

**VIVEKANAND EDUCATION SOCIETY'S INSTITUTE
OF TECHNOLOGY**

Department of Computer Engineering



Project Report on

Chat Application using RMI

8th SEM, 2020-2021

Submitted by

1. Manish S. Shinde (Roll No.-58, D17C)

Project Mentor

Dr. Prashant Kanade

(2020-21)

Project Definition-

develop an RMI-based chat system Java implementation of both client and server. port your client-server Internet Chat System to Java RMI. The new version should exploit the Java RMI framework instead of interfacing the TCP protocol at the socket level.

Abstract

A very simple cross platform client server chat application has been implemented in Java. Its design is described, limitations are discussed, improvements are proposed

Introduction

Implementing a chat server application provides a good opportunity for a beginner to design and implement a network-based system. The design is very simple. It is implemented in Java, since it is easy to program in, it precludes the need to deal with low-level memory management and includes powerful libraries for sockets and threads

Implementation

Remote Method Invocation

RMI is used for a distributed object references system. A distributed object is an object which publishes its interface on other machines. A Remote Object is a distributed object whose state is encapsulated. Stub and Skeleton are two objects used to communicate with the remote object.

Stub: Stub is a gateway for client program which is used to communicate with skeleton object, by establishing a connection between them.

Skeleton: Resides on Server program which is used for passing the request from stub to the remote interface.

Steps to implement Interface

1. Defining a remote interface
2. Implementing the remote interface
3. Creating Stub and Skeleton objects from the implementation class using rmic (rmi compiler)
4. Start the rmiregistry
5. Create and execute the server application program
6. Create and execute the client application program.

Conclusion

A Chat Application using RMI was created.

Code

Client.java

```
import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.util.LinkedList;
import java.util.Scanner;

public class Client extends UnicastRemoteObject implements ClientInterface, Runnable
{
    private ServerInterface server;
    public static String name = null;
    public static int age = 0;
    public static String sex = null;
    public static double locationX = 0.0;
    public static double locationY = 0.0;
    public static int uniqueID;
    public static LinkedList<ClientInterface> clientList = new LinkedList<ClientInterface>();
    protected Client(String[] args, ServerInterface server)
        throws RemoteException {
        this.server = server;
        this.name = args[3];
        this.locationX = Double.parseDouble(args[5]);
        this.locationY = Double.parseDouble(args[6]);
        this.sex = args[8];
        this.age = Integer.parseInt(args[10]);

        server.addClient(this);
    }
}
```

```

    }

    public void retrieveList(ClientInterface clientInfo) throws RemoteException{

        clientList.add(clientInfo);

    }

    public void showErrorMsg(String name, String msg) throws RemoteException{

        System.out.println(name+"(error): " + msg);

        System.exit(1);

    }

    public void clearList() throws RemoteException{

        clientList.clear();

    }

    public void printList() throws RemoteException{

        if(clientList.size() != 0){

            for(int i=0;i<clientList.size();i++){

                System.out.println("Unique ID : " + clientList.get(i).getUniqueID());

                System.out.println("Name : " + clientList.get(i).getName());

                System.out.println("Age : " + clientList.get(i).getAge());

                System.out.println("Sex " + clientList.get(i).getSex());

                System.out.println("Location : " + clientList.get(i).getLocationX() + " " + clientList.get(i).getLocationY());

            }

        }else System.err.println("No client found in the range given.");

    }

    public String getName() throws RemoteException {

        return this.name;

    }

    public double getLocationX() throws RemoteException {

        return this.locationX;

    }

    public void setLocationX(double x) throws RemoteException{

        this.locationX = x;

    }

    public double getLocationY() throws RemoteException {

        return this.locationY;

    }

    public void setLocationY(double y) throws RemoteException{

        this.locationY = y;

    }

```

```

    }

    public void showMesg(String sender_name, String msg)
        throws RemoteException {
        System.out.println(sender_name + " : " + msg);
    }

    public String getSex() throws RemoteException {
        return this.sex;
    }

    public int getAge() {
        return this.age;
    }

    public int getUniqueID() throws RemoteException {
        return this.uniqueID;
    }

    public void setUniqueID(int id) throws RemoteException {
        this.uniqueID = id;
    }

    public void run() {
        Scanner scanner = new Scanner(System.in);
        String msg;
        boolean value = true;

        while (value) {
            System.out.println("Enter your command");
            msg = scanner.nextLine();
            try {

                server.SendMessageOver(uniqueID, msg);

                if (msg.equalsIgnoreCase("quit")) {
                    System.exit(1);
                    value = false;
                }
            } catch (RemoteException ex) {
                ex.printStackTrace();
            }
        }
    }

    public static void main(String[] args) throws NotBoundException, MalformedURLException, RemoteException{
        String host = args[1];

```

```

        String ServerURL = "rmi://" + host + "/RMIServer";

        ServerInterface server = (ServerInterface) Naming.lookup(ServerURL);

        Scanner scanner = new Scanner(System.in);

        new Thread(new Client(args, server)).start();
    }
}

ClientInterface.java

import java.rmi.Remote;
import java.rmi.RemoteException;

public interface ClientInterface extends Remote{

    void showErrorMsg(String name, String msg) throws RemoteException;
    void retrieveList(ClientInterface clientInfo) throws RemoteException;

    void printList() throws RemoteException;
    void clearList() throws RemoteException;

    double getLocationX() throws RemoteException;

    void setLocationX(double x) throws RemoteException;

    double getLocationY() throws RemoteException;
    void setLocationY(double y) throws RemoteException;

    int getAge() throws RemoteException;

    int getUniqueID() throws RemoteException;
    void setUniqueID(int id) throws RemoteException;

    void showMesg(String name, String msg) throws RemoteException;

    String getName() throws RemoteException;
    String getSex() throws RemoteException;
}

```

Server.java

```
import java.rmi.*;
import java.rmi.registry.*;
import java.util.*;
import java.rmi.server.*;

public class Server extends UnicastRemoteObject implements ServerInterface {

    private ArrayList<ClientInterface> clients;
    public int uid = 1;

    public synchronized void addClient(ClientInterface client)
        throws RemoteException {

        boolean cc_a = true;
        for(int i=0;i<clients.size();i++){
            if(clients.get(i).getName().equalsIgnoreCase(client.getName())){
                cc_a = false;
                client.showErrorMsg("Server", "Duplicate username found. Please try again with a different
username.");
                break;
            }
        }

        if(cc_a == true){
            client.setUniqueID(uid);
            this.clients.add(client);
            System.out.println();
            System.out.println("Id   Name   Sex   Age   Location");

            System.out.println(client.getUniqueID() + "   " + client.getName() + "   " +
client.getSex() + "   " + client.getAge()+ "   " + client.getLocationX()+ "   " +client.getLocationY());

            uid++;
        }
    }
}
```

```
}
```

```
protected Server() throws RemoteException {  
    clients = new ArrayList<ClientInterface>();  
    System.out.println();  
    System.out.println("Server connection established !!");  
}
```

```
public synchronized void SendMessageOver(int id, String msg)  
    throws RemoteException { //id = msg sender's id  
  
    ArrayList<String> collection = new ArrayList<String>();  
    for (String retval : msg.split(" ")) {  
        collection.add(retval);  
    }  
  
    if (msg.equalsIgnoreCase("quit")  
        || collection.get(0).equalsIgnoreCase("quit")) {  
        for (int i = 0; i < clients.size(); i++) {  
            if (clients.get(i).getUniqueID() == id) {  
                System.out.println(clients.get(i).getName()  
                    + " has left.");  
                clients.remove(i); //remove client  
            }  
        }  
    } else if (collection.get(0).equalsIgnoreCase("send")) {  
  
        String sendMsg = "";  
        for (int i = 2; i < collection.size(); i++) {  
            sendMsg = sendMsg + collection.get(i) + " ";  
        }  
  
        String sender_name = "$";  
        Client c;  
        for(int i=0;i<clients.size();i++){
```



```

        if(clients.get(i).getUniqueID() == id){
            sender_name = clients.get(i).getName();

        }
    }

    boolean check = true;
    for (int i = 0; i < clients.size(); i++) {
        if (clients.get(i).getUniqueID() == Integer.parseInt(collection.get(1)) &&
(clients.get(i).getUniqueID() != id)) {

            clients.get(i).showMesg(sender_name,sendMsg);

        }else check = false;
    }

} else if(collection.get(0).equalsIgnoreCase("get")){ //getting information

    String location = "";
    for(int i=0;i<clients.size();i++){
        if(clients.get(i).getUniqueID() == id){
            location = location + clients.get(i).getLocationX() + " " + clients.get(i).getLocationY();
            clients.get(i).showMesg("Own Location", location);
        }
    }
}

}else if(collection.get(0).equalsIgnoreCase("go")){ //changing location

    double locationX = 0;
    double locationY = 0;
    for(int i=0;i<clients.size();i++){
        if(clients.get(i).getUniqueID() == id){
            locationX = Double.parseDouble(collection.get(1)) + clients.get(i).getLocationX();
            locationY = Double.parseDouble(collection.get(2)) + clients.get(i).getLocationY();

            clients.get(i).setLocationX(locationX);
            clients.get(i).setLocationY(locationY);

            String location = String.valueOf(locationX) + " " + String.valueOf(locationY);
            clients.get(i).showMesg("New Location", location);
        }
    }
}

}else if(collection.get(0).equalsIgnoreCase("list")){ //find client list with euclid formula

    double range = Double.parseDouble(collection.get(1));
    int index = -1;

```

```

double x=0;

double y=0;

for(int i=0;i<clients.size();i++){

    if(id == clients.get(i).getUniqueID()){

        index = i;

        x = clients.get(i).getLocationX();

        y = clients.get(i).getLocationY();

    }

}

if(index == -1){

    System.err.println("No such Client Found");

    System.exit(1);

}

clients.get(index).clearList();

for(int i=0;i<clients.size();i++){

    if(clients.get(i).getUniqueID() == id){

        continue;

    }

    double a = clients.get(i).getLocationX();

    double b = clients.get(i).getLocationY();

    double circumference = (x-a)*(x-a) + (y-b)*(y-b);

    if(circumference<=range*range){

        clients.get(index).retrieveList(clients.get(i));

    }

}

clients.get(index).printList();

}

}

public static void main(String [] args) throws RemoteException{

    Registry registry = LocateRegistry.getRegistry(1099);

    registry.rebind("RMIServer",new Server());

}

}

```

ServerInterface.java

```
import
java.rmi.Remote;

import java.rmi.RemoteException;

public interface ServerInterface extends Remote{
    void addClient(ClientInterface client) throws RemoteException;
    void SendMessageOver(int id, String msg) throws RemoteException;
}
}
```

OUTPUT

a) Starting RMI Registry

C:\Users\user\manish\DCMiniproject\RMIChatSystem\RMIChatSystem>rmiregistry 1099 &

```
Command Prompt - rmiregistry 1099
Microsoft Windows [Version 10.0.21343.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd manish
C:\Users\user\manish>cd DCMIniproject
C:\Users\user\manish\DCMiniproject>cd RMIChatSystem
C:\Users\user\manish\DCMiniproject\RMIChatSystem>cd RMIChatSystem
C:\Users\user\manish\DCMiniproject\RMIChatSystem\RMIChatSystem>javac *.java
C:\Users\user\manish\DCMiniproject\RMIChatSystem\RMIChatSystem>rmiregistry 1099 &
```

a) Creating the Server

C:\Users\user\manish\DCMiniproject\RMIChatSystem\RMIChatSystem>java Server

```
Command Prompt - java Server
C:\Users\user>cd manish
C:\Users\user\manish>cd DCMIniproject
C:\Users\user\manish\DCMiniproject>cd RMIChatSystem
C:\Users\user\manish\DCMiniproject\RMIChatSystem>cd RMIChatSystem
C:\Users\user\manish\DCMiniproject\RMIChatSystem\RMIChatSystem>java Server

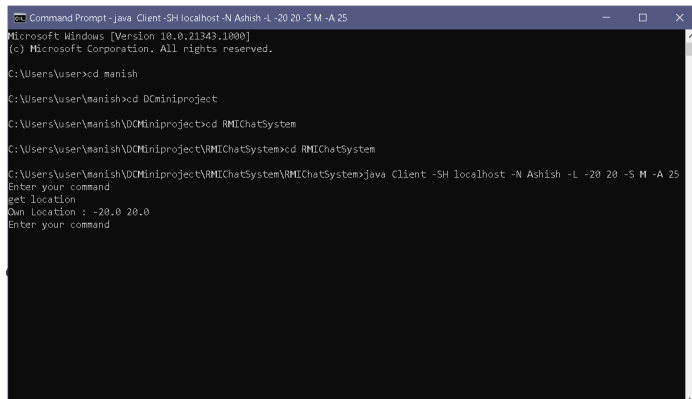
Server connection established !!

Id Name Sex Age Location
1 Ashish M 25 -20.0 20.0

Id Name Sex Age Location
2 Manish M 20 -25.0 30.0
```

C) Connecting the client 1 to server

```
C:\Users\user\manish\DCMiniproject\RMiChatSystem\RMiChatSystem>java Client -SH localhost -N Ashish -L -20 20 -S M -A 25
```

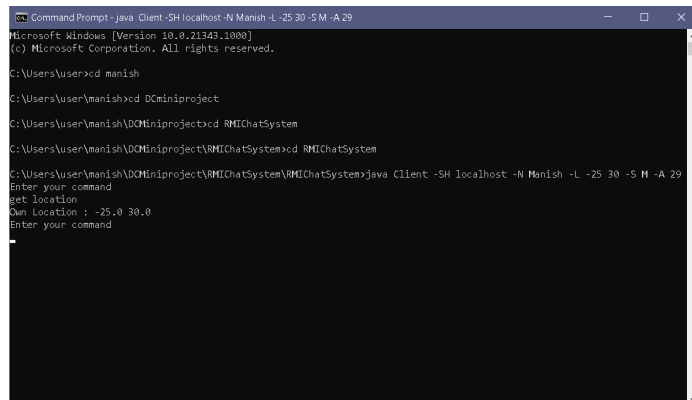


```
Command Prompt - java Client -SH localhost -N Ashish -L -20 20 -S M -A 25
Microsoft Windows [Version 10.0.21343.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd manish
C:\Users\user\manish>cd DCMiniproject
C:\Users\user\manish\DCMiniproject>cd RMiChatSystem
C:\Users\user\manish\DCMiniproject\RMiChatSystem>cd RMiChatSystem
C:\Users\user\manish\DCMiniproject\RMiChatSystem\RMiChatSystem>java Client -SH localhost -N Ashish -L -20 20 -S M -A 25
Enter your command
get Location
Own Location : -20.0 20.0
Enter your command
```

d)Connecting the client 2 to server

```
C:\Users\user\manish\DCMiniproject\RMiChatSystem\RMiChatSystem>java Client -SH localhost -N Manish -L -25 30 -S M -A 29
```

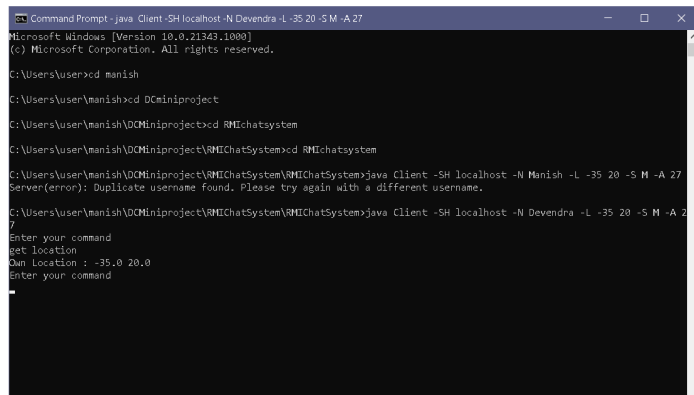


```
Command Prompt - java Client -SH localhost -N Manish -L -25 30 -S M -A 29
Microsoft Windows [Version 10.0.21343.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd manish
C:\Users\user\manish>cd DCMiniproject
C:\Users\user\manish\DCMiniproject>cd RMiChatSystem
C:\Users\user\manish\DCMiniproject\RMiChatSystem>cd RMiChatSystem
C:\Users\user\manish\DCMiniproject\RMiChatSystem\RMiChatSystem>java Client -SH localhost -N Manish -L -25 30 -S M -A 29
Enter your command
get Location
Own Location : -25.0 30.0
Enter your command
```

e) Connecting Client 3

C:\Users\user\manish\DCMiniproject\RMICChatSystem\RMICChatSystem>java Client -SH localhost -N Devendra -L -35 20 -S M -A 27



```
Command Prompt - java Client -SH localhost -N Devendra -L -35 20 -S M -A 27
Microsoft Windows [Version 10.0.21343.1000]
(c) Microsoft Corporation. All rights reserved.

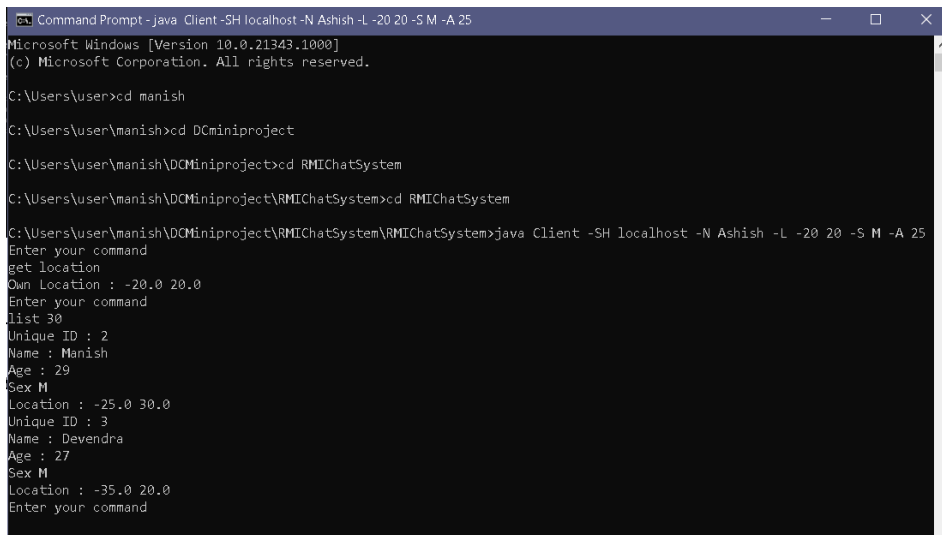
C:\Users\user>cd manish
C:\Users\user\manish>cd DCMiniproject
C:\Users\user\manish\DCMiniproject>cd RMICChatSystem
C:\Users\user\manish\DCMiniproject\RMICChatSystem>java Client -SH localhost -N Manish -L -35 20 -S M -A 27
Server(error): Duplicate username found. Please try again with a different username.

C:\Users\user\manish\DCMiniproject\RMICChatSystem\RMICChatSystem>java Client -SH localhost -N Devendra -L -35 20 -S M -A 27
Enter your command
get location
Own Location : -35.0 20.0
Enter your command
```

f) Clients 1 and 2 search neighbours around them

Enter your command

list 30



```
Command Prompt - java Client -SH localhost -N Ashish -L -20 20 -S M -A 25
Microsoft Windows [Version 10.0.21343.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd manish
C:\Users\user\manish>cd DCMiniproject
C:\Users\user\manish\DCMiniproject>cd RMICChatSystem
C:\Users\user\manish\DCMiniproject\RMICChatSystem>cd RMICChatSystem
C:\Users\user\manish\DCMiniproject\RMICChatSystem\RMICChatSystem>java Client -SH localhost -N Ashish -L -20 20 -S M -A 25
Enter your command
get location
Own Location : -20.0 20.0
Enter your command
list 30
Unique ID : 2
Name : Manish
Age : 29
Sex M
Location : -25.0 30.0
Unique ID : 3
Name : Devendra
Age : 27
Sex M
Location : -35.0 20.0
Enter your command
```

g) Client Ashish sends message to manish and manish receives message from ashish

```
Command Prompt - java Client -SH localhost -N Ashish -L -20 20 -S M -A 25
C:\Users\user\manish>cd D:\miniproject
C:\Users\user\manish\DCMiniproject>cd RMIChatSystem
C:\Users\user\manish\DCMiniproject\RMIChatSystem>cd RMIChatSystem
C:\Users\user\manish\DCMiniproject\RMIChatSystem\RMIChatSystem>java Client -SH localhost -N Ashish -L -20 20 -S M -A 25
Enter your command
get location
Own Location : -20.0 20.0
Enter your command
list 30
Unique ID : 2
Name : Manish
Age : 25
Sex M
Location : -25.0 30.0
Unique ID : 3
Name : Devendra
Age : 27
Sex M
Location : -35.0 20.0
Enter your command
send 2 hello manish
Enter your command
Hello manish, How are you?
Enter your command
```

```
Command Prompt - java Client -SH localhost -N Manish -L -25 30 -S M -A 29
C:\Users\user\manish>cd D:\miniproject
C:\Users\user\manish\DCMiniproject>cd RMIChatSystem
C:\Users\user\manish\DCMiniproject\RMIChatSystem>cd RMIChatSystem
C:\Users\user\manish\DCMiniproject\RMIChatSystem\RMIChatSystem>java Client -SH localhost -N Manish -L -25 30 -S M -A 29
Enter your command
get location
Own Location : -25.0 30.0
Enter your command
list 30
Unique ID : 1
Name : Ashish
Age : 25
Sex M
Location : -20.0 20.0
Unique ID : 3
Name : Devendra
Age : 27
Sex M
Location : -35.0 20.0
Enter your command
Ashish : hello manish
send 2 ashish
Enter your command
Hello I am fine
Enter your command
send to hello ashish
```

h) Client Ashish quits

```
Command Prompt
Age : 20
Sex M
Location : -25.0 30.0
Unique ID : 3
Name : Devendra
Age : 27
Sex M
Location : -35.0 20.0
Enter your command
send 2 hello manish
Enter your command
Hello manish, How are you?
Enter your command
quit
C:\Users\user\manish\DCMiniproject\RMIChatSystem\RMIChatSystem>
```