Design and Implementation of Light Food Mall Based on ASP.NET MVC

Yuanchun Liu and Kaili Xiong

ABSTRACT

The Light Food Mall is developed with ASP.NET MVC technology, which realizes the functions of browsing commodities, browsing recipes, collecting recipes and purchasing commodities. This paper introduces the function design, database design and implementation code of specific modules of Light Food Mall. The aim of this system is to provide users with a platform for healthy diet and create a healthy diet pattern together.

1. INTRODUCTION

With the rapid improvement of living standards in modern society, people's demand for food has also entered a high level. High-value, delicious, healthy, nutritious and simple food has become the pursuit of modern people. However, more and more people are busy with their work and neglect the management of their diet. As a result, "light food doctrine" came into being. "Light food" means simple food that can be eaten without too much time. The raw materials are mainly vegetables and fruits. So we set up such a special platform to provide users all over the country to buy light food raw materials, and jointly create a healthy diet pattern. The light food mall system is developed using C# language, ASP.NET MVC, SQL 2012. This system has many functions, such as commodity browsing, commodity searching, menu browsing, menu collection, commodity
purchasing, commodity reviewing, personal center and so on. It aims to provide users with a more professional platform for healthy diet.

2. ANALYSIS AND DESIGN OF THE SYSTEM

The functions of Light Food Mall main include the display of the home page, the registration of the website, browsing the commodities on the website, searching commodities, collecting commodities, recipe function modules, adding shopping carts, purchasing commodities, commodity reviewing, shop classification, order inventory management, modification and improvement of personal basic information, and viewing order details[1-3]. The following is a detailed description of each module of the system.

(1) Login and Registration Module: Visitors are registered as formal users, and formal users log on the website.

(2) Personal Information Display and Modification Module: formal users log on to the website and enter the personal center page to modify or improve their personal information.

(3) Recipe module: Users can browse the details of recipes and add recipes to the collection.

(4) Commodity module: Special commodities can be recommended in this system. Users can browse all kinds of commodities by category, search for commodity by entering commodity name in search box. Users can also praise commodities they like, and comments on commodities they bought.

(5) Shop classification module: Users can browse the store information and buy commodities when they enter the store. Users can also sort the commodities according to their own choices.

(6) Shopping module: Users can view the details of each commodity, add more than one commodity to the shopping cart, carry out the order settlement operation, and can operate the order.

The functional structure of the system is shown in Figure 1.

According to the data analysis of Light Food Mall, the system includes users, commodities, recipes, orders, reviews, shops and other entities[4]. The relationship model is as follows.

Users: (User number, username, password, head sculpture, sex, date of birth, e-mail, telephone number, security questions, question answers, collection quantity);

Commodities: (commodity number, shop number, commodity name, unit
price of commodity, commodity map, commodity details map, approval ratio, whether private customization, shelf time, inventory, sales volume;
Order: (Order number, user number, order time, quantity, status, address number);
Recipes: (recipe number, recipe name, classification, pictures, introduction, practice, relevant pictures, ingredients, collection, Classification Number);
Classification of recipes: (Classification Number, Classification Name, Introduction);
Address: (Address number, user number, province, city, detailed address, default or not);
Collection: (Collection number, user number, recipe number, collection time);
Comments on commodities: (comment number, user number, commodity number, comment content, comment time);
Order details: (order details number, order number, commodity number, unit price, quantity, total price);
Commodity comment reply: (reply number, user number, comment number, reply content, reply time);
Shopping cart: (shopping cart number, user number, commodity number, quantity, unit price, total price);
Shop: (Shop number, user number, shop name, shop introduction, shop sales, shop pictures, shop top pictures).
3. REALIZATION OF SYSTEM

The implementation of the system is illustrated by retrieving commodities and adding shopping carts.

Users can view on the right side of the home page then search for the goods they want to buy. The main codes for retrieving goods are as follows [5].

```csharp
public ActionResult MallSearchContent(string searchString, string currentFilter, int? page)
{
    var goods = goodsManager.SelectAllGoods();
    // Method of querying all commodities in BLL layer
    if (searchString != null)
    {
        page = 1;
        // If the search content is empty, leave it on the original page
    }
    else
    {
        searchString = currentFilter; // Get the search string
    }
    ViewBag.CurrentFilter = searchString;
    if (!String.IsNullOrEmpty(searchString))
    {
        goods = goodsManager.SelectSearchGoods(searchString);
        // Calling the method in BLL Layer to search for goods
    }
    int pageSize = 12;
    int pageNumber = (page ?? 1);
    if (Request.IsAjaxRequest())
    {
        return PartialView("MallSearchContent",
            goods.ToPagedList(pageNumber, pageSize)); // Jump to search content view
    }
    else
    {
        return View(goods.ToPagedList(pageNumber, pageSize));
    }
}
```

Users can add goods they want to buy to shopping carts. The main code for
adding shopping carts is as follows:

```java
public ActionResult Carts(ShoppingCarts carts)
{
    int goodid = Convert.ToInt32(Session["goodid"]);
    // Read commodity ID
    int userid = Convert.ToInt32(Session["UserID"]);
    // Read User ID
    int result = shoppingcarsmanager.ShopCartsCount(userid, goodid);
    if (result > 0)
    {
        return Content("<script>alert('The merchandise already exists in the shopping cart.\n')\nhistory.go(-1)</script>");
        // Determine if the user has purchased the product
    }
    else
    {
        carts.UserID = userid;
        carts.GoodsID = goodid;
        carts.UnitPrice = Convert.ToDecimal(Request.Form["price"]);
        // Set unit price
        carts.Number = Convert.ToInt32(Request.Form["Im_Amount"]); // Set the purchase quantity
        carts.TotalAmount = carts.Number * carts.UnitPrice;
        // Total amount
        shoppingcarsmanager.AddShopCarts(carts);
        return Content("<script>alert('Added Successfully')\n\nhistory.go(-1)</script>"); // Success Tips
    }
}
```

4. CONCLUSIONS

The system realizes user registration, user personal information management, search, commodity classification, collection of recipes, adding shopping carts, viewing and deleting orders and other functions. There is well-organized
classification of goods and stores in the system. Beautiful and generous system interface allows users to have a better experience.

REFERENCES