



吕梁学院

LULIANG UNIVERSITY

分类号: _____

密 级: _____

课程设计

题 目: _____ 图书管理销售系统

系 别: _____ 计算机科学与技术系

专业年级: _____ 软件工程 1902 班

姓 名: _____ 徐亮、杨龙

_____ 经卫杰、靳宇鑫

组 长: _____ 孔德华

指导教师: _____ 刘书臣

2021 年 12 月 13 日

目录

第 1 章 系统开发的技术	- 1 -
1.1 JSP	- 1 -
1.2 MYSQL	- 1 -
1.3 IDEA	- 2 -
1.4 谷歌验证码 kaptcha	- 2 -
1.5 数据库连接池	- 2 -
1.6 JDBC 工具类——JdbcUtils	- 2 -
1.7 Filter 过滤器	- 3 -
第 2 章 系统分析	- 4 -
2.1 可行性分析	- 4 -
2.1.1 经济可行性分析	- 4 -
2.1.2 技术可行性分析	- 4 -
2.2 功能需求分析	- 4 -
2.3 数据库需求分析	- 6 -
2.3.1 数据库概念结构设计	- 6 -
2.3.2 数据库主要表的建设	- 8 -
2.4 系统设计	- 10 -
2.4.1 系统功能设计	- 10 -
第 3 章 系统详细设计与实现	- 12 -
3.1 主界面	- 12 -
3.2 登录界面	- 16 -
3.3 注册界面	- 18 -
3.4 注册成功界面	- 21 -
3.5 登录成功界面	- 21 -
3.6 用户身份进入主界面	- 22 -
3.7 按区间价格查找图书界面	- 23 -
3.8 翻页的实现界面	- 24 -
3.9 购买图书添加到购物车界面	- 27 -
3.10 查看购物车界面	- 28 -
3.11 结算成功生成订单界面	- 31 -
3.12 以用户身份查看订单状态界面	- 33 -
3.13 订单详情界面	- 35 -
3.14 系统开发人员信息界面	- 36 -
3.15 用户身份查看后台界面	- 38 -
3.16 管理员身份查看后台界面	- 40 -
3.17 图书管理界面	- 41 -
3.18 图书的添加界面	- 43 -
3.19 图书的添加成功界面	- 44 -
3.20 图书的删除界面	- 45 -
3.21 图书的删除提醒界面	- 45 -

3.22 图书的删除以及删除提醒界面	- 46 -
3.23 以管理员身份查看在后台的订单管理界面	- 47 -
第4章 系统测试	- 49 -
4.1 测试用例	- 49 -
4.2 测试结论	- 52 -
第5章 总 结	- 53 -

第 1 章 系统开发的技术

本系统是基于 B/S（浏览器/服务器）模式的 Web 信息管理系统，以 Windows 10 作为系统平台，采用 JSP 作为主要的程序设计语言，将 MY SQL 数据库作为后台数据库进行系统的开发。

1.1 JSP

JSP 在本系统的设计和实现中有重要的作用。JSP 全名为 Java Server Pages，其根本是一个简化的 Servlet 设计。Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的程序段，然后将执行结果连同 JSP 文件中的 HTML 代码一起返回给客户端。插入的 Java 程序段可以操作数据库、重新定向网页等，以此来实现建立动态网页所需要的功能。

本系统以 JSP 作为主要的程序设计语言，通过 Tomcat 服务器实现动态网页等技术，完成系统的动态网页部分，并且以 JavaScript 的各种事件，DOM 对象和正则表达式为辅来提高网页的质量，用到 JSP 中的 EL 表达式来使 jsp 程序代码很简练，通过 Filter 过滤器实现各种页面跳转的权限管理。

1.2 MYSQL

MySQL 是一个精巧的 SQL 数据库管理系统，虽然它不是开放源代码的产品，但在某些情况下你可以自由使用。由于它的强大功能、灵活性、丰富的应用编程接口（API）以及精巧的系统结构，受到了广大自由软件爱好者甚至是商业软件用户的青睐，特别是与 Apache 和 PHP/PERL 结合，为建立基于数据库的动态网站提供了强大动力。MySQL 是一个真正的多用户、多线程 SQL 数据库服务器。SQL（结构化查询语言）是世界上最流行的和标准化的数据库语言。MySQL 是以一个客户机/服务器结构的实现，它由一个服务器守护程序 mysql 和很多不同的客户程序和库组成。

本系统使用 **MYSQL** 作为后台数据库进行系统开发，以数据路连接池来使连接数据库更为方便和高效的存储 web 页面上的各种信息。

1.3 IDEA

IDEA 全称 **IntelliJ IDEA**，是 java 编程语言开发的集成环境。**IntelliJ** 在业界被公认为最好的 java 开发工具，尤其在智能代码助手、代码自动提示、重构、**JavaEE** 支持、各类版本工具(**git**、**svn** 等)、**JUnit**、**CVS** 整合、代码分析、 创新的 **GUI** 设计等方面的功能可以说是超常的。当然在做 **Web** 前端设计的时候也是很方便很高效的。

1.4 谷歌验证码 kaptcha

Kaptcha 是一个非常实用的验证码生成工具，可以通过配置生成多样化的验证码。以图片的形式显示，从而无法进行复制粘贴。提高注册界面的安全性。主要通过 **kaptcha-2.3.2.jar** 和去 **xml** 文件中配置以下信息才能正常使用。

1.5 数据库连接池

连接池的作用就是为了提高性能。连接池的作用：连接池是将已经创建好的连接保存在池中，当有请求来时，直接使用已经创建好的连接对数据库进行访问。这样省略了创建连接和销毁连接的过程。这样性能上得到了提高。

1.6 JDBC 工具类——**JdbcUtils**

JdbcUtils 一个简单的数据库常用操作工具类，它可以实现对数据库操作语句的任何一个方法的测试，通过给方法传入一些参数就可以通过参看数据库来检测方法写的是否有问题，这是一个很高效的测试 **jdbc** 的工具类。

1.7 Filter 过滤器

Filter 过滤器它是 JavaWeb 的三大组件之一。三大组件分别是：Servlet 程序、Listener 监听器、Filter 过滤器；Filter 过滤器它是 JavaEE 的规范，也就是接口；Filter 过滤器它的作用是：拦截请求，过滤响应。拦截请求常见的应用场景有：权限检查、日记操作、事务管理。而在本系统中主要应用域权限检查操作。

第 2 章 系统分析

本章进行系统分析，先对系统进行可行性分析，包括经济、技术和操作可行性分析；接着对系统进行简单的需求分析，得出系统的基本功能等；最后对系统进行数据库需求分析，包括数据库概念结构分析、逻辑结构分析和表的设计等。

2.1 可行性分析

本系统立足图书管理销售的实际工作，从实际出发，结合正确的理论基础，具有很强的现实意义。现对本系统的经济可行性、技术可行性和操作可行性作简要分析。

2.1.1 经济可行性分析

因为本系统只是一般的图书管理销售系统，对系统的管理与操作性要求不是很高，系统管理员可征集有相关专业基础学生义务担任或作为学校勤工助学的岗位。而且系统的经费不高，内部逻辑并不复杂，可由学生担任一部分的开发工作。学生可以使用自己的电脑，机房也有电脑可供使用，而且软件方面的成本接近零成本，所以在经济方面是可行的。

2.1.2 技术可行性分析

本系统采用基于 B/S（浏览器/服务器）的框架结构，数据库采用 MY SQL，采用 JSP 技术实现系统模块功能。B/S 是现在主流的 Web 开发框架，成本降低的同时选择更多。经过详细的分析，设计模式、技术、开发环境和人员配备都可满足系统开发的条件。

2.2 功能需求分析

《图书管理销售系统》是针对网上书店卖家销售书籍和买家购买书籍的系统，通过该系统方便了书籍的售卖和购买。

经过调研，《图书管理销售系统》包含如下主要功能：

- (1) 系统登陆
- (2) 系统注册
- (3) 开发人员信息
- (4) 购物车，包括删除商品、修改数量、清空购物车、去结账。
- (5) 注销
- (6) 我的订单，包括订单信息和订单查看
- (7) 图书分页
- (8) 价格搜索
- (9) 后台管理包含图书管理和订单管理。

其中图书管理包括：图书分页、图书修改、图书删除、图书添加。

订单管理包括订单信息、是否发货和订单详情

功能模块图，如图 2-1 所示：

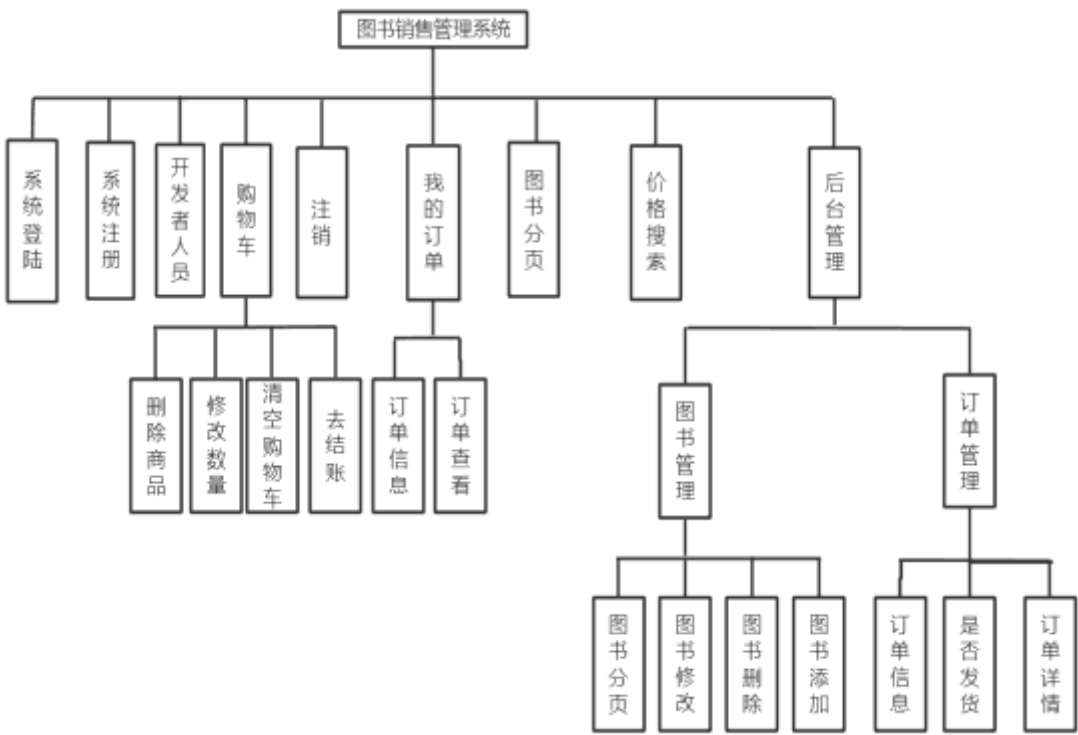


图 2-1 功能模块图

2.3 数据库需求分析

设计一个性能良好的适应本系统的数据库系统，明确系统对应用环境的要求是首要的和基本的。因此，应该充分了解“图书管理”的工作概况及工作流程，明确各种需求，产生数据流图和数据字典，然后在此基础上确定系统的功能。必须注意的是，系统的开发还应考虑系统的安全性：1. 系统使用者必须有账户。2. 卖家系统管理员和消费者用户登陆后所显示的页面不相同。

2.3.1 数据库概念结构设计

针对图书管理销售系统的需求，分析系统所涉及的主要的实体及各自属性，综合它们之间的关系，总结出以下信息：

系统需要设计如下数据项来满足实体的需要：

- （1）用户信息，属性包括：用户编号、用户名、电话、密码、邮箱地址。
- （2）图书信息，属性包括：图书编号、图书名称、作者、价格、销量、库存、书的图片路径。
- （3）图书订单，属性包括：订单号、订单时间、总金额、物流状态、用户编号。
- （4）订单项目，属性包括：图书编号、图书名称、图书单价、图书总金额、图书数量、订单号

由上述数据项，可以设计出能够满足用户需求的各种实体，以及他们之间的关系

用户信息实体概念图，如图 2-2 所示：

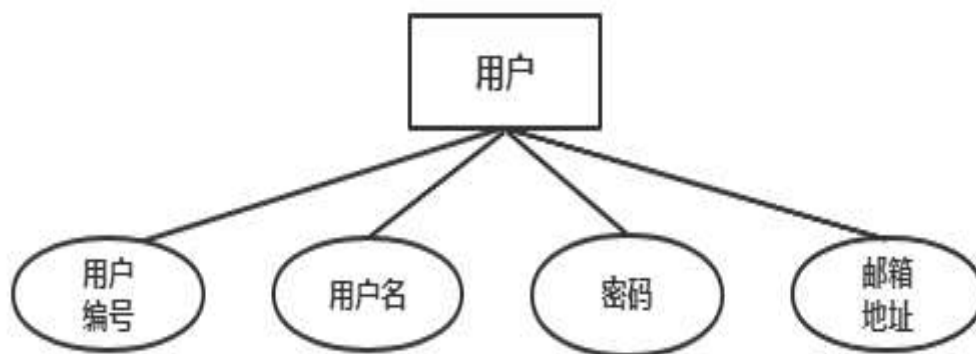


图 2-2 用户信息实体概念图

图书信息实体概念图，如图 2-3 所示：

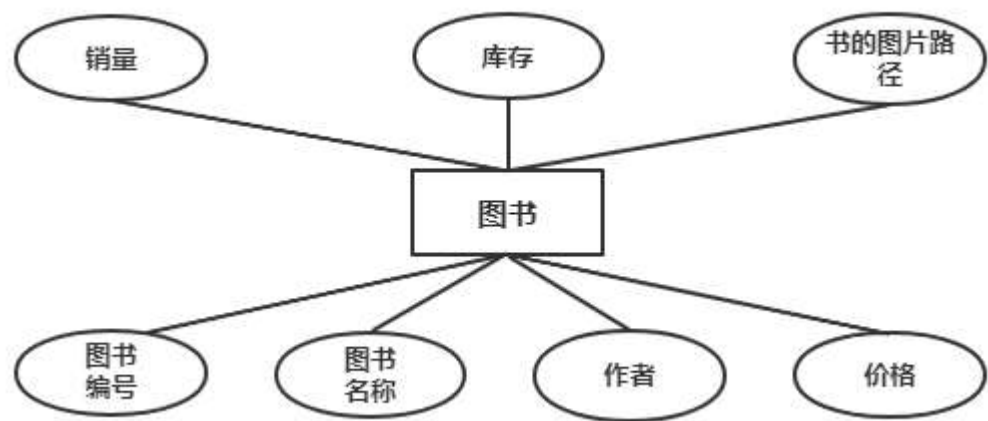


图 2-3 图书信息实体概念图

图书订单实体概念图如图 2-4 所示：

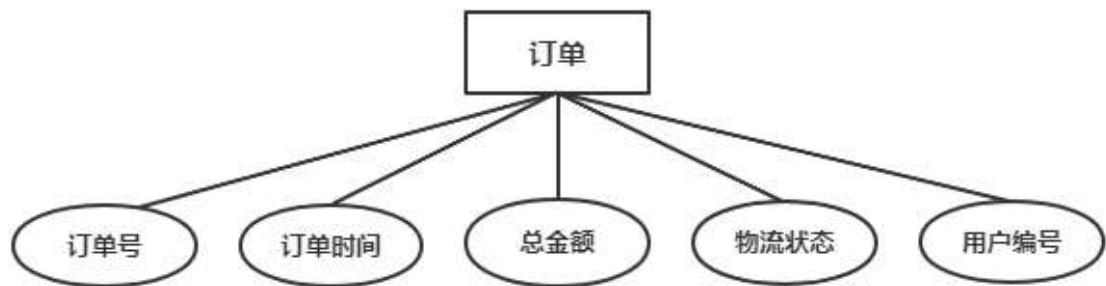


图 2-4 图书订单实体概念图

订单项目实体概念图如图 2-5 所示：

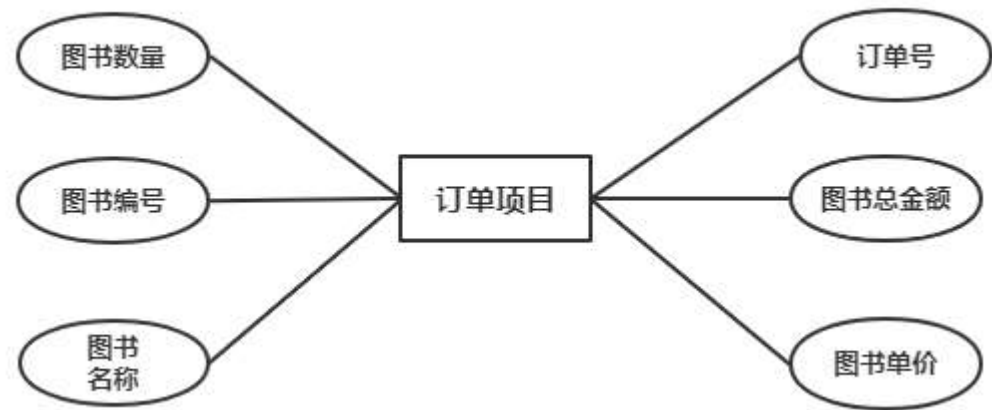


图 2-5 订单项目实体概念图

综上四个实体概念图，画出 E-R 图，如图 2-6 所示：

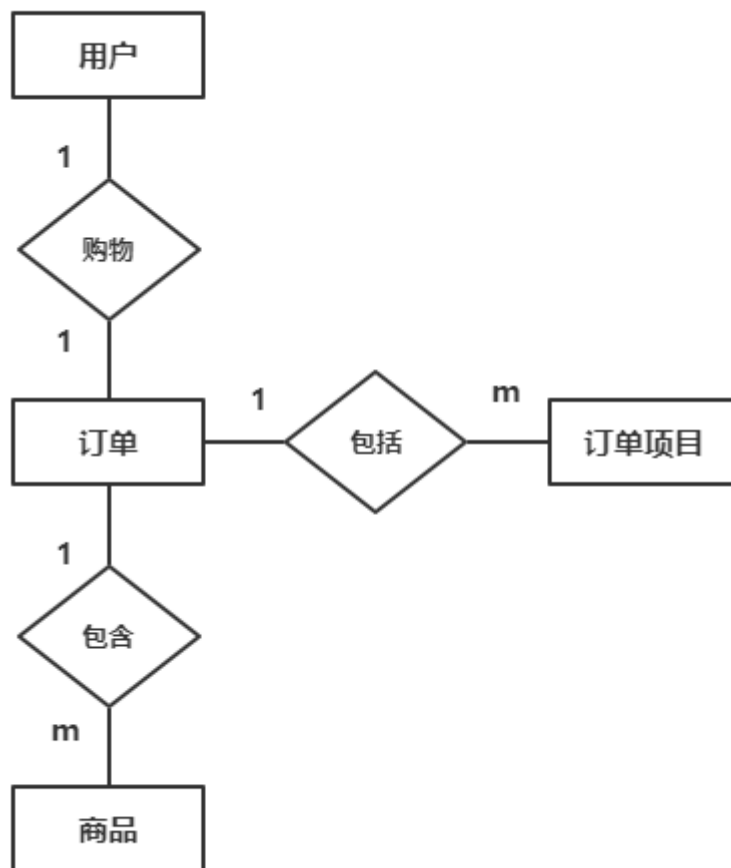


图 2-6 E-R 图

2.3.2 数据库主要表的建设

本系统中数据库采用 MY SQL 作为数据库。首先我们在 MY SQL 创建一个数据库，命名为“book”。

选中 book 数据库，向数据库中添加 4 个表，分别为 t_user、t_book、t_order、t_order_item，分别对应用户注册信息表、图书信息表、图书订单表、订单项目表。系统的 4 个表在系统 E-R 图中有部分体现先详细叙述如下。

t_user 为用户注册信息表，其详细信息如表 2-1 所示：

表 2-1 用户注册信息表

编号	字段名称	数据类型	说明	允许空
1	id	int	用户编号，主键	
2	username	varchar(50)	用户名称	
3	password	varchar(32)	密码	
4	email	varchar(50)	邮箱地址	√

图书信息表记录图书信息包含的属性有 id、name、author、price、sales、stock、img_path 等。其中 id 为主键。

t_book 为图书信息表列表，其详细信息如表 2-2 所示：

表 2-2 图书信息表

编号	字段名称	数据类型	说明	允许空
1	id	int(11)	图书编号，主键	
2	name	varchar(50)	图书名称	
3	author	varchar(50)	作者	
4	price	decimal(11,2)	价格	
5	sales	int(11)	销量	
6	stock	int(11)	库存	
7	img_path	varchar(200)	书的图片路径	

图书订单表包含的属性有 order_id、create_time、price、status、user_id 等，以 order_id 为主键，记录订单编号。

t_order 为图书订单表列表，其详细信息如表 2-3 所示：

表 2-3 图书订单表

编号	字段名称	数据类型	说明	允许空
1	order_id	varchar(50)	订单号，主键	
2	create_time	varchar(20)	订单时间	
3	price	char(50)	总金额	
4	status	varchar(20)	物流状态	
5	user_id	varchar(50)	用户编号，外键	

订单项目表包含的属性有 id、name、price、total、count、order_id，以 id 为主键，记录信息。

t_order_item 为订单项目表，有关订单项目表的详细信息如表 2-4 所示：

表 2-4 订单项目表

编号	字段名称	数据类型	说明	允许空
1	id	int	图书编号，主键（自动生成）	
2	name	varchar(30)	图书名称	
3	price	decimal(11,2)	图书单价	√
4	total	decimal(11,2)	图书总金额	√
5	count	int	图书数量	
6	order_id	Vaechar(50)	订单号，外键	

2.4 系统设计

2.4.1 系统功能设计

图书管理销售系统分为九个子模块，即系统登陆、系统注册、开发者人员查看、购物车、注销、我的订单、图书分页、价格搜索、后台管理。以下对七个子模块做详细叙述并画出系统功能结构模块图：

（1）系统登陆

- 1.输入错误的信息后表单回显并提示错误原因。
- 2.输入正确后在数据库中验证是否存在该用户，存在的话就跳转到欢迎页面，不存在的话就提示请注册会员。
- 3.有注册会员的连接可供选则

（2）注册系统

- 1.输入姓名、密码、确认密码、邮箱、验证码（看不清可刷新），当用户名输入不合法，密码输入不合法或前后不一致、以及邮箱和验证码的输入，只要有一项不正确都会进行表单回显，并提示错误原因。
- 2.输入正确后，会把信息存储到数据库中，并跳转到欢迎页面。

（3）开发者人员

- 1.开发者人员的连接一点，就可以查看到该系统的人员名单。

（4）购物车

- 1.首先通过主页面的图书浏览页面点击“添加到购物车”便可以把该书籍存储到购物车当中（以 Session 的形式把书籍的各种信息都存储起来），并在图书浏览处显示有几本书存储到了购物车当中，最后添加一本书的名字也会做显示。
- 2.当没有商品的时候会提示用户去主页面挑选商品。
- 3.当有商品的时候，有删除商品功能，点击它就可以实现商品的删除。

4.修改购买商品的数量

5.清空购物车

6.结账：结完账之后，便可生成订单号，在我的订单里可查看详细信息。

(5) 注销

1.当点击注销时：就会把当前用户的信息清空并跳转到最初始页面。

(6) 我的订单

1.订单信息显示

2.订单项的详细信息显示

(7) 图书分页

1.点击首页：便可以跳转到首页

2.点击尾页：便可以跳转到尾页

3.点击中间非括号展示的数字，便可以跳转到相应地页数

4.在输入框中输入所要跳转的页数便可以定位到该页。

(8) 价格搜索

1.往区间里面填写想要的价格，便可以在图书阅览处展示相应的图书。

(9) 后台管理

1.图书管理

1.1、图书分页

1.2、图书修改

1.3、图书删除

1.4、图书添加

2.订单管理

2.1、订单信息

2.2、是否发货

2.3、订单详情

第 3 章 系统详细设计与实现

系统界面分为消费者基本信息管理，消费者基本信息有效性审核，消费者项目申请，消费者项目验证进度查询，管理员项目审核，系统登录和系统用户管理。注册界面可以进行学生和管理员两个不同角色的注册；只有在注册之后才可以进行登录，需要输入各自的学号或工号以及对应的密码，登陆页面可供管理员进行登录；消费者登录后可以进行图书的购买，管理员登录后可以对学生进行管理和项目进行审核。

3.1 主界面

从系统主界面处，开始旅程引入系统。主界面展示四个功能：登陆，注册，通过区间查找图书，通过分页条来实现书籍的翻页。主界面如图 3-1 所示：



图 3-1 主界面

主要代码为：

```
<body>

<div id="header">

    <span class="wel_word" >网上书城</span>

</div>
```



```

<%--如果用户还没登陆，显示[注册和登陆]的两个选项--%>

<c:if test="${empty sessionScope.user}">

    <a href="pages/user/login.jsp">登录</a> |

    <a href="pages/user/regist.jsp">注册</a> |

</c:if>

<%--如果用户已经登陆，则显示登录成功之后的用户信息--%>

<c:if test="${not empty sessionScope.user}">

    <span>欢迎<span
class="um_span">${sessionScope.user.username}</span>光临书城</span>

    <a
href="orderServlet?action=showMyOrders&id=${sessionScope.user.id}">我的订单</a> |

    <a href="userServlet?action=logout">注销</a> |

    <a href="pages/developers/developers.jsp">系统开发人员</a> |

</c:if>

<a href="pages/cart/cart.jsp">购物车</a> |

<a
href="manager/bookServlet?action=manager&name=${sessionScope.user.username}">后台管理
</a>

</div>

</div>

<div id="main">

<div id="book">

<div class="book_cond">

<form action="client/bookServlet" method="get">

    <input type="hidden" name="action" value="pageByPrice">

    价格: <input id="min" type="text" name="min" value="${param.min}"> 元

    -

    <input id="max" type="text" name="max" value="${param.max}"> 元

    <input type="submit" value="查询" />

</form>

```

```

</div>

<div style="text-align: center">

    <c:if test="${empty sessionScope.cart.items}">

        <%--购物车为空的话--%>

        <span></span>

        <div>

            <span style="color: red">当前购物车为空</span>

        </div>

    </c:if>

    <c:if test="${not empty sessionScope.cart.items}">

        <%--购物车非空输出--%>

        <span>您的购物车中有${sessionScope.cart.totalCount}件商品</span>

        <div>

            您刚刚将<span style="color: red">${sessionScope.lastName}</span>
加入到了购物车中

        </div>

    </c:if>

</div>

<c:forEach items="${requestScope.page.items}" var="book">

<div class="b_list">

    <div class="img_div">

    </div>

    <div class="book_info">

        <div class="book_name">

            <span class="sp1">书名:</span>

            <span class="sp2">${book.name}</span>

        </div>

        <div class="book_author">

```

```

        <span class="sp1">作者:</span>

        <span class="sp2">${book.author}</span>

    </div>

    <div class="book_price">

        <span class="sp1">价格:</span>

        <span class="sp2">¥ ${book.price}</span>

    </div>

    <div class="book_sales">

        <span class="sp1">销量:</span>

        <span class="sp2">${book.sales}</span>

    </div>

    <div class="book_amount">

        <span class="sp1">库存:</span>

        <span class="sp2">${book.stock}</span>

    </div>

    <div class="book_add">

        <button bookId="${book.id}" class="addToCart">加入购物车

</button>

    </div>

</div>

</div>

</div>

</div>

</div>

<!--静态包含分页条-->

<% @include file="/pages/common/page_nav.jsp"%>

</div>

<!--静态包含页脚内容-->

<% @include file="/pages/common/footer.jsp"%>

</body>

```

3.2 登录界面

所有访客均可点击登录页面，输入合法的用户名称和用户密码进行登录，登录页面里面也有四个功能：登陆不成功后的表单回显，出错信息提示，注册功能，以及登陆后在数据库中匹配，看是否有该用户，从而去判断是否发生页面跳转，登录页面如图 3-2 所示：



图 3-2 登录界面

主要代码为

```
<body>

    <div id="login_header">

    </div>

    <div class="login_banner">

        <div id="l_content">

            <span class="login_word">欢迎登录</span>

        </div>

        <div id="content">
```

```

<div class="login_form">

    <div class="login_box">

        <div class="tit">

            <h1>书城会员</h1>

            <a href="pages/user/regist.jsp">立即注册</a>

        </div>

        <div class="msg_cont">

            <b></b>

            <span class="errorMsg">

                <%--<%=request.getAttribute("msg")==null?"请输入用
户名和密码":request.getAttribute("msg")%>--%>

                ${empty requestScope.msg ? "请输入用户名和密码
":requestScope.msg}

            </span>

        </div>

        <div class="form">

            <form action="userServlet" method="post">

                <input type="hidden" name="action"
value="login"/><!--添加隐藏域，方便两个 servlet 合并成一个 servlet-->

                <label>用户名称: </label>

                <input class="itxt" type="text" placeholder="请输入用
户名" autocomplete="off" tabindex="1" name="username"

                <%--
value="<%=request.getAttribute("username")==null?"":request.getAttribute("username")%>"--%>

                value="${requestScope.username}"

                /> <!--登陆的错误回显用户名--><!--他不用判断是否
为空，因为这样的表达式就是不会给出 null,而是直接为空串-->

                <br />

                <br />

                <label>用户密码: </label>

                <input class="itxt" type="password" placeholder="请输
入密码" autocomplete="off" tabindex="1" name="password" />

```

```

        <br />

        <br />

        <input type="submit" value="登录" id="sub_btn" />

    </form>

</div>

</div>

</div>

</div>

<% @ include file="/pages/common/footer.jsp"%>

</body>

```

3.3 注册界面

注册页面里有以下功能：登陆不成功的表单回显，出错信息提示，匹配用户名，邮箱是否合法，数据密码是否一致，动态验证码的刷新以及校验验证码输入是否正确，还有更新数据库，发生页面跳转。注册页面如图 3-3 所示：



图 3-3 注册界面

主要代码为

```

<body>

<div id="login_header">

```

```
</div>

<div class="login_banner">

    <div id="l_content">

        <span class="login_word">欢迎注册</span>

    </div>

    <div id="content">

        <div class="login_form">

            <div class="login_box">

                <div class="tit">

                    <h1>注册书城会员</h1>

                    <span class="errorMsg">

                        ${ requestScope.msg }

                    </span>

                </div>

                <div class="form">

                    <form action="userServlet" method="post">

                        <input type="hidden" name="action" value="regist">

                        <label>用户名称: </label>

                        <input class="itxt" type="text" placeholder="请输入用户名"

                            value="${requestScope.username}"

                            autocomplete="off" tabindex="1" name="username"

id="username" />

                        <br />

                        <br />

                        <label>用户密码: </label>

                        <input class="itxt" type="password" placeholder="请输入密码"

                            autocomplete="off" tabindex="1" name="password"

id="password" />

                        <br />

                        <br />
```

```

        <label>确认密码: </label>

        <input class="itxt" type="password" placeholder="确认密码"

                autocomplete="off" tabindex="1" name="repwd"

id="repwd" />

        <br />

        <br />

        <label>电子邮件: </label>

        <input class="itxt" type="text" placeholder="请输入邮箱地址"

                value="{requestScope.email}"

                autocomplete="off" tabindex="1" name="email"

id="email" />

        <br />

        <br />

        <label>验证码: </label>

        <input class="itxt" type="text" name="code" style="width: 80px;"

id="code" />

        <br />

        <br />

        <input type="submit" value="注册" id="sub_btn" />

    </form>

</div>

</div>

</div>

</div>

<@ include file="/pages/common/footer.jsp"%>

</body>

```


3.4 注册成功界面

注册成功后和登陆成功后跳转的欢迎界面，如图 3-4 所示：



图 3-4 注册成功界面

主要代码

```
<body>

    <div id="header">

        <span class="wel_word"></span>

        <% @ include file="/pages/common/login_success_menu.jsp"%>

    </div>

    <div id="main">

        <h1>注册成功! <a href="index.jsp">转到主页</a></h1>

    </div>

    <% @ include file="/pages/common/footer.jsp"%>

</body>
```

3.5 登录成功界面

将注册成功的用户名和密码输入登陆界面显示登录成功，如图 3-5 所示：



图 3-5 登录成功界面

主要代码为

```
<body>

    <div id="header">

        <% @ include file="/pages/common/login_success_menu.jsp"%>

    </div>

    <div id="main">

        <h1>欢迎回来 <a href="index.jsp">转到主页</a></h1>

    </div>

    <% @ include file="/pages/common/footer.jsp"%>

</body>
```

3.6 用户身份进入主界面

以用户身份进入主界面主要有以下功能：注销该用户、查看系统开发人员、购物车查看、后台管理、查看我的订单、当然还有按价格来区间查找图书，以及翻页功能和页面定向跳转的功能的实现。以用户身份进入主界面的显示效果，如图 3-6 所示：



图 3-6 用户身份进入主界面

3.7 按区间价格查找图书界面

按照区间价格来查找图书，如图 3-7 所示：



图 3-7 按区间价格查找图书界面

主要代码为

```
<div class="book_cond">

    <form action="client/bookServlet" method="get">

        <input type="hidden" name="action" value="pageByPrice">

        价格: <input id="min" type="text" name="min" value="${param.min}"> 元

        -

        <input id="max" type="text" name="max" value="${param.max}"> 元

        <input type="submit" value="查询" />

    </form>

</div>
```

3.8 翻页的实现界面

翻页的实现，如图 3-8 所示：



图 3-8 翻页的实现界面

主要代码

```
<div id="page_nav">

    <!-- 大于首页，才显示 -->

    <c:if test="${requestScope.page.pageNo}>1}">

        <a href="${requestScope.page.url}&pageNo=1">首页</a>
```

```

        <a href="{requestScope.page.url}&pageNo={requestScope.page.pageNo-1}">上一页
    </a>

</c:if>

<%-- 页码输出的开始 --%>

<c:choose>

    <%-- 情况 1 : 如果总页码小于等于 5 的情况, 页码的范围是: 1- 总页码 --%>

    <c:when test="{ requestScope.page.pageTotal <= 5 }">

        <c:set var="begin" value="1"/>  <%--为了防止伟哥判断语句都去遍历, 这里只需要设置开始和结束标记就可以了, 让最后的条件外的 foreach 输出即可--%>

        <c:set var="end" value="{requestScope.page.pageTotal}"/>

    </c:when>

    <%-- 情况 2 : 总页码大于 5 的情况 --%>

    <c:when test="{requestScope.page.pageTotal > 5 }">

        <c:choose>

            <%-- 小情况 1 : 当前页码为前面 3 个: 1 , 2 , 3 的情况, 页码范围是: 1-5.--%>

            <c:when test="{requestScope.page.pageNo <= 3 }">

                <c:set var="begin" value="1"/>

                <c:set var="end" value="5"/>

            </c:when>

            <%-- 小情况 2 : 当前页码为最后 3 个, 8 , 9 , 10 , 页码范围是: 总页码减 4 - 总页码 --%>

            <c:when test="{requestScope.page.pageNo > requestScope.page.pageTotal-3 }">

                <c:set var="begin" value="{requestScope.page.pageTotal-4}"/>

                <c:set var="end" value="{requestScope.page.pageTotal}"/>

            </c:when>

            <%-- 小情况 3 : 4 , 5 , 6 , 7 , 页码范围是: 当前页码减 2 - 当前页码加 2--%>

            <c:otherwise>

```

```

        <c:set var="begin" value="{requestScope.page.pageNo-2}"/>
        <c:set var="end" value="{requestScope.page.pageNo+2}"/>
    </c:otherwise>
</c:choose>

</c:when>
</c:choose>

<c:forEach begin="{begin}" end="{end}" var="i">
    <c:if test="{i == requestScope.page.pageNo}"> <!--当前页不可跳转，其它页可跳
转--%>

        【{i}】

    </c:if>
    <c:if test="{i != requestScope.page.pageNo}">
        <a href="{requestScope.page.url}&pageNo={i}">{i}</a>
    </c:if>
</c:forEach>

<!-- 页码输出的结束 --%>
<!--如果已经是最后一页，则不显示下一页，末页--%>
<c:if test="{requestScope.page.pageNo < requestScope.page.pageTotal}">
    <a href="{requestScope.page.url}&pageNo={requestScope.page.pageNo+1}">下一页
</a>

    <a href="{requestScope.page.url}&pageNo={requestScope.page.pageTotal}">末页
</a>
</c:if>

共{requestScope.page.pageTotal}页， {requestScope.page.pageTotalCount}条记录
到第<input value="{requestScope.page.pageNo}" name="pn" id="pn_input"/>页，
<input id="searchPageBtn" type="button" value="确定">

<!--想要在输入框中输入数字并条撞到相应的页数，就得给输入框绑定事件--%>
<script type="text/javascript" >

```

```

$(function () {

    //跳转到指定的页码

    $("#searchPageBtn").click(function () {

        var pageNo = $("#pn_input").val(); /*要跳转的页码*/

        var pageTotal = ${requestScope.page.pageTotal}; /*总页码*/

        if(pageNo < 1 || pageNo > pageTotal){

            alert("对不起，你想跳转的页面超出了本该有的限制,已经给您跳到范围
的边界了");

        }else{

            // javaScript 语言中提供了一个 location 地址栏对象

            // 它有一个属性叫 href. 它可以获取浏览器地址栏中的地址

            // href 属性可读，可写

            location.href =

            "${pageScope.basePath}${requestScope.page.url}&pageNo=" + pageNo;

        }

    });

});

</script>

</div>

```

3.9 购买图书添加到购物车界面

购买图书添加到购物车，如图 3-9 所示：



图 3-9 购买图书添加到购物车界面

主要代码

```
<c:if test="${not empty sessionScope.cart.items}">

    <%--购物车非空输出--%>

    <span>您的购物车中有${sessionScope.cart.totalCount}件商品</span>

    <div>

        您刚刚将<span style="color: red">${sessionScope.lastName}</span>

        加入到了购物车中

    </div>

</c:if>
```

3.10 查看购物车界面

查看购物车，可以看到购物车页面有以下功能：清空购物车、删除所购买商品、对购物车商品数据进行更改和查看我的订单，以及去结账。如图 3-10 所示：



图 3-10 查看购物车界面

主要代码

```
<body>

<div id="header">

    <span class="wel_word">购物车</span>

    <%--静态包含，登录 成功之后的菜单 --%>

    <% @ include file="/pages/common/login_success_menu.jsp"%>

    <script type="text/javascript">

        $(function () {
```



```
// 给 【删除】 绑定单击事件

$("a.deleteItem").click(function () {

    return confirm("你确定要删除 【" +
$(this).parent().parent().find("td:first").text() + "】 吗?")

});

// 给清空购物车绑定单击事件

$("#clearCart").click(function () {

    return confirm("你确定要清空购物车吗?");

});

// 给输入框绑定 onchange 内容发生改变事件(即，输入框发生改动，离开输入框就会发生提示)

$(".updateCount").change(function () {

    // 获取商品名称

    var name = $(this).parent().parent().find("td:first").text();

    var id = $(this).attr('bookId');

    // 获取商品数量

    var count = this.value;

    if ( confirm("你确定要将 【" + name + "】 商品修改数量为: " + count + "

    吗?")) {

        //发起请求。给服务器保存修改

        location.href =

"http://localhost:6211/Book_Web/cartServlet?action=updateCount&count="+count+"&id="+id;

    } else {

        // defaultValue 属性是表单项 Dom 对象的属性。它表示默认的 value

        修改之前的属性值。

        this.value = this.defaultValue;

    }

});

});

</script>
```

```
</div>
```

```
<div id="main">
```

```
<table bgcolor="#fff8dc" border="2">
```

```
<tr>
```

```
<td>商品名称</td>
```

```
<td>数量</td>
```

```
<td>单价</td>
```

```
<td>金额</td>
```

```
<td>操作</td>
```

```
</tr>
```

```
<c:if test="${empty sessionScope.cart.items}">
```

```
<%--如果购物车空的情况--%>
```

```
<tr> <%--用 colspan 去跨 5 个单元格输出该文字信息--%>
```

```
<td colspan="5"><a href="index.jsp">亲，当前购物车为空！快跟小伙伴们去浏览商品吧!!! </a> </td>
```

```
</tr>
```

```
</c:if>
```

```
<c:if test="${not empty sessionScope.cart.items}">
```

```
<%--如果购物车非空的情况--%>
```

```
<c:forEach items="${sessionScope.cart.items}" var="entry">
```

```
<tr>
```

```
<td>${entry.value.name}</td>
```

```
<td>
```

```
<input class="updateCount" style="width: 80px;"
```

```
bookId="${entry.value.id}"
```

```
type="text" value="${entry.value.count}">
```

```
</td>
```

```

        <td>${entry.value.price}</td>

        <td>${entry.value.totalPrice}</td>

        <td><a class="deleteItem"
href="cartServlet?action=deleteItem&id=${entry.value.id}">删除</a></td>

    </tr>

</c:forEach>

</c:if>

</table>

<%--如果购物车非空才输出页面的内容--%>

<c:if test="${not empty sessionScope.cart.items}">

    <div class="cart_info">

        <span class="cart_span">购物车中共有<span
class="b_count">${sessionScope.cart.totalCount}</span>件商品</span>

        <span class="cart_span">总金额<span
class="b_price">${sessionScope.cart.totalPrice}</span>元</span>

        <span class="cart_span"><a id="clearCart" href="cartServlet?action=clear">清空
购物车</a></span>

        <span class="cart_span"><a href="orderServlet?action=createOrder">去结账
</a></span>

    </div>

</c:if>

</div>

<%--静态包含页脚内容--%>

<% @include file="/pages/common/footer.jsp"%>

</body>

```

3.11 结算成功生成订单界面

结算成功生成订单界面，如图 3-11 所示：



图 3-11 结算成功生成订单界面

主要代码

```
<body>
```

```
<div id="header">
```

```
    
```

```
    <span class="wel_word">结算</span>
```

```
    <% @ include file="/pages/common/login_success_menu.jsp"%>
```

```
</div>
```

```
<div id="main">
```

```
    <h1>你的订单已结算, 订单号为${sessionScope.orderId}</h1>
```

```
</div>
```

```
    <% @ include file="/pages/common/footer.jsp"%>
```

```
</body>
```

3.12 以用户身份查看订单状态界面

以用户身份查看订单状态，如图 3-12 所示：



图 3-12 以用户身份查看订单状态界面

主要代码

```
<body>
```

```
<div id="header">
```

```
    
```

```
    <span class="wel_word">我的订单</span>
```

```
    <% @ include file="/pages/common/login_success_menu.jsp"% >
```

```
</div>
```

```
<div id="main" >
```

```
    <table bgcolor="#fff8dc" border="2">
```

```
        <tr>
```

```
            <td>订单日期</td>
```

```
            <td>金额</td>
```

```
            <td>状态</td>
```

```
            <td>订单编号</td>
```

```
            <td>查看详情</td>
```

```
        </tr>
```

```
<c:forEach items="${requestScope.myOrders}" var="order"> <%--这里的 books 一定要与提交的 list 中的 key 保持一致--%>
```

```
<c:if test="${empty order.orderId}">
```

```
<tr> <%--用 colspan 去跨 5 个单元格输出该文字信息--%>
```

```
<td colspan="4"><a href="pages/cart/cart.jsp">亲，您当前还没有过订单呢？现在快去购物车里结账吧!!! </a> </td>
```

```
</tr>
```

```
</c:if>
```

```
<c:if test="${not empty order.orderId}">
```

```
<tr>
```

```
<td>${order.createTime}</td>
```

```
<td>${order.price}</td>
```

```
<c:if test="${order.status == 0}">
```

```
<td>待发货</td>
```

```
</c:if>
```

```
<c:if test="${order.status == 1}">
```

```
<td>已发货，注意查收</td>
```

```
</c:if>
```

```
<td>${order.orderId}</td>
```

```
<td><a
```

```
href="orderServlet?action=showOrderDetail&id=${order.orderId}">查看详情</a></td>
```

```
</tr>
```

```
</c:if>
```

```
</c:forEach>
```

```
</table>
```

```
</div>
```

```
<% @ include file="/pages/common/footer.jsp"%>
```

```
</body>
```

3.13 订单详情界面

在订单详情页面下，通过点击页面可以知道自己订单的各项参考指标，订单详情页面如图 3-13 所示：



图 3-13 订单详情界面

主要代码

```
<body>
```

```
  <div id="header">
```

```
    
```

```
    <span class="wel_word">订单详情</span>
```

```
    <% @ include file="/pages/common/login_success_menu.jsp"% >
```

```
  </div>
```

```
  <div id="main" >
```

```
    <table bgcolor="#fff8dc" border="2">
```

```
      <tr>
```

```
        <td>名字</td>
```

```
        <td>价格</td>
```

```
        <td>卖出总数</td>
```

```
        <td>订单编号</td>
```

```
      </tr>
```

```
      <c:forEach items="${requestScope.orderDetails}" var="order">  <!--这里的
      books 一定要与提交的 list 中的 key 保持一致--%>
```

```
    </tr>
```

```

        <td>${order.name}</td>

        <td>${order.price}</td>

        <td>${order.count}</td>

        <td>${order.orderId}</td>

    </tr>

</c:forEach>

</table>

</div>

<% @ include file="/pages/common/footer.jsp"% >

</body>

```

3. 14 系统开发人员信息界面

系统开发人员信息查看，如图 3-14 所示：



图 3-14 系统开发人员信息界面

主要代码

```

<body>

    <div id="header">

    </div>

    <div id="main">

```



```
<table bgcolor="#fff8dc" border="2">
```

```
<tr>
```

```
<td>职务</td>
```

```
<td>姓名</td>
```

```
<td>学号</td>
```

```
<td>年龄</td>
```

```
<td>性别</td>
```

```
</tr>
```

```
<tr>
```

```
<td>组长</td>
```

```
<td>孔德华</td>
```

```
<td>20191106211</td>
```

```
<td>22</td>
```

```
<td>男</td>
```

```
</tr>
```

```
<tr>
```

```
<td>组员</td>
```

```
<td>经卫杰</td>
```

```
<td>20191106210</td>
```

```
<td>22</td>
```

```
<td>男</td>
```

```
</tr>
```

```
<tr>
```

```
<td>组员</td>
```

```
<td>徐亮</td>
```

```
<td>20191106230</td>
```

```
<td>20</td>
```

```
<td>男</td>
```

```
</tr>
```

```
<tr>

    <td>组员</td>

    <td>靳宇鑫</td>

    <td>20191106209</td>

    <td>21</td>

    <td>男</td>

</tr>

<tr>

    <td>组员</td>

    <td>杨龙</td>

    <td>20181106231</td>

    <td>22</td>

    <td>男</td>

</tr>

<tr>

    <td></td>

    <td></td>

    <td></td>

    <td></td>

    <td><a href="index.jsp">请点击我返回^_^</a></td>

</tr>

</table>

</div>

<% @ include file="/pages/common/footer.jsp"%>

</body>
```

3.15 用户身份查看后台界面

以用户的身份去查看后台管理页面，如图 3-15 所示：



图 3-15 用户身份查看后台界面

主要代码

```
<body>

<div id="header">

</div>

<div id="main">

    <h1>这里只有管理员才能查看的呦!!! </h1>

    <table>

        <tr>

            <td></td>

            <td></td>

            <td></td>

            <td></td>

            <td></td>

            <td><a href="index.jsp">请点击我返回^_^</a></td>

        </tr>

    </table>

</div>

<% @ include file="/pages/common/footer.jsp"%>

</body>
```

3. 16 管理员身份查看后台界面

以管理员的身份去查看后台管理页面，如图 3-16 所示：



图 3-16 管理员身份查看后台界面

主要代码

```
<body>
```

```
<div id="header">
```

```
    
```

```
    <span class="wel_word">后台管理系统</span>
```

```
    <% @ include file="/pages/common/manager_menu.jsp"%>
```

```
</div>
```

```
<div id="main">
```

```
    <h1>欢迎管理员进入后台管理系统</h1>
```

```
</div>
```

```
<% @ include file="/pages/common/footer.jsp"%>
```

```
</body>
```

3.17 图书管理界面

图书管理页面主要实现了图书的添加、删除、修改、页翻页的实现，图书管理页面如图 3-17 所示：



图 3-17 图书管理界面

主要代码

<body>

<div id="header">

图书管理系统

<% @ include file="/pages/common/manager_menu.jsp"%>

</div>

<div id="main">

<table bgcolor="#fff8dc" border="2">

<tr>

<td>名称</td>

<td>价格</td>

<td>作者</td>

<td>销量</td>

```

        <td>库存</td>

        <td colspan="2">操作</td>

    </tr>

    <c:forEach items="${requestScope.page.items}" var="book">    <%--这里的 books 一
    定要与提交的 list 中的 key 保持一致--%>

        <tr>

            <td>${book.name}</td>

            <td>${book.price}</td>

            <td>${book.author}</td>

            <td>${book.sales}</td>

            <td>${book.stock}</td>

            <td><a
href="manager/bookServlet?action=getBook&id=${book.id}&pageNo=${requestScope.page.page
No}">修改</a></td>

            <td><a class="deleteClass"
href="manager/bookServlet?action=delete&id=${book.id}&pageNo=${requestScope.page.pageNo
}">删除</a></td>

        </tr>

    </c:forEach>

<tr>

    <td></td>

    <td></td>

    <td></td>

    <td></td>

    <td></td>

    <td></td>

    <td><a
href="pages/manager/book_edit.jsp?pageNo=${requestScope.page.pageTotal}">添加图书
</a></td>    <%--在末尾显示--%>

</tr>

</table>

```

```

        <!--静态包含分页条--%>

        <% @include file="/pages/common/page_nav.jsp"%>

    </div>

    <!--静态脚本内容--%>

    <% @ include file="/pages/common/footer.jsp"%>

</body>

```

3. 18 图书的添加界面

图书的添加界面，如图 3-18 所示：



图 3-18 图书添加界面

主要代码

```

<div id="main">

    <form action="manager/bookServlet" method="post"> <!--manager/bookServlet --->
这是后台--%>

        <input type="hidden" name="pageNo" value="${param.pageNo}">

        <input type="hidden" name="action" value="${ empty param.id ? "add" :
"update" }" /><!--此处设立隐藏域是要告诉服务器中的的 post 请求，要调用 add/update 方法，
并通过 add 方法回到 book_manager 中--%>

        <input type="hidden" name="id" value="${ requestScope.book.id }" /> <!--必须
得有关于 id 隐藏域,这样删除的时候能找到 id,从而达到删除的目的--%>

        <table bgcolor="#fff8dc" border="2">

            <tr>

                <td>名称</td>

                <td>价格</td>

```

```

        <td>作者</td>

        <td>销量</td>

        <td>库存</td>

        <td colspan="2">操作</td>

    </tr>

    <tr>

        <td><input name="name" type="text"
value="\${requestScope.book.name}"/></td>

        <td><input name="price" type="text"
value="\${requestScope.book.price}"/></td> <!--这里的 name 要和 javaBean 中的 set 方法后名
称一致--%>

        <td><input name="author" type="text"
value="\${requestScope.book.author}"/></td>

        <td><input name="sales" type="text"
value="\${requestScope.book.sales}"/></td>

        <td><input name="stock" type="text"
value="\${requestScope.book.stock}"/></td>

        <td><input type="submit" value="提交"/></td>

    </tr>

</table>

</form>

</div>

```

3. 19 图书的添加成功界面

图书的添加成功界面，如图 3-19 所示：

华仔在手，天下我有	123.00	孔德华	8923	10013	修改	删除
						添加图书

首页 上一页 4 5 6 7 [8] 共8页, 30条记录 到第 8 页 确定

图 3-19 图书添加成功界面

3.20 图书的删除界面

图书信息的修改界面，如图 3-20 所示：

名称	价格	作者	销量	库存	操作
教父	29.00	马里奥普佐	400	523449	保存

图 3-20 图书信息修改界面

主要代码

```
protected void delete(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
```

```
    //1、获取请求的参数 id，图书编程
```

```
    int id = WebUtils.parseInt(req.getParameter("id"), 0);
```

```
    //2、调用 bookService.deleteBookById();删除图书
```

```
    bookService.deleteBookById(id);
```

```
    //3、重定向回图书列表管理页面
```

```
    // /book/manager/bookServlet?action=list
```

```
    resp.sendRedirect(req.getContextPath() +
"/manager/bookServlet?action=page&pageNo="+req.getParameter("pageNo"));
}
```

3.21 图书的删除提醒界面

图书信息的修改成功界面，如图 3-21 所示：

名称	价格	作者	销量	库存	操作	
教父	29.00	马里奥普佐	400	523449	修改	删除

图 3-21 图书信息修改成功界面

3.22 图书的删除以及删除提醒界面

图书的删除以及删除提醒界面，如图 3-22 所示：



图 3-22 图书删除以及删除提醒界面

主要代码

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

$(function () {

    // 给删除的 a 标签绑定单击事件，用于删除的确认提示操作

    $("a.deleteClass").click(function () {

        // 在事件的 function 函数中，有一个 this 对象。这个 this 对象，是当前正在
        响应事件的 dom 对象。

        /**

        * confirm 是确认提示框函数

        * 参数是它的提示内容

        * 它有两个按钮，一个确认，一个是取消。

        * 返回 true 表示点击了，确认，返回 false 表示点击取消。

        */

        return confirm("你确定要删除【" + $(this).parent().parent().find("td:first").text()
+ "】?");

        //以上提一个 this 的 parent()找到了 tr 标签，第二个找到了 td 标签，进而去匹
```

配 id

```
// return false// 阻止元素的默认行为===不提交请求

});

});

</script>
```

3.23 以管理员身份查看在后台的订单管理界面

以管理员身份查看在后台的订单管理界面，主要有以下功能：查看订单的信息，是否发货，以及订单详情的查看，以管理员身份查看在后台的订单管理界面，如图 3-23 所示：



图 3-23 以管理员身份查看在后台的订单管理界面

主要代码

```
<body>

<div id="header">

    <span class="wel_word">订单管理系统</span>

    <% @ include file="/pages/common/manager_menu.jsp"%>

</div>
```

```

<div id="main">

<table bgcolor="#fff8dc" border="2">

    <tr>

        <td>日期</td>

        <td>金额</td>

        <td>编号</td>

        <td>发货</td>

        <td>查看详情</td>

    </tr>

    <c:forEach items="${requestScope.orders}" var="order"> <!--这里的 books 一定要与提交的 list 中的 key 保持一致-->

        <tr>

            <td>${order.createTime}</td>

            <td>${order.price}</td>

            <td>${order.orderId}</td>

            <c:if test="${order.status == 0}">

                <td><a

href="manager/orderServlet?action=sendOrder&id=${order.orderId}">点击发货</a></td>

            </c:if>

            <c:if test="${order.status == 1}">

                <td>已经发货</td>

            </c:if>

            <td><a href="orderServlet?action=showOrderDetail&id=${order.orderId}">

查看详情</a></td>

        </tr>

    </c:forEach>

</table>

</div>

<% @ include file="/pages/common/footer.jsp"%>

</body>

```

第 4 章 系统测试

系统测试的目的就是测试本系统的各个功能是否真正的能实现，是否达到预期的目标。同时，系统测试还具有记录系统在测试过程中各个阶段、各个步骤是否按照系统设计的流程进行的作用。

系统测试可以测试出完成的系统存在的问题，并对其进行及时的修改，因此，系统测试还起到预防系统再次出错的作用。

本系统的测试方法采用动态测试中的黑盒测试法。黑盒测试就是测试人员把系统看成是一个黑盒子，并不想要关心盒子内部的结构和特性，而只关注系统的输入数据和输出结果，从而检查软件产品是否符合它的功能。黑盒测试比较简单方便，很适合对本系统进行测试。

4.1 测试用例

现列出功能测试结果，见表 4-1：

表 4-1 功能测试用例

序号	测试目标	输入/操作说明	预期结果	实际结果	测试结果
1	登录	直接点击“登录”按钮	页面不进行任何跳转	页面不进行任何跳转	通过
2	登录	只输入用户名 点击“登录”按钮	页面不进行任何跳转，表单回显并提示错误信息	页面不进行任何跳转表单回显并提示错误信息	通过
3	登录	输入号码 “admin”密码 “admin”点击 “登录”按钮	进入系统欢迎页面	进入系统欢迎页面	通过
4	登录	提示没该点击 “立即注册”	进入注册页面	进入注册页面	通过
5	注册	直接点击“注册”按钮	提示“用户名不合法”对话框	提示“用户名不合法”对话框	通过
6	注册	直接输入 “cainiao” 其他输入“1” 点击“注册”按钮	提示“密码不合法”对话框	提示“密码不合法”对话框	通过

序号	测试目标	输入/操作说明	预期结果	实际结果	测试结果
7	注册	输入“cainiao” 密码两次不一致输入“	提示“确认密码和输入密码不一致“	提示“确认密码和输入密码不一致“	通过
8	注册	输入“cainiao” 密码两次都一致，邮箱输入“12.qqc.om”	提示“邮箱格式不合法”	提示“邮箱格式不合法”	通过
9	注册	用户名合法， 密码一致，邮箱合法，验证码不填或填写错误	提示“验证码不能为空”“验证码错误”	提示“验证码不能为空”“验证码错误”	通过
10	注册	任何输入都没有问题	进入系统欢迎页面	进入系统欢迎页面	通过
11	区间查找	在价格区间输入 1-15	只显示三本书，分页条只有一页，并显示三条记录	只显示三本书，分页条只有一页，并显示三条记录	通过
12	分页	点击分页条和定向查找	能够跳转到正确的显示页面	能够跳转到正确的显示页面	通过
13	分页	点击超过记录的页数的定向查找	提示没有此页面	提示没有此页面	通过
14	系统开发人员	点击系统开发人员	能够跳转到系统开发人员页面	能够跳转到系统开发人员页面	通过
15	添加到购物车	点击添加购物车	能够提示“您刚刚将边城加入到了购物车中”	能够提示“您刚刚将边城加入到了购物车中”	通过
16	购物车	点击购物车	能够跳转到显示添加的商品的购物车页面	能够跳转到显示添加的商品的购物车页面	通过
17	购物车	点击删除	能够把图书给删除	能够把图书给删除	通过
18	购物车	点击清空购物车	页面没有任何商品	页面没有任何商品	通过

序号	测试目标	输入/操作说明	预期结果	实际结果	测试结果
19	购物车	点击去结账	能够显示出订单编号	能够显示出订单编号	通过
20	去结账	点击“去结账”	跳转到二维码支付页面，能支付之后才生成订单号	跳转到二维码支付页面，能支付之后才生成订单号	没通过
21	注销	点击注销	能够跳回初始主页面	能够跳回初始主页面	通过
22	后台管理	以非管理员身份点击后台管理	跳转到温馨提示页面	跳转到温馨提示页面	通过
23	后台管理	以管理员身份点击后台管理	跳转到后台管理页面	跳转到后台管理页面	通过
24	后台管理	点击图书管理	显示所有图书的信息	显示所有图书的信息	通过
25	后台管理	点击修改，并修改部分信息	实现修改并跳转显示	实现修改并跳转显示	通过
26	后台管理	点击删除，并删除部分信息	实现删除并跳转显示	实现删除并跳转显示	通过
27	后台管理	点击添加图书	实现添加并跳转显示	实现添加并跳转显示	通过
28	订单管理	点击“点击发货”	由“点击发货”变为“已发货”	由“点击发货”变为“已发货”	通过
29	订单管理	点击“查看详情”	跳转到具体图书信息页面	跳转到具体点单图书信息页面	通过
30	我的订单	点击“我的订单”	跳转到“我的订单”页面，并显示订购单信息	跳转到“我的订单”页面，并显示订购单信息	通过
31	我的订单	点击“查看详情”	跳转到具体图书信息页面	跳转到具体图书信息页面	通过

4.2 测试结论

1、通过对写好的系统进行了多轮的测试，虽遗留了一些缺陷没有解决，但系统功能已趋于稳定，测试可以结束、该系统以及可以拿出来当课程设计了。

2、通过测试觉得系统无法真实的实现支付扣费的响应来生成订单号，这是我们无法用现有的知识来实现的，日后等我么涉及该领域时，再去完善这一 Bug。

第5章 总结

本系统名称为“图书管理销售系统”。本系统的中心内容是作为用户能够在主页面挑选图书，并把图书加入到购物车中，对商品实现购买以及在我的订单处查看订单的信息；作为系统管理员能够在后台管理处实现对书籍的增删改查、和对用户所提交的订单实现发货和查看订单详情。

本系统属于销售管理系统，让图书实现销售和管理。系统数据库采用 MySQL，以 JSP 为系统主要的设计语言，利用 IDEA2020.1 完成了动态网页的制作和网页与数据库的连接，静态页面的美化主要使用 Photoshop2020 软件进行。各种点击所触发的提示都是 JavaScript 技术实现的，以及动态验证码是通过谷歌的验证码架包所提供的技术实现的。

系统在经过测试之后已经圆满完成，在测试过程中发现的问题也已经全部改正并完善，其中页面中文乱码问题和登陆成功之后的显示问题困扰比较大，但是在咨询老师和同学之后已解决。

虽然系统的各个模块的功能已经实现，但是系统的功能依然比较少，还缺少许多有用的功能，比如点击去结账后，直接生成了订单编号，没有支付系统，支付系统对于现在的我们是个难点；在比如后台的权限管理用的是 Filter 过滤器，但是这个给管理员使用的时候造成了不便；还有 ps 技术不够成熟，这使静态页面不够美观等。

本系统一定还会继续发现其他问题和缺陷，希望可以在以后的学习和工作中不断进行修改和完善，继续扩充本系统的功能，使之更加安全方便、同时具备更强的实用性。