CSCI/ECEN 5673: Distributed Systems
Spring 2022

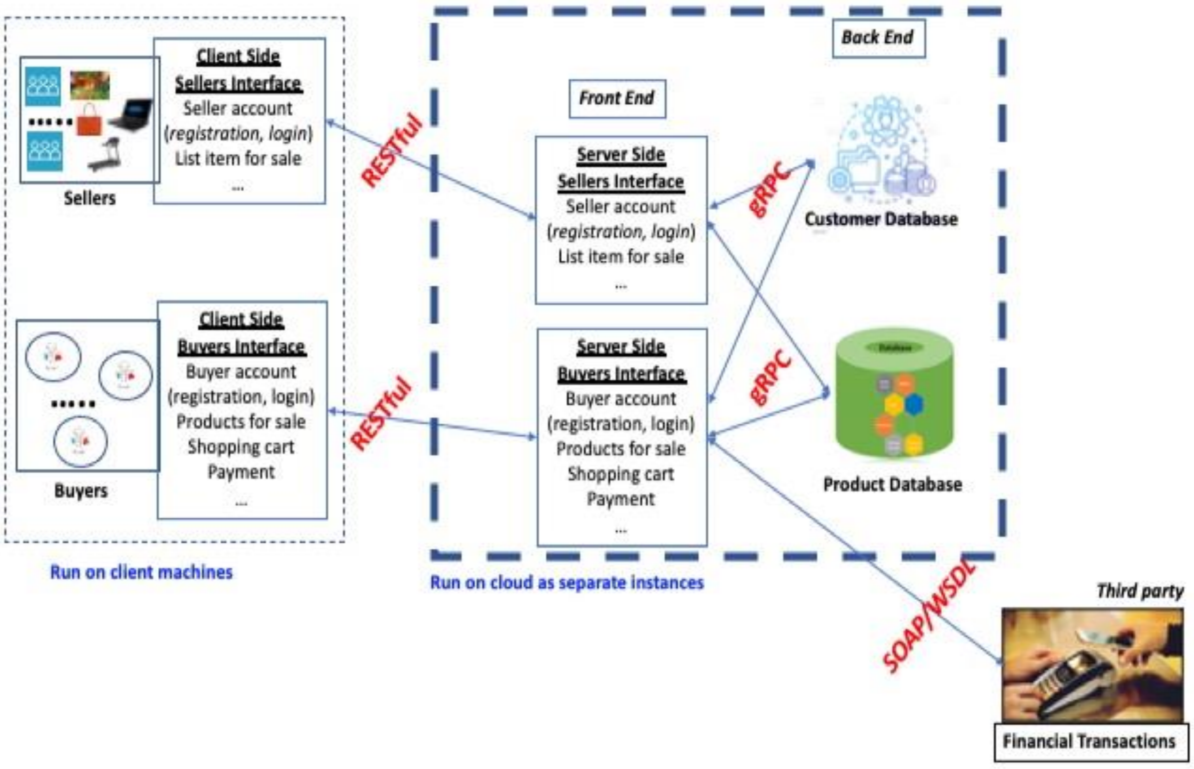
Programming Assignment Two

Due Date and Time: 11:59 PM, Monday, February 21, 2022

Goal: The goal of this programming assignment is to extend the online marketplace system you developed in Programming Assignment One as follows:

- Move all server components (both frontend and backend) to the cloud and run them in separate instances.
- Implement all remaining API functions
- Implement the customer database
- Make the client-server interactions RESTful (Client Side Seller interface–Server Side Seller interface and Client Side Buyer interface–Server Side Buyer interface)
- Implement all communication between front-end and customer/products databases using gRPC
- Implement a simple prototype of Financial Transactions (to be used as a third-party service) using SOAP/WSDL
- Note: keep the front-end stateless in your design

You may work in teams of size two students



Characteristics of an item put up for sale (same as programming assignment one)

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

Seller characteristics (same as programming assignment one)

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 (same as programming assignment one)

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 (same as programming assignment one)

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: *credit card details (name, number, expiration date)*
Provide feedback: *thumbs up or down for each item purchased, at most one feedback per purchased item*
Get seller rating: *provide buyer id*
Get buyer history

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

CSCI/ECEN 5673: Distributed Systems
Spring 2022

Moving all server components to cloud

You will run all server components as separate instances in cloud. Each student will get $50 Google Cloud Platform credit via the Google Cloud Platform Education Grants program. You will get instructions about this via email.

Financial Transactions

Implement a very simple prototype of this component. It receives a request (user name, credit card number) and returns Yes (95% probability) or No (5% probability). Use SOAP/WSDL to implement this service.

Stateless frontend

Design your frontend components stateless. Keep any state you need in the backend databases (customer and products 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.

Requirements of programming assignment two

Evaluation

Run your server components as separate instances on cloud. Measure the average response time of each client function.

What to submit

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