# Two-phase Commit (2PC)

There are three files in the zip

1. Server.py
2. Participant.py
3. Coordinator.py

   1, 2, & 3 are to be opened in the IDE or right click and open in notepad to view the code.

Server.py is my server program. Participant.py is my client and Coordinator.py is my coordinator program. Server and coordinator program is executed (RUN) once and Participant program is executed (RUN) 3 times to create 3 Participants.

In the Participant and Coordinator GUI, the user has to press log in button to connect to the server. First server asks to set up username for the user. The user types the message in box and press 'send' button to send it to other Participants. And any message from other participants and coordinator will be displayed automatically. To quit the chat room the user has to press 'Quit' button.

[uthira]: 0:00:17:hi

In the above message 0:00:17 – it is hours:minutes:second.

After getting connected to the client, the server displays the message 'Client (uthira) connected' and does this every time a client is connected. When the client sends a message the message is displayed in HTTP format that stated in the requirement sheet. The sample is as below.

```
Post/client.http
 Host: 10.219.139.223
 User-Agent : /pycharm
 Content-Type : Text
 Content-Length : 10
 date : 2018-03-03 22:34:52.180845
0:00:17:hi
```

First participants has to log in and then the coordinator logs in at the last

To send a string the coordinator sends it in the following format
string: Hello
Then presses the Request button to ask the participant vote
Case 1: All the participants Gives Commit
    Result: Coordinator gives Global commit
Case2: One of the participant doesn't respond in time
    Result: Coordinator gives global abort
Case 3: Coordinator Crashes
    Result: Participants ask for the status and take action accordingly

The files are return in Text file and can be retrieved if the same username is provided when reopening the participant's Gui.

IDE used: PyCharm , Python 3

References:

Book: Distributed System principle and paradigms by Andrew S. Tanenbaum and Maarten Van steen

https://stackoverflow.com/questions/31080499/python-socket-running-server-and-client-from-the-same-pc

http://www.bogotobogo.com/python/python_network_programming_server_client.php

https://www.binarytides.com/code-chat-application-server-client-sockets-python/

https://github.com/metonimie/python-networking/blob/master/chat/tcp_client.py

https://github.com/metonimie/python-networking/blob/master/chat/tcp_server.py

http://codingnights.com/coding-fully-tested-python-chat-server-using-sockets-part-1/

https://www.geeksforgeeks.org/simple-chat-room-using-python/

https://www.binarytides.com/python-socket-programming-tutorial/

https://www.binarytides.com/category/programming/sockets/python-sockets-sockets/

https://stackoverflow.com/questions/26445331/how-can-i-have-multiple-clients-on-a-tcp-python-chat-server

https://stackoverflow.com/questions/23828264/how-to-make-a-simple-multithreaded-socket-server-in-python-that-remembers-client

https://www.python-course.eu/tkinter_text_widget.php

http://danielhnyk.cz/simple-server-client-aplication-python-3/

http://docs.python-requests.org/en/master/user/quickstart/

http://python-textbok.readthedocs.io/en/1.0/Introduction_to_GUI_Programming.html

https://www.python-course.eu/python_tkinter.php

http://sebsauvage.net/python/gui/

http://www.sfentona.net/?p=2239

https://stackoverflow.com/questions/42976749/python-3-6-multithread-tcp-echo-server-for-more-than-one-client

https://stackoverflow.com/questions/36060346/creating-a-simple-chat-application-in-python-sockets

https://stackoverflow.com/questions/23507779/python-network-threading-simple-chat-waits-for-user-to-press-enter-then-gets-me

https://medium.com/swlh/lets-write-a-chat-app-in-python-f6783a9ac170

https://stackoverflow.com/questions/20745352/creating-a-multithreaded-server-using-socketserver-framework-in-python