



# **Kumaraguru College of Technology**

Department of Computer Science and Engineering  
Coimbatore-641 006



ISO  
9001:2000

P-1097

## **REMOTE CONTROLLING A PC THROUGH LAN**

Project work done at

**SRM SYSTEMS AND SOFTWARE  
CHENNAI.**

### **PROJECT REPORT**

Submitted in partial fulfillment of the  
Requirements for the award of the degree of  
**Master of Science in Applied Science**

**Software Engineering**

Bharathiar University, Coimbatore

Submitted by

**VIKRANTH.P**

**Reg.No-0037S0114**

#### **INTERNAL GUIDE**

**Ms.L.S.Jayashree.M.E**

Dept. of Computer Science & Engineering,  
Kumaraguru College of Technology,  
Coimbatore.

#### **EXTERNAL GUIDE**

**Mr.S.Kamesh B.E,**

**SRM SYSTEMS AND SOFTWARE  
CHENNAI.**



**Department of Computer Science and Engineering**  
**KUMARAGURU COLLEGE OF TECHNOLOGY**

**Coimbatore – 641 006**



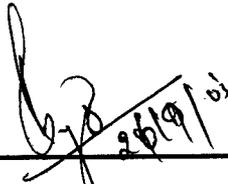
ISO  
9001:200

**CERTIFICATE**

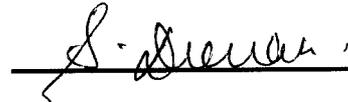
**PROJECT REPORT 2003**

Certified that this is a bonafide report of  
the project work done by

**Vikranth.P.**  
**(Reg. No. 0037S0114)**

  
26/9/03

**Ms.L.S.Jayashree.M.E**  
Project guide  
Computer Science & Engineering



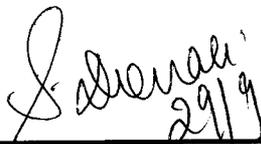
**Prof. S. Thangasamy , Ph.D.,**  
Head of the Department  
Computer Science & Engineering

Place : Coimbatore

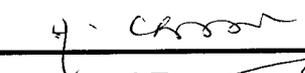
Date : 29.09.03

Submitted for viva-voce examination held on

29.9.03

  
29/9

**Internal Examiner**

  
29/9

**External Examiner**

## **SRM SYSTEMS AND SOFTWARE LIMITED**

24, G.N. Chetty Road, T.Nagar, Chennai - 600 017.  
© : 91 - 44 - 8250771, 8258757, 8269471 Fax : 91 - 44 - 8283359  
E-mail : srm@srmsoft.co.in Web Site : <http://www.srmsoft.com>  
Regd. Off : 2, Veerasamy St., West Mambalam, Chennai - 600 033.



23.09.2003

### **CERTIFICATE**

This is to certify that the project work entitled "**REMOTE CONTROL A PC THROUGH INTERNET**" Analyzed, Designed and Developed by Mr. **VIKRANTH PEERAN** of **KUMARAGURU COLLEGE OF TECHNOLOGY (CBE)**, submitted in partial fulfillment of the requirements of degree of **4<sup>th</sup> Year M.Sc (S.E)** has been carried out in our organization from June 2003 to Sep 2003. This project has been developed using **VC++**.

We wish him success in all his future endeavors.

**For SRM Systems And Software Limited**

A handwritten signature in black ink, appearing to read 'Surya', written over a horizontal line.

**Manager-Projects**

## DECLARATION

I here by declare that the project entitled "REMOTE CONTROLLING A PC THROUGH LAN", submitted to Kumaraguru College of Technology, Coimbatore Affiliated to Bharathiar university as the project work of Master of Science in Applied Science Software Engineering ,is a record of original work done by me under the supervision and guidance of Mr.S.Kamesh.B.E, SRM Systems and Software, Chennai and Ms.LS.Jayashree M.E, CSE Department Kumaraguru College of Technology, Coimbatore and the project work has not found the basis for the award of any Degree/Diploma/Associateship/Fellowship or similar title to any candidate of any University.

Place: Coimbatore

Date: 26.09.03

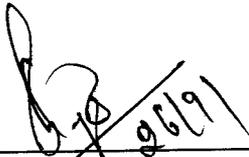


---

(VIKRANTH.P)

Reg.No:0037S0114

Countersigned by



---

(Internal Guide)

Ms. L.S Jayashree M.E.,

Kumaraguru College of Technology,

Coimbatore.

# Acknowledgement

---

## ACKNOWLEDGEMENT

I am immensely grateful to **Dr.K.K.Padmanaban BSc(Engg) , M.Tech., Ph.D.**, Principal , Kumaraguru College of Technology for his valuable support to come out with this project.

I really feel delighted in expressing my heartfelt thanks to **Dr.S.Thangaswamy Ph.D**, Prof & Head of Department of Computer Science and Engineering for his endless encouragement in carrying out this project successfully.

My heartfelt thanks to our project coordinator **Mrs.S.Devaki B.E., M.S**, Assistant Professor, for his unfailing enthusiasm, encouragement and guidance that paved me to the completion of this project.

I am indented to express my heartiest thanks to **Ms.L.S.Jayashree.M.E.** my project guide who rendered his valuable guidance and support to do this project work extremely well.

I am also thankful to all the faculty members of the Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore for their valuable guidance, support and encouragement during the course of my project work..

My humble gratitude and thanks to my parents who have supported, to complete the project and to my friends, for lending me valuable tips, support and cooperation through out my project work.

# Synopsis

---

## SYNOPSIS

Remote accessing has become a watchword in current day computing technology. Having complete access to workstations located in far flung regions is made possible using applications capable of communicating using protocols over LAN based architectures.

Keeping this perspective in mind the project entitled **“Remote controlling a PC through Internet / LAN”** is developed to aid in remote accessing and installation.

Overriding the Client – Server architecture a client socket is installed in the client machine and similarly a server socket is installed in the server workstation. Communication is established by providing the client socket with the IP address of the server machine using the TCP / IP protocol.

Once the communication is established a virtual desktop of the server machine is generated in the client socket using a GUI interface developed in Visual C++.

Now the client socket acts like a gateway to perform all operations that once can perform on the server machine itself. All mouse and keyboard interrupts are captured and imitated thus providing a firsthand experience.

Thus remote accessing is achieved and implemented.

# CONTENTS

Acknowledgement	
Synopsis	
	Page No.
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Project Overview	1
1.2 Profile of the Organization	1
<b>2. RELEVANCE AND IMPORTANCE</b>	<b>5</b>
2.1 Existing System	5
2.2 Need for Proposed System	5
2.3 Proposed System	
<b>3. PROGRAMMING ENVIRONMENT</b>	<b>6</b>
3.1 Hardware Specification	6
3.2 Software Specification	6
3.3 Software Description	6
<b>4. SYSTEM DESIGN</b>	<b>10</b>
4.1 Input Design	10
4.2. Output design	10
4.3 Code Design	11
4.4. System Flow Diagram	11
<b>5. PROJECT DESCRIPTION</b>	<b>14</b>
5.1 Introduction	14
5.2 Module Description	14
<b>6. SYSTEM TESTING</b>	<b>21</b>
6.1 Testing Objectives	21
6.2 Testing Principles	21
6.3 Levels of Testing	22

<b>7. CONCLUSION AND FUTURE OUTLOOK</b>	<b>25</b>
7.1 Conclusion	25
7.2 Limitations	25
7.3 Future Enhancements	26
<b>REFERENCES</b>	<b>27</b>
<b>APPENDICES</b>	<b>28</b>

# Introduction

---

# 1.INTRODUCTION

## 1.1 Project Overview

The project entitled “Remote controlling a PC through LAN” is developed to assist users in controlling a remote PC without having to be in direct physical contact with the computer itself. The application is developed with a view towards minimizing the task of remote accessing and manipulation.

Remote connection is established via TCP/IP protocols by feeding the IP address as input after prompting the application to start the server. This application can be used to connect any two systems over a Local Area Network(LAN).

The system thus reduces manual labour and also facilitates remote accessing and manipulation.

## 1.2. Company Profile

**SRM SYSTEMS AND SOFTWARE** is a company committed to provide support to small, medium and large corporations in the development and management of software essential to their needs over the entire life cycle of a project or system. All corporations, regardless of size, need to process enormous amounts of data in support of the day-to-day operation of the company and the dependence on a corporate information system and up-grade the existing ones. In seeking efficient and cost-effective approaches to manage change, many companies have found outsourcing to be particularly attractive.

SRM Systems and Software is here to provide expert services and support for “change management” in software systems allowing your organization to focus on its core business. SRM Systems and Software offers the expertise of experienced individual software consultants, as well as an offshore facility with a state-of-the-art information technology infrastructure and a well-trained and committed staff, all at extremely competitive prices. We at SRM provide our clients the potential for significant savings without a compromise in quality or schedule. SRM Systems and

Software guarantees that the software services will be delivered to the customer on time, within budget, incomplete conformance means that at SRM, we are indeed “Determined to Make a Difference”.

### **Mission Statement**

The stated mission of the SRM System and Software is to offer value addition to the customer’s Business through IT Solutions of high quality and appropriate Technology on time and on budget.

### **Corporate Background**

- Reputation built over 3 decades
- Global vision
- Asset base of over US \$100 million
- Many interests but one objective - Commitment to

### **Excellence**

SRM Systems and Software is a unit of the renowned SRM Group, which in the past 30 years has established itself in Southern India in the field of Engineering education and Research. Over the years, the SRM Group, with an asset base of more than US \$ 50 million, has expanded into the fields of Health Care, Hospitality, Manufacturing, Financial Services and Construction.

SRM Systems and Software was established with a specific business focus on Software Development and consultancy. As a member of the Software Technology Park of India, SRM Systems and Software benefits through business and customs duty incentives from the Government of India and consequently is committed to export 100% of its products and services.

The overseas office of SRM Systems and Software in Boston provides an effective link to customers in the United States and other parts of the world. Efforts are under way to establish similar offices in Japan, UK, Europe and Australia. Connected by broadband data links, the Headquarter in Chennai and the overseas

offices will be positioned to provide customers global information technology market by an unwavering commitment to quality.

### SRM Systems and Software - A Customer Centric Company

#### **Objectives**

- World Class Products
- Commitment to Quality
- Impeccable Customer Service
- Excellent Technical Support

#### **Business Ethics**

- Customer is God
- Work is Worship
- Employee is Strength
- Humanity is the Base

#### **Strategy**

- Our International strategy is to penetrate and service the market by On-site, Off shore & Turnkey projects based on our expertise and related software solutions
- Our Domestic Strategy in India is to increase market share, expand Client base and focus on large IT contracts.

#### **Unified Strength**

- Three decades of SRM's Track Record
- Strong Team Work
- Excellent Technical Competence
- Structured Project Approach
- Customer Centric and Focus on Customers' Customers
- Japanese Language Competence

## **Services Offered**

SRM Systems and Software through its Strategic Business Units offers the following services.

### **Customized Software Development**

SRM can provide complete business turnkey solutions to small, medium and large size companies spanning every phase of the software life cycle: System Analysis, Design, Implementation, Testing, Installation and Maintenance. The SRM staff has an accumulated experience of more than 300 man-years in varied application areas. SRM offers software services in the following technology areas:

- Web Based Applications and e-commerce
- Client-Server (two-three and n-tier Technology)
- Group Ware and Workflow
- Multimedia and Computer Graphics
- Computer Aided Design and Computer Aided Manufacturing

SRM guarantees each customer that any project executed by SRM will be developed as per the specifications, delivered on time, and without cost overrun. SRM strictly adheres to the latest Software Engineering standards in the development of customized software. The aim of SRM is to win the allegiance of each customer so that the relationship does not end with the completion of the first contract but becomes an ongoing and mutually beneficial association.

## **2. Relevance and Importance**

### **2.1 Existing System**

The existing system is a manual process of checking client activity. The drawbacks of the system are

- Time consuming.
- Requires more manual labor.
- Remote accessing and installation is not possible.
- Administrator lacks complete control.

### **2.2 Need for Proposed System**

Owing to the abovementioned drawbacks in the existing system, an automated system is proposed. The proposed system aims to eliminate these drawbacks. It can be viewed as user friendly, efficient easing the work of the administrator.

The benefits of the proposed system are :

- Less time consuming.
- Client monitoring is made possible.
- Remote accessing and installation is made possible.

### **2.3 Proposed System:**

The proposed system is aimed at providing remote access and installation and eases the task of administrator. The proposed system is user friendly.

The proposed system has been developed using VC++ overriding its foundation classes. Since it makes use of TCP/IP protocol the proposed system will overcome the drawbacks of the existing system.

The advantages of the proposed system are

- Remote access and installation.
- Complete control over remote client.
- Capable of tracking client activities.
- Reduction in manual labor.

---

## 3. PROGRAMMING ENVIRONMENT

### 3.1 Hardware Specification

#### Client/Server:

System	Pentium III @600MHZ.
RAM	128 MB.
Hard Disk	20 GB.
NIC	LAN chord.

### 3.2 Software Specification

Operating system: Windows 2000 workstation.

Software required: Visual C++.

### 3.3 Software Description

Software used to develop the system is Visual C++.

#### 3.3.1. Visual c++:

Visual C++ has various features for which it is selected. It has very good compiling tools. Some of the features of Visual C++ are

- Supports network communication programs
- Supports ActiveX, ODBC, OLE
- Easy to handle graphics and animation
- Easy to write threading applications

Other features of Visual C++ are given below.

### 3.3.2 The visual c++ environment:

An IDE, or Integrated Development Environment, is a program that hosts the compiler, debugger, and application building tools. The central part of the Visual C++ package is Developer Studio, the Integrated Developer Environment (IDE) developer Studio is used to integrate the development tools and the Visual C++ compiler. You can create a windows program, scan through an impressive amount of online help, and debug the program without leaving Developer Studio.

Visual C++ and Developer Studio makes up a fully integrated environment which makes it easier to create Windows programs. By using tools and wizards provided as a part of Developer Studio, along with the MFC class library you can create a program in just a few minutes.

The programs use thousands of lines of source code that are part of MFC class library, They also take advantage of AppWizard and Class Wizard, two of the Developer Studio tools that manage the project.

### 3.3.3 Developer studio wizards:

A Wizard is a tool that helps guide you through a series of steps. In addition to tools that are used for debugging, editing, and creating resources, Developer Studio includes several wizards that are used to simplify developing your windows programs. The most commonly used ones are

- App Wizard (also referred to in some screens as MFC AppWizard) is used to create the basic outline of a windows program. Three types of programs are supported by AppWizard: Single Document and Multiple Document applications based on the Document/View architecture and dialog box-based programs, in which a dialogue box serves as the application's main window.
- Class Wizard is used to define the classes in a program created with AppWizard. Using Class Wizard, you can add classes to your project. You can also add functions that control how messages received by each class are handled. Class Wizard also helps manage controls that are contained in dialog

boxes by enabling you to associate an MFC object or class member variable with each control.

### **3.3.4 MFC library:**

A library is a collection of source code or compiled code that you can reuse in your programs. Libraries are available from compile vendors such as Microsoft, as well as from third parties.

Visual C++ 6 includes Version 6.0 of MFC, the Microsoft Foundation Classes, a class library that makes programming for Windows much easier. By using the MFC classes when writing your program for Windows, you can take advantage of a large amount of source code that has been written for you, This enables you to concentrate on the important parts of your code rather than worry about the details of Windows Programming.

A recent addition to the C++ standard is the standard C++ Library. This library includes a set of classes that were known as Standard Template Library, which is primarily used for Windows Programming, the standard C++ library is used for general-purpose programming.

### **3.3.5. The visual c++ editor:**

Developer Studio includes a sophisticated editor as one of its tools. The editor is integrated with other parts of Developer Studio; files are edited in a Developer Studio; files are edited in a Developer Studio child window. You can use the Developer Studio editor to edit C++ source file that will be compiled into Windows Programs. The editor supplied with Developer Studio is similar to a word processor, but instead of fancy text-formatting features. It has features that make it easy to write source code. You can use almost any editor to write C++ source code, but there are several reasons to consider using the editor integrated with Developer Studio, The editor includes many features that are found in specialized programming editors.

- Automatic syntax highlighting colors keywords, comments, and other source code in different colors.
- Automatic “smart” indenting help lines up your code into easy-to-read Columns
- Emulation for keystrokes used by other editors helps if you are familiar with editors such as Brief and Epsilon.
- Integrated keyword help enables you to get help on any keyword, MFC class, or Windows function just by pressing F1.
- Drag-and-drop editing enables you to move text easily by dragging it with the mouse.

## 4. SYSTEM DESIGN

The system design phase follows system analysis .It provides the way information is to be fed and how the output is to be obtained. The design goes through logical and physical stages of development. Logical Design of the physical System prepares input and output specification .The physical Design maps out the details of the physical system plan the system implementation.

### 4.1 Input Design:

The input design has to be carried out for two applications. They are

- Client socket application.
- Server socket application.

#### 4.1.2 Client socket application:

In the client socket application the inputs for the application are Input from the client to start the application.The IP address must also be provided as input

As soon as the client provides the input to start the application the connection is established by the server as well as the capture of the desktop.

#### 4.1.3. Server socket application

The input for the server socket application is the screen sent by the client socket application via the network.

Input to start the application to the server socket

### 4.2. Output design

The application is successful hence it produces an efficient and effective output. The output from the computer is required to show the results of the processing to user. This system consists of interactive outputs for two applications.

- Client socket application.
- Server socket application.

#### **4.2.1. Client socket application:**

The output that is obtained in the client socket application is the virtual desktop of the machine whose IP address is mentioned in the client socket.

#### **4.2.2. Server socket application.**

The output that is generated is a dialog box showing the status of the connection

### **4.3 Code Design.**

In this design an object physical characteristics or performance characteristics or operational instructions are specified. This can also show the inter relationship and may sometimes be used to achieve secrecy or confidentiality.

The development methodology is used in the code design. The approach used here is the top-down approach. Here codes are used for capturing, sending files and signals across networks.

### **4.4. System Flow Diagram**

The system flow diagram for the project is shown in the Figure. The process starts with the administrator giving IP addresses of the client system. The connection is established between the server and the client communicator in the communication module.

Once the connection has been established the timer in the transfer application is activated and the screen is being captured from the monitor by the capture module, This screen is stored in a file. The file stored on the disk is sent to the receiving application for display.

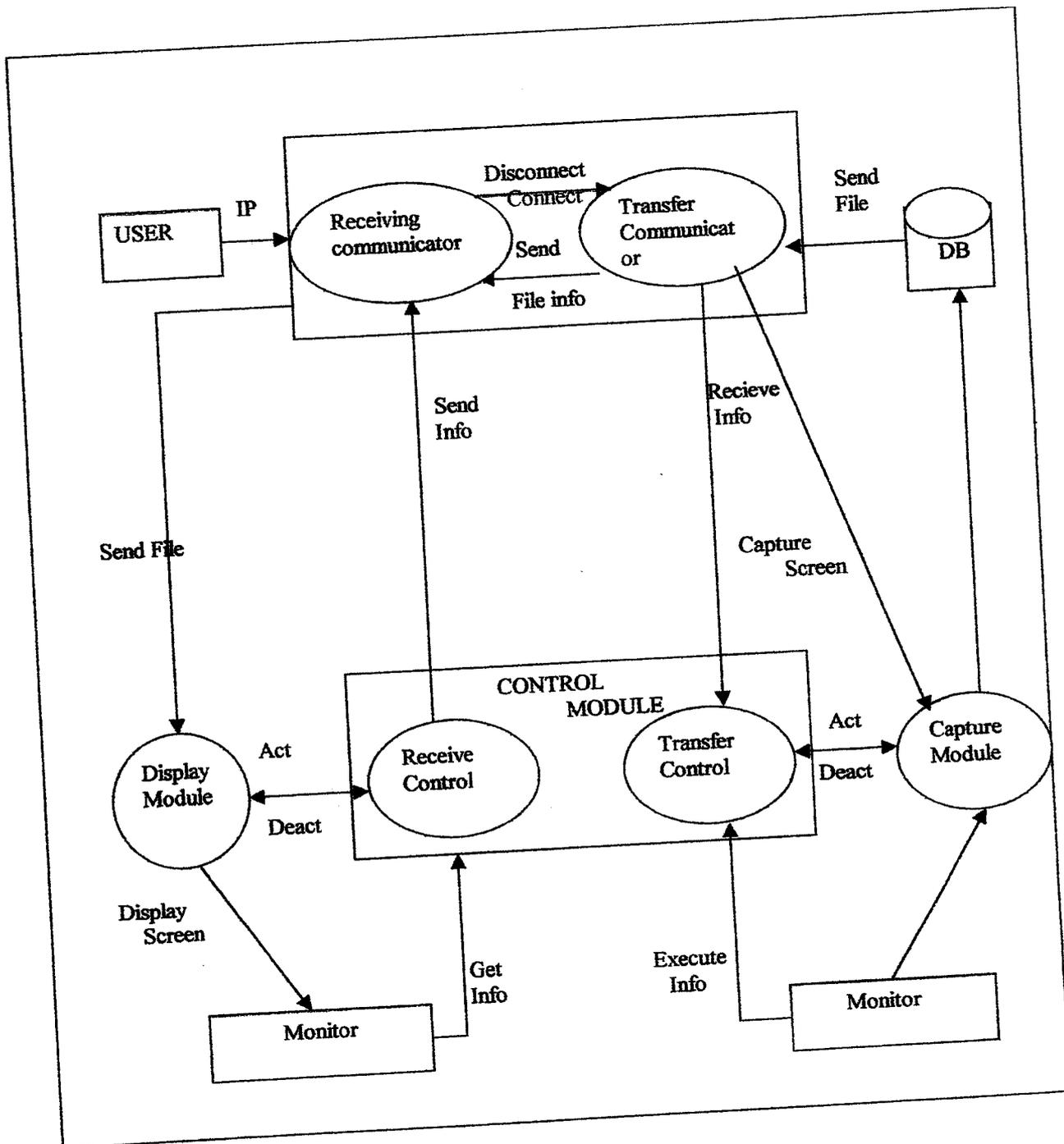
The file is sent to the receiving application via the communication module over the LAN. This file is then sent to the display module ",here the file is displayed in the server application.

When the mouse is moved in the receiving application the mouse information

such as the position and the button clicked is sent to the transfer application via the communication module. This information on reaching the transfer application the mouse information of the mouse is executed by the control module of the transfer application.

This module is also responsible for displaying the messages on the client screen. Thus the control module takes care of the operation of controlling the client system from the server system.

# SYSTEM FLOW DIAGRAM



## **5. PROJECT DESCRIPTION**

### **5.1 Introduction**

The project basically involves the monitoring of the client machine by the server. This system consists of interactive outputs for two applications.

- Client socket application.
- Server socket application.

#### **5.1.2. Client socket application:**

The output that is obtained in the client socket application is the virtual desktop of the machine whose IP address is mentioned in the client socket.

#### **5.1.3. Server socket application**

The output that is generated is a dialog box showing the status of the connection.

### **5.2 Module Description**

The project consists of four modules

- Display Module
- Capture Module
- Communication Module
- Control Module

#### **5.2.1 Display Module**

In general the display module is used in the receiving application. The sole purpose of the display module is to display the client screen sent by the client program on the server screen.

The display module involves the following steps.

- Open the bitmap file
- Creating a logical palette
- Selecting the bitmap image into device context

### **1. Open the bitmap file**

The bitmap file that is transferred from the transfer application to the receiving application is to be loaded into the screen. For this first the bitmap file is to be opened. A buffer is created with the size of the file and the bitmap information is loaded into the buffer from the bitmap file. After which the bitmap information apart from the header file information is read and then the next step is performed. The information read is read from the buffer.

### **2. Creating a logical palette**

Depending on the bit count of the bitmap the logical palette is created. The logical palette is created for the bitmaps which has the colour less than or equal to 256 colours. A logical palette is not created for the colours above 256 colours.

Generally a palette means an array of colors. A logical palette is like a buffer between color intensive applications and the system, allowing an application to use as many colors as needed without interfering with its own displayed colors or with colors used by other windows. After the creation of a palette "Realize Palette()" function is used to put the bitmap image colors onto the screen.

When a window has the input focus and calls "Realize Palette()" function, windows ensures that the window will display all the requested colors, upto the maximum no simultaneously available on the screen. Windows also displays colors not found in the windows palette by matching them to available colors.

### **3. Selecting the bitmap image into the device context**

A device context is a windows data structure containing information about the drawing attributes of the device such as a display or a printer. A device context has to

be created in order to select the bitmap image into the server's screen. The bits of the bitmap have to be sent to the screen for display. To perform this operation "Set DIBitsToDevice()" function is used, This function sets the bits of the bitmap to the device context specified.

### **5.2.2 Capture Module**

This module is responsible for copying the screen from the client screen after which the image is converted into a bitmap image. To capture the screen the following steps are followed

- Creating of Device Context
- Getting the image from the Device Context
- Converting the Device Dependent Bitmap(DBB) into Device Independent Bitmap(DIB)
- Writing the DIB into a bitmap file.

#### **1. Creating a device context**

A device context is a windows data structure containing the information about the drawing attributes of a device such as a printer or a display. A device context is necessary for capturing a screen. A device context is created for the display using the "CreatDC()" function. Before creating a device context it is first declared and initialized. In this module two device context are created

- One Device context is used to point out the displayed device
- Another device context is used to copy the image from the display device into its device context

The second device context is used to create the DDB.

#### **2. Getting the image from the device context**

The image in the display from the first device context is copied into the second device context. first a bitmap object is created and initialized, then a bitmap is created with a width and height of the screen using the "Create Compatible Bitmap()" function. The device context used here is a display device context. Then this bitmap is

selected into the second device context and the bits of the display are copied using the function "BitBlt()". The image obtained is a device dependent bitmap which depends on the device upon which it is used. To view it in another system it has to be converted into a device independent bit map which is the next step.

### **3. Creating DDB into DIB**

In this step device dependent bitmap is converted into device independent. First the DDB information is read. The DDB bitmap contains the height, width, bits, bit count of the bitmap. The information corresponding to the DIB that is BITMAPINFOHEADER is written into it. The size of the image is calculated and the information in the BITMAPINFOHEADER is filled. Then the bits of the DDB is obtained and put into the buffer using the "GETDIBits()" function. Thus the device dependent bit map has to be stored in a file for transfer. Thus moving on to the next step.

### **4. Writing the DIB into a bit map file**

In this step the device independent bit map is written into a file. First an object is created for C File class which handles the file operations then the file is created with a required file name. First the header file information is entered saying it is a BMP file. After which the size of the file is filled. Then this information is written into the file followed by bits of DIB bitmaps.

#### **5.2.3. Communication Module:**

This module is used for establishing the connection between the client and the server system. The module used the TCP/IP protocol for the transfer of files and information across the network.

This module consists of two parts:

- Transfer Communicator.
- Receiving Communicator.

## 1. Transfer communicator:

This part deals with the transfer of the screen from the transfer application. The entire information passing to and from the receiving application is done handled by this part. This module deals with the establishment and passing of information across the network. The steps involved in this part is

- Creation of sockets
- Putting it in listen mode
- Accepting the connection
  
- **Creation of socket:**

In any networking project first a socket has to be developed. A socket is an object that represents an endpoint for communication between processes across a network transport. Without a socket the connection cannot be established.

The class used here is C Socket. In this class a socket is created with help of “Create ()” function. In this creation of the socket the port number has to be specified.

- **Putting in listen mode:**

The application is put in listening mode after the creation of the socket. In the listening mode the application checks if any request has been obtained for connection. At a time the listen mode is put to listen to only one request. The function used here is “Listen ()”.

- **Accepting a connection:**

As soon as there is a request the connection is established with the help of the function “Accept ()”.

## 2. Receiving communicator:

This part deals with the reception of the screen from the transfer application and the receiving application. The entire information passing to and from the transfer application is done handled by this part.

This module deals with the establishment and passing of information across the network. The steps involved in this part is

- Creation of socket
- Getting the IP address
- Connection to system

- **Creation of socket:**

In any networking project first a socket has to be developed. A socket is an object that represents an endpoint for communication between processes across a network transport. Without a socket the connection cannot be established.

The class used here is C Socket. In this class a socket is created with the help of “Create ()” function. In the creation of the socket the port number has to be specified.

- **Getting the IP address:**

The IP address of the system to be connected has to be given by the administrator. The IP address is obtained in the dialog box.

- **Connecting to system:**

After the IP address of the system to be connected is obtained, the connection is established with the help of the “Connect ()”function.

#### 5.2.4. Control Module

This module is responsible for sending files, messages and mouse information across the network between the transfer application and the receiving application.

This module has two parts

- Transfer control
- Receive control

- **Transfer control:**

In the transfer control all the operations after the establishment of connection is performed. The transfer control performs the following operations

- Send file
- Execute Mouse Operations
- Display messages

### **1. Send file:**

Once the connection has been established the capture module captures the screen and stored it in a file. This file is transferred from the transfer application to the receiving application by this module. The files are sent using the function “Send ()”.

### **2. Execute mouse operations:**

The mouse information is obtained from the receiving application. This information is executed via the function “mouse event ()” function.

### **3. Displaying messages:**

The message sent by the receiving application, which is displayed with the help of dialog box.

- **Receiving control**

The receiving control resides in the receive application. It is responsible for receiving the file that is sent by the transfer application, extracting the mouse information, sending messages.

The functions of the receiving control is

- Receive File
- Extract Mouse Information and send it
- Send the messages to the transfer application
- Sending shut down signal to transfer application

## 6. SYSTEM TESTING

Testing is an activity to verify that correct system is being built and is performed with intent of finding faults in the system. Testing is an activity, however not restricted to being performed after the development phase is complete. But this is to be carried out in parallel with all stages of system development, starting with requirements specification. Testing results, once gathered and evaluated, provide a qualitative induction of software quality and reliability and serve as a basis for design modification if required. A project is set to the incomplete without project testing.

System testing is process of checking whether the development system is working according to the original objectives and requirements. The system should be tested experimentally with the test data so as to ensure that the system works according to the required specification. When the system is found working, test it with actual data and check performance.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant "cost" associated with a software failure is motivation forces for a well planned, through testing.

### 6.1 Testing Objectives:

The testing objectives are summarized in the following three steps. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as yet undiscovered. A successful test is one that uncovers as-yet-undiscovered error.

### 6.2 Testing Principles:

All tests should be traceable to customer requirements. Tests should be planned long before testing begins, that is, the test planning can begin as soon as the requirement model is complete. Testing should be "in the small" and progress towards testing "in large". The focus of testing will shift progressively from progressively from programs, to individual modules and finally to the entire project. Exhaustive

testing is not possible. To be more effective, testing should be one, which has highest probability of finding errors

The following are attributes of good test:

- A good test has a high probability of finding an error
- A good test is not redundant
- A good test should be “ best of breed”
- A good test should be neither too simple nor too complex

### **6.3 Levels of Testing:**

The details of the software functionality tests are given below. The testing procedure that has been used is as follows

- Unit Testing
- Integration testing
- Validation testing
- Output testing

### 6.3.1 Unit Testing:

Unit testing is carried out to verify and uncover errors within the boundary of the smallest unit or a module. In this testing step, each module was found to be working satisfactory as per the expected output of the module. In the package development, each module is tested separately after it has been completed and checked with valid data. Unit testing exercises specific paths in the modules control structure to ensure complete coverage and maximum error detection.

### 6.3.2 Integration Testing:

Integration testing address the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high-order test are conducted. The main objective in this testing process is to take unit tested modules and build a program structure that has been dictated by design.

The following are the types of Integrated Testing:

- **Top-down integration:**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module. The module subordinates to the main program module are incorporated to the structure in either a depth first or breath-first manner.

- **Bottom up integration:**

This method designs the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low level modules are combined into clusters that perform a specific software sub-function.
- A driver (i.e.) the control program for testing is returned to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upwards in the program structure.

### **6.3.3 Validation Testing:**

At the end of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and correction testing begins.

#### **Validation test criteria:**

Software testing and validation is achieved through a series of black box tests that demonstrate conformity with the requirements are achieved, documentation is correct and other requirements are met.

### **6.3.4 Output Testing:**

Output testing is a series of different test whose primary purpose is to fully exercise the computer based. Although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

Output testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operations commence. The input screens, output documents were checked and required modifications made to suit the program specification. Then using the rest data prepared, the whole system was tested and found to be a successful one.

## **7. CONCLUSION AND FUTURE OUTLOOK**

### **7.1 Conclusion:**

The complete design and development of the system is presented in this dissertation. The system has user-friendly features. It is possible for any user to use this system.

The programming techniques used in the design of the system provide a scope for further expansion and implementation of any changes, which may occur in the future. The system has been tested by connecting with many systems and they provide satisfactory performance.

This system is developed with the specifications and abiding by the existing rules and regulations of the company.

Since the requirements of any organizations and their standards are changing day by day the system has been designed in such a way that its scope and boundaries could be expanded in future with little modifications. As a further enhancement this system can be integrated with any other system.

This system has been developed using Visual C++. The main aim behind the development of this system is to provide a solution for the administrator to watch over the client form the server.

### **7.2 Limitations:**

The system developed can monitor only one system at a time. It can be run only in windows platform. The system can monitor only those systems connected in one server and not many servers. The future enhancement can monitor multiple systems at a particular time.

### **7.3 Future Enhancements:**

There are many features that can be added to the system. Due to the insufficient time it could not be incorporated in this system. The future enhancements that can be provided are:

- Access to multiple clients at a particular time.
- Monitoring the system of another server from a server.

## REFERENCES

**David J. Kruglinski, George Shepherd, Scot Wingo, "Programming Microsoft Visual C++", Microsoft Press; Fifth Edition**

**Richard C. Leinecker and Archer Tom, "Visual C++ 6 Programming Bible", IDG Books India (P) Ltd; Fifth Edition.**

**Roger S. Pressman "Software Engineering and Application", McGraw Hill; Fourth Edition**

**John Paul Mueller "Visual C++ 6 from the Ground up", Tata McGraw Hill.**

*no relevant ref.*

# Appendices

---

## APPENDICES

### SAMPLE CODING

```
#include "MainWnd.h"
#include "WndProc.h"
#include "client.h"
#include "ServerIP.h"
#include "Compression.h"
#include "ColorMode.h"
#include "GridSpacing.h"
#include "Gdi.h"
#include "resource.h"

#define DISCONNECT_MSG "DISCONNECT"

// Pointer to the Server IP Class
extern CServerIP *pServerIP;

// Pointer to the Compression Class
extern CCompression *pCompression;

// Pointer to the Grid Spacing Class
extern CGridSpacing *pGridSpacing;

// Pointer to the Color Mode Class
extern CColorMode *pColorMode;

// Bitmap and Socket Variables
extern "C" char *pDIBitmap;
extern "C" BOOL fDIBitmap;
extern "C" int iWidth,iHeight;
extern "C" int iScreenWidth,iScreenHeight;
```

```

extern "C" int
    iWidth1,iWidth2,iHeight1,iHeight2;
extern "C" int
    iCompressionLevel;
extern "C" LPBITMAPINFOHEADER lpBitmapIH;
extern "C" SOCKET sClient;
extern "C" BOOL fConnected;
extern "C" struct GdiList GdiStart;
extern "C" struct GdiList *pGdiNode;
extern "C" int nGridX,nGridY;
extern "C" BOOL fGridX,fGridY;
extern "C" WORD bmBitsPixel;

```

// Connected, Sending and Receiving Variables

```

int iSent,iRecv;
char szMouseMessage[81];

```

// Constructor

```

CMainWnd::CMainWnd()

```

```
{
```

```
    // Initialize the Scroll Bars
```

```
    iHScrollPos = 0;
```

```
    iVScrollPos = 0;
```

```
    GdiStart.pNext = NULL;
```

```
    // Set the Initial Timer Interval
```

```
    m_iTimerInterval = 500;
```

```
}
```

// Destructor

```

CMainWnd::~~CMainWnd()

```

```
{
```

```
    // Delete the Pointer to the Main Window
```

```
    delete this;
```

```
// Free the Global Memory
if (fDIBitmap)
    free(pDIBitmap);

// Free the Global Regional Memory
pGdiNode = GdiStart.pNext;
while (pGdiNode)
{
    // Check if the Memory has been Allocated
    if (pGdiNode->Gdi.fDIBitmap)
    {
        free(pGdiNode->Gdi.pDIBitmap);
        free(pGdiNode->Gdi.pDIB);
    }

    // Move to the Next Element
    pGdiNode = pGdiNode->pNext;
}

// Set the Program Instance
void CMainWnd::SetInstance(HINSTANCE hInst)
{
    // Set the Instance Variable
    m_hInst = hInst;
}

// Handle the WM_CREATE Message
BOOL CMainWnd::OnCreate(HWND hWnd,CREATESTRUCT FAR
*lpCreateStruct)
{
    // Show the window according to WinMains nCmdShow
parameter
```

```
ShowWindow(hWnd,*(LPINT)lpCreateStruct-
>lpCreateParams);

// Set the Main Window Variable
m_hWnd = hWnd;

// Get a Handle to the Menu
m_hMenu = GetMenu(m_hWnd);

// Disable the Disconnect Menu Item
EnableMenuItem(m_hMenu,IDM_DISCONNECT,MF_BYCO
MMAND|MF_GRAYED);

// Disable the Color Mode Menu Item
EnableMenuItem(m_hMenu,IDM_COLORMODE,MF_BYCO
MMAND|MF_GRAYED);

// Set the Default Compression Level
iCompressionLevel = 0;

return TRUE;
}

// Handle the WM_DESTROY Message
void CMainWnd::OnDestroy(HWND hWnd)
{
    // Check if a DIB was Retrieved
    if (fDIBitmap)
    {
        // Kill the Refresh Timer
        KillTimer(m_hWnd,0);
    }

    // Check for a Current Connection
```

```
if (fConnected)
{
    // Send the Message to the Server to End the Thread

memset(szMouseMove, '\0', sizeof(szMouseMove));
strcpy(szMouseMove, DISCONNECT_MSG);

    // Send the Message to the Server
    iSent = Send(sClient, szMouseMove);

    // Close the Socket
    closesocket(sClient);

    // Cleanup
    WSACleanup();

    // Set the Connection State
    fConnected = FALSE;
}

// Post the Windows Quit message to the window
PostMessage(hWnd, WM_QUIT, (WPARAM)0, 0L);
}

// Handle the WM_SIZE Message
void CMainWnd::OnSize(HWND hWnd, UINT state, int cx, int cy)
{
    // Check if we have a DIB
    if (fDIBitmap)
    {
        // Get the Client Rectangle
        GetClientRect(m_hWnd, &m_Rect);

        // Set the Scroll Ranges for the New Screen Size
    }
}
```

```

        SetScrollRange(m_hWnd,SB_HORZ,0,iScreenWidth -
m_Rect.right,FALSE);
        SetScrollPos(m_hWnd,SB_HORZ,iHScrollPos,TRUE);
        SetScrollRange(m_hWnd,SB_VERT,0,iScreenHeight -
m_Rect.bottom,FALSE);
        SetScrollPos(m_hWnd,SB_VERT,iVScrollPos,TRUE);

        // Generate a WM_PAINT Message
        InvalidateRect(hWnd,NULL,FALSE);
    }
}

// Handle the WM_PAINT Message
void CMainWnd::OnPaint(HWND hWnd)
{
    PAINTSTRUCT        ps;
    HDC                hDC;
    struct GdiList      *pNode;
    int                iX1,iX2,iY1,iY2;

    // Start the Painting of the Screen
    hDC = BeginPaint(hWnd,&ps);

    // Draw the Full Bitmap
    if (fDIBitmap)
    {
        // Get the Client Rectangle
        GetClientRect(m_hWnd,&m_Rect);

        // Only Create the Bkg DC and Bitmap on the Fly if
Offline (To Preserve Screen State)
        if (!fConnected)
        {

```

```

// Build the Full Bitmap into a Mem DC from
the Regions
m_hMemDC = CreateCompatibleDC(hDC);

// Create a Compatible Bitmap for the Mem DC
m_hDDBitmap =
CreateCompatibleBitmap(hDC,iScreenWidth,iScreenHeight);

// Select the Bitmap into the Memory DC for
Drawing
SelectObject(m_hMemDC,m_hDDBitmap);
}

// Loop through the Regions
pNode = GdiStart.pNext;
while (pNode)
{
// Check if we have a DIB for the Region
if (pNode->Gdi.fDIBitmap)
{
// Draw the Region into the Memory DC
StretchDIBits(m_hMemDC,
pNode->Gdi.iWidth1,pNode-
>Gdi.iHeight1,
pNode->Gdi.lpBitmapIH-
>biWidth,pNode->Gdi.lpBitmapIH->biHeight,
0,0,
pNode->Gdi.lpBitmapIH-
>biWidth,pNode->Gdi.lpBitmapIH->biHeight,
(LPBYTE)pNode-
>Gdi.lpBitmapIH + (pNode->Gdi.lpBitmapIH->biSize + (1 <<
pNode->Gdi.lpBitmapIH->biBitCount) * sizeof(RGBQUAD)),
(LPBITMAPINFO)pNode-
>Gdi.lpBitmapIH,DIB_RGB_COLORS,SRCCOPY);

```

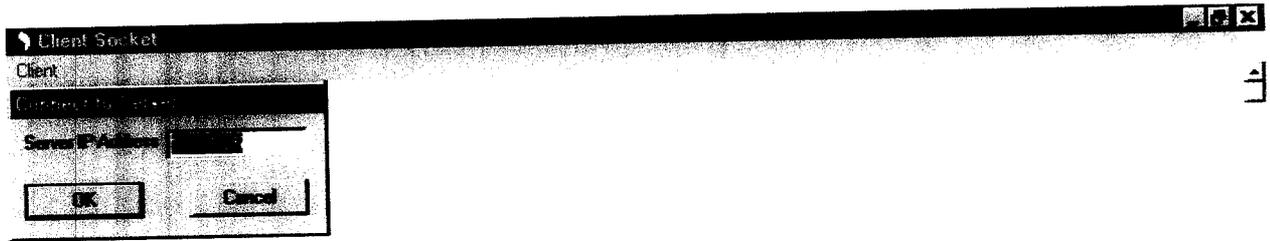
Screen Shots

---

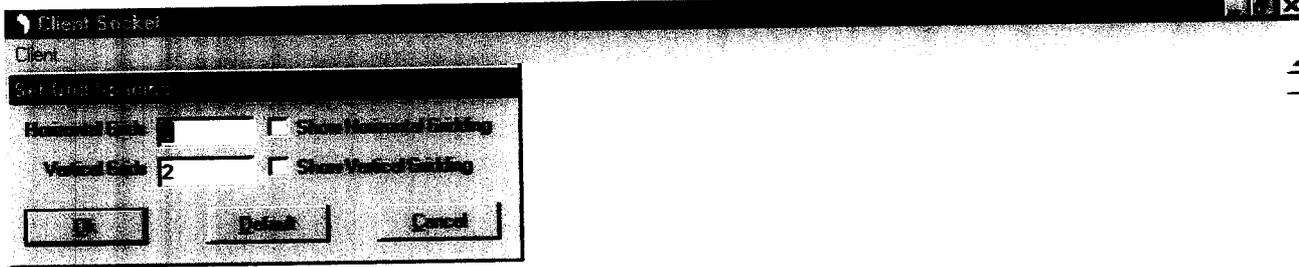
## SCREEN SHOTS



• | |



411



◀ ▶