**实验五 综合训练**

**计算机222 章崇文 202202296**

一、实验目的

1．了解MVC思想。

2．熟悉Java Web的各种开发技术。

二、实验学时

4H

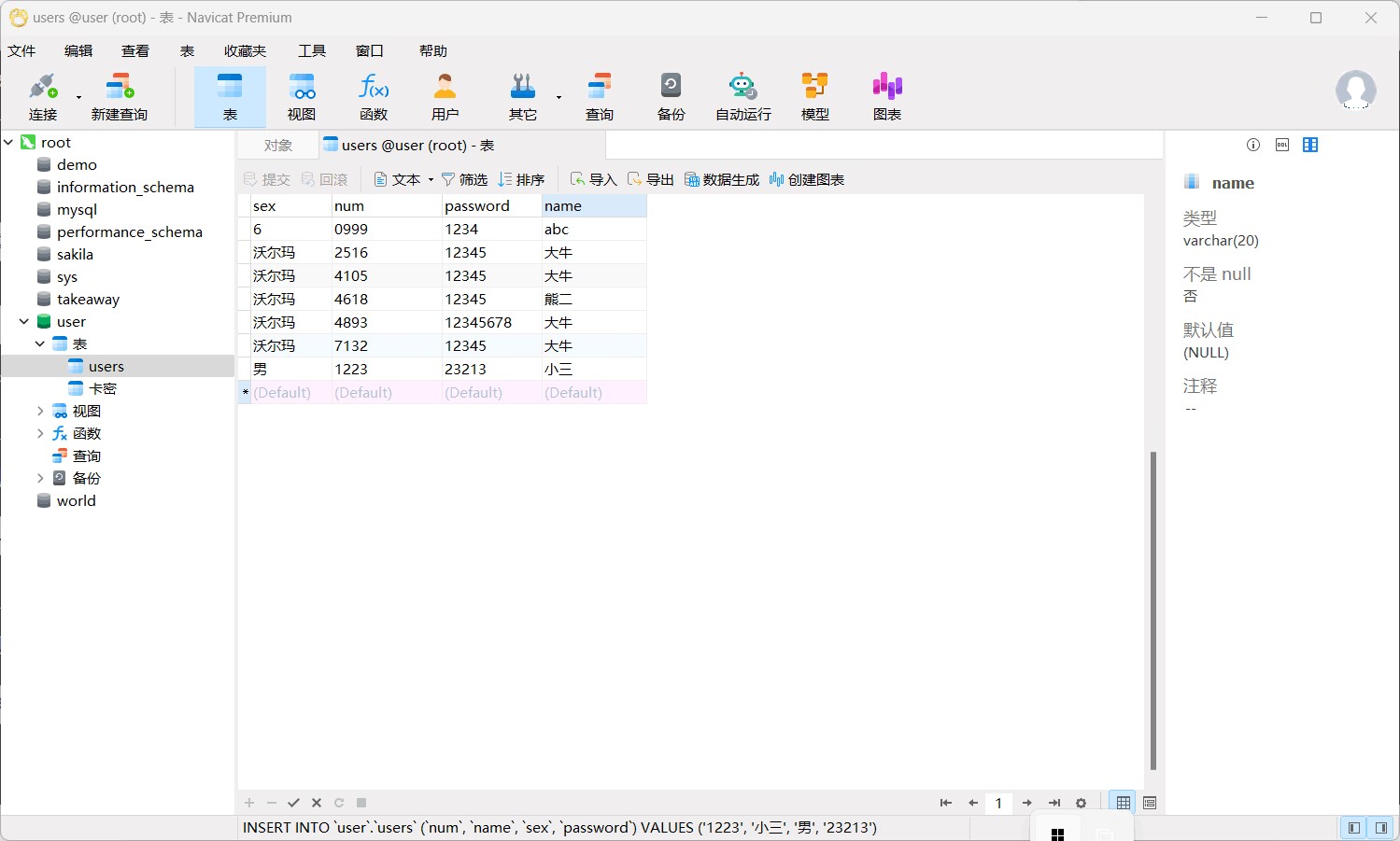
三、实验性质

综合性实验

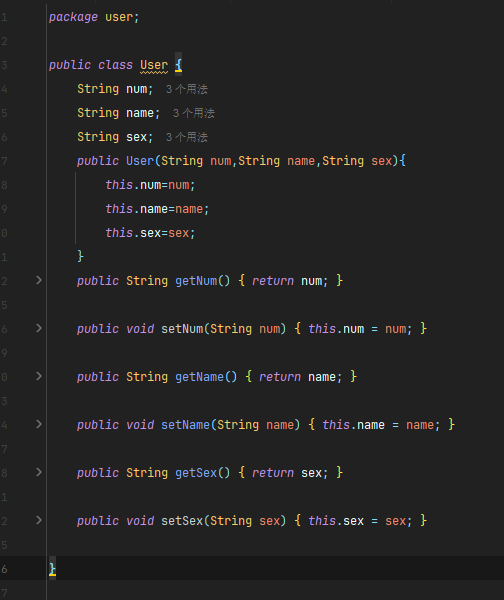
四、实验内容

1．创建表t\_student，其中包含学号stuno，姓名stuname，性别stusex三个字段，在表中插入一些测试数据。（5分）

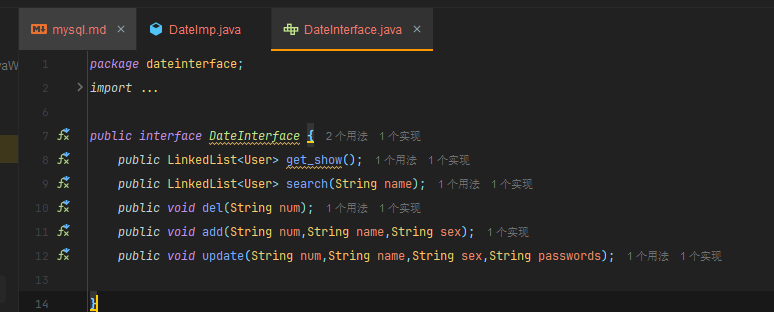




如图创建user表以及对应的字段，插入一些测试数据

、

定义的user类



定义了一个名为DateInterface的Java接口，它用于规范一系列数据操作的方法，这些方法主要用于用户信息管理的后台逻辑，比如获取、搜索、删除、添加和更新用户信息。

2．实现下述功能：

* 编写显示所有学生资料信息页面，其中包含一个表单用来根据学生姓名模糊查询匹配的学生信息。（50分）
* 在学生信息后面增加一个“删除学生信息”链接，单击链接可以将学生信息从数据库中删除，然后跳转到显示所有学生资料信息页面。（25分）



这段代码定义了一个名为Service的Java接口，它也定义了一组方法，用于描述与用户信息管理相关的服务操作。

图1 显示所有学生信息



图2 查询姓名为大的学生

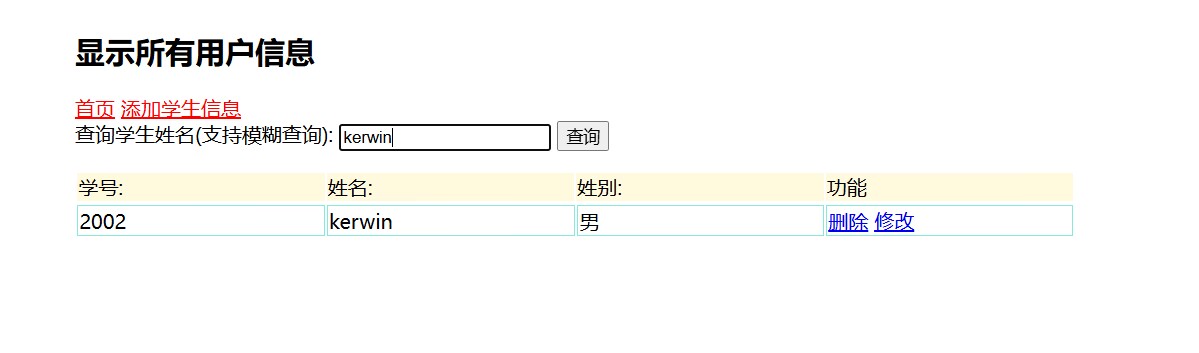


图3查询姓名为kerwin的学生（全查）

3．提高题： 尝试实现学生信息的添加和修改功能。（10分）

**添加函数和修改函数**

*public void* update (*HttpServletRequest* request,*HttpServletResponse* response) *throws* ServletException{  
 String num = request.getParameter("num");  
 String name = request.getParameter("name");  
 String sex = request.getParameter("sex");  
 String passwords = request.getParameter("password");  
 *new* service\_imp().update(num,name,sex,passwords);  
 *try* {  
 response.sendRedirect("../index.jsp");  
 } *catch* (IOException e) {  
 *throw new* RuntimeException(e);  
 }  
}  
  
*public void* adds (*HttpServletRequest* request,*HttpServletResponse* response) *throws* ServletException{  
 String num = request.getParameter("num");  
 String name = request.getParameter("name");  
 String sex = request.getParameter("sex");  
 *new* service\_imp().adds(num,name,sex);  
 *try* {  
 response.sendRedirect("../index.jsp");  
 } *catch* (IOException e) {  
 *throw new* RuntimeException(e);  
 }  
}



图4 添加学生界面



图5 添加学生界面结果



图6 添加学生界面结果

<%

response.setHeader("Pragma","N-cache");

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

response.addDateHeader("Expires",0);

int width=70;

int height=20;

BufferedImage image = new BufferedImage(width,height,BufferedImage.TYPE\_INT\_BGR);

Graphics g = image.getGraphics();

g.setColor(new Color(200,200,200));

g.fillRect(0,0,width,height);

Random rnd = new Random();

int randNum = rnd.nextInt(8999)+1000;

String ranStr = String.valueOf(randNum);

session.setAttribute("ranStr",ranStr);

g.setColor(new Color(rnd.nextInt(255),rnd.nextInt(255),rnd.nextInt(255)));

g.setFont(new Font("",Font.PLAIN,20));

g.drawString(ranStr,8,17);

for(int i=0;i<100;i++){

g.setColor(new Color(rnd.nextInt(255),rnd.nextInt(255),rnd.nextInt(255)));

int x=rnd.nextInt(width);

int y=rnd.nextInt(height);

g.drawOval(x,y,1,1);

}

ImageIO.write(image,"JPEG",response.getOutputStream());

out.clear();

out = pageContext.pushBody();

%>

这段代码用于生成并输出一个包含随机数字的验证码图片，代码的功能是在服务器端动态生成一个包含随机数字的验证码图片，并通过HTTP响应发送给客户端，同时将验证码的值保存在用户的session中，以便后续验证之用。

五、实验总结

请书写你对本次实验有哪些实质性的收获和体会，以及对本次实验有何良好的建议？（10分）

通过本次实验，我深刻理解并实践了MVC（Model-View-Controller）设计模式在Java Web开发中的应用。这一模式将应用程序的数据模型、用户界面和控制逻辑分离，提高了代码的可维护性和扩展性。具体来说，我在实验中：

1. **加深了对MVC架构的理解**：在实现学生信息管理系统的各个功能模块时，我明确划分了模型（Model，负责处理业务逻辑和数据访问）、视图（View，负责数据显示）和控制器（Controller，负责接收请求并调用模型处理数据），这种清晰的分层让我对如何组织代码有了更深入的认识。
2. **熟练掌握了Java Web开发技术**：通过使用Servlet、JSP、JDBC等技术，我不仅复习了Java Web开发的基础知识，还学习了如何将它们整合起来构建完整的Web应用。特别是对于JDBC的操作，我学会了如何创建数据库连接、执行SQL语句及处理结果集，这对于数据操作至关重要。
3. **提升了问题解决能力**：在实现学生信息的增删改查功能过程中，遇到了诸如SQL注入防范、页面跳转逻辑处理、数据验证等问题，通过查阅资料、调试代码，我逐步解决了这些问题，这个过程极大地锻炼了我的问题分析与解决能力。
4. **实践了前端与后端交互**：通过表单提交、超链接点击事件与后台Servlet的交互，我理解了Web应用前后端数据传递的基本流程，这对我今后进行复杂的Web应用开发大有裨益。

对于本次实验的建议，我认为：

* **增强安全性**：实验中可以进一步引入预编译SQL语句来防止SQL注入，以及加入更多的数据验证机制，如前端JavaScript验证和后端服务器验证，以提升系统的安全性。
* **优化用户体验**：在学生信息的增删改查功能上，可以增加更多的用户交互提示，比如操作成功或失败的即时反馈，以及更加友好的错误消息提示，从而提升用户的使用体验。
* **引入框架学习**：虽然手动编码有助于理解底层原理，但未来实验中可以考虑引入Spring MVC、MyBatis等框架，让学生体验企业级开发工具的便捷，同时理解框架背后的原理。