**Chat Relay – Test Plan Document**

*Software Testing Specification*

Table of Contents

[1. Classes to be Tested 5](#_Toc197433536)

[2. Operations to be Tested 6](#_Toc197433537)

[1. Packet Class: 6](#_Toc197433538)

[2. Chat Class: 6](#_Toc197433546)

[3. Message Class: 6](#_Toc197433553)

[4. User/AbstractUser Class: 7](#_Toc197433561)

[5. DBManager Class: 7](#_Toc197433567)

[6. Server Class: 7](#_Toc197433573)

[7. Client/ClientHandler Class: 8](#_Toc197433578)

[8. ITAdmin Class: 8](#_Toc197433584)

[3. Specific Test Cases 9](#_Toc197433587)

[1. Packet Tests: 9](#_Toc197433588)

[2. Chat Tests: 9](#_Toc197433593)

[3. Message Tests: 9](#_Toc197433598)

[4. User/AbstractUser Tests: 9](#_Toc197433604)

[5. DBManager Tests: 9](#_Toc197433610)

[6. Server Tests: 10](#_Toc197433619)

[7. Client/ClientHandler Tests: 10](#_Toc197433626)

[8. ITAdmin Tests: 10](#_Toc197433634)

[9. Integration Tests: 10](#_Toc197433639)

[4. Instructions to Run Test Suite 12](#_Toc197433644)

# Classes to be Tested

* Packet - has tests in PacketTesting.java
* Chat - has tests in ChatTest.java
* Message – has tests in MessageTesting.java
* User/Abstract User – has tests in UserTest.java
* DB Manager – has tests in DBManagerTesting.java
* Server – has tests in ServerTesting.java
* Client/Client Handler – has tests in ClientTesting.java
* IT Admin – has tests in ITAdminTest.java

# Operations to be Tested

1. Packet Class:
   * Constructor initialization
   * getActionType() method
   * getActionArguments() method
   * getSenderId() method
   * getTimeCreated() method
   * Serialization/deserialization
   * Various action types
2. Chat Class:
   * Constructors
   * Adding users to chat
   * Removing users from chat
   * Adding messages
   * Privacy settings
   * toString() representation
3. Message Class:
   * Constructor initialization
   * getCreatedAt() method
   * getId() method
   * getContent() method
   * getChat() method
   * getSender() method
   * toString() method
4. User/AbstractUser Class:
   * Constructor initialization
   * User information getters (username, password, ID)
   * Adding chats to user
   * isAdmin and isDisabled status
   * CreateChat, addUserToChat, sendMessage functionalities
5. DBManager Class:
   * User loading/saving
   * Chat loading/saving
   * Message loading/saving
   * ID-based retrieval methods
   * Data persistence
6. Server Class:
   * Connection handling
   * Packet processing
   * Client management
   * Error handling
7. Client/ClientHandler Class:
   * Connection to server
   * Packet sending/receiving
   * User interface integration
   * Login/logout process
   * Message sending
8. ITAdmin Class:
   * User management (create, enable, disable)
   * Chat log operations

# Specific Test Cases

1. Packet Tests:
   * Verify packet construction with all action types
   * Test packet serialization and deserialization
   * Test handling of null parameters
   * Test timestamp accuracy
2. Chat Tests:
   * Verify chat creation with appropriate ownership
   * Test adding and removing users from chat
   * Test privacy settings changes
   * Test message addition to chat
3. Message Tests:
   * Test message construction with valid parameters
   * Test message ID uniqueness
   * Test message timestamp accuracy
   * Test message content retrieval
   * Test message-to-string conversion
4. User/AbstractUser Tests:
   * Test user creation with valid parameters
   * Test user ID uniqueness
   * Test adding chats to user
   * Test user roles (regular user vs. admin)
   * Test disabled status effects
5. DBManager Tests:
   * Test loading users from text files
   * Test loading chats from text files
   * Test loading messages from text files
   * Test user retrieval by ID
   * Test chat retrieval by ID
   * Test message retrieval by ID
   * Test data persistence (save & reload)
   * Test file format validation
6. Server Tests:
   * Test server initialization with various ports
   * Test client connection acceptance
   * Test packet reception and handling
   * Test concurrent client handling
   * Test error handling for invalid packets
   * Test server shutdown
7. Client/ClientHandler Tests:
   * Test client connection to server
   * Test login packet sending
   * Test logout packet sending
   * Test message sending
   * Test chat creation
   * Test response handling from server
   * Test disconnection handling
8. ITAdmin Tests:
   * Test user creation by admin
   * Test user disabling by admin
   * Test user enabling by admin
   * Test chat log extraction
9. Integration Tests:
   * End-to-end test: Login → Create Chat → Send Message → Logout
   * Admin operations test: Create User → Disable User → Enable User
   * Chat persistence test: Create Chat → Close App → Reopen → Verify Chat Exists
   * Multiple client test: Two clients sending messages to each other

# Instructions to Run Test Suite

1. Setup Instructions:
   * Ensure you have JUnit 5 in your classpath
   * Ensure the project is properly built
   * Create a test database folder structure with sample data for testing
2. Running the Test Suite:
   * From the command line:
   * *cd* /Users/(YourUserName)/Desktop/ChatRelayProject/ChatRelayProject/src
   * *javac* -cp .:junit-platform-console-standalone-1.8.2.jar chatRelay/\*.java unitTesting/\*.java
   * *java* -cp .:junit-platform-console-standalone-1.8.2.jar org.junit.platform.console.ConsoleLauncher --scan-classpath
3. From IDE (Eclipse/IntelliJ):
   * Right-click on the test package
   * Select "Run All Tests"
4. Expected Results:
   * All tests should pass with no errors
   * Console output should indicate successful test runs
   * Any failures will be reported with the test name and reason
5. Test Suite Implementation
   * The following test classes need to be created:
     + MessageTest.java
     + UserTest.java
     + DBManagerTest.java
     + ServerTest.java
     + ClientTest.java
     + ITAdminTest.java
     + IntegrationTest.java
6. Testing Schedule
   * Unit Testing Phase:
     + Complete all individual class tests
     + Verify each class works in isolation
   * Integration Testing Phase:
     + Test component interactions
     + Verify message flow from client to server and back
   * System Testing Phase:
     + End-to-end tests with multiple clients
     + Performance testing under load
7. Reporting
   * Test results should be documented including:
     + Number of tests passed/failed
     + Code coverage metrics
     + Any issues found during testing
     + Performance benchmarks