目 录

1研究背景及可行性分析 ………………………………………………… 2

2 需求分析…………………………………………………………………… 2

2.1 用户需求的描述…………………………………………………… 2

　　2.2 项目的核心功能分析……………………………………………… 2

3 系统设计……………………………………………………………………3

3.1 系统总体设计………………………………………………………3

　 3.2 项目命名规范……………………………………………………3

3.3 数据库设计………………………………………………………4

3.4 系统核心功能设计………………………………………………5

4 系统实现…………………………………………………………………6

5 系统测试…………………………………………………………………15

6 结 语……………………………………………………………………22

1.研究背景及可行性分析

随着Internet技术的发展，人们的日常生活已经离不开网络。未来社会人们的生活和工作将越来越依赖于数字技术的发展，越来越数字化、网络化、电子化、虚拟化。网上购物也成为了电子商务的一个时尚，而网上书店是最普遍，也最流行的一个动态的电子商务网站，它使每- -位顾客不用出门在家里就能够通过上网来轻松购买自己喜爱的书籍。为此创建该网上书店系统是为了能够扩展书店的客户群、科学管理、提高效益。该书店以经营英语、计算机书籍为主。在网上书店建立初期，要求在确保基本功能正常的情况下，尽量简化。在技术方面SSH框架技术与JSP等技术的结合，能够使得用户体验最大化。在项目组人员方面，技术相差不大，外部有优秀工程师指导。在风险方面，政府有资金方面的支持。与市面上的同类产品相比，我们的产品界面友好,符合大多数用户使用习惯，会让用户满意度更高。

1. 需求分析

2.1用户需求的描述

本系统的网上书店分为前后台两个管理系统:用户管理系统是在Internet上接受来自全国各地的客户访问的网上书店前台销售管理系统(网站):管理员管理系统是在Internet内部用于书店的管理员管理的网上书店后台管理系统。本系统的客户管理系统实现的功能类似于现实生活中的商店销售，顾客可以浏览商品、选择商品、提交信息完成购物等。本系统设计的网上购物的过程为，当客户通过Internet访问书店的网站时，如果他要购买商品，那么必须先注册，然后再进行登录(确认身份)，然后才可以把购买的图书放入“购物车”，通过购物车完成购物后，客户进行提交信息，生成网上订单。

本系统的后台管理系统触入了企业传统的进销存概念，对图书进行分类管理，提供图书信息的添加，修改和删除等功能，支持图书的封面图片上传功能, 从而增强了图书商品的宣传效果。

2.2项目的核心功能分析

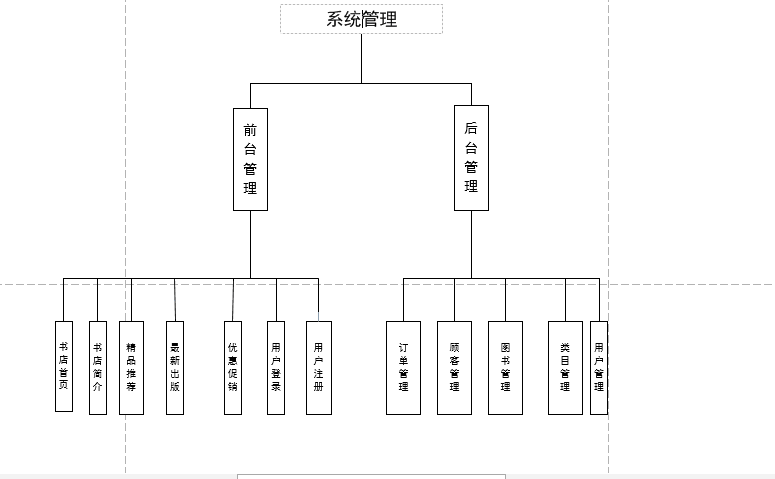
我们的这个项目主要分为前台管理和后台管理两个部分，前台管理中的主要功能是书店的简介，特卖图书以及精品推荐图书以及优惠促销，图书可以根据不同的类别进行信息查看，用户可以在书店把自己喜欢的图书添加购物车，填写相关信息进行购买。

项目的后台主要是对订单进行管理，同时还有顾客管理，可以添加顾客，查看顾客列表；后台中的图书管理功能也尤为重要，可以对图书进行增删改查；我们的后台功能非常全面，还有对图书类别的添加，删除以及修改的功能，同时也有对管理员的管理功能。

3 系统设计

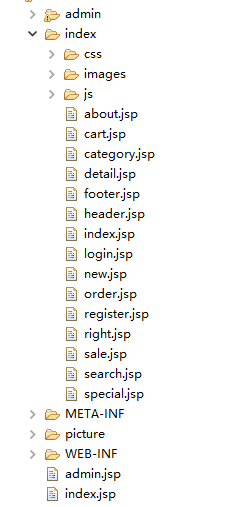
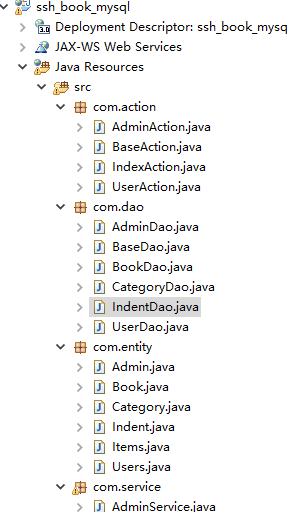
3.1系统总体设计

我们这个项目的系统设计图如下:



3.2项目命名规范

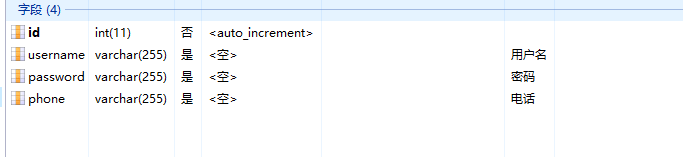
我们的项目命名规范，主要都是由相应的英文简写而来，而且层次分明，不同类型的放在不同的包下，很整齐。截图如下:



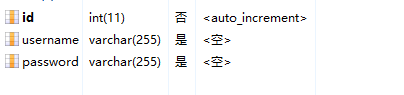
3.3数据库设计

数据库主要有六张表,分别是users,admin,book,indent,items,category

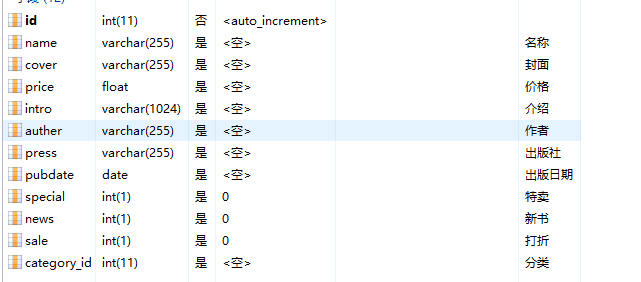
每张表都有其相应的属性，截图如下:



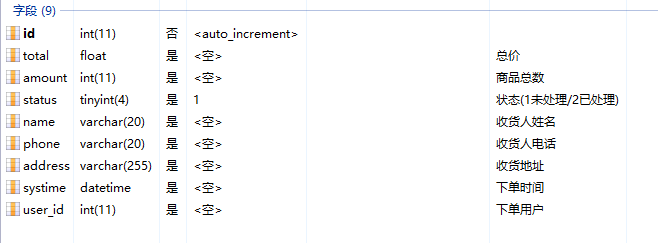
uses表



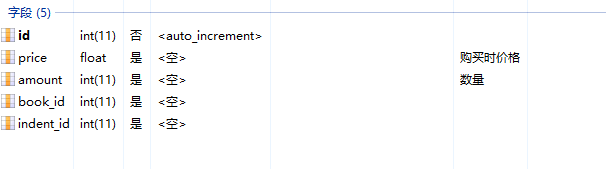
admin表



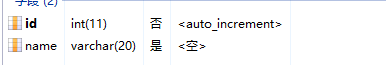
book表



indent表



items表



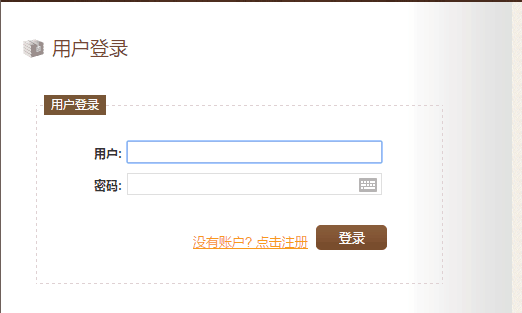
category表

3.4系统核心功能设计

我们的这个网上书店主要分为前台管理和后台管理两个部分，前台管理主要是针对普通用户而言，后台管理是针对管理员而言。前台管理中的主要功能是书店的简介，用户可以根据自己喜爱的图书类型进行分类查找，登录后可以加入购物车进行购买。项目的后台主要是对订单进行管理，同时还有顾客管理，对管理员的管理，对图书的管理，可以对图书进行增加，删除，查找，图书数据信息修改等操作，管理员也可以添加，删除，修改。

4.系统实现

前台的登录，注册功能的截图如下:



关键代码如下:

@Action("login")

public String login() {

flag = 6;

if(userService.checkUser(user.getUsername(), user.getPassword())){

getSession().put("username", user.getUsername());

return "index";

} else {

addActionMessage("用户名或密码错误!");

return "login";

}

}

@Action("register")

public String register(){

flag = 7;

if (user.getUsername().isEmpty()) {

addActionMessage("用户名不能为空!");

return "register";

}else if (userService.isExist(user.getUsername())) {

addActionMessage("用户名已存在!");

return "register";

}else {

userService.add(user);

addActionMessage("注册成功, 请登录!");

return "login";

}

}

@Action("logout")

public String logout() {

getSession().remove("username");

getSession().remove("indent");

return "login";

}

前台首页的截图:



主要代码如下:

<li <s:if test="flag==1">class="selected"</s:if>><a href="index.action">书店首页</a></li>

<li <s:if test="flag==2">class="selected"</s:if>><a href="about.action">书店简介</a></li>

<li <s:if test="flag==3">class="selected"</s:if>><a href="special.action">精品推荐</a></li>

<li <s:if test="flag==4">class="selected"</s:if>><a href="new.action">最新出版</a></li>

<li <s:if test="flag==5">class="selected"</s:if>><a href="sale.action">优惠促销</a></li>

<s:if test="#session.username==null">

<li <s:if test="flag==6">class="selected"</s:if>><a href="login.jsp">用户登陆</a></li>

<li <s:if test="flag==7">class="selected"</s:if>><a href="register.jsp">用户注册</a></li>

<li><a href="../admin/login.jsp" target="\_blank">后台管理</a></li>

</s:if>

<s:else>

<li><a href="order.action">我的订单</a></li>

<li><a href="logout.action">注销登录</a></li>

</s:else>

<li style="float: right; margin-right: 10px;">

<form action="search.action" method="post" id="form\_search">

<a href="javascript:void(0);" onclick="$('#form\_search').submit()" style="float: right;margin-left:0">搜索</a>

<input type="text" name="searchName" value="${searchName}" placeholder="输入图书名称" />

</form>

</li>

</ul>

</div>

</div>

前台图书分类的截图如下:



代码如下:

<title>图书分类</title>

</head>

<body>

<div id="wrap">

<jsp:include page="header.jsp"/>

<div class="center\_content">

<div class="left\_content">

<div class="title"><span class="title\_icon"><img src="images/bullet1.gif"/></span>图书分类</div>

<div class="new\_products">

<s:iterator var="book" value="bookList">

<div class="new\_prod\_box"> <a href="details.htm">${name}</a>

<div class="new\_prod\_bg">

<s:if test="special==true"><span class="new\_icon"><img src="images/special\_icon.gif"/></span></s:if>

<s:if test="news==true"><span class="new\_icon"><img src="images/new\_icon.gif"/></span></s:if>

<s:if test="sale==true"><span class="new\_icon"><img src="images/promo\_icon.gif"/></span></s:if>

<a href="detail.action?bookid=${id}"><img src="../${cover}" class="thumb" border="0" /></a>

</div>

</div>

</s:iterator>

前台图书搜索的截图如下:



关键代码如下:

@Action("search")

public String search(){

if (searchName!=null && !searchName.trim().isEmpty()) {

bookList = bookService.getList(searchName, page, 12);

pageTool = PageUtil.getPageTool(servletRequest, bookService.getTotal(searchName), page, 12);

}

return "search";

}

前台添加购物车和查看购物车截图如下:



关键代码如下:

@Action("cart")

public String cart() {

Object username = getSession().get("username");

if (username!=null && !username.toString().isEmpty()) {

List<Indent> indentList = indentService.getListByUserid(userService.get(username.toString()).getId());

if (indentList!=null && !indentList.isEmpty()) {

indent = indentList.get(0); // 最后一次订单信息

}

}

return "cart";

}

@Action("buy")

public void buy(){

Indent indent = (Indent) getSession().get(indentKey);

if (indent==null) {

getSession().put(indentKey, indentService.createIndent(bookService.get(bookid)));

}else {

getSession().put(indentKey, indentService.addIndentItem(indent, bookService.get(bookid)));

}

sendResponseMsg("ok");

}

后台管理员的登录截图如下:



关键代码如下:

@Action("login")

public String login() {

if (adminService.checkUser(admin.getUsername(), admin.getPassword())) {

getSession().put("admin", admin.getUsername());

return "main";

}

addActionError("用户名或密码错误!");

return "login";

}

后台订单列表截图如下:



关键代码如下:

@Action("indentList")

public String indentList(){

indentList = indentService.getList(status, page, rows);

pageTool = PageUtil.getPageToolAdmin(servletRequest, indentService.getTotal(status), page, rows);

return "indent";

}

@Action("indentDispose")

public String indentDispose(){

indentService.dispose(id);

return "reindent";

}

@Action("indentDelete")

public String indentDelete(){

indentService.delete(id);

return "reindent";

}

后台图书列表截图如下:



关键代码如下:

@Action("bookList")

public String bookList(){

bookList = bookService.getList(status, page, rows);

pageTool = PageUtil.getPageToolAdmin(servletRequest, bookService.getTotal(status), page, rows);

return "book";

}

后台类目列表截图如下:



关键代码如下:

@Action("categoryList")

public String categoryList(){

categoryList = categoryService.getList(page, rows);

pageTool = PageUtil.getPageToolAdmin(servletRequest, categoryService.getTotal(), page, rows);

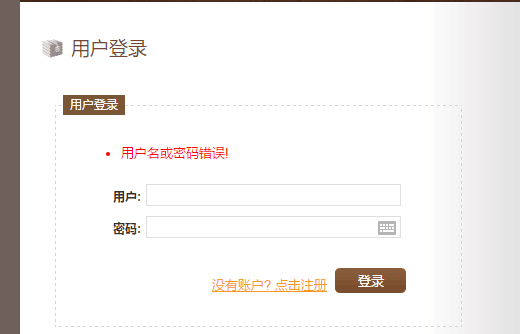
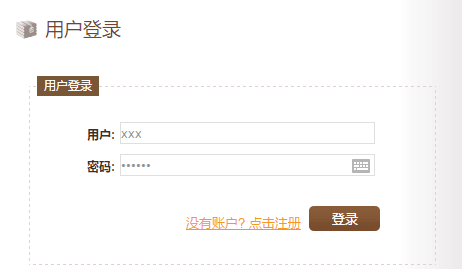
return "category";

}

5.系统测试

我们的系统分为前台和后台两部分，我们分别针对一些主要功能进行了测试。

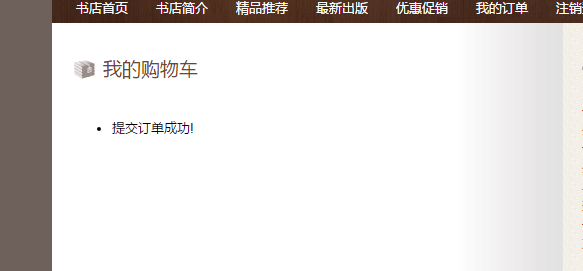
前台的登录功能:输入正确的用户名和密码，能够进入书店首页，否则提示用户名或密码不正确。截图如下:



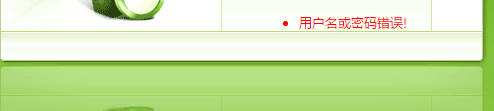
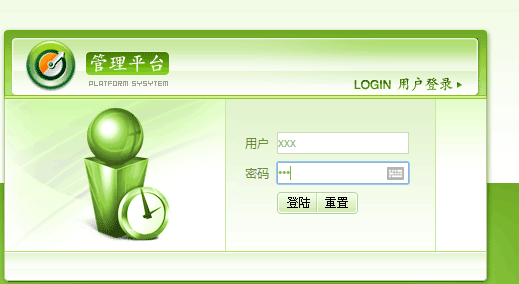
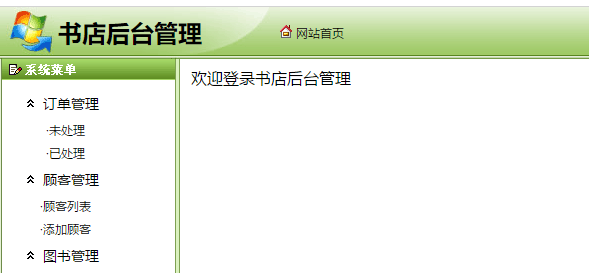
添加购物车,必须登录后提交订单，退出系统之后，加入购物车提交订单会显示让重新登录 截图如下:



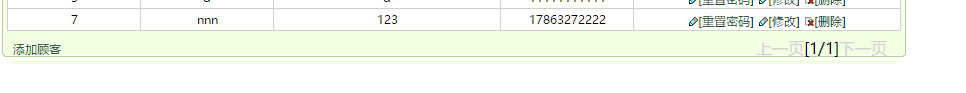
登录后再添加购物车，提交订单，会显示订单提交成功。截图如下:



后台的登录也是如此，输入正确的用户名和密码，就可以登录进去，否则显示用户名或密码不正确，截图如下:



对添加用户的功能进行测试，输入相应的信息，进行添加，截图如下:



对添加图书进行测试，添加成功后截图如下所示:

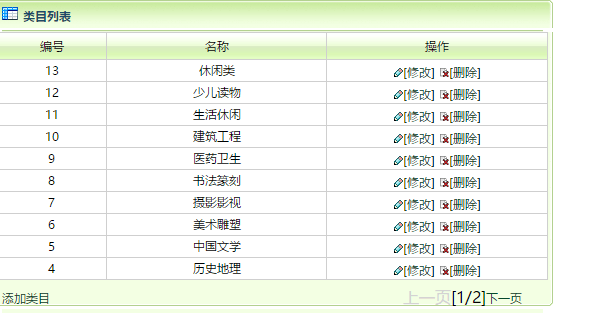
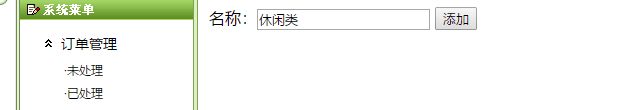




对后台把图书设置成为最新进行测试，截图如下:



对图书类别进行添加，填写信息后添加，如下图所示：



1. 结语

在设计这个系统中，不断地遇到不少问题，主要有下：（1）不是特别熟悉流程，不知如何下手。 通过参考一些优秀的课程设计范本，仔细观察总结，得出业务流程图实际是要表示此数据库是怎样帮助使用者进行数据管理和查询的。（2）在调试程序时，经常会出现一些错误，一些异常，过程并不是一帆风顺的，不过，在我们大家的共同努力下，一步一步，坚持不懈，终于将程序调试通过。

通过这次网上书店管理系统的课程设计，可以对数据库设计的整个过程有进一步了解，对SQL的运用也熟悉很多。但我们知道这还不是完整的一个管理系统，要让它能够得到实际运用还需要改进很多。

在设计过程中，我们遇到了很多问题，大多数都是大家互相商量解决的。很多时候，同学间的互助是最直接的最快捷的。虽然，每个人并不是都很精通这门课，但我们可以互相讨论，研究。

其实，认真理解课本中的内容也是很重要的，多看看课本中的例子，也使我们受益颇多，打好基础很重要。

这是我们对这次课程设计的一些体会，我们体会到我们的能力还是不够，还是需要改进，但是我们认识到坚持的重要性以及团队合作的重要性，我们以后会继续认真努力学习的，继续一起进步。