**摘要**

随着计算机技术和网络的发展，网络购物平台因为其低成本，无时间，空间限制的优势，渐渐成为新的零售模式和理念，仅仅是信息的浏览和发布已经不能满足人们的需求，人们需要充分享受互连网所带来的的便利，网上购物恰好适应了当今社会快节奏的生活，使人们能够足不出户便可以选购自己心仪的商品。

本系统便是利用jsp+java搭建一个动态的网上购物网站，它运行在windows 10下，使用mysql完成系统数据库的开发，TOMCAT作为服务器，利用ajax完成服务端对客户端的实时响应。本系统分为前台和后台部分，前台针对用户，主要包括用户注册账号，登录系统，购物车管理，订单管理，修改账户密码，添加收货地址信息等。后台针对管理员，主要包括商品管理，处理订单，用户信息管理，公告管理等。

建立后的网站系统是一个动态、交互式、具有用户账号注册登录，商品浏览、商品购买、系统管理、等功能的电子商务网站。

关键字：JSP;Mysql;Tomcat;网上购物系统

**abstract**

With the development of computer technology and network, online shopping platform has gradually become a new retail mode and concept because of its advantages of low cost, no time and space restrictions. Only the browsing and publishing of information can not meet people's needs, and people need to fully enjoy the convenience brought by Internet. Online shopping just adapts to the fast-paced life of today's society People can buy their favorite products without leaving home.

This system uses JSP + java to build a dynamic online shopping website. It runs under Windows 10, uses Mysql to complete the development of the system database, Tomcat as the server, and Ajax to complete the real-time response of the server to the client. This system is divided into foreground and background parts. The foreground is aimed at users, mainly including user registration account, login system, shopping cart management, order management, changing account password, adding receiving address information, etc. Background for administrators, mainly including commodity management, processing orders, user information management, announcement management, etc.

The established website system is a dynamic, interactive, e-commerce website with the functions of user account registration and login, commodity browsing, commodity purchase, system management, etc.

Keywords: JSP; MySQL; Tomcat; online shopping system

**引言**

目前，互联网在以一种令人难以想象的冲击力在改变着人们的生活方式，不仅与人们的生活息息相关，还改变了人们的生产方式。互联网的诞生和迅速发展不仅改变了传统信息传递方式，还打破了传统交流方式中时间和空间的种种限制，极大地改变了人们从形式到内容，从生产到生活，从物质到精神的各种活动，并且不断给人们带来了新的机会和挑战。

自21世纪以来，中国互联网高速发展，涉及到互联网基础建设，互联网政务，互联网学术以及互联网学术等方面。根据有关部门的相关报告，截至2020年，中国网民人数达到9亿，互联网普及率到达64%。庞大的互联网用户规模构成了中国这个世界上最庞大的消费市场。同时，基于计算机科学信息技术的高速发展，数字经济也有了相当夯实的技术基础和用户基础。

本次毕业设计的课题是，利用SSM+bootstrap+jsp技术搭建一个网上商场的购物平台。近年世界各国都出现了电子商务热，越来越多的企业都在建设自己的网站。网上购物这个逐渐流行于21世纪的购物方式已经被越来越多的人所接受我国政府也正全面，积极，稳妥的推进电子商务的发展。网上商店在增加网上支付功能，完善各项服务后以更大的势头发展。电子商务代表着未来世界贸易的发展方向和趋势，其应用和推广将给社会和经济带来巨大的效益。大力发展电子商务在国民经济各个领域的应用，力争在越来越激烈的竞争中占据有利位置。因此对于网上商城的研究和实现，以及搭建一个实用的网上商店是很有必要的。

本系统为消费者提供了网上浏览商品，登陆，注册，搜索商品，加入购物车，购买商品。满足消费者的网上购物的基本活动。为商家提供了，管理注册用户，管理商品信息，增加、删除商品等功能从商家的方面考虑，为商家节约了成本，方便了管理。

1. **绪论**

**1.1研究背景**

21世纪以来，全球各国进入一个崭新的经济时代，[信息](http://www.99inf.net/" \t "_blank)通信[技术](http://www.99inf.net/" \t "_blank)和网络的迅猛发展创造了一个无边界的数字世界，产品和金融服务几乎可以在人们感受不到的时间跨度中在世界各地进行交易，以网络和电子商务为主要特征的新经济以不可争辩的事实席卷了全球每一个行业。

自从中国加入WTO这个国际贸易组织后，传统企业发展的必然趋势是网络化、全球化。通过建设电子商务基础平台，使得企业不必专注于复杂多变的[信息技术](http://www.99inf.net/" \t "_blank)，而用最小的资金投入、最简单的接入方式去联系最广泛的客户和供应商，进行最先进、最有效的电子商务应用服务,相关企业实行电子商务则成为企业发展与国际接轨的必要条件，所以电子购物平台的实现是非常迫切的。

当今的零售企业，如果依旧使用传统的方式例如：发传单，实体店内打广告宣传自己的产品、本公司的文化则显得十分落后，与现代企业的发展不相符，并且难以适应激烈的市场竞争，最终被淘汰。在这个流量为王的时代，互联网明显更适合宣传产品，随意点击一个网站都可以看到铺天盖地的产品广告。因此，当代零售企业纷纷搭建具有本公司文化的网站，利用在互联网技术快速的发展让自己的公司跟上时代的潮流。网上商场就是本人此次毕业设计研究的主要内容。

**1.2研究意义**

随着电子商务的发展，无店铺零售悄然兴起，它是商业经济发展到一定阶段的产物，也是顺应新的消费需求和技术进步的必然结果。无店铺零售则大多不存在地段方面的制约。二是经营成本。借助不断发展创新的计算机技术，无实体店零售往往能够节省从生产到销售过程中的许多过程，进而有效降低经营成本与交易成本，带来消费者、商家双赢的局面[2]。

鉴于互联网的特点以及对国内相关网上购物网站现状的研究和分析，本次毕业设计是以java Web 开发为基础的网上购物网站开发作为本人的毕业设计主要课题。立足于设计一个在网络平台上运行的商城，以小小的力量推动中国互联网的发展。

**1.3发展状况**

随着现代计算机信息相关技术的深入，广泛地应用到现代社会的许多领域并且发挥着不可替代的作用。企业信息管理的必然趋势一定会涉及到计算机信息技术。计算机软件作为计算机信息技术应用的一部分，充分发挥计算机的相关优势，将海量复杂的数据交给计算机系统来处理，相对于人工管理来说，计算机系统有其所无法比拟的优点。例如：查找迅速、可靠性相当高、存储能力极强，不受空间时间约束、保密性好，由于加密技术的存在，重要信息泄露的概率小、寿命长、成本不高等。以上这些特点不仅能够长时间极大地提升企业信息管理的效率，节约成本，是真正的合理利用社会有限的资源，也是企业信息管理数字化、正规化,科学化与全球数字经济接轨的重要条件。

**1.4研究的主要内容**

作为一个网上商城系统,就应该做到能提供强大的业务支持功能，系统能实现顾客的注册账户功能、登录系统功能、商品的查询，订购等功能。该系统基本上具备一个商品销售网站应该具备的功能，该项目基本上体现了构建一个动态商务网站所需要的技术。

主要功能：

（1）用户的功能模块主要是实现了用户对系统账号注册，登录系统，系统公告、商品的浏览以及查询订购商品操作，在注册登录后，可以实现将物品加入购物车并提交订单等功能。

（2）后端信息管理模块主要是实现管理员的登录，以及登陆之后对公告的修改，商品的添加删除修改，用户的订单状态管理，以及删除相关订单，管理前台用户信息等功能

（3）注册登录模块：进入系统主页首先默认是游客角色。可以浏览搜索商品，在注册登陆之后成为会员才能加入商品到购物车并提交订单

（4）购买商品模块：成功登陆之后，才可以将商品加入到购物车，选择完商品之后可以继续浏览商品加入购物车或者直接提交购物车里的商品到订单

（5）管理员模块：登录管理员账户之后可以对商品进行各种操作，以及对公告的修改，对订单状态的处理，对用户信息的维护等。

**第二章 系统开发环境及项目开发工具**

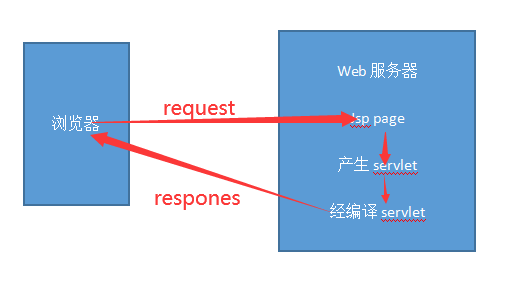
**1.1 开发技术简介**

**1.1.1 JSP简介**

Jsp的全称是 java server page,它是sun公司推出的一种用于开发动态网页的技术。它是在html页面代码中使用jsp标签插入java代码。这个jsp标签通常以<%%>开始和结束。之所以说jsp是动态web开发的技术是因为在编写jsp时虽然像是在写html但是它允许在页面中写java代码，或者利用el，jstl表示java代码。这就使得开发人员在前端页面中能够获得request请求对象和response响应对象，从而实现与浏览器的交互动作。

Jsp页面可以分为2个部分：第一个部分：标准的html标签，这属于静态网页。第二个部分：受java程序控制的部分，这些内容由java程序动态的生成。

Jsp可以被理解为servlet,jsp页面开发完成后由系统编译成为servlet,然后在响应用户发出的请求。也就是说，使用jsp的同时，实际上是在使用servlet。在web项目中，每个jsp页面都会由servlet容器生成对应的servlet。下图可以更好地理解jsp与servlet的关系：



Jsp的优点：

1. 与js相比，虽然javascript可以在浏览器客户端动态的生成html页面，但是它很难与后端服务器交互，因此不能提供相当复杂的服务，比如对数据库的访问和图像处理。
2. 与静态的页面相比，交互性更好，静态页面不能包含动态的信息
3. Jsp性能的比较出色，因为它可以直接在静态的html页面中加入java代码。
4. 因为jsp是基于java servlet api的，所有它拥有许多强大的企业级java api例如：jdbc,ejb,jaxp等等；
5. Java EE包括了jsp，jsp是java EE非常重要的一个部分，它是一个完整的企业级平台，这就表明jsp可以用最简单的方法实现比较复杂的应用。

**1.1.2 Servlet相关简介**

Servlet全称是java servlet,Servlet是一种服务器端的java程序，具有独立性，它可以处理用户传过来的请求，也可以通过响应向用户输送数据。Servlet就是处理用户的HTTP请求，然后回送HTTP响应，若是没有Servlet技术出现的话，我们还要手动的使用一些基本、底层的类去解析HTTP。我们只需要在自己编写的类上重写对应的请求方法就可以处理用户通过给HTTP方法传送过来的请求了。与传统的java程序不同，它由web服务器进行加载，该web服务器必须包含可以启动servlet的jvm。

**1.1.3 MySql相关简介**

世界最流行的关系型数据库管理系统是mysql， 它以其简单，高效，可靠的特点， 在极短的时间就从不知名的数据库系统， 摇身一变变成一个在计算机行业行业几乎是无人不知，无人不晓的开源数据库管理系统。 从微型的嵌入式系统，到小型的web网站，至大型的企业级应用， 到处都可见其身影的存在。

**Mysql的特点：**

1. 它是开源项目，完全免费，自主性极强，使用的成本较低
2. 它支持大型的数据库，可以处理海量数据
3. 操作系统几乎对其不排斥，可以运行各种os之上，并且支持绝大部分开发语言（java,c,php）
4. 它的使用者众多，意味着它的社区活跃，遇到相关的问题可以很快解决
5. 它安装简单，体积小，所以其安装和维护成本低。

**1.1.4 IDEA简介**

 IDEA 全称IntelliJ IDEA，是用于java语言开发的集成环境（也可用于其他语言），它在智能代码、代码自动提示、J2EE支持、JUnit、代码审查、 等方面的体验超棒。IDEA是JetBrains公司的产品，这家公司总部位于捷克共和国的首都布拉格，开发人员以严谨著称的东欧程序员为主。

IDEA的开发效率非常快，尤其是对于java语言，它集成了开发者在开发中需要的许多实用功能，可以让开发者脱离鼠标，做他们想做的事，尽最大的能力加快开发效率，。Idea的优点如下：

(1)对于JUnit单元测试的完美支持

(2)xml语法提示支持：对于市面上几乎所有的流行框架的xml文件都支持语法提示，语法高亮。

(3)不需要任何的插件，完全支持JSP。

（4）使用者众多，意味着社区它的开发者社区活跃，遇到与之相关的问题可以很快解决

(5)Idea 对版本控制支持的很好，无论是SVN 还是Git，亦或是GitHub都支持的非常好。尤其是对Git的支持，图形化的操作页面，会让你省却很多命令的烦恼。让开发者在编程过程中直接在IDEA中就能完成代码的提交、拉取，解决代码冲突，查看版本控制服务器内容等等。

（6）Idea 内嵌了Maven插件，而且也内嵌了Maven，当然也可以使用外部的Maven。

（7）Idea 会把整个工程的文件都做Index，所以搜索文件什么，都是毫秒级别就出来的。

（8）我们还可以有选择地重新编译和加载部分字节码文件，再加上 IDEA 强大的 DEBUG 功能，可以很方便地帮助我们进行代码的调试工作

1.1.5 J2EE

J2EE是一种中间件的基础架构，由于它的存在，开发者只用花费精力去编写业务逻辑的代码，至于其他的系统深层问题，比如内存管理，gc，线程管理等都由系统本身完成。

J2EE：它将应用开发划分为多个不同的层次结构，并在每一个层上定义组件，非常典型的J2EE包括客户层，网页显示层，业务逻辑层，信息管理层。

**1.2 开发环境配置**

**1.2.1 JDK的安装与配置**

（1）从官网下载JDK1.8并安装。

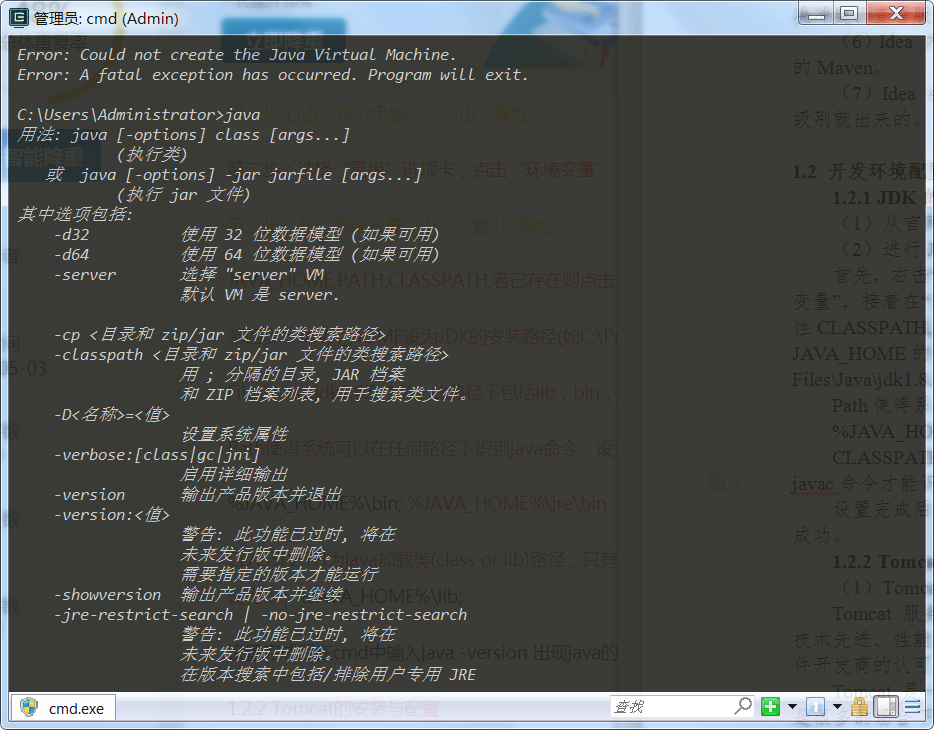
（2）进行JDK1.8的环境变量的设置：

首先，右击“我的电脑”，点击“属性”，然后，选择“高级”选项卡，选择“环境变量”，接着在“系统变量”中，设置三个属性，属性JAVA\_HOME,属性PATH,属性CLASSPATH,如果已存在便点击“编辑”，如果不存在则点击“新建”。最后，把JAVA\_HOME的值改为JDK1.8的安装根路径

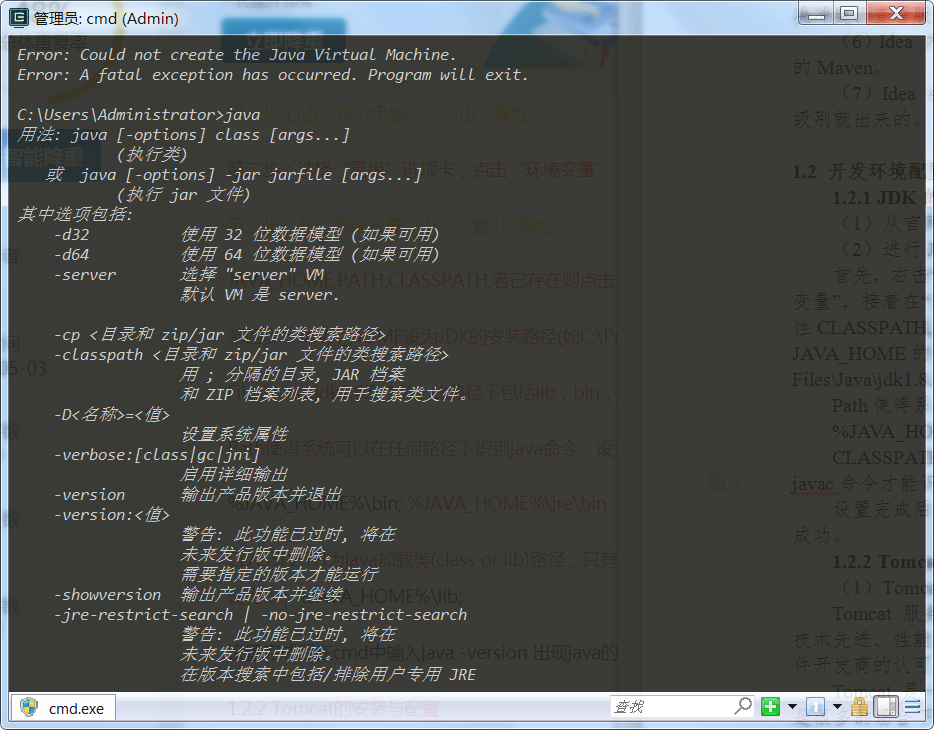
Path设为：%JAVA\_HOME%\bin;

CLASSPATH属性设为：.;%JAVA\_HOME%\lib;

设置完成后在cmd中输入java出现java的版本信息



输入javac出现：

，证明jdk安装成功。

**第三章 需求分析**

**3.1 可行性分析**

**3.1.1经济可行性分析**

缩短了供货周期，压缩了库存，裁减了人员，就可以依据缩短时间的天数和库存物资的减少，以及裁减人员的工资使原材料和劳动力资源得到更合理的利用，从而降低了成本，避免了过多的消耗，减少管理费用的支出等等。所以此项目在经济上是可行的。

**3.1.2技术可行性分析**

Java是一种优秀的编程语言，它最大的优点就算与平台无关。Jsp将前端逻辑和前端的网页设计和显示分离，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。目前J2EE技术已经非常成熟，结合jsp，servlet及相关的关系数据库技术，完全可以达到并实现相关功能。

**3.1.3社会可行性分析**

政府鼓励扶持小型企业发展，网店成为许多小公司的重要选择，加快农村流通体系建设扩大网上消费群体**。**

国家相关部门出台优惠措施，支持传统企业进军电子商务，以网络销售带动门店销售，并对网购平台诚信体系，支付手段等方面予以支持和政策指导，以推进实体市场和网上市场的结合。

对于消费用户来说，会根据消费用户的调查报告设计系统会考虑到人性化界面和操作的方便性。所以社会可行性方面没有问题。

**3.2 系统需求分析**

依据软件工程的理论，系统分析和设计是一个工程开发能否成功的重点，正确获取系统用户的需求为项目开发提供了基础。此网上购物系统的重点是提供产品展示，以及商品订购，是宣传自己商店的商品给用户，让客户能够通过自己的需要来选择要购买的商品：

因为传统购物存在一定的局限性和问题，所以经过总结，有下面几个需求：

顾客需求

（1）普通用户可以通过系统浏览商品；

（2）普通用户可以通过系统搜索框按商品名称搜索商品

（3）普通用户可以注册成为会员；用户可以注册成为会员，并可以登陆进入网站拥有更多功能。

（4）会员可以浏览和检索商品；

（5）会员可以加入购物车；购物车必须会员才可以操作，

（6）会员可以浏览并且管理自己的购物车，根据自己的需求来添加商品或删除商品；

（8）会员可以浏览并且管理自己的订单；

（9）会员可以管理自己的账户信息：修改密码；

管理员的需求：

管理员可以对系统公告进行修改，发布。

管理员可以对顾客，商品，订单进行管理和维护。

购物车的需求：

查看购物车的内容。

清空所有商品的功能。

全选和选部分商品提交。

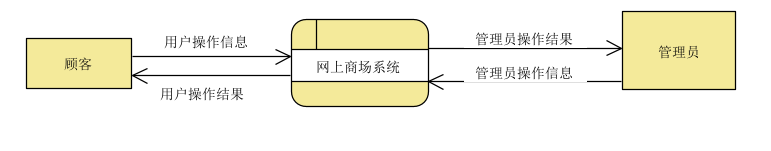
系统优势，方便用户，用户只用短短的几步便可以买到心仪的商品，极大限度的提高了网上商城的运营效率，。搜索功能使用户在极短的时间便可以查询到自己想要的商品，节省了选购时间，在提高用户体验的同时，也降低了服务器压力。

**3.3 结构化分析**

**3.3.1 数据流图**

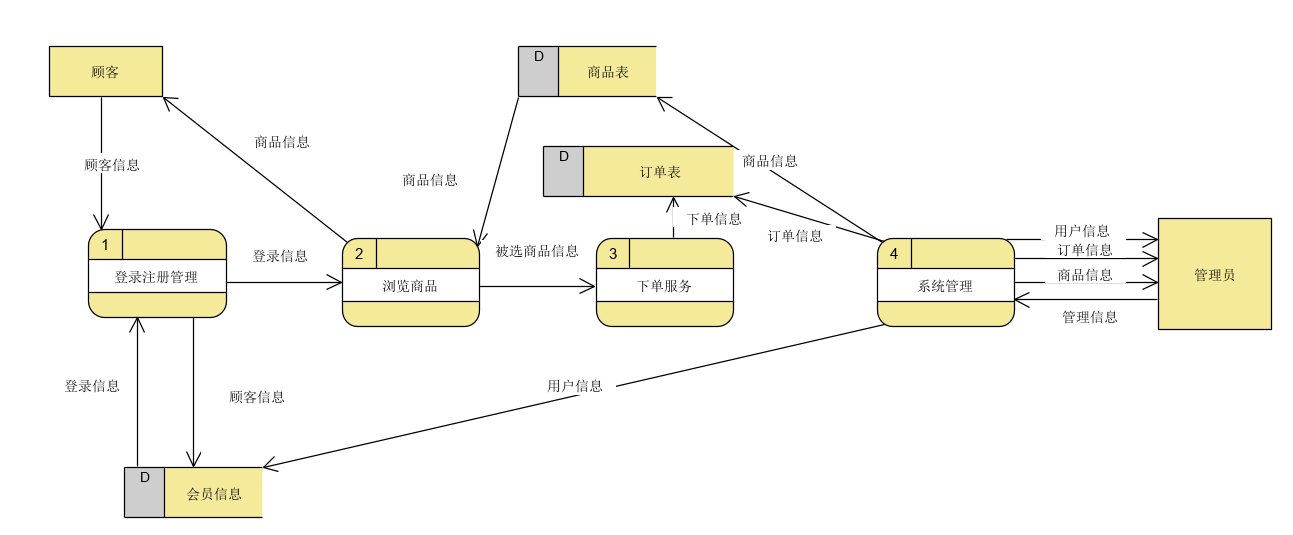
**数据流图是一种描述系统逻辑模型的工具，用少数符号反映信息在系统中的流动，处理存储情况。本系统的数据流图如下所示：**

**首先是顶层图，它主要说明了系统运行的边界。**



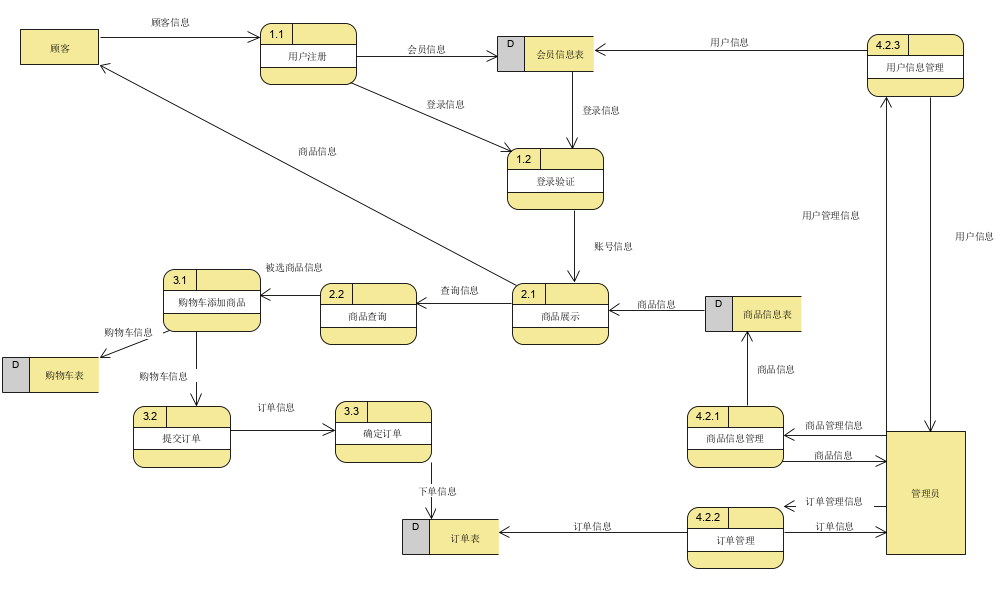
顶层图数据流图

第一层数据流图主要是对上层父图的细化和拆分，如下所示：



一层数据流图

第2层数据流图是对上层1，2，3，4的详细说明和拆分，更清楚的描述了系统数据的处理情况。如下所示：



二层数据流图

**3.3.2 数据字典**

数据字典的作用是对数据流图中的各种成分进行详细说明，作为数据流图的细节补充，和数据流图一起构成完整的系统需求模型数据字典一般应包括对数据项，数据结构、数据存储和数据处理的说明。以下列出本系统的主要数据字典条目

数据存储文件：会员信息表

文件组成：用户id,用户名，密码，姓名，电话，邮箱，性别，生日

数据存储文件：商品表

文件组成：商品id,商品名，商品图片，商品描述，商品价格，商品上架日期，商品库存，商品销量，商品类别

数据存储文件：订单表

文件组成：订单id,用户名，收货人，收货地址，收货人电话，下单时间，订单状态。

数据存储文件：购物车表

文件组成：购物车id,用户名，商品id，数量

数据存储文件：收货信息表

文件组成： 用户名，收货人，收货地址，收货人电话

数据项 数据类型 数据长度

用户id 字符型 不为空

用户名 字符型 8个字母

密码 整型 6个数字

姓名 字符型 255

电话 整型 11个数字

邮箱 字符型 255

性别 字符型 255

生日 date型 不为空

数据流定义：

1. 名称：顾客信息

简述：未注册的用户提供自己的信息注册账号

数据来源：顾客

数据去向：会员信息表

数据组成：用户名+密码+姓名+电话+邮箱+性别+生日

1. 名称：登录信息

简述：以注册的用户提供自己的账号密码登录系统

数据来源：用户

数据去向：会员信息表

数据组成：用户名+密码

1. 名称：被选商品信息

简述：用户选择的商品加入到购物车

数据来源：用户

数据去向：购物车表

数据组成：用户名+商品id+数量

1. 名称：下单信息

简述：用户提交购物车到订单并付款

数据来源：用户

数据去向：订单表

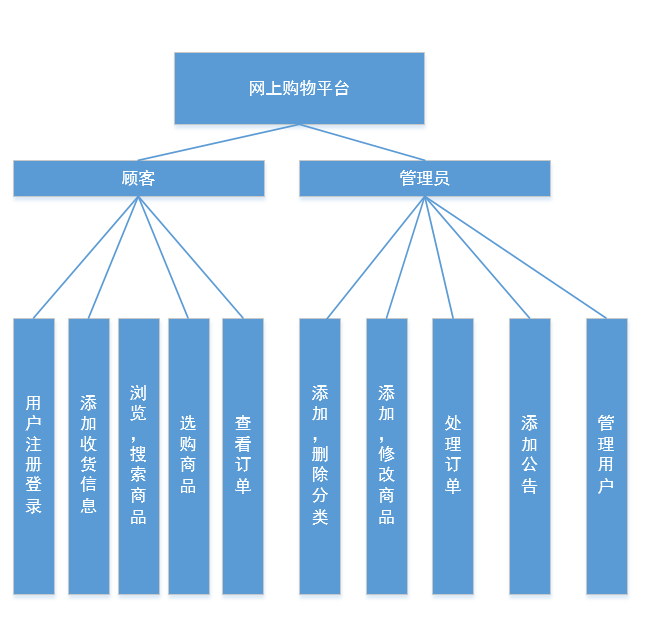
数据组成：用户名+数量+总金额+收货地址+收货人+说活人电话+下单时间+订单状态

1. **系统设计**

**4.1 系统模块设计**

在结构化设计中，模块一般都是按其功能划分的通常称为功能模块，这样做可以保证系统的稳定性以及后期运营的可维护性，合理的功能模块的划分能够极大限度的减少重复劳动，提高开发的工作效率并增大系统的可维护性。

本系统分为两大模块，即前台模块主要面向顾客和后台模块面向管理员。



**4.1.1前台面向用户的功能模块设计**

前台模块主要为登录网站的用户提供商品和服务的信息，填写并提交订单的服务。这样，将前台模块在细分为用户管理模块，商品查询模块，购物车模块，添加收货信息模块，订单模块。顾名思义，这些模块的名称基本解释了他们的功能。下面详细说明一下：

1. 用户管理模块：

为了方便网站的管理，必须由一套完整的用户管理体系，该网站用户管理模块主要实现用户的注册，登录，修改密码等功能。

1. 购物车模块

在超市购物，可以根据自己的需要将很多商品挑选至购物车或购物篮中，然后到收银台结账。而在网上商城中，通常都会采用一种被称为购物车的技术来模拟显示生活，购物车应该一直和用户绑定的，无论什么时候用户在登陆之后都可以看到他原来添加的商品，在网上商城中所选商品必须通过购物车进行暂存，然后生成订单。可以随时添加查看修改清空购物车中的内容，他都会持久化数据到数据库中

1. 商品搜索模块

用户根据商品的名称可以搜索到所需要的商品信息。

1. 添加收货信息模块

用户在登录之后添加收货信息，包括收货地址，收货人，收货人电话，也可以删除收货地址。

1. 订单模块：

用户在购物车中选择商品提交订单，可以查看订单状态，确认收货等等。

**4.1.2 后台模块的功能模块设计**

后台管理模块的功能是对站内的资源进行管理和维护，以后台的业务流程分析和前台各个模块的设计为基础，根据用户的需求分析来确定后台管理模块应具有的功能。后台管理模块的各个功能如下：

1. 管理员身份验证模块：

为合法的用户提供一个后台入口，该模块的功能是对管理员身份进行验证。用户输入登录账号和密码，如果通过验证则跳转后台主页，否则提示错误。

1. 商品管理模块：

向商品表插入商品信息，也可以修改信息，也可以删除，向商品类别表添加商品的类别，删除商品的类别

1. 处理订单模块

管理员对用户订单的执行，修改订单状态，和查看及其对已完成的订单的删除。

1. 用户信息管理模块

管理顾客用户的信息

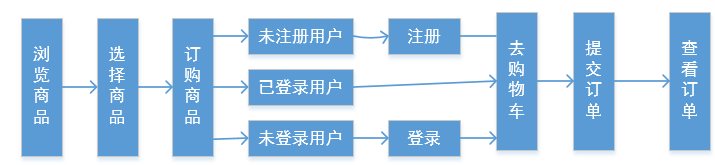
1. 公告管理模块

管理系统公告。

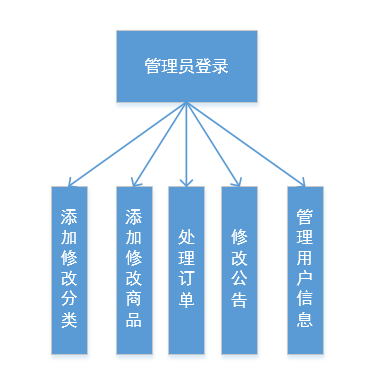
**4.2 系统流程设计**

**本系统分为前台购物流程和后台订单处理流程。**

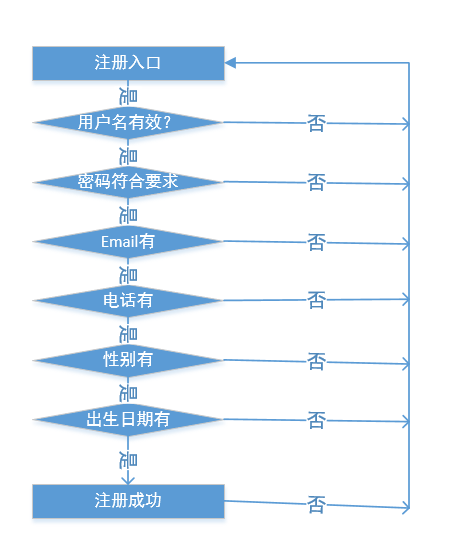
**前台购物流程图**



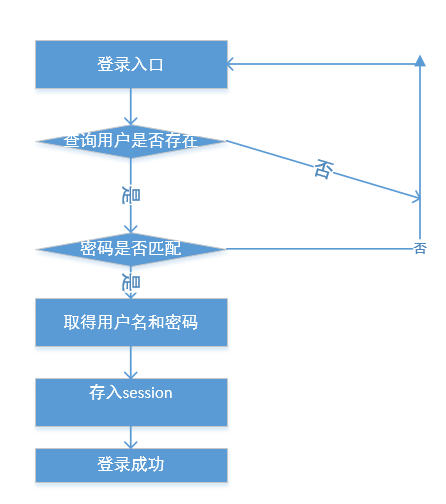
后台管理流程图



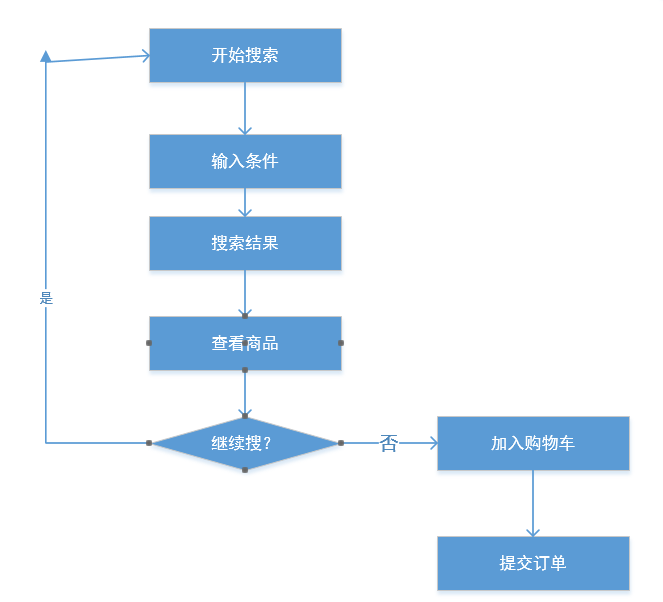
注册功能流程图



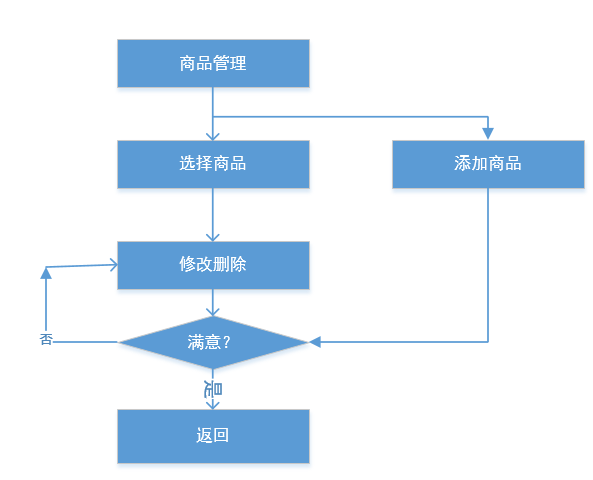
登录流程图



商品搜索购买流程图



商品管理流程图



**4.3 数据库设计**

**4.3.1数据库概要设计**

概念结构是许多模型的共同基础，相对于数据模型，它更独立与机器，更抽象，更稳定。概念结构极其重要，在实际中经常使用ER图来描述概念模型。

E-R模型的组成元素有：实体、属性、联系。E-R模型用E-R图表示。实体是用户工作环境中所涉及的事务，属性是对实体特征的描述。

2.属性是实体的性质。用椭圆框表示，与实体之间用一条线相连表的主码是关键属性，关键属性项加下划线。

3.各子系统模块中主键相同的字段之间存在着相互关联的关系。

4.在程序中实现对他们的完整性和一致性控制。

从数据需求分析中得出系统的实体属性图，遵循三范式原则，对实体之间的依赖关系进行了整合，得出系统E-R图：

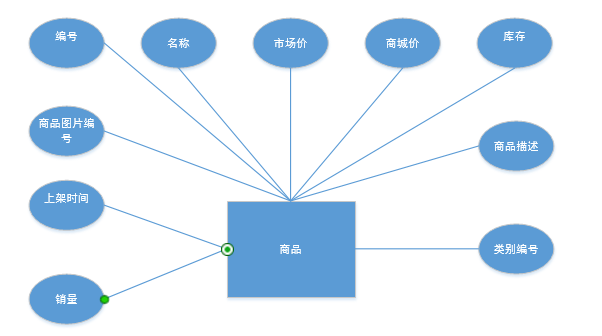
说明：图中用矩形表示实体。实体之间的关系用菱形表示，用无向边把菱形与有关实体连接，在边上标明联系的类型。实体的属性用椭圆表示，并用无向边把实体与属性联系起来。

实体分析：

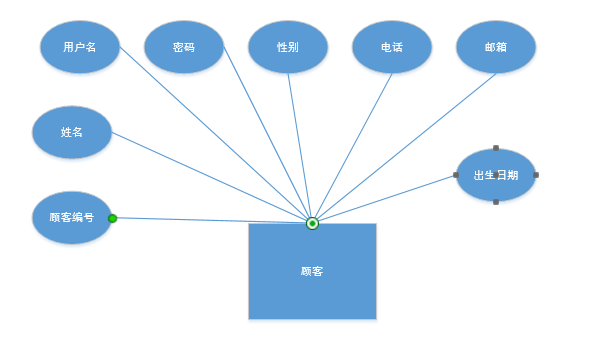
根据网上商城的总体设计，我们定义了网上商城整个系统分为三个模块。由此我们可以首先确定该系统中的实体：管理员、会员、商品、订单、地址信息，公告共6个实体。

实体之间联系

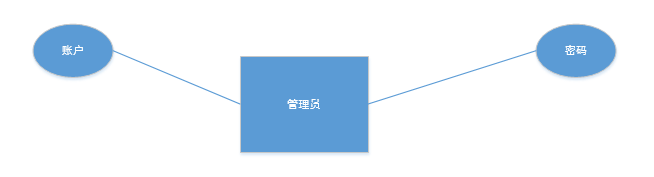
管理员可以管理会员、管理商品、管理订单，一个管理员可以管理多个会员，可以管理多个商品，可以管理多个订单信息。一个订单包括多个商品信息。一个商品可以在多个订单中。一个会员可以有多个订单，一个订单只能被一个会员所有。一个商品可以被多个用户购买，一个用户可以买多个商品，所有其实体属性图，和ER图如下：



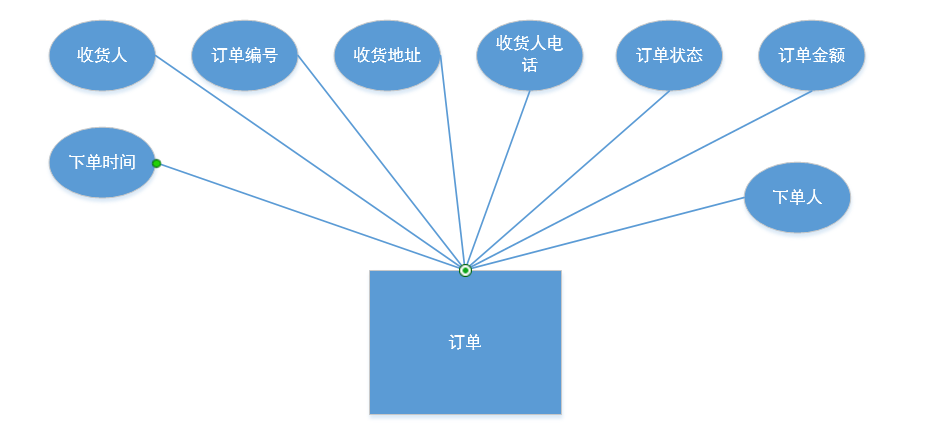
商品实体属性图



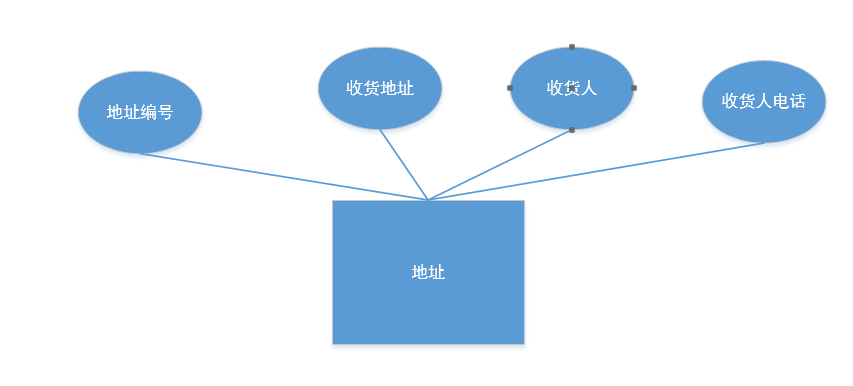
顾客实体属性图



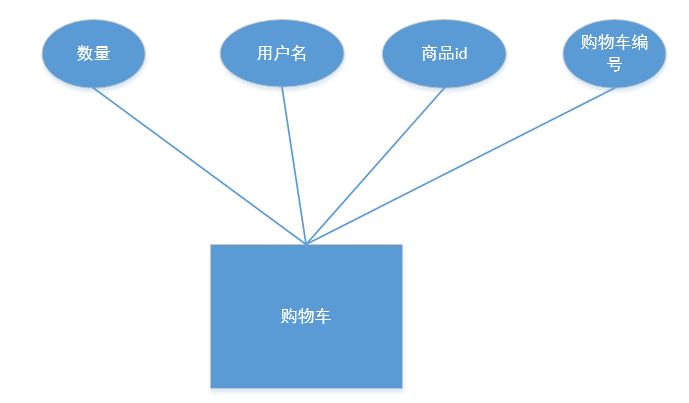
管理员实体属性图



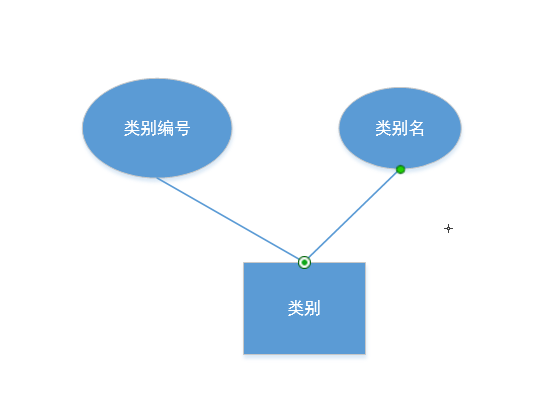
订单实体属性图



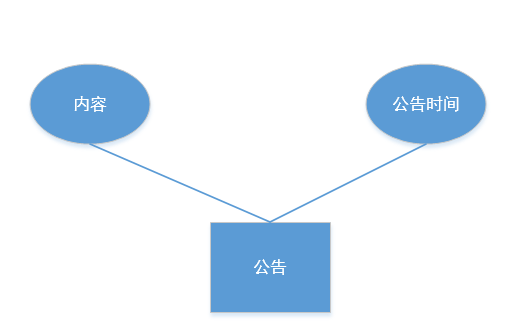
收货地址实体属性图



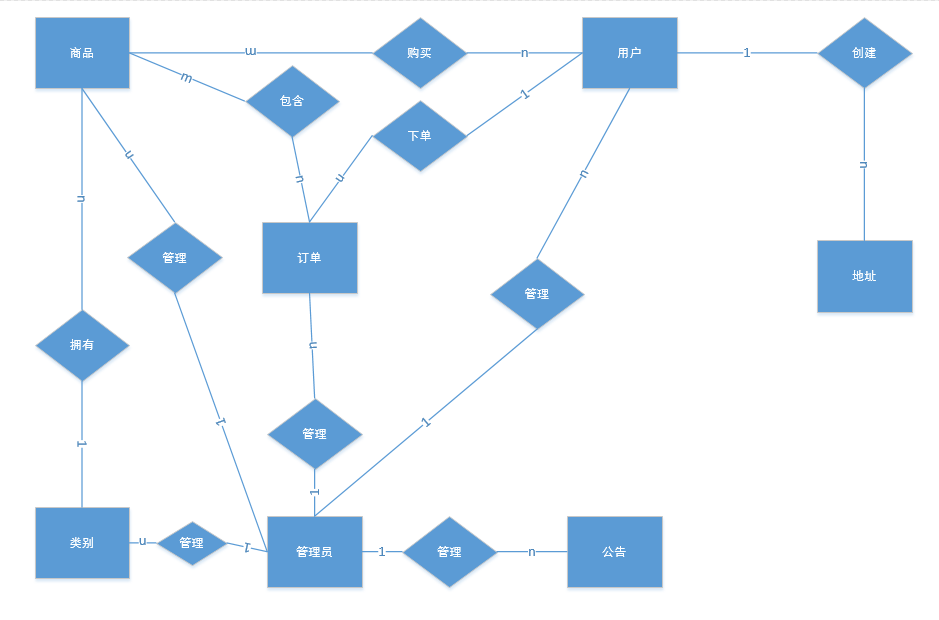
购物车实体属性图



商品类别实体属性图



系统公告实体属性图



ER图

其中，各个实体的属性如下：

（1）用户信息实体的主要属性如下：

用户id ，用户名 ，密码 ，姓名 ，电话 ，邮箱， 性别， 生日；

1. 商品分类实体的主要属性如下：

分类id ，分类名

1. 商品实体的主要属性如下：

商品id ，商品名， 商品图片 ，商品价格， 商品描述 ，商品库存， 商品销量 商品分类 ，商品上架时间

1. 订单实体的主要属性如下：

订单号 订单时间 订单价格 订单收货人 订单收货地址

订单收货人电话，订单状态，订单拥有人

1. 收货信息实体的主要属性如下：

收货地址id

1. 购物车实体的主要属性如下：

购物车id，数量，商品id，用户名

**4.3.2数据库表**

在本系统的数据库中，根据需求建立了如下基本表，具体如下

1. 用户表

User(uid,username,password,name.email,telephone,sex,birthday)

（2）收货信息表

Adress(aid,address,phone,person)

（3）商品表

Product（pid,pimage,pname,pdesc,pdate,pmarket\_price,pcount,is\_latest,cid）

（4）订单表

Orders（oid,address,phone,person,ordertime,state,total,username）

（5）管理员表

admin（acount,pwd）

（6）分类表

Category（cid,cname）

（7）购物车表

Car（cid,count,pid,uid）

（8）订单条目项表

Orderitem（itemid,oid,pid,subtotal,count）

根据以上分析数据库表结构如下：

数据库名：store

表名：user

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 | 说明 |
| 1 | Uid | varchar | 255 | 是 | 不 |  | 用户的唯一id |
| 2 | username | varchar | 255 |  | 不 |  | 登录名字 |
| 3 | password | varchar | 255 |  | 不 |  | 用户密码 |
| 4 | name | varchar | 255 |  |  |  | 用户姓名 |
| 5 | email | varchar | 255 |  |  |  | 邮箱 |
| 6 | telephone | varchar | 255 |  |  |  | 电话 |
| 7 | sex | varchar | 255 |  |  |  | 性别 |
| 8 | birthday | varchar | 255 |  |  |  | 生日 |

表名：product

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 | 说明 |
| 1 | pid | varchar | 255 | 是 | 不 |  | 商品唯一id |
| 2 | pname | varchar | 255 |  | 不 |  | 商品名称 |
| 3 | pimage | varchar | 255 |  | 不 |  | 商品图片 |
| 4 | pdate | varchar | 255 |  | 不 |  | 上架日期 |
| 5 | pdesc | varchar | 255 |  |  |  | 商品描述 |
| 6 | Shop\_price | double | 0 |  | 不 |  | 商品价格 |
| 7 | Market\_price | double | 0 |  | 不 |  | 超市价格 |
| 8 | pcount | int | 11 |  | 不 |  | 库存 |
| 9 | Is\_latest | int | 11 |  | 不 |  | 销量 |
| 10 | cid | varchar | 255 |  | 不 |  | 外键，受类别表id约束 |

表名：orders

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 | 说明 |
| 1 | Oid | varchar | 255 | 是 | 不 |  | 订单唯一id |
| 2 | address | varchar | 255 |  | 不 |  | 该订单地址 |
| 3 | name | varchar | 255 |  | 不 |  | 收件人 |
| 4 | telephone | varchar | 255 |  | 不 |  | 电话 |
| 5 | ordertime | data | 0 |  | 不 |  | 订单时间 |
| 6 | state | int | 11 |  | 不 |  | 订单状态 |
| 7 | total | double | 0 |  | 不 |  | 总金额 |
| 8 | username | varchar | 255 |  | 不 |  | 外键受user表用户名约束 |

表名：orderitem

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 | 说明 |
| 1 | Itemid | varchar | 255 | 是 | 不 |  | 某个订单中的一个购物项 |
| 2 | count | int | 11 |  | 不 |  | 数量 |
| 3 | subtotal | double | 0 |  | 不 |  | 小计 |
| 4 | oid | varchar | 255 |  | 不 |  | 外键受订单表oid约束 |
| 5 | pid | varchar | 255 |  | 不 |  | 外键受商品表oid约束 |

表名：car

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 | 说明 |
| 1 | cid | varchar | 255 | 是 | 不 |  | 购物车唯一id |
| 2 | uid | varchar | 255 |  | 不 |  | 外键 |
| 3 | Pid | varchar | 255 |  | 不 |  | 外键商品id |
| 4 | count | int | 11 |  | 不 |  | 数量 |

表名：admin

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 | 说明 |
| 1 | Acount | varchar | 255 | 是 | 不 |  | 管理员账户 |
| 2 | pwd | varchar | 255 |  | 不 |  | 管理员密码 |

表名：notice

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 | 说明 |
| 1 | nid | varchar | 255 | 是 | 不 |  | 公告id |
| 2 | notice | varchar | 255 |  | 不 |  | 公告内容 |
| 3 | time | date | 0 |  | 不 |  | 增加公告的时间 |

表名：category

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 | 说明 |
| 1 | cid | varchar | 255 | 是 | 不 |  | 分类的id |
| 2 | cname | varchar | 255 |  | 不 |  | 分类名称 |

**4.4本章小结**

**本章完成了网上商城系统所需数据库的设计与实现。按照数据库开发的流程，从需求出发到数据库的概念设计，再到数据的逻辑设计，最后在mysql数据库中完成表的基本创建。**

1. **系统实现**

**5.1前台设计**

网站前台部分提供给用户使用，是系统对外的窗口。主要包括：

1. 网站首页、
2. 查看公告，
3. 产品展示，
4. 用户登录注册，
5. 购买商品，
6. 查看订单

由于篇幅有限，本文只对部分模块的实现进行描述。

**5.1.1用户注册**

**未注册的用户想要登录系统必须要注册账户，只有所有的数据项符合要求才能注册成功，只要有一项不符合，便不能注册成功。**



5.1图 用户注册模块

主要实现代码：

<**script**>

**$**(**"#inputPassword"**).blur(**function** () {

**var** reg=/^[0-9]{6}$/;

**if**(!reg.test(**$**(**"#inputPassword"**).val())){

**$**(**"#ptext"**).text(**"密码需要6个数字"**);

}**else** {

**$**(**"#ptext"**).text(**"密码可以使用"**);

}

})

**$**(**"#inputEmail3"**).blur(**function** () {

**if**(!reg.test(**$**(**"#inputEmail3"**).val())){

**$**(**"#pemail"**).text(**"邮箱格式不对"**);

}**else** {

**$**(**"#pemail"**).text(**"邮箱可以使用"**);

}

})

**$**(**"#inputphone"**).blur(**function** () {

**var** reg= /^1[3456789]\d{9}$/;

**if**(!reg.test(**$**(**"#inputphone"**).val())){

**$**(**"#pphone"**).text(**"电话格式不对"**);

}**else** {

**$**(**"#pphone"**).text(**"电话可以使用"**);

}

})

**$**(**"#name"**).blur(**function** () {

**if**(**$**(**"#name"**).val()!=**""**&&**$**(**"#name"**).val().trim()!=**""**){

**$**(**"#pname"**).text(**"姓名可以使用"**);

}**else** {

**$**(**"#pname"**).text(**"姓名不能为空"**);

}

})

**$**(**"#username"**).blur(**function** () {

**var** reg=/^[A-z]{8}$/;

**if**(reg.test(**$**(**"#username"**).val())){

**$**.**ajax**({

**type**:**"POST"**,

**url**:**"${**pageContext.request.contextPath**}/have.action"**,

**data**:{**"username"**:**$**(**"#username"**).val()},

success:**function**(msg){

**if**(msg == **"yes"**){

**$**(**"#username"**).text(**"该用户名已存在"**);

}**else if**(msg == **"no"**){

**$**(**"#username"**).text(**"该用户名可以使用"**);

}

}

})

}**else**{

**$**(**"#username"**).text(**"用户名需要8个字母"**);

}

})

**function** *m*() {

**if**(**$**(**"#username"**).text()==**"该用户名可以使用"**&&**$**(**"#ptext"**).text()==**"密码可以使用"**&&**$**(**"#pemail"**).text()==**"邮箱可以使用"**&&**$**(**"#pphone"**).text()==**"电话可以使用"**&&**$**(**"#pname"**).text()==**"姓名可以使用"**){

**$**(**".form-horizontal"**).submit();

}**else** {

*alert*(**"请检查所有注册项符合要求！"**)

}

}

</**script**>

### **5.1.2 用户登录**

用户登录页面主要用来接收用户输入的用户名和密码，并更新用户在网站中的状态信息。只有当用户名和密码都正确才能登陆成功，如果输入不正确，系统会显示登录失败。会员登录窗口的运行结果如图所示。

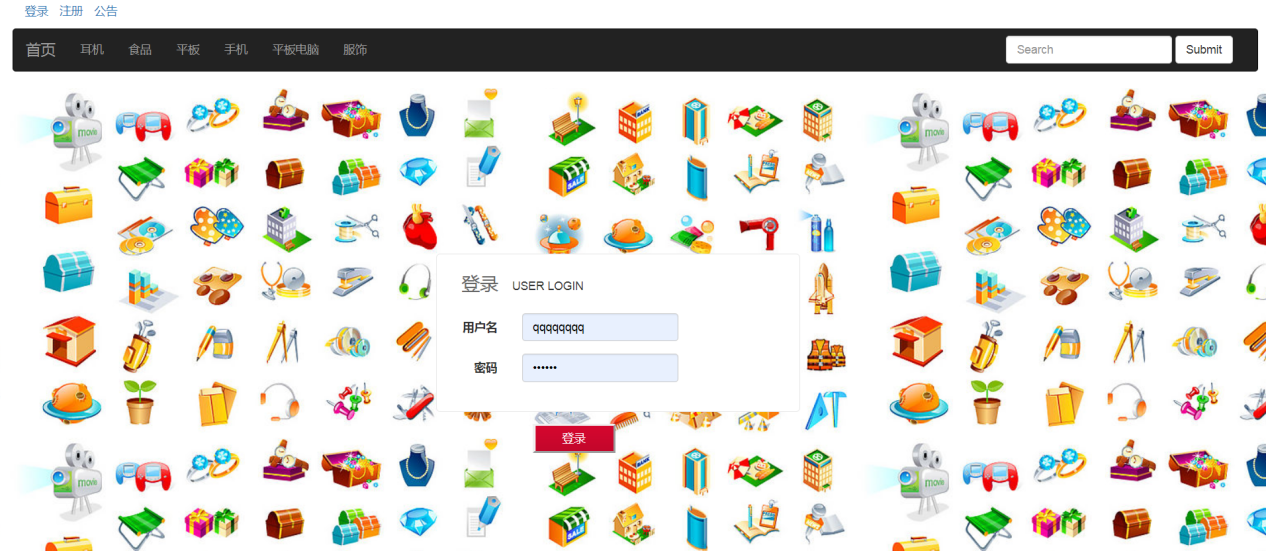


图5-2用户登录

*//登陆验证*

**boolean** isSuccess = **service**.findUserByUsernameAndPassword(user);

ModelAndView mav = **new** ModelAndView();

**if**(isSuccess) {

*// 成功*

mav.addObject(**"msg"**, **"登录成功！，请前往首页选购"**);

*// 添加到session*

session.setAttribute(**"user"**, user);

mav.setViewName(**"index"**);

**return** mav;

}**else** {

*// 失败*

mav.addObject(**"msg"**,**"登录失败"**);

mav.setViewName(**"msg"**);

**return** mav;

}

}

Java代码部分

@RequestMapping(**"/doLogin"**)*//登陆验证***public** ModelAndView doLogin(User user,HttpSession session) {  
  
 **boolean** isSuccess = **service**.findUserByUsernameAndPassword(user);  
 ModelAndView mav = **new** ModelAndView();  
 **if**(isSuccess) {  
 *// 成功* mav.addObject(**"msg"**, **"登录成功！，请前往首页选购"**);  
 *// 添加到session* session.setAttribute(**"user"**, user);  
 mav.setViewName(**"index"**);  
 **return** mav;  
 }**else** {  
 *// 失败* mav.addObject(**"msg"**,**"登录失败"**);  
 mav.setViewName(**"msg"**);  
 **return** mav;  
 }  
}

### **5.1.3 商品浏览**

用户在商品显示页面对具体商品进行详细了解，并在商品显示页面对商品进行订购，已注册登录的会员可以加入购物车并提交订单。

显示主页的热门和最新商品，热门商品是根据销量的多少来排列的，最新商品是按照上架时间来排列显示的。

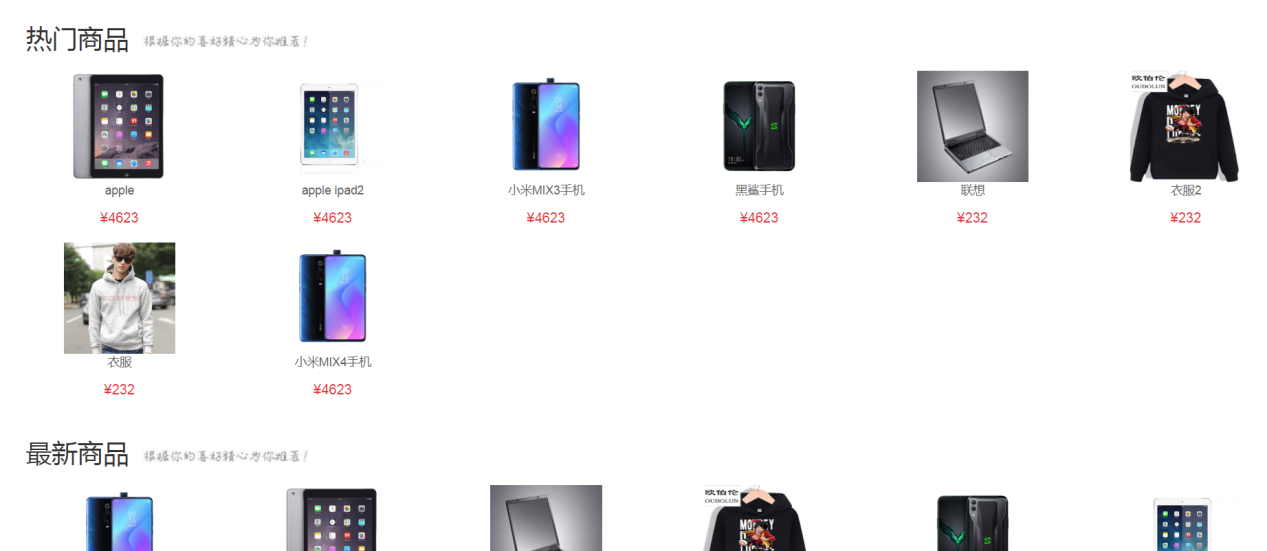


图5-3 首页

*//发送请求，获取最新商品信息*

**$**.getJSON(**"${**pageContext.request.contextPath**}/latestProduct.action"**,**function**(result){

**$**(result).each(**function**(index,element) {

**$**(**"#latest"**).append(**"<div class='col-md-2' ><a href='${**pageContext.request.contextPath**}/infoProduct.action?pid="**+element.**pid**+**"'><imgsrc='${**pageContext.request.contextPath**}/pic/"**+element.pimage+**"' ></a><p><ahref='${**pageContext.request.contextPath**}/infoProduct.action?pid="**+element.**pid**+**">"**+element.pname+**"</a></p>**+element.shop\_price+**"</font></p></div>"**)

})

})

})

单个商品的详细信息页面：



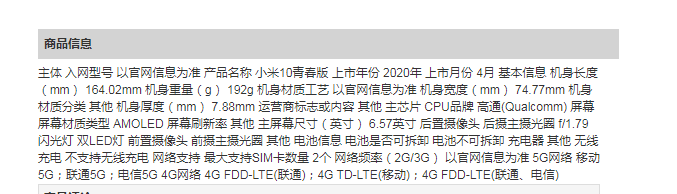


图5-4 商品信息图

实现代码：

<**ahref='${**pageContext.request.contextPath**}/listProductByCategory.action?curPage=1&cid=${**p.category.cid**}'**>**${**p.category.cname **}&nbsp;&nbsp;&gt;**</**a**>

<**a**>**${**p.pname **}**</**a**>

</**div**>

<**div style="margin**:0 **auto**;**width**:950**px**;**"**>

<**div class="col-md-6"**>

<**img style="opacity**: 1;**width**:400**px**;**height**:350**px**;**" title="" class="medium" src="${**pageContext.request.contextPath**}/pic/${**p.pimage**}"**>

</**div**>

<**div class="col-md-6"**>

<**div**><**strong**>**${**p.pname **}**</**strong**></**div**>

<**div style="border-bottom**: 1**px dotted #dddddd**;**width**:350**px**;**margin**:10**px** 0 10**px** 0;**"**>

</**div**>

<**div style="margin**:10**px** 0 10**px** 0;**"**>促销价: <**strong style="color**:**#ef0101**;**"**>￥：**${**p.shop\_price **}**元/份</**strong**>

</**div**>

<**div style="margin**:10**px** 0 10**px** 0;**"**>促销: <**a target="\_blank" title="限时抢购 (2014-07-30 ~ 2015-01-01)" style="background-color**: **#f07373**;**"**>限时抢购</**a**> </**div**>

<**c:if test="${**p.pcount < 10**}"**><**span style="background**: **red"**>销售火爆，库存不足</**span**></**c:if**>

<**c:if test="${**p.pcount>10**}"**>

<**div style="margin**:10**px** 0 10**px** 0;**"**>库存: <**a target="\_blank"** >**${**p.pcount**}**</**a**> </**div**>

</**c:if**>

<**div style="padding**:10**px**;**border**:1**px solid #e7dbb1**;**width**:330**px**;**margin**:15**px** 0 10**px** 0;;**background-color**: **#fffee6**;**"**>

<**form action="${**pageContext.request.contextPath **}/car/addCar1.action" method="post"**>

<**input type="hidden" name="methodStr" value="add2Car"**>

<**input type="hidden" name="pid" value="${**p.pid **}"**>

<**c:if test="${**p.pcount>10**}"**>

<**div style="border-bottom**: 1**px solid #faeac7**;**margin-top**:20**px**;**padding-left**: 10**px**;**"**>购买数量（最多10件）:

<**input id="quantity" name="count" value="1" maxlength="4" size="10" oninput="if**(**value**>10)**value**=10**" type="number"**><**input class="fa fa-cart-plus" aria-hidden="true" value="加入购物车" type="submit"**> </**div**>

</**c:if**>

<**c:if test="${**p.pcount<10**}"**>

暂时无货，非常抱歉

</**c:if**>

</**form**>

**5.1.4添加商品到购物车**

**如果商品库存足够，用户就可以看见加入购物车的按钮，如果库存不足，就会看见库存不足。在商品加入购物车后，用户可以修改数量，另外，用户可以选择部分购物车内的商品提交到订单，而没被选中的商品会留在购物车内，最后也有删除所有商品的按钮，和删除单个商品的按钮。**





图5-5购物车

实现代码：

**public** ModelAndView addCar1(String pid, **int** count,HttpSession session) {

ModelAndView mav=**new** ModelAndView();

User user=(User)session.getAttribute(**"user"**);

*//通过pid找到商品内容*

Product p = **service**.findProductByPid(pid);

*//将count加入到购物车中*

Cart cart=**new** Cart();

cart.setCid(UUIDUtil.*getUUId*());

cart.setCount(count);

cart.setUid(user.getUsername());

cart.setPid(pid);

cart.setProduct(p);

**if**(**service1**.isIncart(cart)){

**service1**.updateCart(cart);

}**else** {**service1**.insetcart(cart);

}

List<Cart> carts=**service1**.findCart(user);

System.***out***.println(carts);

mav.addObject(**"carts"**,carts);

mav.setViewName(**"cart"**);

**return** mav;

}

**public** ModelAndView showCar(HttpSession session) {

User user=(User)session.getAttribute(**"user"**);

List<Cart> carts=**service1**.findCart(user);

ModelAndView mav=**new** ModelAndView();

mav.addObject(**"carts"**,carts);

System.***out***.println(carts);

mav.setViewName(**"cart"**);

**return** mav;

}

**public** ModelAndView delCar(String cid,HttpSession session) {

**service1**.deleteproduct(cid);

User user=(User)session.getAttribute(**"user"**);

List<Cart> carts=**service1**.findCart(user);

ModelAndView mav=**new** ModelAndView();

mav.addObject(**"carts"**,carts);

mav.setViewName(**"cart"**);

**return** mav;

}

**public** String clearCar(HttpSession session) {

User user=(User)session.getAttribute(**"user"**);

**service1**.deletecar(user);

**return "cart"**;

}

}

<**c:if test="${empty** carts**}"**>

<**h1**>购物车空空如也</**h1**>

</**c:if**>

<**c:if test="${not empty** carts **}"**>

<**div class="row"**>

<**c:forEach items="${**carts**}" var="entry"**>

<**tr class="active"**>

<**td width="8%"**><**input type="checkbox" name="option" class="select" onclick="***getMoney*()**"** >选中

</**td**>

<**td width="12%"**>

<**img src="${**pageContext.request.contextPath**}//pic//${**entry.product.pimage**}" width="70" height="60"**>

</**td**>

<**td width="20%"**>

<**a target="\_blank"**>**${**entry.product.pname**}**</**a**>

</**td**>

<**td width="20%"**>

**${**entry.product.shop\_price **}**

</**td**>

<**td width="20%"**>

<**input onblur="***up*(**this**)**" id="count" onchange="***Xiaoji*()**" patten="[1-9][0-9]?" oninput="if**(**value**<1){**value**=1}**else if**(**value**>=10){**value**=10}**" type="number" name="quantity" value="${**entry.count **}" maxlength="4" size="10"**>

</**td**>

<**td width="10%" id="total"**>

<**span class="subtotal" name="price"**></**span**>

</**td**>

<**td width="10%"**>

<**a href="javascript:;" class="delete" onclick="***fn1*(**'${**entry.cid**}'**)**"**>删除</**a**>

</**td**>

<**td style="display**: **none"**>

<**span**>**${**entry.cid**}**</**span**>

</**td**>

<**td style="display**: **none**;**"**>

<**a**>**${**entry.product.pid**}**</**a**>

</**td**>

</**c:if**>

<**div**>

<**c:if test="${not empty** carts**}"**>

<**input style="margin-left**: 100**px" type="checkbox" id="all"**>全选

</**c:if**>

</**div**>

<**div style="text-align**:**right**;**"**>

**&nbsp;** 商品金额(单位：￥): <**input type="text" id="in" readonly**>

</**div**>

<**a href="${**pageContext.request.contextPath **}/car/clearcar.action? id="clear**" class="clear">清空购物车</**a**>

<**button id="submi" onclick="***Getcid*()**"**>提交订单</**button**>

Js部分：

<**script**>

**function** *my*() {

***window***.**location**.**href**=**"${**pageContext.request.contextPath **}/order/addOrder.action?totalRow="**+totalRow;

}

**function** *up*(obj) {

**var** $td = **$**(obj).parents(**'tr'**).**children**(**'td'**);

**var** name= $td.eq(8).find(**"a"**).text();

**var** count= $td.eq(4).find(**"input"**).val();

***console***.log(name+11);

***console***.log(name)

**$**.**ajax**({

**type**: **"post"**,

**url**: **"${**pageContext.request.contextPath **}/car/updatecount.action"**,

**data**: {**"name"**:name,

**"count"**:count},

success: **function**(msg){

**if**(msg==**"yes"**){

***console***.log(1);

*// history.go(0);*

}**else** {

***console***.log(0)

}

}

});

}

***window***.**onload**=**function** (ev) {

*Xiaoji*();

**var** all=***document***.getElementById(**"all"**);

**var** inputs=***document***.getElementsByClassName(**"select"**);

all.addEventListener(**'change'**,**function** () {

**for**(i=0;i<inputs.**length**;i++){

**if**(all.**checked**){

inputs[i].**checked**=all.**checked**;

*getMoney*();

}**else** {

inputs[i].**checked**=**false**;

**$**(**"#in"**).val(0)

}

}

})

**for**(**var** i=0;i<inputs.**length**;i++){

inputs[i].addEventListener(**'change'**,**function** () {

*qx*();

});

}

**var** *qx*=**function** () {

**var** count=0;

**for**(**var** j=0;j<inputs.**length**;j++){

**if**(inputs[j].**checked**){

count++;

}**if**(count==inputs.**length**){

all.**checked**=**true**;

}**else** {

all.**checked**=**false**;

}

}

}

}

**function** *getMoney*() {

**var** totalRow=0;

**$**(**":checkbox:checked"**).parents(**"tr"**).find(**'td:eq(5)'**).each(**function**(){

totalRow += *parseFloat*(**$**(**this**).text());

***console***.log(totalRow);

});

**$**(**"#in"**).val(totalRow)

}

**function** *Xiaoji*() {

**var** tb=***document***.getElementById(**"my"**)

**var** rows=tb.**rows**;

**for**(**var** i=1;i<rows.**length**;i++){

rows[i].**cells**[5].**innerHTML**= *parseFloat*(rows[i].**cells**[3].**innerHTML**)\**parseFloat*(rows[i].**cells**[4].**getElementsByTagName**(**"INPUT"**)[0].**value**)

*getMoney*();

}

}

**function** *Getcid*() {

**var** arr=**new *Array***();

**$**(**":checkbox:checked"**).parents(**"tr"**).find(**'td:eq(7)'**).each(**function**(){

arr.push(**$**(**this**).find(**"span"**).text());

})

**var** t=**$**(**"#in"**).val();

***console***.log(arr);

**var** inputs=***document***.getElementsByClassName(**"select"**);

**var** count=0;

**for**(**var** j=0;j<inputs.**length**;j++){

**if**(inputs[j].**checked**){

count++;

}}

**if**(count==0){

**$**(**"#submi"**).attr(**"disabled"**,**false**);

*alert*(**"已选商品为空，请至少选择一件商品"**)

}**else** { ***window***.**location**.**href**=**"${**pageContext.request.contextPath **}/order/addOrder.action?"**+**"totalRow="**+t+**"&arr="**+arr;

}

}

</**script**>

**5.1.5订单相关页面**

**当用户从购物车选中商品提交到订单时，订单开始生成，此时订单处于未付款状态，这时候用户可以退出，在订单页面可以看到订单状态，只有当用户确认订单之后，订单的状态才是已付款等待发货的状态**





图5-6 订单

代码实现：

<**c:forEach items="${**order.items **}" var="item"**>

<**tr class="active"**>

<**td width="60" width="40%"**>

<**input type="hidden" name="id" value="22"**>

<**img src="${**pageContext.request.contextPath**}/pic/${**item.product.pimage**}" width="70" height="60"**>

</**td**>

<**td width="30%"**>

<**a target="\_blank"**>**${**item.product.pname **}**</**a**>

</**td**>

<**td width="20%"**>

￥**${**item.product.shop\_price **}**

</**td**>

<**td width="10%"**>

**${**item.count **}**

</**td**>

<**td width="15%"**>

<**span class="subtotal"**>￥**${**item.subtotal **}**</**span**>

</**td**>

</**tr**>

</**c:forEach**>

</**tbody**>

</**table**>

</**div**>

<**div style="text-align**:**right**;**margin-right**:120**px**;**"**>

商品金额: <**strong style="color**:**#ff6600**;**"**>￥**${**order.total **}**元</**strong**>

</**div**>

</**div**>

<**div**>

<**hr**/>

<**div class="col-sm-5"**>

收 &nbsp &nbsp 货 &nbsp 人：<**select id="inputPassword3" name="name"**>

<**c:forEach items="${**list**}" var="entry"** >

<**option value="${**entry.person**}" id="username"**>**${**entry.person**}**</**option**>

</**c:forEach**>

</**select**>

</**div**>

</**div**>

<**div class="form-group"**>

<**div class="col-sm-5"**>

收 货 地 址 ：<**select id="username" name="address"**>

<**c:forEach items="${**list**}" var="entry"** >

<**option value="${**entry.address**}"**>**${**entry.address**}**</**option**>

</**c:forEach**>

</**select**>

</**div**>

</**div**>

<**div class="form-group"**>

<**div class="col-sm-5"**>

收货人电话：<**select id="confirmpwd" name="telephone"**>

<**c:forEach items="${**list**}" var="entry"** >

<**option value="${**entry.phone**}"**>**${**entry.phone**}**</**option**>

</**c:forEach**>

</**select**>

</**div**>

</**div**>

<**a href="${**pageContext.request.contextPath**}/adress.action"**>新增收货信息</**a**>

<**hr**/>

<**div style="margin**:0 **auto**; **margin-top**:10**px**;**"**>

<**strong**>我的订单</**strong**>

<**c:forEach items="${**page.list **}" var="order"**>

<**tbody**>

<**tr** >

<**th colspan="2"**>**${**order.oid **}** </**th**>

<**th colspan="2"**>**${**order.address**}** </**th**>

<**th colspan="2"**>**${**order.name **}** </**th**>

<**th colspan="2"**>**${**order.telephone **}** </**th**>

<**th colspan="2"**>

*<!-- 判断，如果这个订单的state为0显示去付款 -->*

<**c:if test="${**order.state == 0 **}"**>

<**a href="${**pageContext.request.contextPath **}/order/payorder.action?oid=${**order.oid**}"**>去付款</**a**>

</**c:if**>

<**c:if test="${**order.state == 1**}"**>

已付款（总金额**${**order.total**}**）等待发货

</**c:if**>

<**c:if test="${**order.state == 2**}"**>

已付款（总金额**${**order.total**}**）<**a onclick="***cl*(**this**)**"**>确认收货</**a**>

</**c:if**>

<**c:if test="${**order.state == 3**}"**>

已完成

</**c:if**>

</**th**>

<**th colspan="2"**>**${**order.ordertime **}**

<**a href="${**pageContext.request.contextPath **}/order/listDetail.action?oid=${**order.oid**}&curPage=1"**>订单详情</**a**>

</**th**>

</**tr**>

Js部分：

<**script**>

**function** *cl*(obj) {

**var** $th = **$**(obj).parents(**'tr'**).**children**(**'th'**);

**var** oid=$th.eq(0).text();

***console***.log(oid);

**$**.**ajax**({

**type**:**"post"**,

**url**:**"${**pageContext.request.contextPath **}/order/utilOrder1.action?"**,

**data**:{**"oid"**:oid},

success:**function** (msg) {

*alert*(**"操作成功"**)

***history***.go(0);

}

})

}

</**script**>

Java部分：

**public** ModelAndView addOrder(HttpSession session , **double** totalRow, @RequestParam(value=**"arr"**) String[] arr) {

ModelAndView mav = **new** ModelAndView();

System.***out***.println(1);

List<Cart> carts=**new** ArrayList<Cart>();

**for**(String a:arr){

**service**.findcid(a);

carts.add(**service**.findcid(a));

}

User loginUser = (User)session.getAttribute(**"user"**);

**if**(loginUser == **null**) {

mav.addObject(**"msg"**, **"请先登录，再继续购买！"**);

mav.setViewName(**"login"**);

**return** mav;

}

Order order = **new** Order();

order.setOid(UUIDUtil.*getUUId*());

order.setState(0);

order.setOrdertime(**new** Date(System.*currentTimeMillis*()));

order.setUser(loginUser);

**service**.addOrder(order);

**for**(Cart c : carts){

OrderItem orderItem = **new** OrderItem();

orderItem.setItemid(UUIDUtil.*getUUId*());

Product p=**service2**.findProductByPid(c.getPid());

c.setProduct(p);

orderItem.setCount(c.getCount());

System.***out***.println(p.getShop\_price());

System.***out***.println(c.getCount());

orderItem.setSubtotal(p.getShop\_price()\*c.getCount());

orderItem.setProduct(c.getProduct());

orderItem.setOrder(order);

order.getItems().add(orderItem);

**service**.addOrderItem(orderItem);

}

**for**(String a :arr){

**service1**.deletebycid(a);

}

List<Address> list=**new** ArrayList<>();

list=**service1**.getlist(loginUser);

mav.addObject(**"list"**,list);

mav.addObject(**"order"** , order);

mav.setViewName(**"order\_info"**);

**return** mav;

}

5.2后台实现部分：

5.2.1商品类型管理

管理员在登录管理系统之后，可以修改商品分类，添加商品分类，删除商品分类。



代码实现：

<**c:forEach var="c" items="${**list**}" varStatus="vst"**>

<**tr onmouseover="this**.style.**backgroundColor** = **'white'"**

**onmouseout="this**.style.**backgroundColor** = **'#F5FAFE'**;**"**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="18%"**>

*<%--分类的序号，从1开始 --%>*

**${**vst.count **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="17%"**>

**${**c.cname **}**

</**td**>

<**td align="center" style="HEIGHT**: 22**px"**>

<**a href="${** pageContext.request.contextPath **}/admin/editCategory.action?cid=${**c.cid**}"**>

<**i class="fa fa-pencil-square-o" aria-hidden="true"**></**i**>

</**a**>

</**td**>

<**td align="center" style="HEIGHT**: 22**px"**>

<**a href="${** pageContext.request.contextPath **}/admin/deleteCategory.action?cid=${**c.cid**}"**>

<**i class="fa fa-trash" aria-hidden="true"**></**i**>

</**a**>

</**td**>

</**tr**>

</**c:forEach**>

</**table**>

Java部分：

**public class** AdminCategroyController {

@Autowired

CategoryService **service**;

@RequestMapping(**"/addCategory"**)

**public** String addCategory() {

**return "admin/category/add"**;

}

@RequestMapping(**"/editCategory"**)

**public** ModelAndView editCategory(String cid) {

ModelAndView mav = **new** ModelAndView();

Category c = **service**.queryCategoryByCid(cid);

System.***out***.println(c);

mav.setViewName(**"admin/category/edit"**);

mav.addObject(**"category"**,c);

**return** mav;

}

@RequestMapping(**"/listCategory"**)

**public** ModelAndView listCategory() {

ModelAndView mav = **new** ModelAndView();

List<Category> list = **service**.listCategory();

mav.setViewName(**"admin/category/list"**); *// 要显示的视图*

mav.addObject(**"list"**, list); *// 交给视图显示的数据*

**return** mav;

}

@RequestMapping(**"/saveCategory"**)

**public** ModelAndView saveCategroy(Category c) {

ModelAndView mav = **new** ModelAndView();

c.setCid(UUIDUtil.*getUUId*());

System.***out***.println(c);

**service**.addCategory(c);

mav.addObject(**"msg"**, **"添加成功"**);

mav.setViewName(**"admin/welcome"**);

**return** mav;

}

@RequestMapping(**"/updateCategory"**)

**public** ModelAndView updateCategory(Category c) {

ModelAndView mav = **new** ModelAndView();

**service**.updateCategory(c);

mav.addObject(**"msg"**, **"修改成功"**);

mav.setViewName(**"admin/welcome"**);

**return** mav;

}

@RequestMapping(**"/deleteCategory"**)

**public** ModelAndView deleteCategory(Category c) {

ModelAndView mav = **new** ModelAndView();

**service**.deleteCategory(c);

mav.addObject(**"msg"**, **"删除成功"**);

mav.setViewName(**"admin/welcome"**);

**return** mav;

}

}

5.2.2商品管理





代码实现：

<**c:forEach items="${**page.list **}" var="p" varStatus="vs"**>

<**tr onmouseover="this**.style.**backgroundColor** = **'white'"**

**onmouseout="this**.style.**backgroundColor** = **'#F5FAFE'**;**"**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="18%"**>

**${**vs.count **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="17%"**>

<**img width="40" height="45" src="${** pageContext.request.contextPath **}/pic/${**p.pimage**}"**>

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="17%"**>

**${**p.pname **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="17%"**>

**${**p.shop\_price **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="17%"**>

<**c:if test="${**p.is\_hot==1 **}"**>是</**c:if**>

<**c:if test="${**p.is\_hot!=1 **}"**>否</**c:if**>

</**td**>

Java部分：

**public class** AdminProductController {

@Autowired

ProductService **service**;

@RequestMapping(**"/showAddProduct"**)

**public** String showAddProduct() {

**return "admin/product/add"**;

}

@RequestMapping(**"/listProduct"**)

**public** ModelAndView listProduct(**int** curPage) {

ModelAndView mav = **new** ModelAndView();

PageBean page = **service**.listProduct(curPage);

mav.setViewName(**"admin/product/list"**);

mav.addObject(**"page"**,page);

**return** mav;

}

@RequestMapping(**"/addProducts"**)

**public** ModelAndView addProduct(Product p,String cid,MultipartFile pictrueFile) **throws** IllegalStateException, IOException {

*// 设置图片名称，不能重复，使用 uuid*

String picName = UUIDUtil.*getUUId*();

*// 获取提交上来的文件名*

String oriName = pictrueFile.getOriginalFilename();

System.***out***.println(oriName);

System.***out***.println(23);

*// 通过文件名获取后缀名*

String extName = oriName.substring(oriName.lastIndexOf(**"."**));

*// 上传图片*

pictrueFile.transferTo(**new** File(**"C:\\Users\\Administrator\\Desktop\\C\\src\\main\\webapp\\pic\\"** + picName + extName));

System.***out***.print(picName);

p.setPid(UUIDUtil.*getUUId*());

p.setPimage(picName + extName);

Category c = **new** Category();

c.setCid(cid);

p.setCategory(c);

p.setPdate(**new** SimpleDateFormat(**"yyyy年MM月dd日 HH:mm:ss"**).format(**new** Date()));

ModelAndView mav = **new** ModelAndView();

**service**.addProduct(p);

mav.addObject(**"msg"**,**"添加商品成功"**);

mav.setViewName(**"admin/welcome"**);

**return** mav;

}

@RequestMapping(**"/updateProduct"**)

**public** ModelAndView updateProduct(MultipartFile pictrueFile,Product p)**throws** IllegalStateException ,IOException{

String picName = UUIDUtil.*getUUId*();

*// 获取提交上来的文件名*

String oriName = pictrueFile.getOriginalFilename();

*// 通过文件名获取后缀名*

String extName = oriName.substring(oriName.lastIndexOf(**"."**));

*// 上传图片*

pictrueFile.transferTo(**new** File(**"C:\\Users\\Administrator\\Desktop\\C\\src\\main\\webapp\\pic\\"** + picName + extName));

p.setPimage(picName + extName);

ModelAndView mav = **new** ModelAndView();

**service**.updateProduct(p);

mav.addObject(**"msg"**, **"修改成功"**);

mav.setViewName(**"admin/welcome"**);

**return** mav;

}

@RequestMapping(**"/editProduct"**)

**public** ModelAndView editProduct(String pid) {

ModelAndView mav = **new** ModelAndView();

Product p = **service**.queryProductByPid(pid);

mav.setViewName(**"admin/product/edit"**);

mav.addObject(**"product"**,p);

**return** mav;

}

@RequestMapping(**"/deleteProduct"**)

**public** ModelAndView deleteProduct(Product p) {

ModelAndView mav = **new** ModelAndView();

**service**.deleteProduct(p);

mav.addObject(**"msg"**, **"删除成功"**);

mav.setViewName(**"admin/welcome"**);

**return** mav;

}

}

5.2.3订单管理



代码实现：

<**c:forEach items="${**page.list **}" var="o" varStatus="vs"**>

<**tr**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="5%"**>

**${**vs.count **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.oid **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.total **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.address **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.telephone **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.name **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

<**c:if test="${**o.state == 0**}"**>

未付款

</**c:if**>

<**c:if test="${**o.state == 1 **}"**>

<**a href = "${**pageContext.request.contextPath **}/admin/utilOrder.action?oid=${**o.oid**}" id="sub"**>去发货</**a**>

</**c:if**>

<**c:if test="${**o.state == 2 **}"**>

已发货

</**c:if**>

<**c:if test="${**o.state== 3 **}"**>

订单完成

</**c:if**>

</**td**>

<**td align="center" style="HEIGHT**: 22**px"**>

<**input class="info" type="button" value="订单详情" oid='${**o.oid **}'**/>

</**td**>

<**td align="center" style="HEIGHT**: 22**px"**>

<**a href = "${**pageContext.request.contextPath **}/admin/deleteOrder.action?oid=${**o.oid**}"**>删除</**a**>

</**td**>

</**tr**>

</**c:forEach**>

<**c:if test="${**page.curPage != 1 **}"**>

<**li**><**a href="${**pageContext.request.contextPath **}/admin/listOrder.action?curPage=${**page.curPage - 1**}&state=${**param.state**}" aria-label="Previous"**><**span aria-hidden="true"**>**&laquo;**</**span**></**a**></**li**>

</**c:if**>

<**c:forEach begin="1" end="${**page.totalPage **}" var="i"**>

<**c:if test="${**page.curPage == i **}"**>

<**li class="active"**><**a**>**${**i **}**</**a**></**li**>

</**c:if**>

<**c:if test="${**page.curPage != i **}"**>

<**li**><**a href="${**pageContext.request.contextPath **}/admin/listOrder.action?curPage=${**i**}&state=${**param.state**}"**>**${**i **}**</**a**></**li**>

</**c:if**>

</**c:forEach**>

<**c:if test="${**page.curPage != page.totalPage **}"**>

<**li**>

<**a href="${**pageContext.request.contextPath **}/admin/listOrder.action?curPage=${**page.curPage + 1**}&state=${**param.state**}" aria-label="Next"**>

<**span aria-hidden="true"**>**&raquo;**</**span**>

</**a**>

</**li**>

</**c:if**>

</**ul**>

</**div**>

</**form**>

Java部分：

**public class** AdminOrderController {

@Autowired

OrderService **service**;

@Autowired

ProductService **service1**;

@RequestMapping(**"/listOrder"**)

**public** ModelAndView listOrder(**int** curPage) {

ModelAndView mav = **new** ModelAndView();

PageBean page = **service**.listOrder(curPage);

mav.setViewName(**"admin/order/list"**);

mav.addObject(**"page"**,page);

**return** mav;

}

@RequestMapping(**"/adminOrder"**)

@ResponseBody

**public** List<OrderItem> adminOrder(String oid){

**return service**.selectOrderItemByOid(oid);

}

@RequestMapping(**"/listOrderQita"**)

**public** ModelAndView listOrderyfk(**int** curPage,**int** state) {

ModelAndView mav = **new** ModelAndView();

PageBean page = **service**.listOrderQita(curPage , state);

System.***out***.println(page);

mav.setViewName(**"admin/order/list2"**);

mav.addObject(**"page"**,page);

**return** mav;

}

@RequestMapping(**"/utilOrder"**)

**public** ModelAndView utilOrder(String oid) {

ModelAndView mav = **new** ModelAndView();

Order order = **service**.selectOrderByOid(oid);

System.***out***.println(1);

List<OrderItem> orderItems=**service**.selectOrderItemByOid(oid);

System.***out***.println(orderItems);

**for**(OrderItem o:orderItems){

System.***out***.println(o.getCount());

**int** s=o.getProduct().getPcount()-o.getCount();

System.***out***.println(s);

o.getProduct().setPcount(s);

o.getProduct().setIs\_latest(o.getCount()+o.getProduct().getIs\_latest());

**service1**.updateProduct(o.getProduct());

}

**int** state = order.getState();

**if**(state != 3) {

state = state+1;

order.setState(state);

**service**.updateOrder(order);

}**else** {

order.setState(3);

**service**.updateOrder(order);

}

mav.addObject(**"msg"**,**"success"**);

mav.setViewName(**"admin/msg"**);

**return** mav;

}

@RequestMapping(**"/deleteOrder"**)

**public** ModelAndView deleteProduct(Order o) {

ModelAndView mav = **new** ModelAndView();

**service**.deleteOrder(o.getOid());

System.***out***.println(o);

mav.addObject(**"msg"**, **"删除成功"**);

mav.setViewName(**"admin/welcome"**);

**return** mav;

}

}

5.2.4公告管理



代码实现：

<**c:forEach items="${**list**}" var="o" varStatus="vs"**>

<**tr**>

<**td align="center" width="5%"**>**${**vs.count**}**</**td**>

<**td align="center" width="5%"**>**${**o.notice**}**</**td**>

<**td align="center" width="5%"**><**fmt:formatDate value="${**o.time**}" pattern="yyyy-MM-dd"**/></**td**>

<**td align="center" width="5%"**><**a href="${**pageContext.request.contextPath**}/admin/deletenotice.action?nid=${**o.nid**}"**>删除</**a**></**td**>

</**tr**>

</**c:forEach**>

</**table**>

Java部分：

**public** ModelAndView notice(){

List<notice> list= **service**.findnotice();

ModelAndView mav=**new** ModelAndView();

mav.setViewName(**"/admin/notice"**);

mav.addObject(**"list"** ,list);

**return** mav;

}

**public** ModelAndView deletenotice(String nid){

**service**.deletenotice(nid);

ModelAndView mav=**new** ModelAndView();

mav.setViewName(**"/admin/welcome"**);

mav.addObject(**"msg"** ,**"删除成功"**);

**return** mav;

}

5.2.5用户管理





代码实现：

<**tr**>

<**td width="18%" align="center" bgColor="#f5fafe" class="ta\_01"**>

用户名：

</**td**>

<**td class="ta\_01" bgColor="#ffffff"**>

<**input type="text" name="username" value="${**user.username**}" id="userAction\_save\_do\_logonName" class="bg" readonly**/>

</**td**>

<**td width="18%" align="center" bgColor="#f5fafe" class="ta\_01"**>

密码：

</**td**>

<**td class="ta\_01" bgColor="#ffffff"**>

<**input type="text" name="password" value="${**user.password**}" id="password" maxlength=6 Onkeyup="this**.**value**=**this**.**value**.replace(/[^0-9-]+/,**''**);**"**>

<**div id="ptext"**></**div**>

</**td**>

</**tr**>

<**tr**>

<**td width="18%" align="center" bgColor="#f5fafe" class="ta\_01"**>

姓名：

</**td**>

<**td class="ta\_01" bgColor="#ffffff"**>

<**input type="text" name="name" value="${**user.name**}" id="name" class="bg"**/>

<**div id="pname"**></**div**>

</**td**>

<**td width="18%" align="center" bgColor="#f5fafe" class="ta\_01"**>

邮箱：

</**td**>

<**td class="ta\_01" bgColor="#ffffff"**>

<**input type="text" name="email" value="${**user.email**}" id="email" class="bg"**/>

<**div id="pemail"**></**div**>

</**td**>

</**tr**>

<**tr**>

<**td width="18%" align="center" bgColor="#f5fafe" class="ta\_01"**>

电话：

</**td**>

<**td class="ta\_01" bgColor="#ffffff"**>

<**input type="text" name="telephone" value="${**user.telephone**}" id="phone" maxlength=11 Onkeyup="this**.**value**=**this**.**value**.replace(/[^0-9-]+/,**''**);**"**/>

<**div id="pphone"**></**div**>

</**td**>

<**td width="18%" align="center" bgColor="#f5fafe" class="ta\_01"**>

性别：

</**td**>

<**td class="ta\_01" bgColor="#ffffff"**>

<**input type="text" name="sex" id="sex" value="${**user.sex**}"**>

<**div id="psex"**></**div**>

</**td**>

</**tr**>

<**tr**>

<**td class="ta\_01" style="WIDTH**: 100%**" align="center"**

**bgColor="#f5fafe" colSpan="4"**>

<**button type="button" onclick="***m*()**" id="userAction\_save\_do\_submit" value="确定" class="button\_ok"**>

**&#30830;&#23450;**

</**button**>

<**FONT face="宋体"**>**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**</**FONT**>

<**INPUT class="button\_ok" type="button" onclick="*history***.go(-1)**" value="返回"**/>

<**span id="Label1"**></**span**>

</**td**>

</**tr**>

</**table**>

</**form**>

</**body**>

<**script**>

**$**(**"#password"**).blur(**function** () {

**var** reg=/^[0-9]{6}$/;

**if**(!reg.test(**$**(**"#password"**).val())){

**$**(**"#ptext"**).text(**"密码需要6个数字"**);

}**else** {

**$**(**"#ptext"**).text(**"密码可以使用"**);

}

})

**$**(**"#email"**).blur(**function** () {

**var** reg= /^([[A-Za-z0-9\_\-\.])+\@([A-Za-z0-9\_\-\.])+\.([A-Za-z]{2,4})$/;

**if**(!reg.test(**$**(**"#email"**).val())){

**$**(**"#pemail"**).text(**"邮箱格式不对"**);

}**else** {

**$**(**"#pemail"**).text(**"邮箱可以使用"**);

}

})

**$**(**"#phone"**).blur(**function** () {

**var** reg= /^1[3456789]\d{9}$/;

**if**(!reg.test(**$**(**"#phone"**).val())){

**$**(**"#pphone"**).text(**"电话格式不对"**);

}**else** {

**$**(**"#pphone"**).text(**"电话可以使用"**);

}

})

**$**(**"#name"**).blur(**function** () {

**if**(**$**(**"#name"**).val()!=**""**&&**$**(**"#name"**).val().trim()!=**""**){

**$**(**"#pname"**).text(**"姓名可以使用"**);

}**else** {

**$**(**"#pname"**).text(**"姓名不能为空"**);

}

})

**$**(**"#sex"**).blur(**function** () {

**if**(**$**(**"#sex"**).val()==**"男"**||**$**(**"#sex"**).val()==**"女"**){

**$**(**"#psex"**).text(**"性别可以使用"**);

}**else** {

**$**(**"#psex"**).text(**"性别只能为男或女"**);

}

})

**function** *m*() {

**if**(**$**(**"#ptext"**).text()==**"密码可以使用"**&&**$**(**"#pemail"**).text()==**"邮箱可以使用"**&&**$**(**"#pphone"**).text()==**"电话可以使用"**&&**$**(**"#pname"**).text()==**"姓名可以使用"**&&(**$**(**"#sex"**).val()==**"男"**||**$**(**"#sex"**).val()==**"女"**)){

**$**(**"#userAction\_save\_do"**).submit();

}**else** {

*alert*(**"请检查所有项符合要求且不为空！"**)

}

}

***window***.**onload**=**function** () {

**function** *x*() {

**var** reg=/^[0-9]{6}$/;

**if**(!reg.test(**$**(**"#password"**).val())){

**$**(**"#ptext"**).text(**"密码需要6个数字"**);

}**else** {

**$**(**"#ptext"**).text(**"密码可以使用"**);

}

}

**function** *x1*() {

**var** reg= /^([[A-Za-z0-9\_\-\.])+\@([A-Za-z0-9\_\-\.])+\.([A-Za-z]{2,4})$/;

**if**(!reg.test(**$**(**"#email"**).val())){

**$**(**"#pemail"**).text(**"邮箱格式不对"**);

}**else** {

**$**(**"#pemail"**).text(**"邮箱可以使用"**);

}

}

**function** *x2*() {

**var** reg= /^1[3456789]\d{9}$/;

**if**(!reg.test(**$**(**"#phone"**).val())){

**$**(**"#pphone"**).text(**"电话格式不对"**);

}**else** {

**$**(**"#pphone"**).text(**"电话可以使用"**);

}

}

**function** *x3*() {

**if**(**$**(**"#name"**).val()!=**""**&&**$**(**"#name"**).val().trim()!=**""**){

**$**(**"#pname"**).text(**"姓名可以使用"**);

}**else** {

**$**(**"#pname"**).text(**"姓名不能为空"**);

}

}

**function** *x4* () {

**if**(**$**(**"#sex"**).val()==**"男"**||**$**(**"#sex"**).val()==**"女"**){

**$**(**"#psex"**).text(**"性别可以使用"**);

}**else** {

**$**(**"#psex"**).text(**"性别只能为男或女"**);

}

}

*x*();

*x1*();

*x2*();

*x3*();

*x4*();

}

</**script**>

<**form id="Form1" name="Form1" method="post"**>

<**table cellSpacing="1" cellPadding="0" width="80%" align="center" bgColor="#808080" border="0"**>

<**TBODY**>

<**tr**>

<**td class="ta\_01" align="center" bgColor="gray"**>

<**strong**>用户列表</**strong**>

</**TD**>

</**tr**>

<**c:forEach items="${**page.list **}" var="o" varStatus="vs"**>

<**tr**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.username **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.password**}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.name **}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="17%"**>

**${**o.telephone**}**

</**td**>

<**td style="CURSOR**: **hand**; **HEIGHT**: 22**px" align="center"**

**width="10%"**>

**${**o.email**}**

</**td**>

<**td align="center" style="HEIGHT**: 22**px"**>

**${**o.sex**}**

</**td**>

<**td align="center" style="HEIGHT**: 22**px"**>

**${**o.birthday**}**

</**td**>

<**td align="center" style="HEIGHT**: 22**px"**>

<**a href = "${**pageContext.request.contextPath **}/admin/updateuser.action?uid=${**o.uid**}"**>

<**img src="${**pageContext.request.contextPath**}/images/i\_edit.gif" border="0" style="CURSOR**: **hand"**>

</**a**>

</**td**>

<**td align="center" style="HEIGHT**: 22**px"**>

<**a href = "${**pageContext.request.contextPath **}/admin/deleteuser.action?uid=${**o.uid**}"**>

<**img src="${**pageContext.request.contextPath**}/images/i\_del.gif" width="16" height="16" border="0" style="CURSOR**: **hand"**>

</**a**>

</**td**>

</**tr**>

</**c:forEach**>

</**table**>

</**td**>

</**tr**>

</**TBODY**>

</**table**>

<**div style="text-align**: **center**;**"**>

<**ul class="pagination"**>

<**c:if test="${**page.curPage != 1 **}"**>

<**li**><**a href="${**pageContext.request.contextPath **}/admin/getuser.action?curPage=${**page.curPage - 1**}" aria-label="Previous"**><**span aria-hidden="true"**>**&laquo;**</**span**></**a**></**li**>

</**c:if**>

<**c:forEach begin="1" end="${**page.totalPage **}" var="i"**>

<**c:if test="${**page.curPage == i **}"**>

<**li class="active"**><**a**>**${**i **}**</**a**></**li**>

</**c:if**>

<**c:if test="${**page.curPage != i **}"**>

<**li**><**a href="${**pageContext.request.contextPath **}/admin/getuser.action?curPage=${**page.curPage - 1**}"**>**${**i **}**</**a**></**li**>

</**c:if**>

</**c:forEach**>

<**c:if test="${**page.curPage != page.totalPage **}"**>

<**li**>

<**a href="${**pageContext.request.contextPath **}/admin/getuser.action?curPage=${**page.curPage - 1**}" aria-label="Next"**>

<**span aria-hidden="true"**>**&raquo;**</**span**>

</**a**>

</**li**>

</**c:if**>

</**ul**>

</**div**>

</**body**>

Java部分：

**public** ModelAndView getuser(**int** curPage){

ModelAndView mav=**new** ModelAndView();

PageBean page =**service**.listUser( curPage);

mav.addObject(**"page"**,page);

mav.setViewName(**"admin/usermanage/messageuser"**);

**return** mav;

}

**public** ModelAndView deleteuser(String uid){

ModelAndView mav=**new** ModelAndView();

**service**.deleteuser(uid);

mav.addObject(**"msg"**, **"删除成功"**);

mav.setViewName(**"admin/welcome"**);

**return** mav;

}

**public** ModelAndView updateuser(String uid){

ModelAndView mav=**new** ModelAndView();

User user =**service**.finduser(uid);

mav.addObject(**"user"**, user);

mav.setViewName(**"admin/usermanage/edit"**);

**return** mav;

}

5.3对数据库的访问代码实现

<**bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"**>  
 *<!-- 匹配属性 -->* <**property name="driverClassName" value = "com.mysql.jdbc.Driver"**></**property**>  
 <**property name="url" value = "jdbc:mysql://localhost:3306/store3?characterEncoding=utf-8"**></**property**>  
 <**property name="username" value= "root"**></**property**>  
 <**property name="password" value= "root"**></**property**>  
   
</**bean**>

对数据库的访问主要是借助applicationContext.xml配置文件实现，在项目启动时，创建bean实例,连接数据库，这样就使得开发者关注sql本身，而不需要花费精力去处理数据库驱动，创建连接，关闭连接等操作，大大提高开发效率。

5.3系统测试

5.3.1软件测试

软件测试是为了发现系统存在的bug或者错误。目的是为了在正式上线之前，尽可能的发现并解决排除软件中潜藏的错误或者bug，以此来提高软件的可用性，此部分将采用黑盒测试对系统进行单元测试和综合测试，合适的选取测试用例务求最大程度覆盖此系统的方方面面，降低系统出现错误的可能性。

测试方法：黑盒测试

选择测试用例的原则

测试用例包括被测试的功能，因输入的数据和预期的输出结果，测试所用的数据选用少量，有效的测试数据进行测试，尽可能覆盖所有方面。基本目标是：设计一组数据，比较与预期结果是否相同，此测试用例应包含的方面：

1. 正确性测试：用户输入的数据是否能被系统正确处理。测试用例中的测试点至少要保证功能正常。
2. 容错性测试：当用户输入非法数据（非法类型，不符合要求的数据）时，系统应能给出提示并进行相应的处理。当用户输入正确的数据，系统给出的结果应该与预期结果相同。
3. 完整性测试：对没有经过系统验证的人使用软件系统的企图，程序的数据处理能够保持数据库文件的完整性。
4. 接口之间的测试，测试各个模块之间的数据交互，保证数据输入输出的一致性和正确性。
5. 压力测试：输入10条数据运行各个功能，输入30条数据运行，输入100条数据运行测试。
6. 边界值分析法：确定边界情况（刚好等于，稍小于，大于），针对系统在测试过程中主要输入一些合法或者非法的数据，主要在边界值附件取
7. 错误推测：根据测试的经验和感觉，参照以往的系统出现错误之处设计测试用例
8. 效率：完成预定的功能，系统运行的时间
9. 回归测试，按照测试用例将所有的测试点测试完毕，并且解决发现的问题，再测试

5.3.2测试内容

1. 主页测试

在浏览器地址栏输入<http://localhost:8080/C/>



1. 注册账号测试



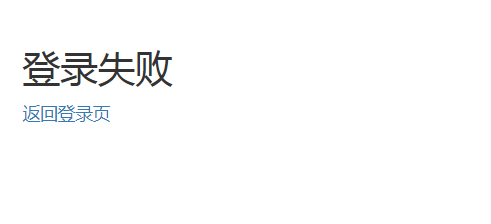




只要当所有项符合要求时，才会注册成功，当鼠标焦点失去的时候会自动判断输入是否符合。

1. 登录测试





如果账号密码匹配的话，就会登陆成功并进入首页。不匹配的话会显示登录失败。

1. 购物车功能测试





刚加入购物车的商品是没有被选中的，所以购物车的总金额为0，只会计算被选中商品的价格，可实现单选，全选，多选，删除商品等功能。

**5.3.3测试计划执行情况**

|  |  |  |  |
| --- | --- | --- | --- |
| **测试类型** | **测试内容** | **测试目的** | **测试方法** |
| **功能测试** | **前台用户注册登录，后台管理员登录** | **确定所有相关功能正确与否** | **黑盒测试法** |
| **功能测试** | **商品的浏览，查询，加入购物车，以及购物车操作，订单功能** | **确定功能是否正常，数据是否正确** | **边界值法，等价类法** |
| **功能测试** | **后台管理** | **增删改查是否正确，是否能正确走预期的过程** | **手动测试** |
| **用户界面测试** | **导航，连接，颜色等** | **页面风格是否合理，一致** | **手工测试** |
| **安全测试** | **密码，超时** | **系统是否安全** | **黑盒法，手动测试** |

**5.3.4测试效果**

**测试计划是对系统的功能进行的测试，测试范围包括了系统所有的功能，并且对相关功能的按钮做了多次测试，但是，测试计划不包含没有被发现的问题，所以当系统通过测试用例，并进行修改之后，再进行多次测试，系统无错误就认为通过测试。**