

**实 验（实训）报 告**

**项 目 名 称**  教务选课管理系统

**所属课程名称**  Web jsp开发

**项 目 类 型**  课程设计

**实验(实训)日期**  2019.6.12

**班 级**  17信息1班

**学 号**  170104100101

**姓 名**  蔡怡帅

**指导教师**  陈明晶

浙江财经大学教务处制

**目录**

[摘 要 3](#_Toc11330042)

[一、功能简介 3](#_Toc11330043)

[二、详细代码 4](#_Toc11330044)

[2.1登录界面 4](#_Toc11330045)

[2.2注册界面 11](#_Toc11330046)

[2.3学生界面 14](#_Toc11330047)

[2.3.1个人信息 14](#_Toc11330048)

[2.3.2成绩查看 19](#_Toc11330049)

[2.3.3自主选课 20](#_Toc11330050)

[2.3.4选课情况统计 24](#_Toc11330051)

[2.4教师界面 33](#_Toc11330052)

[2.4.1个人信息 33](#_Toc11330053)

[2.4.2我开的课 35](#_Toc11330054)

[2.4.3所有课程信息 37](#_Toc11330055)

[2.4.4学生选课情况统计 40](#_Toc11330056)

[2.5管理员界面 46](#_Toc11330057)

[2.5.1学生信息管理 47](#_Toc11330058)

[2.5.2学生成绩查看 49](#_Toc11330059)

[2.5.3课程信息管理 50](#_Toc11330060)

[2.5.4学生选课情况 55](#_Toc11330061)

[2.5.5学生选课情况统计 56](#_Toc11330062)

[2.5.6 教师信息查看 65](#_Toc11330063)

[三、软件所涉及到的数据表： 65](#_Toc11330064)

[四、Sql server中的代码（事务，视图，触发器，存储过程） 67](#_Toc11330065)

[4.1触发器 67](#_Toc11330066)

[4.2视图 69](#_Toc11330067)

[五、课程设计心得 70](#_Toc11330068)

# 摘 要

本文描述的是基于Windows环境的选课系统，主要功能模块包括：课程信息管理与学生选课管理。使用Eclipse进行web框架的开发，使用MySql、SQL server 2008建立数据库,最终实现选课系统的各个功能。

本程序提供了管理员、教师、学生三个用户，其中管理员可以对该系统中的学生信息、课程信息进行修改、删除以及查询。学生可以查询个人信息、选课信息、成绩。教师可以增加、修改、删除、查询自己所开设的课程。

# 一、功能简介

**登录界面：**实现对角色的判断功能，用户登录无需自己选择角色；实现了保存用户名与密码功能

**注册界面：**实现根据学号/工号来区分注册角色的身份

**学生界面：**实现了选课与退选的功能；个人信息的修改；查询成绩；查看课程详细信息；对学生所选课程统计，包括选课门数，选课学分总计，按学院统计选课结果并以可交互的饼图可视化展示结果；查看选课详情

**教师界面：**个人信息的查看；对自己所开课程的查看以及所有课程的查看；自己所开课程的选课学生的查看；以可交互的饼图可视化展示所开课程各门课选课人数

**管理员界面：**实现对学生信息的删除、修改功能；课程、学生成绩的查看；教师信息的查看；课程的增加、修改、删除；以可交互的柱状图可视化展示各学院课程所选人数；多角度统计学生选课情况，包括按课程统计，按学生统计，按教师统计，按学生学分统计；显示未选课的学生名单；

# 二、详细代码

## 2.1登录界面



//保存用户名和密码

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

String isSave = request.getParameter("remember");

String uname = request.getParameter("username");

//System.out.println(isSave);

**if**(isSave!=**null**) {

Cookie cookie = **new** Cookie("username", URLEncoder.*encode*(uname,"UTF-8"));

cookie.setMaxAge(60\*24\*30);

response.addCookie(cookie);

}

}

//注销

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

HttpSession session = request.getSession();

session.invalidate();

RequestDispatcher rd = request.getRequestDispatcher("index.jsp");

ServletContext application = request.getServletContext();

usercount--;

application.setAttribute("usernum", usercount);

request.setAttribute("status", "注销成功！");

rd.forward(request, response);

}

//登录

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

String uname = request.getParameter("username");//获取学号或工号

String upass = request.getParameter("userpass");//获取用户名

UserBean ub = **new** UserBean();//从数据库中获取是否存在该用户

RequestDispatcher rd;

Login user = **new** Login();

user = ub.login(uname, upass);

**if**(user.getUid()!=**null**){

System.***out***.println("登录成功！");

/\* 定义application \*/

ServletContext application = **this**.getServletContext();

Object num = application.getAttribute("usernum");

**if**(num==**null**){

usercount = 0;

}**else**{

usercount = (Integer)num;

}

usercount++;

application.setAttribute("usernum", usercount);

/\* 调用保存 cookie 的方法 \*/

saveUser(request, response);

/\* 定义session \*/

HttpSession session = request.getSession();

**if**(uname.equals("蔡军帅")){

rd = request.getRequestDispatcher("loginAdmin.jsp");

session.setAttribute("username", uname);

session.setAttribute("StuName", uname);

}

**else** **if**(uname.length()==4){

rd = request.getRequestDispatcher("loginOK.jsp");

session.setAttribute("username", uname);

String StuNname = ub.getStuName(uname);

//System.out.println("StuNname "+StuNname);

session.setAttribute("StuName", StuNname);

}

**else** **if**(uname.length()==3) {

rd = request.getRequestDispatcher("loginTea.jsp");

session.setAttribute("username", uname);

String TeaNname = ub.getTeaName(uname);

session.setAttribute("TeaName", TeaNname);

}

**else**{

System.***out***.println("登录失败！");

rd = request.getRequestDispatcher("failure.jsp");

}

}**else**{

System.***out***.println("登录失败！");

rd = request.getRequestDispatcher("failure.jsp");

}

System.***out***.println("准备跳转页面");

rd.forward(request, response);

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// **TODO** Auto-generated method stub

/\*

\* 不带状态的跳转：response.sendRedirext("index.jsp")

\* 带状态的页面跳转：RequestDispatcher.forward(request,response)

\*/

//判断登录还是登出

String act = request.getParameter("action");

**if**(act.equals("in")) {

login(request,response);

}**else** {

logout(request,response);

}

}

//登录用户类

**public** **class** Login {

**private** String uid;

**private** String ucode;

**public** String getUid() {

**return** uid;

}

**public** **void** setUid(String uid) {

**this**.uid = uid;

}

**public** String getUcode() {

**return** ucode;

}

**public** **void** setUcode(String ucode) {

**this**.ucode = ucode;

}

}

//用户类

**public** **class** UserBean {

**public** Login login(String un,String up) {

Login log = **new** Login();

String sql = "select uid,ucode from user where uid='"+un+"' and ucode='"+up+"'";

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

**if**(rs.next())

{

log.setUid(rs.getString(1));

log.setUcode(rs.getString(2));

}

}**catch**(Exception e) {

e.printStackTrace();

}

**return** log;

}

**public** String getStuName(String un) {

String StuName="";

String sql = "select Sname from student where Sno='"+un+"'";

System.***out***.println(sql);

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

**if**(rs.next())

{

StuName=rs.getString(1);

}

}**catch**(Exception e) {

e.printStackTrace();

}

**return** StuName;

}

**public** String getTeaName(String un) {

String TeaName="";

String sql = "select Tname from teacher where Tno='"+un+"'";

System.***out***.println(sql);

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

**if**(rs.next())

{

TeaName=rs.getString(1);

}

}**catch**(Exception e) {

e.printStackTrace();

}

**return** TeaName;

}

**public** **void** ins\_stu(String un,String uc ,String um,String ug,String ud,String ugr,String date,String up) {

String sql = "insert into student ";

sql += "values('"+un+"','"+um+"','"+ug+"','"+ud+"','"+ugr+"','"+date+"','"+up+"')";

System.***out***.println(sql);

DBBean db = **new** DBBean();

db.getConnection();

db.executeUpdata(sql);

sql = "insert into user values('"+un+"','"+uc+"')";

db.getConnection();

db.executeUpdata(sql);

}

**public** **void** ins\_tec(String un,String uc,String um,String ug,String ud) {

String sql = "insert into teacher ";

sql += "values('"+un+"','"+um+"','"+ug+"','"+ud+"')";

System.***out***.println(sql);

DBBean db = **new** DBBean();

db.getConnection();

db.executeUpdata(sql);

sql = "insert into user values('"+un+"','"+uc+"')";

db.getConnection();

db.executeUpdata(sql);

}

}

//数据库连接类

**public** **final** **class** DBBean {

**private** Connection conn;

**private** Statement stmt;

**private** ResultSet rs;

**public** DBBean() {

}

//建立数据库连接

**public** Connection getConnection() {

**try** {

/\*使用静态数据库连接池\*/

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

//建立连接池连接

conn = ds.getConnection();

}**catch**(Exception e) {

e.printStackTrace();

}

**return** conn;

}

// 提取SQL生成记录集

**public** ResultSet executeQuery(String sql) {

**try** {

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

System.***out***.println("返回结果集");

}**catch**(Exception e) {

e.printStackTrace();

}

**return** rs;

}

// 提取SQL添加记录

**public** **int** executeUpdata(String sql){

**int** result = 0;

**try**{

stmt = conn.createStatement();

result = stmt.executeUpdate(sql);

System.***out***.println("result "+result);

}**catch**(Exception e){

e.printStackTrace();

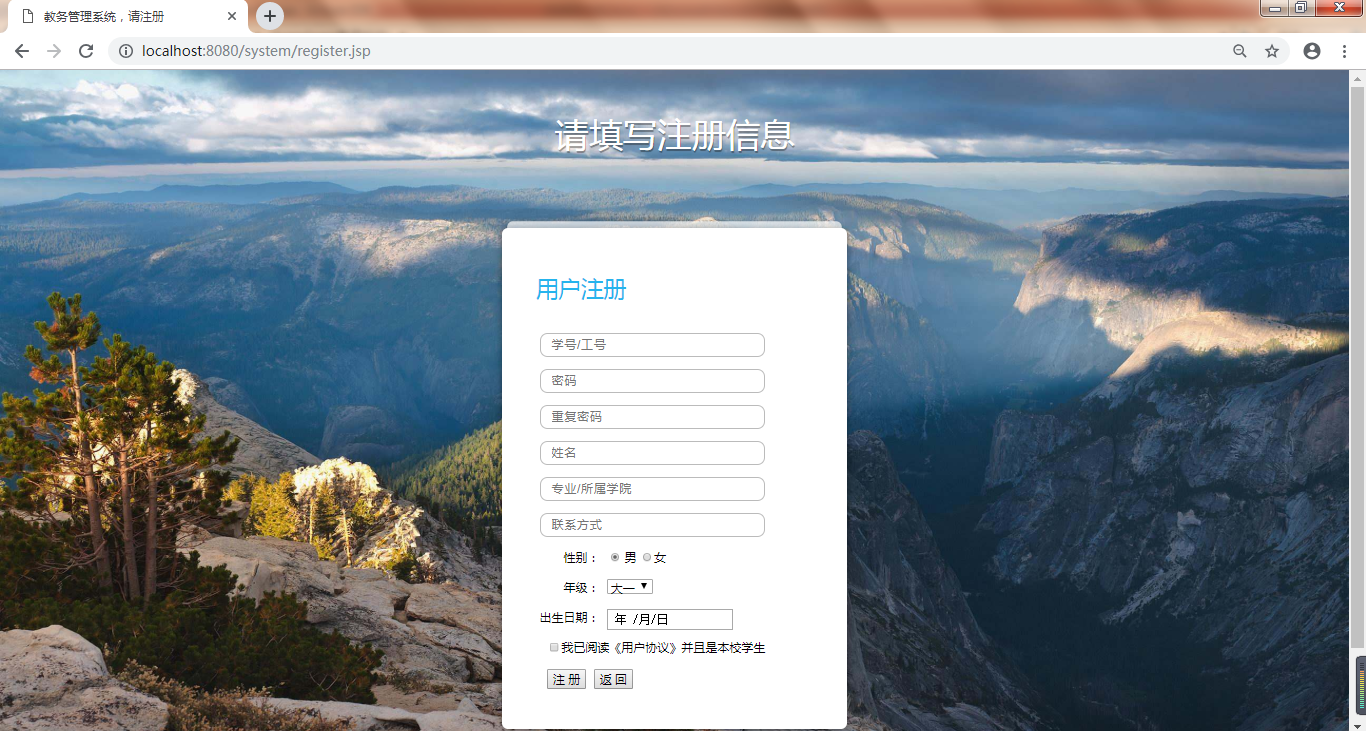
}

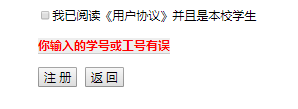
**return** result;

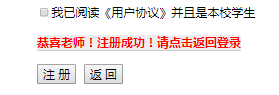
}

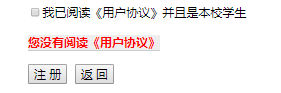
}

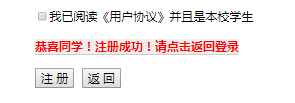
## 2.2注册界面











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

// **TODO** Auto-generated method stub

/\*

\* 获取所有数据

\*/

String un = request.getParameter("userNumber");

String um = request.getParameter("userName");

String uc = request.getParameter("userPass");

String cc = request.getParameter("confPass");

String ud = request.getParameter("userDept");

String up = request.getParameter("userPhone");

String ug = request.getParameter("gender");

String ub = request.getParameter("birthday");

String ugr = Integer.*toString*(Tools.*getNum*(request.getParameter("grade")));

String checkboxes[] = request.getParameterValues("read");

/\*

\* 处理

\*/

String msg;

**if**(checkboxes==**null**)

{

msg = "您没有阅读《用户协议》";

request.setAttribute("server\_info", msg);

RequestDispatcher rd = request.getRequestDispatcher("register.jsp");

rd.forward(request, response);

}

**if**(un==**null**||um==**null**||ud==**null**||up==**null**||ub.length()==0) {

msg = "请将信息填写完整";

request.setAttribute("server\_info", msg);

RequestDispatcher rd = request.getRequestDispatcher("register.jsp");

rd.forward(request, response);

}

**if**(!uc.equals(cc)) {

msg = "两次输入的密码不同";

request.setAttribute("server\_info", msg);

RequestDispatcher rd = request.getRequestDispatcher("register.jsp");

rd.forward(request, response);

}

//判断是un学生还是老师

//学生四位，第一位1-4

//老师一位，第一位是1

**int** f=1;

**if**(un.length()==4) {

**char** ch=un.charAt(0);

**if**(ch!='1' && ch!='2' && ch!='3'&& ch!='4') {

f=0;

}

}**else** **if**(un.length()==3) {

**char** ch=un.charAt(0);

**if**(ch!='1') {

f=0;

}**else** {

f=2;

}

}**else** {

f=0;

}

**if**(f==0) {

msg = "你输入的学号或工号有误";

request.setAttribute("server\_info", msg);

RequestDispatcher rd = request.getRequestDispatcher("register.jsp");

rd.forward(request, response);

}

/\*

\* 1是学生，2是老师，插入数据

\*/

UserBean UB = **new** UserBean();

**if**(f==1) {

UB.ins\_stu(un,uc,um,ug,ud,ugr,ub,up);

msg = "恭喜同学！注册成功！请点击返回登录";

request.setAttribute("server\_info", msg);

RequestDispatcher rd = request.getRequestDispatcher("register.jsp");

rd.forward(request, response);

}**else** **if**(f==2){

UB.ins\_tec(un,uc,um,ug,ud);

msg = "恭喜老师！注册成功！请点击返回登录";

request.setAttribute("server\_info", msg);

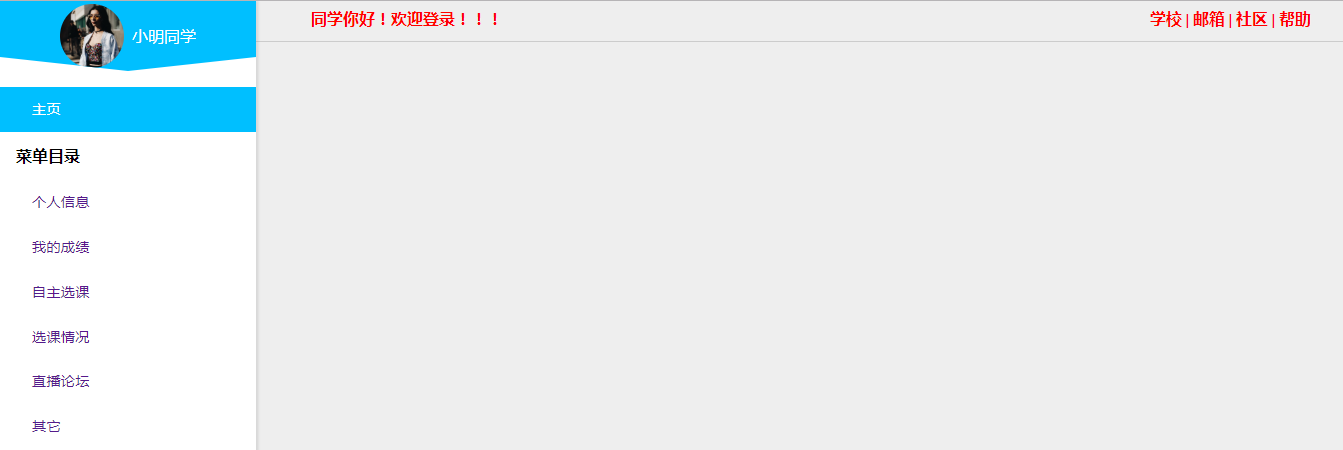
RequestDispatcher rd = request.getRequestDispatcher("register.jsp");

rd.forward(request, response);

}

}

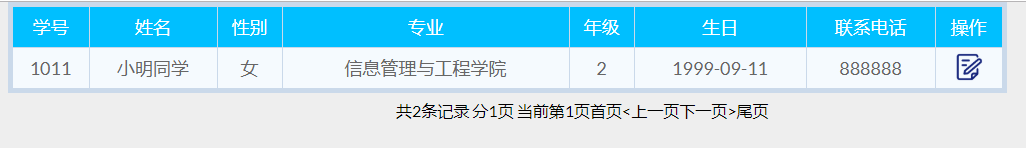
## 2.3学生界面





### 2.3.1个人信息

#### (1)信息查看



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//System.out.println("user "+user);

List studentlist = **new** ArrayList();

Student student;

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

**int** rowCount=0;

**if**(conn != **null**){

//out.print("数据库连接成功！");

Statement stmt = conn.createStatement();

//创建结果集合，集合与表的结构类似

String sql = "select \* from student where Sno=" + "'"+user+"'";

//System.out.println("sql "+sql);

ResultSet rs = stmt.executeQuery(sql);

rs.last();

rowCount=rs.getRow();

System.out.println("rowCount "+rowCount);

rs.first();

student = **new** Student(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7),"mima");

studentlist.add(student);

**while**(rs.next()){

//System.out.println("rs"+rs);

student = **new** Student(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7),"mima");

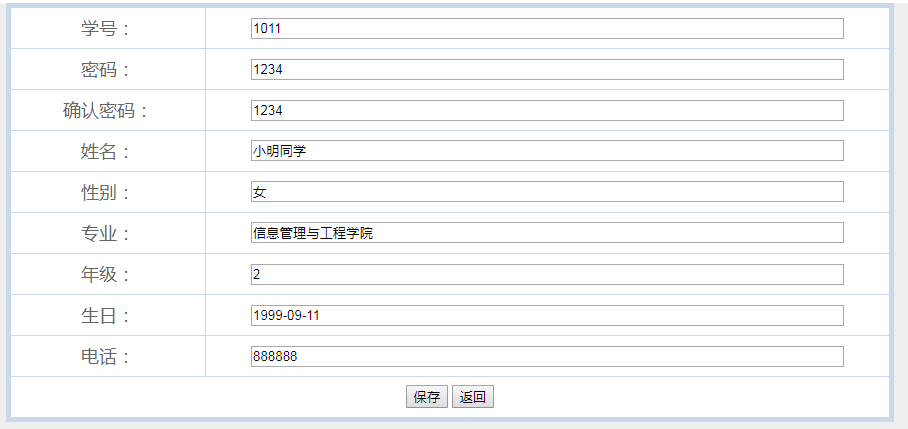
studentlist.add(student);

}

}

%>

#### (2)信息修改



//修改学生信息

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

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

System.***out***.println("nid " +nid);

StudentBean sb = **new** StudentBean();

Student stu = sb.getStuBySno(nid);

String code = sb.getCode(nid);

stu.setScode(code);

request.setAttribute("one\_student", stu);

RequestDispatcher rd;

rd = request.getRequestDispatcher("modStudent.jsp");

rd.forward(request, response);

}

//更新学生信息

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

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

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

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

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

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

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

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

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

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

StudentBean SB = **new** StudentBean();

**if**(co.equals(cco)) {

SB.upd(sno,sna,sg,sd,sgr,sb,sp,co);

}

**else** {

response.getWriter().print("<script>alert('对不起，你两次输入的密码不一致！')</script>");

}

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

System.***out***.println("admin "+admin+"!");

**if**(admin.equals("1")) {

response.sendRedirect("studentInformation.jsp");

}

**else**{

response.sendRedirect("myInformation.jsp");

}

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// **TODO** Auto-generated method stub

String act = request.getParameter("action");

**if**(act.equals("add")) {

addstu(request, response);

}**else** **if**(act.equals("mod")) {

modstu(request, response);

}**else** **if**(act.equals("del")) {

delstu(request, response);

}**else** **if**(act.equals("upd")) {

updstu(request, response);

}**else** **if**(act.equals("save")) {

insstu(request, response);

}

}

//学生信息与数据库连接类

**public** **class** StudentBean {

**public** Student getStuBySno(String nid) {

Student stu = **new** Student();

String sql = "select \* from student where Sno="+nid;

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

**if**(rs.next()) {

stu.setSno(rs.getString(1));

stu.setSname(rs.getString(2));

stu.setSgender(rs.getString(3));

stu.setSdept(rs.getString(4));

stu.setSgrade(rs.getString(5));

stu.setSbirth(rs.getString(6));

stu.setSphone(rs.getString(7));

}

}**catch**(Exception e) {

e.printStackTrace();

}

**return** stu;

}

**public** **void** del(String nid) {

String sql = "delete from student where Sno="+nid;

DBBean db = **new** DBBean();

db.getConnection();

db.executeUpdata(sql);

}

**public** String getCode(String nid) {

String sql = "select ucode from user where uid="+nid;

String code="";

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

**if**(rs.next()) {

code = rs.getString(1);

}

}**catch**(Exception e) {

e.printStackTrace();

}

**return** code;

}

**public** **void** upd(String sno,String sna,String sg,String sd,String sgr,String sb,String sp,String co) {

String sql = "update student set Sno='"+sno+"',Sname='"+sna+"',Sgender='"+sg;

sql += "',Sdept='"+sd+"',Sgrade='"+sgr+"',Sbirth='"+sb+"',Sphone='"+sp+"' where Sno="+"'"+sno+"'";

System.***out***.println("here: "+sql);

DBBean db = **new** DBBean();

db.getConnection();

db.executeUpdata(sql);

sql="update user set uid='"+sno+"',ucode='"+co+"' where uid="+"'"+sno+"'";

System.***out***.println("here: "+sql);

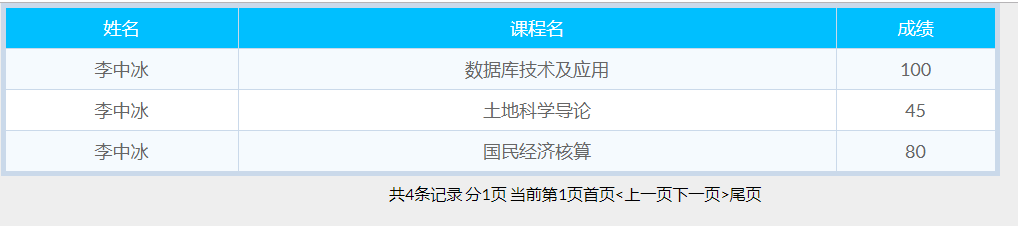
db.getConnection();

db.executeUpdata(sql);

}

}

### 2.3.2成绩查看



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

//Class c = new Class(driverName);

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

//建立数据库连接

Connection conn = DriverManager.getConnection(url);

//out.println(conn);//能输出就连接成功了

//建立语句执行环境

Statement stat = conn.createStatement();

//执行sql语句

String sql = "select Sname,Cname,Score from v\_grade where Sno="+"'"+user+"'";

System.out.println("sql "+sql);

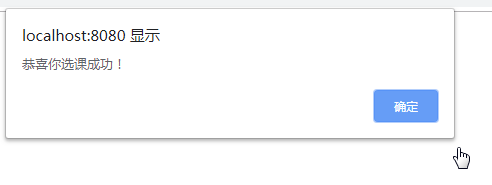
//弄个对象来接它，结果集

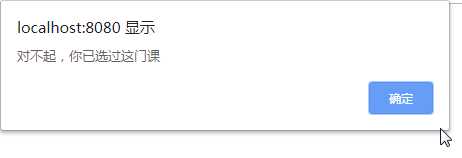
ResultSet rs = stat.executeQuery(sql);

%>

### 2.3.3自主选课







<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

List coursetlist = **new** ArrayList();

Course course;

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

**int** rowCount=0;

**if**(conn != **null**){

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from v\_choose\_course");

rs.last();

rowCount=rs.getRow();

rs.first();

course = **new** Course(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7));

coursetlist.add(course);

**while**(rs.next()){

course = **new** Course(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7));

coursetlist.add(course);

}

rs.last();

rowCount=rs.getRow();

rs.close();

stmt.close();

conn.close();

}

%>

<%

**if**((request.getAttribute("is\_success"))!=**null**)

{

String ok =(String)request.getAttribute("is\_success");

**if**(ok.equals("1")){

%>

<script>

alert('恭喜你选课成功！');

</script>

<%

}**else**{

%>

<script>

alert('对不起，你已选过这门课');

</script>

<%

}

}

**int** who=0;

**if**(user.equals("蔡军帅")){

who=1;

}**else** **if**(user.length()==3){

who=2;

}

%>

//学生选课

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

// **TODO** Auto-generated method stub

String act = request.getParameter("action");

**if**(act.equals("ins")) {

choose(request, response);

}**else** **if**(act.equals("del")) {

delete(request, response);

}**else** **if**(act.equals("find")) {

find(request,response);

}

}

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

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

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

SelectedBean Sb = **new** SelectedBean();

String ok = Sb.add(Sno,Cno);

RequestDispatcher rd = request.getRequestDispatcher("chooseCourse.jsp");

request.setAttribute("is\_success", ok);

rd.forward(request, response);

}

//选课与数据库连接类

**public** **class** SelectedBean {

**public** String add(String Sno, String Cno) {

String ok="1";

String sql = "insert into selection(Sno,Cno) values("+"'"+Sno+"','"+Cno+"')";

System.***out***.println("here: "+sql);

DBBean db = **new** DBBean();

db.getConnection();

//以前没选过

**if**(search(Sno,Cno)==1) {

//那么选

db.executeUpdata(sql);

}**else** {

ok="0";

}

//ok=1表示选课成功

**return** ok;

}

**public** **int** search(String Sno,String Cno) {

**int** ok=1;

String sql = "select \* from selection where Sno="+"'"+Sno+"' and Cno='"+Cno+"'";

System.***out***.println("here2: "+sql);

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

System.***out***.println("rs "+rs);

**try** {

**if** (rs.next()) {

//已经存在的了，就不ok

ok=0;

}

}**catch**(Exception e){

e.printStackTrace();

}

System.***out***.println("ok "+ok);

**return** ok;

}

**public** **void** del(String Sno,String Cno) {

String sql = "delete from selection where Sno="+"'"+Sno+"' and Cno='"+Cno+"'";

System.***out***.println("here2: "+sql);

DBBean db = **new** DBBean();

db.getConnection();

db.executeUpdata(sql);

}

}

### 2.3.4选课情况统计



#### 选课门数及学分



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

Statement stmt = conn.createStatement();

//执行sql语句

String sql = "select SUM(course.Credit) ";

sql+="from v\_selected\_stu\_tea\_cou,course ";

sql+="where v\_selected\_stu\_tea\_cou.Cno = course.Cno ";

sql+="and Sno='"+user+"'";

System.out.println("sql "+sql);

ResultSet rs = stmt.executeQuery(sql);

**int** fen=0;

**while**(rs.next()){

fen=rs.getInt(1);

}

initContext = **new** InitialContext();

envContext = (Context)initContext.lookup("java:/comp/env");

ds = (DataSource)envContext.lookup("jdbc/system");

conn = ds.getConnection();

//out.print("数据库连接成功！");

stmt = conn.createStatement();

**int** men=0;

sql = "select count(\*) ";

sql+="from v\_selected\_stu\_tea\_cou ";

sql+="where Sno='"+user+"'";

System.out.println("sql "+sql);

//弄个对象来接它，结果集

ResultSet rs2 = stmt.executeQuery(sql);

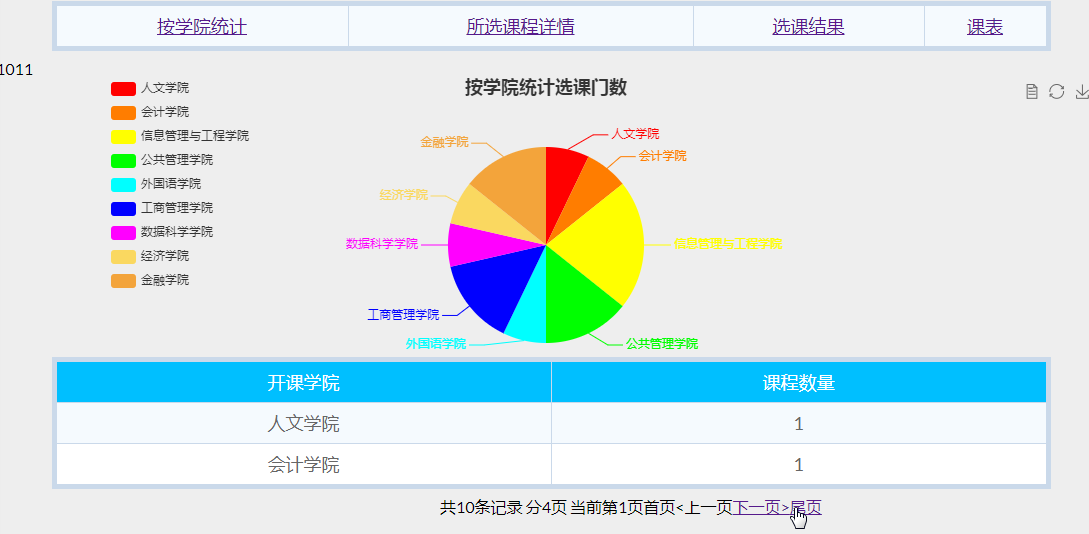
**while**(rs2.next()){

men=rs2.getInt(1);

}

%>

#### 按学院统计



**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

System.***out***.println("hhh");

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

System.***out***.println("useruser "+user);

String sql="Select College,Count(\*) from course where cno in ";

sql+="(SELECT Cno from v\_choose\_course where Cno in ";

sql+="(SELECT Cno from v\_selected\_admin where Sno='"+user+"')) group by College";

List<Product> list = **new** ArrayList<Product>();

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

**while**(rs.next())

{

list.add(**new** Product(rs.getString(1), rs.getInt(2)));

System.***out***.println(rs.getString(1));

}

}**catch**(Exception e) {

e.printStackTrace();

}

//System.out.println("这里是doPost");

Gson gson2 = **new** Gson();

String json = gson2.toJson(list);

System.***out***.println(json);

// 将json字符串数据返回给前端

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

response.getWriter().write(json);

}

<script type=*"text/javascript"*>

**var** myChart = echarts.init(document.getElementById('main'));

**var** user = document.getElementById('xuehao').innerHTML;

//alert(user);

//3.初始化，默认显示标题，图例和xy空坐标轴

myChart.setOption({

title : {

text: '按学院统计选课门数',

//subtext: '纯属虚构',

x:'center'

},

tooltip : {

trigger: 'item',

formatter: "{a} <br/>{b} : {c} ({d}%)"

},

legend: {

orient : 'vertical',

x : 'left',

data:[]

},

toolbox: {

show : **true**,

feature : {

mark : {show: **true**},

dataView : {show: **true**, readOnly: **false**},

magicType : {

show: **true**,

type: ['pie', 'funnel'],

option: {

funnel: {

x: '25%',

width: '50%',

funnelAlign: 'left',

max: 1548

}

}

},

restore : {show: **true**},

saveAsImage : {show: **true**}

}

},

calculable : **true**,

series : [

{

name:'选课门数',

type:'pie',

radius : '65%',

center: ['50%', '60%'],

data:[]

}

]

});

//4.设置加载动画(非必须)

myChart.showLoading(); //数据加载完之前先显示一段简单的loading动画

//5.定义数据存放数组(动态变)

**var** statisticsData = []; //这是我自己建的空数组，为了把异步拿到的数据push进去

**var** statisticsName = [];

//6.ajax发起数据请求

$.ajax({

type : "post",

async : **true**, //异步请求（同步请求将会锁住浏览器，其他操作须等请求完成才可执行）

url : "stuCouEcharts", //请求发送到TestServlet

data : {'user':user},

dataType : "json", //返回数据形式为json

//7.请求成功后接收数据name+num两组数据

success : **function**(result) {

//result为服务器返回的json对象

**if** (result) {

//8.取出数据存入数组

**for** (**var** i = 0; i < result.length; i++) {

**var** statisticsObj = {name:'',value:''}; //因为ECharts里边需要的的数据格式是这样的

statisticsObj.name = result[i].name;

statisticsName.push(result[i].name);

statisticsObj.value = result[i].num;

statisticsData.push(statisticsObj); //把拿到的异步数据push进我自己建的数组里

}

myChart.hideLoading(); //隐藏加载动画

//9.覆盖操作-根据数据加载数据图表

myChart.setOption({

legend: {

show: **true**,

x: '10%',

data: statisticsName //这里是图表上的数据

},

series: [{

name: '选课门数',

type: 'pie',

radius : '70%',

center: ['50%', '60%'],

data: statisticsData, //这里是异步加载系列列表

itemStyle: {

normal: {

color: **function**(params) {

// build a color map as your need.

**var** colorList = [

'#FF0000','#FF7D00','#FFFF00','#00FF00','#00FFFF',

'#0000FF','#FF00FF','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

**return** colorList[params.dataIndex];

},

}

},

}]

});

}

},

error : **function**(errorMsg) {

//请求失败时执行该函数

alert("图表请求数据失败!");

myChart.hideLoading();

}

})

</script>

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

System.***out***.println("hhh");

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

System.***out***.println("useruser "+user);

String sql="Select College,Count(\*) from course where cno in ";

sql+="(SELECT Cno from v\_choose\_course where Cno in ";

sql+="(SELECT Cno from v\_selected\_admin where Sno='"+user+"')) group by College";

List<Product> list = **new** ArrayList<Product>();

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

**while**(rs.next())

{

list.add(**new** Product(rs.getString(1), rs.getInt(2)));

System.***out***.println(rs.getString(1));

}

}**catch**(Exception e) {

e.printStackTrace();

}

//System.out.println("这里是doPost");

Gson gson2 = **new** Gson();

String json = gson2.toJson(list);

System.***out***.println(json);

// 将json字符串数据返回给前端

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

response.getWriter().write(json);

}

#### 所选课程详情



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

List coursetlist = **new** ArrayList();

Course course;

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

//out.print("数据库连接成功！");

Statement stmt = conn.createStatement();

//创建结果集合，集合与表的结构类似

String sql="SELECT \* from v\_choose\_course where Cno in (SELECT Cno from v\_selected\_admin where Sno="+"'"+user+"'"+")";

ResultSet rs = stmt.executeQuery(sql);

%>

#### 选课结果



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

List coursetlist = **new** ArrayList();

Course course;

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

//out.print("数据库连接成功！");

Statement stmt = conn.createStatement();

//创建结果集合，集合与表的结构类似

ResultSet rs = stmt.executeQuery("select \* from v\_selected\_admin where Sno="+"'"+user+"'");

%>

//退选课程

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

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

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

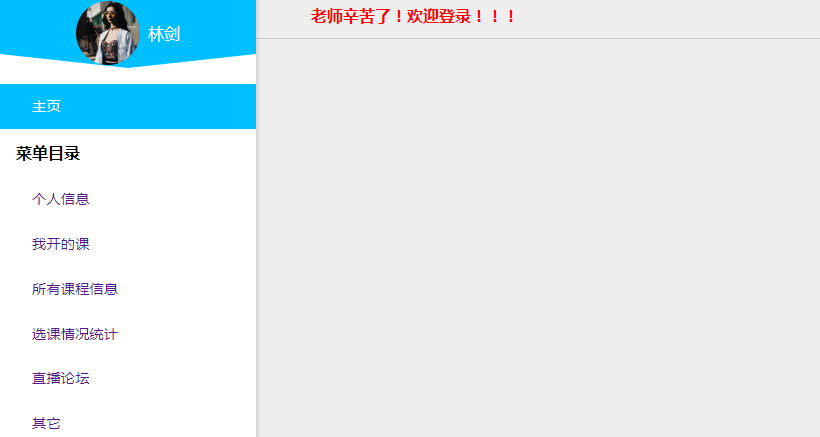
SelectedBean Sb = **new** SelectedBean();

Sb.del(Sno,Cno);

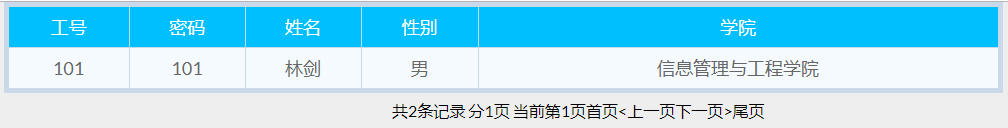
response.sendRedirect("mySelected.jsp");

}

## 2.4教师界面



### 2.4.1个人信息



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//System.out.println("user "+user);

List TeacherList = **new** ArrayList();

Teacher teacher;

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

**int** rowCount=0;

**if**(conn != **null**){

Statement stmt = conn.createStatement();

//创建结果集合，集合与表的结构类似

String sql = "select \* from teacher where Tno=" + "'"+user+"'";

ResultSet rs = stmt.executeQuery(sql);

rs.last();

rowCount=rs.getRow();

System.out.println("rowCount "+rowCount);

rs.first();

teacher = **new** Teacher(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),"mima");

TeacherList.add(teacher);

**while**(rs.next()){

//System.out.println("rs"+rs);

teacher = **new** Teacher(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),"mima");

TeacherList.add(teacher);

}

}

%>

//教师类

**public** **class** Teacher {

**private** String Tno;

**private** String Tname;

**private** String Tg;

**private** String Tc;

**private** String Tcode;

**public** String getTno() {

**return** Tno;

}

**public** **void** setTno(String tno) {

Tno = tno;

}

**public** String getTname() {

**return** Tname;

}

**public** **void** setTname(String tname) {

Tname = tname;

}

**public** String getTg() {

**return** Tg;

}

**public** **void** setTg(String tg) {

Tg = tg;

}

**public** String getTc() {

**return** Tc;

}

**public** **void** setTc(String tc) {

Tc = tc;

}

**public** String getTcode() {

**return** Tcode;

}

**public** **void** setScode(String scode) {

Tcode = scode;

}

**public** Teacher(String a, String b, String c, String d,String e) {

**super**();

**this**.Tno = a;

**this**.Tname = b;

**this**.Tg = c;

**this**.Tc = d;

**this**.Tcode = e;

}

**public** Teacher() {

}

}

### 2.4.2我开的课



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

//Class c = new Class(driverName);

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

//建立数据库连接

Connection conn = DriverManager.getConnection(url);

//out.println(conn);//能输出就连接成功了

//建立语句执行环境

Statement stat = conn.createStatement();

//执行sql语句

String sql = "select \* from course where Tno="+"'"+user+"'";

System.out.println("sql "+sql);

ResultSet rs = stat.executeQuery(sql);

List coursetlist = **new** ArrayList();

Course course;

rs.last();

**int** rowCount=0;

rowCount=rs.getRow();

rs.first();

course = **new** Course(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7));

coursetlist.add(course);

**while**(rs.next()){

course = **new** Course(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7));

coursetlist.add(course);

}

rs.last();

rowCount=rs.getRow();

rs.close();

stat.close();

conn.close();

%>

### 2.4.3所有课程信息



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

List coursetlist = **new** ArrayList();

Course course;

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

**int** rowCount=0;

**if**(conn != **null**){

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from v\_choose\_course");

rs.last();

rowCount=rs.getRow();

rs.first();

course = **new** Course(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7));

coursetlist.add(course);

**while**(rs.next()){

course = **new** Course(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7));

coursetlist.add(course);

}

rs.last();

rowCount=rs.getRow();

rs.close();

stmt.close();

conn.close();

}

**int** who=0;

**if**(user.equals("蔡军帅")){

who=1;

}**else** **if**(user.length()==3){

who=2;

}

%>

//课程类

**public** **class** Course {

**private** String Cno;

**private** String Cname;

**private** String Tno;

**private** String Time;

**private** String TestTime;

**private** String Credit;

**private** String College;

**public** String getCno() {

**return** Cno;

}

**public** **void** setCno(String cno) {

Cno = cno;

}

**public** String getCname() {

**return** Cname;

}

**public** **void** setCname(String cname) {

Cname = cname;

}

**public** String getTno() {

**return** Tno;

}

**public** **void** setTno(String tno) {

Tno = tno;

}

**public** String getTime() {

**return** Time;

}

**public** **void** setTime(String time) {

Time = time;

}

**public** String getTestTime() {

**return** TestTime;

}

**public** **void** setTestTime(String testTime) {

TestTime = testTime;

}

**public** String getCredit() {

**return** Credit;

}

**public** **void** setCredit(String credit) {

Credit = credit;

}

**public** String getCollege() {

**return** College;

}

**public** **void** setCollege(String college) {

College = college;

}

**public** Course(String Cno, String Cname, String Tno, String Time,String TestTime, String Credit, String College) {

**super**();

**this**.Cno = Cno;

**this**.Cname = Cname;

**this**.Tno = Tno;

**this**.Time = Time;

**this**.TestTime = TestTime;

**this**.Credit = Credit;

**this**.College = College;

}

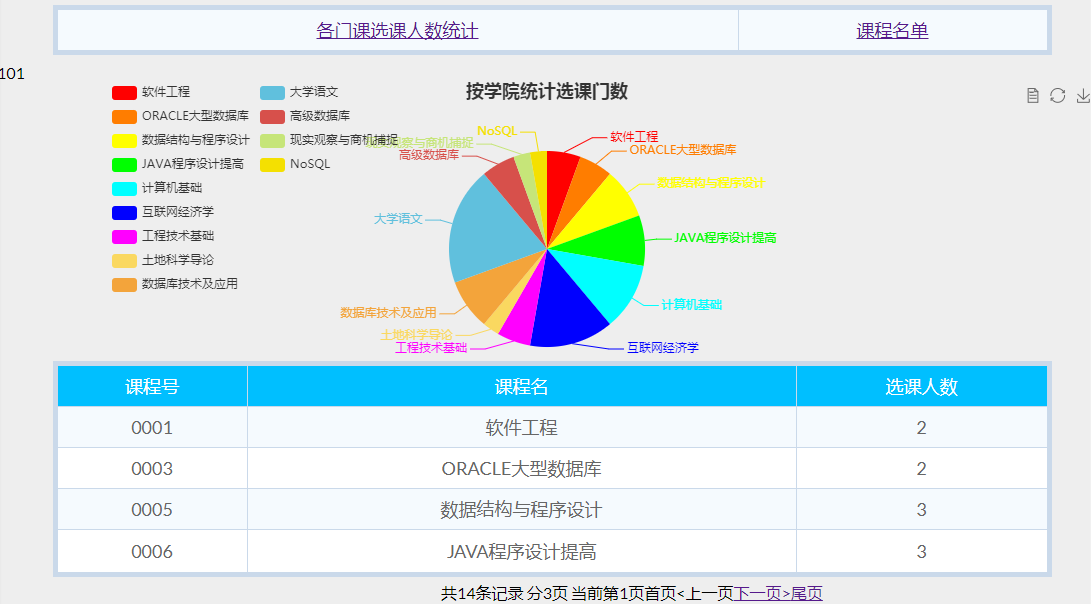
**public** Course() {

}

}

### 2.4.4学生选课情况统计

#### (1)所开课程统计学生人数



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

//Class c = new Class(driverName);

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

Connection conn = DriverManager.getConnection(url);

Statement stat = conn.createStatement();

String sql = "select Cno,Cname,COUNT(\*) from v\_selected\_stu\_tea\_cou where Tno='"+user+"' group by Cno";

System.out.println("界面sql "+sql);

ResultSet rs = stat.executeQuery(sql);

%>

<script type=*"text/javascript"*>

**var** myChart = echarts.init(document.getElementById('main'));

**var** user = document.getElementById('xuehao').innerHTML;

//alert(user);

//3.初始化，默认显示标题，图例和xy空坐标轴

myChart.setOption({

title : {

text: '按学院统计选课门数',

//subtext: '纯属虚构',

x:'center'

},

tooltip : {

trigger: 'item',

formatter: "{a} <br/>{b} : {c} ({d}%)"

},

legend: {

orient : 'vertical',

x : 'left',

data:[]

},

toolbox: {

show : **true**,

feature : {

mark : {show: **true**},

dataView : {show: **true**, readOnly: **false**},

magicType : {

show: **true**,

type: ['pie', 'funnel'],

option: {

funnel: {

x: '25%',

width: '50%',

funnelAlign: 'left',

max: 1548

}

}

},

restore : {show: **true**},

saveAsImage : {show: **true**}

}

},

calculable : **true**,

series : [

{

name:'选课门数',

type:'pie',

radius : '65%',

center: ['50%', '60%'],

data:[]

}

]

});

//4.设置加载动画(非必须)

myChart.showLoading(); //数据加载完之前先显示一段简单的loading动画

//5.定义数据存放数组(动态变)

**var** statisticsData = []; //这是我自己建的空数组，为了把异步拿到的数据push进去

**var** statisticsName = [];

//6.ajax发起数据请求

$.ajax({

type : "post",

async : **true**, //异步请求（同步请求将会锁住浏览器，其他操作须等请求完成才可执行）

url : "teaCouEcharts", //请求发送到TestServlet

data : {'user':user},

dataType : "json", //返回数据形式为json

//7.请求成功后接收数据name+num两组数据

success : **function**(result) {

//result为服务器返回的json对象

**if** (result) {

//8.取出数据存入数组

**for** (**var** i = 0; i < result.length; i++) {

**var** statisticsObj = {name:'',value:''}; //因为ECharts里边需要的的数据格式是这样的

statisticsObj.name = result[i].name;

statisticsName.push(result[i].name);

statisticsObj.value = result[i].num;

statisticsData.push(statisticsObj); //把拿到的异步数据push进我自己建的数组里

}

myChart.hideLoading(); //隐藏加载动画

//9.覆盖操作-根据数据加载数据图表

myChart.setOption({

legend: {

show: **true**,

x: '10%',

data: statisticsName //这里是图表上的数据

},

series: [{

name: '选课门数',

type: 'pie',

radius : '70%',

center: ['50%', '60%'],

data: statisticsData, //这里是异步加载系列列表

itemStyle: {

normal: {

color: **function**(params) {

// build a color map as your need.

**var** colorList = [

'#FF0000','#FF7D00','#FFFF00','#00FF00','#00FFFF',

'#0000FF','#FF00FF','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

**return** colorList[params.dataIndex];

},

}

},

}]

});

}

},

error : **function**(errorMsg) {

//请求失败时执行该函数

alert("图表请求数据失败!");

myChart.hideLoading();

}

})

</script>

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

// **TODO** Auto-generated method stub

System.***out***.println("hhh");

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

System.***out***.println("useruser "+user);

String sql="select Cname,COUNT(\*) from v\_selected\_stu\_tea\_cou where Tno='"+user+"' group by Cno";

List<Product> list = **new** ArrayList<Product>();

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

System.***out***.println("这里是teachCouE");

**while**(rs.next())

{

list.add(**new** Product(rs.getString(1), rs.getInt(2)));

System.***out***.println(rs.getString(1));

}

}**catch**(Exception e) {

e.printStackTrace();

}

Gson gson2 = **new** Gson();

String json = gson2.toJson(list);

System.***out***.println(json);

// 将json字符串数据返回给前端

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

response.getWriter().write(json);

}

#### (2)所开课程的课程名单



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

Connection conn = DriverManager.getConnection(url);

Statement stat = conn.createStatement();

String sql = "select Sno,Sname,Cno,Cname from v\_selected\_stu\_tea\_cou where Tno='"+user+"' order by Cno desc";

System.out.println("sql "+sql);

ResultSet rs = stat.executeQuery(sql);

%>

## 2.5管理员界面



### 2.5.1学生信息管理

#### (1)信息查看



<%

List studentlist = **new** ArrayList();

Student student;

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

**int** rowCount=0;

**if**(conn != **null**){

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from student");

rs.last();

rowCount=rs.getRow();

rs.first();

student = **new** Student(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7),"mima");

studentlist.add(student);

**while**(rs.next()){

student = **new** Student(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7),"mima");

studentlist.add(student);

}

}

%>

#### (2)信息修改



//更新学生信息

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

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

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

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

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

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

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

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

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

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

StudentBean SB = **new** StudentBean();

**if**(co.equals(cco)) {

SB.upd(sno,sna,sg,sd,sgr,sb,sp,co);

//System.out.println("已更新！！！");

}

**else** {

response.getWriter().print("<script>alert('对不起，你两次输入的密码不一致！')</script>");

}

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

System.***out***.println("admin "+admin+"!");

**if**(admin.equals("1")) {

response.sendRedirect("studentInformation.jsp");

}

**else**{

response.sendRedirect("myInformation.jsp");

}

}

#### (3)信息删除

//删除学生信息

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

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

StudentBean sb = **new** StudentBean();

sb.del(nid);

response.sendRedirect("studentInformation.jsp");

}

### 2.5.2学生成绩查看



<%

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

Connection conn = DriverManager.getConnection(url);

Statement stat = conn.createStatement();

String sql = "select Sname,Cname,Score from v\_grade ";

ResultSet rs = stat.executeQuery(sql);

%>

### 2.5.3课程信息管理

#### (1)信息查看



<%

String user =(String)session.getAttribute("username");

session.setAttribute("usernum", user);

List coursetlist = **new** ArrayList();

Course course;

Context initContext = **new** InitialContext();

Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds = (DataSource)envContext.lookup("jdbc/system");

Connection conn = ds.getConnection();

**int** rowCount=0;

**if**(conn != **null**){

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from course");

rs.last();

rowCount=rs.getRow();

rs.first();

course = **new** Course(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7));

coursetlist.add(course);

**while**(rs.next()){

//System.out.println("rs"+rs);

course = **new** Course(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7));

coursetlist.add(course);

}

rs.last();

rowCount=rs.getRow();

rs.close();

stmt.close();

conn.close();

}

**int** who=0;

**if**(user.equals("蔡军帅")){

who=1;

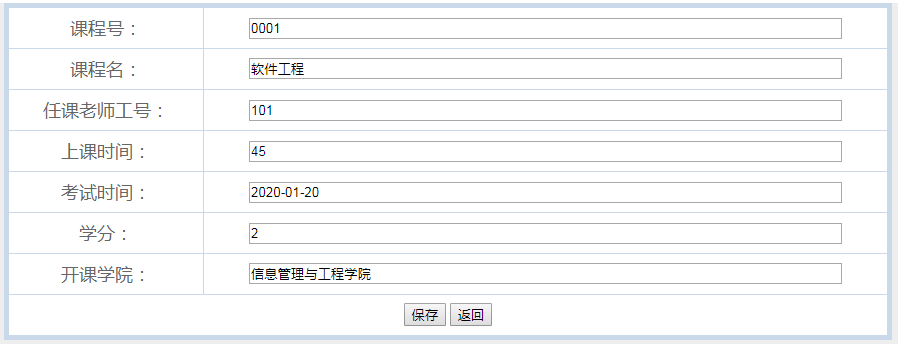
}**else** **if**(user.length()==3){

who=2;

}

%>

#### (2)信息修改



//修改课程

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

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

System.***out***.println("nid " +nid);

CourseBean sb = **new** CourseBean();

Course cou = sb.getCourseByCno(nid);

request.setAttribute("one\_course", cou);

RequestDispatcher rd;

rd = request.getRequestDispatcher("modCourse.jsp");

rd.forward(request, response);

}

//更新课程

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

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

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

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

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

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

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

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

CourseBean cb = **new** CourseBean();

cb.upd(Cno,Cname,Tno,Time,TestTime,Credit,College);

response.sendRedirect("CourseAdmin.jsp");

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// **TODO** Auto-generated method stub

String act = request.getParameter("action");

**if**(act.equals("add")) {

addcou(request, response);

}**else** **if**(act.equals("mod")) {

modcou(request, response);

}**else** **if**(act.equals("del")) {

delcou(request, response);

}**else** **if**(act.equals("upd")) {

updcou(request, response);

}**else** **if**(act.equals("save")) {

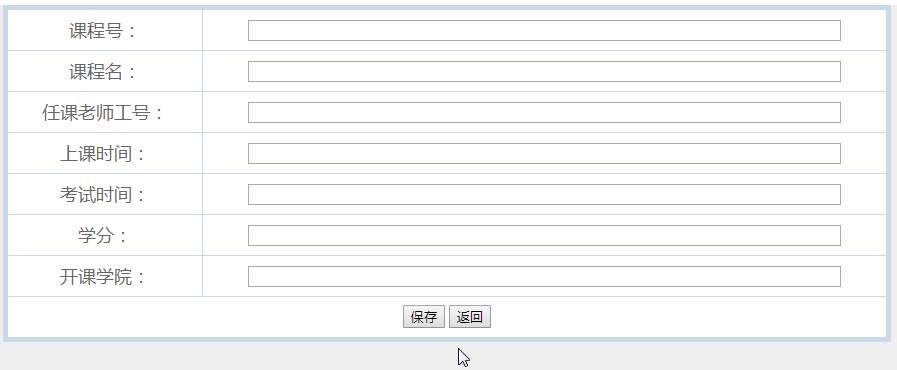
inscou(request, response);

}

}

#### (3)信息增加





//增加课程

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

RequestDispatcher rd;

rd = request.getRequestDispatcher("addCourse.jsp");

rd.forward(request, response);

}

//插入课程

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

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

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

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

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

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

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

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

CourseBean cb = **new** CourseBean();

cb.add(Cno,Cname,Tno,Time,TestTime,Credit,College);

response.sendRedirect("CourseAdmin.jsp");

}

#### (4)信息删除

//删除课程

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

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

CourseBean cb = **new** CourseBean();

cb.del(nid);

response.sendRedirect("CourseAdmin.jsp");

}

### 2.5.4学生选课情况



<%

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

Connection conn = DriverManager.getConnection(url);

Statement stat = conn.createStatement();

String sql = "select \* from v\_selected\_admin ";

ResultSet rs = stat.executeQuery(sql);

%>

### 2.5.5学生选课情况统计

#### (1)按学院统计



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

//Class c = new Class(driverName);

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

//建立数据库连接

Connection conn = DriverManager.getConnection(url);

//out.println(conn);//能输出就连接成功了

//建立语句执行环境

Statement stat = conn.createStatement();

//执行sql语句

String sql = "select \* from v\_count\_college";

System.out.println("sql "+sql);

//弄个对象来接它，结果集

ResultSet rs = stat.executeQuery(sql);

%>

<script type=*"text/javascript"*>

**var** myChart = echarts.init(document.getElementById('main'));

//3.初始化，默认显示标题，图例和xy空坐标轴

myChart.setOption({

title : {

text : '按学院统计选课人数',

x:'center',

y:'top'

},

tooltip : {},

legend : {

data : [ '人数' ],

left:'10%',

},

xAxis : {

data : []

},

yAxis : {},

series : [ {

name : '人数',

type : 'bar',

data : []

} ]

});

//4.设置加载动画(非必须)

myChart.showLoading(); //数据加载完之前先显示一段简单的loading动画

//5.定义数据存放数组(动态变)

**var** names = []; //建立一个类别数组（实际用来盛放X轴坐标值）

**var** nums = []; //建立一个销量数组（实际用来盛放Y坐标值）

//6.ajax发起数据请求

$.ajax({

type : "post",

async : **true**, //异步请求（同步请求将会锁住浏览器，其他操作须等请求完成才可执行）

url : "TestServlet", //请求发送到TestServlet

data : {},

dataType : "json", //返回数据形式为json

//7.请求成功后接收数据name+num两组数据

success : **function**(result) {

//result为服务器返回的json对象

**if** (result) {

//8.取出数据存入数组

**for** (**var** i = 0; i < result.length; i++) {

names.push(result[i].name); //迭代取出类别数据并填入类别数组

}

**for** (**var** i = 0; i < result.length; i++) {

nums.push(result[i].num); //迭代取出数量并填入销量数组

}

myChart.hideLoading(); //隐藏加载动画

//9.覆盖操作-根据数据加载数据图表

myChart.setOption({

xAxis : {

data : names,

axisLabel:{

interval:0,//横轴信息全部显示

}

},

series : [ {

// 根据名字对应到相应的数据

name : '人数',

data : nums,

itemStyle: {

normal: {

color: **function**(params) {

// build a color map as your need.

**var** colorList = [

'#FF0000','#FF7D00','#FFFF00','#00FF00','#00FFFF',

'#0000FF','#FF00FF','#FAD860','#F3A43B','#60C0DD',

'#D7504B','#C6E579','#F4E001','#F0805A','#26C0C0'

];

**return** colorList[params.dataIndex];

},

}

},

} ]

});

}

},

error : **function**(errorMsg) {

//请求失败时执行该函数

alert("图表请求数据失败!");

myChart.hideLoading();

}

})

</script>

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

String sql = "select \* from v\_count\_college order by number desc";

//System.out.println(sql);

List<Product> list = **new** ArrayList<Product>();

DBBean db = **new** DBBean();

db.getConnection();

ResultSet rs = db.executeQuery(sql);

**try** {

**while**(rs.next())

{

list.add(**new** Product(rs.getString(1), rs.getInt(2)));

System.***out***.println(rs.getString(1));

}

}**catch**(Exception e) {

e.printStackTrace();

}

System.***out***.println("这里是doPost");

//定义一个list集合

Gson gson2 = **new** Gson();

String json = gson2.toJson(list);

System.***out***.println(json);

// 将json字符串数据返回给前端

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

response.getWriter().write(json);

}

#### (2)按课程统计



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

Connection conn = DriverManager.getConnection(url);

Statement stat = conn.createStatement();

String sql = "select v\_selected\_stu\_tea\_cou.Cno,v\_selected\_stu\_tea\_cou.Cname,COUNT(\*) ";

sql+="from v\_selected\_stu\_tea\_cou,course ";

sql+="where v\_selected\_stu\_tea\_cou.Cno = course.Cno ";

sql+="group by Cno ";

sql+="order by COUNT(\*) desc ";

System.out.println("sql "+sql);

ResultSet rs = stat.executeQuery(sql);

%>

#### (3)按学生统计



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

//Class c = new Class(driverName);

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

//建立数据库连接

Connection conn = DriverManager.getConnection(url);

//out.println(conn);//能输出就连接成功了

//建立语句执行环境

Statement stat = conn.createStatement();

//执行sql语句

String sql = "select Sno,Sname,count(\*) ";

sql+="from v\_selected\_stu\_tea\_cou ";

sql+="group by Sno ";

sql+="order by count(\*) desc";

System.out.println("sql "+sql);

//弄个对象来接它，结果集

ResultSet rs = stat.executeQuery(sql);

//rs.last();//先到最后一行

//out.println("共有"+rs.getRow()+"人");//看看行号输出

%>

#### (4)按老师统计



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

Connection conn = DriverManager.getConnection(url);

Statement stat = conn.createStatement();

String sql = "select Tno,Tname,count(\*) ";

sql+="from v\_selected\_stu\_tea\_cou ";

sql+="group by Tno ";

sql+="order by count(\*) desc";

System.out.println("sql "+sql);

ResultSet rs = stat.executeQuery(sql);

%>

#### (5)按学生学分统计



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

//数据库连接实例

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

//Class c = new Class(driverName);

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

//建立数据库连接

Connection conn = DriverManager.getConnection(url);

//out.println(conn);//能输出就连接成功了

//建立语句执行环境

Statement stat = conn.createStatement();

//执行sql语句

String sql = "select Sno,Sname,SUM(course.Credit)";

sql+="from v\_selected\_stu\_tea\_cou,course ";

sql+="where v\_selected\_stu\_tea\_cou.Cno = course.Cno ";

sql+="group by Sno ";

sql+="order by SUM(course.Credit) desc ";

System.out.println("sql "+sql);

//弄个对象来接它，结果集

ResultSet rs = stat.executeQuery(sql);

%>

#### (6)未选课学生



<%

String user =(String)session.getAttribute("username");

session.setAttribute("username", user);

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

//建立数据库连接

Connection conn = DriverManager.getConnection(url);

Statement stat = conn.createStatement();

//执行sql语句

String sql = "select \* ";

sql+="from student ";

sql+="where Sno not in ";

sql+="(select Sno from v\_selected\_stu\_tea\_cou)";

System.out.println("sql "+sql);

ResultSet rs = stat.executeQuery(sql);

%>

### 2.5.6 教师信息查看



<%

String driverName = "com.mysql.jdbc.Driver";

Class.forName(driverName).newInstance();

String url="jdbc:mysql://localhost:3306/courseselectsystem?user=root&password=caiyishuai";

url += "&useUnicode=true&characterEncoding=utf8";

Connection conn = DriverManager.getConnection(url);

Statement stat = conn.createStatement();

String sql = "select \* from teacher ";

ResultSet rs = stat.executeQuery(sql);

%>

# 三、软件所涉及到的数据表：

#### (1)用户登录表(user)



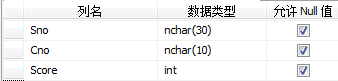
#### (2)学生信息表(student)



#### (3)课程信息表(course)



#### (4)成绩信息表(cscore)



#### (5)学生选课表(selection)



# 四、Sql server中的代码（事务，视图，触发器，存储过程）

## 4.1触发器

**触发器:**使Student表学号位数保持4位且必须以“1”或“2”或“3”或“4”开头，否则撤销相应操作；

CREATE TRIGGER tri\_insert\_sno

ON student

FOR INSERT

AS

DECLARE @sno VARCHAR(8)

SELECT @sno = Sno FROM inserted

IF(LEN(@sno)=4)

BEGIN

IF(@sno LIKE '1%' or @sno LIKE '2%'or @sno LIKE '2%'or @sno LIKE '2%')

PRINT '增加成功'

END

ELSE

BEGIN

RAISERROR('学号不对',16,1)

ROLLBACK TRANSACTION

END

GO

**触发器:**使Teacher表学号位数保持3位且必须以“1”开头，否则撤销相应操作；

CREATE TRIGGER tri\_insert\_tno

ON teacher

FOR INSERT

AS

DECLARE @tno VARCHAR(8)

SELECT @tno = Tno FROM inserted

IF(LEN(@tno)=3)

BEGIN

IF(@tno LIKE '1%')

PRINT '增加成功'

END

ELSE

BEGIN

RAISERROR('工号不对',16,1)

ROLLBACK TRANSACTION

END

GO

**触发器:**若Course表中的课程号发生变化，在SC表中的对应课程号也相应变化

CREATE TRIGGER tri\_update\_changeCno

ON course

FOR UPDATE

AS

DECLARE @CnoA VARCHAR(8),@CnoB VARCHAR(8)

SELECT @CnoB = Cno FROM deleted

SELECT @CnoA = Cno FROM inserted

IF UPDATE(Cno)

BEGIN

UPDATE cscore SET Cno = @CnoA

WHERE Cno = @CnoB

END

GO

**触发器:**在学生信息表Student上创建触发器，当数据删除学生记录时，判断如果该学生已经有考试成绩，则不允许删除；

CREATE TRIGGER tri\_delete\_Dgrade

ON student

FOR DELETE

AS

IF EXISTS(SELECT cscore.Sno FROM cscore,deleted WHERE deleted.Sno = cscore.Sno AND Score IS NOT NULL)

BEGIN

PRINT 'NOT ALLOWED TO DELETE'

RAISERROR('THE STUDENT Grade IS NOT NULL',16,1)

ROLLBACK TRANSACTION

END

GO

**触发器:**当删除学生表中的学生记录时，同步删除选课表中该学生的选课信息；

CREATE TRIGGER tri\_delete\_DSinfo

ON student

FOR DELETE

AS

IF EXISTS(SELECT cscore.Sno FROM deleted,Score WHERE cscore.Sno = deleted.Sno)

DELETE FROM cscore WHERE Sno IN(SELECT Sno FROM deleted)

GO

在mysql里

create trigger del\_stu

after delete on student

for each row

delete from selection where Sno = old.Sno

**触发器:**当增加学生表中的学生记录时，同时增加登录信息表的账号密码

CREATE TRIGGER tri\_insert\_account

ON student

AFTER INSERT

AS

BEGIN INSERT INTO user(uid,ucode)

SELECT Sno,Sno

FROM INSERTED

END

**触发器:**当增加教师表中的学生记录时，同时增加登录信息表的账号密码

在mysql里

CREATE TRIGGER tri\_insert\_Taccount

ON teacher

AFTER INSERT

AS

BEGIN INSERT INTO user(uid,ucode)

SELECT Tno,Tno

FROM INSERTED

END

**触发器:**删除一个老师时，则删除所有该老师开的所有课的选课记录

create trigger del\_tea

after delete on teacher

for each row

delete from selection where Cno in(

select Cno from course where Tno = old.Tno

)

## 4.2视图

**视图:**创建学生成绩展示视图

CREATE VIEW v\_grade AS

select student.Sno,student.Sname,course.Cno,course.Cname,cscore.Score

from student,course,cscore

where cscore.Sno=student.Sno and cscore.Cno=course.Cno;

**视图:**创建学生选课结果展示视图

create view v\_selected as

select selection.Sno,student.Sname,selection.Cno,course.Cname

from student,course,selection

where student.Sno=selection.Sno and course.Cno=selection.Cno;

**视图:**创建学号和学院的视图

create view v\_selected\_college as

select v\_selected\_admin.Sno,College from v\_selected\_admin,course where course.Cno=v\_selected\_admin.Cno;

**视图:**创建按学院统计选课人数的视图

create view v\_count\_college as

select College,count(\*) from v\_selected\_college group by College;

**视图:**创建选课呈现给学生的视图

create view v\_choose\_course as

select Cno,Cname,Tname,Time,TestTime,Credit,College

from course,teacher

where teacher.Tno=course.Tno

**视图:**创建含学生老师课程的已选课视图

create view v\_selected\_stu\_tea\_cou as

select selection.Sno,Sname,teacher.Tno,Tname,selection.Cno,course.Cname

from selection,course,teacher,student

where selection.Cno = course.Cno

and teacher.Tno = course.Tno

and selection.Sno = student.Sno

# 五、课程设计心得

通过此次课程设计，加深了我们对Java这门语言和数据库操作的了解，熟悉了数据库的工作原理。在整个过程中，我们遇到了许多的问题，包括对一些控件功能的不了解，调试时出现的种种错误，效果与预期不相符，以及对两种语言的结合使用等等。通过查找网络资料以及翻阅教辅书，最终将所遇到的问题逐一解决。