Java高级编程

**课程设计报告**

题　　目 酒店人力资源设计与实现

组　　长

组　　员

　2020年7月

**课程设计分工明细**

设计分工明细情况：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 学号 | 姓名 | 工作内容 | 备注 |
| 1 |  |  | 查询功能的实现；系统设计（css样式的设计与实践） | 组长 |
| 2 |  |  | 修改信息功能的实现；数据库的设计； |  |
| 3 |  |  | 删除员工信息功能的实现；系统分析；系统css设计 | 原计划和组长一人一半css实践与设计；但是中途数据库出了问题（不能根据系统数据进行实时调整），所以最后只参与了设计页面 |
| 4 |  |  | 添加员工信息功能的实现；系统设计 | 抄写报告 |

**第1章　系统分析**

**1.1 需求分析**

通过调查，根据人事管理的基本需求，要求系统需要完成以下功能；详细的企业的员工信息管理；企业员工奖惩信息管理；企业员工人员调动信息管理；企业员工考勤管理；设置企业员工薪酬管理；人才招聘管理及企业人才库管理功能；企业员工培训管理；提供系统用户管理功能。

本系统实现了企业员工基础信息管理和员工工资管理功能。

**1.2  可行性分析**

1.2.1 经济性

通过网络化的员工信息管理，大大的提高了企业人才的利用率，使之为企业创造了更大价值。人才利用率的提高，增强了企业的核心竟争力，全面提升了企业的管理能力，从而企业适应了信息时代的网络化管理要求。

1.2.2 技术性

通过网络化管理，加强和员工信息管理的方便性，准备确性。使用JSP技术,Web页面开发人员可以使用HTML或者XML标识来设计和格式化最终页面。使用JSP标识或者小脚本来生成页面上的动态内容.生成的内容的逻辑被封装在标识和JavaBeans组件中，那么其他人，如Web管理人员和页面设计者,能够编辑和使用JSP页面而不影响内容的生成。很多的OS中都带有JDBC驱动程序，Java就通过JDBC与数据库连接，执行查询，提取数据等操作.Sun公司还提供了JDBC-ODBC的桥连方式，因此Java也可以访问带有ODBC驱动的数据库,目前大多数数据库系统都带有ODBC驱动，所以Java能访问诸如Oracle，Sybase，MS SQL Server 和MySQL等类型的数据库。

1.2.3 操作可行性

对于用户来说，本系统只要求使用者掌握一定的上网经验，经过仔细设计和测试之后的系统具有操作简单，方便灵活等优点，足可以满足各种用户的不同需求，同时也方便了企业的内部管理。管理人员及用户一定会在短时间内掌握并熟练使用。只要动一动鼠标键盘就可以达到想要的结果。

1.2.4 系统结构可行性分析

本系统采用B/S模式。

B/S（Browser/Server）结构即浏览器和服务器结构。它是随着Internet技术的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过WWW浏览器来实现，极少部分事务逻辑在前端（Browser）实现，但是主要事务逻辑在服务器端（Server）实现，形成所谓三层结构。这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本（TCO）。以目前的技术看，局域网建立B/S结构的网络应用，并通过Internet/Intranet模式下数据库应用，相对易于把握、成本也是较低的。它是一次性到位的开发，能实现不同的人员，从不同的地点，以不同的接入方式（比如LAN,WAN,Internet/Intranet等）访问和操作共同的数据库；它能有效地保护数据平台和管理访问权限，服务器数据库也很安全。特别是在JAVA这样的跨平台语言出现之后，B/S架构管理软件更是方便、快捷、高效。其具有以下特点：维护和升级方式简单，成本降低,选择更多，只要安装了浏览器就可以了。

**1.3 总体设计**

1.3.1项目规划

人力资源是企业所有资源中最活跃与最昂贵的资源,如何让更多的员工以积极主动的态度参与公司的运作，最大限度的发挥企业人力资源的效用是每一个管理者关心的问题。本系统完全按照"以人为本"的设计要求,以综合利用信息资源，调动管理人员积极性、

综合智慧和创造性为目的,将人事管理所涉及的各类业务按照其内在的联系组织成一个有机整体。清沐酒店人力资源管理管理系统由登录模块，信息查询、信息添加，信息更改和删除4个功能组成，规划系统功能模块如下：

管理员登录模块：此模块实现用户进入系统管理前的验证操作，只有输入正确的用户名和密码才能进入系统，进行各种后台操作。

人员信息查询模块：此模块可以根据人员的编号进行在职人员的信息展示。

工资信息查询模块：此模块可以根据人员的编号进行在职人员工资信息的展示。

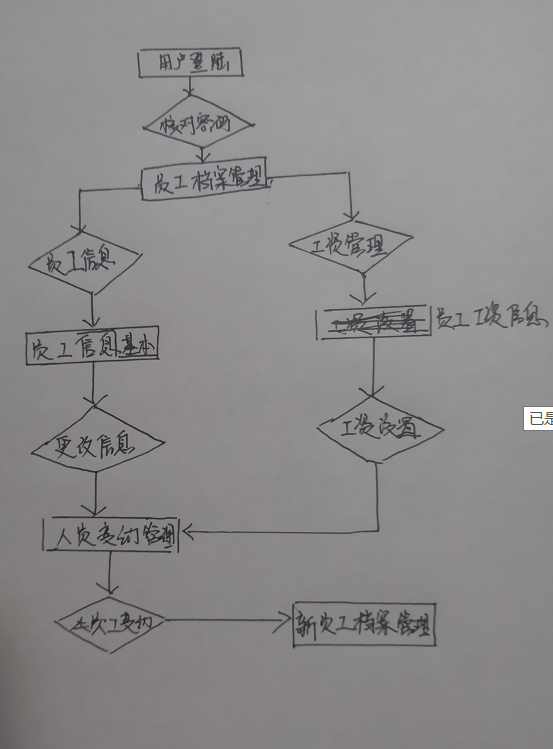
人员信息更改模块：本模块提供了管理企业人员信息更改功能，可以根据需要，对某条人员记录进行修改。

工资信息更改模块：本模块提供了企业人员工资信息更改功能，可以根据需要来对某条人员记录进行修改。

人员信息添加模块：本模块提供了管理企业人员信息添加功能，可以根据需要来添加一条新的人员记录。

人员信息删除模块：本模块提供了企业人员信息删除功能，可以根据需要来对某人员记录进行删除。

操作过程如下：



1.3.2 设计目标

本系统主要用于对员工进行管理。通过员工信息的各种信息的管理来提高的竞争力。为具体实现目标如下：实现对员工信息进行全面管理的功能。企业员工人员调动信息管理；系统最大限度地实现了易安装性、易维护性和易操作性。系统运行稳定、安全可靠。透过信息化方式取代传统人工作业，帮助人事单位化繁为简,快速有效的处理奖惩考核、训练发展、任用迁调等各项人事作业，大幅提升人事单位的工作效率,使其作业控管事半功倍,使企业达到“事得其人，人尽其才”的目标，创造企业竞争力，让管理者轻松的体味人事管理系统的新观念。

1.4业务流程分析

该系统只是管理员对员工信息的操作，首先，管理人员登陆进入系统，对员工

信息进行增删改查，进而对工资等信息进行增删改查，然后实现对部门员工部分信息的统计及修改，操作员亦可对自己的信息进行操作，管理员对系统的操作都将被记录。

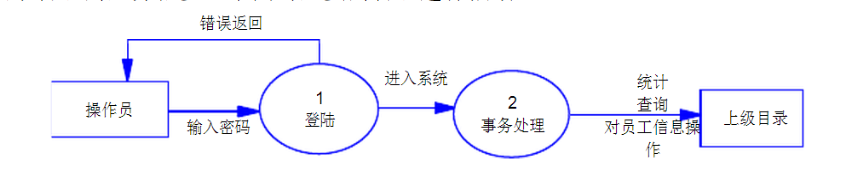


图1.系统业务流程图

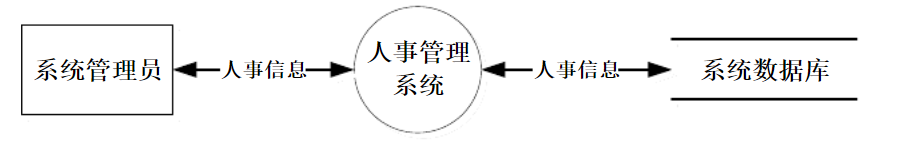


图2.第一层数据流图

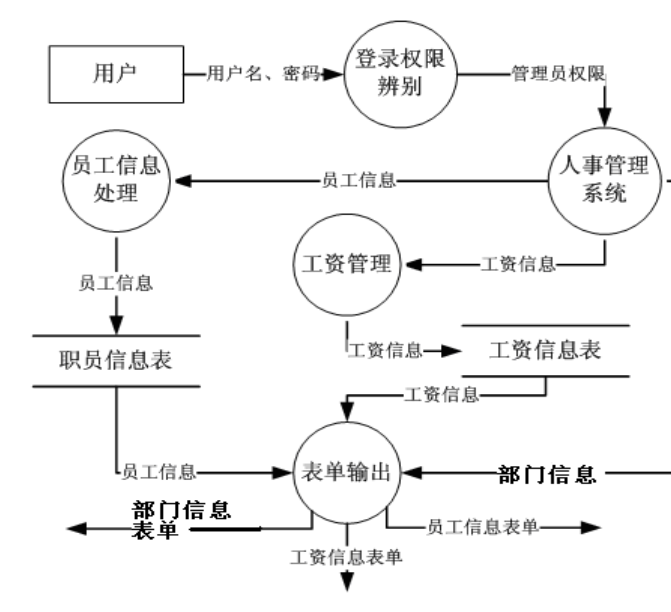


图3.底层数据流图

第2章 系统设计

2.1功能模块设计

酒店人力资源管理系统平台被定位为一个信息管理平台,不直接参与工作过程。管理员在平台记录要人员信息，可以进行职位更改、离职、入职等多种工作状态的记录。平台主要功能模块如图:

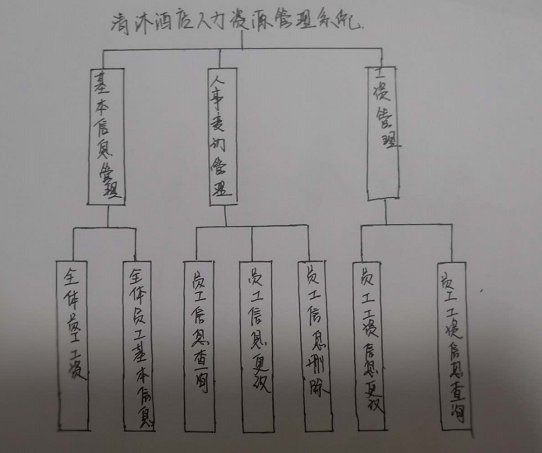


图3-1 功能模块图

2.2 输入输出设计

I/O设计的内容很广泛，主要包括I/O方式的设计、时间的设计、地点的设计、信息介质化的设计及最终界面的设计。

在I/O方式设计方面，常见的输入方式有：键盘输入，模/数、数/模输入，网络传送，磁盘/光盘读入等几种形式。在输入方面，对于键盘、鼠标的输入，可用VISUAL C++的标准输入/输出，对输入进行处理。

2.2.1 输入设计



2.2.2 输出设计



2.3 数据库设计

2.3.1 概念结构设计

概念性数据模型关心的是如何完整、正确地反映客观实际情况，不关心在数据库  
中如何实现。这种数据模式能真实地反应用户要求的实际情况，是一种容易被人  
们理解的直观的数据库结构模式。同时也是--种相对稳定统一的数据模式，--般  
情况下很少变动。概念性数据在用户和设计者之间建立了桥梁。是设计数据库结  
构的基础。  
 设计中自顶向下的实体分析方法，即常用的实体联系模型(简称E-R模型)，  
对具体数据进行抽象加工，将实体集合抽象成实体类型。用实体间联系反映现实  
世界事物间的内在联系。E-R模型是建立概念性数据模型的有力工具。有了具体信息管理需求以后，就可以设计出能够满足用户需求的各种实体,以及它们之间的关系，为后面的逻辑结构设计打下基础。这些实体包含各种具体信息，通过相互之间的作用形成数据的操作及其变化。

画出系统的E-R图，并对属性做介绍。

2.3.2 逻辑结构设计

数据库的逻辑结构设计的结果不是唯一的。为了提高数据库应用系统的性能，还应该根据应用需要适当的修改，调整关系模式，这就是数据模型的优化。规范设计人员判断关系模式的优劣提供了理论标准，可用来预测模式出现的问题，使数据库设计工作有了严格的理论基础。关系数据模型的优化通常以规范化理论为指导，本系统中的表如下：

表2-1 employee

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 名 | 类型 | 长度 | 小数点 | 不是null | 虚拟 | 键 | 注释 |
| 1 | id | int | 50 | 0 | √ |  | 1 |  |
| 2 | name | varchar | 255 | 0 |  |  |  |  |
| 3 | sex | varchar | 255 | 0 |  |  |  |  |
| 4 | nation | varchar | 255 | 0 |  |  |  |  |
| 5 | birth | varchar | 255 | 0 |  |  |  |  |
| 6 | phone | varchar | 255 | 0 |  |  |  |  |
| 7 | idcard | varchar | 255 | 0 |  |  |  |  |
| 8 | depart | varchar | 255 | 0 |  |  |  |  |
| 9 | profe | varchar | 255 | 0 |  |  |  |  |
| 10 | email | varchar | 255 | 0 |  |  |  |  |
| 11 | school | varchar | 255 | 0 |  |  |  |  |
| 12 | educat | varchar | 255 | 0 |  |  |  |  |
| 13 | entry | varchar | 255 | 0 |  |  |  |  |

表2-2 工资表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 名 | 类型 | 长度 | 小数点 | | 不是null | 虚拟 | 键 | | 注释 |
|  | id | int | 50 | 0 | √ | |  | | 1 | 主键 |
|  | Leave | varchar | 255 | 0 |  | |  | |  | 请假天数 |
|  | belate | varchar | 255 | 0 |  | |  | |  | 迟到次数 |
|  | absent | varchar | 255 | 0 |  | |  | |  | 旷工天数 |
|  | basepay | varchar | 255 | 0 |  | |  | |  | 基本工资 |
|  | bonus | varchar | 255 | 0 |  | |  | |  | 奖金 |
|  | depay | varchar | 255 | 0 |  | |  | |  | 应扣工资 |
|  | rpay | varchar | 255 | 0 |  | |  | |  | 实际工资 |

第3章　系统实施

3.1 系统平台

本系统使用MYSQL 数据库、Tomcat为服务器。Tomcat是开源软件，可以从网上免费得到其正式的发行版本，而MYSQL数据库是当前使用得最广泛的数据库，其性能，安全性等方面都有很好的保证。Tomcat虽然是开源软件，但许多大型网站也因其稳定的性能和出色的性能而选择它作为专业的JSP容器。由此组成的整个系统的底层框架，既能节省开支，配置过程也相对简单。

使用面向对象技术，将各个角色封装成类，这样能帮助程序员更直观地去理解整个业务流程，将重点放在业务逻辑的处理上，降低了开发难度，这样也便于将来的扩展。

3.1.1 硬件平台

由于Java具有跨平台的特点，所以只要能够安装JDK，就能在此安装JSP服务器和。JSP服务器和能够运行在目前绝大多数的操作系统上，目前在普通用户中用到的系统绝大多数为Windows系列。具体到本系统的硬件环境为：

硬件平台：

CPU：AMD双核；

内存：512MB

3.1.2 软件环境

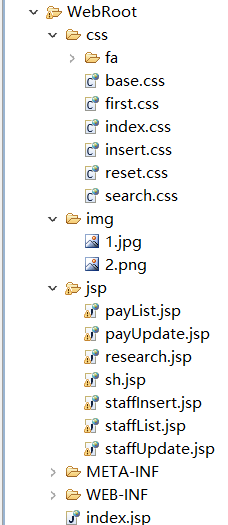
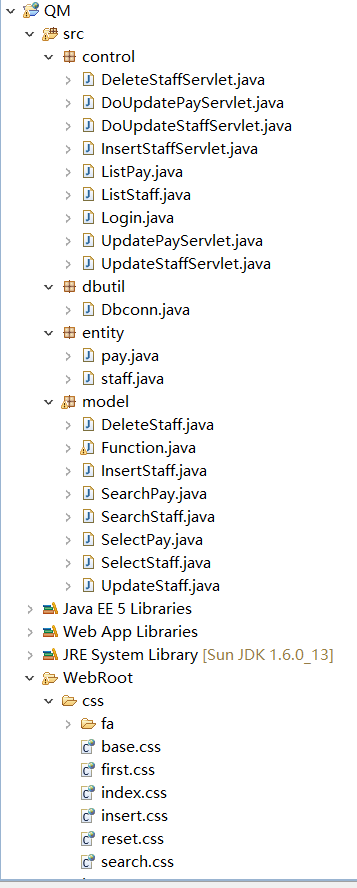
操作系统：windows10

数据库服务器：mysql

Web服务器：IIS

浏览器：谷歌

3.2 部分代码说明



model包：包括了插入，删除，修改，增加功能的代码、

package model：

public List<staff> getAllStaff() {

List<staff>listStaff = new ArrayList<staff>();

try {

Connection connection=Dbconn.getConnection();

String sql="select \* from employee";

PreparedStatement statement=connection.prepareStatement(sql);

ResultSet rs= statement.executeQuery();

while(rs.next()){

staff staffs=new staff();staffs.setId(rs.getInt("id"));

staffs.setName(rs.getString("name"));staffs.setSex(rs.getString("sex"));

staffs.setNation(rs.getString("nation"));staffs.setPhone(rs.getString("phone"));staffs.setIdcard(rs.getString("idcard"));staffs.setDepart(rs.getString("depart"));staffs.setProfe(rs.getString("profe"));staffs.setEmail(rs.getString("email"));staffs.setEntry(rs.getString("entry"));

staffs.setSchool(rs.getString("school"));staffs.setEducat(rs.getString("educat"));staffs.setBirth(rs.getString("birth"));listStaff.add(staffs)； }

public static staff selectById(int id){

staff s=null;

try {

Connection connection=Dbconn.getConnection();

String sql="select \* from employee where id=?";

PreparedStatement statement=connection.prepareStatement(sql);

statement.setInt(1, id);

ResultSet rs=statement.executeQuery();

if(rs.next()){

s=new staff();s.setId(rs.getInt("id"));s.setName(rs.getString("name"));

s.setSex(rs.getString("sex"));s.setNation(rs.getString("nation"));

s.setBirth(rs.getString("birth"));s.setPhone(rs.getString("phone"));

s.setIdcard(rs.getString("idcard"));s.setDepart(rs.getString("depart"));

s.setProfe(rs.getString("profe"));s.setEmail(rs.getString("email"));

s.setSchool(rs.getString("school"));s.setEducat(rs.getString("educat"));

s.setEntry(rs.getString("entry"));}

public boolean Insert(staff s){

try{

Connection connection=Dbconn.getConnection();

String sql="insert into employee (id,name,sex,nation,birth,phone,idcard,depart,profe,email,school,educat,entry) values (?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement statement=connection.prepareStatement(sql);

statement.setInt(1, s.getId());

statement.setString(2, s.getName());

。。。。。。。。。。。。。。。。。。。。。。。。。。。。

statement.setString(13, s.getEntry());

int result= statement.executeUpdate();

public boolean Delete(int id) {

// TODO Auto-generated method stub

boolean result=false;

try {

Connection connection=Dbconn.getConnection();

String sql="delete from employee where id=?";

PreparedStatement statement=connection.prepareStatement(sql);

statement.setInt(1, id);

result= statement.execute();

public static boolean Update(staff s){

boolean result=false;

try {

Connection connection=Dbconn.getConnection();

String sql="update employee set name=?,sex=?,nation=?,birth=?,phone=?,idcard=?,depart=?,profe=?,email=?,school=?,educat=?,entry=? where id=?";

PreparedStatement statement=connection.prepareStatement(sql);

statement.setString(1, s.getName());

statement.setString(2, s.getSex());

。。。。。。。。。。。。。。。。。。

statement.setInt(13, s.getId());

public List<pay> getAllPay() {

List<pay> listPay = new ArrayList<pay>();

try {

Connection connection = Dbconn.getConnection();

String sql = "select \* from pay";

PreparedStatement statement = connection.prepareStatement(sql);

ResultSet rs = statement.executeQuery();

while (rs.next()) {

pay p = new pay();

p.setId(rs.getInt("id"));p.setLeave(rs.getString("leave")); p.setBelate(rs.getString("belate"));p.setAbsent(rs.getString("absent"));p.setBasepay(rs.getString("basepay"));p.setBonus(rs.getString("bonus"));p.setDepay(rs.getString("depay"));p.setRpay(rs.getString("rpay"));

listPay.add(p);

public pay load(int pararmId) {

pay apay = null;

Connection connection = Dbconn.getConnection();

String sql = "select\* from pay where id=?";

try {PreparedStatement st1 = connection.prepareStatement(sql);

st1.setInt(1, pararmId);

ResultSet rsa = st1.executeQuery();

if (rsa.next()) {apay = new pay();

apay.setId(rsa.getInt("id"))apay.setLeave(rsa.getString("leave"));apay.setBelate(rsa.getString("belate"));apay.setAbsent(rsa.getString("absent"));apay.setBasepay(rsa.getString("basepay"));apay.setBonus(rsa.getString("bonus"));apay.setDepay(rsa.getString("depay"));apay.setRpay(rsa.getString("rpay"));

public static boolean Update(pay s){

boolean result=false;

try {

Connection connection=Dbconn.getConnection();

String sql="UPDATE employee INNER JOIN pay ON employee.id=pay.id SET pay.leave=?,pay.belate=?,pay.absent=?,pay.basepay=?,pay.bonus=?,pay.depay=?,pay.rpay=? WHERE pay.id=? ";

PreparedStatement statement=connection.prepareStatement(sql);

statement.setString(1, s.getLeave());

statement.setString(2, s.getBelate());

。。。。。。。。。。。。。。。。

statement.setInt(8, s.getId());

result=statement.execute();

dbutil包：进行数据库的连接

public class Dbconn {

public static Connection getConnection() {Connection connection=null;

Class.forName("com.mysql.cj.jdbc.Driver");Stringurl="jdbc:mysql://localhost:3306/mydatebase?useSSL=true&characterEncoding=UTF-8&serverTimezone=UTC";String user="root";String password="admin"; connection=DriverManager.getConnection(url, user, password);

entity包：设计属性（字段）；设计访问器

public class staff {

private int id;private String name;private String sex;private String nation;private String phone;private String idcard;private String depart;private String profe;private String email;private String entry;private String school;private String educat;private String birth;

public class pay {

private int id;private String leave;private String belate;private String absent;private String basepay;private String bonus;private String depay;

private String rpay;

control包： request.setCharacterEncoding("utf-8");

request.getContextPath();

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

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

if(name.equals("admin") && password.equals("admin")){

response.sendRedirect("http://pc20200512jwam:8080/QM/servlet/ListStaff);

}else{response.sendRedirect("http://pc20200512jwam:8080/QM/index.jsp");

}

SearchStaff model = new SearchStaff();

List<staff>listStaffs = model.getAllStaff();

request.getSession().setAttribute("staffs",listStaffs); request.getRequestDispatcher("/jsp/staffList.jsp").forward(request, response);

request.setCharacterEncoding("utf-8");

int id=Integer.parseInt( request.getParameter("id"));

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

if(name==null || name.equals("")){

request.setAttribute("error", "姓名不能为空");

request.getRequestDispatcher("/jsp/staffInsert.jsp").forward(request, response);return;}

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

。。。。。。。。。。。

String entry=request.getParameter("entry");staff s=new staff();

s.setId(id);s.setName(name);s.setSex(sex);s.setNation(nation);s.setBirth(birth);s.setPhone(phone);s.setIdcard(idcard);s.setDepart(depart);s.setProfe(profe);s.setEmail(email);s.setSchool(school);s.setEducat(educat);s.setEntry(entry);InsertStaff model=new InsertStaff();boolean result=model.Insert(s);if(result==true){response.sendRedirect(request.getContextPath()+"/servlet/ListStaff");}else{response.sendRedirect(request.getContextPath()+"/jsp/satffInsert.jsp");}

int id=Integer.parseInt(request.getParameter("id"));

DeleteStaff model=new DeleteStaff();boolean r=model.Delete(id);

SearchStaff model2=new SearchStaff();List<staff>list= model2.getAllStaff();request.setAttribute("staffs",list);request.getRequestDispatcher("/jsp/staffList.jsp").forward(request,response);if(r==true) {response.sendRedirect("/servlet/ListStaff");}

request.setCharacterEncoding("utf-8");

int staffId=Integer.parseInt(request.getParameter("id"));

staff s=SelectStaff.selectById(staffId);if(s==null){

}else{request.setAttribute("staff", s); request.getRequestDispatcher("/jsp/staffUpdate.jsp").forward(request, response);return;

request.setCharacterEncoding("utf-8");

int id=Integer.parseInt( request.getParameter("id"));

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

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

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

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

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

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

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

pay s=new pay();

s.setId(id);s.setLeave(leave);s.setBelate(belate);s.setAbsent(absent);s.setBasepay(basepay);s.setBonus(bonus);s.setDepay(depay);s.setRpay(rpay);@SuppressWarnings("unused")boolean r=Function.Update(s);

response.sendRedirect("ListPay");

页面部分：

1.css样式分为6个部分其中

fa文件是图标字体包在系统中只用在在导航栏上面的搜索图标

图片1.jpg是背景图片

图片2.png是在阿里巴巴图标库里下载的图标用在了导航栏中的最左端

base.css导航栏和导航栏下面白色部分的样式（所有页面都需要引入）

.clearfix::before,.clearfix::after {content: '';display: table; clear: both;}

body {background: url(../img/1.jpg) no-repeat center center; background-size: cover;

background-attachment: fixed;}

.outer {width: 100vw; height: 100vh;background-color: rgba(0, 0, 0, .3);}

input { border: 0;outline: none;}

a { text-decoration: none;}

.down\_inner { margin-top: 25px;margin-left: 100px;width: 400px;}

h1 { font-weight: 700;font-size: 25px;margin-bottom: 26px;}

/\*下面整个都是框架样式\*/

.nav\_wrap { background-color: rgba(0, 0, 0, .3);height: 50px; width: 100%;line-height: 50px;}

.nav\_inner { width: 1400px; height: 100%; margin: 0 auto;}

.nav\_inner a { color: white;}

.nav\_left {float: left;}

.nav\_left img { width: 25px; }

.nav\_left span { color: white;line-height: 50px;font-size: 20px;}

.nav\_right {width: 650px; height: 100%; float: right;}

.search {width: 270px;height: 40px; margin: 7px 0; line-height: 50px; float: left;}

.search input {outline: none; width: 230px;float: left; height: 40px; padding: 0 15px; border: none;}

.search button { float: right; width: 40px;height: 40px; border: none;}

.search button:hover {color: white;background-color: #938c8c;}

.exit,.pay,.insert { float: right;}

.aa {display: block;height: 50px;width: 110px;text-align: center;}

.aa:hover { background-color: white; color: #333;}

.down\_wrap {overflow-y: auto; width: 1400px;height: 650px;margin: 0 auto;background-color: white;}

reset.css:重置样式表（略）所有页面都引入

first.css:浏览数据表格的样式（在工资表和员工表中引入）

a { text-align: center;line-height: 40px;display: block;width: 100%;height: 100%; background-color: rgba(0, 0, 0, 0.476); color: white;}

a:hover { background-color: white;color: black;}

table { width: 100%;border: 1px solid black;margin: 0 auto;border-collapse: collapse;}

td, th { border: 1px solid black;height: 40px;vertical-align: middle;text-align: center;}

tbody > tr:nth-child(even) { background-color: #d6d6d6;}

Index.css:登录页面的样式

.inner\_wrapper {background-color: rgba(0, 0, 0, 0.5);width: 450px;height: 300px;margin: 190px auto;border-radius: 18px;}

p,span { color: white;}

.inner\_wrapper p {font-size: 25px; display: block;height: 110px; width: 450px;text-align: center;line-height: 110px;}

.inner\_first,.inner\_sec,.inner\_third {font-size: 15px;line-height: 40px;height: 40px; width: 375px;margin: 0 auto;margin-bottom: 20px;border-radius: 10px;overflow: hidden;}

#username,#password {background-color: #938e8e;height: 40px;width: 375px;}

#sign {width: 375px;height: 40px;background-color: #44dd9b;border: none;}

#sign:hover {background-color: #403f3f;}

Insert.css是插入页面的样式

.down\_wrap {position: relative;box-sizing: border-box;}

.down\_inner { margin-top: 25px; margin-left: 100px;width: 400px;}

h1 {font-weight: 700;font-size: 25px; margin-bottom: 26px;}

.down\_wrap .down\_inner span {display: inline-block;height: 30px;line-height: 35px;}

.inner\_list {margin-right: 200px;font-weight: 300;font-size: 20px; margin-bottom: 7px;width: 100%;height: 30px;}

.inner\_inp {line-height: 30px;font-weight: 300;width: 250px;height: 90%;border-bottom: 1px solid #333; font-size: 17px;float: right;}

.inner\_inp:focus {background-color: rgba(51, 51, 51, 0.054);}

.inner\_btn {box-shadow: 5px 5px 10px #a9a9a9;font-size: 15px;margin-top: 25px;display: inline-block;width: 200px;height: 40px;border-bottom: 1px #333 solid;}

.btn1 { position: absolute;left: 100px;bottom: 25px;}

.btn2 {position: absolute;left: 400px;bottom: 25px;}

.btn3 { position: absolute;right: 100px;bottom: 25px;}

.inner\_btn:hover {background-color: #111111b4;color: white;box-shadow: 5px 5px 10px #a9a9a9;}

search.css是搜索结果页面的样式

.down\_wrap {position: relative;}

.down\_inner {font-size: 20px;font-weight: 300;line-height: 32px;}

..return { box-shadow: 5px 5px 10px #a9a9a9; position: absolute;bottom: 25px;font-size: 15px;display: inline-block;width: 200px;height: 40px;border-bottom: 1px #333 solid;}

.return1 {left: 100px;}

.return2 {left: 380px;}

.return3 {right: 100px;}

return:hover {background-color: #111111b4;color: white;border-bottom: 1px #333 solid;}

Jsp页面代码：页面的导航栏基本相同，所以每个页面只展示.down\_wrap里面结构

其中导航栏的内容为

<div class="nav\_wrap"><div class="nav\_inner"><div class="nav\_left">

<img src="${pageContext.request.contextPath}/img/2.png"></img> <span>清沐酒店人力资源管理系统</span></div>

<div class="nav\_right"><div class="search"><form action="./jsp/sh.jsp" method="post">

<input type="number" placeholder="请输入您要搜索的员工编号" name="payid"οnkeyup="if(/\D/.test(this.value)){alert('只能输入数字');this.value='';}">

<button type="submit"><i class="fas fa-search"></i></button>

</form></div><div class="exit"><a class="aa"href="javascript:if(confirm('确认退出登录吗?'))window.location='index.jsp'">退出系统 </a></div>

<div class="pay"><a class="aa" href="<%=path%>/servlet/ListStaff"> 员工信息表 </a></div></div></div></div>

payList.jsp工资表（接收ListPay.java传输的数据）

<div class="down\_wrap"><table class="table"><c:forEach items="${p }" var="p"><tr><td>${p.rpay}</td><td><a href="<%=path %>/servlet/UpdatePayServlet?id=${p.id}">修改</a></td></tr></c:forEach></table></div>

payUpdate.jsp更改工资（接收UpdatePayServlet的数据并将数据传递到DoUpdatePayServlet进行数据更新）

<form method="post"action="<%=request.getContextPath()%>/servlet/DoUpdatePayServlet">

<div class="inner\_list"><span>编号：</span><input type="number" name="id" value="${pay.id}"class="inner\_inp"οnkeyup="if(/\D/.test(this.value)){alert('只能输入数字');this.value='';}"></div>

<input class="inner\_btn btn1" type="submit" value="保存更新">

<input class="inner\_btn btn2" type="button" value="取消更新"onclick="window.location.href = '<%=path%>/servlet/ListPay'">

<input class="inner\_btn btn3" type="button" value="修改员工信息"onclick="window.location.href = '<%=path %>/servlet/UpdateStaffServlet?id=${pay.id}'"></form>

research.jsp员工信息查询（调用SearchStaff 方法进行数据展示）

int id = Integer.parseInt(request.getParameter("staffid"));

SearchStaff model1 = new SearchStaff();

staff staffq = model1.load(id);sh,jsp工资查询

sh.jsp工资查询（调用SearchPay 方法进行数据展示）

int ida = Integer.parseInt(request.getParameter("payid"));

SearchPay modelq = new SearchPay();

pay pays = modelq.load(ida);

staffInset.jsp新增员工（将数据传递到InsertStaffServlet来增加员工信息）

<form method="post"action="<%=request.getContextPath()%>/servlet/InsertStaffServlet">

<div class="inner\_list"><span>编号：</span><input type="number" name="id" class="inner\_inp" οnkeyup="if(/\D/.test(this.value)){alert('只能输入数字');this.value='';}"></div>

staffList.jsp员工信息表（接收Liststaff.java传输的数据并进行展示）

<c:forEach items="${staffs }" var="staffss"><tr>td>${staffss.birth }</td><td><a

href="<%=path %>/servlet/UpdateStaffServlet?id=${staffss.id}">修改</a></td>

<td><a href="javascript:if(confirm('确认删除吗?'))window.location='<%=path %>/servlet/DeleteStaffServlet?id=${staffss.id}'">删除</a></td></tr></c:forEach>

staffUpdate.jsp员工信息更新表（接收UpdateStaffServlet的数据并将数据传递到DoUpdateStaffServlet进行数据更新）

<form method="post" action="<%=request.getContextPath()%>/servlet/DoUpdateStaffServlet">

<input class="inner\_btn btn3" type="button" value="修改工资"onclick="window.location.href = '<%=path %>/servlet/UpdatePayServlet?id=${staff.id}'">

Index.jsp登录首页（登录程序中心结构）

<form action="./servlet/Login" method="post">

<div class="inner\_first">

<input type="text" name="username" id="username" placeholder="管理员姓名" autocomplete="off"><br></div><div class="inner\_sec"><input type="password" name="password" id="password" placeholder="管理员密码"><br></div>

<div class="inner\_third">

input type="submit" value="登 录" id="sign" /></div></form>

课程设计心得体会

经过全组人员的共同努力,我们终于完成了本课程设计。从当初定下题目到最后一个功能模块的完成，我们经历了无数次的错误->修改代码->重启服务器->运行的过程。感觉到平时学的知识是多么的浅薄，书到用时方恨少，现在是体验的真真切切。由于对JSP和Tomcat以及MyEclipse的使用不足够熟悉，所以本次实验只选取的原计划中的几个重点模块进行编写代码实现。并且通过本次实验，我们也感受到了开源的方便,遇到什么问题，搜索一下也知道自己其实并不孤独，在编程的路上我们还有很长的路要走。

在此课程设计中我们进行的总结有：本系统的特点1.根据具体人事管理情况开发，针对性强；2.基本能够满足人事管理的需求；3.易于管理和维护；4.可移植性强，可在多种系统下运行。本系统的不足：由于我们开发小组的人员，水平，有限，开发时间较短，没有过相关的开发经验，大部分组员都在一边学习，一边用于开发，使得这套系统在很多细节方面还需要修补和完善。例如，在前期工作判断失误，显示结果不好，导致再重新确定变量；程序整体细节上有很多需要修改完善的地方。

通过本次实验，让我们对程序有了更深一步的了解，提高了我们自身的编程能力，优化了我们的编程思维，也培养了独立思考问题和自学能力。我们也深刻的意识到必须不断学习，用心思考才能在工作中体现自己的价值,来适应社会的需要。

**致谢**

首先衷心感谢王铁老师在整个课题设计期间给予我们无微不至的关怀和细心的指导。在设计过程中指导老师为我们提出了许多宝贵的意见和建议，谨向王老师表示深深的谢意。