**《数据库原理》综合性实验报告**

**实验题目：在线电子商务系统**

# 

[**１引言** 3](#_Toc18660)

[**2 需求分析阶段** 3](#_Toc22465)

[**2.1需求分析阶段的目标** 3](#_Toc5672)

[**2.2需求分析阶段的任务** 3](#_Toc28009)

[2.2.1 处理对象： 3](#_Toc30197)

[2.2.2处理功能及要求： 4](#_Toc8276)

[**2.3 需求分析阶段成果** 4](#_Toc18527)

[2.3.1 在线电子商务系统业务流程图和数据流图 4](#_Toc16697)

[2.3.2 在线电子商务系统数据字典 12](#_Toc4202)

[**3 概念设计阶段** 15](#_Toc3981)

[3.1 概念设计阶段的任务和目标 15](#_Toc15920)

[3.2 E-R图 15](#_Toc6037)

[**4 逻辑设计阶段** 17](#_Toc24366)

[4.1 逻辑设计的任务和目标 17](#_Toc16888)

[4.2 数据组织 17](#_Toc3590)

[4.2.1 将E-R图转换为关系模型 17](#_Toc10503)

[4.2.2数据库模式定义 18](#_Toc5256)

[**5 实施阶段** 20](#_Toc3185)

[5.1 建立数据库、表 20](#_Toc30569)

[5.1.1 建立数据库 20](#_Toc31067)

[5.1.2 建立数据表 21](#_Toc14886)

[5.2数据入库 24](#_Toc13903)

[5.3数据库连接 25](#_Toc3815)

[**6 系统设计** 26](#_Toc17372)

[6.1 界面设计 26](#_Toc14066)

[6.2 功能模块设计 34](#_Toc32060)

[6.2.1 用户模块设计 34](#_Toc2923)

[6.2.2 商家模块设计 34](#_Toc16043)

[6.3 关键技术 34](#_Toc18610)

[6.3.1 事务处理 34](#_Toc18360)

[6.3.2 jsp技术 34](#_Toc6665)

[6.3.3 cookie和session机制 34](#_Toc222)

[6.3.4 Ajax技术 35](#_Toc3891)

[6.4 主要功能模块的实现 35](#_Toc18216)

[6.4.1 商品上架 35](#_Toc13342)

[6.4.2 删除购物车 37](#_Toc13740)

[**7 系统测试** 38](#_Toc736)

[7.1 功能测试 38](#_Toc25630)

[7.2 运行时间测试 39](#_Toc27271)

[7.3 数据库操作与安全测试 39](#_Toc2770)

[7.4 错误测试 39](#_Toc14738)

[7.5 测试结果 39](#_Toc16347)

[**8 个人心得** 42](#_Toc28944)

# **１引言**

电子商务是指整个商务活动和贸易活动的电子化，它通过先进的信息网络，将事物活动和贸易活动中发生关系的各方有机地联系起来。电子商务网站实际上就是销售企业为消费者提供的网上购物商城，在该网站中用户可以购买任何商品，而管理员可以对商品和订单等信息进行管理。

随着Internet的迅速崛起，互联网已逐渐成为查找收集信息的最佳方式，与此同时，互联网也逐步进入传统的商务领域，互联网的跨空间性及可交互性使其在与传统媒体行业及传统贸易行业竞争下凸显出不可抗拒的优势，因而互联网发展十分迅速。在电子商务在中国逐步兴起的大背景下，建立利用互联网来开拓信息来源，帮助企业及时掌握市场动向并及时的调整产品结构，协助经销商打开货源的信息门户网站成为解决信息流通不通畅的有效方案。毫无疑问，电子商务有利于企业转换经营机制，建立现代企业制度，提高企业的管理水平和国际竞争力。

按照数据库设计流程用业务流程图、数据流程图，数据字典详尽确切的表达电子商务系统的工作流程。进行概念模型设计，逻辑模型设计，物理设计，建立一个能满足用户和商家要求的方便实用的数据库系统。

# **2 需求分析阶段**

需求分析简单地说就是分析用户的要求。需求分析是设计数据库的起点，需求分析的结果是否准确地反映了用户的实际要求，将直接影响到后来各个阶段的设计，并影响到设计结果是否合理和使用。

## **2.1需求分析阶段的目标**

1.了解在线电子商务系统的基本内容；

2.了解在线电子商务系统的基本业务流程；

3.了解其他在线电子商务系统；

4.确定业务需求；

## **2.2需求分析阶段的任务**

### 2.2.1 处理对象：

系统的处理对象是顾客的登录注册，浏览商品，购买商品，信息管理，订单管理，购物车管理，评论管理和商家的登录注册，商品管理，订单管理，信息管理，评论管理，销售统计，评论管理。

系统的处理对象涉及以下几个方面的信息：

顾客基本信息（Customer）：顾客编号，信用卡编号，顾客姓名，顾客性别，顾客账号，顾客密码，顾客邮箱，顾客头像

信用卡（CreditCard）：信用卡编号，信用卡账号，信用卡密码，信用卡余额

收货地址（DeliveryAddress）：收货地址编号，顾客编号，收件人姓名，收件人电话，收货第一级地址，收货第二级地址，收货第三级地址，详细地址

商家（Shopper）：商家编号，商家账号，商家密码，商家邮箱，商家身份证，商家名称，商家头像

商店（Shop）：商店编号，商家编号，商店名称，商家介绍，开店日期，商店头像

商品（Product）：商品编号，商店编号，商品名称，商品介绍，商品价格，商品数量，商品种类，商品显示图片，商品参数，商品介绍图片，作者，出版社，上架时间

购物车（ShoppingCart）：顾客编号，商品编号，购物车商品数量

浏览历史（browserHistory）：顾客编号，商品编号，浏览时间

订单主表（OrderMaster）：订单编号，顾客编号，收货地址编号，订单日期，订单状态

订单明细（OrderDetail）：订单编号，商品编号，数量

评论（Comment）：商品编号，顾客编号，评论内容，评论时间，好评率

### 2.2.2处理功能及要求：

1. 能够方便进行存取数据，并方便对有效的经行数据的操作和管理，这主要包括：
2. 信息的录入，删除和修改
3. 信息的关键字检索查询

2.能够提供一定的安全机制，提供数据信息授权访问，修改和删除，防止随意查询，修改及删除。

## **2.3 需求分析阶段成果**

### 2.3.1 在线电子商务系统业务流程图和数据流图

#### 2.3.1.1 业务流程图

#### 2.3.2.2 数据流图

1. 顶层数据流图



1. 功能分解图

功能分解图（用户）

 功能分解图（商家）



1. 数据流图

用户注册



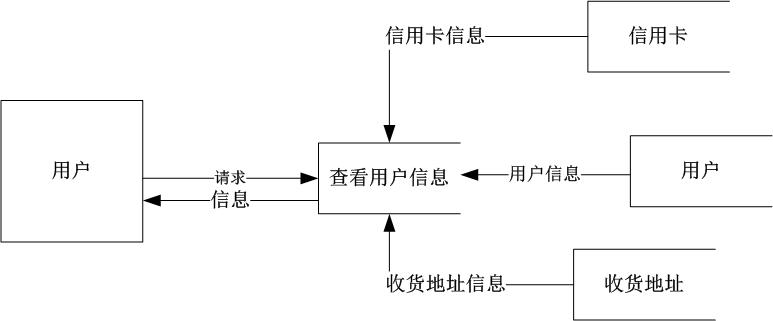
用户登录



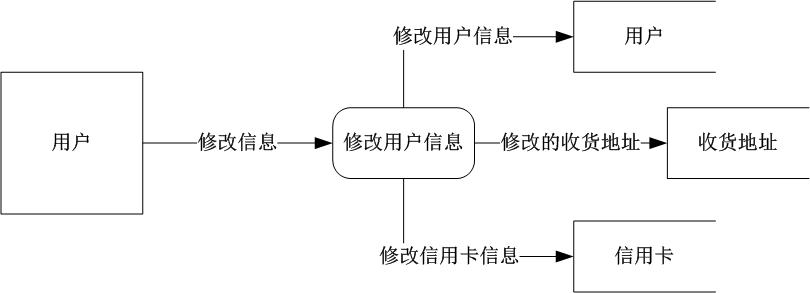
用户修改密码



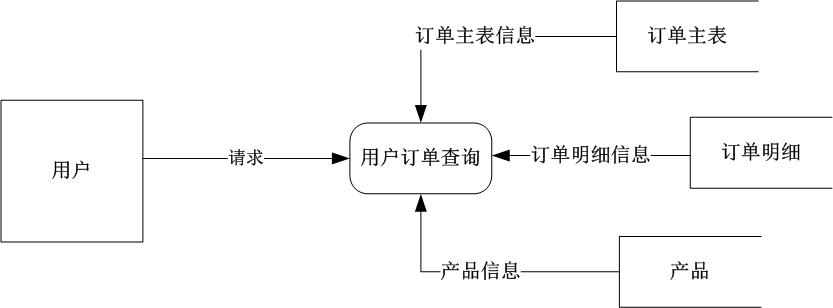
查看用户信息



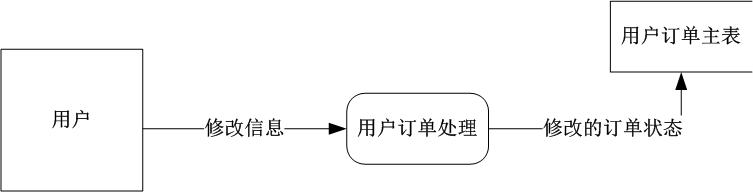
修改用户信息



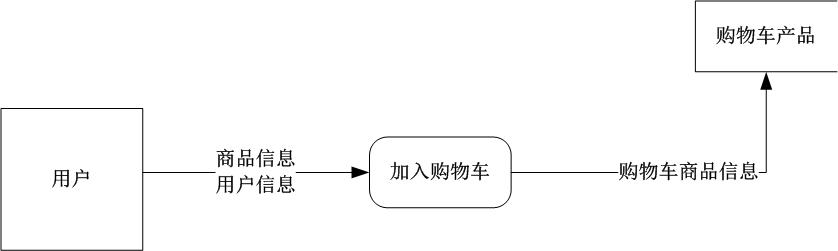
用户订单查询



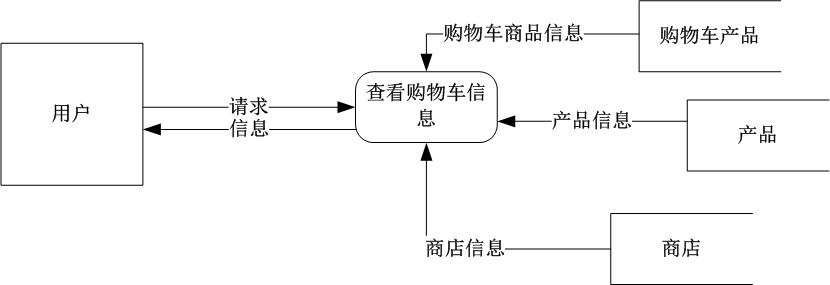
用户订单处理



加入购物车



查看购物车信息



修改购物车信息



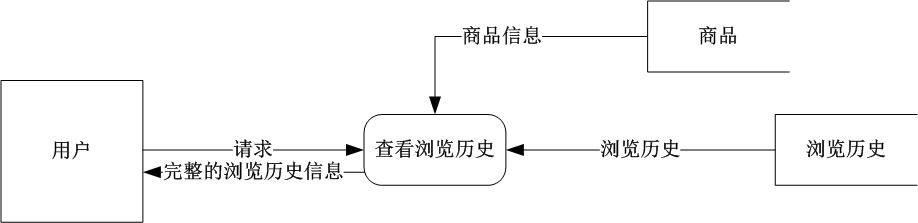
删除购物车商品



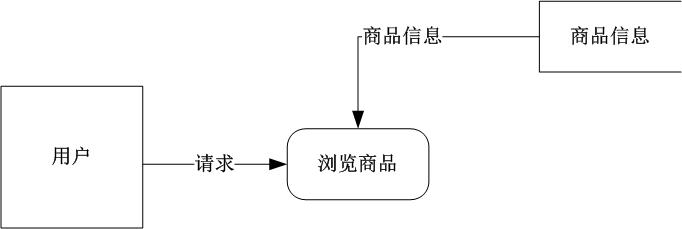
用户评论



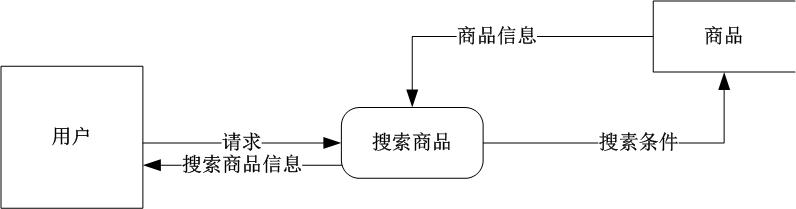
查看浏览历史



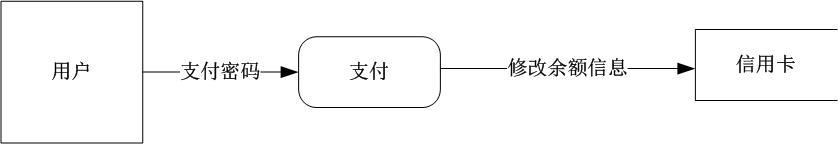
浏览商品



搜索商品



支付



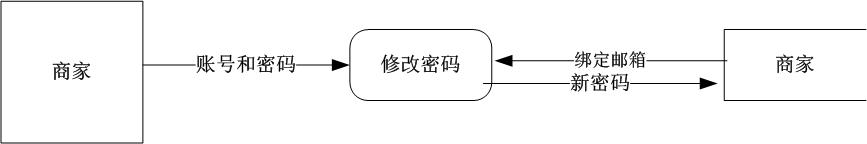
商家注册



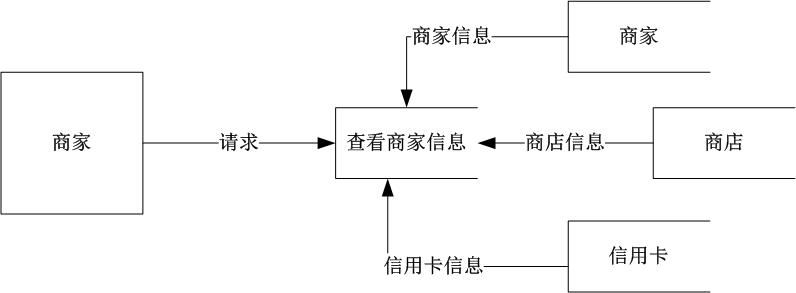
店铺注册



修改密码



查看商家信息



修改商家信息



商品上架



商品下架



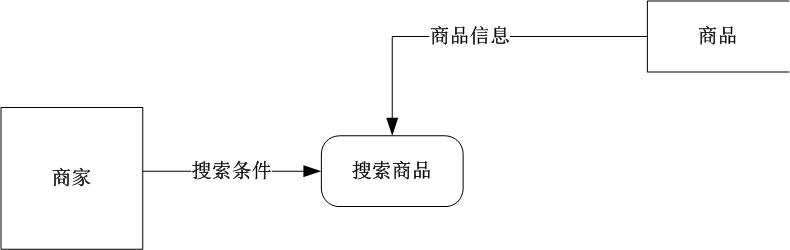
修改商品信息



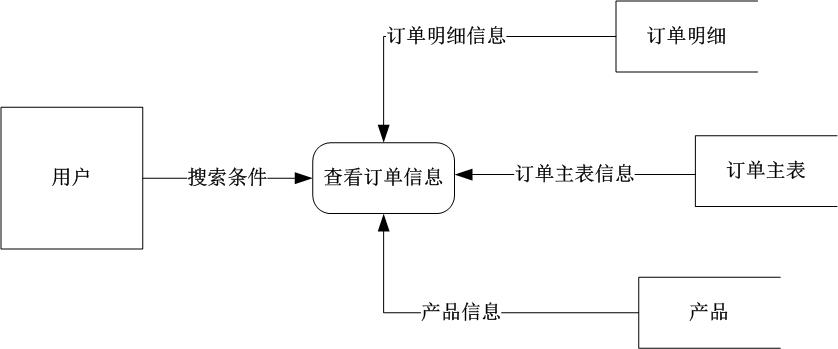
浏览商品详情



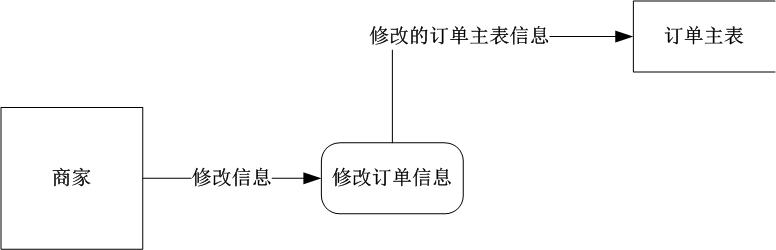
搜索商品



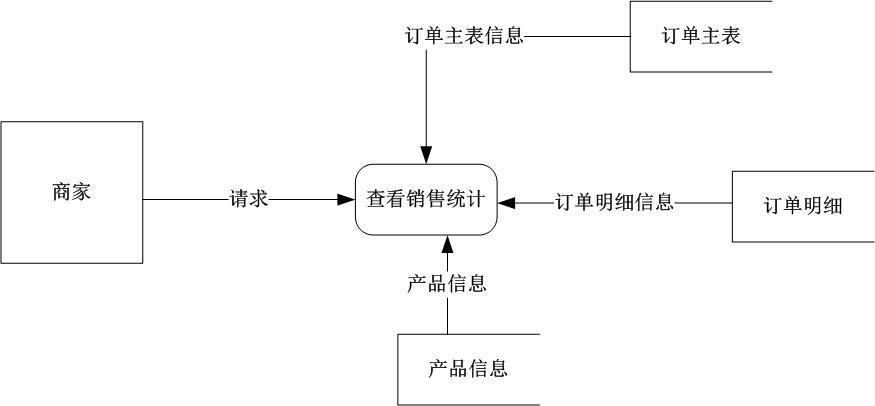
查看订单信息



修改订单信息



查看销售统计



### 2.3.2 在线电子商务系统数据字典

数据字典是系统中各类数据描述的集合，是进行详细的数据收集和数据分析所获得的主要成果。数据字典在数据库设计中占有很重要的地位。数据字典通常包括数据项、数据结构、数据流、数据存储、处理过程和外部实体6个部分。

数据项是不可再分的数据单位。

数据结构反映了数据之间的组合关系。

数据流是数据结构在系统内传输的路径。

数据存储是数据结构停留或保存的地方也是数据流的来源和去向之一。

处理过程的具体处理逻辑一般用判定表或判定树来描述。数据字典是需要描述处理过程的说明性信息。

**数据项**

| 数据相编号 | 数据项名称 | 数据项别名 | 数据项类型 | 长度 | 取之范围 | 备注 |
| --- | --- | --- | --- | --- | --- | --- |
| Dl-1 | customerId | 顾客编号 | INT | 11 |  |  |
| Dl-2 | creditCardId | 信用卡编号 | INT | 11 |  |  |
| Dl-3 | customerSex | 顾客性别 | enum |  | {‘男　’  ,’女’} |  |
| Dl-4 | customerAccount | 顾客账号 | VARCHAR | 14 |  |  |
| Dl-5 | customerPassword | 顾客密码 | VARCHAR | 14 |  |  |
| Dl-6 | customerMail | 顾客邮箱 | VARCHAR | 25 |  |  |
| Dl-7 | displayPicture | 顾客头像 | VARCHAR | 100 |  |  |
| Dl-8 | creditCardId | 信用卡编号 | INT | 11 |  |  |
| Dl-9 | creditAccount | 信用卡账号 | VARCHAR | 10 |  |  |
| Dl-10 | creditPassword | 信用卡密码 | VARCHAR | 10 |  |  |
| Dl-11 | money | 信用卡余额 | DOUBEL |  |  |  |
| Dl-12 | deliveryAddressId | 收货地址编号 | INT | 11 |  |  |
| Dl-13 | receiverName | 收件人姓名 | VARCHAR | 15 |  |  |
| Dl-14 | receiverTelephone | 收件人电话 | VARCHAR | 11 |  |  |
| Dl-15 | receiverAddressFirst | 收货第一级地址 | VARCHAR | 10 |  |  |
| Dl-17 | receiverAddressSecond | 收货第二级地址 | VARCHAR | 10 |  |  |
| Dl-18 | receiverAddressThird | 收货第三级地址 | VARCHAR | 10 |  |  |
| Dl-19 | receiverAddressDetail | 收货详细地址 | VARCHAR | 50 |  |  |
| Dl-20 | shopperId | 商家编号 | INT | 11 |  |  |
| Dl-21 | account | 商家账号 | VARCHAR | 14 |  |  |
| Dl-22 | password | 商家密码 | VARCHAR | 14 |  |  |
| Dl-23 | mail | 商家邮箱 | VARCHAR | 20 |  |  |
| Dl-24 | idCard | 商家身份证 | VARCHAR | 18 |  |  |
| Dl-25 | onlineName | 商家名称 | VARCHAR | 20 |  |  |
| Dl-26 | displayPicture | 商家头像 | VARCHAR | 100 |  |  |
| Dl-27 | shopId | 商店编号 | INT | 11 |  |  |
| Dl-28 | shopName | 商店名称 | VARCHAR | 40 |  |  |
| Dl-29 | introduction | 商店介绍 | VARCHAR | 200 |  |  |
| Dl-30 | openDate | 开店日期 | DATETIME |  |  |  |
| Dl-31 | displayPicture | 商店头像 | VARCHAR | 100 |  |  |
| Dl-32 | productId | 商品编号 | INT | 11 |  |  |
| Dl-33 | productName | 商品名称 | VARCHAR | 11 |  |  |
| Dl-34 | productIntroduction | 商品介绍 | VARCHAR | 200 |  |  |
| Dl-35 | productSale | 商品价格 | DOUBLE |  |  |  |
| Dl-36 | productStock | 商品数量 | INT | 11 |  |  |
| Dl-37 | productClass | 商品种类 | VARCHAR | 30 |  |  |
| Dl-38 | productShowPicture | 商品显示图片 | VARCHAR | 140 |  |  |
| Dl-39 | productAttribute | 商品参数 | VARCHAR | 400 |  |  |
| Dl-40 | productIntroductionPicture | 产品介绍图片 | VARCHAR | 160 |  |  |
| Dl-41 | writer | 作者 | VARCHAR | 40 |  |  |
| Dl-42 | publishingHouse | 出版社 | VRACHAR | 40 |  |  |
| Dl-43 | shelfTime | 上架时间 | DATETIME |  |  |  |
| Dl-44 | quantity | 数量 | INT | 11 |  |  |
| Dl-45 | BrowserTime | 浏览时间 | DATETIME |  |  |  |
| Dl-46 | orderDate | 订单日期 | DATETIME |  |  |  |
| Dl-47 | orderState | 订单状态 | VARCHAR | 20 |  |  |
| Dl-48 | commerntContent | 评论内容 | VARCHAR | 200 |  |  |
| Dl-49 | commentTime | 评论时间 | DATETIME |  |  |  |
| Dl-50 | goodRate | 好评率 | DOUBLE |  |  |  |
| Dl-51 | orderId | 订单编号 | INT | 11 |  |  |

**数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构编号 | 数据结构名 | 数据结构含义 | 数据结构组成 |
| DS-1 | customer | 顾客信息 | customerId,creditCardId,customerSex,  customerAccount,  customerPassword,customerMail,displayPicture |
| DS-2 | creditCard | 信用卡信息 | creditCardId，creditAccount,  creditPassword,money |
| DS-3 | deliveryAddress | 收货地址 | deliveryAddressId,receiverName,  receiverTelephone,receiverAddressFirst,  receiverAddressSecond,receiverAddressThird,  receiverAddressDetail |
| DS-4 | shopper | 商家 | shopperId,account,password,mail,idCard,  onlineName,displayPicture |
| DS-5 | shop | 商店 | shopId,shopName,introduction,openDate,  displayPicture |
| DS-6 | product | 商品 | productId,productName,productIntroduction,  productSale,productStock,productClass,  productShowPicture,productAttribute,  productIntroductionPicture,  writer,publishingHouse,shelfTime |
| DS-7 | ShoppingCart | 购物车 | customerId,productId,quantity |
| DS-8 | browserHistory | 浏览历史 | customerId,productId,BrowserTime |
| DS-9 | OrderMaster | 订单主表 | orderId,customerId,deliveryAddressId,  deliveryAddressId,orderDate,orderState |
| DS-10 | OrderDetail | 订单明细 | orderId,productId,quantity |
| DS-11 | Comment | 评论 | customerId,productId,commerntContent,  commentTime,goodRate |

# **3 概念设计阶段**

将需求分析得到的用户需求抽象为信息结构即概念模型的过程就是概念结构设计。它是整个数据库设计的关键。

## 3.1 概念设计阶段的任务和目标

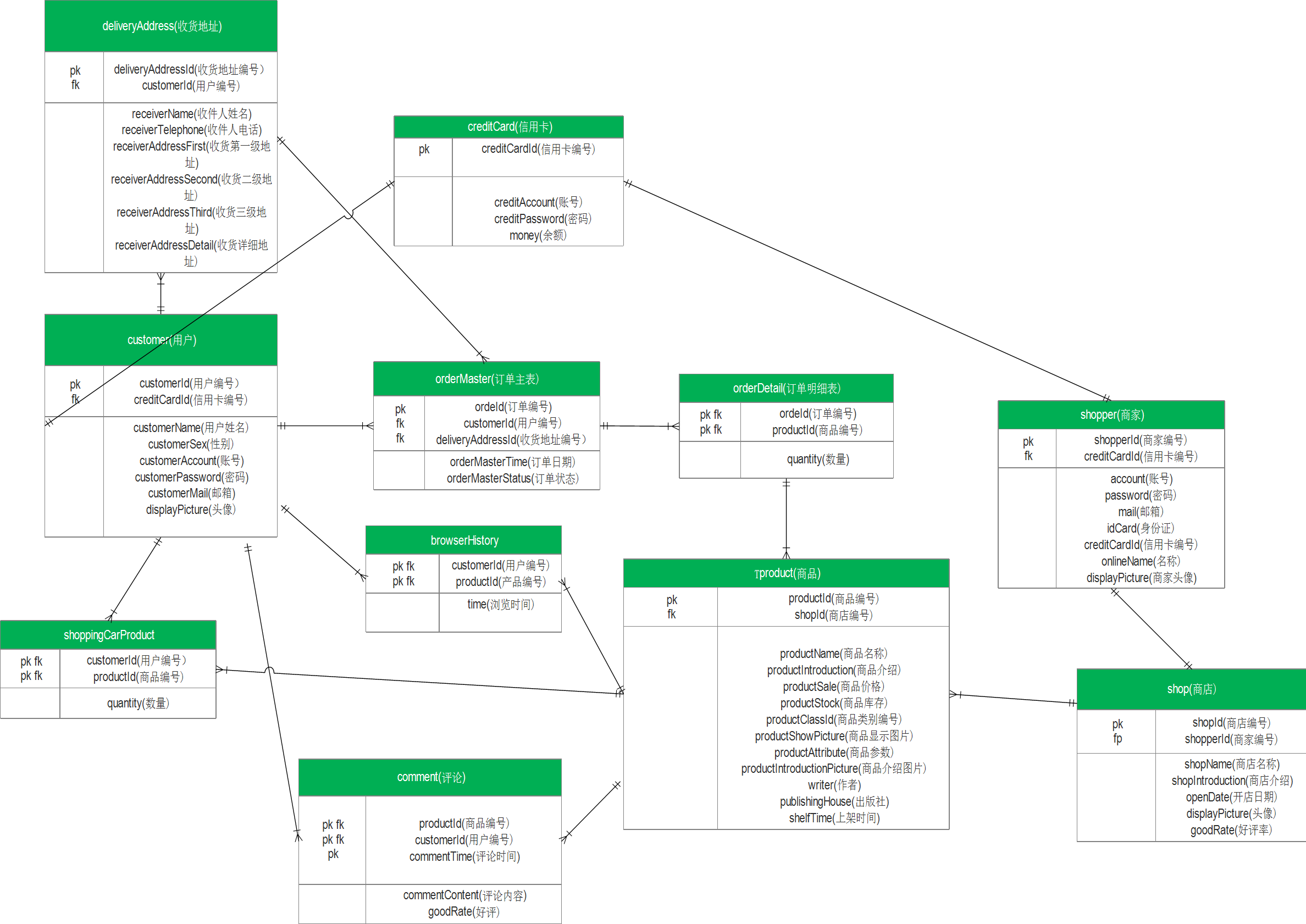
（1）选择底层数据流为切入点，通常选择实际系统中的子系统；

（2）设计分E-R图，即各子模块的E-R图；

（3）生成初步E-R图，通过合并方法，做到各子系统实体、属性、联系统一；

（4）生成全局E-R图，通过消除冲突等方面。

## 3.2 E-R图



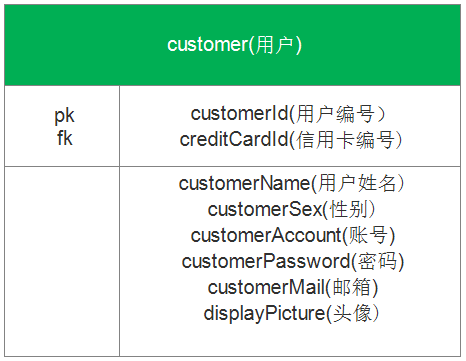
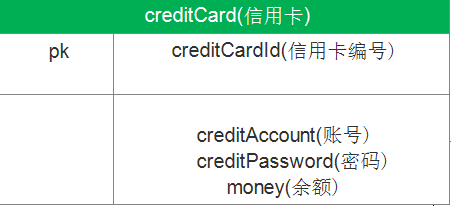
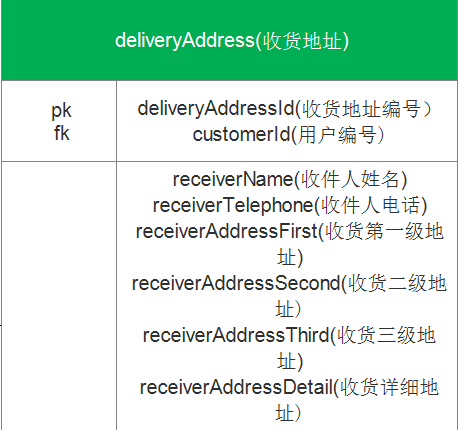
# **4 逻辑设计阶段**

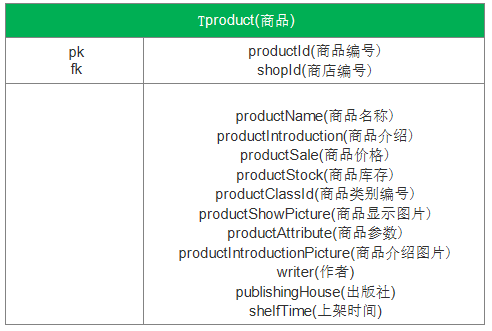
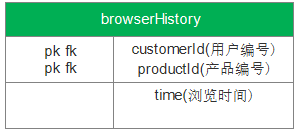
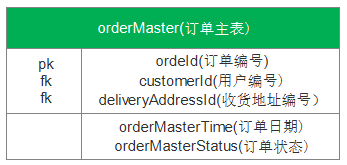
## 4.1 逻辑设计的任务和目标

以上的概念设计阶段是独立于任何一种数据模型的，但是逻辑设计阶段就与选用的DBMS产品发生关系了，系统逻辑设计的任务就是将概念设计阶段设计好的基本E-R图转换为选用DBMS产品所支持的数据模型相符合的逻辑结构。具体内容包括数据组织（将E-R图转换成关系模型，数据库模式定义）、数据处理（画出系统功能模块图）两大任务。

## 4.2 数据组织

### 4.2.1 将E-R图转换为关系模型





### 

### 

### 4.2.2数据库模式定义

数据库的模式定义包括三个方面：

1. 数据的逻辑结构；
2. 数据之间的联系；
3. 数据有关的完整性约束、安全性要求。

customer

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| customerId | INT | not null | 顾客编号 |
| creditCardId | INT | not null | 信用卡编号 |
| customerSex | enum | not null | 顾客性别 |
| customerAccount | VARCHAR(14) | not null | 顾客账号 |
| customerPassword | VARCHAR(14) | not null | 顾客密码 |
| customerMail | VARCHAR(25) | not null | 顾客邮箱 |
| displayPicture | VARCHAR(100) | not null | 顾客头像 |

**CreditCard**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| creditCardId | INT | not null | 信用卡编号 |
| creditAccount | VARCHAR(10) | not null | 信用卡账号 |
| creditPassword | VARCHAR(10) | not null | 信用卡密码 |
| money | DOUBEL | not null | 信用卡余额 |

**deliveryAddress**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| deliveryAddressId | INT | not null | 收货地址编号 |
| receiverName | VARCHAR(15) | not null | 收件人姓名 |
| receiverTelephone | VARCHAR(11) | not null | 收件人电话 |
| receiverAddressFirst | VARCHAR(10） | not null | 收货第一级地址 |
| receiverAddressSecond | VARCHAR(10) | not null | 收货第二级地址 |
| receiverAddressThird | VARCHAR(10) | not null | 收货第三级地址 |
| receiverAddressDetail | VARCHAR(50) | not null | 收货详细地址 |

**Shopper**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| shopperId | INT | not null | 商家编号 |
| account | VARCHAR(14) | not null | 商家账号 |
| password | VARCHAR(14) | not null | 商家密码 |
| mail | VARCHAR(20) | not null | 商家邮箱 |
| idCard | VARCHAR(18) | not null | 商家身份证 |
| onlineName | VARCHAR(20) | not null | 商家名称 |
| displayPicture | VARCHAR(100) | not null | 商家头像 |

**shop**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| shopId | INT | not null | 商店编号 |
| shopName | VARCHAR(40) | not null | 商店名称 |
| introduction | VARCHAR(200) | not null | 商店介绍 |
| openDate | DATETIME | not null | 开店日期 |
| displayPicture | VARCHAR(100) | not null | 商店头像 |

**Product**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| productId | INT | not null | 商品编号 |
| productName | VARCHAR(11) | not null | 商品名称 |
| productIntroduction | VARCHAR(200) | not null | 商品介绍 |
| productSale | DOUBLE | not null | 商品价格 |
| productStock | INT | not null | 商品数量 |
| productClass | VARCHAR(30) | not null | 商品种类 |
| productShowPicture | VARCHAR(140) | not null | 商品显示图片 |
| productAttribute | VARCHAR(400) | not null | 商品参数 |
| productIntroductionPicture | VARCHAR(400) | not null | 产品介绍图片 |
| writer | VARCHAR(40) | not null | 作者 |
| publishingHouse | VRACHAR(40) | not null | 出版社 |
| shelfTime | DATETIME | not null | 上架时间 |

**browserHistory**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| customerId | INT | not null | 顾客编号 |
| productId | INT | not null | 产品编号 |
| BrowserTime | DATETIME | not null | 浏览时间 |

**OrderMaster**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| orderId | INT | not null | 订单编号 |
| customerId | INT | not null | 顾客编号 |
| deliveryAddressId | INT | not null | 收货地址编号 |
| orderDate | DATETIME | not null | 订单日期 |
| orderState | VARCHAR（20） | not null | 订单状态 |

**orderDetail**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| orderId | INT | not null | 订单编号 |
| productId | INT | not null | 产品编号 |
| quantity | INT | not null | 数量 |

**Comment**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| customerId | INT | not null | 顾客编号 |
| productId | INT | not null | 产品编号 |
| commentTime | DATETIME | not null | 评论时间 |
| commerntContent | VARCHAR（200） | not null | 评论内容 |
| goodRate | DOUBlE | not null | 好评率 |

**shoppingCart**

| 列名 | 数据类型 | 可否为空 | 说明 |
| --- | --- | --- | --- |
| customerId | INT | not null | 顾客编号 |
| productId | INT | not null | 产品编号 |
| quantity | INT | not null | 数量 |

# **5 实施阶段**

## 5.1 建立数据库、表

### 5.1.1 建立数据库

Create database commercialcity

### 5.1.2 建立数据表

（1）用户表

CREATE TABLE `customer` (

`customerId` int(11) NOT NULL AUTO\_INCREMENT,

`creditCardId` int(11) NOT NULL,

`customerName` varchar(24) NOT NULL,

`customerSex` enum('男','女') DEFAULT NULL,

`customerAccount` varchar(14) NOT NULL,

`customerPassword` varchar(14) NOT NULL,

`customerMail` varchar(25) NOT NULL,

`displayPicture` varchar(100) NOT NULL,

PRIMARY KEY (`customerId`),

KEY `creditCardId` (`creditCardId`),

CONSTRAINT `customer\_ibfk\_1` FOREIGN KEY (`creditCardId`) REFERENCES `creditcard` (`creditCardId`)

)

(2)信用卡表

CREATE TABLE `creditcard` (

`creditCardId` int(11) NOT NULL AUTO\_INCREMENT,

`creditAccount` varchar(10) NOT NULL,

`creditPassword` varchar(10) NOT NULL,

`money` double NOT NULL,

PRIMARY KEY (`creditCardId`),

UNIQUE KEY `creditAccount` (`creditAccount`)

)

（3）收货地址表

CREATE TABLE `deliveryaddress` (

`deliveryAddressId` int(11) NOT NULL AUTO\_INCREMENT,

`customerId` int(11) NOT NULL,

`receiverName` varchar(15) NOT NULL,

`receiverTelephone` varchar(11) NOT NULL,

`receiverAddressFirst` varchar(10) NOT NULL,

`receiverAddressSecond` varchar(10) NOT NULL,

`receiverAddressThird` varchar(10) NOT NULL,

`receiverAddressDetail` varchar(50) NOT NULL,

PRIMARY KEY (`deliveryAddressId`)

)

（4）商家表

CREATE TABLE `shopper` (

`shopperId` int(11) NOT NULL AUTO\_INCREMENT,

`creditCardId` int(11) NOT NULL,

`account` varchar(14) NOT NULL,

`password` varchar(14) NOT NULL,

`mail` varchar(20) NOT NULL,

`idCard` varchar(18) NOT NULL,

`onlineName` varchar(20) NOT NULL,

`displayPicture` varchar(100) NOT NULL,

PRIMARY KEY (`shopperId`),

UNIQUE KEY `account` (`account`),

UNIQUE KEY `mail` (`mail`),

UNIQUE KEY `idCard` (`idCard`),

UNIQUE KEY `onlineName` (`onlineName`),

KEY `creditCardId` (`creditCardId`),

CONSTRAINT `shopper\_ibfk\_1` FOREIGN KEY (`creditCardId`) REFERENCES `creditcard` (`creditCardId`)

1. ) 商店表

CREATE TABLE `shop` (

`shopId` int(11) NOT NULL AUTO\_INCREMENT,

`shopperId` int(11) NOT NULL,

`shopName` varchar(40) NOT NULL,

`introduction` varchar(200) NOT NULL,

`openDate` datetime NOT NULL,

`displayPicture` varchar(100) NOT NULL,

`goodRate` double NOT NULL,

PRIMARY KEY (`shopId`),

UNIQUE KEY `shopName` (`shopName`),

KEY `shopperId` (`shopperId`),

CONSTRAINT `shop\_ibfk\_1` FOREIGN KEY (`shopperId`) REFERENCES `shopper` (`shopperId`)

)

（6）商品表

CREATE TABLE `product` (

`productId` int(11) NOT NULL AUTO\_INCREMENT,

`shopId` int(11) NOT NULL,

`productName` varchar(40) NOT NULL,

`productIntroduction` varchar(200) NOT NULL,

`productSale` double NOT NULL,

`productStock` int(11) NOT NULL,

`productClass` varchar(30) NOT NULL,

`productShowPicture` varchar(140) NOT NULL,

`productAttribute` varchar(400) NOT NULL,

`writer` varchar(40) NOT NULL,

`publishingHouse` varchar(40) NOT NULL,

`shelfTime` datetime NOT NULL,

`productIntroductionPictureOne` varchar(100) NOT NULL,

`productIntroductionPictureTwo` varchar(100) NOT NULL,

`productIntroductionPictureThree` varchar(100) NOT NULL,

PRIMARY KEY (`productId`),

KEY `shopId` (`shopId`),

CONSTRAINT `product\_ibfk\_1` FOREIGN KEY (`shopId`) REFERENCES `shop` (`shopId`)

)

（7）订单主表

CREATE TABLE `ordermaster` (

`orderId` int(11) NOT NULL AUTO\_INCREMENT,

`customerId` int(11) NOT NULL,

`deliveryAddressId` int(11) NOT NULL,

`orderMasterTime` datetime NOT NULL,

`orderMasterStatus` varchar(10) NOT NULL,

PRIMARY KEY (`orderId`),

KEY `customerId` (`customerId`),

KEY `deliveryAddressId` (`deliveryAddressId`),

CONSTRAINT `ordermaster\_ibfk\_1` FOREIGN KEY (`customerId`) REFERENCES `customer` (`customerId`)

CONSTRAINT `ordermaster\_ibfk\_2` FOREIGN KEY (`deliveryAddressId`) REFERENCES `deliveryAddress` (`deliveryAddressId`)

)

（8）订单明细表

CREATE TABLE `orderdetail` (

`orderId` int(11) NOT NULL,

`productId` int(11) NOT NULL,

`quantity` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`orderId`,`productId`),

KEY `productId` (`productId`),

CONSTRAINT `orderdetail\_ibfk\_1` FOREIGN KEY (`orderId`) REFERENCES `ordermaster` (`orderId`),

CONSTRAINT `orderdetail\_ibfk\_2` FOREIGN KEY (`productId`) REFERENCES `product` (`productId`)

)

（9）浏览记录表

CREATE TABLE `browserhistory` (

`customerId` int(11) NOT NULL,

`productId` int(11) NOT NULL,

`time` datetime NOT NULL,

PRIMARY KEY (`customerId`,`productId`,`time`),

KEY `productId` (`productId`),

CONSTRAINT `browserhistory\_ibfk\_1` FOREIGN KEY (`customerId`) REFERENCES `customer` (`customerId`),

CONSTRAINT `browserhistory\_ibfk\_2` FOREIGN KEY (`productId`) REFERENCES `product` (`productId`)

)

（10）评论表

CREATE TABLE `comment` (

`customerId` int(11) NOT NULL,

`productId` int(11) NOT NULL,

`orderId` int(11) NOT NULL,

`commentContent` varchar(200) NOT NULL,

`commentTime` datetime NOT NULL,

`goodrate` int(11) NOT NULL,

PRIMARY KEY (`customerId`,`productId`,`commentTime`),

KEY `productId` (`productId`),

KEY `orderId` (`orderId`),

CONSTRAINT `comment\_ibfk\_1` FOREIGN KEY (`customerId`) REFERENCES `customer` (`customerId`),

CONSTRAINT `comment\_ibfk\_2` FOREIGN KEY (`productId`) REFERENCES `product` (`productId`),

CONSTRAINT `comment\_ibfk\_3` FOREIGN KEY (`orderId`) REFERENCES `ordermaster` (`orderId`)

)

1. 购物车表

CREATE TABLE `shoppingcartproduct` (

`customerId` int(11) NOT NULL DEFAULT '0',

`productId` int(11) NOT NULL DEFAULT '0',

`quantity` int(11) NOT NULL DEFAULT '0',

`time` datetime NOT NULL,

PRIMARY KEY (`customerId`,`productId`),

KEY `productId` (`productId`),

CONSTRAINT `shoppingcartproduct\_ibfk\_1` FOREIGN KEY (`customerId`) REFERENCES `customer` (`customerId`),

CONSTRAINT `shoppingcartproduct\_ibfk\_2` FOREIGN KEY (`productId`) REFERENCES `product` (`productId`)

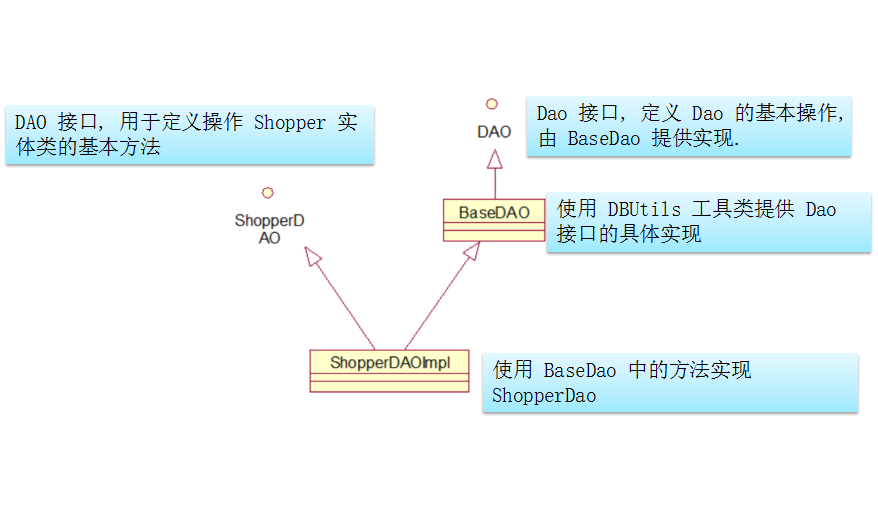
)

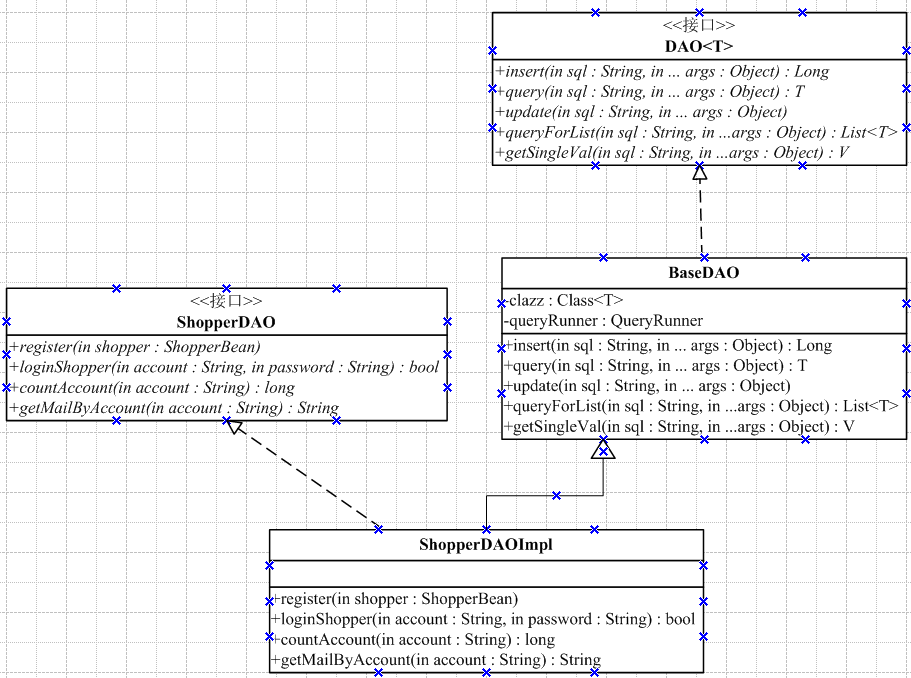
## 5.2数据入库

系统包含顾客的登录注册，浏览商品，购买商品，信息管理，订单管理，购物车管理，评论管理和商家的登录注册，商品管理，订单管理，信息管理，评论管理，销售统计，评论管理。共有11张基本表,采用事先在Excel中录入数据,然后使用MySQL数据导入/导出向导功能,直接将数据导入到相应的基本表中。

## 5.3数据库连接

1. 数据库连接采用c3po数据库连接池经行连接.
2. 数据访问层的设计

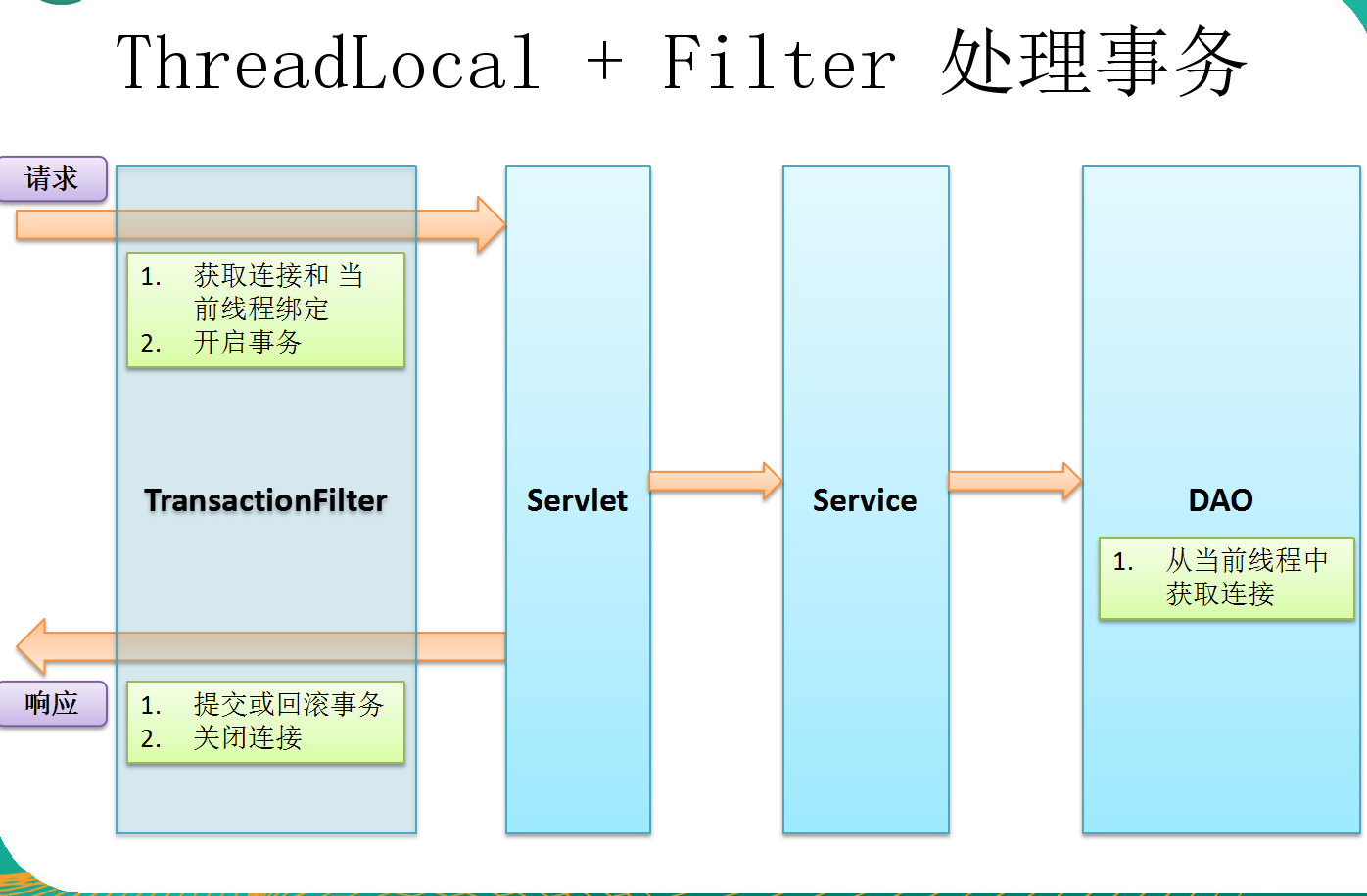




1. 事务操作处理

通过 ThreadLocal.set() 将数据库连接对象的引用保存到各线程的自己的一个 map 中，每个线程都有这样一个map，执行ThreadLocal.get()时，各线程从自己的map中取出放进去的数据库连接对象，因此取出来的是各自自己线程中的对象，ThreadLocal 实例是作为map的key来使的。

原理图

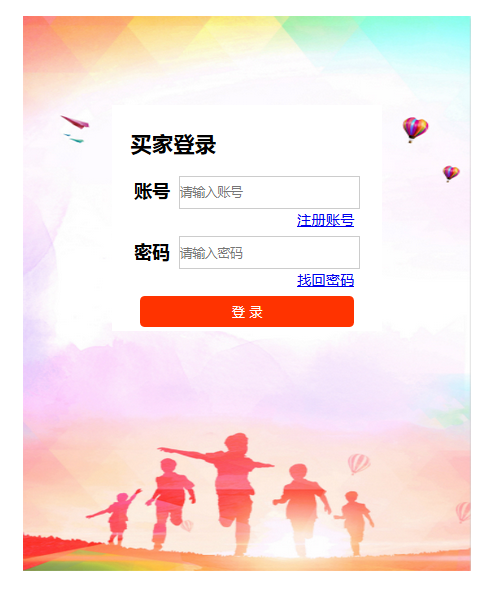


# **6 系统设计**

## 6.1 界面设计

该系统包含用户、商家登录、注册、信息修改，商品浏览、加入购物车、购买，商品上架，订单管理等界面。

用户登录界面：



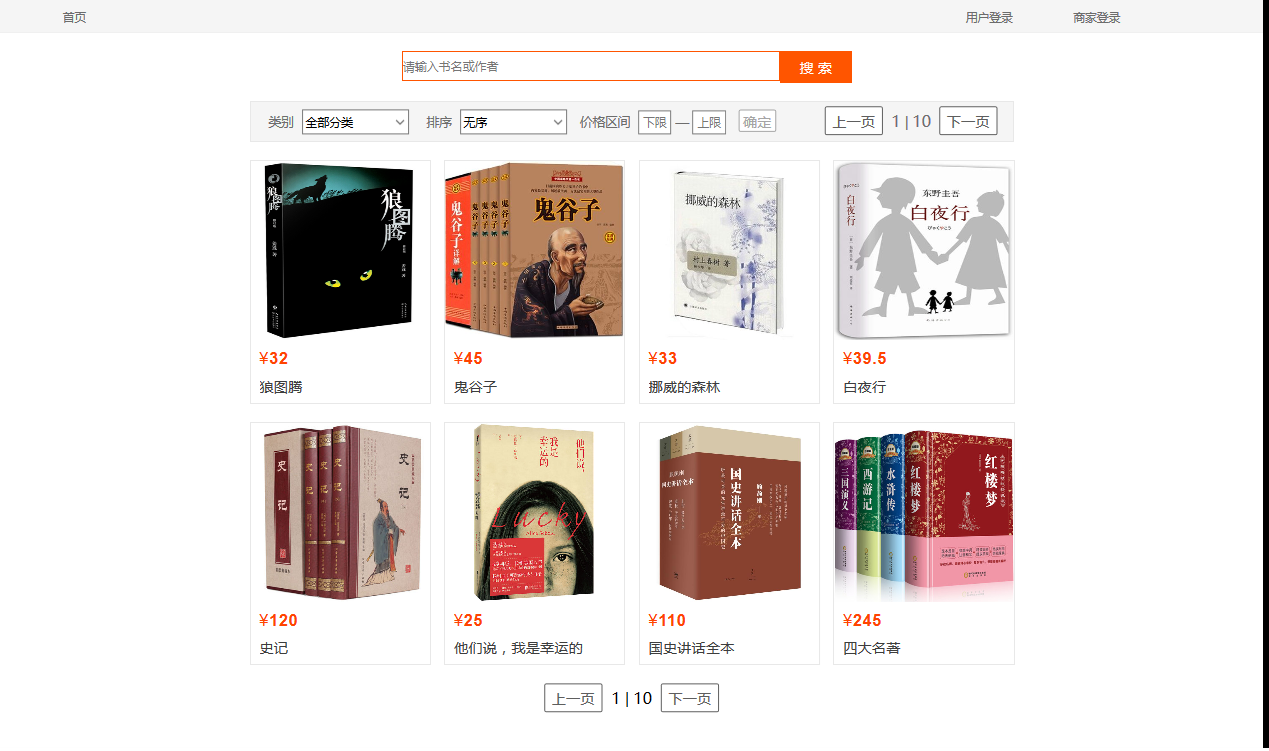
用户注册界面



系统首页界面



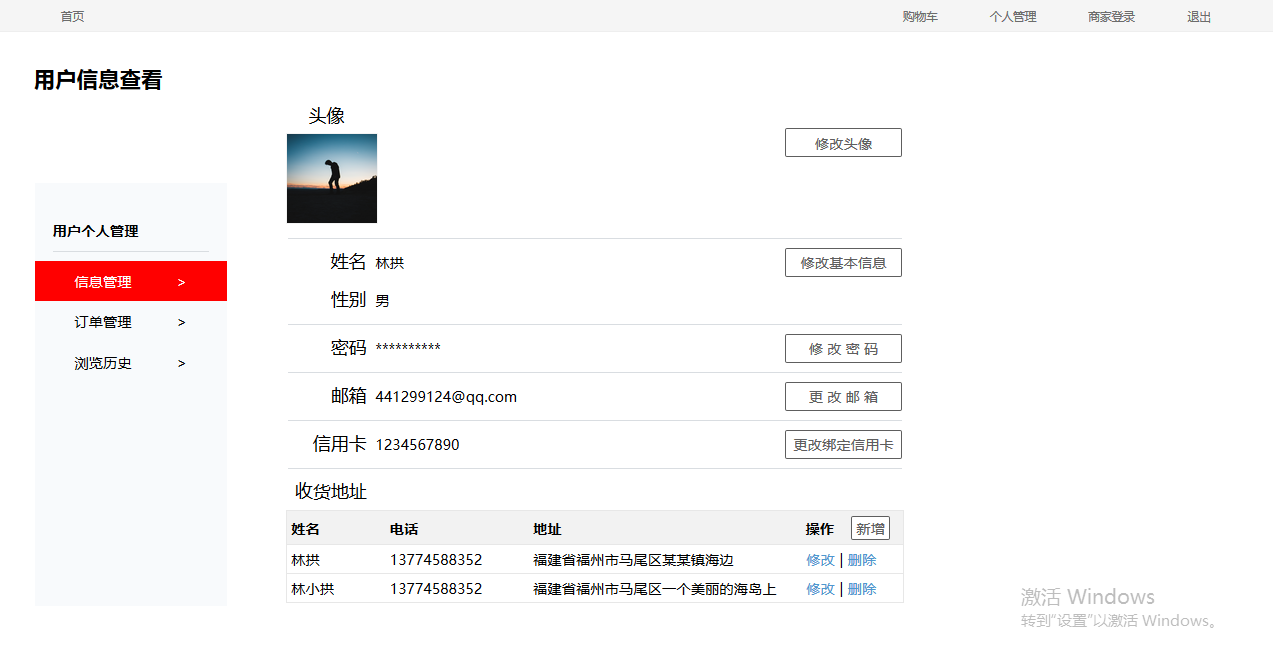
商品浏览



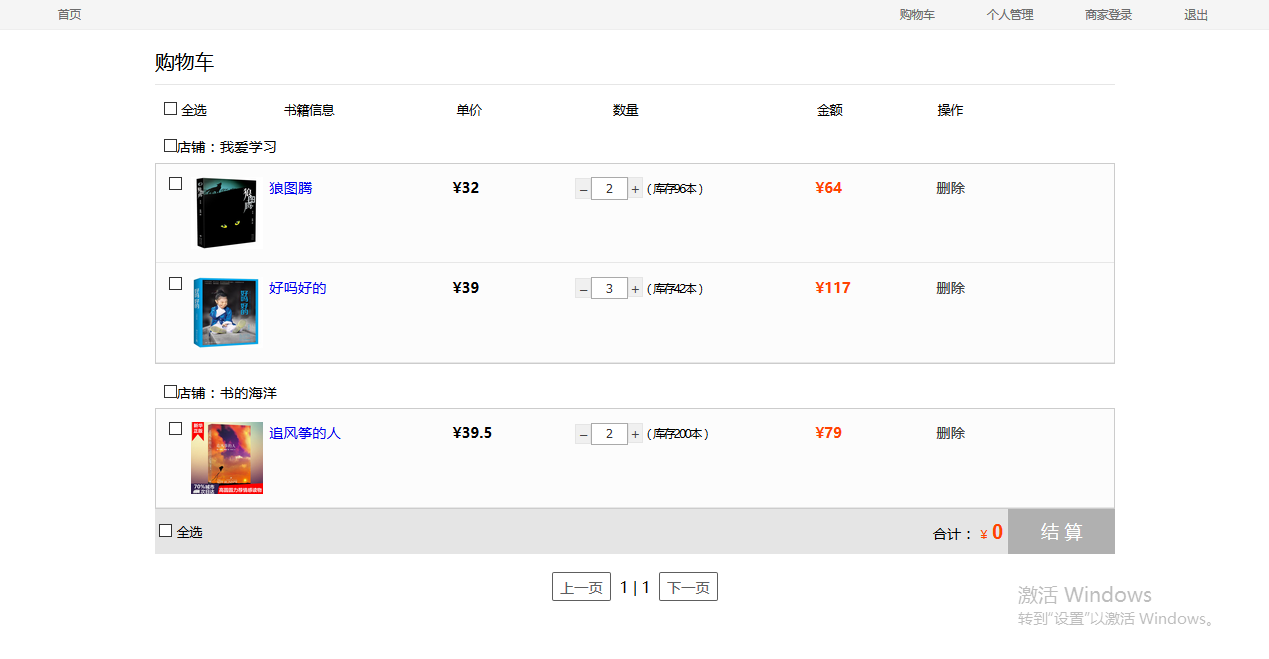
商品详情页面



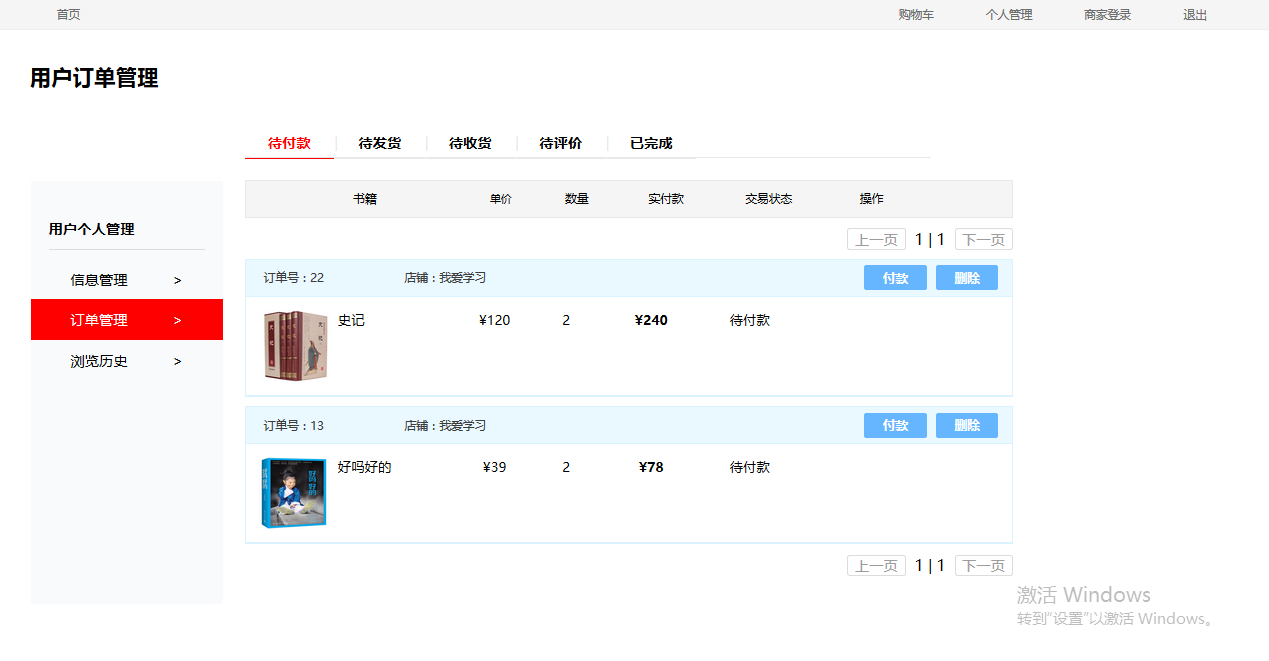
个人信息页面



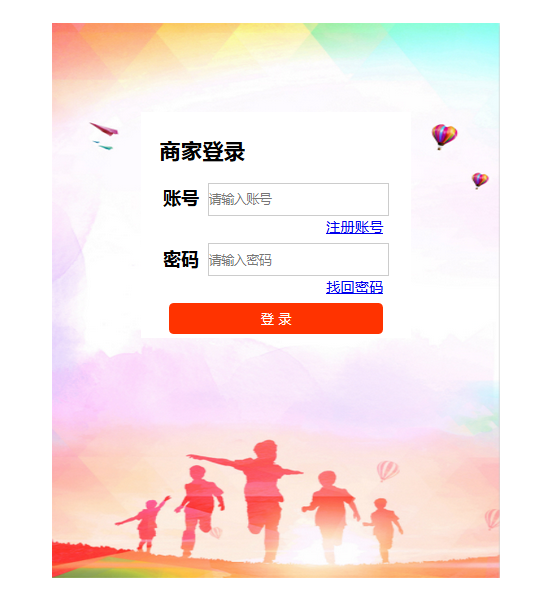
用户订单管理

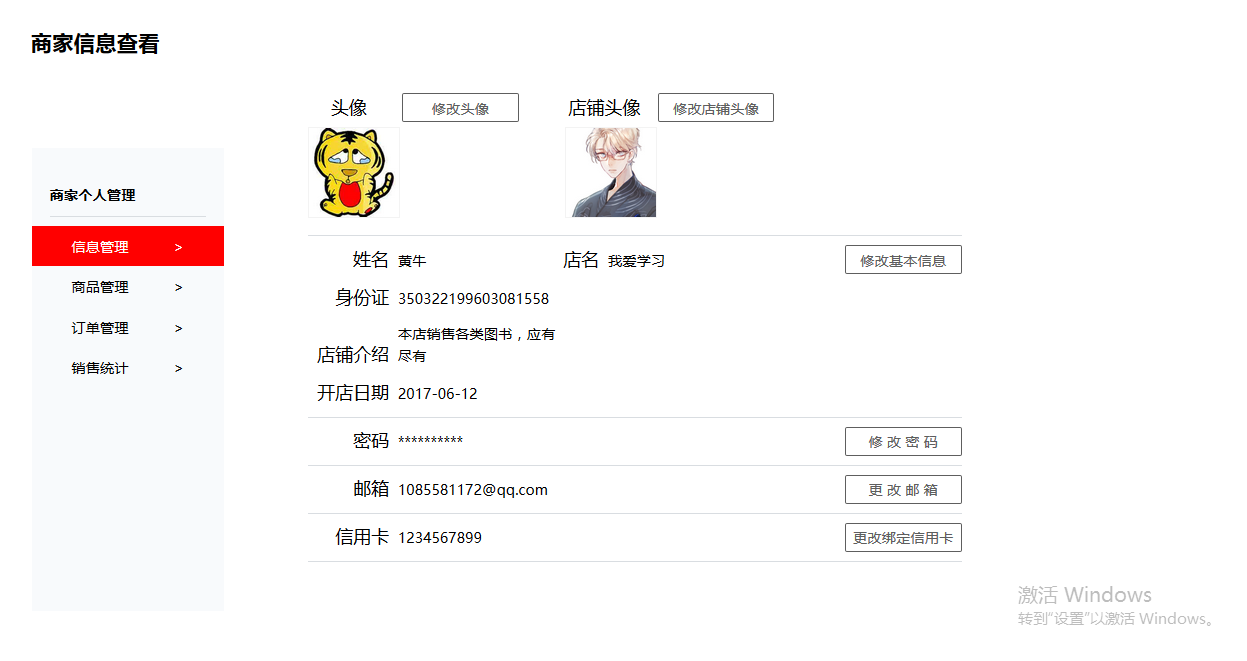


用户购物车

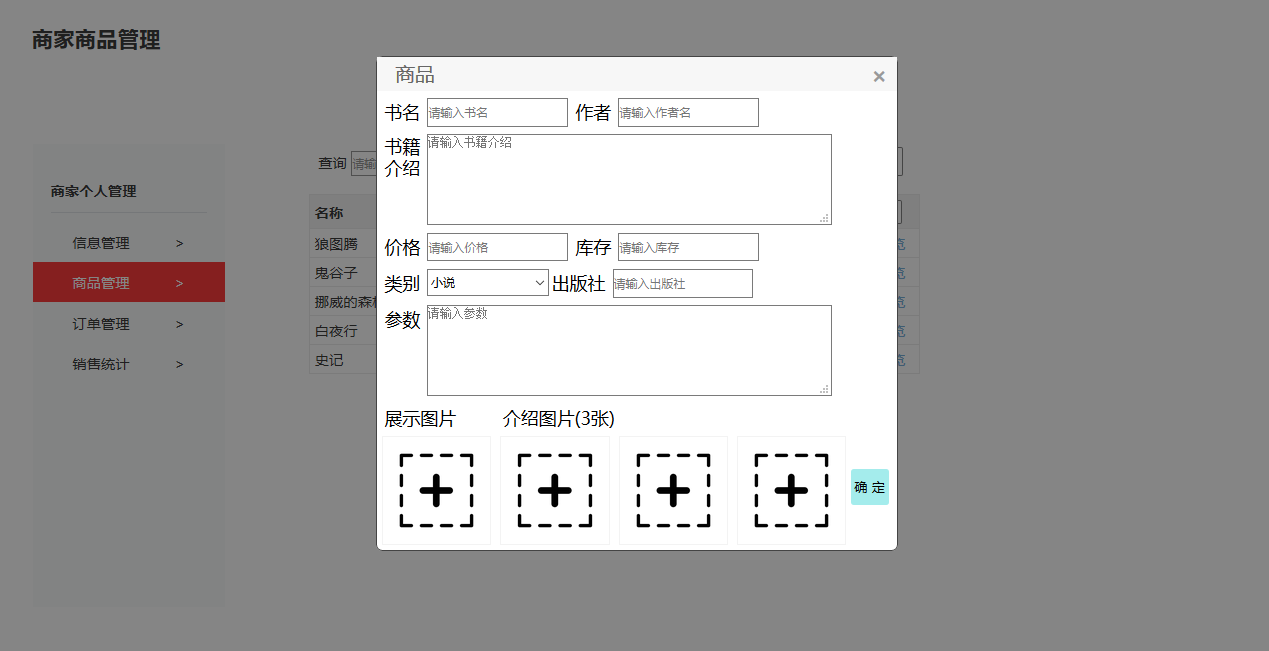


商家登录





商品上架界面



商家订单管理



商家销售统计管理



## 6.2 功能模块设计

### 6.2.1 用户模块设计

功能描述：账号注册、账号登录、找回密码、查看个人信息、修改个人信息、修改购物车、结算购物车、订单查询、订单修改、浏览商品、搜索商品、商品支付、加入购物车、物流查询。

### 6.2.2 商家模块设计

功能描述：账号注册、账号登录、找回密码、店铺注册、查看店铺信息、修改店铺信息、订单查询、订单处理、商品信息修改、商品上架、商品下架、查看评论、查看物流。

## 6.3 关键技术

### 6.3.1 事务处理

[数据库事务](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E4%BA%8B%E5%8A%A1)(Database Transaction) ，是指作为单个逻辑工作单元执行的一系列[操作](http://baike.baidu.com/item/%E6%93%8D%E4%BD%9C/33052)，要么完全地执行，要么完全地不执行。 事务处理可以确保除非事务性单元内的所有操作都成功完成，否则不会永久更新面向数据的资源。通过将一组相关操作组合为一个要么全部成功要么全部失败的单元，可以简化错误恢复并使应用程序更加可靠。例如该系统中商品付款就使用了事务处理。

### 6.3.2 jsp技术

JSP技术使用Java编程语言编写类XML的tags和scriptlets，来封装产生动态网页的处理逻辑。网页还能通过tags和scriptlets访问存在于服务端的资源的应用逻辑。JSP将网页逻辑与网页设计的显示分离，支持可重用的基于组件的设计，使基于Web的应用程序的开发变得迅速和容易。 JSP(JavaServer Pages)是一种动态页面技术，它的主要目的是将表示逻辑从[Servlet](http://baike.baidu.com/item/Servlet)中分离出来。

### 6.3.3 cookie和session机制

会话（Session）跟踪是Web程序中常用的技术，用来跟踪用户的整个会话。常用的会话跟踪技术是Cookie与Session。Cookie通过在客户端记录信息确定用户身份，Session通过在服务器端记录信息确定用户身份。

### 6.3.4 Ajax技术

AJAX 是一种用于创建快速动态网页的技术。

通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

## 6.4 主要功能模块的实现

### 6.4.1 商品上架

private void addProductForm(HttpServletRequest request){

Integer shopId = (Integer) request.getSession().getAttribute("LoginShopId");

//Integer productId = Integer.parseInt(request.getParameter("productId"));

Integer productId = null;

String productName = request.getParameter("productName");

String productIntroduction = request.getParameter("productIntroduction");

System.out.println("productIntroduction="+productIntroduction);

Double productSale = Double.parseDouble(request.getParameter("productSale"));

Integer productStock = Integer.parseInt(request.getParameter("productStock"));

String productClass = request.getParameter("productClass");

String productAttribute = request.getParameter("productAttribute");

String writer = request.getParameter("writer");

String publishingHouse = request.getParameter("publishingHouse");

Date shelfTime = new Date();

ProductBean productBean = new ProductBean(1, shopId, productName, productIntroduction, productSale,

productStock, productClass, null,

productAttribute, writer, publishingHouse, shelfTime,

null, null, null);

productId = (int) productDAO.addProductMessage(productBean);

//记录状态

PageShopProduct pageShopProduct = getPageShopProduct(request);

request.getSession().setAttribute("PageProductAttribute", pageShopProduct);

request.getSession().setAttribute("PageProductId", productId);

request.getSession().setAttribute("PageProductAttributeState", "true");

}

private void addProductNotForm(HttpServletRequest request){

Integer productId = (Integer) request.getSession().getAttribute("PageProductId");

ProductBean productBean = new ProductBean();

changeDisplayPicture(request, productBean);

productBean.setProductId(productId);

productDAO.changeProductForPictureByProductId(productBean);

}

@Override

public void addProduct(HttpServletRequest request, HttpServletResponse response) {

String goal = request.getParameter("goal");

System.out.println("goal="+goal);

if(goal.equals("Form")){

addProductForm(request);

}

if(goal.equals("NotForm")){

addProductNotForm(request);

try {

response.sendRedirect("/CommercialCity/Shop/product/ShopProductManage.jsp");

} catch (Exception e) {}

}

}

### 6.4.2 删除购物车

public void deleteShoppingCartProduct(HttpServletRequest request, HttpServletResponse response) {

Integer customerId = getCustomerId(request);

Integer productId = Integer.parseInt(request.getParameter("productId"));

shoppingCartProductDAO.deleteShoppingCartProductByCustomerIdAndProductId(customerId, productId);

Integer pageNo = Integer.parseInt(request.getParameter("pageNo"));

Page page = new Page();

page.setPageNo(pageNo);

List<ProductBean> listProductBeanOnlyShopId = productDAO.getShoppingCartProductAssociateShopId(customerId);

List<Integer> listShopId = new ArrayList<>();

for(ProductBean productBean:listProductBeanOnlyShopId){

listShopId.add(productBean.getShopId());

}

page.setTotalNumber(listShopId.size());

if(page.getTotalNumber() == 0){

page.setTotalPageNo(0);

}else{

page.setTotalPageNo((page.getTotalNumber()-1)/page.getPageSize()+1);

if(page.getTotalPageNo() < page.getPageNo()){

page.setPageNo(page.getTotalPageNo());

}

}

System.out.println(page);

ObjectMapper mapper = new ObjectMapper();

try {

String message = mapper.writeValueAsString(page);

System.out.println(mapper.writerWithDefaultPrettyPrinter().writeValueAsString(page));

response.getWriter().println(message);

} catch (Exception e) {}

}

# **7 系统测试**

## 7.1 功能测试

安照功能表对电子商务系统用户模块进行一步一步地测试，测试的目的是发现实现和预期出现不符合的部分。在发现不符的情况下，按照序号记录发生的问题，问题的概要，发生的条件和推断导致问题的原因。

同上一步过程一样，对商家模块进行测试。整合完毕后将所得的信息生成报告书，存留备份。

## 7.2 运行时间测试

测试每步操作及实现其功能所需要的时间。需要重复进行，确保由于时间延误导致的失败尽可能少。假如时间延误达到一定的阀值，将其作为一个错误记录下来。

## 7.3 数据库操作与安全测试

测试对数据库删除，写入，修改的操作，并根据数据库的权限要求，进一步测试系统安全性与稳定性的目的。确保系统的安全机制不出现问题。

## 7.4 错误测试

由于错误输入发生的不可避免性质，需要特别强化错误信息的输入。在此，可以将两个模块作为相互独立的部分来进行。

在此还要思考的问题是如果错误输入导致的破坏是不可避免的，应该如何去改进或者怎么去尽量提醒用户不要出现错误的输入。

## 7.5 测试结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 输入数据 | | | 预期输出结果 | 运行输出结果 | 是否正常 |
| 正  常  测  试  输  入  数  据 | 1 | 用户注册 | 注册成功 | 注册成功 | 是 |
| 2 | 用户登录 | 登录成功 | 登录成功 | 是 |
| 3 | 找回密码 | 取得新密码 | 取得新密码 | 是 |
| 4 | 查看个人信息 | 显示个人信息 | 显示个人信息 | 是 |
| 5 | 修改信息 | 修改成功 | 修改错误 | 否 |
| 6 | 修改购物车 | 修改成功 | 修改成功 | 是 |
| 7 | 结算购物车 | 结算成功 | 结算成功 | 是 |
| 8 | 订单查询 | 显示订单信息 | 显示订单信息 | 是 |
| 9 | 订单修改 | 修改成功 | 显示空白 | 否 |
| 10 | 浏览商品 | 显示商品信息 | 显示商品信息 | 是 |
| 11 | 搜索商品 | 显示商品信息 | 显示商品信息 | 是 |
| 12 | 商品支付 | 支付成功 | 支付成功 | 是 |
| 13 | 加入购物车 | 加入购物车成功 | 加入购物车成功 | 是 |
| 14 | 店铺注册 | 注册成功 | 注册失败 | 否 |
| 15 | 店铺信息查看 | 显示店铺信息 | 显示店铺信息 | 是 |
| 16 | 店铺信息修改 | 修改成功 | 修改成功 | 是 |
| 17 | 订单处理 | 处理成功 | 处理成功 | 是 |
| 18 | 商品修改 | 修改成功 | 修改成功 | 是 |
| 19 | 商品上架 | 上架成功 | 上架成功 | 是 |
| 20 | 商品下架 | 下架成功 | 下架成功 | 是 |
| 21 | 查看评论 | 显示评论信息 | 显示评论信息 | 是 |
| 导  致  反  常  输  入  数  据 | 1 | 注册：某项留空 | 提示输入不能为空 | 注册失败 | 是 |
| 2 | 注册：用户名重复 | 提示用户名重复 | 提示用户名重复 | 是 |
| 3 | 登录：密码错误 | 提示密码错误 | 提示密码错误 | 是 |
| 4 | 搜索：关键字留空 | 提示关键字不能为空 | 提示关键字不能为空 | 是 |
| 5 | 购物车：修改数量为0 | 移除该商品 | 移除该商品 | 是 |
| 6 | 修改信息：某项留空 | 提示修改内容不能为空 | 提示修改内容不能为空 | 是 |

# **8 个人心得**

这次实验让我更深刻地认识到电子商务的特点和应用领域，学习和体会到了电子商务的核心思想。在亲身实践中学习和提高。在这个过程中，我更加认识到掌握实际应用技能的重要性，感觉到了电子商务的优越性。更体会到团体配合作业的效率性问题。