FACULTY OF INFORMATION TECHNOLOGY

Bachelor of Science in Informatics and Computer Science

ICS4104 – Distributed Systems

Assignment – Inter-process Communications in Distributed Environment (Worth 15%)

GROUP MEMBERS:

MUNYUI JULIE	**97460**
WANJIKU STEPHEN	**101019**
MUKOSI MEGAN	**101681**
MWANGI THOMAS	**102600**

This document presents the summary of all activities that were to be achieved in the task.

→ On running the program, we meet the landing page or server side as shown in figure 1:

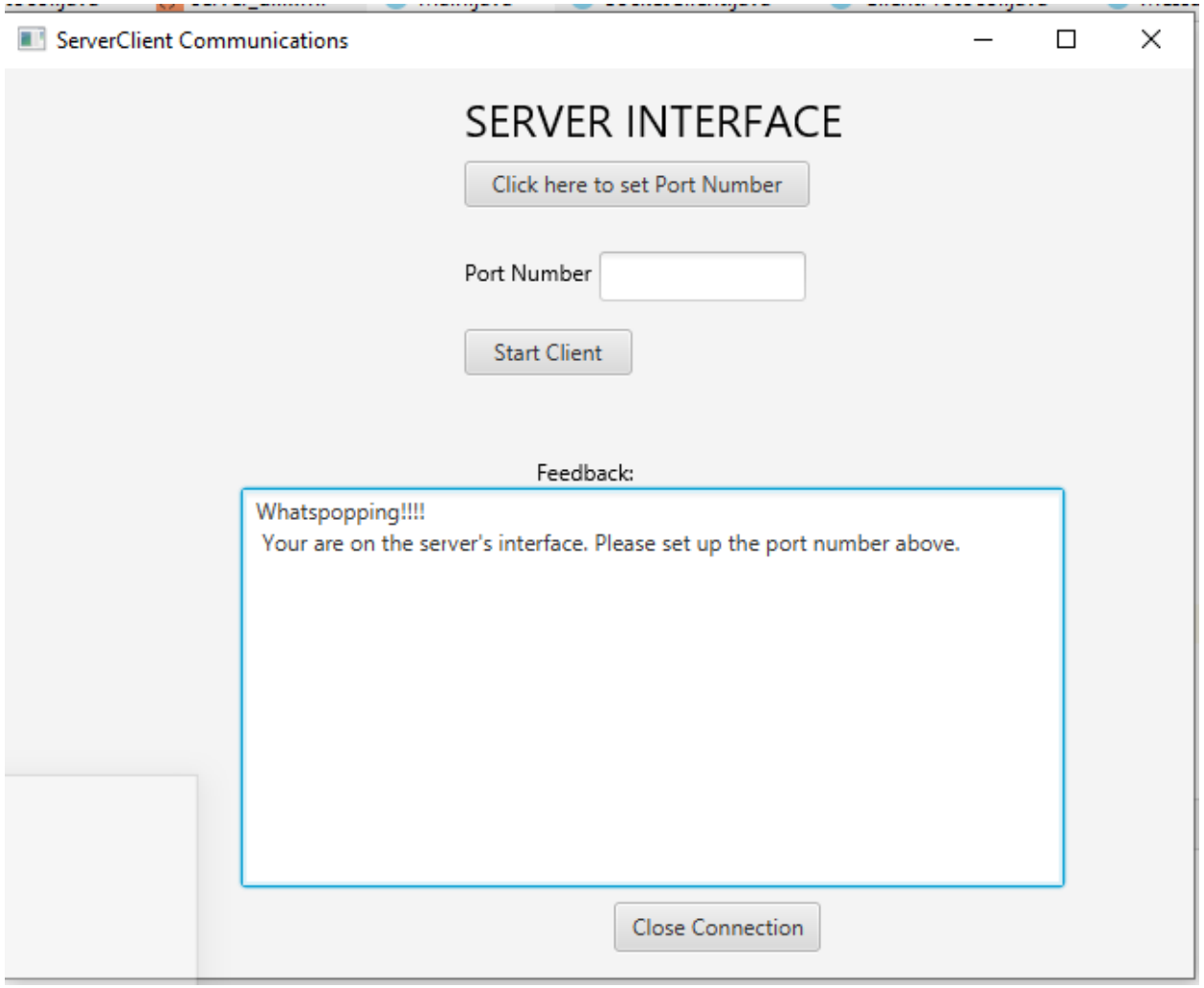

Figure 1 Server side

→ We are then requested to set the server's port number which will be necessary for setting up a connection with a client.
→ We proceed to set the port number as shown in figure 2 below:

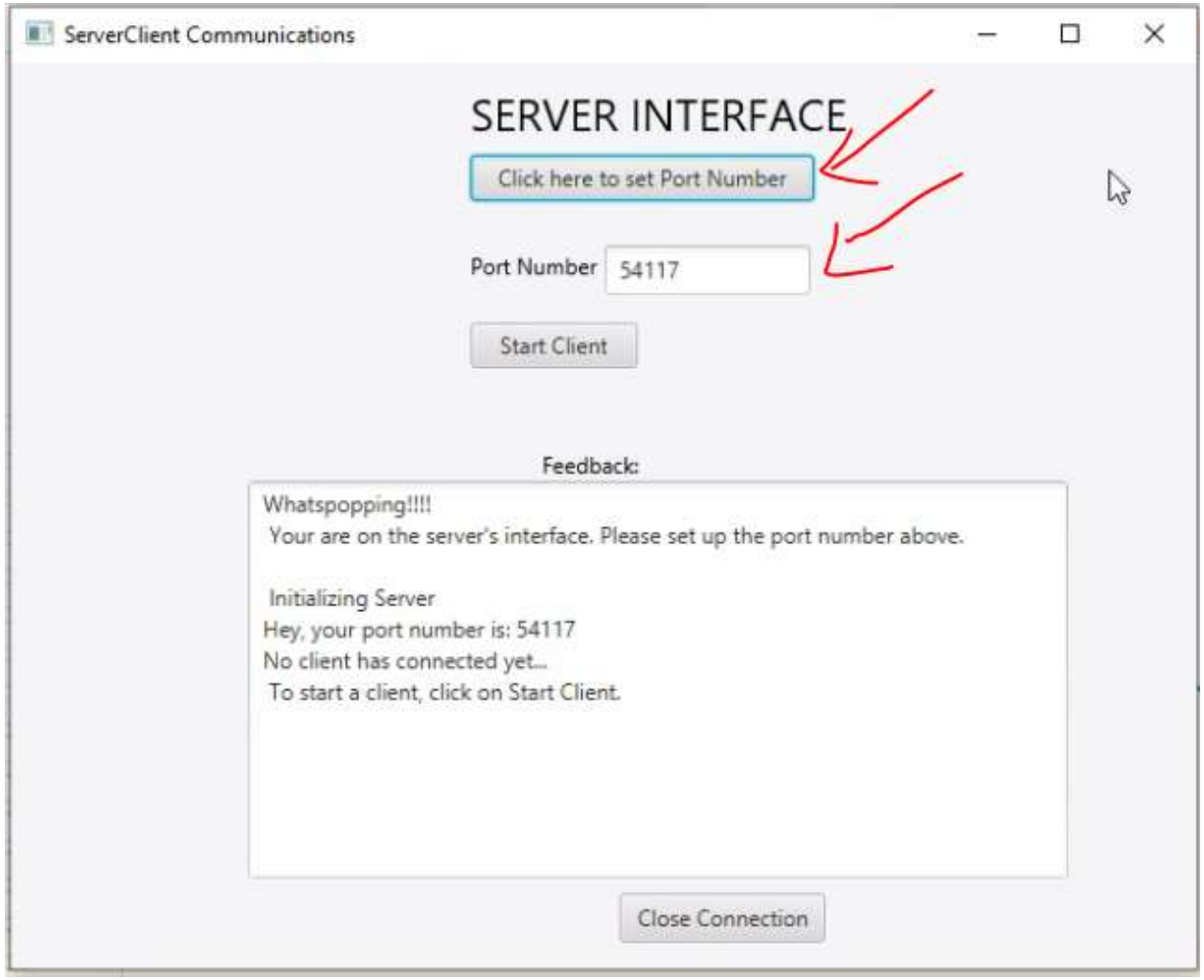

Figure 2 Set port number

→ Thereafter, we must start a client as shown in figure 3 below:

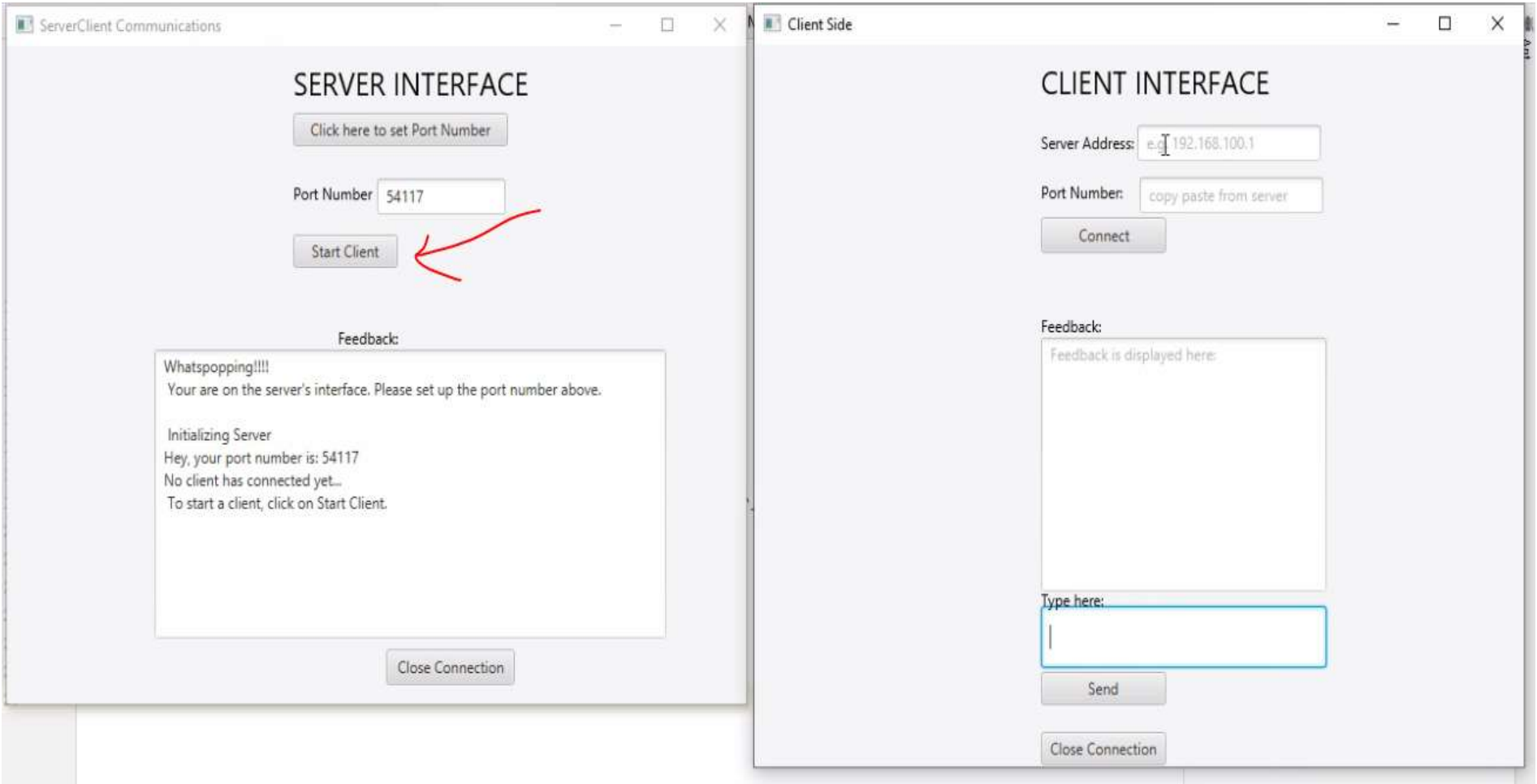

Figure 3 Start Client

→ For a connection to be established between the client and server, we must key in the server's address and port number.
→ Therefore, check the server's address on cmd as shown in figure 4 below:

```
C:\Users\Julie Munyui>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 4:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::d45a:4c01:fcd1:44f3%7
   IPv4 Address. . . . . . . . . . . : 192.168.100.36
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.1
```

Figure 4 Check Server's IP

→ Once you have the ip and port, set up a connection as shown in figure 5:

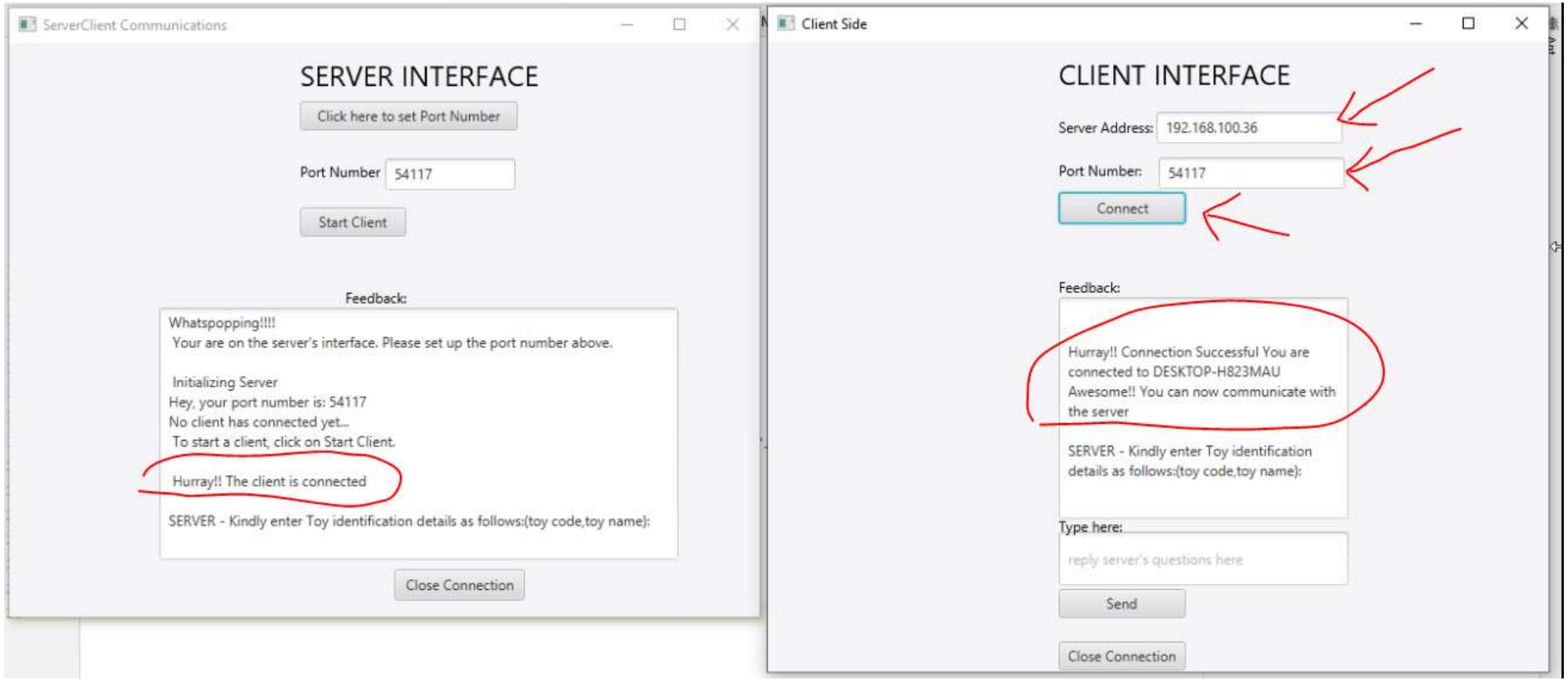

Figure 5 Connect client to server

→ Now that we are connected, we can carry out the given tasks

1.

- ✓ Server: Ask the client program to send the toy identification details (toy code, toy name)
- ✓ Client: Send the toy identification details (toy code, toy name) to the server program

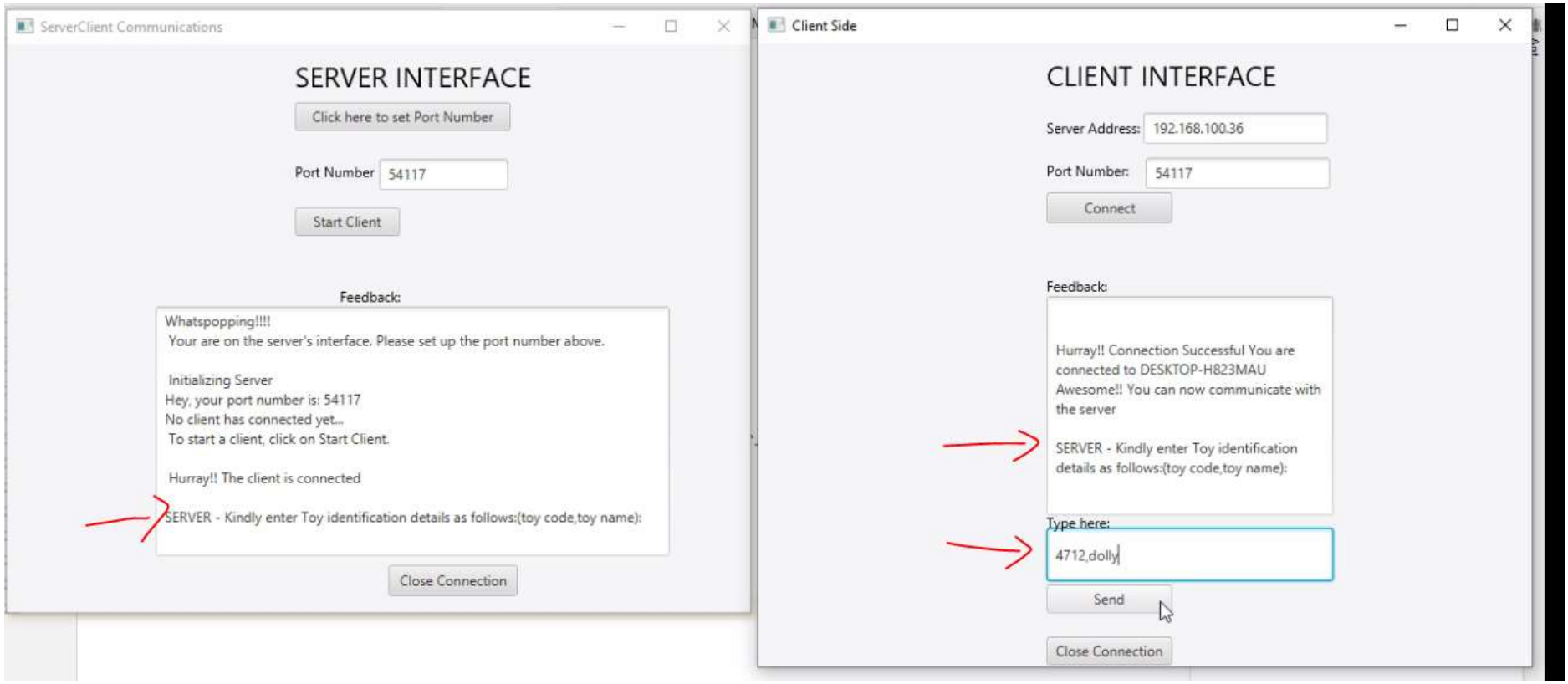

Figure 6 toycode,toyname

2.

- ✓ Server: Ask the client program to send the toy information (name, description, price, date of manufacture, batch number)
- ✓ Client: Send the toy information (name, description, price, date of manufacture, batch number) to the server program

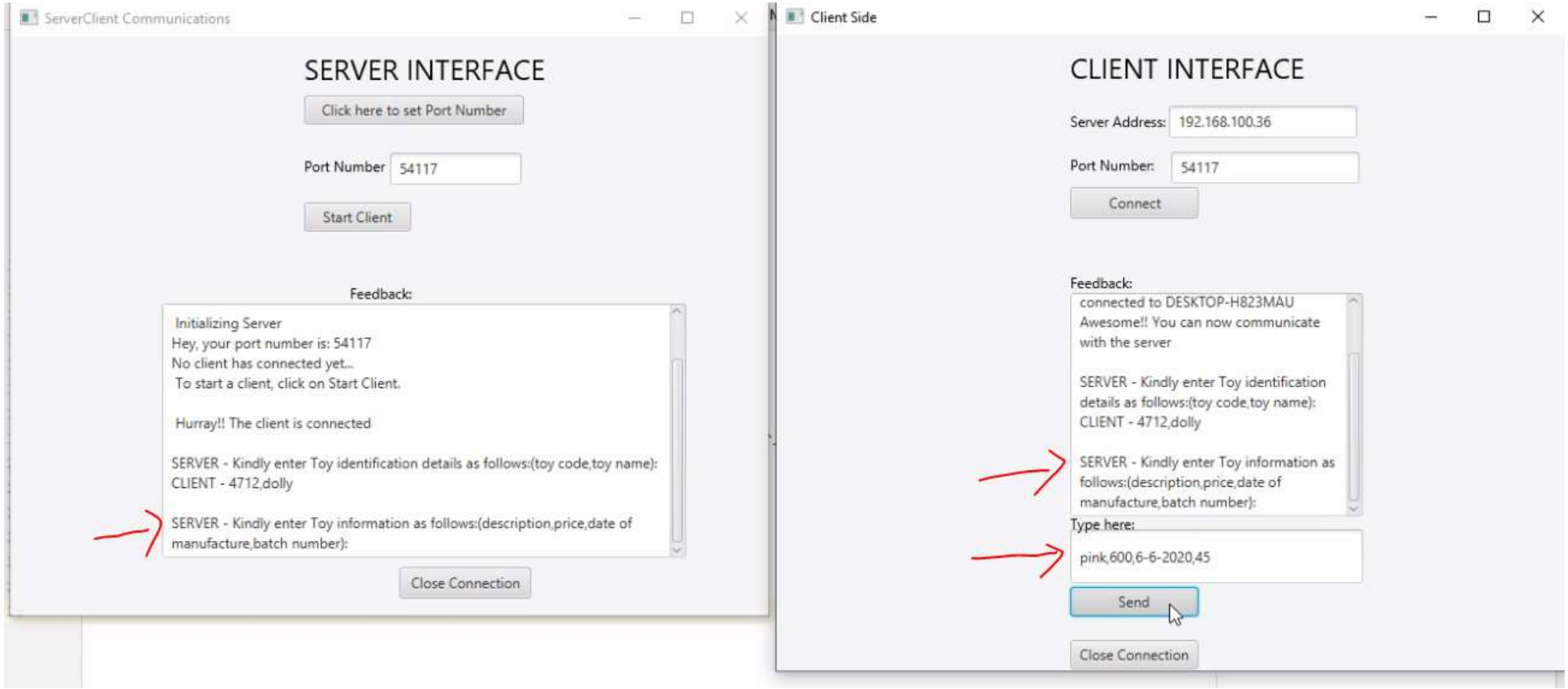

Figure 7 name, description, price, date of manufacture, batch number

3.

- ✓ Server: Ask the client program to send the toy manufacturer details (company name, street address, zip-code, country).
- ✓ Client: Send the toy manufacturer details ((company name, street address, zip-code, country) to the server program

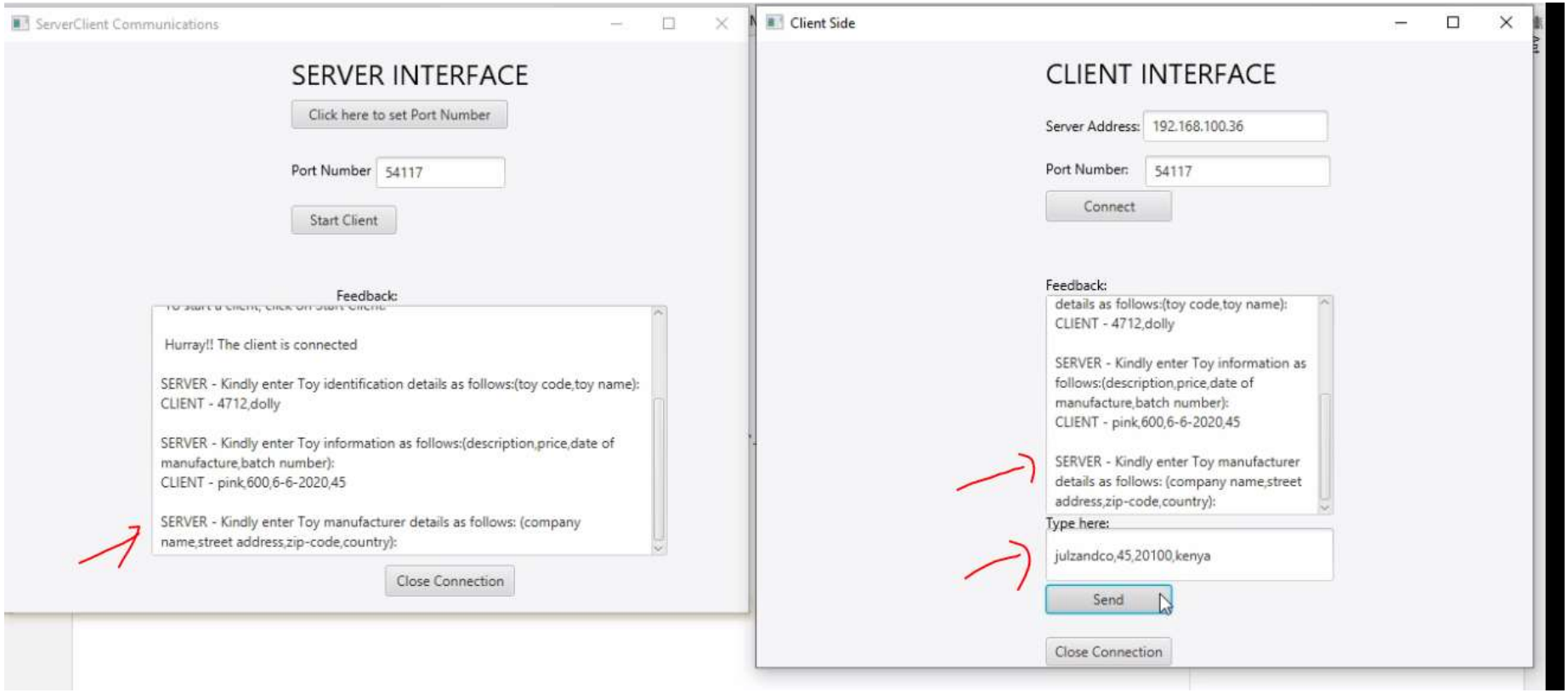

Figure 8 toy manufacturer details ((company name, street address, zip-code, country)

4.

- ✓ Server: Ask the client program to send a thank you message with a unique identification code.
- ✓ Client: Send a thank you message with a unique code (Innovate J) to the server program

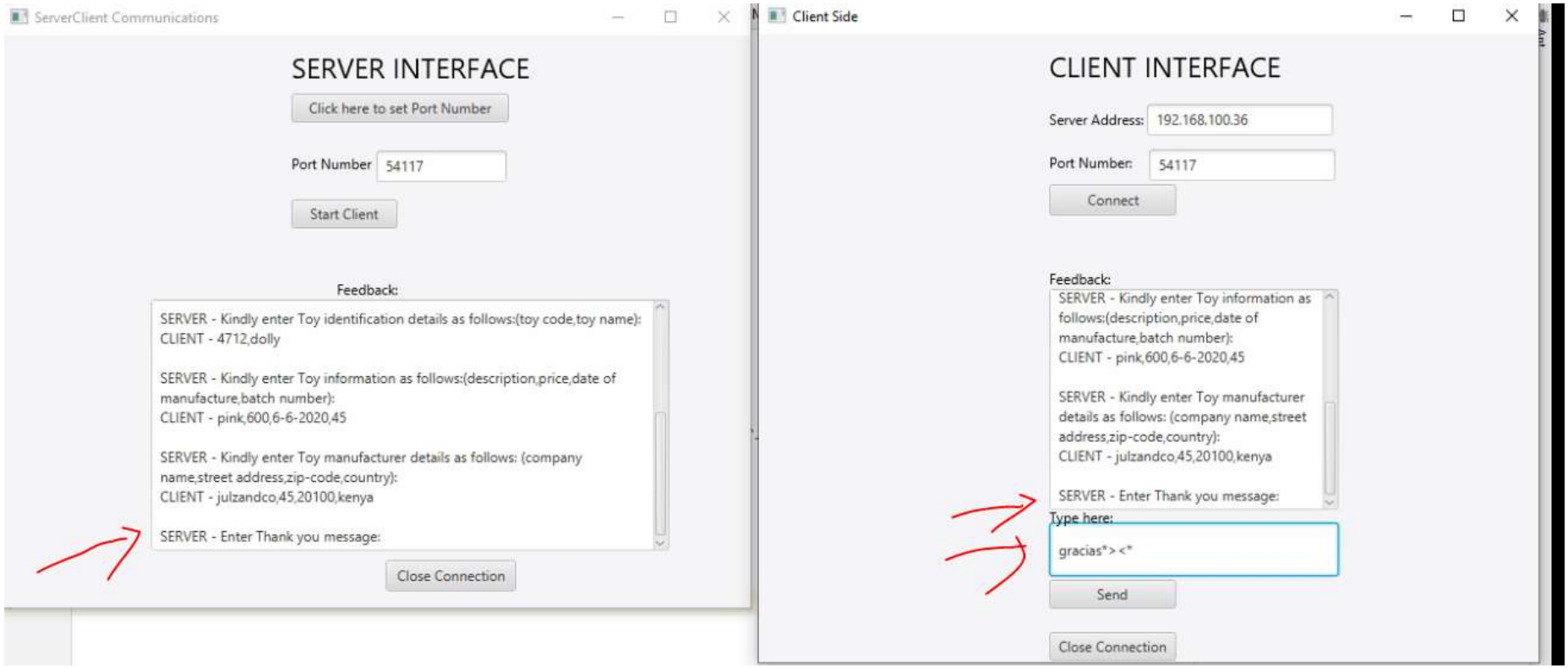

Figure 9 Thank you

5.

- ✓ Server: Ask the client program to send all the toy information in one single instruction.
- ✓ Client: Send all the above toy information in one single instruction to the server program

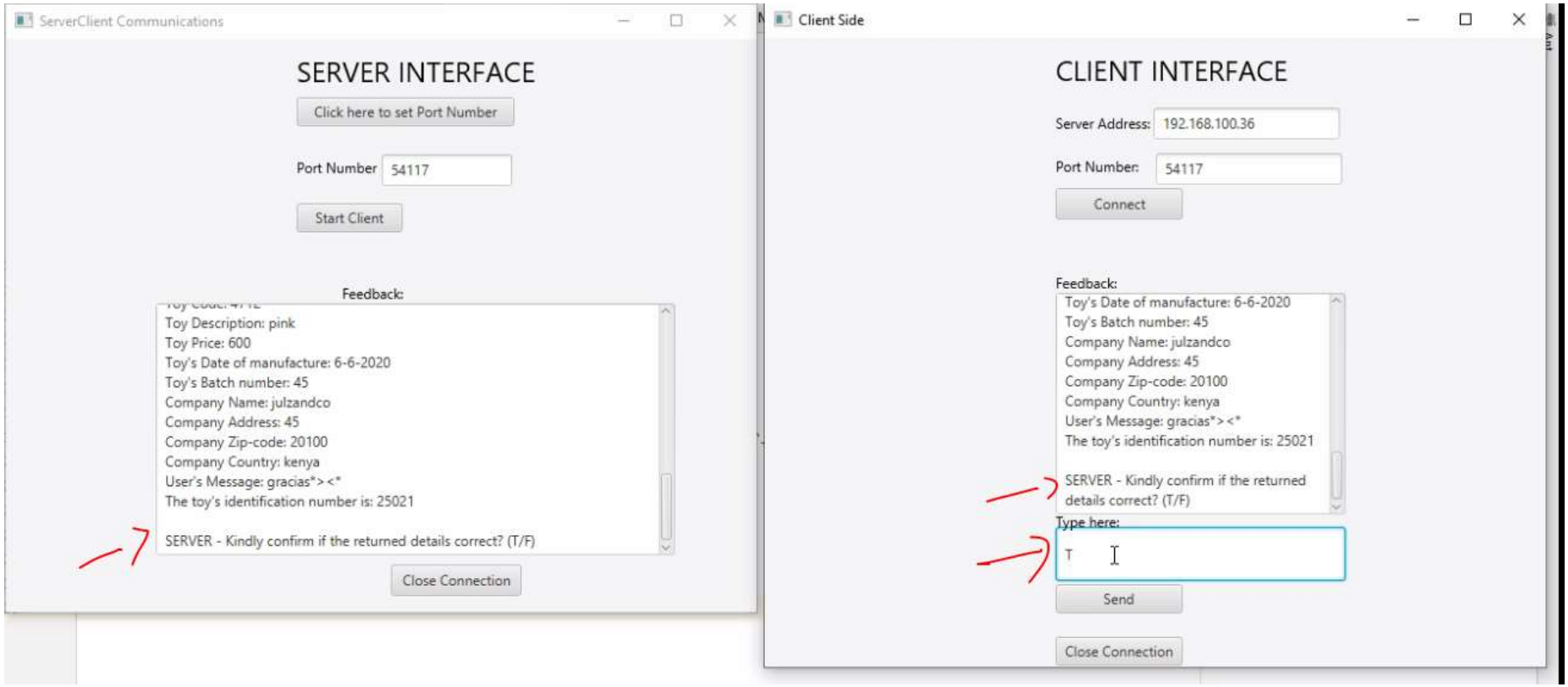

Figure 10 all in one

6.

- ✓ Server: Server to send to client a message to indicate the communication succeeded or aborted.

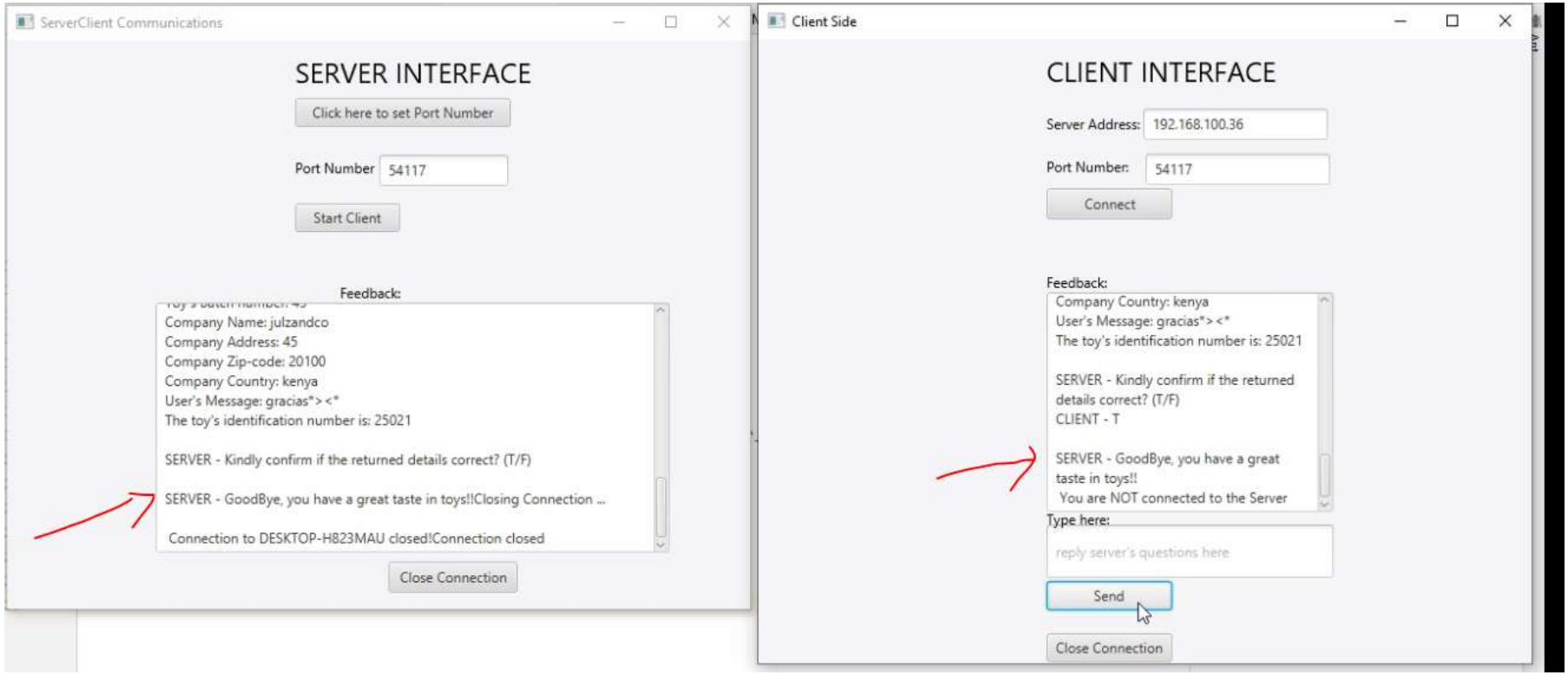

Figure 11 succeeded or not