CS 5523: Operating Systems Homework 5 Report of the work

This mini project tends to implement the concepts of distributed systems using java RMI. The code for this assignment is contained in the **src** directory. Following are the steps to get the program running:

- 1. Extract the zipped file for all the code.
- 2. Compile using "javac *.java".
- 3. Open at least 4 tabs of the terminal:
 - a) to compile all the code and run rmiregistry in the first tab.
 - b) in the **second tab run "java HelloServer"** to start the server.
 - c) in the **third and fourth tabs run "java HelloClient"** to start 2 client systems that would be served under this server.
- Follow the instructions on the screen.

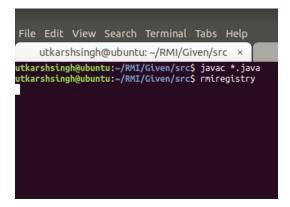
The zipped directory also contains a README.md file which has the above instructions (how to compile and successfully run the code).

The assignment tends to implement the following functionalities:

- 1. **go -50 30**: client wants to go 50m South and 30m East, server updates the location and reports the new location (go 40 -20 means go 40m North and 20m West).
- 2. **get location**: client wants to know its current (x,y) location, server reports the current location of this client.
- 3. **list 30**: client wants to get the list of users within 30m, server determines such clients/users within 30m of requesting client and send their information (user-id, name, age, additional info for extra credits) to the client, client keeps that list in a linked list and prints their user-id, name, and age on the screen.
- 4. **chat :** (extra credits) client sends msg directly to user-id. Server not involved.
- 5. **quit**: client leaves the system, server takes it out from the list.

These above mentioned functionalities can be implemented with any number of clients in the system as this uses the concepts of Factory (the number of clients do not need to be pre defined, any number of clients can come and go from the system). The only limitation using this project would be the machine the code runs on (memory limitations in handling increasing number of clients).

The implementation of the above functionalities work as expect. Below are the screenshots to demonstrate the functionalities:



All the java code is compiled and rmiregistry is run.

Then in the second tab we run **HelloServer** to startup the server program:

```
File Edit View Search Terminal Tabs Help

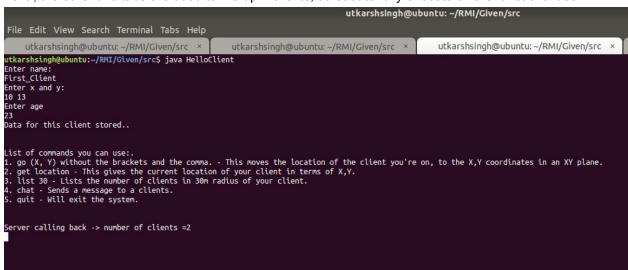
utkarshsingh@ubuntu:~/RMI/Given/src × utkarsh

utkarshsingh@ubuntu:~/RMI/Given/src$ java HelloServer

Server is ready
number of clients = 2
```

Server is up and running.

Next, the other two tabs are used to fire up 2 clients, at least to fully execute all the functionalities:



Client 1 is up and running.

Now we run the second client:

```
utkarshsingh@ubuntu: -/RMI/Given/src × utkarshsingh@ubunt
```

Client 2 is up and running.

First functionality: go x y:

```
Server calling back -> number of clients =2
go 13 38
Connecting to server now... Server is moving the client to the new location...
The new location has been updated with the values provided: 13 and 38

List of commands you can use:.
1. go (X, Y) without the brackets and the comma. - This moves the location of the client you're on, to the X,Y coordinates in an XY plane.
2. get location - This gives the current location of your client in terms of X,Y.
3. list 30 - Lists the number of clients in 30m radius of your client.
4. chat - Sends a message to a clients.
5. quit - Will exit the system.
```

Second functionality: get location:

```
File Edit View Search Terminal Tabs Help

utkarshsingh@ubuntu:-/RMI/Given/src × utkarshsingh@ubuntu:-/RMI/Given/src vutkarshsingh@ubuntu:-/RMI/Given/src vutkarshsingh
```

Third functionality: list n ('n' being the radius of search):

```
utkarshsingh@ubuntu:-/RMI/Given/src ×

utkarshsingh@ubuntu:-/RMI/Given/src v

utkarshallesseldessendessendessendessendessendessendessendessendessendessendessendessendessendessendessendessendessendessendessendessen
```

Fourth functionality: chat:

```
List of commands you can use:.

1. go (X, Y) without the brackets and the comma. - This moves the location of the client you're on, to the X,Y coordinates in an XY plane.

2. get location - This gives the current location of your client in terms of X,Y.

3. list 30 - Lists the number of clients in 30m radius of your client.

4. chat - Sends a message to a clients.

5. quit - Will exit the system.

Chat
Enter client's name:
Second_Client
Enter a message to send:
Hit there
Hi from First_Client to Second_Client : message sent to you is Hi there

List of commands you can use:.

1. go (X, Y) without the brackets and the comma. - This moves the location of the client you're on, to the X,Y coordinates in an XY plane.

2. get location - This gives the current location of your client in terms of X,Y.

3. list 30 - Lists the number of clients in 30m radius of your client.

5. quit - Will exit the system.
```

```
File Edit View Search Terminal Tabs Help

utkarshsingh@ubuntu: -/RMI/Given/src × utkarshsingh@ub
```

Fifth functionality: quit:

