

摘要

客户资源是企业的宝贵财富，客户管理系统可以有效地管理企业的客户资源，记录企业与客户之间的商业活动，这对于现代企业是非常重要的。

本系统后台数据库使用 Access，前台开发工具使用 Visual Basic。系统采用目前比较流行的 ADO 数据访问技术，并将每个数据库表的字段和操作封装到类中，从而成功地将面向对象的程序设计思想应用到数据库应用程序设计中。这也是本系统的特色和优势。

用户通过登录窗体的用户认证进入系统主界面，可以对系统的所有功能进行操作，根据不同的用户身份权限，对客户管理系统进行各种信息的录入、删除、修改、等一系列操作。

关键词：客户管理系统，Access 数据库，Visual Basic 开发语言，管理信息系统

装
订
线

Abstract

The customer resources are enterprise's precious wealth, the customer resource management system may manage the enterprise effectively the customer resources, between the recording enterprise and the customer trade activity, this regarding the modern enterprise is extremely important.

This system backstage database uses Access, the onstage development kit uses Visual Basic. The system uses the present quite popular ADO data accessing technology, and seals each database table field and the operation in the kind, thus applies successfully the object-oriented programming thought in the database application programming. This also is this system characteristic and the superiority.

The users can get into the main form by passing the user's login and they can operate the related function of the system according to their identity power, such as the administer can add、modify、delete, inquire and print the information, but the common user can only inquire and print the information.

Key words: The customer resource management system, the Access database, Visual the Basic development language, the manages of information.

装
订
线

目录

绪 论.....	1
1. 开发工具综述.....	2
1.1. Visual Basic 简介.....	2
1.1.1. Visual Basic 概述.....	2
1.1.2. Visual Basic 的特点.....	2
1.1.3. Visual Basic 开发步骤:	4
1.2. ADO 简介.....	5
1.2.1. 为什么需要 ADO.....	5
1.2.2. 什么是 ADO.....	5
1.2.3. ADO 编程模型.....	6
1.2.4. ADO 对象模型.....	6
1.3. Access 2000 简介.....	8
1.3.1. Access 2000 概述.....	8
1.3.2. Access 的功能及优点.....	9
2. 系统需求分析与总体设计.....	10
2.1. 系统可行性分析.....	10
2.1.1. 技术可行性.....	10
2.1.2. 经济可行性.....	10
2.1.3. 操作可行性.....	10
2.1.4. 法律可行性.....	10
2.2. 系统需求分析.....	10
2.2.1. 引言.....	10
2.2.2. 目的.....	11
2.2.3. 任务.....	11
2.2.4. 功能需求说明.....	11
2.2.5. 环境需求说明.....	11
2.2.6. 数据流图.....	11
2.3. 系统总体设计.....	13
2.3.1. 引言.....	13
2.3.2. 目的.....	13
2.3.3. 系统功能概述.....	13
2.3.4. 系统功能模块设计.....	14
2.4. 系统数据库设计.....	15
2.4.1. E-R 图绘制.....	15
2.4.2. 数据库建表.....	16
3. 系统详细设计.....	21
3.1. 登录窗体和系统主界面设计.....	21
3.2. 基本信息管理模块设计.....	25
3.2.1. 设计地域信息编辑窗体.....	25
3.2.2. 设计地域信息管理窗体.....	27
3.2.3. 设计选择地域信息窗体.....	31

装
订
线

3.2.4. 设计客户类别信息编辑窗体.....	33
3.3. 客户信息管理模块设计.....	33
3.3.1. 设计客户单位信息编辑窗体.....	33
3.3.2. 设计客户单位信息管理窗体.....	34
3.3.3. 设计客户联系人信息编辑窗体.....	35
3.3.4. 设计客户联系人信息管理窗体.....	36
3.3.5. 设计客户联系人信息查看窗体.....	36
3.4. 商业往来管理模块设计.....	37
3.4.1. 设计商业机会信息编辑窗体.....	37
3.4.2. 设计商业机会信息管理窗体.....	38
3.4.3. 设计商业机会查看管理窗体.....	38
3.5. 用户管理模块设计.....	39
3.5.1. 用户管理窗体.....	39
3.5.2. 编辑用户信息窗体.....	39
4. 客户管理系统的维护与调试.....	40
4.1. 测试环境.....	40
4.2. 软件测试的目的和原则.....	40
结 论.....	42
致 谢.....	43
主要参考文献.....	44
附录：部分窗体源代码.....	45

装
订
线

绪 论

本系统是在Windows 平台下，以Visual Basic 为开发工具，以SQL 语言为数据库查询语言开发的客户管理软件。该软件的功能主要是管理客户信息的添加、修改、删除和查询，客户信息包括客户单位和联系人。商业往来信息的添加、修改、删除和查询。商业往来信息包括商业机会和客户跟踪。可以实现多点操作的信息共享，相互之间可以进行准确，快捷的信息传递。采用汉字图形界面，系统界面友好美观，操作简单易行，查询灵活方便。系统维护方便可靠。

课程特点及意义：

(1) 特点：

目前市面上流行的客户资源管理系统不少。但是，对于具体的企业单位来说，因为具有自己的特殊性，因此只需要一个操作方便，功能实用，能同时满足本企业的管理及需求的系统。我做的这个系统就是一个功能实用、操作方便，简单明了的客户资源管理系统。

(2) 意义：

通过设计客户管理管理系统，让我了解了如何去开发一个简单的软件系统，开发一个系统需要分哪几个步骤以及设计思想对系统重要性；也让我了解了与广告公司客户管理系统相关方面的知识（由哪些模块、哪些功能组成等）。

装
订
线

1. 开发工具综述

1.1. Visual Basic 简介

众所周知，Visual Basic 是美国微软公司的第一个编程工具和系统开发的第一个产品。多少年来，Visual Basic 经历过初级产品试验、16 位开发平台和目前的 32 位新产品的多次的升级换代，Visual Basic 已经成了一个全新的成熟的高端产品。它以优良的性能、极强的系统开发功能、完美的视觉界面和简单易学的特点，已经赢得了全世界广大用户的偏爱。

1.1.1. Visual Basic 概述

Visual Basic 中的“Visual”是指开发图形用户界面（GUI, Graphical User Interface）的方法。Visual 的意思是“可视的”，也就是直观的编程方法。在 VB 中引入了控件的概念，各种各样的按钮、文本框、选择框等。VB 把这些控件模式化，并且每个控件都由若干属性来控制其外观、工作方法。这样，采用 Visual 方法无需编写大量代码去描述界面元素的外观和位置，而只要把预先建立的控件加到屏幕上即可。就像使用画图之类的绘图程序，通过选择画图工具来画图一样。

“basic”是指 BASIC (Beginners All-purpose Symbolic Instruction Code) 语言，一种在计算机技术发展史上应用得最为广泛的语言。Visual Basic 在原有 BASIC 语言的基础上进一步发展，至今包含了数百条语句、函数及关键词，其中很多和 Windows GUI 有直接关系。专业人员可以用 VB 实现其他任何 Windows 编程语言功能，而初学者只要掌握几个关键词就可以建立实用的应用程序。

1.1.2. Visual Basic 的特点

Visual Basic 是从 BASIC 发展而来的，对于开发 Windows 应用程序而言，VB 是目前所有开发语言中最简单、最容易使用的语言。总的来说，Visual Basic 有以下主要特点。

(1) 可视化的设计平台

用传统程序设计语言编程时，需要通过编写代码设计用户界面，在设计过程中看到界面的实际显示效果，必须在运行程序时才能观察到。如果对界面的效果不满意，还要回到程序中修改，这一过程常常需要反复多次，大大影响了软件开发效率。Visual Basic 提供的可视化设计平台，把 Windows 界面设计的复杂性“封装”起来，开发人员不必为界面的设计而编写大量程序代码，只需按照设计的要求，用系统提供的工具在屏幕上画出各种对象即可。Visual Basic 自动产生界面设计代码，程序员只需要编写实现程序功能的那部分代码，从而大大提高了程序设计的效率。

(2) 面向对象的设计方法

面向对象的设计方法(OOP, Object Oriented Programming)从应用领域内的问题着手，以直观自然的方式描述客观世界的实体。Visual Basic 作为一种面向对象的编程方法，把程序和数据封装起来作为一个对象，并为每个对象赋予相应的属性。在设

计对象时，不必编写建立和描述每个对象的程序代码，而是用工具画在界面上，由 Visual Basic 自动生成对象的程序代码并封装起来。

(3) 结构化的设计语言

Visual Basic 是在 BASIC 语言基础上发展起来的，具有高级程序设计语言的语句结构，接近与自然语言和人类的逻辑思维方式，其语句简单易懂。其编辑器支持彩色代码，可自动进行语法错误检查，具有功能强且使用灵活的调试器和编译器。在设计 Visual Basic 程序的过程中，随时可以运行程序，而在整个应用程序设计好之后，可以编译生成 .exe 可执行文件，.exe 文件可脱离 Visual Basic 环境直接在 Windows 环境下运行。

(4) 事件驱动的编程机制

Visual Basic 通过事件来执行对象的操作，例如命令按钮是一个对象，当用户单击该按钮时，将产生一个单击事件，而在产生该事件时执行一段程序，用来实现指定的操作。在用 Visual Basic 设计应用程序时，不必建立具有明显开始和结束的程序，而是编写若干个微小的子程序，即过程。这些过程分别面向不同的对象，由用户操作引发某个事件来驱动完成某种特定功能，或由事件驱动程序调用通过过程执行指定的操作。

(5) 充分利用 Windows 资源

Visual Basic 提供的动态数据交换 (DDE, Dynamic Data Exchange) 编程技术，可以在应用程序中实现与其他 Windows 应用程序建立动态数据交换、在不同的应用程序之间进行通信。对象链接与嵌入 (OLE, Object Linking and Embedding) 技术将程序都看作一个对象，把不同的对象链接起来，嵌入到某应用程序中，从而可以得到具有声音、影像、图像、动画。文字各种信息的集合式文件。动态链接库 (DLL, Dynamic Link Libraries) 将 C / C++ 或汇编语言编写的程序加入到 Visual Basic 的应用程序中，或是调用 Windows 应用程序接口 (API, Application Programming Interface) 函数，实现 SDK (Software Development Kit) 所具有的功能。

(6) 开放的数据库功能与网络支持

Visual Basic 具有很强的数据库管理功能，不仅可以管理 MS Access 格式的数据库，还能访问其他如 FoxPro 等格式的数据库。同时 VB 还提供了开放式数据库连接 (ODBC, Open DataBase Connectivity) 功能，可以通过直接访问或建立连接的方式使用并操作后台大型网络数据库，如 SQL Server 等。在应用程序中，可以使用结构化查询语言 (SQL, Structured Query Language) 直接访问 Server 上的数据库，并提供简单的面向对象的库操作命令、多用户数据库的加锁机制和网络数据库的编程技术，为单机上运行的数据库提供 SQL 网络接口，以便在分布式环境中快速而有效地实现客户 / 服务器 (Client / Server) 方案。

1.1.3. Visual Basic 开发步骤:

- (1) 设计并建立用户界面。
- (2) 编写响应事件的代码。
- (3) 创建过程所需并调用其他过程。
- (4) 测试和调试。
- (5) 转换为运行时版本。
- (6) 准备可发布的文件。

装
订
线

1.2. ADO 简介

1.2.1. 为什么需要 ADO

从简单的文本文件到各种复杂的关系型数据库，数据库应用程序需要面对各种各样的数据源。Visual Basic 6.0 提供 ADO(Active Data Objects) 作为应用程序和 OLE-DB 连接的桥梁。

尽管用户对数据的要求种类繁多，但典型的数据源都支持 ODBC（开放式数据库连接标准）和 SQL（结构化查询语言）的。

1.2.2. 什么是 ADO

ADO, 即 Active 数据对象 (Active Data Objects); 实际是一种提供访问各种数据类型的连接机制。ADO 设计为一种极简单的格式, 通过 ODBC 的方法同数据库接口相连。用户可以使用任何一种 ODBC 数据源, 即不仅适合于 SQL Server、Oracle、Access 等数据库应用程序, 也适合于 Excel 表格、文本文件、图形文件和无格式的数据文件。ADO 是基于 OLE-DB 之上的技术, 因此 ADO 通过其内部的属性和方法提供统一的数据访问接口方法。

Microsoft 在 Visual Basic 6.0 以后的版本都集成了 ADO。它与以前 Visual Basic 的 DAO、RDO 相比有了很大的提高。DAO(Data Access Objects) 即数据访问对象, 是一种面向对象的界面接口。通过 DAO / Jet 功能可以访问 ISAM 数据库, 使用 DAO / ODBC Direct 功能可以实现远程 RDO 功能。RDO(Remote Data Objects) 即远程数据对象, 为使用代码来创建和操作一个远程 ODBC 数据库系统的各个部件提供了一个框架。RDO 是 ODBC API 的一个浅层界面接口, 是专为访问远程 ODBC 关系数据源而设计的。

ADO 集中了 DAO 和 RDO 的优点, 可以通过简单的编程实现和各种数据结构进行连接。

1.2.3. ADO 编程模型

ADO 通过下面几步来完成对数据库的操作：

- (1) 创建一个到数据源的连接(Connection), 连接到数据库; 或者开始一个事务(Transaction)。
- (2) 创建一个代表 SQL 命令行 (包括变量、参数、可选项等) 的对象。
- (3) 执行命令行。
- (4) 如果返回以表格形式组织的数据, 则将它们保存在缓存中, 产生相应的数据集对象 (Recordset)。这样便于查找、操作数据。
- (5) 通过数据集对象对数据进行各种操作, 包括修改、增加、删除等。
- (6) 更新数据源, 如果使用事务, 确认是否接受事务期间发生的数据变化。
- (7) 结束连接和事务。

1.2.4. ADO 对象模型

使用 ADO 的目的是进入数据源, 使得修改、更新数据成为可能。ADO 提供一系列的类和对象来完成各种操作。

(1) 连接对象——Connection

一个连接 (Connection) 对象代表一个到指定数据源的成功连接。应用程序通过一个连接 (包括实现数据交换的环境设置) 访问数据源, 也可以直接访问数据源 (在两层数据结构的系统中比较常见), 或者通过类似 Microsoft Internet Information Server 的中间层间接访问数据源 (这就是通常说的三层结构)。连接成功, Connection 以对象的形式存在。

使用连接对象的各种属性和方法可以完成如下操作：

- 在打开连接前需要设置 ConnectionString、ConnectionTimeout 和 Mode 属性。
- 设置 Cursorlocation 属性, 可以设置或者返回指针位置。
- 设置 DefaultDatabase 属性可以为连接指定一个默认的数据库。
- 设置 IsolationLevel 属性可以确定事务 (Transaction) 在连接的隔离等级。
- 设置 Provider 可以为连接指定一个 OLE-DB 寄主。
- 使用 Open 和 Close 方法来建立和中断一个连接。
- 使用 Execute 方法在连接上执行命令, 设置 CommandTimeout 属性来指定或者返回命令执行时间。
- 使用 BeginTrans、CommitTrans 和 RollbackTrans 方法以及 Attributes

属性来管理事务。

- 通过检查 Errors 对象可以知道数据源返回的错误。
- 通过 Version 属性可以知道 ADO 的版本。
- 使用 OpenSchema 方法可以知道数据库的计划信息。

(2) 命令行对象—Command

命令行对象是对数据源将要执行的一系列操作的定义。使用命令行对象来查询数据库并返回数据集对象（Recordset）形式的查询结果。命令行对象指定变量、参数以及可选项来完成对数据的操作，如添加记录、删除记录、更新记录等。

使用命令行对象的各种属性和方法可以完成如下操作：

- 设置 CommandText 属性可以定义命令行的内容，比如 SQL 语句等。
- 使用 Execute 方法执行命令行并返回一个数据集对象（Recordset）。
- 设置 CommandType 属性可以优化命令的效率。
- 设置 CommandTimeout 属性可以指定服务器等待命令执行的时间。
- 设置 ActiveConnection 属性可以将一个命令行对象与一个打开的连接关联。

(3) 数据集对象—Recordset

如果命令行指定数据以表的形式组织，执行命令行后将返回相应的数据集。通过数据集可以对记录以及组成记录的列进行各种操作。

1.3. Access 2000 简介

1.3.1. Access 2000 概述

Access 2000 是 Microsoft 公司推出的 Office 2000 办公软件中的组件，是数据库程序。利用它可以实现：

- 存储几乎无限多的信息；
- 按照用户的工作方式组织信息；
- 根据指定的选择标准检索信息；
- 创建窗体，使用户输入信息更方便；
- 生成可以合并数据、文本、图片和其他对象的实用直观的报告；
- 方便地通过 Web 共享信息。

“数据库”是一个相当专业的词汇，可以定义为将信息组织成列表的集合。这种定义也许不太全面，但无论什么时候使用或制作信息列表（姓名、地址、产品、顾客或发票等），其实都是在使用数据库。

存储在计算机中的数据库与记录在纸上、卡片上或通讯簿上的简单列表相比更加灵活实用。例如，用数据库可以生成最重要的电话号码列表、打印贺卡的邮递标签、生日列表等等。

将数据存储在计算机中有几种方法，例如，可以将列表存储在 Microsoft Word 或 Microsoft Excel 工作表中。但是这样做将会遇到一个问题，即在一些重复信息上消耗很多精力。如某人因家庭搬迁或姓名更改，就不得不更新他的大量信息。对于一个小列表来说，这算不上什么，但对于经常需要更新信息的列表来说（例如地址列表），用这种方法来保持最新的数据无疑是一项繁重的工作。

如果将地址信息保存在 Access 数据库中，可以给每条信息分配一个唯一的标识号，以便在每次需要使用或更改该信息时，只引用该标识号即可，这样就做到了每条信息只需输入一次。Access 数据库中最基本的组件是“表”，在表中信息排成若干记录（行），每条记录包括多个字段（列）。除了表，Access 数据库中还包括查询、窗体、报表、数据访问页和模块等组件，功能强大而且使用方便。

1.3.2. Access 的功能及优点

Access 是一个基于关系型数据库的中小型数据库应用系统，虽然它的“出道”时间比较晚，但由于它的强大的功能和出众的易用性，很快得到了广大数据库应用程序员们的青睐，从而成为当今最通用的数据库软件。

作为一种出色的数据库软件，Access 可以实现以下的功能：

- (1) 数据库中包含多个表，每个表可以分别表示和存储不同类型的信息。
- (2) 通过建立各个表之间的关联，从而将存储在不同表中的相关数据有机地结合起来。
- (3) 通过创建查询，用户可以在一个或多个数据表中检索、更新和删除记录，并且可以对数据库中的数据执行各种计算。
- (4) 通过创建联机窗体，用户可以直接对数据库中的记录执行查看和编辑操作。
- (5) 通过创建报表，用户可以将数据以特定的方式加以组织，从而达到分析和打印的目的。
- (6) 利用宏或 Visual Basic for Application 语言将各种数据库对象组织起来，形成一个数据库应用系统。

对于以上的各项功能，在 Access 2000 中都提供了多种向导，用户可以通过向导的帮助轻松地加以实现。

Access 2000 的优点：

完善地管理各种数据库对象，提供了很高的系统安全性。

通过向导提供的一系列的可视化过程，用户可以方便地生成和操作数据库对象。

通过导入和导出操作，用户可以和不同类型的数据库实现数据共享，简便而且快捷。

2. 系统需求分析与总体设计

2.1. 系统可行性分析

2.1.1. 技术可行性

硬件要求：主要配置为 CPU (PII300 以上)、内存 (64M 以上)、显卡 (4M 以上) 以及大容量硬盘 (10G 以上) 等。

软件要求：系统采用 Visual Basic 6.0 和 MS Access 开发，用 VB 制作系统的用户界面，Access 制作后台数据库，两者结合完成系统功能，可运行在 Windows98 及以上的系列操作系统上。

2.1.2. 经济可行性

由于本系统开发的费用比较低，而在交付使用之后几年内所得的效益完全超过软件的开发费用，此后就会带来相应的经济效益，也会因为此软件的高效性、准确性、及时性赢得广大客户的尊重，取得无形的社会效益。

2.1.3. 操作可行性

本系统运行在通用的 Windows 操作系统上，具有 Windows 相似的操作界面。具有简单易学，使用方便等特点，熟悉计算机的人员不需要再经过专门的培训即可熟练操作。

2.1.4. 法律可行性

本系统是完全有计算机学院的学生单独开发完成的软件，使用的是 VB 和 MS Access 等授权软件，它不是模仿他人已有的软件，在软件中也不会设置各种逻辑陷阱、特洛伊木马以及其他非法程序，也不会借此发布非法事件，故在法律上、专利问题上不会存在违反法律的行为。

2.2. 系统需求分析

2.2.1. 引言

需求分析是软件定义时期的最后一个阶段，这个阶段的任务仍然不是具体地解决问题，而是准确地确定“为了解决这个问题，目标系统必须做什么”。虽然在可行性研究阶段已经粗略地了解了用户的需求，甚至还提出了可行的方案，但是可行性研究的基本目的是用较小的成本在较短的时间内确定是否存在可行的解法，因此许多细节被忽略了。然而在最终的系统中却不能遗漏任何一个微小的细节，所以可行性研究并不能代替需求分析，它实际上并没有准确地回答“系统必须做什么？”这个问题。

2.2.2. 目的

需求分析不仅是软件定义时期的最后一个阶段，而且是软件开发期的第一个阶段，也是关系到软件开发成败的关键步骤。只有通过需求分析才能把软件功能和性能的总体概念描述为具体的软件需求规格说明，从而奠定软件开发的基础。本阶段所研究的对象是软件项目的用户要求，必须全面理解用户的各项要求，但又不能全盘接受所有的要求。为了达到这一目的，我们必须对其中模糊的要求进行澄清，然后才能决定是否可以采纳。准确地表达被接受的用户要求，也是需求分析的另一个重要方面，只有经过确切描述的软件需求才能成为软件设计的基础。

2.2.3. 任务

需求分析的任务还不是确定系统怎样完成它的工作，而仅仅是确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、具体的要求。需求分析的任务，在于完全地弄清用户对软件系统的确切要求，并按需求规格说明书的格式表达出来。

2.2.4. 功能需求说明

完成所有信息管理，包括信息录入、删除、修改、查询、报表打印以及对系统登录的身份验证和系统的密码修改。

2.2.5. 环境需求说明

(1) 软件运行硬件环境:

PII 或更高档的微机，笔记本电脑；

运行内存要求: $\geq 32MB$ ；

安装所需硬盘: $\geq 20MB$ ；

(2) 软件要求:

中文 Windows 95/98/2000/Me/XP；

英文 Windows 95/98/2000/Me/XP；

Microsoft Access 97/2000/2002/XP

(3) 软件开发环境:

Visual Basic 6.0 、 Access 2000

2.2.6. 数据流图

(1) 绘制数据流图的基本要求

系统的顶层图使用来表示整个系统的功能结构。随着需求分析活动的逐渐深入，较高抽象级别的复杂转换可以精化为一系列的相互关联的数据流和子转换。在数据流方法中，对数据（数据流）的精化是伴随着对转换的精化而同步进行的。

在进行逐层精化的过程中，必须维持层间数据流图的平衡，即，被精化的的转换的输入、输出流必须与精化它的数据流子图的初始输入流和最终输出流严格一致。这是确保需求模型一致性的一条重要规则。同时，要注意逐层精化必须适可而止，因为需求分析活动只求对问题的全面、清晰的理解，并不关心软件的设计细节。

(2) 客户管理系统数据流图的绘制

根据上述的一些基本要求，经过本系统实际的需求分析，绘制数据流图如图 1 示：

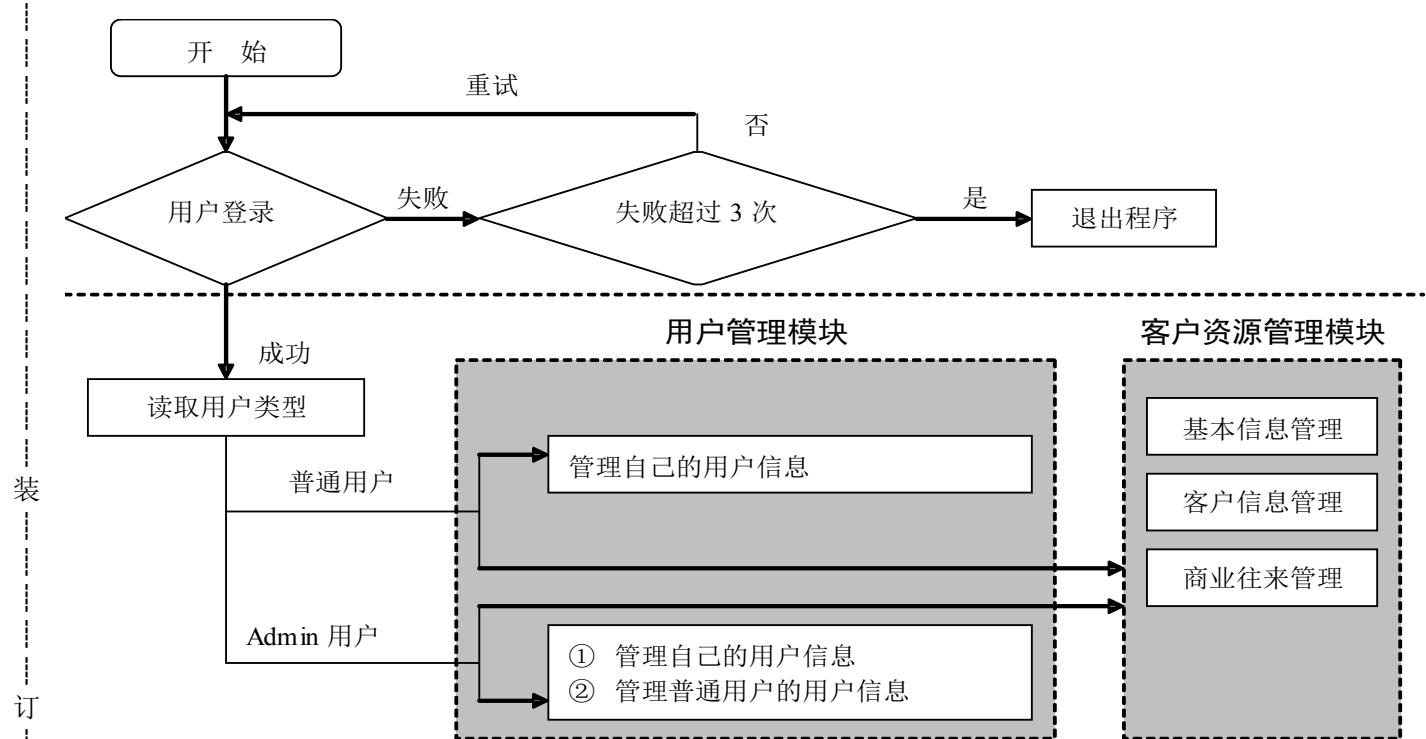


图 1 系统数据流图

装
订
线

2.3. 系统总体设计

2.3.1. 引言

在需求分析阶段的工作，系统必须“做什么”已经清楚了，现在是决定“怎样做”的时候。总体设计的基本目的就是回答“从总体上说，系统应该如何实现？”这个问题。总体设计阶段的另一项重要任务是设计软件的结构，也就是要确定系统中每个程序是由哪些模块组成的。

2.3.2. 目的

在软件开发时期中，设计阶段是最富有活力，最需要发挥创造精神的阶段。在本阶段，主要就是软件需求说明转换为用适当方式表示的软件概要设计文档，从而得出本模块的具体实现方案与总体结构以及各组成部分之间的关系。

2.3.3. 系统功能概述

本课题属于一个数据库开发的问题，主要功能简述如下：

(1) 登录窗体和系统主界面模块

当系统运行时，首先打开登录窗体，只有有权限的用户才能进行系统。登录成功后，将显示系统主界面。

(2) 基本信息管理模块

基本信息管理模块包括地域信息管理、客户类别管理、行业分类管理、职务名称管理、项目类型管理和销售人员管理。

(3) 客户信息管理模块

客户信息管理模块包括客户单位管理和客户联系人管理等功能。

(4) 商业往来管理模块

商业往来管理模块包括商业机会管理和客户跟踪管理等功能。

(5) 用户管理模块

本模块只对管理员开放。管理员可以添加用户和对所有用户的密码信息进行修改等操作。

2.3.4. 系统功能模块设计

对上述各功能进行集中、分块，按照结构化程序设计的要求，得到图 2 所示的系统功能模块图：

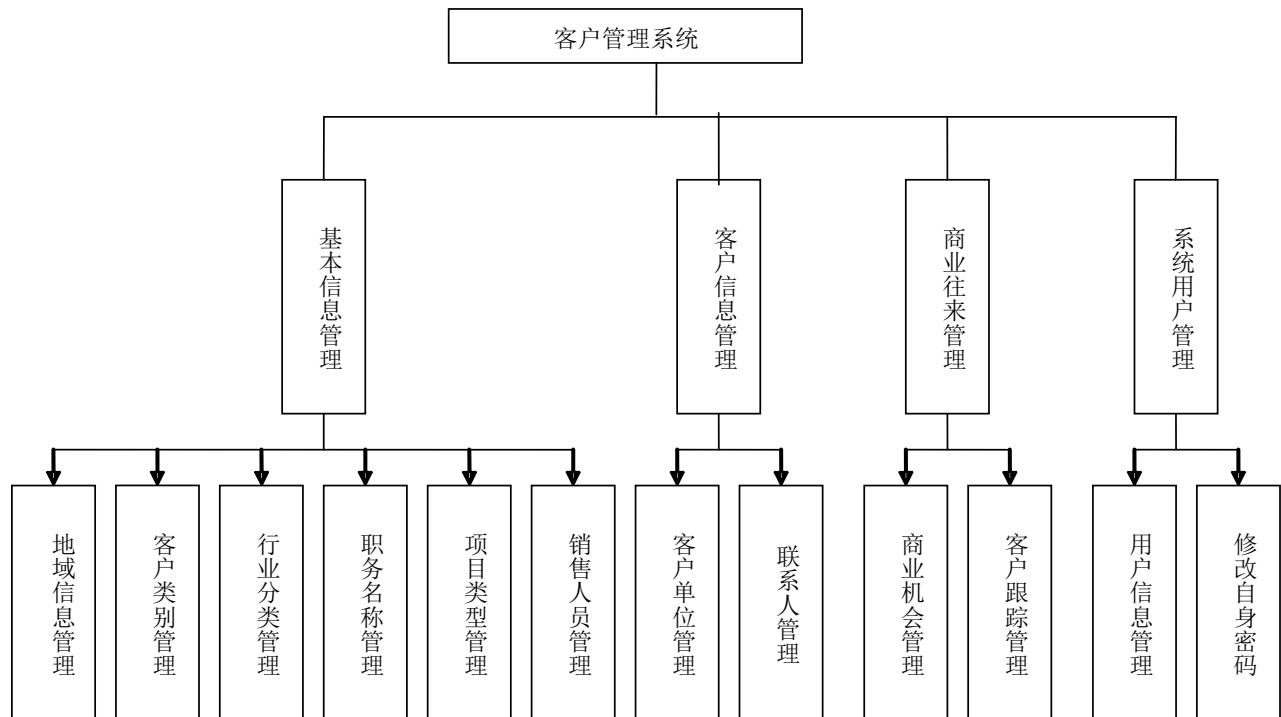


图 2 系统功能模块图

装
订
线

2.4. 系统数据库设计

数据库设计是建立数据库及其应用系统的技术，是信息系统开发和建设中的核心技术，具体说，数据库设计是指对于一个给定的应用环境，构造最优的数据模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求（信息要求和处理要求）。

2.4.1. E-R 图绘制

系统的 E-R 图如图 3 所示：

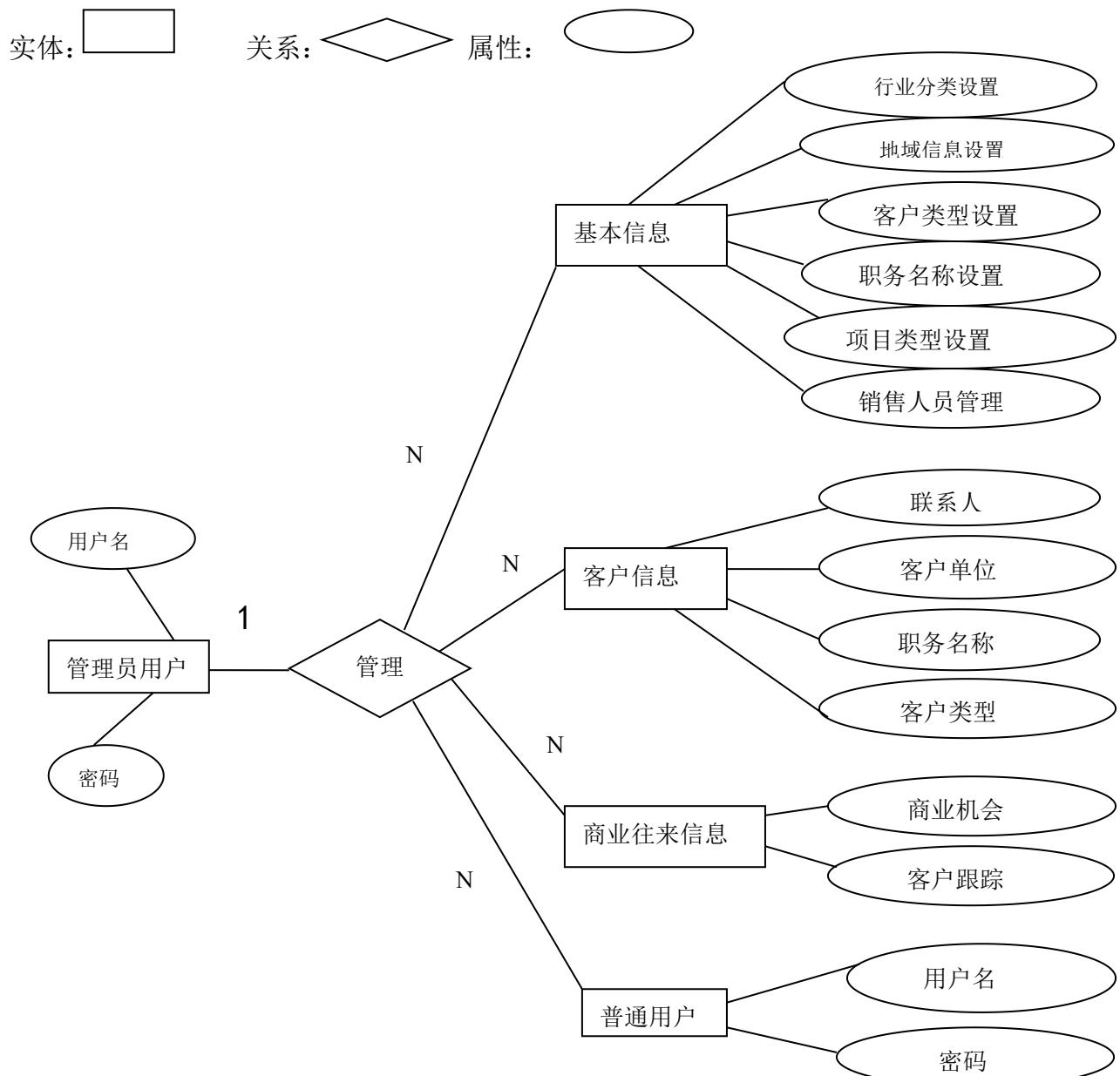


图 3 系统实体联系图

2.4.2. 数据库建表

本系统采用 Access2000 作为后台数据库。

数据库结构设计是总体设计阶段非常重要的环节，好的数据库结构可以简化开发过程，使系统功能更加清晰明确。因为数据库结构的变化会造成编码的改动，所以必须认真设计数据库结构后再进行编码，从而避免无所谓的重复工作。

本系统数据库中包含 8 个表，即基本信息表 Types、地域信息表 Area、销售人员信息表 Employess、客户单位信息表 Customer、联系人信息表 Contact、商业机会表 SaleChance、客户跟踪表 SaltTask 和用户信息表 Users。这样整个数据库设计后的结果如图 4 所示：

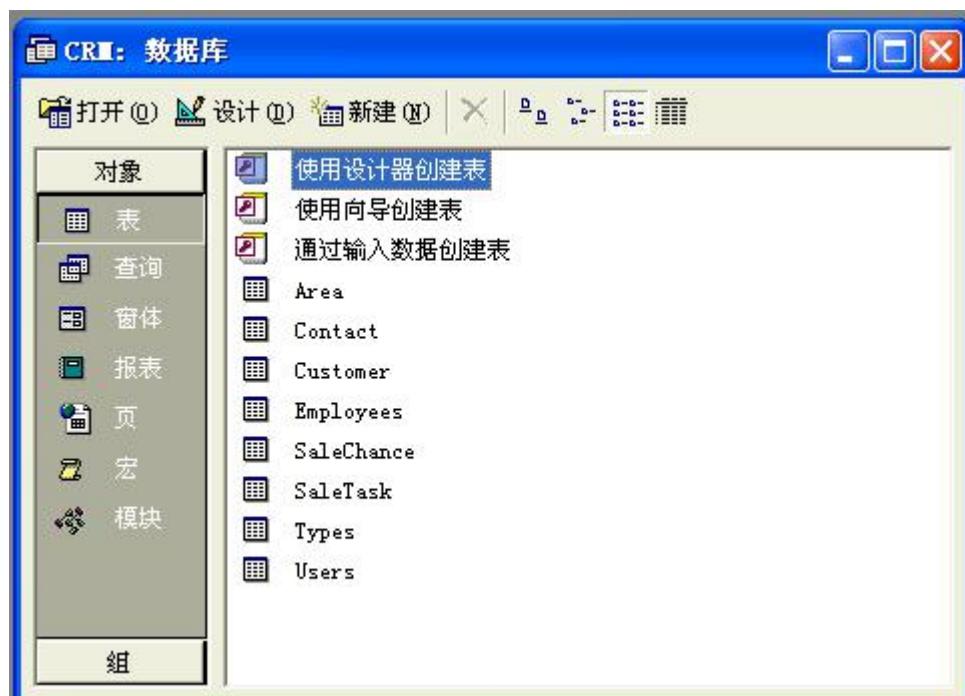


图 4 客户管理系统数据库

装
订
线

下面分别介绍各表结构：

(1) 基本信息表 Types

表 Types 用来保存客户类别信息、行业分类信息、职务名称信息和项目类型信息，结构如表 1 所示：

表 1 Types 的结构

编号	字段名称	数据结构	说 明
1	Id	自动编号	记录编号, 主键
2	TypeName	文本, 长度 50	信息名称
3	Flag	数字 / 字节	类型标记。1 表示客户类别；2 表示客户所属行业；3 表示职务名称；4 表示项目类型

(2) 地域信息表 Area

表 Area 用来保存地域信息，结构如表 2 所示：

表 2 Area 的结构

编号	字段名称	数据结构	说 明
1	AreaId	数字 / 长整型	记录编号, 主键
2	AreaName	文本, 长度 50	地域名称
3	AreaType	数字 / 字节	地域类型 1 表示洲；2 表示国家。3 表示省；4 表示市县
4	UpperId	数字 / 长整型	上级地域编号

表 Employess 用来保存销售人员信息，结构如表 3 所示：

表 3 Employess 的结构

编号	字段名称	数据结构	说 明
1	EmpId	自动编号	人员编号, 主键
2	EmpName	文本, 长度 50	姓名
3	Sex	文本, 长度 2	性别
4	DepName	文本, 长度 50	所在部门
5	ETitle	文本, 长度 50	职务

(3) 客户单位信息表 Customer

表 Customer 用来保存客户单位信息，结构如表 4 所示：

表 4 Customer 表

编号	字段名称	数据结构	说 明
1	CustId	自动编号	记录编号, 主键
2	CEastman	文本, 长度 200	课程名称
3	CustTypeId	数字 / 长整型	所属客户类型编号
4	Address	文本, 长度 200	通信地址
5	Postcode	文本, 长度 50	邮政编码
6	AreaId	数字 / 长整型	所属地域编号
7	BussTypeId	数字 / 长整型	所属行业类别编号
8	EmpId	数字 / 长整型	销售人员

装

(4) 联系人信息表 Contact

表 Contact 用来保存联系人信息，结构如表 5 所示：

表 5 Contact 的结构表

编号	字段名称	数据结构	说 明
1	Id	自动编号	记录编号, 主键
2	CustId	数字 / 长整型	客户单位编号
3	CName	文本 / 长度 50	姓名
4	Sex	文本 / 长度 2	性别
5	TitleId	数字 / 长整型	职务编号
6	Office	文本, 长度 50	办公电话
7	Mobile	文本, 长度 50	移动电话
8	Birth	文本, 长度 50	生日
9	Hobby	文本, 长度 50	兴趣爱好
10	Memo1	备注	备注信息
11	Input_date	文本, 长度 50	记录录入日期
12	Input_time	文本, 长度 50	记录录入时间

(5) 商业机会表 SaleChance

表 SaleChance 用来保存商业机会信息，结构如表 6 所示：

表 6 SaleChance 的结构表

装
订
线

编号	字段名称	数据结构	说 明
1	Id	文本, 长度 50	记录编号
2	CustId	数字 / 长整型	客户编号
3	CtId	数字 / 长整型	联系人编号
4	ProName	文本 / 长度 200	产品名称
5	PTypeId	数字 / 长整型	项目类型编号
6	EmpId	数字 / 长整型	销售人员编号
7	SignDate	文本, 长度 50	预计签订日期
8	SignSum	数字 / 长整型	预计成交金额
9	PPercent	文本, 长度 50	目前进度
10	Status	文本, 长度 50	状态
11	Detail	备注	详细介绍
12	CreateDate	日期 / 时间	创建日期
13	Poster	文本, 长度 50	创建人

(6) 客户跟踪表 SaleTask

表 SaleTask 用来保存客户任务跟踪信息, 结构如表 7 所示:

表 7 SaleTask 的结构表

编号	字段名称	数据结构	说 明
1	Id	自动编号	记录编号
2	SaleId	文本, 长度 50	商业机会编号
3	EmpId	数字 / 长整型	销售人员编号
4	TaskDate	文本 / 长度 50	任务日期
5	Purpose	文本 / 长度 50	拜访目的
6	CtId	数字 / 长整型	拜访人员编号
7	Place	文本, 长度 100	拜访地点
8	StartTime	数字 / 长整型	开始时间
9	EndTime	文本, 长度 50	目前进度
10	EndTime	Detail	结束时间
11	Detail	备注	详细介绍
12	Result	备注	任务结果、进展总结

(7) 用户信息表 Users

表 Users 用来保存系统用户信息，结构如表 8 所示。

表 8 Users 的结构表

编号	字段名称	数据结构	说明
1	UserName	文本, 长度 50	用户名, 主键
2	UserPwd	文本, 长度 50	密码

装
订
线

3. 系统详细设计

3.1. 登录窗体和系统主界面设计

当系统运行时，首先打开登录窗体，只有有权限的用户才能进行系统。登录成功后，将显示系统主界面。

(1) 设计登录窗体

登录窗体的名称为 FrmLogin，它的界面布局如图 5 所示：



图 5 登录窗体的界面布局

登录窗体的属性表如表 9 所示。

表 9 登录窗体的属性表

窗体属性	设置值	具体说明
名称	FrmLogin	设置窗体名称
BorderStyle	1-Fixed Single	设置窗体的边框属性
Caption	身份验证	窗体的标题条文本
ControlBox	False	取消控制按钮，防止用户通过控制按钮关闭对话框
StartPosition	2-屏幕中心	设置窗体弹出时，位置在屏幕中心

登录窗体中控件的属性如表 10 所示：

表 10 登录窗体包含的控件及其属性

对象名	属性	属性值
Label1	Caption	用户名
Label2	Caption	密码
txtUser	Text	空
txtPwd	Text	空
	PasswordChar	*
Cmd_Ok	Caption	确定
Cmd_Cancel	Caption	取消

下面介绍登录窗体的部分代码

①变量声明

登录窗体的变量声明如下：

```
Public PasswordKey As String  
Public Try_times As Integer  
Public NameKey As String
```

②Cmd_Click 过程

当用户单击“确定”按钮时触发 Cmd_Click 事件，进行身份验证，代码如下：

```
Private Sub Cmd_OK_Click()  
    Dim j As Single  
    ' 数据有效性检查  
    If txtUser = "" Then  
        MsgBox "请输入用户名"  
        txtUser.SetFocus  
        Exit Sub  
    End If  
    If txtPwd = "" Then  
        MsgBox "请输入密码"  
        txtPwd.SetFocus  
        Exit Sub  
    End If  
    NameKey = MakeStr(txtUser)  
    PasswordKey = MakeStr(txtPwd)
```

’ 判断用户名是否存在

```
If MyUser. In_DB(NameKey) = False Then  
    MsgBox "用户名不存在"  
    Try_times = Try_times + 1  
    If Try_times >= 3 Then  
        MsgBox "您已经三次尝试进入本系统，均不成功，系统将关闭"  
        DBapi_Disconnect  
        End  
    Else  
        Exit Sub  
    End If  
End If
```

’ 判断密码是否正确

```
MyUser. GetInfo (NameKey)  
If MyUser. UserPwd <> PasswordKey Then  
    MsgBox "密码错误"  
    Try_times = Try_times + 1  
    If Try_times >= 3 Then  
        MsgBox "您已经三次尝试进入本系统，均不成功，系统将关闭"  
        DBapi_Disconnect  
        End  
    Else  
        Exit Sub  
    End If  
End If
```

’ 登录成功，将当前用户的信息保存在 CurUser 中

```
CurUser. GetInfo (MyUser. UserName)
```

’ 关闭自己

```
Unload Me
```

```
End Sub
```

程序的运行过程如下：

- 判断是否输入了用户名和密码，如果没有输入，则返回，要求用户输入。
- 将输入的用户名和密码赋值到变量 NameKey 和 Passwordkey 中，便于处理。
- 调用 MyUser. In_DB() 函数，判断当臆用户名是否存在。如果不存在，则返回，

要求用户重新输入，同时将计数变量 Try_times 加 1。

- 如果用户名存在，则调用 MyUser.GetInfo() 函数，读取此用户的数据，并将用户密码与输入的密码进行比较。如果密码不同，要求用户重新输入，同时将计数变量 Try_times 加 1。
- 如果 Try_times 大于或等于 3，则退出系统。
- 如果通过密码验证，则将当前用户的信息保存到 curUser 对象中，以便以后使用。
- 关闭登录窗体。

(2)设计主界面

主窗体是工程的启动对象，可以在系统菜单中选择“工程” / “属性”，打开属性窗口设置启动对象。本系统中主窗体名为 FrmMain。

打开窗体的设计窗口，选择“工具”菜单项中的“菜单编辑器”，对菜单内容进行编辑，菜单编辑器如图 6 所示：

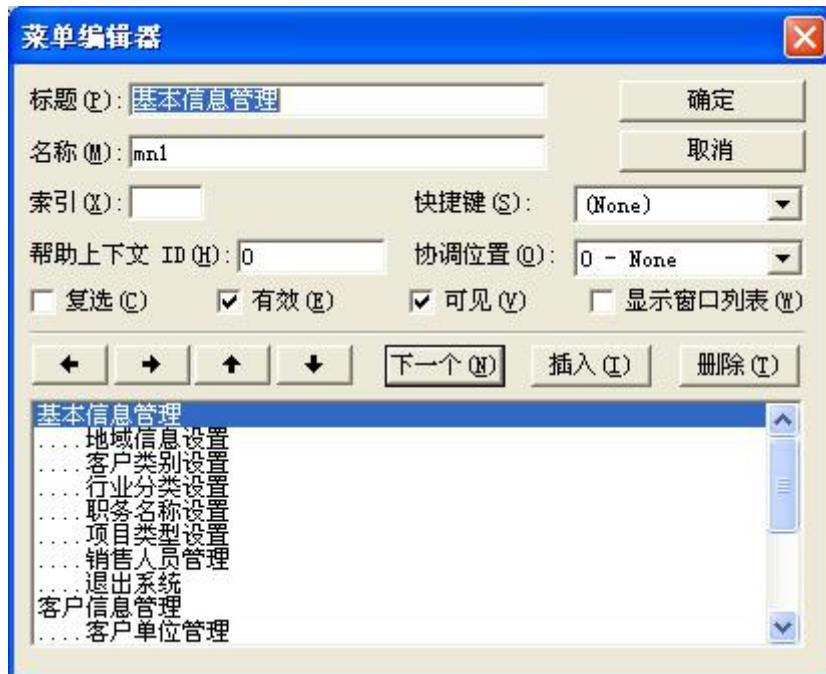


图 6 菜单编辑器

本系统的主界面如图 7 所示：

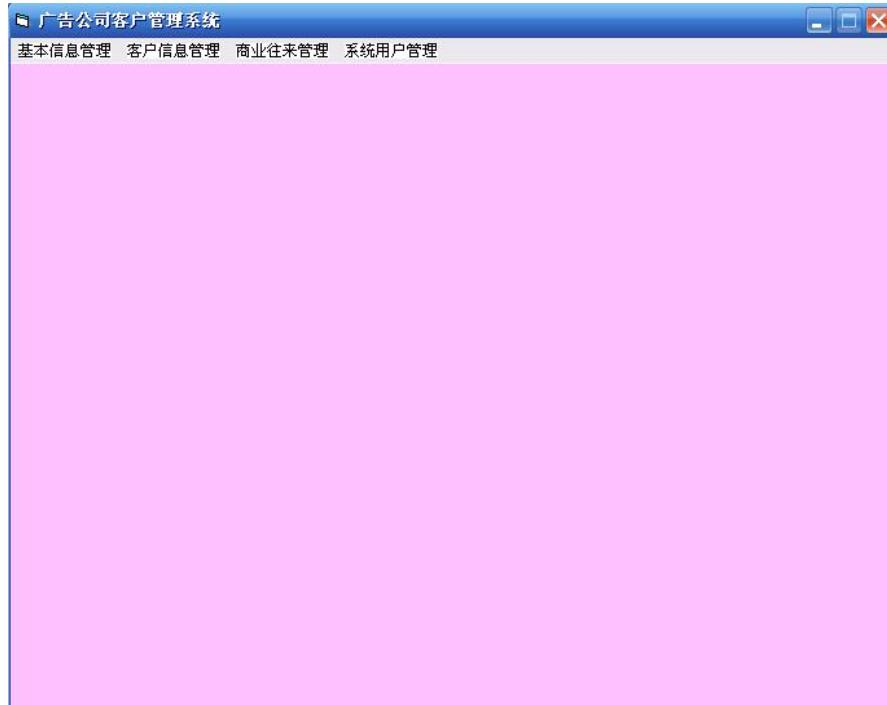


图 7 系统主界面

在主窗体 FrmMain 中添加如下代码：

```
Private Sub Form_Load()
    FrmLogin. Show 1
End Sub
```

当主窗体启动时，将打开“登录”对话框。如果不能通过身份验证，则不能进行系统。

当选择“退出系统”菜单项时触发 mn_Click 事件，代码如下：

```
Private Sub mn_Exit_Click()
    DBapi_Disconnect
End
End Sub
```

3.2. 基本信息管理模块设计

基本信息管理模块包括地域信息管理、客户类别管理、行业分类管理、职务名称管理、项目类型管理和销售人员管理。

3.2.1. 设计地域信息编辑窗体

编辑地域信息的窗体可以用来添加和修改地域信息，窗体名称设置为 FrmAreaEdit，窗体布局如图 8 所示：



图 8 窗体 FrmAreaEdit 布局

Cmd_Ok_Click 过程

当用户单击“确定”按钮时触发 Cmd_Ok_Click 事件，对应的程序代码如下：

```
Private Sub Cmd_OK_Click()
    Dim TmpId As Integer
    If Trim(txtArea) = "" Then
        MsgBox "请输入地域名称"
        Exit Sub
    End If
    ' 将用户输入的输入赋值到 MyArea 对象中
    With MyArea
        .AreaName = Trim(txtArea)
        .AreaType = CurArea.AreaType + 1
        .UpperId = CurArea.AreaId
        If Modify = False Then ' 插入
            If .In_DB(Trim(txtArea), .UpperId) = True Then
                MsgBox Trim(txtArea) + " 已经存在"
                Exit Sub
            End If
            TmpId = .Insert
            ' 生成 TreeView 控件中新结点的关键字
            Tmp_Key = Chr(Asc("a") + .AreaType)
            Tmp_Key = Tmp_Key + Trim(Str(TmpId))
            FrmAreaMan.TreeView1.SelectedItem.Image = 1
            FrmAreaMan.TreeView1.SelectedItem.ExpandedImage = 2
        End If
    End With
End Sub
```

装
订
线

```
FrmAreaMan.TreeView1.SelectedItem.SelectedImage = 3
Set TmpNode =
FrmAreaMan.TreeView1.Nodes.Add(FrmAreaMan.TreeView1.SelectedItem.Key, _
    twvChild, Tmp_Key, .AreaName, 4, 5)
TmpNode.Selected = True
Else '修改
    If CurArea.AreaName <> Trim(txtArea) Then
        If .In_DB(Trim(txtArea), .UpperId) = True Then
            MsgBox Trim(txtArea) + " 已经存在"
            Exit Sub
        End If
    End If
    Call .Update(CurArea.AreaId)
    If CurArea.AreaName <> Trim(txtArea) Then
        FrmAreaMan.TreeView1.SelectedItem.Text = Trim(txtArea)
    End If
    CurArea.AreaName = Trim(txtArea)
End If
End With
Unload Me
End Sub
```

装
订
线

3.2.2. 设计地域信息管理窗体

地域信息管理窗体的名称设置为 FrmAreaMan。在窗体中添加一个 ImageList 控件。右击此控件，选择菜单中的“属性”命令，打开“属性页”对话框。单击“图像”选项卡，通过单击“插入图像”和“删除图像”来编辑图像列表，如图 9 所示：

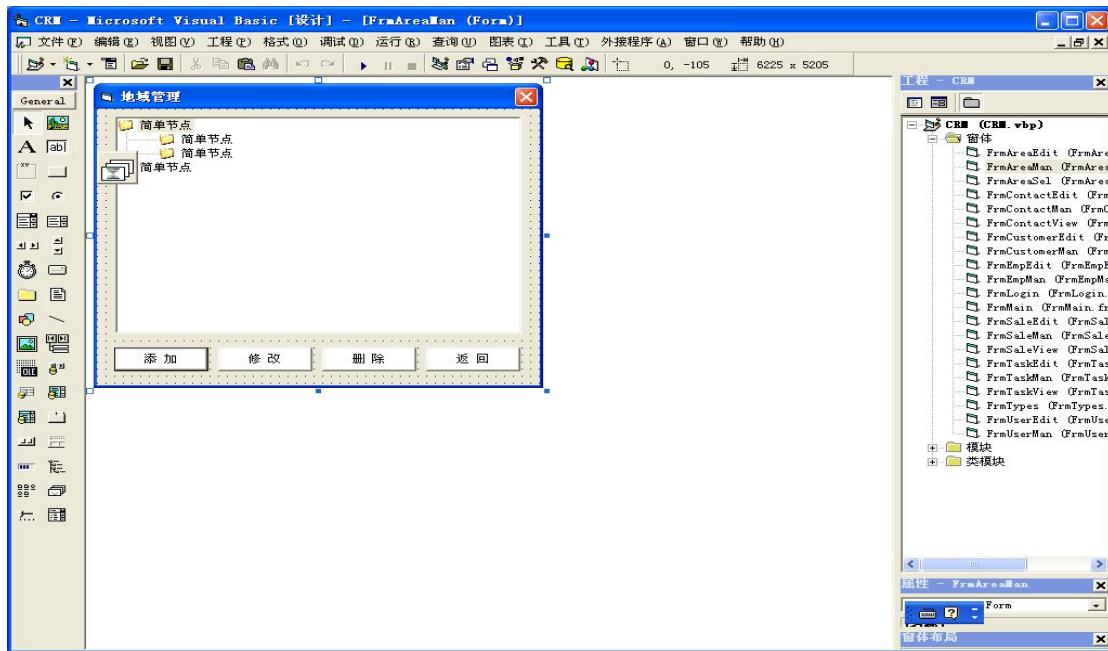


图 9 地域信息管理窗体

下面分析窗体 FrmClassMan 中部分过程的代码。

1、Form_Load 过程

```
Private Sub Form_Load()
FocusAreaName = "中国"
    '添加根结点
Set TmpNode = TreeView1.Nodes.Add(, , "a0", "全球地域", 1, 3)
TmpNode.Selected = True
TmpNode.ExpandedImage = 2
```

```
Call Add_AreaToTree(TreeView1)
```

```
TreeView1_Click
```

```
End Sub
```

(1) Cmd_Add_Click 过程

```
Private Sub Cmd_Add_Click()
If CurArea.AreaType >= 4 Then
    MsgBox "此地域已不能再细分"
    Exit Sub
End If
    '打开编辑窗体
FrmAreaEdit.lblUpper = CurArea.AreaName
```

```
FrmAreaEdit.txtArea = ""  
FrmAreaEdit.Modify = False  
FrmAreaEdit.Show 1  
TreeView1_Click  
End Sub
```

(2)Cmd_Modi_Click 过程

当单击“修改”按钮时触发 Cmd_Modi_Click 事件，对应的代码如下：

```
Private Sub Cmd_Modi_Click()  
    If TreeView1.SelectedItem.Key = "a0" Then  
        MsgBox "此项不能修改"  
        Exit Sub  
    End If  
    FrmAreaEdit.lblUpper = MyArea.GetName(CurArea.UpperId)  
    FrmAreaEdit.txtArea = CurArea.AreaName  
    FrmAreaEdit.Modify = True  
    FrmAreaEdit.Show 1  
    TreeView1_Click  
End Sub
```

(3)Cmd_Del_Click 过程

当单击“删除”按钮时触发 Cmd_Del_Click 事件，对应的代码如下：

```
Private Sub Cmd_Del_Click()  
    Dim TmpOrg As String  
    If CurArea.AreaType = 0 Then  
        MsgBox "此项不能删除"  
        Exit Sub  
    End If  
    If CurArea.HaveSon(CurArea.AreaId) = True Then  
        MsgBox CurArea.AreaName + " 包含下级地域，不能删除"  
        Exit Sub  
    End If  
    TmpOrg = MyCust.Area_In_Customer(TmpAreaId)  
    If TmpOrg <> "" Then  
        MsgBox "地域中包含客户信息- " + TmpOrg + "，不允许删除"  
        Exit Sub
```

装订线

```
End If
' 确认删除
If MsgBox(" 是 否 确 定 要 删 除 " +
Trim(TreeView1.SelectedItem.Text), vbYesNo, "请确认") = vbNo Then
    Exit Sub
End If
CurArea.Delete (CurArea.AreaId)
TreeView1.Nodes.Remove TreeView1.SelectedItem.Index
TreeView1_Click
End Sub
```

(4)TreeView1_Click 过程

当单击 TreeView 列表时触发 TreeView1_Click 事件，对应的代码如下：

```
Private Sub TreeView1_Click()
    Dim TmpType As Integer
    ' 根据关键字获取地域信息
    FocusKey = TreeView1.SelectedItem.Key
    CurArea.AreaName = TreeView1.SelectedItem.Text
    TmpType = Asc(Left(TreeView1.SelectedItem.Key, 1))
    CurArea.AreaType = TmpType - Asc("a")
    CurArea.AreaId = Val(Right(TreeView1.SelectedItem.Key,
Len(TreeView1.SelectedItem.Key) - 1))
    ' 如果是根结点，则不再处理
    If CurArea.AreaType = 0 Then
        Exit Sub
    End If
    ' 决定结点显示的图像
    If CurArea.HaveSon(CurArea.AreaId) = False Then
        TreeView1.SelectedItem.ExpandedImage = 4
        TreeView1.SelectedItem.Image = 4
        TreeView1.SelectedItem.SelectedImage = 5
    End If
    ' 将地域信息读取到 CurArea 对象中
    CurArea.GetInfo (CurArea.AreaId)
End Sub
```

3.2.3. 设计选择地域信息窗体

为了方便用户在其他模块中选择地域信息，设计一个选择地域信息的窗体，窗体名为 FrmAreaSel。它的布局如图 10 所示：

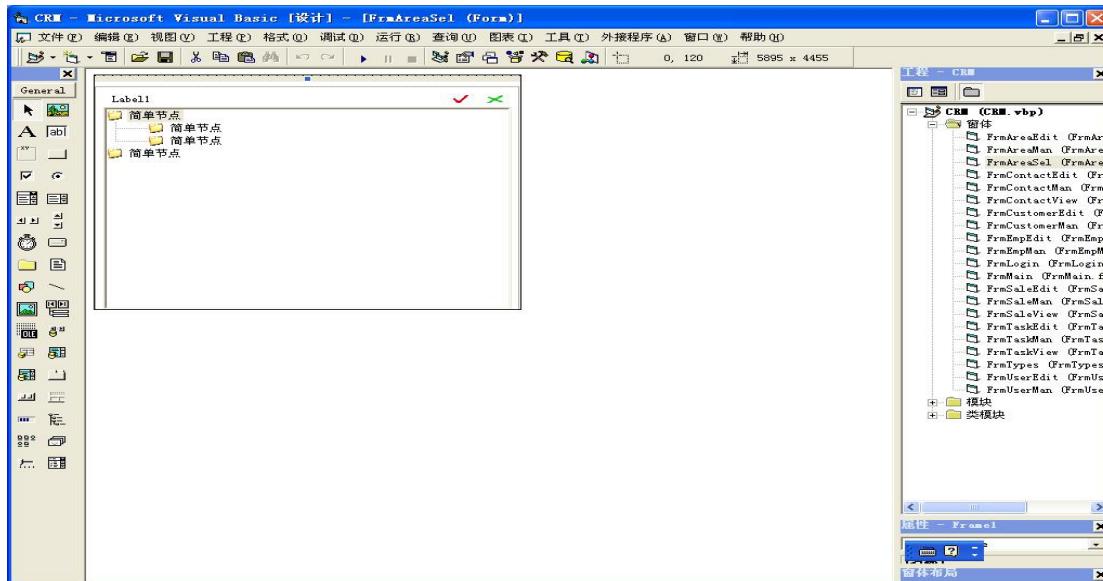


图 10 选择地域信息窗体

窗体的属性如表 11 所示：

表 11 设置窗体的属性

窗体属性	设置值	具体说明
名称	FrmAreaSel	设置窗体名称
BorderStyle	1-Fixed Single	设置窗体的边框属性
Caption		窗体的标题条文本为空
ControlBox	False	取消控制按钮
StartPosition	0-手动	设置窗体弹出时，位置由用户指定

当用户双击 TreeView 控件时触发 TreeView1_DbClick 事件，对应的代码如下：

```
Private Sub TreeView1_DbClick()
    Dim TmpStr As String
    If AreaLevel > 0 And AreaLevel < 5 And AreaLevel <> MyArea.AreaType
    Then
        Select Case AreaLevel
            Case 1
                TmpStr = "洲"
            Case 2
                TmpStr = "国家"
        End Select
    End If
End Sub
```

```
Case 3
TmpStr = "省"
Case 4
TmpStr = "市/县"
End Select
MsgBox "请选择 " + TmpStr
Exit Sub
Else
If AreaLevel = 5 Then '选择国家、省、市/县
If MyArea.AreaType < 2 Or MyArea.AreaType > 4 Then
MsgBox "请选择 国家、省、市或县"
Exit Sub
End If
End If
If AreaLevel = 6 Then '选择省、市/县
If MyArea.AreaType < 3 Or MyArea.AreaType > 4 Then
MsgBox "请选择 省、市或县"
Exit Sub
End If
End If
End If
CurArea.AreaId = MyArea.AreaId
CurArea.AreaName = MyArea.AreaName
CurArea.AreaType = MyArea.AreaType
ClickOK = True
Unload Me
End Sub
```

装
订
线

3.2.4. 设计客户类别信息编辑窗体

客户类别信息编辑窗体的名称为 FrmTypes，窗体中包含的控件及其属性如表 12 所示：

表 12 窗体 FrmTypes 中包含的控件及其属性

对象名	属性	属性值
DataList1	Name	DataList1(用来显示客户类别名称)
AdoType	Visible	False(用于提供数据源)
txtName	Caption	txtName(用来编辑分类名称)
Cmd_Add	Caption	添加
Cmd_Modify	Caption	修改
Cmd_Delete	Caption	删除
Cmd_Exit	Caption	退出

装

订

线

窗体 FrmType 的布局如图 11 所示：



图 11 客户类别信息窗体

3.3. 客户信息管理模块设计

客户信息管理模块包括客户单位管理和客户联系人管理等功能。

3.3.1. 设计客户单位信息编辑窗体

客户单位信息编辑窗体的窗体为 FrmCustomerEdit，窗体布局如图 12 所示：



图 12 客户单位信息窗体

3.3.2. 设计客户单位信息管理窗体

客户单位信息管理窗体的名称为 FrmCustomerMan，窗体布局如图 13 所示：

单位名称	地域名称	客户类型	所属行业	通信地址	邮政编码	销售人员
安徽江淮扬天汽车股份有限公司	滁州市	供应商	金融行业	安徽滁州市南谯区乌衣239000	王明	
安徽喜梦床具有限公司	滁州市	生产商	制造业	安徽滁州市城南科技	239000	王明
上海大众汽车滁州销售服务	滁州市	代理	金融行业	安徽滁州市凤阳路6	239000	赫赫

图 13 客户单位信息管理窗体

装
订
线

窗体属性与登录窗体相似，窗体中主要控件的属性如表 13 所示：

表 13 设置窗体中主要控件的属性

对象名	属性	属性值
ComboArea		用于选择地域
Check1		用于选择全部地域
Dco_Ctype		用于选择客户类别
Check2		用于选择全部客户类别
AdoType		为 dco_Ctype 提供数据源
lblTotal		用于显示全部客户单位数量
lblSel		用于显示当前选择客户单位的数量
DataGrid1		用于显示客户单位信息
Adodc1		为 DataGrid1 控件提供数据源
Cmd_Add	Caption	添加
Cmd_Modify	Caption	修改
Cmd_Delete	Caption	删除
Cmd_Exit	Caption	返回

3.3.3. 设计客户联系人信息编辑窗体

客户联系人信息管理窗体的名称为 FrmContactEdit，窗体布局如图 14 所示：



图 14 客户联系人信息管理窗体

3.3.4. 设计客户联系人信息管理窗体

客户联系人信息管理窗体的名称为 FrmContactMan，窗体布局如图 15 所示：



图 15 客户联系人信息管理窗体

3.3.5. 设计客户联系人信息查看窗体

客户联系人信息查看窗体的名称为 FrmContactView，窗体布局如图 16 所示：

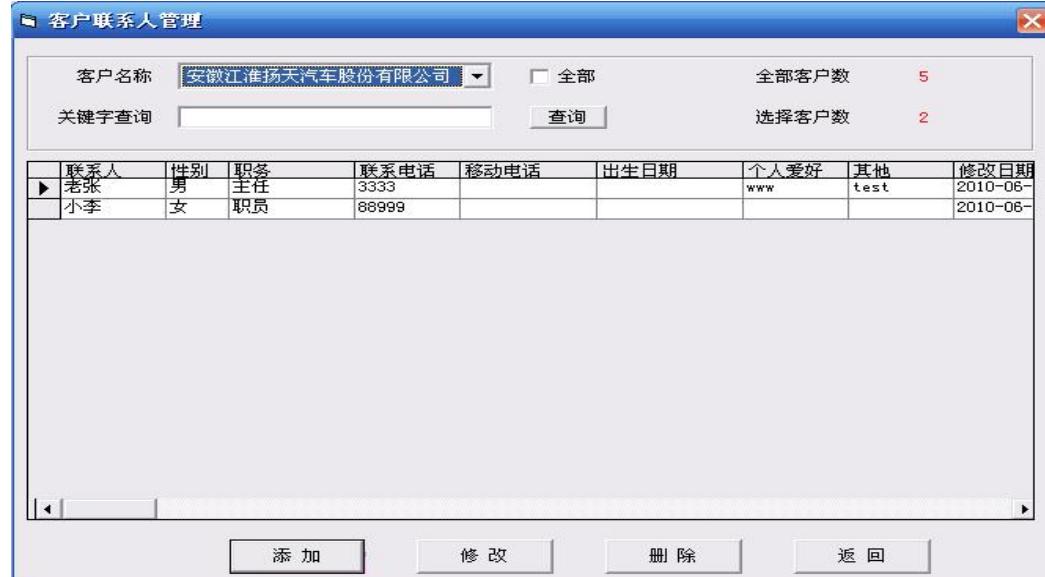


图 16 客户联系人信息查看窗体

3.4. 商业往来管理模块设计

商业往来管理模块包括商业机会管理和客户跟踪管理等功能。

3.4.1. 设计商业机会信息编辑窗体

商业机会信息编辑窗体的名称为 FrmSaleEdit，窗体布局如图 17 所示：



图 17 商业机会信息编辑窗体

装
订
线

3.4.2. 设计商业机会信息管理窗体

商业机会信息管理窗体的名称为 FrmSaleMan，窗体布局如图 18 所示：

The screenshot shows a Windows application window titled "商业机会管理". At the top left is a dropdown menu icon. On the right side are standard window controls: a close button (X), a minimize button, and a maximize button. Below the title bar is a toolbar with four buttons: "重置" (Reset), "添加" (Add), "修改" (Modify), and "删除" (Delete). Underneath the toolbar is another row of buttons: "启动" (Start), "中止" (Stop), "结束" (End), and "返回" (Return). The main area contains a scrollable grid table with the following data:

销售编号	项目名称	客户名称	联系人	项目类型	销售人员	状态	目前进度
2UM01	公交车候车厅广告	安徽江淮扬天汽	老张	公交站台广告	张新	启动	40%
PP01	在楼顶做大牌广告	安徽喜梦床具有	李洁	楼顶大牌	张新	启动	10%
XX01	在南谯路做候车厅广告	上海大众汽车源	王新	公交站台广告	赫赫	启动	20%

图 18 商业机会信息管理窗体

3.4.3. 设计商业机会查看管理窗体

在客户单位信息管理窗体中，选择一个客户单位记录，然后再单击“查看销售机会”按钮，将打开商业机会信息查看窗体。商业机会信息查看管理窗体的名称为 FrmSaleView，窗体布局如图 19 所示：

The screenshot shows a Windows application window titled "查看客户商业机会". At the top left is a dropdown menu icon. On the right side are standard window controls: a close button (X), a minimize button, and a maximize button. Below the title bar is a toolbar with a single button: "返回" (Return). The main area contains a scrollable grid table with the following data:

客户单位名称	安徽江淮扬天汽车股份有限公司								
商业机会编号	项目名称	客户名称	联系人	项目类型	销售人员	状态	目前进度	签约日期	
2UM01	公交车候车厅广告	安徽江淮扬天汽	老张	公交站台广告	张新	启动	40%	2010-4-8	50

图 18 商业机会查看管理窗体

3.5. 用户管理模块设计

用户管理模块包括用户管理和修改密码等功能。

3.5.1. 用户管理窗体

用户管理窗体的名称为 FrmUserMan，窗体布局如图 20 所示：

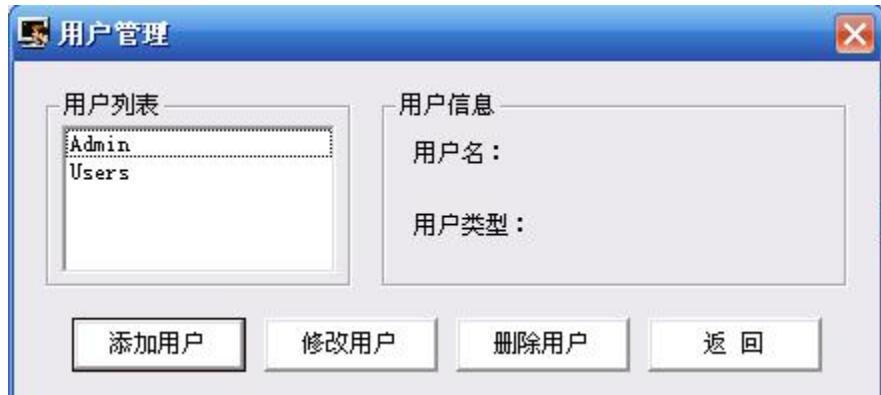


图 20 用户管理窗体

3.5.2. 编辑用户信息窗体

编辑用户信息窗体的名称为 FrmUserEdit，窗体布局如图 21 所示：



图 21 编辑用户信息窗体

4. 客户管理系统的维护与调试

4.1. 测试环境

操作系统：Window XP Professional

数据库：Microsoft Access 2000

开发工具平台：Microsoft Visual Basic 6.0

4.2. 软件测试的目的和原则

在开发大型软件系统的过程中，需要面对错综复杂的问题，因此，在软件生存周期的每个阶段都不可避免地会产生错误。我们力求在每个阶段结束之前通过严格的技术审查，尽可能早的发现并纠正错误。测试的目的就是在软件投入生产性运行之前，尽可能多地发现软件中的错误。目前软件测试仍然是保证软件质量的关键步骤，它是对软件规格说明、设计和编码的最后复审。

尽管面向对象技术的基本思想保证了软件应该有更高的质量，但无论采用什么样的编程技术，编程人员的错误都是不可避免的，而且由于面向对象技术开发的软件代码重用率高，更需要严格测试，避免错误的繁衍。因此，软件测试在面向对象编程里更具有它的重要性。

与开发过程类似，测试过程也必须分步骤进行，每个步骤在逻辑上是前一个步骤地继续。大型软件系统通常由若干个子系统组成，每个子系统又由许多模块组成。因此，大型软件系统的测试的基本层次如图所示。

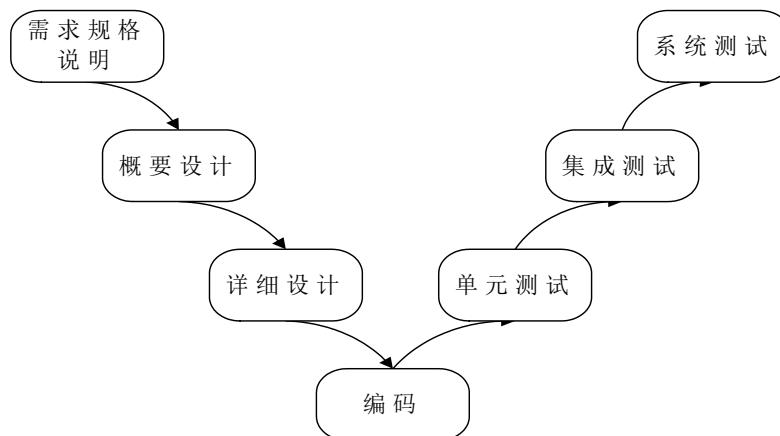


图 20 测试的基本层次

(1) 单元测试 单元测试的用例从单元详细设计中导出。在单元测试中可以采用功能性测试和结构性测试两种。

(2) 集成测试 在这个测试步骤中所发现的往往是概要设计的错误。

(3) 系统测试 在这个测试步骤中所发现的往往是需求规格说明的错误。一般来说，系统测试是功能性测试，不是结构性测试。

4.3 子系统测试

在客户资源管理系统的开发过程中，每开发完一个模块都会进行模块功能测试，由于系统还没有最后完成，现在暂时还不能进行集成和系统测试，这部分将在系统完成后才能进行。软件测试和调试过程中，系统显示出数据冗余度少、数据一致性强。系统功能设计符合用户要求。

由于本人自身能力的局限性，所以做编写的代码，即使经过反复检查也难免出错所以在本阶段力求使用有限的时间找出尽可能多的错误，力求系统尽量正确。我们在本系统的测试中使用了黑盒法（即不关心程序内部的逻辑结构，而是根据程序的功能来设计是检测）请一位不熟悉本系统的人来进行随意性的操作，打破习惯的操作顺序，从中发现错误，在此阶段系统的大量错误得到了改正。

装
订
线

结 论

(1) 系统存在的不足与改进方案

本系统是本人参于 MIS 设计的第一个项目，由于经验的不足和相关知识的欠缺，导致系统开发中有些方面设计得不够周全。比如在数据库建模之初，由于急于开发，对建模这一块就没有很认真地设计，导致系统开发的过程中经常回过头来修改数据库，浪费了很多的时间。现在的系统还有一些数据的显示不是很符合实际，这要在数据窗口中进行统一设置。我们会对系统不断进行完善和升级，力求将不足之处一一改正过来。

(2) 毕业设计心得与收获

通过参与一个实际项目的设计与开发，我学习到了好多从课本中学不到的知识，深刻理解了理论联系实际的客观意义，也明白了搞软件开发不是一件孤立的事情，从系统的设计到最终的成品推出，每一个步骤都需要我们严谨对待，同时也认识到对这一类专业软件业务的熟悉是软件最终能否在实际中使用的关键。这次毕业设计培养了我注重实际、勇于创新的良好思维能力，为以后的工作起到了很好的铺垫作用。

装
订
线

致 谢

在论文的选题及其后的设计过程中得到指导老师的悉心指导，使本文得以顺利完成，在此向赵老师表示衷心的感谢。本论文是在赵老师的指导下修改完成的。在此，要对她们的细心帮助和指导表示由衷的感谢。在这段时间里，我从她们身上不仅学到了许多的专业知识，更感受到了她们工作中的兢兢业业，生活中的平易近人。此外，她们严谨的治学态度和忘我的工作精神值得我去学习。

感谢给予我帮助的其他人员，他们在我的毕业设计过程中经常给我提出许多关键性的问题，使我受益匪浅。

感谢多年来传授我知识的老师们，更要感谢我的家人对我学业上的支持和鼓励，感谢所有关心帮助过我的人。

在此我要特别感谢我的指导老师对我的精心指导和帮助，感谢本次答辩过程中在座的各位老师。你们的言传身教将使我终生受益！感谢所有帮助过我的人！

装
订
线

主要参考文献

- (1)马力,《Visual Basic 程序设计院》,北京大学出版社, 2004-8-1
- (2)冯硕, 费志博,《Visual Basic 数据库编程宝典》, 电子工业出版社, 2001-5-1
- (3)王守茂,《管理信息系统的分析与设计》, 天津科技翻译出版公司, 1993 年
- (4)曹青、邱李华、郭志强,《Visual Basic 程序设计教程》,机械工业出版社, 2002 年
- (5)姜旭平,《信息系统开发方法》, 清华大学出版社, 2002 年
- (6)王珊,《数据庫系统概述》, 高等教育出版社, 1999 年
- (7)姚达,《数据庫设计》, 高等教育出版社, 2000 年
- (8)王珊、陈红,《数据库系统原理教程》, 清华大学出版社, 1999 年
- (9)Korth, HenryF. and Silberschatz, Abraha,《Database System Concepts》, New York : McGraw-Hill, Inc. 1999.3
- (10)Patrick O' Neil Elizabeth O' Neil,《DATABASE Principles, Programming, and Performance》(Second Edition), New York:Higher Education Press Morgan Kaufmann Publishers, 2001.5

装
订
线

附录：部分窗体源代码

(1) 地域管理窗体代码

```
Private Sub Cmd_Add_Click()
    If CurArea.AreaType >= 4 Then
        MsgBox "此地域已不能再细分"
        Exit Sub
    End If
    ' 打开编辑窗体
    FrmAreaEdit.lblUpper = CurArea.AreaName
    FrmAreaEdit.txtArea = ""
    FrmAreaEdit.Modify = False
    FrmAreaEdit.Show 1
    TreeView1_Click
End Sub

Private Sub Cmd_Back_Click()
    Unload Me
End Sub

Private Sub Cmd_Del_Click()
    Dim TmpOrg As String

    If CurArea.AreaType = 0 Then
        MsgBox "此项不能删除"
        Exit Sub
    End If
    If CurArea.HaveSon(CurArea.AreaId) = True Then
        MsgBox CurArea.AreaName + " 包含下级地域，不能删除"
        Exit Sub
    End If
    TmpOrg = MyCust.Area_In_Customer(TmpAreaId)
    If TmpOrg <> "" Then
        MsgBox "地域中包含客户信息- " + TmpOrg + "，不允许删除"
        Exit Sub
    End If
    ' 确认删除
End Sub
```

装
订
线

```
If MsgBox("是否确定要删除 " + Trim(TreeView1.SelectedItem.Text), vbYesNo, "请确认") = vbNo Then
    Exit Sub
End If
CurArea.Delete (CurArea.AreaId)

TreeView1.Nodes.Remove TreeView1.SelectedItem.Index
TreeView1_Click
End Sub

Private Sub Cmd_Modi_Click()
If TreeView1.SelectedItem.Key = "a0" Then
    MsgBox "此项不能修改"
    Exit Sub
End If
FrmAreaEdit.lblUpper = MyArea.GetName(CurArea.UpperId)
FrmAreaEdit.txtArea = CurArea.AreaName
FrmAreaEdit.Modify = True
FrmAreaEdit.Show 1
TreeView1_Click
End Sub

Private Sub Form_Load()
FocusAreaName = "中国"
'添加根结点
Set TmpNode = TreeView1.Nodes.Add(, , "a0", "全球地域", 1, 3)
TmpNode.Selected = True
TmpNode.ExpandedImage = 2

Call Add_AreaToTree(TreeView1)
TreeView1_Click
End Sub

Private Sub TreeView1_Click()
Dim TmpType As Integer
'根据关键字获取地域信息
FocusKey = TreeView1.SelectedItem.Key
CurArea.AreaName = TreeView1.SelectedItem.Text
TmpType = Asc(Left(TreeView1.SelectedItem.Key, 1))
CurArea.AreaType = TmpType - Asc("a")
```

CurArea.AreaId = Val(Right(TreeView1.SelectedItem.Key, Len(TreeView1.SelectedItem.Key) - 1))
'如果是根结点，则不需要再处理
If CurArea.AreaType = 0 Then
 Exit Sub
End If
'决定结点显示的图像
If CurArea.HaveSon(CurArea.AreaId) = False Then
 TreeView1.SelectedItem.ExpandedImage = 4
 TreeView1.SelectedItem.Image = 4
 TreeView1.SelectedItem.SelectedImage = 5
End If
'将地域信息读取到CurArea对象中
CurArea.GetInfo(CurArea.AreaId)
End Sub

(2) 行业分类设置窗体代码

```
Private OriId As Long          '类型编号
Private OriTypeName As String   '原来的类型名称
'类型分类编号，1-客户类型；2-行业分类；3-职务类型；4-项目类型
Public nFlag As Integer
Private Sub Cmd_Add_Click()
    '添加课程类型名称，先判断域是否为空
    If Len(Trim(txtName)) = 0 Then
        MsgBox("请输入类型名称")
        txtName.SetFocus
        Exit Sub
    End If
    '判断数据库中是否已经存在此类型名称
    If MyType.In_DB(Trim(txtName), nFlag) = True Then
        MsgBox("已经存在此类型名称")
        txtName.SetFocus
        Exit Sub
    End If
    '插入新记录
    With MyType
        .TypeName = Trim(txtName)
        .Flag = nFlag
    End With
End Sub
```

```
. Insert
    MsgBox "添加成功"
End With
DataRefresh
End Sub

Private Sub Cmd_Del_Click()
    If Len(Trim(DataList1.BoundText)) = 0 Then
        MsgBox ("请选择记录")
        Exit Sub
    End If
    ' 确定删除
    If MsgBox("是否确定要删除 ", vbYesNo, "请确认") = vbNo Then
        Exit Sub
    End If
    ' 删除类型
    MyType.Delete (Val(DataList1.BoundText))
    MsgBox "删除成功"
    DataRefresh ' 刷新 DataList1
End Sub

Private Sub Cmd_Exit_Click()
    Unload Me
End Sub

Private Sub Cmd_Modi_Click()
    ' 修改类型名称
    If Len(Trim(DataList1.BoundText)) = 0 Then
        MsgBox ("请选择记录")
        Exit Sub
    End If
    ' 判断新的名称是否和原来的相同
    If Trim(txtName) = OriTypeName Then
        Exit Sub
    End If
    ' 判断新的名称是否已经存在
    If MyType.In_DB(Trim(txtName), nTypeId) = True Then
```

```
    MsgBox ("已经存在此类型名称")
    txtName.SetFocus
    Exit Sub
End If
' 更新名称
MyType.TypeName = Trim(txtName)
MyType.Update (OriId)
MsgBox "修改成功"
DataRefresh ' 刷新类型列表
End Sub
```

```
Private Sub DataList1_Click()
    ' 点击列表记录，将编号和名称放入变量和 txtName 域
    If Len(Trim(DataList1.BoundText)) = 0 Then
        MsgBox ("请选择记录")
        Exit Sub
    End If
    OriId = Val(DataList1.BoundText)      ' 记录编号
    If MyType.GetInfo(OriId) = True Then ' 判断是否存在此记录信息
        OriTypeName = Trim(MyType.TypeName) ' 将类型名称保存到变量中
        txtName = Trim(MyType.TypeName)     ' 文本框中显示类型名称
    End If
End Sub
```

```
Private Sub Form_Load()
    ' 刷新 DataList1 中的数据
    DataRefresh
End Sub
```

```
Private Sub DataRefresh()
    ' 设置连接字符串
    AdoType.ConnectionString = Conn
    ' 设置 SQL 语句
    AdoType.RecordSource = "SELECT * FROM Types WHERE Flag=" + Trim(nFlag)
    AdoType.Refresh
    ' 设置数据源
    Set DataList1.DataSource = AdoType
```

```
DataList1.ListField = "TypeName"    ' 设置列表中显示字段  
DataList1.BoundColumn = "Id"      ' 设置列表绑定字段  
DataList1.Refresh  
End Sub
```

(3) 客户基本信息管理窗体

```
Private Sub Refresh_Customer()  
Dim TmpSource As String  
If CurArea.AreaId <= 0 And Check1.Value = 0 Then  
    lblSel = 0  
    Exit Sub  
End If
```

’ 计算全部单位总数

```
total_num = MyCust.CountCustomer(0)  
lblTotal = Trim(Str(total_num))
```

```
TmpSource = "SELECT c.CustId, c.CustName as 单位名称, a.AreaName as 地域名  
称," _  
        + "t.TypeName as 客户类型, t1.TypeName AS 所属行业, c.Address as  
通信地址," _  
        + "c.Postcode as 邮政编码, a.AreaId, c.EmpId, e.EmpName AS 销售人  
员" _  
        + "FROM Customer c,Employees e," _  
        + "Area a,Types t,Types t1 WHERE c.CustTypeId = t.Id And " _  
        + "c.BussTypeId=t1.Id AND c.EmpId=e.EmpId"
```

’ 未选择地域

```
If Check1.Value = 1 Then
```

’ 全部类别

```
If Len(Trim(dco_CType.Text)) = 0 Then
```

```
    TmpSource = TmpSource + " And c.AreaId=a.AreaId ORDER BY c.CustName"
```

’ 选择类别

```
Else
```

```
    TmpSource = TmpSource + " And c.AreaId=a.AreaId And c.CustTypeId=" _  
        + Trim(dco_CType.BoundText) + " ORDER BY c.CustName"
```

```
End If
```

```
Else
```

```
    If CurArea.AreaType = 4 Then    ’ 市
```

装
订
线

```
    ' 全部类别
    If Len(Trim(dco_CType.Text)) = 0 Then
        TmpSource = TmpSource + " And c.AreaId=a.AreaId And c.AreaId=" +
                    + Trim(Str(CurArea.AreaId)) + " ORDER BY c.CustName"
    Else
        TmpSource = TmpSource + " And c.AreaId=a.AreaId And c.AreaId=" +
                    + Trim(Str(CurArea.AreaId)) + " And c.CustTypeId=" +
                    + Trim(dco_CType.BoundText) +
                    + " ORDER BY c.CustName"
    End If
Else      ' 省
    If Len(Trim(dco_CType.Text)) = 0 Then
        TmpSource = TmpSource + " And (c.AreaId=a.AreaId And a.UpperId=" +
                    + Trim(Str(CurArea.AreaId)) + " Or c.AreaId=a.AreaId And
a.AreaId=" +
                    + Trim(Str(CurArea.AreaId)) + ") ORDER BY c.CustName"
    Else
        TmpSource = TmpSource + " And (c.AreaId=a.AreaId And a.UpperId=" +
                    + Trim(Str(CurArea.AreaId)) + " Or c.AreaId=a.AreaId And
a.AreaId=" +
                    + Trim(Str(CurArea.AreaId)) + ") And c.CustTypeId=" +
                    + Trim(dco_CType.BoundText) +
                    + " ORDER BY c.CustName"
    End If
End If
End If
Adodc1.ConnectionString = Conn
Adodc1.RecordSource = TmpSource
Adodc1.Refresh
Set DataGrid1.DataSource = Adodc1
lblSel = Trim(Str(Adodc1.Recordset.RecordCount))
DataGrid1.Columns(0).Width = 0
DataGrid1.Columns(1).Width = 2200
DataGrid1.Columns(2).Width = 1400
DataGrid1.Columns(3).Width = 1400
DataGrid1.Columns(4).Width = 1400
DataGrid1.Columns(5).Width = 1600
```

```
DataGrid1.Columns(6).Width = 1200  
DataGrid1.Columns(7).Width = 0  
DataGrid1.Columns(8).Width = 0  
DataGrid1.Columns(9).Width = 1200  
End Sub
```

```
Private Sub Check1_Click()  
If Check1.Value = 1 Then  
    ComboArea.Enabled = False  
    ComboArea.Text = "全部地域"  
Else  
    ComboArea.Text = MyArea.GetName(CurArea.UpperId) + " " +  
CurArea.AreaName  
    ComboArea.Enabled = True  
End If
```

```
Refresh_Customer  
End Sub
```

```
Private Sub Check2_Click()  
If Check2.Value = 1 Then  
    dco_CType.Enabled = False  
    dco_CType.Text = "全部客户类型"  
Else  
    dco_CType.Enabled = True  
End If
```

```
Refresh_Customer  
End Sub
```

```
Private Sub Cmd_Add_Click()  
    ' 设置添加信息  
    FrmCustomerEdit.Modify = False  
    FrmCustomerEdit.ComboArea.Text = ""  
    FrmCustomerEdit.Show 1  
    Refresh_Customer  
End Sub
```

```
Private Sub Cmd_AddCt_Click()
    If Adodc1.Recordset.BOF = True Then
        MsgBox "请选择记录"
        Exit Sub
    End If
    FrmContactEdit.Modify = False
    FrmContactEdit.OriCustId = Adodc1.Recordset.Fields(0)
    FrmContactEdit.lbl_CustName = Trim(Adodc1.Recordset.Fields(1))
    FrmContactEdit.ComboSex.ListIndex = 1
    FrmContactEdit.Show 1
End Sub
```

```
Private Sub Cmd_Back_Click()
    Unload Me
End Sub
```

```
Private Sub Cmd_Del_Click()
    If Adodc1.Recordset.BOF = True Then
        MsgBox "请选择记录"
        Exit Sub
    End If
    '是否存在此客户的商业机会
    If MyChance.HaveCustomer(Adodc1.Recordset.Fields(0)) = True Then
        MsgBox "商业机会中存在此客户信息，不能删除！"
        Exit Sub
    End If
    '确认删除
    If MsgBox("是否确定要删除客户信息，删除时同时删除客户联系人信息", vbYesNo, "请确认") = vbNo Then
        Exit Sub
    End If
    '删除客户信息
    Call MyCust.Delete(Adodc1.Recordset.Fields(0))
    '删除对应的联系人信息
    Call MyContact.DeleteByCustId(Adodc1.Recordset.Fields(0))
    Refresh_Customer
End Sub
```

装
订
线

```
Private Sub Cmd_Modify_Click()
    Dim TmpAreaId As Long
    If Adodc1.Recordset.BOF = True Then
        MsgBox "请选择记录"
        Exit Sub
    End If
    ' 客户单位
    FrmCustomerEdit.OriCustomerId = Adodc1.Recordset.Fields(0)
    FrmCustomerEdit.txtName = Trim(Adodc1.Recordset.Fields(1))
    ' 地域
    FrmCustomerEdit.ComboArea.Text = Trim(Adodc1.Recordset.Fields(2))
    ' 类别
    FrmCustomerEdit.dco_CType.Text = Trim(Adodc1.Recordset.Fields(3))
    ' 所属行业
    FrmCustomerEdit.dco_BType.Text = Trim(Adodc1.Recordset.Fields(4))
    ' 通信地址
    FrmCustomerEdit.txtAddr = Trim(Adodc1.Recordset.Fields(5))
    ' 邮政编码
    FrmCustomerEdit.txtCode = Trim(Adodc1.Recordset.Fields(6))
    ' 销售人员
    FrmCustomerEdit.dco_Emps.Text = Trim(Adodc1.Recordset.Fields(9))
    ' 保存 CurArea 对象
    TmpAreaId = CurArea.AreaId
    CurArea.AreaId = Adodc1.Recordset.Fields(7)
    FrmCustomerEdit.Modify = True
    FrmCustomerEdit.Show 1
    ' 恢复 CurArea 的值
    CurArea.AreaId = TmpAreaId
    CurArea.GetInfo (CurArea.AreaId)
    ' 刷新数据
    Refresh_Customer
End Sub
```

装
订
线

```
Private Sub Cmd_ViewCt_Click()
    If Adodc1.Recordset.BOF = True Then
        MsgBox "请选择记录"
        Exit Sub
    End If
    FrmContactView.SchId = Adodc1.Recordset.Fields(0)
    FrmContactView.lbl_CustName = Trim(Adodc1.Recordset.Fields(1))
    FrmContactView.Show 1
End Sub
```

```
Private Sub Cmd_ViewSC_Click()
    If Adodc1.Recordset.BOF = True Then
        MsgBox "请选择记录"
        Exit Sub
    End If
    FrmSaleView.SchId = Adodc1.Recordset.Fields(0)
    FrmSaleView.lbl_CustName = Trim(Adodc1.Recordset.Fields(1))
    FrmSaleView.Show 1
End Sub
```

```
Private Sub ComboArea_GotFocus()
    Check1.SetFocus
    ' 设置显示菜单的位置
    FrmAreaSel.AreaLevel = 6
    FrmAreaSel.Left = Me.Left + ComboArea.Left + 350
    FrmAreaSel.Top = Me.Top + ComboArea.Top + ComboArea.Height + 450
    FrmAreaSel.Show 1

    If FrmAreaSel.ClickOK = True Then
        CurArea.GetInfo (CurArea.AreaId)
        ComboArea.Text = MyArea.GetName(CurArea.UpperId) + " " +
        CurArea.AreaName
    End If
    Refresh_Customer
End Sub
```

```
Private Sub dco_CType_Click(Area As Integer)
    Refresh_Customer
End Sub
```

```
Private Sub Form_Load()
    CurArea. Init
    '装入客户类别
    AdoType. ConnectionString = Conn
    AdoType. RecordSource = "SELECT * FROM Types WHERE Flag=1"
    AdoType. Refresh
    Set dco_CType. RowSource = AdoType
    dco_CType. ListField = "TypeName"
    dco_CType. BoundColumn = "Id"
    dco_CType. Enabled = False
    Check2. Value = 1
    Check1. Value = 1
    Refresh_Customer
End Sub
```

装
订
线