

REMOTE FILE ACCESSING SYSTEM



PROJECT WORK DONE AT

PPP INFOTECH,
CHENNAI

PROJECT REPORT

φ- 561

*Submitted in partial fulfillment of the
Requirements for the award of the degree of
M.Sc Applied Science (Computer Technology)
Of Bharathiar University, Coimbatore.*

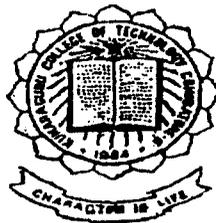
Submitted By

T.RAJINI KANTH
(Reg. No. 9937Q0015)

Guided By

EXTERNAL GUIDE
Mr. Prameshwar babu
Director,
PPP InfoTech,
Chennai.

INTERNAL GUIDE
Mr. R.Dinesh B.Tech.,M.S.,
Senior Lecturer,
Dept. of Comp.Sci & Engg,
Kumaraguru College of Tech,
Coimbatore-641 006.



Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore – 641 006.

May 2001.

CERTIFICATE

CERTIFICATE

This is to certify that the project work entitled

“REMOTE FILE ACCESSING SYSTEM”

Done by

**T.RAJINI KANTH
(REG. NO. 9937Q0015)**

Submitted in partial fulfillment of the requirements for the award of the degree of
M.Sc Applied Science (Computer Technology) of Bharathiar University.

S. Jayaram
Professor and Head 23/4/01

R. Srinivasan
(201805182)
Internal Guide 2001

Submitted to University Examination held on 27/04/2001

R. Srinivasan
(27/04/2001)
Internal Examiner

H. Abdul Kalam
(27/4/2001)
External Examiner

PPP INFOTECH LIMITED

16, Periyar Street, Gandhi Nagar
Saligramam, Chennai - 600 093. India

Phone : 4831145, 4848328
Telefax : 91-44-4848328
e-mail : info@pppindia.com
website : www.pppindia.com
www.pppinfotech.com

BONAFIDE CERTIFICATE

Certified that this Project work on **Remote File Accessing System** is the Bonafide work of **T. RAJINI KANTH** who carried out the project under my Supervision. Certified further that the project was finished by him successfully.


2/04/2001

DECLARATION

DECLARATION

I hereby declare that the project entitled

“REMOTE FILE ACCESSING SYSTEM”

Submitted to **Bharathiar University** as the project work of **M.Sc Applied Science Computer Technology** Degree, is a record of original work done by me under the supervision and guidance of Mr. Prameshwar babu Director PPP InfoTech Chennai and Mr.Dinesh Ranganathan,B.Tech.,M.S.(Comp.Sci, Wisconsin) **Lecturer, Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore** and this project work has not found the basis for the award of any Degree/Diploma/ Associate ship/Fellowship or similar title to any candidate of any university.

Place: *Coimbatore*

Date: *29-04-2001*


Signature of the student

CONTENTS

CONTENTS

	Page No.
ACKNOWLEDGEMENT	1
SYNOPSIS	2
1. INTRODUCTION	3
1.1 PROJECT OVERVIEW	3
1.2 ORGANIZATION PROFILE.	5
2. SYSTEM STUDY AND ANALYSIS	7
2.1 EXISTING SYSTEM – LIMITATIONS.	7
2.2 PROPOSED SYSTEM.	8
2.3 REQUIREMENTS ON NEW SYSTEM.	8
2.4 USER CHARACTERISTICS.	9
3. PROGRAMMING ENVIRONMENT.	11
3.1 HARDWARE CONFIGURATION.	11
3.2 DESCRIPTION OF SOFTWARE AND TOOLS USED.	12
4. SYSTEM DESIGN & DEVELOPMENT.	16
4.1 INPUT DESIGN	19
4.2 PROCESS DESIGN	24
4.3 OUTPUT DESIGN.	29
4.4 DATABASE DESIGN.	30
5. SYSTEM IMPLEMENTATION & TESTING	32
5.1 SYSTEM IMPLEMENTATION.	32
5.2 SYSTEM TESTING.	32
5.3 REFINEMENTS MADE ON FEEDBACK	41
6. CONCLUSION AND SCOPE FOR FUTURE DEVELOPMENT	42

BIBLIOGRAPHY

43

APPENDICES

44

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I would like to thank the following people who helped me in accomplishing the task of completing this project and writing this report.

I thank **Mr.Parameshwar Babu** who as a guide gave me excellent guidance, valuable suggestions and guided me through out the project.

Its my great privilege and pleasure of mine to express my deep sense of gratitude to esteemed **Dr. K.K.Padmanaban B.Sc., (Engg)., M.Tech., Ph.D.,** Principal, K.C.T college.

I express my deep sense of gratitude to for guiding me with his valuable and timely support and extended me through out the project, with his valuable guidance comments and through out my M.Sc Computer Technology course.

I reciprocate the kindness shown to me by my friends and family members, staff members of my department of Computer Science.

SYNOPSIS

Synopsis

This system is a solution for accessing the local computer's file system from any remote computer through the World Wide Web and is intended to provide secure functionality, allowing for sharing of an existing File System among multiple hosts. The wide file-sharing environment is characterized as typically having many client machines that may be geographically dispersed and under the control of several different groups of administrators. The client machines can access many different file servers and use them to share data— so that one user can access another user's document.

This system allows the user to explore his computer's file system and to see the contents of it from any remote location. It does not rely upon the older protocols such as telnet or ftp. This is an Http based file transfer system and the advantage is that, it is free from any firewall's restrictions.

At the core of the design is the interaction of the client and server. The File Server resides on the local computer and connects to the server, which is running on a remote computer with the master copy of the File System. The client/server design ensures that the program handles all communication and that the File System is protected from editing by multiple sources. The server is configured via a configuration file to define, who can connect and what their base path will become on the local file system. The server also maintains multiple user security by only allowing a client that has provided a correct user name and password to access the file system. Multiple users will have the ability to have individual task files since they will be logging onto their unique account and accessing their individual files.

INTRODUCTION

1. Introduction

The World Wide Web (WWW) is a phenomenon that has taken much of the world by storm over the last few years. Historically, the Internet, a vast worldwide network of computers, has been used to share documents, send electronic mail, and login to remote computers, with file transfers accounting for much of the Internet traffic.

This system is again a solution for accessing the local computer's file system from any remote computer through the World Wide Web and is intended to provide secure functionality, allowing for sharing of an existing File System among multiple hosts. The wide file-sharing environment is characterized as typically having many client machines that may be geographically dispersed and under the control of several different groups of administrators. The client machines can access many different file servers and use them to share data—for example, so that one user can access another user's document.

1.1. Project Overview

This system allows the user to explore his computer's file system and to see the contents of it from any remote location. It does not rely upon the older protocols such as telnet or ftp. This is an Http based file transfer system and the advantage is that, it is free from any firewall's restrictions.

At the core of the design is the interaction of the client and server. The File Server resides on the local computer and connects to the server, which is running on a remote computer with the master copy of the File System. The client/server design ensures that the program handles all communication and that the File System is protected from editing by multiple sources. The server is configured via a configuration file to define, who can connect and what their base path will become on the local file system. The server also

maintains multiple user security by only allowing a client that has provided a correct user name and password to access the file system. Multiple users will have the ability to have individual task files since they will be logging onto their unique account and accessing their individual files

To ensure that the program is easy to use, a complete graphical interface has been designed to handle all users input. The graphical interface will coordinate with the client, passing along any instructions it received from the user. It will also include a tree structure that will simplify file system organization by allowing the user to view in like windows explorer style. In addition to delegating tasks taken from input, this interface will display any useful messages received from other objects, including errors encountered or general events. The other user interface includes,

- Hiding internals
- Providing an uniform appearance and look
- Dividing up the choices into manageable frames.

The client machines are typically themselves File Servers providing services to their own set of client machines. The File Servers typically run an identical set of operating system and application software, and are configured so that the ultimate users of the application see a single service.

The client will have to use a specific operating system due to a lack of unified user level file system interface. We have chosen Windows as the remote file client platform, although our intent is to make it as simple as possible to port the client to another operating system. We have chosen Windows due to familiarity, ease of modification, and widespread use.

1.2. Organization Profile



PPP India

The history of PPP is as old as the history of Internet in India itself. Beginning within months of Internet's arrival in India, the company has grown in products and services, expertise and reputation as Internet revolution snowballed into a real people's movement.

Founded in 1995, PPP is perhaps the only company in India which shunned the popular way of tapping the IT pie, that is by harnessing offshore software development contracts, body-shopping and the like. Instead, they invested on technology development and that paid dividends. The company is the first in India to come out with Internet products when product development was not an accepted practice at all in India. The products went on to become real hits not just in India, but abroad too. The media reports the feedback by users and our growth in the Indian and global markets testify to this success.

PPP started with software and utilities, scripts for UNIX shell accounts, creation of Internet databases and promotion of Internet through tailor-made workshops. It founded the Chennai chapter of the Internet Users Club of India (IUCI) and acted as an agent between various agencies to propel the growth of Internet use in the southern part of the country.

PPP was also instrumental in unleashing the Internet booth revolution in Chennai, which led the way for the whole country. From just about seven, the number of Internet browsing centers in the city galloped to a whopping 300, gem of idea promoted by the

company. Its maiden product, PPPshar, as media thanks to the reports hailed, powered this growth.

It was the first and only company to enter the global market leveraging the Internet. It was also able to explore new markets and develop products and solutions to suit the conditions in these markets. The company has proven expertise in Internet, Java & networking, E-commerce, Web database and Web programming.

With its dominant strength in the area of Internet and networking software, the company is all set to develop innovative technologies and software in the emerging area of Internet devices.

PPP InfoTech Ltd. is a member of

- Indo-US Chamber of Commerce
- Indo-German Chamber of Commerce
- Software Technology Parks of India (STPI)
- STPI Business Support Center, Silicon Valley, USA
- Association of Shareware Professionals, USA

PPP InfoTech Ltd.,

16, Periyar Street, Gandhi Nagar,

Saligramam, Chennai 600 093

Tamilnadu, India. Phone: 91-44-4831145

Telefax: 91-44-4848328.

Email: info@pppindia.com

www.pppindia.com

SYSTEM STUDY AND ANALYSIS

2. System study & Analysis

This section describes the general flow of information in the system. It describes the various problems the software must solve, functions of the project and the user characteristics. The system study was made on the existing system to find out the Drawbacks.

2.1. Existing System Limitations

In the existing system a user can access his remote computer file system through FTP.

The disadvantages in using FTP

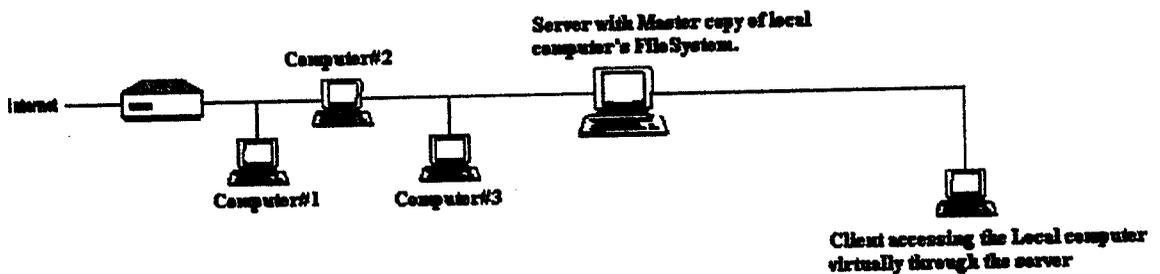
- More expensive.
- Needs more space and difficult to connect.
- Traditionally limited to text only (Binary file transfer is possible, but usually platform specific).
- Often difficult, especially for novice Internet users due to multi step sequence of commands.

On the other hand we have HTTP file transfer, which were restricted only for downloading. There are some other tools available, which bring the actual desktop to the remote computer. The drawbacks of this system are that the transfer rate is very slow, due to the vendor specific compression algorithm, and in the base they all were using the same File transfer protocol.

2.2. Proposed System

In the proposed system it was decided to reduce the overhead caused by the older systems. Http file transfers have been traditionally restricted to download only. This proposed system allows us to upload different files simultaneously. This upload does not need any special tools and can be uploaded directly from the browser itself.

The following sketch illustrates the proposed system



2.3. Requirements on new System

This chapter contains the specific requirements for this product, organized by interfaces and the various user classes.

The scope of the system involves

- Setting up a virtual network neighborhood environment.
- Tools for accessing, sharing files across the virtual network or from the local computer.
- Providing authentication, file-accessing permissions to the account holder.
- Developing a tool for direct interaction with the virtual network or the local computer.
- Developing a tool (RMI File Server), which will run on the client's local computer.

The central goal is to test the following claims:

1. A simple design can be used to achieve most of the functionality of NFS (Network file system), as well as provide security and cache consistency.
2. A File system implemented mostly in user level and mostly in Java can still provide adequate performance.

External Interface Requirements

The external interface requirements include user and communications interfaces.

User Interfaces

1. The user interface in this system has an explorer-based interface in much the same form as the existing Windows explorer interface. This interface shall provide the functionality described in Section <3.2.1> below.
2. The product shall have the capability to present web-based interfaces to all classes of users. All the users shall be presented with same interfaces based on their needs, as described in Section 3.2.1 below.

Communications Interfaces

The product shall be able to communicate with the remote server to acquire additional information about users as necessary.

User Characteristics

There are three distinct types of users for this system:

- Browser client accessing remote computer.
- Server Administrator.
- Remote system administrator.

Browser client and Remote system administrator.

The browser client accessing remote computer is the owner of the file system. This user also acts as administrator when he likes to share the files with other users. He has the permission to grant or revoke rights to other users. These users need an easy interface and cannot dedicate much time to learning a complex interface for administrative purpose.

Server Administrator

Administrators are responsible for managing all the users in the system. The administrative tasks are mostly automated, except some tasks like retrieval of lost password or server idiosyncrasies.

PROGRAMMING ENVIRONMENT

3. Programming Environment

In approaching this design, it was decided that Java was the best choice for the implementation language. Java is a widely supported platform independent language, it is suited for network applications, and the developers were already familiar with the latest version.



System Environments:

Application Server	Tomcat
Tools	JDK 1.2.1 JSDK 2.1
Platform	Windows 98

3.1. Hardware configuration

Processor	Pentium III 450MHz
Mother Board	Intel YM430tx
Clock Speed	200 MHz MMX
RAM	64 MB RAM
Video RAM	512KB
Network Interface Card	16 Bit Ethernet
Ports	2 Serial, 2 Parallel
File System	FAT-32

3.2. Description of software and tools used – Reasons for the choice.

Tools Used

- Remote Method Invocation (RMI)
- Servlets
- JavaScript

Description of software

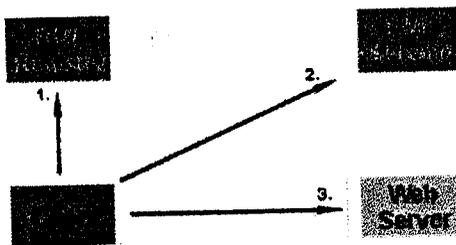
Remote Method Invocation (RMI)

Remote method invocation allows Java developers to invoke object methods, and have them execute on remote Java Virtual Machines (JVMs). Under RMI, entire objects can be passed and returned as parameters, unlike much remote procedure call based mechanisms, which require parameters to be either primitive data types, or structures composed of primitive data types. That means that any Java object can be passed as a parameter - even new objects whose class has never been encountered before by the remote virtual machine.

This is an exciting property, because it means that new code can be sent across a network and dynamically loaded at run-time by foreign virtual machines. Java developers have a greater freedom when designing distributed systems, and the ability to send and receive new classes is an incredible advantage. Developers don't have to work within a fixed codebase - they can submit new classes to foreign virtual machines and have them perform different tasks. When working with remote services, RMI clients can access new versions of Java services as they are made available - there's no need to distribute code to all the clients that might wish to connect. While code can be accessed from a local or remote file-system, it can also be accessed via a web server, making distribution easier. RMI also supports a registry, which allows clients to perform lookups for a particular service. The following diagram shows the interaction between different components of an

RMI system. Clients that know about a service can look up its location from a registry and access the service. If a new class is required, it can be downloaded from a web server.

Remote method invocation has a lot of potential, from remote processing and load sharing of CPU's to transport mechanisms for higher level tasks, such as mobile agents which execute on remote machines (Reilly, 1998). Because of the flexibility of remote method invocation, it has become an important tool for Java developers when writing distributed systems.

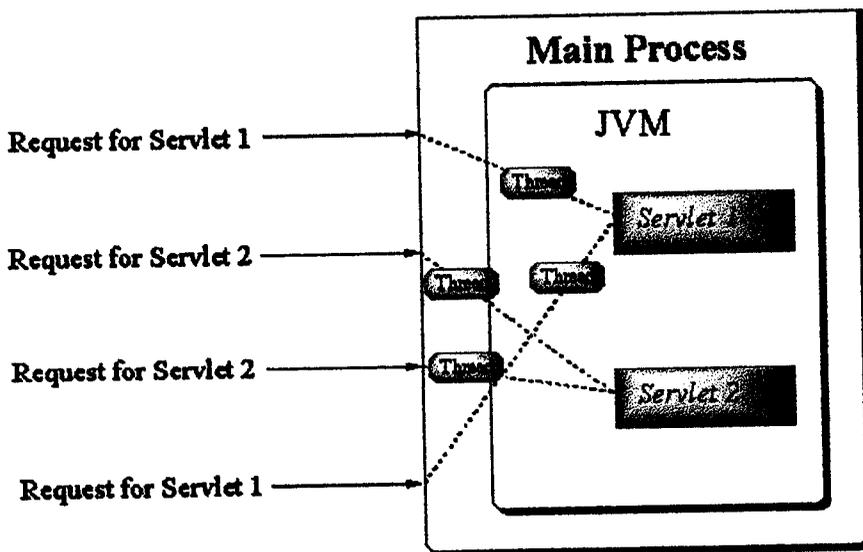


Servlets

The rise of server side Java application is one of the latest and most exciting trends in Java programming. The Java language was originally intended for use in small, embedded devices. It was first hyped as a language for developing client side programming in the form of applet. Until recently, Java's potential, as a server side programming has been overlooked. Now, Java is coming into its own as a language ideally suited for server side development. The cross-platform nature of Java is extremely useful for organizations that have a heterogeneous collection of servers running various Unix and windows operating system. Java's modern, object-oriented, memory-protected design allows developers to cut development cycles and increase reliability. In addition,

Java's built-in support for networking and enterprise APIs provides access to legacy data; easing the transition from older client/server systems to Java and Java Servlet based system. Java Servlets are a key component of server side Java development. A Servlet is a small, pluggable extension to a server that enhances the server's functionality. Servlets allow developer to extend and customize any Java-enabled server: a web server, a mail server, an application server, or any custom server.

The life cycle is the one of most exciting features of Servlets. Once Servlet is created and initialized by `init()` method, it runs inside a Java Virtual Machine (JVM) on the server (see Figure), so it is safe and portable. Servlets operate solely with the domain of the server; they can interact very closely with the server. The multi-requests from clients are handling by multi-thread of a single Servlets instance. This means that Servlets are efficiency, scalable and persistent. When necessary, the Servlets will be destroyed by Servlets' `destroy()` method and garbage collected.



The Servlet Life Cycle

HTML

HTML stands for Hyper Text Markup Language, which is an application of Standard Generalized Markup Language. It is a simple language used to define and describe the layout of a web page. HTML also supports Multimedia and document links. It consists of special codes which when embedded in text, adds formatting. The special characters, which separate HTML from ordinary text, are the left and right brackets (< & >). These brackets contain instructions known as TAGS that are not case sensitive.

JavaScript

JavaScript is Netscape's cross-platform, object-based scripting Language for client and server applications. JavaScript allows one to create Applications that run over the Internet. Using JavaScript, dynamic HTML pages can be created

SYSTEM DESIGN & DEVELOPMENT

4. System Design & Development

There are four major modules in this server:

- (1) RMI File Server
- (2) File Manager Servlet
- (3) File uploading and downloading
- (4) Message Handler.

Module Functionality

RMI File Server

This is the core module of the entire system. When this Server is requested for a file it simply delivers the requested file. It can respond to as many requests simultaneously and successively. The Security manager takes responsibility for ensuring that no files are supplied below the document root directory. This is done by resolving all relative references in the path of the requested file (/path/.../file.dat corresponds to /file.dat) and determining the absolute path of the document root directory as well as of the requested file. These canonical paths are unique for each file system. Thus, if the path of the requested file does not begin with the path of the document root directory, this is an unauthorized request. Every time a request (for a file) is made there is a comparison algorithm to check whether the requested path begins with the canonical path of the document root directory. System files, hidden files, and driver files are not supplied to the client for security reasons.

When a change takes place in the master copy of the file system, resides in the file manager Servlet, this system is updated to the changes. The entire mechanism takes place through message passing. Each object received a message is responsible for handling it in

a suitable way. Responsibilities are defined based on level of knowledge of the system that this particular object has. For example, A File copy object will receive an appropriate message from the message handler to call the respective method, and the respective method in turn may call another method depends upon the definition.

File Manager Servlet

The File Manager Servlet encapsulates the services of a typical file manager. It uses simple data structures to store the files such as a tree structure. It maintains a configuration file about the user's profile. When a client request is made to access the file system, it simply checks in the configuration file.

The format of the user structure on the disc will be as follows:

- User Name (String)
- Login Name (String)
- User ID (UID) (short)
- Password (String)
- Admin flag to tell if the user has admin rights (1 byte)

After start-up this information is stored in User objects.

The format of the group structure on the disc will be as follows:

- Group Name (String)
- Group ID (GID: short)
- Group Members

After start-up this information is stored in a runtime list

The format of the File structure connects a file to a group and a user. It also contains the rights for a File. The structure on the disc is:

- File name (String)
- File ID (FID: short)
- Group ID (GID: short)
- File owner (UID: short)
- Rights (byte)
- Date of the last modification (int)
- Location of the file (String)

User Log

This file contains Information about users login, logout and writings. This is a ASCII file. The RMI File Server updates the following user configuration file whenever it is starting up. The file has the following values

KEY	VALUE	EXAMPLE
RMI_FS.IPADDRESS	IP address of the remote machine	213.47.119.173
RMI_FS.DOCBASE	The canonical path of the remote machine	Root/user specified
RMI_FS.PORT	port on which the remote file is listening	Port No.
RMI_FS.USERNAME	<<User Name>>	Scott
RMI_FS.PASSWORD	<<Password>>	Tiger

The client IP address is stored in a configuration file, since the user may be using a dial-up line, in that case the IP address is said to be dynamic. To avoid ambiguity, the IP address is stored in the server config file each time it starts up.

File Uploading/Downloading

File uploading/downloading is an individual module because, when this request is made the message handler directly takes the request to the RMI File Server. When a file is uploaded it is saved in the RMI File Server.

Message Handler

The Message Handler is responsible for sending and receiving messages. This Message Handling mechanism is used to exchange messages inside the subsystems (entire process). This Message Handler has a Method invoking sequence description, which shows that whenever a message enters the system it is first interpreted by the object that receives it. Interpreting means here creating corresponding XxxMsg object based upon message id derived from message that was received. Created message object is then sent to next object by calling its receive method.

4.1. Input Design

The input to the system is caused by two ways,

1. User Input
2. Messages

1. User Input

The user makes the following inputs,

- Login Details
- File Managing Operations

The users of the remote system need to be able to log on to the system. The connection to the system will be over secure HTTP (SSL). The file managing operations are interpreted as messages, which in turn will call the message handler.

Requirements for Login Details

- Login to system.
- Encrypt user details over HTTPS.
- Logout of system.
- Go to register screen for non-registered user

Requirements for File Managing Operations

- File Rename
- File Delete
- File Update
- File Upload/download

2. Messages

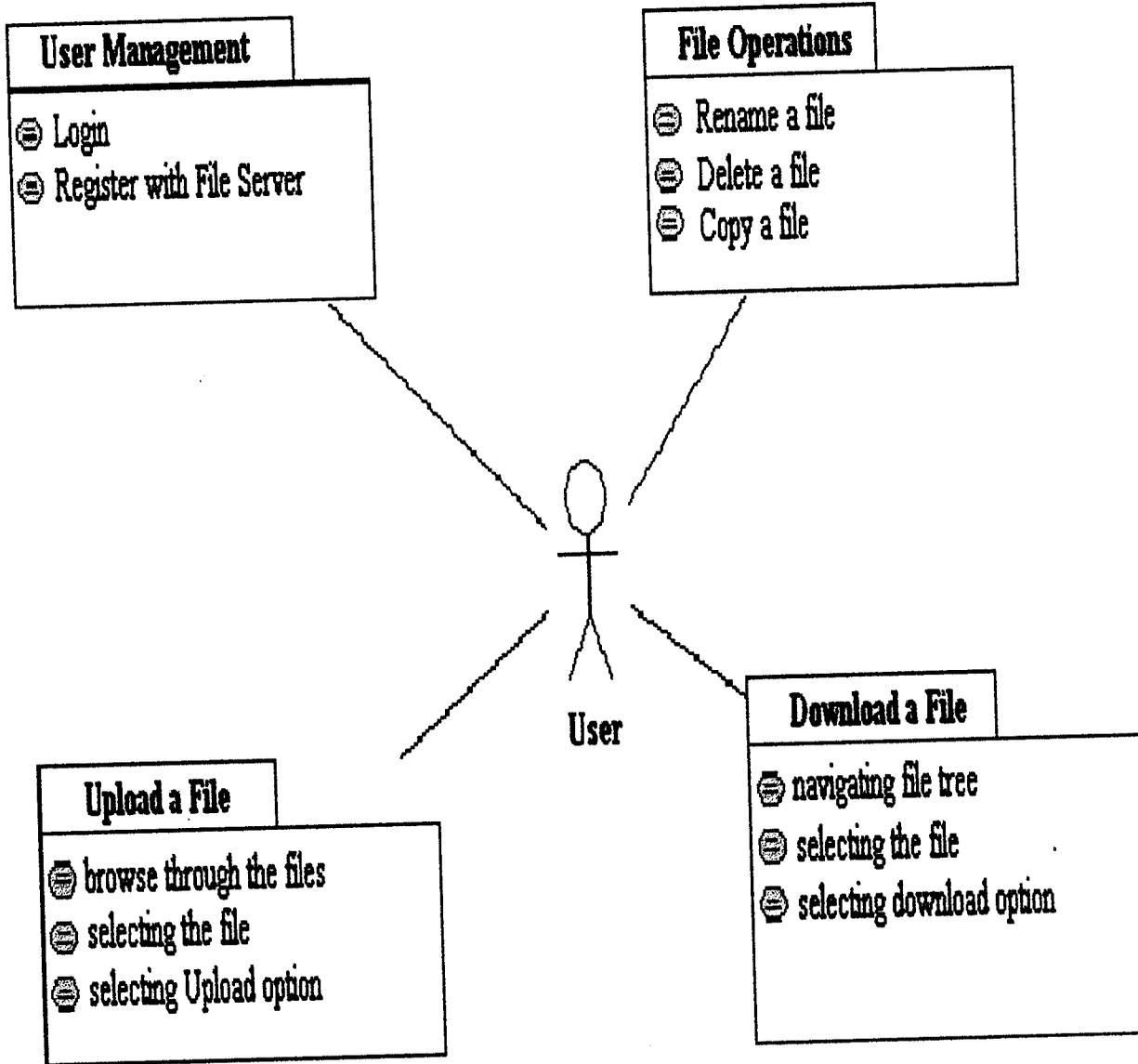
A client input may cause a message to be generated. These messages will in turn cause triggering another message or simply call the appropriate method.

Some of the sample messages are given below,

MESSAGE ID	Description	Action	Result
LOGIN_MSG	User logs in.	Redirects to login Servlet	User logging allowed/denied
COPY_MSG	Copy the file.	Redirects to File Manager	File copied/error
DELETE_MSG	Delete the file.	Redirects to File Manager	User logging allowed/denied
RENAME_MSG	Rename the file.	Redirects to File Manager	File copied/error
UPLOAD_MSG	Upload the file.	Redirects to Upload Servlet	User logging allowed/denied
DOWNLOAD_MSG	Download the file.	Redirects to Download Servlet	File copied/error

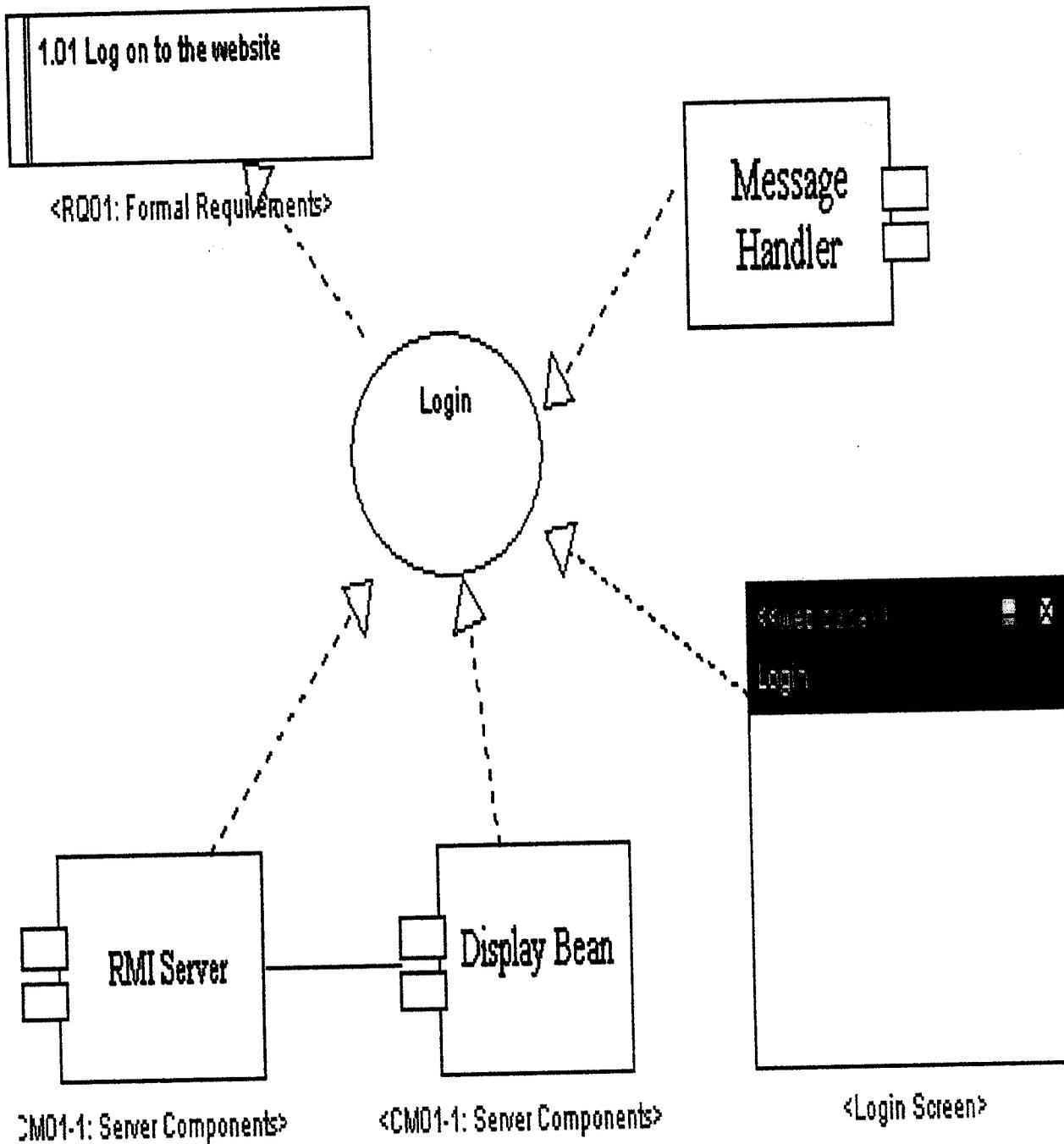
User Input design

The following sketch illustrates the use case model of the user input.



User Input Design

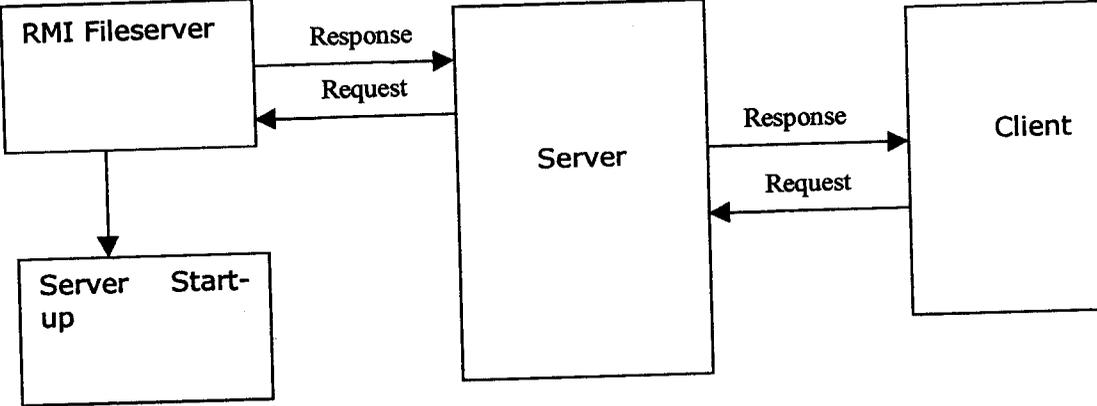
The following sketch implements the design of the use case model for input design.



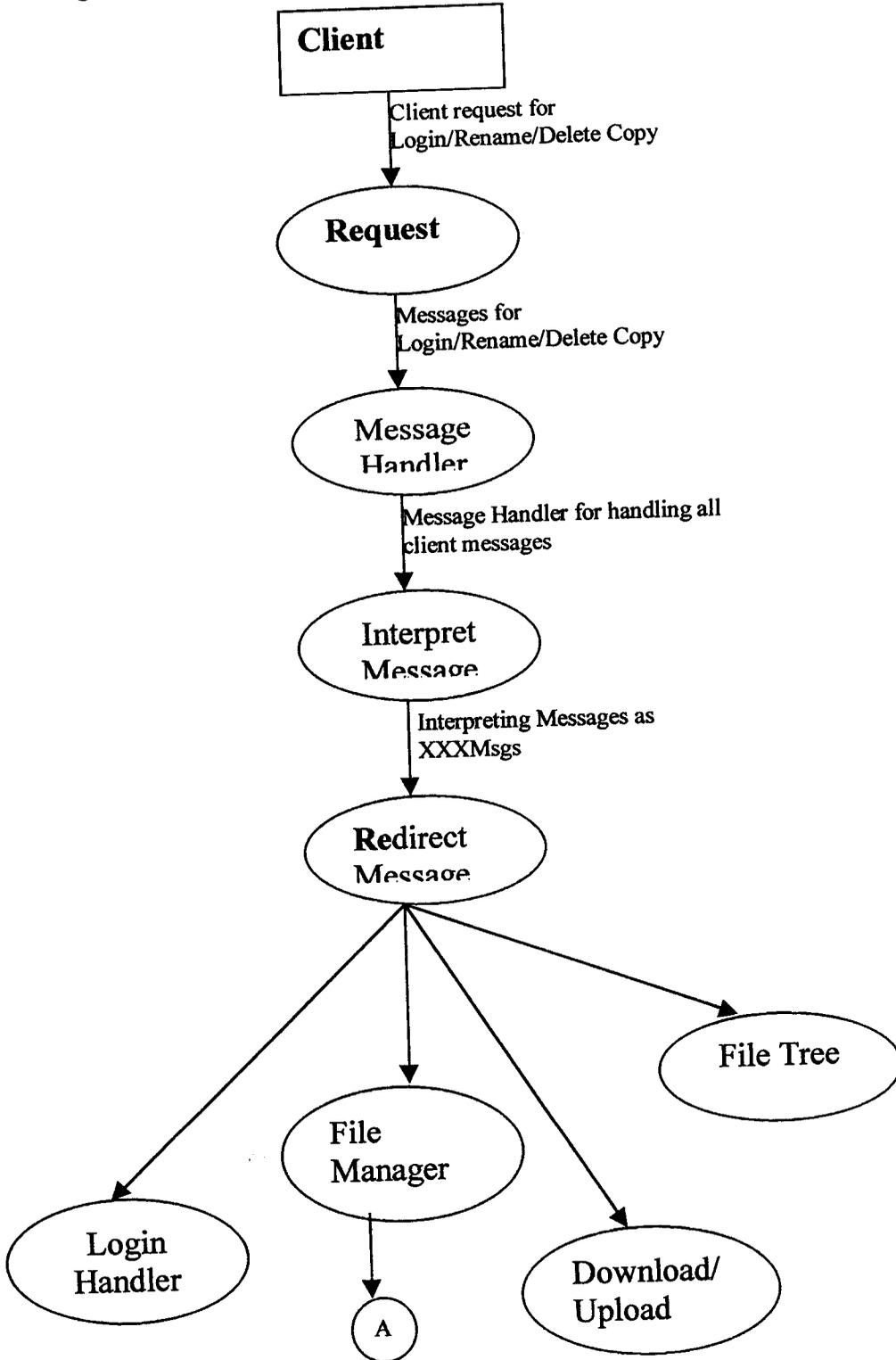
User Input Design Implementation

4.2. Process Design

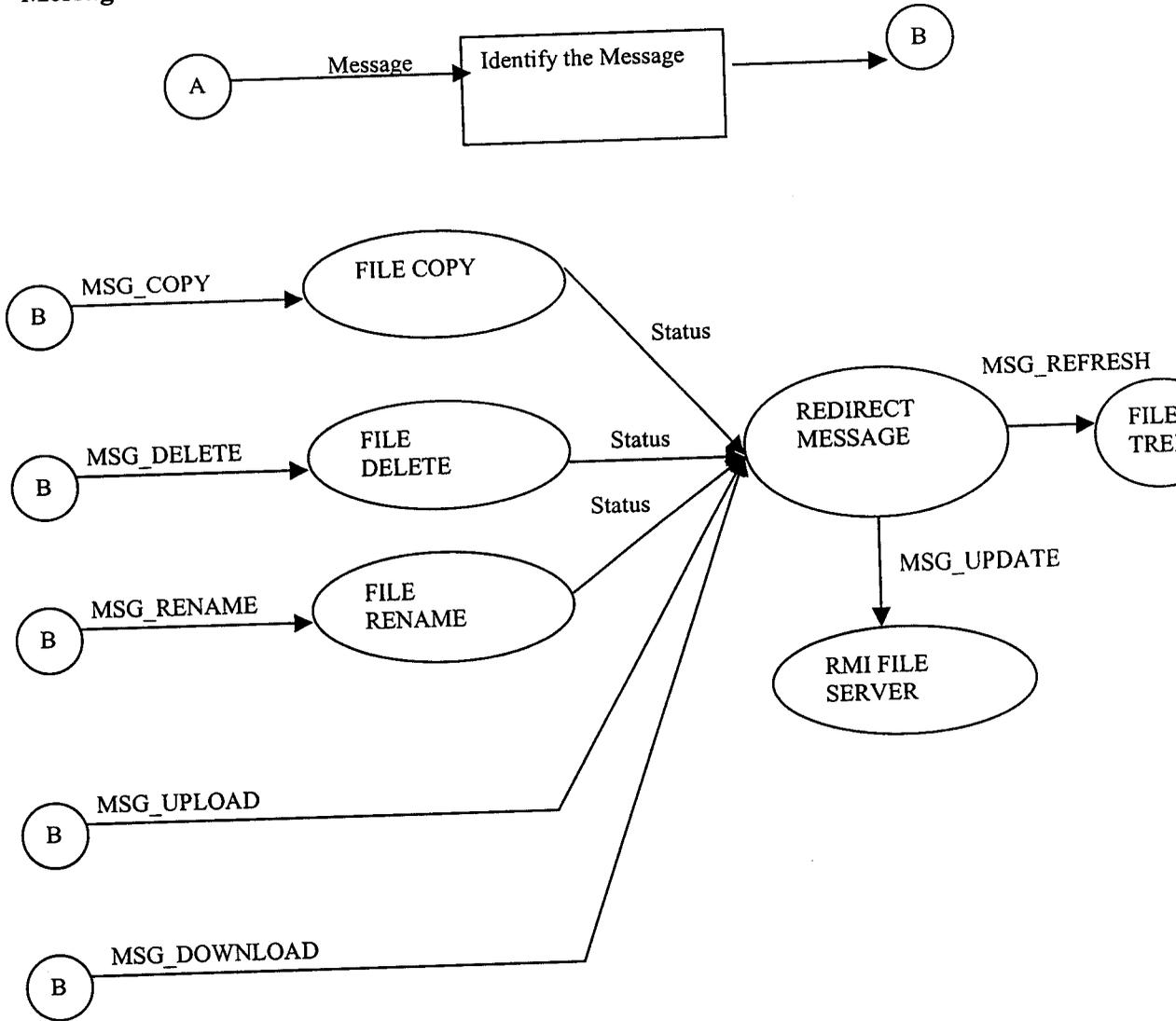
Architecture of the system

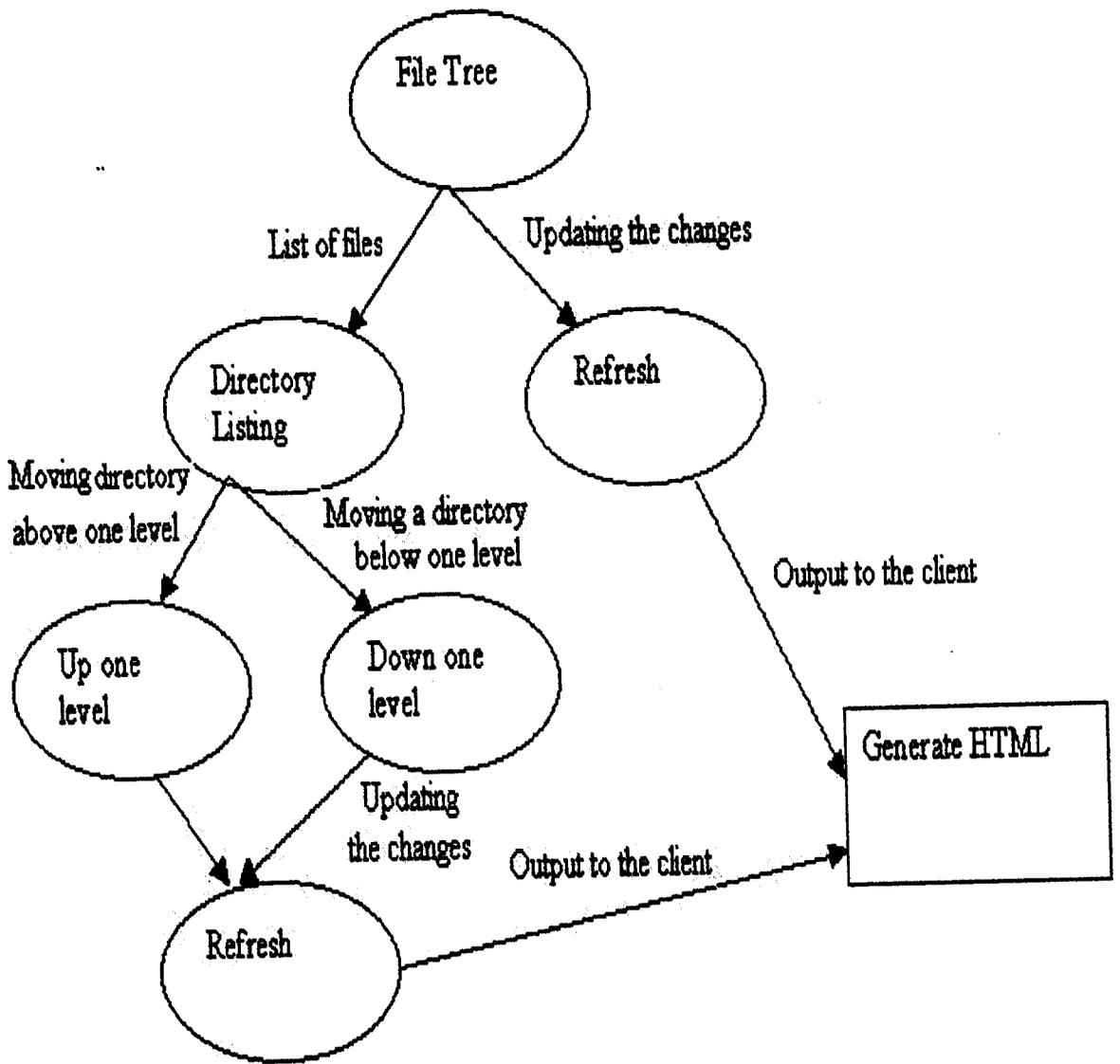


Message Handler

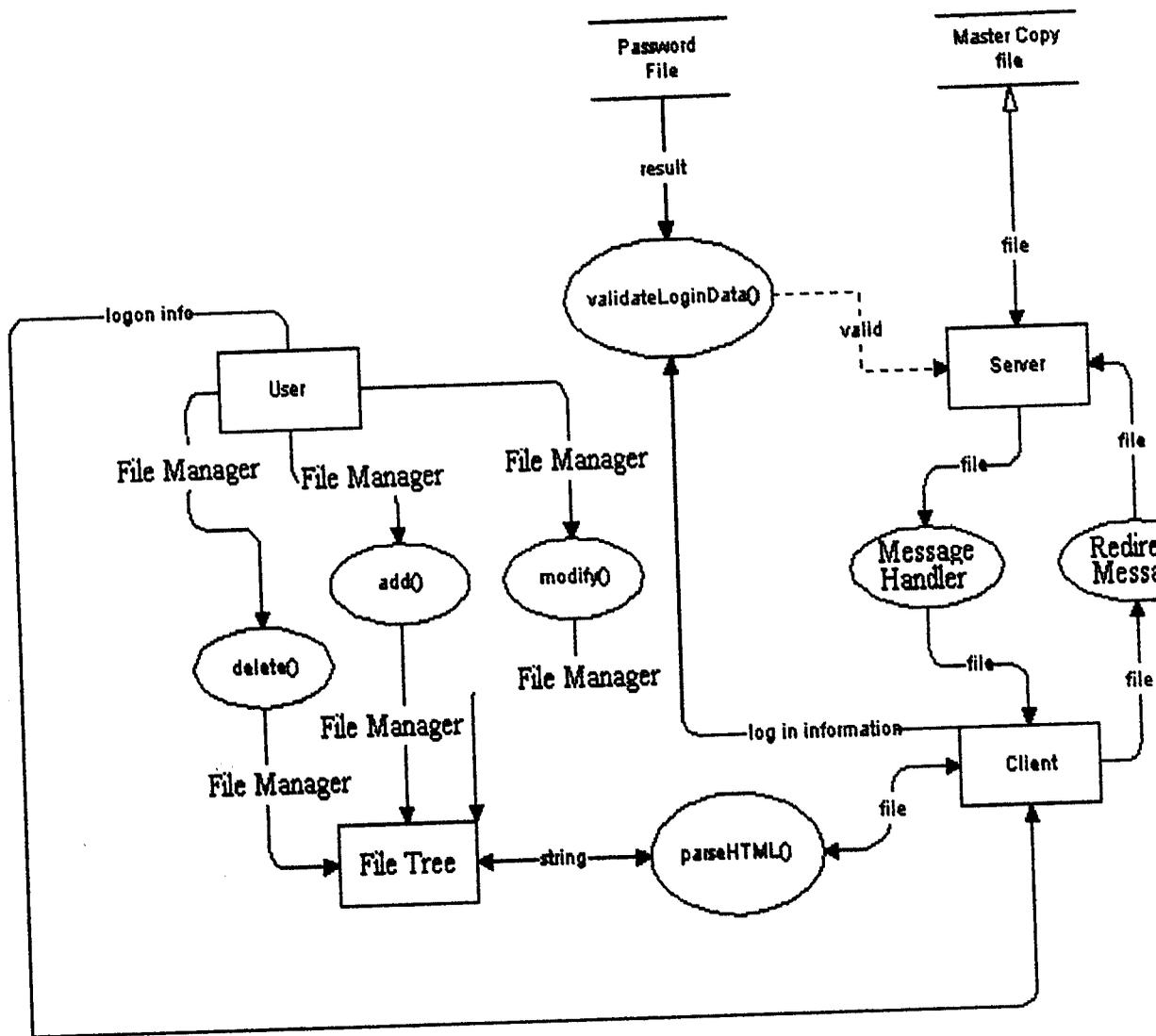


Message Handler

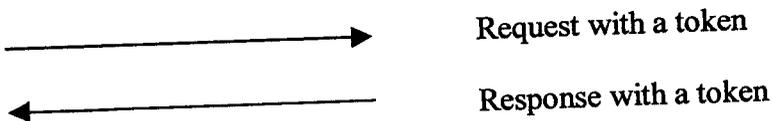
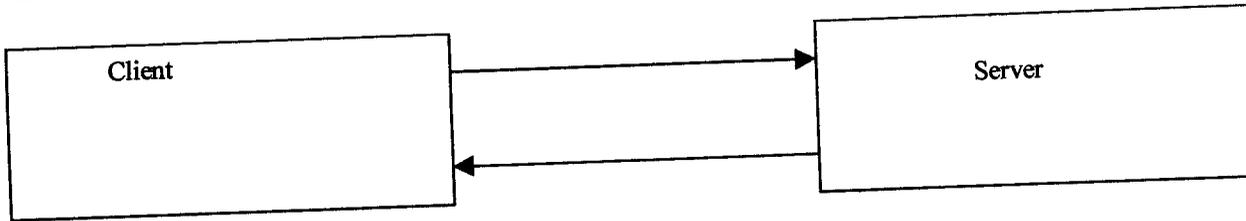




The illustration given below is the data flow diagram of the File Manager Servlet



Session Tracking



4.3. Output Design

Output design is very important since the accuracy and ease of understanding in the outputs are important. The output should be in a suitable format so that the user is fully satisfied with the result. The output should be able to convey a clear message about the status of that system. In this system the user should have the look and feel of accessing the real system.

Remote File Explorer

Remote File Explorer is used to navigate around the file system. The user feels that he is working in his/her real system. This look and feel is taken care of the Java Swing components.

File Manager

File Manager is used to do all the file operations within the file system. The user can copy, rename, delete, upload, and download files. The output design is taken care by the HTML Generator.

4.4. Database Design

The internal data structures used are given below,

The format of the user structure on the disc will be as follows:

- User Name (String)
- Login Name (String)
- User ID (UID) (short)
- Password (String)
- Admin flag to tell if the user has admin rights (1 byte)

After start-up this information is stored in User objects.

The format of the group structure on the disc will be as follows:

- Group Name (String)
- Group ID (GID: short)
- Group Members

After start-up this information is stored in a runtime list

The format of the File structure connects a file to a group and a user. It also contains the rights for a File. The structure on the disc is:

- File name (String)
- File ID (FID: short)
- Group ID (GID: short)
- File owner (UID: short)
- Rights (byte)
- Date of the last modification (int)
- Location of the file (String)

User Log

This file contains Information about user login, logout and all console outputs writings.

This is an ASCII file.

SYSTEM IMPLEMENTATION & TESTING

5. System Implementation & Testing

5.1 System Implementation

A crucial phase in the system development is the successful implementation of the new system. Implementation is the process of converting a new system design into an operational one. This involves creating computer compatible files to store the data, converting the data flow diagrams into coding and documentation.

Implementation is the stage of the project when the theoretical design is turned into a working system. At this stage the main workload, the upheaval and the major impact on the existing practice shifts to the user department. If the implementation stage is not carefully planned and controlled, it can cause chaos. Thus it can be considered to be most crucial stage in achieving a new successful system and in giving the users confidence that the new system will work and be effective. Implementation involves careful planning, investigation of the current system and its constraints on implementation, design of the methods to achieve the changeover, training of staff in the changeover procedures and evaluation of changeover methods. The first task is implementation planning, i.e., deciding on the methods and time scale to be adopted.

5.2 System Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error.

Remote File Accessing System is tested thoroughly by

- a. Unit testing
- b. Integration testing

- c. Stress testing
- d. Validation testing

Unit testing:

Each and every module implemented is thoroughly tested. In this testing each module of Remote File Accessing System is tested completely for interface, local data structures, boundary conditions, independent paths, error handling paths.

The following things are also tested

1. Improper or inconsistent typing
2. Erroneous initialization or default values
3. Incorrect variable names
4. Inconsistent data type
5. Underflow, overflow, and addressing exceptions.

Integration Testing:

Remote File Accessing System is subjected to integration testing. Each and every module of the system after unit testing is integrated and tested for correctness. Integrating step by step each modules and then testing does integration testing. In this system bottom up integration is followed.

Integration testing for Remote File Accessing System

Stress testing:

Remote File Accessing System is tested with stress data the result of the test is good. The results from the test case match the requirement given by the customer. The stress test is conducted with different test cases to evaluate the performance of the system.

Validation testing:

Validation succeeds when software functions in a manner that can be reasonably expected by the customer. Validation for Remote File Accessing System is achieved through a series of black box tests that demonstrate conformity with requirements.

Test cases planned & performed for this system.

Test Number:	1
Purpose:	Ensure that the user can remotely rename a file
Input	It is assumed that the user has valid remote access. The user first clicks the Rename button. Next the user enters the full path of the file to be copied in the client-side text field. The user then presses the submit button.
Output	After the rename button is pressed a text box is prompted to enter the full path of the file to be copied. After the submit button is pressed , the tree window is refreshed to changes
Process	<ol style="list-style-type: none">1. User clicks the Rename button2. The user enters the full path of the file to be renamed e.g. /usr1/bin/changedfilename.dat3. The user then clicks the submit button.4. The process is done in the server. The client receives the output.

Test Number	2
Purpose	Ensure that the user can remotely delete a file
Input	It is assumed that the user has valid remote access. The user first clicks the Delete button. Next the user enters the full path of the file to be deleted in the client-side text field. The user then presses the submit button.
Output	After the delete button is pressed a text box is prompted to enter the full path of the file to be deleted. After the submit button is pressed , the tree window is refreshed to changes
Process	<ol style="list-style-type: none"> 1. User clicks the Delete button 2. The user enters the full path of the file to be deleted e.g. /usr1/bin/deletefilename.dat 3. The user then clicks the submit button. 4. The process is done in the server. <p>The client receives the output.</p>

Test Number	3
Purpose	Ensure that the user has the ability to remotely organize the structure of the File Explorer.
Input	It is assumed that the user has valid remote access. The user clicks on the Directory node.
Output	The File Explorer gets one level down to show the descendants of the directory node.
Process	<ol style="list-style-type: none"> 1. User clicks a directory in the left navigation frame. 2. The descendants of the directory are listed down. 3. in the right side of the frame, the properties of the files are listed. 4. The process is done at the client machine.

Test Number	4
Purpose	Ensure that the program starts properly.
Input	User starts the program.
Output	A login screen appears requesting login name and password.
Process	<ol style="list-style-type: none"> 1. The user enters the proposed system website. 2. A login screen appears requesting login name and password.

Test Number	5
Purpose	Ensure that valid login data is processed correctly by the server.
Input	The login data is used to connect to and access the server.
Output	The FolderTree.html (Applet) file is copied from the server and a message screen is displayed, confirming that access to the specified server was granted.
Process	<ol style="list-style-type: none">1. The user enters the username and password. e.g. Username Scott Password Tiger2. The validation is carried out in the server.3. The server checks for the username and password in the user configuration file.4. The folderTree.html is sent to the client.

Test Number	6
Purpose	Ensure that an invalid login or password denies the user access to the server.
Input	The login data is sent to the server to be verified.
Output	A warning screen is displayed informing the user that access is denied.
Process	<ol style="list-style-type: none">1. The user enters the username and the password. e.g. Username Scott Password badpassword2. The server checks it in the user configuration file.3. The error message is sent.

Test Number	7
Purpose	Ensure that a failed connection of the remote server (RMI File Server) is handled properly.
Input	The login data is used to attempt to connect to the server (RMI File Server).
Output	The client cannot connect to the specified server, and the user is made aware of this with a message screen.
Process	<ol style="list-style-type: none">1. User enters the login data2. the server checks in the user configuration file, whether the RMI File Server is listening.3. If the RMI File Server is not listening then a n appropriate message is flashed.

5.3 Refinements Made On Feedback:

This system is very much useful to the mobile users. The users of this system is of great importance since it is they who ultimately going to use the system. A positive approach was made to all the suggestions made by the technical and non-technical reviews of this project. All the valid suggestions were taken into account and solutions were made found for them. The reasons for discarding the other recommendations were properly examined. Some of the recommendations, which were catered, are examined below.

- The users should have more configuration capabilities. This system as yet, only allowed the user to specify their base root of the system. The user needed to restrict the file types, which they wanted to access from the remote place.
- The users were satisfied with the security level provided by the system. The rigorous tests conducted proved that security performance was good.
- The users suggestion was to have drag and drop method for copying, deleting, uploading and downloading the files.
- Some users suggested of having their files public to all the users in the Virtual Network Neighbourhood.

CONCLUSION & SCOPE FOR FUTURE DEVELOPMENT

6. CONCLUSION & SCOPE FOR FUTURE DEVELOPMENT

Remote File Accessing system is successfully designed and developed at PPP InfoTech Ltd., Chennai. This project was done keeping in mind the fact that it should follow all the steps of the software engineering process and covers the complete software development in life cycle. The user interface provided by this project is widely accepted by the users in general. This project has left the scope of further enhancements wide open as it was developed in an environment that supports Internet widely that in use of the Servlets , RMI. A complete documentation that is provided makes the changes and enhancements that are to be done very easy and provides the vitality of documentation. This system, developed for global need is found to work effectively and efficiently.

The system has been made with a clear abstraction, which will help the future developers, to just add programs rather than modifying it.

BIBLIOGRAPHY

7. BIBLIOGRAPHY

1. Pressman .S. Roger
Software Engineering – A Practitioners Approach
Fourth Edition, McGraw – Hill Book Company, 1990.
2. Patrick Naughton
The Complete Reference Java2
Third Edition, Tata McGraw-Hill, 1999
3. James Goodwin
Mastering Servlets
TechMedia, 1997
4. <http://www.remotefileexplorer.com>
5. <http://www.vnn.com>
6. <http://www.developer.java.sun.com>
7. <http://www.ljavastreet.com>

APPENDICES

Appendix A

**SOFTWARE REQUIREMENT SPECIFICATION
IEEE 830 1985**

I. INTRODUCTION:

1.1 Purpose

The software requirement specification (SRS) provides a detailed analysis of the Product to be developed. It is intended to serve as an agreement between the end user and the product developer. This document explains the specific functionality and attributes required to develop the product.

1.2 Scope

The proposed system is a solution for accessing local computer's file system from any remote computer through the world wide web.

The scope of the system involves

- Setting up a virtual network neighborhood environment.
- Tools for accessing, sharing files across the virtual network or from the local computer.
- Providing authentication, file accessing permissions to the account holder.
- Developing a tool for direct interaction with the virtual network or the local computer.
- Developing a tool (RFE), which will run on the clients local computer and will automatically executed when the system opens a connection with the Internet.
- The prerequisite of this system is that the tool (RFE) on the client's computer should be running in order to handle the request/response from the remote computer.

1.3 Definitions, Acronym and Abbreviation:

- VNN Virtual network neighborhood
- RFE Remote file explorer

- **WWW** world wide web
- **Http** hypertext transfer protocol
- **Http-s** hypertext transfer protocol secure.

VNN

This is called virtual network neighborhood because without having a direct physical connection a user may setup a network by connecting any remote machine through the World Wide Web

RFE

Remote file explorer is a tool, which is used to view the contents of the local computer's file system like Microsoft windows explorer.

1.4 References:

<http://www.vnn.com>

<http://www.remotefileexplorer.com>

2 GENERAL DESCRIPTION:

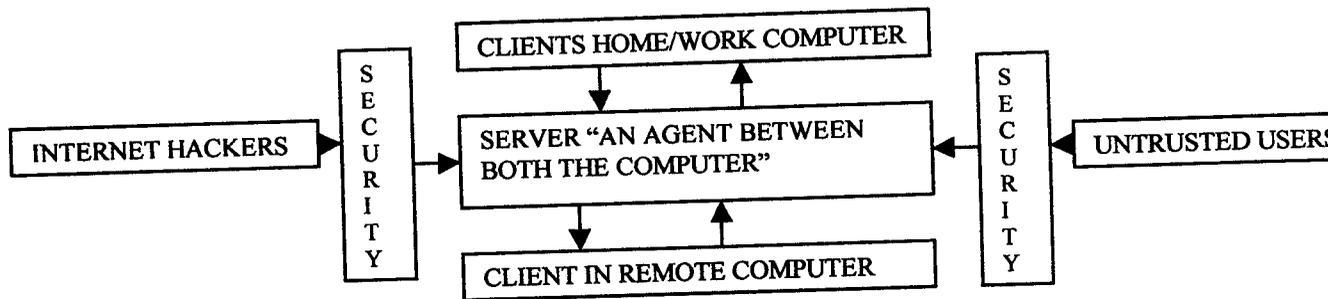
2.1 Product perspective:

There is no existing system. This product is a solution for remote file accessing system. The older protocols such as telnet, ftp were costlier in terms of speed and other parameters. This system is feasible because it is almost like running a web server. When a web server is asked to retrieve a file by specifying a URL it simply sends us a copy of the file. The same way our system will provide the user a copy of the file when a request is made.

The proposed system is a solution for accessing local computer's file system from any remote computer through the World Wide Web. The complexity of today's network and the explosion of information demands for new methods and tools which should make the process of interaction with the network in general and Internet in particular more manageable.

An approach to ease these tasks is to provide the user with a virtual network neighborhood environment. By which a user can form a logical network without having any physical connection and share the information. This system will allow the user to explore his computer's file system and to see the contents of it, for security reasons some of the files are not allowed for reading or modification.

2.2 Product Function:



This is a web-based service based on client server technology. The system walk through will be as follows,

- The user enters the site of the proposed system; he should authenticate himself to prove his identity. This validation will be carried out in the server where the application logic is stored. When the user is identified, the same user's remote machine will wake up to handle the request and Response. It will register its IP address with the server. (Refer the above illustration).

2.3 User characteristics:

The user can at any time form a virtual network with other users. The user will be the administrator of his virtual network and can provide access permission and rights to the user in his network.

2.4 General Constraints:

The RFE running in the remote computer should be in the wake-up state when the user is demanding for files. If the utility (RFE) is not running the other users in his virtual network also become unavailable of the service.

III Specific Requirements:

3.1 Functional requirements:

3.1.2 List of inputs:

- User name
- Password
- Access permission on specific files (a list will be provided during the registration to provide his usage permission)
- Virtual network formation (user has to provide the usernames to share his file system)
- Access permission he is granting to other users.
- IP address of the remote machine where the RFE is running.

3.2 Performance requirements:

3.2.1 Security:

The system will be completely secured since it will be running in the HTTP-S mode. Session tracking will be made so that a user's activities can be traced out to provide more security features. Un trusted users could be identified by this procedure.

3.2.2 Availability:

This system is available to everyone since it is deployed over Internet.

3.2.3 Capacity:

The capacity of the application is purely depends upon the network traffic.

3.2.4 Response time:

Response time depends upon the network traffic, since it is a multi-user system, Multitasking is possible so many users will be accessing the system; according to the priority the system is responded.

3.3 Design constraints:

3.3.1 Hardware limitations:

Since the entire software is developed using java technology it is architecture neutral, but for deployment of the server at least the following configuration is required.

Pentium +, 800 MHz

128 MB RAM

3.3.2 User interfaces, Screen formats:

Mouse, keyboard.

GUI implementation.

3.4 Other requirements:

3.4.1 Operational requirements by user:

- The user has to register.
- The user can enter the proposed system site, the RFE opens a connection and listens for request.
- The user can access his files. (For security reasons some of the files are not allowed for accessing)

Appendix B

SCREENS

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address http://localhost:8080/servlet/FolderTree

Links Best of the Web Channel Guide Customize Links Free HTML Internet Start Microsoft

Rename Delete MakeDir Copy Upload Download Submit

Type	Name	Size
	creditcardv...	43,439
	FileManag...	756
	FileManag...	1,211
	FileManag...	602
	FileManag...	602
	FileManag...	522
	FileManag...	520
	FileManag...	3,411
	FileManag...	5,647
	FileManag...	7,873

- /
- /
- 1999
- Acrobat3
 - Reader
 - ActiveX
 - Browser
 - FONTS
 - HELP
 - Optional
 - PLUG_INS
- Anne'sArk - Working Spec
- Base Peak Journals_files
- CA_LIC
- Design Document Templa
- Dialers
- FileManager
 - FileManager1
- JBuilder3
- JavaWebServer1.1

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address: <http://localhost:8080/servlet/FolderTree> Go

Links: [Best of the Web](#) [Channel Guide](#) [Customize Links](#) [Free Hotmail](#) [Internet Start](#) [Microsoft](#)

Rename Delete MakeDir Copy Upload Download Submit

Type	Name	Size
	creditcardv...	43,439
	FileManag...	756
	FileManag...	1,211
	FileManag...	602
	FileManag...	602
	FileManag...	522
	FileManag...	520
	FileManag...	3,411
	FileManag...	5,647
	FileManag...	7,873

- /
- /
- 1999
- Acrobat3
 - Reader
 - ActiveX
 - Browser
 - FontS
 - HELP
 - Optional
 - PLUG_INS
 - Anne'sArk - Working Spec...
 - Base Peak Journals_files
 - CA_LIC
 - Design Document Templa
 - Dialers
 - FileManager
 - FileManager1
 - JBuilder3
 - JavaWebServer1.1

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address: http://localhost:8080/servlet/FolderTree

Links: Best of the Web Channel Guide Customize Links Free Download Manager Site Microsoft

Rename Delete MakeDir Copy Upload Download Submit

Type	Name	Size
	creditcardv...	43,439
	FileManag...	756
	FileManag...	1,211
	FileManag...	602
	FileManag...	602
	FileManag...	522
	FileManag...	520
	FileManag...	3,411
	FileManag...	5,647
	FileManag...	7,873

- /
- /
- 1999
- Acrobat3
 - Reader
 - ActiveX
 - Browser
 - FONTS
 - HELP
 - Optional
 - PLUG_INS
- Anne'sArk - Working Spec
- Base Peak Journals_files
- CA_LIC
- Design Document Templa
- Dialers
- FileManager
- FileManager1
- JBuilder3
- JavaWebServer1.1

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/servlet/FolderTree> Go

Links [Best of the Web](#) [Channel Guide](#) [Customize Links](#) [Free HotMail](#) [Internet Start](#) [Microsoft](#)

Rename Delete MakeDir Copy Upload Download Submit

Type	Name	Size
	creditcardv...	43,439
	FileManag...	756
	FileManag...	1,211
	FileManag...	602
	FileManag...	602
	FileManag...	522
	FileManag...	520
	FileManag...	3,411
	FileManag...	5,647
	FileManag...	7,873

- /
- /
- 1999
- Acrobat3
 - Reader
 - ActiveX
 - Browser
 - FONTS
 - HELP
 - Optional
 - PLUG_INS
 - Anne'sArk - Working Spec.
 - Base Peak Journals_files
 - CA_LIC
 - Design Document Templa
 - Dialers
 - FileManager
 - FileManager1
 - JBuilder3
 - JavaWebServer1.1

