

```

package client;

import ocsf.client.*;
import common.*;
import java.io.*;

public class ChatClient extends AbstractClient{

    ChatIF clientUI;
    boolean isLogin;

    public ChatClient(String host, int port, ChatIF clientUI) throws IOException{
        super(host, port);
        this.clientUI = clientUI;
        isLogin=false;
        openConnection();
        clientUI.display("You current status is not logged in.");
    }

    public void handleMessageFromServer(Object msg){
        if(msg.toString().substring(0,7).equals("Welcome"))
            this.isLogin=true;
        clientUI.display(msg.toString());
    }

    public void handleMessageFromClientUI(String message){
        if(message.charAt(0)=='#'){
            String[] command;
            command = message.split("\\s+");
            this.handleCommandFromClientUI(command);
        }
        else{
            if(isLogin){
                try{
                    this.sendToServer(message);
                }
                catch(IOException e){
                    clientUI.display("<System> Could not send message to server. Terminating client.");
                    quit();
                }
            }
            else clientUI.display("<System> Not Loged in");
        }
    }

    public void handleCommandFromClientUI(String[] command){
        try{
            if(command[0].equals("#quit")){
                if(isLogin)
                    this.sendToServer("#logoff");
                clientUI.display("<System> Session has ended.");
                this.quit();
            }
        }
    }
}

```

```

}
else if(command[0].equals("#logout")){
    if(isLogin){
        this.sendToServer("#logout");
        this.closeConnection();
        isLogin=false;
        clientUI.display("<System> You have Logged off from the server.");
    }
    else clientUI.display("<System> You have already logged off.");
}
else if(command[0].equals("#sethost")){
    if(isLogin) clientUI.display("<System> Please log off first");
    else {
        this.setHost(command[1]);
        clientUI.display("<System> New host (" + command[1] + ") has been set.");
    }
}
else if(command[0].equals("#setport")){
    if(isLogin) clientUI.display("<System> Please log off first");
    else {
        try{
            this.setPort(Integer.parseInt(command[1]));
            clientUI.display("<System> New port (" + command[1] + ") has been set.");
        }
        catch(NumberFormatException e){
            clientUI.display("<System> Invalid port");
        }
    }
}
else if(command[0].equals("#login")){
    if(isLogin) clientUI.display("<System> You have already logged in.");
    else {
        try{ this.openConnection();
            this.sendToServer("#login " + command[1] + " " + command[2]);
        }
        catch(ArrayIndexOutOfBoundsException e){
            clientUI.display("<System> Invalid Input .");
        }
    }
}
else if(command[0].equals("#gethost")){
    if(isLogin) clientUI.display(this.getHost());
    else clientUI.display("<System> You are currently not connected.");
}
else if(command[0].equals("#getport")){
    if(isLogin) clientUI.display(this.getPort());
    else clientUI.display("<System> You are currently not connected.");
}
else if(command[0].equals("#reg")){
    clientUI.display("<System> Requesting ...");
    this.sendToServer(command[0] + " " + command[1] + " " + command[2] + " " + command[3]);
}
else if(command[0].equals("#regInfo")){

```

```

        clientUI.display("<System> Sending infomation ...");
        this.sendToServer(command[0] + " " + command[1]);
    }
    else clientUI.display("<System> Command not found.");
}
catch(IOException e){
    clientUI.display("<System> Could not reach server. Terminating client.");
    quit();
}
}

public void quit(){
    try{
        closeConnection();
    }
    catch(IOException e){}
    System.exit(0);
}

public void connectionException(Exception exception){
    try{
        this.closeConnection();
        isLogin=false;
        clientUI.display("<System> The connection to the Server (" + this.getHost() + ":" +
this.getPort() + ") has been disconnected.");
    }
    catch(IOException e){
        clientUI.display("<System> Unexpected error. Program terminates.");
        quit();
    }
}
}
}

```