**User Documentation**

(In the version4 folder) You should run one server first, and then run two clients. Whatever you input in the client (except 0) will send to another client and save to both local blockchain. If you input a 0 it will show the local block chain.

**Develop Documentation**

**Server**

In the server I create socket and use two threads to keep deal with two clients separately. Each thread receive message from one client then send to another client.

**Client**

In the client I initiate the local blockchain. The client has two threads to deal with the message from server and the message send to server. When the client gets system input or get message from server, the client will print out the message and then store this message to local blockchain.

**Block**

Each block will store previous hash, data string array and current block hash. Current hash is generated from the data string array and the previous hash.

**BlockChain**

It is an ArrayList of block. It has a first block when you initiate it by construct. It has an addBlock() method and a toString() method.

**Conclusion**

This big assignment used *socket* to realize the communication between client and server. *Socket* is a useful transient communication tech. This assignment used *thread* in both client and server to keep deal with different things. This assignment used the basic idea of blockchain to store the data.