****

A Project Report on

*“*LAMON*”*

Submitted by

Aakash Shah – 1380044

Prerak Shah –1380048

Dharmil Shah-1380051

Guided By

Mr. Janardan Kulkarni

Term – December’15 to April’16

Department of Computer Engineering

SVKM Shri Bhagubhai Mafatlal Polytechnic,

Irla, N. R. G. Marg, Vile Parle (W), Mumbai -400056



**ABSTRACT**

In our project we have introduced Web Based LAN Monitoring (LAMON), having functionalities such as process view, process kill, remote shutdown, logoff, and restart, creation of log file and sending of e-mail, creating a session for each user, giving access rights and permission and revoking them.

There are forms prepared for backhand connectivity of client and server.SQL Database is used as mediator to our backhand process and front hand process. For Web based, we have hosted a website having administrators login and actions can be performed from the website.

**ACKNOWLEDGEMENT**

There is always some external source and guidance incorporated through someone that leads to the successful completion of a project.

We are deeply grateful to our project guide **MR. JANARDAN KULKARNI** sir, who has been an unweaving source of support and a guideline inspiration and has an uncanny knack for explaining the most difficult concepts in a way that everyone can understand.

We thank him for believing in us and sanctioning our project **“WEB BASED LAN MONITORING A.K.A LAMON”**. We would also like to thank our technician **MR. CHETAN DESAI** sir and the entire teaching and non-teaching staff for enduring their incomparable suggestion for the project and for their countless efforts in completing the same. We would also like to thank and mention PANKAJ SIR & NEETA MAAM for helping us solving our doubts and difficulties. Fulfillment of all our needs and requirements were taken care of, which is the reason we are able to present our fully functional project **LAMON**.

**INDEX:**

**TITLES PAGE NO’S**

1. TIMING DIAGRAM 07

2. PROBLEM DEFINATION 08

3. PROJECT FOUNDATION AND MOTIVATION 09

4. INTRODUCTION 10

5. EXISTING SYSTEM

5.1 E-MAIL BASED 12

5.2 GSM BASED 12

6. PROPOSED SYSTEM 14

7. SYSTEM ARCHITECTURE 17

8. AIMS AND OBJECTIVES 22

9. TECHNOLOGY USED

9.1 VB.NET 25

9.2 ASP.NET 28

9.3 SQL SERVER 2008 30

10. DIAGRAMS

10.1 SEQUENCE DIAGRAM 32

10.2 DFD DIAGRAM 33

10.3 FLOWCHART 34

11. SOURCE CODE

11.1 SERVER CODE 36

11.2 CLIENT CODE 55

11.3 LOGIN ASPX CODE 74

11.4 LOGIN ASPX VB CODE 79

11.5 HOME ASPX CODE 82

11.6 HOME ASPX VB CODE 91

11.7 KILL ASPX VB CODE 95

11.8 KILL ASPX VB CODE 97

12. SCREENSHOTS 100

13. SYSTEM REQUIREMNETS 113

14. ADVANTAGES 115

15. DISADVANTAGES 117

16. APPLICATIONS 119

17. FUTURE ENHANCEMENTS 122

18. CONCLUSION 123

19. REFERENCE 124

20.ABOUT US 125

**TIMING DIAGRAM:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| WEEKS | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 14 |
| ANALYSIS |  |  |  |  |  |  |  |  |  |  |  |  |  |
| FORMS DESIGN |  |  |  | | |  |  |  |  |  |  |  |  |
| WEBSITE DESIGN |  |  |  |  | | | |  |  |  |  |  |  |
| CODING |  |  |  |  |  |  |  | | | |  |  |  |
| TESTING |  |  |  |  |  |  |  |  |  | | |  |  |
| DEBUGGING |  |  |  |  |  |  |  |  |  |  | | | |
| MODIFICATION |  |  |  |  |  |  |  |  |  | | | |  |
| REVIEW |  |  |  |  |  |  |  |  |  |  |  |  | |

**PROBLEM DEFINATION:**

The most highly demanding task in the field of IT industries is the efficient computer network management. There are many urgent issues or requests related to such networks which network manager needs to solve immediately for avoiding the any kind of interruptions. But sometimes, network managers may be situated at different places, so in such cases there is not possible to resolve any urgent issues with the office network tasks. Thus in order to solve such problems in this project we describes the architecture of a novel tool for network management using GPRS /Web mobile devices.

In a concern, computers are grouped together to form a network. To manage and control the activities of the network while in office is an easy task. But, while you are outstation / away from office, how do you go about with monitoring and controlling of network? Instead of depending on third party information, you can always have your cell phone serve the purpose. Just load the project in your cell phone, login anytime to the application and see who is busy with what in the office. Consider a LAN setup with the server machine connected to Web service. The interaction between the clients and the wireless media happens through this server. At the end, it shows results by depicting the screen of several mobile devices, which provide network management information.

Many organizations have a huge network of LAN which is difficult to manage by a single system administrator manually. The LAN Monitoring with Web system will monitor all the physical devices i.e. PC’s from handheld from Website. The system has shutdown, Restart, Logoff, process list, net view modules to trace and keep track of various client activities. The targeted users are labs in colleges and various other organizations

**PROJECT FOUNDATION & MOTIVATION**

We all have been there when restriction are imposed on the usage of our personal computers at work or school/college or any other institute, we tend to break these restrictions and find new ways to exploit these restriction. Well any exploitation of the laws can be fixed but you never come to know about the person who committed the exploitation. For this reason we have tried to build our project i.e. lan monitoring.

Basically what it does is captures the current process list, kills a certain task, has the ability to shutdown/restart/log off the personal computer remotely. This can be done at the ease of the administrator from any distance across the world. But of course the personal computer needs to be connected in the local area network.

Certainly this does not reduce the exploitation carried out by the users but it definitely can get to the responsible personal computer.

****

**INTRODUCTION**

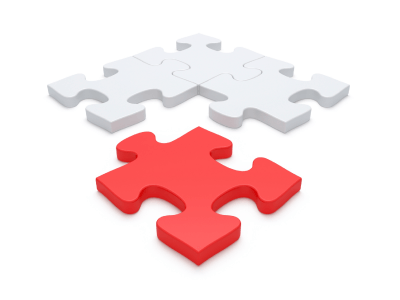
**INTRODUCTION**

Network management is a hard task. It requires the coordination of several issues, such as different hardware, software and communication environments. A Network Management System (NMS) is a set of software and hardware tools for the monitoring and management of a network. Sometimes, we think that Wide Area Networks (WANs) are harder to manage than Local Area Networks. Modern LANs are so complex that the difficulty of managing them is similar to, or even bigger than, the difficulty of managing a WAN.

Nowadays, almost every IT department settles Service Level Agreements (SLAs) with their users, which requires additional resources in network monitoring systems. Network monitoring has many disciplines, such as installation, integration, and the coordination of hardware, software and human resources for the monitoring, test, configuration, analysis and evaluation. The final goal is to control the network in real-time, knowing its resources, its performance, and then comparing this information with the user requirements. All these tasks should be executed within an acceptable budget.

The increasing demand for applications and services that allow the development of new business, stimulate companies to increase the investment on network infrastructure services and telecommunication systems. These systems have had their complexity increased, interconnecting equipments of different manufacturers, which generally use different technologies. Accordingly, the increasing use of SLAs has demanded sophisticated monitoring and management systems.

IP based network management protocols appeared in the 80´s with the development of the Host Monitoring Protocol (HMP), High-Load Entity Management System (HEMS) and Simple Gateway Monitoring Protocol (SGMP). These protocols evolved towards the Simple Network Management Protocol (SNMP). The Internet Engineering Task Force (IETF) Application MIB Working Group is responsible for the definition of a set of managed objects that provides information about configuration, faults, performance, accountability and security. These managed objects offer the necessary information to define the quality of service and reliability of a network infrastructure. The basic parameters of SLAs are established by this information.

****

**EXISTING SYSTEM**

**5.1 EMAIL BASED LAN MONITORING**

The **EMAIL BASED LAN MONITORING** system provides maximum details about the network to the administrator on their email account, when administrator is away from office or goes out station. Also using email we can develop various network utilities which are required to effectively monitor a LAN network. It aims to develop an integrated software solution that allows a network administrator to remotely monitor his LAN network by his email account. But there are lots of disadvantages to monitor and control the network using email. There can be connection problem in email. In the era of internet services, emails are widely used and it has penetrated every part of our life, but remote monitoring of networks through email is still mirage.

**5.2 GSM BASED LAN MONITORING**

GSM/SMS Based LAN monitoring can monitored network by sending messages from anywhere using GSM modem. Admin sends his request to the server via GSM modem. GSM modem is the middleware between admin and sever. Sever identifies client. It sends request to the client and get response back from client. Then server will send response to admin. The SMS consists mobile no. of admin, client name and operation to be performed. But there are various drawbacks of GSM based LAN monitoring system. These may be cost of SMS is high and failure may occur due to low balance also due to the unavailability of SMS service.

****

**PROPOSED SYSTEM**

**LAN MONITORING WITH WEB**

* In the present generation systems, there is a need for the administrator has to go all around the network in order to terminate any system that is left non-terminated.
* The processes that are running in a particular system can be viewed in website.
* We can kill that process from website also.
* The processes that are running in a particular system can be viewed only in that system itself by using the present generation software’s.

**Technical Overview:-**

Here aim is to build a user interface, using which we can interface with different PCs connected in LAN by sending web Request to perform large number of operations on them. It is a tool which can be used in various fields. Using mobile computer communication using Web we can operate any PC with particular IP address in LAN. Suppose administrator wants to shutdown any computer connected to the LAN he/she need to send Web Request to the Client connected to the server. Client continually read for any new Web Request if he found new Request then command stated in the Request will be executed by the client and client will shutdown the stated computer. Numbers of operations that can be performed by administrator are as follows:

**Process Management:**

* Shut down.

Admin of the application can shut down the user PC.

* Restart.

Admin of the application can restart the system.

* Logoff.

Admin of the application can logoff the system.

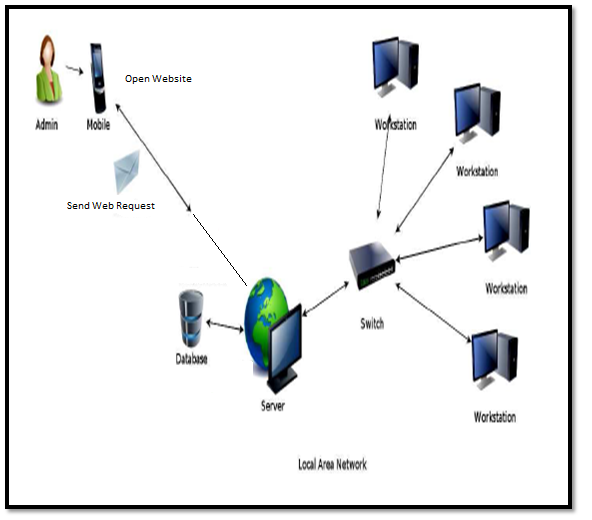
* Kill the process.

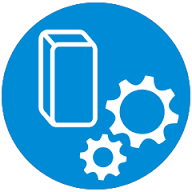
Admin of the application can kill the process of the application.

* VIEW THE RUNNING PROCESS LIST

Admin can view the list of the running processes.

**DIAGRAM:**



****

**SYSTEM ARCHITECHTURE**

**SYSTEM ARCHITECHTURE:**

Design of system characterized into three modules: Client module, Web Server module and Web Service Module.

* **Client:**

The client or user on the network requests for information (kill, shutdown, Restart) to access, requests for rights & privileges to access.

* **Web Server:**

The requests received are analyzed by server. Then server performs the respective action as directed by the Administrator.

* **Web service:**

Request is send to the client machine is done by Web service.

* **Authentication:**

Description**:** The system offers access to client details only by validating the admin with the unique username and password.

Functional Requirements**:** All system should have client program running. The server should identify all clients by their respective static addresses.

* **Input:** Username and password.
* **Output:** Client details.
* **Process List**

Description**:** This utility is used to view the activities of various clients. Moreover it provides real time reporting. The administrator using this module can determine the processes that are currently running in a particular system, present in a network. The administrator by specifying the IP address of the remote system obtains the details of the processes. The administrator can terminate any of the process by selecting that particular process. By terminating a process, that particular application will be stopped. This module is similar to that of a task manager present in the windows operating system.

Stimulus/response sequences**:** Reporting and alerting.

Functional Requirements**:** All system should have client program running. The server should identify all clients by their respective static addresses.

* **Output:** process report, process list, process kil
* **Net view**

Description**:** This utility gives information of all the clients which are connected the server. Moreover it provides real time reporting.

Stimulus/response sequences**:** Reporting and alerting.

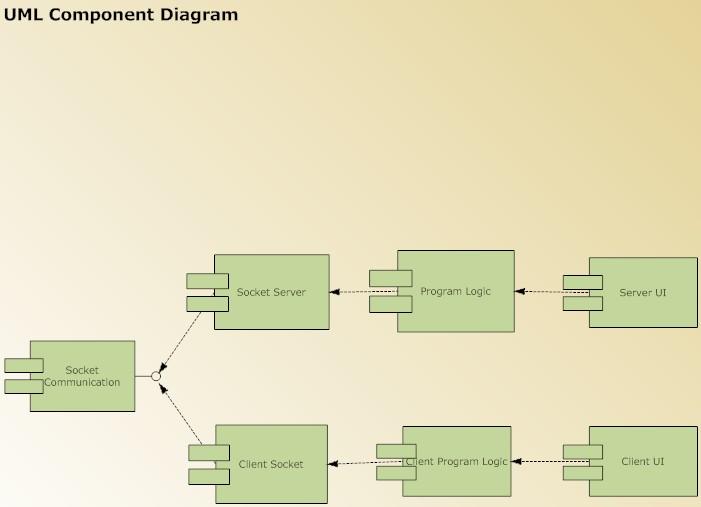
Functional Requirements**:** All system should have client program running. The server should identify all clients by their respective static addresses.

* **Output:** clients report
* **Remote Handling**

Description**:** The Remote Handling module handles the terminal operations such as shutdown, restart and logoff, by specifying the IP address of the remote system. This module asks the administrator to enter the system address on which he wants to perform the terminal operations. It also asks the administrator to mention one of the above three terminal operations to be performed on the remote system. All system should have client program running. The server should identify all clients by their respective static addresses.

* **Output:** particular pc shutdown, restart, logoff

**DIAGRAM:**



****

**AIMS AND OBJECTIVES**

**AIMS AND OBJECTIVES**

The main aim of this project is to provide maximum details about the network to the administrator on their mobile phone, when he/she is away from office / goes out station. Apart from this following are the objectives of this project:

* To design the strategies for the remote monitoring of LAN.
* To implement the designed skeleton for monitoring the LAN.
* To provide the network management information’s on the mobile screens as well as on website.

Today, the world is rapidly changing the statement “We are in the world” to “World is in our hands”. The main aim of our project is to control and monitor the LAN network from our wireless handheld device i.e. cell phone from anywhere irrespective of distance. Say, you have a LAN setup at your office. Sitting at home you want to learn the LAN status. You can do so by storing this project in your cell phone and executing the same.

****

****

**TECHNOLOGY USED**

**9.1 VB.NET**

The new language created by is Microsoft: C# (pronounced “C-Sharp”). This is a new language, based on C/C++. C#, like VB.NET, is built specifically for the .NET Framework, and much has been written about it. Given all the hype, some people might wonder why they should choose VB.NET over C#.

Although both VB.NET and C# projects are created in the Visual Studio.NET environment, VB.NET was created specifically for VB developers and has a number of unique features that make it a great choice for building .NET applications.

VB.NET is still the only language in VS.NET that includes background compilation, which means that it can flag errors immediately, while you type.

VB.NET is the only .NET language that supports late binding. In the VS.NET IDE, VB.NET provides a dropdown list at the top of the code window with all the objects and events; the IDE does not provide this functionality for any other language.

VB.NET is also unique for providing default values for optional parameters, and for having a collection of the controls available to the developer. C#, like its C and C++ brethren, is case sensitive, something that drives most experienced VB developers crazy. In addition, C# uses different symbols for equality (=) and comparison (==). Finally, if you know VB, you are further down the road with VB.NET than you are with C#.

**Advantages of VB.NET**

1**.** Build Robust Windows-based Applications

With new Windows Forms, developers using Visual Basic.Net can build Windows-based applications that leverage the rich user interface features available in the Windows operating system. All the rapid application development (RAD) tools that developers have come to expect from Microsoft are found in Visual Basic .NET, including drag-and-drop design and code behind forms. In addition, new features such as automatic control resizing eliminate the need for complex resize code.

New controls such as the in-place menu editor deliver visual authoring of menus directly within the Windows Forms Designer. Combined with greater application responsiveness, as well as simplified localization and accessibility, these new features in Windows Forms make Visual Basic .NET the choice for today's Visual Basic developers.

2**.** Resolve Deployment and Versioning Issues Seamlessly

Visual Basic .NET delivers the answer to all of your application setup and maintenance problems. With Visual Basic .NET, issues with Component Object Model (COM) registration and DLL overwrites are relics of the past. Side-by-side versioning prevents the overwriting and corruption of existing components and applications.

XCOPY deployment enables Windows-based applications to be deployed to client machines simply by copying files into the desired application directory. In addition, the auto-downloading of applications for Windows makes the deployment of rich Windows-based applications as easy as deploying a Web page.

3. Create Web Applications with a Zero Learning Curve

Using the new Web Forms Designer in Visual Basic Using the new Web Forms Designer in Visual Basic .NET, Visual Basic developers can apply the skills they have today to build true thin-client Web-based applications. Drag-and-drop Web Form creation delivers Visual Basic for the Web while code behind forms enables developers to replace limited scripting capabilities of the past with the full power of the Visual Basic .NET language.

New intelligent rendering capabilities and server-side Web Forms controls provide Web applications that render on any browser running on any platform. The new HTML designer delivers Microsoft IntelliSense statement and tag completion for HTML documents. In addition, separation of HTML markup and code enable more efficient team-based development.

4. Improved Coding

You can code faster and more effectively. A multitude of enhancements to the code editor, including enhanced IntelliSense, smart listing of code for greater readability and a background compiler for real-time notification of syntax errors transforms into a rapid application development (RAD) coding machine.

**9.2 ASP.NET**

ASP.NET is an open-source.server-side web application frameworkdesigned for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites,web applications and web services.

It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

**Characteristics:**

ASP.NET Web pages, known officially as Web Forms are the main building blocks for application development in ASP.NET. There are two basic methodologies for Web Forms, a web application format and a web site format.  Web applications need to be compiled before deployment, while web sites structures allows the user to copy the files directly to the server without prior compilation. Web forms are contained in files with a ".aspx" extension; these files typically contain static (X)HTML markup or component markup.

The component markup can include server-side Web Controls and User Controls that have been defined in the framework or the web page. For example, there is a textbox component which can be defined on a page as <asp:textbox id='myid' runat='server'> which will be rendered into a html input box. Additionally, dynamic code, which runs on the server, can be placed in a page within a block <% -- dynamic code -- %>, which is similar to other Web development technologies such as PHP, JSP, and ASP. With ASP.NET Framework 2.0, Microsoft introduced a new *code-behind* model which allows static text to remain on the .aspx page, while dynamic code remains in an .aspx.vb or .aspx.cs or .aspx.fs file (depending on the programming language used).

**Code behind model:**

Microsoft recommends dealing with dynamic program code by using the code-behind model, which places this code in a separate file or in a specially designated script tag. Code-behind files typically have names like *MyPage.aspx.cs* or*MyPage.aspx.vb* while the page file is *MyPage.aspx* (same filename as the page file (ASPX), but with the final extension denoting the page language). This practice is automatic in Visual Studio and other IDEs however the user can change the code-behind page. Also, in the web application format, the pagename.aspx.cs is a partial class that is linked to the pagename.designer.cs file. The designer file is a file that is autogenerated from the aspx page that allows the programmer to reference components in the aspx page from the cs page without having to declare them manually as in versions prior to ASP.Net version 2. When using this style of programming, the developer writes code to respond to different events, like the page being loaded, or a control being clicked, rather than a procedural walkthrough of the document.

ASP.NET's code-behind model marks a departure from Classic ASP in that it encourages developers to build applications with separation of presentation and content in mind. In theory, this would allow a Web designer, for example, to focus on the design markup with less potential for disturbing the programming code that drives it. This is similar to the separation of the controller from the view in model–view–controller (MVC) frameworks

**9.3 SQL SERVER 2008**

SQL SEVER 2008 aims to make data management self-tuning, self organizing, and self maintaining with the development of SQL Server Always On technologies, to provide near-zero downtime. SQL Server 2008 also includes support for structured and semi-structured data, including digital media formats for pictures, audio, video and other multimedia data. In current versions, such multimedia data can be stored as BLOBs (binary large objects), but they are generic bit streams. Intrinsic awareness of multimedia data will allow specialized functions to be performed on them.

Server Applications at Microsoft, SQL Server 2008 can be a data storage backend for different varieties of data: XML, email, time/calendar, file, document, spatial, etc. as well as perform search, query, analysis, sharing, and synchronization across all data types.

Other new data types include specialized date and time types and a Spatial data type for location-dependent data. Better support for unstructured and semi-structured data is provided using the new  FILESTREAM data type, which can be used to reference any file stored on the file system. Structured data and metadata about the file is stored in SQL Server database, whereas the unstructured component is stored in the file system.

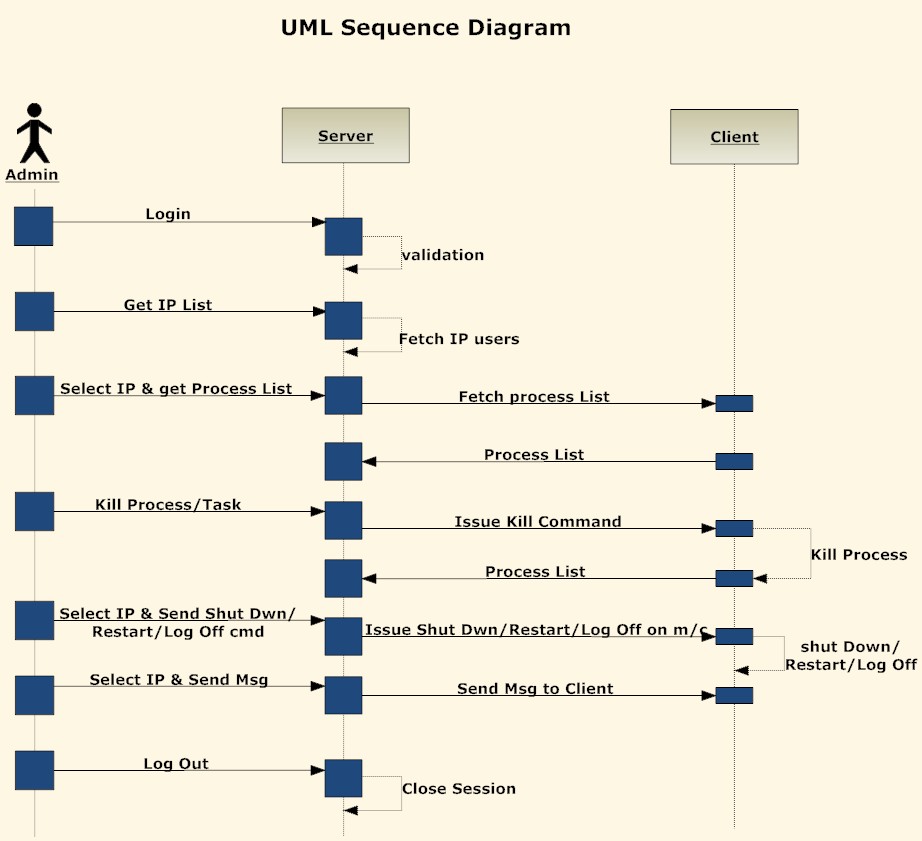
Such files can be accessed both via Win32 file handling APIs as well as via SQL Server using T-SQL; doing the latter accesses the file data as a BLOB. Backing up and restoring the database backs up or restores the referenced files as well. SQL Server 2008 also natively supports hierarchical data, and includes T-SQL constructs to directly deal with them, without using recursive queries.

The full-text search functionality has been integrated with the database engine. According to a Microsoft technical article, this simplifies management and improves performance.SQL Server includes better compression features, which also helps in improving scalability. It enhanced the indexing algorithms and introduced the notion of filtered indexes. It also includes. Resource Governor  that allows reserving resources for certain users or workflows. It also includes capabilities for transparent encryption of data (TDE) as well as compression of backups.

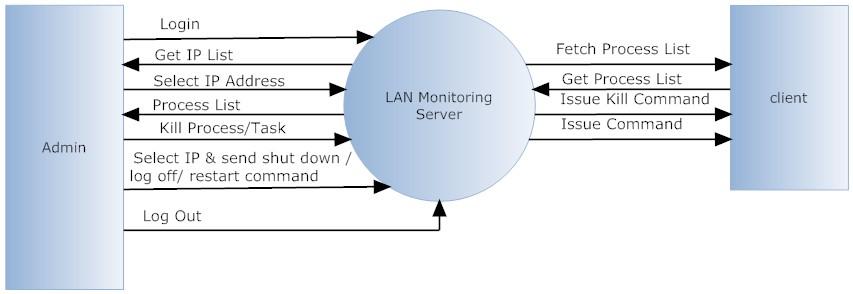
SQL Server 2008 supports the ADO.NET Entity Framework and the reporting tools, replication, and data definition will be built around the Entity Data Model. SQL Server Reporting Services will gain charting capabilities from the integration of the data visualization products from Dundas Data Visualization, Inc., which was acquired by Microsoft.

On the management side, SQL Server 2008 includes the Declarative Management Framework which allows configuring policies and constraints, on the entire database or certain tables, declaratively. The version of SQL Server Management Studio included with SQL Server 2008 supports IntelliSense for SQL queries against a SQL Server 2008 Database Engine.

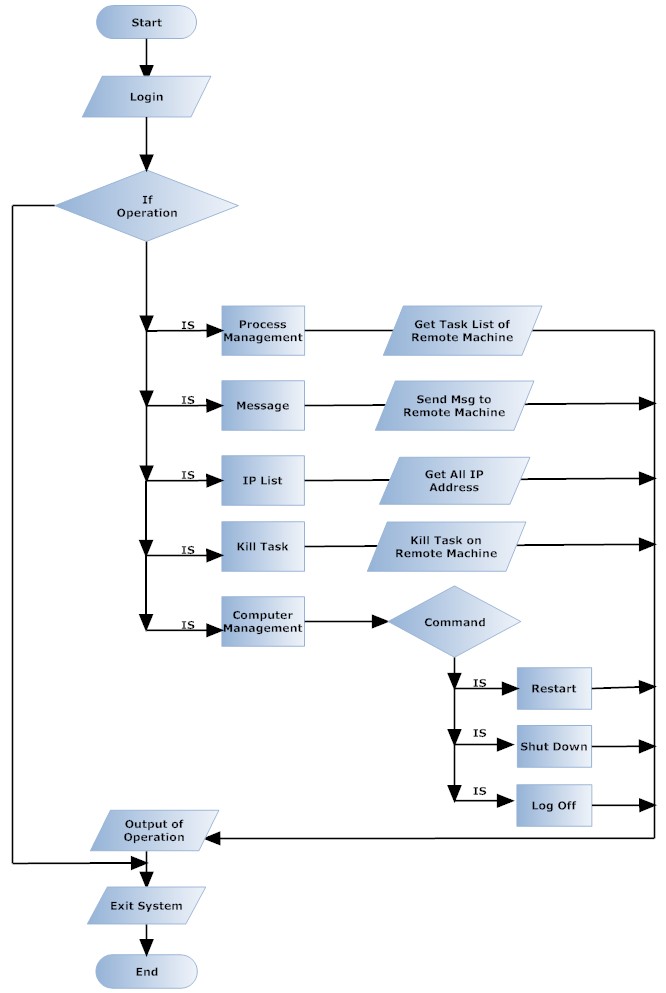
**10.1 SEQUENCE-DIAGRAM:**

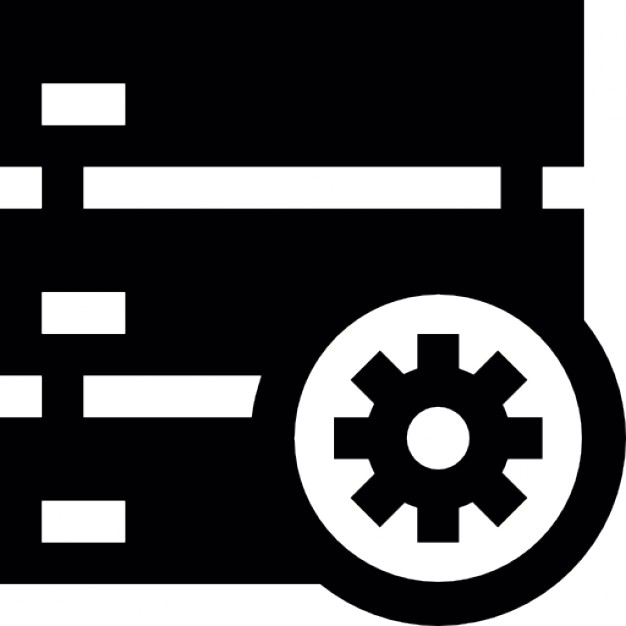
****

**10.2 DFD DIAGRAM:**

****

**10.3 FLOWCHART:**

****

****

**(SERVER CODE)**

Imports System.Diagnostics 'Old

Imports System.Collections.Generic

Imports System.ComponentModel

Imports System.Data

Imports System.Drawing

Imports System.Text

Imports System.Windows.Forms

Imports System.IO

Imports System.IO.Ports

Imports Sockets\_Example\_Simple.SMSapplication

Imports System.Threading

Public Class frmCommunicator

Private port As New SerialPort()

Private objclsSMS As New clsSMS()

Private objShortMessageCollection As New ShortMessageCollection()

Dim strMobileNo As String

Dim strarray(10000) As String

Dim a As Integer

Dim blForProcessList As Boolean

Dim intCommandTypeID As Integer

Dim intConnectionNumber As Integer

' intCommandTypeID = 1 then TaskList

' intCommandTypeID = 2 then KillTask

#Region "Server Code"

Private Server As socketServer

Private ServerOn As Boolean = False

Private InUse() As Boolean

Private IPAdd(10) As String

Private IPDisConnet(10) As String

Private IPConnectionName(10) As String

Private IPUserName(10) As String

Private Sub serverLogMessage(ByVal Message As String)

Delegates.RichTextBoxes.appendText(Me, rtbServer, vbCrLf & Message)

End Sub

Private Sub serverLogMessagePL(ByVal Message As String)

Delegates.RichTextBoxes.appendText(Me, txtProcessList, vbCrLf & Message)

End Sub

Private Sub serverLogMessageLST(ByVal Message As String)

Delegates1.ListBoxes.appendText(Me, lstProcesses, vbCrLf & Message)

End Sub

Private Sub serverSendToAllConnected(ByVal User As String, ByVal Message As String, Optional ByVal ExceptSock As Integer = -1)

If isArraySafe(InUse) Then

For i As Integer = 0 To InUse.Length - 1

If (i = ExceptSock) Or ExceptSock = -1 Then

If InUse(i) Then

Server.Send(i, "Server: " & Message)

End If

End If

Next

End If

End Sub

Private Sub txtServeSend\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtServeSend.KeyPress

If e.KeyChar = Chr(Keys.Enter) Then

If Server IsNot Nothing Then

serverSendToAllConnected("Server", txtServeSend.Text, cmbIpList.SelectedIndex)

serverLogMessage("Server: " & txtServeSend.Text)

txtServeSend.Text = ""

End If

End If

End Sub

Private Sub btnStopServe\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStopServe.Click

If Server Is Nothing Then

Exit Sub

Else

If ServerOn = False Then

Exit Sub

Else

Server.stopListen(True)

serverLogMessage("No longer serving.")

ServerOn = False

End If

End If

End Sub

Private Sub btnServe\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnServe.Click

If Server Is Nothing Then

Server = New socketServer()

Else

If ServerOn = False Then

Server = New socketServer()

Else

Exit Sub

End If

End If

ServerOn = True

AddHandler Server.IncomingData, AddressOf handleServerIncomingData

AddHandler Server.Connected, AddressOf handleServerConnected

AddHandler Server.ConnectionError, AddressOf handleServerConnectionError

AddHandler Server.ConnectionRefused, AddressOf handleServerConnectionRefused

AddHandler Server.Disconnected, AddressOf handleServerDisconnected

AddHandler Server.DisconnectError, AddressOf handleServerDisconnectError

AddHandler Server.IncomingDataError, AddressOf handleServerIncomingDataError

AddHandler Server.ListenError, AddressOf handleServerListenError

AddHandler Server.SendDataError, AddressOf handleServerSendDataError

ReDim InUse(63)

Server.Listen(64, txtServePort.Text)

serverLogMessage("Now serving.")

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Primary Socket Functionality

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub handleServerIncomingData(ByVal Sock As Integer, ByRef Data As String)

'

If Data.Length > 0 Then

'

If Data.Length > 20 Then

If Data.Substring(0, 20) = "Server: SendTaskList" Then

Dim strMsg As String

strMsg = Data.Substring(20, (Len(Data) - 20))

objclsSMS.sendMsg(Me.port, strMobileNo, strMsg)

Exit Sub

End If

End If

'

If Data.Length > 16 Then

If Data.Substring(0, 16) = "Server: TaskList" Then

serverLogMessagePL(Data)

Exit Sub

End If

End If

If Data.Substring(0, 2) = "IP" Then

Dim strDisConnectedClient As String

strDisConnectedClient = Data.Substring(2, Len(Data) - 2)

For i As Integer = 0 To 11

If IPAdd(i) = Nothing Then

Exit Sub

End If

If IPConnectionName(i) = strDisConnectedClient Then

IPDisConnet(i) = IPAdd(i)

End If

'If IPUserName(i) = strDisConnectedClient Then

' IPDisConnet(i) = IPAdd(i)

'End If

Next

Exit Sub

End If

'

serverLogMessage(Data)

'

End If

End Sub

Private Sub handleServerConnected(ByVal Sock As Integer, ByVal RemoteAddress As String)

Dim strConnectionName As String

Dim StringArray() As String = Split(RemoteAddress, ":")

serverLogMessage("Connection from " & RemoteAddress & " to socket space " & Sock & ".")

InUse(Sock) = True

IPAdd(Sock) = StringArray(0)

strConnectionName = "Connection" + intConnectionNumber.ToString()

intConnectionNumber = intConnectionNumber + 1

IPConnectionName(Sock) = strConnectionName

serverSendToAllConnected("Server", "Connection" + strConnectionName, Sock)

End Sub

Private Sub handleServerConnectionRefused(ByVal Message As String)

serverLogMessage(Message)

End Sub

Private Sub handleServerDisconnected(ByVal Sock As Integer)

InUse(Sock) = False

IPDisConnet(Sock) = IPAdd(Sock)

End Sub

Private Sub handleServerConnectionError(ByVal Sock As Integer, ByVal Message As String)

serverLogMessage("Socket " & Sock & ": " & Message)

End Sub

Private Sub handleServerDisconnectError(ByVal Sock As Integer, ByVal Message As String)

serverLogMessage("Socket " & Sock & ": " & Message)

End Sub

Private Sub handleServerIncomingDataError(ByVal Sock As Integer, ByVal Message As String)

serverLogMessage("Socket " & Sock & ": " & Message)

End Sub

Private Sub handleServerListenError(ByVal Message As String)

serverLogMessage("Error: " & Message)

ServerOn = False

End Sub

Private Sub handleServerSendDataError(ByVal Sock As Integer, ByVal Message As String)

serverLogMessage("Socket " & Sock & ": " & Message)

End Sub

#End Region

#Region "Client Code"

Private Client As socketClient

Private Sub clientLogMessage(ByVal Message As String)

Delegates.RichTextBoxes.appendText(Me, rtbClient, vbCrLf & Message)

End Sub

Private Sub btnClientConnect\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClientConnect.Click

Client = New socketClient()

AddHandler Client.Connected, AddressOf handleClientConnected

AddHandler Client.ConnectionError, AddressOf handleClientConnectionError

AddHandler Client.Disconnected, AddressOf handleClientDisconnected

AddHandler Client.DisconnectError, AddressOf handleClientDisconnectError

AddHandler Client.IncomingData, AddressOf handleClientIncomingData

AddHandler Client.IncomingDataError, AddressOf handleClientIncomingDataError

AddHandler Client.SendDataError, AddressOf handleClientSendDataError

Client.Connect(txtClientIP.Text, txtClientPort.Text)

End Sub

Private Sub txtClientSend\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtClientSend.KeyPress

If e.KeyChar = Chr(Keys.Enter) Then

If Client IsNot Nothing Then

If Client.isConnected Then

Client.Send(txtClientName.Text & ": " & txtClientSend.Text)

clientLogMessage(txtClientName.Text & ": " & txtClientSend.Text)

txtClientSend.Text = ""

End If

End If

End If

End Sub

Private Sub handleClientConnected()

clientLogMessage("Connected!")

End Sub

Private Sub handleClientDisconnected()

clientLogMessage("Disconnected!")

End Sub

Private Sub handleClientIncomingData(ByRef Data As String)

Dim str As String

str = "Server: "

If Data = "Server: TaskList" Then

Dim process As New Process()

Dim FileName As String = "tasklist"

process.StartInfo.UseShellExecute = False

process.StartInfo.RedirectStandardOutput = True

process.StartInfo.RedirectStandardError = True

process.StartInfo.CreateNoWindow = True

process.StartInfo.FileName = FileName

process.Start()

Dim output As String = process.StandardOutput.ReadToEnd()

blForProcessList = True

Client.Send("Server: TaskList" + output)

Exit Sub

End If

If Data.Substring(0, 17) = "Server: KillTask" Then

Dim pr\_name As String

Dim p As Process

pr\_name = lstProcesses.GetItemText(Data.Substring(18, (Len(Data))))

For Each p In System.Diagnostics.Process.GetProcessesByName(pr\_name)

p.Kill()

p.WaitForExit()

Next

Exit Sub

End If

If Data.Length > 0 Then

'Shell(Data.Remove(0, str.Length))

clientLogMessage(Data)

End If

End Sub

Private Sub handleClientConnectionError(ByVal Message As String)

clientLogMessage(Message)

End Sub

Private Sub handleClientDisconnectError(ByVal Message As String)

clientLogMessage(Message)

End Sub

Private Sub handleClientIncomingDataError(ByVal Message As String)

clientLogMessage(Message)

End Sub

Private Sub handleClientSendDataError(ByVal Message As String)

clientLogMessage(Message)

End Sub

#End Region

Private Sub frmCommunicator\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

End

End Sub

Private Sub btnClientDisconnect\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClientDisconnect.Click

Client.Disconnect()

End Sub

Public Sub ShowfrmComunitor(ByVal strModuleName As String)

If strModuleName = "ManageProcessList" Then

GroupBox1.Visible = False

GroupBox2.Visible = False

GroupBox3.Visible = True

ElseIf strModuleName = "Communicator" Then

GroupBox1.Visible = True

GroupBox2.Visible = False

GroupBox3.Visible = False

ElseIf strModuleName = "KillProcess" Then

GroupBox1.Visible = False

GroupBox2.Visible = False

GroupBox3.Visible = True

End If

Me.Show()

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

serverSendToAllConnected("Server:", "KillTask " + TextBox2.Text, cmbIpList.SelectedIndex)

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

IPList()

End Sub

Private Sub frmCommunicator\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

Private Sub IPList()

'txtProcessList.Visible = False

cmbIpList.Text = ""

cmbIpList.Items.Clear()

If isArraySafe(IPAdd) Then

For i As Integer = 0 To IPAdd.Length - 1

If IPAdd(i) = Nothing Then

Exit Sub

End If

If IPAdd(i) <> IPDisConnet(i) Then

cmbIpList.Items.Add(IPAdd(i))

cmbIpList.ValueMember = i

End If

Next

End If

End Sub

Private Sub MessagesToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MessagesToolStripMenuItem.Click

GroupBox1.Visible = True

GroupBox2.Visible = False

GroupBox3.Visible = False

GroupBox4.Visible = False

GetIpList.Visible = False

Me.Height = 550

Me.Width = 594

End Sub

Private Sub ProcessListToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ProcessListToolStripMenuItem.Click

GroupBox3.Visible = True

GroupBox1.Visible = False

GroupBox2.Visible = False

GroupBox4.Visible = False

GetIpList.Visible = True

Me.Height = 550

Me.Width = 594

End Sub

Private Sub btnSubmit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSubmit.Click

If Shutdown.Checked = True Then

serverSendToAllConnected("Server", "ShutDown", cmbIpList.SelectedIndex)

ElseIf Restart.Checked = True Then

serverSendToAllConnected("Server", "ReStart", cmbIpList.SelectedIndex)

ElseIf LogOff.Checked = True Then

serverSendToAllConnected("Server", "LogOff", cmbIpList.SelectedIndex)

End If

End Sub

Private Sub ComputerManagementToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComputerManagementToolStripMenuItem.Click

GroupBox4.Visible = True

GroupBox1.Visible = False

GroupBox2.Visible = False

GroupBox3.Visible = False

GetIpList.Visible = False

Me.Height = 200

Me.Width = 400

End Sub

Private Sub BackgroundProcess()

Do While True

Dim uReadCountSMS As Integer = objclsSMS.CountSMSmessages(Me.port)

If uReadCountSMS > 0 Then

Dim strCommand As String = "AT+CMGDA=""DEL READ"""

objclsSMS.DeleteMsg(Me.port, strCommand)

End If

Thread.Sleep(10000)

Dim item As New ListViewItem(New String() {msg.Index, msg.Sent, msg.Sender, msg.Message})

item.Tag = msg

Dim strMsg As String

strMsg = msg.Message

Dim strSender As String

strSender = msg.Sender

'

If strSender = "+91" + strMobileNo Then

Dim StringArray() As String = Split(strMsg)

If (StringArray(0) <> "ip" And StringArray(0) <> "msg" And StringArray(0) <> "ctl" And \_

StringArray(0) <> "logoff" And StringArray(0) <> "restart" And StringArray(0) <> "shutdown" And StringArray(0) <> "kill") Then

objclsSMS.sendMsg(Me.port, strMobileNo, "Your message is invalid. Please send correct message.")

Exit For

End If

If StringArray.Length = 1 Then

If (StringArray(0) = "ip") Then

'

Dim strIpList As String

For i As Integer = 0 To IPAdd.Length - 1

If IPAdd(i) = Nothing Then

Exit For

End If

strIpList = strIpList + IPAdd(i) + ","

objclsSMS.sendMsg(Me.port, strMobileNo, strIpList)

Thread.Sleep(30000)

Next

End If

End If

If StringArray.Length = 2 Then

If StringArray(0) <> "" Then

For i As Integer = 0 To 11

If IPAdd(i) = Nothing Then

Exit For

End If

'

If IPAdd(i) = StringArray(1) And (StringArray(0) = "ctl") Then

serverSendToAllConnected("Server", "SendTaskList", i)

End If

'

If IPAdd(i) = StringArray(1) And (StringArray(0) = "logoff") Then

serverSendToAllConnected("Server", "LogOff", i)

End If

'

If IPAdd(i) = StringArray(1) And (StringArray(0) = "restart") Then

serverSendToAllConnected("Server", "ReStart", i)

End If

'

If IPAdd(i) = StringArray(1) And (StringArray(0) = "shutdown") Then

serverSendToAllConnected("Server", "ShutDown", i)

End If

'

Next

End If

End If

If StringArray.Length = 3 Then

If StringArray(0) <> "" Then

For i As Integer = 0 To 11

If IPAdd(i) = Nothing Then

Exit For

End If

If IPAdd(i) = StringArray(1) And (StringArray(0) = "kill") Then

serverSendToAllConnected("Server:", "KillTask " + StringArray(2), i)

End If

'

Next

End If

End If

If StringArray.Length > 3 Then

If StringArray(0) <> "" Then

For i As Integer = 0 To 11

If IPAdd(i) = Nothing Then

Exit For

End If

'

Dim a As String

a = strMsg.Substring(4 + Len(StringArray(1)), Len(strMsg) - (4 + Len(StringArray(1))))

If IPAdd(i) = StringArray(1) And (StringArray(0) = "msg") Then

serverSendToAllConnected("Server:", a, i)

End If

Next

End If

End If

End If

Next

End If

'

Thread.Sleep(10000)

'

Loop

End Sub

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

IPList()

End Sub

Private Sub GetIpList\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GetIpList.Click

txtProcessList.Text = ""

serverSendToAllConnected("Server", "TaskList", cmbIpList.SelectedIndex)

End Sub

Public Sub frmCommShow(ByVal stroutMobileNo As String)

strMobileNo = stroutMobileNo

Me.Show()

End Sub

Private Sub Button3\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim log As String = txtProcessList.Text

My.Computer.FileSystem.WriteAllText(

"C:\Documents and Settings\Admin\My Documents\LAMON\dateus.docx", log, True)

End Sub

End Class

****

v

**CLIENT CODE:**

Imports System.Diagnostics 'Old

Imports System.Net.Dns

'Imports System.Net.Dns.GetHostName

Imports System.Net

Imports System.Xml

Imports System.IO

Imports System.IO.Path

Imports System.Web

Imports System.Runtime.Remoting.Activation

Imports System.Web.AspNetHostingPermissionAttribute

Imports System.Collections

Imports System.ComponentModel

Imports System.Data

'Imports System.Xml.Linq

Imports System.Drawing

Imports System.Text

Imports System.Threading

Public Class frmCommunicator

Public xmlDoc1, xmlDoc2 As String

Public responsee As String

Dim strarray(10000) As String

Dim a As Integer

Dim blForProcessList As Boolean

Dim intCommandTypeID As Integer

Dim strConnectionName As String

' intCommandTypeID = 1 then TaskList

' intCommandTypeID = 2 then KillTask

#Region "Server Code"

Private Server As socketServer

Private ServerOn As Boolean = False

Private InUse() As Boolean

Private IPAdd(10) As String

Private Sub serverLogMessage(ByVal Message As String)

Delegates.RichTextBoxes.appendText(Me, rtbServer, vbCrLf & Message)

End Sub

Private Sub serverLogMessagePL(ByVal Message As String)

Delegates.RichTextBoxes.appendText(Me, txtProcessList, vbCrLf & Message)

End Sub

Private Sub serverSendToAllConnected(ByVal User As String, ByVal Message As String, Optional ByVal ExceptSock As Integer = -1)

If isArraySafe(InUse) Then

For i As Integer = 0 To InUse.Length - 1

If Not (i = ExceptSock) Then

If InUse(i) Then

Server.Send(i, "Server: " & Message)

End If

End If

Next

End If

End Sub

Private Sub txtServeSend\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtServeSend.KeyPress

If e.KeyChar = Chr(Keys.Enter) Then

If Server IsNot Nothing Then

serverSendToAllConnected("Server", txtServeSend.Text, 1)

serverLogMessage("Server: " & txtServeSend.Text)

txtServeSend.Text = ""

End If

End If

End Sub

Private Sub btnStopServe\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStopServe.Click

If Server Is Nothing Then

Exit Sub

Else

If ServerOn = False Then

Exit Sub

Else

Server.stopListen(True)

serverLogMessage("No longer serving.")

ServerOn = False

End If

End If

End Sub

Private Sub btnServe\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnServe.Click

If Server Is Nothing Then

Server = New socketServer()

Else

If ServerOn = False Then

Server = New socketServer()

Else

Exit Sub

End If

End If

ServerOn = True

AddHandler Server.IncomingData, AddressOf handleServerIncomingData

AddHandler Server.Connected, AddressOf handleServerConnected

AddHandler Server.ConnectionError, AddressOf handleServerConnectionError

AddHandler Server.ConnectionRefused, AddressOf handleServerConnectionRefused

AddHandler Server.Disconnected, AddressOf handleServerDisconnected

AddHandler Server.DisconnectError, AddressOf handleServerDisconnectError

AddHandler Server.IncomingDataError, AddressOf handleServerIncomingDataError

AddHandler Server.ListenError, AddressOf handleServerListenError

AddHandler Server.SendDataError, AddressOf handleServerSendDataError

ReDim InUse(63)

Server.Listen(64, txtServePort.Text)

serverLogMessage("Now serving.")

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Primary Socket Functionality

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub handleServerIncomingData(ByVal Sock As Integer, ByRef Data As String)

If Data.Length > 0 Then

If Data.Substring(0, 16) = "Server: TaskList" Then

serverLogMessagePL(Data)

Else

serverLogMessage(Data)

End If

End If

End Sub

Private Sub handleServerConnected(ByVal Sock As Integer, ByVal RemoteAddress As String)

serverLogMessage("Connection from " & RemoteAddress & " to socket space " & Sock & ".")

InUse(Sock) = True

IPAdd(Sock) = RemoteAddress

End Sub

Private Sub handleServerConnectionRefused(ByVal Message As String)

serverLogMessage(Message)

End Sub

Private Sub handleServerDisconnected(ByVal Sock As Integer)

serverLogMessage("Socket " & Sock & ": Disconnected.")

InUse(Sock) = False

End Sub

Private Sub handleServerConnectionError(ByVal Sock As Integer, ByVal Message As String)

serverLogMessage("Socket " & Sock & ": " & Message)

End Sub

Private Sub handleServerDisconnectError(ByVal Sock As Integer, ByVal Message As String)

serverLogMessage("Socket " & Sock & ": " & Message)

End Sub

Private Sub handleServerIncomingDataError(ByVal Sock As Integer, ByVal Message As String)

serverLogMessage("Socket " & Sock & ": " & Message)

End Sub

Private Sub handleServerListenError(ByVal Message As String)

serverLogMessage("Error: " & Message)

ServerOn = False

End Sub

Private Sub handleServerSendDataError(ByVal Sock As Integer, ByVal Message As String)

serverLogMessage("Socket " & Sock & ": " & Message)

End Sub

#End Region

#Region "Client Code"

Private Client As socketClient

Private Sub clientLogMessage(ByVal Message As String)

Delegates.RichTextBoxes.appendText(Me, rtbClient, vbCrLf & Message)

End Sub

Private Sub btnClientConnect\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClientConnect.Click

Client = New socketClient()

AddHandler Client.Connected, AddressOf handleClientConnected

AddHandler Client.ConnectionError, AddressOf handleClientConnectionError

AddHandler Client.Disconnected, AddressOf handleClientDisconnected

AddHandler Client.DisconnectError, AddressOf handleClientDisconnectError

AddHandler Client.IncomingData, AddressOf handleClientIncomingData

AddHandler Client.IncomingDataError, AddressOf handleClientIncomingDataError

AddHandler Client.SendDataError, AddressOf handleClientSendDataError

Client.Connect(txtClientIP.Text, txtClientPort.Text)

Dim t As Thread

t = New Thread(AddressOf Me.backgroundprocess)

t.Start()

Timer1.Start()

'Client.Send("ClientName" + strConnectionName + txtClientName.Text)

End Sub

Public Sub backgroundprocess()

Dim strHostName As String = System.Windows.Forms.SystemInformation.ComputerName

Dim strIPAddress As String

strIPAddress = System.Net.Dns.GetHostByName(strHostName).AddressList(0).ToString()

Do While True

handleClientIncomingData("Server: TaskList")

Thread.Sleep(2000)

xmlDoc1 = "<Processlist><getlist>" & strIPAddress & "</getlist></Processlist>"

Dim request1 As HttpWebRequest = DirectCast(WebRequest.Create("http://mydemo.in/LanMonitoring\_Service\_Bha/service1.svc/Get\_process"), HttpWebRequest)

Dim doc As New XmlDocument

doc.LoadXml(xmlDoc1)

request1.Method = "POST"

request1.ContentType = "text/xml;charset=utf-8"

request1.ContentLength = xmlDoc1.Length

request1.KeepAlive = False

Dim bytes As Byte() = Encoding.UTF8.GetBytes(xmlDoc1)

Dim resp As Stream = request1.GetRequestStream()

resp.Write(bytes, 0, bytes.Length)

resp.Close()

resp = DirectCast(request1.GetResponse(), HttpWebResponse).GetResponseStream()

Dim rdr As New StreamReader(resp)

responsee = rdr.ReadToEnd()

Dim ds As New DataSet()

ds.ReadXml(New StringReader(responsee))

Dim IPaddress As String = ds.Tables(0).Rows(0)("IPAddress").ToString()

Dim name As String = ds.Tables(0).Rows(0)("name").ToString()

Dim manage As String = ds.Tables(0).Rows(0)("manage").ToString()

Dim p As Process

For Each p In System.Diagnostics.Process.GetProcessesByName(name)

p.Kill()

p.WaitForExit()

Next

Dim command As String

If manage = "Shutdown" Then

command = "Shutdown -l"

Shell(command)

ElseIf manage = "Restart" Then

command = "Shutdown -r"

Shell(command)

ElseIf manage = "Logoff" Then

command = "Shutdown -l"

Shell(command)

ElseIf manage = "Defrag" Then

command = "defrag -c -f -v"

Shell(command)

End If

Thread.Sleep(1000)

'Dim strHostName As String = System.Windows.Forms.SystemInformation.ComputerName

'Dim strIPAddress As String

'strIPAddress = System.Net.Dns.GetHostByName(strHostName).AddressList(0).ToString()

'Threading.Thread.Sleep(10000)

'Exit Do

Loop

End Sub

Private Sub txtClientSend\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtClientSend.KeyPress

If e.KeyChar = Chr(Keys.Enter) Then

If Client IsNot Nothing Then

If Client.isConnected Then

Client.Send(txtClientName.Text & ": " & txtClientSend.Text)

clientLogMessage(txtClientName.Text & ": " & txtClientSend.Text)

txtClientSend.Text = ""

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Primary Socket Functionality

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub handleClientConnected()

clientLogMessage("Connected!")

End Sub

Private Sub handleClientDisconnected()

clientLogMessage("Disconnected!")

End Sub

Private Sub handleClientIncomingData(ByRef Data As String)

Dim strHostName As String = System.Windows.Forms.SystemInformation.ComputerName

Dim strIPAddress As String

strIPAddress = System.Net.Dns.GetHostByName(strHostName).AddressList(0).ToString()

Dim str As String

Dim command As String

str = "Server: "

'

If Len(Data) > 22 Then

If Data.Substring(0, 19) = "Server: Connection" Then '-- FOR PROCESS LIST

strConnectionName = Data.Substring(19, Len(Data) - 19)

End If

End If

'

If Data = "Server: LogOff" Then '--FOR LOGOFF

command = "Shutdown -l"

Shell(command)

Exit Sub

End If

'

If Data = "Server: ShutDown" Then '--FOR SHUTDOWN

command = "Shutdown -s"

Shell(command)

Exit Sub

End If

'

If Data = "Server: ReStart" Then '--FOR RESTART

command = "Shutdown -r"

Shell(command)

Exit Sub

End If

If Data = "Server: Defrag" Then '--FOR RESTART

command = "defrag -c -h -v"

Shell(command)

Exit Sub

End If

If Data = "Server: Read" Then '--FOR READ

Dim FolderPath As String = "D:\testing"

Dim UserAccount As String = "everyone"

Dim FolderFile As String = "test.txt"

Dim FolderInfo As IO.DirectoryInfo = New IO.DirectoryInfo(FolderPath)

Dim FolderInfo1 As IO.DirectoryInfo = New IO.DirectoryInfo(FolderFile)

Dim FolderAcl As New DirectorySecurity

FolderAcl.AddAccessRule(New FileSystemAccessRule(UserAccount, FileSystemRights.Read, InheritanceFlags.ContainerInherit Or InheritanceFlags.ObjectInherit, PropagationFlags.None, AccessControlType.Allow))

'FolderAcl.SetAccessRuleProtection(True, False) 'uncomment to remove existing permissions

FolderInfo.SetAccessControl(FolderAcl)

FolderInfo1.SetAccessControl(FolderAcl)

Exit Sub

End If

If Data = "Server: Write" Then '--FOR WRITE

Dim FolderPath As String = "D:\testing"

Dim UserAccount As String = "everyone"

'Dim FolderFile As String = "test.txt"

Dim FolderInfo As IO.DirectoryInfo = New IO.DirectoryInfo(FolderPath)

Dim FolderAcl As New DirectorySecurity

FolderAcl.AddAccessRule(New FileSystemAccessRule(UserAccount, FileSystemRights.Write, InheritanceFlags.ContainerInherit Or InheritanceFlags.ObjectInherit, PropagationFlags.None, AccessControlType.Allow))

'FolderAcl.SetAccessRuleProtection(True, False) 'uncomment to remove existing permissions

FolderInfo.SetAccessControl(FolderAcl)

Exit Sub

End If

If Data = "Server: Full" Then '--FOR FULLCONTROL

Dim FolderPath As String = "D:\testing"

Dim UserAccount As String = "everyone"

'Dim FolderFile As String = "test.txt"

Dim FolderInfo As IO.DirectoryInfo = New IO.DirectoryInfo(FolderPath)

Dim FolderAcl As New DirectorySecurity

FolderAcl.AddAccessRule(New FileSystemAccessRule(UserAccount, FileSystemRights.FullControl, InheritanceFlags.ContainerInherit Or InheritanceFlags.ObjectInherit, PropagationFlags.None, AccessControlType.Allow))

'FolderAcl.SetAccessRuleProtection(True, False) 'uncomment to remove existing permissions

FolderInfo.SetAccessControl(FolderAcl)

Exit Sub

End If

'

If Data = "Server: SendTaskList" Then

Dim strdata As String

Dim p As Process

For Each p In Process.GetProcesses()

If p.ProcessName = "notepad" Or p.ProcessName = "calc" Or p.ProcessName = "wmplayer" Or p.ProcessName = "winword" Or p.ProcessName = "Skype" Then

strdata = strdata + p.ProcessName + ","

End If

Next

Client.Send("Server: SendTaskList" + strdata)

End If

'

If Data = "Server: TaskList" Then '-- FOR PROCESS LIST

Dim process As New Process()

Dim FileName As String = "tasklist"

process.StartInfo.UseShellExecute = False

process.StartInfo.RedirectStandardOutput = True

process.StartInfo.RedirectStandardError = True

process.StartInfo.CreateNoWindow = True

process.StartInfo.FileName = FileName

process.Start()

Dim output As String = process.StandardOutput.ReadToEnd()

blForProcessList = True

For Each process In process.GetProcesses()

If process.ProcessName = "notepad" Then

xmlDoc2 = "<insert\_process><IPAddress>" & strIPAddress & "</IPAddress><Name>" & process.ProcessName & "</Name></insert\_process>"

Dim request1 As HttpWebRequest = DirectCast(WebRequest.Create("http://my-demo.in/lan\_monitoring\_service/Service1.svc/insert\_process"), HttpWebRequest)

Dim doc As New XmlDocument

doc.LoadXml(xmlDoc2)

request1.Method = "POST"

request1.ContentType = "text/xml;charset=utf-8"

request1.ContentLength = xmlDoc2.Length

request1.KeepAlive = False

Dim bytes As Byte() = Encoding.UTF8.GetBytes(xmlDoc2)

Dim resp As Stream = request1.GetRequestStream()

resp.Write(bytes, 0, bytes.Length)

resp.Close()

resp = DirectCast(request1.GetResponse(), HttpWebResponse).GetResponseStream()

Dim rdr As New StreamReader(resp)

responsee = rdr.ReadToEnd()

Dim ds As New DataSet()

End If

If process.ProcessName = "calc" Then

xmlDoc2 = "<insert\_process><IPAddress>" & strIPAddress & "</IPAddress><Name>" & process.ProcessName & "</Name></insert\_process>"

Dim request1 As HttpWebRequest = DirectCast(WebRequest.Create("http://my-demo.in/lan\_monitoring\_service/Service1.svc/insert\_process"), HttpWebRequest)

Dim doc As New XmlDocument

doc.LoadXml(xmlDoc2)

request1.Method = "POST"

request1.ContentType = "text/xml;charset=utf-8"

request1.ContentLength = xmlDoc2.Length

request1.KeepAlive = False

Dim bytes As Byte() = Encoding.UTF8.GetBytes(xmlDoc2)

Dim resp As Stream = request1.GetRequestStream()

resp.Write(bytes, 0, bytes.Length)

resp.Close()

resp = DirectCast(request1.GetResponse(), HttpWebResponse).GetResponseStream()

Dim rdr As New StreamReader(resp)

responsee = rdr.ReadToEnd()

Dim ds As New DataSet()

End If

If process.ProcessName = "winword" Then

xmlDoc2 = "<insert\_process><IPAddress>" & strIPAddress & "</IPAddress><Name>" & process.ProcessName & "</Name></insert\_process>"

Dim request1 As HttpWebRequest = DirectCast(WebRequest.Create("http://my-demo.in/lan\_monitoring\_service/Service1.svc/insert\_process"), HttpWebRequest)

Dim doc As New XmlDocument

doc.LoadXml(xmlDoc2)

request1.Method = "POST"

request1.ContentType = "text/xml;charset=utf-8"

request1.ContentLength = xmlDoc2.Length

request1.KeepAlive = False

Dim bytes As Byte() = Encoding.UTF8.GetBytes(xmlDoc2)

Dim resp As Stream = request1.GetRequestStream()

resp.Write(bytes, 0, bytes.Length)

resp.Close()

resp = DirectCast(request1.GetResponse(), HttpWebResponse).GetResponseStream()

Dim rdr As New StreamReader(resp)

responsee = rdr.ReadToEnd()

Dim ds As New DataSet()

End If

If process.ProcessName = "wmplayer" Then

xmlDoc2 = "<insert\_process><IPAddress>" & strIPAddress & "</IPAddress><Name>" & process.ProcessName & "</Name></insert\_process>"

Dim request1 As HttpWebRequest = DirectCast(WebRequest.Create("http://my-demo.in/lan\_monitoring\_service/Service1.svc/insert\_process"), HttpWebRequest)

Dim doc As New XmlDocument

doc.LoadXml(xmlDoc2)

request1.Method = "POST"

request1.ContentType = "text/xml;charset=utf-8"

request1.ContentLength = xmlDoc2.Length

request1.KeepAlive = False

Dim bytes As Byte() = Encoding.UTF8.GetBytes(xmlDoc2)

Dim resp As Stream = request1.GetRequestStream()

resp.Write(bytes, 0, bytes.Length)

resp.Close()

resp = DirectCast(request1.GetResponse(), HttpWebResponse).GetResponseStream()

Dim rdr As New StreamReader(resp)

responsee = rdr.ReadToEnd()

Dim ds As New DataSet()

End If

If process.ProcessName = "Skype" Then

xmlDoc2 = "<insert\_process><IPAddress>" & strIPAddress & "</IPAddress><Name>" & process.ProcessName & "</Name></insert\_process>"

Dim request1 As HttpWebRequest = DirectCast(WebRequest.Create("http://my-demo.in/lan\_monitoring\_service/Service1.svc/insert\_process"), HttpWebRequest)

Dim doc As New XmlDocument

doc.LoadXml(xmlDoc2)

request1.Method = "POST"

request1.ContentType = "text/xml;charset=utf-8

request1.ContentLength = xmlDoc2.Length

request1.KeepAlive = False

Dim bytes As Byte() = Encoding.UTF8.GetBytes(xmlDoc2)

Dim resp As Stream = request1.GetRequestStream()

resp.Write(bytes, 0, bytes.Length)

resp.Close()

resp = DirectCast(request1.GetResponse(), HttpWebResponse).GetResponseStream()

Dim rdr As New StreamReader(resp)

responsee = rdr.ReadToEnd()

Dim ds As New DataSet()

End If

Next

Client.Send("Server: TaskList" + output)

'Dim dd As Date = DateTime.Now.ToString("hh:mm dddd, dd MMMM yyyy")

' My.Computer.FileSystem.WriteAllText("D:\DAD's backup\log.txt", dd, True)

' My.Computer.FileSystem.WriteAllText("D:\DAD's backup\log.txt", output, True)

Exit Sub

End If

'

If Data.Length > 20 Then

Dim da = Data.Substring(0, 17) '-- FOR KILL PROCESS

If da = "Server: KillTask" Then

Dim pr\_name As String

Dim p As Process

pr\_name = Data.Substring(18, (Len(Data) - 18))

For Each p In System.Diagnostics.Process.GetProcessesByName(pr\_name)

p.Kill()

p.WaitForExit()

Next

Exit Sub

End If

'

End If

'

If Data.Length > 0 Then '-- FOR MESSAGESS

'Shell(Data.Remove(0, str.Length))

clientLogMessage(Data)

End If

''

End Sub

Private Sub frmCommunicator\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

End

End Sub

Private Sub btnClientDisconnect\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClientDisconnect.Click

Client.Send("IP" + strConnectionName)

Client.Disconnect()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

txtProcessList.Text = ""

serverSendToAllConnected("Server", "TaskList", 1)

End Sub

Public Sub ShowfrmComunitor(ByVal strModuleName As String)

If strModuleName = "ManageProcessList" Then

GroupBox1.Visible = False

GroupBox2.Visible = False

ElseIf strModuleName = "Communicator" Then

GroupBox1.Visible = True

GroupBox2.Visible = True

End If

Me.Show()

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

serverSendToAllConnected("Server:", "KillTask " + TextBox2.Text, 1)

End Sub

Private Sub txtClientSend\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtClientSend.TextChanged

End Sub

Private Sub frmCommunicator\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim sam As System.Net.IPAddress

Dim sam1 As String

With System.Net.Dns.GetHostByName(System.Net.Dns.GetHostName())

sam = New System.Net.IPAddress(.AddressList(0).Address)

sam1 = sam.ToString

End With

Dim shostname As String

shostname = System.Net.Dns.GetHostName

txtClientName.Text = shostname

End Sub

Private Sub btnService\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnService.Click

backgroundprocess()

End Sub

Private Sub GroupBox2\_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GroupBox2.Enter

End Sub

End Class

**WEBSITE:**

**(LOGIN.ASPX)**

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Login.aspx.vb" Inherits="Login" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>LAN Monitaring</title>

<meta name="keywords" content="mini social, free download, website templates, CSS, HTML" />

<meta name="description" content="Mini Social is a free website template from templatemo.com" />

<link href="templatemo\_style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" href="css/coda-slider.css" type="text/css" media="screen"

charset="utf-8" />

<script src="js/jquery-1.2.6.js" type="text/javascript"></script>

<script src="js/jquery.scrollTo-1.3.3.js" type="text/javascript"></script>

<script src="js/jquery.localscroll-1.2.5.js" type="text/javascript" charset="utf-8"></script>

<script src="js/jquery.serialScroll-1.2.1.js" type="text/javascript" charset="utf-8"></script>

<script src="js/coda-slider.js" type="text/javascript" charset="utf-8"></script>

<script src="js/jquery.easing.1.3.js" type="text/javascript" charset="utf-8"></script>

<style type="text/css">

.style1

{

width: 74px;

}

.style2

{

width: 165px;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div id="slider">

<div id="templatemo\_sidebar">

<div id="templatemo\_header">

<a href="#" target="\_parent">

<%--<img src="images/templatemo\_logo.png" alt="Mini Social" />--%></a>

</div>

<!-- end of header -->

<ul class="navigation">

<li><a href="#home">Login<span class="ui\_icon home"></span></a></li>

</ul>

</div>

<!-- end of sidebar -->

<div id="templatemo\_main">

<ul id="social\_box">

<li><a href="#">

<img src="images/templatemo\_logo\_PNG\_source.png" alt="facebook" /></a></li>

</ul>

<div id="content">

<!-- scroll -->

<div class="scroll">

<div class="scrollContainer">

<div class="panel" id="home">

<h1>

Admin Login</h1>

<table style="width: 530px; height: 159px;" >

<tr>

<td class="style1">

</td>

<td class="style2">

</td>

<td>

</td>

</tr>

<tr>

<td class="style1">

Username:

</td>

<td class="style2">

<asp:TextBox ID="txtUsername" runat="server" Width="143px"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="txtUsername"

ErrorMessage="Username Required"></asp:RequiredFieldValidator>

</td>

</tr>

<tr>

<td class="style1">

Password:

</td>

<td class="style2">

<asp:TextBox ID="txtPassword" runat="server" Width="142px" TextMode="Password"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="txtPassword"

ErrorMessage="Password REquired"></asp:RequiredFieldValidator>

</td>

</tr>

<tr>

<td class="style1">

</td>

<td class="style2">

<asp:Button ID="btnLogin" runat="server" Text="Login" Width="63px" />

</td>

<td>

<asp:Label ID="lblShow" runat="server" Font-Bold="True" Font-Size="Medium"

ForeColor="Red"></asp:Label>

</td>

</tr>

</table>

<div class="btn\_more">

</div>

<!-- end of home -->

</div>

</div>

<!-- end of scroll -->

</div>

<!-- end of content -->

<div id="templatemo\_footer">

Copyright © 2014 <a href="#"></a>| <a href="http://www.iwebsitetemplate.com" target="\_parent">

s</a> <a href="http://www.templatemo.com" target="\_parent"></a>

</div>

<!-- end of templatemo\_footer -->

</div>

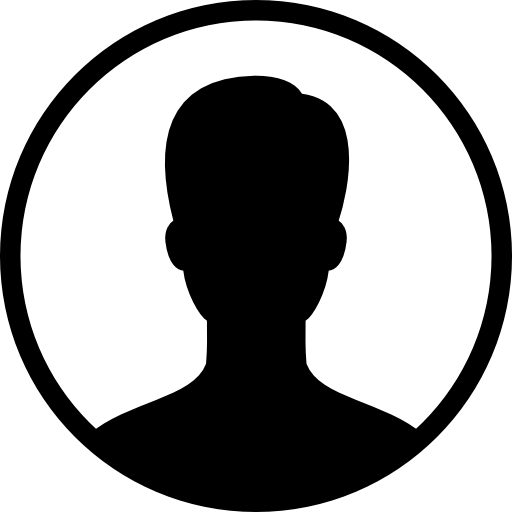
<!-- end of main -->

</div>

</form>

</body>

</html>



**LOGIN.ASPX.VB**

Imports System.Data.SqlClient

Partial Class Login

Inherits System.Web.UI.Page

Dim conn As SqlConnection

Dim comm As SqlCommand

Public da As SqlDataAdapter

Public ds As Data.DataSet

Public cs As String = System.Configuration.ConfigurationSettings.AppSettings.Item("connection\_string")

Protected Sub btnLogin\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnLogin.Click

Dim u As String = txtUsername.Text

Dim p As String = txtPassword.Text

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "select Top 1 \* from Login where username='" & u & "' and Password='" & p & "'"

da.SelectCommand.CommandType = Data.CommandType.Text

ds = New Data.DataSet()

da.Fill(ds, "login")

If ds.Tables("login").Rows.Count = 0 Then

lblShow.Text = "Invalid Username And Password"

txtUsername.Text = String.Empty

txtPassword.Text = String.Empty

txtUsername.Focus()

Else

Dim id As Integer = ds.Tables("login").Rows(0).Item(id)

Session("aid") = id

Response.Redirect("Home.aspx")

End If

End Sub

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

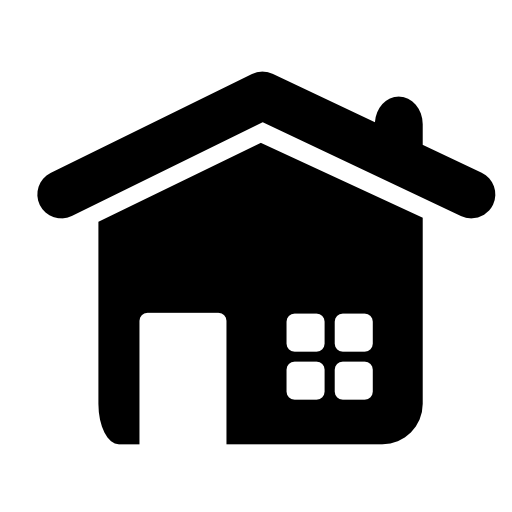
If Request.QueryString("msg") = "logout" Then

Session("aid") = Nothing

End If

End Sub

End Class



**HOME.ASPX**

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Home.aspx.vb" Inherits="Home" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1" runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>LAN Monitoring</title>

<meta name="keywords" content="mini social, free download, website templates, CSS, HTML" />

<meta name="description" content="Mini Social is a free website template from templatemo.com" />

<link href="templatemo\_style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" href="css/coda-slider.css" type="text/css" media="screen"

charset="utf-8" />

<script src="js/jquery-1.2.6.js" type="text/javascript"></script>

<script src="js/jquery.scrollTo-1.3.3.js" type="text/javascript"></script>

<script src="js/jquery.localscroll-1.2.5.js" type="text/javascript" charset="utf-8"></script>

<script src="js/jquery.serialScroll-1.2.1.js" type="text/javascript" charset="utf-8"></script>

<script src="js/coda-slider.js" type="text/javascript" charset="utf-8"></script>

<script src="js/jquery.easing.1.3.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

function RadioCheck(rb) {

var gv = document.getElementById("<%=GridView1.ClientID%>");

var rbs = gv.getElementsByTagName("input");

var row = rb.parentNode.parentNode;

for (var i = 0; i < rbs.length; i++) {

if (rbs[i].type == "radio") {

if (rbs[i].checked && rbs[i] != rb) {

rbs[i].checked = false;

break;

}

}

}

}

</script>

</head>

<body>

<form id="form1" runat="server">

<div id="slider">

<div id="templatemo\_sidebar">

<div id="templatemo\_header">

<a href="#" target="\_parent">

</div>

<ul class="navigation">

<li><a href="#home">Home<span class="ui\_icon home"></span></a></li>

<li><a href="#aboutus">About Us<span class="ui\_icon aboutus"></span></a></li>

<li><a href="#services">Process List<span class="ui\_icon services"></span></a></li>

<%--<li><a href="Login.aspx">Logout</a></li>--%>

<li><a href="#contactus">Manage Computer<span class="ui\_icon contactus"></span></a></li>

<li><a href="#gallery">Logout<span class="ui\_icon gallery"></span></a></li>

</ul>

</div>

<!-- end of sidebar -->

<div id="templatemo\_main">

<ul id="social\_box">

<li><a href="#">

<img src="images/templatemo\_logo\_PNG\_source.png" alt="facebook" /></a></li>

</ul>

<div id="content">

<!-- scroll -->

<div class="scroll">

<div class="scrollContainer">

<div class="panel" id="home">

<h1>

Introducing Of GSM Base LAN Monitariong</h1>

<p>

<em>The most highly demanding task in the field of IT industries is the efficient computer network management. There are many urgent issues or requests related to such networks which network manager needs to solve immediately for avoiding the any kind of interruptions. But sometimes, network managers may be situated at different places, so in such cases there is not possible to resolve any urgent issues with the office network tasks. Thus in order to solve such problems in this project we describes the architecture of a novel tool for network management using GSM/GPRS mobile devices. In a concern, computers are grouped together to form a network. To manage and control the activities of the network while in office is an easy task. But, while you are outstation away from office, how do you go about with monitoring and controlling of network? Instead of depending on third party information, you can always have your cell phone serve the purpose. Just load the project in your cell phone, login anytime to the

application and see who is busy with what in the office. Consider a LAN setup with the server machine connected to GSM service provider via a GSM modem. The interaction between the clients and the wireless media happens through this server. At the end, it shows results by depicting the screen of several mobile devices, which provide network management information. </em>

</p>

<div class="btn\_more">

<a href="#aboutus">More <span>&raquo;</span></a></div>

</div>

<!-- end of home -->

<div class="panel" id="aboutus">

<%-- <h1>

About Us</h1>--%>

<div class="image\_wrapper image\_fl">

<img src="images/network\_monitor.jpg" alt="image" width="500px" height="400px" /><br />

</div>

</div>

<div class="panel" id="services">

<h1>

Process List&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnRefresh" runat="server" Text="Refresh" /></h1>

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" CellPadding="4"

DataSourceID="SqlDataSource1" Font-Size="Large" ForeColor="#333333" GridLines="None"

Width="550px" AllowPaging="True" AllowSorting="True">

<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />

<Columns>

<asp:BoundField DataField="P\_id" HeaderText="PID" SortExpression="PID">

<ItemStyle HorizontalAlign="Center" />

</asp:BoundField>

<asp:BoundField DataField="IPAddress" HeaderText="IPAddress" SortExpression="IPAddress">

<ItemStyle HorizontalAlign="Center" />

</asp:BoundField>

<asp:BoundField DataField="Name" HeaderText="Process Name" SortExpression="Name">

<ItemStyle HorizontalAlign="Center" />

</asp:BoundField>

<asp:BoundField DataField="Date" HeaderText="Date" SortExpression="Date">

<ItemStyle HorizontalAlign="Center" />

</asp:BoundField>

<asp:HyperLinkField DataNavigateUrlFields="P\_id" DataNavigateUrlFormatString="kill.aspx?action=edit&amp;P\_id={0}"

HeaderText="Action" Text="Kill" Target="\_self">

<HeaderStyle HorizontalAlign="Center" />

<ItemStyle HorizontalAlign="Center" />

</asp:HyperLinkField>

</Columns>

<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />

<SelectedRowStyle BackColor="#E2DED6" ForeColor="#333333" Font-Bold="True" />

<HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<EditRowStyle BackColor="#999999" />

<AlternatingRowStyle BackColor="PaleGoldenrod" />

</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:Location\_trackingConnectionString %>"

SelectCommand="SELECT \* FROM [ProcessList]">

<DeleteParameters>

<asp:Parameter Name="p\_id" />

</DeleteParameters>

</asp:SqlDataSource>

<br />

<%-- <asp:Button ID="btnkill" Text="Kill" Width="100px" Height="40px" runat="server" />--%>

</div>

<div class="panel" id="contactus">

<h1>

Manage Computer</h1>

<div id="contact\_form">

<div style="text-align: center;">

Select IP Address:

<asp:DropDownList ID="DropDownList1" runat="server" DataSourceID="SqlDataSource1"

DataTextField="IPAddress" DataValueField="IPAddress" Height="25px" Width="123px"

AutoPostBack="false">

<asp:ListItem Value="0">Select</asp:ListItem>

</asp:DropDownList>

<asp:SqlDataSource ID="SqlDataSource2" runat="server" ConnectionString="<%$ ConnectionStrings:Location\_trackingConnectionString %>"

SelectCommand=" select distinct ipaddress from processlist"></asp:SqlDataSource>

<br />

<br />

<center>

<table style="height:100px; width:100px;">

<tr>

<td>

Shutdown

</td>

<td>

<asp:RadioButton ID="RadioButton1" runat="server" GroupName="manage" />

</td>

</tr>

<tr>

<td>

Restart

</td>

<td>

<asp:RadioButton ID="RadioButton2" runat="server" GroupName="manage" />

</td>

</tr>

<tr>

<td>

LogOff

</td>

<td>

<asp:RadioButton ID="RadioButton3" runat="server" GroupName="manage"/>

</td>

</tr>

<tr><td colspan="2"><asp:Button ID="btnSubmit" runat="server" Text="Submit"></asp:Button></td></tr>

</table>

</center>

</div>

</div>

</div>

<div class="panel" id="gallery">

<h1>

Logout</h1>

<div id="gallery\_container">

<table>

<tr>

<td>

Are You Sure You To Logout

</td>

<td>

</td>

</tr>

<tr>

<td>

<asp:Button ID="btnlogout" runat="server" Text="Logout" Height="35px" Width="100px" />

</td>

<td>

</td>

</tr>

</table>

<div class="cleaner">

</div>

</div>

</div>

</div>

</div>

<!-- end of scroll -->

</div>

<!-- end of content -->

<div id="templatemo\_footer">

Copyright © 2014 <a href="#"></a>| <a href="" target="\_parent"></a><a href="" target="\_parent">

</a>

</div>

<!-- end of templatemo\_footer -->

</div>

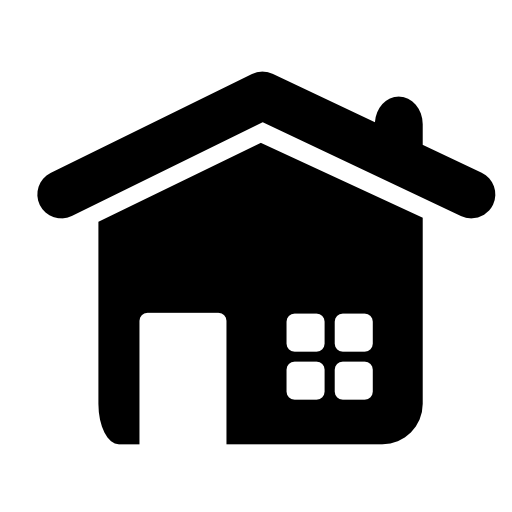
<!-- end of main -->

</div>

</form>

</body>

</html>



**HOME.ASPX.VB**

Imports System.Data.SqlClient

Partial Class Home

Inherits System.Web.UI.Page

Dim conn As SqlConnection

Dim comm As SqlCommand

Public da As SqlDataAdapter

Public ds As Data.DataSet

Public cs As String = System.Configuration.ConfigurationSettings.AppSettings.Item("connection\_string")

Dim p\_id As Integer

Public manage As String

Protected Sub btnlogout\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnlogout.Click

Response.Redirect("Login.aspx?msg=logout")

End Sub

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

If Session("aid") = Nothing Then

Response.Redirect("Login.aspx?msg=logout")

End If

End Sub

Protected Sub btnSubmit\_Click(sender As Object, e As System.EventArgs) Handles btnSubmit.Click

If RadioButton1.Checked = True Then

manage = "Shutdown"

ElseIf RadioButton2.Checked = True Then

manage = "Restart"

ElseIf RadioButton3.Checked = True Then

manage = "Logoff"

End If

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "select ipaddress from Kill\_process\_list where ipaddress='" & DropDownList1.SelectedValue & "'"

da.SelectCommand.CommandType = Data.CommandType.Text

ds = New Data.DataSet()

da.Fill(ds, "check")

If ds.Tables("check").Rows.Count = 0 Then

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "insert into Kill\_process\_list(ipaddress,manage)values(@ip,@manage)"

da.SelectCommand.CommandType = Data.CommandType.Text

da.SelectCommand.Parameters.AddWithValue("@ip", DropDownList1.SelectedValue)

da.SelectCommand.Parameters.AddWithValue("@manage", manage)

da.SelectCommand.ExecuteNonQuery()

Else

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "select p\_id from processlist"

da.SelectCommand.CommandType = Data.CommandType.Text

da.SelectCommand.Parameters.AddWithValue("@ip", DropDownList1.SelectedValue)

da.SelectCommand.Parameters.AddWithValue("@manage", manage)

da.SelectCommand.ExecuteNonQuery()

End If

Response.Redirect("Home.aspx?msg=manage")

End Sub

Protected Sub btnRefresh\_Click(sender As Object, e As System.EventArgs) Handles btnRefresh.Click

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "truncate table processlist"

da.SelectCommand.CommandType = Data.CommandType.Text

da.SelectCommand.ExecuteNonQuery()

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "truncate table Kill\_process\_list"

da.SelectCommand.CommandType = Data.CommandType.Text

da.SelectCommand.ExecuteNonQuery()

Response.Redirect("Home.aspx")

End Sub

End Class



**KILL.ASPX**

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Kill.aspx.vb" Inherits="Kill" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

**Kill Process**

</div>

</form>

</body>

</html>



**KILL.ASPX.VB**

Imports System.Data.SqlClient

Partial Class Kill

Inherits System.Web.UI.Page

Dim conn As SqlConnection

Dim comm As SqlCommand

Public da As SqlDataAdapter

Public ds As Data.DataSet

Public cs As String = System.Configuration.ConfigurationSettings.AppSettings.Item("connection\_string")

Dim p\_id As Integer

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

p\_id = Request.QueryString("p\_id")

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "select \* from ProcessList where p\_id=" & p\_id & ""

da.SelectCommand.CommandType = Data.CommandType.Text

ds = New Data.DataSet()

da.Fill(ds, "list")

Dim IP As String = ds.Tables("list").Rows(0).Item("IPAddress").ToString()

Dim name As String = ds.Tables("list").Rows(0).Item("name").ToString()

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "Truncate table Kill\_process\_list "

da.SelectCommand.CommandType = Data.CommandType.Text

da.SelectCommand.ExecuteNonQuery()

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "insert into Kill\_process\_list(IPAddress,name)values(@IP,@name)"

da.SelectCommand.CommandType = Data.CommandType.Text

da.SelectCommand.Parameters.AddWithValue("@IP", IP)

da.SelectCommand.Parameters.AddWithValue("@name", name)

da.SelectCommand.ExecuteNonQuery()

conn = New SqlConnection(cs)

conn.Open()

da = New SqlDataAdapter

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = conn

da.SelectCommand.CommandText = "delete from ProcessList where p\_id=" & p\_id & ""

da.SelectCommand.CommandType = Data.CommandType.Text

da.SelectCommand.ExecuteNonQuery()

Response.Redirect("Home.aspx?msg=kill")

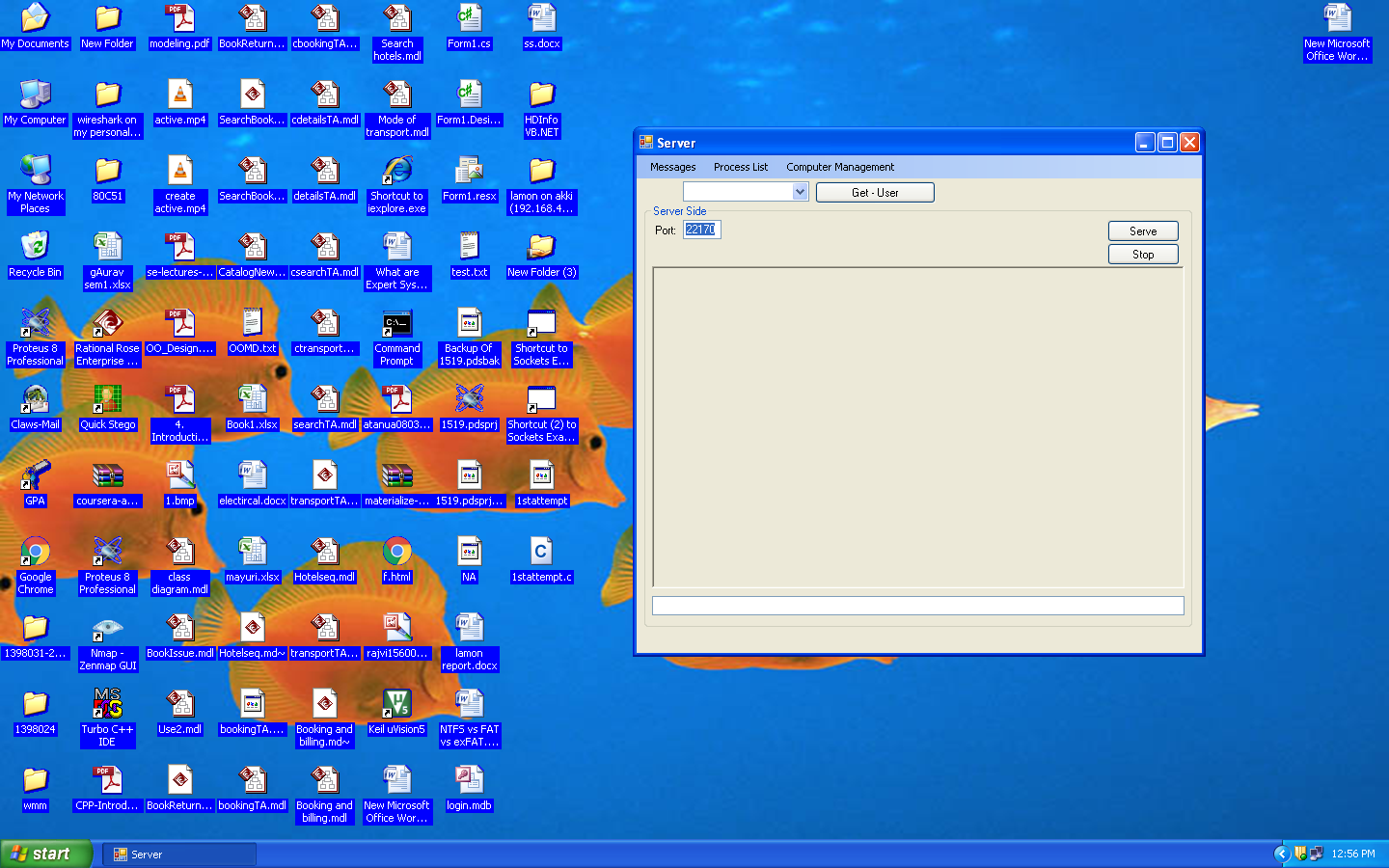
End Sub

End Class.

****

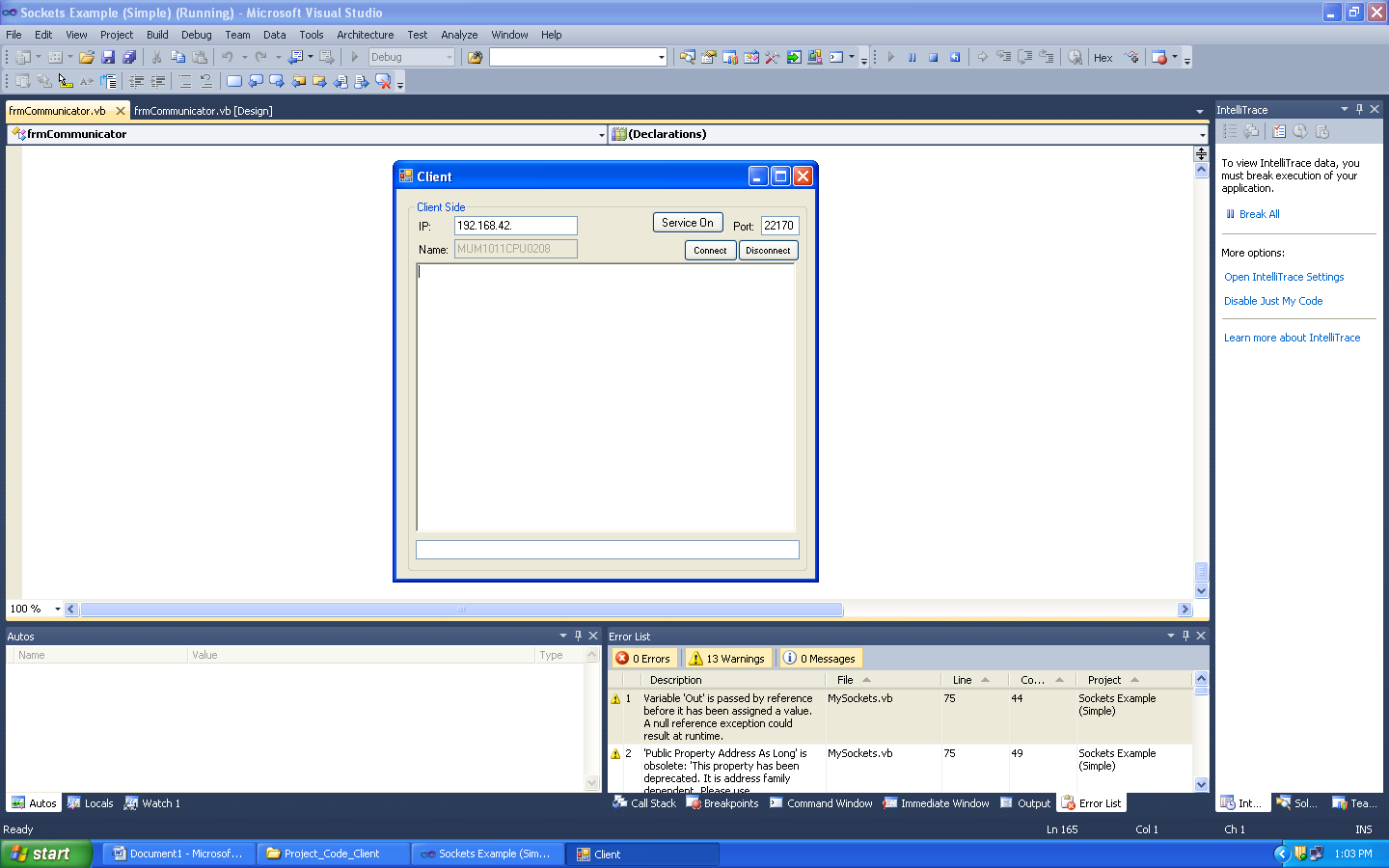
**SCREENSHOTS**

**SERVER SOCKET:**

****

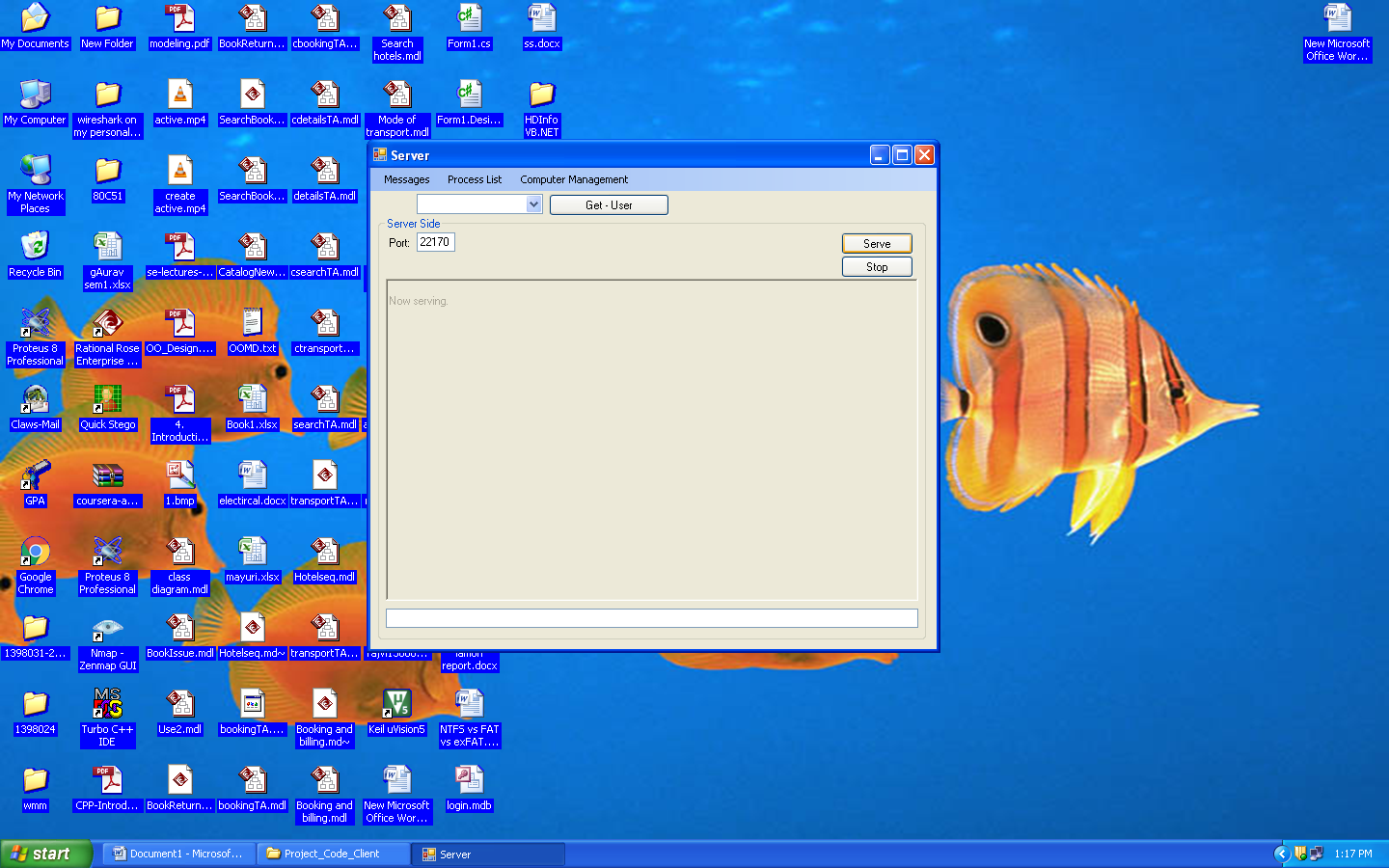
**FIGURE 1.1**

**CLIENT SOCKET**

****

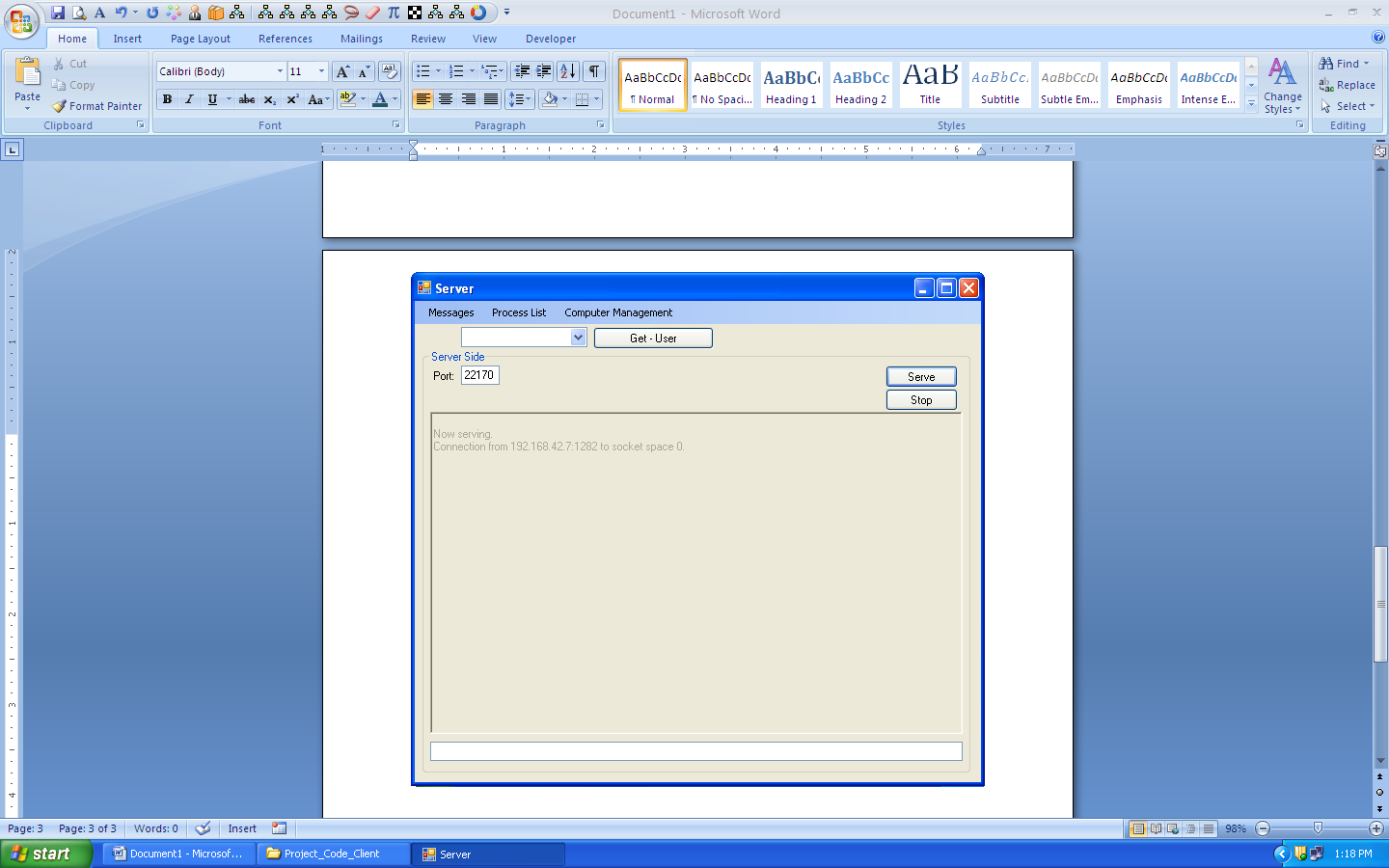
**FIGURE 1.2**

**SERVER STARTED**

****

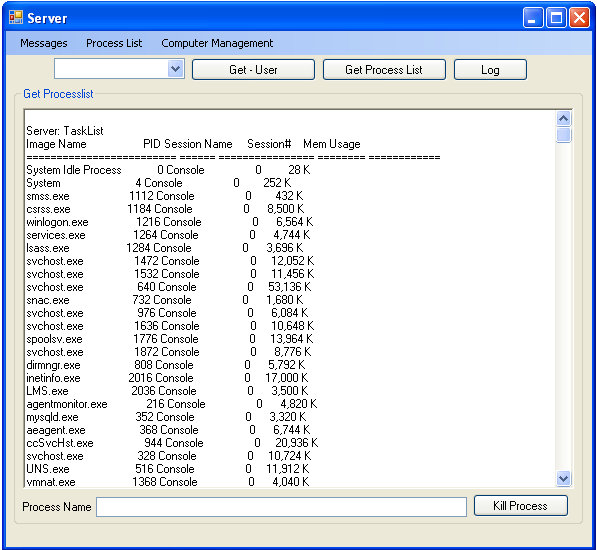
**FIGURE 1.3**

**CLIENT CONNECTED**

****

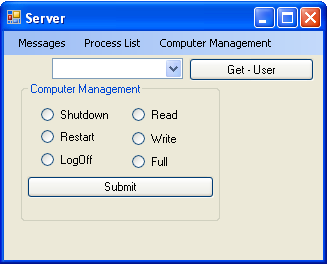
**FIGURE 1.4**

**PROCESS LIST & KILL**

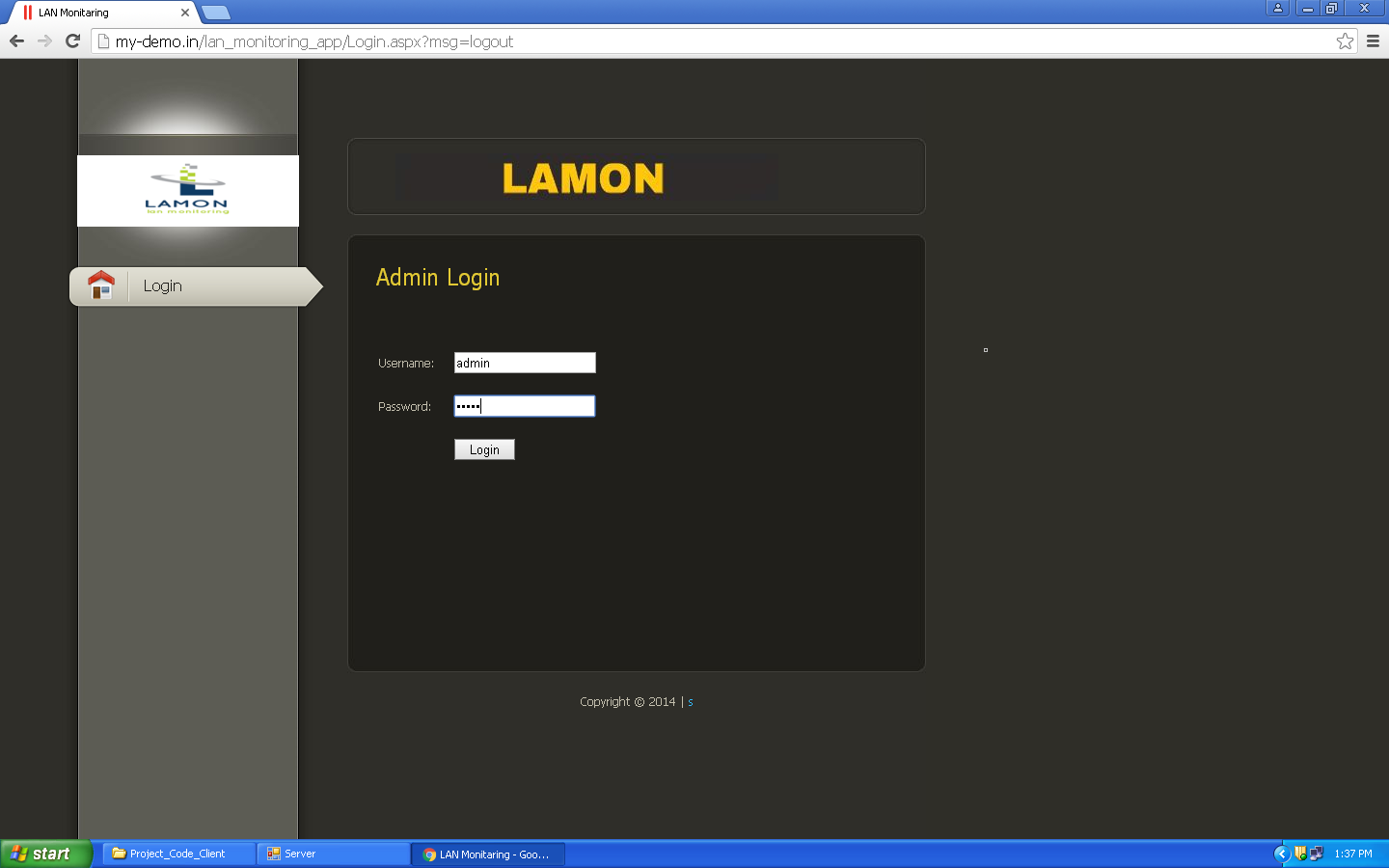
****

**FIGURE 1.5**

**CLIENT REMOTE HANDLING**

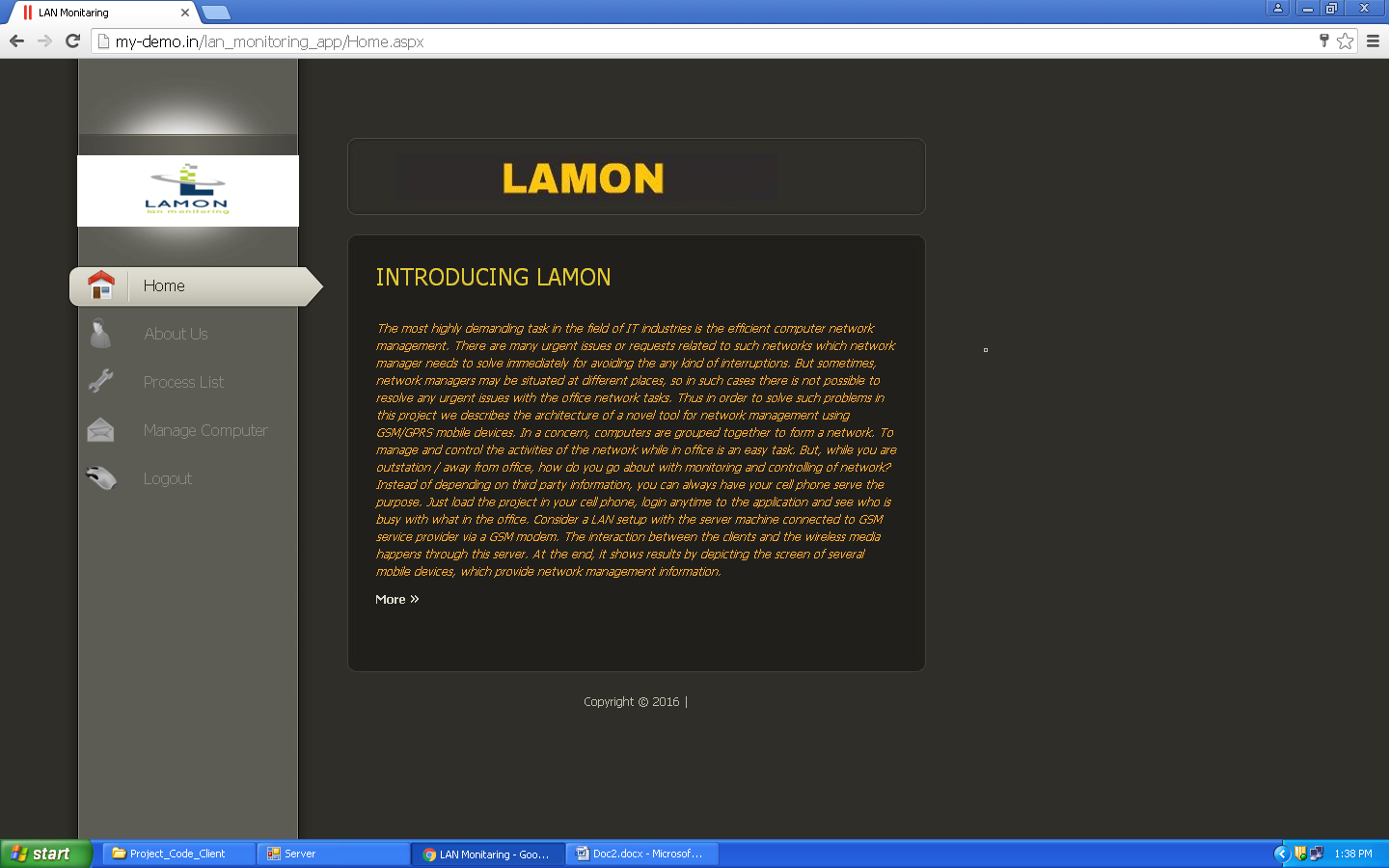
****

**FIGURE 1.6**

****

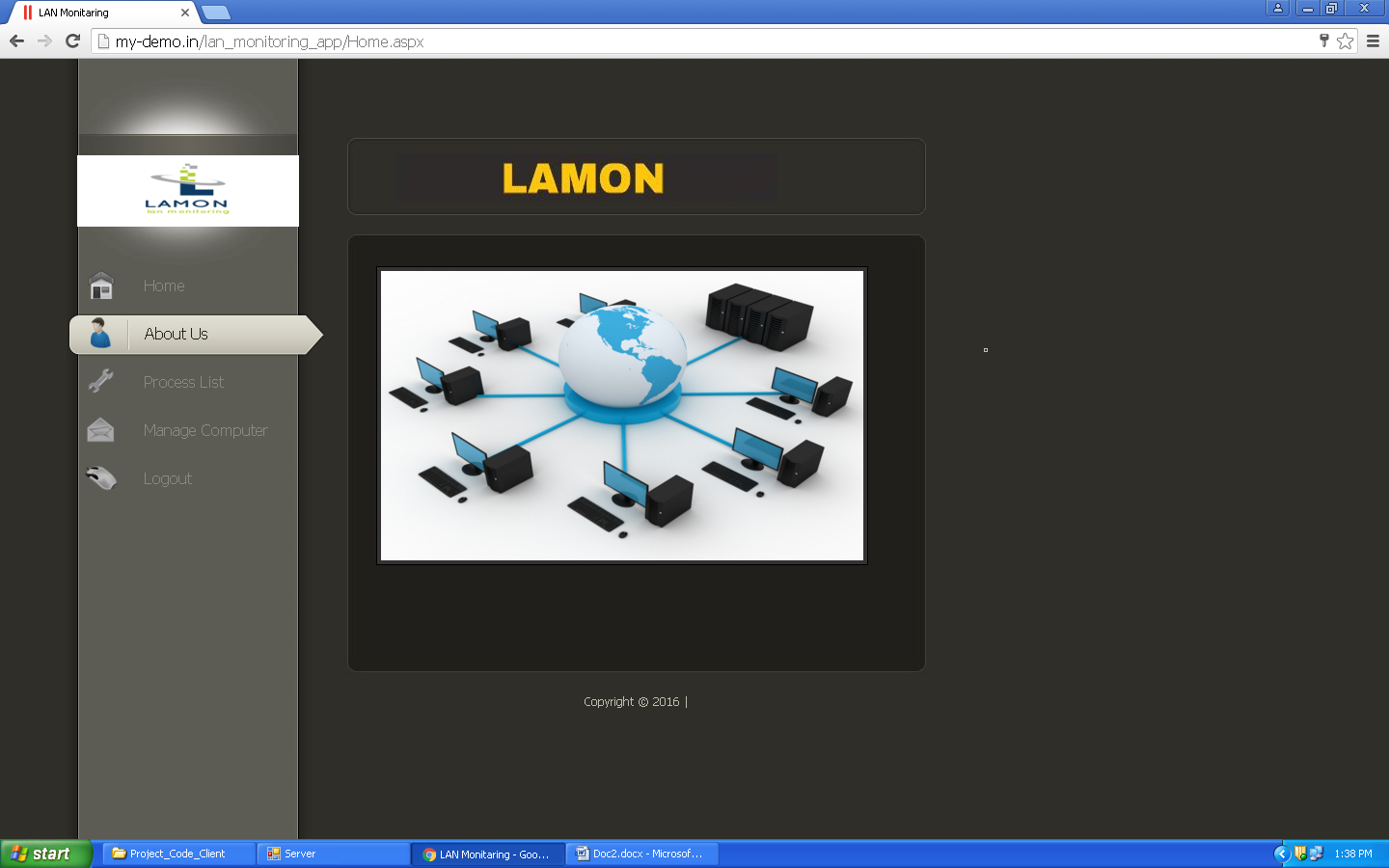
**WEBSITE LOGIN**

**FIGURE 2.1**

****

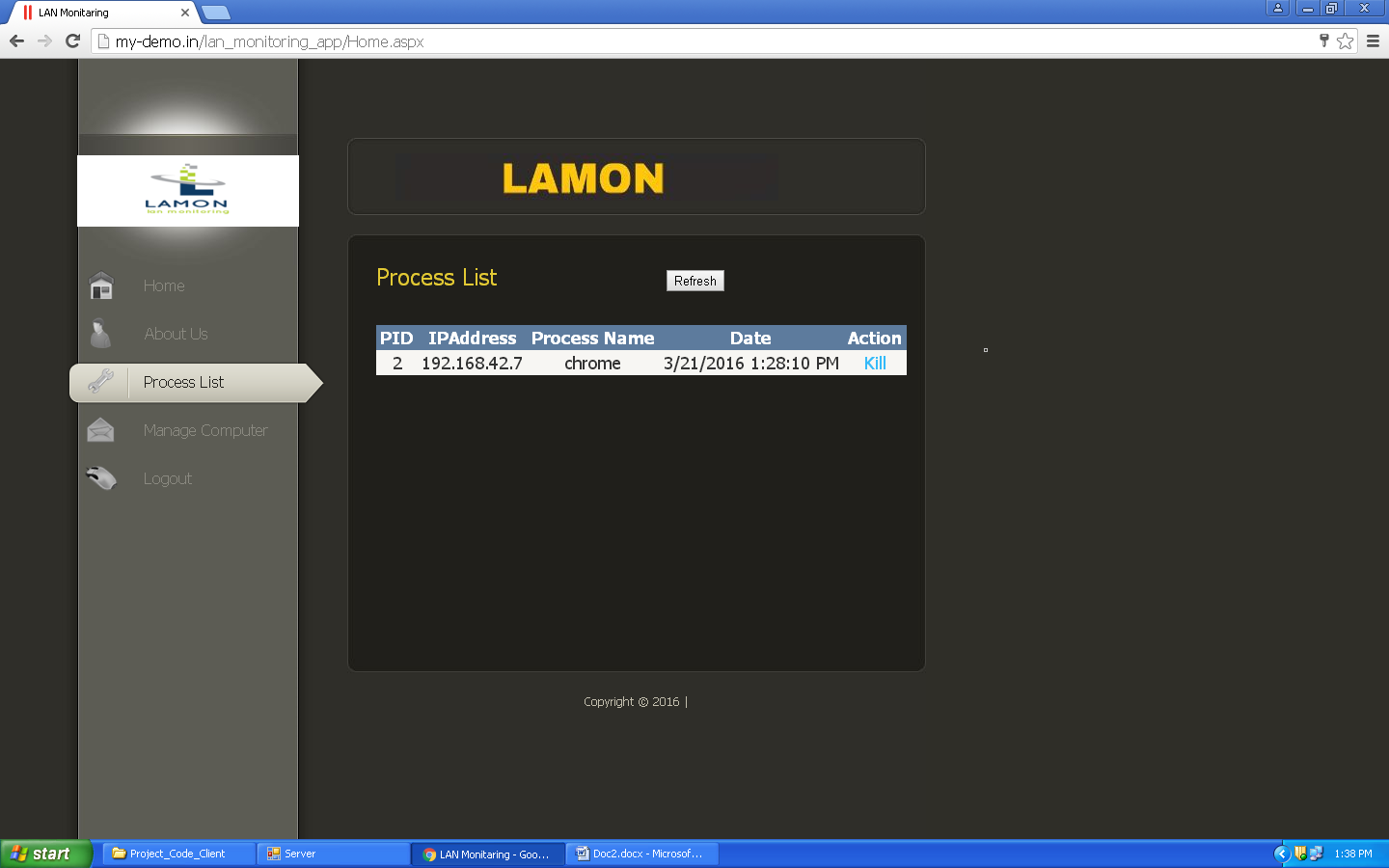
**HOME PAGE**

**FIGURE 2.2**

****

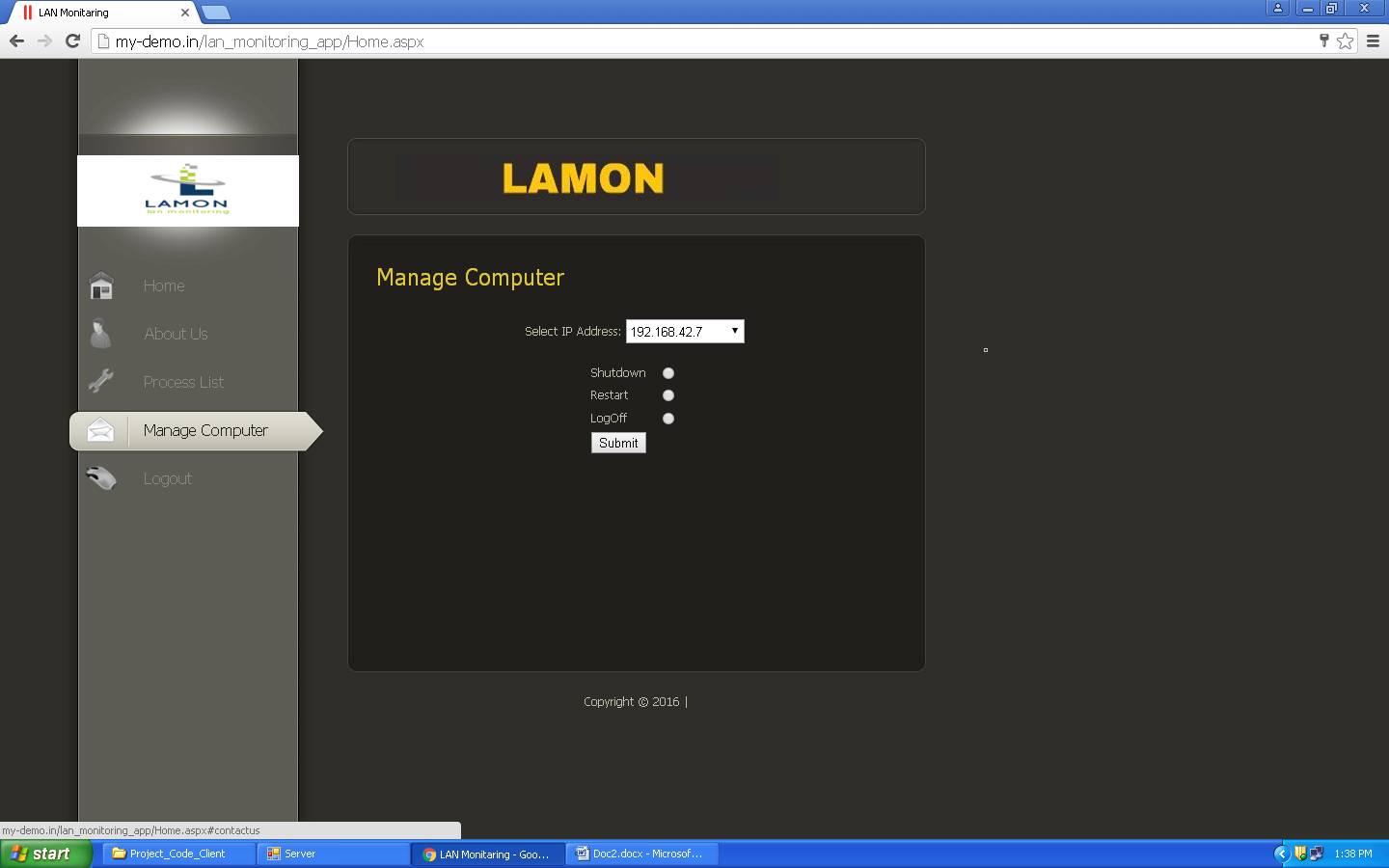
**PROJECT INFO**

**FIGURE 2.3**

****

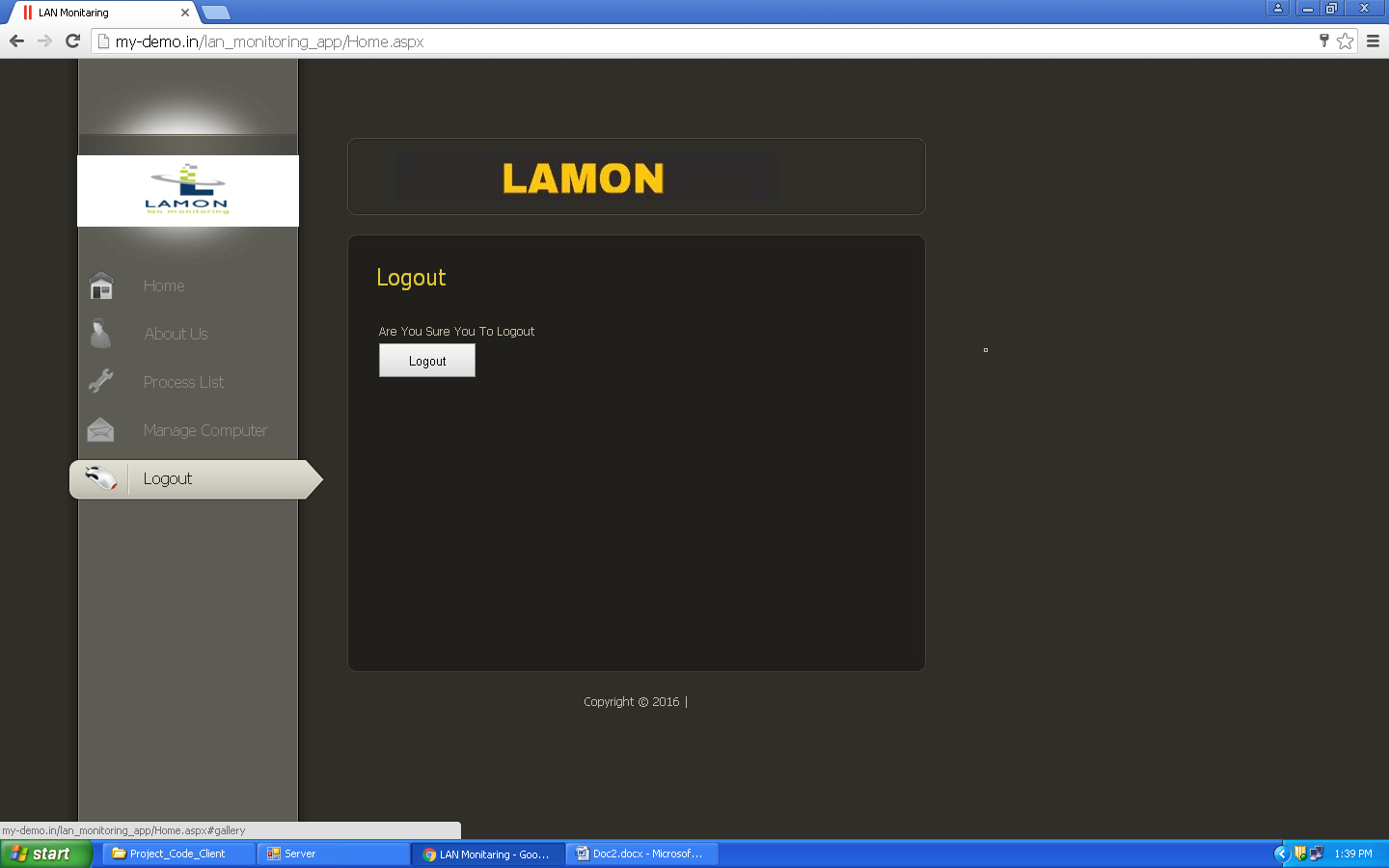
**PROCESS LIST & KILL**

**FIGURE 2.4**

****

**CLIENT REMOTE HANDLING**

**FIGURE 2.5**

****

**LOGOUT**

**FIGURE 2.6**

****

**SYSTEM REQUIREMENTS**

**SYSTEM REQUIREMENTS:**

Hardware and software Requirement:-

**Hardware Requirement**

* LAN
* PCs (At least two)

**Software Requirement**

* Microsoft Visual studio 2010
* Microsoft SQL SERVER 2008
* Web Server

****

**ADVANTAGES**

**ADVANTAGES:**

* Secure LAN permission oriented system
* It can be implemented with ‘n’ of clients, it should be connected in Lan
* It has moderate cost
* Remotely accessible
* Can be accessible via cell phones via Internet
* It is quiet User-Friendly for Network Admin



**DISADVANTAGES**

**DISADVANTAGES:**

* It requires internet connectivity in cell phone
* Client should manually connect to Server
* At times when Server is down, website cannot be accessed remotely



**APPLICATIONS**

**APPLICATIONS:**

* Server monitoring at the university/college level can be used for monitoring, logging & retention of network traffic that transverse university networks
* Server at the office level can be used to monitor the office Server by Administrator any time if at a particular time administrator is not present
* Industries, education Sectors, It Firms
* Online examination, viewing processes used by students

****

**FUTURE ENHANCEMENT**

**FUTURE ENHANCEMENT:**

* Creation of Access Control Matrix of all users connected in the Lan
* When user accessing the process , only the application assigned to it should be allowed to open other process must hidden
* When client opens a new process an immediate message should be send to the Server application
* Classification of processes in log files(recently used, most used, not used)
* Detecting the screen of user and keystroke(Ctrl+Alt+del) and accordingly take action on it

**CONCLUSION:**

Thus from this experiment of Web Based LAN Monitoring we tried to implement most of the admin features that network administrator can remotely access and monitor. The Website feature added easy accessibility and user friendly interface to communicate with the LAN clients and also monitoring them. Also we added two more functionality other than the problem definition such as creation of log file & sending e-mail to network admin. We also tried to detect online clients on PC, defragmenting and creating a multiple client interface but we did not succeed in doing so, which as of now falls under the future scope of our project development. As our website is hosted, main advantage is getting accessibility of LAN remotely.**REFERENCES:**

* <http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-3-718-722.pdf>
* <https://msdn.microsoft.com/en-us/library/33487zw6.aspx>
* <http://stackoverflow.com/questions/27885482/list-of-all-running-processes-in-android>
* <http://www.codeproject.com/search.aspx?q=socket+connection&x=0&y=0&sbo=kw>
* <https://msdn.microsoft.com/en-us/library/s4yys16a(v=vs.90).aspx>
* <http://www.tutorialspoint.com/vb.net/vb.net_event_handling.htm>
* <http://www.codeproject.com/Tips/471799/jQuery-introduction-and-how-to-use-jQuery-with-ASP>
* <https://adriantomic.se/development/jquery-localscroll-tutorial/>
* <http://gsgd.co.uk/sandbox/jquery/easing/>
* <http://stackoverflow.com/questions/5936509/scrollto-localscroll-js-not-working-at-all>

**ABOUT US:**

AAKASH SHAH(1380044)

([skyshah97@gmail.com](mailto:skyshah97@gmail.com))

PRERAK SHAH(1380048)

([sprerak789@gmail.com](mailto:sprerak789@gmail.com))

DHARMIL SHAH(1380051)

([dharmil.shah70@gmail.com](mailto:dharmil.shah70@gmail.com))