

**本科毕业设计(论文)**

**FINAL PROJECT/THESIS OF UNDERGRADUATE**

**(2018 届)**

**学生选课管理信息系统设计与实现**

**Design and Implementation of Students Academic Affairs Center**

|  |  |
| --- | --- |
| **学　　院** | 光电信息与计算机工程学院 |
| **专　　业** | 计算机科学与技术 |
| **学生姓名** | 王启帆 |
| **学　　号** | 1412480521 |
| **指导教师** | 刘亚 |
| **完成日期** | 2018年05月 |

# 承诺书

本人郑重承诺：所呈交的毕业论文“信息安全课程在线系统设计与实现”是在导师的指导下，严格按照学校和学院的有关规定由本人独立完成。文中所引用的观点和参考资料均已标注并加以注释。论文研究过程中不存在抄袭他人研究成果和伪造相关数据等行为。如若出现任何侵犯他人知识产权等问题，本人愿意承担相关法律责任。

承诺人(签名)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

日期： 年 月 日

# 摘 要

针对现在多数大学采取选课制度，授课教师和学生缺乏紧密联系，本次毕业设计完成一个与其有一定差别的学生选课信息系统。该系统提供老师和学生交流的平台，让学生和教师随时随地的进行沟通交流，排除地理位置的限制和固定答疑时间的干扰，让沟通变得更加简单方便。对于教师来说，这是一个明确教学进度，方便教学管理的平台。教师可以编辑课程信息，布置作业，上传课件，管理报名参加这门课的学生信息，查看并下载学生上传的作业，回复学生留言并在线答疑。对于学生来说，这是一个跟进课程信息，解决学习困难的平台。学生可以在此浏览并下载课程资料，上传作业，留言，在线提问等。对于课程管理人员来说，这是一个方便管理学生、教师和课程信息的平台。课程管理人员可以在此查看、新增或修改学生和老师的信息，管理留言及评论。

本系统设计结合ASP.NET开发框架，实现基本UI界面的搭建，利用框架自带的控件和功能完成部分基本功能。使用MVC架构，数据层，控制层，视图层分层明确，加强系统的代码可读性和后期维护性。结合关系型数据库MySQL InnoDB引擎存储数据，提供高性能的数据访问。另外还使用Json传值实现页面间的数据交互，格式简单，易于读写。使用Ajax实现在线聊天室功能，支持客户端和服务器端的双向通信。

**关键词**：选课、课程管理系统、ASP.NET、MVC、Ajax

# ABSTRACT

In view of the elective system which most university uses nowadays and the lack of close contact between teachers and students, this final project is mainly designed to complete an information security course online system. This system will provide a great platform for them to discuss about courses and solve students’ questions. It makes communication easier without the limitation of location and time between teachers and students. For teachers, it’s a platform where they can write their teaching agenda and manage students in the system. They can edit course information, assign assignments, upload courseware, and manage all the students who have assigned their courses. For students, it’s a platform where they can follow-up course updates and solve confusions. They can browse and download course materials, upload homework, leave messages or comments and contact teacher online. For course managers, it’s a platform where they can manage students, teachers and courses. They can browse, add, or modified teachers’ or students’ information, manage comments and messages.

The system combined with DWZ domestic open source framework to build the basic UI interface. And it uses the original functions of framework to complete some of the basic functions. By using MVC, model, controller and view level is clearer, which strongly enhances readability and post-maintenance for the code. Combined with relational database SQL Server to store data, the system provides a high-performance data access. In addition, the usage of Json to achieve data exchange between pages makes the data format is more simple and easier to read and write. Using websocket to achieve chat room applications can support the communication between client and server.

**KEY WORDS**: online course system Java Web MVC

# 目 录

**摘要**

**ABSTRACT**

[第1章 绪论 1](#_Toc483344113)

[1.1 背景 1](#_Toc483344114)

[1.2 学生选课管理信息系统的优点 1](#_Toc483344115)

[1.3 系统实现的目的与意义 1](#_Toc483344116)

[1.4 本论文的主要内容与结构 2](#_Toc483344117)

[第2章 开发使用技术简介 3](#_Toc483344118)

[2.1 ASP.NET框架与MVC模式 3](#_Toc483344119)

[2.2 jQuery与Bootstrap 3](#_Toc483344120)

[2.3 AJAX与JSON 4](#_Toc483344121)

[2.4 开发工具 4](#_Toc483344122)

[2.4.1 MySQL Workbench工具简介 4](#_Toc483344123)

[2.4.2 Visual Studio 2017 Community工具简介 4](#_Toc483344124)

[第3章 需求分析 5](#_Toc483344125)

[3.1 系统需求及设计思路 5](#_Toc483344126)

[3.1.1系统需求 5](#_Toc483344127)

[3.1.2系统设计思路 5](#_Toc483344128)

[3.2系统需求分析 5](#_Toc483344129)

[3.2.1功能需求分析 5](#_Toc483344130)

[3.2.3数据库需求分析 9](#_Toc483344131)

[3.2.4性能需求分析 10](#_Toc483344132)

[3.3 可行性分析 10](#_Toc483344133)

[3.3.1 设计目标 10](#_Toc483344134)

[3.3.2技术可行性 10](#_Toc483344135)

[3.3.3经济可行性 10](#_Toc483344136)

[3.3.4操作可行性 11](#_Toc483344137)

[第4章 系统概要设计 13](#_Toc483344138)

[4.1 系统结构图 13](#_Toc483344139)

[4.2数据库设计 13](#_Toc483344140)

[4.2.1数据库详细设计 13](#_Toc483344141)

[第5章 详细设计 17](#_Toc483344142)

[5.1 页面设计 17](#_Toc483344143)

[5.1.1登录/注册页面 17](#_Toc483344144)

[5.1.2管理员管理教师/学生 17](#_Toc483344145)

[5.1.3管理员管理课程 18](#_Toc483344146)

[5.1.4教师管理课程 18](#_Toc483344147)

[5.1.5教师/学生互相评论 18](#_Toc483344148)

[5.1.6教师管理学生 19](#_Toc483344149)

[5.1.7学生选择/收藏课程 19](#_Toc483344150)

[5.1.8教师/学生修改个人信息 20](#_Toc483344151)

[5.1.9教师/学生私信互动 20](#_Toc483344152)

[5.2 实体类设计 20](#_Toc483344153)

[5.2.1用户类设计 20](#_Toc483344154)

[5.2.2课程类设计 21](#_Toc483344155)

[5.2.3课表类设计 21](#_Toc483344156)

[5.2.4教室类设计 21](#_Toc483344157)

[5.2.5考试类设计 22](#_Toc483344158)

[5.2.6学科类设计 22](#_Toc483344159)

[5.2.7教室占用类设计 23](#_Toc483344160)

[第6章 软件的实现与测试 25](#_Toc483344161)

[6.1 软件界面实现 25](#_Toc483344162)

[6.1.1用户登录/注册界面 25](#_Toc483344163)

[6.1.2用户首页展示界面 25](#_Toc483344164)

[6.1.3教师管理课程信息界面 26](#_Toc483344165)

[6.1.4用户管理课表信息界面 27](#_Toc483344166)

[6.1.5查看当前所有课程界面 28](#_Toc483344167)

[6.1.6收藏/选择/评论课程界面 28](#_Toc483344168)

[6.1.7查看所有教师/学生信息界面 28](#_Toc483344169)

[6.1.8查看指定教师/学生信息界面 29](#_Toc483344170)

[6.1.9教师/学生发送私信界面 29](#_Toc483344171)

[6.1.10教师/学生事务中心页面 30](#_Toc483344172)

[6.1.11四六级/图书借还信息界面 31](#_Toc483344173)

[6.1.12管理员管理课程界面 31](#_Toc483344174)

[6.2 测试环境和测试工具 33](#_Toc483344175)

[6.3测试用例 33](#_Toc483344176)

[6.3 测试结论 41](#_Toc483344177)

[第7章 结束语 43](#_Toc483344178)

[7.1 工作总结 43](#_Toc483344179)

[7.2 需改进的点 43](#_Toc483344180)

[参考文献 44](#_Toc483344181)

[致 谢 45](#_Toc483344182)

[附录A: 主要源代码 46](#_Toc483344183)

[附录B: 软件使用说明书 59](#_Toc483344184)

[1.软件描述 59](#_Toc483344185)

[1.1 系统主要功能 59](#_Toc483344186)

[2.软件的安装 59](#_Toc483344187)

[2.1 系统环境需求 59](#_Toc483344188)

[2.2 使用说明 59](#_Toc483344189)

# 第1章 绪论

## 1.1 背景

由于现在高校多数采取选课制度，而经过了许多年，教务管理系统的发展并没有跟上互联网浪潮向前的脚步——Bootstrap响应式布局系统。过去的许多网站，因为历史遗留性，其渐渐变得只适合在PC端浏览，而对于现在非常流行的平板电脑和智能手机，它们的支持还不够。而教务管理系统就是这样一种具有历史遗留性的网站，它包含了庞大的后台数据库，而前端页面在手机和平板电脑上难以适配，这使得大量数据在嵌入式操作系统的浏览器中显得非常冗余和杂乱，而使用当前最新的网站栅格布局系统，在Windows操作系统上进行开发，使用栅格系统布局，则在其他所有类型的设备上都可以进行访问，它会自适应屏幕的尺寸并对相关内容进行排版，然后再进行显示。

除此（Bootstrap栅格布局）带来的方便之外，本学生选课管理信息系统还具有优化的前端，将最复杂的信息以最简单的形式展示出来，Glyphicon和Font-Awesome前端插件使得展示大量信息达到新的水平，其界面美观自然，信息内容充分又不冗余，且一目了然。

与旧版系统（即以往的教务管理系统）相比，除了实现对教师发布的课程完成编排和学生选课的进行实现，它的功能更多地侧重了学生和教师的互动，学生和老师可以互相评价（访问个人主页之后才能评价）。。。学生和老师之间的联系更偏向于多对多的灵活的随机组合，导致了教师和学生并没有长时间固定交流的客观条件，相比初高中沟通相对缺乏。针对这种现状，近日上海市教委出台了教师坐班答疑的规定。规定教师必须坐班答疑，教授、副教授每周坐班答疑的时间不少于8小时或一天；讲师不少于24小时或两天，助教不少于48小时或4天。

然而尽管如此，依旧会因为时间空间上的限制导致答疑不能覆盖到每一个学生及老师的需求。这种坐班答疑的制度会导致在答疑时间段内有其他课程的安排的学生错过答疑的机会，也使得部分老师因为没有前来答疑的学生浪费大量宝贵的时间。因为这些种种客观原因的存在，导致坐班答疑没有达到最初的目标，在给学生和老师提供交流机会这个目的上起到的作用有限。

由此，本次毕业设计将实现完成一个信息安全课程在线系统，这个系统旨在设计一个能够使得学生和老师能够充分利用互联网实现答疑以及跟踪课程，方便学生咨询，便于老师回复的平台。

## 1.2 学生选课管理信息系统的优点

学生选课管理信息系统是专为老师的课程编排和学生课表安排而设计的信息系统。其中包括了发布课程和管理课程功能，主要是为了解决教学中在某一时间段遇到的课程冲突或者教学地点冲突等问题。。。比如说教学资源共享，师资力量不足等问题。通过借助网络技术从而构建虚拟教学环境可以补充多样的数字化共享资源，实现自动化和智能化教学，弥补传统课堂的不足。

网络技术作为教学辅助工具，可以充分有效的支持教学的各个环节，除了考虑系统的操作便捷性，界面友好性，还要注重学生间、师生间、学生和资源间的交互，强调教师教学活动设计功能的支持，提倡合作学习的理念。。。在网络教学中，学生是教学活动的主体，教师是活动的设计者，起到引导监督作用，它打破传统方式的局限，以灵活多样的教学组织形式，组织丰富有效的教学活动，调动学生学习积极性，提高老师教学高效性。

## 1.3 系统实现的目的与意义

本次信息安全课程在线系统的目的在于提供学生和老师一个相互沟通交流的平台，希望借由这个平台上学生可以自由的提问及留言，发表自己的疑惑或者看法，也可以查看课件及上传作业；而老师也可以在这里管理自己的课程，与学生进行及时有效的沟通，从而更好的安排后续课程以及管理学生。

一个便捷方便的沟通平台能够使得学生更好的学习理解课程，也能够使得老师更快更及时的得到学生信息反馈，为接下来的课程设计做出更好准备计划。通过这个系统，使得“线上”和“线下”有机结合，支持教师在课堂教学之外的任何场所为学生答疑解惑，有效保证了教与学的延续性，并且展示了教学相长的魅力。在师生交流学习的过程中，解决学习上疑惑的同时，也分享了学生学习可能遇到的困难的心路历程。希望通过线上线下相结合的答疑方式，进一步强化教师和学生的联系，让更多的学生受益，也为老师减轻负担。

## 1.4 本论文的主要内容与结构

本论文研究并设计实现了基于ASP.NET MVC的学生选课管理信息系统。其前端使用了目前国内外最流行的开源框架——Bootstrap，界面美观，开发方便；后端应用了ASP.NET MVC架构，开发分层十分明确，同时便于维护；数据库同样也使用了Oracle旗下最流行也是最稳定的小型关系型数据库MySQL Database 5.1.7，既简单清晰，又高效安全；另外还结合了jQuery，Ajax等前端技术，实现了教师管理自己的课程（例如发布/修改课程）、学生选择课程、安排课表、管理员管理（如审核）课程、教师学生互相留言等功能。

第一章概要介绍了设计该选课管理信息系统的背景原因、设计目的以及现实意义。一章主要介绍设计信息安全课程在线系统的背景、目的以及意义。

第二章概要介绍了开发该系统所使用到的技术和开发工具（包括开发环境和运行环境）等。

第三章对该系统的设计进行了简单的需求分析，由先整体后局部的分析方法，先对该系统整体进行需求分析，弄清楚一些最基本的需求（类别）；然后在对每个需求类别进行细致的需求分析。随后针对前面得到的结论进行数据库分析和可行性分析。将分析结果以其他形式展示出来例如Excel图标和流程图等等；使用这种图文结合的方式不仅可以使他人更便于理解该系统，同时也更利于开发者（自己）梳理思路，整理脉络，从而快速高效的进行系统的开发。

第四章是概要设计，介绍了系统的基本结构以及数据库设计。

第五章是详细设计，介绍了系统的页面设计和类设计。通过图表介绍各个页面间的跳转以及相互关系和各个类的设计。

第六章是实现与测试，展示了系统实现的页面效果，介绍了后期单元测试的用例及结果。

第七章是总结，针对本次毕业设计做出总结以及反思毕业设计的不足以及后期可以改进的地方。

# 第2章 开发使用技术简介

## 2.1 ASP.NET框架与MVC模式

[Java](http://baike.baidu.com/item/Java/85979)是一门[面向对象](http://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1)编程语言，它不仅兼容[C++](http://baike.baidu.com/item/C%2B%2B)语言的各种优点，还摒弃了C++里难以理解的指针、多继承等概念，所以Java语言拥有两大特征——功能强大、简单易用。作为静态面向对象编程语言的佼佼者，Java极好地实现了面向对象理论，允许程序员以简洁直接的思维方式进行复杂的编程。Java具有简单性、面向对象、[分布式](http://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F/19276232)、[健壮性](http://baike.baidu.com/item/%E5%81%A5%E5%A3%AE%E6%80%A7)、[安全性](http://baike.baidu.com/item/%E5%AE%89%E5%85%A8%E6%80%A7)、平台独立与可移植性、[多线程](http://baike.baidu.com/item/%E5%A4%9A%E7%BA%BF%E7%A8%8B)、动态性等特点[1]。Java可以编写[桌面应用程序](http://baike.baidu.com/item/%E6%A1%8C%E9%9D%A2%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)、[Web应用程序](http://baike.baidu.com/item/Web%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)、[分布式系统](http://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E7%B3%BB%E7%BB%9F)和[嵌入式系统](http://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F%E7%B3%BB%E7%BB%9F)应用程序等。

MVC是一种软件设计典范，它体现了关注点分离这一基本的设计仿真，它将构成一个人机交互应用涉及的功能分为Model，Controller和View三个部分，他们各自具有相应的职责。Model是对应用状态和业务功能的封装，我们可以将它理解为同时包含数据和行为的领域模型。Model接收Controller的请求并完成相应的业务处理，在状态改变的时候向View发出相应的通知[2]。

## 2.2 jQuery与Bootstrap

jQuery是一个简洁快速的JavaScript框架。jQuery设计的宗旨是“write Less，Do More”，使用写更少的代码完成更多的事情。它封装了JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、Ajax交互等各种功能。

它的核心特性可以总结为：具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的css选择器，并且可对CSS选择器进行扩展；拥有便捷的插件扩展机制和丰富的插件。

DWZ富客户端框架(jQuery RIA framework)，是中国人自己开发的基于jQuery实现的Ajax RIA开源框架。其设计目标是快速开发、简单实用、扩展方便、轻量级、RIA思路。

DWZ框架区别于其它JS框架，具有以下优点：

1. 完全开源，其源码没有做任何混淆处理，相当方便扩展。
2. CSS和JS代码彻底分离，方便开发人员修改样式。
3. 简单实用，扩展方便，轻量级框架，快速开发。
4. 支持HTML扩展方式调用UI组件，开发人员不需写JS。
5. 开发人员只要懂HTML语法不需精通JS，他就可以使用Ajax开发后台。
6. 基于jQuery，其UI组件以jQuery插件形式发布，方便后期开发扩展。

## 2.3 AJAX与JSON

AJAX指的是一种异步JavaScript和XML，它是一种新的网页开发技术，用于创建交互式网页应用。AJAX 是一种用于创建快速动态网页的技术。它通过在服务器与后台进行轻量级的数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

JSON是一种轻量级的数据交换格式。它采用完全独立于编程语言的文本格式来存储和表示数据。因为其简洁和清晰的层次结构，它成为理想的数据交换语言。它有易于人阅读和编写、易于机器解析和生成以及有效地提升网络传输效率的优点。

## 2.4 开发工具

### 2.4.1 MySQL Workbench工具简介

SQL Server 是Microsoft 公司推出的关系型数据库管理系统。它具有很多优点，比如可伸缩行好、与其它相关软件集成程度高以及使用方便等，它可跨多种不同大型多处理器的服务器平台使用。

Microsoft SQL Server 数据库引擎为关系型数据和结构化数据提供了更安全可靠的存储功能，数据库引擎是SQLSERVER 系统的核心服务，负责完成数据的存储、处理和安全管理，提供了受控访问和快速事务处理，和大量的支持以保持高可用性的服务[3]。

### 2.4.2 Visual Studio 2017 Community工具简介

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台，它实际上只是一个框架和一组服务，但附带了一个标准的插件集，包括Java开发工具（Java Development Tools，JDT）等，通过插件组件构建开发环境。

Eclipse 最初由OTI和IBM两家公司的IDE产品开发组创建，起始于1999年4月。IBM提供了最初的Eclipse代码基础，包括Platform、JDT 和PDE。尽管 Eclipse 是使用Java语言开发的，但它的用途并不限于 Java 语言；例如，支持诸如C/C++、COBOL、PHP、Android等编程语言的插件已经可用，或预计将会推出。经可用，或预计将会推出。

# 第3章 需求分析

## 3.1 系统需求及设计思路

### 3.1.1系统需求

本系统需求如表3.1 系统需求所示。

表3.1 系统需求

|  |  |
| --- | --- |
| 系统名称 | 功能模块 |
| 学生选课管理信息系统 | 学生课程管理模块 |
| 教师课程管理模块 |
| 学生教师交互模块 |

### 3.1.2系统设计思路

本次毕业设计主要通过前端DWZ框架和后台MVC框架思想结合实现，用户界面友好且易于操作，用户可以根据不同的角色权限对不同的信息进行浏览、查看以及编辑。前端的DWZ框架是一个国产开源框架，简单实用，适合快速开发。使用该框架可以省去前端UI设计，提高界面的可读性和美观。后台MVC框架分层明确，将模型、视图和控制器分开，它们各自处理自己的分工任务，这种模式便于理解也便于测试和后期维护。

## 3.2系统需求分析

### 3.2.1功能需求分析

本系统主要有以下用例：

1. 用户登录/注册
2. 教师管理（发布/修改/查看）课程
3. 教师/学生互相查看信息
4. 学生管理（选择/收藏/查看）课程
5. 学生/教师相互留言
6. 教师/学生管理个人信息

用例图见图3.1

[](UsecaseDiagram.pdf)

图3.1 学生选课管理信息系统用例图

用例说明：

表 3.2 用户登录/注册用例描述

|  |
| --- |
| 用例名：用户登录/注册 |
| 主要参与者：教师/学生 |
| 概述：教师/学生分别根据学号/工号进行系统注册/登录；并根据其身份在系统内进行相关的操作。 |
| 前置条件：登录：该用户存在且合法；注册：无重复用户名和密码。 |
| 主要流程：  基本路径：  (1) 当用户进入该系统并准备登录/注册时，用例开始；  (2) 用户输入正确用户名及密码；  (3) 用户成功登录并显示相应权限界面，用例结束。否则进行第（4）步。  (4) 用户根据输入的用户名和密码进行该身份的注册。  可选路径：  (1) 在选择提交前的任何时候，用户都可以选择取消。若登录取消，用例结束；  (2) 在基本路径第II步，如果用户名或密码有误，系统提示用户用户名或密码有误，请重新输入用户名和密码。 |
| 后置条件：如果登录成功，跳转到教师/学生对应的个人主页界面；不成功停留在登录界面并给出相应错误提示。 |

表3.3 管理课程（增删改查）用例说明

|  |
| --- |
| 用例名：管理课程（增删改查） |
| 主要参与者：教师 |
| 概述：教师增/删/改/查课程（并交由管理员审核），此处以“**增**”为用例进行说明。 |
| 前置条件：一个存在且合法的用户——教师已经正确登录系统 |
| 主要流程：  基本路径：  (1) 当用户进入该系统并点击新增课程时，用例开始；  (2) 用户输入课程相关信息，具体见数据库设计中的课程类设计。课程ID，教师ID将分别由系统生成和获取；系统对输入进行判断无误之后，点击提交；  (3) 用户成功提交课程，用例结束。  可选路径：  (1) 在选择提交前的任何时候，用户都可以选择取消。这次新增取消，用例结束。  (2) 在基本路径第II步，如果课程相关信息填写不符合规范，系统提示用户检查并重新填写。如果一直存在不合法内容，则只能取消或停留在当前页面。 |
| 后置条件：如果提交成功则更新数据库，否则数据库将不做更新。 |

表3.4 管理学生（查看/标记）用例说明

|  |
| --- |
| 用例名：上传作业 |
| 主要参与者：学生 |
| 概述：系统用户提交课程作业 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径：  (1) 当用户进入该系统并准备上传作业时，用例开始；  (2) 用户点击我的课程，选择要提交作业的课程名称，点击查看详情，点击要提交作业的章节，点击上传作业；  (3) 用户点击浏览，选择要上传的作业路径并选择文件上传；  (4) 用户上传成功且调回主页面，用例结束。  可选路径：  (1) 在选择提交前的任何时候，用户都可以选择取消。这次上传取消，用例结束。  (2) 在基本路径第III步，如果用户上传文件不符合规则，例如文件过大或者文件类型不对，系统提示用户相应问题，请重新上传。 |
| 后置条件：如果上传成功跳转主界面，不成功则停留在当前界面 |

表3.5 评价学生用例说明

|  |
| --- |
| 用例名：新增评论 |
| 主要参与者：教师、学生 |
| 概述：系统用户对某个话题进行评论 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径  (1) 当用户进入该系统并准备新增评论时，用例开始  (2) 用户点击论坛，选择要评论的话题，点击查看详情并点击新增评论。  (3) 用户成功发表评论，用例结束  可选路径  (1) 在选择提交前的任何时候，用户都可以选择取消。这次评论取消，用例结束。 |
| 后置条件：如果评论成功则更新数据库，否则数据库不做更新。 |

表3.6 管理学生（增删改查）用例说明

|  |
| --- |
| 用例名：在线聊天 |
| 主要参与者：教师、学生 |
| 概述：系统用户在线聊天 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径  (1) 当用户进入该系统并聊天时，用例开始；  (2) 用户输入聊天昵称并点击进入聊天室；  (3) 用户输入聊天内容并点击发送消息；  (4) 用户点击退出聊天则结束聊天，用例结束。  可选路径：  (1) 在基本路径第II步，如果用户没有输入聊天昵称，系统提示用户必须输入聊天昵称。 |
| 后置条件：如果进入成功进入聊天显示相应聊天界面，不成功提示错误信息。 |

表3.7 管理课程（增删改查）用例说明

|  |
| --- |
| 用例名：修改用户信息 |
| 主要参与者：所有用户 |
| 概述：系统用户修改个人信息 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径  I. 当用户进入该系统并点击我的信息时，用例开始  II.用户修改可编辑的个人信息  III.用户点击提交，用例结束  可选路径  I.在选择提交前的任何时候，用户都可以选择取消。这次修改取消，用例结束。 |
| 后置条件：如果修改成功则更新数据库，否则数据库不做更新。 |

表3.8 管理教师（增删改查）用例说明

|  |
| --- |
| 用例名：修改用户信息 |
| 主要参与者：所有用户 |
| 概述：系统用户修改个人信息 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径  I. 当用户进入该系统并点击我的信息时，用例开始  II.用户修改可编辑的个人信息  III.用户点击提交，用例结束  可选路径  I.在选择提交前的任何时候，用户都可以选择取消。这次修改取消，用例结束。 |
| 后置条件：如果修改成功则更新数据库，否则数据库不做更新。 |

表3.9 评价课程（评价老师）用例说明

|  |
| --- |
| 用例名：修改用户信息 |
| 主要参与者：所有用户 |
| 概述：系统用户修改个人信息 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径  I. 当用户进入该系统并点击我的信息时，用例开始  II.用户修改可编辑的个人信息  III.用户点击提交，用例结束  可选路径  I.在选择提交前的任何时候，用户都可以选择取消。这次修改取消，用例结束。 |
| 后置条件：如果修改成功则更新数据库，否则数据库不做更新。 |

表3.10 查看课程用例说明

|  |
| --- |
| 用例名：修改用户信息 |
| 主要参与者：所有用户 |
| 概述：系统用户修改个人信息 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径  I. 当用户进入该系统并点击我的信息时，用例开始  II.用户修改可编辑的个人信息  III.用户点击提交，用例结束  可选路径  I.在选择提交前的任何时候，用户都可以选择取消。这次修改取消，用例结束。 |
| 后置条件：如果修改成功则更新数据库，否则数据库不做更新。 |

表3.11 查看教师用例说明

|  |
| --- |
| 用例名：修改用户信息 |
| 主要参与者：所有用户 |
| 概述：系统用户修改个人信息 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径  I. 当用户进入该系统并点击我的信息时，用例开始  II.用户修改可编辑的个人信息  III.用户点击提交，用例结束  可选路径  I.在选择提交前的任何时候，用户都可以选择取消。这次修改取消，用例结束。 |
| 后置条件：如果修改成功则更新数据库，否则数据库不做更新。 |

表3.12 管理课程（选课/收藏）用例说明

|  |
| --- |
| 用例名：修改用户信息 |
| 主要参与者：所有用户 |
| 概述：系统用户修改个人信息 |
| 前置条件：一个存在且合法的用户已经正确登录系统 |
| 主要流程：  基本路径  I. 当用户进入该系统并点击我的信息时，用例开始  II.用户修改可编辑的个人信息  III.用户点击提交，用例结束  可选路径  I.在选择提交前的任何时候，用户都可以选择取消。这次修改取消，用例结束。 |
| 后置条件：如果修改成功则更新数据库，否则数据库不做更新。 |

### 3.2.3数据库需求分析

本系统采用的是SQL SERVER 2008数据库。本系统实体主要有：课件、作业、教师、管理员、学生、话题、课程、聊天室以及评论。其中教师可以上传或下载课件、发表或删除话题以及评论、加入聊天室聊天、下载作业以及编辑课程；管理员可以管理教师和学生信息、删除话题和评论以及管理课程；学生可以下载课件、上传或下载自己的作业、发表或删除话题以及评论并加入聊天室。

教师所具有的属性是ID、姓名、密码、电话以及可用信息。管理员所具有的属性是ID、名称、密码以及可用信息。学生所具有的属性是ID、姓名、密码、电话以及可用信息。课程所具有的属性是ID、课程名称、教师ID、报名学生数、课程简介以及可用信息。课件所具有的属性是ID、上传人、文件路径、课程ID、上传时间、文件名称以及可用信息。作业所具有的属性是ID、上传人、文件路线、课程ID、上传时间、文件名称以及可用信息。话题所具有的属性是ID、作者ID、发表时间、回复话题评论数以及可用信息。评论所具有的属性是ID、评论作者ID、内容、发表时间息、对应话题ID以及可用信息。聊天室所具有的属性是参与对象以及聊天内容。

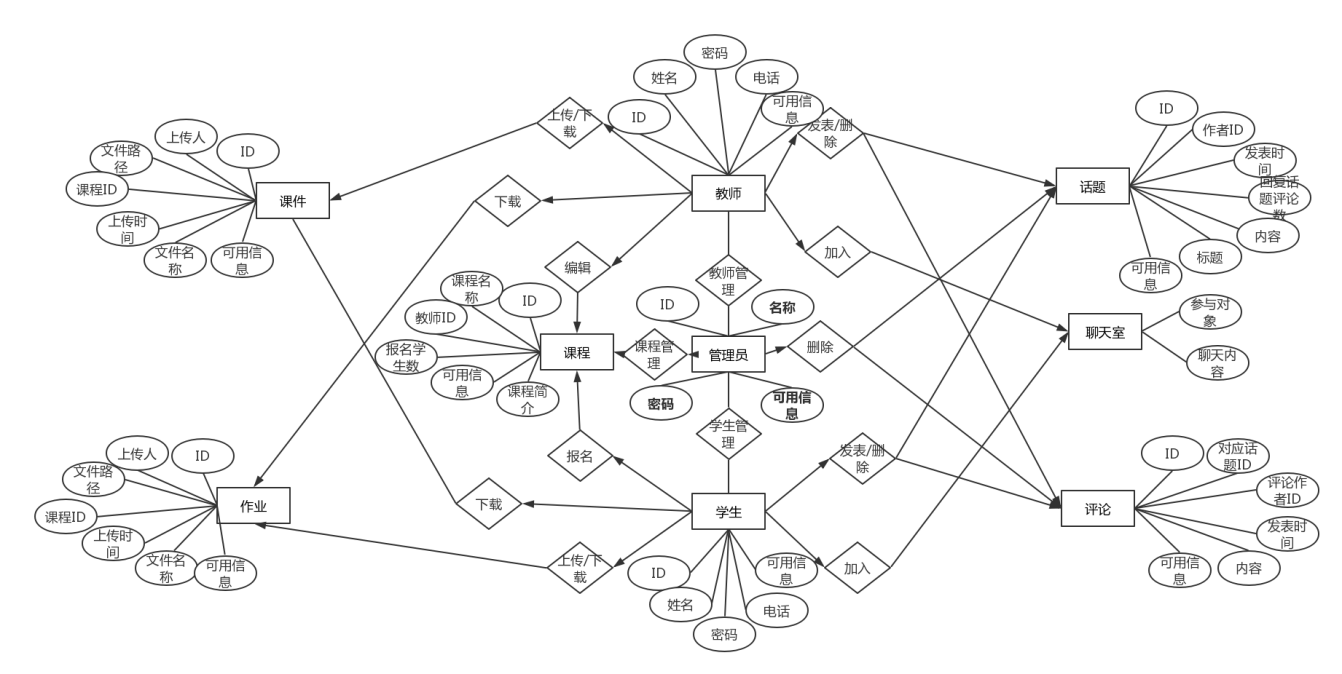


图3.2 系统E-R图

### 3.2.4性能需求分析

本系统开发软件采用Eclipse作为开发工具；后台配合SQL SERVER作为存储数据的数据库开发工具；并于windows10操作系统下完成本次系统开发。但是本系统支持Windows7、8、10等平台。其硬件需求为512M以上内存；至少80G以上的硬盘；32倍速以上的CD-ROM；10MB/100MB自适应的网络适配器；并使用TCP/IP协议的局域网。

## 3.3 可行性分析

### 3.3.1 设计目标

本毕业设计使用Java语言设计并实现信息安全课程在线系统。系统根据不同角色分配不同权限。管理员可以通过系统增加或者修改其他角色的登录信息和基本信息，新增课程以及管理留言信息；教师可以通过系统查看编辑自己的课程，上传课件，查看作业，查看报名学生的基本信息，查看其他老师的开设的课程，并且在线答疑、回复留言；学生可以通过系统报名参加老师的课程，查看相关课程课件并下载，上传作业，在线提问以及留言。

### 3.3.2技术可行性

技术可行性指的是使用现有的技术能实现这个系统吗？该系统采用Eclipse实现开发，并配合SQLSERVER 2008存储数据，在软件开发平台这一条件上已经成熟可行。硬件方面根据现在普遍的计算机配置都可满足。由上述因素综合分析，本系统在技术上是可行的[4]。

### 3.3.3经济可行性

经济可行性指的是这个系统的经济效益能超过它的开发成本吗？本系统采用的开发软件、使用的框架都是开源免费的，开发成本较低，没有特别的额外支出。另外本系统采用MVC架构，分层明确，便于维护，后期的测试维护更新不会占用和浪费过多资源，由上述因素综合分析，本系统在经济上是可行的。

### 3.3.4操作可行性

操作可行性指的是系统操作方式在这个用户组织内行得通吗？本系统适用于学校组织，根据权限功能的不同划分了不同的角色，每种角色能够完成自己能力范围内的功能。另外，本系统操作简单，界面人性化，对用户没有特别的技术要求，易于掌握，由上述因素综合分析，本系统在操作上是可行的。经可用，或预计将会推出。

# 第4章 系统概要设计

## 4.1 系统结构图

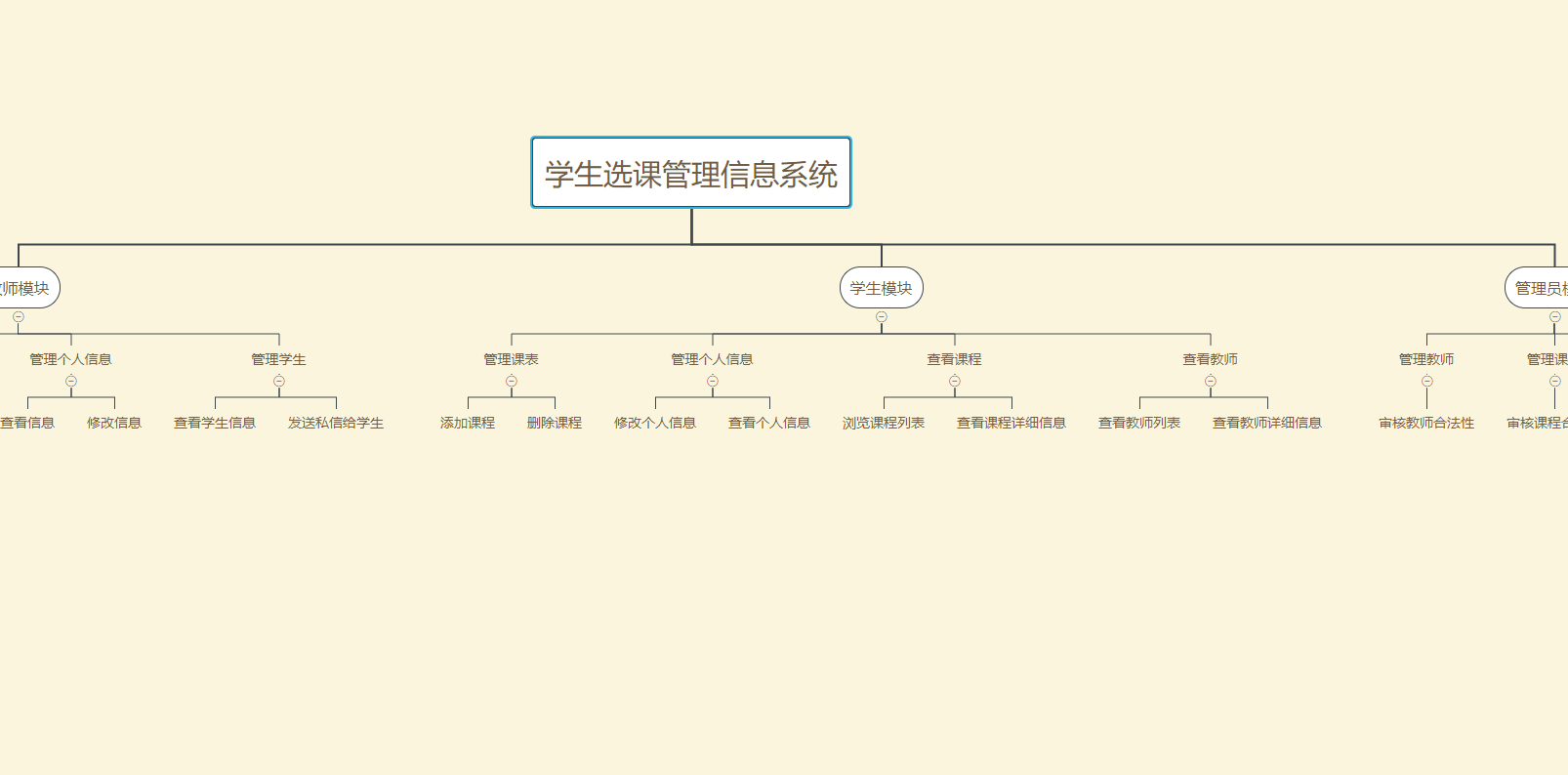
[](https://www.processon.com/view/link/5afa8fece4b07510694e0a36)

图4.1 系统逻辑结构图（Ctrl+Click联网查看详细文件）

本信息安全课程在线系统分为三大模块，分别为课程信息管理模块、用户信息管理模块以及留言信息管理模块。根据角色权限的不同又能够实现各种不同的功能。管理员角色可以添加课程信息、查看所有教师信息、查看所有学生信息、删除所有留言以及删除所有评论。教师角色可以编辑课程信息、上传课程课件、查看上传课件、查看上传作业、查看已报名学生信息、新增留言、新增评论、删除我的留言、删除我的评论以及加入聊天室。学生角色可以查看上传课件、报名参加课程、上传课程作业、查看已报名教师信息、新增留言、新增评论、删除我的留言、删除我的评论以及加入聊天室。

## 4.2数据库设计

### 4.2.1数据库详细设计

表4.1 教师信息表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| uniqueClientID | 唯一编号 | char(10) |
| name | 姓名 | varchar(25) |
| gender | 性别 | char(1) |
| age | 年龄 | smallint |
| email | 邮箱 | varchar(50) |
| wechat | 微信 | varchar(25) |
| qqNumber | QQ号码 | varchar(25) |
| phone | 手机号码 | varchar(25) |
| university | 学校 | varchar(100) |
| college | 学院 | varchar(100) |
| major | 专业 | varchar(100) |
| courses | 所授课程 | varchar(125) |
| workNo | 教工号 | varchar(20) |
| workPass | 教工密码 | varchar(20) |

表4.1 学生信息表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| uniqueClientID | 唯一编号 | char(10) |
| name | 姓名 | varchar(30) |
| gender | 性别 | char(1) |
| age | 年龄 | smallint |
| email | 邮箱 | varchar(50) |
| wechat | 微信 | varchar(50) |
| qqNumber | QQ号码 | varchar(12) |
| phone | 手机号码 | varchar(30) |
| university | 学校 | varchar(100) |
| college | 学院 | varchar(100) |
| major | 专业 | varchar(100) |
| grade | 年级 | varchar(10) |
| cardNo | 学生学号 | varchar(20) |
| cardPass | 学号密码 | varchar(20) |

表4.2 登录信息表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| uniqueClientID | 唯一编号 | char(10) |
| username | 用户名 | varchar(50) |
| password | 密码 | varchar(50) |
| registryDate | 注册日期 | datetime |
| registryType | 注册类型 | char(1) |

表4.1 课程信息表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| courseID | 课程代号 | char(10) |
| courseName | 课程名称 | varchar(100) |
| courseTeacher | 代课老师 | char(10) |
| publishDate | 发布日期 | date |
| startDate | 开始日期 | date |
| endDate | 结束日期 | date |
| venue | 上课地点 | varchar(60) |
| period | 上课时间 | varchar(50) |
| examDate | 考试时间 | datetime |
| examDuration | 考试时长 | digit(2) |
| examVenue | 考试地点 | varchar(35) |
| examID | 考试信息代号 | char(10) |
| chosen | 历史选课人数 | int |
| collected | 被收藏次数 | int |
| courseInfo | 课程介绍 | varchar(255) |
| detailInfo | 详细课程介绍 | varchar(2048) |
| subTag | 二级学科名称 | varchar(50) |

表4.1 XXXX表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

表4.3 考试信息表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| examID | 考试代号 | char(10) |
| courseID | 课程代号 | char(10) |
| examDate | 考试时间 | datetime |
| examDuration | 考试时长 | digit(2) |
| examVenue | 考试地点 | varchar(60) |
| examType | 考试类型 | varchar(10) |
| examTaken | 考试人数 | int |
| examRate | 通过率 | varchar(10) |
| ava | 是否结束 | char |

表4.4 教室信息表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| roomID | 教室代号 | char(10) |
| roomName | 教室名称 | varchar(60) |
| roomTag | 教室别称 | varchar(35) |
| roomBuilding | 所属建筑 | varchar(35) |
| roomCapacity | 可容人数 | Int |

表4.5 教室使用表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| roomID | 教室代号 | char(10) |
| weekday | 占用工作日 |  |
| period | 占用时段 |  |
| type | 占用类型 | char(1) |
| occupier | 占用人 | char(10) |
| dayperiod | （结合字段） | char(2) |

表4.6 学科门类表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| name | 学科门类 | varchar(20) |
| code | 学科代号 | char(2) |

表4.7 XXXX表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

表4.8 XXXX表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

表4.9 评论信息表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 说明 |
| id | Int | 评论唯一标识ID |
| bbsid | Int | 评论对应话题ID |
| auth | Nchar(20) | 发表评论作者ID |
| time | Datetime | 发表评论时间 |
| main | Nchar(200) | 评论内容 |
| retired | Int | 表示可用信息 |

# 第5章 详细设计

## 5.1 页面设计

### 5.1.1登录/注册页面

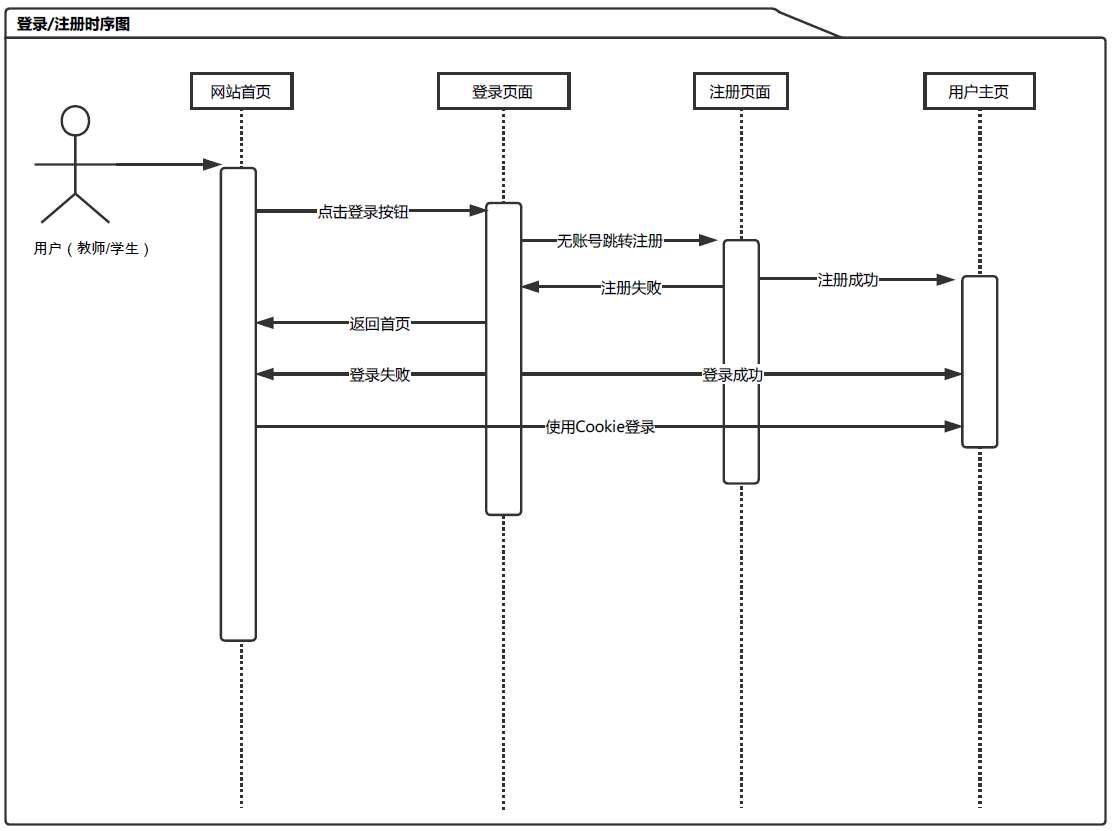
[](GP-详细设计-顺序图-登录注册.pdf)

图5.1 用户登录/注册顺序图

### 5.1.2管理员管理教师/学生

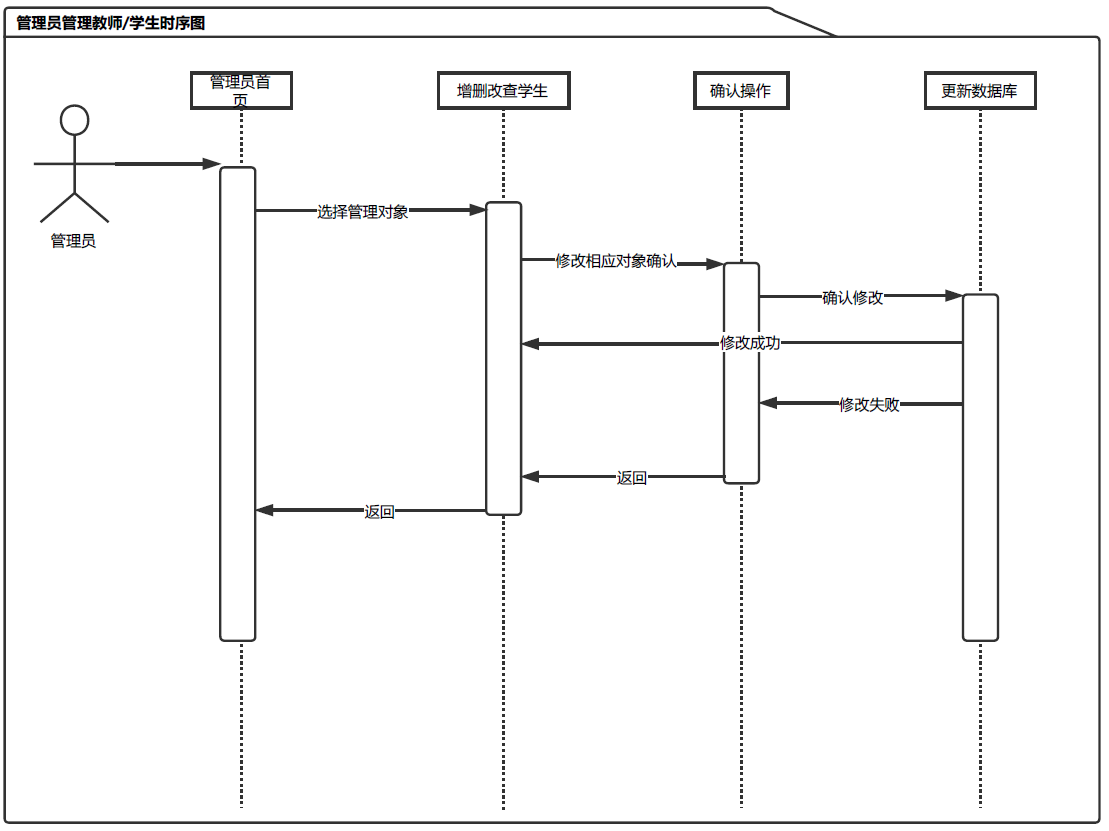
[](GP-详细设计-顺序图-管理员管理教师-学生.pdf)

图5.2 管理员管理教师/学生顺序图

### 5.1.3管理员管理课程

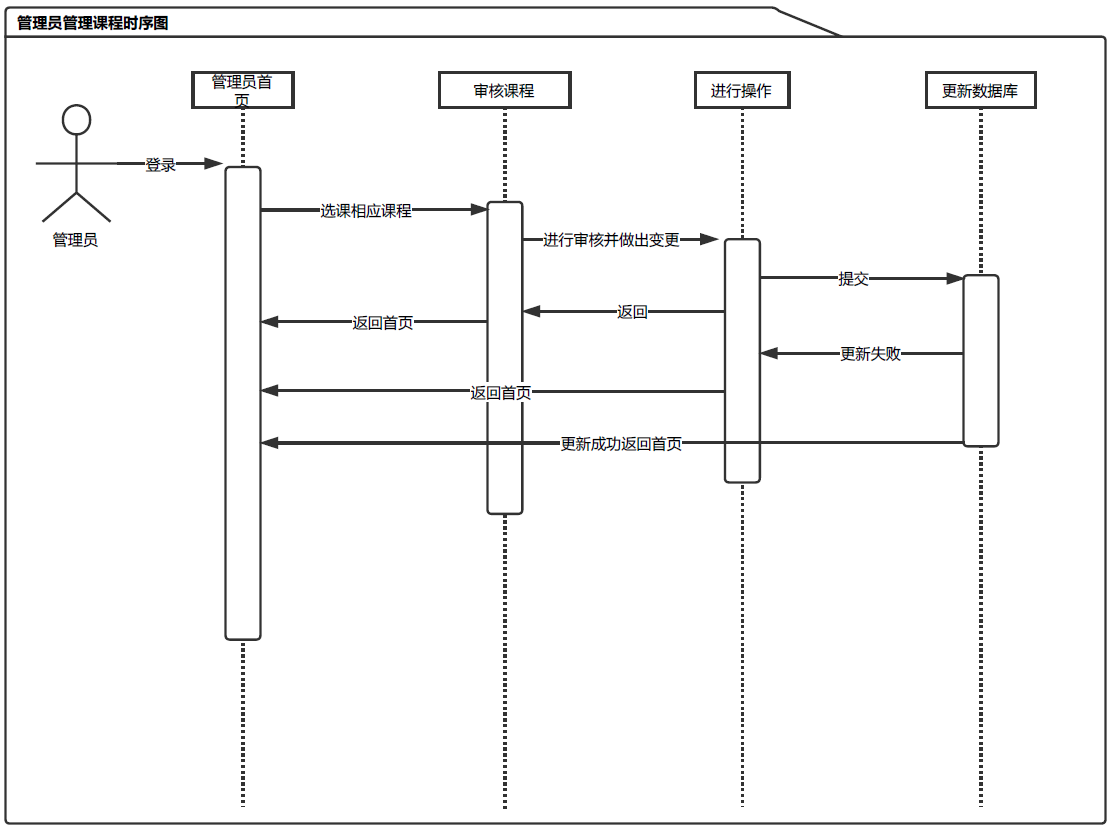
[](GP-详细设计-顺序图-管理员管理课程.pdf)

图5.3 管理员管理课程时序图

### 5.1.4教师管理课程

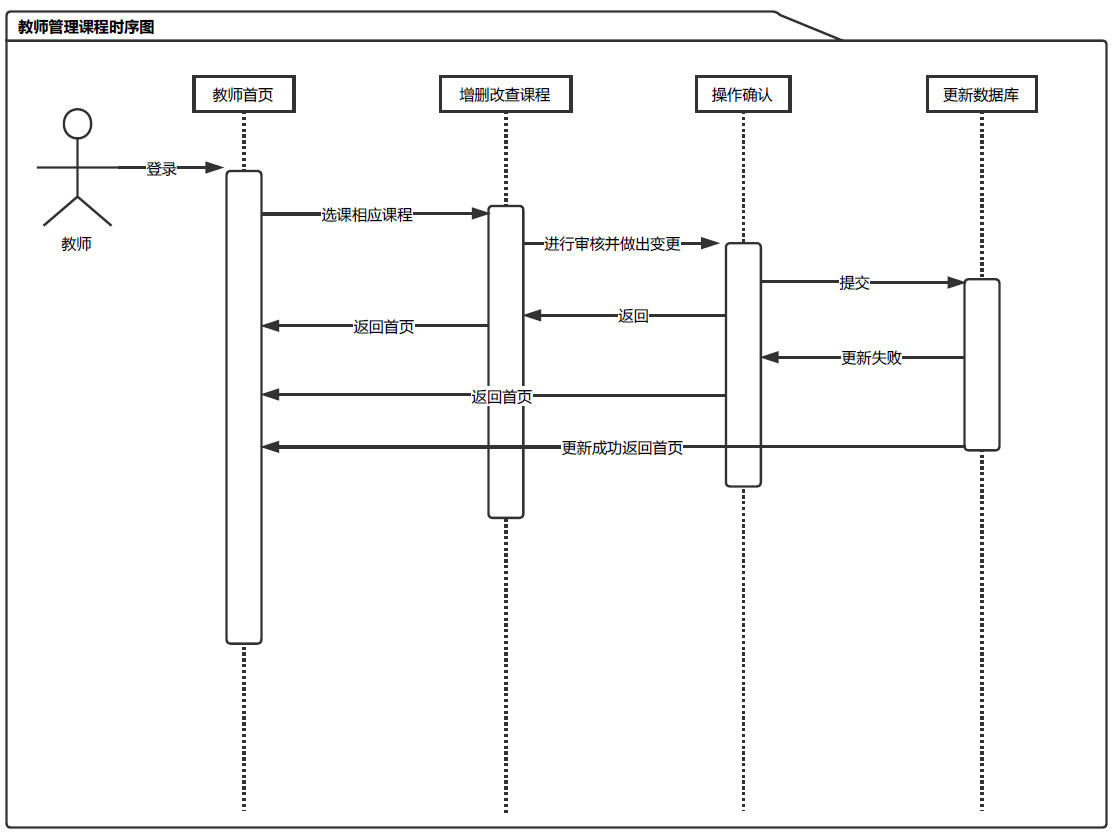
[](GP-详细设计-顺序图-教师管理课程.pdf)

图5.4 教师上传课件顺序图

### 5.1.5教师/学生相互评论

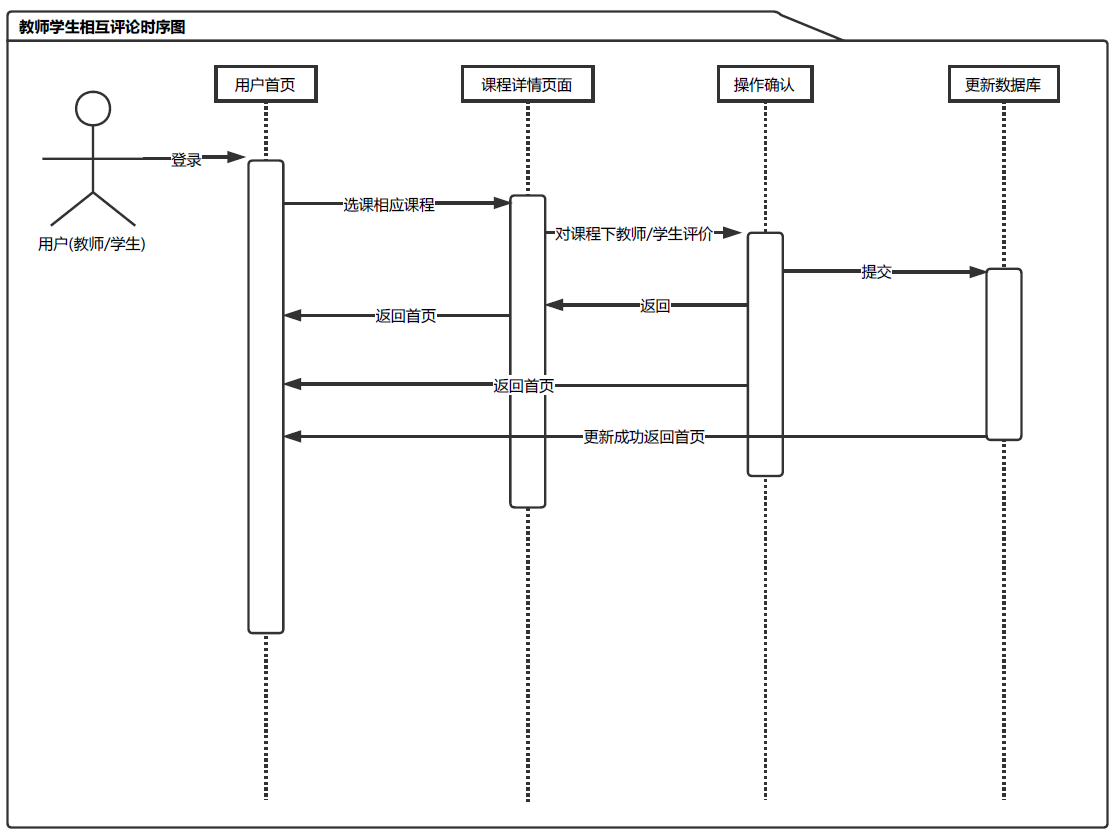
[](GP-详细设计-顺序图-教师学生相互评论.pdf)

图5.5 教师/用户新增评论顺序图

### 5.1.6教师管理学生

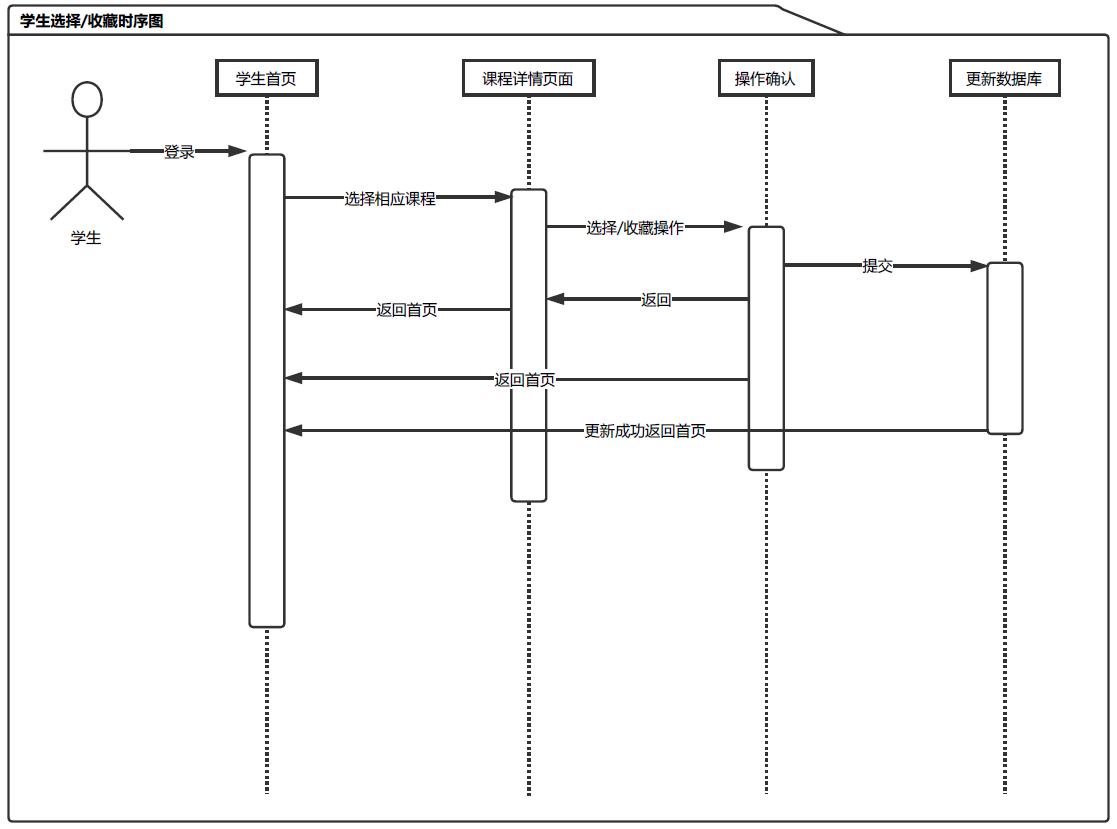
[](GP-详细设计-顺序图-学生选课-收藏课程.pdf)

图5.6教师下载作业顺序图

### 5.1.7学生选择/收藏课程

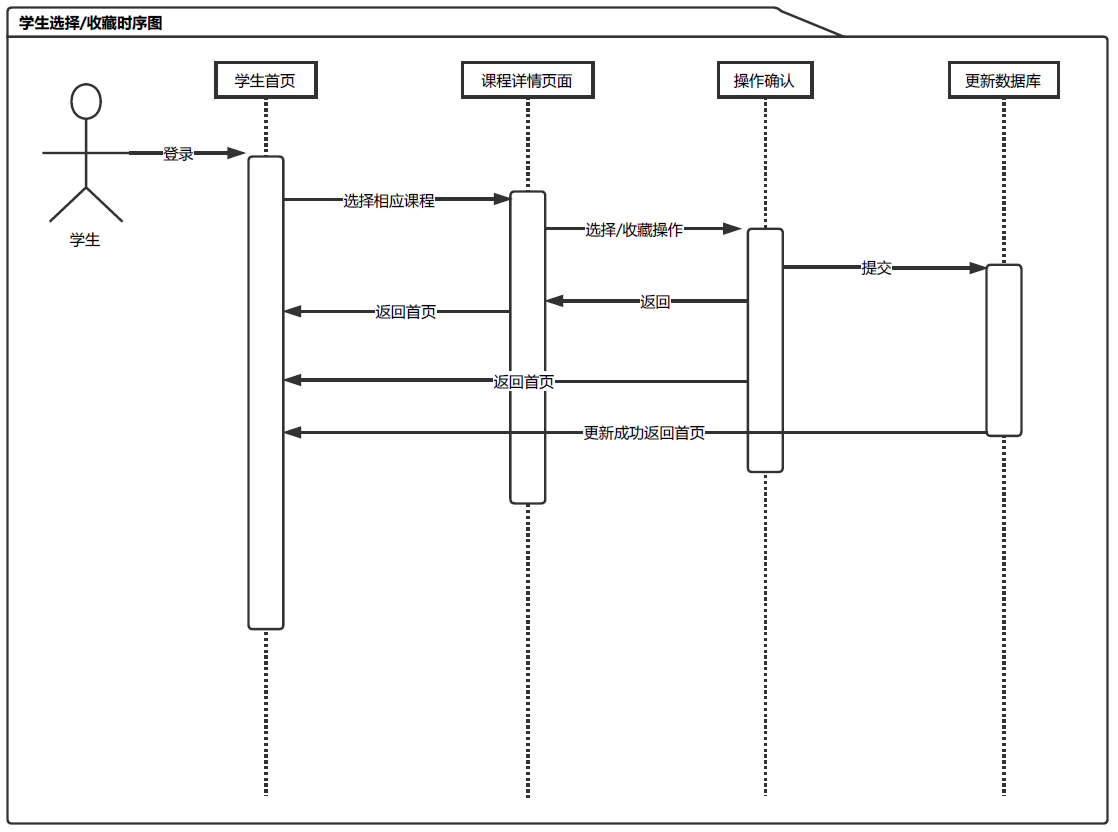
[](GP-详细设计-顺序图-学生选课-收藏课程.pdf)

图5.7 学生报名课程顺序图

### 5.1.8教师/学生修改个人信息

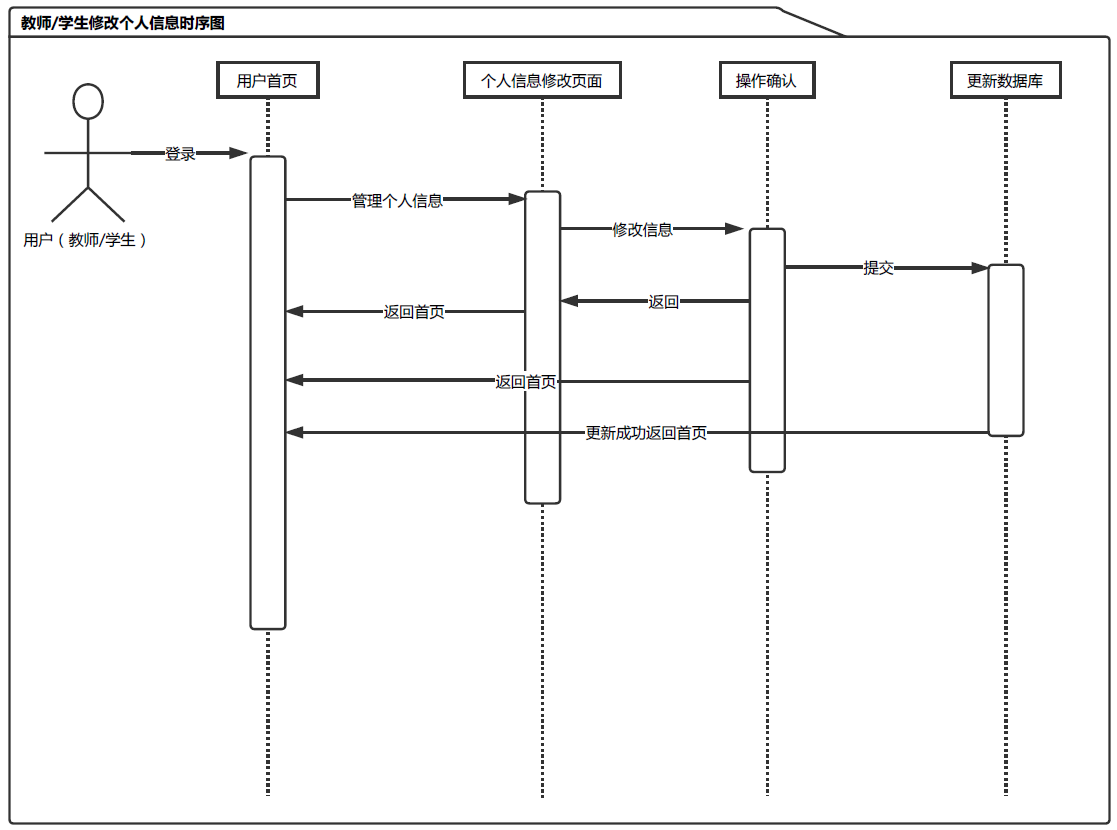
[](GP-详细设计-顺序图-教师学生修改个人信息.pdf)

图5.8 教师/学生修改个人信息顺序图

### 5.1.9教师/学生私信互动

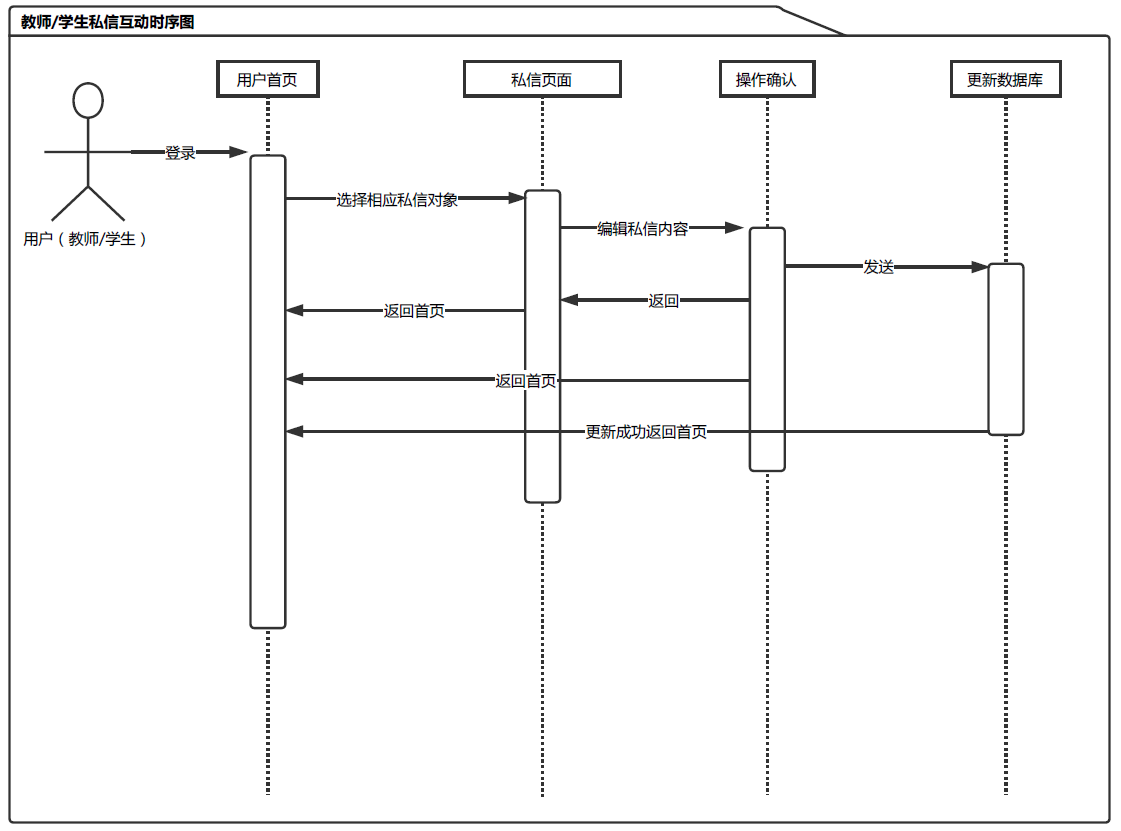
[](GP-详细设计-顺序图-教师学生私信互动.pdf)

图5.9 教师/学生聊天顺序图

## 5.2 实体类设计

### 5.2.1用户类设计

表5.1 用户类

|  |
| --- |
| class User |
| public string UniqueClientID { get; set; }  public string Username { get; set; }  public string Password { get; set; }  public string RegistryDate { get; set; }  public char RegistryType { get; set; } |
| //构造函数：初始化字段  public User  {  } |

表5.1.1 教师类（继承）用户类

|  |
| --- |
| class Teacher |
| public string UniqueClientID { get; set; }  public string Name { get; set; }  public char Gender { get; set; }  public int Age { get; set; }  public string Email { get; set; }  public string Wechat { get; set; }  public string QQ { get; set; }  public string Phone { get; set; }  public string University { get; set; }  public string College { get; set; }  public string Major { get; set; }  public char Position { get; set; }  public string Courses { get; set; }  public string WorkNo { get; set; }  public string WorkPass { get; set; } |
| //构造函数：初始化字段  public Teacher  {  } |

表5.1.2 学生类（继承）用户类

|  |
| --- |
| class Student |
| public string UniqueClientID { get; set; }  public string Name { get; set; }  public char Gender { get; set; }  public int Age { get; set; }  public string Email { get; set; }  public string Wechat { get; set; }  public string QQ { get; set; }  public string Phone { get; set; }  public string University { get; set; }  public string College { get; set; }  public string Major { get; set; }  public char Position { get; set; }  public string Grade { get; set; }  public string CardNo { get; set; }  public string CardPass { get; set; } |
| //构造函数：初始化字段  public Student  {  } |

### 5.2.2课程类设计

表5.2 课程类

|  |
| --- |
| class Course |
| public string CourseType { get; set; }  public string CourseID { get; set; }  public string CourseName { get; set; }  public string CourseTeacher { get; set; }  public string PublishDate { get; set; }  public string StartDate { get; set; }  public string EndDate { get; set; }  public string Venue { get; set; }  public string Period { get; set; }  public string ExamID { get; set; }  public int Chosen { get; set; }  public int Collected { get; set; }  public string CourseInfo { get; set; }  public string DetailInfo { get; set; }  public string SubTag { get; set; } |
| //构造函数：初始化字段  public Course()  {  } |

### 5.2.3课表类设计

表5.3 课表类

|  |
| --- |
| class CourseTable |
| public string UniqueClientID{ get; set; }  public string CourseID { get; set; }  public string ManiType{ get; set; }  public string ManipDate{ get; set; }  public string IfDeleted{ get; set; } |
| //构造函数：初始化字段  public CourseTable()  {  } |

### 

### 5.2.4教室类设计

表5.4 教室类

|  |
| --- |
| class Room |
| public string RoomID { get; set; }  public string RoomName { get; set; }  public string RoomTag { get; set; }  public string RoomBuilding { get; set; }  public int RoomCapacity { get; set; } |
| //构造函数：初始化字段  public Room()  {  } |

### 5.2.5考试类设计

表5.5 考试类

|  |
| --- |
| class Exam |
| public string ExamID { get; set; }  public string CourseID { get; set; }  public string ExamDate { get; set; }  public string ExamDuration { get; set; }  public string ExamVenue { get; set; }  public string ExamType { get; set; } |
| //构造函数：初始化字段  public Exam()  {  } |

### 5.2.6学科类设计

表5.6 学科类（枚举）

|  |
| --- |
| enum Subject |
| //**Code** **Name**  //"CA"> 哲学  //"CB"> 经济学  //"CC"> 法学  //"CD"> 教育学  //"CE"> 文学  //"CF"> 历史学  //"CG"> 理学  //"CH"> 工学  //"CI"> 农学  //"CJ"> 医学  //"CK"> 军事学  //"CL"> 管理学  //"CM"> 艺术学 |
|  |

### 5.2.7教室占用类设计

表5.7 教室占用类

|  |
| --- |
| class RoomUse |
| public string weekday { get; set; }  public string period { get; set; }  public string type { get; set; }  public string occupier { get; set; }  public string dayperiod { get; set; }  public string roomID { get; set; } |
| //构造函数：初始化字段  public RoomUse  {  } |

# 第6章 软件的实现与测试

## 6.1 软件界面实现

### 6.1.1用户登录/注册界面

图6.1为用户登录界面，用户在这里输入正确的用户名密码即可登录信息安全课程在线系统中。

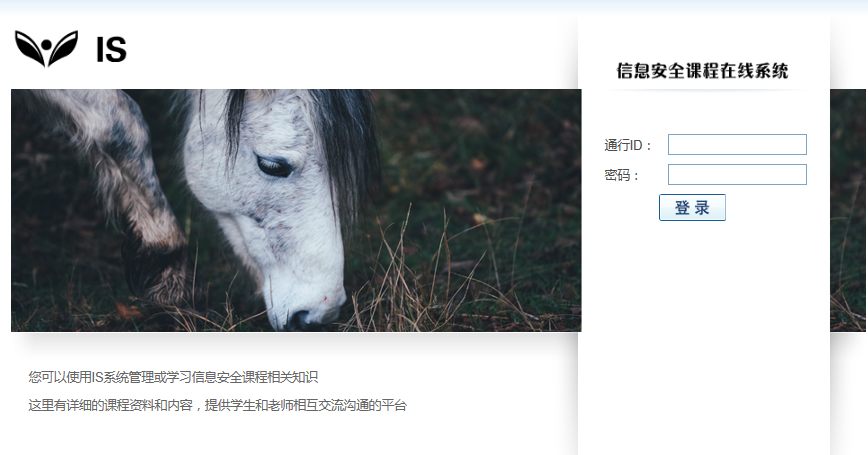


图6.1 前台登录界面

### 6.1.2用户首页展示界面

图6.2为管理员用户添加课程信息界面。用户可以通过点击右侧主菜单里的课程管理下的课程管理，显示课程管理页面。该页面会显示当前可用的课程ID、课程名称以及教师姓名的信息，并且有新增课程以及删除课程两个功能按钮。



图6.2 添加课程信息界面

管理员点击新增课程按钮会出现如下图6.3的界面，用户可以填写相关信息并且点击提交按钮保存课程信息。管理员也可以通过点击删除课程按钮删除课程。

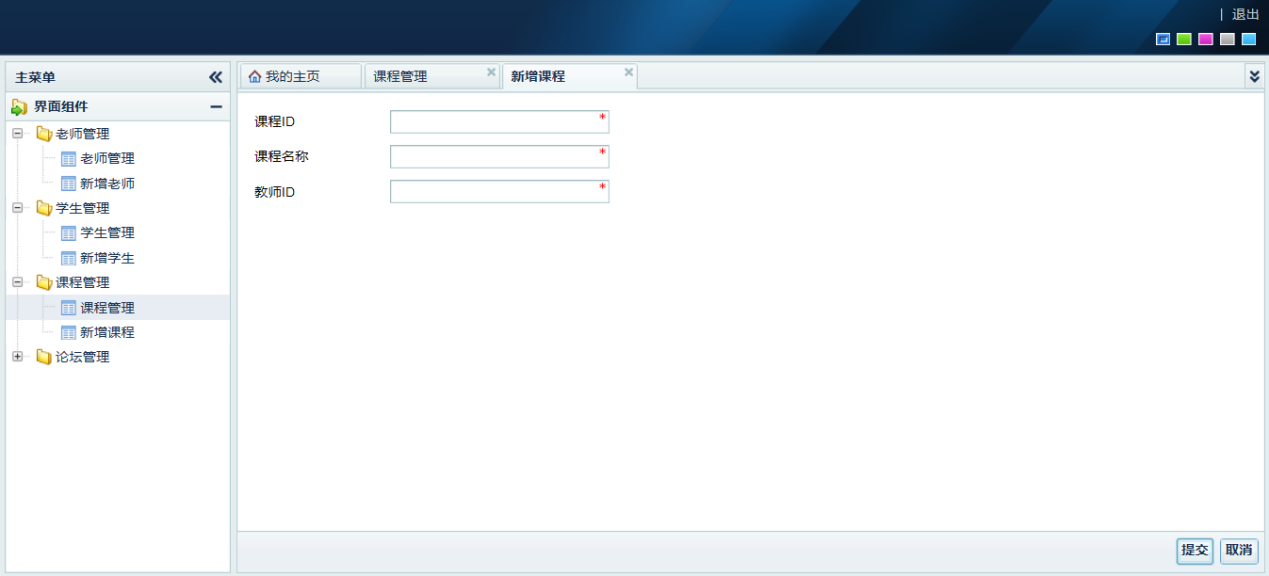


图6.3 新增课程信息界面

### 6.1.3教师管理课程信息界面

图6.4为教师查看课程列表信息界面。教师通过点击右侧主菜单栏里我的课程下我的课程可以显示所有本人开设课程信息。在该页面上有查看详情按钮，通过点击查看详情按钮可以查看这门课程的详细信息，如图6.5课程详细信息界面。在课程详细信息界面上可以查看不同作业的作业详情，如图6.6作业详细信息界面。在作业详细信息界面可以编辑作业要求简介，查看已经上传的课件以及选择上传作业。

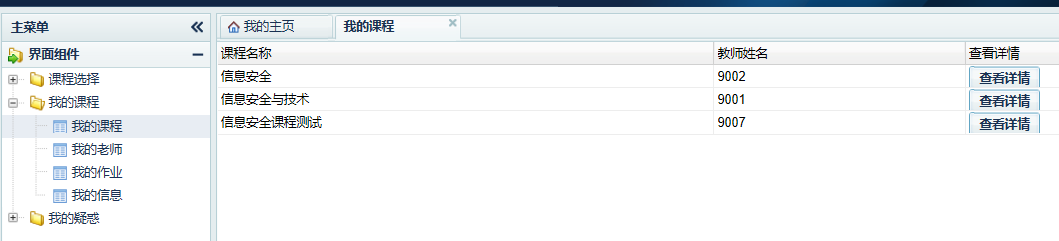


图6.4 课程列表信息界面



图6.5 课程详细信息界面



图6.6 作业详细信息界面

### 6.1.4用户管理课表信息界面

图6.7为教师或学生上传课件/作业界面。教师或学生可以通过该页面上传作业或者课件，点击浏览并选择需要上传的文件的文件夹路径并点击选择，然后点击上传就可以成功上传课件或者作业。

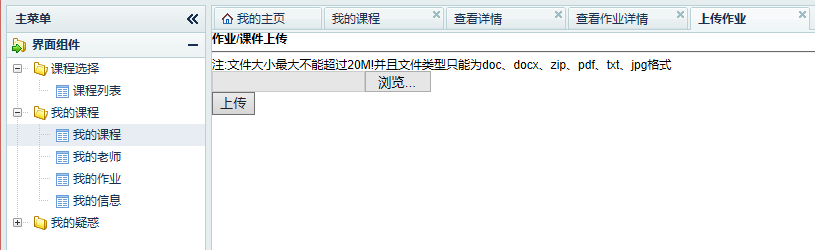


图6.7 上传课件/作业界面

### 6.1.5查看当前所有课程界面

图6.8以及图6.9为教师或学生查看上传课件/作业界面。教师或者学生可以通过该界面查看已上传作业或课件信息。教师或学生可以通过查看课件页面查看该课件的课程ID、作业ID、文件名称以及下载该课件；他们也可以通过查看作业界面查看该作业的课程ID、课程名称、作业ID、文件名称并选择下载。

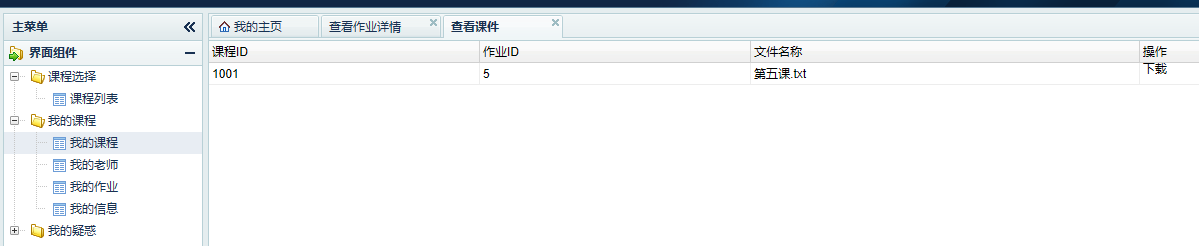


图6.8查看课件信息界面

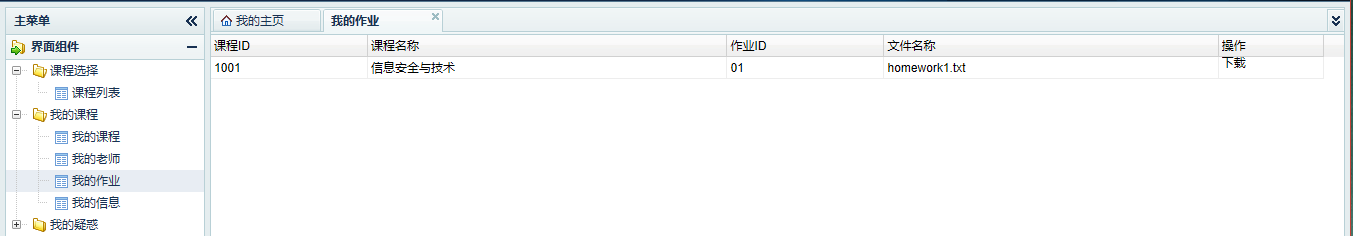


图6.9 查看作业信息界面

### 6.1.6收藏/选择/评论课程界面

图6.10为学生报名参加课程界面。学生可以通过该页面报名参加该项课程，点击报名并且报名成功以后，该课程会自动添加到我的课程的课程列表中。并且可以查看该课程的每章作业详情。



图6.10 报名参加课程界面

### 6.1.7查看所有教师/学生信息界面

图6.11为管理员查看教师信息界面。该界面会详细显示所有可用教师的教师id、教师姓名、登录密码以及电话等详细信息，另外还有新增教师以及删除教师两个功能按钮。点击新增教师会出现如图6.11的新增教师信息界面，填写教师id，教师姓名以及教师电话等基本信息并点击提交就可以新增教师。查看所有学生信息界面同该界面类似，在此不做赘述。



图6.11 所有教师信息界面

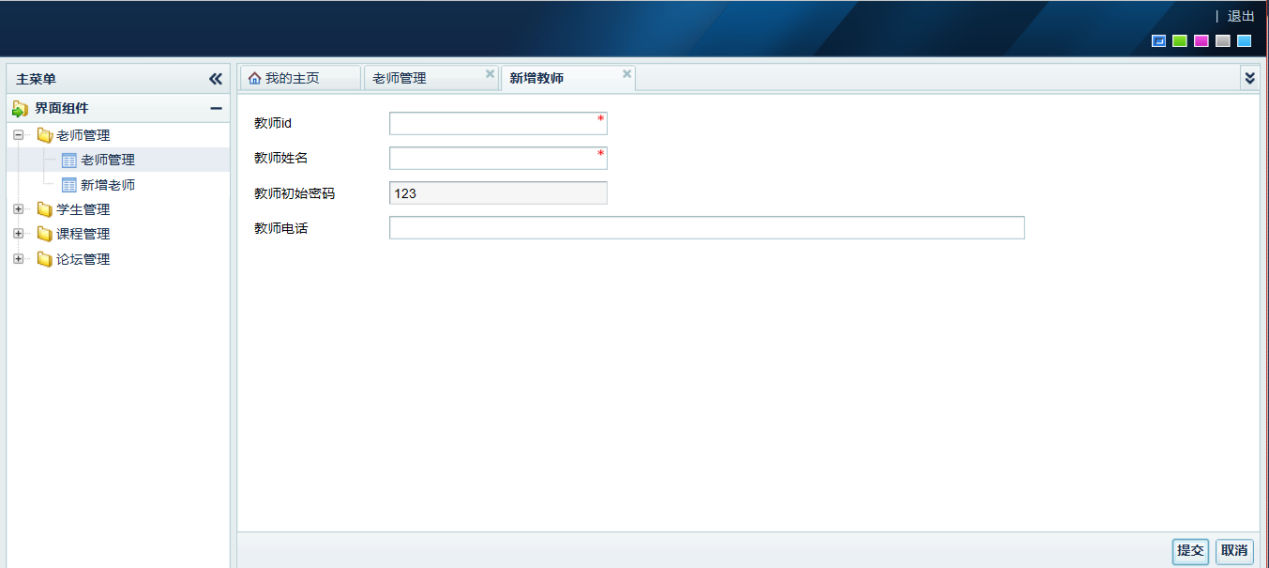


图6.12新增教师信息界面

### 6.1.8查看指定教师/学生信息界面

图6.13为学生查看我的教师信息界面。该界面会显示该学生已报名课程教师的教师ID、教师姓名以及电话信息。教师查看我的学生信息界面同该界面相似，在此不做赘述。

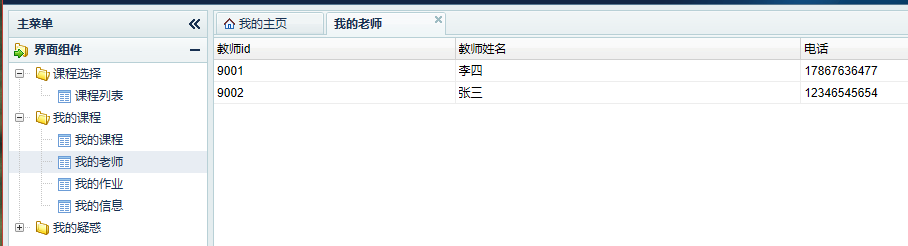


图6.13 我的老师信息界面

### 6.1.9教师/学生发送私信界面

图6.14为教师或学生查看所有留言话题界面，点击新增话题即可进入新增话题界面，如图6.15所示。点击查看详情可查看该话题详细信息，如图6.16所示。点击新增评论按钮，可进入新增评论界面，如图6.17所示。当新的评论提交成功是，话题的回复数会自动加1，并且显示在该话题下。

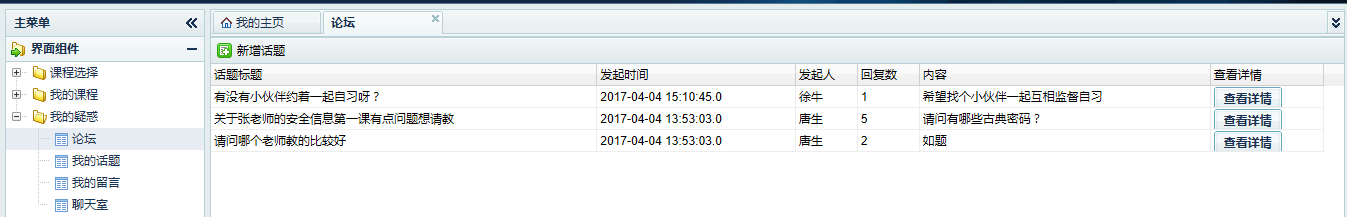


图6.14 所有留言话题界面

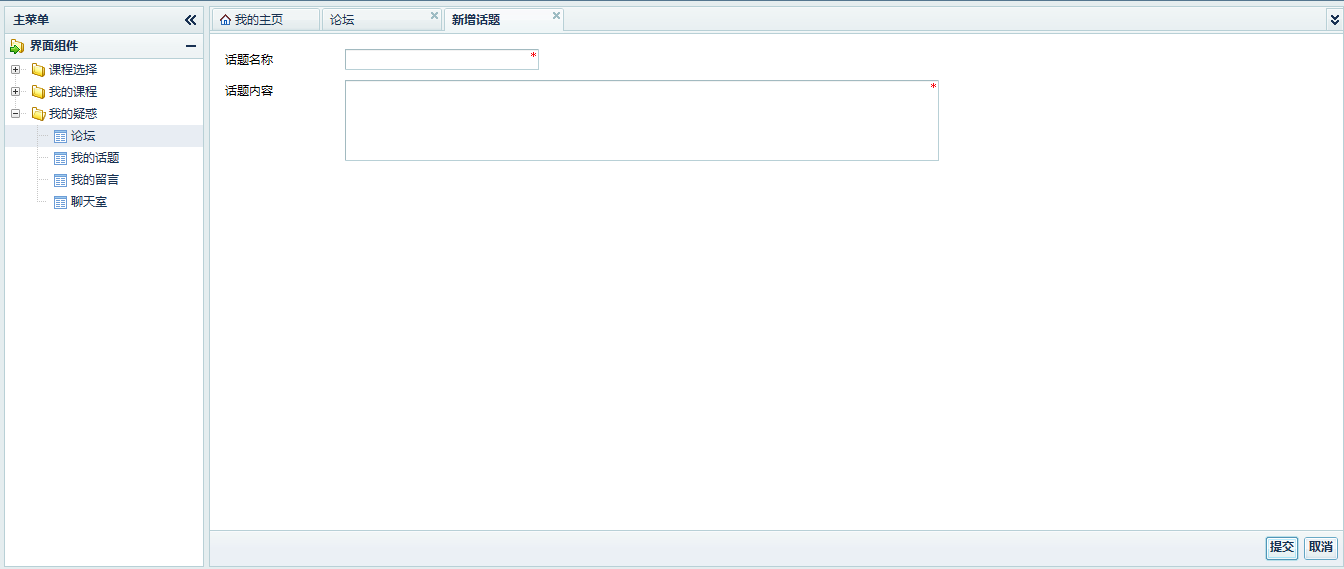


图6.15 新增话题界面

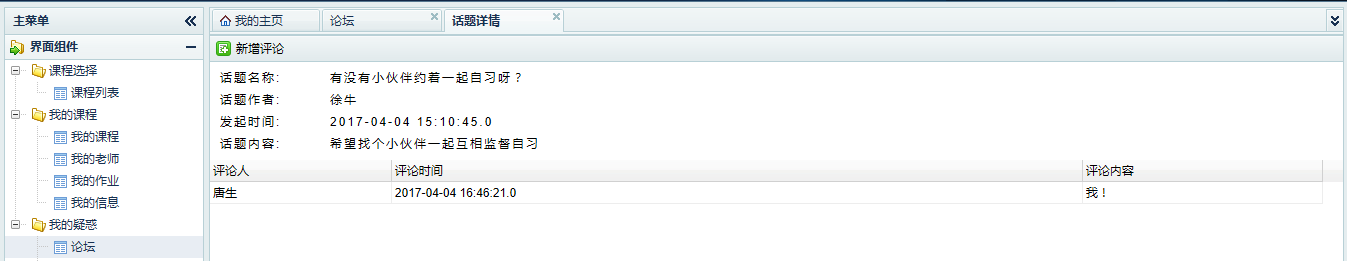


图6.16 话题详细信息界面



图6.17 新增评论界面

### 6.1.10教师/学生事务中心界面

图6.18为管理员删除留言话题界面。管理员在这里查看所有话题信息，并可以通过删除话题按钮删除选中话题。删除评论界面和该页面相似，在此不做赘述。

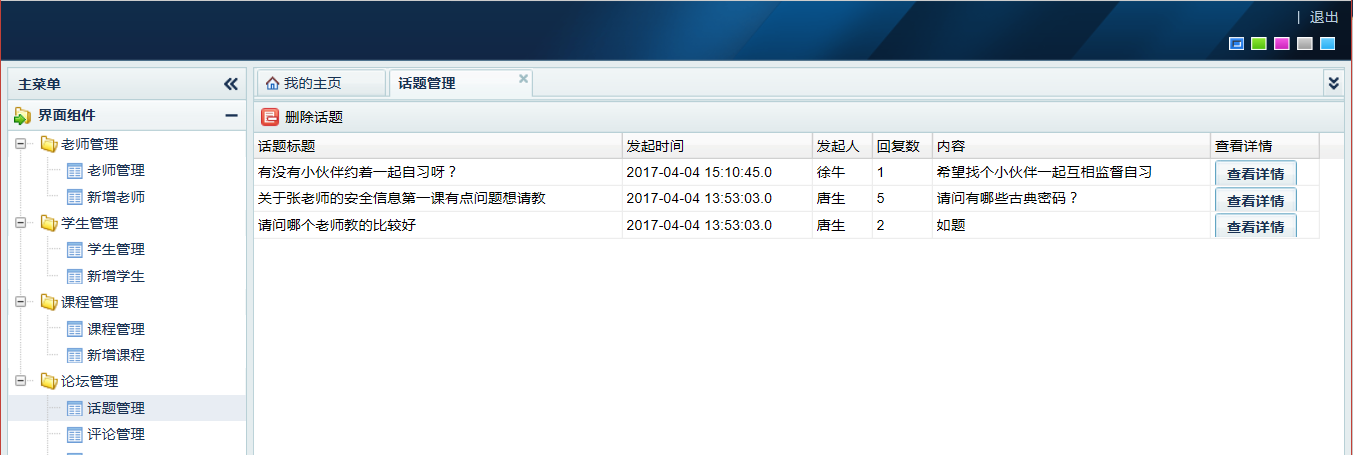


图6.18 删除留言话题界面

### 6.1.11四六级/图书借还信息界面

图6.19为教师或学生删除我的留言话题界面，教师或学生可以通过这个界面查看所有自己发表过的留言话题，并且选择删除或者查看详情。学生或教师删除评论界面同该界面相似，在此不做赘述。

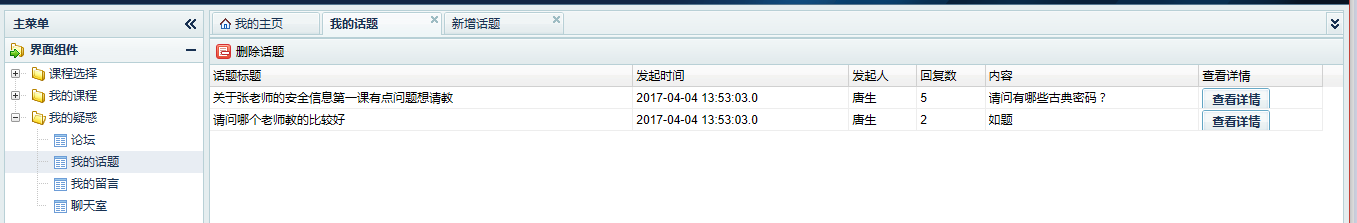


图6.19 删除我的留言话题界面

### 6.1.12管理员管理课程界面

图6.20为教师或学生进入聊天室界面，教师或学生可以通过填写一个昵称进入并开始本次聊天，进入聊天室后界面如图6.21所示。当用户结束聊天，点击退出聊天按钮，即可成功退出聊天，界面如图6.22所示。

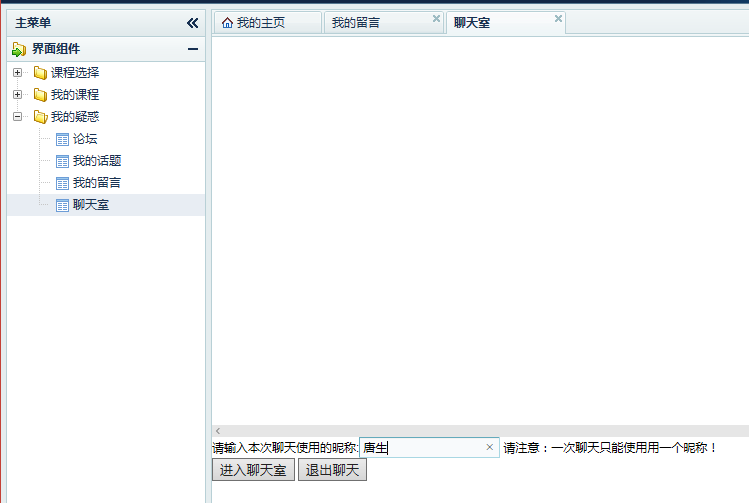


图6.20 进入聊天室界面



图6.21 聊天界面

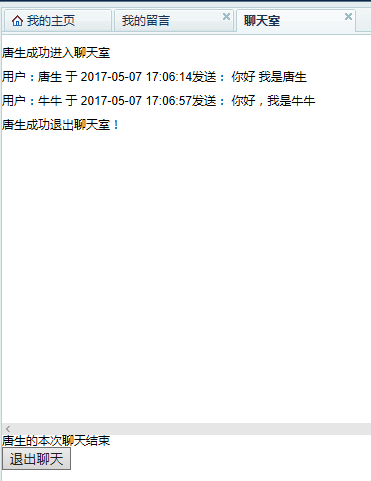


图6.21退出聊天室界面

## 6.2 测试环境和测试工具

1. 操作系统：windows2008及以上
2. CPU：1ghz
3. Ram：512
4. 网络畅通

## 6.3测试用例

本次系统测试主要采用单元测试。单元测试通常用于集中监测软件设计的模块，即其最小单元。一般来说，编码和单元测试属于软件过程中同一个阶段。程序员在编写出源程序并且通过基础语法检查之后，就可以参考系统详细设计的描述，对重要的执行通路进行单元测试，以便及时发现模块内部的错误并加以改正。单元测试可以应用计算机检测和人工检测两种不同类型的测试方法，两种不同的测试方法各有所长，互相补充。大多数单元测试主要使用白盒测试技术，可以并行的对多个模块的进行测试。

1. 用户登录

表6.1用户登录测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 用户登录 | | |
| 用例目的 | 测试合法用户是否可以正确登录 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，用户正确打开系统页面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户输入正确用户名及密码 | | 成功跳转相应界面 | 成功跳转相应界面 |
| 用户输入错误的用户名或密码 | | 提示用户名或密码错误 | 网页显示输入用户ID或密码不正确 |

1. 添加课程信息

表6.2 添加课程信息测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 课程信息添加 | | |
| 用例目的 | 测试管理员是否可以正常的添加课程信息 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，管理员用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击添加课程，并且填写所有必要信息。例：课程ID：1010；课程名称：密码学历史；教师ID：9001。并点击提交。 | | 提示提交成功 | 网页提示提交成功，并且正确更新数据库。 |
| 用户点击添加课程，但是部分必要信息为空。例：课程ID：；课程名称：密码学历史；教师ID：9001。并点击提交。 | | 提示必填信息为空 | 网页提示必填信息为空，数据库没有更新。 |
| 用户点击添加课程，但是课程ID重复。例：课程ID：1001；课程名称：密码学历史；教师ID：9001。并点击提交 | | 提示课程ID重复 | 网页提示ID重复，数据库没有更新。 |
| 用户点击添加课程，但是教师ID不存在。例：课程ID：1010；课程名称：密码学历史；教师ID：9018。并点击提交。 | | 提示教师不存在 | 网页提示教师不存在，数据库没有更新。 |
| 用户点击查看详情，填写修改信息。并点击提交。 | | 提示提交成功 | 网页提示提交成功，并正确更新数据库。 |

1. 编辑课程信息

表6.3 编辑课程测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 编辑课程信息 | | |
| 用例目的 | 测试教师是否可以正确编辑课程信息 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，教师用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击查看详情，填写修改信息。并点击提交。 | | 提示提交成功 | 网页提示提交成功，并正确更新数据库。 |

1. 课程课件/作业上传

表6.4 课程课件/作业上传测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 课程课件/作业上传 | | |
| 用例目的 | 测试教师或学生是否可以正确上传课程课件/作业 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，教师或学生用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击浏览，选择需要上传的文件路径并选择符合上传条件的相应文件。点击上传。 | | 提示上传成功。 | 网页提示上传成功，并正确更新数据库。 |
| 用户点击浏览，选择需要上传的文件路径并选择文件类型为.png格式的相应文件。点击上传。 | | 提示上传失败。 | 网页提示上传失败，数据库并未更新。 |
| 用户点击浏览，选择需要上传的文件路径并选择文件大小超过20M的相应文件。点击上传 | | 提示上传失败。 | 网页提示上传失败，数据库并未更新。 |

1. 课程报名

表6.5 课程报名测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 课程报名 | | |
| 用例目的 | 测试学生是否可以报名参加课程 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，学生用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击报名未报名过的课程。 | | 提示报名成功。 | 网页提示是否确认报名，点击确定提示报名成功，并正确更新数据库。 |
| 用户点击报名已报名的课程。 | | 提示已报名。 | 网页提示已报名该课程，数据库未更新。 |

1. 上传课件/作业查看

表6.6 上传课件/作业查看测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 上传课件/作业查看 | | |
| 用例目的 | 测试教师或学生是否可以正确查看课件/作业 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，教师或学生用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击查看课件/作业。 | | 显示该课程章节下已上传的课件/作业。 | 网页显示该课程章节下已上传的课件/作业。 |
| 用户点击下载课件/作业。 | | 成功下载课件/作业。 | 网页提示是否保存该课件/作业。点击保存即保存到浏览器默认保存路径；点击另存为则选择保存路径并可重命名下载文件。 |

1. 教师/学生信息管理

表6.7 教师/学生信息管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 教师/学生信息管理 | | |
| 用例目的 | 测试管理员是否可以正确查看所有教师/学生信息 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，管理员用户已正确的登录系统，并打开相应界面。 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击老师管理/学生管理。 | | 列出所有可用教师/学生信息。 | 网页列出所有可用教师/学生信息。 |
| 用户点击新增教师/学生。填写必要信息，例：教师ID：9012；教师姓名：吴雪；教师电话：13687634321。并点击提交。 | | 提示提交成功。 | 网页显示提交成功，并正确更新数据库。 |
| 用户点击新增教师/学生。未填写所有必要信息，例：教师ID：；教师姓名：吴雪；教师电话：13687634321。并点击提交。 | | 提示必须填写所有必填信息。 | 网页提示必须填写所有必填信息，数据库没有更新。 |
| 用户点击新增教师/学生。填写重复教师/学生ID，例：教师ID：9001；教师姓名：吴雪；教师电话：13687634321。并点击提交。 | | 提示ID重复。 | 网页提示ID已存在，数据库没有更新。 |

1. 个人信息修改

表6.8 个人信息测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 个人信息修改 | | |
| 用例目的 | 测试教师或学生是否可以正确修改个人信息 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，教师或学生用户已正确的登录系统，并打开相应界面。 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户修改个人信息并点击提交。 | | 提示提交成功。 | 网页提示提交成功，并正确更新数据库。 |
| 用户修改个人信息但使某个必填项为空并点击提交 | | 提示必须填写所有必填信息。 | 网页提示必须填写所有必填信息，数据库没有更新。 |

1. 所有课程信息查看

表6.9 所有课程信息查看测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 所有课程信息查看 | | |
| 用例目的 | 测试教师或学生是否可以正确查看所有课程信息 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，学生用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击课程列表。 | | 显示所有可用课程。 | 网页显示所有可用课程。 |
| 用户选择搜索条件并输入搜索关键词，例：搜索条件：课程ID；搜索关键词：1。并点击搜索。 | | 显示所有课程ID包含1的所有可用课程。 | 网页显示所有课程ID包含1的所有可用课程。 |
| 用户选择搜索条件并不输入搜索关键词，例：搜索条件：课程ID；搜索关键词：。并点击搜索。 | | 显示所有课程。 | 网页显示所有课程。 |
| 用户点击查看详情可查看该课程详情 | | 显示查看详情界面。 | 网页显示查看详情界面。 |

1. 已报名教师/学生信息查看

表6.10 已报名教师/学生信息查看测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 已报名教师/学生信息查看 | | |
| 用例目的 | 测试教师或学生是否可以正确查看已报名教师/学生信息 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，学生/教师用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击我的教师/学生。 | | 显示已报名课程教师/学生信息。 | 网页显示已报名课程教师/学生信息。 |
| 用户点击查看详情可查看该课程详情 | | 显示查看详情界面。 | 网页显示查看详情界面。 |

1. 留言新增

表6.11 留言新增测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 留言新增 | | |
| 用例目的 | 测试教师或学生是否可以正确新增留言 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，学生/教师用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户点击论坛，然后点击新增话题，并填写必要信息，例：话题名称：这里有一个问题；话题内容：我想请问一下信息安全是必修课嘛？并点击提交。 | | 显示提交成功 | 网页显示提交成功，并正确更新数据库。 |
| 用户点击论坛，然后点击新增话题，并未填写必要信息，例：话题名称：这里有一个问题；话题内容：；并点击提交。 | | 显示必须填写所有必填项。 | 网页显示必须填写所有必填项，数据库没有更新。 |

1. 评论新增

表6.12 评论新增测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 评论新增 | | |
| 用例目的 | 测试教师或学生是否可以正确新增留言 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，学生/教师用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户选择话题，点击新增评论并填写必要信息，例：评论内容：是的。并点击提交。 | | 显示提交成功。 | 网页显示提交成功，并正确更新数据库。话题回复数自动加1。 |
| 用户选择话题，点击新增评论并为填写必要信息，例：评论内容：。并点击提交。 | | 显示必须填写所有必填项。 | 网页显示必须填写所有必填项，数据库没有更新。 |

1. 所有留言/评论删除

表6.13 所有留言/评论删除测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 所有留言/评论删除 | | |
| 用例目的 | 测试管理员是否可以正确删除留言 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，管理员用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户选择话题/评论管理。 | | 显示所有话题/评论。 | 网页显示所有话题/评论。 |
| 用户选中话题/评论，并点击删除话题/评论。 | | 显示提交成功。 | 网页显示提交成功，并正确更新数据库。 |

1. 我的留言/评论删除

表6.14 我的留言/评论删除测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 我的留言/评论删除 | | |
| 用例目的 | 测试教师/学生是否可以正确删除本人留言 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，教师或学生用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户选择我的话题/评论。 | | 显示所有本人话题/评论。 | 网页显示所有本人话题/评论。 |
| 用户选中话题/评论，并点击删除话题/评论。 | | 显示提交成功。 | 网页显示提交成功，并正确更新数据库。 |

1. 聊天室聊天

表6.15 聊天室聊天测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 聊天室聊天 | | |
| 用例目的 | 测试教师/学生是否可以正确进入聊天室聊天 | | |
| 前提条件 | Windows 2008及以上操作系统，cpu 1ghz ram 512mb,网络畅通，教师或学生用户已正确的登录系统，并打开相应界面 | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 用户选择聊天室填写本次使用昵称并进入聊天室。 | | 成功进入聊天室。 | 成功进入聊天室，网页提示已进入聊天室并显示发送消息窗口。 |
| 用户进入聊天室后填写聊天内容并点击发表。 | | 成功发表聊天信息。 | 网页聊天记录信息正确显示发表的聊天信息。 |
| 用户点击退出聊天，并结束聊天 | | 成功退出聊天室。 | 退出聊天室，并且隐藏消息窗口。 |

## 6.3 测试结论

本次测试主要进行了单元性的功能测试，基本覆盖了所有功能需求所提及的所有功能，保证了各个功能的实现和完善，测试目标基本完成，通过所有测试。但是界面过于简单统一，可在日后进行进一步的修改。

# 第7章 结束语

## 7.1 工作总结

时间如梭，2013届入学的我们已经临近毕业，在校的四年间我们学习了各种不同的基础知识以及专业知识，而毕业设计以及毕业论文正是检验我们学习成果的一种好方式，这是我们大学生涯交上的最后一份答卷，是帮助我们重新审核自己大学四年学习内容的机会，是运用我们所掌握技能、提升自我的实战演练。

在2017年十二月份，同老师一起讨论拟定毕业设计的题目；在二月份基本确定毕业设计系统的基本结构并做课题分析，查找资料以及数据库设计等前期准备工作；在三月份完善毕业设计思路并学习可能用到的技术，研究需求分析以及可行性方案；在四月份动手实现毕业设计各种功能并进行测试确认系统是否完善；五月份撰写毕业设计论文并准备毕业答辩等事宜。

一个完整的准备及完成毕业设计过程要求我们融会贯通大学所学到的所有知识，训练我们独立思考，自我成长，克服困难，静心学习。我们在这个过程中温故知新，重拾可能已经遗忘的知识，仔细思索各个实施环境的所有细节，上网搜索补充自身的知识薄弱点，潜心研究遇到的困难。尽管在此期间会碰到很多意想不到的意外，可能会有计划之外的疏忽、技术上的难关、客观因素的阻挠，但是学会如何随机应变、及时解决问题也正是我们在其中学到的珍贵的技能之一。在遇到问题，解决问题的过程中，我们慢慢学会了发现，思考，总结，记录，反思，改正。

这次毕业设计让我感觉到我还有很多不足，学习是一件永无止境的事情，我们应该在之后的日子中继续学习，努力提高自己。不仅要享受学习的成果，也要享受学习的过程。

## 7.2 需改进的点

本次系统前台使用了dwz国产开源框架，它使得前台页面的开发更加容易便捷，但也使得使用这个框架时，限于对框架本身的不了解，局限了部分功能的实现。在前期准备时应该更多参考框架本身的使用手册以及理解开源代码。

本次系统使用了MVC最根本的Severlet结合JSP页面的方法实现。虽然较为基础通用，但是却不是现今主流使用框架，之后应该更多学习类似spring等现今主流框架。

# 参考文献

1. 陈显亭． Java程序设计[M].　长春: 吉林大学出版社, 2011
2. 蒋金楠． ASP.NET MVC4框架揭秘[M]. 北京：电子工业出版社, 2013
3. 王红．Java Web应用开发技术实用教程[M]．北京：中国水利水电出版社，2008
4. 张海藩，牟永敏．软件工程导论（第6版）[M]．北京：清华大学出版社，2013
5. 蔡剑，景楠． Java Web应用开发：J2EE和Tomcat[M]. 北京：清华大学出版社，2004
6. 王卫红．软件工程实践教程[M]. 北京：机械工程出版社, 2015
7. Eckel B. Thinking in Java. Upper Saddle River:Prentice Hall, 2006
8. Goodwill J. Pure Java Server Pages. Indianapolis Ind: Sams, 2000

# 致 谢

在这次完成毕业设计的过程中，我要感谢我的指导老师刘亚，感谢她在这整个过程中，一直给予我耐心指导。从立题到分析需求，从系统设计到实施构建，从完成编程到后期测试，刘亚老师一直都悉心为我解答遇到的疑惑和困难，教导我如何克服遇到的困难。另外，我还要感谢每一位传授过我知识的老师，在大学的四年间我从不同的老师那里学习到了各种知识，而这些知识也必将成为我未来工作的基石和生活的基础。其次，我要感谢我的同事和朋友以及家人，感谢他们在完成毕设的期间给予我的帮助和建议。

# 附录A: 主要源代码

1、ISclassdetailsearch.html

<script type="text/javascript">

$(function() {

$.ajax({

url : "/BN1/classdetailsearchtest",

dataType : "json",

type : "post",

async:false,

success : showresult,

error : function(){

alert("error");

}

});

});

function showresult(jsonto) {

if(jsonto!=null){

var str="<thead><tr><th>课程ID<th>课程名称<th>教师ID<th>教师姓名<th>查看详情</tr></thead><tbody>";

$.each( jsonto, function(index,content){

 str+="<tr><td>"+content.id+"</td><td>"+content.name+"</td><td>"+content.teacherid+"</td><td>"+content.teacher+"</td>";

str+="<td><a class='button' onclick='show("+content.id+")' target='navTab' href='ISaclassdetail.html' rel='classdetail'><span>查看详情</span></a></td>"

});

str+="</tbody>"

document.getElementById("table1").innerHTML=str;

}else{

alert("null");

}

}

function show(oid){

setCookie("id",oid);

}

function setCookie(id,value){

var exp = new Date();

exp.setTime(exp.getTime() + 10 \* 1000); //10s过期

document.cookie = id+"=" + encodeURIComponent(value) + ";expires=" + exp.toGMTString()+";path=/"; //to get cookie must reset path

return true;

}

function getsearch(){

var searchcon=document.getElementById('selectcons').value;

var searchvalue=document.getElementById('keywords').value;

setCookie("selection",searchcon);

setCookie("searchvalue",searchvalue);

}

</script>

<form action='' onsubmit='' class="pageForm required-validate"  method='post'>

<div class='searchBar'>

<table class='searchContent'>

<tr>

<td>

<select class='combox' name='key' id='selectcons'>

<option value=''>选择查询条件</option>

<option value='1'>课程ID</option>

<option value='2'>教师ID</option>

<option value='3'>课程名称</option>

</select>

</td>

<td>

搜索关键词：<input type='text' name='keyword' id='keywords'/>

</td>

</tr>

</table>

<div class='subBar'><ul>

<li><a class='button' onclick='getsearch()' href='ISclassdetailsearch.html' target='navTab' rel='page2'><span>搜索</span></a></li>

</ul>

</div>

</div>

</form>

<div id='div' style='width:1200px; height:900px; overflow:auto;'>

<table id='table1' class='table' width='100%' layoutH='56'>

</table>

</div>

</body>

</html>

2、ISCourselist.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<script type="text/javascript">

$(function() {

$.ajax({

url : "/BN1/courselist",

dataType : "json",

type : "post",

async:false,

success : showresult,

error : function(){

alert("error");

}

});

});

function showresult(jsonto) {

if(jsonto!=null){

var str="<thead><tr><th>课程id<th>课程名称<th>教师姓名</tr></thead><tbody>";

$.each( jsonto, function(index,content){

 str+="<tr target='sid\_user' rel='"+content.id+"'><td>"+content.id+"</td><td>"+content.name+"</td>";

str+="<td>"+content.teacher+"</td></tr>"

});

str+="</tbody>"

document.getElementById("table1").innerHTML=str;

}else{

alert("null");

}

}

</script>

<form method='post' action='' class='ageForm required-validate' onsubmit='return validateCallback(this, navTabAjaxDone)'>

<div class='panelBar'>

<ul class='toolBar'>

<li><a class='add' href='ISaddcourse.html' target='navTab'><span>新增课程</span></a></li>

<li><a class='delete' href='/BN1/deletecourseinfo?uid={sid\_user}' target='ajaxTodo' title='确定要删除该课程?'><span>删除课程</span></a></li>

</ul>

</div>

<div id='div'>

<table id='table1'class='table' width='100%' layoutH='56'>

</table>

</div>

</form>

</body>

</html>

3、ISteacherinfo.html

<script type="text/javascript">

$(function() {

$.ajax({

url : "/BN1/teacherinfo",

dataType : "json",

type : "post",

async:false,

success : showresult,

error : function() {

alert("error");

}

});

});

function showresult(jsonto) {

if(jsonto!=null){

var str="";

$.each( jsonto, function(index,content){

str+="<dl><dt><label>教师ID</label></dt><dd><input name='id' readonly='true' type='text' value='"+content.id+"' size='30' /></dd></dl>";

str+="<dl><dt><label>教师姓名</label></dt><dd><input name='name' readonly='true' type='text' value='"+content.name+"' size='30' /></dd></dl>";

str+="<dl><dt><label>密码</label></dt><dd><input name='password' type='text' value='"+content.password+"' size='30' /></dd></dl>";

str+="<dl><dt><label>联系电话</label></dt><dd><input name='tel' class='phone' type='text' value='"+content.tel+"' size='30' /></dd></dl>";

setCookie("assid",content.assid);

});

document.getElementById("div1").innerHTML=str;

}else{

alert("null");

}

}

</script>

4、index.html

<!DOCTYPE html>

<html>

<head>

<title>Echo Chamber</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

</head>

<body>

<!-- Server responses get written here -->

<div id="messages" style=" overflow:scroll; width:1150px; height:400px;"></div>

<div>

<div id="inputnickname">

<span>请输入本次聊天使用的昵称:</span><input type="text" id="nickname"/>

<span>请注意：一次聊天只能使用用一个昵称！</span>

</div>

<div>

<p id="shownickname"></p>

</div>

<p></p>

<!-- <span>请输入聊天内容:</span><input type="text" id="messageinput" /> -->

<div id="messagecontext" style="display:none;">

<p>请输入聊天内容:</p>

<textarea rows="3" cols="100" id="messageinput"></textarea>

<button type="button" onclick="send();" >发送消息</button>

</div>

</div>

<div>

<button type="button" onclick="openSocket();" id="inchatroom">进入聊天室</button>

<button type="button" onclick="closeSocket();" >退出聊天</button>

</div>

<!-- Script to utilise the WebSocket -->

<script type="text/javascript">

var webSocket;

var messages = document.getElementById("messages");

function openSocket(){

var nickname=document.getElementById("nickname").value;

if(nickname==""){

alert("请输入本次聊天的昵称");

}else{

if(webSocket !== undefined && webSocket.readyState !== WebSocket.CLOSED){

//writeResponse("已进入聊天室");

return;

}

var target=document.getElementById("messagecontext");

target.style.display="";

document.getElementById("inputnickname").style.display="none";

document.getElementById("shownickname").innerHTML="本次聊天昵称："+nickname;

document.getElementById("shownickname").style.display="";

document.getElementById("inchatroom").style.display="none";

// Create a new instance of the websocket

webSocket = new WebSocket("ws://localhost:8081/chattest/echo/"+nickname);

/\*\*

\* Binds functions to the listeners for the websocket.

\*/

webSocket.onopen = function(event){

// For reasons I can't determine, onopen gets called twice

// and the first time event.data is undefined.

// Leave a comment if you know the answer.

if(event.data === undefined)

return;

writeResponse(event.data);

};

webSocket.onmessage = function(event){

writeResponse(event.data);

};

webSocket.onclose = function(event){

writeResponse(nickname+"成功退出聊天室！</br>");

document.getElementById("shownickname").innerHTML=nickname+"的本次聊天结束";

var target=document.getElementById("messagecontext");

target.style.display="none";

};

}

}

/\*\*

\* Sends the value of the text input to the server

\*/

function send(){

var text = document.getElementById("messageinput").value;

document.getElementById("messageinput").value="";

webSocket.send(text);

}

function closeSocket(){

webSocket.close();

}

function writeResponse(text){

messages.innerHTML += "<br/>" + text;

}

</script>

</body>

</html>

5、thread.java

//输出thread

public static ResultSet importthreadsbybbsid(String bbsid) {

Connection con = sqlconnection.getConnection();

ResultSet rs=null;

try{

String sql = "select t.\*,a.name from thread as t left join (select \* from teacher union select \* from student) as a on t.auth=a.ID and t.retired=1 and t.bbsid="+bbsid+" order by t.time desc";

Statement stmt = con.createStatement();

rs = stmt.executeQuery(sql);

System.out.print("import course ok! sql："+sql);

}catch (Exception e) {

System.out.print("error"+e);

e.printStackTrace();

}

return rs;

}

//输出本人评论

public static ResultSet importthreadbyid(String userid) {

Connection con = sqlconnection.getConnection();

ResultSet rs=null;

try{

String sql = "select t.\*,b.title from (select \* from thread where retired =1 and auth="+userid+") as t inner join (select \* from bbs where retired =1) as b on t.bbsid=b.ID order by t.time desc";

Statement stmt = con.createStatement();

rs = stmt.executeQuery(sql);

}catch (Exception e) {

System.out.print("error"+e);

e.printStackTrace();

}

return rs;

}

//新建评论

public int savethread(String titleid,thread thread) {

System.out.println("in thread->function savethread");

Connection con = sqlconnection.getConnection();

String sql = "insert into thread(main,time,auth,retired,bbsid) values(?,?,?,?,?)";

try{

con.setAutoCommit(false);

PreparedStatement pstmt = con.prepareStatement(sql);

System.out.println("in thread->function savebbs");

Date a = new Date();

SimpleDateFormat sd = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

//session.setAttribute("timereply",sd.format(a));

String time=sd.format(a);

System.out.print(sd.format(a));

//System.out.println(bbs.getId());

//pstmt.setString(1, bbs.getId());

pstmt.setString(1, thread.getMain());

pstmt.setString(2, thread.getTime());

pstmt.setString(3, thread.getAuth());

pstmt.setString(4, "1");

pstmt.setString(5, titleid);

int i = pstmt.executeUpdate();

System.out.print("add");

if (i == 1) {

con.commit();

int flag=autoaddbbsnum(titleid);

if(flag==1){

return 1;

}else{

return 2;

}

}

}catch (Exception e) {

System.out.println("in thread->function savethread:fail");

try {

con.rollback();

} catch (SQLException e1) {

System.out.print("error2"+e1);

e1.printStackTrace();

}

}finally {

try {

con.close();

} catch (SQLException e) {

}

}

return 0;

}

//自增长回复数

public int autoaddbbsnum(String bbsid){

//调用javabean连接数据库

Connection con = sqlconnection.getConnection();

//ResultSet rs=null;

try{

String sql = "update bbs set num=num+1 where id="+bbsid;

Statement stmt = con.createStatement();

//rs = stmt.executeQuery(sql);

stmt.executeUpdate(sql);

return 1;

}catch (Exception e) {

System.out.print("error"+e);

e.printStackTrace();

}

return 0;

}

public int autominbbsnum(String bbsid){

//调用javabean连接数据库

Connection con = sqlconnection.getConnection();

//ResultSet rs=null;

try{

String sql = "update bbs set num=num-1 where id="+bbsid;

Statement stmt = con.createStatement();

//rs = stmt.executeQuery(sql);

stmt.executeUpdate(sql);

return 1;

}catch (Exception e) {

System.out.print("error"+e);

e.printStackTrace();

}

return 0;

}

//删除评论

public int deletethread(String titleid,String id) {

//调用javabean连接数据库

Connection con = sqlconnection.getConnection();

//ResultSet rs=null;

try{

String sql = "update thread set retired=0 where id="+id;

Statement stmt = con.createStatement();

stmt.executeUpdate(sql);

int flag=autominbbsnum(titleid);

if(flag==1){

return 1;

}

else{

return 2;

}

}catch (Exception e) {

System.out.print("error"+e);

e.printStackTrace();

}

return 0;

}

public static ResultSet importallthread() {

Connection con = sqlconnection.getConnection();

ResultSet rs=null;

try{

String sql = "select t.\*,b.title from (select \* from thread where retired =1 ) as t inner join (select \* from bbs where retired =1) as b on t.bbsid=b.ID order by t.time desc";

Statement stmt = con.createStatement();

rs = stmt.executeQuery(sql);

//System.out.print("import course ok!");

}catch (Exception e) {

System.out.print("error"+e);

e.printStackTrace();

}

return rs;

}

}

6、classdetailsearchtest.java

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

//中文乱码

response.setContentType("text/html;charset=UTF-8");

response.setCharacterEncoding("UTF-8");

int searchflag=0;

String searchcon=null;

String searchvalue=null;

Cookie[] cookies = request.getCookies();

if (cookies != null && cookies.length > 0) { //如果没有设置过Cookie会返回null

for (Cookie cookie : cookies){

String cookiename=cookie.getName();

if(cookiename.equals("selection")){

searchcon= URLDecoder.decode(cookie.getValue(),"utf-8");

}else if(cookiename.equals("searchvalue")){

searchvalue= URLDecoder.decode(cookie.getValue(),"utf-8");

}

}

}else{

System.out.println("cookie null");

}

if(searchcon!=null&searchvalue!=null){

searchflag=1;

}

//get session

HttpSession session = request.getSession(false);

String userid=null;

userid=(String)session.getAttribute("userid");

//json封装

JSONArray array=new JSONArray();

JSONObject object=new JSONObject();

int ID,num;

String name,teacher;

try{

ResultSet courseinfo=null;

if(searchflag==0){

courseinfo=course.importcourses(userid);

}else if(searchflag==1){

System.out.println("in classdetail: import with search condition");

//have to convert to exactly sql element name

String searchcond="";

switch(searchcon){

case "1" : searchcond = "id";break;

case "2" : searchcond = "teacher";break;

case "3" : searchcond="name";break;

}

System.out.println("searchcon:"+searchcon);

courseinfo = course.importfilitercourses(searchcond,searchvalue);

}

while(courseinfo.next()){

object.put("id", courseinfo.getInt(1));

object.put("name", courseinfo.getString(2));

object.put("teacherid", courseinfo.getString(3));

object.put("teacher", courseinfo.getString(14));

object.put("statusCode", "200");

object.put("message", "success");

array.add(object);

}

int totalamount=courseinfo.getRow();

object.put("totalamount", totalamount);

}catch(Exception e1){

System.out.println("in classdetail error: "+e1);

}

response.getOutputStream().write(array.toString().getBytes("UTF-8"));

response.setContentType("text/json; charset=UTF-8");

System.out.println("end of calssdetailtest");

}

7、uploadlist.java

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

//中文乱码

response.setContentType("text/html;charset=UTF-8");

response.setCharacterEncoding("UTF-8");

//json封装

JSONArray array=new JSONArray();

JSONObject object=new JSONObject();

int ID,num;

String classid="";

String assnum="";

Cookie[] cookies = request.getCookies();

if (cookies != null && cookies.length > 0) { //如果没有设置过Cookie会返回null

for (Cookie cookie : cookies){

String cookiename=cookie.getName();

if(cookiename.equals("id")){

classid= cookie.getValue();

}else if(cookiename.equals("assnum")){

assnum= cookie.getValue();

}

}

}

else{

}

String assid="";

assid+=classid;

assid+="0";

assid+=assnum;

try{

ResultSet uploadlist=null;

uploadlist=assignment.importassignmentfilelist(assid);

while(uploadlist.next()){

object.put("id", uploadlist.getString(1));

object.put("url", uploadlist.getString(2));

object.put("filename", uploadlist.getString(3));

object.put("courseid", classid);

object.put("assnum", assnum);

array.add(object);

}

}catch(Exception e1){

}

response.getOutputStream().write(array.toString().getBytes("UTF-8"));

response.setContentType("text/json; charset=UTF-8");

}

# 

# 附录B: 软件使用说明书

## 1.软件描述

本系统是为了给教师和学生一个相互沟通交流的平台，帮助学生解决课堂上遇到的困难，使得教师更好的安排课程。

### 1.1 系统主要功能

本系统主要功能如下：

1. 登录：根据角色权限不同，登录后可以完成不同角色功能。
2. 课程信息管理：教师用户可以上传或下载课件、下载作业以及编辑课程；管理员用户可以及管理课程；学生用户可以下载课件、上传或下载自己的作业。
3. 用户信息管理： 管理员用户可以管理教师或学生信息；教师或学生用户可以管理个人信息。
4. 留言信息管理：教师用户可以发表或删除话题以及评论、加入聊天室聊天；管理员用户可以删除话题和评论；学生用户发表或删除话题以及评论并加入聊天室。

## 2.软件的安装

### 2.1 系统环境需求

1. 基本硬件平台

 CPU：PII 或更高

 内存：412MB 或更高

 显示器：17’或更高

1. 基本软件平台

 操作系统：Windows Server 2003 或Windows XP 以上

 数据库：SQLSERVER 2008

 服务器：IIS5.0 或更高

 分辨率：1024\*768 或更高

### 2.2 使用说明

在系统部署好以后，管理员或者用户可以直接打开软件，进入主页即可使用本系统。