```java
import org.junit.jupiter.api.Test;

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

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

class UserTest {
    @Test
    void processUndoLastMessage() {
        User testUser = new User("testUser");

        Message message1 = new Message("sender1", "Hello", new User[]{testUser}, "123");
        Message message2 = new Message("sender1", "Hi", new User[]{testUser}, "456");

        testUser.updatePercievedChat(message1);
        testUser.updatePercievedChat(message2);

        int initialPerceivedChatSize = testUser.getPercievedChat().size();
        int initialChatHistorySize = testUser.getChatHistory().getWholeHistory().size();

        testUser.processUndoLastMessage(new User("sender1"));

        assertEquals(initialPerceivedChatSize - 1, testUser.getPercievedChat().size());

        assertEquals(initialChatHistorySize - 1, testUser.getChatHistory().getWholeHistory().size());
    }

    @Test
    void updatePercievedChat() {
        User testUser = new User("testUser");

        Message testMessage = new Message("sender1", "Hello", new User[]{testUser}, "123");

        testUser.updatePercievedChat(testMessage);

        LinkedList<Message> perceivedChat = testUser.getPercievedChat();
        assertNotNull(perceivedChat);
        assertEquals(1, perceivedChat.size());
        assertTrue(perceivedChat.contains(testMessage));

        LinkedList<MessageMomento> chatHistory = testUser.getChatHistory().getWholeHistory();
        assertNotNull(chatHistory);
        assertEquals(1, chatHistory.size());
        assertEquals(chatHistory.getFirst().getChatVersion(), testMessage);
    }

    @Test
    void iterator() {
        User testUser = new User("testUser");

        Message message1 = new Message("sender1", "Hello", new User[]{testUser}, "123");
        Message message2 = new Message("sender2", "Hi", new User[]{testUser}, "456");

        testUser.updatePercievedChat(message1);
        testUser.updatePercievedChat(message2);

        Iterator<MessageMomento> iterator = testUser.iterator();

        assertNotNull(iterator);

        assertTrue(iterator.hasNext());
        MessageMomento momento1 = iterator.next();
        assertEquals(message2, momento1.getChatVersion());

        assertTrue(iterator.hasNext());
        MessageMomento momento2 = iterator.next();
        assertEquals(message1, momento2.getChatVersion());

        assertFalse(iterator.hasNext());
    }
}
```

PDF document made with CodePrint using Prism