Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**Лабораторна робота №1**

з дисципліни «Розробка та експлуатація систем Інтернет речей»

|  |  |
| --- | --- |
| Виконав:  студент групи ІА-32мп  Кац Е. О. |  |

Київ 2024

Репозиторій з програмою: https://github.com/KATSEDUARD/IoT\_Lab1

file\_datasource.py:

from csv import reader

from datetime import datetime

from domain.accelerometer import Accelerometer

from domain.gps import Gps

from domain.aggregated\_data import AggregatedData

from domain.parking import Parking

import config

class FileDatasource:

    def \_\_init\_\_(self, accelerometer\_filename: str, gps\_filename: str, parking\_filename: str) -> None:

        self.accelerometer\_filename = accelerometer\_filename

        self.gps\_filename = gps\_filename

        self.parking\_filename = parking\_filename

    def read(self) -> AggregatedData:

        """Метод повертає дані отримані з датчиків"""

        if self.\_position == len(self.\_data):

            self.\_position = 0

        data = self.\_data[self.\_position]

        data = AggregatedData(

            data.accelerometer,

            data.gps,

            data.parking,

            datetime.now(),

            config.USER\_ID

        )

        self.\_position += 1

        return data

    def startReading(self, \*args, \*\*kwargs):

        """Метод повинен викликатись перед початком читання даних"""

        self.\_position = 0

        self.\_data = []

        with open(self.accelerometer\_filename, "r") as accelerometer\_file:

            with open(self.gps\_filename, "r") as gps\_file:

                with open(self.parking\_filename, "r") as parking\_file:

                    accelerometer\_data\_reader = reader(accelerometer\_file)

                    gps\_data\_reader = reader(gps\_file)

                    parking\_data\_reader = reader(parking\_file)

                    next(accelerometer\_data\_reader)

                    next(gps\_data\_reader)

                    next(parking\_data\_reader)

                    for accelerometer\_row, gps\_row, parking\_row in zip(

                        accelerometer\_data\_reader, gps\_data\_reader, parking\_data\_reader

                    ):

                        if len(accelerometer\_row) == 0 or len(gps\_row) == 0 or len(parking\_row) == 0:

                            continue

                        x, y, z = map(int, accelerometer\_row)

                        longitude, latitude = map(float, gps\_row)

                        empty\_count = float(parking\_row[0])

                        p\_longitude, p\_latitude = map(float, parking\_row[1:3])

                        aggregated\_data = AggregatedData(

                            Accelerometer(x, y, z),

                            Gps(longitude, latitude),

                            Parking(empty\_count, Gps(p\_longitude, p\_latitude)),

                            datetime.now(),

                            config.USER\_ID

                        )

                        self.\_data.append(aggregated\_data)

    def stopReading(self, \*args, \*\*kwargs):

        """Метод повинен викликатись для закінчення читання даних"""

        if self.accelerometer\_file:

            self.accelerometer\_file.close()

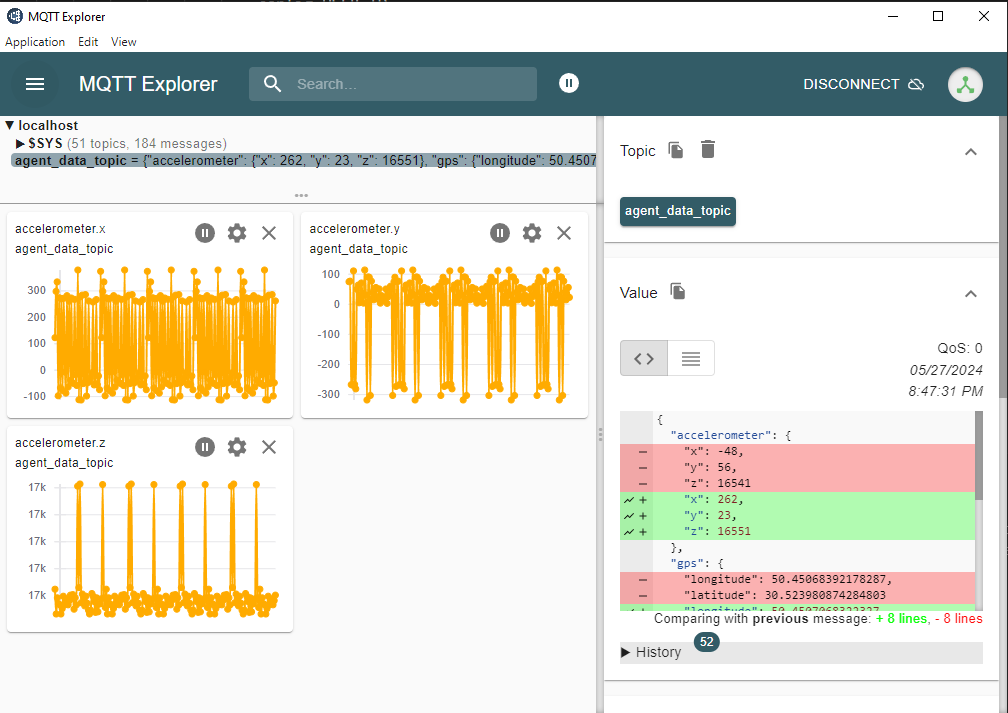
        if self.gps\_file:

            self.gps\_file.close()

        if self.parking\_file:

            self.parking\_file.close()

Результат у *MQTT Explorer*:



**Висновки**: Під час виконання лабораторної роботи я ознайомився із роботою з Docker та MQTT Explorer для реалізації проекту моніторингу та аналізу стану дорожнього покриття.