**在线销售服务系统**

**摘 要**

伴随着计算机技术在各行各业日益广泛和深入的应用，网络的概念早已深入人心。网络在各行各业的发展战略中占据了重要的位置，成为商家不可分割的部分。商品的宣传已不只局限于电视与报纸，网络已成为商家展示自己的另一个舞台。商家建立网站，将商家各方面的宣传与服务展现于网络中，通过网络更可实现如网上购物、信息查寻等功能，这些在改变我们原有经营方式与经营理念的同时，也为商家带来了更高的效益。因此，对于商家来说，拥有一个属于自己的网站是至关重要的。

本论文研究了一种基于数据关联规则的网上购物系统。其开发主要包括后台数据库的建立、后台管理以及前台页面的Web设计。系统使用Microsoft公司以C#为核心语言的ASP.NET开发工具，利用微软提供的IIS 5.1建立运行环境，再结合SQL Server 2008建立数据连接关系。利用其提供的各种组件，在短时间内建立数据库，对数据库进行分析与建立ASP.NET页面，不断改进，直到功能基本实现的可行系统。

**关键词** 软件生命周期法；信息系统开发；浏览器/服务器；动态网站

Online Sales Service System

Abstract

Follow the technology of the computer in the enterprise and deep application day by day of all trades and professions, the concept internet has been deeply taken up people’s heart. Today, the internet takes a very important part in the development tactics of all walks of life, became an integral part of a company. The merchandise has not only limited in TV set and newspaper. The internet has become another stage which company can display themselves. Company set up website, and make propaganda and service of all aspects of company, display on the internet. Though internet we can achieve buying products on the internet, searching message and so on. Those have changed old way of management and idea of management, meanwhile those also bring more yield for company. So, for the business company, have a website which is belong to themselves is very important.

I studied one kind of system of shopping online based on related-data rule. Its developments mainly include database setting-up, back-stage management and web design of the foreground pages. The system uses ASP.NET developing instrument, which taking C# as key language of Microsoft Company, exploit IIS 5.1 offered by Microsoft as running environment, and then combine with SQL Server 2008 to establish the database. At first, utilize its various kinds of packages offered to set up the database within short time. Then, I analyzed database and made pages with ASP.NET, improved constantly, until basic functions of the system were realized.

**Keywords** Software Life Cycle Method; Information System Developing; B/S; Dynamic Website

目 录

[第1章 绪论 1](#_Toc294888791)

[1.1 引言 1](#_Toc294888792)

[1.2 电子商务系统 1](#_Toc294888793)

[1.3 电子商务系统Web平台的开发方式 5](#_Toc294888794)

[1.3.1 Web平台的体系结构 5](#_Toc294888795)

[1.3.2 Web服务器平台 5](#_Toc294888796)

[1.3.3 Web平台的动态设计技术 7](#_Toc294888797)

[1.3.4 Web平台系统的开发方法 9](#_Toc294888798)

[第2章 系统规划 10](#_Toc294888799)

[2.1 系统规划概述 10](#_Toc294888800)

[2.1.1 系统规划的目标、任务和主要工作 10](#_Toc294888801)

[2.1.2 系统规划工作的特点 11](#_Toc294888802)

[2.2 项目实施前期工作 12](#_Toc294888803)

[2.3 在线销售服务系统的总体设计思路 12](#_Toc294888804)

[2.4 在线销售服务系统的建设原则 13](#_Toc294888805)

[第3章 系统分析 14](#_Toc294888806)

[3.1 系统分析概述 14](#_Toc294888807)

[3.2 系统初步调查 14](#_Toc294888808)

[3.3 可行性研究 15](#_Toc294888809)

[3.4 现行系统的详细调查 16](#_Toc294888810)

[3.4.1 组织结构分析 17](#_Toc294888811)

[3.4.2 功能分解分析 18](#_Toc294888812)

[3.4.3 信息关联分析 19](#_Toc294888813)

[3.4.4 业务流程分析 20](#_Toc294888814)

[3.5 新系统逻辑方案的提出 20](#_Toc294888815)

[3.5.1 数据流图 21](#_Toc294888816)

[3.5.2 数据字典 22](#_Toc294888817)

[第4章 系统设计 25](#_Toc294888818)

[4.1 系统设计概述 25](#_Toc294888819)

[4.2 系统总体结构设计 25](#_Toc294888820)

[4.2.1 系统总体功能结构 25](#_Toc294888821)

[4.2.2 系统总体物理结构 28](#_Toc294888822)

[4.2.3 系统体系结构 29](#_Toc294888823)

[4.3 系统详细设计 29](#_Toc294888824)

[4.3.1 代码设计 30](#_Toc294888825)

[4.3.2 数据库设计 30](#_Toc294888826)

[4.3.3 用户界面设计 38](#_Toc294888827)

[第5章 系统实施与维护 41](#_Toc294888828)

[5.1 开发工具的选择 41](#_Toc294888829)

[5.1.1 ASP.NET开发平台 41](#_Toc294888830)

[5.1.2 Microsoft SQL Server 2008 42](#_Toc294888831)

[5.2 程序设计 42](#_Toc294888832)

[5.2.1 程序设计概要 42](#_Toc294888833)

[5.2.2 功能模块的实现 43](#_Toc294888834)

[5.3 系统测试 52](#_Toc294888835)

[5.3.1 系统测试的原则 53](#_Toc294888836)

[5.3.2 系统测试的方法及步骤 53](#_Toc294888837)

[5.3.3 系统测试的用例及结果 53](#_Toc294888838)

[5.4 系统转换 57](#_Toc294888839)

[5.5 系统维护 58](#_Toc294888840)

[第6章 结 论 59](#_Toc294888841)

[6.1系统总结 59](#_Toc294888842)

[6.2系统展望 59](#_Toc294888843)

[6.2.1电子商务系统的发展趋势 59](#_Toc294888844)

[6.2.2系统体系结构的发展趋势 60](#_Toc294888845)

[致 谢 62](#_Toc294888846)

[参 考 文 献 63](#_Toc294888847)

[附录A 中文翻译 64](#_Toc294888848)

[译文1：《Build your own ASP.NET 3.5 Website using C# & VB (3rd ed.)》 64](#_Toc294888849)

[译文2：《Beginning ASP.NET 3.5 in C# 2008 From Novice to Professional Second Edition》 66](#_Toc294888850)

[译文3：《Online shopping - Selling becomes sociable》 70](#_Toc294888851)

[附录B 外文原文 73](#_Toc294888852)

**第1章** **绪论**

## 1.1 引言

电子商务是伴随着信息经济的脚步，在20世纪90年代兴起于美国、欧洲等发达国家的一个新概念。目前，国际上对电子商务尚无统一的定义。一般来说，狭义的概念是指在开放的国际互联网及其它的网络环境下，通过服务器的运用，实现消费者的网上购物、企业间商业交易的一种新型交易运营模式，即电子商务＝Web（服务器）＋IT（信息技术）+企业业务。

(1) 电子商务的类型：从其交易对象来看，电子商务的应用包括企业与企业之间的电子商务(B2B)、企业与消费者之间的电子商务(B2C)、企业与政府之间的电子商务(B2G)、政府与消费者之间的电子商务(G2C)、企业内部的电子商务。

(2) 电子商务的本质特点：重点在于商务、核心竞争力在于信息平台、电使命在于利用互联网技术对传统资源的整合。

电子商务实现了贸易的全球化、直接化、网络化，发展电子商务是不可阻挡的趋势，而我国的电子商务尚不成熟。要实现全过程的电子商务，就要加快网络基础设施建设和网络互联，实现金融电子化及海关、税务、行政监管部门电子化，市场环境网上支付，建立安全体系认证结构，健全相关法律法规等等。

## 1.2 电子商务系统

电子商务系统是保证以电子商务为基础的网上交易实现的体系。网上交易的信息沟通是通过数字化的信息渠道实现的，交易双方必须拥有相应的技术工具。为保证交易双方进行等价交换，必须提供相应的货物配送和支付结算手段。此外，为保证企业、组织和消费者能够利用数字化沟通渠道，保证交易能顺利进行，需要提供专门服务的中间商参与，即需要电子商务服务商。一个完整的电子商务系统，如图1-1所示：

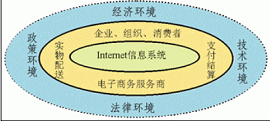


图1－1　电子商务系统

下面按电子商务系统的基础和环境来分别介绍。

**1、基础电子商务系统：**

基础电子商务系统包括Internet信息系统、电子商务服务商、企业、组织与消费者、实物配送和支付结算五个方面。这五方面有机地结合在一起，缺少任何一部分都可能影响网上交易的顺利进行。

1. Internet信息系统

Internet信息系统是指企业、组织和电子商务服务商在Internet的基础上开发设计的信息系统。Internet信息系统的作用是提供一个开放、安全和易于控制的信息交换平台。它是电子商务系统的核心和基石。

1. 电子商务服务商

Internet作为蕴藏巨大商机的平台，需要有一大批专业化分工者进行相互协作，为企业、组织与消费者在Internet上进行交易提供支持。电子商务服务商就起着这种作用。

电子商务服务商主要有以下四种：

a、接入服务商（Internet Access Provider，IAP），它主要提供Internet通信和线路租借服务。

b、服务提供商（Internet Service Provider，ISP），它主要为企业建立电子商务系统提供全面支持。一般企业、组织与消费者上网时都通过ISP接入Internet，由ISP向IAP租借线路。

c、内容服务提供商（Internet Content Provider，ICP），它主要为企业提供信息内容服务，如财经信息、搜索引擎。

d、应用服务系统提供商（Application Service Provider，ASP），它主要是为企业、组织建设电子商务系统时提供解决方案。

(3) 企业、组织与消费者

企业、组织与消费者是网上交易的主体，他们是进行网上交易的基础。企业电子商务系统由基于Intranet（企业内部网）的企业管理信息系统、电子商务站点和企业经营管理组织的人员组成。 如图1-2所示。

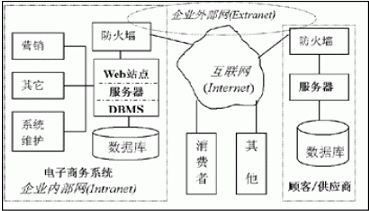


图1－2 企业电子商务系统结构图

a、企业内部网络系统

企业在组建电子商务系统时，应该考虑企业的经营对象是谁，如何采用不同的策略与这些客户进行联系。一般说来，可以将客户分为三个层次并采取相应的对策。对于特别重要的战略合作伙伴，企业允许他们进入企业的Intranet系统直接访问有关信息；对于与企业业务相关的合作企业，企业同他们共同建设Extranet，实现企业之间的信息共享；对普通客户，则可以通过Internet进行联系。

b、企业管理信息系统

企业管理信息系统的作用是在组织中发挥收集、处理、存储和传送信息以及支持组织进行决策和控制。根据组织所具有的不同功能，可以将信息系统划分为营销、制造、财务、会计和人力资源等信息系统。要使各职能部门的信息系统能够有效地运转，必须实现各职能部门信息化。

根据组织内部不同的组织层次，企业管理信息系统可划分为操作层、知识层、管理层、战略层四种信息系统。操作层是支持日常管理人员对基本经营活动和交易进行跟踪和记录。知识层是用知识和数据支持工作人员的工作，帮助公司整理和提炼有用的信息和知识。管理层是用来为中层经理的监督、控制、决策以及管理活动提供服务。战略管理层，主要是关注外部环境和企业内部制定和规划的长期发展方向等问题。

c、电子商务站点

电子商务站点起着承上启下的作用，一方面，它可以直接连接到Internet，顾客或者供应商可以直接通过网站了解企业信息并与企业进行交易。另一方面，它将市场信息同企业内部的管理信息系统连接在一起，将市场需求信息传送到企业的管理信息系统，使企业可以根据市场的变化组织经营管理活动。

企业电子商务系统是由上述三个部分组成。企业内部网络系统是信息传输的媒介，企业管理信息系统是信息加工、处理的工具，电子商务站点是企业拓展网上市场的窗口。

(4) 实物配送

进行网上交易时，如果用户与消费者通过Internet定货、付款后，不能及时送货上门，便不能实现满足消费者的需求。因此，一个完整的电子商务系统，如果没有高效的实物配送物流系统支撑，是难以维系交易顺利进行的。

(5) 支付结算

一个完整的网上交易，它的支付也应该是在网上进行的。但由于目前网上交易尚处在演变过程中，诸多问题尚未解决，如信用问题及网上安全问题，因而许多网上交易并不是完全在网上完成交易的。

**2、电子商务系统环境**

电子商务系统在提供交易所必须的信息交换、支付结算和实物配送这些基础服务的同时，还将面临使用信息技术作为交易平台带来的新问题。如信息安全问题、身份识别问题、信用问题、法律问题、隐私问题、税收问题等。

此外，电子商务发展还面临着企业、组织与消费者是否愿意上网以及是否愿意进行网上购物以及购物时是否愿意使用网上支付手段等问题。这些都是发展电子商务时必须解决的问题。

上述问题的解决必须从外部市场环境开始着手。对于信用、税收问题需要通过制定相关经济政策进行推进。对于安全和身份识别问题需要通过技术进步来保证。对于法律和隐私等问题则需要加强电子商务方面的立法。对于推动消费者上网购物，则需要全社会的参与和引导。因此，发展电子商务是一项系统性的工程，它需要企业主导、政府引导和社会参与。

## 1.3 电子商务系统Web平台的开发方式

### 1.3.1 Web平台的体系结构

Web平台的三层分布式体系结构为图1-3所示。

图1-3 Web平台的三层分布式体系结构

客户端浏览器

Web服务器

应用服务器

数据库服务器

反馈页面

HTTP请求

事务逻辑请求

结果

三层分布式体系结构的第一层是客户端表示层，它仅仅是整个应用系统的图形界面表示；处于第二层的应用服务层，由一台或多台服务器组成，其中Web服务器主要负责对客户端应用程序的集中管理，应用服务器(Application Server)主要负责应用逻辑的集中管理，即事务处理；处于第三层的是数据中心层，有数据库系统，主要负责数据的存储和组织、数据库的分布式管理、数据库的备份和同步等。

电子商务的应用可以看作是一个客户与电子商务站点之间的一系列交互过程。整个交互过程从客户端所浏览的一个页面开始。用户单击该页面上的一个按钮或链接就产生一个请求，该请求通过HTTP传递到Web服务器上，Web服务器将页面中的请求及有关数据取出来传给应用服务器，应用服务器调用相关的商务逻辑，实际上是一个Web应用程序，进行相应的处理，从外部数据库服务器存取数据，然后将结果传回Web服务器，Web服务器再将数据结果转换为页面的形式传给浏览器，于是在客户端产生了新的页面。该页面是这一次请求的结果，也是下一次请求的开始。

### 1.3.2 Web服务器平台

在电子商务系统的开发中，Web服务器平台的选择至关重要，它包括Web服务器操作系统和Web服务器软件，而Web服务器软件通常又决定了将采用什么样的操作系统。Web服务器平台决定了电子商务网站的可靠性、对客户的服务能力，以及为了维持网站运行所需要付出的代价。只有选择了正确的产品和做出相应正确的设计，电子商务系统才能成为商家和顾客之间进行沟通的良好途径。三种最常用的Web服务器平台是:基于Apache HTTP Server的服务器平台、基于网景的NES的服务器平台、基于微软的IIS的服务器平台。微软的Windows 2000 Server + IIS架构的Web服务器平台功能强大，配置灵活，支持目前流行的多种服务器端的开发技术，开发成本低、周期短、维护相对简单，很适合中小企业进行电子商务系统的开发，我们采用了基于微软的IIS的Web服务器平台。下面就三种Web服务器平台的特点作一比较。

(1) 基于Apache HTTP Server的服务器平台

Apache是一种功能强大、高灵活性、高可扩展性的免费Web服务器软件。通过灵活的配置，Apache Web服务器可以实现诸多强大的功能。Apache源代码开放，所以很容易与第三方模块相结合，便于进一步扩展功能，定制成适合自己需要的系统。Apache可运行在多种操作系统及硬件平台上，但通常是与Linux操作系统搭配来组建Web服务器平台。如果采用Linux + Apache架构，可以搭建大型Web服务器平台，实现复杂的企业级电子商务系统。由于Linux和Apache都是免费的软件，只需要花费很低的成本就可以搭建出灵活的电子商务系统Web平台，但是它的开发是复杂而艰巨的，由于Linux和Apache都是开放的、不断扩展中的软件，还不是很完善，系统的配置和维护比较困难，需要专业人员的参与，对于小型的企业电子商务系统并不适合。

(2) 基于网景的NES的服务器平台

网景公司的Netscape Enterprise Server (NES)是一个非常流行的Web服务器平台，很适合开发复杂的企业级网站，几乎可以在所有流行的操作系统上运行，包括Digital、Unix、HP-UX、Solaris和Windows NT等。因特网上很多有名的网站都是用NES开发的。NES功能强大，拥有广泛的第三方组件及工具支持，而且支持多种编程模型。NES通常构建在Linux或Unix类操作系统上，一般适用于开发大型的企业电子商务系统。

(3) 基于微软的IIS的服务器平台

微软公司的Internet Information Server (IIS)是因特网上非常流行的WWW服务器软件，既适用于小网站，也适合于交易量很大的企业级网站。目前，IIS只能运行在Windows NT或Windows 2000 Server上。IIS内有一个集成的搜索引擎，允许用户用多种工具(包括ASP、ActiveX Data Objects以及SQL数据库查询)来创建搜索表，IIS允许用远程浏览器对服务器进行管理，支持多个虚拟主机。此外由于引入了ASP技术，你可以很方便的利用它来创建动态网页。Windows 2000 Server + IIS的架构是当前非常流行的一种Web服务器平台，它的配置和开发都比较容易，很适合中小企业开发B2C型的电子商务系统。

### 1.3.3 Web平台的动态设计技术

目前比较流行的Web开发技术有:CGI， ASP， PHP， JSP，ASP.NET等。在本电子商务系统的开发中，我们主要采用了目前比较流行的，来自微软公司的Web动态设计技术ASP.NET。

1. CGI技术

CGI(公共网关接口Common Gateway Interface)是最古老的Web程序设计方式，它提供了一个外部应用程序与Web服务器交互的标准接口。CGI最大的好处在于其与浏览Web站点的用户之间的交互能力，使信息网关、反馈机制、访问数据库、订阅和查询等一系列灵活复杂的操作得以实现。CGI程序可以使用Perl, C, C++等任何能够形成可执行程序的语言编写，跨平台性能极佳，几乎可以在任何操作系统上实现。但是CGI开发很复杂，且不具备事务(Transaction)处理功能，会大量挤占Web系统的资源，影响系统的运行效率。

1. ASP技术

ASP(Active Server Pages)内含于IIS中，提供了一个服务器端的脚本运行环境，现在的IIS5.1已经支持ASP3.0了。利用它可以产生和运行动态的、交互的Web服务应用程序。与常见的在客户端实现动态主页的技术如Java Applet、ActiveX Control、VBScript、JavaScript等不同，ASP中的命令和Script(描述性的语言)语句都是由服务器来解释执行的，执行结果产生动态生成的Web页面并送到浏览器，而客户端技术的Script命令则是由浏览器来解释执行的。ASP的优点是使用Script，不需要编译，可嵌入到Web页面中，并且由于ASP在服务器端来解释执行，开发者可以不必考虑浏览器是否支持。ASP的缺点是它的跨平台能力差，目前只能运行在微软的Web服务器上，且安全性不好，需要经常下载微软发布的补丁程序。

(3) PHP技术

PHP是一种跨平台的服务器端的嵌入式脚本语言，它大量地借用C、Perl语言的语法，并结合PHP自己的特性，使Web开发者能够快速写出动态页面。它可以在Windows、Unix、Linux的Web服务器平台上正常运行，还支持IIS、Apache等通用服务器软件，用户在更换服务器平台时，只需要作少量的改变即可。PHP最大的好处在于它是完全免费的，可以不受限制的获得源码，使得开发者容易加入自己的特色。PHP的缺点在于虽然它支持目前大多数的数据库，但是提供的数据库接口不统一，比如对Oracle、My SQL、Sybase的接口都不一样，而且缺乏对分布式多层系统结构的支持，由于电子商务系统会涉及到多种数据源，而且通常采用分布式体系结构，因此并不适合用来开发电子商务系统。

(4) JSP技术

JSP是Sun公司推出的新一代站点开发语言，它最大的特点是将Web程序设计的逻辑处理和内容与外观设计隔开，通过使用JSP标记和小脚本来生成页面上的动态内容。在服务器端，JSP引擎解释JSP标记和小脚本，生成所请求的内容，例如通过访问JavaBeans组件，使用JDBC/JDO技术访问数据库，并且将结果以HTML或XML页面的形式发送回浏览器。由于JSP页面的内置脚本语言是基于Java编程语言的，拥有Java编程语言“一次编写，各处运行”的跨平台无关性的好处，开发效率高，安全性好。

(5) ASP.NET技术

.NET 框架是 .NET 平台中居于核心地位的编程引擎，它提供了.NET代码的公共运行环境和一个语言无关的基本类库。.NET 框架是高生产率的、基于标准的多语言应用程序执行环境，处理基本杂务和减轻部署任务。应用程序执行环境管理内存，解决版本控制问题以及提高应用程序的可靠性、可伸缩性和安全性。公共语言运行库是 .NET 框架中的引擎，它提供了一个安全的托管执行环境，并且旨在用于支持开发人员使用多种不同的语言来创建应用程序。它有统一的类型系统并允许跨语言继承和调试。通过使用 .NET 框架，开发人员就拥有了速度最快、效率最高的生成应用程序的方法。.NET平台具有高安全性，.NET框架通过其核心安全功能提供保护客户机和服务器的方法。由于公共语言运行库在核心层透明地处理安全性，因此开发人员可将精力集中于应用程序逻辑。开发人员仍然可以随时使用和扩展安全模型。

目前，CGI技术由于开发复杂，效率也不是很高，现在已经很少有人再用它来开发Web应用程序。ASP和PHP技术则在国内比较流行，但是由于PHP自身的缺点，并不是很适合开发电子商务系统，JSP在国际上比较流行，适合开发一些企业级电子商务系统，但现在国内运用的还比较少，在技术上和成本上要求都比较高。因此对于一些中小企业开发B2C型电子商务系统，采用微软的IIS + ASP方式，开发周期短，成本低，通过引入XML技术来描述及传送数据内容，可以开发出灵活的电子商务应用，但是ASP的跨平台能力差，目前只能运行在微软的Web服务器上，且安全性不好，需要经常下载微软发布的补丁程序。因此，本系统采用C#3.0 + ASP.NET ＋ IIS5.1 + SQL Server 2008为开发工具，具有良好的扩展性。

### 1.3.4　Web平台系统的开发方法

本系统主要运用了系统开发生命周期法(System Development Life Cycle，简称SDLC)和结构化方法(Structured Approach)来阐述该系统的开发过程，经过系统规划、系统分析、系统设计、系统实施及系统维护逐步完成系统各功能的实现，将计算机作为管理决策的支持手段，融入日常业务的各个环节，使得管理更加规范，同时也能够更好的为顾客服务，该系统大大提高了工作效率，为公司带来了更好的社会效益和经济效益。

第2章 系统规划

## 2.1 系统规划概述

凡事预则立，不预则废。科学的规划对于任何需要经过较长时间努力才能实现的事情都是非常重要的。

规划，一般是指对较长时期的活动进行总体的、全面的计划。现代社会组织，特别是企业的结构和活动内容都很复杂，实现一个组织的管理信息化需要经过长期的努力，因而必需对一个组织的管理信息系统的建设进行规划，根据组织的目标和发展战略以及管理信息系统建设和客观规律，并考虑到组织面临的内外环境，科学的制定管理信息系统的发展战略和总体方案，合理安排系统建设的进程。

### 2.1.1 系统规划的目标、任务和主要工作

系统规划是管理信息系统生命周期的第一个阶段，是管理信息系统的概念形成时期，这一阶段的主要目标，就是制定出管理信息系统的长期发展方案，决定管理信息系统在整个生命周期内的发展方向、规模和发展进程。

系统规划阶段的主要任务是：根据组织的整体目标和发展战略，确定管理信息系统的发展战略，明确组织总的信息需求，制定管理信息系统建设总计划，其中包括确定拟建系统的总体目标、功能、大体规模和粗略估计所需资源，并根据需求的轻重缓急还有资源和应用环境的约束，把规划的系统建设内容分解成若干开发项目，以长期分批进行系统开发。

上述三项任务也规定了管理信息系统规划工作进程的三个主要阶段，这就是包曼(B.J.Bowman)和戴维斯(G.B.Davis)等人提出的管理信息系统规划三阶段模型的基本框架，如图2-1所示。

战略计划

组织信息

需求分析

　资源分配

阶段I 阶段II 阶段III

图2-1　系统规划的三个阶段

1、系统战略规划

这一阶段的关键是要使管理信息系统的战略与整个组织的战略和目标协调一致。要进行的工作有：

(1) 评价组织的目标和战略。

(2) 根据组织的目标和战略确定管理信息系统的使命，对管理信息系统的建设或更新提出报告。

(3) 对目前管理信息系统的功能、应用环境和应用现状进行评价。

(4) 制定建设管理信息系统的政策、目标和战略。

2、组织的信息需求分析

(1) 确定目前和规划中的组织在决策支持和事务处理方面的信息需求，以便为整个组织或其主要部门提出管理信息系统的总体结构方案。

(2) 制定主要发展计划，即根据发展战略和系统总体结构，确定系统和应用项目的开发次序和时间安排。

3、资源分配

制定为实现主开发计划而需要的硬、软件资源、数据通讯设备、人员、技术、服务、资金等计划，提出整个系统的建设的概算。

### 2.1.2 系统规划工作的特点

由于管理信息系统的建设耗资巨大，历时较长，技术复杂且涉及面广，系统规划是这一复杂工作的起始阶段，这项工作的好坏将直接影响到整个系统建设的成败。因此，我们应该充分认识这一阶段工作所具有的特点，以提高系统规划工作的科学性和有效性。

1、系统规划工作是面向长远的、未来的、全局性和关键性的问题，因此它具有较强的不确定性，非结构化程度较高。

2、其工作环境是组织管理环境，高层管理人员（包括高层信息管理人员）是工作的主体。

3、系统规划是为整个系统建设确定目标、战略、系统总体结构方案和资源计划，因而整个工作过程是一个管理决策过程。同时，系统规划也是技术与管理相结合的过程，它确定利用现代信息技术有效的支持管理决策的总体方案。

4、系统规划人员对管理与技术环境的理解程度、对管理与技术发展的见识，以及开创精神与务实态度是规划工作的决定因素。

5、规划工作的结果是要明确回答规划工作内容中提出的问题，描述出系统的总体概貌和发展进程，为系统的发展制定一个科学而又合理的目标和达到该目标的可行途径，而不是替代后续阶段的工作。

## 2.2 项目实施前期工作

企业实施电子商务系统，要有目的、有计划、有组织地在正确的方法指导下分步实施。电子商务系统实施的前期工作是关系到是否能够取得预期效益的非常重要的一步。

企业实施电子商务系统总体上分为两个阶段：前期工作和项目实施。之所以这样分是考虑前期工作主要以企业自己提出工作安排为主，而后再转入电子商务系统的项目实施，此时是以咨询机构或软件公司的实施方法为主。概括地说，企业要推行电子商务系统，一般有以下几个原因：

1、企业领导的认识，认为电子商务系统能对企业有一定的促进作用。

2、企业管理人员的认识，认为电子商务系统能对企业有一定的促进作用，进而向领导提出建议书。

3、市场的导向，如政府机构的指令和建议，同行业的实施应用，还有媒体的宣传等。这些因素导致企业对电子商务系统产生兴趣，进而去了解与立项。

4、企业自身的需求。企业确实因为要解决某些问题，要树立企业良好的社会形象及充分利用丰富的网络资源为其拓展销售渠道，获取最大限度的经济效益。经过认证认为电子商务系统是这些问题的解决方案。

## 2.3 在线销售服务系统的总体设计思路

随着科学技术日新月异的发展，崭新的传播媒体——因特网诞生了。在Internet和Web迅猛发展的今天，数字设计技术对21世纪的平面设计家提出了许多挑战。面对新时代的到来，人们无不感到社会经济和文化所经历着的快速而深刻的变化，一张结构合理、制作精美、创意独特、风格鲜明的网页，不仅能够传递最丰富的信息，同时还能给人以美的艺术享受，也树立了公司企业、服务机构等的良好社会形象，获取最大限度的经济效益。

电子购物商城设计是一种建立在新型媒体之上的新型设计，它与传统媒体相比有许多的不同，它在信息传播方面具有声、像、图、文、视觉、听觉、互动的特点，通过视觉传达各种各样的商品信息，而且更新和修正较为快捷方便，可以在任何时间更新，而不受时间和地域的限制。在线销售服务系统的总体设计思路：

1、需求的综合叙述：主要是说明建站的总体需求和期望的目标浏览者，并且确定希望浏览者所能看到的设计与所需购买商品的最终效果，建站需求是设计策划网站功能的第一步。

2、网站的基本功能描述：主要包括描述网站浏览者如何运用这个网站，网站管理员如何管理，具体到一步一步的功能细节描述，以及如何运用相应的技术去实现建设目标和商业需求。

3、网站的信息布局图：网站的信息布局图是网站各模块或各网页的直观图，主要目的是提供一个明晰的网站导航直观图，以及分模块或网页的网站层次结构分布图。

4、数据流程图及数据库示意图：这是网站建设蓝图中的技术部分，主要直观地说明网站使用与维护的数据流程，分析数据来源；以及浏览者和网站管理者如何运用与更新网站数据。

## 2.4 在线销售服务系统的建设原则

为了达到在线销售服务系统的目标和功能，在系统建设的过程中应遵循以下原则：

1、统一规划、分步设计

2、坚持标准化和可行性

3、保证系统的稳定性和安全性

4、保证网络系统的先进性和有效性

5、操作简单、人机界面友好，便于用户理解和使用

第3章 系统分析

## 3.1 系统分析概述

在管理信息系统的生命周期中，紧接系统规划阶段的是系统开发，而系统分析阶段是管理信息系统生命周期中系统开发的第一环节，这一环节工作质量的高低直接影响到后继工作的进行, 特别是系统分析中产生的可行性研究报告和系统说明书。其中，可行性研究报告是系统开发任务是否下达的决策依据。系统说明书是整个系统分析阶段的工作总结，是系统分析人员和用户交流的主要手段，是系统建设领导部门决定是否进行系统建设的主要依据。

系统分析阶段的目标就是在系统规划所定的某个开发项目范围内明确系统开发的目标和用户的信息需求，提出系统的逻辑方案。系统分析在整个系统开发过程中，是要解决“做什么”的问题，提出逻辑方案，为了下一阶段进行物理方案设计，解决“怎么做”提供依据。

系统分析阶段的主要活动有：系统初步调查、可行性研究、系统详细调查、新系统逻辑方案的提出。

系统分析的工作特点：

1、工作内容涉及面广，不确定性大。

2、系统分析工作主要面向组织管理问题，工作方式主要是和人打交道。

3、系统分析工作主要成果（产品）是文件。

4、系统分析的工作追求的是有限目标。

## 3.2 系统初步调查

系统初步调查是系统分析阶段的第一项活动，它的主要目标是从系统分析人员和管理人员的角度看新项目开发有无必要和可能。

系统初步调查采用的主要方式是系统分析人员与在线销售服务系统各个管理者和使用者进行交流，并到经营管理现场参观了解情况，经过初步调查了解到，电子购物商城的基本情况为：

网上购物商城是“当当网”有限公司为拓展公司业务所建立的商业网站。“当当网”有限公司是全球最大的综合性中文网上购物商城，由国内著名出版机构科文公司、美国老虎基金、美国IDG集团、卢森堡剑桥集团、亚洲创业投资基金（原名软银中国创业基金）共同投资成立。公司成立至今已逐步在零售行业中崭露头角。互联网提供了可以无限伸展的展示空间，可以容纳无限的商品或图样以及内容。在当当网，消费者无论是购物还是查询，都不受时间和地域的任何限制。在消费者享受“鼠标轻轻一点，精品尽在眼前”的背后，是当当网耗时11年修建的“水泥支持”——庞大的物流体系，位于六个城市的十大物流中心，全国库房面积达到18万平方米，成为国内库房面积最大的电子商务企业，提供货到付款服务的城市超过750个，并为联营商户开通了COD(Central Office District，中央办公区)服务。

因此，公司希望通过电子购物商城的建立，更加发展壮大公司的运营结构和收益，使之能适应信息化的发展战略和将来的行业竞争。

## 3.3 可行性研究

建设Web平台系统的必要性取决于需求的迫切性和实现的可能性。可行性并不等于可能性，它还包括必要性，如果领导或管理人员对信息的需求并不迫切或各方面的条件不具备，就是不具备可行性。

可行性研究是系统开发阶段的第二项活动。它是对系统进行全面、概要的分析。在本阶段可以从技术、经济和管理三个方面对在线销售服务系统的可行性进行分析研究，从而为确定项目开发决策提供科学依据。

1、技术可行性

由于公司领导对在线销售服务系统开发项目达成了共识，并拨出专项资金，用以购置建立网络中心所需的网络设备和软件，并配备系统管理员，负责整个系统的维护管理工作，具备了开发Web平台系统的基本条件。

为了今后的系统维护，公司准备联合具有丰富经验的软件开发人员共同研究，这为今后系统的顺利开发提供了有力的技术条件。

2、经济可行性

目前，公司的管理工作和服务工作存在盲目性、随意性、和无效消耗，不能保证营销部门的工作质量，影响商品的销售，给公司带来实际的和潜在的经济损失。虽然系统开发初期投资较大，但是，若开发成功本系统，将有助于公司更好地预测市场，更好的开发客户及时调整经营销售策略，在激烈的市场竞争中把握主动。因此，从长远利益考虑，本项目若能开发成功，它所带来的效益将远高于系统投入。

3、营运可行性

在公司的业务方面，由于日常信息处理量大，耗费时间长，出错效率高，在系统投入运行后，可以实现业务中的信息集中处理、分析利用信息和信息的交流辅助公司的业务监管和重大决定。并且可以实现公司进一步向网络服务发展，为信息工作打下良好的基础。

人员方面公司现有的员工中，其中60%以上具有IT专业技术职称，他们掌握的计算机技术，在这样的企业队伍中推行信息化建设可以达到事半功倍的效果。在线销售服务系统建立后，可以改善公司的运营结构和作业方式，使之能适应信息化的发展战略和将来的行业竞争，同时提高公司员工的信息化意识，使公司能在将来的发展过程中不断壮大。

通过以上的研究和分析，认为系统开发所产生的效益将大于投入，所以开发本项目是可行的、必要的。

## 3.4 现行系统的详细调查

对现行的管理业务进行详细调查研究是了解系统需求和进行系统分析与设计的重要基础工作，因此要对电子购物商城的各项管理工作进行全面、细致、充分的调查研究。系统调研采取的主要方法是到经营管理的现场去，通过座谈、询问、观察甚至直接参与管理活动，自上而下逐步细化的进行了解，调查内容涉及上至企业发展目标、经营管理情况，下至具体职能管理工作和业务处理操作的网站经营管理全过程。

首先与网站高层管理人员进行座谈，了解网站的整体经营管理活动及他们对信息系统的需求，在此基础上，调研工作深入到具体各管理部门，从部门经理到具体基层管理人员，逐一进行详细的调查，了解各层管理人员的工作内容、方式和方法，了解他们对信息系统的需求情况，与他们共同讨论如何通过信息系统开展他们的工作以及辅助高层管理人员的决策。

本项活动的目标，在可行性研究基础上，进一步对现行系统进行全面深入的调查分析，弄清现行系统的运行状况，发现其薄弱环节，找出要解决的问题实质，确保信息系统比原来系统更有效。

### 3.4.1 组织结构分析

通过与当当网工作人员进行交流，并参观其业务活动，对其组织结构进行分析，了解到当当网下设行政事业部、财务部、营销中心等3个部门。其中电子购物商城，市场部，企划部，质检部，售后支持部，每个部门由均配备部门经理，当当网组织机构图如图3-1所示。

总经理

财务部

营销中心

行政事业部

售后支持部

质检部

企划部

市场推广部

在线销售服务系统

李冥

图3-1 当当网组织机构图

各部门岗位的主要职能如下：

1、财务部：负责公司全面的财务管理及监督相关税务。

2、行政事业部：是公司全面日常管理工作的组织协调、综合管理部门，并负责公司人事、行政、后勤的管理。

3、营销中心：是公司的营销方面的参谋和执行机构。内容包括：营销策划、市场计划管理和产品调配、广告策划及宣传、对客户的管理，信息及反馈。

4、企划部：营销计划及任务安排，广告策划，市场动态分析，组织促销活动，业务预警等。

5、售后支持部：对所售商品，进行安装指导，客户使用过程中的信息反馈，并将商品质量意见反馈各有关部门。

6、市场推广部：负责公司商品和企业形象的宣传和推广工作。

7、质检部：按照国家质量管理要求对公司商品进行全面质量检测，达到最终合格。

8、在线销售服务系统：负责网上商城的管理和运行工作。

### 3.4.2 功能分解分析

随着公司规模的扩大，业务量和客户群的增多以及公司管理方式的不断改进，加之营销工作本身的复杂性和多样性，原有组织中某些业务部门的业务范围和工作性质都发生了变化，从公司的组织结构中已不能完全反映出该部分所包含的业务。如果从功能的角度来考虑系统的建立和开发，那么该系统对组织的变化就有一定的相对独立性。所以，要在公司的组织结构中进行功能层次的分析，在线销售服务系统组织结构功能层次图，如图3-2所示。

　营销管理

商品管理

服务管理

销售管理

客户管理

图3-2 组织结构功能层次图

各模块功能情况的说明如下：

1、客户管理：

客户管理主要是收集、制定和归档与企业相关的各级客户、各级分销商、竞争对手以及供应商等经济实体的详细资料。

2、商品管理：

主要负责对公司所销售的各类产品的详细信息、出库入库、转入转出、库存调整以及与以上业务相关的各类报表的管理。

3、销售管理：

主要负责为进行销售而开展的各项活动的管理、商品的发货管理以及对商品的退货管理和对各类应收账款的管理。

4、服务管理：

将客户通过各种手段反映到公司的各类信息和服务请求进行分类汇总，并提交相关部门解决。

### 3.4.3 信息关联分析

在线销售服务系统作为公司商品销售活动的中心，需要联络和协调各业务部门，为各户提供产品和服务。与其他部门加强信息交流，互相配合形成紧密的合作关系。为了描述营销中心与其他部门的业务联系，可采用信息关联图来描述这种业务关系。

与营销中心相关的主要部门有：财务部、物质保障部、质检部、行政事业部等。在线销售服务系统与这些部门的信息沟通，可用图3-3所示。

　　　　　　　1 　　　　　　　　　　 7

在线销售服务系统

　　　　　　　2 8

行政事业部

财务部

　　　　　 　　　3　　4 6 　　5

质检部

物质保障部

图3-3 信息关联图

1、日报表、现金、支票；2、各种收据；3、销售商品原材料采购；4、提供合格零配件；5、商品确保质量；6、保证售出商品合格；7、待聘人员资料，培训计划；8、信息回馈

### 3.4.4 业务流程分析

为了弄清各部门的信息处理工作，哪些与系统建设有关，哪些与系统建设无关，就必须了解组织的业务流程。按业务活动中信息流动过程，逐个调查所有环节的处理业务、处理内容、处理顺序和对处理时间的要求，弄清楚各环节需要的信息内容、信息来源、去向和处理方法，提供信息的时间和信息形态等有关调查情况并用“业务流程图”来表示。

业务流程图是一种表明系统内各单位，人员之间业务关系，作业顺序和管理信息流动的流程图。它可以帮助分析人员找出业务流程中的不合理迂回等，业务流程图中的符号说明如图3-4所示。

处理 数据存储 物或单据 物或信息流 业务处理单位

图3-4 业务流程图基本图例

## 3.5 新系统逻辑方案的提出

这一阶段的主要目标是明确用户的信息需求，包括组织、发展、改革的总信息需求和各级管理人员完成各自工作任务的信息需求。确定新系统的逻辑功能，提出新系统的逻辑方案，是系统分析阶段的最终成果，是系统分析的最后一项活动，也是形成系统分析结果的关键工作，这一阶段不但要利用前几个活动的工作成果，而且要在此基础上进一步调查、明确问题、进行创新。

本系统的逻辑模型主要以数据流图和数据字典为描述工具。在对电子购物商城的管理方式和业务流程进行分析和研究的基础上，通过对系统的初步调查和详细调查，按信息系统中应有的数据流和数据结构来勾画系统的概貌。

### 3.5.1 数据流图

为了建立一个满足用户需求的管理信息系统，系统分析人员应在深入调查、详细占有资料的基础上，集中分析管理活动中信息运动的规律和存在的问题，研究如何改善信息流的运动，满足用户管理决策活动中的信息需求。因此要对调查材料进行加工、提炼、抽出其中之反映企业信息运动规律的部分，把用户的需求“翻译”成对信息处理功能的需求，以建立系统的逻辑模型。

数据流图 (DFD) 是组织中信息运动的抽象，是管理信息系统逻辑模型的主要形式。这个模型不涉及硬件、软件、数据结构与文件组织，它与系统的物理描述无关，只是用一种图形及与此相关的注释来表示系统的逻辑功能。图形描述简明，清晰，不涉及技术细节，所描述的内容是面向用户的。因此数据流图是系统分析人员与用户进行交流的有效手段，也是系统设计的主要依据之一。

数据流图是在对业务流程图进行分析的基础上，从系统的科学性、管理的合理性、实际运行的可行性角度出发，将信息处理功能和彼此之间的联系自顶向下、逐层分解，从逻辑上精确的描述新系统具有的数据加工功能、数据输入、数据输出、数据存储及数据来源和去向。

1、数据流图四种基本成分组成：

外部项（外部实体）、加工（数据加工）、数据存储、数据流。

(1) 外部项（外部实体）：在数据流程图中表示所描述系统的数据来源和去处的各种实体或工作环节。这些实体或环节向所开发的系统发出或接受信息。系统开发不能改变这些外部实体本身的结构和固有属性。

(2) 数据加工（加工）：又称数据处理逻辑，描述系统对信息进行处理的逻辑功能。在数据流程图上这种逻辑功能由一个或一个以上的输入数据流转换成一个或一个以上输出流来表示。

(3) 数据存储：逻辑意义上的数据存储环节，即系统信息处理功能需要的、不考虑存储物理介质和技术手段的数据存储环节。

(4) 数据流：与所描述系统信息处理功能有关的各类信息的载体，是各加工环节进行处理和输出的数据集合。在数据流程图中数据流用箭线表示，箭头指处表示数据流的输送处，箭尾连接处表示数据流的来源。

2、数据流程图不但能够反映现行系统中数据处理的逻辑功能，而且能用它来反映将要建立的系统的逻辑模型，因此数据流程图是系统分析有利的工具，熟练的掌握数据流程图的画法，对系统分析至关重要，在数据流程的绘制过程之中应该遵循以下几个原则：

(1) 首先确定系统的外部项，系统分析员通过对现行系统的业务调查首先要识别出不受系统控制但影响系统运行的外部因素，确定系统的数据输入来源和输出去向。

(2) 高层数据流程图中一般只画出系统正常运行时的主要输入和输出，只反映主要的、正常的逻辑功能。这样做的优点在于这张高层数据流图使人一目了然，便于系统分析员和业务人员的理解和沟通。

(3) 数据流程图可以与规划的企业模型相对应，系统各职能域以及相互关系用一级数据流图表示，对每个职能域的扩展形成的业务过程及相互关系用二级数据流图表示，对所有业务过程的扩展所形成的业务活动及其相互关系用三级数据流图表示。

(4) 绘制数据流程图一般遵循从左到右、从上到下的原则进行，先从左侧开始画起，标出外部项。左侧的外部实体通常是系统主要的数据输入来源，然后画出由该外部实体产生的数据流和相应的处理功能。接受系统数据的外部实体一般要画在数据流图的右侧。

(5) 数据流程图需要进过反复的修改才能最终形成，因此可以首先画出草图，一旦定稿，就要正式绘制出系统的数据流图，在正式绘制数据流图的过程中要尽量避免线条的交叉，整张流程图的布局要合理、整洁和清楚。

(6) 所有数据流图一定要经过仔细的检查以保证其正确性。

以在线销售服务系统订单审核数据流程图为例，其顶层数据流图，如图3-5所示。

在线销售服务系统

订单审核

在线销售服务系统

图3-5 订单审核顶层数据流程图

### 3.5.2 数据字典

数据流图重点在描述以处理逻辑为中心的数据的流入与流出，而对数据本身未加描述，影响了用户对数据流图的理解。数据字典是对数据流图的很好的补充，它是数据流图上所有成分的定义和解释的文字集合，它是对数据流图的各种成分起注解、说明的作用，同时它还对系统分析中其他需要说明的问题进行定义和说明。系统分析人员把不便在数据流图上注明而对于系统分析应该获得，对整个系统开发以至将来系统运行与维护是必须的信息尽可能放入数据字典。

数据字典对数据流图的描述尽可能说明下列问题：

1、什么？(是什么或是什么)

2、何出？(在何处来或者来自何处，去向何处)

3、何时？(何时出现，时间长短)

数据字典描述的主要内容有：数据流、数据元素、数据存储、加工和外部项，其中数据元素是组成数据流的基本成分。在系统分析中，数据字典起着重要的作用。它包含关于系统详细信息。一般说来，系统分析人员把不便在数据流图上注明而对于系统分析应该获得，对整个系统开发必需的信息，尽可能放入数据字典，部分数据字典如下所示。

表3-1 数据字典：数据流

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据流  系统名：销售订货 编号：  条目名：订货公司 　别名： | | | | | |
| 来源：　客户 | | 去处： 接收订货信息 | | | |
| 数据流结构：  合同编号＋客户名称＋签订日期＋商品型号＋数量＋单价＋总价＋供货日期  ＋付款情况＋签订人员＋备注 | | | | | |
| 简要说明：营销人员根据客户要求签订订货合同，随时了解合同付款情况。 | | | | | |
| 修改记录： | 编写 | | xxx | 日期 | 年.月.日 |
| 审核 | | xxx | 日期 | 年.月.日 |

表3-2 数据字典：数据存储

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据存储  系统名：销售订货 编号：  条目名：订货客户记录 别名： | | | | | | |
| 存储组织：订货客户记录 | 记录数：  数据量： | | | 主键：客户编号  副键： | | |
| 记录组成：  　合同编号、客户名称、签订日期、商品型号、数量、单价、总价、供货日期、付款情况、签订人员、备注 | | | | | | |
| 简要说明：营销人员根据订货客户记录为客户准备商品。 | | | | | | |
| 修改记录： | | 编写 | xxx | | 日期 | 年.月.日 |
| 审核 | xxx | | 日期 | 年.月.日 |

第4章 系统设计

## 4.1 系统设计概述

系统设计是在系统分析的基础上由抽象到具体的过程，同时，还应考虑到系统所实现的内外环境和主客观条件，本着实事求是的态度进行这一阶段的工作。

系统设计阶段的主要目的是将系统分析阶段所提出的反映用户信息需求的系统逻辑方案转换成可以实施的基于计算机与通信系统的物理方案。

这一阶段的主要任务就是从管理信息系统的总体目标出发，根据系统分析阶段对系统的逻辑功能的要求，并考虑到经济、技术和运行环境等方面的条件，确定系统的总体结构和系统各组成部分的技术方案，合理选择计算机和通信的软、硬件设备，提出系统的实施计划，确保系统总体目标的实现。

系统设计工作的特点:

1、系统设计阶段，大量工作是技术性的。

2、允许用户对已提出的信息需求做非原则性的修改或补充。

3、用户在操作使用和运行环境等方面的具体要求也要在系统设计阶段加以明确并在系统的技术方案中得反映，因此系统设计人员还要同管理环境打交道。

4、系统设计工作的环境是管理环境和技术环境的结合，是系统设计工作的重要特点。

系统设计分为两部分：系统总体结构设计和系统详细设计。

## 4.2 系统总体结构设计

系统的总体结构设计是指整个系统由哪些部分组成，以及各部分在物理上、逻辑上的相互关系，包括硬件部分和软件部分。为了实现系统的总体功能，提高系统的各项指标，总体结构设计的主要任务是将整个系统合理的划分为各个功能模块，正确地处理模块之间与模块内部的联系以及它们之间的调用关系和数据联系，定义整个模块的内部结构。

### 4.2.1 系统总体功能结构

根据系统分析的结果数据流程图所描述的系统的逻辑模型，把数据流程图上每个处理模块进一步分解，确定系统的层次结构关系，把逻辑模型变为物理模型。

在管理信息系统的设计过程中，系统划分的一般原则如下：

1、子系统要具有独立性。即应把相对独立的事务处理过程作为一个子系统。这样的子系统具有较高的内聚性，与其它的藕合较弱，便于软件开发、使用及日后系统维护。

2、子系统的划分结果应使数据冗余度较小。应避免部分数据在多个子系统中出现，尤其是修改，以保证系统的安全性、可靠性和数据的一致性。

3、子系统的划分应能满足目前和日后管理上的需要，功能应齐全，操作要方便且为今后发展留有接口。

4、子系统的划分应以该单位的管理职能为依据。应尽量避免过分依赖于组织机构，以防止机构变更后为系统带来不必要的麻烦。

从公司的组织结构上出发，把整个管理信息系统划分成多个子系统。每个子系统的功能相对独立，但是子系统之间有必要的数据联系。

对在线销售服务系统的数据流图采用变换中心法和事务中心法进行分析，对其逐步进行功能分解，直到分解成为含义明确、功能单一的单元功能模块，从而得到系统的功能模块结构图，即系统的功能图。如图4-1，反映了电子购物商城的系统功能模块的划分。

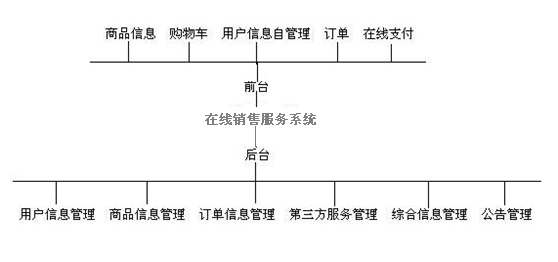


图4-1 在线销售服务系统功能模块结构图

1、前台模块详细功能描述

其中，用户信息管理以及订单是禁止匿名用户访问的内容。

1. 商品信息模块：

商品分类浏览、按商品名称搜索、高级搜索、商品详细信息、商品分类列表、收藏商品、查看商品评论。

1. 购物车模块：

添加商品到购物车、购物车的信息修改、结账、在购物车中添加商品到收藏夹。

1. 用户信息管理模块：

注册新用户、登录、修改密码、修改Email地址、修改收货地址、编辑个人资料、个人留言板管理、用户投诉管理、用户对商品评价的管理。

1. 订单模块：

查询个人订单列表、查询历史支付订单、查询某笔订单的详细信息。

1. 在线支付模块：

针对顾客选择“网银支付”的订单提供在线支付功能，支持国内多家银行。

2、后台模块详细功能描述

(1) 用户信息管理：

登录管理；系统管理员信息的新增、删除、查看、修改；部分重要操作的日志（记录管理员的部分重要操作）；会员信息的新增、删除、查看、修改；商户信息的新增、删除、查看、修改。

(2) 商品信息管理：

商品类别的新增、删除、查看、修改；商品信息的新增、删除、查看、修改；会员收藏管理。

(3) 订单信息管理：

订单审核管理、收货地址管理、会员购物车管理。

(4) 第三方的服务管理：

送货方式管理、付款方式管理、网银服务管理。

(5) 综合信息管理：

顾客投诉管理、顾客留言管理、商品评论管理。

(6) 公告管理：

公告的新增、删除、查看、修改。

### 4.2.2 系统总体物理结构

电子购物商城各个单位在不同的地点工作，网络拓扑结构以交换机为基础，各部门通过主交换机连接到公司主服务器，电子购物商城有多个不同办公地点的办公处组成。根据这个特点，系统利用微机局域网作为系统的基本物理结构。为了提高系统的可靠性和扩展性，网络采用以网络集线器为基础，以双绞线为传输介质的星型结构。在每个业务发生点以及每个管理员分别配制微型计算机作为系统结点工作站，系统功能的实现，子系统间的数据共享与交互通过网络服务器完成。电子购物商城的系统网络拓扑图如图4-2所示。



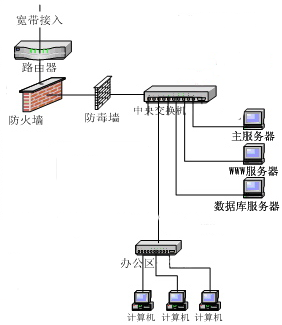


图4-2 在线销售服务系统网络拓扑图

系统具体物理配置如下：

利用公司已经架设好的企业局域网，主服务器采用：服务器、交换机、网络集线器，传输介质采用10M屏蔽双绞线为介质的星型网络拓扑结构，服务器作为系统功能实现和数据共享中心，应具有较好的性能和可靠性。在此基础上再配上一台提供Web服务的服务器和一台SQL SERVER 2008数据库服务器即可实现对本系统的全部物理支持。

系统具体硬件配置如下：

1、WWW服务器：INTER Core2 T5500 2G DDR2 ROM 120GB 硬盘。

2、数据库服务器：INTER Core2 T5500 2G DDR2 ROM 120GB 硬盘。

3、客户机：目前流行的PC兼容机。

4、网络连接设备：路由器、交换机、打印机、扫描仪等。

系统具体软件配置如下：

1、操作系统：Microsoft公司Windows 2000 Server,网络协议采用TCP/IP协议，客户机WINDOWS 2000/XP等。

2、WWW服务器：IIS 5.1。

3、数据库软件：SQL SERVER 2008。

### 4.2.3 系统体系结构

目前，Web项目较为流行的开发模式是三层体系结构，在线销售服务系统的开发按照Windows DNA (Windows Distributed Internet Application，分布式互联网应用结构)，采用基于XML的三层B/S体系结构。Windows DNA提供了一种开发三层结构Web应用程序的通用方法是一个电子商务开发的良好环境。

三层体系结构是指数据访问层(DAL)、业务逻辑层(BLL)、用户表示层(UI)。DAL涉及到数据库本身、存储过程以及提供数据库接口的组件；BLL指的是封装了应用程序商务逻辑的组件；而UI指的是Web应用程序页，如图4-3所示。

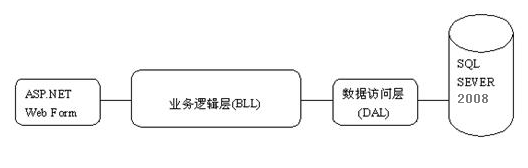
****

图4-3 三层体系结构

## 4.3 系统详细设计

### 4.3.1 代码设计

代码，是用来表示的客观事物的实体类型，以及属性的一个或一组易于计算机识别和处理的特定符号或记号，它可以是字符、数字、某些特殊符号或它们的组合，它可以起到表示、统计、检索和随想状态的描述的作用。

代码是代表系统中客观存在的事物名称、属性或状态的符号。由于现代管理活动中产生的数据量大，所需要的信息种类也很多，为了满足各种管理职能和各个管理层次对信息的需求，必须经过分类整理后才能更有效的利用。代码设计就是信息分类的具体表现，将系统中具有某些共同属性或特征的信息归并到一起。用一些便于计算机或人来识别和处理的符号来表示各种信息。

代码设计实际是一个科学管理的问题。设计出一个好的代码方案，对于系统的开发工作极为有利，它使许多处理工作变得十分方便，通过对数据库以及彼此之间的联系进行分析，对所有数据库文件中的“主码”和“外码”的构成进行分析。

为了编程时方便记忆，本系统均采用英文缩写，作为该项的代码。例如：“Goods”表示商品信息表，“ID”表示商品编号，“Goods\_Name”表示商品名称。

### 4.3.2 数据库设计

几乎所有的WEB项目都是基于数据库的，这使得数据库的设计在整个项目中的地位举足轻重。相关研究表明，在数据应用系统的开发中，项目成功50%靠数据库的设计。良好的数据设计往往能够减少开发的负担。

从数据应用系统和开发的全过程来考虑，数据库的设计可分为六个阶段：需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施、数据库的运行与维护，按照这样的设计过程，数据库结构设计的不同阶段可形成数据库的各级模式。数据库设计的核心问题是如何从系统的观点出发建立一个数据库模式。一个良好的数据库应该充分反映物流发展变化的状况，充分满足组织的各级管理要求，同时还应该使后继系统开发工作方便、快捷，系统开发小，易于管理和维护。在物理设计阶段根据DBMS的特点和处理要求，进行物理储存安排。

1、数据库需求分析

EShop(在线销售服务系统)是一个典型的电子商务系统，针对它将要实现的业务功能，数据库中应该至少具有以下的模块：

(1) 用户信息

(2) 商品信息

(3) 购物车

(4) 订单

(5) 管理员信息

2、数据库概念结构设计

在数据库技术中，我们用模型的概念描述数据库的结构与语义，对现实世界进行抽象。表示实体类型及实体间联系的模型称为“数据模型”，其中之一就是实体联系模型(ER模型)。ER模型是独立于计算机系统的模型，完全不涉及信息在系统中的表示，只是用来描述某个特定组织关心的信息结构，这类模型称为“概念数据模型”。

在线销售服务系统中，存在用户、商品、购物车、订单、管理员等几个对象实体，以上述实体为例，一个用户对应多个商品，同时一个商品属于多个用户；一个用户对应多个购物车，同时一个购物车只属于一个用户；一个用户对应多个订单，一个订单只能属于一个用户；一个商品对应多个购物车，一个购物车对应多个商品；一个商品对应多个订单，一个订单对应多个商品；一个订单对应多个购物车，一个购物车只能属于一个订单；一个管理员对应多个用户、商品、购物车、订单，一个用户、商品、购物车、订单只能属于一个管理员。

由此可知用户和购物车之间、用户和订单之间、订单和购物车之间、管理员和用户、商品、购物车、订单之间都是一对多的关系；商品和购物车之间、用户和商品之间是多对多的关系。

根据上述分析可对应的实体-关系图如图4-4：

实体类型 实体间联系 属性

图4-4 ER图的元素说明

3、数据库逻辑结构设计

通过E－R图分析，下面设计具体的逻辑结构。

所涉及到的主要实体集如下：

用户、商品、购物车、订单、管理员

4、数据库物理结构设计

本系统共涉及到19个表，分别是：管理员角色表EShop\_Admin\_Roles、EShop\_Administrators管理员信息表、公告信息表EShop\_Bullitins、商品种类表EShop\_Classes、用户评论表EShop\_Comments\_C\_To\_G、用户信息表EShop\_Customers、网上银行表EShop\_EBank、用户收藏表EShop\_Favourites、商品信息表EShop\_Goods、商家信息表EShop\_Merchants、给客服留言表EShop\_Messages\_C\_To\_S、用户投诉表EShop\_Messages\_Complaint、订单表EShop\_Orders、订单详细信息表EShop\_Orders\_Details、支付方式表EShop\_Pay\_Methods、向商家问询表EShop\_Questions\_C\_To\_M、收货地址信息表EShop\_Receivers、发送方式表EShop\_Ship\_Methods、购物车表EShop\_Shopping\_Cart。

主要设计思想：

(1) 尽管在需求分析中，商品类别仅仅是商品的一个属性，但数据库设计中一般将其分离出去创建单独的表。事实上，这是一种面向对象的思想，这种做法有利于系统的扩展，例如商城的管理者将要拓宽业务、推出某种新类型的商品，这样的设计将令其变得异常容易。同样的思想也体现在EShop\_Administrators表和EShop\_Admin\_Roles表的设计中。

(2) 由于订单可能包括多种商品，而且数量不一，所以用两个表来共同体现订单这个实体。在EShop\_Orders表中存放订单的基本信息：订单号、所属用户的编号以及下订单的时间。而在EShop\_Orders\_Details表中则存放订单的具体信息，每条记录体现某个订单所包含的某种商品的购买信息。

在本系统中数据库具体表结构的设计：

(1) 管理员角色表 EShop\_Admin\_Roles

用来记录管理员角色的一些基本信息，其结构如表4.1所示。

表4-1 EShop\_Admin\_Roles的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | 角色编号（主键） |
| 2 | Role\_Name | nvarchar(20) | 角色名 |
| 3 | Role\_Desc | nvarchar(200) | 角色说明 |

(2) 管理员信息表 EShop\_Administrators

用来记录管理员的一些基本信息，如表4.2所示。

表4-2 EShop\_Administrators的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | 管理员编号（主键） |
| 2 | NickName | nvarchar(20) | 管理员昵称 |
| 3 | Account | nvarchar(20) | 账号 |
| 4 | Password | nvarchar(50) | 密码 |
| 5 | is\_in\_use | smallint | 是否启用 |
| 6 | Role | nvarchar(20) | 角色 |
| 7 | Create\_Time | nvarchar(50) | 创建时间 |

(3) 公告信息表 EShop\_Bullitins

用来表示公告信息。其结构如表4.3所示。

表4-3 EShop\_Bullitins的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | 公告ID（主键） |
| 2 | Title | nvarchar(20) | 主题 |
| 3 | Detials | nvarchar(500) | 详细内容 |
| 4 | Publisher | nvarchar(20) | 发布人 |
| 5 | Bullitin\_Time | nvarchar(50) | 发布时间 |

(4) 商品种类表 EShop\_Classes

用来存储商品种类的信息。其结构如表4.4所示。

表4-4 EShop\_Classes的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | 编号ID（主键） |
| 2 | Class\_ID | int | 种类编号 |
| 3 | Class\_Name | nvarchar(20) | 种类名称 |
| 4 | Parent\_ID | int | 父类编号 |

(5) 用户评论表 EShop\_Comments\_C\_To\_G

保存用户对商品评论的基本信息。结构如表4.5所示。

表4-5 EShop\_Comments\_C\_To\_G的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | 评论号（主键） |
| 2 | Title | nvarchar(50) | 标题 |
| 3 | Details | nvarchar(200) | 评论内容 |
| 4 | Time | nvarchar(50) | 评论时间 |
| 5 | Customer\_NickName | nvarchar(20) | 顾客昵称 |
| 6 | Customer\_ID | nvarchar(50) | 顾客ID |
| 7 | Goods\_ID | nvarchar(50) | 商品ID |

(6) 用户信息表　EShop\_Customers

保存会员的详细信息，结构如表4.6所示。

表4-6 EShop\_Customers的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | Customer\_ID | nvarchar(50) | 会员编号（主键） |
| 2 | Account | nvarchar(50) | 登入账号 |
| 3 | Password | nvarchar(50) | 密码 |
| 4 | NickName | nvarchar(20) | 昵称 |
| 5 | Customer\_Type | nvarchar(20) | 会员类型 |
| 6 | Register\_Time | nvarchar(50) | 注册时间 |
| 7 | is\_in\_user | smallint | 是否启用 |
| 8 | Sex | nvarchar(10) | 性别 |
| 9 | Birthday | nvarchar(50) | 生日 |
| 10 | HeadImage | nvarchar(200) | 头像 |

(7) 网上银行表 EShop\_EBank

保存提供网上银行服务的单位的基本信息，如表4.7所示。

表4-7 EShop\_EBank的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | EBank\_Name | nvarchar(50) | 银行名称 |
| 3 | EBank\_Image | nvarchar(200) | 银行标志图片 |

(8) 用户收藏表 EShop\_Favourites

用来存储用户的收藏信息，如表4.8所示。

表4-8 EShop\_Favourites的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Customer\_ID | nvarchar(50) | 会员编号 |
| 3 | Goods\_ID | nvarchar(50) | 商品编号 |

(9) 商品信息表 EShop\_Goods

存储商品的详细信息，如表4.9所示。

表4-9 EShop\_Goods的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Goods\_Name | nvarchar(50) | 商品名称 |
| 3 | Goods\_Img\_Big | nvarchar(200) | 商品大图片 |
| 4 | Goods\_Img\_Small | nvarchar(200) | 商品小图片 |
| 5 | Goods\_Price\_Market | int | 市场价格 |
| 6 | Goods\_Price\_EShop | int | EShop价格 |
| 7 | Goods\_Stock | int | 库存量 |
| 8 | Goods\_Desc | nvarchar(500) | 商品描述 |
| 9 | Goods\_Colletion | int | 被收藏次数 |
| 10 | Goods\_Merchant\_ID | nvarchar(50) | 商家ID |
| 11 | Goods\_Class\_Big | nvarchar(20) | 所属大类 |
| 12 | Goods\_Class\_Middle | nvarchar(20) | 所属中类 |
| 13 | Goods\_Class\_Small | nvarchar(20) | 所属小类 |

(10) 商家信息表 EShop\_Merchants

存储商家的详细信息，如表4.10所示。

表4-10 EShop\_Merchants的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | Merchant\_ID | nvarchar(50) | ID（主键） |
| 2 | Merchant\_Name | nvarchar(50) | 商品名称 |
| 3 | Merchant\_Province | nvarchar(20) | 商家所在省 |
| 4 | Merchant\_Address | nvarchar(200) | 商家地址 |
| 5 | Merchant\_OpenTime | nvarchar(50) | 开业时间 |
| 6 | Phone | nvarchar(20) | 联系电话 |
| 7 | ShopKeeper | nvarchar(50) | 店主 |
| 8 | is\_in\_use | smallint | 是否启用 |

(11) 给客服留言表 EShop\_Messages\_C\_To\_S

存储给客服留言的详细信息，如表4.11所示。

表4-11 EShop\_Messages\_C\_To\_S的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Message\_Details | nvarchar(200) | 留言内容 |
| 3 | Customer\_ID | nvarchar(50) | 会员ID |
| 4 | Customer\_NickName | nvarchar(20) | 会员昵称 |
| 5 | Message\_Time | nvarchar(50) | 留言时间 |
| 6 | Reply\_Details | nvarchar(200) | 回复内容 |
| 7 | Reply\_Admin\_ID | nvarchar(50) | 回复信息的管理员ID |
| 8 | Reply\_Admin\_NickName | nvarchar(20) | 管理员昵称 |
| 9 | Reply\_Time | nvarchar(50) | 回复时间 |
| 10 | is\_Reply | smallint | 是否已回复 |

(12) 用户投诉表 EShop\_Messages\_Complaint

存储用户投诉的详细信息，如表4.12所示。

表4-12 EShop\_Messages\_Complaint的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Complaint\_Details | nvarchar(200) | 投诉内容 |
| 3 | Customer\_ID | nvarchar(50) | 会员ID |
| 4 | Customer\_NickName | nvarchar(20) | 会员昵称 |
| 5 | Complaint\_Time | nvarchar(50) | 投诉时间 |
| 6 | Reply\_Details | nvarchar(200) | 回复内容 |
| 7 | Reply\_Admin\_ID | nvarchar(50) | 回复信息的管理员ID |
| 8 | Reply\_Admin\_NickName | nvarchar(20) | 管理员昵称 |
| 9 | Reply\_Time | nvarchar(50) | 回复时间 |
| 10 | Progress\_Rate | nvarchar(20) | 处理进度 |

(13) 订单表 EShop\_Orders

存储订单的详细信息，如表4.13所示。

表4-13 EShop\_Orders的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | Order\_ID | nvarchar(50) | ID（主键） |
| 2 | Customer\_ID | nvarchar(50) | 顾客ID |
| 3 | Order\_Content | nvarchar(200) | 订单内容 |
| 4 | Receiver\_Name | nvarchar(50) | 收件人姓名 |

续表4-13

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Receiver\_Addr | nvarchar(200) | 收件人地址 |
| 6 | Receiver\_PostCode | nvarchar(10) | 收件人邮编 |
| 7 | Receiver\_Phone | nvarchar(20) | 收件人电话 |
| 8 | Totle\_Money | int | 总额 |
| 9 | Create\_Time | nvarchar(50) | 下单时间 |
| 10 | Ship\_Method | nvarchar(50) | 送货方式 |
| 11 | Pay\_Method | nvarchar(20) | 付款方式 |
| 12 | Order\_Status | nvarchar(100) | 订单状态 |

(14) 订单详细信息表 EShop\_Orders\_Details

用来存储订单所含商品内容的详细信息。其结构如表4.14所示。

表4-14 EShop\_Orders\_Details的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | Order\_ID | nvarchar(50) | 订单ID（主键） |
| 2 | Goods\_ID | nvarchar(50) | 商品ID（主键） |
| 3 | Amount | int | 数量 |
| 4 | Subtotal | int | 小计 |

(15) 支付方式表 EShop\_Pay\_Methods

保存支付方式的信息，如表4.15所示。

表4-15 EShop\_Pay\_Methods的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Pay\_Method | nvarchar(20) | 支付方式名称 |
| 3 | Method\_Desc | nvarchar(200) | 支付方式描述 |

(16) 向商家问询表 EShop\_Questions\_C\_To\_M

记录顾客向商家问询的相关信息，如表4.16所示。

表4-16 EShop\_Questions\_C\_To\_M的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Question\_Details | nvarchar(200) | 询问内容 |
| 3 | Customer\_ID | nvarchar(50) | 顾客ID |
| 4 | Question\_Time | nvarchar(50) | 提问时间 |
| 5 | Merchant\_ID | nvarchar(50) | 商家ID |

续表4-16

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | Reply\_Details | nvarchar(200) | 回复内容 |
| 7 | Reply\_Time | nvarchar(50) | 回复时间 |
| 8 | is\_reply | smallint | 是否已回复 |

(17) 收货地址信息表 EShop\_Receivers

记录顾客收货地址的信息，如表4.17所示。

表4-17 EShop\_Receivers的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Customer\_ID | nvarchar(50) | 顾客ID |
| 3 | Receiver\_Name | nvarchar(50) | 收件人姓名 |
| 4 | Receiver\_Addr | nvarchar(200) | 收件人地址 |
| 5 | Receiver\_PostCode | nvarchar(10) | 收件人邮编 |
| 6 | Receiver\_Phone | nvarchar(20) | 收件人电话 |

(18) 发送方式表 EShop\_Ship\_Methods

保存送货方式的信息，如表4.18所示。

表4-18 EShop\_Ship\_Methods的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Ship\_Method | nvarchar(50) | 送货方式名称 |
| 3 | Method\_Desc | nvarchar(200) | 送货方式描述 |

(19) 购物车表 EShop\_Shopping\_Cart

存储购物车详细信息，如表4.19所示。

表4-19 EShop\_Shopping\_Cart的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 说明 |
| 1 | ID | nvarchar(50) | ID（主键） |
| 2 | Customer\_ID | nvarchar(50) | 顾客ID |
| 3 | Goods\_ID | nvarchar(50) | 商品ID |
| 4 | Goods\_Name | nvarchar(50) | 商品名称 |
| 5 | Goods\_Price\_EShop | int | EShop价格 |
| 6 | Credit | int | 积分 |
| 7 | Amount | int | 数量 |
| 8 | Create\_Time | nvarchar(50) | 创建时间 |

### 4.3.3 用户界面设计

目前，界面设计已成为评价软件质量的一条重要指标，一个计算机系统的成功与否，很大程度上取决于用户使用人机界面、人机对话的体验或感受。所谓用户界面是指软件系统与用户交互的接口，也是控制和选择信息输入输出的主要途径，通常包括输入、输出、人机对话的界面与方式等。用户界面设计应坚持友好、简便、实用、易于操作的原则，尽量避免过于烦琐和花哨。

1、输入设计

输入数据的正确性对于整个系统质量的好坏具有决定性的意义，输入设计不当有可能使数据发生错误，这时，所使用的计算和处理十分正确，也不能得到可靠的输出信息。因此，输入设计必须考虑周全，尽量使出错率降到最低。

2、输出设计

系统设计的过程正好和实际实施过程相反，并不是从输入设计到输出设计，而是从输出设计到输入设计，这是因为输出表格直接与使用者相联系，设计的出发点应当是保证输出表格方便的为使用者服务，正确及时的反映和组成用户生产和服务部门的有用信息。

3、人－机对话设计

人－机对话主要是指在计算机程序运行中，使用者与计算机系统之间通过终端屏幕或其他装置进行一系列交替的询问与回答。对话设计的任务是与用户共同的确定对话方式、内容与具体格式。

信息的输入主要采用手工操作方式，以键盘、鼠标作为输入设备，输出设备采用显示终端、打印机、以及纸张、磁盘作为输出介质。

本系统采用了基于B/S结构的菜单选择、填写表格和简单提示等友好的人机交互方式，如图4-5所示。

主菜单区

各模块 显示区

的信息链接

网站相关信息

提示区

图4-5　系统基本用户界面

屏幕上方是在线销售服务系统主功能模块链接区，显示系统登录等主链接，用户通过鼠标选择访问登录页面。屏幕左边的子功能信息链接区，是各子功能模块信息具体体现，用户可通过鼠标选择所需信息。当用户选择了所要进行的工作后，系统进入了相应的业务处理功能模块，整个业务处理活动在显示区域进行，显示相应的查询结果信息。当用户业务处理完毕后系统返回到之前的菜单显示状态；当用户不使用时，可选择退出整个系统。

4、处理过程设计

处理过程设计要根据模块划分以及结构化程序设计的基本原则，对每个程序模块中的详细处理过程进行设计，设计的工具有程序流程图、合图（N-S图）、过程设计语言、判定树和判定表等，本系统中采用的是程序流程图，如在线销售服务系统前台系统程序流程图，如图4-6所示。

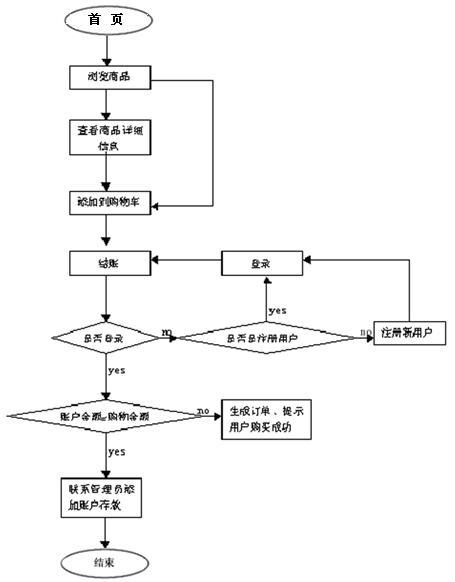


图4-6 在线销售服务系统前台系统程序流程图

第5章 系统实施与维护

系统实施是继系统规划、系统分析、系统设计之后的又一个重要阶段，它将按照系统设计选定的方案具体实施。在系统分析和系统设计阶段，系统开发工作主要是集中在逻辑、功能和技术设计上，工作成果是以各种系统分析和设计文档来体现的。系统实施阶段要继承此前各阶段的工作成果，将技术设计转化为物理实现。

管理信息系统在完成系统实施、投入正常运行后，就进入系统运行与维护阶段。系统维护的目的是保证管理信息系统正常而可靠的运行，并能够使系统不断得到改善和提高，以充分发挥作用。因此，系统维护是为了保证系统中的各个要素随着环境的变化始终处于最新、最正确的工作状态，是信息系统可靠运行的重要技术保障。这一阶段包括开发工具的选择，功能模块的实现等主要活动。

## 5.1 开发工具的选择

程序设计的编写是以程序设计语言为工具的，并要选择适当的程序设计语言。本系统的前端软件开发工具选用了C#.NET和ASP.NET，后台数据库选用 Microsoft SQL Server 2008。

### 5.1.1 ASP.NET开发平台

随着计算机技术和网络技术的不断发展，人们对网络的要求也越来越高。传统的静态网页技术已不能满足广大互联网用户的需要，各种动态网页技术应运而生。ASP是目前网页制作中最常用的重要技术之一，许多网站都使用ASP 制作网页。ASP并不是一种编程语言，而是一种综合的网络解决方案,它允许用户根据自己的习惯采用不同的语言环境，在服务器端经过语言解释器的解释或 HTML 代码发送给客户端。Microsoft新近推出的ASP.NET不是对ASP在一般意义上的版本更新，而是一门全新的技术，是对ASP如何提供动态Web开发环境的跳跃性发展。ASP.NET几乎完全基于组件和模块化，用户使用的每一个页面、部件和HTML单元都是一个运行时的组件对象。ASP.NET的面向对象功能使开发者可以通过创建新组件或继承基础类，为整个环境提供延展性。

.NET是Microsoft公司应用于Web服务的开发环境而推出的一个整体技术框架。通过.Net Runtime统一执行的机制,所有开发语言(VB，C++，C#)都被统一在一个平台上,使开发工具和平台支持的提供代价大大降低,适用性大大提高。.NET的突出特点是它的实用性和高度的集成性。

VisualStudio.Net是该公司推出的下一代可视化开发环境。它为基于组件的应用程序(尤其是在使用Web服务和XML技术方面)开发提供了很大的方便。VisualStudio.Net提供了一个支持VB、C++和C#开发和运行的新环境。

ASP.NET 是建立在公共语言运行库上的编程框架，可用于在服务器上生成功能强大的 Web 应用程序。

### 5.1.2 Microsoft SQL Server 2008

Microsoft SQL Server 2008是一个分布式的关系型数据库管理系统，支持B/S结构，具有强大的网络功能，支持发布Web页面以及接受电子邮件。Microsoft SQL Server 2008是一个应用广泛的数据库管理系统，具有许多显著特点，例如，用户喜欢的易用性、适合分布式组织的可伸缩性、用于决策支持的数据仓库功能、与许多其他服务器软件紧密关联的集成性、良好的性价比等。

在Microsoft SQL Server 2008系统中，数据库的管理采取了先进的动态管理机制，数据库的大小可以随着数据量的变化而缩小或者扩大。另外，用户数据都是非常重要的和保密的，这些数据存储在Microsoft SQL Server 2008系统中有绝对的安全性保障，有完善的用户账户策略和许可机制，只有经过授权的用户才能访问系统，并且执行相应的操作和访问允许访问的数据。未经授权的用户既不能执行相应的操作，也不能访问数据库中的相应数据。

Microsoft SQL Server 2008系统具有强大的网上功能，可以在Internet上发布数据库中的数据。在Microsoft SQL Server 2008系统中，允许用户简单的使用一些向导和工具把数据库的数据发布到网络上，并且可以根据需要来调度修改已经发布的数据。

## 5.2 程序设计

### 5.2.1 程序设计概要

程序设计又称编码，为了保证编码工作正确而顺利进行，一方面程序设计人员必须仔细阅读系统设计的全部文档资料，充分理解程序模块的内部过程和外部接口，以保证系统实施与系统设计的一致性。另一方面，程序设计人员必须深刻的理解，熟练的掌握和正确的运用程序设计语言以及软件开发环境和工具，以保证功能的正确实现。

对程序设计的基本质量要求是程序的正确性，即一方面正确运用程序设计语言环境，以避免语法的错误。另一方面，程序所描述的过程和算法要满足系统设计的功能要求，以避免语义错误。我们尽量使程序不仅能在计算机上正确执行，而且对于阅读人来说犹如一篇好文章，结构清晰，逻辑简明，易读易懂，便于修改和扩充。

### 5.2.2 功能模块的实现

本系统由前台系统和后台管理系统系统组成。前台系统就是面向Internet上用户的网站。后台系统则是商店管理员对网站及其上的信息进行管理和维护的系统，也可以说是一个Intranet系统。

主要有前台系统的主页、商品信息模块、购物车模块、用户信息管理模块、订单模块和在线支付模块，后台的用户信息管理、商品信息管理、订单信息管理、第三方的服务管理、综合信息管理以及公告管理模块。

后台管理模块是一个完整的电子商务系统不可或缺的部分，主要包括：登录（Admin\_Login.aspx）、查看管理员（AdminList.aspx）、添加新管理员（AddNewAdmin.apx）、编辑管理员信息（EditAdmin.aspx）、查询、修改商品信息（GoodsInfoList.aspx）、查询用户信息（CustomerList.aspx）等功能页面。

本系统的用户界面分为以下几个层次(实现代码见附录)：

1、首页(Default.aspx)

是直接展示给在线销售服务系统(EShop)用户的部分，界面效果如图5-1所示。

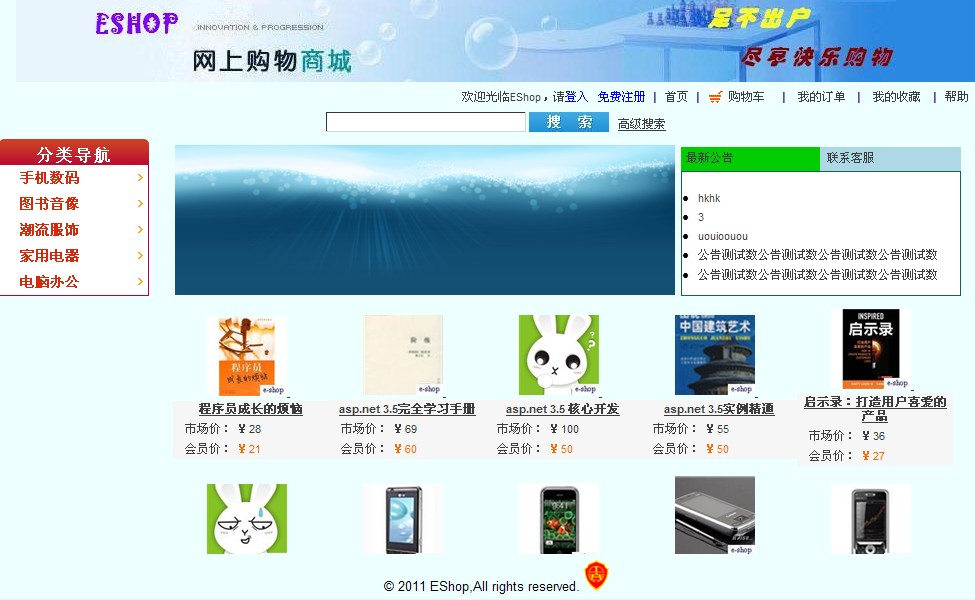
****

图5-1 首页设计状态图以及界面效果

在这一模块中，具有技巧性的地方是如何展示给登录用户和匿名用户不同的导航条，例如，匿名用户看到的导航链接有登录、购物车和帮助，而登录用户看到的则是账号名称、退出登入、购物车、我的订单等。如图5-2、5-3所示。



图5-2 匿名用户导航条



图5-3 登录用户导航条

商品分类导航控件显示所有商品的列表，作为页面的左侧导航目录，如图5-1所示。该控件显示所有的商品分类，将鼠标悬停在目录中某一个类别名称上时将显示相应类别的商品列表页面。

热门商品列表控件，如图5-1中间下方所示。这个控件的功能是在首页显示相应的商品列表，与商品分类导航控件类似，单击某一商品将显示相应商品详细信息页面。

2、商品信息模块共包含如下的页面

(1) 分类显示商品(ProductList.aspx):分类显示某种类型的所有商品。

商品信息包括每个商品的商品名称、简单描述等等。ProductList.aspx的界面效果如图5-4所示。



图5-4 分类显示商品的界面效果

(2) 商品详细信息(ProductInfo.aspx):显示某个商品的详细信息。

这个页面将显示某一商品的详细信息。在EShop中，商品的详细信息包括商品的名称、价格、简介以及浏览次数。商品详细信息的界面效果如图5-5所示。



图5-5 商品详细信息的界面效果

1. 商品搜索结果(QueryResault.aspx):显示搜索结果。

在页面头部的搜索框中输入相应的关键词，提交之后，将在搜索页面显示出模糊查询的结果。显示方式与ProductList.aspx类似。例如，在搜索框输入“a”,则显示出商品名称中包含“a”的所有商品。实现效果如图5-6所示。



图5-6 商品搜索结果的界面效果

3、购物车功能模块

用户单击商品详细信息页面的“购买”按钮时，将进入购物车的功能页面,对购物车的内容进行管理。

该页面实现的功能比较多，包括购物车的记录绑定显示、移入收藏夹、删除购物车中的记录以及显示购物总额等，还可以分页显示。其界面效果如图5-7所示。

****

图5-7 购物车管理的界面效果

4、用户信息管理模块

用户信息管理是一个会员制的电子商务系统中必不可少的部分。在这一模块中实现的功能包括注册、登录、找回密码、修改密码、修改个人详细资料等。具体包括以下几个页面：

1. 登录验证(Customer\_Login.aspx)

登录验证涉及到整个系统的身份验证机制。作为用户验证的窗口，登录页面的实现十分重要，其界面如图5-8所示。



图5-8 登录的界面

1. 注册新用户(Customer\_Register.aspx)

新用户注册时，仅需输入最基本的信息，个人详细资料在注册之后再进行修改。界面如图5-9所示。



图5-9 用户注册的界面

1. 找回密码

当用户忘记密码时，系统将会生成新的随机密码发送到用户的指定E-mail。界面如图5-10所示。

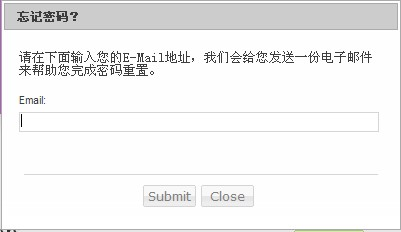


图5-10 找回密码的界面

1. 用户管理导航目录(CustomerMain.aspx)

当进入用户管理页面时，左侧的目录应该显示为用户的管理项导航。当用户进入用户管理（例如修改密码或个人资料）页面时，左侧不应该再显示为商品分类导航目录，而应该列出用户管理菜单，界面如图5-11所示。

****

图5-11 用户管理导航目录的界面

5、订单模块

订单模块包括的功能有支付购物车生成订单、查询个人订单信息和查询订单详细信息。涉及的页面有以下3个：

1. 结算购物车(Checkout.aspx)

该页面显示的数据与ShopCartList.aspx相同，该界面中内容是用来让用户确认的，界面如图5-12所示。



图5-12 结算购物车的界面

1. 查询个人订单列表(OrderList.aspx)

该页面显示用户订单列表，界面如图5-13所示。



图5-13 查询个人订单列表的界面

(3) 订单详细信息(OrderDetails.aspx)

该页面显示用户某一订单的详细信息，界面如图5-14所示。

****

图5-14 订单详细信息的界面

6、在线支付模块

这几页主要提供顾客选择网上银行在线支付的功能，界面如图5-15所示。



图5-15 网上银行在线支付的界面

7、后台管理模块

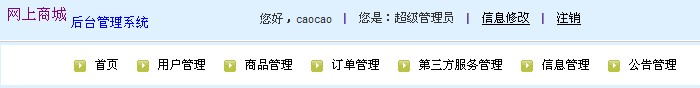
在项目设计中，考虑到安全性和独立性，将后台管理模块单独作为一个Web项目，但同属 EShop解决方案。

后台管理模块是一个完整的电子商务系统不可或缺的部分，主要包括以下几个功能页面：

1. 权限控制的实现

在EShop的后台模块中存在两种管理员角色：一种为“超级管理员”，具有管理员模块的一切权限；一种为“普通管理员”，具有被超级管理员赋予的特定功能。

像用户管理一样，在管理员管理页面也需要在左侧显示功能导航目录。不同的是，管理员具有多种角色，所以针对不同的角色应该显示不同的功能列表，其界面如图5-16所示。



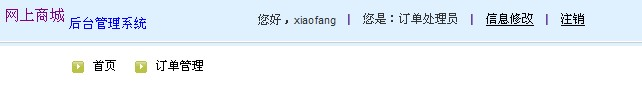


图5-16 管理员功能导航目录的界面

(2) 查询、修改会员信息(CustomerList.aspx)

在这个页面，管理员可以分页查看用户的基本信息。界面如图5-17所示。



图5-17 查询、修改会员信息界面

(3) 订单信息查看（OrderList.aspx）

超级管理员和订单处理员可以分页查看订单的信息。界面如图5-18所示。

****

图5-18查询订单信息界面

## 5.3 系统测试

系统测试是管理信息系统开发周期中一个十分重要而漫长的阶段。其重要性体现在它是保证系统质量与可靠性的最后关口，是对整个系统开发过程包括系统分析、系统设计和系统实现的最终审查。

系统测试的对象不仅仅是源程序，而是整个软件，即程序和文档。系统测试的目的是发现软件的错误，系统测试中可能发现的错误包括：功能错误、系统错误、过程错误、数据错误、编码错误。

在系统测试工作中，要事先准备好测试方案和测试数据，并在测试过程中如实填写测试记录，以便于进行系统的更新和维护。

在本系统的测试过程中，我们运用了各种系统测试方法，对整个系统的主页、各功能模块及页面、数据库操作、程序代码和整体功能分别进行了详细的测试，保证了系统的质量、可靠性和可维护性。

### 5.3.1 系统测试的原则

1、测试工作由非原来开发软件的个人或小组承担。

2、设计测试方案时，不仅包括确定的输入数据，而且应包括从系统功能出发预期的测试结果。只要将运行结果与预期测试结果进行对比即可发现有无错误。

3、测试用例不仅包括合理、有效的数据，还要包括无效的或不合理的输入数据。

4、不仅要检验程序是否作了该做的事，还要检验程序是否同时作了不该做的事。

5、软件中存在错误的概率和已经发现错误的个数是成正比的。

6、保留测试用例，作为软件文档的组成部分。

### 5.3.2 系统测试的方法及步骤

1、对软件测试的主要方法有：源程序通过编译后，先经过人工测试，然后再进行机器测试。

(1) 人工测试是采用人工方式进行，目的在于检查程序的静态结构，找出编译不能发现的错误。其主要有三种方法：个人复查、走查、会审。然而这些测试只是针对小规模程序常用的方法，效率不高。

(2) 机器测试是运用事先设计好的测试用例，执行被测程序，对比运行结果与预期结果的差别来发现错误。机器测试有黑盒测试与白盒测试两种方法。

2、系统测试工作一般有以下步骤：单元测试、组装测试、确认测试和系统测试。

### 5.3.3 系统测试的用例及结果

1、界面检查

打开系统页面，对照其中的界面名称及功能名称是否正确；对每个界面的各个下拉菜单、复选按钮进行检查，检查其是否有错误条目或缺少项目；并对链接按钮进行检查，检查其是否能够实现正常链接、安全返回的功能。

2、功能测试

检查每个模块下的显示、添加、修改、删除等功能是否可以正确实现，进而检查程序代码是否存在错误、数据库连接是否正常。

3、综合测试

在对各个模块分别调试之后，进行系统的综合测试，检测了几个模块共同调用一个数据库中的表对执行结果的影响。并检测了整体功能实现情况，最终实现了测试结果正确性。

4、测试用例

以后台管理模块中的添加商品信息为例进行功能的测试：输入要进行添加的商品信息，点击添加按钮后，却不能执行，通过仔细查看源代码，寻找错误，得出了我们所看到的正确的代码。如下：

添加商品信息(AddNewGoods.aspx)的部分代码实现：

try

{

#region--保存包括头像在内的所有数据--

//设置根目录

string root = Server.MapPath("~/");

//设置商品图片保存路径

string photopath = "UpLoadImage/goods\_photos/";

//设置商品图片缩略图的保存路径

string thumbn\_photopath = "UpLoadImage/goods\_thumbnails/";

#region--获取图片的格式(即扩展名)--

//验证图片的扩展名

bool fileOK = false;

//获取文件的扩展名

string fileExtension = null;

if (FileUpload1.HasFile)

{

//返回文件的扩展名

fileExtension = System.IO.Path.GetExtension(FileUpload1.FileName).ToLower();

//设置限制的文件类型

String[] allowExtension = { ".jpg", ".gif", ".png", ".jpeg" };

//判断用户选择的文件类型是否受限制

for (int i = 0; i < allowExtension.Length; i++)

{

if (fileExtension == allowExtension[i])

{

fileOK = true;

}

}

//大于0.5M的文件不能上传

if (FileUpload1.PostedFile.ContentLength > 512000)

{

CommonMethods.ShowMessage("图片不能超过0.5M!");

return;

}

//小于1K的文件不能上传

if (FileUpload1.PostedFile.ContentLength < 1024)

{

CommonMethods.ShowMessage("图片太小了!");

return;

}

}

#endregion

//获取被选中编辑的商品ID

string username = Guid.NewGuid().ToString();

//设置自定义的商品图片文件名称

string newfilename = DateTime.Now.Ticks.ToString();

newfilename = newfilename.Replace("-", "");

newfilename = newfilename.Replace(" ", "");

newfilename = newfilename.Replace(":", "");

newfilename = username + "\_" + newfilename;

//新图片文件全名称，加上扩展名

newfilename = newfilename + fileExtension;

//设置缩略图的名称

string thumbn\_newfilename = "thumbn\_" + newfilename;

//设置服务器上面存储的【图片原文件】的相对路径

string mpath = photopath + newfilename;

//设置服务器上面存储的【缩略图文件】的相对路径

string thumbn\_mpath = thumbn\_photopath + thumbn\_newfilename;

//设置服务器上面存储的带图片水印图片的保存路径，【商品图片采用带水印图片的缩略图】

string aawatermark\_tupian\_path = "UpLoadImage/goods\_watermark\_tupian/" + "watermark\_tupian\_" + thumbn\_newfilename;

if (!fileOK)

{

CommonMethods.ShowMessage("图片的格式不正确!");

return;

}

//验证图片是否已经存在

if (dalogic.CheckHeadImage(mpath) == true)

{

CommonMethods.ShowMessage("该图片已存在!");

return;

}

if (fileOK && !dalogic.CheckHeadImage(mpath))

{

//首先将路径等数据存至数据库

EShop\_Goods newgoodsEntity = new EShop\_Goods();

newgoodsEntity.ID = username;

newgoodsEntity.Goods\_Name = this.txtGoodsName.Text;

newgoodsEntity.Goods\_Class\_Big = null;

//保存图片路径，目前为方便使用只启用小图片

newgoodsEntity.Goods\_Img\_Small = aawatermark\_tupian\_path;

newgoodsEntity.Goods\_Price\_Market = Convert.ToInt32(this.txtMarketPrice.Text);

newgoodsEntity.Goods\_Price\_EShop = Convert.ToInt32(this.txtEShopPrice.Text);

newgoodsEntity.Goods\_Stock = Convert.ToInt32(this.txtStore.Text);

newgoodsEntity.Goods\_Desc = this.txtDesc.Text;

//新增goods时将收藏数设置为0

newgoodsEntity.Goods\_Colletion = 0;

newgoodsEntity.Goods\_Merchant\_ID = this.DDLMerchant.SelectedValue;

//这里两个DDL赋值绑定有点问题，需要修改

newgoodsEntity.Goods\_Class\_Big = this.DDLBigClass.SelectedValue;

newgoodsEntity.Goods\_Class\_Middle = this.DDLMidClass.SelectedValue;

dalogic.InsertIntoEShop\_Goods(newgoodsEntity);

//然后在服务器文件目录下【保存原图片数据】

FileUpload1.PostedFile.SaveAs(root + photopath + newfilename);

//再在服务器文件目录下【保存缩略图数据】

ImageManage.MakeThumbnail(root + photopath + newfilename, root + thumbn\_photopath + thumbn\_newfilename, 130, 130, "Cut");

#region--对缩略图【添加水印文字】--

//设置生成的文字水印图片的存放路径

string watermarkpath = root + "UpLoadImage/goods\_watermark\_wenzi/" + "watermark\_wenzi\_" + thumbn\_newfilename;

ImageManage.AddShuiYinWord(root + thumbn\_photopath + thumbn\_newfilename, watermarkpath);

#endregion

#region--对缩略图【添加水印图片】--

//设置预定水印图片的路径

string watermark\_pic\_path = root + "UpLoadImage/watermark\_pic/" + "shuiyin.jpg";

//设置生成的图片水印图片的保存路径

string watermark\_tupian\_path = root + "UpLoadImage/goods\_watermark\_tupian/" + "watermark\_tupian\_" + thumbn\_newfilename;

ImageManage.AddShuiYinPic(root + thumbn\_photopath + thumbn\_newfilename, watermark\_tupian\_path, watermark\_pic\_path);

#endregion

CommonMethods.ShowMessage("新增成功!");

this.ButtonSubmit.Enabled = false;

#endregion

}

}

catch (Exception aa)

{

CommonMethods.ShowMessage("错误：" + aa.ToString());

return;

}

## 5.4 系统转换

系统转换就是指新系统代替旧系统的过程，即旧系统停止使用，新系统开始运行。系统转换的目的是保证新老系统进行平稳而可靠的交接，最后使整个新系统正式交付使用。系统转换过程由项目开发人员、系统操作员、用户单位领导以及业务部门的通力协作才能完成，这是系统开发周期中动用人力物力最多的一个步骤，因此非常需要相互间的配合与协调。

系统转换的内容主要包括数据文件转换、人员设备和组织机构的工作流程的改造和调整，有关资料的建档和移交等工作。系统转换方式可分为：直接转换、并行转换、分段转换。

在比较了各种系统转换方式的特点和要求，并考虑到公司工作特点和要求，本系统的转换采用分段转换的方式，这样可以保证公司正常工作运转和新系统顺利投入使用。

## 5.5 系统维护

系统测试之后，我们进入了试运行及维护阶段，这一阶段的目的是保证管理信息系统正常而可靠地运行，并能使系统不断得到改善和提高，以充分发挥系统本身的作用。在整个系统的运行过程中，系统的维护和管理是始终贯穿其中的，包括进行系统应用程序的维护、代码的维护、数据的备份与恢复、硬件设备维护。系统在运行时，也要随着环境的变化根据不同的需求及变化对系统进行必要的修改，使得系统功能更加完善。

1、系统维护工作的内容

系统维护工作贯穿于系统的整个运行过程中，包括：系统应用程序的维护、数据的维护、代码的维护、硬件设备维护，系统维护的重点是系统应用软件的维护工作。而系统维护工作不应总是被动的等待用户提出要求后才进行，应进行主动的预防性维护。

2、系统维护的组织与管理

系统维护性工作不仅是技术性工作，为了保证系统维护性工作的质量，需要付出大量的管理工作。系统投入运行后，设有系统管理员，专门负责整个系统维护的管理工作。系统维护要按照严格的步骤进行，防止未经允许擅自修改系统。因为无论是用户直接找程序人员还是程序人员自行修改程序，都将引起混乱，如不及时更新文档造成程序与文档不一致，多个人修改的不一致，以及缺乏全局考虑的局部修改。为了评价维护的有效性，确定系统的质量，记载系统所经历的维护内容，要将维护工作的全部内容以文档的规范化形式记录下来。维护就意味着修改，修改对子系统来讲是有副作用的。

第6章 结 论

## 6.1系统总结

Web开发技术的飞速发展促进了电子商务的普及。目前，越来越多的商业活动开始迁移到Internet中来进行，一种新的购物方式－在线销售服务已成为一种必然的趋势，这种购物方式无疑给商家和消费者带来极大的便利，使以前很多不可能的商业活动变得异常快捷。

本文研究了一种基于数据关联规则的网上购物系统。其开发主要包括后台数据库的建立、后台管理以及前台页面的Web设计。系统使用Microsoft公司以C#为核心语言的ASP.NET开发工具，利用微软提供的IIS建立运行环境，再结合Microsoft SQL Server 2008建立数据连接关系。利用其提供的各种组件，首先在短时间内建立数据库，然后，对数据库进行分析与建立页面，不断修正与改进，直到功能基本实现的可行系统。系统采用了B/S结构，以Web为中心，采用TCP/IP、HTTP传输协议、客户端通过浏览器访问Web以及与Web相连的后台数据库，为在线销售服务系统的管理提供了较好的工具。

总体来说，本系统的设计具有良好的扩展能力并且便于维护，使用较为简便，对于管理人员来说是较好的管理工具，但是由于时间和其他方面的限制原因，系统还有很多方面需要改进。我相信在以后的学习生活中定能一步一步的去完善这些，使在线销售服务系统功能越来越强大。

## 6.2系统展望

### 6.2.1电子商务系统的发展趋势

现阶段我国大部分电子商务网站功能单一，大部分是搞电子市场商情，在网上发布广告、电子目录、电子查询、网上互通商品信息；少数搞电子交易（电子洽谈、订购），利用网上进行商务洽谈,签订购货合同,交换文本及单证，进行交易；能完成网上购物、网上支付的较少，即网上营销，网下支付。

随着我国发展电子商务的环境（网络基础建设等运行环境、法律环境、市场环境网上支付、信息安全、认证中心建设、系统相关核心技术的成熟等条件）逐步完善,电子商务将向纵深挺进，新一代的电子商务将浮出水面，取代目前简单地依托“网站＋电子邮件"的方式。电子商务企业将从网上商店和门户的初级形态，过渡到将企业的核心业务流程、客户关系管理等都延伸到Internet上，使产品和服务更贴近用户需求。互动、实时成为企业信息交流的共同特点，网络成为企业资源计划、客户关系管理及供应链管理的中枢神经。企业将创建、形成新的价值链，把新老利益相关者联合起来，形成更高效的战略联盟，共同谋求更大的利益。

电子商务系统将实现商务高级电子化、网络化、智能化，开展协同电子商务，全面实现信息流、资金流、物流等三流的网络化。实现支付型电子商务与现代物流，网上订货与企业内部ERP结合，及时精良生产，实现零库存。从产品的设计研发、生产制造、产品交货、物流配送、财务处理、甚至是最后的成效评估等，都通过电子形式使交易各方能够同步作业。

### 6.2.2系统体系结构的发展趋势

1、基于B／S结构与C／S结构结合的体系结构：

B／S与C／S这两种技术是各有利弊的。C／S技术是20年前的主流开发技术，它主要局限于内部局域网的需要。因而缺乏作为应用平台的一些特性，难以扩展到互联网这样的环境上去，而且要求开发者自己去处理事务管理、消息队列、数据的复制和同步、通信安全等系统级的问题。这对应用开发者提出了较高的要求，而且迫使应用开发者投入很多精力来解决应用程序以外的问题。这使得应用程序的维护、移植和互操作变得复杂，成了C／S的一大缺陷。

但是，与B／S结构相比，C／S技术发展历史更为“悠久”。从技术成熟度及软件设计、开发人员的掌握水平来看，C／S技术更成熟、更可靠。在某些情况下，采用100％的B／S方式将造成系统响应速度慢、服务器开销大、通信带宽要求高、安全性差、总投资增加等问题。而且，对于一些复杂的应用，B／S方式目前尚没有合适方式进行开发。

2、采用这种结构优点在于：

(1) 充分发挥了B／S与C／S体系结构的优势，弥补了二者不足。充分考虑用户利益，保证浏览查询者方便操作的同时也使得系统更新简单，维护简单灵活，易于操作。

(2) 信息发布采用B／S结构，保持了瘦客户端的优点。装入客户机的软件可以采用统一的WWW浏览器。而且由于WWW浏览器和网络综合服务器都是基于工业标准，可以在所有的平台上工作。

(3) 数据库端采用C／S结构，通过ODBC／JDBC连接。这一部分只涉及到系统维护、数据更新等，不存在完全采用C／S结构带来的客户端维护工作量大等缺点。并且在客户端可以构造非常复杂的应用，界面友好灵活，易于操作，能解决许多B／S存在的固有缺点。

(4) 对于原有的基于C／S体系结构的应用，只需开发用于发布的WWW界面，就可非常容易地升级到这种体系结构，并保留原来的某些子系统。这样就充分地利用现有系统的资源。

(5) 通过在浏览器中嵌入ActiveX控件可以实现在浏览器中不能实现或实现起来比较困难的功能。例如通过浏览器进行报表的应答。

(6) 将服务器端划分为WEB服务器和WEB应用程序两部分。WEB应用程序采用组件技术实现三层体系结构中的逻辑部分，达到封装的目的。

B／S结构与C／S结构各具优缺点，建立C／S、B／S结构相结合的网络构架已成为必然趋势。怎样结合B／S与C／S开发系统是开发MIS（管理信息系统）系统普遍关注的问题。在应用过程中，应结合实际情况，并根据实际情况进行系统的选型与构建，从而开发出高效、安全的应用系统。

致 谢

本篇论文是在王海龙老师的精心指导下完成的，从选题到论文内容都给予了精心的指导和严格的教诲，无论从学术水平还是学术造诣上都使我受益非浅,在系统的设计中，自己动手，亲身经历了从学习、研究到社会实践的过程。不仅是我学到了许多社会经验，更重要的是让我开拓了眼界，是我意识到现代化科学技术的发展，以及它的优越性。

在这次毕业论文设计中，我亲身尝试到了作为一个缺乏社会实践经验的学生所面临的困难。通过对本系统的开发使我对书本上的知识点和理论有了更深一层的了解，使我学会了如何利用所学的知识,把理论结合于实践中,进一步掌握管理信息系统开发的规律,从而对本专业有了更深刻的认识。此后,我会加倍努力的去研究,在熟练的基础上掌握更多的新型技术,来提高自己的业务水平。

通过这一阶段的学习和研究，该项目终于完成，虽然耗费了许多精力和时间，但从另一方面它验证了我的知识和能力，让我有了不断提高自身素质的意识。

在这里我衷心的感谢我的指导老师王海龙老师对我的指导和教诲。没有您的启发和帮助，我是不会这么顺利圆满地完成论文的工作的，非常感谢！

参 考 文 献

1 郑耀东. C#从入门到实践.北京：清华大学出版社，2009.

2 王国胜. C#.Net 基础与案例开发详解.北京：清华大学出版社，2009.

3 微软公司. 基于C#的Windows应用程序设计.北京：高等教育出版社，2004.

4 赵俊荣. SQL Server 2005数据库技术及应用.北京：高等教育出版社，2007.

5 郭伟伟，刘瑞阳. JavaScript全程指南. 北京：电子工业出版社，2008.

6 Dino Esposito(意). ASP.NET 3.5 核心编程. 北京：清华大学出版社，2009.

7 Matthew MacDonald. Beginning ASP.NET 3.5 in C# 2008 From Novice to Professional Second Edition. Computing reviews，2007,11.

8 Cristian Darie，Wyatt Barnett. Build your own ASP.NET 3.5 Website using C# & VB (3rd ed.). SitePoint Pty.Ltd，2008.

9 The Economist. Online shopping - Selling becomes sociable. The economist，2010.

10 ROGERS.PRESSMAN. Software Engineering A Practitioner’s Approach (Fifth Edition). Thomas Casson，2005.

附录A 中文翻译

译文1：《Build your own ASP.NET 3.5 Website using C# & VB (3rd ed.)》

**第15章** **ASP.NET AJAX**

在这本书中，我们构建了 ASP.NET Web 窗体，任何用户互动后, 把所有网页的数据传回服务器。如果你在一台快速的计算机上做本地开发，在这种做法下你可能不注意到任何问题,但一旦您的应用程序部署到互联网，它会变得相当缓慢并且延迟用户响应。然而，利用 ASP.NET AJAX，你可以不费吹灰之力就能把一个又慢又笨重的web应用程序流畅的运行起来。

在这一章中，我们将学习有关：

■Ajax 是什么以及它是如何工作的

■ASP.NET AJAX

■UpdatePanel 和 UpdateProgress 控件

■使用触发器

■ASP.NET AJAX控件工具包

■ValidatorCalloutExtender 控件扩展

■UpdatePanelAnimationExtender控件扩展

**什么是Ajax？**什么是Ajax？你会问。首字母缩写词本身，杜撰的杰西·詹姆斯·阁楼，2005年，代表异步JavaScript 和XML。总括来说，这个词表示的方法和技术，允许开发人员都可以感受到的web应用程序交互和响应得多。显著的例子，利用Ajax的应用程序包括 Gmail、谷歌地图、雅虎邮件、皮艇、Flickr 和 Facebook。此外，几乎每个重要的web站点，您今天可能与交互利用一些Ajax概念，以改善用户体验以某种方式。

在技术层面上，Ajax 使用几个关键问题：

■最关键的一点是客户端 JavaScript 对象。原来微软发明的互联网浏览器4，现在已经成为所有的浏览器的标准功能。如此强大功能的对象允许开发人员使用 JavaScript，在不提交整个页面的情况下来使HTTP获取和发送请求，然后访问服务器响应的内容。请注意Xml部件的名称是有点误导。在实践中，你可以自由地选择响应数据的格式XML、JSON、一小个HTML代码段，或可以通过 HTTP 发送的任何格式。

■第二个最重要的技术是浏览器 DOM 脚本：使用 JavaScript 的文档对象模型操纵。JavaScript 需要创造XmlHttpRequest对象和提出要求，但也把原始数据回来的要求，解析出来并将其集成到用户的页面上。

■Ajax 应用程序的最后一站,是站在基于标准的 HTML 和 CSS。HTML 提供结构和内容，而CSS则用来控制界面的显示。

和任何一种技术一样，Ajax并不是没有挑战。首先，用Ajax做什么，你需要由JavaScript在浏览器中启用。从方便的角度来看，大量的互动网站会造成人们不方便使用如屏幕阅读器等的辅助阅读技术。Ajax的应用程序在一个网站上可以巧妙地改变了方式，用户可以找到混乱浏览器的行为--例如，他们可能无法使用后退按钮，或准确找到书签页。最后，如果你的网站内容不能被任何其他手段比如JavaScript获取，搜索引擎将无法索引你的内容。

Ajax听起来吓人，并且对原来的开发团队来说仍然有复杂的问题需要解决。但是，随着Ajax的成熟，并成为现代Web开发的标准组件，一些工具的建立都是为了使“Ajaxifying”你的应用程序非常容易。您可能听说过，甚至这些二手的工具，包括jQuery，Prototype和MooTools的部分。但是，作为一个新进ASP.NET开发人员，你真的需要熟悉自己的一个工具，特别是：微软的ASP.NET AJAX。

**ASP.NET AJAX**

ASP.NET AJAX中，最初名为ASP.NET Atlas的，原本是单独的附加到ASP.NET2.0中，但已作为ASP.NET的3.5.6库，本身就有三个组成部分的核心部分包括在内。第一个是客户端JavaScript库与Microsoft AJAX Library中着称，它提供了跨浏览器的功能框架，这是由ASP.NET AJAX服务器端组件使用的设置。其次，在服务器方面，微软开发的HTTP处理程序和模块设置为包装标准的基于XML的Web服务，让他们响应Ajax请求使用JSON。最后，有重点的组成部分，我们将在这一章的服务器端控件，使得您的应用程序非常容易Ajaxifying上。

包括在核心ASP.NET AJAX 库的几个关键的服务器控件：

ScriptManager控件是相当枯燥，默默无闻，但它绝对是至关重要的。正如其名称所示，它管理你的网页上注册客户端脚本，喜欢做一些处理，你的Ajax控件有他们需要的脚本，更重要的是，这些脚本只包含一次平凡的工作。

Timer 控件提供管道在规定的时间间隔来执行“回发”操作。再次，它是一个简单的，又往往被忽视的控制，但它使这些操作非常方便。

最后一个组成部分是相互关联的UpdatePanel控件和UpdateProgresscontrols 控件。UpdatePanel中，当一个ScriptManager使用，允许开发者使用的浏览器，而无需重新加载整个页面部分呈现给其中一个页面的不同部分被更新独立的程序。 UpdateProgress控件通知用户的是，部分呈现操作正在进行中。

虽然它可能不像有品种繁多的服务器控件的方式，但提供的这些核心功能，可以让我们在几乎所有使用ASP.NET2.0或更高版本的网站非常快速、方便地添加Ajax功能。

译文2：《Beginning ASP.NET 3.5 in C# 2008 From Novice to Professional Second Edition》

**第十五章 ADO.NET基础**

在本书开始，您了解到，ASP.NET只是微软的雄心勃勃的.NET平台一个组成部分。正如你所知，.NET中还包括新的语言，一个新的哲学，为跨语言整合，更简单的方法来部署代码和一个类的工具包中，让您尽一切所能处理错误来分析XML文档。在这一章中， 您将探讨另一个.NET框架众多功能之一：ADO.NET数据访问模式。

很简单，ADO.NET是一项.NET应用程序用来与数据库交互的技术。在这一章中，你会了解ADO.NET和为它提供功能的对象的成员。您还可以学习如何让这些对象运作起来创造简单的检索和更新数据库记录的网页。然而，你不会了解使用ADO.NET的最简单的方法——在ASP.NET数据绑定的帮助下。虽然数据绑定是一个功能强大的和实用的功能，每一个ASP.NET的开发者应开始就牢牢的掌握ADO.NET基础。这是因为您需要撰写自己的ADO.NET代码优化性能敏感的数据库程序，执行任务，数据未涵盖的数据绑定模式，并制定数据库组件（如描述的第23章）。一旦您掌握了这一章里的ADO.NET基础在，您将准备探索第16章和17章里的节约时间快捷的数据绑定模型。

**了解数据管理**

几乎每块软件代码永远都是基于数据工作而编写的。事实上，一个典型的web应用往往只是一个薄薄的用户界面外壳，在它上面由精密的数据驱动的代码来从数据库读取和写入的信息。通常，网站的用户不知道（或不关心）显示的信息来自一个数据库。他们只想能够搜索产品目录、订单，或检查其付款记录。

**数据库的角色**

最常见的管理数据方式是使用一个数据库。数据库技术尤其对商业软件有用，即通常需要设置的相关信息。例如，一个典型的销售程序的数据库包括的客户名单，产品名单，和从其它两个表中抽取的销售信息。这种类型的信息是使用关系模型的最好说明，这是构成所有现代数据库产品，包括SQL Server，Oracle，Microsoft Access的基础理念。

正如你可能知道的，一个关系模型将信息分解成最小的和最简洁的单位。例如，销售记录不储存的所有已售出产品的信息。相反，它只存储指向全部产品记录的product表的产品ID号，如图15-1。



图15-1 基本的表关系

虽然这是技术上是可行的，将数据组织成表，用一个或多个文件存储在硬盘驱动器里（或许像使用标准的XML一样），这种做法将不会很灵活。相反，一个web应用程序需要一个完整的关系数据库管理系统（关系型数据库管理系统），如SQL Server。该数据库管理系统处理数据的基础结构，确保最佳性能和可靠性。例如，RDBMS负责同时给多个用户提供数据存储，驳回无效数据，并使用事务一次性提交一组行为。

在大多数的ASP.NET应用程序里，您为某些任务需要使用一个数据库。以下是一些在网络应用程序中使用数据的基本例子：

•电子商务网站（如Amazon）使用详细的数据库存储产品目录。他们还在一系列巨大的编排好相关关系的表格中跟踪订单，客户，发货记录和库存信息。

•搜索引擎（如谷歌）使用数据库来存储网页的网址索引，链接和关键字。

•知识库（如微软支援）使用更少的结构性数据库，其中存储广阔大量的信息或链接到的各种文件和资源。

•媒体网站（如纽约时报）在数据库中存储他们的文章。

你可能不会有任何问题，考虑在你的ASP.NET应用程序中在哪里需要使用数据库技术。什么web应用无法从一个使用后端数据库存储潜在客户或联系人清单，记录用户的意见或一个简单的e-mail地址的提交表单的留言簿受益？这就是ADO.NET引人注意的地方。ADO.NET是一种技术，目的是让一个ASP.NET程序（或任何其他.NET程序，对于这个问题）获得的数据。

提示：如果你完全是一个的数据库新手，你可以使用视频教程http://msdn.microsoft.com/vstudio/express/sql/learning加速你对数据库基本概念的了解。在那里，你会找到超过9小时的讲授，介绍了如何使用免费的Visual Studio自带的SQL Server 2005 Express版。该教程从绝对基础，涵盖了诸如数据库的数据类型和表的关系，以及更高级的如全文检索，报告服务，以及网络安全等主题。

**在Web世界的数据库访问**

在Web应用程序中访问一个数据库，和访问中一个典型的C/S的桌面应用程序的数据库是完全不同的情景。大多数开发者在桌面世界磨练其数据库技能，当他们试图运用他们在单机应用程序中学到的技能应用到Web世界时遇到了严重的问题时。简单地说，Web应用程序带来两个新的因素：问题的规模和问题的状态。

问题是大规模多用户的网络性质可能导致的规模的问题。一个web应用程序可能有数百或数千同步使用的用户。这意味着它不能随便使用服务器内存或资源有限的如数据库连接。如果你设计一个应用程序，获取数据库连接，保持甚至是多余的几秒，其他用户可能会注意到一个明显的放缓。 如果你不仔细考虑数据库并发问题（换句话说，不同的用户重叠时发生的事），您可能遇到重要的令人头痛的事，如未能更新数据和不一致的数据。

注：当开发传统的C/S桌面应用程序时，规模的问题也会发生。不同的是，在多数C/S应用程序远远低于Web应用程序可能产生任何负面影响，因为典型负载（并发用户数）显着偏低。数据库的操作对C/S程序可能造成轻微的性能限制，但在一个web应用程序中，这个性能限制可能迅速膨胀，造成重大问题。

状态的国家的问题可能是由互联网断开连接的性质导致的。正如你所知，HTTP是一种无状态协议。当用户在一个ASP.NET应用程序请求一个网页时，Web服务器处理代码，返回提供的HTML同时立即关闭连接。尽管用户可能有错觉，他们和一个不断运行的应用程序交互，实际上他们只是收到了一连串的静态网页。

由于无状态性质的HTTP ，Web应用程序需要执行其所有工作空间的一个单一的请求。典型的做法是连接到一个数据库，读取资料，显示，然后关闭数据库连接。这一办法陷入困境，如果你希望用户可以修改检索到的信息。在这种情况下，应用程序需要一定数额的情报，以便能够找出原始记录，建立一个SQL语句以选中它，然后更新给它新的值。

幸运的是，ASP.NET和ADO.NET是为这些挑战设计的。当您以您的方式阅读这一章（及以下两章），您将学习如何安全和有效地处理数据库。

译文3：《Online shopping - Selling becomes sociable》

电子商务愈加趋于社交化，且与现实世界联系更加紧密。

在新创办的网站Swipely上，用户现在可以发布他们的购物信息。看重隐私的人可能会避犹不及，但对于那些在数码方面乐意出风头的人可谓是梦想成真。每当他们刷自己的信用卡或借记卡时，（Swipely公司的名字有“刷（卡）”的含义），交易内容便可列在网上以供其他用户讨论。“将购物转换为对话”是这个公司的理念。

Swipely是蓬勃发展的社会化电子商务领域中最新兴起的众多企业之一。在这个市场中，公司会将电子商务与社交网络或其他网络团体活动结合，目的在于同时改变线上线下的购物方式。Swipely的老板安格斯•戴维斯指出，因特网已经搅乱了内容产业，下一步会向商业进军的。

在上个世纪90年代末出现的第一代电子商务网站，仅仅是电子化的邮购目录。像Epinions这样的网站收集用户的评论和推荐，但不卖东西——而且有很多在网络公司萧条阶段纷纷倒闭了。只有亚马逊公司将销售与社交反馈结合在一起，并取得了极佳的效果。将反馈信息进行整合选择，便可在其他人购买情况的基础上提出建议了。

第二代电子商务公司则截然不同，只有少数几家在硅谷兴起。的确，这些公司会有实体店，而且有时也会鼓励顾客光临其实体店。他们的利润通常来自限时销售，即为注册用户提供一些在短期内大降价的商品。

率先举办“限时抢购”促销活动的Vente Privée来自法国服装产业（公司名字含义为“私人销售”）。雅克•安东尼•格朗琼，在2001年与7个伙伴一起创建了Vente Privée公司，他说甚至到现在公司的重心仍是离线销售。数百名设计师、摄影师及发型师共同负责策划和组织该公司的网络销售活动。Vente Privée的起步虽慢，但后劲十足。它在欧洲的5个当地网站共有注册会员1200万个，且今年将有可能带来8亿欧元（10亿美元）的收入。

Vente Privée的成功也激励了其他人。其中最有名的是来自纽约的Gilt Groupe，它效仿奢侈品零售商，在纽约进行样品销售。Gilt比Vente Privée的规模小，会员人数仅有250万，且本年的营业额将会达到4-5亿美元。Gilt的老板苏珊•琳恩说，他们想成为各种社交型商务的平台。其最近在美国成立了几家当地网站，提供“每日特价产品”。

Gilt Groupe现已转向另一类以城市为主的电子商务网站，这类网站为集体性购买提供便利。网站每天提供一家当地服务，如某餐馆的一餐饭、温泉疗养、租用昂贵的轿车，并且最多可以给予9折优惠（他们通常赚取销售价格的一半）。但必须在有一定会员数目购买的情况下，交易才可达成。因此买家会热衷于将消息转告给其他人，且通常是通过社交网站。

许多类似的网站如雨后春笋般涌现出来，但最成功的还是“团宝网Groupon”（Groupon一词是"group"团队和"coupon"优惠券的组合，买家将礼券打印出来以获得服务）。这家公司虽仅于2008年的下旬成立，但在29个国家已然拥有230个地方网站，1500万注册会员了。由于来自投资商的资金源源不断，因此这一项目也引发了抢地之战，在其他国家许多效仿团宝网的网站也纷纷涌现出来，如德国的CityDeal。

团宝网更看重的是人，而不是技术。它发展于总部设在芝加哥的网站The Point，该网站提供组织集体活动的工具。公司雇佣了将近两千人组成全球销售力量，来发掘有趣的地方商业，并且雇佣150个写手来对服务内容进行描述。团宝网的主管罗布•所罗门解释到，他们的公司希望可以最终帮助小型企业参与电子商务。

Forrester研究中心的Sucharita Mulpuru说，也许只有第三代电子商务网站才当之无愧“社交购物”这个标签。最近一批公司试图在“社区图景”的基础上发展业务，即在社交网络上建立朋友圈。他们使用虚拟货币和日渐流行的智能手机，帮助追踪消费者的所处方位。

ModCloth主要销售独立设计师设计的服装，它在Facebook上有一个非常活跃的论坛，并且会让顾客投票，来决定网站应该进那些货。另一家刚起步的公司Lockerz，为鼓励会员观看带广告的视频、邀请朋友以及参与网站活动，他们会给会员一些“积分”。会员便可用这些“积分”换取折扣。同样，Shopkick也会回馈消费者的线下活动，比如去参观实体店，或通过智能手机浏览商品。

对于新一代电子商务公司来说，线下世界与线上世界同样重要，Swipely就是个典型代表。通过自动上传交易数据，消费者便比较愿意与朋友分享他们是如何在现实生活中消费的，若还要他们将数据一一敲进电脑，他们估计就不太愿意做了。同时，消费者若不想让别人分享交易信息，还可以将其设为保密。

购物愈加趋向于社交化，真的会像戴维斯先生预测的那样会扰乱商业吗？很难预测第二代和第三代电子商务网站是否可以持续迅速增长。就像对网上拍卖一样，消费者也可能对迅速销售感到厌倦。即使团购也有其局限性。团宝网最大问题之一便是有成千上万的当地公司希望能在网站上露面，但每个网站每天只提供一个交易。因此团宝网便开始将一些网站个人化，即不同的用户将会根据年龄、性别、爱好等信息看到不同的交易。但若个人化过度，用户将只能看到那些自己想要的东西，而这他们早已知道了，这种购物便不那么有趣了。

来自创业投资公司Accel Partners的Sonali de Rycker预测到，无论私人公司和销售模式的命运如何，电子商务一定会更加趋于社交化的。零售业有几个顽固问题：吸引人们来逛的开销很高，来逛的人又不太消费，且让他们再次光顾也很难。社交型的电子商务便为所有这些问题提供了潜在的解决方案。所以期待你最喜欢的网站也添加些社交功能吧，只是就像那些创新者一样，先锋往往最易受到攻击。

附录B 外文原文

原文1：《Build your own ASP.NET 3.5 Website using C# & VB (3rd ed.)》

**C H A P T E R 1 5 ASP.NET AJAX**

Throughout this book, we’ve been building ASP.NET Web Forms that, after any user interaction, send all of the page’s data back to the server. You may not really notice any problems with this approach if you’re developing locally on a fast computer, but once your application is deployed to the Internet, it may appear to be quite slow and unresponsive to users. However, by harnessing ASP.NET AJAX, you can take a slow and clunky web application and make it sing with surprisingly little effort.

In this chapter we’ll learn about:

■what Ajax is and how it works

■ASP.NET AJAX

■the UpdatePanel and UpdateProgress controls

■using triggers

■the ASP.NET AJAX Control Toolkit

■the ValidatorCalloutExtender control extender

■the UpdatePanelAnimationExtender control extender

**What is Ajax?** What is Ajax, you ask? The acronym itself, coined by Jesse James Garret in 2005, stands for Asynchronous JavaScript and XML. In a nutshell, the term represents the methodologies and technologies that allow developers to make web applications feel far more interactive and responsive. Notable examples of applications that utilize Ajax include Gmail, Google Maps, Yahoo! Mail, Kayak, Flickr, and Facebook. Moreover, just about every significant web site you interact with today probably takes advantage of some Ajax concepts to improve the user experience in some way.

On a technical level, Ajax uses a few key technologies:

■The most critical piece is the JavaScript XmlHttpRequest object. Originally invented by Microsoft for Internet Explorer 4, it’s now a standard feature of all browsers. This very powerful object allows developers to use JavaScript to make HTTP GET and POST requests without submitting the whole page, and to then access the content of the server response. Note that the Xml part of the name is a bit misleading. In practice, you’re free to choose the format of the response data—XML, JSON, an HTML snippet, or just about any format that can be sent over HTTP.

■The second most important technology is browser DOM scripting: Document Object Model manipulation using JavaScript. JavaScript is required to create the XmlHttpRequest object and make the request, but also to take the raw data returned from the request, parse it, and integrate it back into the page for the user.

■The final leg on which Ajax applications stand is standards–based HTML and CSS. HTML provides the structure and the content, while CSS is used to control the presentation of the interface.

Like any technology, Ajax is not without its challenges. First and foremost, to do anything with Ajax, you need to have JavaScript enabled in the browser. From an accessibility viewpoint, heavily interactive web sites can be very difficult for people using assistive technology like screen readers to use. The application of Ajax within a web site can subtly alter the behavior of the browser in ways that users can find confusing-- for example, they may be unable to use the Back button, or accurately bookmark pages. Finally, if your web site content can’t be accessed by any means other than JavaScript, search engines won’t be able to index your content.

Ajax may sound intimidating, and for the original trail blazers there were a number of complex problems to solve. But, as Ajax has matured and become a standard component of modern web development, a number of tools have been created to make “Ajaxifying” your application very easy. You may have heard of—or even used—some of these tools, including jQuery, Prototype, and MooTools. However, as a budding ASP.NET developer, you really need to familiarize yourself with one tool in particular: Microsoft’s ASP.NET AJAX.

**ASP.NET AJAX**

ASP.NET AJAX, originally called ASP.NET Atlas, was originally a separate add-onto ASP.NET 2.0, but has been included as a core part of ASP.NET 3.5.6 The library itself has three components. The first is a set of client-side JavaScript libraries, known as the Microsoft AJAX Library, which provide the framework of cross-browser functionality that’s used by the ASP.NET AJAX server-side components. Second, on the server-side, Microsoft developed a set of HTTP handlers and modules to wrap the standard XML-based web services and allow them to respond to Ajax requests using JSON. Finally, there’s the component we’ll focus on in this chapter—the server-side controls that make Ajaxifying your application incredibly easy.

Included in the core ASP.NET AJAX library are just a few key server controls:

1. The ScriptManager control is rather boring and unsung, but it’s absolutely critical. As its name indicates, it manages the registration of client scripts on your web page, handling mundane tasks like making certain that your Ajax controls have the scripts they need and, more importantly, that these scripts are included only once.
2. The Timer control provides the plumbing to perform postback operations at defined intervals. Again, it’s a straightforward, often-overlooked control, but it’s very handy for those operations.
3. The final components are the interrelated UpdatePanel and UpdateProgresscontrols. The UpdatePanel, when used with a ScriptManager, allows a developer to use partial rendering—a procedure by which different sections of a page are updated independently, without requiring the browser to reload the entire page. The UpdateProgress control informs users that a partial rendering operation is in progress.

While it probably doesn’t seem like there’s much variety in the way of server controls, these provide the core functionality to let us very quickly and easily add Ajax functionality to just about any web site that uses ASP.NET 2.0 or later versions.

原文2：《Beginning ASP.NET 3.5 in C# 2008 From Novice to Professional Second Edition》

**C H A P T E R 1 5 ADO.NET Fundamentals**

At the beginning of this book, you learned that ASP.NET is just one component in Microsoft’s ambitious .NET platform. As you know, .NET also includes new languages, a new philosophy for cross-language integration, an easier way to deploy code, and a toolkit of classes that allows you to do everything from handling errors to analyzing XML documents. In this chapter, you’ll explore another one of the many features in the .NET Framework: the ADO.NET data access model.

Quite simply, ADO.NET is the technology that .NET applications use to interact with a database. In this chapter, you’ll learn about ADO.NET and the family of objects that provides its functionality. You’ll also learn how to put these objects to work by creating simple pages that retrieve and update database records. However, you won’t learn about the easiest way to use ADO.NET—with the help of ASP.NET data binding. Although data binding is a powerful and practical feature, every ASP.NET developer should start with a solid grasp of ADO.NET fundamentals. That’s because you’ll need to write your own ADO.NET code to optimize performance-sensitive database routines, to perform data tasks that aren’t covered by the data binding model, and to craft database components (as described in Chapter 23). Once you’ve mastered the basics of ADO.NET in this chapter, you’ll be ready to explore the timesaving shortcuts of the data binding model in Chapter 16 and Chapter 17.

**Understanding Data Management**

Almost every piece of software ever written works with data. In fact, a typical web application is often just a thin user interface shell on top of sophisticated data-driven code that reads and writes information from a database. Often, website users aren’t aware (or don’t care) that the displayed information originates from a database. They just want to be able to search your product catalog, place an order, or check their payment records.

**The Role of the Database**

The most common way to manage data is to use a database. Database technology is particularly useful for business software, which typically requires sets of related information. For example, a typical database for a sales program consists of a list of customers, a list of products, and a list of sales that draws on information from the other two tables. This type of information is best described using a relational model, which is the philosophy that underlies all modern database products, including SQL Server, Oracle, and even Microsoft Access.

As you probably know, a relational model breaks information down to its smallest and most concise units. For example, a sales record doesn’t store all the information about the products that were sold. Instead, it stores just a product ID that refers to a full record in a product table, as shown in Figure 15-1.



Figure 15-1. Basic table relationships

Although it’s technically possible to organize data into tables and store it on the hard drive in one or more files (perhaps using a standard like XML), this approach wouldn’t be very flexible. Instead, a web application needs a full relational database management system (RDBMS), such as SQL Server. The RDBMS handles the data infrastructure, ensuring optimum performance and reliability. For example, the RDBMS takes the responsibility of providing data to multiple users simultaneously, disallowing invalid data, and committing groups of actions at once using transactions.

In most ASP.NET applications, you’ll need to use a database for some tasks. Here are some basic examples of data at work in a web application:

• E-commerce sites (like Amazon) use detailed databases to store product catalogs. They also track orders, customers, shipment records, and inventory information in a huge arrangement of related tables.

• Search engines (like Google) use databases to store indexes of page URLs, links, and keywords.

• Knowledge bases (like Microsoft Support) use less structured databases that store vast quantities of information or links to various documents and resources.

• Media sites (like The New York Times) store their articles in databases.

You probably won’t have any trouble thinking about where you need to use database technology in an ASP.NET application. What web application couldn’t benefit from a guest book that records user comments or a simple e-mail address submission form that uses a back-end database to store a list of potential customers or contacts? This is where ADO.NET comes into the picture. ADO.NET is a technology designed to let an ASP.NET program (or any other .NET program, for that matter) access data.

Tip: If you’re a complete database novice, you can get up to speed on essential database concepts using the video tutorials at http://msdn.microsoft.com/vstudio/express/sql/learning. There, you’ll find over nine hours of instruction that describes how to use the free SQL Server 2005 Express Edition with Visual Studio. The tutorials move from absolute basics—covering topics such as database data types and table relationships—to more advanced subject matter such as full-text search, reporting services, and network security.

**Database Access in the Web World**

Accessing a database in a web application is a completely different scenario than accessing a database in a typical client-server desktop application. Most developers hone their database skills in the desktop world and run into serious problems when they try to apply what they have learned with stand-alone applications in the world of the Web. Quite simply, web applications raise two new considerations: problems of scale and problems of state.

Problems of scale are the problems that can result from the massively multiuser nature of the Web. A web application has the potential to be used by hundreds or thousands of simultaneous users. This means it can’t be casual about using server memory or limited resources such as database connections. If you design an application that acquires a database connection and holds it for even a few extra seconds, other users may notice a definite slowdown. And if you don’t carefully consider database concurrency issues (in other words, what happens when the changes from different users overlap), you can run into significant headaches, such as failed updates and inconsistent data.

Note: Problems of scale can occur when developing traditional client-server desktop applications. The difference is that in most client-server applications they are far less likely to have any negative effect because the typical load (the number of simultaneous users) is dramatically lower. Database practices that might slightly hamper the performance of a client-server application can multiply rapidly and cause significant problems in a web application.

Problems of state are problems that can result from the disconnected nature of the Internet. As you already know, HTTP is a stateless protocol. When a user requests a page in an ASP.NET application, the web server processes the code, returns the rendered HTML, and closes the connection immediately. Although users may have the illusion that they are interacting with a continuously running application, they are really just receiving a string of static pages.

Because of the stateless nature of HTTP, web applications need to perform all their work in the space of a single request. The typical approach is to connect to a database, read information, display it, and then close the database connection. This approach runs into difficulties if you want the user to be able to modify the retrieved information. In this scenario, the application requires a certain amount of intelligence in order to be able to identify the original record, build a SQL statement to select it, and update it with the new values.

Fortunately, both ASP.NET and ADO.NET are designed with these challenges in mind. As you work your way through this chapter (and the following two chapters), you’ll learn how to deal with databases safely and efficiently.

原文3：《Online shopping - Selling becomes sociable》

E-commerce is becoming more social and more connected to the offline world.

Those who cherish privacy will recoil in horror, but for digital exhibitionists it is a dream. At Swipely, a web start-up, users can now publish their purchases. Whenever they swipe their credit or debit card (hence the service’s name), the transaction is listed on the site—to be discussed by other users. “Turn purchases into conversations” is the firm’s mantra.

Swipely is among the latest entrants in the growing field of social commerce. Firms in this market combine e-commerce with social networks and other online group activities. They aim to transform shopping both online and off. Angus Davis, Swipely’s boss, points out that the internet has already disrupted the content industry. Commerce will be next, he says.

The first generation of e-commerce sites, which hit the web in the late 1990s, were essentially digitised mail-order catalogues. Websites like Epinions collected user reviews and recommendations, but they did not sell anything—and many collapsed during the dotcom crash. Only Amazon brought together selling and social feedback, to great effect. By means of collective filtering, it made suggestions based on other buyers’ purchases.

The second generation of e-commerce firms is quite different. Few emerged from Silicon Valley. Indeed, they tend to have offline roots, and sometimes seek to drive customers to actual shops. Many make their money from flash sales—brief offers of steep discounts on products—that are advertised to registered members.

The pioneer of flash sales, Vente Privée, grew out of the French apparel industry (the name means “private sale”). Even today, its centre of gravity is offline, says Jacques-Antoine Granjon, Vente Privée’s boss, who founded the firm in 2001 along with seven partners. Hundreds of designers, photographers and hairstylists organise its online sales events. After a slow start, Vente Privée has been growing quickly. Its five local sites in Europe have more than 12m members and are expected to bring in about €800m ($1 billion) in revenues this year.

Vente Privée’s success has inspired others. The best known is Gilt Groupe, which emulates the sample sales of luxury retailers in New York, where it is based. Gilt is smaller than Vente Privée. It has only 2.5m members and expects to turn over between $400m and $500m in revenues this year. Gilt wants to become a platform for all sorts of social commerce, says Susan Lyne, its boss. It recently launched several local sites in America, offering “deals of the day”.

Gilt Groupe is straying into the territory of another clutch of city-based e-commerce sites, which facilitate collective buying. Every day these sites offer the service of a local business—a restaurant meal, a spa-treatment, the rental of an expensive car—at a discount of up to 90% (they generally keep half of the sale price). But a deal is struck only if a minimum number of members pounce. Buyers thus have an interest in spreading the word, which they do mostly on social networks.

Many such sites have sprung up. The most successful is Groupon (a combination of the words “group” and “coupon”, which buyers print out to pay for their service). Although the firm launched only in late 2008, it already operates some 230 local websites in 29 countries and boasts 15m subscribers. Flush with money from investors, it has embarked on a global land-grab, buying Groupon clones in other countries, such as Germany’s CityDeal.

Groupon is more about people than technology. It grew out of The Point, a Chicago-based website that offers tools to organise collective action. The firm employs a worldwide sales force of nearly 2,000 to identify interesting local merchants and 150 writers to describe the offers. It wants to be the company that finally allows small businesses to participate in e-commerce, explains Rob Solomon, its president.

Yet it may be a third generation of social-shopping sites that really deserves the label, says Sucharita Mulpuru of Forrester Research. The latest batch of firms try to build their business on top of the “social graph”: the network of friends spun on social networks. They make use of virtual currencies and the growing popularity of smart-phones, which can track consumers’ location.

ModCloth, which sells clothing from independent designers, has an active forum on Facebook and lets customers vote on which products the site should stock. Lockerz, another upstart, pays members “pointz” if they watch videos with advertisements, invite friends and do things with them. They can then use this currency to obtain discounts. Similarly, Shopkick rewards consumers for offline activities such as visiting stores and scanning products with their smart-phones.

For the new generation of e-commerce firms, the offline world is as important as the online one. Swipely is a good example. By uploading transaction data, the start-up makes it easy for customers to tell their friends how they are spending their money in the real world, something they probably would not do if they had to type all the data in. At the same time, customers can keep those transactions secret that they do not want to share.

Will making shopping more social really disrupt commerce as much as Mr Davis expects? It is hard to predict whether the second and third generations of e-commerce sites will continue their rapid growth. Consumers may tire of flash sales, as they did of online auctions. Even collective buying may have its limits. One of Groupon’s biggest problems is that tens of thousands of local firms want to be featured, but each of its sites offers only one deal per day. The outfit has started to personalise some of its sites, meaning that different users will see different deals depending on things like age, sex and interests. But if personalisation goes too far, and users see only the sort of things they already knew they wanted, this kind of shopping could become less fun.

Whatever the fate of individual firms and sales models, e-commerce is bound to become more social, predicts Sonali de Rycker of Accel Partners, a venture-capital firm. Retailing has several persistent problems: the high cost of attracting visitors, the low probability that they become buyers and the difficulty of getting them to come back. Sociable e-commerce offers potential solutions to all of them. So expect your favourite site to add social features, whereas many of the pioneers will end up with arrows in their backs, as innovators often do.