CSE 460

Programming Assignment Report

Posting ID:5574-632

November 26th

Fall 2022

1. Description and Assumptions

Producer Buyer Software system is a simple system that links the Producers (Vehicles Manufactures) with Buyers (Rental Companies). It basically helps the Producers publish their product and then notify the Buyers who are subscribed in the same published category. It uses a special software engineering design pattern called Broker architecture, responsible basically for controlling the flow of the system and separate each class from another to avoid any direct link and to make the whole system more secure and to hide and encapsulate classes.

Assumptions:

- All vehicles are having one of the given three categories: CAR, TRUCK, SPORT UTILITY.
- All vehicles are having one of the given three fuel types: HYBRID, ELECTRIC, GASOLINE.
- The above names are not case sensitive ("Hybrid" same as "hyBrid")
- No User Interface is provided.
- The allowed commands are only: "subscribe", "unsubscribe", "publish"
- The commands are case sensitive (only small is acceptable).
- All producer names are acceptable (even if it is not realistic).

2. Use Case Diagrams

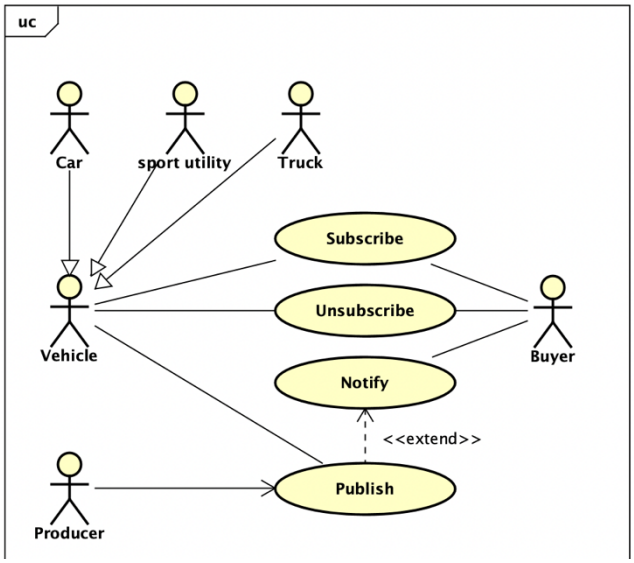

Figure 1. Use Case Diagram

Description:

- Vehicle; Truck, Car, Sport Utility: The vehicle that would be published to the subscribers.
  Truck and car and sport utility is subcategory of vehicle.
- Producer: The vehicle manufactures that would publish the vehicle.
- Buyer: The rental companies that would subscribe to a specific vehicle category.
- Publish: The command to let the Producer publish their vehicle.
- Subscribe: The command to subscribe to a specific vehicle category.
- Unsubscribe: The command to unsubscribe to a specific vehicle category.
- Notify: The command to notify the buyers about the published categories.

Scenarios:

1- The producers publish a vehicle that buyer is not subscribed on:
   a. Publish.
2- The producers publish a vehicle that buyer is subscribed on:
   a. Publish → Publish extended Notify and then notify buyers about this vehicle.

3. Class Diagrams

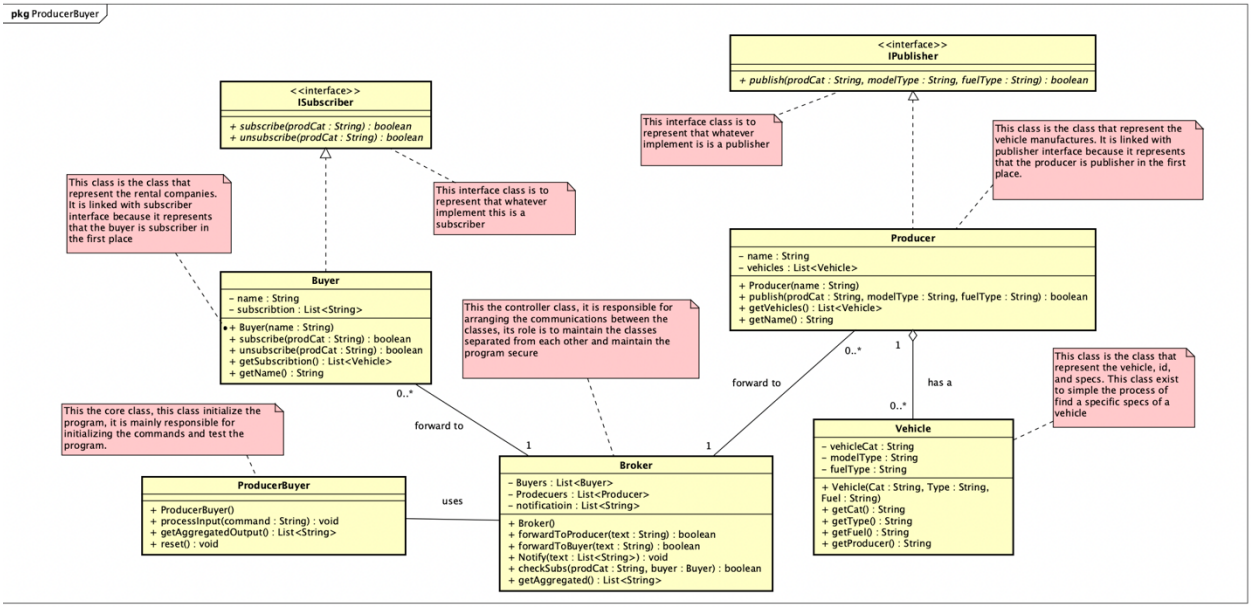

Figure 2. Class Diagram

4. Sequence Diagrams

1. Producer Buyer:

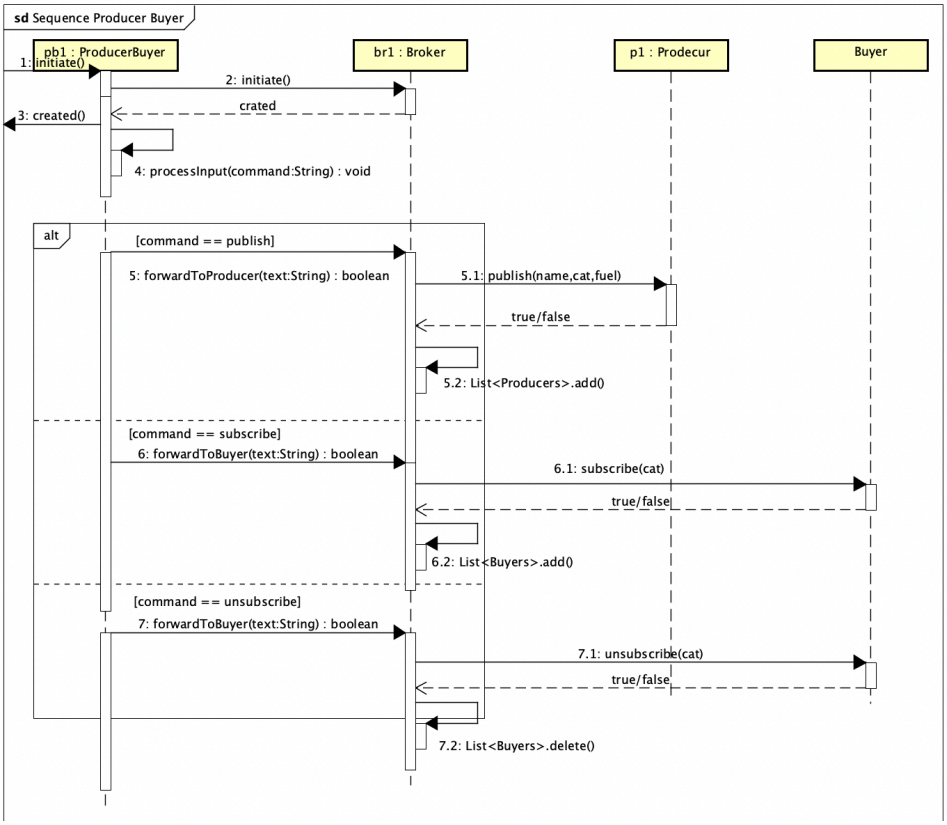

Figure 3. Sequence Diagram

a. This the main sequence diagram, it begins with the query of the ProducerBuyer class, then it goes to the next class “Broker: which is responsible of controlling the flow, then depending on the input it forward the text to Buyer or Producer.

2. Buyer

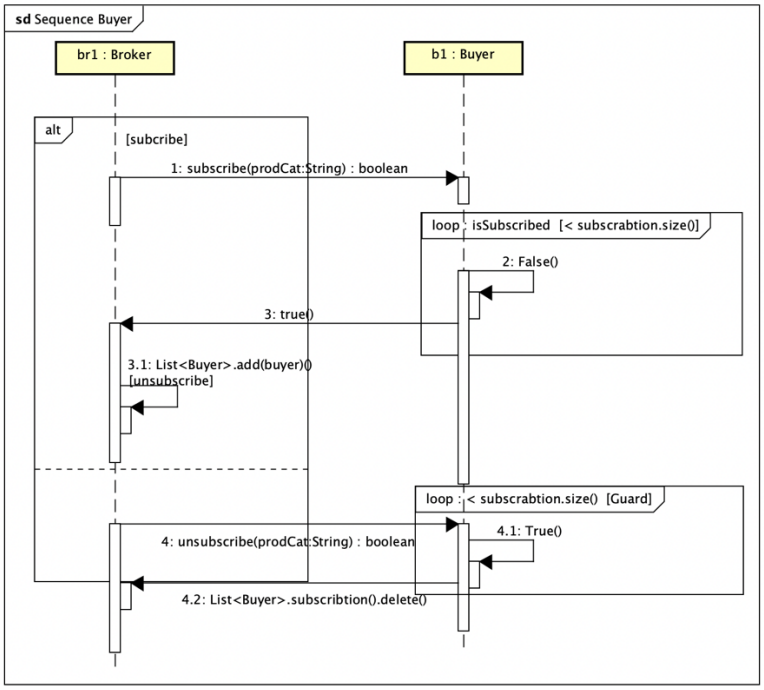

Figure 4. Sequence Diagram

a. This is a completion of the previous sequence diagram, after broker forwarding the text to the Buyer, the buyer take control and check the input, then it creates an instance and return the control to the broker again.

3. Producer

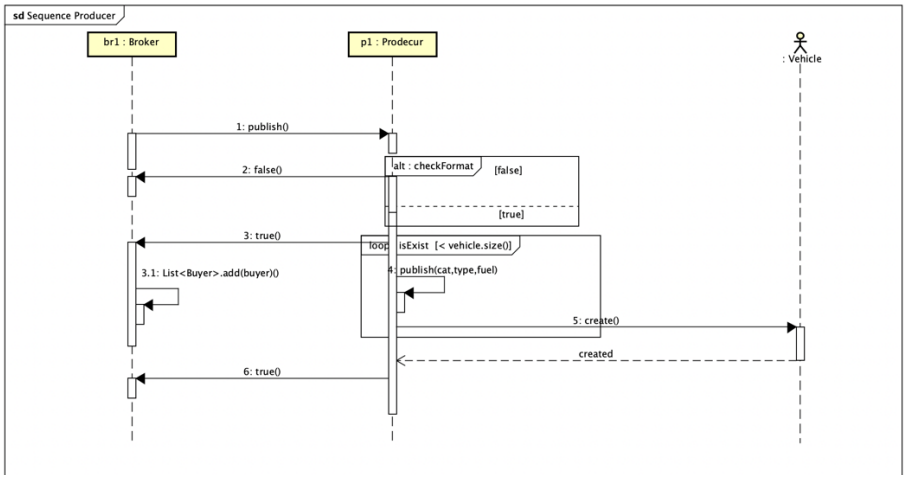

Figure 5. Sequence Diagram

a. This is a completion of the previous sequence diagram “PruducerBuyer”, after broker forwarding the text to the producer, the producer take control and check for the text format, then create an Vehicle instance and finally return the control to the broker.

5. State Machine Diagrams

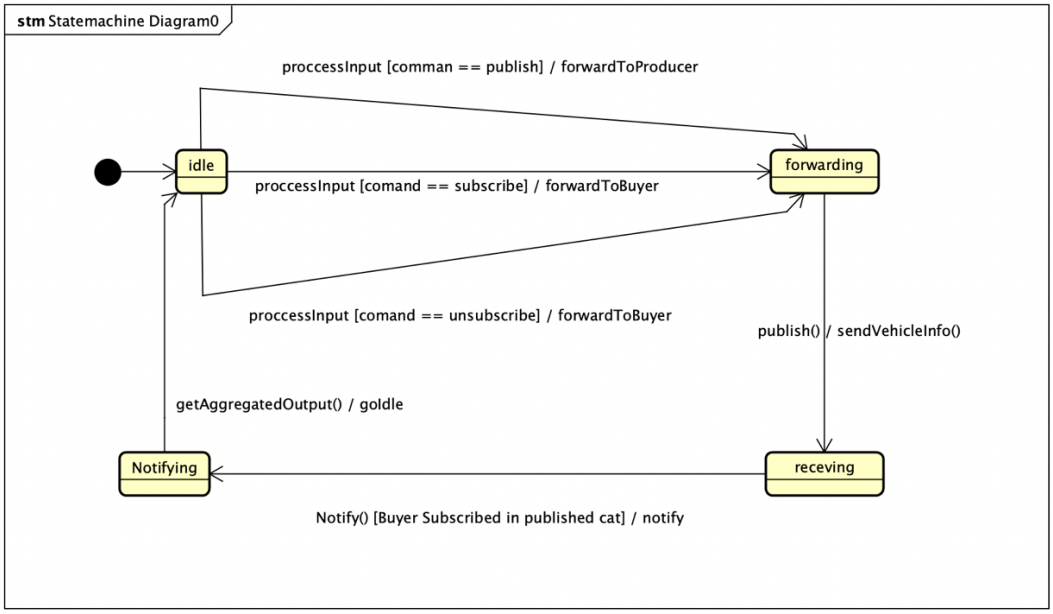

Figure 6. State Machine Diagram

The Broker is idle initially, when it is triggered, it goes to forwarding mode, then when a producer publishes a category that another buyer subscribed on, the Broker goes to receiving mode and finally notifying the buyers when getAggregatedOutput triggered, after that it comes back to idle until another proccessInput command called.

6. Appendix & Credits

- Inspiration of broker State Machine Diagram. https://www.researchgate.net/figure/Hierarchy-of-the-interactions-among-nodes_fig2_337043741.

- To remove the leading and trailing spaces. https://www.geeksforgeeks.org/how-to-remove-all-white-spaces-from-a-string-in-java/