REMOTE INSTALLATION AND ADMINISTRATION

A PROJECT REPORT P- 717

Submitted in partial fulfillment of the requirement for the award of the degree of

Master of Science (Applied Sciences -Computer Technology)

of Bharathiar University, Coimbatore.

Submitted By

D.DIVYA

0037Q0031

Under the guidance of

Mr.K.RamaSubramanian, M.C.A,

Computer Science and Engineering Department,

Kumaraguru College of Technology,

Coimbatore.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE

APRIL 2002

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

Affiliated to Bharathiar University

Coimbatore-641006

CERTIFICATE

This is to certify that the project entitled

REMOTE INSTALLATION AND ADMINISTRATION

Done By

D.DIVYA

0037Q0031

Submitted in partial fulfillment of the requirement for the award of the degree of

Master of Science (Applied Sciences-Computer Technology)

of Bharathiar University,Coimbatore.

S Thangasamy 15/4/02

Prof.Dr.S.Thangasamy
Head of the Department

Mr.K.RamaSubramanian
Internal Guide

Submitted to University Examination held on 23.04.02

R. Dinesh 25/4/02

Internal Examiner

External Examiner

UNITED INFOTECH

7B, SRI RAM LAYOUT, THADAGAM ROAD, COIMBATORE - 11. ✆ : 455761 CELL : 98422 55762

DATE:09/04/2002

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Miss.D.Divya (2kmct07)** II M.Sc Applied Sciences-Computer Technology of Kumaraguru College of Technology has completed a project entitled **"REMOTE INSTALLATION AND ADMINISTRATION"** at UNITED INFOTECH, COIMBATORE successfully during the period January-2002 to April-2002 using **Visual C++.**

I wish her all success in her future endeavors.

for UNITED INFOTECH

UNITED INFOTECH,
7B, Sri Ram Lay-Out,
Saibaba Colony, Thadagam Road,
COIMBATORE 641 011.
Ph: 455122

Mr.S.SHANMUGAM

DECLARATION

I, hereby declare that the project entitled '**REMOTE INSTALLATION AND ADMINISTRATION**' submitted to Bharathiar University as the project work of M.Sc. (Applied Sciences-Computer Technology) degree is a record of original work done by me under the supervision and guidance of Mr.Ragupathy, Project Manager, United Infotech.Coimbatore and Mr.K.RamaSubramanian,M.C.A., Kumaraguru College of Technology, Coimbatore and this project work has found the basis of award of any Degree/Diploma/Associateship/Fellowship or similar title to any candidate of any University.

Place: Coimbatore
Date : 25.04.2002

D.Divya

Signature of Guide 15/04/02

Mr.K.RamaSubramanian.
Lecturer,
Kumaraguru College of Technology,
Coimbatore-641006.

ACKNOWLEDGEMENT

I am bound to express my gratitude to **Dr.K.K.Padmanaban,** B.Sc (Engg), M.Tech., Ph.D, Principal, Kumaraguru College of Technology, Coimbatore, for constant encouragement throughout my course.

I wish to thank **Prof.Dr.S.Thangasamy**, B.E (Hons), Ph.D, Head, Department of Computer Science and Engineering, for supporting me to pursue new goals and ideas.

I admit my heartfelt thanks to my tutor, **Mr.R.Dinesh**, B.Tech, M.S., Assistant Professor, for being supportive during the tenure of my project.

I express my earnest gratitude to my internal project guide, **Mr.K.Ramasubramanian**, M.C.A, Lecturer in Computer Science & Engineering Department for his creative support and timely advice in completing this project.

I extend my sincere thanks to **Mr.Shanmugham**, Managing Director, United InfoTech, Coimbatore, for providing me with an opportunity of working with my choice and encouraging me through the project.

I owe much to **Mr.Raghupathy**, Project Manager, United InfoTech, Coimbatore, for his immense help and technical support in completing this project.

Last but not the least, I extend my heartfelt thanks to all my well wishers for their cooperation in my endeavor.

SYNOPSIS

The project entitled **"Remote Installation and Administration"** comprises of two modules, namely

1. Software Installation.

2. Network Administration.

The Software Installation module involves installing software over multiple clients in a network simultaneously. The project focuses in providing a user-friendly environment with efficient results by reducing execution time.

It comprises of two programs, namely a client and a server. In a network, a system loaded with the server program act as a server for this particular application. All the systems loaded with the client program act as clients.

Once the server name is given in the client, the connection is established. At the server end, the client's name gets displayed in a list box as and when they get connected to the server. The user can view the drives available of a particular client and its type .The user selects the clients and their respective drive in which the software has to be installed. Once the selection is done, the user specifies the source path of the software to be installed which marks the beginning of the installation process at the client end and at the termination of this process the registry is updated for all the selected clients simultaneously.

Network Administration module manages clients in a network which has the following options.

Shut Down	- shuts down the selected clients.
Log Off	- logs off selected clients.
Restart	- restarts selected clients.
Message Passing	- between clients and the server.

CONTENTS

1. INTRODUCTION

1.1. ABOUT THE ORGANIZATION

United InfoTech. a software development cum training center witnessed its humble beginning a way back five years. United InfoTech believes in a world of quick changing frontiers to be a successful enterprise. They are committed to reach out their clients with IT products and solutions specific to their business needs.

United InfoTech use their expertise and experience to help clients to anticipate, initiate and manage changes better than others. This commitment and supportive relationship has helped clients to win.

United InfoTech believes in recognizing and encouragement the merit of its people as individuals. The environment thus facilitates freethinking, experimentation and innovation. With an emphasis on personal responsibility, United InfoTech provides an atmosphere that supports original thinking.

The most striking feature of this growing organization is the commitment of the staffs and project guides, who are always trying to give the best of their knowledge to the students.

1.2. INTRODUCTION TO THE SYSTEM

The Installation of a particular software in the existing network environment, over a large number of systems starts with manually reaching out to every individual system to accomplish the task of installation. In case of restarting the system after installation the administrator has to again reach out to every individual system. This is obviously a time consuming and a hectic process to the administrator.

To ease out an administrator's job of handling the above situation in a versatile manner the idea of remotely installing software was given a thought and thus developed. This view was also extended towards administration, which allows the administrator to capably handle the clients. Thus the design was sketched out into two modules, namely Software Installation and Network Administration.

Being an excellent system side development tool VC++ was used in order to deliver improved support for reuse, registry handling, tool interoperability and resource management.

Installing software over multiple clients in a network simultaneously from the server end was captured in the Software Installation module. It facilitates the administrator to select/deselect-connected clients, to choose among drives of every client, view their drive types and keep in track with the installation process.

The administrator is additionally provided with features to manage clients in a network remotely as proposed in the network administration module.

The administration module provides with the following features as listed below

Shut Down	- shuts down the selected clients.
Log Off	- logs off selected clients.
Restart	- restarts selected clients.
Message Passing	- between clients and the server.

1.3. SOFTWARE OBJECTIVE

The software could be remotely installed simultaneously on selected clients from the server in a network. The network can be administered with the following options as *Shutdown, Logoff, Restart* and *Message Passing*.

1.4. SOFTWARE SCOPE

The installation of the software and administration of the network are done remotely from the server, which aids extensive and flexible use of the network. The workload of the network administrator is extremely reduced.

1.5. MOTIVATION

The idea of remotely installing software in a much more dynamic manner than the usual procedures followed in labs during installation made me get along with the project as favored by my company.

1.6. EXISTING SYSTEM

Installation Module

The Softwares are installed in every system in a network individually.

Even in a shared mode, the user has to move to each and every system and run the setup.

Administration Module

The facilities such as Shutdown, Logoff, Message Passing and Restart are available only with latest versions of operating systems like Windows 2000, Novell, etc wherein a particular system cannot be restricted from the above operations.

These options are not available within Windows 98 and Windows NT, which are most commonly used.

1.7. PROPOSED SYSTEM

Installation Module

The Softwares could be remotely installed from the server side alone.

Administration Module

The facilities could be used as an attachment with Windows 98 and Windows NT Operating System.

Operations such as Shutdown, Logoff, Message Passing and Restart can be evaded to a particular system over the network.

2. PROGRAMMING ENVIRONMENT

2.1. Software Requirements

Language	:	VISUAL C++ 6.0
Operating System	:	Windows 9x, NT

2.2. Hardware Requirements

Processor	:	PENTIUM II
CPU Speed	:	600 MHz
RAM	:	64 MB
CD ROM	:	52X
Hard Disk	:	4 GB
Keyboard	:	101Keys

2.3. Additional Requirement : LAN Networking.

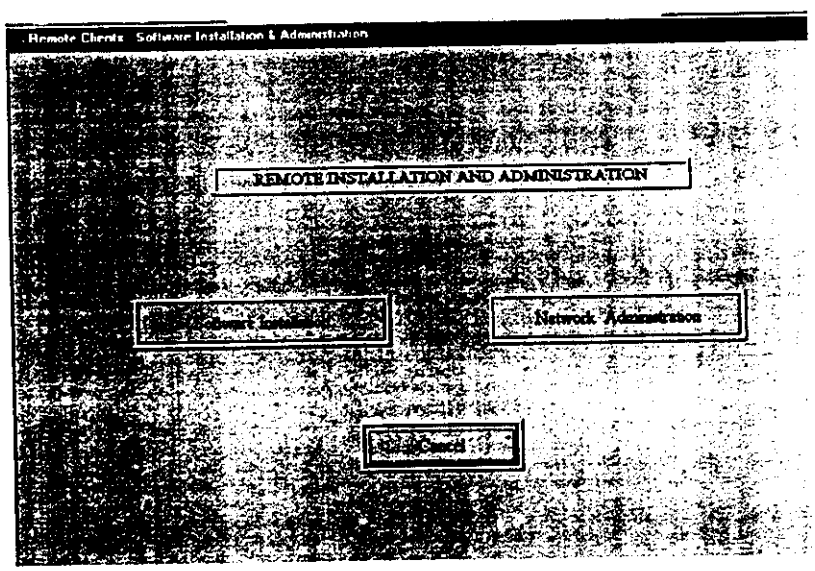
3. SYSTEM DESIGN

3.1. Input Design

To enter the various data required for the system, 3 input screens are designed, namely a home screen, Installation and an administration screen. The various controls such as list box, edit box, command buttons and a progress bar are used for designing the screen by which the user can easily follow. The input to the system was designed such that the required information can be collected and manipulated accordingly by the server and the client as per requirements.

The goal of the input design is to make the data entry easier, logical and free from errors. The application is being developed in a user-friendly manner. The forms are designed in such a way that during processing the cursor is placed in the position where the data must be entered. The user is also provided with an option of selecting an appropriate input from the list values.

The Home screen is provided for the user to choose between the two modules as shown below,



3.1.1. Software Installation

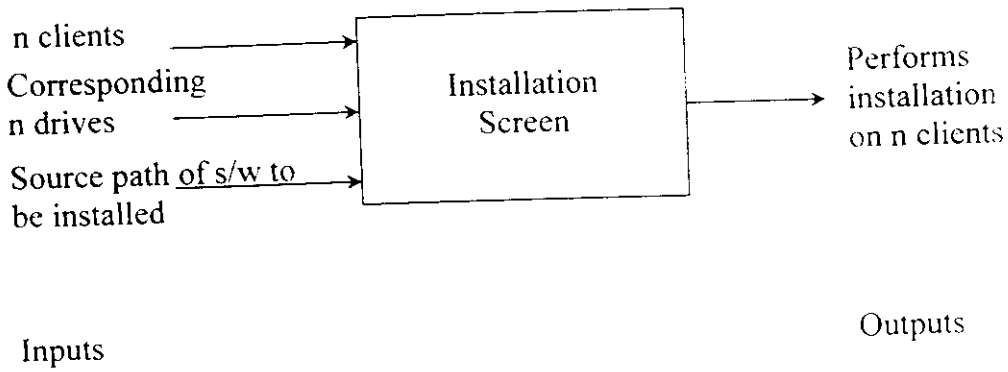


Figure 1.1 – Installation

The screen for the Software Installation module requests the user to provide with the information as shown in the Figure 1.1 to achieve the target of installation on multiple clients simultaneously.

A list box shows the list of system names of connected clients. The items of the list box are dynamically added or removed as and when a client gets connected or disconnected respectively. Once the user selects a particular client another List box displays available drives in that particular system.

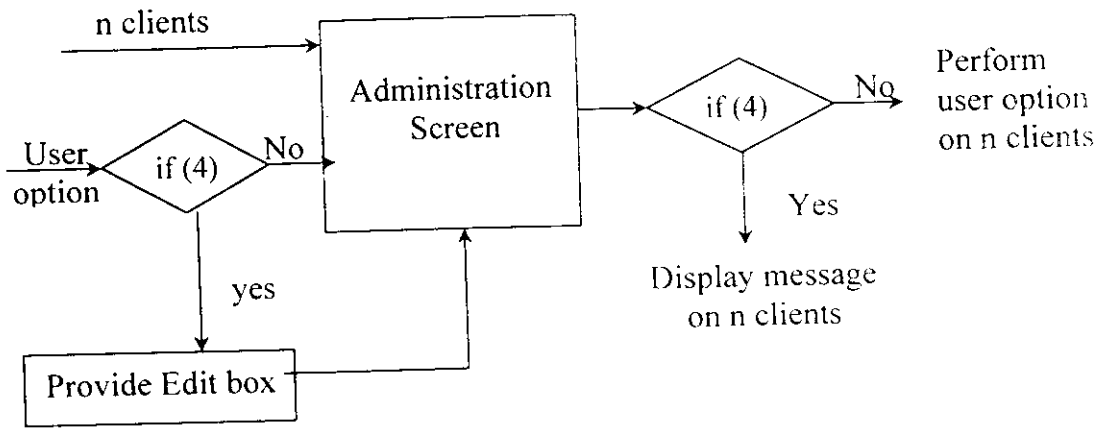
On selecting a particular drive, the drive type (fixed/network) is displayed in a textbox. Once the user clicks the add button the selected system name and its corresponding drive will be displayed in a list box as a list of selected clients. The user repeats the above process for the required clients. In case the user decides to revoke the clients that are selected, he can perform it by clicking the remove button after selecting the particular client from the list box, which contains the list of selected clients.

A text box is provided for the user to enter the source path (the path where the required software was installed in the server) of the software to be

nstalled. The process of installation is commenced on the click of the Install button. Another text box is provided which displays the path of the files currently copied from the server to the client.

To indicate the user about the progress of installation process, a progress bar is provided. The newly added details of the server's registry as a result of the installation process are reflected on the client's registry on the click of the Update registry button.

3.1.2. Network Administration



Inputs

Outputs

User option - Shutdown (1)/Logoff (2)/Restart (3)/Message Passing (4)

Figure 1.2 - Administration

The administration module deals with listing out the clients connected to the server and the various operations that can be performed on them. At this point, the input from the user would be the selection of n clients from the list and the required operation as shown in figure 1.2. To provide the user with a standard way to make selections, carry out commands, and perform input and output tasks, appropriate controls have been incorporated.

A list box shows the list of system names of connected clients. The items of the list box are dynamically added or removed as and when a client gets connected or disconnected respectively. List box is suitably designed to support multiple selections so that the user can choose more than one client at a time.

Various administrative operations are provided in the form of radio buttons for the user to efficiently select one operation at a time. The options include *Shutdown, Logoff, Message passing and Restart*. Once the option is selected the perform button is clicked and the operation is carried out on the selected clients. A text box is provided wherein the user can type the message to be sent to the clients chosen.

3.2. OUTPUT DESIGN

The user is constantly provided with messages during every event that is invoked. Outputs generated by the system during the process to the end users are also promptly displayed.

At the client's side, appropriate messages are fired when the user attempts to gets connected to the server say, *connection established* or *connection failed*. If the server quits, a notification is given on the client side.

At the server side, with respect to the Administration module, the end user is provided by the confirmation message about the completion of the operation selected by him.

With respect to the Installation module, to indicate the user about the progress of installation process, a progress bar is provided. On the termination of the file transfer process, an indication is given to the user that the file transfer is completed. On the completion of the installation process, the server displays the status of installation of each and every client selected.

4. SYSTEM DEVELOPMENT AND IMPLEMENTATION

4.1. DEVELOPMENT ENVIRONMENT

Microsoft Visual C++ 6.0

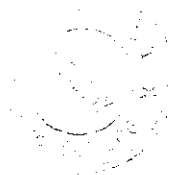
Visual C++ 6.0 includes the Microsoft Developer Studio Integrated Development Environment (IDE). This environment is the centerpiece of most any interaction to create Visual C++ projects, including source file creation, resource editing, compiling, linking, debugging, and many other useful features that will make the development tasks much simpler.

Visual C++ is the first and foremost C++ compiler, made up of many components, which paves the way for efficient programming.

Project Workspace

Working with developer studio is working with Project workspaces. These workspaces represent a particular set of projects, which can represent anything from a single application, to a function library, or to an entire suite of applications. Each workspace may include any number of different projects that we want to group into a single workspace so that we can work closely with each separate project at the same time.

The project workspace (.dsw) file is responsible for maintaining all of the information that defines our workspace and the projects that we have included in it.



MFC AppWizard

The project was developed using Microsoft Foundation Classes (MFC).To support wide range of windows applications, MFC AppWizard is used to work with different projects.

Each of the AppWizards will guide in creating a new project, prompting various option selections along the way. MFC is a tremendous time saver, which creates a skeleton of any application. The project created by MFC AppWizard (.exe) will build without any further modifications and has all the setup for the application features selected.

Windows Sockets

As the project deals with network communications, MFC includes the usage of sockets. A "socket" is an endpoint of communication — an object through which our application communicates with other Windows Sockets applications across a network. It provides a mechanism to send and receive data.

The Sockets allows creating and maintaining a connection to a remote computer. Using the connection, both computers can stream data between themselves.

If we are creating a client application, you must know the server computer's name or IP address (**Remote Host** property), as well as the port (**Remote Port** property) on which it will be "listening." Then invoke the **Connect** method.

If you are creating a server application, set a port (**Local Port** property) on which to listen, and invoke the **Listen** method. When the client computer requests

a connection, the Connection Request event will occur. To complete the connection, invoke the **Accept** method within the Connection Request event.

Once a connection has been made, either computer can send and receive data. To send data, invoke the **Send Data** method. Whenever data is received, the DataArrival event occurs. Invoke the **Get Data** method within the DataArrival event to retrieve the data.

Registry

The Registry is a system-defined database that applications and system components use to store and retrieve configuration data.

The Registry is a hierarchically organized store of information. Each entry in this tree-like information structure is called a *key*. A key may contain any number of subkeys; it can also contain data entries called *values*. In this form, the Registry stores information about the system, its configuration, hardware devices, and software applications. A Registry key is identified by its name. Keynames consist of printable ASCII characters except the backslash (\), space, and wildcard (* or ?) characters. The use of Keynames that begin with a period (.) is reserved. Keynames are not case sensitive.

A value in the Registry is identified by its name. Value names consist of the same characters as key names. The value itself can be a string, binary data, or a 32-bit unsigned value.

Predefined Registry Keys

The Registry contains several predefined keys.

The HKEY_LOCAL_MACHINE key contains entries that describe the computer and its configuration. This includes information about the processor, system board, memory, and installed hardware and software.

The HKEY_CLASSES_ROOT key is the root key for information relating to document types and OLE types. This key is a subordinate key to HKEY_LOCAL_MACHINE. Information that is stored here is used by shell applications such as the Program Manager, File Manager, or the Explorer, and by OLE applications.

The HKEY_USERS key serves as the root key for the default user preference settings as well as individual user preferences.

The HKEY_CLASSES_USER key is the root key for information relating to the preferences of the current (logged in) user.

Under Windows 95, there are two additional predefined keys. The HKEY_CURRENT_CONFIG key contains information about the current system configuration settings. This key is equivalent to a sub key (such as 0001) of the key HKEY_LOCAL_MACHINE\Config.

The HKEY_DYN_DATA key provides access to dynamic status information, such as information about plug and play devices.

4.2. SYSTEM IMPLEMENTATION

The project entitled "**Remote Installation and Administration**" demands two ends of communication that is between a server and clients. To meet this need, MFC Socket (WinSock) was implemented.

The WinSock allows you to create and maintain a connection to a remote computer. Using the connection, both computers can stream data between themselves.

In the server application, a port is made to listen by invoking the *Listen* method. When the client computer requests a connection by providing the server name or IP address, the *Connection Request* event will occur. To complete the connection, the *Accept* method is invoked within the *Connection Request* event. A socket is created for each and every client.

As per the user's selection between **Software Installation** and **Network Administration**, the application proceeds thereby.

4.2.1. Software Installation

Initially, the names of folders, subfolders and files under the server system's current windows directory are read and written into a separate file. The similar process is repeated for registry contents.

As and when the client application gets connected to the server application, the client sends its system name and the available drives to the server. The server displays the system name in the list box and the drive details are collected in a buffer.

Once the user selects a particular client, the available drives of respective client is identified from the buffer, retrieved and displayed in a list box

as shown in Figure 1.3. On the selection of a particular drive by the user, the server identifies the socket corresponding to the selected client and sends the selected drive name to the client application through the identified socket.

On receiving the drive name from the server, the client application identifies the drive type (Network/Fixed) and sends it to the server. At the server side, the received drive type is displayed in a text box for the user's verification as shown in Figure 1.4. The selected client and its respective drive in which the software has to be installed is displayed in a separate list box on the click of the *Add* button as shown in Figure 1.5. The user has to repeat this process for the n number of clients. The user is also provided with a facility to deselect the clients with the help of the *Remove* button.

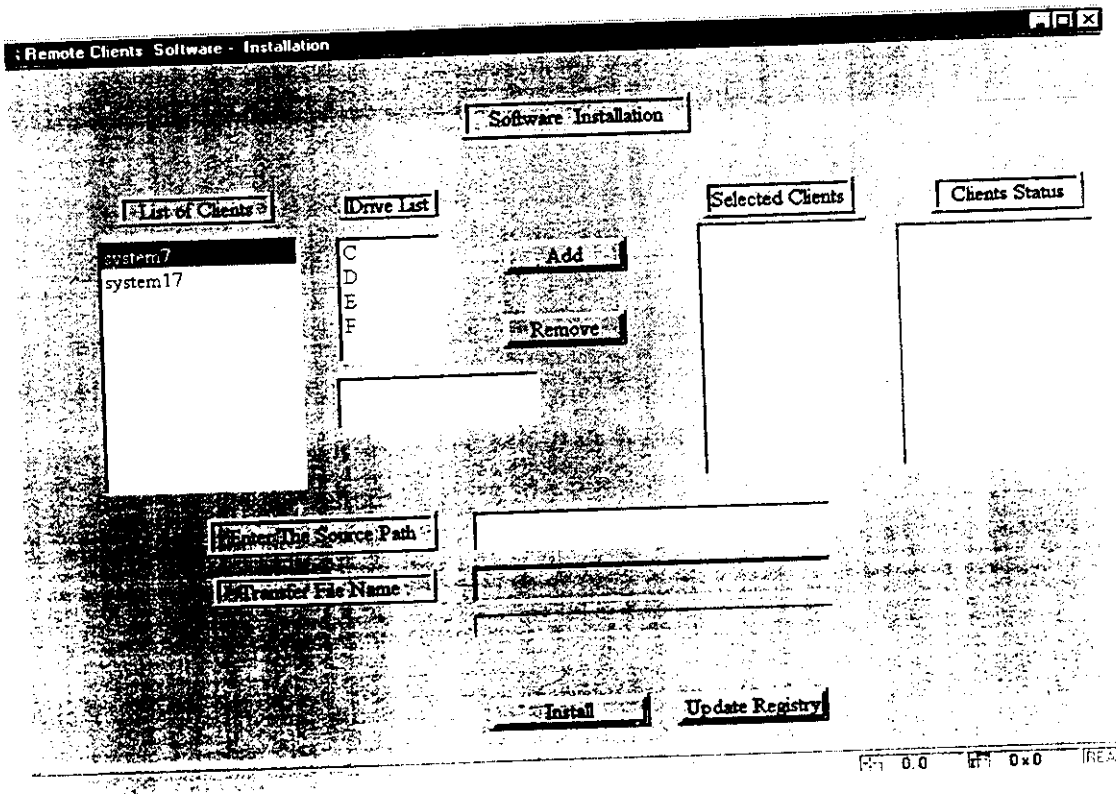


Figure 1.3

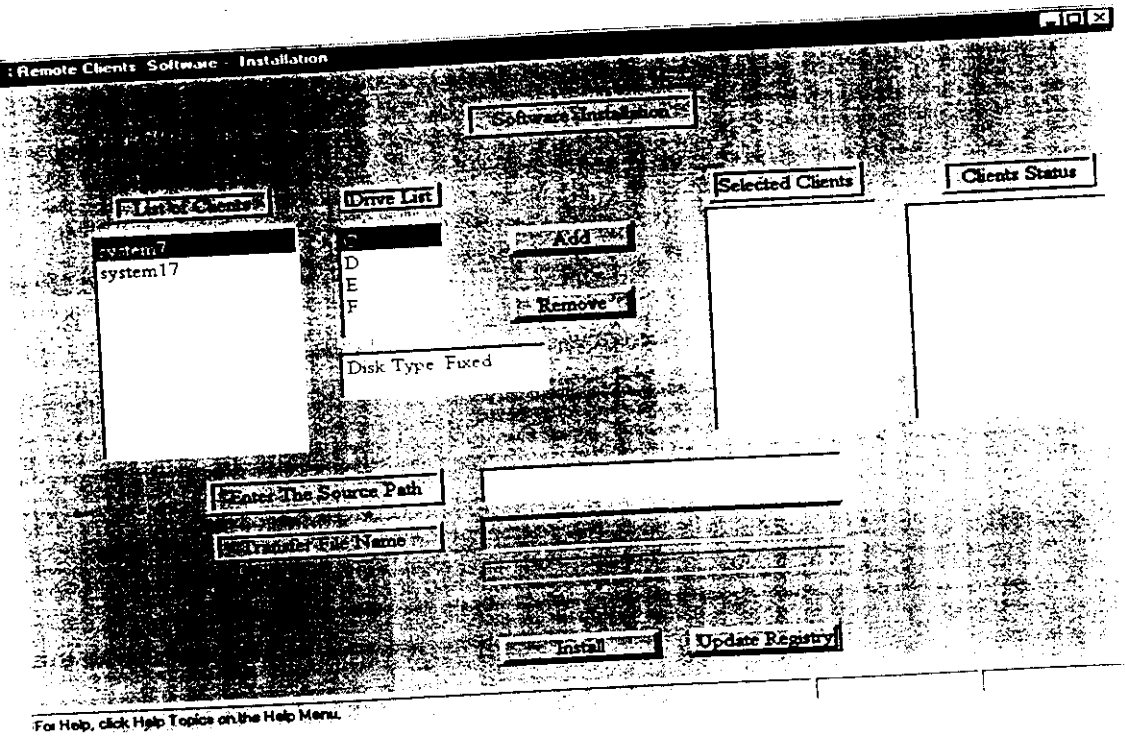


Figure 1.4

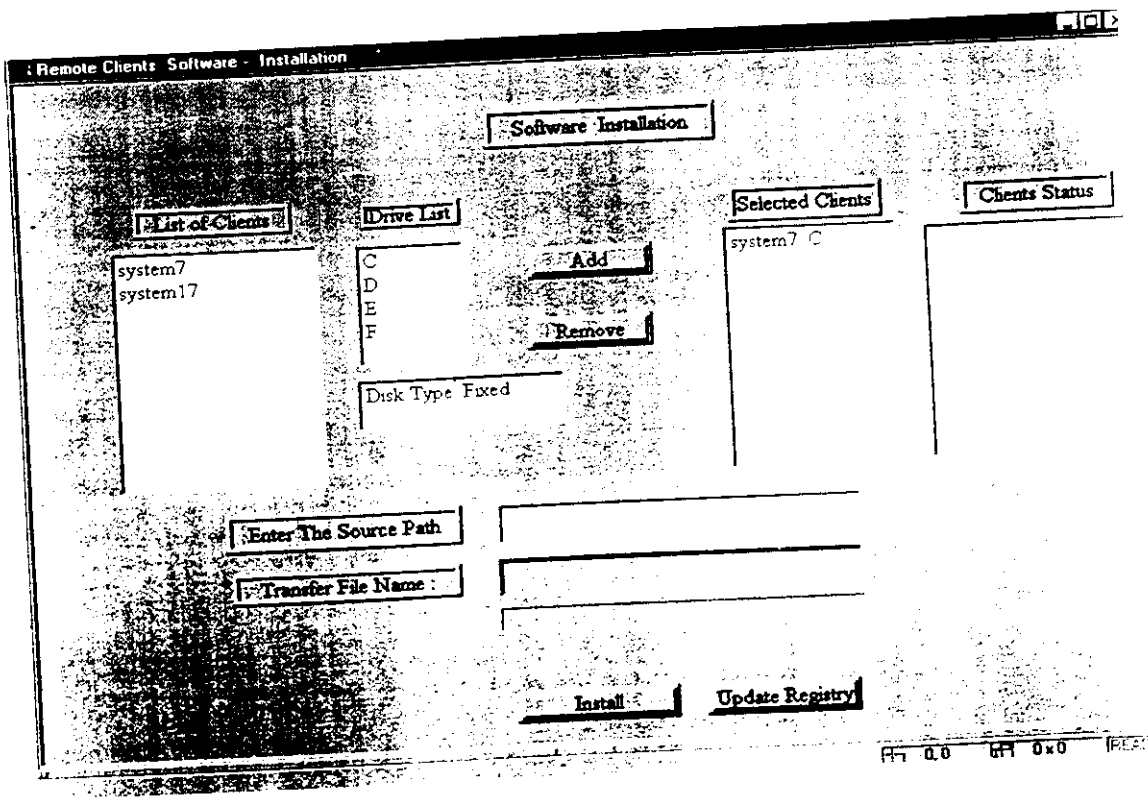


Figure 1.5

The sockets corresponding to the selected clients are identified and the any further communications are done via these sockets simultaneously.

The user starts installing the required software on the server. Then, the user provides the *source path name* of the software currently installed as shown in Figure 1.6. On the click of the *Install* button, similar to initial process a new backup of the windows directory is taken in a file.

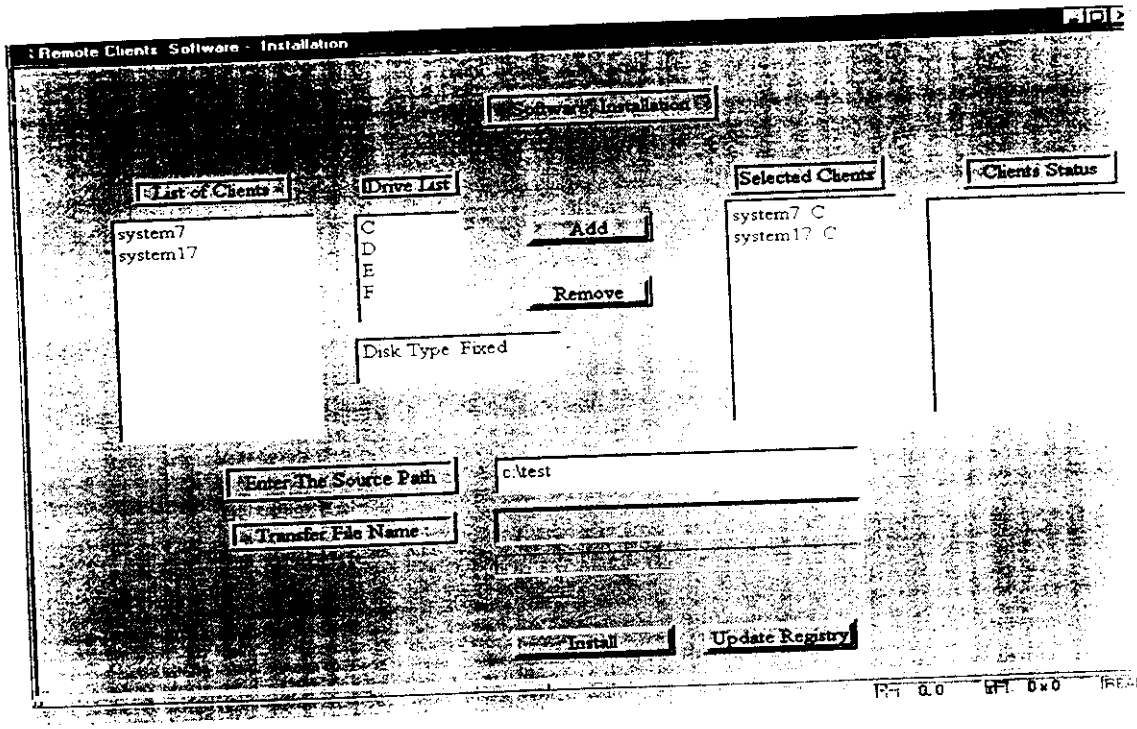


Figure 1.6

Every line in the new file is compared for its occurrence in the old file. A mismatch during comparison indicates the existence of an additional file in the windows directory. This mismatched line along with a control message is sent to the clients' application, which identifies the content of the line to be a file name with the help of the control message. This is followed by the creation of a file in that particular client's windows directory and an acknowledgement is sent to server.

On receiving the acknowledgement, the server starts reading the file contents and transfers it to the clients until end of file is reached. The clients reciprocate by writing the contents into the file created. The comparison is thus proceeded.

Thus, all the newly created files during installation in the windows directory are reflected in the client's windows directory.

The files under the *source path name* specified by the user are also transferred to the clients. On the termination of this process, an indication is given to the user that the file transfer is completed as shown in Figure 1.7.

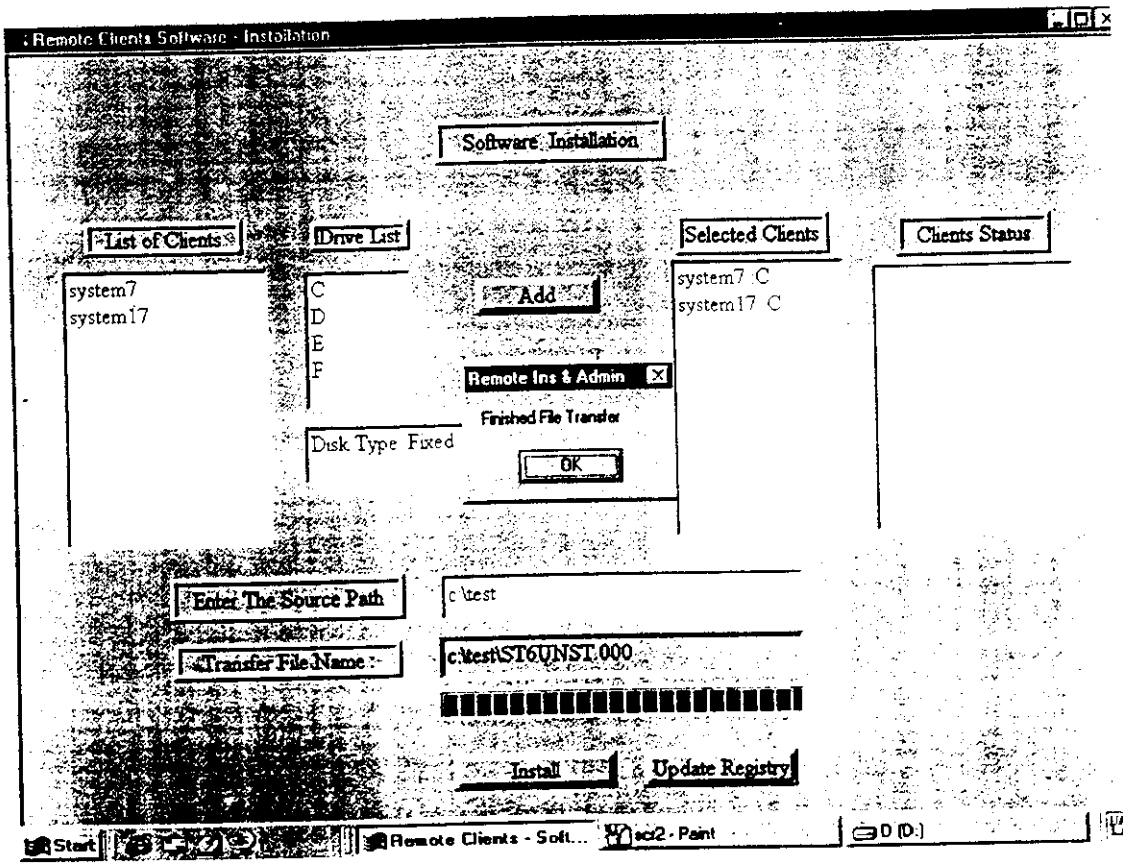


Figure 1.7

The user then clicks the *update registry* button to initiate the registry updations. The new registry backup is taken. Every line in the new file is compared for its occurrence in the old file.

A mismatch during comparison indicates the existence of an additional key in the registry. The values and data of that key are identified, the value is then checked for its type namely *string, DWORD, binary*. All the above details along with the key are sent to the clients.

The clients on receiving the details perform the updations accordingly in the registry. Once the updations terminates, an acknowledgement is sent to the server. The server now displays the status of installation of each and every client selected as shown in Figure 1.8.

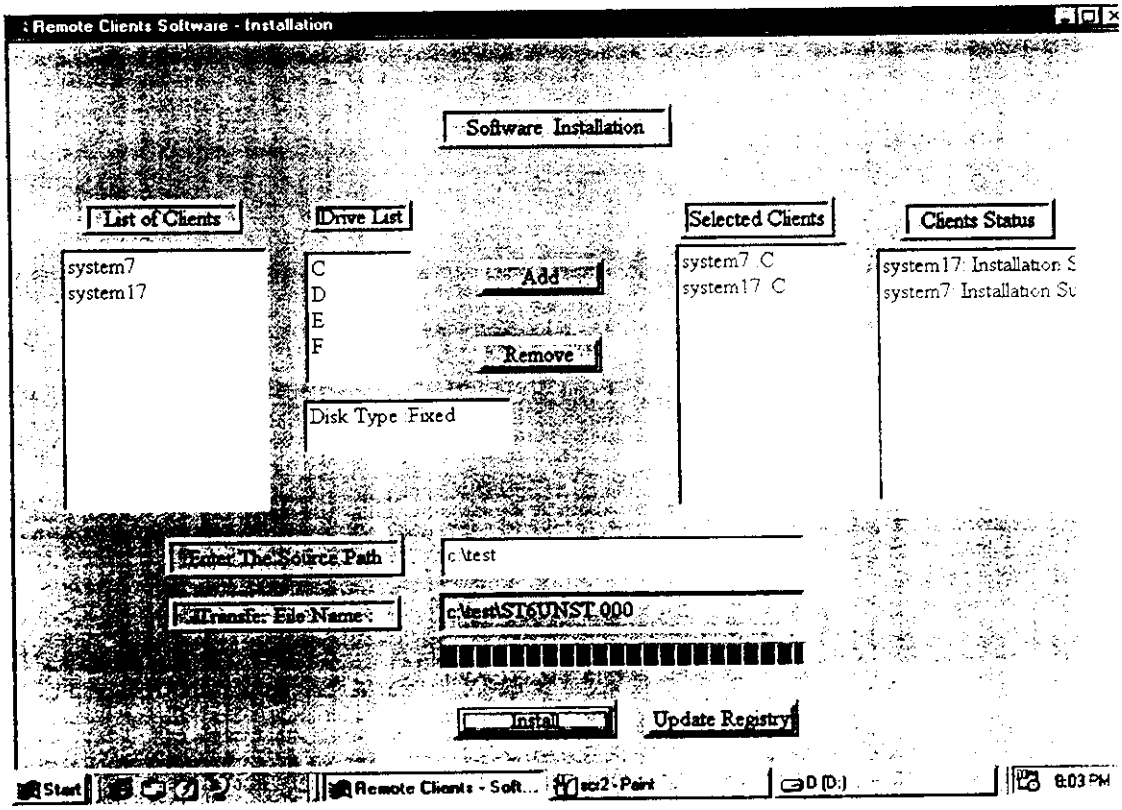


Figure 1.8

4.2.2. Network Administration

As and when the client application gets connected to the server application, the client sends its system name and the server displays the system name in the list box. The user can select as many clients from the list box. Various administrative operations are provided in the form of radio buttons for the user to efficiently select one operation at a time. The options include Shutdown, Logoff, Message passing and Restart.

Once the option is selected, the *Perform* button is clicked. The server then identifies the sockets corresponding to the selected clients and sends an appropriate message, which uniquely defines the selected operation via the sockets. On receiving the message, the client performs the needed task. In case of message passing, the user is provided with a text box, wherein the message to be sent to the client is typed as shown in Figure 1.9. On the click of the perform button, the message is sent to the selected clients via the sockets. The clients on receiving the message display the same for the user.

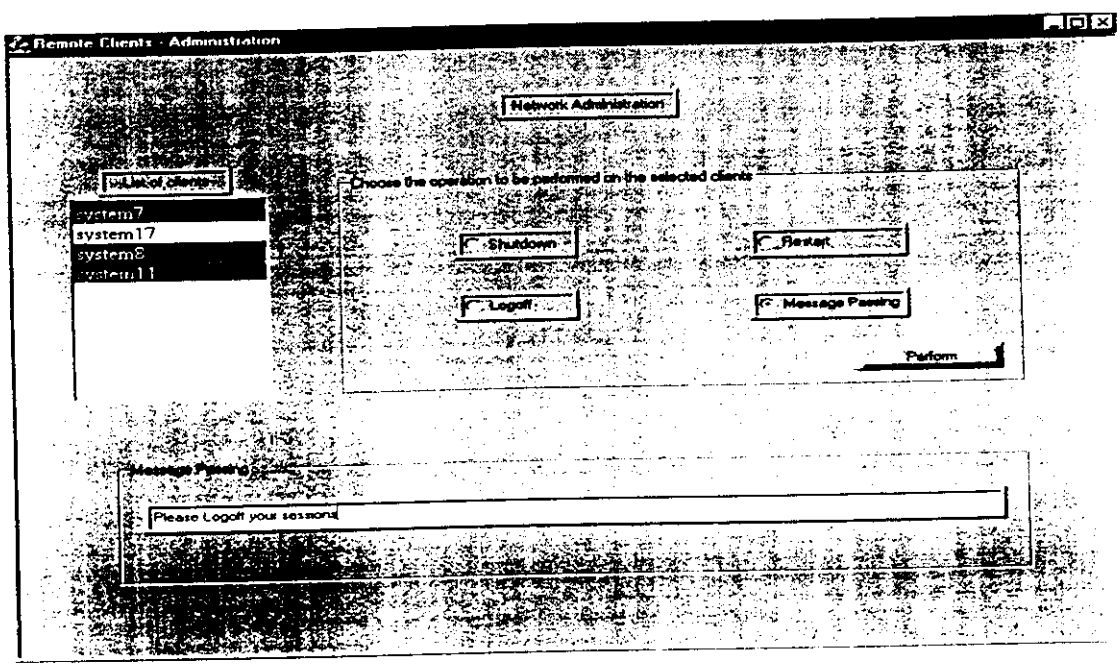
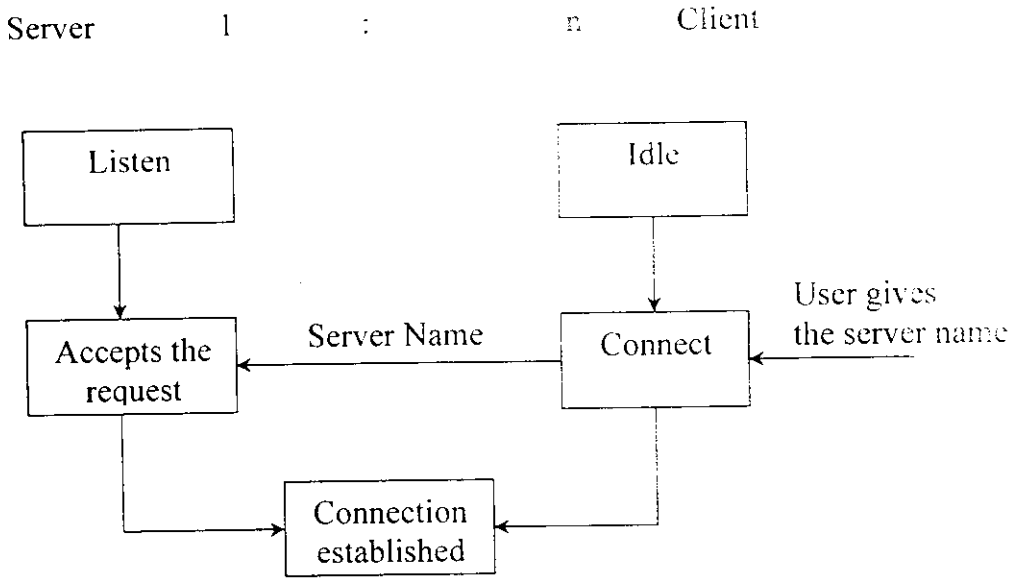


Figure 1.9

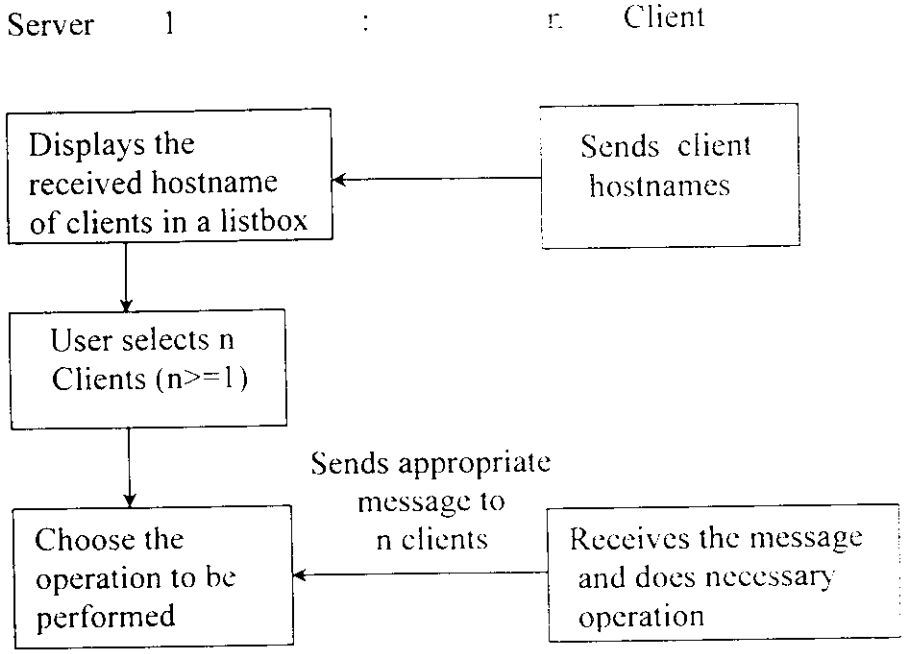
4.3. FLOW DIAGRAM

Connection establishment,



After establishing connection,

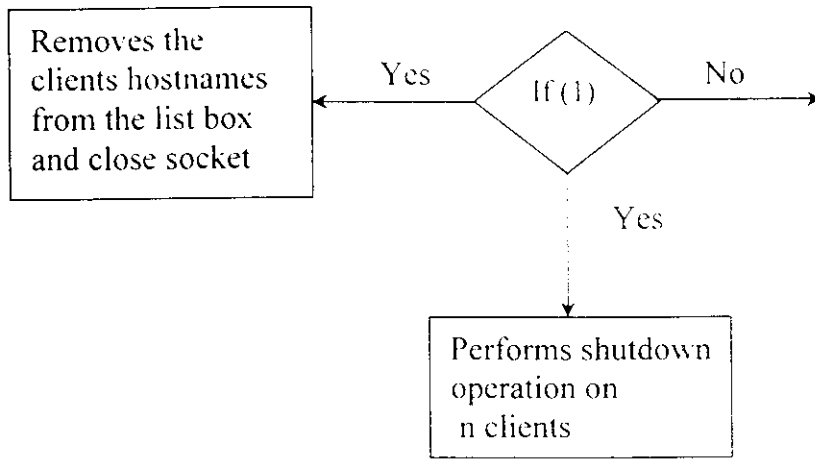
Administration



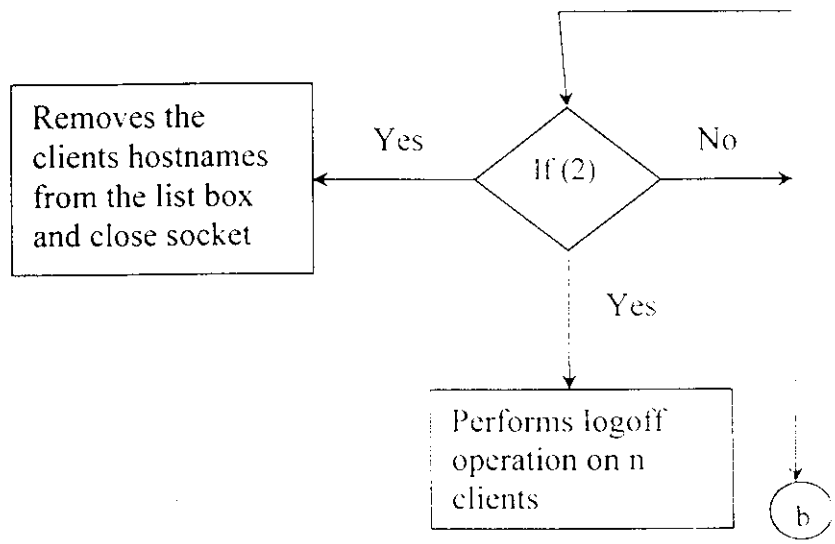
After receiving appropriate message from server,

Server 1 : n Client

(1)



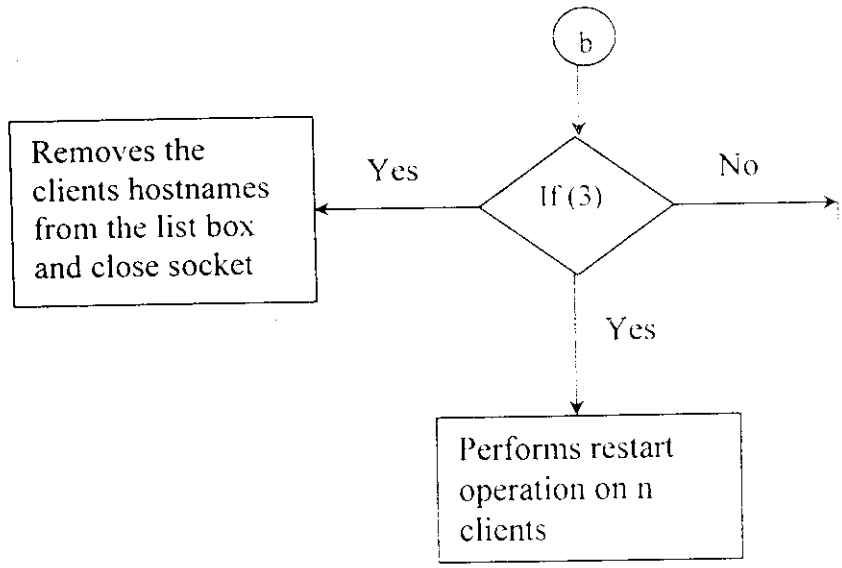
(2)



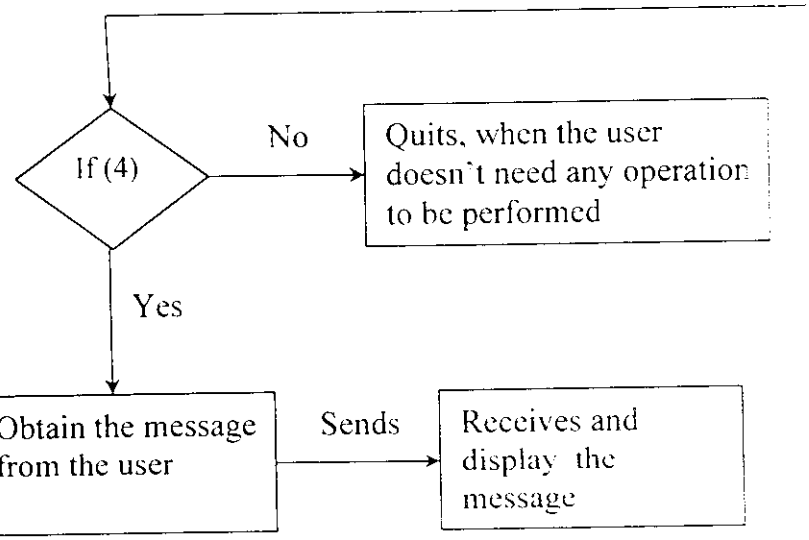
(1) - *ShutDown* (2) - *Logoff*

Server 1 : n Client

(3)



(4)

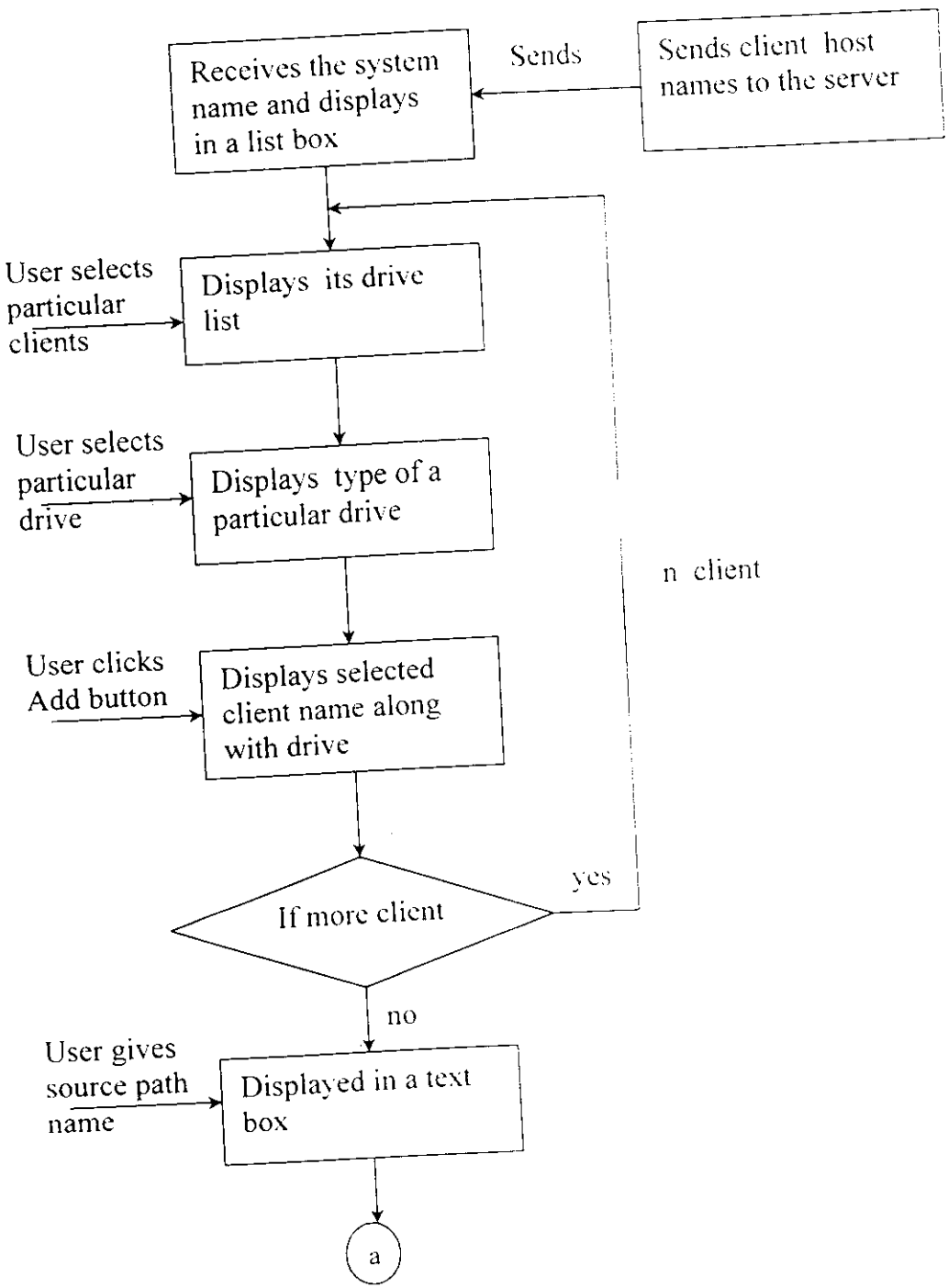


(3) - *Restart* (4) - *Message Passing*

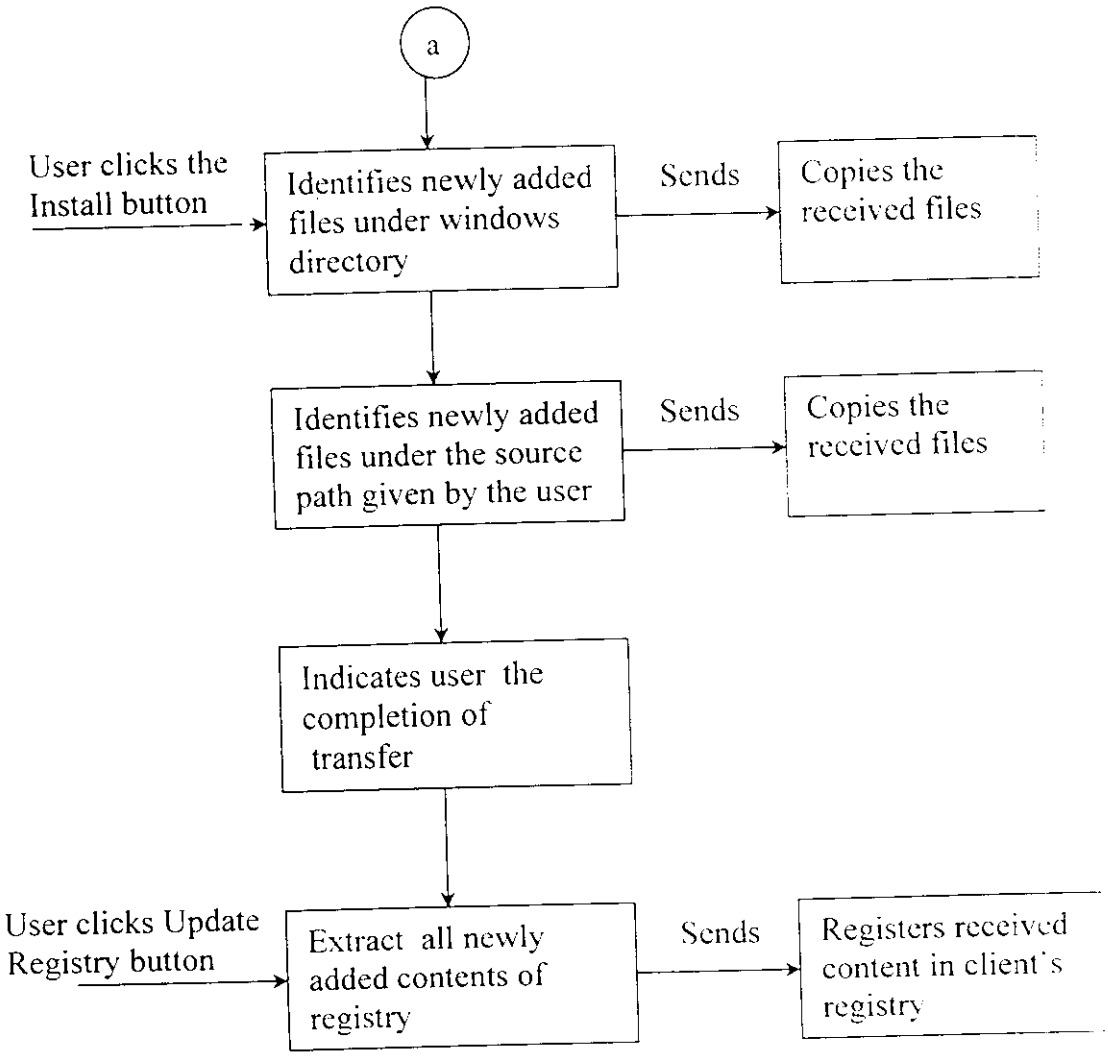
Installation

After establishing connection with the server.

Server 1 : n Client



Server 1 : n Client



5. TESTING

.1. Introduction

Testing is a mandatory and continuous process for the successful ompletion of any project Testing is done at various levels. Integration testing ollows Unit Testing. Here, different dependant units are assembled and tested for ny bugs that may surface due to the integration of the modules.

A final testing is done to access the correctness of the whole software as such. Here several test cases are taken and real life situations are created while testing. The program timings are also noted and compared with present or express results.

5.2. Unit Testing

In the installation module, during file transfer from the server to the client, data was getting transferred in a much slower rate. This was rectified and corrected by increasing the byte transfer rate.

In the registry access module, a VC++ command was used to read its contents into a file. On testing, it was observed that the file contents were in ASCII format. Since the identification of keys, values and data from ASCII format file was tedious many VC++ commands were supplemented for reading the above details more logically.

During updation of the registry at the client end the updations could not be performed on the client's machine due to some mismatching. It was

as observed that the registry contents were a combination of three types String, WORD and Binary. The necessary conversion was thus performed in the server self before passing it on to the client.

When transferring the windows directory contents from the server to ne client, "Unknown path error message" occurred. It was found that by default ne windows directory were updated at the client side assuming the same path as nat of the server system. Since it is not necessary that the client systems should ave the windows directory in the same path as that of the server, its source was etected and then updating was done.

Initially during file transfer, the process of transfer wasn't knowledgeable as the screen appeared stand still since larger files naturally took a span of moment to transfer. Hence as an indication of file transfer a progress bar was included.

5.3. Integration Testing:

While integrating Installation module and Administration module many errors were listed due to the usage of individual sockets for each of the modules. Hence a common socket programming was written by introducing a home page.

When clients got randomly connected and disconnected into any one of the above module, the incoming connections weren't detected properly. This was because both the modules were implemented using modal dialog, which expects a response for any event that occurs. Hence they were converted into modeless dialogs.

The Regression testing, the last step in the Integration Testing was carried out to find out various complications that aroused in the individual modules.

5.4. Control Structure Testing

Condition Testing

The Condition testing is a test case design method that exercises the logical conditions contained in a program module. The various condition testing strategies carried out were as follows.

- **Branch Testing**

  For all compound condition C, the true and false branches of C and every simple condition in C were executed at least once.

- **Branch and Relational operator Testing**

  This technique was performed to guarantee the detection of branch and relational operator errors in a condition.

- **Loop Testing**

  All simple loops, nested loops, concatenated loops and unstructured loops were tested, bugs were detected and corrected.

CONCLUSION

The project "Remote Installation and Administration" can be used where there are large numbers of clients and different softwares are to be installed often. This project could be used to monitor the operation of clients and can be logged off their session by the administrator when they go beyond their restrictions.

Having utilized VC++ to develop this project, the profundity with which most of the operations were supported gave clearer insight into the tool. VC++ supported many components in flexible handling of registry, which was a part in revealing the successful implementation of the project.

As when compared to the existing system of Software Installation over multiple clients in a network, the developed software was found to be much faster and the results were accurate and precise.

This endeavor has triggered my ideas to perform even more upgrades in future.

FUTURE SCOPE

The future holds a lot to offer to the development and refinement of his project. The software is designed in such a way that it will be helpful for any further enhancement.

The facility of checking for enough disk spaces at the client end during installation could be made possible in future.

BIBLIOGRAPHY

1. **Microsoft Corporation.** "Microsoft Developer Network Developer Library", July 2000.
2. **Greg Perry.** "Teach Yourself Object-Oriented Programming with Visual C++", 1995
3. **Steven Holznes.** " Visual C++ Programming", 1996.
4. **Viktor Toth.** "Visual C++ 6.0 Unleashed".
5. **Sandra Oshorne.** "Windows NT Registry", 1998.
6. **Chris Pappas, William H Murray.** "Complete Reference Visual C++ 5", 1997.
7. **Richard M Jones,** "MFC Programming with Visual C++", 2000.

APPENDIX A-Input Screens

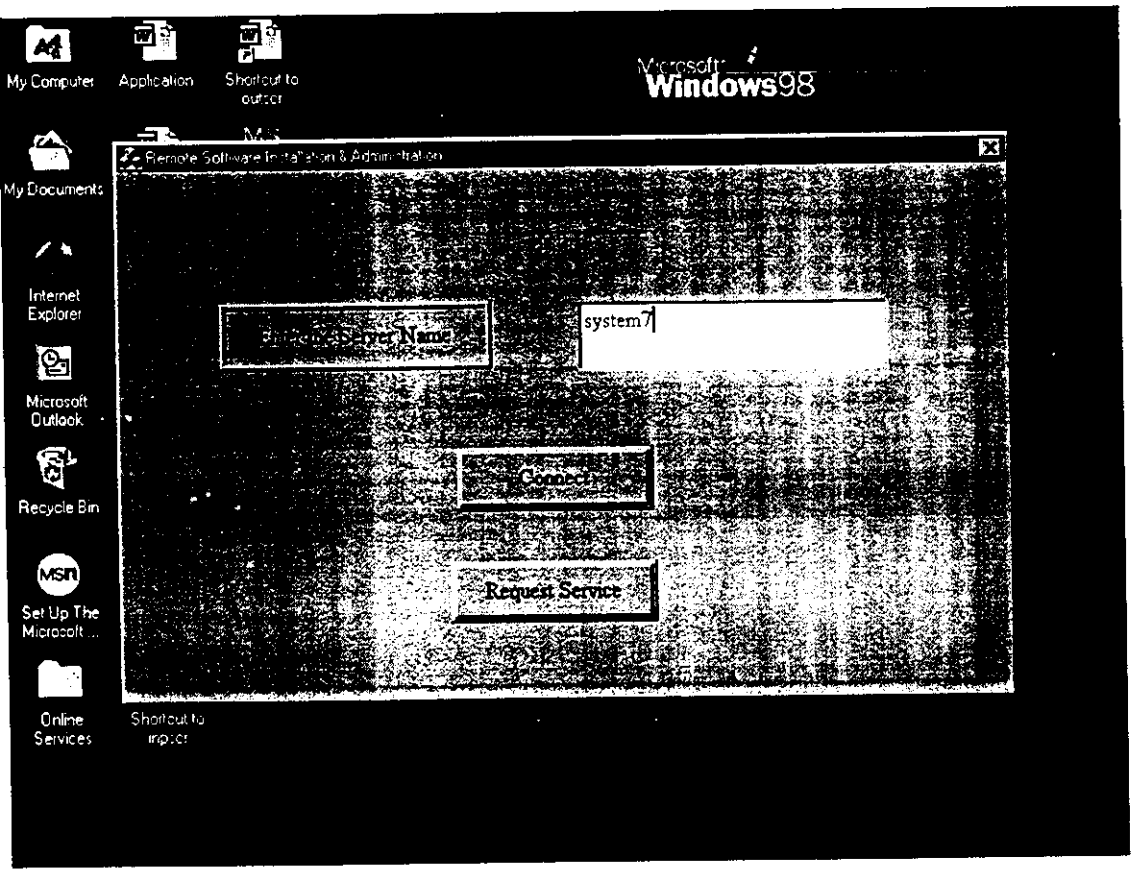


Figure 1-Connection Establishment

Figure1 shows the operation in the client side. The user has to provide the server name to get connected to the server. The server name should be the server's IP Address or the System Identification name.

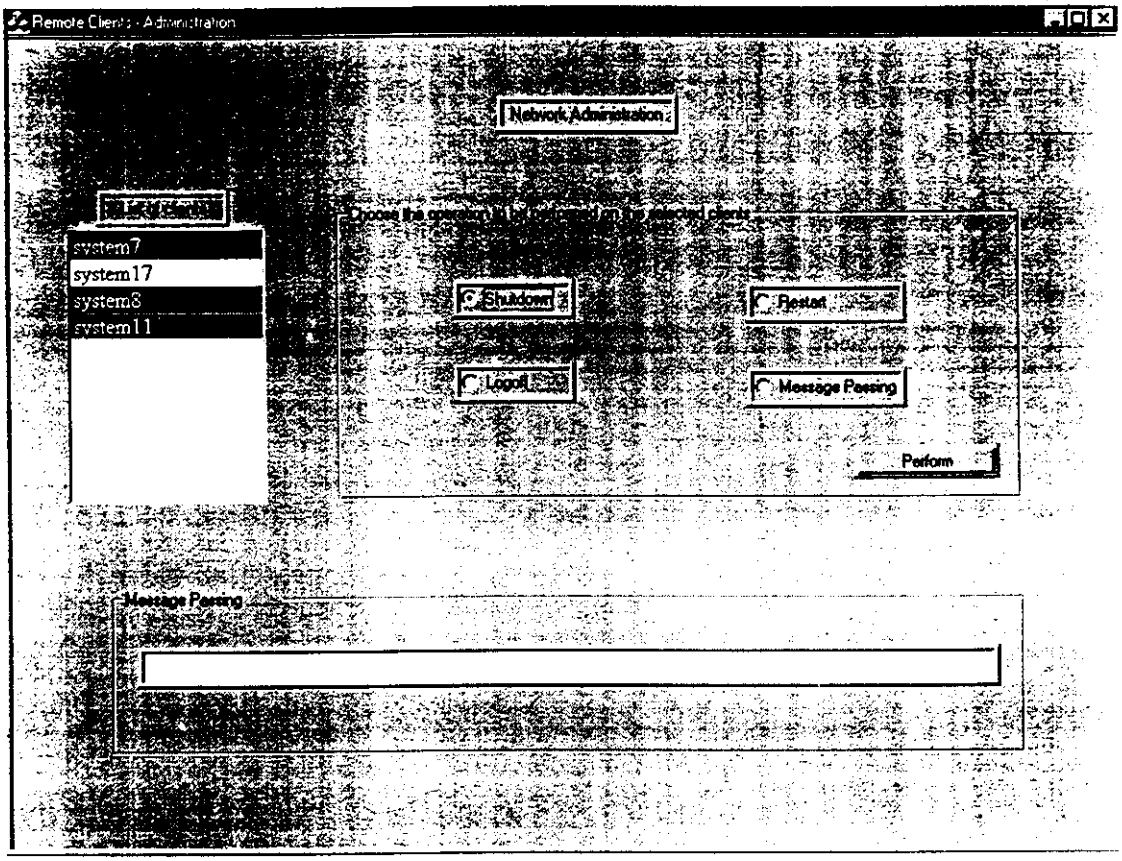


Figure 2-Select client and the necessary operation

Figure 2 depicts the framework of the administration module in the server side. Initially, the user has to select the list of clients from the list box. Then, he has to choose the necessary operation to be performed on the selected clients. Message passing can also be done to the needed clients.

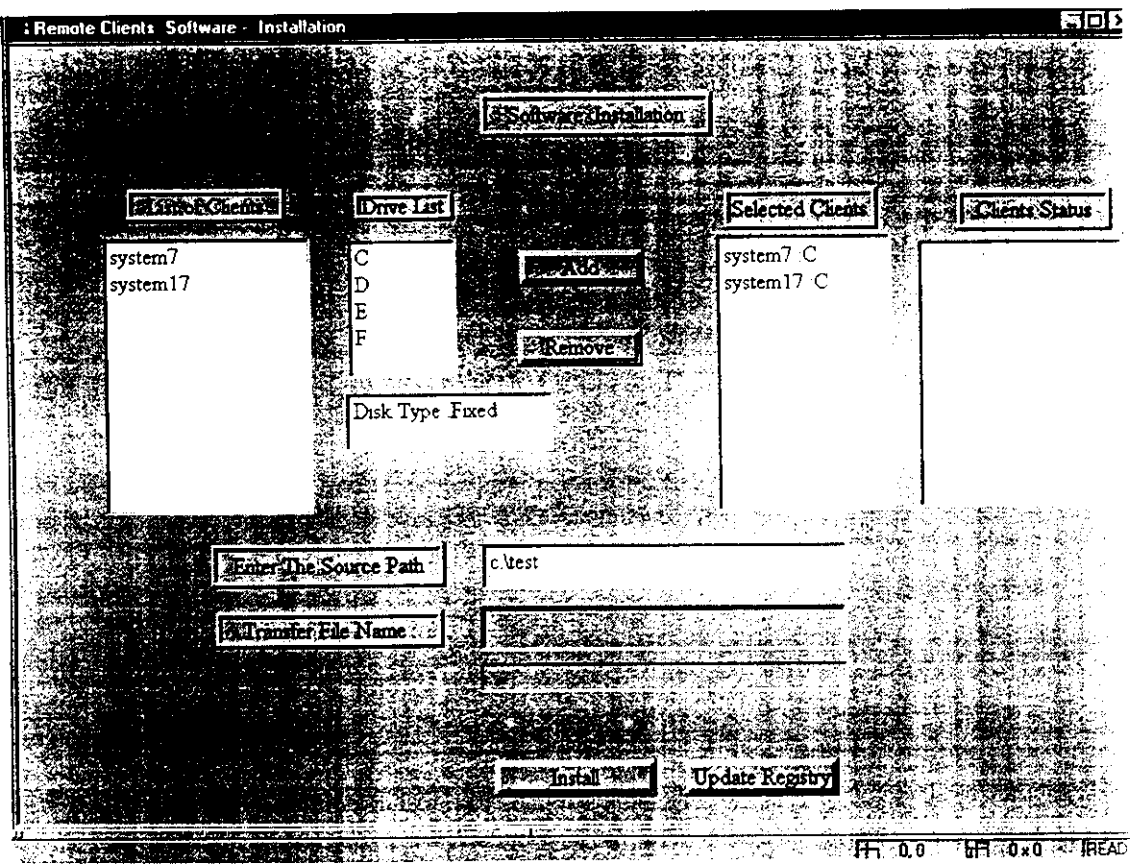


Figure 3-Key-in Source path

Figure 3 reveals the server side Installation module. The user selects the client name from the list box and its corresponding drive. The client name along with its drive gets added to the list box. The user has to provide the source path where the software has to be installed and then clicks the Install button for the commencement of the installation process at the client side.

APPENDIX-B Output Screens

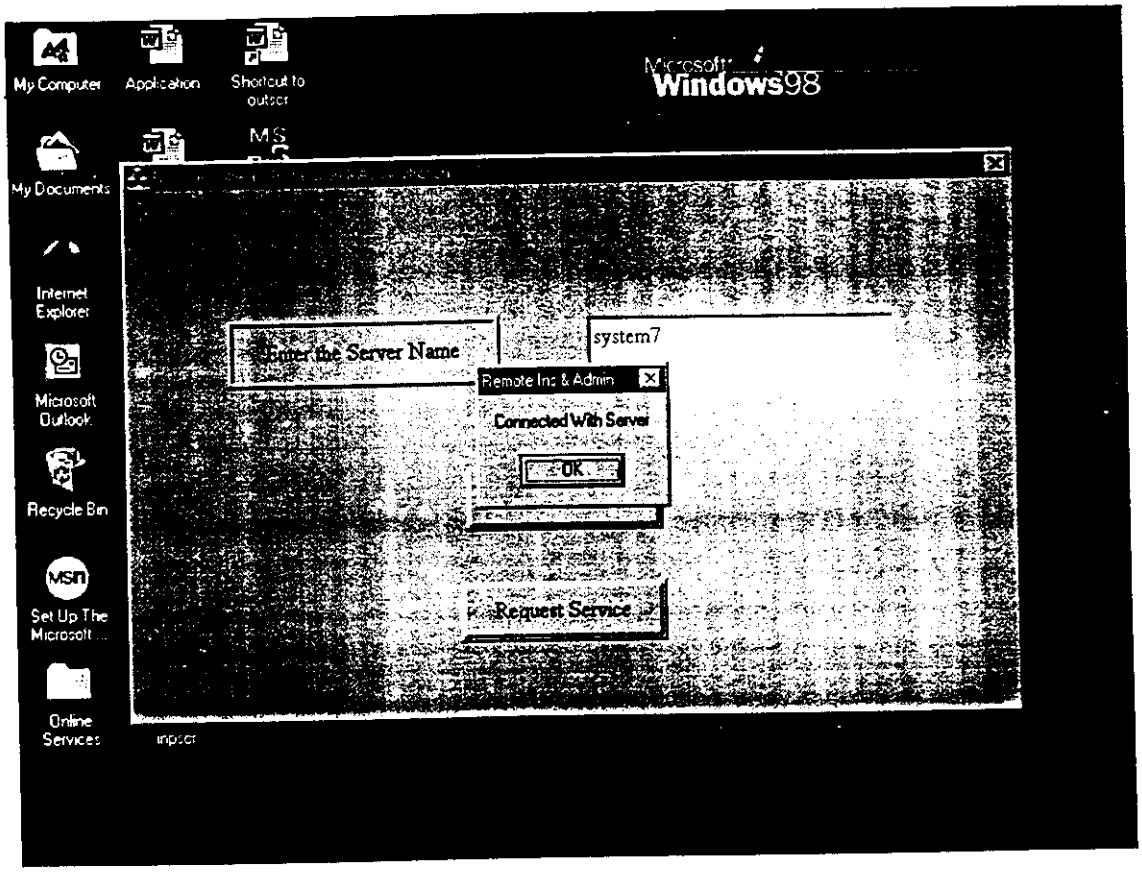


Figure 4-Connection accomplishment report

In the client side, on the click of the "Connect" button, the client gets connected to the server. Figure 4 reveals the intimation in a message box about the connection establishment between client and the server.

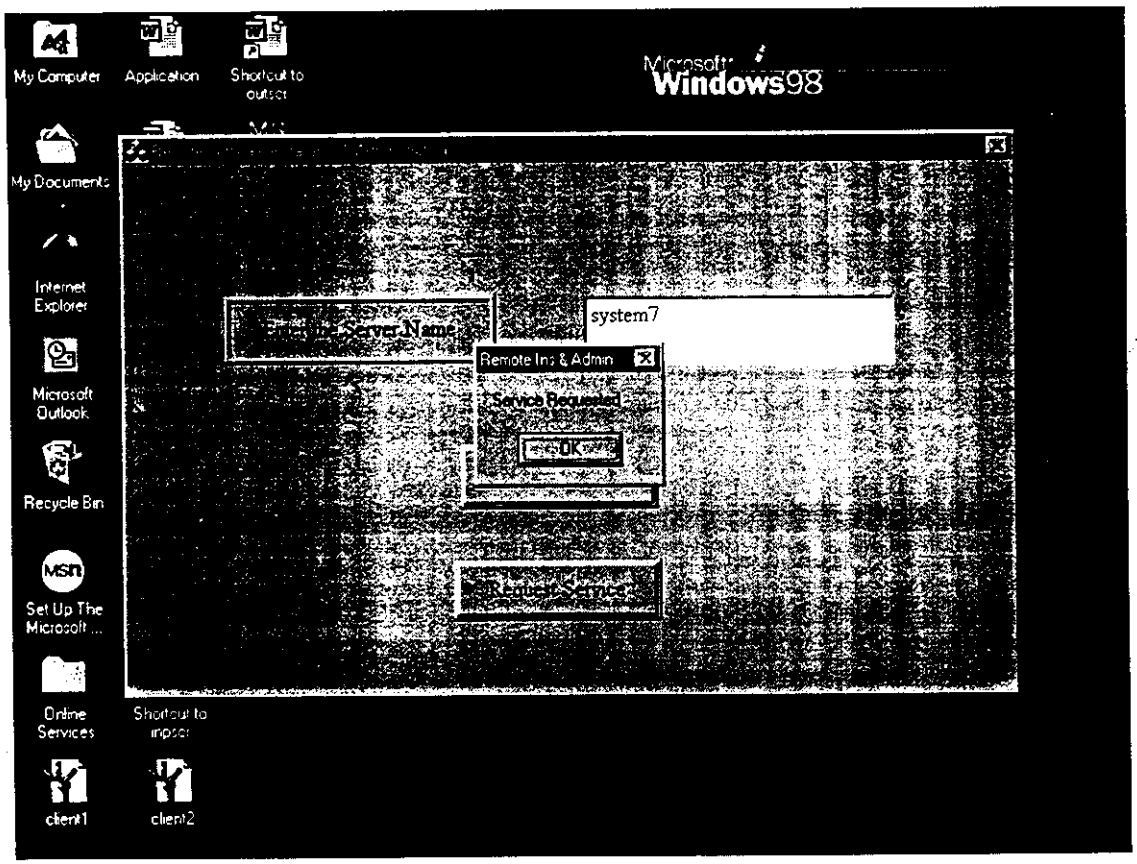


Figure 5-Service Accomplishment report

Figure 5 represents the indication given to the client on successful transfer of the client hostname and its drive names to the server, which marks the first move to offer the service of the Installation and Administration module to the client.

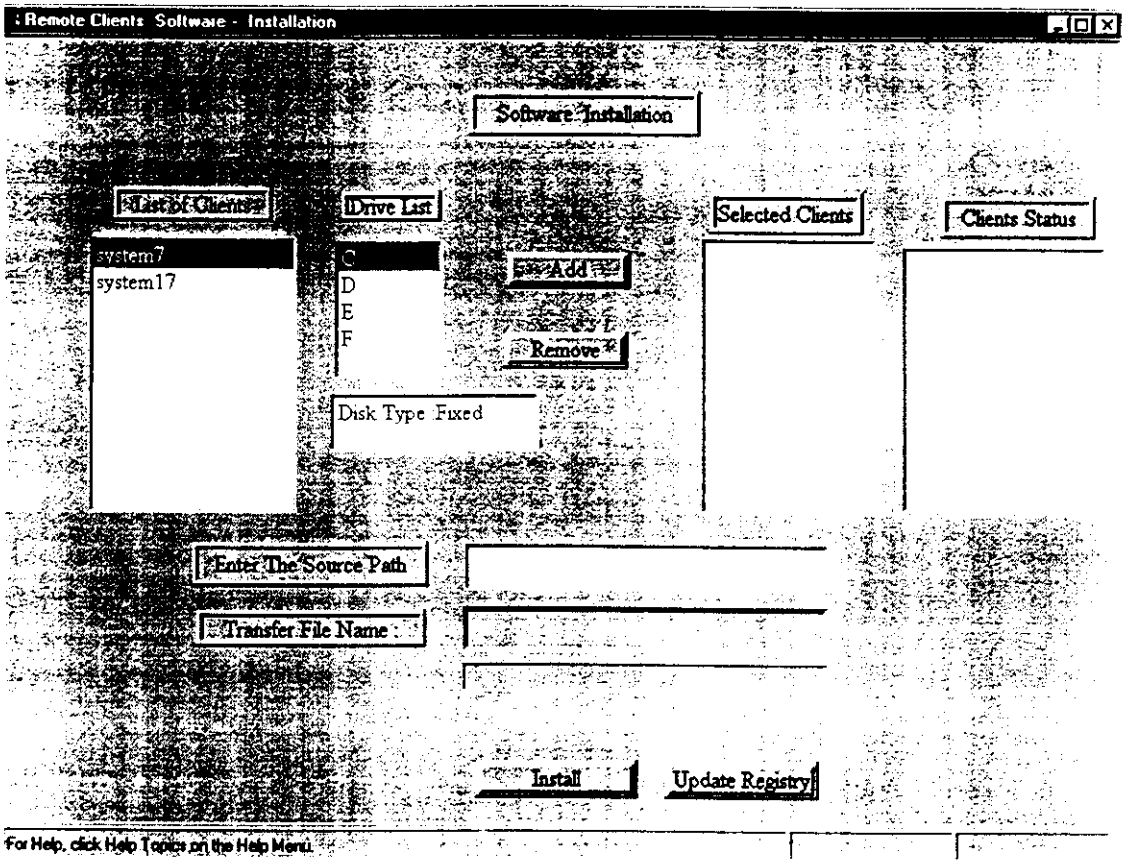


Figure 6-Drive type report

Figure 6 shows the report of Installation module in the server side. On the click of the client name and its respective drive, the details about the drive whether it is a Fixed Drive or a Network Drive is displayed for the reference of the user.

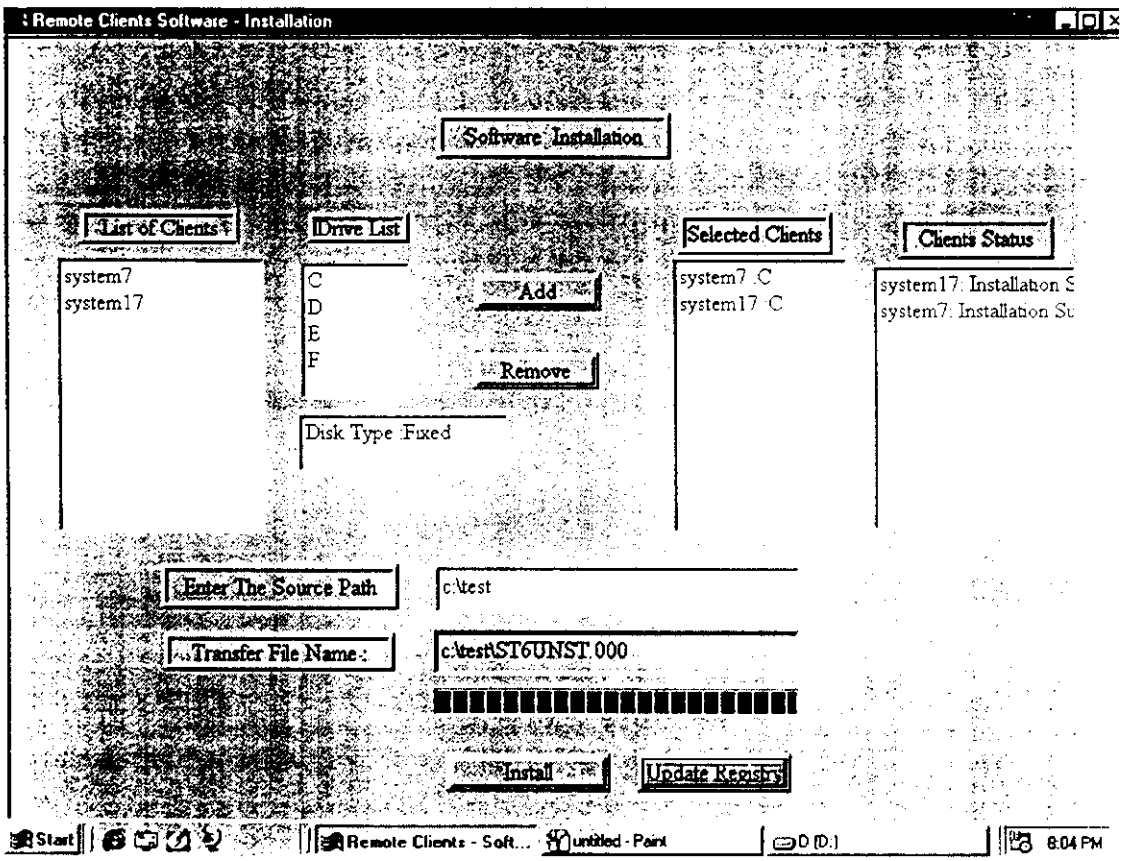


Figure 7-Installation Status report

The report about the status of installation process on all the selected clients is revealed to the user in Figure 7. Also, it marks the updation of the registry as a part of the installation process, after transfer of necessary files to the clients.