**基于MVC的企业人事管理系统的设计与实现**

**姓名：肖 雅 支 学号：2015874123**

**摘 要：** 随着计算机科学的高速发展，越来越多的企业利用计算机帮助人们对企业进行管理。本人结合当今企业管理系统的主流特点，采用MVC设计模式，通过选择Java为开发语言,并以JSP，Servlet技术为基础，采用MySQL为后台数据库, 实现了对员工、部门、工资信息的增删查改四大功能以及用户修改个人信息的功能。这些功能的实现使得企业对员工、部门和工资信息可以更高效便捷的管理。

1. 目的与意义

企业人事管理系统是一个企业单位不可缺少的部分，是适应现代企业制度，推动企业人事管理走向科学化、规范化、自动化的必要条件。为了使企业的管理层对员工有一个高效的管理，本系统一方面实现了公司的人力资源部管理员对公司里的员工、部门以及工资信息进行高效便捷的管理，另一方面为员工提供一个查看公司当前人事情况和修改自己信息的平台。

管理员页面分为三大模块，分别是：员工、部门、工资模块。每个模块都包含增、删、查、改四个功能。当企业中出现了员工调动、部门变动或者是工资更改的情况，都可以通过本系统进行相应的管理。这些功能为公司领导提供充足的信息和快捷的查询手段，以帮助企业领导了解企业现有的人力资源状况。为人事管理和高层决策领导提供高校的决策支持。

员工（即普通用户）页面分为四大模块，分别是员工、部门、工资和修改个人信息模块。其中员工、部门和工资模块，普通用户只拥有查找和查看功能，不能随意新增、删除、或者修改数据。修改个人信息模块中，普通用户只可以修改自己的个人信息，不能随意更改其他员工的基本信息，这在一定层度上保护了用户的隐私信息。

总而言之，本系统是一个企业单位不可缺少的部分，能在很大层度上提高公司管理的高效性，推动企业人事管理走向科学化、规范化、自动化。

1. 理论基础及技术路线

2.1 B/S访问模式

本系统所有的页面访问模式都是采用了B/S访问模式，B/S(浏览器/客户端)模式实现，B/S(Browser/Server)结构，也就是浏览器/服务器模式，是在当前web数据库的开发中应用比较广泛的一种,它是随着计算机技术的崛起，从传统的C/S(客户机/服务器)模式发展起来的网络结构。在B/S模式中，客户端运行浏览器软件。浏览器以超文本形式向Web服务器提出访问数据库的要求，Web服务器接受客户端请求后，将这个请求转化为SQL语法，并交给数据库服务器，数据库服务器得到请求后，验证其合法性，并进行数据处理，然后将处理后的结果返回给Web服务器，Web服务器再一次将得到的所有结果进行转化，变成HTML文档形式，转发给客户端浏览器以友好的Web页面形式显示出来。用户界面是通过访问浏览器来实现的，很小部分事务逻辑是在前端(Browser)实现的。但其主要的事务逻辑还是在服务器端(Server)实现的。这样就大大减轻了客户端负担，降低了系统维护的成本和难度。

2.2 MVC设计模式

本系统采用了MVC设计模式，MVC全名是Model View Controller，是模型(model)、视图(view)、控制器(controller)的缩写，一种软件设计典范，用于组织代码，用一种业务逻辑和数据显示分离的方法。本系统控制器类放在servlet包下，业务逻辑类放在dao包下，数据传输类放在dto包下。这个方法的假设前提是如果业务逻辑被聚集到一个部件里面，而且界面和用户围绕数据的交互能被改进和个性化定制而不需要重新编写业务逻辑，MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

视图是用户看到并与之交互的界面。对老式的Web应用程序来说，视图就是由HTML元素组成的界面，在新式的Web应用程序中，HTML依旧在视图中扮演着重要的角色，但一些新的技术已层出不穷，它们包括Adobe Flash和象XHTML、XML/XSL、WML等一些标识语言和Web services。MVC好处是它能为应用程序处理很多不同的视图。在视图中其实没有真正的处理发生，不管这些数据是联机存储的还是一个雇员列表，作为视图来讲，它只是作为一种输出数据并允许用户操纵的方式。

模型表示企业数据和业务规则。在MVC的三个部件中，模型拥有最多的处理任务，例如它可能用像EJBs和ColdFusion Components这样的构件对象来处理数据库，被模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型能为多个视图提供数据，由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

控制器接受用户的输入并调用模型和视图去完成用户的需求，所以当单击Web页面中的超链接和发送HTML表单时，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后再确定用哪个视图来显示返回的数据。

2.3 Tomcat

Tomcat是一个免费的开源的Serlvet容器，在Tomcat中，应用程序的部署很简单，只需将WAR放到Tomcat的webapp目录下，Tomcat会自动检测到这个文件，并将其解压。当在浏览器中访问这个应用的JSP时，通常第一次会很慢，因为Tomcat要将JSP转化为Servlet文件，然后编译。编译以后，访问将会很快。另外Tomcat也提供了一个应用：manager，访问这个应用需要用户名和密码，用户名和密码存储在一个xml文件中。通过这个应用，辅助于Ftp，你可以在远程通过Web部署和撤销应用。当然本地也可以。

本系统是基于Tomcat开发的，其实主要是JSP和Servlet的开发，开发JSP和Servlet非常简单，我们可以用普通的文本编辑器或者IDE，然后将其打包成WAR即可。另外，在Servlet1.2中，可以利用标签库实现Java代码与Html文件的分离，使JSP的维护更方便。

2.4 JSP

本系统中的JSP页面包含了JavaScript、自定义标签、JSTL标签的内容。JSP(Java Server Pages)是一种动态网页技术标准。 JSP技术使用JAVA作为脚本语言， JSP页面会为整个服务器端的JAVA库单元提供一个接口来为HTTP的应用程序服务。JSP技术通过在传统的网页文件中插入java程序段和JSP标记，从而形成JSP网页。本系统中通过大量运用自定义<%= %>插入java代码片段，从数据库中读取到信息，再<%= %>把数据显示在页面上， 从而实现数据的接收和提交。程序操作均在服务器端运行，传送给客户端的仅是得到的结果，与客户端的浏览器无关，由此解决了客户端的限制。

2.5 Servlet

Servlet是在服务器上运行的小程序。这个词是在Java applet的环境中创造的，Java

applet是一种当作单独文件跟网页一起发送的小程序，它通常用于在客户端运行，结果得到为用户进行运算或者根据用户互作用定位图形等服务。在服务器上运行Java，这种程序可使用Java编程语言实现。在通信量大的服务器上，Javaservlet的优点在于它们的执行速度更快于CGI程序。各个用户请求被激活成单个程序中的一个线程，而无需创建单独的进程，这意味着服务器端处理请求的系统开销将明显降低。

2.6 MySQL

MySQL是本系统采用的后台数据库，MySQL是一个关系型数据库管理系统，MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

1. 模块总体设计
   1. 用例图

用户分为两大类：

管理员：可以对员工、部门、工资表的信息进行增、删、查、改四大操作

普通员工：可以对员工、部门、工资表的信息进行查找操作以及对个人信息进行修改操作



* 1. 数据流图

数据流图所用到的基本符号和意义：





图一：顶层数据流图



图二：分层数据流图

* 1. 模块图

本系统分为管理员界面和普通户界面：

管理员界面有三大模块：员工信息、部门信息和工资信息模块，各个模块都采用MVC模式

普通用户界面有四大模块：员工信息、部门信息、工资信息和个人信息模块，各个模块都采用MVC模式



* 1. E-R图

E-R图所用的基本符号和意义如下所示：





图一：员工信息E-R图



图二：部门信息E-R图



图三：工资信息E-R图



图四：整体E-R图

* 1. 数据库表

表1 员工信息表（employee）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 功能描述 |
| staffid | varchar | 20 | 员工编号（主键） |
| staffname | varchar | 20 | 员工姓名 |
| staffpassword | varchar | 20 | 员工密码 |
| ssuper | int | 1 | 权限值（默认为0） |
| staffflag | int | 1 | 软删除标志（默认为1） |

表2 部门信息表（department）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 功能描述 |
| departmentid | varchar | 20 | 部门编号（主键） |
| departmentname | varchar | 20 | 部门姓名 |
| departmentflag | int | 1 | 软删除标志（默认为1） |

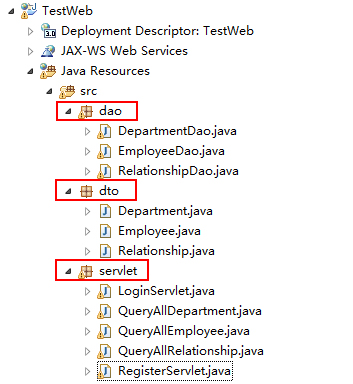
表3 工资信息表（relationship）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 功能描述 |
| staffid | varchar | 20 | 员工编号（主键） |
| departmentid | varchar | 20 | 部门编号（主键） |
| salary | varchar | 10 | 员工工资 |
| work | varchar | 10 | 所在职位 |
| rflag | int | 1 | 软删除标志（默认为1） |

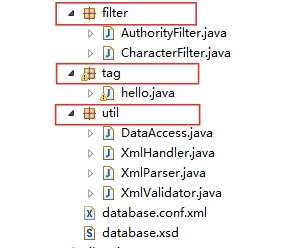
1. 系统实现

4.1开发环境说明

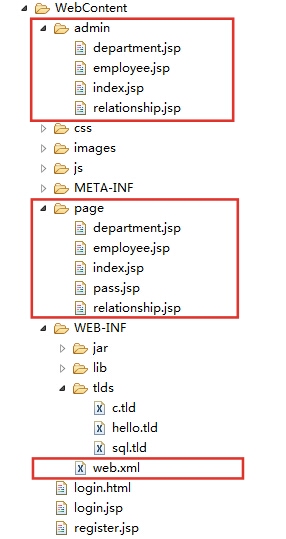
* 采用MVC模式设计整个系统，控制器类放在servlet包下，业务逻辑类放在dao包下，数据传输类放在dto包下：



* 工具类放在util包下，自定义标签放在tag下，过滤器（字符过滤器、权限过滤器）放在filter包下：



* 管理员页面放在admin下，普通用户页面放在page下， 以及配置文件web.xml：

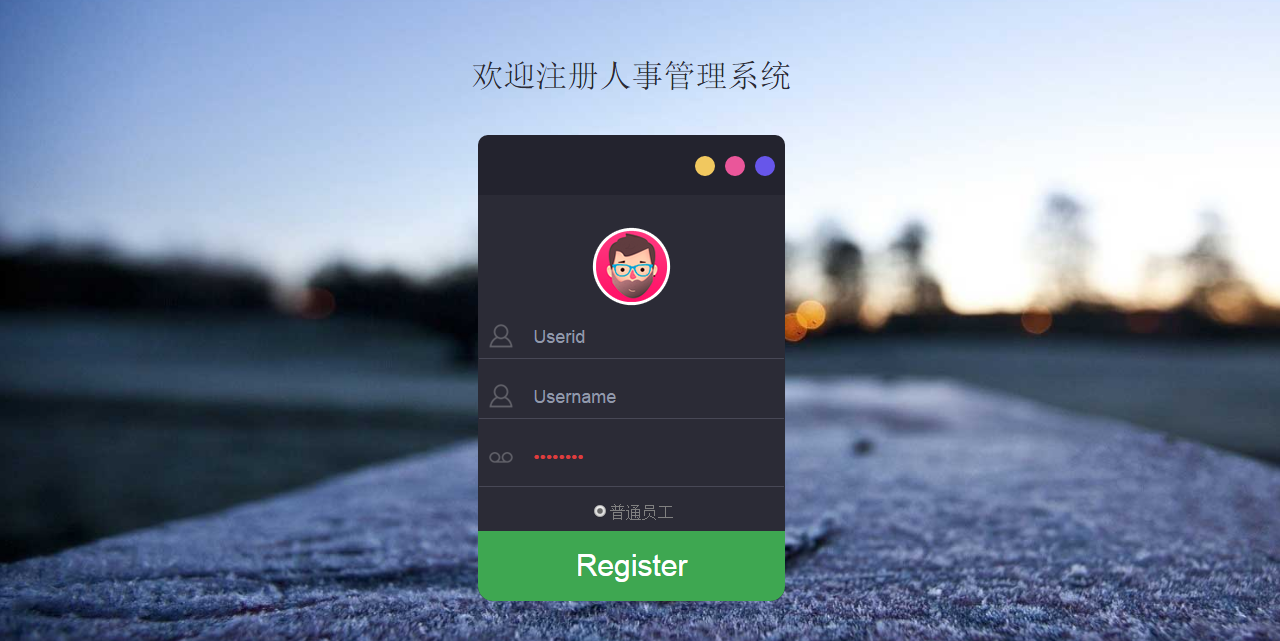


4.2系统运行说明

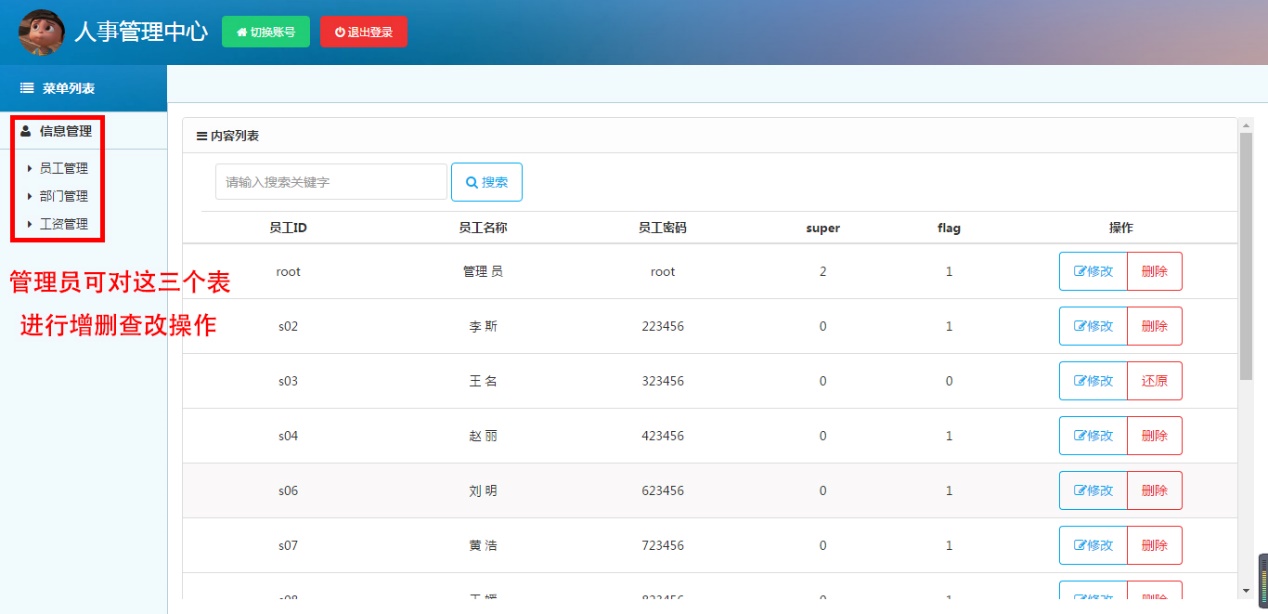
* 登录页面：
* 可以从登录界面切换到注册界面
* 登录分为管理员登录和普通员工登录两种，其中管理员权限页面的访问必须验证用户权限，权限的控制采用过滤器完成



* 注册页面：
* 注册的时候只能注册普通员工，考虑到安全隐患和权限问题，管理员是不可以随意注册的
* 注册成功会自动跳转到登录界面
* 注册失败会有相应错误提示



* 管理员界面：
* 管理员可以对员工、部门和工资表的数据进行增、删、查、改四大功能
* 通过使用字符过滤器实现对中文的转译



* 管理员添加数据界面：
* 当添加的主键已经存在时，会显示添加失败的提示



* 添加成功或者失败提示：

* 管理员删除数据界面：
* 使用软删除方法，数据删除之后flag值为0
* 通过还原按钮把数据还原，并把flag值赋为1





* 管理员查询数据界面：
* 使用模糊查询，通过输入关键字就可以查询到多条数据：



* 管理员修改数据界面：
* 点击修改按钮时，会弹出修改页面的遮罩层，系统获取了需要修改数据的原始值，可以在原数据的基础上修改，使得操作更加人性化：



* 当修改的新数据中，如果新的主键已经存在、员工ID不存在、部门ID不存在的时候，会显示修改失败提示：



* 修改成功会有修改成功的提示：



* 普通用户界面：
* 普通用户界面可以对员工、部门和工资表的信息进行查询和查看
* 普通用户可以对自己的个人信息进行修改



* 普通用户的个人信息修改界面：
* 当用户没有输入正确的原始密码时会出现错误提示
* 当用户的新置密玛没有写时会出现错误提示



5、总结

本人事管理系统实现的整个过程经过了较长的时间，把老师上课讲的每个知识点几乎都学以致用到系统的实现中。本系统实现了人事管理系统对企业人事管理的基本操作，麻雀虽小五脏俱全。通过这次的实践，对系统的开发整体框架和流程有了整体的把握，巩固了所学的知识。明白了书本上的知识只有真正付诸实践才能真正掌握。

一开始遇到最大的困难就是在做好了静态的网页（\*.html）之后，不知道要怎么改成动态网页（\*.jsp），这个不会做接下来的都做不下去了，于是我找了之前老师的录屏还有去网上看论坛看教程，并且与同学讨论，最后终于解决了。接下来就很顺利的进行下去，写代码遇到了不少bug，通过和同学交流讨论以及看博客看论坛得到结论，在不断地查找错误调试后顺利解决了，也明白了编程中要注意细节，可能会仅仅因为一个英文单词报出几十行bug，从错误中学习到很多以前不以为然的事情。写程序最主要的是沟通与交流，闭门造车的方法在编程中是不可取的，多与同学老师交流，在别人的想法中可以发现新的解决方法，同时我意识到了自身的不足，需要在以后的学习中不断补充和拓展。我希望以后有更多的机会让我们独立去完成一个系统，或者是小组完成，这样可以很好的锻炼我们的实践能力。

**参考文献**

[1] 燕卫. 计算机辅助教学中有关问题研究与实践初探[J]. 福建电大泉州分校,2008,7(4):31～47.

[2] 张立敏,曾绍庚. Java Webf 应用开发基础教程[M]. 北京:中国水利水电出版社, 2016.

-------------------------------------------------------------------------------------------------------