

```
1 import java.util.LinkedList;
2
3 public class ChatServer {
4     private static ChatServer instance;
5     private static LinkedList<User> listOfUsers;
6     private static LinkedList<User> listOfBlockedUsers;
7
8     public ChatServer()
9     {
10         listOfUsers = new LinkedList<>();
11         listOfBlockedUsers = new LinkedList<>();
12     }
13
14     public static ChatServer getChatServerInstance()
15     {
16         if (instance == null)
17         {
18             instance = new ChatServer();
19         }
20
21         return instance;
22     }
23
24     public void write(Message givenMessage)
25     {
26         if (!isSenderBlocked(givenMessage))
27         {
28             for(User user : givenMessage.getRecipients())
29             {
30                 user.updatePerceivedChat(givenMessage);
31             }
32         }
33     }
34
35     public User registerUser(String givenUsername)
36     {
37         User newUser = new User(givenUsername);
38         listOfUsers.add(newUser);
39
40         return newUser;
41     }
42
43     public void unregisterUser(String givenUsername)
44     {
45         LinkedList<User> users = listOfUsers;
46         User currentUser = users.pop();
47
48         System.out.println();
49
50         while(!(givenUsername.equals(currentUser.getUsername())))
51         {
52             users.addLast(currentUser);
53             currentUser = users.pop();
54         }
55     }
56 }
```

```

54     }
55
56     currentUser = users.pop();
57 }
58
59 public void addBlockedUser(User userToBlock)
60 {
61     listOfBlockedUsers.add(userToBlock);
62 }
63
64 public boolean isSenderBlocked(Message messageToEvaluate)
65 {
66     boolean returnValue = false;
67
68     for(User user : listOfBlockedUsers)
69     {
70         if (user.getUsername().equals(messageToEvaluate.getSender())) {
71             returnValue = true;
72             break;
73         }
74     }
75
76     return returnValue;
77 }
78
79 public void sendUndoRequest(User userRequesting)
80 {
81     for(User user : listOfUsers)
82     {
83         user.processUndoLastMessage(userRequesting);
84     }
85 }
86 }

```