File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\User.java

```java
import java.util.Iterator;
import java.util.Vector;

public class User implements IterableByUser {
    private MessageMomento momento;
    private Message message;
    private Vector<Message> receivedMessages;
    private ChatServer chatServer;
    private ChatHistory chatHistory;
    private User otherUser;
    private String userName;

    private Vector<User> blockedUsers;

    public User(String user_name, ChatServer server) {
        chatHistory = new ChatHistory();
        message = new Message(this);
        ConnectToServer(server);
        receivedMessages = new Vector<>();
        blockedUsers = new Vector<>();
        userName = user_name;
    }
    public void SetMessage(String text) {
        message.SetText(text);

    }
    public void Receive(Message message) {

        System.out.println(userName + " received message from " + message.GetUser().
GetUsername());
        message.AddUserReceiver(this);

        receivedMessages.add(message);
        otherUser = message.GetUser();
        chatHistory.AddText(message);
    }
    public void Send() {

        System.out.println("Sending Message to Server...");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }

        chatHistory.AddText(message);
        chatServer.send(message);
    }

    public void Send(User user) {

        System.out.println("Sending Message to Server...");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }

        chatHistory.AddText(message);
        chatServer.send(message, user);
    }

    public void Send(Vector<User> users) {

        System.out.println("Sending Message to Server...");
        try {
            Thread.sleep(1000);
```

```
67          } catch (InterruptedException e) {
68              throw new RuntimeException(e);
69          }
70
71          chatHistory.AddText(message);
72          chatServer.send(message, users);
73      }
74
75      public void Save() {
76          momento = message.Save();
77      }
78
79      public void Restore() {
80          message.Restore(momento);
81      }
82
83      public String GetLastMessage() {
84          return chatHistory.GetLastText().GetText();
85      }
86
87      public String GetUsername() {
88          return userName;
89      }
90
91      public void ConnectToServer(ChatServer chat_server) {
92          chatServer = chat_server;
93          chatServer.addUser(this);
94      }
95
96      public void AddBlockedUser(User user) {
97          blockedUsers.add(user);
98      }
99
100     public Vector<User> GetBlockUsers() {
101         return blockedUsers;
102     }
103
104     public void ViewChatHistory() {
105         chatServer.ViewChatHistory(this);
106     }
107
108     public ChatHistory GetChatHistory() {
109         return chatHistory;
110     }
111
112     public void ViewCHatHistoryOfUser() {
113         chatServer.ViewChatHistory(otherUser);
114     }
115
116     @Override
117     public Iterator iterator(User user_to_search_with) {
118         return chatHistory.iterator(user_to_search_with);
119     }
120 }
121
```

File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\Driver.java

```java
import java.util.Arrays;
import java.util.Iterator;
import java.util.Vector;

public class Driver {
    public static void main(String[] args) {
        ChatServerImpl chat_server = new ChatServerImpl();
        User user_john = new User("John", chat_server);
        User user_amy = new User("Amy", chat_server);
        User user_melody = new User("Melody", chat_server);
        User user_jason = new User("Jason", chat_server);

        user_john.SetMessage("Test");
        user_john.Save();
        user_john.Send();

        user_john.SetMessage("Throwout!");
        user_john.Send();
        user_john.Restore();
        user_john.Send();

        user_john.SetMessage(user_john.GetLastMessage());
        user_john.Send();

        user_amy.SetMessage("Hello, John!");
        user_amy.Send(user_john);

        user_john.AddBlockedUser(user_amy);
        user_amy.Send(user_john);

        user_john.Send(new Vector<User>(Arrays.asList(user_amy, user_jason)));
        user_john.SetMessage("Every else but Amy!");
        user_john.Send();

        Iterator chat_history_iter = user_john.iterator(user_amy);

        System.out.println("Amys messages from Johns history...");
        while(chat_history_iter.hasNext()) {
            Message msg = (Message) chat_history_iter.next();
            System.out.println(msg.GetText());
        }
    }
}
```

File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\Message.java

```java
import java.util.Vector;

public class Message {
    private String text;
    private final User user;
    private Vector<User> receivingUsers;


    Message(User user) {
        this.user = user;
        receivingUsers = new Vector<>();
    }
    User GetUser() {
        return user;
    }
    public void SetText(String set_text) {
        text = set_text;
    }

    public String GetText() {
        return text;
    }

    public MessageMomento Save() {
        System.out.println("Memento Created!");
        return new MessageMomento(text);
    }

    public void Restore(MessageMomento momento) {
        System.out.println("Memento Restored!");
        text = momento.GetText();
    }

    public void AddUserReceiver(User receiver) {
        receivingUsers.add(receiver);
    }

    Vector<User> GetReceivers() {
        return receivingUsers;
    }
}

```

File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\ChatServer.java

```java
import java.util.Vector;

public interface ChatServer {
    void send(Message message);
    void send(Message message, User user);
    void send(Message message, Vector<User> users);
    void addUser(User user);
    void ViewChatHistory(User user);
}

```

File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\ChatHistory.java

```java
import java.util.Iterator;
import java.util.Vector;

public class ChatHistory implements IterableByUser {
    private final Vector<Message> history;

    public ChatHistory() {
        history = new Vector<>();
    }

    public void AddText(Message text) {
        history.add(text);
    }

    public Message GetLastText() {
        Message last_text = null;

        if(!history.isEmpty()) {
            last_text = history.get(history.size()-1);
        }

        return last_text;
    }

    Vector<Message> GetHistory() {
        return history;
    }

    @Override
    public Iterator iterator(User user_to_search_with) {
        return new SearchMessageByUser(this, user_to_search_with);
    }
}

```

File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\ChatServerImpl.java

```java
import java.util.Vector;

public class ChatServerImpl implements ChatServer {
    private Vector<User> users;

    ChatServerImpl() {
        users = new Vector<>();
    }

    @Override
    public void addUser(User user) {
        users.add(user);
    }

    @Override
    public void ViewChatHistory(User user) {
        ChatHistory chat_history = user.GetChatHistory();

        System.out.println("Chat history of " + user.GetUsername() + ":");
        for (Message msg : chat_history.GetHistory()) {
            System.out.println(msg.GetText());
        }
    }

    @Override
    public void send(Message message) {
        boolean users_are_sent_a_message = false;
        for (User user: users) {
            if (!message.GetUser().equals(user)) {

                boolean user_is_blocked = false;
                User blocked_user = null;
                for (User buser : message.GetUser().GetBlockUsers()) {
                    if (user.equals(buser)) {
                        user_is_blocked = true;
                        blocked_user = buser;
                        break;
                    }
                }

                if (!user_is_blocked) {
                    user.Receive(message);
                    System.out.println("User (" + message.GetUser().GetUsername() + ") sent: " + message.GetText() + " to " + user.GetUsername());
                }
                else {
                    System.out.println("User (" + user.GetUsername() + ") has blocked " + blocked_user.GetUsername());
                }

                users_are_sent_a_message = true;
            }
        }

        if (!users_are_sent_a_message) {
            System.out.println("No other users on server to send to!");
        }
    }

    @Override
    public void send(Message message, User user) {
        boolean users_are_sent_a_message = false;
        for (User connected_user : users) {
            if (connected_user.equals(user)) {

                boolean user_is_blocked = false;
                User blocked_user = null;
```

```
66                  for (User buser : user.GetBlockUsers()) {
67                      if (message.GetUser().equals(buser)) {
68                          user_is_blocked = true;
69                          blocked_user = buser;
70                          break;
71                      }
72                  }
73
74                  if (!user_is_blocked) {
75                      user.Receive(message);
76                      System.out.println("User (" + message.GetUser().GetUsername() + ")
   sent: " + message.GetText() + " to " + user.GetUsername());
77                  }
78                  else {
79                      System.out.println("User (" + user.GetUsername() + ") has blocked "
    + blocked_user.GetUsername());
80                  }
81
82                  users_are_sent_a_message = true;
83                  break;
84              }
85          }
86
87          if (!users_are_sent_a_message) {
88              System.out.println("No other users on server to send to!");
89          }
90      }
91
92      @Override
93      public void send(Message message, Vector<User> users) {
94          for (User user : users) {
95              send(message, user);
96          }
97      }
98  }
99
```

File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\IterableByUser.java

```java
import java.util.Iterator;

public interface IterableByUser {
    Iterator iterator(User user_to_search_with);
}

```

File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\MessageMomento.java

```java
public class MessageMomento {
    private final String text;

    public MessageMomento(String set_text) {
        text = set_text;
    }

    public String GetText() {
        return text;
    }
}

```

File - C:\msys64\home\Sean\soft_eng_examples\Assignment_5\java\momento\src\SearchMessageByUser.java

```java
import java.util.Iterator;

public class SearchMessageByUser implements Iterator {

    private int counter = 0;
    private ChatHistory chatHistory;
    private Iterator iter;
    User user;

    SearchMessageByUser(ChatHistory chat_history, User user_to_search_with) {
        chatHistory = chat_history;
        user = user_to_search_with;

        Iterator<Message> it_next = chat_history.GetHistory().iterator();
        Iterator<Message> it_prev = chat_history.GetHistory().iterator();
        while(it_next.hasNext()) {
            Message msg = it_next.next();
            counter++;

            if (msg.GetUser().equals(user_to_search_with) || msg.GetReceivers().contains(
user))
            {
                iter = it_prev;
                break;
            }

            it_prev.next();
        }
    }

    @Override
    public boolean hasNext() {
        boolean found = false;

        Iterator it_check = chatHistory.GetHistory().iterator();
        for(int i=0; i<counter; i++) {
            it_check.next();
        }

        while(iter.hasNext()) {
            Message msg = (Message) iter.next();

            if (msg.GetUser().equals(user) || msg.GetReceivers().contains(user))
            {
                iter = it_check;
                found = true;
                break;
            }

            it_check.next();
        }

        return found;
    }

    @Override
    public Object next() {
        Message msg = null;

        if (hasNext()) {
            counter++;
            Message check_msg = (Message) iter.next();

            if (check_msg.GetUser().equals(user) || check_msg.GetReceivers().contains(
user)) {
                msg = check_msg;
            }
```

```
66          }
67
68          return msg;
69      }
70
71  }
72
```