**引言**

目前，网络正以一种前所未有的冲击力在影响着人类的生活，包括人类的生产和日常生活。网络的诞生和发展颠覆了传统信息传播方式，冲破了存在于传统交流方式中时间和空间的种种壁垒，极大地改变了人类从物质到精神，从形式到内容，从生产到生活的各种活动，并且给人类带来了新的机遇和挑战。

2006年1月17日，中国互联网络信息中心在京发布“第十七次中国互联网络发展状况统计报告”。报告显示，截至2005年12月12月31日，我国上网用户总数突破1亿，为1.11亿人，其中宽带上网人数达到6430万人。目前，我国网民数和宽带上网人数位居世界第二。国家顶级域名CN注册量首次突破百万，达到109万，成为国内用户注册域名的首选，稳居亚洲第一。

而本次毕业设计的任务是，构建一个电子商务网站，采用的主要技术是SSM+bootstrap+jsp;

jsp是Java server page 的缩写是由sun Microsystem公司倡导，许多公司参与建立一种动态网页技术，他在动态网页的建设中有强大而特别的功能，jsp是一种服务器端HTML嵌入代码的脚本语言是开发web动态网站快速而有效的工具，jsp的技术优势：

Jsp技术是用Java语言作为脚本语言的。Java语言是成熟的强大的，已扩充的变成语言。

高效性与安全性，jsp在执行前先被编译成字节码，字节码由Java虚拟机解释执行，比源码解释效率高。

可维护性。由于jsp技术是一种开放的，跨平台的结构，因此web服务器，平台及其他组件很容易升级或切换，且不会影响jsp基本的应用程序。

产品的多样性。目前，jsp+servlet+jdbc+javaBean目前已经成为开发电子商务平台的主流技术。随着国内与国际的接轨，jsp必将成为网站开发技术的首选。

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。极大的方面了用户。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世纪全球进入一个新经济时代，[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，产品和服务几乎可以瞬时在世界各地交易，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。

自我国加入WTO后，传统企业发展的必然趋势是网络化、全球化，而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，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,所以电子购物网站的实现是一个非常迫切的需求。

当今的企业，如果仍然依靠传统的方式宣传自己企业的形象和文化、采购自己企业所需的原材料和配套部件、推销自己企业的产品，则显得十分落后而且难以适应激烈的市场竞争。因此，当代企业纷纷建立自己的网站，借助于互联网快速的信息传递加速自身的发展。一个企业网站的规划与建设，自然而然也就成了众多类型的网站规划与建设的典范；企业网站的规划与建设这一研究课题，也自然而然地成为当今乃至今后一段长时间内，企业内部和互联网上的一个相当重要的讨论对象。我这次毕业设计主要研究的就是网上商城在网络上的实现。

**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。借助日益发展的科学技术，无店铺零售通常能够省却从生产到销售过程中的许多环节，从而达到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]。

鉴于互联网的优势以及对国内外相关电子商务网站现状的研究分析，本设计以基于Web的商城电子商务网站开发作为我的毕业设计主题。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**

随着信息技术广泛、深入地应用到人类社会的各个领域并发挥着越来越重要的作用。计算机软件技术应用于信息管理是发展的必然。作为计算机应用的一部分，充分发挥计算机的优势，将大量复杂的数据交给计算机来处理，有着手工管理所无法比拟的优点。如：查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点能够极大地提高信息管理的效率，是真正意义上的合理利用资源，也是企事业管理科学化、正规化,与世界接轨的重要条件。

**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，系统能实现用户的注册功能、登录功能、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，该设计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）前台用户功能模块主要是实现了用户对公告、商品的浏览以及查询，在注册登录后，可以实现物品加入购物车和购买等功能。

（2）后台管理模块主要实现了管理员对公告的修改，商品的添加删除，用户的订单管理等功能

（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：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提交订单

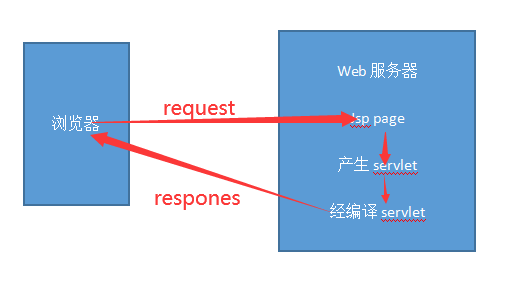
（5）管理员模块：登录管理员账户之后可以对商品进行各种操作，以及对公告的修改，对订单的处理

**第二章 开发环境及开发工具**

**1.1 开发技术简介**

**1.1.1 JSP简介**

JSP即Java Server Page，是Sun公司推出的动态网页技术标准



JSP的特点：

1. 一次编写，到处运行。除了系统之外，代码不用做任何更改。
2. 系统的多平台支持。基本上可以在所有平台上的任意环境中开发，在任意环境中进行系统部署，在任意环境中扩展。相比ASP的局限性JSP的优势是显而易见的。 （3）强大的可伸缩性。从只有一个小的Jar文件就可以运行Servlet/JSP，到由多台服务器进行集群和负载均衡，到多台Application进行事务处理，消息处理，一台服务器到无数台服务器，Java显示了一个巨大的生命力。
3. 多样化和功能强大的开发工具支持。这一点与ASP很像，Java已经有了许多非常优秀的开发工具，而且许多可以免费得到，并且其中许多已经可以顺利的运行于多种平台之下。

(5)支持服务器端组件。web应用需要强大的服务器端组件来支持，开发人员需要利用其他工具设计实现复杂功能的组件供web页面调用，以增强系统性能。JSP可以使用成熟的JAVA BEANS 组件来实现复杂商务功能。

**1.1.2 Servlet简介**

Servlet是在服务器上运行的小程序。这个词是在Java applet的环境中创造的，Java applet是一种当作单独文件跟网页一起发送的小程序，它通常用于在客户端运行，结果得到为用户进行运算或者根据用户互作用定位图形等服务。

Servlet的主要功能在于交互式的浏览和修改数据，生成动态Web内容。

它的执行过程：

客户端发生请求到服务端；

服务端讲请求信息发送到Servlet；

Servlet生成响应内容并将其传给服务器，响应内容动态生成，通常取决于客户端的请求；

服务器将响应返回给客户端；

Servlet的优点：

（1）功能强大

在Servlet中，许多使用传统CGI程序很难完成的任务都可以轻松地完成。例如，Servlet能够直接和Web服务器交互，而普通的CGI程序不能。Servlet还能够在各个程序之间共享数据，使得数据库连接池之类的功能很容易实现。

1. 可移植性好

Servlet用Java编写，ServletAPI具有完善的标准。因此，为IPlanet Enterprise Server写的Servlet无需任何实质上的改动即可移植到Apache、MicrosoftIIS或者WebStar。几乎所有的主流服务器都直接或通过插件支持Servlet。

1. 节省投资

不仅有许多廉价甚至免费的Web服务器可供个人或小规模网站使用，而且对于现有的服务器，如果它不支持Servlet的话，要加上这部分功能也往往是免费的(或只需要极少的投资)。

**1.1.3 MySql简介**

MySQL名字的来历：

MySQL是一个小型关系型数据库管理系统MySQL被广泛地应用在Internet上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库。

MySQL的特性：

（1）使用C和C++编写，并使用了多种[编译器](http://baike.baidu.com/view/487018.htm" \t "_blank)进行测试，保证源代码的可移植性；

（2）支持AIX、[FreeBSD](http://baike.baidu.com/view/21459.htm" \t "_blank)、HP-UX、[Linux](http://baike.baidu.com/view/1634.htm" \t "_blank)、Mac OS、[Novell](http://baike.baidu.com/view/68826.htm" \t "_blank) Netware、OpenBSD、OS/2 Wrap、Solaris、Windows等多种操作系统；

（3）为多种编程语言提供了API。这些编程语言包括[C](http://baike.baidu.com/view/10075.htm" \t "_blank)、C++、[Python](http://baike.baidu.com/view/21087.htm" \t "_blank)、Java、[Perl](http://baike.baidu.com/view/46614.htm" \t "_blank)、[PHP](http://baike.baidu.com/view/99.htm" \t "_blank)、Eiffel、Ruby和Tcl等；

（4）支持[多线程](http://baike.baidu.com/view/65706.htm" \t "_blank)，充分利用CPU资源；

（5）优化的[SQL](http://baike.baidu.com/view/34.htm" \t "_blank)查询算法，有效地提高查询速度；

（6）既能够作为一个单独的应用程序应用在客户端服务器网络环境中，也能够作为一个库而嵌入到其他的软件中提供多语言支持，常见的[编码](http://baike.baidu.com/view/237708.htm" \t "_blank)如中文的GB 2312、BIG5，日文的Shift\_JIS等都可以用作数据表名和数据列名；

（7）提供TCP/IP、ODBC和JDBC等多种数据库连接途径；

（8）提供用于管理、检查、优化数据库操作的管理工具；

（9）可以处理拥有上千万条记录的大型数据库。

**1.1.4 IDEA简介**

IDEA 全称 IntelliJ IDEA，是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具，尤其在智能代码助手、代码自动提示、重构、J2EE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。

(1)JUnit的完美支持

(2)xml全提示支持：所有流行框架的xml文件都支持全提示，

(3)不需要任何的插件，完全支持JSP。

(4)需要任何插件完全支持EJB(6.0 支持EJB3.0)

(5)集成了市面上常见的所有版本控制工具插件，包括git、svn、github，让开发人员在编程的工程中直接在intellij idea里就能完成代码的提交、检出、解决冲突、查看版本控制服务器内容等等。

**1.2 开发环境配置**

**1.2.1 JDK的安装与配置**

（1）下载JDK并安装。

（2）进行JDK的环境变量的配置：

第一步:右击“我的电脑”，点击“属性”：

第二步：选择“高级”选项卡，点击“环境变量”：

第三步：在“系统变量”中，设置3项属性，

JAVA\_HOME,PATH,CLASSPATH,若已存在则点击“编辑”，不存在则点击“新建”：

第四步：JAVA\_HOME设为JDK的安装路径(如C:\Program

Files\Java\jdk1.6.0\_10)，此路径下包括lib，bin，jre等文件夹；

Path使得系统可以在任何路径下识别java命令，设为：

%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin

CLASSPATH为java加载类(class or lib)路径，只有类在classpath中，java命令才能识别，设为：.;%JAVA\_HOME%\lib;

设置完成后在cmd中输入java -version 出现java的版本信息，证明jdk安装成功。

**1.2.2 Tomcat的安装与配置**

（1）Tomcat简介

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

Tomcat 是一个小型的轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好Apache 服务器，可利用它响应对HTML 页面的访问请求。实际上Tomcat 部分是Apache 服务器的扩展，但它是独立运行的，所以当你运行tomcat 时，它实际上作为一个与Apache 独立的进程单独运行的。

（2）Tomcat的安装

下载并安装tomcat7.0。进入 \bin目录，双击startup.bat执行文件后启动Tomcat服务器，打开浏览器，键入http://localhost:8080，如果成功进入到Tomcat服务器的主页面，则表示安装成功。

**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，而他的网络编程语言jsp更加便利，jsp技术使用Java编程语言编写类XML的tags来封装动态网页的处理逻辑。Jsp将网页逻辑和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**

该系统是基于B/S(浏览器/服务器)架构的系统，主要完成商品的浏览，购买，以及在后台管理商品；公告的添加、删除、修改（更新），会员的在线购物，此系统分为前台购物和后台管理。前台购物是友好的操作界面，供用户注册、浏览、订购商品；后台管理是提供给管理员的，其中包括：商品管理、用户管理、公告管理等。使管理员从繁琐的手工操作中解脱出来，并提高了工作效率。

按照系统功能的实现，本系统可以划分为以下几个功能模块：管理用户模块，管理商品模块，管理商品类别模块等。管理用户模块包括了修改用户信息，修改用户密码以及删除用户。管理商品模块包括商品上传，修改商品信息，删除商品，增加商品图片。管理商品类别模块包括添加，修改，删除商品类别等功能。该系统对可靠性、易维护性、安全性、可操作性等性能有较高的要求。可靠性----要求系统在发生故障或输入数据不合理等情况下有较高的要求；易维护性---系统的变更（因系统需求变化和弥补系统缺陷而引起）要简单易行；安全性---本系统所处理的数据都要具有实际意义，不能随意存取和改动。因此，必须对系统数据的存取和改动进行控制，对系统数据进行有效的保护，以杜绝对数据的非法操作和防止计算机病毒的破坏；可操作性----本系统的操作人员是一般管理人员而非计算机专业人员，为使本系统有效地发挥作用，要求本系统容易理解，人机界面简明、清晰、直观，功能实用，操作简单方便，上机培训量小。此外，系统还需有较快的响应速度，以尽量减少工作人员的等待时间。

其基本功能是，用户能够通过该系统浏览商品信息，以及进行购物，而管理员要能对网站做必要的管理，因此必须要有针对用户和管理员的前台用户操作功能和后台管理员操作功能。

（1）普通用户可以通过系统浏览商品；

可以查看所有的商品及类别。

（2）普通用户可以通过系统搜索商品

（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实现更多功能。

（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；

（7）会员可以购买商品；

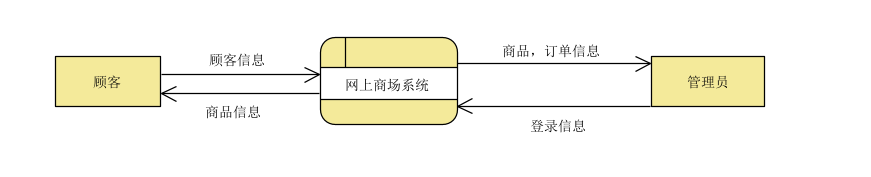
（8）会员可以浏览并且管理自己的订单；

（9）会员可以管理自己的信息：

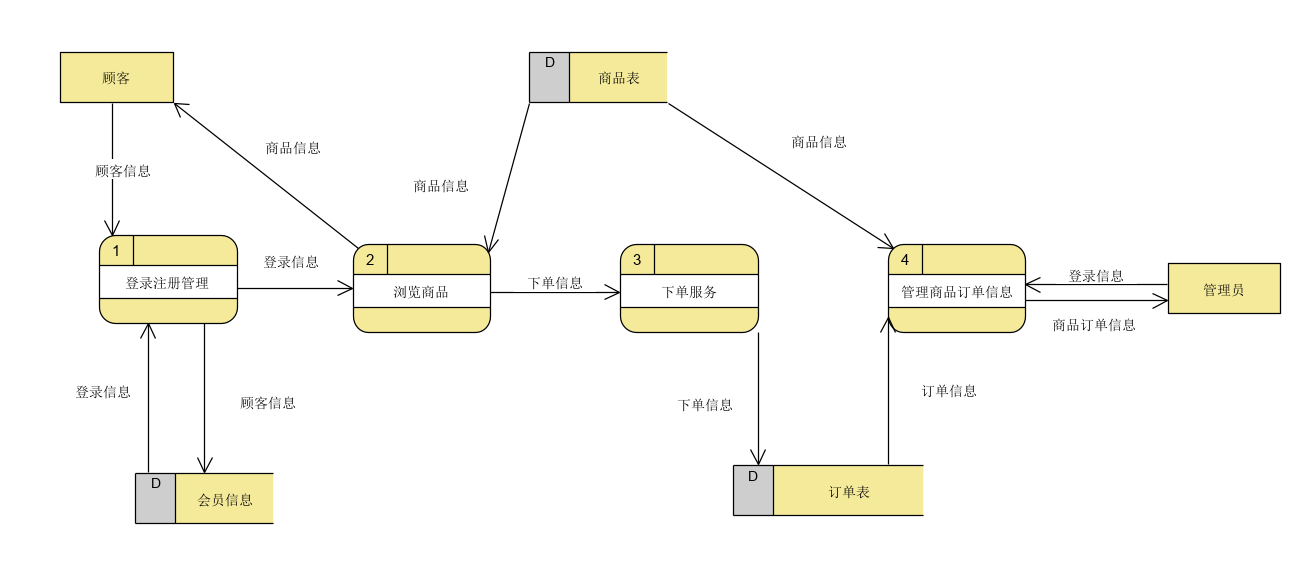
（i）修改密码

**3.3 结构化分析**

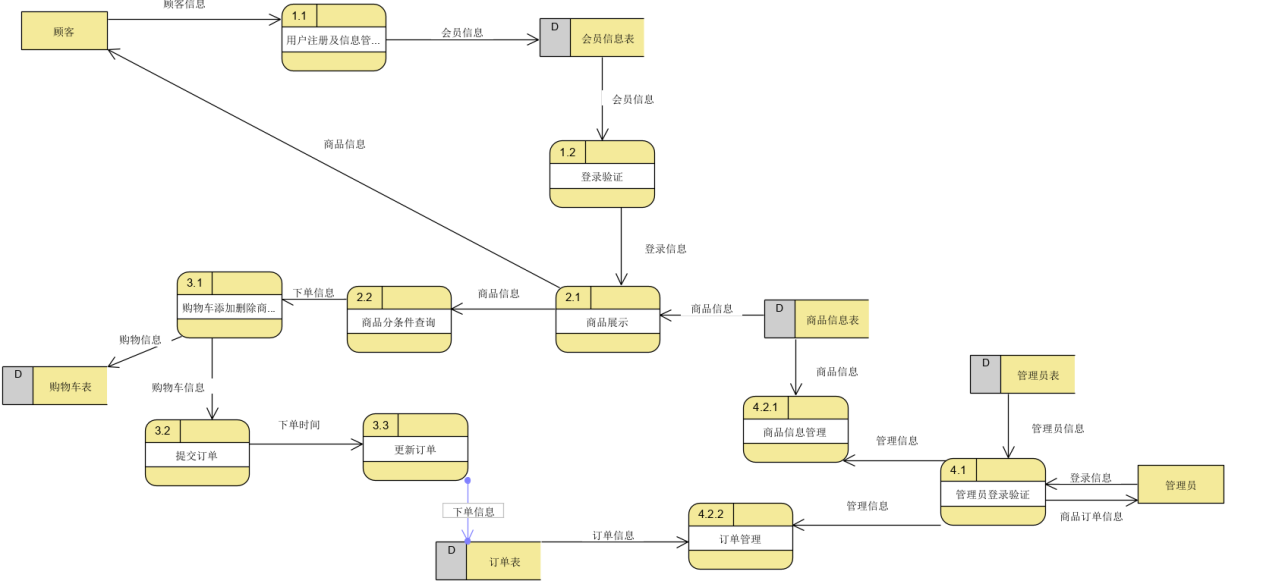
**3.3.1 数据流图**



顶层图



一层数据流图



二层数据流图

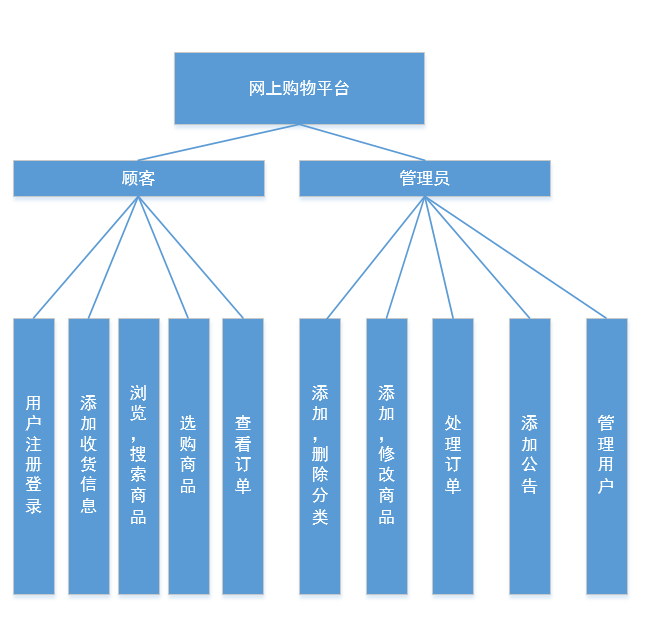
**3.3.2 数据字典**

1. **系统设计**

**4.1 系统模块设计**

在结构化设计中，模块一般都是按功能划分的通常称为功能模块。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，即前台模块和后台模块。



**4.1.1前台模块的功能模块设计**

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，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。不但可以随时添加查看修改清空购物车中的内容，还可以随时去收银台结账。

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

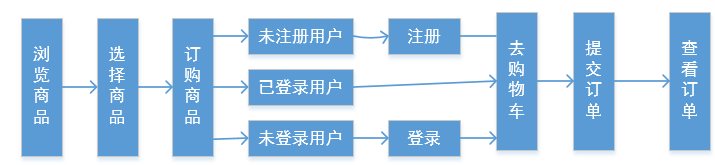
1. 公告管理模块

管理系统公告。

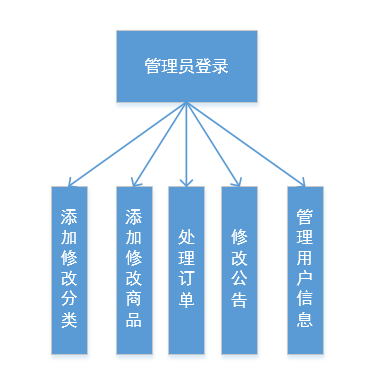
**4.2 系统流程设计**

**本系统分为前台购物流程和后台订单处理流程。**

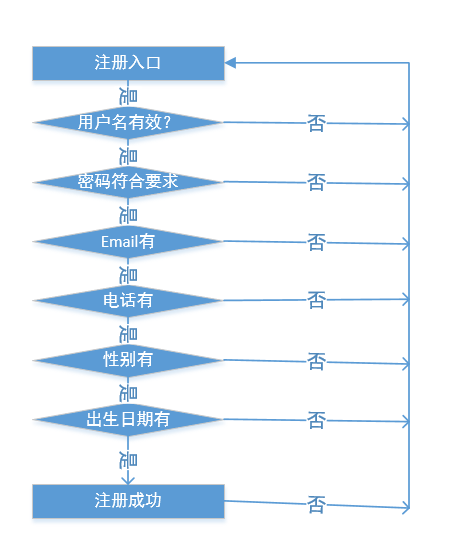
**前台购物流程图**



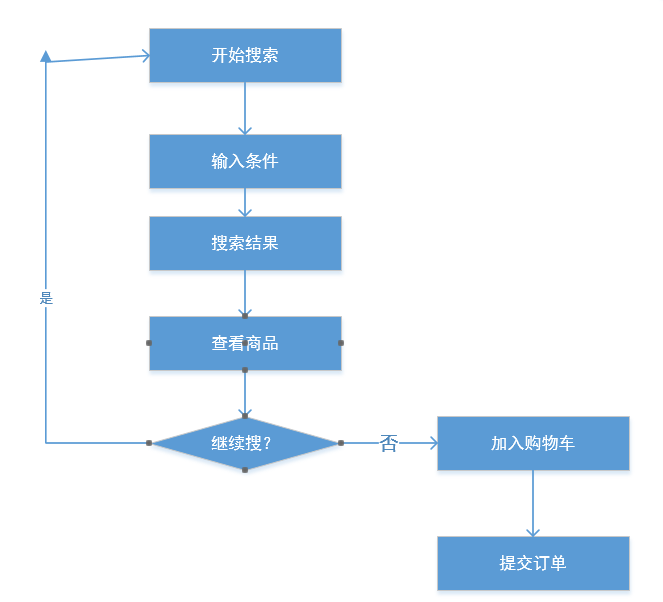
后台管理流程图



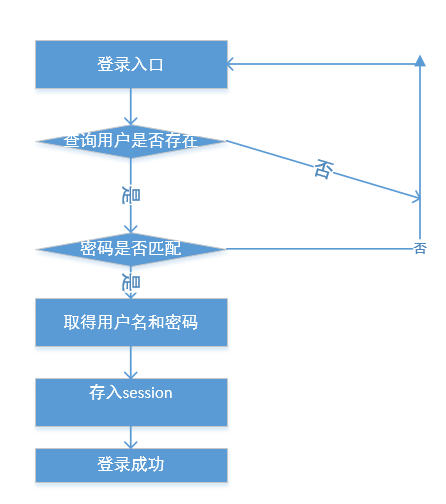
注册功能流程图



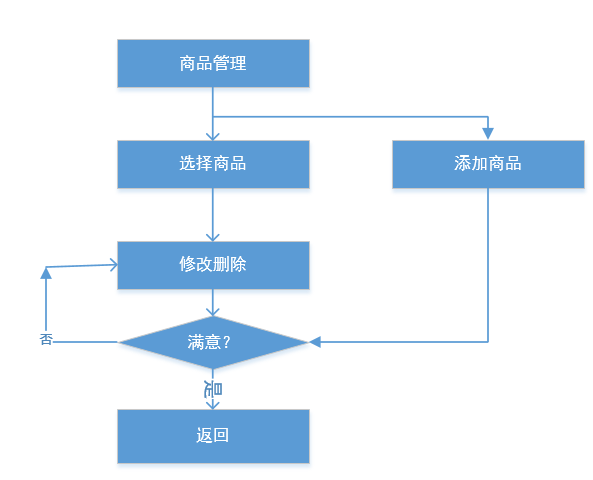
商品搜索流程图



登录流程图



商品管理流程图



**4.3 数据库设计**

**4.3.1数据库概要设计**

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。一个会员可以有多个订单，一个订单只能被一个会员所有

数据库的概念设计：

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。

1.模型中的实体相当于实体集、一个表，而不是单个实体或表中的一行。实体用矩形框表示，实体名称标注在矩形框内。用菱形表示实体间的联系，菱形框内写上联系名，用无向边把菱形分别与有关实体相连接，在无向边旁标上联系的类型。

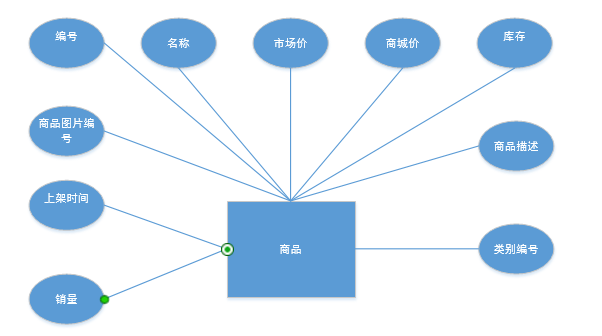
2.属性是实体的性质。用椭圆框表示，与实体之间用一条线相连表的主码是关键属性，关键属性项加下划线。

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。

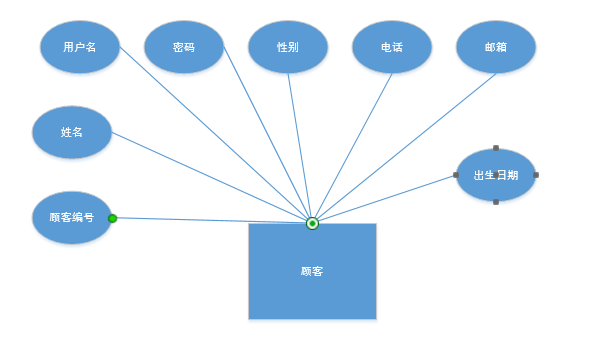
4.在程序中实现对他们的完整性和一致性控制。

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：

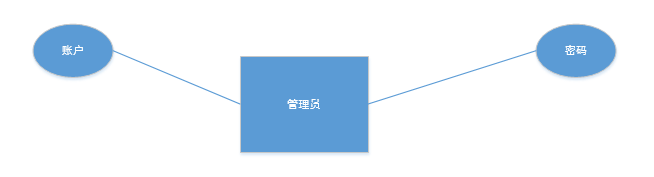
说明：图中用矩形表示实体。实体之间的关系用菱形表示，用无向边把菱形与有关实体连接，在边上标明联系的类型。实体的属性用椭圆表示，并用无向边把实体与属性联系起来。



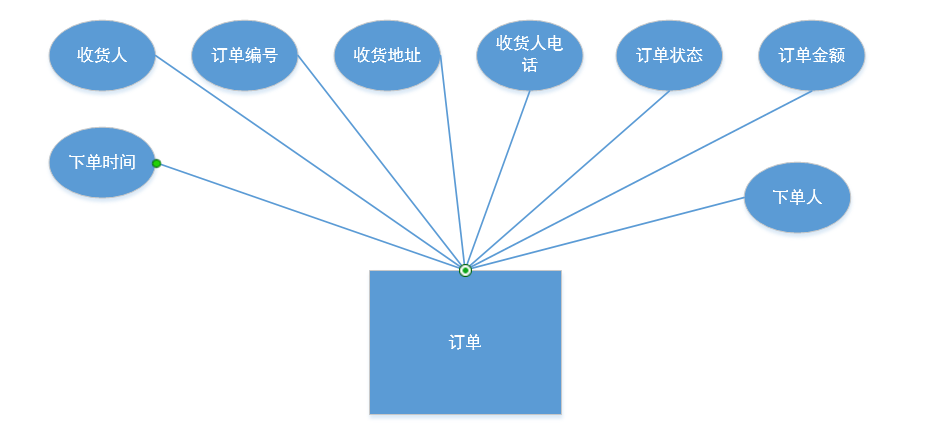
商品实体属性图



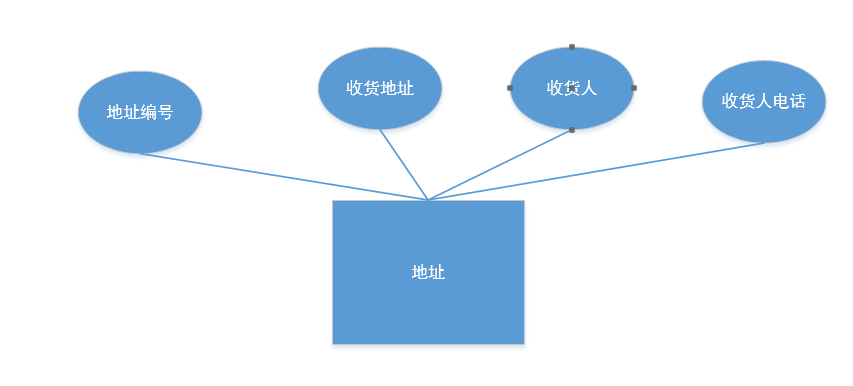
顾客实体属性图



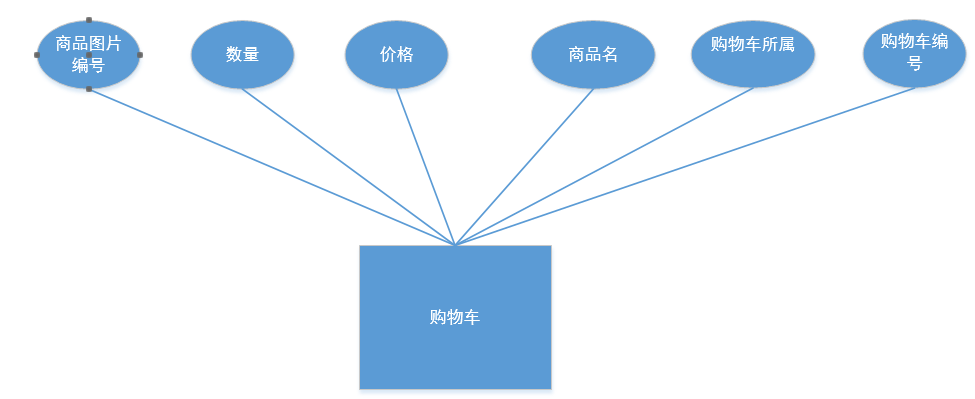
管理员实体属性图



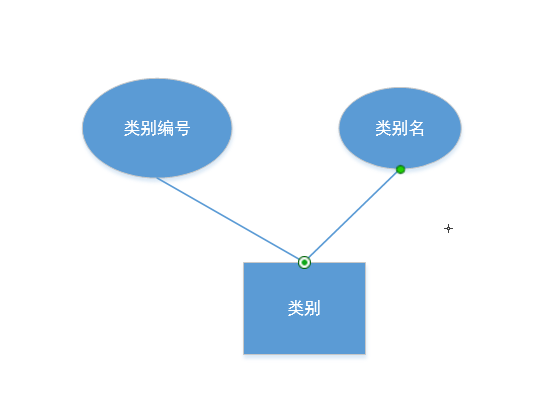
订单实体属性图



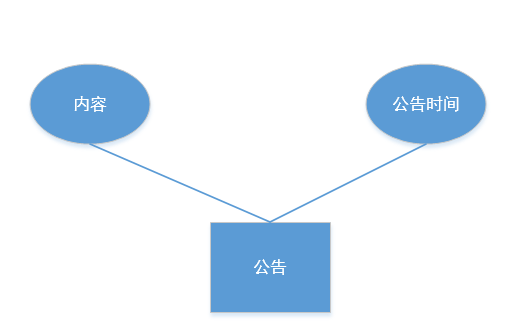
收货地址实体属性图



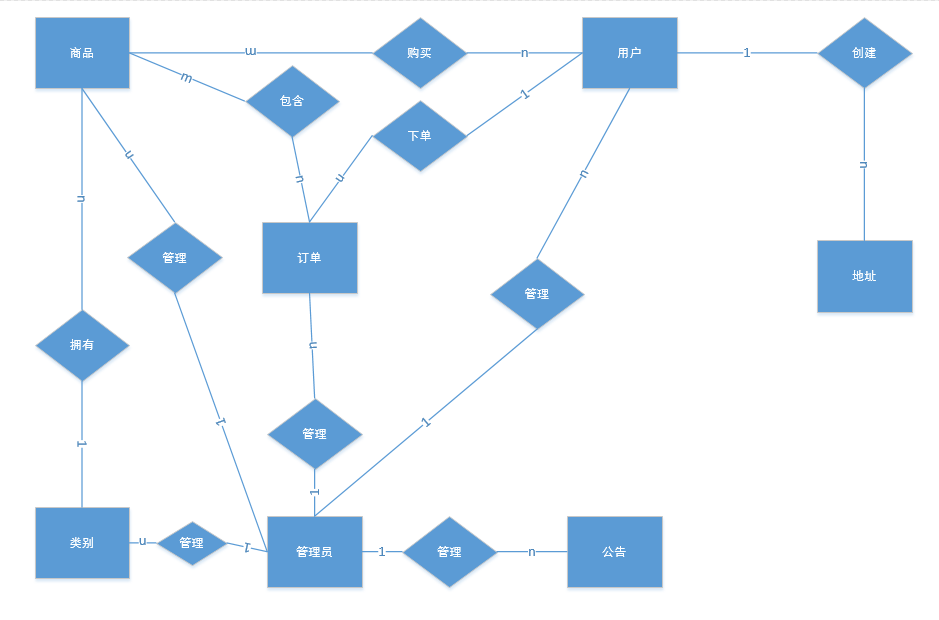
购物车实体属性图



商品类别实体属性图



系统公告实体属性图



ER图

**4.3.2数据库逻辑设计**

数据库技术是信息资源管理最有效的手段。数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，有效存储数据，满足用户信息要求和处理要求。

数据库设计中需求分析阶段综合各个用户的应用需求，在概念设计阶段形成独立于机器特点、独立于各个DBMS产品的概念模式，用E-R图来描述。在逻辑设计阶段将E-R图转换成具体的数据库产品支持的数据模型如关系模型，形成数据库逻辑模式。然后根据用户处理的要求，安全性的考虑，在基本表的基础上再建立必要的视图形成数据的外模式。在物理设计阶段根据DBMS特点和处理的需要，进行物理存储安排，设计索引，形成数据库内模式。

数据库设计的主要任务是针对一个给定的应用环境，在给定的硬件环境、操作系统、以及数据库管理信息系统等软件环境下，创建一个性能良好的数据库模式，建立数据库及其应用系统，使之能有效地收集、存储、操作和管理数据，满足用户地各类需求。数据库设计通常是在一个通用的DBMS支持下，即利用现成的DBMS作为开发的基础。

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 |  | 属于哪个类型的商品 |

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 | usernameid | varchar | 255 |  | 不 |  | 订单拥有人 |

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 |  | id |
| 2 | count | int | 11 |  | 不 |  | 数量 |
| 3 | subtotal | double | 0 |  | 不 |  | 小计 |
| 4 | oid | varchar | 255 |  | 不 |  | 订单号 |
| 5 | pid | varchar | 255 |  | 不 |  | 商品id |

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 |  | 用户名 |
| 3 | Pid | varchar | 255 |  | 不 |  | 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 |

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。主要包括网站首页、查看公告，产品展示，用户登录注册，购买商品，查看订单等几个部分。由于篇幅有限，本文只对部分模块的实现进行描述。

**5.1.1用户注册登录模块**



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** () {

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

**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"**){

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

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

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

}

}

})

}**else**{

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

}

})

**function** *m*() {

**if**(**$**(**"#accountCheck"**).text()==**"该用户名可以使用"**&&**$**(**"#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。会员登录窗口的运行结果如图所示。

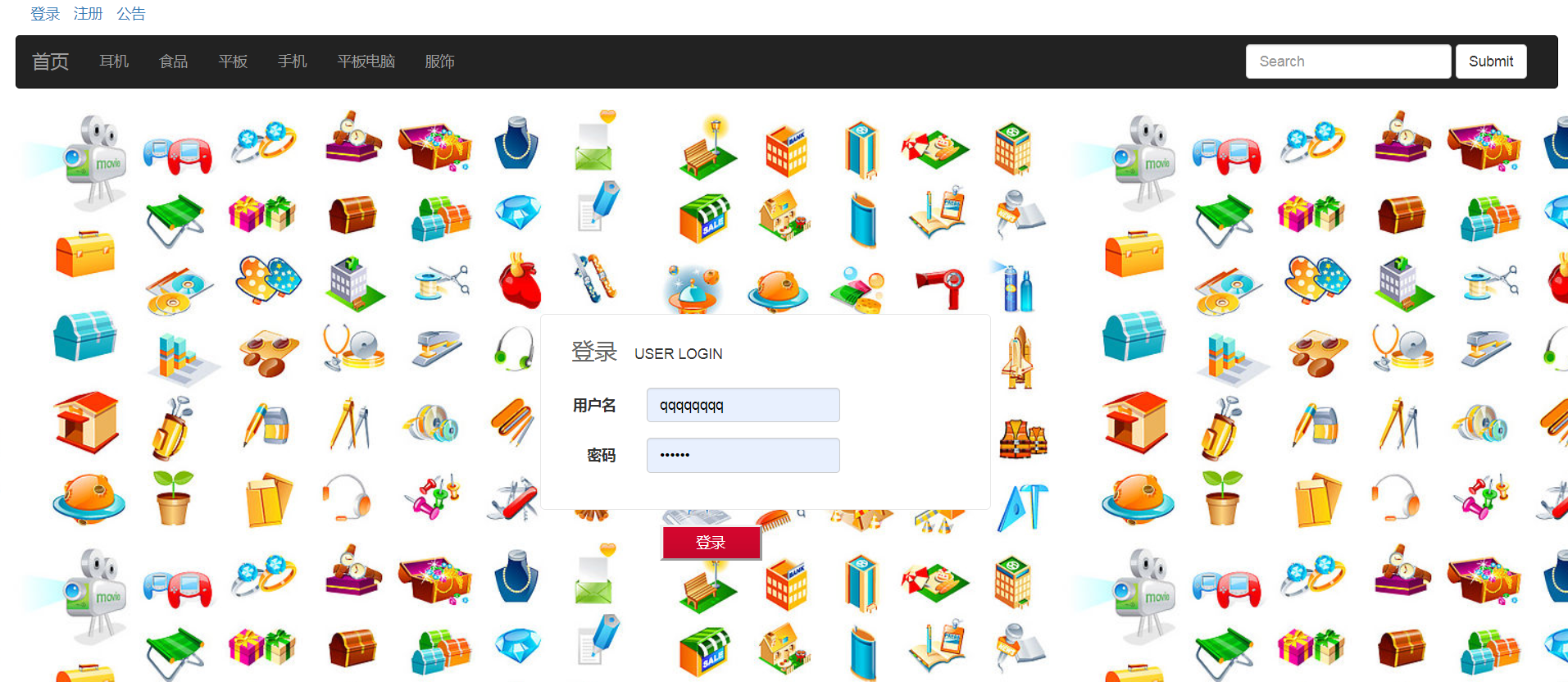


图5-2用户登录

@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：

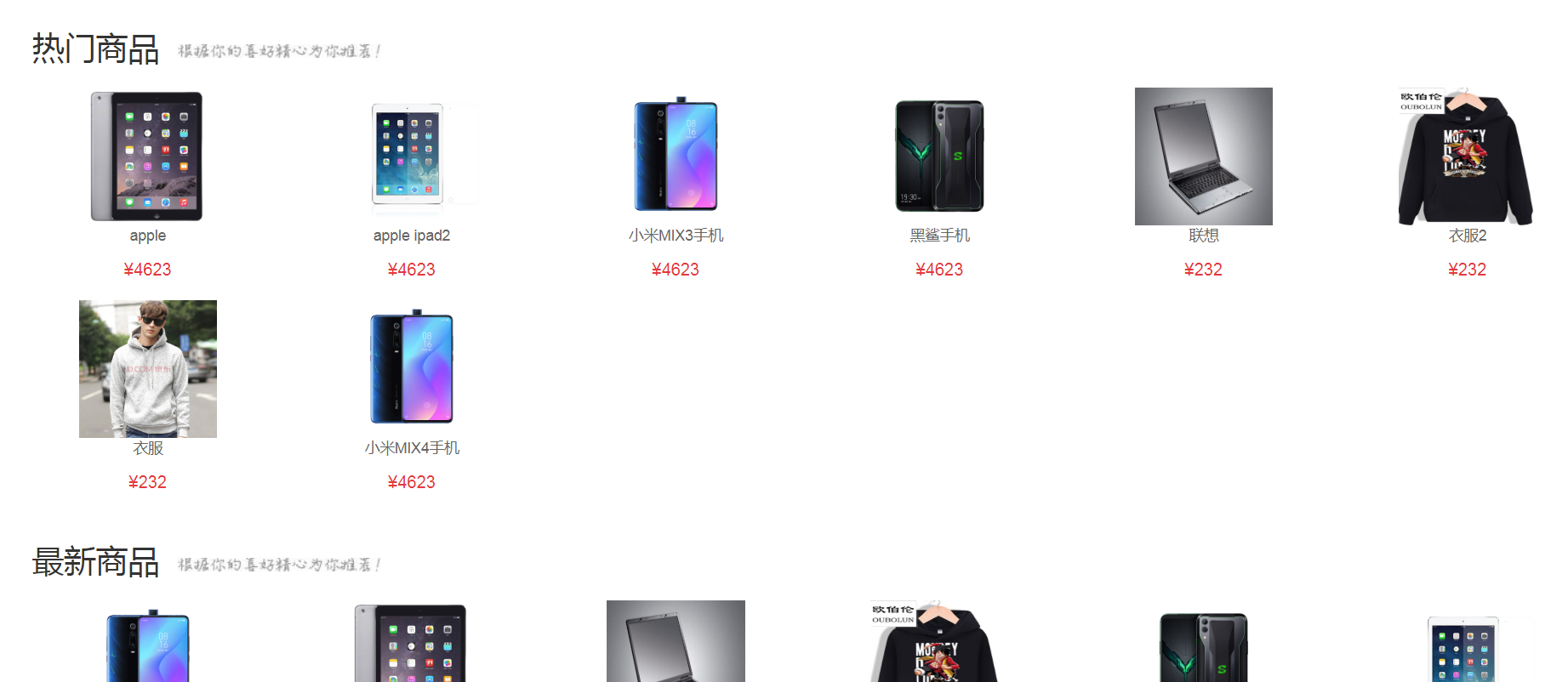


图5-3 首页

**$**(**function**() {

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

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

**$**(**"#hot"**).append(**"<div class='col-md-2' style='text-align:center;height:200px;padding:10px 0px;'><a href='${**pageContext.request.contextPath**}/infoProduct.action?pid="**+element.**pid**+**"'><img src='${**pageContext.request.contextPath**}/pic/"**+element.pimage+**"' width='130' height='130' style='display: inline-block;'></a><p><a href='${**pageContext.request.contextPath**}/infoProduct.action?pid="**+element.**pid**+**"' style='color:#666'>"**+element.pname+**"</a></p><p><font color='#E4393C' style='font-size:16px'>&yen;"**+element.shop\_price+**"</font></p></div>"**)

})

})

*//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) {

**$**(**"#latest"**).append(**"<div class='col-md-2' style='text-align:center;height:200px;padding:10px 0px;'><a href='${**pageContext.request.contextPath**}/infoProduct.action?pid="**+element.**pid**+**"'><img src='${**pageContext.request.contextPath**}/pic/"**+element.pimage+**"' width='130' height='130' style='display: inline-block;'></a><p><a href='${**pageContext.request.contextPath**}/infoProduct.action?pid="**+element.**pid**+**"' style='color:#666'>"**+element.pname+**"</a></p><p><font color='#E4393C' style='font-size:16px'>&yen;"**+element.shop\_price+**"</font></p></div>"**)

})

})

})

单个商品的详细信息页面：



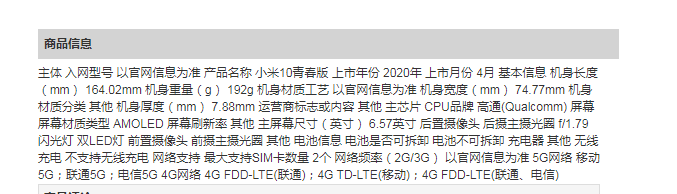


图5-4 商品信息图

实现代码：

<**div class="container"**>

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

<**div style="border**: 1**px solid #e4e4e4**;**width**:930**px**;**margin-bottom**:10**px**;**margin**:0 **auto**;**padding**:10**px**;**margin-bottom**:10**px**;**"**>

<**a href="./index.action"**>首页**&nbsp;&nbsp;&gt;**</**a**>

<**a href='${**pageContext.request.contextPath**}/listProductByCategory.action?curPage=1&cid=${**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**





图5-5购物车

实现代码：

@RequestMapping(**"/addCar1"**)

**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;

}

@RequestMapping(**"/showCar"**)

**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;

}

@RequestMapping(**"/delcar"**)

**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;

}

@RequestMapping(**"/clearcar"**)

**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"**>

<**div style="margin**:0 **auto**; **margin-top**:10**px**;**width**:950**px**;**"**>

<**strong style="font-size**:16**px**;**margin**:5**px** 0;**"**>购物车详情</**strong**>

<**table class="table table-bordered" id="my"**>

<**tbody**>

<**tr class="warning"**>

<**th**>选择</**th**>

<**th**>图片</**th**>

<**th**>商品</**th**>

<**th**>价格(单位：￥)</**th**>

<**th**>数量</**th**>

<**th**>小计(单位：￥)</**th**>

<**th**>操作</**th**>

<**th style="display**: **none"**>购物号</**th**>

<**th style="display**: **none**;**"**>商品pid</**th**>

</**tr**>

<**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**>

</**tr**>

</**c:forEach**>

</**tbody**>

</**table**>

</**div**>

</**div**>

</**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="margin-right**:130**px**;**"**>

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

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

</**div**>

<**div style="text-align**:**right**;**margin-top**:10**px**;**margin-bottom**:10**px**;**"**>

<**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**





图5-6 订单

代码实现：

<**div style="margin**:0 **auto**;**margin-top**:10**px**;**width**:950**px**;**"**>

<**strong**>订单详情</**strong**>

<**table class="table table-bordered"**>

<**tbody**>

<**tr class="warning"**>

<**th colspan="5"**>订单编号:**${**order.oid **}** </**th**>

</**tr**>

<**tr class="warning"**>

<**th**>图片</**th**>

<**th**>商品</**th**>

<**th**>价格</**th**>

<**th**>数量</**th**>

<**th**>小计</**th**>

</**tr**>

<**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**/>

<**form id="orderForm" action="${**pageContext.request.contextPath **}/order/orderForm.action" method="post" class="form-horizontal" style="margin-top**:5**px**;**margin-left**:150**px**;**"**>

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

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

<**input type="hidden" name="oid" value="${**order.oid **}"**>

<**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**>

<**table class="table table-bordered"**><**tr**>

<**th colspan="2"**>订单编号</**th**>

<**th colspan="2"**>订单地址</**th**>

<**th colspan="2"**>收件人</**th**>

<**th colspan="2"**>电话号码</**th**>

<**th colspan="2"**>订单状态</**th**>

<**th colspan="2"**>订单时间</**th**>

</**tr**>

<**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：

@RequestMapping(**"/addOrder"**)

**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.setTotal(totalRow);

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



代码实现：

<**table cellspacing="0" cellpadding="1" rules="all"  
 bordercolor="gray" border="1" id="DataGrid1"  
 style="BORDER-RIGHT**: **gray** 1**px solid**; **BORDER-TOP**: **gray** 1**px solid**; **BORDER-LEFT**: **gray** 1**px solid**; **WIDTH**: 100%; **WORD-BREAK**: **break-all**; **BORDER-BOTTOM**: **gray** 1**px solid**; **BORDER-COLLAPSE**: **collapse**; **BACKGROUND-COLOR**: **#f5fafe**; **WORD-WRAP**: **break-word"**>  
 <**tr  
 style="FONT-WEIGHT**: **bold**; **FONT-SIZE**: 12**pt**; **HEIGHT**: 25**px**; **BACKGROUND-COLOR**: **grey"**>  
 <**td align="center" width="18%"**>  
 序号  
 </**td**>  
 <**td align="center" width="17%"**>  
 分类名称  
 </**td**>  
 <**td width="7%" align="center"**>  
 编辑  
 </**td**>  
 <**td width="7%" align="center"**>  
 删除  
 </**td**>  
 </**tr**>  
 <**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：

<**table cellspacing="0" cellpadding="1" rules="all"  
 bordercolor="gray" border="1" id="DataGrid1"  
 style="BORDER-RIGHT**: **gray** 1**px solid**; **BORDER-TOP**: **gray** 1**px solid**; **BORDER-LEFT**: **gray** 1**px solid**; **WIDTH**: 100%; **WORD-BREAK**: **break-all**; **BORDER-BOTTOM**: **gray** 1**px solid**; **BORDER-COLLAPSE**: **collapse**; **BACKGROUND-COLOR**: **#f5fafe**; **WORD-WRAP**: **break-word"**>  
 <**tr  
 style="FONT-WEIGHT**: **bold**; **FONT-SIZE**: 12**pt**; **HEIGHT**: 25**px**; **BACKGROUND-COLOR**: **gray"**>  
 <**td align="center" width="18%"**>  
 序号  
 </**td**>  
 <**td align="center" width="17%"**>  
 商品图片  
 </**td**>  
 <**td align="center" width="17%"**>  
 商品名称  
 </**td**>  
   
 <**td align="center" width="17%"**>  
 商品价格  
 </**td**>  
 <**td align="center" width="17%"**>  
 是否热门  
 </**td**>  
 <**td width="7%" align="center"**>  
 编辑  
 </**td**>  
 <**td width="7%" align="center"**>  
 删除  
 </**td**>  
 </**tr**>  
 <**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**>  
 <**td align="center" style="HEIGHT**: 22**px"**>  
 <**a href="${** pageContext.request.contextPath **}/admin/editProduct.action?pid=${**p.pid**}"**>  
 <**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="${** pageContext.request.contextPath **}/admin/deleteProduct.action?pid=${**p.pid**}"**>  
 <**img src="${**pageContext.request.contextPath**}/images/i\_del.gif" width="16" height="16" border="0" style="CURSOR**: **hand"**>  
 </**a**>  
 </**td**>  
 </**tr**>  
 </**c:forEach**>  
</**table**>

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;  
 }  
}

1. **系统实现**