Design a simple, fully functional

“Client/Server Reliable Chat Application”

-----------------------------------------------

**Minh Ngo & Thi Phan**

University of Memphis

**1 Introduction**

“Client/Server Reliable Chat Application” which is named “chatApp”. The application will contain a server and two or multiple clients for chatting. The application allows a client connect with the server and with other clients through the server. It also has function of forwarding message and exiting whenever the client want.

**2 Requirements**

**2.1 Tentative plan**

Here is our plan for the project:

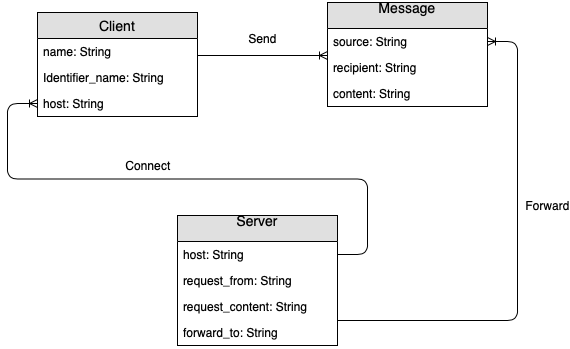
* Design document for the chat application include the text description, use class diagram/s, flow chart, and state transition diagram to explain our tentative implementation and design.
* Install necessary packages for programming
* Writing code framework for both client and server which have the major data structures and functions (APIs) defined based on the design document.
* Writing code showing the connection between clients and server which should show how the server assigns unique identification for each client.
* Finalize the code showing communication between the clients include the instructions on how to install and run the program.
* Testing, analysis and final report including design & implementation details.

**2.2 Final project features**

The features of the final project include a server and two or more than two clients for chatting. Each client be able to communicate with the server. The server will assign a unique identifier name to each of clients. The server will provide the list of all the available clients connected to it when a new client is connected to the server. The client will use the identification which is provide from the server to send the message to another clients. This app will have the functions such as forwarding message and exiting when the clients want.

The final project might have abilities implement chatting between multiple clients and simple encryption (e.g TLS). However, we do not decide to do it yet, since we are not sure how much the time consuming would be.

**3. Class Diagram/s**



**4. Flow Chart**

Connect to server

Get identifier name

Receive online list

Send request

to server

Send message via server

Chat

“.exit”

Flow chart on client side

Receive message from other client(s)

Client connect

Assign identifier name

Provide online list

Client request

Identify recipient

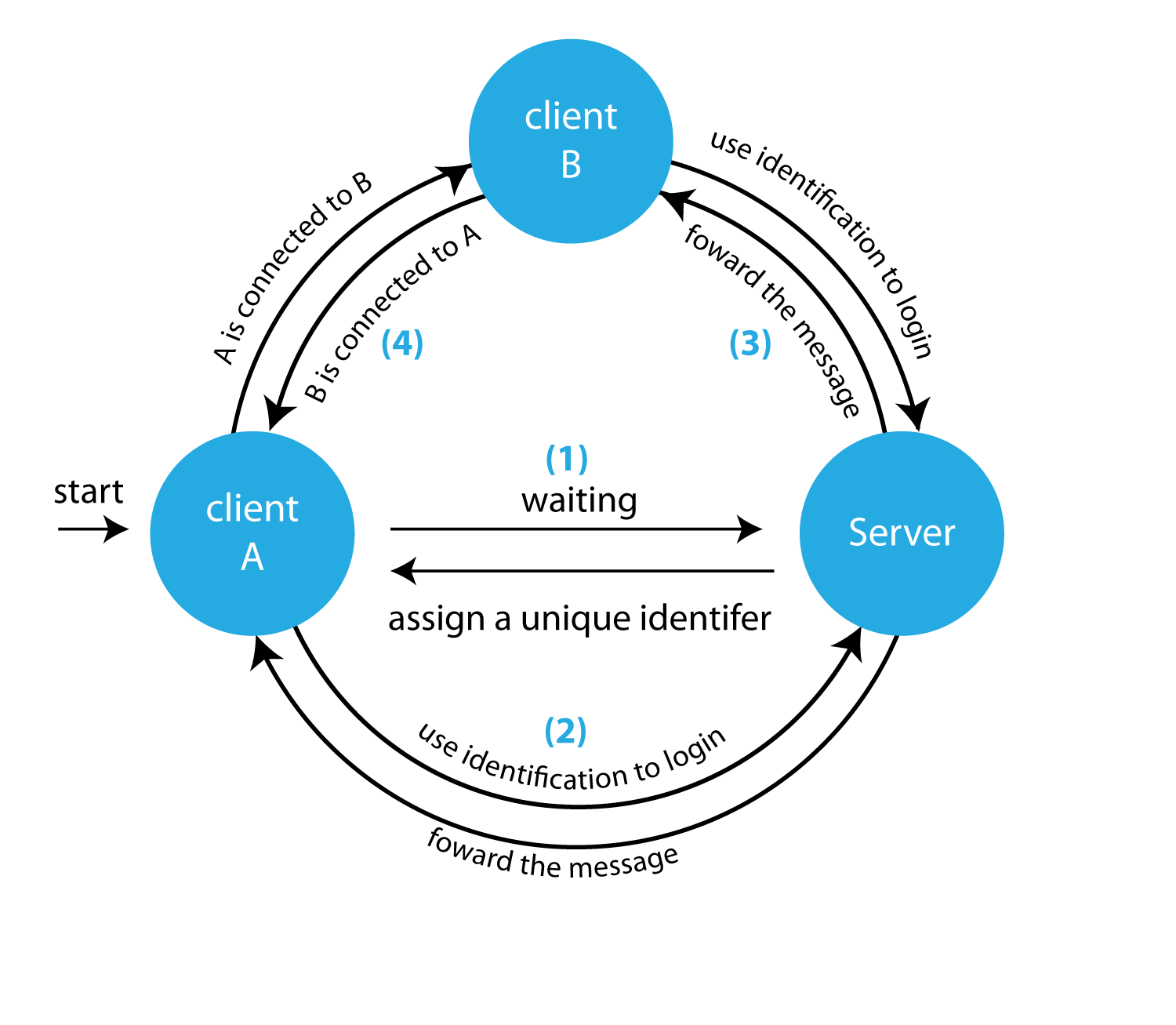
Forward message

Chat

“.exit”

Flow chart on server side

**5. State Transition Diagram**



**6. Computing resources**

Programming language for this project is Python.

**7. How to use chatApp**

**Install:**

In terminal:

* To run server, use command: $python3 server.py
* To run client, use command: $python3 client.py localhost

**Using chat room:**

* Type in message and press ‘enter’ or ‘return’ to send message to chat room

**Using private chat:**

* Type in message with a client ID in the following structure to send message to the client.

**message $$ client\_ID**

* For example: hello $$ 1234567[John]

‘hello’ is a message.

$$ is to indicates that this is a private message.

1234567[John] is a unique ID of a client.

**7. References**

# Rules for Creating Flowcharts - <https://www.edrawsoft.com/flowchart-rules.php>