

Talha Ahmad Mela

28532

Report for second demo CS408

UI changes:

1. **Server:** Server GUI, disconnect button is wider because text was not clear. "IF100" and "SPS101" were changed to "IF 100" and "SPS 101".
2. **Client:** Client GUI, login button was made longer and unsubscribe button for IF100 was made wider. "IF100" and "SPS101" were changed to "IF 100" and "SPS 101".

Client code:

1. **Socket Initialization:** In the original code, the **clientSocket** was not initialized within **button_connect_Click** before calling **clientSocket.Connect**. In the new code, this has been fixed.
2. **Button enabling:** Correct button enabling lines were added to ensure only correct functions are available for the current state.
3. **Thread for Receiving Messages:** In the original code, messages were received in the **ReceiveFromServer** function, which was called directly within certain button click events. In the new code, this method is now executed in a separate thread started in **button_connect_Click**. This allows for continuous asynchronous message reception.
4. **Improved Message Handling:** The **ReceiveFromServer** method has been updated to handle various types of messages more effectively. It now includes cases for **SUBSCRIBED_SUCCESS**, **UNSUBSCRIBE_SUCCESS**, **login_success**, **login_failure**, and messages for the channels **IF100** and **SPS101**. Each of these cases has corresponding methods to handle the received data (**HandleSubscribeSuccess**, **HandleUnsubscribeSuccess**, **HandleLoginSuccess**, **HandleLoginFailure**).
5. **Graceful Thread Termination:** The new code includes checks to gracefully terminate the **ReceiveFromServer** thread. It checks for socket disconnection and handles **SocketException** to ensure that the thread exits properly when the client disconnects or the application closes.
6. **UI Updates for Various Actions:** The methods handling successful actions (**HandleSubscribeSuccess**, **HandleUnsubscribeSuccess**, **HandleLoginSuccess**,

HandleLoginFailure) also update the UI accordingly, enabling or disabling certain buttons based on the context.

7. **Form Closing Handling:** The **Form1_FormClosing** method has been updated to ensure proper closing of the socket and the application. This avoids abrupt crashes and ensures that resources are released properly.
8. **Removal of Unused Code:** Unused methods like **HandleReceivedMessage** and others that were part of the original code have been removed in the new version, as their functionality has been integrated into the updated **ReceiveFromServer** method.

Server code:

1. **Socket Initialization and Handling:** In the original code, the **serverSocket** was initialized but not adequately managed for accepting clients and handling disconnections. The new code includes robust handling for accepting client connections (**Accept** method) and managing client disconnections (**HandleClientDisconnection** method).
2. **Thread for Each Client:** In the new code, each client is handled in a separate thread (**HandleClient** method), which allows for concurrent management of multiple clients. This is a significant change from the original code, which lacked this level of concurrency.
3. **Dictionary for Client Management:** The new code uses a **Dictionary<string, Socket>** (**clientSocketDictionary**) to map usernames to their corresponding sockets, enabling more efficient client management, especially for broadcasting messages and handling user-specific actions.
4. **Improved Message Broadcasting:** The **BroadcastMessage** method in the new code uses the **clientSocketDictionary** to efficiently send messages to specific subscribed users, as opposed to the original code's approach of iterating through all sockets and checking usernames separately.
5. **Enhanced Subscription and Unsubscription Handling:** The new code includes more comprehensive methods (**HandleSubscribe** and **HandleUnsubscribe**) for managing user subscriptions to different channels, with appropriate responses sent back to clients.
6. **UI Updates and Logging:** The new code extensively updates the server's UI to reflect various actions, such as client connections, disconnections, subscriptions, and message broadcasts. This was less detailed in the original code.
7. **Graceful Shutdown Handling:** The **Form1_FormClosing** method in the new code ensures a graceful shutdown of the server, including closing all client sockets and the server socket. The original code lacked this comprehensive shutdown process.

8. **Login Handling Improvements:** The **HandleLogin** method in the new code properly checks if a user is already logged in using the **clientSocketDictionary**, and handles new logins more efficiently, including updating the UI and internal lists.
9. **Removal of Unused Code:** Some redundant or unused methods and code segments in the original code have been removed.