```java
import java.util.Iterator;
import java.util.LinkedList;

public class User implements Iterable<MessageMomento>{
    private final ChatServer server;
    private final String username;
    private LinkedList<Message> percievedChat;
    private ChatHistory chatHistory;

    public User (String givenUsername)
    {
        this.username = givenUsername;
        this.server = ChatServer.getChatServerInstance();
        percievedChat = new LinkedList<>();
        this.chatHistory = new ChatHistory();
    }

    public void writeToChat(String givenMessage, User[] givenRecipients)
    {
        server.write(new Message(this.username, givenMessage, givenRecipients, "123"));
    }

    public String getUsername()
    {
        return this.username;
    }

    public void requestUndoLastMessage()
    {
        server.sendUndoRequest(this);
    }

    public void processUndoLastMessage(User userToUndoMessage)
    {
        for(MessageMomento momento : this.chatHistory.getWholeHistory())
        {
            if(momento.getChatVersion().getSender() == userToUndoMessage.getUsername())
            {
                this.chatHistory.getWholeHistory().remove(momento);
            }
        }

        for(Message message : this.percievedChat)
        {
            if(message.getSender() == userToUndoMessage.getUsername()) {
                this.percievedChat.remove(message);
            }
        }
    }

    public void updatePercievedChat(Message givenMessage)

    {
        this.percievedChat.addLast(givenMessage);
        chatHistory.addMessageToHistory(givenMessage);
    }

    public void printPercievedChat()
    {
        System.out.println(this.username + " sees: ");
        for (Message message : this.percievedChat)
        {
            System.out.println("Timestamp [" + message.getTimestamp() + "] " +
                    message.getSender() + ": " + message.getText());
        }
    }

    @Override
    public Iterator<MessageMomento> iterator() {
        return new SearchMessagesByUser(this.chatHistory.getWholeHistory());
    }

    class ChatHistory {
        LinkedList<MessageMomento> history;

        public ChatHistory()
        {
            this.history = new LinkedList<>();
        }

        public void addMessageToHistory(Message message)
        {
            history.addFirst(new MessageMomento(message));
        }

        public LinkedList<MessageMomento> getWholeHistory()
        {
            return this.history;
        }

        public MessageMomento getPriorHistory()
        {
            return history.pop();
        }

    }
}
```