SECURED ARCHITECTURE FOR WORKSTATIONS

A PROJECT REPORT

Submitted by

S. PRABAHAR **71202104028**
K. VELMURUGAN **71202104067**

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2006

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"SECURED ARCHITECTURE FOR WORKSTATIONS"** is the bonafide work of "PRABAHAR .S, VELMURUGAN .K" who carried out the project work under my supervision.

SIGNATURE

Dr. S. THANGASAMY

HEAD OF THE DEPARTMENT

Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore – 6

SIGNATURE

Ms. S. RAJINI

SUPERVISOR

Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore – 6

Viva-voce examination is conducted on 02.05.2006

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project entitled **"SECURED ARCHTECTURE FOR WORKSTATIONS"** is done by us and to the best of our knowledge a similar work has not been submitted to the Anna University or any other Institution, for fulfillment of the requirement of the course study.

This report is submitted on the partial fulfillment of the requirements for the award of the Degree of Bachelor of Computer Science & Engineering of Anna University, Chennai.

Place: COIMBATORE

Date: 26.04.06

PRABAHAR .S

VELMURUGAN .K

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

The successful completion of our project can be attributed to the combined efforts made by us and the contribution made in one form or the other, by the individuals we hereby acknowledge.

We express our sincere gratitude to the respected and beloved Principal, **Dr. K.K. Padmanaban, B.Sc.,(Engg)., M.Tech., Ph.D., MISTE.,** and **FIE** for the opportunity and the excellent infrastructure provided to carry out this project work.

We are deeply indebted to the Dean Of the Department of Computer Science and Engineering **Dr. S. Thangasamy B.E., Ph.D.,** for his kind encouragement for the successful completion of this project.

We are extremely grateful to our guide **Ms. S. Rajini B.E.,** Senior Lecturer in CSE Department, for her immense and valuable guidance, constructive remarks and outstanding cooperation rendered at any stage of our project.

We also thank our project coordinator **Prof. Devaki, M.S.,** and our beloved class advisor **Mrs. V. Amutha M.E.,** for their invaluable assistance.

We are also thankful to all teaching and non-teaching staff of Computer Science and Engineering department for their kind help and encouragement in making our project successful.

We extend our special thanks to our **family members, friends and classmates** who have rendered their help in this endeavor.

Finally we thank one and all who helped directly and indirectly for the development of this project.

ABSTRACT

ABSTRACT

A number of security problems with the Internet can be remedied or made less serious through the use of existing and well-known techniques and controls for host security. A firewall is one of several ways of protecting private network from another untrusted network. A firewall can significantly improve the level of site security while at the same time permitting access to vital Internet services.

Firewall implementation is developed specifically to provide security using firewalls for systems connected in WAN and MAN. This firewall approach provides numerous advantages to sites by helping to increase overall host security.

A firewall can greatly improve network security and reduce risks to hosts on the subnet by filtering inherently insecure services. As a result, the subnet network environment is exposed to fewer risks, since only selected protocols will be able to pass through the firewall.

The system is designed specifically for organizations having widespread Departments. Here the International Standard Protocols HTTP and FTP are followed for web server and FTP server respectively.

CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO FIREWALL

A number of security problems with the Internet can be remedied or made less serious through the use of existing and well-known techniques and controls for host security. A firewall can significantly improve the level of site security while at the same time permitting access to vital Internet services.

A firewall is a device that allows multiple networks to communicate with one another according to a defined security policy. They are used when there is a need for networks of varying levels of trust to communicate with one another. For example, a firewall typically exists between a corporate network and a public network like the Internet. It can also be used inside a private network to limit access to different parts of the network. Wherever there are different levels of trust among the different parts of a network, a firewall can and should be used.

Firewalls are similar to routers, in that they connect networks together. Firewall software runs on a host, which is connected to both trusted and untrusted networks. The host operating system is responsible for performing routing functions, which many operating systems are capable of doing. The host operating system should be as secure as possible prior to installing the firewall software.

This not only means knowing how the operating system was installed but also making sure that all of the security patches are applied and that unnecessary services and features are disabled or removed.

Firewalls are different from routers in that they are able to provide security mechanisms for permitting and denying traffic, such as authentication, encryption, content security, and address translation. Although many routers provide similar capabilities (such as high-end devices from Cisco), their primary function is to route packets between networks. Security was not part of their initial design but rather an afterthought. A firewall's primary function is to enforce a security policy, and hence the system is designed with firewall security.

Why Firewalls

The general reasoning behind firewall usage is that without a firewall, a subnet's systems expose themselves to inherently insecure services such as NFS or NIS and to probes and attacks from hosts elsewhere on the network. In a firewall-less environment, network security relies totally on host security and all hosts must, in a sense, co-operate to achieve a uniformly high level of security. The larger the subnet, the less manageable it is to maintain all hosts at the same level of security. As mistakes and lapses in security become more common, break-ins occur not as a result of complex attacks, but because of simple errors in configuration and inadequate passwords.

A firewall approach provides numerous advantages to sites by helping to increase overall host security. The following sections summarize the primary benefits of using a firewall.

Protection from Vulnerable Services

A firewall can greatly improve network security and reduce risks to hosts on the subnet by filtering inherently insecure services. As a

result, the subnet network environment is exposed to fewer risks, since only selected protocols will be able to pass through the firewall.

For example, a firewall could prohibit certain vulnerable services such as NFS from entering or leaving a protected subnet. This provides the benefit of preventing the services from being exploited by outside attackers, but at the same time permits the use of these services with greatly reduced risk to exploitation. Services such as NIS or NFS that are particularly useful on a local area network basis can thus be enjoyed and used to reduce the host management burden.

Firewalls can also provide protection from routing-based attacks, such as source routing and attempts to redirect routing paths to compromised sites via ICMP redirects. A firewall could reject all source-routed packets and ICMP redirects and then inform administrators of the incidents.

Controlled Access to Site Systems

A firewall also provides the ability to control access to site systems. For example, some hosts can be made reachable from outside networks, whereas others can be effectively sealed off from unwanted access. A site could prevent outside access to its hosts except for special cases such as mail servers or information servers.

This brings to the fore an access policy that firewalls are particularly adept at enforcing: do not provide access to hosts or services that do not require access. Put differently, why provide access to hosts and services that could be exploited by attackers when the access is not used or required? If, for example, a user requires little or no network access to her desktop workstation, then a firewall can enforce this policy.

Concentrated Security

A firewall can actually be less expensive for an organization in that all or most modified software and additional security software could be located on the firewall systems as opposed to being distributed on many hosts. In particular, one-time password systems and other add-on authentication software could be located at the firewall as opposed to each system that needed to be accessed from the Internet.

Other solutions to network security such as Kerberos involve modifications at each host system. While Kerberos and other techniques should be considered for their advantages and may be more appropriate than firewalls in certain situations, firewalls tend to be simpler to implement in that only the firewall need run specialized software.

Enhanced Privacy

Privacy is of great concern to certain sites, since what would normally be considered innocuous information might actually contain clues that would be useful to an attacker. Using a firewall, some sites wish to block services such as finger and Domain Name Service. Finger displays information about users such as their last login time, whether they've read mail, and other items. But, finger could leak information to attackers about how often a system is used, whether the system has active users connected, and whether the system could be attacked without drawing attention.

Firewalls can also be used to block DNS information about site systems, thus the names and IP addresses of site systems would not be available to Internet hosts. Some sites feel that by blocking this information, they are hiding information that would otherwise be useful to attackers.

Logging and Statistics on Network Use, Misuse

If all access to and from the Internet passes through a firewall, the firewall can log accesses and provide valuable statistics about network usage. A firewall, with appropriate alarms that sound when suspicious activity occurs can also provide details on whether the firewall and network are being probed or attacked.

It is important to collect network usage statistics and evidence of probing for a number of reasons. Of primary importance is knowing whether the firewall is withstanding probes and attacks, and determining whether the controls on the firewall are adequate. Network usage statistics are also important as input into network requirements studies and risk analysis activities.

Policy Enforcement

Lastly, but perhaps most importantly, a firewall provides the means for implementing and enforcing a network access policy. In effect, a firewall provides access control to users and services. Thus, a network access policy can be enforced by a firewall, whereas without a firewall, such a policy depends entirely on the cooperation of users. A site may be able to depend on its own users for their cooperation; however it cannot nor should not depend on Internet users in general.

What a Firewall Cannot Do

It is important to realize that a firewall is a tool for enforcing a security policy. If all access between trusted and untrusted networks is not mediated by the firewall, or the firewall is enforcing an ineffective policy, the firewall is not going to provide any protection for your network.

However, even a properly designed network with a properly configured firewall cannot protect the user from the following dangers.

• *Malicious use of authorized services:* A firewall cannot, for instance, prevent someone from using an authenticated Telnet session to compromise your internal machines or from tunneling an unauthorized protocol through another authorized protocol.

• *Users not going through the firewall:* A firewall can only restrict connections that go through it. It cannot protect the user from people who can go around the firewall, for example, through a dial-up server behind the firewall. It also cannot prevent an internal intruder from hacking an internal system. To detect and thwart these kinds of threats, the user may need a properly configured intrusion detection/prevention system.

• *Social engineering:* If intruders can somehow obtain passwords they are not authorized to have or otherwise compromise authentication mechanisms through social engineering mechanisms, the firewall won't stop them. For example, a hacker could call your users pretending to be a system administrator and ask them for their passwords to "fix some problem."

• *Flaws in the host operating system:* A firewall is only as secure as the operating system on which it is installed. There are many flaws present in operating systems that a firewall cannot protect against. This is why it is important to properly secure the operating system and apply the necessary security patches before installing the firewall and on a periodic basis thereafter. It also explains why "appliance" firewalls such as those provided by Nokia and NetScreen, which contain a purpose-built, hardened operating system, are becoming more popular.

• *All threats that may occur:* Firewall designers often react to problems discovered by hackers, who are usually at least one step ahead of the firewall manufacturers.

1.2 ABOUT THE PROJECT

The Internet has made large amounts of information available to the average computer user at home, in business and in education. For many people, having access to this information is no longer just an advantage, it is essential. Yet connecting a private network to the Internet can expose critical or confidential data to malicious attack from anywhere in the world. Users who connect their computers to the Internet must be aware of these dangers, their implications and how to protect their data and their critical systems.

Firewalls can protect both individual computers and corporate networks from hostile intrusion from the Internet, but must be understood to be used correctly.

This software is to detect inappropriate, incorrect, anomalous, or suspicious activity on computers or computer networks. Besides firewall, integrity verifiers, virus scanners and log-file monitors are key element when it comes to securing computers and networks.

It detects intrusions based on collected data. There are two ways to collect data: Network based data collection and host based data collection.

In general, these systems respond actively or passively to detected attacks. Active responses block or otherwise affect the progress of the attack. Passive response simply report or record the incident. It should always log detection results, regardless of whether or not active responses are enabled.

A firewall protects networked computers from intentional hostile intrusion that could compromise confidentiality or result in data corruption or denial of service. It must have at least two network interfaces, one for the network it is intended to protect, and one for the network it is exposed to. A firewall sits at the junction point or gateway between the two networks, usually a private network and a public network such as the Internet. The earliest firewalls were simply routers. The term firewall comes from the fact that by segmenting a network into different physical sub networks, they limited the damage that could spread from one subnet to another just like fire doors or firewalls.

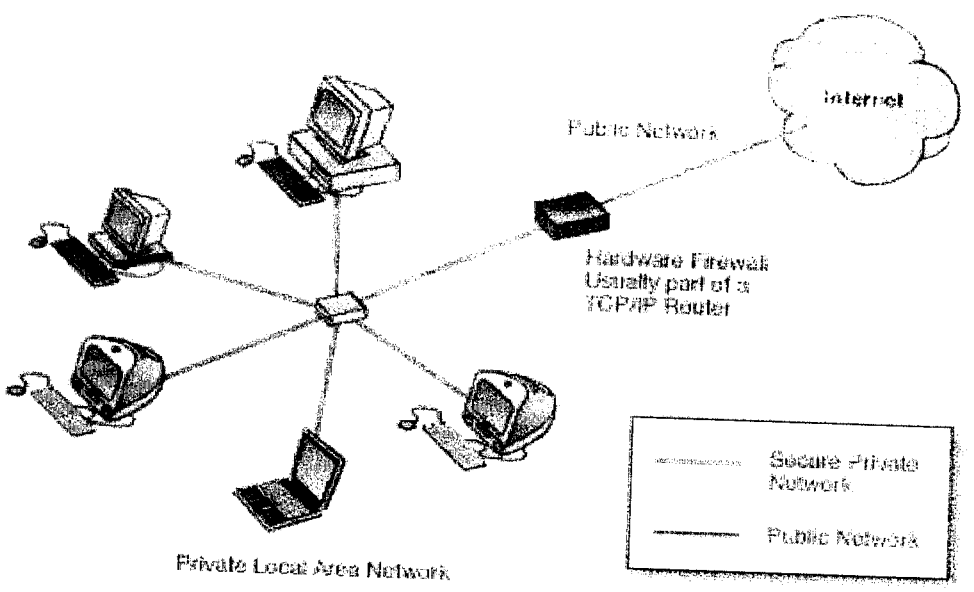


FIGURE 1.1 Networks with Firewall

Working style

A firewall examines all traffic routed between the two networks to see if it meets certain criteria. If it does, it is routed between the networks, otherwise it is stopped. A firewall filters both inbound and outbound traffic. It can also manage public access to private networked resources such as host

applications. It can be used to log all attempts to enter the private network and trigger alarms when hostile or unauthorized entry is attempted. Firewalls can filter packets based on their source and destination addresses and port numbers. This is known as address filtering. Firewalls can also filter specific types of network traffic. This is also known as protocol filtering because the decision to forward or reject traffic is dependant upon the protocol used, for example HTTP, FTP. Firewalls can also filter traffic by packet attribute or state.

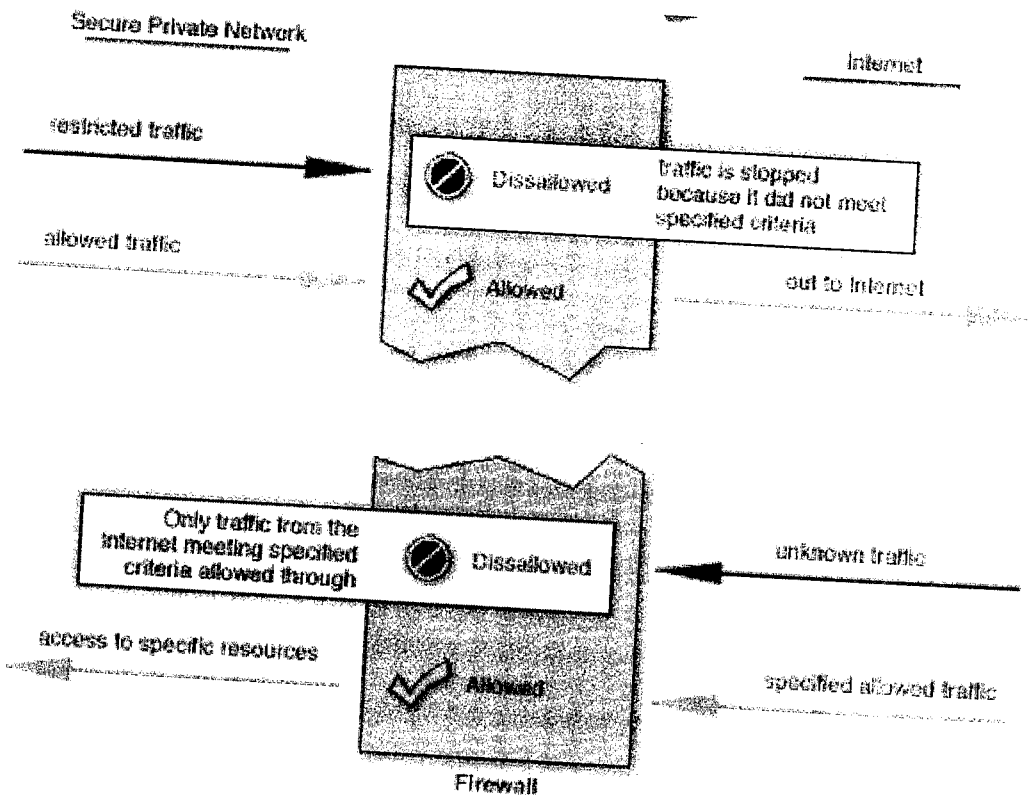


FIGURE 1.2 Working style with firewall

1.3 BENEFITS OF A FIREWALL

Firewalls protect private local area networks from hostile intrusion from the Internet. Consequently, many LANs are now connected to the Internet where Internet connectivity would otherwise have been too great a

risk. Firewalls allow network administrators to offer access to specific types of Internet services. This selectivity is an essential part of any information management program, and involves not only protecting private information assets, but also knowing who has access to what. Privileges can be granted according to job description and need rather than on an all-or-nothing basis.

1.4 INTRODUCTION TO JAVA

Java is a technology that makes it easy to build distributed application over the network. Java allows the user to do the following

- Writing robust and reliable programs.
- Build an application on almost any platform and run the application on any platform without recompiling the code.
- Distribute your application over the network in a secured fashion.

Features

There are six features in Java that makes Java as a power tool for Internet applications :

- Security
- The core API
- Open Standard
- Distributed and Dynamic
- Object oriented
- Multithreaded

Security:

Security is one of the main problems for Internet application developers. So the users must be afraid of two things. One is that the confidential information will be compromised and the next is that their computer systems are vulnerable to corruption or destruction by hackers. Java's security model has three primary components

i) Class Loader - The class loader retrieves classes from the network and it separates the server class from the local class.

ii) Byte Code Verifier - The Byte code verifier ensures that the Java programs have been compiled correctly.

iii) Security Manager - The Security manager implements a security policy for the JVM.

The Core API:

API stands for Application Programming Interface. The core API provides a common set of functions on all platforms. The API is divided into packages, which are groups of classes that perform related functions.

Open Standard:

The most exciting aspect of Java's cross platform capability is that Java class file does not need to be compiled for each platform in advance. The same compiled Java program will work on the PC and every platform that runs JAVA. JAVA application can be written on the system and it can be run on every supported platform.

Distributed and Dynamic:

In Windows operating system parts of programs can be placed into DLL (Dynamic Link Library) so that they can be shared and loaded dynamically. The JVM class loader fetches class files from the network, as well as from the disk, making Java application distributed as well as dynamic.

Object Oriented:

Object oriented programming is a way to write software that is reusable, extensible and maintainable. Java is an object oriented programming language.

Multithreaded:

Java was designed to meet the real-world requirements of creating interactive, networked programs. To accomplish this, Java supports multithreaded programming. A multithreaded application can have several threads of execution running independently and simultaneously. These threads may communicate and co-operate, and to the user will appear to be a single program with the following functions

- Maintain user interface responsively
- Multitasking
- Multi-user applications
- Multiprocessing

1.4.1 NETWORKING BASICS

Computers running on the Internet communicate with each other using either the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP), as this diagram illustrates:

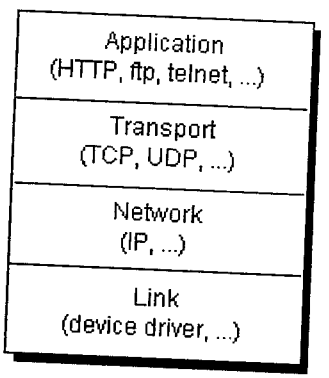


FIGURE 1.3 Various layers and Protocols

While writing Java programs that communicate over the network, programming is done at the application layer. Typically, there is no need to be concerned with the TCP and UDP layers. Instead, the classes in the java.net package can be used. These classes provide system-independent network communication. However, to decide which Java classes the programs should use, there is a need to understand how TCP and UDP differ.

TRANSMISSION CONTROL PROTOCOL

When two applications want to communicate with each other reliably, they establish a connection and send data back and forth over that connection. TCP provides a point-to-point channel for applications that require reliable communications. The Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP) and Telnet are all examples of applications that require a reliable communication channel. The order in which the data is sent

and received over the network is critical to the success of these applications. When HTTP is used to read from a URL, the data must be received in the order in which it was sent. Otherwise, we end up with a jumbled HTML file, a corrupt zip file, or some other invalid information.

Definition: TCP (Transmission Control Protocol) is a connection-based protocol that provides a reliable flow of data between two computers.

USER DATAGRAM PROTOCOL

The UDP protocol provides for communication that is not guaranteed between two applications on the network. UDP is not connection-based like TCP. Rather, it sends independent packets of data, called datagrams together from one application to another. Sending datagrams is much like sending a letter through the postal service: The order of delivery is not important and is not guaranteed, and each message is independent of any other.

Definition: UDP (User Datagram Protocol) is a protocol that sends independent packets of data, called datagrams, from one computer to another with no guarantee about arrival. UDP is not connection-based like TCP.

UNDERSTANDING PORTS

Generally speaking, a computer has a single physical connection to the network. All data destined for a particular computer arrives through that connection. However, the data may be intended for different applications running on the computer. Data transmitted over the Internet is accompanied by addressing information that identifies the computer and the port for which it is destined. The computer is identified by its 32-bit IP address, which is used to deliver data to the right computer on the network. Ports are identified by a 16-bit number through which TCP and UDP use to deliver the data to

the right application. In connection-based communication such as TCP, a server application binds a socket to a specific port number. This has the effect of registering the server with the system to receive all data destined for that port. A client can then rendezvous with the server at the server's port, as illustrated here:

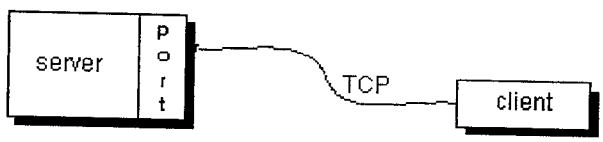


FIGURE 1.4 Client rendezvous at server's port

Definition: The TCP and UDP protocols use ports to map incoming data to a particular process running on a computer.

NETWORKING CLASSES IN THE JDK

Through the classes in java.net, Java programs can use TCP or UDP to communicate over the Internet. The URL, URLConnection, Socket and ServerSocket classes all use TCP to communicate over the network. The DatagramPacket, DatagramSocket and MulticastSocket classes are for use with UDP.

1.4.2 SOCKETS

Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

On the client-side: The client knows the hostname of the machine on which the server is running and the port number to which the server is connected. To make a connection request, the client tries to rendezvous with the server on the server's machine and port.

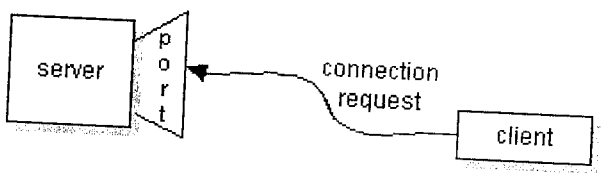


FIGURE 1.5 Client Connection request

If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to a different port. It needs a new socket (and consequently a different port number) so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client.

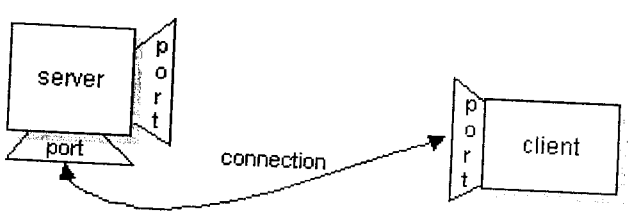


FIGURE 1.6 Server accepting connection

On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with

the server. Note that the socket on the client side is not bound to the port number used to rendezvous with the server. Rather, the client is assigned a port number local to the machine on which the client is running. The client and server can now communicate by writing to or reading from their sockets.

Definition: A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent.

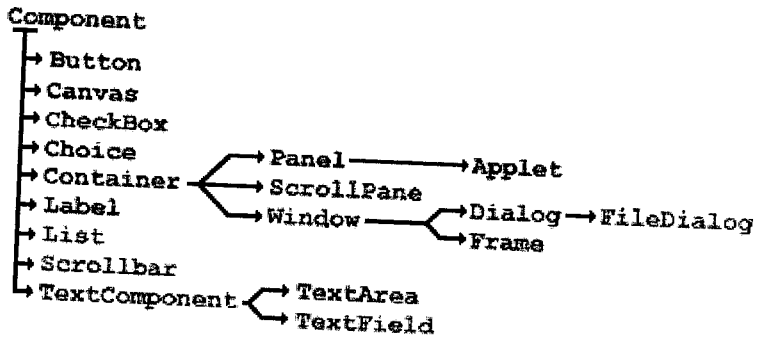
The java.net package in the Java platform provides a class, Socket, that implements one side of a two-way connection between a Java program and another program on the network. The Socket class sits on top of a platform-dependent implementation, hiding the details of any particular system from the Java program. By using the java.net.Socket class instead of relying on native code, Java programs can communicate over the network in a platform-independent fashion. Additionally, java.net includes the ServerSocket class, which implements a socket that servers can use to listen for and accept connections to clients.

1.4.3 SWING OVERVIEW

ABSTRACT WINDOW TOOLKIT

AWT (the Abstract Window Toolkit) is the part of Java designed for creating user interfaces and painting graphics and images. It is a set of classes intended to provide everything a developer requires in order to create a graphical interface for any Java applet or application.

Most AWT components are derived from the java.awt.Component classes



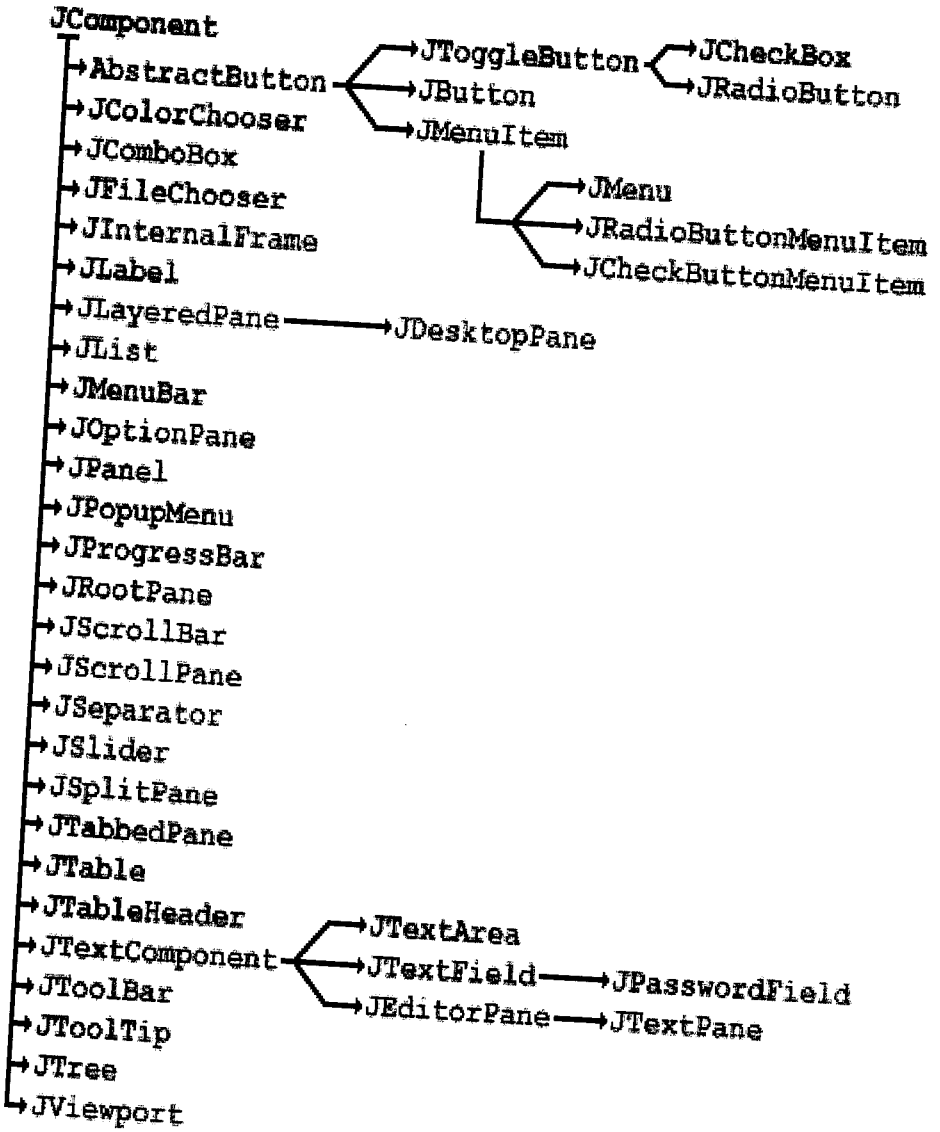
The Java Foundation Classes consist of five major parts: AWT, Swing, Accessibility, Java 2D and Drag and Drop. Java 2D has become an integral part of AWT Swing is built on top of AWT and accessibility support is built into Swing.

JFC SWING

Swing is a large set of components ranging from the very simple, such as labels, to the very complex, such as tables, trees, and styled text documents. Almost all Swing components are derived from a single parent called JComponent which extends the AWT Container class. Thus, Swing is best described as a layer on top of AWT rather than a replacement for it. The diagram below shows a partial JComponent hierarchy. If this is compared with the AWT Component hierarchy, it can be noticed that for each AWT component there is a Swing equivalent with prefix "J". The only exception to this is the AWT Canvas class, for which JComponent, JLabel or JPanel can be used as a replacement.

The below diagram represents only a small fraction of the Swing library, but this fraction consists of the classes user will be dealing with

most. The rest of Swing exists to provide extensive support and customization capabilities for the components these classes define.



Swing components are referred to as lightweights while AWT components are referred to as heavyweights. The difference between lightweight and heavyweight components is z-order: the notion of depth or layering. Each heavyweight component occupies its own z-order layer. All lightweight components are contained inside heavyweight components and maintain their own layering scheme defined by Swing. When a heavyweight

is placed inside another heavyweight container it will, by definition, overlap all lightweights in that container.

What this ultimately means is that using both heavyweight and lightweight components in the same container should be avoided whenever possible. This does not mean that AWT and Swing components can never be mixed successfully. It just means that which situations are safe and which are not should be known carefully. Since it is not possible to completely eliminate the use of heavyweight components anytime soon, ways to make the two technologies work together in an acceptable way have to be found.

The most important rule to follow is that heavyweight components should never be placed inside lightweight containers that commonly support overlapping children. Some examples of these containers are JInternalFrame, JScrollPane, JLayeredPane and JDesktopPane. Secondly, if popup menu is used in a container holding a heavyweight component, there is a need to force that popup to be heavyweight. To control this for a specific JPopupMenu instance , its setLightWeightPopupEnabled () method can be used.

PLATFORM INDEPENDENCE

The most remarkable thing about Swing components is that they are written in 100% Java and do not depend on peer components as most AWT components do. This means that a Swing button or text area will look and function identically on Macintosh, Solaris, Linux and Windows platforms. This design eliminates the need to test and debug applications on each target platform.

SWING PACKAGE OVERVIEW

javax.swing

Contains the most basic Swing components, default component models and interfaces.

javax.swing.border

Classes and interfaces used to define specific border styles. Note that borders can be shared by any number of Swing components as they are not components themselves.

javax.swing.colorchooser

Classes and interfaces supporting the JColorChooser component are used for color selection.

javax.swing.event

The event package contains all Swing-specific event types and listeners. Swing components also support events and listeners defined in java.awt.event and java.beans.

javax.swing.filechooser

Classes and interfaces supporting the JFileChooser component are used for file selection.

javax.swing.plaf

Contains the pluggable look-and-feel API used to define custom user interface components. Most of the classes in this package are abstract. They are subclassed and implemented by look-and-feel implementations such as

metal, motif and basic. The classes in this package are intended for use only by developers who, for one reason or another, cannot build on top of existing look-and-feels.

javax.swing.plaf.basic

Consists of the Basic look-and-feel implementation which all look-and-feels provided with Swing are built on top of. We are normally expected to subclass the classes in this package if we want to create our own customized look-and-feel.

javax.swing.plaf.multi

This is the Multiplexing look-and-feel. This is not a regular look-and-feel implementation in that it does not define the actual look or feel of any component. Rather, it provides the ability to combine several look-and-feels for simultaneous use. A typical example might be using an audio-based look-and-feel in combination with metal or motif. Currently Java 2 does not ship with any multiplexing look-and-feel implementation.

javax.swing.table

Classes and interfaces supporting the JTable control. This component is used to manage tabular data in spreadsheet form. It supports a high degree of customization without requiring look-and-feel enhancements.

javax.swing.text

This contains Classes and interfaces used by the text components including support for plain and styled documents, the views of those documents, highlighting, caret control and customization, editor actions and keyboard customization.

javax.swing.text.html

This is an extension of the text package which contains support for HTML text components.

javax.swing.text.html.parser

This provides Support for parsing HTML.

javax.swing.text.rtf

Contains support for RTF documents.

javax.swing.tree

Classes and interfaces supporting the JTree component. This component is used for the display and management of hierarchical data. It supports a high degree of customization without requiring look-and-feel enhancements.

javax.swing.undo

The undo package contains support for implementing and managing undo/redo functionality.

PROBLEM FORMULATION

CHAPTER 2

PROBLEM FORMULATION

2.1. MAIN OBJECTIVE

Information and communication are two of the most important strategic issues for the success of an enterprise. While today nearly every organization uses a substantial number of computers and communication tools (telephones, fax and personal handheld devices), they are often still isolated. While managers today are able to use the newest applications, many departments still do not communicate and much needed information cannot be readily accessed.

To overcome these obstacles with an effective usage of information technology, computer networks are necessary. They are a new kind (one might call it paradigm) of organization of computer systems produced by the need to merge computers and communications. At the same time they are the means to converge the two areas; the unnecessary distinction between tools to process and store information and tools to collect and transport information can disappear. Computer networks can manage to put down the barriers between information held on several (not only computer) systems. Only with the help of computer networks can a borderless communication and information environment be built.

Computer networks allow the user to access remote programs and remote databases either of the same organization or from other enterprises or public sources. Computer networks provide communication possibilities faster than other facilities. Because of these optimal information and communication possibilities, computer networks may increase the

organizational learning rate, which many authors declare as the only fundamental advantage in competition.

Besides this major reason why any organization should not fail to have a computer network, there are other reasons as well: cost reduction by sharing hardware and software resources, high reliability by having multiple sources of supply, cost reduction by downsizing to microcomputer-based networks instead of using mainframes and greater flexibility because of possibility to connect devices from various vendors

Because of the importance of this technology, decisions of purchase, structure and operation of computer networks cannot be left to technical staff. Management as well has a critical need for understanding the technology of computer networks.

The internet is one kind of network that has made large amounts of information available to the average computer user at home, in business and in education. For many people, having access to this information is no longer just an advantage, it is essential. Yet connecting a private network to the Internet can expose critical or confidential data to malicious attack from anywhere in the world. Users who connect their computers to the Internet must be aware of these dangers, their implications and how to protect their data and their critical systems. Firewalls can protect both individual computers and corporate networks from hostile intrusion from the Internetbut must be understood to be used correctly.

2.2 INTERNET SECURITY

As of August 9th, 2003, the Internet is connecting an estimated 820 million computers in 260 countries on every continent, including Antarctica. The Internet is not a single network, but a vast array of loosely connected

networks situated all over the world, easily accessible by individual computer hosts in a variety of ways. Today, the Internet uses gateways, routers, dial-up connections and Internet service providers (ISPs) to make itself readily available at all times. Individuals and organizations worldwide can reach any point on the network without regard to national or geographic boundaries or time of day.

The great importance of Internet security

However, while using the Internet, along with the convenience and speed of access to information we have new risks. Among them are the risks that valuable information will be lost, stolen, corrupted or misused and that the computer systems will be corrupted. If information is recorded electronically and is available on networked computers, it is more vulnerable than if the same information is printed on paper and locked in a file cabinet. Intruders do not need to enter an office or home and may not even be in the same country. They can steal or tamper with information without touching a piece of paper or a photocopier. They can create new electronic files, run their own programs and even hide all evidence of their unauthorized activity.

The three basic security concepts important to information on the Internet are:

- Confidentiality
- Integrity
- Availability

Concepts related to people using this information are authentication, authorization and non repudiation. When information is read or copied by someone not authorized to do so, the result is known as loss of confidentiality. For some types of information, confidentiality is a very

important attribute. Examples include research data, medical and insurance records, new product specifications and corporate investment strategies. In some locations, there may be a legal obligation to protect the privacy of individuals. This is particularly true for most banks and loan companies, debt collecting agencies, businesses that offer credit to their customers or issue credit cards, hospitals, doctor's offices and medical testing laboratories, individuals or agencies that offer services such as psychological counseling or drug treatment and agencies that collect any form of taxes.

Information can be corrupted when it is available on an insecure network. When information is modified in unexpected ways, the result is known as loss of integrity. This means that unauthorized changes are made to information, whether by human error or intentional tampering. Integrity is particularly important for critical safety and financial data used for activities such as electronic funds transfers, air traffic control and financial accounting.

Information can be erased or become inaccessible, resulting in loss of availability. This means that people who are authorized to get information cannot get what they need. Availability is often the most important attribute in service-oriented businesses that depend on information (e.g., airline schedules and online inventory systems). Availability of the network itself is important to anyone whose business or education relies on a network connection. When a user cannot get access to the network or specific services provided on the network, they experience a denial of service.

To make information available to those who need it and who can be trusted with it, organizations use authentication and authorization. Authentication is proving that a user is whom he or she claims to be. That proof may involve something the user knows (such as a password), something the user has (such as a "smartcard"), or something about the user

that proves the person's identity (such as a fingerprint). Authorization is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity, such as reading a file or running a program. Authentication and authorization go hand in hand. Users must be authenticated before carrying out the activity they are authorized to perform. Security is strong when the means of authentication cannot later be refuted - the user cannot later deny that he or she performed the activity. This is known as non repudiation.

2.3 SPECIFIC OBJECTIVE

The Situation

More and more companies are using the Internet for their business needs, with electronic mail, file transfer processing (FTP), offers and orders (WWW). Additionally, market globalization, development distribution and production at various world-wide sites increasingly require that sensitive data be transferred across the Internet or Extranet.

The Challenge

The challenge here is to make the internal network as transparent as possible for employees and yet still protect sensitive data on its way across the Internet. Internal networks have to be protected against unauthorized access from inside and outside without impairing access functionality to internal and external services. Data connections on the Internet have to be protected to prevent confidential company information from being tapped.

The Concept

The firewall security tools have been using their own mail, file and web servers (http, ftp protocols) along with secure gateways between Internet and Intranet for many years.

The Solution

First step includes a definition of the structure and the security scope of firewall. An analysis of the communication needs throughout business processes can detect possible risks. IT Consultation section can reliably and efficiently help to evaluate any risks for individual communication types. A wide range of alternatives is available for hardware and software specific implementation of firewall in a rapidly changing technical environment.

2.4 SYSTEM REQUIREMENTS

2.4.1 Hardware Requirements

- Pentium Processors or better
- 32 Mb of RAM
- Up to 15 Mb of disk space available

2.4.2 Software Requirements

- Operating Systems: Microsoft Windows 98, NT4 SP3, 2000, Me or XP
- Java 1.3 or 1.4 or Jcreator that runs as a service under Windows NT, 2000 and XP

SYSTEM ANALYSIS

CHAPTER 3

SYSTEM ANALYSIS

Analysis is the first phase of system development life cycle. System analysis involves a substantial amount of effort and cost and is therefore undertaken only after a management has decided that the systems development project under consideration has merit and should be pursued through this phase.

The project initiation and planning phase provides the basis for go-head decision for analysis and the Baseline Project Plan (BPP) it is the structure for conducting the analysis phase.

Analysis is a large and involved process, which can be divided into three main activities to make the overall process easy to understand.

Analysis includes the following three steps:

- Requirements determination - This is primarily a fact-finding activity.
- Requirements structuring - This activity creates a thorough and clear description of current business operations and new information processing services. It has three sub activities that concentrate on structuring different views or dimensions of the information system.

- Alternative generation and selection - This process results in a choice among alternative strategies for subsequent systems design.

The purpose of the analysis is to determine what information and information processing services are needed to support the selected objectives and functions of the organization; consequently, analysis is fundamentally an intelligent activity in which analysts capture and structure information. Gathering the information about both the current and replacement system is called requirements determination.

During the requirement determination process, the systems development team attempts to discover important information about how employees now perform and how they will need to perform their jobs in order to meet future business conditions. Included in this information will be answers to the following questions:

- How does the current system function? Is the system manual or automated?
- What data are necessary for proper functioning of the supported business area?
- What kinds of reports are generated?
- How do people use the system to perform their work?

A study of current operations gives insights on stream requirements, grounds analysis in actual information system activity, is the basis for discovering, possible incremental improvements, and provides necessary information for subsequent steps involved in converting from the current to a replacement system.

Analysts must focus on the following additional questions:

- How should a new and replace system function?
- What data is needed to operate smoothly?
- What kind of reports would it need to generate?

- What new or improved information services are needed to support the future organizational goals, objectives, strategies and functions?

Information about current operations and requirements for a replacement must somehow be organized in order to be useful during analysis and subsequently to a design team.

Organizing or structuring system requirements is the second major activity of analysis phase. The results of the requirements determination phase can be structured according to three essential views of the current and replacement information systems.

- Process - the sequence of data movement and handling operations within the system
- Logic and timing - the rules by which data movement are transformed and an indication of what triggers data transformation
- Data – the inherent structure of data independent of how or when it is processed

The goal is to capture as complete a specification of the required system as possible. Ideally, the specification would be so complete that, along with the necessary steps in the design phase, the replacement or new system could be generated without human programming. The structured specifications provide a clear picture for later design, implementation, and most importantly, maintenance activities. Maintaining system with functional specifications is more reliable and significantly more productive than manually modifying code as the inevitable system changes occur.

The process view of a system can be represented by data flow diagrams. System logic and timing or what goes on inside the black boxes

of that are identified as processes in data flow diagrams and when these processes occur, can be represented in many ways, including the Structured English, Decision tables and decision trees. Finally Data view of the system shows the rules that govern the structure and integrity of data and concentrates on what data about business entities and relationships among these entities must be accessed within the system.

3.1 DETERMINING SYSTEM REQUIREMENTS

In many ways, gathering system requirements is like conducting any investigation. Like solving the puzzles, we can detect some similar characteristics for good systems analysts during the requirements sub phase. These characteristics include

- Impertinence which means questioning for every thing.
- Impartiality which means finding the best solution.
- Relax constraints means assuming that anything is possible and eliminating the infeasible.
- Attention details means every fact must fit with every other fact.
- Reframing means analysis is, in part, a creative process. Look at the organization in new ways. Must consider each user's views and requirements.

3.2 STRUCTURING SYSTEM REQUIREMENTS

Process Modeling

Process modeling involves graphically representing the functions or processes, with capture, manipulate, store and distribute data between a system and its environment and between components within the system. A common form of a process model is a **data flow diagram.**

Data Flow Diagram

Data flow diagrams are versatile diagramming tools. With only four symbols we can represent both the physical and logical information systems. There are two different standard sets of data flow diagram symbols, but each consists of four symbols that represent the same things: data flows, data store, processes and sources / sinks. A data flow can be best understood as data in motion.

SYSTEM DESIGN

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM OVERVIEW

The system is developed to implement the firewall security for a private network, using Java. Basically the system concentrates on the services offered by HTTP and FTP. The HTTP and FTP are the application level protocols which can offer various services over internet and it is very difficult to control and protect. In this system an integrated approach is designed using simple HTTP server and FTP server and providing the firewall security for private networks from HTTP and FTP services. The services are filtered out using the Application level filtering.

Application level Filtering provides access control at the application level layer. It can act as a Gateway between the two networks, but it will not be transparent to the users. The traffic can be filtered based on the IP address from where the request can come.

In this system a Firewall Manager is developed, which can be integrated with HTTP manager and FTP manager. The HTTP manager can manage the Authorized and Unauthorized users who can access the HTTP server. The FTP manager can be designed to manage the authorized and unauthorized users who can access the FTP server and additionally the system has FTP manager with the functionality of adding and removing the files to and from the FTP server.

In this system the firewall security can be provided for only the developed Web Server and File Server from the HTTP and FTP services and the system cannot provide security for any third party servers.

The simple HTTP server is developed using Java and FTP server. Using servlets both can listen for the request from the remote system and provide the service to the remote system if it can be authorized by the firewall.

The database is maintained which is developed in MS-ACCESS which can be used to store the authorized users for the HTTP server and FTP server as well as the files that can be put in the File server.

4.2 MODULES SPECIFICATION

- MANAGER
- HTTP SERVER
- FTP SERVER
- FILTERING

    1. HTTP FILTERING
    2. FTP FILTERING

4.2.1 MANAGER:

Manager is an user interface which can be developed using the JFC Swing in Java. The Manager is developed with the help of various lightweight components available in Swing.

It can be developed in the user friendly way where it can be fully enabled by the hotkeys and the HTTP manager and FTP manager are integrated into the Manager.

There will be a Frame which can be developed using JFrame component in the Swing where the Menus can be placed over the top of the Frame which can be developed using JMenu component and the menu items can be developed

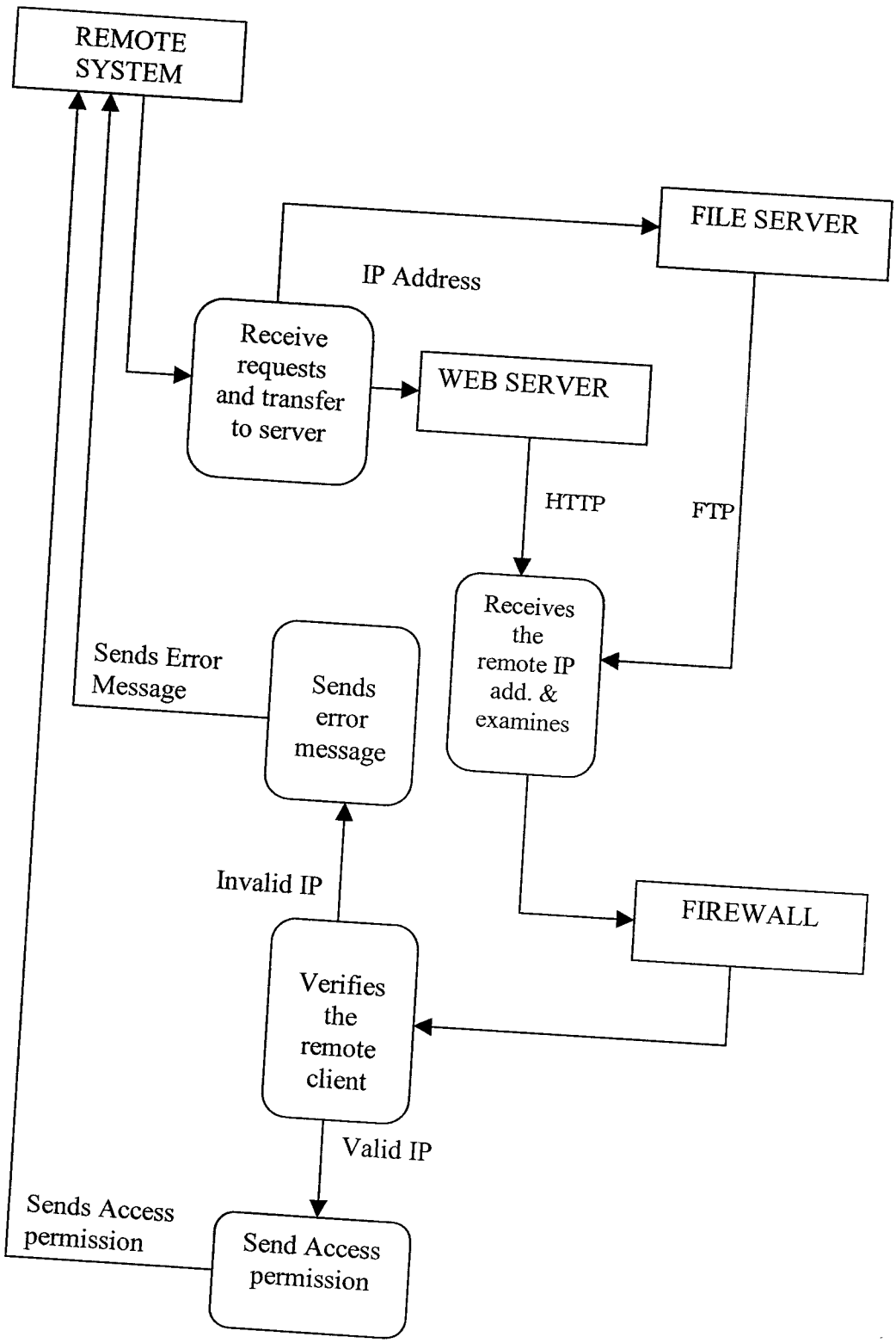


FIGURE 4.1 System Data Flow Diagram

using the JMenuItem and when the user clicks or accesses the menus using the hotkeys the appropriate event can be performed.

There will be Frame which can be developed using JFrame component in the swing where the Menus can be placed over the top of the Frame which can be developed using JMenu component and the menu items can be developed using the JMenuItem and when the user clicks or accesses the menus using the hotkeys the appropriate event can be performed.

The Manager will be designed with the separate Internal Frames using JInternalFrame where it can be used for displaying the Frames for the separate process. The Frame can be designed with the other lightweight components such as buttons, text fields and combo boxes by using the appropriate component.

The manager can be designed with the various components for those the ActionPerformed event. The message boxes displayed for the Manager window are designed separately and can be displayed correctly according to the event performed.

The system launches the Manager by running the "manager.bat" batch file and it can display the Manager window integrated with the HTTP and FTP services.

4.2.2 HTTP SERVER

The HTTP server can be designed using Java that can be configured on the port 6000 because the ports 0-1024 are registered for the predefined services. It can be designed with the minimum requirement properties such as timeout and workers property. Whenever the Http server starts running, it can be loaded with the properties such as displaying the root directory where

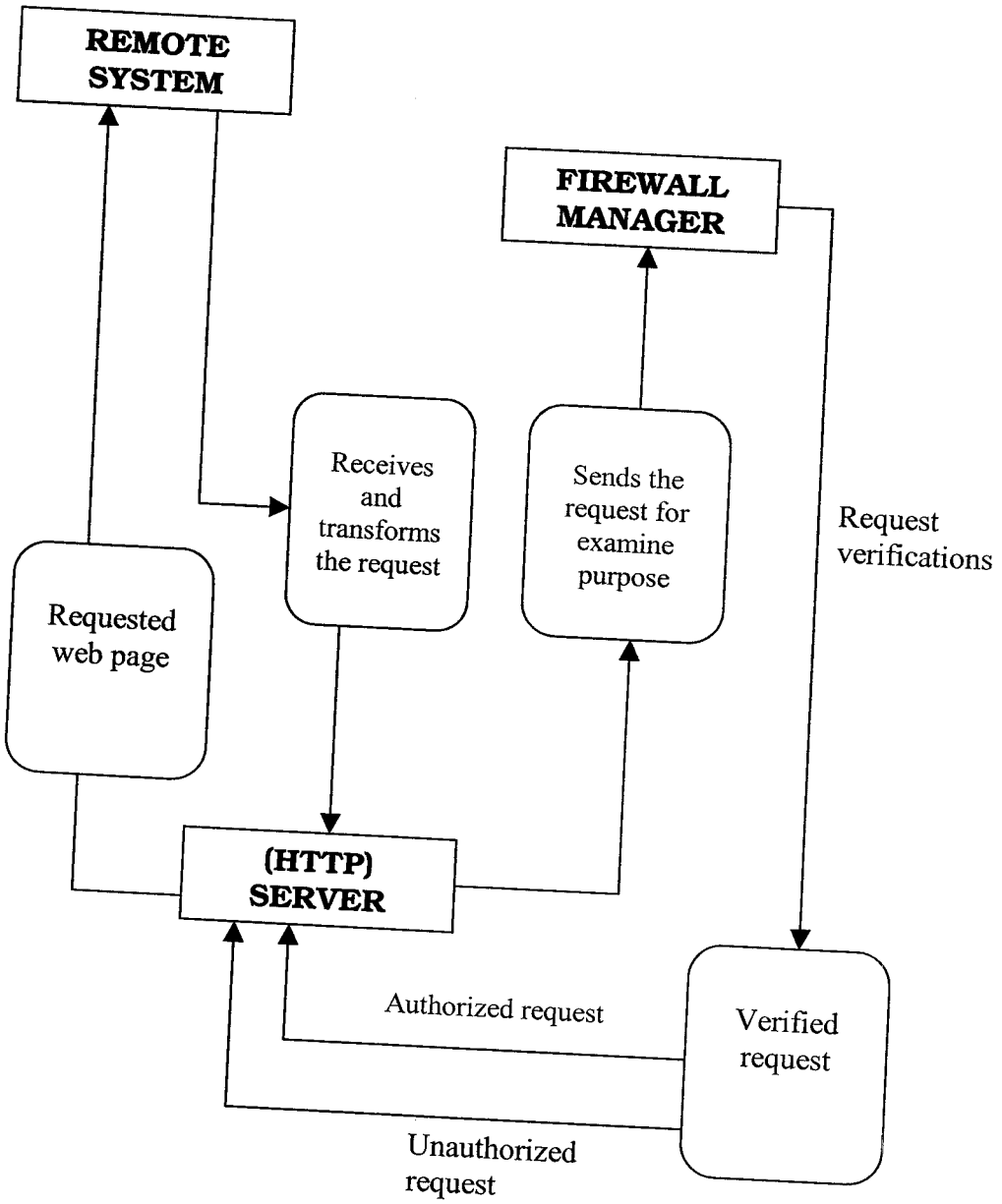


FIGURE 4.2 Data Flow Diagram for HTTP

it can be run and the total timeout and the workers request to the server.

The server is designed to use the HTTP GET and HEAD methods. The system takes advantage by using the FileURLConnection that will parse the directory to list the content in HTML for us. The system can also add some of the file extensions that can map to the content type that the user can request. Finally the system has HTTP Constants for all the 2xx, 3xx, 4xx,

5xx which are for the generally ok, relocation/redirect, client error, and server error.

The system launches server by running the httpstart.bat batch file. When the server is running it can listen to the port and it can be configured for the request from any remote system. When the request is received from any remote system, the Firewall Manager can verifies the IP of the remote system by extracting it from the packet and checking it for authorized users to access the server. These things are implemented with the help of sockets.

4.2.3 FTP SERVER

The FTP server can be designed using Java Servlets which can be configured on the port 8080 which is the default port for the Servlets. The server can be designed to give the response on the content type based on text and HTML. The server can connect to the database from where it can fetch the files that can be uploaded to the FTP server. Then it can get the request from the remote system from it can extract the remote IP and it can check it against the FTP database to know whether it is an authorized user. If ok, then the server will fetch the files from the database of the file server and display the files to the remote user and he will be permitted to access the file for viewing and transferring. If the requested IP is not authorized then it can send back the error massage to the remote system.

The system launches the FTP server by running the startserver.bat batch file and by running it, the configuration files for the server can be loaded and the server is ready to run. After all the operations are over the server can be stopped by running the "stopserver.bat" batch file.

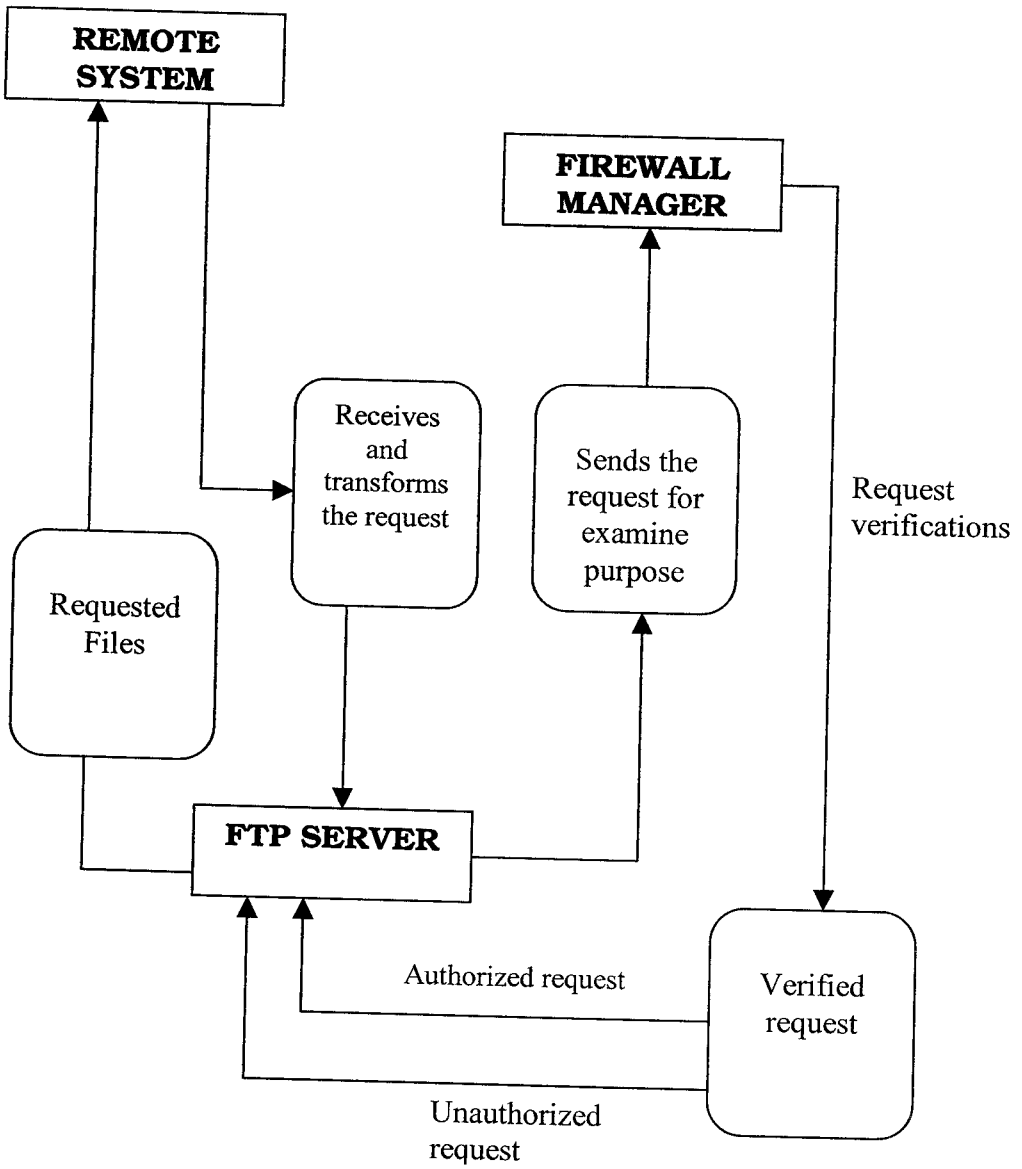


FIGURE 4.3 Data Flow Diagram for FTP

4.2.4 FILTERING

HTTP FILTERING AND FTP FILTERING

The HTTP filtering and FTP filtering can be done using Application level filtering where the requested IP can be examined by the Firewall Manger and if the requested IP is an authorized onethen it can be permited access services provided by the servers. If it is not an authorized one, it can

filter out the request by sending the appropriate error message to the remote system.

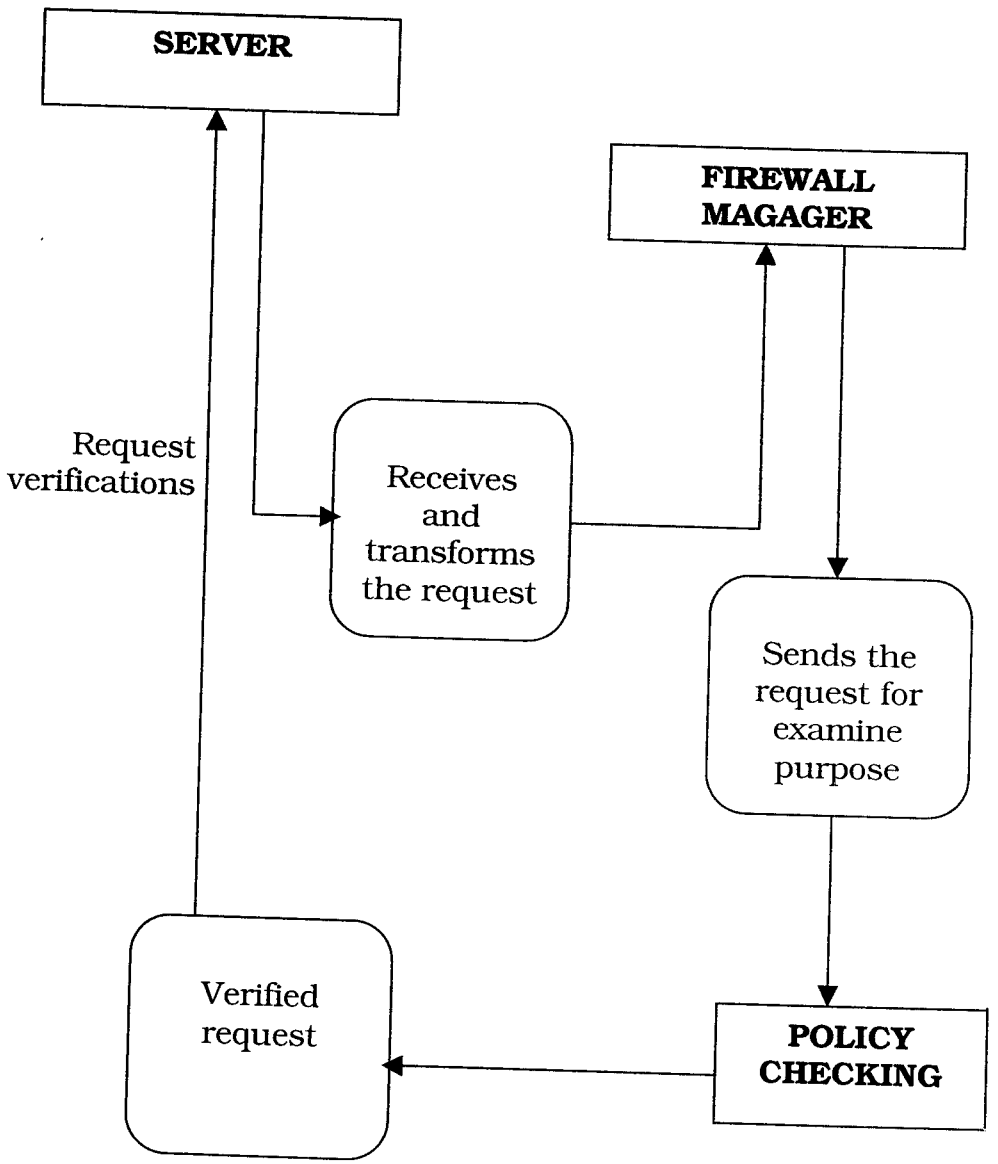


FIGURE 4.4 Data Flow Diagram for Filtering

SYSTEM IMPLEMENTATION AND TESTING

CHAPTER 5

SYSTEM IMPLMENTATION AND TESTING

System implementation is made up of many activities. They are,

- Coding
- Testing
- Installation
- Documentation
- Training
- Support

The purpose of these steps is to convert the final physical system specification into working and reliable software.

5.1 CODING

Coding is a process where by the physical design specifications created by the analysis team are turned into working computer code by the programming team. Depending upon the size and complexity of the system coding can be an involved and intensive activity.

5.2 TESTING

Once coding has begun, the testing process can be carried out and when each program module is produced. It can be tested individually, then as a part of a larger program and then as a part of larger system.

Types of tests

Software application testing is an umbrella term that covers several types of tests. Mosley organized the types of tests according to whether they

employ static or dynamic techniques and whether the test is automated or manual.

Static testing means that the code being tested is not executed. The results of running the code are not an issue for the particular test. Dynamic testing, on the other hand, involves the execution of code. Automated testing means the computer conducts the tests while manual means the people do.

Categorization of test types

	Manual	**Automated**
Static	Inspections	Syntax Checking
Dynamic	Walkthrough	Unit Test
	Desk checking	Integration Test

System test

Inspections

Inspections are formal group of activities where the participants manually examine code for error occurrences of well-known errors. Syntax, grammar and some other routine errors can be checked by automated inspection software.

Walkthrough

Walkthrough is a very effective method of detecting errors in code. It's not able to correct the errors. It's the programmer's job to correct the errors uncovered in the walkthrough.

Desk Checking

A testing is the technique in which the program code is sequentially executed manually by the reviewer.

Unit Testing

Each module is tested alone in an attempt to discover any errors in its code.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. Modules are typically integrated in a top down, incremental fashion.

System Testing

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system.

Testing purpose

- The purpose of testing is confirming that the system satisfies the requirements.
- Testing must be planned.

Acceptance testing by users

Once the system tests have been satisfactorily completed, the system is ready for **acceptance testing**, which is testing the system in the environment where it will eventually be used. Acceptance refers to the fact

that users typically sign off on the system and "accept" it once they are satisfied with it. The purpose of the acceptance testing is for users to determine whether the system meets their requirements. The extent of acceptance testing will vary with the organization and with the system in question. The most complete acceptance testing will include

- Alpha testing - where simulated but typical data are used for system testing.
- Beta testing – in which live data are used in the user's real working environment.

During alpha testing, the entire system is implemented in a test environment to discover whether or not the system is overtly destructive to itself or to the rest of the environment. The type of tests performed during alpha testing includes the following:

- Recovery testing – forces the software to fail in order to verify that recovery is properly performed.
- Security testing - verifies that protection and mechanisms built in to the system will protect it from improper penetration.
- Stress testing - tries to break the system.
- Performance testing – determines the system perform once on the range of possible environments in which it may be used.

In beta testing, a subset of the intended users runs the system on their environments using their own data.

5.3 Installation

Installation is the process during which the current system is replaced by the new system. This includes the conversion of existing data, software, documentation and work procedures to those consistent with the new system.

5.4 Maintenance

When a system is in the maintenance phase, some person within the systems development group is responsible for collecting maintenance requests from system users and other interested parties, like system auditors, data center and network management staff and data analysts.

The process of maintaining a system is the process of returning to the beginning of the system development life cycle and repeating development steps until the changes are implemented. Four major activities occur within maintenance:

- Obtaining maintenance requests
- Transforming requests into changes
- Designing changes
- Implementing changes

CONCLUSION AND FUTURE DEVELOPMENT

CHAPTER 6

CONCLUSION

The key objective of the project is to provide firewall security to the users in the network environment. System provides a complete solution for the personalization and user friendliness, a web server and an ftp server can offer. The product is portable across all Operating Systems and provides efficiency and security. The user-friendly feature that has been incorporated in the system allows any user to exploit them to get the maximum benefit. In this application user can schedule to prevent unauthorized access over HTTP and FTP.

Thus the project fulfills the above objectives and ensures security on both the server sides. The performance of the system complies with its surroundings and thus it makes the system management more versatile. The scalability of the system can also be enhanced which makes it more adaptive and easy to up-grade. The techniques applied in the design of the programs provide a scope for expansion and implementation changes, which may be required in future. All the programs have been tested with sample data and found to execute correctly.

FUTURE DEVELOPMENT

Firewall security is a product which is developed in such a way that it is compatible for all platforms in MAN and WAN environment. In future, the scalability of the system can be improved using the DMZ (demilitarized zone)environment to prevent the back door attack through the dial up modems. Then the scalability of FTP server can be increased in such a way as to provide secure accounts and scalability of the HTTP server can also increased to parse and display all types of content types.

APPENDIX

APPENDIX – 1

SOURCE CODES

CODE FOR httpftp.java :

```
import java.sql.*;
import java.awt.*;
import java.io.*;
public class HttpFtp extends javax.swing.JFrame {
  public HttpFtp() {
    initComponents();
          msgbox=new MessageBox();
          try
          {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      con = DriverManager.getConnection("jdbc:odbc:httpftp");
         stmt = con.createStatement();
          }catch(Exception ex)
          {
          this.setVisible(false);
                msgbox.setMessage("Unable To Connect With
                Database...",ex.toString(),0);
          }
}
  private void initComponents() {//GEN-BEGIN:initComponents
  jInternalFrame4 = new javax.swing.JInternalFrame();
  jLabel5 = new javax.swing.JLabel();
  jComboBox2 = new javax.swing.JComboBox();
  jButton8 = new javax.swing.JButton();
```

```
jButton9 = new javax.swing.JButton();
jInternalFrame3 = new javax.swing.JInternalFrame();
jLabel3 = new javax.swing.JLabel();
jButton5 = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jButton6 = new javax.swing.JButton();
jButton7 = new javax.swing.JButton();
jInternalFrame2 = new javax.swing.JInternalFrame();
jLabel2 = new javax.swing.JLabel();
jButton3 = new javax.swing.JButton();
jButton4 = new javax.swing.JButton();
jComboBox1 = new javax.swing.JComboBox();
jInternalFrame1 = new javax.swing.JInternalFrame();
jLabel1 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
jMenuItem2 = new javax.swing.JMenuItem();
jMenuItem3 = new javax.swing.JMenuItem();
jMenu3 = new javax.swing.JMenu();
jMenuItem4 = new javax.swing.JMenuItem();
jMenuItem5 = new javax.swing.JMenuItem();
jSeparator1 = new javax.swing.JSeparator();
jMenuItem6 = new javax.swing.JMenuItem();
```

```
jMenuItem7 = new javax.swing.JMenuItem();
jMenu4 = new javax.swing.JMenu();
jMenuItem8 = new javax.swing.JMenuItem();
jMenuItem9 = new javax.swing.JMenuItem();

getContentPane().setLayout(new
  org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("HttpFtp Manager");
addWindowListener(new java.awt.event.WindowAdapter() {
   public void windowClosing(java.awt.event.WindowEvent evt) {
      exitForm(evt);
   }
});
jInternalFrame4.getContentPane().setLayout(new
  org.netbeans.lib.awtextra.AbsoluteLayout());
jInternalFrame4.setTitle("Remove File");
jLabel5.setText("Select File");
jInternalFrame4.getContentPane().add(jLabel5, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10, -1, -1));
jComboBox2.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jComboBox2ActionPerformed(evt);
   }
});
jInternalFrame4.getContentPane().add(jComboBox2, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(90, 10, 100, -1));
jButton8.setText("Cancel");
jButton8.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton8ActionPerformed(evt);
        }
    });

    jInternalFrame4.getContentPane().add(jButton8, new
      org.netbeans.lib.awtextra.AbsoluteConstraints(20, 40, -1, -1));
    jButton9.setText("Remove");
    jButton9.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton9ActionPerformed(evt);
        }
    });
    jInternalFrame4.getContentPane().add(jButton9, new
      org.netbeans.lib.awtextra.AbsoluteConstraints(110, 40, -1, -1));
    getContentPane().add(jInternalFrame4, new
      org.netbeans.lib.awtextra.AbsoluteConstraints(20, 30, 210, 110));
    jInternalFrame3.getContentPane().setLayout(new
      org.netbeans.lib.awtextra.AbsoluteLayout());
    jInternalFrame3.setTitle("Add File");
    jLabel3.setText("Select File");
    jInternalFrame3.getContentPane().add(jLabel3, new
      org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10, -1, -1));
    jButton5.setText("Browse...");
    jButton5.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton5ActionPerformed(evt);
        }
    });
```

```
jInternalFrame3.getContentPane().add(jButton5, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(80, 10, -1, -1));
jLabel4.setText("Description");
jInternalFrame3.getContentPane().add(jLabel4, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(10, 40, -1, -1));
jInternalFrame3.getContentPane().add(jTextField2, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(80, 40, 100, -1));
jButton6.setText("Cancel");
jButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton6ActionPerformed(evt);
    }
});
jInternalFrame3.getContentPane().add(jButton6, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, -1, -1));
jButton7.setText("Add");
jButton7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton7ActionPerformed(evt);
    }
});
jInternalFrame3.getContentPane().add(jButton7, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(100, 70, -1, -1));
getContentPane().add(jInternalFrame3, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(30, 30, 200, 130));
jInternalFrame2.getContentPane().setLayout(new
  org.netbeans.lib.awtextra.AbsoluteLayout());
jInternalFrame2.setTitle("Remove IP");
jLabel2.setText("IP");
```

```
jInternalFrame2.getContentPane().add(jLabel2, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(20, 10, -1, -1));
jButton3.setText("Cancel");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton3ActionPerformed(evt);
    }
});
jInternalFrame2.getContentPane().add(jButton3, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50, -1, -1));
jButton4.setText("Remove");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton4ActionPerformed(evt);
    }
});
jInternalFrame2.getContentPane().add(jButton4, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(90, 50, -1, -1));
jComboBox1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jComboBox1ActionPerformed(evt);
    }
});
jInternalFrame2.getContentPane().add(jComboBox1, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(70, 10, 100, -1));
getContentPane().add(jInternalFrame2, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(30, 30, 200, 120));
jInternalFrame1.getContentPane().setLayout(new
  org.netbeans.lib.awtextra.AbsoluteLayout());
```

```
jInternalFrame1.setTitle("Allow IP");
jLabel1.setText("IP");
jInternalFrame1.getContentPane().add(jLabel1, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(30, 20, -1, -1));
jInternalFrame1.getContentPane().add(jTextField1, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(80, 20, 100, -1));
jButton1.setText("Cancel");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton1ActionPerformed(evt);
    }
});
jInternalFrame1.getContentPane().add(jButton1, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(30, 50, 80, -1));
jButton2.setText("Allow");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton2ActionPerformed(evt);
    }
});
jInternalFrame1.getContentPane().add(jButton2, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(110, 50, -1, -1));
getContentPane().add(jInternalFrame1, new
  org.netbeans.lib.awtextra.AbsoluteConstraints(30, 30, 200, 120));
jMenu1.setText("File");
jMenuItem1.setMnemonic('X');
jMenuItem1.setText("Exit");
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jMenuItem1ActionPerformed(evt);
        }
    });

    jMenu1.add(jMenuItem1);
    jMenuBar1.add(jMenu1);
    jMenu2.setText("Http");
    jMenuItem2.setText("Allow IP");
    jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItem2ActionPerformed(evt);
        }
    });
    jMenu2.add(jMenuItem2);
    jMenuItem3.setText("Remove IP");
    jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItem3ActionPerformed(evt);
        }
    });
    jMenu2.add(jMenuItem3);
    jMenuBar1.add(jMenu2);
    jMenu3.setText("Ftp");
    jMenuItem4.setText("Allow IP");
    jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItem4ActionPerformed(evt);
        }
    });
```

```
    jMenu3.add(jMenuItem4);
    jMenuItem5.setText("Remove IP");
    jMenuItem5.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem5ActionPerformed(evt);
      }
    });
    jMenu3.add(jMenuItem5);
    jMenu3.add(jSeparator1);
    jMenuItem6.setText("Add File");
    jMenuItem6.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem6ActionPerformed(evt);
      }
    });
    jMenu3.add(jMenuItem6);
    jMenuItem7.setText("Remove File");
    jMenuItem7.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem7ActionPerformed(evt);
      }
    });
    jMenu3.add(jMenuItem7);
    jMenuBar1.add(jMenu3);
    jMenu4.setText("Help");
    jMenuItem8.setText("Content");
    jMenuItem8.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem8ActionPerformed(evt);
```

```
        }
    });
    jMenu4.add(jMenuItem8);
    jMenuItem9.setText("About");
    jMenuItem9.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem9ActionPerformed(evt);
      }
    });
    jMenu4.add(jMenuItem9);
    jMenuBar1.add(jMenu4);
    setJMenuBar(jMenuBar1);
    pack();
  }
  private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent
      evt) {
                  try
                  {
                        Runtime rr=Runtime.getRuntime();
                        File hf=new File("about.htm");
                        rr.exec("C:/Program Files/Internet
                        Explorer/iexplore.exe "+hf.getAbsolutePath());
                  }
                  catch(Exception euui)
                  {
                        System.out.println(euui);
                  }
}
```

```
    private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent
  evt) {
                        try
                        {
                              Runtime rr=Runtime.getRuntime();
                              File hf=new File("content.htm");
                              rr.exec("C:/Program Files/Internet
                              Explorer/iexplore.exe "+hf.getAbsolutePath());
                        }
                        catch(Exception euui)
                        {
                              System.out.println(euui);
                        }
    }
    private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent
evt) {
              jInternalFrame4.setVisible(true);
              jComboBox2.removeAllItems();
              try
              {
                    rs=stmt.executeQuery("select * from ftpfile");
                    while(rs.next())
                          jComboBox2.addItem(rs.getString(1));
              }catch(Exception ex)
              {
                    msgbox.setMessage("Error in Retriving File List","",1);
                    ex.printStackTrace();
              }
}
```

```
    private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent
  evt) {
      jInternalFrame3.setVisible(true);
    }
    private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
      try
      {
       int abc=0;
            abc=stmt.executeUpdate("delete * from ftpfile where
            fname='"+selec+"'");
                if(abc==1)
                    msgbox.setMessage("File Removed...","",1);
                jInternalFrame4.setVisible(false);
      }catch(Exception e)
      {
            msgbox.setMessage("Error in Deletion","",1);
            e.printStackTrace();
      }
  }
  private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
      jInternalFrame4.setVisible(false);
  }
  private void jComboBox2ActionPerformed(java.awt.event.ActionEvent
evt) {
    selec=(String)jComboBox2.getSelectedItem();
  }
  private void jButton7ActionPerformed(java.awt.event.ActionEvent evt)
{
    String descr=jTextField2.getText();
```

```
    if(fname.length()>0&fdir.length()>0&descr.length()>0)
    {
     try
     {
            int abc=stmt.executeUpdate("insert into ftpfile
            values('"+fname+"','"+fdir+"','"+descr+"')");
            if(abc==1)
                msgbox.setMessage("File Added Successfully...","",1);
            else
                msgbox.setMessage("Error in Adding File...","",1);
            jInternalFrame3.setVisible(false);
    }catch(Exception e)
    {
            msgbox.setMessage("Exception in Adding File...","",1);
            e.printStackTrace();
    }
   }
   else
     msgbox.setMessage("Please Select File Name& Description...","",1);
}
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField2.setText("");
  jInternalFrame3.setVisible(false);
}
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    fname="";
          fdir="";
          FileDialog fd = new FileDialog(this,"Select File..",0);
  fd.show();
```

```
        if ((fd.getDirectory() != null) && (fd.getFile() != null))
        {
                fname = fd.getFile();
          fdir = fd.getDirectory();
        }
    }
    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent
 evt) {      System.exit(0);
          }
    private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent
 evt) {
      flg=2;
      jComboBox1.removeAllItems();
              selec="";
      jInternalFrame2.setVisible(true);
      populateIP();
   }
   private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent
evt) {
      flg=2;
      jTextField1.setText("");
      jInternalFrame1.setVisible(true);
  }
  private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent
evt) {
      flg=1;
            selec="";
     jComboBox1.removeAllItems();
     jInternalFrame2.setVisible(true);
```

```
        populateIP();
    }
        public void populateIP()
        {
                try
                {
                        if(flg==1)
                                rs=stmt.executeQuery("select * from httpip");
                        if(flg==2)
                                rs=stmt.executeQuery("select * from ftpip");
                        while(rs.next())
                                jComboBox1.addItem(rs.getString(1));
                }catch(Exception ex)
                {
                        msgbox.setMessage("Error in Retriving IP List","",1);
                        ex.printStackTrace();
                }
        }
   private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent
evt) {
      flg=1;
      jTextField1.setText("");
      jInternalFrame1.setVisible(true);
   }
   private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
       jInternalFrame2.setVisible(false);
   }
   private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
   {
```

```
        try
        {
         int abc=0;
         if(flg==1)
                abc=stmt.executeUpdate("delete * from httpip where
                ip='"+selec+"'");
                if(flg==2)
                abc=stmt.executeUpdate("delete * from ftpip where
                ip='"+selec+"'");
                      if(abc==1)
                            msgbox.setMessage("IP Removed...","",1);
                      jInternalFrame2.setVisible(false);
                }catch(Exception e)
                {
                      msgbox.setMessage("Error in Deletion","",1);
                      e.printStackTrace();
                }
  }
   private void jComboBox1ActionPerformed(java.awt.event.ActionEvent
evt) {
      selec=(String)jComboBox1.getSelectedItem();
   }
   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
     jInternalFrame1.setVisible(false);
   }
   private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
       String ip=jTextField1.getText();
       if(ip.length()>0)
       {
```

```
        if(flg==1)
    {
        try
        {
                int abc=stmt.executeUpdate("insert into httpip
                values('"+ip+"')");
                if(abc==1)
                        msgbox.setMessage("New IP Allowed For
                        Http...","",1);
                jInternalFrame1.setVisible(false);
        }catch(Exception e)
        {
                msgbox.setMessage("IP Already Present in Http...","",1);
        }
}
    if(flg==2)
  {
        try
        {
                int abc=stmt.executeUpdate("insert into ftpip
                values('"+ip+"')");
                if(abc==1)
                        msgbox.setMessage("New IP Allowed to
               Ftp...","",1);
               jInternalFrame1.setVisible(false);
    }catch(Exception e)
    {
               msgbox.setMessage("IP Already Present in Ftp...","",1);
    }
```

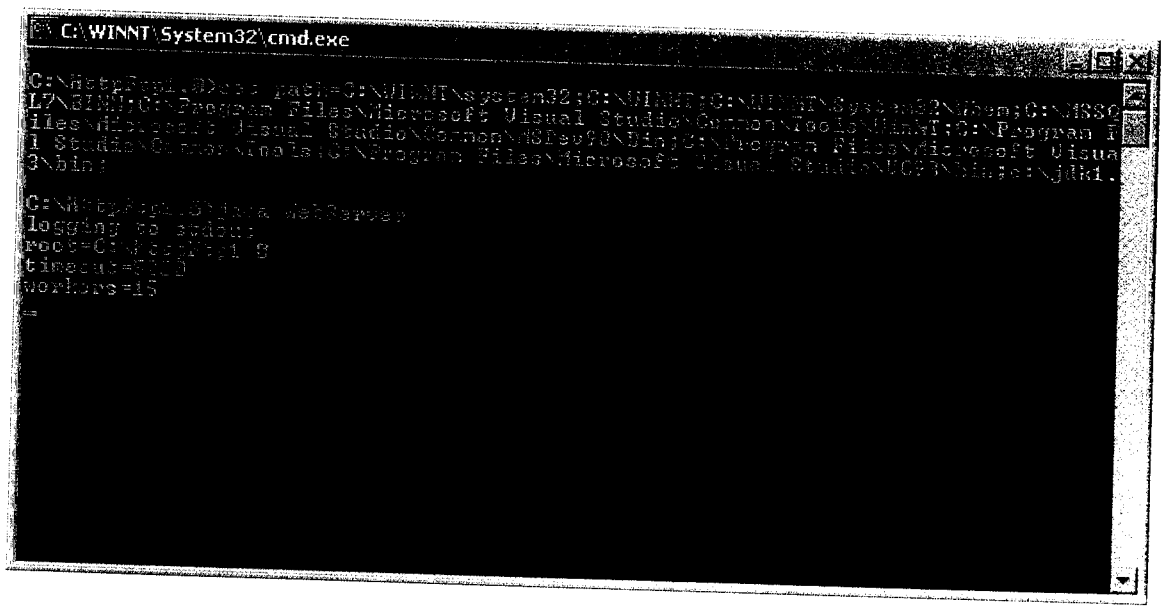
```
        }
      }
        else
              msgbox.setMessage("Please Enter Valid IP Address...","",1);
  }
  private void exitForm(java.awt.event.WindowEvent evt) {//GEN-
      FIRST:event_exitForm
     System.exit(0);
  }
  public static void main(String args[]) {
     new HttpFtp().show();
  }
  private javax.swing.JButton jButton9;
  private javax.swing.JButton jButton8;
  private javax.swing.JButton jButton7;
  private javax.swing.JButton jButton6;
  private javax.swing.JButton jButton5;
  private javax.swing.JButton jButton4;
  private javax.swing.JButton jButton3;
  private javax.swing.JButton jButton2;
  private javax.swing.JSeparator jSeparator1;
  private javax.swing.JButton jButton1;
  private javax.swing.JComboBox jComboBox2;
  private javax.swing.JMenu jMenu4;
  private javax.swing.JComboBox jComboBox1;
  private javax.swing.JMenu jMenu3;
  private javax.swing.JMenu jMenu2;
  private javax.swing.JMenu jMenu1;
  private javax.swing.JMenuItem jMenuItem9;
```

```
    private javax.swing.JMenuItem jMenuItem8;
    private javax.swing.JMenuItem jMenuItem7;
    private javax.swing.JMenuItem jMenuItem6;
    private javax.swing.JMenuItem jMenuItem5;
    private javax.swing.JMenuItem jMenuItem4;
    private javax.swing.JMenuItem jMenuItem3;
    private javax.swing.JMenuItem jMenuItem2;
    private javax.swing.JMenuItem jMenuItem1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JInternalFrame jInternalFrame4;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JInternalFrame jInternalFrame3;
    private javax.swing.JInternalFrame jInternalFrame2;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JInternalFrame jInternalFrame1;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JMenuBar jMenuBar1;
    private int flg = 0;
    private Connection con;
    private Statement stmt;
    private ResultSet rs;
    private MessageBox msgbox;
    private String selec;
    private String fname;
    private String fdir;
}
```

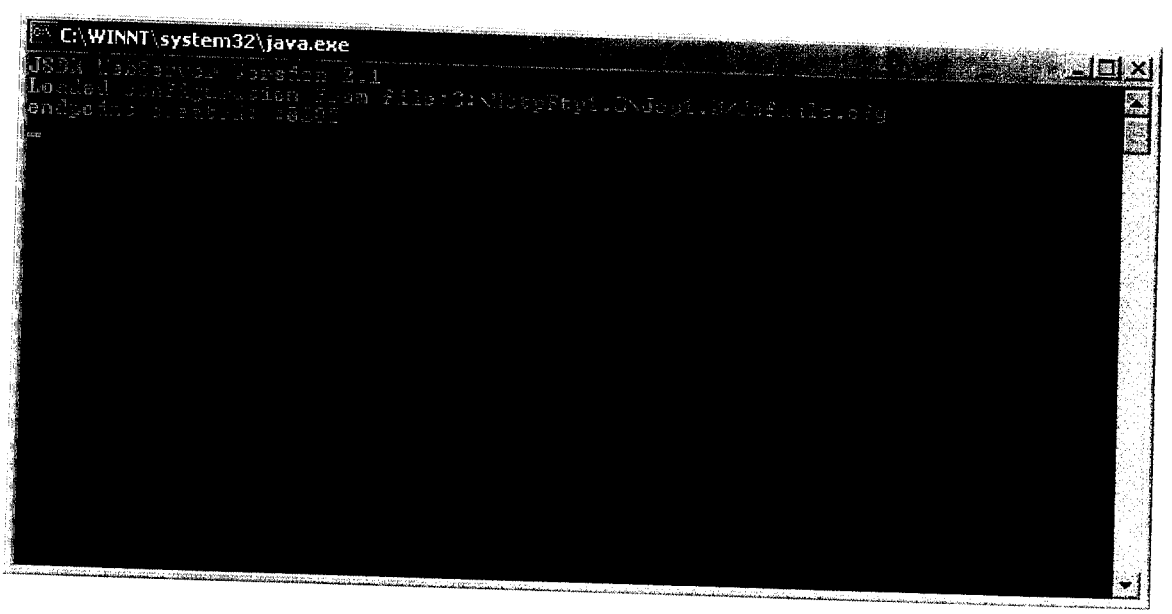
APPENDIX – 2

SNAP SHOTS

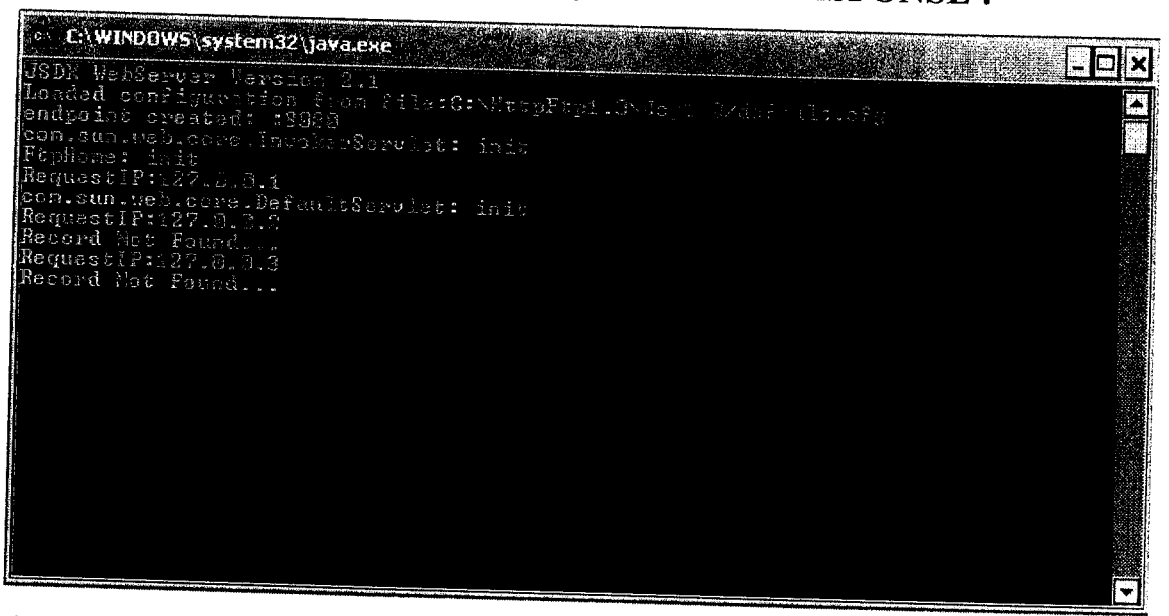
1. WEB SERVER :



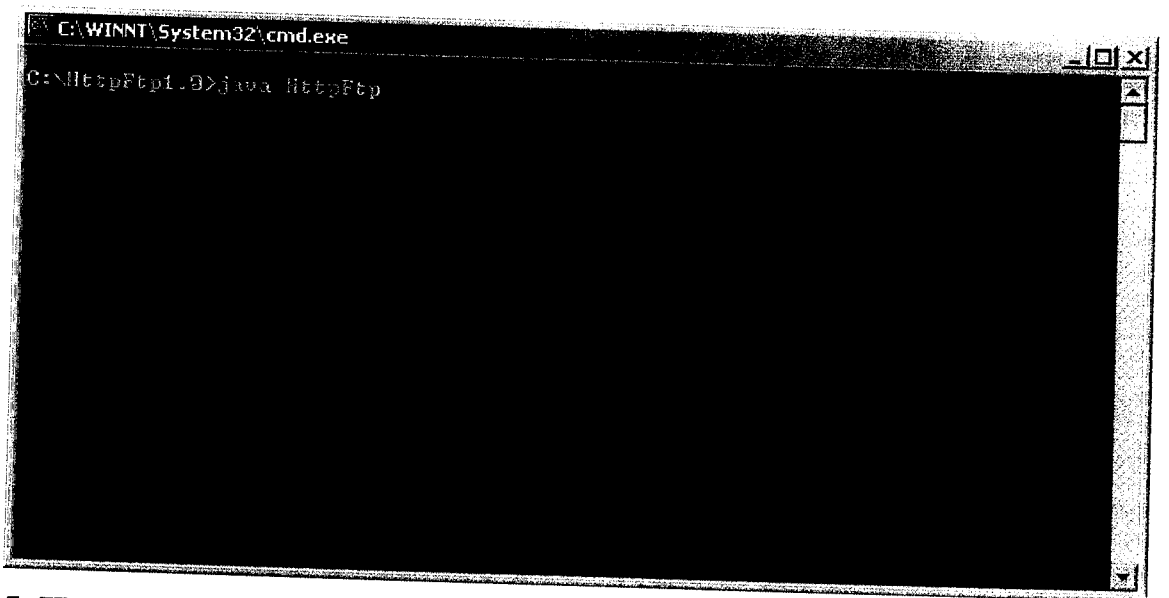
2. JSDK WEBSERVER :



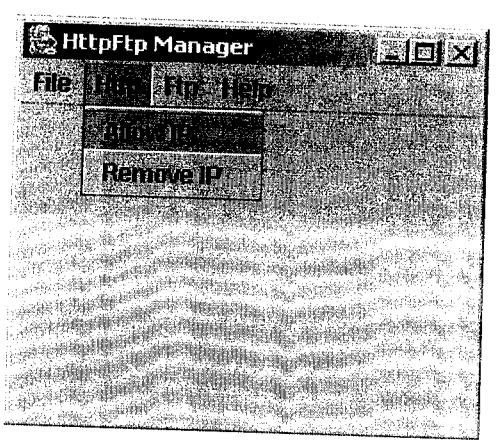
3. JSDK MONITORING FTP REQUEST AND RESPONSE :



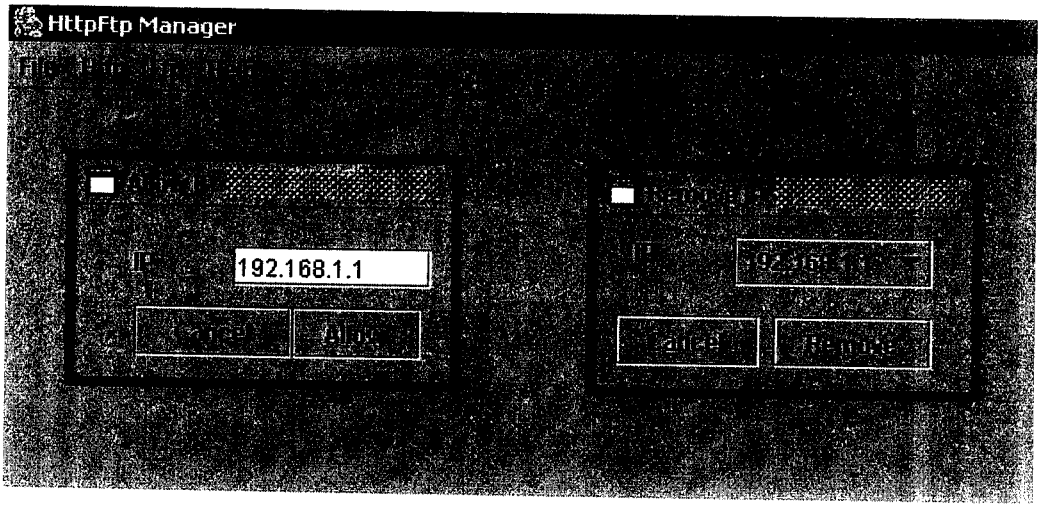
4. HTTPFTP MANAGER :



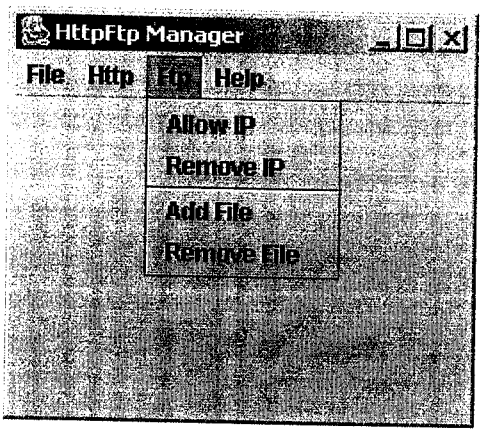
5. HTTP MENU :



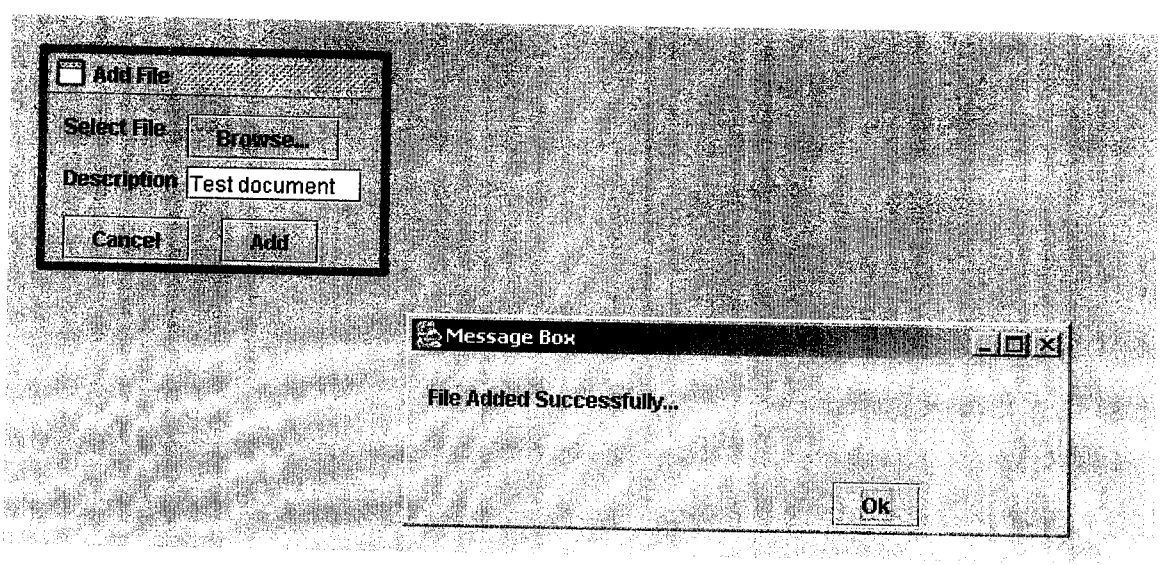
5. ADDING AND REMOVING IP FOR HTTP :



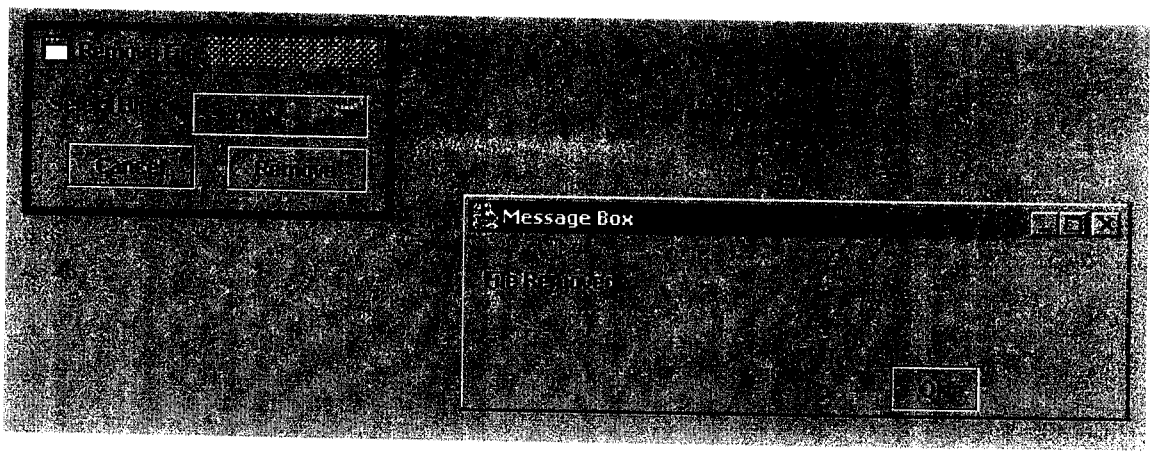
6. FTP MENU :



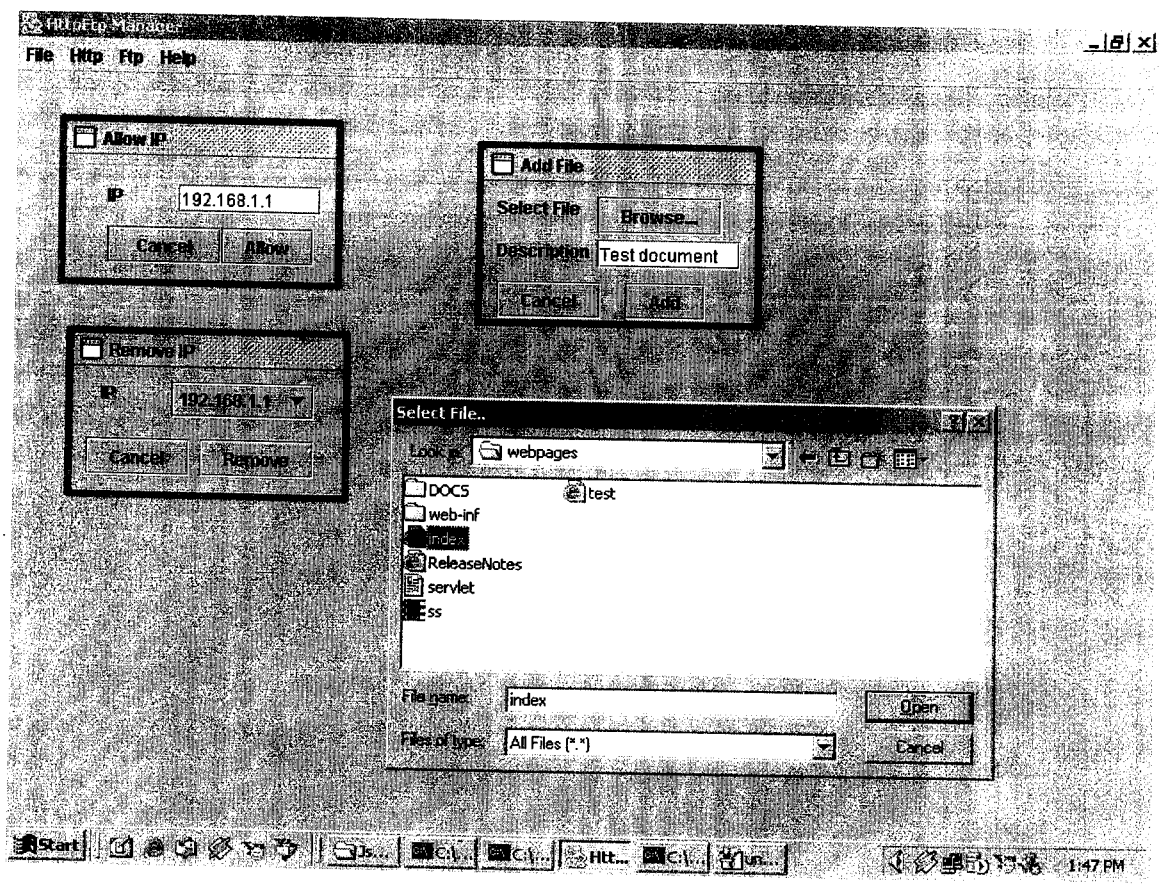
7. ADDING THE FILE FOR FTP :



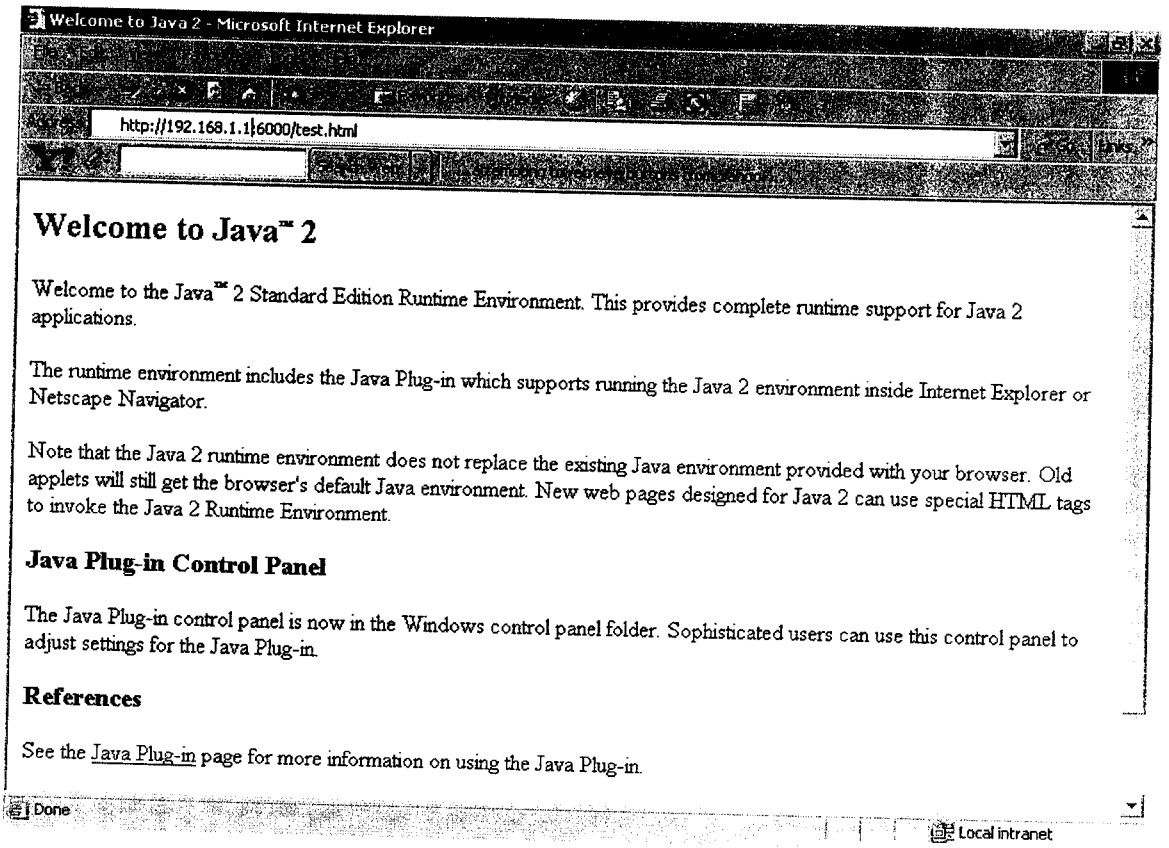
8. REMOVING THE FILE FOR FTP :



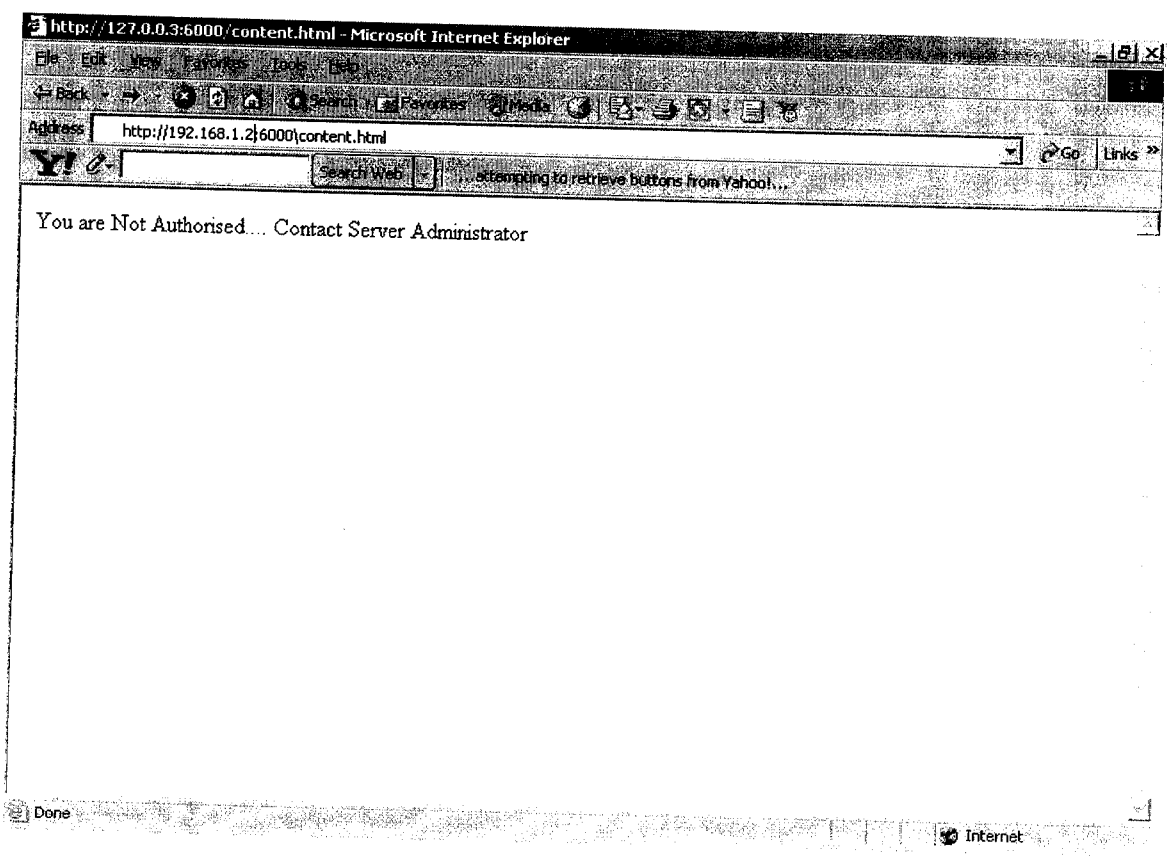
9. ADDING AND REMOVING IP FOR FTP :



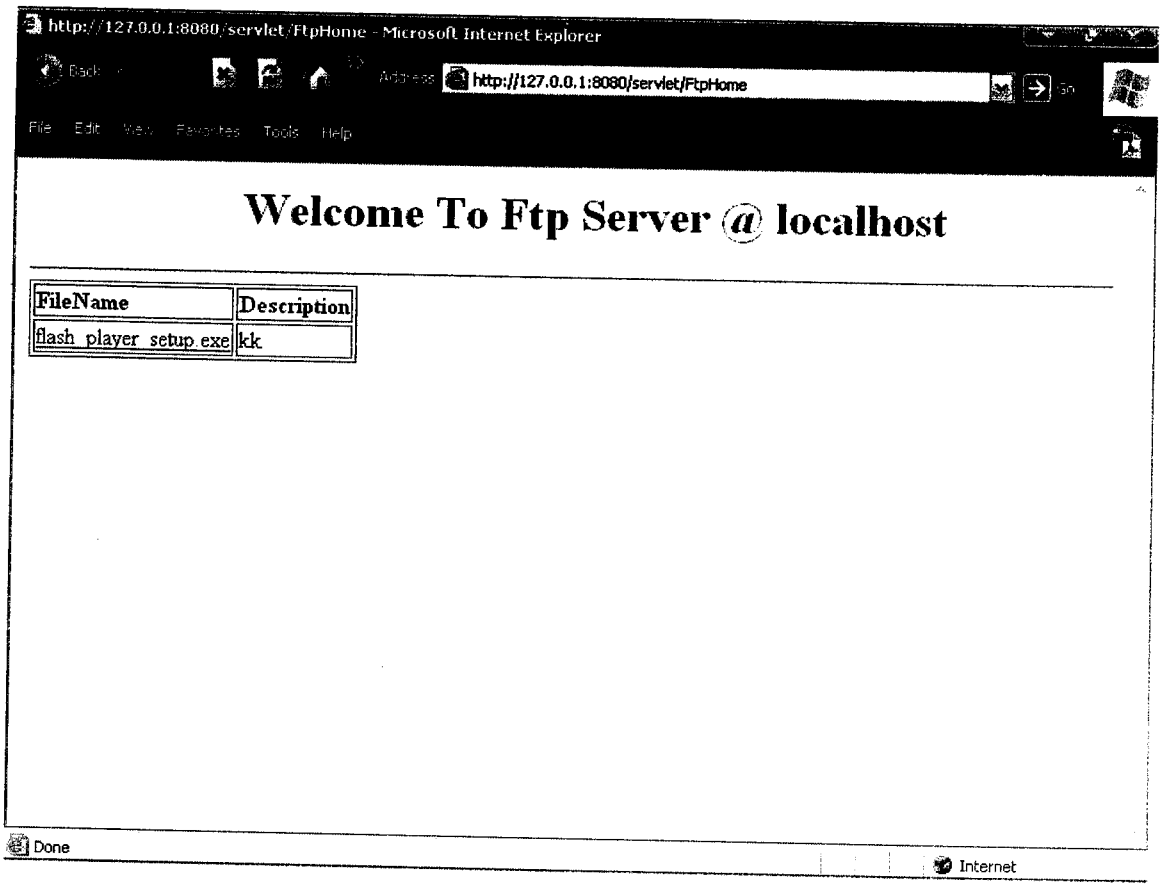
10. HTTP VALID IP OUTPUT :



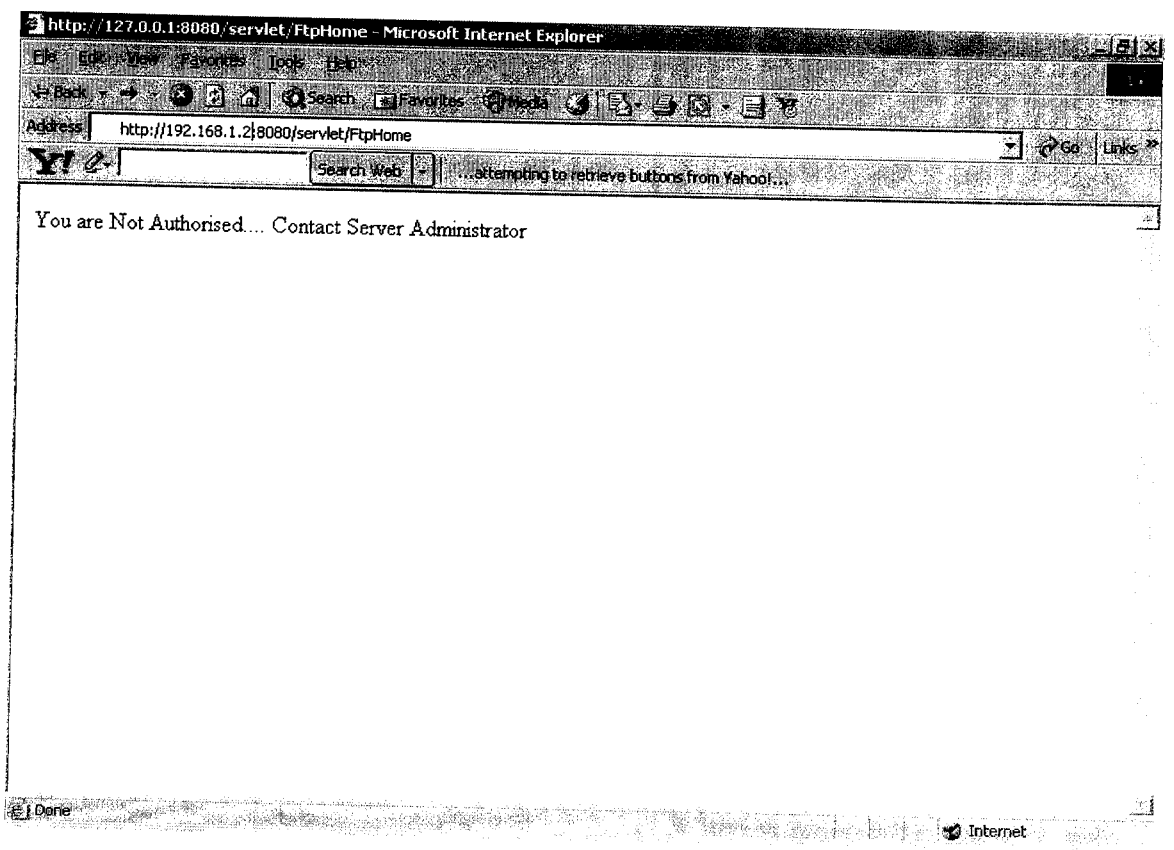
11. HTTP INVALID IP OUTPUT :



12. FTP VALID IP OUTPUT :



13. FTP INVALID IP OUTPUT :



REFERENCES

REFERENCES

1. A. Rubin, D. Geer, and M. Ranum, (1997) *Web Security Sourcebook*, Wiley Computer Publishing,.
2. W.R. Cheswick and S.M. Bellovin, (1994) *Firewalls and Internet Security*: Repelling the Wily Hacker, Addison Wesley.
3. A. Wool, "*The Use and Usability of Direction-Based Filtering in Firewalls,*" Computers & Security, in press; available online 2 Apr. 2004; www.sciencedirect. com/science/journal/01674048.
4. Peter Nortons (2001), "*Guide to java Programming*". Tata McGraw-Hill Publishing Company.
5. *Programming with Java*, 2nd edition Tata McGraw-Hill Publishing Company.
6. Rick Proctor (1998), "*The Java Communications API A Working Example*", Prentice Hall.
7. John Zukowski (1997), "*Mastering Java*", BPB publishers.
8. Complete Java NIIT course material