.NET Programming Trial Task

Trial task for .NET programming developer

Please note: Please do as many requirements as you can, better deliver less but show best! The following task is part of a recruitment and selection process only. Doing the exercise is not treated as working for Fujitsu and will not be paid for

The objective of the trial task is to assess .NET programming skills, selecting appropriate tools for completing the task, OOP skills, and application design skills, including object model design, working with API-s, and documenting the code.

How to do the task

The trial task's outcome must be submitted to Fujitsu before the technical interview. You will be asked to do a walk-through and describe your solution at the interview.

What we will look at:

- The OOP design of the solution, layering, clean code, common patterns, etc
- That code is human readable
- Public methods should be documented
- Error handling
- Test coverage
- Where code is stored and how work is organized

Technologies to use

- .NET
- C#
- Entity framework
- Freely chosen database

Description of the task

The objective of the task is to develop a sub-functionality of the food delivery application, which calculates the delivery fee for food couriers based on regional base fee, vehicle type, and weather conditions.

Core modules to implement:

• Database for storing and manipulating data

• Configurable scheduled task for importing weather data (CronJob)

• Functionality to calculate delivery fee

• REST interface, which enables to request of the delivery fee according to input parameters

Database

There has to be at least 1 table in the database for weather data where the following business information on weather conditions is stored:

• Name of the station

• WMO code of the station

• Air temperatuure

• Wind speed

• Weather phenomenon

• Timestamp of the observations

Additional tables could be created in the database if it's considered necessary to store classifications, parameters for fee calculation, etc.

CronJob for importing weather data

CronJob must be implemented to the code to request weather data from the weather portal of the Estonian Environment Agency. The frequency of the cronjob has to be configurable. The default configuration to run the CronJob is once every hour, 15 minutes after a full hour (HH:15:00).

URL to get data:

https://www.ilmateenistus.ee/ilma_andmed/xml/observations.php

Additional information about the interface:

https://www.ilmateenistus.ee/teenused/ilmainfo/eesti-vaatlusandmed-xml

As a result of the request, the weather data described in the database chapter has to be inserted into the database for the following stations:

• Tallinn-Harku (city of Tallinn)

• Tartu-Tõravere (city of Tartu)

• Pärnu (city of Pärnu)

NB! The history of imported weather data has to be permanently stored. Therefore, new entries must be inserted into the database as a result of each importing process (not overwrite existing entries of the station).

Delivery fee calculation

A delivery fee has to be calculated according to input parameters from REST interface requests, weather data from the database, and business rules. The total delivery fee consists of a regional base fee for a specific vehicle types and extra fees for some weather conditions:

Business rules to calculate regional base fee (RBF):

- In case City = Tallinn and:
    - Vehicle type = Car, then RBF = 4 €
    - Vehicle type = Scooter, then RBF = 3,5 €
    - Vehicle type = Bike, then RBF = 3 €
- In case City = Tartu and:
    - Vehicle type = Car, then RBF = 3,5 €
    - Vehicle type = Scooter, then RBF = 3 €
    - Vehicle type = Bike, then RBF = 2,5 €
- In case City = Pärnu and:
    - Vehicle type = Car, then RBF = 3 €
    - Vehicle type = Scooter, then RBF = 2,5 €
    - Vehicle type = Bike, then RBF = 2 €

Business rules to calculate extra fees for weather conditions:

- Extra fee based on air temperature (ATEF) in a specific city is paid in case Vehicle type = Scooter or Bike and:
    - Air temperature is less than -10̊ C, then ATEF = 1 €
    - Air temperature is between -10̊ C and 0̊ C, then ATEF = 0,5 €

- Extra fee based on wind speed (WSEF) in a specific city is paid in case Vehicle type = Bike and:
    - Wind speed is between 10 m/s and 20 m/s, then WSEF = 0,5 €
    - In case of wind speed is greater than 20 m/s, then the error message “Usage of selected vehicle type is forbidden” has to be given

• Extra fee based on weather phenomenon (WPEF) in a specific city is paid in case Vehicle type = Scooter or Bike and:

- Weather phenomenon is related to snow or sleet, then WPEF = 1 €
- Weather phenomenon is related to rain, then WPEF = 0,5 €
- In case the weather phenomenon is glaze, hail, or thunder, then the error message “Usage of selected vehicle type is forbidden” has to be given

NB! Extra fees for weather conditions are paid only for conditions listed above. Calculations must base on the latest weather data for a specific City

Example calculation:

- Input parameters: TARTU and BIKE -> RBF = 2,5 €
- Latest weather data for Tartu (Tartu-Tõravere):
  - Air temperature = -2,1̂ C -> ATEF = 0,5 €
  - Wind speed = 4,7 m/s -> WSEF = 0 €
  - Weather phenomenon = Light snow shower -> WPEF = 1 €
- Total delivery fee = RBF + ATEF + WSEF + WPEF = 2,5 + 0,5 + 0 + 1 = 4 €

REST interface

REST interface (endpoint), which enables other parts of the application to request delivery fees according to the following input parameters:

• City: Tallinn / Tartu / Pärnu

• Vehicle type: Car / Scooter / Bike

In response to the request, the total delivery fee (calculated according to the description in the chapter “Delivery fee calculation”) or an error message must be given. REST interface must be documented.

Bonus

If you want to stand out:

• You can solve the delivery fee calculation and business rules related to the process in a way that business rules for base fees and extra fees could be managed (CRUD) through the REST interface.

• You can add additional datetime parameters to the REST interface request. This parameter should not be mandatory, but if it’s valued, delivery fee calculations have to be done based on business rules and weather conditions, which were valid at the specific time.

• Implement authentication to REST interface