

B/S课程设计

[文档副标题]



2017-4-25

软件1402班吕雪杰

学号：2014234060301

目录

[**1、** **功能分析** 2](#_Toc481141728)

[**2、** **数据库设计** 2](#_Toc481141729)

[**3、** **架构设计** 3](#_Toc481141730)

[**3.1总体设计：** 3](#_Toc481141731)

[**3.2目录说明** 3](#_Toc481141732)

[**3.3代码简单说明**： 4](#_Toc481141733)

[3.3.1model层 4](#_Toc481141734)

[3.3.2Controller层 6](#_Toc481141735)

[3.3.3view层 7](#_Toc481141736)

[**4、** **应该改进的地方：** 9](#_Toc481141737)

[4.1数据通信问题： 9](#_Toc481141738)

[4.2结构问题： 9](#_Toc481141740)

[**5、** **遇到的问题和解决的方法** 9](#_Toc481141742)

[5.1多表查询问题 9](#_Toc481141743)

[5.2json格式问题 10](#_Toc481141744)

[**6、** **心得和体会** 10](#_Toc481141745)

1. **功能分析**

程序角色有两个：学生和老师（管理员）

学生功能：查看自己的课程成绩，选课

老师功能：对于学生信息表和课程信息表的维护（增删改查），对于课程进行成绩评分

1. **数据库设计**

**2.1系统E-R图**

**2.2表结构**

**user表结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 特殊说明 | 备注 |
| Id | Int | 11 | 主键，自增 | 学生id |
| Name | Char | 32 | 独一无二 | 学生姓名 |
| Password | Char | 32 |  | 学生密码 |

**Course表结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 特殊说明 | 备注 |
| Id | Int | 11 | 主键 | 课程ID |
| Name | Char | 32 |  | 课程名称 |
| Xueshi | Int | 11 |  | 课程学时 |
| Xuefen | Double |  |  | 课程学分 |
| Des | Char | 150 |  | 课程简介 |

**Xuanke表结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 特殊说明 | 备注 |
| Course\_id | Int | 11 | 外键，参考course（id） | 课程id |
| User\_id | Int | 11 | 外键，参考user（id） | 学生id |
| Score | Double |  |  | 课程成绩 |

1. **架构设计**

**3.1总体设计：**

系统总体采用mvc模式进行设计编码，将系统抽象成为三个层次：Model（模型层），View（视图层），Controller（控制层）

Model层：负责提取数据库的相关信息

View层：负责向用户提供页面视图

Contoller层：负责进行数据逻辑的处理，连接model层和view层

**3.2目录说明**

3.2.1Admin目录：存放关于管理员的所有界面

changeCourse.jsp 维护课程表页面

changestu.jsp 维护学生表页面

changxuan.jsp 给学生成绩页面

index.jsp 管理员功能界面

3.2.2Front目录：前台学生的所有界面

Elective.jsp 学生选课界面

View.jsp 学生查看课程成绩界面

Index.jsp 学生功能主界面

3.2.3Public目录：存放一些公共的资源

Css目录存放css文件

Js目录存放js文件

3.2.4Util目录：存放一些工具文件

checkUserSession.jsp 检查用户session的文件（每个前台用户页面都要引入，防止恶意后台地址输入）

3.2.5根目录：index.jsp学生登录界面

Admin.jsp 管理员登录界面

**3.3代码简单说明**：

采用javabean+servlet的形式，项目有两个servlet，userservlet和adminservlet分别负责学生和老师的逻辑处理操作

3.3.1model层

DBConnect.java数据库连接文件 使用单例模式连接数据，关键代码如下

public static Connection getConnection(){

//If instance has not been created yet, create it

if(DBConnect.connection == null){

initConnection();

}

return DBConnect.connection;

}

//Gets JDBC connection instance

private static void initConnection(){

try{

Class.forName("com.mysql.jdbc.Driver");

System.out.print("成功加载驱动");

String url = "jdbc:mysql://localhost:3306/class?useUnicode=true&characterEncoding=utf8";

String user = "root";

String pw = "123";

DBConnect.connection =

DriverManager.getConnection(url, user, pw);

System.out.print("成功连接到数据库");

}

catch (ClassNotFoundException e){

System.out.println(e.getMessage());

System.exit(0);

}

catch (SQLException e){

System.out.println(e.getMessage());

System.exit(0);

}

catch (Exception e){

}

}

Model层的其他文件，一个类对应着一个数据表，如MUser.java对应user数据表，里面写有关于user表的所有操作，MUser里通过DBConnect连接数据库，然后进行数据库的相关操作，其中一个方法如下

Connection connection = DBConnect.getConnection();：

public String GetUserPassword(String name){

String password="";

String sql="select password from user where name='"+name+"'";

System.out.println("sql语句是"+sql);

try{

Statement stmt = connection.createStatement();

rs=stmt.executeQuery(sql);

while(rs.next()){

password=rs.getString(1);

}

}catch(SQLException e){

e.printStackTrace();

}

return password;

}

其他如MXuankejava,MCourse.java逻辑如上。

3.3.2Controller层

通过功能分析我们知道项目有两个角色，学生与老师，所以控制层有两个主要类CAdmin和CUser分别处理关于老师和学生的逻辑操作，如CUser，学生可以选课，查询成绩，那么CUser类里就有Xuanke(int user\_id,int course\_id)方法和ShowAllScore(int id)方法，CAdmin类的方法按照老师的功能进行编写。另外控制层还有adminservlet和userservlet两个servlet分别处理前台老师和学生的数据请求，我们抽取一处代码进行展示

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=UTF-8"); //这行放在流输出前才好使

HttpSession session=request.getSession();

PrintWriter out = response.getWriter();

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

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

String res[][];

String type=request.getParameter("sub");

CAdmin cAdmin=new CAdmin();

//如果是登录

if(type.equals("登录")){

if(cAdmin.CheckUser(name,password) == true){

session.setAttribute("admin",name);

out.println("您是管理员 您已经登录成功 3秒后跳入后台页面")

response.setHeader("Refresh","5;URL=/XuankeNew/Admin/index.jsp");

}else{

out.println("您登录失败 3秒后返回主页面");

response.setHeader("Refresh","5;URL=/XuankeNew/admin.jsp");

}

}

3.3.3view层

提供给用户看的视图层，运用前台的html，css，js，jquery技术给用户良好的体验。

良好体验1：修改用户时运用jquery技术弹出对话窗并将要修改的原始数据填上，方便用户进行修改。创建用户和修改成绩也是如此。



良好体验2：在查询用户和老师评分页面，为减少页面刷新次数，采用ajax进行异步刷新，并且修改成绩页面，提供了每种课程的分别改成绩和所有课程的修改成绩，方便老师修改



1. **应该改进的地方：**

## 4.1数据通信问题：

本项目前台与后台之间通信的数据的类型是二维数组，但更好地解决方法是返回相应的对象，对象具有更高的可塑性，想取什么数据可根据需要去修改，但是二维数组需要记住下标对应的键值含义，小型项目还可以，但项目越大二维数组就费力了。

## 4.2结构问题：

应该是model层写的每个表的对象，dao层写的是对每个表操作的类，即dao层相等于本项目目前的model层。

1. **遇到的问题和解决的方法**

## 5.1多表查询问题

我们model层执行一个理念，一个表对应一个类，但是前台用户需要的数据可能需要多表查询，那么我们将这个多表操作放在哪里，解决方法：放到任意多表其中任意一个类里。

## 5.2json格式问题

修改成绩页面，之前用jsp代替servlet时ajax没有任何问题，后来改成servlet却后台返回不到前台数据。解决的方法：json对于格式要求非常严格，我就是因为在servlet里多输出了几个字符导致前台无法识别成json格式数据，导致前台无法获得后台数据。

1. **心得和体会**

对于架构和项目的实际开发有了更深的理解。对于sql语句的有了更加深一步的学习。和同学们的交流让自己对于自己所掌握的知识和自己应该去掌握却未掌握的知识有了更深的理解。收获很多。