

中文摘要

摘要：随着全球经济信息化步伐明显加快，愈来愈多的企业意识到效率管理和科学管理的重要性。销售是企业经营的重要组成部分，它可以体现一个企业的经营情况和效益，建立一个科学高效的企业销售管理系统，可以有效地提高企业宏观控制和经营管理的质量，是企业走现代化管理的重要一步。

本文在这样的前提下，进行了基于.NET 技术的企业销售管理系统的设计与实现。文章首先概括地介绍了企业销售管理的现状，企业应用体系结构的发展和多层结构的优势，并由此引入了.NET 技术，简要阐述了.NET 组成及其关键技术。接着，从企业销售管理系统的需求分析入手，对系统进行了总体设计。最后，详细阐述了系统中销售商品管理、发票信息管理、按商品名称统计销售信息和用户管理模块的实现细节。

关键词：.NET；B/S 结构；销售管理系统

分类号：

ABSTRACT

ABSTRACT:

Along with the acceleration of the Economy Informationization in the world, more and more companies realize the importance of efficiency management and scientific management. Distribution is a substantial part of a company, as it can reflect the operation and profit of that company. Therefore, to have a scientifically efficient Enterprise Sales Management System can effectively enhance the macro-control and the management quality of the company.

Under such a premise, this paper works on the design and implementation of the Enterprise Sales Management System based on .Net technology. This paper first briefly introduces the company's current situation of the sales management, the development of the application system structure and the advantages of the multi-layered structure. And then it talks about .Net technology, focusing on the structures and the key techniques. Next, it introduces the overall design of the system with the demand analysis of the enterprise sales management. And finally, it elaborates on the implementation details.

KEYWORDS: .Net; B/S Structure; Sales Management System

CLASSNO:

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：任勇

导师签名：任勇

签字日期：2007年12月20日

签字日期：2007年12月20日

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：

签字日期： 年 月 日

致谢

本论文的工作是在我的导师冯凤娟老师的悉心指导下完成的，冯凤娟老师严谨的治学态度和科学的工作方法给了我极大的帮助和影响。在此衷心感谢三年来冯老师对我的关心和指导。

在实验室工作及撰写论文期间，文博，王伟、于文庆等同学对我论文中出现的问题给予了热情帮助，在此向他们表达我的感激之情。

深深的感谢我的家人，谢谢他们对我无微不至的关怀，是他们给予了我物质和精神上的支持和鼓励，让我能够全身心的投入研究生的学习、工作中。

最后，感谢在百忙之中为我论文审稿的各位老师。希望母校北京交通大学明天会更好！

第 1 章 绪论

1.1 课题研究背景和意义

销售管理在现代企业经营管理中占据着重要的地位，一个企业销售管理的水平几乎决定着企业的经济命脉。随着信息技术的迅速发展，市场竞争日趋激烈，信息化建设已经成为了企业参与竞争的主体工程。采用现代化生产技术，用 CIM 哲理来组织、管理、改造现行企业，旨在提高制造型企业的生产率和响应能力，由此，企业的所有功能、信息、组织管理方面都是一个集成整体的各个部分。管理信息系统(MIS)是企业 CIMS 工程中的一个分支，销售管理系统是企业 MIS 建设中重要的组成部分。

MIS 是集计算机技术、网络通信技术为一体的信息系统工程。采用先进、适用、有效的企业管理体制，运用于企业管理的各个环节和层次，可以改善企业的经营环境、降低经营生产成本，提高企业的竞争力;使企业领导层的生产、经营销售决策依据充分，更具科学性，更好地把握商机，创造更多的发展机会;有利于企业科学化、合理化、制度化、规范化的管理，使企业的管理水平跨上新台阶，为企业持续、健康、稳定的发展打下基础。企业信息化是用信息技术改造传统产业和实现信息、化带动工业化的突破口。随着经济全球化的进一步加快，新一轮产业结构调整正在进行，信息化无疑将成为提高其整体素质和综合竞争力的唯一选择^[1]。

在 CIMS 和 MIS 信息化发展的推动下，各企业正在采用先进的网络技术，通过智能化的网络设备及软件，实现对计算机应用系统的有效控制和管理;销售管理系统应该满足当前企业各个层次、各个环节的管理、核算需求。本文就是对企业信息管理系统中企业销售管理系统的设计与实现，旨在推动企业信息化进程。

1.2 管理信息系统概述

1.2.1 管理信息系统的概念

管理信息系统(MIS)是一种以计算机为基础、为支持管理活动和管理功能的信息系统。更具体的定义为：管理信息系统是由人和计算机结合的管理信息进行收集、存储、维护、加工传递和使用的系统。它是一种集成化的人机系统，能为

组织机构的作业管理和决策职能提供信息支持。

1.2.2 管理信息系统的功能

企业是一个复杂的系统，整个系统中各环节的活动构成了企业的经营生产活动。在企业的经营生产活动中贯穿两个运动过程，一是物流，一是信息流。物流是企业的基本流，它的运动进程产生各种运动信息，而企业管理者针对这些信息做出决策，以决策信息再控制物流运动，规划调节物流数量、方向、速度、目标，使之按一定目的和规划运动。信息流是管理的基础和管理的体现，管理信息是企业的神经中枢，是生命线。这在信息时代更为重要。管理信息系统不仅为企业管理提供决策所需要的一切信息，而且还有一定的决策、控制功能。管理信息系统具有以下六个功能：

(1) 数据处理：数据的收集,输入,传输,存储,加工处理和输出。

(2) 事务处理：将管理人员从繁重的重复性的事务处理中解脱出来，以更多的精力思考管理问题，从事创造性劳动。

(3) 预测功能：运用数学、统计或模拟等方法，根据过去的数据预测未来的情况。

(4) 计划功能：合理安排各职能部门的计划，并按照不同的管理层提供相应的计划报告。

(5) 控制功能：对计划的执行情况进行监测、检查，比较执行情况与计划的差异，并分析其原因，辅助管理人员及时用各种方法加以控制。

(6) 辅助决策功能：运用数学模型，及时推导出有关问题的最优解，辅助各级管理人员进行决策。

1.3 销售管理系统概述

1.3.1 企业销售管理的应用现状

现代企业的发展规模越来越趋于集团化，它不再是计划经济体制下进行单纯生产的工厂。在市场经济体制下，必须面对市场、参与市场竞争。这样，许多企业逐渐形成了自己的销售公司，并在全国各地设有分公司、办事处等分销机构;甚至设有直接面向消费者的销售商店，形成了庞大的销售网络。然而，面对这样数据量庞大、对速度和效率要求都很高的业务流程，这一行业中的绝大部分企业，尤其是中小型企业却仍然使用着传统、手工的销售管理模式。在日益激烈的竞争

面前，这些企业要想进一步提高销售管理力度，就不得不正视传统销售模式中所固有的一些弊端。比如，企业在传统模式下，根据已有数据进行一些统计计算时，却往往发现计算出的库存变动与实际清点数不符。公司对销售人员的管理也颇感头疼，在绩效考核方面也很难做到足够的公允。与此同时，在手工操作模式下，工作人员每天下班前都要手动提交繁复的调查表和订单，这不仅花费了他们大量的时间，延误后续工作，并且也容易造成因笔误而引起的各种问题。公司不得不使用大量的人力来对几乎每天都浩如烟海的订单进行人工检查和手工开票操作。此外，对于市场调查这样的庞大的数据，公司也必须要指派专门的人员来将这些数据录入电脑系统。即使他们的速度再快，市场调查的报表也要到第二天才可以被看到。因此，对销售机构的管理迫在眉睫，其管理程度的好坏直接关系到企业产品在市场上的占有率，是企业资金、利润的直接来源。一些企业管理层开始考虑上一套切实有效的系统，以根本解决当前销售管理中存在的问题，并提高整个销售和市场管理的力度和销售市场团队的效率。

1.3.2 销售管理系统开发的目的

开发中小型企业销售管理系统时应充分考虑企业销售实际，以对产成品分布存储、统一调拨、就近送货为原则，围绕销售库存展开，实现集中式管理。系统不但应适合在全国形成营销网络的企业集团，同时也应适用专门从事代理、分销、批零、服务等商业企业。系统应具有相对的独立性，可单独应用于企业的销售机构。可考虑设有同其它子系统的数据和功能接口，使其与财务、决策、生产系统集成使用，实现企业全面的现代化管理。

系统开发主要应实现以下目标：

1.实现信息共享:信息集中存放于项目管理数据库中，任何人只要有权限就可以在自己的电脑前查看所有项目资料。

2.实现网络办公:产品的订购、审批、产品信息的查询、收集和发送以及报表的打印、产品数据的统计可以在办公室的电脑上完成，极大减少人员在各部门往来移动的时间和频率，减少会议次数，从而极大提高工作效率。

3.实现规范管理:公司职能管理部门严格按公司销售管理相关程序、规定、要求进行科学管理。

4.控制产品:监督部门对产品销售进行实时监控，随时得到产品即时的报价、成本、质量、进货量、成交额等等情况和目前存在的问题。根据产品实际销售情况和计划的对比，动态调整进货计划和报价，如计划的提前或延期，是否追加进货量，范围是否变更或订单终止等。

5.科学决策:为企业高层决策者(如总经理)服务,为他们的宏观决策提供科学的依据。

第 2 章 系统设计主要技术

2.1 .NET

.Net 是微软公司建立的新一代开发平台,.Net 开发平台使得开发者创建运行在 Internet Information Server (IIS)(互联网信息服务器)Web 服务器上的 web 应用程序更为容易,它也使创建稳定、可靠而又安全的 windows 桌面应用程序更为容易。

2.1.1 .Net 的组成

.Net 开发平台包括以下内容

(1) .Net Framework, 包括 CLR (通用语言运行环境), 它是用于运行和加载应用程序的软件组件; 新的类库, 分级组织了开发者可以在他们的程序中用来显示图形用户界面、访问数据库和文件以及在 Web 上通信的代码集。

(2) .Net 开发者工具, 包括: Visual Studio.Net 集成开发环境, 用来开发和测试应用程序; .Net 编程语言, 用于创建运行在 CLR 下, 并且使用类库的应用程序。

(3) ASP.NET, 是一个取代以前的 ASP 的特殊类库, 用来创建动态的 Web 内容和 Web 服务器应用程序。

2.1.2 .Net 的优势

(1) 标准的集成: XML、SOAP 以及其他

以前, Microsoft 的构件都是建立在 COM/DCOM 上的, 不能够很容易地和其他软件平台协同工作。除了 COM 缺乏互用性以外, 数据是另一个问题, ADO 能够允许数据的自由访问, 但把这个数据从一个地方传到另一个地方的同时很容易出问题。而 SOAP 是一个基于 XML 能同 Web 服务进行通信的协议。SOAP 的集成使得任何客户端都能够通过编程来轻易地访问, 而不管客户端正在运行的是否是 Microsoft 的操作系统。

(2) 轻松部署

COM 部署一直是最困难的问题, COM 大量使用了 Windows 注册表来定位机器里的组件。它的好处是, 已注册的组件只会有一个单一的实例, 所有的应用程序都会使用同一个版本, COM 承诺新版本能够保持和旧版本的兼容性, 但开发人员有时可以不受限制而破坏这种兼容性。而 .Net 使用了不同的方法: 它根本就不

使用注册表, Microsoft 推荐的方法是, 让组件都成为每个应用程序的本地组件。

(3) Web 服务支持

有了 .Net, 开发 Web 服务变得非常简单。可以使用记事本来创建简单的 Web 服务, 甚至还不需要通过编译器来运行他们, 只需简单的调用他们, .Net 就会对其进行编译, 生成一个测试页面, 这样可以查验他们是否能够正常工作。 .Net 拥有所需要的全部工具, 并能生成所需要的文件。

(4) 用于任何 .Net 语言的标准工具集

有一个统一的 IDE、调试器和其他类似的工具, 这就可以让其他公司把他们自己的语言插入到 IDE 里, 同时还能获得 .Net 工具的全部支持。

(5) 对移动设备的支持

在发布 Visual Studio.NET 后不久, Microsoft 就发布了 Microsoft 移动因特网工具包 (MMIT), 这样就可以利用 .NET 创建更多的移动应用程序。这允许直观地把控件拖放到用于移动通信设备的窗体里。这个工具包处理的是编写适当的标记语言。

(6) 自我管理的代码

.NET 的管理代码特性是很重要的, 为了减少错误、创建更多的可伸缩的应用程序等。 .NET 能够处理的任务如分配和恢复内存、创建和清除线程和进程, 以及处理运行代码的访问许可。

(7) 对平台非依赖性

虽然 .NET 是为 Microsoft 的操作系统创建的, 但 Microsoft 向 ECMA 标准委员会提交了 Framework 和 C# 的一部分。MONO 计划, 一个用于 LINUX 的 .NET 工具, 包括了 Framework 和一个 C# 的编译器, 这就意味着有许多应用程序, 尤其是基于 Web 的应用程序, 一次写好就能够运行在多种平台上。

2.1.3 .Net Framework

在系统架构中, .Net Framework 位于 .Net 工具之下, 而在所用的 .Net 通信协议之上。 .Net Framework 使开发人员可以更容易建立网络应用程序和 Web Services, 它提供了生成、部署、扩展和维护这些 Web Services 的途径。 .Net Framework 构成如图 2-1 所示。

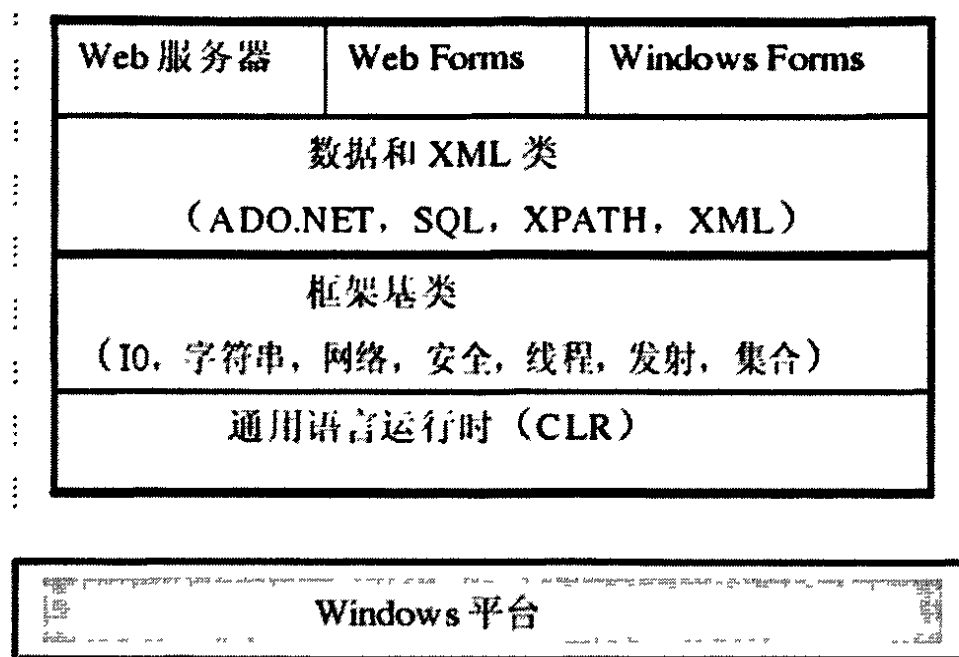


图 2-1 .Net Framework 构成

从上图可以看出通用语言运行时以及它所提供的一组基础类库构成了 .Net 框架的基础；在开发技术方面，.Net 框架提供了全新的数据库访问技术 ADO.Net，以及网络开发技术 ASP.Net 和 Windows 编程技术 Windows Forms；在开发语言方面，.Net 提供了 VB，VC++，C#，Jscript 等多种语言支持；而 Visual Studio.Net 则是全面支持 .Net 的开发工具^[2]。

2.2 ASP.NET

2.2.1 ASP.NET 的特性

ASP.NET 是一种建立在通用语言上的程序构架，能被用于一台 Web 服务器来建立强大的 Web 应用程序。ASP.NET 具有许多比现在的 Web 开发模式强大的优势。

(1) 增强的编译性能 ASP.NET 是在服务器上运行的编译好的公共语言运行库代码，它可利用早期绑定、实时编译、本机优化和盒外缓存服务，这相当于在编写代码行之前，便显著提高了性能。

(2) 可视化的开发工具支持 ASP.NET 框架补充了 Visual Studio 集成开发环境中的大量工具箱和设计器。这一强大的工具提供了 WYSIWYG (What you see is what you get, 所见即所得) 编辑、拖放服务器控件和自动部署等功能。

(3) 灵活性和威力性由于 ASP.NET 基于公共语言运行库，因此 Web 应用程

序开发人员可以利用整个平台的威力和灵活性。.NET 框架类库、消息处理和数据访问解决方案都可从 Web 无缝访问。ASP.NET 也与语言无关，所以可以选择最适合应用程序的语言，如 VC++、VB、Jscript 及其支持 .NET 框架的语言，或跨多种语言分割应用程序。同时，公共语言运行库的交互性保证在迁移到 ASP.NET 时保留基于 COM 开发中的现有投资。

(4) 简易性 ASP.NET 使执行常见任务变得容易，从简单的窗体提交和客户端身份验证到部署和站点配置。同时，公共语言运行库利用托管代码简化了开发。

2.2.2 ASP.NET 的运行环境

ASP.NET 应用程序的运行环境如图 2-2 所示。

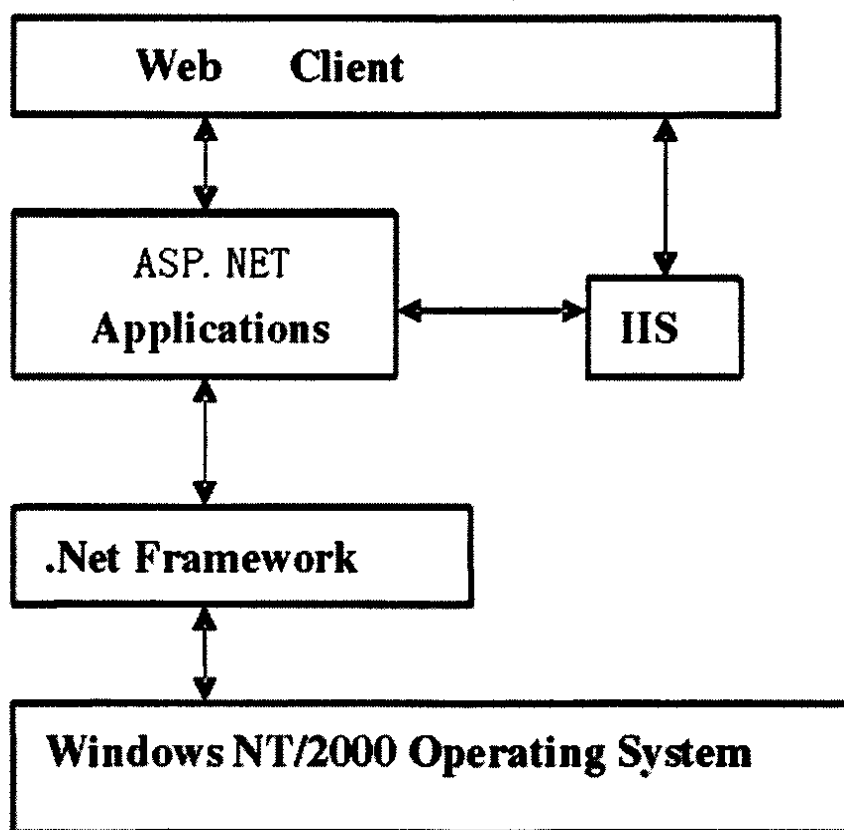


图 2-2 ASP.NET 运行环境

2.2.3 ASP.NET 的关键技术

(1) 面向对象技术在 .NET 中，所有的变量、系统服务等都是以对象的形式存在，各种 .NET 开发语言也都是完全的面向对象的程序设计语言。整个 .NET 的程序设计概念都是以面向对象的方式思考的，采用面向对象的机制，其目的就是用

到对象的属性 (Property)，方法 (Method) 和事件 (Event)。

面向对象技术是一种以对象为基础，以事件或消息来驱动对象执行处理的程序设计技术。它以数据为中心而不是以功能为中心来描述系统，数据相对于功能而言具有更强的稳定性。它将数据和对数据的操作封装在一起，作为一个整体来处理，采用数据抽象和信息隐蔽技术，将这个整体抽象成一种新的数据类型——类，并且考虑不同类之间的联系和类的重用性。类的集成度越高，就越适合大型应用程序的开发。另一方面，面向对象程序的控制流程由运行时各种事件的实际发生来触发，而不再由预定顺序来决定，更符合实际。事件驱动程序执行围绕消息的产生与处理，靠消息循环机制来实现。更重要的是，可以利用不断扩充的框架产品 MFC (Microsoft Foundation Classes)，在实际编程时可以采用搭积木的方式来组织程序，站在“巨人”肩上实现自己的愿望。面向对象的程序设计方法使得程序结构清晰、简单，提高了代码的重用性，有效地减少了程序的维护量，提高了软件的开发效率。

对象(Object)由属性(Attribute)和行为(Action)两部分组成。对象只有在具有属性和行为的情况下才有意义，属性是用来描述对象静态特征的一个数据项，行为是用来描述对象动态特征的一个操作。对象是包含客观事物特征的抽象实体，是属性和行为的封装体，在程序设计领域，可以用“对象=数据+作用于这些数据上的操作”这一公式来表达。

类(Class)是具有相同属性和行为的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和行为两个主要部分，类是对象集合的再抽象。类给出了属于该类的全部对象的抽象定义，而对象则是符合这种定义的一个实体。所以，一个对象又称作类的一个实例(Instance)。

消息(Message)是描述事件发生的信息，事件(Event)由多个消息组成。消息是对象之间发出的行为请求。封装使对象成为一个相对独立的实体，而消息机制为它们提供了一个相互间动态联系的途径，使它们的行为能互相配合，构成一个有机的运行系统。

对象通过对外提供的行为在系统中发挥自己的作用，当系统中的其他对象请求这个对象执行某个行为时，就向这个对象发送一个消息，这个对象就响应这个请求，完成指定的行为。程序的执行取决于事件发生的顺序，由顺序产生的消息驱动，不必预先确定消息产生的顺序，更符合客观世界的实际。

面向对象技术强调在软件开发过程中，面向客观世界或问题域中的事物，采用人类在认识客观世界的过程中普遍运用的思维方法，直观、自然地描述客观世界中的有关事物。面向对象技术的基本特征主要有抽象性、封装性、继承性和多态性。

(2)事件驱动技术事件是对象发送的消息,通过发信号通知操作的发生。操作可能是用户交互引起的,也可能是由某些其他的程序代码逻辑触发的。要在应用程序中使用事件,就必须提供一个事件处理程序,该处理程序执行程序以响应事件。

ASP.NET 就可以实现处理捕获、传送和解释事件的所有技术细节。因此,当在 Web 窗体页中创建事件处理程序时,无需考虑如何捕获事件信息以及如何使事件信息在代码中可用等这些技术细节。因为 ASP.NET 的可视化开发工具简化了事件处理的细节。

(3)编译语言的支持 ASP.NET 不再支持任何的解释程序语言,它提供三种内建的程序语言:C#, VB 和 Jscript,其中 C#是 C/C++的自然演进,适合原来使用 C/C++, Java 和 Javascript 的程序员;熟悉 Vbscript 的人则可以使用 VB;Jscript 适合于喜欢 Javascript 的爱好者。

(4) Codebehind 技术 ASP.NET 采用结构化的网页,为了很好地将逻辑代码和表现代码分开,引用 Codebehind 技术,通过使用这项技术,可以使逻辑代码和表现代码两者互不影响。在一般的 ASP.NET 文件中, Code Behind 主要是用二个文件来创建一个 ASP.NET 的页面,其中一个设计文件,一般以.aspx 或者.ascx 做为扩展名,而另外一个程序代码文件,一般以.vb 或者.cs 做为扩展名,其程序设计语言主要是 VB. Net 或者是 C#。采用 Codebehind 技术编写程序代码,至少需要两个文件:.ASPX 文件和.ASPX. CS 文件。在.ASPX 文件中编写表现代码和而在.ASPX. CS 文件中编写逻辑代码。这种技术需要程序员做的工作大大减少,而且可以方便多个程序员分工合作。而 ASP 程序是把界面设计和程序设计混合在一起的。

因此当程序设计人员要修改应用程序界面布局的时候,就往往需要更改大量与界面无关的代码,对于一个小程序,工作量不是很大,如果对于代码量很大的程序,就是一项工作量不小的事情。而 Code Behind 把界面设计代码和程序设计代码以不同的文件分开,对于代码的重复使用,程序的调试和维护都是革命性的。还有一点是,当你发布你的网站的时候,你就可以利用这种技术来有效的保护你的代码。这对于程序的安全性是一个提高。

(5)数据绑定技术数据存取是 Web 应用程序的一项重要功能,数据绑定语法允许开发人员不仅可以绑定到数据源(Data Source),而且可以绑定到简单属性、集合、表达式甚至是从方法调用返回的结果。DataBind 是一种全新的技术,通过他可以将程序中的执行数据与页面中的属性、集合、表达式以及函数结果“绑定”在一起。也就是说,只要程序中相应的数据发生变化,那么这些与数据“绑定”在一起的属性、集合、表达式以及函数结果的值就会随之改变。

DataBind 包括三大方法, **Repeater**, **DataList** 和 **DataGrid**, 这些控件都位于 **System. Web. UI. WebControl** 命名空间中, 从 **WebControl** 基类中直接或间接派生出来的。这些方法都是通过 **HTML** 来显示数据的内容, 所有的 **DataBind** 都应该用 **DataBind()** 函数来建立数据绑定, 是整个页面 **Page** 和所有控件的一个方法, 也就是说, 它可以被所有的控件使用, 当建立数据绑定的时候, **DataBind** 可以作为控件的一个子项, 比如 **DataList1.DataBind()**, 再比如 **Page.DataBind()**, 会绑定整个页面。**DataBind** 常在页面载入时就被绑定。一般情况下, **Bound** 用来控制数据命令和读出, **HyperLink** 让数据以超链的形式显示, **Button** 建立动态的数据按钮, **Template** 样本模板方式输出数据。

2.3 数据库访问技术:ADO. NET

ADO. NET 是一种新型的数据存取机制, 它是在用于直接满足用户开发可伸缩应用程序需求的 **ADO** 数据访问模型的基础上发展而来的。它是专门为 **Web** 设计的, 并且考虑了伸缩性, 无状态性和 **XML** 的问题。

2.3.1 ADO. NET 和 ADO 的比较

(1) 二者最主要的不同在于 **ADO. NET** 将焦点集中在脱机的数据, 也就是说, 当 **ADO. NET** 从数据库获取部分的数据集后, 便切断与数据库系统之间的连接, 变成脱机状态。

(2) 常驻内容数据表示形式不同。**ADO** 使用 **Recordset** 对象, 该对象类似一个单独的表; **ADO.NET** 使用 **DataSet** 对象, 该对象包含由 **DataTable** 对象表示的一个或多个表。

(3) 为了访问各种不同的数据库系统或不同的数据源, **ADO.NET** 提供了若干个命名空间供 **ASP.NET** 使用; 编程时, 只需要在程序代码中把这些命名空间包含进来, 以声明正确的数据类型, 而非像 **ASP** 网页是通过 **Server** 对象的 **Createobject()** 方法创建。

2.3.2 ADO. NET 组件

ADO.NET 组件的两个核心组件是: **.NET** 数据提供程序和 **DataSet**。如图 2-3 所示。

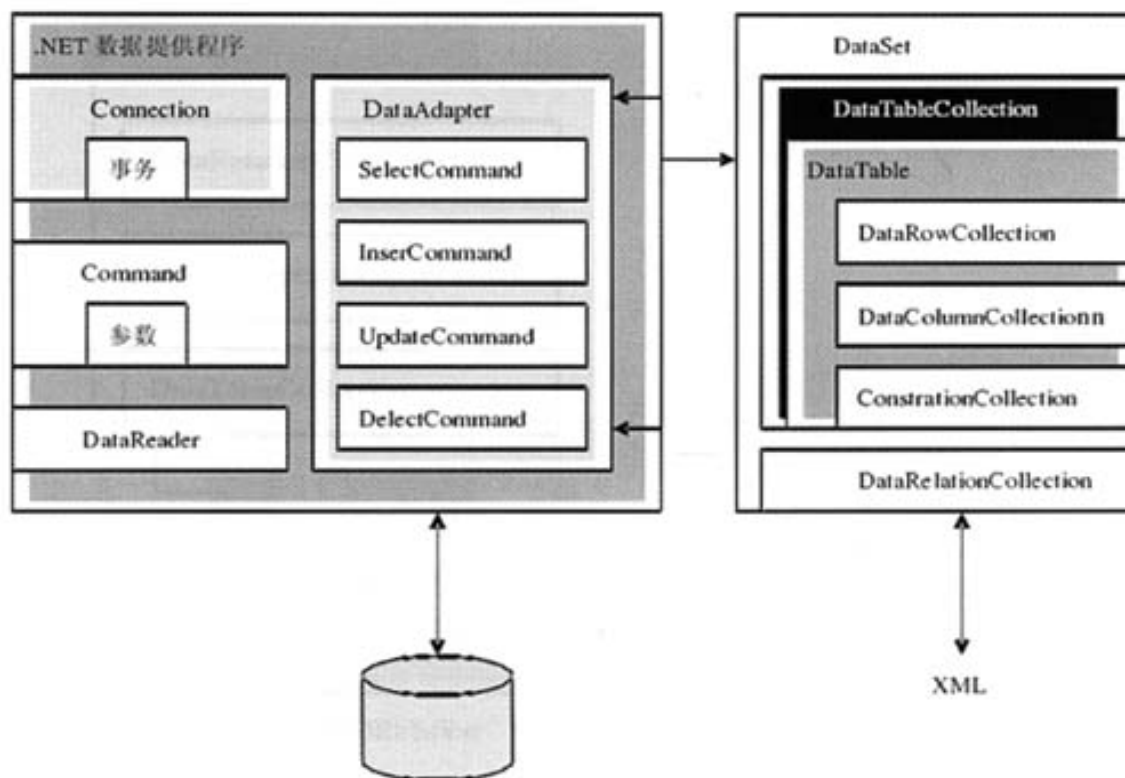


图 2-3 ADO.NET 组件

(1) .NET 数据提供程序使管理员能够连接到数据源，以便检索和修改数据源中的数据。它附带了两个.NET 数据提供程序：SQL Server.NET 和 OLE DB.NET。在.NET 框架下的两种数据提供程序具有功能相同的对象，但这些对象的名称、部分属性或者方法可能不同。SQLServer 数据提供程序的对象名称以 SQL 为前缀；而 OLE DB 对象名称以 OLEDB 为前缀。提供程序具有的对象主要包括 Command、Connection、DataReader 和 DataAdapter 等。Connection 对象提供与数据源的连接；Command 对象使用户能够访问用于返回数据、修改数据、运行存储过程以及发送或检索参数信息的数据库命令；DataReader 从数据源中提供高性能的数据流；DataAdapter 使用 Command 对象在数据源中执行 SQL 命令，以便将数据加载到 DataSet 中，并使对 DataSet 中数据的更改与数据源保持一致。

(2) .DataSet 对象与 ADO RecordSet 对象相似，但功能更强大，并具有重要区别：DataSet 始终是断开的。DataSet 对象表示数据的缓存，具有类似数据库的结构，如表、列、关系和约束。但不直接与数据或其他数据源进行交互。这使得开发人员能够使用始终保持一致的编程模型，而不用理会数据源的驻留位置。DataSet 对象模型如图 2-4 所示。

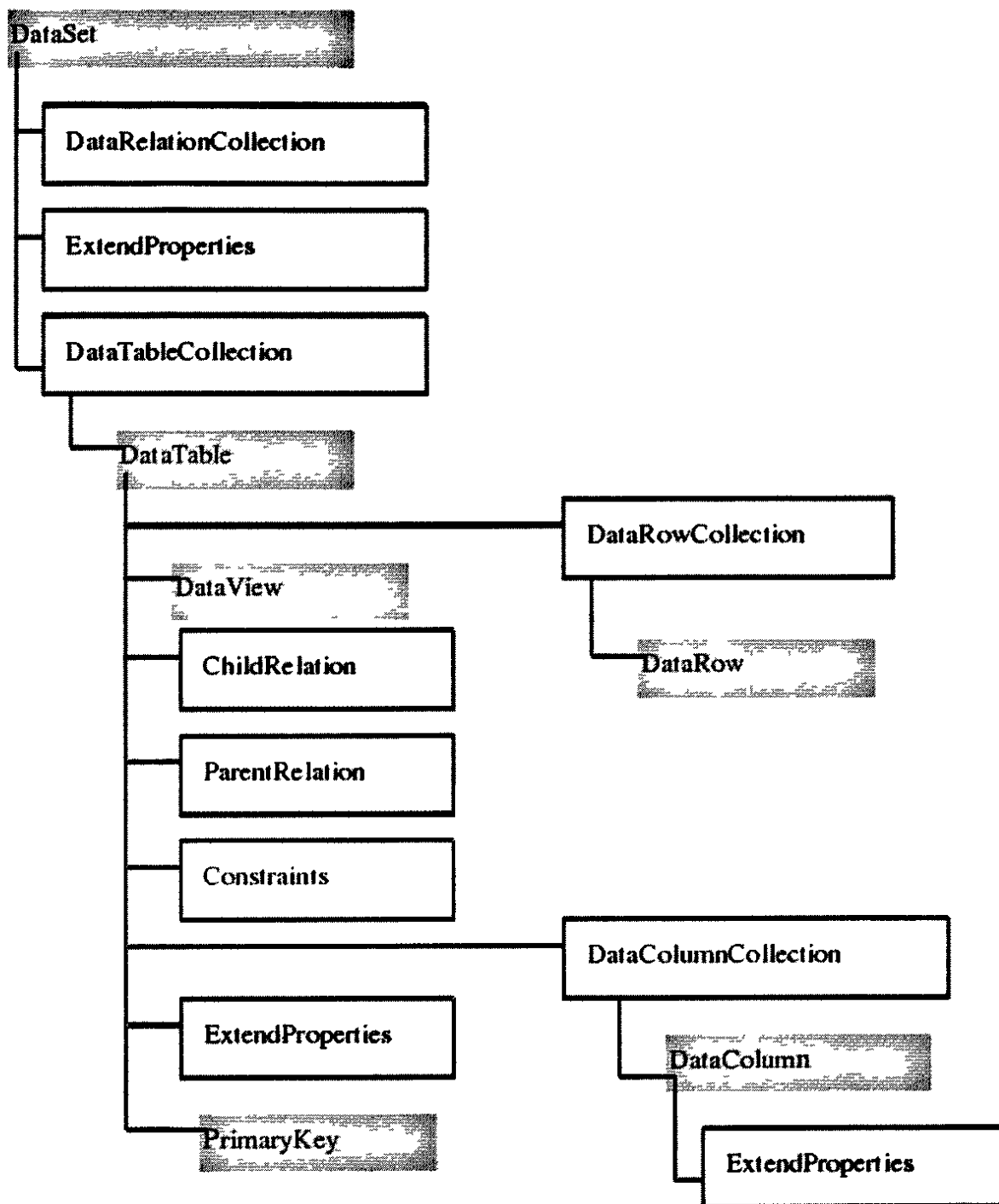


图 2—4 DataSet 对象模型

第3章 系统分析

3.1 需求分析

需求分析是系统设计的最重要环节，需求分析的结果是系统开发的基础，关系到系统设计、实现的成败和质量。

通过对企业的实地考察，与用户交流、调研，并分析了企业现有的工作模式，参考其它先进的销售管理系统，从系统的角度得出销售管理系统有如下的功能需求：

- (1) 系统要采用人机对话方式，界面美观良好，信息查询灵活、方便、快捷、准确，数据存储安全可靠。
- (2) 如果系统的使用对象较多，则要求有较好的权限管理。
- (3) 商品的订单、出入库、退货、销售流程清晰。
- (4) 在各种单据中根据输入的数据自动完成计算，提高工作效率。
- (5) 方便的商品统计查询功能，提供多条件统计查询。
- (6) 对用户输入的数据，系统进行严格的数据检验，尽可能排除人为的错误。
- (7) 系统最大限度地实现易维护性和易操作性。
- (8) 系统运行稳定、安全可靠。

3.2 可行性分析

可行性分析是明确应用项目开发的必要性和可行性。必要性来自实现开发任务的迫切性，这一点在分析企业销售管理的现状时已经论述过；而可行性则取决于实现应用系统的资源和条件。在研究与此类似的系统后，具体从以下三个方面考虑：

(1) 管理上的可行性

在当前的发展环境下，企业信息化建设已是迫在眉睫。企业的各层管理人员都已认识到这一点，因此他们对于企业应用系统是持积极态度的，能够对此提供有力支持。而一般的中小型企业和管理方面也具备了上信息系统的条件，这也是企业管理发展到今天本身所产生的要求。

(2) 技术上的可行性

由于国内外对企业销售管理系统进行开发已有一定的时期，有很多成功的实

例，技术基础也非常雄厚，因此本系统借鉴成功的实例和技术，从中小企业的实际出发，可有效地管理企业的商品销售情况，并降低企业运营成本、及时调整营销策略、提高库房的利用率。

（3）经济上的可行性

当前许多中小型企业都使用人工管理方式（即纸和笔）来管理企业的商品销售，这样的管理方式既困难又浪费时间和成本，并且容易出现漏账、差账的情况，因此中小企业应该向大型企业那样采用先进的管理方式，提高企业效率、降低企业运营成本。

对于企业而言，尤其是处于激烈竞争中的中小型企业而言，这些都是极其重要的因素。因此，开发一套高效的信息管理系统，是非常必要的。

第 4 章 系统设计

4.1 系统体系结构

4.1.1 三层体系结构

B/S 模式(Browser/Server, 浏览器/服务器, 简称 B/S)的多层结构的应用是在对 C/S 结构的总结基础上产生的, 它将应用划分为 3 层: 用户界面层、商业逻辑层、数据库层。B/S 三层体系结构如图 4-1 所示。

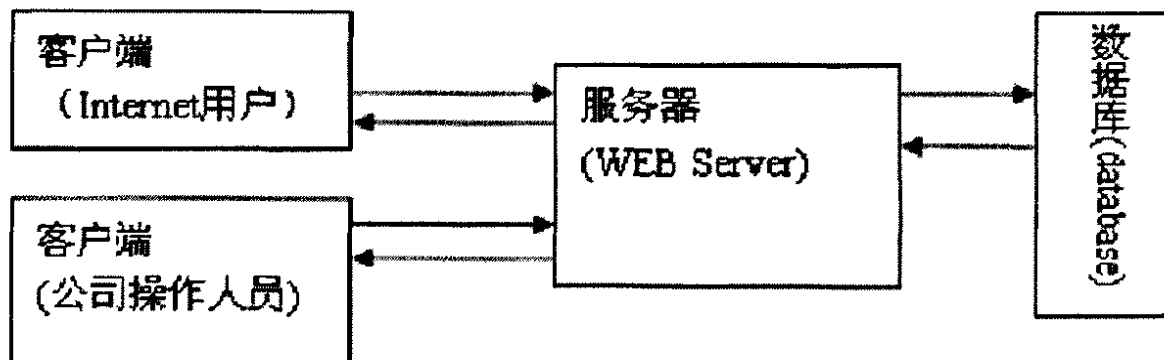


图 4-1 B/S 三层体系结构

用户界面层负责处理用户的输入和向用户的输出, 但并不负责解释其含义(出于效率的考虑, 它可能在向上传输用户输入前进行合法性验证), 这一层是采用前端工具(C#, VB(.NET), ASP(.NET), 等)开发;

商业逻辑层是上下两层的纽带, 它建立实际的数据库连接, 根据用户的请求生成 SQL 语句检索或更新数据库, 并把结果返回给客户端, 这一层通常以动态链接库 (DLL) 的形式存在并注册到服务器的注册簿(Registry)中, 它与客户端通讯的接口符合某一特定的组件标准(如 COM, CORBA), 可以用任何支持这种标准的工具开发;

数据库层负责实际的数据存储和检索^[4]。

三层体系结构的主要特征是将应用逻辑(对应应用层)从软件系统中分离出来形成了一个单独的逻辑中间层。表示层相对来说基本不处理应用业务过程, 窗口将任务请求转发给中间层。中间层与后端的存储层之间还要进行通信。

这种三层体系结构与两层的设计形成了对比, 在两层设计中, 例如, 应用逻辑是在窗口定义中安置的, 应用逻辑直接读写数据库, 应用逻辑没有被作为中间层而被分离出来。两层结构的一个缺点是用分离的构件来表达应用逻辑的能力不

足,从而限制了软件重用。两层结构设计不可能将应用逻辑分配到分开的机器中去。

对于逻辑上的三层体系结构可以进行各种各样的物理部署,这些部署主要包括下列情况:一是表示层和应用逻辑层位于客户端机器上,存储层位于服务器端机器上;二是表示层在客户端机器上,应用逻辑层在应用服务器上并且存储层在另一个单独的数据服务器上^[5]。

4.1.2 C/S 与 B/S 模式的比较

传统的信息管理系统的核心是 C/S 结构, C/S 结构的系统如图 4-2 所示。

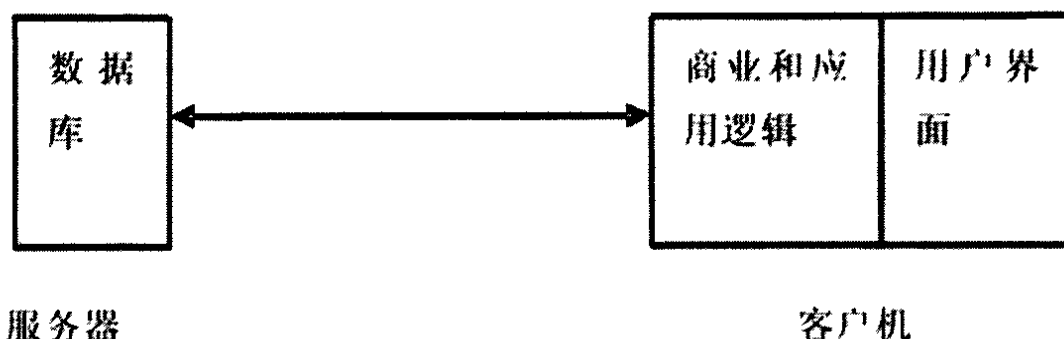


图 4-2 C/S 结构的系统

而由于本系统是基于 Internet 的所以核心是 B/S 模式。B/S 构架比传统的 C/S 结构有更大的优越性。首先它简化了客户端。它无需像 C/S 模式那样在不同的客户机上安装不同的客户应用程序,而只需安装通用的浏览器软件。这样不但可以节省客户的硬盘空间与内存,而且使安装过程更加简便、网络结构更加灵活。

其次,它简化了系统的开发和维护。系统的开发者无须再为不同级别的用户设计开发不同的客户应用程序了,只需把所有的功能都实现在 Web 服务器上,并就不同的功能为各个组别的用户设置权限就可以了。各个用户通过 HTTP 请求在权限范围内调用 Web 服务器上不同处理程序,从而完成对数据的查询或修改。面对着日新月异的竞争环境,不同企业内部的销售和管理也各自不同。相对于 C/S, B/S 的维护具有更大的灵活性。当形势变化时,它无须再为每一个现有的客户应用程序升级,而只需对 Web 服务器上的服务处理程序进行修订。这样不但可以提高商务的运作效率,还省去了维护时协调工作的不少麻烦。如果一个公司有上千个客户,并且分布在不同的地点,那么便于维护将会显得更加重要。

再次,它使用户的操作变得更简单。对于 C/S 模式,客户应用程序有自己特定的规格,使用者需要接受专门培训。而采用 B/S 模式时,客户端只是一个简单易用的浏览器软件。无论是用户层还是操作层的人员都无需培训,就可以直接使

用。B/S 模式的这种特性，还使系统维护的限制因素减少。

最后，B/S 模式通过 internet 进行通信，特别适用于网上信息发布，比传统的 C/S 模式有大的扩展。而这种新增的网上信息发布功能恰是现代企业所需的。这使得商业中的大部分新的产品和价格等可以用电子文件取代，从而提高了企业的工作效率，使企业行政手续简化，节省人力物力。

4.1.3 本系统的体系结构

通过上面的分析，本系统采用 B/S 模式三层结构来设计系统。采用 ASP.NET 来构建前端和中间层，SQL Server 2000 来管理后端数据库。

由于本系统访问数据库比较频繁，因此使用了一个数据库工程（SQLHelper）单独实现访问数据库的功能，这样，系统就可以基于该数据工程来访问数据库，提高了系统的效率，避免在开发过程中直接使用 SQL 语句，而是通过存储过程来访问数据库。如图 4-3 所示。

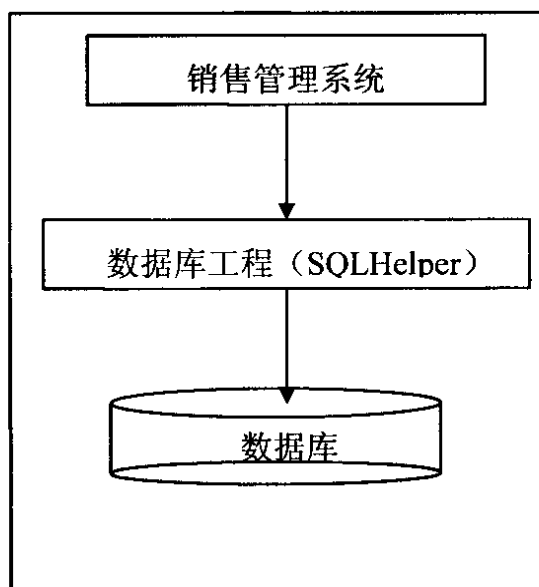


图 4-3 系统设计 三层结构

4.2 系统的组成

4.2.1 系统功能结构图

本系统的目的就是原来由手工操作对销售信息的管理用计算机进行全程管理，这减少了手工操作出错的可能，确保了数据的正确性和完整性，还大大提高

企业的工作效率，减少了人力成本的付出，使得销售管理实现信息化。下面就是本系统的功能结构图，如图 4-4 所示。

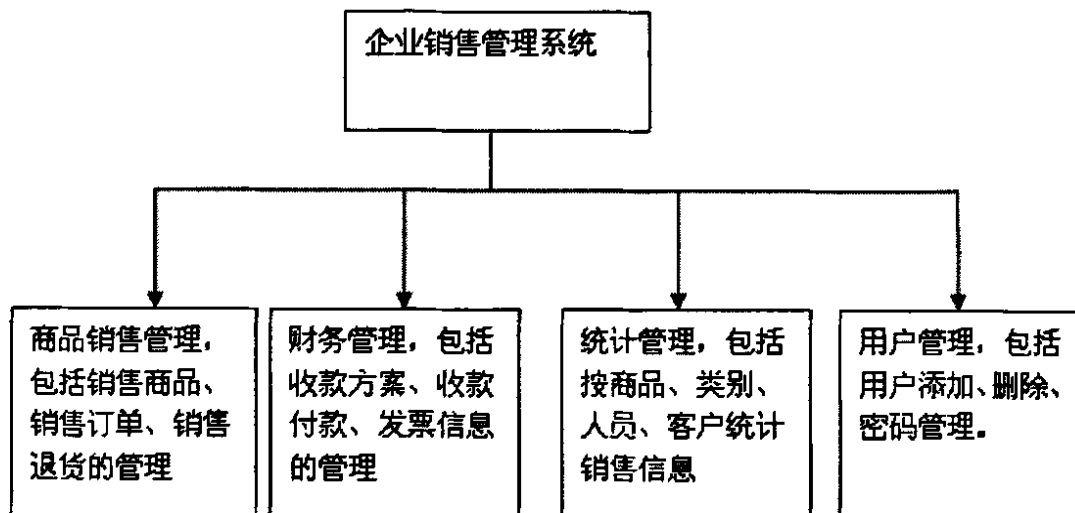


图 4-4 系统 功能结构

4.2.2 系统功能模块划分

企业销售管理系统包括商品销售管理、财务管理、统计管理和用户管理四个模块。各模块主要功能：

- (1) 商品销售管理模块主要完成商品销售、销售订单及销售退货信息管理。
- (2) 财务管理模块主要完成收款方案、收款付款及发票信息管理。
- (3) 统计管理模块主要完成按商品名称、商品种类、销售人员及客户这四种条件统计商品销售信息。
- (4) 用户管理模块主要完成用户的添加、删除和密码修改。

4.2.3 系统设计架构

系统设计架构采用三层结构开发模式，如图 4-5 所示。

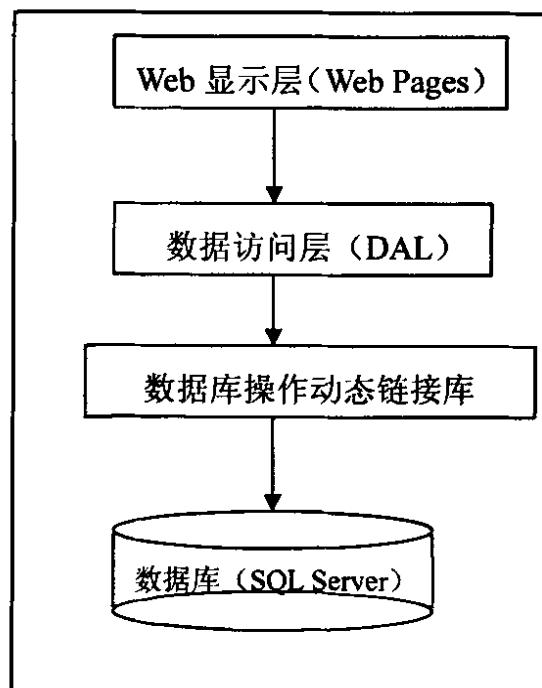


图 4-5 系统设计架构

(1) Web 显示层即为 ASP.NET 页面 (Web Pages) 层，为客户或用户提供对应用程序的访问，本系统 Web 显示层以 Web 页面的形式实现，如用户的登录页面、用户管理界面等。该层通过数据访问层访问数据库。

(2) 数据访问层为 Web 显示层提供数据服务，一般封装操作数据库的选择、添加、更新和删除等操作，同时还为 Web 显示层提供访问数据库的接口或函数等。

本系统数据访问层的实现原则是根据数据库的表来创建相应的类，如在数据库中存在用户信息表 User，因此系统中创建类 User，并用文件 User.cs 保存。类 User (文件 User.cs) 封装系统中的用户信息的添加、选择、更新、删除和登录等操作。

(3) 数据库操作动态链接库 (DLL)，该层直接访问 ASP.NET Mis 系统的数据库，它由工程 SQLHelper 实现。

(4) 数据库是系统最底层，它存储系统的所有数据。

4.3 数据库访问方案

数据库访问方案是企业销售管理系统用来存储、检索和管理数据库中数据的方法，本系统采用以下方案：系统中对数据库的访问全部采用存储过程实现，使用存储过程的好处是可以将多个 SQL 语句打包在一起作为一个组来执行，如果在应用程序中恰当地使用存储过程可以提高应用程序的效率。

4.4 开发及运行环境

硬件平台：

CPU: P4 1.8GHz。

内存: 256MB 以上。

软件平台：

操作系统: Windows XP 操作系统。

数据库: Microsoft SQL Server 2000。

技术平台: Microsoft.NET Framework 1.1.43220.0 版本

IIS: Internet Information Server 5.1 版本

开发工具: Microsoft Visual Studio.NET 2003 中文企业版。

浏览器: IE6.0。

分辨率: 最佳效果 1024×768 像素。

第 5 章 数据库设计

5.1 Microsoft SQL Server 2000 简介

SQL Server 2000 是 Microsoft 公司推出的 SQL Server 数据库管理系统的最新版本。该版本继承了 SQL Server 7.0 版本的优点同时又比它增加了许多更先进的功能，具有使用方便可伸缩性好与相关软件集成程度高等优点。

由于今天 Windows 操作系统占领着主导地的位，选择 SQL Server 一定會在兼容性方面取得一些优势。另外，SQL Server 2000 除了具有扩展性，可靠性以外，还具有可以迅速开发新的因特网系统的功能。尤其是它可以直接存贮 XML 数据，可以将搜索结果以 XML 格式输出等特点，有利于构建了异构系统的互操作性，奠定了面向互联网的企业应用和服务的基石。这些特点在 .NET 战略中发挥着重要的作用。

本系统采用 Microsoft SQL Server 2000，开发设计容易上手，维护成本较低。

5.2 数据库设计

一个合理的数据库模型，不仅会降低客户端和服务端编程和维护的难度，而且将会增强系统实际运行的性能。一般来讲，在一个系统分析、设计、测试和试运行阶段，因为数据量较小，设计人员和测试人员往往只注意到功能的实现，很难注意到性能的薄弱之处，等到系统投入实际运行的一段时间后，才发现系统的性能在降低，这时再来考虑提高系统性能则需要花费更多的人力物力，而整个系统也就形成了一个不断打补丁工程。因此，设计数据库时要注意以下几个方面：

命名的规范：不同的数据库产品对对象的命名有不同的要求，因此，数据库中的各种对象的命名、程序的代码编写应采用大小写敏感的形式，命名长度最好不要超过 30 个字符，这样便于适应不同的数据库。

索引的使用：创建索引一般有两个目的：维护被索引列的唯一性和提供快速访问表中数据的策略。大型数据库有簇索引和非簇索引两种索引类型，一个没有簇索引的表是按堆结构存储的，所有的数据均添加在表的尾部，而建立了簇索引的表，数据在物理上会按簇索引键的顺序存储，一个表只允许有一个簇索引，因此，根据 B 树结构，可以理解为添加任何一种索引均能提高按索引列查询数据的速度，但会降低插入、更新、删除操作的性能，所以对于索引较多的表进行频繁

的插入、更新、删除操作，建立表和索引时应尽量设置较小的充填因子，以便在各数据页中留下较多的自由空间，减少页分割及重组的工作。

数据的一致性和完整性：为了保证数据的一致性和完整性，往往设计过多的表间关联，尽可能的降低数据的冗余。表间关联是一种强制性措施，建立后，对父表和子表的插入、修改、删除等操作均要占用系统的开销，另外，最好不要用 Identify 属性字段作为主键与子表关联。如果数据冗余低，数据的完整性容易等到保证，但也增加了表间连接查询的操作，为了提高系统的响应时间，合理的数据冗余也是必要的。使用规则和约束来防止操作人员误输入造成数据的错误是设计人员常用的一种手段，但是，不必要的规则和约束也会占用系统的不必要开销，需要注意的是，约束对数据的有效性验证要比规则快。所有这些，在设计阶段应根据系统操作的类型、频度加以均衡考虑。

数据类型的选择：数据类型的合理选择对于数据库的性能和操作具有很大的影响，Identify 字段不要作为表的主键与其它表关联，这将会影响到该表的数据迁移。Text 和 Image 字段属指针型数据，这类数据的操作相比其它数据类型较慢，因此要尽量避免使用。日期型字段的优点是有众多的日期函数支持，因此，在日期的大小比较、加减操作上非常简单。但是，在按照日期作为条件的查询操作也要用函数，相比其它数据类型速度上就慢许多。系统设计过程中出现了就出现了数据库设计类型与程序类型不匹配的问题，对于程序调试带来了很大麻烦，因此在数据库类型选择和程序设计中一定要匹配。

另外，还要注意在需要提高性能的地方，多试几种 SQL 写法，不同的 SQL 写法往往功能相同，但是效率却存在差异。

5.3 数据库表结构

根据系统的功能模块分析设计，建立如下数据库表：

(1) 销售订单表 SaleOrder。表结构如图 5-1 所示

编号	字段名称	数据类型	说明
1	SaleOrderID	int	销售订单编号
2	EmplayeeID	int	销售负责人编号
3	ContactID	int	客户负责人编号
4	Status	int	订单状态（创建、审核、出库、完成）
5	SignDate	datetime	合同签订日期
6	ProSum	money	商品金额

7	OtherSum	money	其他费用
8	TotalSum	money	总费用
9	PubDate	datetime	创建日期
10	PubName	varchar(200)	创建人
11	Remark	text	备注

图 5-1 SaleOrder 表结构

(2) 销售商品表 OrderProduct。表结构如图 5-2 所示。

编号	字段名称	数据类型	说明
1	OrderProductID	int	销售商品编号
2	ProductID	int	商品编号
3	SaleOrderID	int	销售订单编号
4	Quantity	int	销售数量
5	Price	money	销售价格
6	Flag	int	订单和退货标志, 0 表示订单, 1 表示退货
7	IsCheck	int	是否审核, 0 表示未审核, 1 表示已审核
8	Remark	text	备注

图 5-2 OrderProduct 表结构

(3) 销售退货表 SaleReturn。表结构如图 5-3 所示

编号	字段名称	数据类型	说明
1	SaleReturnID	int	销售退货表编号
2	SaleOrderID	int	销售订单编号
3	ProSum	money	商品金额
4	OtherSum	money	其他费用
5	TotalSum	money	总费用
6	Status	int	状态 (创建、审核、入库、完成)
7	ReturnDate	datetime	退货日期
8	PubDate	datetime	创建日期
9	PubName	varchar(200)	创建人
10	Flag	Int	标志, 0 表示未退货, 1 表示已退货
11	Remark	int	备注

图 5-3 SaleReturn 表结构

(4) 收款方案表 GatheringMethod。表结构如图 5-4 所示

编号	字段名称	数据类型	说明
1	GatheringMethodID	int	报价编号
2	SaleOrderID	int	销售订单编号
3	TypeID	int	收款类型编号
4	TypeFlag	int	收款阶段编号
5	PubDate	datetime	收款日期
6	GatheringSum	money	收款金额
7	Remark	text	备注

图 5-4 GatheringMethod 表结构

(5) 收款付款表 Exchange。表结构如图 5-5 所示

编号	字段名称	数据类型	说明
1	ExchangeID	int	收款付款编号
2	CustomerID	int	收款付款客户编号
3	ExchangeType	int	收款付款类型
4	Status	int	状态
5	IsInvoice	int	是否开发票
6	ExchangeSum	money	收款付款金额
7	ExchangeDate	datetime	收款付款日期
8	Remark	text	备注

图 5-5 Exchange 表结构

(6) 发票信息表 Invoice。表结构如图 5-6 所示

编号	字段名称	数据类型	说明
1	InvoiceID	int	发票信息编号
2	CustomerID	int	收付款客户编号
3	Title	varchar(200)	发票标题 (抬头)
4	TypeID	int	发票类型编号
5	InvoiceSum	money	发票金额
6	InvoiceDate	datetime	发票日期
7	Flag	int	状态 (0 表示创建, 1 表示审核)
8	Remark	text	备注

图 5-6 Invoice 表结构

(7) 商品信息表 Product。表结构如图 5-7 所示

编号	字段名称	数据类型	说明
1	ProductID	int	商品 ID
2	ProNo	varchar(200)	商品编号
3	ProName	varchar(200)	商品名称
4	ProductTypeID	int	商品所属种类
5	ProStyle	varchar(200)	商品规格
6	ProUnit	varchar(200)	商品计量单位
7	Price	money	价格
8	Quantity	int	数量
9	LowerSum	int	最小数量
10	UpperSum	int	最大数量
11	Remark	text	备注

图 5-7 Product 表结构

(8) 商品种类表 ProductType。表结构如图 5-8 所示

编号	字段名称	数据类型	说明
1	ProductTypeID	int	商品种类编号
2	Name	int	商品种类名称
3	ParentID	int	上级种类编号
4	Remark	text	备注

图 5-8 ProductType 表结构

(9) 类型表 Type。表结构如图 5-9 所示

编号	字段名称	数据类型	说明
1	TypeID	int	类型编号
2	Name	varchar(200)	类型描述
3	Flag	int	类型标志

图 5-9 Type 表结构

5.4 数据库关系图

数据库 ASP.NET DB 销售信息系统模块的各表间关系结构如图 5-10 所示

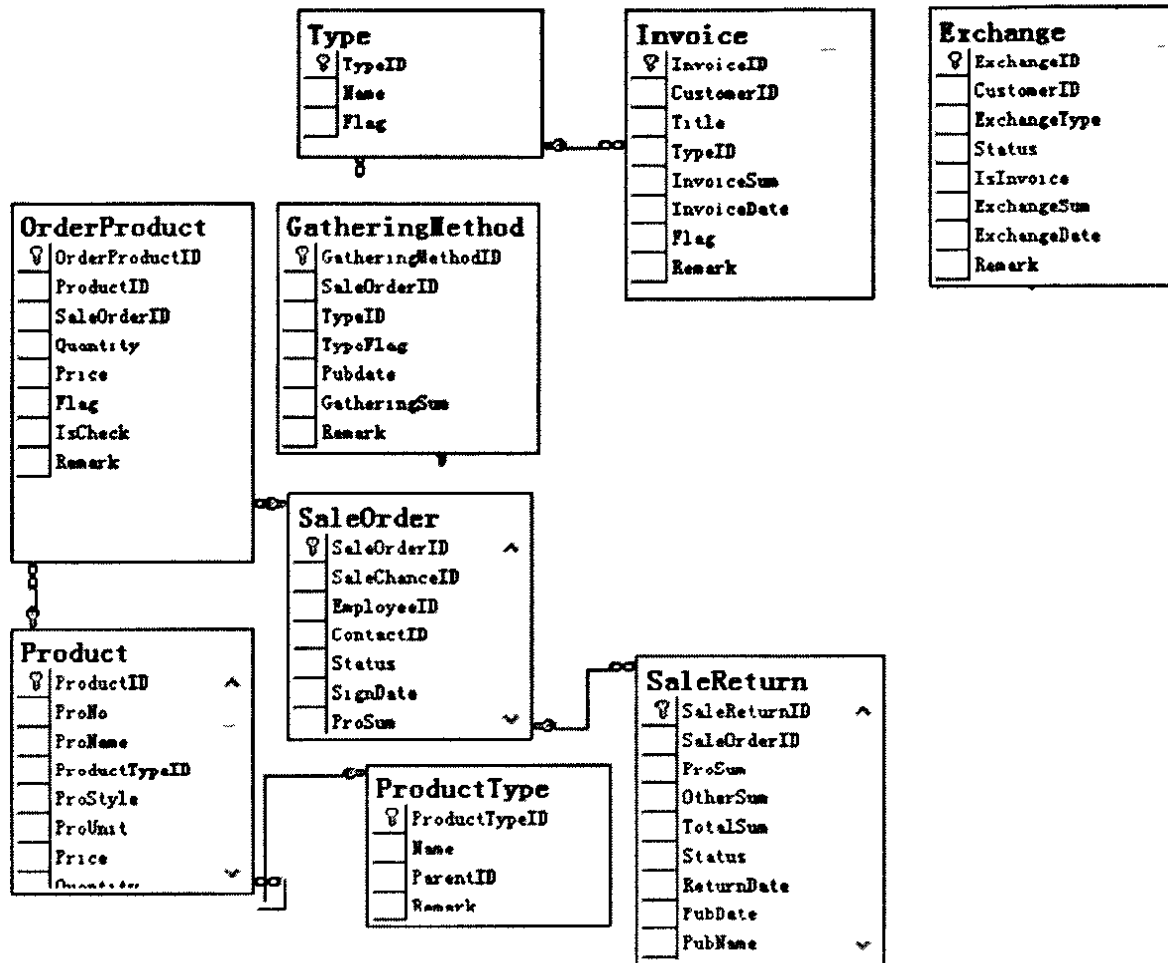


图 5-10 数据库关系图

第 6 章 销售管理系统的实现

6.1 系统登录

用户通过登录页面，进入企业销售管理系统首页。该模块主要用于验证用户的身份及密码，其页面如图 6-1 所示。

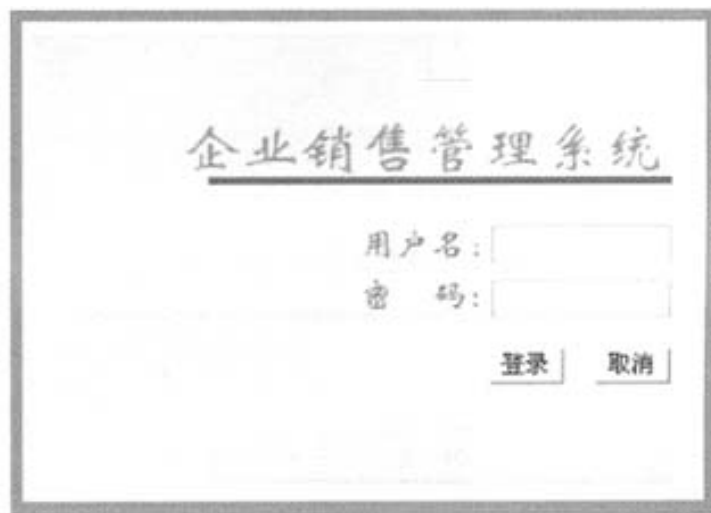


图 6-1 登录页面

输入用户名和密码点击登录按钮，在服务器端响应该按钮的 click 事件，在该事件的处理函数 LoginBtn_Click 中调用数据访问层 User 类的 GetUserLogin 方法，通过执行存储过程 Pr_GetUserLogin，核实用户的信息，最终完成判断用户合法性，合法用户可以正常登录，进入销售管理系统首页，如图 6-2 所示。若不是合法用户，系统会提示出错，如图 6-3 所示。点击取消按钮，重新返回到登录页面。

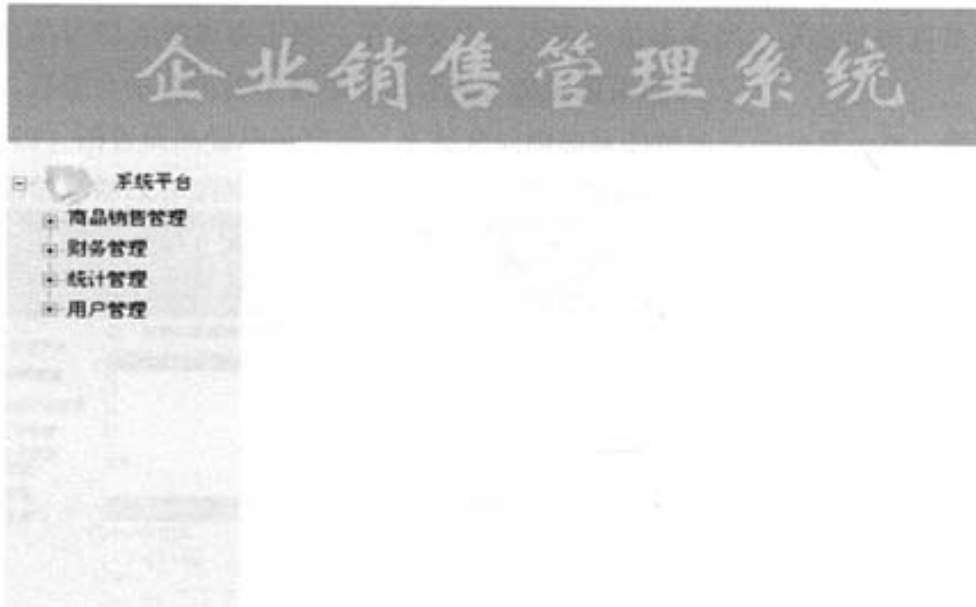


图 6-2 销售管理系统首页

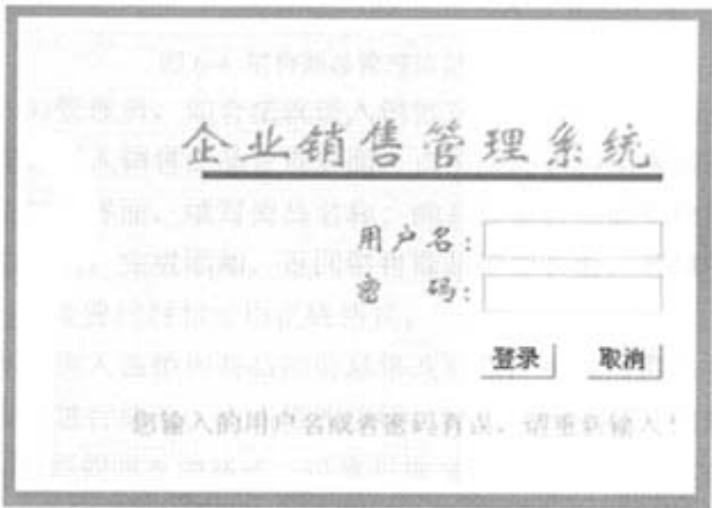


图 6-3 用户登录错误页面

6.2 商品销售管理

商品销售管理模块主要包括销售商品管理、销售订单管理和销售退货管理三个子模块。各子模块的具体功能是：

销售商品管理主要完成对销售商品的添加、修改和删除。

销售订单管理主要完成对销售订单的添加、修改和删除。

销售退货管理主要完成对销售退货的添加、修改和删除。

6.3.1 商品销售管理模块的实现

商品销售管理模块下的三个子模块总体设计思路相似，下面以销售商品管理为例，详细介绍该模块下销售商品管理子模块的实现。

(1) 销售商品管理子模块，主要完成添加新的销售商品信息、对已创建的销售商品信息修改和删除。销售商品管理页面如图 6-4 所示

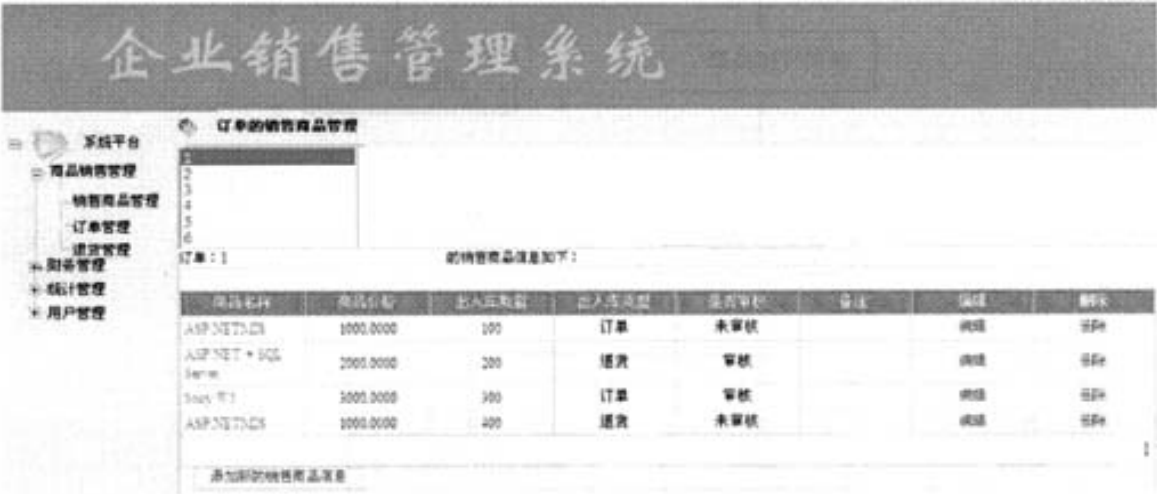


图 6-4 销售商品管理信息页面

首先判断是否为管理员，如合法就进入销售系统平台。点击商品销售管理，点击销售商品管理，进入销售商品管理页面，点击添加新的销售商品信息按钮，进入添加销售商品信息界面，填写商品名称、购买数量、商品单价等信息，填写完成后，点击添加按钮，完成添加，返回销售商品管理页面，若填写信息出现错误，系统会提示出错位置同时显示出正确格式。

点击修改热键，进入该销售商品的信息修改页面，可以对商品名称、所属订单、购买数量等信息进行修改，点击修改按钮，完成编辑，返回销售商品管理页面，如出现错误，系统提示正确格式，可重新填写。

点击删除热键，进入删除页面，点击删除，确定删除，完成删除。

(2) 销售商品管理子模块流程如图 6-5 所示：

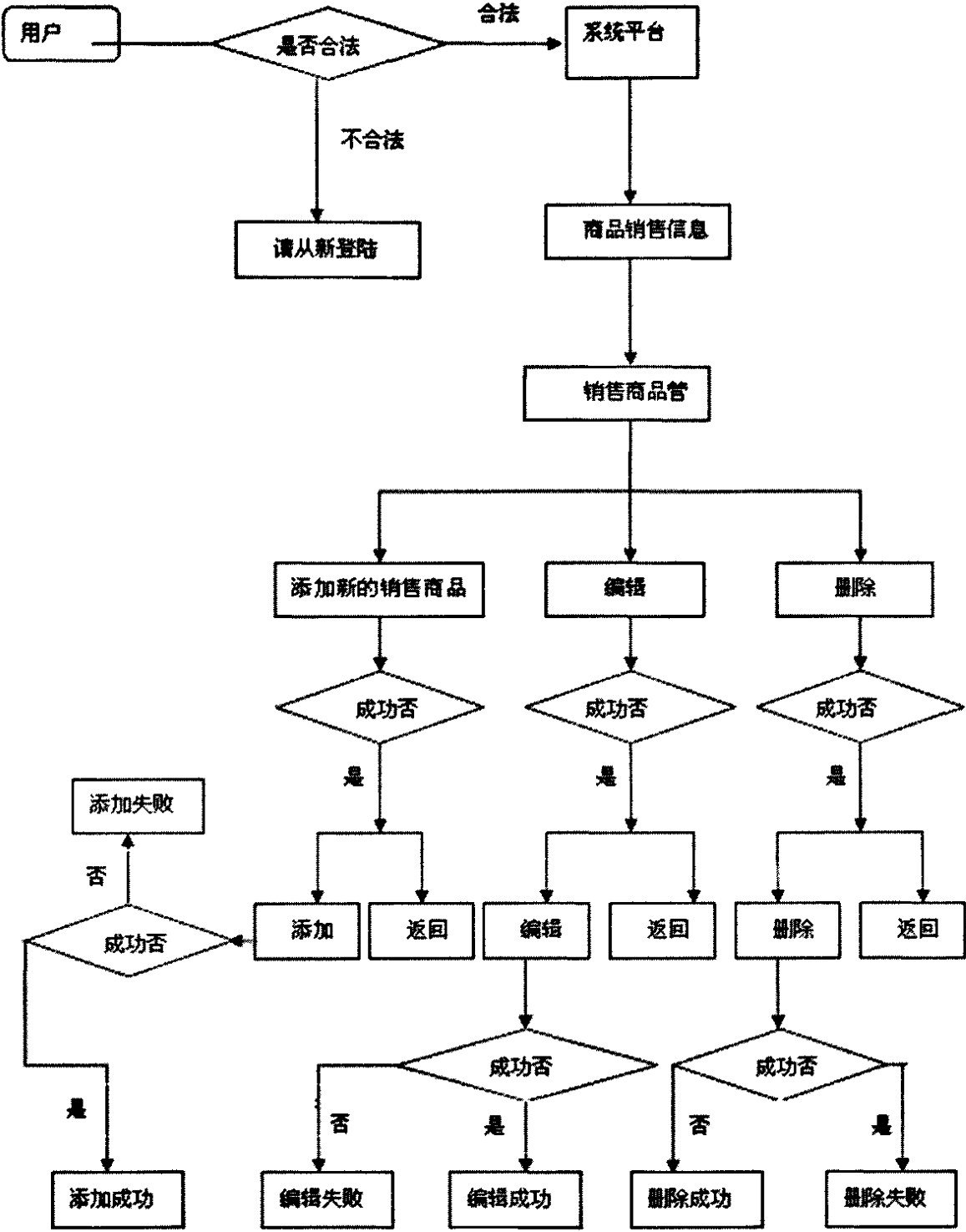


图 6-5 销售商品管理模块流程图

(3) 销售商品管理子模块实现如下：

点击添加新的销售商品信息按钮调用 AddBtn_Click 方法跳转到添加销售商品信息页面 AddOrderProduct.aspx，如图 6-6 所示。



图 6-6 添加销售商品信息页面

在输入完信息后，点击添加按钮，将页面回发到服务器，在服务器端，响应按钮的 click 事件，在该事件处理函数中，调用数据访问层 OrderProduct 类的 AddOrderProduct 方法，将用户输入的相关信息添加到数据库中。同时将操作的结果返回到客户端。成功添加弹出如下界面：如图 6-7 所示。

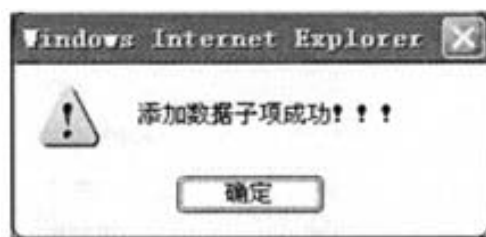


图 6-7 添加成功弹出页面

为了保证数据库数据的完整性，在添加数据时，对用户的输入进行验证，确保用户输入符合要求，如果添加的数据为空或错误，系统会给用户相应的提示，如图 6-8 所示。



图 6-8 系统提示出错页面

(4) 在销售商品信息表中，点击某一商品的编辑热键跳转到 UpdateOrderProduct.aspx 页面，对该商品信息进行修改，如图 6-9 所示



图 6-9 编辑该商品信息页面

当用户修改完该商品的信息后，点击修改按钮，将页面回发到服务器，在服务器端，通过 UpdateBtn_Click 函数，处理该按钮的 click 事件，在该事件处理函数中，调用 OrderProduct 类的 UpdateOrderProduct 方法，更新该销售商品的信息，同时将操作结果返回到客户端。成功编辑弹出如界面：如图 6-10 所示。

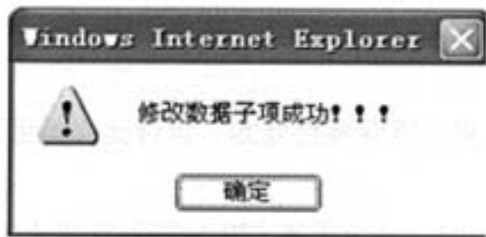


图 6-10 修改成功界面

(5) 在销售商品信息表中，点击某一商品的删除热键跳转到 DeleteOrderProduct.aspx 页面，进入删除该商品的状态，如图 6-11 所示。

删除销售商品信息	
商品名称:	ASP.NETMIS
所属订单:	1
购买数量:	100
商品单价:	1000.0000
是否退货标志:	订单
是否审核:	未审核
备注:	
<div>删除 返回</div>	

图 6-11 删除该商品页面

点击删除按钮，系统弹出确认删除信息对话框，点击确认后页面回发到服务器，在服务器端，通过 DeleteBtn_Click 函数，处理该按钮的 click 事件，在该事件处理函数中，调用 OrderProduct 类的 DeleteOrderProduct 方法，删除该商品信息，同时将操作结果返回到客户端。成功删除弹出如下页面：如图 6-12 所示。

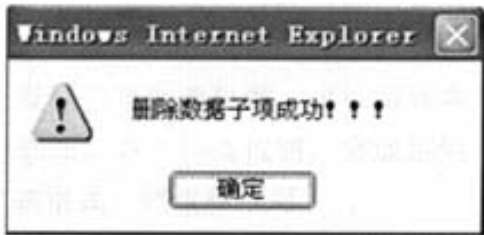


图 6-12 删除成功界面

6.3 财务管理

财务管理模块包括收款方案管理、收款付款管理、发票信息管理三个子模块。各子模块的具体功能是：

- (1) 收款方案管理主要完成对收款方案的添加、修改和删除。
- (2) 收款付款管理主要完成对收款付款的添加、修改和删除。
- (3) 发票信息管理主要完成对发票信息的添加、修改和删除。

6.3.1 财务管理模块的实现

财务管理模块下的三个子模块总体设计思路相似，下面以发票信息管理为例，详细介绍该模块下发票信息管理子模块的实现。

(1) 发票信息管理模块，主要完成添加新的发票信息、对已创建的发票信息编辑、删除功能。发票信息界面如图 6-13 所示



图 6-13 发票信息管理界面

首先判断用户是否为管理员，如合法进入销售系统平台。点击财务信息，点击发票管理，进入发票管理页面，点击添加新的发票信息按钮，进入添加新的发票信息页面，填写所属客户、发票标题、发票类型、发票金额等信息，填写完成后，点击添加按钮，完成添加，返回发票管理页面，若填写信息出现错误，系统会提示出错并显示出正确格式。

点击编辑热键，进入发票信息修改页面，可以所属客户、发票标题、发票类型、发票金额等信息进行修改，点击修改按钮，完成编辑，返回发票管理页面，如出现错误，系统提示正确格式，可重新填写。

点击删除热键，进入删除页面，点击删除，确定删除，完成删除。

(2) 发票信息管理模块流程图如图 6-14 所示：

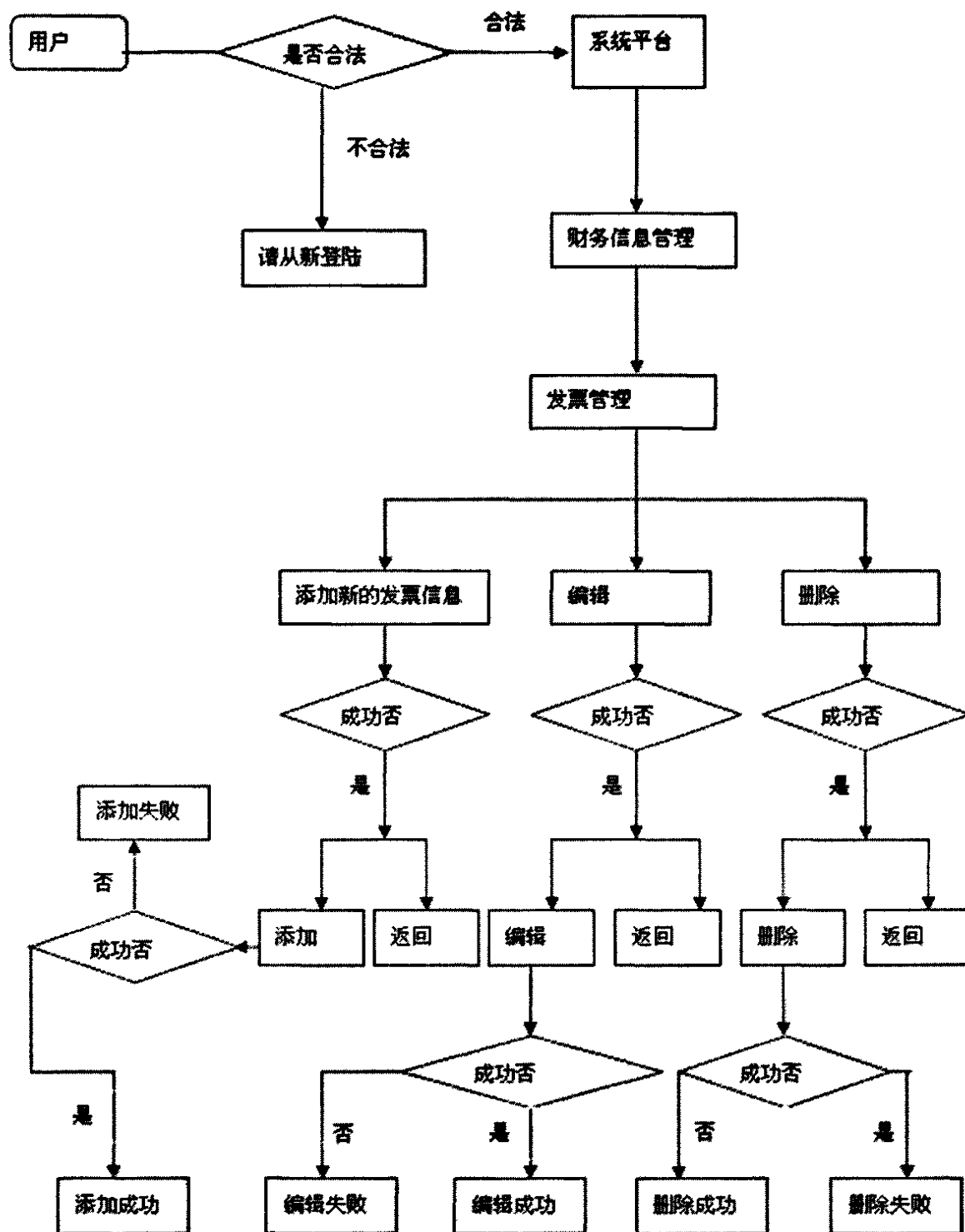


图 6-14 发票信息管理模块流程图

(3) 发票信息管理模块实现如下：

点击添加新的发票信息按钮调用 AddBtn_Click 方法跳转到添加发票信息页面 AddInvoice.aspx 如图 6-15 所示：

添加发票信息

所属客户: 000001

发票标题:

发票类型: 个人全额发票

发票金额:

开发票日期:

状态: 创建

备注:

添加 返回

图 6-15 添加发票信息页面

在输入完信息后，点击添加按钮，将页面回发到服务器，在服务器端，响应按钮的 click 事件，在该事件处理函数中，调用 Invoice 类的 AddInvoice 方法，用户输入的相关信息添加到数据库中。同时将操作的结果返回到客户端。成功添加会弹出如下界面。如图 6-7 所示。

为了保证数据库数据的完整性，在添加数据时，对用户的输入进行验证，确保用户输入符合要求，如果添加的数据为空或错误，系统会给用户相应的提示，如图 6-16 所示。

添加发票信息

所属客户:

000001

发票抬头:

不能为空!

发票类型:

个人金额发票

发票金额:

不能为空!

开发票日期:

日期不能为空!

状态:

创建

备注:

添加

返回

图 6-16 系统提示出错页面

在发票信息表中，点击某一发票的编辑热键跳转到 UpdateInvoice.aspx 页面，对该发票信息进行编辑，如图 6-17 所示。

修改发票信息

所属客户:

000001

发票抬头:

AAA

发票类型:

个人金额发票

发票金额:

1000.0000

开发票日期:

2007-1-1

状态:

创建

备注:

修改

返回

图 6-17 编辑该商品信息页面

当用户对该发票的信息修改完毕后，点击修改按钮，将页面回到服务器，在

服务器端，通过 Updatetn_click 函数，处理该按钮的 click 事件，在该事件处理函数中，调用 Invoice 类的实例 updateInvoice 方法，更新编辑的销售商品的信息，同时将操作结果返回到客户端。成功编辑弹出如下界面：如图 6-10 所示。

在发票信息表中，点击某一发票的删除热键跳转到 DeleteInvoice.aspx 页面，进入删除该发票的状态，如图 6-18 所示。

图 6-18 删除该商品信息页面

点击删除按钮，系统弹出确认删除信息对话框，点击确认后页面回发到服务器，在服务器端，通过 DeletInvoice_click 函数，处理该按钮的 click 事件，在该事件处理函数中，调用 Invoice 类的实例 DeleteInvoice 方法，更新删除的销售商品信息，同时将操作结果返回到客户端。成功删除弹出如下界面：如图 6-12 所示。

6.4 统计管理

6.4.1 统计管理的数据访问层的实现

统计管理模块主要实现按商品、商品类别、销售人员、客户信息四种方式统计商品的销售信息，在数据访问层 product 类中实现了这四种统计方法。

GetProductByProduct 方法根据商品 ID 查询该商品的销售信息；
GetProductByProductType 方法根据商品所属种类查询该类别商品销售信息；
GetProductByContact 方法根据销售人员编号查询该销售人员所销售商品的统计信

息：**GetProductByCustomer** 方法根据客户编号查询与该客户相关的商品的统计信息。

这四个统计方法在实现流程上基本一致，下面以商品名称统计销售信息为例说明统计方法的实现过程。

Product 类中的 **GetProductByProduct** 方法，根据商品 ID，在数据库中查询出该商品的销售信息，其具体实现流程为：首先创建 **SQLHelper** 类的对象 **sqlHelper**，然后调用该对象的 **RunProc** 方法并传入相关参数，执行存储过程 **Pr_GetProductByProduct** 查询数据库，将查询结果保存到类型为 **SqlDataReader** 的变量 **dr** 中并返回。

存储过程 **Pr_GetProductByProduct** 的主要作用是根据用户选择的商品 ID 查询 **Product** 表、**SaleOrder** 表和 **OrderProduct** 表，返回该商品的商品 ID、商品名、总费用、合同签订日期、销售数量、销售价格和备注等信息。

6.4.2 统计管理模块实现

统计模块主要是根据用户选择的统计类型统计商品的销售信息，方便管理人员了解当前的销售状况。下面以按商品名称统计销售信息为例，详细介绍统计管理模块的实现。

按商品统计销售信息，主要根据用户在商品下拉列表中选择的商品，查询出该商品的信息，具体实现过程如下：当用户选择某一商品后，触发 **DropDownList**（下拉列表）控件的 **SelectedIndexChanged** 事件，在该事件的处理函数 **PProductList_SelectedIndexChanged** 中调用数据访问层 **product** 对象的 **GetProductByProduct** 方法，此方法调用存储过程 **Pr_GetProductByProduct**，查询出该商品的统计信息，在从数据库查询出统计信息后，将其通过 **BindProductData** 方法，绑定到 **DataGrid** 控件上，将统计信息显示出来，如图 6-19 所示。



图 6-19 按商品统计销售信息页面

6.5 用户信息

6.5.1 系统访问权限控制

在系统中，我们实现了对用户权限的分级控制，根据用户的级别，赋予相应页面的访问权限，用户分为普通用户和管理员两种类型。普通用户只能进入系统平台，不能对各种数据信息进行具体操作，管理员进入系统平台，可对各种数据信息进行操作和维护。具体的实现，在数据访问层 `user` 类中，实现了一个名为 `IsAuthorityAdmin` 的方法，该方法根据用户 ID 从用户数据表中，查询出该用户所属的类型。

在页面中，要实现对页面的访问控制，只需要在页面的 `Page_Load` 事件中，调用 `user` 类的 `IsAuthorityAdmin` 方法，获取该用户的类型，与存储在 `Session["UserID"]` 变量相比较，如果权限满足要求，则允许访问该页，不具备该页面访问权限的用户，当点击该页面，会提示没有权限，提示信息页面。如图 6-20 所示。

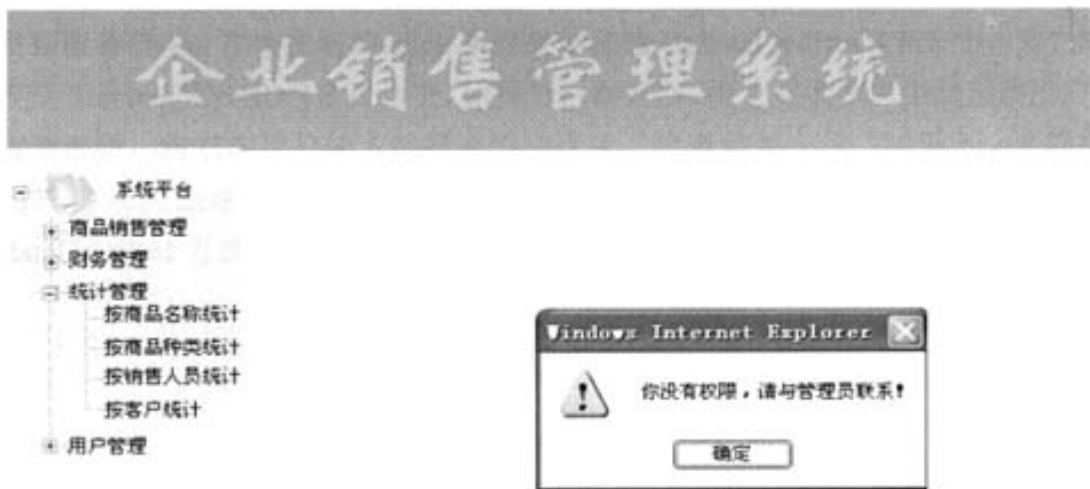


图 6-20 提示信息页面

6.5.2 用户管理

在系统中我们实现了对用户的添加、删除和修改用户密码等操作。只有超级管理员才能访问用户管理模块。

(1) 添加用户功能的实现

在 `AddUser.aspx` 页面的后台代码 `AddUser.aspx.cs` 中，首先在 `Page_Load`

方法中判断用户是否登录，如果没登录则返回登录页面；接着判断用户是否具有超级管理员权限，如果权限不正确则提示用户出错。在添加按钮的 click 事件的处理函数 AddUserBtn_Click 中，首先判断该页面用户输入的验证是否成功，然后调用数据访问层 User 类的 AddUser 方法，将用户名和密码以及用户类型等信息添加到用户表中。

（2）删除用户功能的实现

在 UserManage.aspx 页面中首先用一个列表框控件列出当前系统中所有用户，用户可以在该列表框中选择某个用户对其进行删除操作，当选择了某个用户，单击删除按钮，执行 deleteBtn_Click 方法，该方法首先判断是否已经选择了要删除的用户，如果未选择则给出“请选择用户”提示；接下来调用 User 类的 DeleteUser 方法，删除该用户。

（3）修改用户密码的实现

在 UserUpdatePwd.aspx 页面中，设置了三个文本框分别用于接收用户输入的原密码、新密码和确认新密码，该页面提供了一个按钮，用于将用户输入的信息回发到服务器端，在该按钮的 click 事件处理函数 UpdatePwdBtn_Click 中，首先判断变量 Session["UserID"]是否为空，如果为空则返回，接着在数据库中找出该用户，获取原密码，将它与用户输入的原密码相比较，检查是否一致，如果不一致则提示用户输入的原密码有误，并返回；如果一致则调用数据访问层 User 类的 UpdateUserPwd 方法，修改该用户的密码。

结 论

本文通过对信息管理系统(MIS)的概念、功能的初步研究,对管理信息系统有了比较详细的认识,在此基础上,运用当前比较主流的开发技术 ASP.NET 和 SQL Server 2000 成功开发出企业信息管理系统的一个子系统---企业销售管理系统。

本系统是基于 B/S 机构的 Web 应用管理信息系统,满足现代商用软件的远程实时通信要求。通过对销售情况的及时掌握和分析,企业可以有效地提高企业宏观控制和经营管理的质量,从而适应现代商业活动瞬息万变的节奏。

本文的主要贡献有:

- (1) 分析了一般企业的销售流程,明确了销售信息管理系统应具备的功能;
- (2) 开发结构采用了比较主流的 B/S 结构,开发模式采用了安全性高的三层设计模式;
- (3) 开发的过程中采用面向对象的方法进行分析和设计;
- (4) 系统整体进行模块化划分,使得后期维护工作变的容易;

本文的不足在于仅仅分析了一般企业的销售流程,不能深入到某一特定行业进行分析和调研,导致系统的功能趋于一般化,针对性不强。

参考文献

- [1] 薛华 成主编, 管理信息系统(MIS).第三版.清华大学出版社, 2002 年 5 月
- [2] 基于 .net 平台的中小企业网上销售系统的分析和设计, 中国优秀硕士学位论文全文数据库, 2006 年 3 月
- [3] 基于 J2EE 构建企业销售管理系统的研究与设计, 中国优秀硕士学位论文全文数据库, 2005 年 4 月
- [4] 胡海璐 彭 接文 胡 智宇, XML Web Services 高级编程范例, 电子工业出版社出版, 2003 年
- [5] [美]Craig Larman 姚淑珍 李 虎等译, UML 和模式应用面向对象分析与设计导论, 机械工业出版社出版, 2003 年
- [6] 启明工作室, 《MIS 系统开发与参考》, 人民邮电出版社, 2005 年
- [7] 刘大伟, Microsoft SQL Server 2000 数据库管理, 北京希望电子出版社 2001 年 5 月
- [8] (美) Kalen Delaney, Microsoft SQL Server 2005 技术内幕: 存储引擎, 电子工业出版社 2007 年 9 月
- [9] [美]Kris Jamsa.NET Web, 服务解决方案应用编程电子工业出版社, 2003 年
- [10] 陈学 英, Microsoft ASP.NET, 深入编程北京希望电子出版, 2002 年
- [11] 李 万伦等, Microsoft ASP.NET 程序设计, 清华大学出版社, 2002 年
- [12] [美]阿德斯坦尼, 《ASP.NET Intranet 高级编程》, 机械工业出版社, 2003 年
- [13] [美]维特 (Vitter, D), 《Visual Studio.NET Application 程序设计》, 电子工业出版社, 2003 年
- [14] Har old Davis, 《Visual Basic.NET 编程指南》电子工业出版社, 2003 年
- [15] 求是科技,ASP 网络办公及商务应用系统开发实例导航, 人民邮件出版社, 2003 年
- [16] 东方人华,ASP.NET 数据库开发,清华大学出版社, 2004 年

作者简历

任勇，北京交通大学计算机学院计算机技术与应用专业硕士（2005.9-2008.3）

教育经历：

1997.9-2000.7 解放军广州通信学院四系 野战程控交换指挥 大专

2001.9-2003.7 南京陆军指挥学院研究生系本科队 合同战术指挥 本科

工作经历：

2004.6-2005.6 北京卫戍区政治部秘群处 干事

2005.9 至今 北京交通大学 硕士

攻读学位期间发表的论文：

《轻松实现强口令》网管员世界杂志社 2007.10

《紧急行动，采取应对策略》网管员世界杂志社 2007.11