Bangladesh University of Engineering and Technology
Department of Computer Science and Engineering
CSE308: Software Engineering Sessional
July 2023 Semester

Assignment 4: Behavioral Design Patterns

Deadline: January 27, 2024, 11:55 PM

Problem Description [20 marks]

You have to build a stock trading platform where each user can subscribe to the different kinds of stocks in the market.
Consider an input file containing the stock names, followed by their counts and prices.

init_stocks.txt

```
P1 3 40.00
P2 4 30.00
P3 5 80.00
P4 6 25.00
P5 7 15.00
P6 9 50.00
```

After a user logs into the system, he will be shown the number of stocks and their respective prices.
A user can:

1. **Subscribe:** Subscribe to a particular stock by using the S command (e.g., 'S P3').
2. **Unsubscribe:** Unsubscribe from a subscribed stock using the U command (e.g. 'U P3')
3. **View:** View all his/her subscribed stocks using V command (e.g. 'V').

There is also a system administrator who updates the status of the stocks. S/he can:

1. **Increase price**: Increase stock price. Example: 'I P1 10.00'
2. **Decrease price**: Decrease a stock price. Example: 'D P4 5.00'
3. **Change in count**: Change in stock count (always **positive**). Example: 'C P2 2`

Make sure to properly maintain the states (counts and price of stocks).
Subscribers will be notified if the state of their subscribed stocks has changed. You have to show the notifications in **real time** [if he is logged in] or when the user logs in [if he is currently not logged in].

Now, implement the above scenario by writing the necessary classes and using an ***appropriate design pattern***.

You have to implement **client-server** architecture to solve the problem. **Multiple** clients can log in *simultaneously*. You may need to use *threading* and *socket-programming*.

Special Instructions

- Please first make a UML diagram of the classes.
- You are encouraged to discuss the design with your peers.
- You should implement the code in Java.
- Don't copy solutions from anywhere.

Submission Guidelines

- Create a folder that is named after your 7-digit student ID.
- Place all the essential files (with .java extension) inside the folder, and then zip that folder.
- Submit the zipped file in Moodle.