**Course Project Report**

**Course： Web Programming**

**Name of System：Outdoor Sports E-commerce Management System**

**Student ID： 3130104101**

**Name： Jianfu Zhang**

**Date：** **2016-6-18**

**Content**

[1 课程设计说明与业务功能简述 3](#_Toc454100022)

[1.1 课程设计说明 3](#_Toc454100023)

[1.2 业务功能简述 3](#_Toc454100024)

[2 系统分析与设计 4](#_Toc454100025)

[2.1 系统分析 4](#_Toc454100026)

[2.1.1 可行性分析 4](#_Toc454100027)

[2.1.2 用例模型 5](#_Toc454100028)

[2.1.3 系统框架描述 6](#_Toc454100029)

[2.1.4 数据字典设计 13](#_Toc454100030)

[2.2 系统设计 14](#_Toc454100031)

[2.2.1 概念结构设计 14](#_Toc454100032)

[2.2.2 逻辑结构设计 17](#_Toc454100033)

[2.2.3 范式校验 18](#_Toc454100034)

[2.2.4 物理设计 18](#_Toc454100035)

[3 系统实现 20](#_Toc454100036)

[3.1 数据库的实现 20](#_Toc454100037)

[3.1.1 数据库建立 20](#_Toc454100038)

[3.1.2 数据样本 21](#_Toc454100039)

[3.2 模块功能实现 24](#_Toc454100040)

[3.2.1 开发环境与编程技术 24](#_Toc454100041)

[3.2.2 系统操作说明 24](#_Toc454100042)

[4安装与使用说明 26](#_Toc454100043)

[4.1 环境支持 26](#_Toc454100044)

[4.2 进入系统 26](#_Toc454100045)

[4.3 注册、登录 28](#_Toc454100046)

[4.4 选购商品 30](#_Toc454100047)

[4.5 使用帮助 33](#_Toc454100048)

[4.6 管理员登录 34](#_Toc454100049)

[4.7 管理商品信息 35](#_Toc454100050)

[4.8 查询顾客信息 38](#_Toc454100051)

[4.9 查询订单信息 39](#_Toc454100052)

[4.10 Ajax应用 40](#_Toc454100053)

[5 网页结构 42](#_Toc454100054)

[5.1 网页清单 42](#_Toc454100055)

[5.2 网页功能逻辑与调用关系 43](#_Toc454100056)

[6 总结 44](#_Toc454100057)

[参考文献 45](#_Toc454100058)

# 1 Course Design Description and Business Function Description

## 1.1 Course Design Description

Develop a complete web application to obtain a complete business function. No less than 5 pages are required. The database must be used, and no less than 3 tables of business data are stored in the database. The Entity Framework (Code First or Database First) is used to operate the database. In addition, write a system description document (Word). The contents of the document are divided into the following parts: 1 business function description, 2 use case model (as far as possible detailed description of each role and use case), 3 database entities - Contact diagrams (at least two types of entities), 4 web page listings and functional logic and call relationships for each web page, 5 simple installation and usage instructions, etc. The entire project is packaged with the database file (or the backup file) and the documentation.

## 1.2 Brief description of business functions

Design an outdoor sports e-commerce management system to achieve the following functions:

1, User registration, login function;

2, User classification query product features;

3, User's shopping cart records function;

4, User order creation;

5, Enable administrators to add, delete, change, check product information function;

6, Enable administrators to query user information function;

7, Enable administrators to query sales information function;

8, Fuzzy query and multi-condition query.

# 2 System Analysis and Design

## 2.1 System Analysis

### 2.1.1 Feasibility Analysis

This chapter mainly explains the technical, economic and operational feasibilities of the system development project, and reviews various possible implementation options for reasonably meeting the development goals, and finally explains and demonstrates the reasons for the selected implementation plan.

(1) Commercial feasibility

As a new business model, e-commerce is recognized as a new economic model and catalyst for the world landscape in the 21st century. E-commerce plays an increasingly important role in changing the traditional business management model, reducing transaction costs, improving economic operating efficiency, and promoting regional economic development.

At present, at home and abroad, there are many well-known e-commerce sites, such as Amazon, Taobao, Jingdong and so on. This trend presents a new challenge to the design and implementation of database systems: how to meet the increasingly large trading needs of buyers and sellers.

This project is intended to focus on this topic, design a simple database schema to describe the above requirements, and write a program to implement an application system that satisfies the needs of users as much as possible.

(2) Technical feasibility

The technical feasibility analysis mainly analyzes whether the technical conditions can be met and whether the hardware and software can meet the needs of the developers. This project uses Microsoft Visual Studio 2015 as a development tool and uses C# programming languag. The underlying database of the system is a SQL server. The SQL server database function is very complete and easy to operate. It is suitable for the development of small and medium-sized software and systems. And the platform environment used in this project also reduces the time and cost of developing and managing their data infrastructure, allowing the system to run their most mission-critical applications with high security, reliability, and scalability. Therefore, the system's software development platform is mature and feasible. With regard to hardware, today's rapid development of technology has resulted in faster and faster hardware updates, greater capacity, higher reliability, and lower prices. Its hardware platform can fully meet the needs of this system.

In summary, the technology used in this system can fully meet the development of the system.

(3) Economic feasibility

The operating platform and development tools involved in the development of this project are all basically software that ordinary laptops or desktop computers already have, and the current market situation is: more and more e-commerce systems are becoming more popular, such as Taobao, Jingdong, Amazon, etc. Both system administrators and customers can complete the transaction process through the system. The basic interface is relatively simple and does not require other resources, so there is no economic burden.

(4) Operational feasibility

Because the system uses data and charts as the human-computer interaction carrier, users need to input relevant information during the use of the entire system, and can be selected according to the corresponding prompts. The operation is simple, the human-machine interaction interface is friendly, and the system is relatively strong. With the affinity and ease of use, users can quickly master the use of the system. Therefore, from the aspect of operational feasibility, this system is also completely feasible.

### 2.1.2 Use case model

**（1）Use case diagram**

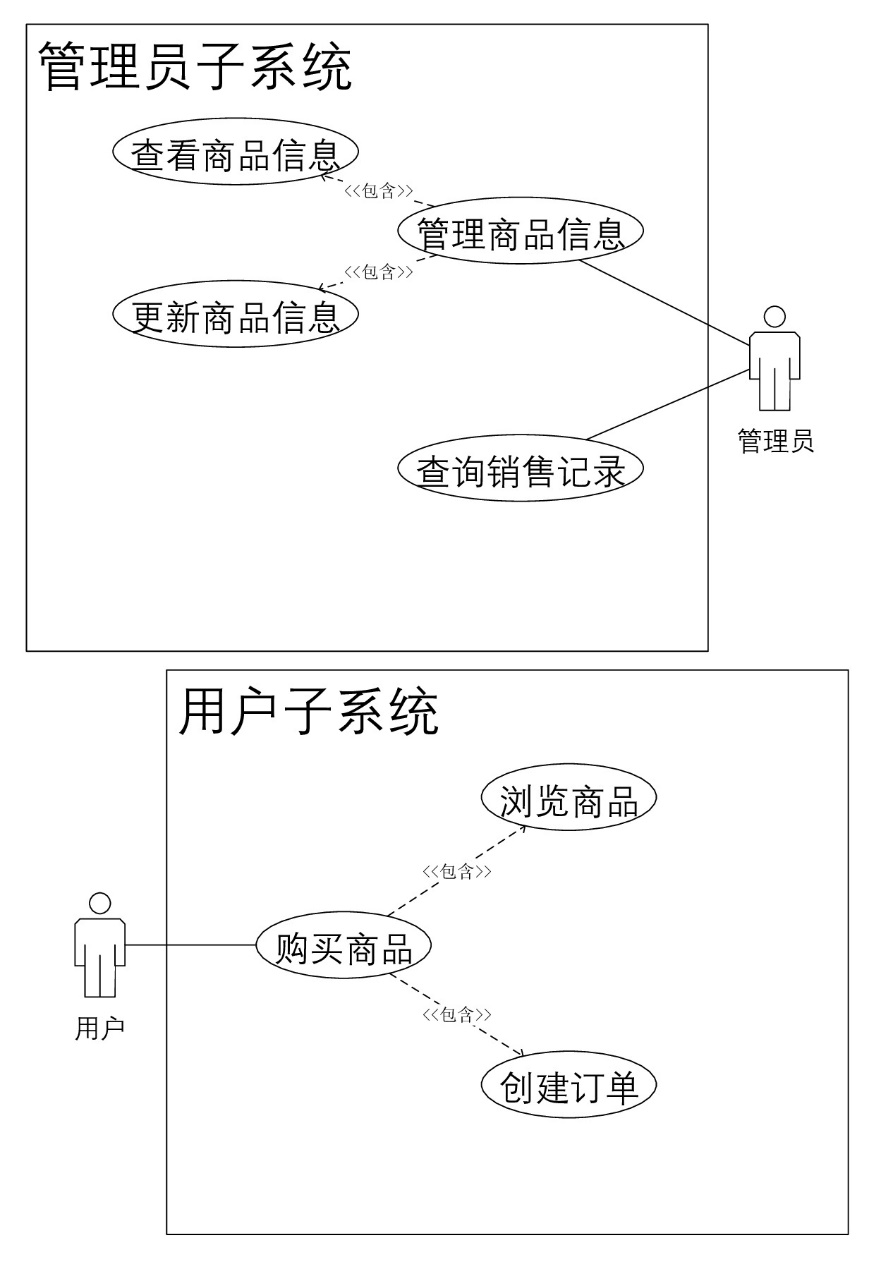


Figure 2.1 System use case diagram [[1]](#footnote-1)

The use case diagram is a powerful tool for describing the needs of system participants in the Unified Modeling Language (UML). Often, the design and development of the system require a use case diagram to accurately describe the customer's needs, providing critical guidance for the system's functionality and architecture. This step usually requires careful communication and contact with customers before they can proceed smoothly.

The quality of system analysis and design stage often determines the difficulty of system implementation and customer satisfaction. Due to the limited time of this project, at the same time, the system to be implemented in this project has a large number of application examples in reality, and in the last semester, one of my professional courses allowed me to master the analysis and design points of software development. Therefore, progress at this stage has been relatively smooth.

After summarizing the experience and personal in-depth thinking, I think the participants of the system can be divided into two: buyers and sellers. Or as shown in the use case diagram: Users and administrators.

Among them, the needs of ordinary users are embodied in the need to be able to purchase goods in an e-commerce system. At a minimum, users need to be able to browse the products, that is, to see what products, price of the products, and the basic description of the products. At the same time, users also need to be able to submit their own orders to the system to complete the shopping process.

Administrators are another integral part of the system. He needs to manage the information of the products in this system, including viewing, adding, modifying, and deleting product information. In addition, the administrator also needs to know the sales status of his shop.

This requires that the design of the database must be reasonable, so that it can fully meet the input and output of various information to ensure the reliability of data storage, and can quickly remove and deposit. The foreground display section should have a user-friendly interface. The system should also have strong security. Users with different identities cannot operate beyond their rights. And the system requires that non-legitimate users cannot operate on data.

(2) Functional requirements

Through the analysis of system functions, combined with the basic framework of other e-commerce management systems on the Internet, especially for similar products of this system that have been put into operation and obtained good evaluations, the following requirements information is summarized:

First, permission settings can be made. Administrators and ordinary users should have different functions.

Second, the administrator's needs: 1, edit product information (including add, delete, change, check); 2, query user information; 3, query sales information;

Third, 1, registration, login function, provide effective users with the permission to create an order; 2, classified search product features, allowing users to quickly locate the product of interest in a wide array of goods; 3, shopping cart function, at any time Record the goods that users need during the shopping process; 4. Create an order function to let users pass their purchase information to the administrator.

### 2.1.3 System Framework Description

**(1) System Architecture**

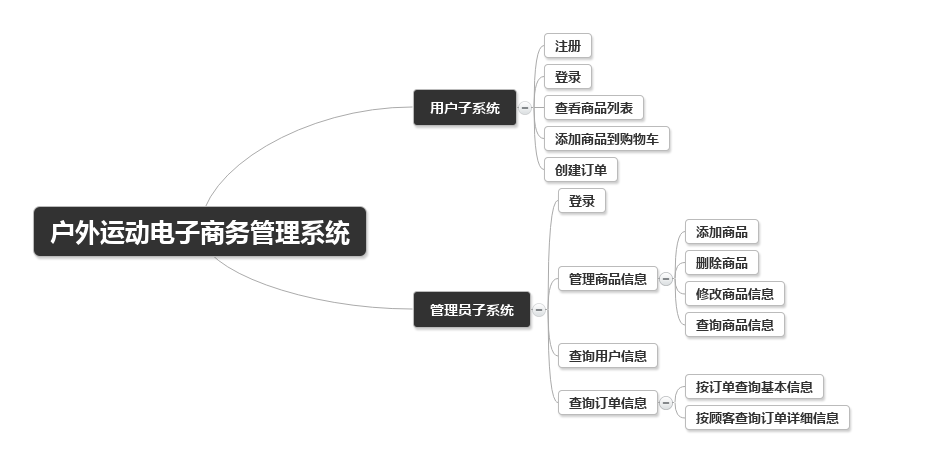


Figure 2.2 System Architecture[[2]](#footnote-2)

As shown in Figure 2.2, the outdoor sports e-commerce management system I will establish will include two major sections: the user subsystem and the administrator subsystem.

Under the user subsystem, there are five modules, namely, registering, logging in, viewing a list of products, adding items to a shopping cart, creating an order, and the like. Among them, viewing the product list, adding the product to the shopping cart, and creating an order are the core functions, satisfying the user's demand for purchasing the product. Registering and logging in as an auxiliary module helps administrators manage users and order information.

Under the administrator's subsystem, there are four major modules, namely login, managing product information, querying user information, and querying order information. Among them, the management of product information and query of order information is a core function, which satisfies the need for administrators to manage information in their own stores and to understand their own sales status.

**（2）Data Flow Diagram Design of Outdoor Sports E-commerce Management System**

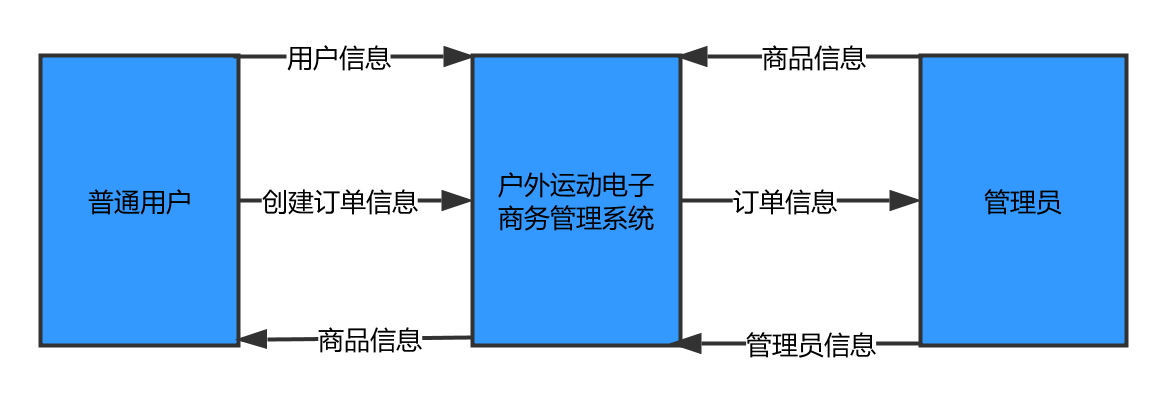
The top-level diagram of the system is shown in Figure 2.1.As shown in Figure 2.2, the outdoor sports e-commerce management system includes two major sections: the user subsystem and the administrator subsystem.

Under the user subsystem, there are five modules, namely, registering, logging in, viewing a list of products, adding items to a shopping cart, creating an order, and the like. Among them, viewing the product list, adding the product to the shopping cart, and creating an order are the core functions, satisfying the user's demand for purchasing the product. Registering and login as an auxiliary module helps administrators manage users and order information.

Under the administrator's subsystem, there are four major modules, namely login, managing product information, querying user information, and querying order information. Among them, the management of product information and inquiry of order information is a core function, which satisfies the need for administrators to manage information in their own stores and to understand their own sales status.

(2) Data flow diagram design of outdoor sports e-commerce management system

The top-level diagram of the system is shown in Figure 2.1.

Figure 2.3 System Top-level Diagram

**（3）Activity diagram**

**The activity diagram is a UML diagram used to describe the workflow of the business use case realization. Each module of the system will be described in detail through the activity diagram below.**

**1, User Registration**

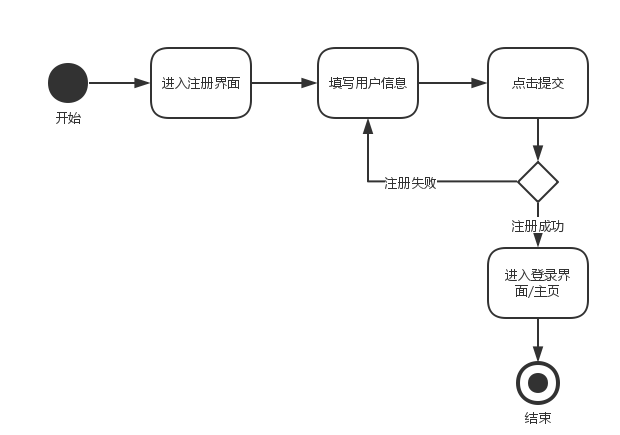


Figure 2.4 Activity Diagram of Registration Module [[3]](#footnote-3)

As shown in Figure 2.4, the registration process is relatively clear. The only thing worth noting is that after clicking on submit user registration information, the system should screen the submitted content to see whether it is received successfully, whether there is duplicate registration, etc. Once an abnormal situation occurs, The system should provide the user with the correct prompt information. In addition, after the registration is successful, the user should jump to the login screen, or directly to the shop homepage. How to achieve it should be determined according to the circumstances.

**2,User Login**

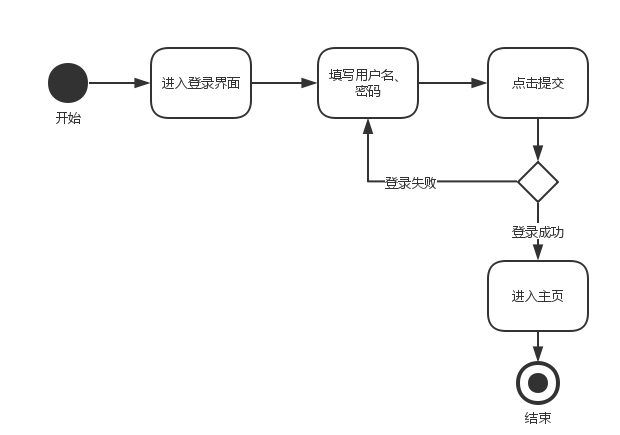


Figure 2.5 Activity Diagram of User Login

As shown in Figure 2.5, as with the registration activities, the module workflow for the login activity is also relatively clear. After the user clicks submit, if the username and password pass the system verification, the login is successful, the shop home page is entered, and shopping begins. Otherwise, if you enter an account number and password that do not match the system database, you cannot log in successfully and remain on the login screen until the login is successful.

**3,** **Users view product list**

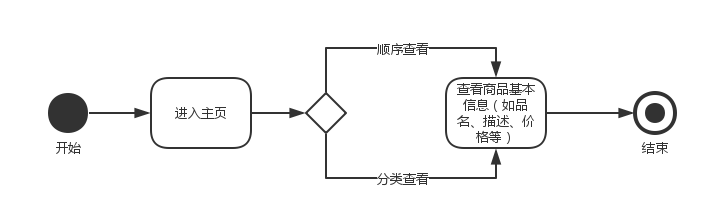


Figure 2.6 Activity Diagram of Users Viewing Product List

As shown in Figure 2.6, on the homepage of the store, users need to be concerned with some basic information about the merchandise to assist in purchasing decisions. In this case, in addition to viewing the product list in the default order, the user may be provided with a way to view the product by category in order to facilitate the user to view the product.

**4,** **Users add items to the shopping cart**

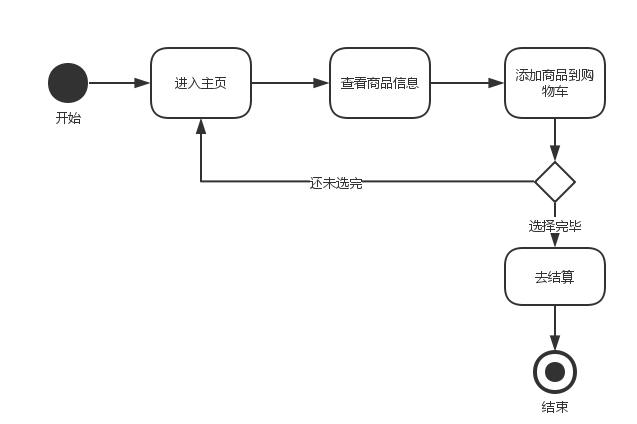


Figure 2.7 Activity Diagram of Users Adding Items to Shopping Cart

In the process of browsing the product, the user can add the desired product to his/her shopping cart anytime and anywhere, and let the shopping cart temporarily record the product he has purchased. This process can continue until the user believes that he has selected the product. This can be settled along with the checkout process.

**5,Users create orders**

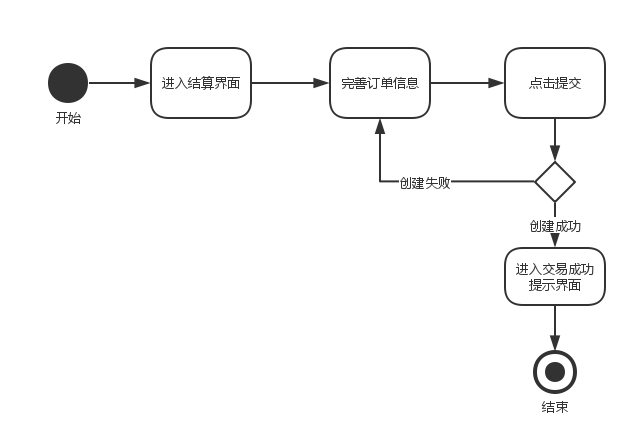


Figure 2.8 Activity Diagram of User Creating Orders

After entering the settlement interface, the user is also required to provide relevant necessary information, such as the receiving address, receiving telephone and other information, which may not be consistent with the user's own, so need to be filled out separately. If the order is successfully created, the interface for successful transactions is entered, otherwise the order filling interface is returned until the order is successfully created.

**6,** **Administrator login**

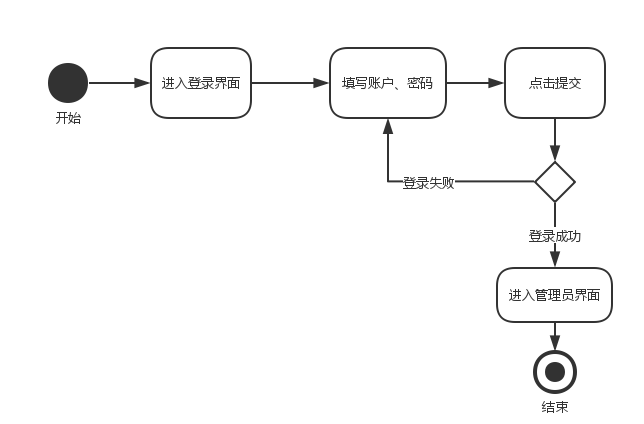


Figure 2.9 Activity Diagram of Administrator Login

There is no essential difference between the administrator login and the user login. Only the privilege and the interface are different.

**7,** **Administrators manage product information**

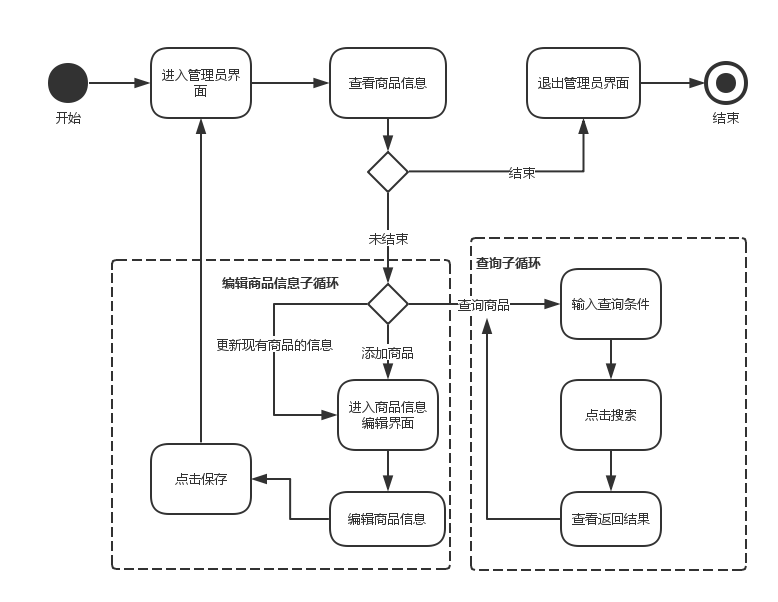


Figure 2.10 Activity Diagram of Administrators Managing Product Information

As shown in Figure 2.10, after entering the administrator interface, the administrator can view the product information. In order to improve the administrator's query efficiency, the system should provide the ability to query the product by condition. In addition, administrators should also be able to add merchandise and edit product information. In fact, the two interfaces can be combined into one, so it can be seen as the intersection of two actions.

It is worth noting that in the module for managing merchandise information, there are two sub-cycles, that is, the query product sub-loop and the edit product information sub-loop, that is, these two actions can be repeated multiple times until the administrator decides to log out.

**8,** **Administrators query user information**

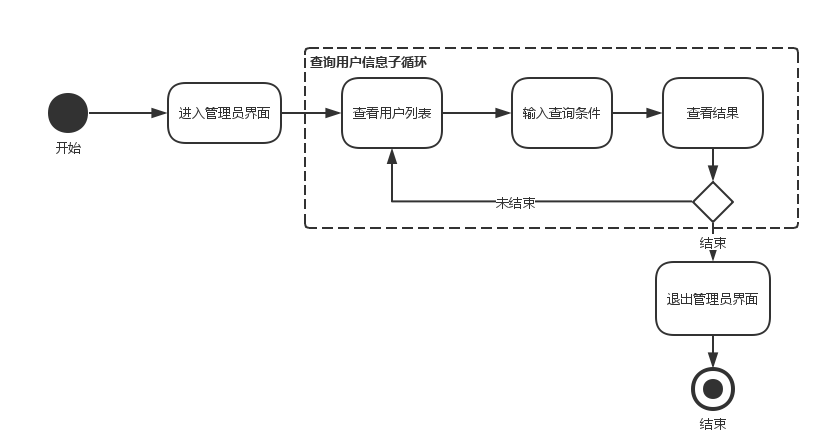


图2.11 Activity Diagram of Administrators Querying User Information

As shown in Figure 2.11, the administrator can view the registered users’ information, and can also enter the query conditions to find a specific user.

**9,** **Administrators query order information**

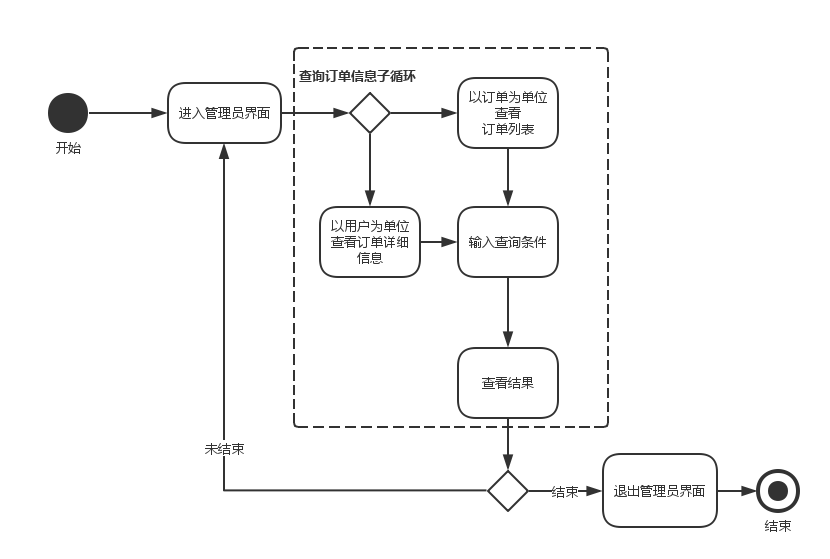


图2.12 Activity Diagram of Administrators Querying Order Information

The administrator can query the sales status through the order and the user as the reference unit respectively. The implementation of this module function involves the design of a specific database, which will be discussed in detail later.

### 2.1.4 Data Dictionary Design

The data dictionary of the user information table designed is as shown in Table 2.1.

Table 2.1 User Information Table Data Dictionary Design

|  |  |
| --- | --- |
| Item | Description |
| Table Name | Users |
| Brief Description | Basic information of users |
| Data storage composition | [User ID + Name + Password + Confirm Password + Email + Phone] |
| Associated processing | Query user information, permission management |

The data dictionary of the product information table designed is as shown in Table 2.1.

Table 2.2 Product Information Table Data Dictionary Design

|  |  |
| --- | --- |
| Item | Description |
| Table Name | Products |
| Brief Description | Basic information of products |
| Data storage composition | [Product number + product name + product introduction + category + price + image + image type] |
| Associated processing | Search products, shopping |

The data dictionary of the order information table designed is as shown in Table 2.3.

Table 2.3 Order Information Table Data Dictionary Design

|  |  |
| --- | --- |
| Item | Description |
| Table Name | Orders |
| Brief Description | Basic information of users and products |
| Data storage composition | [Order number + user number + shipping address + total price] |
| Associated processing | Inquiry orders, order processing |

The data dictionary of the order information details table designed is as shown in Table 2.4.

Table 2.4 Order Information Details Table Data Dictionary Design

|  |  |
| --- | --- |
| Item | Description |
| Table Name | OrderItems |
| Brief Description | Basic information of the relationship between orders and products |
| Data storage composition | [Order number + product number + quantity] |
| Associated processing | Inquiry orders, order processing |

## 2.2 System Design

### 2.2.1 Conceptual structure design

#### 1，Entity abstraction

According to the previous analysis of use cases, several actors entities can be abstracted from it, namely: user, product, and order.

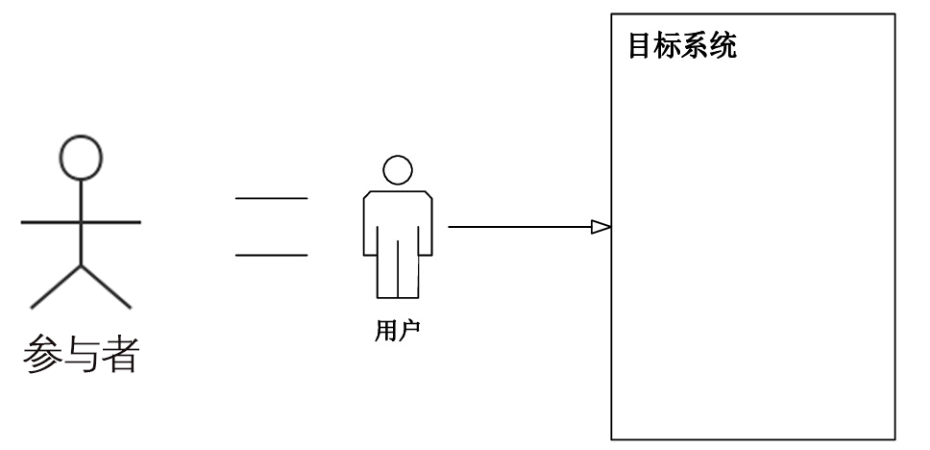


Figure 2.13 Participants and users

A participant is someone or something outside of the system that interacts with the system and is a key point in the conceptual model building phase. The user entity describes the customer in the real world as a participant, which contains the customer's key identity information, which should have key values ​​that can identify different customers. Participants are initiators of each business in the system. Therefore, the business design of the system should be based on the participants, that is, “people-oriented”.

In an e-commerce management system, goods are undoubtedly an indispensable entity, and their specific attributes need to be determined according to the needs of the real world. For example, in this example, in addition to having key values ​​that can identify different products. It should also have product category information, etc., to assist in the classified query, which will be further elaborated in the physical model design.

Another key entity is the order. Orders are key entities that connect users with products. One problem I encountered here is that whether an order should be an entity or a link? On the surface, the order actually represents the purchase relationship between the user and the product. In fact, it can be abstracted as a system use case, that is, a complete business. From this perspective, buying can be used as a link to replace the order entity.

However, the specific analysis of specific issues, for a product, the customer may have multiple purchases, and each purchase will contain different goods, so it is foreseeable that if the purchase is a relationship, then it lose this level of expression, and the database will encounter serious redundancy and information loss. Therefore, in the system of this project, I abstracted the entity of the order separately and added order details between the order and the goods as the link between the two to avoid possible redundancy.

#### 2，E-R diagram

From the E-R graph, it can be seen that order and product are related by the relationship of Order Items. An order can be related to one to several products, and a product can appear 0 to n times in Order Items.

Therefore, according to the above analysis, the entities and their attribute maps involved in this system include the following aspects.

First, user and its attributes, the id is the user identity, as shown in Figure 2.14.

**用户**

Figure 2.14 User Identity and Its Properties

Second, product and its properties，the id is the product identity，as shown in Figure 2.15.

**商品**

Figure 2.15 Product Identity and Its Properties

Third, order and its properties，the id is the order identity，as shown in Figure 2.16.

**订单**

Figure 2.16 Order Identity and Its Properties

According to the demand analysis, the entity relationship (E-R diagram) can be obtained as follows:

**用户**

**创建**

**订单**

**订单明细**

**商品**

Figure 2.17 E-R Diagram of Outdoor Sports E-commerce Management System

### 2.2.2 Logical structure design

According to the E-R diagram in the conceptual structure design, through the analysis of the attributes and relationships between entities, we transform it from the conceptual model to the relational model, and optimize according to the paradigmatic theory.

1. **Users** (**UserID**, Name, Password, ConfirmPassword, Email, PhoneNumber)
2. **Products** (**ProductID**, Name, Description, Category, Price, ImageData, ImageMimeType)
3. **Orders** (**OrderID**, UserID, Address, TotalPrice)
4. **OrderItems** (**OrderID**, **ProductID**, Quantity)

Explanation of the meaning of each table:

**Users**

|  |  |
| --- | --- |
| **Attribute** | **Meaning** |
| **UserID（Primary Key）** | User ID |
| Name | User Name |
| Password | Password |
| ConfirmPassword | Confirm Password |
| Email | Email |
| PhoneNumber | Phone Number |

**Products**

|  |  |
| --- | --- |
| **Attribute** | **Meaning** |
| **ProductID（Primary Key）** | Product ID |
| Name | Product Name |
| Description | Brief Description of Products |
| Category | Product Category |
| Price | Unit Price |
| ImageData | Image data, turned to bits in database |
| ImageMimeType | Type of image |

**Orders**

|  |  |
| --- | --- |
| **Attribute** | **Meaning** |
| **OrderID（Primary Key）** | Order ID |
| UserID（Foreign Key） | Purchaser’s ID，references Users（UserID） |
| Address | Address |
| TotalPrice | Total Price |

**OrderItems**

|  |  |
| --- | --- |
| **Attribute** | **Meaning** |
| **OrderID（Primary Key）** | Order Number，references Orders（OrderID） |
| **ProductID（Primary Key）** | Product ID，references Products（ProductID） |
| Quantity | Quantity |

There are several places worth considering: The first one is the Users table, which adds a ConfirmPassword attribute, in order to increase the reliability of the user registration; followed by the Products table, which also added the ImageData and ImageMimeType attributes, which represent the product's image and the type of picture, this is done to provide users with more abundant content when the user chooses products; then the Orders table, the Address attribute is placed in the Orders table, rather than the Users table, because the user’s delivery address each time may not be the same.

### 2.2.3 Paradigm verification

The function dependencies that exist in the above database schema are：

**UserID** → Name, Password, ConfirmPassword, Email, PhoneNumber

**ProductID** → Name, Description, Category, Price, ImageData, ImageMimeType

**OrderID** → UserID, Address, TotalPrice

**OrderID, ProductID** → Quantity

The left-hand side of each dependency is the superkey of the corresponding relationship, so these four relationships belong to BCNF.

### 2.2.4 Physical design

Database storage structure:

Through the demand analysis, the outline design and the logic design process we get the database structure of this system.

Database table design:

This chapter further defines the type and length of each data item in the designed relational schema in the logic design, converts each relation into a two-dimensional table in the database, and determines the primary key and foreign key of each table through MS Visual Studio. The SQL LocalDB operation gets the following table structure:

Table 2.5 describes the attribute table of user information

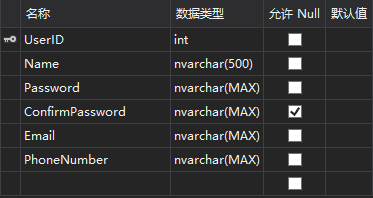
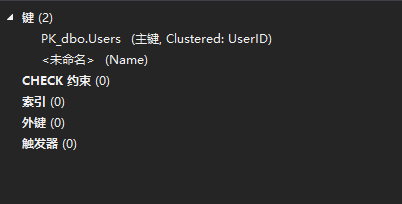
 

Table 2.5 Attribute Table of User Information

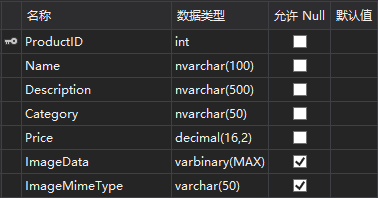
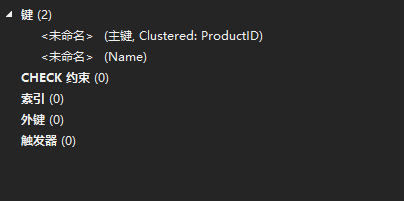
Table 2.6 describes the attribute table of product information 

Table 2.6 Attribute Table of Product Information

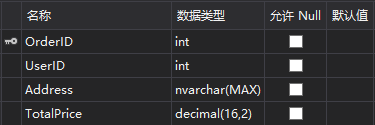
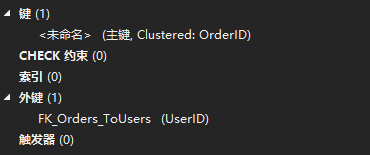
Table 2.7 describes the attribute table of order information  

Table 2.7 Attribute Table of Order Information

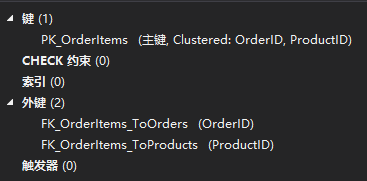
Table 2.8 describes the attribute table of order items information  

Table 2.8 Attribute Table of Order Items Information

# 3 System Implementation

## 3.1 The implementation of the database

### 3.1.1 Database creation

According to the analysis of the system, the system database mainly includes four tables, in which the sql statements created by the database and the table are as follows:

Database SportsStore creation statement

CREATE DATABASE SportsStore;

Users, Products, Orders, OrderItems creation statements:

CREATE TABLE [dbo].[Users] (

[UserID] INT IDENTITY (1, 1) NOT NULL,

[Name] NVARCHAR (500) NOT NULL,

[Password] NVARCHAR (MAX) NOT NULL,

[ConfirmPassword] NVARCHAR (MAX) NULL,

[Email] NVARCHAR (MAX) NOT NULL,

[PhoneNumber] NVARCHAR (MAX) NOT NULL,

CONSTRAINT [PK\_dbo.Users] PRIMARY KEY CLUSTERED ([UserID] ASC),

UNIQUE NONCLUSTERED ([Name] ASC)

);

CREATE TABLE [dbo].[Products] (

[ProductID] INT IDENTITY (1, 1) NOT NULL,

[Name] NVARCHAR (100) NOT NULL,

[Description] NVARCHAR (500) NOT NULL,

[Category] NVARCHAR (50) NOT NULL,

[Price] DECIMAL (16, 2) NOT NULL,

[ImageData] VARBINARY (MAX) NULL,

[ImageMimeType] VARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([ProductID] ASC),

UNIQUE NONCLUSTERED ([Name] ASC)

);

CREATE TABLE [dbo].[Orders] (

[OrderID] INT IDENTITY (1, 1) NOT NULL,

[UserID] INT NOT NULL,

[Address] NVARCHAR (MAX) NOT NULL,

[TotalPrice] DECIMAL (16, 2) NOT NULL,

PRIMARY KEY CLUSTERED ([OrderID] ASC),

CONSTRAINT [FK\_Orders\_ToUsers] FOREIGN KEY (UserID) REFERENCES [Users]([UserID])

);

CREATE TABLE [dbo].[OrderItems] (

[OrderID] INT NOT NULL,

[ProductID] INT NOT NULL,

[Quantity] INT NOT NULL,

CONSTRAINT [PK\_OrderItems] PRIMARY KEY CLUSTERED ([OrderID] ASC, [ProductID] ASC),

CONSTRAINT [FK\_OrderItems\_ToOrders] FOREIGN KEY ([OrderID]) REFERENCES [dbo].[Orders] ([OrderID]),

CONSTRAINT [FK\_OrderItems\_ToProducts] FOREIGN KEY ([ProductID]) REFERENCES [dbo].[Products] ([ProductID])

);

### 3.1.2 Data Samples

The following is a sample of data stored in the database at the time of completion of the project for use by test queries.

**Users**

Note: The first five are the names of my roommates and me.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UserID | Name | Password | ConfirmPassword | Email | PhoneNumber |
| 2 | 张剑富 | 123 | 123 | 11@a.com | 1111 |
| 3 | 张家豪 | 123 | 123 | 22@b.com | 2222 |
| 4 | 陈计 | 123 | 123 | 33@c.com | 3333 |
| 5 | 陈潜 | 123 | 123 | 44@d.com | 4444 |
| 6 | 谢佳梁 | 123 | 123 | 55@e.com | 5555 |
| 7 | 老司机 | 123 | 123 | 66@f.com | 6666 |
| 8 | 科比 | 123 | 123 | 77@g.com | 7777 |

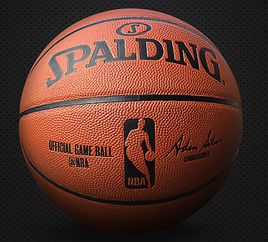
**Pruducts**

Note: Because ImageData is stored in Byte data, it has a larger space and is skipped in the table below.

ImageData example: ImageData for basketball is

0xFFD8FFE000104A46494600010101006000600000FFDB00430006040506050406060506070706080A100A0A09090A140E0F0C1017141818171416161A1D251F1A1B231C1616202C20232627292A29191F2D302D283025282928FFDB0043010707070A080A130A0A13281A161A2828282828282828282828282828282828282828

The corresponding picture is:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ProductID | Name | Description | Category | Price | ImageMimeType |
| 1 | 篮球 | NBA比赛指定用球 | 球类 | 139.00 | image/jpeg |
| 2 | 足球 | FIFA指定用球 | 球类 | 176.00 | image/jpeg |
| 3 | 排球 | 老少皆宜，活动量适中的球类运动 | 球类 | 156.00 | image/jpeg |
| 4 | 乒乓球 | 中国国球，十个一袋，你值得拥有！ | 球类 | 20.00 | image/jpeg |
| 5 | 网球 | 一项盛行全世界，优美而激烈的体育运动！8个一袋 | 球类 | 25.00 | image/jpeg |
| 7 | 羽毛球 | 国际羽协指定用球，六个一袋 | 球类 | 82.00 | image/jpeg |
| 8 | NBA球衣 | 我湖限量版纪念球衣！ | 服装类 | 750.00 | image/jpeg |
| 9 | T恤 | 时尚时尚最时尚的T-shirt！ | 服装类 | 49.00 | image/jpeg |
| 10 | 短裤 | 带给你夏日的一股清凉！ | 服装类 | 39.00 | image/jpeg |
| 11 | 背心 | 耍酷、装逼、秀肌肉必备用品！ | 服装类 | 68.00 | image/jpeg |
| 12 | 长运动裤 | 跑步、健身、出行必备！ | 服装类 | 189.00 | image/jpeg |
| 13 | 夹克 | 翻领，对襟，便于工作和活动 ，炫酷大方！ | 服装类 | 198.00 | image/jpeg |
| 14 | 登山鞋 | 专门为爬山和旅行而设计制造的鞋子，非常适合户外运动！ | 鞋类 | 219.00 | image/jpeg |
| 15 | 休闲鞋 | 运动鞋爆红的当下,不系带的休闲鞋也成为了时尚人士的必备单品 | 鞋类 | 179.00 | image/jpeg |
| 16 | 网球鞋 | 用力大，方向多变，要求耐冲击、稳定性佳、减震好、止滑性好 | 鞋类 | 368.00 | image/jpeg |
| 17 | 跑鞋 | 提供减震性、提供稳定性、提供运动控制，专业的，值得信赖！ | 鞋类 | 439.00 | image/jpeg |
| 18 | 足球鞋 | FIFA指定用鞋 | 鞋类 | 345.00 | image/jpeg |
| 19 | 篮球鞋 | NBA比赛官方指定战靴！ | 鞋类 | 539.00 | image/jpeg |
| 20 | 短袜 | 保护并美化亲的脚！ | 装备类 | 32.00 | image/jpeg |
| 21 | 背包 | 居家、运动、旅行必备用品！ | 装备类 | 128.00 | image/jpeg |
| 22 | 帽子 | 具有遮阳、装饰、增温和防护等作用 | 装备类 | 29.00 | image/jpeg |
| 23 | 运动手套 | 保护亲的手在运动过程中不受伤~ | 装备类 | 39.00 | image/jpeg |
| 24 | 帐篷 | 撑在地上遮蔽风雨﹑日光并供临时居住的棚子 | 装备类 | 229.00 | image/jpeg |
| 25 | 山地车 | 可以在公路外骑行的自行车 | 装备类 | 2299.00 | image/jpeg |
| 26 | 山地摩托 | 适合山路骑行的摩托车 | 装备类 | 6780.00 | image/jpeg |
| 27 | 氧气瓶 | 珠峰大本营专用 | 装备类 | 148.00 | image/jpeg |
| 28 | 智能手环 | 记录日常生活中的锻炼、睡眠、饮食等实时数据 | 装备类 | 79.00 | image/jpeg |
| 29 | 彩带 | 民间传统手工艺品 | 装饰类 | 7.00 | image/jpeg |
| 30 | 挂件 | 高僧开光，保您平安！ | 装饰类 | 82.00 | image/jpeg |
| 31 | 运动贴纸 | 让亲在运动的人群里更加拉风！ | 装饰类 | 12.00 | image/jpeg |

**Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| OrderID | UserID | Address | TotalPrice |
| 1 | 2 | 浙大紫金港青溪 | 139.00 |
| 3 | 3 | 浙大紫金港青溪 | 45.00 |
| 4 | 2 | 浙大紫金港丹阳 | 889.00 |
| 5 | 4 | 浙大紫金港蓝田 | 1980.00 |
| 6 | 5 | 浙大紫金港紫云 | 94.00 |
| 7 | 6 | 浙大紫金港碧峰 | 2528.00 |
| 8 | 7 | 浙大紫金港白沙 | 678.00 |
| 10 | 2 | 浙大玉泉七舍 | 6928.00 |
| 11 | 5 | 浙大之江 | 398.00 |

**OrderItems**

Note: The following order details correspond to the Orders table’s information

|  |  |  |
| --- | --- | --- |
| OrderID | ProductID | Quantity |
| 1 | 1 | 1 |
| 3 | 4 | 1 |
| 3 | 5 | 1 |
| 4 | 1 | 1 |
| 4 | 8 | 1 |
| 5 | 13 | 10 |
| 6 | 30 | 1 |
| 6 | 31 | 1 |
| 7 | 24 | 1 |
| 7 | 25 | 1 |
| 8 | 1 | 1 |
| 8 | 19 | 1 |
| 10 | 26 | 1 |
| 10 | 27 | 1 |
| 11 | 14 | 1 |
| 11 | 15 | 1 |

## 3.2 Module function implementation

### 3.2.1 Development Environment and Programming Technology

Development Environment: Windows 10 Home Edition (64-bit)

System memory size: 8.00 GB

Development Tools: Microsoft Visual Studio 2015 Community

Database Version: Microsoft SQL Server 2014 Express LocalDB

Development languages: C#, HTML, CSS, JavaScript

The programming language used in this experiment project is mainly C#. The interface design uses HTML, CSS, and some JavaScript technologies. The running project framework is Microsoft ASP.NET MVC5, so the project may appear to be rather large. The entire project was completed using the Microsoft Visual Studio 2015 Community.

In addition, the database still uses the SQL server database. For ease of debugging, the Entity Framework is used to integrate the database file into the project. The local database stored in Visual Studio is used to store data. There is no difference between this database and the SQL database.

### 3.2.2 System operation instructions

**（1）Operation instructions**

Because the system used in this experiment is based on Visual Sdutio as the development environment, and C# as the development language, so to successfully open the project, you need to refer to the book "3.2.1 development environment and programming technology" to configure the environment, and due to the adoption of .NET MVC framework, the project is relatively large, the process of opening and running may be relatively slow. In addition, because this system is a web-based program, the running process and major operations can be performed in the browser. The Firefox browser is recommended here. If other browsers are used, the interface may be presented in way that is different from this instruction. This is due to the different rendering effects of different browsers on the interface, which does not affect the normal use of the system.

(2) Functions completed

The expected functions have been achieved, as follows:

1, User registration, login function;

2, User classification query product features;

3, User's shopping cart records function;

4, User order creation;

5, Enable administrators to add, delete, change, check product information function;

6, Enable administrators to query user information function;

7, Enable administrators to query sales information function;

8, Fuzzy query and multi-condition query.

(3) Precautions

In the process of operation, there are still some defects, which have not yet been completely resolved, and users need to pay more attention to the operation.

1, enter the settlement page to return to the shopping cart

After entering the checkout page, if you click to view the shopping cart to go back to the shopping cart interface, and then click to continue shopping, you will not return to the shop home page but will return to the checkout page. This is because the system recorded the page before the shopping cart when performing this operation. The previous interface, when clicked to continue shopping, is to return to the previous interface instead of returning to the shopping interface.

Originally, I considered that the user had already turned several pages while shopping, so if a fixed interface is returned, the user needs to turn several pages again to return to the previous interface, which affects the user experience.

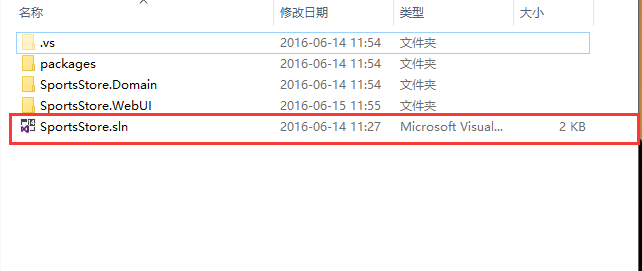
Therefore, this function has both advantages and disadvantages. The user should try to avoid returning to the shopping cart interface after entering the checkout page.

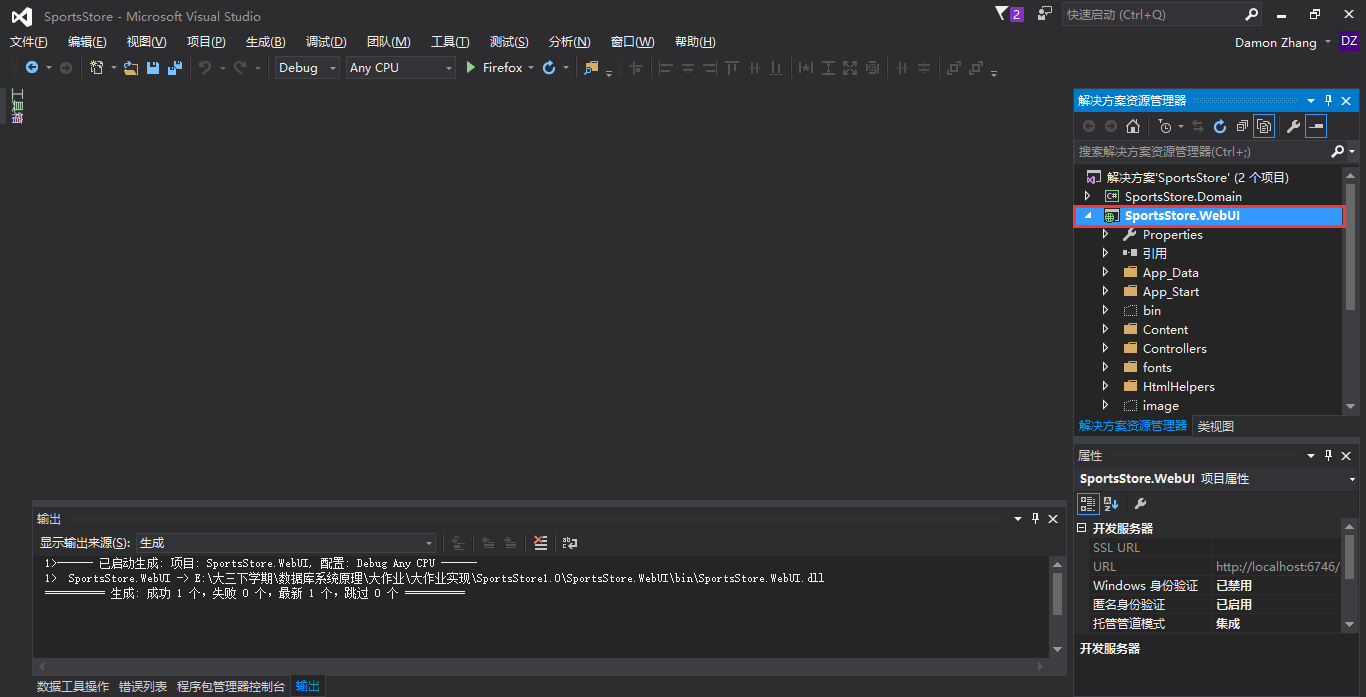
# 4 Installation and Operation Instructions

## 4.1 Environment

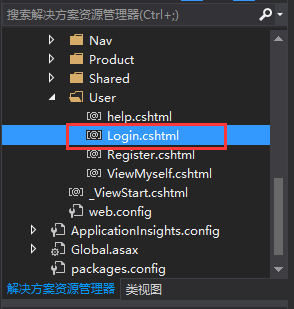
* Microsoft Visual Studio 2015
* Windows10
* Microsoft SQL Server 2014
* Microsoft SQL Server 2014 Express LocalDB
* Firefox

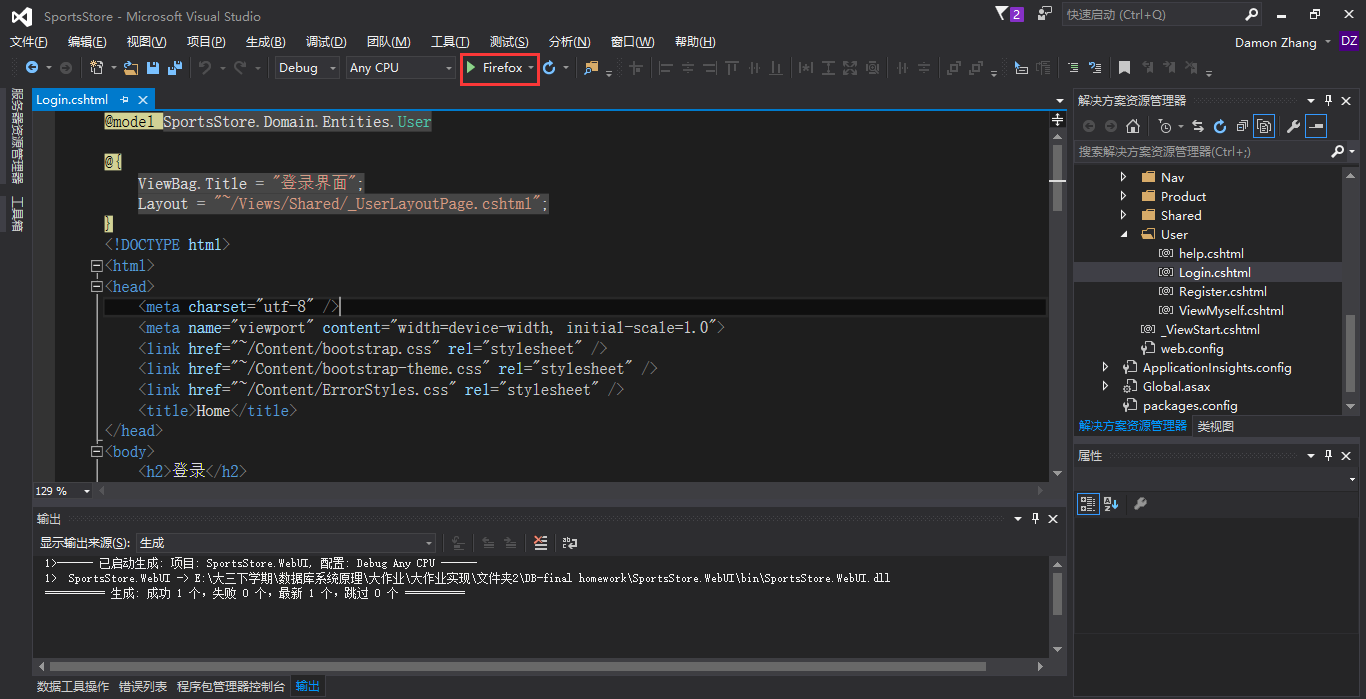
## 4.2 Enter the System

Open the SportsStore.sln file in a folder with Microsoft Visual Studio 2015

Click on the SportsStore.WebUI folder in Solution Explorer on the right side of the page

Find the "Login.cshtml" file in the Views/User directory and double-click to open



Press Ctrl+F5 to run the program and enter the login interface. Please make sure the selected browser is Firefox.



## 4.3 Register and Log in

Click "Register Now" in the title bar, enter the registration interface, fill in the user information





Click "Register", if the registration is successful, automatically jump to the login interface, of course, you can always return to the login interface



Enter the login screen, enter the user information you just registered, click Login, if the login is successful, you will enter the shop homepage



Go to the homepage of the store and note that there is authentication information at the bottom of the page, indicating that you have entered the login status. You can also log out at any time and log in again:



## 4.4 Purchase Products

Browse the products in the store, you can click the category on the left side of the screen to filter the products, or click the page number to turn pages.

Click "Buy Buy Buy!" to add the item to the shopping cart, jump to the shopping cart interface, you can see the product information in the shopping cart, and at the same time, there is an overview of the current status of the shopping cart in the upper left corner, showing the quantity of products and the total price



You can delete items in your shopping cart at any time, or you can choose to continue shopping



Choose to continue shopping and add another item



If you have finished the selection, you can click to settle and enter the order filling interface



Since I can't yet create an order automatically, the necessary information needs to be filled in by the user. Of course, this is not practical in reality, but I have met the technical bottleneck so that I am not been able to solve this problem when doing this project. I came up with this compromise method to create an order.



If the user forgets his ID number, he can find his ID number in “Check ID”. Of course, this function is also unreasonable. In practice, a user should not be allowed to see the information of other users.



Enter your own username (such as "Kobe"), you can find your UserID ("8")

After completing the order information, you can complete the order creation. Click Submit order and you will be prompted to create a success message.





## 4.5 Help

If the customer encounters any problems during the shopping process, he can also enter the help interface to contact the administrator (the contact information is fake)



## The user function is as described above, next let’s talk about the administrator function description

## 4.6 Administrator login

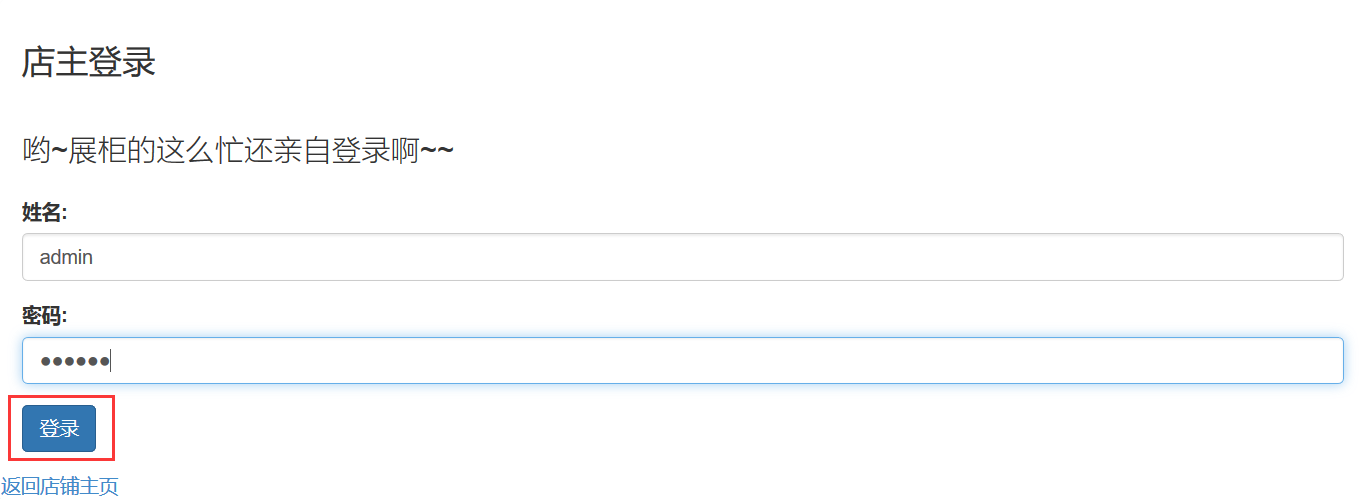
Click Logout to return to the login screen



Click on the administrator login to enter the administrator login interface



Enter the administrator account password (admin, secret), click login, enter the administrator backstage



## 4.7 Manage product information

The product list is displayed first, and the administrator can specify the search conditions according to their own needs and query the product information.



For example, if you want to search for a clothing item whose price is in the range of (50,500), you can set it as follows, and then click on search.

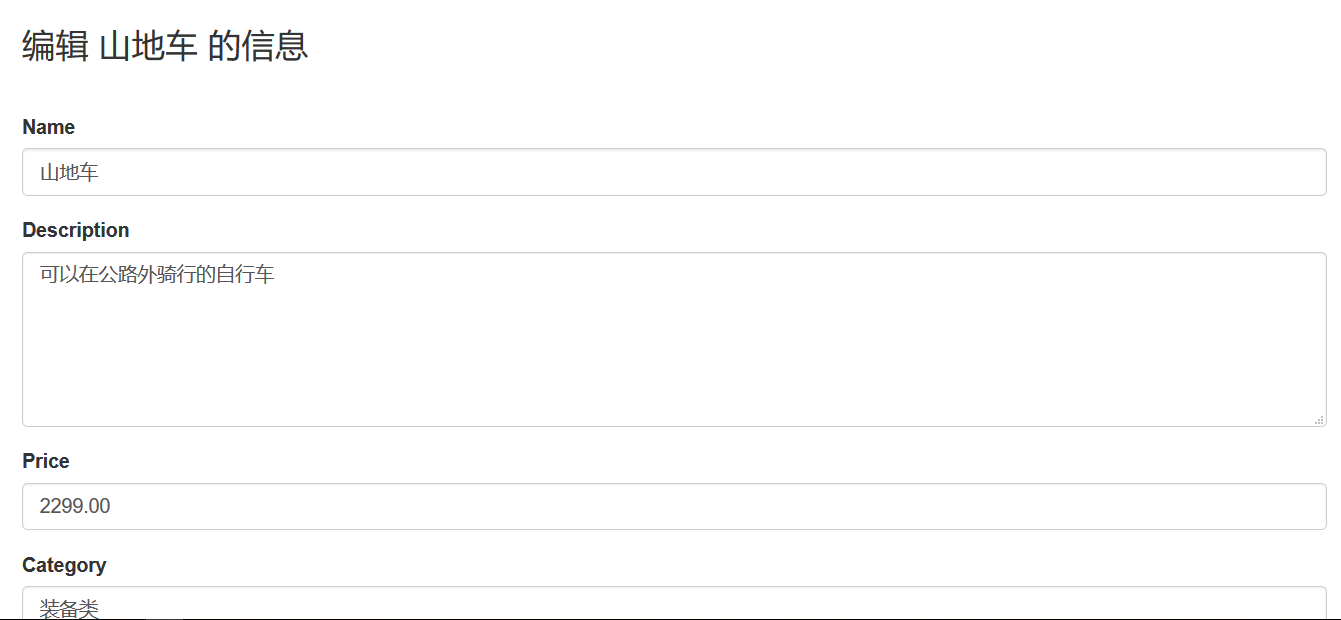
You can also perform fuzzy queries. For example, to search for equipment items whose names contain the word "地", you can search as follows.



Click on the "Delete" button in this interface to remove a piece of product information from the database. In order not to destroy the original data, there is no demonstration here, you may try yourself.

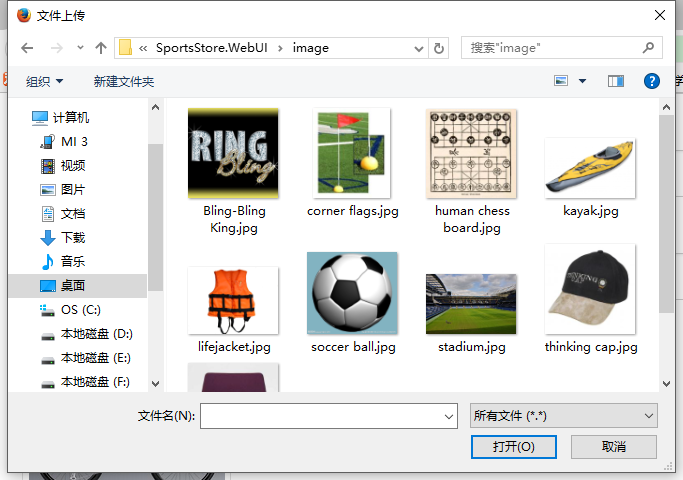


Click on the product name, you can enter the product information editing page, modify the product information, the information in the database will be updated accordingly

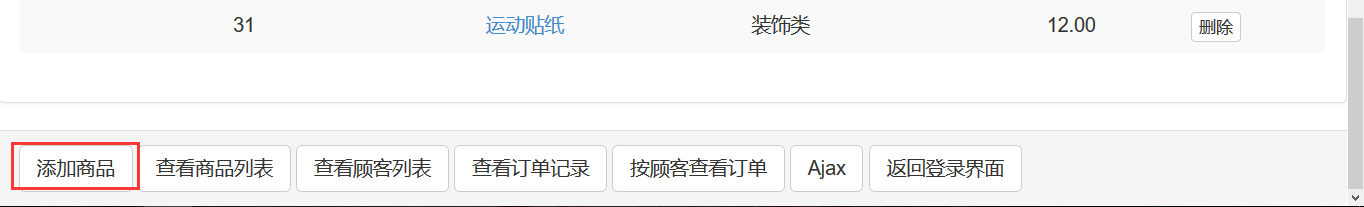
 

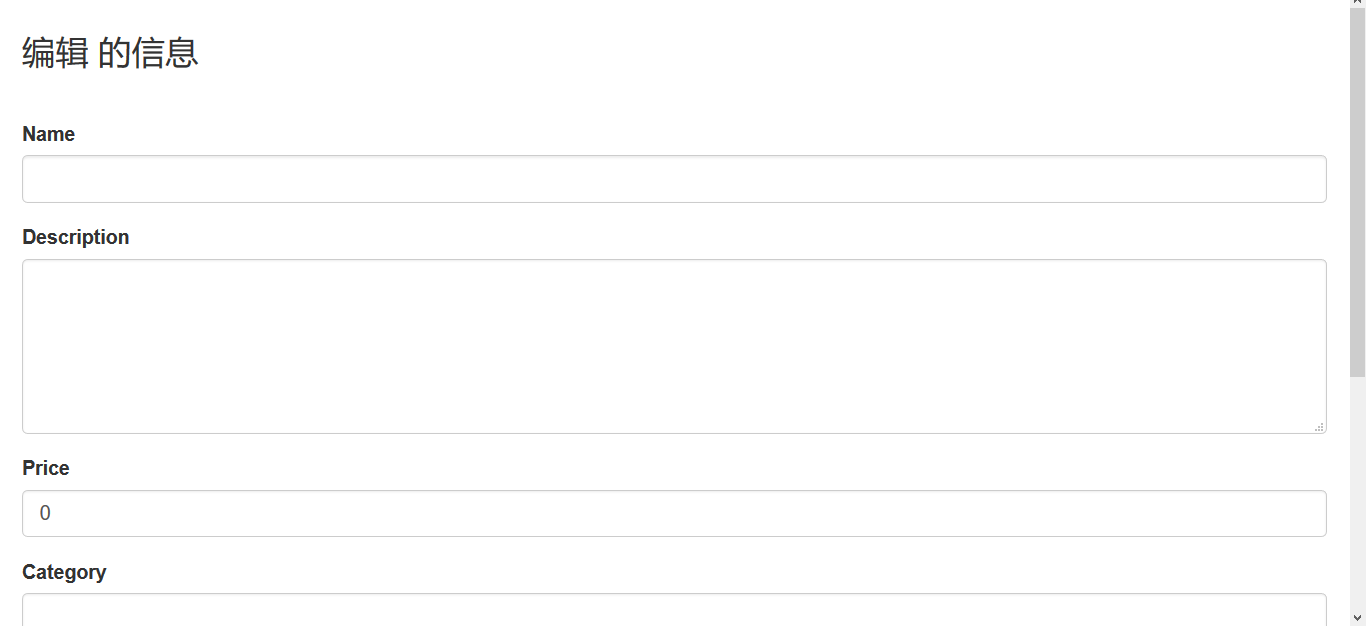


Click "Select Picture File" to replace the current picture with the local picture file

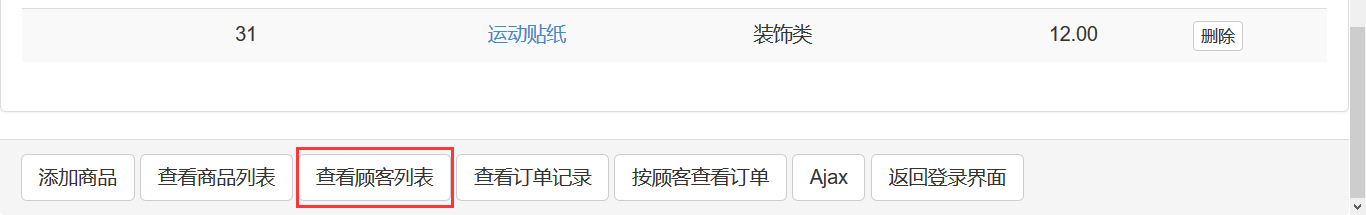
If you want to add an item, you can click on the Add Item Information page and you will also enter the product information edit page, but the form will be empty

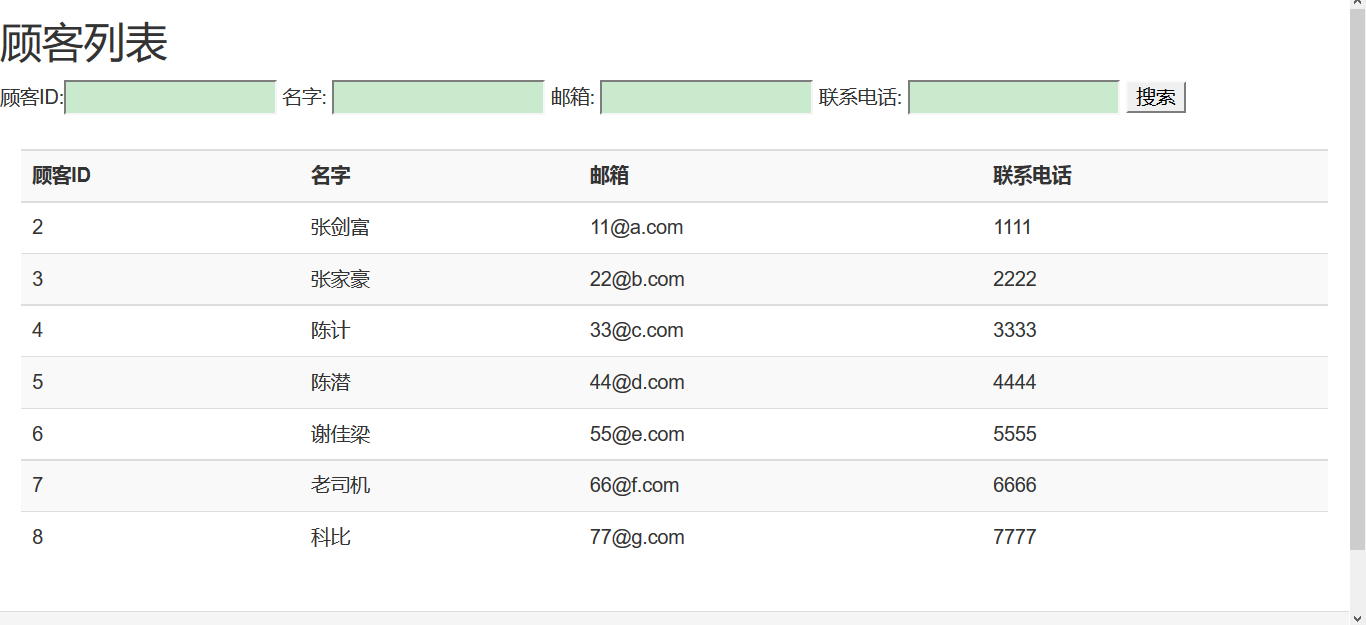




## 4.8 Query Customer Information

Click to view the list of customers, you can query the customer's information, here are the same fuzzy query





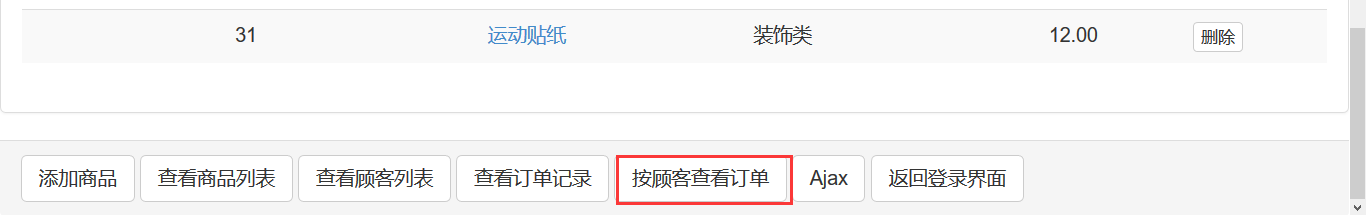
## 4.9 Query order information

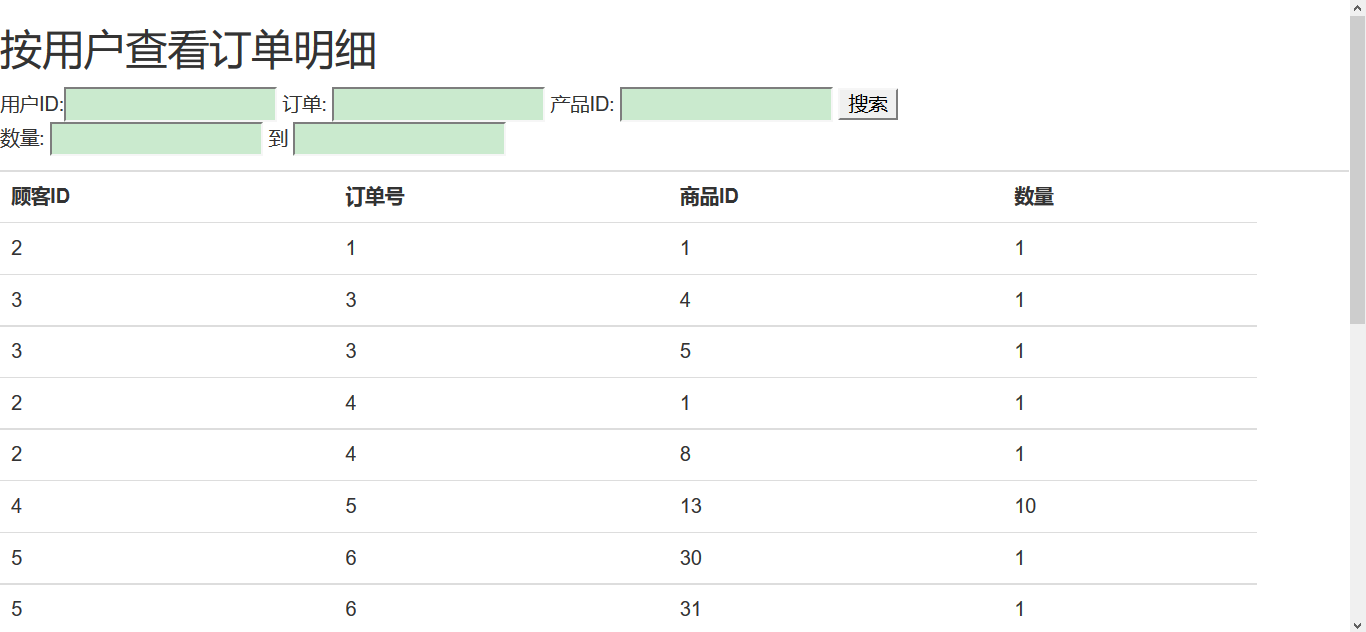
Click "View order records", you can view the information in the Orders table, fuzzy query and multi-condition query in similar manner



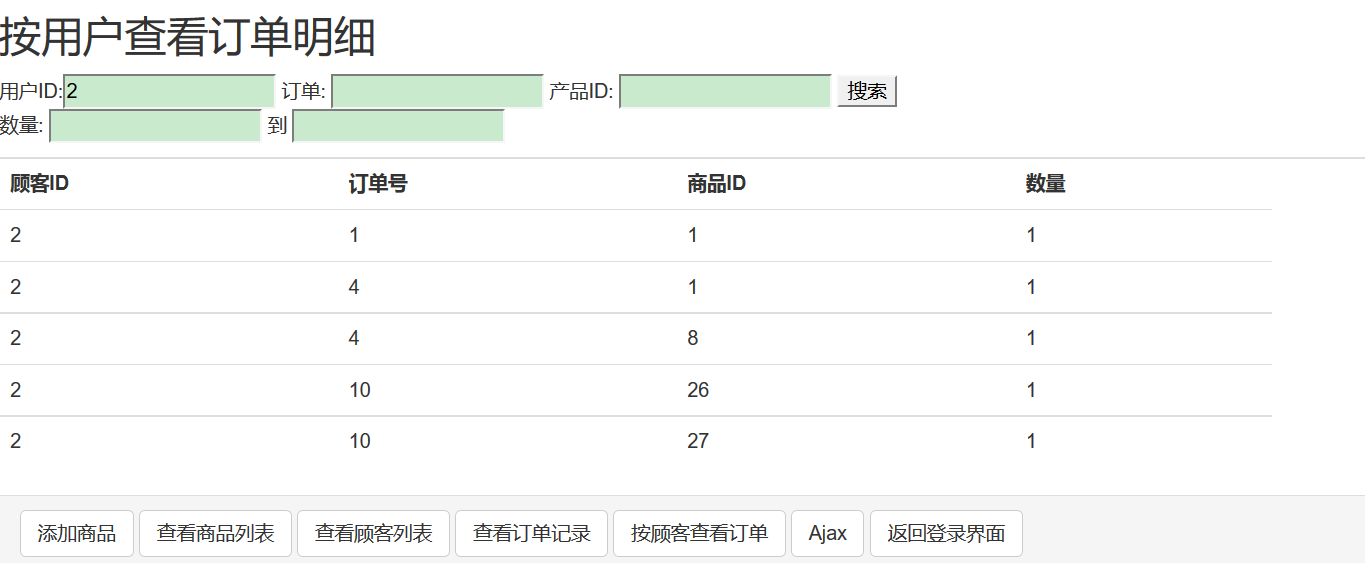


Click "View order by customer", you can query the joint information of Orders table and OrderItems table. You can also use multi-condition query and fuzzy query.



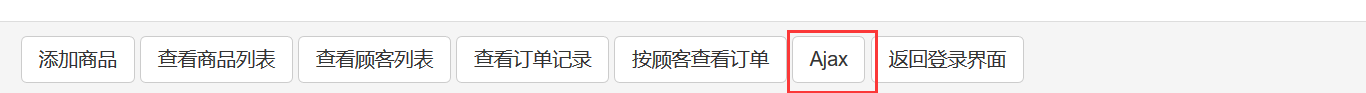


For example, to search for orders from customer number 2, you search as follows



## 4.10 Ajax

Click Ajax, you can see a different form submission method, this way to query information only need to refresh part of the interface, without reloading the entire interface, it will increase the interface refresh speed, for which I set a static Employee dataset that is not stored in the database





To search for common employees, you can select and search in the drop-down menu



# 5 Web structure

## 5.1 List of web pages

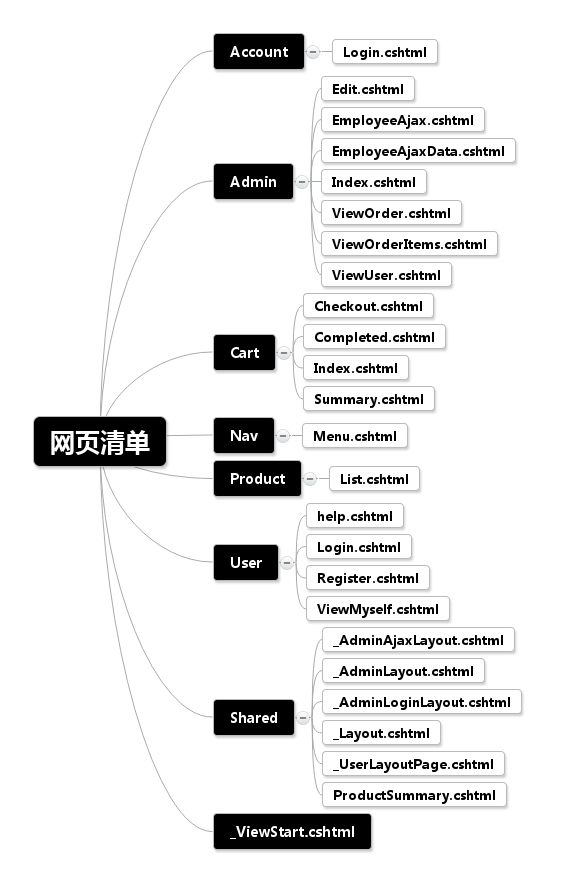


Figure 5.1 List of Web Pages

## 5.2 Web Page Functional Logic and Relationships

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Folder | Name | Functions | Layout or partialview | Pages that are linked to other pages |
| Account | Login | Administrator login screen | \_AdminLoginLayout | User/Login |
| Admin | Edit | Administrator edit product information interface | \_AdminLayout | Every screen under Admin |
| EmployeeAjax | Ajax form query employee view | \_AdminLayout, EmployeeAjaxData | Every screen under Admin |
| EmployeeAjaxData | Ajax form query employee partial view |  |  |
| Index | Admin Home / View Product List Interface | \_AdminLayout | Every screen under Admin |
| ViewOrder | Check order record interface | \_AdminLayout | Every screen under Admin |
| ViewOrderItems | View orders by customer interface | \_AdminLayout | Every screen under Admin |
| ViewUser | View customer list interface | \_AdminLayout | Every screen under Admin |
| Cart | Checkout | Fill in the order interface | \_Layout | Every screen under Cart and User |
| Completed | Order completion interface | \_Layout | Every screen under Cart and User except Checkout |
| Index | Shopping cart interface | \_Layout | Every screen under Cart and User |
| Summary | Shopping cart overview section view |  |  |
| Nav | Menu | The left side of the shop home page tab bar (partial view) |  |  |
| Product | List | Shop Home | \_Layout | Every screen under Cart and User |
| User | help | Help page | \_Layout | Every screen under Cart and User |
| Login | User login interface | \_Layout | Every screen under Cart and User |
| Register | User registration interface | \_Layout | Every screen under Cart and User |
| ViewMyself | Check the user ID interface | \_Layout | Every screen under Cart and User |
| Shared | \_AdminLayout | Administrator's layout interface |  |  |
| \_AdminLoginLayout | Administrator login layout interface |  |  |
| \_Layout | The default layout interface | Summary, Nav/Menu |  |
| \_UserLayoutPage | User's layout interface |  |  |
| ProductSummary | Store Home Products Section View |  |  |
| \_ViewStart |  | Used to specify the default view |  |  |

Table 5.2 Web Page Functional Logic and Relationships

# 6 Summary

In this project, I began with the demand analysis, and through systematic analysis, the design analysis until the establishment of the database and the employment of the system. I have completely experienced the development process of a Web application system.

The initial demand analysis and the system analysis and design are indispensable parts. In the actual development, developers often need to communicate with customers and understand their demand. In this way, a design can be described clearly. The quality of system analysis and design determines whether the development process can be carried out smoothly. This is the reason why system analysis and design work account for a large proportion of the time.

In the system implementation stage, I designed a simple e-commerce transaction management system. Among them, the database involves 3 entities: users, orders and commodities, and draws I designed an E-R diagram according to actual demand. I also designed conceptual structure, logical structure and physical storage mode. Then, around this core database, I completed the system running program. I make this system's interface more friendly and easy to use by continuous testing. In addition, this system has good scalability. Based on the existing database system, it can also continuously expand the relationships, and then correspondingly add the system's functional modules to meet a wider range of needs.

Of course, due to the lack of time, there are still some defects in the final version, but they do not affect the normal use of the system.

# References

[1] 马骏. ASP.NET MVC程序设计教程(第三版). 北京人民邮电出版社，2015

[2] 王珊, 萨师煊. 数据库系统概论(第四版). 北京:高等教育出版社, 2006: 198-225

[3] 亚当·弗里曼. 精通ASP.NET MVC5. 北京:人民邮电出版社, 2015

1. Drawn with Microsoft Visio 2016 [↑](#footnote-ref-1)
2. Drawn by the online drawing tool ProcessOn (https://www.processon.com/) [↑](#footnote-ref-2)
3. Figure 3 and the following activity diagrams are drawn through the online drawing tool ProcessOn [↑](#footnote-ref-3)