Linux Chat

The Simple chat program that allows you to communicate with your peers in a free and friendly environment.

Local Area Communication

Contents

[Technical Report 3](#_Toc446528748)

[Summary 3](#_Toc446528749)

[Introduction 4](#_Toc446528750)

[General Test Evaluations: 4](#_Toc446528751)

[Iteration 1: Client Connects to Server 5](#_Toc446528752)

[Test Environment: 5](#_Toc446528753)

[Test Purpose: 5](#_Toc446528754)

[Test Results: 5](#_Toc446528755)

[Test Conclusion: 5](#_Toc446528756)

[Iteration 2: Client Sends a Message 6](#_Toc446528757)

[Test Environment: 6](#_Toc446528758)

[Test Purpose: 6](#_Toc446528759)

[Test Results: 6](#_Toc446528760)

[Test Conclusion: 6](#_Toc446528761)

[Iteration 3: Client Receives a New Message 7](#_Toc446528762)

[Test Environment: 7](#_Toc446528763)

[Test Purpose: 7](#_Toc446528764)

[Test Results: 7](#_Toc446528765)

[Test Conclusion: 7](#_Toc446528766)

[Iteration 4: Client Has a Bad Username 8](#_Toc446528767)

[Test Environment: 8](#_Toc446528768)

[Test Purpose: 8](#_Toc446528769)

[Test Results: 8](#_Toc446528770)

[Test Conclusion: 8](#_Toc446528771)

[Iteration 5: Client Disconnects 9](#_Toc446528772)

[Test Environment: 9](#_Toc446528773)

[Test Purpose: 9](#_Toc446528774)

[Test Results: 9](#_Toc446528775)

[Test Conclusion: 9](#_Toc446528776)

[Iteration 6: Client Reconnects 10](#_Toc446528777)

[Test Environment: 10](#_Toc446528778)

[Test Purpose: 10](#_Toc446528779)

[Test Results: 10](#_Toc446528780)

[Test Conclusion: 10](#_Toc446528781)

[Iteration 7: Client Logs the Chat Session 11](#_Toc446528782)

[Test Environment: 11](#_Toc446528783)

[Test Purpose: 11](#_Toc446528784)

[Test Results: 11](#_Toc446528785)

[Test Conclusion: 11](#_Toc446528786)

[Iteration 8: Client Has Bad Credientials 12](#_Toc446528787)

[Test Environment: 12](#_Toc446528788)

[Test Purpose: 12](#_Toc446528789)

[Test Results: 12](#_Toc446528790)

[Test Conclusion: 12](#_Toc446528791)

[Iteration 9: Creation of a New Server Instance 13](#_Toc446528792)

[Test Environment: 13](#_Toc446528793)

[Test Purpose: 13](#_Toc446528794)

[Test Results: 13](#_Toc446528795)

[Test Conclusion: 13](#_Toc446528796)

[Iteration 10: Normal Server Operation 14](#_Toc446528797)

[Test Environment: 14](#_Toc446528798)

[Test Purpose: 14](#_Toc446528799)

[Test Results: 14](#_Toc446528800)

[Test Conclusion: 14](#_Toc446528801)

[Iteration 11: Server Stops Running 15](#_Toc446528802)

[Test Environment: 15](#_Toc446528803)

[Test Purpose: 15](#_Toc446528804)

[Test Results: 15](#_Toc446528805)

[Test Conclusion: 15](#_Toc446528806)

[Conclusion of the resulting tests: 16](#_Toc446528807)

# Technical Report

## Summary

## Introduction

### General Test Evaluations:

Blah blah blah, we are looking for these results and in general the environment looks like blah blah blah

## Iteration 1: Client Connects to Server

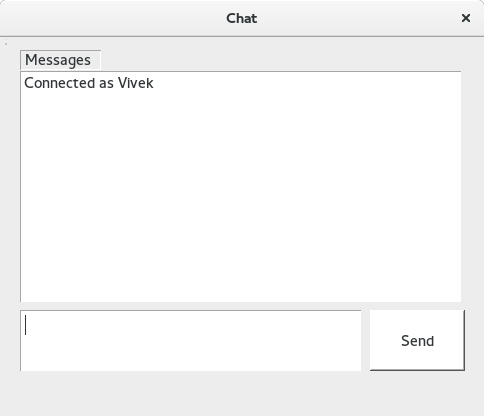
### Test Environment:

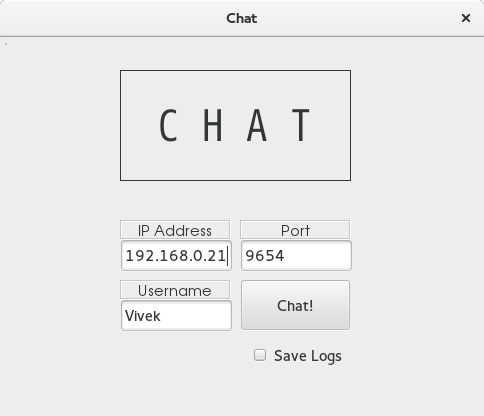
On a Fedora machine, we are running a local server while having another computer acting as a client. We will enter in our username, the server’s IP Address and our desired username.

### Test Purpose:

We will test the basic functionality of a Client connected to a running server.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

****

### 

### Test Conclusion:

Connecting to an active server is functional. The very first message seen will be the window announcing that you are connected as “your username.” **This is success.**

## Iteration 2: Client Sends a Message

### Test Environment:

On a Fedora machine, we are running a local server while having another computer acting as a client. After a successful connection, we are going to send a message to the server.

### Test Purpose:

We will test the basic functionality of a Client connected to a running server and sending a message to the server.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Iteration 3: Client Receives a New Message

### Test Environment:

On a Fedora machine, we are running a local server while having two computers acting as a client. After a successful connection, we will remain connected until the other client sends a message.

### Test Purpose:

We will test the basic functionality of a Client connected to a running server and how an active chat operates.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Iteration 4: Client Has a Bad Username

### Test Environment:

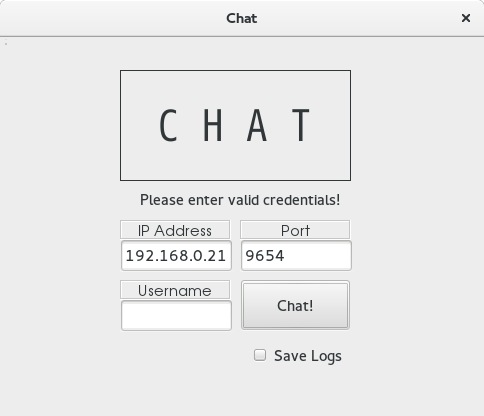
On a Fedora machine, we are running a local server while having two computers acting as a client. We will attempt to connect to a client with no username.

### Test Purpose:

We are testing to see if, essentially, a client successfully connects and specifies no username. A connection will occur initially but will be refused when there is no name.

### Test Results:

**Test 4– Fig**

****

### Test Conclusion:

The client refuses to connect to the server and displays an error message indicating that user has inputted invalid credentials. **This is an intended failure and will be declared as a success!**

## Iteration 5: Client Disconnects

### Test Environment:

On a Fedora machine, we are running a local server while having two computers acting as a client. After a successful connection, we will disconnect after a chat session.

### Test Purpose:

We are checking to see if a server properly handles a client’s disconnection and sends a message to all the clients when those connections end.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Iteration 6: Client Reconnects

### Test Environment:

On a Fedora machine, we are running a local server while having two computers acting as a client. After previously being disconnected, we will re-enter the chat as a new user with the same IP Address as before.

### Test Purpose:

We are checking to see if a server properly destroys a disconnected client’s information and then recreates the client with a new username and chat instance.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Iteration 7: Client Logs the Chat Session

### Test Environment:

On a Fedora machine, we are running a local server while having two computers acting as a client. Right before the client connects, the client specifies that they want to log this chat session. Afterwards a normal chat session will occur.

### Test Purpose:

We want to see the result of how the chat log appends every new chat message until a client session ends. Any non-ascii characters will be regarded as errors.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Iteration 8: Client Has Bad Credientials

### Test Environment:

On a Fedora machine, we are running a local server while having two computers acting as a client. The client will attempt to connect to a server with:

1. The wrong port number
2. An invalid IP Address

### Test Purpose:

A client that attempts to connect with the wrong credentials should immediately fail and return to the login page.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Iteration 9: Creation of a New Server Instance

### Test Environment:

On a Fedora machine, we will start the server with no clients.

### Test Purpose:

Checking to see if a server can initialize.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Iteration 10: Normal Server Operation

### Test Environment:

On a Fedora machine, we will see the normal operation of a server when multiple clients connect, disconnect, reconnect and interact with each other.

### Test Purpose:

We want to see normal client interaction and how the system handles it. Anytime the server crashes will result in a failure.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Iteration 11: Server Stops Running

### Test Environment:

On a Fedora machine, we are running a local server while having two computers acting as a client. After normal user interaction, the server will suddenly stop. This will destroy all current client connections.

### Test Purpose:

We want to see what happens on the client end when we can’t send and receive messages to a server. A client instance should terminate when there are too many failures.

### Test Results:

**Test 1 – Fig 1 Test1 – Fig 2**

### Test Conclusion:

## Conclusion of the resulting tests:

It work good, plz use. Also this doc is awesome.