

Hi and thank you for inviting me to this challenge.

It was very nice and interesting to work on this assessment,

I had some consideration during my implementation that I would like to mention here:

1. First, I did not follow the TDD in this case since It is a bit new to me and I'm still not 100% sure about it but I would like to get a chance to work on it and I'm making some improvement in that area.
2. I wrote only one unit test. It is because the only testable operation in this case is to see if an event gets raised or not. And I thought that might be sufficient as a proof on concept to be able to test a code that raises an event and since this one is not a production project; I just skipped the other test.
3. At some point some of those operations sounded a bit categorized to me but then again, they can be all encapsulated into a module responsible for membership for instance one for external communication and the other one for packaging, but I also considered them some obvious matters that can be skipped.
4. There are some other ways of handling this challenge by using some sort of polymorphism for all different types of payments, but I decided to use event-based decoupling in this case which I believe can bring more value and it might have more of a microservice look that I prefer a lot over monolithic.

At the end, I hope you will find my solution way off from your expectation 😊 and I'll be waiting to hear about your feedback.