Communication Test Plan

White Box Test

Test User

- Creates a User object with a username of “user” and a password of “user123” before all of the other methods.
- Test userID uses assertEquals(1, user.getUserID()) to verify that the user ID is set as expected.

Getter and Setter Test

- This test examines the functionality of the getter and setter methods for the username and user password. The assertions assertEquals("user", user.getUsername()) and assertEquals(“user123”, user.getUserPassword()) confirm that the getter methods return the expected values.
- Try setting password and username with different user and password, using assertNotEqual(“user”, user.getUsername()), and assertNotEqual(“user123”, user.getUserPassword()) . Those setter methods return the expected values.
- This test checks if the user is banned. Start by setting the ban status to true. Use assertTrue(user.isBanned()) to confirm the user is marked as banned.

Test Admin

- Creates an Admin object with the username "BobAdmin" and password "Bob123". Sets up necessary dependencies, including MessageHandler, AuthenticationSystem, and StorageManager.
- Test Add User
  Verifies the addUser() method by adding a new user. Use assertTrue(admin.addUser(newUser)) to confirm the user is added successfully.
- Test Delete User
  Tests the deleteUser() method. Add a user first, then use assertTrue(admin.deleteUser(user.getUserID())) to verify successful deletion.
- Test Reset Password
  Checks the resetUserPassword() method by resetting a user's password. Use assertTrue(admin.resetUserPassword(user.getUserID(), "newPass")) to confirm the reset.
- Test Ban User
  Tests the banUser() method. Add a user and then ban them using assertTrue(admin.banUser(user.getUserID())).

- Test Unban User
  Tests the unbanUser() method by unbanning a previously banned user. Use assertTrue(admin.unbanUser(user.getUserID())) to verify.

Test Message

- Creates a Message object with a sender ID of 1 and content "This is a New Message" before all other methods.
- Testing Message constructor, this test verifies the functionality of the Message constructor. Use assertNotNull(message) to confirm that the Message object is successfully created. To ensure the constructor initializes fields correctly, use assertEquals(1, message.getSenderID()) to verify the sender ID, and assertEquals("This is a New Message", message.getContent()) to check the content.
- Test Message constructor with Message Object. Creates a new Message object using an existing Message instance. Use assertNotNull(newMessage) to ensure the new object is created successfully. Verify the copied fields using assertEquals(1, newMessage.getSenderID()) for the sender ID and assertEquals("This is a New Message", newMessage.getContent()) for the content..
- Test get senderID, this test ensures the getter method for the sender ID works correctly. Use assertEquals(1, message.getSenderID()) to confirm the method returns the expected value.
- Test get content,This test verifies the getter method for message content. Use assertEquals("This is a New Message", message.getContent()) to confirm that the content is retrieved as expected.

Test ChatBox

- Creates a ChatBox object named "Test ChatBox" before testing its methods. Initializes two User objects (user1 and user2) and a HashSet of participants to aid testing.
- Test Constructor
  This test verifies the constructor of the ChatBox class. Use assertNotNull(chatBox) to confirm that the ChatBox object is successfully created.
- Test ChatBox Name and ID
  This test ensures the getName() and getChatBoxID() methods work as expected. Use assertEquals("Test ChatBox", chatBox.getName()) to verify the name, and assertEquals(3, chatBox.getChatBoxID()) to confirm the chat box ID is correctly set.
- Test Get Participants
  Adds user1 and user2 to the chat box and sets the participants. Use assertNotNull(chatBox.getParticipants()) to confirm the participants are retrieved successfully.

- Test Remove Participant

  Adds user1 to the chat box and removes them. Use assertTrue(chatBox.removeParticipant(user1)) to verify that the participant is successfully removed.
- Test Get Empty ChatBox

  This test checks the getEmpty() method. Use assertNotNull(emptyChatBox) to confirm the empty chat box object is created. Verify its properties using assertEquals(chatBox.getChatBoxID(), emptyChatBox.getChatBoxID()) to confirm the ID matches, and assertEquals(chatBox.getParticipants(), emptyChatBox.getParticipants()) to confirm the participants are correctly set.

Test Message Handler

- Initializes a MessageHandler object with dependencies such as StorageManager, ChatBox, ConcurrentHashMap for chatBoxes and userDB, and a Server object.
- Test Create ChatBox

  Tests the createChatBox() method by creating a chat box with a list of participants. Use assertNotNull(msgHandler.createChatBox(participants, "chat1")) to verify successful creation.
- Test Get ChatBox

  Verifies the getChatBox() method by retrieving a chat box using its ID. Use assertNotNull(msgHandler.getChatBox(boxId)) to confirm the chat box is retrieved correctly.
- Test Send Message

  Check the sendMessage() method by sending a message to a chat box. Use assertTrue(msgHandler.sendMessage(boxId, msg)) to ensure the message is sent successfully.
- Test Send Message to User

  Tests the sendMessageToUser() method by sending a direct message to a specific user. Use assertTrue(msgHandler.sendMessageToUser(userId, "Hi There")) to confirm the message is sent.
- Test Remove Participant from ChatBox

  Verifies the removeParticipantFromChatBox() method by removing a participant from a chat box. Use assertTrue(msgHandler.removeParticipantFromChatBox(boxId, userId)) to confirm successful removal.

Test Storage Manager

- Test Store and Retrieve ChatBox
- Creates a ChatBox object with the name "ChatBox1".
- Tests the storeChatBox() and retrieveChatBox() methods.

- Store the chat box using assertTrue(storageManager.storeChatBox(chatBox)) to confirm it is stored successfully.
- Retrieve the chat box using ChatBox retrievedChatBox = storageManager.retrieveChatBox(chatBox.getChatBoxID()).
- Use assertNotNull(retrievedChatBox) to confirm it is retrieved.
- Verify the retrieved chat box matches the stored chat box using assertEquals(chatBox.getChatBoxID(), retrievedChatBox.getChatBoxID()).

Test Authentication System

- Test Register User
- Creates a User object with a username of "testUsername" and a password of "testPassword" before all other methods.
- Tests the registerUser() method by registering the user.
- Use assertTrue(authSystem.registerUser(user)) to confirm the user is registered successfully.
- Verify that the user is retrievable using assertNotNull(authSystem.findUser(user.getUserID())).
- Test Validate Credentials
- Creates a User object with a username of "testUsername10" and a password of "testPassword10".
- Registers the user and tests the validateCredentials() method.
- Use assertEquals(user, authSystem.validateCredentials("testUsername10", "testPassword10")) to verify the method validates correct credentials.
- Use assertNull(authSystem.validateCredentials("testUsername10", "wrongPassword")) to confirm invalid credentials are not accepted.
- Test Ban and Unban User
- Creates a User object with a username of "testUsername20" and a password of "testPassword20".
- Registers the user and tests the banUser() and unbanUser() methods.
- Use assertTrue(authSystem.banUser(user.getUserID())) to confirm the user is banned.
- Verify the user's banned status using assertTrue(user.isBanned()).
- Then unban the user using assertTrue(authSystem.unbanUser(user.getUserID())) and confirm the user is no longer banned with assertFalse(user.isBanned()).
- Test Reset Password
- Creates a User object with a username of "testUsername30" and a password of "testPassword30".
- Registers the user and tests the resetPassword() method.
- Use assertTrue(authSystem.resetPassword(user.getUserID(), "newPassword")) to confirm the password is reset.

- Verify the updated password using assertEquals("newPassword", authSystem.findUser(user.getUserID()).getPassword()).