



# 毕业设计（论文） 说明书

设计(论文)

人事管理系统

题 目

专 业

班 级

设计人

学 号

指导教师

山东科技大学

2008 年 6 月 3 日

# 毕业设计（论文）任务书

发给 \_\_\_\_\_ 系 \_\_\_\_\_ 专业 \_\_\_\_\_ 班 学生 \_\_\_\_\_

一、毕业设计(论文)题目:

\_\_\_\_\_

二、设计专题(子题):

\_\_\_\_\_  
\_\_\_\_\_

三、毕业设计(论文)原始资料:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

四、设计(论文)应解决下列各主要问题:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

五、设计(论文)附件(包括:图纸、主要源程序、软件光盘、科技译文等):

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

六、命题发出日期: \_\_\_\_\_ 设计( 论文) 完成日 \_\_\_\_\_

指导教师 (签 \_\_\_\_\_)

教研室主任 (签 \_\_\_\_\_)

系主任 (签 \_\_\_\_\_)

\_\_\_\_\_

年 月 日

# 指导教师对毕业设计的评语

指导教师（签章）： \_\_\_\_\_

年 月 日

---

## 特邀评阅人意见

评阅人（签章）：\_\_\_\_\_

职 称（签章）：\_\_\_\_\_

年 月 日

# 答辩（考试）委员会鉴定意见

答辩（考试）成绩： \_\_\_\_\_

鉴定意见：

主任（签章）： \_\_\_\_\_

副主任（签章）： \_\_\_\_\_

年 月 日



## 摘 要

在现代化的教育技术模式下，计算机已经广泛应用于企业的人事管理，给传统的人事管理方式带来了重大的革命。如何有效进行人事管理是摆在我们面前的一个新的课题。

本文在研究目前各个企业人事管理情况的基础上，从企业的实际需求出发，采用当前较为流行的 B/S（浏览器/服务器）结构和 JSP 动态网页开发技术，基于 WindowsXP 操作系统和 SQL 数据库，设计并实现了一个功能较为完善的小型企业管理系统。通过对系统进行一系列的需求分析、设计、编码、测试等工作，使系统实现了对企业员工信息的查看、添加、修改、删除的功能，以及对员工培训、奖惩、调动的管理功能。重点对系统设计过程作了相应描述，同时也对系统需要改进的地方做了总结。本系统经过测试，运行效果稳定，操作方便、快捷。适用于一般企业的人事管理要求。

**关键词：**JSP；SQL 数据库；人事管理

## ABSTRACT

In the modern educational technology mode, the computer has been widely used in the personnel management of enterprises, to the traditional personnel management methods brought about a major revolution. How effective personnel management is in front of us a new topic.

Based on the study the current management of all enterprise personnel on the basis of the actual demand from enterprises starting with the more popular B / S (browser / server) JSP structure and dynamic web development technology, based on WindowsXP Server operating system and SQLdatabase designed and implemented a small function better enterprise management system. Through a series of system requirements analysis, design, coding, testing, the system has information on the employees view, add, modify, delete, and search functions. Focus on the system design process accordingly described, but also the need to improve the system of localities to sum up. After testing the system, operating results stability, and easy to operate, fast. Apply to general corporate personnel management requirements.

**Keywords:** JSP; SQL database; Personnel Management

# 目 录

1	绪 论.....	1
1.1	引言.....	1
1.2	概述.....	1
2	相关技术简介.....	2
2.1	JSP 概述.....	2
2.2	JSP 运行环境.....	3
2.2.1	硬件环境.....	3
2.2.2	软件环境.....	4
2.3	安装和配置 JDK.....	4
2.4	安装 Tomcat.....	4
2.4	JavaScript 的实现.....	5
2.4	隐藏验证码技术.....	5
3	系统分析与设计.....	7
3.1	需求分析.....	7
3.2	可行性分析.....	7
3.2.1	经济可行性分析.....	7
3.2.2	技术可行性分析.....	8
3.2.3	操作可行性分析.....	8
3.2.4	系统结构可行性分析.....	9
3.3	总体设计.....	9
3.3.1	项目规划.....	9
3.3.2	设计目标.....	10
3.3.3	系统功能结构图.....	10
4	系统功能的实现.....	11
4.1	模块功能介绍.....	11
4.1.1	登陆模块.....	11
4.1.2	人员档案模块.....	11
4.1.3	人员调动模块.....	11
4.1.4	奖惩管理模块.....	11
4.1.5	培训管理模块.....	11
4.1.6	职称评定模块.....	12
4.2	数据库的设计实现.....	12
4.2.1	数据库的需求分析.....	12
4.2.2	数据库表的设计.....	12

## 目 录

4.2.3 数据库的测试连接.....	15
4.2.4 注册数据库驱动程序.....	16
4.2.5 建立数据库连接.....	17
4.3 模块功能的实现.....	18
4.3.1 登陆模块.....	18
4.3.2 人员档案模块.....	18
4.3.3 奖惩管理模块.....	20
4.3.4 人员调动管理模块.....	21
4.3.5 培训管理模块.....	23
4.3.6 职称评定管理模块.....	24
5 系统演示.....	27
6 系统测试与维护.....	29
6.1 系统测试.....	29
6.1.1 单元测试.....	29
6.1.2 集成测试.....	29
6.1.3 测试用例.....	30
6.2 系统性能.....	31
7 结束语.....	32
参考文献.....	33
致 谢.....	34
附 录.....	35
A 主要源程序.....	35
B 用户手册.....	58
B.1 系统功能简介.....	58
B.2 系统的支持平台.....	58
B.3 安装说明.....	59
B.4 使用说明.....	59
B.5 系统维护方法.....	59
C 软件光盘.....	61
C.1 光盘的树形目录.....	61
C.2 光盘文件一览表.....	61
D 科技译文.....	62
Analysis of network security technology.....	62
浅析网络安全技术.....	69



## 2 相关技术简介

### 2.1 JSP 概述

JSP 技术是由 Servlet 技术发展起来的，自从有了 JSP 后，在 Java 服务器端编程中普遍采用的就是 JSP，而不是 Servlet。因为 JSP 在编写表示页面时远远比 Servlet 简单，并且不需要手工编译，目前 Servlet 主要用于做视图控制器，处理后台应用等。由于 JSP 构建在 Servlet 上，所以它有 Servlet 所有强大的功能。

JSP 基于强大的 Java 语言，具有良好的伸缩性，与 Java Enterprise API 紧密地集成在一起，在网络数据库应用开发领域具有得天独厚的优势，基于 Java 平台构建网络程序已经被越来越多的人认为是未来最有发展前途的技术。

从 JSP 这几年的发展来看，JSP 已经获得了巨大的成功，它通过和 EJB 等 J2EE 组件进行集成，可以编写出处理具有大的伸缩性，高负载的企业级应用。JSP 技术在多个方面加速了动态 Web 页面的开发。

通过分析和总结，列出了以下几点说明 JSP 的特点。

1. 绝大多数 JSP 页面依赖于可重用的，跨平台的组件来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使用这些组件为更多的使用者或者客户团体所使用。基于组件的方法加速了总体开发过程，并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

2. JSP 能提供所有的 Servlets 功能

与 Servlet 相比，JSP 能提供所有的 Servlets 功能，它比用 `Println` 书写和修改 HTML 更加方便。此外，您可以更明确地进行分工，Web 页面设计

人员编写 HTML，只需要留出空间让 Servlets 程序员插入动态部分即可。

### 3. 一次编写，随处运行

作为 Java 平台的一部分，JSP 拥有 Java 编程语言“一次编写，随处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中，您可以使用自己所选择的服务器和工具，但并不影响当前的应用。

### 4. Java 中连接数据库的技术是 JDBC

很多数据库系统带有 JDBC 驱动程序，Java 程序通过 JDBC 驱动程序与数据库相连，执行查询，提取数据等操作。Sun 公司还开发了 JDBC-ODBC Bridge，用此技术 Java 程序就可以访问带有 ODBC 驱动程序的数据库，目前大多数数据库系统都带有 ODBC 驱动程序，所以 Java 程序能够访问如 Oracle，Sybase，MS SQL Server 和 MS Access 等类型的数据库。

### 5. JSP 的平台适应性更广

这是 JSP 与 ASP 的优越之处。几乎所有的平台都支持 Java，JSP+Beans，它们可以在任何平台下通行无阻。Windows NT 下的 IIS 通过一个插件就能支持 JSP，例如 JRUN 或者 ServletExec，著名的 Web 服务器 Apache 已经能够支持 JSP。由于 Apache 广泛应用在 Windows NT，UNIX 和 Linux 上，因此 JSP 有更广泛的运行平台。

## 2.2 JSP 运行环境

### 2.2.1 硬件环境

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。JSP 服务器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: PentiumIII以上;

内存: 512MB。

## 2.2.2 软件环境

浏览器要支持 HTML 元素的表达, WEB 服务器为 Apache Tomcat 系列, 开发工具为 MyEclipse。

## 2.3 安装和配置 JDK

JDK 中包含了 Java 开发中必需的工具和 Java 程序运行环境。可以从 SUN 公司网站 <http://java.sun.com> 免费下载。

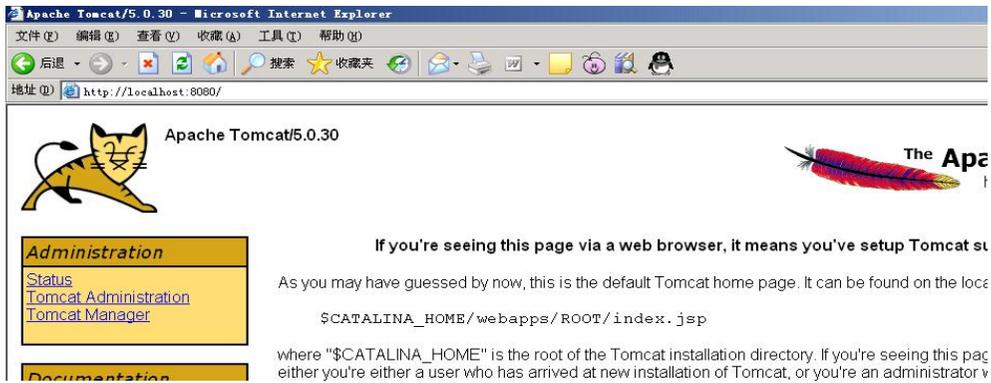
1. 在其页面上分类选择下载安装程序 j2sdk-1\_4\_2-windows-i586, 进入安装界面。
2. 根据安装向导安装到一个目录下, 建议安装目录为 c:\j2sdk。
3. JDK 安装结束。

安装好 JDK 后, 还需要在环境变量中进行对应的配置。主要进行以下配置:

```
JAVA_HOME=C:\j2sdk1.4.2 (安装路径)
CLASSPATH=C:\j2sdk1.4.2\lib\dt.jar;C:\j2sdk1.4.2\bin;C:\j2sdk1.4.2\lib\tools.jar;.;
PATH=C:\j2sdk1.4.2\bin;
```

## 2.4 安装 Tomcat

Tomcat 的安装包可以从 sun 公司的网站获得。下载完后点击安装, 完毕后再地址栏输入 <http://localhost:8080> (Tomcat 默认端口为 8080), 或 <http://127.0.0.1:8080>。如果在浏览器中看到如下 Tomcat 的欢迎界面, 表示 Tomcat 工作正常。测试图:



测试 Tomcat 图

注意：本系统的 web 程序都是在 Tomcat 的 webapps 文件夹的目录下存放。

## 2.4 JavaScript 的实现

为使本系统使用过程中更人性化，本系统中加入了一些 JavaScript 代码，即日历的实现，介于本人水平有限，系统中使用的较难的 JavaScript 代码一部分摘自互联网，其他的是参考其他书中所介绍的案例。

## 2.4 隐藏验证码技术

在一个有密码保护的 Web 应用中，正确处理用户退出过程并不仅仅只需调用 HttpSession 的 invalidate() 方法。现在大部分浏览器上都有后退和前进按钮，允许用户后退或前进到一个页面。假如用户在退出一个 Web 应用后按了后退按钮浏览器把缓存中的页面呈现给用户，这会使用户产生疑惑，他们会开始担心他们的个人数据是否安全。因此为了提高系统的安全性，就需要使用一些隐藏域。

隐藏域，就是可以将要传递的信息隐藏以后传递给下一个页面。一个用户通过浏览器访问一个页面 Web 应用展现一个登陆页面，需要用户输入

有效的验证信息。用户输入了用户名和密码后，此时我们假设用户提供的身份验证信息是正确的，经过了验证过程，Web 应用允许用户浏览他有权访问的区域。用户想退出时，点击退出按钮，Web 应用需要用户确认他是否真的需要退出，假如用户确定退出，Session 结束，Web 应用重新定位到登陆页面。由于使用隐藏域用户可以放心的离开而不用担心他的信息会泄露。另一个用户坐到了同一台电脑前，他点击后退按钮，Web 应用不应该出现上一个用户访问过的任何一个页面。事实上，Web 应用在第二个用户提供正确的验证信息之前应当一直停留在登陆页面上。

隐藏域具体实现部分代码如下：

```
<input id="hiddenCode" type="hidden"
value="<%= (String) session.getAttribute("certCode") %>" ></input>
var hiddenCode = document.myform.hiddenCode.value;
if (hiddenCode != document.myform.certCode.value) {
document.myform.certCode.focus();
alert("\n\n"+"验证码输入错误");
return false;
}
```



时又减少了数据的流通环节，不必要花费那么多的时间，也就是说最重要的是提高了效率，而又保证了各项数据的准确性，也避免了工作人员的流动造成的收据丢失等问题，适应了当前的发展形式。

本系统是为各个企业公司进行服务的，使用范围广泛，具有很强的使用推广性，能够为公司企业节省下大量的人力资源，并提升公司的人力资源管理水平，一旦投入使用可以迅速见到成效，经济可行性极高，具有很好的使用推广价值。

### 3.2.2 技术可行性分析

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，使得 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 PostgreSQL 等类型的数据库。

### 3.2.3 操作可行性分析

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。

### 3.2.4 系统结构可行性分析

本系统采用 B/S 模式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，只要安装浏览器便可以。

## 3.3 总体设计

### 3.3.1 项目规划

人事管理系统由人员信息管理、人员调动信息管理、人员培训管理、人员职称评定、系统用户管理等几个部分组成，规划系统功能模块如下：

人员信息管理模块：主要功能是对员工基本信息等进行管理。

人员调动信息模块：主要是对员工调动信息进行管理。

人员培训管理模块：主要是对员工培训安排进行管理。

人员职称评定模块：主要是对员工职称评定进行管理。

系统用户管理模块：主要功能是管理员密码修改功能的管理。

### 3.3.2 设计目标

通过本系统软件，按照人资管理科学体制，结合公司共同特点，利用先进的计算机数据库技术实现以下功能：

管理员可以方便、快捷地对员工进行添加、删除、奖罚、培训、职称评定、信息修改等管理。

形象、科学、完整地表现人员调整及相关信息的改变。

可以快捷、准确查阅员工、部门等重要信息。

### 3.3.3 系统功能结构图

人事管理系统所有模块和功能的结构图如图 3.1。

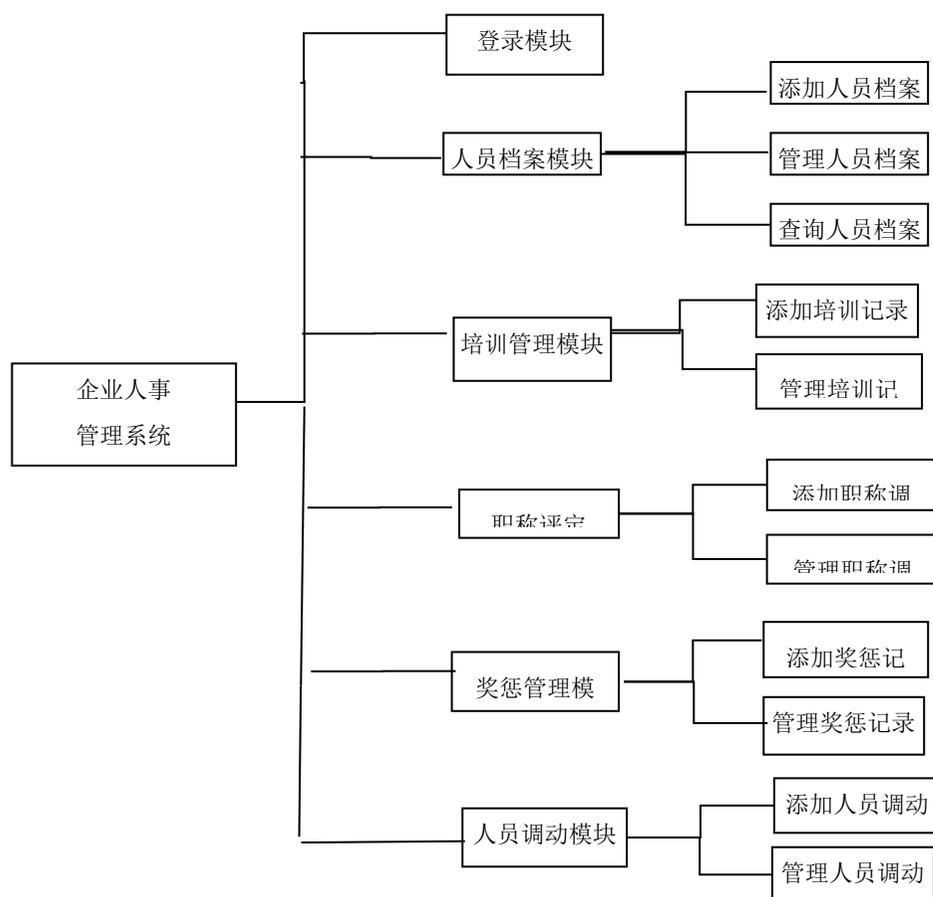


图 3.1 企业员工管理系统功能结构图

## 4 系统功能的实现

### 4.1 模块功能介绍

#### 4.1.1 登陆模块

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，进行各种操作。

#### 4.1.2 人员档案模块

此模块实现对企业所有员工的档案进行管理，并且根据需要可以对某个员工的档案进行修改、删除和查询。

#### 4.1.3 人员调动模块

本模块提供了管理企业人员调动信息管理的功能，可以根据需要对某条调动记录进行修改和删除。

#### 4.1.4 奖惩管理模块

本模块提供了记录奖惩类别、内容、原因等奖惩信息的管理。可以根据需要添加奖惩详细信息、删除奖惩详细信息、修改奖惩详细信息。

#### 4.1.5 培训管理模块

本模块提供了安排企业人员进行培训管理的功能，可以根据需要对某员工培训记录进行修改和删除。

## 4.1.6 职称评定模块

此模块实现对企业职工职称进行管理，并且根据需要可以对某个员工的职称进行修改、删除。

## 4.2 数据库的设计实现

### 4.2.1 数据库的需求分析

依据项目的处理需求,对应数据表的设计及功能如下:

1. 员工基本信息表：主要用于保存员工信息。
2. 奖惩管理信息表：主要用于保存奖惩信息。
3. 人事调动管理信息表：主要用于保存内部人员调动信息。
4. 系统用户管理表：主要用于保存用户信。
5. 培训管理表：主要用于保存员工培训信息。
6. 职称评定表：主要用于保存员工职信息。

### 4.2.2 数据库表的设计

本系统数据库有以下几张表组成:

表 4.1 人事调动管理信息表 (translate)

名称	字段名	类型	长度	主键	非空
记录编号	TranslateNO	int	——	yes	yes
员工编号	EmployeeID	varchar	6	no	yes
调动时间	TranslateDate	date	——	no	yes
前部门	PriorBranch	varchar	40	no	yes
现部门	NextBranch	varchar	40	no	yes
前职务	PriorDuty	varchar	40	no	yes
现职务	NextDuty	varchar	40	no	yes
经办人	HandleName	varchar	20	no	yes
备注	Remark	varchar	400	no	no

表 4.2 员工基本信息表（employee）

名称	字段名	类型	长度	主键	非空
员工编号	EmployeeID	varchar	6	yes	yes
姓名	UserName	varchar	20	no	yes
性别	Sex	varchar	10	no	yes
部门	Branch	varchar	40	no	yes
出生日期	Birthday	date	——	no	yes
籍贯	NativePlace	varchar	20	no	yes
婚姻	Marriage	varchar	20	no	yes
身份证号	IdentityID	varchar	18	no	yes
政汉面貌	Politics	varchar	20	no	yes
民族	Folk	varchar	20	no	yes
学历	Education	varchar	20	no	no
专业	Department	varchar	40	no	no
毕业时间	GraduateDate	date	——	no	no
毕业院校	University	varchar	40	no	no
公积金账号	AccumulateID	varchar	20	no	yes
行政级别	AdministrationLevel	varchar	20	no	yes
职务	Duty	varchar	40	no	yes
职称	Position	varchar	20	no	yes
外语语种	ForeignLanguage	varchar	20	no	no
外语等级	FLLevel	varchar	20	no	no
计算机等级	ComputerLevel	varchar	20	no	no
家庭电话	Phone	varchar	20	no	no
手机号码	MobilePhone	varchar	11	no	no
家庭住址	Address	varchar	50	no	no
在职情况	Incumbency	varchar	20	no	yes
用式形式	IncumbencyType	varchar	20	no	yes
个人简历	Resume	varchar	400	no	no

表 4.3 系统用户管理表（users）

名称	字段名	类型	长度	主键	非空
用户名	UserName	varchar	16	yes	yes
密码	Password	varchar	16	no	yes

表 4.4 奖惩管理信息表 (reward)

名称	字段名	类型	长度	主键	非空
记录编号	RewardNO	int	——	yes	yes
员工编号	EmployeeID	varchar	6	no	yes
职称	Position	varchar	20	no	yes
奖/惩	RewardPunish	varchar	10	no	yes
奖惩内容	Rewardcontent	varchar	50	no	yes
原因	Reason	varchar	100	no	yes
批准部门	Branch	varchar	40	no	yes
审核人	HandleName	varchar	20	no	yes
开始时间	StartDate	date	——	no	yes
撤销时间	CancelDate	date	——	no	no
撤销原因	CancelReason	varchar	100	no	no
备注	Remark	varchar	400	no	no

表 4.5 培训管理信息表 (train)

名称	字段名	类型	长度	主键	非空
员工编号	EmployeeID	varchar	10	no	yes
培训名称	TrainName	varchar	20	no	yes
培训名称	TrainWhat	varchar	30	no	yes
开始时间	StartDate	Date	--	no	yes
结束时间	EndDate	Date	--	no	yes
培训单位	TrainOrgan	varchar	30	no	yes
培训结果	TrainResult	varchar	30	no	yes
备注	Remark	varchar	300	no	no

表 4.6 职称评定信息表 (title)

名称	字段名	类型	长度	主键	非空
员工编号	EmployeeID	varchar	10	no	yes
职称	position	varchar	20	no	yes
取得日期	GetDate	Date	--	no	yes
取得方式	Getposition	varchar	50	no	yes
职务	Duty	varchar	30	no	yes
开始时间	StartDate	Date	--	no	yes
结束时间	EdDate	Date	--	no	yes
备注	Remark	varchar	300	no	no

1. 员工基本信息表 (employee), 是用来记录每一名员工的基本信息的, 如受教育的情况;

2. 系统用户管理表 (users), 是用来存放系统管理员的信息的, 如用户名, 密码;

3. 奖惩管理信息表 (rpmanage), 记录获奖/惩罚的人员的情况;

4. 人事调动管理信息表(translate), 记录人员的调动信息。

5. 培训管理信息表(train), 记录人员的培训管理情况。

6. 职称评定信息表(title), 记录人员的职称评定信息。

### 4.2.3 数据库的测试连接

信息发布系统的数据库使用的是 SQL Server2000,在建立数据库之前,首先要对数据库进行连接测试。在 JSP 中进行数据库的连接测试编码如下所示:

jsp 连接 SQL Server2000 数据库 , testsqlserver.jsp 如下:

```
<%@ page contentType="text/html;charset=gb2312"%>
<%@ page import="java.sql.*"%>
<html>
<body>
<%Class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver").newInstance();
String url="jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=pubs";
//pubs 为你的数据库
String user="sa";
String password="";
```

```
Connection conn= DriverManager.getConnection(url,user,password);
Statement
stmt=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.
CONCUR_UPDATABLE);
String sql="select * from test";
ResultSet rs=stmt.executeQuery(sql);
while(rs.next()) {%>
您的第一个字段内容为: <%=rs.getString(1)%>
您的第二个字段内容为: <%=rs.getString(2)%>
<%}%>
<%out.print("数据库操作成功, 恭喜你");%>
<%rs.close();
stmt.close();
conn.close();
%>
</body>
</html>
```

#### 4.2.4 注册数据库驱动程序

在一个 JDBC 的驱动程序被用来建立数据库连接之前, 必须向数据库驱动程序管理器注册该驱动程序。管理器的主要职责是保证所有的驱动程序对象可被 JDBC 的用户程序访问。当一个 JDBC 驱动程序被载入时, 它将自动地在驱动程序管理器上注册。

可以调用 `Class.forName()` 方法载入一个 JDBC 驱动程序:

```
String strDBDriver="sun.jdbc.odbc.JdbcOdbcDriver";
```

```
Class.forName( "strDBDriver " );
```

Class.forName()是 Java 的 Class 类的静态方法,它使 Java 虚拟机动态的寻找, 载入并连接指定的类。

一个 JDBC 的驱动程序就是一个可以通过 classpath 找到一个 Java 类。这里载入的 JDBC\_ODBC 桥驱动程序就在 sun.jdbc.odbc 包中。

#### 4.2.5 建立数据库连接

载入一个正确的驱动程序后, 就可以用它来建立与数据库的连接了。一个 JDBC 的数据库连接是用数据库 URL 来标记的。连接标记将告诉驱动程序管理器使用哪个驱动程序和连接哪个数据库。数据库 URL 的基本形式:

```
jdbc:<JDBC 驱动程序名>: <数据源标记>
```

URL 的第一部分表示使用 JDBC 建立数据库连接。第 2 部分表示所使用的 JDBC 驱动程序名或网络协议名。第 3 部分是数据源标记, 它映射所需连接的数据库。例如数据库的数据源标记是 firm 则同 JDBC-ODBC 桥连接该数据库所用的数据库 URL 为 jdbc:odbc:firm.

可以通过 DriverManager 类的静态方法 getConnection()来建立一个数据库连接:

```
String strDBUrl="jdbc:odbc:firm";
conn=DriverManager.getConnection(strDBUrl,"sa","");
```

一旦建立数据库连接, 就可以打开该目标数据库, 并通过该连接向目标数据库 SQL 指令。SQL 指令通常会执行查询, 插入, 更新和删除等数据库操作。要执行 SQL 指令, 必须通过 Connection 对象的 createStatement()方法来创建一个 STMT 对象。STMT 对象提供了许多方法来执行不同的数据库操作。

## 4.3 模块功能的实现

### 4.3.1 登陆模块

实现该功能的文件有：

Head.jsp, ChkHead.jsp, 这两个文件实现了登录模块的功能, 其中 Head.jsp 通过 request.getParameter () 方法来获取输入的用户名和用户密的值, 并交由 ChkHead.jsp 页面处理, ChkHead.jsp 通过一系列的 JSP 方法, 函数 (如 response.sendRedirect () 方法,) 组成业务逻辑来判断所获取的数据是否合法, 进而决定是否允许来访者登录系统。

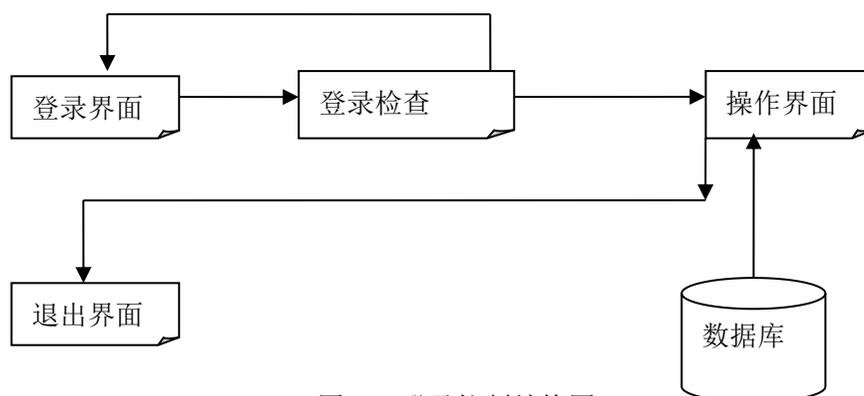


图 4.1 登录控制结构图

### 4.3.2 人员档案模块

人员档案是人事管理系统的基本数据资料,是员工管理系统的基本功能,其提供的功能有:添加人员档案、修改人员档案、删除人员档案和查询人员档案。其模块结构如图 4-2 所示。

本模块包含的 JSP 文档及其功能如图 4.2 的结构图所示, 其中 Personer.jsp 功能与 head.jsp 类似,主要用于显示本模块网页的功能链接。

AddPersoner1.jsp 主要用于提供用户输入人员档案信息的界



详细信息以及其余模块的信息，如奖惩情况和调动情况。

此外是关于 javabean 的，Employee.java 定义了类 Employee，用来封装与人员档案相关的方法，如增加、修改、删除、查询等，其定义的关键方法有：Add()用于添加人员档案，Modify()用于修改人员档案，Delete(String \_EmployeeID)用于删除人员档案，ListAll()用于列出所有人员档案，Search()用于查询人员档案。

### 4.3.3 奖惩管理模块

奖惩管理模块主要用于记录企业员工的奖惩情况，其主要提供的功能有添加奖惩记录和修改奖惩记录。

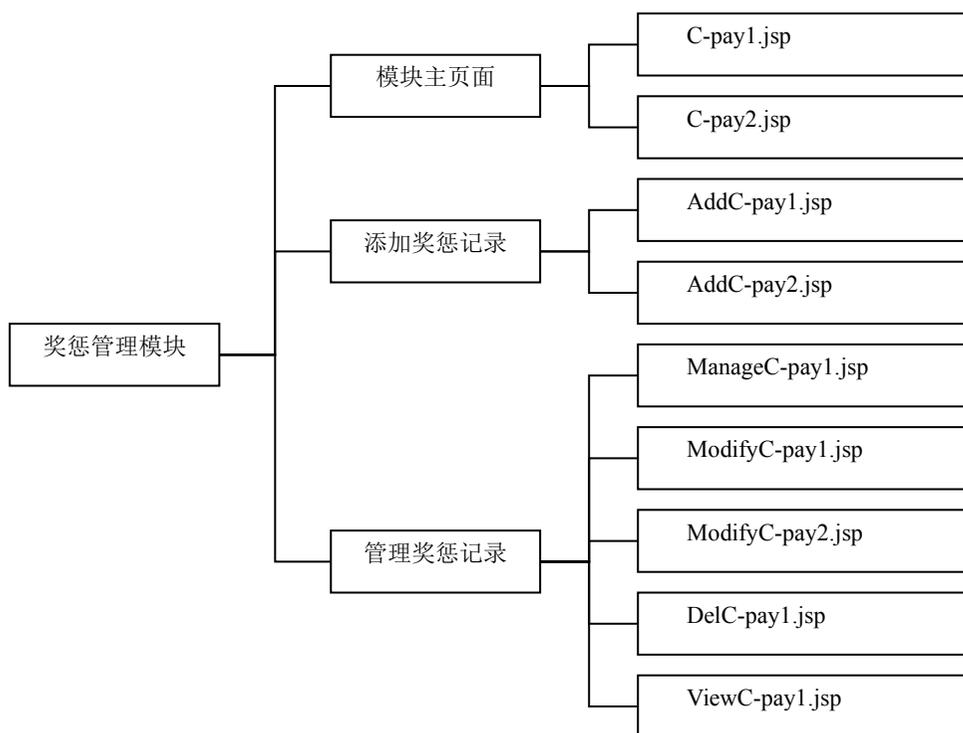


图 4.3 奖惩管理模块结构图

本模块主要包含页面关系如图 4.3 的结构图所示，其中 C-pay1.jsp 功能与 head.jsp 作用相似，C-pay2.jsp 作用是提供了本模块的功能链接，如添加

奖惩记录，管理奖惩记录和退出系统。

添加奖惩记录主要由 AddC-pay1.jsp 和 AddC-pay2.jsp 两个文件组成，其中前者提供了添加记录的界面，后者用于调用 JavaBean 中的 Reward 中的 Add（）方法，向数据库传递填写的信息，更新数据库。

管理奖惩记录由 ManageC-pay1.jsp、ManageC-pay2.jsp、ModifyC-pay1.jsp、ModifyRewardStep2.jsp、ViewC-pay1.jsp 和 DelC-pay1.jsp 五个文件组成，其中 ManageC-pay1.jsp 用于显示管理奖惩记录界面，当单击图中的修改链接后，则进入 ModifyC-pay1.jsp 页面，当点击修改按钮时刚进入 ModifyC-pay2.jsp 页面，用于和数据库进行交互，修改数据库中的记录，并返回修改结果。如点击删除链接，刚进入 DelC-pay1.jsp 页面，系统会提示是否确认删除操作，单击确定后则进行删除操作，并返回删除结果；如点击员工编号，则可进入 ViewC-pay1.jsp 页面，显示该员工的详细奖惩信息。

相应的对于该模块的 javabean 要说明的是：eward.java 中定义了类 Reward，用来封装与奖惩管理有关的功能方法，比如添加，修改，删除，查看奖惩记录等，其定义的各方法及其功能描述如下：FindByNO（）该方法主要用于实现根据记录号查找奖惩信息，用于整体查询；Add（）主要用于实现奖惩记录的添加操作；Modify（）主要实现奖惩信息的修改操作；Delete（）实现删除操作；SearchByID（）通过员工编号进行查询操作，主要用于一般查询；ListAll（）主要用于列出查询结果。

#### 4.3.4 人员调动管理模块

人员调动管理模块主要用于记录企业员工的人员调动情况，其主要提供的功能有添加人员调动记录和修改人员调动记录。人员调动管理模块的结构如图 4.4 所示。

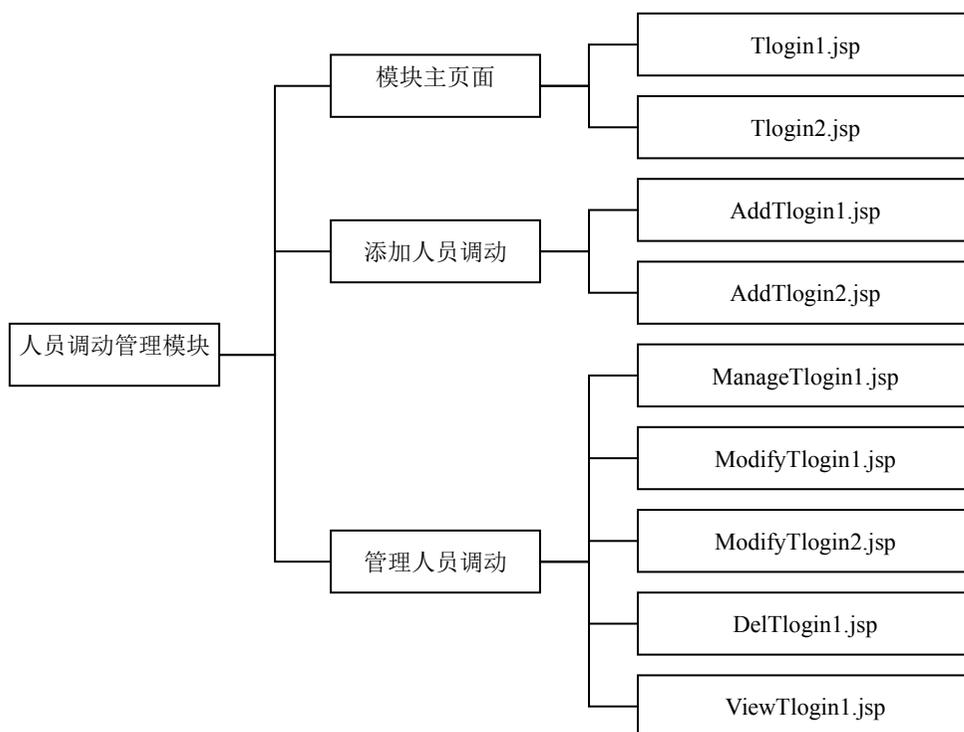


图 4.4 人员调动管理模块结构图

本模块主要包含页面关系如图 4.4 的结构图所示,其中 Tlogin1.jsp 功能与 head.jsp 作用相似, Tlogin2.jsp 作用是提供了本模块的功能链接,如添加人员调动记录,管理人员调动记录和退出系统。

添加人员调动记录主要 AddTlogin1.jsp、AddTlogin2.jsp 两个文件组成,其中前者提供了添加记录的界面,后者用于调用 JavaBean 中的 Translate 中的 Add () 方法,向数据库传递填写的信息,更新数据库。

管理人员调动记录主要由 ManageTlogin1.jsp、ManageTlogin2.jsp、ModifyTlogin1.jsp、ModifyTlogin2.jsp、ViewTTlogin1.jsp 和 DelTlogin1.jsp 五个文件组成,其中 ManageTlogin1.jsp 用于显示管理人员调动记录界面,当单击图中的修改链接后,则进入 ModifyTlogin1.jsp 页面,当点击修改按钮时刚进入 ModifyTlogin2.jsp 页面,用于和数据库进行交互,修改数据库中

的记录，并返回修改结果。如点击删除链接，刚进入 DelTlogin1.jsp 页面，系统会提示是否确认删除操作，单击确定后则进行删除操作，并返回删除结果；如点击员工编号，则可进入 ViewTlogin1.jsp 页面，显示该员工的详细人员调动信息。

相应的说明一下 javabean 的作用，Translate.java 中定义了类 Translate，用来封装与人员调动管理有关的功能方法，比如添加，修改，删除，查看人员调动记录等，其定义的各方法及其功能描述如下：FindByNO（）该方法主要用于实现根据记录号查找人员调动信息，用于整体查询；Add（）主要用于实现人员调动记录的添加操作；Modify（）主要实现人员调动信息的修改操作；Delete（）实现删除操作；SearchByID（）通过员工编号进行查询操作，主要用于一般查询；ListAll（）主要用于列出查询结果。

### 4.3.5 培训管理模块

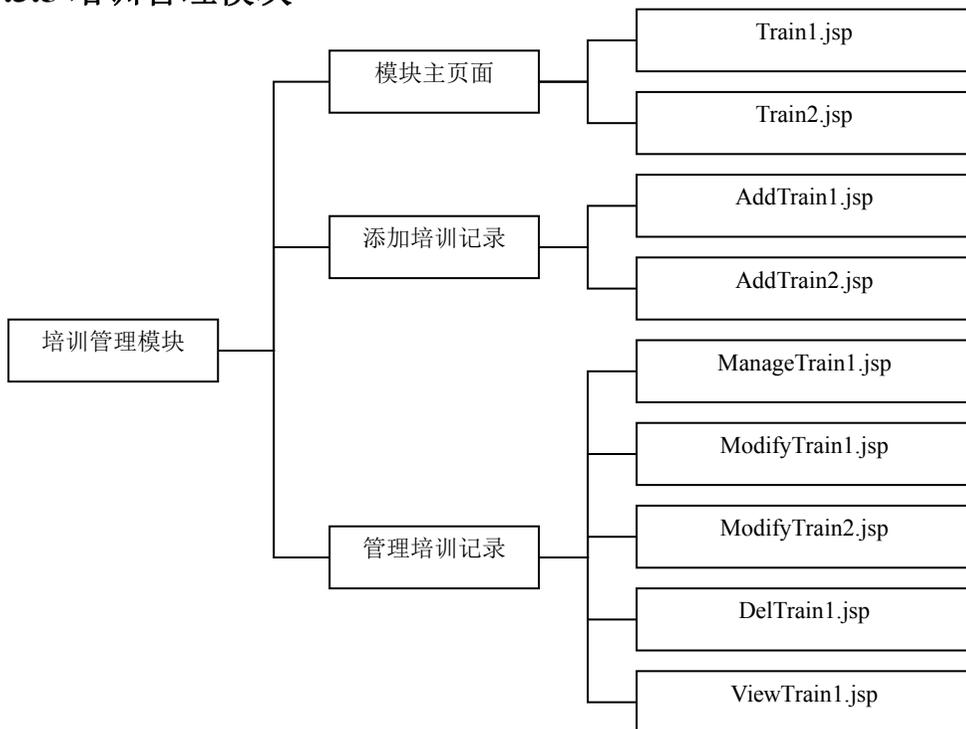


图 4.5 人员调动管理模块结构图

培训管理模块主要用于记录企业员工培训管理情况，其主要提供的功能有添加人员培训记录和修改人员培训记录。人员培训管理模块的结构如图 4.5 所示。

本模块主要包含如图 4.5 所示结构图，其中 Train1.jsp 功能与 head.jsp 作用相似，Train2.jsp 作用是提供了本模块的功能链接，如添加培训记录，管理培训记录和退出系统。

添加培训记录主要由 AddTrain1.jsp 和 AddTrain2.jsp 两个文件组成，其中前者提供了添加记录的界面，后者用于调用 JavaBean 中的 Reward 中的 Add（）方法，向数据库传递填写的信息，更新数据库。

管理培训记录由 ManageTrain1.jsp、ManageTrain2.jsp、ModifyTrain1.jsp、ModifyTrain2.jsp、ViewTrain1.jsp 和 DelTrain1.jsp 五个文件组成，其中 ManageTrain1.jsp 用于显示管理培训记录界面，当单击图中的修改链接后，则进入 ModifyTrain1.jsp 页面，当点击修改按钮时刚进入 ModifyTrain2.jsp 页面，用于和数据库进行交互，修改数据库中的记录，并返回修改结果。如点击删除链接，刚进入 DelTrain1.jsp 页面，系统会提示是否确认删除操作，单击确定后则进行删除操作，并返回删除结果；如点击员工编号，则可进入 ViewTrain1.jsp 页面，显示该员工的详细奖惩信息。

相应的对于该模块的 javabean 要说明的是：Train.java 中定义了类 Reward，用来封装与培训管理有关的功能方法，比如添加，修改，删除，查看培训记录等。

### 4.3.6 职称评定管理模块

职称评定管理模块主要用于记录企业员工职称评定的管理情况，其主要提供的功能有添加人员职称记录和修改人员职称记录。职称评定管理模块的结构如图 4.6 所示。



单击确定后则进行删除操作，并返回删除结果；如点击员工编号，则可进入 ViewTitle1.jsp 页面，显示该员工的详细职称评定信息。

相应的说明一下 javabean 的作用，Title.java 中定义了类 Title，用来封装与职称评定有关的功能方法，比如添加，修改，删除，查看人员调动记录等。

## 5 系统演示

待系统的文件目录部署好后，启动 Tomcat，在地址栏输入 `http://localhost:8080/hrms/login.jsp` 进入登陆界面如图 5.1 所示。



图 5.1 登陆界面如图

输入正确的用户名，密码后就会进入系统的主页面，如图 5.2 所示。



图 5.2 主页面

该页面相当于一个操作平台,你可以选择要进行的操作，并进入相关的

操作界面，假如选择了培训管理这一项就会进入如图 5.3 所示页面。



图 5.3 人员档案页面

可以通过点击按钮来完成相应的操作，比如点了添加人员按钮就会出现图 5.4 页面。此时就可以在此页面上添加培训纪录的档案。



图 5.4 管理培训纪录页面

本系统的其他功能就不演示了。相信本系统会给您的事业提供帮助。

## 6 系统测试与维护

### 6.1 系统测试

任何软件在编写之后都会有错误的，所以说需要对软件程序进行优化与调试。测试任何产品都有两种方法：如果已经知道了产品应该具有的功能，可以通过测试来检验产品的内部动作是否按照规格说明书的规定正常进行。前一个方法称为黑盒测试，后一个方法称为白盒测试。我在模块测试的时候采用的是白盒测试，在系统测试时采用的是黑盒测试。

#### 6.1.1 单元测试

单元测试集中的检验软件设计的最小单元----模块。正式测试之前必须先通过编译程序检查并且改正所有语法错误，然后用详细设计描述作指南，对重要的招待通路进行测试，以便发现模块内部的错误。单元测试可以使用白盒测试法，而且对多个模块的测试可以并行地进行。

在测试过程中出现了不少问题，而问题主要是集中在数据类型不兼容和输入/输出错误的检查和处理两方面。对其进行了大量的测试，并得出了正确的结果。至此在测试中的一些错误都已经解决，并通过了其后的测试。

#### 6.1.2 集成测试

集成测试是组装软件的系统技术。由于我已经对各个模块进行了细致的测试，所以说在集成测试阶段我采用的是黑盒测试，从总体上检查了软件的功能，因为在之前的模块调试中解决了大部分错误，所以在此测试中只发现了几个模块间的错误，并且也已经将其解决。

### 6.1.3 测试用例

为了确保本系统能够正常运行，需要在发布之后做一次较全面的测试。现将具体操作及过程举例说明如下：

1. 测试用户登陆是否成功。

打开人事管理系统，在“用户名”里填入 1，“密码”框里我们填入 1 作为用户登录密码。填写完成点击“确定”按钮，将会出现操作程序页面，即该用户已经登陆成功了。

2. 测试其他用户是否能够登录。

打开人事管理系统登录页面，输入任意密码，点击“登录”按钮。将出现密码出错提示页面。然后点“确定”返回人事管理系统登录页面。

3. 测试添加功能是否成功。

在登录成功之后，即可进入相应的管理页面，点击相应的添加按钮即出现编辑页面，即可以添加、修改、删除员工的个人信息资料。

4. 测试查询功能是否成功。

在登录成功之后，点击相应查询按钮，即可出现相应的结果页面，页面内显示查询出来的员工资料内容，表示测试成功。

5. 用户退出系统的测试。

在管理操作已全部完成，需要退出程序的时候，在系统页面选择退出的选项，点击之后便可退出。

6. 测试密码更改。

在系统用户管理页面中，打开修改密码选项，就会出现对应的修改页面进行修改密码。

经过以上各项的测试，证明本系统完全可以正常运行，至此测试成功。



## 7 结束语

本次设计是对大学四年来所学知识的综合运用，是对学习成果的检验。通过这次毕业设计使我对 JSP 和 Tomcat 服务器以及 SQL Server 数据库得到了进一步的了解，同时对 Tomcat 和 SQL Server 数据库的连接问题进行了深入研究，通过网络提供的资料解决了数据库相关的连接问题，在这次毕业设计发现数据库方面的知识仍有很多不足，在今后的学习生活里会多多学习该方面技术弥补自己的缺陷。由于时间比较仓促，加上本人软件设计经验的不足。因此，在分析问题、解决问题时显得不够严密、完善，还需要在以后的工作中不断地改进和完善。

在设计过程中我认识到理论与实际相结合的重要性及理论与实际之间的区别，真正体会到学以致用乐趣。通过理论作指导，通过实践来体会、验证理论，这样既加深了对理论知识的认识，又从实践活动中不断地充实自己。

在这次设计期间我学到了很多的东西，这些知识多数是书本上学不到的，其中包括了我在求知路上的收获快乐和失败痛苦，各种滋味相交融，但是最后的感觉是收获的喜悦。通过本次设计提高了实践能力，增强了专业技术的水平，为今后从事本专业的工作打下了一定的基础。

## 参考文献

- [1]萨师煊、王珊编著.《数据库系统概论》（第三版）高等教育出版社.
- [2]马文刚、谢乐建、管知时编著.《JSP 实用编程实例集锦》清华大学出版社.
- [3]布霍恩·赖特著，赵明昌译.《JSP 数据库编程指南》北京希望电子出版社.
- [4] Herbert 著，张玉清等译. Java2 参考大全. 北京清华大学出版社.
- [5] 微软公司著，高国连，李国华译. SQL Server 2000 使用 Transact-SQL 进行数据库查询. 北京：希望电子出版社.
- [6] 汪晓平等编著. 精通 Java 网络编程. 北京：清华大学出版社.
- [7] 林上杰，林康司著. JSP2.0 技术手册. 西安：电子工业出版社.
- [8] 微软公司著，高国连，李国华. SQL Server 2000 使用 Transact-SQL 进行数据库查.
- [9] 汪孝宜等著，JSP 数据库开发实例精粹. 北京：电子工业出版社.
- [10] 微软公司著，高国连，李国华译. SQL Server 2000 使用 Transact-SQL 进行数据库查.

## 致 谢

首先要感谢我的指导老师徐老师，从开题到设计结束一直耐心给予指导，并在设计思路给予了很大的帮助；其次老师对我程序设计的可行性、规模是否适当等问题上提出了很多的宝贵的建议；最后还要感谢帮助过的同学们，有很多问题是通过请教他们才得以解决的，使我学习到了很多新知识；还要感谢所有教导过我的老师们，感谢你们让我具有一定的理论知识与实际经验，这样才使论文能够以很好的理论知识作后盾，使论文能够顺利的完成，衷心的感谢你们！

## 附 录

### A 主要源程序

数据库连接：

#### 1. Connect.java

```
package hrms.database;
import java.sql.*;
import hrms.database. ConnectionManager;

public class Connect {
private Connection conn = null;
private Statement stmt = null;
private PreparedStatement prepstmt = null;
private ConnectionManager dcm = null;
void init() {
dcm = ConnectionManager.getInstance();
conn = dcm.getConnection("sqlserver");
}
public Connect() throws Exception {
init();
stmt = conn.createStatement();
}
public Connect(int resultSetType, int resultSetConcurrency)
throws Exception {
init();
```

```
        stmt = conn.createStatement(resultSetType, resultSetConcurrency);
    }
    public Connect(String sql) throws Exception {
        init();
        this.prepareStatement(sql);
    }
    public Connect(String sql, int resultSetType, int resultSetConcurrency)
        throws Exception {
        init();
        this.prepareStatement(sql, resultSetType, resultSetConcurrency);
    }
    public Connection getConnection() {
        return conn;
    }
    public void prepareStatement(String sql) throws SQLException {
        prepstmt = conn.prepareStatement(sql);
    }
    public void prepareStatement(String sql, int resultSetType,
        int resultSetConcurrency) throws SQLException {
        prepstmt = conn.prepareStatement(sql, resultSetType,
            resultSetConcurrency);
    }
    public void setString(int index, String value) throws SQLException {
        prepstmt.setString(index, value);
    }
    public void setInt(int index, int value) throws SQLException {
```

```

        pstmt.setInt(index, value);
    }
    public void setBoolean(int index, boolean value) throws SQLException {
        pstmt.setBoolean(index, value);
    }
    public void setDate(int index, Date value) throws SQLException {
        pstmt.setDate(index, value);
    }
    public void setTimestamp(int index, Timestamp value) throws
SQLException {
        pstmt.setTimestamp(index, value);
    }
    public void setLong(int index, long value) throws SQLException {
        pstmt.setLong(index, value);
    }
    public void setFloat(int index, float value) throws SQLException {
        pstmt.setFloat(index, value);
    }
    public void setBytes(int index, byte[] value) throws SQLException {
        pstmt.setBytes(index, value);
    }
    public void clearParameters() throws SQLException {
        pstmt.clearParameters();
        pstmt = null;
    }
    public PreparedStatement getPreparedStatement() {

```

```
        return pstmt;
    }
    public Statement getStatement() {
        return stmt;
    }
    public ResultSet executeQuery(String sql) throws SQLException {
        if (stmt != null) {
            return stmt.executeQuery(sql);
        } else {
            return null;
        }
    }
    public ResultSet executeQuery() throws SQLException {
        if (pstmt != null) {
            return pstmt.executeQuery();
        } else {
            return null;
        }
    }
    public void executeUpdate(String sql) throws SQLException {
        if (stmt != null) {
            stmt.executeUpdate(sql);
        }
    }
    public void executeUpdate() throws SQLException {
        if (pstmt != null) {
            pstmt.executeUpdate();
        }
    }
    public void close() throws Exception {
```

```

    if (stmt != null) {
        stmt.close();
        stmt = null;
    }
    if (prepstmt != null) {
        prepstmt.close();
        prepstmt = null;
    }
    if (conn != null) {
        dcm.freeConnection("sqlserver", conn);
    }
}
}

```

## 二 ConnectionManager.java

```

package hrms.database;

import java.io.*;
import java.sql.*;
import java.util.*;
import java.util.Date;

public class ConnectionManager {
    static private ConnectionManager instance; // 唯一实例
    static private int clients;
    private Vector drivers = new Vector();
    private Hashtable pools = new Hashtable();
    private void log(String msg) {

```

```
        System.out.println(new Date() + ": " + msg);
    }
    static synchronized public ConnectionManager getInstance() {
        if(instance == null) {
            instance = new ConnectionManager();
        }
        clients++;
        return instance;
    }
    Private ConnectionManager() {
        init();
    }
    public void freeConnection(String name, Connection con) {
        DBConnectionPool pool = (ConnectionPool) pools.get(name);
        if(pool != null) {
            pool.freeConnection(con);
        }
    }
    public Connection getConnection(String name) {
        DBConnectionPool pool = (ConnectionPool) pools.get(name);
        if(pool != null) {
            return pool.getConnection();
        }
        return null;
    }
    public Connection getConnection(String name, long time) {
```

```

ConnectionPool pool = (ConnectionPool) pools.get(name);
...
if (pool != null) {
...
    return pool.getConnection(time);
...
}
...
return null;
...
}
...
public synchronized void release() {
...
    // 等待直到最后一个客户程序调用
    if (--clients != 0) {
...
        return;
...
    }
...
    Enumeration allPools = pools.elements();
...
    while (allPools.hasMoreElements()) {
...
        ConnectionPool pool = (ConnectionPool) allPools.nextElement();
...
        pool.release();
...
    }
...
    Enumeration allDrivers = drivers.elements();
...
    while (allDrivers.hasMoreElements()) {
...
        Driver driver = (Driver) allDrivers.nextElement();
...
        try {
...
            DriverManager.deregisterDriver(driver);
...
            log("撤销 JDBC 驱动程序 " + driver.getClass().getName() + "
的注册");
...
        } catch (SQLException e) {
...
            log("无法撤销下列 JDBC 驱动程序的注册: " +
driver.getClass().getName());
...
        }
...
    }
...
}

```

```
    }  
  }  
}  
private void createPools(Properties props) {  
    Enumeration propNameNames = props.propertyNames();  
    while (propNameNames.hasMoreElements()) {  
        String name = (String) propNameNames.nextElement();  
        if (name.endsWith(".url"))  
        {  
            String poolName = "sqlserver";  
            String url = props.getProperty(poolName + ".url");  
            if (url == null)  
            {  
                log("没有为连接池" + poolName + "指定 URL");  
                continue;  
            }  
            String user = props.getProperty(poolName + ".user");  
            String password = props.getProperty(poolName +  
".password");  
  
            String maxconn = props.getProperty(poolName + ".maxconn",  
"0");  
  
            int max;  
            try {  
                max = Integer.valueOf(maxconn).intValue();  
            } catch (NumberFormatException e) {
```

```

        log("错误的最大连接数限制: " + maxconn + ".连接池: "
+ poolName);
...
        max = 0;
...
    }
...
    DBConnectionPool pool = new DBConnectionPool(poolName,
url,
        user, password, max);
...
    pools.put(poolName, pool);
...
    log("成功创建连接池" + poolName);
...
}
...
}
...
}
...
private void init() {
...
    InputStream is = getClass().getResourceAsStream("/hrms.txt");
...
    Properties dbProps = new Properties();
...
    try {
...
        dbProps.load(is);
...
    } catch (Exception e) {
...
        log("不能读取属性文件.请确保 db.properties 在 CLASSPATH 指
定的路径中");
...
        return;
...
    }
...
    loadDrivers(dbProps);
...
    createPools(dbProps);
...
}
...
private void loadDrivers(Properties props) {
...
    String driverClasses = props.getProperty("driver");

```

```
StringTokenizer st = new StringTokenizer(driverClasses);
while (st.hasMoreElements()) {
    String driverClassName = st.nextToken().trim();
    try {
        Driver driver = (Driver) Class.forName(driverClassName)
            .newInstance();
        DriverManager.registerDriver(driver);
        drivers.addElement(driver);
        log("成功注册 JDBC 驱动程序" + driverClassName);
    } catch (Exception e) {
        log("无法注册 JDBC 驱动程序: " + driverClassName + ", 错误: " + e);
    }
}

class DBConnectionPool {
    private int userCount;
    private Vector freeConnections = new Vector();
    private int maxConn;
    private String name;
    private String password;
    private String URL;
    private String user;
    public DBConnectionPool(String name, String URL, String user,
        String password, int maxConn) {
        this.name = name;
```

```

this.URL = URL;
...
this.user = user;
...
this.password = password;
...
this.maxConn = maxConn;
...
}
...
public synchronized void freeConnection(Connection con) {
...
// 将指定连接加入到向量末尾
...
freeConnections.addElement(con);
...
userCount--;
...
notifyAll();
...
}
...
public synchronized Connection getConnection() {
...
Connection con = null;
...
if (freeConnections.size() > 0) {
...
// 获取向量中第一个可用连接
...
con = (Connection) freeConnections.firstElement();
...
freeConnections.removeElementAt(0);
...
try {
...
if (con.isClosed()) {
...
log("从连接池" + name + "删除一个无效连接");
...
// 递归调用自己,尝试再次获取可用连接
...
con = getConnection();
...
}
...
} catch (SQLException e) {
...
log("从连接池" + name + "删除一个无效连接");
...
// 递归调用自己,尝试再次获取可用连接

```

```
        con = getConnection();
    }
} else if (maxConn == 0 || userCount < maxConn) {
    con = newConnection();
}
if (con != null) {
    userCount++;
}
return con;
}

public synchronized Connection getConnection(long timeout) {
    long startTime = new Date().getTime();
    Connection con;
    while ((con = getConnection()) == null) {
        try {
            wait(timeout);
        } catch (InterruptedException e) {
        }
        if ((new Date().getTime() - startTime) >= timeout) {
            // wait()返回的原因是超时
            return null;
        }
    }
    return con;
}

public synchronized void release() {
```

```

Enumeration allConnections = freeConnections.elements();
while (allConnections.hasMoreElements()) {
    ...
    Connection con = (Connection) allConnections.nextElement();
    ...
    try {
        ...
        con.close();
        ...
        log("关闭连接池" + name + "中的一个连接");
    } catch (SQLException e) {
        ...
        log("无法关闭连接池" + name + "中的连接");
    }
    ...
}
freeConnections.removeAllElements();
}

private Connection newConnection() {
    Connection con = null;
    try {
        if (user == null) {
            con = DriverManager.getConnection(URL);
        } else {
            con = DriverManager.getConnection(URL, user,
password);
        }
        log("连接池" + name + "创建一个新的连接");
    } catch (SQLException e) {
        log("无法创建下列 URL 的连接: " + URL);
        return null;
    }
}

```

```
        return con;
    }
}
}
```

设置连接属性的 hrms.txt 内容

```
driver=com.microsoft.jdbc.sqlserver.SQLServerDriver
sqlserver.url=jdbc:microsoft:sqlserver://localhost:1433;databaseName=hrms
sqlserver.user=sa
sqlserver.password=""
sqlserver.maxconn=2000
```

数据库操作

### **Translate.java**

```
package hrms.action;
import hrms.database.DBConnect;
import hrms.util.*;
import java.sql.*;
import java.util.*;
import hrms.util.*;

public class Translate {
    public String EmployeeID,
        PriorBranch,NextBranch,PriorDuty,NextDuty,HandleName,Remark;
    public java.sql.Date TranslateDate;
    public int TranslateNO;
    public Translate() {
    };
    public String getEmployeeID() {
```



```
public void setNextDuty(String s) {
    this.NextDuty = s;
}
public String getHandleName() {
    return HandleName;
}
public void setHandleName(String s) {
    this.HandleName = s;
}
public String getRemark() {
    return Remark;
}
public void setRemark(String s) {
    this.Remark = s;
}
public void setTranslateNO(int i) {
    this.TranslateNO = i;
}
public String getTranslateNO() {
    return DataConvert.IntToStr(TranslateNO);
}
public String getTranslateDate() {
    if (this.TranslateDate != null)
        return DataConvert.sqlDateToStr(TranslateDate);
    else
        return "";
```

```

}
...
public void setTranslateDate(String str) {
...
    if (str != null)
...
        this.TranslateDate = DataConvert.StrToSqlDate(str);
...
    else
...
        this.TranslateDate = null;
...
}
...
public static Translate FindByNO(String _TranslateNO) {
...
    DBConnect dbc = null;
...
    Translate translate = new Translate();
...
    try {
...
        dbc = new DBConnect();
...
        dbc.prepareStatement("SELECT * FROM translate WHERE
TranslateNO=?");
...
        dbc.setString(1, _TranslateNO);
...
        ResultSet rs = dbc.executeQuery();
...
        if (rs.next()) {
...
            translate.setTranslateNO(rs.getInt("TranslateNO"));
...
            translate.setEmployeeID(rs.getString("EmployeeID"));
...
            translate.setTranslateDate(rs.getString("TranslateDate"));
...
            translate.setPriorBranch(rs.getString("PriorBranch"));
...
            translate.setNextBranch(rs.getString("NextBranch"));
...
            translate.setPriorDuty(rs.getString("PriorDuty"));
...
            translate.setNextDuty(rs.getString("NextDuty"));
...
            translate.setHandleName(rs.getString("HandleName"));
...
            translate.setRemark(rs.getString("Remark"));
...

```

```
        } else {
            translate = null;
        }
    } catch (Exception e) {
        System.err.println(e);
    } finally {
        try {
            dbc.close();
        } catch (Exception e) {
            e.printStackTrace();
            translate = null;
        }
    }
    return translate;
}

public static String Add(String _EmployeeID, String _TranslateDate,
    String _PriorBranch, String _NextBranch, String _PriorDuty,
    String _NextDuty,String _HandleName,String _Remark) {
    DBConnect dbc = null;
    try {
        dbc = new DBConnect();
        dbc.prepareStatement("INSERT INTO Translate
EmployeeID,TranslateDate,PriorBranch,NextBranch,"
            + "PriorDuty,NextDuty,HandleName,Remark)
VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
        dbc.setString(1, _EmployeeID);
```

```

dbc.setDate(2, DataConvert.StrToSqlDate(_TranslateDate));
...
dbc.setString(3, _PriorBranch);
...
dbc.setString(4, _NextBranch);
...
dbc.setString(5, _PriorDuty);
...
dbc.setString(6, _NextDuty);
...
dbc.setString(7, _HandleName);
...
dbc.setString(8, _Remark);
...
dbc.executeUpdate();
...
dbc.close();
...
return "1";
...
} catch (Exception e) {
...
    System.err.println(e);
...
    return e.toString();
...
}
...
}

public static String Modify(String _TranslateNO,String _EmployeeID,
String _TranslateDate,
订 String _PriorBranch, String _NextBranch, String _PriorDuty,
... String _NextDuty,String _HandleName,String _Remark) {
...
    DBConnect dbc = null;
...
    try {
...
        dbc = new DBConnect();
...
        dbc.prepareStatement("UPDATE translate SET
... EmployeeID=?,TranslateDate=?,PriorBranch=?,NextBranch=?,
...
... +
... "PriorDuty=?,NextDuty=?,HandleName=?,Remark=? WHERE
... TranslateNO=?");

```

```
        dbc.setString(1, _EmployeeID);
        dbc.setDate(2, DataConvert.StrToSqlDate(_TranslateDate));
        dbc.setString(3, _PriorBranch);
        dbc.setString(4, _NextBranch);
        dbc.setString(5, _PriorDuty);
        dbc.setString(6, _NextDuty);
        dbc.setString(7, _HandleName);
        dbc.setString(8, _Remark);
        dbc.setInt(9, DataConvert.StrToInt(_TranslateNO));
        dbc.executeUpdate();
        dbc.close();
        return "1";
    } catch (Exception e) {
        e.printStackTrace();
        return e.toString();
    }
}

public static String Delete(String _TranslateNO) {
    try {
        DBConnect dbc = new DBConnect();
        dbc.prepareStatement("DELETE FROM translate WHERE
TranslateNO=?");
        dbc.setInt(1, DataConvert.StrToInt(_TranslateNO));
        dbc.executeUpdate();
        dbc.close();
        return "1";
    }
```

```

    } catch (Exception e) {
        e.printStackTrace();
        return e.toString();
    }
}

public static Vector SearchByID(String _EmployeeID) {
    DBConnect dbc = null;
    Vector TranslateVector = new Vector();
    try {
        dbc = new DBConnect();
        dbc
            .prepareStatement("SELECT * FROM translate WHERE
EmployeeID=? ORDER BY TranslateDate ASC");
        dbc.setString(1, _EmployeeID);
        ResultSet rs = dbc.executeQuery();
        while (rs.next()) {
            Translate translate = new Translate();
            translate.setTranslateNO(rs.getInt("TranslateNO"));
            translate.setEmployeeID(rs.getString("EmployeeID"));
            translate.setTranslateDate(rs.getString("TranslateDate"));
            translate.setPriorBranch(rs.getString("PriorBranch"));
            translate.setNextBranch(rs.getString("NextBranch"));
            translate.setPriorDuty(rs.getString("PriorDuty"));
            translate.setNextDuty(rs.getString("NextDuty"));
            translate.setHandleName(rs.getString("HandleName"));
            translate.setRemark(rs.getString("Remark"));
        }
    }
}

```

```
        TranslateVector.add(translate);
    }
} catch (Exception e) {
    System.err.println(e);
} finally {
    try {
        dbc.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
return TranslateVector;
}

public static Vector ListAll() {
    DBConnect dbc = null;
    Vector TranslateVector = new Vector();

    String strSQL = "SELECT * FROM translate ORDER BY
EmployeeID,TranslateDate ASC";

    try {
        dbc = new DBConnect();
        dbc.prepareStatement(strSQL);
        ResultSet rs = dbc.executeQuery();
        while (rs.next()) {
            Translate translate = new Translate();
            translate.setTranslateNO(rs.getInt("TranslateNO"));
        }
    }
}
```

```
translate.setEmployeeID(rs.getString("EmployeeID"));
...
translate.setTranslateDate(rs.getString("TranslateDate"));
...
translate.setPriorBranch(rs.getString("PriorBranch"));
...
translate.setNextBranch(rs.getString("NextBranch"));
...
translate.setPriorDuty(rs.getString("PriorDuty"));
...
translate.setNextDuty(rs.getString("NextDuty"));
...
translate.setHandleName(rs.getString("HandleName"));
...
translate.setRemark(rs.getString("Remark"));
...
    TranslateVector.add(translate);
...
    }
...
} catch (Exception e) {
...
    System.err.println(e);
...
} finally {
...
    try {
...
        dbc.close();
...
    } catch (Exception e) {
...
        e.printStackTrace();
...
    }
...
}
...
return TranslateVector;
...
}
...
}
```

装

订

## B 用户手册

### B.1 系统功能简介

本系统主要用于对员工进行管理。利用 JSP（Java Server Pages）技术进行人事管理系统的开发，系统采用 SQL 数据库作为后台数据库，在设计上具有友好的交互界面，整个系统从符合操作简便、界面友好、灵活、实用、安全的要求出发，实现企业的员工信息管理、企业员工奖惩信息管理、员工培训管理、员工职称评定、企业员工人员调动信息管理和系统用户管理功能六个功能，包括添加、删除和修改部门信息，人员加入时人事档案的建立、人员的调转、员工信息查询等人事管理工作。

### B.2 系统的支持平台

#### 1. 硬件环境

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：PentiumIII以上；

内存：512MB。

#### 2. 软件环境

浏览器要支持 HTML 元素的表达，WEB 服务器为 Apache Tomcat 系列，开发工具为 MyEclipse。

3. 数据库环境：SQL Server 2000

4. 服务器环境：Windows XP

### B.3 安装说明

1. 安装好 JDK，在环境变量中进行对应的配置。

JAVA\_HOME=C:\j2sdk1.4.2（安装路径）

CLASSPATH=C:\j2sdk1.4.2\lib\dt.jar;C:\j2sdk1.4.2\bin;C:\j2sdk1.4.2\lib\tools.jar,;

PATH=C:\j2sdk1.4.2\bin;

2. 启动 Tomcat 的安装包。下载完后点击安装，完毕后再地址栏输入 <http://localhost:8080>（Tomcat 默认端口为 8080），或 <http://127.0.0.1:8080>。

3. 数据库是 SQL Server。

### B.4 使用说明

1. 将 hrms 整个文件夹拷贝到 Tomcat 的 webapps 目录下,启动 Tomcat。

2. 找到数据库\sql2000 目录下找到 hrms\_Data.MDF 文件到数据库中建表。

3. 存放待系统的文件目录部署好后，启动 Tomcat，在地址栏输入 <http://localhost:8080/hrms/login.jsp> 进入登陆界面。

### B.5 系统维护方法

1. 开发 java 应用出现乱码是很常见的，毕竟现在 unicode 的使用还不是很广泛，在使用 gb2312（包含了 gbk 简体，big5 繁体）的系统中要正确实现中文的 display 和数据库的存储是最基本的要求。由于浏览器中的参数传递是按字节传送,因此英文和数字不会有任何的问题,而中文汉字则由两个字节组成，因此导致了中文无法正常显示。本次实验中碰到了许多 JSP 和数据库中的中文显示问题，以及 JSP 网页的中文显示问题，本人经过上网搜索了好多种中文问题解决方案，但有些并不能解决本次实验所碰到的

问题, 经过多次测试, 现将本系统中文显示问题解决方案列出如下:

(1) 在首页将此句`<%@page contentType="text/html; charset= gb2312"%>`包含进去, 但是经过测试, 许多公用模块无法正常显示中文, 如 `Header.jsp` 和 `Footer.jsp`, 以及各模块主页也无法正常显示汉字, 如将此句包含到显示乱码的页面中, 则系统编译错误, 显示如下错误:

```
org.apache.jasper.JasperException: /Header.jsp(0,0) Page directive: can't have multiple occurrences of contentType. 所以经过多次试验, 把将此句包含进文件可以正常显示的文件则保留
```

此句, 不能正常显示的则采用方法 (2)。

(2) 在页首包含此句

`<%@page pageEncoding="GB2312"%>`, 经过测试, 使用此句后, 无法正常显示的页面, 能够正常显示中文了。

(3) 针对 JSP 与数据库之间的中文转化问题, 本系统采取的方法是在每次传参数的前面都进行编码转换如: `String username = new String(request.getParameter("username").getBytes("ISO8859_1"),"GBK")`

2. 将项目发布到互联网上后, 有些用户不是点击超链接打开网站的某个页面, 而是在地址栏中直接写, 而后进入, 试图避开身份验证, 处于对安全性的考虑, 本项目在用户登陆后才能进入的页面中, 加入了如下的一行判断, 如果用户非法进入, 则自动跳转到登录界面: `<%@ include file=".." "%>` 其中 `Check.jsp` 中的代码如下:

```
<% if((String)session.getValue("IsLogin")!="true")
{
String loginmsg="loginfirst";
response.sendRedirect("../login.jsp?loginmsg="+loginmsg);
}%>
```

## C 软件光盘

### C.1 光盘的树形目录

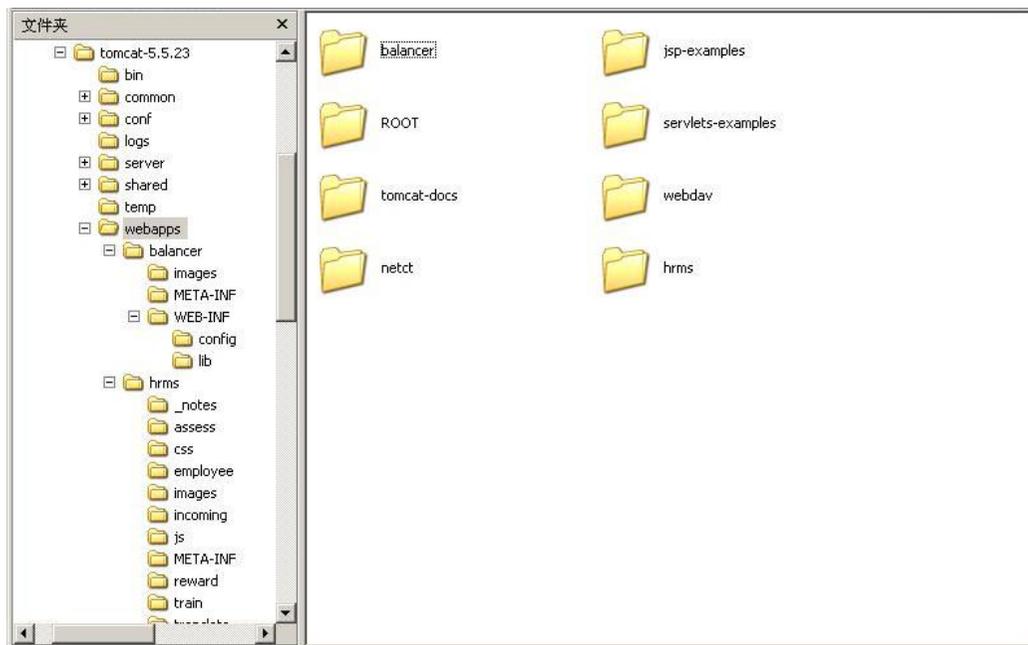


图 c.1 树形目录

### C.2 光盘文件一览表

表 c.2 光盘目录文件

文件名称	大小	实现的功能
head.jsp	4KB	实现了登录模块的功能
Personer.jsp	1KB	显示本模块网页的功能链接
Add Personer1.jsp	8KB	提供用户输入人员档案信息的界面
Add Personer2.jsp	3KB	用于提交信息连接数据库和修改数据库
Manage Personer1.jsp	11KB	用于显示人员管理的条件查询页面
ManageEmployeeStep2.jsp	3KB	用于操作数据库并显示查询结果列表
DelEmployeeStep1.jsp	4KB	实现删除操作

---

Search Personer1.jsp	1KB	主要用于查询人员档案
Search Personer2.jsp	12KB	用于显示查询结果列表
View Personer.jsp	4KB	显示员工的详细信息以及其余模块的信息
C-pay1.jsp	2KB	显示管理奖惩记录界面
Add Train1.jsp	2KB	提供了添加记录的界面
Manage Train1.jsp	3KB	用于显示管理人员培训记录界面
Modify Train2.jsp	2KB	用于和数据库进行交互, 修改数据库中的记录, 并返回修改结果
View Train1.jsp	5KB	显示该员工的详细人员培训信息
Add Title1.jsp	2KB	提供了职称评定的界面
Manage Title1.jsp	3KB	用于显示职称评定界面
Search Title1.jsp	1KB	主要用于查询职称评定的纪录

---

## D 科技译文

### Analysis of network security technology

#### Abstract

Network security information on the content of fundamental changes on China's development of national information security system and the importance of the establishment of the network with Chinese characteristics and the need for security system. On the network firewall security classification and its main technical characteristics.

Key words: network security features of firewall technology

#### 1. Summary

The 21st century will be the world's computers together through the Internet, information security and the connotations it has fundamentally changed. It not only from the general's defense has become a very ordinary precautions, but also



industries, and should see that the development of the security industry policy information security system is an important component of, or even should see its future in China electronic, information The development will play a very important role.

## 2. Firewall

A network firewall technology is used to strengthen the network between the access control and prevent external network users to illegal means to enter the internal network through an external network, access internal network resources and protect the internal network operating environment of the special networking equipment. Its network between two or more packets of data transmission links such as a certain way in accordance with the strategy to implement the security check to determine whether communications between the network was allowed to, and monitor network running.

The current firewall products are the main bastion host, packet filtering routers, Application Layer Gateway (proxy) and the circuit of the gateway, shielding host firewall, host, and other types of dual-places.

Although the firewall is to protect the networks from hacker attacks effective means, but there are also clearly insufficient: You can not prevent through the firewall other than the way the attacks, not prevent defections from within and not Jingxin users of the threat, Can not completely prevent transmission of the virus has infected software or documents, and can not guard against data-driven attacks.

Since 1986 the U.S. company Digital on the Internet to install the world's first commercial firewall systems, proposed the concept of a firewall, the firewall technology has been rapid development. Dozens of companies at home and

abroad have launched a function of different firewall products.

Firewall in a five-story network security system in the bottom, a network layer security technology areas. In this layer, the business-to-safety system put the question is: Does all of the IP can access to the internal corporate network system «If the answer is" yes ", then intranet in the network layer has not taken corresponding preventive Measures.

As the internal network and the outside public network between the first barrier, the firewall is the first by the people of the importance of network security products. Although in theory, the firewall in the bottom of network security, network responsible for the safety certification and transmission, but with the network security technology to the overall development of network applications and the constant changes in modern firewall technology has gradually outside the network layer Other safety level, not only to complete the traditional firewall filtering tasks, while for the full range of network applications provide the security services. There are also a variety of firewall products are moving towards data security and user authentication, to prevent viruses and hacking, and other direction.

According to the firewall technology used in different, we can it will be divided into four basic types: Type packet filtering, Network Address Translation - NAT, and monitoring agent-type.

### 2.1. Packet filter -

Packet filtering firewall products are the primary products, its technology is based on the network in the sub-transmission technology. Data on the network are "package" for the transmission units, data Pifengecheng to a certain size data packets, each packet will contain some of the specific information, such as data

source address, destination address, TCP / UDP source Port and other ports and objectives. Firewall by reading the address in the packet of information to judge the "package" is from the trusted security site

Once found from dangerous sites data packets, firewall, these data will be excluded. Administrators can also determine the actual situation flexible formulation of the rules.

Packet filtering technology is simple and practical advantages, to achieve lower cost, relatively simple in the application of environmental circumstances, can a smaller price to a certain extent, guarantee the security system.

However, packet filtering technology is also obvious defects. Packet filtering technology is a fully web-based layer of security technology, can only be based on the source of data packets, objectives and port networks, such as information judgement, based on the application layer does not recognize the malicious intrusion, such as malicious Java applets and e-mail With the virus. Experienced hackers can easily forged IP addresses, type Pianguo packet filtering firewall.

## 2.2. Network address transformation - NAT

Network Address Translation is an IP address used to convert a temporary, external, the IP address of the registered standards. It allows a private IP address of the internal network access to the Internet. It also means not allowed users to its network every machine obtain registration of IP addresses.

NAT the work process In the internal network through the security card access to external networks, will produce a map records. System will go out to the source address and source port mapping for a disguised address and port, so that the address and port camouflage through non-security card with external network connections, so foreign to hide the true address of the internal network.

In the external network through non-security visit to the internal network card, it does not know the internal network connections, but only through an open IP address and port to request a visit. OLM firewall mapping based on pre-defined rules to determine whether this visit is safe. When the rules, the firewall that is safe to visit, you can request an interview, can also request maps will be connected to different internal computer. When not in line with the rules, the firewall that the visit is unsafe and can not be accepted, the firewall will shield the external connection requests. Network Address Translation for the process is transparent to the user, does not require users to set up, users can conduct normal operations.

### 2.3. Agent -

Acting Firewall can also be called a proxy server, its security is higher than that of packet filtering products, and has already begun to develop application layer. Proxy server located in between the client and server, completely blocking the exchange of data between the two. From the client, the proxy server equivalent of a real server, and from the server, the proxy server is a real client. When clients need to use the data on the server, the first data request to the proxy server, a proxy server to server based on this request to obtain data, and then by the proxy server will transmit data to the client. Due to external and internal server system is not directly between the data channel, external malicious injury against it very difficult to internal network systems.

Agents of the advantages of firewall security is high, can target detection and application-level scanning, based on the application layer to deal with the intrusion and viruses are very effective. Its disadvantage is that the overall performance of the system have a greater influence, but the proxy server must

be possible for clients of all types of applications to set up one by one, greatly increases the complexity of the systems management.

#### 2.4. Monitoring of

Monitoring of the firewall is the next generation of products, this technology has actually gone beyond the initial definition of a firewall. Firewall to monitor the levels of the data active and real-time monitoring, analysis of these data, based on monitoring of the firewall can effectively determine levels of the illegal invasion. At the same time, the detection of a firewall products are generally also distributed detectors, these detectors placed in various application servers and other network nodes, not only to detect network from external attacks, from within Vandalism have very strong preventive role. According to an authoritative statistical agency, in response to the attacks in the network system, a considerable proportion of attacks from within the network. Therefore, the monitoring of firewalls not only goes beyond the traditional definition of a firewall, and security also exceeded the previous two generations of products.

Although the monitoring of firewall security on the outside of the packet filtering firewall and proxy server type, but because of firewall technology to monitor the realization of higher cost, easily managed, in practical in firewall products are still the second-generation agent - Products, but in some areas have already begun monitoring the use of firewalls. Based on the system cost and security technology costs considered, the user can selectively monitor the use of certain technology. In this way, to ensure that network security needs, but also to effectively control the security system of the total cost of ownership.

In fact, as the current trend of the mainstream firewall products, the majority of proxy server (also known as Application Gateway) also integrates a packet

filtering technology, this hybrid of two technologies alone is greater than the advantages. Because this product is based on the application, application gateways can provide the filters of the agreement. For example, it can filter out FTP connection in the PUT command, but through a proxy application, the application gateway can effectively prevent the leakage of internal information network.

## 浅析网络安全技术

### 摘要

文中就信息安全内涵发生的根本变化，阐述我国发展民族信息安全体系的重要性及建立有中国特色的网络安全体系的必要性。论述了网络防火墙安全技术的分类及其主要技术特征。

关键词：网络安全 防火墙 技术特征

### 1.概述

21 世纪全世界的计算机都将通过 Internet 联到一起,信息安全的内涵也发生了根本的变化。它不仅从一般性的防卫变成了一种非常普通的防范,而且还从一种专门的领域变成了无处不在。当人类步入 21 世纪这一信息社会、网络社会的时候,我国将建立起一套完整的网络安全体系,特别是从政策上和法律上建立起有中国自己特色的网络安全体系。

一个国家的信息安全体系实际上包括国家的法规和政策,以及技术与市场的发展平台。我国在构建信息防卫系统时,应着力发展自己独特的安全产品,我国要想真正解决网络安全问题,最终的办法就是通过发展民族的安全产业,带动我国网络安全技术的整体提高。

网络安全产品有以下几大特点:第一,网络安全来源于安全策略与技术的多样化,如果采用一种统一的技术和策略也就不安全了;第二,网络的安全机制与技术要不断地变化;第三,随着网络在社会个方面的延伸,

进入网络的手段也越来越多，因此，网络安全技术是一个十分复杂的系统工程。为此建立有中国特色的网络安全体系，需要国家政策和法规的支持及集团联合研究开发。安全与反安全就像矛盾的两个方面，总是不断地向上攀升，所以安全产业将来也是一个随着新技术发展而不断发展的产业。信息安全是国家发展所面临的一个重要问题。对于这个问题，我们还没有从系统的规划上去考虑它，从技术上、产业上、政策上来发展它。政府不仅应该看见信息安全的发展是我国高科技产业的一部分，而且应该看到，发展安全产业的政策是信息安全保障系统的一个重要组成部分，甚至应该看到它对我国未来电子化、信息化的发展将起到非常重要的作用。

## 2.防火墙

网络防火墙技术是一种用来加强网络之间访问控制，防止外部网络用户以非法手段通过外部网络进入内部网络，访问内部网络资源，保护内部网络操作环境的特殊网络互联设备。它对两个或多个网络之间传输的数据包如链接方式按照一定的安全策略来实施检查，以决定网络之间的通信是否被允许，并监视网络运行状态。

目前的防火墙产品主要有堡垒主机、包过滤路由器、应用层网关(代理服务)以及电路层网关、屏蔽主机防火墙、双宿主机等类型。

虽然防火墙是目前保护网络免遭黑客袭击的有效手段，但也有明显不足：无法防范通过防火墙以外的其它途径的攻击，不能防止来自内部变节者和不经心的用户们带来的威胁，也不能完全防止传送已感染病毒的软件或文件，以及无法防范数据驱动型的攻击。

自从 1986 年美国 Digital 公司在 Internet 上安装了全球第一个商用防火墙系统，提出了防火墙概念后，防火墙技术得到了飞速的发展。国内外已有数十家公司推出了功能各不相同的防火墙产品系列。

防火墙处于 5 层网络安全体系中的最底层,属于网络层安全技术范畴。在这

一层上,企业对安全系统提出的问题是:所有的 IP 是否都能访问到企业的内部网络系统?如果答案是“是”,则说明企业内部网还没有在网络层采取相应的防范措施。

作为内部网络与外部公共网络之间的第一道屏障,防火墙是最先受到人们重视的网络安全产品之一。虽然从理论上,防火墙处于网络安全的最底层,负责网络间的安全认证与传输,但随着网络安全技术的整体发展和网络应用的不断变化,现代防火墙技术已经逐步走向网络层之外的其他安全层次,不仅要完成传统防火墙的过滤任务,同时还能提供各种网络应用提供相应的安全服务。另外还有多种防火墙产品正朝着数据安全与用户认证、防止病毒与黑客侵入等方向发展。

根据防火墙所采用的技术不同,我们可以将它分为四种基本类型:包过滤型、网络地址转换—NAT、代理型和监测型。

### 2.1.包过滤型

包过滤型产品是防火墙的初级产品,其技术依据是网络中的分包传输技术。网络上的数据都是以“包”为单位进行传输的,数据被分割成为一定大小的数据包,每一个数据包中都会包含一些特定信息,如数据的源地址、目标地址、TCP/UDP 源端口和目标端口等。防火墙通过读取数据包中的地址信息来判断这些“包”是否来自可信任的安全站点,一旦发现来自危险站点的数据包,防火墙便会将这些数据拒之门外。系统管理员也可以根据实际情况灵活制订判断规则。

包过滤技术的优点是简单实用,实现成本较低,在应用环境比较简单的情况下,能够以较小的代价在一定程度上保证系统的安全。

但包过滤技术的缺陷也是明显的。包过滤技术是一种完全基于网络层的安全技术,只能根据数据包的来源、目标和端口等网络信息进行判断,无法识别基于应用层的恶意侵入,如恶意的 Java 小程序以及电子邮件中附带的病毒。

有经验的黑客很容易伪造 IP 地址,骗过包过滤型防火墙。

## 2.2 网络地址转化—NAT

网络地址转换是一种用于把 IP 地址转换成临时的、外部的、注册的 IP 地址标准。它允许具有私有 IP 地址的内部网络访问因特网。它还意味着用户不许要为其网络中每一台机器取得注册的 IP 地址。

NAT 的工作过程在内部网络通过安全网卡访问外部网络时,将产生一个映射记录。系统将外出的源地址和源端口映射为一个伪装的地址和端口,让这个伪装的地址和端口通过非安全网卡与外部网络连接,这样对外就隐藏了真实的内部网络地址。在外部网络通过非安全网卡访问内部网络时,它并不知道内部网络的连接情况,而只是通过一个开放的 IP 地址和端口来请求访问。OLM 防火墙根据预先定义好的映射规则来判断这个访问是否安全。当符合规则时,防火墙认为访问是安全的,可以接受访问请求,也可以将连接请求映射到不同的内部计算机中。当不符合规则时,防火墙认为该访问是不安全的,不能被接受,防火墙将屏蔽外部的连接请求。网络地址转换的过程对于用户来说是透明的,不需要用户进行设置,用户只要进行常规操作即可。

## 2.3.代理型

代理型防火墙也可以被称为代理服务器,它的安全性要高于包过滤型产品,并已经开始向应用层发展。代理服务器位于客户机与服务器之间,完全阻挡了二者间的数据交流。从客户机来看,代理服务器相当于一台真正的服务器;而从服务器来看,代理服务器又是一台真正的客户机。当客户机需要使用服务器上的数据时,首先将数据请求发给代理服务器,代理服务器再根据这一请求向服务器索取数据,然后再由代理服务器将数据传输给客户机。由于外部系统与内部服务器之间没有直接的数据通道,外部的恶意侵害也就很难伤害到企业内部网络系统。

代理型防火墙的优点是安全性较高,可以针对应用层进行侦测和扫描,对付基于应用层的侵入和病毒都十分有效。其缺点是对系统的整体性能有较大的影响,而且代理服务器必须针对客户机可能产生的所有应用类型逐一进行设置,大大增加了系统管理的复杂性。

#### 2.4.监测型

监测型防火墙是新一代的产品,这一技术实际已经超越了最初的防火墙定义。监测型防火墙能够对各层的数据进行主动的、实时的监测,在对这些数据加以分析的基础上,监测型防火墙能够有效地判断出各层中的非法侵入。同时,这种检测型防火墙产品一般还带有分布式探测器,这些探测器安置在各种应用服务器和其他网络的节点之中,不仅能够检测来自网络外部的攻击,同时对来自内部的恶意破坏也有极强的防范作用。据权威机构统计,在针对网络系统的攻击中,有相当比例的攻击来自网络内部。因此,监测型防火墙不仅超越了传统防火墙的定义,而且在安全性上也超越了前两代产品。

虽然监测型防火墙安全性上已超越了包过滤型和代理服务器型防火墙,但由于监测型防火墙技术的实现成本较高,也不易管理,所以目前在实用中的防火墙产品仍然以第二代代理型产品为主,但在某些方面也已经开始使用监测型防火墙。基于对系统成本与安全技术成本的综合考虑,用户可以选择性地使用某些监测型技术。这样既能够保证网络系统的安全性需求,同时也能有效地控制安全系统的总拥有成本。

实际上,作为当前防火墙产品的主流趋势,大多数代理服务器(也称应用网关)也集成了包过滤技术,这两种技术的混合应用显然比单独使用具有更大的优势。由于这种产品是基于应用的,应用网关能提供对协议的过滤。





