Client đăng ký đến Server và gửi dữ liệu định kỳ cho Server:

1. Người dùng chạy file server.py
2. Server mở register socket (TCP) và chờ đợi client connect để đăng ký
3. Người dùng chạy file client.py
4. Người dùng nhập IP server và bấm apply
5. Người dùng nhập name, ip và UDP port của client và bấm submit
6. Client connect đến server và gửi REGISTER message chứa các thông tin (………) cho Server
7. Server kiểm tra REGISTER message xem có đúng giao thức đã quy định không
   1. Nếu đúng giao thức, Server gửi REGISTER-REPLY message chứa các thông tin (………) cho client
   2. Nếu sai giao thức Server gửi ERROR message cho Client, chương trình báo lỗi và dừng lại
      1. Lỗi thiếu header….
      2. Lỗi sai thông tin….
      3. …..
8. Client kiểm tra REGISTER-REPLY message có đúng giao thức đã quy định không
   1. Nếu đúng giao thức, Client gửi DATAPORT message chứa các thông tin (………) cho server
   2. Nếu sai giao thức Client gửi ERROR message cho Server, chương trình báo lỗi và dừng lại
      1. …
      2. …
      3. …
9. Server kiểm tra DATAPORT message có đúng giao thức đã quy định không
   1. Nếu đúng giao thức, Server mở data socket (TCP) và gửi DATAPORT-REPLY message chứa các thông tin (………) cho Client
   2. Nếu sai giao thức Server gửi ERROR message cho Client, chương trình báo lỗi và dừng lại
      1. …
      2. …
      3. …
10. Client kiểm tra DATAPORT-REPLY message có đúng giao thức đã quy định không
    1. Nếu đúng giao thức, Client kết nối đến data socket mà Server vừa mở
    2. Nếu sai giao thức Client gửi ERROR message cho Server, chương trình báo lỗi và dừng lại
       1. …
       2. …
       3. …
11. Client gửi thông tin CPU, RAM, DISK cho Server định kỳ theo khoảng thời gian đã được quy định trong REGISTER-REPLY message
12. Server phân tích và hiển thị thông tin được Client gửi lên cho người dùng xem

Server đổi thời gian gửi định kỳ của Client:

1. Ở phía giao diện của Server, người dùng nhập thời gian gửi định kỳ của Client cần thay đổi và bấm Change interval
2. Server gửi CHANGE-INTERVAL message chứa thời gian định kỳ cần thay đổi cho Client thông qua UDP socket
3. Client kiểm tra CHANGE-INTERVAL message có đúng giao thức đã quy định không
   1. Nếu đúng giao thức, Client cập nhật lại thời gian gửi định kỳ
   2. Nếu sai giao thức Client gửi ERROR message cho Server, chương trình báo lỗi và dừng lại
      1. …
      2. …
      3. …

Server đổi TCP port mà Client dùng để gửi dữ liệu:

1. Ở phía giao diện của Server, người dùng nhập TCP port cần thay đổi và bấm Change TCP port
2. Server gửi CHANGE-DATAPORT chứa TCP port cần thay đổi cho Client thông qua UDP socket
3. Server mở TCP socket mới và chờ Client connect lại để gửi dữ liệu
4. Client kiểm tra CHANGE-DATAPORT message có đúng giao thức đã quy định không
   1. Nếu đúng giao thức, Client cập nhật lại TCP port để gửi dữ liệu đến Server
   2. Nếu sai giao thức Client gửi ERROR message cho Server, chương trình báo lỗi và dừng lại
      1. …
      2. …
      3. …
5. Client đóng TCP socket hiện tải và mở TCP socket mới theo TCP port đã thay đổi để gửi dữ liệu cho Server