Create a Multi-Client TCP Servers using threading (thread class or runnable interface).

Server:

1. If the sever is idle (no client connected to the server), an information message should be displayed "Waiting for clients on port {*port_num*}"
    Note: *port_num* is the port number on which your server is running
2. When a client tries to connect to the server, an information message should be displayed "Got connection from {*client_IP*}:{*client_port*}"
    Note: *client_IP* is the IP Address of your client and *client_port* is the port number on which your client is running
3. Show the number of active clients after every client connection, an information message should be displayed "Active Connections = {*active_count*}"
    Note: *active_count* is the number of active clients
4. Read the information sent from the client
    a. Display an information message "Welcome to the server {server_IP}:{server_port}"
        Note: server_IP is the IP Address of your server and server_port is the port number on which your server is running
    b. If the client wants to save the data to the server
        i. If the file with the client IP exists on the server (i.e. 10.10.0.187.txt)
            a) Append the information in the file
        ii. If the file with client IP does not exists on the server
            a) Create a file with IP address (i.e. 10.10.0.187.txt)
            b) Save the information to the file
        iii. Display information message "Information saved for client {*client_IP*}"
            Note: *client_IP* is the IP Address of your client
    c. If the client wants to read the data from the server
        i. Check if the file for client exists on the sever
            a) Display information message "Information for client {*client_IP*}"
                Note: *client_IP* is the IP Address of your client
            b) Display information from the file
        ii. If no file for the client exists on server
            a) Display information message "No Information found for client {*client_IP*}"
                Note: *client_IP* is the IP Address of your client

Client:

1. Display information message when client becomes active "Client {client_IP} is active."
    Note: *client_IP* is the IP Address of your client

2. Check if the Client wants to Get information from the sever of upload some data to the sever
   a. If client wants to upload data to server
      i. Read data from input stream and upload to server
   b. If the client wants to get data from server
      i. Read the data from the server
      ii. Write the data to the file on client side
         Note: File name should be {ClientIP}_{ServerIP}.txt (I.e. 192.168.0.1_10.10.0.187.txt)

Important:

1. Do proper exception handling (try, catch, finally)
2. Connections to be manage properly. Close connection when you are done
3. Use character streams
4. Check all corner cases (valid connection, file not found etc.)

Hint:
1. Do point 4 of server in run method.