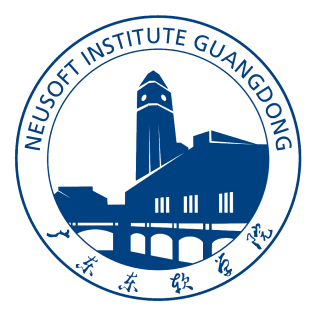
**

**Practice Report**

|  |  |
| --- | --- |
| **Course Name** | java |
| **Project Name** | Online bookstore |
| **Class** | 17 Software engineering |
| **Major** | Software engineering |
| **Lecturer** | Harry |
| **Student Number** | **17272020105**  **17272020106** |
| **Student Name** | **张鸿鑫（Pthahnil）**  **李健豪（Howard）** |

|  |  |
| --- | --- |
| Marked by: | Mark: |
| Marker’s comments: | |

**index**

[1. Introduction 5](#_Toc28001294)

[2. Practice Content 6](#_Toc28001295)

[About us.JSP 6](#_Toc28001296)

[Cart.jsp 11](#_Toc28001297)

[Error.jsp 14](#_Toc28001298)

[Footer.jsp 16](#_Toc28001299)

[Head.jsp 18](#_Toc28001300)

[Login.jsp 20](#_Toc28001301)

[Register.jsp 24](#_Toc28001302)

[Main.jsp 28](#_Toc28001303)

[message\_en\_US.properties 32](#_Toc28001304)

[message\_zh\_CN.properties 35](#_Toc28001305)

[hibernate.cfg.xml 38](#_Toc28001306)

[BookAction.java 41](#_Toc28001307)

[carAction.java 43](#_Toc28001308)

[LoginLocale.java 46](#_Toc28001309)

[registerAction.java 47](#_Toc28001310)

[registerAction-login-validation.xml 49](#_Toc28001311)

[registerAction-register-validation.xml 50](#_Toc28001312)

[BookDao.java 51](#_Toc28001313)

[CarDao.java 52](#_Toc28001314)

[BookDaoImpl.java 54](#_Toc28001315)

[CarDaoImpl.java 56](#_Toc28001316)

[UserDaoImpl.java 59](#_Toc28001317)

[Book.java 61](#_Toc28001318)

[Car.java 64](#_Toc28001319)

[User.java 68](#_Toc28001320)

[Book.hbm.xml 70](#_Toc28001321)

[Car.hbm.xml 71](#_Toc28001322)

[User.hbm.xml 72](#_Toc28001323)

[HibernateUtils.java 73](#_Toc28001324)

[struts.xml 75](#_Toc28001325)

[3. Diagram 78](#_Toc28001326)

[4. Screenshots 79](#_Toc28001327)

[5. Conclusion 84](#_Toc28001328)

1. Introduction

As we know, a web-style bookstore. Is a high quality, faster, more convenient way to buy books. Online bookstores can not only be used for online sales of books, but also for online sales of audio and video discs.

And the website type bookstore to the book management more reasonable, information.

While selling books, it also has the functions of book commodity management, shopping cart, order management, membership management, and so on. It has very flexible website content and article

management functions.

Buy a book in the online bookstore, you can check to buy more information, because online bookstore is a website, it has its unique way of selling books and functions.

Advantages of online bookstore: it involves a wide range of varieties, customers throughout the network coverage area, low operating costs, low discount prices, information dissemination speed.

1. Practice Content

About us.JSP

This page is mainly about our project name, team members,

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"q"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE html >

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

<div class=*"container"*>

<div class=*"breadcrumb-all-page"*>

<div class=*"bn-top"*>

<div class=*"container\_img\_bread no-img-cate img-breadcrumb-on-fieldthemecustomizer"*>

<img class=*"img-responsive back\_ct\_img"* src=*"*<%=basePath %>*/img/20190905113142.jpg"* style="width:*100%*;hight:*auto*;">

<div class=*"breadcrumb-abs"*>

</div>

</div>

</div>

</div>

<div id=*"content-wrapper"*>

<section id=*"main"*>

<h2 style="display:*none*"><font style="vertical-align: *inherit*;"><font style="vertical-align: *inherit*;">。</font></font></h2>

<header class=*"page-header"*>

<h1><font style="vertical-align: *inherit*;"><font style="vertical-align: *inherit*;">

关于我们

</font></font></h1>

</header>

<section id=*"content"* class=*"page-content page-cms page-cms-4"*>

<div id=*"cms-about-us"* class=*"block"* style="text-align:*left*;">

<div class=*"cms-row block block\_top clearfix"*>

<div class=*"row"*>

<div class=*"col-xs-12 col-md-12 col-lg-6 part\_2"*>

<div class=*"cms-box"*>

<p><font style="vertical-align: *inherit*;"><font style="vertical-align: *inherit*;">Focus on Internet Development</font></font></p>

<p><font style="vertical-align: *inherit*;"><font style="vertical-align: *inherit*;">A nice web page.</font></font></p>

</div>

</div>

</div>

</div>

<div class=*"cms-row block\_center block clearfix"*>

<h3 class=*"page-subheading"*><font style="vertical-align: *inherit*;"><font style="vertical-align: *inherit*;">About our team</font></font></h3>

<p class=*"des\_title"*><font style="vertical-align: *inherit*;"><font style="vertical-align: *inherit*;"></font><font style="vertical-align: *inherit*;">online bookshop</font></font></p>

<style type=*"text/css"*>

**body** {

background-color: *#E8E8E8*;

}

*.img-circle* {

z-index: *99*;

-moz-border-radius: *50%*;

-webkit-border-radius: *50%*;

border-radius: *50%*;

border: *8px solid*;

border-color: *rgba(255, 255, 255, 0.2)*;

}

*.img-circle:hover* {

-moz-box-shadow: *0px 0px 25px rgba(0, 255, 178, 1)*;

-webkit-box-shadow: *0px 0px 25px rgba(0, 255, 178, 1)*;

box-shadow: *0px 0px 25px rgba(0, 255, 178, 1)*;

cursor: *pointer*;

}

</style>

<img class=*"img-circle"* alt=*"我是头像"* src=*"*<%=basePath %>*/img/erzhi.png"* style="width: *100px*; height: *100px*;">

<img class=*"img-circle"* alt=*"我是头像"* src= *"*<%=basePath %>*/img/baba.jpg"*style="width: *100px*; height: *100px*;">

<p><font style="vertical-align: *inherit*;"><font style="vertical-align: *inherit*;">Howard 17272020106</font></font></p>

<p><font style="vertical-align: *inherit*;"><font style="vertical-align: *inherit*;">pthahnil 17272020105</font></font></p>

</div>

</div>

</div>

</div>

</div>

</div>

</head>

</html>

Cart.jsp

This page is the shopping cart page, click buy books, books will appear in the shopping cart

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"* import=*"java.util.\*"*%>

<%@ taglib prefix=*"q"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<%@ taglib prefix=*"s"* uri=*"/struts-tags"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML>

<html>

<head>

<title>Insert title here</title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

</head>

<body>

<q:import url=*"head.jsp"*/>

<div id=*"right"* align=*"left"*>

<div style="padding-right: *3%*; padding-left: *5%*; width: *90%*; height: *100%*;">

<div align=*"center"*>

<div id=*"title"* align=*"left"*>

<s:if test=*"carbooks.size()>0"*>

<table style="width:*100%*;border:*1*;">

<tr style="text-align: *center*">

<td width=*"100px"*><s:text name=*"picture\_book"*/></td>

<td width=*"180px"*><s:text name=*"name\_book"*/></td>

<td width=*"180px"*><s:text name=*"author\_book"*/></td>

<td width=*"90px"*><s:text name=*"price\_book"*/></td>

<td colspan=*"2"* width=*"150px"*><s:text name=*"operation"*/></td>

</tr>

<s:iterator value=*"carbooks"* status=*"carbook"*>

<s:if test=*"#userStatus.odd == true "*>odd</s:if>

<tr style="text-align: *center*">

<td width=*"100px"* ><img src=*"*<s:property value=*"picture"*/>*"* alt=*"book"* /></td>

<td width=*"180px"*><s:property value=*"name\_ch"* /></td>

<td width=*"180px"*><s:property value=*"author\_ch"* /></td>

<td width=*"90px"*><span>＄</span><s:property value=*"price"* /></td>

<td colspan=*"2"* width=*"150px"*><a href=*"cardel.action?username=*<s:property value=*"username"*/>*&bookid=*<s:property value=*"bookid"* />*"*>删除</a></td>

</tr>

</s:iterator>

</table>

</s:if><s:else><div><s:text name=*"not\_cart"*/></div></s:else>

</div>

</div>

</div>

</div>

<q:import url=*"footer.jsp"*/>

</body>

</html>

Error.jsp

An error will occur if the account password is entered incorrectly

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"s"* uri=*"/struts-tags"*%>

<%@ taglib prefix=*"q"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<base href=*"*<%=basePath%>*"*>

<title>error</title>

<meta http-equiv=*"pragma"* content=*"no-cache"*>

<meta http-equiv=*"cache-control"* content=*"no-cache"*>

<meta http-equiv=*"expires"* content=*"0"*>

<meta http-equiv=*"keywords"* content=*"keyword1,keyword2,keyword3"*>

<meta http-equiv=*"description"* content=*"This is my page"*>

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<q:import url=*"head.jsp"*/>

<body>

<s:text name=*"error.text"*></s:text>

<br>

<s:actionerror/>

</body>

<q:import url=*"footer.jsp"*/>

</html>

Footer.jsp

Each JSP page has a footer

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"* import=*"java.util.\*"*%>

<%@ taglib prefix=*"s"* uri=*"/struts-tags"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML>

<html><head>

<title></title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

</head>

<body>

<div class=*"footer"*>

<span class=*"span\_1"*>Copyright &copy; onlie bookshop book

<a href=*"*<%=basePath %>*/jsp/about us.jsp"*><s:text name=*"aboutus"*></s:text></a>

</span>

<a href=*"locale?requset\_locale=zh\_CN"*>简体中文</a>

<a href=*"locale?requset\_locale=en\_US"*>English</a>

</div>

</body>

</html>

Head.jsp

Each JSP page has a head

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"* import=*"java.util.\*"*%>

<%@ taglib prefix=*"s"* uri=*"/struts-tags"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML>

<html>

<head>

<base href=*"*<%=basePath%>*"*>

<title></title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<link type=*"text/css"* rel=*"stylesheet"* href=*"css/main.css"* />

</head>

<body>

<div class=*"top\_title"*>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<a href=*"jsp/login.jsp"*><s:text name=*"p\_login"*/></a>

<a href=*"jsp/register.jsp"*><s:text name=*"p\_registr"*/></a>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<s:if test=*"username!=null"*>

<a href=*"main?username=*<s:property value=*"username"*/>*"*><s:text name=*"buy"*/></a>

</s:if>

</div>

</body>

</html>

Login.jsp

The login page

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"* import=*"java.util.\*"*%>

<%@ taglib prefix=*"s"* uri=*"/struts-tags"*%>

<%@ taglib prefix=*"q"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML>

<html>

<head>

<base href=*"*<%=basePath%>*"*>

<title><s:text name=*"login.title"*></s:text></title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<script type=*"application/x-javascript"*> addEventListener("load", **function**() { setTimeout(hideURLbar, 0); }, **false**);

**function** hideURLbar(){ window.scrollTo(0,1); } </script>

<link type=*"text/css"* rel=*"stylesheet"* href=*"css/main.css"* />

<link type=*"text/css"* rel=*"stylesheet"* href=*"css/flexslider.css"* />

</head>

<body>

<q:import url=*"head.jsp"*/>

<div class=*"headline"*>

<h1><s:text name=*"onliebookshop"*/></h1>

</div>

<div class=*"login-main"*>

<div class=*"login-left-side"*>

<div class=*"flexslider"*>

<ul class=*"slides"*>

<li>

<img src=*"img/g5.jpg"* alt=*"image"*/>

</li>

<li>

<img src=*"img/g2.jpg"* alt=*"image"*/>

</li>

<li>

<img src=*"img/g3.jpg"* alt=*"image"*/>

</li>

<li>

<img src=*"img/g4.jpg"* alt=*"image"*/>

</li>

<li>

<img src=*"img/g1.jpg"* alt=*"image"*/>

</li>

</ul>

</div>

</div>

<div class=*"login-rigt-side"*>

<tr><td colspan=*"2"*><FONT color=*"red"*><s:actionerror/></FONT></td><!--显示错误信息 --></tr>

<s:head/>

<s:form action=*"login"* method=*"post"*>

<s:textfield name=*"username"* key=*"username"* />

<s:password name=*"password"* key=*"password"* />

<s:submit key=*"login.submit"*/>

</s:form>

</div>

</div>

<q:import url=*"footer.jsp"*/>

<script type=*"text/javascript"* src=*"js/jquery-2.1.4.min.js"*></script>

<script defer src=*"js/jquery.flexslider.js"*></script>

<script type=*"text/javascript"*>

$(window).load(**function**(){

$('.flexslider').flexslider({

animation: "slide",

start: **function**(slider){

$('body').removeClass('loading');

}

});

});

</script>

</body>

</html>

Register.jsp

Register page

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"* import=*"java.util.\*"*%>

<%@ taglib prefix=*"s"* uri=*"/struts-tags"*%>

<%@ taglib prefix=*"q"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML>

<html>

<head>

<base href=*"*<%=basePath%>*"*>

<title><s:text name=*"register.title"*></s:text></title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<script type=*"application/x-javascript"*> addEventListener("load", **function**() { setTimeout(hideURLbar, 0); }, **false**);

**function** hideURLbar(){ window.scrollTo(0,1); } </script>

<link type=*"text/css"* rel=*"stylesheet"* href=*"css/main.css"* />

<script type=*"text/javascript"*src=*"js/check.js"*></script>

<link type=*"text/css"* rel=*"stylesheet"* href=*"css/flexslider.css"* />

</head>

<body id=*"reg"* onload="$('uname').focus();">

<q:import url=*"head.jsp"*/>

<div class=*"headline"*>

<h1><s:text name=*"onliebookshop"*></s:text></h1>

</div>

<div class=*"login-main"*>

<div class=*"login-left-side"*>

<div class=*"flexslider"*>

<ul class=*"slides"*>

<li>

<img src=*"img/g5.jpg"* alt=*"image"*/>

</li>

<li>

<img src=*"img/g2.jpg"* alt=*"image"*/>

</li>

<li>

<img src=*"img/g3.jpg"* alt=*"image"*/>

</li>

<li>

<img src=*"img/g4.jpg"* alt=*"image"*/>

</li>

<li>

<img src=*"img/g1.jpg"* alt=*"image"*/>

</li>

</ul>

</div>

</div>

<div class=*"login-rigt-side"*>

<tr><td colspan=*"2"*><FONT color=*"red"*><s:actionerror/></FONT></td><!--显示错误信息 --></tr>

<s:head/>

<s:form action=*"register"* method=*"post"*>

<s:textfield name=*"username"* key=*"username"* />

<s:password name=*"password"* key=*"password"* />

<s:radio list=*"*#{'1':getText('man'),'0':getText('women')}*"*

name=*"gender"* key=*"gender"* />

<s:textfield name=*"address"* key=*"address"* />

<s:submit key=*"register.submit"* />

</s:form>

</div>

</div>

<q:import url=*"footer.jsp"*/>

<script type=*"text/javascript"* src=*"js/jquery-2.1.4.min.js"*></script>

<script defer src=*"js/jquery.flexslider.js"*></script>

<script type=*"text/javascript"*>

$(window).load(**function**(){

$('.flexslider').flexslider({

animation: "slide",

start: **function**(slider){

$('body').removeClass('loading');

}

});

});

</script>

</body>

</html>

Main.jsp

The price of books by author, classification, etc

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"* import=*"java.util.\*"*%>

<%@ taglib prefix=*"s"* uri=*"/struts-tags"*%>

<%@ taglib prefix=*"q"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML>

<html>

<head>

<base href=*"*<%=basePath%>*"*>

<title>Insert title here</title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<link type=*"text/css"* rel=*"stylesheet"* href=*"css/main.css"* />

<script language =*javascript* >

**var** curIndex=0;

//时间间隔 单位毫秒

**var** timeInterval=5000;

**var** arr=**new** Array();

arr[0]="M1.jpg";

arr[1]="M2.jpg";

setInterval(changeImg,timeInterval);

**function** changeImg()

{

**var** obj=document.getElementById("showpic");

**if** (curIndex==arr.length-1)

{

curIndex=0;

}

**else**

{

curIndex+=1;

}

obj.src="img/"+arr[curIndex];

}

</script>

</head>

<body>

<q:import url=*"head.jsp"*/>

<div class=*"login-main"*>

<div class=*"main-mian"*>

<div class=*"top\_banner"*>

<div class=*"top\_logo"*><a href=*"/JSP/main.jsp"*><img src=*"img/top\_logo.jpg"* alt=*"A BOOK APART LOGO"* /></a></div>

<div class=*"top\_shop\_cur"*>

<a href=*"carshow?username=*<s:property value=*"username"*/>*"*><img src=*"img/top\_shop\_cur.jpg"* alt=*"shopping car"* /></a>

</div>

</div>

</div>

<span class=*"index\_img"*>

<img src=*"img/M1.jpg"* id=*"showpic"* />

</span>

<p class=*"index\_hr"*></p>

<s:if test=*"books.size()>0"*>

<div class=*"content"*>

<h1 class=*"h1\_book\_title"*>Also from A Book Apart--</h1>

<div class=*"content"*>

<ul>

<s:iterator value=*"books"* status=*"book"*>

<li class=*"*<s:if test=*"#userStatus.odd == true "*>*odd*</s:if><s:else>*even*</s:else>*"*>

<dl>

<dd style="text-align: *center*;">

<img src=*"*<s:property value=*"picture"*/>*"* alt=*"book"* />

<p class=*"book\_title"*><s:property value=*"name\_en"* /></p>

<p class=*"book\_title"*><s:property value=*"author\_en"* /></p>

<p class=*"book\_inline"*><span>＄</span><s:property value=*"price"* /></p>

<div style="width:*250px*;text-align: *center*;">

<a class=*"book\_buy"* href=*"addcar.action?username=*<s:property value=*"username"*/>*&bookid=*<s:property value=*"id"* />*"*target=*"\_blank"*>BUY</a>

</div>

</dd>

</dl>

</li>

</s:iterator>

</ul>

</div>

</div>

</s:if>

</div>

<q:import url=*"footer.jsp"*/>

</body>

</html>

message\_en\_US.properties

Application in I18n

# Information required to log in

username=username

password=password

repassword=repassword

gender=gender

address=address

man=man

women=women

#login.jsp

login.title=login bookshop

login.submit=login

#register.jsp

register.title=register bookshop

register.submit=register

#footer.jsp

aboutus=about us

#head.jsp

weclome\_1=Hello, welcome to bookshop!

weclome\_2=, welcome to bookshop!

weclome\_3=Hello

p\_login=[please login]

p\_registr=[Free registration]

writeoff=[write off]

buy=BUY

#main.jsp

#cart.jsp

picture\_book=picture

name\_book=book name

author\_book=author

price\_book=price

operation=operation

not\_cart=There are no items in your shopping cart, please return to add

#validate

validate\_name\_null=Username cannot be empty

validate\_pass\_null=Password cannot be empty

validate\_pass\_scope=Password must be between 6 and 12 digits and can only be letters or numbers

validate\_repass\_null=rePassword cannot be empty

validate\_repass\_scope=The two passwords must be the same

validate\_gender\_null=Gender cannot be empty

validate\_address\_null=Address cannot be empty

#error.jsp

error.title=error

error.text=You have entered the bug written by the editor. Please contact the administrator to solve the problem.

#Common

onliebookshop=onlie bookshop

#input

User\_full=User already exists

User\_empty=The user does not exist

Account\_error=Incorrect username/password

message\_zh\_CN.properties

Application in I18n

# Information required to log in

username=\u7528\u6237\u540D

password=\u5BC6\u7801

repassword=\u786E\u8BA4\u5BC6\u7801

gender=\u6027\u522B

address=\u5730\u5740

man=\u7537

women=\u5973

#login.jsp

login.title=\u767B\u5F55\u7F51\u4E0A\u4E66\u5E97

login.submit=\u767B\u5F55

#register.jsp

register.title=\u6CE8\u518C\u7F51\u4E0A\u4E66\u5E97

register.submit=\u6CE8\u518C

#footer.jsp

aboutus=\u5173\u4E8E\u6211\u4EEC

#head.jsp

weclome\_1=\u60A8\u597D\uFF0C\u6B22\u8FCE\u6765bookshop\uFF01

weclome\_2=\uFF0C\u6B22\u8FCE\u6765bookshop\uFF01

weclome\_3=\u60A8\u597D

p\_login=[\u8BF7\u767B\u5F55]

p\_registr=[\u514D\u8D39\u6CE8\u518C]

writeoff=[\u6CE8\u9500]

buy=\u6211\u8981\u4E70

#main.jsp

#cart.jsp

picture\_book=\u56FE\u7247

name\_book=\u56FE\u4E66\u540D

author\_book=\u4F5C\u8005

price\_book=\u4EF7\u94B1

operation=\u64CD\u4F5C

not\_cart=\u60A8\u8D2D\u7269\u8F66\u91CC\u8FD8\u6CA1\u6709\u4EFB\u4F55\u7269\u54C1\uFF0C\u8BF7\u8FD4\u56DE\u6DFB\u52A0

#validate

validate\_name\_null=\u7528\u6237\u540D\u4E0D\u80FD\u4E3A\u7A7A

validate\_pass\_null=\u5BC6\u7801\u4E0D\u80FD\u4E3A\u7A7A

validate\_pass\_scope=\u5BC6\u7801\u5FC5\u987B\u57286\u523012\u4F4D\u4E4B\u95F4\uFF0C\u4E14\u53EA\u80FD\u662F\u5B57\u6BCD\u6216\u6570\u5B57

validate\_repass\_null=\u786E\u8BA4\u5BC6\u7801\u4E0D\u80FD\u4E3A\u7A7A

validate\_repass\_scope=\u4E24\u6B21\u8F93\u5165\u7684\u5BC6\u7801\u5FC5\u987B\u4E00\u81F4

validate\_gender\_null=\u6027\u522B\u4E0D\u80FD\u4E3A\u7A7A

validate\_address\_null=\u5730\u5740\u4E0D\u80FD\u4E3A\u7A7A

#error.jsp

error.title=\u9519\u8BEF\u5566

error.text=\u4F60\u8FDB\u5165\u4E86\u5C0F\u7F16\u7F16\u5199\u597D\u7684bug\uFF0C\u8BF7\u8054\u7CFB\u7BA1\u7406\u5458\u89E3\u51B3\u95EE\u9898\u54E6

#Common

onliebookshop=\u7F51\u4E0A\u4E66\u5E97

#input

User\_full=\u7528\u6237\u5DF2\u5B58\u5728

User\_empty=\u7528\u6237\u4E0D\u5B58\u5728

Account\_error=\u7528\u6237\u540D/\u5BC6\u7801\u9519\u8BEF

hibernate.cfg.xml

Hibernate new framework for connecting to databases

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- 1、配置数据库连接的4个参数 -->

<property name=*"hibernate.connection.driver\_class"*>com.mysql.cj.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql:///bookstore?useUnicode=true&amp;characterEncoding=utf8&amp;serverTimezone=GMT%2B8</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*>a451697920</property>

<!-- javax.persistence.validation.mode默认情况下是auto的

就是说如果不设置的话它是会自动去你的classpath下面找一个bean-validation\*\*包

但是找不到，所以beanvalitionFactory错误。 -->

<property name=*"javax.persistence.validation.mode"*>none</property>

<!-- 是否显示sql语句 -->

<property name=*"show\_sql"*>false</property>

<!-- 是否格式化sql语句 -->

<property name=*"format\_sql"*>false</property>

<!-- 是否自动提交事务，针对insert有效，针对delete无效 -->

<property name=*"hibernate.connection.autocommit"*>true</property>

<!-- 开启与当前线程绑定session的功能

ThreadLocal<Connection>

ThreadLocal<ActionContext>

ThreadLocal<session>

-->

<property name=*"hibernate.current\_session\_context\_class"*>thread</property>

<!--hibernate.hbm2ddl.auto

配置映射文件与数据库表的关系

update:如果数据库没有表，自动帮你创建表【常用】

如果hbm与数据表不一样，会更新

create：每次启动hibernate都帮你创建表

create-drop，每次启动hibernate都帮你创建表，执行完后删除表

validate:检验hbm文件，如果与数据库的字段不一致，就抛出异常【过掉】

-->

<property name=*"hibernate.hbm2ddl.auto"*></property>

<!-- hibernate.doalect:数据库方言

mysql:分页limit

oracle:分页rownum

-->

<property name=*"hibernate.doalect"*>org.hibernate.dialect.MySQL5Dialect</property>

<!-- 2、配置JavaBean与表的映射文件 -->

<mapping resource=*"com/bookshop/entity/User.hbm.xml"*/>

<mapping resource=*"com/bookshop/entity/Book.hbm.xml"*/>

<mapping resource=*"com/bookshop/entity/Car.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

BookAction.java

The action mapping is the mapping of a request bookaction to an action class, and when a request matches the name of bookaction, the framework USES this mapping to determine how to handle the request.

package com.bookshop.action;

import java.util.List;

import com.bookshop.dao.BookDao;

import com.bookshop.dao.Impl.BookDaoImpl;

import com.bookshop.entity.Book;

import com.opensymphony.xwork2.ActionSupport;

public class BookAction extends ActionSupport{

private static final long serialVersionUID = 1L;

public List<Book> books;

BookDao bookdao=new BookDaoImpl();

public String showbook(){

books=bookdao.getBook();

return SUCCESS;

}

}

carAction.java

The action mapping is the mapping of a request cartaction to an action class, and when a request matches the name of the caartaction, the framework USES this mapping to determine how to handle the request.

package com.bookshop.action;

import java.util.List;

import com.bookshop.dao.BookDao;

import com.bookshop.dao.CarDao;

import com.bookshop.dao.Impl.BookDaoImpl;

import com.bookshop.dao.Impl.CarDaoImpl;

import com.bookshop.entity.Book;

import com.bookshop.entity.Car;

import com.bookshop.entity.User;

import com.opensymphony.xwork2.ActionSupport;

public class carAction extends ActionSupport{

private static final long serialVersionUID = 1L;

public List<Car> carbooks;

private String username;

CarDao cardao=new CarDaoImpl();

BookDao bookdao=new BookDaoImpl();

private String bookid;

public String addcar(){

Book book = bookdao.findBook(Integer.parseInt(bookid));

cardao.Addbook(new Car(username,book,1));

return SUCCESS;

}

public String deletecar(){

cardao.deletebook(username,Integer.parseInt(bookid));

return SUCCESS;

}

public String showcar(){

carbooks=cardao.getcar(username);

System.out.println("carbooks的大小为："+carbooks.size());

return SUCCESS;

}

public String getBookid() {

return bookid;

}

public void setBookid(String bookid) {

this.bookid = bookid;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

}

LoginLocale.java

Applied to i18n

package com.bookshop.action;

import java.util.Locale;

import java.util.ResourceBundle;

import com.opensymphony.xwork2.ActionSupport;

public class LoginLocale extends ActionSupport{

private static final long serialVersionUID = 1L;

public String execute() {

return SUCCESS;

}

}

registerAction.java

Action mapping is the mapping of a request registeraction to an action class, and when a request matches registeraction's name, the framework USES this mapping to determine how to handle the request.

**ackage** com.bookshop.action;

**import** com.bookshop.dao.UserDao;

**import** com.bookshop.dao.Impl.UserDaoImpl;

**import** com.bookshop.entity.User;

**import** com.opensymphony.xwork2.ActionSupport;

**import** com.opensymphony.xwork2.ModelDriven;

**public** **class** registerAction **extends** ActionSupport **implements** ModelDriven<User>{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** User user;

UserDao userDAO = **new** UserDaoImpl();

@Override

**public** User getModel() {

user=**new** User();

**return** user;

}

**public** String register() {

**if**(userDAO.get(user.getUsername())==**null**){

userDAO.AddUser(user);

**return** ***SUCCESS***;

}**else**{

addActionError(getText("User\_full"));

**return** ***INPUT***;

}

}

**public** String login(){

User t\_user=userDAO.get(user.getUsername());

**if**(t\_user!=**null**){

**if**(t\_user.getPassword().equals(user.getPassword())){

**return** ***SUCCESS***;}

**else**{

addActionError(getText("Account\_error"));

**return** ***INPUT***;}

}**else**{

addActionError(getText("User\_empty"));

**return** ***INPUT***;

}

}

}

registerAction-login-validation.xml

Login validation prevents null values

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE validators PUBLIC

"-//Apache Struts//XWork Validator 1.0.3//EN"

"http://struts.apache.org/dtds/xwork-validator-1.0.3.dtd">

<validators>

<validator type=*"requiredstring"*>

<param name=*"fieldName"*>username</param>

<message key=*"validate\_name\_null"*></message>

</validator>

<field name=*"password"*>

<field-validator type=*"requiredstring"*>

<param name=*"trim"*>true</param>

<message key=*"validate\_pass\_null"*></message>

</field-validator>

</field>

</validators>

registerAction-register-validation.xml

Registration verification prevents null values

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE validators PUBLIC

"-//Apache Struts//XWork Validator 1.0.3//EN"

"http://struts.apache.org/dtds/xwork-validator-1.0.3.dtd">

<validators>

<validator type=*"requiredstring"*>

<param name=*"fieldName"*>username</param>

<message key=*"validate\_name\_null"*></message>

</validator>

<field name=*"password"*>

<field-validator type=*"requiredstring"*>

<param name=*"trim"*>true</param>

<message key=*"validate\_pass\_null"*></message>

</field-validator>

</field>

</validators>

BookDao.java

package com.bookshop.dao;

import java.util.List;

import com.bookshop.entity.Book;

public interface BookDao {

public List<Book> getBook();

public Book findBook(int bookid);

}

CarDao.java

package com.bookshop.dao;

import java.io.Serializable;

import java.util.List;

import com.bookshop.entity.Car;

public interface CarDao {

public void Addbook(Car car);

public void deletebook(String username,int bookid);

public List<Car> getcar(String username);

public void changcar(Car car);

}

package com.bookshop.dao;

import java.io.Serializable;

import com.bookshop.entity.User;

public interface UserDao {

/\*\* 添加一条记录

\* @return

\* @throws Exception \*/

public void AddUser(User user);

/\*\* 根据主键找对象 \*/

public User get(String username);

}

BookDaoImpl.java

Create an instantiation class: bookdao is used to retrieve the contents of the bookinfo table

package com.bookshop.dao.Impl;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import com.bookshop.dao.BookDao;

import com.bookshop.entity.Book;

import com.bookshop.util.HibernateUtils;

public class BookDaoImpl implements BookDao {

@Override

public List<Book> getBook() {

try{

Session session=HibernateUtils.openSession();

Query query=session.createQuery("from Book");

List<Book> books = query.list();

session.close();

return books;

}catch(Exception e){

e.printStackTrace();

return null;

}

}

@Override

public Book findBook(int bookid) {

try{

Session session=HibernateUtils.openSession();

Book book=(Book)session.get(Book.class, bookid);

session.close();

return book;

}catch(Exception e){

e.printStackTrace();

return null; } } }

CarDaoImpl.java

Create the instantiated class cardao to retrieve the contents of the carinfo table

**package** com.bookshop.dao.Impl;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Hibernate;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** org.hibernate.criterion.Restrictions;

**import** com.bookshop.dao.CarDao;

**import** com.bookshop.entity.Book;

**import** com.bookshop.entity.Car;

**import** com.bookshop.entity.User;

**import** com.bookshop.util.HibernateUtils;

**public** **class** CarDaoImpl **implements** CarDao {

@Override

**public** **void** Addbook(Car car) {

**try**{

Session session=HibernateUtils.*openSession*();

Transaction trans=session.getTransaction();

trans.begin();

session.save(car);

trans.commit();

session.close();

}**catch**(Exception e){

e.printStackTrace();

}

}

@Override

**public** List<Car> getcar(String username) {

**try**{

Session session=HibernateUtils.*openSession*();

Query query=session.createQuery("from Car");

List<Car> cars=query.list();

session.close();

**return** cars;

}**catch**(Exception e){

e.printStackTrace();

**return** **null**;

}

}

@Override

**public** **void** changcar(Car car) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** deletebook(String username, **int** bookid) {

**try**{

Configuration cfg = **new** Configuration().configure();

**final** SessionFactory factory=cfg.buildSessionFactory();

Session session=factory.openSession();

session.getTransaction().begin();

Query query=session.createQuery("from Car where username=? and bookid=?");

query.setParameter(0, username);

query.setParameter(1, bookid);

Car car=(Car) query.uniqueResult();

session.delete(car);

session.getTransaction().commit();

session.clear();

factory.close();

}**catch**(Exception e){

e.printStackTrace();

}

}

}

UserDaoImpl.java

Create an instantiation class: UserDAO is used to retrieve the contents of the userinfo table in the struts database

UserDaoImpl.java

**package** com.bookshop.dao.Impl;

**import** java.io.Serializable;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** com.bookshop.dao.UserDao;

**import** com.bookshop.entity.User;

**import** com.bookshop.util.HibernateUtils;

**public** **class** UserDaoImpl **implements** UserDao {

@Override

**public** **void** AddUser(User user){

**try**{

Session session=HibernateUtils.*openSession*();

Transaction trans=session.getTransaction();

trans.begin();

session.save(user);

trans.commit();

session.close();

}**catch**(Exception e){

e.printStackTrace();

}

}

@Override

**public** User get(String username) {

**try**{

Session session=HibernateUtils.*openSession*();

Transaction trans=session.getTransaction();

trans.begin();

User user=(User)session.get(User.**class**, username);

System.***out***.println(user);

trans.commit();

session.close();

**return** user;

}**catch**(Exception e){

e.printStackTrace();

**return** **null**;

}

}

}

Book.java

The Java Bean class that creates the book contains the id, picture, price, name, author, and its get and set methods

**package** com.bookshop.entity;

**import** java.io.Serializable;

/\*\*书籍实体类\*/

**public** **class** Book **implements** Serializable {

**private** Integer id;

**private** String picture;

**private** Double price;

**private** String name\_ch;

**private** String name\_en;

**private** String author\_ch;

**private** String author\_en;

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getPicture() {

**return** picture;

}

**public** **void** setPicture(String picture) {

**this**.picture = picture;

}

**public** Double getPrice() {

**return** price;

}

**public** **void** setPrice(Double price) {

**this**.price = price;

}

**public** String getName\_ch() {

**return** name\_ch;

}

**public** **void** setName\_ch(String name\_ch) {

**this**.name\_ch = name\_ch;

}

**public** String getName\_en() {

**return** name\_en;

}

**public** **void** setName\_en(String name\_en) {

**this**.name\_en = name\_en;

}

**public** String getAuthor\_ch() {

**return** author\_ch;

}

**public** **void** setAuthor\_ch(String author\_ch) {

**this**.author\_ch = author\_ch;

}

**public** String getAuthor\_en() {

**return** author\_en;

}

**public** **void** setAuthor\_en(String author\_en) {

**this**.author\_en = author\_en;

}

@Override

**public** String toString() {

**return** "Book [id=" + id + ", picture=" + picture + ", price=" + price + ", name\_ch=" + name\_ch + ", name\_en="

+ name\_en + ", author\_ch=" + author\_ch + ", author\_en=" + author\_en + "]";

}

}

Car.java

Create a Java Bean class for car that contains id, picture, price, name, author, price, and its get and set methods

**package** com.bookshop.entity;

**import** java.io.Serializable;

/\*\*书籍实体类\*/

**public** **class** Car **implements** Serializable {

**private** String username;

**private** **int** bookid;

**private** **int** number;

**private** String picture;

**private** String name\_ch;

**private** String name\_en;

**private** String author\_ch;

**private** String author\_en;

**private** Double price;

**public** Car() {

}

**public** Car(String username, Book book, **int** number) {

**this**.username = username;

**this**.bookid = book.getId();

**this**.number = number;

**this**.picture = book.getPicture();

**this**.name\_ch = book.getName\_ch();

**this**.name\_en = book.getName\_en();

**this**.author\_ch = book.getAuthor\_ch();

**this**.author\_en = book.getAuthor\_en();

**this**.price = book.getPrice();

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** **int** getBookid() {

**return** bookid;

}

**public** **void** setBookid(**int** bookid) {

**this**.bookid = bookid;

}

**public** **int** getNumber() {

**return** number;

}

**public** **void** setNumber(**int** number) {

**this**.number = number;

}

**public** String getPicture() {

**return** picture;

}

**public** **void** setPicture(String picture) {

**this**.picture = picture;

}

**public** String getName\_ch() {

**return** name\_ch;

}

**public** **void** setName\_ch(String name\_ch) {

**this**.name\_ch = name\_ch;

}

**public** String getName\_en() {

**return** name\_en;

}

**public** **void** setName\_en(String name\_en) {

**this**.name\_en = name\_en;

}

**public** String getAuthor\_ch() {

**return** author\_ch;

}

**public** **void** setAuthor\_ch(String author\_ch) {

**this**.author\_ch = author\_ch;

}

**public** String getAuthor\_en() {

**return** author\_en;

}

**public** **void** setAuthor\_en(String author\_en) {

**this**.author\_en = author\_en;

}

**public** Double getPrice() {

**return** price;

}

**public** **void** setPrice(Double price) {

**this**.price = price;

}

@Override

**public** String toString() {

**return** "Car [username=" + username + ", bookid=" + bookid + ", number=" + number + ", picture=" + picture

+ ", name\_ch=" + name\_ch + ", name\_en=" + name\_en + ", author\_ch=" + author\_ch + ", author\_en="

+ author\_en + ", price=" + price + "]";

}

}

User.java

Create the Java Bean class for User, containing username and password, address. Gender, and its get and set methods

**package** com.bookshop.entity;

**import** java.io.Serializable;

**public** **class** User **implements** Serializable {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** String username;

**private** String password;

**private** String address;

**private** String gender;

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

@Override

**public** String toString() {

**return** "User username=" + username + ", password=" + password + ", address=" + address

+ ", gender=" + gender + "]";

}

}

Book.hbm.xml

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<!-- name:模型的全名称 -->

<class name=*"com.bookshop.entity.Book"* table=*"bookinfo"*>

<!-- name:模型属性名 -->

<id name=*"id"* column=*"id\_book"*>

<!-- generator:id的生成策略

native:如果是mysql数据库，id会自动增长 -->

<generator class=*"native"*></generator>

</id>

<!-- 如果模型的属性和数据库的列名，就不用写column -->

<property name=*"picture"* column=*"picture\_book"*></property>

<property name=*"name\_ch"* column=*"name\_ch\_book"*></property>

<property name=*"name\_en"* column=*"name\_en\_book"*></property>

<property name=*"author\_ch"* column=*"author\_ch\_book"*></property>

<property name=*"author\_en"* column=*"author\_en\_book"*></property>

<property name=*"price"* column=*"price\_book"*></property>

</class>

</hibernate-mapping>

Car.hbm.xml

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<class name=*"com.bookshop.entity.Car"* table=*"car"*>

<id name=*"bookid"* column=*"id\_book"*></id>

<property name=*"username"* column=*"name\_user"*></property>

<property name=*"number"* column=*"number"*></property>

<property name=*"picture"* column=*"picture\_book"*></property>

<property name=*"name\_ch"* column=*"name\_ch\_book"*></property>

<property name=*"name\_en"* column=*"name\_en\_book"*></property>

<property name=*"author\_ch"* column=*"author\_ch\_book"*></property>

<property name=*"author\_en"* column=*"author\_en\_book"*></property>

<property name=*"price"* column=*"price\_book"*></property></class>

</hibernate-mapping>

User.hbm.xml

!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<!-- name:模型的全名称 -->

<class name=*"com.bookshop.entity.User"* table=*"userinfo"*>

<id name=*"username"* column=*"name\_user"*></id>

<property name=*"password"* column=*"pwd\_user"*></property>

<property name=*"address"* column=*"address\_user"*></property>

<property name=*"gender"* column=*"gender\_user"*></property>

</class>

</hibernate-mapping>

HibernateUtils.java

package com.bookshop.util;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import org.hibernate.classic.Session;

public class HibernateUtils {

private static SessionFactory factory;

static{

//1.获取核心配置文件对象，默认是加载src的hibernate.cfg.xml文件

Configuration cfg = new Configuration().configure();

//2.创建会话工厂

factory=cfg.buildSessionFactory();

//监听程序关闭

Runtime.getRuntime().addShutdownHook(new Thread(){

public void run() {

factory.close();

};});

}

public static Session openSession(){

return factory.openSession();

}

public static Session getSession(){

return factory.getCurrentSession();

}

}

struts.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"

"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>

<constant name=*"struts.enable.DynamicMethodInvocation"*

value=*"false"* />

<!-- 2.配置开发模式 ：开发模式：改了配置文件不需要重启。输出更多的错误信息。开发阶段建议为true -->

<constant name=*"struts.devMode"* value=*"false"* />

<constant name=*"struts.custom.i18n.resources"* value=*"resources.message"* />

<constant name=*"struts.i18n.encoding"* value=*"UTF-8"* />

<package name=*"p1"* extends=*"struts-default"* namespace=*"/"*>

<!-- 注册 -->

<action name=*"register"* class=*"com.bookshop.action.registerAction"* method=*"register"*>

<result name=*"success"*>/jsp/login.jsp</result>

<result name=*"input"*>/jsp/register.jsp</result>

<result name=*"error"*>/jsp/error.jsp</result>

</action>

<!-- 登录 -->

<action name=*"login"* class=*"com.bookshop.action.registerAction"* method=*"login"*>

<!-- 转发到main页面 -->

<result name=*"success"* type=*"chain"*>

<param name=*"namespace"*>/</param>

<param name=*"actionName"*>main</param>

</result>

<result name=*"input"*>/jsp/login.jsp</result>

<result name=*"error"*>/jsp/error.jsp</result>

</action>

<!-- main页面 -->

<action name=*"main"* class=*"com.bookshop.action.BookAction"* method=*"showbook"*>

<result name=*"success"*>/jsp/main.jsp</result>

</action>

<!-- 加入购物车 -->

<action name=*"addcar"* class=*"com.bookshop.action.carAction"* method=*"addcar"*>

<result name=*"success"* type=*"chain"*>

<param name=*"namespace"*>/</param>

<param name=*"actionName"*>carshow</param>

</result>

<!-- <result name="success">/jsp/login.jsp</result>-->

</action>

<!-- 删除购物车 -->

<action name=*"cardel"* class=*"com.bookshop.action.carAction"* method=*"deletecar"*>

<result name=*"success"* type=*"chain"*>

<param name=*"namespace"*>/</param>

<param name=*"actionName"*>carshow</param>

</result>

<!-- <result name="success">/jsp/login.jsp</result>-->

</action>

<!-- 显示购物车 -->

<action name=*"carshow"* class=*"com.bookshop.action.carAction"* method=*"showcar"*>

<result name=*"success"*>/jsp/cart.jsp</result>

</action>

<action name=*"locale"* class=*"com.bookshop.action.LoginLocale"* method=*"execute"*>

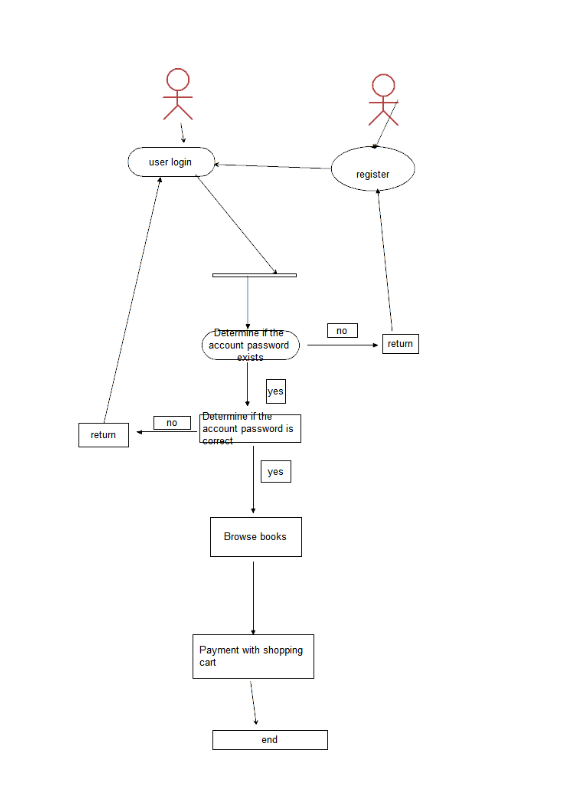
<result name=*"success"*>/jsp/register.jsp</result>

</action>

</package>

</struts>

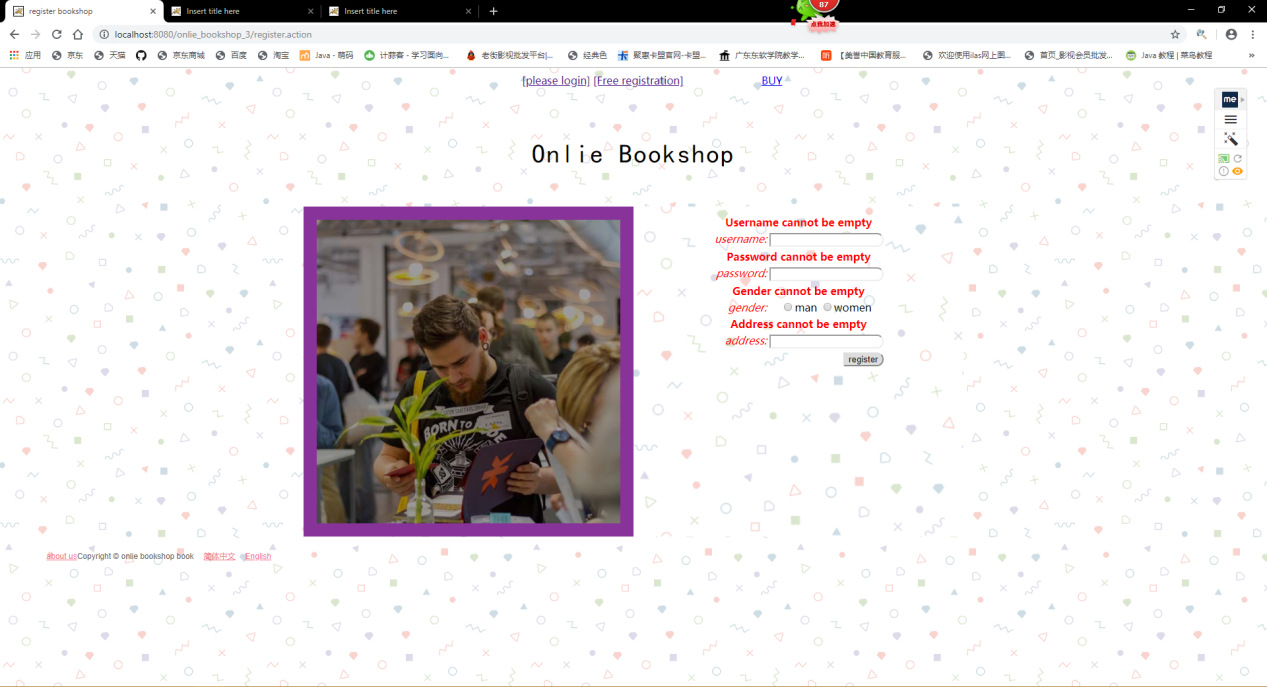
1. Diagram



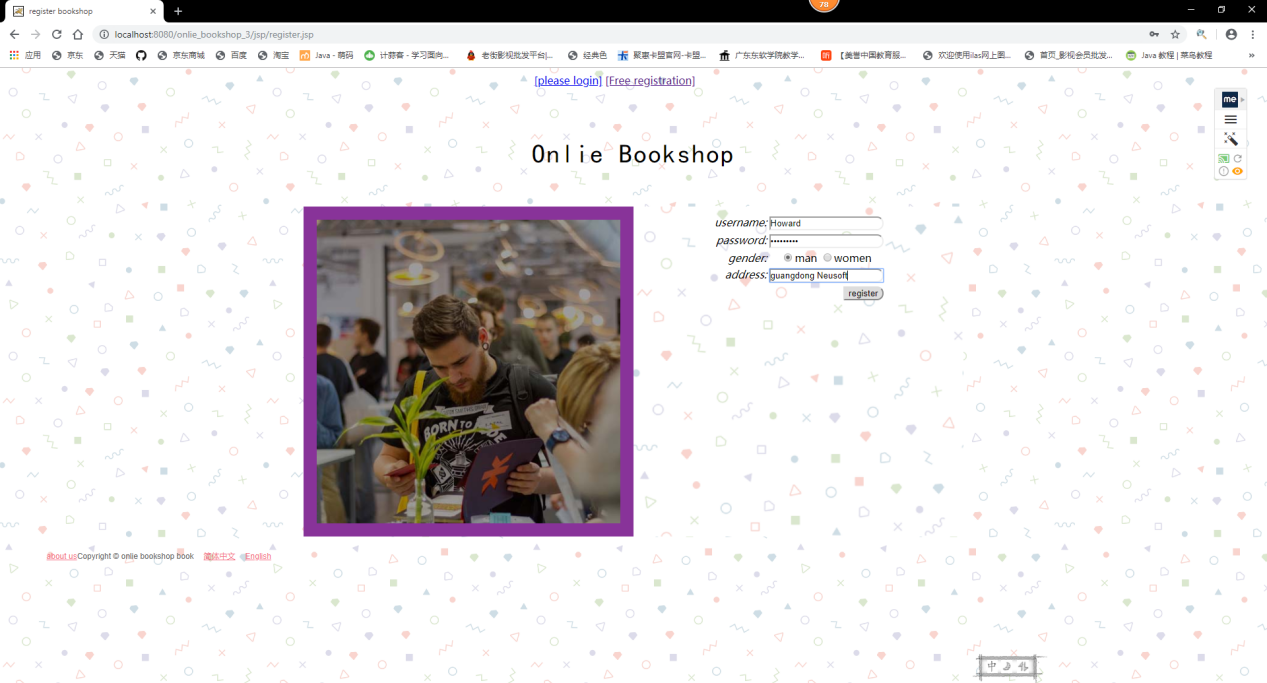
1. Screenshots

4.1 register.jsp

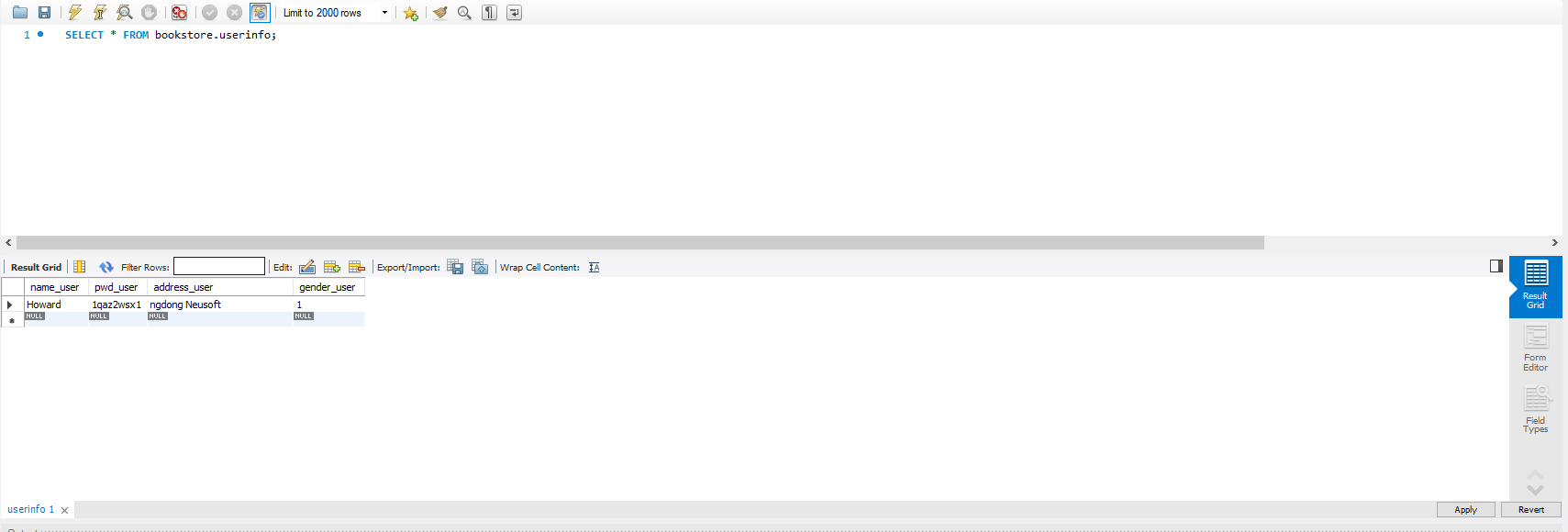
validation



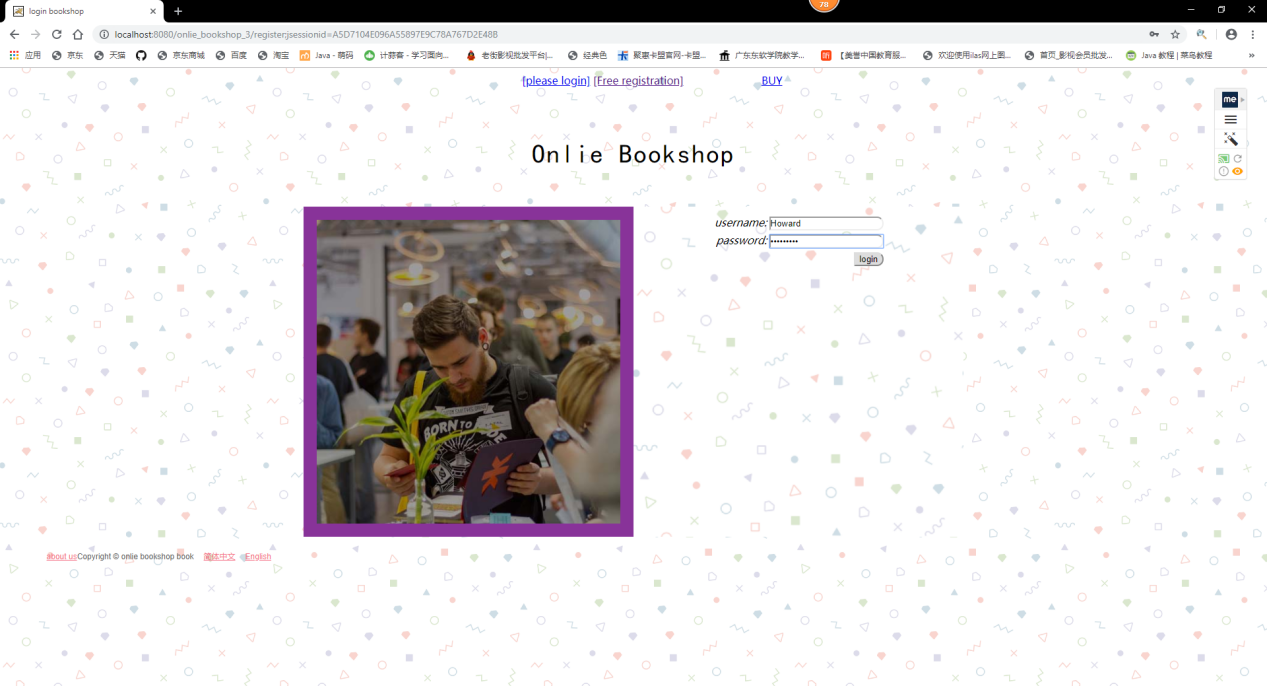
4.2 register.jsp



4.3 database

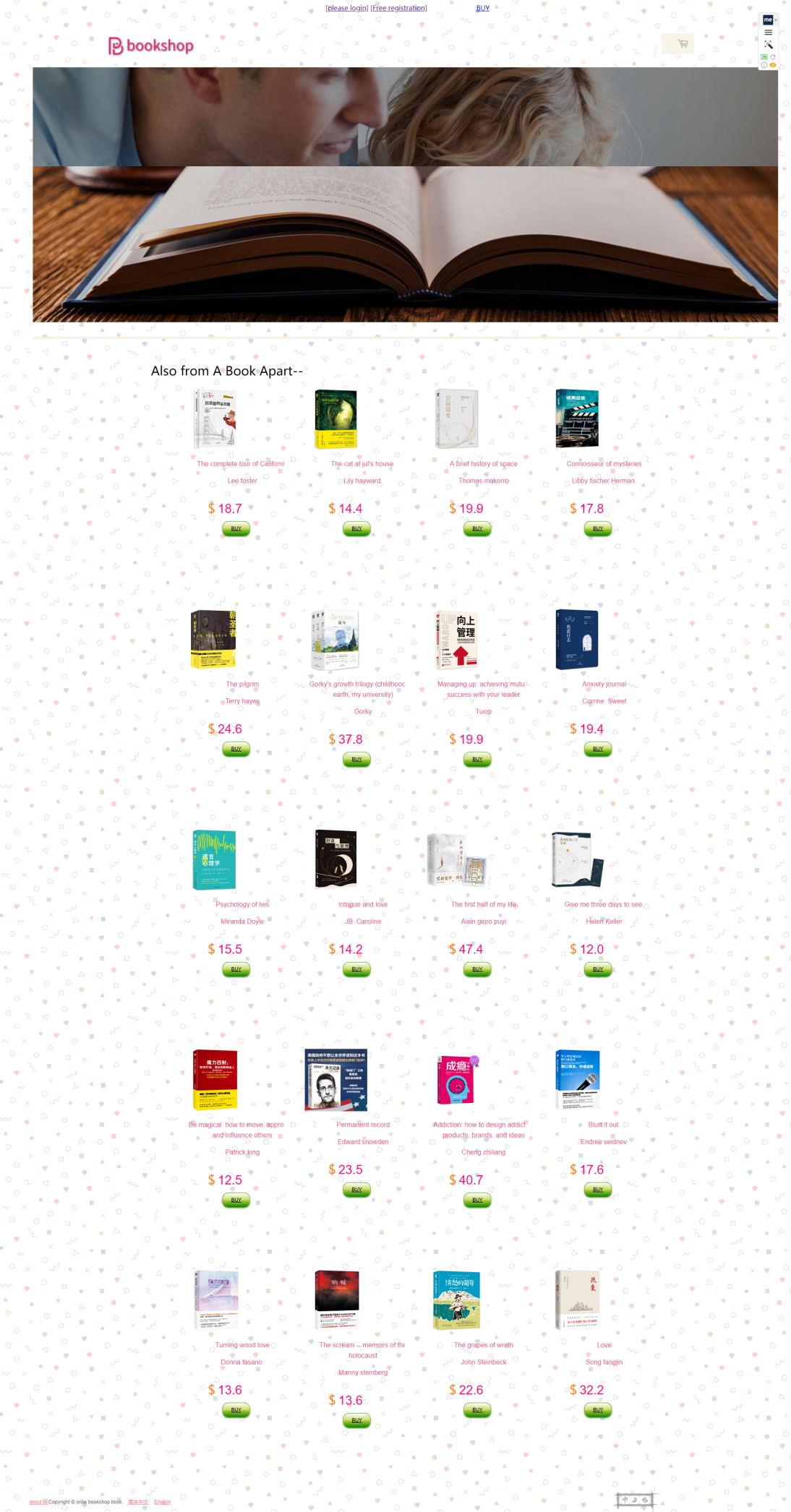


4.4 login.jsp



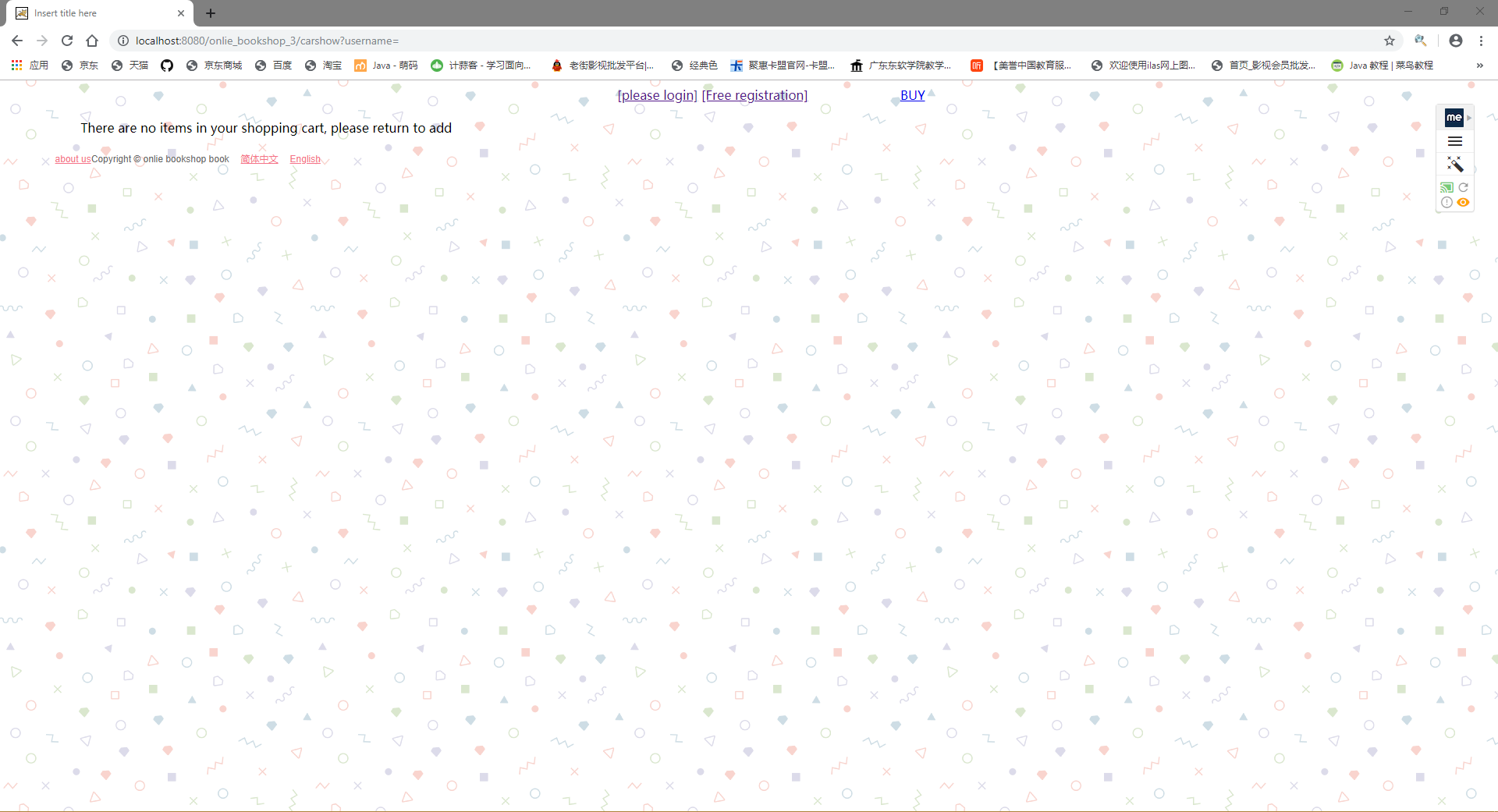
4.5 main.jsp

Book’s price ，name and picture



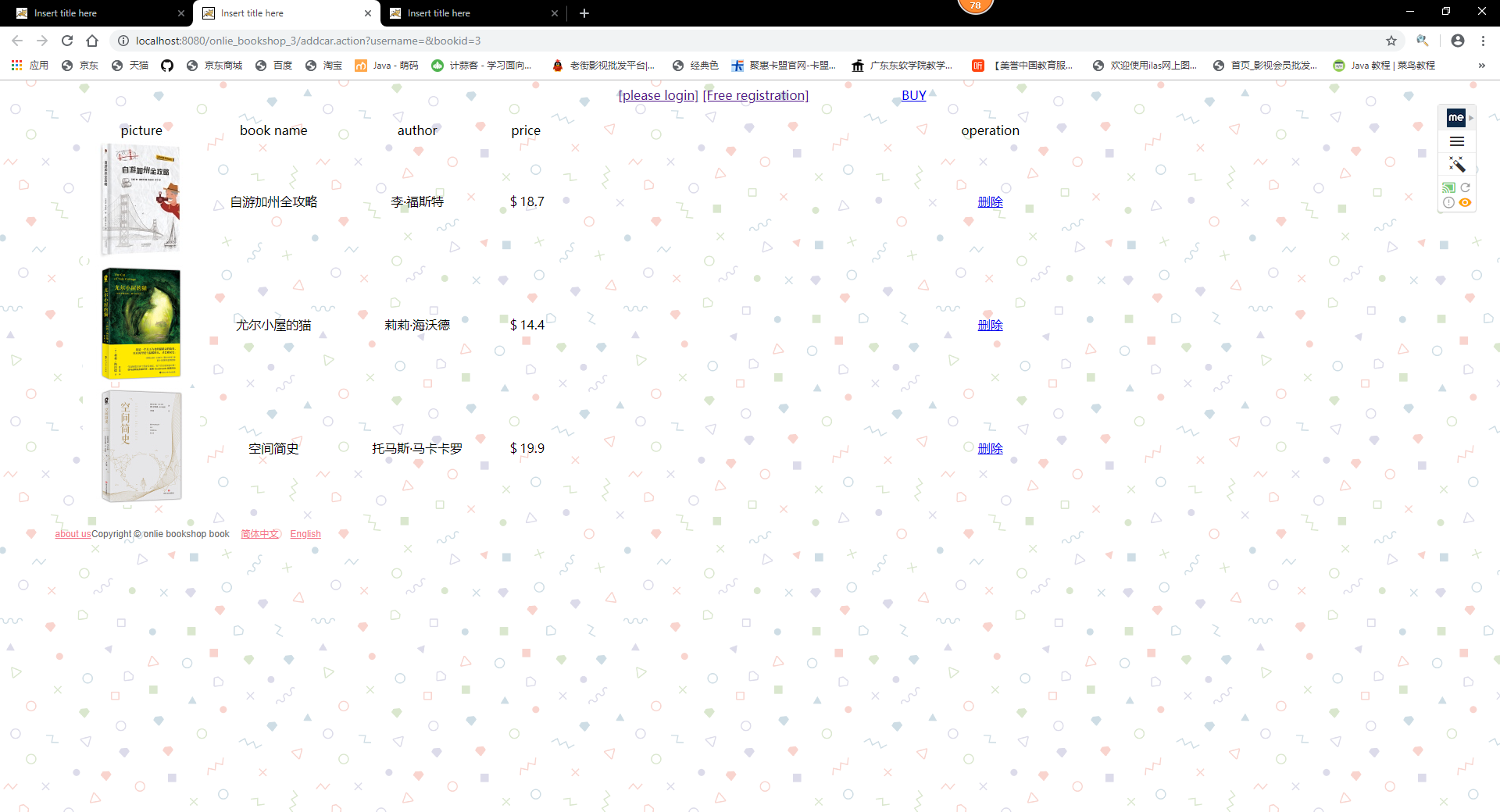
4.6 car.jsp

The shopping no books



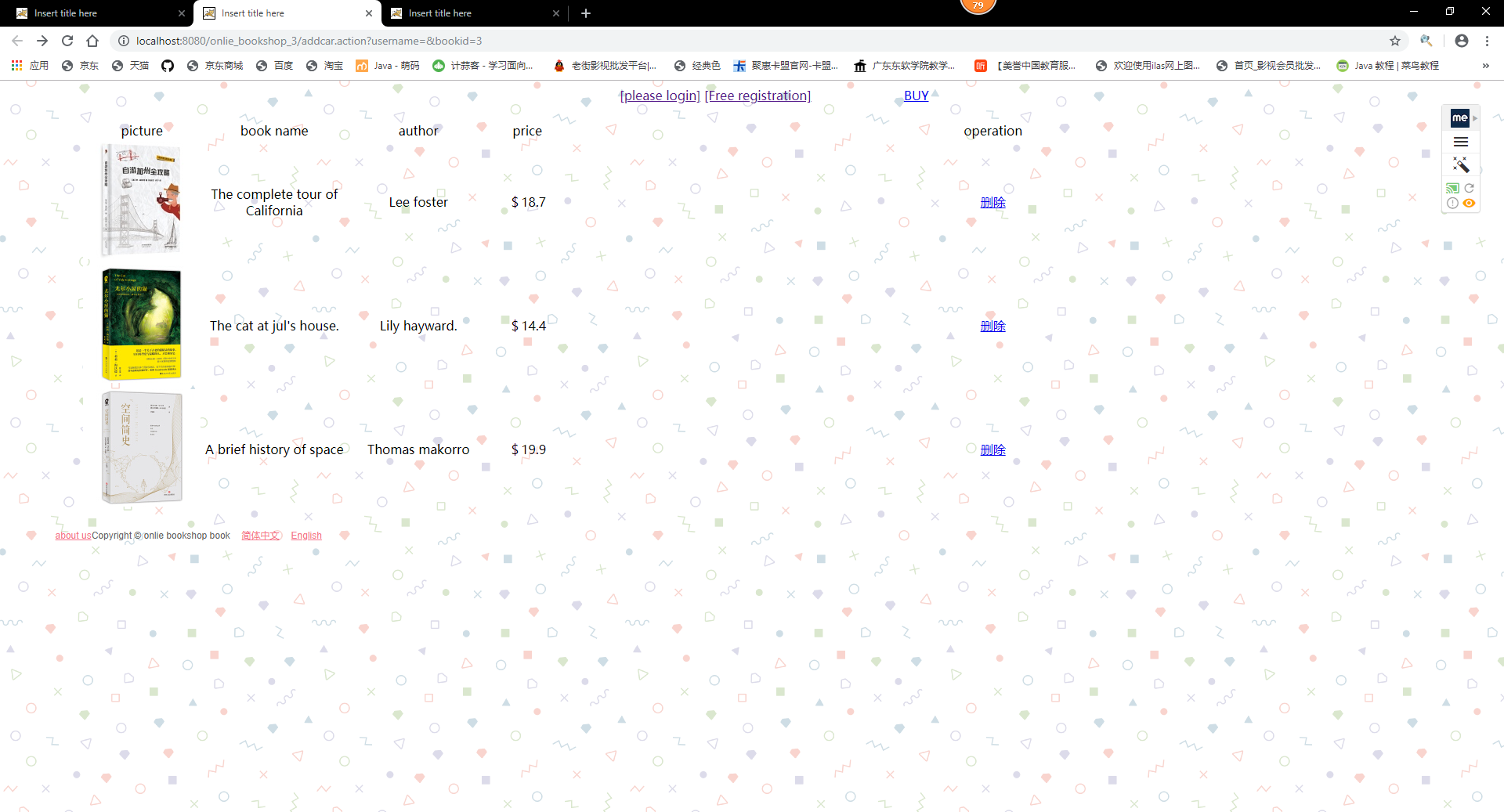
4.7 car.jsp chinese page

The shoppingcar has books

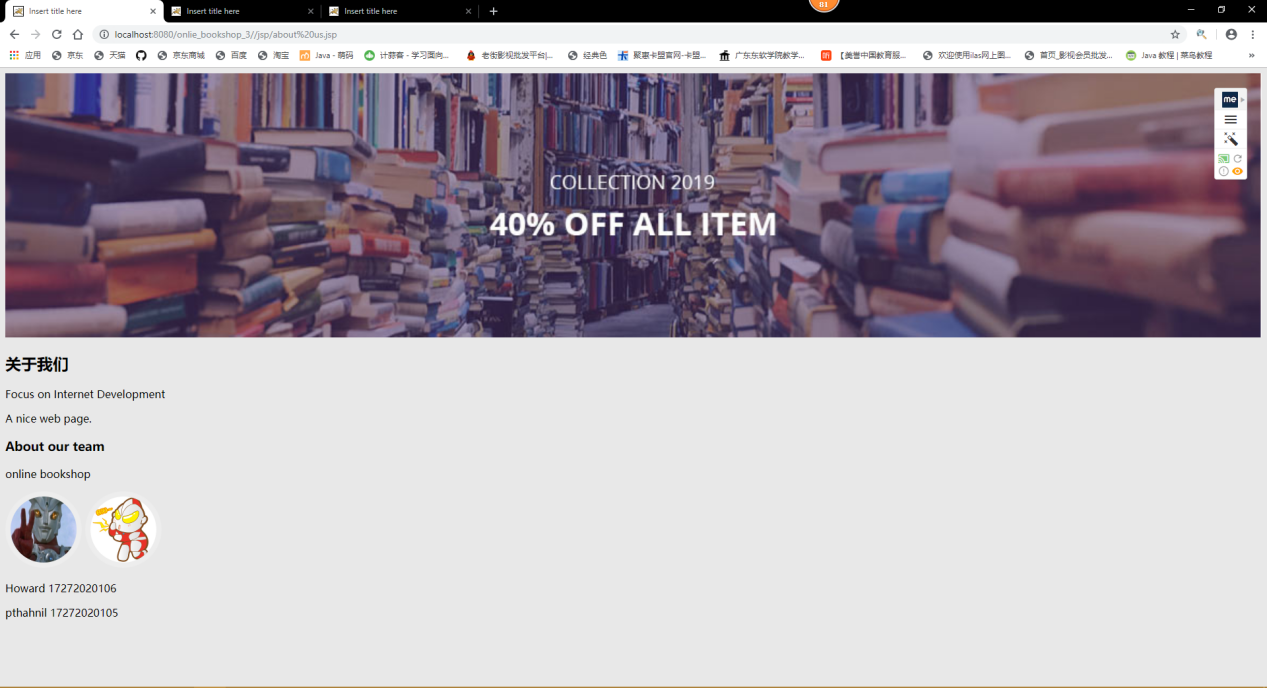


4.8 car.jsp english page

The shoppingcar has books



4.9 about us .jsp



1. Conclusion

Through this experimental assignment, I have a general understanding of the design and production of a struts2 online bookstore. I learned the steps, what to prepare, and what skills to learn to build a struts2 online bookstore.

As the saying goes, "true knowledge comes from practice". Studying textbook knowledge is far from enough. In the process of practice, I encountered many problems. I met a lot of puzzling questions. There are a lot of problems, I looked up all kinds of information, card for a long time can not solve, resulting in the delay of the project schedule. Finally, I still ask for help from the students around me and discuss with each other to solve the problem. I think the discussion of communication learning methods, the effect of group.

Through this experiment, I learned to make a struts2 online bookstore. However, the learning mode and thinking mode contained in it are more worthy for us to learn. I will also in the coming days, carry forward the scientific spirit, serious thinking, study hard, and constantly improve their level in all aspects!