HumanITcare

Hi! If you've reached this point in the process, there is a match between us!

Now, in order to see that your skills are on par with what we want for our team, we have prepared this test for you! Do your best, and try to argue about all the decisions you've made in the process. We expect to receive a document (word/ zip/ file or others) by answering to this email: hiring@humanitcare.com. You have 5 days to answer these tasks.

Thank you very much and good luck!

Showoff

Show us the piece of code you are most proud of! Then explain to us why it is your favorite piece of code.

Coding tasks

Define zeroes

Define zeroes(n) as the number of zeroes in the decimal expansion of the integer n. A number n is zero_special if zeroes(n) > zeroes(n-1). Write a function that determines whether n is zero_special.

Managing observations

In this task, you will work with core concepts to our business. *Observations* are a central element in healthcare, used to support diagnosis, monitor progress, determine baselines and patterns. Uses for the Observation resource include:

- Vital signs such as body weight, blood pressure, and temperature
- Activity Data like step count

Observations with multiple values like blood pressure (systolic and diastolic values), are stored in what are known as *Observation Components*.

A *Monitored* is a user that is being followed-up by a healthcare professional. The Monitored registers Observations with their app so that the healthcare professional can see them remotely.

This coding exercise consists in implementing a little backend micro-service which aims to serve the Observation data. You can use any framework, but it must be a Python framework!

This component will expose a REST API that will allow to:

- Introduce Observations into the system
- For any given Monitored:
    - Fetch all Observations given an observation name
    - Fetch the latest Observation, for a specific type of Observation.
- Compute and fetch the mean of Observations values (or component's values) for a given `observation_name`.

To help you build the data models, here you have a couple of examples of Observations:

<table>
<tr><th>Temperature Observation</th><th>Blood Pressure Observation</th></tr>
<tr><td><pre>
{
    "monitored_id": 123,
    "observation_name": "temperature",
    "issued": "2022-02-21T08:20:01.52+01:00",
    "value": "36.0",
    "value_type": "float",
    "value_units": "celsius"
  }
</pre></td><td><pre>
{
   "monitored_id": 456,
   "observation_name": "blood_pressure",
   "issued": "2022-02-21T08:20:01.52+01:00",
   "components": [
     {
       "observation_name": "systolic",
       "value": "106",
       "value_type": "int",
       "value_unit": "mmHg"
     },
     {
       "observation_name": "diastolic",
       "value": "60",
       "value_type": "int",
       "value_unit": "mmHg"
     }
   ]
 }
</pre></td></tr>
</table>

The application has to be configured to be run in an easy to test environment (Docker, pipenv, Poetry, etc.).

Open questions

- In your opinion, what are the characteristics of high quality code?
- Can you share your thoughts and experiences regarding Domain-Driven Design (DDD)? Have you ever applied DDD in a real-world project, and if so, what challenges or drawbacks did you encounter during its implementation?