武 汉 轻 工 大 学

数学与计算机学院

**UML面向对象建模与设计 课程设计报告**

专 业： 软件工程

班 级： 软工1805

学 号： 1807020003

姓 名： 陶章法

1. 设计简述

**1.实训内容**

超市会在日常运作的过程中，会接到许多订单，同时这些订单往往来自不同的客户。

超市也会从不同的供货商那里购买货物。

这些订单，以及超市购买的货物，需要不同的人员来进行管理。

超市的员工，也可以通过系统进行管理。

如超市经理把普通员工添加进系统。

**2 .UML知识点内容**：UML用例图，顺序图，模块图，类图。

1. **本系统项目结构为和涉及的知识点**：

三层架构jsp+jdbc+servlet

操作数据库的技术：dao (接口) impl（实现类）

业务层(service)：service（接口） impl（实现类）

视图层(jsp) ：页面。

1. 需求分析

**根据需求，smbms(Supermarket bill management system)超市订单管理系统至少应该实现以下几大功能：**

**实现登陆以及账号注销功能；**

**实现管理员账号对普通账户的管理；**

**实现所有账号对订单的管理；**

**实现所有账号对供应商的管理；**

**其中，不同的账号类型代表这个超市不同的职位。**

**而“管理”也应该至少包括“增删改查”。**

1. **供应商信息的管理**，
2. **建立数据库，录入所有的基本信息，进入供应商界面便可看见所有用户信息的表单。**
3. **根据供应商姓名，可以查讯单个供应商的信息。**
4. **点击添加按钮，可以加入新的信息**
5. **可以以表单的形式查看单个供应商信息。**
6. **可以修改供应商的单个信息5删除供应商信息**
7. **账单信息的管理**，
8. **建立数据库，录入所有的基本信息，进入账单界面便可看见所有账单信息的表单。**
9. **根据供货名称，可以查讯单个账单的信息。**
10. **点击添加按钮，可以加入新的信息。**
11. **可以以表单的形式查看单个供应商信息。**
12. **可以修改表单的单个信息、删除表单信息**
13. 系统功能

**、1：系统应该包括以下几个模块：**

**登陆**

**修改密码**

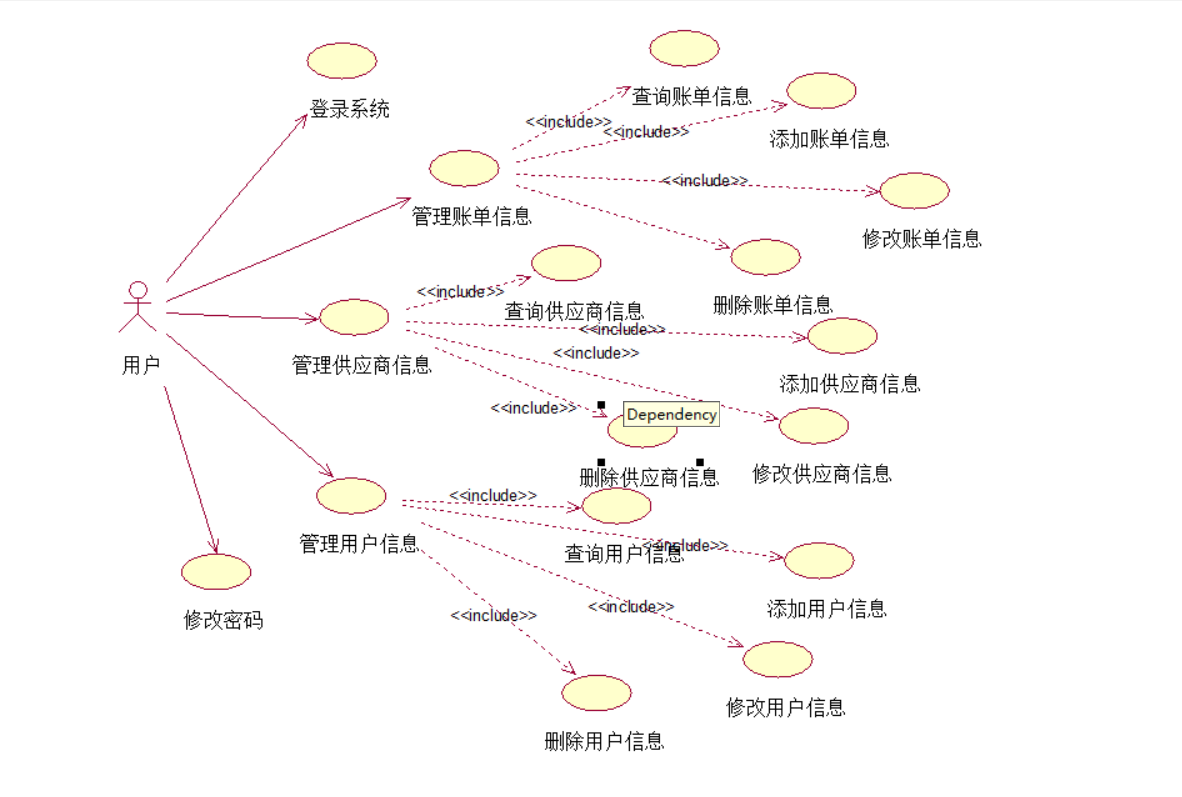
**用户管理**

**订单管理**

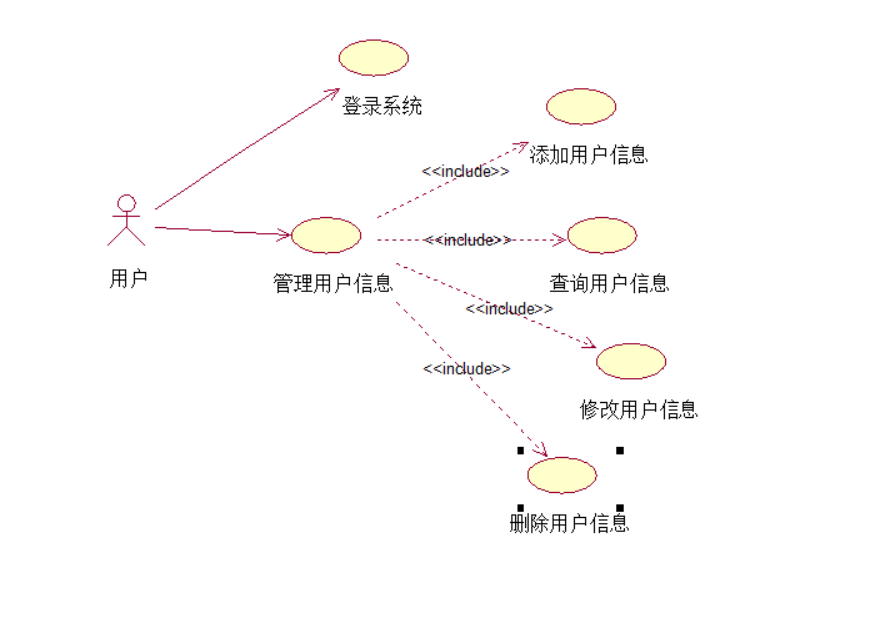
**供应商管理**

**二：超市管理系统用例图如下所示：**

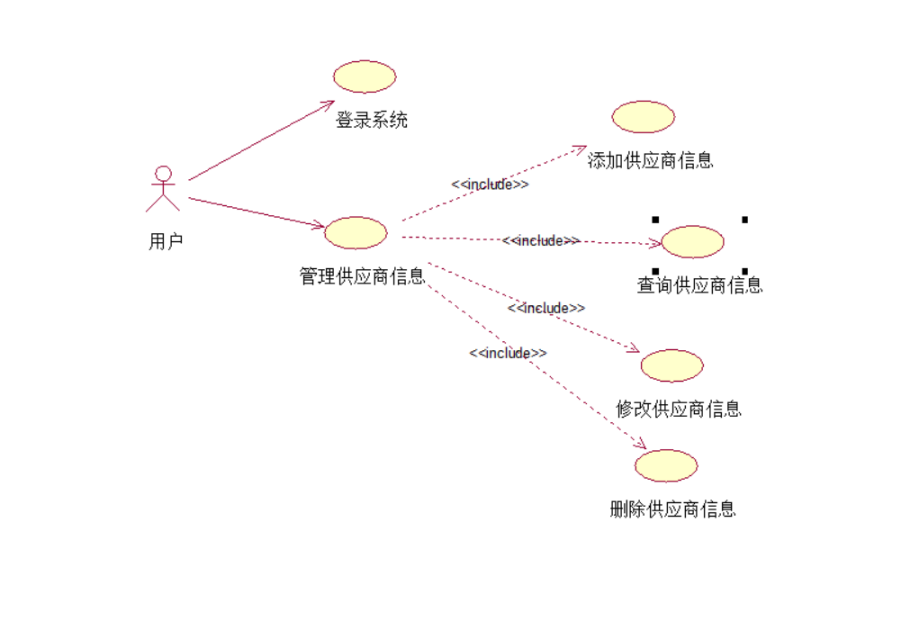
**1.总用例图如下所示：**



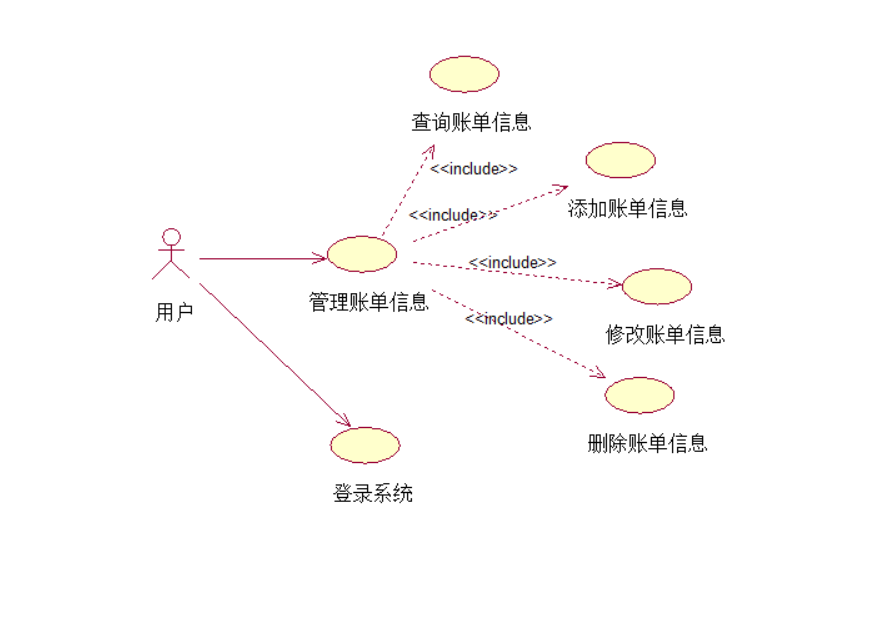
**2.用户分用例图：**



**3.供应商:分用例图**

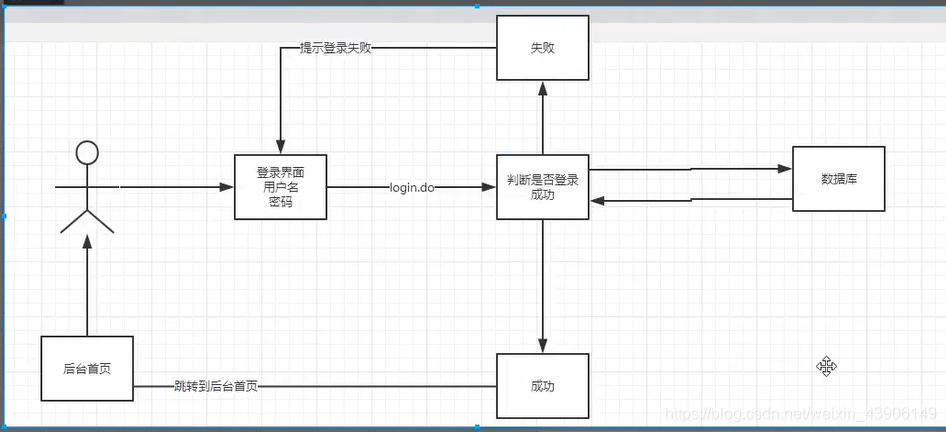


**4：账单分用例图：**

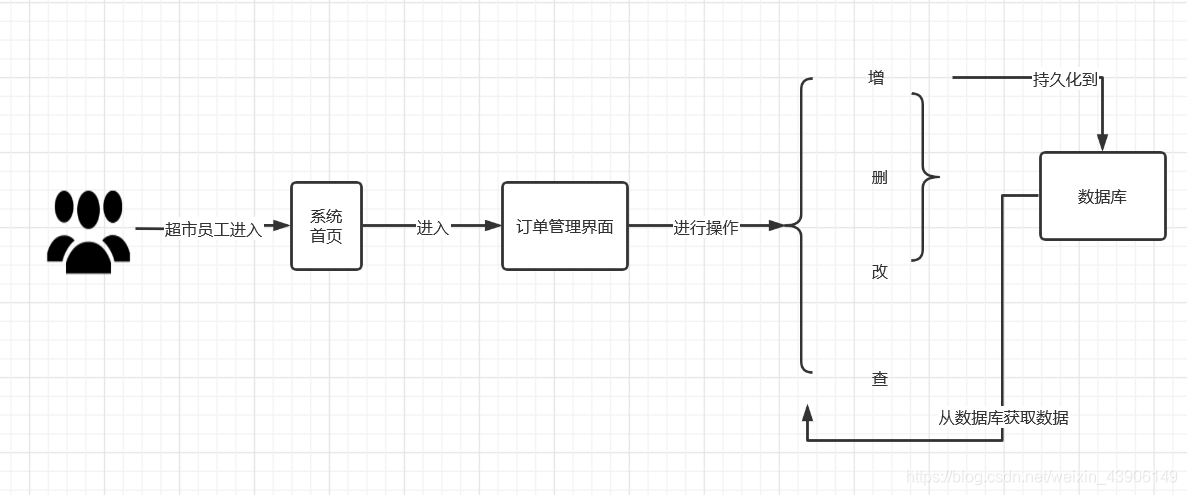


**流程示意图：**

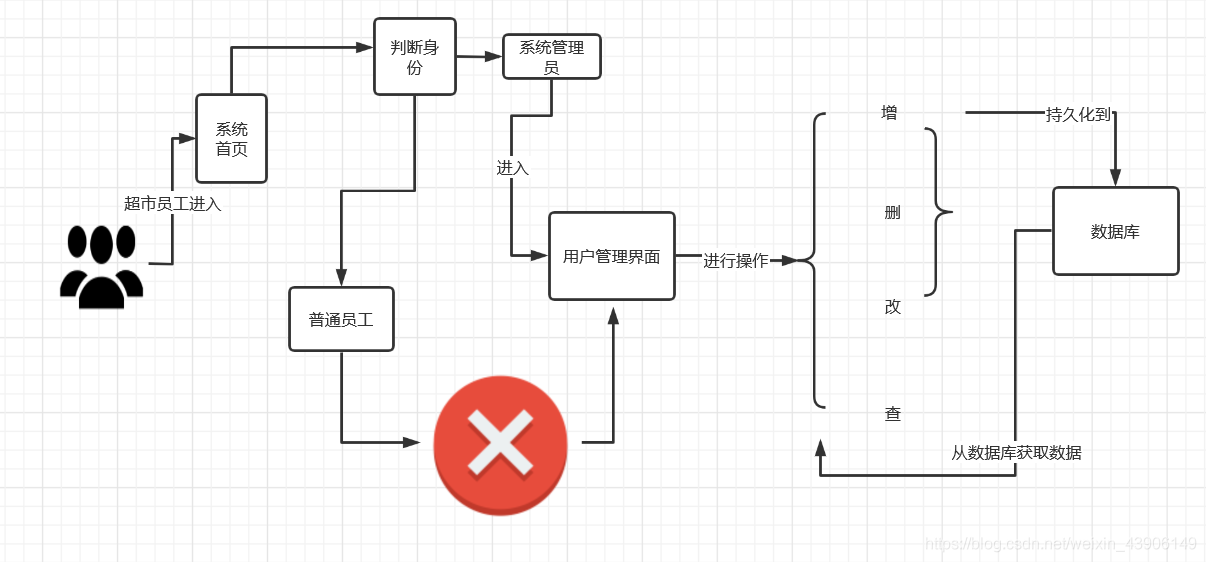
**登录：**



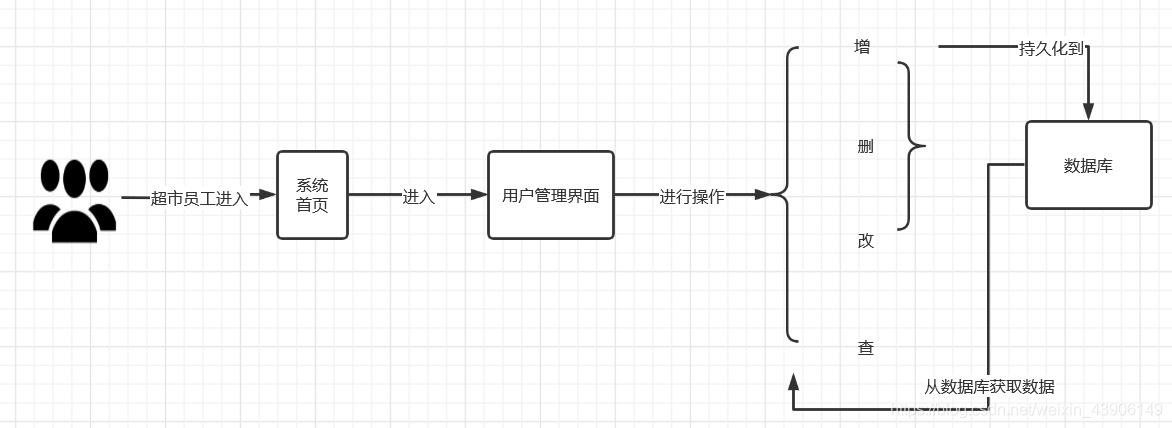
用户管理模块：



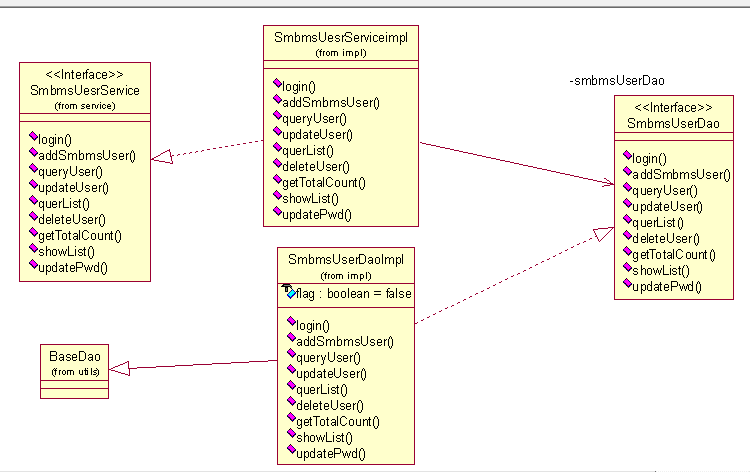
订单管理模块：

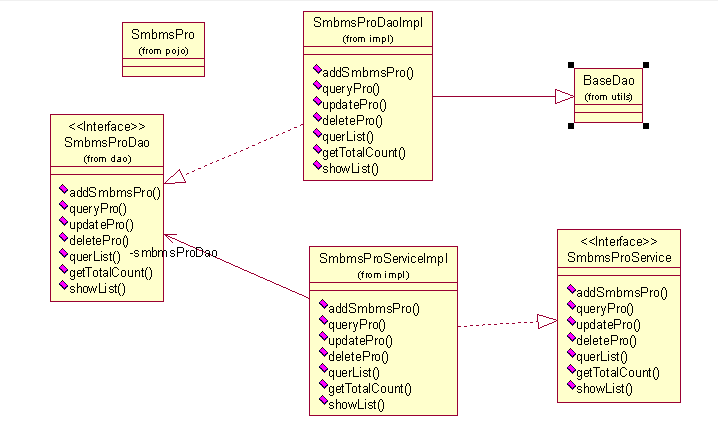


用户管理模块：



**四：类图如下图所示：**





1. 项目实施步骤

1：创建超市管理系统数据库的表：

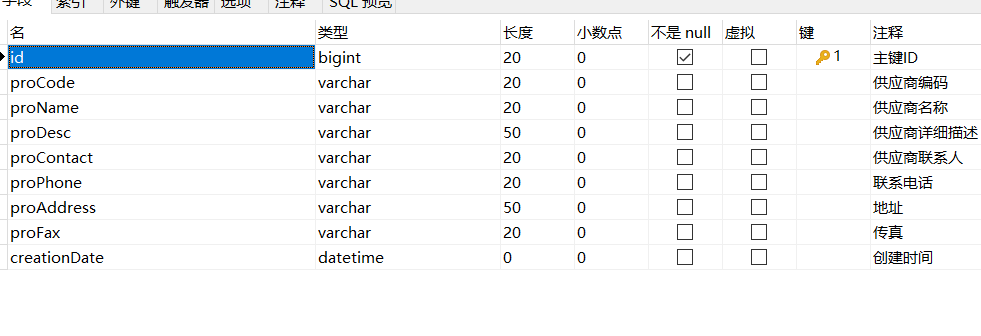
smbms\_user表结构如下：



smbms\_role表结构如下：



smbms\_provider表结构如下：



smbms\_bill表结构如下：



smbms\_address表结构如下：



二：用户信息管理，账单信息管理，和供应商管理都是基于对数据库的增，删，改，查，以下是对于供应商三层架构的代码实现：

1：该系统的工具类如下所示：

PageSupport类代码如下所示：

package org.qf.utils;  
public class PageSupport {  
 //当前页  
 private int currentPageNo=1;  
 //总数量  
 private int totalCount=0;  
 //页面容量  
 private int pageSize=0;  
//总页数  
 private int totalPageCount=1;  
  
 public int getCurrentPageNo() {  
 return currentPageNo;  
 }  
 public void setCurrentPageNo(int currentPageNo) {  
 if(currentPageNo>0){  
 this.currentPageNo = currentPageNo;  
 }  
  
 }  
 public int getTotalCount() {  
 return totalCount;  
 }  
 //总数量  
 public void setTotalCount(int totalCount) {  
 if(totalCount>0){  
 this.totalCount = totalCount;  
 //总页数  
 setTotalPageCountByRs();  
 }  
 }  
 //总页数  
 public void setTotalPageCountByRs(){  
 if(this.totalCount%this.pageSize==0){  
 this.totalPageCount=this.totalCount/this.pageSize;  
 }else if(this.totalPageCount%this.pageSize!=0){  
 this.totalPageCount=this.totalCount/this.pageSize+1;  
 }else{  
 this.totalPageCount=0;  
 }  
 }  
  
 public int getPageSize() {  
 return pageSize;  
 }  
 public void setPageSize(int pageSize) {  
 if(pageSize>0){  
 this.pageSize = pageSize;  
 }  
 }  
 public int getTotalPageCount() {  
 return totalPageCount;  
 }  
 public void setTotalPageCount(int totalPageCount) {  
 this.totalPageCount = totalPageCount;  
 }  
}

**BaseDao类来实现对于数据库的增删改，其代码如下所示：**

package org.qf.utils;  
import java.io.IOException;  
import java.io.InputStream;  
import java.sql.\*;  
import java.util.Properties;  
  
public class BaseDao {  
 private Connection connection;  
 private PreparedStatement preparedStatement;  
 private ResultSet resultSet;  
 private static String *driver*;  
 private static String *url*;  
 private static String *username*;  
 private static String *password*;  
  
 static {  
 *initial*();  
 }  
 //加载配置文件  
 public static void initial(){  
 Properties properties=new Properties();  
 InputStream inputStream=BaseDao.class.getClassLoader().getResourceAsStream("database.properties");  
 try {  
 properties.load(inputStream);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 *driver*=properties.getProperty("driver");  
 *url*=properties.getProperty("url");  
 *username*=properties.getProperty("username");  
 *password*=properties.getProperty("password");  
 }  
  
 //创建连接  
 public boolean getConnection(){  
 try {  
 Class.*forName*(*driver*);  
 connection= DriverManager.*getConnection*(*url*,*username*,*password*);  
 } catch (ClassNotFoundException e) {  
 e.printStackTrace();  
 return false;  
 } catch (SQLException e) {  
 e.printStackTrace();  
 return false;  
 }  
 return true;  
 }  
  
  
 //增 删 改  
 public int exectueUpdate(String sql,Object[] params){  
 int count=0;  
 if(this.getConnection()){  
 //执行命令  
 try {  
 preparedStatement=connection.prepareStatement(sql);  
 for (int i=0;i<params.length;i++){  
 preparedStatement.setObject(i+1,params[i]);  
 }  
 //执行命令  
 count= preparedStatement.executeUpdate();  
  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 return count;  
  
 }  
 //查  
 public ResultSet executeSQL(String sql,Object[] params){  
 if(this.getConnection()){  
 try {  
 preparedStatement=connection.prepareStatement(sql);  
 for (int i=0;i<params.length;i++){  
 preparedStatement.setObject(i+1,params[i]);  
 }  
 resultSet=preparedStatement.executeQuery();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 return resultSet;  
  
 }  
}

**用户在Dao层的方法，用于数据库的操作：**

**SmbmsUserDao接口内代码如下：**

**package org.qf.dao;  
import org.qf.pojo.SmbmsUser;  
import java.sql.SQLException;  
import java.util.List;  
  
public interface SmbmsUserDao {  
 public abstract boolean login(String username,String pwd) throws SQLException;  
 // List<SmbmsUser> queryList() throws Exception;  
 int addSmbmsUser(SmbmsUser smbmsUser) throws Exception;  
 //通过用户名进行模糊查询  
 //List<SmbmsUser> queryUserByName(String username) throws Exception;  
 SmbmsUser queryUser(SmbmsUser smbmsUser) throws Exception;  
 int updateUser (SmbmsUser smbmsUser) throws Exception;  
 List<SmbmsUser> querList(String username,int pageNo,int pageSize) throws Exception;  
 int deleteUser(SmbmsUser smbmsUser) throws Exception;  
List<SmbmsUser> showList(int pageNo,int pageSize) throws Exception;  
  
  
 int getTotalCount(String username) throws Exception;  
  
 int updatePassworld(String username,String newPassword,String oldPassword) throws Exception;  
}**

**SmbmsUserDao接口的实现类代码如下：**

**package org.qf.dao.impl;  
import org.qf.dao.SmbmsUserDao;  
import org.qf.pojo.SmbmsUser;  
import org.qf.utils.BaseDao;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
public class SmbmsUserDaoImpl extends BaseDao implements SmbmsUserDao {  
 @Override  
 public boolean login(String username, String pwd) throws SQLException {  
 boolean falg=false;  
 String sql="select \* from smbms\_user where userName=? and userPassword=?";  
 Object[] params={username,pwd};  
 ResultSet resultSet=this.executeSQL(sql,params);  
 while (resultSet.next()){  
 falg=true;  
 }  
 return falg;  
 }  
 //增加功能  
 @Override  
 public int addSmbmsUser(SmbmsUser smbmsUser) throws Exception {  
 String sql="insert into smbms\_user(userCode, userName, userPassword, age, gender, birthday, phone, address, userRole, creationDate) values(?,?,?,?,?,?,?,?,?,?)";  
 Object[] params={smbmsUser.getUserCode(),smbmsUser.getUserName(),smbmsUser.getUserPassword(),smbmsUser.getAge(),smbmsUser.getGender(),smbmsUser.getBirthday(),smbmsUser.getPhone(),smbmsUser.getAddress(),smbmsUser.getUserRole(),smbmsUser.getCreatetionDate()};  
 int count=this.exectueUpdate(sql,params);  
 return count;  
 }  
  
 @Override  
 public SmbmsUser queryUser(SmbmsUser smbmsUser) throws Exception {  
 SmbmsUser u=new SmbmsUser();  
 String sql="select \* from smbms\_user where id=?";  
 Object[] params={smbmsUser.getId()};  
 ResultSet resultSet = this.executeSQL(sql, params);  
 while (resultSet.next()){  
  
 u.setId(resultSet.getLong("id"));  
 u.setUserCode(resultSet.getString("userCode"));  
 u.setUserName(resultSet.getString("userName"));  
 u.setAge(resultSet.getInt("age"));  
 u.setBirthday(resultSet.getDate("birthday"));  
 u.setGender(resultSet.getInt("gender"));  
 u.setPhone(resultSet.getString("phone"));  
 u.setAddress(resultSet.getString("address"));  
 u.setUserRole(resultSet.getInt("userRole"));  
  
 }  
  
 return u;  
 }  
  
 @Override  
 public int updateUser(SmbmsUser smbmsUser) throws Exception {  
 String sql="update smbms\_user set userName=?,gender=?,birthday=?,phone=?,address=?,userRole=? where id=?";  
 Object[] params={smbmsUser.getUserName(),smbmsUser.getGender(),smbmsUser.getBirthday(),smbmsUser.getPhone(),smbmsUser.getAddress(),smbmsUser.getUserRole(),smbmsUser.getId()};  
 return this.exectueUpdate(sql, params);  
 }  
  
 @Override  
 public List<SmbmsUser> querList(String username,int pageNo,int pageSize) throws Exception {  
 List<SmbmsUser> list=new ArrayList<SmbmsUser>();  
 String sql="select \* from smbms\_user";  
 Object[] params={};  
 StringBuilder stringBuilder=new StringBuilder();  
 if("".equals(username) || username==null){  
 stringBuilder.append(sql);  
 }else{  
 stringBuilder.append(sql);  
 stringBuilder.append(" where userName like '%"+username+"%'");  
 }  
 stringBuilder.append(" limit "+(pageNo-1)\*pageSize+","+pageSize+"");  
 System.*out*.println(stringBuilder.toString()+"===============================");  
 ResultSet resultSet = this.executeSQL(stringBuilder.toString(), params);  
 while (resultSet.next()){  
 SmbmsUser u=new SmbmsUser();  
 u.setId(resultSet.getLong("id"));  
 u.setUserCode(resultSet.getString("userCode"));  
 u.setUserName(resultSet.getString("userName"));  
 u.setAge(resultSet.getInt("age"));  
 u.setBirthday(resultSet.getDate("birthday"));  
 u.setGender(resultSet.getInt("gender"));  
 u.setPhone(resultSet.getString("phone"));  
 u.setAddress(resultSet.getString("address"));  
 u.setUserRole(resultSet.getInt("userRole"));  
 list.add(u);  
 }  
 return list;  
 }  
  
 @Override  
 public int deleteUser(SmbmsUser smbmsUser) throws Exception {  
 int count=0;  
 String sql="delete from smbms\_user where id=?";  
 Object[] params={smbmsUser.getId()};  
 count=this.exectueUpdate(sql,params);  
 return count;  
 }  
  
 @Override  
 public List<SmbmsUser> showList(int pageNo, int pageSize) throws Exception {  
 List<SmbmsUser> list=new ArrayList<SmbmsUser>();  
 String sql="select id, usercode, username, age, gender, birthday, phone, userrole from smbms\_user limit ?,?";  
 Object[] params={(pageNo-1)\*pageSize,pageSize};  
 ResultSet resultSet = this.executeSQL(sql, params);  
 while (resultSet.next()){  
 SmbmsUser u=new SmbmsUser();  
 u.setId(resultSet.getLong("id"));  
 u.setUserCode(resultSet.getString("userCode"));  
 u.setAge(resultSet.getInt("age"));  
 u.setUserName(resultSet.getString("userName"));  
 u.setBirthday(resultSet.getDate("birthday"));  
 u.setGender(resultSet.getInt("gender"));  
 u.setPhone(resultSet.getString("phone"));  
 u.setUserRole(resultSet.getInt("userRole"));  
 list.add(u);  
 }  
  
 return list;  
 }  
  
 @Override  
 public int getTotalCount(String username) throws Exception {  
 int count=0;  
 String sql="SELECT COUNT(\*) FROM smbms\_user";  
 Object[] params={};  
 StringBuilder stringBuilder=new StringBuilder();  
 if("".equals(username) || username==null){  
 stringBuilder.append(sql);  
 }else{  
 stringBuilder.append(sql);  
 stringBuilder.append(" where userName like '%"+username+"%'");  
 }  
 System.*out*.println(stringBuilder.toString()+"===============================");  
 ResultSet resultSet = this.executeSQL(stringBuilder.toString(), params);  
 while (resultSet.next()) {  
 count = resultSet.getInt(1);  
 }  
 return count;  
 }  
  
 @Override  
 public int updatePassworld(String username,String newPassword,String oldPassword) throws Exception {  
 BaseDao baseDao=new BaseDao();  
 String sql="update smbms\_user set userPassword=? where userName=? and userPassword=?";  
 Object[] params={newPassword,username,oldPassword};  
 return baseDao.exectueUpdate(sql,params);  
 }  
  
  
}**

**用户对象类代码如下：**

**package org.qf.pojo;  
import java.util.Date;  
  
public class SmbmsUser {  
 private Long id;  
 private String userCode;  
 private String userName;  
 private String userPassword;  
 private int age;  
 private int gender;  
 private String phone;  
 private String address;  
 private Date birthday;  
 private int userRole;  
 private Date createtionDate;  
  
 public Long getId() {  
 return id;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public String getUserCode() {  
 return userCode;  
 }  
  
 public void setUserCode(String userCode) {  
 this.userCode = userCode;  
 }  
  
 public String getUserName() {  
 return userName;  
 }  
  
 public void setUserName(String userName) {  
 this.userName = userName;  
 }  
  
 public String getUserPassword() {  
 return userPassword;  
 }  
 public void setUserPassword(String userPassword) {  
 this.userPassword = userPassword;  
 }  
 public int getAge() {  
 return age;  
 }  
 public void setAge(int age) {  
 this.age = age;  
 }  
 public int getGender() {  
 return gender;  
 }  
 public void setGender(int gender) {  
 this.gender = gender;  
 }  
 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 Date getBirthday() {  
 return birthday;  
 }  
  
 public void setBirthday(Date birthday) {  
 this.birthday = birthday;  
 }  
  
 public int getUserRole() {  
 return userRole;  
 }  
  
 public void setUserRole(int userRole) {  
 this.userRole = userRole;  
 }  
  
 public Date getCreatetionDate() {  
 return createtionDate;  
 }  
  
 public void setCreatetionDate(Date createtionDate) {  
 this.createtionDate = createtionDate;  
 }  
}**

**service层接口如下：**

**SmbmsUserService代码如下：**

**package org.qf.service;  
import org.qf.pojo.SmbmsUser;  
import java.sql.SQLException;  
import java.util.List;  
public interface SmbmsUserService {  
 boolean login(String username,String pwd) throws SQLException;  
 //List<SmbmsUser> queryList() throws Exception;  
 */\*\*  
 \* 增加功能  
 \* @param smbmsUser  
 \* @return  
 \* @throws Exception  
 \*/* int addSmbmsUser(SmbmsUser smbmsUser) throws Exception;  
 //List<SmbmsUser> queryUserByName(String username) throws Exception;  
 SmbmsUser queryUser(SmbmsUser smbmsUser) throws Exception;  
 int updateUser (SmbmsUser smbmsUser) throws Exception;  
 List<SmbmsUser> querList(String username,int pageNo,int pageSize) throws Exception;  
 int deleteUser(SmbmsUser smbmsUser) throws Exception;  
 int getTotalCount(String username) throws Exception;  
 List<SmbmsUser> showList(int pageNo,int pageSize) throws Exception;  
  
 int updatePassworld(String username,String newPassword,String oldPassword) throws Exception;  
}**

**SmbmsUserService接口的实现类如下所示：**

**package org.qf.service.impl;  
import org.qf.dao.SmbmsUserDao;  
import org.qf.dao.impl.SmbmsUserDaoImpl;  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import java.sql.SQLException;  
import java.util.List;  
  
public class SmbmsUserServiceImpl implements SmbmsUserService {  
 //调用dao层  
 private SmbmsUserDao smbmsUserDao=new SmbmsUserDaoImpl();  
 @Override  
 public boolean login(String username, String pwd) throws SQLException {  
 return smbmsUserDao.login(username,pwd);  
 }  
  
// @Override  
// public List<SmbmsUser> queryList() throws Exception {  
// return smbmsUserDao.queryList();  
// }  
  
 @Override  
 public int addSmbmsUser(SmbmsUser smbmsUser) throws Exception {  
 return smbmsUserDao.addSmbmsUser(smbmsUser);  
 }  
  
// @Override  
// public List<SmbmsUser> queryUserByName(String username) throws Exception {  
// return smbmsUserDao.queryUserByName(username);  
// }  
  
 @Override  
 public SmbmsUser queryUser(SmbmsUser smbmsUser) throws Exception {  
 return smbmsUserDao.queryUser(smbmsUser);  
 }  
  
 @Override  
 public int updateUser(SmbmsUser smbmsUser) throws Exception {  
 return smbmsUserDao.updateUser(smbmsUser);  
 }  
  
 @Override  
 public List<SmbmsUser> querList(String username,int pageNo,int pageSize) throws Exception {  
 return smbmsUserDao.querList(username,pageNo,pageSize);  
 }  
  
 @Override  
 public int deleteUser(SmbmsUser smbmsUser) throws Exception {  
 return smbmsUserDao.deleteUser(smbmsUser);  
 }  
  
 @Override  
 public int getTotalCount(String username) throws Exception {  
 return smbmsUserDao.getTotalCount(username);  
 }  
  
 @Override  
 public List<SmbmsUser> showList(int pageNo, int pageSize) throws Exception {  
 return smbmsUserDao.showList(pageNo,pageSize);  
 }  
  
 @Override  
 public int updatePassworld(String username, String newPassword,String oldPassword) throws Exception {  
 return smbmsUserDao.updatePassworld(username,newPassword,oldPassword);  
 }  
  
  
}**

**servlet代码如下：**

**1实现登录功能的servlet代码如下：**

**package org.qf.servlet;  
  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import java.io.IOException;  
import java.sql.SQLException;  
  
public class SmbmsLoginServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req, resp);  
 }  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 String username = req.getParameter("username");  
  
 String pwd = req.getParameter("password");  
 System.*out*.println(username);  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
 boolean admin = false;  
 try {  
  
 admin = smbmsUserService.login(username,pwd);  
  
 if(admin){  
 HttpSession session=req.getSession();  
 session.setAttribute("sessionName",username);  
 req.getRequestDispatcher("welcome.jsp").forward(req,resp);  
 }else{  
 resp.sendRedirect("login.jsp");  
 }  
  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
  
  
 }  
}**

**添加用户信息的代码如下：**

**package org.qf.servlet;  
  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
public class AddServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req, resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 try {  
 //从界面接收输入的值  
 String userCode=req.getParameter("userId");  
 String userName=req.getParameter("userName");  
 String pwd=req.getParameter("userpassword");  
 String gender=req.getParameter("gender");  
 System.*out*.println(userCode);  
 int sex=0;  
 if(gender.equals("男")){  
 sex=2;  
 }else{  
 sex=1;  
 }  
 String data=req.getParameter("data");  
 //字符串怎么转Date类型 2000-11-12 2000/11/12  
 SimpleDateFormat format=new SimpleDateFormat("yyyy-mm-dd");  
 Date birthday=format.parse(data);  
  
 String userphone=req.getParameter("userphone");  
 String userAddress=req.getParameter("userAddress");  
 String userlei=req.getParameter("userlei");  
 //调用service层  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
 SmbmsUser smbmsUser=new SmbmsUser();  
 //注意，类型转换  
 smbmsUser.setUserCode(userCode);  
 smbmsUser.setUserName(userName);  
 smbmsUser.setUserPassword(pwd);  
 smbmsUser.setGender(sex);  
 smbmsUser.setBirthday(birthday);  
 smbmsUser.setPhone(userphone);  
 smbmsUser.setAddress(userAddress);  
 smbmsUser.setUserRole(Integer.*parseInt*(userlei));  
 System.*out*.println(smbmsUser);  
 int i = smbmsUserService.addSmbmsUser(smbmsUser);  
 if(i>0){  
 //成功  
 //页面跳转  
 req.getRequestDispatcher("/userlist").forward(req,resp);  
 }else{  
 resp.sendRedirect("userAdd.jsp");  
 }  
  
  
 } catch (ParseException e) {  
 e.printStackTrace();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
  
 }  
  
}**

**删除用户信息的代码如下：**

**package org.qf.servlet;  
  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
public class DeleteUserServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req, resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 //获取id  
 Long id=Long.*parseLong*(req.getParameter("id"));  
 SmbmsUser smbmsUser=new SmbmsUser();  
 smbmsUser.setId(id);  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
 try {  
 int i = smbmsUserService.deleteUser(smbmsUser);  
 if(i>0){  
 req.getRequestDispatcher("/userlist").forward(req,resp);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
}**

**查询操作代码如下：**

**package org.qf.servlet;  
  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
public class userQueryServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req, resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 String id=req.getParameter("id");  
 SmbmsUser smbmsUser=new SmbmsUser();  
 smbmsUser.setId(Long.*parseLong*(id));  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
 SmbmsUser smbmsUser1 = null;  
 try {  
 smbmsUser1 = smbmsUserService.queryUser(smbmsUser);  
 if(smbmsUser1!=null){  
 req.setAttribute("u",smbmsUser1);  
 req.getRequestDispatcher("userUpdate.jsp").forward(req,resp);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}**

**SmbmsUserByIdServlet代码如下：**

**package org.qf.servlet;  
  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
public class SmbmsUserByIdServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req, resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 String id=req.getParameter("id");  
 SmbmsUser smbmsUser=new SmbmsUser();  
 smbmsUser.setId(Long.*parseLong*(id));  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
 SmbmsUser smbmsUser1 = null;  
 try {  
 smbmsUser1 = smbmsUserService.queryUser(smbmsUser);  
 if(smbmsUser1!=null){  
 req.setAttribute("users",smbmsUser1);  
 req.getRequestDispatcher("userView.jsp").forward(req,resp);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}**

**SmbmsUserByNameServlet代码如下：**

**package org.qf.servlet;  
  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
import org.qf.utils.PageSupport;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.util.List;  
  
public class SmbmsUserByNameServlet extends HttpServlet {  
  
  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req, resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 //获取用户名  
 String name=req.getParameter("username");  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
  
 //当前页  
 String currPage=req.getParameter("pageNo");  
 if(currPage==null){  
 currPage="1";  
 }  
 //当前页码  
 int pageNo=Integer.*parseInt*(currPage);  
  
 //总数量  
 try {  
 int totalCount=smbmsUserService.getTotalCount(name);  
 //每页显示几条数据  
 int pageSize=3;  
 //计算总页数  
 PageSupport pageSupport=new PageSupport();  
 //当前页  
 pageSupport.setCurrentPageNo(pageNo);  
 //页面容量  
 pageSupport.setPageSize(pageSize);  
 //总数量  
 pageSupport.setTotalCount(totalCount);  
 //总页数  
 int totalPage=pageSupport.getTotalPageCount();  
  
 //控制首页和尾页  
 if(pageNo<1){  
 pageNo=1;  
 }else if(pageNo>totalPage) {  
 pageNo = totalPage;  
 }  
 List<SmbmsUser> list = smbmsUserService.querList(name,pageNo,pageSize);  
 if(list.size()>0){  
 //转发  
 req.setAttribute("username",name);  
 req.setAttribute("page",pageSupport);  
 req.setAttribute("list",list);  
 req.getRequestDispatcher("userList.jsp").forward(req,resp);  
 }  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
}**

**SmbmsUserServlet代码如下：**

**package org.qf.servlet;  
  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
import org.qf.utils.PageSupport;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.util.List;  
  
public class SmbmsUserServlet extends HttpServlet {  
  
  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req,resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
  
 //当前页  
 String currPage=req.getParameter("pageNo");  
 if(currPage==null){  
 currPage="1";  
 }  
 //当前页码  
 int pageNo=Integer.*parseInt*(currPage);  
  
 //总数量  
 try {  
 int totalCount=smbmsUserService.getTotalCount(null);  
 //每页显示几条数据  
 int pageSize=3;  
 //计算总页数  
 PageSupport pageSupport=new PageSupport();  
 //当前页  
 pageSupport.setCurrentPageNo(pageNo);  
 //页面容量  
 pageSupport.setPageSize(pageSize);  
 //总数量  
 pageSupport.setTotalCount(totalCount);  
 //总页数  
 int totalPage=pageSupport.getTotalPageCount();  
  
 //控制首页和尾页  
 if(pageNo<1){  
 pageNo=1;  
 }else if(pageNo>totalPage) {  
 pageNo = totalPage;  
 }  
 List<SmbmsUser> list = smbmsUserService.querList(null,pageNo,pageSize);  
 if(list.size()>0){  
 //转发  
 req.setAttribute("page",pageSupport);  
 req.setAttribute("list",list);  
 req.getRequestDispatcher("userList.jsp").forward(req,resp);  
 }  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
  
  
  
 }  
}**

**UpdateUserServlet代码如下：**

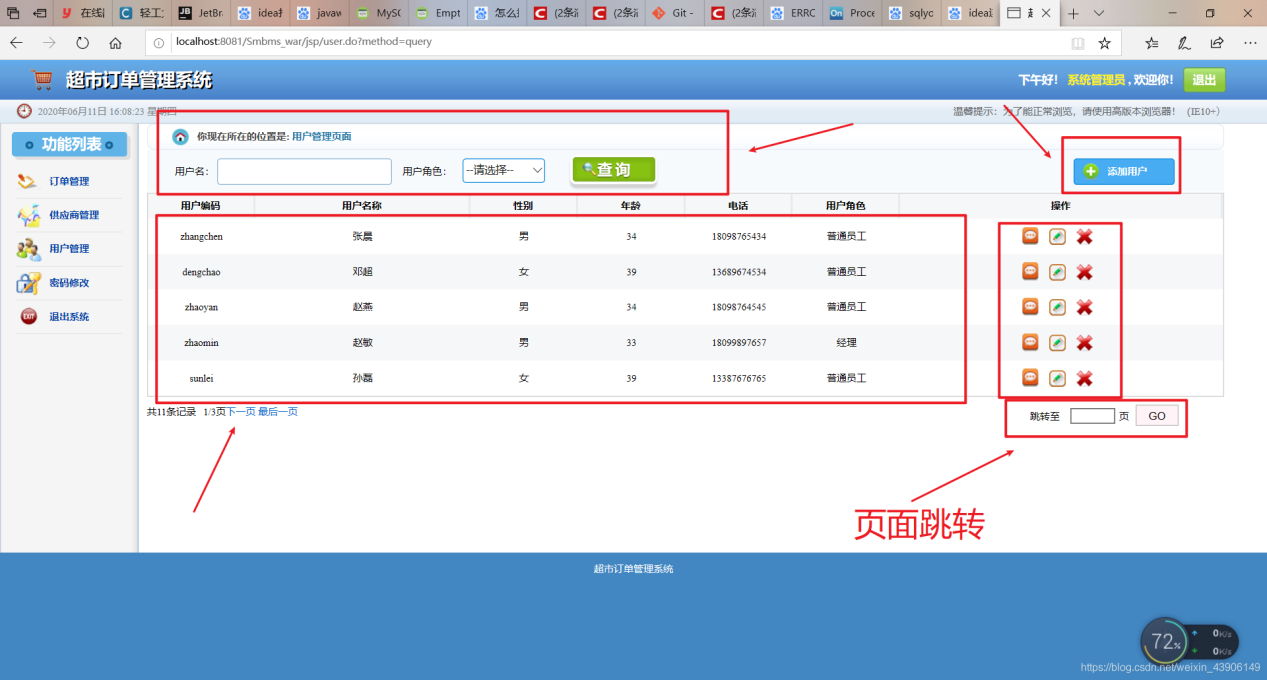
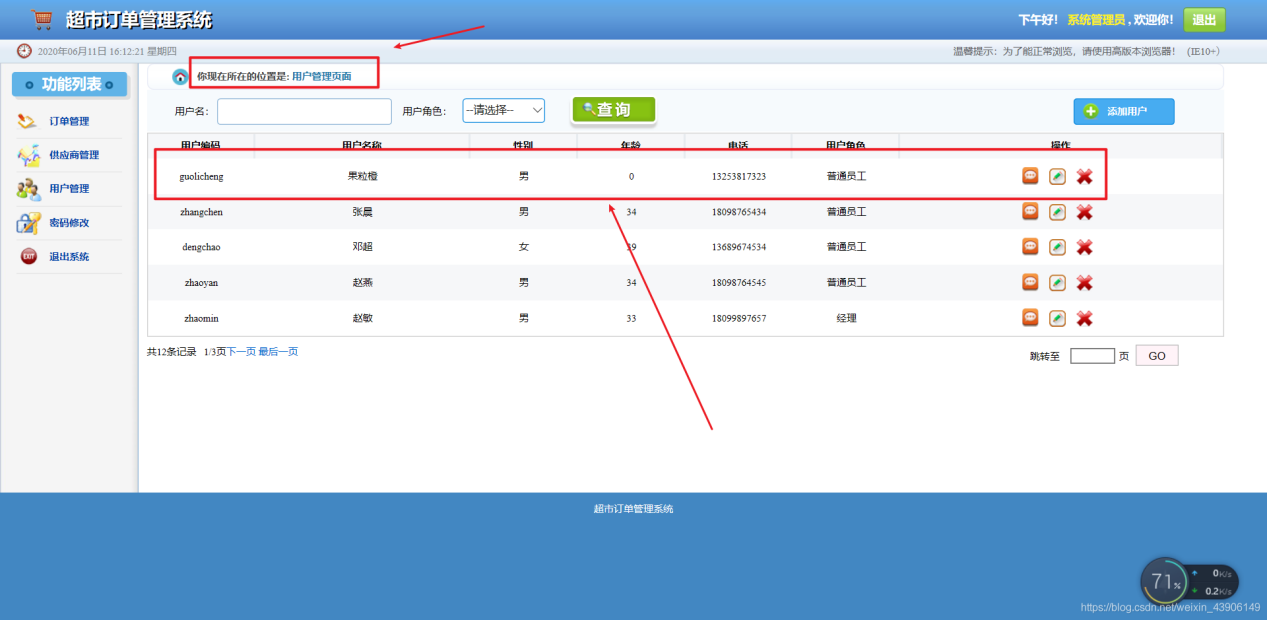
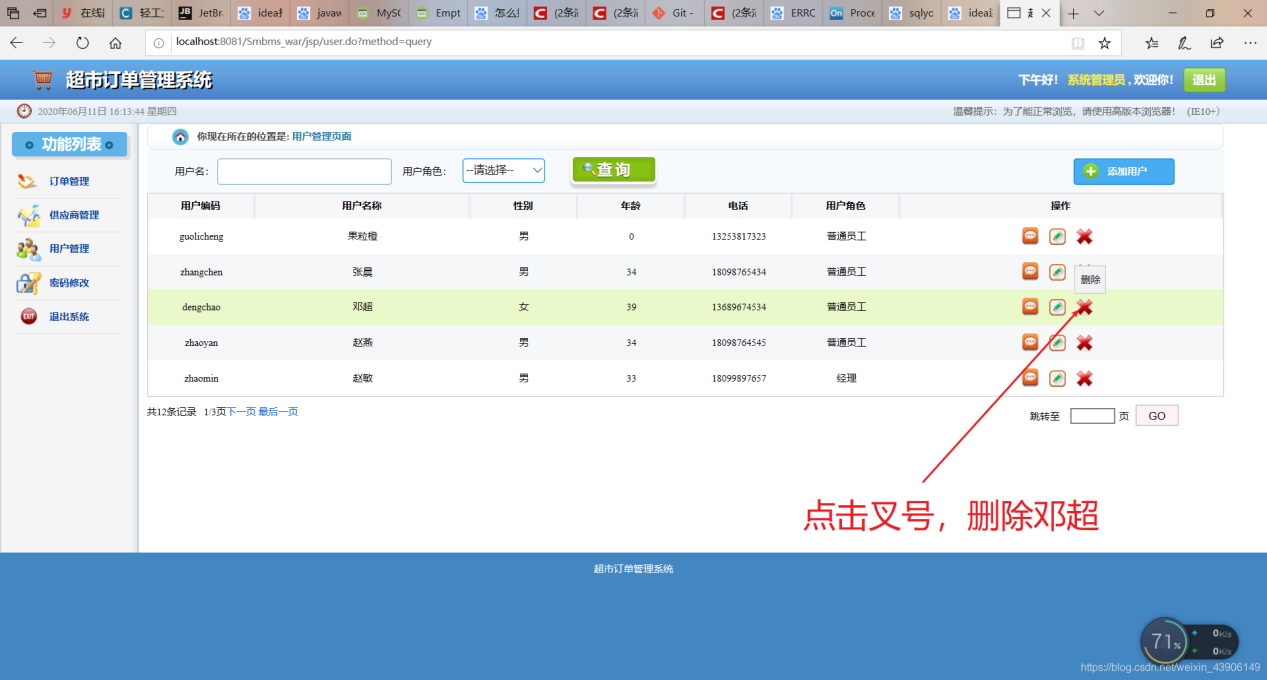
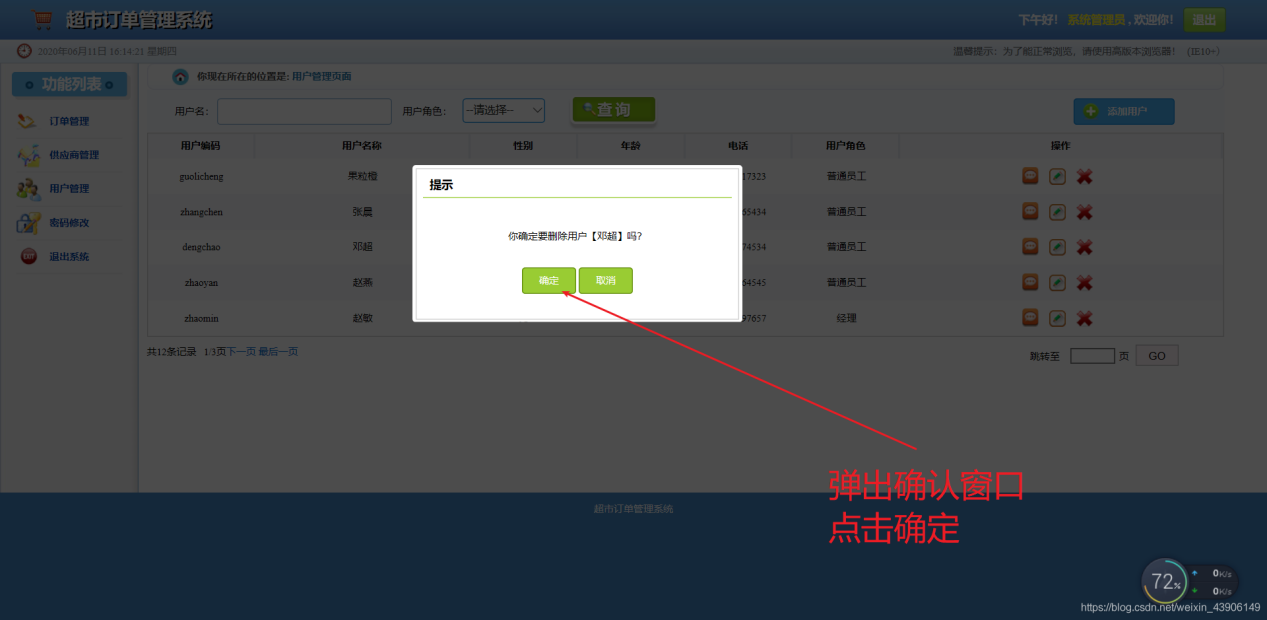
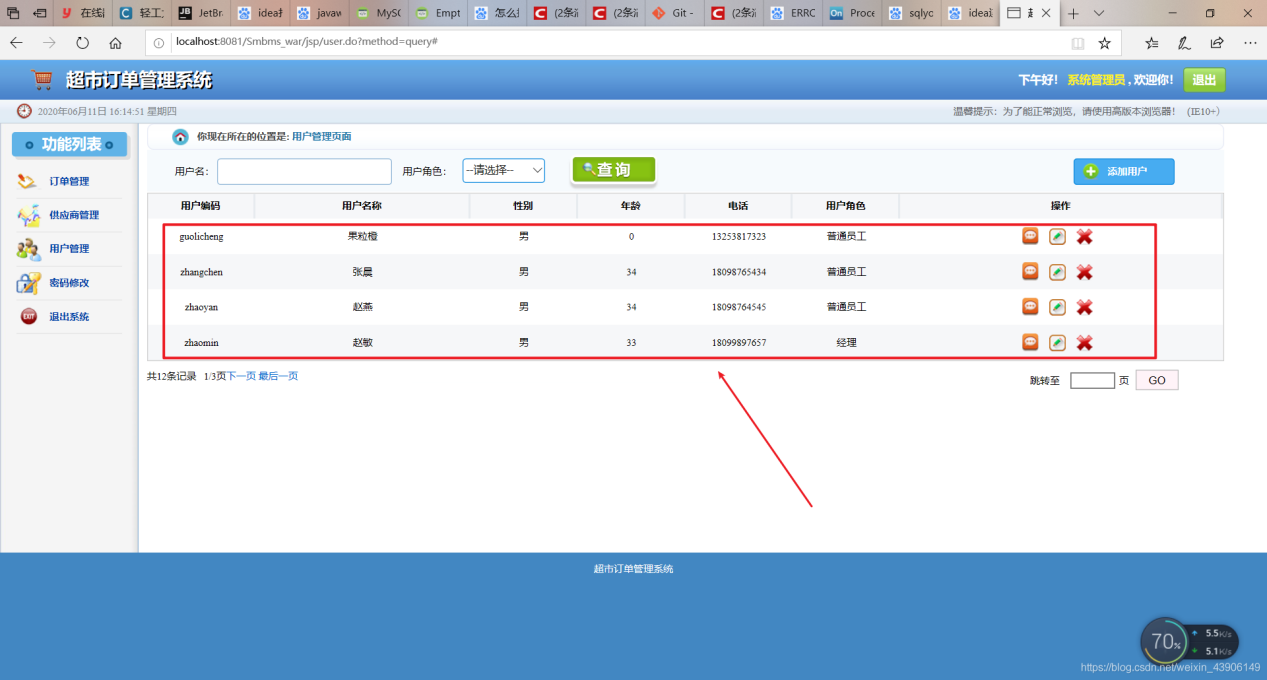
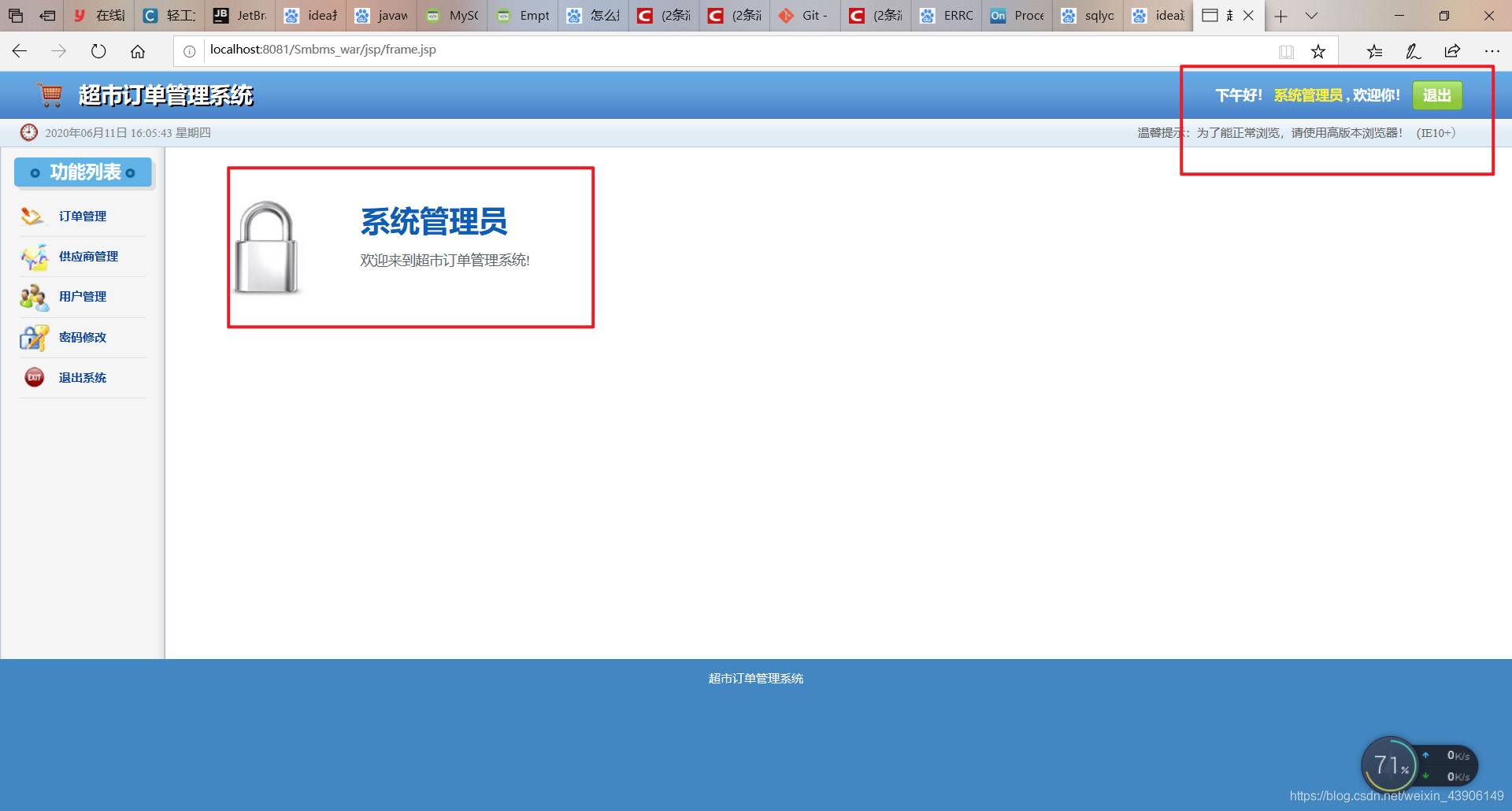
**package org.qf.servlet;  
  
  
  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
public class UpdateUserServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req, resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
  
 Long id=Long.*parseLong*(req.getParameter("id"));  
 //获取界面输入的值  
 String username=req.getParameter("username");  
 Integer gender=Integer.*parseInt*(req.getParameter("gender"));  
 //出生日期  
 String time=req.getParameter("data");  
 //把string类型转为date  
 SimpleDateFormat simpleDateFormat=new SimpleDateFormat("yyyy-MM-dd");  
 try {  
 Date date=simpleDateFormat.parse(time);  
 String phone=req.getParameter("userphone");  
 String address=req.getParameter("userAddress");  
 Integer userlei=Integer.*parseInt*(req.getParameter("userlei"));  
 SmbmsUser smbmsUser=new SmbmsUser();  
 smbmsUser.setId(id);  
 smbmsUser.setUserName(username);  
 smbmsUser.setGender(gender);  
 smbmsUser.setBirthday(date);  
 smbmsUser.setPhone(phone);  
 smbmsUser.setAddress(address);  
 smbmsUser.setUserRole(userlei);  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
 int i = smbmsUserService.updateUser(smbmsUser);  
 if(i>0){  
 //转发  
 req.getRequestDispatcher("/userlist").forward(req,resp);  
 }else {  
 resp.sendRedirect("/userQuery");  
 }  
  
  
 } catch (ParseException e) {  
 e.printStackTrace();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
  
 }  
}**

**修改密码的代码如下：**

**package org.qf.servlet;  
  
import org.qf.pojo.SmbmsUser;  
import org.qf.service.SmbmsUserService;  
import org.qf.service.impl.SmbmsUserServiceImpl;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import java.io.IOException;  
  
public class UpdatePassword extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 this.doPost(req, resp);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
  
  
  
 try {  
 HttpSession session=req.getSession();  
 String userName =(String)session.getAttribute("sessionName");  
 System.*out*.println(userName);  
 String newPassword=req.getParameter("newPassword");  
 String oldPassword=req.getParameter("oldPassword");  
 System.*out*.println(newPassword);  
 System.*out*.println(oldPassword);  
 SmbmsUserService smbmsUserService=new SmbmsUserServiceImpl();  
  
 int i=smbmsUserService.updatePassworld(userName,newPassword,oldPassword);  
  
 if(i>0){  
 //转发  
 req.getRequestDispatcher("welcome.jsp").forward(req,resp);  
 }else {  
 resp.sendRedirect("password.jsp");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}**

1. 截图





1. **总结**

**通过本次的实验，我掌握了系统的三层架构以及UML建模的各个过程，也通过本次实训中遇到的错误掌握了解决他们的办法，当在MySQL插入中文时出现ERROR 1406 (22001): Data too long for column ‘name’ at row 1**

**使用命令行方式登陆到MySQL服务器， 建立一个数据库，数据库编码设为UTF-8。此时，如果直接在命令行窗口使用insert语句插入中文，就遇到类似 ERROR 1406 (22001): Data too long for column ‘name’ at row 1 错误。乍一看，是字段长度引起的问题，但是实际是字符编码的问题。可以尝试以下解决方法：**

**在Windows下，命令行窗口不支持UTF-8编码，所以使用“set names utf8;”不会达到转化中文的效果。但是这个问题还是可以解决的：**

**（1）使用默认编码建立数据库。这种情况下就可以直接输入中文了，但是相应的问题，就是会失去UTF-8编码的灵活性。特别是不利于软件的国际化。**

**（2）放弃命令行窗口登录MySQL，使用图形化客户端。客户端工具可以MySQL的官方网站上找到。**

**在实训的过程中也出现了404错误，404意味着该页面不存在，那么是什么导致404错误产生呢？**

**1、原始的URL失效，可能是管理员更改了URL。**

**2、文件目录或文件名称更改，导致原网页链接失效，找不到文件。**

**3、文件被移动到其他目录，移动文件也会导致原url链接失效。**

**4、输入错误的地址，调用不到文件，也会形成404。**

**404错误解决办法**

**按照上述错误原因，我们要做到，不要随便改变url，不要随意更改目录名称、文件名称，不要随意移动文件位置，要多检查错误，及时更改。**

**同时也出现了505错误如果启用了 HTTP 1.1，则可能出现此类错误。要禁用 HTTP 1.1，请单击“工具”按钮，单击“Internet 选项”，然后单击“高级”选项卡。在“设置”下，滚动到“HTTP 1.1 设置”部分，清除“使用 HTTP 1.1”复选框。如果第三方产品干扰 Internet Explorer，则可能出现此类错误。请尝试关闭所有程序，然后重新尝试访问该网页。大多数浏览器都默认Web服务器支持 1.x 的版本的 HTTP 协议。但实际上如 0.9 版今天已很少使用，关键是因为和新版本的协议比起来它们的安全性和性能都差得多。 所以，如果您在浏览器中看到这个错误，可以升级您的Web服务器软件。如果版本 1.x 的请求失败，很可能是因为您的 Web 服务器不是很好地支持 1.x 协议版本， 而不是根本不支持。**

**通过本次实训当中出现的错误，让我认识到我的不足，以及学习的方向。**