Technology Application Base on ASP.NET Model View Controller

Yuanchun Liu* and Heyi Zhu
School of Software, Jiangxi Normal University, Nanchang, China

*Corresponding author email: ycliuliuliu@jxnu.edu.cn

Abstract. The Music Mall System is developed by ASP.NET model view controller technology. It realizes the functions of registration and login, song search, playing songs, collecting and commenting, shopping mall and so on. This paper firstly introduces the function design, the database design of the music mall system, secondly describes the architecture and finally elaborates the implementation codes of the specific modules. Using this system, users can quickly search and listen to relevant music, buy related music peripheral products. Through reasonable page layout and detailed user experience, users can feel happy and satisfied.

Keywords: ASP.NET; Music; Mall.

1. Introduction
With the progress of the times and the rapid development of society and technology, the development of the Internet industry is also changing with each passing day. People can no longer study and work without the Internet, the Internet has a great influence on people's production and lifestyle. At the same time, under the influence of the current music trend, music has entered into thousands of households, loved by the majority of students and office workers. Music has been everywhere, and has been inseparable from us.

Traditional music websites usually only provide users with services related to listening to songs, and will not jump out of this thinking and provide users with extra high-quality services. On the basis of realizing the function of listening to songs, this system adds the module of Mall to combine listening to songs with Mall. Some music related products, such as MP3 Walkman, headphones, CDs, etc., will be put on the shelves in the mall. The products will be combined with music culture to improve the user's experience. Through this system, you can first listen to songs like other music websites, and then you can buy related peripheral products according to your own needs. It not only improves the users' experiences, but also increases the revenue source of the system.

2. Analysis and Design of the System
Music mall system aims to provide music lovers with a platform for sharing and exchanging music. It provides users with the best browsing experience with a simple interface, complete functions and rich music library. The system includes registration module, login module, song and commodity search module, song and commodity classification module, song comment module, song collection module, shopping cart module and personal center module[1-8].

The detailed description of the specific system modules is as follows:
(1)Registration module: This is the module that visitors need to upgrade to become registered users. Visitors can become registered users of the system by clicking the sign-up Button, filling in some necessary information and passing the audit, and the relevant operating rights will be enhanced.
(2) Login module: after the registered user clicks the login button and enters the correct account information, the system can log in after passing the system verification. Login users can view their personal information, as well as their own song playing history and merchandise purchase history.

(3) Personal center module: after logging in, registered users can view their own information, songs they have heard and some products they have purchased in the mall.

(4) Song and commodity search module: Visitors and login users can click the search box and input keywords to search the music library and commodity library, and the system will display the relevant content of user search.

(5) Song and commodity classification module: This module is designed to improve the display of songs and commodities, which is conducive to saving the time cost of browsing songs and commodities and improving the experience of the system.

(6) Song comment module: This module provides a platform for login users to publish their personal feelings of songs, which can improve the user's sense of participation and increase the stickiness of the system.

(7) Song collection module: This module is for login users, users can collect their favorite songs, facilitate users to listen to the collected songs, and improve the browsing experience of users.

(8) Shopping Cart module: this system provides shopping cart management function for users. Users can add different goods to the shopping cart when purchasing goods or viewing the details of goods, so as to facilitate the subsequent order generation.

The system function structure diagram is shown in Figure 1.

Figure 1. The System function structure diagram.

Combined with the relationship between the modules in the music mall system, we can get the following entities: user, song, singer, commodity, shopping cart, address, order. The specific entity and contact attributes are shown as below:[9].

(1) User: user ID, user name, user nickname, user type, gender, head sculpture, password;
(2) Songs: song ID, song name, singer, type, release time, free or not, path, background map, MV;
(3) Singer: singer ID, singer name, type, nickname, gender, head sculpture, description;
(4) Commodity: Commodity ID, commodity name, inventory, unit price, picture, type;
(5) Shopping cart: shopping cart ID, user ID, commodity ID, commodity name, commodity picture, quantity of goods, total amount of goods, time of adding shopping cart;
(6) Address: address ID, user ID, user name, detailed address, recipient phone number;
(7) Order: order ID, user ID, commodity ID, address ID, commodity name, commodity picture, order creation time, order total amount;
(8) Song Collection: collection number, user number, song number, collection time;
(9) Song Comment: comment number, user number, song number, comment time, comment content;
(10) Play History: play history ID, song ID, user ID, play time.

3. Software Architecture of the System
The development framework of this music mall system is ASP.NET MVC (model view controller). Model is the part responsible for processing system data logic, usually model object is responsible for accessing data in database; view is the part of project responsible for processing data display, usually view is created based on data in model; controller is the part of project dealing with user interaction, usually controller is responsible for reading data from view layer and getting Take and control user input, and send data to background model[10]. The specific system architecture is shown in Figure 2.

4. Realization of System
The user enters the song name or singer name or song type to be searched in the search bar on the home page of the system, then the background controller directly accesses the song table in the database through Language Integrated Query (LINQ). After obtaining the corresponding data, it returns the corresponding song to view for front page layout. In addition, the system also has the function of fuzzy search. Users do not need to input all the names of songs or singers or song types, but only need to input a small part of information. The system will return all the data containing the keyword in the database. The main codes are as follows[10]:

```csharp
public ActionResult Search(string Search)
{
    // the user input content is fuzzy matched with the song name, singer name and song type in the song table in the database. As long as the content input by the user is included, the result will be returned to view for display
    var songs = db.Song.Where(item => item.SongName.Contains(Search) ||
```
item.Singer.Contains(Search) || item.Type.Contains(Search)).ToList();
if (songs == null) return View();
List<MySongViewModel> lists = new List<MySongViewModel>();
foreach(var s in songs)
  // go through the whole song list
  {var singer = db.Singer.Where(item => item.SingerName == s.Singer).FirstOrDefault();
   MySongViewModel model = new MySongViewModel()
   { SongId = s.id, SongName = s.SongName, SongPic = s.SongBgURL, SongURL =
     s.SongURL, Singer = singer.SingerName, SingerId = singer.id, time = null };
   lists.Add(list);
  }
return View(lists.ToList());
}

The login users of the system can add their favorite goods to the shopping cart. The method of adding
the shopping cart is as follows: after entering the mall homepage of the system, the user can click the
"buy now" button in the lower left corner of the picture of the product to enter the product details page,
and then find it in the product details page and click the "add to shopping cart" button to complete the
process of adding shopping cart. The main codes are as follows:

```csharp
public ActionResult AddGoodsToShoppingCart()
{
  var sr = new StreamReader(Request.InputStream);
  var stream = sr.ReadToEnd();
  JavaScriptSerializer js = new
  JavaScriptSerializer();
  var list = js.Deserialize<List<NewOrderRecord>>(stream);
  // Use the ternary operator to determine whether the current user has logged into the system
  string UserName = (string)Session["UserName"] != null ?
    (string)Session["UserName"] : null;
  if (UserName == null)// User not logged in
  {
    return Content("NoneUser");
  }
  int Userid = db.Users.Where(item => item.UserName == UserName).FirstOrDefault().id;
  // Get user ID
  int GoodId = list[0].GoodId;
  double Amout = list[1].Amount;// Calculate the total amount of goods
  int quantity = (int)Amout / (int)(db.Product.Where(item => item.id ==
    GoodId).FirstOrDefault().ProductPrice);
  // Calculate the quantity of goods
  string Productname = db.Product.Where(item => item.id ==
    GoodId).FirstOrDefault().ProductName;
  string ProductPic = db.Product.Where(item =>
    item.id == GoodId).FirstOrDefault().ProductPicture;
  if (Amout <= 0) {return Content("illegal");}
  // Create a new shoppingcart object and assign a value to it
  ShoppingCart shoppingCart = new ShoppingCart()
  { UserID = Userid, ProductID = GoodId, ProductAmount = (int)Amout, ProductName =
    Productname, ProductPic = ProductPic, AddShoppingCartTime = DateTime.Now };
  var result = db.ShoppingCart.Add(shoppingCart);
  // Add the object you just create to the shopping cart table in the database
  if (result != null)
  {
    db.SaveChanges();//Save changes
  }
  return View("AddGoodsToShoppingCart");
}
```
5. Conclusion
This article elaborates on the design and implementation of a music mall system based on ASP.NET MVC (model view controller) technology. The system completes functions such as the user registration and login, song search and play, collection and comment on songs, classification of songs, singers and commodity, users purchasing goods and adding goods to the shopping cart, etc. First, you can listen to songs like other music websites, and then you can buy related peripheral products according to your own needs. It not only improves the user experience, but also increases the revenue source of the system.

References
[1] Jia Xuebin, Research on Computer Network Virtual Laboratory Based on ASP.NET, 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), IEEE Press, Dec. 2017, pp.464-467.
[2] Zou Changsheng, Zhang Zixuan, He Xin. Design and Implementation of Remote Monitoring System Based on asp.net. 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation, IEEE Press, June. 2015, pp.354-357.
[3] Liu Junhui, Wang Jianping. Design and Implementation of Temperature and Humidity Control System Based on ASP.NET MVC Architecture. Proceedings of 2015 World Conference on Control, Electronics and Electrical Engineering(WCEE 2015), WIT Press, Jan. 2015, pp1069-1076.
[4] Xue Lv, Bensheng Yang, Wenquan Yao, Rongjun Jiang. A Study on Information Security Emergency Plan Management System Based on ASP.NET, 2015 International Conference on Computational Intelligence and Communication Networks (CICN), IEEE Press, Dec. 2015, pp1023-1026.
[5] Yuanchun Liu, Honghao Cheng. Design and Implementation of Photographic Community System Based on ASP.NET MVC. 2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), IEEE Press, Nov. 2019, pp.112-115.
[6] Lei Zhang, Development of an Information-Based Online Foreign Language Teaching Platform with ASP.NET, INTERNATIONAL JOURNAL OF EMERGING TECHNOLOGIES IN LEARNING, vol.14(13), 2019, pp.117-128.
[7] Yang Jing, Zhao Zhen, Development and Implementation of Computer Assisted Instruction System in Physical Education Based on ASP.NET Technology, INTERNATIONAL JOURNAL OF EMERGING TECHNOLOGIES IN LEARNING, vol.14(13), 2019, pp. 145-155.
[8] Yu Yan, Design and Implementation of a Teaching Assistance Platform for College Students Based on ASP.NET, INTERNATIONAL JOURNAL OF EMERGING TECHNOLOGIES IN LEARNING, vol. 14, 2019, pp. 97-107.
[9] Abraham Silberschatz, Henry F. Korth, S. Sudarshan. Database System Concepts. McGraw-Hill Education, 2010.
[10] Adam Freeman. Pro ASP.NET MVC 5 Platform. Apress, 2014.