**Chat Application Documentation DS Lab 1 (Write up)**

**System Requirements:**

* Eclipse IDE or any other Java IDE
* Windows
* Java

**Compiling and Running the Program:**

1. Import the project into eclipse IDE. Project name is ChatApplicationDS.
2. Start the server by running ChatAppServer class as “java application”
3. Start the client by running ChatAppClient class as “run configurations”
   1. Change the Main class to ChatAppClient
   2. Name the configuration with a unique name say ChatClient1
   3. In the arguments tab enter the Username of the client that wants to be registered and

Whether the user wants others to see him/her online by giving (Y or N).

* 1. Apply and run
  2. The above process pops up a client window
  3. Repeat the same process for as many clients as needed

1. One can then check the available users who are online (displayed in text area in each client window).
2. Once all required clients are up and running, from client1 enter the Name of the user say client2 that you want to connect to in the text box beside the connect button and connect.
3. Similarly, from client2, connect to client1 which establishes a two-way communication between client1 and client2
4. Any message that is sent from a client is displayed on the chat box of the other client and on the server console as well (with header information)
5. Logout button closes the window of that client and disconnects him from the client he is connected to.
6. The process is same for as many client pairs as required.
7. User has to type the message and click on send button for sending the message.
8. All the communication that happens between the clients is displayed on the server console.

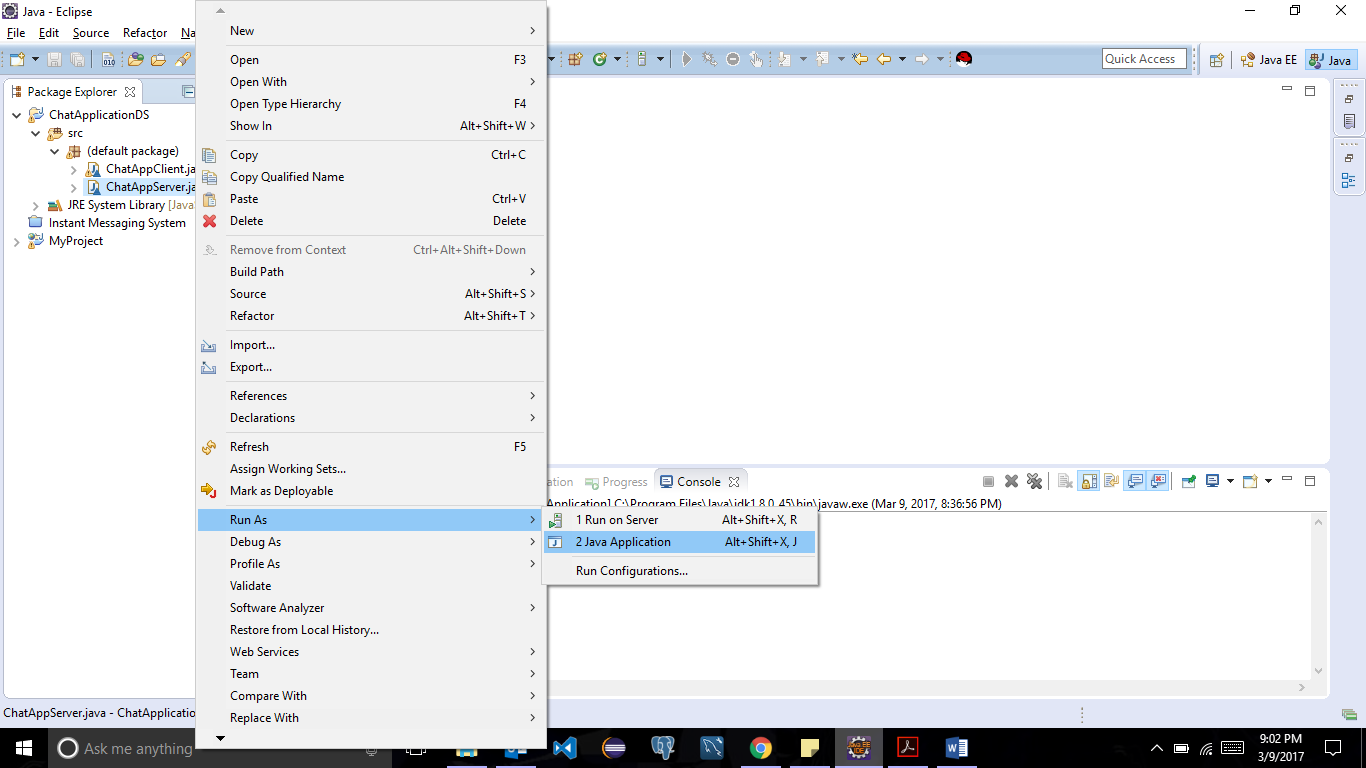
**Extra Functionality (Extra credit functionalities):**

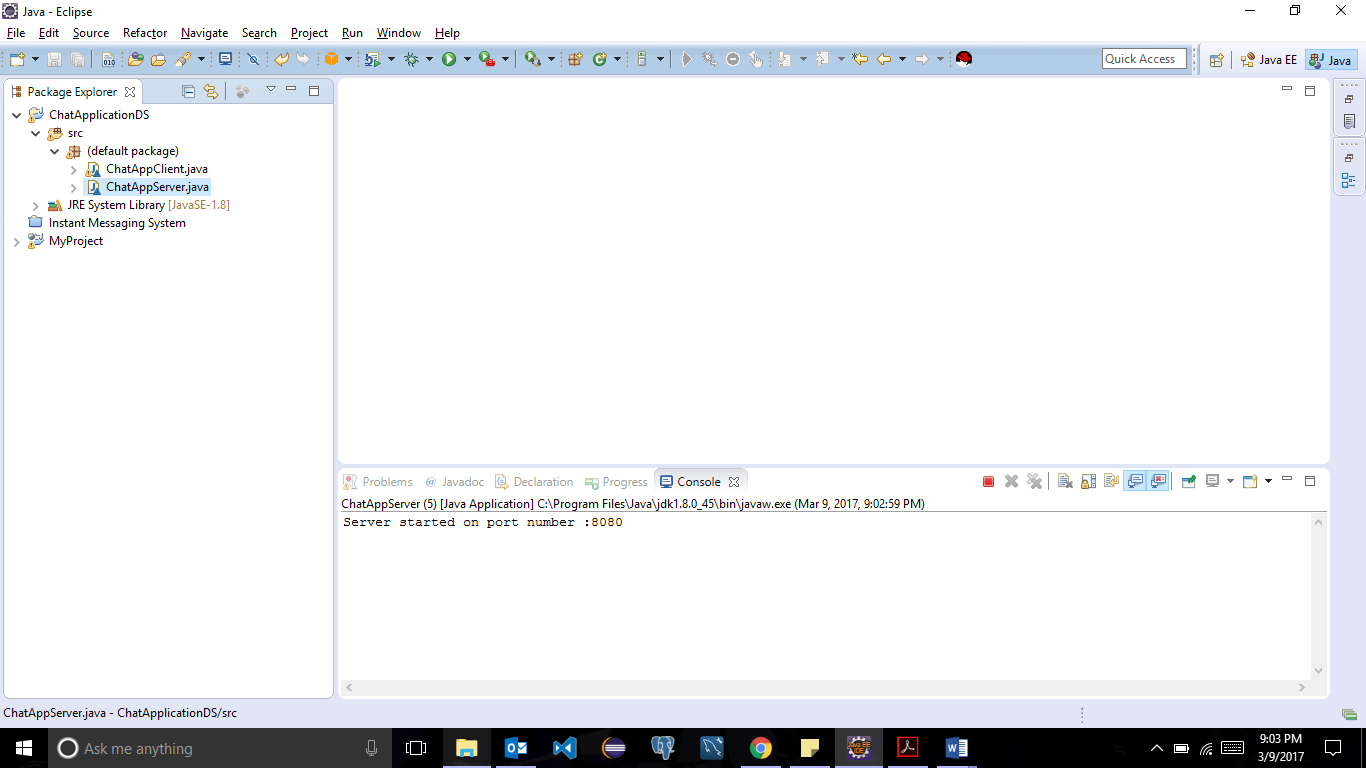
1. Names of all the users that are online are displayed on the chat box of each client when the client logs in.
2. The server is multithreaded. The task manager shows all the connected clients along with the server till the chat application is closed.

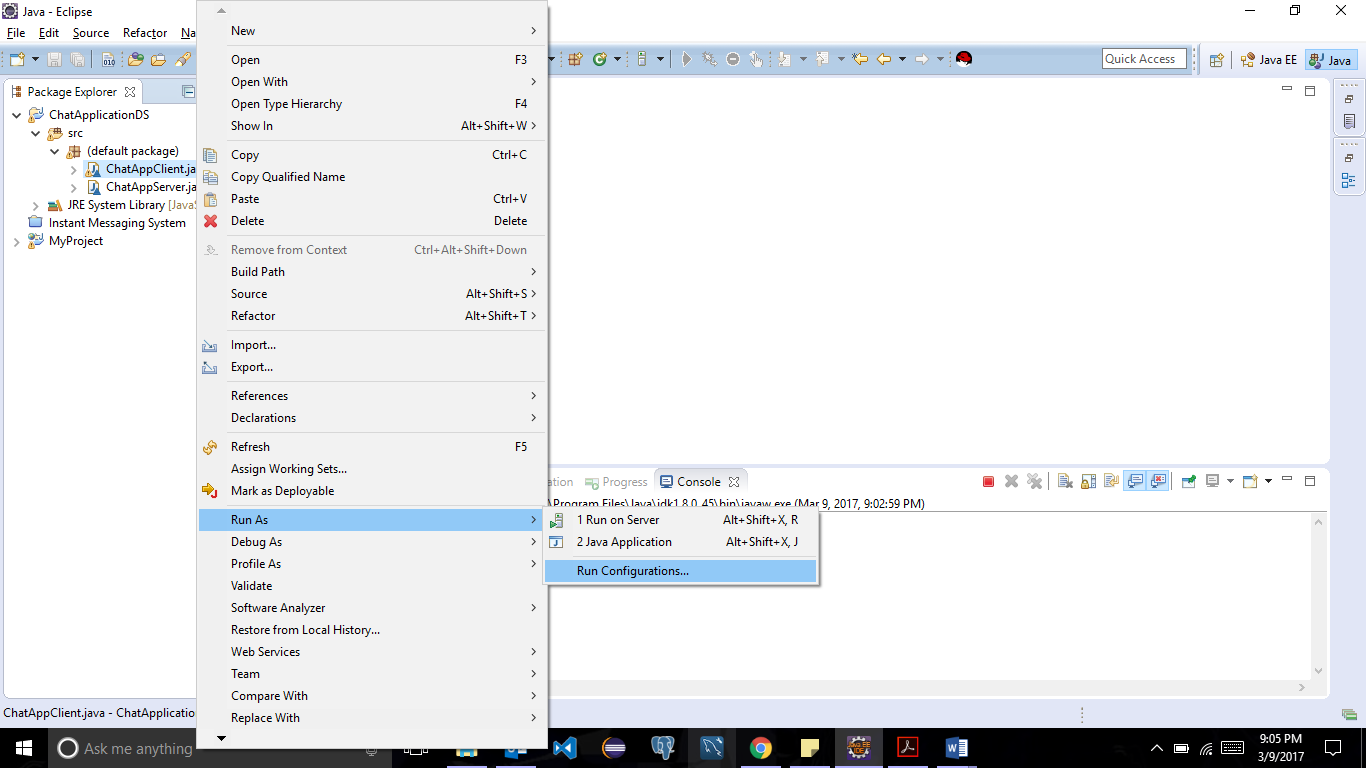
**Screen Shots of the execution process:**

STEP WISE EXECUTION OF THE APPLICATION (SCREEN SHOTS)

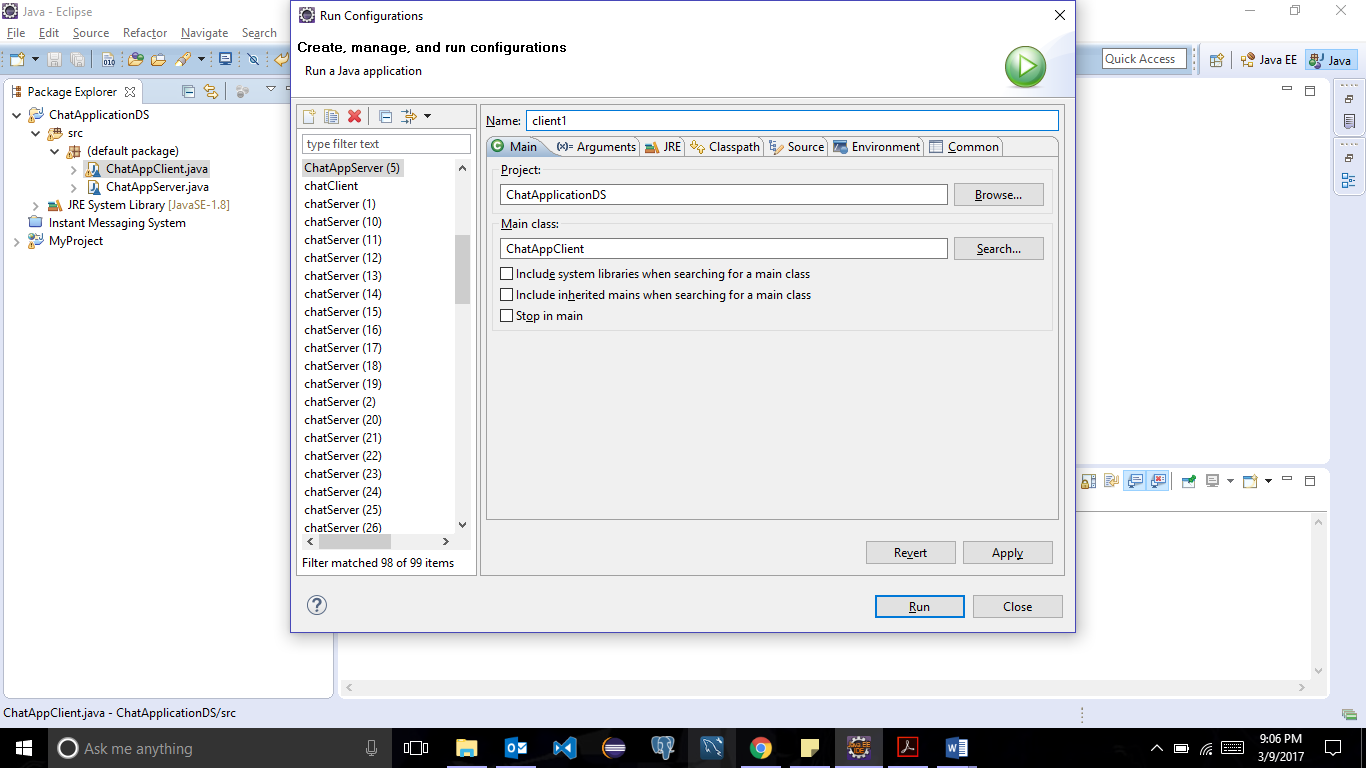
1. Starting the server:



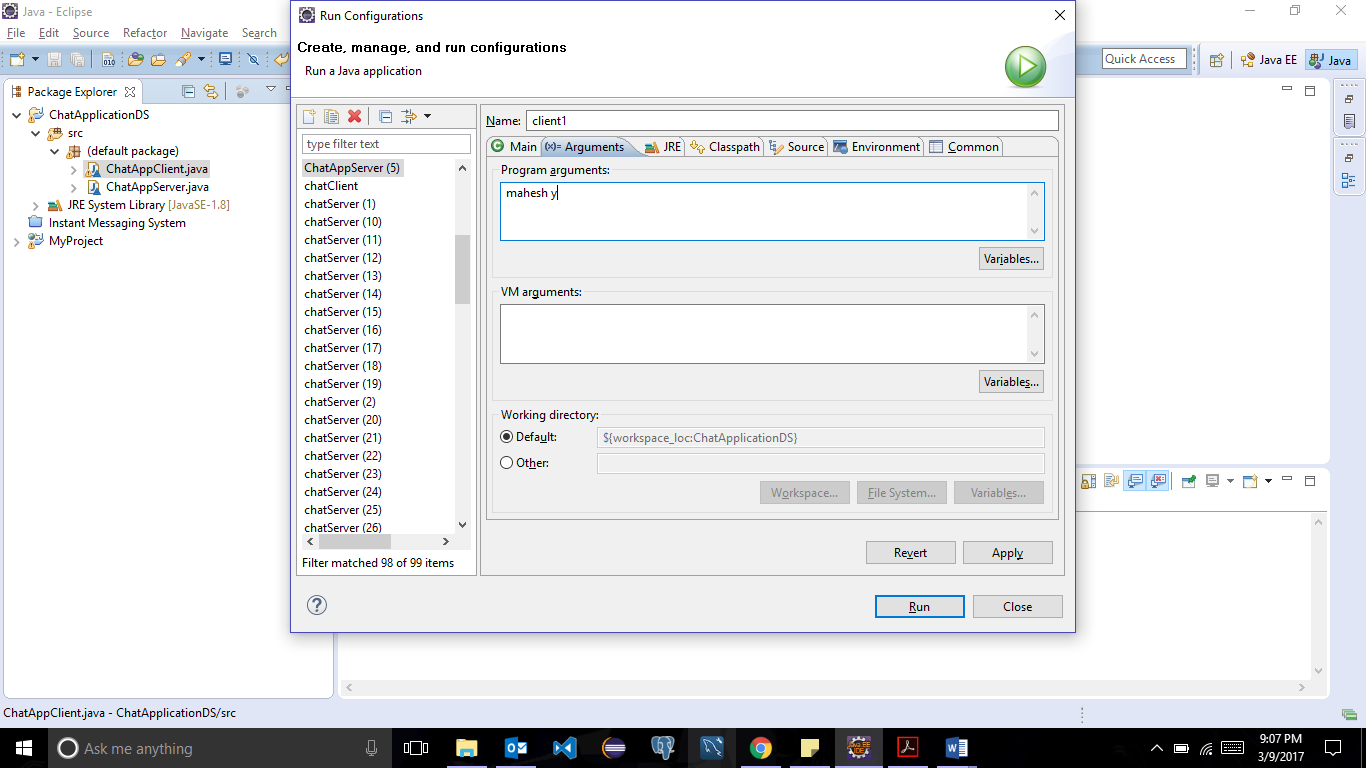
1. Server started :
2. Starting a client :



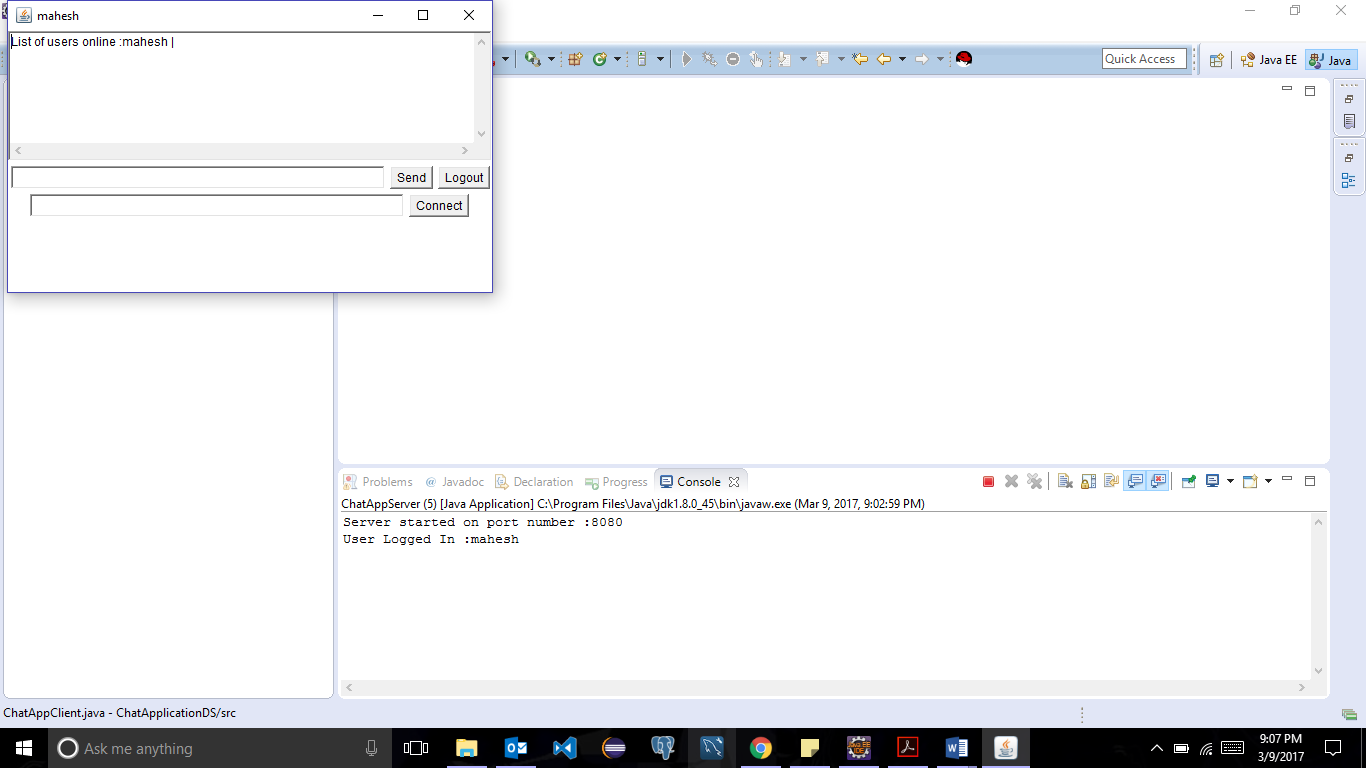
1. Main class should be client class:

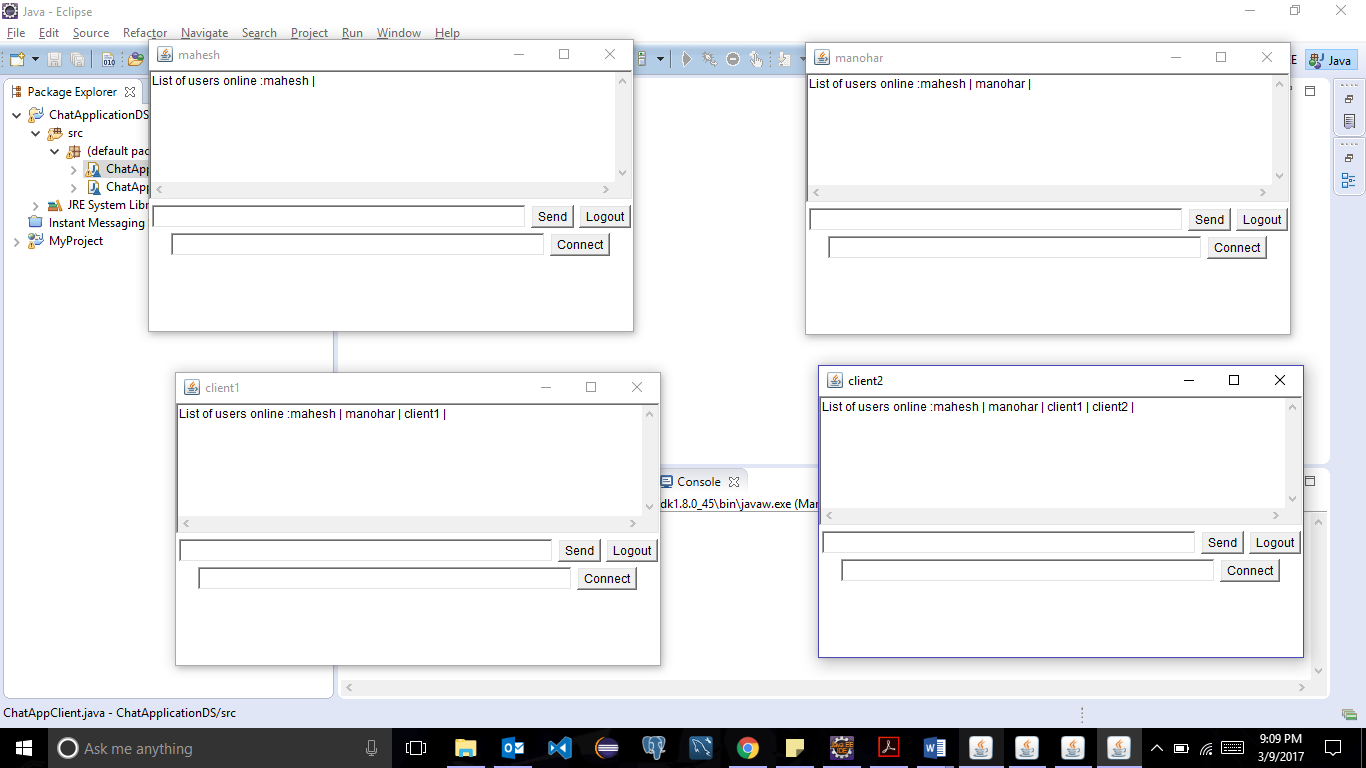


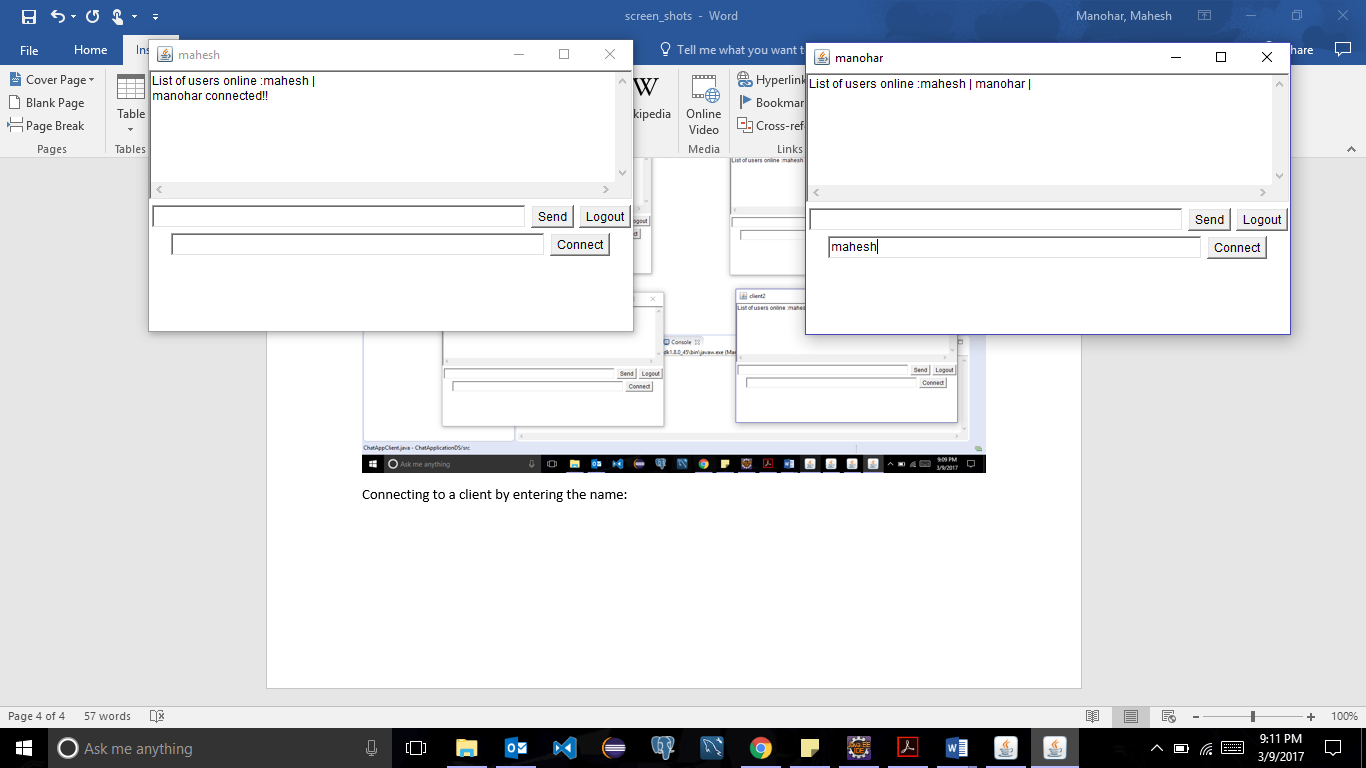
1. Entering arguments (Username and whether clients wants to be visible to other clients(Y/N)):



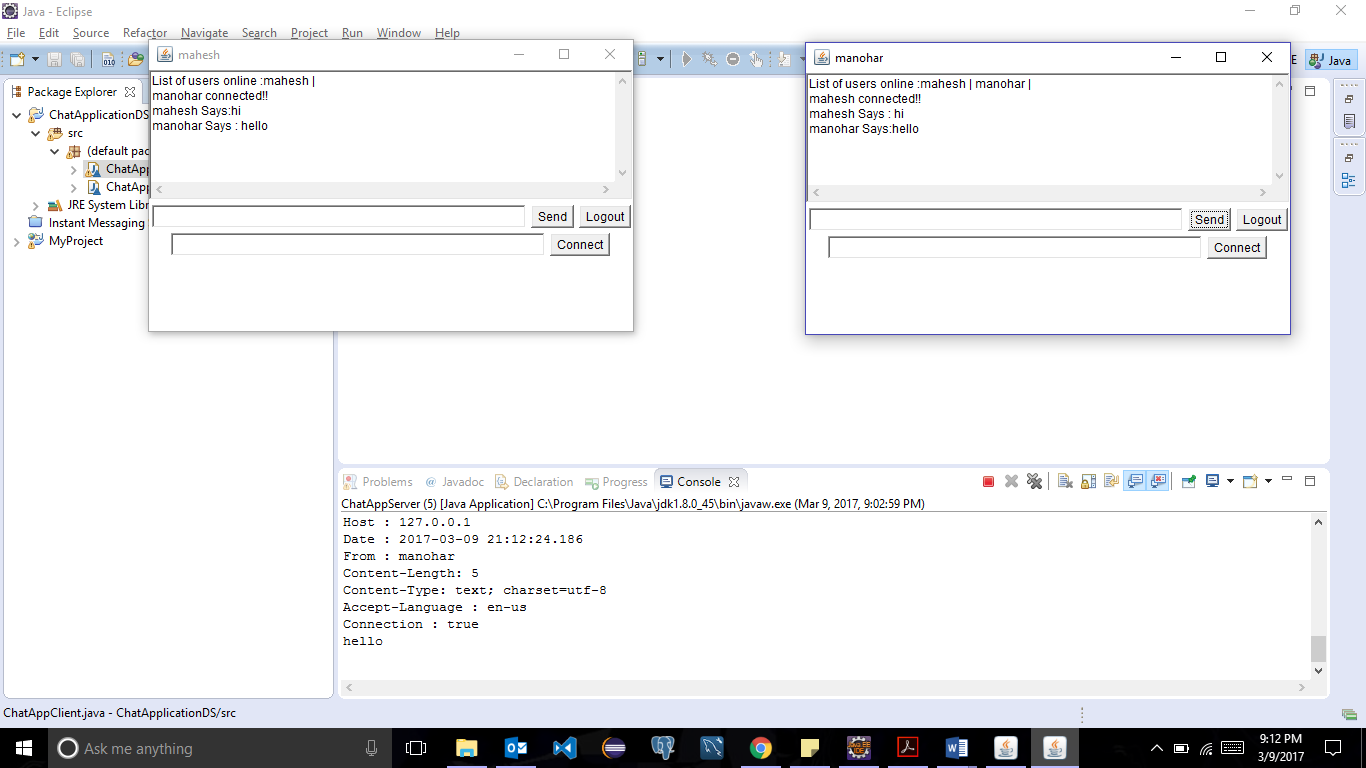
1. Once run is clicked a client window pops up:



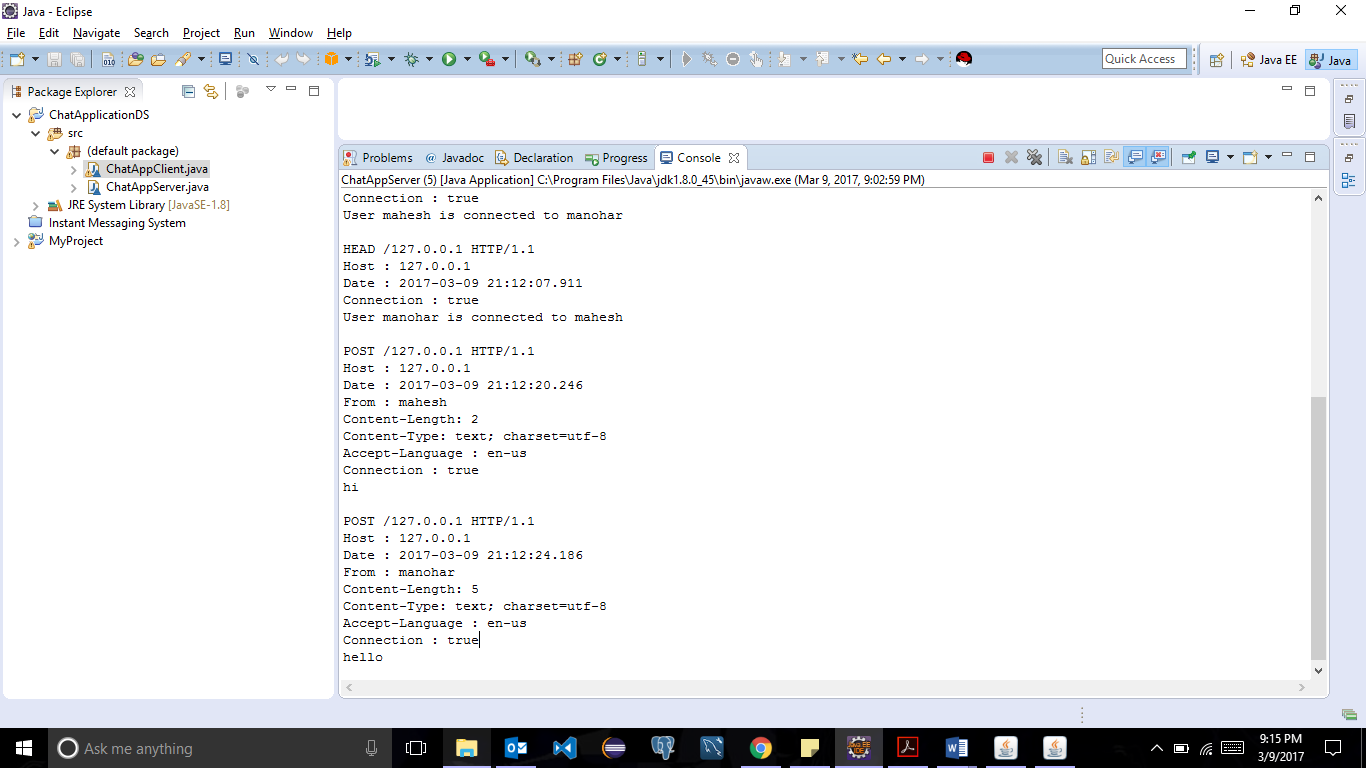
1. Similar process for any number of clients (just by changing the arguments):
2. Connecting to a client by entering the name of the other client(here mahesh is already connected to manohar and manohar has entered the name of mahesh to connect):



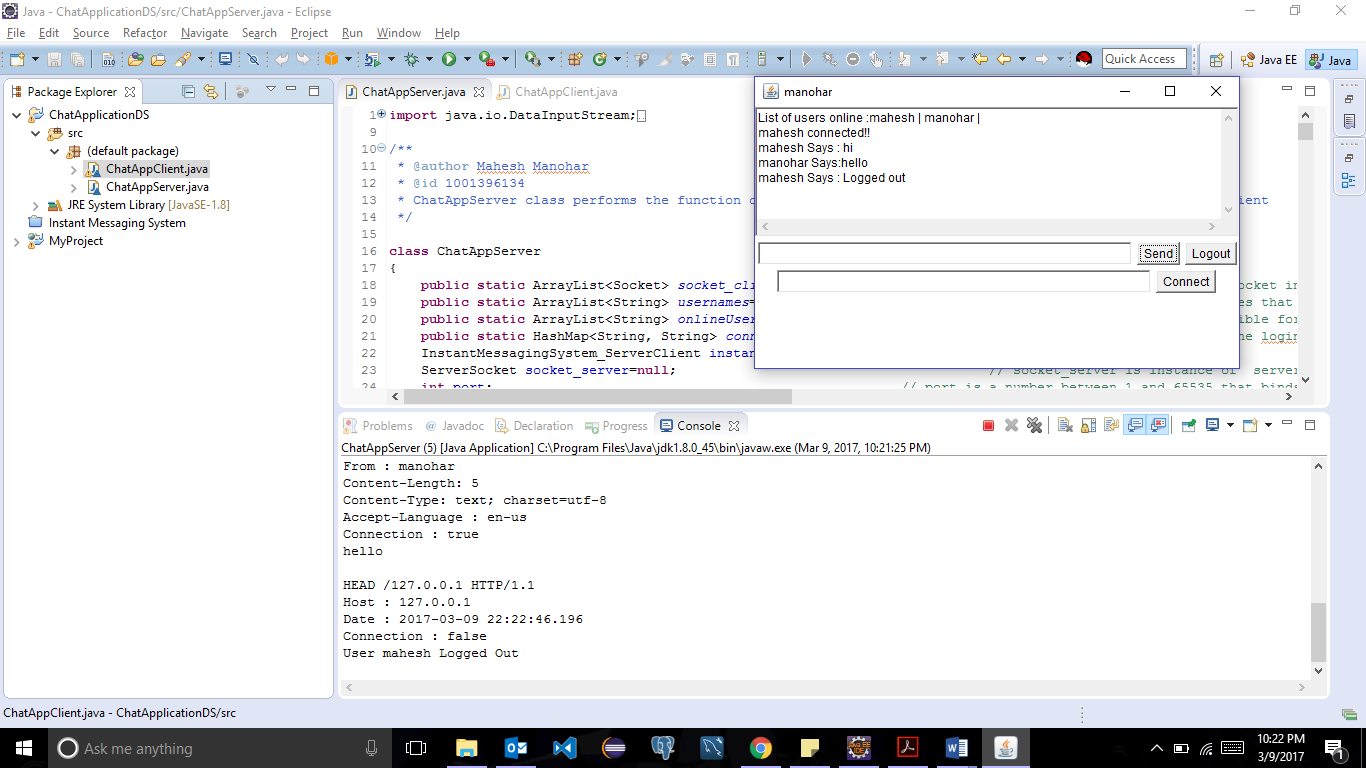
1. All the communications can happen after this :



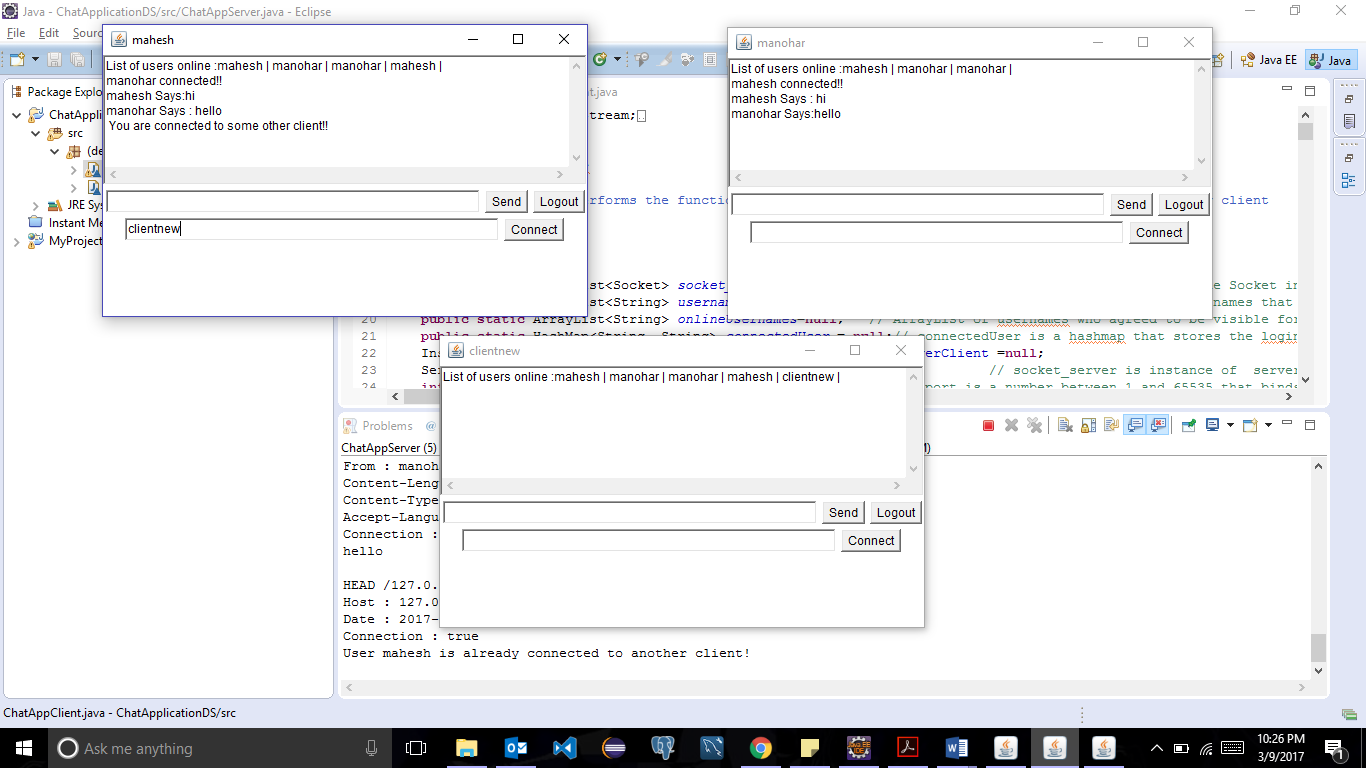
1. All the message logging happens in server console in eclipse:



1. When one of the client clicks the logout button and logs out then the other client is notified:

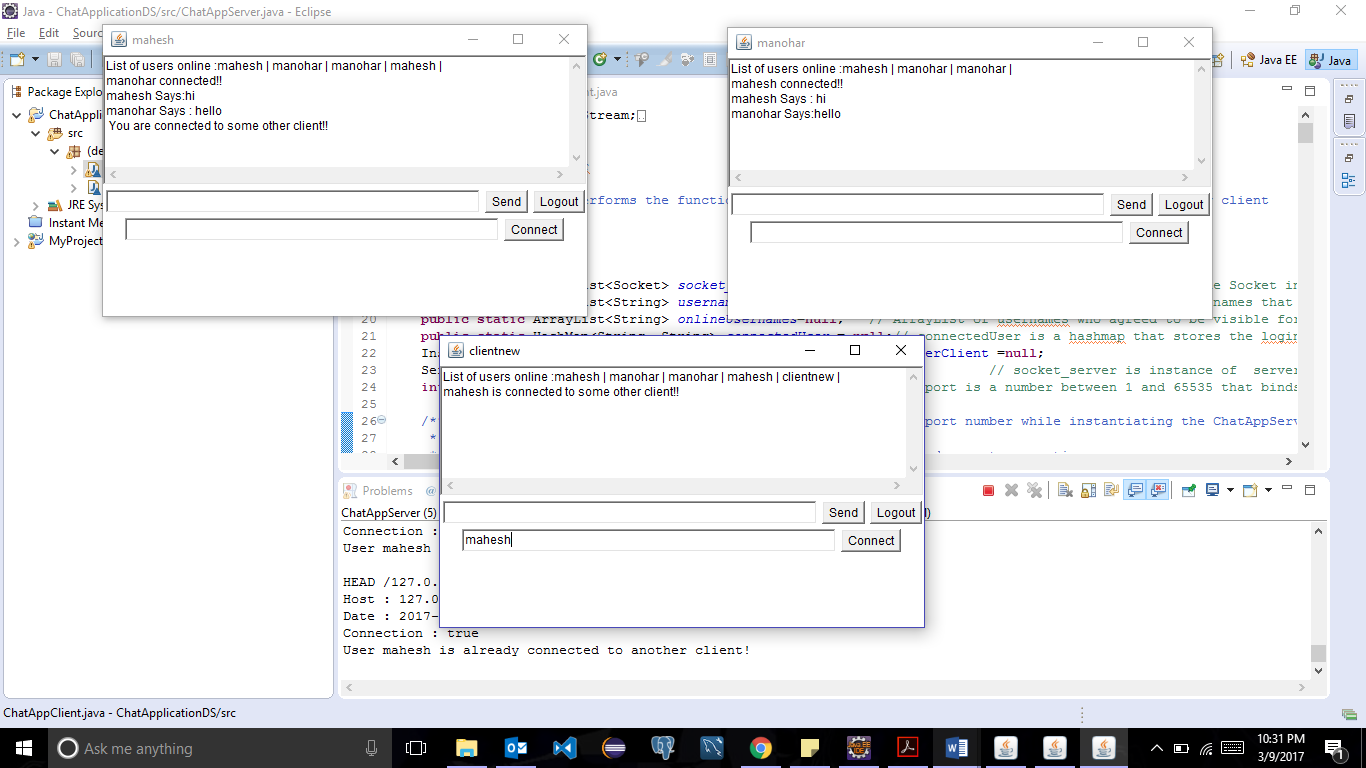


1. When a client is connected to someone else and tries to connected to a second client then it shows a notification saying that client is already connected to some one else:

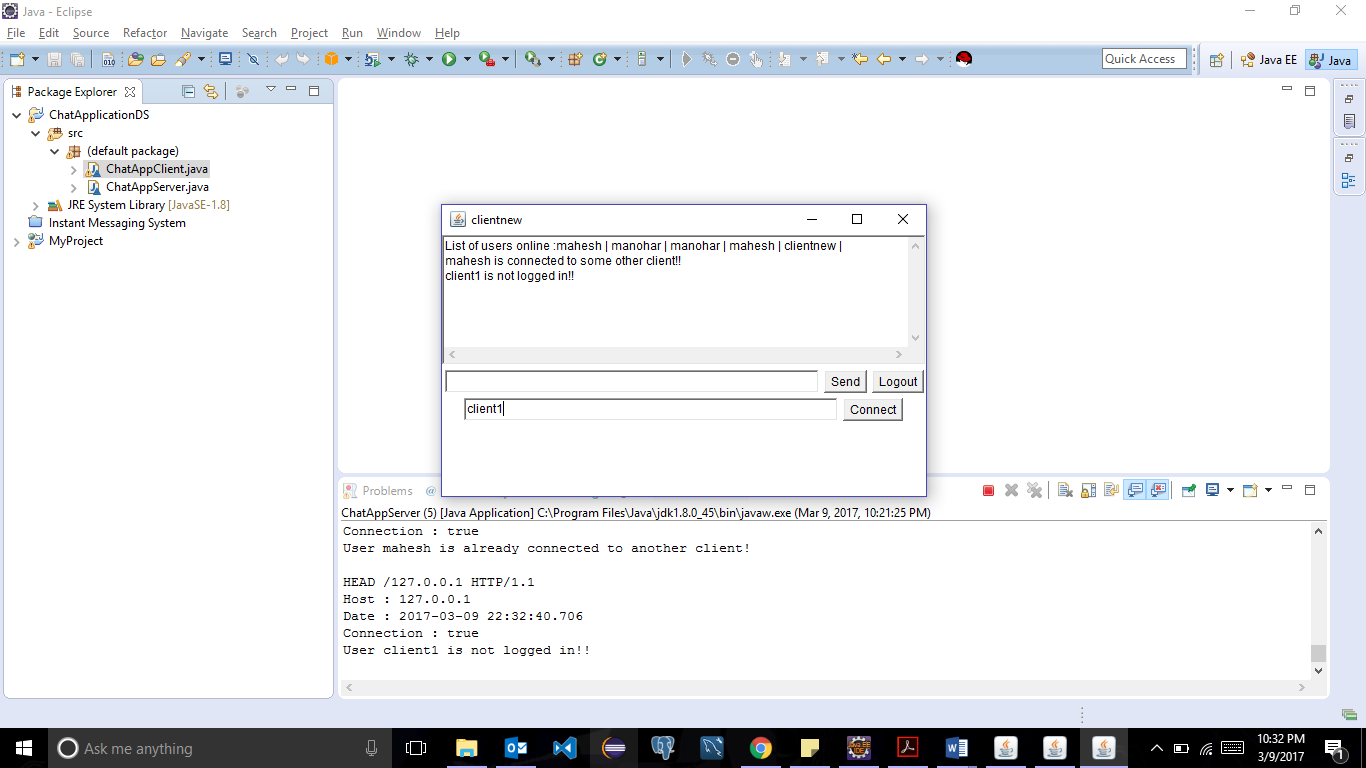


1. If a client gives the name of other client who is not logged in or who is connected to someone else, then the respective error messages are displayed in the below two screen shots:

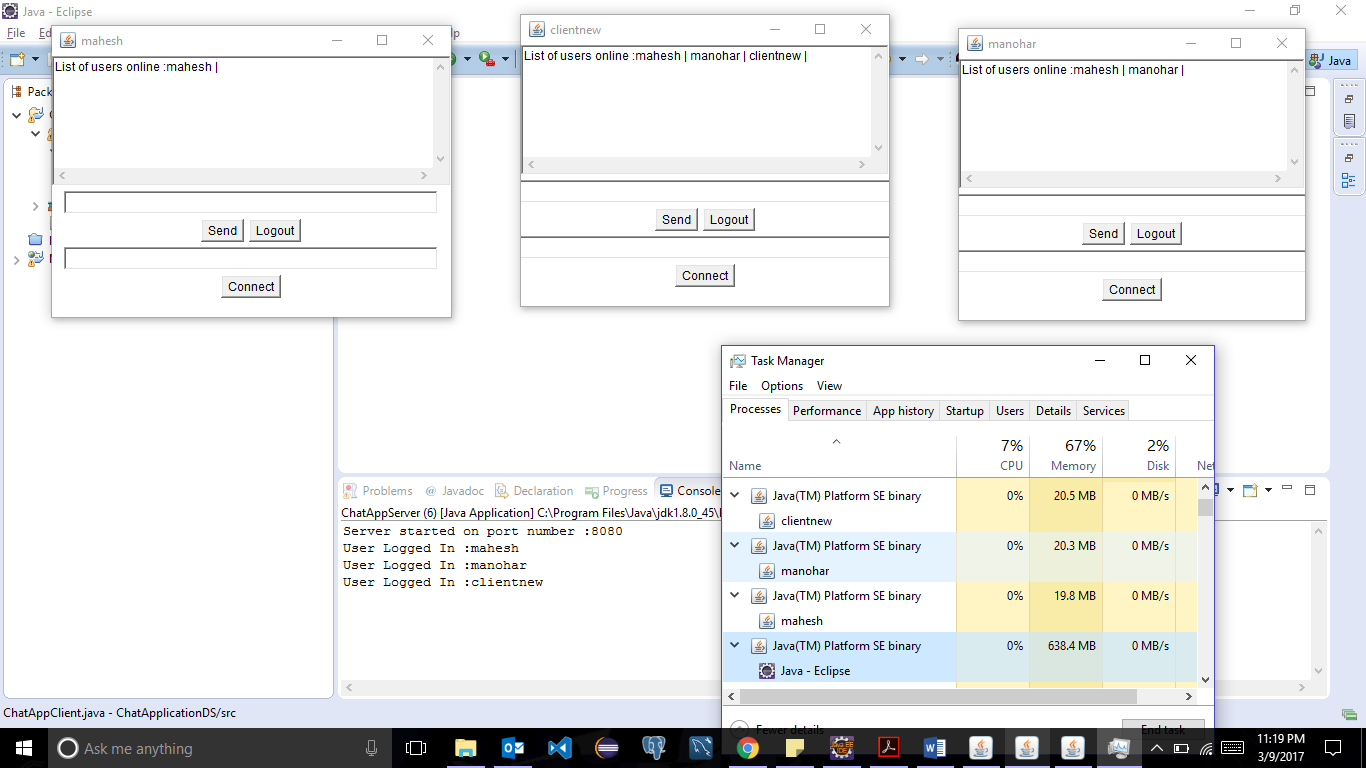
When client is connected to some one else:



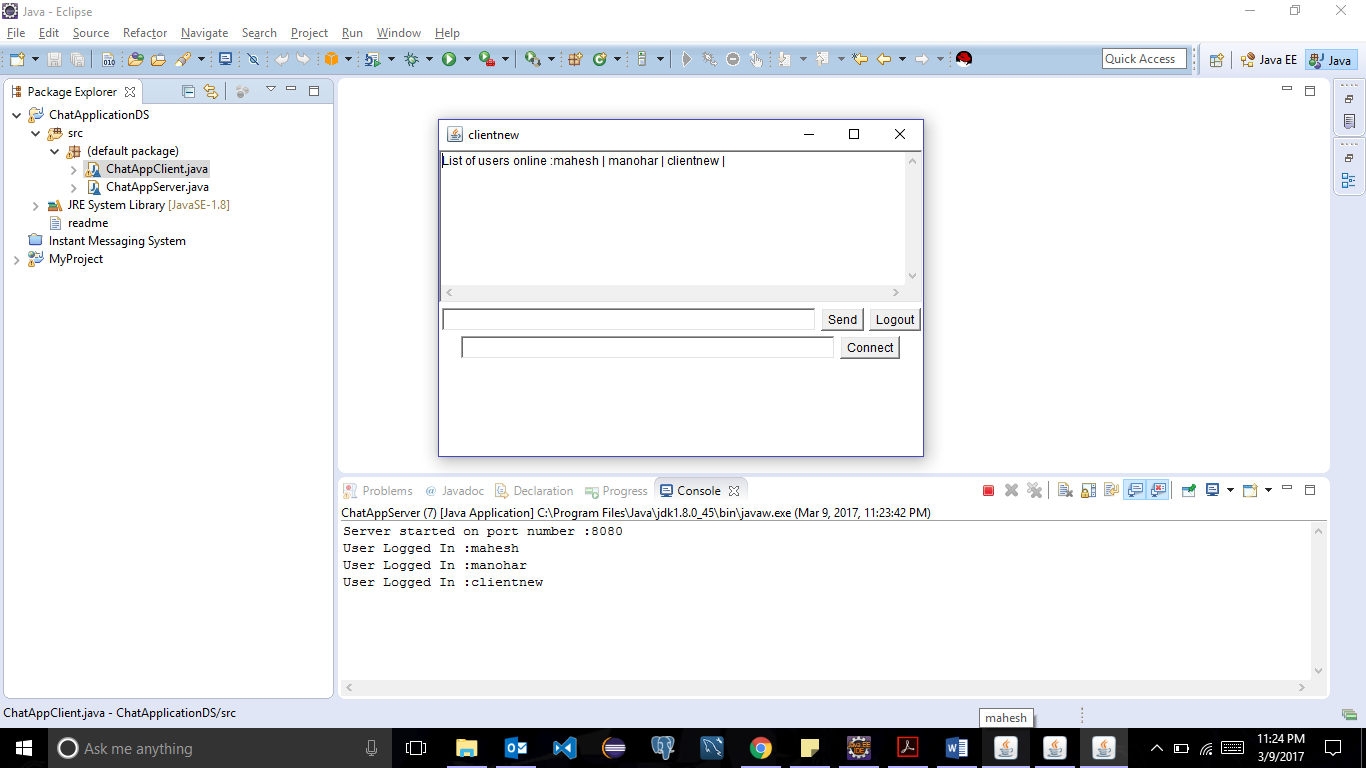
When client to which the connection is requested is not logged in:



1. Showing the Multi-threading part of the application. Each client is a separate thread and server is a separate thread:



1. Showing the list of users who are online when a new user logs in (it includes the current user name as well):



**Limitations:**

1. User can give only first name or last name only. Basically, it has to be just one string as each one will be created a arguments and the second argument is being used for other purpose.

All the requirements are addressed.

Note: Do not stop the server when the client window is up and running. Doing so will make the program go into loop searching for the client socket or thread connection from client side.

**Assumptions:**

1. When the client class is run both the arguments are necessary i.e. Username and Y or N. Reason being both the arguments are checked when the class is run and are used for registering the user. Arguments are supposed to be given with a space in between.
2. A user connects to the other user only with the first name of the latter.
3. It is assumed that all the clients know the names of the client they want to get connected to.

**References :**

<http://www.dailyfreecode.com/Code/socket-multi-client-chat-server-1252.aspx>

http://www.tutorialspoint.com/http/