

**NEAR EAST UNIVERSITY**



**Faculty Of Engineering**

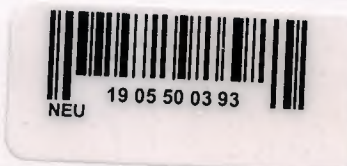
**Department Of Computer Engineering**

**CUSTOMER TRACKING & ACCOUNTING  
PROGRAM USING VISUAL BASIC  
PROGRAMMING LANGUAGE**

**Graduating Project  
COM 400**

**Student: Turgay Aksoy(20010256)**

**Supervisor : Assoc.Prof.Dr. Umid ilhan**



**Nicosia 2007**



## ACKNOWLEDGEMENTS

*"firstly I would like to thank my supervisor Mr. Umid Ilhan for his greatness in his way of treatment with students, He never make us feeling shy to ask any question whenever we faced an obstacle or difficulty not just during this project but during our whole life in Near East University.*

*We have not to forget other Instructors, Especially Mr. Okan Donangil, And Mr. Kaan Uyar, whenever I cross any problem morally or educationally they helped me, I thank them very much with my all for his advices and guiding.*

*Secondly I would like to give my best regards to my family, however their financial situation was not good at first two year but they resisted and forced me to continue my studding, they never made me need to anyone.*

*Thirdly I have not to forget my dear friends Ercan Kasapoğlu, Mücahit C. Bozkurt and Yusuf Alak, they helped me in my project, either by giving their thoughts or with working with me.*

*Finally I would like to thank All my friends who aid me during my university life, to resist and study."*

## ABSTRACT

Customer Tracking & Accounting Program is written to a private company with their wish so this program is not a useful program in general. By using this program the company can control all their customers and their operations, sales and buys in details, payments... etc. of their workplace.

I used Visual basic programming language and Access Data Base in this program, the program is easy in use, everyone can use it easily without getting difficulty. The program shows you all the antennas in stock that you have, you can make sales, enter payments record, add customers, add services served by the company query information's of any customer ... etc.

Also when you add antenna to your stock list you can add by special page for adding new stocks, or you can increase and decrease the quantity or change the price of the antennas existing in stock.

I used many forms in this project, the main form is the MDIForm in which the menu takes place, the program records everything, we can query all the information's at any time, the user can add antenna to stock and search antenna from the stock, can include services to the data base, can add customer and query the customer from the data base and can control the incomes and expenses with this program.

So using this kind of programs make easy to tracking the customers and can take statistics of the past time and at present time and can see if the company is earning many or loses money and depends on that the manager can take precautions to make profit.

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT.....</b>	<b>i</b>
<b>ABSTRACT.....</b>	<b>ii</b>
<b>TABLE OF CONTENTS.....</b>	<b>iii</b>
<b>LIST OF FIGURES.....</b>	<b>v</b>
<b>INTRODUCTION .....</b>	<b>vi</b>
<b>CHAPTER I: VISUAL BASIC.....</b>	<b>01</b>
1.1.What is Visual Basic?.....	01
1.2.The Advantages and Disadvantages of Visual Basic.....	01
1.2.1.The Advantages of Visual Basic.....	01
1.2.2. The Disadvantages of Visual Basic.....	02
1.3.Significant Language Features.....	02
1.4.Areas of Application.....	03
1.5.The Development Environment.....	04
1.6.The Project Explore Window.....	05
1.6.1.Properties Window.....	05
1.6.2.The Default Layout.....	05
1.6.3. Understanding The Tool Box.....	06
1.6.4. Opening an existing Visual Basic Project.....	06
1.6.5. Opening a new Visual Basic File & Inserting Source code.....	07
1.6.6. Running and Viewing the project in Detail.....	07
1.7.Understanding Events.....	09
1.7.1. What an Event is?.....	09
1.7.2. Click Event.....	10
1.7.3. Writing a Code.....	10
1.7.4. Using the Event GotFocus event.....	10
1.7.5. CheckBoxes.....	11
1.7.6. Option Buttons.....	11
1.7.7. ListBoxes.....	12
1.7.7.1 Add to a Listbox.....	13
1.7.7.2. Removing from Listboxes & Advanced Options.....	14
1.7.8. MsgBoxes.....	14
1.7.9. Opening & Retrieving Information From files.....	16
1.7.10. Storing Information to a file.....	17
1.7.11. Printing Text to the printer.....	17
1.8. Control Arrays.....	18
1.9. Brief Information to the usages of Access data bases.....	18
1.10. Database Object.....	19
1.10.1. RecordSet Object.....	19
1.10.2. Accessing Records.....	19
1.10.3. Searching the RecordSet.....	20
1.10.4. Updating The database.....	21
1.10.5. Deleting And Adding records.....	21
1.10.5.1. Deleting Records.....	21
1.10.5.2. Adding Records.....	22
1.11. Managing Data Types.....	22
1.12. Introduction to VB Functions.....	24
1.13. Database Aplication.....	25



## **CHAPTER II: DATABASE STRUCTURE.....**

- 2.1. Microsoft Access Description.....
- 2.2. Creating a database without using the Database Wizard.....
- 2.3. Tables.....
- 2.4. Primary Key.....
- 2.5. Tables Structure in Database.....
- 2.6. Microsoft Access Database.....
  - 2.6.1. Microsoft Access Database Fundamentals.....
  - 2.6.2. Creating Table.....
- 2.7. Introducing Database.....
- 2.8. Database Keys.....
- 2.9. Working With SQL.....
  - 2.9.1. Data Manipulation Language.....
    - 2.9.1.1. Insert.....
    - 2.9.1.2. Select.....

## **CHAPTER III: CUSTOMER TRACKING AND ACCOUNT PROGRAM USING.....**

- 3.1. Main Page.....
- 3.2. Menu Bar.....
  - 3.2.1. Customers Menu.....
  - 3.2.2. Companies Menu.....
  - 3.2.3. IT-Managers Menu.....
  - 3.2.4. Antennas Menu.....
  - 3.2.5. Services Menu.....
  - 3.2.6. Accounting Menu.....
    - 3.2.6.1. Incomes Menu.....
    - 3.2.6.2. Expenses Menu.....
- 3.3. Add New Customer Page.....
- 3.4. Search Customer Page.....
- 3.5. Add New Company Page.....
- 3.6. Search Company Page.....
- 3.7. Add New IT-Manager Page.....
- 3.8. Search IT-Manager Page .....
- 3.9. Add Antenna Page .....
- 3.10. Search Antenna Page .....
- 3.11. Add New Service Page .....
- 3.12. Search Service Page .....
- 3.13. Add New Income Invoice Page .....
- 3.14. Search Income Invoice Page .....
- 3.15. Revenues Page .....
- 3.16. Add New Expenses Invoice Page .....
- 3.17. Search Expenses Invoice Page .....
- 3.18. Expenses Page .....
- 3.19. Net Income Page .....

## **CONCLUSION.....**

## **REFERENCES.....**

## **APPENDICES.....**

## LIST OF FIGURES & TABLES

Figure 1.1: An Empty Form.....	02
Figure 1.2: A Simple Program.....	03
Figure 1.3: Development Environment.....	04
Figure 1.4: Properties Window.....	05
Figure 1.5: Project of Program in Detail.....	07
Figure 1.6: Form & Load.....	09
Figure 1.7: Click Event Form .....	10
Figure 1.8: CheckBox Example.....	11
Figure 1.9: Background Changing Example.....	12
Figure 1.10: A list on Form.....	13
Figure 1.10.1: Adding to List.....	13
Figure 1.11: Showing Message.....	16
Figure 2.1: Creating A database file.....	27
Figure 2.2: Company table as an Example in Access.....	27
Figure 2.3: Sample Table.....	30
Figure 3.1: Main Page.....	35
Figure 3.2: Menu Bar.....	35
Figure 3.3: Customers Menu.....	35
Figure 3.4: Companies Menu .....	36
Figure 3.5: IT-Managers Menu.....	36
Figure 3.6: Antennas Menu.....	36
Figure 3.7: Services Menu .....	37
Figure 3.8: Accounting Menu.....	37
Figure 3.9: Incomes Menu.....	37
Figure 3.10: Expenses Menu.....	37
Figure 3.11: Add New Customer Page.....	38
Figure 3.12: Search Customer Page .....	38
Figure 3.13: Add New Company Page.....	39
Figure 3.14: Search Company Page.....	39
Figure 3.15: Add New IT-Manager Page .....	39
Figure 3.16: Search IT-Manager Page .....	40
Figure 3.17: Add Antenna Page.....	40
Figure 3.18: Search Antenna Page .....	40
Figure 3.19: Add New Service Page .....	41
Figure 3.20: Search Service Page.....	41
Figure 3.21: Add New Income Invoice Page.....	41
Figure 3.22: Search Income Invoice Page.....	42
Figure 3.23: Revenues Page.....	42
Figure 3.24: Add New Expenses Invoice Page.....	43
Figure 3.25: Search Expenses Invoice Page.....	43
Figure 3.26: Expenses Page.....	44
Figure 3.27: Net Income Page.....	44
Table 1.1: Numeric Data Types.....	22
Table 1.2: Nonnumeric Data Types.....	23
Table 2.1: antens Database Table.....	29
Table 2.2: company Database Table.....	29
Table 2.3: it Database Table.....	29
Table 2.4: service Database Table.....	29

# INTRODUCTION

Customer Tracking & Accounting Program is a useful program for Wireless Internet Providers company, by the way of this program we can arrange everything of the wireless internet serving, if we use the program in a correct way nothing will be confused during any operation that take place in this sector for present time and future time, also we can investigate past time operations too.

By the way of this program we can calculate the net income of the company, see our expenses, investigate what happened daily, list customers, list the stocks, Enter payments, giving every customer a special code, etc.

In the first Chapter I described Visual basic, it's properties, components and some examples, I used Visual Basic Program in my project, because I find it easy and I like it's coding system. Visual basic for applications delivers a competitive advantage for ISV seeking to provide full customization and integration capabilities to customer.

In the Second Chapter I described Database system, I used Microsoft Access Database system in my program with Visual Basic, Advantages of using access is to provide exactly the same options for the problems we write as it does for the problems we selected from a data base. Secondly, the process of writing or selecting problems is almost independent of page layout dictions. Also you can see more details about access and SQL.

Third Chapter is Customer Tracking And Accounting Program, I explained it above, it is the most important Chapter of this project, because I made that program by myself alone by the help of some friends also through investigations from internet.



# CHAPTER I: VISUAL BASIC

## 1.1. What Is Visual Basic?

**VISUAL BASIC** is a high level programming language evolved from the earlier DOS version called BASIC. BASIC means **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. It is a fairly easy programming language to learn. The codes look a bit like English Language. Different software companies produced different version of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA and so on. Programmers have undergone a major change in many years of programming various machines. For example what could be created in minutes with Visual Basic could take days in other languages such as "C" or "Pascal". Visual Basic provides many interesting sets of tools to aid you in building exciting applications. Visual Basic provides these tools to make your life far more easier because all the real hard code is already written for you.

With controls like these you can create many applications which use certain parts of windows. For example, one of the controls could be a button, which we have demonstrated in the "Hello World" program below. First create the control on the screen, then write the code which would be executed once the control button is pressed. With this sort of operation in mind, simple programs would take very little code. Why do it like the poor old "C" programmer who would have to write code to even display a window on the screen, when Visual Basic already has this part written for you.

Even though people tend to say Visual Basic's compiler is far behind the compilers of Pascal and C, it has earned itself the status of a professional programming language, and has almost freed BASIC of the reputation of a children's language. Overall you would class Visual Basic as a Graphics User Interface(GUI). Because as you draw, you write for the program. This must always be remembered in any kind of creation of a Visual Basic program. All in all, VB is the preferred language of many future programmers. If you want to start programming Windows, and don't know how to start, give Visual Basic a shot.

## 1.2. The advantages & disadvantages of Visual Basic.

### 1.2.1. The advantages of Visual Basic:

- 1) It's simple language. Things that may be difficult to program with other language, can be done in Visual Basic very easily.
- 2) Because Visual Basic is so popular, There are many good resources (Books, Web sites, News groups and more) that can help you to learn the language. You can find the answers to your programming problems much more easily than other programming languages.
- 3) You can find many tools (Sharewares and Freewares) on the internet that will spare you some programming time.



For example, if you want to ping a user over the internet in your program, Instead of writing the ping function yourself, you can download a control that does it, and use it in your program.

Compare to other languages, Visual Basic have the widest variety of tools that you can download from the internet and use them in your programs.

### 1.2.2. The disadvantages of Visual Basic:

- 1) Visual Basic is powerful language, but it's not suit for programming really sophisticated games.
- 2) It's much more slower than other langauges.

### 1.3. Significant Language Features.

Visual Basic is not only a programming language, but also a complete graphical development environment. This environment allows users with little programming experience to **quickly** develop useful Microsoft Windows applications which have the ability to use OLE ( Object Linking and Embedding ) objects, such as an Excel spreadsheet. Visual Basic also has the ability to develop programs that can be used as a front end application to a database system, serving as the user interface which collects user input and displays formatted output in a more appealing and useful form than many SQL versions are capable of.

Visual Basic's main selling point is the ease with which it allows the user to create nice looking, graphical programs with little coding by the programmer, unlike many other languages that may take hundreds of lines of programmer keyed code. As the programmer works in the graphical environment, much of the program code is automatically generated by the Visual Basic program. In order to understand how this happens it is necessary to understand the major concepts, objects and tools used by Visual Basic. The main object in Visual Basic is called a **form**. When you open a new project, you will start with a clear form that looks similar to this :

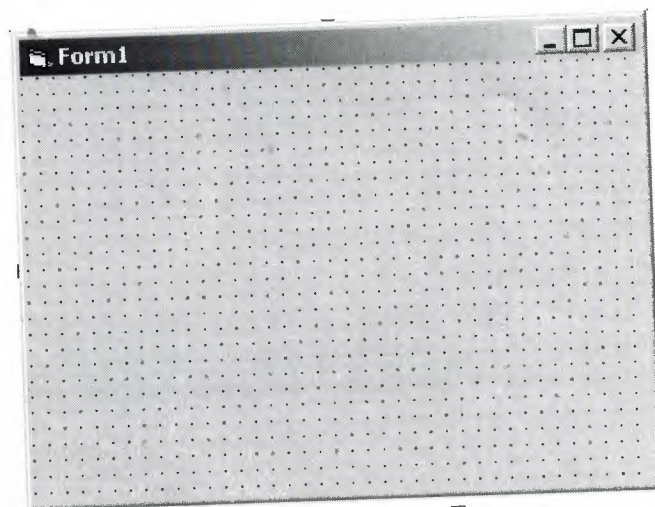


Figure 1.1. An Empty Form

This form will eventually be incorporated into our program as a window. To this form we add **controls**. Controls are things like text boxes, check boxes and command buttons...etc.

We can add **events** to our controls. Events are responses to actions performed on controls. For example, in the "Hello world" program sample on this page, when you click on the command button on our form the event that is triggered is the output of the message "Hello world" to the screen. Code must be written to create an event. we can do this in Visual Basic's **code window**.

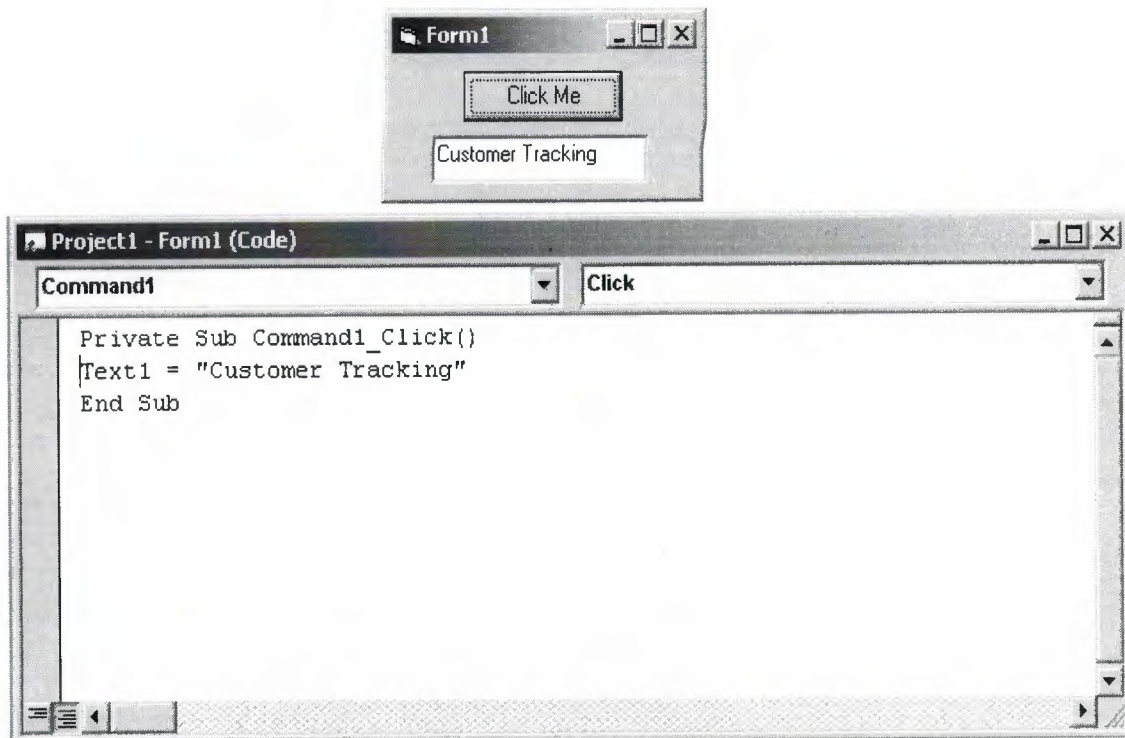


Figure 1.2. A Simple Program

Once the code box is open, you select the object to create an event for and the triggering action ( such as a certain mouse action ) from the drop down menus in the code box. You can open a code box for a particular form by choosing it from the **project window** and selecting the **View Code** button. The project window contains a list of objects associated with that project. Below is an example of a project window :

#### 1.4. Areas of Application.

The term "*Personal Programming*" refers to the idea that, wherever we work, whatever we do, we can expand our computer's usefulness by writing applications to use in our own job. Personal Programming is what Visual Basic is all about.

Using Visual Basic's tools, we quickly translate an abstract idea into a program design, we can actually see on the screen. VB encourages us to experiment, revise, correct, and network your design until the new project meets our requirements. However , most of all, it inspires our imagination and creativity.



Visual Basic is ideal for developing applications that run in the new Windows 95 operating system.

**VB presents a 3-step approach for creating programs:**

1. Design the appearance of application.
2. Assign property settings to the objects of program.
3. Write the code to direct specific tasks at runtime.

**Visual Basic can be used in a number of different areas, for example:**

- Education
- Research
- Medecine
- Business
- Commerce
- Marketing and Sales
- Accounting
- Consulting
- Law
- Science

## 1.5. The Development Environment.

Learning the ins and outs of the Development Environment before we learn visual basic is somewhat like learning for a test, we must know where all the functions belong and what their purpose is. First we will start with labelling the development environment.

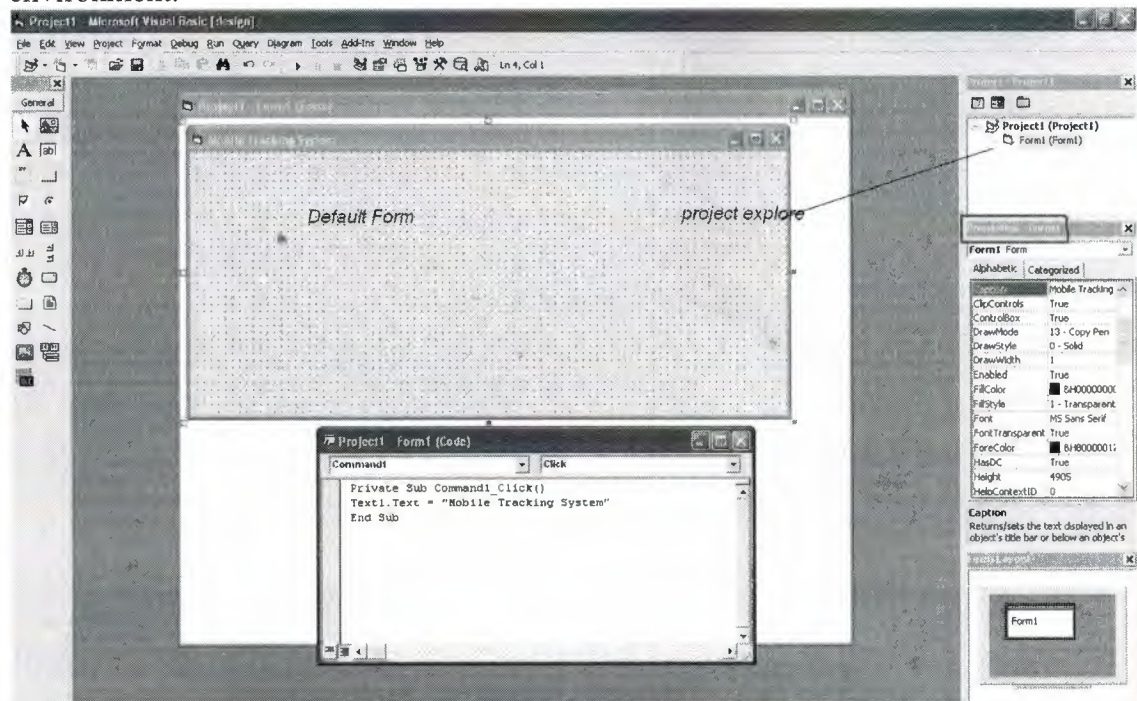


Figure 1.3. Development Environment



The above diagram shows the development environment with all the important points labelled. Many of Visual basic functions work similar to Microsoft word eg the **Tool Bar** and the tool box is similar to other products on the market which work off a single click then drag the width of the object required. The **Tool Box** contains the control we placed on the form window. All of the controls that appear on the **Tool Box** controls on the above picture never runs out of controls as soon as we place one on the form another awaits us on the **Tool Box** ready to be placed as needed.

## 1.6. The project explorer window.

The Project explorer window gives us a tree-structured view of all the files inserted into the application. We can expand these and collapse branches of the views to get more or less detail (**Project explorer**). The project explorer window displays forms, modules or other separators which are supported by the visual basic like class'es and Advanced Modules. If we want to select a form on its own simply double click on the project explorer window for a more detailed look. And it will display it where the **Default form** is located.

### 1.6.1.Properties Window.

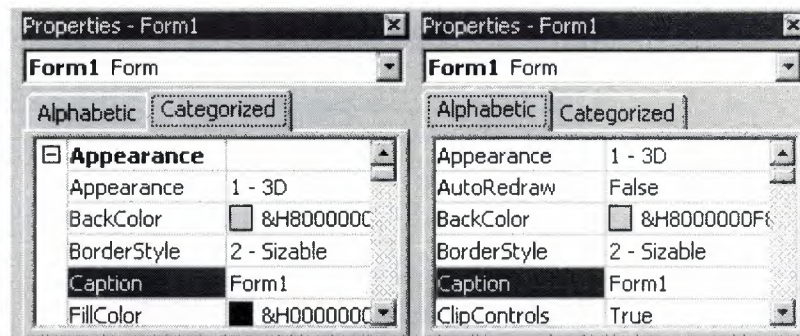


Figure 1.4. Properties Window

Some programmers prefer the Categorized view of the properties window. By defaulting, the properties window displays its properties alphabetically (with the exception of the name value) when we click on the categorized button the window changes to left picture.

### 1.6.2.The Default Layout.

When we start Visual Basic, we are provided with a VB project. A VB project is a collection of the following modules and files.

- The **global module**( *that contains declaration and procedures*)
- The **form module**(*that contains the graphic elements of the VB application along with the instruction* )
- The **general module** (*that generally contains general-purpose instructions not pertaining to anything graphic on-screen*)
- The **class module**(*that contains the defining characteristics of a class, including its properties and methods*)

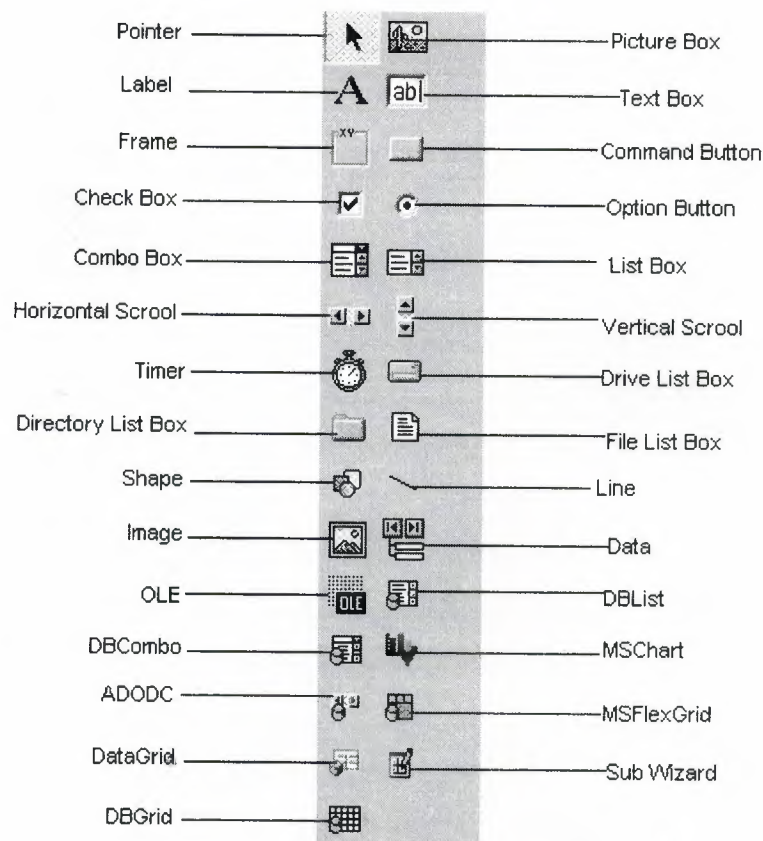
- The **resource files**(that allows you to collect all of the texts and bitmaps for an application in one place)

**On start up, Visual Basic will displays the following windows :**

- The **Blank Form** window
- The **Project** window
- The **Properties** window

It also includes a **Toolbox** that consists of all the controls essential for developing a VB Application. Controls are tools such as boxes, buttons, labels and other objects draw on a form to get **input** or **display output**. They also add visual appeal.

### 1.6.3. Understanding the tool box.



When we click on different controls the Properties Window changes slightly this is due to different controls having different functions. Therefore more options are needed, for example if we had a picture then we want to show an image. But if we wanted to open an internet connection we would have to fill in the remote host and other such settings. When we use the command () we will find that a new set of properties come up the following will provide a description and a property.

### 1.6.4. Opening an existing Visual Basic project.

Microsoft have included some freebies with visual basic to show its capabilities and functions. Dismantling or modifying these sample projects is a good way to

understand what is happening at runtime. These files can be located at our default directory /SAMPLES/

To Open these projects choose 'Open Project' from the 'File' menu. Then Double click on the samples folder to open the directory then Double click on any project to load it.

### 1.6.5. Opening a New Visual Basic File & Inserting Source code.

When we open the program we choose 'New Project' from the 'File' menu. We use the blank form1 to design a simple interface for an estate agents database, have some textboxes for names and other details. We insert some controls and make it look professional. Textboxes can be used to store there name and other details, we make sure that we put a picture box in for a picture of the house.




Later we insert the following source code for application.

#### Private Sub

```
Form_Load()Picture1.Picture=LoadPicture("C:\ProgramFiles\VB\Graphics\Icons\Misc\MISC42.ICO")
```

#### End Sub

### 1.6.6. Running and viewing the project in detail.

Once an application is loaded it can be run by click on the  icon from the toolbar, to pause press  and to terminate use .

Once a project is loaded, the name of the form(s) that it contains is displayed in the project window. To view a form in design mode, we select the form required by clicking with the mouse to highlight its name, then clicking on the view form button.

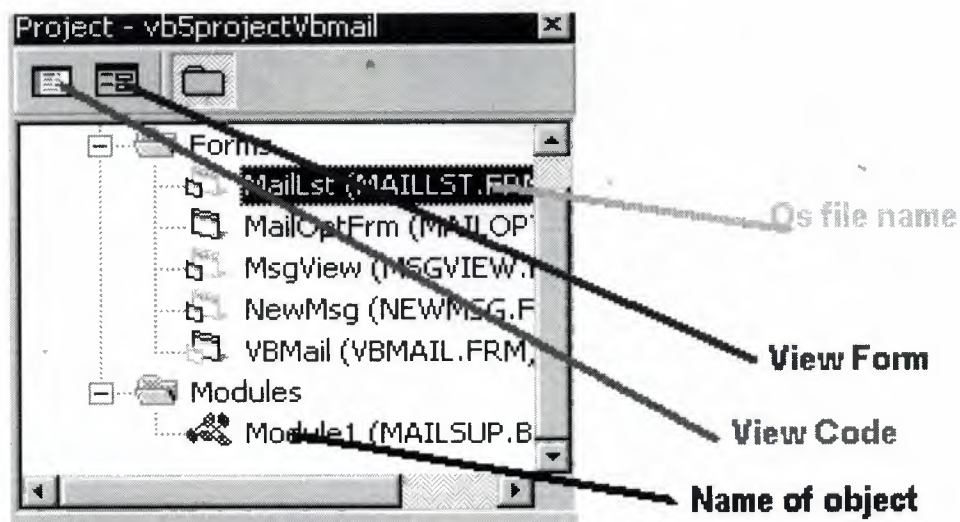


Figure 1.5. Project of the Program in Detail



## Button Properties for reference


 Command Button &  labels properties

Property	Description
<b>Name</b>	The name of the object so you can call it at runtime
<b>BackColor</b>	This specifies the command button's background color. Click the <i>BackColor's</i> palette down arrow to see a list of common Windows control colours, you must change this to the style property from 0 - standard to 1 - graphical
<b>Cancel</b>	Determines whether the command button gets a Click event if the user presses escape
<b>Caption</b>	Holds the text that appears on the command button.
<b>Default</b>	Determines if the command button responds to an enter keypress even if another control has the focus
<b>Enable</b>	Determines whether the command button is active. Often, you'll change the enable property at runtime with code to prevent the user pressing the button
<b>Font</b>	Produces a Font dialog box in which you can set the caption's font name , style and size.
<b>Height</b>	Positions the height of the object - can be used for down
<b>Left</b>	Positions the left control - can be used for right
<b>MousePointer</b>	If selected to an icon can change the picture of the mouse pointer over that object
<b>Picture</b>	Hold's the name of an icon graphic image so that it appears as a picture instead of a <b>Button</b> for this option to work the graphical tag must be set to 1
<b>Style</b>	This determines if the <b>Command Button</b> appears as a standard windows dialog box or a graphical image
<b>Tab index</b>	Specifies the order of the command button in tab order
<b>Tab Stop</b>	Whether the object can be tabbed to ( this can be used in labels which have no other function )
<b>Tool Tip Text</b>	If the mouse is held over the object a brief description can be displayed (for example hold your mouse over one of the above pictures to see this happening
<b>Visible</b>	If you want the user to see the button/label select true other wise just press false
<b>Width</b>	Show the width of the object

## 1.7. Understanding Events.

- Know what an Event is.
- Determine what Events a control can have
- Write code for one or more Events.
- Using optionbuttons to produce an event
- Using checkboxes to produce an event
- Grouping controls using a frame
- Make a simple alteration to the interface, such as changing background colour, at run time.
- Creating a listbox.
- Remove and Add listboxs functions.
- Creating Combo Boxes
- What the different types of combo boxes are.

### 1.7.1. What an event is?

The 'look' of a Visual Basic application is determined by what controls are used, but the 'feel' is determined by the events. An event is something which can happen to a control. For example, a user can click on a button, change a text box, or resize a form. As explained in Creating a Visual Basic Application, writing a program is made up of three events: 1) select suitable controls, 2) set the properties, and 3) write the code. It is at the code writing stage when it becomes important to choose appropriate events for each control. To do this double click on the control the event will be used for, or click on the  icon in the project window (usually top right of screen). A code window should now be displayed similar to the one shown below.

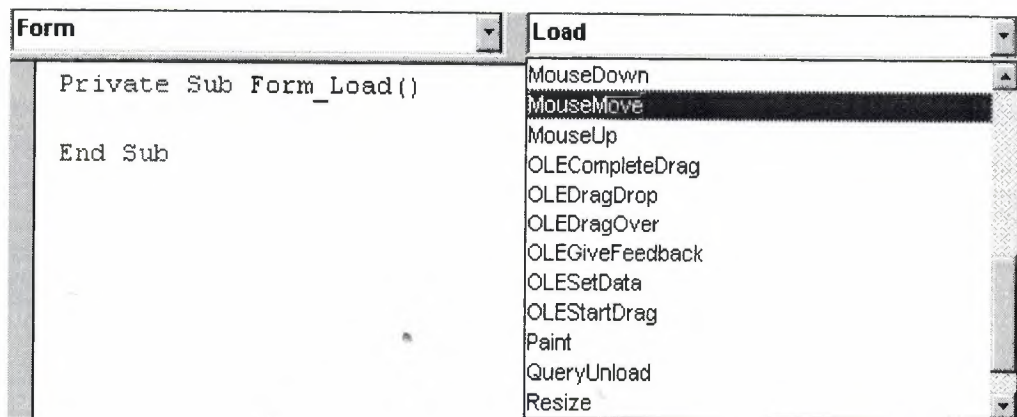


Figure 1.6. Form & Load

The left hand dropdown box provides a list of all controls used by the current form, the form itself, and a special section called General Declarations. The corresponding dropdown box on the right displays a list of all events applicable to the current control (as specified by the left hand dropdown box). Events displayed in bold signify that code has already been written for them, unbold events are unused. To demonstrate that different events can play a significant role in determining the feel of an application, a small example program will be written to add two numbers together and display the answer. The first solution to this problem will use the click event of a command button, while the second will the change event of two text boxes.



### 1.7.2. Click Event.

Before any events can be coded it is necessary to design the interface from suitable controls. As shown in the screen shot below use: 2 text boxes to enter the numbers, a label for the '+' sign, a command button for the '=' sign, and another label for the answer.

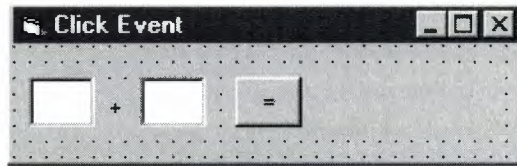
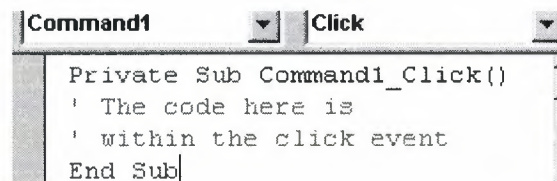


Figure 1.7. Click Event Form

Making the click event is very simple just select the button with the mouse and double click visual basic will generate



You can see on the top right there is a 'click' dropdown list this is known as a event handler.

### 1.7.3. Writing a Code


In the first example the user enter two numbers and then click on the equals button to produce an answer. However, the program can be changed so that the answer will be calculated every time either of the two numbers are changed without requiring an equals button.

```
Private Sub txtNumber1_Change()  
label2.Caption = Str$(Val(text1.Text) + Val(text2.Text))  
End Sub
```

```
Private Sub txtNumber2_Change()  
label2.Caption = Str$(Val(text1.Text) + Val(text2.Text))  
End Sub
```

When we run the program, we enter two numbers and observe what happens. Each time a digit changes the answer is recalculated.

### 1.7.4. Using the event GotFocus event.

So far only one event has been used per control, however this does not have to be the case! Add a StatusBar control to the bottom of the form, bring up the code window using , select the first text box (txtNumber1) from the left hand dropdown box, and then select the **GotFocus** event from the right hand dropdown box. Now some basic instructions can be written in the status bar so that when the cursor is in the text box (the text box has focus) the status bar reads "Enter the first number". After



completing this change to the second text box and using the same GotFocus event change the statusbar text to "Enter a second number". The code to set the status bar can be seen below.

### 1.7. 5. CheckBoxes.

Option bars are used quite often in the windows environment as they can only have two outputs 0 and 1 these get used to process the form. In this example it will be used to change the some text from normal to bold or to italic.

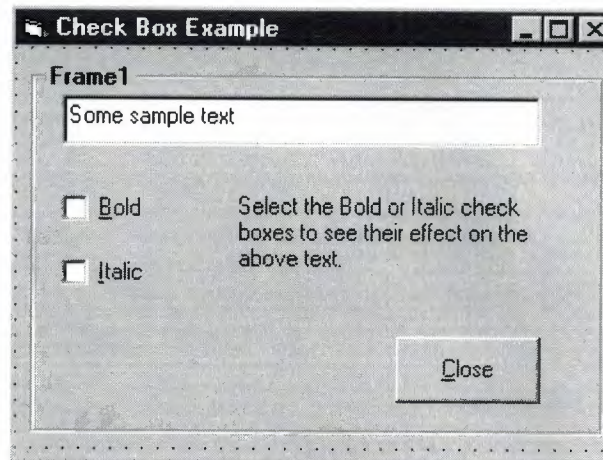


Figure 1.8. Check Box Example

#### **Private Sub chkBold\_Click()**

If chkBold.Value = 1 Then ' If checked.

txtDisplay.FontBold = True

Else ' If not checked.

txtDisplay.FontBold = False

End If

**End Sub**

#### **Private Sub chkItalic\_Click()**

If chkItalic.Value = 1 Then ' If checked.

txtDisplay.FontItalic = True

Else ' If not checked.

txtDisplay.FontItalic = False

End If

**End Sub**

### 1.7.6.Option Buttons.

#### **Changing the background colour**

Changing the background colour gets used mostly by freeware, or the type of programs we generate for use with banners or adverts, anyway it might come in useful sometime. This example shows an ordinary picture of a smiley face then I put in some nice background colours to make it stand out more.

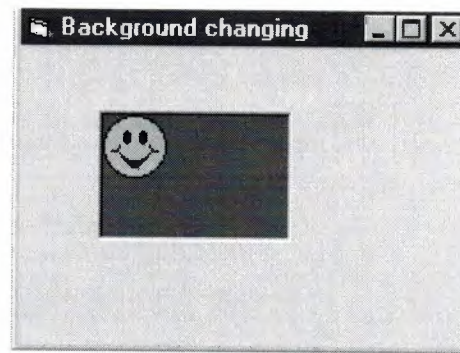


Figure 1.9. Back Ground Changing Example

#### **Private Sub Form\_Load()**

```
BackColor = QBColor(Rnd * 15)
```



```
ForeColor = QBColor(Rnd * 10)
```

```
Picture1.BackColor = QBColor(Rnd * 15)
```

```
Picture1.ForeColor = QBColor(Rnd * 10)
```


```
End Sub
```

#### **1.7.7. List boxes.**

List boxes and combo boxes are used to supply a list of options to the user. The toolbox icons representing these two controls are  for list box and  for combo box.

A list box is used when the user is to be presented with a fixed set of selections (i.e. a choice must be made only from the items displayed, there is no possibility of the user typing in an alternative).

Examples might be offering a list of the days in a week, the available modules in an elective catalogue, the holiday destinations available from a particular airport or the types of treatment offered by a beauty salon.

To create a list box, double click on the toolbox icon . Drag the resulting box into place on the form and size it according to the data it will hold. The left hand picture below shows a list box that has been created and sized on Form1. In the middle is the code that is required to load a selection of cars into the list. The data has been included in the procedure Sub Form\_Load so that it appears when Form1 is loaded. Finally, the picture on the right shows what appears on the form when the application is run. Notice that vertical scroll bars are added automatically if the list box is not deep enough to display all the choices.

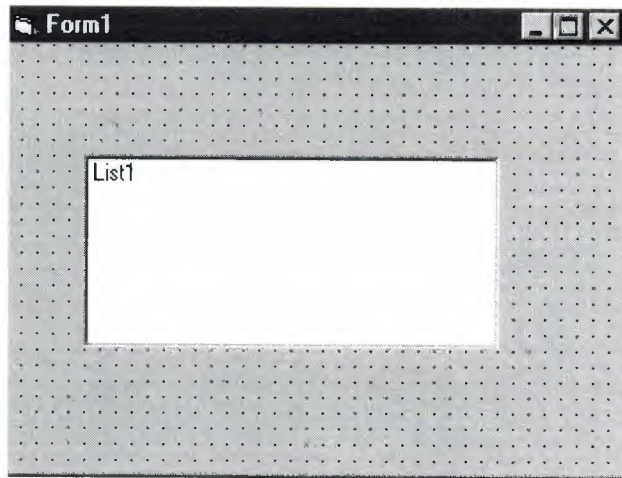


Figure 1.10. A list On Form

If we however add the following source code to this project

#### 1.7.7.1. Add to a Lisbox.

```
Private Sub Form_Load()  
List1.AddItem "Monday"  
List1.AddItem "Tuesday"  
List1.AddItem "Wednesday"  
List1.AddItem "Thursday"  
List1.AddItem "Friday"  
List1.AddItem "Saturday"  
List1.AddItem "Sunday"  
End Sub
```

The source code should look something like this :

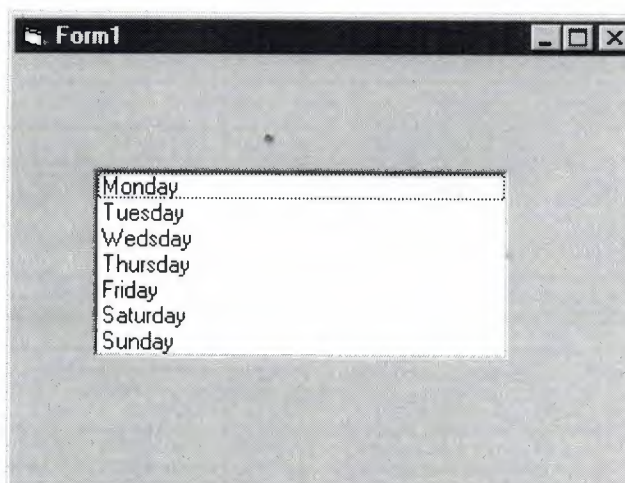


Figure 1.10.1. Adding To List



### 1.7.7.2.Removing & Advanced Options.

They appear in the order they were typed if you changed the properties window by changing sort order = true then they will go into alphabetical order. List items can be added and deleted all the list is counted as the order they are in so for example if you wanted to delete "Tuesday" you would type

#### **list1.RemoveItem 1**

And to add to the list in a particular order just add

#### **list1.additem "My Day", 5**

This will be added after Saturday.

And finally to clear the listbox type

**List1.clear** This will completely clear the contents of the listbox.

The property ListCount stores the number of items in a list, so list1.ListCount can be used to determine the number of items in list box list1.

The property ListIndex gives the index of the currently selected list item. So the statement list1.RemoveItem List1.ListIndex removes the currently highlighted list item.

Adding an item can be accomplished very neatly using an input dialog box. Try this:

#### **list1.AddItem InputBox("Enter a day", "Add a Day")**

This will open a message box and prompt you for a new day to enter this will then be added to the list index at the bottom unless you specify where it should go.

### 1.7.8. Msgboxes.

Message boxes are used when you want to ask the user a question or display an error message(s) and advise the user. There are six types of message boxes here are their functions and what they do. Here is the listing of all the possible msgbox events

The Buttons displayed in a message here Button Layout	Value	Short Description
vbOKOnly	0	Displays the OK button.
vbOKCancel	1	Displays the ok and cancel button.
vbAbortRetryIgnore	2	Displays the Abort , Retry , Ignore
vbYesNoCancel	3	Displays Yes , No and Cancel button
vbYesNo	4	Displays the Yes / No button

vbRetryCancel	5	Displays the retry and Cancel buttons.
---------------	---	--

**The Icons displayed in the message box are here**

Icon on message	Value	Short Description
vbCritical	16	Displays critical message icon
vbQuestion	32	Displays question icon
vbExclamation	48	Displays exclamation icon
vbInformation	64	Displays information icon

<b>The Default button displayed in a message form</b> Default Button	Value	Short Description
vbDefaultButton1	0	Button 1 is default
vbDefaultButton2	256	Button 2 is default
vbDefaultButton3	512	Button 3 is default

Msgbox Return Value	Value	Short Description
vbOk	1	The User Clicked OK
vbCancel	2	The User Clicked Cancel
vbAbort	3	The User Clicked Abort
vbRetry	4	The User Clicked Retry
vbIgnore	5	The User Clicked Ignore
VbYes	6	The User Clicked Yes
VbNo	7	The User Clicked No

The syntax for use of the message box in MsgBox "TEXT", VALUE, "TITLE"  
If you want to use two or more tables just add the values together. Therefore to print a message box to say "The Device was not Found!" OK & Explanation :

Source code 1

**Private Sub Form\_Load()**

MsgBox "The Device was not Found!", 48, "Header"

**End Sub**

Source code 2

**Private Sub Form\_Load()**

```
MsgBox "The Device was not found!", vbExclamation, "Header"
```

**End Sub**

we should get the picture shown below whatever source code you used.

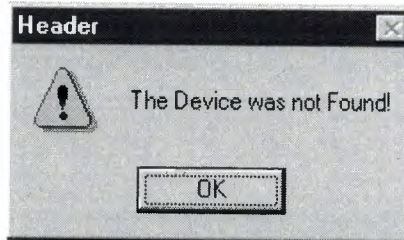


Figure 1.11. Showing message

This is a basic msgbox which in this case has not been processed in any way. The following Source code displays a msgbox that ask you for specific text. For example lets make a password program out of this message box.

**Private Sub Form\_Load()**

```
lngBefore = Timer  
Do  
strAns = InputBox("What is the password Password is Example", "Password Required")  
Loop Until Val(strAns) = Example  
lngAfter = Timer  
msgbox "Correct Password", vbInformation
```

**End Sub**

Once you copy and paste the source code you should get prompted for a password in a different type of msgbox as it includes text. From looking at this example you should be able to see how the loop function works to generate and then check the value. All of the Return Values work in the same way to return an object input.

### 1.7.9.Opening & Retriving information from files.

When applications are loaded they normal get some setting out of the registry or file this section will show you how to retrieve 1 string out of a file.

**Private Sub Form\_Load()**

```
Dim F As Integer, password As String  
F = FreeFile  
Open App.Path & "\password.txt" For Input As F  
Input #F, password
```



Close #F

### **End Sub**

As you can see from this source code the password is previously declared as a string. After this is done be sure to close the file otherwise next time you want to store or read the file the computer will think it is being used by another application and windows will not let you do anything with it. So as you can see it is **Very Important** to close the file

#### **1.7.10. Storing Information to a file**

FTP programs often store information to a file such as a username and password or host information in the same way. This following example will put some information into the file.

### **Private Sub Form\_Load()**

```
Dim F As Integer, pass As String
F = FreeFile
save = txtNew
Open App.Path & "\password.txt" For Output As F
Write #F, Text1.text
Close #F
```

### **End Sub**

Although this is a bit obvious I think I should include it just incase I think differently to other people.

#### **1.7.11. Printing text to the printer.**

This is rather easy to do and it gets used in notepad etc...

### **Private Sub Form\_Load()**

```
Printer.Print " The printer will print this text "
Printer.Print ""
Printer.Print " It will leave a line here"
Printer.Print " It should add the contence of text1.text here : " & Text1.Text & " As you
can see it works"
Printer.Print ""
Printer.EndDoc "This will tell the printer it has finished.
```

### **End Sub**

Everything that appears in position (A) will get printed by the default printer printer.print " A ". The printer enddoc is used to tell the printer the job is finished if this is not added the printer can not estimate how near it will be until it has finish and when it has finished it will think it has'nt so be sure to include this to prevent confusion.

## **1.8. Control Arrays.**

A control array is a list of controls with the same name. Therefore, instead of using five command buttons with separate five names, you can place a command button control array on the form, and that control array holds five command buttons. The control array can have a single name, and you will distinguish the control from each other with a subscript. One of the best reasons to use control array from that first control, all the elements in the control array take on the same property values, You then can change those properties that need to be changed without having to set every property for each control individually. Control arrays have a lot in common with data arrays. A control array has one array, and you distinguish all the array's controls from each other with the zero-based subscript. ( The index property holds the controls subscript number ). All of the control elements must be the same data type. As soon as you place a control on a form that has the same name as an existing control, Visual Basic makes sure you that you want to begin a control array by issuing the warning message to show that the control is already in use. This is used as a built in safety so that you do not over right an existing control by putting it some where else on the same form. If you answer the warning box with a no button, Visual Basic uses a default control name for the placed control.

All event procedures that use control from a control array require a special argument value passed to them that the determines which control is being worked on. For example if your application contains a single control command button named cmdtotal the click () event begins and ends as follows

If however you create a control array named the same name as before (cmdtotal) it will end up like this

## **1.9. Brief introduction to the usages of Access data bases.**

What I think is the most compelling thing about Visual Basic is it's easy way of accessing and modifying databases. This is what I think you should learn next; you will find many applications for this knowledge. I almost never make a program without using a database for data storage.

There are many ways to work with databases in Visual Basic, and I would think you have at least glanced at the Data control. I will not even mention the Data control further in this text, since it is so easy to use and too limited to be interesting for a professional developer. (Ok, there are some exceptions to this.)

What I will teach you to use in this text is DAO (Data Access Objects). You will get familiar with opening a database and retrieving/adding/deleting/updating records from tables. I will only use an Access Database (\*.mdb) in my examples, since this is the most used DBMS (DataBase Management System) for smaller applications made in Visual Basic. We will at the end of this lesson have made a simple, yet functional, phone book application.

## 1.10. Database Object.

The first thing we must do in our application is to open a database where our tables are stored. we need to declare a variable to hold our database in order to do this. This is done with:

```
Dim dbMyDB As Database
```

This gives us a variable/object that can hold a reference to our database. To open a simple Access database named "MyDatabase.mdb", do this:

```
Set dbMyDB = OpenDatabase("MyDatabase.mdb")
```

So, now we have opened a database. This won't give us any data. What we need to do is open a table in the database. We're not limited to open a single table; sometimes we have two or more tables that are related to each other and linked together with foreign keys, and there are ways to handle this to.

### 1.10.1. RecordSet Object.

Visual Basic uses an object called RecordSet to hold our table. To declare such an object and to open the table, we do this:

```
Dim rsMyRS As RecordSet
```

```
Set rsMyRS = dbMyDB.OpenRecordSet("MyTable", dbOpenDynaset)
```

Well, I declared a RecordSet object and used the Database object's OpenRecordSet method to open a table of type Dynaset. We can open a RecordSet in several modes. VB's online help file explains the different modes and what they are for. The Dynaset mode is the mode I use mostly. It gives us a RecordSet that we can add, delete and modify records in.

### 1.10.2. Accessing records.

Now that we have opened a table (referred to as RecordSet from now on) we want to access the records in it. The RecordSet object allows us to move in it by using the methods MoveFirst, MoveNext, MovePrevious, MoveLast (among others). I will use some of these to fill up a list box with the records of our RecordSet.

To get this example to work, we make a database (with Access) called "MyDatabase.mdb" with the table "MyTable" in it. This table should have the fields "ID" of type "Counter" that you set to be the primary key, the field "Name" of type Text and a field "Phone" of type Text. We add some records to it. We put a list box on a form and call it "lstRecords".



```
Dim dbMyDB As Database
Dim rsMyRS As RecordSet
```

#### **Private Sub Form\_Load()**

```
Set dbMyDB = OpenDatabase("MyDatabase.mdb")
Set rsMyRS = dbMyDB.OpenRecordSet("MyTable", dbOpenDynaset)
```

```
If Not rsMyRS.EOF Then rsMyRS.MoveFirst
Do While Not rsMyRS.EOF
    lstRecords.AddItem rsMyRS!Name
    lstRecords.ItemData(lstRecords.NewIndex) = rsMyRS!ID
    rsMyRS.MoveNext
Loop
```

#### **End Sub**

This will make the list box fill up with our records when the form loads. I have introduced some new concepts with this example. We have all ready covered the first part where we open the table. The line that says `If Not rsMyRS.EOF Then rsMyRS.MoveFirst` tells the program to move to the first record in case there are any records at all. The `EOF` is a Boolean property that is true if the current record is the last. It is also true if there are no records in the RecordSet.

Then we make the program add the "Name" field of all records to the list box by adding the current records field "Name" and moving to the next record. You ask for a field of a RecordSet by putting a `!` between the name of the RecordSet object and the name of the field. The while loop checks to see if there are more records to add.

### **1.10.3. Searching the RecordSet.**

I put the value of the field "ID" in the list box's ItemData property. I did this so that we would know the primary key for all the records in order to search for a record.

If we put a text box somewhere on the form and call it "txtPhone". Then copy the following code to the project.

#### **Private Sub lstRecords\_Click()**

```
rsMyRS.FindFirst "ID=" & Str(lstRecords.ItemData(lstRecords.ListIndex))
txtPhone.Text = rsMyRS!Phone
```

#### **End Sub**

This will display the phone number of the selected person when clicking in the list box. It uses the FindFirst method of the RecordSet object. This takes a string parameter that is like what is after WHERE in a SQL expression. We state the field that we want to search in (here "ID"), then the evaluation criteria (here "=") and last the value to search for.

So what we did was to search for the record with the "ID" field value that was the same as the ItemData property of the selected item in the list box. Then we show the value of the "Phone" field in the text box.

#### **1.10.4. Updating the Database.**

We want to be able to update some value of some field when doing database programming. This is done with `Edit` and `Update`. We will try to change the value of the "Phone" field by editing the text in the text box and clicking a button.

If we put a command button on the form and name it "cmdUpdate". Then copy the following code to the project.

##### **Private Sub cmdUpdate\_Click()**

```
rsMyRS.Edit  
rsMyRS!Phone = txtPhone.Text  
rsMyRS.Update
```

##### **End Sub**

This changes the value of the "Phone" field of our current record. Imagine the current record being a set of boxes, with a field in each box. The `Edit` method takes the lid off all of the boxes and `Update` puts them back on. When we write `rsMyRS!Phone = txtPhone.Text` we replace the content of the "Phone" box with the content in the text box.

#### **1.10.5.Deleting and Adding records.**

##### **1.10.5.1. Deleting.**

Deleting records couldn't be simpler. To delete the current record we just invoke the `Delete` method of the RecordSet object. We will put this feature in our little project. If we make one more command button named "cmdDelete" and the following code will do the work of deleting our currently selected person.

##### **Private Sub cmdDelete\_Click()**

```
rsMyRS.Delete  
lstRecords.RemoveItem lstRecords.ListIndex
```

##### **End Sub**

I won't even bother to explain that in greater detail =). The first statement deletes the record and the second removes the list box entry.

### 1.10.5.2. Adding.

Adding records is much like updateing, except that we use AddNew instead of Edit. Let's add one more command button to our application.

#### Private Sub cmdNew\_Click()

```
rsMyRS.AddNew
rsMyRS!Name = "A New Person"
lstRecords.AddItem rsMyRS!Name
lstRecords.ItemData(lstRecords.NewIndex) = rsMyRS!ID
rsMyRS!Phone = "Person's Phone Number"
rsMyRS.Update
```

End Sub

### 1.11. Managing Data Types.

There are many types of data we come across in our daily life. For example, we need to handle data such as names, addresses, money, date, stock quotes, statistics and etc everyday. Similarly in Visual Basic, we are also going to deal with these kinds of data. However, to be more systematic, VB divides data into different types.

#### Numeric Data

Numeric data are data that consists of numbers, which can be computed mathematically with various standard operators such as add, minus, multiply, divide and so on. In Visual Basic, the numeric data are divided into 7 types, they are summarized in Table1.1

Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values 1.401298E-45 to 3.402823E+38 for positive values.
Double	8 bytes	-1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.

Table 1.1: Numeric Data Types

#### Non-numeric Data Types

The nonnumeric data types are summarized in Table1.2



Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False

**Table 1.2: Nonnumeric Data Types**

## Declaring Variables

In Visual Basic, one needs to declare the variables before using them by assigning names and data types. They are normally declared in the general section of the code's windows using the **Dim** statement. The format is as follows:

Dim variableName as DataType

### Example 1.3

```
Dim password As String
Dim yourName As String
Dim firstnum As Integer
Dim secondnum As Integer
Dim total As Integer
Dim doDate As Date
```

You may also combine them in one line, separating each variable with a comma, as follows: Dim password As String, yourName As String, firstnum As Integer,.....

### If....Then.....Else Statements with Operators

To effectively control the VB program flow, we shall use If...Then...Else statement together with the conditional operators and logical operators. The general format for the if...then...else statement is

```
If conditions Then
VB expressions
Else
VB expressions
End If
```

### Select Case

If you have a lot of conditional statements, using If..Then..Else could be very messy. For multiple conditional statements, it is better to use Select Case. The format is:

## **Select Case expression**

Case value1  
    Block of one or more VB statements  
Case value2  
    Block of one or more VB Statements  
End Select

## **Looping**

Visual Basic allows a procedure to be repeated as many times as long as the processor could support. This is generally called looping .

1. **Do**  
    Block of one or more VB statements  
    **Loop While condition**

2. **Do**  
    Block of one or more VB statements  
  
    **Loop Until condition**

### **3. For....Next Loop**

For counter=startNumber to endNumber (Step increment)  
  
One or more VB statements  
  
Next

## **1.12. Introduction to VB Functions.**

### **Creating Functions**

The general format of a function is as follows:

Public Function **functionName** (Arg As dataType,.....) As dataType  
or  
Private Function **functionName** (Arg As dataType,.....) As dataType

\*Public indicates that the function is applicable to the whole program and  
Private indicates that the function is only applicable to a certain module or procedure.

### **Declaring Array**

We could use Public or Dim statement to declare an array just as the way we declare a single variable. The Public statement declares an array that can be used throughout an application while the Dim statement declare an array that could be used only in a local procedure. The general format to declare an array is as follow:

**Dim arrayName(subs) as dataType**

where subs indicates the last subscript in the array.

**Example 13.1**

**Dim FindName(10) as String**

**1.13.Database Application.**

Finally, I want to put end point in that part, with telling about database application. Of course, All topics is not restricted what I emphasize in that part.

Visual basic allows us to manage databases created with different database program such as MS Access, Dbase, Paradox and etc. In this lesson, we are not dealing with how to create database files but we will see how we can access database files in the VB environment

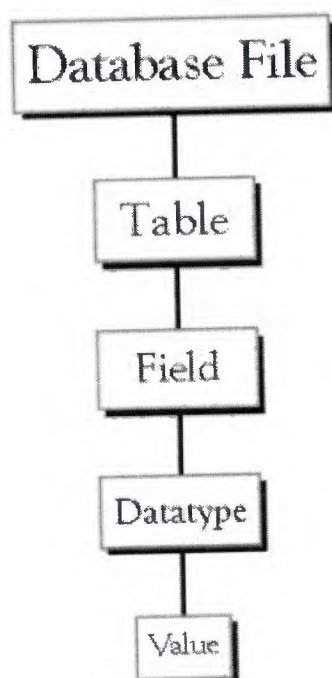


## CHAPTER II: DATABASE STRUCTURE

### 2.1. Microsoft Access Description.

Microsoft Access is a powerful program to create and manage our databases. It has many built in features to assist us in constructing and viewing our information. Access is much more involved and is a more genuine database application than other programs such as Microsoft Works.

First of all we need to understand how Microsoft Access breaks down a database. Some keywords involved in this process are: *Database File, Table, Record, Field, Data-type*. Here is the Hierarchy that Microsoft Access uses in breaking down a



**Database File:** This is our main file that encompasses the entire database and that is saved to our hard-drive or floppy disk.

Example) Mobile.mdb

**Table:** A table is a collection of data about a specific topic. There can be multiple tables in a database.

Example #1) Customer

Example #2) Stock

**Field:** Fields are the different categories within a Table. Tables usually contain multiple fields.

Example #1) stkname

Example #2) stktype

**Datatypes:** Datatypes are the properties of each field. A field only has 1 datatype.

FieldName) username

Datatype) Text

database.

### 2.2. Creating a database without using the Database Wizard.

1. When Microsoft Access first starts up, a dialog box is automatically displayed with options to create a new database or open an existing one. If this dialog box is displayed, we click **Blank Access Database**, and then click **OK**.

If we have already opened a database or closed the dialog box that displays when Microsoft Access starts up, we click **New Database** on the toolbar, and then double-click the **Blank Database** icon on the **General** tab.

2. We specify a name and location for the database and we click **Create**. (Below is the screen that shows up following this step)

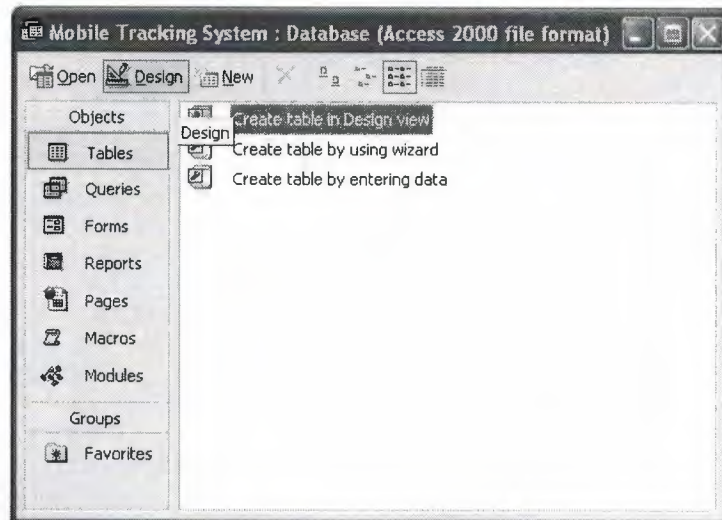


Figure 2.1. Creating A database File

## 2.3. Tables.

A table is a collection of data about a specific topic, such as Mobiles or Customers. Using a separate table for each topic means that you store that data only once, which makes your database more efficient, and reduces data-entry errors.

Tables organize data into columns (called **fields**) and rows (called **records**).

company : Table									
	id	cname	tel	fax	branch	address	email	web	adddate
		AKSOTECH LTI	03922281758	03922281758	1	Yzb. Tekin Yurc	info@aksotech.	www.aksotech.r	12/18/2006
		5 RSB							12/18/2006
		6 CWS							12/18/2006
*	(AutoNumber)								

Figure 2.2. Company Table As an Example in Access

## 2.4. Primary Key.

- One or more fields (columns) whose value or values uniquely identify each record in a table. A primary key does not allow Null values and must always have a unique value. A primary key is used to relate a table to foreign keys in other tables.

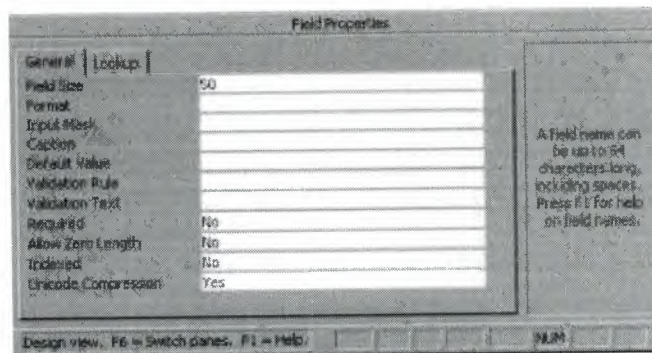
- We do not have to define a primary key, but it's usually a good idea. If you don't define a primary key, Microsoft Access asks us if we would like to create one when we save the table.
- For our tutorial, we make the **Soc Sec #** field the primary key, meaning that every one has a social security number and no 2 are the same.

## Advanced Table Features w/Microsoft Access

- **Assigning a field a specific set of characters**

Example: Customer Name needs maximum 20 character.

1. Switch to Design View
2. Select the field you want to alter
3. At the bottom select the General Tab



4. Select **Field Size**
5. Enter the number of characters you want this field to have

- **Formatting a field to look a specific way**

Example) Formatting Phone Number w/ Area Code (xxx) xxx-xxxx

1. Switch to Design View
  2. Select the field you want to format
  3. At the bottom select the General Tab
  4. Select **Input Mask Box** and click on the ... button at the right.
- **Selecting a value from a dropdown box with a set of values that you assign to it. This saves you from typing it in each time**

Example) Choosing a city that is either Auburn, Bay City, Flint, Midland, or Saginaw

1. Switch to Design View
2. Select the field you want to alter (City)



3. At the bottom select the Lookup Tab
4. In the **Display Control** box, select **Combo Box**
5. Under **Row Source Type**, select **Value List**

## 2.5. Tables Structure In Database.

As I mentioned before I used seven tables in Access Database. All the tables are given below.

antens			
Field Name	Type	Size	Key
rownum	AutoNumber		*
aname	Text	50	
price	Text	7	
unit	Text	4	
alimit	Text	2	

**Table 2.1. antens Database Table**

company			
Field Name	Type	Size	Key
id	AutoNumber		*
cname	Text	50	
tel	Text	11	
fax	Text	11	
branch	Text	3	
address	Text	255	
email	Text	50	
web	Text	50	
adddate	Date		

**Table 2.2. company Database Table**

it			
Field Name	Type	Size	Key
id	AutoNumber		*
cname	Text	100	
mtel	Text	11	
tel	Text	11	
email	Text	50	
adddate	Date		

**Table 2.3. it Database Table**

service			
Field Name	Type	Size	Key
code	AutoNumber		*

pname	Text	50	
price	Text	6	

**Table 2.4. service Database Table**

## **2.6. Microsoft Access Database.**

### **2.6.1. Microsoft Access Database Fundamentals.**

Are you overwhelmed by the large quantities of data that need to be tracked in your organization? Perhaps you're currently using a paper filing system, text documents or a spreadsheet to keep track of your critical information. If you're searching for a more flexible data management system, a database might be just the salvation you're looking for.

What is a database? Quite simply, it's an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

Microsoft Access provides users with one of the simplest and most flexible DBMS solutions on the market today. Regular users of Microsoft products will enjoy the familiar Windows "look and feel" as well as the tight integration with other Microsoft Office family products. An abundance of wizards lessen the complexity of administrative tasks and the ever-present Microsoft Office Helper (you know... the paper clip!) is available for those who care to use it. Before purchasing Access, be sure that your system meets Microsoft's minimum system requirements. To further our discussion, let's first examine three of the major components of Access that most database users will encounter – tables, queries, forms. Once we've completed that we'll look at the added benefits of reports, web integration and SQL Server integration:

Payment.DB			
Field Name	Type	Size	Key
Cuscode	Text	50	
Date	Date/Time	Short Date	
Total	Currency	Standart, Decimal: 2	

**Figure2.3. Sample Table**

The table above contains the employee information for our organization characteristics like name, date of birth and title. Examine the construction of the table and you'll find that each column of the table corresponds to a specific employee characteristic (or **attribute** in database terms). Each row corresponds to one particular employee and contains his or her information. That's all there is to it! If it helps, think of each one of these tables as a spreadsheet-style listing of information.

In the previous section, we learned how tables allow us to create the framework for storing information in a database. Obviously, a database that only stored information would be useless -- we need methods to retrieve information as well. If you simply want to recall the information stored in a table, Microsoft Access allows you to open the table and scroll through the records contained within it. However, the real power of a database lies in its capabilities to answer more complex requests, or **queries**. Access queries provide the capability to combine data from multiple tables and place specific conditions on the data retrieved.

### **2.6.2. Creating Table.**

Many techniques allow you to create a database, the fastest of which consists of using one of the provided examples. Microsoft Access 97 shipped with 22 sample databases while Microsoft Access 2000 ships with 10. Furthermore, the 97 version allowed to provide sample data into the database. This is not available with the 2000 release. The databases that ship with Microsoft Access can help you in two main ways: they provide a fast means of creating a database and you can learn from their structure. To create a database using one of the samples, there is a little detail to follow depending on whether you had launched the program already or not. If Microsoft Access is not running, you can start it. When the first dialog box comes up, you can click the second radio button: Access Database Wizard, Pages

The New dialog box displays two property pages labeled General and Databases. If you want to create a database based on one of the samples, you can click the Databases property page. A list of the sample databases appears. You can then choose one and click OK.

When creating a database using one of the samples, depending on the sample you selected, the Database Wizard will display a few objects and suggest some fields for your database. Some fields are already associated with the objects and some other fields can be added. You can examine them, then add some fields you think are important for your database. You will also have the option of selecting a design layout. Some of the sample databases have been configured to require information about the company you are creating the database for.

## **2.7. Introducing Database.**

Databases are designed to offer an organized mechanism for storing, managing and retrieving information. They do so through the use of tables. If you're familiar with spreadsheets like Microsoft Excel, you're probably already accustomed to storing data in tabular form. It's not much of a stretch to make the leap from spreadsheets to databases.

Just like Excel tables, database tables consist of columns and rows. Each column contains a different type of attribute and each row corresponds to a single record. For example, imagine that we were building a database table that contained names and telephone numbers. We'd probably set up columns named "FirstName", "LastName" and "TelephoneNumber." Then we'd simply start adding rows underneath those columns that contained the data we're planning to store.



At this point, you're probably asking yourself an obvious question if a database is so much like a spreadsheet, why can't I just use a spreadsheet? Databases are actually much more powerful than spreadsheets in the way you're able to manipulate data. Here are just a few of the actions that you can perform on a database that would be difficult if not impossible to perform on a spreadsheet:

- Retrieve all records that match certain criteria

- Update records in bulk

- Cross-reference records in different tables

- Perform complex aggregate calculations

As we walk through this tutorial, you'll learn how you can use databases to achieve each of these objectives. Page 2 of this lesson provides you with an overview of how database keys can be used to uniquely identify records and form relationships between tables. Page 3 describes how the Structured Query Language allows you to interact with your database. On page 4, we examine the different types of databases available on the market today.

## 2.8. Database Keys.

On the previous page of this article, you learned how databases use tables to organize data. As you probably recall, each table consists of a number of rows, each of which corresponds to a single database record. So, how do databases keep all of these records straight? It's through the use of **keys**.

The first type of key we'll discuss is the **primary key**. Every database table should have one or more columns designated as the primary key. The value this key holds should be unique for each record in the database. For example, assume we have a table called Employees that contains personnel information for every employee in our firm. We'd need to select an appropriate primary key that would uniquely identify each employee. Your first thought might be to use the employee's name.

This wouldn't work out very well because it's conceivable that you'd hire two employees with the same name. A better choice might be to use a unique employee ID number that you assign to each employee when they're hired. Some organizations choose to use Social Security Numbers (or similar government identifiers) for this task because each employee already has one and they're guaranteed to be unique. However, the use of Social Security Numbers for this purpose is highly controversial due to privacy concerns.

Once you decide upon a primary key and inform the database of this decision, it will enforce the uniqueness of the key. If you try to insert a record into a table with a primary key that duplicates an existing record, the insert will fail.

Most databases are also capable of generating their own primary keys. Microsoft Access, for example, may be configured to use the AutoNumber data type to assign a unique ID to each record in the table. While effective, this is a bad design practice because it leaves you with a meaningless value in each record in the table. Why not use

that space to store something useful?

The other type of key that we'll discuss in this course is the **foreign key**. These keys are used to create relationships between tables. Natural relationships exist between tables in most database structures. Returning to our employees database, let's imagine that we wanted to add a table containing departmental information to the database. This new table might be called Departments and would contain a large amount of information about the department as a whole. We'd also want to include information about the employees in the department, but it would be redundant to have the same information in two tables (Employees and Departments). Instead, we can create a **relationship** between the two tables.

Let's assume that the Departments table uses the Department Name column as the primary key. To create a relationship between the two tables, we add a new column to the Employees table called Department. We then fill in the name of the department to which each employee belongs. We also inform the database that the Department column in the Employees table is a **foreign key** that references the Departments table. The database will then enforce *referential integrity* by ensuring that all of the values in the Departments column of the Employees table have corresponding entries in the Department table.

Note that there is no uniqueness constraint for a foreign key. We may (and most likely do!) have more than one employee belonging to a single department. Similarly, there's no requirement that an entry in the Departments table have *any* corresponding entry in the Employees table. It is possible that we'd have a department with no employees.

## 2.9. Working with SQL.

SQL (pronounced "ess-que-el") stands for structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingress, etc. although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "select", "insert", "delete", "create" and "drop" can be used to accomplish almost everything that one needs to do with a database.

### 2.9.1. Data Manipulation Language.

The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

### **2.9.1.1.INSERT.**

The INSERT command in SQL is used to add records to an existing table. Returning to the personal\_info example from the previous section, let's imagine that our HR department needs to add a new employee to their database. They could use a command similar to the one shown below:

```
INSERT INTO department  
VALUES('c01','computer','engineering',5,6)
```

Note that there are four values specified for the record.

These correspond to the table attributes in the order they were defined: first\_name, last\_name, employee\_id, and salary.

### **2.8.1.2.SELECT.**

The SELECT command is the most commonly used command in SQL. It allows database users to retrieve the specific information they desire from an operational database. Let's take a look at a few examples, again using the personal\_info table from our employees database.

The command shown below retrieves all of the information contained within the personal\_info table. Note that the asterisk is used as a wildcard in SQL. This literally means "Select everything from the personal\_info table."

```
SELECT *  
FROM student
```

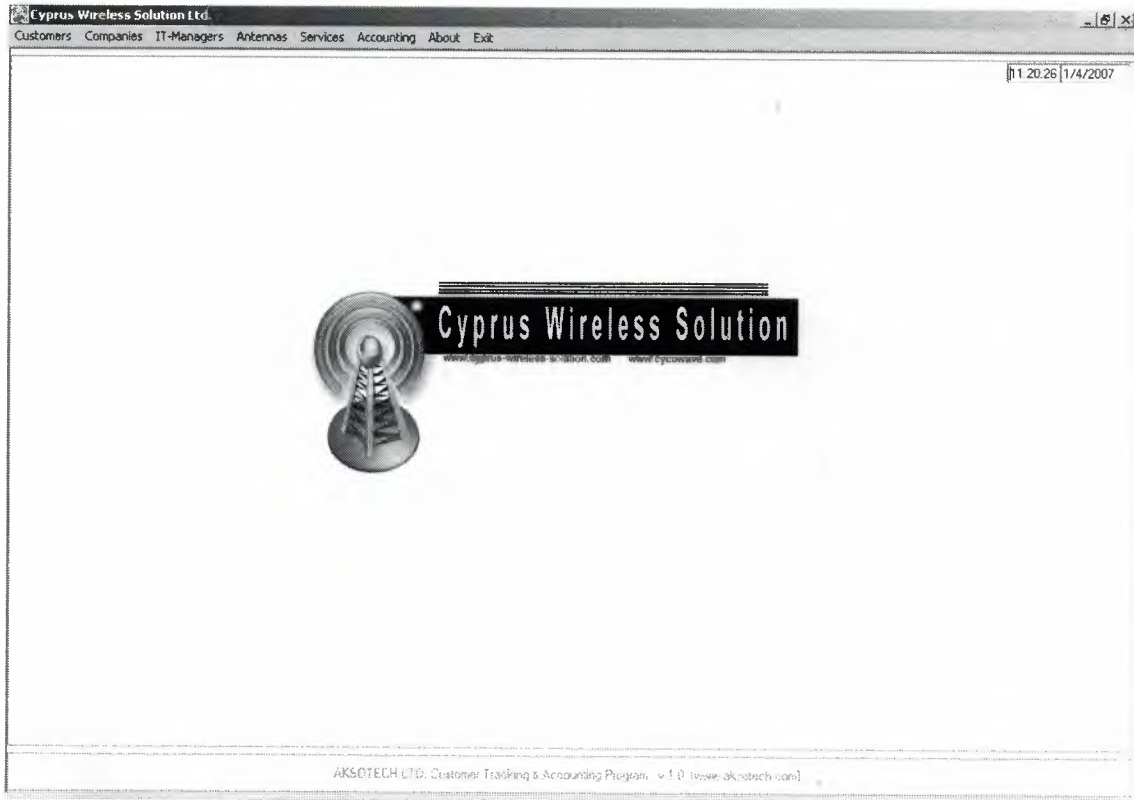
Alternatively, users may want to limit the attributes that are retrieved from the database. For example, the Human Resources department may require a list of the last names of all employees in the company. The following SQL command would retrieve only that information:

```
SELECT stname  
FROM student WHERE stno=20020872
```



# CHAPTER III : CUSTOMER TRACKING & ACCOUNTING PROGRAM USING VISUAL BASIC LANGUAGE

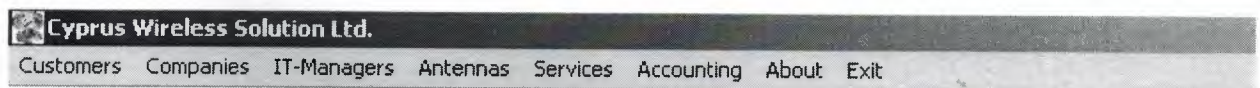
## 3.1. Main Page.



**Figure 3.1 Main Page**

When we run our program main page window appears and this window contains menu bar, Time and Date and at the middle logo of the company.

## 3.2. Menu Bar.



**Figure 3.2 Menu Bar**

### 3.2.1. Customers Menu.



**Figure 3.3 Customers Menu**

This menu contains Add Customer menu and Search Customer menu. When we press to the Add Customer menu the program opens the customer adding page. When we press to the Search Customer menu the program opens the customer search and editing page.

### 3.2.2. Companies Menu.

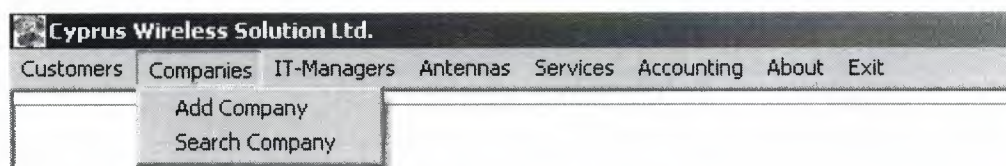


Figure 3.4 Companies Menu

This menu contains Add Company menu and Search Company menu. If we press to the Add Company the program opens the Add new company page, if we press to the Search Company menu the program will open the searching and editing company page.

### 3.2.3. IT-Managers Menu.

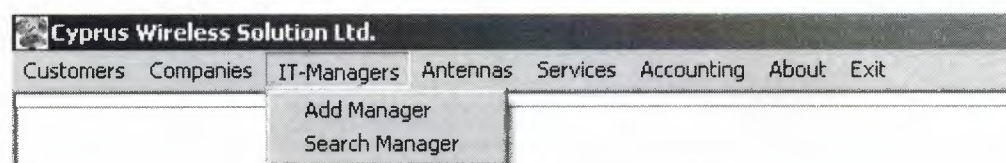


Figure 3.5 IT-Managers Menu

This menu contains Add Manager menu and Search Manager menu. When the Add Manager is pressed the Adding new manager page is opening. If the Search Manager menu is pressed the program is opening the search and editing page of IT-manager.

### 3.2.4. Antennas Menu.

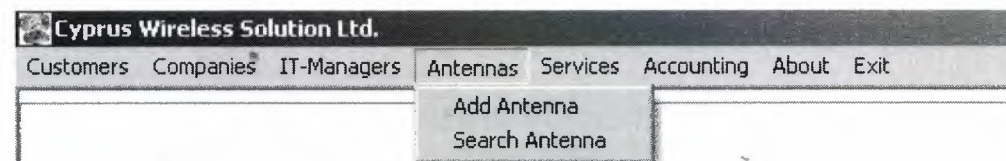


Figure 3.6 Antennas Menu

This menu contains Add Antenna menu and Search Antenna menu, when we press to the Add Antenna menu the page to adding the antenna information to the data base will be opened, when we press to the Search Antenna menu the antenna search from data base and editing page will be opened.



### 3.2.5. Services Menu.

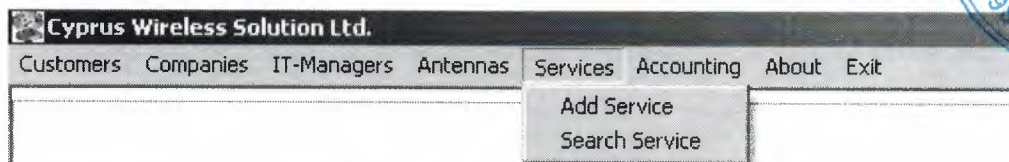


Figure 3.7 Services Menu

This menu contains Add Service menu and Search Service menu. When we press to the Add Service menu the program runs service adding page and when we press to the Search Service menu the program runs the searching and editing page of information of services.

### 3.2.6. Accounting Menu.

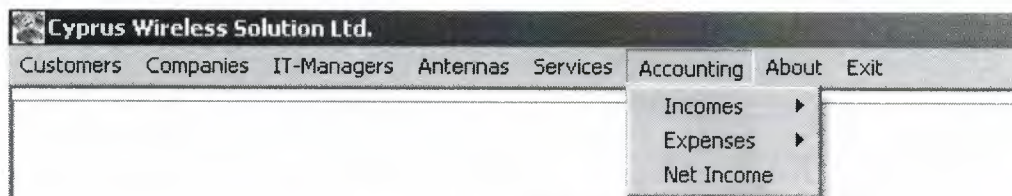


Figure 3.8 Accounting Menu

This menu contains three menus Incomes Menu, Expenses menu and Net Income menu, the two menus of this three menus are also contains three sub menus each of them.

#### 3.2.6.1. Incomes Menu.

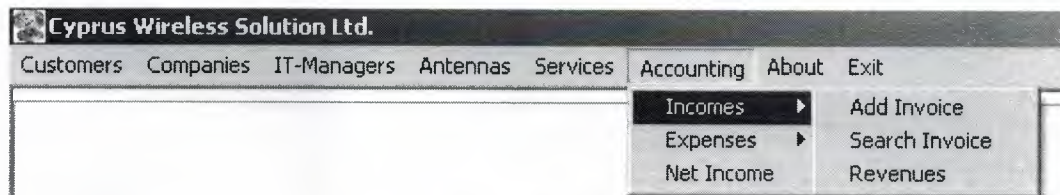


Figure 3.9 Incomes Menu

The Incomes menu contains Add Invoice menu, Search Invoice menu and Revenue menu when we press to the add invoice menu the program runs the income invoice adding page, when we press to the Search invoice menu the program runs the income invoice search and editing page and revenue menu runs the page where the income invoices are listing and give periodic revenue report.

#### 3.2.6.2. Expenses Menu.



Figure 3.10 Expenses Menu



The Expenses menu contains Add Invoice menu, Search Invoice menu and Expense menu, when we press to the Add Invoice menu the program runs the page where the expenses invoices are adding, and when we press to the Search Invoice menu the program runs the page in which we can Search any invoice and edit it and when we press to the Expense menu the program runs the age where the expense invoices are listing and gives periodic expense report.

And Net Income Menu runs the page where the periodic incomes and expenses are listing and gives periodic net income report.

In the menu bar there are two more menus these are About menu which gives information about the program and exit menu which when it pressed it closes the program.

### 3.3. Add New Customer Page.

Figure 3.11 Add New Customer Page

In this page the user can add a new customer to the data base

### 3.4. Search Customer Page.

Figure 3.12 Search Customer Page

In this page the user can search a customer can list all customer and can edit informatin of any of them and also can delete the data from the data base. The user also can print a list of customer with their information from the printer.

### 3.5. Add New Company Page.

**Add New Company**

Company Name :

Telephone :  Fax :  Number Of Branch :

Address :

E-Mail :  Web :

Buttons: Save, New, Cancel

**Figure 3.13 Add New Company Page**

In this page the user can add a new company information to the data base.

### 3.6. Search Company Page.

**Search Company**

By Name :

Company Name	Telephone	E-Mail
AKSOTECH LTD.	03922281758	info@aksotech.net
CWS		
RSB		

Company Name :  AKSOTECH LTD.

Telephone :  03922281758 Fax :  03922281758 Number Of Branch :  1

Address :  Yzb. Tekin Yurdabak Cad. No:17/4 Ortaköy Lefkosa

E-Mail :  info@aksotech.net Web :  www.aksotech.net

Buttons: Edit, Delete, Cancel

**Figure 3.14 Search Company Page**

In this page the user can search any company information from the data base, can list all data, can delete any of the data and can edit any of information of any of company in the data base.

### 3.7. Add New IT-Manager Page.

**Add New IT-Manager**

Name Surname :

M.Telephone :  Telephone :  E-Mail :

Buttons: Save, New, Cancel

**Figure 3.15 Add New IT-Manager Page**



In this page the user can add information of a new IT manager to the data base.

### 3.8. Search IT-Manager Page.

Company Name	Mobile	Telephone	E-Mail
Ercan Kasapoglu	05338470937		trakyahost@hotmail.com
Hüseyin Küksal	05338622900		koksah@yahoo.com
Turgay Aksoy	05338683353	03922281758	turgay@aksotech.net

Name Surname: Ercan Kasapoglu  
M. Telephone: 05338470937 Telephone: Telephone: E-Mail: trakyahost@hotmail.com  
Edit Delete Cancel

Figure 3.16 Search IT-Manager Page

In this page of the program the user can list of the IT managers recorded in the data base, can search, can delete and can edit any of the information of the IT managers recorded in the data base.

### 3.9. Add Antenna Page.

Antenna Name: Price (\$): Unit: Limit:  
Save New Cancel

Figure 3.17 Add Antenna Page

In this part of the program the user can add antenna informations as price and quantity and antenna name to the data base.

### 3.10. Search Antenna Page.

Antenna Name	Price (\$)	Quantity
AP	650	3
Client Antenna	550	49
Dish Antenna	550	20

Antenna Name: AP  
Price (\$): 650 Unit: 3 Limit: 4  
Edit Delete Cancel Print

Figure 3.18 Search Antenna Page



In this part of the program the user can search any information of any antenna type and he/she can list all information of each antenna type from the data base and if any of them is not necessary the user can delete it or if some information are written wrong the user can edit them and also user can print a antenna list from the printer.

3.11. Add New Service Page.



Figure 3.19 Add New Service Page

Here the user is adding new information about the new services if the company starts to give new type of services.

3.12. Search Service Page.

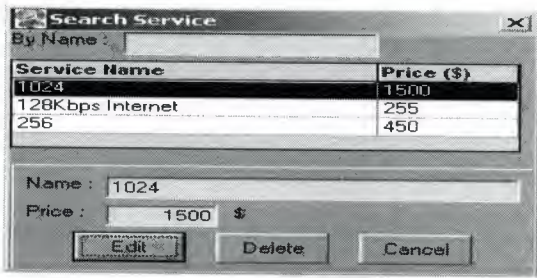


Figure 3.20 Search Service Page

In this page the user can list all services which company is serving and if the company stop to serving some of these services the user can delete them from the data base in this part, and if the prices are changed of some of these services user can change them from this part of the program.

3.13. Add New Income Invoice Page.

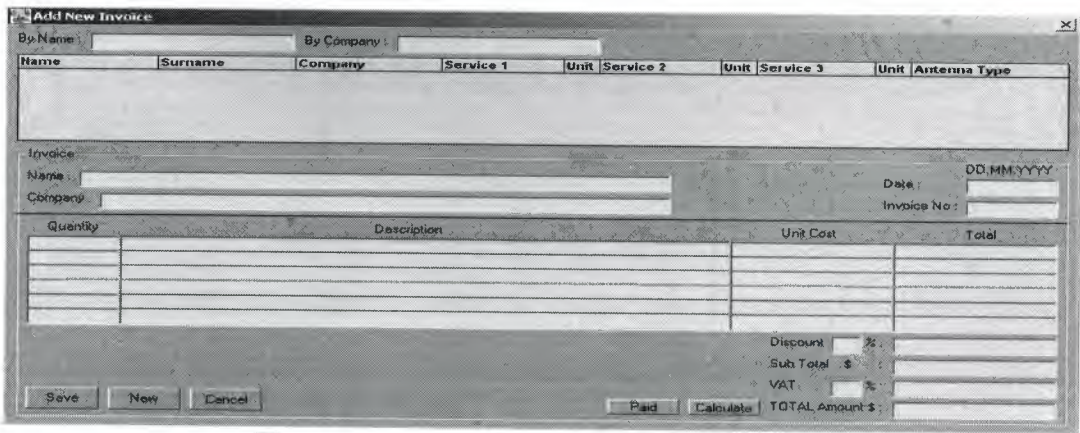


Figure 3.21 Add New Income Invoice Page





### 3.16. Add New Expenses Invoice Page.

Company: \_\_\_\_\_ Date: DD.MM.YYYY  
Invoice No: \_\_\_\_\_

Quantity	Description	Unit Cost	Total

Discount ☐ %  
Sub Total \$  
VAT ☐ %  
TOTAL Amount \$:

Save New Cancel Paid Calculate

Figure 3.24 Add New Expenses Invoice Page

This part of the program is used to add expense invoice to the data base.

### 3.17. Search Expenses Invoice Page.

By Invoice No: \_\_\_\_\_ By Company: \_\_\_\_\_

Invoice No	Company	Date	Amount \$	Status
1650530	Alemdar Electronic	2007.01.01	152.95	Paid

Invoice  
Company: Alemdar Electronic Date: 01.01.2007  
Invoice No: 1650530

Quantity	Description	Unit Cost	Total
10	RJ-45 Jack	0.5	5
1	18V 6A Adaptor	130	130

Discount ☐ 5 %  
Sub Total \$  
VAT ☐ 15 %  
TOTAL Amount \$:

Edit Delete Cancel Calculate

Figure 3.25 Search Expenses Invoice Page

This part of the program is used to search any expense invoice and show its details and if necessary to delete to delete it and if changes is necessary to change it.



### 3.18. Expenses Page.

Invoice No	Company	Date	Status	Amount \$
------------	---------	------	--------	-----------

Print Cancel

Monthly Total Expense \$

Yearly Total Expense \$

**Figure 3.26 Expenses Page**

This part of the program is used to list the expense invoices and to see the periodic expenses of the company (Yearly or Monthly) and can print that list.

### 3.19. Net Income Page.

Invoice No	Customer Name	Company	Date	Status	Amount \$
------------	---------------	---------	------	--------	-----------

Print Cancel

Net Income :

**Figure 3.27 Net Income Page**

This part of the program is used to list all the invoices (revenue and expense) and to get periodic (yearly or monthly) report for net income of the company and also can print the list.

## CONCLUSION

Through building of this project I learned many things, because until now I was just learning programs literary or I was doing small programs, but by the way of this program I investigated a lot in internet, I read many books, I tried many examples in order to improve my knowledge about Visual Basic to make my program a useful thing.

There are many advantages and disadvantages of using Visual Basic in programming, the one of advantages of using VB it is easy in use, and one of the disadvantages of using Visual Basic it is to slow than the some of the other programming languages and it does not have so much properties.

The Program can be updated in the future, we can add other properties to it, and we can use it through internet by adding extra codes, while doing the project wonderful imaginations come to my mind but if I applied them I couldn't be able to finish this project on time, but I'll do my imaginations project in the future if God Wishes.

My Project program is a tracking system program, I used Customer Tracking & Accounting Program, because this program wanted from a wireless internet provider company. And this program is designed for a private company so that this program is not useful in general purpose.

## REFERENCES

1. Ihsan Karagülle and Zeydin Pala, Microsoft Visual Basic 6.0 Pro, Türkmen Printing House, Istanbul ,2001.
2. Yakup Varol and Hasan Koca, VB KodBank 1.7, E-Book, World Wide Web [www.vbkodbank.com](http://www.vbkodbank.com) ,2001.
3. A guide research for writing program. Retrieved December 15, 2006 from the World Wide Web "<http://www.programlama.com>
4. A guide research for writing program. Retrieved January 3, 2007 from the World Wide Web "[http:// www.devarticles.com](http://www.devarticles.com)



# APPENDICES

## MDIForm1.frm

```
Private Sub Label11_Click()  
antennas.Show  
End Sub
```

---

```
Private Sub Label123_Click()  
santenna.Show  
End Sub
```

---

```
Private Sub Label14_Click()  
addservice.Show  
End Sub
```

---

```
Private Sub Label15_Click()  
sservice.Show  
End Sub
```

---

```
Private Sub Label16_Click()  
help.Show  
End Sub
```

---

```
Private Sub Label17_Click()  
Unload Me  
Unload liste1  
Unload liste2  
Unload help  
Unload netincome  
Unload expense  
Unload sinvoicee  
Unload addinvoicee  
Unload Revenue  
Unload sinvoice  
Unload addinvoice  
Unload sservice  
Unload sit  
Unload scompany  
Unload scustomer  
Unload addcompany  
Unload addcustomer  
Unload addit  
Unload antennas  
Unload addservice  
End Sub
```

---

```
Private Sub Label19_Click()  
addinvoice.Show  
End Sub
```

---

```
Private Sub Label2_Click()  
addcustomer.Show  
End Sub
```

---

```
Private Sub Label22_Click()  
sinvoice.Show  
End Sub
```

---

```
Private Sub Label23_Click()  
netincome.Show  
End Sub
```

---

```
Private Sub Label24_Click()  
Revenue.Show  
End Sub
```

---

```
Private Sub Label25_Click()  
addinvoicee.Show  
End Sub
```

---

```
Private Sub Label26_Click()  
sinvoicee.Show  
End Sub
```

---

```
Private Sub Label27_Click()  
expense.Show  
End Sub
```

---

```
Private Sub Label3_Click()  
scustomer.Show  
End Sub
```

---

```
Private Sub Label5_Click()  
addcompany.Show  
End Sub
```

---

```
Private Sub Label6_Click()  
scompany.Show  
End Sub
```

---

```
Private Sub Label8_Click()  
addit.Show  
End Sub
```

---

```
Private Sub Label9_Click()  
sit.Show  
End Sub
```

### **Main1.frm**

```
Private Sub Timer1_Timer()  
Text1 = Date  
Text2 = Time  
End Sub
```

### **addcompany.frm**

```
Private Sub Command1_Click()  
If Text2.Text <> "" Then  
    Data2.Recordset.AddNew  
    Data2.Recordset.cname = Text2.Text  
    Data2.Recordset.tel = Text4.Text  
    Data2.Recordset.fax = Text5.Text  
    Data2.Recordset.branch = Text8.Text  
    Data2.Recordset.address = Text3.Text  
    Data2.Recordset.email = Text6.Text  
    Data2.Recordset.web = Text7.Text  
    Data2.Recordset.adddate = Date  
    Data2.Recordset.Update  
    Data2.Refresh  
    Text2.Text = ""  
    Text3.Text = ""  
    Text4.Text = ""  
    Text5.Text = ""  
    Text6.Text = ""  
    Text7.Text = ""  
    Text8.Text = ""  
    Text2.SetFocus  
Else  
    MsgBox "You can not leave the Company Name empty.", 48, "Caution Message"  
    If Text2.Text = "" Then  
        Text2.SetFocus  
    End If  
End If  
End Sub
```

---

```
Private Sub Command2_Click()  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""  
Text5.Text = ""  
Text6.Text = ""  
Text7.Text = ""  
Text8.Text = ""  
Text2.SetFocus  
End Sub
```

---

```
Private Sub Command3_Click()  
Unload Me  
End Sub
```



```

Private Sub Form_Load()
Data2.DatabaseName = App.Path & "\cwsca.mdb"
End Sub

```

#### **addcustomer.frm**

```

Private Sub Command1_Click()
If Text2.Text <> "" And Text3.Text <> "" Then
    If Data5.Recordset.RecordCount = 1 And DBCombo6 <> "" Then
        Data5.Refresh
        A = DBCombo6
        B = "aname=" & A & ""
        Data5.Recordset.FindFirst B
        A1 = Data5.Recordset![aname]
        A2 = Data5.Recordset![price]
        A3 = Data5.Recordset![unit]
        A4 = Data5.Recordset![alimit]
        If A3 < A4 And A3 > 0 Then
            MsgBox "The stock is too low.", 48, "Caution Message"
        End If
        If A3 = 0 Then
            MsgBox "The stock is empty!!!", 16, "Error"
            Exit Sub
        End If
        Data5.Recordset.Edit
        Data5.Recordset![aname] = A1
        Data5.Recordset![price] = A2
        Data5.Recordset![unit] = A3 - 1
        Data5.Recordset![alimit] = A4
        Data5.Recordset.Update
    End If
    Data1.Recordset.AddNew
    Data1.Recordset.cname = Text2.Text
    Data1.Recordset.surname = Text3.Text
    Data1.Recordset.address = Text4.Text
    Data1.Recordset.mtel = Text5.Text
    Data1.Recordset.tel = Text6.Text
    Data1.Recordset.fax = Text7.Text
    Data1.Recordset.email = Text8.Text
    Data1.Recordset.it = DBCombo2
    Data1.Recordset.company = DBCombo1
    Data1.Recordset.ip = Text11.Text
    Data1.Recordset.service1 = DBCombo3
    Data1.Recordset.unit1 = Text1.Text
    Data1.Recordset.service2 = DBCombo4
    Data1.Recordset.unit2 = Text9.Text
    Data1.Recordset.service3 = DBCombo5
    Data1.Recordset.unit3 = Text12.Text
    Data1.Recordset.sid = Text10.Text
    Data1.Recordset.atype = DBCombo6

```

```

Data1.Recordset.adddate = Date
Data1.Recordset.Update
Data1.Refresh
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
Text9.Text = ""
Text10.Text = ""
Text11.Text = ""
Text12.Text = ""
Text2.SetFocus
Else
MsgBox "You can not leave the Name and Surname empty.", 48, "Caution Message"
If Text2.Text = "" Then
    Text2.SetFocus
End If
If Text3.Text = "" Then
    Text3.SetFocus
End If
End If
End Sub

```

---

```

Private Sub Command2_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
Text9.Text = ""
Text10.Text = ""
Text11.Text = ""
Text12.Text = ""
Text2.SetFocus
End Sub

```

---

```

Private Sub Command3_Click()
Unload Me
End Sub

```

---

```

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\cwsca.mdb"
Data2.DatabaseName = App.Path & "\cwsca.mdb"

```

```
Data3.DatabaseName = App.Path & "\cwsca.mdb"
Data4.DatabaseName = App.Path & "\cwsca.mdb"
Data5.DatabaseName = App.Path & "\cwsca.mdb"
End Sub
```

### **addinvoice.frm**

```
Dim statu As String
Private Sub Command1_Click()
If Text3.Text <> "" And Text4.Text <> "" And Text5.Text <> "" Then
    Data6.Recordset.AddNew
    Data6.Recordset.customer = Text3.Text
    Data6.Recordset.company = Text4.Text
    Data6.Recordset.ino = Text5.Text
    dat = Right(Text6.Text, 4) + Mid(Text6.Text, 3, 4) + Left(Text6.Text, 2)
    Data6.Recordset.idate = dat
    Data6.Recordset.amount = Text34.Text
    Data6.Recordset.statuss = statu
    Data6.Recordset.disc = Text35.Text
    Data6.Recordset.vat = Text36.Text
    Data6.Recordset.q1 = Text7.Text
    Data6.Recordset.q2 = Text11.Text
    Data6.Recordset.q3 = Text15.Text
    Data6.Recordset.q4 = Text19.Text
    Data6.Recordset.q5 = Text23.Text
    Data6.Recordset.q6 = Text27.Text
    Data6.Recordset.D1 = Text8.Text
    Data6.Recordset.D2 = Text12.Text
    Data6.Recordset.d3 = Text16.Text
    Data6.Recordset.d4 = Text20.Text
    Data6.Recordset.d5 = Text24.Text
    Data6.Recordset.d6 = Text28.Text
    Data6.Recordset.Update
    Data6.Refresh
    Text3 = ""
    Text4 = ""
    Text5 = ""
    Text6 = ""
    Text7 = ""
    Text8 = ""
    Text11 = ""
    Text12 = ""
    Text15 = ""
    Text16 = ""
    Text19 = ""
    Text20 = ""
    Text23 = ""
    Text24 = ""
    Text27 = ""
    Text28 = ""
    Text31 = ""
```



```

Text32 = ""
Text33 = ""
Text34 = ""
Text35 = ""
Text36 = ""
statu = "Not Paid"
Command6.Enabled = True
Else
MsgBox "You can not leave the Name, Company and Invoice No empty.", 48,
"Caution Message"
If Text3.Text = "" Then
    Text3.SetFocus
End If
If Text4.Text = "" Then
    Text4.SetFocus
End If
If Text5.Text = "" Then
    Text5.SetFocus
End If
End If
End Sub

```

---

```

Private Sub Command2_Click()

```

```

Text3 = ""
Text4 = ""
Text5 = ""
Text6 = ""
Text7 = ""
Text8 = ""
Text11 = ""
Text12 = ""
Text15 = ""
Text16 = ""
Text19 = ""
Text20 = ""
Text23 = ""
Text24 = ""
Text27 = ""
Text28 = ""
Text31 = ""
Text32 = ""
Text33 = ""
Text34 = ""
Text35 = ""
Text36 = ""

```

```

statu = "Not Paid"
Command6.Enabled = True
End Sub

```

---

```

Private Sub Command3_Click()

```

Unload Me  
End Sub

---

```
Private Sub Command4_Click()  
tot = Val(Text10) + Val(Text14) + Val(Text18) + Val(Text22) + Val(Text26) +  
Val(Text30)  
If Text35 = "" Then  
    Text32 = 0  
End If  
Text32 = (Val(tot) * Val(Text35)) / 100  
Text31 = Val(tot) - Val(Text32)  
If Text36 = "" Then  
    Text33 = ""  
End If  
Text33 = (Val(Text31) * Val(Text36)) / 100  
Text34 = Val(Text31) + Val(Text33)  
End Sub
```

---

```
Private Sub Command6_Click()  
statu = "Paid"  
Command6.Enabled = False  
End Sub
```

---

```
Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)  
If Data1.Recordset.RecordCount = 0 Then  
    Text3 = ""  
    Text4 = ""  
Exit Sub  
End If  
Text3 = Data1.Recordset.cname & "" + " " + Data1.Recordset.surname & ""  
Text4 = Data1.Recordset.company & ""  
End Sub
```

---

```
Private Sub Form_Load()  
statu = "Not Paid"  
Data1.DatabaseName = App.Path & "\cwsca.mdb"  
Data4.DatabaseName = App.Path & "\cwsca.mdb"  
Data6.DatabaseName = App.Path & "\cwsca.mdb"  
squ = "SELECT * From customer WHERE cname LIKE "  
squ = squ & Text1 & "*" order by cname ASC;"  
Data1.RecordSource = squ  
Data1.Refresh  
End Sub
```

---

```
Private Sub Text1_Change()  
squ = "SELECT * From customer WHERE cname LIKE "  
squ = squ & Text1 & "*" order by cname ASC;"  
Data1.RecordSource = squ  
Data1.Refresh  
End Sub
```

```

Private Sub Text11_Change()
Text14 = Val(Text11) * Val(Text13)
If Text11 = "" And Text13 = "" Then
    Text14 = ""
End If
End Sub

```

---

```

Private Sub Text12_Change()
If Data4.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text12.Text
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text13 = Data4.Recordset![price]
If Text12 = "" Then
    Text13 = ""
End If
End Sub

```

---

```

Private Sub Text13_Change()
Text14 = Val(Text11) * Val(Text13)
If Text11 = "" And Text13 = "" Then
    Text14 = ""
End If
End Sub

```

---

```

Private Sub Text15_Change()
Text18 = Val(Text15) * Val(Text17)
If Text15 = "" And Text17 = "" Then
    Text18 = ""
End If
End Sub

```

---

```

Private Sub Text16_Change()
If Data4.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text16.Text
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text17 = Data4.Recordset![price]
If Text16 = "" Then
    Text17 = ""
End If
End Sub

```

---

```

Private Sub Text17_Change()
Text18 = Val(Text15) * Val(Text17)
If Text15 = "" And Text17 = "" Then

```



```
Text18 = ""  
End If  
End Sub
```

---

```
Private Sub Text19_Change()  
Text22 = Val(Text19) * Val(Text21)  
If Text19 = "" And Text21 = "" Then  
Text22 = ""  
End If  
End Sub
```

---

```
Private Sub Text2_Change()  
squ = "SELECT * From customer WHERE company LIKE ""  
squ = squ & Text2 & '*' order by company ASC;"  
Data1.RecordSource = squ  
Data1.Refresh  
End Sub
```

---

```
Private Sub Text20_Change()  
If Data4.Recordset.RecordCount = 0 Then  
Exit Sub  
End If  
A = Text20.Text  
B = "pname="" & A & ""  
Data4.Recordset.FindFirst B  
Text21 = Data4.Recordset![price]  
If Text20 = "" Then  
Text21 = ""  
End If  
End Sub
```

---

```
Private Sub Text21_Change()  
Text22 = Val(Text19) * Val(Text21)  
If Text19 = "" And Text21 = "" Then  
Text22 = ""  
End If  
End Sub
```

---

```
Private Sub Text23_Change()  
Text26 = Val(Text23) * Val(Text25)  
If Text23 = "" And Text25 = "" Then  
Text26 = ""  
End If  
End Sub
```

---

```
Private Sub Text24_Change()  
If Data4.Recordset.RecordCount = 0 Then  
Exit Sub  
End If  
A = Text24.Text
```

```
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text25 = Data4.Recordset![price]
If Text24 = "" Then
    Text25 = ""
End If
End Sub
```

---

```
Private Sub Text25_Change()
Text26 = Val(Text23) * Val(Text25)
If Text23 = "" And Text25 = "" Then
    Text26 = ""
End If
End Sub
```

---

```
Private Sub Text27_Change()
Text30 = Val(Text27) * Val(Text29)
If Text27 = "" And Text29 = "" Then
    Text30 = ""
End If
End Sub
```

---

```
Private Sub Text28_Change()
If Data4.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text28.Text
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text29 = Data4.Recordset![price]
If Text28 = "" Then
    Text29 = ""
End If
End Sub
```

---

```
Private Sub Text29_Change()
Text30 = Val(Text27) * Val(Text29)
If Text27 = "" And Text29 = "" Then
    Text30 = ""
End If
End Sub
```

---

```
Private Sub Text7_Change()
Text10 = Val(Text7) * Val(Text9)
If Text7 = "" And Text9 = "" Then
    Text10 = ""
End If
End Sub
```

---

```
Private Sub Text8_Change()
```

```

If Data4.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text8.Text
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text9 = Data4.Recordset![price]
If Text8 = "" Then
    Text9 = ""
End If
End Sub

```

```

-----
Private Sub Text9_Change()
Text10 = Val(Text7) * Val(Text9)
If Text7 = "" And Text9 = "" Then
    Text10 = ""
End If
End Sub

```

#### **addinvoicce.frm**

```

Dim statu As String
Private Sub Command1_Click()
If Text4.Text <> "" And Text5.Text <> "" Then
    Data7.Recordset.AddNew
    Data7.Recordset.company = Text4.Text
    Data7.Recordset.ino = Text5.Text
    dat = Right(Text6.Text, 4) + Mid(Text6.Text, 3, 4) + Left(Text6.Text, 2)
    Data7.Recordset.idate = dat
    Data7.Recordset.amount = Text34.Text
    Data7.Recordset.statuss = statu
    Data7.Recordset.disc = Text35.Text
    Data7.Recordset.vat = Text36.Text
    Data7.Recordset.q1 = Text7.Text
    Data7.Recordset.q2 = Text11.Text
    Data7.Recordset.q3 = Text15.Text
    Data7.Recordset.q4 = Text19.Text
    Data7.Recordset.q5 = Text23.Text
    Data7.Recordset.q6 = Text27.Text
    Data7.Recordset.D1 = Text8.Text
    Data7.Recordset.D2 = Text12.Text
    Data7.Recordset.d3 = Text16.Text
    Data7.Recordset.d4 = Text20.Text
    Data7.Recordset.d5 = Text24.Text
    Data7.Recordset.d6 = Text28.Text
    Data7.Recordset.u1 = Text9.Text
    Data7.Recordset.u2 = Text13.Text
    Data7.Recordset.u3 = Text17.Text
    Data7.Recordset.u4 = Text21.Text
    Data7.Recordset.u5 = Text25.Text
    Data7.Recordset.u6 = Text29.Text

```



```

Data7.Recordset.Update
Data7.Refresh
Text4 = ""
Text5 = ""
Text6 = ""
Text7 = ""
Text8 = ""
Text9 = ""
Text11 = ""
Text12 = ""
Text13 = ""
Text15 = ""
Text16 = ""
Text17 = ""
Text19 = ""
Text20 = ""
Text21 = ""
Text23 = ""
Text24 = ""
Text25 = ""
Text27 = ""
Text28 = ""
Text29 = ""
Text31 = ""
Text32 = ""
Text33 = ""
Text34 = ""
Text35 = ""
Text36 = ""
statu = "Not Paid"
Command6.Enabled = True
Else
MsgBox "You can not leave the Company and Invoice No empty.", 48, "Caution
Message"
If Text4.Text = "" Then
    Text4.SetFocus
End If
If Text5.Text = "" Then
    Text5.SetFocus
End If
End If
End Sub
-----
Private Sub Command2_Click()
Text4 = ""
Text5 = ""
Text6 = ""
Text7 = ""
Text8 = ""
Text9 = ""

```

```

Text11 = ""
Text12 = ""
Text13 = ""
Text15 = ""
Text16 = ""
Text17 = ""
Text19 = ""
Text20 = ""
Text21 = ""
Text23 = ""
Text24 = ""
Text25 = ""
Text27 = ""
Text28 = ""
Text29 = ""
Text31 = ""
Text32 = ""
Text33 = ""
Text34 = ""
Text35 = ""
Text36 = ""
statu = "Not Paid"
Command6.Enabled = True
End Sub

```

---

```

Private Sub Command3_Click()
Unload Me
End Sub

```

---

```

Private Sub Command4_Click()
tot = Val(Text10) + Val(Text14) + Val(Text18) + Val(Text22) + Val(Text26) +
Val(Text30)
If Text35 = "" Then
    Text32 = 0
End If
Text32 = (Val(tot) * Val(Text35)) / 100
Text31 = Val(tot) - Val(Text32)
If Text36 = "" Then
    Text33 = ""
End If
Text33 = (Val(Text31) * Val(Text36)) / 100
Text34 = Val(Text31) + Val(Text33)
End Sub

```

---

```

Private Sub Command6_Click()
statu = "Paid"
Command6.Enabled = False
End Sub

```

---

```

Private Sub Form_Load()

```

```
statu = "Not Paid"
Data7.DatabaseName = App.Path & "\cwsca.mdb"
End Sub
```

---

```
Private Sub Text11_Change()
Text14 = Val(Text11) * Val(Text13)
If Text11 = "" And Text13 = "" Then
    Text14 = ""
End If
End Sub
```

---

```
Private Sub Text13_Change()
Text14 = Val(Text11) * Val(Text13)
If Text11 = "" And Text13 = "" Then
    Text14 = ""
End If
End Sub
```

---

```
Private Sub Text15_Change()
Text18 = Val(Text15) * Val(Text17)
If Text15 = "" And Text17 = "" Then
    Text18 = ""
End If
End Sub
```

---

```
Private Sub Text17_Change()
Text18 = Val(Text15) * Val(Text17)
If Text15 = "" And Text17 = "" Then
    Text18 = ""
End If
End Sub
```

---

```
Private Sub Text19_Change()
Text22 = Val(Text19) * Val(Text21)
If Text19 = "" And Text21 = "" Then
    Text22 = ""
End If
End Sub
```

---

```
Private Sub Text21_Change()
Text22 = Val(Text19) * Val(Text21)
If Text19 = "" And Text21 = "" Then
    Text22 = ""
End If
End Sub
```

---

```
Private Sub Text23_Change()
Text26 = Val(Text23) * Val(Text25)
If Text23 = "" And Text25 = "" Then
    Text26 = ""
End If
End Sub
```



End If  
End Sub

---

```
Private Sub Text25_Change()  
Text26 = Val(Text23) * Val(Text25)  
If Text23 = "" And Text25 = "" Then  
    Text26 = ""  
End If  
End Sub
```

---

```
Private Sub Text27_Change()  
Text30 = Val(Text27) * Val(Text29)  
If Text27 = "" And Text29 = "" Then  
    Text30 = ""  
End If  
End Sub
```

---

```
Private Sub Text29_Change()  
Text30 = Val(Text27) * Val(Text29)  
If Text27 = "" And Text29 = "" Then  
    Text30 = ""  
End If  
End Sub
```

---

```
Private Sub Text7_Change()  
Text10 = Val(Text7) * Val(Text9)  
If Text7 = "" And Text9 = "" Then  
    Text10 = ""  
End If  
End Sub
```

---

```
Private Sub Text9_Change()  
Text10 = Val(Text7) * Val(Text9)  
If Text7 = "" And Text9 = "" Then  
    Text10 = ""  
End If  
End Sub
```

#### **addit.frm**

```
Private Sub Command1_Click()  
If Text2.Text <> "" Then  
    Data3.Recordset.AddNew  
    Data3.Recordset.cname = Text2.Text  
    Data3.Recordset.mtel = Text4.Text  
    Data3.Recordset.tel = Text5.Text  
    Data3.Recordset.email = Text6.Text  
    Data3.Recordset.adddate = Date  
    Data3.Recordset.Update  
    Data3.Refresh  
    Text2.Text = ""  
End If  
End Sub
```

```

Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text2.SetFocus
Else
MsgBox "You can not leave the Name empty.", 48, "Caution Message"
If Text2.Text = "" Then
Text2.SetFocus
End If
End If
End Sub

```

```

-----
Private Sub Command2_Click()
Text2.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text2.SetFocus
End Sub

```

```

-----
Private Sub Command3_Click()
Unload Me
End Sub

```

```

-----
Private Sub Form_Load()
Data3.DatabaseName = App.Path & "\cwsca.mdb"
End Sub

```

#### **addservice.frm**

```

Private Sub Command1_Click()
Unload Me
End Sub

```

```

-----
Private Sub Command2_Click()
Text2.Text = ""
Text4.Text = ""
Text2.SetFocus
End Sub

```

```

-----
Private Sub Command3_Click()
If Text2.Text <> "" Then
Data4.Recordset.AddNew
Data4.Recordset.pname = Text2.Text
Data4.Recordset.price = Text4.Text
Data4.Recordset.Update
Data4.Refresh
Text2.Text = ""
Text4.Text = ""
Text2.SetFocus
Else

```

```

MsgBox "You can not leave the Name empty.", 48, "Caution Message"
If Text2.Text = "" Then
    Text2.SetFocus
End If
End If
End Sub

```

---

```

Private Sub Form_Load()
Data4.DatabaseName = App.Path & "\cwsca.mdb"
End Sub

```

#### **antennas.frm**

```

Private Sub Command1_Click()
If Text1.Text = "" Then
    MsgBox "You can not leave Antenna Name empty.", 48, "Caution Message"
    Text1.SetFocus
Else
    Data5.Recordset.AddNew
    Data5.Recordset.aname = Text1.Text
    Data5.Recordset.price = Text2.Text
    Data5.Recordset.unit = Text3.Text
    Data5.Recordset.alimit = Text4.Text
    Data5.Recordset.Update
    Data5.Refresh
    Text1 = ""
    Text2 = ""
    Text3 = ""
    Text4 = ""
End If
End Sub

```

---

```

Private Sub Command2_Click()
Text1 = ""
Text2 = ""
Text3 = ""
Text4 = ""
End Sub

```

---

```

Private Sub Command3_Click()
Unload Me
End Sub

```

---

```

Private Sub Form_Load()
Data5.DatabaseName = App.Path & "\cwsca.mdb"
End Sub

```

#### **expense.frm**

```

Dim dat As String
Dim A As String
Dim B As String

```



```
Dim D1 As String
Dim D2 As String
Dim R As String
Option Explicit
Dim conv As String
```

---

```
Private Sub Command1_Click()
List1.Clear
List3.Clear
List4.Clear
List5.Clear
List6.Clear
If Combo2.Text <> "" Then
    If Combo2.Text = "January" Then
        dat = Combo1.Text + ".01"
    End If
    If Combo2.Text = "February" Then
        dat = Combo1.Text + ".02"
    End If
    If Combo2.Text = "March" Then
        dat = Combo1.Text + ".03"
    End If
    If Combo2.Text = "April" Then
        dat = Combo1.Text + ".04"
    End If
    If Combo2.Text = "May" Then
        dat = Combo1.Text + ".05"
    End If
    If Combo2.Text = "June" Then
        dat = Combo1.Text + ".06"
    End If
    If Combo2.Text = "July" Then
        dat = Combo1.Text + ".07"
    End If
    If Combo2.Text = "August" Then
        dat = Combo1.Text + ".08"
    End If
    If Combo2.Text = "September" Then
        dat = Combo1.Text + ".09"
    End If
    If Combo2.Text = "October" Then
        dat = Combo1.Text + ".10"
    End If
    If Combo2.Text = "November" Then
        dat = Combo1.Text + ".11"
    End If
    If Combo2.Text = "December" Then
        dat = Combo1.Text + ".12"
    End If
    If Data7.Recordset.RecordCount = 0 Then
```

```

Exit Sub
End If
Data7.Recordset.MoveFirst
Do Until Data7.Recordset.EOF
    D1 = Data7.Recordset![idate]
    R = Left(D1, 7)
    If R = dat Then
        List1.AddItem Data7.Recordset![ino]
        List3.AddItem Data7.Recordset![company]
        conv = Data7.Recordset![idate]
        List4.AddItem Right(conv, 2) + Mid(conv, 5, 4) + Left(conv, 4)
        List5.AddItem Data7.Recordset![status]
        List6.AddItem Data7.Recordset![amount]
    End If
    Data7.Recordset.MoveNext
Loop
Text2 = ""
A = 0
B = 0
If Data7.Recordset.RecordCount = 0 Then
    Text1 = ""
    Text2 = ""
    Exit Sub
End If
Data7.Recordset.MoveFirst
Do Until Data7.Recordset.EOF
    D1 = Data7.Recordset![idate]
    R = Left(D1, 7)
    D2 = Data7.Recordset![status]
    If R = dat And D2 = "Paid" Then
        B = Data7.Recordset![amount]
        A = Val(A) + Val(B)
    End If
    Data7.Recordset.MoveNext
Loop
Text1 = A
Else
    dat = Combo1.Text
    If Data7.Recordset.RecordCount = 0 Then
        Exit Sub
    End If
    Data7.Recordset.MoveFirst
    Do Until Data7.Recordset.EOF
        D1 = Data7.Recordset![idate]
        R = Left(D1, 4)
        If R = dat Then
            List1.AddItem Data7.Recordset![ino]
            List3.AddItem Data7.Recordset![company]
            conv = Data7.Recordset![idate]
            List4.AddItem Right(conv, 2) + Mid(conv, 5, 4) + Left(conv, 4)

```

```

        List5.AddItem Data7.Recordset![status]
        List6.AddItem Data7.Recordset![amount]
    End If
    Data7.Recordset.MoveNext
Loop
Text1 = ""
A = 0
B = 0
If Data7.Recordset.RecordCount = 0 Then
    Text1 = ""
    Text2 = ""
    Exit Sub
End If
Data7.Recordset.MoveFirst
Do Until Data7.Recordset.EOF
    D1 = Data7.Recordset![idate]
    R = Left(D1, 4)
    D2 = Data7.Recordset![status]
    If R = dat And D2 = "Paid" Then
        B = Data7.Recordset![amount]
        A = Val(A) + Val(B)
    End If
    Data7.Recordset.MoveNext
Loop
Text2 = A
End If
End Sub

```

---

```

Private Sub Command2_Click()
Dim yb, ys, x1, x2, x3, x4, x5, i
With Printer
    .FontSize = 12
    .FontName = "Courier New Tr"
Printer.Print "Cyprus Wireless Solution Ltd."
Printer.Print "26A Atatürk Avenue. Candemir"
Printer.Print "Apt7 Floor1 Yenisehir/Nicosia"
Printer.Print "Tel:+90 392 227 0490"
Printer.Print "Tel:+90 392 227 5883"
Printer.Print ""
    .FontSize = 10
    yb = .CurrentY
    x1 = .TextWidth(Space(25))
    x2 = x1 + .TextWidth(Space(100))
    x3 = x2 + .TextWidth(Space(24))
    x4 = x3 + .TextWidth(Space(20))
    x5 = x4 + .TextWidth(Space(25))
Printer.Line (30, yb)-(x5 - 30, yb)
    .CurrentX = 30
    .FontBold = True
Printer.Print "Invoice No";

```



```

        .CurrentX = x1 + 30
Printer.Print "Company";
        .CurrentX = x2 + 30
Printer.Print "Date";
        .CurrentX = x3 + 30
Printer.Print "Status";
        .CurrentX = x4 + 30
Printer.Print "Amount $"
        .FontBold = False
Printer.Line (30, .CurrentY)-(x5 - 30, .CurrentY)
For i = 0 To List1.ListCount - 1
    .CurrentX = 30
Printer.Print List1.List(i);
    .CurrentX = x1 + 30
Printer.Print List3.List(i);
    .CurrentX = x2 + 30
Printer.Print List4.List(i);
    .CurrentX = x3 + 30
Printer.Print List5.List(i);
    .CurrentX = x4 + 30
Printer.Print List6.List(i)
Printer.Line (30, .CurrentY)-(x5 - 30, .CurrentY)
Next i
ys = .CurrentY
Printer.Line (30, yb)-(30, ys)
Printer.Line (x1 + 30, yb)-(x1 + 30, ys)
Printer.Line (x2 + 30, yb)-(x2 + 30, ys)
Printer.Line (x3 + 30, yb)-(x3 + 30, ys)
Printer.Line (x4 + 30, yb)-(x4 + 30, ys)
Printer.Line (x5 + 30, yb)-(x5 + 30, ys)
If Combo2.Text <> "" Then
    .CurrentX = x2
    Printer.Print "Monthly Total Expense:";
    .CurrentX = x4
    Printer.Print Text1.Text
End If
If Combo2.Text = "" Then
    .CurrentX = x2
    Printer.Print "Yearly Total Expense:";
    .CurrentX = x4
    Printer.Print Text2.Text
End If
.EndDoc
End With
MsgBox "Document is Printing."
End Sub

-----
Private Sub Command3_Click()
Unload Me
End Sub

```

```
Private Sub Form_Load()
Data7.DatabaseName = App.Path & "\cwsca.mdb"
End Sub
```

### **help.frm**

```
Private Sub Command1_Click()
Unload Me
End Sub
```

### **liste1.frm**

Option Explicit

---

```
Private Sub Command1_Click()
Dim yb, ys, x1, x2, x3, x4, x5, x6, x7, i
With Printer
.FontSize = 12
.FontName = "Courier New Tr"
Printer.Print "Cyprus Wireless Solution Ltd."
Printer.Print "26A Atatürk Avenue. Candemir"
Printer.Print "Apt7 Floor1 Yenisehir/Nicosia"
Printer.Print "Tel:+90 392 227 0490"
Printer.Print "Tel:+90 392 227 5883"
Printer.Print ""
.FontSize = 10
yb = .CurrentY
x1 = .TextWidth(Space(35))
x2 = x1 + .TextWidth(Space(35))
x3 = x2 + .TextWidth(Space(24))
x4 = x3 + .TextWidth(Space(24))
x5 = x4 + .TextWidth(Space(38))
x6 = x5 + .TextWidth(Space(28))
x7 = x6 + .TextWidth(Space(21))
Printer.Line (30, yb)-(x7 - 30, yb)
.CurrentX = 30
.FontBold = True
Printer.Print "Customer Name";
.CurrentX = x1 + 30
Printer.Print "Company";
.CurrentX = x2 + 30
Printer.Print "Telephone";
.CurrentX = x3 + 30
Printer.Print "Mobile";
.CurrentX = x4 + 30
Printer.Print "E-Mail";
.CurrentX = x5 + 30
Printer.Print "IP Address";
.CurrentX = x6 + 30
Printer.Print "Software ID"
.FontBold = False
Printer.Line (30, .CurrentY)-(x7 - 30, .CurrentY)
```

```

For i = 0 To List1.ListCount - 1
    .CurrentX = 30
Printer.Print List1.List(i);
    .CurrentX = x1 + 30
Printer.Print List2.List(i);
    .CurrentX = x2 + 30
Printer.Print List3.List(i);
    .CurrentX = x3 + 30
Printer.Print List4.List(i);
    .CurrentX = x4 + 30
Printer.Print List5.List(i);
    .CurrentX = x5 + 30
Printer.Print List6.List(i);
    .CurrentX = x6 + 30
Printer.Print List7.List(i)
Printer.Line (30, .CurrentY)-(x7 - 30, .CurrentY)
Next i
ys = .CurrentY
Printer.Line (30, yb)-(30, ys)
Printer.Line (x1 + 30, yb)-(x1 + 30, ys)
Printer.Line (x2 + 30, yb)-(x2 + 30, ys)
Printer.Line (x3 + 30, yb)-(x3 + 30, ys)
Printer.Line (x4 + 30, yb)-(x4 + 30, ys)
Printer.Line (x5 + 30, yb)-(x5 + 30, ys)
Printer.Line (x6 + 30, yb)-(x6 + 30, ys)
Printer.Line (x7 + 30, yb)-(x7 + 30, ys)
.EndDoc
End With
MsgBox "Document is Printing."
End Sub

```

---

```

Private Sub Command2_Click()
Unload Me
End Sub

```

---

```

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\cwsca.mdb"
Data1.Refresh
If Data1.Recordset.RecordCount <> 0 Then
    Do Until Data1.Recordset.EOF
        List1.AddItem Data1.Recordset.cname & "" + " " + Data1.Recordset.surname & ""
        List2.AddItem Data1.Recordset.company & ""
        List3.AddItem Data1.Recordset.tel & ""
        List4.AddItem Data1.Recordset.mtel & ""
        List5.AddItem Data1.Recordset.email & ""
        List6.AddItem Data1.Recordset.ip & ""
        List7.AddItem Data1.Recordset.sid & ""
        Data1.Recordset.MoveNext
    Loop
End If

```



End Sub

## liste2.frm

Option Explicit

```
-----
Private Sub Command1_Click()
Dim yb, ys, x1, x2, x3, i
With Printer
    .FontSize = 12
    .FontName = "Courier New Tr"
Printer.Print "Cyprus Wireless Solution Ltd."
Printer.Print "26A Atatürk Avenue. Candemir"
Printer.Print "Apt7 Floor1 Yenisehir/Nicosia"
Printer.Print "Tel:+90 392 227 0490"
Printer.Print "Tel:+90 392 227 5883"
Printer.Print ""
    .FontSize = 10
    yb = .CurrentY
    x1 = .TextWidth(Space(100))
    x2 = x1 + .TextWidth(Space(35))
    x3 = x2 + .TextWidth(Space(35))
Printer.Line (30, yb)-(x3 - 30, yb)
    .CurrentX = 30
    .FontBold = True
Printer.Print "Antenna Name";
    .CurrentX = x1 + 30
Printer.Print "Quantity";
    .CurrentX = x2 + 30
Printer.Print "Price $"
    .FontBold = False
Printer.Line (30, .CurrentY)-(x3 - 30, .CurrentY)
For i = 0 To List1.ListCount - 1
    .CurrentX = 30
Printer.Print List1.List(i);
    .CurrentX = x1 + 30
Printer.Print List2.List(i);
    .CurrentX = x2 + 30
Printer.Print List3.List(i)
Printer.Line (30, .CurrentY)-(x3 - 30, .CurrentY)
Next i
ys = .CurrentY
Printer.Line (30, yb)-(30, ys)
Printer.Line (x1 + 30, yb)-(x1 + 30, ys)
Printer.Line (x2 + 30, yb)-(x2 + 30, ys)
Printer.Line (x3 + 30, yb)-(x3 + 30, ys)
.EndDoc
End With
MsgBox "Document is Printing."
End Sub
-----
```

```
Private Sub Command2_Click()
```

Unload Me  
End Sub

---

```
Private Sub Form_Load()  
Data5.DatabaseName = App.Path & "\cwsca.mdb"  
Data5.Refresh  
If Data5.Recordset.RecordCount <> 0 Then  
    Do Until Data5.Recordset.EOF  
        List1.AddItem Data5.Recordset.aname & ""  
        List2.AddItem Data5.Recordset.unit & ""  
        List3.AddItem Data5.Recordset.price & ""  
        Data5.Recordset.MoveNext  
    Loop  
End If  
End Sub
```

#### **netincome.frm**

```
Dim dat As String  
Dim A1 As String  
Dim A2 As String  
Dim B1 As String  
Dim B2 As String  
Dim D11 As String  
Dim D12 As String  
Dim D21 As String  
Dim D22 As String  
Dim R1 As String  
Dim R2 As String  
Dim conv As String  
Option Explicit
```

---

```
Private Sub Command1_Click()  
List1.Clear  
List2.Clear  
List3.Clear  
List4.Clear  
List5.Clear  
List6.Clear  
List7.Clear  
List8.Clear  
List9.Clear  
List10.Clear  
List11.Clear  
If Combo2.Text <> "" Then  
    If Combo2.Text = "January" Then  
        dat = Combo1.Text + ".01"  
    End If  
    If Combo2.Text = "February" Then  
        dat = Combo1.Text + ".02"  
    End If
```

```

If Combo2.Text = "March" Then
    dat = Combo1.Text + ".03"
End If
If Combo2.Text = "April" Then
    dat = Combo1.Text + ".04"
End If
If Combo2.Text = "May" Then
    dat = Combo1.Text + ".05"
End If
If Combo2.Text = "June" Then
    dat = Combo1.Text + ".06"
End If
If Combo2.Text = "July" Then
    dat = Combo1.Text + ".07"
End If
If Combo2.Text = "August" Then
    dat = Combo1.Text + ".08"
End If
If Combo2.Text = "September" Then
    dat = Combo1.Text + ".09"
End If
If Combo2.Text = "October" Then
    dat = Combo1.Text + ".10"
End If
If Combo2.Text = "November" Then
    dat = Combo1.Text + ".11"
End If
If Combo2.Text = "December" Then
    dat = Combo1.Text + ".12"
End If
If Data6.Recordset.RecordCount = 0 Then
    Exit Sub
End If
Data6.Recordset.MoveFirst
Do Until Data6.Recordset.EOF
    D11 = Data6.Recordset![idate]
    R1 = Left(D11, 7)
    If R1 = dat Then
        List1.AddItem Data6.Recordset![ino]
        List2.AddItem Data6.Recordset![customer]
        List3.AddItem Data6.Recordset![company]
        conv = Data6.Recordset![idate]
        List4.AddItem Right(conv, 2) + Mid(conv, 5, 4) + Left(conv, 4)
        List5.AddItem Data6.Recordset![statuss]
        List6.AddItem Data6.Recordset![amount]
    End If
    Data6.Recordset.MoveNext
Loop
If Data7.Recordset.RecordCount = 0 Then
    Exit Sub

```



```

End If
Data7.Recordset.MoveFirst
Do Until Data7.Recordset.EOF
    D21 = Data7.Recordset![idate]
    R2 = Left(D21, 7)
    If R2 = dat Then
        List7.AddItem Data7.Recordset![ino]
        List8.AddItem Data7.Recordset![company]
        conv = Data7.Recordset![idate]
        List9.AddItem Right(conv, 2) + Mid(conv, 5, 4) + Left(conv, 4)
        List10.AddItem Data7.Recordset![status]
        List11.AddItem Data7.Recordset![amount]
    End If
Data7.Recordset.MoveNext
Loop
A1 = 0
A2 = 0
B1 = 0
B2 = 0
If Data6.Recordset.RecordCount = 0 And Data7.Recordset.RecordCount = 0 Then
    Text1 = ""
Exit Sub
End If
If Data6.Recordset.RecordCount <> 0 Then
    Data6.Recordset.MoveFirst
    Do Until Data6.Recordset.EOF
        D11 = Data6.Recordset![idate]
        R1 = Left(D11, 7)
        D12 = Data6.Recordset![status]
        If R1 = dat And D12 = "Paid" Then
            B1 = Data6.Recordset![amount]
            A1 = Val(A1) + Val(B1)
        End If
    Data6.Recordset.MoveNext
    Loop
End If
If Data7.Recordset.RecordCount <> 0 Then
    Data7.Recordset.MoveFirst
    Do Until Data7.Recordset.EOF
        D21 = Data7.Recordset![idate]
        R2 = Left(D21, 7)
        D22 = Data7.Recordset![status]
        If R2 = dat And D22 = "Paid" Then
            B2 = Data7.Recordset![amount]
            A2 = Val(A2) + Val(B2)
        End If
    Data7.Recordset.MoveNext
    Loop
End If
Text1 = Val(A1) - Val(A2)

```

```

Else
    dat = Combo1.Text
    If Data6.Recordset.RecordCount = 0 Then
        Exit Sub
    End If
    Data6.Recordset.MoveFirst
    Do Until Data6.Recordset.EOF
        D11 = Data6.Recordset![idate]
        R1 = Left(D11, 4)
        If R1 = dat Then
            List1.AddItem Data6.Recordset![ino]
            List2.AddItem Data6.Recordset![customer]
            List3.AddItem Data6.Recordset![company]
            conv = Data6.Recordset![idate]
            List4.AddItem Right(conv, 2) + Mid(conv, 5, 4) + Left(conv, 4)
            List5.AddItem Data6.Recordset![statuss]
            List6.AddItem Data6.Recordset![amount]
        End If
        Data6.Recordset.MoveNext
    Loop
    If Data7.Recordset.RecordCount = 0 Then
        Exit Sub
    End If
    Data7.Recordset.MoveFirst
    Do Until Data7.Recordset.EOF
        D21 = Data7.Recordset![idate]
        R2 = Left(D21, 4)
        If R2 = dat Then
            List7.AddItem Data7.Recordset![ino]
            List8.AddItem Data7.Recordset![company]
            conv = Data7.Recordset![idate]
            List9.AddItem Right(conv, 2) + Mid(conv, 5, 4) + Left(conv, 4)
            List10.AddItem Data7.Recordset![statuss]
            List11.AddItem Data7.Recordset![amount]
        End If
        Data7.Recordset.MoveNext
    Loop
    A1 = 0
    A2 = 0
    B1 = 0
    B2 = 0
    If Data6.Recordset.RecordCount = 0 And Data7.Recordset.RecordCount = 0 Then
        Text1 = ""
        Exit Sub
    End If
    If Data6.Recordset.RecordCount <> 0 Then
        Data6.Recordset.MoveFirst
        Do Until Data6.Recordset.EOF
            D11 = Data6.Recordset![idate]
            R1 = Left(D11, 4)

```

```

    D12 = Data6.Recordset![status]
    If R1 = dat And D12 = "Paid" Then
        B1 = Data6.Recordset![amount]
        A1 = Val(A1) + Val(B1)
    End If
    Data6.Recordset.MoveNext
Loop
End If
If Data7.Recordset.RecordCount <> 0 Then
    Data7.Recordset.MoveFirst
    Do Until Data7.Recordset.EOF
        D21 = Data7.Recordset![idate]
        R2 = Left(D21, 4)
        D22 = Data7.Recordset![status]
        If R2 = dat And D22 = "Paid" Then
            B2 = Data7.Recordset![amount]
            A2 = Val(A2) + Val(B2)
        End If
        Data7.Recordset.MoveNext
    Loop
End If
Text1 = Val(A1) - Val(A2)
End If
End Sub

```

---

```

Private Sub Command2_Click()
    Dim yb, ys, x1, x2, x3, x4, x5, i
    With Printer
        .FontSize = 12
        .FontName = "Courier New Tr"
        Printer.Print "Cyprus Wireless Solution Ltd."
        Printer.Print "26A Atatürk Avenue. Candemir"
        Printer.Print "Apt7 Floor1 Yenisehir/Nicosia"
        Printer.Print "Tel:+90 392 227 0490"
        Printer.Print "Tel:+90 392 227 5883"
        Printer.Print ""
        .FontSize = 10
        yb = .CurrentY
        x1 = .TextWidth(Space(30))
        x2 = x1 + .TextWidth(Space(30))
        x3 = x2 + .TextWidth(Space(35))
        x4 = x3 + .TextWidth(Space(30))
        Printer.Line (30, yb)-(x4 - 30, yb)
        .CurrentX = 30
        .FontBold = True
        Printer.Print "Invoice No";
        .CurrentX = x1 + 30
        Printer.Print "Date";
        .CurrentX = x2 + 30
        Printer.Print "Status";
    End With
End Sub

```



```

    .CurrentX = x3 + 30
Printer.Print "Amount $"
.FontBold = False
Printer.Line (30, .CurrentY)-(x4 - 30, .CurrentY)
For i = 0 To List1.ListCount - 1
    .CurrentX = 30
Printer.Print List1.List(i);
    .CurrentX = x1 + 30
Printer.Print List4.List(i);
    .CurrentX = x2 + 30
Printer.Print List5.List(i);
    .CurrentX = x3 + 30
Printer.Print "+"; List6.List(i)
Printer.Line (30, .CurrentY)-(x4 - 30, .CurrentY)
Next i
For i = 0 To List7.ListCount - 1
    .CurrentX = 30
Printer.Print List7.List(i);
    .CurrentX = x1 + 30
Printer.Print List9.List(i);
    .CurrentX = x2 + 30
Printer.Print List10.List(i);
    .CurrentX = x3 + 30
Printer.Print "-"; List11.List(i)
Printer.Line (30, .CurrentY)-(x4 - 30, .CurrentY)
Next i
ys = .CurrentY
Printer.Line (30, yb)-(30, ys)
Printer.Line (x1 + 30, yb)-(x1 + 30, ys)
Printer.Line (x2 + 30, yb)-(x2 + 30, ys)
Printer.Line (x3 + 30, yb)-(x3 + 30, ys)
Printer.Line (x4 + 30, yb)-(x4 + 30, ys)
Printer.Line (x5 + 30, yb)-(x5 + 30, ys)
If Combo2.Text <> "" Then
    .CurrentX = x2
    Printer.Print "Monthly Net Income:";
    .CurrentX = x3
    Printer.Print Text1.Text
End If
If Combo2.Text = "" Then
    .CurrentX = x2
    Printer.Print "Yearly Net Income:";
    .CurrentX = x3
    Printer.Print Text1.Text
End If
.EndDoc
End With
MsgBox "Document is Printing."
End Sub

```

```
Private Sub Command3_Click()
Unload Me
End Sub
```

---

```
Private Sub Form_Load()
Data7.DatabaseName = App.Path & "\cwsca.mdb"
Data6.DatabaseName = App.Path & "\cwsca.mdb"
End Sub
```

### **Revenue.frm**

```
Dim dat As String
Dim A As String
Dim B As String
Dim D1 As String
Dim D2 As String
Dim R As String
Dim conv As String
Option Explicit
```

---

```
Private Sub Command1_Click()
List1.Clear
List2.Clear
List3.Clear
List4.Clear
List5.Clear
List6.Clear
If Combo2.Text <> "" Then
    If Combo2.Text = "January" Then
        dat = Combo1.Text + ".01"
    End If
    If Combo2.Text = "February" Then
        dat = Combo1.Text + ".02"
    End If
    If Combo2.Text = "March" Then
        dat = Combo1.Text + ".03"
    End If
    If Combo2.Text = "April" Then
        dat = Combo1.Text + ".04"
    End If
    If Combo2.Text = "May" Then
        dat = Combo1.Text + ".05"
    End If
    If Combo2.Text = "June" Then
        dat = Combo1.Text + ".06"
    End If
    If Combo2.Text = "July" Then
        dat = Combo1.Text + ".07"
    End If
    If Combo2.Text = "August" Then
        dat = Combo1.Text + ".08"
```

```

End If
If Combo2.Text = "September" Then
    dat = Combo1.Text + ".09"
End If
If Combo2.Text = "October" Then
    dat = Combo1.Text + ".10"
End If
If Combo2.Text = "November" Then
    dat = Combo1.Text + ".11"
End If
If Combo2.Text = "December" Then
    dat = Combo1.Text + ".12"
End If
If Data6.Recordset.RecordCount = 0 Then
    Exit Sub
End If
Data6.Recordset.MoveFirst
Do Until Data6.Recordset.EOF
    D1 = Data6.Recordset![idate]
    R = Left(D1, 7)
    If R = dat Then
        List1.AddItem Data6.Recordset![ino]
        List2.AddItem Data6.Recordset![customer]
        List3.AddItem Data6.Recordset![company]
        conv = Data6.Recordset![idate]
        List4.AddItem Right(conv, 2) + Mid(conv, 5, 4) + Left(conv, 4)
        List5.AddItem Data6.Recordset![status]
        List6.AddItem Data6.Recordset![amount]
    End If
    Data6.Recordset.MoveNext
Loop
Text2 = ""
A = 0
B = 0
If Data6.Recordset.RecordCount = 0 Then
    Text1 = ""
    Text2 = ""
    Exit Sub
End If
Data6.Recordset.MoveFirst
Do Until Data6.Recordset.EOF
    D1 = Data6.Recordset![idate]
    R = Left(D1, 7)
    D2 = Data6.Recordset![status]
    If R = dat And D2 = "Paid" Then
        B = Data6.Recordset![amount]
        A = Val(A) + Val(B)
    End If
    Data6.Recordset.MoveNext
Loop

```



```

Text1 = A
Else
    dat = Combo1.Text
    If Data6.Recordset.RecordCount = 0 Then
        Exit Sub
    End If
    Data6.Recordset.MoveFirst
    Do Until Data6.Recordset.EOF
        D1 = Data6.Recordset![idate]
        R = Left(D1, 4)
        If R = dat Then
            List1.AddItem Data6.Recordset![ino]
            List2.AddItem Data6.Recordset![customer]
            List3.AddItem Data6.Recordset![company]
            conv = Data6.Recordset![idate]
            List4.AddItem Right(conv, 2) + Mid(conv, 5, 4) + Left(conv, 4)
            List5.AddItem Data6.Recordset![status]
            List6.AddItem Data6.Recordset![amount]
        End If
    Data6.Recordset.MoveNext
    Loop
    Text1 = ""
    A = 0
    B = 0
    If Data6.Recordset.RecordCount = 0 Then
        Text1 = ""
        Text2 = ""
        Exit Sub
    End If
    Data6.Recordset.MoveFirst
    Do Until Data6.Recordset.EOF
        D1 = Data6.Recordset![idate]
        R = Left(D1, 4)
        D2 = Data6.Recordset![status]
        If R = dat And D2 = "Paid" Then
            B = Data6.Recordset![amount]
            A = Val(A) + Val(B)
        End If
    Data6.Recordset.MoveNext
    Loop
    Text2 = A
End If
End Sub

```

---

```

Private Sub Command2_Click()
    Dim yb, ys, x1, x2, x3, x4, x5, i
    With Printer
        .FontSize = 12
        .FontName = "Courier New Tr"
    Printer.Print "Cyprus Wireless Solution Ltd."

```

```

Printer.Print "26A Atatürk Avenue. Candemir"
Printer.Print "Apt7 Floor1 Yenisehir/Nicosia"
Printer.Print "Tel:+90 392 227 0490"
Printer.Print "Tel:+90 392 227 5883"
Printer.Print ""
    .FontSize = 10
    yb = .CurrentY
    x1 = .TextWidth(Space(25))
    x2 = x1 + .TextWidth(Space(100))
    x3 = x2 + .TextWidth(Space(24))
    x4 = x3 + .TextWidth(Space(20))
    x5 = x4 + .TextWidth(Space(25))
Printer.Line (30, yb)-(x5 - 30, yb)
    .CurrentX = 30
    .FontBold = True
Printer.Print "Invoice No";
    .CurrentX = x1 + 30
Printer.Print "Customer Name";
    .CurrentX = x2 + 30
Printer.Print "Date";
    .CurrentX = x3 + 30
Printer.Print "Status";
    .CurrentX = x4 + 30
Printer.Print "Amount $"
    .FontBold = False
Printer.Line (30, .CurrentY)-(x5 - 30, .CurrentY)
For i = 0 To List1.ListCount - 1
    .CurrentX = 30
Printer.Print List1.List(i);
    .CurrentX = x1 + 30
Printer.Print List2.List(i);
    .CurrentX = x2 + 30
Printer.Print List4.List(i);
    .CurrentX = x3 + 30
Printer.Print List5.List(i);
    .CurrentX = x4 + 30
Printer.Print List6.List(i)
Printer.Line (30, .CurrentY)-(x5 - 30, .CurrentY)
Next i
ys = .CurrentY
Printer.Line (30, yb)-(30, ys)
Printer.Line (x1 + 30, yb)-(x1 + 30, ys)
Printer.Line (x2 + 30, yb)-(x2 + 30, ys)
Printer.Line (x3 + 30, yb)-(x3 + 30, ys)
Printer.Line (x4 + 30, yb)-(x4 + 30, ys)
Printer.Line (x5 + 30, yb)-(x5 + 30, ys)
If Combo2.Text <> "" Then
    .CurrentX = x2
    Printer.Print "Monthly Total Revenue:";
    .CurrentX = x4

```

```

Printer.Print Text1.Text
End If
If Combo2.Text = "" Then
    .CurrentX = x2
    Printer.Print "Yearly Total Revenue:";
    .CurrentX = x4
    Printer.Print Text2.Text
End If
.EndDoc
End With
MsgBox "Document is Printing."
End Sub

```

---

```

Private Sub Command3_Click()
Unload Me
End Sub

```

---

```

Private Sub Form_Load()
Data6.DatabaseName = App.Path & "\cwsca.mdb"
End Sub

```

#### **santenna.frm**

```

Private Sub Command1_Click()
If Text1.Text <> "" Then
    If Data5.Recordset.RecordCount = 0 Then
        Exit Sub
    End If
    Data5.Recordset.Edit
    Data5.Recordset.aname = Text1.Text
    Data5.Recordset.price = Text2.Text
    Data5.Recordset.unit = Text3.Text
    Data5.Recordset.alimit = Text4.Text
    Data5.Recordset.Update
    Data5.Refresh
Else
    MsgBox "You can not leave the Name empty.", 48, "Caution Message"
    If Text1.Text = "" Then
        Text1.SetFocus
    End If
End If
End Sub

```

---

```

Private Sub Command2_Click()
If Data5.Recordset.RecordCount = 0 Then Exit Sub
answer = MsgBox("Do you want to DELETE this record ?", vbQuestion + vbYesNo +
vbDefaultButton2, "Record Deleting")
If answer = vbYes Then Data5.Recordset.Delete
Data5.Refresh
End Sub

```



```
Private Sub Command3_Click()  
Unload Me  
End Sub
```

---

```
Private Sub Command4_Click()  
liste2.Show  
End Sub
```

---

```
Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)  
If Data5.Recordset.RecordCount = 0 Then  
    Text1 = ""  
    Text2 = ""  
    Text3 = ""  
    Text4 = ""  
    Exit Sub  
End If  
Text1 = Data5.Recordset.aname & ""  
Text2 = Data5.Recordset.price & ""  
Text3 = Data5.Recordset.unit & ""  
Text4 = Data5.Recordset.alimit & ""  
End Sub
```

---

```
Private Sub Form_Load()  
Data5.DatabaseName = App.Path & "\cwsca.mdb"  
squ = "SELECT * From antens WHERE aname LIKE ""  
squ = squ & Text5 & "*" order by aname ASC;"  
Data5.RecordSource = squ  
Data5.Refresh  
End Sub
```

---

```
Private Sub Text5_Change()  
squ = "SELECT * From antens WHERE aname LIKE ""  
squ = squ & Text5 & "*" order by aname ASC;"  
Data5.RecordSource = squ  
Data5.Refresh  
End Sub
```

#### **scompany.frm**

```
Private Sub Command1_Click()  
If Text2.Text <> "" Then  
    If Data2.Recordset.RecordCount = 0 Then  
        Exit Sub  
    End If  
    Data2.Recordset.Edit  
    Data2.Recordset.cname = Text2.Text  
    Data2.Recordset.tel = Text4.Text  
    Data2.Recordset.fax = Text5.Text  
    Data2.Recordset.branch = Text8.Text  
    Data2.Recordset.address = Text3.Text  
    Data2.Recordset.email = Text6.Text
```

```

Data2.Recordset.web = Text7.Text
Data2.Recordset.adddate = Date
Data2.Recordset.Update
Data2.Refresh
Else
MsgBox "You can not leave the Company Name empty.", 48, "Caution Message"
If Text2.Text = "" Then
    Text2.SetFocus
End If
End If
End Sub

-----

Private Sub Command2_Click()
If Data2.Recordset.RecordCount = 0 Then Exit Sub
answer = MsgBox("Do you want to DELETE this record ?", vbQuestion + vbYesNo +
vbDefaultButton2, "Record Deleting")
If answer = vbYes Then Data2.Recordset.Delete
Data2.Refresh
End Sub

-----

Private Sub Command3_Click()
Unload Me
End Sub

-----

Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
If Data2.Recordset.RecordCount = 0 Then
    Text2 = ""
    Text3 = ""
    Text4 = ""
    Text5 = ""
    Text6 = ""
    Text7 = ""
    Text8 = ""
    Exit Sub
End If
Text2 = Data2.Recordset.cname & ""
Text3 = Data2.Recordset.address & ""
Text4 = Data2.Recordset.tel & ""
Text5 = Data2.Recordset.fax & ""
Text6 = Data2.Recordset.email & ""
Text7 = Data2.Recordset.web & ""
Text8 = Data2.Recordset.branch & ""
End Sub

-----

Private Sub Form_Load()
Data2.DatabaseName = App.Path & "\cwsca.mdb"
squ = "SELECT * From company WHERE cname LIKE ""
squ = squ & Text1 & ""*'' order by cname ASC;"
Data2.RecordSource = squ
Data2.Refresh

```

End Sub

---

```
Private Sub Text1_Change()  
squ = "SELECT * From company WHERE cname LIKE "  
squ = squ & Text1 & "*" order by cname ASC;"  
Data2.RecordSource = squ  
Data2.Refresh  
End Sub
```

#### **scustomer.frm**

```
Private Sub Command1_Click()  
If Text2.Text <> "" And Text3.Text <> "" Then  
    If Data1.Recordset.RecordCount = 0 Then  
        Exit Sub  
    End If  
    Data1.Recordset.Edit  
    Data1.Recordset.service1 = Text1.Text  
    Data1.Recordset.cname = Text2.Text  
    Data1.Recordset.surname = Text3.Text  
    Data1.Recordset.address = Text4.Text  
    Data1.Recordset.mtel = Text5.Text  
    Data1.Recordset.tel = Text6.Text  
    Data1.Recordset.fax = Text7.Text  
    Data1.Recordset.email = Text8.Text  
    Data1.Recordset.it = Text9.Text  
    Data1.Recordset.company = Text12.Text  
    Data1.Recordset.ip = Text11.Text  
    Data1.Recordset.sid = Text10.Text  
    Data1.Recordset.unit2 = Text14.Text  
    Data1.Recordset.atype = Text17.Text  
    Data1.Recordset.unit1 = Text16.Text  
    Data1.Recordset.service2 = Text18.Text  
    Data1.Recordset.service3 = Text19.Text  
    Data1.Recordset.unit3 = Text20.Text  
    Data1.Recordset.Update  
    Data1.Refresh  
Else  
    MsgBox "You can not leave the Name and Surname empty.", 48, "Caution Message"  
    If Text2.Text = "" Then  
        Text2.SetFocus  
    End If  
    If Text3.Text = "" Then  
        Text3.SetFocus  
    End If  
End If  
End Sub
```

---

```
Private Sub Command2_Click()  
If Data1.Recordset.RecordCount = 0 Then Exit Sub
```



```

answer = MsgBox("Do you want to DELETE this record ?", vbQuestion + vbYesNo +
vbDefaultButton2, "Record Deleting")
If answer = vbYes Then Data1.Recordset.Delete
Data1.Refresh
End Sub

```

---

```

Private Sub Command3_Click()
Unload Me
End Sub

```

---

```

Private Sub Command4_Click()
liste1.Show
End Sub

```

---

```

Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
If Data1.Recordset.RecordCount = 0 Then
Text1 = ""
Text2 = ""
Text3 = ""
Text4 = ""
Text5 = ""
Text6 = ""
Text7 = ""
Text8 = ""
Text9 = ""
Text10 = ""
Text11 = ""
Text12 = ""
Text14 = ""
Text16 = ""
Text17 = ""
Text18 = ""
Text19 = ""
Text20 = ""
Exit Sub
End If
Text1 = Data1.Recordset.service1 & ""
Text2 = Data1.Recordset.cname & ""
Text3 = Data1.Recordset.surname & ""
Text4 = Data1.Recordset.address & ""
Text5 = Data1.Recordset.mtel & ""
Text6 = Data1.Recordset.tel & ""
Text7 = Data1.Recordset.fax & ""
Text8 = Data1.Recordset.email & ""
Text9 = Data1.Recordset.it & ""
Text10 = Data1.Recordset.sid & ""
Text11 = Data1.Recordset.ip & ""
Text12 = Data1.Recordset.company & ""
Text14 = Data1.Recordset.unit2 & ""
Text16 = Data1.Recordset.unit1 & ""

```

```

Text17 = Data1.Recordset.atype & ""
Text18 = Data1.Recordset.service2 & ""
Text19 = Data1.Recordset.service3 & ""
Text20 = Data1.Recordset.unit3 & ""
End Sub

```

---

```

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\cwsca.mdb"
squ = "SELECT * From customer WHERE cname LIKE ""
squ = squ & Text15 & ""*"" order by cname ASC;"
Data1.RecordSource = squ
Data1.Refresh
End Sub

```

---

```

Private Sub Text13_Change()
squ = "SELECT * From customer WHERE company LIKE ""
squ = squ & Text13 & ""*"" order by company ASC;"
Data1.RecordSource = squ
Data1.Refresh
End Sub

```

---

```

Private Sub Text15_Change()
squ = "SELECT * From customer WHERE cname LIKE ""
squ = squ & Text15 & ""*"" order by cname ASC;"
Data1.RecordSource = squ
Data1.Refresh
End Sub

```

### **sinvoise.frm**

```

Dim statu As String
Dim ans As String

```

---

```

Private Sub Command1_Click()
If Data6.Recordset.RecordCount = 0 Then
Exit Sub
End If
If Text3.Text <> "" And Text4.Text <> "" And Text5.Text <> "" Then
Data6.Recordset.Edit
Data6.Recordset.customer = Text3.Text
Data6.Recordset.company = Text4.Text
Data6.Recordset.ino = Text5.Text
dat = Right(Text6.Text, 4) + Mid(Text6.Text, 3, 4) + Left(Text6.Text, 2)
Data6.Recordset.idate = dat
Data6.Recordset.amount = Text34.Text
Data6.Recordset.statuss = statu
Data6.Recordset.disc = Text35.Text
Data6.Recordset.vat = Text36.Text
Data6.Recordset.q1 = Text7.Text
Data6.Recordset.q2 = Text11.Text
Data6.Recordset.q3 = Text15.Text

```

```

Data6.Recordset.q4 = Text19.Text
Data6.Recordset.q5 = Text23.Text
Data6.Recordset.q6 = Text27.Text
Data6.Recordset.D1 = Text8.Text
Data6.Recordset.D2 = Text12.Text
Data6.Recordset.d3 = Text16.Text
Data6.Recordset.d4 = Text20.Text
Data6.Recordset.d5 = Text24.Text
Data6.Recordset.d6 = Text28.Text
Data6.Recordset.Update
Data6.Refresh
Else
    MsgBox "You can not leave the Name, Company and Invoice No empty.", 48,
    "Caution Message"
    If Text3.Text = "" Then
        Text3.SetFocus
    End If
    If Text4.Text = "" Then
        Text4.SetFocus
    End If
    If Text5.Text = "" Then
        Text5.SetFocus
    End If
End If
End Sub

-----
Private Sub Command2_Click()
    If Data6.Recordset.RecordCount = 0 Then Exit Sub
    answer = MsgBox("Do you want to DELETE this record ?", vbQuestion + vbYesNo +
    vbDefaultButton2, "Record Deleting")
    If answer = vbYes Then Data6.Recordset.Delete
    Data6.Refresh
End Sub

-----
Private Sub Command3_Click()
    Unload Me
End Sub

-----
Private Sub Command4_Click()
    tot = Val(Text10) + Val(Text14) + Val(Text18) + Val(Text22) + Val(Text26) +
    Val(Text30)
    If Text35 = "" Then
        Text32 = 0
    End If
    Text32 = (Val(tot) * Val(Text35)) / 100
    Text31 = Val(tot) - Val(Text32)
    If Text36 = "" Then
        Text33 = ""
    End If
    Text33 = (Val(Text31) * Val(Text36)) / 100

```



```
Text34 = Val(Text31) + Val(Text33)
End Sub
```

---

```
Private Sub Command6_Click()
statu = "Paid"
Command6.Enabled = False
End Sub
```

---

```
Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
If Data6.Recordset.RecordCount = 0 Then
```

```
Text3 = ""
```

```
Text4 = ""
```

```
Text5 = ""
```

```
Text6 = ""
```

```
Text7 = ""
```

```
Text8 = ""
```

```
Text11 = ""
```

```
Text12 = ""
```

```
Text15 = ""
```

```
Text16 = ""
```

```
Text19 = ""
```

```
Text20 = ""
```

```
Text23 = ""
```

```
Text24 = ""
```

```
Text27 = ""
```

```
Text28 = ""
```

```
Text31 = ""
```

```
Text32 = ""
```

```
Text33 = ""
```

```
Text34 = ""
```

```
Text35 = ""
```

```
Text36 = ""
```

```
statu = "Not Paid"
```

```
Command6.Enabled = True
```

```
Exit Sub
```

```
End If
```

```
Text3 = Data6.Recordset.customer & ""
```

```
Text4 = Data6.Recordset.company & ""
```

```
Text5 = Data6.Recordset.ino & ""
```

```
dat = Data6.Recordset.idate & ""
```

```
Text6 = Right(dat, 2) + Mid(dat, 5, 4) + Left(dat, 4)
```

```
Text7 = Data6.Recordset.q1 & ""
```

```
Text8 = Data6.Recordset.D1 & ""
```

```
Text11 = Data6.Recordset.q2 & ""
```

```
Text12 = Data6.Recordset.D2 & ""
```

```
Text15 = Data6.Recordset.q3 & ""
```

```
Text16 = Data6.Recordset.d3 & ""
```

```
Text19 = Data6.Recordset.q4 & ""
```

```
Text20 = Data6.Recordset.d4 & ""
```

```
Text23 = Data6.Recordset.q5 & ""
```

```

Text24 = Data6.Recordset.d5 & ""
Text27 = Data6.Recordset.q6 & ""
Text28 = Data6.Recordset.d6 & ""
Text35 = Data6.Recordset.disc & ""
Text36 = Data6.Recordset.vat & ""
End Sub

```

---

```

Private Sub Form_Load()
statu = "Not Paid"
Data4.DatabaseName = App.Path & "\cwsca.mdb"
Data6.DatabaseName = App.Path & "\cwsca.mdb"
squ = "SELECT * From invoiced WHERE ino LIKE ""
squ = squ & Text37 & ""*'' order by ino ASC;"
Data6.RecordSource = squ
Data6.Refresh
End Sub

```

---

```

Private Sub Text1_Change()
squ = "SELECT * From invoiced WHERE customer LIKE ""
squ = squ & Text1 & ""*'' order by customer ASC;"
Data6.RecordSource = squ
Data6.Refresh
End Sub

```

---

```

Private Sub Text11_Change()
Text14 = Val(Text11) * Val(Text13)
If Text11 = "" And Text13 = "" Then
    Text14 = ""
End If
End Sub

```

---

```

Private Sub Text12_Change()
If Data4.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text12.Text
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text13 = Data4.Recordset![price]
If Text12 = "" Then
    Text13 = ""
End If
End Sub

```

---

```

Private Sub Text13_Change()
Text14 = Val(Text11) * Val(Text13)
If Text11 = "" And Text13 = "" Then
    Text14 = ""
End If
End Sub

```

```

Private Sub Text15_Change()
Text18 = Val(Text15) * Val(Text17)
If Text15 = "" And Text17 = "" Then
    Text18 = ""
End If
End Sub

```

---

```

Private Sub Text16_Change()
If Data4.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text16.Text
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text17 = Data4.Recordset![price]
If Text16 = "" Then
    Text17 = ""
End If
End Sub

```

---

```

Private Sub Text17_Change()
Text18 = Val(Text15) * Val(Text17)
If Text15 = "" And Text17 = "" Then
    Text18 = ""
End If
End Sub

```

---

```

Private Sub Text19_Change()
Text22 = Val(Text19) * Val(Text21)
If Text19 = "" And Text21 = "" Then
    Text22 = ""
End If
End Sub

```

---

```

Private Sub Text2_Change()
squ = "SELECT * From invoiced WHERE company LIKE '"
squ = squ & Text2 & "*" order by company ASC;"
Data6.RecordSource = squ
Data6.Refresh
End Sub

```

---

```

Private Sub Text20_Change()
If Data4.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text20.Text
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text21 = Data4.Recordset![price]
If Text20 = "" Then

```



```
Text21 = ""  
End If  
End Sub
```

---

```
Private Sub Text21_Change()  
Text22 = Val(Text19) * Val(Text21)  
If Text19 = "" And Text21 = "" Then  
Text22 = ""  
End If  
End Sub
```

---

```
Private Sub Text23_Change()  
Text26 = Val(Text23) * Val(Text25)  
If Text23 = "" And Text25 = "" Then  
Text26 = ""  
End If  
End Sub
```

---

```
Private Sub Text24_Change()  
If Data4.Recordset.RecordCount = 0 Then  
Exit Sub  
End If  
A = Text24.Text  
B = "pname=" & A & ""  
Data4.Recordset.FindFirst B  
Text25 = Data4.Recordset![price]  
If Text24 = "" Then  
Text25 = ""  
End If  
End Sub
```

---

```
Private Sub Text25_Change()  
Text26 = Val(Text23) * Val(Text25)  
If Text23 = "" And Text25 = "" Then  
Text26 = ""  
End If  
End Sub
```

---

```
Private Sub Text27_Change()  
Text30 = Val(Text27) * Val(Text29)  
If Text27 = "" And Text29 = "" Then  
Text30 = ""  
End If  
End Sub
```

---

```
Private Sub Text28_Change()  
If Data4.Recordset.RecordCount = 0 Then  
Exit Sub  
End If  
A = Text28.Text
```

```

B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text29 = Data4.Recordset![price]
If Text28 = "" Then
    Text29 = ""
End If
End Sub

```

---

```

Private Sub Text29_Change()
Text30 = Val(Text27) * Val(Text29)
If Text27 = "" And Text29 = "" Then
    Text30 = ""
End If
End Sub

```

---

```

Private Sub Text37_Change()
squ = "SELECT * From invoiced WHERE ino LIKE '"
squ = squ & Text37 & "' order by ino ASC;"
Data6.RecordSource = squ
Data6.Refresh
End Sub

```

---

```

Private Sub Text5_Change()
If Data6.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text5.Text
B = "ino=" & A & ""
Data6.Recordset.FindFirst B
ans = Data6.Recordset![status]
If ans = "Paid" Then
    Command6.Enabled = False
    statu = "Paid"
Else
    Command6.Enabled = True
    statu = "Not Paid"
End If
End Sub

```

---

```

Private Sub Text7_Change()
Text10 = Val(Text7) * Val(Text9)
If Text7 = "" And Text9 = "" Then
    Text10 = ""
End If
End Sub

```

---

```

Private Sub Text8_Change()
If Data4.Recordset.RecordCount = 0 Then
    Exit Sub
End If

```

```

A = Text8.Text
B = "pname=" & A & ""
Data4.Recordset.FindFirst B
Text9 = Data4.Recordset![price]
If Text8 = "" Then
    Text9 = ""
End If
End Sub

```

```

-----
Private Sub Text9_Change()
Text10 = Val(Text7) * Val(Text9)
If Text7 = "" And Text9 = "" Then
    Text10 = ""
End If
End Sub

```

#### **sinvoicee.frm**

```

Dim statu As String
Dim ans As String

```

```

-----
Private Sub Command1_Click()
If Data7.Recordset.RecordCount = 0 Then
    Exit Sub
End If
If Text4.Text <> "" And Text5.Text <> "" Then
    Data7.Recordset.Edit
    Data7.Recordset.company = Text4.Text
    Data7.Recordset.ino = Text5.Text
    dat = Right(Text6.Text, 4) + Mid(Text6.Text, 3, 4) + Left(Text6.Text, 2)
    Data7.Recordset.idate = dat
    Data7.Recordset.amount = Text34.Text
    Data7.Recordset.statuss = statu
    Data7.Recordset.disc = Text35.Text
    Data7.Recordset.vat = Text36.Text
    Data7.Recordset.q1 = Text7.Text
    Data7.Recordset.q2 = Text11.Text
    Data7.Recordset.q3 = Text15.Text
    Data7.Recordset.q4 = Text19.Text
    Data7.Recordset.q5 = Text23.Text
    Data7.Recordset.q6 = Text27.Text
    Data7.Recordset.D1 = Text8.Text
    Data7.Recordset.D2 = Text12.Text
    Data7.Recordset.d3 = Text16.Text
    Data7.Recordset.d4 = Text20.Text
    Data7.Recordset.d5 = Text24.Text
    Data7.Recordset.d6 = Text28.Text
    Data7.Recordset.Update
    Data7.Refresh
Else

```



MsgBox "You can not leave the Company and Invoice No empty.", 48, "Caution  
Message"

```
If Text4.Text = "" Then
    Text4.SetFocus
End If
If Text5.Text = "" Then
    Text5.SetFocus
End If
End If
End Sub
```

---

```
Private Sub Command2_Click()
If Data7.Recordset.RecordCount = 0 Then Exit Sub
answer = MsgBox("Do you want to DELETE this record ?", vbQuestion + vbYesNo +
vbDefaultButton2, "Record Deleting")
If answer = vbYes Then Data7.Recordset.Delete
Data7.Refresh
End Sub
```

---

```
Private Sub Command3_Click()
Unload Me
End Sub
```

---

```
Private Sub Command4_Click()
tot = Val(Text10) + Val(Text14) + Val(Text18) + Val(Text22) + Val(Text26) +
Val(Text30)
If Text35 = "" Then
    Text32 = 0
End If
Text32 = (Val(tot) * Val(Text35)) / 100
Text31 = Val(tot) - Val(Text32)
If Text36 = "" Then
    Text33 = ""
End If
Text33 = (Val(Text31) * Val(Text36)) / 100
Text34 = Val(Text31) + Val(Text33)
End Sub
```

---

```
Private Sub Command6_Click()
statu = "Paid"
Command6.Enabled = False
End Sub
```

---

```
Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
If Data7.Recordset.RecordCount = 0 Then
    Text4 = ""
    Text5 = ""
    Text6 = ""
    Text7 = ""
    Text8 = ""

```

```

Text9 = ""
Text11 = ""
Text12 = ""
Text13 = ""
Text15 = ""
Text16 = ""
Text17 = ""
Text19 = ""
Text20 = ""
Text21 = ""
Text23 = ""
Text24 = ""
Text25 = ""
Text27 = ""
Text28 = ""
Text29 = ""
Text31 = ""
Text32 = ""
Text33 = ""
Text34 = ""
Text35 = ""
Text36 = ""
statu = "Not Paid"
Command6.Enabled = True
Exit Sub
End If
Text4 = Data7.Recordset.company & ""
Text5 = Data7.Recordset.ino & ""
dat = Data7.Recordset.idate & ""
Text6 = Right(dat, 2) + Mid(dat, 5, 4) + Left(dat, 4)
Text7 = Data7.Recordset.q1 & ""
Text8 = Data7.Recordset.D1 & ""
Text11 = Data7.Recordset.q2 & ""
Text12 = Data7.Recordset.D2 & ""
Text15 = Data7.Recordset.q3 & ""
Text16 = Data7.Recordset.d3 & ""
Text19 = Data7.Recordset.q4 & ""
Text20 = Data7.Recordset.d4 & ""
Text23 = Data7.Recordset.q5 & ""
Text24 = Data7.Recordset.d5 & ""
Text27 = Data7.Recordset.q6 & ""
Text28 = Data7.Recordset.d6 & ""
Text35 = Data7.Recordset.disc & ""
Text36 = Data7.Recordset.vat & ""
Text9 = Data7.Recordset.u1 & ""
Text13 = Data7.Recordset.u2 & ""
Text17 = Data7.Recordset.u3 & ""
Text21 = Data7.Recordset.u4 & ""
Text25 = Data7.Recordset.u5 & ""
Text29 = Data7.Recordset.u6 & ""

```

End Sub

---

```
Private Sub Form_Load()  
statu = "Not Paid"  
Data7.DatabaseName = App.Path & "\cwsca.mdb"  
squ = "SELECT * From invoicee WHERE ino LIKE "  
squ = squ & Text37 & "*" order by ino ASC;"  
Data7.RecordSource = squ  
Data7.Refresh  
End Sub
```

---

```
Private Sub Text11_Change()  
Text14 = Val(Text11) * Val(Text13)  
If Text11 = "" And Text13 = "" Then  
    Text14 = ""  
End If  
End Sub
```

---

```
Private Sub Text13_Change()  
Text14 = Val(Text11) * Val(Text13)  
If Text11 = "" And Text13 = "" Then  
    Text14 = ""  
End If  
End Sub
```

---

```
Private Sub Text15_Change()  
Text18 = Val(Text15) * Val(Text17)  
If Text15 = "" And Text17 = "" Then  
    Text18 = ""  
End If  
End Sub
```

---

```
Private Sub Text17_Change()  
Text18 = Val(Text15) * Val(Text17)  
If Text15 = "" And Text17 = "" Then  
    Text18 = ""  
End If  
End Sub
```

---

```
Private Sub Text19_Change()  
Text22 = Val(Text19) * Val(Text21)  
If Text19 = "" And Text21 = "" Then  
    Text22 = ""  
End If  
End Sub
```

---

```
Private Sub Text2_Change()  
squ = "SELECT * From invoicee WHERE company LIKE "  
squ = squ & Text2 & "*" order by company ASC;"  
Data7.RecordSource = squ
```

```
Data7.Refresh
End Sub
Private Sub Text21_Change()
Text22 = Val(Text19) * Val(Text21)
If Text19 = "" And Text21 = "" Then
    Text22 = ""
End If
End Sub
```

---

```
Private Sub Text23_Change()
Text26 = Val(Text23) * Val(Text25)
If Text23 = "" And Text25 = "" Then
    Text26 = ""
End If
End Sub
```

---

```
Private Sub Text25_Change()
Text26 = Val(Text23) * Val(Text25)
If Text23 = "" And Text25 = "" Then
    Text26 = ""
End If
End Sub
```

---

```
Private Sub Text27_Change()
Text30 = Val(Text27) * Val(Text29)
If Text27 = "" And Text29 = "" Then
    Text30 = ""
End If
End Sub
```

---

```
Private Sub Text29_Change()
Text30 = Val(Text27) * Val(Text29)
If Text27 = "" And Text29 = "" Then
    Text30 = ""
End If
End Sub
```

---

```
Private Sub Text37_Change()
squ = "SELECT * From invoicee WHERE ino LIKE '"
squ = squ & Text37 & "' order by ino ASC;"
Data7.RecordSource = squ
Data7.Refresh
End Sub
```

---

```
Private Sub Text5_Change()
If Data7.Recordset.RecordCount = 0 Then
    Exit Sub
End If
A = Text5.Text
B = "ino=" & A & ""
```



```

Data7.Recordset.FindFirst B
ans = Data7.Recordset![status]
If ans = "Paid" Then
    Command6.Enabled = False
    statu = "Paid"
Else
    Command6.Enabled = True
    statu = "Not Paid"
End If
End Sub

```

---

```

Private Sub Text7_Change()
Text10 = Val(Text7) * Val(Text9)
If Text7 = "" And Text9 = "" Then
    Text10 = ""
End If
End Sub

```

---

```

Private Sub Text9_Change()
Text10 = Val(Text7) * Val(Text9)
If Text7 = "" And Text9 = "" Then
    Text10 = ""
End If
End Sub

```

#### **sit.frm**

```

Private Sub Command1_Click()
If Text2.Text <> "" Then
    If Data3.Recordset.RecordCount = 0 Then
        Exit Sub
    End If
    Data3.Recordset.Edit
    Data3.Recordset.cname = Text2.Text
    Data3.Recordset.mtel = Text4.Text
    Data3.Recordset.tel = Text5.Text
    Data3.Recordset.email = Text6.Text
    Data3.Recordset.Update
    Data3.Refresh
Else
    MsgBox "You can not leave the Name empty.", 48, "Caution Message"
    If Text2.Text = "" Then
        Text2.SetFocus
    End If
End If
End Sub

```

---

```

Private Sub Command2_Click()
If Data3.Recordset.RecordCount = 0 Then Exit Sub
answer = MsgBox("Do you want to DELETE this record ?", vbQuestion + vbYesNo +
vbDefaultButton2, "Record Deleting")

```

```

If answer = vbYes Then Data3.Recordset.Delete
Data3.Refresh
End Sub
Private Sub Command4_Click()
Unload Me
End Sub

```

```

-----
Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
If Data3.Recordset.RecordCount = 0 Then
    Text2 = ""
    Text4 = ""
    Text5 = ""
    Text6 = ""
Exit Sub
End If
Text2 = Data3.Recordset.cname & ""
Text4 = Data3.Recordset.mtel & ""
Text5 = Data3.Recordset.tel & ""
Text6 = Data3.Recordset.email & ""
End Sub

```

```

-----
Private Sub Form_Load()
Data3.DatabaseName = App.Path & "\cwsca.mdb"
squ = "SELECT * From it WHERE cname LIKE ""
squ = squ & Text1 & "*" order by cname ASC;"
Data3.RecordSource = squ
Data3.Refresh
End Sub

```

```

-----
Private Sub Text1_Change()
squ = "SELECT * From it WHERE cname LIKE ""
squ = squ & Text1 & "*" order by cname ASC;"
Data3.RecordSource = squ
Data3.Refresh
End Sub

```

#### **sservice.frm**

```

Private Sub Command1_Click()
Unload Me
End Sub

```

```

-----
Private Sub Command2_Click()
If Data4.Recordset.RecordCount = 0 Then Exit Sub
answer = MsgBox("Do you want to DELETE this record ?", vbQuestion + vbYesNo +
vbDefaultButton2, "Record Deleting")
If answer = vbYes Then Data4.Recordset.Delete
Data4.Refresh
End Sub

```

```

-----
Private Sub Command3_Click()

```

```

If Text2.Text <> "" Then
    If Data4.Recordset.RecordCount = 0 Then
        Exit Sub
    End If
    Data4.Recordset.Edit
    Data4.Recordset.pname = Text2.Text
    Data4.Recordset.price = Text4.Text
    Data4.Recordset.Update
    Data4.Refresh
Else
    MsgBox "You can not leave the Name empty.", 48, "Caution Message"
    If Text2.Text = "" Then
        Text2.SetFocus
    End If
End If
End Sub

-----
Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
If Data4.Recordset.RecordCount = 0 Then
    Text2 = ""
    Text4 = ""
    Exit Sub
End If
Text2 = Data4.Recordset.pname & ""
Text4 = Data4.Recordset.price & ""
End Sub

-----
Private Sub Form_Load()
Data4.DatabaseName = App.Path & "\cwsca.mdb"
squ = "SELECT * From service WHERE pname LIKE ""
squ = squ & Text1 & ""*'' order by pname ASC;"
Data4.RecordSource = squ
Data4.Refresh
End Sub

-----
Private Sub Text1_Change()
squ = "SELECT * From service WHERE pname LIKE ""
squ = squ & Text1 & ""*'' order by pname ASC;"
Data4.RecordSource = squ
Data4.Refresh
End Sub

```