Computer Life
ISSN: 1819-4818 | Vol. 12, No. 1, 2024

Design of Front Desk in Electronic Mall

Yunzheng Ding

School of Information Engineering, Jingdezhen Ceramic university, Jingdezhen 333403, P.R.China
* **Corresponding author**: Yunzheng Ding (Email: jciddyz@163.com)

Abstract: The main purpose of business website design is generally to achieve the interconnection of enterprise products and services through website promotion, and to keep customers informed of the company and the latest product information provided by the company at any time. At the same time, it also provides customers with timely online services and order processing functions. The online shopping system mainly provides online shopping services centered on displaying and ordering laptops and related IT products. It is a platform for establishing more direct communication and exchange between enterprise products and customer service; Promoting one's own product is one of the main purposes for enterprises to go online; Present your products to customers, allowing them to freely choose and purchase the company's products through the website.

Keywords: online shopping; JSP; MySQL; Hibernate; Struts.

1. Requirements Analysis

The online shopping system is mainly a system that realizes functions such as searching for goods online, purchasing goods, and generating corresponding orders. A typical online shopping system generally requires modules such as dynamic display of product information, shopping cart management, user information registration and login management, and order processing.

According to the basic requirements of the online shopping mall front-end system, the specific tasks that this system needs to complete are as follows.

1.1. Product search

When users enter the system, they should be able to view the latest product information by category on the main page, for example, by different laptop brands. At the same time, it should also provide the function of quickly querying the required product information based on keywords such as product name or model.

1.2. Management of shopping cart

When users choose to purchase a certain product, they should be able to record the corresponding product information, such as price, quantity, etc., in the shopping cart and allow users to return to the other product information query page to continue selecting other products. At the same time, users should also be able to perform operations such as deleting purchased items, adding new items, and clearing the shopping cart in the shopping center cart. After the order corresponding to the shopping cart is generated, the information in the shopping cart will be automatically cleared.

1.3. Order processing

After the user chooses to go to the cashier, they are prompted to choose the delivery method and payment method, and the corresponding order record is generated to facilitate the website's distribution personnel to process subsequent shipments and deliveries based on the order information. At the same time, users can also enter the order management page at any time, query order information related to themselves, and cancel orders at any time.

1.4. Member information management

In order to purchase goods, users need to register and log in correctly. The related information generated from this, such as contact information, delivery location, etc., needs to be managed by the system. Users should also be allowed to modify their relevant information.

2. Overall Design

Based on the actual needs of customers during the software development process, in addition to coding work, the overall architecture of the entire software development project will also be determined, including system selection, determination of operating environment, division of functional modules, and design of user interface.

2.1. Design Principles

In the initial stage of software system development, it is generally necessary to select the appropriate development tools and software architecture based on the actual functional requirements of the system. The reliability and stability of online shopping systems have relatively high requirements. When designing this system, popular B/S designs include patterns based on JSP, ASP, PHP, CGI, and J2EE. Compared to others, PHP has relatively simple functions and is not suitable for making large programs; However, the efficiency of CGI is relatively low, so it is not considered either. Because the system does not have the original basic platform to expand and does not require too much interaction with other systems,

the use of J2EE patterns cannot reflect the advantages of J2EE itself. JSP is one of the core technologies of J2EE, which can be upgraded to J2EE programs at any time.

The project ultimately believes that using JSP is a more suitable choice at this stage, and choosing the Struts architecture as the main framework for development and Hibernate as the data persistence processing layer takes into account its high-speed development efficiency, high code reusability, easy maintenance, and other advantages. The ultimate goal is to improve the reusability of the underlying business logic of the system, increase system scalability, and reduce system maintenance costs.

2.2. Functional module division

The key to system development lies in system design, and the results of user interface design directly determine the user's evaluation of the system. Therefore, a good user interface design is a necessary condition for the success of the system, especially for the design of pages in business systems.

Fig.1 shows the hierarchical structure of the front-end page of the online shopping system.

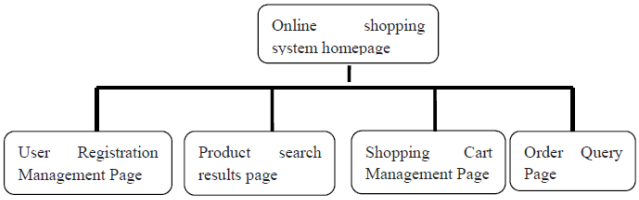

Fig.1 Front page hierarchy diagram

The hierarchical structure diagram of this item shows the link relationship between the homepage and each first level page. The link relationship between the first level page and the related second level pages will be detailed in the following chapters. Table 1 shows the settings for each column of the business system website determined from customer requirements.

Table 1 Column Settings for Front Desk Website of Online Shopping System

column	explain
home page	The homepage needs to provide the following operation interfaces: website hotspot recommendation product display, navigation of various sub columns, customer registration and login interface, product classification query and keyword search interface, and hot selling product ranking.
User registration and login management	Implement user registration and login management functions. In addition, an operation interface for modifying user information should also be provided.
Product search	Provide product classification display, search by product related keywords, and display hot selling product rankings, etc.
User shopping management	Provide display and management of user shopping carts, order generation and management, display information of payment methods, etc.

Based on the relevant settings of each column listed in the table above, the following functional modules of the online shopping system can be determined.

2.2.1. User registration and login management module

The user registration and login management module is used to establish a fixed customer group for the enterprise, and can record customer profiles in detail, so that new products and enterprise dynamics can be easily conveyed to every customer through the customer profile library. This functional module can provide the following sub functions:

Users register online;

User login management;

User profile modification;

Provide online services to members.

Only users who have logged in and verified their identity can proceed with subsequent processing such as online shopping. When the user finishes shopping and leaves the website, they can choose to log out and clear the corresponding login success information.

2.2.2. Product search and query module

The product search and query module is used to provide users with convenient search for the desired products and to understand relevant product information. At the same time, timely feedback on the corresponding product sales situation to customers to guide their shopping choices. This module should be able to provide the latest product information online at any time. This information needs to be updated regularly, and company customers can access the latest product information at any time. When customers choose a certain product, a detailed product introduction can be displayed, providing a comprehensive product description. Customers can fully understand all products and flexibly choose from them.

In addition, in order to enable customers to see the required product information as soon as possible, this module provides a search function. Customers can search for product information to quickly and accurately find the products they

need, avoiding unnecessary browsing of multiple pages to find suitable product information. This functional module should be able to provide the following sub functions: Retrieve product information by manufacturer classification; Keyword search for product information; The ranking list of best-selling products is displayed.

2.2.3. Shopping management module

The shopping management module is used to promote the company's products, allowing customers to freely choose to purchase the company's products through the website. This module can save the selected product information in the shopping cart after the user selects the corresponding product to purchase, until the user confirms the purchased product at the checkout, and generates the relevant order after confirmation. During the website residency period, users can view their shopping cart at any time and perform related operations on generated orders. This functional module can provide the following sub functions:

Customers can view the current ordering status in their shopping cart at any time and make modifications to it at any time;

Customers can fill out orders online, freely choose to book products, and manage reservations;

Registered customers can check detailed order content after passing membership verification.

3. Database Design

3.1. Logical design of database

According to the processing requirements of the online shopping system, the design and functions of the corresponding data table are as follows:

Basic Product Information Table: Stores information about products sold in online shopping malls

Basic Information Table for Product Classification: Stores information on product classifications provided by online shopping malls

User Basic Information Table: Stores basic information of online mall users

Order Information Table: Stores basic order information related to users

Order Item Information Table: stores specific item information related to the order, including the price and quantity of the products involved in a certain order

Based on the above requirements analysis, after determining the primary key fields of each table, the relationships between the tables were established based on the relationships between the relevant fields of the tables.

The classification number "sortid" in the product table is related to the "id" in the product category table, the "id" in the order table is related to the "orderid" in the order detail table, and the "productid" in the order detail table is related to the "id" in the product table. The "id" in the member table is related to the "userid" in the order table. The basic structure of each data table in the online shopping system is as follows.

3.1.1. Product list

A product table is a data table used to store the basic information of each product in an online shopping mall. It is the basis for displaying, maintaining, and managing product data. Tables 2 show the fields and descriptive information contained in this table.

Table 2 Basic Product Information Table

field name	description	type	length	allowed to be empty or not	primary key
id	Product number	INTEGER		N	Y
sortid	Product classification number	INTEGER		N	N
name	Product Name	VARCHAR	50	N	N
price	commodity price	DOUBLE		N	N
saleprice	Sales price	DOUBLE	4	N	N
descript	Product Description	TEXT	500	N	N
contents	Detailed product introduction	TEXT	2000	Y	N
saledate	Shipment date	DATE		N	N
salecount	Sales quantity	INTEGER		Y	N
image	Storage path for product cover images	AVRCHAR	50	Y	N

In order to display promotional images of the corresponding products on the page, an "image" field was designed in the data table to save the path of the corresponding product images, so that the corresponding images can be obtained and displayed according to this path in the future.

3.1.2. Product category table

The basic information table for product classification (sort) records the relevant information of various product classifications in online shopping malls, in order to facilitate the classification and display of product information. The field settings are shown in Table 3.

Table 3 Product Category Table

field name	description	type	length	allowed to be empty or not	primary key
id	Product classification number	INTEGER		N	Y
name	Product classification Name	VARCHAR	40	N	N

In order to facilitate users in finding the required product information, the displayed products can be classified and managed by brand.

3.1.3. Membership form

The member table is used to store the information of registered users in online shopping malls, including their names, contact information, and other information. The field settings are shown in Table 4.

Table 4 Membership Table

field name	description	type	length	allowed to be empty or not	primary key
id	User ID	INTEGER		N	Y
username	user name	VARCHAR	20	N	N
password	password	VARCHAR	20	N	N
realname	User Name	VARCHAR	20	Y	N
tel	User phone number	VARCHAR	20	Y	N
address	User address	VARCHAR	100	Y	N
zip	zip code	VARCHAR	6	Y	N
email	User email address	VARCHAR	50	Y	N

3.1.4. Order form

The orders table is used to store specific order information, and its field settings are shown in Tables 5.

Table 5 Order Table

field name	description	type	length	allowed to be empty or not	primary key
id	Order number	INTEGER		N	Y
orderno	Generate order number	VARCHAR	50	N	N
userid	User ID	INTEGER		N	N
realname	Consignee Name	INTEGER	20	N	N
Address	Receiving address	VARCHAR	100	N	N
zip	zip code	VARCHAR	6	Y	N
tel	Contact phone number	VARCHAR	20	Y	N
payment	Payment method	VARCHAR	20	Y	N
email	Email address	VARCHAR	50	Y	N
memo	Remarks Description	TEXT	2000	Y	N
price	price	DOUBLE		Y	N
time	Order generation time	VARCHAR	20	Y	N
tag	Flag whether the order has been processed	INTEGER		Y	N

3.1.5. Order breakdown table

The structure of a Shopcart is shown in Tables 6, which is used to store specific item information related to an order, including the prices and quantities of the products involved in a certain order.

Table 6 Order Entry Information Table

field name	description	type	length	allowed to be empty or not	primary key
id	number	INTEGER		N	Y
orderid	Order number	INTEGER		N	N
bookid	Product number	INTEGER		N	N
count	Order quantity	INTEGER		N	N
price	The corresponding price for this item	INTEGER		N	N

3.2. Script for creating data tables

After the structure of the data table is determined, the creation of the data table can be completed in MySQL. Below is the SQL script for creating the corresponding data table.

3.2.1. Basic product information table

```
create table product
(
id integer primary key,
sortid integer not null
references sort (id) on delete cascade,
name varchar (50) not null,
price double not null,
saleprice double not null,
descript text (500) not null,
```

```
contents        text(2000)        null ,
saledate        double            not  null ,
salecount       integer           null ,
image           varchar(50)       null  )
```

3.2.2. Basic information table of product classification

```
create  table  sort  (
  id       integer          primary  key ,
  name  varchar(40)    not  null  )
```

3.2.3. Basic user information table

```
create  table  member  (
  id              integer            primary  key ,
  username     varchar(20)      not  null ,
  password     varchar(20)      not  null ,
  realnaem     varchar(20)      null ,
  tel             varchar(20)      null ,
  address      varchar(100)     null ,
  zip             varchar(6)       null ,
  email          varchar(50)      null  )
```

3.2.4. Order information table

```
create    table    orders  (
  id              integer            primary  key ,
  orderno      varchar(50)      not  null ,
  userid         integer            not  null
 references  member (id)  on  delete  cascade ,
  realname     varchar(20)      not  null ,
  address      varchar(100)     not  null ,
  zip             varchar(6)       null ,
  tel             varchar(20)      null ,
  payment      varchar(20)      null ,
  email          varchar(50)      null ,
  memo         text(2000)       null ,
  price          double           null ,
  time           varchar(20)      null ,
tag            integer            null  )
```

3.2.5. Order entry details table

```
create    table    shopcart  (
  id              integer        primary  key ,
  orderid       integer        not  null
  references  orders(id)  on  delete  cascade ,
  productid     integer        not  null
  references    product(id)  on  delete  cascade ,
  count          integer        not  null )
```

4. Detailed Design

In the user presentation layer, there are mainly some related JSP pages that should be placed in the root directory of the corresponding project.

4.1. Online mall homepage

The main function of this page (index. jsp) is to provide relevant links for subsequent pages, including user registration, product inquiries, shopping cart viewing, and order viewing. The specific corresponding page files and functions are as follows.

Top. JSP page: responsible for displaying the navigation toolbar at the top of the page. On this page, different page effects are displayed in the main frame at the bottom right by clicking on different hyperlinks to make corresponding requests.

Bottom. JSP page: responsible for displaying copyright information at the bottom of the page.

Left.jsp page: Provides a user registration, login, and product query interface.

Main. JSP page: Display the latest recommended product information and product sales rankings.

The core code for implementing this function in top.jsp is as follows:

```
if (member==null){                    // User name error

    errors.add(ActionMessages.GLOBAL_MESSAGE,
          new ActionMessage("errors.loginFail"));
          if (!errors.isEmpty()) {
                saveErrors(request, errors);
          }
          return   mapping.findForward("toWrong");
    }
    else{
          String dbPassword = member.getPassword();
         if (dbPassword == null || !dbPassword.equals(psw))
{   // password error

    errors.add(ActionMessages.GLOBAL_MESSAGE,
                    new ActionMessage("errors.loginFail"));
               if (!errors.isEmpty()) {
                    saveErrors(request, errors);
    }
               return   mapping.findForward("toWrong");
          }
          else{
              HttpSession session = request.getSession();

    session.setAttribute(Constants.LOGIN_USER_KEY,m
ember);
          return      (mapping.findForward("toSuccess"));//
Login successful }
```

4.2. Display product details page

The main function of this page (detail. jsp) is to display detailed information of the selected product for users to make

purchases. The detailed information of the specified product will be displayed on this page. If you click "Order" on this page, a "buy. do? ProductId=<%=product. getId()%>" request will be submitted for processing after placing it in the shopping cart.

The core code for implementing this function is as follows:

```
DynaActionForm productIdForm = (DynaActionForm) form;
Integer productId = (Integer)productIdForm.get("productId");
HttpSession session = request.getSession(); // Get product details with Hibernate
Product product=new DbOperate().getProduct(productId.intValue());
session.setAttribute(Constants.CUR_PRODUCT_KEY ,product);
        return (mapping.findForward("toDetail"));
```

4.3. Display shopping cart information page

On this page (basket. jsp), users can set purchase quantities, clear shopping carts, and redirect to the checkout counter. This page is accessed after selecting to purchase a certain product. All existing purchase records in the user's shopping cart will be displayed on this page. At the same time, it provides an operation interface for updating shopping carts, which can modify purchase quantities or delete purchase record records. Click the "Clear Shopping Cart" button to clear the information in the shopping cart. Meanwhile, if the user confirms to go to the cashier, they can click the "Cashier Payment" button to jump to the cashier processing page. Click the "Confirm" button to submit an "updateShopCart. do" request, which updates the shopping cart based on the quantity filled in by the user and whether the item has been deleted. Click the "Clear Shopping Cart" button to submit a "clearShopCart. do" request for clearing the shopping cart. Clicking the "Cashier Payment" button will redirect the submission to the "cash. jsp" request for order generation processing.

The core code for implementing this function is as follows:

```
List shopCartList=(List)session.getAttribute
(Constants.SHOPCART_KEY);
ShopCart shopCart = null;
if(shopCartList != null ) {
for(int i = 0;i < shopCartList.size();i++)
    {    shopCart = (ShopCart)shopCartList.get(i);  }
```

4.4. Display shopping cart processing information page

This page (basketOk. JSP) is the page that the user jumps to after clicking the "Confirm" button on the shopping cart processing page. Click the "Confirm" button on this page to redirect to the cash processing page at the checkout counter. It should be noted that when entering this page, if the system finds that the user has not registered and logged in legally, after clicking the "Confirm" button, it will prompt the user to register and log in first, so as to obtain relevant user information when generating orders.

The core code for implementing this function is as follows:

```
HttpSession session = request.getSession();
Vector shopCartList=(Vector)session.getAttribute
(Constants.SHOPCART_KEY);
    if(shopCartList!=null){
       shopCartList.clear();

    session.setAttribute(Constants.SHOPCART_KEY,shopCartList);
        }
        return (mapping.findForward("toShopCartOk"));

    }
```

4.5. Online mall user registration page

On this page (registMain. JSP), users can enter the corresponding registration information in the form, click the "Register" button, and submit a "register. do" request, along with the relevant registration information of the new user, to facilitate the processing of user registration.

The core code for implementing this function is as follows:

```
DbOperate db=new DbOperate();
    Member member=db.getMember(username);
if (member==null) {
    member=new Member();
    member.setUsername(username);
member.setPassword(psw);
    member.setRealname(name);
    member.setTel(tel);
    member.setEmail(email);
    member.setZip(zip);
    member.setAddress(address);
    db.save(member);
     return (mapping.findForward("toRegistRight"));//login was successful
    }
else{
    ActionMessages errors = new ActionMessages();
    errors.add(ActionMessages.GLOBAL_MESSAGE,
```

```
new ActionMessage("label.registError"));
if (!errors.isEmpty()) {
        saveErrors(request, errors);
    return mapping.findForward("toWrong"); // login has failed
}
```

4.6. Login success display page

This page (loginOkLeft. jsp) is displayed after correctly entering user information and verifying login. Relevant user information will be displayed on this page.

The core code for implementing this function is as follows:

```
Member member = (Member)session.getAttribute
(Constants.LOGIN_USER_KEY);
List sortList = (List)session.getAttribute
(Constants.SORT_LIST_KEY);
member.getUsername();
member.getRealname();
member.getEmail();
member.getTel();
```

4.7. Cashier product display page

This page (cash. jsp) is the page that jumps to after clicking "checkout payment" on the shopping cart page. The page is mainly used to display information about the selected product for user confirmation. The source code of this program is similar to "basketOk. jsp". The core code for implementing this function is as follows:

```
List shopCartList = (List)session.getAttribute
(Constants.SHOPCART_KEY);
    ShopCart shopCart = null;
    double total=0;
    shopCart.getPrice();
    shopCart.getCount();
shopCart.getPrice()*shopCart.getCount();
```

4.8. Fill in the recipient information page

This page (cashOk. JSP) is the page that the user redirects to after confirming the corresponding product information in the cashier. On this page, first obtain the corresponding user information and provide the corresponding form to prompt the user whether to change the recipient information and payment method. After the user clicks the "Confirm" button, the order can be generated. On this page, when the "confirm" button is clicked, a "createOrder. do" request will be submitted to generate an order based on user information and information about the purchased product.

The core code for implementing this function is as follows:

```
Member member = (Member)session.getAttribute
(Constants.LOGIN_USER_KEY);
    if(member != null ){
}
    else{
     out.println("<center><font size=5 color=red>
User not logged in </font></center>");
    }
```

4.9. Confirmation order generation page

This page (orderOk. JSP) will display the order information generated based on the content of the current user's shopping cart and user information.

The core code for implementing this function is as follows:

```
// Using Hibernate to complete database related operations
    DbOperate db=new DbOperate();
    Order order=new Order();
// Save Order
    SimpleDateFormat df = new SimpleDateFormat("MMddhhmmss");
    String orderno = member.getUsername() + df.format(new Date());
    order.setOrderno(orderno);
    order.setUserid(member.getId());
    order.setRealname(realname);
    order.setAddress(address);
    order.setZip(zip);
    order.setTel(tel);
    order.setEmail(email);
    df  = new SimpleDateFormat("yyyy-MM-dd hh:mm");
    order.setTime( df.format(new Date()));
    order.setPayment(payment);
    order.setMemo(memo);
    order.setTag(0);
    db.save(order);
    // Save shopping cart information to order details table
        order=db.getOrder(orderno);
        double totalPrice=0;
        ShopCart shopCart = new ShopCart();
      for(int i = 0; i < shopCartList.size() ;i++){
      shopCart = (ShopCart)shopCartList.get(i);
      shopCart.setOrderid(order.getId());
      db.save(shopCart);
// Modify the sales quantity of the product
    shopCart.getProduct().setSalecount(shopCart.getProduct().getSalecount() + shopCart.getCount());
```

```
db.update(shopCart.getProduct());
totalPrice=totalPrice+shopCart.getPrice();
```

4.10.Display current user order information page

This page (orderDisplay. jsp) is the page that users navigate to after clicking on "My Orders" in the navigation bar at the top of the page. All order information for the current user is displayed on this page, and the user can also click "cancel this order" to delete the order information. On this page, after clicking the "Cancel This Order" hyperlink, a "delOrder. do? OrderNo=%=order. getOrderno()%" request will be submitted to delete the corresponding order information.

The core code for implementing this function is as follows:

```
Member member = (Member)session.getAttribute
(Constants.LOGIN_USER_KEY);
   // Determine whether the user is logged in
if (member==null) {
   errors.add(ActionMessages.GLOBAL_MESSAGE,
        new ActionMessage("errors.userUnLogin"));
   if (!errors.isEmpty()) {
            saveErrors(request, errors);
        }
  return mapping.findForward("toWrong");  // User not logged in
        }
   else {
        DbOperate db=new DbOperate();
        List orderList=db.getOrders(member.getId());

   session.setAttribute(Constants.ORDER_LIST_KEY,orderList);
        return (mapping.findForward("toOrderList"));
        }
```

4.11.Product search result display page

This page (listMain. JSP) is the result display page that is redirected to after selecting the product category or entering the keywords of the product to be searched. The hyperlink provided on this page can also lead to the display page of product details.

This page is the result display page where users click on keywords in the left frame of the homepage and click the "Search" button. After submitting a "search. do? ShrtId=-1&pageId=0" request, they are redirected to search according to keywords. At the same time, after selecting the category of the corresponding product, jump to the result display page of searching by keywords. At the same time, after selecting the corresponding product category, a request will be submitted to "search. do? SortId=<%=sort. getId()%> &keyword= &pageId=0" to query the product information of the specified category and display it on the page.

The core code for implementing this function is as follows:

```
ActionMessages errors = new ActionMessages();
List productList=new DbOperate().getMatchProducts
(sortId.intValue(),keyword);
        HttpSession session = request.getSession(true);
// No product found for processing
if (productList.size()==0){
   errors.add(ActionMessages.GLOBAL_MESSAGE,
        new ActionMessage("errors.noProduct"));
  if (!errors.isEmpty()) {
            saveErrors(request, errors);
                }
        PageForward="toWrong";
}
// Get total number of pages
if
(productList.size()%Constants.PRODUCT_PAGE_SIZE
==0){
     pageCount=productList.size()                    /
Constants.PRODUCT_PAGE_SIZE;
          }
else{
     pageCount=productList.size()                    /
Constants.PRODUCT_PAGE_SIZE+1;
        }
// Pagination display
If          ((productList.size()>iPageId          *
Constants.PRODUCT_PAGE_SIZE )
&&(iPageId>=0)){
        List  dispList=new ArrayList();
for (int i=iPageId*Constants.PRODUCT_PAGE_SIZE;
i<(iPageId+1)*Constants.PRODUCT_PAGE_SIZE;i++){
                if (i<productList.size()){
                    dispList.add(productList.get(i));
                }
    }
```

4.12.User profile modification page

This page (modifyUser. jsp) is the page that the user jumps to after clicking on "Change Data" in the navigation bar at the

top of the page. Display the relevant information of the current user in the form on this page for the user to modify their personal information. This page is basically the same as the registration page.

The core code for implementing this function is as follows:

```
if (member!=null) {
                member.setPassword(psw);
                member.setRealname(name);
                member.setTel(tel);
                member.setEmail(email);
                member.setZip(zip);
                member.setAddress(address);

                new DbOperate().update(member);
                return
(mapping.findForward("toRegistRight"));//          Modified
successfully
            }
            else{
                ActionMessages          errors          =          new
ActionMessages();
    errors.add(ActionMessages.GLOBAL_MESSAGE,
                    new
ActionMessage("errors.needRegist"));
                if (!errors.isEmpty()) {
                    saveErrors(request, errors);
                }
                return      mapping.findForward("toWrong");
// User not registered
            }
```

References

[1] Yang Hongbo ,Wang Zhishun, “J2SE evolution history”, Programmers,2005(07),P50-52

[2] Chen Limin ,Twinings nine,” JAVA graphical interface development exploration”, Journal of Southwest University for Nationalities (Natural Science Edition), 2006(02),P405-409

[3] Hua Weizhong, Zhao Chunyun,” An in-depth look at Java threads”, Computer and Information Technology. 1997(02), P29-30+33

[4] Song Weiwei, Chen Shuzhen, Sun Xiao 'an, “ Multithreading and dual buffering in the Java language”,Electronic Computers and External Equipment,1998(06),P30-31