**Classes used :**

**UDPServer** :

This class performs all the operations of the server.

There are two threads running : One is the main thread and the other one is just for receiving data.

**UDPClient:**

This class performs all the operations of the client.

There are two threads running:One is the main thread and the other one is just for receiving data.

**Item:**

It contains all the required details of an Item.

item\_code

name

owner

translimit

startbid

buynow

description

current\_bid

transaction\_count

buyername

**client:**

It contains the necessary details of the client.

username

cli\_IP

cli\_port

**offline\_user:**

It contains the details of an offline message.

name

message

**Features Implemented :**

The start commands on the server as well as the client :

**sobs -s port\_no**

**sobs -c IP port\_no**

**Errors thrown :**

Port number out of range.

Invalid IP.

Less or more number of arguments.

Anything other than –s on the server.

Anything other than –c on the client.

**Client commands :**

**register and deregister**

All the requested features have been implemented.

When the commands are entered the server sends acknowledgement and appropriate messages are printed on the client window.

The client information tables are broadcasted and updated.

No information can be sent or received after deregister.

Sends offline message if any are present.

**sell**

Results printed on the window :

[Error :arguments]

[<item-name> added with number <item-code>]

The description for the item needs to be entered separated by underscores. Example can be seen in the test cases mentioned after this section.

**info**

[Error : Empty]

[Error : Item not found]

[Details of a particular item]

Details of all the items.

**bid**

[Error : arguments]

[Error:owner]

[Error:duplicate bid]

[Error:item not found]

[<item number> <item name> <current bid>]

Sold [<item number> <item name> <current bid>]

Purchased [<item number> <item name> <current bid>]

Adds to offline messages if the client is not online anymore.

**buy**

Implemented like the diagram shown.

Customer to seller, seller to the server and then sold and purchased indications to both.

**offline-buy**

Customer to server, purchased and seller offline indication to customer and offline message saved for the seller.

**Feature not implemented:**

I could not successful ly implement the sending of acknowledgement from client to server and server to client everytime it receives a packet . If the acknowledgement is not received even after 5 times the message is stored in the offline messages for that user. Everytime I added this portion of my code the program went into an infinite loop on the register command itself. I could not successfully debug it. I have commented that portion of the code in my program.

**Testing**

The program has been tested with one server and three client running on the clic lab machine.

**Makefile contents :**

JFLAGS =  -g

JC = javac

JCLASSPATH = -cp .:../../bin

.SUFFIXES: .java .class

.java.class:

        $(JC) $(JFLAGS) $(JCLASSPATH) $\*.java

CLASSES = \

        Item.java \

        client.java \

        offline\_user.java

SOME =  UDPClient.java \

        UDPServer.java

default:classes

classes :$(CLASSES:.java=.class)

main :$(SOME:.java=.class)

clean:

        $(RM) \*.class

SubmissionFolder/FinalUDPServerClient/src/bidding : make classes (builds the dependency classes)

SubmissionFolder/FinalUDPServerClient/src/bidding : make main (builds the Server and Client classes)

To run the code you need to come back one folder to src and execute java bidding.UDPServer(for server) , java bididng.UDPClient(for client)

**Test Cases:**

**Server:**

sobs -s 3567869

[Error : port number out of range ]

sobs -s

[Error : 2 arguments required ]

sobs -s 1236

Server has been started

**Client 1:**

sobs -c 767838 1234

[Error : Invalid IP ]

sobs -c 127.0.0.1 1236

sobs>register nandita

sobs> [Welcome nandita, you have successfully signed in.]

sobs>sobs>sobs>[Client table updated]

sobs>info

sobs>[Error : empty]

sobs>sell pen 3 200 600 fountain\_pen

sobs>[pen added with number 1]

sobs>sell pencil 2 500 800 new\_pencil

sobs>[pencil added with number 2]

sobs>sobs>[Client table updated]

sobs>bid 1 100

sobs>[ Error : owner ]

sobs>sobs>[Client table updated]

sobs>sobs>[ sold :1 pen 600 ]

sobs>sobs>[ geetha wants to buy your item 2 ]

sobs>sobs>sold 2 pencil

sobs>sobs>[Client table updated]

sobs>sobs>[Client table updated]

sobs>

**Client 2:**

sobs>register geetha

sobs> [Welcome geetha, you have successfully signed in.]

sobs>sobs>sobs>[Client table updated]

sobs>info

sobs>1 pen nandita 200 600 fountain\_pen

sobs>2 pencil nandita 500 800 new\_pencil

sobs>bid 1 300

sobs>1 pen 500

sobs>bid 1 200

sobs>[ Error : duplicate bid ]

sobs>sobs>[Client table updated]

sobs>buy nandita 2

sobs>purchased 2 pencil 800

sobs>sobs>[Client table updated]

sobs>info

sobs>3 pan satish 400 600 frying\_pan

sobs>buy satish 3

sobs>[purchased 3 pan 600 ]

sobs> [ satish is offline. Request has been forwarded to the server ]

sobs>sobs>[Client table updated]

sobs>

**Client 3:**

Please enter your command

sobs -c 127.0.0.1 1236

sobs>info

sobs>[Error : Not registered]

sobs>register satish

sobs> [Welcome satish, you have successfully signed in.]

sobs>sobs>sobs>[Client table updated]

sobs>info

sobs>1 pen nandita 500 600 fountain\_pen

sobs>2 pencil nandita 500 800 new\_pencil

sobs>bid 1 100

sobs>[ purchased :1 pen 600 ]

sobs>sell pan 2 400 600 frying\_pan

sobs>[pan added with number 3]

sobs>deregister

register satish

sobs> [Welcome satish, you have successfully signed in.]

sobs>sobs> [ Offline message :[sold 3 pan ]

sobs>sobs>[Client table updated]

sobs>