

# Study and Design of Online Bookstore Backend

Yunzheng Ding\*

School of Information Engineering, Jingdezhen Ceramic university, Jingdezhen 333403, China

\*jciddy@163.com

## Abstract

**This article focuses on the actual needs of bookstores and designs and develops a small e-commerce website based on JSP, namely the online bookstore system. The technical solution adopted by the bookstore system is: JSP+Tomcat+SQLServer2000. The main functions of an online bookstore system based on JSP are two-fold: front-end user and back-end management. The main functions of the backend include: adding and deleting book information, querying book information, setting user levels, publishing notifications, managing book categories and publishers, querying and summarizing book sales information, and so on. The system is easy to operate, has stable performance, is easy to maintain and upgrade, and should be supported by a large number of sellers.**

## Keywords

**Online Bookstore; JSP; JavaBean; B/S Structure.**

## 1. Introduction

In today's increasingly developed internet, online shopping has become a fashion trend, and more and more netizens have adopted this new way of shopping. As a form of e-commerce, online bookstore is the product of the Internet's gradual maturity.

Online bookstore is a specific form of enterprise to consumer e-commerce. It is a business model in which enterprises open online bookstores on the Internet, consumers browse book information through the Internet, place orders online, and pay online. This business model is convenient and fast, not limited by time and space, and operates 24 hours a day, making it popular among consumers. Nowadays, large bookstores across the country are gradually using the internet as a service and transaction method, transforming traditional management methods into online management methods. Online bookstores across the country are also on the rise, demonstrating that online bookstores are the future development trend of book management. Nowadays, the country is actively building network infrastructure, and the impact of the internet on people's lives is increasing. People's recognition of online shopping is also growing. These situations indicate that there is a very large market space for conducting online bookstore business at present.

## 2. Overall Design

### 2.1. Functions of Online Bookstore System

With the popularity of e-commerce, the trend of online shopping is constantly emerging. Online shopping has transcended the limitations of time and space, bringing extraordinary changes to the commercial circulation field. The true beneficiaries of online shopping are consumers. Similar to other e-commerce websites, the construction of online bookstores is generally divided into two parts: backend management and frontend users.

The front-end system, also known as the customer system, is a web interface designed for users to directly access and engage in purchasing activities. Users can register, log in, search for books,

browse, order books, and view order information online. The front-end system should also include some other auxiliary functions, such as bookstore introduction, latest news, website announcements, etc., but these functions are not unique to online bookstores. The backend management system, often referred to as "backend", is a system used by website owners to manage and control various functions of e-commerce websites through computers. Due to the fact that web servers and database servers are often not "local", backend management systems typically perform remote control management. The main functions of this backend are: adding, deleting, and modifying book information, querying book information, setting user levels, publishing notifications, managing book categories and publisher information, querying and summarizing book sales information, and so on.

The process of an online bookstore runs through four stages: front-end process, membership operation process, back-end management process, and complete book purchase process. Here, I will only introduce the back-end management process:

## 2.2. Database Design

Database is the core and foundation of information systems, and an important component of information system development and construction. It organizes a large amount of data in the information system according to a certain model, provides storage, maintenance, and retrieval functions, and enables the information system to easily, timely, and accurately obtain the required information from the database. The importance of database design is self-evident in the design of dynamic websites. Reasonable design of database structure can improve the efficiency of data storage and ensure the integrity and uniformity of data. If not designed properly, it will affect the query efficiency of the server and the performance of the program. Database design generally includes the following steps:

- (1) Database requirement analysis;
- (2) Database conceptual structure design;
- (3) Logical structure design.

## 2.3. Database Requirements Analysis

According to the requirements of the online bookstore system and the division of functional modules, as well as the general needs of online bookstores, the functions of the database are mainly reflected in the provision, storage, updating, and querying of various information, including user information, book information, and order information. Moreover, the data content of each part is internally related. Therefore, the following information table is summarized. (1) List of Book Information. (2) Users are divided into general users and administrators. (3) Orders are divided into individual detailed orders and order summary tables. (4) A user can purchase multiple books, corresponding to one or more orders. (5) Order Summary Table. This system has designed a database that includes 9 data tables, namely books, categories, distinctions, feedback, sales logs, suppliers, visits, customers, and votes. After the above functional analysis and requirement summary, the following data items and data structures are designed.

- (1) Book list, including book number, book classification, book name, author, supplier number, publication date, specific introduction, cost price, selling price, number of votes, etc.
- (2) Book classification table, including book classification number and book classification name.
- (3) User category table, including category number, price restrictions, and price discounts.
- (4) User table, including username, user account, user password, support category, phone number, province, city, address, email, etc.
- (5) Purchase log table, including purchase number, book number, user number, purchase time, quantity, and price.

- (6) Supplier list, including supplier number, publisher name, contact person name, contact phone number, contact address, etc.
- (7) Access table, including access sequence, visitor IP address, access frequency, etc.
- (8) Voting form, including voting number, name of voting book, number of votes, etc.
- (9) Feedback form, including book number, number of votes, etc.

### 2.4. Conceptual Design of Database

Based on the above data items and data structures, various entities that meet the requirements can be designed. The entities planned by this system include: book entities, book classification entities, administrator entities, user entities, order entities, and order detail entities. The entities and their relationships can be represented using entity relationship diagrams, also known as E-R (Entity Relationship) diagrams.

### 2.5. Database Logic Design

Transform the conceptual structure of the database into the logical structure of the database, storing tables of relevant information for each entity. The relationship diagram between these tables is as follows:

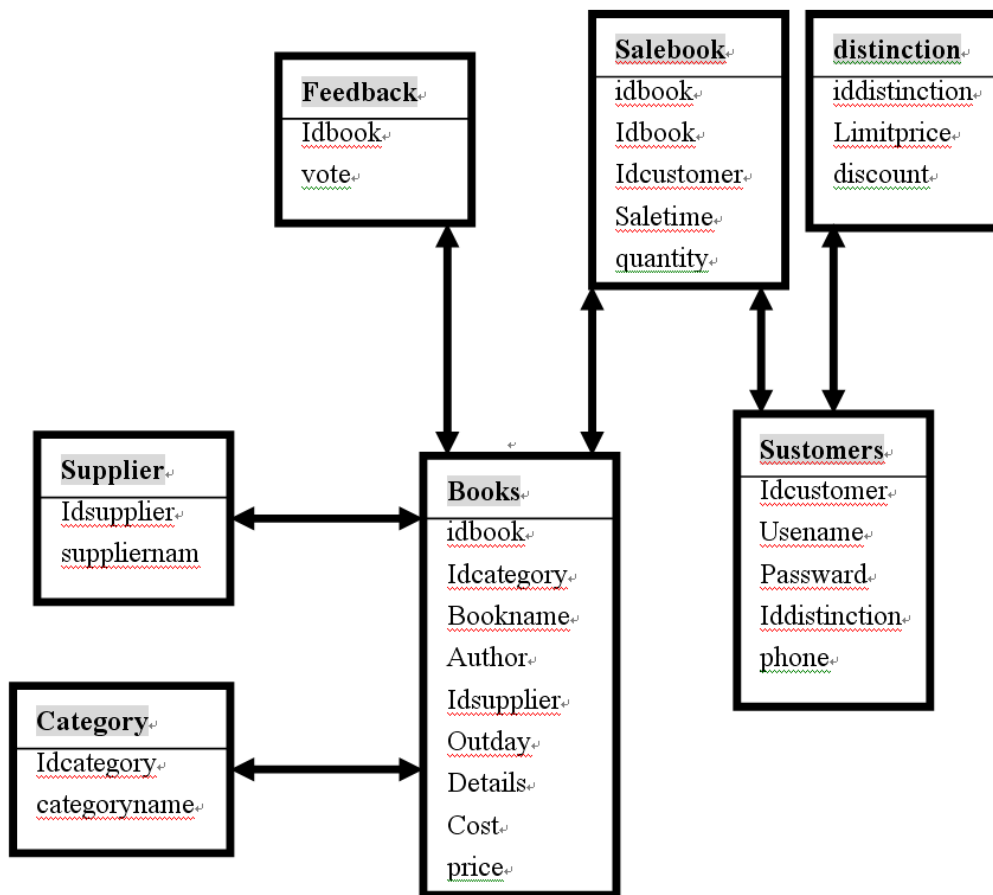


Figure 1. Relationship diagram

### 2.6. Database Configuration and Connection

In the development of online bookstores, the SQL Server 2000 database system is used. The database name of this system is BookShop. A structure of 9 data tables was established in the database as described above. There are three database connection techniques for JSP database technology. The first method is to use JDBC-ODBC bridge to connect to SQL Server 2000; The second method is to use a connection pool to connect to SQL Server 2000; The third method is

that JSP accesses the database through JavaBean. The JSP database technology used in this online bookstore is the second type, which is connection pooling technology.

The following are some of the main source code snippets for Tomcat's server.xml:

```

<Context path="/BookShop" reloadable="true" docBase="BookShop" >
  <Resource name="jdbc/BookShop" auth="Container"
    type="javax.sql.DataSource"/>
    <ResourceParams name="jdbc/BookShop">
      <parameter>
        <name>factory</name>
        <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
      </parameter>
      <parameter>
        <name>maxActive</name>
        <value>100</value>
      </parameter>
      <parameter>
        <name>maxIdle</name>
        <value>30</value>
      </parameter>
      <parameter>
        <name>maxWait</name>
        <value>10000</value>
      </parameter>
      <!-- MySQL dB username and password for dB connections
-->
      <parameter>
        <name>username</name>
        <value>sa</value>
      </parameter>
      <parameter>
        <name>password</name>
        <value></value>
      </parameter>
      <parameter>
        <name>driverClassName</name>
        <value>com.microsoft.jdbc.sqlserver.SQLServerDriver</value>
      </parameter>
      <name>url</name>
      <value>jdbc:      microsoft:
sqlserver://127.0.0.1:1433 ;DatabaseName=BookShop; user=sa; password=;
//JDBC and database connection
</value>
</parameter>
</ResourceParams>
</Context>

```

The above program mainly plays the role of JDBC connection to the database, where the port and database name, as well as the database name, username, and password are all mentioned.

The following explains the configuration and connection of SQL Server 2000 database. Firstly, install SQLServer2000,. Then, install JDBC and its drivers. When using JSP to connect to a SQL Server 2000 database, the JSP SQLServer2000 driver must first connect to the SQL Server using msbase.jar, Add three files, mssqlserver.jar and msutil.jar, to% Tomcat 5.0% \ common \ lib. The above three files are the drivers for JSP connection to SQL Server 2000. Only by adding these three files to% Tomcat 5.0% \ common \ lib can the connection pool connect to the database successfully.

### **3. Detailed Design and Implementation**

#### **3.1. System Design Features**

##### **3.1.1. Page Modularization**

The construction of online bookstores is generally divided into two parts: backend management and frontend users. The main framework of an online bookstore consists of three pages. On the left side of the framework is the category list page for books, which is used to meet the needs of users who view books by category. The head of the framework is quite complex and is used for functional navigation in bookstores.

##### **3.1.2. Three Layer Structure Design**

This system adopts a three-tier structure design, which divides the program logic structure into a user interface layer, a business logic layer, and a data storage layer. Due to the strong functionality of the online bookstore system, the corresponding code design is relatively complex, and the JSP webpage text will also become large, requiring a large amount of embedded Java code and HTML statements. Among them, Java code is logical code, which is used to complete the logical functions of the system. Embedding Java code, HTML statements, and JavaScript statements makes it difficult to write and maintain JSP web pages, but using JavaBeans can effectively solve this problem. Combining JSP and JavaBean, the business logic processing layer is implemented by JavaBean, and then called in the JSP web page. The JSP text focuses on the web page interface design, which will make the JSP web page clear, save software development time, and greatly improve the security, maintainability, reusability, and scalability of the system.

##### **3.1.3. Object Oriented Programming**

In this system, books, users, orders, etc. are all encapsulated into corresponding classes, and each class has its corresponding operation class, thereby further improving the security of database operations and the scalability of programs.

##### **3.1.4. Humanized Design**

This system has clear navigation, easy and simple operation, and users will not get lost no matter where they are in this bookstore.

Design of Class 3.2

In this online bookstore, books are the most basic information and have many applications. Create a separate category for them.

#### **3.2. Design of Main Backend Pages**

This system adopts a modular approach in interface design, with the main framework consisting of three pages. On the left side of the framework is the book category list page, which is used to meet the needs of users who view books by category. The head of the framework is quite complex and is used for functional navigation in bookstores. The main body of the framework is mainly used for users to view and operate book information. To enter the main framework of this bookstore, you need to jump from index.hk. Click on the four words 'Sunshine Bookstore' on the homepage to enter the bookstore.

### 3.3. Administrator Entrance

After the administrator successfully logs in through the member login page, they will not immediately enter the bookstore but will be redirected to a transit page. On this page, there are some functions for members, and the functions for members will be designed in the "front-end user operation" section. I am only responsible for the back-end part, so I will not introduce it here. There is also an option that only appears when the administrator logs in, which is an entrance prepared for administrators.

In the program, determine the identity of the logged in user. If it is an administrator, display the "Administrator Entry" option. Otherwise, the option will not be output, and the administrator's entrance will be dynamically displayed. The specific implementation is as follows:

```
<%
String id=(String)session.getAttribute("id");
String sqls="select iddistinction from customers where idcustomer='"+id+"' and iddistinction=6";
//Determine whether you are a system administrator
int temp=-2;
temp=login.select(sqls);
%>
<table width="413" height="412" border="0" align="right">
<tr>
<td width="530" height="80" align="center" valign="middle" class="two"><a ref="../customer/bookstore.html" class="two">Enter Sunshine Bookstore</a></td>
</tr>
<tr>
<td width="530" height="80" align="center" valign="middle" class="two"><a ref="../admin/books_sale.js" class="two">View book purchase status</a></td>
//Click on 'View Book Purchase Status' to jump to the books_sale.jsp page
</tr>
<tr>
<td width="530" height="80" align="center" valign="middle" class="two"><a ref="change.js" class="two">Modify member information</a></td>
//Click on 'Modify Member Information' to jump to the change.jsp page
</tr>
<tr>
<td height="80" align="center" valign="middle" class="two">
<%
if(temp==1){
%>
<a ref="..admin/admin.js" class="two">Administrator entry item</a>
<%
}
%></td>
</tr>
</tr>
```

```

<td height="80" align="center" valign="middle" class="two"><a ref="exit. js"
class="two">Exit</a></td>
</tr>
</table>
</body>
</html>

```

The administrator clicks on the "Administrator Entry" option to enter the administrator page, which is a left-right structured framework page. The left side is the function link page, and the right side is the administrator operation platform.

### 3.4. Book Inquiry, Membership Status Inquiry, and Sales Status Inquiry

In order to effectively manage book information, multiple query conditions can be established. In this system, most fields of book information are included in the query conditions to provide users with more accurate queries.

When all query conditions are empty, all book information in the system database will be returned. If the query condition is greater than or equal to one, only the book information that meets the query condition will be returned. If there are no books that meet the query condition, it will be returned as empty. The specific implementation is as follows:

```

<table width="825" border="1" align="center" cellpadding="-1" cellspacing="0" bordercolor=
"#ffffff" bordercolorlight="#0066ff" bordercolordark="#ffffff">
<tr>
<th width="71" height="22">Book ID</th>
<th width="175">Book name</th>
<th width="60">Author</th>
<th width="88">Publication time</th>
<th width="39">Purchase price</th>
<th width="40">Price</th>
<th width="66">Book cover</th>
<th width="34">Purchasing</th>
<th width="33">Sold</th>
<th width="37">Publish</th>
<th width="51">Hot Books</th>
<th width="66">Category</th>
<th width="116">Supplier</th>
</tr>
<%

```

```

//Read the information in the request and complete it through Javabeans
    idbook=(String)request.getParameter("idbook");
    bookname=(String)request.getParameter("bookname");
    category=(String)request.getParameter("category");
    supplier=(String)request.getParameter("supplier");
    author=(String)request.getParameter("author");
    active=(String)request.getParameter("active");
    hotdeal=(String)request.getParameter("hotdeal");
    idbook=books.inStr(idbook);
    bookname=books.inStr(bookname);
    category=books.inStr(category);

```

```

author=books.inStr(author);
supplier=books.inStr(supplier);
active=books.inStr(active);
hotdeal=books.inStr(hotdeal);
idbook=books.selectStr(idbook);
bookname=books.selectStr(bookname);
category=books.selectStr(category);
author=books.selectStr(author);
supplier=books.selectStr(supplier);
active=books.selectStr(active);
hotdeal=books.selectStr(hotdeal);

```

```
//Search for data
```

```

String sqls="SELECT
idbook,bookname,author,outday,details,cost,price,imageurl,stock,active,hotdeal,sale,cat
egoryname,suppliername FROM books as a,category as b,supplier as c WHERE idbook
"+idbook+" AND bookname "+bookname+" AND categoryname "+category+" AND
author "+author+" AND suppliername "+supplier+" AND active "+active+" AND hotdeal
"+hotdeal+" AND a.idcategory=b.idcategory AND a.idsupplier=c.idsupplier";
Statement stmt=books.getstmthead();
ResultSet rs=stmt.executeQuery(sqls);

```

After the user submits the data, the above program receives and processes the data in two layers, both completed by JavaBesn. The first layer is to transcode and filter the data; The second layer is to convert the data into SQL statements for fuzzy queries, and use the converted data when querying the data later. Connecting them together forms a complete multifunctional query statement. The query results display all the information of the books that meet the criteria.

The status query of members is similar to book query, consisting of multiple fields that can be queried accurately or vaguely. Of course, some information of members is also protected here, such as member passwords. Sales status queries are conducted by searching by time period.

### 3.5. Library Management

There are three main functions of book management: adding books, modifying books, and deleting book information, Users can view detailed information about each book through this link, mainly achieved as follows:

```

.... while(rs.next()){
//Read book information from the database
idbook=rs.getString("idbook");
bookname=rs.getString("bookname");
author=rs.getString("author");
outday=rs.getDate("outday");
details=rs.getString("details");
cost=rs.getString("cost");
price=rs.getString("price");
imageurl=rs.getString("imageurl");
stock=rs.getInt("stock");
active=rs.getInt("active");
hotdeal=rs.getInt("hotdeal");
sale=rs.getInt("sale");
categoryname=rs.getString("categoryname");
suppliername=rs.getString("suppliername");

```



```

stock=rs.getInt("stock");
active=rs.getInt("active");
hotdeal=rs.getInt("hotdeal");
sale=rs.getInt("sale");
categoryname=rs.getString("categoryname");
suppliername=rs.getString("suppliername");
//Shared book ID, which refers to the names of the books to be processed and the books in the
database
    idbook=books.outStr(idbook);
    bookname=books.outStr(bookname);
    author=books.outStr(author);
    details=books.outStr(details);
    imageurl=books.outStr(imageurl);
    categoryname=books.outStr(categoryname);
    suppliername=books.outStr(suppliername);
    %>

```

In the modification of book information page, the book category and supplier are obtained from the database and displayed in the form of a dropdown list for administrators to choose from, which makes the operation much more convenient.

Adding and deleting books is relatively simple, and their operation interfaces are all in the book information list.

### 3.6. Publisher Management and Book Category Management

Book category management is similar to publisher management, with detailed information about publishers added to different publisher management. Here, we will only introduce publisher management. Click on the "Publisher Management" link in the framework console to display a list of all publishers, which is directly retrieved from the database. Each publisher's name is a link to submit their publisher ID, which can be used to display detailed information about the publisher. The same method was also used in the publisher's deletion, and the implementation of this feature is as follows:

```

    while(rs.next()){
//Read publisher information from the publisher page
        int idsupplier=rs.getInt("idsupplier");
            String suppliername=rs.getString("suppliername");
            suppliername=supplier.outStr(suppliername);
        %>
        <tr>
            <td width="304" height="22" align="left" ><a
href="supplier_view.jsp?idsupplier=<%=idsupplier%>"><%=suppliername%></a>&n
bsp;</td>
            <td width="38" align="center" ><a
href="supplier_del.jsp?idsupplier=<%=idsupplier%>"></a></td>
        </tr>
        <%
    }
    supplier.close();
    pageContext.removeAttribute("supplier");

```

```
%>
```

When viewing the detailed information of publishers, if any errors are found, they can be corrected at any time.

When clicking on the "modify" link text, the publisher's ID will be submitted. The modification page will use the received publisher ID to retrieve data from the database and display it through a text box to provide modification functionality, which is very convenient. The specific implementation code is as follows:

```
//Read publisher information from the database
    String idsupplier=(String)request.getParameter("idsupplier");
    String sqls="SELECT * FROM supplier WHERE idsupplier="+idsupplier;
    Statement stmt=supplier.getstmthead();
    ResultSet rs=stmt.executeQuery(sqls);
    if(rs.next()){
//Shared publisher ID
    String suppliername=rs.getString("suppliername");
    String sname=rs.getString("sname");
    String phone=rs.getString("phone");
    String address=rs.getString("address");
    suppliername=supplier.outStr(suppliername);
    sname=supplier.outStr(sname);
    phone=supplier.outStr(phone);
    address=supplier.outStr(address);
    %>
```

### 3.7. Membership Level Setting

In the process of purchasing books, a consumption record will be made for each member, and the membership level will be automatically set based on the consumption amount. Newly registered users will be at level 0. There are a total of 0-5 levels, which are set in the database. The higher the membership level, the higher the discount rate. The discount rate and membership level consumption amount are freely set by the administrator. The implementation is as follows:

```
//Set membership levels and discount rates in the database
    String sqls1="SELECT limitprice,discount FROM distinction WHERE
    iddistinction="+select;
    Statement stmt1=dis.getstmthead();
    ResultSet rs1=stmt1.executeQuery(sqls1);
    while(rs1.next()){
        limitprice=rs1.getInt(1);
        discount=rs1.getInt(2);
    %>
```

Every time the membership level is changed, the corresponding discount rate and limit amount will also change, because every time the membership level is viewed from the drop-down list, a submission work is done, and the text box values for discount rate and limit amount will also change accordingly.

```
<form name="form1" method="post" action="">
<tr>
<td width="88" height="27" align="center">Membership level</td>
<td width="224" height="27" align="center">
<select name="select" onChange="javascript:submit()">
</td>
```

```
//Retrieve member information from the database
String sqls="SELECT iddistinction FROM distinction";
Statement stmt=dis.getstmread();
ResultSet rs=stmt.executeQuery(sqls);
while(rs.next()){
    if(rs.getInt(1)==6){
        break;
    }
else{ iddistinction=rs.getInt(1);
    }
}%>
<option value="<%=iddistinction%>"
<%
    if(iddistinction==select){
        out.print("selected");
    }%>
><%=iddistinction%>level member</opt>
```

It can be seen that in the membership level page, the function of submitting two forms on the same page has been added.

#### 4. Conclusion

The design and development of an online bookstore system based on JSP is based on the analysis and design of the needs of a large local bookstore. The system consists of two subsystems: front-end and back-end. The front-end system implements functional modules such as online book purchase, dynamic query, user registration, and login. The backend system is responsible for maintaining and managing the book information, user information, and order information in the database, ensuring the consistency of data in the database. I am mainly responsible for the requirement analysis, design, coding, and testing of the system backend in the design of this system, while the frontend is completed by other classmates.

In the design and development process of the system, object-oriented methods and modular design were adopted to standardize the website based on the characteristics of online bookstores and the functional requirements of the system, making the entire system process clear and logically reasonable, creating favorable conditions for the implementation of the system.

This system is built using popular technologies such as Java and JSP, and is an application based on the B/S application architecture. It separates business logic, front-end pages, and data storage layers, and any changes in one layer will not affect the other two layers, making the system more robust and flexible, and able to adapt to the continuous changes and developments of the system.

The client supported by the system is a basic browser. This system has a user-friendly interface, clear modules, simple operation, and easy to use. It is easy to understand and learn at a glance.

#### References

- [1] Zhu Guojun, Liu Wenye, "An introduction to HTML, the Hypertext Markup Language", Computer Knowledge and Technology,1999(07),P39.
- [2] Sun Yurong, Wu Lenan,"Unique web applet -JavaApplet", Digital Communications, 1996 (04),P30-32.

- [3] Li rui-ge," Computer software development of Java programming language and applications", Programmers, 2022(06), P15-16.
- [4] Wang chun-li, "Network programming using Java", CHINA SCIENCE AND TECHNOLOGY INFORMATION, 2006(04), P186-187.
- [5] LIU Jin-ding, YAN Han. Improved AODV routing algorithms based on neighbor survival ratio [J]. Application Research of Computers, Vol 27 No.3 Mar. 2010.