## **FeedMe Software Engineer Take Home Assignment**

Below is a take home assignment before the interview of the position. You are required to

1. Understand the situation and use case. You may contact the interviewer for further clarification.
2. Implement the requirement with your most effective tools.
3. Complete the requirement and perform your own testing.
4. Provide documentation for the any part that you think is needed.
5. Commit into your own github and share your repo with the interviewer.
6. Bring the source code and functioning prototype to the interview session.

### **Situation**

McDonald is transforming their business during COVID-19. They wish to build the automated cooking bots to reduce workforce and increase their efficiency. As one of the software engineer in the project. You task is to create an order controller which handle the order control flow.

### **User Story**

As below is part of the user story:

* As McDonald's customer, after I submit my order, I wish to see my order flow into "PENDING" area and moved to "COMPLETE" area after the cooking bot has completed my order.
* As McDonald's VIP member, after I submit my order, I want my order being process by cooking bot first before all normal order. If there's another VIP order came here, my order should queue behind his/her order.
* As McDonald's manager, I should able to increase or decrease number of cooking bot available in my restaurant. When I add a bot, it should start working immediately on the order. When I remove a bot, the processing dish should remain un-process.
* As McDonald bot, it can only pickup and process 1 order at a time, each cooking bot need 10 seconds to process an order.

### **Requirements**

* When "New Normal Order" clicked, a new order should show up Pending Area.
* When "New VIP Order" clicked, a new order should show up in Pending Area. It should place in-front of all existing "Normal" order but behind of all existing "VIP" order.
* The Order number should be unique and increasing.
* When "+ Bot" clicked, a bot should be created and process the order inside "PENDING", after 10 seconds, the order should move to "COMPLETE". Then the bot should start processing another order if there is any left.
* If there is no more order in the "PENDING", the bot should do nothing until a new order come in.
* When "- Bot" clicked, the newest bot should be destroyed. If the bot is processing some order, the process should be abandon. The order now become "PENDING" and ready to process by other bot.
* No data persistance is needed for this prototype, you may perform all the process inside memory.

### **Functioning Prototype**

You may demostrate your final funtioning prototype with ****one (only)**** of the following method:

* CLI application
* UI application
* E2E test case

### **Tips on completing this task**

* Testing, testing and testing. Make sure the prototype is functioning and meeting all the requirements.
* Do not over engineering. Try to scope your working hour within 3 hours (1 hour per day). You may document all the optimization or technology concern that you think good to bring in the solution.
* Complete the implementation as clean as possible, clean code is a strong plus point, do not bring in all the fancy tech stuff.