NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

INTERNET CAFÉ RESERVATION PROGRAM

Graduation Project
COM- 400

Student: Tolga Dokuzer (950714)

Supervisor: Miss. Besime Erin

Nicosia - 2002

ACKNOWLEDGMENT

First of all I am happy to complete the task which I had given with blessing of God and also I am grateful to my brother and my family who have, supported me, advised me, taught me and who have always encouraged me follow my ambitions.

I wish to thank my supervisor, Miss Besime Erin, for intellectual support, encouragement, and enthusiasm, which made this project possible.

Also thank all teachers for their advice and support. The comment of these teachers enabled me to present this project successfully.

ABSTRACT

The reservation or automation in some of the business execution is very important level for a productive solution in market. The producers pay much attention on that part of the production so it needs a well-designed structure to operate efficiently.

Internet café concept is also an important commercial area in these days, because they serve a technological product called Internet.
Internet refers the both the project and the prototype network system that researchers built. Today it is a global network that connects different geographical parts of the world.
Internet connection requires a computer and some networking tools with service. After supplying these requirements we are ready to control our Internet café.

To organize a good atomized Internet café we need to use also good operating automation program loaded on computers. Reservation is needed for café because the customers who comes to the Internet café for using Internet wants well organized structure of services and payment system.

The main problem is controlling user's time information (service start and stop times) and other services used with Internet service. For this reason using client-server architecture is more conventional to solve problem but a complete structure is used for dynamic programs. We can use a static application as in this project to solve this problem also again using client-server architecture.

Internet user or customer must understand visualization of program to satisfy about services he or she used. For this problem a program must reflect all needed information to the user. We can use a graphical interfaced program for creating user-friendly menus or displays.

Another problem is the security part of the program; some of the unwanted actions for the program or using security parts in applications must eliminate escapes.

Any process or extra functions inside the program must not effect other running applications and must not slow down the system. Automation program should use system resources at minimum because it runs until the user stops using services.

TABLE OF CONTENTS

INTRODUCTION

The Internet Café's are the new commercial areas that serve to their customers Internet connection including café concept. These places have started take place early 1990's with a café style, because they also have tables with the computers on it and customer can have café services. As the Internet grows the thrill of foreign communication has faded over the years. When an individual first encounters the Internet, they usually use a specific service. Peoples realized many useful services and more specific are designed with Internet.

Nowadays, the Internet seems as a necessity for many computer owners. Internet had a global popularity by making information reachable and usable from any certain computer. After the development of communication devices and techniques, Internet became faster and reliable data source everyone can access. Also many computers still not connected to this global network because of some requirements of its as hardware and lines. The computer users who want to connect to the Internet need an extra hardware called modem and a service provider for a dial-up connection. All these requirements make Internet connectivity a little difficult for computer users. Also users may not have a chance to use their computer for any reason.

Internet café's solves problems by bringing today's personal computers to the public with Internet connections. With number of connected computer's users can deal with Internet and even can enjoy their time by drinking or eating as in a café. For this reason Internet café's still has a growing popularity as in every commercial area the executives of this business are in a competition for giving a high quality service. As a point of view Internet is a product to serve for the companies and most problem is reservation of customers at different times to have productive results.
Many Internet café's uses ticketing method as a solution of this problem by just writing starting time of service of customers. But this method needs operational costs as more employee and as preparing printed tickets for orders and Internet cost. Also have many security problems as lost services and escapes. In time the programmers found a way to atomize or coordinate that problem by preparing some programs. In this project I aimed to prepare a program that atomize that reservation and billing problems for an Internet café.

The programs are prepared by using a visual interfaced language called Visual Basic that uses Windows graphical user interface support (GUI). Visual Basic is capable to prepare Windows based programs using Windows functions. Mostly Visual Basic version 6.0 brings many developing tools to create powerful applications. These programs will organize this network inside Internet café with Internet connection.

1. BACKGROUND

Since it has introduced the Visual Basic version 6.0 brought an enjoyable programming platform to the programmers. Many programmers have started to choose this language because of their tools that bringing easiness and high-level visual Windows effects to their applications. Windows operating system altered all high-level languages by saving from monotony of the programming and renewing itself at each version with better tools.

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

B (Beginner's)
A (All-Purpose)
S (Symbolic)
I (Instruction)
C (Code)

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. You can see many modern applications prepared in Visual Basic at many platforms that use graphical user interfaces.

The Visual Basic version 6.0 also uses common Visual Basic tools, wizards, references, and components. You can see the design window of the Visual Basic version 6.0 in Picture 1.1.

Visual Basic is a tool that allows you to develop Windows (Graphic User Interface - GUI) applications and the applications have a familiar appearance to the user.

Visual Basic is event-driven; meaning code remains idle until called upon to respond to some event (button pressing, menu selection...). It is governed by an event processor. Nothing happens until an event is detected. Once an event is detected, the code corresponding to that event (event procedure) is executed. Program control is then returned to the event processor

Some Features of Visual Basic:

- Full set of objects - you 'draw' the application
- Lots of icons and pictures for your use
- Response to mouse and keyboard actions
- Clipboard and printer access
- Full array of mathematical, string handling, and graphics functions
- Can handle fixed and dynamic variable and control arrays
- Sequential and random access file support
- Useful debugger and error-handling facilities
- Powerful database access tools
- ActiveX support
- Setup Wizard makes distributing your applications simple

To design professional projects a programmer must have an idea about Visual Basic controls (OCX Files), database management and using Windows controls in Visual Basic (API).
As in every object oriented language programmers can create their classes and objects. In Visual Basic version 6.0 with some new control elements programmers do not need write long codes. When Visual Basic starts the toolbox and standard Visual Basic elements appears. Inside the chapters you can find information about used tools and controls in programs. The programs will provide features for operators and users. Programs do not use complete client-server architecture to operate and minimize problems. Automation applications includes user-friendly menus and each part of the have maximum security especially client side of programs and they operates fast.

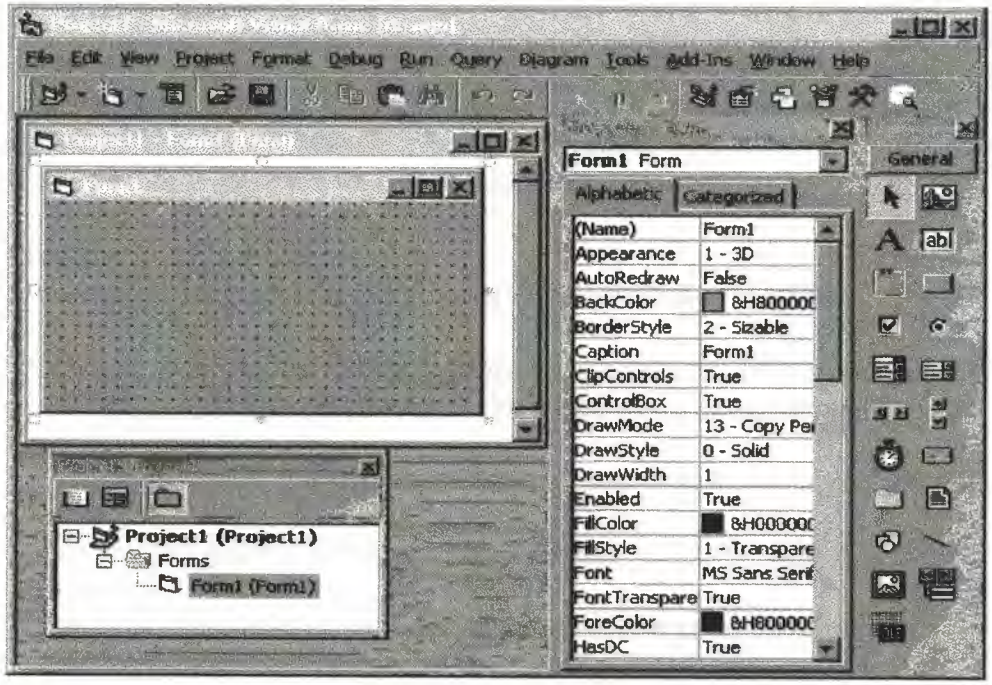


Figure 1.1. The Visual Basic 6.0 design window that includes design tools

2. SOFTWARE SPECIFICATIONS

The programs are prepared for Internet café's at every capacity and it makes operators work easier, solve problems and minimize lost. User and operator would be able to control complete hall. For analyzing general structure of program first we need to plan a software feasibility study.
Firstly organize client-server architecture for an efficient database relation.

- Café station: main program (server), which is operated by café administrator.
- User-friendly interface ensures quick and easy operation even by novice computer uses fully customizable through options, locks workstations.
- Café terminal: second small program that resides on remote computer, displays time notifications, start and stops the timer.

The idea behind client-server relation is quite simple some computers over a network offer some conditions or services that others can use. A client program can contact such a server or its related files and requests required information.
Other serves required information or gives permissions. A user runs a client program and program requests information from the remote computer or its related data files.

2.1.General Properties of Client Software

Software that runs on terminal computers for sending user's start and stop times to the main program is our client program. Client program can also have some extra applications for the users as an alarm or clock.
Client program must satisfy security because deals with users.

2.2.General Properties of Main Software

Software that runs on main computer of the café for taking times from client, calculates charges for Internet and services and see which computer is busy. Server program has an administrative because it can reflect all users start and stop times, can change hourly rates and clear accounts. All the charges must be calculated perfectly with minimum time slices. Operators must not be able to replace any controls.

2.3.Advantages of programs

We can determine some of the advantages of these reservation programs:

1. Operator is able to see all terminals start and stop times from main computer.
2. Start and stop time and services can be added to the same charge many times by realizing them from server program.
3. By management of programs company can save their time and prevent any loses or escapes.
4. Also the user can see his/her start and stop times together with their elapsed times
5. Users can use a timer or an alarm to organize their connection times.
6. Users can use common programs from applications menu.

Programs also do not take up almost any system resources; do not slow down local network nor Internet. Both programs can install themselves automatically.

3. STRUCTURE OF THE PROGRAM

The program separated into two parts as explained before and they are called client and server programs. They are designed for controlling users start and stop times from a main computer and calculate all charges. The operator or operators in Internet café must be able to use programs easily. So they must supported with user-friendly menus.
Server takes information (start and stop times) from client program.
Other part is the client part and it must be installed on each computer (terminals) in Internet café. User can start using Internet service after taking one of permitted computer number from café operator and pushing start button at the same numbered computer in café. Program then allows using Windows desktop by another button or user can reach some known applications on program. Client program also has an alarm and clock; if the user sets alarm then program warns the user with a message. Both programs have a Benchmark program to test the actual performances of the computers.

3.1.Main Program (Server Program)

Main program is able to realize some manipulations on user's start and stop times. To do so, the main program must get the times from the database that sent by client software. Main program can give permissions for terminal computers for security also has security protection for operator entry.

3.1.1.Password Protection for Main Program

To enter and use main program operator has to enter a password. This process is to defend program from attacks. You can see password verification process in Figure 3.1.

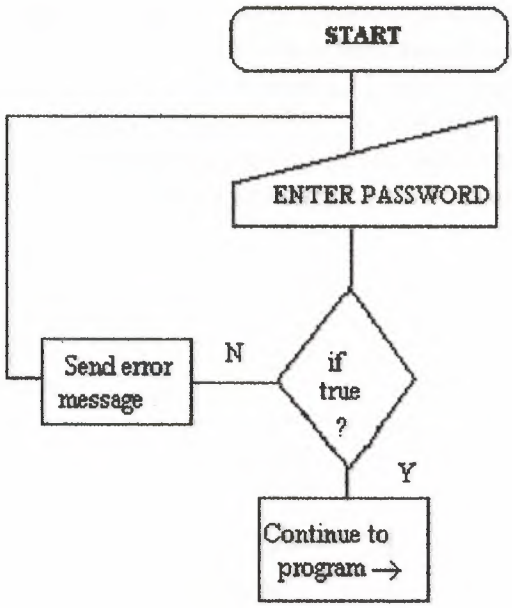


Figure 3.1. The flow of the password verification process

For the password verification process we use some simple Visual Basic codes as follows:

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

```
Private Sub Form_Load()
Text1.Text = ""
End Sub
```

Also in the form we use some other components as labels and picture boxes that you can realize from working forms of Visual Basic project. Each Sub function determines a beginning of object code. Above code checks the contents of text object and if it is correct it loads another form else sends a message to the user in a message box.

3.1.2. Giving Permission to the Client

This process is done for satisfying security of starting of services for a certain user. If the permission state is 1 client's start button is visible and user can start using Internet but if 0 user is not able to start and use Internet. Permission can be giving by just entering computer number and state of permission from main program's main form. The state of certain computer is sent to the database and client program checks the state of computer continuously. When it is adjusted as 1 from main computer the client program realizes that and makes buttons visible. If it is adjusted as 1 from main program user cannot 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.

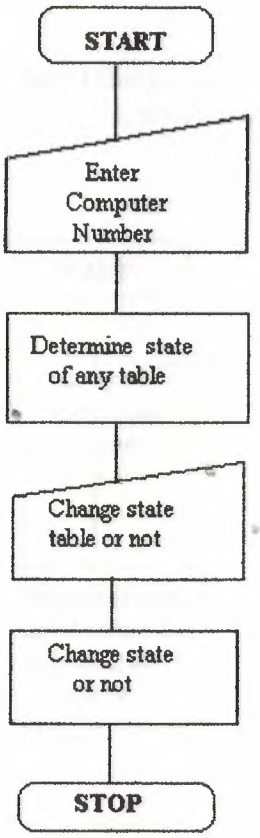


Figure 3.2. The flow of the table permissions

The program codes of this process which written in a form of the project are below:

```
Private Sub Command4_Click()
compno$ = InputBox("enter the table no :")
ölçüt = "compno =" & compno$ & ""
Data1.Recordset.FindFirst ölçüt
End Sub

Private Sub Command5_Click()
Text1.Text = 1
Data1.UpdateRecord
End Sub

Private Sub Command6_Click()
Text1.Text = 0
Data1.UpdateRecord
End Sub

Private Sub Form_Load()
End Sub
```

Here, three command buttons are used to get table or computer number and change state of tables. Every command deals with a database file checks some fields and makes changes in these fields by commands.

3.1.3. Changing Prices

We can create a database file to control this process. The operator can change Internet charge rates or other service prices (foods, drinks) and all changes will be recorded in that database file.

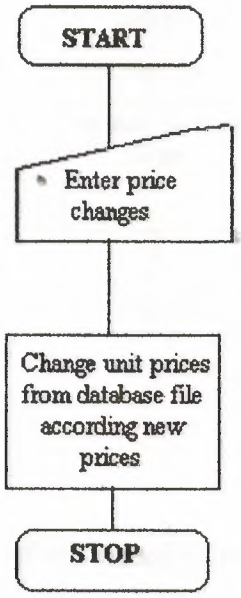


Figure 3.3. The flow of the changing prices process

This part of the project does not requires codes because of some useful tools, some text areas in a form is connected to the database file and the prices which written in that text areas or boxes has been sent to that file.
If the operator wants to change service's prices he/she will just enter new prices on the textboxes. The database file contains only one field and whenever operator writes new prices, the new prices have been sorted on that field.

3.1.4. Recording Orders and Calculate Charges
The service orders of the customer can be organized by adding the ordered services on a database file from the main program. To realize that work again possible by using database files. One database for the store amount of orders and another one in same database with price changing process. The user's orders have been recorded according to their computer numbers. Charge of services calculated by multiplying unit prices by amount of orders.

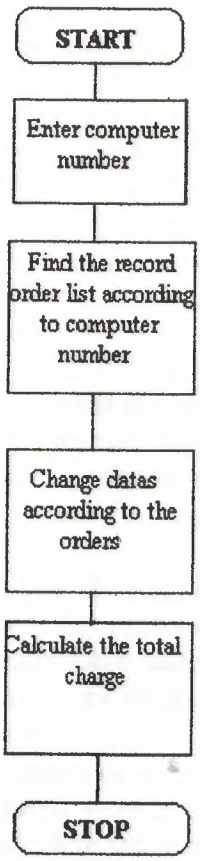


Figure 3.4. The flow of the recording order changing prices process

The program codes of these processes are:

```
Private Sub Command1_Click()
Text8.Text = (Val(Combo1.Text) * Val(Form6.Text1.Text)) + (Val(Combo2.Text) *
Val(Form6.Text3.Text)) + (Val(Combo3.Text) * Val(Form6.Text2.Text)) +
(Val(Combo4.Text) * Val(Form6.Text4.Text)) + (Val(Combo5.Text) *
Val(Form6.Text6.Text)) + (Val(Combo6.Text) * Val(Form6.Text5.Text))
Form4.Text7.Text = Val(Text8.Text)
Form4.Text5.Text = Val(Text8.Text)
End Sub

Private Sub Command2_Click()
compno$ = InputBox(" Please Enter computer no! : ")
ölçüt = "compno=" & compno$ & ""
Data1.Recordset.FindFirst ölçüt
End Sub

Private Sub Command3_Click()
Combo1.Text = "0"
Combo2.Text = "0"
Combo3.Text = "0"
Combo4.Text = "0"
Combo5.Text = "0"
Combo6.Text = "0"
Text8.Text = "0"
End Sub
```

Here three buttons used for entering amount of used services, calculating charges and clearing for new entry. Multiplying values that taking prices from database and amount of services by combo boxes over a form performs calculation.

3.1.5. Calculation of Total Charge

This process calculates the total charge including Internet and other services. Internet charge is calculated in this part of the program, by using user's time information (Elapsed time) on database file then this value multiplying by Internet price rates. Also other charges have been added in here to the total charge. Internet price is calculated for each second after the user press start button.

To find total charge, first the operator must calculate the orders from the order services form. The find button is used to point to the computer that the operator wants to calculate charge and he/she can calculate services for this computer. After operator calculated ordered services he/she also can calculate total charge by adding Internet charge.

The operator can see the possible escapes from the services menu by realizing the same computer numbers start and stop times at different time slices. If same user starts and stops many times operator is able to see these entries from database file.

Also it is possible to add all charges of all entries of a computer.

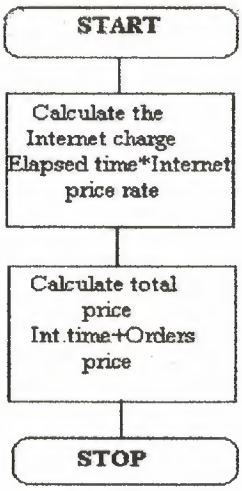


Figure 3.5. The flow of calculation of total charge process

The codes of these calculations are:

```
Private Sub Command1_Click()
computerno$ = InputBox("Which Number of computer you want to find?")
ölçüt = "computerno =" & computerno$ & ""
Data1.Recordset.FindFirst ölçüt
End Sub

Private Sub Command2_Click()
Text6.Text = Val(Text4.Text) * Val(Form6.Text7.Text)
Text5.Text = Val(Text5.Text) + Val(Text6.Text)
End Sub

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

By above codes three command buttons can find a computer, calculate total charge and clear the areas.

3.2. Terminal Program (Client Program)

The client program is more important part of the reservation project because it deals directly with the users. Basically this program sends the user's start and stop times to the certain database file for using by main program. After operator's permission, the user can see start button and starts the using services by pressing it on client program.

The security is realized by making malfunctioning all applications and Windows desktop without server permission and pressing start button.

If a user do not have an open table he/she can't see start button and cannot use services but if he/she have/has an open table can push start button and activate his/her account.

3.2.1. Password Protection for Client Program

To enter and use client program user has to enter a valid password. This process is to defend program from attacks. You can see password verification process in Figure 3.6. Operators also required entering computer numbers for terminals at startup.

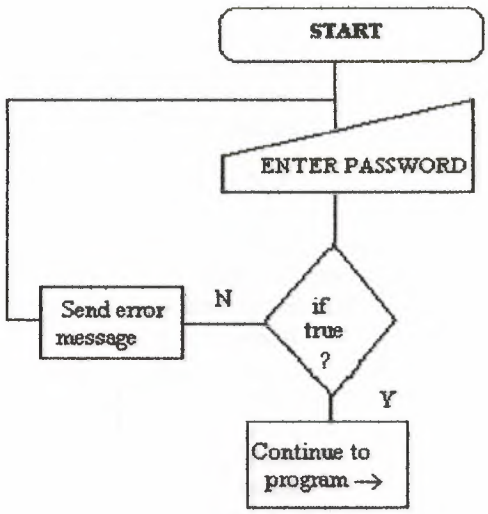


Figure 3.6. The flow of the password verification process

The codes for password verification is below:

```
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 = ""
End Sub
```

Above code checks the contents of text object and if it is correct it loads another form else sends a message to the user in a message box.

3.2.2. Recording of Start and Stop Times

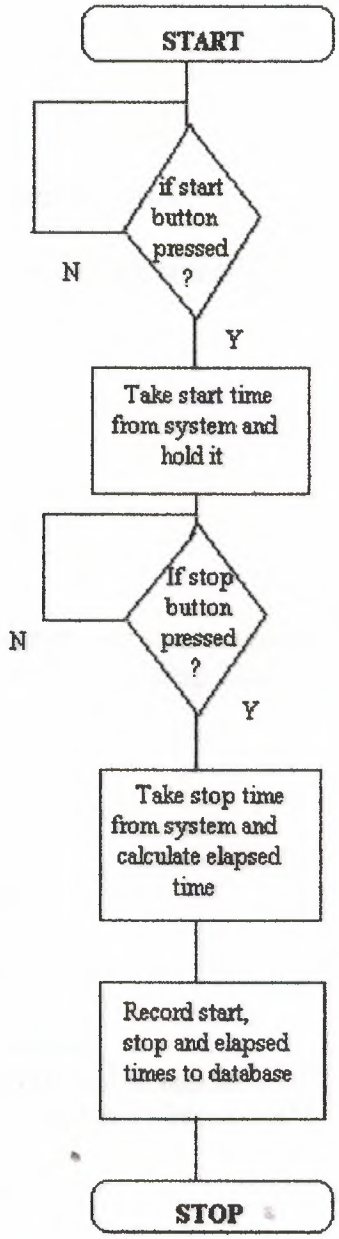


Figure 3.7. The flow of the start stop time recording process

That process records the start time whenever user starts the use services from system and again records the stop time whenever the user presses stop button. Elapsed time is also recorded in database file. A timer is used for rounded values for charges, the timer counts in each second by increasing a variable. This is more conventional way for calculating Internet price in a certainty. You can see codes below for start button:

```
Private Sub Command9_Click()
Timer1.Enabled = True
Command9.Visible = False
Command1.Enabled = True
start.Enabled = False
stop1.Enabled = True
Command9.Enabled = False
Command10.Enabled = True
'Establish start time and start timer control
starttime = Now
Label2.Caption = Format(starttime, "hh:mm:ss")
Label4.Caption = ""
Label6.Caption = ""
Command15.Enabled = True
Text1.Text = ""
End Sub

Private Sub stop1_Click()
Dim selection As Variant
Text1.Text = ""
selection = MsgBox("You cannot start your account timer again if you stop now!  Are
you sure want to stop and leave?", 52, "Stop Timer")
If selection = 6 Then
'stop button pressed
'stop button disabled
'Enabled start and exit buttons
Command1.Enabled = False
Command10.Enabled = False
Command9.Enabled = True
Command11.Enabled = True
EndTime = Now
ElapsedTime = EndTime - starttime
Label4.Caption = Format(EndTime, "hh:mm:ss")
Label6.Caption = Format(ElapsedTime, "hh:mm:ss")
Timer1.Enabled = False
Command15.Enabled = False
Text1.Text = i
i = 0
Form6.Data1.Recordset.AddNew
Form6.Text4.Text = Text1.Text
Form6.Text2.Text = Label2.Caption
Form6.Text1.Text = y
Form6.Text3.Text = Label4.Caption
Command9.Visible = True
start.Enabled = True
stop1.Enabled = False
End If
End Sub
```

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

Here the start time reflected to the user taken from system and timer activated for counting seconds.

3.2.3. Start Button Visibility Control

This is a security process for prevent non-permitted users. This process code checks a database field that can be changed by server program. If state of field is 1 user can see start button and starts to using services but if state is 0 he/she cannot see start button and cannot start using services. If users do not press start button they cannot use Windows desktop also.

If server changes state any time the client program recognizes that and makes start button visible or not. At this point client program uses dynamic record refreshing method and checks the state continuously. Refresh method works owing the timer.

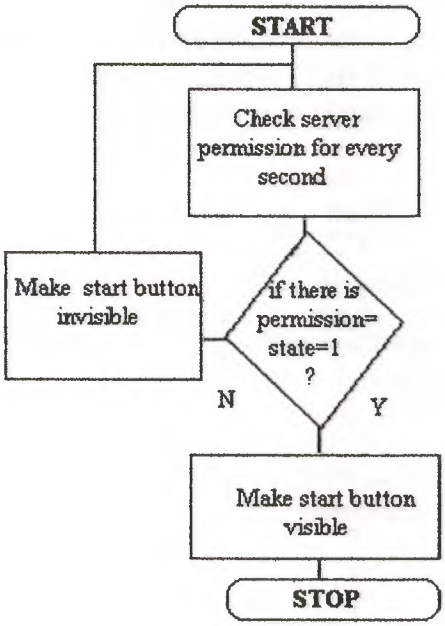


Figure 3.8. The flow of the start button visibility check process

The codes for start button visibility control are:

```
Private Sub Timer2_Timer()
Form6.Data2.UpdateRecord
If Form6.Text5.Text = 0 Then
Command9.Enabled = False
Command10.Enabled = False
start.Enabled = False
stop1.Enabled = False
Else
Command9.Enabled = True
End If
End Sub
```

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

An invisible form in client software includes the database fields on it, and it is used for visibility control by checking this field in every second with timer component.

4. OTHER APPLICATIONS

4.1 Benchmark 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 operations. At the end of calculations, the application reflects the results to the charts. 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.

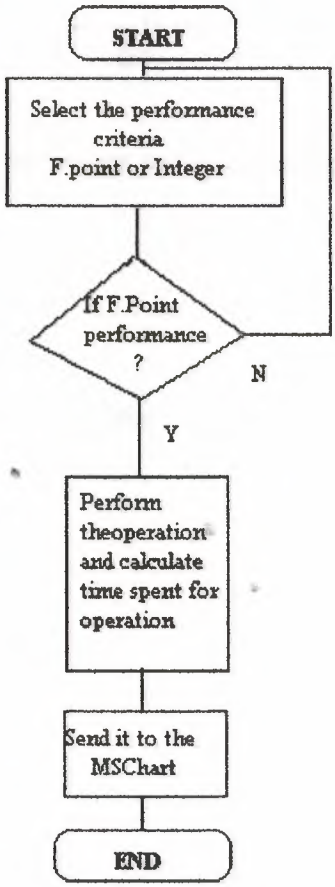


Figure 4.1. The flow of the floating-point performance benchmark application

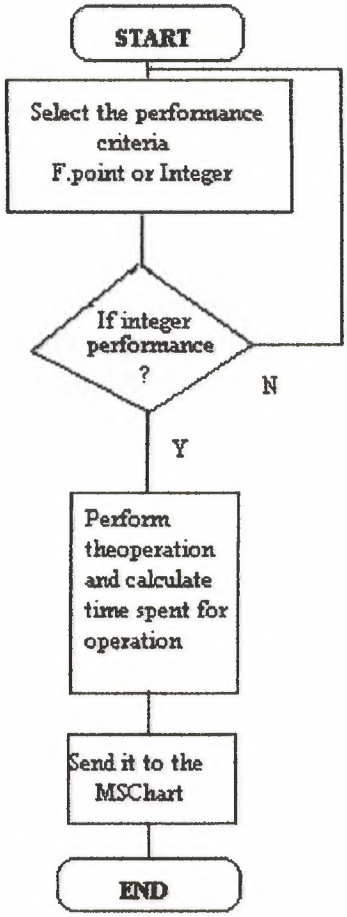


Figure 4.2. The flow of the integer performance benchmark application

The program codes in the form are:

```
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
```

Here the some values for some rows are predefined. Your CPU's data comes from time calculation, which reflects the calculation speed of your CPU. And below code performs calculations and sends the calculation times.

```
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. An Alarm for User

Any user who is working on a terminal may wish some limits to him or her about time in an Internet café. The client program has an alarm function, which uses determined time spaces as one, two and three hours. At the end of each time period alarm warns users with a message box. You can see codes 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
```

And also a flowchart for alarm application:

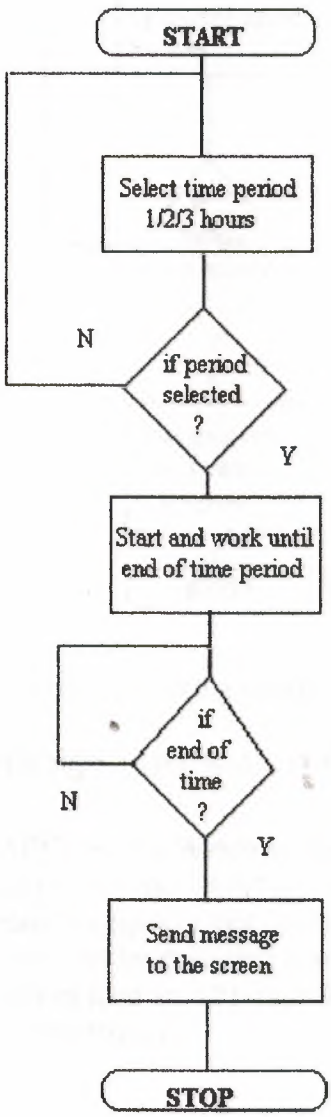


Figure 4.3. Flow of the alarm application

In this application for specified time spaces, timer counts the minutes by increasing a variable at a sequence. At the end of time a message box will appear, user can ignore the message and can set the alarm again.

4.2. Using Another Programs Over Client Program

Client program provides using some other programs running with an executable file. Many common programs are added under applications menu and user can easily run these programs over client program. Shell command lets to programmer execute other applications that are not embedded to the client program. You can execute the other application in your program window by equalizing them to some variables.

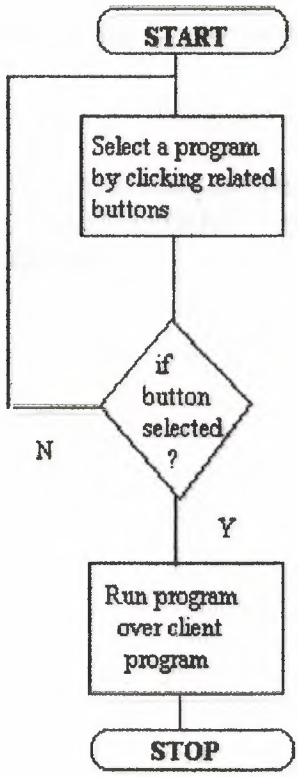


Figure 4.4. The flow of the executing another programs over client program

5.WINDOWS API'S AND COMPONENTS

As we considered before API's are the Windows functions that programmer can use it in their project. Windows supports many interfaces as forms, dialog boxes and controls. 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.

5.1 Notify Icons

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). 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

The Visual Basic language can handle or facilitate to prepare complex structured object oriented programs. Programmer can create useful creative visual applications using visual structure of the Visual Basic. The language provides to design Windows graphical interfaced applications and add Windows operating system functions to the applications. Also the programmer can insert other executable and embedded programs to his/her application for running over it.

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. Internet café programs also use some database connections and controls the fields of these files. Database files can be created and controlled easily by Visual Data Manager of Visual Basic version 6.0.

Internet café program includes some API functions and user controls on it for designing a more powerful application. Supplying moderate security and usability for users and operators, the program arises the reliability of exploitation. The architecture of the programs related with the client-server structure.

Internet café's can find some solutions for their problem on these programs, but user and operator relations still important for well working commercial.

This project described the Internet café reservation and automation technologies can adapt on programs. It also made recommendations for designing secure charging system. It covers some Visual Basic controls and functions that used in many programs.

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, İSTANBUL, 1999

[3] VBNet A guide for sample applications. Retrieved January 6, 2002 from the World Wide Web: http://www.mvps.org/vbnet/.

[4] Visual Basic Program Developers Suggestions. Retrieved January 10, 2002 from the World Wide Web: http://gan.dalf.pht.com/pub/vbbbn/cgi/am.cgi.

[5] Microsoft Visual Basic Developers Site. Retrieved January 6, 2002 from World Wide Web: http://www.microsoft.com/vbasic/.

APPENDIX A

Complete Code Structure of the Programs:

This appendix provides a reference for fundamental construction of the both programs in Visual Basic language. Our goal is cover the complete program codes; the manuals that your compiler perform that task. In Visual Basic projects are consist of the user interface windows (forms), controls, methods et al. The programmers can add every design properties on related forms; also can add a component of another forms. For that reason given codes are ordered for each form. For more comprehensive look, you need some knowledge about programming on Visual Basic.

Main Program (Server Program)

Codes are in an ordered way according to their form numbers (printed above each code block).

Form1 (All services):

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

Private Sub Form_Load()
End Sub
```

Form2 (Password verification):

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

Private Sub Form_Load()
Text1.Text = ""
End Sub
```

Form3 (Main window):

```
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 about_Click()
frmAbout.Show
End Sub

Private Sub all_Click()
Form1.Show
End Sub

Private Sub benchmark_Click()
Form8.Show
End Sub

Private Sub charges_Click()
Form4.Show
End Sub

Private Sub close_Click()
Text1.Text = 0
Data1.UpdateRecord
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 Command10_Click()
Text1.Text = 1
Data1.UpdateRecord
End Sub

Private Sub Command10_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 = "Open Table"
End Sub
```

```
Private Sub Command11_Click()
Form8.Show
End Sub

Private Sub Command11_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 Benchmark"
End Sub

Private Sub Command2_Click()
Form6.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 = "Service Rates"
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 = "Service Order"
End Sub

Private Sub Command4_Click()
compno$ = InputBox("Enter Computer number")
If compno = "" Then
Exit Sub
End If
ölçüt = "compno =" & compno$ & ""
Data1.Recordset.FindFirst ölçüt

End Sub

Private Sub Command5_Click()
Text1.Text = 1
Data1.UpdateRecord
End Sub

Private Sub Command6_Click()
Text1.Text = 0
```

```
Data1.UpdateRecord
End Sub

Private Sub Command7_Click()
Form7.Show
End Sub

Private Sub Command8_Click()
Text1.Text = 0
Data1.UpdateRecord
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_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 = "Close Table"
End Sub

Private Sub Command9_Click()
Form1.Show
End Sub

Private Sub Command9_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 = "All Charges"
End Sub

Private Sub exit_Click()
Form7.Show
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
If KeyAscii = vbkeystartkey Then
KeyAscii = 0
End If
End Sub

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

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

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

Private Sub rates_Click()
Form6.Show
End Sub

Private Sub services_Click()
Form5.Show
End Sub

Private Sub shut_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 support_Click()
MsgBox ("You can send E-mail to tdokuzer@yahoo.com for technical support")
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

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) = "ı" 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
```

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()
Text6.Text = Val(Text4.Text) * Val(Form6.Text7.Text)
Text5.Text = Val(Text5.Text) + Val(Text6.Text)
End Sub

Private Sub Command3_Click()
Data1.Recordset.Delete
Data1.Recordset.MovePrevious
End Sub

Private Sub Command4_Click()
Form1.Show
End Sub

Private Sub Command5_Click()
Unload Form4
End Sub

Private Sub Form_Load()
End Sub
```

Form5 (Service orders):

```
Private Sub Command1_Click()
Text8.Text = (Val(Combo1.Text) * Val(Form6.Text1.Text)) + (Val(Combo2.Text) *
Val(Form6.Text3.Text)) + (Val(Combo3.Text) * Val(Form6.Text2.Text)) +
(Val(Combo4.Text) * Val(Form6.Text4.Text)) + (Val(Combo5.Text) *
Val(Form6.Text6.Text)) + (Val(Combo6.Text) * Val(Form6.Text5.Text))
Form4.Text7.Text = Val(Text8.Text)
Form4.Text5.Text = Val(Text8.Text)
End Sub

Private Sub Command2_Click()
compno$ = InputBox(" Please Enter computer no! : ")
ölçüt = "compno=" & compno$ & ""
Data1.Recordset.FindFirst ölçüt
End Sub

Private Sub Command3_Click()
Combo1.Text = "0"
Combo2.Text = "0"
Combo3.Text = "0"
Combo4.Text = "0"
Combo5.Text = "0"
Combo6.Text = "0"
Text8.Text = "0"
End Sub
```

```
Private Sub Command4_Click()
Combo1.Text = "0"
Combo2.Text = "0"
Combo3.Text = "0"
Combo4.Text = "0"
Combo5.Text = "0"
Combo6.Text = "0"
Text8.Text = "0"
Unload Form5
End Sub

Private Sub Command5_Click()
Unload Me
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 "0"
Combo4.AddItem "1"
Combo4.AddItem "2"
Combo4.AddItem "3"
Combo5.AddItem "0"
Combo5.AddItem "1"
Combo5.AddItem "2"
Combo5.AddItem "3"
Combo6.AddItem "0"
Combo6.AddItem "1"
Combo6.AddItem "2"
Combo6.AddItem "3"
End Sub

Private Sub Text8_Change()
Text1.Text = ""
End Sub
```

Form6 (Service rates):

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

```
End Sub

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()
End Sub
```

Form7 (Benchmark application):

```
Private Sub Command1_Click()
End
End Sub

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()
End Sub
```

Form8:

```
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

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

Form Splash:
Private Sub Form_KeyPress(KeyAscii As Integer)
Unload Me
End Sub

Private Sub Form_Load()
Show
CurrentX = 2700
CurrentY = 3350
FontSize = 10
FontBold = False
Print "Server"
End Sub

Private Sub Timer1_Timer()
Unload frmSplash
Form2.Show
End Sub
```

Form About (About box):

```
Option Explicit
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
               Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1
Const REG_DWORD = 4
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

Public Sub StartSysInfo()
On Error GoTo SysInfoErr
Dim rc As Long
Dim SysInfoPath As String
If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
Else
```


```
GoTo SysInfoErr
End If
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
Dim rc As Long
Dim hKey As Long
Dim hDepth As Long
Dim KeyValType As Long
Dim tmpVal As String
Dim KeyValSize As Long
rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey)
If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError
tmpVal = String$(1024, 0)
KeyValSize = 1024
rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
KeyValType, tmpVal, KeyValSize)
If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError
If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then
tmpVal = Left(tmpVal, KeyValSize - 1)
Else
tmpVal = Left(tmpVal, KeyValSize)
End If
Select Case KeyValType
Case REG_SZ
KeyVal = tmpVal
Case REG_DWORD
For i = Len(tmpVal) To 1 Step -1
KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))
Next
KeyVal = Format$("&h" + KeyVal)
End Select
GetKeyValue = True
rc = RegCloseKey(hKey)
Exit Function
GetKeyError:
KeyVal = ""
GetKeyValue = False
rc = RegCloseKey(hKey)
End Function
```

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

Form Splash (Splash screen):

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

Private Sub Form_Load()
Show
CurrentX = 2700
CurrentY = 3350
FontSize = 10
FontBold = False
Print "Server"
End Sub

Private Sub Timer1_Timer()
Unload frmSplash
Form2.Show
End Sub
```

Client Program (Terminal Program)

Codes are in an ordered way according to their form numbers (printed above each code block). Same form applications of the programs are not duplicated.

Form1 (Computer number request):

```
'Form1 also includes same keyword verification application with server program
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 window):

```
'Form2  have similar components with server program's form2
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
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 Image1_Click()
End Sub

Private Sub Command0_Click()
Dim t2
t2 = Shell("c:\windows\Iexplore.exe", 4)
End Sub

Private Sub about_Click()
frmAbout.Show
End Sub

Private Sub alarm_Click()
Form8.Show
End Sub

Private Sub Command1_Click()
Dim minimized As VbAppWinStyle
Form3.WindowState = 1 - minimized
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 = "Windows Desktop"
End Sub

Private Sub Command10_Click()
Dim selection As Variant
Text1.Text = ""
'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
Command1.Enabled = False
alarm.Enabled = False
Command10.Enabled = False
Command9.Enabled = True
Command11.Enabled = True
Command2.Enabled = False
EndTime = Now
ElapsedTime = EndTime - starttime
Label4.Caption = Format(EndTime, "hh:mm:ss")
Label6.Caption = Format(ElapsedTime, "hh:mm:ss")
Timer1.Enabled = False
Command15.Enabled = False
Text1.Text = i
i = 0
Form6.Data1.Recordset.AddNew
Form6.Text4.Text = Text1.Text
Form6.Text2.Text = Label2.Caption
Form6.Text1.Text = Y
Form6.Text3.Text = Label4.Caption
Command9.Visible = True
start.Enabled = True
stop1.Enabled = False
End If
StatusBar1.Panels(1).Text = "Timer is not active"
End Sub

Private Sub Command10_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 Command11_Click()
Form6.Show
End Sub

Private Sub Command12_Click()
Load Form4
Form4.Show
End Sub

Private Sub Command12_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 Command15_Click()
Form7.Show
End Sub

Private Sub Command15_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 = "Applications"
End Sub

Private Sub Command16_Click()
Form5.Show
End Sub

Private Sub Command16_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 Command2_Click()
Form8.Show
End Sub

Private Sub Command3_Click()
Form1.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 = "Alarm"
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 = "Benchmark"
End Sub

Private Sub Command9_Click()
Timer1.Enabled = True
alarm.Enabled = True
```

```
Command2.Enabled = True
Command9.Visible = False
Command1.Enabled = True
start.Enabled = False
stop1.Enabled = True
Command9.Enabled = False
Command10.Enabled = True
'Establish start time and start timer control
starttime = Now
Label2.Caption = Format(starttime, "hh:mm:ss")
Label4.Caption = ""
Label6.Caption = ""
Command15.Enabled = True
Text1.Text = ""
StatusBar1.Panels(1).Text = "Timer is active"
End Sub

Private Sub Command9_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 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!")
Text1.Text = ""
Form3.Caption = "Net@Cafe1.0(Com400)"
Command2.Enabled = False
Command10.Enabled = False
alarm.Enabled = False
stop1.Enabled = False
Label2.Caption = ""
Label4.Caption = ""
Label6.Caption = ""
Command15.Enabled = False
Command1.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 shut_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 start_Click()
Timer1.Enabled = True
alarm.Enabled = True
Command2.Enabled = True
Command1.Enabled = True
Command9.Visible = False
Command9.Enabled = False
Command10.Enabled = True
'Establish start time and start timer control
starttime = Now
Label2.Caption = Format(starttime, "hh:mm:ss")
Label4.Caption = ""
Label6.Caption = ""
Command15.Enabled = True
Command9.Caption = False
Text1.Text = ""
start.Enabled = False
stop1.Enabled = True
StatusBar1.Panels(1).Text = "Timer is active"
End Sub

Private Sub stop1_Click()
Dim selection As Variant
Text1.Text = ""
selection = MsgBox("You cannot start your account timer again if you stop now!  Are
you sure want to stop and leave?", 52, "Stop Timer")
If selection = 6 Then
'stop button pressed
'stop button disabled
'Enabled start and exit buttons
alarm.Enabled = False
Command1.Enabled = False
Command10.Enabled = False
Command9.Enabled = True
```

```
Command11.Enabled = True
Command2.Enabled = False
EndTime = Now
ElapsedTime = EndTime - starttime
Label4.Caption = Format(EndTime, "hh:mm:ss")
Label6.Caption = Format(ElapsedTime, "hh:mm:ss")
Timer1.Enabled = False
Command15.Enabled = False
Text1.Text = i
i = 0
Form6.Data1.Recordset.AddNew
Form6.Text4.Text = Text1.Text
Form6.Text2.Text = Label2.Caption
Form6.Text1.Text = Y
Form6.Text3.Text = Label4.Caption
Command9.Visible = True
start.Enabled = True
stop1.Enabled = False
End If
StatusBar1.Panels(1).Text = "Timer is not active"
End Sub

Private Sub support_Click()
MsgBox ("You can send e-mail to us for your questions tdokuzer@yahoo.com")
End Sub

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

Private Sub Timer2_Timer()
Form6.Data2.UpdateRecord
Form6.Data1.UpdateRecord

If Form6.Text5.Text = 0 Then
Command9.Enabled = False
Command10.Enabled = False
start.Enabled = False
stop1.Enabled = False
Command1.Enabled = False
Else
Command9.Enabled = True
End If
End Sub
```
"Form2 and 4 includes same applications with server program as verification and benchmark

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 (Applications):

```
Private Sub Command3_Click()
Dim pr
pr = Shell("c:\programfiles\netspace\netspace.exe", 4)
End Sub

Private Sub Command4_Click()
Dim t
t = Shell("c:\windows\pbrush.exe", 4)
End Sub

Private Sub Command5_Click()
Dim m
m = Shell("c:\windows\mplayer.exe", 4)
End Sub

Private Sub Command6_Click()
Dim l
l = Shell("c:\programfiles\winzip\winzip32.exe")
End Sub

Private Sub Command8_Click()
Dim w
w = Shell("c:\windows\outlook.exe", 4)
End Sub

Private Sub Form_Load()
End Sub
```

Form8 (Alarm):

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

Private Sub Command2_Click()
Unload Form8
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 If
End Sub

Private Sub Timer2_Timer()
```

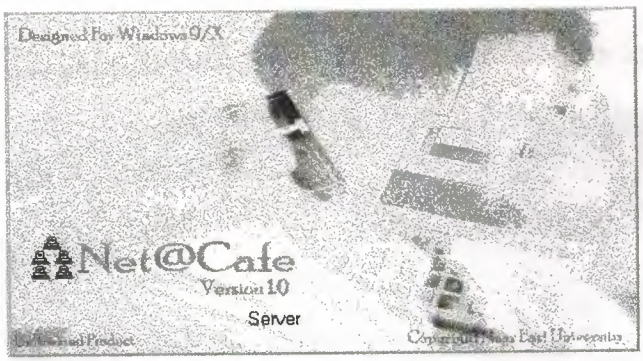
```
k = k + 1
If  k = 60 Then
MsgBox ("Time is up")
End If
End Sub

Private Sub Timer3_Timer()
k = k + 1
If  k = 120 Then
MsgBox ("Time is up")
End If
End Sub
'Client program also includes a splash and about form similar to server program
```

APPENDIX B

Visuals of the Programs

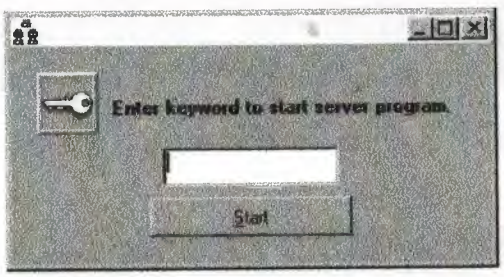
This appendix includes the graphical user interface part of the programs that offered by the programming language Visual Basic for the each form.

Main Program (Server Program)

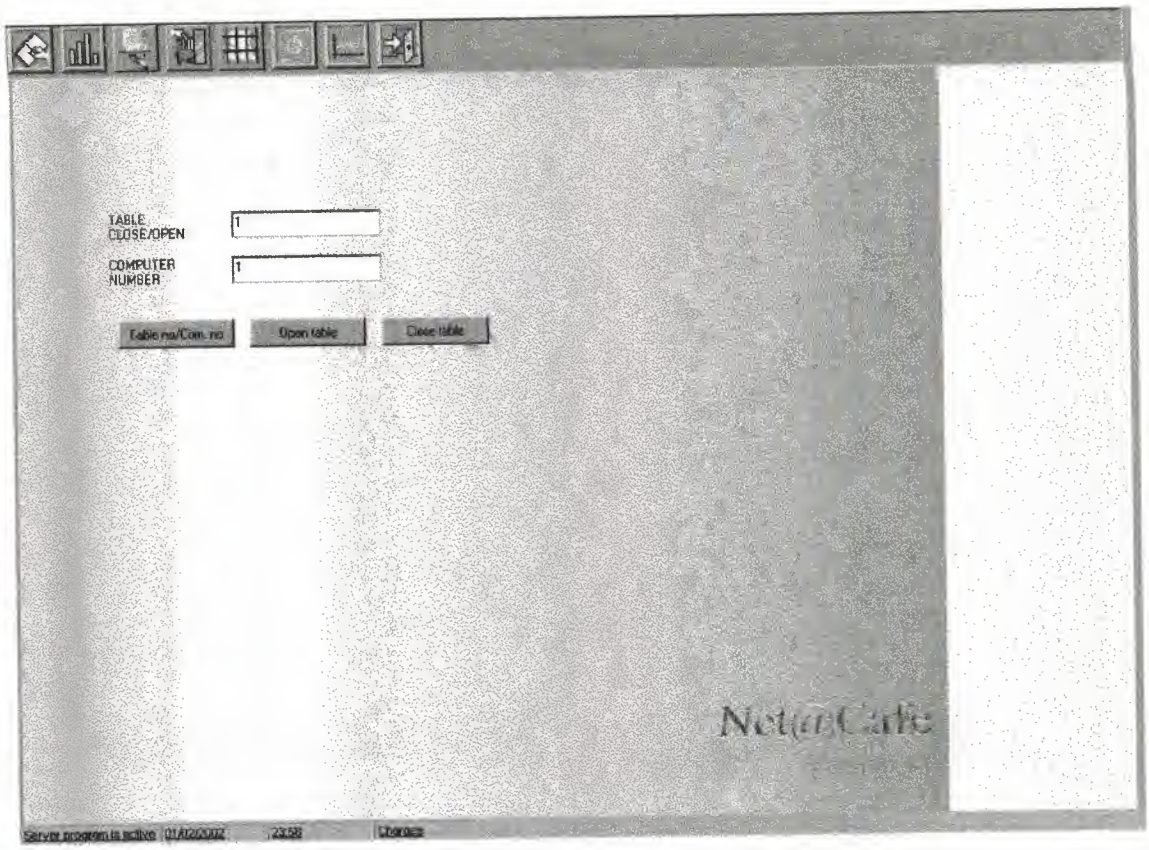


Picture1. Form Splash Server

Splash screen is used in both programs for reflect that the program is preparing to run.

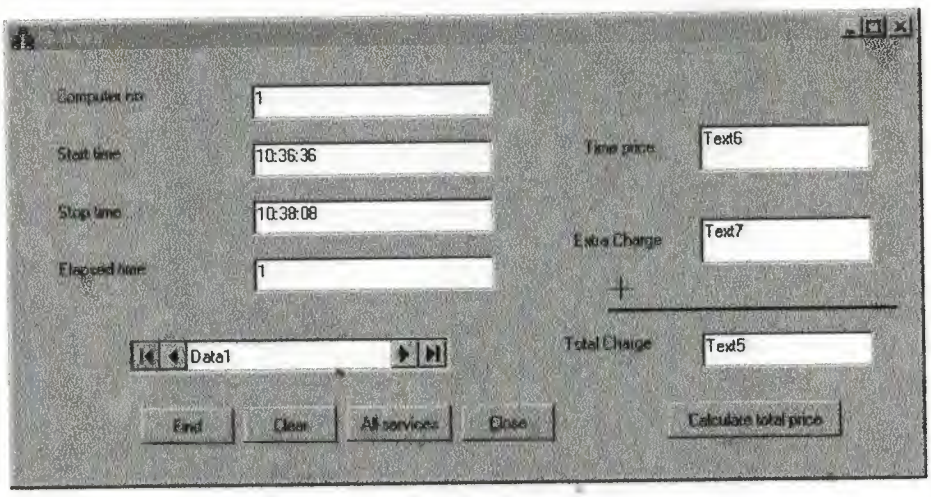


Picture 2. Form 2 to verify password



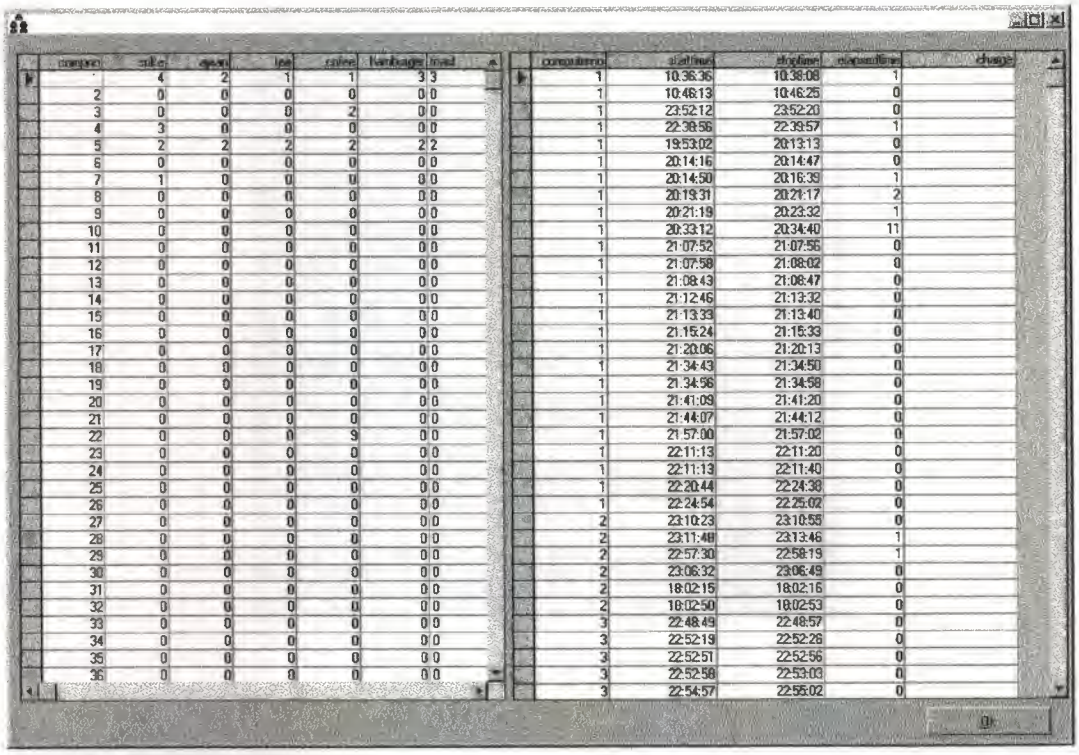
Picture 3. Form 3 main window.

Operator is able to use all properties of the server program on main form3.



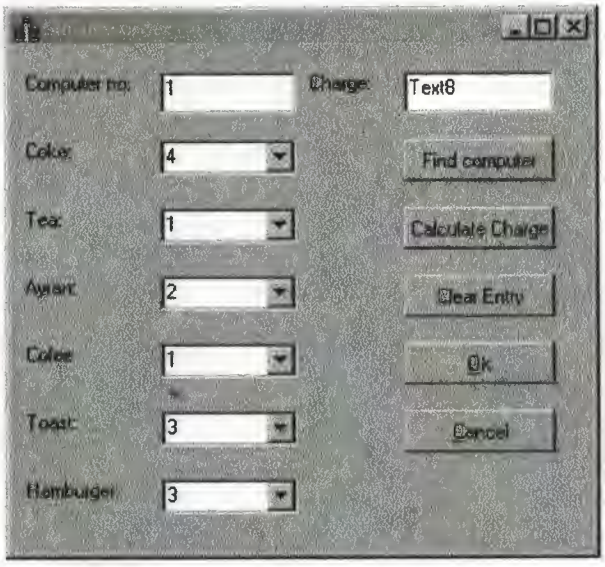
Picture 4. Form 4 charges form used to charge calculations

Picture 5. Find computer message box to calculate charge on form4



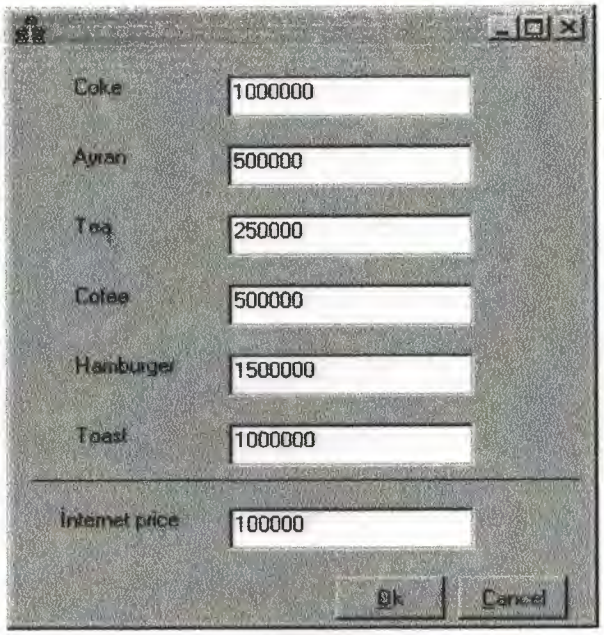
Picture 6. The all charges window on form4

The operator can see all computers start/stop times and services by that window.



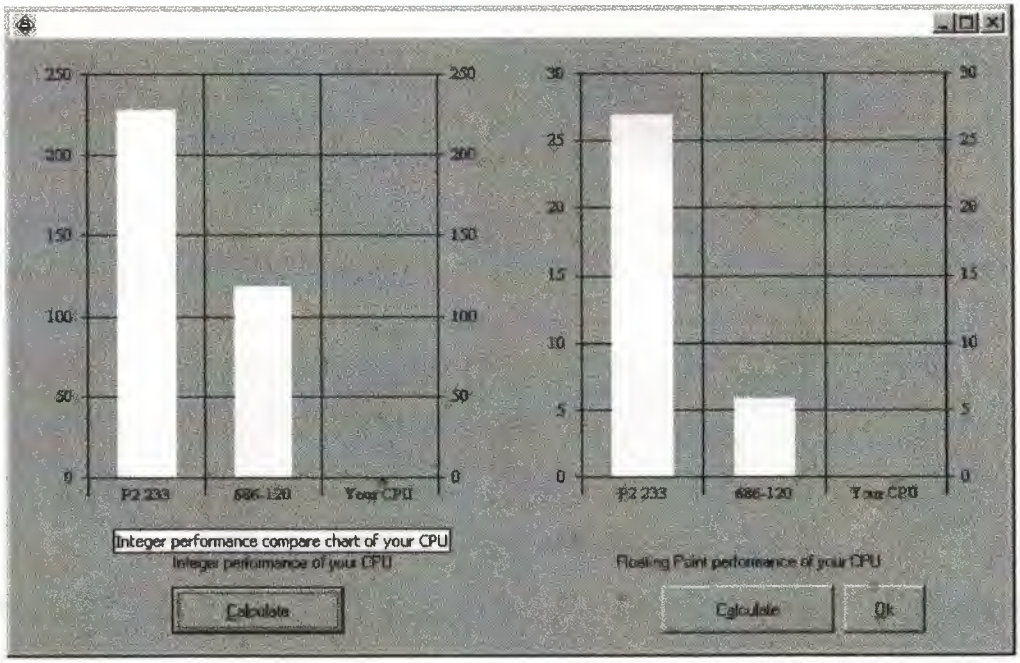
Picture 7. Form 5 service orders

The operator can add services by combo boxes after finding the computer, and send charge to the charge form 4 to calculate total charge.

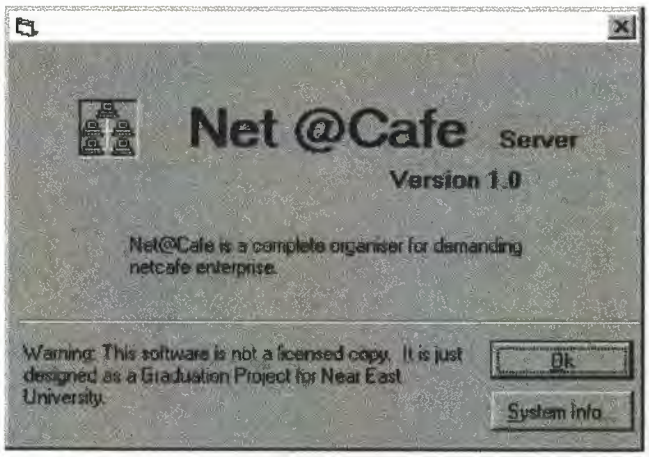


Picture 8. Form 6 service rates

Operator can easily change rates of goods & services included Internet price for per minute.

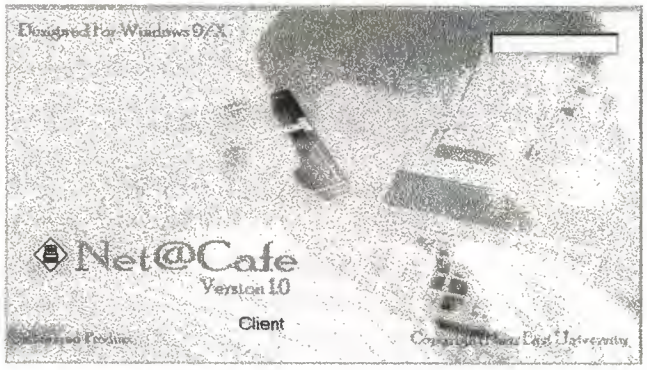


Picture 9. Form 7 Benchmark applications



Picture 10. About form that includes system information application

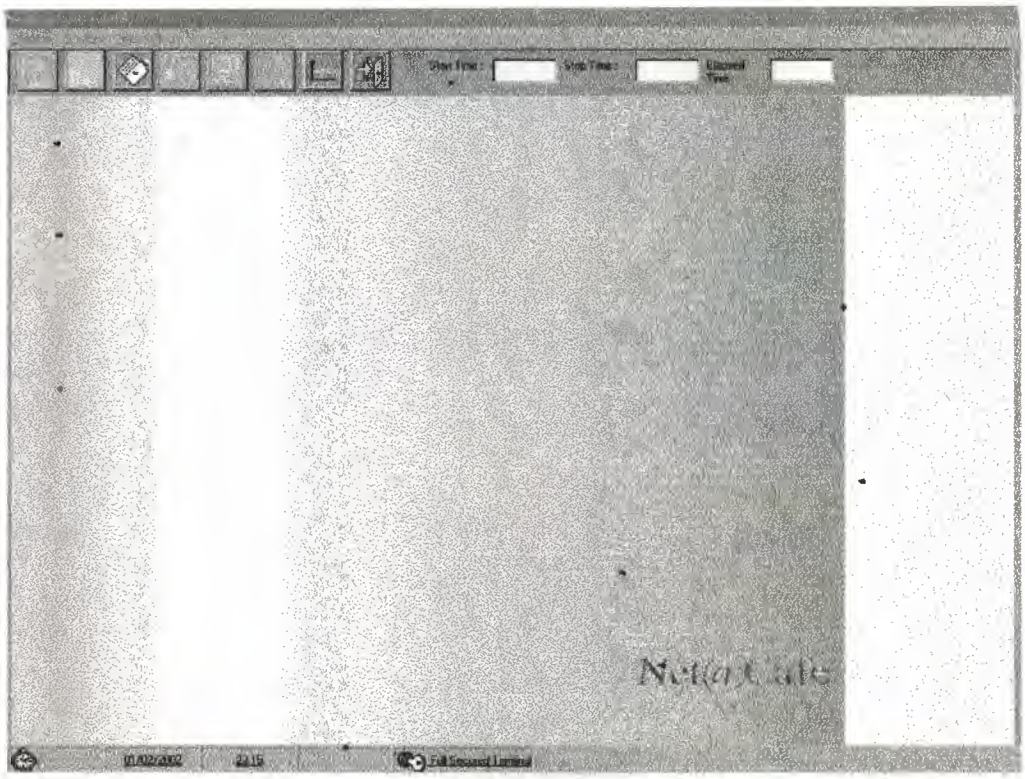
Client Program (Terminal Program)



Picture 11. Form Splash Client

Splash screen is used in both programs for reflect that the program is preparing to run.

Picture 12. Computer number message box to reach state database file to initialize session

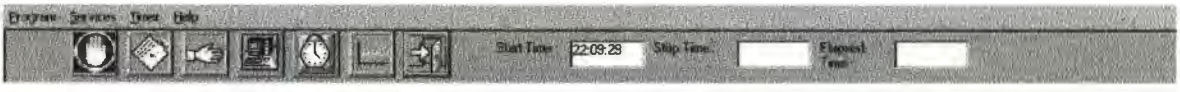


Picture 13.Form 3 main window

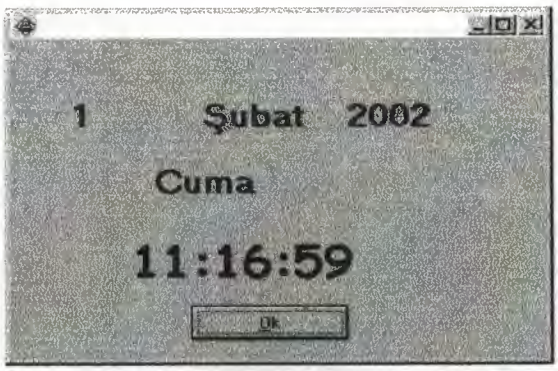
User is able to use all properties of the client program on main form3. Form 3 is the active program window for user so it needs to have permissions from server. In Picture 13, there is no permission; buttons and options are not enabled to the user.

Picture 14. Permitted client toolbar buttons (Start button not pressed yet)

Picture 15. Message box that shows client program ready to start

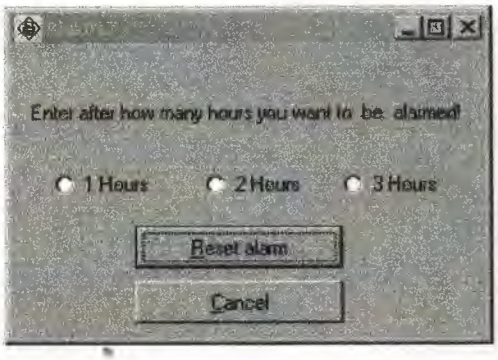


Picture 16. Client program toolbar buttons after start button pressed

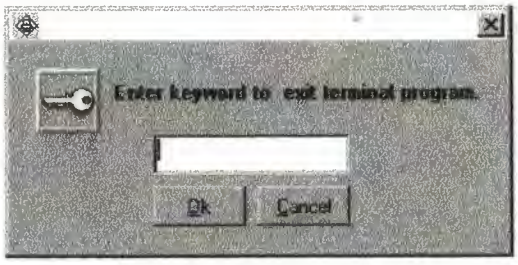


Picture17. Form 5 to reflect date and time to user

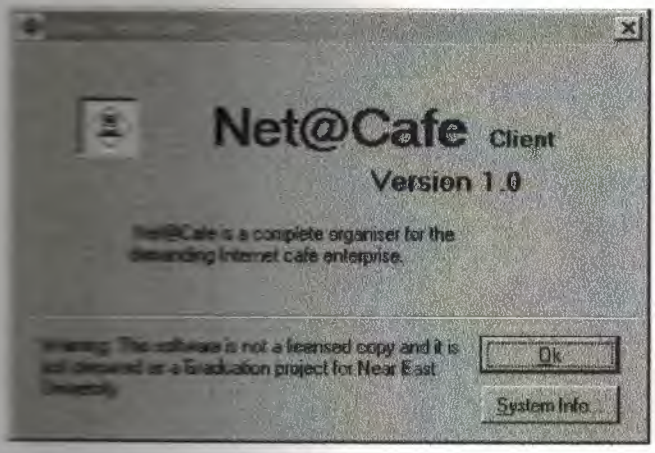
Picture 18. Form 7 Other programs that run under client program



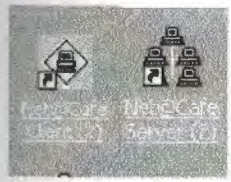
Picture 19. Form 8 Alarm that warns user



Picture 20. Form exit to close client program



Picture 21. About form that includes system information application



Picture 22. Icons of both programs on desktop

Picture 23. Icons of both working program visible in System Tray