High Level Solution/Design

This application is developed to create a Stock Simulator which can be subscribed by subscribers and get notified every second with new stock prices.

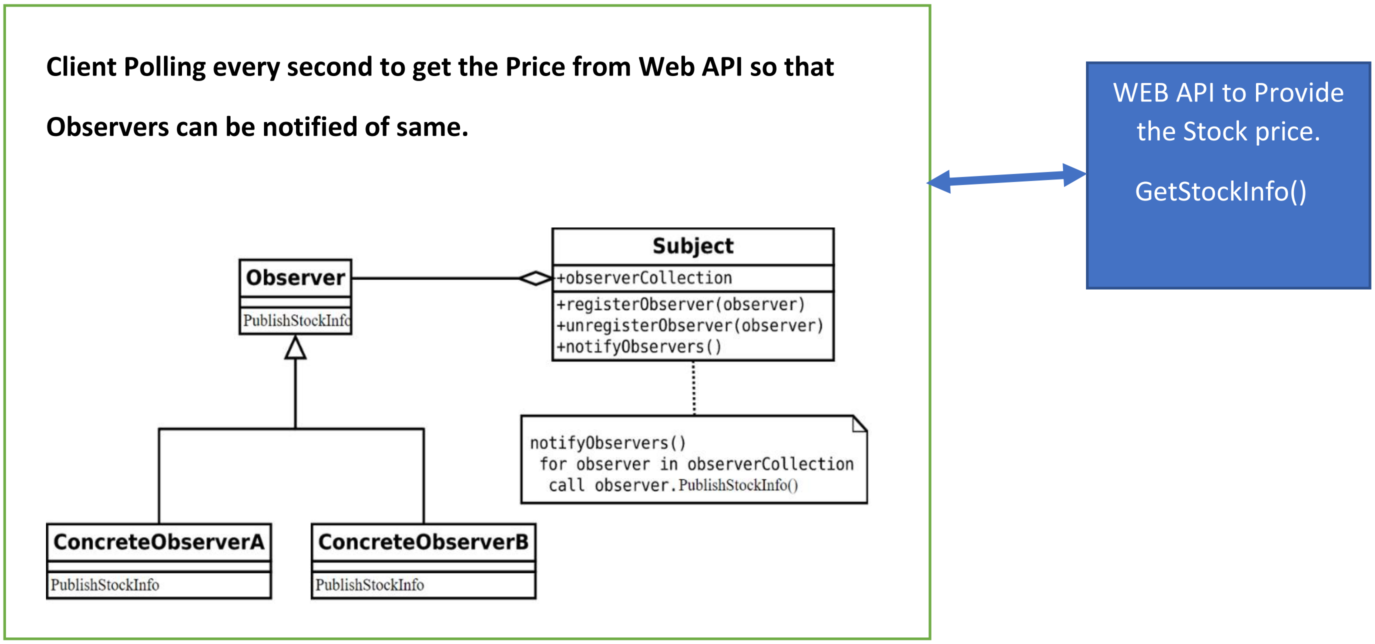
To Achieve this, I have Designed the application in two modules.

1)**Server** - A web-api which will provide new random Stock Prices.

2)**Client** - A console-based application which will be using service of API and will subscribe to the Stock prices.

I have used the behavioral design pattern i.e. "observer pattern" to achieve this functionality of notifying to the Subscribers.

Observer pattern seemed the best choice to meet this requirement of subscribing/unsubscribing the messages or notifications.



Client Polling every second to get the Price from Web API so that

Observers can be notified of same.

WEB API to Provide the Stock price.

GetStockInfo()

I have developed a Subject Class and an Observer class and as the name suggests Subject class is the one in which observer class is interested or say they will be subscribing to the events (new stock prices) happening at Subject class.

Subject holds the list of Observers. The Subscribe/Unsubscribe method are the ones through which the Observers register-unregister themselves to the Subject.

The Notify method is the one which has the responsibility of notifying all the Observers.

To get this running Please open the solution in Visual Studio (I have used VS 2017) and run a Instance of **PricePUB (WebAPI)** in browser and then Open an Instance of **ClassLib** (Console application).