齐 齐 哈 尔 大 学

程序源代码

题 目： 基于JavaEE的齐大体育馆管理系统的实现

学 院： 计算机与控制工程学院

专业班级： 软件工程131

学生姓名： 许建华

指导教师： 王一萍

2017 年06 月18 日

关于管理员的EquipmentAction类：

package com.gym.admin.act311ion;

import java.io.IOException;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.gym.admin.service.impl.EquipmentServiceImpl;

import com.gym.model.EquipmentModel;

import com.gym.utils.CheckOnline;

import com.gym.utils.Constant;

public class EquipmentAction extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

String action = request.getParameter("action"); // 获取动作

// 检查管理员是否在线

if (!CheckOnline.isAdminOnline(request)) {

response.sendRedirect("../index.html");

return;

}

EquipmentServiceImpl equipmentServiceImpl = new EquipmentServiceImpl();

if (action == null) {

List equipmentList = equipmentServiceImpl.queryEquipment();

List equipmentTypeList = equipmentServiceImpl.queryAllType();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/equipment/index.jsp");

request.setAttribute("equipmentList", equipmentList);

request.setAttribute("equipmentTypeList", equipmentTypeList);

dispatcher.forward(request, response);

} else if (action.equals("add")) {

List equipmentTypeList = equipmentServiceImpl.queryAllType();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/equipment/add.jsp");

request.setAttribute("equipmentTypeList", equipmentTypeList);

dispatcher.forward(request, response);

} else if (action.equals("delete")) {

List equipmentList = equipmentServiceImpl.queryEquipment();

List equipmentTypeList = equipmentServiceImpl.queryAllType();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/equipment/delete.jsp");

request.setAttribute("equipmentList", equipmentList);

request.setAttribute("equipmentTypeList", equipmentTypeList);

dispatcher.forward(request, response);

} else if (action.equals("setting")) {

List equipmentList = equipmentServiceImpl.queryEquipment();

List equipmentTypeList = equipmentServiceImpl.queryAllType();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/equipment/setting.jsp");

request.setAttribute("equipmentList", equipmentList);

request.setAttribute("equipmentTypeList", equipmentTypeList);

dispatcher.forward(request, response);

} else if (action.equals("delbyid")) { // 转发

System.out.println(111);

doPost(request, response);

} else if (action.equals("punchcard")) { // 显示模拟打卡页面

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/equipment/punchcard.jsp");

dispatcher.forward(request, response);

} else if (action.equals("rent")) { // 转发

doPost(request, response);

} else if (action.equals("return")) { // 转发

doPost(request, response);

}

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

String action = request.getParameter("action"); // 获取动作

// 检查管理员是否在线

if (!CheckOnline.isAdminOnline(request)) {

response.sendRedirect("../index.html");

return;

}

EquipmentModel equipmentModel = new EquipmentModel();

EquipmentServiceImpl equipmentServiceImpl = new EquipmentServiceImpl();

if (action == null) {

} else if (action.equals("add")) { // 执行添加场地动作

equipmentModel.seteTypeId(request.getParameter("type"));

equipmentModel.setBorrowCount("0");

equipmentModel.setBuyCost(request.getParameter("cost"));

equipmentModel.setBuyDate(request.getParameter("date"));

equipmentModel.seteFee(request.getParameter("fee"));

equipmentModel.setStatus("1");

// 验证必填参数是否为空

if (equipmentModel.getBuyCost().equals("")

|| equipmentModel.getBuyDate().equals("")

|| equipmentModel.geteFee().equals("")) {

response.sendRedirect("../error.jsp?errorCode="

+ Constant.PARAMEMPTY);

return;

}

switch (equipmentServiceImpl.addEquipment(equipmentModel)) {

case Constant.SUCCESS:

response.sendRedirect("../success.jsp");

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

} else if (action.equals("setting")) { // 执行修改场地动作

equipmentModel.seteId(request.getParameter("eid"));

equipmentModel.seteFee(request.getParameter("fee"));

equipmentModel.seteTypeId(request.getParameter("type"));

switch (equipmentServiceImpl.alterEquipment(equipmentModel)) {

case Constant.SUCCESS:

response.sendRedirect("index.html?action=setting");

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

} else if (action.equals("delbyid")) { // 执行删除场地动作

switch (equipmentServiceImpl.delEquipment(request

.getParameter("eid"))) {

case Constant.SUCCESS:

response.sendRedirect("index.html?action=delete");

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

} else if (action.equals("punchcard")) { // 执行打卡动作

List rentList = equipmentServiceImpl.queryRentByUserId(request

.getParameter("uid"));// 读卡器传来的参数

// uid-一卡通卡号

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/equipment/punchcard.jsp");

request.setAttribute("rentList", rentList);

dispatcher.forward(request, response);

} else if (action.equals("rent")) {

switch (equipmentServiceImpl.rentGet(request.getParameter("uid"),

request.getParameter("eid"),

request.getParameter("erentid"))) {

case Constant.SUCCESS:

response.sendRedirect("../success.jsp?");

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

case Constant.PARAMEMPTY:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.PARAMEMPTY);

break;

case Constant.HASNOTBOOK:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.HASNOTBOOK);

break;

case Constant.NOTBOOKSTATUS:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.NOTBOOKSTATUS);

break;

default:

break;

}

} else if (action.equals("return")) { // 执行器材归还动作

switch (equipmentServiceImpl.rentRentrn(

request.getParameter("uid"), request.getParameter("eid"),

request.getParameter("erentid"))) {

case Constant.PARAMEMPTY:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.PARAMEMPTY);

break;

case Constant.SUCCESS:

response.sendRedirect("../success.jsp?");

break;

case Constant.NOTUSESTATUS:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.NOTUSESTATUS);

break;

case Constant.HASNOTRENT:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.HASNOTRENT);

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

}

}

}

关于管理员的GroundAction类：

package com.gym.admin.action;

import java.io.IOException;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.gym.admin.service.impl.GroundServiceImpl;

import com.gym.model.GroundModel;

import com.gym.utils.CheckOnline;

import com.gym.utils.Constant;

public class GroundAction extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

String action = request.getParameter("action"); // 获取动作

// 检查管理员是否在线

if (!CheckOnline.isAdminOnline(request)) {

response.sendRedirect("../index.html");

return;

}

GroundServiceImpl groundServiceImpl = new GroundServiceImpl();

if (action == null) { // 显示管理场地页面

List groundList = groundServiceImpl.queryGround();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/ground/index.jsp");

request.setAttribute("groundList", groundList);

dispatcher.forward(request, response);

} else if (action.equals("add")) { // 显示添加场地页面

List groundTypeList = groundServiceImpl.queryAllType();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/ground/add.jsp");

request.setAttribute("groundTypeList", groundTypeList);

dispatcher.forward(request, response);

} else if (action.equals("delete")) { // 显示删除场地页面

List groundList = groundServiceImpl.queryGround();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/ground/delete.jsp");

request.setAttribute("groundList", groundList);

dispatcher.forward(request, response);

} else if (action.equals("reserve")) { // 显示预留场地页面

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/ground/reserve.jsp");

dispatcher.forward(request, response);

} else if (action.equals("setting")) { // 显示修改场地页面

List groundList = groundServiceImpl.queryGround();

List groundTypeList = groundServiceImpl.queryAllType();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/ground/setting.jsp");

request.setAttribute("groundList", groundList);

request.setAttribute("groundTypeList", groundTypeList);

dispatcher.forward(request, response);

} else if (action.equals("delbyid")) { // 转发

doPost(request, response);

} else if (action.equals("punchcard")) { // 显示模拟打卡页面

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/ground/punchcard.jsp");

dispatcher.forward(request, response);

} else if (action.equals("use")) { // 转发

doPost(request, response);

}

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

String action = request.getParameter("action"); // 获取动作

// 检查管理员是否在线

if (!CheckOnline.isAdminOnline(request)) {

response.sendRedirect("../index.html");

return;

}

GroundServiceImpl groundServiceImpl = new GroundServiceImpl();

if (action == null) {

} else if (action.equals("add")) { // 执行添加场地动作

GroundModel groundModel = new GroundModel();

groundModel.setgFee(request.getParameter("fee"));

groundModel.setgName(request.getParameter("name"));

groundModel.setgRemark(request.getParameter("remark"));

groundModel.setgStatus("1");

groundModel.settId(request.getParameter("type"));

// 验证必填参数是否为空

if (groundModel.getgFee().equals("")

|| groundModel.getgName().equals("")

|| groundModel.gettId().equals("")) {

response.sendRedirect("../error.jsp?errorCode="

+ Constant.PARAMEMPTY);

return;

}

switch (groundServiceImpl.addGround(groundModel)) {

case Constant.SUCCESS:

response.sendRedirect("../success.jsp");

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

} else if (action.equals("reserve")) { // 执行保留场地动作

} else if (action.equals("setting")) { // 执行修改场地动作

GroundModel groundModel = new GroundModel();

groundModel.setgName(request.getParameter("name"));

groundModel.setgFee(request.getParameter("fee"));

groundModel.setgId(request.getParameter("gid"));

groundModel.settId(request.getParameter("type"));

groundModel.setgRemark(request.getParameter("remark"));

switch (groundServiceImpl.alterGround(groundModel)) {

case Constant.SUCCESS:

response.sendRedirect("index.html?action=setting");

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

} else if (action.equals("delbyid")) { // 执行删除场地动作

switch (groundServiceImpl.delGround(request.getParameter("gid"))) {

case Constant.SUCCESS:

response.sendRedirect("index.html?action=delete");

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

} else if (action.equals("punchcard")) { // 执行打卡动作

List bookList = groundServiceImpl.queryBookByUserId(request

.getParameter("uid")); // 根据读取的一卡通号查询该用户预约表

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/ground/punchcard.jsp");

request.setAttribute("bookList", bookList);

dispatcher.forward(request, response);

} else if (action.equals("use")) {

switch (groundServiceImpl.groundUse(request.getParameter("uid"),

request.getParameter("gid"), request.getParameter("bid"))) { // 场地使用

case Constant.SUCCESS:

response.sendRedirect("../success.jsp?");

break;

case Constant.PARAMEMPTY:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.PARAMEMPTY);

break;

case Constant.HASNOTBOOK:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.HASNOTBOOK);

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

}

}

}

关于管理员的NoticeAction类：

package com.gym.admin.action;

import java.io.IOException;

import java.text.DateFormat;

import java.util.Date;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.gym.admin.service.impl.NoticeServiceImpl;

import com.gym.model.NoticeModel;

import com.gym.utils.CheckOnline;

import com.gym.utils.Constant;

public class NoticeAction extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

String action = request.getParameter("action"); // 获取动作

// 检查管理员是否在线

if (!CheckOnline.isAdminOnline(request)) {

response.sendRedirect("../index.html");

return;

}

RequestDispatcher dispatcher = request

.getRequestDispatcher("/admin/notice/index.jsp");

dispatcher.forward(request, response);

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

String action = request.getParameter("action"); // 获取动作

// 检查管理员是否在线

if (!CheckOnline.isAdminOnline(request)) {

response.sendRedirect("../index.html");

return;

}

NoticeModel noticeModel = new NoticeModel();

NoticeServiceImpl noticeServiceImpl = new NoticeServiceImpl();

if (action == null) {

} else if (action.equals("add")) { // 执行添加公告动作

System.out.println(11);

noticeModel.setnDate(DateFormat.getDateTimeInstance().format(

new Date()));

noticeModel.setnText(request.getParameter("content"));

noticeModel.setnTitle(request.getParameter("title"));

// 调试

noticeModel.setaId("1000");

// 验证必填参数是否为空

if (noticeModel.getnText().equals("")

|| noticeModel.getnTitle().equals("")) {

response.sendRedirect("../error.jsp?errorCode="

+ Constant.PARAMEMPTY);

return;

}

switch (noticeServiceImpl.addNotice(noticeModel)) {

case Constant.SUCCESS:

response.sendRedirect("../success.jsp");

break;

case Constant.ERROR:

response.sendRedirect("../error.jsp?errorCode="

+ Constant.ERROR);

break;

default:

break;

}

}

}

}

Gym.dao.impl.GroundDaoImpl类：

package com.gym.dao.impl;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import com.gym.dao.GroundDao;

import com.gym.model.GroundModel;

import com.gym.model.GroundTypeModel;

import com.gym.utils.DbConnection;

import com.mysql.jdbc.PreparedStatement;

import com.mysql.jdbc.ResultSet;

import com.mysql.jdbc.ResultSetMetaData;

public class GroundDaoImpl implements GroundDao {

public int addGround(GroundModel groundModel) {

// TODO Auto-generated method stub

int affect = 0;

String sql = "INSERT INTO ground ( gName, tId, gFee, gStatus, gRemark) VALUES (?,?,?,?,?)";

try {

PreparedStatement statement = (PreparedStatement) DbConnection

.getConn().prepareStatement(sql);

statement.setString(1, groundModel.getgName());

statement.setString(2, groundModel.gettId());

statement.setString(3, groundModel.getgFee());

statement.setString(4, groundModel.getgStatus());

statement.setString(5, groundModel.getgRemark());

affect = statement.executeUpdate();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return affect;

}

public int delGround(String gId) {

// TODO Auto-generated method stub

Statement statement = DbConnection.getStatement();

int affect = 0;

String sql = "UPDATE ground SET gStatus = 0 WHERE gId = " + gId;

// System.out.println(sql);

try {

affect = statement.executeUpdate(sql);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return affect;

}

public int alterGround(GroundModel groundModel) {

// TODO Auto-generated method stub

Statement statement = DbConnection.getStatement();

int affect = 0;

String sql = "UPDATE ground SET gName = '" + groundModel.getgName()

+ "' ,tId = '" + groundModel.gettId() + "' ,gFee = '"

+ groundModel.getgFee() + "',gStatus = '1' ,gRemark = '"

+ groundModel.getgRemark() + "' WHERE gId = "

+ groundModel.getgId();

// System.out.println(sql);

try {

affect = statement.executeUpdate(sql);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return affect;

}

public List queryGround() {

// TODO Auto-generated method stub

Statement statement = DbConnection.getStatement();

ResultSet resultSet = null;

String sql = "SELECT \* FROM ground g,groundtype t WHERE g.tId=t.tId and g.gStatus = 1 ORDER BY gId ASC ";

// System.out.println(sql);

try {

resultSet = (ResultSet) statement.executeQuery(sql);

ResultSetMetaData resultSetMetaData = (ResultSetMetaData) resultSet

.getMetaData(); // 得到结果集(rs)的结构信息，比如字段数、字段名等

List list = new ArrayList();

GroundModel groundModel;

while (resultSet.next()) {

groundModel = new GroundModel();

groundModel.setgFee(resultSet.getString("gFee"));

groundModel.setgId(resultSet.getString("gId"));

groundModel.setgName(resultSet.getString("gName"));

groundModel.setgRemark(resultSet.getString("gRemark"));

groundModel.setgStatus(resultSet.getString("gStatus"));

groundModel.settId(resultSet.getString("tId"));

groundModel.settName(resultSet.getString("tName"));

list.add(groundModel);

}

return list;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

public List queryAllType() {

// TODO Auto-generated method stub

Statement statement = DbConnection.getStatement();

ResultSet resultSet = null;

String sql = "SELECT \* FROM `groundtype`";

System.out.println(sql);

try {

resultSet = (ResultSet) statement.executeQuery(sql);

ResultSetMetaData resultSetMetaData = (ResultSetMetaData) resultSet

.getMetaData(); // 得到结果集(rs)的结构信息，比如字段数、字段名等

List list = new ArrayList();

GroundTypeModel groundTypeModel;

while (resultSet.next()) {

groundTypeModel = new GroundTypeModel();

groundTypeModel.settId(resultSet.getString("tId"));

groundTypeModel.settName(resultSet.getString("tName"));

list.add(groundTypeModel);

}

return list;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

public GroundModel queryById(String gId) {

// TODO Auto-generated method stub

Statement statement = DbConnection.getStatement();

ResultSet resultSet = null;

String sql = "SELECT \* FROM ground g,groundtype t WHERE g.tId=t.tId and g.gStatus = 1 AND gId = "

+ gId;

System.out.println(sql);

try {

resultSet = (ResultSet) statement.executeQuery(sql);

ResultSetMetaData resultSetMetaData = (ResultSetMetaData) resultSet

.getMetaData(); // 得到结果集(rs)的结构信息，比如字段数、字段名等

GroundModel groundModel = new GroundModel();

while (resultSet.next()) {

groundModel.setgFee(resultSet.getString("gFee"));

groundModel.setgId(resultSet.getString("gId"));

groundModel.setgName(resultSet.getString("gName"));

groundModel.setgRemark(resultSet.getString("gRemark"));

groundModel.setgStatus(resultSet.getString("gStatus"));

groundModel.settId(resultSet.getString("tId"));

groundModel.settName(resultSet.getString("tName"));

}

return groundModel;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

public GroundModel queryByName(String gName) {

// TODO Auto-generated method stub

return null;

}

}

Gym.user.action.LoginAction类:

package com.gym.user.action;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.gym.model.UserModel;

import com.gym.user.service.impl.LoginServiceImpl;

import com.gym.utils.Constant;

public class LoginAction extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

if (null == action) {

RequestDispatcher dispatcher = request

.getRequestDispatcher("login.jsp");

dispatcher.forward(request, response);

} else if (action.equals("logout")) {

request.getSession().removeAttribute("uName");// 清除session

request.getSession().removeAttribute("uId");// 清除session

RequestDispatcher dispatcher = request

.getRequestDispatcher("login.jsp");

dispatcher.forward(request, response);

}

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

UserModel userModel = new UserModel();

userModel.setuId(request.getParameter("uid"));

userModel.setuPassword(request.getParameter("password"));

// 验证必填参数是否为空

if (userModel.getuId().equals("")

|| userModel.getuPassword().equals("")) {

response.sendRedirect("error.jsp?errorCode=" + Constant.PARAMEMPTY);

return;

}

LoginServiceImpl loginServiceImpl = new LoginServiceImpl();

switch (loginServiceImpl.login(userModel)) {

case Constant.SUCCESS:

System.out.println("用户登陆成功");

request.getSession().setAttribute("uName", userModel.getuName()); // 设置session

request.getSession().setAttribute("uId", userModel.getuId()); // 设置session

response.sendRedirect("index.html");

break;

case Constant.USERNOTEXIST:

response.sendRedirect("error.jsp?errorCode="

+ Constant.USERNOTEXIST);

break;

case Constant.USERPWDERROR:

response.sendRedirect("error.jsp?errorCode="

+ Constant.USERPWDERROR);

break;

}

}

}

com.gym.user.action. RegistAction类:

package com.gym.user.action;

import java.io.IOException;

import java.text.DateFormat;

import java.util.Date;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.gym.model.UserModel;

import com.gym.user.service.impl.RegistServiceImpl;

import com.gym.utils.Constant;

public class RegistAction extends HttpServlet {

/\*\*

\* The doGet method of the servlet. <br>

\*

\* This method is called when a form has its tag value method equals to get.

\*

\* @param request

\* the request send by the client to the server

\* @param response

\* the response send by the server to the client

\* @throws ServletException

\* if an error occurred

\* @throws IOException

\* if an error occurred

\*/

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request

.getRequestDispatcher("regist.jsp");

dispatcher.forward(request, response);

}

/\*\*

\* The doPost method of the servlet. <br>

\*

\* This method is called when a form has its tag value method equals to

\* post.

\*

\* @param request

\* the request send by the client to the server

\* @param response

\* the response send by the server to the client

\* @throws ServletException

\* if an error occurred

\* @throws IOException

\* if an error occurred

\*/

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

UserModel userModel = new UserModel();

userModel.setStatus("1");

userModel.setuEmail(request.getParameter("email"));

userModel.setuId(request.getParameter("uid"));

userModel.setuIdCard(request.getParameter("idcard"));

userModel.setuName(request.getParameter("name"));

userModel.setuPassword(request.getParameter("password"));

userModel.setuPhone(request.getParameter("phone"));

userModel.setDate(DateFormat.getDateTimeInstance().format(new Date()));

// 验证必填参数是否为空

if (userModel.getuEmail().equals("") || userModel.getuId().equals("")

|| userModel.getuIdCard().equals("")

|| userModel.getuName().equals("")

|| request.getParameter("password").equals("")

|| request.getParameter("password2").equals("")) {

RequestDispatcher dispatcher = request

.getRequestDispatcher("error.jsp");

request.setAttribute("errorCode", "所有必填项不可为空");

dispatcher.forward(request, response);

return;

}

// 验证两次密码是否一致

if (!request.getParameter("password").equals(

request.getParameter("password2"))) {

RequestDispatcher dispatcher = request

.getRequestDispatcher("error.jsp");

request.setAttribute("errorCode", "两次密码不一致");

dispatcher.forward(request, response);

return;

}

RegistServiceImpl registServiceImpl = new RegistServiceImpl();

RequestDispatcher dispatcher;

switch (registServiceImpl.regist(userModel)) {

case Constant.SUCCESS:

response.sendRedirect("success.jsp");

break;

case Constant.ERROR:

response.sendRedirect("error.jsp?errorCode=" + Constant.ERROR);

break;

case Constant.USEREXIST:

response.sendRedirect("error.jsp?errorCode=" + Constant.USEREXIST);

break;

default:

break;

}

}

}

com.gym.utils.DbConnection类:

package com.gym.utils;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

/\*\*

\* 数据库连接类

\*

\* @author Feng

\*

\*/

public class DbConnection {

static Connection conn = null;

static Statement statement = null;

static private String dbUser = "root";

static private String dbPassword = "root";

/\*\*

\* 获取connection连接

\*

\* @return

\*/

public static Connection getConn() {

try {

if (conn != null && !conn.isClosed()) {

return conn;

}

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

// String url = "jdbc:mysql://localhost:3306/gymdb?user=" + dbUser;

String url = "jdbc:mysql://localhost:3306/gymdb?user=" + dbUser

+ "&password=" + dbPassword + "&characterEncoding=utf-8";

try {

Class.forName("org.gjt.mm.mysql.Driver").newInstance();

} catch (InstantiationException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IllegalAccessException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

conn = DriverManager.getConnection(url, dbUser, dbPassword);

System.out.println("数据库连接成功");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return conn;

}

/\*\*

\* 获取 Statement

\*

\* @return

\* @throws SQLException

\*/

public static Statement getStatement() {

try {

// 判断conn是否为空或已关闭

if (conn == null || conn.isClosed()) { // 重新连接数据库

// String url = "jdbc:mysql://localhost/gymdb?user=" + dbUser;

String url = "jdbc:mysql://localhost/gymdb?user=" + dbUser

+ "&password=" + dbPassword

+ "&characterEncoding=utf-8";

try {

Class.forName("org.gjt.mm.mysql.Driver").newInstance();

} catch (InstantiationException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IllegalAccessException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (ClassNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

try {

conn = DriverManager.getConnection(url, dbUser, dbPassword);

System.out.println("数据库连接成功");

statement = (Statement) conn.createStatement();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return statement;

} else {// 使用旧连接

statement = (Statement) conn.createStatement();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return statement;

}

}

用户登录jsp代码：

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

contentType=*"text/html; charset=utf-8"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme() + "://"

+ request.getServerName() + ":" + request.getServerPort()

+ path + "/";

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=utf-8"* />

<title>登录 -齐齐哈尔大学学体育馆网上系统</title>

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

<script src=*"js/nav.js"* type=*"text/javascript"*></script>

<script src=*"js/focus.js"* type=*"text/javascript"*></script>

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

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

</head>

<body>

<%@ include file=*"include/header.jsp"*%>

<div id=*"main"*>

<div id=*"row\_1"*>

<div id=*"login\_div"*>

<form action=*"login.html"* method=*"post"*>

<span class=*"tmp"*>用户名：</span> <input name=*"uid"* type=*"text"*

class=*"input\_text"* /> <br /> 密&nbsp;&nbsp;&nbsp;码： <input

name=*"password"* type=*"password"* class=*"input\_text"* /> <br /> <br />

<br /> <input type=*"submit"* value=*"登录"* class=*"btn"* /> <input

type=*"reset"* value=*"重置"* class=*"btn"* />

</form>

</div>

</div>

</div>

<script>

**var** def = "1";

</script>

<%@ include file=*"include/footer.jsp"*%>

</body>

</html>

用户注册jsp代码:

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

contentType=*"text/html; charset=utf-8"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme() + "://"

+ request.getServerName() + ":" + request.getServerPort()

+ path + "/";

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=utf-8"* />

<title>注册 -齐齐哈尔大学学体育馆网上系统</title>

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

<script src=*"js/nav.js"* type=*"text/javascript"*></script>

<script src=*"js/focus.js"* type=*"text/javascript"*></script>

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

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

</head>

<body>

<%@ include file=*"include/header.jsp"*%>

<div id=*"main"*>

<div id=*"row\_1"*>

<div id=*"reg\_banner"*>

<img src=*"images/login\_banner.jpg"* />

</div>

<div id=*"reg\_div"*>

<form action=*"regist.html"* method=*"post"*>

校园卡号： <input name=*"uid"* type=*"text"* class=*"input\_text"* /> \* <br />

姓&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;名： <input name=*"name"*

type=*"text"* class=*"input\_text"* /> \* <br />

密&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;码： <input name=*"password"*

type=*"password"* class=*"input\_text"* /> \* <br /> 确认密码： <input

name=*"password2"* type=*"password"* class=*"input\_text"* /> \* <br />

邮&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;箱： <input name=*"email"*

type=*"text"* class=*"input\_text"* /> \* <br /> 手机号码： <input

name=*"phone"* type=*"text"* class=*"input\_text"* /> \* <br /> 身份证号： <input

name=*"idcard"* type=*"text"* class=*"input\_text"* /> \* <br /> <br />

<br /> <input type=*"submit"* value=*"注册"* class=*"btn"* /> <input

type=*"reset"* value=*"重置"* class=*"btn"* />

</form>

</div>

</div>

</div>

<script>

**var** def = "1";

</script>

<%@ include file=*"include/footer.jsp"*%>

</body>

</html>

web.xml代码：

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

<web-app version=*"2.5"* xmlns=*"http://java.sun.com/xml/ns/javaee"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*

*http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*>

<display-name></display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

</welcome-file-list>

<filter>

<filter-name>jspFilter</filter-name>

<filter-class>com.gym.utils.jspFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>jspFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<servlet>

<servlet-name>IndexAction</servlet-name>

<servlet-class>com.gym.user.action.IndexAction</servlet-class>

</servlet>

<servlet>

<servlet-name>LoginAction</servlet-name>

<servlet-class>com.gym.user.action.LoginAction</servlet-class>

</servlet>

<servlet>

<servlet-name>RegistAction</servlet-name>

<servlet-class>com.gym.user.action.RegistAction</servlet-class>

</servlet>

<servlet>

<servlet-name>MatchAction</servlet-name>

<servlet-class>com.gym.user.action.MatchAction</servlet-class>

</servlet>

<servlet>

<servlet-name>GroundAction</servlet-name>

<servlet-class>com.gym.user.action.GroundAction</servlet-class>

</servlet>

<servlet>

<servlet-name>EquipmentAction</servlet-name>

<servlet-class>com.gym.user.action.EquipmentAction</servlet-class>

</servlet>

<servlet>

<servlet-name>UserAction</servlet-name>

<servlet-class>com.gym.user.action.UserAction</servlet-class>

</servlet>

<servlet>

<servlet-name>NoticeAction</servlet-name>

<servlet-class>com.gym.user.action.NoticeAction</servlet-class>

</servlet>

<servlet>

<servlet-name>HelpAction</servlet-name>

<servlet-class>com.gym.user.action.HelpAction</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminIndexAction</servlet-name>

<servlet-class>com.gym.admin.action.IndexAction</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminPersonnelAction</servlet-name>

<servlet-class>com.gym.admin.action.PersonnelAction</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminMatchAction</servlet-name>

<servlet-class>com.gym.admin.action.MatchAction</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminGroundAction</servlet-name>

<servlet-class>com.gym.admin.action.GroundAction</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminEquipmentAction</servlet-name>

<servlet-class>com.gym.admin.action.EquipmentAction</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminNoticeAction</servlet-name>

<servlet-class>com.gym.admin.action.NoticeAction</servlet-class>

</servlet>

<servlet>

<servlet-name>AdminReportAction</servlet-name>

<servlet-class>com.gym.admin.action.ReportAction</servlet-class>

</servlet>

<servlet>

<servlet-name>PunchCardAction</servlet-name>

<servlet-class>com.gym.admin.action.PunchCardAction</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>IndexAction</servlet-name>

<url-pattern>/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>LoginAction</servlet-name>

<url-pattern>/login.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>RegistAction</servlet-name>

<url-pattern>/regist.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>MatchAction</servlet-name>

<url-pattern>/match/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>GroundAction</servlet-name>

<url-pattern>/ground/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>EquipmentAction</servlet-name>

<url-pattern>/equipment/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>UserAction</servlet-name>

<url-pattern>/user/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>NoticeAction</servlet-name>

<url-pattern>/notice/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>HelpAction</servlet-name>

<url-pattern>/help/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminIndexAction</servlet-name>

<url-pattern>/admin/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminPersonnelAction</servlet-name>

<url-pattern>/admin/personnel/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminMatchAction</servlet-name>

<url-pattern>/admin/match/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminGroundAction</servlet-name>

<url-pattern>/admin/ground/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminEquipmentAction</servlet-name>

<url-pattern>/admin/equipment/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminNoticeAction</servlet-name>

<url-pattern>/admin/notice/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AdminReportAction</servlet-name>

<url-pattern>/admin/report/index.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>PunchCardAction</servlet-name>

<url-pattern>/admin/punchcard/index.html</url-pattern>

</servlet-mapping>

</web-app>