

# Chat Server Test Document

---

## Contents

1. Functional Test Case .....	3
1.1. Connect to Server .....	3
1.2. Close connection of server .....	3
1.3. Input nick name .....	3
1.4. Input duplicated nick name .....	3
1.5. Join Room .....	3
1.6. Display Rooms .....	3
1.7. Send Message .....	4
1.8. Receive Message .....	4
1.9. Receive User entered Room Message .....	4
1.10. Leave Room .....	4
1.11. Receive User left Room Message .....	4
1.12. Unknown command .....	5
1.13. Repeat join same room .....	5
2. Performance Test Case .....	6
2.1. One room maximum users support .....	6
2.2. Maximum rooms support .....	6
2.3. Message delay limitation .....	6
3. Stability Test Case .....	6
3.1. Server stability time .....	6

# 1. Functional Test Case

## 1.1. Connect to Server

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

## 1.2. Close connection of server

- 1) Precondition: TC 1.1 passed
- 2) Step:
  - a. input "CTRL + C" or "/quit"
- 3) Expectation : Server can close connection

## 1.3. Input nick name

- 1) Precondition: TC1.1 passed
- 2) Step:
  - a. input a nick name "tester"
- 3) Expectation : show "Welcome tester"

## 1.4. Input duplicated nick name

- 1) Precondition: TC1.3 passed
- 2) Step:
  - a. Open new terminal.
  - b. telnet ip port
  - c. input nick name "tester"
- 3) Expectation: Sorry, name take.

## 1.5. Join Room

- 1) Precondition: TC1.3 passed
- 2) Step:
  - a. Input /join test\_room
- 3) Expectation: "entering root: test\_room" and user list

## 1.6. Display Rooms

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

### **1.7. Send Message**

- 1) Precondition: TC1.5 passed
- 2) Step:
  - a. Input " test message"
- 3) Expectation: no error.

### **1.8. Receive Message**

- 1) Step:
  - a. Open first terminal by telnet address port
  - b. Input nick name "test1"
  - c. /join room1
  - d. Open Second terminal by telnet address port
  - e. Input nick name "test2"
  - f. /join room1
  - g. Input " test2 send test message"
- 2) Expectation: in first terminal, you should see" [test2] :test2 send test message"

### **1.9. Receive User entered Room Message**

- 1) Step:
  - a. Open first terminal by telnet address port
  - b. Input nick name "test1"
  - c. /join room1
  - d. Open Second terminal by telnet address port
  - e. Input nick name "test2"
  - f. /join room1
- 2) Expectation: in first terminal, you should see" new user entered chat room room1"

### **1.10. Leave Room**

- 3) Step:
  - a. Open first terminal by telnet address port
  - b. Input nick name "test1"
  - c. /leave
- 4) Expectation: in first terminal, you should see" User has left chat test1 (\*\* this is you)"

### **1.11. Receive User left Room Message**

- 1) Step:
  - a. Open first terminal by telnet address port

- b. Input nick name "test1"
  - c. /join room1
  - d. Open Second terminal by telnet address port
  - e. Input nick name "test2"
  - f. /join room1
  - g. /leave
- 2) Expectation: in first terminal, you should see "User has left chat test1"

### **1.12. Unknown command**

- 1) Precondition: TC1.3 passed
- 2) Step:
  - a. Input " /room"
- 3) Expectation: "Unknown command :[/join /leave /quit /rooms]"

### **1.13. Repeat join same room**

- 1) Precondition: TC1.5 passed
- 2) Step:
  - b. Input " /join test\_room"
- 3) Expectation: You have in this room

## **2. Performance Test Case**

### **2.1. One room maximum users support**

Target 200 users

### **2.2. Maximum rooms support**

Unlimited, test 200 rooms, works well.

### **2.3. Message delay limitation**

2 Cores 1.6G 4G memory 2Mb bandwidths.

When connect more than 100 concurrent users send message, sometimes TCP ACK lost..

## **3. Stability Test Case**

### **3.1. Server stability time**

Until now: 3\* 24 hour.