HW1: Mid-term assignment report

João Correia nº104360

April 11, 2023

Contents

1 Introduction **2**
1.1 Overview of the work . . . 2
1.2 Current Limitations . . . 2

2 Product Specifications **2**
2.1 Functional scope and supported interactions . . . 2
2.2 System Architecture . . . 2
2.3 API for developers . . . 2

3 Quality Assurance **3**
3.1 Overall strategy for testing . . . 3
3.2 Unit and Integration Testing . . . 3
3.3 Functional Testing . . . 3
3.4 Code Quality Analysis . . . 3

4 References & Resources **3**
4.1 References . . . 3
4.2 Resources . . . 3

1 Introduction

1.1 Overview of the work

This report presents the midterm individual project required for TQS, covering both the software product features and the adopted quality assurance strategy.

This Project consists of an application that provides details on air quality for a certain region/city using an external API (to get the real information) We needed to create a minimilist web page, an in-memory cache system and our own Rest API.

1.2 Current Limitations

Unfortunatly, not all suggested features were completely implemented, the main one being the possibility to get the air quality information from a specific day; it is only possible to get the data from the current date.

2 Product Specifications

2.1 Functional scope and supported interactions

2.2 System Architecture

Main Technologies used:

1. Spring Boot (for the API/Cache)
2. Redis (Cache System)
3. React (Web App)

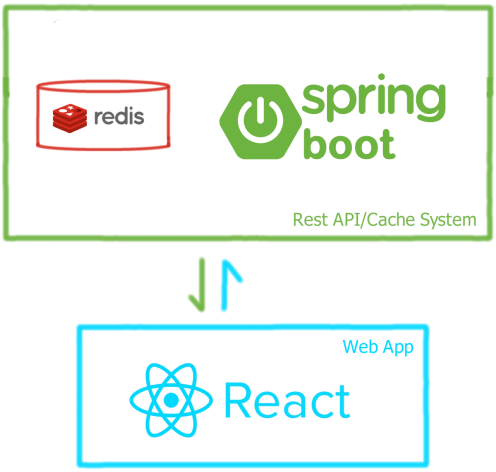

Figure 1: Technology Diagram

2.3 API for developers

We have 2 main endpoint types, the problem domain, and the cache usage.

Figure 2: Given the wanted coordinates, it will correspond with the air quality information in the moment

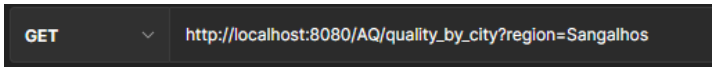

Figure 3: Given the wanted city/region, it will correspond with the air quality information in the moment

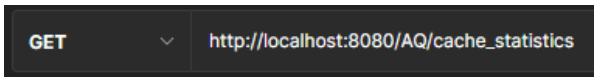

Figure 4: Will respond with the cache information (number of hits, total calls, and hit percentage)

3 Quality Assurance

3.1 Overall strategy for testing

Doing Test-driven development I ended up using most of the testing technologies studied in class.

3.2 Unit and Integration Testing

Created Basic tests to ensure that my services worked correctly

3.3 Functional Testing

Using Selenium I Tested the application as User would, they ended up being really simple given the simple nature of the web application

3.4 Code Quality Analysis

Using Sonarqube I was able to correct some bugs and code smells that would otherwise scape me.

4 References & Resources

4.1 References

Git Repository
Video Demo

4.2 Resources

API used: https://openweathermap.org/api/air-pollution