

Application demonstration



**ASSIGN
BUSTER**

1 Server + 2 The following sequence of actions depicts the chat-based interaction between two and the role of the server in such an interaction.

The server application acts as the bridge between any clients. It keeps track of the various clients by ensuring their online status, maintains a list of Client IP addresses helps in creating sockets between the interacting clients and ensures facilitates the provision for message encryption and NAT traversal. On the other hand, the clients are supposed to receive the encrypted message from the server, extract the message and the sender client details and display all the relevant information to the client user in the requisite format. The sequence of interaction is as shown below:

The server object is initiated itself and starts to listen for incoming clients. A client logs in from the client application and upon proper authentication with the server, the client application's IP address is stored by the server and the client's status is made to 'Online'.

As such, at any given time, the server maintains two basic things, the list of contacts of any client (who are clients themselves), the clients who are online and the IP addresses of the online clients.

Upon successful login, the server sends a message (encrypted) to the customer that contains the list of contacts of the client and their online status.

At the client end, the message is decrypted using predefined logic and the online and offline contacts are displayed to the customer in predefined separate formats (the online contacts are colored while the offline ones are grayed).

The client tries to chat with a client by double clicking on a contact. This opens a dialog and sends an appropriate message giving details of the

intended contact to the server.

The server receives the message and determines the contact. It checks whether the client contact is online, in which case, a socket with the client contact is opened. Otherwise, the server responds to the client with a 'Contact offline at the moment, send offline message' message.

Using the new dialog, the client types a message and chooses the 'Send' option. The message is displayed in the message area and is transmitted to the server.

The server receives the message, identifies the client contact to whom the message is to be sent and using the assigned socket, sends the message to the contact.

The message is received by the client contact and is displayed in the message area on top of all previous messages within the client's chat dialog.

Thus, any two clients continue to send and receive messages through the server application.

In case a contact is offline, the server stores the message in its database. Whenever the client contact comes online, the server initially queries the database to determine if there were any offline messages. If there were any such messages, the server relays the offline messages to the client, wherein they are displayed in a separate dialog with the messages being grouped according to the client who sent the message.