

# Term Project: *ChatApp*

## Test Plan Document

### Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>2</b>
1.1	<i>Purpose and Scope.....</i>	2
1.2	<i>Target Audience.....</i>	2
1.3	<i>Terms and Definitions.....</i>	2
<b>2</b>	<b>Test Plan Description.....</b>	<b>3</b>
2.1	<i>Scope of Testing.....</i>	3
2.2	<i>Testing Schedule.....</i>	3
2.3	<i>Release Criteria.....</i>	4
<b>3</b>	<b>Unit Testing.....</b>	<b>5</b>
3.1	Server.....	5
3.1.1	Running Test.....	5
3.1.2	Client Connection Test.....	5
3.1.3	Account Test.....	5
3.1.4	General and Personal Messaging Test.....	6
3.2	<i>Client.....</i>	6
3.2.1	Client Connection Test.....	6
3.2.2	Send Recieve Tests.....	6
3.2.3	GUI Test.....	7
<b>4</b>	<b>Integration Testing.....</b>	<b>8</b>
4.1	Login.....	8
4.2	General Message.....	8
4.3	Personal Message.....	8

# 1 Introduction

This document describes the tests that need to be completed in order for the product to be released. It includes both the unit tests that will be carried out and will treat the unit as a black box, testing for correct behavior as well as the integrated tests which will test from inside of the unit to ensure that it is functioning properly.

## 1.1 Purpose and Scope

The purpose of this document is to outline the various tests needed to release the product and ensure that it is functioning properly. The scope of this document is limited to the Chat application only looking at both the server and client.

## 1.2 Target Audience

The target audience for this document is myself, the TA and teacher.

## 1.3 Terms and Definitions

The following terms are used throughout the document:

- GUI – graphical user interface
- Client – the user side application that makes up the front end of the chat application
- Server – the program that interacts between clients and makes up the back end of the application.

## 2 Test Plan Description

This section is designed to explain the plan for testing the application. It includes both the unit tests to be conducted as well as the integrated tests and schedule.

### 2.1 Scope of Testing

The tests are designed to ensure that first and foremost users can connect to the server and pass messages to each other. Secondly to ensure that users can create and log in to accounts and those accounts remain in the system after reboot. Thirdly to ensure that users can both send messages to all people online as well as send private messages. Forth will be testing that the GUI works and is intuitive. Lastly to ensure that the conversations are saved and re-loaded to users when they login so that they do not miss any of the conversation. For this project I am omitting all tests relating to volume as there is no volume requirement and I do not have the resources to test with enough clients in order to sufficiently test this.

### 2.2 Testing Schedule

- Testing server up and running
- Testing server connection to client
- Testing client
- Testing client connection back to server
- Testing sending messages from server to client
- Testing sending messages from client to server
- Testing multi-client connection
- Testing client to client communication
- Testing login
- Testing account creation
- Testing chat history save
- Testing chat history load

- Testing GUI works and keeps all of the previously tested pieces functioning.

### **2.3 Release Criteria**

In order to be released the product must be able to support at least two clients store their chat histories and account information. Ideally it would support 10+ clients.

## 3 Unit Testing

This section describes the unit tests that will be used to ensure that the program is behaving as expected. These tests are conducted from outside of the unit that is being tested and treat that unit as a black box.

### 3.1 server

This unit is the communications hub that passes messages to clients and receives them it is also responsible for authenticating logins and saving and loading the chat history. The tests that this unit need to go through are as follows: ensure it is running, test that it can connect to clients, test that it can load and save account information, test that it can load and save chat history, test that the correct user gets logged in when they transmit their account details, and test that it can handle the distribution of both general messages and personal messages. If each of these tests pass then the unit passes and meets the minimum requirements needed to release.

#### 3.1.1 Running test

This is the simplest of tests, all that is required is that the application be run and not throw an error.

#### 3.1.2 Client Connection test

For this test the server must be running and then a client needs to be run, the server is set up to send a username message to a client after it connects, if this is observed on the client then the first part of this test passes. The next step is to ensure that more than one client can connect for this another client must be connected and if again the message is observed displaying on the new client then the test passes and the server has successfully connected to more than one client.

#### 3.1.3 Account test

For this test a client will again be connected to the server and from the client an account will be made, then if that succeeded the client will be disconnected. Next a new client will be run and from that client first an account will try to be made with the same username, which should fail. Next the client will try to login with the username but a wrong password this should also fail, then the reverse of this will be tried and that should fail too. Lastly the correct username and password will be passed in and this should connect the client properly. This will ensure that the usernames are working and unique.

### **3.1.4 General and Personal message Test**

For this test three clients will be created and logged into the server. A general message will be sent from one of the clients this should appear on the other two. Then on one of the other clients a personal message will be sent to one of the others and this should not appear on the third client. This will ensure that the communication is being handled properly by the server.

## **3.2 Client**

The client is the unit that the user interacts with it has a graphical user interface and allows for the user to type text into the text field and send it to either all of the other users or a specific user. In order to ensure that this unit is working properly the following test need to be conducted on it: test that the client can connect to the server, test that the client can send and receive messages, and test that the graphical user interface(GUI) is working properly.

### **3.2.1 Client Connection Test**

To conduct this test the server must be running and then a client must be fired up. When a client makes a connection to the server the server makes a note of this on its console window so for this test the console window of the server must be observed so that the connection can be verified.

### **3.2.2 Send and receive messages**

For this test the server will again be started and then two clients will be started, from one of the clients a message will be sent from the other one and from the other one a response will be sent back. These messages should be observed on the clients to prove that the test worked correctly and this feature passes inspection.

### **3.2.3 GUI test**

For this test the client needs to be run and then the features of the GUI need to be tried one at a time. These features are: login, text field, response field, general message, and personal message. Each of these need to be used one after another to ensure that they are working properly.

# 4 Integration Testing

In this section the server and client are tested together to ensure that they are working cohesively. This will check the three basic use cases of login, general messaging and personal messaging.

## 4.1 Login

For this test the server needs to be running and a client needs to be started. The client should first create an account and then be closed. Next the client should be restarted and try to login first with an account that does not exist, next with a username that does but a password that does not and then with a password that does and username that does not and finally with the username and password that were originally created. This ensures that saved account information is coming back when a client starts up and that each username and password is unique and will only allow for login if put together correctly.

## 4.2 General message

For this test the server needs to be running and three clients all logged in. From each of the clients a message will be written and for each client that writes a message that message should be observed on the other two clients. If this observation happens it can be confirmed that the server is receiving the messages and retransmitting them to the correct clients.

## 4.3 Personal Message

For this test the server and three clients need to be running. From one of the clients a personal message needs to be sent to one of the other two and on that client the message should be observed, but on the third it should not. This will ensure that private conversations are being kept private by the server.