## Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Use Case Description** | **Input Data** | **Expected Output** | **Test Steps** |
| 01 | Start the chat server and listen for client connections. | Run the server program. | Server starts successfully and begins listening on a port. | 1. Run server application  2. Check for confirmation message indicating server started successfully  3. Verify that the server is listening on the specified port. |
| 02 | Connect a client to the server. | Run the client program and enter the server’s IP address. | Client connects to the server and receives a welcome message. | 1. Run the server application.  2. Run the client application and input the server IP  3. Verify that the client successfully connects to the server. |
| 03 | Send a message from one client and verify it is received by other clients. | Message text entered in the client’s input field. | Message appears on the sending and receiving clients’ screens. | 1. Run the server application.  2. Run two client applications and connect them to the server.  3. In one client, enter a message and send it.  4. Verify the message appears on both the sending and receiving clients. |
| 04 | Disconnect a client from the server. | Close the client application. | Server logs the client’s disconnection, and other clients are notified. | 1. Run the server application.  2. Run two client applications and connect them to the server.  3. Close one of the clients.  4. Verify the server logs the disconnection and that the other client is notified. |
| 05 | Multiple clients connect to the server simultaneously. | Run multiple clients at the same time. | Server handles concurrent client connections without error. | 1. Run the server application.  2. Simultaneously run multiple client applications and connect them to the server.  3. Verify that all clients connect successfully and can send/receive messages. |

## Testing Notes:

* **GUI Testing**: Ensure all GUI elements (buttons, text fields, labels) are responsive and display correct information.
* **Integration Testing**: Test how the GUI interacts with the server, especially during message sending and receiving.