

Web 应用技术 课程实验报告

学号：201620130145	姓名：王光雨	班级：2 班
-----------------	--------	--------

实验题目：Java 访问数据库

实验内容：

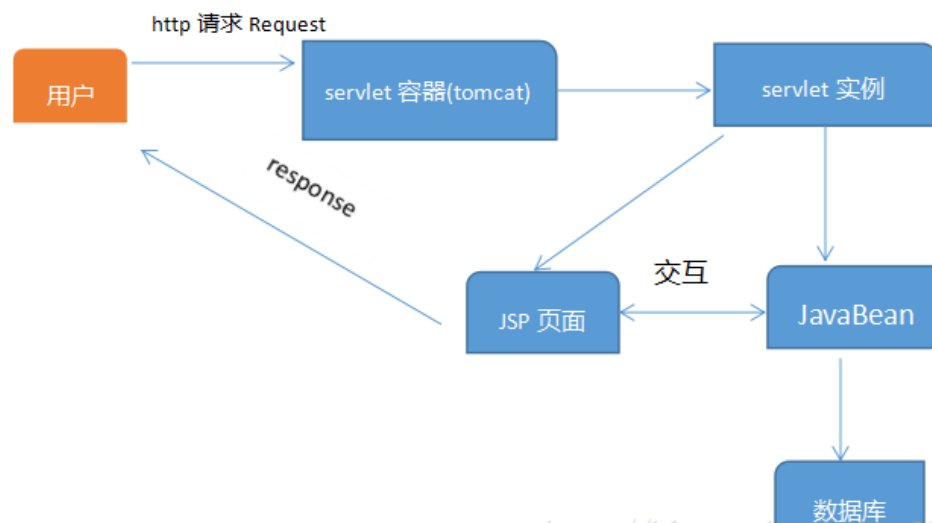
一、实验要求：

完成数据录入，修改，查询，删除功能

要求：提交网页截图和源代码，用不同颜色标出相关代码，并给出文字说明

二、实验说明：

既然是要实现数据库的访问，自然而然要选择一种语言来处理各种事务，这里我选择使用 java 来实现。现在很多网站都是使用框架来进行开发，不过为了奠定基础，更好的锻炼自己的能力，我选择没有框架的开发。这里只是说不使用现成的框架，项目本身的框架还是要有的，如下图所示：



http://blog.csdn.net/qq_33745102

1. 下面我会给出框架的整体说明，具体的细节我会在实验过程一环节讲到

①servlet

Servlet 是 Java Web 技术的核心基础，Servlet 本质就是一个服务端的 java 对象，在这个对象中有一些方法来对浏览器的请求进行处理，并反馈给浏览器。

②servlet 与 servlet 容器

首先我们应该理解 Servlet 与 Servlet 容器的区别与联系，Servlet 容器是用来管理众多 Servlet 的，有个比喻很好，Servlet 容器和 servlet 就和枪与子弹的关系一样，互相依存但又互相独立，在 tomcat 中 Servlet 容器所对应的部分是 Context 容器，而一个 web 应用对应一个 Context 容器，因此一个 web 应用只有一个 Servlet 容器。在这里不扯 tomcat 的内部实现方式，我们只需要记住

Servlet 容器所做的工作就是在程序运行开始之前对系统配置文件进行解析，将相关的 Servlet 对象、filter、listener 等信息进行整合与处理。

③JSP

JSP 全名为 Java Server Pages，中文名叫 java 服务器页面，是一种动态网页技术标准，与传统的 html 不同，jsp 是在服务端运行的，这里要明确一点 jsp 不是一种语言，而是一种技术，运用这种技术可以在 html 中用<% %>来实现 java 语言的书写，也就是在 html 页面实现了动态编程，但 jsp 页面还是属于前端页面部分。也就是说我们可以将我们之前完成的 html 界面转移到 jsp 中，只需要做轻微的修改。

④Servlet 与 JSP

那么 Servlet 与 jsp 的关系到底是什么，为什么 jsp 页面中还可以写 java 代码，这就得从他们的运行机制说起。一个工程运行后会加载所有的 Servlet 对象，同时也会扫描所有的 jsp 页面，将 jsp 页面转化为 Servlet，在进行统一处理，因此一个 jsp 页面实质上也就是一个 Servlet。

那么有一个问题了，既然二者相似那么为什么还要分出两个东西，其实用 jsp 我们完全可以实现一个简单的小网站，需要 Servlet 的原因就是为了前后台的分离，也就是所谓的解耦，web 技术发展的很大一个推动力就是为了解耦，很多的技术都是为此而具有巨大优势，解耦的好处①提高问题的解决概率；②提高问题的解决效果；③提高问题的解决速度；④降低将来爆发隐患的可能性。

⑤JavaBean

你可能见过这个词，也可能没见过，我稍后会给出说明

首先明确，JavaBean 是一个遵循特定写法的 Java 类，它通常具有如下特点：

- 这个 Java 类必须具有一个无参的构造函数
- 属性必须私有化
- 私有化的属性必须通过 public 类型的方法暴露给其他程序，并且方法的命名也必须遵守一定的命名规范。

JavaBean 的属性可以是任意类型，并且一个 JavaBean 可以有多个属性。每个属性通常都需要具有相应的 setter、getter 方法，setter 方法称为属性修改器，getter 方法称为属性访问器。

- ◆ 属性修改器必须以小写的 set 前缀开始，后跟属性名，且属性名的第一个字母要改为大写，例如 name 属性的修改器名称为 setName，password 属性的修改器名称为 setPassword。
- ◆ 属性访问器通常以小写的 get 前缀开始，后跟属性名，且属性名的第一个字母也要改为大写，例如 name 属性的访问器名称为 getName，password 属性的访问器名称为 getPassword。
- ◆ 一个 JavaBean 的某个属性也可以只有 set 方法或 get 方法，这样的属性通常也称之为只写、只读属性。

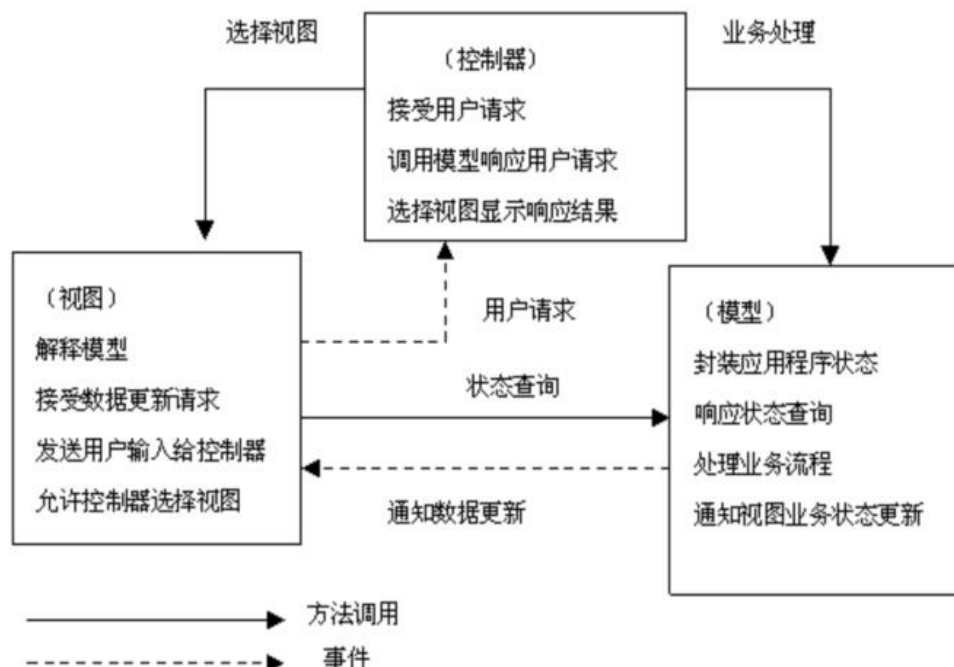
2. 前后端交互可以使用 form 表单的方式、URL 的方式或 COOKIES 的方式，这里我实现了前两种方式，但是现在的正规网站都会采用 COOKIES 技术，有兴趣可以继续深入了解。

三、模式说明：

学习过基本 Java Web 开发的人都已经了解了如何编写基本的 Servlet，如何编写 jsp 及如何更新浏览器中显示的内容。Java 是一门应用设计模式比较广泛

的语言。目前主流提出的 23 种设计模式均可在 Java 语言编写的程序中所应用。目前主流在 Java Web 应用中应用的最广泛的设计模式便是 MVC 模式，目前的主流 Web 框架大多也是基于 MVC 设计模式所编写的。

MVC 设计模式图：



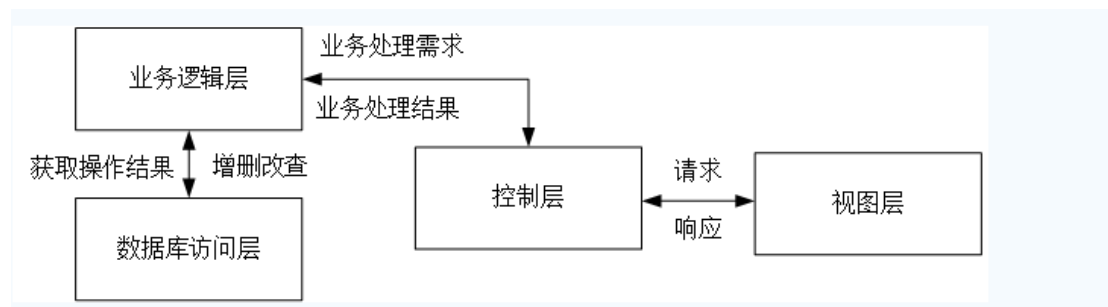
所谓 MVC，即 Model-View-Controller。

(1) Model 层：Model 指模型部分，一般在应用中 Model 层包括业务处理层和数据访问层。数据访问层主要是对数据库的一些操作的封装。业务处理层应用 JavaBean 构建，JavaBean 主要是用作将从 View 层获取的数据和数据库的数据进行桥接。除却 JavaBean 以外，若想构建分布式应用系统，可以应用 EJB 组件进行业务逻辑层的构建。

(2) Controller 层：Controller 指控制部分，一般是对 View 层提交的请求为其设置对应的 Servlet 进行特定功能的处理，这里的进行特定功能的处理一般是编写在 Model 中的业务处理层中的。Controller 一般只是在 Web 应用中充当一个中介者的作用。

(3) View 层：View 指视图部分，这一部分的内容是展示给用户实际进行交互的，通常使用 JSP 和 HTML 进行构建（个人比较喜欢以 HTML 嵌入 JSP 的方式来构建网页）。

综上所述，一个小型完整的基于 MVC 设计模式的 Web 应用程序的处理流程应该如下：



由上面的图中我们可以看出，用户在客户端（Web 应用的客户端即为浏览器）中发出请求的时候，请求首先由 View 层的 JSP/HTML 将 HTTP 请求传给控制器中对应的 Servlet，然后由 Servlet 负责调用 Model 层中的业务逻辑处理部分进行要

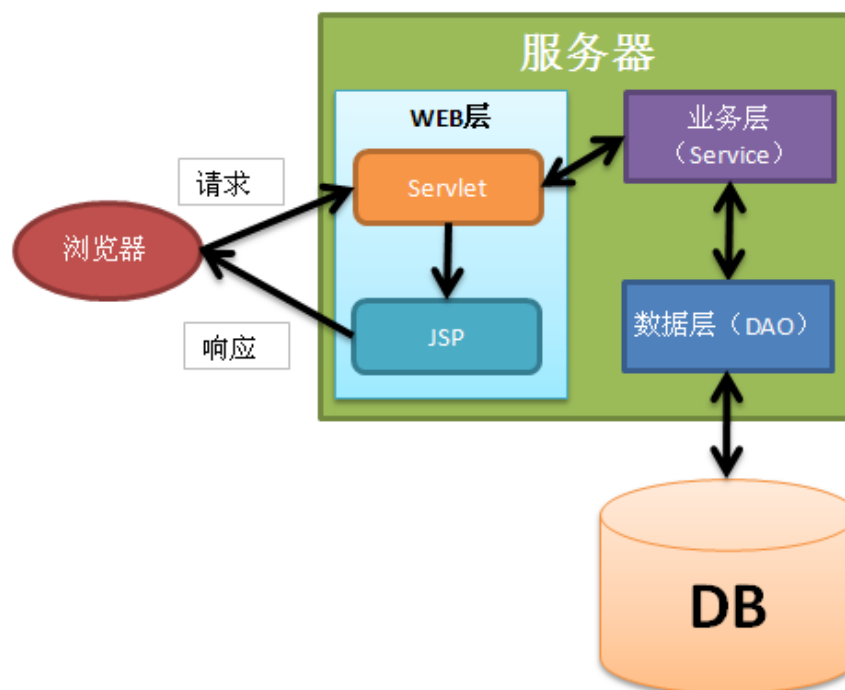
求的处理，处理期间如果设计数据库的操作，则与数据库进行操作，最后全部操作结束之后，由业务逻辑层将结果发给控制层，控制层以 HTTP 响应的形式将结果发送回客户端。

我在本实验中使用的是 JavaWeb 经典三层框架，我们常说的三层框架是由 JavaWeb 提出的，也就是说 JavaWeb 独有的，这里所谓的三层是

表述层 (WEB 层)、业务逻辑层 (Business Logic)，以及数据访问层 (Data Access)

- WEB 层：包含 JSP 和 Servlet 等与 WEB 相关的内容；
- 业务层：业务层中不包含 JavaWeb API，它只关心业务逻辑；
- 数据层：封装了对数据库的访问细节；

注意，在业务层中不能出现 JavaWeb API，例如 request、response 等。也就是说，业务层代码是可重用的，甚至可以应用到非 Web 环境中。业务层的每个方法可以理解成一个万能，例如转账业务方法。业务层依赖数据层，而 Web 层依赖业务层！如下图所示：



四、实验内容：

1. View 层的编写

a. jsp 的页面跳转

①功能：点击教师框和学生框后跳转到登录界面。

说明：我们实现用户的登录界面，并在表单中指定接受请求并处理该页面的 servlet，view 层主要是使用 jsp 和 html 编写而成的。

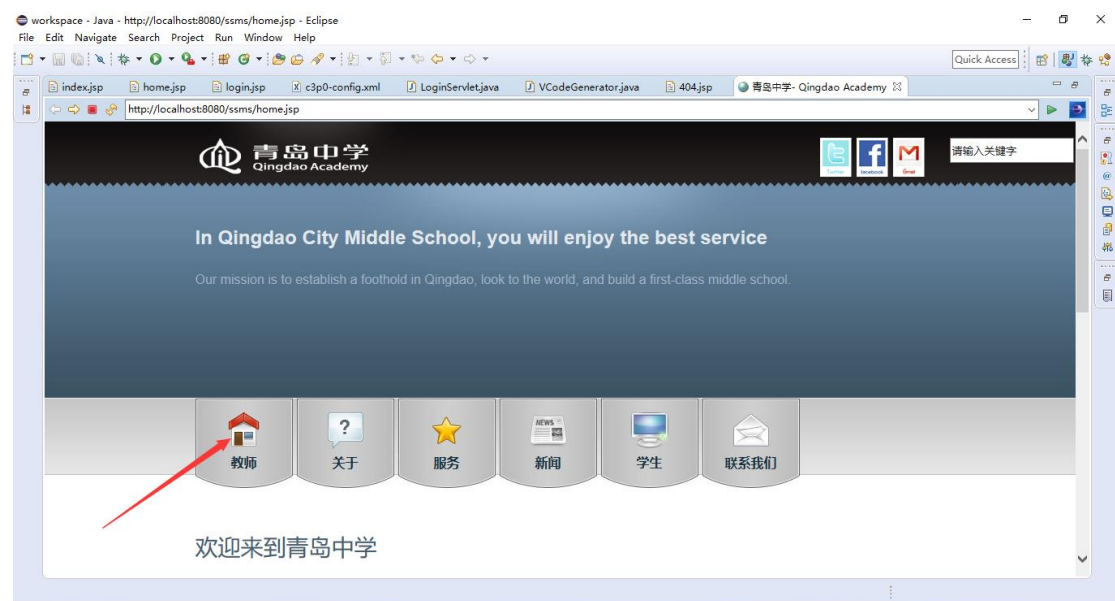
②代码实现如下：

```

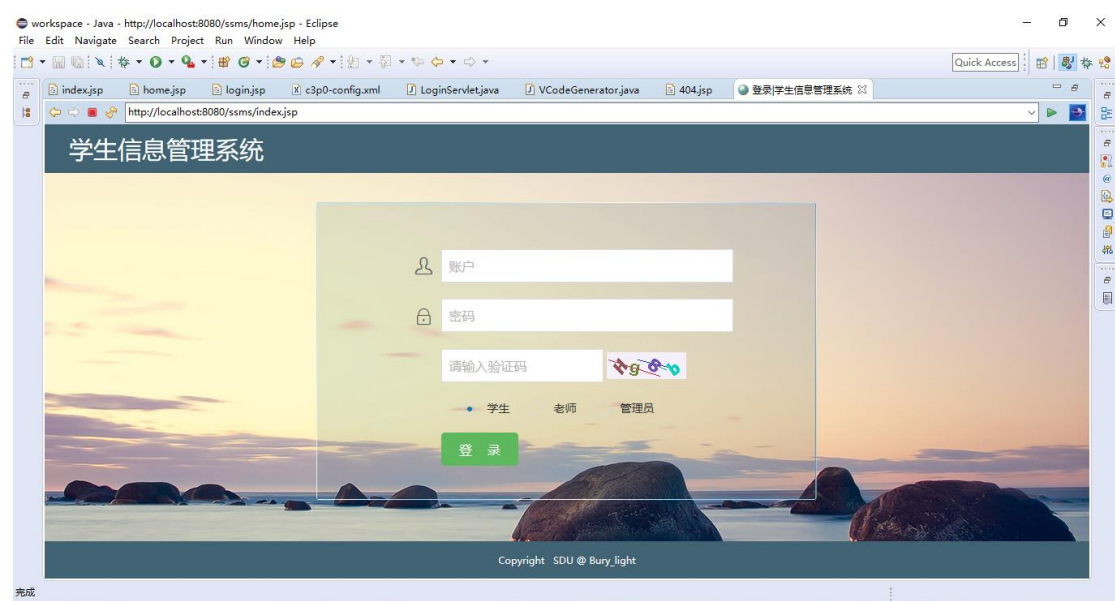
<div id="templatemo_menu">
  <div class="current">
    <ul>
      <li><a href="index.jsp"><span class="teacher">教师</span></a></li>
      <li><a href="content.jsp"><span class="aboutus">关于</span></a></li>
      <li><a href="#"><span class="services">服务</span></a></li>
      <li><a href="#"><span class="news">新闻</span></a></li>
      <li><a href="index.jsp"><span class="student">学生</span></a></li>
      <li><a href="#"><span class="contact">联系我们</span></a></li>
    </ul>
  </div>
</div> <!-- end of templatemo_menu -->

```

③效果展示



点击教师框后跳转到对应的 Jsp 登录界面



b. form 表单实现

代码实现: Login.jsp 中直接定义 form 表单

```

<div class="loginWrapper">

```

```

<div id="loginform" class="loginBox">
  <form id="form" class="form form-horizontal" method="post">
    <div class="row cl">
      <label class="form-label col-3"><i
class="Hui-iconfont">&#xe60d;</i></label>
      <div class="formControls col-8">
        <input id="" name="account" type="text"
placeholder="账户"
          class="input-text size-L">
      </div>
    </div>
    <div class="row cl">
      <label class="form-label col-3"><i
class="Hui-iconfont">&#xe60e;</i></label>
      <div class="formControls col-8">
        <input id="" name="password" type="password"
placeholder="密码"
          class="input-text size-L">
      </div>
    </div>
    <div class="row cl">
      <div class="formControls col-8 col-offset-3">
        <input class="input-text size-L" name="vcode"
type="text"
          placeholder="请输入验证码" style="width: 200px;">
        
      </div>
    </div>

    <div class="mt-20 skin-minimal" style="text-align:
center;">
      <div class="radio-box">
        <input type="radio" id="radio-2" name="type"
checked value="2" />
        <label for="radio-1">学生</label>
      </div>
      <div class="radio-box">
        <input type="radio" id="radio-3" name="type"
value="3" /> <label
          for="radio-2">老师</label>
      </div>
      <div class="radio-box">

```



```
<input type="radio" id="radio-1" name="type"  
value="1" /> <label  
  
for="radio-3">管理员</label>  
  
</div>  
</div>  
  
<div class="row">  
  <div class="formControls col-8 col-offset-3">  
    <input id="submitBtn" type="button"  
      class="btn btn-success radius size-L"  
      value="登&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&录  
&nbsp;&nbsp;& ">  
  
    </div>  
  </div>  
</form>  
</div>  
</div>
```

2. controller 层的编写

①说明:controller 层主要编写接受 view 层的 serlet

②代码实现:

```
package com.lizhou.servlet;

import java.awt.image.BufferedImage;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.lizhou.bean.User;
import com.lizhou.service.SystemService;
import com.lizhou.tools.VCodeGenerator;

/**
 * 验证码
 * @author bury_light
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private SystemService service = new SystemService();
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //获取请求的方法
        String method = request.getParameter("method");
        if("GetVCode".equalsIgnoreCase(method)){
```

```

        getVCode(request, response);
    }
}

protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    //获取请求的方法
    String method = request.getParameter("method");
    if("Login".equals(method)){ //验证登录
        login(request, response);
    }
}

/**
 * 验证用户登录
 * @param request
 * @param response
 * @throws IOException
 */
private void login(HttpServletRequest request, HttpServletResponse
response) throws IOException {
    //获取用户输入的账户
    String account = request.getParameter("account");
    //获取用户输入的密码
    String password = request.getParameter("password");
    //获取用户输入的验证码
    String vcode = request.getParameter("vcode");
    //获取登录类型
    int type = Integer.parseInt(request.getParameter("type"));
    //返回信息
    String msg = "";
    //获取 session 中的验证码
    String sVcode = (String)
request.getSession().getAttribute("vcode");
    if(!sVcode.equalsIgnoreCase(vcode)){//先判断验证码是否正确
        msg = "vcodeError";
    } else{ //判断用户名和密码是否正确
        //将账户和密码封装
        User user = new User();
        user.setAccount(account);
        user.setPassword(password);
        user.setType(Integer.parseInt(request.getParameter("type")));
        //创建系统数据层对象
        //查询用户是否存在
        User loginUser = service.getAdmin(user);
    }
}

```



```

        if(loginUser == null){//如果用户名或密码错误
            msg = "loginError";
        } else{ //正确
            if(User.USER_ADMIN == type){
                msg = "admin";
            } else if(User.USER_STUDENT == type){
                msg = "student";
            } else if(User.USER_TEACHER == type){
                msg = "teacher";
            }
            //将该用户名保存到 session 中
            request.getSession().setAttribute("user", loginUser);
        }
    }

    //返回登录信息
    response.getWriter().write(msg);
}

/**
 * 获取验证码
 * @param request
 * @param response
 * @throws IOException
 */
private void getVCode(HttpServletRequest request,
    HttpServletResponse response) throws IOException {
    //创建验证码生成器对象
    VCodeGenerator vcGenerator = new VCodeGenerator();
    //生成验证码
    String vcode = vcGenerator.generatorVCode();
    //将验证码保存在 session 域中,以便判断验证码是否正确
    request.getSession().setAttribute("vcode", vcode);
    //生成验证码图片
    BufferedImage vImg =
vcGenerator.generatorRotateVCodeImage(vcode, true);
    //输出图像
    ImageIO.write(vImg, "gif", response.getOutputStream());
}
}

```

在 xml 中进行相关配置, 如下所示:

```

<servlet>
    <description>登录</description>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>com.lizhou.servlet.LoginServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/LoginServlet</url-pattern>
</servlet-mapping>
<servlet>
    ...

```

3. Model 层的编写

Model 层在具体的编写中包括 dao 层、bean 层和 service 层

①说明：因为我这里只展示接受用户登录跳转到新的页面，也就是说登录不全部涉及这一层，但是后续的实验内容我会展示出来。另外说明一下，也可以使用 model 层和数据库的连接，但是我这里使用的 JDBC，在下一个实验中我会给出说明。

②bean 层：

```

package com.lizhou.bean;

/**
 * 系统用户类
 * @author bury_light
 *
 */
public class User {

    /**
     * 管理员类型用户
     */
    public static final int USER_ADMIN = 1;

    /**
     * 学生类型用户
     */
    public static final int USER_STUDENT = 2;

    /**
     * 教师类型用户
     */
    public static final int USER_TEACHER = 3;

    private int id; //ID
    private String account; //账户
    private String password = "111111"; //密码：默认'111111'
    private String name; //用户姓名
    private int type = USER_STUDENT; // 账户类型：默认 2 为学生；1 为管理员，
    2 为学生，3 为教师

    public int getId() {
        return id;
    }

    public void setId(int id) {

```

```

        this.id = id;
    }
    public String getAccount() {
        return account;
    }
    public void setAccount(String account) {
        this.account = account;
    }
    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getType() {
        return type;
    }
    public void setType(int type) {
        this.type = type;
    }
}

```

说明：对应提交的数据要在后台使用同一个命名，并设置了 set 和 get 方法。

4. 数据库连接

①说明：用户输入的时候去数据库中查找用户名（标志）和密码，存在的话登录到主页面，否则不能实现登录。

②代码：c3p0-config.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<c3p0-config>
    <default-config>
        <property name="driverClass">com.mysql.jdbc.Driver</property>
        <property
name="jdbcUrl">jdbc:mysql://localhost:3306/ssms</property>
        <property name="user">root</property>
        <property name="password">wang123456</property>
    </default-config>

```

```

    <!-- <default-config> <property
name="driverClass">com.mysql.jdbc.Driver</property>
    <property
name="jdbcUrl">jdbc:mysql://192.168.1.13:3306/df07fb99668844964bc0266
b841e65ed9</property>
    <property name="user">5d8cf882-2db4</property> <property
name="password">aa19a42d-1d5a</property>
    </default-config> -->
</c3p0-config>

```

5. 在 js 中对于数据进行核验

```

<script type="text/javascript">
    $(function(){
        //点击图片切换验证码
        $("#vcodeImg").click(function(){
            this.src="LoginServlet?method=GetVcode&t="+new
Date().getTime();
        });
        //登录
        $("#submitBtn").click(function(){
            if($("#radio-2").attr("checked") &&
"${systemInfo.forbidStudent}" == 1){
                $.messager.alert("消息提醒", "学生暂不能登录系统!",
"warning");
                return;
            }
            if($("#radio-3").attr("checked") &&
"${systemInfo.forbidTeacher}" == 1){
                $.messager.alert("消息提醒", "教师暂不能登录系统!",
"warning");
                return;
            }
            var data = $("#form").serialize();
            $.ajax({
                type: "post",
                url: "LoginServlet?method=Login",
                data: data,
                dataType: "text", //返回数据类型
                success: function(msg){
                    if("vcodeError" == msg){
                        $.messager.alert("消息提醒", "验证码错误!",
"warning");

```

```

        $("#vcodeImg").click();//切换验证码
        $("input[name='vcode']").val(""); //清空验证码输入框
    } else if("loginError" == msg){
        $.messenger.alert("消息提醒", "用户名或密码错误!",
"warning");
        $("#vcodeImg").click();//切换验证码
        $("input[name='vcode']").val(""); //清空验证码输入框
    } else if("admin" == msg){
        window.location.href =
"SystemServlet?method=toAdminView";
    } else if("student" == msg){
        window.location.href =
"SystemServlet?method=toStudentView";
    } else if("teacher" == msg){
        window.location.href =
"SystemServlet?method=toTeacherView";
    }
}
});
});

//设置复选框
$(".skin-minimal input").iCheck({
    radioClass: 'iradio-blue',
    increaseArea: '25%'
});
})
</script>

```

6. 事务的实现

该系统中，所有关于查询的事务基本上都可以由 Select 语句实现，而更新的事务则用 Update 语句，删除的事务用 Delete 语句实现。

项目的结构如下所示，采用 MVC 模式。

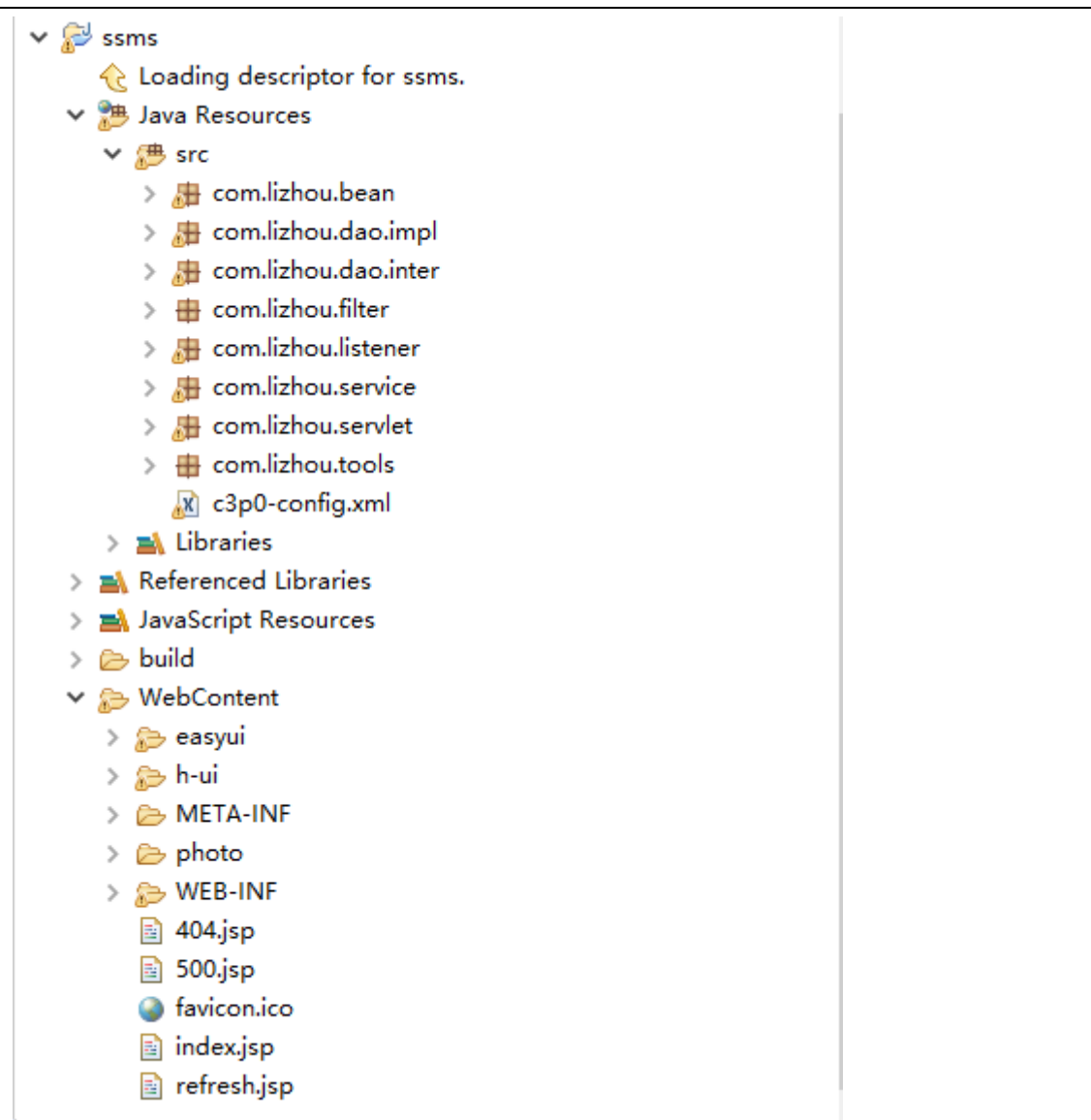
MVC 是三个单词的缩写：Model（模型）；View（视图）；Control（控制）。

MVC 模式的目的是实现 Web 系统的职能分工

Model 层：实现系统的业务逻辑(JavaBean)，实现了类的封装

View 层：负责与用户交互，即在界面上展示数据对象给用户(HTML、JSP 等等)

Control 层：Model 与 View 之间沟通的桥梁，它可以分派用户的请求并选择恰当的视图以用于显示，同时它也可以解释用户的输入并将它们映射为模型层可执行的操作(Servlet)



下面我主要给出的是 `service` 这个包中的代码实例，也就是各种服务的具体实现。因为代码过长，下面我只给出其中一个实际例子。

有关学生的事务包括学生的添加、查询、删除、修改，设置头像等。具体实现代码如下：

```
package com.lizhou.service;

import java.sql.Connection;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import com.lizhou.bean.Page;
import com.lizhou.bean.Student;
import com.lizhou.bean.User;
```

```
import com.lizhou.dao.impl.StudentDaoImpl;
import com.lizhou.dao.inter.StudentDaoInter;
import com.lizhou.tools.MySqlTool;
import com.lizhou.tools.StringTool;

import net.sf.json.JSONObject;

/**
 * 学生信息服务层
 * @author bury_light
 *
 */
public class StudentService {
    private StudentDaoInter dao;
    public StudentService(){
        dao = new StudentDaoImpl();
    }

    /**
     * 修改学生信息
     * @param student
     */
    public void editStudent(Student student){

        String sql = "";

        List<Object> params = new LinkedList<>();
        params.add(student.getName());
        params.add(student.getSex());
        params.add(student.getPhone());
        params.add(student.getQq());

        if(student.getGrade() == null || student.getClazz() == null){
            sql = "UPDATE student SET name=?, sex=?, phone=?, qq=? WHERE
number=?";
        } else{
            sql = "UPDATE student SET name=?, sex=?, phone=?, qq=?, clazzid=?,
gradeid=? WHERE number=?";
            params.add(student.getClazzid());
            params.add(student.getGradeid());
        }
        params.add(student.getNumber());

        //更新学生信息
    }
}
```



```

        dao.update(sql, params);

        //修改系统用户名
        dao.update("UPDATE user SET name=? WHERE account=?",
            new Object[]{student.getName(), student.getNumber()});
    }

    /**
     * 删除学生
     * @param ids 学生id集合
     * @param numbers 学生学号集合
     * @throws Exception
     */
    public void deleteStudent(String[] ids, String[] numbers) throws Exception{
        //获取占位符
        String mark = StringTool.getMark(numbers.length);
        Integer sid[] = new Integer[ids.length];
        for(int i =0 ;i < ids.length;i++){
            sid[i] = Integer.parseInt(ids[i]);
        }

        //获取连接
        Connection conn = MysqlTool.getConnection();
        //开启事务
        MysqlTool.startTransaction();
        try {
            //删除成绩表
            dao.deleteTransaction(conn, "DELETE FROM escore WHERE studentid
IN("+mark+")", sid);
            //删除学生
            dao.deleteTransaction(conn, "DELETE FROM student WHERE id
IN("+mark+")", sid);
            //删除系统用户
            dao.deleteTransaction(conn, "DELETE FROM user WHERE account
IN("+mark+")", numbers);

            //提交事务
            MysqlTool.commit();
        } catch (Exception e) {
            //回滚事务
            MysqlTool.rollback();
            e.printStackTrace();
            throw e;
        } finally {

```

```

        MysqlTool.closeConnection();
    }

}

/**
 * 添加学生
 * @param student
 */
public void addStudent(Student student){

    //添加学生记录
    dao.insert("INSERT INTO student(number, name, sex, phone, qq, clazzid,
gradeid) value(?,?,?,?,?,?,?)",
        new Object[]{
            student.getNumber(),
            student.getName(),
            student.getSex(),
            student.getPhone(),
            student.getQq(),
            student.getClazzid(),
            student.getGradeid()
        });
    //添加用户记录
    dao.insert("INSERT INTO user(account, name, type) value(?,?,?)", new
Object[]{
        student.getNumber(),
        student.getName(),
        User.USER_STUDENT
    });
}

/**
 * 分页获取学生
 * @param student 学生信息
 * @param page 分页
 * @return
 */
public String getStudentList(Student student, Page page){
    //sql语句
    StringBuffer sb = new StringBuffer("SELECT * FROM student ");
    //参数
    List<Object> param = new LinkedList<>();
    //判断条件

```

```

        if(student != null){
            if(student.getGrade() != null){//条件: 年级
                int gradeid = student.getGrade().getId();
                param.add(gradeid);
                sb.append("AND gradeid=? ");
            }
            if(student.getClazz() != null){
                int clazzid = student.getClazz().getId();
                param.add(clazzid);
                sb.append("AND clazzid=? ");
            }
        }
        //添加排序
        sb.append("ORDER BY id DESC ");
        //分页
        if(page != null){
            param.add(page.getStart());
            param.add(page.getSize());
            sb.append("limit ?,?");
        }
        String sql = sb.toString().replaceFirst("AND", "WHERE");
        //获取数据
        List<Student> list = dao.getStudentList(sql, param);
        //获取总记录数
        long total = getCount(student);
        //定义Map
        Map<String, Object> jsonMap = new HashMap<String, Object>();
        //total键 存放总记录数, 必须的
        jsonMap.put("total", total);
        //rows键 存放每页记录 list
        jsonMap.put("rows", list);
        //格式化Map, 以json格式返回数据
        String result = JSONObject.fromObject(jsonMap).toString();
        //返回
        return result;
    }

    /**
     * 获取记录数
     * @param student
     * @param page
     * @return
     */
    private long getCount(Student student){

```

```

//sql语句
StringBuffer sb = new StringBuffer("SELECT COUNT(*) FROM student ");
//参数
List<Object> param = new LinkedList<>();
//判断条件
if(student != null){
    if(student.getGrade() != null){//条件: 年级
        int gradeid = student.getGrade().getId();
        param.add(gradeid);
        sb.append("AND gradeid=? ");
    }
    if(student.getClazz() != null){
        int clazzid = student.getClazz().getId();
        param.add(clazzid);
        sb.append("AND clazzid=? ");
    }
}
String sql = sb.toString().replaceFirst("AND", "WHERE");

long count = dao.count(sql, param).intValue();

return count;
}

/**
 * 获取跟学生一个班的班级同学
 * @param account
 * @param page
 * @return
 */
public String getStudentList(String account, Page page) {

    Student student = (Student) dao.getObject(Student.class, "SELECT * FROM
student WHERE number=?", new Object[]{account});

    return getStudentList(student, page);
}

/**
 * 获取学生详细信息
 * @param account
 * @return
 */
public Student getStudent(String account) {

```

```

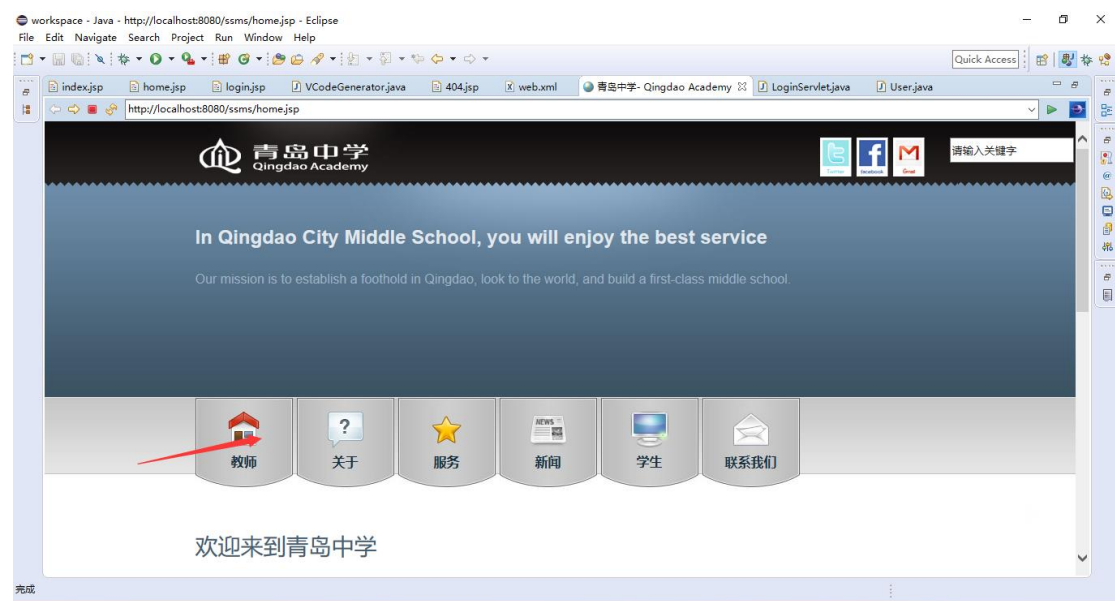
        Student student = dao.getStudentList("SELECT * FROM student WHERE
number="+account, null).get(0);

        return student;
    }

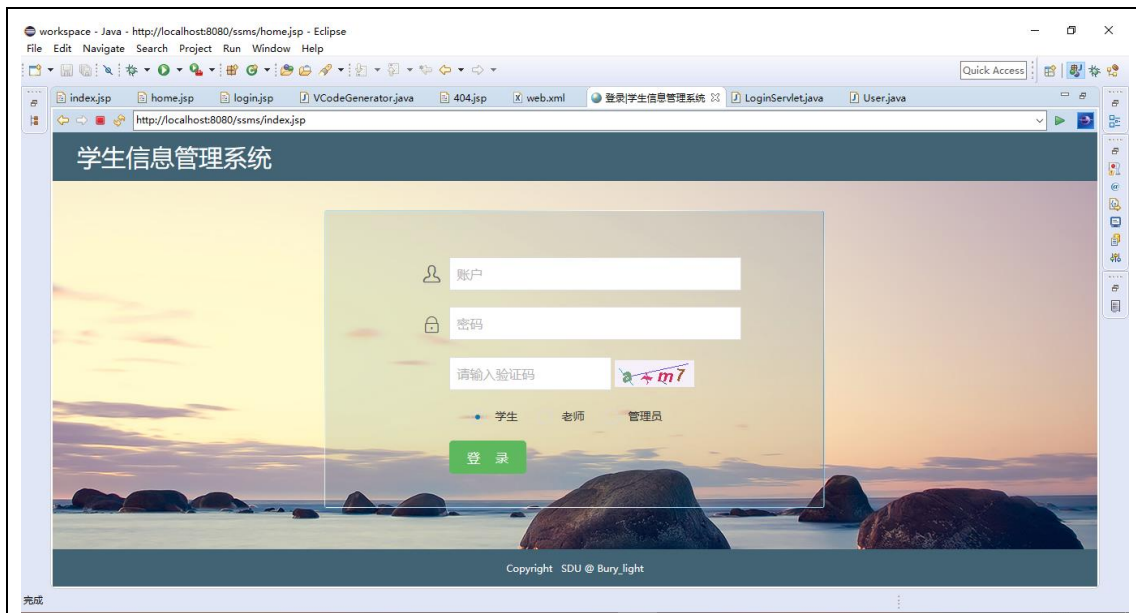
    /**
     * 设置照片
     * @param number
     * @param fileName
     */
    public void setPhoto(String number, String fileName) {
        String photo = "photo/"+fileName;
        dao.update("UPDATE student SET photo=? WHERE number=?", new
Object[]{photo, number});
    }
}

```

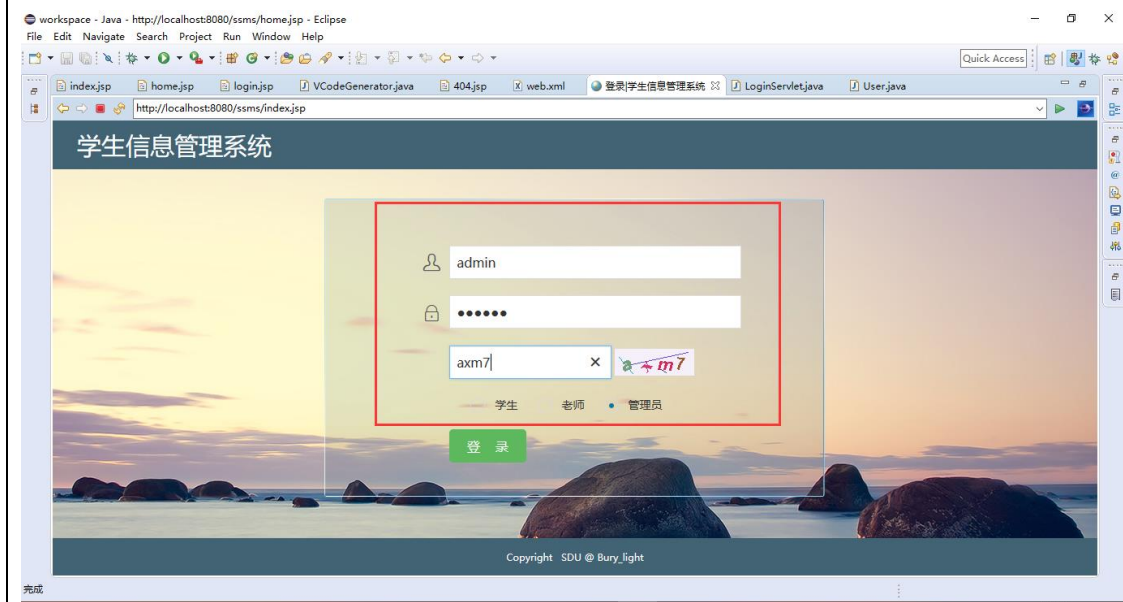
五、实验结果展示



点击教师框，跳转到登录界面



输入用户名、密码和验证码（这里我使用管理员身份登录，密码默认是 111111）



数据库中有 user 表记录，如下所示：

Unnamed

information_schema8.0 KiB

library

mysql

score

ssms480.0 KiB

clazz32.0 KiB

clazz_course_tea...80.0 KiB

course16.0 KiB

escore96.0 KiB

exam48.0 KiB

grade16.0 KiB

grade_course48.0 KiB

student64.0 KiB

system16.0 KiB

teacher32.0 KiB

user32.0 KiB

ssms1

ssms.user: 42 总记录 (大约)

id

account

password

name

type

1	admin	111111	管理员	1
2	1978001	womenhuizai17	张勇	3
3	201501001	111111	王志强	2
4	201501002	111111	王志强	2
5	201502001	111111	李志强	2
6	201502002	111111	李志强	2
7	201503001	111111	张志勇	2
8	201503002	111111	张志强	2
9	201504001	111111	刘志勇	2
11	201601001	111111	林小斌	2
12	201601002	111111	林大斌	2
13	201602001	111111	王小斌	2
14	201602002	111111	王大斌	2
15	201603001	111111	张小斌	2
16	201603002	111111	张大斌	2
17	201604001	111111	刘小斌	2
18	201604002	111111	刘大斌	2
27	1978002	111111	王勇	3
28	1978003	111111	李勇	3
29	1979001	111111	刘木	3

登录成功后进入主界面

workspace - Java - http://localhost:8080/ssms/home.jsp - Eclipse

File Edit Navigate Search Project Run Window Help

index.jsp

home.jsp

login.jsp

VCodeGenerator.java

404.jsp

web.xml

学生信息管理系统 管理员后台

LoginServlet.java

User.java

http://localhost:8080/ssms/SystemServlet?method=toAdminView

SDU学生信息管理系统

管理员 登录 安全退出

导航菜单

成绩统计分析

三考试列表

学生信息管理

教师信息管理

基础信息管理

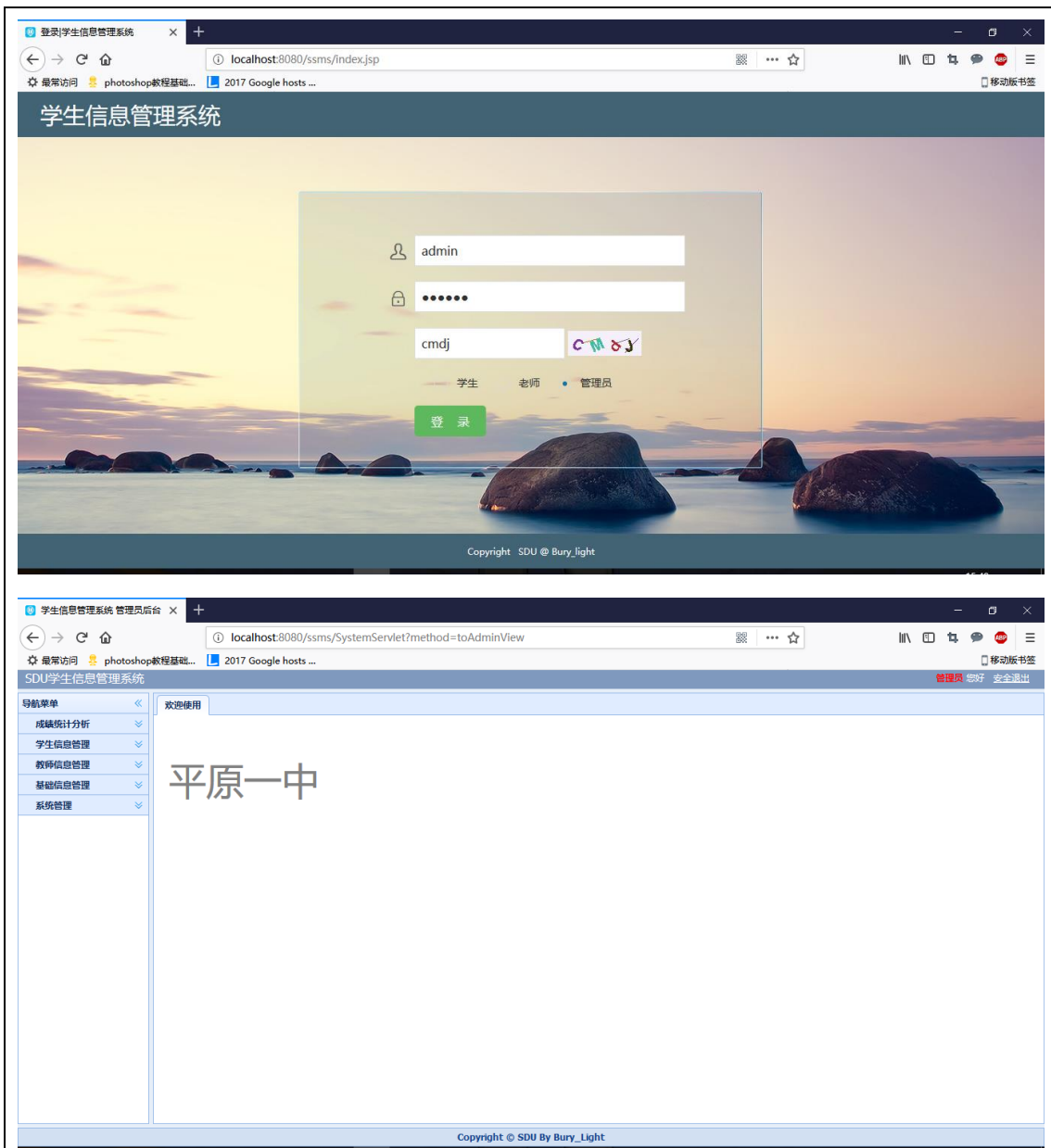
系统管理

欢迎使用

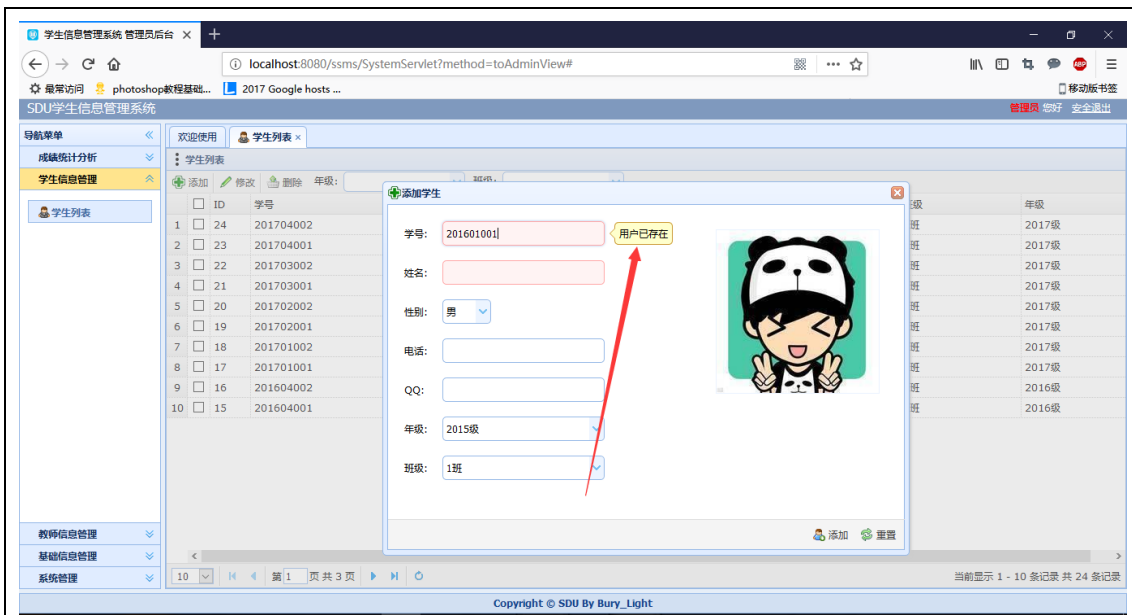
平原一中

Copyright © SDU By Bury_Light

1. 管理员登录



(1) 学生信息管理：添加、修改、查看、删除学生信息
添加学生时若账号已存在会提示用户已存在



添加若干学生的数据库中学生表

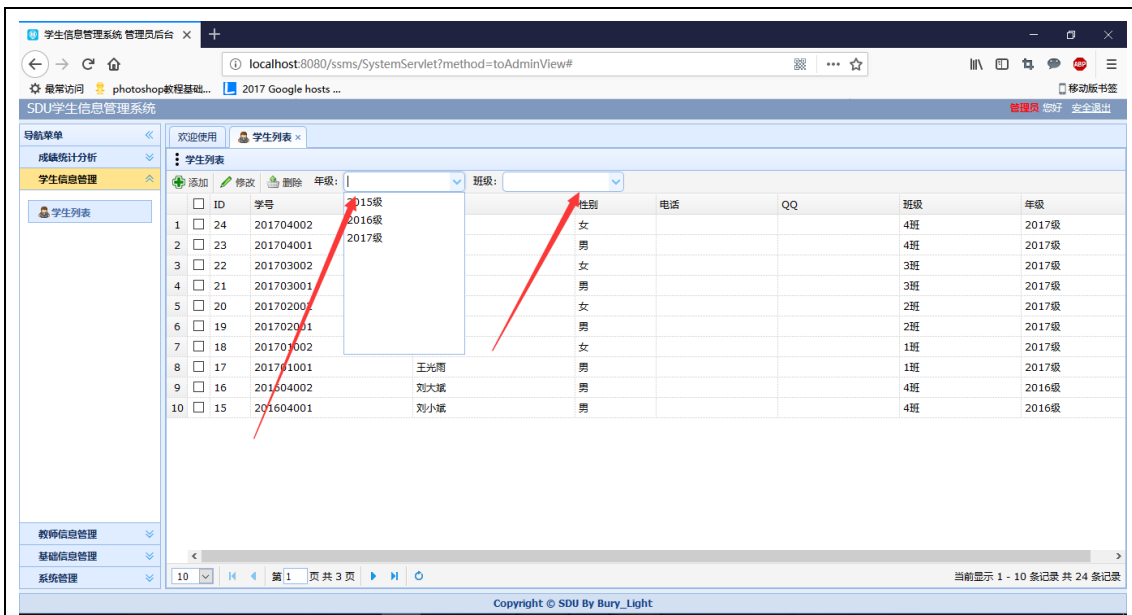
student @ssms (mysql80) - 表

id	number	name	sex	phone	qq	photo	clazzid	gradeid
1	201501001	王志勇	男			(Null)	1	1
2	201501002	王志强	男			(Null)	1	1
3	201502001	李志强	男			(Null)	2	1
4	201502002	李志勇	男			(Null)	2	1
5	201503001	张志勇	男			(Null)	3	1
6	201503002	张志强	男			(Null)	3	1
7	201504001	刘志勇	男			(Null)	4	1
8	2015002	李志强	男			(Null)	4	1
9	201601001	林小斌	男			(Null)	5	2
10	201601002	林大斌	男			(Null)	5	2
11	201602001	王小斌	男			(Null)	6	2
12	201602002	王大斌	男			(Null)	6	2
13	201603001	张小斌	男			(Null)	7	2
14	201603002	张大斌	男			(Null)	7	2
15	201604001	刘小斌	男			(Null)	8	2
16	201604002	刘大斌	男			(Null)	8	2
17	201701001	王光雨	男			(Null)	9	3
18	201701002	王光月	女			(Null)	9	3
19	201702001	王春林	男			(Null)	10	3

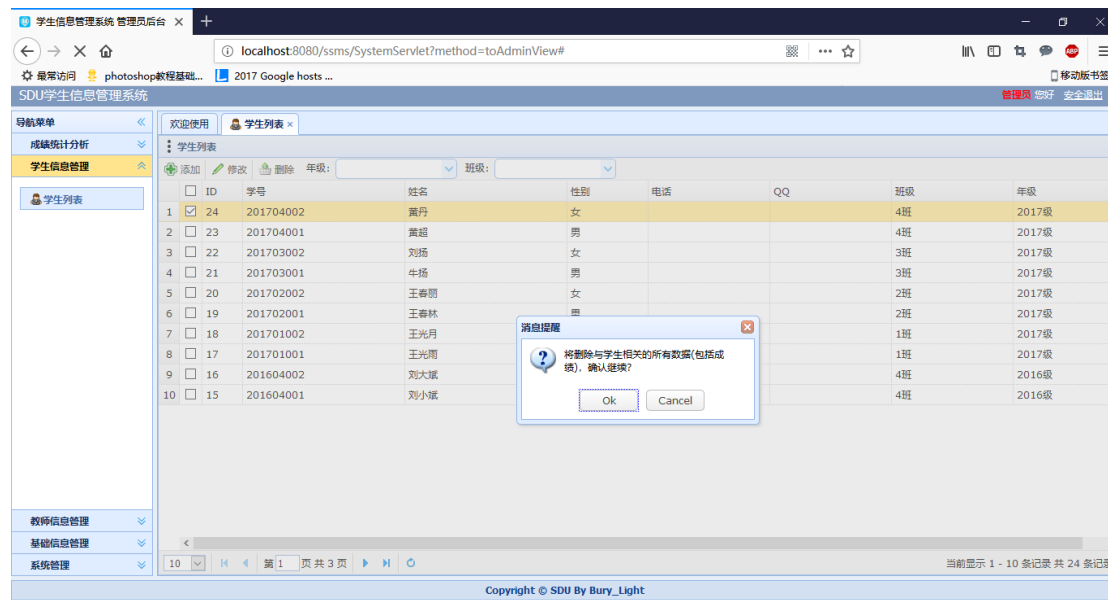
SELECT * FROM `student` LIMIT 0

第 1 条记录 (共 24 条) 于 1 页

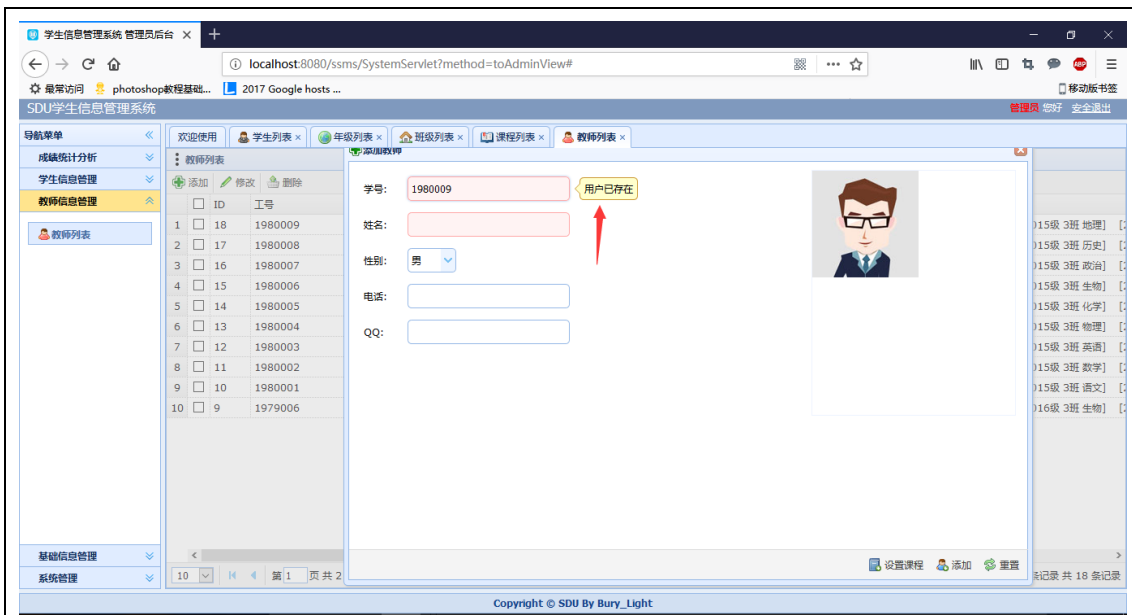
可以查看某个年级下某个班级的学生



删除某个学生的信息，可同时删除多个



(2) 教师信息管理：添加、修改、查看、删除教师信息
添加时，若账号已存在会提示该用户已存在



添加若干教师信息后的教师表

teacher @ssms (mysql80) - 表

文件 编辑 查看 窗口 帮助

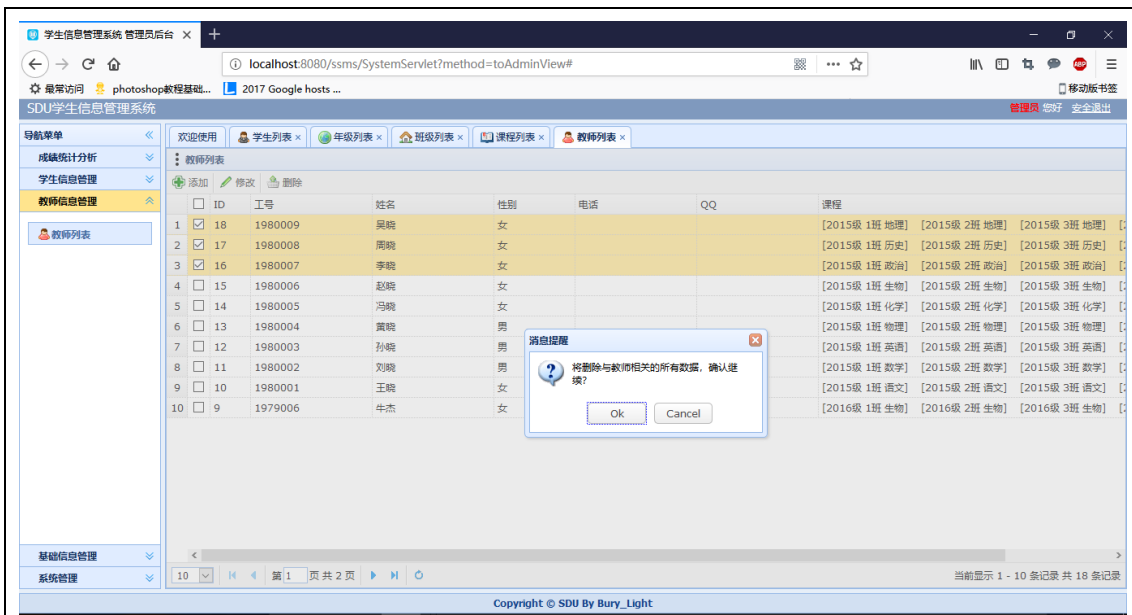
导入向导 导出向导 筛选向导 网格查看 表单查看 备注 十六进制 图像 A-Z 升序排序

id	number	name	sex	phone	qq	photo
1	1978001	张勇	男			(Null)
2	1978002	王勇	男			(Null)
3	1978003	李勇	男			(Null)
4	1979001	刘杰	男			(Null)
5	1979002	王杰	男			(Null)
6	1979003	李杰	女			(Null)
7	1979004	林杰	男			(Null)
8	1979005	孙杰	男			(Null)
9	1979006	牛杰	女			(Null)
10	1980001	王晓	女			(Null)
11	1980002	刘晓	男			(Null)
12	1980003	孙晓	男			(Null)
13	1980004	黄晓	男			(Null)
14	1980005	冯晓	女			(Null)
15	1980006	赵晓	女			(Null)
16	1980007	李晓	女			(Null)
17	1980008	周晓	女			(Null)
18	1980009	吴晓	女			(Null)

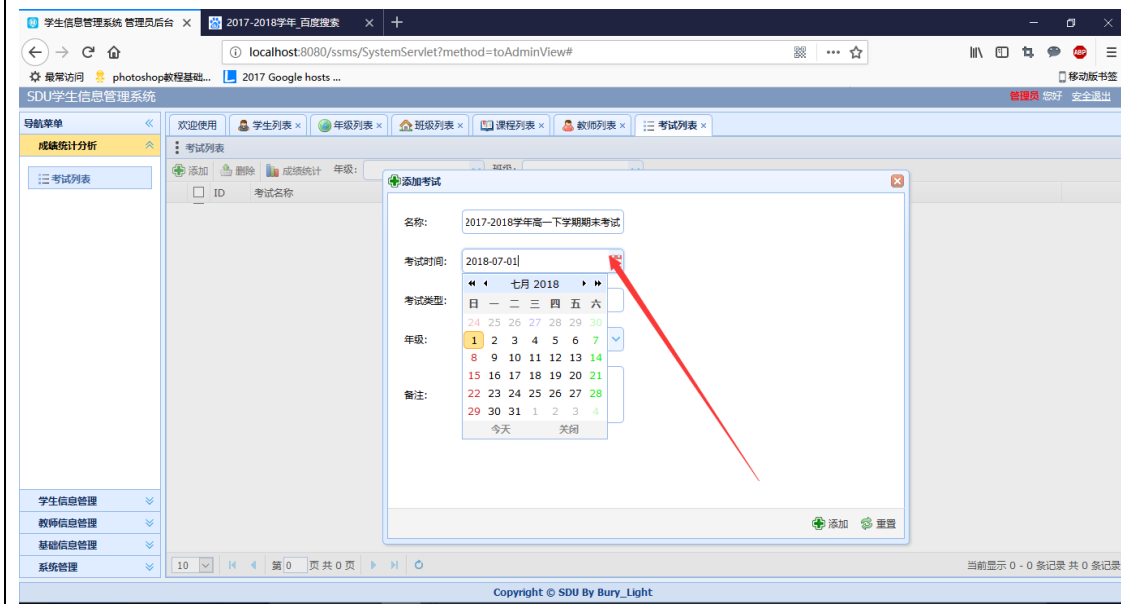
SELECT * FROM `teacher` LIMIT 0

第 1 条记录 (共 18 条) 于 1 页

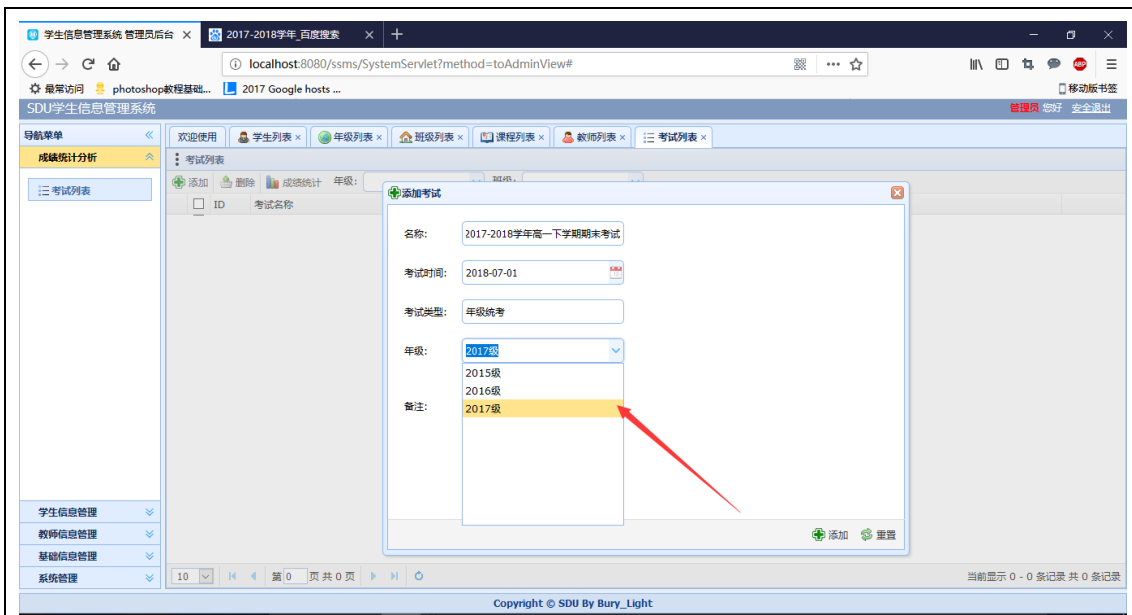
可以选择一致多条教师信息进行删除



(3) 成绩统计分析:
组织年级大考, 点击日历选择考试日期



选择考试年级



数据库中的考试表

exam @ssms (mysql80) - 表

文件 编辑 查看 窗口 帮助

导入向导 导出向导 筛选向导 网格查看 表单查看 备注 十六进制 图像 A-Z 升序排序

id	name	time	remark	type	gradeid	clazzid	courseid
1	2017-2018学年高一下学	2018-07-01		1	3	0	0

第 1 条记录 (共 1 条) 于 1 页

SELECT * FROM `exam` LIMIT 0, 1

根据条件查看学生成绩信息

SDU学生信息管理系统 管理员后台

2017-2018学年 百度搜索

localhost:8080/ssms/SystemServlet?method=toAdminView#

SDU学生信息管理系统

导航菜单

成绩统计分析

考试列表

成绩统计

导出 班级: 2班

学号	姓名	语文	数学	英语	总分
1 201701001	王光雨	0	0	0	0
2 201701002	王光月	0	0	0	0
3 201702001	王春林	0	0	0	0
4 201702002	王春雨	0	0	0	0
5 201703001	牛扬	0	0	0	0
6 201703002	刘扬	0	0	0	0
7 201704001	黄超	0	0	0	0
8 201704002	黄丹	0	0	0	0

显示 1 - 1 条记录 共 1 条记录

Copyright © SDU By Bury_Light

SDU学生信息管理系统 管理员后台

2017-2018学年 百度搜索

localhost:8080/ssms/SystemServlet?method=toAdminView#

SDU学生信息管理系统

导航菜单

成绩统计分析

考试列表

成绩统计

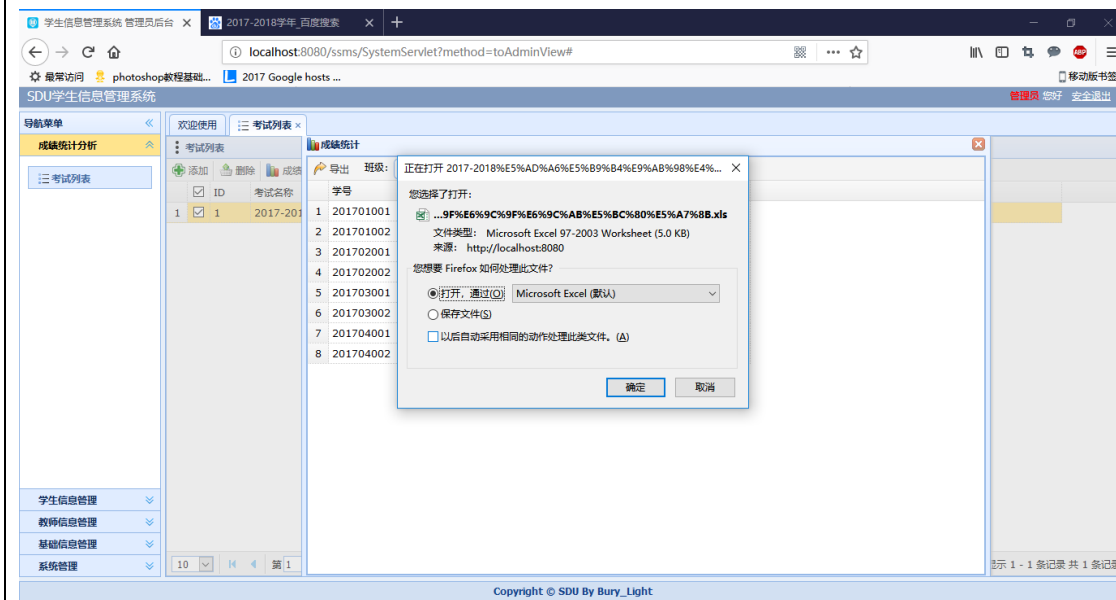
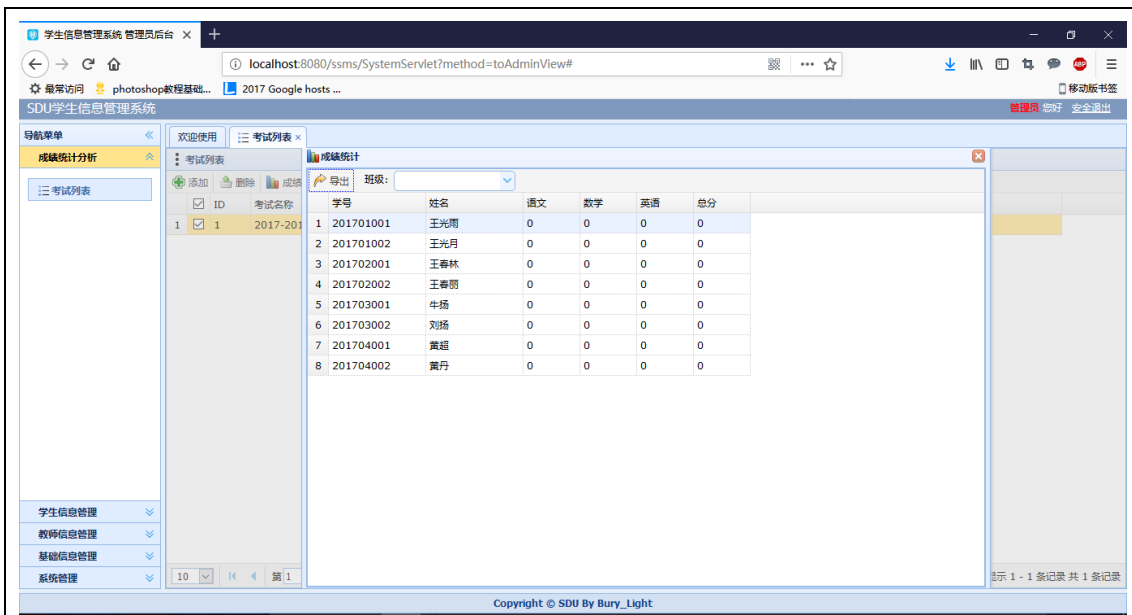
导出 班级: 2班

学号	姓名	语文	数学	英语	总分
1 201702001	王春林	0	0	0	0
2 201702002	王春雨	0	0	0	0

显示 1 - 1 条记录 共 1 条记录

Copyright © SDU By Bury_Light

将所要查看的学生成绩信息导出到 excel 表格中



可以看到我们需要数据已成功导出到 excel 表格中

2017-2018年E5%AD%A6%E5%B9%B4%E9%AB%98%E4%B8%80%E4%B8%BB%E5%AD%A6%E6%9C%9F%E6%9C%AB%E5%BC%80%E5%A7%88.xls [只...]

文件 开始 插入 页面布局 公式 数据 审阅 视图 告诉我你想要做什么 共享

剪贴板 格式刷 粘贴 格式刷 字体 对齐方式 数字 样式 单元格 编辑

A1 姓名

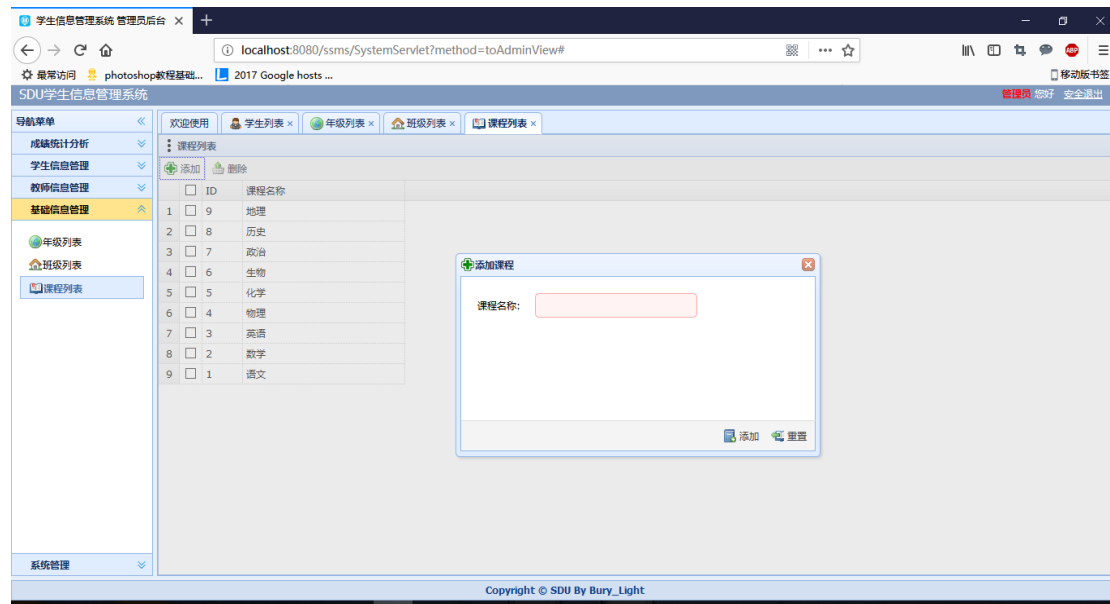
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	姓名	学号	语文	数学	英语	总分															
2	王光海	201701001	0	0	0	0															
3	王光月	201701002	0	0	0	0															
4	王春林	201702001	0	0	0	0															
5	王春丽	201702002	0	0	0	0															
6	牛扬	201703001	0	0	0	0															
7	刘坤	201703002	0	0	0	0															
8	黄超	201704001	0	0	0	0															
9	黄丹	201704002	0	0	0	0															
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19																					
20																					
21																					
22																					
23																					
24																					

Sheet0

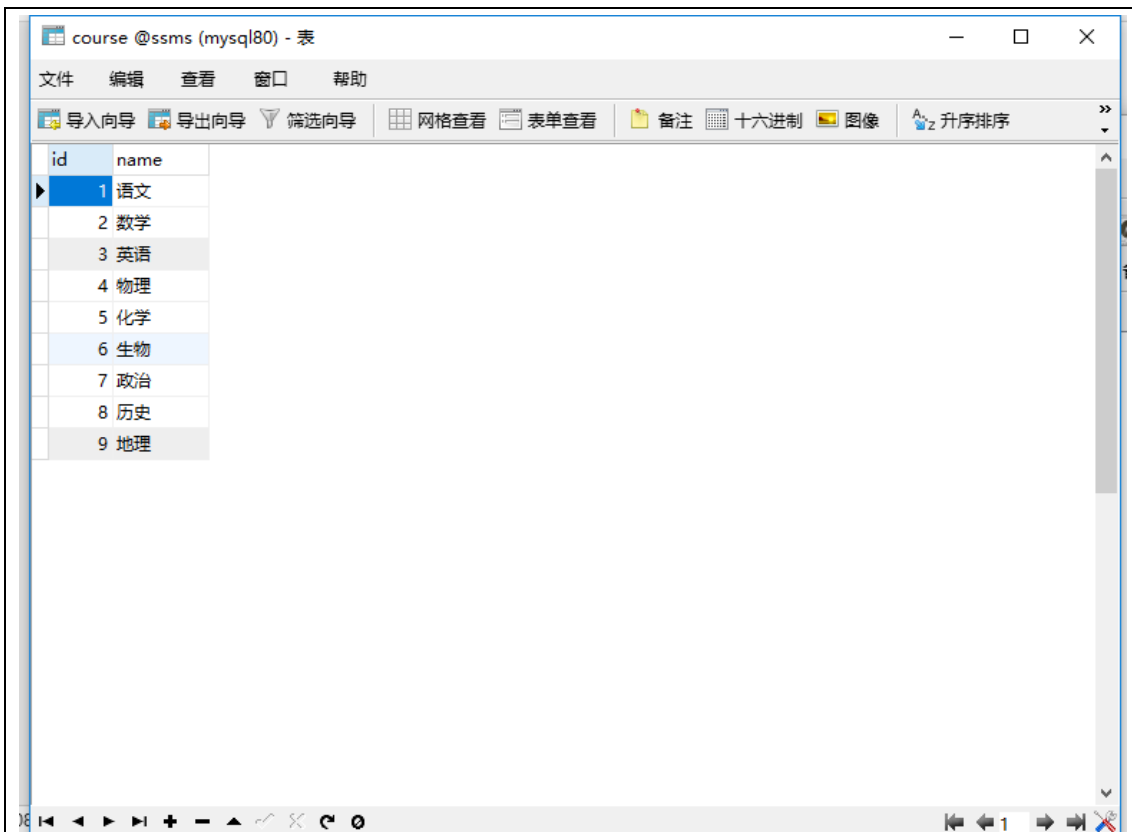
就绪

(4) 基础信息管理：

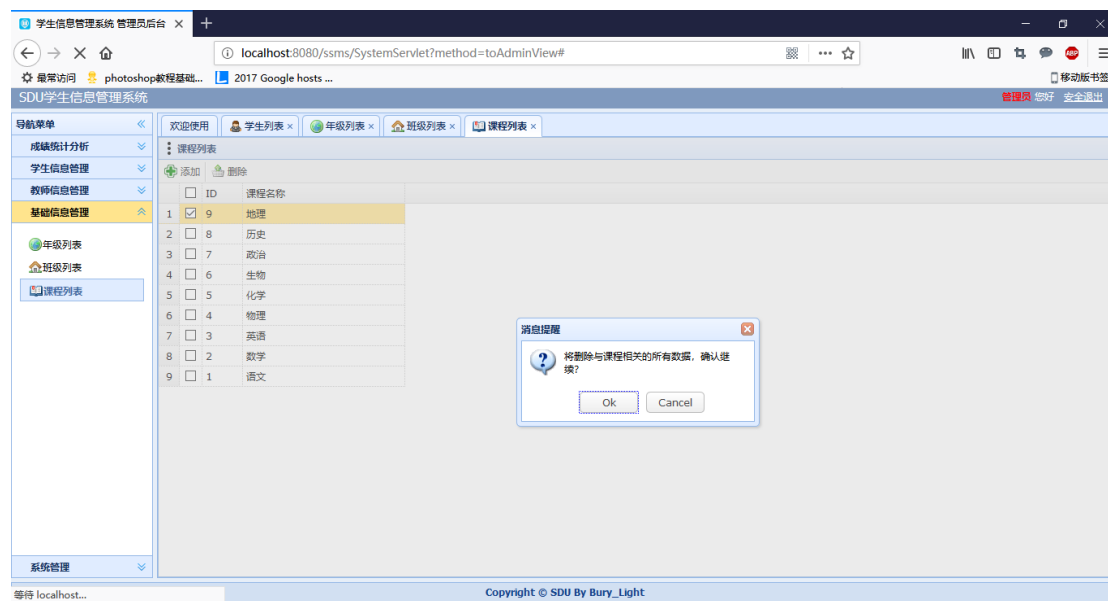
①添加、查看、删除课程信息



添加课程后的数据库课程表信息

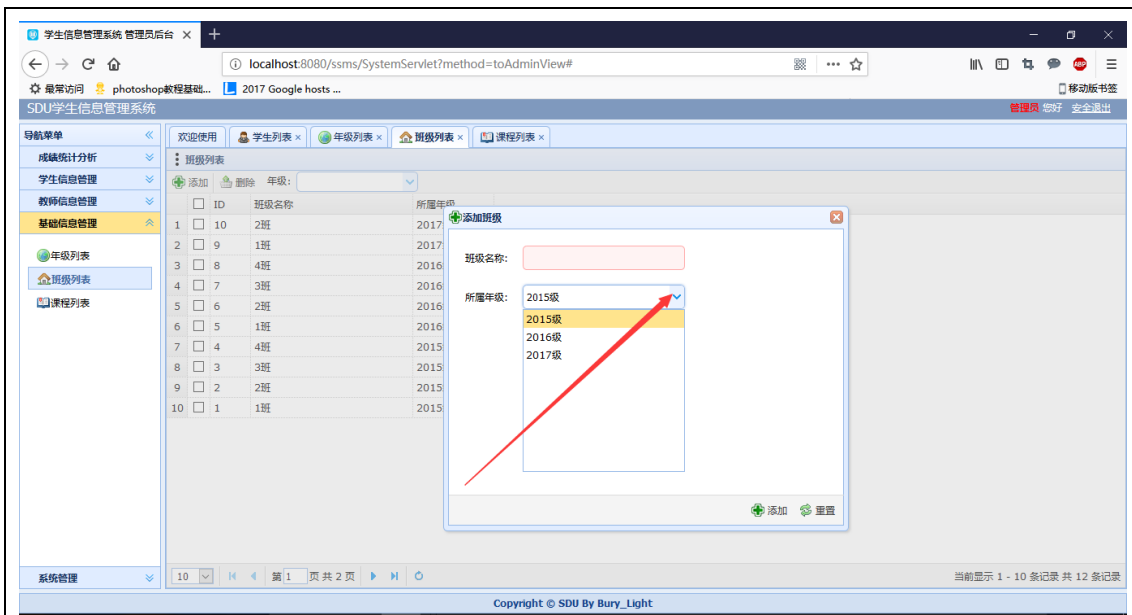


可以选择删除某门课程



②添加、查看、删除班级信息

添加班级，同样的班级名会有不同的年级选项框



数据库中的班级表

clazz @ssms (mysql80) - 表

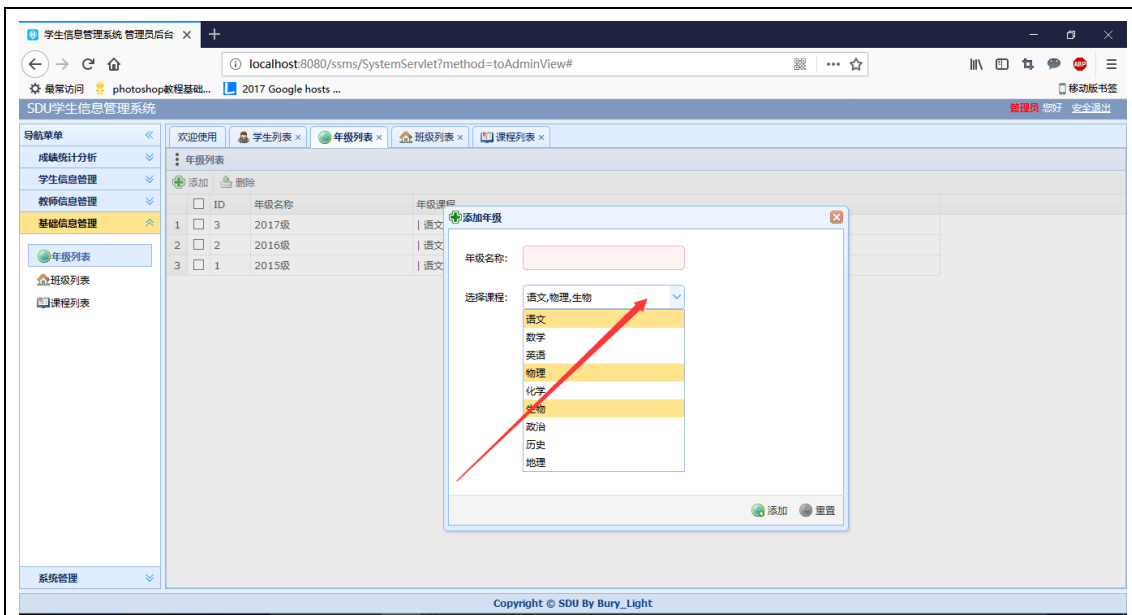
id	name	gradeid
1	1班	1
2	2班	1
3	3班	1
4	4班	1
5	1班	2
6	2班	2
7	3班	2
8	4班	2
9	1班	3
10	2班	3
11	3班	3
12	4班	3

SELECT * FROM `clazz` LIMIT 0, 11

第 1 条记录 (共 12 条) 于 1 页

③添加、查看、删除年级信息

添加年级信息，可以设置年级下所学的课程



数据库中的年级表

id	name
1	2015级
2	2016级
3	2017级

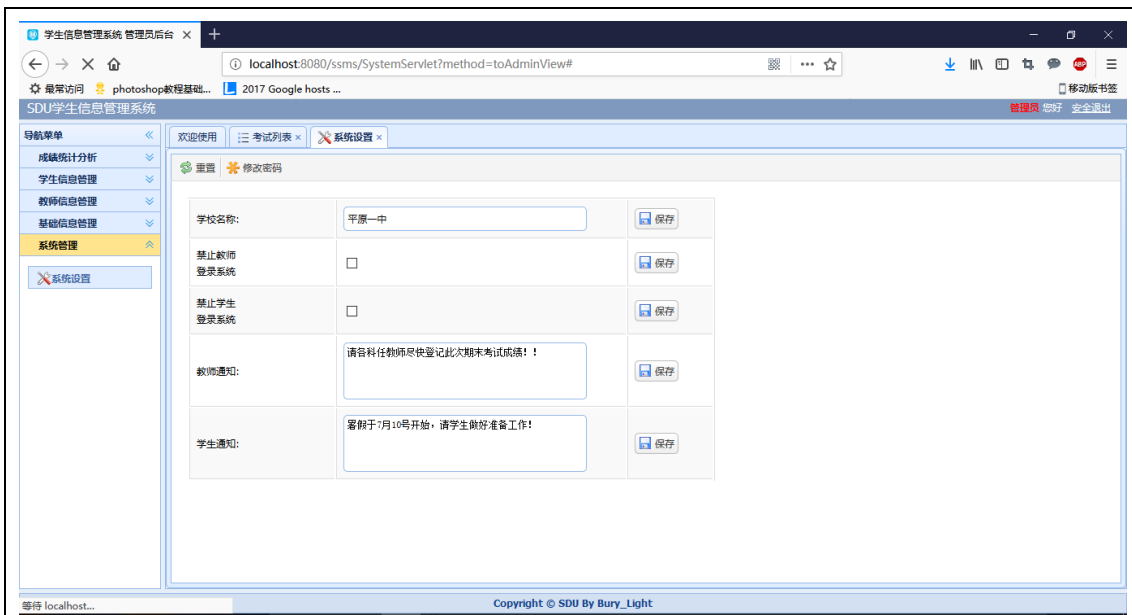
(5) 系统信息管理:

修改学校名称

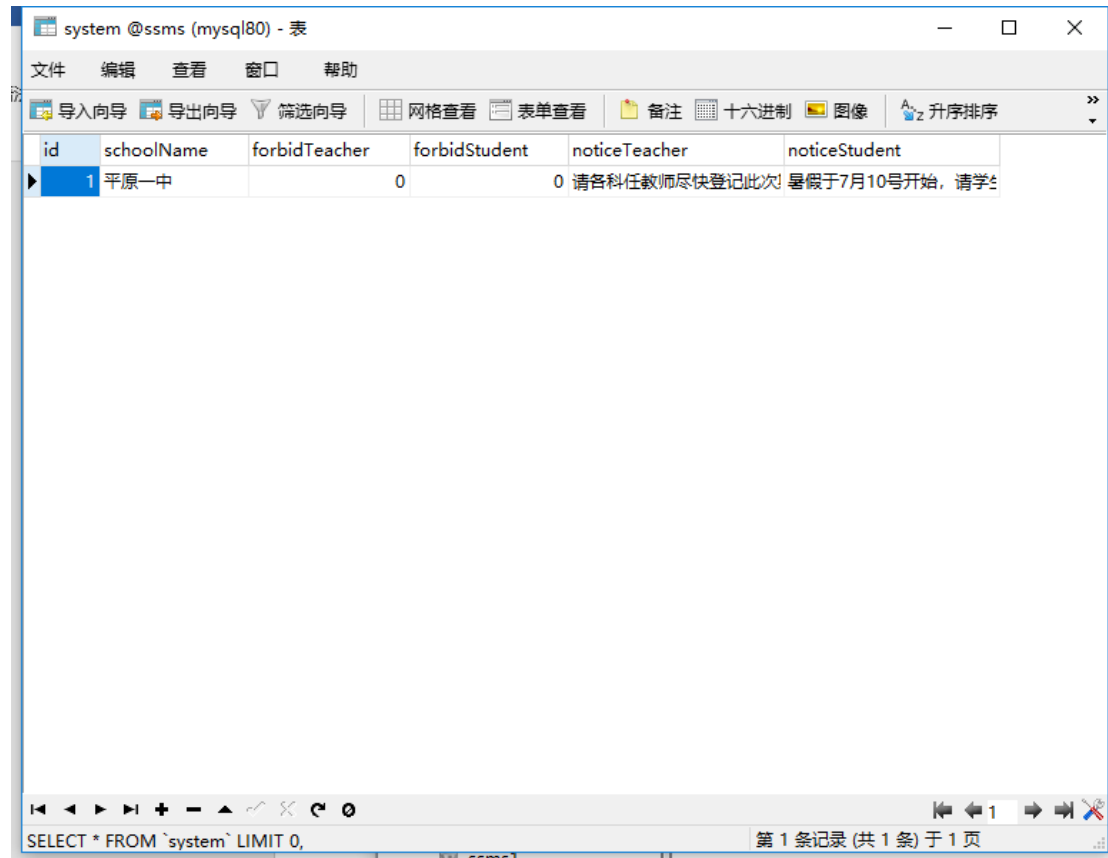
禁止学生或者教师登录系统

发布学生通知

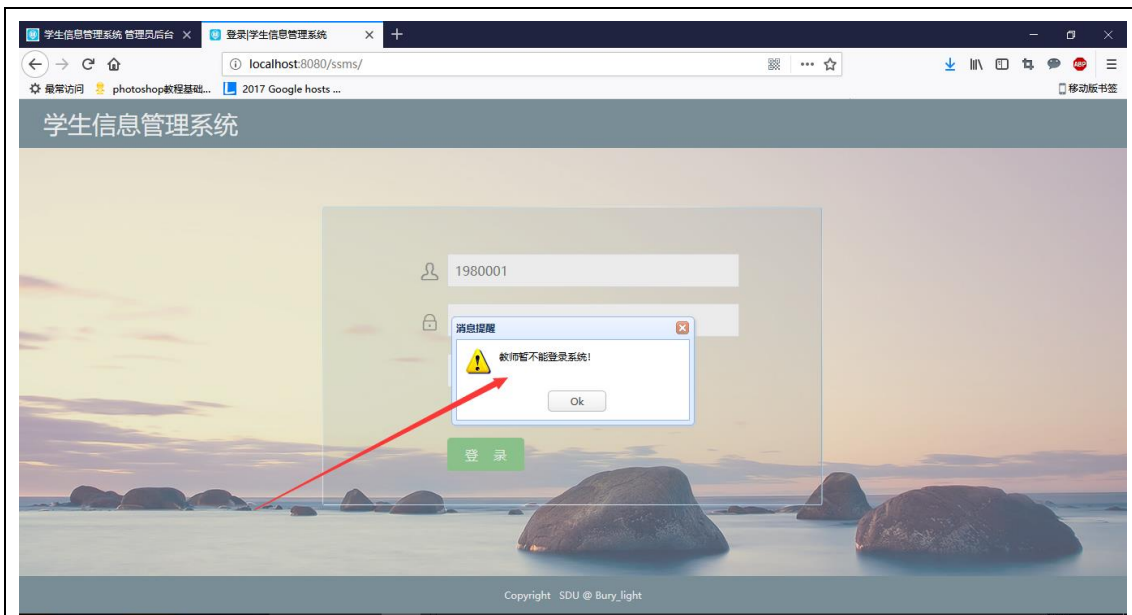
发布教师通知



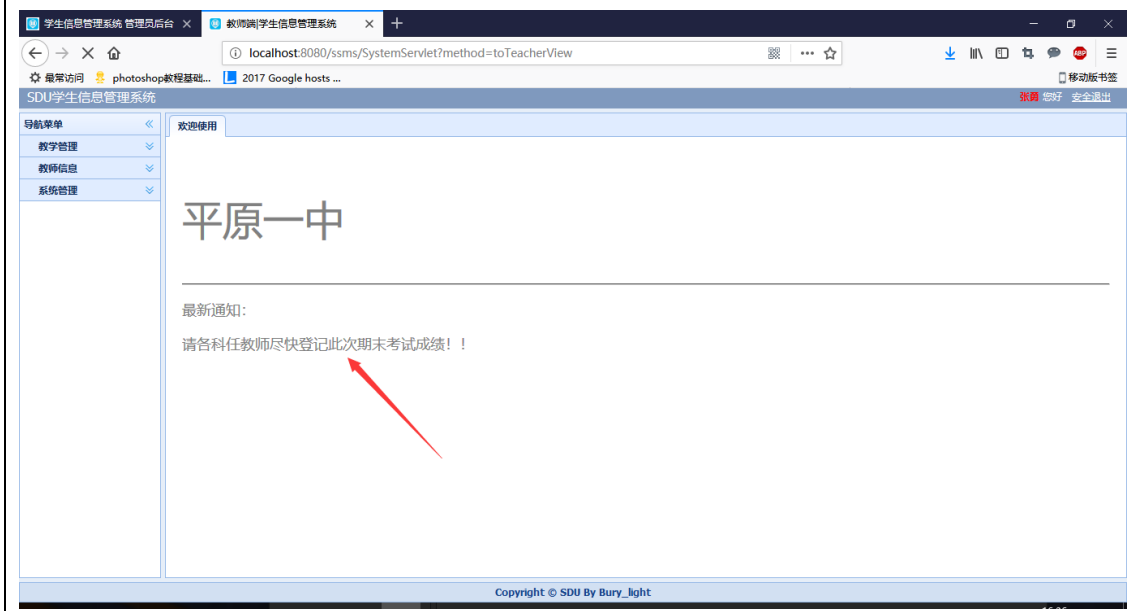
数据库中的系统信息表

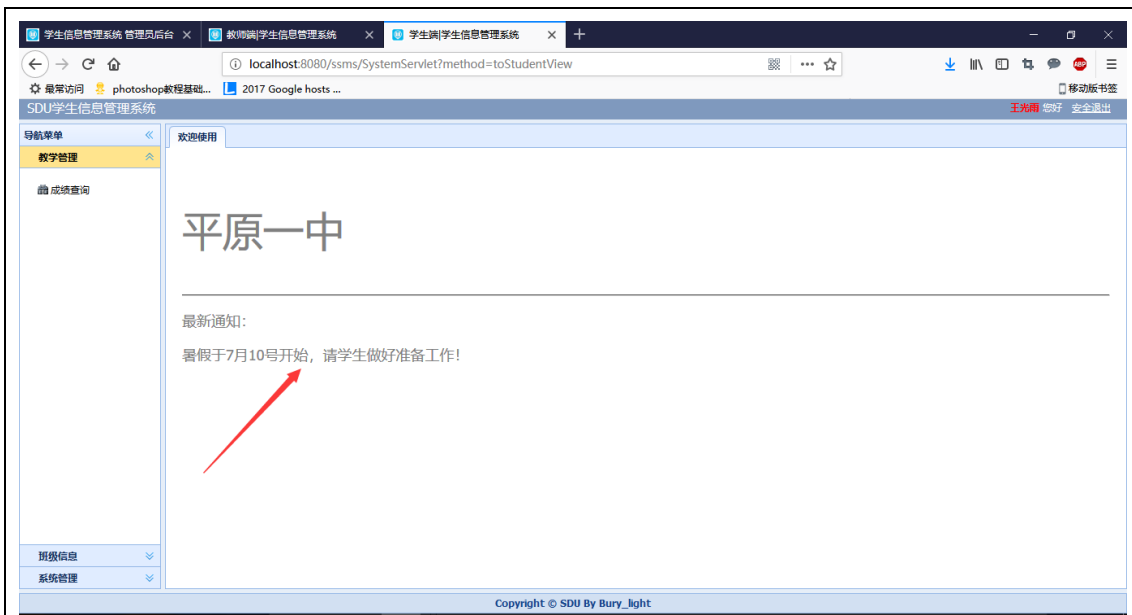


勾选禁止教师登陆选项后



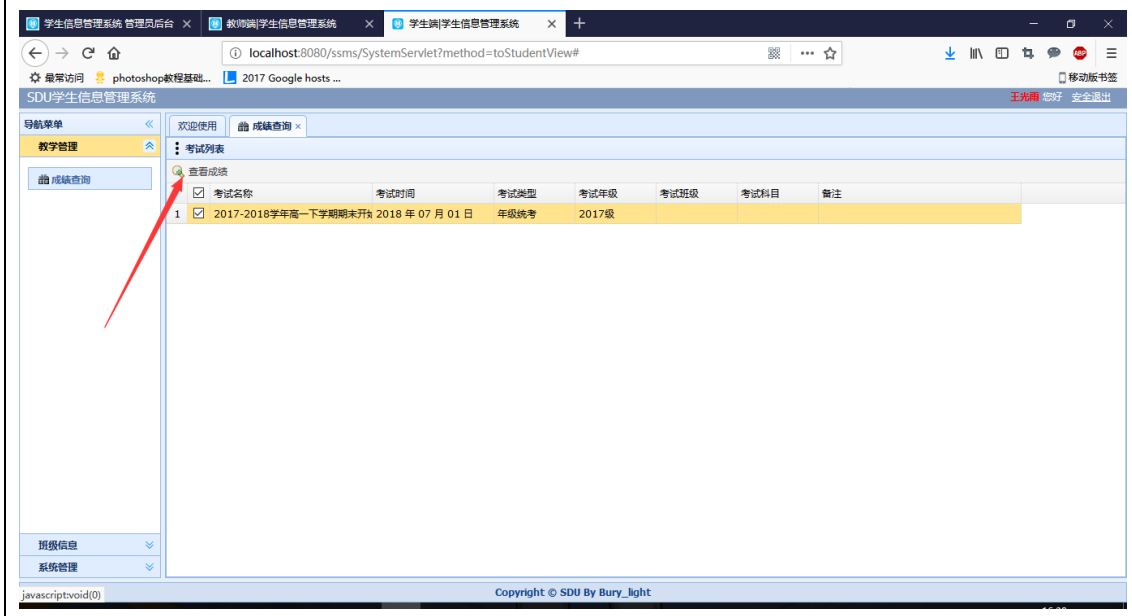
管理员在系统信息中发布的消息会显示在学生和教师的登录界面上

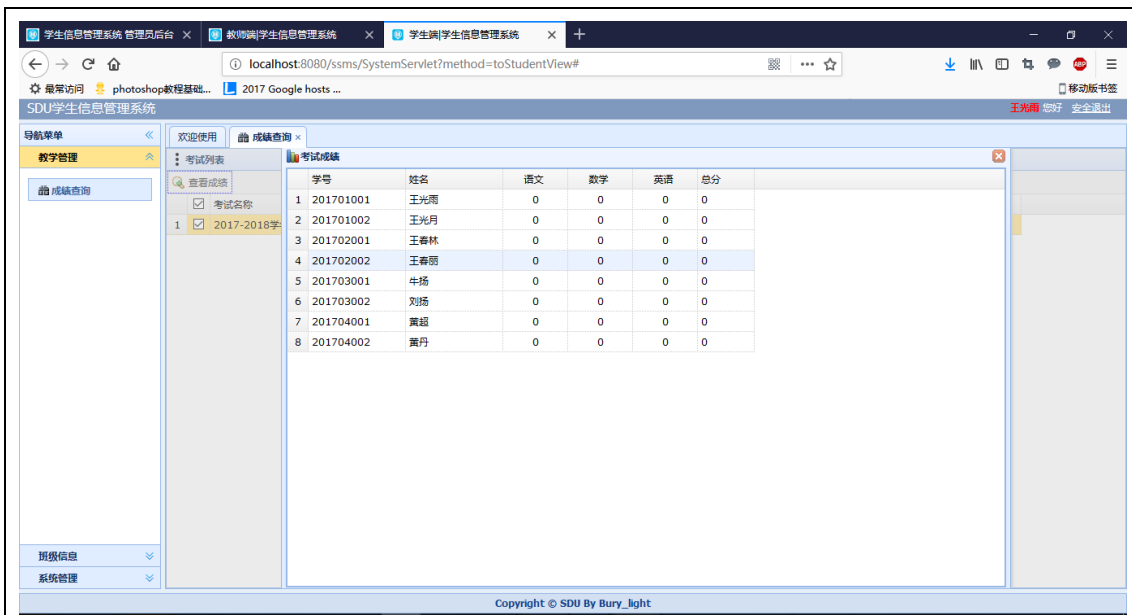




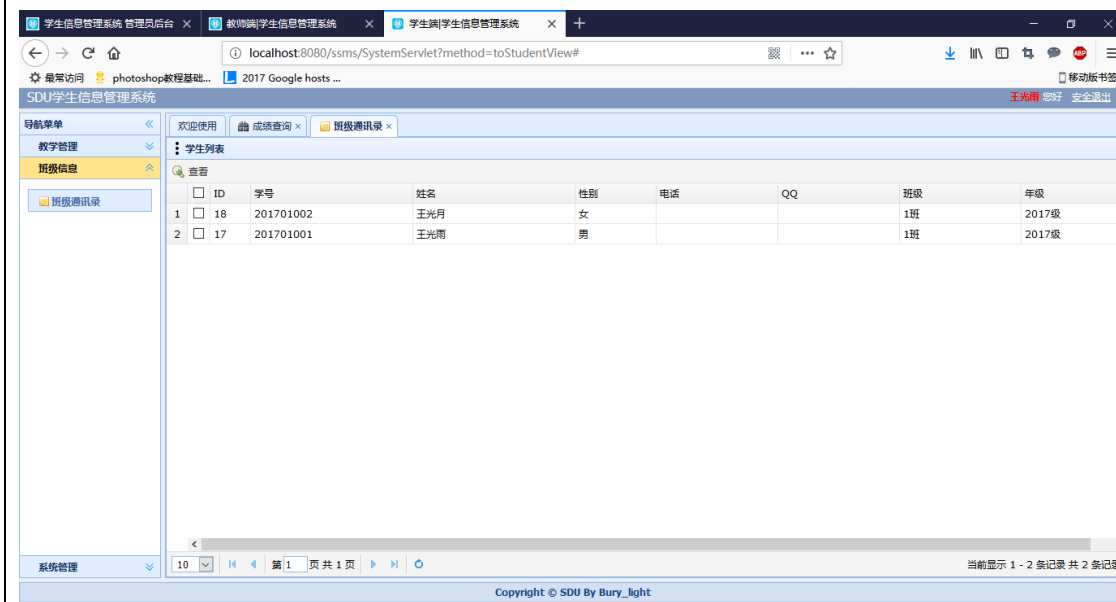
2. 学生登录

(1) 教学管理:
查看成绩

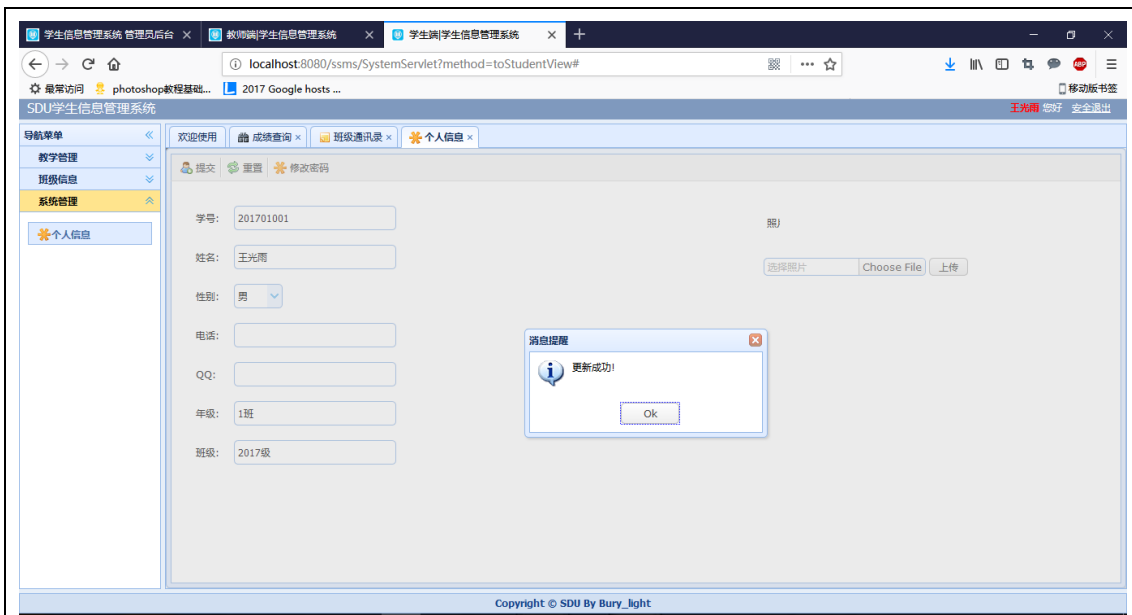




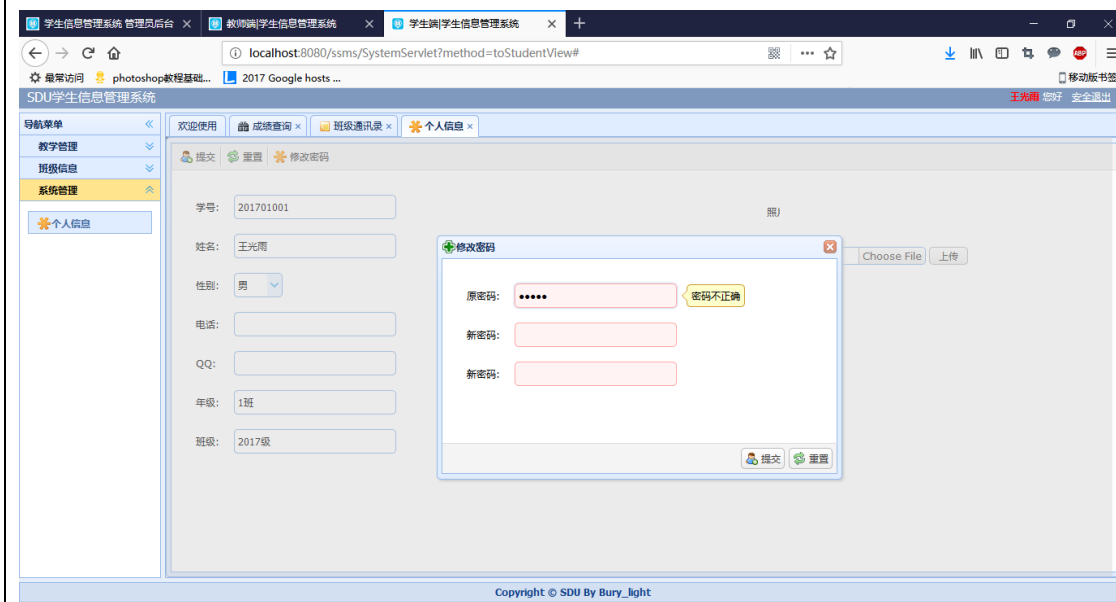
查看班级通讯录，可以查看本班学生信息



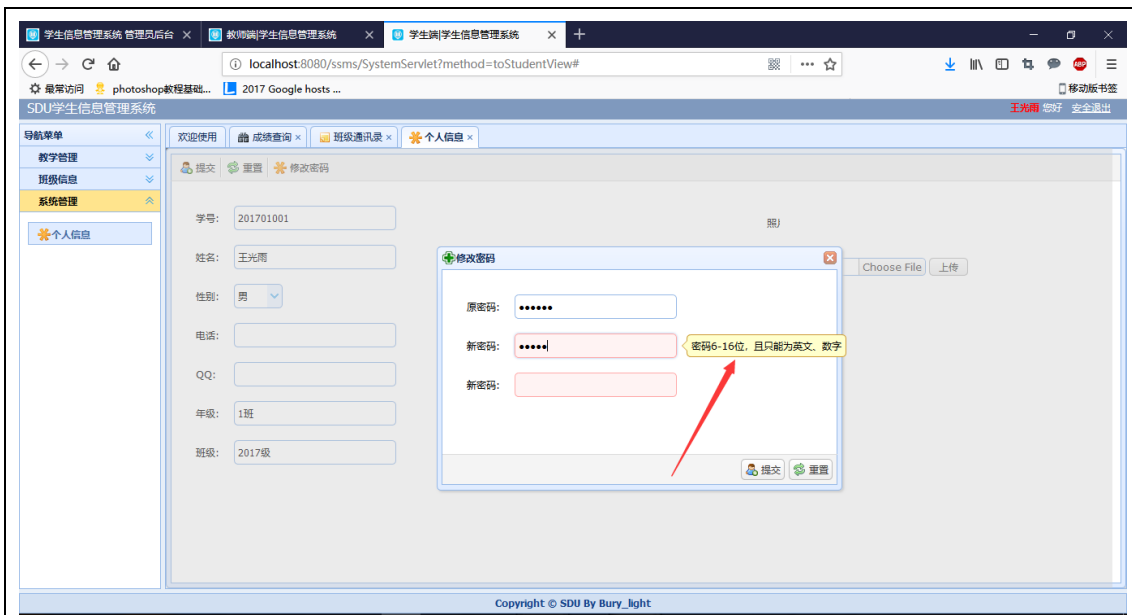
(2) 系统管理：
修改个人信息



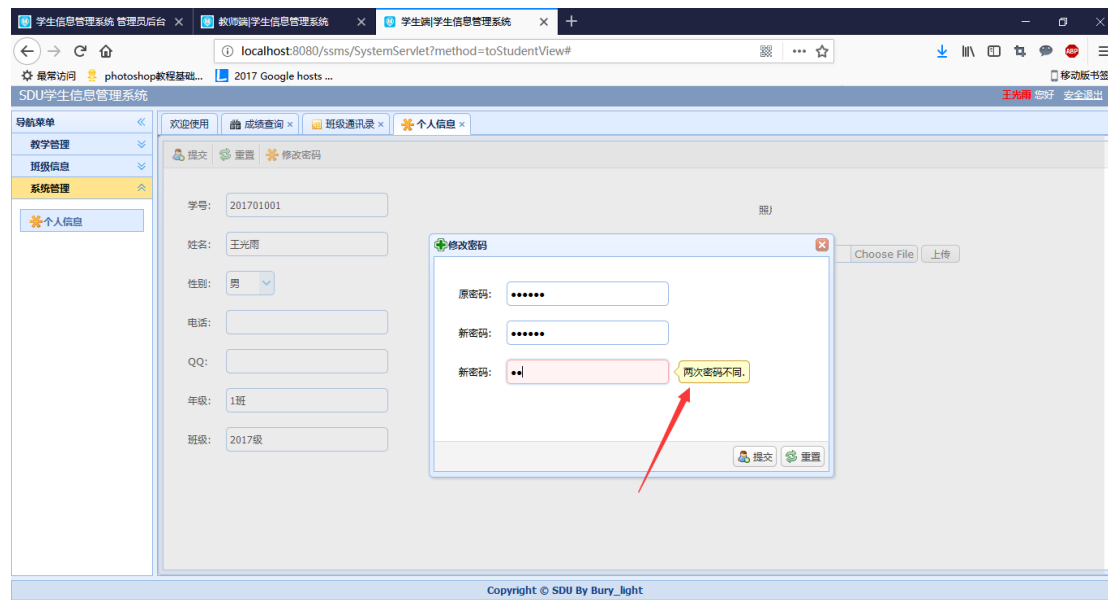
修改登录密码



对于新密码有设置的要求

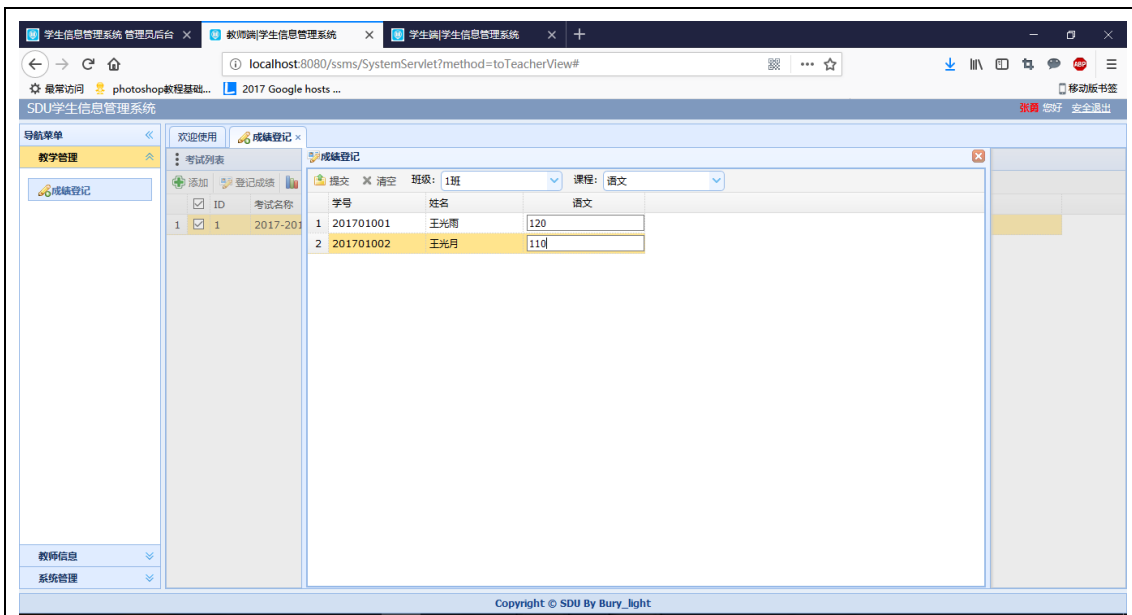


若两次新密码输入不一致会提示



3. 教师登录

(1) 教学管理:
录入学生成绩



可以看到成绩表被更新

escore @ssms (mysql80) - 表

文件 编辑 查看 窗口 帮助

导入向导 导出向导 筛选向导 网格查看 表单查看 备注 十六进制 图像 升序排序

id	examid	clazzid	studentid	gradeid	courseid	score
1	1	9	17	3	1	120
2	1	9	17	3	2	(Null)
3	1	9	17	3	3	(Null)
4	1	9	18	3	1	110
5	1	9	18	3	2	(Null)
6	1	9	18	3	3	(Null)
7	1	10	19	3	1	130
8	1	10	19	3	2	(Null)
9	1	10	19	3	3	(Null)
10	1	10	20	3	1	100
11	1	10	20	3	2	(Null)
12	1	10	20	3	3	(Null)
13	1	11	21	3	1	100
14	1	11	21	3	2	(Null)
15	1	11	21	3	3	(Null)
16	1	11	22	3	1	120
17	1	11	22	3	2	(Null)
18	1	11	22	3	3	(Null)
19	1	12	23	3	1	120

SELECT * FROM `escore` LIMIT 0, 第 1 条记录 (共 24 条) 于 1 页

查看与自己相关的学生的成绩

SDU学生信息管理系统

教师端学生信息管理系统

localhost:8080/ssms/SystemServlet?method=toTeacherView#

SDU学生信息管理系统

导航菜单

教学管理

成绩统计

学号	姓名	语文	数学	英语	总分
1 201701001	王光雨	120	0	0	120
2 201701002	王光月	110	0	0	110
3 201702001	王春林	130	0	0	130
4 201702002	王春丽	100	0	0	100
5 201703001	牛扬	100	0	0	100
6 201703002	刘扬	120	0	0	120
7 201704001	黄超	120	0	0	120
8 201704002	黄丹	130	0	0	130

Copyright © SDU By Bury_light

将学生成绩导出到 excel 表中

exam @ssms (mysql80) - 表

文件 编辑 查看 窗口 帮助

导入向导 导出向导 筛选向导 网格查看 表单查看 备注 十六进制 图像 A_z 升序排序

id	name	time	remark	type	gradeid	clazzid	courseid
1	2017-2018学年高一下学	2018-07-01		1	3	0	0
2	高一一班语文小测	2018-06-27		2	3	9	1

SELECT * FROM `exam` LIMIT 0, 1

第 1 条记录 (共 2 条) 于 1 页

查看教师通讯录

SDU学生信息管理系统

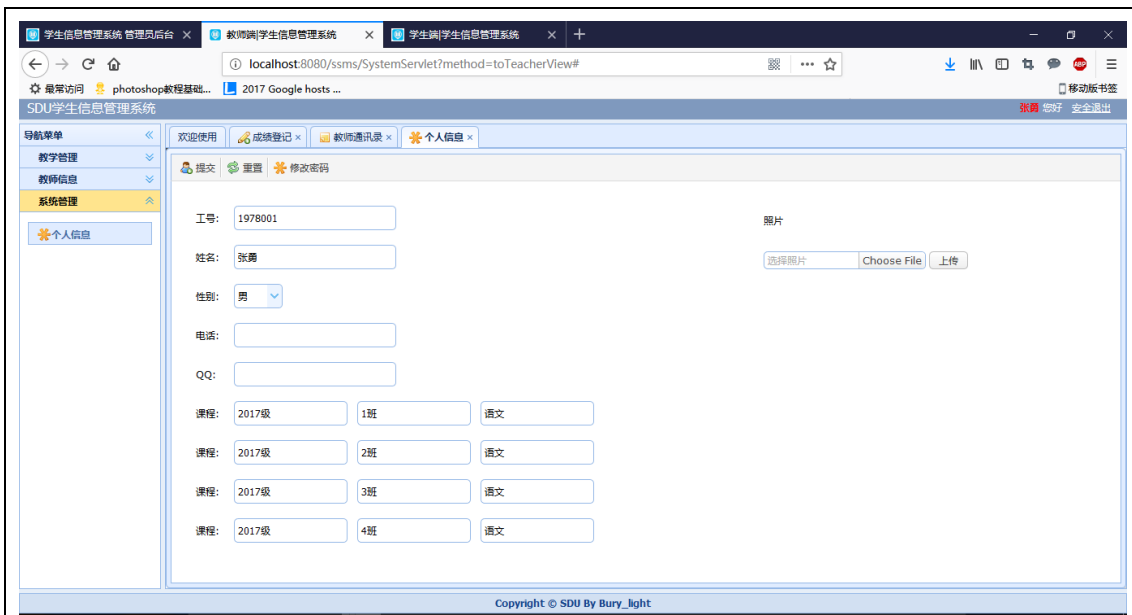
教师通讯录

ID	工号	姓名	性别	电话	QQ	课程
18	1980009	吴晓	女			[2015级 1班 地理] [2015级 2班 地理] [2015级 3班 地理] [2015级 1班 历史] [2015级 2班 历史] [2015级 3班 历史] [2015级 1班 政治] [2015级 2班 政治] [2015级 3班 政治] [2015级 1班 生物] [2015级 2班 生物] [2015级 3班 生物] [2015级 1班 化学] [2015级 2班 化学] [2015级 3班 化学] [2015级 1班 物理] [2015级 2班 物理] [2015级 3班 物理] [2015级 1班 英语] [2015级 2班 英语] [2015级 3班 英语] [2015级 1班 数学] [2015级 2班 数学] [2015级 3班 数学] [2015级 1班 语文] [2015级 2班 语文] [2015级 3班 语文] [2016级 1班 生物] [2016级 2班 生物] [2016级 3班 生物]
17	1980008	周晓	女			
16	1980007	李响	女			
15	1980006	赵晓	女			
14	1980005	冯晓	女			
13	1980004	黄晓	男			
12	1980003	孙晓	男			
11	1980002	刘响	男			
10	1980001	王响	女			
9	1979006	牛杰	女			

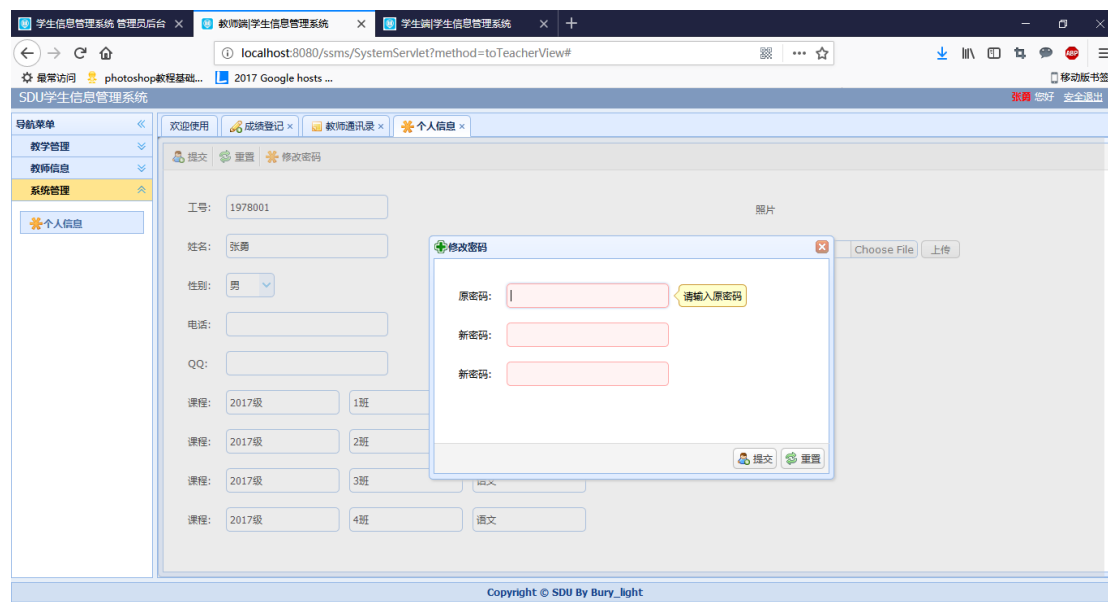
当前显示 1 - 10 条记录 共 18 条记录

Copyright © SDU By Bury_light

(2) 系统管理：
修改个人信息，只能修改部分信息



修改登录密码



实验过程中遇到和解决的问题：

（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

结论分析与体会：

系统优点：

1、系统设计全面、功能齐全。系统可以满足学生、教师和管理员不同身份的人登录，不同用户登录后能够完成不同的功能，设计十分清晰明快能够很好的满足实际操作需求。

- 2、因为使用的 B/S 架构，所以界面美观，软件的易用性高，操作简单，能够很好的满足需求，可以应用到某些中小学的信息管理中。
- 3、数据库设计内容具体详细，条理清晰，关系明确，能够遵循数据库设计的准则来描述信息关系，可以稳定地为系统提供服务。
- 4、信息提示系统细致完善，对于用户可能发生的错误操作，给予错误信息提示。比如用户在修改登录密码时，会有不同的提示。
- 5、对异常进行了处理，在用户操作不规范时给出错误信息，例如查看信息时选择了多条信息等。
6. 一定程度上保证系统的安全，用户再登录时需要输入验证码，防止了黑客等违法分子恶意破解密码等。

系统不足：

- 1、系统的稳定性不高，长时间的运行会导致程序的运行缓慢且影响软件的效率。
- 2、系统具有一定的并发能力，但是过高的并发操作会使得运行十分缓慢。
- 3、数据库的设计存在一些问题，可以进行进一步的优化，比如为了实现某些特定功能数据库的设计在一定程度上存在冗余。

收获：

通过一学期的实验，我学会了短时间内的系统开发，我认为培养这种能力是十分重要的，我认为在开发过程中应该全神贯注，珍惜时间，应该先把系统的主要功能认真实现，对细枝末节应该尽量完善，避免因为某些小错误而导致出现系统崩溃这种不良现象。此外，前期的准备过程也很重要，不要犯好高骛远的错误，应该在开始开发前打好基础，这也是我选择 MVC 模型开发的一个主要原因。

下面给出部分开发过程中的认知与感悟（开发过程就是一个不断学习不断提高的过程）：

①C3P0 是一个开源的 JDBC 连接池，它实现了数据源和 JNDI 绑定

②用户可以使用 JavaBean 将功能、处理、值、数据库访问和其他任何可以用 java 代码创造的对象进行打包，并且其他的开发者可以通过内部的 JSP 页面、Servlet、其他 JavaBean、applet 程序或者应用来使用这些对象。用户可以认为 JavaBean 提供了一种随时随地的复制和粘贴的功能，而不用关心任何改变。

JavaBean 可分为两种：一种是有用户界面（UI，User Interface）的 JavaBean；还有一种是没有用户界面，主要负责处理事务（如数据运算，操纵数据库）的 JavaBean。JSP 通常访问的是后一种 JavaBean。

③SSH 架构，代表了三层吧，model 是实例，service 提供各种方法，dao 实现各种方法，impl 指 implement，即实现，service 和 dao 都需要 impl，dao 内部的方法一般是实现与数据库交互的，而 model 一般就是数据库中表的实例

④HttpServlet 指能够处理 HTTP 请求的 servlet, 它在原有 Servlet 接口上添加了一些与 HTTP 协议处理方法, 它比 Servlet 接口的功能更为强大。因此开发人员在编写 Servlet 时, 通常应继承这个类, 而避免直接去实现 Servlet 接口。HttpServlet 在实现 Servlet 接口时, 覆写了 service 方法, 该方法体内的代码会自动判断用户的请求方式, 如为 GET 请求, 则调用 HttpServlet 的 doGet 方法, 如为 Post 请求, 则调用 doPost 方法。因此, 开发人员在编写 Servlet 时, 通常只需要覆写 doGet 或 doPost 方法, 而不要去覆写 service 方法。

附: 框架采用的是 EasyUi 和 H-ui, 整个项目使用 Eclipse 开发, 要运行的话只需要将其在 Eclipse 打开后部署到 tomcat 上即可(首先电脑上需要配置 tomcat, 自行谷歌即可), 整个项目包在文件夹下。