Team Members:

Ravneet Singh Sidhu ---------1001106510

Brunda Padmaraju ---------1001095203

AnuroopSai Potthuru ---------1001095202

Suhaib Siraj Shaik ---------1001106493

PROJECT-2 DOCUMENTATION

CSE-5306-003-DISTRIBUTED SYSTEMS

CHAT ROOM USING PEER TO PEER NETWORK ARCHITECTURE

In this project we have used a light server to implement the Peer to Peer network architecture. There is one light server, which maintains the list of all the connected peers to that network and it also helps peers to locate their desired resources or act as task scheduler to coordinate actions among them. To locate resources, a peer sends a request to the server to get the list of connected peers to that network. In this P2P system, Peer A submits a request to the server to acquire a list of nodes that satisfy the request. Once Peer A obtains the list (which contains Peer B and Peer C), it communicates directly with the nodes. Each peer has tcp socket which acts independently.

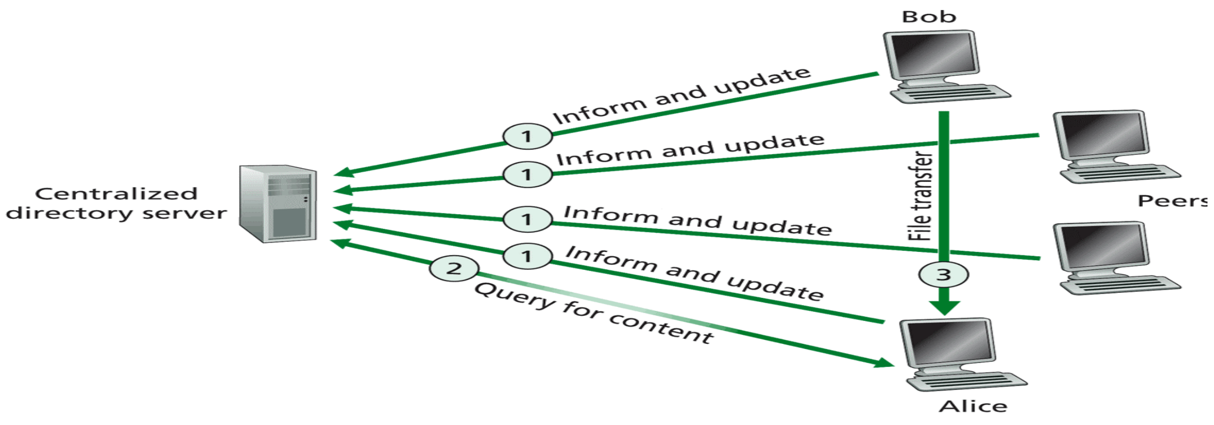


Figure1:Peer to Peer Architecture

Peers have to login and the authentication is done through database in XAMPP. The user names and passwords are saved in the database if a new peer has to register. The **password is hashed using MD5 with Salt** to maintain security. After the login is done we are presented to a window where we need to register onto the network using available port IDs.

Once we are registered to the network the light server acts as the peer list provider which are actively connected to the network. When we need to send a message to all the connected peers on the network it is done through **direct broad-cast** i.e.., each peer which needs to send the message does it directly using respective sockets. As each peer has a Server Socket listening on a desired port, messages are sent to the peers without the intervention of the server. We have also implemented one-to-one communication using **direct uni-cast** where from the list of online peers a user can choose a particular peer to communicate with directly.

Once the user disconnects from the network its status is changed in the server (database) as disconnected and message is sent to all the peers stating that the particular user has disconnected from the network, so that the future messages are not sent to it.

The entire code is implemented in Java and the integral part is java Threads and java Socket programming. The databases are maintained in XAMPP. Net Beans was used as a platform to write code and create GUI for the project.

Extra Credit: List of users connected to the server are displayed on the right side of the chat when we click on the Online Peers button. We have also implemented message broadcast and message uni-cast that user can control by selecting peers from the online list.

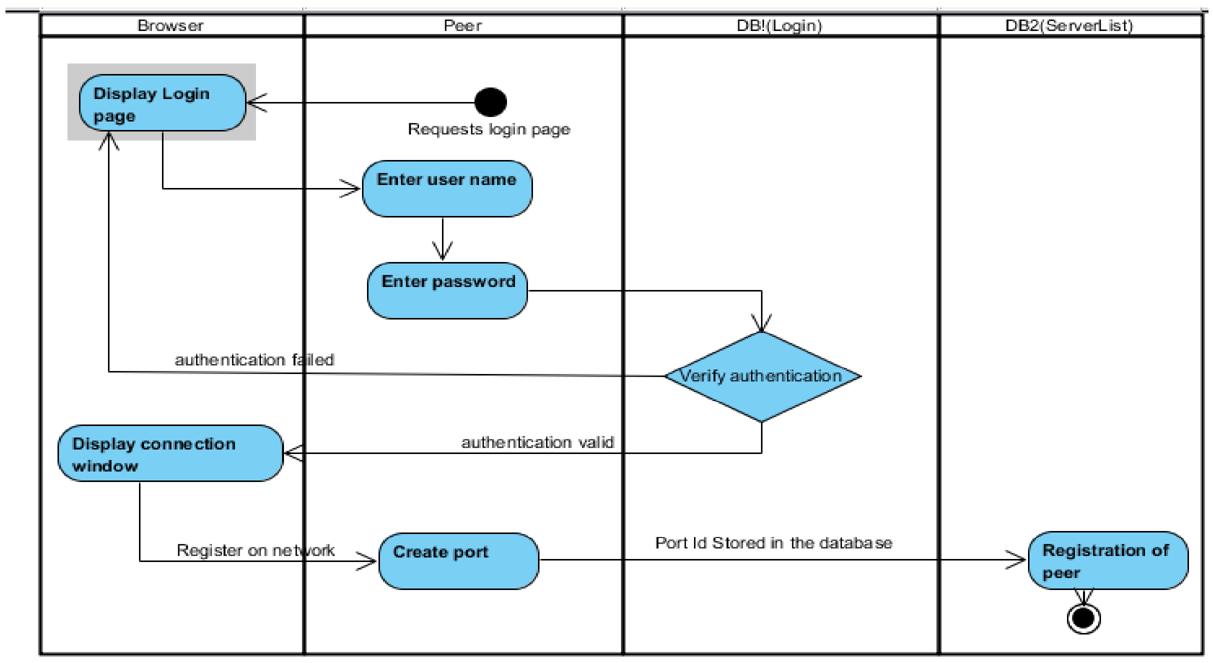


Figure2:Login and Registration of each peer to a network

Steps to run the project:

Initially run **PeerLogin.java** which in turn invokes the **PeerRegisterDBconnect.java.** This displays the login and registration Window of the user.

To participate in a Chat Room one needs to register first. The registered users information is stored in the MySQL in database **Login** in the table **login_info** for authentication. For Security, **password is hashed using MD5 with Salt**. Once registered ,user can login to the chat room which displays the Register on to the network Window.

After logging in it invokes **RegisterOnpeerNetwork.java** which enables the peers to register on to a network using light server using Respective PORT ID's by invoking **PeerServerDBconnect.java .**The information related to the peer Port Id and peer name are stored in the database **ServerList** in the table **Server_info.**

After connecting to peers on network, it displays CHAT GUI by invoking **ChatGUI1.java .**

Tasks done by ChatGUI Window:

1)Allows peers to connect, disconnect from the network.

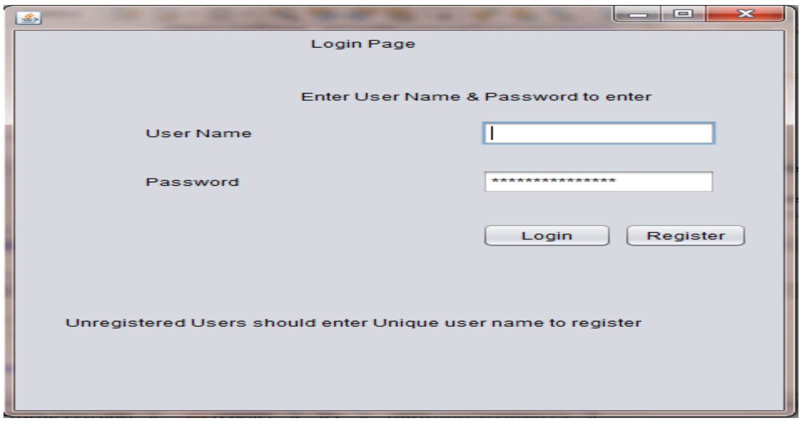
2)Unicast and multicast messages to the connected peers.

3)Displays the list of online peers on the right side of the window.

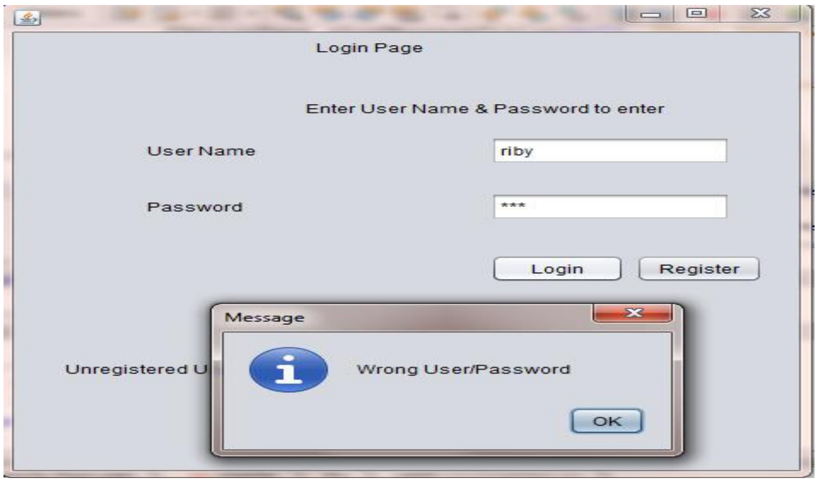
4)Allows to store history of the chat.

Screen shots:

1) Login and Registration Window

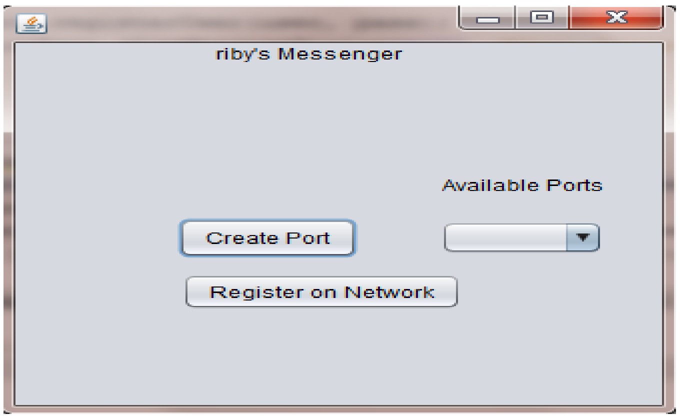

Login Page
Enter User Name & Password to enter
User Name
Password
****
Login
Register
Unregistered Users should enter Unique user name to register

2) Authentication for User Login(Invalid Login)

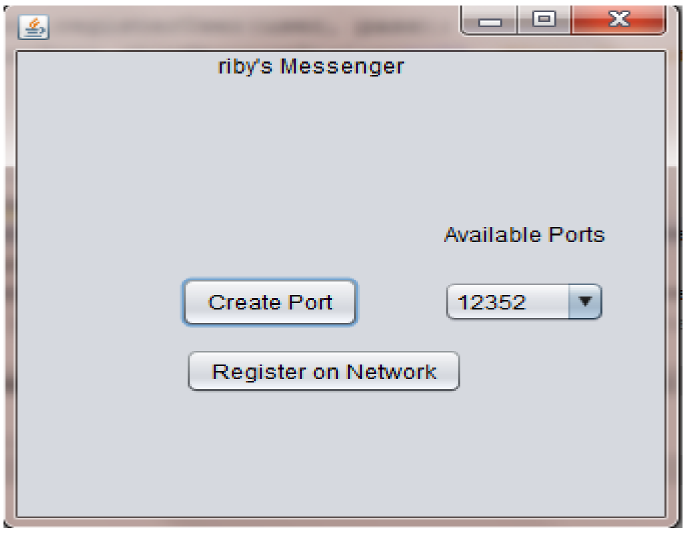


Wrong User/Password

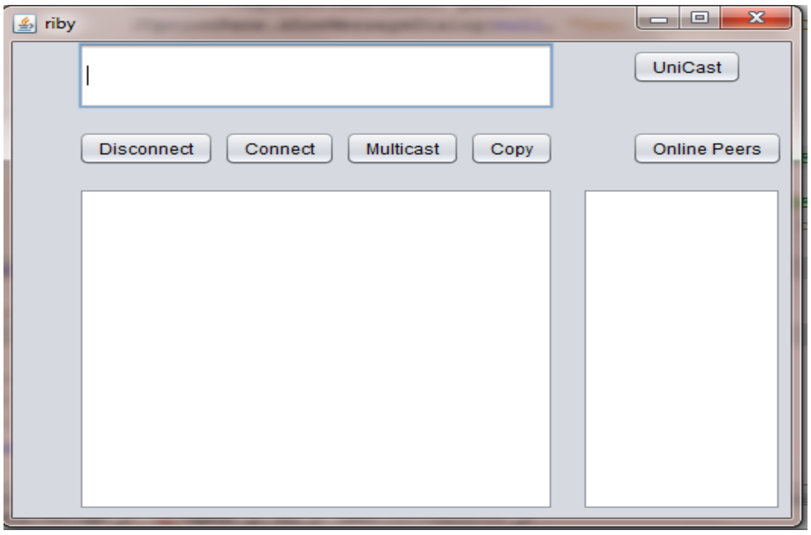
3)Creation of Port



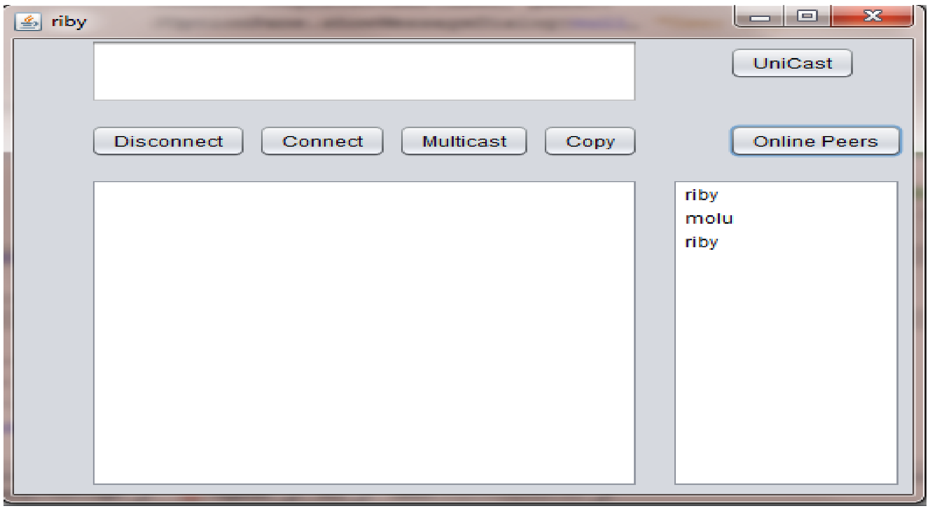
4)Register on to a Network



5)Peer GUI Window

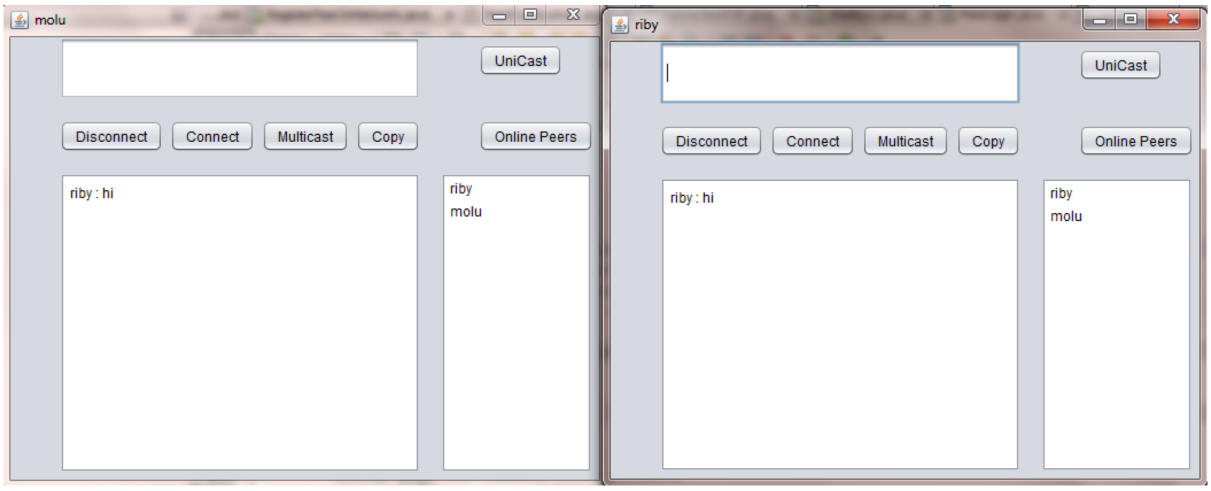


6)List of users Connected



riby
molu
riby

7)Multicast a message



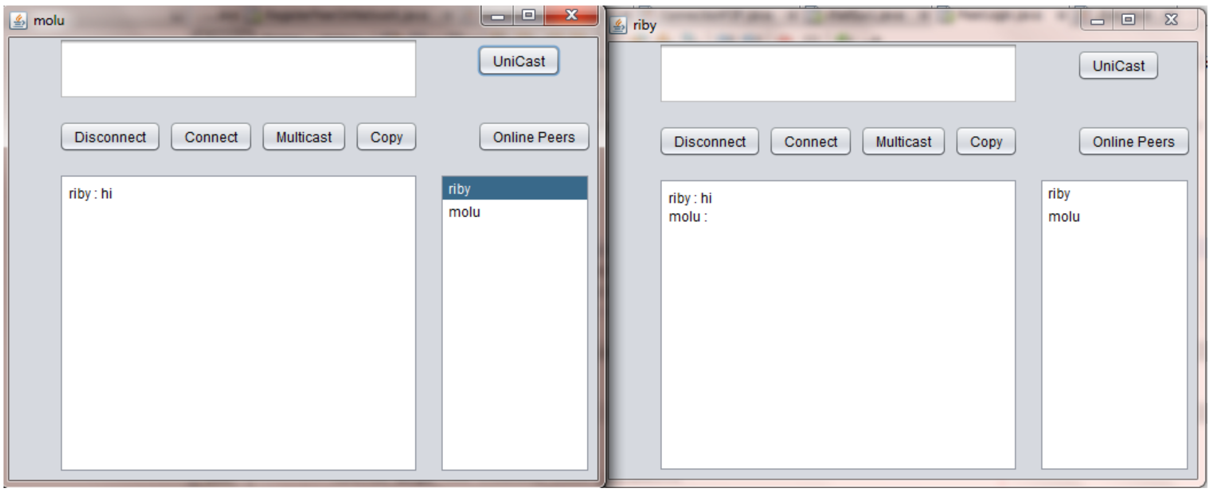
riby : hi

riby
molu

riby : hi

riby
molu

8)Unicast a message



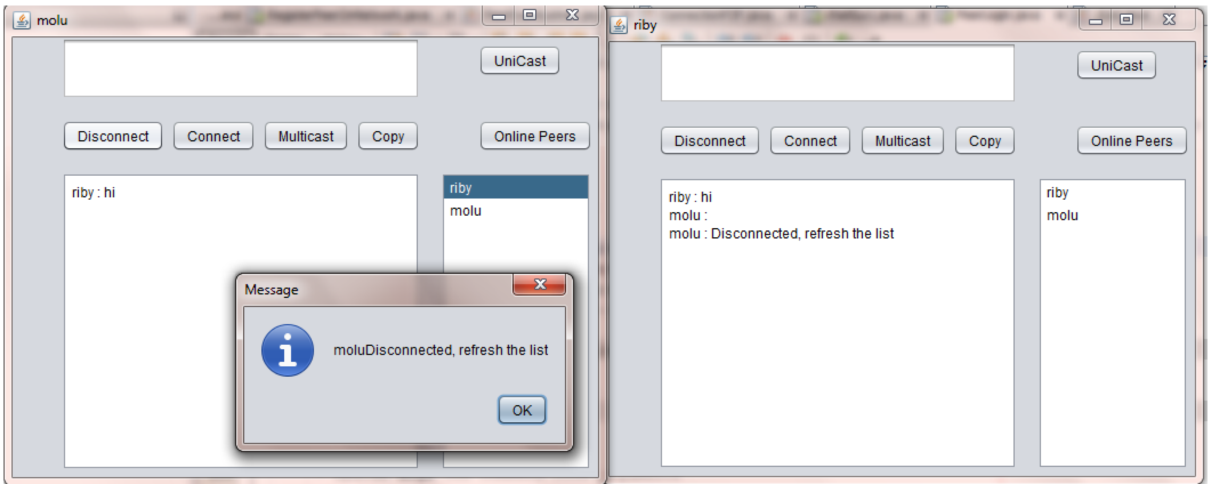
riby : hi

riby
molu

riby : hi
molu :

riby
molu

9)Disconnect from the network



riby : hi

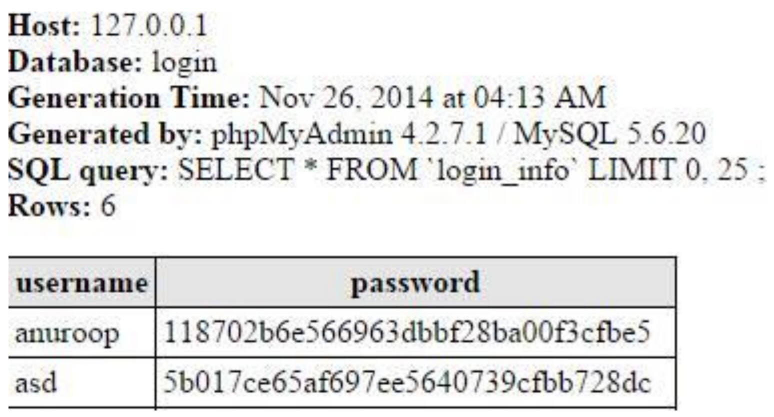
riby
molu

moluDisconnected, refresh the list

riby : hi
molu :
molu : Disconnected, refresh the list

riby
molu

10) Login_Info Table:



Host: 127.0.0.1
Database: login
Generation Time: Nov 26, 2014 at 04:13 AM
Generated by: phpMyAdmin 4.2.7.1 / MySQL 5.6.20
SQL query: SELECT * FROM `login_info` LIMIT 0, 25 ;
Rows: 6

username	password
anuroop	118702b6e566963dbbf28ba00f3cfbe5
asd	5b017ce65af697ee5640739cfbb728dc

References:

1. Fig 1: http://mscancer22.tripod.com/applicationlayer/id35.html
2. www.stackoverflow.com
3. www.github.com
4. http://www.javaworld.com/article/2071877/enterprise-java/peer-to-peer-applications-made-easy.html
5. http://www.javaworld.com/article/2075733/enterprise-java/the-jxta-solution-to-p2p.html?page=2
6. http://en.wikipedia.org/wiki/Peer-to-peer
7. http://docs.oracle.com/javase/tutorial/collections/interfaces/set.html
8. https://www.youtube.com/watch?v=Uo5DY546rKY
9. http://tutorials.jenkov.com/java-concurrency/creating-and-starting-threads.html
10. http://www.dreamincode.net/forums/topic/259777-a-simple-chat-program-with-clientserver-gui-optional/