Home Assignment:
Senior Backend Engineer Position

Medication Periods

Medications are adminstered to patients during their hospital stay, at different times. Medication period starts when a patient starts receiving a medication and ends when the patient stops receiving that medication. There could be several periods of the same medication, as well as overlapping periods of different medications.

Design and implement two services: publisher and consumer. The services communicate through rabbitMQ. Services should be written in **async python code**. Each service should include a Dockerfile and docker-compose file.

Publisher service:
1) Reads medication administation events from input source of your choice: API / file / database / message queue / etc
2) Publishes medication administration events to a rabbitMQ exchange

Consumer service:
1) Consumes medication administration events
2) Saves events to a database of your choice
3) Provides API to get a list of medication periods for a patient

Medication Administration events

Each event includes:
- Unique patient identifier
- Medication name
- Action: start / stop / cancel_start / cancel_stop
- Event Time

"start" means medication period started. "stop" means medication period ended. "cancel_start" means a "start" was reported by mistake and should be ignored when calculating medication periods. Same for "cancel_stop". Event Time represents events order. "stop" or "cancel" events may be published and consumed before a "start" event for the same medication. When calculating medication periods, always use Event Time.

Your code will be executed and evaluated according to the following criteria:
1) Working code
2) Code readability, code structure
3) Unit tests
4) Handling special cases

Special cases may include:
- Patient did not receive any medication
- Several events from the same kind
- "start" event and "stop" event at the exact same time
- Medications with "stop" event but no "start" event, and vice-versa
- Invalid input, for example - wrong date format

Code should be committed and pushed into a github repository, send us the URL.
Repo should include a short README file that coveres –
1) Starting the services
2) Expected input for the publisher service
3) Sample query for the consumer service

Your code will be reviewed and executed on our Kubernetes environment.

Good Luck!