Chat App (Memento, Mediator and Iterator) Unit Test

```java
package edu.cpp.cs5800.chatapp;

import org.junit.jupiter.api.Test;

import java.util.Arrays;

import static org.junit.jupiter.api.Assertions.*;

public class UserTest {
    ChatServer chatServer = new ChatServer();
    User user1 = new User("testUser1", chatServer);
    User user2 = new User("testUser2", chatServer);

    @Test
    public void testGetUserName() {
        assertEquals("testUser1", user1.getUsername());
    }

    @Test
    public void testSendMessage() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        assertEquals(1, user1.sendMessage("test 123", Arrays.asList(user2)));

        Message message = chatServer.getLastMessage(user1);
        assertEquals("test 123", message.getContent());
        assertEquals(user1, message.getSender());
    }

    @Test
    public void testUndoLastMessage() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        user1.sendMessage("test 123", Arrays.asList(user2));
        assertTrue(user1.undoLastMessage());
    }

    @Test
    public void testGetLastMessage() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        user1.sendMessage("test 123", Arrays.asList(user2));
        String expected = String.format("%s sent to %s: %s", user1.getUsername(), Arrays.asList(user2), "test 123");
        assertEquals(expected, user1.getLastMessage());
    }

    @Test
    public void testBlockUser() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        assertTrue(user1.blockUser(user2));
    }

    @Test
    public void testUnblockUser() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        user1.blockUser(user2);

        assertTrue(user1.unblockUser(user2));
    }

}

```

```java
package edu.cpp.cs5800.chatapp;

import org.junit.jupiter.api.Test;

import java.util.Arrays;
import java.util.Date;
import java.util.List;

import static org.junit.jupiter.api.Assertions.assertEquals;

public class MessageTest {
    ChatServer chatServer = new ChatServer();
    User user1 = new User("testUser1", chatServer);
    User user2 = new User("testUser2", chatServer);
    Message message = new Message("test content", Arrays.asList(user2), user1);

    @Test
    public void testGetContent() {
        assertEquals("test content", message.getContent());
    }

    @Test
    public void testGetRecipients() {
        assertEquals(Arrays.asList(user2), message.getRecipients());
    }

    @Test
    public void testGetSender() {
        assertEquals(user1, message.getSender());
    }

    @Test
    public void testTakeSnapshot() {
       Message.Memento memento =  message.takeSnapshot();
       assertEquals(memento.getContent(), message.getContent());
       assertEquals(memento.getTimestamp(), message.getTimestamp());
    }

    @Test
    public void testRestoreSnapshot() {
        Message.Memento memento =  message.takeSnapshot();
        Message message2 = Message.restoreSnapshot(memento);
        assertEquals(message.getId(), message2.getId());
        assertEquals(message.getContent(), message2.getContent());
        assertEquals(message.getSender(), message2.getSender());
        assertEquals(message.getRecipients(), message2.getRecipients());
        assertEquals(message.getTimestamp(), message2.getTimestamp());
    }
}
```

Chat App (Memento, Mediator and Iterator) Unit Test

```java
package edu.cpp.cs5800.chatapp;

import org.junit.jupiter.api.Test;

import java.util.Arrays;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertTrue;

public class ChatServerTest {
    ChatServer chatServer = new ChatServer();
    User user1 = new User("testUser1", chatServer);
    User user2 = new User("testUser2", chatServer);

    @Test
    public void testRegisterUser() {
        assertTrue(chatServer.registerUser(user1));
    }

    @Test
    public void testUnregisterUser() {
        chatServer.registerUser(user1);
        assertTrue(chatServer.unregisterUser(user1));
    }

    @Test
    public void testBlockUser() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);

        assertTrue(chatServer.blockUser(user1, user2));
    }

    @Test
    public void testUnblockUser() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        chatServer.blockUser(user1, user2);

        assertTrue(chatServer.unblockUser(user1, user2));
    }

    @Test
    public void testSendMessage() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        Message message = new Message("test message", Arrays.asList(user2), user1);
        assertEquals(1, chatServer.sendMessage(message));
    }

    @Test
    public void testUnsendMessage() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        Message message = new Message("test message", Arrays.asList(user2), user1);
        chatServer.sendMessage(message);
        assertTrue(chatServer.unsendMessage(message));
    }

    @Test
    public void testGetLastMessage() {
        chatServer.registerUser(user1);
        chatServer.registerUser(user2);
        Message message = new Message("test message", Arrays.asList(user2), user1);
        chatServer.sendMessage(message);
        assertEquals(message, chatServer.getLastMessage(user1));
    }
}

```

Chat App (Memento, Mediator and Iterator) Unit Test

```java
package edu.cpp.cs5800.chatapp;

import org.junit.jupiter.api.Test;

import java.util.Arrays;
import java.util.Iterator;
import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

public class ChatHistoryTest {
    ChatServer chatServer = new ChatServer();
    User user1 = new User("testUser1", chatServer);
    User user2 = new User("testUser2", chatServer);
    User user3 = new User("testUser3", chatServer);

    ChatHistory chatHistory = new ChatHistory();

    @Test
    public void testAddMessage() {
        Message message = new Message("test content", null, user1);
        assertTrue(chatHistory.addMessage(message));
        assertEquals(message, chatHistory.getLastMessage(user1));
    }

    @Test
    public void testGetLastMessage() {
        Message message2 = new Message("test content 2", null, user1);
        chatHistory.addMessage(message2);
        assertEquals(message2, chatHistory.getLastMessage(user1));
    }

    @Test
    public void testGetHistory() {
        Message message1 = new Message("test content 1", Arrays.asList(user2), user1);
        Message message2 = new Message("test content 2", Arrays.asList(user3), user2);
        Message message3 = new Message("test content 3", Arrays.asList(user2), user1);

        chatHistory.addMessage(message1);
        chatHistory.addMessage(message2);
        chatHistory.addMessage(message3);

        List<Message> expected = Arrays.asList(message1, message3);
        assertEquals(expected, chatHistory.getHistory(user1));
    }

    @Test
    public void testRemoveMessage() {
        Message message1 = new Message("test content 1", Arrays.asList(user2), user1);
        Message message2 = new Message("test content 2", Arrays.asList(user3), user2);
        Message message3 = new Message("test content 3", Arrays.asList(user2), user1);

        chatHistory.addMessage(message1);
        chatHistory.addMessage(message2);
        chatHistory.addMessage(message3);

        assertTrue(chatHistory.removeMessage(message3));
        assertEquals(message1, chatHistory.getLastMessage(user1));
    }

    @Test
    public void testIterator() {
        Message message1 = new Message("test content 1", null, user1);
        chatHistory.addMessage(message1);
        assertInstanceOf(Iterator.class, chatHistory.iterator(user1));
    }
}
```

```java
package edu.cpp.cs5800.chatapp;

import org.junit.jupiter.api.Test;

import java.util.Iterator;

import static org.junit.jupiter.api.Assertions.*;

public class ChatIteratorTest {
    ChatServer chatServer = new ChatServer();
    User user1 = new User("testUser1", chatServer);

    ChatHistory chatHistory = new ChatHistory();

    @Test
    public void testHasNext() {
        Message message1 = new Message("test content 1", null, user1);
        chatHistory.addMessage(message1);
        Iterator<Message> iterator = chatHistory.iterator(user1);
        assertTrue(iterator.hasNext());
    }

    @Test
    public void testNext() {
        Message message1 = new Message("test content 1", null, user1);
        chatHistory.addMessage(message1);
        Iterator<Message> iterator = chatHistory.iterator(user1);
       assertEquals(message1, iterator.next());
    }
}

```