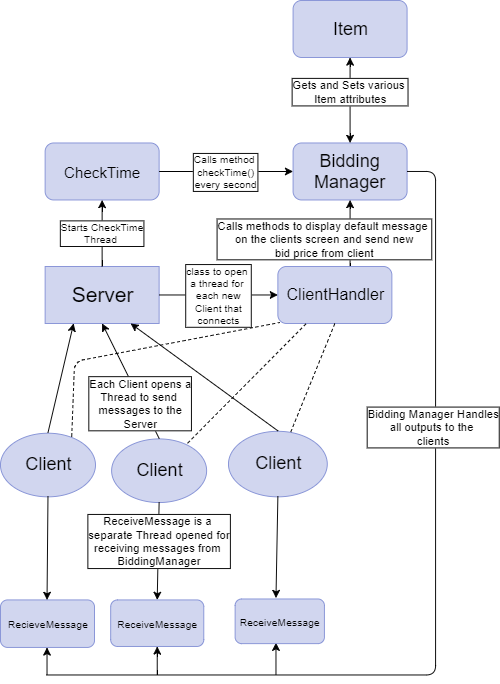
Distributed Systems – Assignment

Haashim Bari|C14498008

22.11.2017

**Declaration: “**I declare that this work, which is submitted as part of my coursework, is entirely my own, except where clearly and explicitly stated."

**Class Hierarchy**



**BiddingManager**

The diagram above covers mostly everything, but the Bidding Manager class needs to be explained further as it is doing the bulk of the work. It contains the following methods:

*Constructor – BiddingManager()*

* This is where I initialise array list holding the bid items
* The array list holds objects of the class “Item”, which has the attributes “itemName” and “startPrice”, which changed to “bidPrice” when a client bids on an item
* It then calls the method setMessage() with details of the bid item and bid price passed in as a String parameter

setMessage()

* This message is only ever called from inside BiddingManager
* Every time it is called it updates the message String and calls the printToClients() method

getMessage

* Only called by the printToClients() method to get String message

*addPrintWriter()*

* This method is called from the ClientHandler thread
* It is called every time a new client connects to the server
* It adds the client to the array list of printwriters that will be outputted to

*checkBidPrice()*

* This method is called from the ClientHandler thread
* It is called every time a client enters in a new bid price
* It checks to see if the bid price entered is less than, equal to or greater than the current bid price
* If less than or equal to it sends an error message to the client
* Else it will set the new bid price

printToClients()

* This method is only ever called from inside the bidding manager
* It iterates through the list of print writers and checks to see if they are null
* If they aren’t it will send them all a message by calling the getMessage() method

displayDefaultMessage()

* This method is called every time the user inputs a new bid price
* It calls the setMessage() method with details of the bid item with the new bid price

*getIsBidRunning()*

* Gets Boolean variable isBidRunning
* Used by the CheckTime class to see whether or not too continue running the program on the clientside

*updateBidItem()*

* Called in the checkTime() method
* If there are bid items left in the list it will get the current item
* Then it will call displaDefaultMessage() to set the message
* Else it will set isBidRunning to false and will set a message saying the auction is over which will be printed to the clients

*checkTime()*

* This is called from the CheckTime class as shown in the architecture diagram above
* It is called every second, and takes in an integer as a parameter called sec
* Sec is being decremented each second in the CheckTime class and when it reaches zero resets back to the original 15
* If sec = 0 a nested if statement is used
* If the current bid price is the same as the starting price, the current item is re-added to the array list in the last position, and set a message saying no one bid on the item
* Else it will remove the current item from the array list and set a message saying who won the bid.
* Else if sec = 5 it sets a message telling the clients 5 seconds are left

**Run the Server/Client**

You can run the server and clients by running the bat files “server” and “client”

Or

In the command line, navigate to the correct folder and use:

“java Server” to run the server

“java client” to run the client