VIRTUALHEALTH®

# VirtualHealth

# Sr. Software Engineer Assessment

## General information

- We are not going to run your code so do not spend your time to resolve runtime issues.
- We will extrapolate your solution of this task to your approach to work on daily tasks.
- Any database changes (indexes, column alterations) are allowed and welcomed.

## Preconditions

- Part of database schema and mock data are provided in the attached SQL file. It contains:
    - Table `patient` with **MEDREC_ID** and **PATIENT_NAME** fields
    - Table `disease` with **MEDREC_ID** and **ICD** fields
    - Table `medication` with **MEDREC_ID** and **NDC**
- **MEDREC_ID** is the unique patient number.
- **ICD** is the diagnosis code (International Classification of Diseases).
- **PATIENT_NAME** is the full name of patient.
- **NCD** is the code of prescribed medication (National drug code).
- One patient may have multiple diagnoses.
- One patient may have multiple medications.
- The same medication may be prescribed several times for the same patient.

## Main task

Create a REST API endpoint containing results of the following query:

*All patients who have the same medication (NDC) prescribed more than* ***N*** *times.*

Value of **N** variable should be taken from URL parameter. Example of URL is http://localhost/query/{N} or http://localhost/query?n={N}.

REST API should return json with collection of objects containing the following information:

- **ID** *(mandatory)* – patient's MEDREC_ID
- **Name** *(mandatory)* – patient's full name
- **Medications count** *(optional)* – count of all patient's medications

For your convenience, you can limit number of returned patients (but there should be not less than 30 patients).

## Requirements

Please, implement your code using either option #1 or option #2.

### Option #1:

- Code should be implemented using Go programming language.
- Solution should be based on DAO and DTO design patterns.
- Do not implement unused methods and other unused code.
- Incapsulate patients search in a separate method.
- Usage of any framework or additional packages is acceptable.
- If you don't use code for DB schema changes, please, send a description of DB alterations made.
- Git repository should be initialized inside project and contain progress of code implementation. **Please, do not store your code in a public git repository!** ***Send us your code in zip or tar archive.***
- If some of requirements listed above are not feasible in your solution, please, describe the reason.

### Option #2:

- Code should be implemented using PHP programming language (version 7 or 8).
- Solution should be based on DAO and DTO design patterns.
- Use dependency injection. You may use DI container or initialize all dependencies manually in controller.
- Do not implement unused methods inside your classes.
- Do not use ORM, use DAO + DTO instead. But you may use Doctrine DBAL, PDO or another library for accessing database.
- Do not use Factory or Singleton for anything except initialization of services inside DI container.
- Incapsulate patients search in a separate method.
- It's preferrable to perform all database schema changes via migrations. If you don't use migrations, please, send a description of DB alterations made.
- Usage of any PHP framework is acceptable, but not required.

- Git repository should be initialized inside project and contain progress of code implementation. **Please, do not store your code in a public git repository!** ***Send us your code in zip or tar archive.***
- If some of requirements listed above are not feasible in your solution, please, describe the reason.