PERSONAL FIREWALL SYSTEMS

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

**BACHELOR OF ENGINEERING –
INFORMATION TECHNOLOGY**

BHARATHIAR UNIIVERSITY, COIMBATORE.

SUBMITTED BY

K.MAHESH
S.RAMKUMAR
M.VIKRAM

External Guidance
Mr. Murugandham, B.E,
BIC TECH,
Coimbatore.

Internal Guidance
Mrs. S.Devaki,B.E,M.S.,
Assistant Professor.
Dept of Computer Science.
Kumaraguru College of Technology.
Coimbatore.

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore
MARCH 2002

Certificate

BIC Tech

Ref:ADM/GEN/497 11/03/2002

PROJECT CERTIFICATE

To Whomsoever It May Concern

Certified that this thesis "**PERSONAL FIREWALL SYSTEMS**" is the bonafide work of MR.K.MAHESH, MR.S.RAMKUMAR,MR.M.VIKRAM who have carried out the project under our supervision, certified further that to the best of our knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate. The Documentation submitted by the group was well compiled and the project completely satisfied our expectations.

For BIC Tech(P) Ltd.

Muruganandam.M

Manager-Admn.

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore - 641006.

This is to certify that the project work entitled
"PERSONAL FIREWALL SYSTEM "
has been submitted by

Mr. K.MAHESH	9827S0013
Mr.S.RAMKUMAR	9827S0022
Mr. M.VIKRAM	9827S0031

In partial fulfillment of the award of the degree of
Bachelor of Engineering
Information Technology
of Bharathiar University , Coimbatore
during the academic year 2001 - 2002

14/3/02 10/3/2002

Head of the Department Guide

Certified that the candidate was examined by us in the Project Work
Viva Voce Examination held on 18/03/2002 .

Rajini 18·3·02 18/3/2002

Internal Examiner **External Examiner**

Acknowledgement

Any endeavor over a long period can be successful only by the advice and support of many well-wishers. We avail this opportunity to express our gratitude and appreciation of all of them.

We express our profound respect and sincere gratitude to our Principal **Dr. K.K.Padmanaban, B.Sc. (Engg), M.Tech, Ph.D.,** for having provided the necessary facilities to complete this project.

We are greatly indebted to our Head of the Department **Dr.S.Thangasamy, B.E (Hons), Ph.D.,** Computer Science and Engineering, Kumaraguru College of Technology, for our source of inspiration and encouragement rendered by him.

We are greatly privileged to express our deep sense of gratitude to our guide **Mrs. S.Devaki, B.E, M.S.,** Asst.Professor, Department of Computer Science and Engineering, who has been a motivating force behind all our deeds.

We wish to extend our gratitude to **Mr.Muruganandam.M,** Member – Technical Staff, BIC TECH, Coimbatore for allowing us to carry out this project work at his concern and guiding us to complete this project successfully.

Last but not the least, we wish to thank all our friends for their continuous support and encouragement during the course of this project.

Dedicated to our
beloved parents and
well wishers

SYNOPSIS

Java is a blend of the best elements of its rich heritage combined with the innovative concepts required by its unique environment. In short Java is a small, simple, safe, object-oriented, interpreted or dynamically optimized, byte-coded, architecture-neutral, garbage-collected, multithreaded programming language with a strongly typed exception-handling mechanism for writing distributed, dynamically extensible programs.

Innumerous Java applications are being developed in recent times and the community of java programmers is constantly on the rise. Though the major development packages for java were created with fundamental console behavior, recently a lot of building tools have propped up.

The major challenge to implement the general architecture is how to trade off among usability, flexibility, security and performance concerns. But the present day networks provides the minimal amount of the above features. This project tries to provide flexibility and security within a set of well defined rules.

Introduction part of the project deals with the limitations of the existing networks. Design document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. Product testing includes unit testing and integration testing.

CONTENTS

Introduction

1.1 Existing system and its limitations

Traditional Personal Firewall Assistant

A **firewall** puts up a barrier that controls the flow of traffic between networks. Application-level proxy servers today provide the highest level of protection. A **proxy server** is a component of a firewall that controls how internal users access. A typical application-level firewall can provide proxy services for applications and protocols like FTP (file transfers) and HTTP (Web services). Firewall allows to strip computer names to hide internal systems, and evaluates the contents of packets for **appropriateness** and security.

Stateful Inspection Techniques

The server remembers contents original request like port number, and source and destination address. This is referred to as **saving the state**. When the system responds to request, the firewall server compares the received packets with the saved state to determine if they are allowed in.

Features incorporated in any firewall

- ***Accounts***

User authentication is key to any network service. Significant services and specific requests are provided only upon the reception of a request from an authenticated account. The user specifies an account and its password to logon to the network. This represents the user's identity on behalf of all the programs that run during the logon session.

➢ *Domains & Account Matching*

Each user has an account stored on only one server named as a "local account". When a session involves service requiring any other server "account matching" principle is used to provide services. Here local Server provides authentication to enable file access in the other servers.

➢ *Rights*

Rights are special properties that administrators assign to accounts. Account's Rights propagate to all the programs run in the local or remote logon sessions. Each server has his own Rights database that associates Rights with some or all of the accounts usable on that computer. Rights that govern local and remote logon are given the utmost consideration in the network firewall.

➢ *Access Control Lists*

Along with authentication, access control lists (ACL's) form the cornerstone of any firewall. An ACL designates who can access the object in what manner. The kinds of access it allows differs from object to object but are generically: read, write, delete, and change the ACL of the object, or make one the owner of the object. By creating an object one becomes the owner of it. The owner of an object can always change its ACL.

Every data object on a network, whether it's a user data file, Registry key, printer, or user-invisible "device object" has an access control list (ACL). "Bypass Traverse Checking " is a mysterious method that allows access to an object in a directory that is not accessible to user. BTC is prevented in the network firewall by specifying it as a permission that could be disabled or enabled at specific request.

➢ *The Registry & System Root Directory*

The internal hierarchy serves as the root of all accessible data objects on the system including the file system and Registry. It's critical that many items in this tree have proper ACL's. Registry is used to secure applications from being deleted from the network. Hence controlled access is provided to the registry and the system directory to prevent data corruption.

TCP/IP

Transmission control protocol / Internet protocol has emerged as the standard protocol for use in the Internet.

All communication via the Internet have to use the TCP/IP. This protocol was written very back in 1970's by a group of students under the guidance of Dr.Vinton Cerf and Dr.Robert Kahn.The TCP/IP model developed by them is very reliable and its use on Internet is a testimony to its scalability. It is worthwhile to discuss their main features briefly.

Main Features:

1) Network level interconnection:

TCP/IP inserted between technology dependent communication mechanisms and application programs, will hide the low level details and make the collection of networks appear to be a single large network. Therefore, the user can transfer data between two machines irrespective of the multiple under lying hardware technologies using a single protocol. This is achieved by having the layer beneath IP, use underlying technology-dependent communication facilities.

2) Connection oriented services:

Running TCP over IP provides a reliable end to end byte stream over an unreliable internetwork. This is achieved by having acknowledgements for each unit of data transferred. The IP layer gives no guarantee that TCP uses time out and retransmits datagrams do not arrive in order TCP/IP reassembles it into proper sequence thereby giving impression of reliability that IP does not provide.

3) Routing:

The IP layer gets the destination address from the datagram header and routes them to correct destination via gateways known as routers. These routers incorporate IP and use destination network address when routing a packet. The layers above assume packet is delivered at destination.

4) Congestion Control:

When load offered to any network is more than it can handle, congestion builds up. Both TCP/IP manage congestion with most of heavy lifting done by TCP. TCP detects congestion and suitably adjusts sender's window size to control congestion.

Limitations:

The TCP/IP have few limitations and they are

1) Security:

The recent attacks on YAHOO, hotmail, etc prove the TCP/IP offers virtually nil security. It is upto the application layer to implement the security. Therefore, the protocol fails if it is tampered within the layers below. In this era of e-business, this is not a good scenario and much work is being done on it.

2) Flexibility:

The TCP/IP offers no flexibility or customizability to the users. The user is powerless to change the way his data is transferred at transport and network layer levels. The user can only change his application layer protocol. This lack of flexibility means that user has to use TCP/IP for transferring mail, audio, video, and even e-transactions over Internet.

3) Essence

The TCP/IP suite though being for around 3 decades does not offer enough advantages to user to carry on using it the need has now arisen for protocol which can be upgraded easily in tune with the rapid advancements made in fields of networking.

1.2 Proposed System and its advantages

Actually this project explains the interaction between one Server and many clients. This is the software which would take care of File Transfers in a computer network. This project manages the Files created in each and every node of the network.

The server grants the permission to the clients whether they can get the files or not. After the server is connected to the granted clients , the Server shares the required Files with the clients. If a particular change is made in the

Server side, the reflection is made in the client side.The received files can be saved in the client side with the help of Auto-fetch and save option. The request made by the client to server is done or in fact reflected with the required files.

Advantages

1. No dedicated system is required since the project is based on Server's response
2. Client Side users can easily save the files they received.
3. Any changes in files is shown on the screen immediately
4. Anonymous users cannot have access if they are not registered users.

System Requirements

2.1 Product Definition

2.1.1 Problem Statement

The product is concerned with the sending and receiving of files within a network connection with added facilities of html and text files, workflow management, all with respect to response of Server To the clients.

2.1.2 Functions to be provided

The various functions to be provided to the users are the sending and receiving files with various facilities, workflow management used for automatically sending files to other designated authorities and saving at the client end.

2.1.3 Processing Environment

Hardware Specification

The proposed system is being implemented on the following hardware configuration.

Minimum configuration

Processor	- Pentium I
Speed	- 600 MHz
Main Memory	- 333 MB
Monitor	- 800x600 at 16-bit true color

Recommended Configuration

Processor	-Pentium III
Speed	- 750 MHz
Main Memory	- 512 MB
Monitor	- 800x600 at 32-bit true color

Software Specifications

Language	- JAVA

Operating System

Windows NT/ 98/95

2.1.4 Solution Strategy

The problem was approached in a step-by-step fashion. First and foremost the functions and their purposes are learnt. Then analysis is done on problem definition. Then the rough draft is made on the analysis and finally ends up in a solution by breaking down the defined problem into four modules.

The solution is to have a Link option as the Interface for this product. The Server can choose the desired link at any time of instance and the file is sent to the client

2.1.5 Product Features

The main feature of this package is authentication packed package. The server grants accesses to the clients. Another important feature of our project is the possibility of auto-save option The workflow management order and access can be formed and changed dynamically with the requirements. The requirements of the clients are studied and are met which ultimately leads to client sharing with Server. The response time is less and it is user friendly too.

2.1.6 Acceptance Criteria

The acceptance criteria is based on the server permission. The server grants permission to the clients based on the request. The request from client side is based on files.

2.2 Project Plan

Life Cycle Model

The Spiral Model is proposed to be life cycle mode followed while developing the software. It provides the potential for the rapid development of incremental version of the software. The software is developed in the series of incremental releases. The spiral model has six tasks region.

STEP 1:

- Terminology : CLIENT Communication
- Work Product :

The software programmer defines about their existing communication between the server and the client. Server grants permission based on ip-address or system name.

STEP 2:

- Terminology : Planning
- Work Product : Analysis of the product definition.

What function the product has to Perform, Processing environment, Product features, Programming Language and development tools to be implemented are all decided in this stage.

STEP 3:

- Terminology : Risk Analysis
- Work Product :

Technical Risk:

Since Design Document plays a very vital role in coding, Preparation of Design Documentation consumes more time. Therefore coding is based on how we design the screen pages and the various links, what are all the logics, concepts that can be used. Once the design is made the coding can be started from scratch without any confusion.

Managerial Risk:

For every module time limits are set to be fixed for its completion. The project duration is between Nov – March. With in the given time slot for each module it has to be completed. The modules cannot be postponed because of the construction of various individual modules. If there is a delay in any one of the modules there is a delay in completion of the whole project. Therefore the modules have to be completed in the appropriate time as per the schedule.

STEP 4:

- Terminology: Engineering.
- Work Product: SRS

Based on the needs, the software requirement specification is prepared. SRS includes Product Overview, Processing

- Work Product :

Customer Evaluation is that the feedback of the customers after seeing the partial or final outcome of the product. Even then also the product may be undergone for change, if the customer is in need of.

Documents to be Prepared

It is suggested that the following documents can be prepared during the time of the project.

- A System Definition consisting of a Product Definition and a Project Plan.
- A Software Requirements Specification.
- A Design document consisting of external design, architectural design, and detailed design specification.
- A test Plan.
- A User's manual.
- A properly documented, debugged, and tested program.
- A project legacy document.

Software Requirements and Specifications

Environment, External interface and dataflow, Functional Specifications, Performance requirements, Exception condition & handling, Early subsets, Foreseeable modification, Acceptance criteria, Design guidelines.

- Work Product: Design Document.

Based on the needs, the Design document is prepared. Designing plays an important role during coding. Once the design is framed well, the programmer can start the coding very easily. Design document includes external design specification, Architectural design overview and Detailed design Specification.

STEP 5:

- Terminology : Construction and Release.
- Work Product :

* Front End Design like adding menu, popup menu with various images and gif images various attaching handlers to link options or its functions
* Coding for the entire menu and popup menu that accept through click.
* All the transactions between the clients and the Server are done through coding
* Preparing user Manual.
* Preparing Test Plan.
* Demonstrating the whole project.

STEP 6:

- Terminology : Customer Evaluation.

3.1 Introduction

PURPOSE:

The purpose of this SRS is to elaborate the requirements of Personal Information Systems and Automated File Transfer .This is the software which would take care of file transfer in a computer network. This project manages the Files created in each and every node of the network.

SCOPE:

This SRS focuses mainly on the requirement to be met by the Information Systems. The well-structured format and descriptive nature of the SRS is aimed at aiding the developers in developing an efficient File Manager in a network.

OVERVIEW:

Exploring further the software requirement specification gives information about the specific requirements like functionality, usability and supportability of the project developed.

OVERALL DESCRIPTION:

-In the layman's perspective, the Automated file transfer is viewing of files across the network by authorized users.

-For a developer it gives a comprehensive enumeration of various aspects of files sharing across the network.

-The objective of the project ends with developing a comprehensive File Transfer across the network for privileged network users.

3.2 Functionality.

The functionality of the project can be explained as taking care of file sharing across the network.

3.3 Start up:

The user is prompted to specify the path of the file he wants to view across the network. Each node has a list of server files it intends to share in the network.

3.4 Processing:

The application program shares files managed by the application to privileged users as specified by the owner of the file.Updated file content is displayed to the end user.

3.5 Messages:

Messages allows users to know the success of intended operation. Messages indicate success or failure of connection establishment and file transfer.

3.6 Interfaces:

User Interfaces: User Interface is graphical based for better interpretation. Browsing across the file system of each node helps to specify the information in a better fashion.

Communication Interfaces: This information is shared across the network using TCP/IP sockets .The front end obtains the data from these sockets and displays them.

Design Document

4.1 Introduction

System design is an iterative process through which requirements are translated into a "Blue Print" for constructing the software. The following are the characteristics that a design document should possess,

1. The design must implement all the explicit requirements contained in the analysis model and it must accommodate all of the implicit requirements designed by the customers.
2. The design must be readable and understandable guide for those who generate and test code and subsequently maintain the software.
3. The design should provide a complete picture of software, addressing the data, functional and behavioral domains from an implementation perspective.

4.2 Design Phase Activities

Following the guidelines the various activities that were to be carried during the design phase of the **PERSONAL FIREWALL SYSTEM** are...

- INPUT DESIGN
- OUTPUT DESIGN
- SERVER DESIGN
- CLIENT DESIGN
- SCREEN DESIGN

The whole system consists of the following modules... The description of the modules in terms of system design is...

Process Design

Software engineering process is the glue that holds the technology layers together and enables rationale and timely development of the computer software. Process defines framework for a set of key process area that must established for effective delivery of the software. The key process areas form the basis management control of software projects.

Screen Design

The modules consist of screens in which menu names are server, client and help. The popup menu for server menu consist of fields like IPADDRESSES,MACHINES,SERVER FILES AND EXIT. The popup menu for CLIENT are CONNECT TO SERVER,AUTOFETCH AND AUTOFETCH AND SAVE OPTIONS.

Product testing

In the development of the software involves series production activities were opportunities of injection of Human fallibilities are enormous. Errors may begin to occur at the very inception of the process were the objectives may be erroneously or imperfectly specified, as well as [in] later design and development stages. Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity. Software testing is very critical of the software quality assurance and represents the ultimate review of specification, design and coding.

Objectives of Testing

The objectives of testing are as follows:

- Testing is a process of executing a program with the intent of finding an error.
- A good test is one that has a high probability of finding an undiscovered error.
- A successful test is one that uncovers an as-yet undiscovered error.

Test Cases

The testing of software is a means of accessing or measuring the software to determine its quality. The area of testing is one of the key process areas in ensuring the quality the software. A detailed test case was designed after requirement stage with 100% condition coverage, boundary conditions (below, at, above), stress conditions (huge data, abnormal condition).

Testing is done to ensure that each instruction is executed at least once. All parts and branches are executed at least once. Identify critical parts, proper updation of intermediate and temporary files and finally check for tested output. Test results of invalid conditions and errors were recorded for corrections. After completion of individual module testing, integration testing is done with the whole integrated system.

Testability:

Software testability determines how easily a program can be tested.

- ➢ Operability – The better it works the more efficiently it can be tested.
- ➢ Observability – What you see is what you Test. Incorrect output is easily identified.
- ➢ Controllability – The better the control over the software the greater is the optimization of testing.
- ➢ Understandability – The more the information the smarter is the test.

Foreseeable modifications and Enhancements

We cannot consider our project to be complete no matter how much we work on it. There will always be flaws and there will be thousands of solutions to rectify them. According to us many of the flaws are already rectified but some may need more attention.

Our project, at present, has the option of transferring text and html files. But, in future it can be enhanced such that it can transfer Images, Documents, and other file types.

The second option is that enhance the package in future to have a list in which the nodes connected in network can be pinged and displayed when the package is executed.

Also we can modified the present configuration in which a single and multiple clients supported are changed so that multiple servers and multiple clients can be supported.

Conclusion

The **PERSONAL FIRWEWALL SYSTEMS** has been successfully designed and implemented and tested. The software works effectively and efficient to its requirement..

The administrator has sole rights over the system. Any restriction over the system can be imposed only through the administrator. The workflow can be changed dynamically depending on the requirement policies of the company. The project has helped us to gain a wide knowledge in java and its components.

Bibliography

Text References:

1. Patrick Naughton & Herbert Schildt, 'Java Complete Reference', Tata McGraw Hill, 2000.
2. Michael Morrison, 'Java 1.1 Unleashed', Sams net, 1996
3. MSDN library – October 1999

Web References:

www.sun.com

Net References:

http://www.rational.com/java/resources

Appendix

SOURCE CODE

Code for the File Transfer

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;
import java.sql.*;
import java.text.*;
import java.net.*;
import javax.swing.*;
import javax.swing.text.*;
import javax.swing.event.*;
import javax.swing.border.*;

 public class FileTransfer implements  WindowListener
 {
 /**
  * A class which implements a mechanism to activate server, client and
intiate transefer
  * add/edit/view operations.
  */
        JMenuBar mb;
        JMenu m1,m2,m3,m4,m5,misa,mia,mig;
        JMenuItem
mi1,mi2,mi3,mi4,mi5,mi6,mi7,mi8,mi9,mi10,mi11,mi12,mi13, mi20,
mimove, miinsert, logfile ,mi14, gotopage, pickacolor,
changedState,zoomin,revertZ;
        JMenuItem mis1,mis2,mis3,mis4,mis5,mis6,viewgr,hidegr,enterent:
        JCheckBoxMenuItem drag;
        JToolBar toolBar;
        JOptionPane jo;

        int bbc;
        int mabc;

        boolean flag;
```

```
client cli;

static FileTransfer ft ;
String servername,inside, clientname;
Socket s;

JButton   newBut,openBut,saveBut,exitBut,addBut,removeBut,
 removeallBut,findBut,replaceBut,helpBut;
JButton   dragBut,exBut;

JButton  printBut;
Vector   tempvect;


//JPanel toolBarPanel;

Container c;
public static JFrame f;
public static JDesktopPane jdp;
JDialog j1,j2,j3,j4,j5;


public FileTransfer()
{
        main();

}

public void main()
{
        f = new JFrame("File Transfer");
        f.setBounds(0,0,800,575);

        jdp = new JDesktopPane();
        jdp.setBounds(0,60,793,490);
        jdp.setBackground(Color.gray);
        jdp.setBorder(BorderFactory.createLoweredBevelBorder());

        c = f.getContentPane();
        c.setLayout(new BorderLayout());
        f.setResizable(false);
        f.getContentPane().add(jdp, BorderLayout.CENTER);


        mb = createMenuBar();
        f.setJMenuBar(mb);
        f.setIconImage(new ImageIcon("chiplogo.jpg").getImage());
        new fr2(f,jdp);

        f.addWindowListener(this);
        f.setVisible(true);

   try {

    UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.
    WindowsLookAndFeel");
   }
        catch(Exception e)
        {
        }

 } //end method main

 public static void main(String args[])
 {
        //dates();

        ft =  new FileTransfer();
 }

public void windowActivated(WindowEvent e){ }
public void windowClosed(WindowEvent e)
  {
        System.exit(1);
  }
public void windowClosing(WindowEvent e){ } // window closing
public void windowDeactivated(WindowEvent e) { }
public void windowDeiconified(WindowEvent e) { }
public void windowIconified(WindowEvent e) { }
```

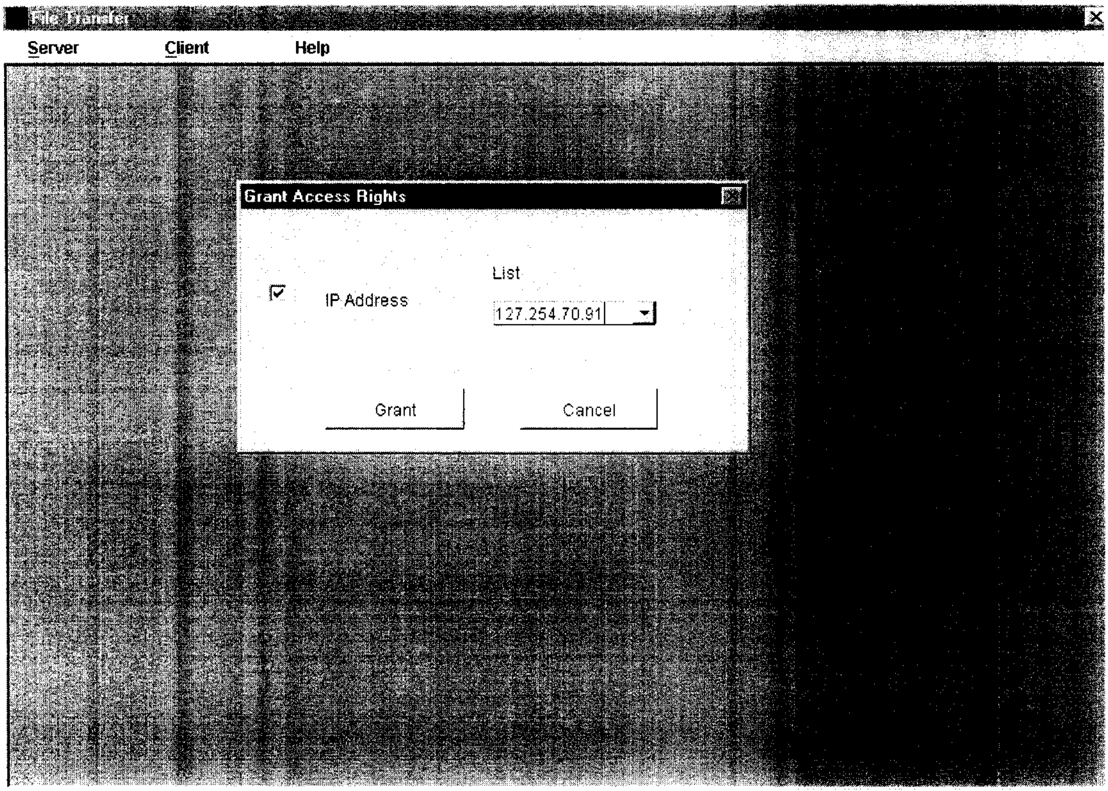
```
        public void windowOpened(WindowEvent e)
{System.out.println("this is opening");}

        protected JMenuBar createMenuBar()
        {
                mb = new JMenuBar();

                m1 = new JMenu("Server        ", true);
                m1.setMnemonic(KeyEvent.VK_S);

                ImageIcon iconNew = new ImageIcon("new.gif");
                Action actionNew = new AbstractAction("IP Addresses") {

                        public void actionPerformed(ActionEvent e)
                        {
                        new Ipaddress(f,jdp);
                }
        };

                mi1 = m1.add(actionNew);
                m1.addSeparator();

                ImageIcon iconOpen = new ImageIcon("open.gif");

                Action actionOpen = new AbstractAction("Machines      ")
                {

                        public void actionPerformed(ActionEvent e)
                        {
                        new Names(f,jdp);
                }
        };

                mi2 = m1.add(actionOpen);
                m1.addSeparator();

                ImageIcon iconSave = new ImageIcon("save.gif");
                Action actionSave = new AbstractAction("Server Files") {

                        public void actionPerformed(ActionEvent e)
                {
                        JFileChooser chooser = new JFileChooser();
                        ExampleFileFilter filter = new
ExampleFileFilter();
                        filter.addExtension("txt");
                        filter.addExtension("java");
                        filter.addExtension("html");
                        chooser.setFileFilter(filter);
                        chooser.setMultiSelectionEnabled(true);
                        int returnVal =
chooser.showDialog(FileTransfer.ft.f,"Select");

                  if(returnVal == JFileChooser.APPROVE_OPTION)
                  {
                    String two= chooser.getSelectedFile().getPath();
                    File a [] = new File
[chooser.getSelectedFiles().length];
                    a = chooser.getSelectedFiles();
                    System.out.println("this isthe size of the vector in
files"  + a.length);
                    String fin = two;
                    System.out.println(fin);
                    String p = "|";
                    byte pipe [] = p.getBytes();

                            if(a.length >0)
                            {


                    try
                    {
                    OutputStream f1 = new FileOutputStream("file.txt");

                    for( int i = 0 ; i < a.length; i++)
                    {
                    String s = a[i].getPath();
                    byte buf [] = s.getBytes();
                    f1.write(buf);
                    f1.write(pipe);
                    }
                    }
```

```
                   catch(Exception de)
                   {
                               System.out.println(de);
                  }
                      }
                      else
                      {
                           try
                           {
                           OutputStream f1 = new
FileOutputStream("file.txt");
                           byte buf [] = two.getBytes();
                           f1.write(buf);
                           f1.write(pipe);
                         }
                              catch(Exception fe)
                              {
                                   System.out.println(fe);
                              }
                      }
          }
          }
     };

          mi20 = m1.add(actionSave);

          Action actionSaveAs = new AbstractAction("Start Server ") {
          public void actionPerformed(ActionEvent e)
          {
               try
               {
               int port = 1234;
               }
               catch(Exception de)
               {
                    System.out.println(de);
               }
          }
    };
```

```
            ImageIcon iconPrint = new ImageIcon("print.gif");
            Action actionPrint = new AbstractAction("            ") {

                    public void actionPerformed(ActionEvent e)
                    {
                    }
        };

        m1.addSeparator();

            ImageIcon iconExit = new ImageIcon("exit.gif");
            Action actionExit = new AbstractAction("Exit") {

                    public void actionPerformed(ActionEvent e)
                    {
                    System.exit(0);
            }
        };

            mi4 = m1.add(actionExit);

        mb.add(m1);

            m2 = new JMenu("Client       ");
            m2.setMnemonic(KeyEvent.VK_C);

//***********************
            mia = new JMenu("File       ");

            ImageIcon iconAdd = new ImageIcon("add.gif");
            Action actionAdd = new AbstractAction("File     ")
            {
                    char [] sla = { '/'};

                    public void actionPerformed(ActionEvent e)
                    {
                    Vector size = new Vector();
                    String  a = jo.showInputDialog(FileTransfer.ft.f,"Enter
the path of the file ","Fetch From
Server",JOptionPane.INFORMATION_MESSAGE);
```

```
                for ( int i = 0; i <a.length(); i++)
                {
                        if ( a.charAt(i) == '\\')
                        {
                                String s = Integer.toString(i);
                                size.add(s);
                        }
                }
                StringBuffer ins_slash = new StringBuffer(a);
                for( int i= 0 ;i <size.size();i++)
                {
                        String s =(String)size.elementAt(i);
                        int slashpt = Integer.parseInt(s)+i;
                        ins_slash.insert(slashpt,'\\');
                }
                System.out.println(" the final String is  : " + new
String(ins_slash));
                String tosend = new String(ins_slash);
                cli = new client(tosend,f,jdp);
                //new newwindow1(f,jdp);
                }

        };
        mi5=mia.add(actionAdd);
        mia.addSeparator();

        ImageIcon iconRemove = new ImageIcon("remove.gif");
        Action actionRemove = new AbstractAction("Close")
        {
                public void actionPerformed(ActionEvent ae)
                {
                        cli.ji.dispose();
                }


        };

        mi6 = mia.add(actionRemove);
```

INPUT SCREEN



Grant Access Rights

☑ IP Address

List: 127.254.70.91