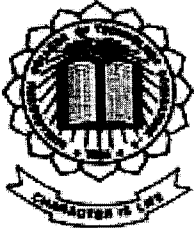


# GUI BASED FILE TRANSFER PROTOCOL USING JAVA



## Project Report

P-1198



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

**Bachelor of Computer Science and Engineering of  
Bharathiar University, Coimbatore.**

Submitted by

**M.Arun kumar**  
0027K0158

**S.Deepan**  
0027K0169

Under the guidance of

**Ms. S. Rajini B.E.,**  
**Senior Lecturer**

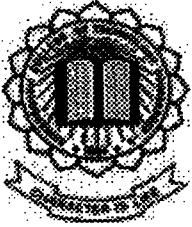
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY,  
COIMBATORE – 641006.

MARCH 2004.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY  
(Affiliated to Bharathiar University, Coimbatore)



**CERTIFICATE**

This is to certify that the project entitled

**GUI BASED FILE TRANSFER PROTOCOL USING JAVA**

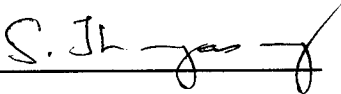
Is done by

**M.Arun kumar**  
**0027K0158**

**S.Deepan**  
**0027K0169**

And submitted in partial fulfillment of the  
Requirement for the award of the degree of the

**Bachelor of Computer Science and Engineering of  
Bharathiar University, Coimbatore.**

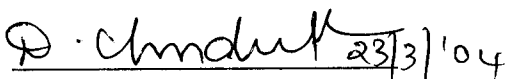


**Professor & Head of the department**  
**(Dr.S.THANGASAMY)**

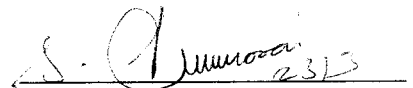


**Project Guide**  
**(Ms.S.RAJINI)**

Certified that the candidates were examined by us in the project work  
Viva voce examination held on 23.03.04.



**Internal Examiner**



**External Examiner**



# DECLARATION

We,

**M.Arun Kumar**      **0027K0158**

**S.Deepan**            **0027K0169**

declare that the project entitled “Adaptive Filtering of Images in Medical Diagnosis”, is done by us and to the best of our knowledge, a similar work has not been submitted earlier to the Bharathiar University or any other institution, for fulfillment of the requirement of the course study.

This project report is submitted on the partial fulfillment of the requirement for all awards of the degree of Bachelor of Computer Science and Engineering of Bharathiar University.

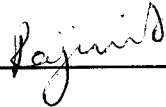
Place: Coimbatore.

Date: 23.03.04

  
[M.Arun Kumar]

  
[S.Deepan]

**Project Guided by**



**Ms.S.Rajini B. E.**

## **ACKNOWLEDGEMENT**

The exhilaration achieved upon the successful completion of any task should be definitely shared with the people behind the venture. This project is an amalgam of study and experience of many people without whose help this project would not have taken shape.

At the onset, we take this opportunity to thank the management of our college for having provided us excellent facilities to work with. We express our deep gratitude to our Principal Dr.K.K.Padmanabhan B.Sc (Engg), M.Tech, for ushering us in the path of triumph.

We are thankful to our beloved Professor and the Head of the Department, Prof.S.Thangasamy B.E. (Hons) Ph.D, whose consistent support and enthusiastic involvement helped us a great deal.

We are greatly indebted to our beloved guide Ms S.Rajini, B.E., Senior Lecturer, Department of Computer Science and Engineering for her excellent guidance and timely support during the course of this project.

We also thank our project coordinator Mrs.D.Chandrakala M.E., and our beloved class advisor Mrs.M.S.Hema B.E., for their invaluable assistance.

We also feel elated in manifesting our deep sense of gratitude to all the staff and lab technicians in the Department of Computer Science and Engineering.

We feel proud to pay our respectful thanks to our Parents for their enthusiasm and encouragement and also we thank our friends who have associated themselves to bring out this project successfully.

## SYNOPSIS

The project entitled “**GUI based File Transfer Protocol using Java**” is the kind of file transferring protocol server. The application aids in easy way of file transfer between any two types of operating system such as Windows NT, Linux, and SunSolarisis etc. The advantages of using the application over FTP are Graphical User Interface, maintaining a complete record log about the file transfers, and better responses even when the network servers are heavily loaded.

The installation and working of this application has been made simple with the help of user friendly forms. But traditional FTP lacks in this aspect, for it requires the users to remember numerous commands.

And by introducing the login procedure, only the authorized users can enter into the system. Any file transfer between two systems of the network can be tracked using the transfer log entries. These aspects improve the security of file transfers in the network.

With this application, any File Transfer request is directed to our own server side program and not to the actual network servers. So better response can be expected even at heavy server loads.

The application’s Graphical User Interface has been designed using Java Swing Components. File sharing between the clients and the server is done with Java Sockets and Java Streams.

# CONTENTS

1.	Introduction	1
	1.1 Company profile	
	1.2 Existing System & Limitations	
	1.1 Proposed System & Advantages	
2.	Software Requirement & Analysis	7
	2.1 Product Definition	
	2.2 Project Plan	
3.	Software Requirement Specification	12
	3.1 Purpose	
	3.2 Scope	
	3.3 Product Overview and Summary	
	3.4 Development & Operating Environment	
	3.5 Functional Specification	
	3.6 Exception Handling	
	3.8 User characteristics	
	3.9 Our client	
4.	System Design	17
5.	System Testing	24
	5.1 Testing Objectives	
	5.2 Testing Principles	
	5.3 Levels of Testing	
6.	Future Enhancements	28
7.	Conclusion	30
8.	Bibliography	32
9.	Annexure	34
	9.1 Sample Code	
	9.2 Output Screens	

---

# **INTRODUCTION**

---



# 1. INTRODUCTION

## 1.1 COMPANY PROFILE:

### *KG Information Systems (P) Ltd (KGISL)*

*“Join Us, Go Places”*

KG Information Systems (P) Ltd (KGISL) is a leader in information technology consulting and services, based at Coimbatore, India. It is located in KG Campus, 12 Kms from the heart of Coimbatore, a one-hour flight south of Bangalore.

Its unique value proposition is that it is both Information Technology (IT) consulting company and also a provider of specialty business process outsourcing (BPO) services. KGISL has a proven track record in delivering a wide range of IT services, thanks to

- World class infrastructure available within the KG campus, a 25 acre, 24 by 7, Self-Sustained “walk-to-walk” techno-park with more than 100,000 sq.ft of office space and high speed fiber-based optic communication links.
- Global software delivery system optimized for the offshore software development model
- Cost effective customized solution delivery capability
- Shorter time-to-market with the ability to deploy big teams.
- Ability to quickly scale up any BPO service offerings (call centers, financial services, payroll processing, engineering design) through its proven HRD capabilities in sourcing, recruiting and training professionals to deliver services that meet accepted global quality standards
  - Coimbatore has 22 engineering colleges, 67 science colleges
  - iTECHCAMPUS, a finishing school at KG campus

- ISO-9001 and ISO-9002 quality certification in software application development and engineering design services capability

### **History and Current Status:**

KGISL was founded in 1994 and it started its commercial operations in 1996-97. Initially it focused on meeting the Y2K remediation needs of fortune 500 companies. Since then, it has successfully delivered cost-effective solutions in the area of Enterprise Resource Planning (ERP), Engineering Design Services and web solutions to clients worldwide.

During 1998-2001, it involved itself in the pioneering effort of establishing and operating the largest medical transcription facility in Asia in partnership with HCR Manor Care, S&P 500 US based healthcare major, employing over 1200 professionals. This entity was spun-off into a separate joint venture in July 2001.

KGISL was awarded the National Award for Exports in 1999-2000 by the Export Promotion Council, a society under the Ministry of Information Technology, Government of India.

### **Markets & Products:**

KGISL primarily caters to the need of companies in the manufacturing, healthcare, and financial services industries, focused primarily at the North America market.

## **1.2 EXISTING SYSTEM AND LIMITATIONS**

FTP was first proposed in 1971 at M.I.T to promote file transfers between the hosts of the M.I.T lab.

Basically there are two types of FTP,

1. Server FTP,
2. Client FTP.

### **Server FTP:**

This FTP is mainly used for web applications. Here file transfer takes place between web clients & FTP servers'. Using Hypertext Transfer Protocol (HTTP), web clients' requests files from web servers. Here File transfer service goes both ways, up to the server from the web clients as well as down to the clients from the FTP server. FTP Servers in web runs its own FTP software and any browser, which speaks http acts as the client software.

Any user from a web client can access the FTP server with the help of authenticated user id and password. Anonymous user id aids in file transfers without an authenticated User id, but with limited transfer rights. FTP manager is a person intermediate to FTP server and web clients, who checks for viruses or worms before uploading files to FTP server.

### **Client FTP:**

This type of FTP is used in small network like LANs, WANs etc. It uses Client/Server architecture to transfer files across the network. File transfer request from any system in the network is directed to the FTP server side program which acts as a bridge between the two systems of file transfer. In most FTP implementations the network server acts as the FTP server. FTP server side program controls over rest of the file transfer process.

Cross Platform file transfer is generally achieved by two ways.

- By using a network protocol, which controls the file format conversion specific to different platforms.
- By using specific implementation techniques that can trigger file format conversions in the source or destination platform.

### **Limitations of client FTP:**

- When the network server is heavily loaded with applications, it reflects with very bad response to requests made to it. FTP server side software generally runs on the network server, which may lead to very bad responses reflected to file transfer requests. Cross platform file transfers may lead to even more worse response when network servers of both the platforms are heavily loaded.
- The existing system File Transfer Protocol is command based and the user must have the knowledge of FTP commands in detail. It is not interactive or user-friendly.
- The acknowledgement for transferred file is unknown.
- There is no provision for a tracking system which can track file transfer information such as list of files transferred, date of transfer, number of bytes transferred, source machine address from where file has been transferred, destination machine address to which file has been transferred.

### **1.3 PROPOSED SYSTEM AND ADVANTAGES**

The aim of the system proposed is to overcome the limitations of the existing FTP softwares used to transfer files with in small networks such as LAN, WAN etc.

The system should provide facility to transfer files between systems of different platform. This is achieved by implementing the whole system in JAVA. The system should update itself with the network to identify and list all the active machines of the network. It should maintain separate user rights (administrator and ordinary users) and it should provide them appropriate limitations of using the application. Thus the system proposed should overcome limitations such as Graphical User Interface, File transfer tracking system, better response on heavy network loads.

**Advantages:**

The system proposed has been planned to provide,

- Better user interface with the help of GUI, so that the users need not remember all the FTP commands.
- Better response for the file transfer requests made, even when the network server is heavily loaded.
- Provision to track the files transfer information including the date of transfer, source and destination machine, and number of bytes transferred.
- Acknowledgement provision to check if the file has been successfully transferred.
- Better user friendliness in working and installation.
- Platform independence in system implementation.

---

# **SOFTWARE REQUIREMENTS ANALYSIS**

---

## **2. SOFTWARE REQUIREMENTS ANALYSIS**

System study is an activity that encompasses most of the tasks that we have collectively called computer system engineering. System study is conducted with the following objectives

- Identify the needs.
- Evaluate the system concept for feasibility.
- Perform economic and technical analysis.
- Allocate function to hardware, software, people and other system elements.
- Create a system definition that forms the foundation for all subsequent Engineering works

### **2.1 PRODUCT DEFINITION**

“GUI based FTP using Java” is developed in order to enhance the existing command based File Transfer Protocol and to overcome the disadvantages faced by it. The primary target is to develop a GUI for the existing system to make it user-friendly, a file transfer information tracking system, to bring in better response on heavy network loads and to transfer files across different platform.

### **2.2 PROJECT PLAN**

The whole application shall be implemented in Java to facilitate platform independent file transfer.

The system requires the following aspects to be planned in detail,

- Connection-Establishment
- Graphical User Interface (GUI)
- File-Listing

- File-transfer module
- Transfer Log

### **Connection Establishment:**

This module deals with establishing connection between the server and the client. First the IP address of server is identified. Using this IP address and port number, the client tries to connect itself with the server and it waits for the acknowledgement back from server

The server already should be waiting for the client in the same port. As soon as the server receives the request from client, connection gets established for data transfer between the server and client.

After the above process, the client and server can exchange any information between them including control and data information. So, before any other module, this connection module should be done successfully.

### **Graphical User Interface:**

This module should bring user authentication and user friendliness.

The above-mentioned two aspects have been planned to be achieved with the help of user friendly forms on the following aspects,

- User Authentication

It requests login id and password from the user. It identifies the user to be an administrator or an ordinary user, and the following forms appears depending upon the type of the user,

- Administrator form,
- Ordinary user form.

- Administrator functionality

The administrator form has the following options,

- Creating new user (administrator or an ordinary user).
- Deleting existing user,
- Changing own password,



- Viewing logs information (information on file transfers that have been recently made).
- Ordinary user functionality
  - Ordinary user form has the following operations,
    - Changing own password.
- File transfer form
  - This is the main form to be used by the application. It has the following options,
    - Active network systems listing,
    - Remote file listing,
    - File transfer options,
    - Utility menu bar.

### **File-Listing:**

It involves two aspects,

- Remote file listing
- Source file listing

### ***Source- file listing***

It involves listing of files of the local machine. The directory path shall be got from the user and depending on the path, the files in that directory shall be listed.

### ***Remote- file listing***

It involves listing the files of remote machine are listed. First the host name and directory of machine is sent to server. After establishing connection with the remote machine, the server will retrieve the remote file listing information which shall be listed in the source machine.

### **Transfer- module:**

Here in this module we concentrate over transferring files. The following opinions shall be facilitated,

- Source to remote machine
- Remote to source
- Deletion of files (both in source and remote machines).

**Transfer Log:**

Any file transfer should take place through the server. So server side program should be implemented so that it makes log entries with the database whenever a file transfer is made. The log entries should include source machine and destination machine addresses, date and size of file transfers.

---

# **SOFTWARE REQUIREMENTS SPECIFICATION**

---

## **3. SOFTWARE REQUIREMENTS SPECIFICATION**

### **3.1 PURPOSE**

The purpose of the project is to develop an application that facilitates transferring files across different platform providing better performance capability than that of the present FTP systems. Existing system has certain drawbacks such as bad request responses on heavy server loads, absence of user friendly GUI and absence of file transfer tracking system.

### **3.2 SCOPE**

The scope of the planned system is limited to only those operating systems, which can support Java 2 or any version ahead of it. The system where our server side program runs should support Oracle8i or any version ahead of it.

### **3.3 PRODUCT OVERVIEW AND SUMMARY**

Our product aids in transferring any type of file between two systems of networks such as LANs and WANs. The network may involve a single platform or more than one platform. The product should support user friendly Graphical user interface, file transfer information tracking facility, and should provide better response even when the network servers are heavily loaded.

The product should also provide user authentication, and administrator capability of monitoring product usage by the ordinary users. Active clients (platform independent) of the network should be automatically listed and should be updated with the network.

### **3.4 DEVELOPMENT AND OPERATIONAL ENVIRONMENT**

The development environment gives the minimum hardware and software requirements.

#### **Hardware Specification**

- Processor                      Pentium III
- RAM                              64 MB
- Cache                            128MB
- Hard Disk                       10 GB
- Floppy Drive                    1.44 FDD
- Monitor                         14" Monitor

#### **Software Specification**

- Operating System                Platform independent.
- Front-End                         Java 2 or any version ahead
- Back-End                         Oracle 8i or any version ahead

### **3.5 FUNCTIONAL SPECIFICATION**

The system involves the following modules,

#### **Developing GUI**

This module involves developing user-friendly forms with the help of java Swings. Swings are lightweight components so they have better look and feel while compared to the previous AWT components.

***Login Form:*** This form authenticates valid user with the help of user id and password. It also identifies the user to be ordinary user or administrator.

***Administrator Form:*** This form allows the administrator to do operations like creating new user, deleting existing user, changing administrator password, viewing file transfer log information.

***Ordinary user Form:*** This form allows the ordinary users to change their own password.

***File transfer Form:*** This form lists out information like Source machine file listing, destination machine file listing, Active machines listing, source to destination machine file transfer option, destination to source machine file transfer option, viewing transfer log option, disconnecting remote machine option etc.

### **File Transfer module:**

***Source client program:*** This program is the client program running which provides GUI user interface and instigates any file transfer. For both source to destination and destination to source file transfers this program communicates with the server side program and requests for file transfer providing it the required information.

***Server side program:*** This program acts as a bridge between the source and remote machines involved in file transfers. On the request of source machine client program, it retrieves required information from the remote machine client program and passes it to source machine client program. It controls the log information entry to the database.

***Remote client program:*** This program retrieves and passes the required information to the server program on its request.

## **Database Connectivity:**

*Client database:* This database maintains the ip address and host name of all the active clients of the network.

*Login database:* This database maintains user id, password and administrative level of the users of the application.

*Log database:* This database maintains the file transfer log information including name, size, time, source and destination of files transferred.

All the above databases are controlled by the server side program.

## **3.6 EXCEPTION HANDLING**

The project has to handle all sort of exceptions that arises due to wrong user input entry, incomplete file transfers, file formats which are not supported operating system platforms. Transfer request made to inactive clients and servers etc.

## **3.7 USER CHARACTERISTICS**

The users though have little knowledge on limited FTP commands, they are not aware of the complete set of commands.

## **3.8 OUR CLIENT**

Our product is directed to “**KG Information Systems (p) Ltd**”, Coimbatore.

---

# **SYSTEM DESIGN**

---



## **4. SYSTEM DESIGN**

### **4.1 FUNDAMENTAL DESIGN CONCEPTS**

The objective of design is to get the program right. The following concepts must be considered while designing any software:

#### **Levels of abstraction:**

In modular design, the levels of abstraction method are used. In the highest level, the solution is stated and in each lower level, solution step is elaborated. At the lowest level, solution is stated in the manner that can be directly implemented.

#### **Stepwise refinement:**

This is a top down methodology. The architecture of program is developed by successively refining levels of procedural detail. A hierarchy is developed by decomposing the procedural abstraction in a stepwise fashion until programming language statements are reached. Refinement is actually a process of elaboration.

#### **Modularity:**

Software is divided into different modules that are integrated. Since the monolithic software is difficult to grasp, we need to decompose them into modules. But care should be taken when modularizing.

#### **Data structures:**

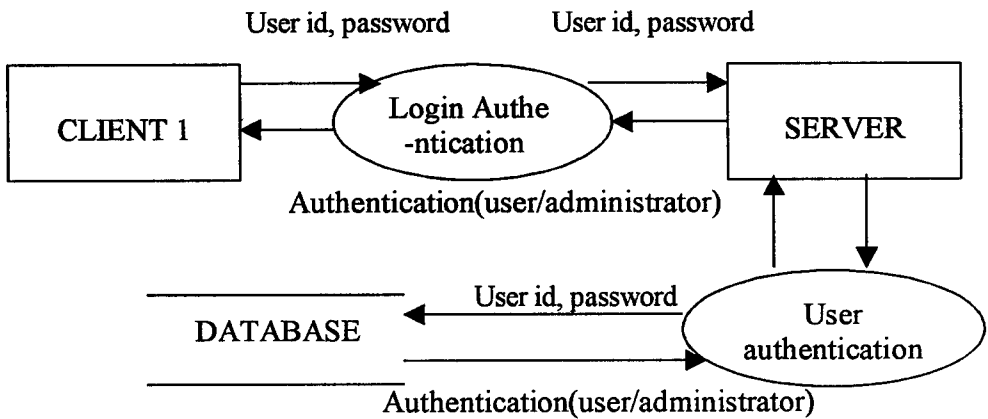
Software designed must consider the data structures that are required in the modules.

#### **Information hiding:**

Modules may hide some part of the information so that the hidden part is inaccessible to other modules, thus modular design reduces complexity and results in easier representation

## 4.2 DESIGN AND AUTHENTICATION MODULE

The GUI login form in the client machine obtains user id and password from the user and communicates it to the server, the server checks the validity of the user with the database and authenticates him/her as administrator or ordinary user.



**Fig a. LOGIN AUTHENTICATION**

### 4.3 DESIGN AND ADMINISTRATIVE OPERATION MODULE

Once the user has been authenticated as an administrator, he can perform administrative operations such as creating new user, Deleting existing users , viewing file transfer log and changing own password.

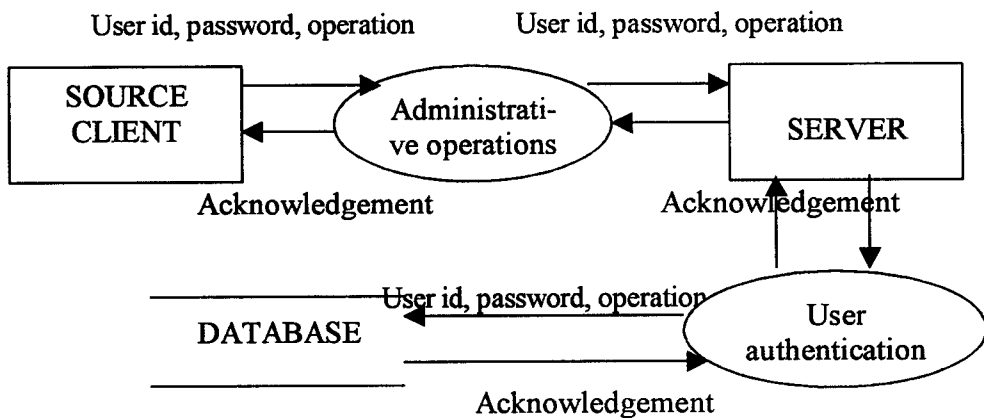


Fig b. ADMINISTRATIVE OPERATION

Administrative Operations: Creating new user, Deleting existing users, viewing file transfer log and changing own password.

#### 4.4 DESIGN OF REMOTE FILE LISTING MODULE

The source machine file listing is very simple and needs no server communication. But on listing the remote machine files, source machine sends remote machine host name and file path to server and the server retrieves the file listing from the remote machine and returns to the source machine.

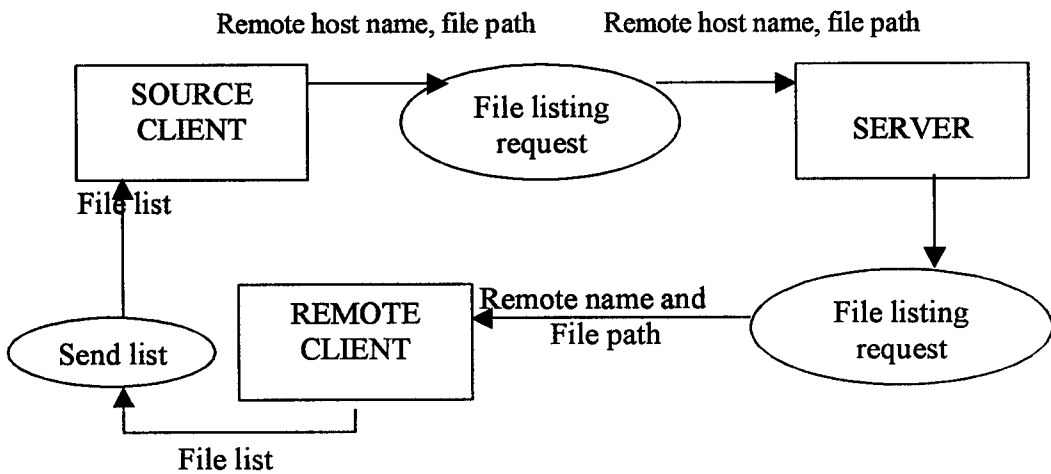


Fig c. REMOTE FILE LISTING

#### 4.5 DESIGN OF SOURCE TO REMOTE FILE TRANSFER MODULE

Once the source and remote machine files are listed, file transfers can be easily facilitated. When the user makes a source to destination file transfer request, we send the destination file path and file data to server through the sockets. The server sends the same to the remote machine, where new file is created with the destination file path and file data.

Log entry involving the file transfer information is made to the database by the server. Socket programming and stream programming of Java can be used to facilitate this file transfer.

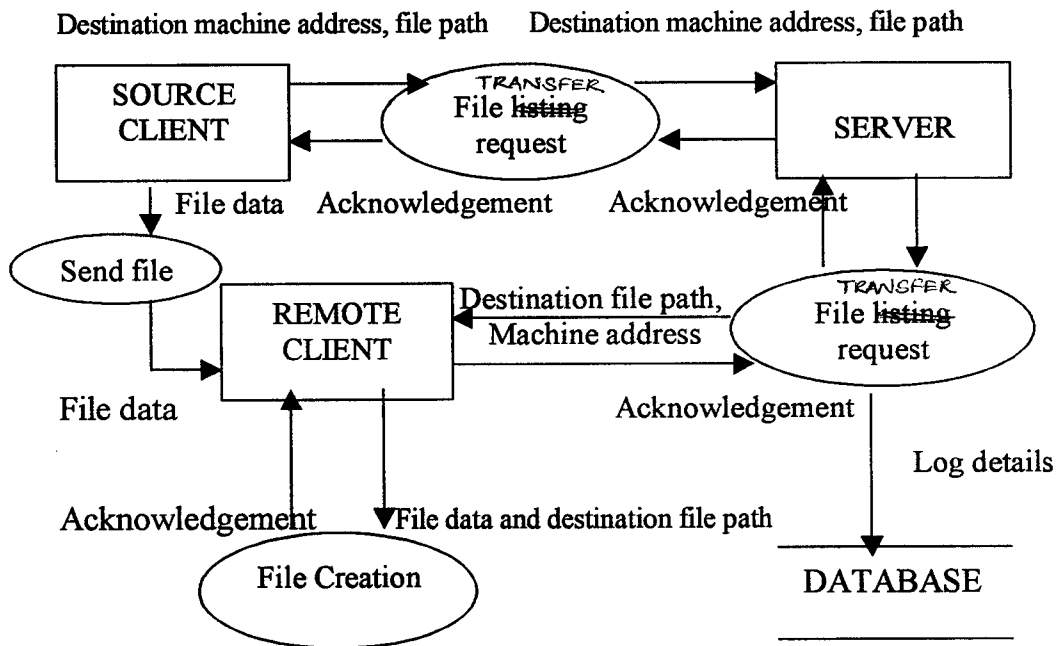


Fig d. SOURCE TO REMOTE FILE TRANSFER

#### 4.6 DESIGN OF REMOTE TO SOURCE FILE TRANSFER MODULE

After listing the source and remote machine files, if the user makes a destination to source file transfer request, we send the remote machine host name and destination file path to server through the sockets. The server sends the same to the remote machine, the remote machine sends back the file data to the server and the server sends it back to source machine.

Log entry involving the file transfer information is made to the database by the server.

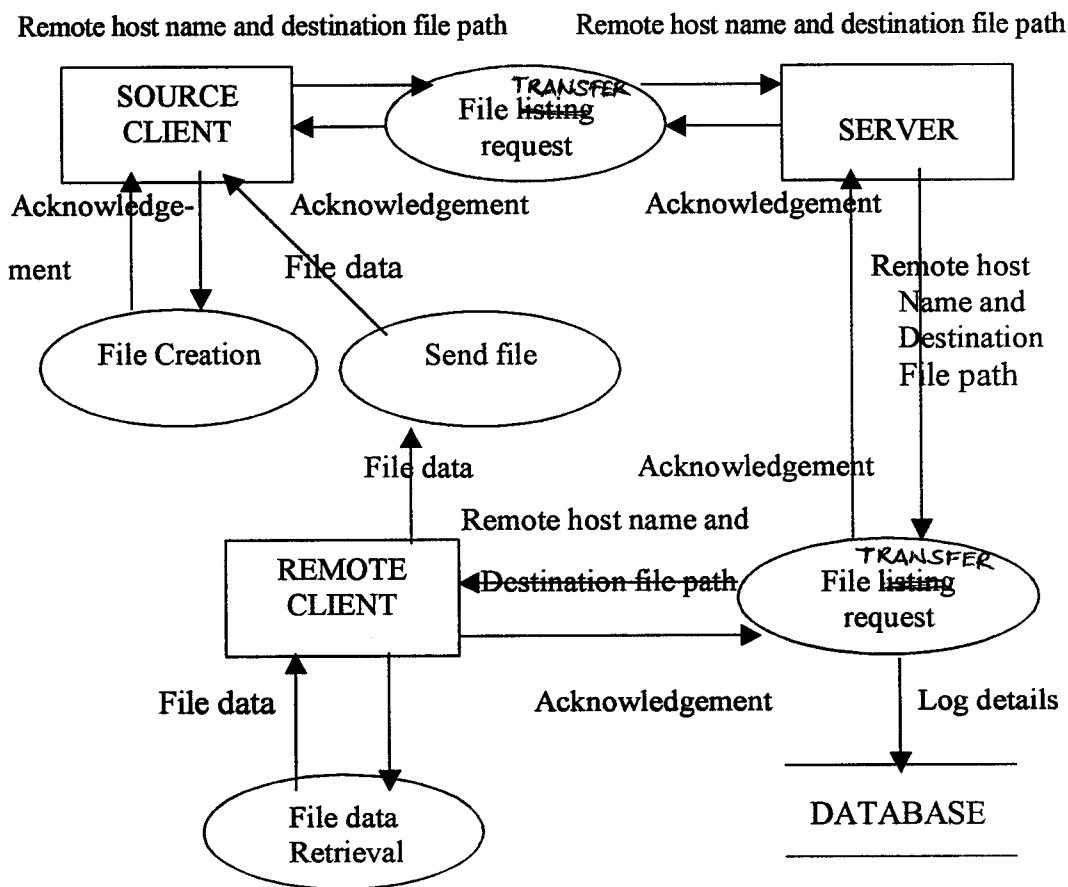


Fig e. REMOTE TO SOURCE FILE TRANSFER

---

# **SYSTEM TESTING**

---

## **5. SYSTEM TESTING**

Testing is an activity to verify that a correct system is being built and is performed with the intent of finding faults in the system. Testing is an activity, however not restricted to being performed after the development phase is complete. But this is to be carried out in parallel with all stages of system development, starting with requirement specification.

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

The major aspects that are brought in to attention for testing purpose are,

- Test for proper authentication of users,
- Test for various Administrator and ordinary user operations,
- Test for transfer of files of different formats,
- Test for file transfer across different platforms,
- Test on how application responds for inactive remote clients
- Test on how fast the system responds on heavy network loads.

### **5.1 TESTING OBJECTIVE**

Testing of all the above aspects have to be carried out with an intention to find all errors that can occur on real time situations. This can be achieved by running the implemented programs with different test case data that can identify all real time error situations.



## **5.2 TESTING PRINCIPLES**

Few of the testing principles that have been followed are,

- All tests should be traceable to customer requirements
- Testing should begin “in the small” and progress towards testing “in large”. The focus of testing will shift progressively from programs to individual modules and finally to the entire project.
- A good test is not redundant.
- A good test should be neither too simple nor too complex.

## **5.3 LEVELS OF TESTING**

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

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

### **Unit Testing**

Unit testing has been carried out to verify and uncover errors within the boundary of each module. Each module was tested separately after its completion, with valid and invalid data. In this testing step, each module was found to be working satisfactory as per the expected output of the module.

## **Integration Testing**

Integration Testing address the issues associated with the dual problems of verification and program construction. When the software was integrated from lower level modules to higher-level modules, a set of higher-order tests was conducted. The main objective in this testing process was to take unit-tested modules and build a program structure that had been dictated by design.

## **Output Testing**

Here the whole system was checked with sample runs and the complete functionality of the system was checked so that the requirements were met. Each aspect of security, user interface as well as system interface was checked to be error free.

---

**FUTURE ENHANCEMENT**

---

## **6. FUTURE ENHANCEMENTS**

Time is the best healer. Things are to be made better over time. A successful system should always accommodate future changes in order to enhance its performance. The application developed is so flexible that it can be enhanced so easily without the need for major changes to the present system.

Our ideas on future enhancements are,

- File transfer progress indicator can be added to the system
- Application status indicator can be added to the system
- Further options like create directory, delete directory, refresh , pause and resume file transfer operations can be added to the system
- Active clients can be listed separately categorizing them based on the platform they belong to.
- Multithreading concept can be used to facilitate more than one file transfer at the same time.
- Database can be structured with minimum redundancy.
- Remote to remote file transfer facility which can be instigated from the server can be added to the system

---

# CONCLUSION

---

## **7. CONCLUSION**

The complete design and development of the system “GUI based FTP using Java” is presented in this dissertation. A good amount of user friendly features have been incorporated in this system and it is possible for any novice user to use the system to the maximum benefit.

The programming techniques used in the design of the system provide a scope for further expansion and implementation of any changes, which may occur in the future. Maximum care and concentration has been focused to troubleshoot this project. This design can be applied for any organization of similar circumstances.

Since, the requirements of any organization and their standards are changing day by day, the system has been designed in such a way that its scope and boundaries could be expanded in the future with little modifications.

The main aim behind the development of this package is to provide a comprehensive solution that is capable of handling and meeting the company’s stated and implied requirements.

---

# **BIBLIOGRAPHY**

---

## **8. BIBLIOGRAPHY**

1. Deitel & Deitel, "Java How to program", Third Edition, 1999.
2. Patrick naughton & Herbert schildt, "The Complete Reference Java 2", Fifth Edition, 2002.
3. Cay S.Horstmann & Gary Cornell, "Core Java – 2"-Volume 1", First Edition, 1999.

## **WEBSITES VISITED**

1. [www.javaforum.com](http://www.javaforum.com)
2. [www.sunjava.com](http://www.sunjava.com)



---

# **ANNEXURE**

---

## 9. ANNEXURE

### 9.1 SAMPLE CODE

#### Implementation of Remote machine file listing module:

#### *Client side program requesting server side file listing service:*

```
try
{
    Socket cli = new Socket(serverIP,1970);
    PrintStream dout = new PrintStream(cli.getOutputStream());
    DataInputStream din = new DataInputStream
(cli.getInputStream());
    dout.println("remoteFiles");
    dout.println(sel);
    String tem = new String();
    for(int i=0;i<k;i++)
        {tem = tem.concat("a");}
    tem=tem.trim();
    dout.println(tem);
    for(int i=0;i<k;i++)
        {data[i] = din.readLine();}
    din.close();
    dout.close();
    cli.close();
    remotep9.setListData(data);
    remotep9.setSelectedIndex(0);
    remotesys.setText("Remote Files : " + sel);
    remotel.setEnabled(true);
    sremotel = "Remote System " + sel;
    remotel.setText("Remote System " + sel);
}
catch(IOException ioe)
{
    card.show(p,"ConnectionFail");
    setSize(500,400);
    logo.setBounds(1,0,700,100);
    p.setBounds(1,100,700,300);
removeMenubar();
Point point = new Point(150,100);
setLocation(point);
setResizable(false);
setVisible(true);
validate();
}
```

***Equivalent Server side implementation which requests back the remote machine client program for file listing.***

```
if(param1.compareTo("remoteFiles") == 0)
{
    String lName = din.readLine();
    String nums =din.readLine();
    nums = nums.trim();
    r = stat.executeQuery("select * from clients");
    String rm = new String();
    while(r.next())
    {
        s1 = r.getString(1);
        s2 = r.getString(2);
        if(s1.compareTo(lName) == 0)
        {
            rm = s2;
        }
    }
    try
    {
        try
        {
            Socket cli2 = new Socket(rm,1965);
            PrintStream dout2 = new PrintStream
            (cli2.getOutputStream());
            DataInputStream din2 = new DataInputStream
            (cli2.getInputStream());
            dout2.println("remotefiles");
            dout2.println("c:");
            dout2.println("welcome");
            String tt = new String();
            System.out.println(nums.length());
            for(int i=0;i<nums.length();i++)
            {
                tt = din2.readLine();
                dout.println(tt);
                tt="";
            }
            din2.close();
            dout2.close();
            cli2.close();
        }
        catch(NoRouteToHostException rtoh){}
    }
    catch(ConnectException ce){}
```

```
}
```

### ***Equivalent remote client file listing service program***

```
if(i==7)
{
    try
    {
        String hai=inp.readLine();
        System.out.println("hello hi");
        String wastel=inp.readLine();

        File f1=new File(dirname);
        if(!(f1.exists()))
        {
            dirname = new String("/");
        }
        f1=new File(dirname);
        String s12[] = f1.list();
        String ss=new String();
        System.out.println(s12.length);
        for(int i12=0;i12<s12.length;i12++)
        {
            ss = dirname.trim() + "/" + s12[i12];
            File f = new File(ss);
            if(f.isDirectory())
            {
                outp.println("/".concat(s12[i12]));
            }
            else
            {
                outp.println(s12[i12]);
            }
            ss="";
        }
    }
    catch(NullPointerException ee){}
}
```

### **Implementation of Active Clients listing**

#### ***Server side implementation***

```
import java.io.*;
import java.sql.*;
import java.net.*;
```

```

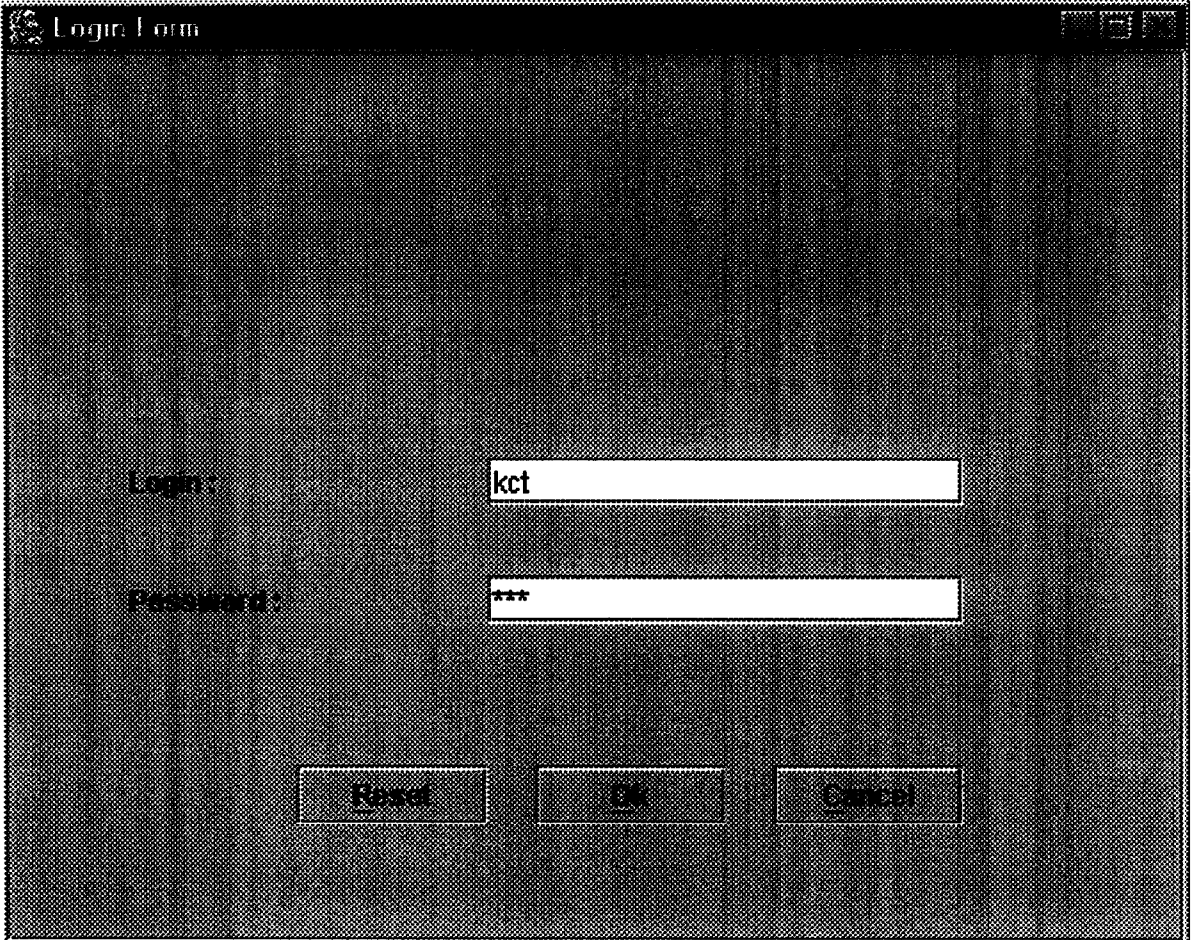
class server2
{
    public static void main(String a[])throws Exception
    {
        while(true)
        {
            ServerSocket ser =new ServerSocket(1998);
            Socket cli;
            cli=ser.accept();
            System.out.println("im accepted");
            DataInputStream di = new DataInputStream
            (cli.getInputStream());
            String str =di.readLine();
            String str1 =di.readLine();
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection con = DriverManager.getConnection
            ("jdbc:odbc:red","cse43","cse43");
            Statement stat = con.createStatement();
            ResultSet r;
            try
            {
                stat.executeQuery("insert into clients values(" + str
                + "," + str1 + ")");
            }
            catch(SQLException sqle)
            {
                System.out.println(sqle);
            }
            try
            {
                stat.executeQuery("commit");
            }
            catch(SQLException sqle)
            {
                System.out.println(sqle);
            }
            cli.close();
            ser.close();
        }
    }
}

```

### *Equivalent client side implementation*

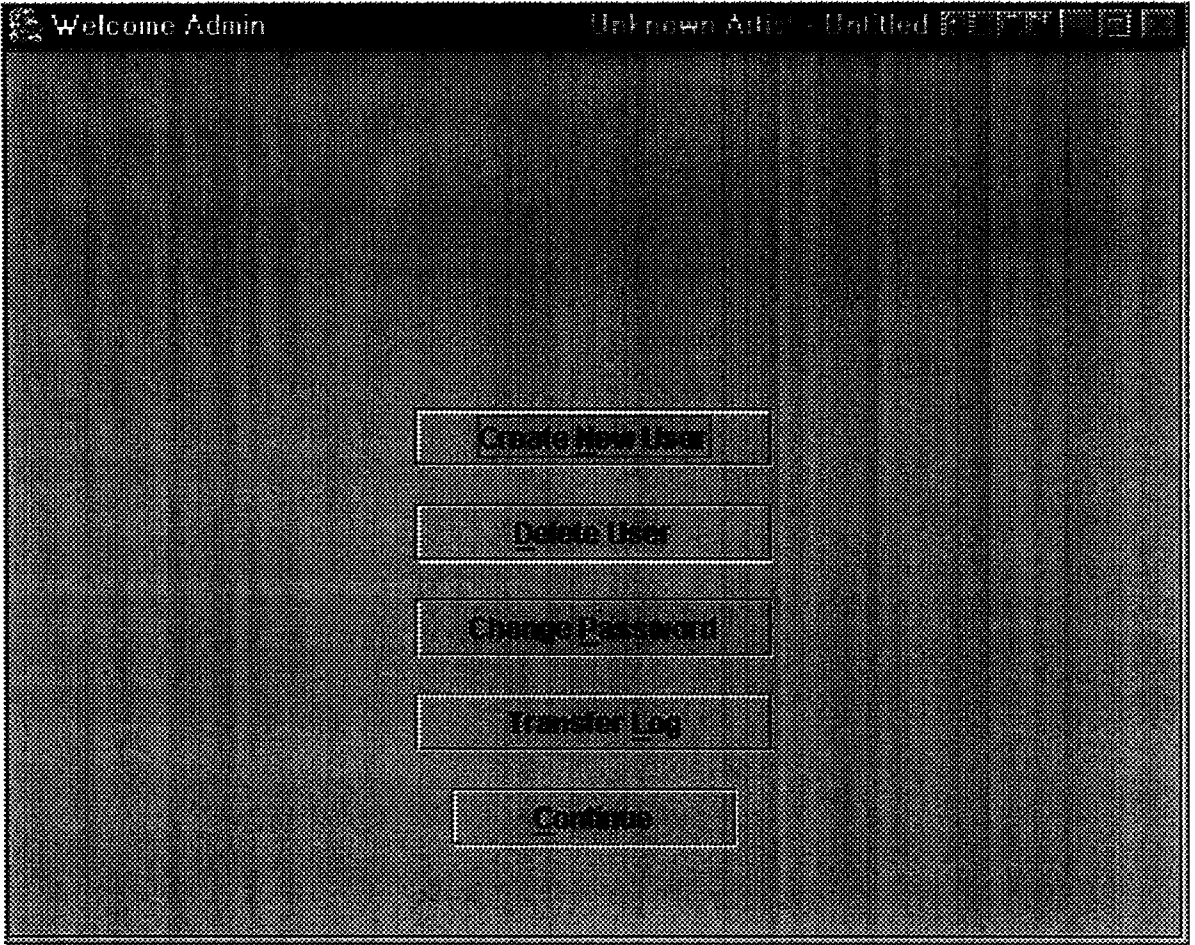
```
import java.io.*;
import java.net.*;
class client2
{
    public static void main(String a[])throws Exception
    {
        Socket ser =new Socket("Serveripaddress",1998);
        System.out.println("im in accepted");
        PrintStream outp=new PrintStream(ser.getOutputStream());
        InetAddress cadd=InetAddress.getLocalHost();
        String nsys=cadd.getHostName();
        String ipsys=cadd.getHostAddress();
        System.out.println("hi" + nsys +" hello "+ipsys);
        outp.println(nsys);
        outp.println(ipsys);
        ser.close();
    }
}
```

## 9.2 OUTPUT SCREENS



The image shows a screenshot of a 'Login Form' window. The window title bar contains the text 'Login Form' and standard window control icons (minimize, maximize, close) on the right. The main area of the window has a dark, textured background. On the left side, there are two labels: 'Login:' and 'Password:'. To the right of 'Login:' is a text input field containing the text 'kct'. To the right of 'Password:' is a password input field containing four asterisks '\*\*\*\*'. At the bottom of the form, there are three buttons: 'Reset', 'OK', and 'Cancel', arranged horizontally from left to right.

**Fig a: Login Form**



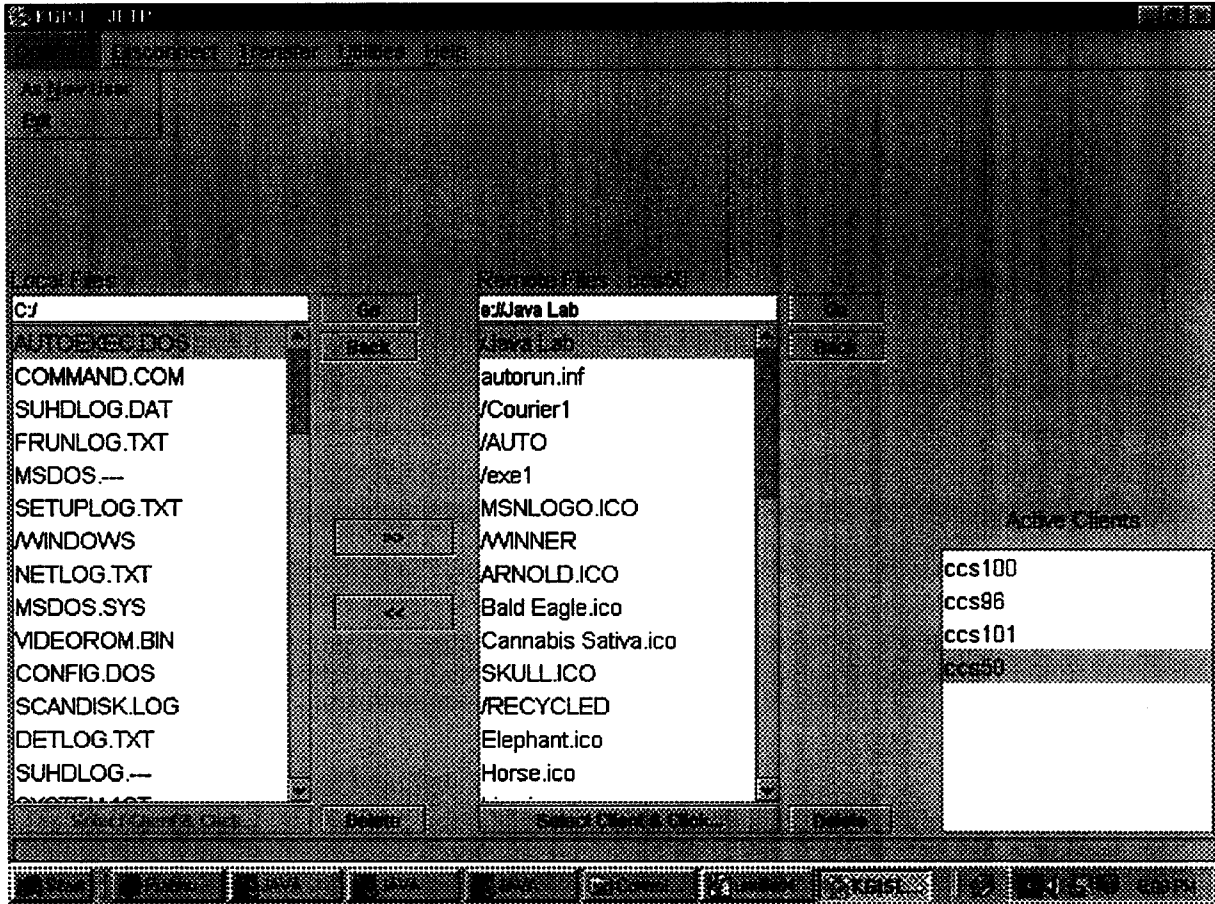
**Fig b: Administrator Form**



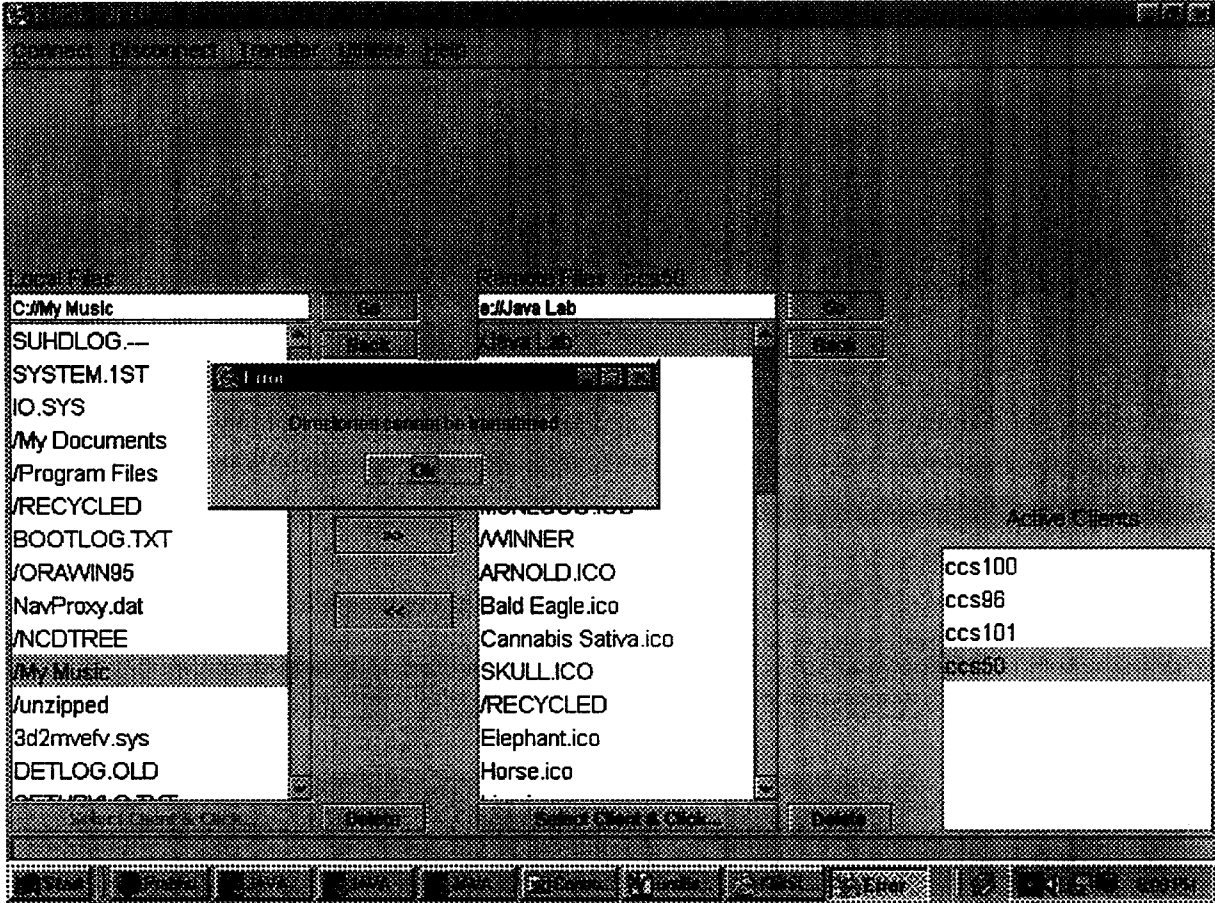
The image shows a window titled "Create New User" with a dark background. The form contains the following elements:

- Login Name:** A text input field containing the text "admin".
- Password:** A password input field containing six asterisks "\*\*\*\*\*".
- Confirm Password:** A password input field containing six asterisks "\*\*\*\*\*".
- Role:** A dropdown menu with "Admin" selected and a downward arrow on the right.
- Buttons:** Three buttons at the bottom: "Reset", "OK", and "Back".

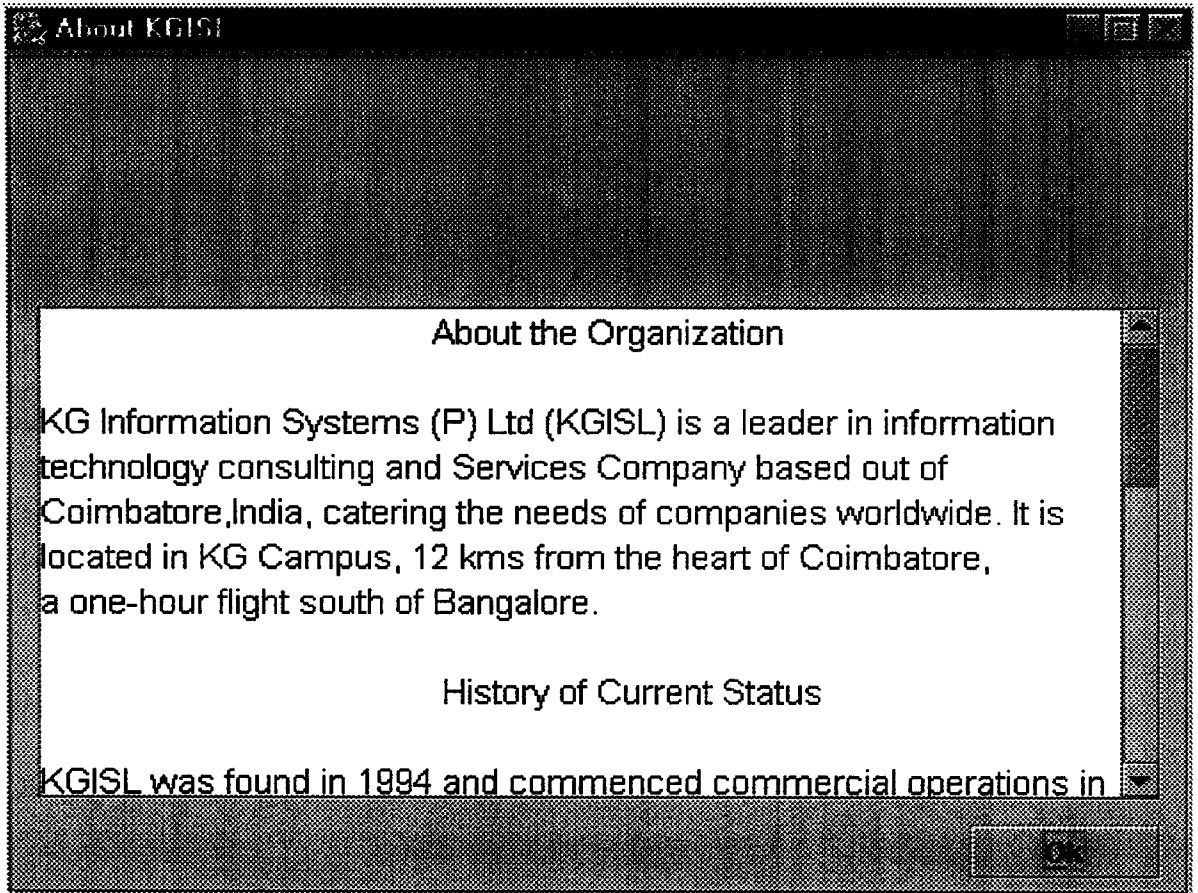
**Fig c: Create User Form**



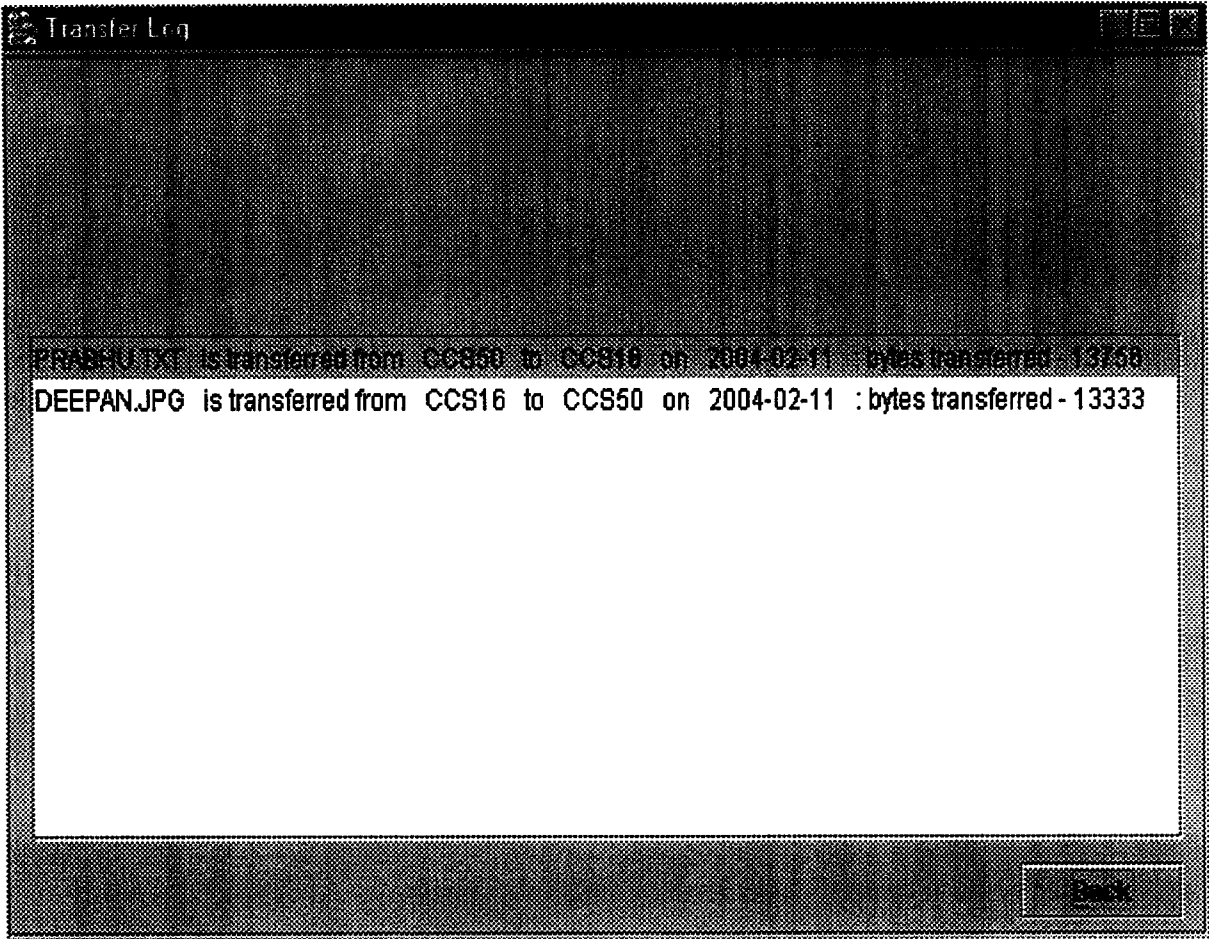
**Fig d: File Transfer Form**



**Fig e: Error Indication in File Transfer**



**Fig f: Utility Form**



**Fig g: Transfer Log Form**