

分类号_____

学校代码_____

学号 M201276141

密级_____

硕士学位论文

基于 ASP.NET 的 B2C 电子交易系统 的设计与实现

学位申请人

学 科 专 业： 软件工程

指 导 教 师： 讲师

答 辩 日 期： 2015.01.12

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree for the Master of Engineering**

**The Design and Implementation of a B2C
Electronic Transaction System Based on
ASP.NET**

Candidate :

Major : Software Engineering

Supervisor :

January , 2015

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于 ☐ 保密， 在 _____ 年解密后适用本授权书。
☐ 不保密。

（请在以上方框内打“√”）

学位论文作者签名：

日期： 年 月 日

指导教师签名：

日期： 年 月 日

摘 要

随着互联网技术的发展,人们的生活节奏越来越快,购物已经不仅仅只是逛街、逛商场了,更多的包括淘宝、京东等电子购物网站,电子商城渐渐成为人们生活中不可缺少的一部分,也为工作繁忙的人们提供了很大便利。本文讨论的是一个基于 ASP.NET 技术的电子商城系统的设计与实现。

本文首先研究电子商务和电子商城的发展,以及国内外的现状分析,叙述了课题研究的目的和意义。根据国内外现有的相似网站,分析它们的功能,结合实际的使用感受,设计电子商城的功能,还有各个功能模块的划分和设计分析,各个功能的使用场景、方式和用例,确定它们的使用流程。在开发方案上,系统开发运用了 B/S 架构,减少了客户端的开发工作。在系统架构上,采用了比较通用的三层架构——表示层、业务逻辑层和数据访问层,三层架构技术具有灵活性、简易性、可用性、扩展性、安全性等优点,介绍了每层结构的功能作用及其所要负责的工作。其中,详细介绍了数据库的架构以及表的设计。详细阐述了部分重要功能模块的实现细节以及其实现运行的界面效果。

总体上,本系统实现了电子交易系统的主要功能,如用户的登录注册,购买商品、管理员对商品、用户和订单的管理等。系统虽实现预期功能,但其他的附属功能尚未实现,这些都需要在以后的版本中添加和考虑。

关键词: 服务器端脚本技术 浏览器/服务器结构 电子商务 电子商城
商对客模式

ABSTRACT

With the development of Internet technology, people's life in an increasingly fast pace, shopping is not just shopping, but includ Taobao, Jingdong electronic shopping website, E-mall has gradually become an indispensable part of people's life, but also provides great convenience for people with busy work. This article's discusses is the design and implementation of E-mall system based on the ASP.NET technology.

This article firstly studies the development of E-commerce and E-mall, as well as the analysis of the present situation of domestic and foreign, expounds the purpose and significance of this research. According to the domestic and foreign similar to an existing site, analysis of their function, combined with the actual use of the experience, design functional of E-mall, there are various functional modules and the design analysis, each function usage scenarios, and use case, determine their use process. In the plan of development, The development of system applies the B/S architecture, reduced the development work of the client. In the framework of the system, using generic three layer architecture, the three layer architecture technique has the flexibility, simplicity, availability, scalability, security, introduces the function of each layer structure and their work. Among them, in detail introduced the database architecture and the design and implementation of data table. Finally, to illustrate the running environment of the system, described in detail the implementation details of some main function module and their effect of realization operation interface.

Overall, this system has realized the main function of the electronic transaction system, such as user login and registration, the purchase of goods, the management of the goods, the user and order etc.. However, the system has achieved the anticipated function, but other functions are not affiliated to the system, these all need to add and considerat in later editions.

Key words: ASP.NET B / S structure E-commerce E-mall B2C

目 录

摘 要	I
Abstract.....	II
1 绪论	
1.1 课题研究的背景	1
1.2 课题研究的目的地意义.....	2
1.3 国内外研究现状的分析.....	3
1.4 课题研究内容.....	5
1.5 本文组织结构.....	5
2 系统开发方案分析及相干技术	
2.1 系统开发方案分析	7
2.2 系统开发相干技术	9
2.3 本章小结	16
3 系统的需求分析	
3.1 系统需求介绍.....	17
3.2 系统功能需求分析	17
3.3 系统非功能需求分析.....	22
3.4 本章小结	23
4 系统的架构与设计	
4.1 系统概要设计.....	24

4.2 系统数据库设计	25
4.3 系统功能设计	30
4.4 本章小结	33
5 系统的实现与测试	
5.1 系统实现环境	35
5.2 用户信息管理的实现	35
5.3 购物车的实现	36
5.4 支付货款的实现	39
5.5 商品管理的实现	40
5.6 订单管理的实现	42
5.7 用户管理的实现	44
5.8 浏览记录的实现	45
5.9 精准广告的实现	46
5.10 系统测试	48
5.11 本章小结	49
6 总结与展望	
6.1 全文总结	50
6.2 课题展望	51
致 谢	52
参考文献	53

1 绪论

1.1 课题研究的背景

在现代的社会中，计算机从出现到普及经历了短短的六十多年，现在社会的各个领域中都已经存在计算机，它已经成为人们的生活中最不可缺少的一部分。而互联网的出现使人们对计算机变得更加依赖，在其出现的四十多年里，人们的生活方式发生着翻天覆地的变化^[1]。电子商务就是在这样的时机中应运而生，它更是从根本上改变了人们的生活习惯。

电子商务作为时下最火的词之一，09年之后更是提高了这一词的火爆程度，跟以前有着翻天覆地的变化，双十一更是成为电子商务发展的里程碑^[2]。在09年之前，双十一只是一个很普通的日子，在09年，双十一成为一个节日——光棍节，并且迅速成为各大电商的节日盛典^[3]。2009年11月11日天猫商城的销售额为0.5亿元，2010年11月11日的销售额提高到了18倍，9.36亿元，2011年是33.6亿，2012年的当天就天猫商城的销售额达到了132亿元，2013年350.19亿元，2014年更是达到了571亿元。这只是淘宝在双十一当天的交易额，其他电商虽然没有这么大的销售额，但也是不容忽视的。1998年3月，中国第一笔互联网网上交易成功，到2013年底为止，中国在电子商务市场方面的交易额已经达到10.2万亿，电子商务只用了短短的十几年时间就发展到了目前的程度，其未来的发展前景也是不可估量的。

电子商城作为电子商务的一种实现形式，对它的发展起到了巨大的推动作用，如天猫网站对天猫商城的销售传奇起的作用也是巨大的，没有强大的网站平台是无法支撑起如此大量的消费群体。商务本来是一种很高级的东西，而电子商城就是将这种商务大众化了，使普通消费者很容易接触到，也正是这样的普通消费者才促进了电子商务有如今的发展^[4]。电子商城一般有三种形式，B2B，如阿里巴巴，B2C，如天猫商城、京东，C2C，如淘宝网。最大众化的是B2C和C2C，其实这两者的主要区别在于商家是个人还是品牌公司，对于消费者的使用方式没有什么大的影响，只是品牌公司的商家信誉或许更好一些。

电子商城是商家在互联网售卖虚拟或实体商品的平台，它作为一种虚拟商城，有着比真实商城更方便的地方，它将商品按照不同的方式进行分类，更方便消费者对同类型的商品进行对比，而且消费者也可以在成千上万的商品中查找到所需要的商品，作为一种互联网平台，也很方便通过多媒体的方式向消费者展示商品的信息，如广告信息、使用方法、注意事项等，为消费者更加全面和丰富的商品信息，满足消费者的各种消费愿望。

电子商城更加拉近了消费者与商家的距离，一般的消费模式是生产者生产出商品卖给批发商，批发商再买给零售商，最后才到消费者手中，经过这么多级别，每个级别都需要利润，正如羊毛出在羊身上，这些都需要消费者来买单，这大大增加了消费者支出^[5]，而电子商城可以让生产者将商品卖给消费者，没有那么多额外的消费，大大节约了成本，而且消费者可以更直接的与生产者沟通，也方便商品的改进。电子商城是虚拟的，不需要店面，更不需要豪华的装修，也大大节约了成本，这些都会降低消费者的支出。

1.2 课题研究的目及意义

电子商务是以互联网作为平台的，而电子商城是它的一个载体，就像商家需要一个店面向消费者出售商品，没有这个载体就没法使消费者与商家联系起来，为了推动商家与消费者的联系，方便商家出售其商品^[6]，也方便消费者更好的了解商品信息和购买商品，需要建设一个功能完善且强大的电子商城系统。这里需要了解用户，也就是商家和消费者的需求，从他们的角度出发，了解他们需要什么，注意什么和他们关注什么，只有了解了这些才能设计出一个功能完善的系统。

要设计和实现这样的一个电子商城系统还是会涉及到很多技术上的问题的，如编程语言的运用、网页制作、数据库设计等。通过来完成这样的一个实际的系统，可以更好的将我所学习的理论知识运用到实践中，也能将这些技术更好的运用到未来的实际工作中，也是增长实践经验的一种方式。通过实际的项目开发，可以提高我对这些技术的掌握程度，更熟练的使用这些技术。

1.3 国内外研究现状的分析

1.3.1 国外状况分析

全球的电子商务发展呈现一个不均衡的趋势，地区之间的发展差距也比较大，发展比较好的主要有美国、欧盟和亚洲三个地区。电子商务最早发展于美国，是在上世纪的七十年代，主要通过电子邮件的方式，所以美国也是目前电子商务发展最成熟的国家，全球的电子商务的发展也是由他带领着。电子商务在九十年代的发展比较迅速，主要借助于 Web 技术的信息发布系统，而这成为当时 Internet 的主要应用^[7]。这阶段的电子商务也较电子邮件阶段有所发展，但这还不是现在的电子商务，真正的电子商务，即 Electronic Commerce，发展于九十年代末，成为 Internet 的最终主要商业用途，商业信息主要通过 Internet 传递。

欧盟国家的电子商务起步比美国要晚一些，但是他的发展特别迅速，在全球电子商务上很快就成为了仅次于美国的地区。电子商务发展亚洲比欧盟更要晚一些，虽然亚洲是一个新秀，但是它的市场潜力是非常大的，然而亚洲近年的电子商务发展速度不理想，所占份额所占的份额也偏低，这样就说明亚洲是持续发展地区在全球电子商务上^[8]。

在美国，大型企业作为主导是它的 B2B 电子商务发展的特点，起点是集成供应链管理，降低成本是它的主要的目标。美国 B2B 电子商务发展的主流模式都是通过这些构成^[9]。在线购物的重要性也越来越明显，主要表现有其在零售市场中的占有率逐年递增。一份调查报告指出，已经有 4 家纯在线零售商跻身于最优秀的 10 家零售商。另外，人们都认为实体店的业绩在接下来的若干年中必然出现下滑的情况，这样认为的人包括观察员、分析师和零售店的管理人员，他们都有长期关注电子商务的发展形势^[10]。从 2008 年至今，美国已经有几大零售商不得不选择破产来保护自己，他们都是选择破产，而不是重组，也有很多零售商店纷纷关闭分店，以此来减少开支。在 2009 年估计约有 75000 家零售店关门，未来 3 到 5 年甚至将有 150000 家关门，这是“国际购物中心协会”做出的估计。线下的供应将在未来的几年内会有显著的减少，在线零售商将是这种状况的最大受益方。这些对于消费者来说其实

是更方便的，因为在线购物更方便购买^[11]。

1.3.2 国内状况分析

我国的电子商务发展起步比欧美国家晚，这主要由于我国的互联网发展比他们要滞后，但是我国的电子商务以一种稳定上升的趋势在发展，我国的电子商务在国内不同的地区发展也不尽相同，块状是目前主要呈现的比较典型的经济特征，较为发达的是东南沿海地区，快速发展的是北部和中部地区，相对落后的则是西部。毕竟电子商务跟经济状况的关系还是很大的，经济状况决定商品的购买能力^[12]。

从 2005 年开始，我国的电子商务市场交易额发展很良好，在 07 年的时候突破了 1.7 万亿元的规模，我国电子商务投资规模在未来的三年内仍然会持续增长和爆发。“十一五”时期，国家政府推动电子商务已经形成总体布局的标志是国家对电子商务发展规划的颁布；电子商务对企业的推动作用日益明显，如在应用成效上，对经济、社会发展^[13]。

在目前的电子商务市场中，B2B 的占有比较高，我国在 07 年的 B2B 交易额为 1.25 万亿，占电子商务总额的 70% 以上。我国的电子商务网站主要集中在几个大企业中，最多的是综合类网站的阿里巴巴，占全国的近 7 成的份额，其次就是 B2C 的京东商城和苏宁易购。2007 年我国电子商务 B2C 交易额约为 52.2 亿元，比 06 年增长了 33.5%。B2C 是一个主要形式，特别是传统企业开展电子商务，网络购物环境随着时间也慢慢变好，大中型企业电子商务化程度也逐渐提升，随着这些情况的好转，B2C 的发展将会得到十分充分，而且未来的网上购物主流也必然是 B2C。我国在 2007 年 C2C 的总收入约为 41040000000 元，这比上一年提升了 90%，我国目前的网上购物主流形式也是 C2C^[14]。

“十二五”时期，电子商务交易、技术和信息等类型的服务企业不断涌现出，我国电子商务行业的发展速度也十分快速，产业规模扩大的速度也十分迅速。中国电子商务市场的收入在 2010 年达到了 45000 亿元，这比 2009 年增长了 22%。在 2011 年，这一纪录再次被刷新，达到了 58800 亿元，而这些交易额中有约 60% 为中小企业的电子商务交易额。

2013 年 5 月 28 日国家发展改革委表示要支持电子商务的发展，出台了系列政策措施的部门有 13 个，主要从几个方面出发，如移动支付、可信交易、物流配送、商贸流通和网络电子发票等^[15]。

1.4 课题研究内容

本文的主要重点是，查看国内外的电子商务系统，通过了解他们的功能，进行分析和研究，收集他们的亮点和存在的问题，针对他们的问题，设计和实现功能完善的电子商城系统，能够面向大众的实用系统。对收集到的需求进行分析，了解大众真正所需要的功能，设计好大众真正需要的功能^[16]。主要实现用户登录、购物车、订单管理、库存管理等比较核心的功能，注重时效性、产品易用性的用户体验。

1.5 本文组织结构

本文主要针对电子商城的基本功能进行需求分析、系统架构和设计、系统实现和展示，在实现系统功能的前提下，注重系统的易用性和用户体验。论文的安排如下：

第一章：绪论

介绍电子商城系统研究的背景，提出了课题研究的目的地及意义，分析讨论电子商城在国内外发展的现状，最后介绍本文的研究内容及其组织结构。

第二章：系统开发方案分析及关键技术

分析对比系统开发的方案，介绍系统开发相关的关键技术。

第三章：系统的需求分析

介绍电子商城所需要实现的功能和非功能的需求，对它们用图表的形式进行系统分析描述。

第四章：系统的架构与设计

设计系统的整体架构，根据前一章的需求分析，介绍系统的概要设计、数据库设计和系统的功能设计。

第五章：系统的实现与测试

介绍实现环境，对功能进行实现，实现的系统代码，然后通过示例的形式功能进行展示，最后介绍功能测试和性能测试的情况。

第六章：总结和展望

给出本文的结论，并对本电子商城系统的后续功能的开发设计进行展望。

2 系统开发方案分析及相关技术

通过对比分析 C/S 架构和 B/S 架构的优缺点，并结合本系统的实际需要，最终开发方案采用 B/S 架构，开发工具为 Visual Studio 2010。本系统所采用的技术有 .NET、ASP.NET、ADO.NET、XML、SQL Server 和 Cookie 等。现将详细介绍系统开发的方案和相关技术。

2.1 系统开发方案分析

现在软件开发的整体架构主要分为 C/S 架构与 B/S 架构，选择哪种架构不仅对于软件开发很重要，也对实际应用也很重要^[17]。

C/S 架构是两层架构中比较典型的一种，C 表示 Client，而 S 表示 Server，其包含的两层是客户端和服务端，在用户的设备上运行的程序为两层中的客户端，服务器端一般分为两种，数据库类型和 Socket 类型，前者是通过数据库连接访问，后者是通过 Socket 来实现访问。

C/S 架构也称为是胖客户端架构。因为客户端需要承担系统的大部分工作，包括大部分的业务逻辑和界面展示。这种架构中，大部分的压力都在客户端，因为需要处理逻辑和事务，为了满足系统的需要通常用数据库的交互实现持久化数据^[18]，如 SQL、存储过程等。C/S 结构图如图 2-1 所示。

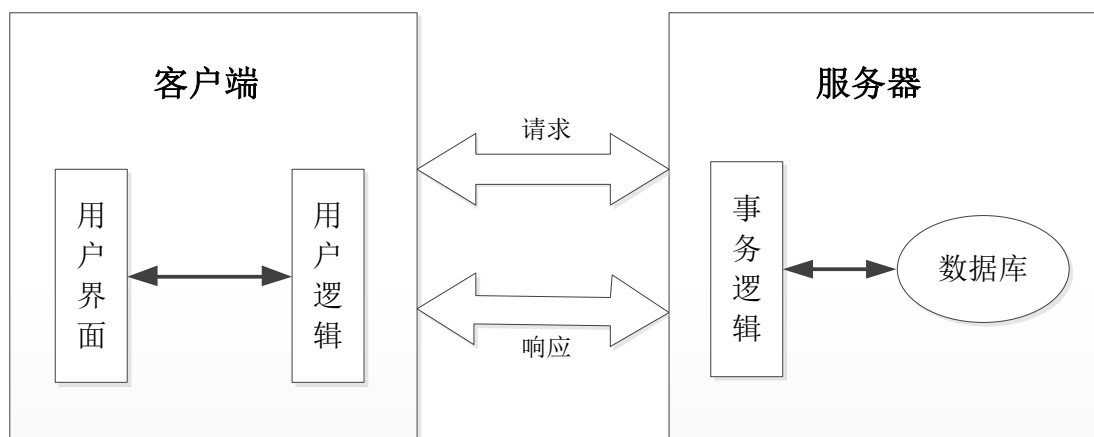


图 2-1 C/S 结构图

B/S 架构跟 C/S 架构不一样，它是三层的，B 表示 Browser，即用户设备上的浏览器，S 表示 Server，即服务器，浏览器负责少量的事务逻辑，但服务器需要负责绝大部分的事务逻辑，它的三层架构包括 Browser，WebApp 和 DB。B/S 架构的系统跟 C/S 最大的区别在于，C/S 结构需要安装客户端程序，而 B/S 则不需要，它通过 Web 浏览器即可实现功能^[19]。

B/S 架构中，用户设备上的浏览器负责显示逻辑的处理，WebApp 服务器则负责事务逻辑的处理，这样就减小了客户端，也使客户端的压力减轻了好多，因为这样的原因，B/S 架构也被称为瘦客户端架构。B/S 结构图如图 2-2 所示。

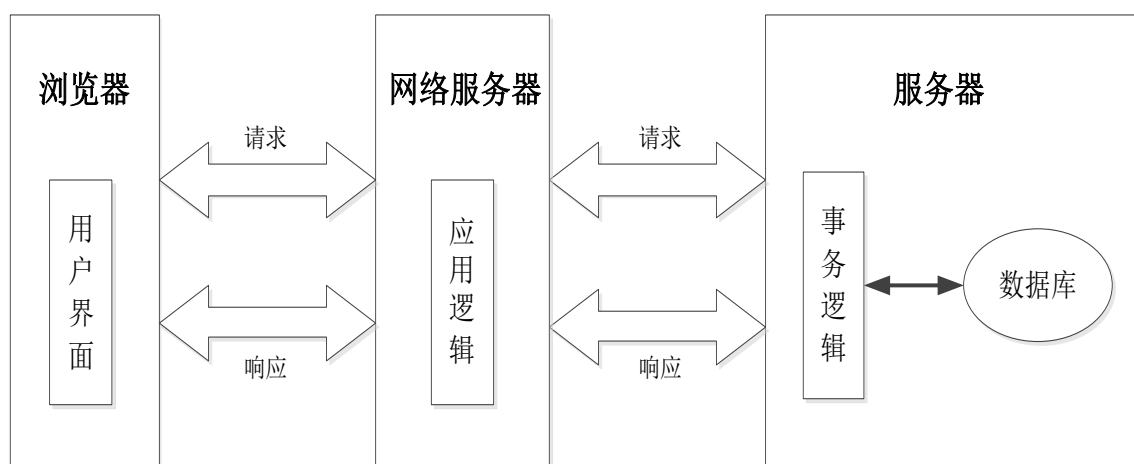


图 2-2 B/S 结构图

下面主要从几个方面来对比分析一下 C/S 和 B/S 的优缺点。

1) 硬件环境：C/S 架构的网络一般是专用的，网络环境也是小范围的，要想实现局域网通信，必须提供连接和数据交换服务，而这一般都是专门的服务器实现。B/S 则不需要专用网络环境，一般的广域网即可，电话上网，租用服务器都行，只需自己管理信息，它的适用范围比 C/S 更广，用户只需操作系统和浏览器即可完成功能^[20]。

2) 安全要求：C/S 的用户一般都是比较固定的人群，控制信息的能力也比较高，信息安全也能很大的保证，高度机密的信息系统就一般比较适合 C/S 架构。由于 B/S 是通过广域网实现的，面向的用户群是未知的，所以对安全控制就比较弱，适合用来发布可公开的信息。

3) 程序架构: C/S 架构的流程比较注重, 可以校验多层权限, 可以较少考虑系统的运行速度^[21]。B/S 架构则需要更多的考虑系统安全和访问速度的问题, 需要做很多的优化, 这些方面的要求都比 C/S 架构的高^[22]。B/S 架构是程序架构中的一种发展趋势。从微软的 .Net 系列全面支持网络的构件搭建的系统。

4) 软件重用: C/S 架构的整体性考虑是不可避免的, 这必然导致它的构件的重用性就要差一些。B/S 的构件需要完成的功能都是相对比较独立, 它的重用性就比较好。

5) 系统维护: C/S 架构是一个整体, 需要考虑它的整体性, 这导致处理系统问题和升级都很复杂, 修改一部分, 其他部分都可能需要修改, 有时候还需要对整个系统进行重新设计^[23]。

6) 处理问题: C/S 架构对系统安全方面的要求也比 B/S 高, 客户端运行与操作系统都有关, 必须在特定的系统下。

7) 用户接口: C/S 架构多是建立在特定的平台上, 各个平台之间也不相同, 从事这方面的程序员需要比较高的能力。建立在浏览器上的 B/S 架构, 有许多的表现方式, 都比较丰富和生动, 并且它们当中的普遍难度较低, 在开发成本上也不高^[24]。

对于本系统, 考虑到 C/S 架构需要设计客户端程序, 而且对于不同的设备需要不同的客户端程序, 如 PC 客户端、android 客户端、winphone 客户端、iOS 客户端等, 还需要设计服务端的程序, 工程比较庞大, 所以在这里就采用了 B/S 架构, 只需要设计服务端的程序^[25]。

2.2 系统开发相关技术

2.2.1 .NET

.NET, 英文全称 Microsoft .NET, 是微软公司的 XML Web services 平台, 无论选择的操作系统、设备或编程语言, 它都允许应用程序的通讯和共享数据通过 Internet 进行。 .NET 平台将 XML Web services 等这些服务集成在一起, 使它们之间的连接无缝, 方便用户, 这也是它最吸引人的体验^[26]。

.NET 开发平台主要由一组用于应用程序的组件构成, 如建立 Web 服务和

Windows 桌面应用程序组件，它包括三个部分，.NET 框架、.NET 开发者工具和 ASP.NET^[27]。

2.2.2 .NET Framework

.NET Framework 是一个运行执行环境，主要管理面向 .NET Framework 的应用程序的运行。它包括提供内存管理的类库、其他系统服务的类库，使程序员可以利用这些类库的健壮和可靠代码在应用程序开发的所有主要领域^[28]。

.NET Framework 作为一个托管执行环境，提供各种服务给其运行的应用程序，它的组件主要有两个：CLR（公共语言运行时）和 .NET 框架类库，前者处理运行的应用程序的执行引擎，后者提供可重用代码库，开发人员可以调用自己应用程序中已测试的代码库^[29]。用于运行应用程序，.NET Framework 提供的服务包括：

1) 内存管理。

许多程序语言没有自动管理内存的机制，需要程序员自己手动分配和释放内存，并且手动处理对象生存期。.NET Framework 应用程序则添加了自动的管理机制，CLR 为应用程序提供这些服务。

2) 常规类型系统。

传统编程语言中的基本类型都是编译器定义的，当两个语言需要交互时，这使它们的互操作性复杂化。.NET Framework 与传统的编程语言不一样，.NET Framework 类型系统取代了编译器，它定义了基本类型，并且 .NET Framework 的所有语言都共有这些基本类型^[30]。

3) 一个全面的类库。

开发人员可以直接使用 .NET Framework 类库中类型和其成员易于访问的库，这样的方法可以避免使用大量代码来实现常见或低级的编程操作。

4) 开发框架和技术。

.NET Framework 为应用程序开发提供了一些特定区域的库，如 ASP.NET，用于 Web 应用程序；ADO.NET，用于数据访问；Windows Communication Foundation，用于面向服务的应用程序。

5) 语言互操作性。

名为 CIL (公共中间语言) 的中间代码由面向 .NET Framework 的语言编译器发出, 反之, 通过 CLR 在运行时进行编译。此功能可以解决跨语言访问的问题, 一种语言可访问由另一种语言编写的例程, 这样开发人员可以将精力集中在其应用程序上, 也就是使用其首选语言创建的程序^[31]。

6) 版本兼容性。

使用特定 .NET Framework 版本开发的应用程序可以运行, 当 .NET Framework 版本升级时, 应用程序无需修改也可以正常运行, 当然少数例外的程序还是需要做一定的修改工作。

7) 并行执行。

同一台电脑上允许存在 CLR (公共语言运行时) 的多个版本, .NET Framework 可以帮助解决版本上的冲突问题。这表示应用程序也可以同时存在多个版本, 并且各个版本的应用程序都可以在与它的对应的 .NET Framework 版本上运行。

8) 多定向。

开发人员可以通过定向 .NET Framework 可移植类库创建在多个程序集在 .NET Framework 平台上工作, 如 Windows 7、Windows 8、Xbox 360 和 Windows Phone 平台。

ASP.NET 提供可伸缩的服务器端环境给托管代码, ASP.NET 启用 ASP.NET 应用程序和 XML Web Services 通过使用 CLR。

IE 是一个非托管应用程序的示例, 它是承载 CLR (以 MIME 类型扩展的形式) 的, 能够在 HTML 文档中嵌入 Windows 窗体控件或托管组件, 通过使用 Internet Explorer 承载 CLR。可以使托管移动代码成为可能, 通过使用这种方式承载 CLR, 这种方式也有一定的缺陷, 它需要进行的重大改进只有托管代码才能提供。

公共语言运行时、类库、运行库、应用程序、操作系统之间的关系如图 2-3 显示, 图 2-3 还显示在更大的结构内托管代码运行的方式。

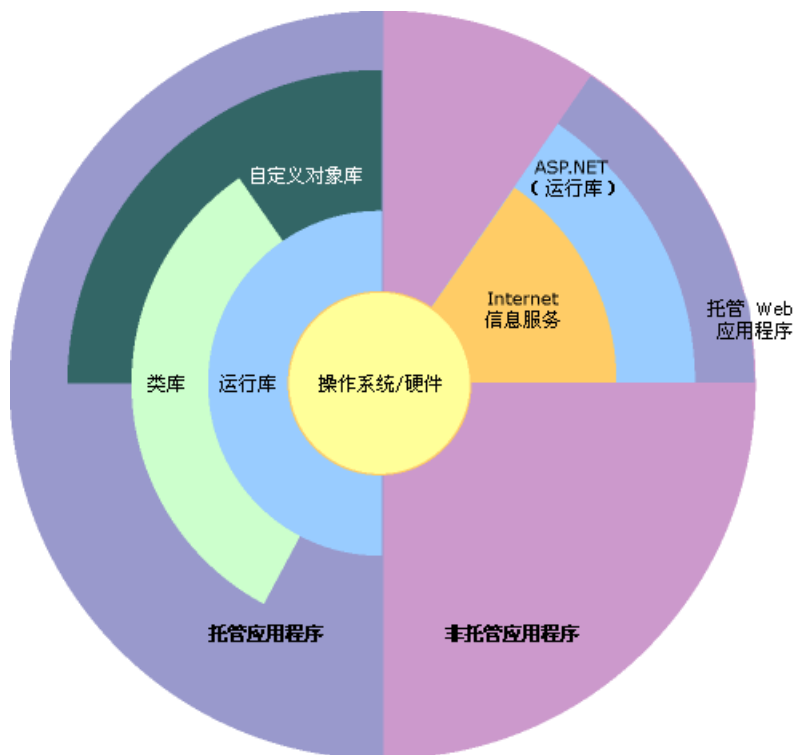


图 2-3 关系图

2.2.3 ASP.NET

ASP.NET 是 Microsoft 的一项技术，是一种服务器端脚本技术，它可以在 Web 服务器上动态创建它们当 HTTP 请求文档时。它指动态服务器页面（Active Server Pages），运行于 Internet 信息服务管理器（IIS，微软开发的 Web 服务器，全称 Internet Information Server）之中的程序^[32]。

ASP.NET 是一个开发框架，使用 JavaScript、CSS、HTML 和服务器脚本创建网页和网站。ASP.NET 支持 Web 页面（Web Pages）、模型-视图-控制器（MVC，全称 Model View Controller）、Web 窗体（Web Forms）三种不同的开发模式，其对比如表 2-1 所示。

表 2-1 三种开发模式对比

Web Pages [↵] 单页面模式 [↵]	MVC [↵] 模型-视图-控制器 [↵]	Web Forms [↵] 事件驱动模式 [↵]
最简单的 ASP.NET 模式 [↵] ，与 PHP 和经典 ASP 相似，内置了数据库、视频、图形、社交媒体等模板和帮助器 [↵]	MVC 将 Web 应用程序分成 3 个不同的组成部分： [↵] 模型负责数据 [↵] 视图负责显示 [↵] 控制器负责输入 [↵]	传统的 ASP.NET 事件驱动开发模式： [↵] 带有服务器控件、服务器事件和服务器代码的网页。 [↵]

ASP 技术是 ASP.NET 的前身，是在 Internet 信息服务管理器 2.0(Windows NT 3.51)上首次推出，当时一起推出的还有 ADO 1.0，在 Internet 信息服务管理器 3.0 上得到了更大的发展，也就是 Windows NT 4.0 上，IIS 成为热门开发工具用于服务器端应用程序的开发^[33]，Visual Inter Dev 就是 Microsoft 特别为它量身打造的开发工具，在 1994 年到 2000 年，在 Microsoft 的 Windows NT 4.0 平台上，ASP 技术已经成为关键技术之一，如雨后春笋般出现在网络上的数以万计的网站都是采用的 ASP 技术。ASP 也有它的缺点，而且这些缺点也都慢慢的被发现，它的程序开发方法是面向过程的，这大大提高了维护的难度，大型的 ASP 应用程序就更为麻烦。性能没有办法得到完全的发挥，虽然开发人员可以用 COM 组件，但是有一些特殊的功能要开发时，内置的支持无法使用，只能通过第三方控件商的控件^[34]。

2.2.4 ADO.NET

ADO.NET 技术的起源是 ADO，是一个用于在以往的微软技术中访问数据的 COM 组件库。ADO.NET 是一个数据访问接口可以优先使用在 .NET 编程环境中。

ADO.NET 可以让程序员存取资料的来源，通过一致的方式（例如 SQL Server 与 XML），以及公开资料的来源通过 OLE DB 和 ODBC。对于开发资料共用的应用程序，可以连接到这些资料的来源通过使用 ADO.NET，并且获取、处理及更新这些资料库所含的资料。

ADO.NET 是对微软 ADO 一个跨时代的改进，它的大的改进就是，提供了可伸缩的数据访问和平台互用性。任何应用程序都可以进行数据处理，只要能够读取 XML 格式，因为传送的数据都是 XML 格式的。事实上，不是只有 ADO.NET 组件才可以接受数据，基于微软的 VS 的解决方案，甚至是在其它任何平台上运行的任何应用程序都可以^[35]

2.2.5 XML

XML 即可扩展标记语言，属于标准通用标记语言的子集，是一种标记语言，主要用于标记具有结构性的电子文件^[36]。

它的作用不仅可以标记数据，而且可以定义数据类型，是一种源语言，用户定

义自己的标记语言。万维网传输就非常适合这样的语言，描述和交换独立于供应商或应用程序的结构化数据就是通过它提供统一的方法。XML 是因特网环境中跨平台的技术，它依赖于内容，也是有效工具用于当今处理分布式结构信息。在 1998 年，XML1.0 规范由 W3C 发布，通过它的优点来简化因特网的文档信息传输。

2.2.6 SQL Server

SQL 即结构化查询语言（英文全称 Structured Query Language）。SQL 语言主要用于应用程序和数据库之间建立联系，并且进行数据交换。可以利用 SQL 执行各种数据库的操作，例如写入，提取，删除，更新等^[37]。虽然 SQL 语句被很多数据库进行了扩展和再开发，但是很多标准的 SQL 命令还是可以完成几乎所有的数据库操作，如 Select, Insert, Update, Delete, Create, 以及 Drop^[38]。

SQL Server 是最初由微软、Sybase 和阿什顿 - 泰特三家公司共同开发的一个关系数据库管理系统。在 1988 年，三家推出了第一个 OS/2 版本。对于 SQL Server 的开发，在 Windows NT 推出后微软与 Sybase 就分道扬镳了，在 Windows NT 系统上，SQL Server 就被微软移植添加到其中，并且微软专注于开发推广 Windows NT 系统版本的 SQL Server，而 Sybase 专注另一个方向，它较专注于 UNIX 操作系统上 SQL Server 的应用^[39]。

2.2.7 Cookie

Cookie，在有些时候也被用做 Cookies，其实是一种储存在用户本地终端上的数据，主要是某些网站用于辨别用户的身份还有进行 session 跟踪，这些数据通常是经过加密的。

Cookies 是一种技术，网站服务器可以通过这种技术把少量数据储写入客户端的硬盘或内存中，也可以通过同样的方式从客户端的硬盘读取数据。

当用户浏览某网站时，Web 服务器会在用户的硬盘上存放一个非常小的文本文件，其实就是某类别的数据库，它可以记录用户的一些信息，如用户 ID、密码、浏览过的网页、停留的时间等^[40]。

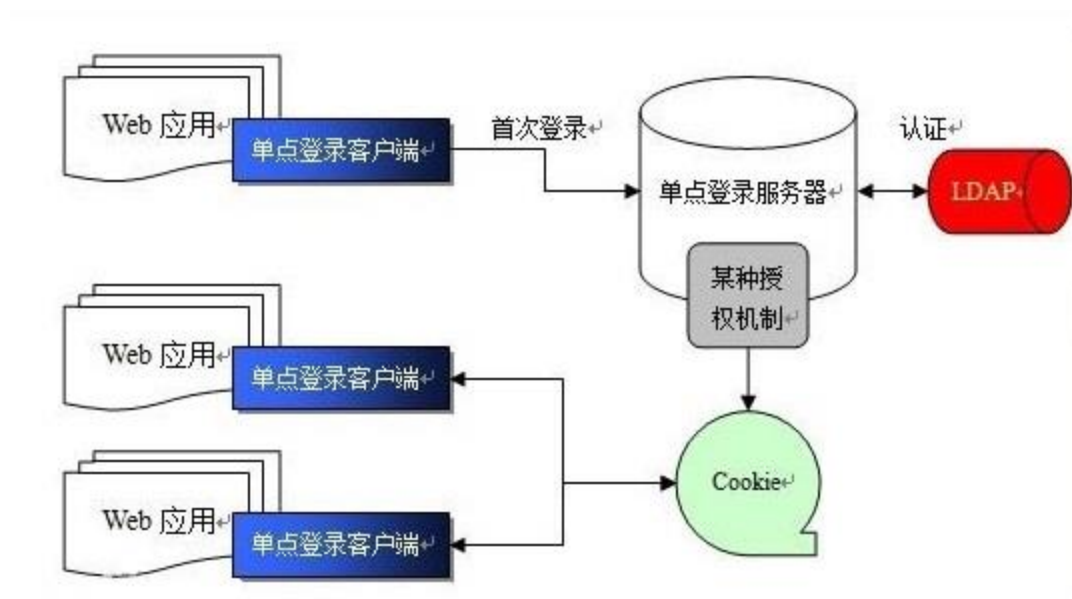


图 2-4 使用 cookie 实现单点登录

Cookies 的工作原理是，当用户第一次访问网站，由服务器端将 Cookies 写入到客户端的系统中，以后用户每次访问这个网页时，Cookies 信息都要先被客户端程序发送给服务器端，并对 Cookies 进行判断，然后返回给客户端服务器产生的 HTML 代码^[41]。使用 cookie 实现单点登录如图 2-4 所示。

在当前的互联网中，主要是通过跟踪 Cookies 信息这一方式来实现定向广告，用户每次访问网站时，网站应用程序读取的用户信息都包含在终端设备的 Cookies 中^[42]。用户操作行为相关的 Cookies 信息被一些网站获取到之后，不仅仅作为自己网站使用，更是供给其他企业使用，如一些与该网站有利益关系的其他企业或团体，这些信息都涉及到了用户的隐私信息。这就导致了问题，用户的隐私信息，不只是当前的网站能够获取，该网站将信息共享，其他企业也可以获取用户的信息并从中获取利益^[43]。

跟踪 Cookie 是一种更特殊的 Cookie，它与一般的 Cookie 不同，但它的基础还是 Cookie 文件，在两个或两个以上不相关的网站之间，跟踪 Cookie 实现了分布、共享和读取，跟踪 Cookie 的最终目的就是向客户端浏览器输出显示自定义的数据，如商品广告信息，通过网站脚本代码收集信息^[44]。当然，并非所有 Cookie 都是跟踪 Cookie。虽然跟踪 Cookie 不具有较强的危害性，不像蠕虫、恶意软件或病毒，

但是它包含了用户的许多隐私信息，这就涉及到了用户的隐私泄露问题^[45]。例如，若用户访问一个网站，该网站是由第三方供应商托管在线广告的，供应商便会放置一个 Cookie 在用户的终端设备上。如果该供应商还给另一个网站提供了广告服务，则该供应商便了解到这两个网站被用户访问过，便可以分析跟踪到的用户操作行为信息，将用户可能感兴趣的广告提取出来，并向用户定向投放广告^[46]。

2.3 本章小结

本章介绍的内容主要分成两个部分，系统开发方案的分析 and 工具，系统开发相关的技术。系统开发方案分析主要介绍 C/S 和 B/S 架构的基本原理，并从多个方面对这两种开发方案进行优劣分析，以及最后的开发方案的取舍。系统开发相关的技术主要介绍系统开发所需要用到的关键技术，包括 ASP.NET、.NET Framework、XML、SQL Server 和 cookie。下一章节主要介绍系统的需求分析，主要分为功能需求、功能划分和非功能需求等部分。

3 系统的需求分析

通过国内外现成相关系统的功能，结合实际生活的需要以及开发的难易程度，设计本系统的功能，系统分成两个子系统——前台系统和后台系统。现将详细介绍各个子系统所包括的模块及其所要实现的功能，并从软件的质量特性方面介绍本系统的非功能需求。

3.1 系统需求介绍

电子商城系统的服务对象分为两种，消费者和商家，它是消费者和商家之间纽带，核心功能是购买商品，为实现购买商品功能，需要有基础功能的支持，如消费者的注册登录，消费者个人信息的编辑修改，商家能够管理商品，而且能够管理用户，最后对消费者已经购买的商品信息能够处理，即订单的管理，只有这些基础功能实现稳定，才能够支持稳定的核心功能。

3.2 系统功能需求分析

电子商城系统主要分为两个大的模块——前台系统和后台系统，前台的用户主要是普通顾客，包括注册的用户和未注册的用户，该模块主要为用户提供的功能有：登录注册、浏览商品、购买商品，个人信息管理等，当然未注册的用户只能浏览商品，而不能直接购买，购买商品的前提需要先注册。后台的用户主要是系统的管理员，包括商品的卖家，主要的功能有商品管理、订单管理、用户管理和系统管理。

3.2.1 电子商城前台系统

电子商城的前台系统主要是实现商品的购买功能，针对的用户是网站的普通用户，注册和非注册用户都可以进行操作，该子系统包括用户子模块、商品浏览模块和商品购买模块这三个功能模块。

1) 用户子模块

用户子模块主要实现跟前台用户相关的功能，主要是网站的消费者，分为注

册用户和非注册用户，包括的子功能有用户登录、用户注册和用户信息修改。

先要介绍的是非注册的用户，主要是针对第一次访问该网站的用户且该用户还没有注册账号，对于该类型的用户，需要比较华丽的界面来吸引他们，将这些潜在用户发展成网站的忠实的用户群，可以通过注册页面完成账号的注册，也可以浏览网站首页的商品，或者点击查看商品的详细信息。非注册用户的用例图如图 3-1 所示。

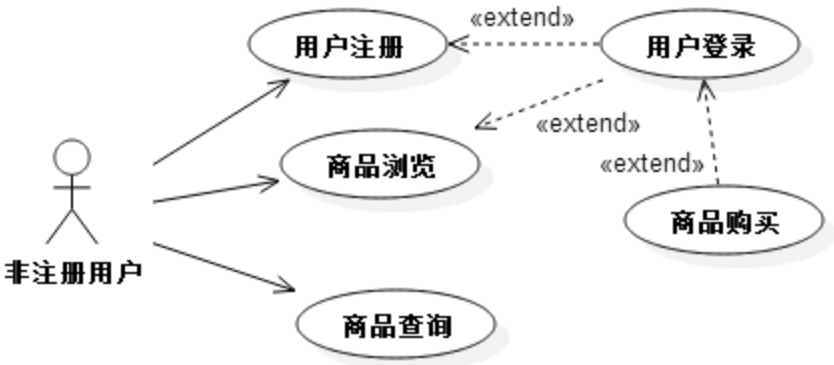


图 3-1 非注册用户用例图

网站系统比较稳定的用户群是已经注册的用户，需要提供比较实用而且简洁的功能。可以通过登录页面登录网站系统，在查看商品时购买商品，或者将商品添加到购物车中并且完成商品下单，也可以查看自己的用户信息，并且对这些用户信息进行编辑修改，如登录的账号密码、收货信息。注册用户的用例图如图 3-2 所示。

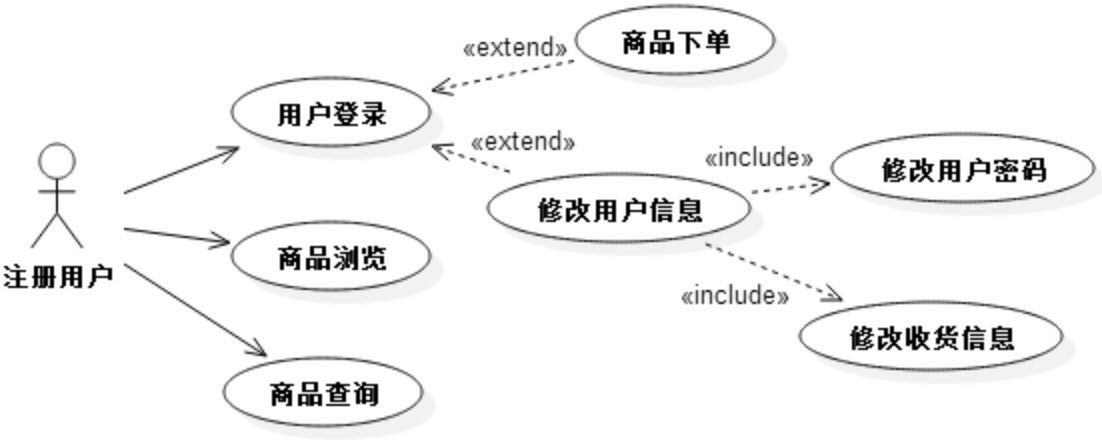


图 3-2 注册用户用例图

2) 商品浏览模块

商品浏览模块主要实现网站用户对商品的查看功能，用户是否已经注册都不影响浏览查看商品，未注册用户也可以商品浏览功能是因为第一次浏览该网站的用户还没有账号，可以通过商品信息或优惠活动吸引他们注册，将潜在这些潜在的用户发展成为稳定的用户。用户可以浏览网站的首页商品，一般为新推出的商品、优惠活动的商品或者热销的商品，用户可以预览到这部分商品的部分信息，如商品的名称、商品的图片、商品的优惠活动或热销商品的销售量等。除了浏览首页商品，用户还可以通过商品类别查看同类别的商品，方便用户进行分析对比，以便购买更好的商品，也可以通过搜索栏搜索查询商品，需要输入商品名称的全称或者部分名称，列出包含该名称的商品。用户除了浏览首页和查询到的商品，可以点开某个商品查看商品的详细信息。以上主要对于用户正常浏览商品，而用户在长时间浏览商品时有时需要查看之前浏览的记录，就需要添加商品浏览历史记录的功能，已登录的用户可以查看商品浏览记录，对于未登录的用户同样需要查看历史记录的功能，需要将用户的浏览信息保存在本地。浏览商品的用例图如图 3-3 所示。

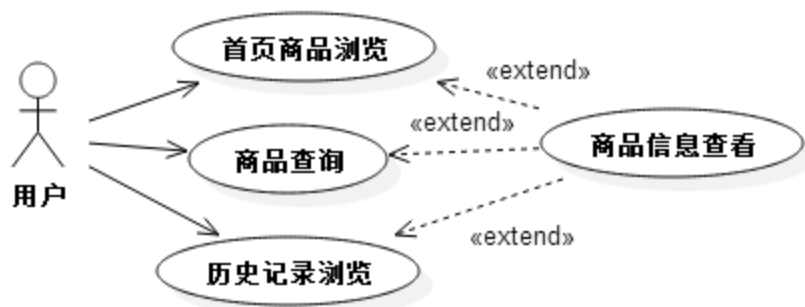


图 3-3 浏览商品用例图

3) 商品购买模块

商品购买模块主要是实现用户购买商品的功能，这部分的功能只是针对已经注册且登录成功的用户，未登录的用户不能购买商品，因为购买商品需要提交用户的信息，如姓名电话、收货地址等，未登录的用户无法提供用户信息。用户可以在查看用户商品信息界面直接点击购买商品，也可以添加到购物车中，对于选中的商品，用户可以在购买商品中形成订单，也可以对购物车中的商品下单，商品下单之

后就可以进行支付了，完成了支付就等着卖家将商品通过快递的方式寄送给用户了。购买商品的用例图如图 3-4 所示。

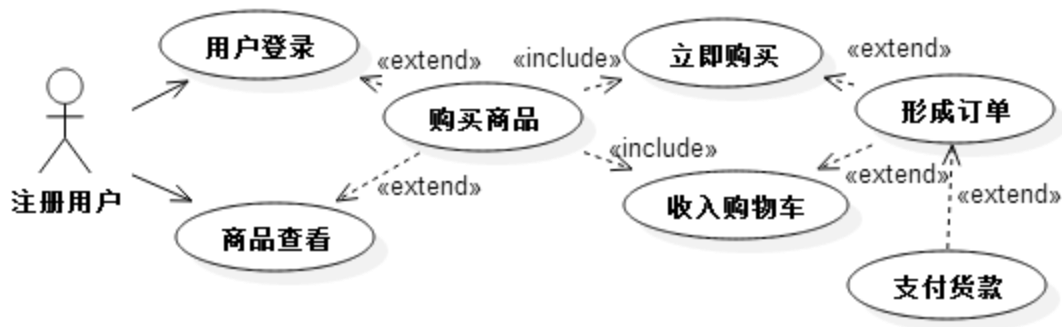


图 3-4 购买商品用例图

3.2.2 电子商城后台系统

电子商城的后台系统主要实现后台信息的管理功能，针对的用户是网站的管理员或者商品的卖家，需要利用管理员账号才可以登录操作，未注册用户和普通账号的用户没有权限操作，而且后台系统的登录地址也与前台的不一样。该子系统包括商品管理模块、订单管理模块和用户信息管理模块三个子模块。

1) 商品管理模块

商品管理模块主要实现对商品的管理功能，这个模块是针对网站的管理员或者商品的卖家，普通用户没有权限进行操作。管理员登录后台系统后对商品信息进行管理，包括商品信息的增删改查，对应添加商品、删除商品、编辑商品和查询商品。管理员在添加新商品时需要填写商品的信息，只有这些信息的输入符合要求才可算添加商品成功。管理员也可以直接删除一个商品所有信息，或者编辑修改商品的信息，修改和添加的商品信息不包括商品的销售量。在删除或修改商品时，会遇到不容易找到目标商品的情况，需要用到商品查找功能，便于管理员快速找到目标商品并管理商品。管理员还可以管理网站的首页商品，如添加、删除、修改。首页只有一部分可以修改，另一部分是热销商品，根据所有商品的销售量进行排序筛选出来的，管理员无法直接进行修改。管理商品的用例图如图 3-5 所示。

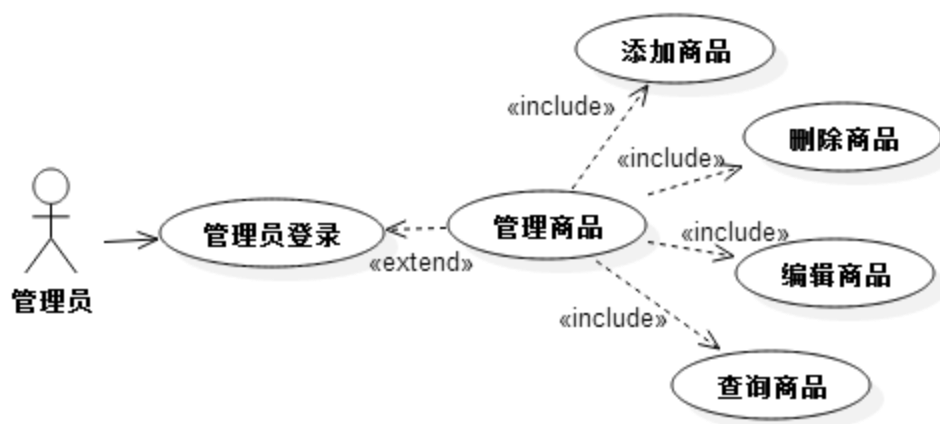


图 3-5 管理商品用例图

2) 订单管理模块

订单管理模块主要实现商品订单管理功能，这个模块是管理员对商品订单的处理操作。普通用户对商品下单之后系统会形成一个订单，管理员可以对这个订单进行操作，包括订单的查看和处理。订单查看可以查看订单的详细信息，如订单的生成时间、商品名称、商品数量，订单状态等信息，管理员可以处理订单，对于已经付款的修改状态并且将商品寄送出去，在商品的物流状态发生变化时及时修改订单的状态，如买家收到货物后管理员可以将订单状态修改成已交易完成状态。管理员可以通过该模块功能对商品订单进行跟踪处理。管理订单的用例图如图 3-6 所示。

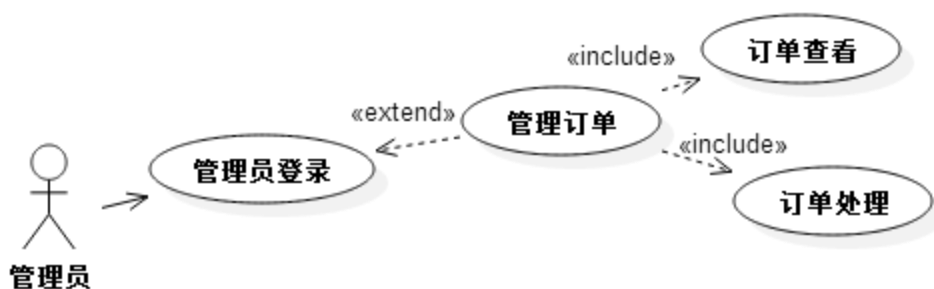


图 3-6 管理订单用例图

3) 用户管理模块

用户管理模块主要实现管理用户的功能，用户包括系统的前台普通用户和后台管理员用户。该模块主要分为两个部分，对普通用户的管理模块和对管理员用户的管理模块，而这两部分实现实现的功能都是一样，对用户信息的添加、删除和修改，

考虑到用户信息的安全问题，这部分的功能只有一个管理员用户才可以操作，而其他的管理员无法对用户的信息进行操作或查看。用户管理模块的用例图如图 3-7 所示。



图 3-7 管理用户用例图

3.3 系统非功能需求分析

3.3.1 软件质量特性

1) 可靠性

容错性：用户在系统的输入区域输入非法的数据或进行的操作不合理，系统不会因为这崩溃，而且数据也不会缺失。系统自动检查用户的输入内容，防止不合理的输入内容造成系统运行不正常。

可恢复性：系统因为故障而崩溃或断电等异常情况，系统在重新启动时应该能够恢复数据，并且能正常运行。

2) 易用性

易懂性：用户能够从系统的界面容易理解该系统所包含的功能的功能及功能的适用性。

易操作性：用户交互界面要良好，用户操作容易，对于用户的不当操作都需要给予相关的提示信息。对于用户非法输入或操作，系统要阻止。

3) 正确性

正确性：用户正确操作后能够正确或相符的结果或效果。

4) 可重用性

可重用性：软件能作为软件包而被再利用。

5) 可维护性

易分析性：软件出现诊断缺陷或失效，用户方便获得原因。

易改变性：维护人员方便对软件进行修改或排除错误。

稳定性：维护人员对软件做正确修改时软件能够稳定运行。

6) 可测试性

可测试性：维护人员对软件做出正确修改时方便测试的进行。

7) 可用性

可用性：用户能够根据软件的提示进行正确操作。

8) 适应性

适应性：软件无需采用有别于为该系统准备的活动或手段就可能适应不同的规定环境。

9) 安全性

安全性：用户使用自己账号登录无权访问其他用户的账号数据。

3.4 本章小结

本章节通过对电子商城系统的需求分析进行了比较详细的介绍包括系统的功能和非功能需求分析。主要介绍电子商城系统的核心功能以及为核心功能提供基础支持的其他附加功能。系统功能需求分析主要对电子商城系统的功能进行分析，对系统功能模块的划分，详细介绍每个模块所要实现的功能以及用例图。系统非功能需求主要从几个方面介绍电子商城的质量需求，如可靠性、可用性、安全性等。下一章节主要介绍系统的架构与设计，分为系统概要设计、数据库设计和模块功能设计。

4 系统的架构与设计

根据对功能的需求分析，本系统采用 B/S 架构，并分为表示层、业务逻辑层和数据访问层这三层。设计数据库的实体及其属性、并确定实体之间的关系。系统的前后台子系统所要包含的模块和每个模块的工作流程。现将介绍系统的架构、数据库和系统功能。

4.1 系统概要设计

为了整个系统开发考虑，电子商城系统采用 B/S（Browser/Server，即浏览器/服务器结构）架构。系统分为两分子系统——前台系统和后台系统，前台系统主要实现用户信息自管理，查看、购买商品等功能，后台系统则实现管理员管理商品、订单和用户等功能。系统开发用 ASP.NET 的三层体系结构，系统结构图如图 4-1 所示。

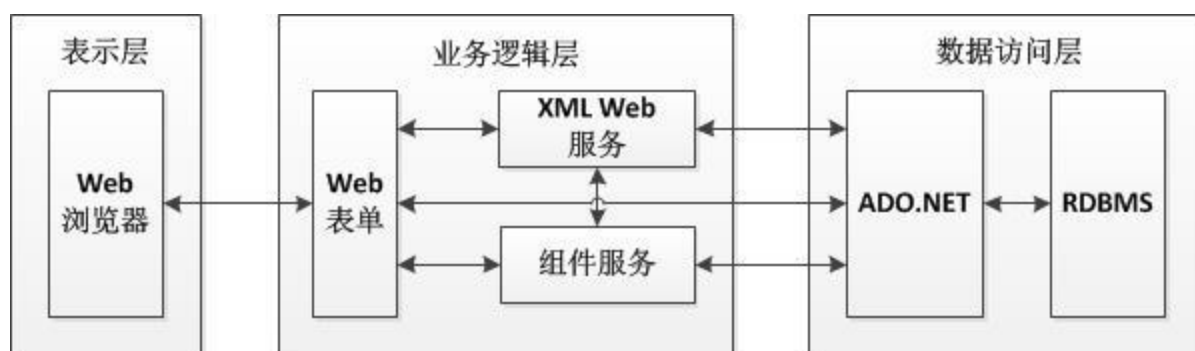


图 4-1 系统结构图

系统分为表示层、业务逻辑层和数据访问层这三层结构。

表示层（UI）：处理系统的输入输出，用户与该层直接交互，用户的输入信息直接传给表示层，将接收到的服务器端信息呈现给用户。需要给用户呈现出友好简洁的界面，并且对用户的输入信息做一定规则的合法性检测，对于非法输入需要以简单明确的方式提示用户。用户的直观感受都在表示层中体现。

业务逻辑层（BLL）：主要负责系统的业务处理，对表现层和数据访问层起着

承接的作用，浏览器的请求传给该层，该层再将请求传给数据访问层，数据访问层的处理结果也需要先返回给该层。该层主要分为三部分，网页表单、XML 网页服务和组件服务。

数据访问层（DAL）：该层只是与数据库进行交互，对数据库进行增、删、改、查和判断存在性等数据访问方法，数据访问层只是对数据的操作层，所以该层不是数据库。该层为业务逻辑层提供服务，通过 ADO.NET 操纵，如将数据操作结果存储、将数据检索结果返回给业务逻辑层等。

ASP.NET 三层架构技术具有很多的优点，如灵活性、简易性、安全性等。

4.2 系统数据库设计

4.2.1 数据库逻辑设计

分析系统的需求、系统的流程以及系统的功能结构，规划出系统中使用的数据库实体，如“商品类型”、“商品信息”、“订单”、“管理员”、“用户”、“购物车”实体。数据库 E-R 图如图 4-2 所示。

图 4-2 中各实体对象的属性及其对象之间的关系如下：

- 1) 电子商城包含多个商品类别，每个类别包含多种商品。
- 2) 商品包含多种属性，包括商品类别、商品名称、商品价格等信息，而销售量和库存量属性则根据业务逻辑层的操作进行实时的更新。
- 3) 每个用户有一个购物车，而购物车可以包含任意数量的商品，甚至没有商品则为空购物车。
- 4) 用户购买商品之后会生成商品订单，一个订单可以包含多件商品，也包含一个用户信息和一个管理员
- 5) 非注册用户通过注册成为注册用户，拥有用户名和密码等信息。
- 6) 管理员也可以编辑修改商品信息和用户信息。

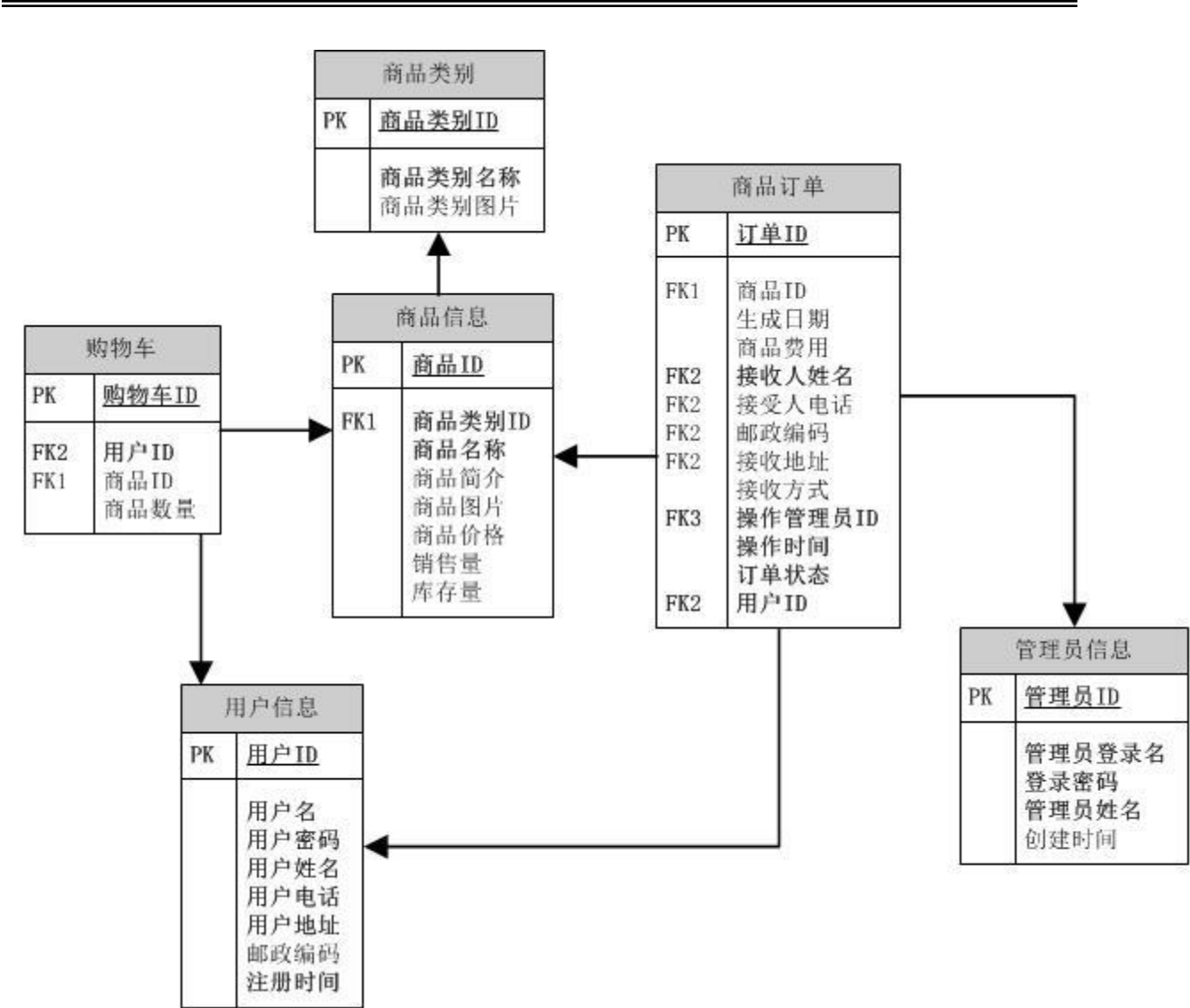


图 4-2 数据库 E-R 图

表 4-1 数据库表功能表

表名称↕	表 ID↕	系统功能↕
商品类别表↕	Tb_Class↕	记录商品的类别↕
商品信息表↕	Tb_GoodsInfo↕	记录商品的详细信息↕
商品订单表↕	Tb_OrderInfo↕	记录商品订单的信息↕
管理员信息表↕	Tb_Admin↕	记录管理员的信息↕
用户信息表↕	Tb_Member↕	记录用户的信息↕
购物车表↕	Tb_Trolley↕	记录用户的购物车信息↕

4.2.2 数据库表设计

根据数据库的 E-R 图进行数据库表的设计，对表的功能进行了定义。为后续的系统开发实现提供数据库支持和文档支持。表 4-1 列出了数据库定义的表名称、表 ID 和对应的系统功能。

表 4-2 商品类别表 Tb_Class 用于保存商品类别的基本信息。主键为 ClassID，且商品类别名称不能为空，是确保每个类别都有一个名称。

表 4-2 商品类别表

列名↵	数据类型↵	说明↵	备注↵
ClassID↵	int↵	商品类别 ID↵	Primary Key↵
ClassName↵	Varchar(50)↵	商品类别名称↵	Not Null↵
CategoryUrl↵	Varchar(50)↵	商品类别图片↵	↵

表 4-3 商品信息表用于保存商品的基本信息。在商品信息表中，ClassID 是该商品类别的 ID，对应的是商品类别表的 ClassID。ClassID 为商品信息表的外键，是每种商品都属于一个种类。商品的名称、价格、销售量和库存量字段不能为空，这些属性是商品销售的基本要求，销售量默认值为 0，且会根据业务逻辑的处理而实时的更新。上架日期字段是在商品添加的时候由系统根据当时的时间自动添加。

表 4-3 商品信息表

列名↵	数据类型↵	默认值↵	说明↵	备注↵
GoodsID↵	Int↵	↵	商品 ID↵	Primary Key↵
ClassID↵	Int↵	↵	类别 ID↵	Foreign Key↵
GoodsName↵	Varchar(50)↵	↵	商品名称↵	Not Null↵
GoodsIntroduce↵	Ntext↵	↵	商品简介↵	↵
GoodsUrl↵	Varchar(200)↵	↵	商品图片↵	↵
MarketPrice↵	Float↵	↵	商品价格↵	Not Null↵
SalesVolume↵	Int↵	((0))↵	销售量↵	Not Null↵
Inventory↵	Int↵	↵	库存量↵	Not Null↵
LoadDate↵	Datetime↵	(getdate())↵	上架日期↵	↵

表 4-4 订单信息表 Tb_OrderInfo 存储订单信息。在订单信息表中，OrderDate 字段在订单生成时由系统根据当时时间自动添加，GoodsID 字段表示该订单包含的商品，OrderState 字段表示订单的状态，如已付款，已发货、交易完成等。ConfirmTime 字段表示每次订单状态修改时的时间，而 AdminID 字段则表示修改管理员的 ID。订单中关于买家的收货信息和运输方式都不能为空，确保能将商品送到用户的手中。

表 4-4 订单信息表

列名↕	数据类型↕	默认值↕	说明↕	备注↕
OrderID↕	Int↕	↕	订单 ID 号↕	Primary Key↕
OrderDate↕	Datetime↕	(getdate())↕	订单生成日期↕	↕
GoodsID↕	Int↕	↕	商品 ID↕	Foreign Key↕
GoodssFee↕	Float↕	↕	商品费用↕	↕
ShipType↕	Varchar(50)↕	↕	运输方式↕	Not Null↕
ReceiverName↕	Varchar(50)↕	↕	接收人姓名↕	Not Null↕
ReceiverPhone↕	Varchar(20)↕	↕	接收人电话↕	Not Null↕
ReceiverPostCode↕	Char(10)↕	↕	邮政编码↕	↕
ReceiverAddress↕	Varchar(200)↕	↕	接收人详细地址↕	Not Null↕
MemberID↕	Int↕	↕	用户 ID↕	Foreign Key↕
OrderState↕	Char↕	↕	订单状态↕	Not Null↕
AdminID↕	Int↕	↕	管理员 ID↕	Foreign Key↕
ConfirmTime↕	Datetime↕	(getdate())↕	操作时间↕	↕

表 4-5 管理员信息表 Tb_Admin 用于保存管理员的基本信息。管理员的登录信息字段都不能为空，如登录名、密码和姓名。LoadDate 字段则表示该管理员被创建的时间，由系统添加。

表 4-5 管理员信息表

列名	数据类型	默认值	说明	备注
AdminID	Int		管理员 ID	Primary Key
Adminname	Varchar(50)		管理员登录名	Not Null
Password	Varchar(50)		管理员密码	Not Null
Realname	Varchar(50)		管理员真实名	Not Null
Email	Varchar(50)		Email 地址	
LoadDate	Datetime	(getdate())	创建时间	

表 4-6 用户信息表存储已经注册的会员信息，如用户名、密码、真实姓名等。用户的收货信息不能为空。

表 4-6 用户信息表

列名	数据类型	默认值	说明	备注
MemberID	Int		用户 ID	Primary Key
UserName	Varchar(50)		用户登录名	Not Null
Password	Varchar(50)		用户登录密码	Not Null
RealName	Varchar(50)		用户真实姓名	Not Null
Sex	Bit		用户的性别	
Phonecode	Varchar(20)		电话号码	Not Null
Email	Varchar(50)		用户 Email 地址	
Address	Varchar(200)		用户详细地址	Not Null
PostCode	Char(10)		邮编	
LoadDate	Datetime	(Getdate())	创建时间	

表 4-7 购物车表 Tb_Trolley 主要用来存储每个用户的购物信息，包括商品 ID 和数量。

表 4-7 购物车表

列名	数据类型	说明	备注
TrolleyID	Int	购物车 ID	Primary Key
MemberID	Int	用户 ID	Foreign Key
GoodsID	Int	商品 ID	Foreign Key
GoodsCount	Int	商品数量	

4.3 系统功能设计

电子商城系统有两个服务群体，所以整个系统分为两个部分，前台系统面向消费者用户，实现跟消费者相关的功能，如用户注册登录、商品浏览和购买；后台系统面向商家或管理员，实现他们相关的功能，如商品的管理、订单的管理和用户的管理。前后台系统相互关联。这里涉及到用户的信息，如用户的登录名、密码、个人信息，所以后台系统在用户管理功能上需要设置权限，只有特定管理员才可以操作处理用户的信息，以保证用户的信息安全。系统功能模块图如 4-3 所示。

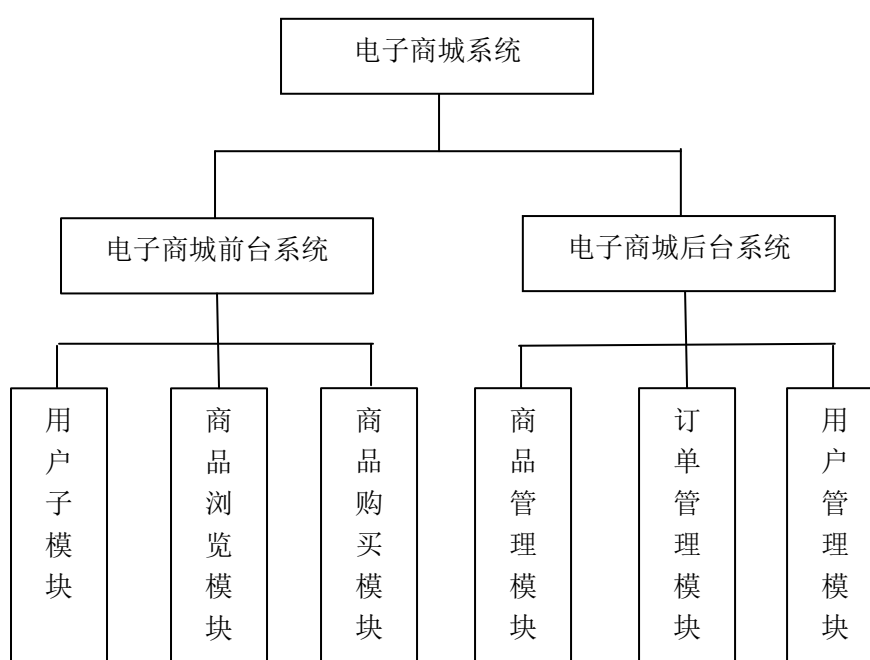


图 4-3 系统功能模块图

4.3.1 用户子模块的设计

用户子模块实现用户的注册、登录和修改信息的功能。用户第一次打开该系统需要注册账号，然后才可以登录系统，并且编辑修改用户的个人信息，若有操作未成功的情况，需要重新上一个步骤。用户模块的流程图如图 4-4 所示。

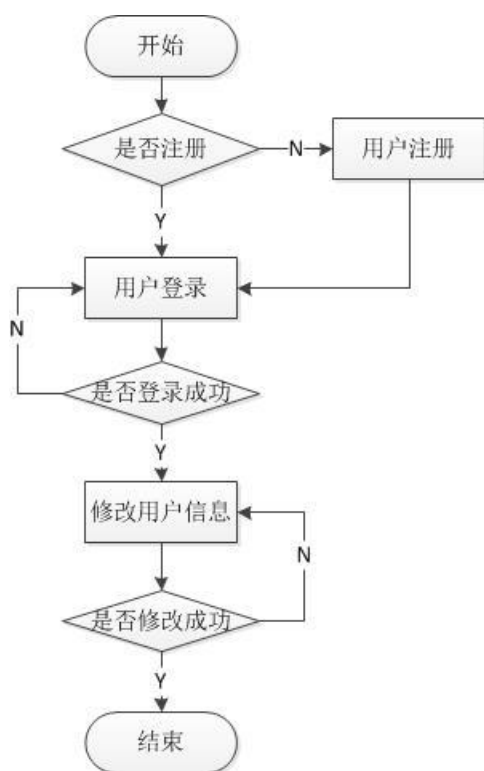


图 4-4 用户子模块流程图

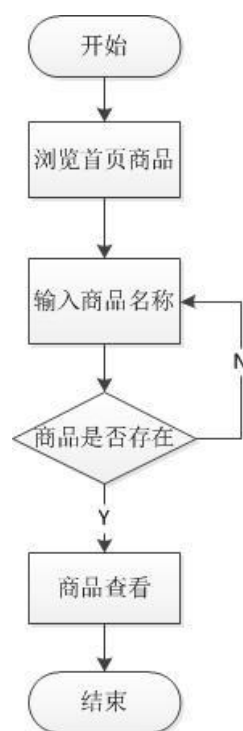


图 4-5 商品浏览模块流程图

4.3.2 商品浏览模块的设计

商品浏览模块实现使用户能够浏览查看商品，如预览商品的信息和查看商品的详细信息。用户在打开系统时系统显示的是首页，首页主要显示推荐的商品和热销的商品，推荐的商品由管理员手动添加，而热销商品则是由系统根据商品的销售量进行排序取销量靠前的商品显示，可以直接点击商品图标查看商品的详细信息；也可以在搜索栏中输入商品名称搜索商品并查看商品的详细信息。浏览商品的流程图如图 4-5 所示。

4.3.3 商品购买模块的设计

商品购买模块实现使用户能够购买商品，如将商品添加到购物车中，订单生成，支付货款。用户在查看商品时可以将商品添加到购物车中，也可以直接购买商品，但前提是用户已经登录成功，若用户未登录则需要先进行登录操作，对用户购买的商品并选择商品运输方式，系统会根据用户买的商品、数量、用户的信息生成订单，用户付款成功之后则购买流程完成，等待卖家发货。购买商品的流程图如图 4-6 所示。

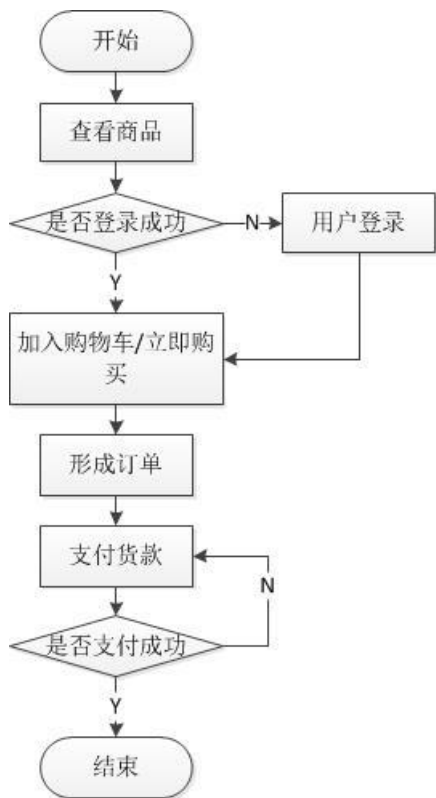


图 4-6 商品购买模块流程图

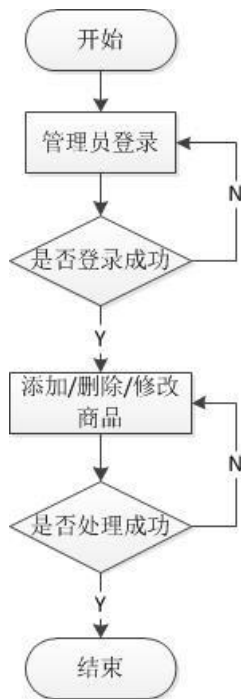


图 4-7 商品管理模块流程图

4.3.4 商品管理模块的设计

商品管理模块实现使管理员能够管理商品信息。管理员登录后台系统，登录成功之后才可以进行商品管理操作。添加商品需要输入商品的信息，系统检测商品的必须信息是否正确输入，返回是否添加成功并提示用户，对于删除和修改商品信息与添加类似。管理商品信息的流程图如图 4-7 所示。

4.3.5 订单管理模块的设计

订单管理模块实现管理订单，即订单的查看和处理。管理订单需要登录后台系统，点击查看订单并对订单进行处理，既根据实际的商品状态修改订单的状态，如对于已经发货的商品修改为已发货，并进行分类，以便将不同种类的订单进行分开处理。管理订单的流程图如图 4-8 所示。

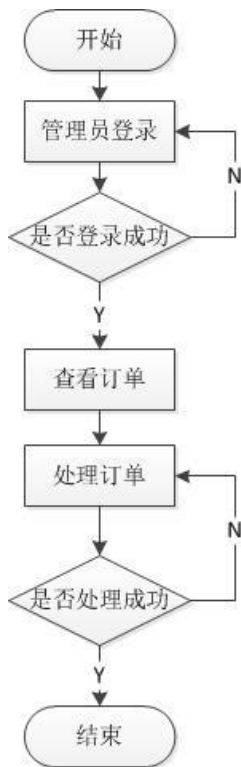


图 4-8 订单管理模块流程图

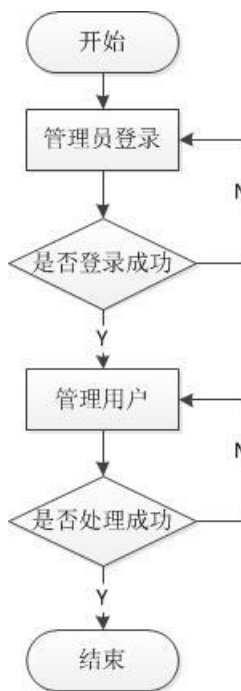


图 4-9 用户管理模块流程图

4.3.6 用户管理模块的设计

用户管理模块实现管理用户，即前台和后台的用户。系统的特定管理员可以在该功能模块上管理用户，如添加删除修改前台用户，也可以管理管理员用户，添加删除编辑用户的信息，出于对用户信息的安全考虑，这些功能都只有特定的管理员才可以进行操作。管理用户的流程图如图 4-9 所示。

4.4 本章小结

本章节主要介绍电子商城系统的架构与设计，分为三个部分，系统概要设计，

即系统的架构，数据库表的设计，每个功能模块的设计。在概要设计中介绍了系统的架构和设计思路，在数据库设计中，介绍了系统的所有实体以及实体之间的关系，借此将数据库的具体形式描述的比较清楚，让我们理解数据库的具体内容。在系统功能设计中，将各个模块分解介绍，借此将系统的具体功能描述的比较清楚，让我们理解系统如何实现的具体功能。下一章节主要介绍系统的实现，首先介绍系统的实现环境，按照功能模块划分详细介绍每个模块的实现方式，并展示部分界面效果图，其次从功能测试和性能测试两种方式介绍系统的测试情况。

5 系统的实现与测试

本系统的开发环境使用 Visual Studio 2010，数据库系统是 SQL Server 2008，服务器采用 IIS。现将详细介绍本系统的开发环境，系统各个功能模块的实现过程，实现的核心代码，以及实现后的部分效果图。

5.1 系统实现环境

操作系统：Windows 7，

集成开发工具：Microsoft Visual Studio 2010，

数据库系统：SQL Server2008，

项目运行服务：Internet 信息服务（IIS）管理器。

5.2 用户信息管理的实现

用户注册实现用户的功能，注册信息有部分为必填内容，先判断用户是否输入了必填信息，查询用户名是否已经存在，这些信息有不正确的情况，向用户显示提示信息。

用户登录主要提供用户登录系统，输入用户名、密码和验证码后，判断验证码的正确性，查询用户 ID 是否存在于数据库中并检测密码的正确性，信息不正确时提示用户重新输入用户名、密码和验证码。

用户信息修改实现修改自己信息的功能，当修改自己的信息时，系统先将用户的原信息显示在对应的控件中供用户查看，用户可以直接修改信息并提交，系统则检测信息是否完整且符合要求，有错误则修改失败并提示用户正确的操作，修改个人信息的部分代码实现如下所示。

```
protected void btnUpdate_Click(object sender, EventArgs e)
{
    //确保用户输入必要信息
    if (txtName.Text.Trim() == "" && txtPassword.Text.Trim() == "" && txtTrueName.Text.Trim()
    == "" && txtPhone.Text.Trim() == "" && txtEmail.Text.Trim() == "" && txtAddress.Text.Trim() ==
```

```

"" && txtPostCode.Text.Trim() == "") {
    {
        Response.Write(ccObj.MessageBoxPage("请输入完整信息!"));
    }
}
else
{
    //验证用户信息输入是否正确
    //更新用户信息表tb_Member
    ucObj.ModifyUser(txtName.Text.Trim(), txtPassword.Text.Trim(),
    txtTrueName.Text.Trim(), transfer(ddlSex.SelectedItem.Text.Trim()),
    txtPhone.Text.Trim(), txtEmail.Text.Trim(), txtAddress.Text.Trim(),
    txtPostCode.Text.Trim(), Convert.ToInt32(Session["UserID"].ToString()));
    Session["Username"] = "";
    Session["Username"] = txtName.Text.Trim();
    Response.Write(ccObj.MessageBox("恭喜您, 修改成功!", "Default.aspx"));
}
}
}

```

5.3 购物车的实现

购物车功能属于电子商城的核心功能，用于显示及管理用户即将购买的商品信息。用户浏览商品时，想买某件商品，点击商品信息中的“加入购物车”按钮，即可将商品添加到用户的购物车中，用户想查看自己的购物车，可以点击导航栏中的“购物车”进入自己的购物车，用户可以查看和修改购物车里的商品。

将商品添加到购物车的代码如下：

```

public void AddShopCart (DataListCommandEvent Args e)
{
    /*判断是否登录*/
    ST_check_Login();
    Hashtable hashCar;
    if (Session["ShopCart"] == null)
    {
        //如果用户没有分配购物车
        hashCar = new Hashtable(); //新生成一个
        hashCar.Add(e.CommandArgument, 1); //添加一个商品
        Session["ShopCart"] = hashCar; //分配给用户
    }
}

```

```

    }
    else
    {
        //用户已经有购物车
        hashCar = (Hashtable) Session["ShopCart"]; //得到购物车的hash表
        if (hashCar.Contains(e.CommandArgument)) //购物车中已有此商品，商品数量加1
        {
            int count = Convert.ToInt32(hashCar[e.CommandArgument].ToString()); //得到该
商品的数量
            hashCar[e.CommandArgument] = (count + 1); //商品数量加1
        }
        else
            hashCar.Add(e.CommandArgument, 1); //如果没有此商品，则新添加一个项
    }
}

```

将商品添加到购物车需要判断用户是否登录，未登录则提示用户登录，若用户没有购物车，生成一个购物车并将商品加入到购物车中，用户有购物车，直接将商品添加到购物车中，购物车没有此商品，添加一项，已经存在此商品则商品数量增加 1。

将购物车中的商品删除的代码如下：

```

protected void lnkbtnDelete_Command(object sender, CommandEventArgs e)
{
    hashCar = (Hashtable) Session["ShopCart"]; //获取其购物车
    //从Hashtable表中，将指定的商品从购物车中移除，其中，删除按钮(lnkbtnDelete)的
CommandArgument参数值为商品ID代号
    hashCar.Remove(e.CommandArgument);
    Session["ShopCart"] = hashCar; //更新购物车
    Response.Redirect("shopCart.aspx");
}

```

从购物车中删除商品时，先获取用户的购物车，将商品 ID 从购物车表中移除并更新购物车，将更新后的购物车显示给用户。清空购物车的代码如下：

```

protected void lnkbtnClear_Click(object sender, EventArgs e)
{
    Session["ShopCart"] = null;
    Response.Redirect("shopCart.aspx");
}

```



图 5-1 购物车界面效果

清空购物车时直接将用户的购物车属性设置为空, 当下次需要购物车时为用户创建一个新的购物车。购物车的运行效果如图 5-1 所示。

积分兑整点聚 0元抢购

商品销售服务台

您的购物车:

序号	商品ID	商品名称	数量	单价(¥)	总价(¥)
1	8	三星Galaxy Note	1	1800	1800

总价为: 1800¥ 商品数为: 1个

- 如果需要修改, 请到购物车处修改
- 如果已确认, 请正确填写下面的信息

收货人详细地址:

配送及运费: 邮局邮寄普通包裹 (10元/本) ▼

收货人姓名: ceshi *

联系电话: 2298989 *

电子信箱: 2222@11.com *

邮编: 521000 *

收货人详细地址: nicai *

备注: 111

您有什么要求, 请在备注中注明。

[提交订单](#)

图 5-2 生成订单的界面效果

生成订单需要先判断用户的购物车是否为空，以及用户的输入的收货信息是否符合规范，输入信息都符合要求后最后生成订单，并将购物车的清空。生成订单的运行效果如图 5-2 所示。

5.4 支付货款的实现

利用网上银行支付接口实现支付。用户将想买的商品添加到购物车之后，填写完用户的收货信息，点击提交订单即可生成订单，用户可以选择在线支付方式，系统将订单信息提交给相应银行的服务器；用户在跳转到的网上银行支付服务页面输入相关信息完成订单支付。支付界面效果如图 5-3 所示。

通过链接直接进入网银支付的页面，前往支付页面的实现方法：

```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect("GoBank.aspx?OrderID=" + Request["OrderID"]);
}
```

支付页面的代码如下：

```
public BankPay GetPayInfo()
{
    //从订单信息表中获取订单编号、订单金额
    string strSql = "select Round(TotalPrice,2) as TotalPrice from tb_OrderInfo where OrderID=" + Convert.ToInt32(Page.Request["OrderID"].Trim());
    DataTable dsTable = dbObj.GetDataSetStr(strSql, "tbOI");
    bankpay.Orderid = Request["OrderID"].Trim(); //订单编号
    bankpay.Amount = Convert.ToString(float.Parse(dsTable.Rows[0]["TotalPrice"].ToString())*100); //订单金额
    bankpay.OrderDate = DateTime.Now.ToString("yyyyMMddhhmmss"); //交易日期时间
    bankpay.Path1 = Server.MapPath(@"bank\user.crt"); //公钥路径
    bankpay.Path2 = Server.MapPath(@"bank\user.crt"); //拆分pfx后缀的证商品后的公钥路径
    bankpay.Path3 = Server.MapPath(@"bank\user.key"); //拆分pfx后缀的证商品后的私钥路径
    //下面是需要加密的明文字符串
    bankpay.Msg = bankpay.InterfaceName + bankpay.InterfaceVersion + bankpay.MerID + bankpay.MerAcct + bankpay.MerURL + bankpay.NotifyType + bankpay.Orderid + bankpay.Amount + bankpay.CurType + bankpay.ResultType + bankpay.OrderDate + bankpay.VerifyJoinFlag;
    return (bankpay);
}
```



图 5-3 支付货款界面效果

5.5 商品管理的实现

当管理员登录后台系统后，在首页中选择商品管理即可开始管理商品的信息。管理员可以添加新的商品信息，也可以添加新的商品类别，或者输入关键字搜索商品或类别，可以将选择的商品或类别删除。

管理员管理商品的实现，获取商品的 ID，从而取得商品的信息并将其显示在对应的控件上，运行效果如图 5-4 所示，实现代码如下所示：

```
/// 获取指定商品的信息，并将其显示在界面上↵
public void GetGoodsInfo()↵
{↵
    string strSql = "select * from tb_GoodsInfo where GoodsID=" +
Convert.ToInt32(Request["id"].Trim());↵
    SqlCommand myCmd = dbObj.GetCommandStr(strSql);↵
    DataTable dsTable = dbObj.GetDataSetStr(strSql, "tbBI");↵
    txtCategory.Text =
gcObj.GetClass(Convert.ToInt32(dsTable.Rows[0]["ClassID"].ToString()));↵
    txtName.Text = dsTable.Rows[0]["GoodsName"].ToString();↵
    txtAuthor.Text = dsTable.Rows[0]["Author"].ToString();↵
    txtCompany.Text = dsTable.Rows[0]["Company"].ToString();↵
    txtMarketPrice.Text = dsTable.Rows[0]["MarketPrice"].ToString();↵
    txtHotPrice.Text = dsTable.Rows[0]["HotPrice"].ToString();↵
    ImageMapPhoto.ImageUrl = dsTable.Rows[0]["GoodsUrl"].ToString();↵
    cbxCommend.Checked = bool.Parse(dsTable.Rows[0]["Isrefinement"].ToString());↵
    cbxHot.Checked = bool.Parse(dsTable.Rows[0]["IsHot"].ToString());↵
    cbxDiscount.Checked = bool.Parse(dsTable.Rows[0]["IsDiscount"].ToString());↵
    txtShortDesc.Text = dsTable.Rows[0]["GoodsIntroduce"].ToString();↵
}↵
```

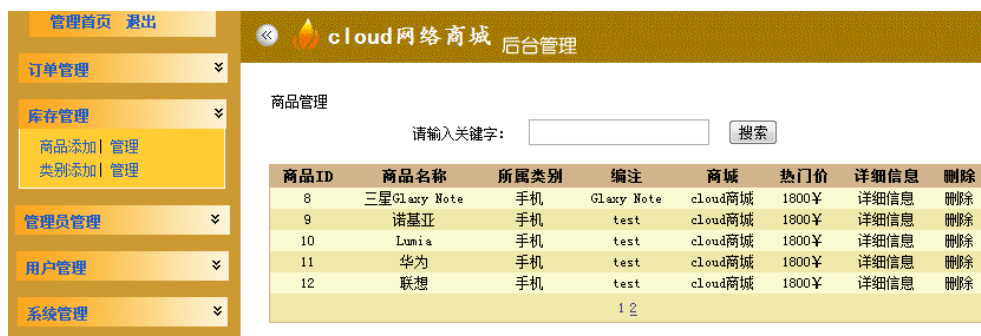


图 5-4 商品管理界面效果

添加商品的实现，需要输入商品的详细信息，系统对用户输入的信息判断是否已经存在，商品插入到数据库中才算添加商品信息成功。以下为添加商品的实现代码。

```
protected void btnSave_Click(object sender, EventArgs e)
{
    int IntClassID = Convert.ToInt32(ddlCategory.SelectedValue); //商品类别号
    string strGoodsName = txtName.Text.Trim(); //商品类别名
    string strGoodsDesc = txtShortDesc.Text.Trim(); //商品简短描述
    string strAuthor = txtAuthor.Text.Trim(); //商品作者
    string strCompany = txtCompany.Text.Trim(); //商品商城
    string strGoodsUrl = ddlUrl.SelectedValue; //商品图片存放路径
    float fltMarketPrice = float.Parse(txtMarketPrice.Text.Trim()); //商品市场价
    float fltHotPrice = float.Parse(txtHotPrice.Text.Trim()); //商品热门价
    bool blrefine = Convert.ToBoolean(cbxCommend.Checked); //是否推价
    bool blHot = Convert.ToBoolean(cbxHot.Checked); //是否热门
    bool blDiscount = Convert.ToBoolean(cbxDiscount.Checked); //是否最新
    string strSql = "select * from tb_GoodsInfo where GoodsName='" + strGoodsName + "' and
Author='" + strAuthor + "' and Company='" + strCompany + "'";
    DataTable dsTable = dbObj.GetDataSetStr(strSql, "tbBI");
    if (dsTable.Rows.Count > 0)
    {
        Response.Write(ccObj.MessageBox("该商品已存在!"));
    }
    else
    {
        //将商品信息插入数据库中
        string strAddSql = "Insert into
tb_GoodsInfo(ClassID,GoodsName,GoodsIntroduce,Author,Company,GoodsUrl,MarketPrice,HotPrice,
Isrefinement,IsHot,IsDiscount)";
    }
}
```



```

        strAddSql += "values (" + IntClassID + "',' + strGoodsName + "',' + strGoodsDesc
+ "',' + strAuthor + "',' + strCompany + "',' + strGoodsUrl + "',' + fltMarketPrice + "',' +
+ fltHotPrice + "',' + blrefine + "',' + blHot + "',' + blDiscount + "',')";
        SqlCommand myCmd = dbObj.GetCommandStr(strAddSql);
        dbObj.ExecNonQuery(myCmd);
        Response.Write(ccObj.MessageBox("添加成功!"));
    }
}

```

5.6 订单管理的实现

用户将想购买的商品添加到购物车中，输入自己的收货信息，如收货地址、收货人、电话等，提交订单，系统结算购物车里的商品，对于用户的商品订单，管理员可以在后台系统进行管理。

在网站后台的此管理模块中，当管理员选择导航栏“订单管理”中的任意一个选项，都可以在页面的主区域显示该类别的订单。在这区域中，管理员可以通过搜索查询指定订单，也可以删除选中的订单，可以点击查看订单的详细信息，但是管理员不能添加订单，订单只能在用户购买商品时由系统自动生成。

另外，订单管理模块可以将各类别订单的商品信息打印出来，以供做备份信息。订单列表界面效果如图 5-5 所示。



跟单员	单号	下单时间	货品总额	运费	总金额	配送方式	收货人	联系电话	订单状态	管理	删除
无	3	2014年1月4日	7200	120	7320	邮局邮寄快递包裹 (30元/本)	shaojie	2222222	未确认 未发货 未归档	管理	删除

图 5-5 订单列表界面效果

加载显示订单列表的代码如下：

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        /*判断是否登录*/
        ST_check_Login();
        //判断是否已点击“搜索”按钮
    }
}

```

```

ViewState["search"] = null;↵
pageBind(); //绑定订单信息↵
}↵
}↵

```

当用户单击页面中的“管理”链接按钮，将会在功能执行区中打开订单详细信息页面，用户可以在该页面中查询某一订单的详细信息，并且可以对订单状态信息进行修改。订单处理界面效果如图 5-6 所示。

cloud 网络商城
后台管理

订单号: 3

下单日期: 2014/1/4 0:57:33

订单信息

商品代号	商品名称	数量	热门价	小计	备注
11	华为	1	1800¥	1800¥	
13	超极本	1	1800¥	1800¥	
8	三星 Galaxy Note	2	1800¥	3600¥	

订单状态:

未确认 | 未发货 | 未归档

商品总金额:

7200.00¥

商品运费:

120.00¥

订单总金额:

7320.00¥

收货人信息

收货人姓名: shaojie
联系电话: 22222222
Email地址: 32@11.com
收货人地址: 111
邮政编码: 521000

修改订单状态

☐ 是否已确认

修改

打印

图 5-6 订单处理界面效果

处理订单的实现代码:

```

protected void btnSave_Click(object sender, EventArgs e)↵
{↵
    bool blConfirm = Convert.ToBoolean(chkConfirm.Checked); //是否被确认↵
    bool blSend = Convert.ToBoolean(chkConsignment.Checked); //是否已发货↵
    bool blEnd = Convert.ToBoolean(chkPigeonhole.Checked); //是否已归档↵
    int IntAdminID = Convert.ToInt32(Session["AID"].ToString()); //根单员ID代号↵
    //修改订单表中订单状态↵
    string strSql = "update tb_OrderInfo ";↵
    strSql += " set IsConfirm=' " + blConfirm + "', IsSend=' " + blSend + "', IsEnd=' " + blEnd↵
+ " ', AdminID=' " + IntAdminID + "', ConfirmTime=' " + DateTime.Now + "' ";↵
    strSql += " where OrderID=' " + Convert.ToInt32(Request["OrderID"].Trim()) + "' ";↵
    SqlCommand myCmd = dbObj.GetCommandStr(strSql);↵
    dbObj.ExecuteNonQuery(myCmd);↵
    Response.Write(ccObj.MessageBox("修改成功!", "main.aspx"));↵
}↵

```

5.7 用户管理的实现

会员用户列表的显示，向用户信息表 `tb_Member` 查询所有的普通用户存到 `dsTable` 变量中，将 `dsTable` 写入到对应的控件中供显示出来，用户列表的显示代码实现如下所示。

```
public void gvMemberBind()↵
{↵
    string strSql = "select * from tb_Member";↵
    DataTable dsTable = dbObj.GetDataSetStr(strSql, "tbMember");↵
    gvMemberList.DataSource = dsTable.DefaultView;↵
    gvMemberList.DataKeyNames = new[] { "MemberID" };↵
    gvMemberList.DataBind();↵
}↵

protected void gvMemberList_PageIndexChanging(object sender, GridViewPageEventArgs e)↵
{↵
    gvMemberList.PageIndex = e.NewPageIndex;↵
    gvMemberBind();↵
}↵
```

删除会员用户，通过用户 ID 在数据表 `tb_Member` 中查询该用户，先将该用户从数据表中删除，刷新数据库和界面显示，实现的代码如下所示。

```
protected void gvMemberList_RowDeleting(object sender, GridViewDeleteEventArgs e)↵
{↵
    string strSql = "delete from tb_Member where MemberID=" +
        Convert.ToInt32(gvMemberList.DataKeys[e.RowIndex].Value.ToString());↵
    SqlCommand myCmd = dbObj.GetCommandStr(strSql);↵
    dbObj.ExecNonQuery(myCmd);↵
    gvMemberBind();↵
}↵
```

管理员用户列表的显示和删除管理员与管理会员用户类似，这里就不赘述了。添加管理员，是通过手动的方式添加管理员用户。输入信息后先检测该管理员 ID 是否已经存在，向数据库中插入成功则添加管理员成功。添加管理员的实现代码如下所示。

```

protected void btnSave_Click(object sender, EventArgs e)
{
    string strSql = "select * from tb_Admin where AdminName='" + txtName.Text.Trim() + "'";
    DataTable dsTable = dbObj.GetDataSetStr(strSql, "tbAdmin");
    if (dsTable.Rows.Count > 0)
    {
        Response.Write(ccObj.MessageBoxPage("该用户名已存在!"));
    }
    else
    {
        string strName = txtName.Text.Trim();
        string strPwd = txtPassWord.Text.Trim();
        string strTrueName = txtTrueName.Text.Trim();
        string strEmail = txtEmail.Text.Trim();
        string strAddSql = "Insert into tb_Admin(AdminName,PassWord,RealName,Email)";
        strAddSql += "values('" + strName + "','" + strPwd + "','" + strTrueName + "','" + strEmail + "')";
        SqlCommand myCmd = dbObj.GetCommandStr(strAddSql);
        dbObj.ExecNonQuery(myCmd);
        Response.Write(ccObj.MessageBoxPage("添加成功!"));
    }
}

```

5.8 浏览记录的实现

用户在浏览商品时，将浏览的商品信息保存在浏览器的 cookie 里，由于单个 cookie 的大小不能超过 4096 字节，只能保存部分商品信息，如商品 ID，在向服务器请求其他页面时读出 cookie 里的商品 ID，将这些 ID 发送给服务器，查询这些商品的详细信息，作为浏览记录显示在相应的区域，这样做的一个好处就是无论用户是否登录都可以用到历史记录的功能，因为 cookie 是本地的一个文本，存放在硬盘中，在没有手动去删除的情况可以一直保存，可以设置 cookie 的属性以限制他的数量，这样就可以保存最近的部分历史记录。写入 cookie，先创建 HttpCookie 对象，设置 cookie 的属性并在本地硬盘生成 cookie 的文件，其实现代码如下：

```
HttpCookie cookie=new HttpCookie("History");//初始化并设置Cookie的名称↵
DateTime dt=DateTime.Now;↵
TimeSpan ts = new TimeSpan(1, 0, 0,0,0);//设置cookie的过期时间↵
cookie.Expires = dt.Add(ts);//添加过期时间↵
cookie.Values.Add("GoodsID", "userid_value");//写入商品ID↵
Response.AppendCookie(cookie);//确认生成cookie↵
```

读取 cookie，判断本地文件中的存在对应名称的 cookie，将商品 ID 从 GoodsID 的 cookie 对象中读取出来写入到字符串中供调用，其实现代码如下：

```
if (Request.Cookies["History"] != null) ↵
{↵
    String temp=Convert.ToString(Request.Cookies["History"].Values["GoodsID"])+↵
    "+Convert.ToString(Request.Cookies["GoodsID"].Values["userid"]); ↵
    if (temp=="") ↵
        Response.Write("NULL"); ↵
    else↵
        Response.Write(temp); ↵
} ↵
else↵
{ ↵
    Response.Write("error"); ↵
}↵
```

5.9 精准广告的实现

精准广告需要三个页面来实现，一个是 Default.aspx，用来产生 cookie，一个是 SSO.ashx，还有一个 GetCookie.aspx。网址 www.first.com 的前台页面就是 Default.aspx，通过地址 <http://www.first.com/Default.aspx> 来访问该页面。它没有后台代码，前台核心代码如下：

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="Admin10000.Web.Default" %>↵
var _frm = document.createElement("iframe");↵
_frm.style.display = "none";↵
_frm.src = "http://www.second.com/SSO.ashx";↵
document.body.appendChild(_frm);↵
```

SSO.ashx 是 www.second.com 的页面，后台核心代码如下：

```
public class SSO : IHttpHandler
{
    public void ProcessRequest(HttpContext context)
    {
        HttpCookie cookie = new HttpCookie("name", "www.Admin10000.com");
        cookie.Domain = "second.com";
        cookie.Path = "/";
        cookie.Expires = DateTime.Now.AddMinutes(10000);
        context.Response.Cookies.Add(cookie);
        context.Response.ContentType = "text/plain";
        context.Response.AddHeader("P3P", "CP=CAO PSA OUR");
        context.Response.Write("");
    }
}
```

GetCookie.aspx 是 www.second.com 下的页面，后台核心代码如下：

```
namespace Admin10000.Web
{
    public partial class GetCookie : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Request.Cookies["name"] != null)
            {
                Response.Write(Request.Cookies["name"].Value);
            }
        }
    }
}
```

用户首先访问第一个页面，即 Default.aspx，该页面会加载元素 `iframe`，调用第二个页面，也就是 SSO.ashx，其后台代码会在本地创建一个关于第二个页面的 cookie，当用户访问 GetCookie.aspx 时，其实就是在获取第二个页面的 cookie 的信息，而用户浏览商品的记录就包含在 cookie 里，也就可以将获取到的信息作为他用，如分析后定向投放广告。通过这样的一个方式就实现了在第一个网页创建的 cookie 可以在第二个网页访问，即使他们的一级域名不相同。第一个页面域下的 cookie 值被元素 `iframe` 的属性 `src` 作为 `get` 方法的参数重定向到 SSO.ashx 页面上，SSO.ashx

获取所传过来的 cookie 值,所获取到的值就被写入到 cookie 中,这样就实现了 cookie 跨域的访问。

在用户访问其他网站时,读取该系统的 cookie 获取用户浏览的商品信息,对用户进行广告投放,当然这些需要与对应的网站合作,而且需要用户没有禁用第三方 cookie 才可以实现精准广告的投放,使潜在的用户购买对应的商品。

5.10 系统测试

系统测试主要通过功能测试和性能测试来发现系统的故障,并进行排除。

5.10.1 功能测试

功能测试主要是测试在需求分析中所设计的功能是否都能够实现,这需要仔细去验证各个功能的实现情况,否则需要找出并解决出现的 BUG。功能测试的操作情况如表 5-1 所示。

表 5-1 功能测试表

测试的功能	测试内容描述	测试结果	处理情况
用户登录	用户是否能够登录成功	正常	
用户注册	用户能否注册,并且注册信息写入数据库	正常	
修改信息	用户信息修改后是否能够存入到数据库中	正常	
查询商品	输入关键字后是否显示包含的商品列表	正常	
浏览商品	登录系统后是否显示商品,并且是否可以点击查看商品信息	正常	
购买商品	用户点击购买是否可以将商品添加到购物车中,形成订单并且对订单付款	正常	
管理商品	管理员是否可以操作商品信息,如添加、修改和删除	正常	
管理订单	管理员用户是否可以修改订单的状态	正常	
管理用户	系统管理员用户是否可以添加普通管理员用户,并且删除修改普通用户信息	正常	

浏览记录	用户浏览的记录能否显示在对应的位置	正常	
精准广告	是否可以将用户浏览的商品显示在其他网站的界面里	商品信息显示异常	需要在相应的网站添加代码

5.10.2 性能测试

性能测试主要是测试系统在满足功能测试的基础上是否具有良好的连接性和互动性以及实验的性能，还有各个页面之间切换是否流畅，页面添加到数据库的数据是否正常，数据安全性如何。性能测试如表 5-2 所示。

表 5-2 性能测试表

性能测试内容	测试描述	测试结果	处理情况
页面风格	页面风格是否一致	一致	
页面连接	页面连接是否流畅	流畅	
按钮触发	是否具有触发指定功能	具备	
数据库连接	数据库连接是否正常	连接正常	
系统安全	异常用户是否可以登录	无法登录	
数据安全	用户是否可以修改数据库	可以修改	需要设置数据库权限
浏览器测试	是否支持不同浏览器	支持	

通过功能测试和性能测试，该系统已经达到了设计的功能需求，核心功能也具有一定的有效性、健壮性和可靠性。

5.11 本章小结

本章节主要介绍电子商城的系统实现，详细介绍每个功能模块在所掌握的技术下是如何实现的，包括功能核心代码的展示，功能实现后的界面效果，以及系统的功能测试和性能测试。下一章节主要对毕业设计的总结和展望。

6 总结与展望

6.1 全文总结

基于互联网的发展，电子商务成为一个新兴的营销模式，作为IT与贸易的一种结合体，在以往的十几年中发展迅猛，它给现代社会提供着巨大的便利，必将成为未来商务发展的主流方向。而电子商城作为电子商务的一种实现方式，它更具有平民化的特点，必然普及程度会更为宽广。随着电子商务的发展，电子商城在未来也会成为热门的研究领域。

在毕设项目之前是调研阶段，主要研究电子商务的发展背景，作为电子商务的基础支持，电子商城系统是必不可少的一部分，从而体现该课题的研究意义。在研究课题之前，就现在国内外对于电子商务的研究状况进行了分析，研究目前部分电子商务网站存在的优点和问题，以及思考问题缺陷的解决办法。

本论文主要介绍基于ASP.NET的电子商城系统设计与实现。首先介绍软件开发的两种方案——C/S和B/S架构，以及对它们的优缺点进行分析，结合目前的现状，最后以B/S的开发方案进行项目的开发。电子商城系统的开发涉及到多种技术的运用，本文也介绍了比较关键的几种技术，如.NET、ASP.NET、SQL Server等。项目的需求分析就是项目功能的确定工作，结合目前几大主流的电子商城，如淘宝、京东、苏宁等，分析电子商城系统所需要具备的功能，以及这些功能的使用角色、场景和方式，确定用例，其次还需要分析系统需要具备的一些质量性的要求。

系统的架构与设计是系统开发的核心部分，决定着项目能否成为符合要求的系统。系统架构采用比较主流的架构方式——三层架构，这一架构方式比较适合网站系统的开发。数据库设计主要确定系统所涉及的实体、如用户、商品、订单等，确定实体之间的关系，定义数据表的属性字段。系统功能设计需要将功能模块进行更详细的划分，确定每个功能所要完成的工作。系统设计之后就实现和测试，就是将设计好的每个功能按照设计的思路进行编码实现，并完成测试，确保每个功能正确的实现，最终形成一个功能稳定的系统。

综上所述，本文通过对ASP.NET技术、数据库SQL server以及C#语言等相关知识的应用，完成了一个电子商城系统的开发实例。

6.2 课题展望

本课题的研究还有很多问题没有讨论和考虑到，主要是因为部分条件限制，还有时间也不是很充足。

(1) 在需求分析时，一个完善的电子商务网站系统的功能是很完善的，如商品的详细介绍和用户的评论，作为一个购物网站，跟物流有很大的关系，商品的物流信息需要能够实时的反馈给用户。在以后的研究中需要将这些功能考虑进去，将更优的辅助功能添加进系统。

(2) 在设计数据库时，考虑的实体还是不够完善，管理员跟商家加以区分，毕竟管理员和商家的考虑工作不一样，还有商品的信息还有很大的完善空间，需要再给商品添加更全的信息，如商品的视频展示功能，使用户能够更加了解所要买的商品。

(3) 消费者在购买商品时需要与商家交流，应该添加沟通的功能，方便用户与商家更好的交流，也方便用户给商家提意见和建议，促进商品的销售。

(4) 系统的界面还存在不少的问题，如不够细致美观，这些由于开发时间的限制，实现的功能不够全面，这些都将在以后的使用过程中不断的进行完善。

(5) 在信息安全部分，需要加入更强的安全策略，比如管理员的权限等级，不同等级设置不同的权限，保证用户的信息安全。

(6) 系统采用的 B/S 架构，需要添加支持手机等移动终端的设备，这方面在设计开发时没有这样的考虑，需要在设计中将这些考虑在内。

因为研究的时间限制，还有本人的技术水平有限，论文中肯定存在一些错误，这也是不可避免的，望各位老师和同学能够批评指正。

致 谢

这次毕业设计的顺利完成，需要感谢很多人的帮助。首先要感谢的是我的导师吕泽华老师，从研究生的入学时就开始对我悉心教导，对我的专业知识的学习起到了很大的帮助作用。这次的毕业设计，他一直是耐心指导我，从毕设的选题、设计、开发，到论文的撰写，从老师那里获得了很多的帮助。这其中也遇到过很多棘手的问题，都是老师给我提供了指导，得以一一解决，使我最后能够顺利的完成毕业设计。在此对我的导师表示由衷的感谢。最后感谢帮助我的各位老师、同学和朋友。由于本人的能力有限，该毕业设计难免有不足之处，望大家谅解！

参考文献

- [1] 金莹, 崔基哲. 电子商务与实体店结合现状与发展趋势. 科技创新与应用, 2014. 8(14): 251-252.
- [2] 钟平, 蔡昭权, 卢庆武. 基于 ASP.NET 的 B2C 电子商城系统的设计与实现. 计算机与现代化, 2009(5): 39-42.
- [3] 张玉孔. B2C 型中小企业电子购物系统的设计. 商城现代化, 2006(33): 119.
- [4] 张昊. 探索中国 B2C 电子商务的发展之路. 科协论坛, 2007(4): 121.
- [5] 李云强, 赵东江. B2C 电子商务网站架构研究. 网络与信息, 2007(9): 30.
- [6] 刘玉, 王素芬, 郭楠. 通用 B2C 电子商务网站构架研究, 微型电脑应用 2005, 21(2): 51-52.
- [7] Thai T. Lam H Q. .NET Framework Essentials. O'Reilly 2003: 64-77.
- [8] Team W A, Loesgen B, Eide A, et al. Professional ASP. NET web services[M]. Wrox Press Ltd., 2001: 103-116.
- [9] Esposito D. Building web solutions with ASP. NET and ADO. NET. Microsoft Press, 2002: 204-220.
- [10] 毛德祥, 罗荣阁. 基于 ASP.NET 技术的 Web 应用程序三层设计模型. 微型电脑应用, 2002, 18(3): 26-28.
- [11] 范振钧. 基于 ASP. NET 的三层结构实现方法研究. 计算机科学, 2007, 34(4): 289-291.
- [12] 邵丽萍, 肖世德. 新一代 Web 开发技术 ASP. NET 的发展与探析. 微计算机信息, 2005, 21(1): 190-192.
- [13] 建峰, 律松, 晓栓. ASP. NET 开发技术详解. 人民邮电出版社, 2005: 153-168.
- [14] 尚俊杰. ASP. NET 程序设计. 清华大学出版社, 2004: 256-272.
- [15] 冯家刚. 中小型城市电子商城系统的设计与实现. 北京工业大学, 2010: 89-102.
- [16] 陈运海. 基于 ASP. NET 的电子商城管理系统的构建. 武汉船舶职业技术学院学

-
- 院学报, 2006, 4(6): 11-13.
- [17] 李劲, 谢江. 电子商城的设计及安全对策. 湖北民族学院学报: 自然科学版, 2006, 23(4): 346-348.
- [18] 李春妍. 网上电子商城的设计与实现. 数字技术与应用, 2010 (3): 32-32.
- [19] Bernstein P A, Cseri I, Dani N, et al. Adapting microsoft SQL server for cloud computing. In: proceedings of Data Engineering (ICDE), 2011 IEEE 27th International Conference on. IEEE, 2011: 1255-1263.
- [20] MacLennan J, Tang Z H, Crivat B. Data mining with Microsoft SQL server 2008. John Wiley & Sons, 2011:124-135.
- [21] Lang D T. XML: Tools for parsing and generating XML within R and S-Plus. R package version, 2012: 39-41.
- [22] Sceppa D. Microsoft Ado. Net (Core Reference). Microsoft Press, 2002:56-71.
- [23] Prosis J. Programming Microsoft. NET. Microsoft Press, 2002:66-74.
- [24] Mössenböck H, Beer W, Birngruber D, et al. . NET Application Development: With C#, ASP. NET, ADO. NET, and Web Services. Pearson Addison Wesley, 2004:189-203.
- [25] Meijer E, Beckman B, Bierman G. Linq: reconciling object, relations and xml in the .net framework. In: proceedings of the 2006 ACM SIGMOD international conference on Management of data. ACM, 2006: 706-715.
- [26] Thai T, Lam H. . NET framework essentials. O'Reilly Media, Inc., 2003:345-361.
- [27] Richter J. Applied Microsoft. NET Framework Programming. Redmond: Microsoft Press, 2002:268-274.
- [28] Rajshekhar A P. . Net Framework 4.5 Expert Programming CookGoods. Packt Publishing Ltd, 2013:214-232.
- [29] 赵阳, 赵谦, 武芃. Microsoft. NET Framework 结构解析. 微机发展, 2003, 13(a02): 51-53.
- [30] Tegels K, Baer T. NET Framework 编程思想. 清华大学出版社, 2002:78-85.
-

-
-
- [31] 郭彬, 张永平. 基于 .NET Framework 的三层软件体系架构的研究. 科技资讯, 2009 (33): 17-18.
- [32] 邓劲生, 张晓明. Visual Studio .NET Framework 技术内幕. 2003:95-104.
- [33] 裴容生, 杨勇. Web 应用程序安全控制的实现. 成都大学学报: 自然科学版, 2006, 25(3): 191-194.
- [34] 邵鹏鸣. ASP. NET 技术与应用. 高等教育出版社, 2004:135-154.
- [35] Watson K. C#入门经典. 清华大学出版社, 2002:368-382.
- [36] 三才, 薰尹. 精通 C#与 ASP. net 程序设计. 中国青年出版社, 2001:236-248.
- [37] 立民. 用实例学 ASP. NET:使用 C#与 ADO. NET. 电子工业出版社, 2004:158-173.
- [38] 马俊, C# 网络应用编程基础. 人民邮电出版社, 2006:179-195.
- [39] 王春楠, XML 高级编程. 机械工业出版社, 2001:233-252.
- [40] 瞿裕忠, 张剑锋. XML 语言及相关技术综述. 计算机工程, 2000, 26(12): 4-6.
- [41] 张素智, 卢正鼎. XML 数据库及其应用研究. 计算机工程与应用, 2002, 38(8): 32-36.
- [42] Tal T, Quadri R F, Muzzin A, et al. Cookie-cutter halos: the remarkable constancy of the stellar mass function of satellite galaxies. arXiv preprint arXiv, 2014:104-106.
- [43] 何智勇, 毛燕琴, 沈苏彬. HIPL 改进的 Cookie 机制研究与设计. 南京邮电大学学报: 自然科学版, 2011, 31(3): 21-26.
- [44] 朱远文, 张煜, 常畅. 基于 Cookie 的安全防护技术研究. 信息网络安全, 2012 (9): 46-49.
- [45] 胡畔, 钟央, 凌力. 一种新的基于 cookie 的互联网个性化推荐系统设计. 微型电脑应用, 2013 (9): 44-47.
- [46] 李爱国. 基于 Cookie 的购物车设计与实现. 信息技术, 2013 (6): 60-62.