**Assignment**

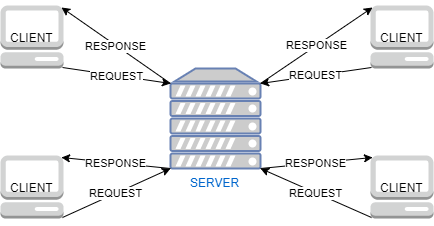
**12TH November 2017**

**Daniel Tilley – C14337041**

**Distributed Systems (CMPU4021)**

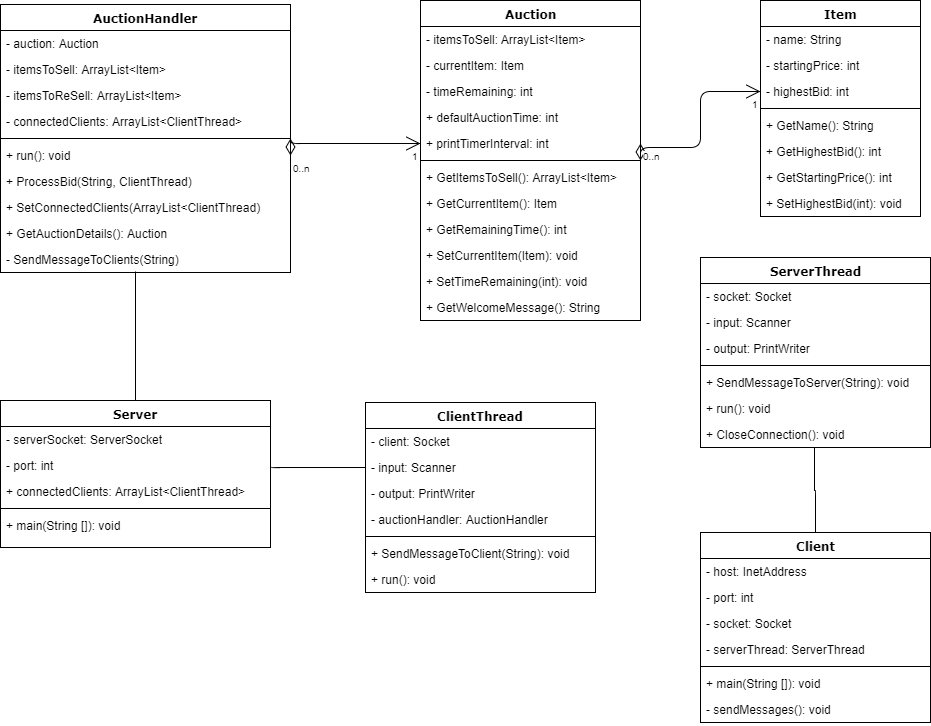


**Design**



My design for this assignment follows a client server architecture model as shown above. Multiple clients can connect to a server using sockets. The server will store and maintain a list of connected clients. Both clients and the server will be able to communicate with one and other via streams.

**Implementation**



The server and client class both contain main methods and will be used as entry points into the respected programmes. Auction Handler, Client Thread and Server Thread will all run as background threads and will be in charge of maintaining the auction and connections respectively. Auction and Item are both model classes used to store data relevant to both the auction and items respectively.

**How to use**

The folder structure should be laid out as follows:

* Parent Folder
  + Code Folder
  + Batch Folder
  + Document Folder

Before running any of the batch files, please modify them so that the directory which is navigated to is correct:



After this has been changed, you can run the batch files in the following order:

1. Compile
2. Server
3. Client (multiple times)

Or alternatively you can navigate to the code folder and run the following commands:

1. Javac \*.java
2. Java -cp . Server
3. Java -cp . Client (multiple times)

The server will run and accept several clients at a time. It will also print out information relating to a single auction at a time. Once a client has run, instructions will be printed on screen to the user about what to do.

**Declaration**

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