

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

INTERNET CAFÉ RESERVATION PROGRAM

**Graduation Project
COM- 400**

Student: Ibrahim Dokuzer (980869)

Supervisor: Miss. Besime Erin

Nicosia - 2002

ACKNOWLEDGMENT

First I am glad to complete my project, which I had given with blessing of God and also I am grateful to my brother and my family, for their encouragement during the preparation of this project.

I would like to thank my supervisor, Miss Besime Erin, invaluable advice and belief in my work and myself over the course of this Graduation.

Also thank all my teachers for their advice, opinion and support. The directives of these teachers prepared me to perform this task successfully.

Finally, I would also like to thank all my friends for their advice and support for my project.

ABSTRACT

The automation is an important part of the production in productive business. The business executives have to be pay attention to this level of business. Automating some of production is having different meanings in different business fields.

The Internet is a computer network made up of thousands of networks worldwide. No one knows exactly how many computers are connected to the Internet. It is certain, however, that these number in the millions and are increasing at a rapid rate.

No one is in charge of the Internet. There are organizations, which develop technical aspects of this network and set standards for creating applications on it, but no governing body is in control. Private companies own the Internet backbone, through which Internet traffic flows.

Internet café's are the new growing business field in today's market. Because they are serving Internet connection that is popular technological product in these days. As in other business fields, the Internet café's can be atomized for more productivity by a reservation system. Internet connection requires some devices as modems on a personal computer but for an Internet cafe, the hubs, Ethernet cards are needed with a dial-up connected computer and terminal computers. For an Internet cafe that requirements must be satisfied.

To organize the well-atomized Internet cafe we need well operating reservation program on computers in cafe. Reservation program realizes all operations well settled and faster.

The important point is controlling user's time information (service start and stop times) and other services used with Internet service. In a local network as in Internet Café (also as in Internet backbone) the server-client relation can be used for reservation programs. This means that the remote server machine provides files and services to the user's local client machine. Software can be installed on a client computer to take advantage of the latest access technology.

We can use a static application as in this project to solve this problem also again using client-server architecture.

In Internet cafe reservation program we have to use some visualizations and controls for customer satisfaction. So we can use graphical interfaces for more comprehensible programs to users.

Reservation programs mainly deals with basic information of user's as times and other services so this information has to be safely stored. That is a security problem to prevent unauthorized actions for the programs and eliminating escapes. The programs must not effect the other applications and must not slow down system.

TABLE OF CONTENTS

ACKNOWLEDGMENT	i
ABSTRACT	ii
TABLE OF CONTENTS	iii
INTRODUCTION	1
CHAPTER ONE: DEFINITION, PROPERTIES AND HISTORY OF VISUAL BASIC	2
1.1. Visual Basic Integrated Development Environment (IDE)	3
CHAPTER TWO: CHARACTERISTICS OF PROGRAM	6
2.1. The Main Properties of Server Program	6
2.2. The Main Properties of Client Program	6
2.3. Advantages of Software	6
CHAPTER THREE: STRUCTURE OF THE PROGRAM	7
3.1. Main Program (Server)	7
3.1.1. Password Verification for Enter Server Program	7
3.1.2. Permission process of server program	8
3.1.3. Changing Unit Prices of Services Process	9
3.1.4. Recording Orders and Calculate Charges Process	10
3.1.5. Calculation of Total Charges of Users Process	12
3.1.6. Reporting Process of server part of program	13
3.1.7. Transferring Time and Orders Data's of users to Report Database Process	14
3.2. Client Program(Terminal)	15
3.2.1. Password Verification for Client Program	15
3.2.2. Determining and Recording Starting and Stopping Time of User Process	16
3.2.3 Visibility Control of Start button Process	18
3.2.4. Sending Data to Report Database Process	19
CHAPTER FOUR: ADDITIONAL APPLICATIONS	20
4.1. CPU Performance Test Application	20
4.2. Alarm	23
4.3. Using another program on Client Program Process	24
4.4. Clock Application	26
4.5. Calender Application	27
4.6. Minimization of Client Program Process	27
CHAPTER FIVE: USED API'S AND COMPONENTS	29
5.1. Notifying Program Icon In System Tray	29
5.2. Hiding Program from Windows Task List	30
CONCLUSION	31
REFERENCES	32
APPENDIX A	33
APPENDIX B	58

INTRODUCTION

The Internet Cafe's are the new business fields in growing global market and they are serving Internet to their customers with a known cafeteria function. After Internet has started to be widespread over the world, more people have intrested with its facility in working. Because Internet was a global network that connects any distance. Peoples realized many useful services and more specific are designed with Internet.

In these days Internet had a global meaning because it permits to reach any information on computers connected in the world. By development of faster Internet communication and computers systems, it became a common source of information. Dial up connections are the frequent connection types for standard users. Still some computers are not connected to Internet because of requirements as a communication line, modem and Internet Service Providers (ISP). These requirements cause extra cost. In this respect Internet connectivity became a little difficult for computer users. Also Internet was having a restrictive structure for connection. The mobility has to be started having importance for Internet. But again mobile devices are still having very high prices.

Internet café's solves some problems by bringing today's personal computers to the public with Internet connections. Internet café's are sheltering the number of computers connected to Internet with traditional café service. For this reason Internet café's still has a growing popularity in market and the executives of this business are in a competition for giving a high quality service. Also need for Internet service provider's accelerated. To offset this demand, the Internet Cafe retooled and rededicated itself to providing the best Internet access.

As in many other business reservation has a great meaning to increasing production and revenues. At beginning they used some simple methods as ticketing. But that was not a conventional way of reservation because it has some security problems as escapes and operational costs. Then the computer programmers have developed some programs to coordinate operations. They prepared some programs that records the users times and calculates charges for services. Also operators are not needed to control one each of users. They have been able to control connections of the specific computers without dealing users or computers.

Programs are designed in Visual Basic programming language, which provides to create user interfaces for Windows. To simplify the task, Microsoft has built in to Visual Basic a software program to help you write your Visual Basic projects. Visual Basic provides more of the actual code for a programmer than any other non-visual programming language.

Visual Basic version 6.0 brings many new developing tools and controls to create more powerful applications. So the programs can easily be adapt to Windows operating system. Both programs are done by using Visual Basic language characteristics to organize entire network in the Internet café.

In this project I intended to design a specific program that performs all user transactions for Internet café.

1.DEFINITION, PROPERTIES AND HISTORY OF VISUAL BASIC

Visual Basic is a much-enhanced version of the BASIC programming language and the BASIC Integrated Development Environment (IDE). The bottom line of the enhancement is the VB can create Windows programs whereas BASIC could only create DOS programs. Ok, so the modifications are very major, but the idea holds true that Visual Basic is BASIC for Windows.

One of the many significant improvements is that VB provides massive support for easily creating the user interface to your applications. This is accomplished within the VB Integrated Development Environment (IDE), in which you use a mouse to "draw" your application and use the keyboard to type in the code that is to be executed. The single largest effect on coding that VB introduced was the concept of an event-driven programming model. In the old BASIC you had to write code to watch for the occurrence of user events (pressing a key, using the mouse). VB performs that function for you, and in fact, the only time code will execute in VB is in response to such an event!

And finally, the other major concept that VB has incorporated is the concept of objects. Objects provides a way to link together both code and data into a "package" in such a way as to make handling and saving the code/data more intuitively. VB forms are objects, menus are objects, and the so are the intrinsic VB controls.

VB has also provided a wide variety of built-in code that programmers once had to handle themselves. Of most significance is the built-in database handling features of VB. It is generally accepted that over half of all VB applications are written to handle databases! You'll find the built-in database features of VB to be very powerful, and that you can tap into them at whatever level of programming skill you possess.

A second area in which VB has begun to provide built-in support is that of Internet access. The VB features are still maturing, but with the tools available you can create very useful applications.

History of Visual Basic:

Basic was an introducing language for a new programmer because of it's easy to understand structure and commands. Language developed in early 1960's at **BASIC** (**B**eginner's **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode).

Answer to complicated programming languages (FORTRAN, Algol, Cobol...). First time-share language. In the mid-1970's, two college students write first Basic for a microcomputer (Altair) - cost \$350 on cassette tape. Every Basic since then essentially based on that early version. Examples include: GW-Basic, QBasic, and QuickBasic. Visual Basic was introduced in 1991. But nowadays Visual Basic is a high capacity language that the programmers can create professional applications and it is still has an easy to understand structure.

Starting with VB5, Visual Basic became an exclusively 32-bit programming language, suitable for programming only Win9X or NT systems. If you must program for Win 3.x, then you'll have to drop back to either VB3 or VB4, both of which are in pretty short supply. VB4 had the dual ability to support Win3.x as well as Win9X/NT systems but my personal recommendation is that if you need 32-bit system support, go straight to VB6 and if 16-bit is your need then stick with VB3.

The VB Learning Edition is the most affordable, and truth is that you can do a lot with it, particularly if you use the Windows API to augment its capabilities. However, in light of its better database features and its greater variety of controls, I suggest you go straight to the Professional Edition if at all possible. The price is steep, but it really does pay itself back in terms of time savings. If you need the VB Enterprise edition then you

should have it paid for by the "Enterprise" which requires it. Individual programmers generally do not need the Enterprise edition.

Programming Basics

To toss in a bit of philosophy, I view a programmer's job as getting to the end product with a minimum of effort. When you're paid by the hour you have an obligation to your employer (even if it's yourself) to minimize the cost/schedule of completing the assignment. To that end, I will emphasize over and over again the importance of using existing capabilities (such as your own reusable code, VB-provided controls, or even controls that you purchase). In light of this optimize-your-time philosophy, I also believe that every VB programmer should strive to become an expert in the following two areas: Databases and Reporting. It's very common for applications to store data and to provide that data in a printed format (by some estimates, 80% of all VB applications use databases). And while this tutorial will discuss various ways to perform these tasks, most programmers will find that the capabilities of VB for creating/editing Access (a Microsoft product) databases and VB's capabilities for reporting that data (as exemplified by the Crystal Reports control or the newer built-in VB reporting features) are the most effective tools you can use for increasing your effectiveness in creating VB applications.

That tools which provide similar functions will be one of the most commonly used tools of those available to a programmer and that any serious programmer must take the time to develop strong skills with these tools.

1.1. Visual Basic Integrated Development Environment (IDE)

It's a little known fact, but you can write a VB program from scratch using nothing more than a simple text editor, such as the Edit or Notepad applications, which come with Windows. In fact, the Visual Basic project files are exactly that text files. However, writing a project from scratch would involve a lot of tedious, detailed manual entries. To simplify the task, Microsoft has built in to VB a software program to help you write your VB projects. That software, known as the Integrated Development Environment (IDE for short) is what comes to the screen when you start VB and is the topic of this section.

Elements of Visual Basic

Like any other Windows application, VB consists of multiple windows, which appear at startup. The windows that are displayed when you start VB are collectively known as the Visual Basic Integrated Development Environment (IDE). When you first start VB all of the windows are locked together in what is called the MDI format. I prefer the SDI format (which you can set in the options menu), which allows each of the windows to be positioned independently on your screen. Here's a sample IDE screen which shows a VB project with one form on which is a single command button.

In particular, VB has the following windows:

Menu / Toolbar:

This is the only element of the IDE, which is always visible. You use it to select which other IDE elements to view and to add forms or controls to your project. There are many other features, which we will discuss later.



Figure 1.1. Menu/Toolbar of Visual Basic

Toolbox

The toolbox is simply a library of controls, which you can place on your application. Once you've placed all the controls you need onto your applications forms, you can hide the toolbox to make room for working in the other elements of the IDE.

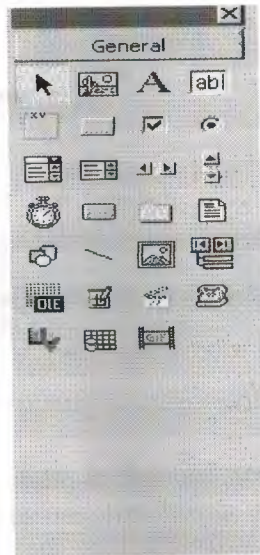


Figure 1.2.Toolbox of Visual Basic

Project Window:

This is simply a list of all the forms, which make up your VB project.

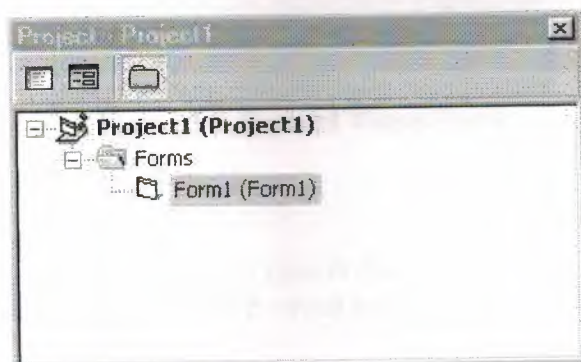


Figure 1.3. Project Window of Visual Basic

Property Window:

We'll talk about controls later, but such things as push buttons, scrolling text boxes, pictures boxes and other features of most VB applications allow you to enter parameters, which define how these controls work. In VB, these parameters are called properties. Some properties can be entered at design time within the IDE, while others must be entered with code while the program is running.

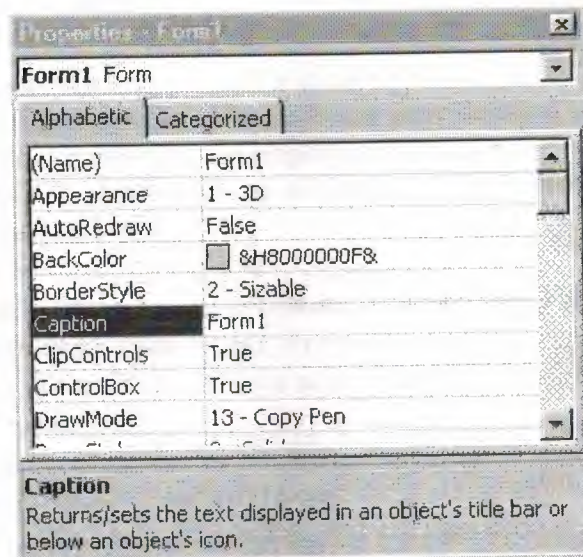


Figure 1.4. Property Windows of Visual Basic

Forms:

You add these to your VB application, as they are needed. They are the windows, which hold the various controls (buttons, text boxes, etc.), which make up your application.

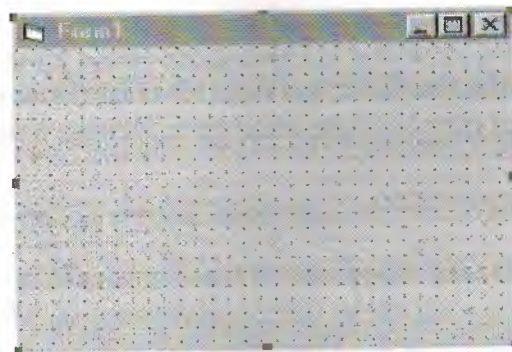


Figure 1.5. Form of Visual Basic

Code Window:

Like its name implies, this is where you type in the code that VB executes. Notice that the heading of the window indicates with which event the code is associated.

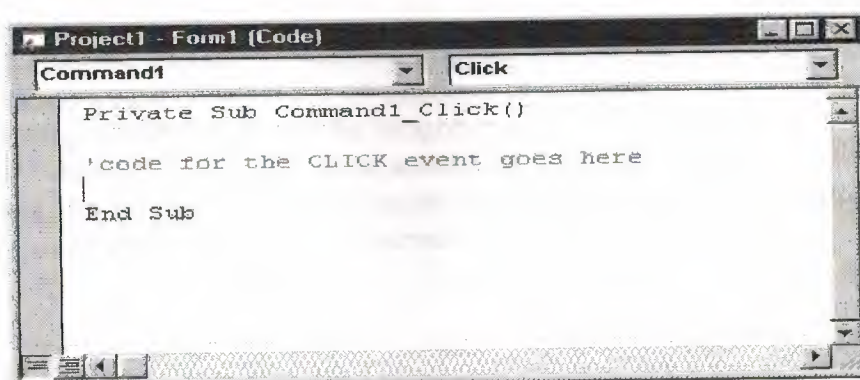


Figure 1.6. Code Window of Visual Basic

2.CHARACTERISTICS OF PROGRAM

These programs created for internet café's at all types and capacity. It makes easy to manage the internet café enterprises. That software provides to cut down the all loses and security problem that created by the employers of the enterprises. Also has a major advantage that is reduce the amount of the workers in that company. Because this software creates an automation.

This software has a server-client system.

- These separately programs communicate with each other owing to database.
- The main program (Server) works on at administrator computer.
- Each two part of the program has a simple interface for achieve easy to use and increase under stability.
- Terminal program are installed to whole computers, it shows time situation of each computer with start and stop timer.

The communication of these two program packet are very ordinary. Its communicate with each other over network. They connected each two program to database. The client (terminal) programs send or read some data from database. Also server (main) program does same operation.

2.1.The Main Properties of Server Program

Server program takes time data from database that is sent by the client program. It calculates order charges with multiply unit price of charges. Also it gives the permission to open or close the program whole computers for access to internet. Server program generates all time and orders reports for each individual computer. Server program has an administrative because it can reflect all users start and stop times can change hourly rates and clear accounts.

2.2.The Main Properties of Client Program

This software runs at whole computers. The user sends start time, stop time, elapsed time and date data's of the customer transactions to the main program from this client programs. The client program also has additional programs clock, alarm, calendar etc.

2.3.Advantages of Software

These software's has following advantages:

1. Operators can see all terminal computers start times, stop times, date of transactions from the server program.
2. the company can make profit from time or escapes by this management techniques.
3. The customer can see starting and stopping and elapsed time exactly for this reason increase the reliability of customer to company.
4. The customer reaches common programs from client program.
5. The customer can arrange alarm according to his/her wants.

Terminal and server program does not reside to system register for this reason it does not slow down to connection rates of internet.

3.STRUCTURE OF THE PROGRAM

This program are separated two parts. These are names main (server) and terminal (client). They uses for reduce the timing application and solve security problems of enterprises. The operator or operators can use simply this program. Because it has user friendly menus. Operator takes all data's about the users from client part of the program. These data's are starting time, stopping time, elapsed time, date and server part of the records orders information's and calculate charge of the orders and total charges that has to pay the user. Client part of the program is installed to whole terminal computers. User takes permission from operator to reach internet connection and other applications on the terminal computer. Operators give the permission of this specific computer with using the server program.

Client program also has an alarm, clock and calendar. If the user sets alarm then program warns the user with a message. Both of programs have a Benchmark program to test the performance of the computers.

3.1. Main Program (Server)

Main (server) program has time data's of all user transactions. If gets this data's from database that is entering by the client program. Server program gives permission to terminal computers for security. It has also security for operator entry.

3.1.1.Password Verification for Enter Server Program

This verification obtains protection to attacks that are coming to outside. We can show this application with flowchart same as at below.

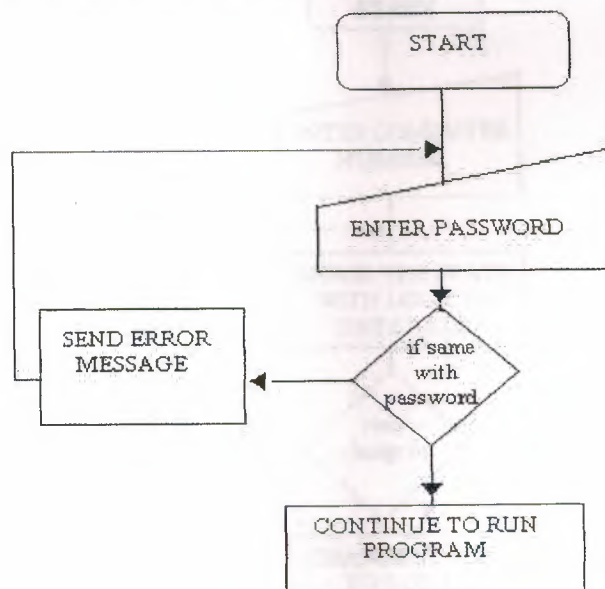


Figure3.1 The Flow of the Password verification program

We have to write some code if we want to realize this password protection. These codes same as follows:

```
Private Sub Command1_Click()  
If Text1.Text = "" Then  
Load Form3  
Form3.Show  
Unload Form2
```

```

Else
MsgBox ("Wrong keyword or accidentally caps lock open!")
Text1.Text = ""
Form2.Show
End If
End Sub

```

At this application we have to put labels, text boxes and command buttons for realize this application. All components have sub functions. It determines beginning of the object code. We can create some events with writing some codes between sub and end statements of each object.

3.1.2. Permission process of server program

This application uses for obtain protection of without permission entry of users. Main program (server) checks the database, which related with permission situation. There are two situations at permission table of database. These are 1 and 0 for each computer number. Server program can change this situation 1 to 0 or 0 to 1. If the operator change the situation client part of the program detect this change with simultaneously. Then it closes the visibility of buttons. Only change starts button visibility. If state is changed as 0 from server part of the program user can not use buttons and also internet because the program is designed to not to allow to use windows desktop and menus without pressing start button of the client program.

We can do this application same as following flow chart:

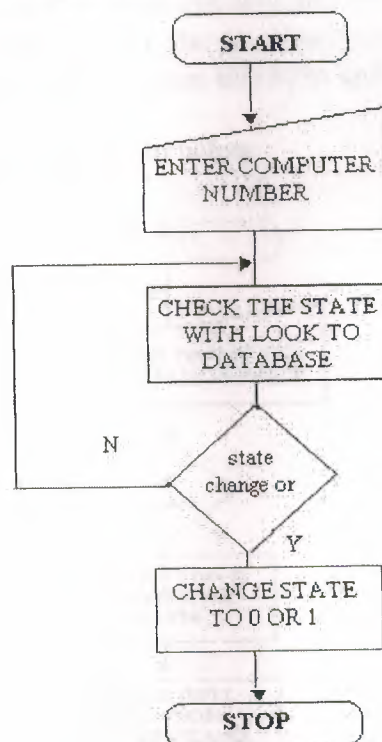


Figure3.2 The Flow of the Table Permission

The program codes of this process as follows:

```

Dim ölçüt As String
Private Sub Command1_Click()

```



```

compno$ = InputBox("Enter Computer number")
If compno = "" Then
Exit Sub
End If
ölçüt = "compno =" & compno$ & ""
Data1.Recordset.FindFirst ölçüt
End Sub

```

```

Private Sub Command2_Click()
Text1.Text = 1
Data1.UpdateRecord
End Sub

```

```

Private Sub Command3_Click()
Text1.Text = 0
Data1.UpdateRecord
End Sub

```

We use three command buttons for do this process computer number and change state of tables. This buttons provide to change to state of permission table of database.

3.1.3 Changing Unit Prices of Services Process

The server part of this software records the unit price of all services in a price table of database. It records coke, coffee, pizza etc. and also internet unit time price database.

Server program provide security in order to change this unit price of services. The operators have to enter password to open this form and changing prices of unit price of services.

The flowchart of this application given below:

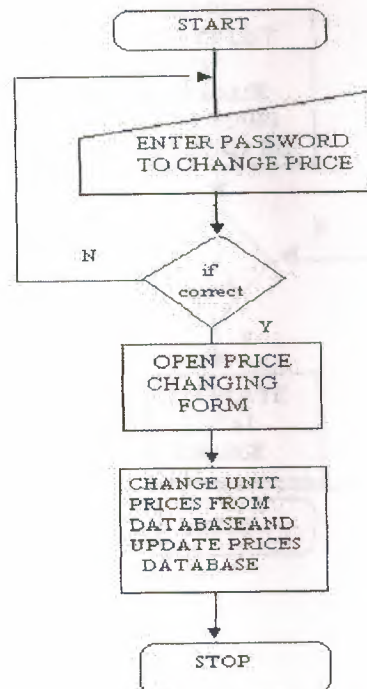


Figure 3.3. The Flow of the Changing Prices process

This part of the project does not require code because the text boxes linked with prices table of database. When the operators change the prices in a text box then close the form. This new data's records to database automatically with this way. Text boxes are linked with record sources data objects. Also data objects are linked to specify prices table of database.

3.1.4. Recording Orders and Calculate Charges Process

Customer can realize the taking orders of users process with use this process. This process resides in server part of the program. This form is connected to orders table of database. Customer can take orders of users with use this form. One table for the store amount of orders and another one in same database resides the unit prices of each orders. The users orders have been recorded according to their computer number. Charges of services are calculated by multiplying unit prices of order by amount of orders.

The flowchart of this application given below:

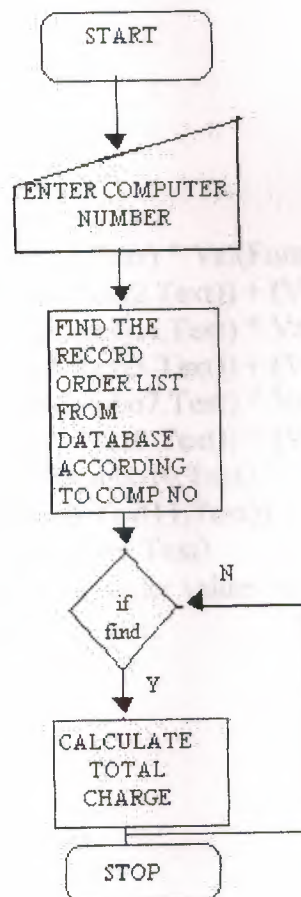


Figure.3.4. The Flow of Recording Orders and Calculate Charges process

The program codes of this process as follows:

```
Private Sub Command1_Click()  
computerno$ = InputBox(" Please Enter computer no! : ")  
If computerno = "" Then  
Exit Sub  
End If  
ölçüt = "computerno=" & computerno$ & ""  
Data1.Recordset.FindFirst ölçüt  
Data2.Recordset.FindFirst ölçüt  
End Sub
```

```
Private Sub Command2_Click()  
Combo1.Text = "0"  
Combo2.Text = "0"  
Combo3.Text = "0"  
Combo4.Text = "0"  
Combo5.Text = "0"  
Combo6.Text = "0"  
Combo7.Text = "0"  
Combo8.Text = "0"  
Combo9.Text = "0"  
Combo10.Text = "0"  
Combo11.Text = "0"  
End Sub
```

```
Private Sub Command3_Click()  
Form4.Text6.Text = (Val(Combo1.Text) * Val(Form6.Text1.Text)) +  
(Val(Combo2.Text) * Val(Form6.Text2.Text)) + (Val(Combo3.Text) *  
Val(Form6.Text3.Text)) + (Val(Combo4.Text) * Val(Form6.Text4.Text)) +  
(Val(Combo5.Text) * Val(Form6.Text5.Text)) + (Val(Combo6.Text) *  
Val(Form6.Text6.Text)) + (Val(Combo7.Text) * Val(Form6.Text7.Text)) +  
(Val(Combo8.Text) * Val(Form6.Text8.Text)) + (Val(Combo9.Text) *  
Val(Form6.Text9.Text)) + (Val(Combo10.Text) * Val(Form6.Text10.Text)) +  
(Val(Combo11.Text) * Val(Form6.Text11.Text))  
Form4.Text7.Text = Val(Form4.Text6.Text)  
'at below statements makes transfer order values to report database  
Text3.Text = Combo1.Text  
Text4.Text = Combo2.Text  
Text5.Text = Combo3.Text  
Text6.Text = Combo4.Text  
Text7.Text = Combo5.Text  
Text8.Text = Combo6.Text  
Text9.Text = Combo7.Text  
Text10.Text = Combo8.Text  
Text11.Text = Combo9.Text  
Text12.Text = Combo10.Text  
Text13.Text = Combo11.Text  
End Sub
```

We have to put 3 buttons to realize this process on form. This buttons resides at forms form calculates orders of charges, finding of computer number, clearing for new entry.

Multiplying values that taking prices from database and amount of services with use combo boxes over a form performs calculations.

3.1.5. Calculation of Total Charges of Users Process

Operator calculates all charge of certain users by using server part of program in this process. First operator has to find certain computer with search in the database. After find this computer number. Operator calculates orders of charges. This value is calculated with multiplying by unit price of the services by the amount of ordered services. When we press calculate total charges button program calculates internet charge of user and add this value to orders charge and finds the total charges. User has to press stop button at the client program for find the internet charge of user. When user press stop button client program calculates elapsed time and recorded to time table of the database. Server uses this record by connected to database. While calculate internet charge of the user. Server program multiply by unit price of internet connection by elapsed time for calculate internet charge. While server calculates orders of charge it multiply by amount of orders values by unit prices of orders.

Also operator can detect escapes from the charges form owing linking database. Because when user press start and stop button all data's about the program records time and computers numbers of users.

The program codes of this process as follows:

```
Private Sub Command1_Click()  
computerno$ = InputBox("Enter number of computer that you want to find? ")  
If computerno = "" Then  
Exit Sub  
End If  
ölçüt = "computerno=" & computerno$ & ""  
Data1.Recordset.FindFirst ölçüt  
End Sub
```

```
Private Sub Command3_Click()  
Data1.Recordset.Delete  
Data1.Recordset.MovePrevious  
Text5.Text = ""  
Text6.Text = ""
```

```
Private Sub Command5_Click()  
Text5.Text = Val(Text4.Text) * Val(Form6.Text12.Text)  
Text7.Text = Val(Text5.Text) + Val(Text6.Text)  
Data1.UpdateRecord  
End Sub
```


The flow of Calculation of Total Charges of User process at below:

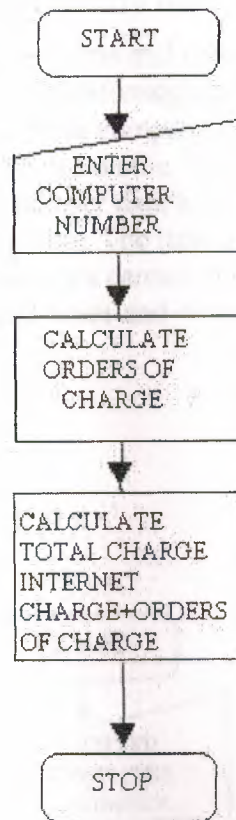


Figure.3.5. The Flow of Calculation of Total Charges process

3.1.6.Reporting Process of server part of program

Orders Report:

This process provides to see of all orders of computers. We can print the whole data's in the database or export like a file or web page. When the operator calculates order charges automatically this data's are transferred to orders table of report database by the server part of the program.

Time and Charge Report:

This report shows times and charges data's with linking time tables of database. Also we can print and export this report. This report calculate total amount of charges of all users and amount of average charge for all users. If operator delete the time data's from time and charge form at server part of the program. This data's are not erased from the reports because when the operator presses calculate total charges button data's are automatically transferred to reports database. For this reason the operator cannot delete this data's in no way. This property achieve protection of losses arise from operators.

Unit Prices of Services Report:

This reports shows unit prices of all services. We can print and export this data's. This report is connected to pries table of database.

3.1.7. Transferring Time and Orders Data's of users to Report Database Process

Server part of the program transfers time and orders data's of user, when operator pressed to the total charge button. Transferring operation is needed because operator deletes user data's after user leave from company to receive new user to this computer. Therefore this time and order data's is volatile. We cannot see user data's at the past. For this reason program transfers all user data's to report database, when the operator pressed to calculate total charge button. The time and order data's of users are permanent at report database. Operators cannot delete this user data's.

We can see the flow of transferred times and orders data's to time and charge report process:

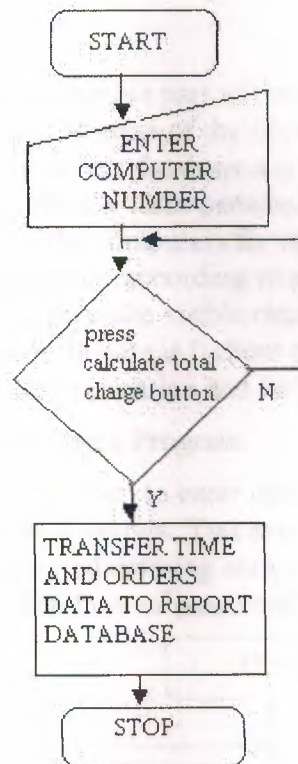


Figure.3.6. The Flow of Transferring Time and Orders data's to Report Database

The program codes of this process as follows:

```
Private Sub Command3_Click()  
Text3.Text = Combo1.Text  
Text4.Text = Combo2.Text  
Text5.Text = Combo3.Text  
Text6.Text = Combo4.Text  
Text7.Text = Combo5.Text
```



```

Text8.Text = Combo6.Text
Text9.Text = Combo7.Text
Text10.Text = Combo8.Text
Text11.Text = Combo9.Text
Text12.Text = Combo10.Text
Text13.Text = Combo11.Text
End Sub

```

```

Private Sub Command5_Click()
Data2.Recordset.AddNew
Text9.Text = Text1.Text
Text10.Text = Text2.Text
Text11.Text = Text3.Text
Text12.Text = Text4.Text
Text8.Text = Text7.Text
Text14.Text = Text13.Text
Data2.UpdateRecord
End Sub

```

3.2.Client Program

The client program is most considerable part of the internet reservation project because user use this part of the program. Mission of the client program is sent time information's of users to certain secure database and control the permission table with simultaneously. It controls periodically table permission states because client program has to detect permission state of the computers for make visible or invisible start button and other permission-required buttons according to permission state of computer. If permission state is 1 client program make visible required buttons and gives to permission to start timer of users. If state is 0 client program make invisible to start button and user cannot reach any application and he/she cannot start timer.

3.2.1. Password Verification for Client Program

The user has to enter a valid password to enter client program. This process is to defend program from unauthorized entries. You can see password verification process in figure.3.7. Operators also required entering computer numbers for terminals at startup for sending it to the database. The flow of password verification at below:

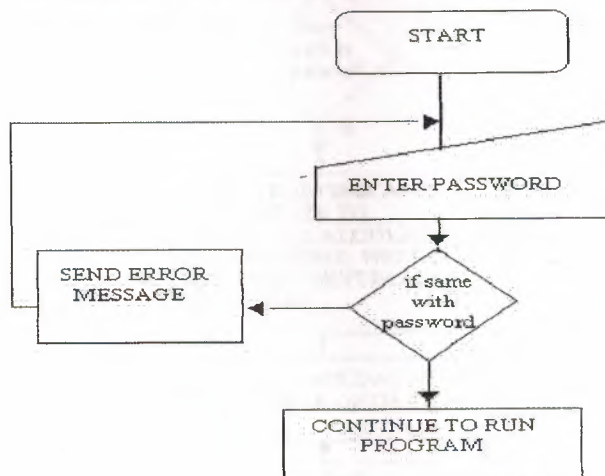


Figure.3.7.The Flow of Password Verification process

The Program codes this application as follow:

```
Private Sub Command1_Click()
```

```
If Text1.Text = "" Then
```

```
Load Form3
```

```
Form3.Show
```

```
Unload Form2
```

```
Else
```

```
MsgBox ("Wrong keyword or accidentally caps lock open!")
```

```
Text1.Text = ""
```

```
Form2.Show
```

```
End If
```

```
End Sub
```

At above codes check in the text box if it is true format it loads main form. If it is not true format it sends error message.

3.2.2.Determining and Recording Starting and Stopping Time of User Process

This process determines starting time of using services, which are taking from system. This event is realized by start button. After this event if user press stop button client part of the program takes time again from system this called stopping time. Elapsed time is calculated that are subtracting starting time from stopping time. Client program records this data's and also dates to time table of database. Recording event is connected stop button. A timer is used for rounded values for changes, the timer counts in each second by increasing a variable. This is more useful way for estimate internet price in a exactly. The flow of this process at below:

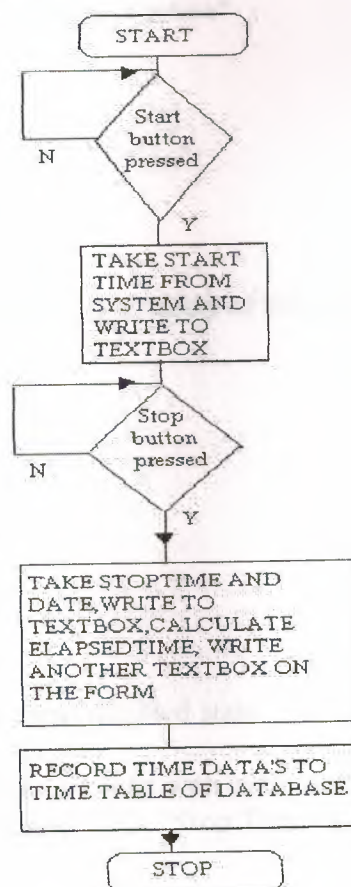


Figure.3.8.The Flow of Determining and Recording Starting and Stopping Time process

The codes below for realize this application:

```
Private Sub Command1_Click()  
'for below statement unlocks the enablety of this programs'  
drwatson.Enabled = True  
dialer.Enabled = True  
media.Enabled = True  
note.Enabled = True  
paint.Enabled = True  
player.Enabled = True  
scan.Enabled = True  
volume.Enabled = True  
'-----'  
Timer1.Enabled = True  
Command5.Enabled = True  
Command1.Visible = False  
Command4.Enabled = True  
timer.Enabled = False  
tx.Enabled = True  
st.Enabled = True  
Command2.Enabled = True  
'these satatements are worked to timer at above  
starttime = Now  
Text1.Text = Format(starttime, "hh:mm:ss")  
StatusBar1.Panels(1).Text = "Timer is active"  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""
```

End Sub

```
Private Sub Command2_Click()  
'for below statement locks again the enablety of this programs'  
drwatson.Enabled = False  
dialer.Enabled = False  
media.Enabled = False  
note.Enabled = False  
paint.Enabled = False  
player.Enabled = False  
scan.Enabled = False  
volume.Enabled = False  
'-----'  
Dim selection As Variant  
'when we press stop button turns disabled state  
'start button turns enabled  
selection = MsgBox("You cannot start with same account again if you stop now! Are  
you sure want to stop and leave?", 52, "Stop Timer")  
If selection = 6 Then  
Command4.Enabled = False
```

```
tx.Enabled = False
```

```
Command2.Enabled = False
```

```
Command1.Visible = True
```

```
Command5.Enabled = False
```

```
EndTime = Now
```

```
dat = Date
```

```
Text5.Text = dat
```

```
ElapsedTime = EndTime - starttime
```

```
Text2.Text = Format(EndTime, "hh:mm:ss")
```

```
Text3.Text = Format(ElapsedTime, "hh:mm:ss")
```

```
Timer1.Enabled = False
```

```
Text4.Text = i
```

```
i = 0
```

```
Form6.Data1.Recordset.AddNew
```

```
Form6.Text4.Text = Text4.Text
```

```
Form6.Text2.Text = Text1.Text
```

```
Form6.Text1.Text = Y
```

```
Form6.Text3.Text = Text2.Text
```

```
Form6.Text7.Text = Text5.Text
```

```
timer.Enabled = True
```

```
st.Enabled = False
```

```
End If
```

```
StatusBar1.Panels(1).Text = "Timer is not active"
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
i = i + 1
```

```
End Sub
```

Here the start time reflected to the user in a text box that taken from system and timer activated for counting seconds, which will be used to calculate Internet charge.

3.2.3. Visibility Control of Start button Process

This process protects the enterprise from non-authorized users. Client part of the program makes control to database for state of the permission. This detection process realized by timer. Timer events are arranged to check periodically the state of database, when the state of the permission 0, timer recognize this situation and makes invisible to start button. Therefore user cannot start to timer and cannot use any application on desktop. If state is 1 timer detect this situation and makes visible to start button automatically.

The Flow of this application as follow:

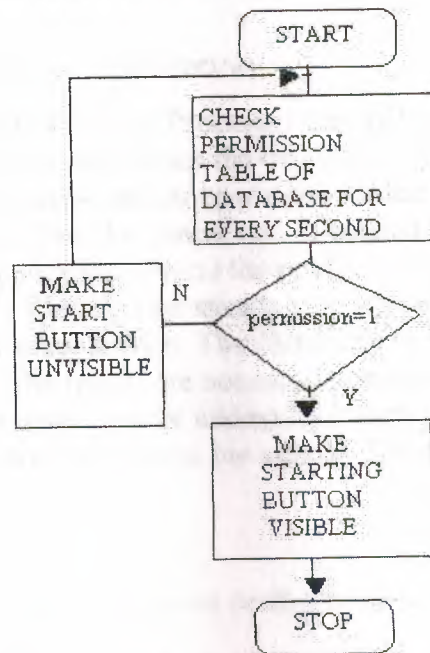


Figure.3.9. The Flow of Visibility Control of Start Button process

The program codes for this process at below:

```

Private Sub Timer2_Timer()
Form6.Data2.UpdateRecord
Form6.Data1.UpdateRecord
If Form6.Text6.Text = 0 Then
Command1.Enabled = False
Command2.Enabled = False
timer.Enabled = False
st.Enabled = False
Command4.Enabled = False
Else
Command1.Enabled = True
End If
End Sub
  
```

```

Private Sub Form_Load()
Form6.Refresh
End Sub
  
```

There is invisible form for check the database fields by the timer. Timer checks for visibility control this permission state of this table in every second.

3.2.4. Sending Data to Report Database Process

This process sends recorded of all data's to report database. This means all data's copied to report database time's data's (computer no, starting time, stopping time, elapsed time and date of the process). This property defends to enterprises from operators. Therefore we can see all time information's about all computers at report.

4.ADDITIONAL APPLICATIONS

4.1.CPU Performance Test Application

The both programs include a Central Processor Unit (CPU) performance benchmarking application that determines the calculation speed of your CPU. The application performs some mathematical operations for the floating-point and integer types. The times are taken from the system and calculated for each operation. At the end of calculations, the application reflects the results to the charts (activex). The charts are includes different CPU's at different speeds to compare with yours. Also these CPU's are tested with same application. The charts can be two dimensioned (2D) or three dimensioned (3D). The results are not certain because application runs under the program. Also the programmer must be adding Microsoft Chart Control (MSCHRT20.OCX) to his/her project to use charts. The structure is in Figure 4.1 and Figure 4.2.

The Flow of the integer and floating-point performance test of CPU application follow:

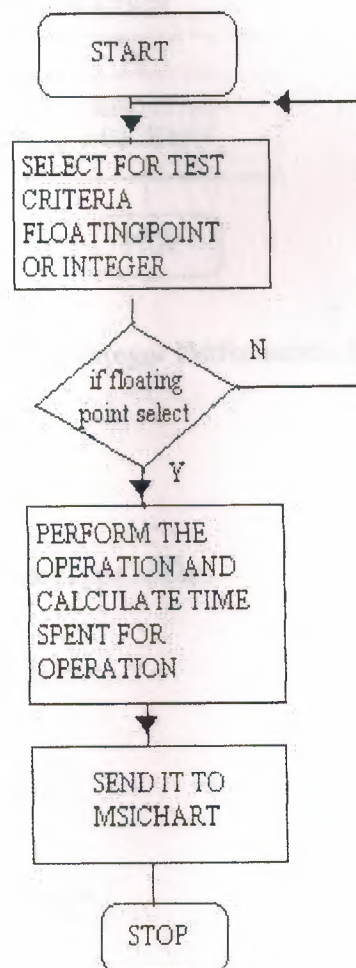


Figure.4.1. The Flow of the Floating-Point Performance Benchmark application

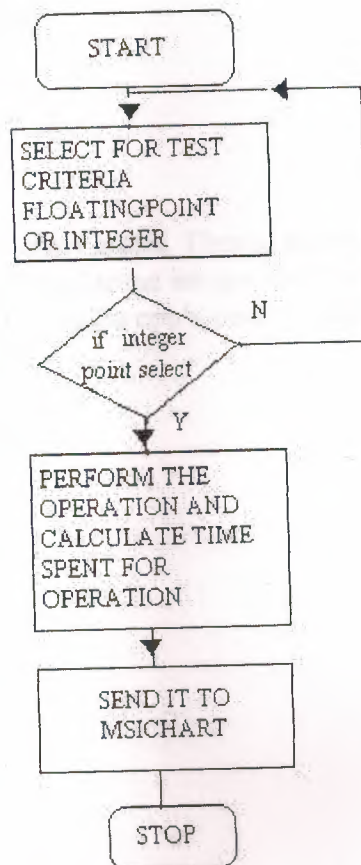


Figure.4.2. The Flow of the Integer Performance Benchmark application

The program codes are at below:

```

Private Sub Form_Load()
MSChart1.RowCount = 3
MSChart1.ColumnCount = 1
MSChart1.chartType = VtChChartType2dBar
MSChart1.Row = 1
MSChart1.Data = 228
MSChart1.RowLabel = "P2 233"
MSChart1.Row = 2
MSChart1.Data = 118
MSChart1.RowLabel = "686-120"
MSChart1.Row = 3
MSChart1.Data = 0
MSChart1.RowLabel = "Your CPU"
  
```

```

MSChart2.RowCount = 3
MSChart2.ColumnCount = 1
MSChart2.chartType = VtChChartType2dBar
MSChart2.Row = 1
  
```

```

MSChart2.Data = 27
MSChart2.RowLabel = "P2 233"
MSChart2.Row = 2
MSChart2.Data = 5.85
MSChart2.RowLabel = "686-120"
MSChart2.Row = 3
MSChart2.Data = 0
MSChart2.RowLabel = "Your CPU"
End Sub

```

Above codes provide to draw chart graph. Then at below codes perform the operation calculates the performance with send some integer and floating point numbers with certain time period to cpu, and compare cpu's performance with recorded criteria. After that sends this result to msichart.

```

Private Sub Command1_Click()
Dim i As Long, X As Long, t
t = Timer
Screen.MousePointer = 12
For i = 0 To 10000000
X = X + i + 10
X = 2 * i - 100 - X
Next
Screen.MousePointer = 0
t = Timer - t
MSChart1.Row = 3
MSChart1.Data = 100 / t
End Sub

```

```

Private Sub Command2_Click()
Dim i As Long, X As Long, t
t = Timer
Screen.MousePointer = 12
For i = 1 To 1000000
X = Log(i) * Sin(i) * Exp(Cos(i))
X = X / i
X = X * i
Next
Screen.MousePointer = 0
t = Timer - t
MSChart2.Row = 3
MSChart2.Data = 100 / t
End Sub

```

```

Private Sub Command3_Click()
Unload Me
End Sub

```


4.2.Alarm

The user can adjust the alarm to certain times 1 hour, 2 hour, 3 hour. This property provides to put limits to user him/herself. At the end of each alarm warns users with message.

To realize this application all codes are below:

```
Private Sub Command1_Click()  
Timer2.Enabled = False  
Timer1.Enabled = False  
Timer3.Enabled = False  
End Sub
```

```
Private Sub Form_Load()  
Dim k As Integer  
Timer2.Enabled = False  
Timer1.Enabled = False  
Timer3.Enabled = False  
End Sub
```

```
Private Sub Option1_Click()  
If Option1.Value = True Then  
Timer1.Enabled = True  
Timer2.Enabled = False  
Timer3.Enabled = False  
End If  
End Sub
```

```
Private Sub Option2_Click()  
If Option2.Value = True Then  
Timer2.Enabled = True  
Timer1.Enabled = False  
Timer3.Enabled = False  
End If  
End Sub
```

```
Private Sub Option3_Click()  
If Option3.Value = True Then  
Timer3.Enabled = True  
Timer1.Enabled = False  
Timer2.Enabled = False  
End If  
End Sub
```

```
Private Sub Timer1_Timer()  
k = k + 1  
If k = 180 Then  
MsgBox ("Time is up")  
End Sub
```

```

Private Sub Timer2_Timer()
k = k + 1
If k = 60 Then
MsgBox ("Time is up")
End If
End Sub
Private Sub Timer3_Timer()
k = k + 1
If k = 120 Then
MsgBox ("Time is up")
End If
End Sub

```

The Flowchart of the alarm application is at below:

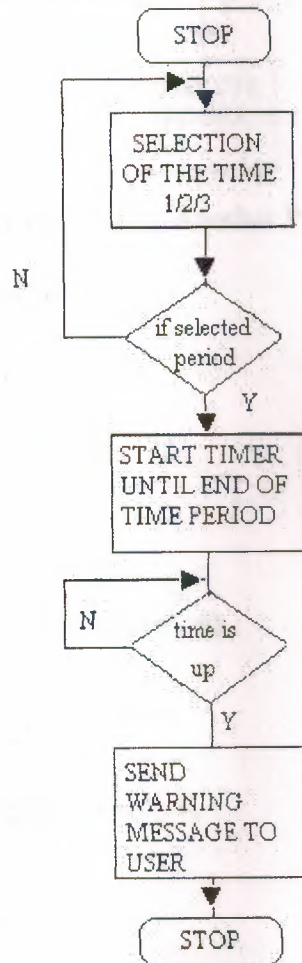


Figure.4.3. The Flow of the Alarm application

4.3. Using another program on Client Program Process:

Client program support some basic programs (mediaplayer, notepad, paintbrush, Drwatson, dialer, cdplayer, scandisk, volume control, internet explorer) to user can execute this program without leave the client program. This application provides to useful and basic programs to user and he/she can reach these programs easily and rapidly. Shell command executes other programs from over client program.

The Flow of the Using Another Program On Client Program Application is below:

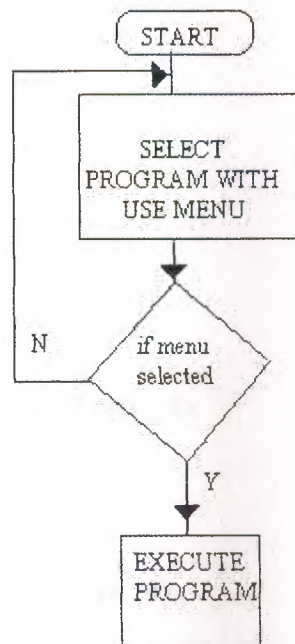


Figure.4.4. The Flow of the Using Another Program On Client Program application

The program codes are at below:

```
Private Sub dialer_Click()
```

```
Dim p
```

```
p = Shell("c:\windows\Dialer.exe", 4)
```

```
End Sub
```

```
Private Sub drwatson_Click()
```

```
Dim k
```

```
k = Shell("c:\windows\Drwatson.exe", 4)
```

```
End Sub
```

```
Private Sub media_Click()
```

```
Dim t
```

```
t = Shell("c:\windows\Mplayer.exe", 4)
```

```
End Sub
```

```
Private Sub note_Click()
```

```
Dim s
```

```
s = Shell("c:\windows\notepad.exe", 4)
```

```
End Sub
```

```
Private Sub paint_Click()
```

```
Dim l
```

```
l = Shell("c:\windows\pbrush.exe", 4)
```

```
End Sub
```

```
Private Sub player_Click()
Dim z
z = Shell("c:\windows\cdplayer.exe", 4)
End Sub
```

```
Private Sub scan_Click()
Dim w
w = Shell("c:\windows\scandiskw.exe", 4)
End Sub
```

```
Private Sub volume_Click()
Dim g
g = Shell("c:\windows\sndvol32.exe", 4)
End Sub
```

```
Private Sub int Cllick()
Dim n
n=Shell("c:\windows\explorer.exe",4)
End Sub
```

4.4.Clock Application

Client program has a clock for use a user. This clock has classical and digital shape it has a very user-friendly shapes.

The program codes of the clock application follow:

```
Private Sub Form_Load()
Width = ScaleHeight
Scale (-20, 20)-(20, -20)
Timer1.Interval = 1000 'saniye
AutoRedraw = True
Circle (0, 0), 20
DrawMode = 7 'xor
End Sub
```

```
Private Sub Timer1_Timer()
Dim aci, saniye, dakika, saat
Static sx, sy, dx, dy, stx, sty
Caption = Time
DrawWidth = 1
Line (0, 0)-(sx, sy), QBColor(10) 'oncekinin uzerine cizerek sil
saniye = Second(Time) 'saniyeyi saatten al
aci = saniye * 6 + 90 'her bir saniye 6 derecedir. 360 derece/60 saniye
sx = 20 * Cos(aci * 3.1415 / 180)
sy = 20 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(sx, sy), QBColor(10) 'saniye ciz
DrawWidth = 3
Line (0, 0)-(-dx, -dy), QBColor(11) 'oncekinin uzerine cizerek sil
dakika = Minute(Time)
aci = dakika * 6 + 90 'her dakika 6 derecedir.360 derece/60 dakika
dx = 20 * Cos(aci * 3.1415 / 180)
```



```

dy = 20 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(-dx, -dy), QBColor(11) 'yelkovanı cizer
DrawWidth = 3
Line (0, 0)-(-stx, -sty), QBColor(12) 'oncekinin uzerine czerek sil
saat = Hour(Time)
aci = saat * 30 + 90 'her saat 30 derece
stx = 12 * Cos(aci * 3.1415 / 180)
sty = 12 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(-stx, -sty), QBColor(12) 'akrebi ciz

```

End Sub

4.5. Calender Application

Very simply the MSCAL.OCX file creates these applications. When we add this ocx file to our project. We can reach easily very useful calendar. User can use calendar without need any permission. We do not need to write program code for create this application.

4.6. Minimization of Client Program Process

User can minimize and send to task bar to client program after press start button therefore he/she can access the all application at the terminal computer.

Flow of minimization of client program is at below:

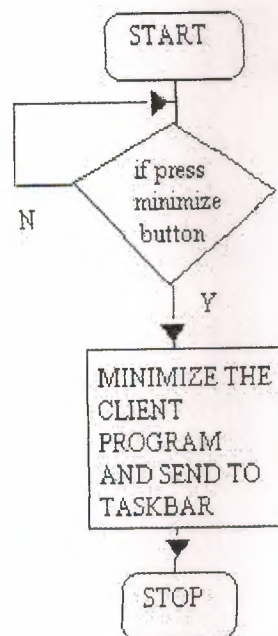


Figure.4.5. The Flow of the Minimization of Client Program application

The program codes for realized this application are below:

```
Private Sub Command4_Click()  
Dim minimized As VbAppWinStyle  
Form3.WindowState = 1 - minimized  
End Sub
```


5. USED API'S AND COMPONENTS

API (Advanced Programmers Interface) is a set of predefined Windows functions used to control the appearance and behavior of every Windows element (from the outlook of the desktop window to the allocation of memory for a new process). Every user action causes the execution of several or more API function telling Windows what's happened. It is something like the native code of Windows. Other languages just act as a shell to provide an automated and easier way to access APIs. Visual Basic has done a lot in this direction. It has completely hidden the APIs and provided a quite different approach for programming under Windows. So it includes many functions to organize that controls. Visual Basic can use that controls globally inside the project by declaring it in a module as a function or a subprogram. Programmer must load an API once from its located DLL or EXE extended file to use it in his/her project. The API functions reside in DLLs (like User32.dll, GDI32.dll, Shell32.dll...) in the Windows system directory.

5.1. Notifying Program Icon In System Tray

Windows 9/x versions supports "System Tray". The window at the right corner of the progress bar is the System Tray. Programmer is able to add his/her program's icon to the System Tray by using Shell_NotifyIcon API. Programmer can define some properties that belong application. Application is defined with NOTIFYICONDATA type that includes different parameters.

cbSize	Variable length
hWnd	HWND number of the form that will be settled to Sytem Tray
uCallbackMessage	The event that should be called when user makes a process on the icon
hIcon	Defines the icon which will be settled in System Tray
szTip	ToolTipText

Table 5.1. The parameters of type NOTIFYICONDATA

The program codes for adding icon to the System Tray are:

```
Private Declare Function Shell_NotifyIcon Lib "shell32.dll" Alias "Shell_NotifyIconA"  
(ByVal dwMessage As Long, lpData As NOTIFYICONDATA) As Long  
Private Type NOTIFYICONDATA  
    cbSize As Long  
    hWnd As Long  
    uID As Long  
    uFlags As Long  
    uCallbackMessage As Long  
    hIcon As Long  
    szTip As String * 64  
End Type  
Dim tray As NOTIFYICONDATA
```

```

Private Sub Form_Load()
Text2.Text = ""
tray.cbSize = Len(tray)
tray.hwnd = Form3.hwnd
tray.szTip = "Net@Cafe" + Chr(0)
tray.uCallbackMessage = &H200
tray.uFlags = 7
tray.hIcon = Form3.Icon
Shell_NotifyIcon 0, tray
End Sub

```

5.2. Hiding Program From Windows Task List

These functions are used for security on the client program. User's can close the program by using Task List (Ctrl+Alt+Del) so it can be distrustful. By using some API's the programmers can hide their program in Task List. To do this, the programmer must add below code to the active form's load and unload events after declared API function.

Declarations are:

```

Private Declare Function GetCurrentProcessId Lib "kernel32" () As Long
Private Declare Function GetCurrentProcess Lib "kernel32" () As Long
Private Declare Function RegisterServiceProcess Lib "kernel32" (ByVal dwProcessID
As Long, ByVal dwType As Long) As Long
Private Const RSP_SIMPLE_SERVICE = 1
Private Const RSP_UNREGISTER_SERVICE = 0
Codes are:

```

```

Private Sub Form_Load()
Dim id As Long
Dim X As Long
id = GetCurrentProcessId()
X = RegisterServiceProcess(id, RSP_SIMPLE_SERVICE)
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
Dim id As Long
Dim X As Long
id = GetCurrentProcessId()
X = RegisterServiceProcess(pid, RSP_UNREGISTER_SERVICE)
End Sub

```


CONCLUSION

Visual Basic is known for its ability to allow a programmer to quickly, and easily, create a Windows program. In part, this is possible because objects provide code that you don't have to write.

In Visual Basic, the basic building block of an application is a form, which is simply a window. The Visual Basic IDE can insert forms into your project, and then you can resize the forms as well as change other properties of the form. However, controls (checkboxes, textboxes...) are also windows. A form is distinguished from a control in that only forms can exist as standalone objects. When controls are used, they must be placed in a form and there are a few exceptions such as the printer object or the screen object, which are not considered part of any form, but are part of a Visual Basic program.

It's worth noting that when managers talk about programmers, one of the common metrics used to describe performance is "lines of code per month". There are all kinds of debate about how good a metric this is, but the fact is that the metric is used. It's not that you don't get credit for novel algorithms, or that you won't be a hero to fellow programmers when they see how you solved a problem with 10 lines of code that took them 100. You'll get that credit (and mental satisfaction) but looking at the big picture it's clear that the volume of code you can crank out will be the visible result of your efforts. Writing a program is essentially a design task, aimed at solving a problem. Like most things, there are many design approaches for every problem. Some solve the problem by grinding out an answer. Some are very elegant. Others are so complex that even the designer has a hard time keeping up with the convolutions of the approach. Here the specifications of a great Visual Basic programmer:

- Great programmers must be coding experts
- Great programmers understand that there are many approaches to a problem
- Great programmers work efficiently.

VB provides two controls, which make the link to the database file, and which creates the recordset that is exposed to the rest of the controls in your application. The two are identical in concept but differ in the flexibility they offer to the programmer. Data controls are the most important part of the programs, and applications. By adding user's information on a database file, the programmers can use this file by adding, reading and deleting fields on it.

In case you have any question about the value of objects, that objects (in the form of controls) are the single biggest reason why Visual Basic has been so successful. In Visual Basic 6.0, Microsoft has expanded the object features of Visual Basic, making it even more powerful than ever.

This Internet café reservation program contains user-friendly forms, some API's, Activex controls on it to fortify the application. It provides sensible security and usability for users and operators. The structure of program is built up on to client-server relation. Internet café's finds appropriate solution to their security or automation problems, but human relation between users and operators still important in the enterprises.

This project uses some Internet café reservation and automation techniques. It supports security and automation advantage to manage calculation of charges and reservation transactions. This program provides these possibilities with use Visual Basic functions, database controls and other many resources of Visual Basic.

REFERENCES

- [1] Douglas E. Comer, The Internet Book: Computer Networking and How to Internet Works, Prentice-Hall International, Inc., NJ, 1994
- [2] İ.Karagülle – Z. Pala, Visual Basic 6.0 Pro: Visual Basic programming for professional users, Türkmen Kitabevi, Beyazıt, İSTANBUL, 1999
- [3] Visual Basic Forum Site. Retrieved May 15, 2002 from the World Wide Web: <http://www.vbonline.com>.
- [4] Visual Basic Programmers Site. Retrieved May 22, 2002 from the World Wide Web: <http://vbi.org>.
- [5] VBNet A guide for sample applications. Retrieved May 23, 2002 from the World Wide Web: <http://www.mvps.org/vbnet/>.
- [6] Visual Basic Program Developers Suggestions. Retrieved May 27, 2002 from the World Wide Web: <http://gan.dalf.pht.com/pub/vbbbn/cgi/am.cgi>.
- [7] Microsoft Visual Basic Developers Site. Retrieved June 1, 2002 from World Wide Web: <http://www.Microsoft.com/vbasic/>.
- [8] Visual Basic Programming Tutorials Site. Retrieved June 3, 2002 From World Wide Web: <http://www.programmingtutorials.com>.
- [9] İ.Karagülle-Z.Pala, Visual Basic Pro 5.0 : Visual Basic Programming, Türkmen Kitabevi, Beyazıt, İSTANBUL, 1997

APPENDIX A

We can see all codes for both parts of the programs in visual basic language. This appendix provide to us more comprehensive look to project.

Main Program (Server)

Codes are in an ordered way according to their form numbers.

Form1(Table Permission):

Dim ölçüt As String

```
Private Sub Command1_Click()  
compno$ = InputBox("Enter Computer number")  
If compno = "" Then  
Exit Sub  
End If  
ölçüt = "compno =" & compno$ & ""  
Data1.Recordset.FindFirst ölçüt  
End Sub
```

```
Private Sub Command2_Click()  
Text1.Text = 1  
Data1.UpdateRecord  
End Sub
```

```
Private Sub Command3_Click()  
Text1.Text = 0  
Data1.UpdateRecord  
End Sub
```

```
Private Sub Form_Load()
```

```
End Sub  
Private Sub Text2_KeyPress(KeyAscii As Integer)  
If Chr(KeyAscii) = "a" Or Chr(KeyAscii) = "A" Or Chr(KeyAscii) = "a" Or  
Chr(KeyAscii) = "b" Or Chr(KeyAscii) = "B" Or Chr(KeyAscii) = "C" Or  
Chr(KeyAscii) = "c" Or Chr(KeyAscii) = "D" Or Chr(KeyAscii) = "d" Or  
Chr(KeyAscii) = "E" Or Chr(KeyAscii) = "e" Or Chr(KeyAscii) = "F" Or  
Chr(KeyAscii) = "f" Or Chr(KeyAscii) = "G" Or Chr(KeyAscii) = "g" Or  
Chr(KeyAscii) = "Ğ" Or Chr(KeyAscii) = "ğ" Or Chr(KeyAscii) = "H" Or  
Chr(KeyAscii) = "h" Or Chr(KeyAscii) = "I" Or Chr(KeyAscii) = "i" Or Chr(KeyAscii)  
= "İ" Or Chr(KeyAscii) = "i" Or Chr(KeyAscii) = "J" Or Chr(KeyAscii) = "j" Or  
Chr(KeyAscii) = "K" Or Chr(KeyAscii) = "k" Or Chr(KeyAscii) = "L" Or  
Chr(KeyAscii) = "l" Or Chr(KeyAscii) = "M" Or Chr(KeyAscii) = "m" Or  
Chr(KeyAscii) = "N" Or Chr(KeyAscii) = "n" Or Chr(KeyAscii) = "O" Or  
Chr(KeyAscii) = "o" Or Chr(KeyAscii) = "Ö" Or Chr(KeyAscii) = "ö" Or  
Chr(KeyAscii) = "P" Or Chr(KeyAscii) = "p" Or Chr(KeyAscii) = "R" Or  
Chr(KeyAscii) = "r" Or Chr(KeyAscii) = "S" Or Chr(KeyAscii) = "s" Or Chr(KeyAscii)  
= "Ş" Then
```



```

KeyAscii = 0
Beep
End If
If Chr(KeyAscii) = "ş" Or Chr(KeyAscii) = "T" Or Chr(KeyAscii) = "t" Or
Chr(KeyAscii) = "U" Or Chr(KeyAscii) = "u" Or Chr(KeyAscii) = "Ü" Or
Chr(KeyAscii) = "ü" Or Chr(KeyAscii) = "V" Or Chr(KeyAscii) = "v" Or
Chr(KeyAscii) = "Y" Or Chr(KeyAscii) = "y" Or Chr(KeyAscii) = "Z" Or
Chr(KeyAscii) = "z" Or Chr(KeyAscii) = "X" Or Chr(KeyAscii) = "x" Or
Chr(KeyAscii) = "Q" Or Chr(KeyAscii) = "q" Or Chr(KeyAscii) = "W" Or
Chr(KeyAscii) = "w" Or Chr(KeyAscii) = "Ç" Or Chr(KeyAscii) = "ç" Or
Chr(KeyAscii) = "<" Or Chr(KeyAscii) = ">" Or Chr(KeyAscii) = "." Or
Chr(KeyAscii) = ":" Or Chr(KeyAscii) = "," Or Chr(KeyAscii) = ";" Or Chr(KeyAscii)
= "-" Or Chr(KeyAscii) = "*" Or Chr(KeyAscii) = "0" Or Chr(KeyAscii) = "é" Or
Chr(KeyAscii) = "!" Or Chr(KeyAscii) = "" Or Chr(KeyAscii) = "^" Or Chr(KeyAscii)
= "+" Or Chr(KeyAscii) = "%" Or Chr(KeyAscii) = "&" Or Chr(KeyAscii) = "/" Or
Chr(KeyAscii) = "(" Or Chr(KeyAscii) = ")" Or Chr(KeyAscii) = "=" Or Chr(KeyAscii)
= "?" Or Chr(KeyAscii) = "#" Or Chr(KeyAscii) = "$" Then
KeyAscii = 0
Beep
End If
End Sub

```

```

Private Sub Text1_Change()
Dim Cancel As Variant
If Val(Text1) < 0 Or Val(Text1) > 1 Then
MsgBox ("You should enter 1 or 0 for state")
Text1.Text = "0"
Cancel = True
End If
End Sub

```

Form2 (Password verification):

```

Private Sub Command1_Click()
If Text1.Text = "" Then
Load Form3
Form3.Show
Unload Form2
Else
MsgBox ("Wrong keyword or accidentally caps lock open!")
Text1.Text = ""
Form2.Show
End If
End Sub

```

```

Private Sub Image1_Click()

```

```

End Sub

```

Private Sub Picture2_Click()

End Sub

Private Sub Form_Load()

End Sub

Form3 (Main Form):

Private Declare Function exitwindowsex Lib "user32" (ByVal uflags As Long, ByVal dwreserved As Long) As Long

Dim ölçüt As String

Private Declare Function Shell_NotifyIcon Lib "shell32.dll" Alias "Shell_NotifyIconA" (ByVal dwMessage As Long, lpData As NOTIFYICONDATA) As Long

Private Type NOTIFYICONDATA

cbSize As Long

hwnd As Long

uID As Long

uflags As Long

uCallbackMessage As Long

hIcon As Long

szTip As String * 64

End Type

Dim tray As NOTIFYICONDATA

Private Sub ab_Click(Index As Integer)

frmAbout.Show

End Sub

Private Sub AniGIF1_DblClick()

Call Shell("start mailto:ibrahimdokuzer@hotmail.com")

End Sub

Private Sub AniGIF1_Step()

End Sub

Private Sub banchmerk_Click(Index As Integer)

Form8.Show

End Sub

Private Sub closing_Click(Index As Integer)

Form1.Show

End Sub

Private Sub Command1_Click()

Form4.Show

End Sub

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    StatusBar1.Panels(4).Style = sbrText
    StatusBar1.Panels(4).AutoSize = sbrContents
    StatusBar1.Panels(4).Text = "Charges"
End Sub
```

```
Private Sub Command2_Click()
    Form10.Show
End Sub
```

```
Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    StatusBar1.Panels(4).Style = sbrText
    StatusBar1.Panels(4).AutoSize = sbrContents
    StatusBar1.Panels(4).Text = "Unit Cost of Services "
End Sub
```

```
Private Sub Command3_Click()
    Form5.Show
End Sub
```

```
Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    StatusBar1.Panels(4).Style = sbrText
    StatusBar1.Panels(4).AutoSize = sbrContents
    StatusBar1.Panels(4).Text = "Orders of Services"
End Sub
```

```
Private Sub Command4_Click()
    Form9.Show
End Sub
```

```
Private Sub Command4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    StatusBar1.Panels(4).Style = sbrText
    StatusBar1.Panels(4).AutoSize = sbrContents
    StatusBar1.Panels(4).Text = "Service List of All Tables "
End Sub
```

```
Private Sub Command5_Click()
    Form8.Show
End Sub
```

```
Private Sub Command5_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    StatusBar1.Panels(4).Style = sbrText
    StatusBar1.Panels(4).AutoSize = sbrContents
    StatusBar1.Panels(4).Text = "CPU Test "
End Sub
```



```

Private Sub Command6_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Report"
Form11.Show
End Sub

```

```

Private Sub Command7_Click()
Form7.Show
End Sub

```

```

Private Sub Command7_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Exit"
End Sub

```

```

Private Sub Command8_Click(Index As Integer)
Form1.Show
End Sub

```

```

Private Sub Command8_MouseMove(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Open/Close Permission "
End Sub

```

```

Private Sub dharges_Click(Index As Integer)
Form4.Show
End Sub

```

```

Private Sub exit_Click(Index As Integer)
Form7.Show
End Sub

```

```

Private Sub Form_Load()
shut.Enabled = False
tray.cbSize = Len(tray)
tray.hwnd = Form3.hwnd
tray.szTip = "Net@Cafe" + Chr(0)
tray.uCallbackMessage = &H200
tray.uflags = 7
tray.hIcon = Form3.Icon
Shell_NotifyIcon 0, tray
If App.PrevInstance = True Then
MsgBox ("Program is already running")

```

```
End  
End If  
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
StatusBar1.Panels(1).Text = "Server program is active"  
End Sub
```

```
Private Sub list_Click(Index As Integer)  
Form9.Show  
End Sub
```

```
Private Sub number_Click(Index As Integer)  
Form1.Show  
End Sub
```

```
Private Sub opening_Click(Index As Integer)  
Form1.Show  
End Sub
```

```
Private Sub orders_Click(Index As Integer)  
Form5.Show  
End Sub
```

```
Private Sub tech_Click(Index As Integer)  
MsgBox ("You can send E-mail to ibrahimdokuzer@hotmail.com for technical  
support")  
End Sub
```

```
Private Sub unit_Click(Index As Integer)  
Form10.Show  
End Sub
```

Form4 (Charges):

```
Private Sub Command1_Click()  
computerno$ = InputBox("Enter number of computer that you want to find? ")  
If computerno = "" Then  
Exit Sub  
End If  
ölçüt = "computerno =" & computerno$ & ""  
Data1.Recordset.FindFirst ölçüt  
End Sub
```

```
Private Sub Command2_Click()  
Form9.Show  
End Sub
```

```
Private Sub Command3_Click()  
Data1.Recordset.Delete
```

```
Data1.Recordset.MovePrevious
Text5.Text = ""
Text6.Text = ""
```

```
End Sub
```

```
Private Sub Command4_Click()
Unload Form4
Form3.Show
End Sub
```

```
Private Sub Command5_Click()
Text5.Text = Val(Text4.Text) * Val(Form6.Text12.Text)
Text7.Text = Val(Text5.Text) + Val(Text6.Text)
Data1.UpdateRecord
Data2.Recordset.AddNew
Text9.Text = Text1.Text
Text10.Text = Text2.Text
Text11.Text = Text3.Text
Text12.Text = Text4.Text
Text8.Text = Text7.Text
Text14.Text = Text13.Text
Data2.UpdateRecord
End Sub
```

```
Private Sub Command6_Click()
Form5.Show
End Sub
```

```
Private Sub Form_Load()
End Sub
```

Form5 (Orders):

```
Private Sub Command1_Click()
computerno$ = InputBox(" Please Enter computer no! : ")
If computerno = "" Then
Exit Sub
End If
ölçüt = "computerno=" & computerno$ & ""
Data1.Recordset.FindFirst ölçüt
Data2.Recordset.FindFirst ölçüt
End Sub
```

```
Private Sub Command2_Click()
Combo1.Text = "0"
Combo2.Text = "0"
Combo3.Text = "0"
Combo4.Text = "0"
Combo5.Text = "0"
Combo6.Text = "0"
```




```
Combo7.Text = "0"  
Combo8.Text = "0"  
Combo9.Text = "0"  
Combo10.Text = "0"  
Combo11.Text = "0"  
End Sub
```

```
Private Sub Command3_Click()  
Form4.Text6.Text = (Val(Combo1.Text) * Val(Form6.Text1.Text)) +  
(Val(Combo2.Text) * Val(Form6.Text2.Text)) + (Val(Combo3.Text) *  
Val(Form6.Text3.Text)) + (Val(Combo4.Text) * Val(Form6.Text4.Text)) +  
(Val(Combo5.Text) * Val(Form6.Text5.Text)) + (Val(Combo6.Text) *  
Val(Form6.Text6.Text)) + (Val(Combo7.Text) * Val(Form6.Text7.Text)) +  
(Val(Combo8.Text) * Val(Form6.Text8.Text)) + (Val(Combo9.Text) *  
Val(Form6.Text9.Text)) + (Val(Combo10.Text) * Val(Form6.Text10.Text)) +  
(Val(Combo11.Text) * Val(Form6.Text11.Text))  
Form4.Text7.Text = Val(Form4.Text6.Text)  
Text3.Text = Combo1.Text  
Text4.Text = Combo2.Text  
Text5.Text = Combo3.Text  
Text6.Text = Combo4.Text  
Text7.Text = Combo5.Text  
Text8.Text = Combo6.Text  
Text9.Text = Combo7.Text  
Text10.Text = Combo8.Text  
Text11.Text = Combo9.Text  
Text12.Text = Combo10.Text  
Text13.Text = Combo11.Text  
End Sub
```

```
Private Sub Form_Load()  
Combo1.AddItem "0"  
Combo1.AddItem "1"  
Combo1.AddItem "2"  
Combo1.AddItem "3"  
Combo2.AddItem "0"  
Combo2.AddItem "1"  
Combo2.AddItem "2"  
Combo2.AddItem "3"  
Combo3.AddItem "0"  
Combo3.AddItem "1"  
Combo3.AddItem "2"  
Combo3.AddItem "3"  
Combo4.AddItem "1"  
Combo4.AddItem "2"  
Combo4.AddItem "3"  
Combo5.AddItem "1"  
Combo5.AddItem "2"  
Combo5.AddItem "3"  
Combo6.AddItem "1"
```

```

Combo6.AddItem "2"
Combo6.AddItem "3"
Combo7.AddItem "1"
Combo7.AddItem "2"
Combo7.AddItem "3"
Combo8.AddItem "1"
Combo8.AddItem "2"
Combo8.AddItem "3"
Combo9.AddItem "1"
Combo9.AddItem "2"
Combo9.AddItem "3"
Combo10.AddItem "1"
Combo10.AddItem "2"
Combo10.AddItem "3"
Combo11.AddItem "1"
Combo11.AddItem "2"
Combo11.AddItem "3"
End Sub

```

Form6 (Cost of Orders):

```

Private Sub Command1_Click()
Unload Form6
Form3.Show
End Sub

```

```

Private Sub Command2_Click()
Unload Form6
End Sub

```

```

Private Sub Form_Load()
End Sub

```

Form7 (Exit):

```

Private Sub Command1_Click()
End
End Sub

```

```

Private Sub Command2_Click()
Unload Me
End Sub

```

```

Private Sub Label1_Click()
End Sub

```

Form8 (CPU Tests):

```

Private Sub Command1_Click()
Dim i As Long, X As Long, t
t = Timer
Screen.MousePointer = 12
For i = 1 To 1000000

```

```

X = Log(i) * Sin(i) * Exp(Cos(i))
X = X / i
X = X * i
Next
Screen.MousePointer = 0
t = Timer - t
MSChart2.Row = 3
MSChart2.Data = 100 / t
End Sub

```

```

Private Sub Command2_Click()
Dim i As Long, X As Long, t
t = Timer
Screen.MousePointer = 12
For i = 0 To 10000000
X = X + i + 10
X = 2 * i - 100 - X
Next
Screen.MousePointer = 0
t = Timer - t
MSChart1.Row = 3
MSChart1.Data = 100 / t
End Sub

```

```

Private Sub Command3_Click()
Unload Me
End Sub

```

```

Private Sub Form_Load()
MSChart1.RowCount = 3
MSChart1.ColumnCount = 1
MSChart1.chartType = VtChChartType2dBar
MSChart1.Row = 1
MSChart1.Data = 228
MSChart1.RowLabel = "P2 233"
MSChart1.Row = 2
MSChart1.Data = 118
MSChart1.RowLabel = "686-120"
MSChart1.Row = 3
MSChart1.Data = 0
MSChart1.RowLabel = "Your CPU"
MSChart2.RowCount = 3
MSChart2.ColumnCount = 1
MSChart2.chartType = VtChChartType2dBar
MSChart2.Row = 1
MSChart2.Data = 27
MSChart2.RowLabel = "P2 233"
MSChart2.Row = 2
MSChart2.Data = 5.85
MSChart2.RowLabel = "686-120"

```



```
MSChart2.Row = 3
MSChart2.Data = 0
MSChart2.RowLabel = "Your CPU"
```

```
End Sub
```

Form9 (Services List):

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

```
Private Sub Form_Load()
End Sub
```

Form10 (Password for Chancing Prices of Orders):

```
Private Sub Command1_Click()
If Text1.Text = "" Then
Load Form6
Form6.Show
Unload Form10
Else
MsgBox ("Wrong keyword or accidentally caps lock open!")
Text1.Text = ""
Form10.Show
End If
End Sub
```

```
Private Sub Form_Load()
End Sub
```

Form11 (Reports):

```
Private Sub Command1_Click()
DataReport2.Show
End Sub
```

```
Private Sub Command2_Click()
DataReport1.Show
End Sub
```

```
Private Sub Command3_Click()
DataReport3.Show
End Sub
```

```
Private Sub Form_Load()
End Sub
```

FrmAbout Form:

```
Option Explicit
' Reg Key Security Options...
Const READ_CONTROL = &H20000
```

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

INTERNET CAFÉ RESERVATION PROGRAM

**Graduation Project
COM- 400**

Student: Ibrahim Dokuzer (980869)

Supervisor: Miss. Besime Erin

Nicosia - 2002

ACKNOWLEDGMENT

First I am glad to complete my project, which I had given with blessing of God and also I am grateful to my brother and my family, for their encouragement during the preparation of this project.

I would like to thank my supervisor, Miss Besime Erin, invaluable advice and belief in my work and myself over the course of this Graduation.

Also thank all my teachers for their advice, opinion and support. The directives of these teachers prepared me to perform this task successfully.

Finally, I would also like to thank all my friends for their advice and support for my project.

ABSTRACT

The automation is an important part of the production in productive business. The business executives have to be pay attention to this level of business. Automating some of production is having different meanings in different business fields.

The Internet is a computer network made up of thousands of networks worldwide. No one knows exactly how many computers are connected to the Internet. It is certain, however, that these number in the millions and are increasing at a rapid rate.

No one is in charge of the Internet. There are organizations, which develop technical aspects of this network and set standards for creating applications on it, but no governing body is in control. Private companies own the Internet backbone, through which Internet traffic flows.

Internet café's are the new growing business field in today's market. Because they are serving Internet connection that is popular technological product in these days. As in other business fields, the Internet café's can be atomized for more productivity by a reservation system. Internet connection requires some devices as modems on a personal computer but for an Internet cafe, the hubs, Ethernet cards are needed with a dial-up connected computer and terminal computers. For an Internet cafe that requirements must be satisfied.

To organize the well-atomized Internet cafe we need well operating reservation program on computers in cafe. Reservation program realizes all operations well settled and faster.

The important point is controlling user's time information (service start and stop times) and other services used with Internet service. In a local network as in Internet Café (also as in Internet backbone) the server-client relation can be used for reservation programs. This means that the remote server machine provides files and services to the user's local client machine. Software can be installed on a client computer to take advantage of the latest access technology.

We can use a static application as in this project to solve this problem also again using client-server architecture.

In Internet cafe reservation program we have to use some visualizations and controls for customer satisfaction. So we can use graphical interfaces for more comprehensible programs to users.

Reservation programs mainly deals with basic information of user's as times and other services so this information has to be safely stored. That is a security problem to prevent unauthorized actions for the programs and eliminating escapes. The programs must not effect the other applications and must not slow down system.

TABLE OF CONTENTS

ACKNOWLEDGMENT	i
ABSTRACT	ii
TABLE OF CONTENTS	iii
INTRODUCTION	1
CHAPTER ONE: DEFINITION, PROPERTIES AND HISTORY OF VISUAL BASIC	2
1.1. Visual Basic Integrated Development Environment (IDE)	3
CHAPTER TWO: CHARACTERISTICS OF PROGRAM	6
2.1. The Main Properties of Server Program	6
2.2. The Main Properties of Client Program	6
2.3. Advantages of Software	6
CHAPTER THREE: STRUCTURE OF THE PROGRAM	7
3.1. Main Program (Server)	7
3.1.1. Password Verification for Enter Server Program	7
3.1.2. Permission process of server program	8
3.1.3. Changing Unit Prices of Services Process	9
3.1.4. Recording Orders and Calculate Charges Process	10
3.1.5. Calculation of Total Charges of Users Process	12
3.1.6. Reporting Process of server part of program	13
3.1.7. Transferring Time and Orders Data's of users to Report Database Process	14
3.2. Client Program(Terminal)	15
3.2.1. Password Verification for Client Program	15
3.2.2. Determining and Recording Starting and Stopping Time of User Process	16
3.2.3 Visibility Control of Start button Process	18
3.2.4. Sending Data to Report Database Process	19
CHAPTER FOUR: ADDITIONAL APPLICATIONS	20
4.1. CPU Performance Test Application	20
4.2. Alarm	23
4.3. Using another program on Client Program Process	24
4.4. Clock Application	26
4.5. Calender Application	27
4.6. Minimization of Client Program Process	27
CHAPTER FIVE: USED API'S AND COMPONENTS	29
5.1. Notifying Program Icon In System Tray	29
5.2. Hiding Program from Windows Task List	30
CONCLUSION	31
REFERENCES	32
APPENDIX A	33
APPENDIX B	58

INTRODUCTION

The Internet Cafe's are the new business fields in growing global market and they are serving Internet to their customers with a known cafeteria function. After Internet has started to be widespread over the world, more people have intrested with its facility in working. Because Internet was a global network that connects any distance. Peoples realized many useful services and more specific are designed with Internet.

In these days Internet had a global meaning because it permits to reach any information on computers connected in the world. By development of faster Internet communication and computers systems, it became a common source of information. Dial up connections are the frequent connection types for standard users. Still some computers are not connected to Internet because of requirements as a communication line, modem and Internet Service Providers (ISP). These requirements cause extra cost. In this respect Internet connectivity became a little difficult for computer users. Also Internet was having a restrictive structure for connection. The mobility has to be started having importance for Internet. But again mobile devices are still having very high prices.

Internet café's solves some problems by bringing today's personal computers to the public with Internet connections. Internet café's are sheltering the number of computers connected to Internet with traditional café service. For this reason Internet café's still has a growing popularity in market and the executives of this business are in a competition for giving a high quality service. Also need for Internet service provider's accelerated. To offset this demand, the Internet Cafe retooled and rededicated itself to providing the best Internet access.

As in many other business reservation has a great meaning to increasing production and revenues. At beginning they used some simple methods as ticketing. But that was not a conventional way of reservation because it has some security problems as escapes and operational costs. Then the computer programmers have developed some programs to coordinate operations. They prepared some programs that records the users times and calculates charges for services. Also operators are not needed to control one each of users. They have been able to control connections of the specific computers without dealing users or computers.

Programs are designed in Visual Basic programming language, which provides to create user interfaces for Windows. To simplify the task, Microsoft has built in to Visual Basic a software program to help you write your Visual Basic projects. Visual Basic provides more of the actual code for a programmer than any other non-visual programming language.

Visual Basic version 6.0 brings many new developing tools and controls to create more powerful applications. So the programs can easily be adapt to Windows operating system. Both programs are done by using Visual Basic language characteristics to organize entire network in the Internet café.

In this project I intended to design a specific program that performs all user transactions for Internet café.

1. DEFINITION, PROPERTIES AND HISTORY OF VISUAL BASIC

Visual Basic is a much-enhanced version of the BASIC programming language and the BASIC Integrated Development Environment (IDE). The bottom line of the enhancement is the VB can create Windows programs whereas BASIC could only create DOS programs. Ok, so the modifications are very major, but the idea holds true that Visual Basic is BASIC for Windows.

One of the many significant improvements is that VB provides massive support for easily creating the user interface to your applications. This is accomplished within the VB Integrated Development Environment (IDE), in which you use a mouse to "draw" your application and use the keyboard to type in the code that is to be executed. The single largest effect on coding that VB introduced was the concept of an event-driven programming model. In the old BASIC you had to write code to watch for the occurrence of user events (pressing a key, using the mouse). VB performs that function for you, and in fact, the only time code will execute in VB is in response to such an event!

And finally, the other major concept that VB has incorporated is the concept of objects. Objects provides a way to link together both code and data into a "package" in such a way as to make handling and saving the code/data more intuitively. VB forms are objects, menus are objects, and the so are the intrinsic VB controls.

VB has also provided a wide variety of built-in code that programmers once had to handle themselves. Of most significance is the built-in database handling features of VB. It is generally accepted that over half of all VB applications are written to handle databases! You'll find the built-in database features of VB to be very powerful, and that you can tap into them at whatever level of programming skill you possess.

A second area in which VB has begun to provide built-in support is that of Internet access. The VB features are still maturing, but with the tools available you can create very useful applications.

History of Visual Basic:

Basic was an introducing language for a new programmer because of it's easy to understand structure and commands. Language developed in early 1960's at **BASIC** (**B**eginner's **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode).

Answer to complicated programming languages (FORTRAN, Algol, Cobol...). First time-share language. In the mid-1970's, two college students write first Basic for a microcomputer (Altair) - cost \$350 on cassette tape. Every Basic since then essentially based on that early version. Examples include: GW-Basic, QBasic, and QuickBasic. Visual Basic was introduced in 1991. But nowadays Visual Basic is a high capacity language that the programmers can create professional applications and it is still has an easy to understand structure.

Starting with VB5, Visual Basic became an exclusively 32-bit programming language, suitable for programming only Win9X or NT systems. If you must program for Win 3.x, then you'll have to drop back to either VB3 or VB4, both of which are in pretty short supply. VB4 had the dual ability to support Win3.x as well as Win9X/NT systems but my personal recommendation is that if you need 32-bit system support, go straight to VB6 and if 16-bit is your need then stick with VB3.

The VB Learning Edition is the most affordable, and truth is that you can do a lot with it, particularly if you use the Windows API to augment its capabilities. However, in light of its better database features and its greater variety of controls, I suggest you go straight to the Professional Edition if at all possible. The price is steep, but it really does pay itself back in terms of time savings. If you need the VB Enterprise edition then you

should have it paid for by the "Enterprise" which requires it. Individual programmers generally do not need the Enterprise edition.

Programming Basics

To toss in a bit of philosophy, I view a programmer's job as getting to the end product with a minimum of effort. When you're paid by the hour you have an obligation to your employer (even if it's yourself) to minimize the cost/schedule of completing the assignment. To that end, I will emphasize over and over again the importance of using existing capabilities (such as your own reusable code, VB-provided controls, or even controls that you purchase). In light of this optimize-your-time philosophy, I also believe that every VB programmer should strive to become an expert in the following two areas: Databases and Reporting. It's very common for applications to store data and to provide that data in a printed format (by some estimates, 80% of all VB applications use databases). And while this tutorial will discuss various ways to perform these tasks, most programmers will find that the capabilities of VB for creating/editing Access (a Microsoft product) databases and VB's capabilities for reporting that data (as exemplified by the Crystal Reports control or the newer built-in VB reporting features) are the most effective tools you can use for increasing your effectiveness in creating VB applications.

That tools which provide similar functions will be one of the most commonly used tools of those available to a programmer and that any serious programmer must take the time to develop strong skills with these tools.

1.1. Visual Basic Integrated Development Environment (IDE)

It's a little known fact, but you can write a VB program from scratch using nothing more than a simple text editor, such as the Edit or Notepad applications, which come with Windows. In fact, the Visual Basic project files are exactly that text files. However, writing a project from scratch would involve a lot of tedious, detailed manual entries. To simplify the task, Microsoft has built in to VB a software program to help you write your VB projects. That software, known as the Integrated Development Environment (IDE for short) is what comes to the screen when you start VB and is the topic of this section.

Elements of Visual Basic

Like any other Windows application, VB consists of multiple windows, which appear at startup. The windows that are displayed when you start VB are collectively known as the Visual Basic Integrated Development Environment (IDE). When you first start VB all of the windows are locked together in what is called the MDI format. I prefer the SDI format (which you can set in the options menu), which allows each of the windows to be positioned independently on your screen. Here's a sample IDE screen which shows a VB project with one form on which is a single command button.

In particular, VB has the following windows:

Menu / Toolbar:

This is the only element of the IDE, which is always visible. You use it to select which other IDE elements to view and to add forms or controls to your project. There are many other features, which we will discuss later.



Figure 1.1. Menu/Toolbar of Visual Basic

Toolbox

The toolbox is simply a library of controls, which you can place on your application. Once you've placed all the controls you need onto your applications forms, you can hide the toolbox to make room for working in the other elements of the IDE.

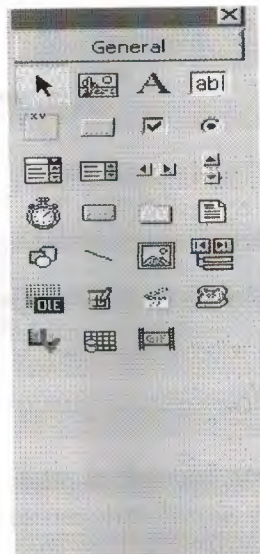


Figure 1.2. Toolbox of Visual Basic

Project Window:

This is simply a list of all the forms, which make up your VB project.

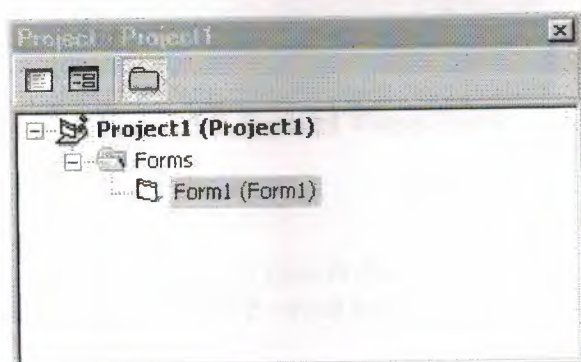


Figure 1.3. Project Window of Visual Basic

Property Window:

We'll talk about controls later, but such things as push buttons, scrolling text boxes, pictures boxes and other features of most VB applications allow you to enter parameters, which define how these controls work. In VB, these parameters are called properties. Some properties can be entered at design time within the IDE, while others must be entered with code while the program is running.

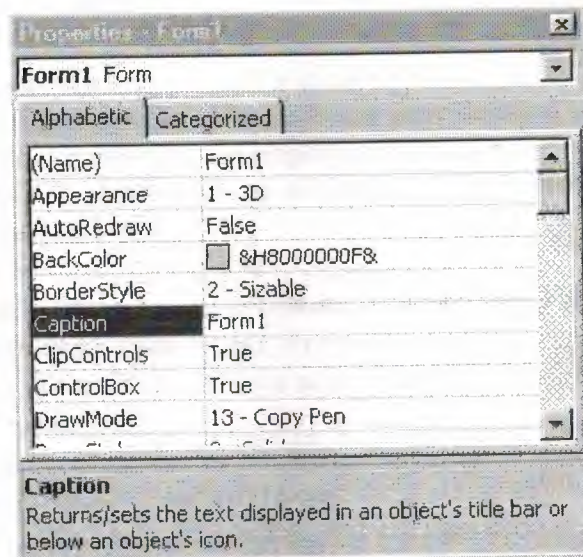


Figure 1.4. Property Windows of Visual Basic

Forms:

You add these to your VB application, as they are needed. They are the windows, which hold the various controls (buttons, text boxes, etc.), which make up your application.

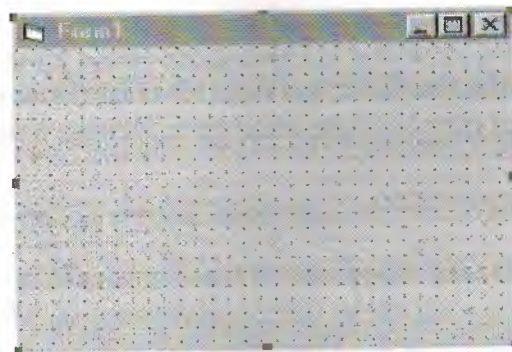


Figure 1.5. Form of Visual Basic

Code Window:

Like its name implies, this is where you type in the code that VB executes. Notice that the heading of the window indicates with which event the code is associated.

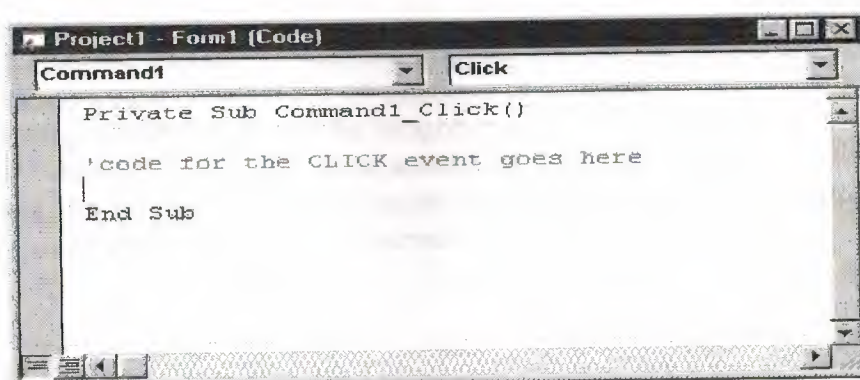


Figure 1.6. Code Window of Visual Basic

2.CHARACTERISTICS OF PROGRAM

These programs created for internet café's at all types and capacity. It makes easy to manage the internet café enterprises. That software provides to cut down the all loses and security problem that created by the employers of the enterprises. Also has a major advantage that is reduce the amount of the workers in that company. Because this software creates an automation.

This software has a server-client system.

- These separately programs communicate with each other owing to database.
- The main program (Server) works on at administrator computer.
- Each two part of the program has a simple interface for achieve easy to use and increase under stability.
- Terminal program are installed to whole computers, it shows time situation of each computer with start and stop timer.

The communication of these two program packet are very ordinary. Its communicate with each other over network. They connected each two program to database. The client (terminal) programs send or read some data from database. Also server (main) program does same operation.

2.1.The Main Properties of Server Program

Server program takes time data from database that is sent by the client program. It calculates order charges with multiply unit price of charges. Also it gives the permission to open or close the program whole computers for access to internet. Server program generates all time and orders reports for each individual computer. Server program has an administrative because it can reflect all users start and stop times can change hourly rates and clear accounts.

2.2.The Main Properties of Client Program

This software runs at whole computers. The user sends start time, stop time, elapsed time and date data's of the customer transactions to the main program from this client programs. The client program also has additional programs clock, alarm, calendar etc.

2.3.Advantages of Software

These software's has following advantages:

1. Operators can see all terminal computers start times, stop times, date of transactions from the server program.
2. the company can make profit from time or escapes by this management techniques.
3. The customer can see starting and stopping and elapsed time exactly for this reason increase the reliability of customer to company.
4. The customer reaches common programs from client program.
5. The customer can arrange alarm according to his/her wants.

Terminal and server program does not reside to system register for this reason it does not slow down to connection rates of internet.

3.STRUCTURE OF THE PROGRAM

This program are separated two parts. These are names main (server) and terminal (client). They uses for reduce the timing application and solve security problems of enterprises. The operator or operators can use simply this program. Because it has user friendly menus. Operator takes all data's about the users from client part of the program. These data's are starting time, stopping time, elapsed time, date and server part of the records orders information's and calculate charge of the orders and total charges that has to pay the user. Client part of the program is installed to whole terminal computers. User takes permission from operator to reach internet connection and other applications on the terminal computer. Operators give the permission of this specific computer with using the server program.

Client program also has an alarm, clock and calendar. If the user sets alarm then program warns the user with a message. Both of programs have a Benchmark program to test the performance of the computers.

3.1. Main Program (Server)

Main (server) program has time data's of all user transactions. If gets this data's from database that is entering by the client program. Server program gives permission to terminal computers for security. It has also security for operator entry.

3.1.1.Password Verification for Enter Server Program

This verification obtains protection to attacks that are coming to outside. We can show this application with flowchart same as at below.

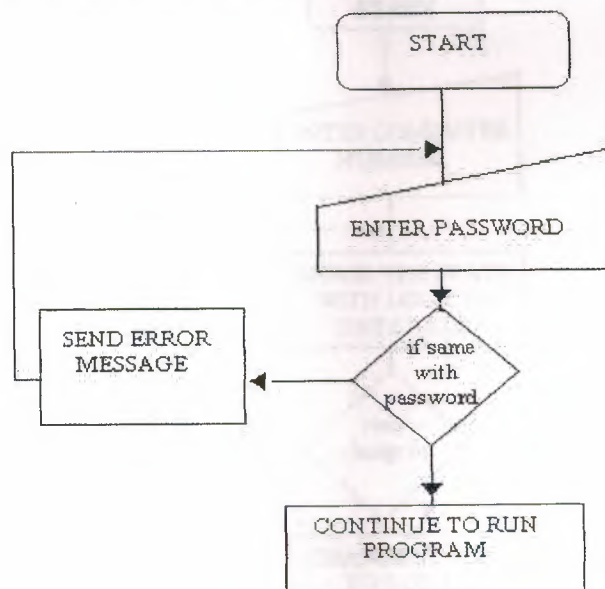


Figure3.1 The Flow of the Password verification program

We have to write some code if we want to realize this password protection. These codes same as follows:

```
Private Sub Command1_Click()  
If Text1.Text = "" Then  
Load Form3  
Form3.Show  
Unload Form2
```



```

Else
MsgBox ("Wrong keyword or accidentally caps lock open!")
Text1.Text = ""
Form2.Show
End If
End Sub

```

At this application we have to put labels, text boxes and command buttons for realize this application. All components have sub functions. It determines beginning of the object code. We can create some events with writing some codes between sub and end statements of each object.

3.1.2. Permission process of server program

This application uses for obtain protection of without permission entry of users. Main program (server) checks the database, which related with permission situation. There are two situations at permission table of database. These are 1 and 0 for each computer number. Server program can change this situation 1 to 0 or 0 to 1. If the operator change the situation client part of the program detect this change with simultaneously. Then it closes the visibility of buttons. Only change starts button visibility. If state is changed as 0 from server part of the program user can not use buttons and also internet because the program is designed to not to allow to use windows desktop and menus without pressing start button of the client program.

We can do this application same as following flow chart:

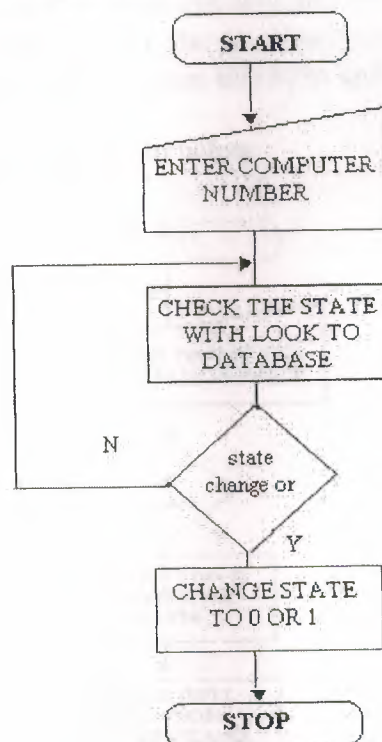


Figure3.2 The Flow of the Table Permission

The program codes of this process as follows:

```

Dim ölçüt As String
Private Sub Command1_Click()

```

```

compno$ = InputBox("Enter Computer number")
If compno = "" Then
Exit Sub
End If
ölçüt = "compno =" & compno$ & ""
Data1.Recordset.FindFirst ölçüt
End Sub

```

```

Private Sub Command2_Click()
Text1.Text = 1
Data1.UpdateRecord
End Sub

```

```

Private Sub Command3_Click()
Text1.Text = 0
Data1.UpdateRecord
End Sub

```

We use three command buttons for do this process computer number and change state of tables. This buttons provide to change to state of permission table of database.

3.1.3 Changing Unit Prices of Services Process

The server part of this software records the unit price of all services in a price table of database. It records coke, coffee, pizza etc. and also internet unit time price database.

Server program provide security in order to change this unit price of services. The operators have to enter password to open this form and changing prices of unit price of services.

The flowchart of this application given below:

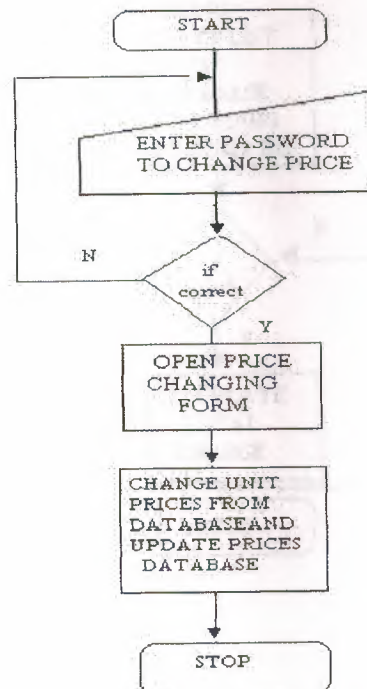


Figure 3.3. The Flow of the Changing Prices process

This part of the project does not require code because the text boxes linked with prices table of database. When the operators change the prices in a text box then close the form. This new data's records to database automatically with this way. Text boxes are linked with record sources data objects. Also data objects are linked to specify prices table of database.

3.1.4. Recording Orders and Calculate Charges Process

Customer can realize the taking orders of users process with use this process. This process resides in server part of the program. This form is connected to orders table of database. Customer can take orders of users with use this form. One table for the store amount of orders and another one in same database resides the unit prices of each orders. The users orders have been recorded according to their computer number. Charges of services are calculated by multiplying unit prices of order by amount of orders.

The flowchart of this application given below:

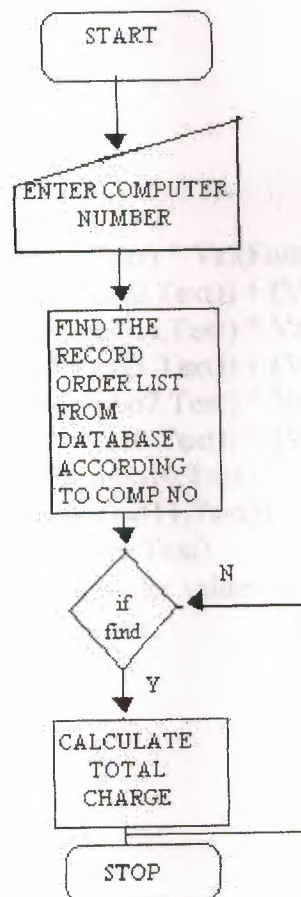


Figure.3.4. The Flow of Recording Orders and Calculate Charges process

The program codes of this process as follows:

```
Private Sub Command1_Click()  
computerno$ = InputBox(" Please Enter computer no! : ")  
If computerno = "" Then  
Exit Sub  
End If  
ölçüt = "computerno=" & computerno$ & ""  
Data1.Recordset.FindFirst ölçüt  
Data2.Recordset.FindFirst ölçüt  
End Sub
```

```
Private Sub Command2_Click()  
Combo1.Text = "0"  
Combo2.Text = "0"  
Combo3.Text = "0"  
Combo4.Text = "0"  
Combo5.Text = "0"  
Combo6.Text = "0"  
Combo7.Text = "0"  
Combo8.Text = "0"  
Combo9.Text = "0"  
Combo10.Text = "0"  
Combo11.Text = "0"  
End Sub
```

```
Private Sub Command3_Click()  
Form4.Text6.Text = (Val(Combo1.Text) * Val(Form6.Text1.Text)) +  
(Val(Combo2.Text) * Val(Form6.Text2.Text)) + (Val(Combo3.Text) *  
Val(Form6.Text3.Text)) + (Val(Combo4.Text) * Val(Form6.Text4.Text)) +  
(Val(Combo5.Text) * Val(Form6.Text5.Text)) + (Val(Combo6.Text) *  
Val(Form6.Text6.Text)) + (Val(Combo7.Text) * Val(Form6.Text7.Text)) +  
(Val(Combo8.Text) * Val(Form6.Text8.Text)) + (Val(Combo9.Text) *  
Val(Form6.Text9.Text)) + (Val(Combo10.Text) * Val(Form6.Text10.Text)) +  
(Val(Combo11.Text) * Val(Form6.Text11.Text))  
Form4.Text7.Text = Val(Form4.Text6.Text)  
'at below statements makes transfer order values to report database  
Text3.Text = Combo1.Text  
Text4.Text = Combo2.Text  
Text5.Text = Combo3.Text  
Text6.Text = Combo4.Text  
Text7.Text = Combo5.Text  
Text8.Text = Combo6.Text  
Text9.Text = Combo7.Text  
Text10.Text = Combo8.Text  
Text11.Text = Combo9.Text  
Text12.Text = Combo10.Text  
Text13.Text = Combo11.Text  
End Sub
```

We have to put 3 buttons to realize this process on form. This buttons resides at forms form calculates orders of charges, finding of computer number, clearing for new entry.

Multiplying values that taking prices from database and amount of services with use combo boxes over a form performs calculations.

3.1.5. Calculation of Total Charges of Users Process

Operator calculates all charge of certain users by using server part of program in this process. First operator has to find certain computer with search in the database. After find this computer number. Operator calculates orders of charges. This value is calculated with multiplying by unit price of the services by the amount of ordered services. When we press calculate total charges button program calculates internet charge of user and add this value to orders charge and finds the total charges. User has to press stop button at the client program for find the internet charge of user. When user press stop button client program calculates elapsed time and recorded to time table of the database. Server uses this record by connected to database. While calculate internet charge of the user. Server program multiply by unit price of internet connection by elapsed time for calculate internet charge. While server calculates orders of charge it multiply by amount of orders values by unit prices of orders.

Also operator can detect escapes from the charges form owing linking database. Because when user press start and stop button all data's about the program records time and computers numbers of users.

The program codes of this process as follows:

```
Private Sub Command1_Click()  
computerno$ = InputBox("Enter number of computer that you want to find? ")  
If computerno = "" Then  
Exit Sub  
End If  
ölçüt = "computerno =" & computerno$ & ""  
Data1.Recordset.FindFirst ölçüt  
End Sub
```

```
Private Sub Command3_Click()  
Data1.Recordset.Delete  
Data1.Recordset.MovePrevious  
Text5.Text = ""  
Text6.Text = ""
```

```
Private Sub Command5_Click()  
Text5.Text = Val(Text4.Text) * Val(Form6.Text12.Text)  
Text7.Text = Val(Text5.Text) + Val(Text6.Text)  
Data1.UpdateRecord  
End Sub
```


The flow of Calculation of Total Charges of User process at below:

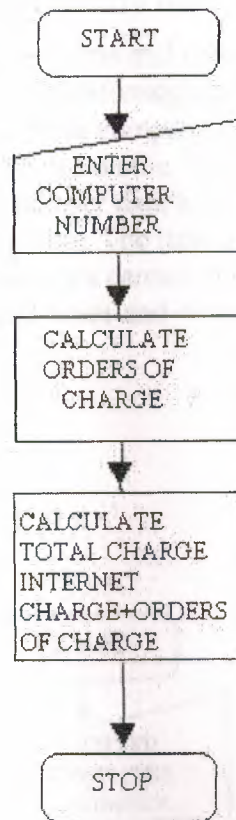


Figure.3.5. The Flow of Calculation of Total Charges process

3.1.6.Reporting Process of server part of program

Orders Report:

This process provides to see of all orders of computers. We can print the whole data's in the database or export like a file or web page. When the operator calculates order charges automatically this data's are transferred to orders table of report database by the server part of the program.

Time and Charge Report:

This report shows times and charges data's with linking time tables of database. Also we can print and export this report. This report calculate total amount of charges of all users and amount of average charge for all users. If operator delete the time data's from time and charge form at server part of the program. This data's are not erased from the reports because when the operator presses calculate total charges button data's are automatically transferred to reports database. For this reason the operator cannot delete this data's in no way. This property achieve protection of losses arise from operators.

Unit Prices of Services Report:

This reports shows unit prices of all services. We can print and export this data's. This report is connected to pries table of database.

3.1.7. Transferring Time and Orders Data's of users to Report Database Process

Server part of the program transfers time and orders data's of user, when operator pressed to the total charge button. Transferring operation is needed because operator deletes user data's after user leave from company to receive new user to this computer. Therefore this time and order data's is volatile. We cannot see user data's at the past. For this reason program transfers all user data's to report database, when the operator pressed to calculate total charge button. The time and order data's of users are permanent at report database. Operators cannot delete this user data's.

We can see the flow of transferred times and orders data's to time and charge report process:

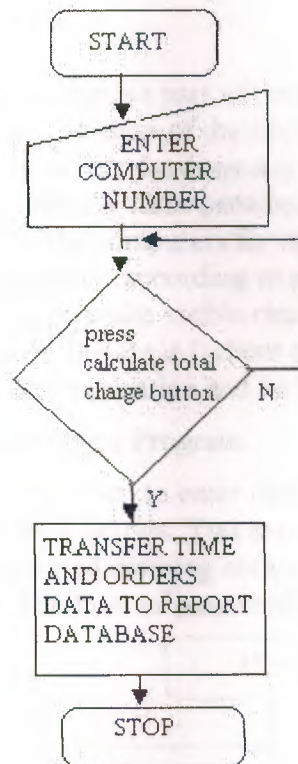


Figure.3.6. The Flow of Transferring Time and Orders data's to Report Database

The program codes of this process as follows:

```
Private Sub Command3_Click()  
Text3.Text = Combo1.Text  
Text4.Text = Combo2.Text  
Text5.Text = Combo3.Text  
Text6.Text = Combo4.Text  
Text7.Text = Combo5.Text
```

```

Text8.Text = Combo6.Text
Text9.Text = Combo7.Text
Text10.Text = Combo8.Text
Text11.Text = Combo9.Text
Text12.Text = Combo10.Text
Text13.Text = Combo11.Text
End Sub

```

```

Private Sub Command5_Click()
Data2.Recordset.AddNew
Text9.Text = Text1.Text
Text10.Text = Text2.Text
Text11.Text = Text3.Text
Text12.Text = Text4.Text
Text8.Text = Text7.Text
Text14.Text = Text13.Text
Data2.UpdateRecord
End Sub

```

3.2.Client Program

The client program is most considerable part of the internet reservation project because user use this part of the program. Mission of the client program is sent time information's of users to certain secure database and control the permission table with simultaneously. It controls periodically table permission states because client program has to detect permission state of the computers for make visible or invisible start button and other permission-required buttons according to permission state of computer. If permission state is 1 client program make visible required buttons and gives to permission to start timer of users. If state is 0 client program make invisible to start button and user cannot reach any application and he/she cannot start timer.

3.2.1. Password Verification for Client Program

The user has to enter a valid password to enter client program. This process is to defend program from unauthorized entries. You can see password verification process in figure.3.7. Operators also required entering computer numbers for terminals at startup for sending it to the database. The flow of password verification at below:

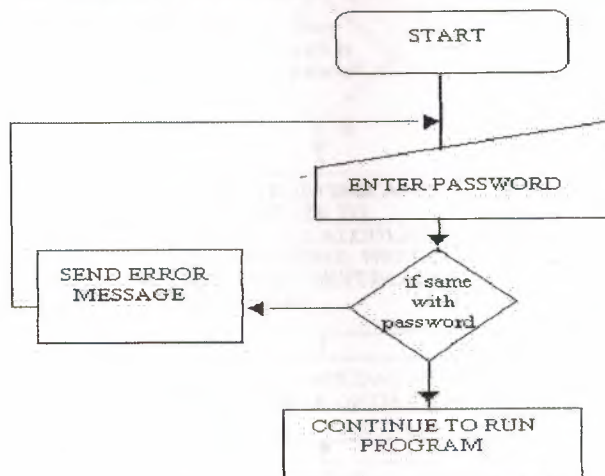


Figure.3.7.The Flow of Password Verification process

The Program codes this application as follow:

```
Private Sub Command1_Click()
```

```
If Text1.Text = "" Then
```

```
Load Form3
```

```
Form3.Show
```

```
Unload Form2
```

```
Else
```

```
MsgBox ("Wrong keyword or accidentally caps lock open!")
```

```
Text1.Text = ""
```

```
Form2.Show
```

```
End If
```

```
End Sub
```

At above codes check in the text box if it is true format it loads main form. If it is not true format it sends error message.

3.2.2.Determining and Recording Starting and Stopping Time of User Process

This process determines starting time of using services, which are taking from system. This event is realized by start button. After this event if user press stop button client part of the program takes time again from system this called stopping time. Elapsed time is calculated that are subtracting starting time from stopping time. Client program records this data's and also dates to time table of database. Recording event is connected stop button. A timer is used for rounded values for changes, the timer counts in each second by increasing a variable. This is more useful way for estimate internet price in a exactly. The flow of this process at below:

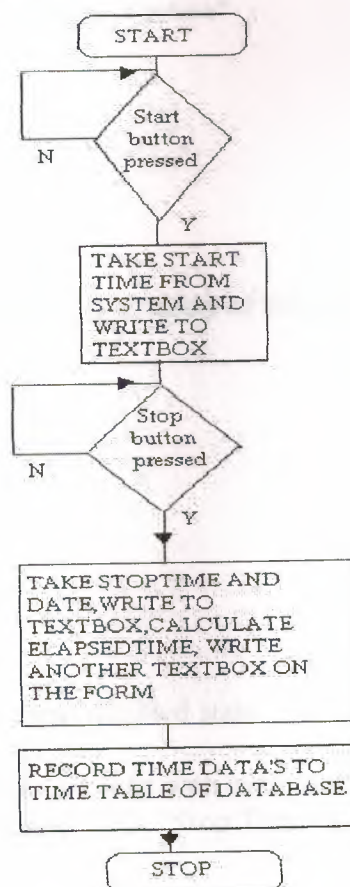


Figure.3.8.The Flow of Determining and Recording Starting and Stopping Time process

The codes below for realize this application:

```
Private Sub Command1_Click()  
'for below statement unlocks the enablety of this programs'  
drwatson.Enabled = True  
dialer.Enabled = True  
media.Enabled = True  
note.Enabled = True  
paint.Enabled = True  
player.Enabled = True  
scan.Enabled = True  
volume.Enabled = True  
'-----'  
Timer1.Enabled = True  
Command5.Enabled = True  
Command1.Visible = False  
Command4.Enabled = True  
timer.Enabled = False  
tx.Enabled = True  
st.Enabled = True  
Command2.Enabled = True  
'these satatements are worked to timer at above  
starttime = Now  
Text1.Text = Format(starttime, "hh:mm:ss")  
StatusBar1.Panels(1).Text = "Timer is active"  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""
```

End Sub

```
Private Sub Command2_Click()  
'for below statement locks again the enablety of this programs'  
drwatson.Enabled = False  
dialer.Enabled = False  
media.Enabled = False  
note.Enabled = False  
paint.Enabled = False  
player.Enabled = False  
scan.Enabled = False  
volume.Enabled = False  
'-----'  
Dim selection As Variant  
'when we press stop button turns disabled state  
'start button turns enabled  
selection = MsgBox("You cannot start with same account again if you stop now! Are  
you sure want to stop and leave?", 52, "Stop Timer")  
If selection = 6 Then  
Command4.Enabled = False
```

```
tx.Enabled = False
```

```
Command2.Enabled = False
```

```
Command1.Visible = True
```

```
Command5.Enabled = False
```

```
EndTime = Now
```

```
dat = Date
```

```
Text5.Text = dat
```

```
ElapsedTime = EndTime - starttime
```

```
Text2.Text = Format(EndTime, "hh:mm:ss")
```

```
Text3.Text = Format(ElapsedTime, "hh:mm:ss")
```

```
Timer1.Enabled = False
```

```
Text4.Text = i
```

```
i = 0
```

```
Form6.Data1.Recordset.AddNew
```

```
Form6.Text4.Text = Text4.Text
```

```
Form6.Text2.Text = Text1.Text
```

```
Form6.Text1.Text = Y
```

```
Form6.Text3.Text = Text2.Text
```

```
Form6.Text7.Text = Text5.Text
```

```
timer.Enabled = True
```

```
st.Enabled = False
```

```
End If
```

```
StatusBar1.Panels(1).Text = "Timer is not active"
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
i = i + 1
```

```
End Sub
```

Here the start time reflected to the user in a text box that taken from system and timer activated for counting seconds, which will be used to calculate Internet charge.

3.2.3. Visibility Control of Start button Process

This process protects the enterprise from non-authorized users. Client part of the program makes control to database for state of the permission. This detection process realized by timer. Timer events are arranged to check periodically the state of database, when the state of the permission 0, timer recognize this situation and makes invisible to start button. Therefore user cannot start to timer and cannot use any application on desktop. If state is 1 timer detect this situation and makes visible to start button automatically.

The Flow of this application as follow:

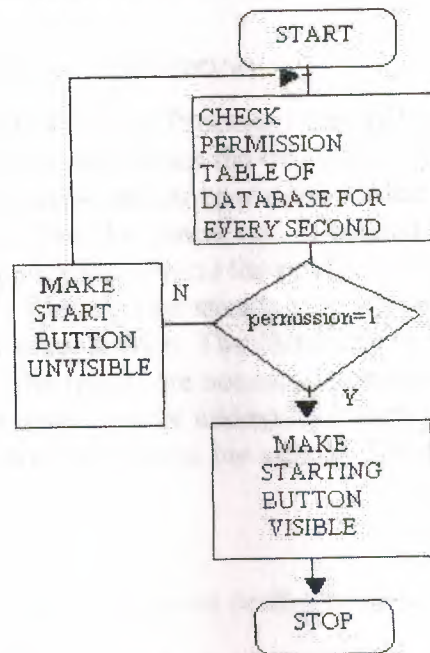


Figure.3.9. The Flow of Visibility Control of Start Button process

The program codes for this process at below:

```

Private Sub Timer2_Timer()
Form6.Data2.UpdateRecord
Form6.Data1.UpdateRecord
If Form6.Text6.Text = 0 Then
Command1.Enabled = False
Command2.Enabled = False
timer.Enabled = False
st.Enabled = False
Command4.Enabled = False
Else
Command1.Enabled = True
End If
End Sub
  
```

```

Private Sub Form_Load()
Form6.Refresh
End Sub
  
```

There is invisible form for check the database fields by the timer. Timer checks for visibility control this permission state of this table in every second.

3.2.4. Sending Data to Report Database Process

This process sends recorded of all data's to report database. This means all data's copied to report database time's data's (computer no, starting time, stopping time, elapsed time and date of the process). This property defends to enterprises from operators. Therefore we can see all time information's about all computers at report.

4.ADDITIONAL APPLICATIONS

4.1.CPU Performance Test Application

The both programs include a Central Processor Unit (CPU) performance benchmarking application that determines the calculation speed of your CPU. The application performs some mathematical operations for the floating-point and integer types. The times are taken from the system and calculated for each operation. At the end of calculations, the application reflects the results to the charts (activex). The charts are includes different CPU's at different speeds to compare with yours. Also these CPU's are tested with same application. The charts can be two dimensioned (2D) or three dimensioned (3D). The results are not certain because application runs under the program. Also the programmer must be adding Microsoft Chart Control (MSCHRT20.OCX) to his/her project to use charts. The structure is in Figure 4.1 and Figure 4.2.

The Flow of the integer and floating-point performance test of CPU application follow:

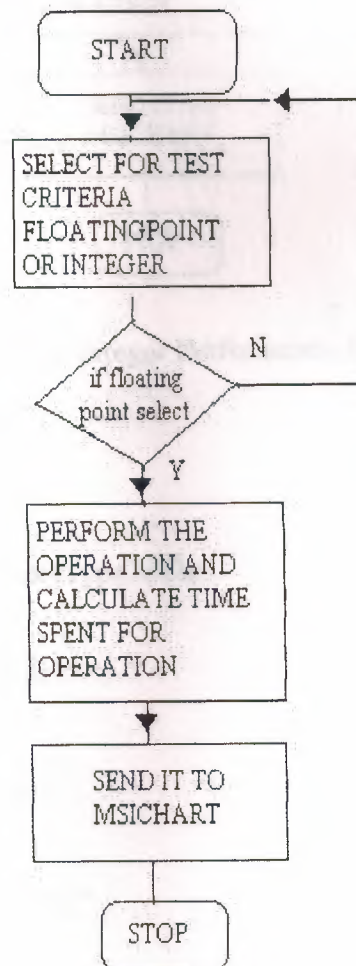


Figure.4.1. The Flow of the Floating-Point Performance Benchmark application

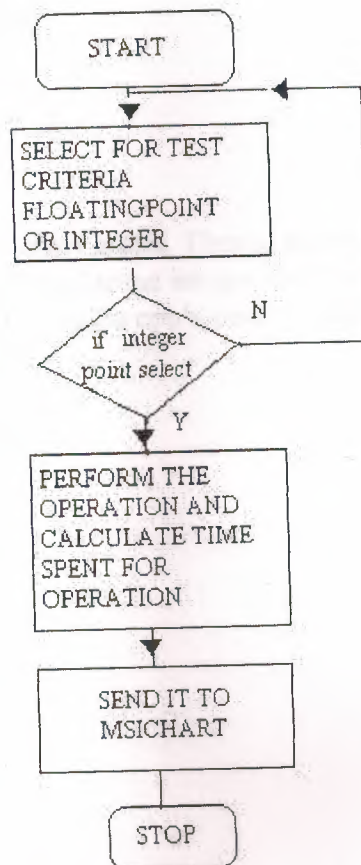


Figure.4.2. The Flow of the Integer Performance Benchmark application

The program codes are at below:

```

Private Sub Form_Load()
MSChart1.RowCount = 3
MSChart1.ColumnCount = 1
MSChart1.chartType = VtChChartType2dBar
MSChart1.Row = 1
MSChart1.Data = 228
MSChart1.RowLabel = "P2 233"
MSChart1.Row = 2
MSChart1.Data = 118
MSChart1.RowLabel = "686-120"
MSChart1.Row = 3
MSChart1.Data = 0
MSChart1.RowLabel = "Your CPU"
  
```

```

MSChart2.RowCount = 3
MSChart2.ColumnCount = 1
MSChart2.chartType = VtChChartType2dBar
MSChart2.Row = 1
  
```

```

MSChart2.Data = 27
MSChart2.RowLabel = "P2 233"
MSChart2.Row = 2
MSChart2.Data = 5.85
MSChart2.RowLabel = "686-120"
MSChart2.Row = 3
MSChart2.Data = 0
MSChart2.RowLabel = "Your CPU"
End Sub

```

Above codes provide to draw chart graph. Then at below codes perform the operation calculates the performance with send some integer and floating point numbers with certain time period to cpu, and compare cpu's performance with recorded criteria. After that sends this result to msichart.

```

Private Sub Command1_Click()
Dim i As Long, X As Long, t
t = Timer
Screen.MousePointer = 12
For i = 0 To 10000000
X = X + i + 10
X = 2 * i - 100 - X
Next
Screen.MousePointer = 0
t = Timer - t
MSChart1.Row = 3
MSChart1.Data = 100 / t
End Sub

```

```

Private Sub Command2_Click()
Dim i As Long, X As Long, t
t = Timer
Screen.MousePointer = 12
For i = 1 To 1000000
X = Log(i) * Sin(i) * Exp(Cos(i))
X = X / i
X = X * i
Next
Screen.MousePointer = 0
t = Timer - t
MSChart2.Row = 3
MSChart2.Data = 100 / t
End Sub

```

```

Private Sub Command3_Click()
Unload Me
End Sub

```


4.2.Alarm

The user can adjust the alarm to certain times 1 hour, 2 hour, 3 hour. This property provides to put limits to user him/herself. At the end of each alarm warns users with message.

To realize this application all codes are below:

```
Private Sub Command1_Click()  
Timer2.Enabled = False  
Timer1.Enabled = False  
Timer3.Enabled = False  
End Sub
```

```
Private Sub Form_Load()  
Dim k As Integer  
Timer2.Enabled = False  
Timer1.Enabled = False  
Timer3.Enabled = False  
End Sub
```

```
Private Sub Option1_Click()  
If Option1.Value = True Then  
Timer1.Enabled = True  
Timer2.Enabled = False  
Timer3.Enabled = False  
End If  
End Sub
```

```
Private Sub Option2_Click()  
If Option2.Value = True Then  
Timer2.Enabled = True  
Timer1.Enabled = False  
Timer3.Enabled = False  
End If  
End Sub
```

```
Private Sub Option3_Click()  
If Option3.Value = True Then  
Timer3.Enabled = True  
Timer1.Enabled = False  
Timer2.Enabled = False  
End If  
End Sub
```

```
Private Sub Timer1_Timer()  
k = k + 1  
If k = 180 Then  
MsgBox ("Time is up")  
End Sub
```

```

Private Sub Timer2_Timer()
k = k + 1
If k = 60 Then
MsgBox ("Time is up")
End If
End Sub
Private Sub Timer3_Timer()
k = k + 1
If k = 120 Then
MsgBox ("Time is up")
End If
End Sub

```

The Flowchart of the alarm application is at below:

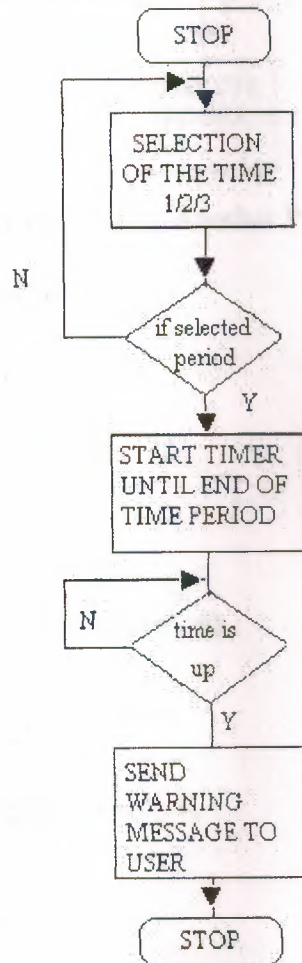


Figure.4.3.The Flow of the Alarm application

4.3.Using another program on Client Program Process:

Client program support some basic programs (mediaplayer, notepad, paintbrush, Drwatson, dialer, cdplayer, scandisk, volume control, internet explorer) to user can execute this program without leave the client program. This application provides to useful and basic programs to user and he/she can reach these programs easily and rapidly. Shell command executes other programs from over client program.

The Flow of the Using Another Program On Client Program Application is below:

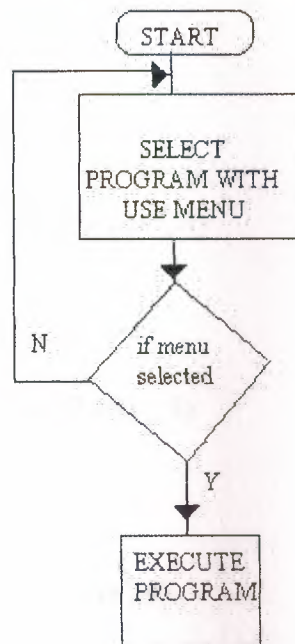


Figure.4.4. The Flow of the Using Another Program On Client Program application

The program codes are at below:

```
Private Sub dialer_Click()
```

```
Dim p
```

```
p = Shell("c:\windows\Dialer.exe", 4)
```

```
End Sub
```

```
Private Sub drwatson_Click()
```

```
Dim k
```

```
k = Shell("c:\windows\Drwatson.exe", 4)
```

```
End Sub
```

```
Private Sub media_Click()
```

```
Dim t
```

```
t = Shell("c:\windows\Mplayer.exe", 4)
```

```
End Sub
```

```
Private Sub note_Click()
```

```
Dim s
```

```
s = Shell("c:\windows\notepad.exe", 4)
```

```
End Sub
```

```
Private Sub paint_Click()
```

```
Dim l
```

```
l = Shell("c:\windows\pbrush.exe", 4)
```

```
End Sub
```



```
Private Sub player_Click()
Dim z
z = Shell("c:\windows\cdplayer.exe", 4)
End Sub
```

```
Private Sub scan_Click()
Dim w
w = Shell("c:\windows\scandiskw.exe", 4)
End Sub
```

```
Private Sub volume_Click()
Dim g
g = Shell("c:\windows\sndvol32.exe", 4)
End Sub
```

```
Private Sub int Click()
Dim n
n=Shell("c:\windows\explorer.exe",4)
End Sub
```

4.4.Clock Application

Client program has a clock for use a user. This clock has classical and digital shape it has a very user-friendly shapes.

The program codes of the clock application follow:

```
Private Sub Form_Load()
Width = ScaleHeight
Scale (-20, 20)-(20, -20)
Timer1.Interval = 1000 'saniye
AutoRedraw = True
Circle (0, 0), 20
DrawMode = 7 'xor
End Sub
```

```
Private Sub Timer1_Timer()
Dim aci, saniye, dakika, saat
Static sx, sy, dx, dy, stx, sty
Caption = Time
DrawWidth = 1
Line (0, 0)-(sx, sy), QBColor(10) 'oncekinin uzerine cizerek sil
saniye = Second(Time) 'saniyeyi saatten al
aci = saniye * 6 + 90 'her bir saniye 6 derecedir. 360 derece/60 saniye
sx = 20 * Cos(aci * 3.1415 / 180)
sy = 20 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(sx, sy), QBColor(10) 'saniye ciz
DrawWidth = 3
Line (0, 0)-(-dx, -dy), QBColor(11) 'oncekinin uzerine cizerek sil
dakika = Minute(Time)
aci = dakika * 6 + 90 'her dakika 6 derecedir.360 derece/60 dakika
dx = 20 * Cos(aci * 3.1415 / 180)
```

```

dy = 20 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(-dx, -dy), QBColor(11) 'yelkovanı cizer
DrawWidth = 3
Line (0, 0)-(-stx, -sty), QBColor(12) 'oncekinin uzerine czerek sil
saat = Hour(Time)
aci = saat * 30 + 90 'her saat 30 derece
stx = 12 * Cos(aci * 3.1415 / 180)
sty = 12 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(-stx, -sty), QBColor(12) 'akrebi ciz

```

End Sub

4.5. Calender Application

Very simply the MSCAL.OCX file creates these applications. When we add this ocx file to our project. We can reach easily very useful calendar. User can use calendar without need any permission. We do not need to write program code for create this application.

4.6. Minimization of Client Program Process

User can minimize and send to task bar to client program after press start button therefore he/she can access the all application at the terminal computer.

Flow of minimization of client program is at below:

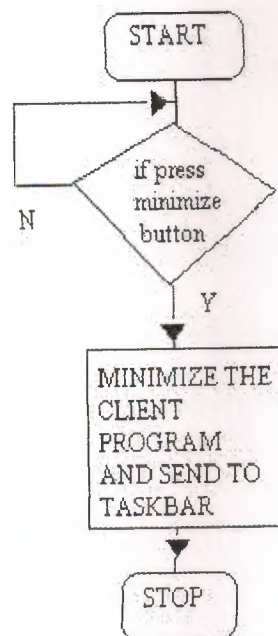


Figure.4.5. The Flow of the Minimization of Client Program application

The program codes for realized this application are below:

```
Private Sub Command4_Click()  
Dim minimized As VbAppWinStyle  
Form3.WindowState = 1 - minimized  
End Sub
```


5. USED API'S AND COMPONENTS

API (Advanced Programmers Interface) is a set of predefined Windows functions used to control the appearance and behavior of every Windows element (from the outlook of the desktop window to the allocation of memory for a new process). Every user action causes the execution of several or more API function telling Windows what's happened. It is something like the native code of Windows. Other languages just act as a shell to provide an automated and easier way to access APIs. Visual Basic has done a lot in this direction. It has completely hidden the APIs and provided a quite different approach for programming under Windows. So it includes many functions to organize that controls. Visual Basic can use that controls globally inside the project by declaring it in a module as a function or a subprogram. Programmer must load an API once from its located DLL or EXE extended file to use it in his/her project. The API functions reside in DLLs (like User32.dll, GDI32.dll, Shell32.dll...) in the Windows system directory.

5.1. Notifying Program Icon In System Tray

Windows 9/x versions supports "System Tray". The window at the right corner of the progress bar is the System Tray. Programmer is able to add his/her program's icon to the System Tray by using Shell_NotifyIcon API. Programmer can define some properties that belong application. Application is defined with NOTIFYICONDATA type that includes different parameters.

cbSize	Variable length
hWnd	HWND number of the form that will be settled to Sytem Tray
uCallbackMessage	The event that should be called when user makes a process on the icon
hIcon	Defines the icon which will be settled in System Tray
szTip	ToolTipText

Table 5.1. The parameters of type NOTIFYICONDATA

The program codes for adding icon to the System Tray are:

```
Private Declare Function Shell_NotifyIcon Lib "shell32.dll" Alias "Shell_NotifyIconA"  
(ByVal dwMessage As Long, lpData As NOTIFYICONDATA) As Long  
Private Type NOTIFYICONDATA  
    cbSize As Long  
    hWnd As Long  
    uID As Long  
    uFlags As Long  
    uCallbackMessage As Long  
    hIcon As Long  
    szTip As String * 64  
End Type  
Dim tray As NOTIFYICONDATA
```

```

Private Sub Form_Load()
Text2.Text = ""
tray.cbSize = Len(tray)
tray.hwnd = Form3.hwnd
tray.szTip = "Net@Cafe" + Chr(0)
tray.uCallbackMessage = &H200
tray.uFlags = 7
tray.hIcon = Form3.Icon
Shell_NotifyIcon 0, tray
End Sub

```

5.2. Hiding Program From Windows Task List

These functions are used for security on the client program. User's can close the program by using Task List (Ctrl+Alt+Del) so it can be distrustful. By using some API's the programmers can hide their program in Task List. To do this, the programmer must add below code to the active form's load and unload events after declared API function.

Declarations are:

```

Private Declare Function GetCurrentProcessId Lib "kernel32" () As Long
Private Declare Function GetCurrentProcess Lib "kernel32" () As Long
Private Declare Function RegisterServiceProcess Lib "kernel32" (ByVal dwProcessID
As Long, ByVal dwType As Long) As Long
Private Const RSP_SIMPLE_SERVICE = 1
Private Const RSP_UNREGISTER_SERVICE = 0
Codes are:

```

```

Private Sub Form_Load()
Dim id As Long
Dim X As Long
id = GetCurrentProcessId()
X = RegisterServiceProcess(id, RSP_SIMPLE_SERVICE)
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
Dim id As Long
Dim X As Long
id = GetCurrentProcessId()
X = RegisterServiceProcess(pid, RSP_UNREGISTER_SERVICE)
End Sub

```


CONCLUSION

Visual Basic is known for its ability to allow a programmer to quickly, and easily, create a Windows program. In part, this is possible because objects provide code that you don't have to write.

In Visual Basic, the basic building block of an application is a form, which is simply a window. The Visual Basic IDE can insert forms into your project, and then you can resize the forms as well as change other properties of the form. However, controls (checkboxes, textboxes...) are also windows. A form is distinguished from a control in that only forms can exist as standalone objects. When controls are used, they must be placed in a form and there are a few exceptions such as the printer object or the screen object, which are not considered part of any form, but are part of a Visual Basic program.

It's worth noting that when managers talk about programmers, one of the common metrics used to describe performance is "lines of code per month". There are all kinds of debate about how good a metric this is, but the fact is that the metric is used. It's not that you don't get credit for novel algorithms, or that you won't be a hero to fellow programmers when they see how you solved a problem with 10 lines of code that took them 100. You'll get that credit (and mental satisfaction) but looking at the big picture it's clear that the volume of code you can crank out will be the visible result of your efforts. Writing a program is essentially a design task, aimed at solving a problem. Like most things, there are many design approaches for every problem. Some solve the problem by grinding out an answer. Some are very elegant. Others are so complex that even the designer has a hard time keeping up with the convolutions of the approach. Here the specifications of a great Visual Basic programmer:

- Great programmers must be coding experts
- Great programmers understand that there are many approaches to a problem
- Great programmers work efficiently.

VB provides two controls, which make the link to the database file, and which creates the recordset that is exposed to the rest of the controls in your application. The two are identical in concept but differ in the flexibility they offer to the programmer. Data controls are the most important part of the programs, and applications. By adding user's information on a database file, the programmers can use this file by adding, reading and deleting fields on it.

In case you have any question about the value of objects, that objects (in the form of controls) are the single biggest reason why Visual Basic has been so successful. In Visual Basic 6.0, Microsoft has expanded the object features of Visual Basic, making it even more powerful than ever.

This Internet café reservation program contains user-friendly forms, some API's, Activex controls on it to fortify the application. It provides sensible security and usability for users and operators. The structure of program is built up on to client-server relation. Internet café's finds appropriate solution to their security or automation problems, but human relation between users and operators still important in the enterprises.

This project uses some Internet café reservation and automation techniques. It supports security and automation advantage to manage calculation of charges and reservation transactions. This program provides these possibilities with use Visual Basic functions, database controls and other many resources of Visual Basic.

REFERENCES

- [1] Douglas E. Comer, The Internet Book: Computer Networking and How to Internet Works, Prentice-Hall International, Inc., NJ, 1994
- [2] İ.Karagülle – Z. Pala, Visual Basic 6.0 Pro: Visual Basic programming for professional users, Türkmen Kitabevi, Beyazıt, İSTANBUL, 1999
- [3] Visual Basic Forum Site. Retrieved May 15, 2002 from the World Wide Web: <http://www.vbonline.com>.
- [4] Visual Basic Programmers Site. Retrieved May 22, 2002 from the World Wide Web: <http://vbi.org>.
- [5] VBNet A guide for sample applications. Retrieved May 23, 2002 from the World Wide Web: <http://www.mvps.org/vbnet/>.
- [6] Visual Basic Program Developers Suggestions. Retrieved May 27, 2002 from the World Wide Web: <http://gan.dalf.pht.com/pub/vbbbn/cgi/am.cgi>.
- [7] Microsoft Visual Basic Developers Site. Retrieved June 1, 2002 from World Wide Web: <http://www.Microsoft.com/vbasic/>.
- [8] Visual Basic Programming Tutorials Site. Retrieved June 3, 2002 From World Wide Web: <http://www.programmingtutorials.com>.
- [9] İ.Karagülle-Z.Pala, Visual Basic Pro 5.0 : Visual Basic Programming, Türkmen Kitabevi, Beyazıt, İSTANBUL, 1997

APPENDIX A

We can see all codes for both parts of the programs in visual basic language. This appendix provide to us more comprehensive look to project.

Main Program (Server)

Codes are in an ordered way according to their form numbers.

Form1(Table Permission):

Dim ölçüt As String

```
Private Sub Command1_Click()  
compno$ = InputBox("Enter Computer number")  
If compno = "" Then  
Exit Sub  
End If  
ölçüt = "compno =" & compno$ & ""  
Data1.Recordset.FindFirst ölçüt  
End Sub
```

```
Private Sub Command2_Click()  
Text1.Text = 1  
Data1.UpdateRecord  
End Sub
```

```
Private Sub Command3_Click()  
Text1.Text = 0  
Data1.UpdateRecord  
End Sub
```

```
Private Sub Form_Load()
```

```
End Sub  
Private Sub Text2_KeyPress(KeyAscii As Integer)  
If Chr(KeyAscii) = "a" Or Chr(KeyAscii) = "A" Or Chr(KeyAscii) = "a" Or  
Chr(KeyAscii) = "b" Or Chr(KeyAscii) = "B" Or Chr(KeyAscii) = "C" Or  
Chr(KeyAscii) = "c" Or Chr(KeyAscii) = "D" Or Chr(KeyAscii) = "d" Or  
Chr(KeyAscii) = "E" Or Chr(KeyAscii) = "e" Or Chr(KeyAscii) = "F" Or  
Chr(KeyAscii) = "f" Or Chr(KeyAscii) = "G" Or Chr(KeyAscii) = "g" Or  
Chr(KeyAscii) = "Ğ" Or Chr(KeyAscii) = "ğ" Or Chr(KeyAscii) = "H" Or  
Chr(KeyAscii) = "h" Or Chr(KeyAscii) = "I" Or Chr(KeyAscii) = "i" Or Chr(KeyAscii)  
= "İ" Or Chr(KeyAscii) = "i" Or Chr(KeyAscii) = "J" Or Chr(KeyAscii) = "j" Or  
Chr(KeyAscii) = "K" Or Chr(KeyAscii) = "k" Or Chr(KeyAscii) = "L" Or  
Chr(KeyAscii) = "l" Or Chr(KeyAscii) = "M" Or Chr(KeyAscii) = "m" Or  
Chr(KeyAscii) = "N" Or Chr(KeyAscii) = "n" Or Chr(KeyAscii) = "O" Or  
Chr(KeyAscii) = "o" Or Chr(KeyAscii) = "Ö" Or Chr(KeyAscii) = "ö" Or  
Chr(KeyAscii) = "P" Or Chr(KeyAscii) = "p" Or Chr(KeyAscii) = "R" Or  
Chr(KeyAscii) = "r" Or Chr(KeyAscii) = "S" Or Chr(KeyAscii) = "s" Or Chr(KeyAscii)  
= "Ş" Then
```

```

KeyAscii = 0
Beep
End If
If Chr(KeyAscii) = "ş" Or Chr(KeyAscii) = "T" Or Chr(KeyAscii) = "t" Or
Chr(KeyAscii) = "U" Or Chr(KeyAscii) = "u" Or Chr(KeyAscii) = "Ü" Or
Chr(KeyAscii) = "ü" Or Chr(KeyAscii) = "V" Or Chr(KeyAscii) = "v" Or
Chr(KeyAscii) = "Y" Or Chr(KeyAscii) = "y" Or Chr(KeyAscii) = "Z" Or
Chr(KeyAscii) = "z" Or Chr(KeyAscii) = "X" Or Chr(KeyAscii) = "x" Or
Chr(KeyAscii) = "Q" Or Chr(KeyAscii) = "q" Or Chr(KeyAscii) = "W" Or
Chr(KeyAscii) = "w" Or Chr(KeyAscii) = "Ç" Or Chr(KeyAscii) = "ç" Or
Chr(KeyAscii) = "<" Or Chr(KeyAscii) = ">" Or Chr(KeyAscii) = "." Or
Chr(KeyAscii) = ":" Or Chr(KeyAscii) = "," Or Chr(KeyAscii) = ";" Or Chr(KeyAscii)
= "-" Or Chr(KeyAscii) = "*" Or Chr(KeyAscii) = "0" Or Chr(KeyAscii) = "é" Or
Chr(KeyAscii) = "!" Or Chr(KeyAscii) = "" Or Chr(KeyAscii) = "^" Or Chr(KeyAscii)
= "+" Or Chr(KeyAscii) = "%" Or Chr(KeyAscii) = "&" Or Chr(KeyAscii) = "/" Or
Chr(KeyAscii) = "(" Or Chr(KeyAscii) = ")" Or Chr(KeyAscii) = "=" Or Chr(KeyAscii)
= "?" Or Chr(KeyAscii) = "#" Or Chr(KeyAscii) = "$" Then
KeyAscii = 0
Beep
End If
End Sub

```

```

Private Sub Text1_Change()
Dim Cancel As Variant
If Val(Text1) < 0 Or Val(Text1) > 1 Then
MsgBox ("You should enter 1 or 0 for state")
Text1.Text = "0"
Cancel = True
End If
End Sub

```

Form2 (Password verification):

```

Private Sub Command1_Click()
If Text1.Text = "" Then
Load Form3
Form3.Show
Unload Form2
Else
MsgBox ("Wrong keyword or accidentally caps lock open!")
Text1.Text = ""
Form2.Show
End If
End Sub

```

```

Private Sub Image1_Click()

```

```

End Sub

```


Private Sub Picture2_Click()

End Sub

Private Sub Form_Load()

End Sub

Form3 (Main Form):

Private Declare Function exitwindowsex Lib "user32" (ByVal uflags As Long, ByVal dwreserved As Long) As Long

Dim ölçüt As String

Private Declare Function Shell_NotifyIcon Lib "shell32.dll" Alias "Shell_NotifyIconA" (ByVal dwMessage As Long, lpData As NOTIFYICONDATA) As Long

Private Type NOTIFYICONDATA

cbSize As Long

hwnd As Long

uID As Long

uflags As Long

uCallbackMessage As Long

hIcon As Long

szTip As String * 64

End Type

Dim tray As NOTIFYICONDATA

Private Sub ab_Click(Index As Integer)

frmAbout.Show

End Sub

Private Sub AniGIF1_DblClick()

Call Shell("start mailto:ibrahimdokuzer@hotmail.com")

End Sub

Private Sub AniGIF1_Step()

End Sub

Private Sub banchmerk_Click(Index As Integer)

Form8.Show

End Sub

Private Sub closing_Click(Index As Integer)

Form1.Show

End Sub

Private Sub Command1_Click()

Form4.Show

End Sub

```

Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Charges"
End Sub

```

```

Private Sub Command2_Click()
Form10.Show
End Sub

```

```

Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Unit Cost of Services "
End Sub

```

```

Private Sub Command3_Click()
Form5.Show
End Sub

```

```

Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Orders of Services"
End Sub

```

```

Private Sub Command4_Click()
Form9.Show
End Sub

```

```

Private Sub Command4_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Service List of All Tables "
End Sub

```

```

Private Sub Command5_Click()
Form8.Show
End Sub

```

```

Private Sub Command5_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "CPU Test "
End Sub

```

```

Private Sub Command6_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Report"
Form11.Show
End Sub

```

```

Private Sub Command7_Click()
Form7.Show
End Sub

```

```

Private Sub Command7_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Exit"
End Sub

```

```

Private Sub Command8_Click(Index As Integer)
Form1.Show
End Sub

```

```

Private Sub Command8_MouseMove(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Open/Close Permission "
End Sub

```

```

Private Sub dharges_Click(Index As Integer)
Form4.Show
End Sub

```

```

Private Sub exit_Click(Index As Integer)
Form7.Show
End Sub

```

```

Private Sub Form_Load()
shut.Enabled = False
tray.cbSize = Len(tray)
tray.hwnd = Form3.hwnd
tray.szTip = "Net@Cafe" + Chr(0)
tray.uCallbackMessage = &H200
tray.uflags = 7
tray.hIcon = Form3.Icon
Shell_NotifyIcon 0, tray
If App.PrevInstance = True Then
MsgBox ("Program is already running")

```



```
End  
End If  
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
StatusBar1.Panels(1).Text = "Server program is active"  
End Sub
```

```
Private Sub list_Click(Index As Integer)  
Form9.Show  
End Sub
```

```
Private Sub number_Click(Index As Integer)  
Form1.Show  
End Sub
```

```
Private Sub opening_Click(Index As Integer)  
Form1.Show  
End Sub
```

```
Private Sub orders_Click(Index As Integer)  
Form5.Show  
End Sub
```

```
Private Sub tech_Click(Index As Integer)  
MsgBox ("You can send E-mail to ibrahimdokuzer@hotmail.com for technical  
support")  
End Sub
```

```
Private Sub unit_Click(Index As Integer)  
Form10.Show  
End Sub
```

Form4 (Charges):

```
Private Sub Command1_Click()  
computerno$ = InputBox("Enter number of computer that you want to find? ")  
If computerno = "" Then  
Exit Sub  
End If  
ölçüt = "computerno =" & computerno$ & ""  
Data1.Recordset.FindFirst ölçüt  
End Sub
```

```
Private Sub Command2_Click()  
Form9.Show  
End Sub
```

```
Private Sub Command3_Click()  
Data1.Recordset.Delete
```

```
Data1.Recordset.MovePrevious
Text5.Text = ""
Text6.Text = ""
```

```
End Sub
```

```
Private Sub Command4_Click()
Unload Form4
Form3.Show
End Sub
```

```
Private Sub Command5_Click()
Text5.Text = Val(Text4.Text) * Val(Form6.Text12.Text)
Text7.Text = Val(Text5.Text) + Val(Text6.Text)
Data1.UpdateRecord
Data2.Recordset.AddNew
Text9.Text = Text1.Text
Text10.Text = Text2.Text
Text11.Text = Text3.Text
Text12.Text = Text4.Text
Text8.Text = Text7.Text
Text14.Text = Text13.Text
Data2.UpdateRecord
End Sub
```

```
Private Sub Command6_Click()
Form5.Show
End Sub
```

```
Private Sub Form_Load()
End Sub
```

Form5 (Orders):

```
Private Sub Command1_Click()
computerno$ = InputBox(" Please Enter computer no! : ")
If computerno = "" Then
Exit Sub
End If
ölçüt = "computerno=" & computerno$ & ""
Data1.Recordset.FindFirst ölçüt
Data2.Recordset.FindFirst ölçüt
End Sub
```

```
Private Sub Command2_Click()
Combo1.Text = "0"
Combo2.Text = "0"
Combo3.Text = "0"
Combo4.Text = "0"
Combo5.Text = "0"
Combo6.Text = "0"
```



```
Combo7.Text = "0"  
Combo8.Text = "0"  
Combo9.Text = "0"  
Combo10.Text = "0"  
Combo11.Text = "0"  
End Sub
```

```
Private Sub Command3_Click()  
Form4.Text6.Text = (Val(Combo1.Text) * Val(Form6.Text1.Text)) +  
(Val(Combo2.Text) * Val(Form6.Text2.Text)) + (Val(Combo3.Text) *  
Val(Form6.Text3.Text)) + (Val(Combo4.Text) * Val(Form6.Text4.Text)) +  
(Val(Combo5.Text) * Val(Form6.Text5.Text)) + (Val(Combo6.Text) *  
Val(Form6.Text6.Text)) + (Val(Combo7.Text) * Val(Form6.Text7.Text)) +  
(Val(Combo8.Text) * Val(Form6.Text8.Text)) + (Val(Combo9.Text) *  
Val(Form6.Text9.Text)) + (Val(Combo10.Text) * Val(Form6.Text10.Text)) +  
(Val(Combo11.Text) * Val(Form6.Text11.Text))  
Form4.Text7.Text = Val(Form4.Text6.Text)  
Text3.Text = Combo1.Text  
Text4.Text = Combo2.Text  
Text5.Text = Combo3.Text  
Text6.Text = Combo4.Text  
Text7.Text = Combo5.Text  
Text8.Text = Combo6.Text  
Text9.Text = Combo7.Text  
Text10.Text = Combo8.Text  
Text11.Text = Combo9.Text  
Text12.Text = Combo10.Text  
Text13.Text = Combo11.Text  
End Sub
```

```
Private Sub Form_Load()  
Combo1.AddItem "0"  
Combo1.AddItem "1"  
Combo1.AddItem "2"  
Combo1.AddItem "3"  
Combo2.AddItem "0"  
Combo2.AddItem "1"  
Combo2.AddItem "2"  
Combo2.AddItem "3"  
Combo3.AddItem "0"  
Combo3.AddItem "1"  
Combo3.AddItem "2"  
Combo3.AddItem "3"  
Combo4.AddItem "1"  
Combo4.AddItem "2"  
Combo4.AddItem "3"  
Combo5.AddItem "1"  
Combo5.AddItem "2"  
Combo5.AddItem "3"  
Combo6.AddItem "1"
```



```

Combo6.AddItem "2"
Combo6.AddItem "3"
Combo7.AddItem "1"
Combo7.AddItem "2"
Combo7.AddItem "3"
Combo8.AddItem "1"
Combo8.AddItem "2"
Combo8.AddItem "3"
Combo9.AddItem "1"
Combo9.AddItem "2"
Combo9.AddItem "3"
Combo10.AddItem "1"
Combo10.AddItem "2"
Combo10.AddItem "3"
Combo11.AddItem "1"
Combo11.AddItem "2"
Combo11.AddItem "3"
End Sub

```

Form6 (Cost of Orders):

```

Private Sub Command1_Click()
Unload Form6
Form3.Show
End Sub

```

```

Private Sub Command2_Click()
Unload Form6
End Sub

```

```

Private Sub Form_Load()
End Sub

```

Form7 (Exit):

```

Private Sub Command1_Click()
End
End Sub

```

```

Private Sub Command2_Click()
Unload Me
End Sub

```

```

Private Sub Label1_Click()
End Sub

```

Form8 (CPU Tests):

```

Private Sub Command1_Click()
Dim i As Long, X As Long, t
t = Timer
Screen.MousePointer = 12
For i = 1 To 1000000

```

```

X = Log(i) * Sin(i) * Exp(Cos(i))
X = X / i
X = X * i
Next
Screen.MousePointer = 0
t = Timer - t
MSChart2.Row = 3
MSChart2.Data = 100 / t
End Sub

```

```

Private Sub Command2_Click()
Dim i As Long, X As Long, t
t = Timer
Screen.MousePointer = 12
For i = 0 To 10000000
X = X + i + 10
X = 2 * i - 100 - X
Next
Screen.MousePointer = 0
t = Timer - t
MSChart1.Row = 3
MSChart1.Data = 100 / t
End Sub

```

```

Private Sub Command3_Click()
Unload Me
End Sub

```

```

Private Sub Form_Load()
MSChart1.RowCount = 3
MSChart1.ColumnCount = 1
MSChart1.chartType = VtChChartType2dBar
MSChart1.Row = 1
MSChart1.Data = 228
MSChart1.RowLabel = "P2 233"
MSChart1.Row = 2
MSChart1.Data = 118
MSChart1.RowLabel = "686-120"
MSChart1.Row = 3
MSChart1.Data = 0
MSChart1.RowLabel = "Your CPU"
MSChart2.RowCount = 3
MSChart2.ColumnCount = 1
MSChart2.chartType = VtChChartType2dBar
MSChart2.Row = 1
MSChart2.Data = 27
MSChart2.RowLabel = "P2 233"
MSChart2.Row = 2
MSChart2.Data = 5.85
MSChart2.RowLabel = "686-120"

```

```
MSChart2.Row = 3
MSChart2.Data = 0
MSChart2.RowLabel = "Your CPU"
```

```
End Sub
```

Form9 (Services List):

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

```
Private Sub Form_Load()
End Sub
```

Form10 (Password for Chancing Prices of Orders):

```
Private Sub Command1_Click()
If Text1.Text = "" Then
Load Form6
Form6.Show
Unload Form10
Else
MsgBox ("Wrong keyword or accidentally caps lock open!")
Text1.Text = ""
Form10.Show
End If
End Sub
```

```
Private Sub Form_Load()
End Sub
```

Form11 (Reports):

```
Private Sub Command1_Click()
DataReport2.Show
End Sub
```

```
Private Sub Command2_Click()
DataReport1.Show
End Sub
```

```
Private Sub Command3_Click()
DataReport3.Show
End Sub
```

```
Private Sub Form_Load()
End Sub
```

FrmAbout Form:

```
Option Explicit
' Reg Key Security Options...
Const READ_CONTROL = &H20000
```



```

Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
    KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
    KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL
' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1 ' Unicode nul terminated string
Const REG_DWORD = 4 ' 32-bit number
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA"
    (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal
    samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
    "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal
    lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData
    As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As
    Long

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Me.Caption = "About " & App.Title
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
    App.Revision
    lblTitle.Caption = App.Title
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr
    Dim rc As Long
    Dim SysInfoPath As String
    ' Try To Get System Info Program Path\Name From Registry...

```

```

    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
        ' Try To Get System Info Program Path Only From Registry...
    ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
        ' Validate Existence Of Known 32 Bit File Version
        If (Dir(SysInfoPath & "MSINFO32.EXE") <> "") Then
            SysInfoPath = SysInfoPath & "MSINFO32.EXE"
            ' Error - File Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If
    ' Error - Registry Entry Can Not Be Found...
    Else
        GoTo SysInfoErr
    End If
    Call Shell(SysInfoPath, vbNormalFocus)
    Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As
String, ByRef KeyVal As String) As Boolean
    Dim i As Long                ' Loop Counter
    Dim rc As Long                ' Return Code
    Dim hKey As Long              ' Handle To An Open Registry Key
    Dim hDepth As Long            '
    Dim KeyValType As Long        ' Data Type Of A Registry Key
    Dim tmpVal As String          ' Tempory Storage For A Registry Key
    Value
    Dim KeyValSize As Long        ' Size Of Registry Key Variable
    '-----
    ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
    '-----
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open
Registry Key

    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError    ' Handle Error...
    tmpVal = String$(1024, 0)                        ' Allocate Variable Space
    KeyValSize = 1024                                ' Mark Variable Size
    '-----
    ' Retrieve Registry Key Value...
    '-----
    rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
        KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value
    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError    ' Handle
Errors
    If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then    ' Win95 Adds Null
Terminated String...

```



```

    tmpVal = Left(tmpVal, KeyValSize - 1)      ' Null Found, Extract From String
Else                                           ' WinNT Does NOT Null Terminate String...
    tmpVal = Left(tmpVal, KeyValSize)         ' Null Not Found, Extract String
Only
End If
'-----
' Determine Key Value Type For Conversion...
'-----
Select Case KeyValType                       ' Search Data Types...
Case REG_SZ                                 ' String Registry Key Data Type
    KeyVal = tmpVal                         ' Copy String Value
Case REG_DWORD                             ' Double Word Registry Key Data
Type
    For i = Len(tmpVal) To 1 Step -1        ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value Char. By Char.
    Next
    KeyVal = Format$("&h" + KeyVal)          ' Convert Double Word To String
End Select
GetKeyValue = True                           ' Return Success
rc = RegCloseKey(hKey)                       ' Close Registry Key
Exit Function                                ' Exit
GetKeyError: ' Cleanup After An Error Has Occured...
KeyVal = ""                                  ' Set Return Val To Empty String
GetKeyValue = False                           ' Return Failure
rc = RegCloseKey(hKey)                       ' Close Registry Key
End Function

```

```

Private Sub Label1_Click()
End Sub

```

```

Private Sub Label1_DblClick()
Call Shell("start mailto:ibrahimdokuzer@hotmail.com")
End Sub

```

FrmSplash Form:

Option Explicit

```

Private Sub Form_KeyPress(KeyAscii As Integer)
    Unload Me
End Sub

```

```

Private Sub Frame1_Click()
    Unload Me
End Sub

```

```

Private Sub Timer1_Timer()
    Unload Me
    Form2.Show
End Sub

```


Client Program (Terminal)

This part is not cover duplicate forms, which exist both part of the program.

Form2 (Entering Computer Number and Password Verification):

```
Private Sub Command1_Click()  
If Text1.Text = "" Then  
    Load Form3  
    Form3.Show  
    Unload Form2  
Else  
    MsgBox ("Wrong keyword or accidentally caps lock open!")  
    Text1.Text = ""  
    Form2.Show  
End If  
End Sub
```

```
Private Sub Form_Load()  
Form2.Caption = "Enter Keyword"  
Text1.Text = ""  
compno$ = InputBox("Please Enter computer number!")  
If compno = "" Then  
End  
End If  
ölçüt = "compno = " & compno$ & ""  
Form6.Data2.Recordset.FindFirst ölçüt  
End Sub
```

Form3 (Main Form):

```
Option Explicit  
Private Declare Function Shell_NotifyIcon Lib "shell32.dll" Alias "Shell_NotifyIconA"  
(ByVal dwMessage As Long, lpData As NOTIFYICONDATA) As Long  
Private Type NOTIFYICONDATA  
    cbSize As Long  
    hwnd As Long  
    uID As Long  
    uFlags As Long  
    uCallbackMessage As Long  
    hIcon As Long  
    szTip As String * 64  
End Type  
Dim tray As NOTIFYICONDATA  
Dim answer As Variant  
Dim Y As Integer  
Dim i As Integer  
Dim ElapsedTime As Variant  
Dim starttime As Variant  
Dim EndTime As Variant
```

```

Dim dat As Variant
Private Declare Function GetCurrentProcessId Lib "kernel32" () As Long
Private Declare Function GetCurrentProcess Lib "kernel32" () As Long
Private Declare Function RegisterServiceProcess Lib "kernel32" (ByVal dwProcessID
As Long, ByVal dwType As Long) As Long
Private Const RSP_SIMPLE_SERVICE = 1
Private Const RSP_UNREGISTER_SERVICE = 0

```

```

Private Sub ab_Click()
frmAbout.Show
End Sub

```

```

Private Sub AniGIF1_DblClick()
Call Shell("start mailto:ibrahimdokuzer@hotmail.com")
End Sub

```

```

Private Sub AniGIF1_Step()
End Sub

```

```

Private Sub cal_Click()
Form9.Show
End Sub

```

```

Private Sub clo_Click()
Form10.Show
End Sub

```

```

Private Sub close_Click()
Dim c
c = MsgBox("Do you really want to exit Windows?", vbYesNo, 36)
If c = vbYes Then
c = ExitWindowsex(42, 1)
End If
End Sub

```

```

Private Sub Command1_Click()
'for below statement unlocks the enablety of this programs'
drwatson.Enabled = True
dialer.Enabled = True
media.Enabled = True
note.Enabled = True
paint.Enabled = True
player.Enabled = True
scan.Enabled = True
volume.Enabled = True
'-----'
Timer1.Enabled = True
Command5.Enabled = True
Command1.Visible = False
Command4.Enabled = True

```

```
timer.Enabled = False
tx.Enabled = True
st.Enabled = True
```

```
Command2.Enabled = True
'Establish start time and start timer control
starttime = Now
Text1.Text = Format(starttime, "hh:mm:ss")
StatusBar1.Panels(1).Text = "Timer is active"
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
End Sub
```

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Start"
End Sub
```

```
Private Sub Command2_Click()
'for below statement locks again the enablety of this programs'
drwatson.Enabled = False
dialer.Enabled = False
media.Enabled = False
note.Enabled = False
paint.Enabled = False
player.Enabled = False
scan.Enabled = False
volume.Enabled = False
'-----'
Dim selection As Variant
'stop button pressed
'stop button disabled
'Enabled start and exit buttons
selection = MsgBox("You cannot start with same account again if you stop now! Are
you sure want to stop and leave?", 52, "Stop Timer")
If selection = 6 Then
Command4.Enabled = False
tx.Enabled = False
Command2.Enabled = False
Command1.Visible = True
Command5.Enabled = False
EndTime = Now
dat = Date
Text5.Text = dat
ElapsedTime = EndTime - starttime
Text2.Text = Format(EndTime, "hh:mm:ss")
Text3.Text = Format(ElapsedTime, "hh:mm:ss")
```



```

Timer1.Enabled = False
Text4.Text = i
i = 0
Form6.Data1.Recordset.AddNew
Form6.Text4.Text = Text4.Text
Form6.Text2.Text = Text1.Text
Form6.Text1.Text = Y
Form6.Text3.Text = Text2.Text
Form6.Text7.Text = Text5.Text
timer.Enabled = True
st.Enabled = False
End If
StatusBar1.Panels(1).Text = "Timer is not active"
End Sub

```

```

Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Stop"
End Sub

```

```

Private Sub Command3_Click()
Form5.Show
End Sub

```

```

Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Date/Time"
End Sub

```

```

Private Sub Command4_Click()
Dim minimized As VbAppWinStyle
Form3.WindowState = 1 - minimized
End Sub

```

```

Private Sub Command4_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Windows Desktop"
End Sub

```

```

Private Sub Command5_Click()
Form7.Show
End Sub

```

```

Private Sub Command5_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Timer"
End Sub

```

```

Private Sub Command6_Click()
Form1.Show
End Sub

```

```

Private Sub Command6_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Benchmark"
End Sub

```

```

Private Sub Command7_Click()
Load Form4
Form4.Show
End Sub

```

```

Private Sub Command7_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
StatusBar1.Panels(4).Style = sbrText
StatusBar1.Panels(4).AutoSize = sbrContents
StatusBar1.Panels(4).Text = "Exit"
End Sub

```

```

Private Sub dialer_Click()
Dim p
p = Shell("c:\windows\Dialer.exe", 4)
End Sub

```

```

Private Sub drwatson_Click()
Dim k
k = Shell("c:\windows\Drwatson.exe", 4)
End Sub

```

```

Private Sub Form_Load()
Show
tray.cbSize = Len(tray)
tray.hwnd = Form3.hwnd
tray.szTip = "Net@Cafe" + Chr(0)
tray.uCallbackMessage = &H200
tray.uFlags = 7
tray.hIcon = Form3.Icon
Shell_NotifyIcon 0, tray
App.TaskVisible = False

```

```

Y = InputBox("Please Enter computer number again!,But number must be the same
with number you entered first!")
Text4.Text = ""
Form3.Caption = "cafe@net1.0(Com400)"
Command5.Enabled = False
Command2.Enabled = False
tx.Enabled = False
st.Enabled = False
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
'for below statement locks the enablety of this programs'
drwatson.Enabled = False
dialer.Enabled = False
media.Enabled = False
note.Enabled = False
paint.Enabled = False
player.Enabled = False
scan.Enabled = False
volume.Enabled = False
'-----'
Command4.Enabled = False
MsgBox ("Please press Start button to start to using this terminal computer!")
If App.PrevInstance = True Then
MsgBox ("Program is already running")
End
End If
Dim id As Long
Dim X As Long
id = GetCurrentProcessId()
X = RegisterServiceProcess(id, RSP_SIMPLE_SERVICE)
End Sub

Private Sub stop_Click(Index As Integer)
End Sub

Private Sub media_Click()
Dim t
t = Shell("c:\windows\Mplayer.exe", 4)
End Sub

Private Sub note_Click()
Dim s
s = Shell("c:\windows\notepad.exe", 4)
End Sub

Private Sub paint_Click()
Dim l
l = Shell("c:\windows\pbrush.exe", 4)
End Sub

```



```
Private Sub player_Click()
```

```
Dim z
```

```
z = Shell("c:\windows\cdplayer.exe", 4)
```

```
End Sub
```

```
Private Sub scan_Click()
```

```
Dim w
```

```
w = Shell("c:\windows\scandiskw.exe", 4)
```

```
End Sub
```

```
Private Sub st_Click()
```

```
Dim selection As Variant
```

```
'stop button pressed
```

```
'stop button disabled
```

```
'Enabled start and exit buttons
```

```
selection = MsgBox("You cannot start with same account again if you stop now! Are  
you sure want to stop and leave?", 52, "Stop Timer")
```

```
If selection = 6 Then
```

```
Command4.Enabled = False
```

```
tx.Enabled = False
```

```
Command2.Enabled = False
```

```
Command1.Enabled = True
```

```
Command5.Enabled = False
```

```
EndTime = Now
```

```
ElapsedTime = EndTime - starttime
```

```
Text2.Text = Format(EndTime, "hh:mm:ss")
```

```
Text3.Text = Format(ElapsedTime, "hh:mm:ss")
```

```
Timer1.Enabled = False
```

```
Text4.Text = i
```

```
i = 0
```

```
Form6.Data1.Recordset.AddNew
```

```
Form6.Text4.Text = Text4.Text
```

```
Form6.Text2.Text = Text1.Text
```

```
Form6.Text1.Text = Y
```

```
Form6.Text3.Text = Text2.Text
```

```
timer.Enabled = True
```

```
st.Enabled = False
```

```
End If
```

```
StatusBar1.Panels(1).Text = "Timer is not active"
```

```
End Sub
```

```
Private Sub tech_Click()
```

```
MsgBox ("You can send e-mail to us for your technical  
problems.ibrahindokuzer@hotmail.com")
```

```
End Sub
```

```
Private Sub timer_Click()
```

```
Timer1.Enabled = True
```

```
Command5.Enabled = True
```

```

Command1.Enabled = False
Command4.Enabled = True
timer.Enabled = False
tx.Enabled = True
st.Enabled = True
Command1.Enabled = False
Command2.Enabled = True
'Establish start time and start timer control
starttime = Now
Text1.Text = Format(starttime, "hh:mm:ss")
StatusBar1.Panels(1).Text = "Timer is active"
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
End Sub

```

```

Private Sub Timer1_Timer()
i = i + 1
End Sub

```

```

Private Sub Timer2_Timer()
Form6.Data2.UpdateRecord
Form6.Data1.UpdateRecord
If Form6.Text6.Text = 0 Then
Command1.Enabled = False
Command2.Enabled = False
timer.Enabled = False
st.Enabled = False
Command4.Enabled = False
Else
Command1.Enabled = True
End If
End Sub

```

```

Private Sub tx_Click()
Form7.Show
End Sub

```

```

Private Sub volume_Click()
Dim g
g = Shell("c:\windows\sndvol32.exe", 4)
End Sub

```

Form5 (Date/Time):

```

Private Sub Command1_Click()
Unload Form5
End Sub

```

```

Private Sub Form_Load()
End Sub

```

```

Private Sub Timer1_Timer()
Dim today As Variant
today = Now
Label1.Caption = Format(today, "dddd")
Label2.Caption = Format(today, "mmmm")
Label3.Caption = Format(today, "yyyy")
Label4.Caption = Format(today, "d")
Label5.Caption = Format(today, "h:mm:ss ampm")
End Sub

```

Form6 (Secret form to get signal from server just includes text boxes):

```

Private Sub Form_Load()
Form6.Refresh
End Sub

```

Form7 (Alarm):

```

Private Sub Command1_Click()
Timer2.Enabled = False
Timer1.Enabled = False
Timer3.Enabled = False
End Sub

```

```

Private Sub Command2_Click()
Unload Form7
End Sub

```

```

Private Sub Form_Load()
Option1.Value = False

```

```

Dim k As Integer
Timer2.Enabled = False
Timer1.Enabled = False
Timer3.Enabled = False
End Sub

```

```

Private Sub Option1_Click()
If Option1.Value = True Then
Timer1.Enabled = True
Timer2.Enabled = False
Timer3.Enabled = False
End If
End Sub

```

```

Private Sub Option2_Click()
If Option2.Value = True Then
Timer2.Enabled = True
Timer1.Enabled = False
Timer3.Enabled = False
End If

```


End Sub

```
Private Sub Option3_Click()  
If Option3.Value = True Then  
Timer3.Enabled = True  
Timer1.Enabled = False  
Timer2.Enabled = False  
End If  
End Sub
```

```
Private Sub Timer1_Timer()  
k = k + 1  
If k = 60 Then  
MsgBox ("Time is up")  
End If  
End Sub
```

```
Private Sub Timer2_Timer()  
k = k + 1  
If k = 120 Then  
MsgBox ("Time is up")  
End If  
End Sub
```

```
Private Sub Timer3_Timer()  
k = k + 1  
If k = 180 Then  
MsgBox ("Time is up")  
End If  
End Sub
```

Form9 (Calender):

```
Private Sub Form_Load()  
End Sub
```

```
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)  
End Sub
```

Form10 (Clock):

```
Private Sub Form_Load()  
Width = ScaleHeight  
Scale (-20, 20)-(20, -20)  
Timer1.Interval = 1000  
AutoRedraw = True  
Circle (0, 0), 20  
DrawMode = 7 'xor  
End Sub
```

```
Private Sub Timer1_Timer()  
Dim aci, saniye, dakika, saat
```

```

Static sx, sy, dx, dy, stx, sty
Caption = Time
DrawWidth = 1
Line (0, 0)-(sx, sy), QBColor(10)
saniye = Second(Time)
aci = saniye * 6 + 90 'her bir saniye 6 derecedir. 360 derece/60
sx = 20 * Cos(aci * 3.1415 / 180)
sy = 20 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(sx, sy), QBColor(10)
DrawWidth = 3
Line (0, 0)-(-dx, -dy),
dakika = Minute(Time)
aci = dakika * 6 + 90
dx = 20 * Cos(aci * 3.1415 / 180)
dy = 20 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(-dx, -dy), QBColor(11)
DrawWidth = 3
Line (0, 0)-(-stx, -sty), QBColor(12)
saat = Hour(Time)
aci = saat * 30 + 90
stx = 12 * Cos(aci * 3.1415 / 180)
sty = 12 * Sin(aci * 3.1415 / 180)
Line (0, 0)-(-stx, -sty), QBColor(12)
End Sub

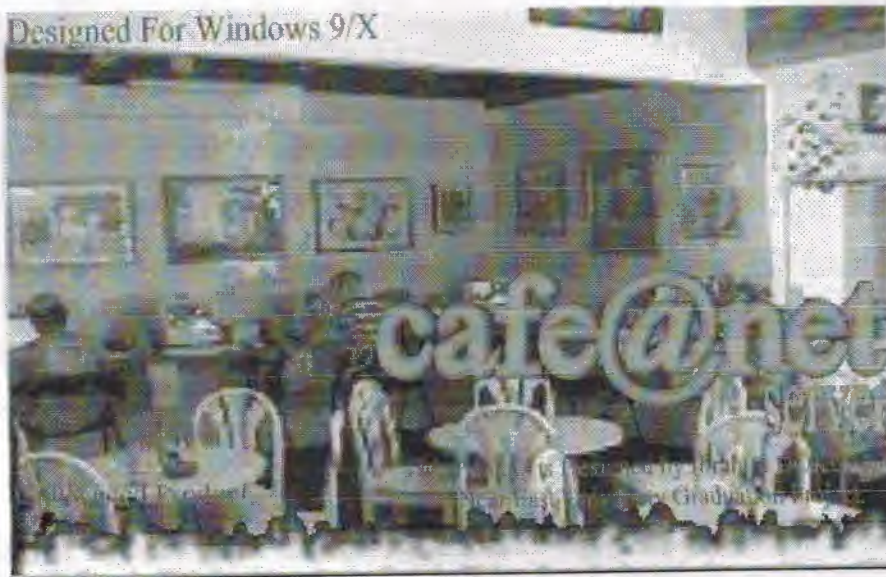
```

(Client part of the program include splash and about form similar to server program for this reason this part does not cover these forms)

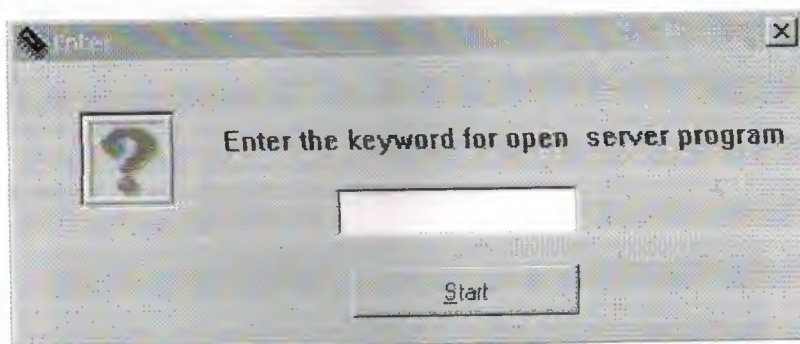
APPENDIX B

This part covers the screen layout of forms, reports and specific parts of the both programs (Server and Client).

Main Program (server)



Picture1. Splash Form of Server part of the program.



Picture2. Verification of password form for open server program (Form2).



Picture3. Main form of the server program (Form3)

Picture4. Calculating the charges of user form of the server (Form4).

Picture5. Input box to accept computer number for find specific computer.

Item	Selected Value
Computer	1
Coffee	4
Tea	3
Coke	1
Ayran	3
Caphucino	2
Espresso	3
Beer	4
Cookie	2
Pizza	1
Toast	3
Hamburger	1

Buttons: Find Computer, Clear Orders, Ok

Logo: cafe@net

Picture6.Order list of users form (form5).

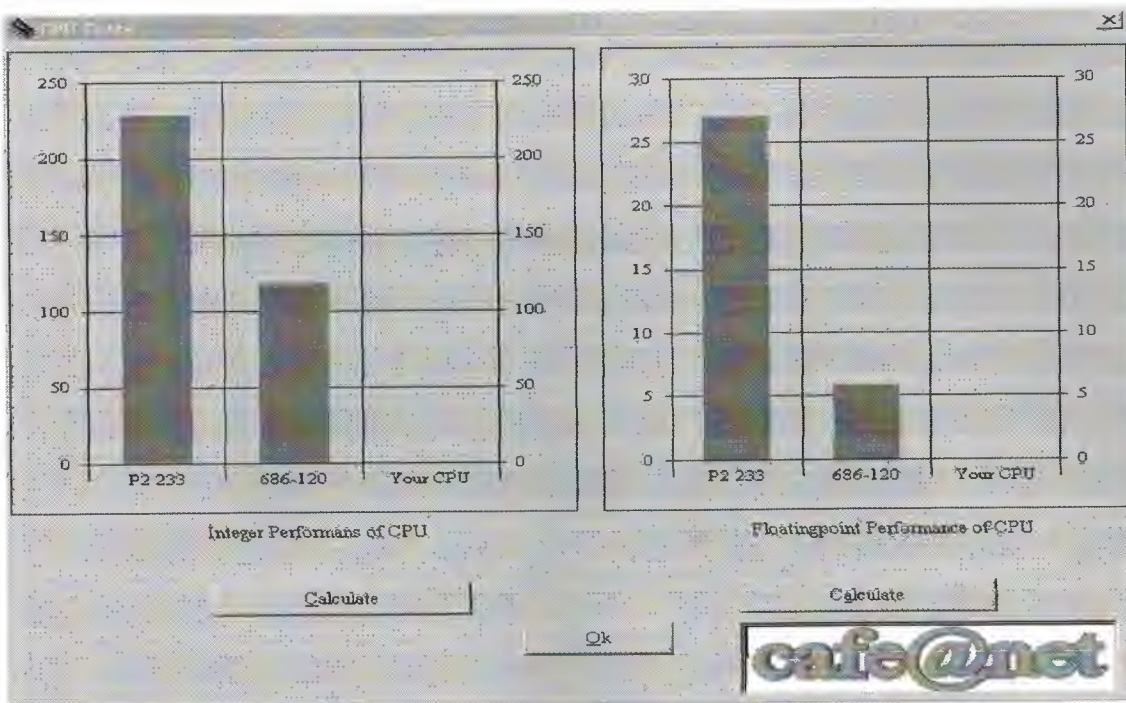
Item	Unit Cost
Coffee	100000
Tea	500000
Coke	1000000
Ayran	600000
Caphucino	750000
Espresso	3000000
Beer	1500000
Cookie	250000
Pizza	4000000
Toast	750000
Hamburger	2000000

Internet Price: Time Price 10000

Button: Ok

Logo: cafe@net

Picture7.The unit cost of the orders list (Form6).

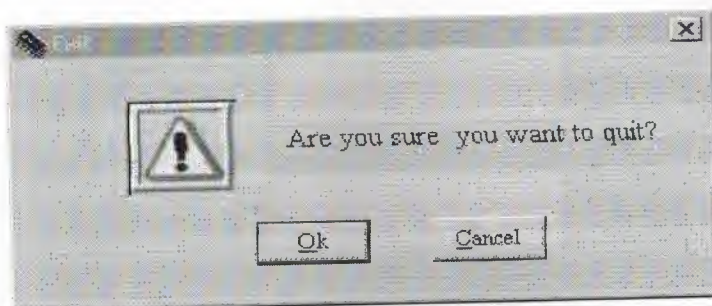


Picture8. The Form of the CPU Test process (Form8).

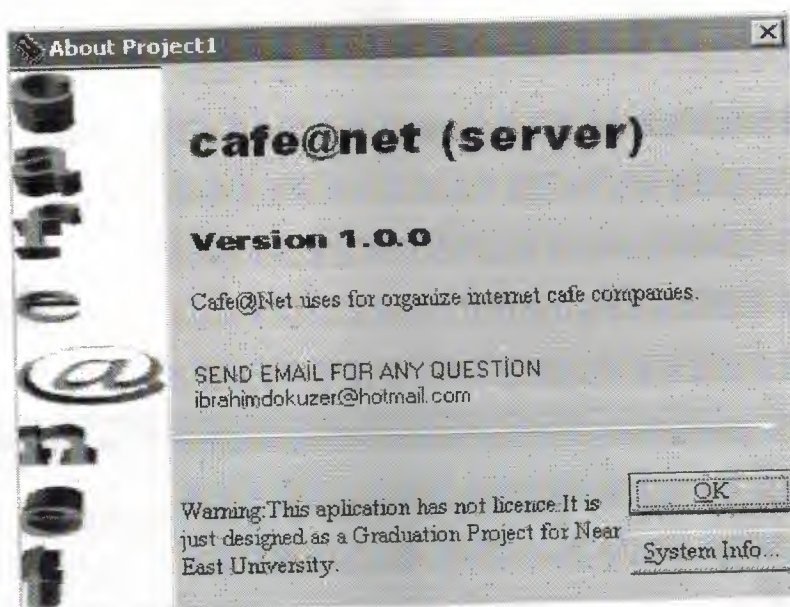
The screenshot shows a window titled 'Services List' containing a table with columns: conculamo, starttime, stoclme, elapsedtime, charact, conculamo, coke, ayran, tea, coffee, hamburger, icaat, cashucino, and exit. The table lists 31 rows of data, including start and stop times and various service counts. An 'Ok' button and a 'cafe@net' logo are at the bottom.

conculamo	starttime	stoclme	elapsedtime	charact	conculamo	coke	ayran	tea	coffee	hamburger	icaat	cashucino	exit
1	21:30:12	21:42:27	0	13900000	1	1	3	3	4	1	3	2	
1	22:33:25	22:33:30	0	17650000	2	3	0	0	0	0	0	0	
1	05:15:09	05:15:33	0		3	0	5	0	2	0	0	0	
1	05:22:44	05:23:12	0	12200000	4	0	0	0	0	0	0	0	
1	05:32:38	05:33:05	0		5	0	0	0	0	0	0	0	
1	05:39:02	05:39:37	1		6	0	0	0	0	0	0	0	
1	05:41:54	05:42:06	0		7	0	0	0	0	0	0	0	
1	05:46:05	05:46:41	0		8	0	0	0	0	0	0	0	
1	05:46:49	05:46:57	0		9	0	0	0	0	0	0	0	
1	05:47:47	05:47:51	0	37200000	10	0	0	0	0	0	0	0	
1	06:00:15	06:00:48	0		11	0	0	0	0	0	0	0	
1	06:06:24	06:06:56	0	34200000	12	0	0	0	0	0	0	0	
1	06:11:52	06:12:25	2	17650000	13	0	0	0	0	0	0	0	
1	06:47:14	06:47:17	0		14	0	0	0	0	0	0	0	
1	17:03:37	17:03:42	0		15	0	0	0	0	0	0	0	
1	17:06:22	17:06:27	0		16	0	0	0	0	0	0	0	
1	17:09:27	17:09:31	0		17	0	0	0	0	0	0	0	
1	02:34:23	02:35:07	0		18	0	0	0	0	0	0	0	
1	11:57:00	11:59:10	2		19	0	0	0	0	0	0	0	
3	18:11:04	18:11:10	0		20	0	0	0	0	0	0	0	
					21	0	0	0	0	0	0	0	
					22	0	0	0	0	0	0	0	
					23	0	0	0	0	0	0	0	
					24	0	0	0	0	0	0	0	
					25	0	0	0	0	0	0	0	
					26	0	0	0	0	0	0	0	
					27	0	0	0	0	0	0	0	
					28	0	0	0	0	0	0	0	
					29	0	0	0	0	0	0	0	
					30	0	0	0	0	0	0	0	
					31	0	0	0	0	0	0	0	

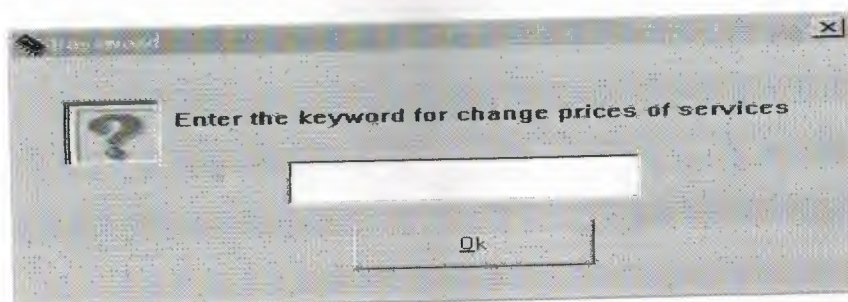
Picture9. Service and time data's of users (Form9)



Picture10.Validation to quit (Form7).



Picture11.About Form of server program (About Form).



Picture12.Password verification for enters unit prices of services form (Form10).

Open/Close Permission

Table Close/open 1

Computer NO 1

Computer No Opening Table Closing Table

cafe@net

Picture13. The Form of the permission situation control (Form1).

Reports

Cafe@net REPORTS

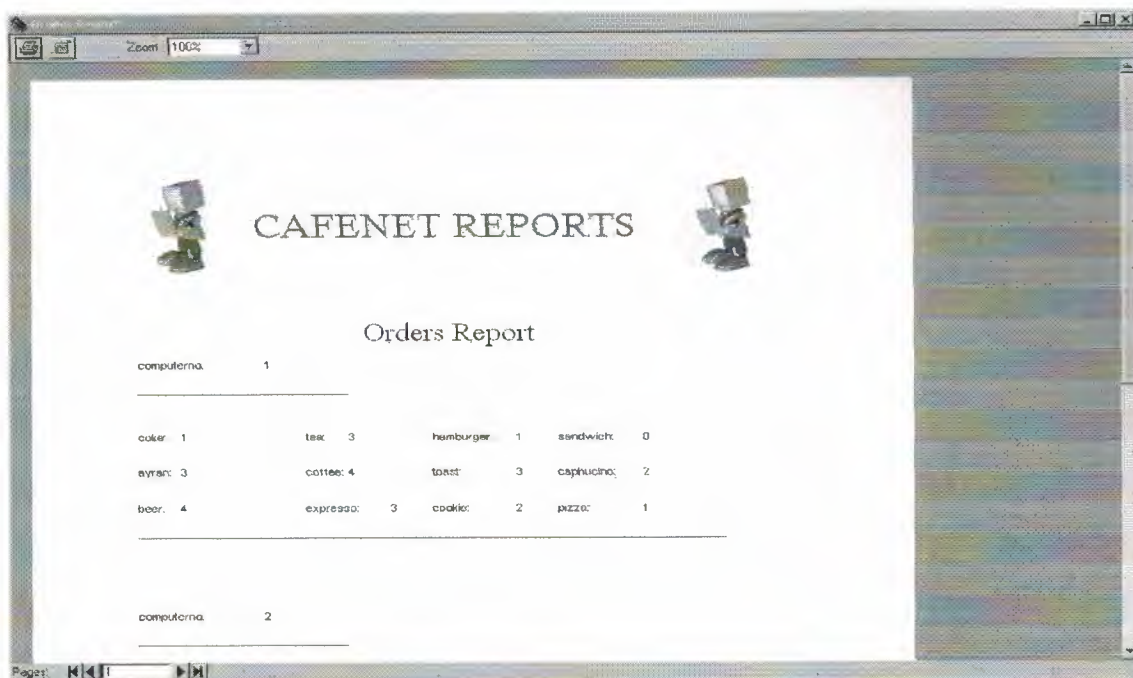
Orders Report

Time and Charge Report

Prices Report

cafe@net

Picture14. Main menu of reports form (Form12)



CAFENET REPORTS

Orders Report

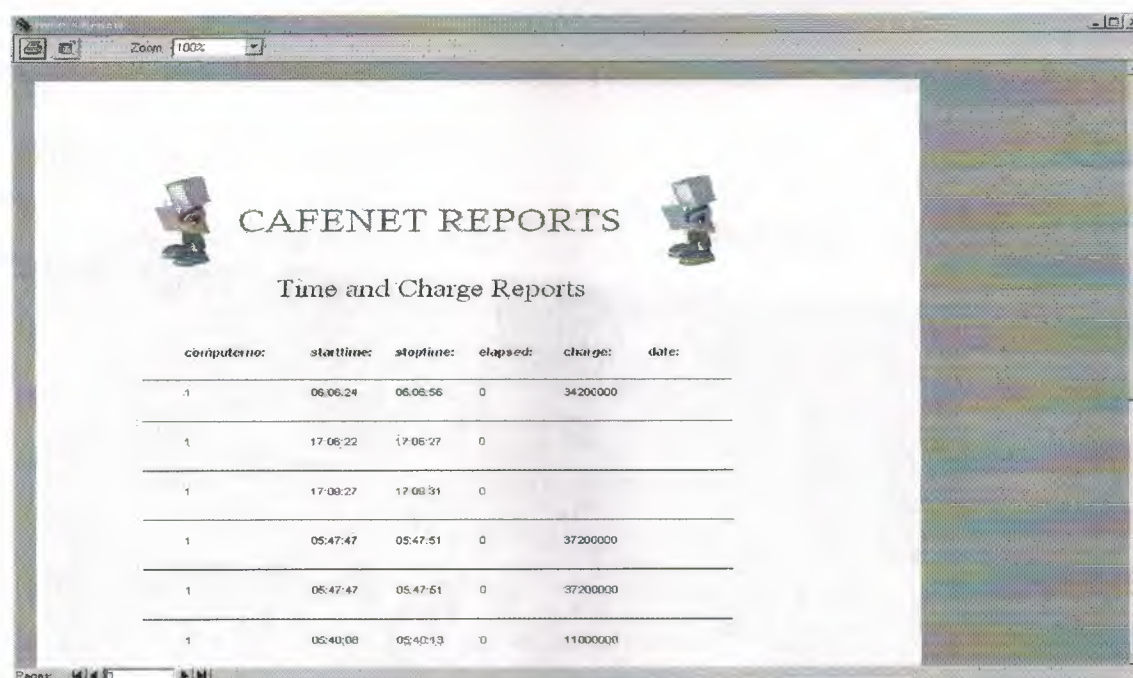
computerno: 1

coke:	1	tea:	3	hamburger:	1	sandwich:	0
ayran:	3	coffee:	4	toast:	3	capuccino:	2
beer:	4	expresso:	3	cookie:	2	pizza:	1

computerno: 2

Page: 1/1

Picture15. The appearance of orders of users report.



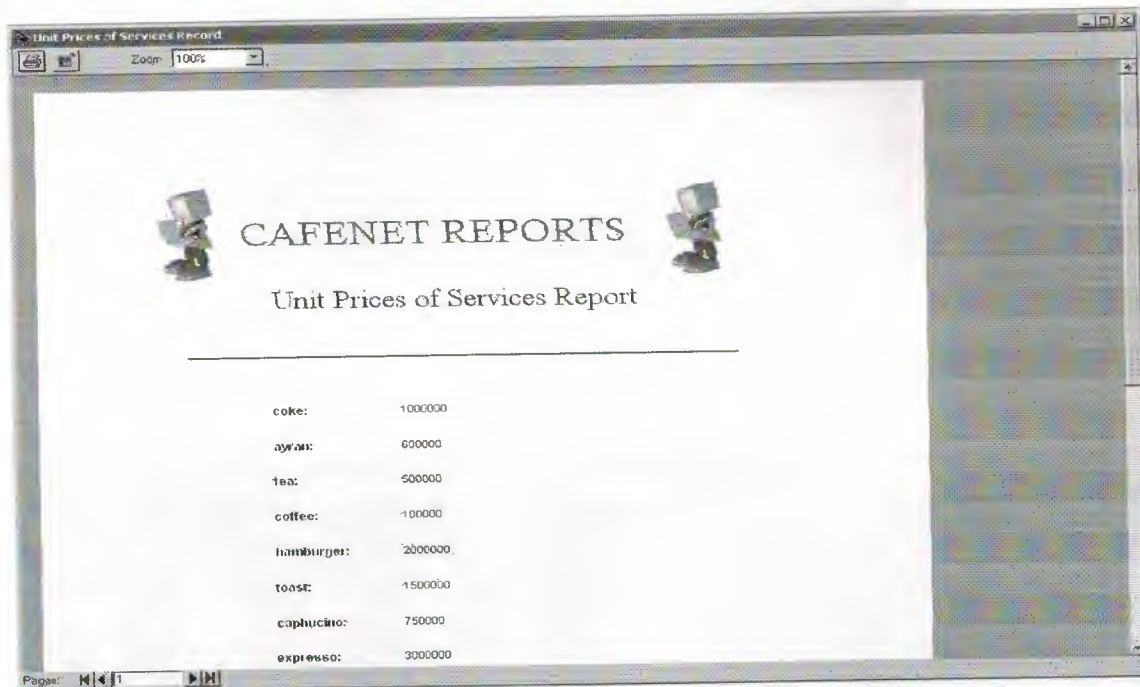
CAFENET REPORTS

Time and Charge Reports

computerno:	starttime:	stopline:	elapsed:	charge:	date:
1	06:06:24	06:06:56	0	34200000	
1	17:06:22	17:06:27	0		
1	17:08:27	17:08:31	0		
1	05:47:47	05:47:51	0	37200000	
1	05:47:47	05:47:51	0	37200000	
1	05:40:08	05:40:13	0	11000000	

Page: 1/1

Picture16. The appearance of time and charge information of users report.



Picture17. The appearance of Unit Prices of Services report.

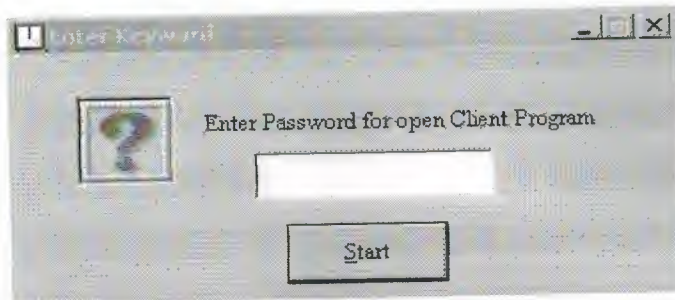


Picture18. The appearance of the Technical Support message box.

Client Program (Terminal)



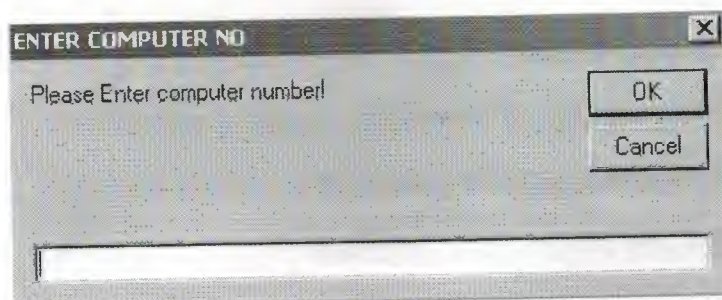
Picture19. Splash Form of Client part of the program.



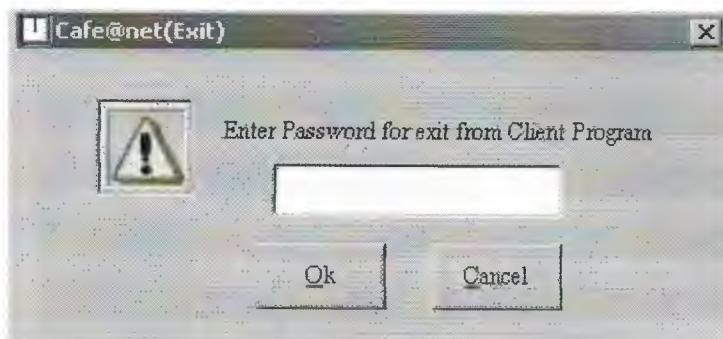
Picture20. Verification of password form for open client program (Form2).



Picture21. Main form of the client program (Form3).



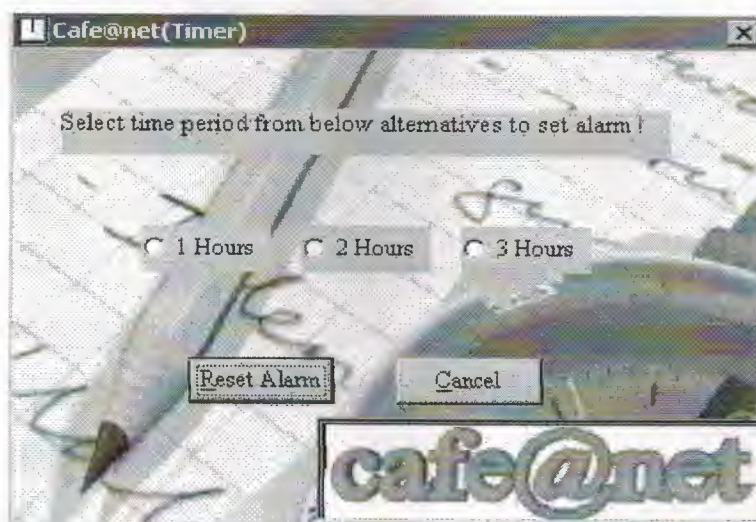
Picture22. Computer number message box to reach database file to initialize session.



Picture23. Verification of password form for exit client program (Form4).



Picture24. Form of the date and time application (Form5).



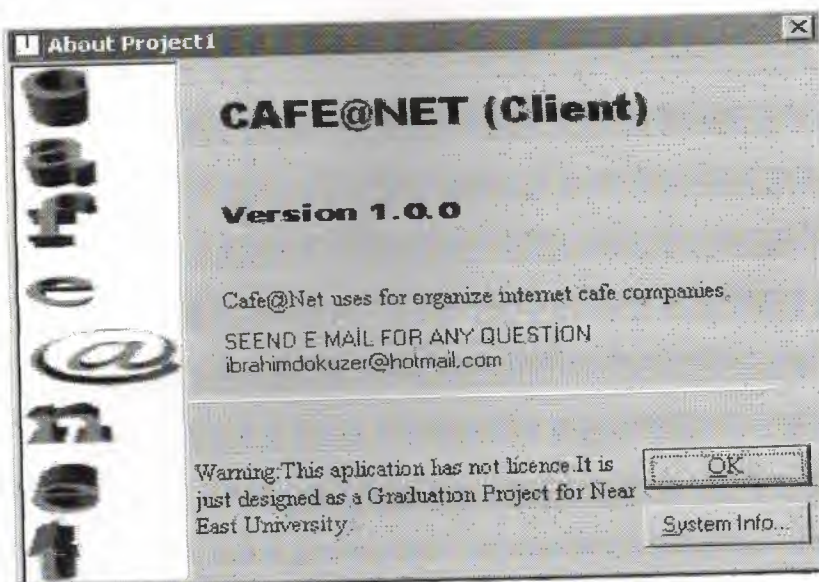
Picture25. Form of the alarm that warns user (Form7).



Picture26. Form of the basic calendar application (Form9).



Picture27. Form of the clock application (Form10).



Picture28. About Form of client program (About Form).