CSCI/ECEN 5673: Distributed Systems
Spring 2023

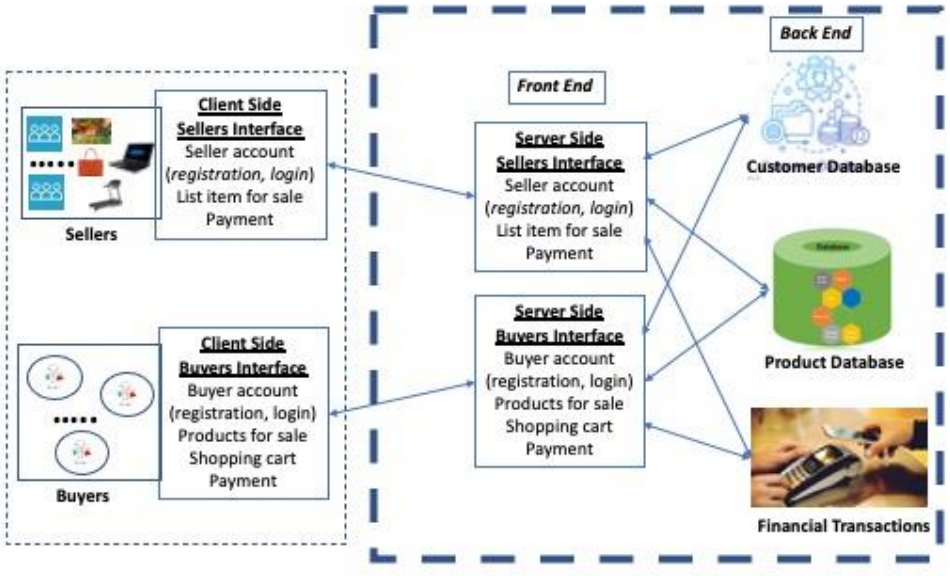
Programming Assignment One

Due Date and Time: 11:59 PM, Wednesday, February 08, 2023

Goal: The goal of this programming assignment is to review client-server programming using TCP/IP (using socket interface) and develop a system that you will enhance over the next 2-3 programming assignments.

You may work in teams of size two students.

An online marketplace is an e-commerce site that brings sellers and buyers together in one place. It allows sellers to put items for sale and interested buyers to purchase those items. We outlined the design of this system in class, comprised of seven logical components (See LectureSetZero - Intro.ppt): Client-side sellers interface, Client-side buyers interface, Server-side sellers interface, Server-side buyers interface, Customer database, Product database and Financial transactions. Over the first few programming assignments, you will implement and enhance this online marketplace.


Back End
Front End
Client Side Sellers Interface
Seller account (registration, login)
List item for sale
Payment
Sellers
Server Side Sellers Interface
Seller account (registration, login)
List item for sale
Payment
Customer Database
Product Database
Client Side Buyers Interface
Buyer account (registration, login)
Products for sale
Shopping cart
Payment
Buyers
Server Side Buyers Interface
Buyer account (registration, login)
Products for sale
Shopping cart
Payment
Financial Transactions

Characteristics of an item put up for sale

Item name: a char string of up to 32 characters, assigned by the seller (item names may not be unique)
Item category: an integer (0 – 9), assigned by the seller
Item Id: <item category, integer>: unique id assigned by the server
Keywords: up to five keywords, assigned by the seller, each keyword is a string of up to 8 characters
Condition: New or Used, assigned by the seller
Sale price: decimal number, assigned by the seller

CSCI/ECEN 5673: Distributed Systems
Spring 2023

Seller characteristics

Seller name: a char string of up to 32 characters, provided by the seller during account creation (Seller names may not be unique)
Seller id: an integer, a unique id provided by the server during account creation
Seller feedback: <# of thumbs up, # of thumbs down>, maintained by the server
Number of items sold: an integer maintained by the server

Buyer characteristics

Buyer name: a char string of up to 32 characters, provided by the seller during account creation (Buyer names may not be unique)
Buyer id: an integer, a unique id provided by the server during account creation
Number of items purchased: an integer maintained by the server

APIs of the logical components

Client-side sellers interface
Create an account: *sets up username and password*
Login: *provide username and password*
Logout
Get seller rating
Put an item for sale: *provide all item characteristics and quantity*
Change the sale price of an item: *provide item id and new sale price*
Remove an item from sale: *provide item id and quantity*
Display items currently on sale put up by this seller

Server-side sellers interface
Same as Client-side Sellers interface

Client-side buyers interface
Create an account: *sets up username and password*
Login: *provide username and password*
Logout
Search items for sale: *provide an item category and up to five keywords*
Add item to the shopping cart: *provide item id and quantity*
Remove item from the shopping cart: *provide item id and quantity*
Clear the shopping cart
Display shopping cart
Make purchase
Provide feedback: *thumbs up or down for each item purchased, at most one feedback per purchased item*
Get seller rating: *provide seller id*
Get buyer purchase history

Server-side buyers interface
Same as Client-side buyers interface

You can design the APIs of the three backend components as you see fit.

CSCI/ECEN 5673: Distributed Systems
Spring 2023

<u>Requirements of programming assignment one</u>

For this assignment, implement six components: Client Side Buyers interface, Client Side Sellers interface, Server Side Buyers interface, Server Side Sellers interface, Product Database and Customer Database. Your implementation must allow for each of these components to run on different hosts (different addresses IP addresses). You may keep the entire product and customer databases in memory, i.e. no secondary storage. Implement all APIs except "Make Purchase".

Design your own (reasonable) semantics for the search function in terms of "best" keyword match, etc. Clearly state your semantics.

Use TCP for interprocess communication. For this assignment, assume that each buyer or seller maintains a separate TCP connection, i.e. each TCP connection from client indicates a new buyer or seller. However, a buyer or a seller may connect to the server simultaneously from multiple hoosts.

YOU ARE REQUIRED TO USE SOCKET-BASED TCP/IP API FOR IMPLEMENTING INTERPROCESS COMMUNICATION. DO NOT USE REST, RPC OR ANY OTHER MIDDLEWARES.

Stateless frontend

Design your frontend server components to be stateless. Keep any state you need in the backend databases (customer and product databases).

Registration and login

Implement a very simple mechanism, e.g. store/transport login name and password in clear text. You will address security issues in a later assignment. Allow a buyer or server to login and interact with the server from multiple client machines simultaneously.

(<u>NOTE</u>: You will possibly extend/modify these APIs in future assignments, so make sure that your implementation can be easily extended in future)

<u>Evaluation</u>

Average response time: Response time of a client-side API function is the duration of time between the time when the client invokes the function and time when a response is received from the server. To measure average response time, measure the response time for ten different runs and then take the average.

Average server throughput: Throughput is the number of client operations completed at the server per second. To measure average throughput, measure the throughput for ten different runs and then take the average. Each run consists of each client invoking 1000 API functions.

Run one instance of each of the four server components as separate processes on possibly different hosts and possibly on cloud. Report the average response time and average throughput for the following scenarios:

Scenario 1: Run one instance of seller and one instance of buyer.
Scenario 2: Run ten instances of buyers and ten instances of sellers concurrently.
Scenario 3: Run 100 instances of buyers and 100 instances of sellers concurrently.

Provide explanations/insights for the performance you observe for each of the three scenarios. Provide an explanation for the differences in the performances you observe among the three scenarios.

CSCI/ECEN 5673: Distributed Systems
Spring 2023

What to submit

Submit a single zipfile that contains all source code files, makefiles, a README file and a performance report file. In the README file, provide a brief description of your system design along with any assumptions (8-10 lines) and current state of your system (what works and what not). In the performance report file, report all performances measured along with your explanation.