Chat Server Test Document

Contents

[1. Functional Test Case 3](#_Toc443736006)

[1.1. Connect to Server 3](#_Toc443736007)

[1.2. Close connection of server 3](#_Toc443736008)

[1.3. Input nick name 3](#_Toc443736009)

[1.4. Input duplicated nick name 3](#_Toc443736010)

[1.5. Join Room 3](#_Toc443736011)

[1.6. Display Rooms 3](#_Toc443736012)

[1.7. Send Message 4](#_Toc443736013)

[1.8. Receive Message 4](#_Toc443736014)

[1.9. Receive User entered Room Message 4](#_Toc443736015)

[1.10. Leave Room 4](#_Toc443736016)

[1.11. Receive User left Room Message 4](#_Toc443736017)

[1.12. Private Message 5](#_Toc443736018)

[2. Performance Test Case 6](#_Toc443736019)

[2.1. One room maximum users support 6](#_Toc443736020)

[2.2. Maximum rooms support 6](#_Toc443736021)

[2.3. Message delay limitation 6](#_Toc443736022)

[3. Stability Test Case 6](#_Toc443736023)

[3.1. Server stability time 6](#_Toc443736024)

# Functional Test Case

## Connect to Server

1. Step :
   1. For Linux input telnet address port
2. Expectation: Can connect to server

## Close connection of server

1. Precondition: TC 1.1 passed
2. Step:
   1. input “CTRL + C” or “/quit”
3. Expectation : Server can close connection

## Input nick name

1. Precondition: TC1.1 passed
2. Step:
   1. input a nick name “tester”
3. Expectation : show “ Welcome tester”

## Input duplicated nick name

1. Precondition: TC1.3 passed
2. Step:
   1. Open new terminal.
   2. telnet ip port
   3. input nick name “tester”
3. Expectation: Sorry, name take.

## Join Room

1. Precondition: TC1.3 passed
2. Step:
   1. Input /join test\_room
3. Expectation: “entering root: test\_room” and user list

## Display Rooms

1. Precondition: TC 1.5 passed
2. Step:
   1. /rooms
3. Expectation: room list( at least exist test\_room)

## Send Message

1. Precondition: TC1.5 passed
2. Step:
   1. Input “ test message”
3. Expectation: no error.

## Receive Message

1. Step:
   1. Open first terminal by telnet address port
   2. Input nick name “test1”
   3. /join room1
   4. Open Second terminal by telnet address port
   5. Input nick name “test2”
   6. /join room1
   7. Input “ test2 send test message”
2. Expectation: in first terminal, you should see” [test2] :test2 send test message”

## Receive User entered Room Message

1. Step:
   1. Open first terminal by telnet address port
   2. Input nick name “test1”
   3. /join room1
   4. Open Second terminal by telnet address port
   5. Input nick name “test2”
   6. /join room1
2. Expectation: in first terminal, you should see” new user entered chat room room1”

## Leave Room

1. Step:
   1. Open first terminal by telnet address port
   2. Input nick name “test1”
   3. /leave
2. Expectation: in first terminal, you should see” User has left chat test1 (\*\* this is you)”

## Receive User left Room Message

1. Step:
   1. Open first terminal by telnet address port
   2. Input nick name “test1”
   3. /join room1
   4. Open Second terminal by telnet address port
   5. Input nick name “test2”
   6. /join room1
   7. /leave
2. Expectation: in first terminal, you should see “User has left chat test1”

## Private Message

1. Step:
   1. Open first terminal by telnet address port
   2. Input nick name “test1”
   3. /join room1
   4. Open Second terminal by telnet address port
   5. Input nick name “test2”
   6. /join room1
   7. Open third terminal by telnet address port
   8. Input nick name “test3”
   9. /join room1
   10. Input /test2 this is private message
2. Expectation: in test2 terminal should see “test2: this is private message” and in test1 terminal should not see “test2: this is private message”

# Performance Test Case

## One room maximum users support

## Maximum rooms support

## Message delay limitation

# Stability Test Case

## Server stability time