目录

[基于SSM框架的软件工程课程web端学生选课管理系统的设计与实现 0](#_Toc25456)

[1.引言 0](#_Toc29974)

[2.相关理论基础 0](#_Toc16966)

[2.1 JSP技术 0](#_Toc31678)

[2.2 MySQL技术 1](#_Toc13808)

[2.3 JAVA技术 2](#_Toc13942)

[2.4 SSM框架 3](#_Toc16065)

[2.5Tomcat简介 3](#_Toc701)

[3.系统总体设计 4](#_Toc28900)

[3.1系统功能层次图 4](#_Toc17935)

[3.1.2设计目标 6](#_Toc15838)

[1教师管理模块 6](#_Toc11758)

[2学生登录管理模块 6](#_Toc24309)

[3.2数据库设计 7](#_Toc28975)

[3.2.1.数据库分析 7](#_Toc4034)

[1.实体的分析 7](#_Toc9398)

[2.实体的属性 7](#_Toc645)

[3.数据表设计 8](#_Toc1795)

[3.2.2 SQL语句实现 9](#_Toc1015)

[4.具体设计分析 11](#_Toc322)

[4.1教师模块设计 11](#_Toc12960)

[4.1.1添加学生信息功能 11](#_Toc20920)

[4.1.2添加新课程功能 11](#_Toc3590)

[4.1.3修改学生信息功能 12](#_Toc13000)

[4.1.5修改课程信息功能 12](#_Toc1071)

[4.2学生模块设计 20](#_Toc10977)

[4.2.1学生选课功能 20](#_Toc6216)

[4.2.2 输入学生的编信息功能 21](#_Toc21776)

[4.2.3查看已选课程功能 21](#_Toc21359)

[4.2.4删除已选课程功能 22](#_Toc12057)

[4.3系统登录界面 28](#_Toc30969)

[4.3.1录入学生信息界面 29](#_Toc8500)

[4.3.2修改学生信息界面 29](#_Toc418)

[4.3.3学生选课界面 30](#_Toc21737)

[5.系统测试 31](#_Toc15758)

[6.结论 31](#_Toc23460)

# 基于SSM框架的软件工程课程web端学生选课管理系统的设计与实现

# 1.引言

随着计算机网络技术发展，Web数据库技术已经成为应用最为广泛的网站架构基础技术在应用系统中，Web提供了与用户进行通信联络的有效手段，利用Web技术，实现Web服务器与数据库系统的连接，完成对数据的处理与查询，用户可以通过操作简单易学的浏览器来查询处理所需要的各种数据。

学生信息管理系统是一个教育单位不可缺少的部分，它的内容对于学校的决策者和管理者来说都至关重要，所以学生信息管理系统应该能够为用户提供充足的信息和快捷的查询手段。但一直以来人们使用传统人工的方式管理文件档案，这种管理方式存在着许多缺点，如:效率低、保密性差，另外时间一长，将产生大量的文件和数据，这对于查找、更新和维护都带来了不少的困难。

随着科学技术的不断提高，计算机科学日渐成熟，其强大的功能已为人们深刻认识，它已进入人类社会的各个领域并发挥着越来越重要的作用。作为计算机应用的一部分，使用计算机对学生信息进行管理，具有手工管理所无法比拟的优点。例如:检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点能够极大地提高学生信息管理的效率，也是企业的科学化、正规化管理，与世界接轨的重要条件。

# 2.相关理论基础

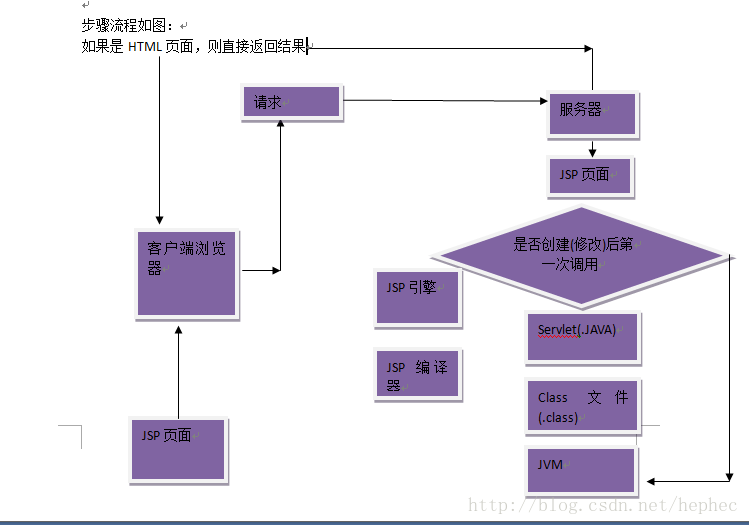
## 2.1 JSP技术

JSP全称Java Server Pages，是一种动态网页开发技术。它使用JSP标签在HTML网页中插入Java代码。标签通常以<%开头以%>结束。JSP是一种Java servlet，主要用于实现Java web应用程序的用户界面部分。网页开发者们通过结合HTML代码、XHTML代码、XML元素以及嵌入JSP操作和命令来编写JSP。JSP通过网页表单获取用户输入数据、访问数据库及其他数据源，然后动态地创建网页。JSP标签有多种功能，比如访问数据库、记录用户选择信息、访问JavaBeans组件等，还可以在不同的网页中传递控制信息和共享信息。

与ASP相比,JSP有两大优势。首先，动态部分用Java编写，而不是VB或其他MS专用语言，所以更加强大与易用。第二点就是JSP易于移植到非MS平台上。与纯 Servlet 相比：JSP可以很方便的编写或者修改HTML网页而不用去面对大量的println语句。与SSI相比：SSI无法使用表单数据、无法进行数据库链接。与JavaScript相比：虽然JavaScript可以在客户端动态生成HTML，但是很难与服务器交互，因此不能提供复杂的服务，比如访问数据库和图像处理等等。与静态HTML相比：静态HTML不包含动态信息。一个JSP文件的执行过程可以分为以下几个步骤：1、浏览器客户端向将JSP引擎发出对JSP页面的请求。2 、JSP引擎将对应的JSP页面转化成Java Servlet源代码3 、JSP引擎编译源代码生成.class文件4 、JSP引擎加载运行对应的.class文件生成响应的结果页面5 、JSP引擎吧响应的输出结果发送到浏览器端JSP引擎管理JSP页面生命周期的两个阶段：转化阶段(Translation Phase)和执行阶段(Execution Phase)。

1、 每当一个对JSP页面的请求到来时，JSP引擎检验JSP页面的语法是否正确，将JSP页面转换为Servlet源文件，然后调用javac工具类编译Servlet源文件生成字节码文件，这一阶段是转换阶段。

2、 Servlet引擎加载转换后的Servlet类，实例化一个对象处理客户端的请求。在请求处理完成后响应对象被JSP引擎吸收，引擎将HTML格式的响应信息发送到客户端，这一阶段是执行阶段。

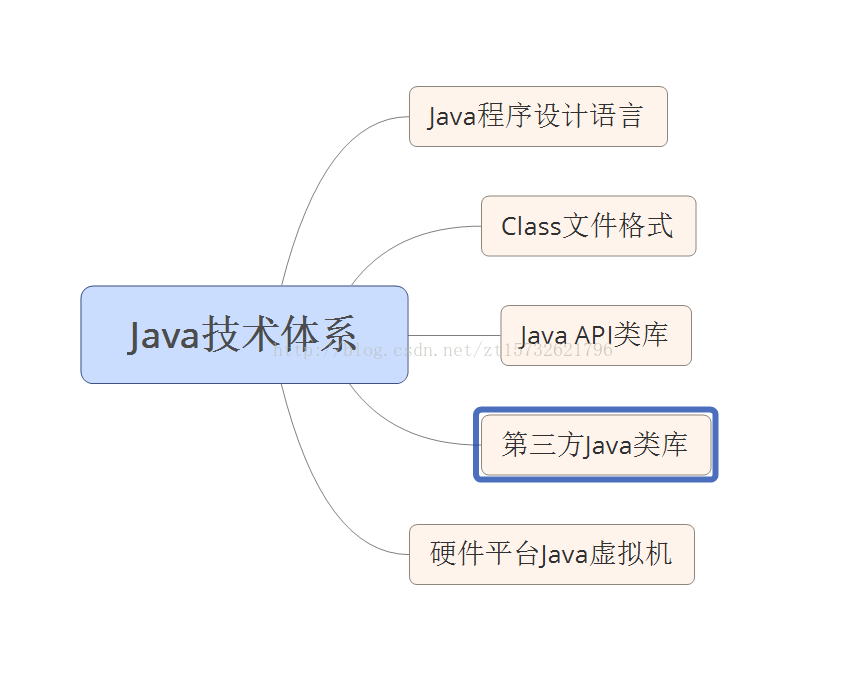


## 2.2 MySQL技术

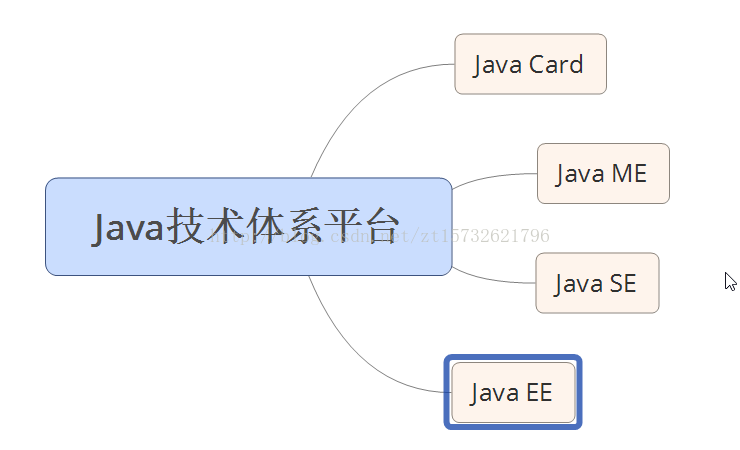
MySQL是一个精巧的SQL数据库管理系统，而且是开源的数据管理系统。由于它的强大功能、灵活性、丰富的应用编程接口（API）以及精巧的系统结构，受到了广大自由软件爱好者甚至是商业软件用户的青睐，特别是与Apache和PHP/PERL结合，为建立基于数据库的动态网站提供了强大动力。MySQL是一个真正的多用户、多线程SQL数据库服务器。SQL（结构化查询语言）是世界上最流行的和标准化的数据库语言。MySQL是以一个客户机/服务器结构的实现，它由一个服务器守护程序mysqld和很多不同的客户程序和库组成。SQL是一种标准化的语言，它使得存储、更新和存取信息更容易。例如，你能用SQL语言为一个网站检索产品信息及存储顾客信息，同时MySQL也足够快和灵活以允许你存储记录文件和图像。MySQL 主要目标是快速、健壮和易用。它能处理与任何可不昂贵硬件平台上提供数据库的厂家在一个数量级上的大型数据库，但速度更快，MySQL就开发出来。自1996年以来，大家一直都在使用MySQL，其环境有超过 40 个数据库，包含 10,000个表，其中500多个表超过7百万行，这大约有100 个吉字节(GB)的关键应用数据。MySQL建立的基础是业已用在高要求的生产环境多年的一套实用例程。尽管MySQL仍在开发中，但它已经提供一个丰富和极其有用的功能集。MySQL是一种开放源代码的关系型数据库管理系统（RDBMS），使用最常用的数据库管理语言--结构化查询语言（SQL）进行数据库管理。MySQL是开放源代码的，因此任何人都可以在General Public License的许可下下载并根据个性化的需要对其进行修改。MySQL因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需要事务化处理的情况下，MySQL是管理内容最好的选择。MySQL，虽然功能未必很强大，但因为它的开源、广泛传播，导致很多人都了解到这个数据库。它的历史也富有传奇性。

## 2.3 JAVA技术

Java是由Sun Microsystems公司于1995年5月推出的Java面向对象程序设计语言和Java平台的总称Java技术既是一种高级的面向对象的编程语言，也是一个平台。Java技术基于Java虚拟机（Java virtual machine，JVM 语言与底层软件和硬件之间的一种转换器）的概念。从传统意义上讲，Sun官方定义的Java技术体系的组成部分包括



Java技术体系可分为4个平台：



其中Java EE：支持使用多层架构的企业的应用的Java平台，提供了Java SE API并做了大量扩充，并提供了相关的部署支持（J2EE）。它有很多特性: Java语言是简单的, Java语言是面向对象的：Java语言是分布式的：Java语言是健壮的：Java语言是安全的：Java语言是体系结构中立的：Java语言是可移植的：Java语言是解释型的：Java是高性能的：Java语言是多线程的Java语言是动态的。

## 2.4 SSM框架

SSM（Spring+SpringMVC+MyBatis）框架集由Spring、MyBatis两个开源框架整合而成（SpringMVC是Spring中的部分内容）。常作为数据源较简单的web项目的框架。

Spring就像是整个项目中装配bean的大工厂，在配置文件中可以指定使用特定的参数去调用实体类的构造方法来实例化对象。也可以称之为项目中的粘合剂。Spring的核心思想是IoC（控制反转），即不再需要程序员去显式地`new`一个对象，而是让Spring框架帮你来完成这一切。SpringMVC在项目中拦截用户请求，它的核心Servlet即DispatcherServlet承担中介或是前台这样的职责，将用户请求通过HandlerMapping去匹配Controller，Controller就是具体对应请求所执行的操作。SpringMVC相当于SSH框架中struts。mybatis是对jdbc的封装，它让数据库底层操作变的透明。mybatis的操作都是围绕一个sqlSessionFactory实例展开的。mybatis通过配置文件关联到各实体类的Mapper文件，Mapper文件中配置了每个类对数据库所需进行的sql语句映射。在每次与数据库交互时，通过sqlSessionFactory拿到一个sqlSession，再执行sql命令。

## 2.5Tomcat简介

Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由Apache、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 5支持最新的Servlet 2.4 和JSP 2.0 规范。

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。

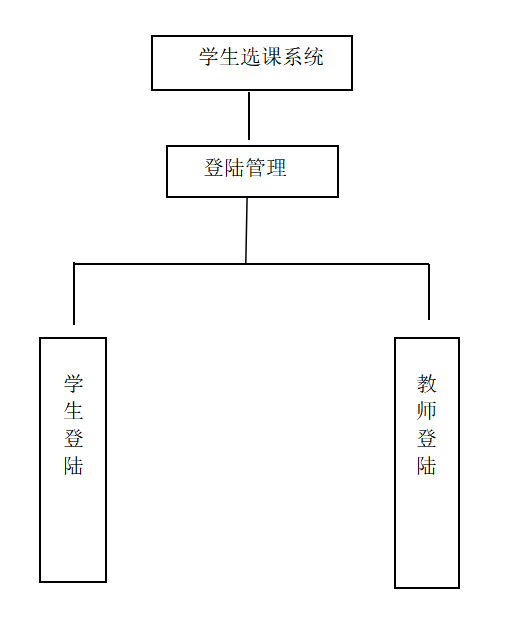
Tomcat 实际上运行JSP 页面和Servlet。另外，Tomcat和IIS等Web服务器一样，具有处理HTML页面的功能，另外它还是一个Servlet和JSP容器，独立的Servlet容器是Tomcat的默认模式。不过，Tomcat处理静态HTML的能力不如Apache服务器。

我们组这次用的是Tomcat 8，它最大的变化是资源Resource。Tomcat 8的资源进行了重构，可以更好地支持外部资源。原先的别名Aliases、VirtualLoader、VirtualDirContext、JAR资源和外部仓库等，现在都以单个的、一致的方法进行配置。这个新的资源的实现也可以用于主WAR包，并作为其它多个Web应用程序的基础。

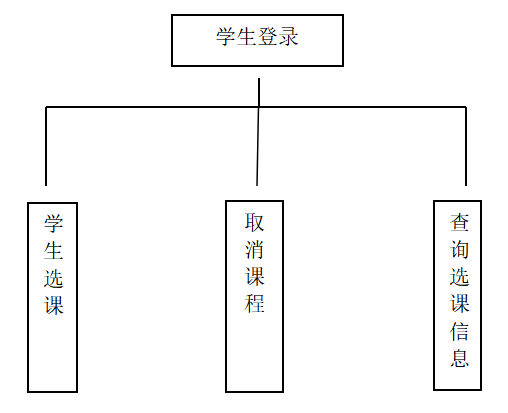
# 3.系统总体设计

## 3.1系统功能层次图

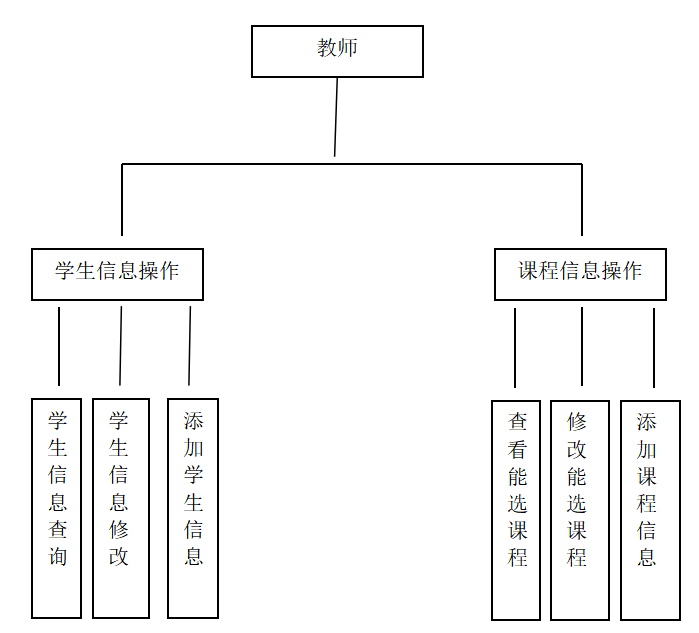
学生信息管理系统是一个典型的数据库开发应用程序,系统是和数据库相结合.,进入系统,有三个用户,分别是学生、教师和管理员,



**图3.1登陆层次图**



**图3-2学生管理层次图**

 **图3-3教师管理层次图**

## 3.1.2设计目标

学生信息管理系统为用户提供信息交流的交互平台,让老师、学生间建立良好联系,主要实现如下目标.

## 1教师管理模块

1.基本信息录入,对系统的所有基本信息进行管理操作.

2.学生信息操作,对学生的信息进行管理操作.

3.教师管理操作,对所教课程和听课通知进行查询操作

## 2学生登录管理模块

1.学生管理操作,对学生自己选课和删除已选课程进行管理操作.

2.系统中数据库起着非常重要的作用,不能随意存取或改动,因此必须对数据库的存取和改动进行严格的控制,对系统数据进行有效的保护,防止对数据库的非法操作,防止计算机病毒对数据的破坏,定期对数据库进行备份.

## 3.2数据库设计

## 3.2.1.数据库分析

## 1.实体的分析

根据学生信息管理系统的总体设计,我们将这个系统分为三大模块.由此我们可以首先确定二大实体：教师和学生,而由于用户可以进行选课、查询等操作.所以确定该系统共有以下实体：

1.教师实体.

2.学生实体.

3.课程实体.

4.选课实体.

## 2.实体的属性

该系统涉及到的实体分别为教师实体,学生实体.根据系统需求,我们将以下实体的属性确定如下：

|  |  |
| --- | --- |
| **数据结构** | **组 成** |
| 教 师 | 教师编号、密码、姓名 |
| 学 生 | 学号、姓名、密码、班级、性别。 |
| 课 程 | 课程编号、课程名称、授课教师、课程学分 |
| 选课关系 | 课程编号、学号、课程学分、上课地点、成绩 |

**表4-1实体属性表**

1.教师实体,包含教师的信息,详细属性如下：

教师实体，包含教师的基本信息，详细属性如下：

（1）教师号.

（2）教师姓名.

（3）登录密码.

3.学生实体,包含学生的基本信息,详细属性如下：

（1）学生号.

（2）学生姓名.

（3）密码.

（4）班级.

（5）性别.

4课程实体,包含课程的信息,详细属性如下：

（1）课程号.

（2）课程名称.

（3）课程学分.

## 3.数据表设计

逻辑结构是独立于任何一种数据模型的信息结构.逻辑结构的任务是把概念结构设计阶段设计好的基本E-R图转化为宜选用的DBMS所支持的数据模型相符合的逻辑结构,并对其进行优化.

定义每个表的数据类型以及字段限制,使数据库达到一定的完整性.每个表的物理设计如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **数据项** | **含义** | **别名** | **类型** | **长度** | **取值范围** |
| 教师编号 | 唯一表示教师 | TeaNo | Varchar | 11 | —— |
| 学号 | 唯一表示学生 | StuNo | Varchar | 11 | —— |
| 教师密码 | 教师用户登录验证 | Teapassword | Varchar | 20 | -—— |
| 学生密码 | 学生用户登录验证 | Stupassword | Varchar | 20 | —— |
| 姓名 | 老师、学生的属性 | Name | Varchar | 11 | —— |
| 班级 | 学生所属班级 | class | Varchar | 11 | —— |
| 课程学分 | 课程属性，学生要修够一定学分 | Creait | Float | 5 | 0-5 |
| 教室 | 学生上课地点 | Classroom | Varchar | 11 | —— |
| 授课教师 | 课程由老师教授 | Teacher | Varchar | 11 | —— |

## 3.2.2 SQL语句实现

（1）教师信息表创建：

CREATE TABLE `teacher` (

`TNo` int(11) NOT NULL AUTO\_INCREMENT,

`Tname` varchar(30) DEFAULT NULL,

`Tpassword` varchar(30) DEFAULT NULL,

PRIMARY KEY (`TNo`)

)

（2）课程信息表创建：

CREATE TABLE `course` (

`CNo` int(11) NOT NULL AUTO\_INCREMENT,

`Cname` varchar(30) NOT NULL,

`Ccredit` float(5,1) DEFAULT '0.0',

PRIMARY KEY (`CNo`)

)

（3）学生信息表创建：

CREATE TABLE `student` (

`SNo` int(11) NOT NULL AUTO\_INCREMENT,

`Sname` varchar(30) NOT NULL,

`Spassword` varchar(30) NOT NULL,

`Sclass` varchar(30) DEFAULT '',

`Ssex` varchar(10) DEFAULT '',

PRIMARY KEY (`SNo`)

)

# 4.具体设计分析

本学生信息管理系统主要模块,即用户登录模块、用户操作模块,用户操作模块又分为学生操作模块、教师操作模块和管理员操作模块.下面将显示系统的主要功能模块.

## 4.1教师模块设计

学生信息管理系统前台登陆模块只有一个登陆窗口分为学生、老师、两个级别登陆,系统在后台程序有自动限制设置,可以自动识别登陆者的限制.其系统登陆模块算法如下：

1 输入用户名和密码;

2.判断用户名和密码是否匹配

（1）若提示输入信息错误,则返回登陆界面重新登陆；

（2）否则进入系统.

## 4.1.1添加学生信息功能

学生信息管理系统对学生信息的管理非常重要,其中对信息的录入是系统最为关键的地方,以往我们管理学生信息的时候都是手工操作,而随社会不断发展的今天计算机的应用已全部取带了手工操作的方法,利用计算机可以方便的录入各类信息,进行高效的管理.学生信息的录入就是利用计算机通过程序读入数据库,录入学生信息模块算法描述如下：

1.教师登陆后,录入学生信息模块；

2.没有错后,学生信息添加成功.

## 4.1.2添加新课程功能

学生选课模块主要是给学生选择课程的功能,本功能主要由教师在后台发布自己的课程信息进入数据库,然后学生登录系统,可以看见所有发布的课程信息,然后学生再进行选择,学生选择的课程信息会自动进入后台数据库,其学生选课模块算法描述如下：

1．教师成功登陆；

2．点击添加新课程；

3．输入课程号；

4．输入课程名；

5．点击提交按钮,完成添加新课程操作.

## 4.1.3修改学生信息功能

此信息修改模块主要是给学生和教师建立信息库,方便教师查询操作,信息修改模块算法描述如下：

1．教师成功登陆；

2．点击修改学生信息；

3．输入学生的编号；

4．输入正确,进入修改页面进行相关修改；

## 4.1.5修改课程信息功能

此信息修改模块主要是给课程和教师建立信息库,方便教师查询操作,信息修改模块算法描述如下：

1．教师成功登陆；

2．点击修改课程信息；

3．输入课程的编号；

4．输入正确,进入修改页面进行相关修改.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class DaoQuery {

/\*

\* 根据学号查询选课情况

\* \*/

public List<QueryResult> selectSno(int sno){

List<QueryResult> list = null;

try{

list = new ArrayList<QueryResult>();

String sql = "select student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +

"where student.sno=sc.sno and sc.cno=course.cno and student.sno=?;";

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, sno);

ResultSet rs = pst.executeQuery();

while(rs.next()){

QueryResult qrs = new QueryResult();

qrs.setSNo(rs.getInt("sno"));

qrs.setCNo(rs.getInt("cno"));

qrs.setSname(rs.getString("sname"));

qrs.setCname(rs.getString("cname"));

qrs.setSclass(rs.getString("sclass"));

qrs.setClassroom(rs.getString("classroom"));

list.add(qrs);

}

}catch(Exception e){e.printStackTrace();}

return list;

}

/\*

\* 根据学生姓名查询选课情况

\* \*/

public List<QueryResult> selectSname(String name){

List<QueryResult> list = null;

try{

list = new ArrayList<QueryResult>();

String sql = "select student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +

"where student.sno=sc.sno and sc.cno=course.cno and student.sname=?;";

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, name);

ResultSet rs = pst.executeQuery();

while(rs.next()){

QueryResult qrs = new QueryResult();

qrs.setSNo(rs.getInt("sno"));

qrs.setCNo(rs.getInt("cno"));

qrs.setSname(rs.getString("sname"));

qrs.setCname(rs.getString("cname"));

qrs.setSclass(rs.getString("sclass"));

qrs.setClassroom(rs.getString("classroom"));

list.add(qrs);

}

}catch(Exception e){e.printStackTrace();}

return list;

}

/\*

\* 根据班级查询选课情况

\* \*/

public List<QueryResult> selectSclass(String \_class){

List<QueryResult> list = null;

try{

list = new ArrayList<QueryResult>();

String sql = "select student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +

"where student.sno=sc.sno and sc.cno=course.cno and student.sclass=?;";

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, \_class);

ResultSet rs = pst.executeQuery();

while(rs.next()){

QueryResult qrs = new QueryResult();

qrs.setSNo(rs.getInt("sno"));

qrs.setCNo(rs.getInt("cno"));

qrs.setSname(rs.getString("sname"));

qrs.setCname(rs.getString("cname"));

qrs.setSclass(rs.getString("sclass"));

qrs.setClassroom(rs.getString("classroom"));

list.add(qrs);

}

}catch(Exception e){e.printStackTrace();}

return list;

}

/\*

\* 根据课程号查询选课情况

\* \*/

public List<QueryResult> selectCno(int cno){

List<QueryResult> list = null;

try{

list = new ArrayList<QueryResult>();

String sql = "select student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +

"where student.sno=sc.sno and sc.cno=course.cno and course.cno=?;";

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, cno);

ResultSet rs = pst.executeQuery();

while(rs.next()){

QueryResult qrs = new QueryResult();

qrs.setSNo(rs.getInt("sno"));

qrs.setCNo(rs.getInt("cno"));

qrs.setSname(rs.getString("sname"));

qrs.setCname(rs.getString("cname"));

qrs.setSclass(rs.getString("sclass"));

qrs.setClassroom(rs.getString("classroom"));

list.add(qrs);

}

}catch(Exception e){e.printStackTrace();}

return list;

}

/\*

\* 根据课程名查询选课情况

\* \*/

public List<QueryResult> selectCname(String cname){

List<QueryResult> list = null;

try{

list = new ArrayList<QueryResult>();

String sql = "select student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +

"where student.sno=sc.sno and sc.cno=course.cno and course.cname=?;";

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, cname);

ResultSet rs = pst.executeQuery();

while(rs.next()){

QueryResult qrs = new QueryResult();

qrs.setSNo(rs.getInt("sno"));

qrs.setCNo(rs.getInt("cno"));

qrs.setSname(rs.getString("sname"));

qrs.setCname(rs.getString("cname"));

qrs.setSclass(rs.getString("sclass"));

qrs.setClassroom(rs.getString("classroom"));

list.add(qrs);

}

}catch(Exception e){e.printStackTrace();}

return list;

}

/\*

\* 根据上课教室查询选课情况

\* \*/

public List<QueryResult> selectRoom(String room){

List<QueryResult> list = null;

try{

list = new ArrayList<QueryResult>();

String sql = "select student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +

"where student.sno=sc.sno and sc.cno=course.cno and classroom=?;";

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, room);

ResultSet rs = pst.executeQuery();

while(rs.next()){

QueryResult qrs = new QueryResult();

qrs.setSNo(rs.getInt("sno"));

qrs.setCNo(rs.getInt("cno"));

qrs.setSname(rs.getString("sname"));

qrs.setCname(rs.getString("cname"));

qrs.setSclass(rs.getString("sclass"));

qrs.setClassroom(rs.getString("classroom"));

list.add(qrs);

}

}catch(Exception e){e.printStackTrace();}

return list;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 4.2学生模块设计

## 4.2.1学生选课功能

学生可以进行网上选课,在每学期快要结束时,学校都会把下学期要修的课程放到学校网站上,供同学们进行选课,这样节省了大家的宝贵时间,也方便了老师的管理,方便又实用.

此学生选课功能主要是方便学生操作,信息模块算法描述如下：

1 学生成功登陆；

2．点击选定课程按钮,进入选定课程的页面；

3．在页面中点击添加课程按钮；

4．点击想要选择的课程的复选框,进行选择；

5．然后点击提交按钮；

6．课程选择成功.

5.2.2查看学生陆；

2．点击查询学生信息；

## 4.2.2 输入学生的编信息功能

学生信息管理系统方便了学生查看自己的信息,如果学校由于疏忽输入了错误信息,自己可以查看并且及时通知学校教师,及时修改,减少不必要的麻烦.

此信息查询模块主要是方便学生查询操作,信息查询模块算法描述如下：

1 学生成功登号；

4．输入正确,进入信息查询页面进行相关修改.

## 4.2.3查看已选课程功能

学生可以查看本学期已选课程,并且可以方便的知道所选课程的教课老师和上课地点等,这样方便了学校专员,方便了老师和我们学生自己,需要的时候从网上下载课程表,随时看都可以.

此信息查询模块主要是方便学生查询操作,选课信息查询模块算法描述如下：

1. 学生成功登陆；

2．点击选定课程；

3．右边显示出本学期已选定的课程信息；

4．如果想继续选课,点击添加课程即可；

## 4.2.4删除已选课程功能

学生可以进行网上选课,就可以删除已选课程,比如说不小心选错了某一门课程,就必须在规定的时间里删除掉,然后重新选择要修的课程,因为有时粗心是不可避免的,所以有删除的功能是必须,这样既方便了老师的管理,又方便了自己.

1. 学生成功登陆；

2．点击选定课程；

3．右边显示出本学期已选定的课程信息,选择想要删除的课程；

4．单击课程后面的删除按钮即可；

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class DaoStu {

/\*\*

\* 学生登录

\* \*\*/

public Student loginStu(Student stu){

Student stu1 = null;

String sql\_loginS="select \* from student where SNo=? and Spassword=?;";

try{

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql\_loginS);

pst.setInt(1, stu.getSNo());

pst.setString(2,stu.getSpassword());

ResultSet rs = pst.executeQuery();

if(rs.next()){

stu1 = new Student(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),0);

System.out.println("student login select over");

}

}catch(Exception e){e.printStackTrace();}

return stu1;

}

/\*

\* 查询所有学生

\* \*/

public List<Student> selectStu(){

List<Student> list = null;

try{

String sql\_select = "select SNo,Sname,Spassword,Sclass,Ssex from student;";//查询全部学生信息+学分学分;//查询所有的sql语句

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql\_select);

ResultSet rs = pst.executeQuery();

list = new ArrayList<Student>();

DaoCou scredit = new DaoCou();

while(rs.next()){

Student stu = new Student(rs.getInt("SNo"),rs.getString("Sname"),rs.getString("Spassword"),rs.getString("Sclass")

,rs.getString("Ssex"), scredit.selectScre(rs.getInt("SNo")));

list.add(stu);

}

}catch(Exception e){e.printStackTrace();}

return list;

}

/\*

\* 查询学生总学分\_某位学生

\* \*/

public Student selectScre(Student stu){

ResultSet rs = null;

try{

String sql\_insert = "select student.sno as sno,sum(course.ccredit) as scredit from student,sc,course" +

" where student.sno=sc.sno" +

" and sc.cno=course.cno" +

" and student.sno=?";//查询某位同学学分

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql\_insert);

pst.setInt(1,stu.getSNo());

rs = pst.executeQuery();

if(rs.next()){

System.out.println("Student\_id"+stu.getSNo()+"select one over!");

stu = new Student(rs.getInt("sno"),null,null,null,null,rs.getFloat("scredit"));

}

}catch(Exception e){e.printStackTrace();}

return stu;

}

/\*

\* 查询某条个学生

\* \*/

public Student selectStu(int id){

Student Stu = null;

try{

String sql\_selectone = "select \* from student where SNo=?";//查询某一条课程

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql\_selectone);

pst.setInt(1, id);

ResultSet rs = pst.executeQuery();

if(rs.next()){

Stu = new Student(rs.getInt("SNo"),rs.getString("Sname"),rs.getString("Spassword"),rs.getString("Sclass")

,rs.getString("Ssex"),0);

}

}catch(Exception e){e.printStackTrace();}

return Stu;

}

/\*

\* 修改某个学生信息

\* \*/

public int updateStu(Student stu){

int rs = 0;

try{

String sql\_update = "update Student set Sname=?,Spassword=?,Sclass=?,Ssex=? where SNo=?;";//修改一条学生信息

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql\_update);

pst.setString(1, stu.getSname());

pst.setString(2, stu.getSpassword());

pst.setString(3, stu.getSclass());

pst.setString(4, stu.getSsex());

pst.setInt(5, stu.getSNo());

rs = pst.executeUpdate();

if(rs!=0){

System.out.println("Student\_id"+stu.getSNo()+"update over!");

}

}catch(Exception e){e.printStackTrace();}

return rs;

}

/\*

\* 删除某个学生信息

\* \*/

public int deleteStu(int id){

int rs = 0;

try{

String sql\_delete = "delete from student where SNo=?;";//删除一条课程信息

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql\_delete);

pst.setInt(1, id);

rs = pst.executeUpdate();

if(rs!=0){

System.out.println("Student\_id"+id+"delete over!");

}

}catch(Exception e){e.printStackTrace();}

return rs;

}

/\*

\* 添加一位学生信息

\* \*/

public int insertStu(Student stu){

int rs = 0;

try{

String sql\_insert = "insert into student(Sname,Spassword,Sclass,Ssex) values(?,?,?,?);";//添加一条课程信息

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql\_insert);

pst.setString(1,stu.getSname());

pst.setString(2, stu.getSpassword());

pst.setString(3, stu.getSclass());

pst.setString(4, stu.getSsex());

rs = pst.executeUpdate();

if(rs!=0){

System.out.println("Student\_id"+stu.getSNo()+"insert over!");

}

}catch(Exception e){e.printStackTrace();}

return rs;

}

}

在完成了上面的所有工作之后,本系统的各种功能已经全部确定,下面要完成的工作就是在详细设计的基础上,建立各功能界面,实现系统的各个功能.

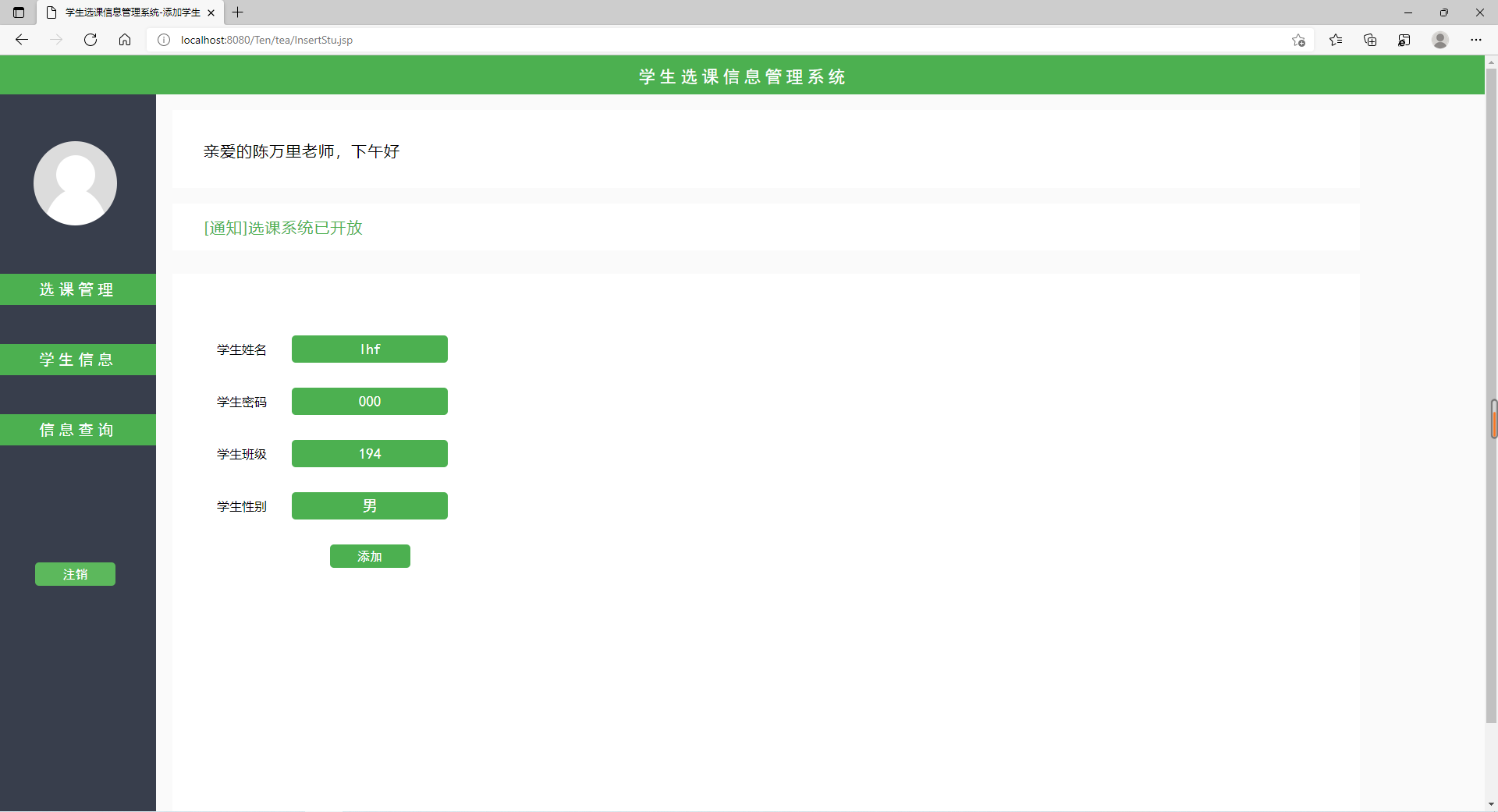
## 4.3系统登录界面

本系统有两个用户,分别是学生、教师/管理员,分配给他们不同的权限.



**图4.3-1用户登录界面**

## 4.3.1录入学生信息界面



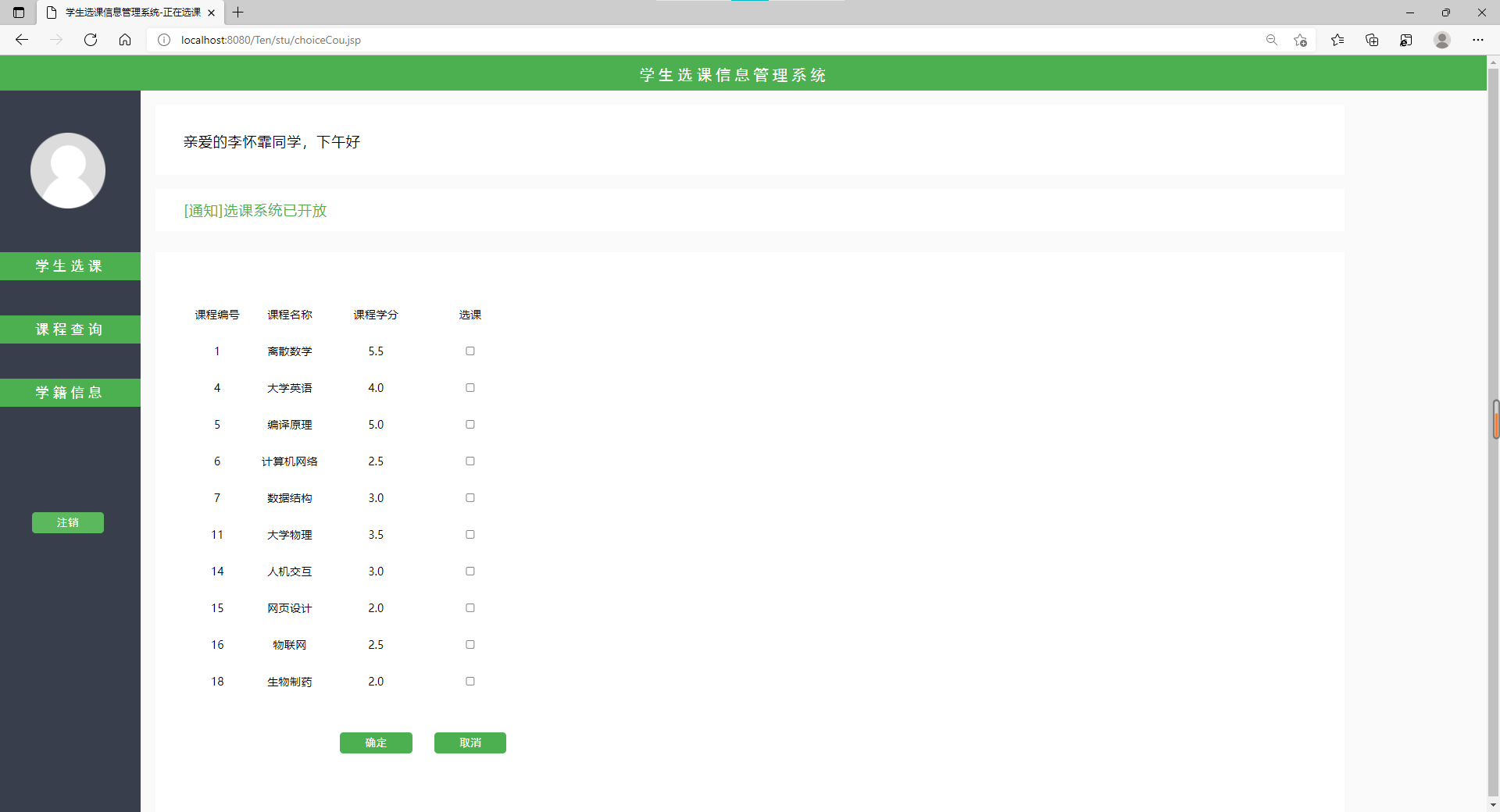
**图4.3-2录入学生信息界面**

## 4.3.2修改学生信息界面



**图4.3-3修改学生信息界面**

## 4.3.3学生选课界面



**图4.3-4学生选课界面**

# 5.系统测试

我们找来其他同学进行不指导使用，期间未发生错误。

# 6.结论

想要将系统做的完美无缺还是很困难的，由于在这个阶段中，基本是处于边学习边实践的状态，基础知识还不是很牢靠，加之自身水平原因，程序还是有许多不足之处：

界面模块做的不够好。

实现的功能不够完整。

界面需要更精确的美化。

经过这一段时间的努力，收获了很多，在收获的同时也发现了很多的不足，通过本次课程设计实训，我们熟悉掌握了web相关的编程技能，对Mysql数据库有了较深的理解。比较全面的巩固了编程思想。我们懂得的谨慎在编程工作中的重要性。程序员的工作态度以及良好的习惯也决定了自己工作的质量。