目录

[1 概述 1](#_Toc9421727)

[1-1 系统介绍 1](#_Toc9421728)

[1-2 系统模块 1](#_Toc9421729)

[2 数据库设计 1](#_Toc9421730)

[2-1 建立数据库 1](#_Toc9421731)

[2-2 设计数据表 2](#_Toc9421732)

[3 系统管理 4](#_Toc9421733)

[3-1 页面管理 4](#_Toc9421734)

[3-2 JavaBean与Servlet管理 6](#_Toc9421735)

[3-3 配置文件管理 7](#_Toc9421736)

[3-4 工具包 10](#_Toc9421737)

[3-5 图像管理 16](#_Toc9421738)

[4 用户 16](#_Toc9421739)

[4-1 用户注册 16](#_Toc9421740)

[4-2 用户登录 25](#_Toc9421741)

[4-3 修改注册信息 32](#_Toc9421742)

[4-4 修改密码 38](#_Toc9421743)

[5 手机 44](#_Toc9421744)

[5-1 浏览手机 44](#_Toc9421745)

[5-2 查询手机 59](#_Toc9421746)

[5-3 查看购物车 65](#_Toc9421747)

[5-4 查看订单 74](#_Toc9421748)

[6 后台管理 76](#_Toc9421749)

[6-1 管理员登录 76](#_Toc9421750)

[6-2 后台管理首页 84](#_Toc9421751)

[6-3 用户管理 84](#_Toc9421752)

[6-4 手机类别管理 103](#_Toc9421753)

[6-5 手机管理 120](#_Toc9421754)

[6-6 订单管理 146](#_Toc9421755)

[7 退出登录 164](#_Toc9421756)

概述

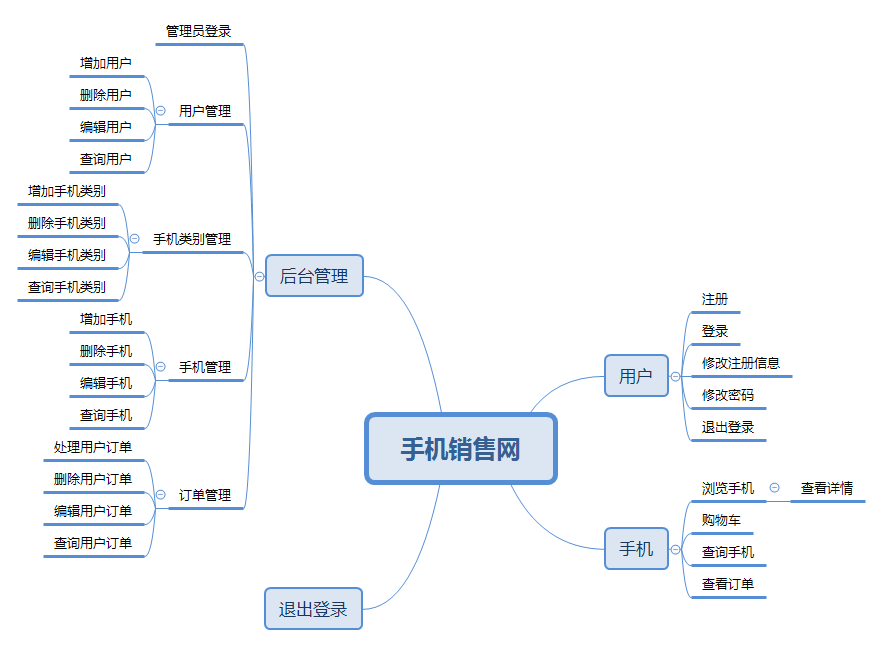
系统介绍

该系统是一个简单的手机销售系统，为用户提供手机查询、生成购物订单以及修改基本信息的功能，其目的是掌握一般Web应用中常用基本模块的开发方法。通过开发这个应用，让我们更加熟悉地掌握和应用JSP相关知识。

该系统的程序是在myeclipse上开发，并运行在Tomcat v8.5上，数据库使用的是MySQL数据库。所以，在开发前还需要在计算机上安装myeclipse、mysql以及数据库管理软件——在这里我用的是navicat。

系统模块

手机销售网的主要模块如下

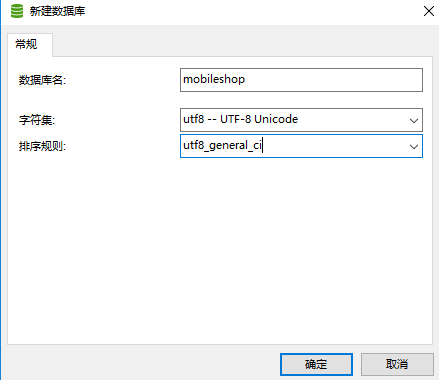


数据库设计

建立数据库

创建数据库“mobileshop”

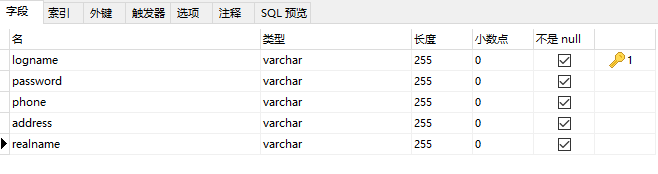
打开navicat新建MYSQL连接，连接到我们计算机的MySQL，在localhost处右击，选择新建数据库，数据库名填写为mobileshop，字符集选择utf8 -- UTF-8 Unicode，排序规则选择utf8\_general\_ci。



设计数据表

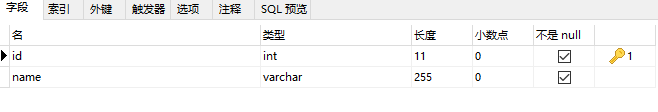
用户表“user”

在建好的数据库mobileshop下的表上方右击，选择新建表，填入四个字段——logname、password、phone、address、realname，这五个字段分别对应用户名、登录密码、邮寄地址、联系电话、真实姓名，点击保存，命名为user。在这边没有对字段做太多限制，所以统一设置为varchar类型，字符集设置为utf8，排序规则设置为utf8\_general\_ci。



手机类别表“mobileclassify”

在建好的数据库mobileshop下的表上方右击，选择新建表，填入两个字段——id、name，这两个字段分别对应手机的分类号、手机的分类名称，其中id设置为int类型，且自增。点击保存，命名为mobileclassify。同样，varchar类型的字段，统一将字符集设置为utf8，排序规则设置为utf8\_general\_ci。

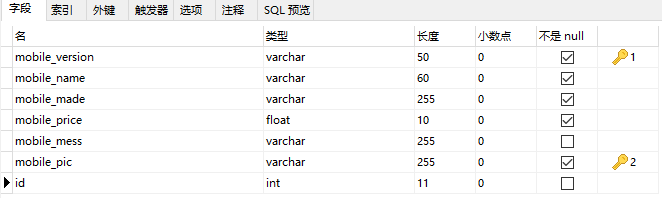


为方便后期系统开发测试，使用navicat在数据表mobileclassify中插入了几条记录。



手机基本信息表“mobileForm”

在建好的数据库mobileshop下的表上方右击，选择新建表，填入七个字段——mobile\_version、mobile\_name、mobile\_made、mobile\_price、mobile\_mess、mobile\_pic、id，这七个字段分别对应手机产品标识号、手机名称、手机的制造商、手机的价格、手机产品的介绍、手机相关的图像、类别ID，其中id作为mobileclassify表的外键。点击保存，命名为info。同样，varchar类型的字段，统一将字符集设置为utf8，排序规则设置为utf8\_general\_ci

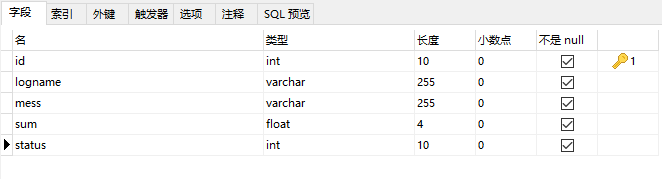


为了方便后期系统开发测试，使用navicat在数据表mobileForm中插入了几条记录。



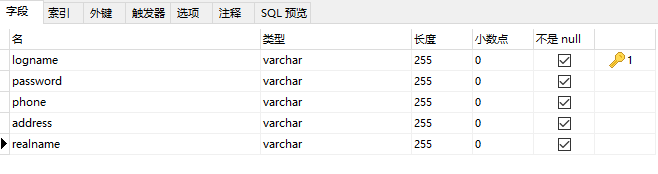
订单信息表“orderForm”

在建好的数据库mobileshop下的表上方右击，选择新建表，填入五个字段——id、logname、mess、sum、status，这五个字段分别对应订单序号、订单用户昵称、订单信息、订单价格综合、订单处理状态，其中status值为0表示未处理状态，值为1表示已处理状态。点击保存，命名为orderForm。同样，varchar类型的字段，统一将字符集设置为utf8，排序规则设置为utf8\_general\_ci



管理员表“user”

在建好的数据库mobileshop下的表上方右击，选择新建表，填入四个字段——logname、password、phone、address、realname，这五个字段分别对应用户名、登录密码、邮寄地址、联系电话、真实姓名，点击保存，命名为management。在这边没有对字段做太多限制，所以统一设置为varchar类型，字符集设置为utf8，排序规则设置为utf8\_general\_ci。



系统管理

页面管理

打开myEclipse，新建Workspace，在这我的Wrkspace是在E盘下的myEclipseProject，你也可以直接使用自己已经创建好的Workspace。打开Workspace，在file处右击，选择New下的Web Project，在Project name处填写mobileShop，Java EE version选择JavaEE7-Web 3.1，然后点击finish。

公用头部文件——head. txt

本系统用的JSP页面全部保存在Web服务目录mobileShop中。所有的页面将包括一个导航条，该导航条由注册、登录、浏览手机、查看订单等组成。为了便于维护，其他页面通过使用JSP的<%@ include…%>标记将head.txt(导航条文件)嵌入到自己的页面中。打开项目mobileShop找到WebRoot>WEB-INF，在目录WEB-INF新建目录tags，在tags中新建文件tophead.txt（文档编码为UTF-8）。tophead.txt文件的内容如下：

**------------------------------ tophead.txt开始 -------------------------------**

<%@ page contentType="text/html;charset=UTF-8"%>

<div align="center">

<H2>"智多星"智能手机销售网</H2>

<table cellSpacing="1" cellPadding="1" width="660" align="center" border="0">

<tr valign="bottom">

<td><a href="<%=registerPath%>"><font size=2>注册</font></a></td>

<td><a href="<%=loginPath%>"><font size=2>登录</font></a></td>

<td><a href="<%=classifyPath%>"><font size=2>浏览手机</font></a></td>

<td><a href="<%=serchMobilePath%>"><font size=2>查询手机</font></a></td>

<td><a href="<%=ShoppingCarPath%>"><font size=2>查看购物车</font></a></td>

<td><a href="<%=lookOrderFormPath%>"><font size=2>查看订单</font></a></td>

<td><a href="<%=changeRegisterPath%>"><font size=2>修改注册信息</font></a></td>

<td><a href="<%=changePassPath%>"><font size=2>修改密码</font></a></td>

<td><a href="http://localhost:8080/mobileShop/management/login.jsp"><font size=2>后台管理</font></a></td>

<td><a href="<%=exitPath%>"><font size=2>退出</font></a></td>

<td><a href="<%=indexPath%>"><font size=2>主页</font></a></td>

</tr>

</table>

</div>

主页——index.jsp

主页 index.jsp由导航条、一个欢迎语和一幅图片 index.jpg组成， index.jpg保存在mobileShop>WebRoot>images中(网站用的图像文件均存在images目录中。用户可以通过在浏览器的地址栏中键入“htp:/服务器IP:8080/ndex.jsp"或“htp:/服务器IP:8080/"访间该主页。主页mobileShop>WebRoot>index.jsp的内容以及运行效果如下：

**--------------------------------- index.jsp开始 --------------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String bannerSrc = path+"/images/index.jpg";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<base href="<%=basePath%>">

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>首页</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

</head>

<body>

<center>

<h1>欢迎光临“智多星”智能手机销售网</h1>

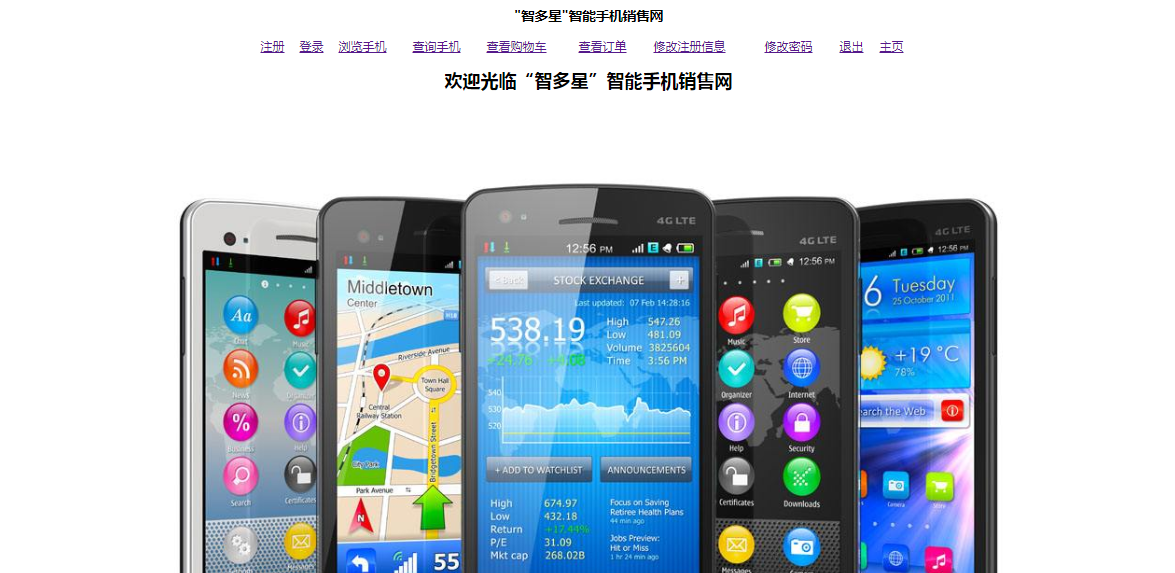
<img src="<%=bannerSrc%>" width="70%"></img>

</center>

</body>

</html>

**--------------------------------- index.jsp结束 --------------------------------**



JavaBean与Servlet管理

为了方便管理商城与后台两部分的代码，本系统的 Javabean类分为两大模块，第一个模块是商城部分的bean包，包名为 mybean.data；第二个模块是后台管理部分的bean包，包名为management.bean。同样的， Servlet类也分为两大模块，商城部分的Servlet类的包名为 myservlet.data，后台管理部分的Servlet类的包名为management.servlet。

配置文件管理

本系统的Servlet文件的包名有两个myservlet.data、myservlet.data，需要配置Web服务目录的web.xml文件,即将下面的web.xml文件保存到 Tomcat安装目录的web服务目录的WEB-INF子目录中。根据我们创建的项目名称，即需要将文件保存到mobileShop>WebRoot>WEB-INF下。

**---------------------------------** web.xml**开始 --------------------------------**

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd" version="3.0">

<servlet>

<description>This is the description of my J2EE component</description>

<display-name>This is the display name of my J2EE component</display-name>

<servlet-name>HandleRegister</servlet-name>

<servlet-class>myservlet.data.HandleRegister</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HandleRegister</servlet-name>

<url-pattern>/servlet/HandleRegister</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>HandleLogin</servlet-name>

<servlet-class>myservlet.data.HandleLogin</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HandleLogin</servlet-name>

<url-pattern>/servlet/HandleLogin</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>QueryRecord</servlet-name>

<servlet-class>myservlet.data.QueryRecord</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>QueryRecord</servlet-name>

<url-pattern>/servlet/QueryRecord</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>PutGoodsToCar</servlet-name>

<servlet-class>myservlet.data.PutGoodsToCar</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>PutGoodsToCar</servlet-name>

<url-pattern>/servlet/PutGoodsToCar</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>HandleDelete</servlet-name>

<servlet-class>myservlet.data.HandleDelete</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HandleDelete</servlet-name>

<url-pattern>/servlet/HandleDelete</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>HandleBuyGoods</servlet-name>

<servlet-class>myservlet.data.HandleBuyGoods</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HandleBuyGoods</servlet-name>

<url-pattern>/servlet/HandleBuyGoods</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>searchByCondition</servlet-name>

<servlet-class>myservlet.data.searchByCondition</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>searchByCondition</servlet-name>

<url-pattern>/servlet/searchByCondition</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>HandleExit</servlet-name>

<servlet-class>myservlet.data.HandleExit</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HandleExit</servlet-name>

<url-pattern>/servlet/HandleExit</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>HandleChangePass</servlet-name>

<servlet-class>myservlet.data.HandleChangePass</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HandleChangePass</servlet-name>

<url-pattern>/servlet/HandleChangePass</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>ChangeInfoOfRegister</servlet-name>

<servlet-class>myservlet.data.ChangeInfoOfRegister</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ChangeInfoOfRegister</servlet-name>

<url-pattern>/servlet/ChangeInfoOfRegister</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>ManagementLogin</servlet-name>

<servlet-class>management.servlet.ManagementLogin</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ManagementLogin</servlet-name>

<url-pattern>/servlet/ManagementLogin</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>UserManage</servlet-name>

<servlet-class>management.servlet.UserManage</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>UserManage</servlet-name>

<url-pattern>/servlet/UserManage</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>GoodsTypeManage</servlet-name>

<servlet-class>management.servlet.GoodsTypeManage</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>GoodsTypeManage</servlet-name>

<url-pattern>/servlet/GoodsTypeManage</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>GoodsManage</servlet-name>

<servlet-class>management.servlet.GoodsManage</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>GoodsManage</servlet-name>

<url-pattern>/servlet/GoodsManage</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>OrderFormManage</servlet-name>

<servlet-class>management.servlet.OrderFormManage</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>OrderFormManage</servlet-name>

<url-pattern>/servlet/OrderFormManage</url-pattern>

</servlet-mapping>

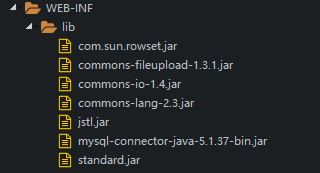
</web-app>

**---------------------------------**web.xml**结束 --------------------------------**

工具包

第三方工具包

打开刚刚新建的项目mobileShop，找到WebRoot>WEB-INF>lib文件夹，将下图中的jar包复制到lib文件夹下，下图的jar包资源可以在项目压缩文件中找到。引入的jar包可以分为3类，第一个jar包是为了应用rowset，倒数第二个是连接MySQL数据库，其余5个是文件上传使用到的相关jar包。



创建工具包

打开mobileShop，找到src目录，右击选择New，点击Package，在Name中填写org.utils，点击finish。同样创建第二个工具包Name为management.tool。

1.JdbcUtil.java

在创建好的org.utils包中右击选择New，点击Class，在name中填写JdbcUtil，点击finish。打开新建好的JdbcUtil.java文件，将下面代码复制到文件中（注意看文件头部中的url、username、password是否和你数据库的连接地址、用户名、密码一致，不一致则需要修改）。这部分的代码主要就是处理与数据库的连接，因为之后要反复使用到所以封装出来比较方便。

**----------------------------- JdbcUtil.java开始 --------------------------**

package org.utils;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class JdbcUtil {

private static String url ="jdbc:mysql://127.0.0.1/mobileshop?useUnicode=true&characterEncoding=utf8&allowMultiQueries=true";

private static String username="root";

private static String password="";

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

// static静态代码块里面代码，会在类加载到内存的时候立刻执行

static {

//1.注册驱动

try {

Class.forName(driverName);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

/\*\*

\* 获取连接

\* @return

\* @throws SQLException

\*/

public static Connection getConnection() throws SQLException {

//2.获取连接

Connection connection = DriverManager.getConnection(url, username, password);

return connection;

}

public static void close(Connection connection,PreparedStatement prepareStatement,ResultSet resultSet) {

//5.释放资源 connection prepareStatement resultSet

if(resultSet!=null) {

try {

resultSet.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(prepareStatement!=null) {

try {

prepareStatement.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(connection!=null) {

try {

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void statementClose(Connection connection,Statement statement,ResultSet resultSet) {

//5.释放资源 connection prepareStatement resultSet

if(resultSet!=null) {

try {

resultSet.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(statement!=null) {

try {

statement.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(connection!=null) {

try {

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

**----------------------------- JdbcUtil.java 结束----------------------------**

2. FileUploadUtil.java

在创建好的org.utils包中右击选择New，点击Class，在name中填写FileUploadUtil，点击finish。打开新建好的FileUploadUtil.java文件，将下面代码复制到文件中这部分的代码主要就是处理文件上传。

**--------------------------- FileUploadUtil.java 开始--------------------------**

package org.utils;

import java.util.Calendar;

import org.apache.commons.lang.StringUtils;

/\*\*

\* @author wzd

\*/

public class FileUploadUtil {

private FileUploadUtil() {

super();

}

/\*\*

\* @param fileName

\* @return

\*/

public static String getFileExt(String fileName){

String ext="";

if(StringUtils.isNotEmpty(fileName)&&

StringUtils.contains(fileName, ".")){

ext=StringUtils.substring(fileName, fileName.lastIndexOf(".")+1);

}

return ext;

}

/\*\*

\*/

public static String getTimeDesPath(){

Calendar cal=Calendar.getInstance();

return "/"+cal.get(Calendar.YEAR)+"/"+(cal.get(Calendar.MONTH)+1)+"/";

}

}

**--------------------------- FileUploadUtil.java 结束---------------------------**

3. PageResult.java

在创建好的management.tool包中右击选择New，点击Class，在name中填写PageResult，点击finish。打开新建好的PageResult.java文件，将下面代码复制到文件中这部分的代码主要就是处理分页。

**------------------------------ PageResult.java 开始----------------------------**

package management.tool;

import java.util.List;

/\*\*

\* @author ljgui

\*

\*/

public class PageResult {

private List<?> list;

private Long totalRecord;

private Integer pageSize;

private Integer currentPage;

private Integer totalPages;

/\*\*

\*/

private Integer beginIndex;

/\*\*

\*/

private Integer endIndex;

public PageResult(List<?> list, Long totalRecord, Integer pageSize, Integer currentPage) {

this.list = list;

this.totalRecord = totalRecord;

this.pageSize = pageSize;

this.currentPage = currentPage;

//

if(totalRecord.intValue() % pageSize==0) {

this.totalPages = totalRecord.intValue() / pageSize;

}else {

this.totalPages = totalRecord.intValue() / pageSize +1;

}

//

if(totalPages<=3) {

this.beginIndex=1;

this.endIndex = totalPages;

}else {

this.beginIndex = currentPage-1;

this.endIndex = currentPage + 1;

if(this.beginIndex<1) {

this.beginIndex=1;

this.endIndex = 3;

}

if(this.endIndex>totalPages) {

this.endIndex=totalPages;

this.beginIndex= totalPages-3;

}

}

}

public List<?> getList() {

return list;

}

public void setList(List<?> list) {

this.list = list;

}

public Long getTotalRecord() {

return totalRecord;

}

public void setTotalRecord(Long totalRecord) {

this.totalRecord = totalRecord;

}

public Integer getPageSize() {

return pageSize;

}

public void setPageSize(Integer pageSize) {

this.pageSize = pageSize;

}

public Integer getCurrentPage() {

return currentPage;

}

public void setCurrentPage(Integer currentPage) {

this.currentPage = currentPage;

}

public Integer getTotalPages() {

return totalPages;

}

public void setTotalPages(Integer totalPages) {

this.totalPages = totalPages;

}

public Integer getBeginIndex() {

return beginIndex;

}

public void setBeginIndex(Integer beginIndex) {

this.beginIndex = beginIndex;

}

public Integer getEndIndex() {

return endIndex;

}

public void setEndIndex(Integer endIndex) {

this.endIndex = endIndex;

}

}

**------------------------------ PageResult.java 结束----------------------------**

图像管理

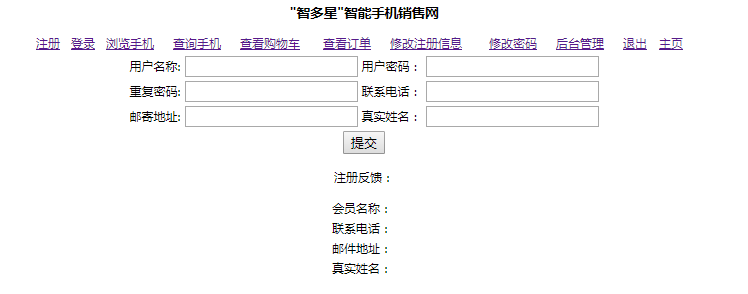
本系统用到的图片对应的图像文件，比如手机图片用的图像文件，均保存在Web服务目录mobileShop>WebRoot>images中。

用户

用户注册

视图（JSP页面）

该模块视图部分由一个JSP页面构成，这个jsp页面负责提交用户的注册信息到 servlet控制器HandleRegister (见配置文件web.xml)，并负责显示注册是否成功的信息 。首先在WebRoot下新建目录pages,在目录pages下 新建Jsp文件命名为inputRegister.jsp效果如图所示。



**------------------------------ inputRegister.jsp 开始--------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<jsp:useBean id="userBean" class="mybean.data.Register" scope="request"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String handleRegisterPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleRegister";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>My JSP 'inputRegister.jsp' starting page</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<body>

<div align="center">

<form action="<%=handleRegisterPath%>" method="post" name="form">

<table>

<tr>

<td>用户名称:</td><td><input type="text" name="logname"></td>

<td>用户密码：</td><td><input type="text" name="password"></td>

</tr>

<tr>

<td>重复密码:</td><td><input type="text" name="again\_password"></td>

<td>联系电话：</td><td><input type="text" name="phone"></td>

</tr>

<tr>

<td>邮寄地址:</td><td><input type="text" name="address"></td>

<td>真实姓名：</td><td><input type="text" name="realname"></td>

</tr>

<tr>

<td colspan="4" align="center"><input type="submit" name="g" value="提交"/></td>

</tr>

</table>

</form>

</div>

<div align="center">

<p>注册反馈：<jsp:getProperty name="userBean" property="backnews"/></p>

<table>

<tr>

<td>会员名称：</td>

<td><jsp:getProperty name="userBean" property="logname"/></td>

</tr>

<tr>

<td>联系电话：</td>

<td><jsp:getProperty name="userBean" property="phone"/></td>

</tr>

<tr>

<td>邮件地址：</td>

<td><jsp:getProperty name="userBean" property="address"/></td>

</tr>

<tr>

<td>真实姓名：</td>

<td><jsp:getProperty name="userBean" property="realname"/></td>

</tr>

</table>

</div>

</body>

</html>

**------------------------------ inputRegister.jsp结束--------------------------**

模型（JavaBean）

打开mobileShop，找到src目录，右击选择New，点击Package，在Name中填写mybean.data，点击finish。在创建好的包mybean.data中新建Class文件命名为Register，Register.java是用于描述用户注册的重要信息，包括对用户信息的设置和获取。Register.java文件的具体代码如下：

**-------------------------------- Register.java开始----------------------------**

package mybean.data;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import org.utils.JdbcUtil;

public class Register {

String logname="",phone="",address="",realname="",backnews="";

public String getLogname() {

return logname;

}

public void setLogname(String logname) {

this.logname = logname;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getRealname() {

return realname;

}

public void setRealname(String realname) {

this.realname = realname;

}

public String getBacknews() {

return backnews;

}

public void setBacknews(String backnews) {

this.backnews = backnews;

}

//处理中文字符串

public String handleString(String s) {

try {

byte bb[]=s.getBytes("ISO8859-1");

s=new String(bb,"UTF-8");

}catch(Exception e){}

return s;

}

public void queryByLogname(String name) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet rs = null;

try {

connection = JdbcUtil.getConnection();

//3.获取statement

String sql ="select \* from user where logname = '"+name+"'";

prepareStatement = connection.prepareStatement(sql);

//4.执行sql

rs = prepareStatement.executeQuery();

//5.处理查询结果

while(rs.next()) {

String logname = rs.getString("logname");

String phone = rs.getString("phone");

String address = rs.getString("address");

String realname = rs.getString("realname");

this.setLogname(logname);

this.setPhone(phone);

this.setAddress(address);

this.setRealname(realname);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtil.close(connection, prepareStatement, rs);

}

}

}

**-------------------------------- Register.java结束----------------------------**

控制器（servlet）

打开mobileShop，找到src目录，右击选择New，点击Package，在Name中填写myservlet.data，点击finish。在创建好的包下新建Class命名为HandleRegister(见配置文件web.xml)，该文件负责连接数据库将用户提交的信息写到user表中，并将用户转发到inputRegister.jsp页面查看注册反馈信息。HandleRegister.java文件内容如下：

**-----------------------------HandleRegister.java开始--------------------------**

package myservlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtil;

import mybean.data.Register;

/\*\*

\* Servlet implementation class HandleRegister

\*/

@WebServlet("/HandleRegister")

public class HandleRegister extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public HandleRegister() {

super();

// TODO Auto-generated constructor stub

}

//处理中文字符串

public String handleString(String s) {

try {

byte bb[]=s.getBytes("ISO8859-1");

s=new String(bb,"UTF-8");

}catch(Exception e){}

return s;

}

/\*\*

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

\*/

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

//创建user的JavaBean模型

Register userBean = new Register();

request.setAttribute("userBean", userBean);

String logname = request.getParameter("logname").trim();

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

String again\_password = request.getParameter("again\_password").trim();

String phone = request.getParameter("phone").trim();

String address = request.getParameter("address").trim();

String realname = request.getParameter("realname").trim();

// out.print(logname+"--"+password);

if(logname == null) {

logname = "";

}

if(password == null) {

password = "";

}

if(!password.equals(again\_password)) {

userBean.setBacknews("两次密码输入不一致，注册失败");

/\*转发不改变地址栏且可以将数据保存到request\*/

RequestDispatcher dispatcher = request.getRequestDispatcher("/pages/inputRegister.jsp");

dispatcher.forward(request, response);

return;

/\*重定向会改变地址栏地址，且并不能将数据保存到request\*/

// response.sendRedirect(request.getContextPath()+"/login.jsp");

// return ;

}

Connection connection = null;

PreparedStatement prepareStatement = null;

String backnews="";

boolean boo = false;

boolean isLeagal = true;

for(int i=0; i<logname.length(); i++) {

char c = logname.charAt(i);

if(!((c<='z'&&c>='a')||(c<='Z'&&c>='A')||(c<='9'&&c>='0')))

isLeagal = false;

}

boo = logname.length()>0&&password.length()>0&&isLeagal;

try {

connection = JdbcUtil.getConnection();

//3.获取statement

//String sql ="insert into user(logname,password,phone,address,realname) value(?,?,?,?,?) ";

String sql ="insert into user values(?,?,?,?,?) ";

prepareStatement = connection.prepareStatement(sql);

if(boo) {

prepareStatement.setString(1, logname);

prepareStatement.setString(2, password);

prepareStatement.setString(3, phone);

prepareStatement.setString(4, address);

prepareStatement.setString(5, realname);

//4.执行sql

int m = prepareStatement.executeUpdate();

if(m!=0) {

backnews = "register succeeded!";

userBean.setBacknews(backnews);

userBean.setLogname(logname);

userBean.setPhone(phone);

userBean.setAddress(address);

userBean.setRealname(handleString(realname));

out.print("register succeeded!");

}else {

backnews="注册失败";

userBean.setBacknews(backnews);

}

}else {

backnews="信息填写不完整或，名字中有非法字符";

userBean.setBacknews(backnews);

}

} catch (Exception e) {

backnews="该会员名称已被使用，请你更换名字";

userBean.setBacknews(backnews);

out.print(e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtil.close(connection, prepareStatement, null);

RequestDispatcher dispatcher = request.getRequestDispatcher("/pages/inputRegister.jsp");

dispatcher.forward(request, response);

}

out.flush();

out.close();

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

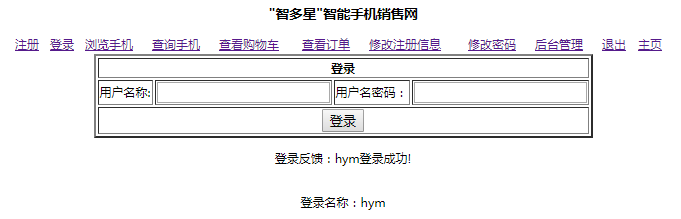
}

**----------------------------HandleRegister.java结束---------------------------**

用户登录

视图（JSP页面）

该模块视图部分由一个JSP页面构成，这个jsp页面负责提交用户的登录信息到 servlet控制器，并负责显示登录是否成功的信息 。在WebRoot下找到建好的目录pages,在目录pages下 新建Jsp文件命名为login.jsp效果如图所示;



**-----------------------------------login.jsp开始-----------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<jsp:useBean id="loginBean" class="mybean.data.Login" scope="session"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String cssPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/assets/css/login.css";

String handleLoginPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleLogin";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>My JSP 'login.jsp' starting page</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<%-- <link rel="stylesheet" type="text/css" href="<%=cssPath%>"> --%>

</head>

<body>

<div align="center">

<form action="<%=handleLoginPath%>" Method="post" name="form">

<table border=2>

<tr><th colspan="4" >登录</th></tr>

<tr>

<td>用户名称:</td><td><input type="text" name="logname"></td>

<td>用户名密码：</td><td><input type="text" name="password"></td>

</tr>

<tr>

<td colspan="4" align="center"><input type="submit" name="l" value="登录"/></td>

</tr>

</table>

</form>

</div>

<div align="center">

<p>登录反馈：<jsp:getProperty name="loginBean" property="backnews"/></p>

<br>登录名称：<jsp:getProperty name="loginBean" property="logname"/>

</div>

</body>

</html>

**---------------------------------- login.jsp结束------------------------------**

模型（JavaBean）

打开mobileShop，找到src目录，在创建好的包mybean.data中新建Class文件命名为Login，Login.java是用于描述用户登录的重要信息，包括对用户登录信息的设置和获取。Login.java文件的具体代码如下：

**---------------------------------- login.java开始----------------------------**

package mybean.data;

import java.util.LinkedList;

public class Login {

String logname="",backnews="",changePassBacknews="",changeRegisterInfoBack="";

public String getChangeRegisterInfoBack() {

return changeRegisterInfoBack;

}

public void setChangeRegisterInfoBack(String changeRegisterInfoBack) {

this.changeRegisterInfoBack = changeRegisterInfoBack;

}

LinkedList<String> car; //链表，用户的购物车

public String getChangePassBacknews() {

return changePassBacknews;

}

public void setChangePassBacknews(String changePassBacknews) {

this.changePassBacknews = changePassBacknews;

}

public Login() {

car = new LinkedList<String>();

}

public LinkedList<String> getCar(){

return car;

}

public String getLogname() {

return logname;

}

public void setLogname(String logname) {

this.logname = logname;

}

public String getBacknews() {

return backnews;

}

public void setBacknews(String backnews) {

this.backnews = backnews;

}

}

**---------------------------------- login.java结束----------------------------**

控制器（servlet）

打开mobileShop，找到src目录，在创建好的包myservlet.data下新建Class命名为HandleLogin (见配置文件web.xml)，该文件负责连接数据库将用户提交的信息与user表中的信息对比，并将用户转发到login.jsp页面查看注册反馈信息。HandleLogin.java文件内容如下：

**-------------------------------HandleLogin.java开始--------------------------**

package myservlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.utils.JdbcUtil;

import mybean.data.Login;

/\*\*

\* Servlet implementation class HandleLogin

\*/

@WebServlet("/HandleLogin")

public class HandleLogin extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

super.init(config);

}

//处理中文字符串

public String handleString(String s) {

try {

byte bb[]=s.getBytes("ISO8859-1");

s=new String(bb,"UTF-8");

}catch(Exception e){}

return s;

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String logname = request.getParameter("logname").trim();

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

logname = handleString(logname);

password = handleString(password);

if(logname == null) {

logname = "";

}

if(password == null) {

password = "";

}

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet rs = null;

String backnews="";

boolean boo = false;

boo = logname.length()>0&&password.length()>0;

try {

connection = JdbcUtil.getConnection();

//3.获取statement

//String sql ="insert into user(logname,password,phone,address,realname) value(?,?,?,?,?) ";

String sql ="select \* from user where logname=? and password=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, logname);

prepareStatement.setString(2, password);

if(boo) {

//4.执行sql

rs = prepareStatement.executeQuery();

boolean m = rs.next();

if(m==true) {

//登录成功

success(request,response,logname,password);

RequestDispatcher dispatcher = request.getRequestDispatcher("/pages/login.jsp");

dispatcher.forward(request, response);

}else {

backnews="你输入的用户名或密码有误！";

fail(request,response,logname,backnews);

}

}else {

backnews="请输入用户名和密码";

fail(request,response,logname,backnews);

}

} catch (Exception e) {

backnews=""+e.toString();

out.print(e.toString());

e.printStackTrace();

fail(request,response,logname,backnews);

}finally {

//5.释放资源 connection prepareStatement

JdbcUtil.close(connection, prepareStatement, rs);

}

out.flush();

out.close();

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

public void success(HttpServletRequest request, HttpServletResponse response,String logname,String password) {

Login loginBean = null;

HttpSession session = request.getSession(true);

try {

//创建user的JavaBean模型

loginBean = (Login)session.getAttribute("loginBean");

if(loginBean == null) {

loginBean = new Login(); //还未登录，创建新的模型

session.setAttribute("loginBean", loginBean);

loginBean = (Login)session.getAttribute("loginBean");

}

String name = loginBean.getLogname();

if(name.equals(logname)) {

loginBean.setBacknews(logname+"已经登录了!");

loginBean.setLogname(logname);

}else {

loginBean.setBacknews(logname+"登录成功!");

loginBean.setLogname(logname);

}

}catch(Exception e) {

loginBean = new Login(); //还未登录，创建新的模型

session.setAttribute("loginBean", loginBean);

loginBean.setBacknews(logname+"登录成功!"+e.toString());

loginBean.setLogname(logname);

}

}

public void fail(HttpServletRequest request, HttpServletResponse response,String logname,String backnews) {

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

try {

PrintWriter out = response.getWriter();

String loginPath = request.getContextPath()+"/pages/login.jsp";

String registerPath = request.getContextPath()+"/index.jsp";

out.println("<html><body><div align=center>");

out.println("<h2>"+logname+"登录反馈结果<br>"+backnews+"</h2>");

out.println("返回登录页面或主页<br>");

out.println("<a href="+loginPath+">登录页面</a>");

out.println("<a href="+registerPath+">登录页面</a>");

out.println("</div></body></html>");

}catch(IOException e) {

Login loginBean = null;

HttpSession session = request.getSession(true);

loginBean = (Login)session.getAttribute("loginBean");

loginBean.setBacknews("抛出异常"+e.toString());

}

}

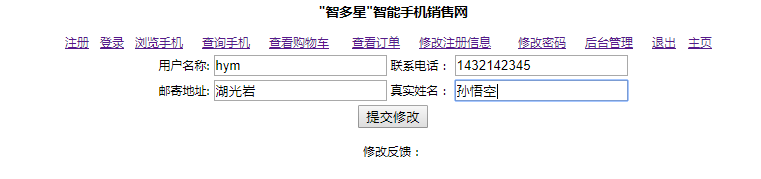
}

**-------------------------------HandleLogin.java结束---------------------------**

修改注册信息

视图（JSP页面）

该模块视图部分由一个JSP页面构成，这个jsp页面负责提交用户的修改后的信息到 servlet控制器，并负责显示修改是否成功的信息 。在WebRoot下找到建好的目录pages,在目录pages下 新建Jsp文件命名为ChangeInfoOfRegister.jsp效果如图所示;



**------------------------- changeRegisterInfo.jsp开始-------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<jsp:useBean id="userBean" class="mybean.data.Register" scope="request"/>

<jsp:useBean id="loginBean" class="mybean.data.Login" scope="session"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String changeRegisterInfoPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/ChangeInfoOfRegister";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>My JSP 'inputRegister.jsp' starting page</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<body>

<%

if(loginBean == null){

response.sendRedirect(loginPath); //重定向到登录页

return;

}else{

boolean b = loginBean.getLogname()==null||loginBean.getLogname().length()==0;

if(b){

response.sendRedirect(loginPath);

return;

}

}

userBean.queryByLogname(loginBean.getLogname().toString());

%>

<div align="center">

<form action="<%=changeRegisterInfoPath%>" method="post" name="form">

<table>

<tr>

<td>用户名称:</td><td><input type="text" name="logname" value=<jsp:getProperty name="userBean" property="logname"/>></td>

<td>联系电话：</td><td><input type="text" name="phone" value=<jsp:getProperty name="userBean" property="phone"/>></td>

</tr>

<tr>

<td>邮寄地址:</td><td><input type="text" name="address" value=<jsp:getProperty name="userBean" property="address"/>></td>

<td>真实姓名：</td><td><input type="text" name="realname" value=<jsp:getProperty name="userBean" property="realname"/>></td>

</tr>

<tr>

<td colspan="4" align="center"><input type="submit" name="g" value="提交修改"/></td>

</tr>

</table>

</form>

</div>

<div align="center">

<p>修改反馈：<jsp:getProperty name="loginBean" property="changeRegisterInfoBack"/></p>

</div>

</body>

</html>

**--------------------------** **changeRegisterInfo.jsp结束-------------------------**

模型（JavaBean）

在修改用户注册信息这块使用的模型是已经创建好的mybean.data包下的 Login.java，与用户登录使用的是同一个bean（详情见4-2-2）

控制器（servlet）

打开mobileShop，找到src目录，在创建好的包myservlet.data下新建Class命名为ChangeInfoOfRegister (见配置文件web.xml)，该文件负责连接数据库将用户提交的修改信息更新到user表中，并将用户转发到changeRegisterInfo.jsp页面查看修改反馈信息。ChangeInfoOfRegister.java文件内容如下：

**-------------------------ChangeInfoOfRegister.java开始------------------------**

package myservlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.utils.JdbcUtil;

import mybean.data.Login;

/\*\*

\* Servlet implementation class ChangeRegisterInfo

\*/

@WebServlet("/ChangeRegisterInfo")

public class ChangeInfoOfRegister extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ChangeInfoOfRegister() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String logname = request.getParameter("logname").trim();

String phone = request.getParameter("phone").trim();

String address = request.getParameter("address").trim();

String realname = request.getParameter("realname").trim();

boolean isLeagal = true;

for(int i=0; i<logname.length(); i++) {

char c = logname.charAt(i);

if(!((c<='z'&&c>='a')||(c<='Z'&&c>='A')||(c<='9'&&c>='0')))

isLeagal = false;

}

Login loginBean = null;

HttpSession session = request.getSession(true);

Connection connection = null;

PreparedStatement prepareStatement = null;

PreparedStatement prepareStatement2 = null;

ResultSet rs = null;

ResultSet rs2 =null;

String backnews="";

boolean boo = false;

boo = (logname.length()>0||logname!=null)&&isLeagal;

try {

connection = JdbcUtil.getConnection();

loginBean = (Login)session.getAttribute("loginBean");

String name = loginBean.getLogname();

//3.获取statement

String sql ="select \* from user where logname=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, name);

if(boo) {

//4.执行sql

rs = prepareStatement.executeQuery();

boolean m = rs.next();

if(m==true) {

//查询成功

String changeSql = "update user set logname=?,phone=?,address=?,realname=? where logname=?";

prepareStatement2 = connection.prepareStatement(changeSql);

prepareStatement2.setString(1, logname);

prepareStatement2.setString(2, phone);

prepareStatement2.setString(3, address);

prepareStatement2.setString(4, realname);

prepareStatement2.setString(5, name);

prepareStatement2.execute();

loginBean.setLogname(logname);

backnews="注册信息修改成功";

}else {

backnews="未查找到相关信息！";

}

}else {

backnews="请输入用户名和密码";

}

loginBean.setChangeRegisterInfoBack(backnews);

RequestDispatcher dispatcher = request.getRequestDispatcher("/pages/changeRegisterInfo.jsp");

dispatcher.forward(request, response);

} catch (Exception e) {

backnews="修改失败"+e.toString();

loginBean.setChangeRegisterInfoBack(backnews);

out.print(e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtil.close(connection, prepareStatement, rs);

JdbcUtil.close(connection, prepareStatement2, rs2);

}

out.flush();

out.close();

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

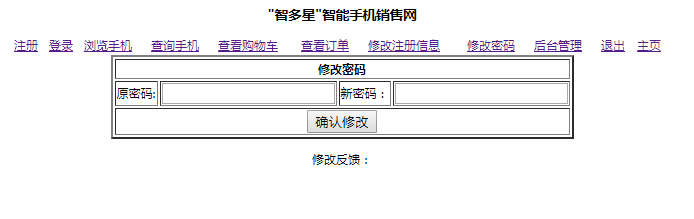
}

**------------------------- ChangeInfoOfRegister.java结束----------------------**

修改密码

视图（JSP页面）

该模块视图部分由一个JSP页面构成，这个jsp页面负责提交用户的修改密码后的信息到 servlet控制器，并负责显示修改是否成功的信息 。在WebRoot下找到建好的目录pages,在目录pages下 新建Jsp文件命名为changePassword.jsp效果如图所示;



**-------------------------**----- **changePassword.jsp开始-------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<jsp:useBean id="loginBean" class="mybean.data.Login" scope="session"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String cssPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/assets/css/login.css";

String handleChangePassPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleChangePass";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>My JSP 'login.jsp' starting page</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<%-- <link rel="stylesheet" type="text/css" href="<%=cssPath%>"> --%>

</head>

<body>

<%

if(loginBean==null){

response.sendRedirect(loginPath); //重定向到登录页

return;

}else{

boolean b = loginBean.getLogname()==null||loginBean.getLogname().length()==0;

if(b){

response.sendRedirect(loginPath);

return;

}

}

%>

<div align="center">

<form action="<%=handleChangePassPath%>" Method="post" name="form">

<table border=2>

<tr><th colspan="4" >修改密码</th></tr>

<tr>

<td>原密码:</td><td><input type="text" name="oldPassword"></td>

<td>新密码：</td><td><input type="text" name="newPassword"></td>

</tr>

<tr>

<td colspan="4" align="center"><input type="submit" name="l" value="确认修改"/></td>

</tr>

</table>

</form>

</div>

<div align="center">

<p>修改反馈：<jsp:getProperty name="loginBean" property="changePassBacknews"/></p>

</div>

</body>

</html>

**-------------------------**----- **changePassword.jsp结束------------------------**

模型（JavaBean）

在修改用户面膜这块使用的模型是已经创建好的mybean.data包下的 Login.java，与用户登录使用的是同一个bean（详情见4-2-2）

控制器（servlet）

打开mobileShop，找到src目录，在创建好的包myservlet.data下新建Class命名为HandleChangePass (见配置文件web.xml)，该文件负责连接数据库将用户提交的信息更新到user表中，并将用户转发到changePassword.jsp页面查看修改密码的反馈信息。HandleChangePass.java文件内容如下：

**----------------------------HandleChangePass.java开始-------------------------**

package myservlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.utils.JdbcUtil;

import mybean.data.Login;

/\*\*

\* Servlet implementation class HandleChangePass

\*/

@WebServlet("/HandleChangePass")

public class HandleChangePass extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public HandleChangePass() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String oldPassword = request.getParameter("oldPassword").trim();

String newPassword = request.getParameter("newPassword").trim();

if(oldPassword == null) {

oldPassword = "";

}

if(newPassword == null) {

newPassword = "";

}

Login loginBean = null;

HttpSession session = request.getSession(true);

Connection connection = null;

PreparedStatement prepareStatement = null;

PreparedStatement prepareStatement2 = null;

ResultSet rs = null;

ResultSet rs2 =null;

String backnews="";

String logname = "";

boolean boo = false;

boo = oldPassword.length()>0&&newPassword.length()>0;

try {

connection = JdbcUtil.getConnection();

loginBean = (Login)session.getAttribute("loginBean");

logname = loginBean.getLogname();

//3.获取statement

//String sql ="insert into user(logname,password,phone,address,realname) value(?,?,?,?,?) ";

String sql ="select \* from user where logname=? and password=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, logname);

prepareStatement.setString(2, oldPassword);

if(boo) {

//4.执行sql

rs = prepareStatement.executeQuery();

boolean m = rs.next();

if(m==true) {

//查询成功

String changeSql = "update user set password=? where logname=?";

prepareStatement2 = connection.prepareStatement(changeSql);

prepareStatement2.setString(1, newPassword);

prepareStatement2.setString(2, logname);

prepareStatement2.execute();

backnews="密码修改成功";

}else {

backnews="你输入的密码有误！";

}

}else {

backnews="请输入用户名和密码";

}

loginBean.setChangePassBacknews(backnews);

RequestDispatcher dispatcher = request.getRequestDispatcher("/pages/changePassword.jsp");

dispatcher.forward(request, response);

} catch (Exception e) {

backnews="修改失败"+e.toString();

out.print(e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtil.close(connection, prepareStatement, rs);

JdbcUtil.close(connection, prepareStatement2, rs2);

}

out.flush();

out.close();

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**----------------------------HandleChangePass.java结束-------------------------**

手机

浏览手机

视图（JSP页面）

该模块视图部分由3个JSP页面 lookMobile.jsp、 byPageShow.jsp和 show Detail.jsp构成,在look Mobile.jsp页面选择某个分类，比如 iphone手机，然后提交给 servlet控制器QueryRecord，该控制器将查询结果放到 Javabean模型 databean中，然后将显示 dataBean 中的数据的任务交给 byPageshow.jsp页面，byPageShow.jsp可以分页显示 dataBean中的数据，即分页显示记录，用户在 bypageShow页面看到产品后，可以选择“查看细节”，到 showdetail.jsp页面查看该产品的细节， lookMobile.jsp、 byPageshow.isp和showDetail.jsp页面均在目录WebRoot> pages下，页面的显示效果如图所示。;



**------------------------------ lookMobile.jsp开始----------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<%@ page import="java.sql.\*"%>

<%@ page import="org.utils.JdbcUtil"%>

<jsp:useBean id="classifyBean" class="mybean.data.Classify" scope="session"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String cssPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/assets/css/login.css";

String handleRegisterPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleRegister";

String queryRecordPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/QueryRecord";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>My JSP 'lookMobile.jsp' starting page</title>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<link rel="stylesheet" type="text/css" href="<%=cssPath%>">

</head>

<body>

<%classifyBean.queryAll();%>

<div align="center">

<jsp:getProperty name="classifyBean" property="backnews"/>

</div>

</body>

</html>

**--------------------------** ----**lookMobile.jsp结束----------------------------**



**------------------------------ byPageShow.jsp开始----------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<%@ page import="mybean.data.DataByPage"%>

<%@ page import="com.sun.rowset.\*"%>

<jsp:useBean id="dataBean" class="mybean.data.DataByPage" scope="session"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String detailPath = path+"/pages/showDetail.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String putGoodsPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/PutGoodsToCar";

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>My JSP 'byPageShow.jsp' starting page</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<body>

<div align="center">

<table border=2>

<tr>

<th>手机识别号</th>

<th>手机名称</th>

<th>手机制造商</th>

<th>手机价格</th>

<th>查看详情</th>

<th><font color=blue>添加到购物车</font></th>

</tr>

<jsp:setProperty name="dataBean" property="pageSize" param ="pageSize"/>

<jsp:setProperty name="dataBean" property="currentPage" param ="currentPage"/>

<%

CachedRowSetImpl rowSet = dataBean.getRowSet();

if(rowSet==null){

out.print("没有任何查询信息，无法浏览");

return;

}

rowSet.last();

int totalRecord = rowSet.getRow();

out.println("全部记录数"+totalRecord); //全部的记录数

int pageSize = dataBean.getPageSize(); //显示的记录数

int totalPage = dataBean.getTotalPage();

if(totalRecord%pageSize==0){

totalPage = totalRecord/pageSize; //总页数

}else{

totalPage = totalRecord/pageSize+1;

}

dataBean.setPageSize(pageSize);

dataBean.setTotalPage(totalPage);

if(totalPage>=1){

if(dataBean.getCurrentPage()<1){

dataBean.setCurrentPage(dataBean.getTotalPage()); //已超过最前一页时，翻到最后一页

};

if(dataBean.getCurrentPage()<1){

dataBean.setCurrentPage(dataBean.getTotalPage());

};

int index = (dataBean.getCurrentPage()-1)\*pageSize+1;

rowSet.absolute(index); //查询位置移动到当前页的起始记录位置

boolean boo = true;

for(int i=1; i<=pageSize&&boo;i++){

String number = rowSet.getString(1);

String name = rowSet.getString(2);

String maker = rowSet.getString(3);

String price = rowSet.getString(4);

String goods = "("+number+","+name+","+maker+","+price+")#"+price; //便于购物车计算价格，尾缀上“#价格值”

goods = goods.replaceAll("\\p{Blank}","");

String button = "<form action = '"+putGoodsPath+"' method='post'>"+

"<input type='hidden' name='java' value="+goods+">"+

"<input type='submit' value='放入购物车'></form>";

String detail = "<form action = '"+detailPath+"' method='post'>"+

"<input type='hidden' name='detail' value="+number+">"+

"<input type='submit' value='查看细节'></form>";

out.print("<tr>");

out.print("<td>"+number+"</td>");

out.print("<td>"+name+"</td>");

out.print("<td>"+maker+"</td>");

out.print("<td>"+price+"</td>");

out.print("<td>"+detail+"</td>");

out.print("<td>"+button+"</td>");

out.print("</tr>");

boo = rowSet.next();

}

}

%>

</table>

<p>每页最多显示<jsp:getProperty name="dataBean" property="pageSize"/> 条信息</p>

<p>当前显示第<font color=blue><jsp:getProperty name="dataBean" property="currentPage"/>页</font>

</p>

<table>

<tr>

<td>

<form action="" method=post>

<input type="hidden" name="currentPage" value="<%=dataBean.getCurrentPage()-1%>"/>

<input type="submit" name="g" value="上一页"/>

</form>

</td>

<td>

<form action="" method=post>

<input type="hidden" name="currentPage" value="<%=dataBean.getCurrentPage()+1%>"/>

<input type="submit" name="g" value="下一页"/>

</form>

</td>

</tr>

<tr>

<td>

<form action="" method=post>

每页显示<input type="text" name="pageSize" value=2 size=3/>

条记录<input type="submit" name="g" value="确定"/>

</form>

<form action="" method=post>

输入页码：<input type="text" name="currentPage" size=2>

<input type="submit" name="g" value="提交"/>

</form>

</td>

</tr>

</table>

</div>

</body>

</html>

**-------------------------------byPageShow.jsp结束-----------------------------**



**------------------------------** showDetail.jsp**开始----------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<%@ page import="java.sql.\*"%>

<%@ page import="mybean.data.Login"%>

<jsp:useBean id="loginBean" class="mybean.data.Login" scope="session"/>

<jsp:useBean id="QueryDetailBean" class="mybean.data.QueryDetail" scope="session"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String picBasePath = path+"/images";

String putGoodsPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/PutGoodsToCar";

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>My JSP 'showDetail.jsp' starting page</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<body>

<%

if(loginBean==null){

response.sendRedirect(loginPath); //重定向到登录页

}else{

boolean b = loginBean.getLogname()==null||loginBean.getLogname().length()==0;

if(b){

response.sendRedirect(loginPath);

}

}

String mobileID = request.getParameter("detail");

QueryDetailBean.queryById(mobileID,putGoodsPath,picBasePath);

%>

<div align=center>

<jsp:getProperty name="QueryDetailBean" property="backnews"/>

</div>

</body>

</html>

**------------------------------** showDetail.jsp**结束------------------------**

模型（JavaBean）

打开mobileShop，找到src目录，在创建好的包mybean.data中新建Class文件命名为DataByPage，DataByPage.java是用于用于存储数据库中的记录。DataByPage.java文件的具体代码如下：

**-------------------------------DataByPage.java开始----------------------------**

package mybean.data;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import org.utils.JdbcUtil;

public class QueryDetail {

String backnews;

public String getBacknews() {

return backnews;

}

public void setBacknews(String backnews) {

this.backnews = backnews;

}

public void queryById(String mobileID,String putGoodsPath,String picBasePath) {

String detail;

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet rs = null;

detail = "<th>产品号"+mobileID;

if(mobileID==null){

detail = "没有产品号，无法查看细节";

return;

}

try {

connection = JdbcUtil.getConnection();

//3.获取statement

//String sql ="insert into user(logname,password,phone,address,realname) value(?,?,?,?,?) ";

String sql ="select \* from mobileForm where mobile\_version = '"+mobileID+"'";

prepareStatement = connection.prepareStatement(sql);

//4.执行sql

rs = prepareStatement.executeQuery();

detail = detail+"<table border=2>"

+ "<tr>"

+ "<th>产品号</th><th>名称</th><th>制造商</th><th>价格</th><th><font color=blue>放入购物车<font></th>"

+ "</tr>";

String picture = picBasePath+"welcome.jpg";

String detailMess = "";

//5.处理查询结果

while(rs.next()) {

String number = rs.getString(1);

String name = rs.getString(2);

String maker = rs.getString(3);

String price = rs.getString(4);

detailMess = rs.getString(5);

picture = picBasePath+"/"+rs.getString(6);

String goods = "("+number+","+name+","+maker+","+price+")#"+price; //便于购物车计算价格，尾缀上“#价格值”

goods = goods.replaceAll("\\p{Blank}","");

String button = "<form action = '"+putGoodsPath+"' method='post'>"+

"<input type='hidden' name='java' value="+goods+">"+

"<input type='submit' value='放入购物车'></form>";

detail = detail+"<tr><td>"+number+"</td>"

+ "<td>"+name+"</td>"

+ "<td>"+maker+"</td>"

+ "<td>"+price+"</td>"

+ "<td>"+button+"</td></tr>";

}

detail = detail +"</table>产品详情：<br>"

+ "<div align=center>"+detailMess+"</div>"

+ "<img src='"+picture+"' width=260></img>";

this.backnews = detail;

} catch (Exception e) {

detail=""+e.toString();

this.backnews = detail;

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtil.close(connection, prepareStatement, rs);

}

}

}

**------------------------------** **DataByPage.java结束----------------------------**

控制器（servlet）

打开mobileShop，找到src目录，在创建好的包myservlet.data下新建Class命名为QueryRecord、再新建一个Class命名为PutGoodsToCar (见配置文件web.xml)。 servlet控制器QueryRecord把从数据库 mobileform表中查询到的记录存到dataBean然后将用户重定向到 byPageshow.jsp页面，当用户在 byPageshow.jsp页面showDetal.jsp看到产品时，给每个产品都添加了一个“添加到购物车”按钮，用户单击该按钮后，PutGoodsToCar控制器将该产品放入用户的购物车。

**-------------------------------QueryRecord.java开始--------------------------**

package myservlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.utils.JdbcUtil;

import com.sun.rowset.CachedRowSetImpl;

import mybean.data.DataByPage;

/\*\*

\* Servlet implementation class QueryRecord

\*/

@WebServlet("/QueryRecord")

public class QueryRecord extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public QueryRecord() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

super.init(config);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

DataByPage dataBean = null;

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

if(classifyNum == null) {

classifyNum = "0";

}

int id = Integer.parseInt(classifyNum);

HttpSession session = request.getSession(true);

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

dataBean = (DataByPage)session.getAttribute("dataBean");

if(dataBean==null) {

dataBean = new DataByPage(); //创建dataBean对象

session.setAttribute("dataBean", dataBean);

}

}catch(Exception e) {

dataBean = new DataByPage();

session.setAttribute("dataBean", dataBean);

}

try {

connection = JdbcUtil.getConnection();

//3.获取statement

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery("select \* from mobileForm where id="+id);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

dataBean.setRowSet(rowSet); //行集数据存储在dataBean中

response.sendRedirect(request.getContextPath()+"/pages/byPageShow.jsp");

} catch (Exception e) {

out.print(e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection

JdbcUtil.statementClose(connection, sql, rs);

}

out.flush();

out.close();

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**-------------------------------QueryRecord.java结束---------------------------**

**-------------------------------PutGoodsToCar.java开始-------------------------**

package myservlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.LinkedList;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import mybean.data.Login;

/\*\*

\* Servlet implementation class PutGoodsToCar

\*/

@WebServlet("/PutGoodsToCar")

public class PutGoodsToCar extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

super.init(config);

}

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public PutGoodsToCar() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

Login loginBean = null;

HttpSession session = request.getSession(true);

String goods = request.getParameter("java");

try {

loginBean = (Login)session.getAttribute("loginBean");

boolean b = loginBean.getLogname() == null||loginBean.getLogname().length()==0;

if(b) {

response.sendRedirect(request.getContextPath()+"/pages/login.jsp");

}

LinkedList<String> car = loginBean.getCar();

car.add(goods); //将产品放入购物车

speakSomeMess(request,response,goods);

}catch(Exception e) {

}

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

protected void speakSomeMess(HttpServletRequest request, HttpServletResponse response, String goods) throws ServletException, IOException {

// TODO Auto-generated method stub

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

//String headTxtPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+request.getContextPath()+"/WEB-INF/tags/tophead.txt";

String lookShoppingCar = request.getContextPath()+"/pages/lookShoppingCar.jsp";

String indexPath = request.getContextPath()+"/index.jsp";

try {

PrintWriter out = response.getWriter();

/\* out.print("<head><%@ include file='"+headTxtPath+"'%></head>");\*/

out.println("<html><body><div align=center>");

out.println("<h2>"+goods+"放入购物车</h2>");

out.println("查看购物车或返回<br>");

out.println("<a href='"+lookShoppingCar+"'>查看购物车</a>");

out.println("<br><a href='"+indexPath+"'>主页</a>");

out.println("</div></body></html>");

}catch(IOException e) {

e.printStackTrace();

}

}

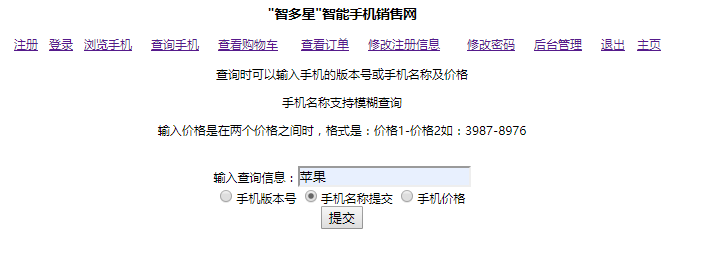
}

**------------------------------PutGoodsToCar.java结束--------------------------**

查询手机

视图（JSP页面）

该模块视图部分由两个个JSP页面构成，其中byPageShow.jsp在5-1-1中部分有详细讲解，另一个是searchMobile.jsp,这个jsp页面输入查询信息，提交到searchByCondition控制器，该控制器将查询结果放到dataBean中，byPageShow.jsp页面负责显示dataBean中的数据。在WebRoot下找到建好的目录pages,在目录pages下 新建Jsp文件命名为searchMobile.jsp，如下图可见searchMobile.jsp和byPageShow.jsp效果;





**---------------------------- searchMobile.jsp开始---------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String searchByConditioPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/searchByCondition";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>My JSP 'searchMobile.jsp' starting page</title>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

</head>

<body>

<div align=center>

<p>查询时可以输入手机的版本号或手机名称及价格</p>

<p>手机名称支持模糊查询</p>

<p>输入价格是在两个价格之间时，格式是：价格1-价格2如：3987-8976</p>

<form action = "<%= searchByConditioPath %>">

<BR>输入查询信息：<input type=text name="searchMess"><br>

<input type=radio name ="radio" value = "mobile\_version">手机版本号

<input type=radio name ="radio" value = "mobile\_name">手机名称提交

<input type=radio name ="radio" value = "mobile\_price">手机价格

<br><input type=submit name="g" value="提交">

</form>

</div>

</body>

</html>

**-----------------------------searchMobile.jsp结束----------------------------**

模型（JavaBean）

在该模块用到的是已经创建好的mybean.data包下的DataByPage.java，与浏览手机使用的是同一个bean（详情见5-1-2）：

控制器（servlet）

打开mobileShop，找到src目录，在创建好的包myservlet.data下新建Class命名为searchByCondition (见配置文件web.xml)，该文件把从数据库表mobileform中查询到的记录存到dataBean中，然后将用户重定向到byPageShow.jsp：

**---------------------------searchByCondition.java开始-------------------------**

package myservlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.utils.JdbcUtil;

import com.sun.rowset.CachedRowSetImpl;

import mybean.data.DataByPage;

/\*\*

\* Servlet implementation class searchByCondition

\*/

@WebServlet("/searchByCondition")

public class searchByCondition extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null; //存储表中全部记录的行集对象

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public searchByCondition() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

super.init(config);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

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

String radioMess = request.getParameter("radio");

if(searchMess == null||searchMess.length()==0) {

fail(request,response,"没有填写查询信息，无法查询");

return;

}

String condition = "";

if(radioMess.equals("monile\_version")) {

condition = "select \* from mobileForm where mobile\_version='"+searchMess+"'";

}else if(radioMess.equals("mobile\_name")) {

condition = "select \* from mobileForm where mobile\_name like '%"+searchMess+"%'";

}else if(radioMess.equals("mobile\_price")) {

double max = 0,min =0;

String regex = "[^0123456789.]";

String[] priceMess = searchMess.split(regex);

if(priceMess.length == 1) {

max = min =Double.parseDouble(priceMess[0]);

}else if(priceMess.length ==2) {

min = Double.parseDouble(priceMess[0]);

max = Double.parseDouble(priceMess[1]);

if(max<min) {

double t = max;

max = min;

min = t;

}

}else {

fail(request,response,"输入价格格式有误");

return;

}

condition = "select \* from mobileForm where mobile\_price <="+max+" and mobile\_price>="+min;

}

HttpSession session = request.getSession(true);

DataByPage dataBean = null;

try {

dataBean = (DataByPage)session.getAttribute("dataBean");

if(dataBean == null) {

dataBean = new DataByPage();

session.setAttribute("dataBean", dataBean);

}

}catch(Exception e) {

dataBean = new DataByPage();

session.setAttribute("dataBean", dataBean);

}

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

connection = JdbcUtil.getConnection();

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery(condition);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

dataBean.setRowSet(rowSet);

out.println(condition+"条件"+rowSet);

response.sendRedirect(request.getContextPath()+"/pages/byPageShow.jsp"); //重定向到结果显示页

} catch (Exception e) {

out.println(condition+"条件"+e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtil.statementClose(connection, sql, rs);

}

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

public void fail(HttpServletRequest request, HttpServletResponse response,String backnews) {

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

try {

String searchMobilePath = request.getContextPath()+"/pages/searchMobile.jsp";

PrintWriter out = response.getWriter();

out.println("<html><body><div align=center>");

out.println("<h2>"+backnews+"</h2>");

out.println("返回主页");

out.println("<a href='"+searchMobilePath+"'>查询手机</a>");

out.println("</div></body></html>");

}catch(IOException e) {

}

}

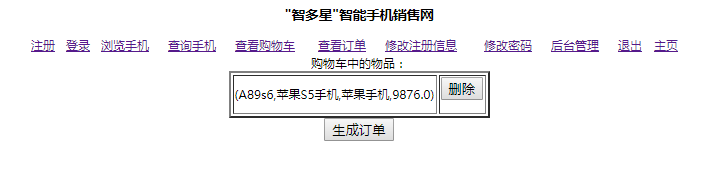
}

**---------------------------searchByCondition.java结束-------------------------**

查看购物车

视图（JSP页面）

该模块视图部分由一个JSP页面构成，这个jsp页面负责显示和选择要删除的物品，即显示JavaBean模型carBean中的数据和删除购物车中的物品，并允许用户生产订单。在WebRoot下找到建好的目录pages,在目录pages下 新建Jsp文件命名为lookShoppingCar.jsp效果如图所示;



**----------------------------lookShoppingCar**.jsp**开始-------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<%@ page import="java.util.\*"%>

<%@ page import="mybean.data.Login"%>

<jsp:useBean id="loginBean" class="mybean.data.Login" scope="session"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String deleteServletPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleDelete";

String buyServletPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleBuyGoods";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>My JSP 'lookShooppingCar.jsp' starting page</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<body>

<div align=center>

<%

if(loginBean==null){

response.sendRedirect(loginPath); //重定向到登录页

}else{

boolean b = loginBean.getLogname()==null||loginBean.getLogname().length()==0;

if(b){

response.sendRedirect(loginPath);

}

}

LinkedList<String> car = loginBean.getCar();

if(car == null){

out.print("<h2>购物车没有物品</h2>");

}else{

Iterator<String> iterator = car.iterator(); //迭代器，调用集合的方法iterator获取出，Iteator接口实现类的对象

StringBuffer buyGoods = new StringBuffer();

int n=0;

double priceSum = 0;

out.print("购物车中的物品：<table border=2>");

while(iterator.hasNext()){

String goods = iterator.next();

String showGoods = "";

n++;

//购物车物品的后缀是“#价格数字”

int index = goods.lastIndexOf("#");

if(index!=-1){

priceSum += Double.parseDouble(goods.substring(index+1));

showGoods = goods.substring(0,index);

}

buyGoods.append(n+":"+showGoods);

String del ="<form action = '"+deleteServletPath+"' method='post'>"+

"<input type='hidden' name='delete' value="+goods+">"+

"<input type='submit' value='删除'></form>";

out.print("<tr><td>"+showGoods+"</td>");

out.print("<td>"+del+"</td></tr>");

}

out.print("</table>");

String orderForm = "<form action = '"+buyServletPath+"' method='post'>"+

"<input type='hidden' name='buy' value="+buyGoods+">"+

"<input type='hidden' name='price' value="+priceSum+">"+

"<input type='submit' value='生成订单'></form>";

out.print(orderForm);

}

%>

</div>

</body>

</html>

**----------------------------lookShoppingCar.jsp结束-------------------------**

模型（JavaBean）

查看购物车信息这块使用的模型是已经创建好的mybean.data包下的 Login.java，与用户登录使用的是同一个bean（详情见4-2-2）

控制器（servlet）

打开mobileShop，找到src目录，在创建好的包myservlet.data下新建Class命名为HandleDelete、再新建Class命名为HandleBuyGoods (见配置文件web.xml)。用户查看购物车后，会看到购物车中的物品，并且每个物品都后缀一个删除按钮，用户单击这个删除按钮，HandleDelete控制器从购物车中删除这件物品。用户在视图部分单击“生成订单”， HandleDelete控制器负责把用户购物车中的物品存放到数据库的orderform表中，即生成订单，然后清空用户的购物车：

**-------------------------------HandleDelete.java开始--------------------------**

package myservlet.data;

import java.io.IOException;

import java.util.LinkedList;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import mybean.data.Login;

/\*\*

\* Servlet implementation class HandleDelete

\*/

@WebServlet("/HandleDelete")

public class HandleDelete extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public HandleDelete() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

/\* PrintWriter out = response.getWriter();\*/

Login loginBean = null;

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

HttpSession session = request.getSession(true);

try {

loginBean = (Login)session.getAttribute("loginBean");

boolean b = loginBean.getLogname()==null||loginBean.getLogname().length() == 0;

if(b) {

response.sendRedirect(request.getContextPath()+"/pages/login.jsp");

}

LinkedList<String> car = loginBean.getCar();

car.remove(delete); //从购物车中删除产品

}catch(Exception e) {

response.sendRedirect(request.getContextPath()+"/pages/login.jsp");

}

RequestDispatcher dispatcher = request.getRequestDispatcher("/pages/lookShoppingCar.jsp");

dispatcher.forward(request, response); //转发

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**-------------------------------HandleDelete.java结束-------------------------**

**--------------------------- HandleBuyGoods.java开始---------------------------**

package myservlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.LinkedList;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.utils.JdbcUtil;

import mybean.data.Login;

/\*\*

\* Servlet implementation class HandleBuyGoods

\*/

@WebServlet("/HandleBuyGoods")

public class HandleBuyGoods extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public HandleBuyGoods() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

super.init(config);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

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

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String buyGoodsMess = request.getParameter("buy");

if(buyGoodsMess == null||buyGoodsMess.length()==0) {

fail(request,response,"购物车没有物品，无法生成订单");

return;

}

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

if(price == null||price.length()==0) {

fail(request,response,"没有计算价格和，无法生成订单");

return;

}

float sum = Float.parseFloat(price);

Login loginBean = null;

HttpSession session = request.getSession(true);

try {

loginBean = (Login)session.getAttribute("loginBean");

boolean b = loginBean.getLogname() == null||loginBean.getLogname().length()==0;

if(b) {

response.sendRedirect(request.getContextPath()+"/pages/login.jsp");

}

}catch(Exception e) {

response.sendRedirect(request.getContextPath()+"/pages/login.jsp");

}

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet rs = null;

String backnews="";

try {

connection = JdbcUtil.getConnection();

//3.获取statement

//String sql ="insert into user(logname,password,phone,address,realname) value(?,?,?,?,?) ";

String sql ="insert into orderform values(?,?,?,?) ";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, 0); //订单号自定义增加

prepareStatement.setString(2, loginBean.getLogname());

prepareStatement.setString(3, buyGoodsMess);

prepareStatement.setFloat(4, sum);

//4.执行sql

int m = prepareStatement.executeUpdate();

if(m!=0) {

LinkedList<String> car = loginBean.getCar();

car.clear(); //清空购物车

success(request,response,"生成订单成功");

}else {

fail(request,response,"生成订单失败");

}

} catch (Exception e) {

backnews=""+e.toString();

out.print(e.toString());

e.printStackTrace();

fail(request,response,"生成订单失败"+backnews);

}finally {

//5.释放资源 connection prepareStatement

JdbcUtil.close(connection, prepareStatement, rs);

}

out.flush();

out.close();

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

public void success(HttpServletRequest request, HttpServletResponse response,String backnews) {

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

try {

String lookOrderFormPath = request.getContextPath()+"/pages/lookOrderForm.jsp";

String registerPath = request.getContextPath()+"/index.jsp";

PrintWriter out = response.getWriter();

out.println("<html><body><div align=center>");

out.println("<h2>"+backnews+"</h2>");

out.println("返回主页");

out.println("<a href='"+registerPath+"'>主页</a>");

out.println("<br>查看订单");

out.println("<a href='"+lookOrderFormPath+"'>查看订单</a>");

out.println("</div></body></html>");

}catch(Exception e) {

}

}

public void fail(HttpServletRequest request, HttpServletResponse response,String backnews) {

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

try {

String registerPath = request.getContextPath()+"/index.jsp";

PrintWriter out = response.getWriter();

out.println("<html><body><div align=center>");

out.println("<h2>"+backnews+"</h2>");

out.println("返回主页");

out.println("<a href='"+registerPath+"'>主页</a>");

out.println("</div></body></html>");

}catch(IOException e) {

}

}

}

**----------------------------HandleBuyGoods.java结束---------------------------**

查看订单

视图（JSP页面）

该模块视图部分由一个JSP页面构成，这个jsp页面负责显示用户订单信息 。在WebRoot下找到建好的目录pages,在目录pages下 新建Jsp文件命名为lookOrderForm.jsp效果如图所示;

**------------------------------lookOrderForm**.jsp**开始---------------------------**

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<%@ page import="java.sql.\*"%>

<%@ page import="mybean.data.Login"%>

<jsp:useBean id="loginBean" class="mybean.data.Login" scope="session"/>

<jsp:useBean id="orderBean" class="mybean.data.QueryOrderForm" scope="session"/>

<%

String path = request.getContextPath();

String registerPath =path+"/pages/inputRegister.jsp";

String loginPath = path+"/pages/login.jsp";

String indexPath = path+"/index.jsp";

String classifyPath = path+"/pages/lookMobile.jsp";

String ShoppingCarPath = path+"/pages/lookShoppingCar.jsp";

String serchMobilePath = path+"/pages/searchMobile.jsp";

String lookOrderFormPath = path+"/pages/lookOrderForm.jsp";

String changePassPath = path+"/pages/changePassword.jsp";

String changeRegisterPath = path+"/pages/changeRegisterInfo.jsp";

String exitPath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/servlet/HandleExit";

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<%@ include file="/WEB-INF/tags/tophead.txt" %>

<title>My JSP 'lookOrderForm.jsp' starting page</title>

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="This is my page">

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<body>

<%

if(loginBean==null){

response.sendRedirect(loginPath); //重定向到登录页

}else{

boolean b = loginBean.getLogname()==null||loginBean.getLogname().length()==0;

if(b){

response.sendRedirect(loginPath);

}

}

orderBean.queryOrder(loginBean.getLogname());

%>

<div align=center>

<jsp:getProperty name="orderBean" property="backnews"/>

</div>

</body>

</html>

**-------------------------------lookOrderForm.**jsp**结束--------------------------**

模型（JavaBean）

查看订单信息这块使用的模型是已经创建好的mybean.data包下的 Login.java，与用户登录使用的是同一个bean（详情见4-2-2）

控制器（servlet）

本模块无控制器

后台管理

管理员登录

视图（JSP页面）

该模块视图部分由一个JSP页面构成，这个jsp页面负责提交用户的登录信息到 servlet控制器ManagementLogin (见配置文件web.xml) 。首先在WebRoot下新建目录management,在目录management下 新建Jsp文件命名为login.jsp效果如图所示。



**---------------------------------login.jsp开始--------------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>用户登录</title>

</head>

<style>

body {

background-color:#EEFFFF;

padding-top:100px;

}

.together {

width: 700px;

margin: 0px auto;

background-color: #FFF;

border: 1px solid #ccc;

border-radius: 5px;

}

.together .choose {

width: 100%;

/\*height: 300px;\*/

margin: 10px auto;

text-align: center;

height: 50px;

line-height: 50px;

}

ul {

margin: 0;

padding: 50px;

padding-top: 0;

}

li {

display: flex;

margin: 20px 0;

}

li span {

min-width: 200px;

font-size: 12px;

line-height: 46px;

float: left;

text-align: center;

opacity: 0.3

}

input,

label {

display: block;

float: left;

height: 46px;

font-size: 23px;

box-sizing: border-box;

color: #333;

}

label {

width: 200px;

line-height: 46px;

margin-right: 30px;

text-align: right;

font-weight: 700;

}

input {

width: 320px;

padding: 8px;

border: 1px solid #cccccc;

outline: none;

/\*text-align: center;\*/

}

input.code {

width: 120px;

}

input.verify {

width: 190px;

margin-left: 10px;

background-color: red;

color: #fff;

text-align: center;

}

input[type=submit] {

border: none;

color: #fff;

background-color:#00FF66;

border: 4px;

cursor: pointer;

}

.tips {

/\*固定定位\*/

position: fixed;

top: 0;

width: 100%;

height: 40px;

text-align: center;

display: none;

}

.tips p {

min-width: 300px;

max-width: 400px;

margin: 0 auto;

line-height: 40px;

color: #fff;

background-color: #C91623;

}

</style>

<body>

<div class="together">

<div class="choose">

<h1>后台管理登录</h1>

</div>

<!-- 登录 -->

<div class="login">

<form action="${pageContext.request.contextPath }/servlet/ManagementLogin" method="post">

<ul>

<li>

<label for="username">用户名</label>

<input type="text" name="phone" id="username" placeholder="请输入手机号" class="username">

</li>

<li>

<label for="psw">密码</label>

<input type="password" name="password" placeholder="请输入密码" id="psw" class="psw">

</li>

<li>

<label for=""></label>

<input type="submit" class="loginBtn" value="登录">

</li>

</ul>

</form>

</div>

</div>

</body>

</html>

**---------------------------------login.jsp结束---------------------------**

模型（JavaBean）

打开mobileShop，找到src目录，右击选择New，点击Package，在Name中填写management.bean，点击finish。在创建好的包management.bean中新建Class文件命名为User，User.java是对用户信息的设置和获取。User.java文件的具体代码如下：

**-----------------------------------User.java开始-----------------------------**

package management.bean;

public class User {

String logname;

String phone;

String address;

String realname;

String password;

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public User() {

}

public User(String name, String phone, String address,String realname) {

this.logname = name;

this.phone = phone;

this.address = address;

this.realname = realname;

}

public User(String name, String phone, String address,String realname, String password) {

this.logname = name;

this.phone = phone;

this.address = address;

this.realname = realname;

this.password = password;

}

public User(String name,String realname) {

this.logname = name;

this.realname = realname;

}

public String getLogname() {

return logname;

}

public void setLogname(String logname) {

this.logname = logname;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getRealname() {

return realname;

}

public void setRealname(String realname) {

this.realname = realname;

}

}

**----------------------------------**-**User.java结束-----------------------------**

控制器（servlet）

打开mobileShop，找到src目录，右击选择New，点击Package，在Name中填写management.servlet，点击finish，完成包的创建。在创建好的包management.servlet下新建Class命名为ManagementLogin (见配置文件web.xml)，该文件负责连接数据库将用户提交的信息与management表中的用户信息对比。ManagementLogin.java文件内容如下：

**---------------------------** ManagementLogin**.java开始--------------------------**

package management.servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import management.service.UserServiceImpl;

import management.bean.User;

/\*\*

\* Servlet implementation class ManagementLogin

\*/

@WebServlet("/ManagementLogin")

public class ManagementLogin extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ManagementLogin() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

HttpSession session = request.getSession(true);

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

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

UserServiceImpl userService = new UserServiceImpl();

User user = userService.queryManageByAccountAndPassword(phone,password);

if(user==null) {

response.sendRedirect(request.getContextPath()+"/management/login.jsp");

return ;

}else {

session.setAttribute("manageBean", user);

}

/\*\*

\*

\*/

request.getServletContext().setAttribute("user", user);

//重定向到首页

response.sendRedirect(request.getContextPath()+"/management/manageIndex.jsp");

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**---------------------------** ManagementLogin**.java结束--------------------------**

后台管理首页

视图（JSP页面）

该模块视图部分由两个JSP页面构成，其中manageFlash.jsp是作为后台管理系统的欢迎页面，只是包含一张图；另一个是manageIndex.jsp,这个jsp页面作为后台管理系统的额首页，左边是树形菜单，右边为对应显示的管理员操作页。在WebRoot下找到建好的目录management,在目录management下 新建Jsp文件命名为manageFlash、同样再新建一个Jsp文件命名为manageIndex。如下图可见manageFlash.jsp和manageIndex.jsp效果;

模型（JavaBean）

该模块没有用到模型

控制器（servlet）

该模块没有用到控制器

用户管理

视图（JSP页面）

该模块视图部分由两个JSP页面构成，userList.jsp负责显示用户表user中的信息，实现查询和删除用户的功能；editUser.jsp负责显示指定用户的信息，实现增加和编辑用户信息的功能。首先在WebRoot目录下找到建好management,在目录management下 新建user目录，在user目录下新建Jsp文件命名为userList；同理再新建Jsp文件且命名为editUser。页面效果如图所示。



**--------------------------------userList.jsp开始---------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>

<html lang="zh-cn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>用户列表</title>

<!-- Bootstrap -->

<link href="${pageContext.request.contextPath }/assets/cssLibs/bootstrap.min.css" rel="stylesheet">

<link href="${pageContext.request.contextPath }/assets/css/style.css" rel="stylesheet">

<script src="${pageContext.request.contextPath }/assets/JsLibs/jquery-3.3.1.min.js"></script>

<script src="${pageContext.request.contextPath }/assets/JsLibs/bootstrap.min.js"></script>

</head>

<body>

<!--路径导航-->

<ol class="breadcrumb breadcrumb\_nav">

<li>首页</li>

<li>用户管理</li>

<li class="active">用户列表</li>

</ol>

<!--路径导航-->

<div class="page-content">

<div class="panel panel-default">

<div class="panel-heading">用户列表</div>

<div class="panel-body">

<div class="pull-right">

<div class="form-group">

<label class="sr-only" for="exampleInputEmail3">用户名</label>

<input type="email" class="form-control" id="userName" placeholder="请输入用户名">

</div>

<button type="button" class="btn btn-default" onclick="queryByPages()" >查询</button>

<a href="${pageContext.request.contextPath }/servlet/UserManage?method=editUI" class="btn btn-default" >添加</a>

</div>

</div>

<div class="table-responsive">

<table class="table table-striped table-hover">

<thead>

<tr>

<th style='width="3%"'>

<!-- <input class="check\_all" type="checkbox"> -->

</th>

<th style='width:20%'>昵称</th>

<th style='width:20%'>手机号</th>

<th style='width:20%'>邮寄地址</th>

<th style='width:20%'>真实姓名</th>

</tr>

</thead>

<tbody>

<c:forEach items="${pageResult.list}" var="item">

<tr>

<td>

<!-- <input class="check\_all" type="checkbox" name="ids[]" value="8"> -->

</td>

<td>${item.logname }</td>

<td>${item.phone}</td>

<td>${item.address}</td>

<td>${item.realname}</td>

<td>

<a href="${pageContext.request.contextPath }/servlet/UserManage?method=editUI&logname=${item.logname}" class="btn btn-info btn-xs"><span class="glyphicon glyphicon-edit"></span>编辑</a>

<a href="${pageContext.request.contextPath }/servlet/UserManage?method=delete&logname=${item.logname}" class="btn btn-warning btn-xs"><span class="glyphicon glyphicon-remove"></span>删除</a>

</td>

</tr>

</c:forEach>

</tbody>

</table>

<div class="panel-footer clearfix">

<div class="pull-left">

<div class="form-group">

<label>总记录数：${pageResult.totalRecord}</label>

<label>每页显示：</label>

<select class="form-control" id="pageSize" onchange="queryByPages()">

<option value="5" <c:if test="${pageResult.pageSize eq 5}">selected="selected"</c:if> >5</option>

<option value="10" <c:if test="${pageResult.pageSize eq 10}">selected="selected"</c:if> >10</option>

<option value="15" <c:if test="${pageResult.pageSize eq 15}">selected="selected"</c:if> >15</option>

</select>

</div>

</div>

<nav class="pull-right">

<ul class="pagination pagination-self">

<!-- 上一页 -->

<c:if test="${pageResult.currentPage == 1}">

<li>

<span aria-hidden="true">&laquo;</span>

</li>

</c:if>

<c:if test="${pageResult.currentPage >1}">

<li>

<a href="${pageContext.request.contextPath }/servlet/UserManage?method=list&currentPage=${pageResult.currentPage-1}" aria-label="Previous">

<span aria-hidden="true">&laquo;</span>

</a>

</li>

</c:if>

<c:forEach begin="${pageResult.beginIndex }" end="${pageResult.endIndex }" varStatus="status">

<c:if test="${status.index eq pageResult.currentPage}">

<li><span style="color: black;">${status.index }</span></li>

</c:if>

<c:if test="${status.index ne pageResult.currentPage}">

<li><a href="${pageContext.request.contextPath }/servlet/UserManage?method=list&currentPage=${status.index}">${status.index }</a></li>

</c:if>

</c:forEach>

<!-- 下一页 -->

<c:if test="${pageResult.currentPage == pageResult.totalPages}">

<li>

<span aria-hidden="true">&raquo;</span>

</li>

</c:if>

<c:if test="${pageResult.currentPage < pageResult.totalPages}">

<li>

<a href="${pageContext.request.contextPath }/servlet/UserManage?method=list&currentPage=${pageResult.currentPage+1}" aria-label="Next">

<span aria-hidden="true">&raquo;</span>

</a>

</li>

</c:if>

</ul>

</nav>

</div>

</div>

</div>

</div>

<script type="text/javascript">

function queryByPages(){

//设置当前浏览器的地址

var pageSize = document.getElementById("pageSize").value;

var userName = document.getElementById("userName").value;

document.location.href="${pageContext.request.contextPath }/servlet/UserManage?method=list&currentPage=1&pageSize="+pageSize+"&userName="+userName;

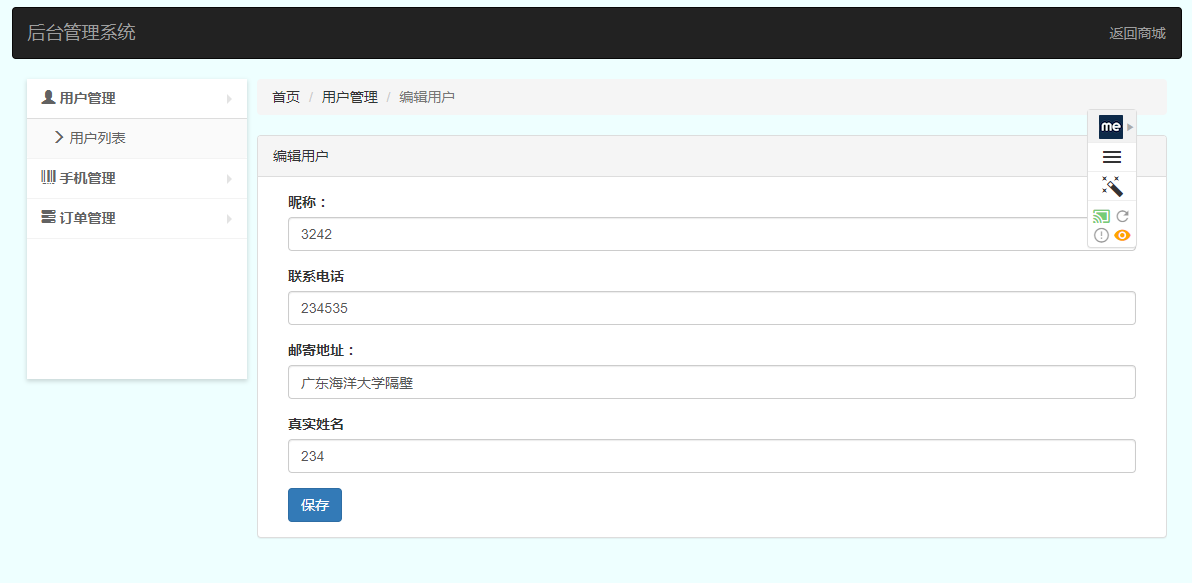
}

</script>

</body>

</html>

**--------------------------------userList.jsp结束---------------------------**



**--------------------------------editUser.jsp开始---------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>

<html lang="zh-cn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="content-type" content = "text/html;charset=utf-8">

<title>添加用户</title>

<!-- Bootstrap -->

<link href="${pageContext.request.contextPath }/assets/cssLibs/bootstrap.min.css" rel="stylesheet">

<link href="${pageContext.request.contextPath }/assets/css/style.css" rel="stylesheet">

<script src="${pageContext.request.contextPath }/assets/JsLibs/jquery-3.3.1.min.js"></script>

<script src="${pageContext.request.contextPath }/assets/JsLibs/bootstrap.min.js"></script>

</head>

<body>

<!--路径导航-->

<ol class="breadcrumb breadcrumb\_nav">

<li>首页</li>

<li>用户管理</li>

<li class="active">编辑用户</li>

</ol>

<!--路径导航-->

<div class="page-content">

<div class="panel panel-default">

<div class="panel-heading">编辑用户</div>

<div class="panel-body">

<form action="${pageContext.request.contextPath }/servlet/UserManage?method=edit" method="post">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="form-group clearfix">

<label>昵称：</label>

<input type="text" class="form-control" id="name" name="logname" value="${user.logname }" placeholder="请输入手机号">

</div>

<div class="form-group clearfix">

<label>联系电话</label>

<input type="text" class="form-control" id="phone" name="phone" value="${user.phone }" placeholder="请输入手机号">

</div>

</div>

<div class="col-md-12">

<div class="form-group clearfix">

<label>邮寄地址：</label>

<input type="text" class="form-control" id="name" name="address" value="${user.address }" placeholder="请输入手机号">

</div>

<div class="form-group clearfix">

<label>真实姓名</label>

<input type="text" class="form-control" id="phone" name="realname" value="${user.realname }" placeholder="请输入手机号">

</div>

</div>

<c:if test="${user.logname==null||user.logname.length()==0}">

<div class="col-md-12">

<div class="form-group clearfix">

<label>密码：</label>

<input type="text" class="form-control" id="password" name="password" placeholder="密码">

</div>

</div>

</c:if>

<input type="hidden" name="isUpdate" value="${user.logname }"/>

</div>

<button type="submit" class="btn btn-primary">保存</button>

</div>

</form>

</div>

</div>

</div>

<%-- <script src="${pageContext.request.contextPath }/assets/js/pcasunzip.js"></script> --%>

<!-- <script type="text/javascript">

new PCAS("province","city",$("#provinceId").val(),$("#cityId").val());

</script> -->

</body>

</html>

**--------------------------------editUser.jsp结束---------------------------**

模型（JavaBean）

该模块使用到的bean文件为User.java和管理员登录使用到的是同一个bean文件，详情见6-1-2。

控制器（servlet）

打开mobileShop，在src目录，右击选择New，点击Package，在Name中填写management.service，完成包的创建；重复上述的步骤，在src目录，右击选择New，点击Package，在Name中填写management.databaseImp。

在创建好的包management.databaseImp下新建Class命名为UserDatabaseImpl；在创建好的包management.service下新建Class命名为UserServiceImpl；在创建好的包management.servlet创建Class文件命名为UserManage，文件内容如下：

**--------------------------**UserDatabaseImpl**.java开始---------------------------**

package management.databaseImp;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import management.bean.User;

import management.tool.PageResult;

import org.utils.JdbcUtil;

public class UserDatabaseImpl {

/\*\*

\* 注册用户

\*/

public void add(User user) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="insert into user(logname, password, phone, address, realname) value(?,?,?,?,?) ";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, user.getLogname());

prepareStatement.setString(2, user.getPassword());

prepareStatement.setString(3, user.getPhone());

prepareStatement.setString(4, user.getAddress());

prepareStatement.setString(5, user.getRealname());

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 通过用户名模糊查询

\*/

public User queryByLogname(String name) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

User user = null;

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from user where logname=? ";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, name);

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

String logname = resultSet.getString("logname");

String phone = resultSet.getString("phone");

String address = resultSet.getString("address");

String realname = resultSet.getString("realname");

user = new User(logname,phone,address,realname);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return user;

}

/\*\*

\* 查询表中所用的用户

\*/

public List<User> queryAll() {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

List<User> userList = new ArrayList<>();

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from user";

prepareStatement = connection.prepareStatement(sql);

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

String logname = resultSet.getString("logname");

String realname = resultSet.getString("realname");

User user = new User(logname,realname);

userList.add(user);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return userList;

}

/\*\*

\* 修改用户信息

\*/

public void update(User user) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="update user set logname=?,phone=?,address=?,realname=? where logname=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, user.getLogname());

prepareStatement.setString(2, user.getPhone());

prepareStatement.setString(3, user.getAddress());

prepareStatement.setString(4, user.getRealname());

prepareStatement.setString(5, user.getLogname());

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 删除用户

\*/

public void delete(String name) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="delete from user where logname=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, name);

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 分页查询用户

\*/

public PageResult queryByPages(Integer currentPage, Integer pageSize, String userName) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

PreparedStatement prepareStatement2 = null;

ResultSet resultSet2 = null;

PageResult pageResult = null;

try {

connection = JdbcUtil.getConnection();

String whereSql = "";

if(userName!=null && userName.trim().length()>0) {

whereSql = "where logname like '%"+userName+"%' ";

}

String sql ="select \* from user "+whereSql+" limit ?,?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, (currentPage-1)\*pageSize);

prepareStatement.setInt(2, pageSize);

resultSet = prepareStatement.executeQuery();

List<User> userList = new ArrayList<>();

while(resultSet.next()) {

String logname = resultSet.getString("logname");

String phone = resultSet.getString("phone");

String address = resultSet.getString("address");

String realname = resultSet.getString("realname");

User user = new User(logname,phone,address,realname);

userList.add(user);

}

//查询表中用户总数

String sql2 ="select \* from user "+whereSql;

prepareStatement2 = connection.prepareStatement(sql2);

resultSet2 = prepareStatement2.executeQuery();

Long totalRecord = 0L;

while(resultSet2.next()) {

totalRecord = totalRecord + 1;

}

pageResult = new PageResult(userList, totalRecord, pageSize, currentPage);

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

JdbcUtil.close(connection, prepareStatement2, resultSet2);

}

return pageResult;

}

/\*\*

\* 号码和密码查找用户

\*/

public User queryByAccountAndPassword(String phone, String password) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

User user = null;

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from user where phone=? and password=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, phone.trim());

prepareStatement.setString(2, password.trim());

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

String logname = resultSet.getString("logname");

String phoneStr = resultSet.getString("phone");

String address = resultSet.getString("address");

String realname = resultSet.getString("realname");

user = new User(logname,phoneStr,address,realname);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return user;

}

/\*\*

\* 号码和密码查找管理员

\*/

public User queryManageByAccountAndPassword(String phone, String password) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

User user = null;

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from management where phone=? and password=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, phone.trim());

prepareStatement.setString(2, password.trim());

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

String logname = resultSet.getString("logname");

String phoneStr = resultSet.getString("phone");

String address = resultSet.getString("address");

String realname = resultSet.getString("realname");

user = new User(logname,phoneStr,address,realname);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return user;

}

}

**-------------------------**UserDatabaseImpl.**java结束---------------------------**

**---------------------------**UserServiceImpl**.java开始---------------------------**

package management.service;

import java.util.List;

import management.bean.User;

import management.tool.PageResult;

import management.databaseImp.UserDatabaseImpl;

public class UserServiceImpl{

public List<User> queryAll() {

UserDatabaseImpl userDao = new UserDatabaseImpl();

return userDao.queryAll();

}

public void add(User user) {

UserDatabaseImpl userDao = new UserDatabaseImpl();

userDao.add(user);

}

public void delete(String logname) {

UserDatabaseImpl userDao = new UserDatabaseImpl();

userDao.delete(logname);

}

public PageResult queryByPages(Integer currentPage, Integer pageSize, String userName) {

UserDatabaseImpl userDao = new UserDatabaseImpl();

return userDao.queryByPages(currentPage, pageSize, userName);

}

public User queryByLogname(String name) {

UserDatabaseImpl userDao = new UserDatabaseImpl();

return userDao.queryByLogname(name);

}

public void update(User user) {

UserDatabaseImpl userDao = new UserDatabaseImpl();

userDao.update(user);

}

public User queryByAccountAndPassword(String phone, String password) {

UserDatabaseImpl userDao = new UserDatabaseImpl();

return userDao.queryByAccountAndPassword(phone,password);

}

public User queryManageByAccountAndPassword(String phone, String password) {

UserDatabaseImpl userDao = new UserDatabaseImpl();

return userDao.queryManageByAccountAndPassword(phone,password);

}

}

**---------------------------**UserServiceImpl.**java结束---------------------------**

**-----------------------------**UserManage**.java开始------------------------------**

package management.servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import management.bean.User;

import management.tool.PageResult;

import management.service.UserServiceImpl;

/\*\*

\* Servlet implementation class UserManage

\*/

@WebServlet("/UserManage")

public class UserManage extends HttpServlet {

private static final long serialVersionUID = 1L;

String userListUrl = "/management/user/userList.jsp";

String editUserUrl = "/management/user/editUser.jsp";

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("UTF-8");

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

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

switch (method) {

case "list":

list(request, response);

break;

case "editUI":

editUI(request, response);

break;

case "edit":

edit(request, response);

break;

case "delete":

delete(request, response);

break;

default:

break;

}

}

/\*\*

\* 删除用户

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

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

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

UserServiceImpl userService = new UserServiceImpl();

userService.delete(name);

list(request, response);

}

/\*\*

\* 修改信息或者注册用户

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void edit(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String isUpdate = request.getParameter("isUpdate"); //用于判断是添加用户还是编辑用户

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

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

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

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

String password = request.getParameter("password"); //当添加用户时

UserServiceImpl userService = new UserServiceImpl();

if(isUpdate!=null&&isUpdate.trim().length()>0) {

User user = new User(logname, phone, address,realname);

userService.update(user);

}else {

User user = new User(logname, phone, address,realname,password);

userService.add(user);

}

list(request, response);

}

/\*\*

\* 根据用户名查找用户

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void editUI(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

if(logname!=null&&logname.trim().length()>0) {

UserServiceImpl userService = new UserServiceImpl();

User user = userService.queryByLogname(logname);

request.setAttribute("user", user);

}

request.getRequestDispatcher(editUserUrl).forward(request, response);

}

/\*\*

\* 列出用户表中的信息

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void list(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

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

Integer pageSizeInt = 5;

Integer currentPageInt = 1;

if(pageSize!=null&&pageSize.length()>0) {

pageSizeInt = Integer.parseInt(pageSize);

}

if(currentPage!=null&&currentPage.length()>0) {

currentPageInt = Integer.parseInt(currentPage);

}

UserServiceImpl userService = new UserServiceImpl();

PageResult pageResult = userService.queryByPages(currentPageInt,pageSizeInt, userName);

request.setAttribute("pageResult", pageResult);

request.getRequestDispatcher(userListUrl).forward(request, response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doGet(request, response);

}

}

**-----------------------------**UserManage.**java结束------------------------------**

手机类别管理

视图（JSP页面）

该模块视图部分由两个JSP页面构成，mobileTypeList.jsp负责显示用户表mobileclassify中的信息，实现查询和删除手机类别的功能；editMobileType.jsp负责显示指定类别的手机类别信息，实现增加和编辑手机类别的功能。首先在WebRoot目录下找到建好management,在目录management下 新建mobile目录，在mobile目录下新建Jsp文件命名为mobileTypeList；同理再新建Jsp文件且命名为mobileTypeList。页面效果如图所示。



**--------------------------- mobileTypeList.jsp开始---------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>

<html lang="zh-cn">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>商品类别列表</title>

<!-- Bootstrap -->

<link href="${pageContext.request.contextPath }/assets/cssLibs/bootstrap.min.css" rel="stylesheet">

<link href="${pageContext.request.contextPath }/assets/css/style.css" rel="stylesheet">

<script src="${pageContext.request.contextPath }/assets/JsLibs/jquery-3.3.1.min.js"></script>

<script src="${pageContext.request.contextPath }/assets/JsLibs/bootstrap.min.js"></script>

</head>

<body>

<!--路径导航-->

<ol class="breadcrumb breadcrumb\_nav">

<li>首页</li>

<li>手机类别管理</li>

<li class="active">手机类别列表</li>

</ol>

<!--路径导航-->

<div class="page-content">

<div class="panel panel-default">

<div class="panel-heading">手机类别列表</div>

<div class="panel-body">

<div class="pull-right">

<div class="form-group">

<label class="sr-only" for="exampleInputEmail3">商品类别名</label>

<input type="email" class="form-control" id="goodsTypeName" placeholder="请输入商品类别名">

</div>

<button type="button" class="btn btn-default" onclick="queryByPages()" >查询</button>

<a href="${pageContext.request.contextPath }/servlet/GoodsTypeManage?method=editUI" class="btn btn-default" >添加</a>

</div>

</div>

<div class="table-responsive">

<table class="table table-striped table-hover">

<thead>

<tr>

<th style='width="3%"'>

<!-- <input class="check\_all" type="checkbox"> -->

</th>

<th style='width:40%'>ID</th>

<th style='width:40%'>类别名称</th>

</tr>

</thead>

<tbody>

<c:forEach items="${pageResult.list}" var="item">

<tr>

<td>

</td>

<td>${item.gtId }</td>

<td>${item.name}</td>

<td>

<a href="${pageContext.request.contextPath }/servlet/GoodsTypeManage?method=editUI&id=${item.gtId}" class="btn btn-info btn-xs"><span class="glyphicon glyphicon-edit"></span>编辑</a>

<a href="${pageContext.request.contextPath }/servlet/GoodsTypeManage?method=delete&id=${item.gtId}" class="btn btn-warning btn-xs"><span class="glyphicon glyphicon-remove"></span>删除</a>

</td>

</tr>

</c:forEach>

</tbody>

</table>

<div class="panel-footer clearfix">

<div class="pull-left">

<div class="form-group">

<label>总记录数：${pageResult.totalRecord}</label>

<label>每页显示：</label>

<select class="form-control" id="pageSize" onchange="queryByPages()">

<option value="5" <c:if test="${pageResult.pageSize eq 5}">selected="selected"</c:if> >5</option>

<option value="10" <c:if test="${pageResult.pageSize eq 10}">selected="selected"</c:if> >10</option>

<option value="15" <c:if test="${pageResult.pageSize eq 15}">selected="selected"</c:if> >15</option>

</select>

</div>

</div>

<nav class="pull-right">

<ul class="pagination pagination-self">

<!-- 上一页 -->

<c:if test="${pageResult.currentPage == 1}">

<li>

<span aria-hidden="true">&laquo;</span>

</li>

</c:if>

<c:if test="${pageResult.currentPage >1}">

<li>

<a href="${pageContext.request.contextPath }/servlet/GoodsTypeManage?method=list&currentPage=${pageResult.currentPage-1}" aria-label="Previous">

<span aria-hidden="true">&laquo;</span>

</a>

</li>

</c:if>

<c:forEach begin="${pageResult.beginIndex }" end="${pageResult.endIndex }" varStatus="status">

<c:if test="${status.index eq pageResult.currentPage}">

<li><span style="color: black;">${status.index }</span></li>

</c:if>

<c:if test="${status.index ne pageResult.currentPage}">

<li><a href="${pageContext.request.contextPath }/servlet/GoodsTypeManage?method=list&currentPage=${status.index}">${status.index }</a></li>

</c:if>

</c:forEach>

<!-- 下一页 -->

<c:if test="${pageResult.currentPage == pageResult.totalPages}">

<li>

<span aria-hidden="true">&raquo;</span>

</li>

</c:if>

<c:if test="${pageResult.currentPage < pageResult.totalPages}">

<li>

<a href="${pageContext.request.contextPath }/servlet/GoodsTypeManage?method=list&currentPage=${pageResult.currentPage+1}" aria-label="Next">

<span aria-hidden="true">&raquo;</span>

</a>

</li>

</c:if>

</ul>

</nav>

</div>

</div>

</div>

</div>

<script type="text/javascript">

//定义js方法格式： function 方法名(){}

// js 只有一种数据类型 var

function queryByPages(){

//设置当前浏览器的地址

var pageSize = document.getElementById("pageSize").value;

var goodsTypeName = document.getElementById("goodsTypeName").value;

document.location.href="${pageContext.request.contextPath }/servlet/GoodsTypeManage?method=list&currentPage=1&pageSize="+pageSize+"&goodsTypeName="+goodsTypeName;

}

</script>

</body>

</html>

**---------------------------** **mobileTypeList.jsp结束---------------------------**



**--------------------------- editMobileType.jsp开始---------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>

<html lang="zh-cn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="content-type" content = "text/html;charset=utf-8">

<title>添加用户</title>

<!-- Bootstrap -->

<link href="${pageContext.request.contextPath }/assets/cssLibs/bootstrap.min.css" rel="stylesheet">

<link href="${pageContext.request.contextPath }/assets/css/style.css" rel="stylesheet">

<script src="${pageContext.request.contextPath }/assets/JsLibs/jquery-3.3.1.min.js"></script>

<script src="${pageContext.request.contextPath }/assets/JsLibs/bootstrap.min.js"></script>

</head>

<body>

<!--路径导航-->

<ol class="breadcrumb breadcrumb\_nav">

<li>首页</li>

<li>手机类别管理</li>

<li class="active">手机类别编辑</li>

</ol>

<!--路径导航-->

<div class="page-content">

<div class="panel panel-default">

<div class="panel-heading">编辑手机类别</div>

<div class="panel-body">

<form action="${pageContext.request.contextPath }/servlet/GoodsTypeManage?method=edit" method="post">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="form-group clearfix">

<label>类别名称：</label>

<input type="text" class="form-control" id="name" name="goodsTypeName" value="${goodsType.name}" placeholder="请输入手机号">

</div>

<div class="form-group clearfix">

<input type="hidden" class="form-control" id="typeId" name="goodsTypeId" value="${goodsType.gtId}" placeholder="请输入手机号">

</div>

</div>

<input type="hidden" name="isUpdate" value="${goodsType.name}"/>

</div>

<button type="submit" class="btn btn-primary">保存</button>

</div>

</form>

</div>

</div>

</div>

</body>

</html>

**---------------------------- editMobileType.jsp结束---------------------------**

模型（JavaBean）

打开mobileShop，找到src目录，在创建好的包management.bean中新建Class文件命名为GoodsType，GoodsType.java是对手机类别信息的设置和获取。GoodsType.java文件的具体代码如下：

**------------------------------** **GoodsType.java开始-----------------------------**

package management.bean;

public class GoodsType {

private Integer gtId;

private String name;

public GoodsType() {

}

public GoodsType(Integer gtId, String name) {

this.gtId = gtId;

this.name = name;

}

public GoodsType(String name) {

this.name = name;

}

public Integer getGtId() {

return gtId;

}

public void setGtId(Integer gtId) {

this.gtId = gtId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**------------------------------** **GoodsType.java结束-----------------------------**

控制器（servlet）

在创建好的包management.databaseImp下新建Class命名为GoodsTypeDatabaseImpl；在创建好的包management.service下新建Class命名为GoodsTypeServiceImpl；在创建好的包management.servlet创建Class文件命名为GoodsTypeManage，文件内容如下：

**------------------------GoodsTypeDatabaseImpl.java开始------------------------**

package management.databaseImp;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import org.utils.JdbcUtil;

import management.bean.GoodsType;

import management.tool.PageResult;

public class GoodsTypeDatabaseImpl {

public void add(GoodsType goodsType) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="insert into mobileclassify(name) value(?) ";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, goodsType.getName());

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

public GoodsType queryById(Integer id) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

GoodsType goodsType = null;

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from mobileclassify where id=? ";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, id);

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

String name = resultSet.getString("name");

goodsType = new GoodsType(id, name);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return goodsType;

}

public List<GoodsType> queryAll() {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

List<GoodsType> goodsTypeList = new ArrayList<>();

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from mobileclassify";

prepareStatement = connection.prepareStatement(sql);

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

Integer id = resultSet.getInt("id");

String name = resultSet.getString("name");

GoodsType goodsType = new GoodsType(id, name);

goodsTypeList.add(goodsType);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return goodsTypeList;

}

public void update(GoodsType goodsType) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="update mobileclassify set name=? where id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, goodsType.getName());

prepareStatement.setInt(2, goodsType.getGtId());

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

public void delete(Integer id) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="delete from mobileclassify where id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, id);

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

public PageResult queryByPages(Integer currentPage, Integer pageSize, String typeName) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

PreparedStatement prepareStatement2 = null;

ResultSet resultSet2 = null;

PageResult pageResult = null;

try {

connection = JdbcUtil.getConnection();

String whereSql = "";

if(typeName!=null && typeName.trim().length()>0) {

whereSql = "where name like '%"+typeName+"%' ";

}

String sql ="select \* from mobileclassify "+whereSql+" limit ?,?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, (currentPage-1)\*pageSize);

prepareStatement.setInt(2, pageSize);

resultSet = prepareStatement.executeQuery();

List<GoodsType> goodsTypeList = new ArrayList<>();

while(resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

GoodsType goodsType = new GoodsType(id, name);

goodsTypeList.add(goodsType);

}

String sql2 ="select \* from mobileclassify "+whereSql;

prepareStatement2 = connection.prepareStatement(sql2);

resultSet2 = prepareStatement2.executeQuery();

Long totalRecord = 0L;

while(resultSet2.next()) {

totalRecord = totalRecord + 1;

}

pageResult = new PageResult(goodsTypeList, totalRecord, pageSize, currentPage);

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

JdbcUtil.close(connection, prepareStatement2, resultSet2);

}

return pageResult;

}

}

**----------------------**-**GoodsTypeDatabaseImpl**.**java结束-------------------------**

**------------------------GoodsTypeServiceImpl.java开始-------------------------**

package management.service;

import java.util.List;

import management.tool.PageResult;

import management.databaseImp.GoodsTypeDatabaseImpl;

import management.bean.GoodsType;

public class GoodsTypeServiceImpl{

public List<GoodsType> queryAll() {

GoodsTypeDatabaseImpl GoodsTypeDao = new GoodsTypeDatabaseImpl();

return GoodsTypeDao.queryAll();

}

public GoodsType queryById(int id) {

GoodsTypeDatabaseImpl GoodsTypeDao = new GoodsTypeDatabaseImpl();

return GoodsTypeDao.queryById(id);

}

public void add(GoodsType goodsType) {

GoodsTypeDatabaseImpl GoodsTypeDao = new GoodsTypeDatabaseImpl();

GoodsTypeDao.add(goodsType);

}

public void update(GoodsType goodsType) {

GoodsTypeDatabaseImpl GoodsTypeDao = new GoodsTypeDatabaseImpl();

GoodsTypeDao.update(goodsType);

}

public void delete(Integer id) {

GoodsTypeDatabaseImpl GoodsTypeDao = new GoodsTypeDatabaseImpl();

GoodsTypeDao.delete(id);

}

public PageResult queryByPages(Integer currentPage, Integer pageSize,String name) {

GoodsTypeDatabaseImpl GoodsTypeDao = new GoodsTypeDatabaseImpl();

return GoodsTypeDao.queryByPages(currentPage, pageSize,name);

}

}

**------------------------GoodsTypeServiceImpl**.**java结束-------------------------**

**--------------------------** **GoodsTypeManage.java开始---------------------------**

package management.servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import management.bean.GoodsType;

import management.tool.PageResult;

import management.service.GoodsTypeServiceImpl;

/\*\*

\* Servlet implementation class GoodsTypeManage

\*/

@WebServlet("/GoodsTypeManage")

public class GoodsTypeManage extends HttpServlet {

private static final long serialVersionUID = 1L;

String goodsTypeListUrl = "/management/mobile/mobileTypeList.jsp";

String editGoodsTypeUrl = "/management/mobile/editMobileType.jsp";

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public GoodsTypeManage() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

request.setCharacterEncoding("UTF-8");

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

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

switch (method) {

case "list":

list(request, response);

break;

case "editUI":

editUI(request, response);

break;

case "edit":

edit(request, response);

break;

case "queryById":

queryById(request, response);

break;

case "delete":

delete(request, response);

break;

default:

break;

}

}

/\*\*

\*

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

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

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

GoodsTypeServiceImpl GoodsTypeServiceImpl = new GoodsTypeServiceImpl();

GoodsTypeServiceImpl.delete(Integer.parseInt(id));

// 鍥炲埌鍒楄〃椤甸潰 鍥犱负鎴戜滑浣跨敤iframe妗嗘灦锛屾墍浠ュ彲浠ョ洿鎺ヨ皟鐢ㄦ柟娉曘��

list(request, response);

// response.sendRedirect(request.getContextPath() + "/user?method=list");

}

/\*\*

\*

\*

\* @param request

\* @param response

\*/

private void queryById(HttpServletRequest request, HttpServletResponse response) {

}

/\*\*

\*

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\* @throws FileUploadException

\*/

private void edit(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String isUpdate = request.getParameter("isUpdate"); //用于判断是添加还是更新

String id = request.getParameter("goodsTypeId");

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

/\*String imageUrl = upload(request);\*/

/\*System.out.println(imageUrl);\*/

GoodsTypeServiceImpl goodsTypeService = new GoodsTypeServiceImpl();

if(isUpdate!=null&&isUpdate.trim().length()>0) {

GoodsType goodsType = new GoodsType(Integer.parseInt(id), name);

goodsTypeService.update(goodsType);

}else {

GoodsType goodsType = new GoodsType(name);

goodsTypeService.add(goodsType);

}

list(request, response);

}

/\*\*

\*

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void editUI(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String typeId = request.getParameter("id");

if(typeId!=null&&typeId.trim().length()>0) {

GoodsTypeServiceImpl goodsTypeService = new GoodsTypeServiceImpl();

GoodsType goodsType = goodsTypeService.queryById(Integer.parseInt(typeId));

request.setAttribute("goodsType", goodsType);

}

request.getRequestDispatcher(editGoodsTypeUrl).forward(request, response);

}

/\*\*

\* 鍟嗗搧绫诲埆鍒楄〃

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void list(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

String typeName = request.getParameter("goodsTypeName");

Integer pageSizeInt = 5;

Integer currentPageInt = 1;

if (pageSize != null && pageSize.length() > 0) {

pageSizeInt = Integer.parseInt(pageSize);

}

if (currentPage != null && currentPage.length() > 0) {

currentPageInt = Integer.parseInt(currentPage);

}

GoodsTypeServiceImpl goodsTypeService = new GoodsTypeServiceImpl();

PageResult pageResult = goodsTypeService.queryByPages(currentPageInt, pageSizeInt,typeName);

request.setAttribute("pageResult", pageResult);

request.getRequestDispatcher(goodsTypeListUrl).forward(request, response);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**--------------------------** **GoodsTypeManage**.**java结束--------------------------**

手机管理

视图（JSP页面）

该模块视图部分由两个JSP页面构成，mobileList.jsp负责显示手机信息表mobileform中的信息，实现查询和删除手机信息的功能；editMobile.jsp负责显示指定手机的信息，实现增加和编辑手机信息的功能。首先在WebRoot目录下找到建好management,在目录management下找到mobile目录，在mobile目录下新建Jsp文件命名为editMobile；同理再新建Jsp文件且命名为editMobile。页面效果如图所示。



**---------------------------** ---**mobileList.jsp开始-----------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>

<html lang="zh-cn">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>商品类别列表</title>

<!-- Bootstrap -->

<link href="${pageContext.request.contextPath }/assets/cssLibs/bootstrap.min.css" rel="stylesheet">

<link href="${pageContext.request.contextPath }/assets/css/style.css" rel="stylesheet">

<script src="${pageContext.request.contextPath }/assets/JsLibs/jquery-3.3.1.min.js"></script>

<script src="${pageContext.request.contextPath }/assets/JsLibs/bootstrap.min.js"></script>

</head>

<body>

<!--路径导航-->

<ol class="breadcrumb breadcrumb\_nav">

<li>首页</li>

<li>手机管理</li>

<li class="active">手机列表</li>

</ol>

<!--路径导航-->

<div class="page-content">

<div class="panel panel-default">

<div class="panel-heading">手机列表</div>

<div class="panel-body">

<div class="pull-right">

<div class="form-group">

<label class="sr-only" for="exampleInputEmail3">手机版本</label>

<input type="email" class="form-control" id="mobileVersion" placeholder="请输入手机版本号或名称">

</div>

<button type="button" class="btn btn-default" onclick="queryByPages()" >查询</button>

<a href="${pageContext.request.contextPath }/servlet/GoodsManage?method=editUI" class="btn btn-default" >添加</a>

</div>

</div>

<div class="table-responsive">

<table class="table table-striped table-hover">

<thead>

<tr>

<th style='width="3%"'>

<!-- <input class="check\_all" type="checkbox"> -->

</th>

<th style='width:9%'>产品号</th>

<th style='width:9%'>手机名称</th>

<th style='width:14%'>制造商</th>

<th style='width:14%'>价格</th>

<th style='width:30%'>介绍</th>

<th style='width:14%'>图片</th>

<th style='width:8%'>类别</th>

</tr>

</thead>

<tbody>

<c:forEach items="${pageResult.list}" var="item">

<tr>

<td>

</td>

<td>${item.mobile\_version}</td>

<td>${item.mobile\_name}</td>

<td>${item.mobile\_made }</td>

<td>${item.mobile\_price}</td>

<td>${item.mobile\_mess}</td>

<td>${item.mobile\_pic}</td>

<td>${item.id}</td>

<td>

<a href="${pageContext.request.contextPath }/servlet/GoodsManage?method=editUI&mobileVersion=${item.mobile\_version}" class="btn btn-info btn-xs"><span class="glyphicon glyphicon-edit"></span>编辑</a>

<a href="${pageContext.request.contextPath }/servlet/GoodsManage?method=delete&mobileVersion=${item.mobile\_version}" class="btn btn-warning btn-xs"><span class="glyphicon glyphicon-remove"></span>删除</a>

</td>

</tr>

</c:forEach>

</tbody>

</table>

<div class="panel-footer clearfix">

<div class="pull-left">

<div class="form-group">

<label>总记录数：${pageResult.totalRecord}</label>

<label>每页显示：</label>

<select class="form-control" id="pageSize" onchange="queryByPages()">

<option value="5" <c:if test="${pageResult.pageSize eq 5}">selected="selected"</c:if> >5</option>

<option value="10" <c:if test="${pageResult.pageSize eq 10}">selected="selected"</c:if> >10</option>

<option value="15" <c:if test="${pageResult.pageSize eq 15}">selected="selected"</c:if> >15</option>

</select>

</div>

</div>

<nav class="pull-right">

<ul class="pagination pagination-self">

<!-- 上一页 -->

<c:if test="${pageResult.currentPage == 1}">

<li>

<span aria-hidden="true">&laquo;</span>

</li>

</c:if>

<c:if test="${pageResult.currentPage >1}">

<li>

<a href="${pageContext.request.contextPath }/servlet/GoodsManage?method=list&currentPage=${pageResult.currentPage-1}" aria-label="Previous">

<span aria-hidden="true">&laquo;</span>

</a>

</li>

</c:if>

<c:forEach begin="${pageResult.beginIndex }" end="${pageResult.endIndex }" varStatus="status">

<c:if test="${status.index eq pageResult.currentPage}">

<li><span style="color: black;">${status.index }</span></li>

</c:if>

<c:if test="${status.index ne pageResult.currentPage}">

<li><a href="${pageContext.request.contextPath }/servlet/GoodsManage?method=list&currentPage=${status.index}">${status.index }</a></li>

</c:if>

</c:forEach>

<!-- 下一页 -->

<c:if test="${pageResult.currentPage == pageResult.totalPages}">

<li>

<span aria-hidden="true">&raquo;</span>

</li>

</c:if>

<c:if test="${pageResult.currentPage < pageResult.totalPages}">

<li>

<a href="${pageContext.request.contextPath }/servlet/GoodsManage?method=list&currentPage=${pageResult.currentPage+1}" aria-label="Next">

<span aria-hidden="true">&raquo;</span>

</a>

</li>

</c:if>

</ul>

</nav>

</div>

</div>

</div>

</div>

<script type="text/javascript">

//定义js方法格式： function 方法名(){}

function queryByPages(){

var pageSize = document.getElementById("pageSize").value;

var mobileVersion = document.getElementById("mobileVersion").value;

document.location.href="${pageContext.request.contextPath }/servlet/GoodsManage?method=list&currentPage=1&pageSize="+pageSize+"&mobileVersion="+mobileVersion;

}

</script>

</body>

</html>

**------------------------------** **mobileList.jsp结束-----------------------------**



**--------------------------------editMobile.jsp开始---------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>

<html lang="zh-cn">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="content-type" content = "text/html;charset=utf-8">

<title>添加用户</title>

<!-- Bootstrap -->

<link href="${pageContext.request.contextPath }/assets/cssLibs/bootstrap.min.css" rel="stylesheet">

<link href="${pageContext.request.contextPath }/assets/css/style.css" rel="stylesheet">

<script src="${pageContext.request.contextPath }/assets/JsLibs/jquery-3.3.1.min.js"></script>

<script src="${pageContext.request.contextPath }/assets/JsLibs/bootstrap.min.js"></script>

</head>

<body>

<!--路径导航-->

<ol class="breadcrumb breadcrumb\_nav">

<li>首页</li>

<li>手机管理</li>

<li class="active">手机编辑</li>

</ol>

<!--路径导航-->

<div class="page-content">

<div class="panel panel-default">

<div class="panel-heading">编辑手机</div>

<div class="panel-body">

<form id="mobileForm" method="post" enctype="multipart/form-data">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="form-group clearfix">

<label>版本号：</label>

<input type="text" class="form-control" id="mobileVersion" name="mobileVersion" value="${goods.mobile\_version}" placeholder="请输入版本号">

</div>

<div class="form-group clearfix">

<label>手机名称：</label>

<input type="text" class="form-control" id="mobileName" name="mobileName" value="${goods.mobile\_name}" placeholder="请输入手机名称">

</div>

<div class="form-group clearfix">

<label>制造商：</label>

<input type="text" class="form-control" id="mobileMade" name="mobileMade" value="${goods.mobile\_made}" placeholder="请输入制造商">

</div>

<div class="form-group clearfix">

<label>价格：</label>

<input type="text" class="form-control" id="mobilePrice" name="mobilePrice" value="${goods.mobile\_price}" placeholder="请输入价格">

</div>

<div class="form-group clearfix">

<label>详情：</label>

<input type="text" class="form-control" id="mobileMess" name="mobileMess" value="${goods.mobile\_mess}" placeholder="请输入手机介绍详情">

</div>

<div class="form-group clearfix">

<label>照片：</label>

<c:if test="${goods.mobile\_pic.length()>0}">

<input type="text" class="form-control" id="mobilePic" name="mobilePic" value="${goods.mobile\_pic}" placeholder="请输入照片链接">

</c:if>

<c:if test="${goods.mobile\_pic.length()==0||goods.mobile\_pic==null}">

<input type="hidden" class="form-control" id="mobilePic" name="mobilePic" value="empty" placeholder="请输入照片链接">

</c:if>

<div class="form-group clearfix" id="updatebox">

<input type="file" id="file" name="mobilePic" />

</div>

</div>

<div class="form-group clearfix">

<label>类别：</label>

<input type="text" class="form-control" id="id" name="id" value="${goods.id}" placeholder="请输入手机类别">

</div>

<input type="hidden" id="isUpdate" name="isUpdate" value="${goods.mobile\_version}"/>

</div>

<input type="hidden" name="isUpdate" value="${goods.mobile\_version}"/>

</div>

<div class="form-group clearfix col-sm-4" id="updatebox">

<a class="btn btn-primary" onclick="submit()">保存</a>

</div>

</div>

</form>

</div>

</div>

</div>

<script type="text/javascript">

function submit(){

var mobileForm = document.getElementById("mobileForm");

var mobileVersion = document.getElementById("mobileVersion").value;

var mobileName = document.getElementById("mobileName").value;

var mobileMade = document.getElementById("mobileMade").value;

var mobilePrice = document.getElementById("mobilePrice").value;

var mobileMess = document.getElementById("mobileMess").value;

var isUpdate = document.getElementById("isUpdate").value;

var mobilePic = document.getElementById("mobilePic").value;

var ifSelect;

var selectFile = $('#file').get(0).files[0];

console.info(selectFile);

if(selectFile.length!=0){

ifSelect = 1; //有选择文件

}else{

ifSelect = 0;

}

if(isUpdate==null||isUpdate.length==0){

isUpdate = 0; //不是更新，是增加

}else{

isUpdate = 1;

}

var id = document.getElementById("id").value;

mobileForm.action="${pageContext.request.contextPath}/servlet/GoodsManage?method=edit&mobileVersion="+mobileVersion+"&mobileName="+mobileName+"&mobileMade="+mobileMade+"&mobilePrice="+mobilePrice+"&mobileMess="+mobileMess+"&mobilePic="+mobilePic+"&ifSelect="+ifSelect+"&id="+id+"&isUpdate="+isUpdate;

mobileForm.submit();

}

</script>

</body>

</html>

**--------------------------------editMobile.jsp开始---------------------------**

模型（JavaBean）

打开mobileShop，找到src目录，在创建好的包management.bean中新建Class文件命名为Goods，Goods.java是对手机信息的设置和获取。Goods.java文件的具体代码如下：

**-----------------------------------Goods.java开始-----------------------------**

package management.bean;

public class Goods {

private String mobile\_version;

private String mobile\_name;

private String mobile\_made;

private float mobile\_price;

private String mobile\_mess;

private String mobile\_pic;

private Integer id;

public String getMobile\_version() {

return mobile\_version;

}

public void setMobile\_version(String mobile\_version) {

this.mobile\_version = mobile\_version;

}

public String getMobile\_name() {

return mobile\_name;

}

public void setMobile\_name(String mobile\_name) {

this.mobile\_name = mobile\_name;

}

public String getMobile\_made() {

return mobile\_made;

}

public void setMobile\_made(String mobile\_made) {

this.mobile\_made = mobile\_made;

}

public float getMobile\_price() {

return mobile\_price;

}

public void setMobile\_price(float mobile\_price) {

this.mobile\_price = mobile\_price;

}

public String getMobile\_mess() {

return mobile\_mess;

}

public void setMobile\_mess(String mobile\_mess) {

this.mobile\_mess = mobile\_mess;

}

public String getMobile\_pic() {

return mobile\_pic;

}

public void setMobile\_pic(String mobile\_pic) {

this.mobile\_pic = mobile\_pic;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public Goods(String mobile\_version, String mobile\_name,String mobile\_made,float mobile\_price,String mobile\_mess, String mobile\_pic,

Integer id) {

this.mobile\_version = mobile\_version;

this.mobile\_name = mobile\_name;

this.mobile\_made = mobile\_made;

this.mobile\_price =mobile\_price;

this.mobile\_mess = mobile\_mess;

this.mobile\_pic = mobile\_pic;

this.id = id;

}

}

**-----------------------------------Goods.java结束-----------------------------**

控制器（servlet）

在创建好的包management.databaseImp下新建Class命名为GoodsDatabaseImpl；在创建好的包management.service下新建Class命名为GoodseServiceImpl；在创建好的包management.servlet创建Class文件命名为GoodsManage，文件内容如下：

**--------------------------GoodsDatabaseImpl.java开始--------------------------**

package management.databaseImp;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import org.utils.JdbcUtil;

import management.bean.Goods;

import management.tool.PageResult;

public class GoodsDatabaseImpl {

/\*\*

\* 注册用户

\*/

public void add(Goods goods) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="insert into mobileform values(?,?,?,?,?,?,?) ";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, goods.getMobile\_version());

prepareStatement.setString(2, goods.getMobile\_name());

prepareStatement.setString(3, goods.getMobile\_made());

prepareStatement.setFloat(4, goods.getMobile\_price());

prepareStatement.setString(5, goods.getMobile\_mess());

prepareStatement.setString(6, goods.getMobile\_pic());

prepareStatement.setInt(7, goods.getId());

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 通过手机版本号查询

\*/

public Goods queryByVersion(String version) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

Goods goods = null;

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from mobileform where mobile\_version=? ";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, version);

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

String mobile\_version = resultSet.getString("mobile\_version");

String mobile\_name = resultSet.getString("mobile\_name");

String mobile\_made = resultSet.getString("mobile\_made");

float mobile\_price = resultSet.getFloat("mobile\_price");

String mobile\_mess = resultSet.getString("mobile\_mess");

String mobile\_pic = resultSet.getString("mobile\_pic");

int id = resultSet.getInt("id");

goods = new Goods(mobile\_version, mobile\_name,mobile\_made, mobile\_price, mobile\_mess,mobile\_pic,id);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return goods;

}

/\*\*

\* 查询表中所用的用户

\*/

public List<Goods> queryAll() {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

List<Goods> goodsList = new ArrayList<>();

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from mobileform";

prepareStatement = connection.prepareStatement(sql);

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

String mobile\_version = resultSet.getString("mobile\_version");

String mobile\_name = resultSet.getString("mobile\_name");

String mobile\_made = resultSet.getString("mobile\_made");

float mobile\_price = resultSet.getFloat("mobile\_price");

String mobile\_mess = resultSet.getString("mobile\_mess");

String mobile\_pic = resultSet.getString("mobile\_pic");

int id = resultSet.getInt("id");

Goods goods = new Goods(mobile\_version, mobile\_name,mobile\_made, mobile\_price, mobile\_mess,mobile\_pic,id);

goodsList.add(goods);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return goodsList;

}

/\*\*

\* 修改信息

\*/

public void update(Goods goods) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="update mobileform set mobile\_name=?, mobile\_made=?, mobile\_price=?, mobile\_mess=?, mobile\_pic=?, id=? where mobile\_version=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, goods.getMobile\_name());

prepareStatement.setString(2, goods.getMobile\_made());

prepareStatement.setFloat(3, goods.getMobile\_price());

prepareStatement.setString(4, goods.getMobile\_mess());

prepareStatement.setString(5, goods.getMobile\_pic());

prepareStatement.setInt(6, goods.getId());

prepareStatement.setString(7, goods.getMobile\_version());

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 删除用户

\*/

public void delete(String mobile\_version) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="delete from mobileform where mobile\_version=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, mobile\_version);

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 分页查询

\*/

public PageResult queryByPages(Integer currentPage, Integer pageSize, String mobileVersion) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

PreparedStatement prepareStatement2 = null;

ResultSet resultSet2 = null;

PageResult pageResult = null;

try {

connection = JdbcUtil.getConnection();

String whereSql = "";

if(mobileVersion!=null && mobileVersion.trim().length()>0) {

whereSql = "where (mobile\_version like '%"+mobileVersion+"%') or (mobile\_name like '%"+mobileVersion+"%') ";

}

String sql ="select \* from mobileform "+whereSql+" limit ?,?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, (currentPage-1)\*pageSize);

prepareStatement.setInt(2, pageSize);

resultSet = prepareStatement.executeQuery();

List<Goods> goodsList = new ArrayList<>();

while(resultSet.next()) {

String mobile\_version = resultSet.getString("mobile\_version");

String mobile\_name = resultSet.getString("mobile\_name");

String mobile\_made = resultSet.getString("mobile\_made");

float mobile\_price = resultSet.getFloat("mobile\_price");

String mobile\_mess = resultSet.getString("mobile\_mess");

String mobile\_pic = resultSet.getString("mobile\_pic");

int id = resultSet.getInt("id");

Goods goods = new Goods(mobile\_version, mobile\_name,mobile\_made, mobile\_price, mobile\_mess,mobile\_pic,id);

goodsList.add(goods);

}

//查询表中用户总数

String sql2 ="select \* from mobileform "+whereSql;

prepareStatement2 = connection.prepareStatement(sql2);

resultSet2 = prepareStatement2.executeQuery();

Long totalRecord = 0L;

while(resultSet2.next()) {

totalRecord = totalRecord + 1;

}

pageResult = new PageResult(goodsList, totalRecord, pageSize, currentPage);

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

JdbcUtil.close(connection, prepareStatement2, resultSet2);

}

return pageResult;

}

}

**-------------------------**-**GoodsDatabaseImpl**.**java结束--------------------------**

**--------------------------GoodsServiceImpl.java开始---------------------------**

package management.service;

import java.util.List;

import management.bean.Goods;

import management.databaseImp.GoodsDatabaseImpl;

import management.tool.PageResult;

public class GoodsServiceImpl {

public List<Goods> queryAll() {

GoodsDatabaseImpl goodsDao = new GoodsDatabaseImpl();

//浠ュ悗浼佷笟椤圭洰鍦╯ervice閲岄潰鍙兘浼氬仛寰堝浜嬫儏

return goodsDao.queryAll();

}

public void add(Goods goods) {

GoodsDatabaseImpl goodsDao = new GoodsDatabaseImpl();

goodsDao.add(goods);

}

public void delete(String mobile\_version) {

GoodsDatabaseImpl goodsDao = new GoodsDatabaseImpl();

goodsDao.delete(mobile\_version);

}

public PageResult queryByPages(Integer currentPage, Integer pageSize, String mobileVersion) {

GoodsDatabaseImpl goodsDao = new GoodsDatabaseImpl();

return goodsDao.queryByPages(currentPage, pageSize, mobileVersion);

}

public Goods queryByVersion(String version) {

GoodsDatabaseImpl goodsDao = new GoodsDatabaseImpl();

return goodsDao.queryByVersion(version);

}

public void update(Goods goods) {

GoodsDatabaseImpl goodsDao = new GoodsDatabaseImpl();

goodsDao.update(goods);

}

}

**--------------------------GoodsServiceImpl**.**java结束---------------------------**

**----------------------------** **GoodsManage.java开始-----------------------------**

package management.servlet;

import java.io.File;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import java.util.UUID;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.fileupload.FileItem;

import org.apache.commons.fileupload.FileUploadException;

import org.apache.commons.fileupload.disk.DiskFileItemFactory;

import org.apache.commons.fileupload.servlet.ServletFileUpload;

import management.bean.Goods;

import management.service.GoodsServiceImpl;

import management.tool.PageResult;

/\*\*

\* Servlet implementation class GoodsManage

\*/

@WebServlet("/GoodsManage")

public class GoodsManage extends HttpServlet {

private static final long serialVersionUID = 1L;

String goodsListUrl = "/management/mobile/mobileList.jsp";

String editGoodsUrl = "/management/mobile/editMobile.jsp";

String imageUrl = "/images";

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public GoodsManage() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

request.setCharacterEncoding("UTF-8");

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

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

switch (method) {

case "list":

list(request, response);

break;

case "editUI":

editUI(request, response);

break;

case "edit":

try {

edit(request, response);

} catch (FileUploadException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

break;

case "queryById":

queryById(request, response);

break;

case "delete":

delete(request, response);

break;

default:

break;

}

}

/\*\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

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

String mobile\_version = request.getParameter("mobileVersion");

GoodsServiceImpl GoodsServiceImpl = new GoodsServiceImpl();

GoodsServiceImpl.delete(mobile\_version);

// 鍥炲埌鍒楄〃椤甸潰 鍥犱负鎴戜滑浣跨敤iframe妗嗘灦锛屾墍浠ュ彲浠ョ洿鎺ヨ皟鐢ㄦ柟娉曘��

list(request, response);

// response.sendRedirect(request.getContextPath() + "/user?method=list");

}

/\*\*

\*

\* @param request

\* @param response

\*/

private void queryById(HttpServletRequest request, HttpServletResponse response) {

}

/\*\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\* @throws FileUploadException

\*/

private void edit(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException, FileUploadException {

PrintWriter out = response.getWriter();

String isUpdate = request.getParameter("isUpdate"); //用于判断是添加还是更新

String ifSelect = request.getParameter("ifSelect"); //用于判断是添加还是更新

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

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

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

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

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

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

if(Integer.parseInt(ifSelect)==1) {

mobilePic = up(request,response); //选择了图片

}

/\*out.print("pic"+mobilePic);\*/

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

GoodsServiceImpl goodsService = new GoodsServiceImpl();

if(Integer.parseInt(isUpdate)!=0) {

out.print(ifSelect+"选择"+isUpdate+"更新"+mobileVersion+"mobileName"+mobileName+"mobileMade"+mobileMade+"mobilePrice"+mobilePrice+"mobileMess"+mobileMess+"mobilePic"+mobilePic);

Goods goods = new Goods(mobileVersion,mobileName,mobileMade,Float.parseFloat(mobilePrice),mobileMess,mobilePic,Integer.parseInt(id));

goodsService.update(goods);

}else {

out.print(ifSelect+"选择"+isUpdate+"add"+mobileVersion+"mobileName"+mobileName+"mobileMade"+mobileMade+"mobilePrice"+mobilePrice+"mobileMess"+mobileMess+"mobilePic"+mobilePic);

Goods goods = new Goods(mobileVersion,mobileName,mobileMade,Float.parseFloat(mobilePrice),mobileMess,mobilePic,Integer.parseInt(id));

goodsService.add(goods);

}

list(request, response);

}

public String up(HttpServletRequest req,HttpServletResponse response) throws ServletException, IOException{

DiskFileItemFactory factory = new DiskFileItemFactory();

ServletFileUpload upload = new ServletFileUpload(factory);

//把文件的中文乱码解决了

upload.setHeaderEncoding("utf-8");

upload.setFileSizeMax(4 \* 1024 \* 1024);// 4M

upload.setSizeMax(4 \* 1024 \* 1024);// 6M

//设置文件的大小

//upload.setSizeMax(20\*1024);

String fileName="";

try {

//[3]获得前台传递过来的所有信息对象

List<FileItem> list = upload.parseRequest(req);

for (FileItem item : list) {

//System.out.println(item.getFieldName()+"---"+item.getName()+"---"+item.getSize()+"----"+item.getContentType()+"----"+item.isFormField());

if(!item.isFormField()){

//证明是文件对象

//f.l.jpg 文件的后缀

String sub = item.getName().substring(item.getName().lastIndexOf("."));

/\*\*\*\*[2]保证上传的文件名称不重复\*\*\*\*\*\*\*\*\*\*\*/

//利用UUID产生文件的 名称

String uuid = UUID.randomUUID().toString();

//拼接文件的名称

fileName=uuid+sub;

/\*\*\*\*[2]上传到指定的服务器目录中\*\*\*\*\*\*\*\*\*\*\*/

String realPath = req.getSession().getServletContext().getRealPath(imageUrl);

/\*\*\*\*\*[1]上传的目录是否存在的解决\*\*\*\*\*\*\*/

File file=new File(realPath);

if (!file.exists() && !file.isDirectory()) {

// 鍒涘缓鐩綍

file.mkdir();

}

//实现文件的上传 把item中的文件上传到指定的目录中

item.write(new File(file,fileName));

return fileName;

}

}

} catch (Exception e) {

e.printStackTrace();

}

return fileName;

}

/\*\*

\* 上传图片方案二

\*

\* @param request

\* @return

\* @throws ServletException

\* @throws IOException

\* @throws FileUploadException

\*/

/\* private String upload(HttpServletRequest request,HttpServletResponse response) throws IOException, ServletException, FileUploadException {

PrintWriter outText = response.getWriter();

String savePath =request.getSession().getServletContext().getRealPath(imageUrl);

String resultUrl = "";

File file = new File(savePath);

if (!file.exists() && !file.isDirectory()) {

// 鍒涘缓鐩綍

file.mkdir();

}

DiskFileItemFactory factory = new DiskFileItemFactory();

ServletFileUpload upload = new ServletFileUpload(factory);

//把文件的中文乱码解决了

upload.setHeaderEncoding("utf-8");

upload.setFileSizeMax(4 \* 1024 \* 1024);// 4M

upload.setSizeMax(4 \* 1024 \* 1024);// 6M

List<FileItem> items = upload.parseRequest(request);

for (FileItem item : items) {

InputStream in = item.getInputStream();

String fileName = item.getName();

if (fileName == null || "".equals(fileName.trim())) {

continue;

}

fileName = fileName.substring(fileName.lastIndexOf("\\") + 1);

String fileExt = FileUploadUtil.getFileExt(fileName);

fileName = System.currentTimeMillis() +"."+ fileExt;

String newStorePath = FileUploadUtil.getTimeDesPath();

newStorePath = savePath +"\\"+ newStorePath + "\\";

String storeFile = makeStorePath(newStorePath) + fileName;

resultUrl = imageUrl+newStorePath+fileName;

OutputStream out = new FileOutputStream(storeFile);

byte[] b = new byte[1024];

int len = -1;

while ((len = in.read(b)) != -1) {

out.write(b, 0, len);

}

outText.print("文件上传");

in.close();

out.close();

item.delete();

}

return resultUrl;

}

private String makeStorePath(String storePath) {

File file = new File(storePath);

if(!file.exists())

{

file.mkdirs();

}

return storePath;

} \*/

/\*\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void editUI(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

if(mobileVersion!=null&&mobileVersion.trim().length()>0) {

GoodsServiceImpl goodsService = new GoodsServiceImpl();

Goods goods = goodsService.queryByVersion(mobileVersion);

request.setAttribute("goods", goods);

}

request.getRequestDispatcher(editGoodsUrl).forward(request, response);

}

/\*\*

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void list(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

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

Integer pageSizeInt = 5;

Integer currentPageInt = 1;

if (pageSize != null && pageSize.length() > 0) {

pageSizeInt = Integer.parseInt(pageSize);

}

if (currentPage != null && currentPage.length() > 0) {

currentPageInt = Integer.parseInt(currentPage);

}

GoodsServiceImpl goodsService = new GoodsServiceImpl();

PageResult pageResult = goodsService.queryByPages(currentPageInt, pageSizeInt,mobileVersion);

request.setAttribute("pageResult", pageResult);

request.getRequestDispatcher(goodsListUrl).forward(request, response);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**----------------------------** **GoodsManage**.**java结束-----------------------------**

订单管理

视图（JSP页面）

该模块视图部分由两个JSP页面构成，orderList.jsp负责显示用户表user中的信息，实现查询订单的功能；editOrder.jsp负责显示指定订单的信息，实现增加和编辑订单信息的功能。首先在WebRoot目录下找到建好management,在目录management下 新建order目录，在order目录下新建Jsp文件命名为orderList；同理再新建Jsp文件且命名为e editOrder。页面效果如图所示。



**---------------------------** --- **orderList.jsp开始-----------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>

<html lang="zh-cn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>用户列表</title>

<!-- Bootstrap -->

<link href="${pageContext.request.contextPath }/assets/cssLibs/bootstrap.min.css" rel="stylesheet">

<link href="${pageContext.request.contextPath }/assets/css/style.css" rel="stylesheet">

<script src="${pageContext.request.contextPath }/assets/JsLibs/jquery-3.3.1.min.js"></script>

<script src="${pageContext.request.contextPath }/assets/JsLibs/bootstrap.min.js"></script>

</head>

<body>

<!--路径导航-->

<ol class="breadcrumb breadcrumb\_nav">

<li>首页</li>

<li>订单管理</li>

<li class="active">订单列表</li>

</ol>

<!--路径导航-->

<div class="page-content">

<div class="panel panel-default">

<div class="panel-heading">订单列表</div>

<div class="panel-body">

<div class="pull-right">

<div class="form-group">

<label class="sr-only" for="exampleInputEmail3">购买者</label>

<input type="email" class="form-control" id="userName" placeholder="请输入用户名">

</div>

<button type="button" class="btn btn-default" onclick="queryByPages()" >查询</button>

</div>

</div>

<div class="table-responsive">

<table class="table table-striped table-hover">

<thead>

<tr>

<th style='width="3%"'>

<!-- <input class="check\_all" type="checkbox"> -->

</th>

<th style='width:20%'>购买者</th>

<th style='width:20%'>详情</th>

<th style='width:20%'>总额</th>

<th style='width:20%'>处理状态</th>

</tr>

</thead>

<tbody>

<c:forEach items="${pageResult.list}" var="item">

<tr>

<td>

<!-- <input class="check\_all" type="checkbox" name="ids[]" value="8"> -->

</td>

<td>${item.logname}</td>

<td>${item.mess}</td>

<td>${item.sum}</td>

<c:if test="${item.status==0}">

<td>

<a href="${pageContext.request.contextPath }/servlet/OrderFormManage?method=changeStatus&id=${item.id}" class="btn btn-info btn-xs"><span></span>未处理</a>

</td>

</c:if>

<c:if test="${item.status==1}">

<td>

<a class="btn btn-info btn-xs" style='background-color:grey;border-color:grey;'><span></span>已处理</a>

</td>

</c:if>

<td>

<a href="${pageContext.request.contextPath }/servlet/OrderFormManage?method=editUI&id=${item.id}" class="btn btn-info btn-xs"><span class="glyphicon glyphicon-edit"></span>编辑</a>

<a href="${pageContext.request.contextPath }/servlet/OrderFormManage?method=delete&id=${item.id}" class="btn btn-warning btn-xs"><span class="glyphicon glyphicon-remove"></span>删除</a>

</td>

</tr>

</c:forEach>

</tbody>

</table>

<div class="panel-footer clearfix">

<div class="pull-left">

<div class="form-group">

<label>总记录数：${pageResult.totalRecord}</label>

<label>每页显示：</label>

<select class="form-control" id="pageSize" onchange="queryByPages()">

<option value="5" <c:if test="${pageResult.pageSize eq 5}">selected="selected"</c:if> >5</option>

<option value="10" <c:if test="${pageResult.pageSize eq 10}">selected="selected"</c:if> >10</option>

<option value="15" <c:if test="${pageResult.pageSize eq 15}">selected="selected"</c:if> >15</option>

</select>

</div>

</div>

<nav class="pull-right">

<ul class="pagination pagination-self">

<!-- 上一页 -->

<c:if test="${pageResult.currentPage == 1}">

<li>

<span aria-hidden="true">&laquo;</span>

</li>

</c:if>

<c:if test="${pageResult.currentPage >1}">

<li>

<a href="${pageContext.request.contextPath }/servlet/OrderFormManage?method=list&currentPage=${pageResult.currentPage-1}" aria-label="Previous">

<span aria-hidden="true">&laquo;</span>

</a>

</li>

</c:if>

<c:forEach begin="${pageResult.beginIndex }" end="${pageResult.endIndex }" varStatus="status">

<c:if test="${status.index eq pageResult.currentPage}">

<li><span style="color: black;">${status.index }</span></li>

</c:if>

<c:if test="${status.index ne pageResult.currentPage}">

<li><a href="${pageContext.request.contextPath }/servlet/OrderFormManage?method=list&currentPage=${status.index}">${status.index }</a></li>

</c:if>

</c:forEach>

<!-- 下一页 -->

<c:if test="${pageResult.currentPage == pageResult.totalPages}">

<li>

<span aria-hidden="true">&raquo;</span>

</li>

</c:if>

<c:if test="${pageResult.currentPage < pageResult.totalPages}">

<li>

<a href="${pageContext.request.contextPath }/servlet/OrderFormManage?method=list&currentPage=${pageResult.currentPage+1}" aria-label="Next">

<span aria-hidden="true">&raquo;</span>

</a>

</li>

</c:if>

</ul>

</nav>

</div>

</div>

</div>

</div>

<script type="text/javascript">

//定义js方法格式： function 方法名(){}

// js 只有一种数据类型 var

function queryByPages(){

//设置当前浏览器的地址

var pageSize = document.getElementById("pageSize").value;

var userName = document.getElementById("userName").value;

document.location.href="${pageContext.request.contextPath }/servlet/OrderFormManage?method=list&currentPage=1&pageSize="+pageSize+"&userName="+userName;

}

</script>

</body>

</html>

**------------------------------** **orderList.jsp结束-----------------------------**



**-------------------------------- editOrder.jsp开始---------------------------**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>

<html lang="zh-cn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="content-type" content = "text/html;charset=utf-8">

<title>添加用户</title>

<!-- Bootstrap -->

<link href="${pageContext.request.contextPath }/assets/cssLibs/bootstrap.min.css" rel="stylesheet">

<link href="${pageContext.request.contextPath }/assets/css/style.css" rel="stylesheet">

<script src="${pageContext.request.contextPath }/assets/JsLibs/jquery-3.3.1.min.js"></script>

<script src="${pageContext.request.contextPath }/assets/JsLibs/bootstrap.min.js"></script>

</head>

<body>

<!--路径导航-->

<ol class="breadcrumb breadcrumb\_nav">

<li>首页</li>

<li>用户管理</li>

<li class="active">编辑用户</li>

</ol>

<!--路径导航-->

<div class="page-content">

<div class="panel panel-default">

<div class="panel-heading">编辑用户</div>

<div class="panel-body">

<form action="${pageContext.request.contextPath }/servlet/OrderFormManage?method=edit" method="post">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<div class="form-group clearfix">

<label>购买者：</label>

<input type="text" class="form-control" id="logname" name="logname" value="${orderForm.logname}" placeholder="请输入手机号">

</div>

<div class="form-group clearfix">

<label>详情：</label>

<input type="text" class="form-control" id="mess" name="mess" value="${orderForm.mess}" placeholder="请输入手机号">

</div>

</div>

<div class="col-md-12">

<div class="form-group clearfix">

<label>总额：</label>

<input type="text" class="form-control" id="sum" name="sum" value="${orderForm.sum}" placeholder="请输入手机号">

</div>

</div>

<input type="hidden" name="id" value="${orderForm.id}">

<input type="hidden" name="status" value="${orderForm.status}"/>

</div>

<button type="submit" class="btn btn-primary">保存</button>

</div>

</form>

</div>

</div>

</div>

<%-- <script src="${pageContext.request.contextPath }/assets/js/pcasunzip.js"></script> --%>

<!-- <script type="text/javascript">

new PCAS("province","city",$("#provinceId").val(),$("#cityId").val());

</script> -->

</body>

</html>

**-------------------------------- editOrder.jsp开始---------------------------**

模型（JavaBean）

打开mobileShop，找到src目录，在创建好的包management.bean中新建Class文件命名为OrderForm，OrderForm.java是对订单信息的设置和获取。OrderForm.java文件的具体代码如下：

**------------------------------** **OrderForm.java开始-----------------------------**

package management.bean;

public class OrderForm {

Integer id;

String logname;

String mess;

Float sum;

Integer status;

public OrderForm() {

}

public OrderForm(Integer id,String logname,String mess,Float sum,Integer status) {

this.id = id;

this.logname = logname;

this.mess = mess;

this.sum = sum;

this.status = status;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getLogname() {

return logname;

}

public void setLogname(String logname) {

this.logname = logname;

}

public String getMess() {

return mess;

}

public void setMess(String mess) {

this.mess = mess;

}

public Float getSum() {

return sum;

}

public void setSum(Float sum) {

this.sum = sum;

}

public Integer getStatus() {

return status;

}

public void setStatus(Integer status) {

this.status = status;

}

}

**------------------------------** **OrderForm.java结束-----------------------------**

控制器（servlet）

在创建好的包management.databaseImp下新建Class命名为OrderFormDatabaseImpl；在创建好的包management.service下新建Class命名为OderFormServiceImpl；在创建好的包management.servlet创建Class文件命名为OrderFormManage，文件内容如下：

**------------------------OrderFormDatabaseImpl.java开始------------------------**

package management.databaseImp;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import org.utils.JdbcUtil;

import management.bean.OrderForm;

import management.tool.PageResult;

public class OrderFormDatabaseImpl {

/\*\*

\* 通过购买者名称模糊查询

\*/

public OrderForm queryById(int order\_id) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

OrderForm orderForm = null;

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from orderform where id=? ";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1,order\_id);

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

int id = resultSet.getInt("id");

String logname = resultSet.getString("logname");

String mess = resultSet.getString("mess");

float sum = resultSet.getFloat("sum");

int status = resultSet.getInt("status");

orderForm = new OrderForm(id,logname,mess,sum,status);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return orderForm;

}

/\*\*

\* 查询表

\*/

public List<OrderForm> queryAll() {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

List<OrderForm> orderFormList = new ArrayList<>();

try {

connection = JdbcUtil.getConnection();

String sql ="select \* from orderform";

prepareStatement = connection.prepareStatement(sql);

resultSet = prepareStatement.executeQuery();

while(resultSet.next()) {

int id = resultSet.getInt("id");

String logname = resultSet.getString("logname");

String mess = resultSet.getString("mess");

float sum = resultSet.getFloat("sum");

int status = resultSet.getInt("status");

OrderForm orderForm = new OrderForm(id,logname,mess,sum,status);

orderFormList.add(orderForm);

}

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

}

return orderFormList;

}

/\*\*

\* 修改信息

\*/

public void update(OrderForm orderForm) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="update orderform set logname=?,mess=?,sum=?,status=? where id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, orderForm.getLogname());

prepareStatement.setString(2, orderForm.getMess());

prepareStatement.setFloat(3, orderForm.getSum());

prepareStatement.setInt(4, orderForm.getStatus());

prepareStatement.setInt(5, orderForm.getId());

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 删除

\*/

public void delete(int id) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="delete from orderform where id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, id);

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 改变为已处理状态

\*/

public void changeStatus(int id) {

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtil.getConnection();

String sql ="update orderform set status=? where id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, 1);

prepareStatement.setInt(2, id);

prepareStatement.execute();

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, null);

}

}

/\*\*

\* 分页查询

\*/

public PageResult queryByPages(Integer currentPage, Integer pageSize, String userName) {

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet resultSet = null;

PreparedStatement prepareStatement2 = null;

ResultSet resultSet2 = null;

PageResult pageResult = null;

try {

connection = JdbcUtil.getConnection();

String whereSql = "";

if(userName!=null && userName.trim().length()>0) {

whereSql = "where logname like '%"+userName+"%' ";

}

String sql ="select \* from orderform "+whereSql+" limit ?,?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, (currentPage-1)\*pageSize);

prepareStatement.setInt(2, pageSize);

resultSet = prepareStatement.executeQuery();

List<OrderForm> orderFormList = new ArrayList<>();

while(resultSet.next()) {

int id = resultSet.getInt("id");

String logname = resultSet.getString("logname");

String mess = resultSet.getString("mess");

float sum = resultSet.getFloat("sum");

int status = resultSet.getInt("status");

OrderForm orderForm = new OrderForm(id,logname,mess,sum,status);

orderFormList.add(orderForm);

}

//查询表中用户总数

String sql2 ="select \* from orderform "+whereSql;

prepareStatement2 = connection.prepareStatement(sql2);

resultSet2 = prepareStatement2.executeQuery();

Long totalRecord = 0L;

while(resultSet2.next()) {

totalRecord = totalRecord + 1;

}

pageResult = new PageResult(orderFormList, totalRecord, pageSize, currentPage);

} catch (Exception e) {

e.printStackTrace();

}finally {

JdbcUtil.close(connection, prepareStatement, resultSet);

JdbcUtil.close(connection, prepareStatement2, resultSet2);

}

return pageResult;

}

}

**-------------------------OrderFormDatabaseImpl**.**java结束-----------------------**

**-----------------------OderFormServiceImpl.java开始---------------------------**

package management.service;

import java.util.List;

import management.bean.OrderForm;

import management.databaseImp.OrderFormDatabaseImpl;

import management.tool.PageResult;

public class OderFormServiceImpl {

public List<OrderForm> queryAll() {

OrderFormDatabaseImpl oderDao = new OrderFormDatabaseImpl();

return oderDao.queryAll();

}

public void delete(int id) {

OrderFormDatabaseImpl oderDao = new OrderFormDatabaseImpl();

oderDao.delete(id);

}

public PageResult queryByPages(Integer currentPage, Integer pageSize, String userName) {

OrderFormDatabaseImpl oderDao = new OrderFormDatabaseImpl();

return oderDao.queryByPages(currentPage, pageSize, userName);

}

public OrderForm queryByLogname(Integer id) {

OrderFormDatabaseImpl oderDao = new OrderFormDatabaseImpl();

return oderDao.queryById(id);

}

public void update(OrderForm oderForm) {

OrderFormDatabaseImpl oderDao = new OrderFormDatabaseImpl();

oderDao.update(oderForm);

}

}

**------------------------OderFormServiceImpl.java结束--------------------------**

**--------------------------** **OrderFormManage.java开始---------------------------**

package management.servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import management.bean.OrderForm;

import management.databaseImp.OrderFormDatabaseImpl;

import management.tool.PageResult;

/\*\*

\* Servlet implementation class OrderFormManage

\*/

@WebServlet("/OrderFormManage")

public class OrderFormManage extends HttpServlet {

private static final long serialVersionUID = 1L;

String oderListUrl = "/management/order/orderList.jsp";

String editOrderUrl = "/management/order/editOrder.jsp";

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public OrderFormManage() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

request.setCharacterEncoding("UTF-8");

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

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

switch (method) {

case "list":

list(request, response);

break;

case "editUI":

editUI(request, response);

break;

case "edit":

edit(request, response);

break;

case "delete":

delete(request, response);

break;

case "changeStatus":

changeStatus(request, response);

break;

default:

break;

}

}

/\*\*

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

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

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

OrderFormDatabaseImpl orderService = new OrderFormDatabaseImpl();

orderService.delete(Integer.parseInt(id));

list(request, response);

}

/\*\*

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void changeStatus(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

OrderFormDatabaseImpl orderService = new OrderFormDatabaseImpl();

orderService.changeStatus(Integer.parseInt(id));

list(request, response);

}

/\*\*

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void edit(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

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

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

String status = request.getParameter("status"); //当添加用户时

OrderFormDatabaseImpl orderService = new OrderFormDatabaseImpl();

OrderForm orderForm = new OrderForm(Integer.parseInt(id),logname,mess, Float.parseFloat(sum),Integer.parseInt(status));

orderService.update(orderForm);

list(request, response);

}

/\*\*

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void editUI(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

if(id!=null&&id.trim().length()>0) {

OrderFormDatabaseImpl orderService = new OrderFormDatabaseImpl();

OrderForm orderForm = orderService.queryById(Integer.parseInt(id));

request.setAttribute("orderForm", orderForm);

}

request.getRequestDispatcher(editOrderUrl).forward(request,response);

}

/\*\*

\* 列出用户表中的信息

\*

\* @param request

\* @param response

\* @throws IOException

\* @throws ServletException

\*/

private void list(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

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

Integer pageSizeInt = 5;

Integer currentPageInt = 1;

if(pageSize!=null&&pageSize.length()>0) {

pageSizeInt = Integer.parseInt(pageSize);

}

if(currentPage!=null&&currentPage.length()>0) {

currentPageInt = Integer.parseInt(currentPage);

}

OrderFormDatabaseImpl orderService = new OrderFormDatabaseImpl();

PageResult pageResult = orderService.queryByPages(currentPageInt,pageSizeInt, userName);

request.setAttribute("pageResult", pageResult);

request.getRequestDispatcher(oderListUrl).forward(request, response);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**---------------------------** **OrderFormManage**.**java结束--------------------------**

退出登录

该模块只有一个名字为HandleExit的servlet控制器，HandleExit负责销毁用户的session对象，导致登录失效。

**-------------------------------**HandleExit**.**java**结束---------------------------**

package myservlet.data;

import java.io.IOException;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class HandleExit

\*/

@WebServlet("/HandleExit")

public class HandleExit extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public HandleExit() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

super.init(config);

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

HttpSession session = request.getSession(true);

session.invalidate(); //销毁用户的session对象

response.sendRedirect(request.getContextPath()+"/index.jsp"); //返回主页

}

/\*\*

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

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**-------------------------------**HandleExit**.**java**结束---------------------------**