Communications Project Test Plan

Test Plan for CS 401, Software Engineering

Prepared by Jose Galeana, Herminia Alvarenga, Lyca Mangarin, Jasan Manimbao, Chris Sanchez

OVERVIEW & PURPOSE

The tests for the Communications Project were made to test the progress of the project, by checking if the message classes were being set up and used correctly. It was going to be difficult getting 100% coverage, so we decided that for this project we aim to get at least 20% coverage.

Testing Goals

1. 20% minimum coverage.
2. Test constructor, getters and setters of message classes.
3. Get majority, or all, tests to pass.

Test Risks/Mitigation

Some risks to consider when making tests, and how we considered them.

Risks	Mitigation
Incomplete coverage	*Check code coverage to check what parts of the code are covered vs uncovered*
Tests passing/failing depending on the run and outcome	*For example, with the setTimestamp it was off by milliseconds so we had to consider*

		within a second as passing. So check the constructor for the tests to try and prevent such cases.
Tests incorrectly failing due to typo		Check what is being compared, what is expected vs actual, and check that the correct method is called for each comparison.

Tests

Below are the tests that were made and the results.

Test Class	Test #	Test Name	Status	Description
AddUsersToChatMessageTest	1	testMessageCreation	Passed	Chat ID, participant IDs, and username should match the one set in constructor
	2	testSetAndGetAddedChat	Passed	Added chat room should be the same as the one set
	3	testMessageParticipants	Passed	New participant IDs should not be empty
	4	testMessageParticipantsSize	Passed	There should be two new participants
	5	testNullParticipants	Passed	Participant IDs should be null if set to null in constructor
ChatMessageTest	6	getMessageTest	Passed	Should return test message
	7	getSenderTest	Passed	Should return the sender from the setUp
ChatRoomTest	8	testGetChatID	Passed	Chat ID should be automatically generated
	9	testSetandGetParticipants	Passed	Participants should include user3 and user4

	10	testAddMessage	Passed	There should be one message in the chat room. The message should be the one added
	11	testAddMember	Passed	Should return true when a new member that wasn't in the chat is added Should return false when a member who is already in the chat is added
	12	testMessageExists	Passed	Should be true when the message ID exists in the chat room. Should return false for a message ID that doesn't exist.
	13	testIsParticipant	Passed	user1 should be a participant. user5 should not be a participant.
	14	testToString	Passed	Should correctly list participants
CreateChatMessageTest	15	getParticipantIdsShouldReturnCorrectParticipants	Passed	Should return correct participants
LoginMessageTest	16	testMessageCreation	Passed	Username and password should correctly equal to the username and password established
	17	testSuccess	Passed	Login should be successful and display status as success
	18	testSetGetCurrentUser	Passed	The current user should be set correctly
	19	testSetGetAllUsers	Passed	All users map should be retrieved correctly. Correct user should be retrieved from map.
LogMessageTest	20	getUserIdShouldReturnCorrectId	Passed	Should return correct Id

LogoutMessageTest	21	testMessageCreation	Passed	LogoutMessage should be created. Username should match. Message type should be LOGOUT Initial status should be PENDING.
	22	testStatusUpdate	Passed	Status should be updated to SUCCESS
MessageTest	23	testGetMessageShouldMatchMessage	Passed	GetMessage should match the message.
	24	testSetMessageShouldEqualNewMessage	Passed	SetMessage should equal newly set message
	25	testGetSender	Passed	Should retrieve the sender.
	26	testSetSender	Passed	Should set the sender name to NewSender.
	27	testGetChatID	Passed	Should retrieve the Chat ID
	28	testSetChatID	Passed	Should set the Chat ID to NewChatID
	29	testGetTimestampShouldReturnTime	Passed	Should return current time.
	30	testSetTimestamp	Passed	Should match the timestamp within a second.
	31	testToStringForFile	Passed	Should match the format in the Message class.
	32	testToString	Passed	Should match the format as a String.
NotifyMessageTest	33	getNotificationTextShouldReturnCorrectText	Passed	getNotification text should return correct text.
PinChatMessageTest	34	getChatIdShouldReturnCorrectChatId	Passed	Should return correct chat ID
ServerMessageTest	35	getTypeShouldReturnUndefined	Passed	Should return undefined.

	36	getStatusShouldReturnUndefined	Passed	Should return undefined.
UpdateUserMessageTest	37	testUserId	Passed	The userId should be "testUser"
	38	testPassword	Passed	The password should be testPassword.
	39	testFirstname	Passed	The firstname should be Jose.
	40	testLastname	Passed	The lastname should be Galeana.
UserTest	41	testSetFirstName	Passed	The first name should be First.
	42	testSetLastName	Passed	The last name should be Last.
	43	testSetUsername	Passed	The username should be username1
	44	testSetPassword	Passed	The password should match the new password
	45	testGetFirstName	Passed	The first name should be Jose.
	46	testGetLastName	Passed	The first name should be Last
	47	testGetUsername	Passed	The username should be username.
	48	testGetPassword	Passed	The password should be password.
	49	testStatus	Passed	The status should be Online.
	50	testIsIT	Passed	isIT should be true.