HW1: Mid-term assignment report

Ana Alexandra Antunes [876543], v2021-05-14

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.
The Web Application done for this report has the goal of getting Air Quality parameters from cities. It uses the openweathermap API to fetch the information for the user and caches it. This Web Application also has its own API where there can be made requests for cache information, which is not available in the web interface

1.2 Current limitations

The Application doesn't account for the unavailability of the external API used.

2 Product specification

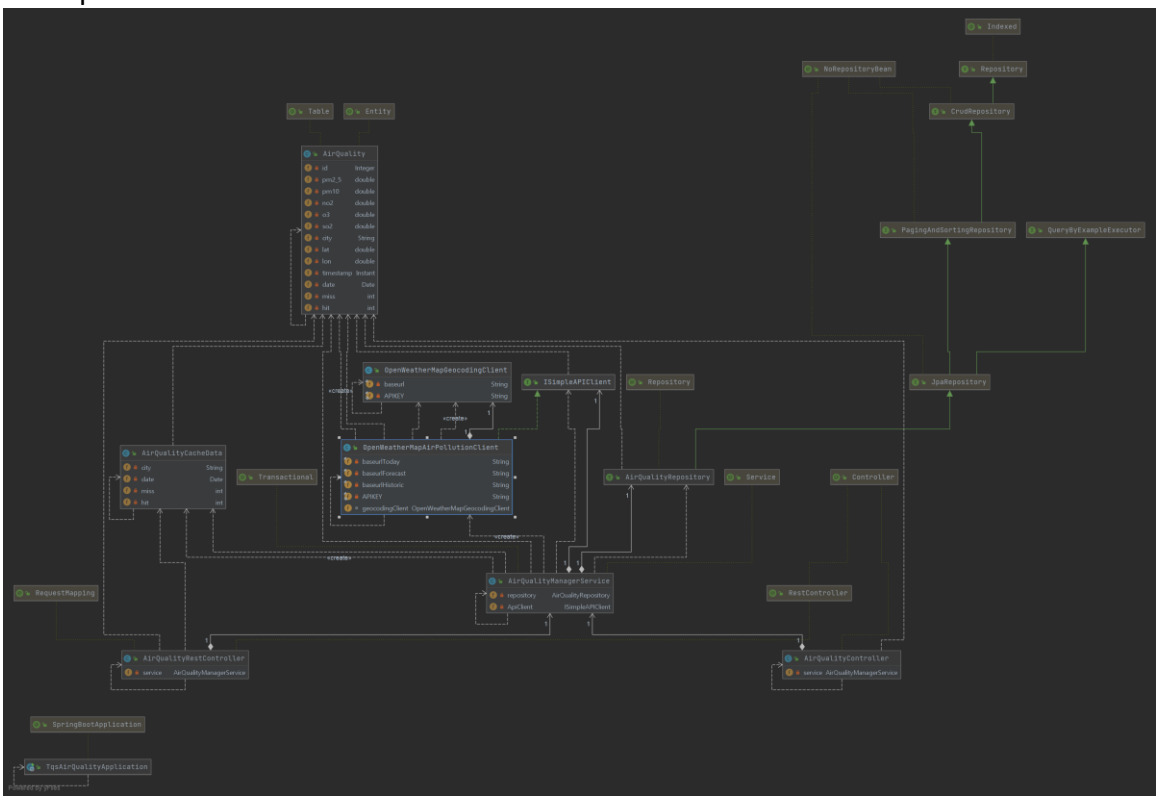
2.1 Functional scope and supported interactions

A user can interact with the Web Application through the web interface or through the API to get Air Quality information or cache information (for the API only).
Main Scenarios:

- A user wants to know the Air Quality in Aveiro today. He goes to this project's website and goes to the "Today" tab after that he can search for Aveiro and see Air Quality information for Aveiro.
- A developer who pretends to make an app that consumes a free API for Air Quality can use this project's API.

2.2 System architecture

The Application services where services/backed base on Spring Boot.
For the external API, I used the Air Pollution API and the Geocoding API from openweathermap.org
The Web Interface was implemented with HTML, JavaScript using thymeleaf templating.
The time to live policy used were 15 minutes.
The AirQualityManagerService class is a @Service SpringBoot class where all the logic is processed. It's in that class where it's decided if the API is called or not and if the values in the repo are valid or not.



2.3 API for developers

API endpoints:

-“api/airquality/today/{city}”:

-Returns the Air Quality for the defined “city”.

-“api/airquality/date/{city}/{date}”

-Returns the Air Quality for the defined “city” on a certain “date” (format: dd-mm-yyyy) .

- “api/airquality/historic/{city}/{datestart}/{dateend}”

-Returns a list of Air Quality for the defined “city” on a certain interval of dates (format: dd-mm-yyyy) including both dates.

-”api/cache”

-Returns a list of Air Quality Cache Date which contains all the hits and misses for a certain Air Quality.

3 Quality assurance

3.1 Overall strategy for testing

For the RestController, the GeocodingAPIClient and the AirPollutionAPIClient I used TDD, as I developed the tests first before the implementation was done. For the rest I used BDD, as the features became more complex and the tests couldn’t be pre-programmed as easily.

3.2 Unit and integration testing

All the classes created for this project were subject of unit testing.
In the controllers, the connections (connection package) and the repository, the unit tests focused on testing the functions.
In the service, as it was a more complex class it had to test more behaviors with multiple tests for each function

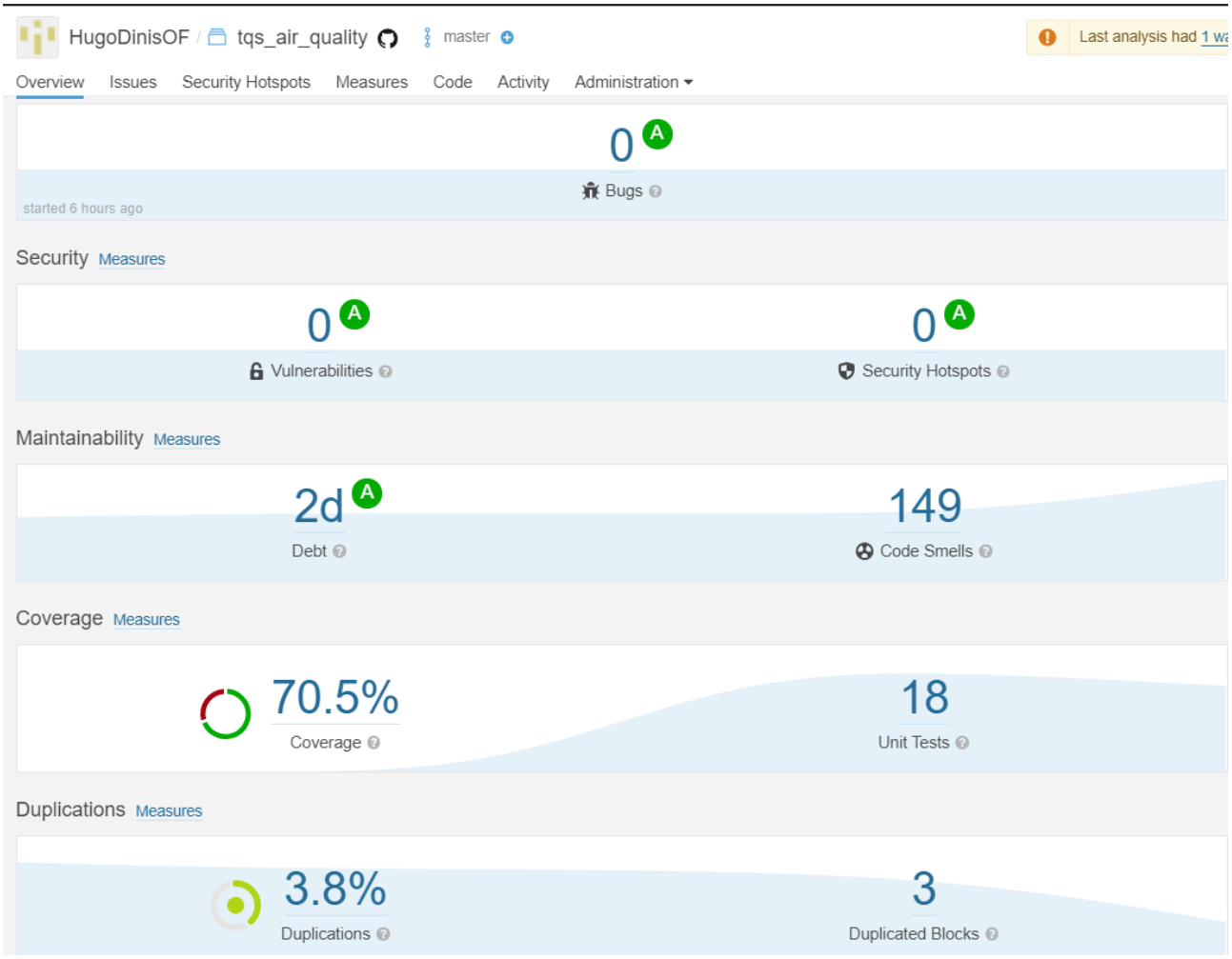
3.3 Functional testing

For Functional Testing of the Web Inteface I used the Selenium IDE and its webdriver with the WebdriverUI class (Had to remove the test of the name so it would pass in SonarCloud) .
It simply tests the branches of the project (“/today”,”/date”,”/historic”).

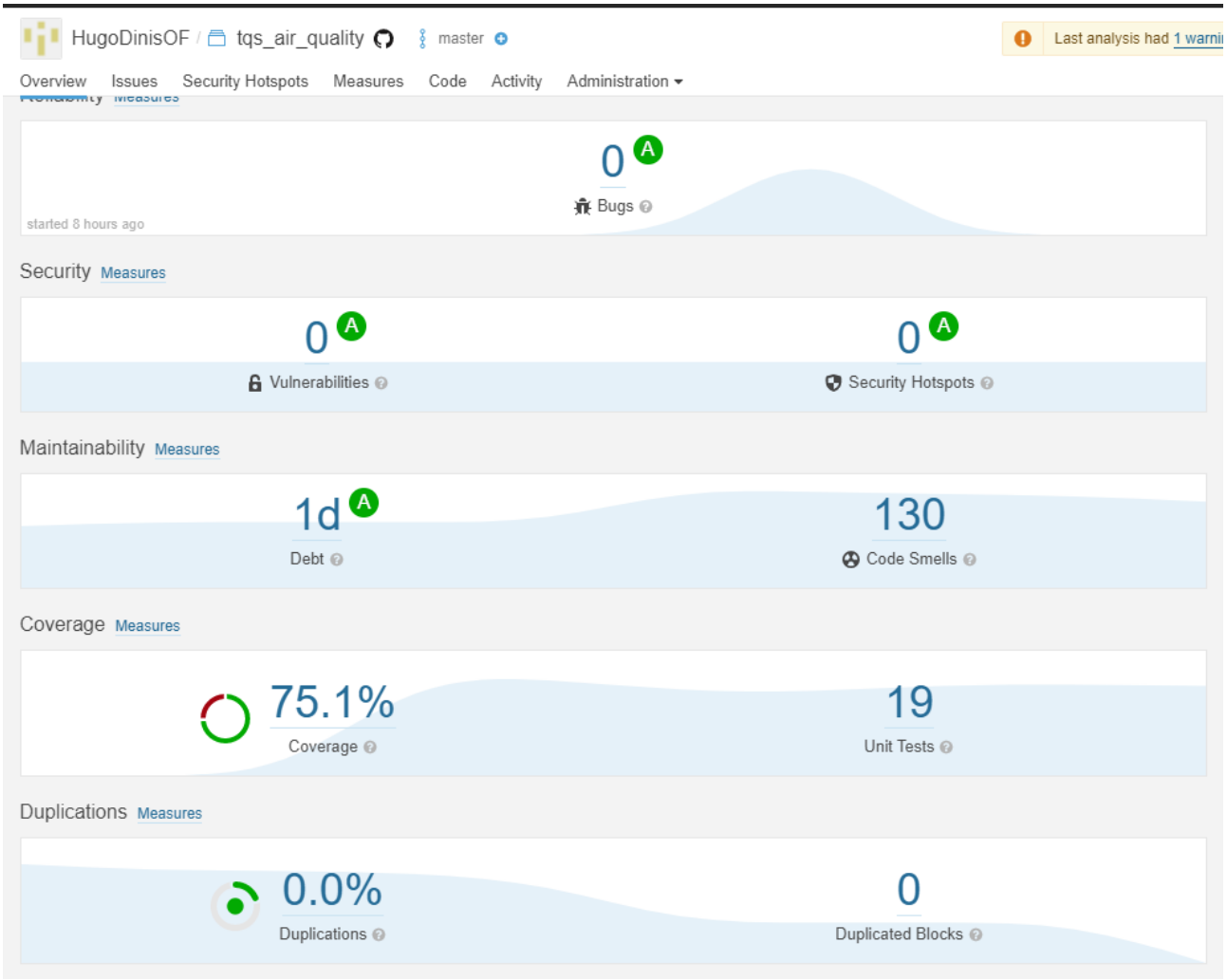
3.4 Static code analysis

For static code analysis I used SonarCloud integrated with JaCoCo to generate the reports of the coverage of the tests.

Before code smelling fixes:



After some code smell fixing:



The use of SonarCloud helped reduce some critical and even blocking code smells. And it helped reduce the duplicate code.

3.5 Continuous integration pipeline [optional]

4 References & resources

Project resources

- Video demo: https://github.com/HugoDinisOF/TQS_solo_project/blob/master/demo.webm
- Git repository: https://github.com/HugoDinisOF/TQS_solo_project
- QA dashboard: https://sonarcloud.io/dashboard?id=HugoDinisOF_TQS_solo_project

Reference materials

https://openweathermap.org/api/air-pollution
https://www.baeldung.com/
https://openweathermap.org/api/geocoding-api
https://www.selenium.dev/