 维护个人信息

教师可以重新修改自己的基础信息，完善信息，修改密码，不可更改。编辑个人信息页面如图 4-11所示：

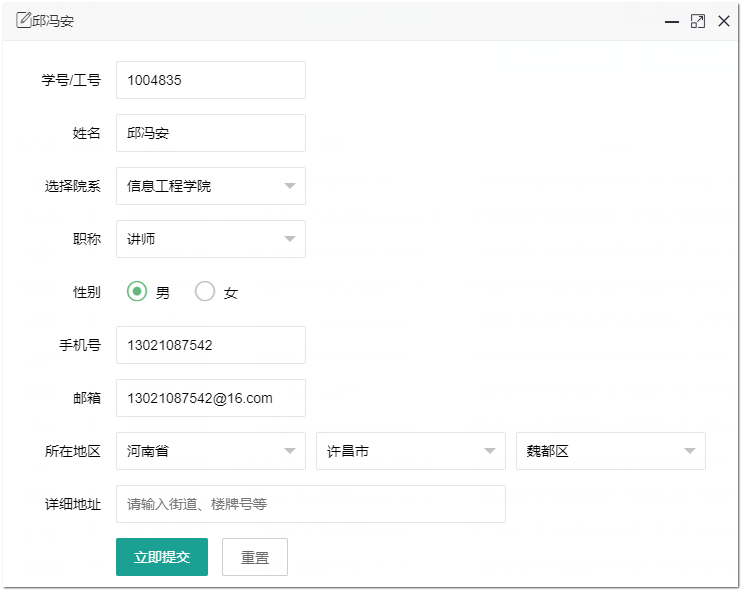


图 4-11 编辑个人信息

1. 教师选课

教师通过选择系统给出的课程来选择教授一门课程，教师选课页面如图 4-12所示：



图 4-12 教师选课

点击我的课程列出自己任教的课程。如图 4-13所示：



图 4-13 我的课程

### 学生端

1. 维护个人信息

在修改个人信息时关键信息只作为显示，输入框中添加禁用属性如学生/职工号为只读项，不可更改，用户可以对自己的基础信息进行维护。

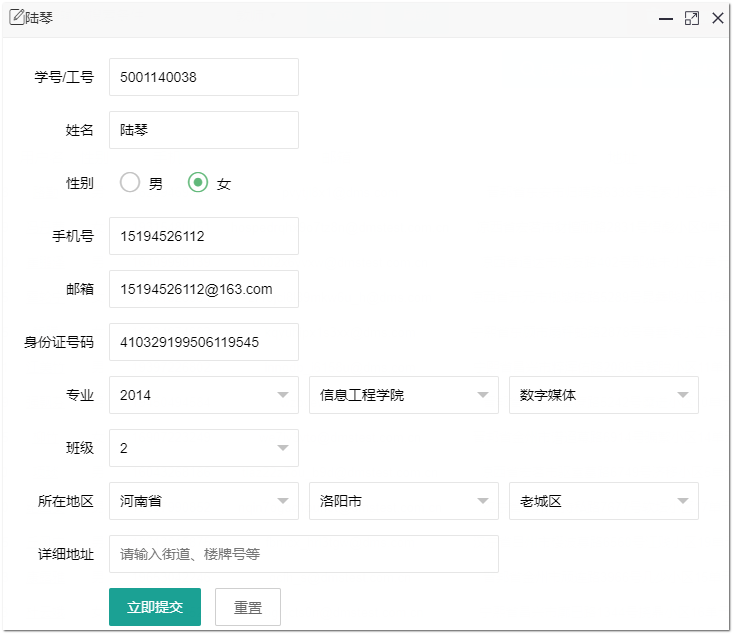


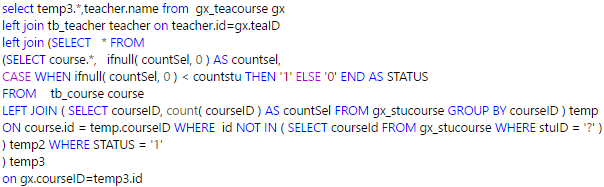
图 4-14 编辑个人信息

1. 学生选课，列出可供选择的课程可按课程编号或课程名查询，可筛选有余量的课进行选课。课程列表见图 4-15 学生选课



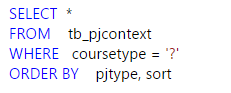
图 4-15 学生选课

在筛选余课的查询时系统首先选取有效的课程（教师已选课程）根据当前登录的用户排除自己已选的课程，排除余量为0的课程。查询实现sql如下：



1. 学生评教

学生可对已选的教师课程进行评教，评教时系统根据学生选择的课程属性查询出对应的评教项目，形成问卷。Sql关键语句如下：



学生评教后，评教记录将保存至数据库，供管理员查看管理，组织数据生成分析报告，评教页面见 图 4-16 学生评教



图 4-16 学生评教

评教页面实现代码

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pingjiao-list.jsp:





# 系统测试

## 概述

软件的稳定性和可用性对于用户体验十分重要，软件能否是否达到了预期要求，能否满足用户需求，软件测试环节必不可少。软件测试时可针对安全模块，系统核心模块进行重点测试。测试时使用的测试方法，测试数据要有多元的考虑，满足和覆盖系统上线后可能遇到的所有情况，通过测试的软件在后期也应经过跟踪试运行时期。本系统针对系统的安全模块和核心模块进行测试。

## 测试方法

考虑到系统的特点，决定对本系统进行黑盒测试方法，前台至后台的输入交互中可采用多种数据形式输入，由此来监测系统的响应情况，看是否满足预期需求，以此来确定该模块的可用性，通过输入非法的，常见的具有攻击性注入性的数据来监测模块的安全可靠性。

## 测试用例

### 系统的登录模块测试用例

表 5-1 系统登录模块测试用例

|  |  |  |
| --- | --- | --- |
| 输入/动作 | 期望的输出或响应 | 实际情况 |
| 不输入账户 | 提示用户名不能为空 | 提示用户名不能空 |
| 不输入密码 | 提示输入密码 | 提示输入密码 |
| 输入不存在的账户 | 提示账户不存在 | 提示账户不存在 |
| 输入错误的密码 | 提示密码错误 | 提示密码错误 |
| 输入错误的验证码 | 提示验证码错误 | 提示验证码错误 |
| 全部输入正确 | 页面跳转至首页 | 页面跳转至首页 |

在登录模块中，系统设计了多种数据校验方式，提供了多样的交互提示信息。只有以正确的方式进行登录操作,用户才能登陆成功，非法的及残缺的操作都进行了处理设计。

### 系统教学评教模块测试

表 5-2 系统教学评教模块测试用例

|  |  |  |
| --- | --- | --- |
| 输入/动作 | 期望的输出或响应 | 实际情况 |
| 评教表单未完成时提交 | 显示未完成题号 | 显示未完成题号 |

续表 5-2

|  |  |  |
| --- | --- | --- |
| 评教完成后提交 | 提示提交成功 | 提示提交成功 |
| 对同一表单多次提交 | 提示更改成功 | 提示更改成功 |

评教表单的提交校验能够保证评教表单的完整性，确保存入数据库的数据具有完整性。

### 查询功能测试

表 5-3 查询功能测试

|  |  |  |
| --- | --- | --- |
| 输入/动作 | 期望的输出或响应 | 实际情况 |
| 单条件查询 | 返回正确数据 | 返回正确数据 |
| 组合条件查询 | 返回正确数据 | 返回正确数据 |
| 模糊查询 | 返回正确数据 | 返回正确数据 |

系统中的查询功能支持组合查询，模糊查询，单条件查询多种方式。

### 表单检验测试

表 5-4 添加学生表单检验测试

|  |  |  |
| --- | --- | --- |
| 输入/动作 | 期望的输出或响应 | 实际情况 |
| 学号为空 | 提示学号不能为空 | 提示学号不能为空 |
| 姓名为空 | 提示姓名不能为空 | 提示姓名不能为空 |
| 电话格式不规范 | 提示输入正确的格式 | 提示输入正确的格式 |
| 邮箱格式不规范 | 提示输入正确的格式 | 提示输入正确的格式 |
| 身份证格式不规范 | 提示输入正确的格式 | 提示输入正确的格式 |

对表单数据的校验纠正使得存入的数据规范，完整。用户输入操作时能够对其非法的和不规范的内容进行拦截确保了存入数据有效性和可用性。

### 用户维护测试

表 5-5 用户管理功能测试

|  |  |  |
| --- | --- | --- |
| 输入/动作 | 期望的输出或响应 | 实际情况 |
| 对用户进行锁定 | 该用户不能进行登录 | 该用户不能进行登录 |
| 对用户进行秘密重置 | 用户秘密被重置 | 用户秘密被重置 |

续表 5-5

|  |  |  |
| --- | --- | --- |
| 删除用户 | 用户信息从数据库删除 | 用户信息从数据库删除 |
| 邮箱格式不规范 | 提示输入正确的格式 | 提示输入正确的格式 |
| 身份证格式不规范 | 提示输入正确的格式 | 提示输入正确的格式 |

# 参考文献

1. 布洛克（Joshua, Bloch）. Effective Java[M]. 北京:机械工业出版社, 2009.
2. 王养廷. 基于Web的Java框架设计与实现[J]. 软件导刊(教育技术), 2014, 13(1):58-59.
3. 缪忠剑. 基于Spring的集成化Web开发平台的研究与实现[M]．北京：机械工业出版社，2013.
4. 赵利庆．Java Web架构中数据库优化模式的研究与实现[D]．北京：北京邮电大学，2015．
5. 库俊国. 基于J2EE技术的Web应用体系研究及实践[M]．北京：人民邮电出版社，2014.
6. 衣鹏, 阎新芳, 吴帅杰. Java设计模式在数据库编程中的应用研究[J]. 信息通信, 2014(11):118-118.
7. 王艳丽. 基于J2EE平台Web系统的软件测试研究和实现[M]．北京：机械工业出版社，2012.
8. 李运莉. web数据库应用系统性能优化[M]．北京：人民邮电出版社，2011.
9. 翁云翔. Java安全框架Shiro在Web中的研究与应用[D].武汉邮电科学研究院,2016.
10. Martin, RobertC. Clean code : a handbook of agile software craftsmanship = 代码整洁之道 : 敏捷软件开发技能手册 / 2nd[M]. 北京:人民邮电出版社, 2011.
11. 路丽娜,王洪才.质性评教:走出学生评教困境的理性选择[J].现代大学教育,2016(02):93-98.
12. 李楠. 我国高等学校教师绩效评价研究[D].首都经济贸易大学,2012.
13. 别敦荣,孟凡.论学生评教及高校教学质量保障体系的改善[J].高等教育研究,2007(12):77-83.
14. 王晓刚.学生评教数据的统计处理[J].今日科苑,2009(16):238-239.

# 附 录

部分程序源代码：

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MyRealm.java

package com.gym.web.shiro;

import com.gym.web.model.User;

import com.gym.web.service.UserSer;

import org.apache.shiro.authc.\*;

import org.apache.shiro.authz.AuthorizationException;

import org.apache.shiro.authz.AuthorizationInfo;

import org.apache.shiro.authz.SimpleAuthorizationInfo;

import org.apache.shiro.crypto.hash.SimpleHash;

import org.apache.shiro.realm.AuthorizingRealm;

import org.apache.shiro.subject.PrincipalCollection;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import javax.annotation.Resource;

public class MyRealm extends AuthorizingRealm {

private static final Logger log = LoggerFactory.getLogger(MyRealm.class);

@Resource

UserSer userSer;

protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken authenticationToken) throws AuthenticationException {

UsernamePasswordToken Uptoken = (UsernamePasswordToken)authenticationToken;

String username = Uptoken.getUsername();

User user = null;

try {

user = this.userSer.findUserByUid(username);

} catch (Exception var6) {

log.error("sqlError" + this.getClass());

var6.printStackTrace();

}

if (null != user) {

if("1".equals(user.getBlock())){

throw new LockedAccountException("用户被锁定！");

}

AuthenticationInfo authcInfo = new SimpleAuthenticationInfo(user.getUid(), user.getUserpwd(), this.getName());

return authcInfo;

} else {

throw new UnknownAccountException("用户不存在！");

}

}

protected AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principals) {

String uid = (String)super.getAvailablePrincipal(principals);

System.out.println("当前登录用户授予角色和权限!");

User user = this.userSer.findUserByUid(uid);

SimpleAuthorizationInfo info = new SimpleAuthorizationInfo();

if (null != user) {

if ("1".equals(user.getIsadmin())) {

info.addRole("admin");

System.out.println("admin");

}

if ("1".equals(user.getIsteacher())) {

info.addRole("teacher");

System.out.println("teacher");

}

if ("1".equals(user.getIsstudent())) {

info.addRole("student");

System.out.println("student");

}

return info;

} else {

throw new AuthorizationException();

}

}

public void onLogout(PrincipalCollection principals) {

System.out.println("用户退出:" + principals.toString());

super.onLogout(principals);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_LoginController.java

package com.gym.web.controller;

import com.gym.commons.helper.RandomValidateCodeHelper;

import com.gym.web.model.User;

import com.gym.web.service.UserSer;

import org.apache.shiro.SecurityUtils;

import org.apache.shiro.authc.IncorrectCredentialsException;

import org.apache.shiro.authc.LockedAccountException;

import org.apache.shiro.authc.UnknownAccountException;

import org.apache.shiro.authc.UsernamePasswordToken;

import org.apache.shiro.subject.Subject;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.util.HashMap;

import java.util.Map;

@Controller

public class LoginController {

@Autowired

private UserSer userSer;

private static transient final Logger log = LoggerFactory.getLogger(LoginController.class);

/\*\*

\* 登录页

\*

\* @return

\*/

@RequestMapping("/login.shtm")

public String login() {

Subject currentUser=SecurityUtils.getSubject();

if(currentUser.isAuthenticated()){

return "redirect:/index.shtm";

}

return "login";

}

/\*\*

\* 登录验证

\*

\* @return

\*/

@ResponseBody

@RequestMapping("/authLogin.chtm")

public Map<String, Object> authLogin(HttpServletRequest request, HttpServletResponse response, User userao) {

Map<String, Object> result = new HashMap<>();

result.put("success", false);

//1：获取页面用户名和密码，验证码

//2：比对验证码

//3：创建subject

//4: 登录

String username = request.getParameter("username");//1

String password = request.getParameter("password");

String inputCode = request.getParameter("validateCode");

String yCode = request.getSession().getAttribute("RANDOMVALIDATECODEKEY").toString();//2

if (!(yCode != null && yCode.equals(inputCode))) {

result.put("msg", "验证码输入错误！");

result.put("status", "validateCode");

return result;

}

Subject currentUser = SecurityUtils.getSubject();

if (!currentUser.isAuthenticated()) {

UsernamePasswordToken upToken = new UsernamePasswordToken(username, password);

try {

//调用shiro验证登录信息

currentUser.login(upToken);

result.put("status", true);

result.put("success", true);

return result;

} catch (UnknownAccountException e) {

result.put("status", "username");

result.put("msg", "账号不存在");

} catch (IncorrectCredentialsException e) {

result.put("status", "password");

result.put("msg", "密码输入错误");

} catch (LockedAccountException e) {

result.put("msg", "用户被锁定");

}

} else {

result.put("success", true);

result.put("status", true);

}

return result;

}

/\*\*

\*

\* @param model

\* @param request

\* @param response

\* @return

\*/

@RequestMapping("/index.shtm")

public String index(Model model, HttpServletRequest request, HttpServletResponse response) {

Subject currentUser = SecurityUtils.getSubject();

if (currentUser.hasRole("admin")) {

return "redirect:/admin.shtm";

}

if (currentUser.hasRole("teacher")) {

return "redirect:/admin.shtm";

}

if (currentUser.hasRole("student")) {

return "redirect:/admin.shtm";

}

model.addAttribute("errorMsg", "您暂无权限");

return "error/error";

}

/\*\*

\* 用于生成验证码

\*

\* @param req

\* @param response

\* @return

\*/

@RequestMapping("/randomValidateCode.chtm")

public void randomValidateCode(HttpServletRequest req, HttpServletResponse response) {

response.setContentType("image/jpeg");//设置相应类型,告诉浏览器输出的内容为图片

response.setHeader("Pragma", "No-cache");//设置响应头信息，告诉浏览器不要缓存此内容

response.setHeader("Cache-Control", "no-cache");

response.setDateHeader("Expire", 0);

RandomValidateCodeHelper randomValidateCode = new RandomValidateCodeHelper();

try {

randomValidateCode.getRandcode(req, response);//输出图片方法

} catch (Exception e) {

e.printStackTrace();

}

}

@RequestMapping("/admin.shtm")

public String majorLeader(HttpServletRequest request, HttpServletResponse response) {

return "decorators/admin";

}

@RequestMapping("/teacher.shtm")

public String zdTeacher(HttpServletRequest request, HttpServletResponse response) {

return "decorators/teacher";

}

@RequestMapping("/student.shtm")

public String secretary(HttpServletRequest request, HttpServletResponse response) {

return "decorators/student";

}

@RequestMapping("/homePage.shtm")

public String homePage(HttpServletRequest request, HttpServletResponse response) {

return "Public\_Page/homePage.jsp";

}

/\*\*

\* 登出

\* @return

\*/

@RequestMapping("/loginout.shtm")

public String loginout() {

Subject currentUser = SecurityUtils.getSubject();

if (currentUser.isAuthenticated()) {

currentUser.logout();

}

return "redirect:/login.shtm";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_UserController.java

package com.gym.web.controller;

import com.alibaba.fastjson.JSON;

import com.alibaba.fastjson.JSONArray;

import com.alibaba.fastjson.JSONObject;

import com.gym.commons.helper.QueryParmFormat;

import com.gym.web.model.Student;

import com.gym.web.model.Teacher;

import com.gym.web.model.User;

import com.gym.web.service.ScopeDicSer;

import com.gym.web.service.StudentSer;

import com.gym.web.service.TeacherSer;

import com.gym.web.service.UserSer;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.annotation.Scope;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import javax.servlet.http.HttpServletRequest;

import java.util.List;

import java.util.Map;

@Scope("prototype")

@Controller

@RequestMapping("/admin/user")

public class UserController {

@Autowired

UserSer userSer;

@Autowired

TeacherSer teacherSer;

@Autowired

StudentSer studentSer;

@Autowired

ScopeDicSer scopeDicSer;

JSONArray jsonArray;

/\*\*

\* 用户列表-view

\*

\* @return

\*/

@RequestMapping("/userList.shtm")

public String userList() {

return "admin/user/user-list";

}

/\*\*

\* 获取所有用户

\*

\* @param req

\* @return

\*/

@RequestMapping("/getUser.shtm")

@ResponseBody

public JSONObject getUsers(HttpServletRequest req) {

Map<String, String> querymap = QueryParmFormat.Format(req.getParameterMap());

JSONObject jsonObject = new JSONObject();

if ("teacher".equals(querymap.get("role"))) {

List<Map<String, Object>> teacherList = teacherSer.findTeacher(querymap);

jsonArray = JSONArray.parseArray(JSON.toJSONString(teacherList));

jsonObject.put("users", jsonArray);

System.out.println(jsonObject);

return jsonObject;

}

if ("student".equals(querymap.get("role"))) {

List<Map<String, Object>> studentList = studentSer.findStudent(querymap);

jsonArray = JSONArray.parseArray(JSON.toJSONString(studentList));

jsonObject.put("users", jsonArray);

System.out.println(jsonObject);

return jsonObject;

}

return null;

}

@RequestMapping("/user-add.shtm")

public String addUser(HttpServletRequest request) {

String role = request.getParameter("role");

request.setAttribute("role", role);

if ("teacher".equals(role)) {

//院系，职称，

JSONArray yuanxis = scopeDicSer.getYuanxi();

request.setAttribute("yuanxis", yuanxis);

}

if ("student".equals(role)) {

JSONArray nianjis = scopeDicSer.getNianji();

JSONArray yuanxis = scopeDicSer.getYuanxi();

JSONArray banjis = scopeDicSer.getAll();

JSONArray zhuanyes = scopeDicSer.getZhuanye();

request.setAttribute("nianjis", nianjis);

request.setAttribute("yuanxis", yuanxis);

request.setAttribute("zhuanyes", zhuanyes);

request.setAttribute("banjis", banjis);

}

return "admin/user/user-add";

}

@RequestMapping("/adduser.shtm")

@ResponseBody

public JSONObject saveuser(HttpServletRequest req, Teacher teacher) {

JSONObject jsonObject = new JSONObject();

Map<String, String> pqramMap = QueryParmFormat.Format(req.getParameterMap());

int userCount = userSer.countUserByUid(pqramMap.get("uid"));

if ("teacher".equals(pqramMap.get("role"))) {

if (userCount > 0) {//用户已存在,进行更新

//更新用户

//更新教师

userSer.updateUser(pqramMap);

teacherSer.updateTeacher(pqramMap);

jsonObject.put("msg", "用户已更新");

jsonObject.put("status", "success");

return jsonObject;

} else {//添加教师

int i = userSer.addUser(pqramMap);

int j = teacherSer.addTeacher(pqramMap);

jsonObject.put("msg", "添加教师成功");

jsonObject.put("status", "success");

return jsonObject;

}

}

if ("student".equals(pqramMap.get("role"))) {

if (userCount > 0) {//用户已存在,进行更新

userSer.updateUser(pqramMap);

studentSer.updateStudent(pqramMap);

req.setAttribute("msg", "用户已更新");

jsonObject.put("status", "success");

return jsonObject;

} else {//添加学生

int i = userSer.addUser(pqramMap);

int j = studentSer.addStudent(pqramMap);

jsonObject.put("msg", "添加学生成功");

jsonObject.put("status", "success");

return jsonObject;

}

}

return jsonObject;

}

@RequestMapping("/blockuser.shtm")

@ResponseBody

public JSONObject blockUser(HttpServletRequest request) {

JSONObject jsonObject = new JSONObject();

String useruid = request.getParameter("useruid");

String status = request.getParameter("status");

int i = userSer.blockuser(useruid, status);

if (i > 0) {

jsonObject.put("status", "success");

return jsonObject;

}

return jsonObject;

}

@RequestMapping("/user-edit.shtm")

public String editUser(HttpServletRequest request) {

String uid = request.getParameter("uid");

User user = userSer.findUserByUid(uid);

if ("teacher".equals(user.getRole())) {

JSONArray yuanxis = scopeDicSer.getYuanxi();

Teacher teacher = teacherSer.findByPrimaryKey(uid);

request.setAttribute("teacher", teacher);

request.setAttribute("yuanxis", yuanxis);

}

if ("student".equals(user.getRole())) {

JSONArray yuanxis = scopeDicSer.getYuanxi();

JSONArray zhuanyes = scopeDicSer.getZhuanye();

JSONArray banjis = scopeDicSer.getAll();

Student student = studentSer.findByPrimaryKey(uid);

request.setAttribute("student", student);

request.setAttribute("yuanxis", yuanxis);

request.setAttribute("zhuanyes", zhuanyes);

request.setAttribute("banjis", banjis);

}

return "admin/user/user-edit";

}

@RequestMapping("/resetpwd.shtm")

@ResponseBody

public JSONObject resetpwd(HttpServletRequest request) {

JSONObject jsonObject = new JSONObject();

String useruid = request.getParameter("useruid");

int i = userSer.resetpwd(useruid);

if (i > 0) {

jsonObject.put("status", "success");

return jsonObject;

}

return jsonObject;

}

@RequestMapping("/deluser.shtm")

@ResponseBody

public JSONObject deluser(HttpServletRequest request) {

JSONObject jsonObject = new JSONObject();

String useruid = request.getParameter("useruid");

int i = userSer.deluserByUid(useruid);

if (i > 0) {

jsonObject.put("status", "success");

return jsonObject;

}

return jsonObject;

}

}

# 致 谢

转瞬即逝，四年的紧张而充实的大学生生活即将结束。在四年的学习中，我得到了很多老师、同学和朋友的关心和帮助。在完成我的毕业论文时，我要对所有在这期间给予我帮助和鼓励的人表示衷心的感谢。

首先，我要感谢我的导师王亚老师。从选题、构思、写作到论文的最终定稿，王老师都给了我很好的指导和热情的帮助，使我的毕业论文顺利完成。在刚开始选题到课题设计系统的一点点实现，王老师都给予了莫大的帮助。提供了很多可参考的文献，网站。让我少走很多弯路。在批改初稿，二稿至终稿的过程中王老师都认真负责，给我指正了很多细节问题。王老师对待学生毕设工作认真负责，学术精神严谨，值得我终身学习。

其次，多亏了信息工程学院的领导和老师，由于他们认真的教学，这四年让我学到了很多计算机科学的专业知识，掌握了扎实的专业技能。，这也是我能顺利完成毕业设计的基础。

最后，我要感谢我的家人对我的宽容、关怀和鼓励，以及所有与我一起学习的同学和朋友。正是因为他们的支持和关心，我才能顺利地学习和完成学业。

毕业后，在今后的工作和生活中，我会记住老师的教诲，继续不懈的努力和追求，报答所有的支持和帮助我的人！