

Term Project: *ChocAn*

Test Plan Document

Table of Contents

1	Introduction	2
1.1	<i>Purpose and Scope</i>	2
1.2	<i>Target Audience</i>	2
1.3	<i>Terms and Definitions</i>	2
2	Test Plan Description	4
2.1	<i>Scope of Testing</i>	4
2.2	<i>Testing Schedule</i>	4
2.3	<i>Release Criteria</i>	4
3	Unit Testing	5
3.1	<i>User Login</i>	5
3.2	<i>View All Online Users & Chat History</i>	5
3.3	<i>Send Message</i>	6
3.3	<i>Receive Message</i>	6
4	Integration Testing	7
4.1	<i>User Login</i>	7
4.2	<i>Send Message</i>	7
4.3	<i>Receive Message</i>	7
4.3	<i>User Logoff</i>	8

Introduction

The Daniel's Chat Application Test Plan outlines the techniques that will be used to assure all coding for the chat application is in expected working condition. The chat application is a program designed to allow two or more users to communicate with each other in real time with the user of a server to relay messages between them.

Functionalities include logging in and out of the program, viewing online users available to chat, and sending, receiving and archiving messages.

Purpose and Scope

The Test Plan allows the chat application programmer to plan out all of the details of the testing procedure to ensure the code and entire program is working as expected. The Test Plan allows the programmer to create a sort of checklist to compare expected versus actual functionality. If all bases are covered, the programmer, and grader, can be assured that the project is on the right path and fewer errors can be expected as the project progresses.

Target Audience

The Test Plan is intended to be reviewed by both the programmer himself, to review his understanding and knowledge of the assignment, as well as the professor and teaching assistant, as they will be able to see the programmer's understanding of the assignment and monitor the project's progress.

Terms and Definitions

Chat Application - refers to Daniel's Chat Application, or the main program assigned for this project. It will be hosted on a local computer and communicates with the program server, which contains all the data for the program. Also referred to as "program."

Home Screen - Once logged into the chat application, the default screen is the home screen which includes a list of online users, menu options, as well as a text field--used to

send a message out to the entire list of online users.

Program - refers to Daniel's Chat Application, or the main program assigned for this project. It will be hosted on a local computer and communicates with the program server, which contains all the data for the program. Also referred to as "chat application."

Server - a non-local computer that stores the underlying data and data structure for the chat application. A user must successfully login to their account in order to communicate with the server

User - a person at a local computer who is communicating with the chat application, which communicates with a server.

Test Plan Description

The following Test Plan outlines the high-level details of what will be tested for this program and how.

Scope of Testing

The programmer intends to test all functionality of the program including the ability for multiple users to login successfully, create a user account, chat with one other person, send a message to all users, and log out. Additionally, it is necessary to test that all GUI components are working as desired as well as server connection and functionality.

Testing Schedule

Unit testing is being conducted as it is being written. Integration testing will take place over the next week as all units are completed or near completion to ensure all units are working together as desired.

Release Criteria

Multiple users should be able to connect to chat application successfully and send messages successfully. Aesthetic faults will be tolerated, particularly with the GUI, however, all precautions will be taken to ensure a smooth delivery.

Unit Testing

This section outlines the smaller units that will be tested through the coding process. Unit testing generally deals with segments of code--ensuring that the coding is accurate and performing tasks as desired.

User Login

On the local computer side, the program should store user login information in a class object and pass it to the server. A test method will be written to test that an object is being created, as desired. Username will have a maximum character length of 15 characters and it is not case-sensitive. A test will be conducted using a username longer than 15 characters and another using a mix of capital and lowercase letters. Additionally the password must be at least 5 characters and a maximum of 20. Tests will be conducted using a password smaller than 5 characters and another using more than 20 characters. Only alphabet characters will be allowed for username and a mix of alphabetical and numeric characters for the password. If username or password is outside of specified boundaries, an error message will be printed stating that the username or password is outside of the boundaries and to please try again.

On the server side, the server should pass in the class object with the login information, find a matching user in the list of existing users and allow a connection or print an error message indication that this user information cannot be found.

View All Online Users & Chat History

Once connected to the chat application--after successfully logging in--a window should open with a list of online users and buttons to send a message or access chat history. Double clicking on a user's name should open a new chat window where the chat can take place. It is unacceptable if this window does not open as the user will not have a way to chat with anyone. Clicking on the "Chat History" button will populate the screen with

all messages the two users have ever sent--it should be displayed in the chat window with a scrolling feature.

Send Message

Once a chat window is open, a user should be able to type in a message of any length using any standard keyboard symbols. The message will be sent as written--there will be no checks for case or length.

Receive Message

When a user received a message, a new chat window should open, if one is not already open for conversation between the two users. If one is already open, the message should appear in the chat window. The message should be formatted exactly how the sender typed it--it will be not case corrected in any way.

Integration Testing

This section outlines the broader segments of the chat application--ensuring that the different pieces of the program work together to create a seamless functionality for the end user.

User Login

Once the user enters in their login information, it should be sent to the server accurately in a class object. The server will then traverse the list of user login information in search of a match. If a match is found, connection is allowed and the user can begin chatting. If a match is not found, the program should inform the user that a match could not be found and to please try again or register an account. For account registration, the server will receive the same class object of login information, only this time it will add the information to the list of users. It will then present the user the login screen again, so that they may enter the information in a login.

Send Message

Once the user has typed their message and hit send or "Enter," the program should send the String message to the server which will pass it in, find the user it is being sent to and send it on to them--opening a new window on their side or adding it to the chat log of an existing open window. Tests will be done to make sure these messages are being sent accurately and to the correct users--be it one individual or all users.

Receive Message

After receiving the message from the sender, the server will find the correct user to send the message to and send accordingly. A window should open on the local computer if one is not already open or the message will be added to the chat log if a window is already open for communication between the two users.

User Logoff

After the user has finished chatting or would like to close the program, they should hit the “logout” button which should alert the server to close connection. The logoff button will be tested to make sure it works properly. A test message will be sent to an offline user--a user that has just signed off--to make sure no new window opens after they have disconnected. The logged off user will sign back on to ensure they receive the message after they have logged back in.