在线订餐系统的设计与实现

# 摘 要

随着社会生活水平的不断提高，人们生活的节奏加快，传统的餐饮服务已经不能满足用户的需求，网上订餐的方式受到了用户的青睐。通过在线订餐系统，为餐厅和用户搭建起一个快捷的网站服务平台，同时也为用户提供更准确更便捷的餐饮信息的平台。本系统基于B/S模式，选择面向对象的、易于扩展的Java作为编程语言，后台使用SSM三大框架和Servlet整合的形式，以MVC作为设计理念，使用MySQL数据库作为后台数据支持，通过Apache Tomcat7.0服务器运行系统。本系统主要实现了企业菜品信息的录入，菜品信息的发布，用户浏览菜品信息和下单等操作，以及评价和反馈等功能。

关键词：毕业生，SSM框架，B/S模式，订餐系统

# Design and Implementation of Online Ordering System ABSTRACT

With the continuous improvement of the living standards of society, the rhythm of people’s lives has accelerated, and traditional catering services have been unable to meet the needs of users. The way of ordering meals online has been favored by users. Through the online ordering system, a fast website service platform is set up for restaurants and users. At the same time, it also provides users with a more accurate and convenient platform for catering information. This system is based on the B/S model, and chooses object-oriented and easy-to-expand Java as the programming language. It uses the three major frameworks of SSM and Servlet integration in the background, uses MVC as the design concept, and uses MySQL database as background data support through Apache Tomcat7. .0 Server running system. This system mainly realizes the input of the company's dish information, the release of dish information, the user's operation of browsing dish information and orders, and the functions of evaluation and feedback.

**Key words：**Graduates, SSM Framework, B/S Model, Ordering system

目 录

[摘要 I](#_Toc484590964)

[ABSTRACT II](#_Toc484590965)

[1 绪论 1](#_Toc484590966)

[1.1 课题研究背景及意义 1](#_Toc484590967)

[1.2 国内外发展状况 1](#_Toc484590968)

[1.3 本文研究内容 2](#_Toc484590969)

[1.3.1 web前端语言 2](#_Toc484590970)

[1.3.2 Java Web整合开发 2](#_Toc484590971)

[1.3.3 数据库相关研究 2](#_Toc484590972)

[2 基础理论及相关技术 3](#_Toc484590973)

[2.1 SSH框架 3](#_Toc484590974)

[2.2 Servlet和JSP 3](#_Toc484590975)

[2.3 MySQL数据库 4](#_Toc484590976)

[2.4 JDK和Tomcat服务器 5](#_Toc484590977)

[2.5 Eclipse 5](#_Toc484590978)

[3 系统分析及设计 6](#_Toc484590979)

[3.1 可行性分析 6](#_Toc484590980)

[3.1.1 经济可行性 6](#_Toc484590981)

[3.1.2 技术可行性 6](#_Toc484590982)

[3.1.3 操作可行性 6](#_Toc484590983)

[3.2 需求分析 6](#_Toc484590984)

[3.3 系统总体设计 7](#_Toc484590985)

[3.3.1 系统设计目标 8](#_Toc484590986)

[3.3.2 系统功能结构 8](#_Toc484590987)

[3.4 数据库设计与实现 9](#_Toc484590988)

[3.4.1 数据库概念设计 9](#_Toc484590989)

[3.4.2 数据库表设计 11](#_Toc484590990)

[3.4.3 数据库逻辑结构设计 14](#_Toc484590991)

[4 系统详细设计 15](#_Toc484590992)

[4.1 系统环境配置 15](#_Toc484590993)

[4.1.1 软件环境 15](#_Toc484590994)

[4.1.2 硬件环境 15](#_Toc484590995)

[4.2 系统框架实现 15](#_Toc484590996)

[4.3 系统公共模块设计 17](#_Toc484590997)

[4.3.1 数据库连接和事务管理器 17](#_Toc484590998)

[4.3.2 系统安全控制 18](#_Toc484590999)

[4.3.3 模型驱动 18](#_Toc484591000)

[4.4系统主要模块功能的实现 20](#_Toc484591001)

[4.4.1登录注册功能 20](#_Toc484591002)

[4.4.2 学生主要功能 23](#_Toc484591003)

[4.4.3 企业主要功能 28](#_Toc484591004)

[4.4.4 辅导员主要功能 31](#_Toc484591005)

[4.4.5 管理员主要功能 32](#_Toc484591006)

[5 系统测试与性能分析 34](#_Toc484591007)

[5.1 测试目的 34](#_Toc484591008)

[5.2 测试方法 34](#_Toc484591009)

[5.2.1 模块测试 34](#_Toc484591010)

[5.2.2 子系统测试 34](#_Toc484591011)

[5.2.3 系统测试 34](#_Toc484591012)

[5.3 功能测试 34](#_Toc484591013)

[5.3.1 登录测试 34](#_Toc484591014)

[5.3.2 添加测试 35](#_Toc484591015)

6 [总结 37](#_Toc484591016)

[致 谢 38](#_Toc484591017)

[参考文献 39](#_Toc484591018)

# 

# 1 绪论

## 1.1 课题研究背景及意义

出门在外，很少会有人自己亲自做顿饭吃，更多的是要么吃食堂，要么去饭店，去食堂免不了排队，食堂吃上一些天就顿觉食不下咽。去饭店，跑到附近的饭店发现人很多，排队变成了坐等。随着网络逐步的渗透到我们的生活中，网上购物已成为人们消费的一种方式。随之兴起的网上订餐可以使人们借助于互联网进行饭菜的选购，餐馆配有专门的人员负责送餐，这就减轻了人们的去食堂去餐馆吃饭所占据的等待吃饭时间，以及受天气等因素的影响造成出行的不便。更重要的是餐馆的经营模式变得更加的灵活，受到地域，天气等因素的影响明显减轻了，餐馆的消费人群明显的增多了。网上订餐系统一方面使得消费者多了一种选择饭菜的方式，一方面拓展的餐馆的销售途径。

## 1.2 国内外发展状况

目前，中国餐饮O2O主要模式可以分为四大种类：团购类，以美团、百度糯米、大众点评为代表，引进美国Groupon模式，全面接入本地餐户，团购起步较早，是目前餐饮O2O的主要模式；点评类，以大众点评为代表，通过长期的积累，汇集了大量餐厅和用户的信息，起到了市场培育的作用；订餐类，以订餐小秘书为代表，帮助用户通过互联网渠道预定就餐作为、甚至餐品，提升用餐质量；外卖类，以饿了么、美团外卖外代表的第三方外卖平台是餐饮O2O领域近期最受资本青睐的O2O模式，千万级美元融资频频发生，并且众多传统餐饮和互联网巨头和纷纷布局，市场在近年来快速发展。   目前中国餐饮O2O主要模式及代表商家

   根据中国在线外卖市场现状调研分析及发展趋势报告认为，现在用户选择在线外卖平台时会考虑平台的食品安全保障，还有平台的送餐速度，还有平台的优惠活动。目前

 餐饮外卖市场保持稳定增长态势，并且目前而言餐饮外卖的互联网渗透率仍然较低，随着送餐物流的不断完善、技术进步、城市扩展等因素驱动，预计互联网餐饮外卖市场在未来5年内仍将维持高速增长的态势。

## 1.3 本文研究内容

在线订餐系统的开发主要包括前端应用程序的开发以及后台数据库的建立和维护两个方面。对于前者则要求具有应用程序功能完备，易使用等特点。而对于后者要求建立起数据一致性和完整性强、数据安全性好的数据库。

### 1.3.1 web前端语言

一个功能完备的网站，网站展示页面的美观简洁非常重要，前端仅是向用户表达所要展示的信息，所以可以使用H5和CSS等来设计页面，用Javascript和JQuery增加页面特效，用ajax实现异步刷新，来增加用户体验，当然在设计时需要注意到浏览器的兼容性问题。

### 1.3.2 Java Web整合开发

当网站的主页给用户展示出基本所要表达的内容之外，当用户选择了产品对后台服务器的请求就显得尤为重要，使用SSM框架来完成后台的制作[2]。其中SpringMVC完成页面的跳转、输入校验、文件上传以及结合jstl标签完成页面显示，Mybatis用来完成数据库中数据的持久化，Spring用来接管整个项目，创建对象并维护对象间的关系。

### 1.3.3 数据库相关研究

数据库的选择由于此系统是所有用户使用的网站系统，所以对数据的安全性和完整性要求较高，可以选择MySQL5.7数据库,结合Mybatis框架完成数据操作，可以得到较快的速度，用户的体验更好。

本系统一共分了两个系统来分别实现其功能。连个系统分别为：用户系统、后台系统。本系统设计完成后，将其安装在Server服务器上，用户可以直接使用浏览器进行查询和注册，后台数据库使用MySQL，用户界面使用JSP动态网页。

# 2 基础理论及相关技术

## 2.1 SSM框架

SSM是 SpringMVC+Spring+Mybatis的一个集成[框架](http://baike.baidu.com/item/%E6%A1%86%E6%9E%B6)，是目前比较流行的一种Web应用程序开源框架。

集成SSM框架的系统从职责上分为四层：[表示层](http://baike.baidu.com/item/%E8%A1%A8%E7%A4%BA%E5%B1%82)、[业务逻辑层](http://baike.baidu.com/item/%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91%E5%B1%82)、[数据持久层](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8C%81%E4%B9%85%E5%B1%82)和域模块层，以帮助开发人员在短期内搭建结构清晰、可复用性好、[维护](http://baike.baidu.com/item/%E7%BB%B4%E6%8A%A4)方便的[Web应用程序](http://baike.baidu.com/item/Web%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)[2]。其中使用Struts2作为系统的整体基础架构，负责MVC的分离，在Struts框架的[模型](http://baike.baidu.com/item/%E6%A8%A1%E5%9E%8B)部分，控制业务跳转，利用Hibernate框架对[持久层](http://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%B1%82)提供支持，Spring做管理，管理Struts2和Hibernate。具体做法是：用[面向对象](http://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1)的分析方法根据需求提出一些模型，将这些模型实现为基本的Java对象，然后编写基本的DAO(Data Access Objects)接口，并给出Hibernate的DAO实现，采用Hibernate架构实现的DAO类来实现[Java](http://baike.baidu.com/item/Java/85979)类与数据库之间的转换和访问，最后由Spring做[管理](http://baike.baidu.com/item/%E7%AE%A1%E7%90%86)，管理Struts2和Hibernate。Hibernate在与数据库的交互时，可以反向生成数据库表，不需手动地去建表，只要环境搭好，简历数据库后，通过配置文件并开启数据库服务，运行项目后便会将所建实体类映射到数据库中。这一方面十分便于数据库的操作。

系统的基本[业务](http://baike.baidu.com/item/%E4%B8%9A%E5%8A%A1)[流程](http://baike.baidu.com/item/%E6%B5%81%E7%A8%8B/31013)是：在[表示层](http://baike.baidu.com/item/%E8%A1%A8%E7%A4%BA%E5%B1%82)中，首先通过JSP页面实现[交互界面](http://baike.baidu.com/item/%E4%BA%A4%E4%BA%92%E7%95%8C%E9%9D%A2)，负责接收请求(Request)和传送响应(Response)，然后Struts2根据[配置](http://baike.baidu.com/item/%E9%85%8D%E7%BD%AE)文件(struts-config.xml)将ActionServlet接收到的Request委派给相应的Action处理。在业务层中，管理服务组件的Spring IoC容器负责向Action提供业务模型(Model)组件和该组件的协作对象数据处理(DAO)组件完成业务逻辑，并提供事务处理、缓冲池等容器组件以提升系统性能和保证数据的完整性。而在[持久层](http://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%B1%82)中，则依赖于Hibernate的对象化映射和数据库交互，处理DAO组件请求的数据，并返回处理结果。

采用上述开发模型，不仅实现了视图、控制器与模型的彻底分离，而且还实现了[业务逻辑层](http://baike.baidu.com/item/%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91%E5%B1%82)与持久层的分离。这样无论前端如何变化，模型层只需很少的改动，并且数据库的变化也不会对前端有所影响，大大提高了系统的可复用性。而且由于不同层之间[耦合度](http://baike.baidu.com/item/%E8%80%A6%E5%90%88%E5%BA%A6)小，有利于团队成员并行工作，大大提高了开发[效率](http://baike.baidu.com/item/%E6%95%88%E7%8E%87)。

## 2.2 Servlet和JSP

Servlet（Server Applet），全称Java Servlet。是用Java编写的[服务器端程序](http://baike.soso.com/ShowTitle.e?sp=S%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF%E7%A8%8B%E5%BA%8F)。其主要功能在于[交互式](http://baike.soso.com/ShowTitle.e?sp=S%E4%BA%A4%E4%BA%92%E5%BC%8F)地浏览和修改数据，生成动态Web内容。Servlet运行于支持Java的应用服务器中。从实现上讲，Servlet可以响应任何类型的请求，但绝大多数情况下Servlet只用来扩展基于[HTTP协议](http://baike.soso.com/ShowTitle.e?sp=SHTTP%E5%8D%8F%E8%AE%AE)的Web服务器。

JSP（全称JavaServer Pages）是由Sun Microsystems公司倡导和许多公司参与共同创建的一种使软件开发者可以响应客户端请求，而动态生成HTML、XML或其他格式文档的Web网页的技术标准。JSP技术是以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。用JSP开发的Web应用是跨平台的，既能在Linux下运行，也能在其他操作系统上运行。

JSP相比ASP和PHP，具有的优势：

（a）一次编写，到处运行。在这一点上Java比[PHP](http://baike.baidu.com/view/99.htm)更出色，除了系统之外，代码不用做任何更改。

（b）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](http://baike.baidu.com/view/2616.htm)/PHP的局限性是显而易见的。

（c）强大的可伸缩性。从只有一个小的[Jar](http://baike.baidu.com/view/50457.htm)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。

（d）多样化和功能强大的开发工具支持。这一点与ASP很像，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。

## 2.3 MySQL数据库

MySQL是一个[关系型数据库管理系统](http://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F)，由瑞典公司开发，目前属于 [Oracle](http://baike.baidu.com/item/Oracle) 旗下产品。MySQL是最流行的[关系型数据库管理系统](http://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F)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](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93)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](http://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E7%A0%81)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。

[MySQL](http://baike.baidu.com/item/MySQL/471251)是一种[开放源代码](http://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E4%BB%A3%E7%A0%81)的关系型[数据库管理](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86)系统（RDBMS）MySQL具有性价比高、灵活、广为使用和具有良好支持的特点。

（a）普及性

MySQL是一个可靠的数据库系统，无论是在嵌入式或大型群集系统的部署中，还是在基于Web的应用程序领域，用户时常会发现其实自己并不是第一个选用MySQL数据库的先驱者。

（b）简单性

与其他数据库相比，MySQL易学易用。对于MySQL数据库，无论是在开发方面，还是支持方面，现在有大量强大的工具可以选择。每一个新手开发者可以轻松地使用MySQL数据库进行开发。甚至一个有经验的Windows管理者也可以轻松部署并开始学习它，而你不需投入一分钱来了解这个数据库。MySQL可以运行的更快速，MySQL中没有多余的功能来拖累CPU或占用内存。

（c）低成本

MySQL是一个开源的数据库，MySQL对硬件的较低要求是其最大的优势之一，MySQL易学、易部署、易管理和易维护。对于现有业务，可以轻松移植到MySQL。MySQL部署迅速，因此移植过程不会导致生产中断。而且，较短的学习曲线可以让你的系统管理员迅速掌握它的运行和维护。

（d）灵活性和可扩展性

由于在MySQL中有如此众多的额外功能可选，诸如存储引擎等，这个特点使得MySQL可以根据你当前的系统的需要来进行调整。

## 2.4 JDK和Tomcat服务器

Tomcat [服务器](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=3850180&ss_c=ssc.citiao.link)是一个免费的[开放源代码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=269184&ss_c=ssc.citiao.link)的Web [应用服务器](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=4341240&ss_c=ssc.citiao.link)，是Apache软件基金会（Apache Software Foundation）的Jakarta项目中的一个核心项目，由Apache、Sun和其他一些公司及个人共同开发而成,是为了使Servlet/JSP能够与Apache服务器一起运行而开发的Servlet/JSP容器。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

JDK（Java Development Kit）是Sun Microsystems针对Java开发的产品。自从Java推出以来，JDK已经成为使用最广泛的Java SDK。JDK 是整个[Java](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=20980)的核心，包括了Java运行环境、Java工具和Java基础类库。JDK是学好Java的第一步。而专门运行在[x86](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=474226)平台的Jrocket在服务端[运行效率](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7808302)也要比Sun JDK好很多。从[Sun](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=471153)的JDK5.0开始，提供了泛型等非常实用的功能，其版本也不断更新，运行效率得到了非常大的提高。

## 2.5 Eclipse

Eclipse 是一个开放源代码的、基于Java的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java开发工具（Java Development Kit，JDK）。对于Struts2，Hibernate4，Spring4等组件，由于Eclipse并未直接包含，但可以通过下载各组件，然后将其一一安装导入Eclipse集成开发环境中，安装完后，在Eclipse中新建File下的other中即可看到有关Hibernate及Spring等的相关配置文件。

# 3 系统分析及设计

## 3.1 可行性分析

根据实际情况，我把面向某高职院校的毕业生就业管理系统的可行性分析分为三个部分来进行阐述，分别为经济可行性、技术可行性和操作可行性。

### 3.1.1 经济可行性

通过本系统可以便捷企业，毕业生和学校之间的交流沟通，为毕业生提供良好有保障的岗位信息，减少了企业招聘毕业生的流程，改变了以往学生自己在网上搜索相关企业招聘信息，辅导员查找后发链接给学生的应聘格式，学生只需通过本系统即可查看相关企业的招聘信息并予以查看了解和投递报名等。

### 3.1.2 技术可行性

本系统采用Eclipse+SSH+Tomcat开发，界面使用css+bootstraps实现，使用Mysql数据库，操作简单，易于开发，采用MVC开发模式，可以清楚的展现不同层面的相关设计，易于操作[3]。SSH整合的方式，可以更好的管理维护和分模块编写[4]。Spring主要用于对各组件的管理，比如管理Struts2、Hibernate等，Struts2主要用于对业务逻辑的控制及相关的请求响应跳转，Hibernate用户持久性数据的控制管理，以及可以更好的在连接数据库方面增加代码的复用性，通过配置文件的方式即可与数据库相连接，减少了以前传统的编码方式多次与数据库交互。通过整合后，层次明确，各模块间互不影响而能更好的交互。

### 3.1.3 操作可行性

本系统采用B/S结构，不同身份的用户可以进入相应界面进行操作，可以清晰的展现各模块的分层和功能，对用户来说操作简单明确，能快速熟悉流程。

## 3.2 需求分析

对于现在的国内的大部分就业平台多数是那种企业发布招聘信息，学生自己网上搜索并投递，通过邮件短信的方式通知后面的流程。而这种平台缺少信息的安全性，也缺乏各用户之间的即时沟通和交流。主要基于信息安全可靠，便于各用户之间的交流沟通，本系统对不同用户设计以下的功能，对于毕业生来说，他可以进行的操作有：登录注册，个人信息查看，修改，对企业所发布的招聘信息进行查看，制作个人简历，并对相应的企业的岗位给以报名，和其他各用户之间进行在线实时交流。对企业而言，主要的功能实现有以下：注册登录，查看自己相关信息和修改，发布招聘信息，并对所报名的学生的相关信息进行查看，和其他各模块用户之间的在线实时交流。对辅导员而言，主要功能如下：注册登录，对学生信息进行审核给予相应的处理，对相关报名学生数据的查看并生成Excel报表的样式，和不同模块人员进行实时沟通。对管理员而言，管理数据库，管理企业，学生及辅导员账户信息，对企业所发布的招聘信息进行核实，对报名信息进行查看，和其他用户的在线实时交流。

## 3.3 系统总体设计

就业管理系统采用了B/S应用方式实现相关管理。对于Java语言或JSP，在实现数据库操作时，可以采用将数据库表与普通的Java类映射，将数据表转换为类（对象），然后利用对象实现对数据库的操作。所采用的MVC工作流程如图3-1所示。

struts-conf.xml

WEB

服务器

数

据

库

（模型M）

JavaBean

DAO

(视图V)

JSP

(控制器C)

ActionServlet

浏览器

图3-1 MVC模式图

浏览器发送请求到服务器，struts-conf.xml配置文件将控制器ActionServlet接收到的请求委派给相应的Action,调用模型及Dao层，从而访问数据库，将相应的页面响应到服务器。

浏览器端使用JSP页面作为表现层载体，显示网页信息，网页与数据库所在数据层和业务层接口所在业务逻辑层分离，便于开发和维护[5]。浏览器接收到访问JSP页面的请求，执行JSP代码显示页面[6]。在用户通过点击页面上的按钮或者连接及填写表单等动作后，JSP根据需求执行JavaScript代码验证后才将网页信息传入Action进行处理。网页中使用了JavaScript能够实现表单信息的初步验证，保证了表单信息的可用性，节省了大量时间，提高系统效率。

接收到自客户端的请求，根据需要调用Action处理请求，将客户端的请求内容传递给业务逻辑层，并返回结果到客户端表现层[7]。Action主要工作是负责业务逻辑跳转操作，完成业务从表现层到业务逻辑层的传递。表现层和业务逻辑层的分离，实现了表现层和数据库的松散耦合。

从Action获取表现层的请求动作，完成于数据库的接洽[8]。它们的主要功能是提供从请求中取得参数的值，转化成相应的模型，再将模型传递给执行业务操作的对象，由业务对象执行业务操作，再返回执行的结果。

数据层使用MySQL数据库，通过前台页面的传值在后台进行对数据库的增加、删除、修改、查询等工作，保证了系统功能的安全性和稳定性。

### 3.3.1 系统设计目标

（a）简单清晰，易于操作。

（b）实现实时在线交流。

（c）提供个人信息修改

（d）实现分页查看,结构清楚展示在用户界面。

### 3.3.2 系统功能结构

面向某高职院校的毕业生就业管理系统功能如下图3-2所示：

图3-2 高职院校就业管理系统

各个模块功能介绍如下：

毕业生模块:毕业生注册登录、个人信息修改、查看招聘信息、制作个人电子简历、报名参加单位的招聘、在线咨询等六部分。

企业模块:企业注册登录、企业信息修改、发布招聘信息、报名信息查询和在线咨询等五个功能。

辅导员模块:注册登录、审核学生信息、就业信息查询、在线咨询等四个功能。

管理员模块:信息审核发布、管理报名信息、账号管理、数据库管理、在线咨询等五个功能。

## 3.4 数据库设计与实现

系统的开发过程中，对数据的操作必不可少，所以数据库的设计就尤为重要，设计过程中需要明确表之间的关联关系。

### 3.4.1 数据库概念设计

本系统有多个模块，各模块需要有相应的实体，各模块之间的联系也需要有相应的实体类来支撑。下面主要介绍本系统中所用到的实体。

（a）毕业生信息实体



图3-3 毕业生信息实体

（b）企业信息实体

图3-4 企业信息实体

（c）辅导员信息实体



图3-5 辅导员信息实体

（d）管理员信息实体



图3-6 管理员信息实体

（e）简历信息实体



图3-7 简历信息实体

（f）职位信息实体



图3-8 职位信息实体

（g）报名信息实体



图3-9 报名信息实体

（h）在线聊天用户信息实体

图3-10 在线聊天用户信息实体

### 3.4.2 数据库表设计

（a）学生信息表

表3-1 学生信息表student

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| ID | INT(11) | No | 学生ID |
| S\_NUM | VARCHAR2(225) | No | 学生学号 |
| NAME | VARCHAR2(225) | No | 学生姓名 |
| PWD | VARCHAR2(225) | No | 学生登陆密码 |
| C\_NUM | VARCHAR2(225) | No | 学生身份证号 |
| POLITSTAT | VARCHAR2(225) | No | 政治面目 |
| SCHOOL | VARCHAR2(225) | No | 学校 |
| MAJOR | VARCHAR2(225) | No | 学生专业 |
| DEGREE | VARCHAR2(225) | No | 学生学位 |
| ADDRESS | VARCHAR2(225) | No | 住址 |
| PHONE | VARCHAR2(225) | No | 联系电话 |
| EMAIL | VARCHAR2(225) | No | 邮箱 |

（b）企业信息表

表3-2 企业信息表enterprise

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| ID | INT(11) | No | 企业用户ID |
| ENTERNAME | VARCHAR2(225) | No | 企业名 |
| EPNAME | VARCHAR2(225) | No | 企业用户名 |
| PASSWORD | VARCHAR2(225) | No | 企业用户密码 |
| PHONE | VARCHAR2(225) | No | 企业用户电话 |
| EMAIL | VARCHAR2(225) | No | 企业邮箱 |
| CONTEXT | VARCHAR2(225) | No | 企业简介 |

（c）辅导员信息表

表3-3 辅导员信息表assist

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| ID | INT(11) | No | 辅导员用户ID |
| ASSISTNAME | VARCHAR2(225) | No | 辅导员名 |
| JOBNUM | VARCHAR2(225) | No | 辅导员工号 |
| PASS | VARCHAR2(225) | No | 辅导员用户密码 |

（d）管理员表

表3-4 管理员表admin

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| ID | INT(11) | No | 管理员ID |
| ADMINNAME | VARCHAR2(225) | No | 管理员名 |
| ADMINPASS | VARCHAR2(225) | No | 管理员密码 |

（e）岗位信息表

表3-5 岗位信息表position

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| ID | INT(11) | No | 岗位ID |
| ENTERPRISE\_ID | INT(11) | No | 企业ID |
| PNAME | VARCHAR2(225) | No | 岗位名称 |
| TYPE | VARCHAR2(225) |  | 岗位类型 |
| PDNUM | VARCHAR2(225) | No | 招聘数量 |

续表3-5

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| DUTY | VARCHAR2(225) | No | 岗位职责 |
| DEMAND | VARCHAR2(225) | No | 岗位需求 |
| PLACE | VARCHAR2(225) | No | 工作地点 |
| DATE | VARCHAR2(225) |  | 发布日期 |

（f）简历表

表3-6 简历表resume

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| ID | INT(11) | No | 简历ID |
| NAME | VARCHAR2(225) | No | 简历名称 |
| PICFILENAME | VARCHAR2(225) | No | 照片名 |
| CONTEXT | VARCHAR2(225) | No | 项目经验 |
| STUDENT\_ID | INT(11) | No | 学生ID |

（g）报名表

表3-7 报名表designform

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| ID | INT(11) | No | ID |
| STUDENT\_ID | INT(11) | No | 学生ID |
| ENTERPRISE\_ID | INT(11) | No | 企业ID |
| RESUME\_ID | INT(11) | No | 简历ID |

（h）聊天室登陆用户信息表

表3-8 聊天室登陆用户信息表chartuser

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可为空 | 说明 |
| ID | INT(11) | No | ID |
| USERNAME | VARCHAR2(225) | No | 用户名 |
| PASSWORD | VARCHAR2(225) | No | 登陆密码 |

### 3.4.3 数据库逻辑结构设计

数据库概念设计中已经分析了学生、企业、管理员、辅导员等主要的数据实体对象，这些实体对象是数据表结构的基本模型，最终的数据模型都要实施到数据库中，形成整体的数据结构，可以使用PowerDesigner工具完成这个数据库的建模。

物业管理系统数据库模型如图3-11所示。

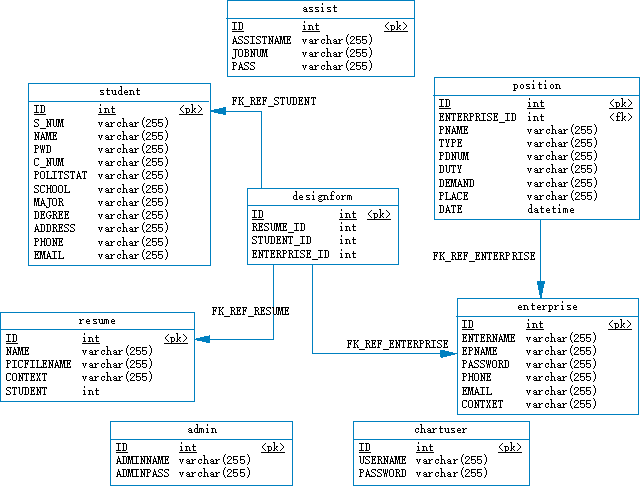


图 3-11 数据库逻辑结构

# 4 系统详细设计

## 4.1 系统环境配置

### 4.1.1 软件环境

本系统采用Apache下的开源项目Tomcat7.0作为服务器，JDK版本为7.0，使用MYSQL5.7作为后台数据库的支持，使用Struts2、Hibernate4、Spring4、JSP技术作为后台服务端的开发，浏览器使用谷歌、IE浏览器。具体各部分环境配置如下：

（a）Java开发工具包：jdk-7u80-windows-x64。JDK环境变量的配置如下：

变量名JAVA\_HOME：D:\Program Files\Java\jdk1.7.0（为JDK的安，装目录）

变量名Path：%JAVA\_HOME%\bin

变量名 ClassPath：.;%JAVA\_HOME%\lib\tools.jar;%JAVA\_HOME%\lib\dt.jar

（b）Web服务器采用Tomcat7.0.5。Tomcat7.0环境变量配置如下：

变量名CATALINA\_HOME：D:\program\apache-tomcat-7.0.54（为Tomcat存放目录）

变量名Path：%CATALINA\_HOME%\lib;%CATALINA\_HOME%\lib\servlet-api.jar;

%CATALINA\_HOME%\lib\jsp-api.jar;

（c）数据库服务器采用MYSQL5.7。

（d）开发工具采用的是Eclipse,解压Struts2、Hibernate、Spring4的源码包，导入相关jar包到项目的WEBCONTEXT/WEB\_INF/bin文件下，并导入。

（e）使用Dreamwear8.0，EditPlus美化页面。

### 4.1.2 硬件环境

（a） 计算机型号 lenovo。

（b） CPU处理器，AMD A10-5750M APU with Radeon(tm)HD Graphics 2.50 GHz。

（c） 磁盘容量，硬盘1T,内存4.00GB。

（d） 操作系统 Windows8.1专业版 64 位。

## 4.2 系统框架实现

（a）Model(VO)层

表4-1 Model层主要类

|  |  |  |
| --- | --- | --- |
| 类名 | 映射表 | 说明 |
| Student | student | 毕业生信息 |
| Enterprise | enterprise | 企业信息 |
| Assist | assist | 辅导员信息 |
| Admin | admin | 管理员信息 |

续表4-1

|  |  |  |
| --- | --- | --- |
| 类名 | 映射表 | 说明 |
| Resume | resume | 简历信息 |
| Position | position | 岗位信息 |
| DesignForm | designform | 报名信息 |
| ChartUser | chartuser | 聊天室用户 |
| PageBean |  | 用于分页 |

（b）持久层

表4-2 业务逻辑主要类

|  |  |  |
| --- | --- | --- |
| 类名 | 实现类 | 说明 |
| StudentService | StudentDao | 毕业生操作 |
| EnterpriseService | EnterpriseDao | 企业操作 |
| AssistService | AssistDao | 辅导员操作 |
| AdminService | AdminDao | 管理员操作 |
| ResumeService | ResumeDao | 简历操作 |
| PositionService | PositionDao | 岗位信息操作 |
| ChartService | ChartDao | 用户聊天操作 |

（c）业务逻辑层Action和Servlet类

表4-3 Action,Servlcet

|  |  |  |
| --- | --- | --- |
| 类名 | 父类 | 说明 |
| StudentAtion | ActionSupport | 毕业生操作 |
| EnterpriseAction | ActionSupport | 企业操作 |
| AssistAction | ActionSupport | 辅导员操作 |
| AdminAction | ActionSupport | 管理员操作 |
| ResumeAction | ActionSupport | 简历操作 |
| PositionAction | ActionSupport | 岗位信息操作 |
| ServletProxy | HttpServlet | 代理类 |
| BaseServlet | HttpServlet | 基类方法调用 |
| UserServlet | BaseServlet | 用户聊天操作 |

## 4.3 系统公共模块设计

### 4.3.1 数据库连接和事务管理器

本系统通过在application.xml文件中配置C3P0数据源连接数据库，具体代码如下：

（a）配置 C3P0 数据源

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">

<property name="user" value="${jdbc.user}"></property>

<property name="password" value="${jdbc.password}"></property>

<property name="driverClass" value="${jdbc.driverClass}"></property>

<property name="jdbcUrl" value="${jdbc.jdbcUrl}"></property>

<property name="initialPoolSize"

value="${jdbc.initPoolSize}"></property>

<property name="maxPoolSize" value="${jdbc.maxPoolSize}"></property>

</bean>

（b）配置会话工厂SessionFactory，并将数据源dataSource通过属性注入进去

<bean id="sessionFactory"

class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">

<property name="dataSource" ref="dataSource"></property>

<property

name="configLocation" value="classpath:hibernate.cfg.xml"></property>

<property

name="mappingLocations" value="classpath:com/lc/entities/\*.hbm.xml"></property>

</bean>

（c）配置 Spring 的声明式事务 ，配置 Hibernate 的事务管理器

<bean id="transactionManager"

class="org.springframework.orm.hibernate4.HibernateTransactionManager">

<property name="sessionFactory" ref="sessionFactory"></property>

</bean>

（d）配置事务属性

<tx:advice id="txAdvice" transaction-manager="transactionManager">

<tx:attributes>

<tx:method name="get\*" read-only="true" />

<tx:method name="\*" />

</tx:attributes>

</tx:advice>

（e）配置事务切入点, 再把事务属性和事务切入点关联起来

<aop:config>

<aop:pointcut expression="execution(\* com.lc.service.\*.\*(..))"

id="txPointcut" />

<aop:advisor advice-ref="txAdvice" pointcut-ref="txPointcut" />

</aop:config>

### 4.3.2 系统安全控制

本系统在登录时，通过登录主页面中的下拉框选择不同的身份，通过输入相应的表单信息，工号身份证号，学号，密码等账户信息，并输入页面随机生成的校验码来验证是否存在该用户信息，若输入无误，则可成功登录进入相应操作页面，否则说明数据库无该信息记录，页面不跳转，重新输入直到成功登录。

### 4.3.3 模型驱动

所谓模型驱动即使用javaBean对象来封装请求参数，在整个MVC设计流程中可以直接使用javaBean对象[9]。在Struts2中就是将Action与Model相分离，代码看起来更加美观，使用起来不再繁琐，累赘。

Struts2中模型驱动实现方式通过配置文件：

在struts-default.xml中有如下配置：

<interceptors>

<!—定义属性驱动的拦截器—>

<interceptor name=”params” class=”com.opensymphony.xwork2.interceptor.Parameterslnterceptor”/>

……

<!—定义模型驱动的拦截器—>

<interceptor name=”model-driven” class=”com.opensymphony.xwork2.interceptor.ModelDrivenlnterceptor”/>

……

<!—定义模型驱动的拦截器栈—>

<interceptor-stack name=”modelDrivenStack”>

<interceptor-ref name=”model-driven”/>

<interceptor-ref name=”basicStack”/>

</interceptor-stack>

……

<interceptor-stack name=”defaultStack”>

<interceptor-ref name=”model-driven”/>

<interceptor-ref name=”params”/>

</interceptor-stack>

</interceptors/>

<!—定义Struts2m默认拦截器栈—>

其中params拦截器负责提取请求的参数，如果是使用的属性驱动模式，它还负责将请求参数传给Action实例的属性，而model-driven拦截器则负责把请求参数传给模型的属性。

以StudentAction为例，StudentAction需实现ModelDriven<Student>接口，具体代码片段如下：

private Student student;

public Student getModel() {

if (id == null) {

student = new Student();

} else {

student = studentService.get(id);

}

return student;

}

这样的好处在于，当Student对象中的属性过多时，页面需要的参数都得在Action中给以getter()，setter()方法，这样使得Action更加臃肿，不便维护。

模型驱动的用法及好处以示例讲述。

实体类:class Entity{

private Integer id;

private String name;

private String pass;

……

省略getter,setter方法

}

Action:

方式一:

Class: EneityAction extends ActionSupport {

private Entity entity;

getter,setter方法;

}

JSP页面:

<s:textfield name="entity.name"></s:textfield>

方式二:

Class: EneityAction extends ActionSupport implements<Entity>{

private Entity entity;

@Override

public Entity getModel() {

if (id == null) {

entity = new Entity ();

}

return entity;

}

}

JSP页面:

<s:textfield name=" name"></s:textfield>

## 4.4系统主要模块功能的实现

### 4.4.1登录注册功能

用户根据下拉选择框选择相应的身份来进行登录，只有输入正确的用户名密码或工号身份证号等才可登录成功，对于新用户可进行注册，点击注册按钮进入相应的注册界面进行注册，注册成功后，会在数据库中生成对应的记录，并跳转至登录界面进行登录。如图4-1，图4-2 所示。具体实现代码以毕业生为例，分别编写DAO，业务逻辑层。

（a）DAO层

根据毕业生学号和身份证号来进行登陆判断，若返回唯一的结果，则说明数据库存在该用户。由于使用的SSH框架整合的方式[10]，所以舍弃了以往的SQL语句，此处尝试使用HQL语句来进行数据的查找。

public Student login(String s\_num, String c\_num) {

String sql = "from Student s where s.s\_num=:s\_num and s.c\_num=:c\_num";

Student student = (Student) getSession().createQuery(sql).setParameter("s\_num", s\_num)

.setParameter("c\_num", c\_num).uniqueResult();

return student;

}

（b）Service层

调用DAO层的登录方法login。

public Student login(String s\_num, String c\_num) {

return studentDao.login(s\_num, c\_num);

}

（c）Action层

public String slogin() {

student = studentService.login(student.getS\_num(), student.getC\_num());

//下面就是将session中保存验证码字符串与客户输入的验证码字符串对比了

String random = (String) ActionContext.getContext().getSession().get("authCode");

if (student != null && random.equals(this.getRand())) {

ActionContext.getContext().getSession().put("name", student.getName());

ActionContext.getContext().getSession().put("student", student);

return "success";

}

this.addActionError("用户名不存在或密码不正确或校验码输入有误");

return "input";

}

struts.xml文件中配置业务跳转的视图资源，如果登录成功即返回success字符串对应的页面，否则登录失败为登陆首页。

<action name="slogin" class="studentAction" method="slogin">

<result name="success">/WEB-INF/views/student/success.jsp</result>

<result name="input">/index.jsp</result>

</action>

毕业生注册方法类，此处使用SSH框架，保存更新方法由Hibernate框架封装好配置相应环境即可直接调用。  
 public void register(Student student) {

getSession().saveOrUpdate(student);

}

在注册时需对用户进行判断，根据身份证号来进行保障用户的独一无二性。

public boolean StuIsExixt(String s\_num) {

boolean flag = true;

String sql = "from Student as s where s.c\_num = '" + c\_num + "' ";

List<Student> student = getSession().createQuery(sql).list();

if (student.size() == 0) {

flag = false;

}

return flag;

}

public void register(Student student) {

studentDao.register(student);

}

public boolean StuIsExixt(String s\_num) {

return studentDao.StuIsExixt(s\_num);

 }

图4-1 系统登录页面



图4-2 学生注册页面

在Action首先根据boolean型来判断用户是否已存在，若为true,说明该用户已经存在，则返回input字符串，否则进行注册，即通常的添加操作。

public String sregister() {

boolean flag = studentService.StuIsExixt(student.getS\_num());

if (flag) {

this.addActionError("该用户已存在，请重新输入!");

return "input";

}

studentService.register(student);

request.put("s\_num", student.getS\_num());

return "register";

}

### 4.4.2 学生主要功能

学生登录成功后，进入毕业生主页面，可以对个人信息查看，修改，制作简易的个人简历，并可查看简历内容，进入聊天室后，进行注册聊天室用户信息成功即可登录聊天室，和其他用户进行在线实时聊天。除此之外，学生可对审核后发布的企业招聘信息进行浏览查看，并根据个人意愿进行相应的报名投递。毕业生用户在查看相关企业所发布的岗位信息后，可进行报名操作，其主要思路是将所选中的信息存放在session中，然后将所需信息数据从session中取出并回显到表单数据中，最后做保存操作即可将报名信息保存到后台数据库中，达到持久化的作用[12]。如图4-3，图4-4，图4-5，图4-6所示。



图4-3 学生相关功能页面



图4-4 职位信息查询



图4-5 职位信息查询



图4-6 职位信息查询

用户登录成功进入操作页面时，对于学生信息列表，企业信息列表，所发布的招聘信息列表等显示问题上，由于信息记录过多，页面无法全部清晰显示信息，这时通过分页功能即可使用户浏览更加便捷流畅[12]。因此，创建PageBean来对分页信息进行封装，在PageBean中将每页要显示的信息列表放进List集合中，页面只需将List集合中后台传过来的数据使用<s:iterator />标签迭代显示出来即可。

public static int curPage(int page){

return page == 0 ? 1 : page;

}

计算当前页开始显示的记录

public static int curOffset(int pageSize,int curPage){

return pageSize \*(curPage -1);

}

根据总记录数获取总页数

public static int getSumPages(int allRowCounts,int pageSize){

return (allRowCounts-1)/pageSize+1;

}

按条件的分页查询的具体实现如下：

（a）DAO层代码

public int findAllRowCount(String hql,String str) {

return getSession().createQuery(hql).setParameter("str", str).list().size();

}

public List<Position> findByCondition(String hql, int off, int len,String str) {

Query query = getSession().createQuery(hql);

query.setFirstResult(off);

query.setMaxResults(len);

query.setParameter("str",str);

List<Position> positions = query.list();

if (positions != null && positions.size() > 0) {

return positions;

}

return null;

}

（b）Service层分页查询，curentPage当前第几页，pageSize每页大小，封装了分页信息包括记录List的Bean如下：

public PageBean queryForCondition(int pageSize, int page,String str) {

String hql = "from Position p LEFT OUTER JOIN FETCH p.enterprise where p.type like :str or p.place like :str";

int allRowCounts = positionDao.findAllRowCount(hql,str);

int curPage = PageBean.curPage(page);

int curOffset = PageBean.curOffset(pageSize, curPage);

int sumPages = PageBean.getSumPages(allRowCounts, pageSize);

List<Position> positions = positionDao.findByCondition(hql, curOffset, pageSize,str);

PageBean pageBean = new PageBean();

pageBean.setAllRowCounts(allRowCounts);

pageBean.setCurPage(curPage);

pageBean.setSumPages(sumPages);

pageBean.setList(positions);// 每一页应该显示的记录

return pageBean;

}

（c）Action层

private int page;// 接收从页面传递过来的当前页数

// 默认为1表示默认显示第一页的内容

private PageBean pageBean;

private List<Position> positions;

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public PageBean getPageBean() {

return pageBean;

}

public void setPageBean(PageBean pageBean) {

this.pageBean = pageBean;

}

public List<Position> getPositions() {

return positions;

}

public void setPositions(List<Position> positions) {

this.positions = positions;

}

private String str;

public String getStr() {

return str;

}

public void setStr(String str) {

this.str = str;

}

public String condition2() {

return "success";

}

// 带条件的分页查询

@SuppressWarnings("unchecked")

public String pageQueryList2() {

int limit =3;

System.out.println("第:" + page + "页");

System.out.println("STR:" + str);

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

pageBean = positionService.queryForPage(limit, page);

} else {

pageBean = positionService.queryForCondition(limit, page, str);

}

positions = pageBean.getList();

ActionContext.getContext().getSession().put("positions", this.positions);

ActionContext.getContext().getSession().put("pageBean", this.pageBean);

return "pageQueryList2";

}

在上面的代码中，将页面表单传来的字符串str进行判断，若为空或空串，则点击查询按钮后调用查找所有审核发布后的招聘信息，若不为空，即用户输入相应的字段属性，比如工作地点，岗位类型来进行查找，查找到的信息必须满足所含字段[13]。其中limit

用来设置每页中显示的记录数。下面的System.out.println("STR:" + str)用来打印查询过程中str的值是否与表单输入一致，由于之前并未准确找到异常或问题所在，分页功能一直没能够正确显示，此处通过打印信息的中文编码格式可以定位异常所在。当str的值为乱码格式时，需将Tomcat服务器Server.xml中Connector标签中配置URIEncoding设置为UTF-8格式，即可解决中文乱码问题，并一同解决分页功能下一页不能正确显示。

### 4.4.3 企业主要功能

企业用户登录成功后进入功能操作页面，可对自己企业信息进行查看修改，发布招聘信息，对已发布的招聘信息进行查看浏览，并及时进行修改，删除。由于考虑到记录信息过多，一页无法正常显示，使用分页功能来进行查看。此外，可对已报名的信息进行查看从而给予后序的安排。最后是和学生用户一样进行注册登录聊天室，和其他用户进行在线实时聊天，以及时获取更多信息。如图4-7所示。



图4-7 企业相关功能页面

具体代码如下：

// 查看企业信息

public String scaninfo() {

return "scaninfo";

}

// 回显信息

public String epedit() {

return "epedit";

}

// 更新企业信息

public String epupdate() {

HttpSession session = ServletActionContext.getRequest().getSession();

if (session.getAttribute("enterprise") == null) {

this.addActionError("对不起，请您先登录");

return INPUT;

}

enterpriseService.update(enterprise);

ActionContext.getContext().getSession().put("enterprise", enterprise);

return "update";

}

//进入信息查看页面

public String condition() {

return "success";

}

private String str;

public String getStr() {

return str;

}

public void setStr(String str) {

this.str = str;

}

//查看当前企业发布的招聘信息

public String pageQuery2() throws UnsupportedEncodingException {

Enterprise ep = (Enterprise)

ActionContext.getContext().getSession().get("enterprise");

if (ep == null) {

this.addActionError("对不起，请您先登录");

return INPUT;

} else {

int limit = 3;

System.out.println("第:" + page+"页");

System.out.println("STR:" + str);

System.out.println("ID:" + ep.getId());

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

pageBean = enterpriseService.queryForPage(limit, page,

ep.getId());

} else {

pageBean = enterpriseService.queryForCondition(limit, page, ep.getId(),

str);

}

positions = pageBean.getList();

ActionContext.getContext().getValueStack().set("positions", positions);

ActionContext.getContext().getValueStack().set("pageBean", pageBean);

return "pageQuery2";

}

}

### 4.4.4 辅导员主要功能

辅导员用户登录成功后，可查看相关学生信息，并进行修改，对报名信息进行查看浏览，通过注册登录聊天室进行在线咨询聊天，及时和其他身份用户进行沟通交流。在查看就业信息时，可对信息进行统计以Excel报表的格式下载下来或查看报表内容。如图4-8所示。



图4-8 辅导员功能操作页面

辅导员的查看就业报信信息及生成Excel报表，首先在Action类初始化输出流变量和所要下载的文件名，并设置getter，setter方法，创建Excel对象，并设置行对象，将表单中所需的数据封装到集合里，最后将其迭代进去，通过字节输出流写入文件。具体代码实现如下：

private InputStream excelStream; // 输出流变量

private String excelFileName; // 下载文件名

public InputStream getExcelStream() {

return excelStream;

}

public void setExcelStream(InputStream excelStream) {

this.excelStream = excelStream;

}

public String getExcelFileName() {

return excelFileName;

}

public void setExcelFileName(String excelFileName) {

this.excelFileName = excelFileName;

}

//导出Excel报表

public String excel() throws IOException {

List<DesignForm> oblist = studentService.getAllObInf();

HSSFWorkbook workbook = new HSSFWorkbook();

HSSFSheet sheet = workbook.createSheet();

HSSFRow row\_title = sheet.createRow(0);

HSSFCell cell\_title1 = row\_title.createCell(0);

HSSFCell cell\_title2 = row\_title.createCell(1);

HSSFCell cell\_title3 = row\_title.createCell(2);

cell\_title1.setCellValue("学生姓名");// 学生姓名

cell\_title2.setCellValue("企业名称");// 企业名称

cell\_title3.setCellValue("求职意向");// 求职意向

…(此处其他字段省略)

for (int i = 0; i < oblist.size(); i++) {

DesignForm form = oblist.get(i);

HSSFRow row = sheet.createRow(i + 1);

HSSFCell cell1 = row.createCell(0);

HSSFCell cell2 = row.createCell(1);

HSSFCell cell3 = row.createCell(2);

cell1.setCellValue(form.getStudent().getName());

cell2.setCellValue(form.getEnterprise().getEnterName());

cell3.setCellValue(form.getResume().getName());

}

ByteArrayOutputStream os = new ByteArrayOutputStream();

workbook.write(os);

byte[] fileContent = os.toByteArray();

ByteArrayInputStream is = new ByteArrayInputStream(fileContent);

excelStream = is; // 文件流

excelFileName = "report.xls"; // 设置下载的文件名

return "excel";

}

struts.xml配置文件

<!-- 导出Excel -->

<action name="excel" class="assistAction" method="excel">

<result type="stream" name="excel">

<!-- 下载文件的类型 -->

<param name="contentType">application/octet-stream</param>

<!-- 返回流 excelStream为action中的流变量名称 -->

<param name="inputName">excelStream</param>

<!--

fileName指定生成的文件名字(适合动态生成文件名)为action中变量

-->

<param name="contentDisposition">

attachment;filename=${excelFileName}

</param>

<param name="bufferSize">1024</param>

</result>

</action>

### 4.4.5 管理员主要功能

管理员用户登录成功后，可对相关信息进行查看编辑和删除，对企业所发布的招聘信息进行审核，和其他用户进行在线咨询沟通实现代码与前面模块类似，通过注册聊天室用户并登录即可与已进入聊天室的其他用户进行一对一的交流。如图4-9，图 4-10所示。



图4-9 管理员操作页面



图4-10 管理员操作页面

# 5 系统测试与性能分析

## 5.1 测试目的

软件测试是为了发现程序中的错误而执行程序的过程，是软件开发周期中非常重要的一步，好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案。正确认识测试是为了找出错误而不是为了证明程序的正确性是非常必要的，只有这样才能真正的测试好开发的程序。测试阶段的根本目标是尽可能地发现并排除软件隐藏的错误，最终把一个高质量的软件系统交给用户使用[16]。系统的测试方法有两种：黑盒测试与白盒测试。其中，黑盒测试是不考虑代码，仅对其功能进行测试的方法；白盒测试是代码测试。本系统主要采用黑盒测试。

## 5.2 测试方法

### 5.2.1 模块测试

每个模块完成一个清晰定义的子功能，而且这个子功能和同级其他模块的功能之间没有相互依赖关系。因此把每个模块作为一个单独的实体进行测试，比较容易检验处模块是否有错误。在这个过程中发现的往往是编码与详细设计的错误。

### 5.2.2 子系统测试

子系统的测试是把经过单元测试的模块放在一起形成一个子系统来测试。模块之间的协调与通信是这个测试中的主要问题，而这部分测试着重测试模块的接口。

### 5.2.3 系统测试

把所有经过测试的子系统装配成一个完整的体系来测试。在这个过程中不近应该发现设计与编码的错误，还应该验证系统确实能提供需求说明书中指定的功能。在这部分测试中发现的往往是软件设计中的错误，也有可能发现需求说明书中的错误。

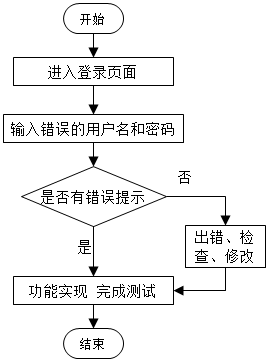
## 5.3 功能测试

功能测试，也称黑盒测试，是对产品的功能特性与操作行为进行测试的方法。以下对部分功能进行测试，列出部分测试结果。主要涉及的功能模块有：用户登录测试、添加测试。

### 5.3.1 登录测试

该测试主要用于验证系统是否能正确的处理用户登录时遇到的各种情况。用户名和密码输入为空时可通过页面上的Javascript函数检测出来，不需要提交到Action处理，节省系统的开销。

登录测试用例的测试过程如图5-1所示。

图5-1 登录模块测试图

系统中，通过index.jsp登录页面输入信息后提交到Action处理，判断是否为合法用户，若登录不合法则给出提示信息，登录合法时根据用户的类型跳转到指定的页面。输入错误的用户名或者密码之后提示错误，并返回到登录页面。

### 5.3.2 添加测试

该测试主要用于测试添加新用户行为。测试主要集中在添加新用户信息以及修改个人信息时的测试类似此处。该测试的主要内容包括:身份证号码填写、电话号码、邮箱格式是否正确。增加用户信息的测试内容如表5-1所示。

表5-1 增加新用户测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 测试用例说明 | 测试数据 | 期望结果 | 测试结果 |
| 身份证 | 长度、格式不正确 | 身份证号码 | 正确不提示，错误提示 | 格式不正确无法存入数据库 |
| 电话 | 长度、格式不正确 | 电话号码 | 正确不提示，错误提示 | 格式不正确无法存入数据库 |
| 邮箱 | 长度。格式不正确 | 邮箱 | 正确不提示，错误提示 | 格式不正确无法存入数据库 |

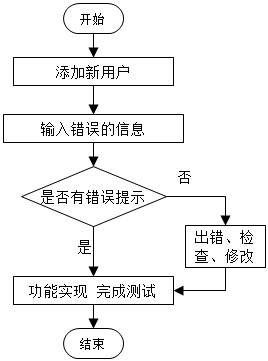
上述测试用例的测试过程如图5-2所示。

图5-2 添加新用户测试图

# 6 总 结

通过本次毕业设计，我收获了许多，也感到有好多遗憾。首先，在拿到题目时，我有点胆怯，看完了整个项目需求，结合自身所掌握的熟练度和所了解的多少，我感觉凭借自己是难以完成的，对于项目本身而言，它是一个网站类题目，对于网站可以通过多种开发语言技术进行开发，我本身学的是Java编程语言，对Struts2,Hibernate,Spring有些许学习和了解，对于早前的Servlet有很多遗忘所以选择使用SSH三大框架整合类完成。而一个完美成型的网站最重要的两部分就是成熟稳定的后台技术，和美观高大上的前端技术，我只知其一，而不知其二。决定先把大体的框架完成起来，美化后面再说，通过查看相关文献资料搭起主要环境，在编写的过程中，好多东西都是以前从未深入地去学习理解过，所以开始编写时进度很慢，每遇到一个问题，就会纠结持续好久，不断地在网上查找资料和问题的导致原因，去一点一点地排错和编写，一个问题经过一天两天甚至更多时间都未能解决，好在自己坚持下来，大体完成架构功能。在后期的编写中，问题也是反反复复，不断的查BUG并解决，并适当的通过查看学习将自己所设计的页面美化了下，比起开始能好许多。通过本次毕业设计网站的开发，我意识到自己对所需知识不扎实牢固，遗忘过多，对相关技术的深度学习十分欠缺。但正因为这些因素，在编写的过程中，我通过不断的学习来解决这些自己从未解决的难点，也为自己的基础奠定了更多的积累和经验。

# 致 谢

到现在，毕业设计已经接近尾声。通过此次毕业设计让我感受颇深，因为这次是对大学四年所学知识的总结和运用。整个项目的设计编写是自己独立完成，或多或少对自己的意义很深。使自己对技术的掌握更加深入，对未来的学习和工作有着深远的意义。通过项目的设计与完成，让我对完成一个项目从开始的需求分析和框架的搭建更加熟悉，对自己还未掌握和掌握不熟练的技术有了进一步的学习和理解。当然这些都归功于学校，学院老师和师哥师姐以及身边的同学的鼓励和帮助，我才可以通过不断地思考和学习去掌握理解更多的先进的技术来丰富自己，充实自己。为此，我特别感谢我的指导老师杨云老师对我的指导，同时感谢学姐秦一梅对我的帮助，让我在诸多正能量的基石上迈的更远。

最后，我要感谢陕西科技大学对我四年的培养，让我学到了不管在生活中，学习上，遇到问题要积极的面对问题，从问题中找解决它的办法。同时感谢大学中所有传道授业于我的每一位老师，也感谢身边一直带给自己正能量的所有同学。

# 参 考 文 献

[1] 张喆.高校就业信息管理系统的设计与实现[D].厦门:厦门大学,2016.

[2] 王文明.基于JavaEE架构的SSH框架[J].电子技术与软件工程,2017,(05):69.

[3] 胡杰,周鹏飞,郭乔进.基于MVC设计模式的SSH框架的研究[J].信息化研究,2016,(01):17-22.

[4] 何雪锋,谢宇. 深入浅出SSH框架[J]. 电脑知识与技术,2016,(14):206-207.

[5] 郭静.基于JSP的动态网页开发技术[J]. 科技展望,2016,(03):10-11.

[6] 邢效崔.基于J2EE的毕业生就业管理系统设计实现[D].天津:天津大学,2015.

[7] Hong Zhang,Hua Bing Li,Huan Jie Liu. Design and Implementation of Management Information System Based on SSH Architecture for Departments of Colleges and Universities[J]. Advanced Materials Research,2013,2534(756):.

[8] 邢效崔.基于J2EE的毕业生就业管理系统的设计与设计[D].天津:天津大学,2015.

[9] Tan Qingyan,Zhang Wei. Java open Source Project-Detailed Spring+Hibernate+Struts Project Development. Tsinghua University press,2008:9-10.

[10] 黄薇薇.某高校毕业生就业管理系统设计与实现[D].厦门:厦门大学,2015.

[11] Yue Hua Ding,Chang Hua Liu,Yu Xin Tang. MVC Pattern Based on JAVA[J]. Applied Mechanics and Materials,2012,1945(198):.

[12] 曹萍.数据库理论实践[J].电脑编程技巧与维护,2016,(13):5-7.

[13] 张晓鹏.基于B/S模式下的高职毕业生就业管理系统的设计与实现[J].化工管理,2016,(17):57.

[14] Chang Jiang Li,Xiao Hui Li,Yu Fen Wang. The Research and Application of Struts Framework Based on MVC Pattern[J]. Applied Mechanics and Materials,2011,1280(63):.

[15] 王鹏邦.基于Web的就业管理系统的设计与实现[D].吉林:吉林大学,2015.

[16] 张海藩.软件工程导论[M].北京:清华大学出版社,2008.