



Distributed Systems

Assignment

On-Line Auction System

Name: **Eryk Szlachetka**

Student Number: **C14386641**

Course Code: **DT282/4**

Date: **22/11/2017**

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

Introduction	4
Client.....	5
Server	5
Data Flow Diagrams.....	6
Client.....	6
Server	7

Introduction

This is a report developed for the Distributed System Assignment as part of the integral course work.

The idea of the system is to simulate an online auction system without implementing the actual credit card transactions. The system has to allow the client (user) to bid for the items that are listed in the server, while at the same time receive appropriate information such as time, price etc. The item has to be sold to the highest bidder.

The system is designed in a way to support socket connection from multiple clients connecting to the same server at the same time. The system consists of two main parts including the **server** and the **client**.

The code for the system has been written using **Java** language with the support of **Java sockets**.

In order to compile the programs, the provided bash files have to be run in the following order:

1. serverStarter.bat
2. clientStarter.bat

Alternatively the following commands will start the system:

```
Javac -classpath . *.java  
Java ResourceServer 8742  
Java ConsumerClient 8742
```

In order to run the program, ensure that PORT is specified in the command line, that the port is the same for both server and the client and that the port is available on the machine.

Client

Client software's main purpose is to provide input and output to the user, while communicating with the server and making necessary error checking.

The client software requirements have been specified in the assignment and are as follow:

- *Connects to the server. The item currently being offered for sale and the current bid or a reserve price is displayed.*
- *Enter the bid. The amount entered should be greater than the current highest bid.*
- *After a new bid is placed, the amount of the new bid must be displayed on the client's window/console.*

Server

Server software's main purpose is to create and **manage** the items that are listed for the auction, accept and **manage** connections but also to communicate with the multiple clients including notifying all of them when necessary.

The server software requirements have been specified in the assignment and are as follow:

- *Receive connections from multiple clients.*
- *After a client connects, notify the client which item is currently on sale and the highest bid (or reserve price).*
- *Specify the bid period. Max allowed 1 minute. When a new bid is raised, the bid period is reset back.*
- *When a new bid is placed, all clients are notified immediately. Clients should be notified about the time left for bidding (when appropriate).*
- *If the bid period elapses without a new bid, then the auction for this item closes. The successful bidder (if any) is chosen and all clients are notified.*
- *When an auction for one item finishes, another item auctioning should start. Minimum of 5 items should be auctioned, one after another and only one item at a time.*
- *Any item not sold should be auctioned again (automatically).*

Data Flow Diagrams

Client

The flow chart in Figure 1 describes the data flow in the **Client Software** that consist of three different classes:

Consumer Client – The main class that declares and executes the background threads.

Input Thread - A thread class that runs in the background, providing user the ability to input a bid for the item auctioned.

Listen Thread – A thread class that runs in the background, providing the software ability to communicate with the server and display the necessary information in real-time.

The connections highlighted using **red** colours are indicating when the background threads are starting to run where the one in **black** colour show the data flow in time sequence.

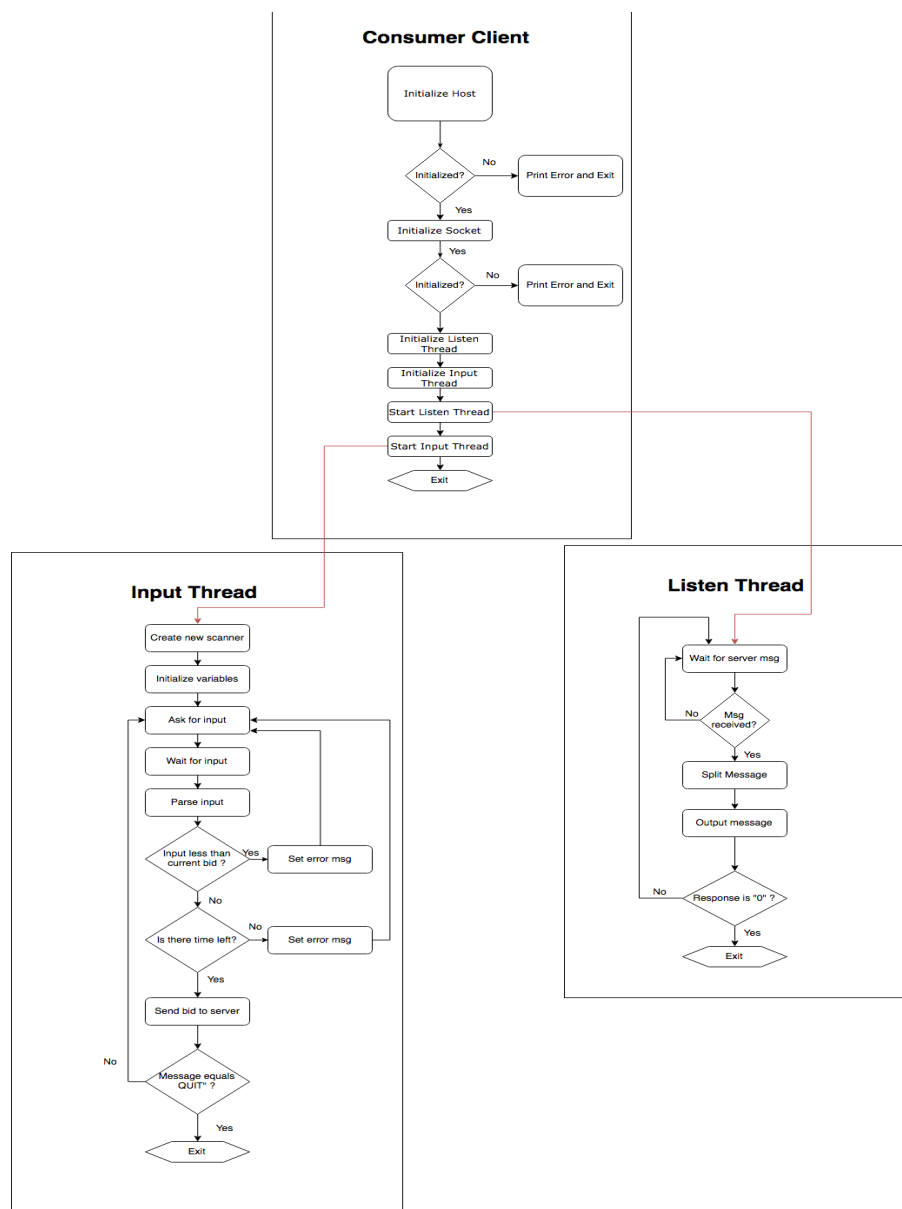


Figure 1: Data Flow Chart representing three different classes for the Client software.

Server

The flow chart in Figure 2 describes the data flow in the **Server Software** that consist of three different classes:

Resource Server – The main class that declares and executes the background threads and other classes.

Time Reducer - A thread class that's main purpose is to validate when to call the reduce the time for the item auctioned function from the Resource class, but also to set the new item to be auctioned and to call the notify functions from Notifier class when necessary.

Resource – A class, which defines the resource's attributes such as description, initial price, time etc.

Notifier – A class that is responsible for informing the client threads about changes, the class keeps track of the connected clients, iterates and informs them.

Client Thread – A class that deals with associated client, meaning that each client has its own Client Thread where the communication happens, such as receiving bid from the client.

The connections highlighted using **red** colours are indicating when the background threads and classes are starting to run where the one in **black** colour show the data flow in time sequence.

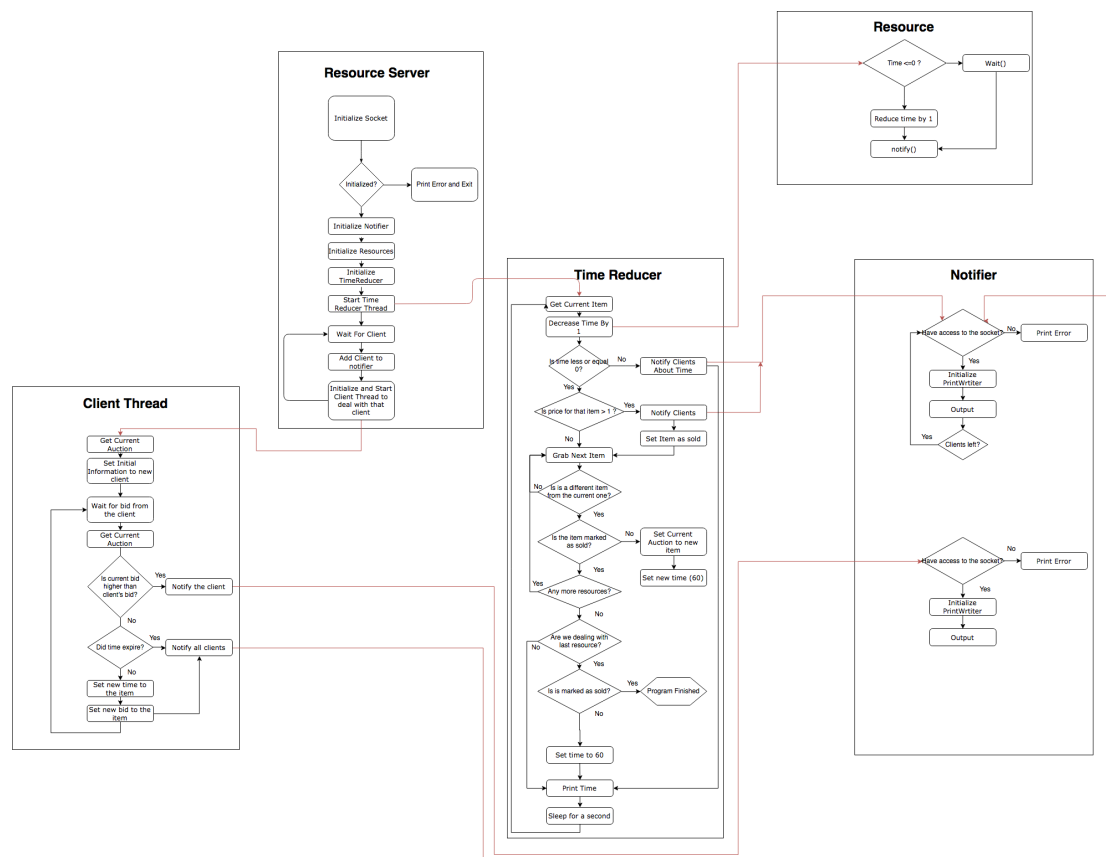


Figure 2: Data Flow Chart for the Server Software in Online Auction System.