

---

# 毕业生考勤管理系统的设计与 实现

---

## 摘 要

随着 Internet 的迅猛发展，当今社会已进入网络时代。利用网络技术提高教学管理水平，对打造高品牌高职院校更有着广泛深远的意义。毕业生考勤系统是高校教学工作的一个重要环节。如何充分利用紧缺的教学资源，提高资源的使用率，做到对毕业生考勤系统的高效的，及时的指导，已成为毕业生考勤系统乃至整个学校管理的焦点。因此，设计并开发一个具有普遍性的便捷的毕业生考勤系统，对于提高高校毕业生考勤系统指导效率很有必要。

基于当前计算机前和网络技术的快速发展与普及，本文设计并开发了基于动态服务器网页的毕业生考勤系统。该系统主要实现的功能主要包括用户管理、系统基本信息维护、毕业生考勤系统、老师和留言管理等。其中用户管理主要用来实现对各类用户的信息进行设置管理。毕业生考勤系统主要用来实现管理员对毕业生考勤系统的格式进行设定。毕业生考勤系统主要是用来实现管理员上传有关毕业生考勤系统的任务如开题报告等。毕业生考勤系统主要是用来实现教师对学生的毕业生考勤基地进行管理，学生对自己的任务进行管理。留言管理主要是用来实现教师和学生之间的相互留言，教师把毕业生考勤系统的修改意见及其它信息以留言的方式发给学生并且支持群发留言的功能。

本文从软件开发生命周期的角度系统的论述了系统分析、总体设计、详细设计和系统具体实现的过程。系统分析着重论述系统开发的必要性、系统功能需求以及系统架构的应用；总体设计包括系统需求分析和数据库的设计；详细设计具体论述系统的实现过程；测试部分主要对系统进行测试，检测其性能。本文具体论述了面向对象设计方法、系统设计的原则、系统设计的总体目标以及体系结构的特点。

本系统前台主要使用 JSP 作为开发语言，后台使用 Mysql 作为数据库管理系统，开发环境是 MyEclipse，服务器采用 tomcat，开发出的一个基于 Web 技术的 B/S 结构的毕业生考勤系统。

**关键词：**毕业生考勤管理，B/S 模式

---

# 目 录

摘 要 .....	I
ABSTRACT .....	错误！未定义书签。
第一章 绪论 .....	1
1.1 课题背景 .....	1
1.2 目的和意义 .....	2
1.3 开发工具及技术 .....	2
1.3.1 开发工具 .....	2
1.3.2 JSP .....	3
1.3.3 JavaScript .....	4
1.4 软硬件需求 .....	4
第二章 需求分析 .....	5
2.1 可行性分析 .....	5
2.1.1 技术的可行性 .....	6
2.2.2 经济的可行性 .....	6
2.2.3 操作可行性 .....	6
2.2.4 法律的可行性 .....	6
2.3 系统用户用例图 .....	6
2.3.1 用户模块用例图 .....	6
2.3.2 基本信息用例图 .....	7
2.4 功能模块需求分析 .....	8
2.5 设计的基本思想 .....	8
2.6 性能需求 .....	9
2.6.1 系统的安全性 .....	9
2.6.2 数据的完整性 .....	9
2.7 界面需求 .....	9
3 系统的详细设计 .....	11
3.1 功能模块介绍 .....	11
3.2 详细设计介绍 .....	11
3.2.1 管理员登录 .....	11
3.2.2 学生信息管理 .....	14
3.2.3 老师管理模块 .....	19
3.2.4 文档管理系统模块 .....	25
3.2.5 修改个人密码 .....	26

---

3.2.6 通知信息管理界面 .....	26
3.2.7 老师管理后台界面 .....	27
3.2.8 退出后台管理模块 .....	27
3.2.9 学生后台管理界面 .....	28
4 系统测试 .....	28
4.1 程序测试 .....	29
4.2 测试方法 .....	30
4.3 系统测试结果 .....	31
总结 .....	32
参考文献 .....	32
附录 1 .....	34
附录 2 .....	35

---

# 第一章 绪论

## 1.1 课题背景

计算机网络技术的发展，给信息时代的人们带来了很大的方便。如今在 Internet 上，随处都可以看到种类繁多的信息管理系统，比如，企业办公化自动管理系统，电子商务系统等等。随着信息技术的日益发展已深入到社会的各个角落，网站系统也不例，所谓网站（website），就是指在国际网络（英特网）上，根据一定的规则，使用 html 等工具制作的用于展示特定内容的相关网页的集合，简单的说。网站就是一种通信工具，就像不搞栏一样，人们可以通过网站来发布自己想要公开的咨询（信息），或者利用网络来提供相关的网络服务，人们可以通过网页浏览器来访问网站，获取自己需要的咨询（信息）或者享受网络服务。

随着各高校的持续扩招，本科学生人数的不断增多，有的学生提前离校毕业生考勤在外，传统的毕业生考勤系统指导方式已不适合新形势下的要求。如何使老师对学生的毕业生考勤系统进行高效、快捷的指导工作，如何检查众多毕业生考勤系统的格式是否符合标准，如何使有限的资源得到充分的利用，已经成为各高校亟待解决的问题。在网络技术迅猛发展的今天，网络作为获取信息以及彼此交流的平台，发挥着重要的作用。网络教学平台打破了传统教学的时空限制，增进了师生的交流，提高了教学质量和效果[1]。充分利用现代技术，依托计算机网络平台，创建毕业生考勤系统，对提高毕业生考勤系统水平，提升毕业生考勤系统写作质量必将大有裨益。

毕业生考勤系统的管理通过网络平台，教师可以实施毕业生考勤系统写作的跟踪监控，了解学生的思想动态，发布毕业生考勤系统指导信息，追踪毕业生考勤系统写作进程，沟通师生信息[2]。毕业生考勤系统指导过程中，常常因为时间的关系，因为地域的限制，因为条件的局限，造成师生之间联系的不便和沟通的困难，利用网络平台，可以实现师生之间实时的、非实时的快速、高效、方便的双向沟通[3]。在毕业生考勤系统写作过程中，由于主客观条件的限制，学生常常会遇到各种各样的困难和障碍，对毕业生考勤系统的写作感到压力很大，心中无数，难以下笔。当学生面对许多困难和障碍时，如果得不到克服和解决，就可能因陋就简，敷衍了事，甚至铤而走险，抄袭造假。充分利用此系统，学生可以获得随时随地的指导。同时也可以大大地提高了毕业生考勤系统的效率。

---

## 1.2 目的和意义

本文主要工作内容是梳理毕业生信息管理工作的流程,吸收、借鉴先进的指导理念,再结合学校的具体情况,实现毕业生信息管理流程再造,使毕业生信息管理体系更加科学化和规范化。在此基础上,以校园网为平台,开发“毕业生信息管理”,完成教务综合管理的信息化、集成化和自动化建设[4],主要研究内容有如下基本功能模块:学生注册、用户登录、学生提交任务、教师对文档管理、学生留言、教师答复、评分和企业内容发布。

毕业生信息管理的用户群有三类:系统管理员、老师、学生。

1、学生,学生作为毕业生考勤教学的主要参与者,可通过系统了

解企业的毕业生考勤岗位与要求,根据自身条件选择适合的企业;并能通过系统将毕业生考勤过程中发生的问题向自己的指导教师进行汇报,及时得到指导教师的指导;在毕业生考勤结束时,通过系统向指导教师提交任务;可通过系统下载在毕业生考勤过程中应填具的表格与文档。

2、教师,作为毕业生考勤教学任务的指导者,要通过系统

及时了解毕业生考勤管理部门发布的对毕业生考勤教学任务的要求与规定;及时了解学生在实

习过程中遇到的问题,及时加以解决;在毕业生考勤结束时,可通过系统下载学生提交

的任务,经批阅后在系统中提交评语,给予分数评定。

3、管理者,作为系统的管理人员,负责注册用户帐户;通知、新闻等信息的发布;根据毕业生考勤计划落实实践教学任务,为学生指派毕业生考勤指导教师;在毕业生考勤过程中,可根据毕业生的情况对毕业生分配进行调整。

## 1.3 开发工具及技术

### 1.3.1 开发工具

此次设计主要采用 MyEclipse 加 Tomcat 后台服务器进行,设计过程中页面主要使用 JSP 技术完成,下面对 MyEclipse、Tomcat 和 Mysql 数据库进行简要介绍。

#### 1.3.1.1 MyEclipse

MyEclipse,是一个十分优秀的用于开发 Java, J2EE 的 Eclipse 插件集合,MyEclipse 的功能非常强大,支持也十分广泛,尤其是对各种开元产品的支持十分不错。MyEclipse 企业级工作平台(MyEclipse Enterprise Workbench, 简称 MyEclipse)是对 Eclipse

---

IDE 的扩展，利用它我们可以在数据库和 JavaEE 的开发、发布，以及应用程序服务器的整合方面极大的提高工作效率。它是功能丰富的 JavaEE 集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持 HTML, Struts, JSP, CSS, Javascript, SQL, Hibernate。

#### 1.3.1.2 Tomcat

Tomcat 是一个小型的轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。可以这样认为，当在一台机器上配置好 Apache 服务器，可利用它响应对 HTML 页面的访问请求。实际上 Tomcat 部分是 Apache 服务器的扩展，但它是独立运行的，所以当运行 tomcat 时，它实际上作为一个与 Apache 独立的进程单独运行的。

#### 1.3.1.3 Mysql

Mysql 使用 C 和 C++编写，并使用了多种编译器进行测试，保证源代码的可移植性；支持 AIX、FreeBSD、HP-UX、Linux、Mac OS、NovellNetware、OpenBSD、OS/2 Wrap、Solaris、Windows 等多种操作系统；为多种编程语言提供了 API；支持多线程，充分利用 CPU 资源；提供 TCP/IP、ODBC 和 JDBC 等多种数据库连接途径；可以处理拥有上千万条记录的大型数据库。对于一般的个人使用者和中小型企业来说，Mysql 提供的功能已经绰绰有余，而且由于 Mysql 是开放源码软件，因此可以大大降低总体拥有成本。

### 1.3.2 JSP

JSP 技术使用 Java 编程语言编写类 XML 的 tags 和 scriptlets，来封装产生动态网页的处理逻辑。网页还能通过 tags 和 scriptlets 访问存在于服务端的资源的应用逻辑。JSP 将网页逻辑与网页设计和显示分离，支持可重用的基于组件的设计，使基于 Web 的应用程序的开发变得迅速和容易。

Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的程序段，然后将执行结果连同 JSP 文件中的 HTML 代码一起返回给客户。插入的 Java 程序段可以操作数据库、重新定向网页等，以实现建立动态网页所需要的功能。JSP 与 Servlet 一样，是在服务器端执行的，通常返回给客户端就是一个 HTML 文本，因此客户端只要有浏览器能浏览。JSP 页面由 HTML 代码和嵌入其中的 Java 代码所组成。服务器在页面被客户端请求以后对这些 Java 代码进行处理，然后将生成的 HTML 页面返回给客户端的浏览器。Servlet 是 JSP 的技术基础，而且大型的 Web 应用程序的开发需要 Java Servlet 和 JSP 配合才能完成。JSP 具备了 Java 技术的简单易用，完全的面向对象，具有平台无关性且安全可靠

---

靠，主要面向因特网的所有特点。JSP 技术的优点：

(1) 一次编写，到处运行。除了系统之外，代码不用做任何更改。

(2) 系统的多平台支持。基本上可以在所有平台上的任意环境中开发，在任意环境中进行系统部署，在任意环境中扩展。相比 ASP/.net 的局限性是显而易见的。

(3) 强大的可伸缩性。从只有一个小的 Jar 文件就可以运行 Servlet/JSP，到由多台服务器进行集群和负载均衡，到多台 Application 进行事务处理，消息处理，一台服务器到无数台服务器，Java 显示了一个巨大的生命力。

(4) 多样化和功能强大的开发工具支持。Java 已经有了许多非常优秀的开发工具，而且许多可以免费得到，并且其中许多已经可以顺利的运行于多种平台之下

(5) 支持服务器端组件。web 应用需要强大的服务器端组件来支持，开发人员需要利用其他工具设计实现复杂功能的组件供 web 页面调用，以增强系统性能。JSP 可以使用成熟的 JAVA BEANS 组件来实现复杂商务功能。

内部对象说明：request 客户端请求，此请求会包含来自 GET/POST 请求的参数；response 网页传回客户端的响应；pageContext 网页的属性是在这里管理；session 与请求有关的会话；application servlet 正在执行的内容；out 用来传送响应的输出流；config 代码片段配置对象；page JSP 网页本身；exception 针对错误网页，未捕捉的例外

### 1.3.3 JavaScript

JavaScript 是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言。同时也是一种广泛用于客户端 Web 开发的脚本语言，常用来给 HTML 网页添加动态功能，比如响应用户的各种操作。JavaScript 的一个重要功能就是面向对象的功能，通过基于对象的程序设计，可以用更直观、模块化和可重复使用的方式进行程序开发。在 HTML 基础上，使用 Javascript 可以开发交互式 Web 网页。Javascript 的出现使得网页和用户之间实现了一种实时性的、动态的、交互性的关系，使网页包含更多活跃的元素和更加精彩的内容。在本系统中很多地方使用了 javascript 技术，比如说，检验用户输入数据的有效性，是否重复，是否为空等等。

## 1.4 软硬件需求

硬件需求：

CPU: Pentium 以上计算机

内存: 512M 以上



---

软件需求：

操作系统版本：Windows XP /vista/Win7

开发工具：MyEclipse 6.0.1

后台服务器：Apache Tomcat 6.0

开发语言：Java

浏览器：IE6.0

## 第二章 需求分析

### 2.1 可行性分析

开发任何一个系统，都会受到时间和资源上的限制。因此，在每一个项目开发之前，都要进行可行性分析，可以减少项目的开发风险，避免人力、物力和财力的浪费。下面

---

就技术、经济、操作和法律四个方面来介绍。

### 2.1.1 技术的可行性

本系统开发工具是 MyEclipse 和 Mysql 数据库，开发语言是 Java，主要使用了 J2EE 的技术，java 是一种面向对象编程语言，简单易学而且灵活方便。大三时就学习了 java 课程，大四 期间也系统的了解了 J2EE 的知识，本系统总体上开发难度不高，数据库的设计和操作的系统是设计的核心。在大学期间学习过软件工程，软件测试，UML 统一建模语言等课程，每个学期也会完成对应的课程设计，具备一定的系统分析、设计和测试能力。因此，完成系统实现在技术上完全具有可行性。

### 2.2.2 经济的可行性

本系统所用的软件都是开源的，为开发软件节省了大量的金钱和时间，达到降低成本，提高开发效率的目的，本系统对计算机配置的要求不高，甚至网吧更换下来的低配置电脑都可以完全满足需要，所以在经济上具有完全的可行性。

### 2.2.3 操作可行性

本系统操作简单，输入信息页面大多数都是下拉框的选择形式，在某些页面，信息可以自动生成，无需输入，时间的输入也是用的日历控件，操作简便，对操作人员的要求很低，只需对 WINDOWS 操作熟练，加之对本系统的操作稍加培训即可工作，而且本系统可视性非常好，所以在技术上不会有很大难度。

### 2.2.4 法律的可行性

此旅游指南展示网是自行开发的网络系统，是很有实际意义的系统，开发环境软件和使用数据库都是开源代码，开发这个系统不同于开发普通的系统软件，不存在侵权等问题，即法律上是可行的。

## 2.3 系统用户用例图

### 2.3.1 用户模块用例图

用户模块用例图如图 2.1 所示：

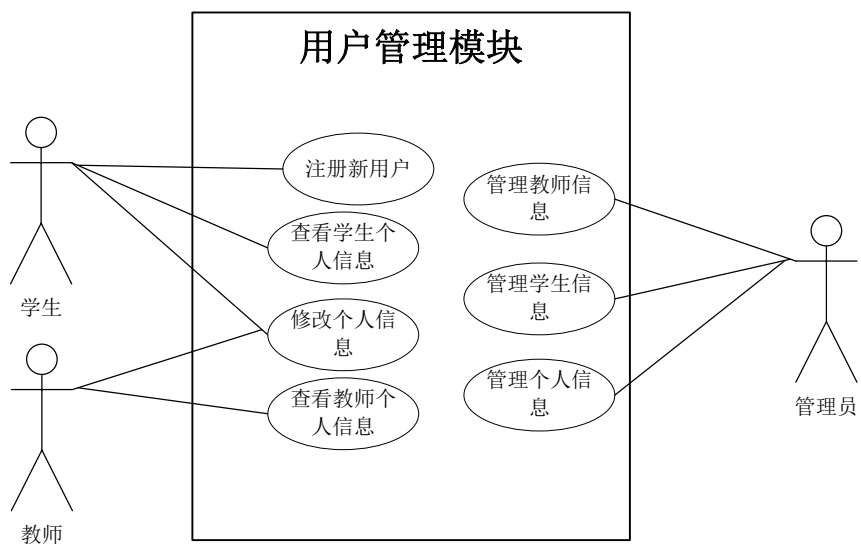


图 2.1 普通用户用例图

### 2.3.2 基本信息用例图

基本信息用例图。

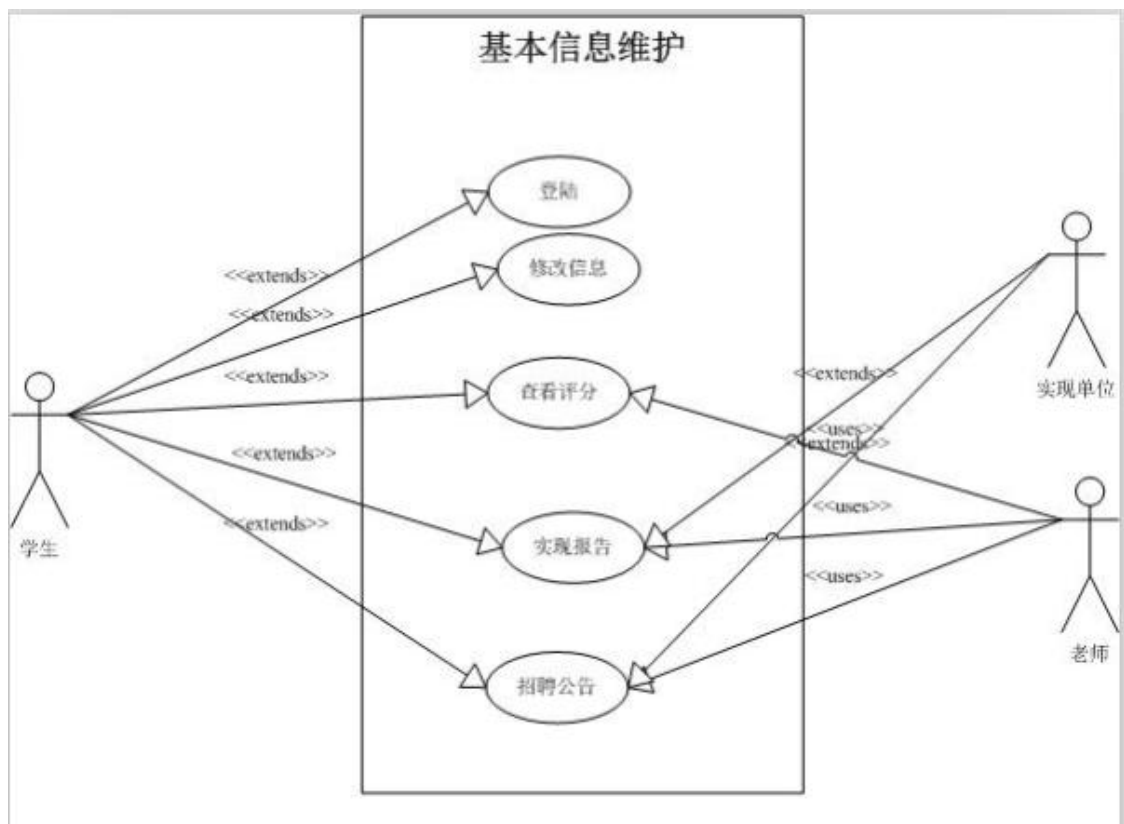


图 2.3 管理员用例图

---

## 2.4 功能模块需求分析

本系统最大的特点是使用操作简单、友好的提示信息。本系统将实现以下基本功能：

（1）教师：考勤管理：查看学生考勤信息

毕业生管理：查看学生信息

个人信息管理：查看本人信息（可以实现改）

文档管理：发布任务

查看学生提交的文档

留言管理：查看留言，回复留言。（可以实现增删改查）

（2）学生：考勤管理：点到

个人信息管理：查看本人信息

文档管理：查看老师发布的任务

留言管理：查看留言，回复留言。

（3）管理员：管理员信息管理

毕业生管理：查看学生信息

教师管理：查看教师信息

留言管理：查看留言，回复留言。

## 2.5 设计的基本思想

设计思想遵循以下几点：

1. 采用 B/S 模式进行开发，其优点是后台与前台处理层次分明，而且符合众多已经习惯网页搜索方式的用户。

2. 采用面向对象的开发与设计理念。运用面向对象技术的前提是对整体系统的高度和准确抽象，通过它可以保证系统良好的框架，进而带来产品较强的稳定性和运行效率。

3. 采用模块化设计。模块化设计要求将整个系统划分成基于小的模块，有利于代码的重载，简化设计和实现过程。

4. 简单方便的系统界面。设计简单友好的系统界面，方便用户较快的适应系统的操作。

5. 速度优先原则。由于此工具最重要的评测标准就是速度，因此在设计过程中，具体过程尽量做到资源占用少，速度快。

6. 设计既要突出重点，又要细致周到。要符合设计需求，在有可能改进的地方进行扩充，使系统更适应用户的需要。

---

## 2.6 性能需求

### 2.6.1 系统的安全性

本毕业生考勤系统和管理权限上要严格进行控制，具体要求如下：

1. 要想管理本系统的信息，必须先要凭用户名和密码登陆到系统中，没有权限的用户不能通过任何方式登录系统查看系统的任何信息和数据，以确保系统的严密性和安全性。

2. 在具体实现中设定不同权限，不同权限用户登录到系统后，不能越级操作，管理员可以使使用所有模块；普通用户只能在线浏览信息，

### 2.6.2 数据的完整性

1. 各种记录信息的完整性，信息记录内容不能为空
2. 各种数据间相互联系的正确性
3. 相同数据在不同记录中的一致性

## 2.7 界面需求

界面设计目前已经成为评价软件质量的一条重要指标，一个好的用户界面可以增加用户使用系统的信心和兴趣，提高工作效率，JSP 技术是用 JAVA 语言作为脚本语言的，JSP 网页为整个服务器端的 JAVA 库单元提供了一个接口来服务于 HTTP 的应用程序。创建动态页面非常方便。用户界面是指软件系统与用户交互的接口，通常包括输出、输入、人-机对话的界面格式等。

### 1. 输出设计

输出是由计算机对输入的原始信息进行加工处理，形成高质量的有效信息，并使之具有一定的格式，提供管理者使用，这是输出设计的主要职责和目标。

系统设计的过程正好和实施过程相反，并不是从输入设计到输出设计，而是从输出设计到输入设计，这是因为输出表格直接与使用者相联系，设计的出发点应当是保证输出表格方便地为使用者服务，正确及时反映和组成用于各部门的有用信息。输出设计的原则是考虑既要全面反映不同管理层的各项需要，又要言简意赅，不要将用户需要和不需要的都提供给用户。

### 2. 输入设计

输入数据的收集和录入是比较费事的，需要大量的人力和一定设备，并且容易出错。如果输入系统的数据有错误，则处理后的输出将扩大这些错误，因此输入数据的正确性对于整个系统质量的好坏是具有决定性意义的。

---

输入设计的原则有如下几点：

1) 输入量应保持在能满足处理要求的最低限度。设计中可采用设置字段初值，下拉式数据窗口等方式尽量减少用户键盘输入量。输入量越少，错误率就越少，数据准备时间也减少。

2) 输入的准备及输入过程应尽量容易进行，从而减少错误的发生。

3) 应尽量早对输入数据进行检查（尽量接近原数据发生点），以便使错误及时得到更正。

4) 输入数据尽早地用其处理所需的形式被记录，以避免数据由一种介质转移到另一种介质时需要转录而可能发生的错误

---

## 3 系统的详细设计

### 3.1 功能模块介绍

本系统最大的特点是使用操作简单、友好的提示信息。本系统将实现以下基本功能：

（1）教师：考勤管理：查看学生考勤信息

毕业生管理：查看学生信息

个人信息管理：查看本人信息（可以实现改）

文档管理：发布任务

查看学生提交的文档

留言管理：查看留言，回复留言。（可以实现增删改查）

（2）学生：考勤管理：点到

个人信息管理：查看本人信息

文档管理：查看老师发布的任务

留言管理：查看留言，回复留言。

（3）管理员：管理员信息管理

毕业生管理：查看学生信息

教师管理：查看教师信息

留言管理：查看留言，回复留言。

### 3.2 详细设计介绍

#### 3.2.1 管理员登录

管理员通过后台登录入口进入后台登录模块，后台登录模块主要用于验证管理员的身份和密码。后台登录模块的运行效果如图 3-1：

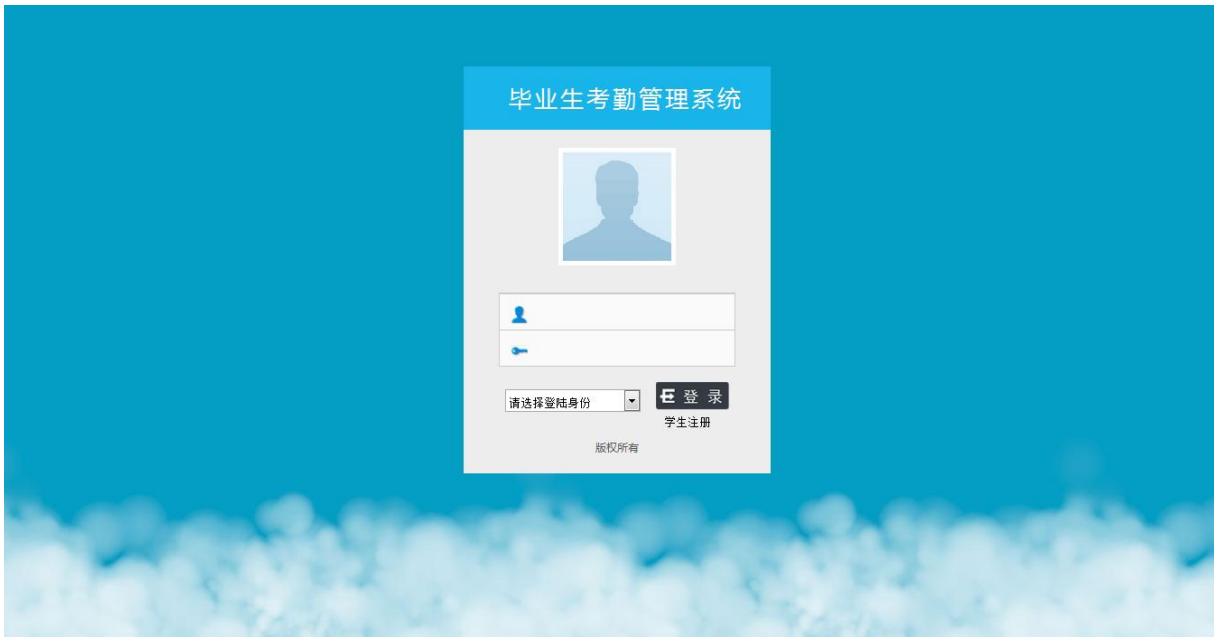


图 3-1 后台登录模块的设计效果图

后台登录流程图如图 3-2 所示：

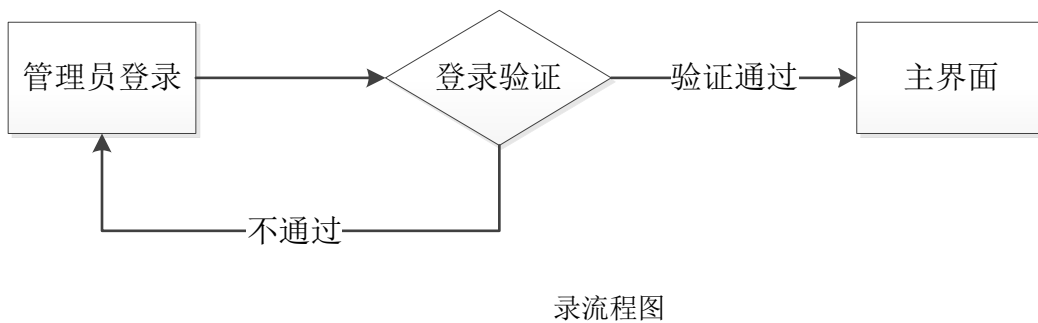


图 3-2 后台登录流程图

判断进入者身份的相应代码：

```

public String login(String userName,String userPw,int userType)
{
    try
    {
        Thread.sleep(700);
    } catch (InterruptedException e)
    {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    String result="no";
    if(userType==0)//系统管理员登陆
    {
        String sql="select * from t_admin where userName=? and userPw=?";
    }
}
  
```



---

```
Object[] params={userName,userPw};
DB mydb=new DB();
mydb.doPstm(sql, params);
try
{
    ResultSet rs=mydb.getRs();
    boolean mark=(rs==null?!rs.next()?false:true);
    if(mark==false)
    {
        result="no";
    }
    else
    {
        result="yes";
        TAdmin admin=new TAdmin();
        admin.setUserId(rs.getInt("userId"));
        admin.setUserName(rs.getString("userName"));
        admin.setUserPw(rs.getString("userPw"));
        WebContext ctx = WebContextFactory.get();
        HttpSession session=ctx.getSession();
        session.setAttribute("userType", 0);
        session.setAttribute("admin", admin);
    }
    rs.close();
}
catch (SQLException e)
{
    System.out.println("登录失败！ ");
    e.printStackTrace();
}
finally
{
    mydb.closed();
}

}
return result;
}
```

进入后台后可以点击左侧学生信息管理、老师管理信息管理、文档管理、修改个人密码等菜单，进入相应的管理页面，管理页面如图 3-2。



图 3-2 后台页面运行结果图

### 3.2.2 学生信息管理

本模块实现对要学生基本信息的管理操作，包括学生信息的添加、删除操作。

点击学生信息管理菜单，进入学生信息管理页面，该页面通过列表列出已添加的学生信息，包括学号、姓名、性别、年级参数。点击列表的删除按钮，实现学生信息的删除操作。如图 3-3：

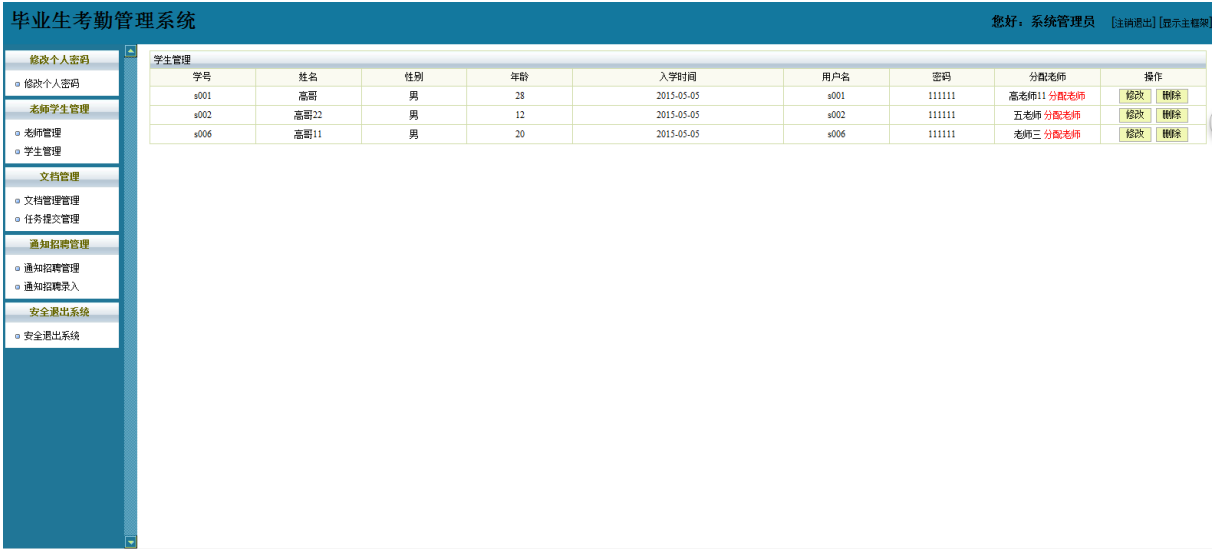


图 3-3 学生信息管理

点击添加按钮，进入学生信息添加页面，输入符合条件的学生信息，完成学生的添加操作。如图 3-4：

学生注册	
学号:	<input type="text"/> *
姓名:	<input type="text"/> *
性别:	<input checked="" type="radio"/> 男 <input type="radio"/> 女
年龄:	<input type="text"/>
入学时间:	<input type="text"/>
用户名:	<input type="text"/> *
密 码:	<input type="password"/> *
<input type="button" value="确定"/> <input type="button" value="取消"/>	

图 3-4 学生信息添加

学生修改模块实现：点击学生信息管理后面的编辑按钮，进入该学生信息的编辑页面。设计界面图 3-5 如下：

学生修改	
学号:	<input type="text" value="s001"/>
姓名:	<input type="text" value="高哥"/>
性别:	<input checked="" type="radio"/> 男 <input type="radio"/> 女
年龄:	<input type="text" value="28"/>
入学时间:	<input type="text" value="2014-04-09"/>
用户名:	<input type="text" value="s001"/>
密 码:	<input type="password" value="....."/>
<input type="button" value="确定修改"/>	

图 3-5 学生信息修改

该模块的主要功能 stuAction 类实现的，该类的主要功能是与数据库交互，查询或保存学生信息，并通过 response 跳转，打开相应的 jsp 页面。主要代码是：

```
public String addXueyuan() throws Exception {  
    xueyuan = new Xueyuan();
```

---

```
        xueyuan.setId(id);
        xueyuan.setXueyuanno(xueyuanno);
        xueyuan.setName(name);
        xueyuan.setSex(sex);
        xueyuan.setAge(age);
        xueyuan.setTel(tel);
        xueyuan.setEmail(email);
        xueyuan.setQq(qq);
        xueyuan.setAddress(address);
        xueyuan.setTijiantime(tijiantime);
        xueyuanDao.save(xueyuan);
        this.setMessage("操作成功");
        this.setPath("listXueyuan.action");
        return "succeed";
    }
```

```
public String listXueyuan() throws Exception {
    List XueyuanList = xueyuanDao.findAll();
    Map request = (Map) ServletActionContext.getContext().get("request");
    request.put("XueyuanList", XueyuanList);
    return ActionSupport.SUCCESS;
}
```

```
public String toEditXueyuan() throws Exception {
    Xueyuan entity = xueyuanDao.findById(id);
    Map request = (Map) ServletActionContext.getContext().get("request");
    request.put("Xueyuan", entity);
    return ActionSupport.SUCCESS;
}
```

```
public String editXueyuan() throws Exception {
```

---

```
        xueyuan = xueyuanDao.findById(id);
        xueyuan.setId(id);
        xueyuan.setXueyuanno(xueyuanno);
        xueyuan.setName(name);
        xueyuan.setSex(sex);
        xueyuan.setAge(age);
        xueyuan.setTel(tel);
        xueyuan.setEmail(email);
        xueyuan.setQq(qq);
        xueyuan.setAddress(address);
        xueyuan.setTijiantime(tijiantime);
        xueyuanDao.merge(xueyuan);
        this.setMessage("操作成功");
        this.setPath("listXueyuan.action");
        return "succeed";
    }
```

```
public String infoXueyuan() throws Exception {
    Xueyuan entity = xueyuanDao.findById(id);
    Map request = (Map) ServletActionContext.getContext().get("request");
    request.put("Xueyuan", entity);
    return ActionSupport.SUCCESS;
}
```

```
public String deleteXueyuan() throws Exception {
    Xueyuan entity = xueyuanDao.findById(id);
    xueyuanDao.delete(entity);
    this.setMessage("操作成功");
    this.setPath("listXueyuan.action");
    return "succeed";
}
```

---

```
public Xueyuan getXueyuan() {  
    return xueyuan;  
}
```

```
public void setXueyuan(Xueyuan xueyuan) {  
    this.xueyuan = xueyuan;  
}
```

```
public XueyuanDAO getXueyuanDao() {  
    return this.xueyuanDao;  
}
```

```
public void setXueyuanDao(XueyuanDAO xueyuanDao) {  
    this.xueyuanDao = xueyuanDao;  
}
```

```
public String getMessage() {  
    return message;  
}
```

```
public void setMessage(String message) {  
    this.message = message;  
}
```

```
public String getPath() {  
    return path;  
}
```

```
public void setPath(String path) {  
    this.path = path;  
}
```

---

```
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getXueyuanno() {
    return xueyuanno;
}
```

### 3.2.3 老师管理模块

本模块实现对老师操作，包括老师信息的添加、删除操作。

点击老师菜单，进入老师页面，该页面通过列表列出已添加的老师点击列表的删除按钮，实现老师信息的删除操作。如图 3-6:

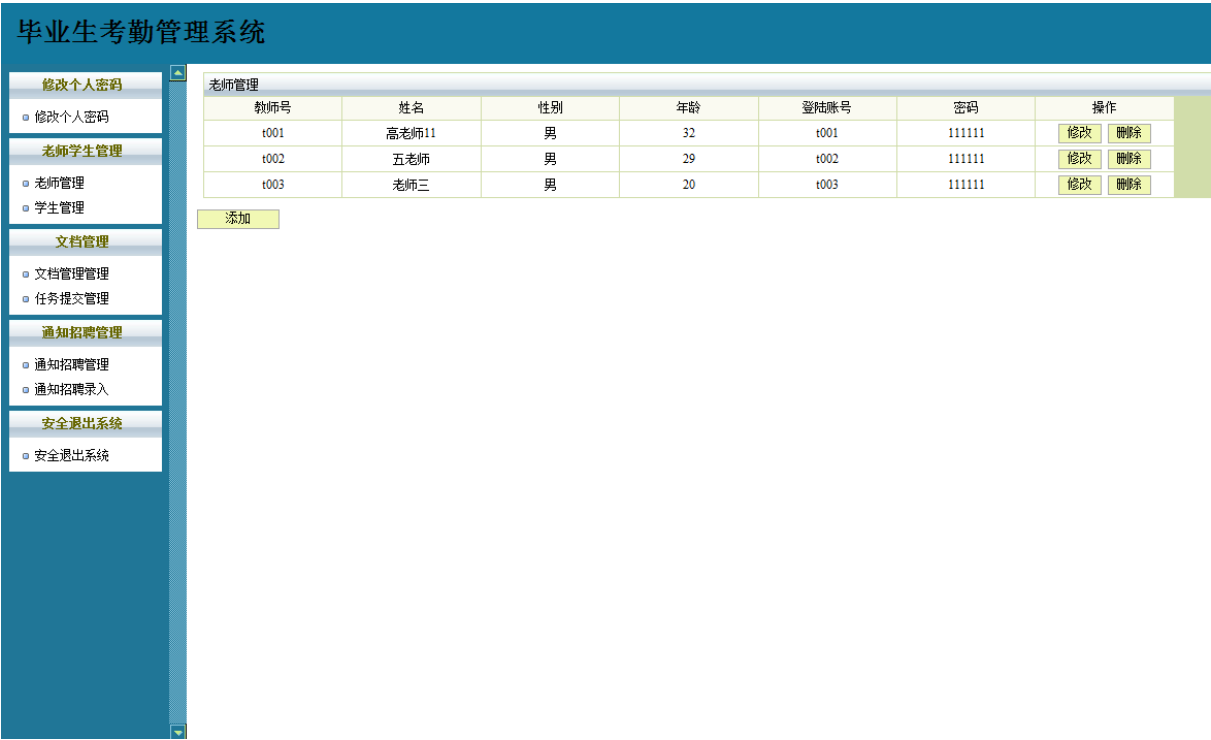


图 3-6 老师信息管理

点击添加按钮，进入老师添加页面，输入符合条件的老师信息，完成老师的添加操作。如图 3-7:

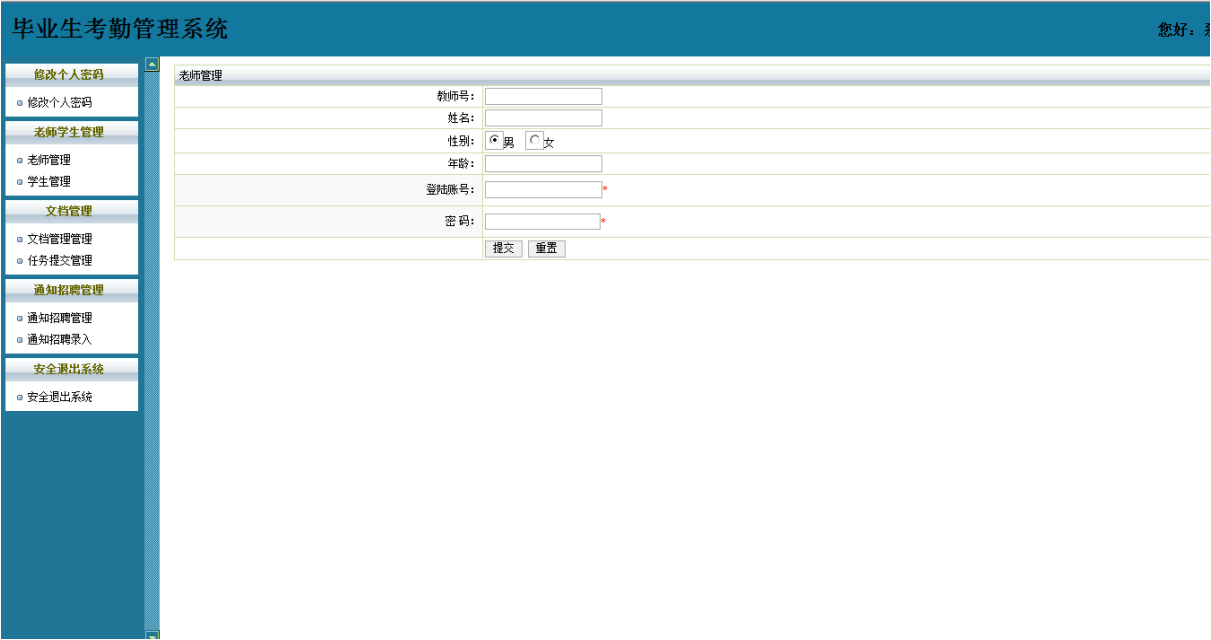


图 3-7 老师信息添加

主要代码:

```
public String teaAdd()
{
```



---

```
    TTea tea=new TTea();
    tea.setBianhao(bianhao);
    tea.setName(name);
    tea.setSex(sex);
    tea.setAge(age);
    tea.setLoginname(loginname);
    tea.setLoginpw(loginpw);
    tea.setDel("jiaoshi");

    Session session= HibernateSessionFactory.getSession();
    Transaction transaction=session.beginTransaction();
    session.save(tea);
    transaction.commit();
    HibernateSessionFactory.closeSession();

    this.setMessage("操作成功");
    this.setPath("teaMana.action");
    return "succeed";
}
```

```
public String teaMana()
{
    String sql="from TTea where del='jiaoshi'";
    Session session= HibernateSessionFactory.getSession();
    Query query = session.createQuery(sql);
    List teaList=query.list();
    HibernateSessionFactory.closeSession();

    Map request=(Map)ServletActionContext.getContext().get("request");
    request.put("teaList", teaList);
    return ActionSupport.SUCCESS;
}
```

```
public String teaDel()
{

    Session session= HibernateSessionFactory.getSession();
    Transaction transaction=session.beginTransaction();
```

---

```
String sql="update TTea set del='yes' where id=?";
Query query = session.createQuery(sql);
query.setInteger(0,id);
int row=query.executeUpdate();
transaction.commit();

HibernateSessionFactory.closeSession();

this.setMessage("操作成功");
this.setPath("teaMana.action");
return "succeed";
}

public String teaToEdit(){
    TTea tea=liuService.getTeaById(id);
    Map request=(Map)ServletActionContext.getContext().get("request");
    request.put("tea", tea);
    return ActionSupport.SUCCESS;
}

public String teaEdit()
{
    TTea tea=liuService.getTeaById(id);
    tea.setBianhao(bianhao);
    tea.setName(name);
    tea.setSex(sex);
    tea.setAge(age);
    tea.setLoginname(loginname);
    tea.setLoginpw(loginpw);
    tea.setDel("jiaoshi");

    Session session= HibernateSessionFactory.getSession();
    Transaction transaction=session.beginTransaction();
    session.update(tea);
    transaction.commit();
    HibernateSessionFactory.closeSession();

    Map request=(Map)ServletActionContext.getContext().get("request");
```

---

```
        request.put("msg", "修改成功");
        return "msg";
    }
```

```
public String teaAdd1()
{
    TTea tea=new TTea();
    tea.setBianhao(bianhao);
    tea.setName(name);
    tea.setSex(sex);
    tea.setAge(age);
    tea.setLoginname(loginname);
    tea.setLoginpw(loginpw);
    tea.setDel("qiye");

    Session session= HibernateSessionFactory.getSession();
    Transaction transaction=session.beginTransaction();
    session.save(tea);
    transaction.commit();
    HibernateSessionFactory.closeSession();

    this.setMessage("操作成功");
    this.setPath("teaMana1.action");
    return "succeed";
}
```

```
public String teaMana1()
{
    String sql="from TTea where del='qiye'";
    Session session= HibernateSessionFactory.getSession();
    Query query = session.createQuery(sql);
    List teaList=query.list();
    HibernateSessionFactory.closeSession();

    Map request=(Map)ServletActionContext.getContext().get("request");
    request.put("teaList", teaList);
}
```

---

```
        return ActionSupport.SUCCESS;
    }

    public String teaDel1()
    {

        Session session= HibernateSessionFactory.getSession();
        Transaction transaction=session.beginTransaction();

        String sql="update TTea set del='yes' where id=?";
        Query query = session.createQuery(sql);
        query.setInteger(0,id);
        int row=query.executeUpdate();
        transaction.commit();

        HibernateSessionFactory.closeSession();

        this.setMessage("操作成功");
        this.setPath("teaMana1.action");
        return "succeed";
    }

    public String teaToEdit1(){
        TTea tea=liuService.getTeaById(id);
        Map request=(Map)ServletActionContext.getContext().get("request");
        request.put("tea", tea);
        return ActionSupport.SUCCESS;
    }

    public String teaEdit1()
    {
        TTea tea=liuService.getTeaById(id);
        tea.setBianhao(bianhao);
        tea.setName(name);
        tea.setSex(sex);
        tea.setAge(age);
        tea.setLoginname(loginname);
        tea.setLoginpw(loginpw);
        tea.setDel("qiye");
```

---

```
        Session session= HibernateSessionFactory.getSession();
        Transaction transaction=session.beginTransaction();
        session.update(tea);
        transaction.commit();
        HibernateSessionFactory.closeSession();

        Map request=(Map)ServletActionContext.getContext().get("request");
        request.put("msg", "修改成功");
        return "msg";
    }
}
```

```
public String getLoginname()
{
    return loginname;
}
```

```
public void setLoginname(String loginname)
{
    this.loginname = loginname;
}
```

```
public String getLoginpw()
{
    return loginpw;
}
```

### 3.2.4 文档管理系统模块

本模块实现对任务信息的管理操作，包括任务信息的添加、删除操作。

点击文档管理信息管理菜单，进入任务信息管理页面，该页面通过列表列出已添加的任务信息，点击列表的删除按钮，实现任务信息的删除操作。如图 3-9:

具体演示如下。如图 3-9:



图 3-9 任务界面设计

### 3.2.5 修改个人密码

本模块主要是为了保证系统的安全性，管理员登录系统后，可以修改自己的登录密码。

点击修改个人管理菜单，进入修改个人密码页面，该页面显示出管理员的系统账号和密码信息，其中账号是只读的，不能进行修改，重新输入新的密码，然后点击提交按钮，即可完成修改密码操作。如图 3-10：

密码修改	
登录名:	<input type="text" value="admin"/>
原密码:	<input type="password"/>
新密码:	<input type="password"/>
确认密码:	<input type="password"/>
	<input type="button" value="修改"/> <input type="button" value="重置"/>

图 3-10 修改个人密码界面设计

### 3.2.6 通知信息管理界面

通知信息管理界面，如图 3-11：

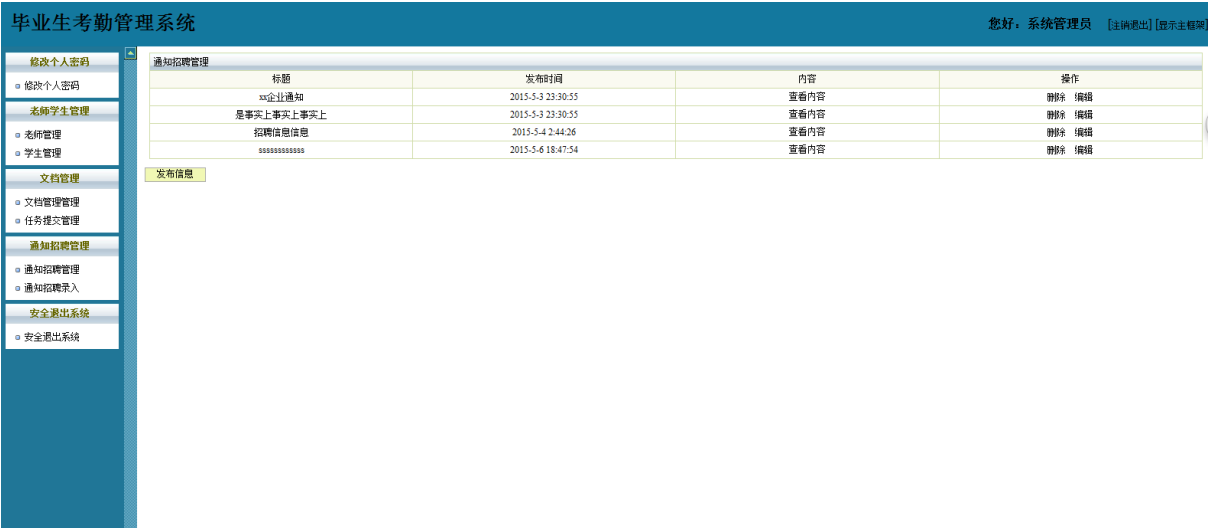


图 3-11 通知信息管理界面设计

### 3.2.7 老师管理后台界面

老师管理后台界面，如图 3-12：



图 3-12 老师管理后台界面设计

### 3.2.8 退出后台管理模块

退出后台管理模块主要用于退出后台管理系统。当管理员退出后台管理系统之后，再单击【退出】按钮时，虽还在后台中，但已不能进行任何操作；单击其他功能模块时，将弹出后台登录窗口，要求用户输入正确的用户名、密码。

### 3.2.9 学生后台管理界面

学生后台管理界面，如图 3-13：



图 3-13 学生后台管理界面设计

## 4 系统测试

系统测试是系统开发过程的重要组成部分，是用来确认一个系统的品质或性能是否符合开发之前所提出的一些要求。系统测试就是在系统投入运行前，对系统需求分析、设计规格说明和编码的最终复审，是系统质量保证的关键步骤。系统测试是为了发现错误而执行程序的过程。

在设计系统的过程中，存在一些错误是必然的。对于语句的语法错误，在程序运行时自动提示，并请求立即纠正，因此，这类错误比较容易发现和纠正。但另一类错误是在程序执行时由于不正确的操作或对某些数据的计算公式的逻辑错误导致的错误结果。这类错误隐蔽性强，有时会出现，有时又不出现，因此，对这一类动态发生的错误的排查是耗时费力的。



---

## 4.1 程序测试

测试的重要性及目的

测试的重要性:

软件的测试在软件生命周期中占据重要的地位，在传统的瀑布模型中，软件测试学仅处于运行维护阶段之前，是软件产品交付用户使用之前保证软件质量的重要手段。近来，软件工程界趋向于一种新的观点，即认为软件生命周期每一阶段中都应包含测试，从而检验本阶段的成果是否接近预期的目标，尽可能早的发现错误并加以修正，如果不在早期阶段进行测试，错误的延时扩散常常会导致最后成品测试的巨大困难。

事实上，对于软件来讲，不论采用什么技术和什么方法，软件中仍然会有错。采用新的语言、先进的开发方式、完善的开发过程，可以减少错误的引入，但是不可能完全杜绝软件中的错误，这些引入的错误需要测试来找出，软件中的错误密度也需要测试来进行估计。测试是所有工程学科的基本组成单元，是软件开发的重要部分。自有程序设计的那天起测试就一直伴随着。统计表明，在典型的软件开发项目中，软件测试工作量往往占软件开发总工作量的 40% 以上。而在软件开发的总成本中，用在测试上的开销要占 30% 到 50%。如果把维护阶段也考虑在内，讨论整个软件生存期时，测试的成本比例也许会有所降低，但实际上维护工作相当于二次开发，乃至多次开发，其中必定还包含有许多测试工作。

在实践中，软件测试的困难常常使人望而却步或敷衍了事，这是由于对测试仍然存在一些不正确的看法和错误的态度，这包括:

- 1.认为测试工作不如设计和编码那样容易取得进展难以给测试人员某种成就感;
  - 2.以发现软件错误为目标的测试是非建设性的，甚至是破坏性的，测试中发现错位是对责任者工作的一种否定;
  - 3.测试工作枯燥无味，不能引起人们的兴趣;
  - 4.测试工作是艰苦而细致的工作;
  - 5.对自己编写的程序盲目自信，在发现错误后，顾虑别人对自己的开发能力的看法。
- 这些观点对软件测试工作是极为不利的，必须澄清认识、端正态度，才可能提高软件产品的质量。

测试的目的:

如果测试的目的是为了尽可能多地找出错误，那么测试就应该直接针对软件比较复杂的部分或是以前出错较多的位置。

- 1.软件测试是为了发现错误而执行程序的过程;
- 2.测试是为了证明程序有错，而不是证明程序无错误;
- 3.一个好的测试用例是在于它能发现至今未发现的错误;
- 4.一个成功的测试是发现了至今未发现的错误的测试。

这种观点可以提醒人们测试要以查找错误为中心，而不是为了演示软件的正确功能。但是仅凭字面意思理解这一观点可能会产生误导，认为发现错误是软件测试的唯一目，查找不出错误的测试就是没有价值的，事实并非如此。

首先，测试并不仅仅是为了要找出错误。通过分析错误产生的原因和错误的分布特征，可以帮助项目管理者发现当前所采用的软件过程的缺陷，以便改进。同时，这种分析也能帮助我们设计出有针对性地检测方法，改善测试的有效性。其次，没有发现错误的测试也是有价值的，完整的测试是评定测试质量的一种方法。

---

### 测试的步骤

与开发过程类似，测试过程也必须分步骤进行，每个步骤在逻辑上是前一个步骤的继续。大型软件系统通常由若干个子系统组成，每个子系统又由若干个模块组成。因此，大型软件系统的测试基本上由下述几个步骤组成：

1.模块测试 在这个测试步骤中所发现的往往是编码和详细设计的错误。

2.系统测试 在这个测试步骤中发现的往往是软件设计中的错误，也可能发现需求说明中的错误。

3.验收测试 在这个测试步骤中发现的往往是系统需求说明书中的错误。

### 测试的主要内容

为了保证测试的质量，将测试过程分成几个阶段，即：代码审查、单元测试、集成测试、确认测试和系统测试。

#### 1.单元测试

单元测试集中在检查软件设计的最小单位—模块上，通过测试发现实现该模块的实际功能与定义该模块的功能说明不符合的情况，以及编码的错误。

#### 2.集成测试

集成测试是将模块按照设计要求组装起来同时进行测试，主要目标是发现与接口有关的问题。如一个模块与另一个模块可能有由于疏忽的问题而造成有害影响；把子功能组合起来可能不产生预期的主功能；个别看起来是可以接受的误差可能积累到不能接受的程度；全程数据结构可能有错误等。

#### 3.确认测试

确认测试的目的是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是确认测试的任务，即软件的功能和性能如同用户所合理期待的那样。

#### 4.系统测试

软件开发完成以后，最终还要与系统中其他部分配套运行，进行系统测试。包括恢复测试、安全测试、强度测试和性能测试等。

单独对系统的测试主要从以下几方面入手：

1.功能测试：测试是否满足开发要求，是否提供设计所描述的功能，是否用户的需求都得到满足。功能测试是系统测试最常用和必须的测试，通常还会以正式的软件说明书为测试标准。

2.强度测试及性能测试：测试系统能力最高实际限度，即软件在一些超负荷情况下功能实现的情况。

3.安全测试：验证安装在系统内的保护机构确实能够对系统进行保护，使之不受各种非常的干扰。针对本系统主要是对权限系统的测试和对无效数据、错数据、和非法数据干扰的能力的测试。

经过上述的测试过程对软件进行测试后，软件基本满足开发的要求，测试宣告结束。

## 4.2 测试方法

测试的方法可分三种：传统的测试方法、功能验证、系统测试。

---

传统的测试方法包括某种形式的简单单元测试，通常由开发人员执行。设计这些测试需要了解系统的内部知识，并且这些测试几乎总是针对产品的非常小的、特定的部分。这些类型的测试非常适合与其他代码组件极少交互，甚至没有交互的简单部分。

功能验证也是一种测试过程，在这个过程中，对产品源代码了解有限的设计者进行测试以确认产品或服务的核心功能。设计这种测试是为了证明这个核心功能符合某个规范。举个例子，登录时输入的邮箱错误时是不是有提示？如果测试失败，通常就意味着检测到了系统的一个基本问题。这种测试也是适合简单的 Web 服务，使您可以检查服务是否能够正确执行它的各个功能。

系统测试通常是在功能验证阶段完成，验证了核心功能后进行。它倾向于把整个系统作为一个整体来查找问题。弄清 Web 服务作为系统的一部分怎样运作，以及 Web 服务相互之间如何交互。由于系统测试是在开发生命周期快结束时才进行，所以通常不能给它分配足够的时间来完成。系统测试阶段经常被忽略，并且一些通常都可以发现的、少见的错误都不能被检测到。即使发现了这种错误，这时也来不及确定错误的原因并设法修复它们了。因此，在查找代码错误时，必需把系统测试应用设计得尽可能高效。

### 4.3 系统测试结果

在系统开发完成后，对系统进行了测试，情况如下：

- (1) 各功能模块都可以正常进行，基本实现了系统设计时的各项功能要求；
- (2) 界面简洁，操作简单，系统使用方便；

本系统的下一步开发方向如下：

- (1) 加强网站个性化设计；
- (2) 加强网站人性化服务功能；

总之，本此系统开发圆满成功，各模块运行正常。本次的设计开发为下一步的完善提供了重要的帮助和支持，将会更加提高我们的系统的功能。

---

## 总结

在毕业设计和开发期间，经过老师的指导和三个多月的认真学习，同时翻阅了大量的资料，终于完成了毕业生信息管理系统的开发设计。其功能基本符合用户需求，能够完成课题的提交。我感觉自己从理论到实践都有了长足的进步和很大的提高，使我在数据库设计方面、JSP 编程技术方面有了长足的进步，对报价系统开发过程、步骤有了比较深刻的认识。

通过毕业设计，我还明白了一点，合作是至关重要的。虽然我是自己一个人为一组，但是在设计过程中也经常和同学们商量，相互借鉴。一个完整软件程序需要若干个人的通力合作。在系统开发过程中，我加强和同学们的沟通与交流，使得对每一个问题的解决获得较优的方案，从而保证系统的合理性、完整性，为该平台的进一步完善和扩充打下了良好的基础。

撰写毕业论文期间，我深刻体会到，实践到理论的重要性。在实践到理论的转化中，我将自己的工作过程清晰的展现在面前，而且增强了自己查阅资料的能力，获得了一些必要的知识。

总之，在整个毕业设计期间，我既有收获，也发现了自己的不足。今后，我一定更加努力，弥补自己的不足，提高自己的水平。在本次毕业设计中，我从指导老师蒋建伟老师的身上学到了很多东西。蒋老师认真负责的工作态度、严谨的治学精神都使我受益匪浅。他无论在理论上还是在实践中，都给予我很大的帮助，使我得到了不少的提高，这对于我以后的工作和学习都有一种巨大的帮助，感谢他耐心的指导。

## 参考文献

- [1] 张孝祥、徐明华, JAVA 基础与案例开发详解, 清华大学出版社, 2009.
- [2] 康牧, JSP 动态网站开发实用教程, 清华大学出版社, 2009.
- [3] 刘亚宾, 精通 Eclipse--JAVA 技术大系, 电子工业出版社, 2005.

- 
- [4] 于万波,网站开发与应用案例教程,清华大学出版社,2009.
  - [5] 杨学瑜,高立军,软件开发过程与老师管理,电子工业出版社,2008.
  - [6] 张洁才,Java web 应用开发教程,科学出版社,2005.
  - [7] 戴维尔,JavaScript 程序员教程,电子工业出版社,2010.
  - [8] 黎连业、王华、李淑春,软件测试与测试技术,清华大学出版社,2009.
  - [9] 黄梯云著.管理信息系统.北京.电子工业出版社.2006 .
  - [10].James D. McCaffrey 等著.NET 软件测试自动化之道[M]. 北京.电子工业出版社.2007.
  - [11].甘仞初著.信息系统开发.北京.经济科学出版社.2007.
  - [12].马赫著. Dreamweaver MX 2004 网页设计经典 108 例.中国青年出版社.2005.
  - [13].唐友国 湛洪波著.JSP 网站开发详解.电子工业出版社 2008 年 10 月.
  - [14].卫红春等著.信息系统分析与设计[M].北京.清华大学出版社.2009.
  - [15].张大方 李玮等著.软件测试技术与管理[M].湖南.湖南大学出版社,2007.
  - [16]. (美)H.M.Deitel,P.Jdeitel 著.Java 程序设计教程(第五版). 清华大学出版社.2004.
  - [17].(美)Elliotte Rusty Harold 著.Java Network Programming.东南大学出版社 .2005.
  - [18] 孙卫琴等著.精通 struts:基于 MVC 的 Java Web 设计与开发. 电子工业出版社.2005.
  - [19] 张文静 林琪著.JSP 程序设计.人民邮电出版社.2005.
  - [20] 孙卫琴 李洪成著.Tomcat 与 Java Web 开发技术详解. 电子工业出版社.2005.
  - [21](美)John J.Patrick 著.SQL 基础(第二版).清华大学出版社.2004.
  - [22](美)David flanagan 著.Java 技术手册(第五版).东南大学出版社.2006.
  - [23]黄裴、徐汀荣,电子商务原理与技术(第二版),科学出版社,2006 年 10 月

---

## 附录 1

### 简介

**SSH: 新的 MVC 软件开发模式。**

**SSH=Struts + Spring + Hibernate**

其中, **Struts** 进行流程控制, **Spring** 进行业务流转, **Hibernate** 进行数据库操作的封装, 这种新的开发模式让我们的开发更加方便、快捷、思路清晰!

**Hibernate** 是一种持久层框架, 跟数据库打交道, 是通用的, 常用的数据库都可以连。

**Spring** 是一个开源框架, 是为了解决物业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构。分层架构允许您选择使用哪一个组件, 同时为 **J2EE** 应用程序开发提供集成的框架。

**Struts** 是一个基于 **Sun J2EE** 平台的 **MVC** 框架。主要是采用 **Servlet** 和 **JSP** 技术来实现的。由于 **Struts** 能充分满足应用开发的需求。简单易用, 敏捷迅速, 在过去的一年中最受关注。**Struts** 把 **Servlet**、**JSP**、自定义标签和信息资源(message resources) 整合到一个统一的框架中, 开发人员利用其进行开发时不用再自己编码实现全套 **MVC** 模式, 极大的节省了时间, 所以说 **Struts** 是一个非常不错的应用框架。

**Hibernate** 是一个 **JDO** 工具。它的工作原理是通过文件把值对象和数据库表之间建立一个映射关系, 这样, 我们只需要通过操作这些值对象和 **Hibernate** 提供的一些基本类, 就可以达到使用数据库的目的。例如, 使用 **Hibernate** 的查询, 可以直接返回包含某个值对象的列表(List)。而不必向传统的 **JDBC** 访问方式一样把结果集的数据逐个装载到一个值对象中, 为我们的编码工作节省了大量的劳动。**Hibernate** 提供的 **HQL** 是一种类型的 **SQL** 语言。它和 **EJBQL** 一样都是提供对象化的数据库查询方式, 但 **HQL** 在功能和使用方式上都非常接近于标准的 **SQL**。

---

## 附录 2

### Introduction

SSH: The new MVC software development model. SSH = Struts Spring Hibernate) which, Struts for process control, Spring for business transfer, Hibernate for database operations of the package, this new development model allows us to develop more convenient, fast, clear thinking! Hibernate is a persistence framework to deal with the database is a common, common database can be connected.

Spring is an open source framework to address the complexity of enterprise application development created. One of the main advantages of the framework is its layered architecture. Layered architecture allows you to choose which one component, as well as to provide an integrated J2EE application development framework.

Struts is a MVC-based framework for Sun J2EE platform. Servlet and JSP is mainly used technology to achieve. The struts can fully meet the needs of application development. Easy to use, quick fast, in the past year, the most attention. Struts the Servlet, JSP, custom tags and information resources (message resources) into a unified framework, developers do not use when developing their own code to achieve full re-MVC model, a great saving of time, so that the Struts is a very good framework.

Hibernate is a JDO tool. It works through the file to value objects and database tables to create a mapping between, so we only need to manipulate the value objects and Hibernate to provide some basic classes, we can achieve the purpose of using the database. For example, using Hibernate's query, can return a value object containing a list (List). Without having access to the same way as the traditional JDBC result set data to be loaded into a value-by-object, for our coding save a lot of work. Hibernate HQL is a class provided by the SQL language. EJBQL as it is to provide an object-oriented database query approach, but the function and use of HQL is very close to the standard SQL.