REMOTE ADMINISTRATION

PROJECT WORK DONE AT
LOGIC SOFTWARE SOLUTIONS PVT.LTD., COCHIN

PROJECT REPORT P-791

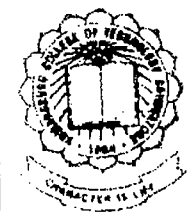
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE AWARD OF THE DEGREE OF
MASTER OF COMPUTER APPLICATIONS
OF BHARATHIAR UNIVERSITY, COIMBATORE

SUBMITTED BY
SAWZAN MURPHY PHILIP
Reg. No. 9938M0636

GUIDED BY

EXTERNAL GUIDE
Mr.JOJAN ANTONY,
Project Leader,
Logic Software Solution Ltd(),
Cochin.

INTERNAL GUIDE
Mr.S.GANESH BABU MCA.,
Lecturer,
Department of Computer science&Engg.
Kumaraguru College of Technology,
Coimbatore.



Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore-641 006

May 2002

FTWARE SOLUTIONS PVT. LTD.
Towers, St. Albert's College Lane, Banerji Road, Cochin - 682 018, Kerala, India.
1-484-393141, 393148, 398448, Fax: 91 - 484 - 396201
ail: lss@eth.net, www.lssindia.com

VUE
AUTHORIZED TESTING CENTER

CERTIFICATE

This is to certify that the project titled "**Remote Administration**" is a bonafide record of the project work done by **Ms. Sawzan Murphy Philip, MCA** student of Kumaraguru College Of Technology(affiliated to Bharathiar University) , Coimbatore during the period **December 2001** to **April 2002**.

The project was successfully completed and tested for its functionality at **LOGIC SOFTWARE SOLUTIONS PVT. LTD.**

Jojan Antony
Project Manager

Biju Joseph Jacob
Director

LOGIC SOFTWARE SOLUTIONS (P) LTD. COCHIN-18

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to the Bharathiar University)

Coimbatore-641 006

CERTIFICATE

This is to certify that the project work entitled

REMOTE ADMINISTRATION

Done by

SAWZAN MURPHY PHILIP

Reg.No: 9938M0636

Submitted in partial fulfillment of the requirements for the award of the degree of **Master of Computer Applications of Bharathiar University.**

S. J. 22/4/'02

rofessor and Head

epartment of Computer Science & Engineering

Internal Guide

ubmitted to University Examination held on 09/05/2002.

ıternal Examiner

External Examiner

DECLARATION

I hereby declare that the project entitled **'REMOTE DMINISTRATION'** submitted to **Bharathiar University** as the project work of **laster of Computer Application Degree**, is a record of Original work done my e under the supervision and guidance of **Mr.JojanAntony MCA, Project eader, Logic Software Solutions, St.Albert'sLane, Cochin** and **Ir.S.GaneshBabu MCA, Kumaraguru College of Technology, Coimbatore** nd this project work has not found the basis for the award of any egree/Diploma/Associateship/Fellowship or similar title to any candidate of any niversity.

Place: Cochin.

Date: 16/04/2002.

SAWZAN MURPHY PHILIP

Internal Guide

External Guide

ACKNOWLEDGMENT

I am incredibly fortunate to have so many people helping me on this project. rst I wish to thank ***Mr. Biju Joseph jacob, Director, Logic Software Solutions Pvt d, Cochin*** *for entrusting this project to me and providing all facilities that made the perience a pleasant one. In particular my deepest thanks goes to* ***Mr. Jogan ntony,Project manager,*** *for recognizing and approving this project and for his mense support, timely suggestions, and technical guidance throughout the project ork.*

I am extremely thankful to ***Dr.S.Thangasamy, Professor and Head of the epartment*** *for his kind suggestion he has given in every step throughout our studies nd in this project work.I am grateful to my internal guide* ***Mr. S Ganesh Babu MCA,*** ***ecturer, Kumaraguru College of Technology*** *who offered his guidance and always upported me with keen interest and constant encouragement suggestions in every step rough out the project.*

I wish to thank my parents for all their encouragement support and love. Vithout their encouragement I would never have gotten where I am.

Last but not the least I would like to thank my Classmates and friends for their ve and moral support. Finally thanking You God for your immense support and love.

SYNOPSIS

The proposed system **Remote Administration** is developed for **LOGIC SOFTWARE SOLUTIONS**. The tool used for developing the software **is Microsoft Visual C++.** This is a remote control program, which enables the administrator to work on any remote computer from his own computer. The remote computer screen can be seen in a window on his desktop or on the full screen of his computer.

This project actually consists of two parts, the client and the server. The client software is to be run on the remote machine whose screen is to be captured. The client software captures the client's screen and converts it into a bitmap and sends it to the server. The server software is to be run on the administrator's machine that wishes to monitor and control the remote machines. The server software running on the administrator's machine draws the display on the administrator's desktop.

The mouse and keyboard of the administrator's machine are logically transferred to the remote computer. As soon as the screen of the remote machine changes corresponding changes are updated in the administrator's machine. So he can work with the remote computer as if he is sitting in front of it.

The remote computer can be anywhere in the network. There exist a need for a fast network connection to work with the remote computer. If a LOCAL NETWORK is used a real-time update speed can be reached.

TABLE OF CONTENTS

INTRODUCTION

1. Project Overview

Remote administration is developed using Two Tier Architecture (Client/Server Model) ethodology.

Remote administration consists of two parts:

- The client, which generates a display.
- The server, which actually draws the display on administrator's screen.

So, to get started with Remote admin the client software must be running on the remote achine, as well as the server software must be running on the administrator's machine. mote admin requires a TCP/IP (Transmission Control Protocol/Internet Protocol) connection tween the server and the client. The IP Address of the remote machine should be known to nnect.

In the server module, a main window is created. This window includes a Menu. The enu contains a menu item called Administrator. The Administrator menu includes the omenu, which contains the options like Child Windows, Machine Names, and Exit.

While clicking on the Child Windows option another window will be displayed. This ndow will include 12 blank child windows. Then in this child window we can see the mpressed desktop of the other 12 machines. For this the mouse is placed on any child window d the right button is clicked. Then a dialog box is displayed. In this dialog box there is list box m which we can select the IP Address of the remote machine that is to be connected. On cking the workstation button a connection is established between this client and the server. en we can view that client's desktop in a compressed manner in a child window. This can be eated for each child window .So that we can see the 12 different client's desktop in a mpressed form in 12 different child windows.

Just by double clicking a client's compressed window the enlarged screen of the rresponding client can be viewed.

On selecting the Machine Names a property page with two tabs visa, Machine Names and rvices will be displayed .On selecting the services he can view the available services and can lect any.

The client software must be running on any in order to capture its screen. This client ftware saves the desktop as a bitmap in a shared disk drive. The updating of screen takes place ly after getting the acknowledgement from the server i.e., only after loading the image into rver screen.

When there arises a need to connect to a client the server sends the server machine's me and the path where to store the bitmap to the client. The client stores the bitmap in the ecified path and sends an acknowledgement to the server. The server loads the image and nds the client an acknowledgement that it can write the next screen.

2. Organization Profile

LOGIC is Cochin based Information Technology consulting firm specializing in the sign, development and integration of custom-built business systems and solutions using vanced internet, electronic commerce and client server technology.

Since 1996, it has built a solid reputation as an enterprise dedicated to offering a mplete and sound approach to systems development in the emerging paradigm of the tworked world.

The clients are industry leaders who pursue innovative ways of exploiting information chnology to create, manage and distribute information to support their business strategies. They ek a new approach to the delivery of custom solutions, backed by a team of dynamic and mpetent professionals. They want a partner with the capacity to effectively develop new stems to realize their corporate objectives whether they be developing stronger customer lations, leveraging human capital, or improving business efficiency.

The management team is structured to fully support and guide the clients through their formation system initiatives. With their solid experience in the field of systems consulting, evelopment and integration, the managers understand the human and business aspects of formation technology and systems development as well as the technical issues. This assures the ients of a greater chance of achieving the desired results.

ddress: Logic Towers,
St.Alberts College Lane,
Banerji Road Cochin-18
hone: 0484-393141, 393148.
mail: lss@vsnl.com
Veb site: lssindia.com

SYSTEM STUDY AND ANALYSIS

. Existing System

In the existing system the administrator who wants to administer remote PC should go r each machine. It's not possible to monitor more than one machine at a time. The monitoring administering of the remote PCs are done manually. At present there is no way to view a nputer's screen from another machine. The only way is to manually go near the system and nitor it. It is not possible to check out what subordinates are doing without going near his/her tem. Sitting in his system the administrator cannot view all the keystrokes and the desktop of subordinates. There is no way to correct the errors of the subordinates other than going near n. If the machine is not in use and the administrator wants to shut down the system he should near it. Sitting on his system the administrator can't lock or unlock any other machine. So e arise the need of a system, which could efficiently do the administration of a remote nputer.

sting System Drawbacks:

There were a lot of reasons for the introduction of the new system. They are mainly due he drawbacks and efficiency of the existing system.

The following drawbacks are observed in the existing system.

- For administering the remote PC administrator should be physically present at the remote computer.
- The delay in reaching nears the computers.
- Problems in being physically present near the remote computer, which may be far away etc.
- The existing system is time consuming.
- It cannot monitor more than one machine at a time.
- The administrator can't provide any immediate remote assistance to his subordinates.

2. Proposed System

The proposed system provides a better way to administer the remote computer. Here ere is no need to go near the remote computer to administer it. The software shows the remote sktop on administrator's computer itself. This helps him to monitor all the activities of his bordinates. The administrator can view the screens of different subordinates at a time in a mpressed form or he can see the enlarged screen of one subordinate. The mouse events and e keystrokes made by the administrator are transferred to the remote computer. Thus he can ntrol all the activities of his subordinates. Thus this software helps the administrator to work th the remote computer as if he is sitting in front of it.

dvantages of Proposed System

Remote Desktop Viewing

Grab screenshots of the remote computer's desktop allowing the administrator to see hat his subordinates are doing on the PC in real time.

Real-time Remote Keystroke Viewing

Helps the administrator to view all keystrokes made by Subordinates in real-time, as they are ped.

Remote assistance

Giving Administrator the ability to aid students as if the support person were sitting next to e subordinates. The administrator can see what his subordinate is doing and is able to share or ke over the control of the mouse or keyboard and guide him.

It can be used to monitor subordinate's activity

If the client module is installed in machines, which subordinates use, the teacher can monitor e subordinate's activity without their knowledge.

Remote system shutdown

The remote systems left unused can be shutdown/restarted.

It can be used to leave the task while working.

One can leave his room while in the middle of writing a document and later reconnect from other room to finish composing the document, from just where he left off.

Remote File System Navigation and Management

Helps to browse and navigate the entire file system of remote PC.

3. Requirements of new system

The process starts with initial feasibility study, the first major stage is the requirement alysis in which the software development staff work with the customers and system end-users find out about the application domain, what services the system should provide, the required rformance of the system, hardware constraints, and so on. Requirement Analysis is an portant process. The acceptability of the system after it has been delivered depends on how ll it meets the customer's needs.

PROGRAMMING ENVIRONMENT

1. Hardware Configuration

The hardware configuration for REMOTE ADMINISTRATION to perform smooth nning is as

- **Server**

  Processor: Intel Pentium III 500 MHz.
  Memory Size: 128 MB RAM
  Hard Disk Drive: 1.2 GB Hard Disk
  Monitor: 14" SVGA Color Monitor

- **Client**

  Processor: Intel Pentium III 500 MHz
  Memory Size: 64 MB RAM
  Hard Disk Drive: 1.2 GB Hard Disk
  Monitor: 14" SVGA Color Monitor

3.2. Description of Software &Tools Used

Programming Language used

Microsoft Visual C++ 6.0

The Operating System required

Server Module - Windows NT 4.0/2000

Client Module - Windows 98 and above

Visual C++ is an extremely powerful tool used for Windows programming.

Many programmers consider it the most powerful Windows Programming tool today. Visual C++ is not actually one tool, it is a collection of tools, wrapped together into one dynamic, ready to use package.

Visual C++ enjoyed a reputation for creating small and efficient programs.

A program, which is written using this language, can operate at nearly the same speed as one written in assembler, without any problems. The C++ is a great language for writing low-level code like operating system, device drivers, and DLLs. These three kinds of code still represent the major places were Visual C++ is used. The small fast code that Visual C++ produces is very much appreciated in the time-critical environment of an operating system.

Visual C++ provides better prototyping facility.

One of the latest features that Microsoft has added is better prototyping capabilities through the use of enhanced wizards. Now this feature won't place Visual C++ on par with the products like VB6, but it reduces the time required to get an application started.

Visual C++ is also an ideal programming environment for writing ActiveX controls, along with IIS-specific code like ISAPI (Internet Sever Application Programming Interface) extensions and ISAPI filters.

Another area where Visual C++ excels is database programming.

Since Microsoft is maker of windows, the people there have the best idea of how things work inside in the operating system. We can find this inside knowledge at work in the Windows API hooks found in the MFC. Visual C++ also offers many features that may not be found in other products.

Visual C++ helps to overcome the creation of windows programs.

The first way to create Windows programs was to use C, and to create large, complex programs. For this it requires many pages of code. It took five pages of difficult and mysterious code to simply put a blank window on the screen. But with the introduction of C++, which is perfect for Windows programming. In this we can wrap large arts of the code into self contained C++ objects. This meant that it become a great deal easier to handle the programs. Instead of one long monolithic, Windows programs are now divided up into neat, manageable chunks.

Introduction of MFC class library provides codes make easier.

In addition Microsoft introduced the Microsoft Foundation Class library, which is an extra ordinary package of prewritten, ready to use code. Introducing MFC was just as important as introducing VC++ to windows. Visual C++ not only makes the use of MFC, but also makes Windows programming far easier by introducing many programming tools, such as menu editor for designing menus, and the dialog editor for designing the dialog boxes. Visual C++ provides us with one integrated design environment in which we can write our programs and run them. In addition Visual C++ organizes the many files a Windows program needs into projects. Visual C++ wizards will even write a good part of our programs for us.

sual C++ workspaces and projects

Visual C++ organizes programming task into the projects, and usually each separate gets o its own project. A project is a collection of files that are all used to create one working, able program. In addition, projects themselves are placed in workspaces, and a workspace can ve several projects in it.

e Parts of a Visual C++ Program

There are four major parts of a Visual C++ AppWizard program: the application object, main window object, the document object, and the view object.

The Application object

The application object supports the definitions of constants and the declarations of variables and methods, when this object is started; it places the main window in the screen.

The Main Window object

The main window object displays the program itself, and this is where we find in the menu bar, the title bar, and tool bar. The main window object is responsible for everything that surrounds the area where the action-the drawing, text, and other graphics in our window.

The Document object

In the document object, we store the data of our program. Visual C++ makes it easier to store the all data in the document object, and then handle the display of the data that will fit into the client area in the view object.

The View object

The view object handles the client area where we will format and display the data in our program, such as the text we are editing if we are creating a word processing program. The view object is really a window itself that appears on the top of the client area. The data we display in the view object is stored in the document object.

pplication Types in Visual C++

Visual C++ is capable of creating any application that we can imagine. However, there e five application types that can be used with Visual C++.

Console Application

It represents that situation where we really need to maintain some type of compatibility with legacy system or don't need a full-fledged interface for the user to work with.

Dialog-based Application

These applications are normally reserved for utilities or an application that is too small to require a complete mending system.

Single-document Application

Single-document Application are representative of simple application that work with their own data like note takers or small database front ends. These applications also require a mending system of some type.

Multiple-document Application

Multiple-document Applications include full-fledged applications like word processors and spread sheets. When we think about it, they represent that fringe area of C++ programming where we need to weigh the flexibility of C++ against the development speed offered by RAD programming environment like Visual Basic.

HTML-based Application

HTML-based Application is new to Visual C++ 6.0.They are applications that work with data of some type (like single-document applications and multiple-document applications) but with an Internal twist. Instead of standard editor, the user can see what amounts to Web browser front end.

ndows NT Server

In the late 1980s software developers created Windows as a graphical environment for grams running on Microsoft DOS. Microsoft and IBM collaborated on a replacement for S on Intel computers. Their new operating system had many advanced features and was led OS/2. At the same time Microsoft recognized the need for another more advanced rating system that would have not only all the features of OS/2 but also the ability to run on

ıer microprocessors, especially reduced instruction set computers (RISC) microprocessors, ıich at the time were much faster than Intel microprocessors. This visionary operating system ɔuld have to be written in a high level language such as C that could be ported to other croprocessors, instead of in Intel assembly language, which was not portable.

The most popular Windows NT was originally given the name as OS/2 NT. The reason as why the name was changed to Windows NT is because of the disagreement between IBM and icrosoft as to how their two operating systems OS/2 and Windows should be marketed. icrosoft wanted their Windows to be promoted were as IBM wanted Windows to be used as a ɔpping-stone for the more advanced OS/2. The collaboration broke down and IBM retained ɛir OS/2 and Microsoft released Windows NT.

Windows NT server may look like Windows 95but in reality it is completely different ernally. Windows 95 is much more complicated than Windows NT because Windows 95 must able to run all the programs written for previous versions of Windows and DOS and meeting s requirement is not easy. Windows NT on the other hand has to run only programs written for ndows NT and those programs for Windows NT and those programs for Windows and DOS t do not interfere with the Windows NT security mechanism.

atures and Capabilities

The Windows NT server is a large and complex operating system with many modular mponents. However, three major constituents- the operating system kernel, the file system, and networking services- interact to provide Windows NT server's characteristic features and ɔabilities as a network server. The figure indicates the components of a Windows NT server.

ndows Sockets

The Windows Sockets specification defines a binary-compatible network-programming erface for Microsoft Windows. Windows Sockets are based on the UNIX® sockets plementation in the Berkeley Software Distribution (BSD, release 4.3) from the University of lifornia at Berkeley. The specification includes both BSD-style socket routines and extensions ɛcific to Windows.

Using Windows Sockets permits your application to communicate across any network at conforms to the Windows Sockets API. On Win32, Windows Sockets provide for thread fety.Many network software vendors support Windows Sockets under network protocols cluding Transmission Control Protocol/Internet Protocol (TCP/IP), Xerox® Network System NS), Digital Equipment Corporation's Decent™ protocol, Novell ® Corporation's Internet cket Exchange/Sequenced Packed Exchange (IPX/SPX), and others. Although the present indows Sockets specification defines the sockets abstraction for TCP/IP, any network protocol n comply with Windows Sockets by supplying its own version of the dynamic link library LL) that implements Windows Sockets. Examples of commercial applications written with indows Sockets include X Window servers, terminal emulators, and electronic mail systems.

Note Keep in mind that the purpose of Windows Sockets is to abstract away the derlying network so you don't have to be knowledgeable about that network and so your plication can run on any network that supports sockets. Consequently, this documentation esn't discuss the details of network protocols.

The Microsoft Foundation Class Library (MFC) supports programming with the indows Sockets API by supplying two classes. One of these classes, Socket, provides a high vel of abstraction to simplify your network communications programming.

The Windows Sockets specification, Windows Sockets: An Open Interface for Network mputing under Microsoft Windows, now at version 1.1, was developed as an open networking ndard by a large group of individuals and corporations in the TCP/IP community and is easily ailable for use. The sockets programming model supports one "communication domain" rrently, using the Internet Protocol Suite. The specification is available in the Win32 SDK.

finition of a Socket

A socket is a communication endpoint — an object through which a Windows Sockets plication sends or receives packets of data across a network. A socket has a type and is sociated with a running process, and it may have a name. Currently, sockets generally change data only with other sockets in the same "communication domain," which uses the ernet Protocol Suite.

Both kinds of sockets are bi-directional: they are data flows that can be communicated in th directions simultaneously (full duplex).

Two socket types are available:

Stream sockets

Stream sockets provide for a data flow without record boundaries - a stream of bytes. Streams were guaranteed to be delivered and to be correctly sequenced and unduplicated.

Data gram sockets

Data gram sockets support a record-oriented data flow that is not guaranteed to be delivered and may not be sequenced as sent or unduplicated.

"Sequenced" means that packets are delivered in the order sent. "Unduplicated" means that you get a particular packet only once. Under some network protocols, such as XNS, streams can be record-oriented — streams of records rather than streams of bytes. Under the more common TCP/IP protocol, however, streams are byte streams. Windows Sockets provides a level of abstraction independent of the underlying protocol.

e SOCKET Data Type

Each MFC socket object encapsulates a handle to a Windows Sockets object. The data e of this handle is SOCKET. A **SOCKET** handle is analogous to the HWND for a window. FC socket classes provide operations on the encapsulated handle.

The **SOCKET** data type is described in detail in the Win32 SDK. See the topic cket Data Type and Error Values under Windows Sockets.

es for Sockets

Sockets are highly useful in at least three communications contexts:

Client/Server models

Peer-to-peer scenarios, such as chat applications

Making remote procedure calls (RPC) by having the receiving application interpret a message as a function call

Windows Sockets: Stream Sockets

This article describes stream sockets, one of the two Windows Socket types available. Stream sockets provide for a data flow without record boundaries — a stream of bytes that can e bi-directional (the application is full-duplex: it can both transmit and receive through the ocket). Streams can be relied upon to deliver sequenced, unduplicated data. ("Sequenced" eans that packets are delivered in the order sent. "Unduplicated" means that you get a articular packet only once.) Receipt of stream messages is guaranteed, and streams are well ited to handling large amounts of data.

The network transport layer may break up or group data into packets of reasonable size. ne Socket class will handle the packing and unpacking for you.

Streams are based on explicit connections: socket A requests a connection to socket B; cket B accepts or rejects the connection request.

A telephone call provides a good analogy for a stream: under normal circumstances, the eiving party hears what you say in the order that you say it, without duplication or loss. eam sockets are appropriate, for example, for implementations such as the File Transfer otocol (FTP), which facilitates transferring ASCII or binary files of arbitrary size.

Stream sockets are preferable to data gram sockets when the data must be guaranteed to ve and when data size is large. For more information about stream sockets, see the Windows kets specification. The specification is available in the Win32 SDK.

ndows Sockets: Data gram Sockets

This article describes data gram sockets, one of the two Windows Socket types available. a gram sockets support a bi-directional data flow that is not guaranteed to be sequenced or uplicated. Data grams also are not guaranteed to be reliable; they can fail to arrive. Data gram may arrive out of order and possibly duplicated, but record boundaries in the data are erved, as long as the records are smaller than the receiver's internal size limit. You are

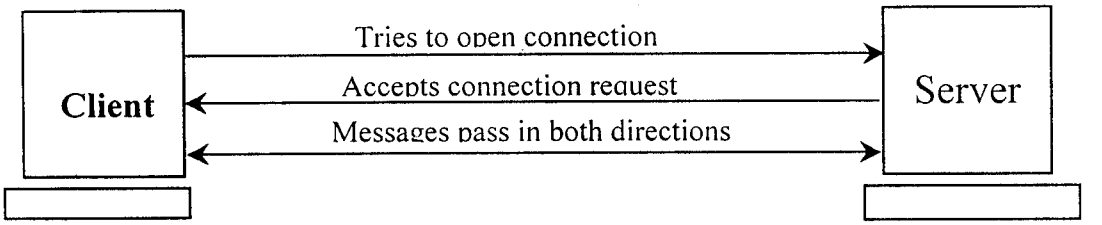
esponsible for managing sequencing and reliability. (Reliability tends to be good on local area etworks (LANs) but less so on wide area networks (Wanes), such as the Internet.

Data grams are "connectionless" — no explicit connection is established; you send a data ram message to a specified socket and you can receive messages from a specified socket.

An example of a data gram socket is an application that keeps system clocks on the etwork synchronized. This illustrates an additional capability of data gram sockets in at least me settings: broadcasting messages to a large number of network addresses.

Data gram sockets are better than stream sockets for record-oriented data. For more formation about data gram sockets, see the Windows Sockets specification. The specification is ailable in the Win32 SDK.

The Basic Socket Connection Process



CP/IP (Transmission Control Protocol/Internet Protocol)

TCP/IP is a set of protocols developed to allow cooperating computers to share sources across a network. A community of researchers centered on the Arpanet developed it. ertainly the Arpanet is the best-known TCP/IP network. However as of June, 87, at least 130 fferent vendors had products that support TCP/IP and thousands of networks of all kinds use it.

ome basic definitions

The Internet is a collection of networks, including the Arpanet, Usenet, regional networks ch as Nicene, and local networks at a number of University and research institutions, and a mber of military networks.

The term "Internet" applies to this entire set of networks. The subset of them that is anaged by the Department of Defense is referred to as the "DDN" (Defense Data Network). his includes some research-oriented networks, such as the Arpanet, as well as more strictly ilitary ones. All of these networks are connected to each other. Users can send messages from y of them to any other, except where there is security or other policy restrictions on access. fficially speaking, the Internet protocol documents are simply standards adopted by the Internet ommunity for its own use.

Whatever it is called, TCP/IP is a family of protocols. A few provide "low-level" nctions needed for many applications. These include IP, TCP, and UDP. (These will be escribed in a bit more detail later.) Others are protocols for doing specific tasks, e.g.

ansferring files between computers, sending mail, or finding out who is logged in on another omputer. Initially TCP/IP was used mostly between minicomputers or mainframes. These achines had their own disks, and generally were self-contained. Thus the most important raditional" TCP/IP services are:

ile transfer

The file transfer protocol (FTP) allows a user on any computer to get files from another omputer, or to send files to another computer. Security is handled by requiring the user to pecify a user name and password for the other computer are made for handling file transfer etween machines with different character set, end of line conventions, etc. This is not quite the me thing as more recent "network file system" or "Net BIOS" protocols, which will be escribed below. Rather, FTP is a utility that you run any time you want to access a file on nother system. You use it to copy the file to your own system. You then work with the local opy.

emote login

The network terminal protocol (TELNET) allows a user to log in on any other computer the network. You start a remote session by specifying a computer to connect to. From that me until you finish the session, anything you type is sent to the other computer. Note that u are really still talking to your own computer. But the telnet program effectively makes your mputer invisible while it is running. Every character you type is sent directly to the other stem. Generally, the connection to the remote computer behaves much like a dialup nnection. That is, the remote system will ask you to log in and give a password, in whatever anner it would normally ask a user who had just dialed it up. When you log off of the other mputer, the telnet program exits, and you will find yourself talking to your own computer. icrocomputer implementations of telnet generally include a terminal emulator for some mmon type of terminal. By the way, the telnet protocol should not be confused with Telnet, a ndor of commercial network services.

twork file systems

This allows a system to access files on another computer in a somewhat more closely egrated fashion than FTP. A network file system provides the illusion that disks or other vices from one system are directly connected to other systems. There is no need to use a cial network utility to access a file on another system. Your computer simply thinks it has ne extra disk drives. These extra "virtual" drives refer to the other system's disks. This bability is useful for several different purposes. It lets you put large disks on a few computers, still give others access to the disk space. Aside from the obvious economic benefits, this ows people working on several computers to share common files. It makes system intenance and backup easier, because you don't have to worry about updating and backing up pies on lots of different machines. A number of vendors now offer high-performance kless computers. These computers have no disk drives at all. They are entirely dependent on disks attached to common "fileservers".

mote printing

This allows you to access printers on other computers as if they were directly attached yours. The most commonly used protocol is the remote line printer protocol from Berkeley ix. Unfortunately, there is no protocol document for this. However the C code is easily tained from Berkeley, so implementations are common.

mote execution

This allows you to request that a particular program be run on a different computer. is is useful when you can do most of your work on a small computer, but a few tasks require resources of a larger system. There are a number of different kinds of remote execution. me operate on a command-by-command basis. That is, you request that a specific command set of commands should run on some specific computer. More sophisticated versions will oose a system that happens to be free. However there are also "remote procedure call" systems t allow a program to call a subroutine that will run on another computer. There are many otocols of this sort. Berkeley Unix contains two servers to execute commands remotely: rsh d rexec. The man pages describe the protocols that they use. The user-contributed software th Berkeley 4.3 contains a "distributed shell" that will distribute tasks among a set of systems.

pending upon load. Remote procedure call mechanisms have been a topic for research for a mber of years; so many organizations have implementations of such facilities. The most despread commercially - supported remote procedure call protocols seem to be Xerox's ourier and Sun's RPC. Protocol documents are available from Xerox and Sun. There is a blic implementation of Courier over TCP as part of the user contributed software with erkeley 4.3. An implementation of RPC was posted to Usenet by Sun, and also appears as part the user-contributed software with Berkeley 4.3.)

SYSTEM DESIGN AND DEVELOPMENT

. Input Design

The input design is the link between the information system and the user. It comprises the eloping specification and procedures for data preparation and those steps are necessary to put nsaction data into a usable form for processing data entry. The activity of putting data into the nputer for processing can be achieved by inspecting the computer to read data from a written printed document or it can occur by having people keying the data directly into the system. e design of input focuses on controlling the amount of input required, controlling errors, oiding delay, avoiding extra steps and keeping the process simple.

So in input design the following things are considered.

- What data should be given as input?
- How the data should be arranged or coded?
- The dialogue to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

The main inputs acquired from the user are as follows.

- Get IP address or domain name of the computer system to be monitored
- User name and password Information required to logon to the server
- Get menu selections
- Trap mouse movements and mouse events of the server and send it to the remote machine
- Trap keystrokes and transfer it to the remote machine

. Output Design

Computer output is the most important and direct information source to the user. Output ign is a process that involves designing necessary outputs in the form of reports that should be en to the users according to the requirements. Efficient, intelligible output design should prove the system's relationship with the user and help in decision making. Since the reports directing referred by the management for taking decisions and to draw conclusions they must designed with almost care and the details in the reports must be simple, descriptive and clear the user. So while designing output the following things are to be considered.

- Determine what information to present
- Arrange the presentation of information in an acceptable format
- Decide how to distribute the output to intended receipts
- Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy.

The main outputs, which are to be handled in this soft ware, are as follows.

- Display the remote machines desk top on the server computer.
- Display the actions corresponding to the mouse events or keystroke in the remote computer and the server machine.

. Process Design

After a detailed study of the existing system and its problem, an outline of the new tem is drawn. A thorough study revealed the structure of the database involved. Then the flow the data involved in the system is synchronized by the process models. The physical models drawn show not only a system does but also how the system is implemented. A process del is a picture of flow of data trough a system and processing performed on that data. Process deling helps us grasp inputs, outputs, processing and the relationship between the processes. ey depict the processes how they interact or interface with one another. These interactions e the form of the data flows between processes. Data flows must be inputs or outputs from ocesses. In any case the key to defining a data flow is known as composition.

LASSES USED IN SERVER MODULE

ServerApp : CWinApp

CWinApp

The CWinApp class is the base class from which you derive a Windows application ject. An application object provides member functions for initializing your application and ch instance of it and for running the application.

Functions Used

1. CWinApp::InitInstance ()

   This is the function that is first called when the instance of the application is created.

2. AfxSocketInit ()

   This function initializes Windows Sockets.

ServerFrameWnd : CFrameWnd

CFrameWnd

The CFrameWnd class provides the functionality of a Windows single document erface (SDI) overlapped or pop-up frame window, along with members for managing the ndows.

Functions Used

1. CFrameWnd::Create ()

   Create initializes the window's class name and window name and registers default values for its style, parent, and associated menu.

2. Userdefinedfunction: ChildWindow()

   Creates small window areas in the current window for client machines

desktop view. So that we can see more than one desktop at a time. Here the userdefined function makechild() is called.

ServerChildWnd : CWnd

CWnd

The CWnd class provides the base functionality of all window classes in the Microsoft undation Class Library.

Functions Used

1. User-defined function: makechild ()

   In this function we call Create function of CWnd and ShowWindow function of CWnd to create child.

2. OnPaint ()

   This member function is called whenever an application makes a request to repaint a portion of an application's window.

3. LoadImage()

The LoadImage function loads bitmap to our device context.

4. CWnd::OnRButtonDown ()

Getting the current working directory and it is converted into string .

This function displays a dialog box whenever the right mouse button is clicked within a ildwindow. The dialog contains the name of the machines in the network.

When a machine name is selected , a socket is created and it tries to establish a nnection. If a connection is established, the socket is sent. The socket contains the current rking directory.

CclientSocket:CSocket

CSocket

Class CSocket derives from CAsyncSocket and inherits its encapsulation of the Windows ckets API. A CSocket object represents a higher level of abstraction of the Windows Sockets PI than that of a CAsyncSocket object. CSocket works with classes CSocketFile and CArchive manage the sending and receiving of data.

ClientSocket

This class is inherited from CSocket. The socket object created from this class is actually ed for sending data between the client and server.

1. CClientSocket::Create ()

No overriding is done.Call the Create member function after constructing a socket object create the Windows socket and attach it. Create then calls Bind to bind the socket to the ecified address.

2. CClientSocket::Connect ()

Return Value: Nonzero if the function is successful; otherwise 0, and a specific error de can be retrieved by calling GetLastError.

No overriding is done. This this member function is called to establish a connection to a eam socket.

3. CClientSocket::Send ()

This member function sends the current working directory of the server program to the ent.

CMachineListDlg: CDialog

CDialog

The CDialog class is the base class used for displaying dialog boxes on the screen. A log box, like any other window, receives messages from Windows. In a dialog box, you are rticularly interested in handling notification messages from the dialog box's controls since that how the user interacts with your dialog box.

MachineListDlg

This class is derived from CDialog. This lists the machine names within the network in a tbox.

1. User defined function: OnDblClkMachine ()

Here the machine name, which was double clicked, is returned.

For this we use GetText method of the CListBox object.

IENT MODULE

ClientApp : CWinApp

AfxSocketInit() is called. Here an object for the mainframe class is created. LoadFrame() called.

1. LoadFrame ()

This function loads the Windows frame window and associated resources and attaches frame window to the CFrameWnd object.

CMainFrame : CframeWnd

The CFrameWnd class provides the functionality of a Windows single document erface (SDI) overlapped or pop-up frame window, along with members for managing the ndow.

erdefined function

CMainFrame : StartTimer()

This function calls the SetTimer() to specify the timer interval(4000 ms) at which the desktop to be **captured,**

CMainFrame : OnTimer()

This function saves the desktop every 4000ms. The desktop is actually stored as a bitmap a .bmp file in the server programs current working directory.

ServerSocket : CSocket

This class is defined for creating a socket object that is used for establishing the nnection.

ServerSocket : OnAccept()

This function accepts the connection from the server and calls the StartTimer() , which s the timer interval.

ClientSocket : CSocket

This class is defined for creating a socket object that is used for data receiving.

ClientSocket:OnReceive()

The current working directory of the server program is received by the client program ough this function.

SYSTEM IMPLEMENTATION AND TESTING

1. System Implementation

Implementation includes all those activities that take place to convert from the old system the new. The old system consists of manual operations, which is operated in a very different anner from the proposed new system. A proper implementation is essential to provide a liable system to meet the requirements of the organizations. An improper installation may fect the success of the computerized system. The new system may be totally new, replacing an isting manual or automated system, or it may be a major amendment of an existing system. In ther case, proper implementation is very essential to provide a reliable system to meet user's as ell as Company's requirements.

nplementation Methods

There are several methods for handling the implementation and the consequent nversion from the old to the new computerized system.

The implementation method used in REMOTE ADMINISTRATION is a direct cut over om the existing manual system to the computerized system.. There are no parallel activities. is strategy requires careful planning.

A working version of the system can also be implemented in one part of the organization d the personnel will be piloting the system and changes can be made as and when required. t this method is less preferable due to the loss of entirety of the system.

nplementation Plan

The implementation plan includes a description of all the activities that must occur to plement the new system and to put it into operation. It identifies the personnel responsible for activities and prepares a time chart for implementing the system.

ıe implementation plan consists of the following steps.

List all files required for implementation.

Identify all data required to build new files during the implementation.

List all new documents and procedures that go into the new system.

The implementation plan should anticipate possible problems and must be able to deal with em. The usual problems may be missing documents; mixed data formats between current and es, errors in data translation, missing data etc.

raining

A well-designed system, if not operated and used properly could fail. Training the users very important, as if not done well it could prevent the successful implementation of an formation system.

Throughout the system development life cycle the user has been involved. By this stage e analyst should posses an accurate idea of the users that need to be trained. They should be rough about their requirements, how to use the system and what the system will do and will not).

Both the system operators and users should need training. During their training, they need be given a trouble-shooting list that identifies possible problems and identifies remedies for e problem.

5.2. System Testing

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Testing is the vital to the success of the system.

In the code testing the logic of the developed system was tested. For this every module of the program was executed to find an error. To perform specification test, the examination of the specifications stating what the program should do and how it should perform under various conditions.

When unit tests are satisfactory concluded the system, as a complete entity must be tested. This enables to detect errors in the coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided. In unit testing step each module has checked separately.

System testing does not test the software as a whole, but rather than integration of each module in the system. The primary concern is the compatibility of individual modules. One has to find areas where modules have been designed with different specifications of data lengths, type and data element name.

Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software was executed several times in order to find out the errors in the different modules of the system.

The testing process is basically divided into two parts

- Module testing
- Test of Integration

Module testing

The module testing process involves testing of each and every module individually without relating to any other modules. Everything in that module alone was tested. Such a test traces bugs that are predominately in the module alone and it is not aimed at tracing bugs that are formed when integration with other module is done.

- **Peak Load Tests:** This determines whether the new system will handle the volume of activities when the system is at the peak of its processing demand. The test has revealed that the new software for the agency is capable of handling the demands at the peak time.
- **Storage Testing:** This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available, because of the use of a number of hard disks.
- **Performance Time Testing:** This test determines the length of the time used by the system to process transaction data.

A study of the system has revealed that the employees due to the user friendliness have accepted the system, reduced the number of errors, increased accuracy and decreased cost of operations. The system also pays for efficient and speedy execution of operations compared to the earlier system.

Test of Integration.

After the successful completion of the module testing, this forms the major part of the testing process. Once found that the modules are working perfectly as individuals, it is now time to integrate it together into a new system and then proceed with the testing. The tests of integration as it is called varies vastly from the module testing process as this involves the whole system as such and so impact on another module due to an action on a certain module is studied upon. This may give rise to some surprising results also and sometimes one has to back track to system development, change the system, carry out module testing and then start the tests of integration all again. This kind of testing is the one just before the final hand over of the system to the users or organization.

. Refinements Based on Feedback

The changeover to the new system does not altogether complete the system life cycle. system is now ready for refinements based on the feedback. The user who is in constant ch with the system finds out minor problems that evaded the entire process mentioned above. se minor errors and other suggestions that can enhance the system and make it more fortable and user friendly are given to the development team as feedback. The team then sits n to correct them and attach these corrected modules into the system and remove the iously working modules. As such this process of refinements on feedback is not a inuous job, it arises only when the users have some feedback. But the truth is that the users always find one or two minor bugs and suggestions with the system. As a result the ements process will continue on for a long time to come.

The above said process can be otherwise called as the backup activities or the system ntenance activities. In actual terms theses activities are done as and when the need arises and the full implementation and testing process the developer gets in touch with the system only n feedback is obtained from the users.

CONCLUSION

The software Remote Administration was developed for Logic Software Solutions. It was ed, debugged and successfully implemented in the company. Since it is easy to use only less ning was needed to use the software.

This software is actually intended to monitor the client systems in Logic Software utions. This software helps project leader to monitor clients screen and to know that their ordinates are working properly or not. This software provides project manager to clear doubts l send messages to them without going near to them. The mouse and keyboard of the ninistrator's machine are logically transferred to the remote computer. As soon as the screen the remote machine changes corresponding changes are updated in the administrator's chine. So he can work with the remote computer as if he is sitting in front of it. The main vantage of this software is viewing more than one client's screens at a time. This software is y easy to train the users.

SCOPE FOR FUTURE DEVELOPMENTS

The system has achieved all the objectives laid down at the beginning of the project. The duct is likely to be released by fall of August in the market. Because of the technological vancements and better availability of facilities the list of needs are always set to grow. The tem is built in such a way that it is flexible for the future enhancements. Facilities such as File nsfer and Video Conferencing may be added to the system as future enhancements. Future difications, which can improve the performance of the system, may be added to the software. rver can send message to all clients system at a time can also be added to software. The speed updating screens of clients can also be included in software. From the server system ministrator can access all the software's that is in the client system.

BIBLIOGRAPHY

ystem Analysis and Design : Elias M Awad
Galgotia Publications

Mastering Visual C++ 6 : Michael J Young
BPB Publications

Mastering Windows NT Server 4 : Minasi
BPB Publications

Visual C++ 6 in record time : Steven Holzner
BPB Publications

Visual C++ 6 from the ground up : John Paul Mueller
Tata Mc Graw Hill

Windows Sockets Network Programming : Dave Shute & Bob Quinn
Addison Wesley Longman

Microsoft Developers Network July2000 : Microsoft Corporation

Fig. A.1. Architecture

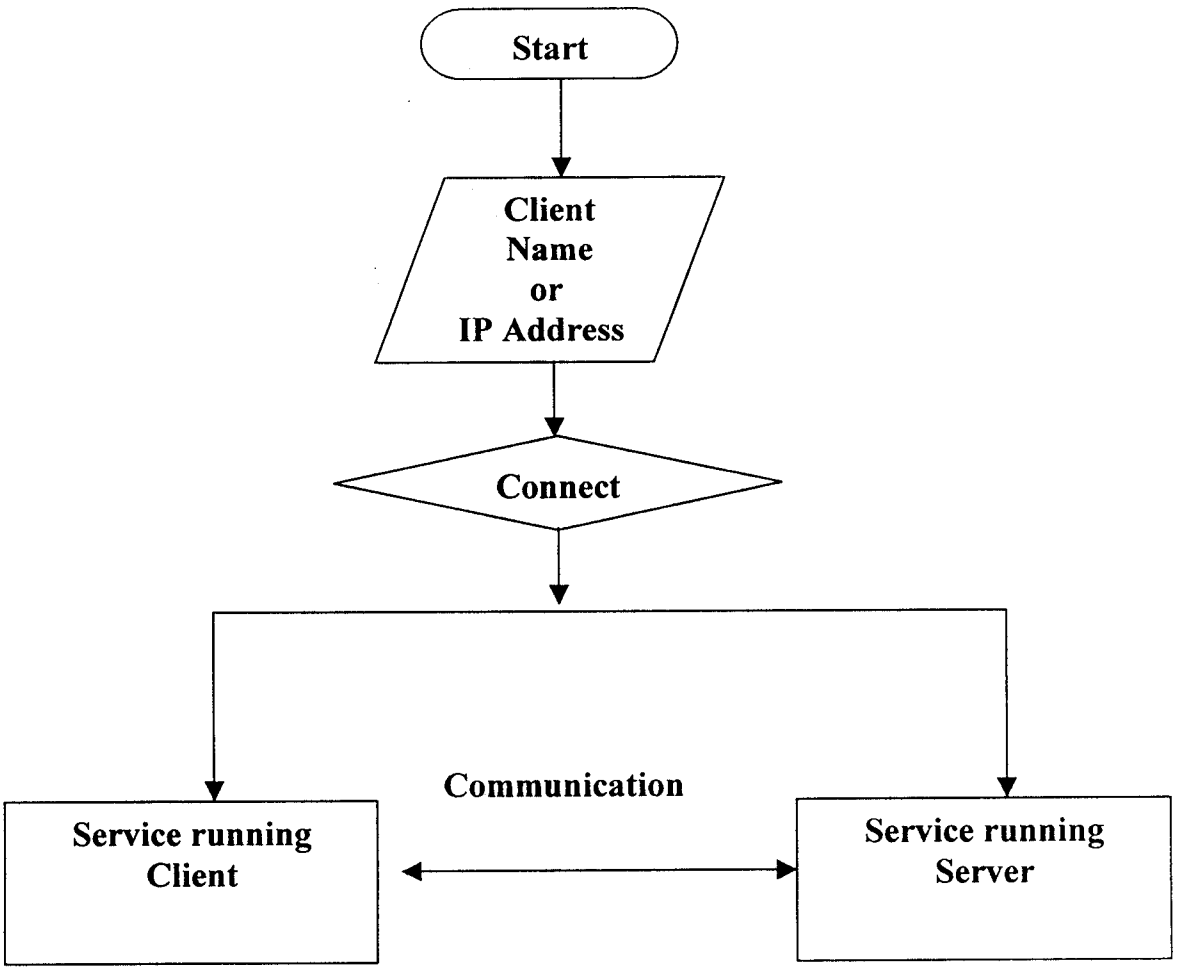


Fig. A.2. Work Flow Diagram

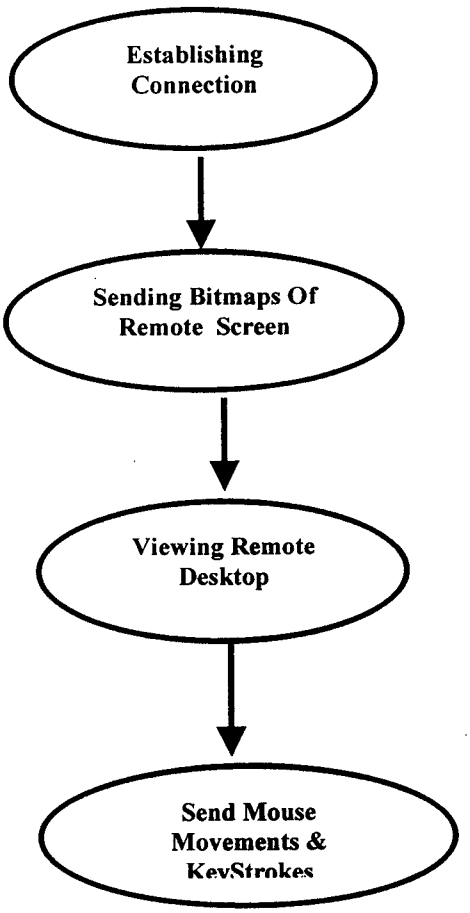


Fig. A.3. Process Flow Diagram

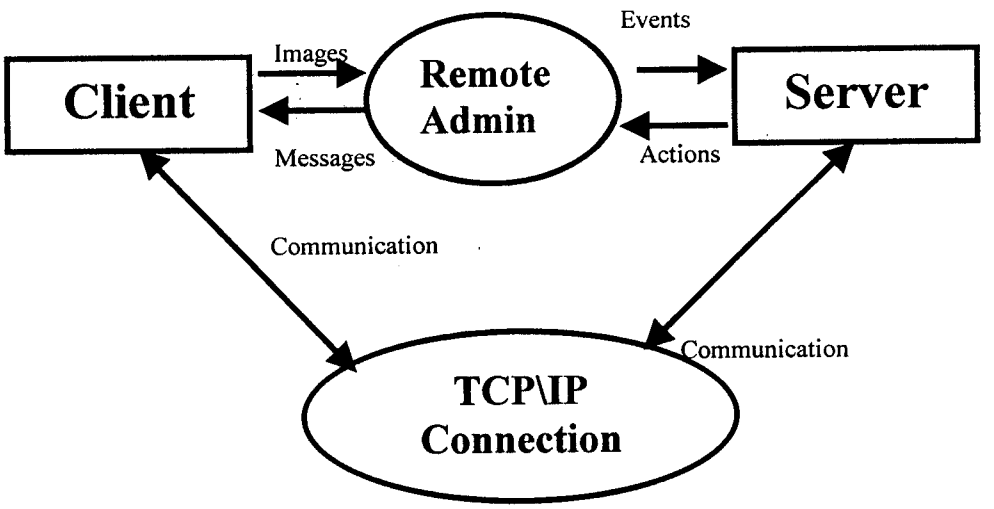
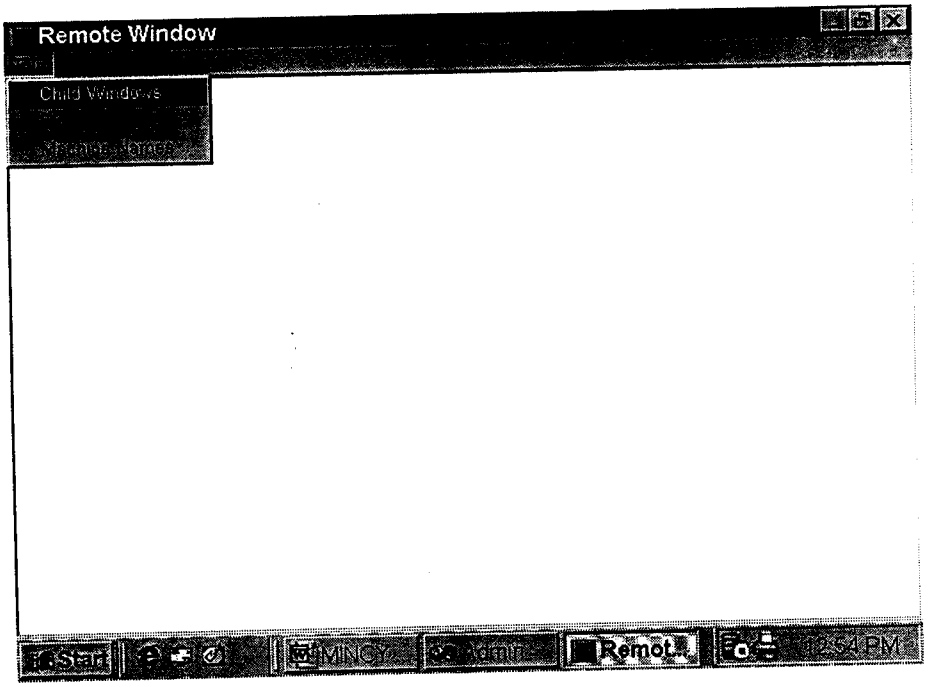


FIG: B.1 MAIN SCREEN OF SREVER



Description:

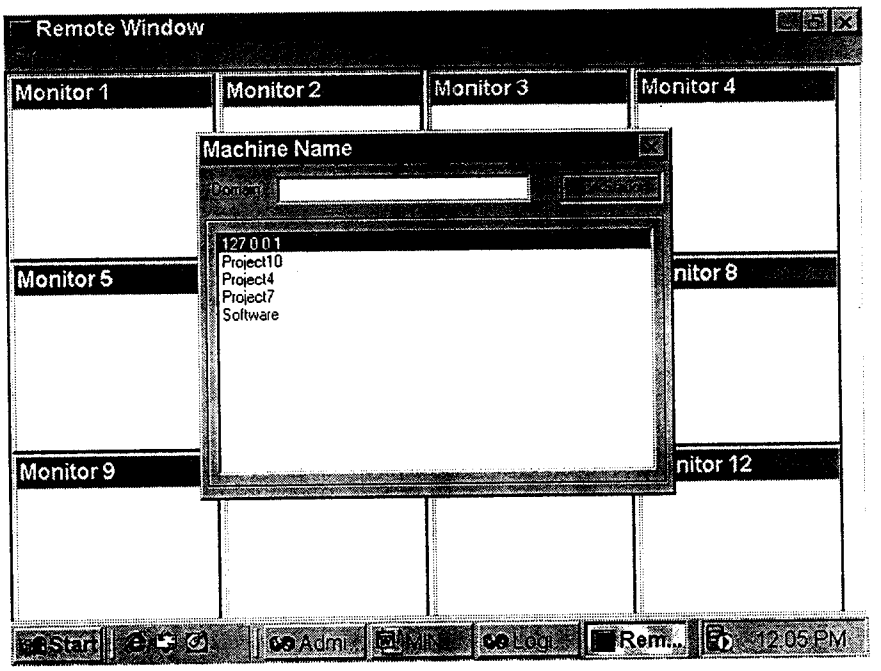
This screen displays the main menu items.

FIG: B.2 CHILD WINDOWS OF 12 CLIENTS

Description:

The screen that displays 12 blank child windows

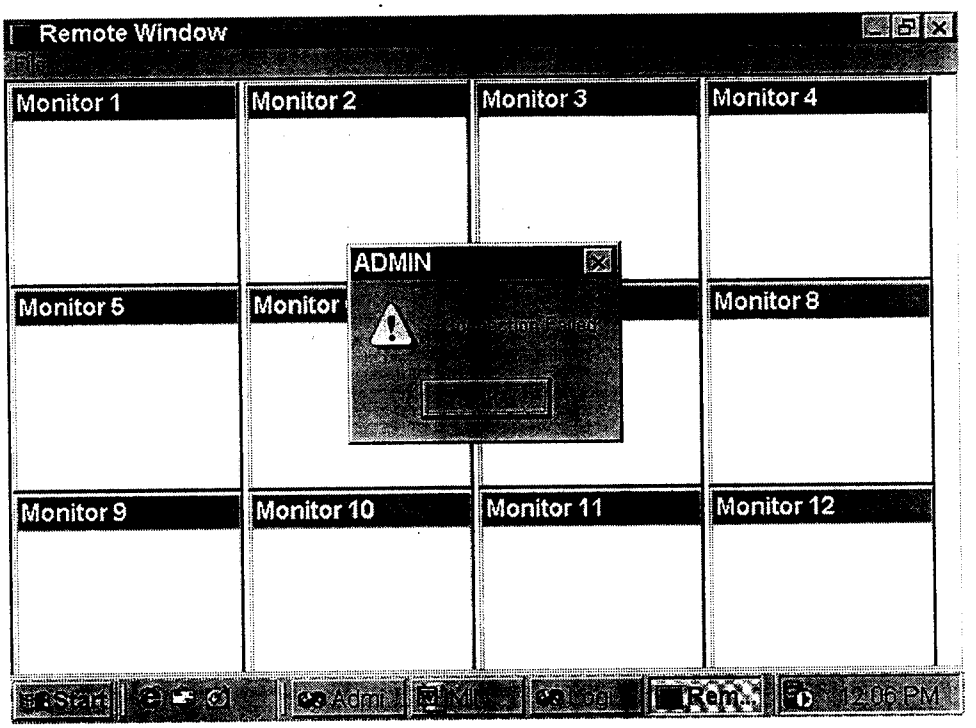
FIG: B.3 SELECTING CLIENT



escription:

The screen that displays a dialog box for selecting client to be connected.

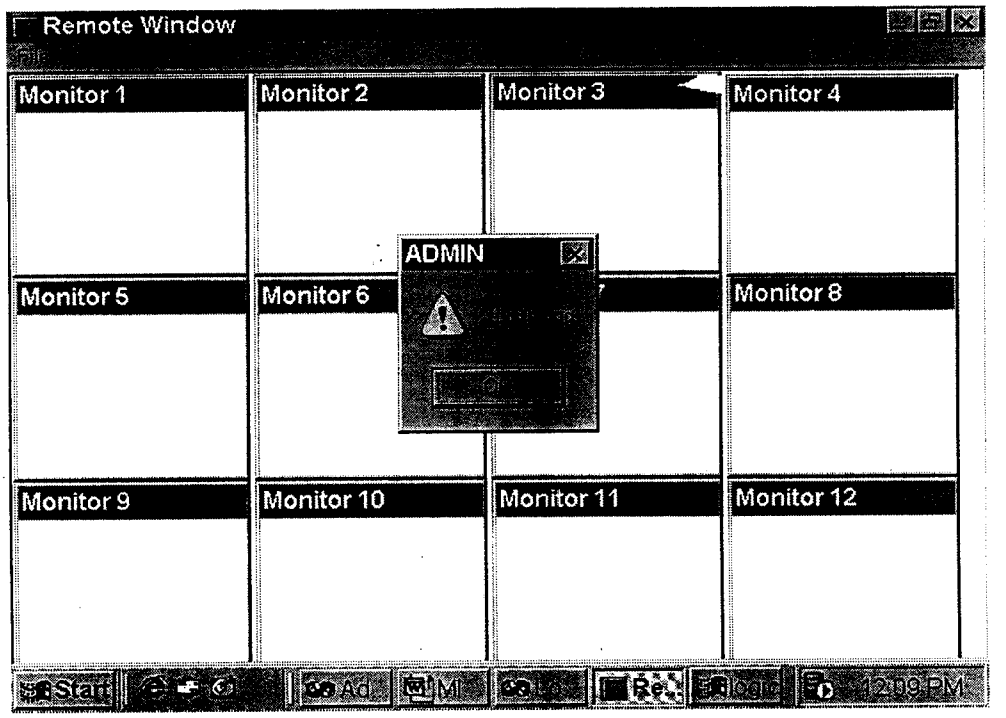
FIG: B.4 CONNECTION FAILED



Description:

The connection fails due to client software not currently being run on remote PC.

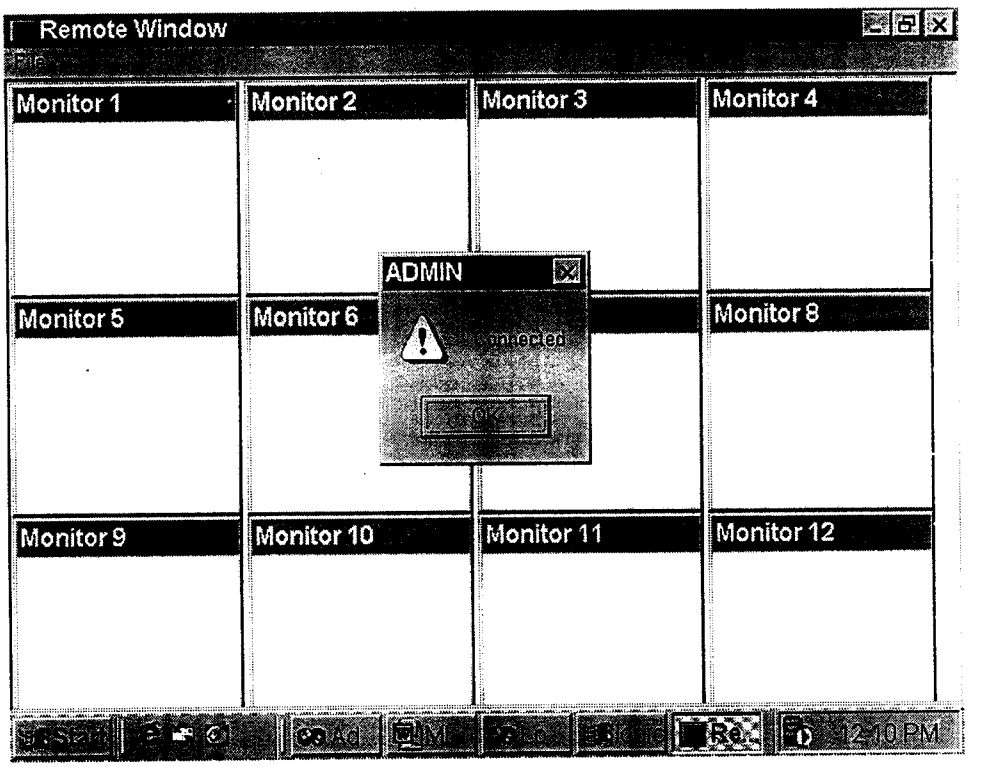
FIG: B.5 LISTENING TO PORT



Description:

This screen shows a dialog box when server listens to port, which we have chosen.

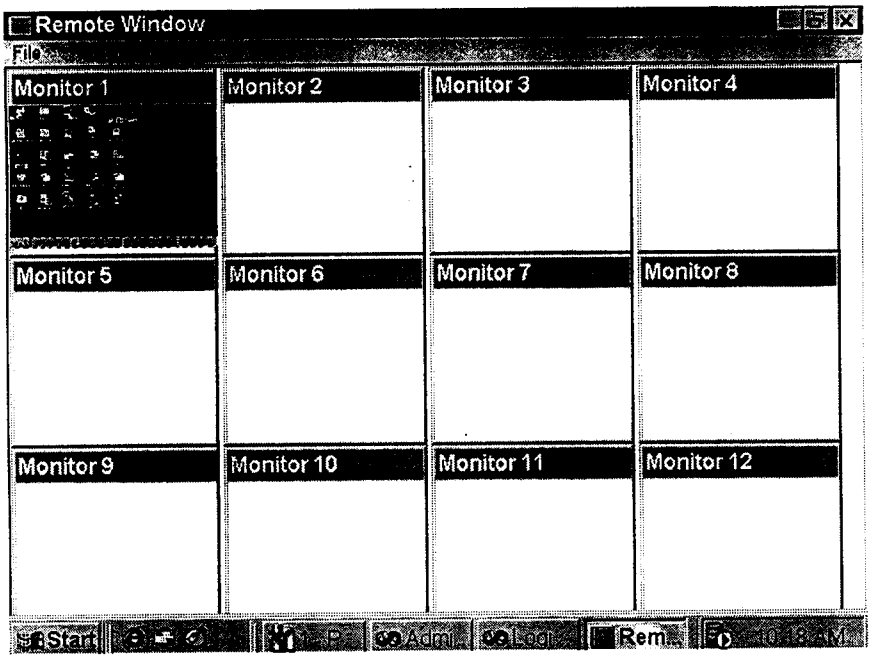
FIG: B.6 CONNECTION SUCCESS



escription:

This screen displays a dialog box when server connected to client.

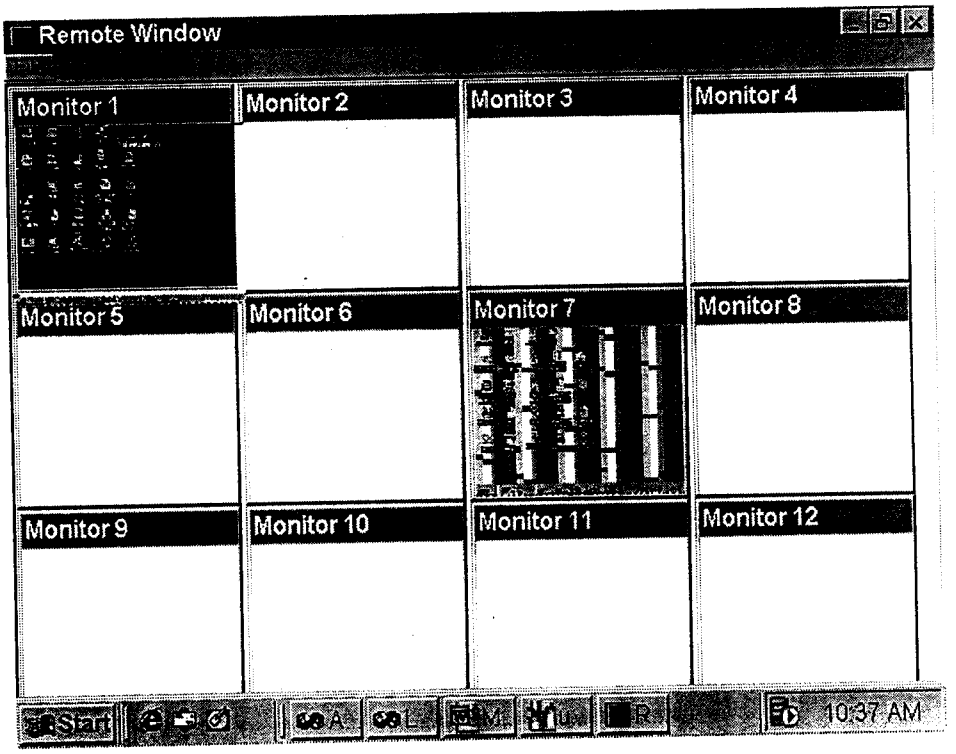
FIG: B.7 DESKTOP OF ONE REMOTE CLIENT



escription:

The screen showing the desktop of the remote client, which we are connected

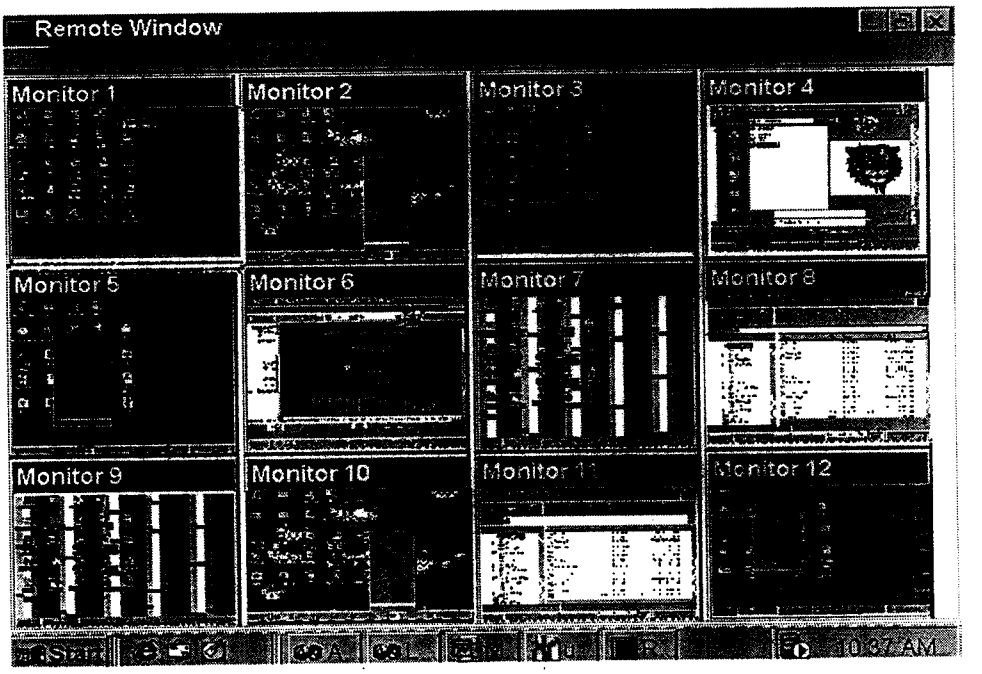
FIG: B.8 DESKTOP OF TWO CLIENTS



escription:

This screen shows two remote clients screen, which we are connected.

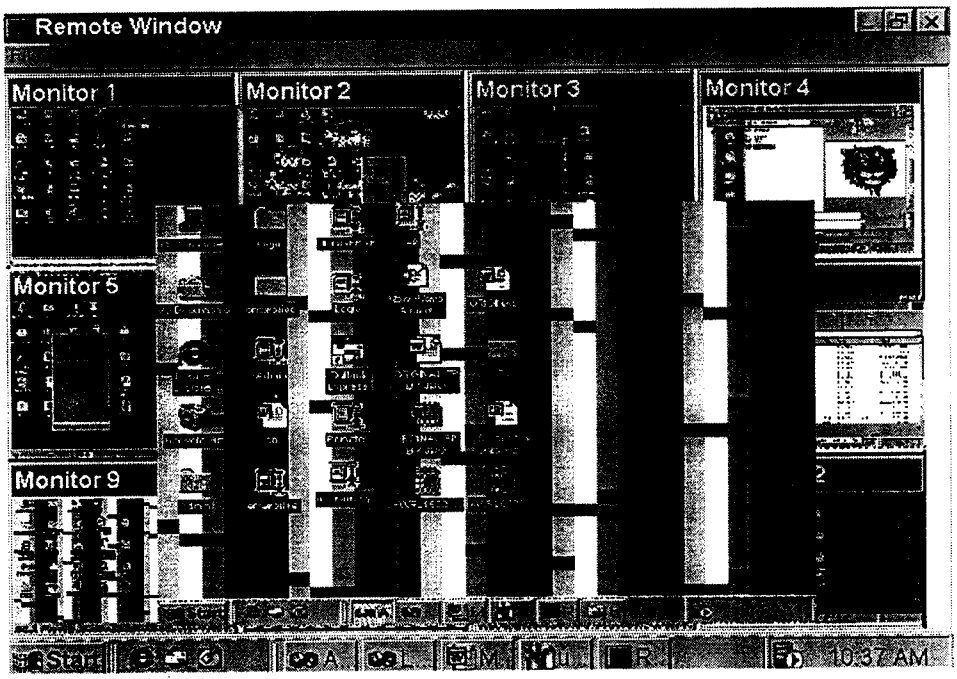
FIG: B.9 DESKTOP OF 12 CLIENTS



Description:

This screen shows the 12 remote clients screen running on server at a time.

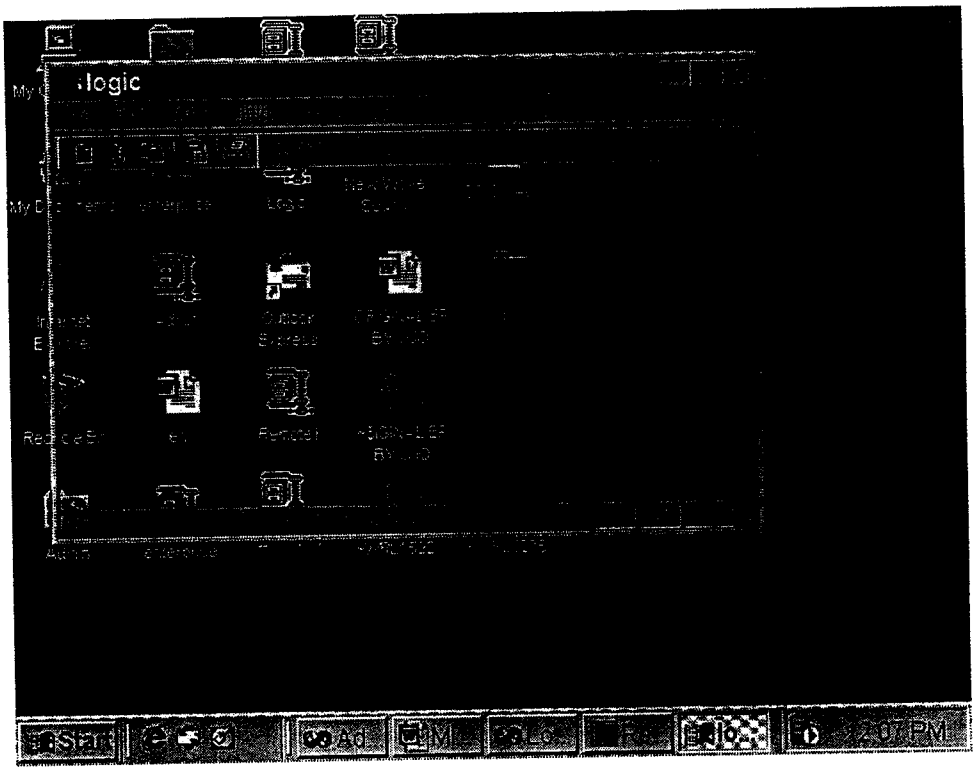
FIG: B.10 ENLARGED IMAGE OF CLIENT SCREEN.



Description:

This screen shows the enlarged image of 9th client screen.

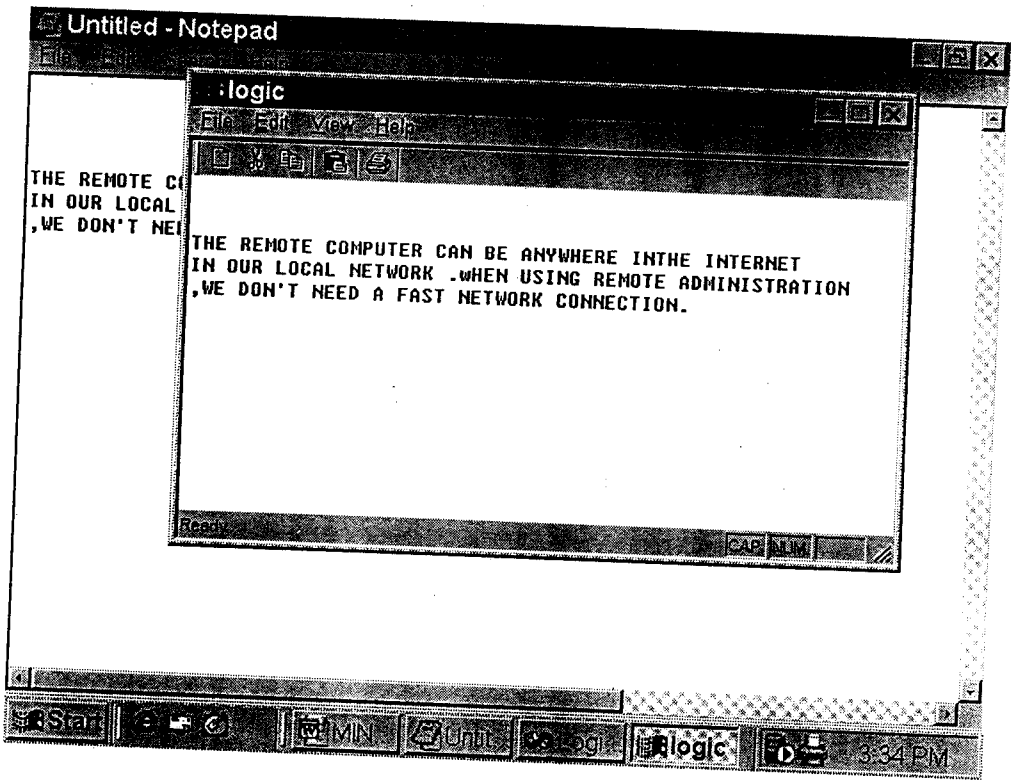
FIG: B.11 MAIN SCREEN OF CLIENT



Description:

This screen displays how the client screen is captured. Here actually client software stores the desktop as a bitmap in a shared disk drive.

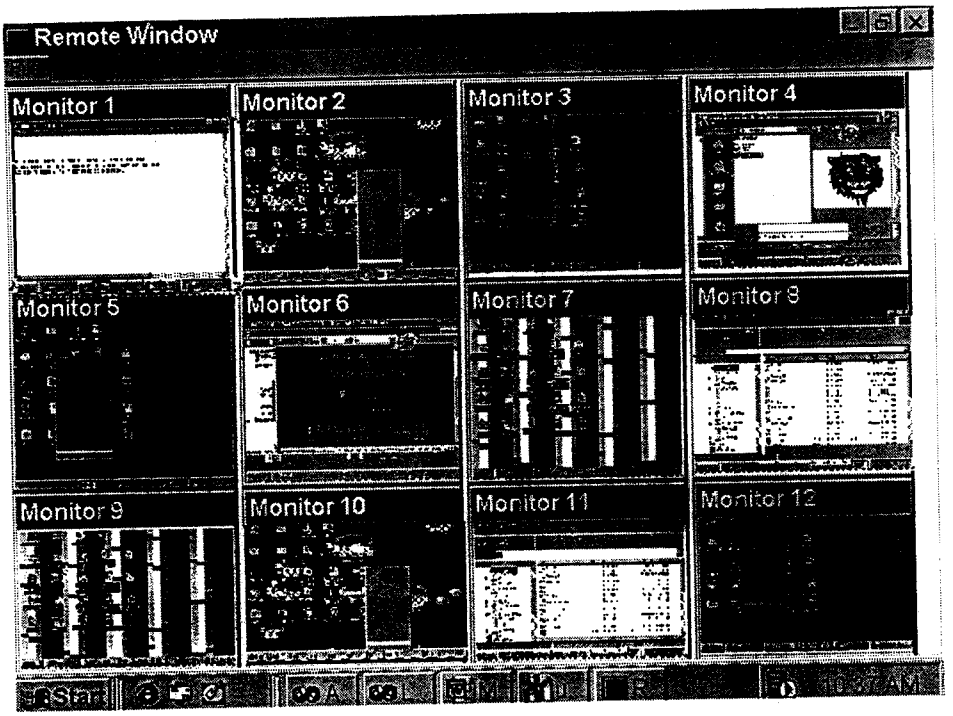
FIG: B.12 KEYSTROKES TRANSFERRING BY CLIENT



cription:

creen displays how the client software transfer the keystrokes made by client.

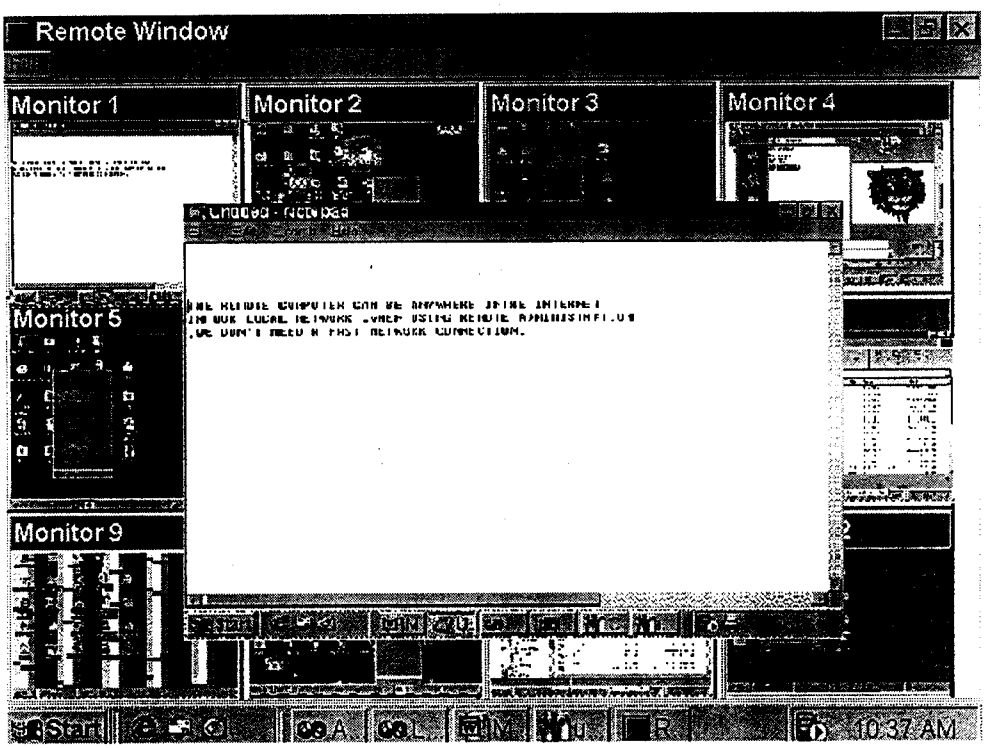
IG: B.13 VIEWING KEYSTROKES TRANSFERRING SCREEN



scription:

This screen shows all the keystrokes made by the 1st remote user, as he is typed.

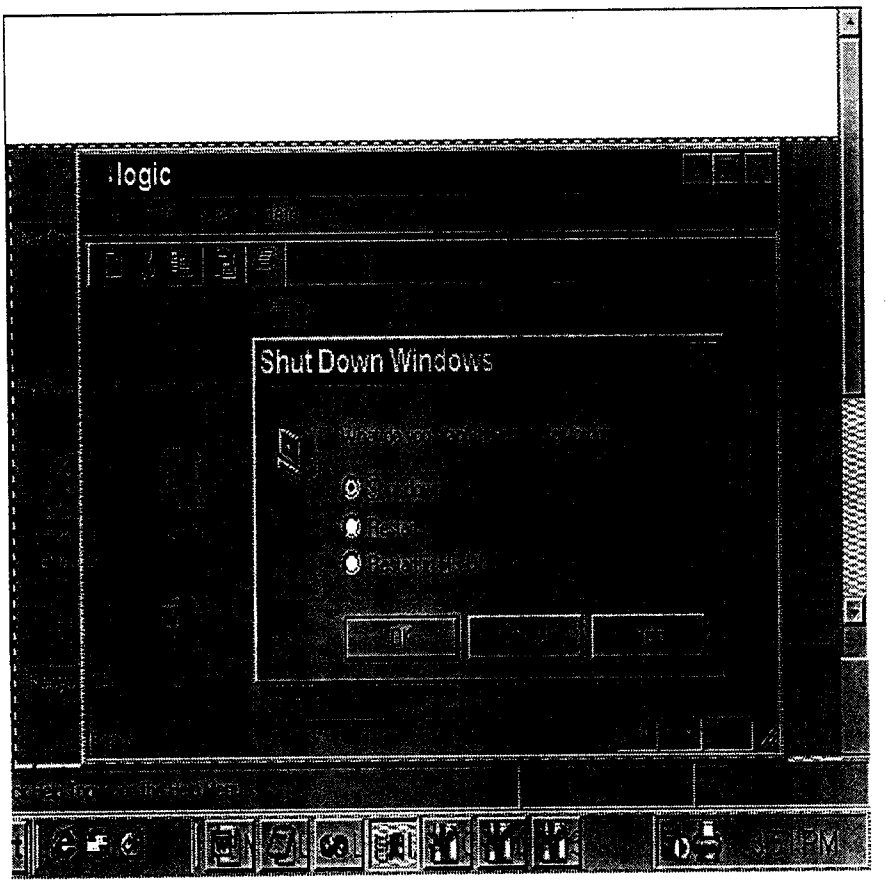
FIG: B.14 ENLARGED SCREEN OF KEYSTROKES TRANSFERRED



scription:

This screen shows the enlarged form of keystrokes made by the remote user viewing on server system. Thereby er can making changes as well as controls the client screen from server system.

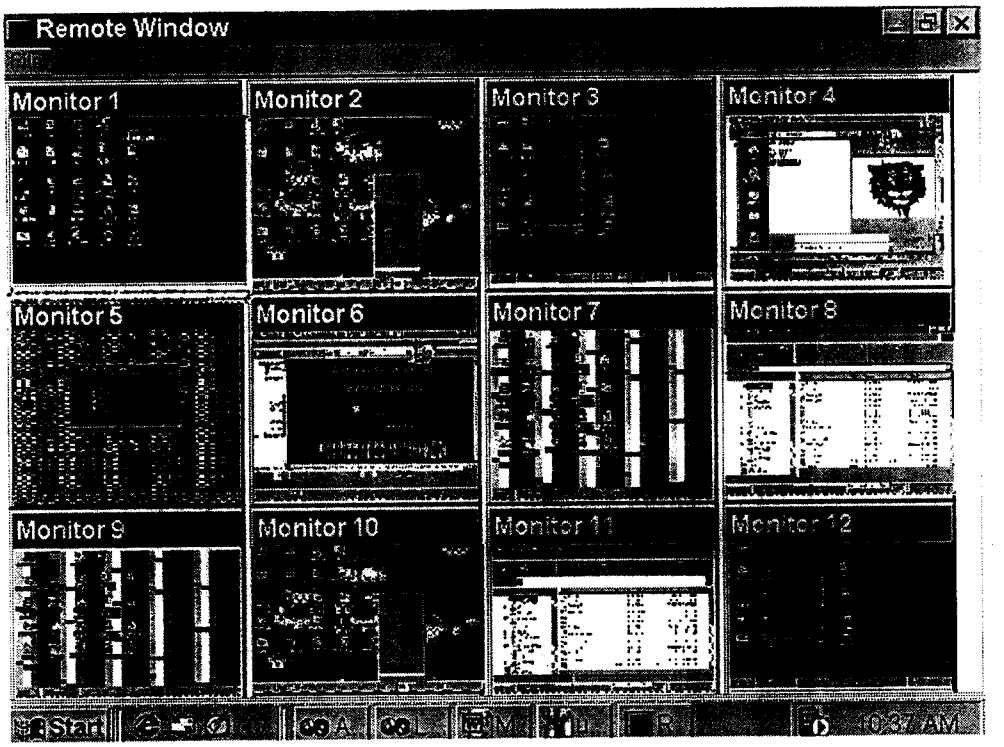
FIG: B.15 SHUTDOWN THE CLIENT SYSTEM



scription:

This screen shows capturing the client screen by client software for shutdown .

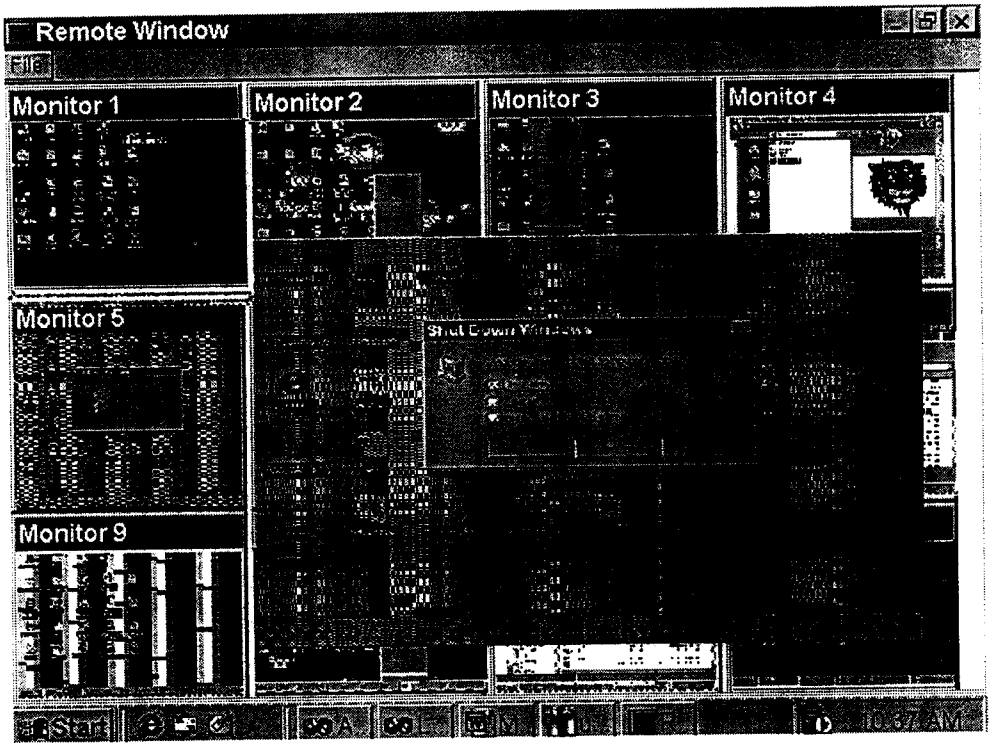
IG: B.16 VIEWING THE SHUTDOWN SCREEN SENT BY CLIENT



scription:

This screen shows the shutdown screen that has been sent by client for shutdown viewing on server
en.

FIG: B.17 ENLARGED IMAGE OF SHUTDOWN SCREEN



cription:

This is the enlarged image of shutdown screen that has been sent by client viewing on server and lets him tdown the client system.