

Term Project: *Chat Application*

Test Plan Document

Table of Contents

1 Introduction..... 2

1.1 Purpose and Scope..... 2

1.2 Target Audience..... 2

1.3 Terms and Definitions..... 2

2 Test Plan Description..... 3

2.1 Scope of Testing..... 3

2.2 Testing Schedule 3

2.3 Release Criteria 4

3 Unit Testing 5

3.1 Login Client 5

3.2 Server*Error! Bookmark not defined.*

3.3 Chat Client.....*Error! Bookmark not defined.*

4 Integration Testing.....

4.1 User logs onto the Server.....*Error! Bookmark not defined.*

4.2 Server and Client Chat.....

4.3 Server and Multiple Clients Chatting*Error! Bookmark not defined.*

1 Introduction

This document is test plan for the Chat Application Program. This Chat Application consists of a chat server and multiple client users. The Server accepts connections from clients and delivers all the message in-between the users. This is a two-way communication tool. To access all these features a user must go through a login client to gain access. This document breaks down the chat application into components and tests each component separately on individual test cases.

1.1 Purpose and Scope

The purpose of this document is to break down the chat application to its core components and test them individually against test cases. This will help insure that every component can work independently and will have fewer errors. By designing our program using OOP, we can test each individual Object and its methods. If each object works well by itself, then we know most likely it will work and integrate with other components without running into issues.

1.2 Target Audience

This document is directed towards Developers and System Architect. The document distributes the chat application into individual components and test the functionality of each object. This testing will allow the developers and System Architect to find any bugs or cases where the components fail to do their task. The Unit and integration testing will help finalize the program and decide what features will stay or be removed.

1.3 Terms and Definitions

User- A client that uses the chat application.

Client- The chat handler for a user. The user interface and communication.

Server- A server that accepts connections from clients and directs all communication

Chat Log- A text file record of a user's chat history.

Account Manager- Class that stores all the registered valid accounts

2 Test Plan Description

The purpose of this section is to outline all the major components in the program and test them each individually. Each component will have a set of test cases that will demonstrate and guarantee that the expected output is true.

2.1 Scope of Testing

The scope of the testing will include all the major components in the chat application. The login client the user interacts with, the main Server that handles all the communication, the Chat Client that handles all the user interaction and handles messages from other users. Each major function of these objects will be tested individually. Testing of removing user accounts and private chat between users will not be included. These at the current time of this document have been pushed until the next iteration.

2.2 Testing Schedule

Login client

- Test of connecting/disconnecting to the server
- Entering of valid username/password
- Entering of invalid username/password
- Entering of blank login details
- Registering a new user

Server

- Starting a Server Socket to allow connections
- Accept 1 user connection
- Accept two connections
- Accept multiple connections at the same time
- Disconnect a single user
- Disconnect multiple connections at the same time
- Continue Listening for multiple clients

Client

- Connect to a Server
- Send single message to server
- Send multiple messages to server
- Receive message from server
- Receive multiple messages from server
- Add user to list of online users
- Remove user to list of online users
- Show chat records
- Disconnect and save messages
- Write Messages to Chat Log
- Read from Message from Chat Log

2.3 Release Criteria

Before this Chat Application can be deployed, all the main major components and features should be working. Login client should be able to login and register a user to the server. The Server should be able to accept at least 3 or more connections simultaneously and allow communication between them. The Client Chat will be able to send and display messages from other users in the correct order. Add and remove users to the list of online users. User should be able to access their chat records and see all the history.

3 Unit Testing

The purpose of this section is to break up the Chat Application into separate units and test all the functionality of them individually. This document will be covering the Login Client, Server and Chat Client. The Login is in charge of connecting users. Server takes care of accepting Client communications. Chat Client is responsible for sending and receiving messages to other users and providing a user interface to chat.

3.1 Login Client

The Login Client is the graphical user interface for allowing a User to Login into the chat server. It consists of two fields, a username and password. Its Job is to take the user information and verify with the server and see if that is a valid account. The testing will check for all possible combinations for user input and login attempts.

Test Case	Expected Output
User Enters Valid Username and Password	Login Successful
User Enters Invalid Username and Password	Login Denied
User Enters Valid Username , wrong password	Login Denied
User Enters Invalid Username , valid password	Login Successful
User Enters Blank Username and Password	Login Denied
User Registers new Username and Password	Login Successful
User Attempts to Register with Existing Username	Login Denied
Login Attempts to running Server	Login Successful

3.2 Server

The Server is the back end of the Chat Application. It is the central component that links everything else together. The Server should be able to listen accept outside client connections and create separate threads for each connection. It should be able to add and remove connections. Communicate with all the connected users and transmit data between them. The following cases will test these features and look for any possible errors.

Test Case	Expected Output
Start Server	Sever to Run
Start Server on valid port	Successful
Start Server on invalid port	Unsuccessful
Accept 1 connection	Successful
Accept 2 connections	Successful
Accept multiple connections at once	Successful
Disconnect 1 connection	Successful
Disconnect 2 connections	Successful
Disconnect multiple at once	Successful
Start 1 Chat Thread	Successful
Start 2 Chat Threads	Successful
Start multiple Chat Threads	Successful
Server Loses connection Client	Unsuccessful
Server changes port	Unsuccessful
All connections leave at once	Unsuccessful
Server Crashes	Unsuccessful

3.3 Chat Client

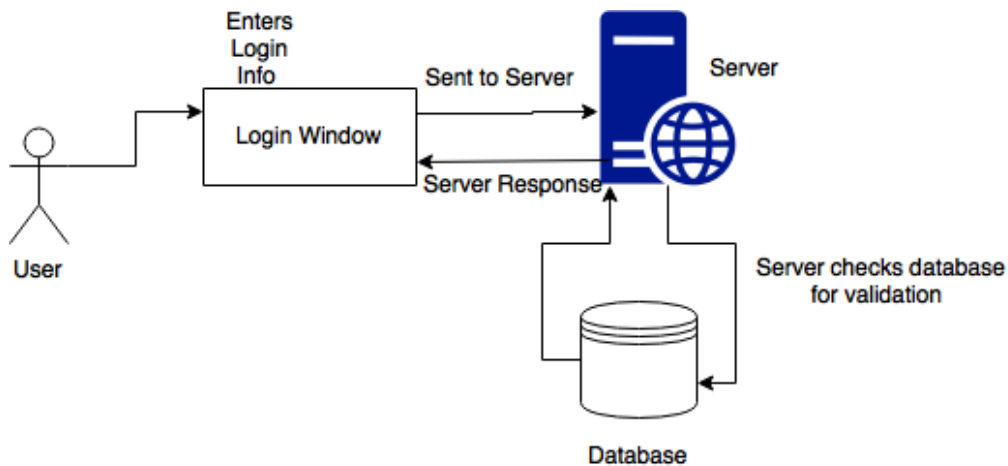
The Chat Client is the Graphical User interface that the User interacts and communicates with. It Also establishes a connection with the Server so two-way communication of reading and writing can occur with the Server. The Graphical User interface lets the User input text and display it in a chat area. The Chat Client must read all incoming messages from other users and display it into the chat area box. It keeps a list of online users and updates that list as new users come and leave. Lastly the chat client is responsible for recording all the chat messages and storing them locally. When A User Logs back into the chat client, the previous messages should be stored and recorded. The user can access and display those records. The following cases bellow will test all these features independently and make sure the system as a whole is working properly.

Test Case	Expected Output
Chat Client Connects to Running Server	Successful , Chat Window Pop Up
Chat Client connects to invalid Server	Unsuccessful, no Chat Window
Chat Client loses connections to Server	Unsuccessful ,Chat Window should close
1 additional User logs on	User added to the list
2 additional Users log on	Users added to the list
Multiple Users log on same time	Users are added to the list in order of arrival
1 User logs off	User is removed from the list
2 Users log off	Users are removed from the list
User enters a message	Message sent to chat area
User enters blank message	Error , message not sent to chat area
User enters message longer than input field	Message will wrap, sent to chat area
1 Message received from other User	Send message to chat area
Multiple Messages received from Users	Send messages to chat area in order received
User clicks records button	Window with chat history appears
User clicks record button (No history)	Window appears with blank area
User Logs off	Chat Records saved to file , window closes

4 Integration Testing

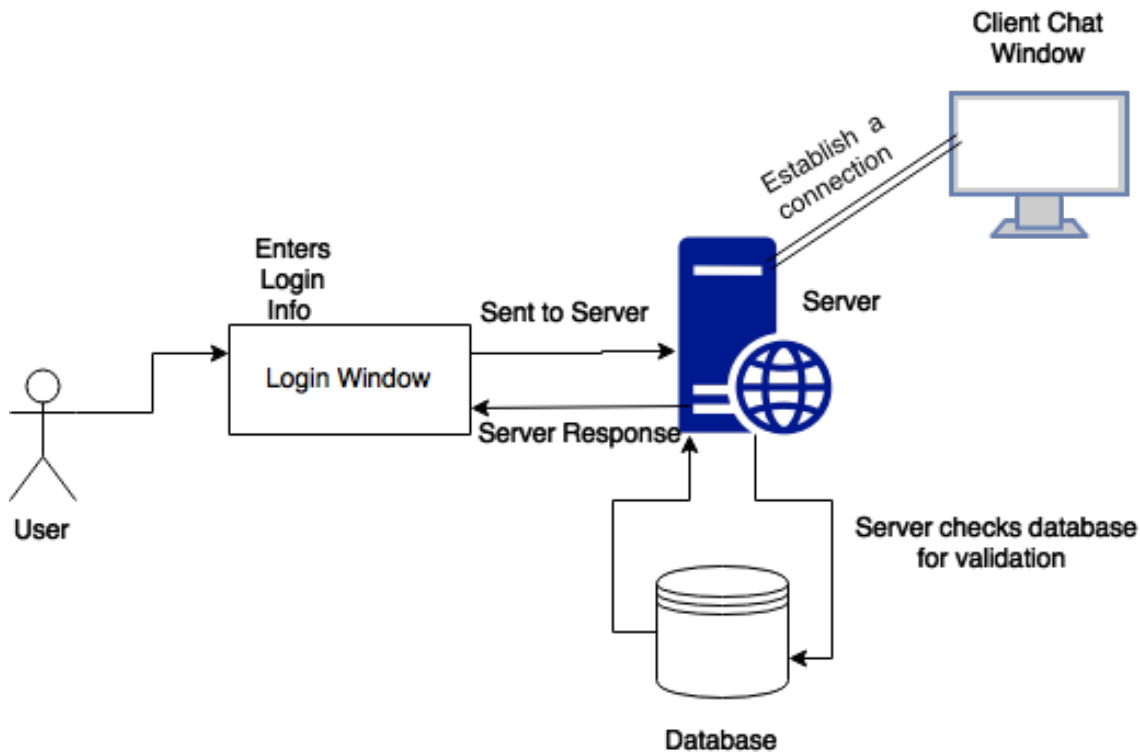
The purpose of integration testing is to take the components we tested in Unit Testing and combined them together. Here we are testing as a group and seeing how they should work together and see the flow of information and data. This part of the testing will simulate actions that a User might perform that require multiple components to be used. Unit and Integration testing will help demonstrate what is successful and where faults may occur in the Chat Application.

4.1 User Logins onto the Server



This Test case simulates when a User wants to login into the chat for the first time. The components being used here are the Login, Server and Account Manager. These 3 together should result in a user successfully logging in. First A User will open the Login window. The Login component should take the username and password and sent it to the Server. From here the Server should take that information and send it to the Account Manager for validation. If the Account manger finds a match and the password is correct, it should send a success message to the Server which then would said another message to the Login Client saying the information was valid. The Login Window Should Close and the Server should take that information and retrieve the correct account. That account will now be used for all chatting purposes. If the details did not match an account, the Server should send a message to the Login client saying the information was not valid.

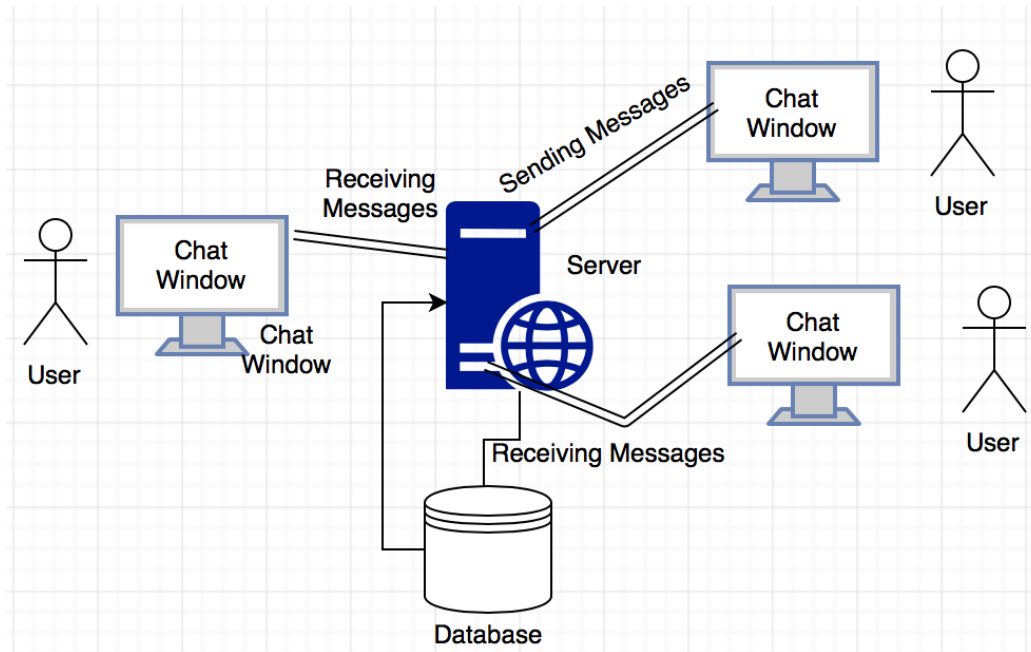
4.2 Server and Client Client



This test case simulates After a User logs in, the Server handles the connection and Opens a new Client Chat Window. This Client Chat Window will be the GUI interface for sending and receiving messages. The Server and the Chat Client have a two-way communication that should stay connected until the user quits. The following results are expected down below.

The Server will start and will listen on the port for incoming connections request. If no connection request is made, keep looping until one happens. The Server should not stop running. Once a connection is accepted, the Server will add that connection to an array of connections that will hold users connected. The Server will than start a new Separate Thread and run a new client chat window with the User Account acquired from the login. The Server and Client Chat will both establish writing and reading communication. The Server will send a welcome message to the Client Chat. If connection to Client Chat fails, an error message should display and Client Chat should shutdown and be removed from the array of connections. If successful, the Client Chat window appear and will now have full access to read and write to the Server. This connection between Client and Server shall maintain until the the User Closes the Client Chat window.

4.3 Server and Multiple Clients chatting



This test case simulates users chatting online together. The components being used here are the Server, Database, and Chat Client. These Three together allow messages to be exchanged back and forth. After a user logs in the graphical user interface will pop up. This interface has a chat area for all the messages, two buttons and a side bar with a list of online users.

When the user types a message and hits the send button, the message should be then sent to the Server, the Server will then parse the message and determine its intended recipient and deliver the message. At the same time, before sending to the message to the Server, the message will be record to the Chat Records of that specific user. If a blank message is sent, the Server should not receive and it should not be recorded on the Chat Record.

The Server is constantly running and is waiting to receive messages from its users.

When another user logs onto the Server, the Server should send all the Client Chat Users a message indicating a new user. The Client Chat will then update its list of online users and the Server will add this user to its array of connections. If a User leaves the Client Chat window its messages will be saved. The Client will send a message to the Server and the Server will notify all the other users that someone has logged off. The Server will then remove that user from its list and keep on listening for new messages and requests. The Server should keep on running until an Admin shuts down the Server.

