Since all the clients belong to the local system, only one server will be used to serve all the clients. The communication between the server and clients takes place entirely through message queues. All message queues are created using keys generated using the *ftok* function. The same pathname is used for all the keys but different *proj_id's* are used.

The server maintains one message queue with a fixed proj_id of 1. Every message is first sent to this message queue by the clients and the server then process the message and take appropriate action. Every client generates its own proj_id(user id) using a predefined hash function which it sends to the server during the login request. The server identifies each client using this unique proj_id which can also be used to get the message queue id associated with the client. The entire state of the server is maintained in a local data structure which contains the list of active users and the list of groups including the list of members and the pending messages of the group(messages sent to the group which have not timed out and will be sent to new members). The following message format is used for all communication:-

MessageType	Action	Destination	Message	Timeout

1. The message type field holds the user id of the client initiating the request.
2. The action field is used to identify the type of message
3. The destination field is used to either hold the destination user id in a private message or the intended group id in the group message.
4. The message field holds the body of the message
5. The timeout field holds the timeout specified by the user for a group message.

Depending on the value of the action field, there are six types of messages:- login, logout, list group, create group, join group and send message. The messages sent to the server's message queue are processed in First-Come-First-Serve order. The following actions are taken corresponding to each type of message.

a. **Login** - The server adds the user id stored in the message type field to the list of active users in the state.
b. **Logout** - The server removes the user id from the list of active users in the state.
c. **List Group** - The server sends the list of groups stored in the state to the message queue of the client.
d. **Create Group** - The server generates a unique group id and adds this to the list of the groups. It adds the client as the first member of the group and sends the group id to the message queue of the client.
e. **Join Group** - The message field contains the group id which the client wants to join. The server adds the user id to the member list of the group. Next, it iterates through all the pending messages of the group. If a message has timed out, it is deleted otherwise the server sends the message to the message queue of the client.
f. **Send Message** - Group messages are differentiated from private messages using the destination field because group ids begin from 2000 whereas user id can at max be 1000. If it is a private message, the message is forwarded to the message queue of the client with user id specified in the destination field. If it is a group message, the server iterates through the list of members of the group and the message is forwarded to the message queue of each of the members. The request is also added to the pending messages of the groups to be sent to later joining members.

The clients can view messages either by specifying the username or group id. All new messages from the specified source are extracted from the message queue of the client and displayed. Messages are not persistent i.e. after viewing a message once, any record of the message is removed from the system.