

```

1 package edu.cpp.cs5800.chatapp;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.Arrays;
6
7 import static org.junit.jupiter.api.Assertions.*;
8
9 public class UserTest {
10     ChatServer chatServer = new ChatServer();
11     User user1 = new User("testUser1", chatServer);
12     User user2 = new User("testUser2", chatServer);
13
14     @Test
15     public void testGetUserName() {
16         assertEquals("testUser1", user1.getUsername());
17     }
18
19     @Test
20     public void testSendMessage() {
21         chatServer.registerUser(user1);
22         chatServer.registerUser(user2);
23         assertEquals(1, user1.sendMessage("test 123", Arrays.asList(user2)));
24
25         Message message = chatServer.getLastMessage(user1);
26         assertEquals("test 123", message.getContent());
27         assertEquals(user1, message.getSender());
28     }
29
30     @Test
31     public void testUndoLastMessage() {
32         chatServer.registerUser(user1);
33         chatServer.registerUser(user2);
34         user1.sendMessage("test 123", Arrays.asList(user2));
35         assertTrue(user1.undoLastMessage());
36     }
37
38     @Test
39     public void testGetLastMessage() {
40         chatServer.registerUser(user1);
41         chatServer.registerUser(user2);
42         user1.sendMessage("test 123", Arrays.asList(user2));
43         String expected = String.format("%s sent to %s: %s", user1.getUsername(), Arrays.
asList(user2), "test 123");
44         assertEquals(expected, user1.getLastMessage());
45     }
46
47     @Test
48     public void testBlockUser() {
49         chatServer.registerUser(user1);
50         chatServer.registerUser(user2);
51         assertTrue(user1.blockUser(user2));
52     }

```

```
53
54 @Test
55 public void testUnblockUser() {
56     chatServer.registerUser(user1);
57     chatServer.registerUser(user2);
58     user1.blockUser(user2);
59
60     assertTrue(user1.unblockUser(user2));
61 }
62
63 }
64
```

```
1 package edu.cpp.cs5800.chatapp;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.Arrays;
6 import java.util.Date;
7 import java.util.List;
8
9 import static org.junit.jupiter.api.Assertions.assertEquals;
10
11 public class MessageTest {
12     ChatServer chatServer = new ChatServer();
13     User user1 = new User("testUser1", chatServer);
14     User user2 = new User("testUser2", chatServer);
15     Message message = new Message("test content", Arrays.asList(user2), user1);
16
17     @Test
18     public void testGetContent() {
19         assertEquals("test content", message.getContent());
20     }
21
22     @Test
23     public void testGetRecipients() {
24         assertEquals(Arrays.asList(user2), message.getRecipients());
25     }
26
27     @Test
28     public void testGetSender() {
29         assertEquals(user1, message.getSender());
30     }
31
32     @Test
33     public void testTakeSnapshot() {
34         Message.Memento memento = message.takeSnapshot();
35         assertEquals(memento.getContent(), message.getContent());
36         assertEquals(memento.getTimestamp(), message.getTimestamp());
37     }
38
39     @Test
40     public void testRestoreSnapshot() {
41         Message.Memento memento = message.takeSnapshot();
42         Message message2 = Message.restoreSnapshot(memento);
43         assertEquals(message.getId(), message2.getId());
44         assertEquals(message.getContent(), message2.getContent());
45         assertEquals(message.getSender(), message2.getSender());
46         assertEquals(message.getRecipients(), message2.getRecipients());
47         assertEquals(message.getTimestamp(), message2.getTimestamp());
48     }
49 }
50
```

```
1 package edu.cpp.cs5800.chatapp;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.Arrays;
6
7 import static org.junit.jupiter.api.Assertions.assertEquals;
8 import static org.junit.jupiter.api.Assertions.assertTrue;
9
10 public class ChatServerTest {
11     ChatServer chatServer = new ChatServer();
12     User user1 = new User("testUser1", chatServer);
13     User user2 = new User("testUser2", chatServer);
14
15     @Test
16     public void testRegisterUser() {
17         assertTrue(chatServer.registerUser(user1));
18     }
19
20     @Test
21     public void testUnregisterUser() {
22         chatServer.registerUser(user1);
23         assertTrue(chatServer.unregisterUser(user1));
24     }
25
26     @Test
27     public void testBlockUser() {
28         chatServer.registerUser(user1);
29         chatServer.registerUser(user2);
30
31         assertTrue(chatServer.blockUser(user1, user2));
32     }
33
34     @Test
35     public void testUnblockUser() {
36         chatServer.registerUser(user1);
37         chatServer.registerUser(user2);
38         chatServer.blockUser(user1, user2);
39
40         assertTrue(chatServer.unblockUser(user1, user2));
41     }
42
43     @Test
44     public void testSendMessage() {
45         chatServer.registerUser(user1);
46         chatServer.registerUser(user2);
47         Message message = new Message("test message", Arrays.asList(user2), user1);
48         assertEquals(1, chatServer.sendMessage(message));
49     }
50
51     @Test
52     public void testUnsendMessage() {
53         chatServer.registerUser(user1);
```

## Chat App (Memento, Mediator and Iterator) Unit Test

```
54     chatServer.registerUser(user2);
55     Message message = new Message("test message", Arrays.asList(user2), user1);
56     chatServer.sendMessage(message);
57     assertTrue(chatServer.unsendMessage(message));
58 }
59
60 @Test
61 public void testGetLastMessage() {
62     chatServer.registerUser(user1);
63     chatServer.registerUser(user2);
64     Message message = new Message("test message", Arrays.asList(user2), user1);
65     chatServer.sendMessage(message);
66     assertEquals(message, chatServer.getLastMessage(user1));
67 }
68 }
69
```

## Chat App (Memento, Mediator and Iterator) Unit Test

```
1 package edu.cpp.cs5800.chatapp;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.Arrays;
6 import java.util.Iterator;
7 import java.util.List;
8
9 import static org.junit.jupiter.api.Assertions.*;
10
11 public class ChatHistoryTest {
12     ChatServer chatServer = new ChatServer();
13     User user1 = new User("testUser1", chatServer);
14     User user2 = new User("testUser2", chatServer);
15     User user3 = new User("testUser3", chatServer);
16
17     ChatHistory chatHistory = new ChatHistory();
18
19     @Test
20     public void testAddMessage() {
21         Message message = new Message("test content", null, user1);
22         assertTrue(chatHistory.addMessage(message));
23         assertEquals(message, chatHistory.getLastMessage(user1));
24     }
25
26     @Test
27     public void testGetLastMessage() {
28         Message message2 = new Message("test content 2", null, user1);
29         chatHistory.addMessage(message2);
30         assertEquals(message2, chatHistory.getLastMessage(user1));
31     }
32
33     @Test
34     public void testGetHistory() {
35         Message message1 = new Message("test content 1", Arrays.asList(user2), user1);
36         Message message2 = new Message("test content 2", Arrays.asList(user3), user2);
37         Message message3 = new Message("test content 3", Arrays.asList(user2), user1);
38
39         chatHistory.addMessage(message1);
40         chatHistory.addMessage(message2);
41         chatHistory.addMessage(message3);
42
43         List<Message> expected = Arrays.asList(message1, message3);
44         assertEquals(expected, chatHistory.getHistory(user1));
45     }
46
47     @Test
48     public void testRemoveMessage() {
49         Message message1 = new Message("test content 1", Arrays.asList(user2), user1);
50         Message message2 = new Message("test content 2", Arrays.asList(user3), user2);
51         Message message3 = new Message("test content 3", Arrays.asList(user2), user1);
52
53         chatHistory.addMessage(message1);
```

## Chat App (Memento, Mediator and Iterator) Unit Test

```
54     chatHistory.addMessage(message2);
55     chatHistory.addMessage(message3);
56
57     assertTrue(chatHistory.removeMessage(message3));
58     assertEquals(message1, chatHistory.getLastMessage(user1));
59 }
60
61 @Test
62 public void testIterator() {
63     Message message1 = new Message("test content 1", null, user1);
64     chatHistory.addMessage(message1);
65     assertInstanceOf(Iterator.class, chatHistory.iterator(user1));
66 }
67 }
68
```

```
1 package edu.cpp.cs5800.chatapp;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.Iterator;
6
7 import static org.junit.jupiter.api.Assertions.*;
8
9 public class ChatIteratorTest {
10     ChatServer chatServer = new ChatServer();
11     User user1 = new User("testUser1", chatServer);
12
13     ChatHistory chatHistory = new ChatHistory();
14
15     @Test
16     public void testHasNext() {
17         Message message1 = new Message("test content 1", null, user1);
18         chatHistory.addMessage(message1);
19         Iterator<Message> iterator = chatHistory.iterator(user1);
20         assertTrue(iterator.hasNext());
21     }
22
23     @Test
24     public void testNext() {
25         Message message1 = new Message("test content 1", null, user1);
26         chatHistory.addMessage(message1);
27         Iterator<Message> iterator = chatHistory.iterator(user1);
28         assertEquals(message1, iterator.next());
29     }
30 }
31
```