

# HW1: Mid-term assignment report

Daniel José Matias Pinto [89341], 2020-04-15

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
1.1	Overview of the work.....	1
1.2	Limitations.....	1
<b>2</b>	<b>Product specification.....</b>	<b>1</b>
2.1	Functional scope and supported interactions.....	1
2.2	System architecture.....	2
2.3	API for developers.....	2
<b>3</b>	<b>Quality assurance.....</b>	<b>2</b>
3.1	Overall strategy for testing.....	2
3.2	Unit and integration testing.....	2
3.3	Functional testing.....	2
3.4	Static code analysis.....	2
3.5	Continuous integration pipeline [optional].....	2
<b>4</b>	<b>References &amp; resources.....</b>	<b>3</b>

## 1 Introduction

### 1.1 Overview of the work

Air Quality is a website where users can check the current statistics about air quality throughout the world. This website was developed with the objective of making a spring boot application that provides information about the air quality of some cities/regions of the world.

The website provides multiple parameters from gases information (CO, O3, SO2, NO2) to particulate matter concentrations (PM10, PM25) and even the pollen levels in the air (tree, grass, weed, mold).

### 1.2 Limitations

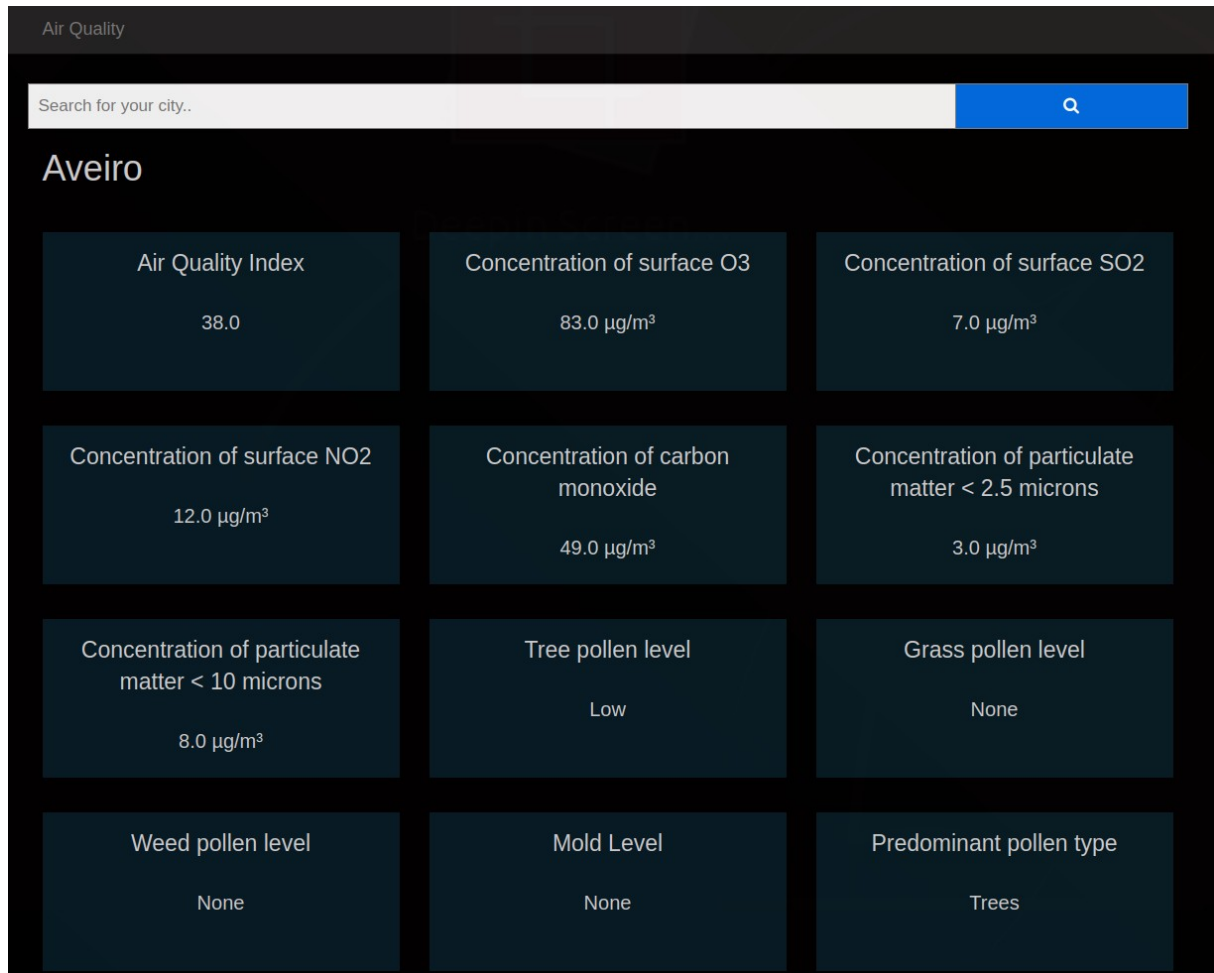
The application only allows city based searches. A search based on coordinates is not possible.

Besides that limitation, the information displayed is the current one relative to that city. It is not possible to search for past information.

## 2 Product specification

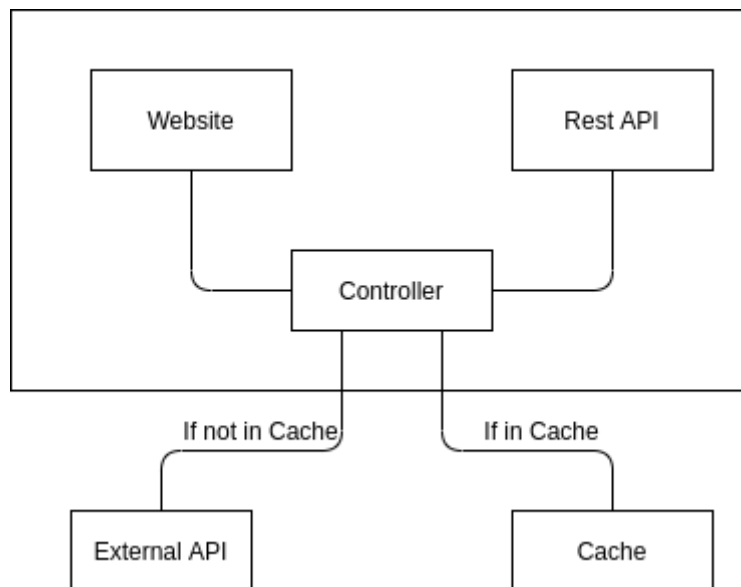
### 2.1 Functional scope and supported interactions

The app is ideal for people that have interest in knowing the current air conditions in their city or in others. The web page is very minimalist, making the app very intuitive. All the user has to do is access the page and search for a city and the information will be displayed.



### 2.2 System architecture

The app was built using Spring-boot and Thymeleaf. The call to the external API is made through a service and the communication between the frontend and the backend is made through the controller. It was also used a Spring framework that provides a cache abstraction API.



### 2.3 API for developers

**GET** <http://localhost:8080/>

Comments 0

<http://localhost:8080/>

sends client to the main page

**GET** <http://localhost:8080/airquality>

Comments 0

<http://localhost:8080/airquality>

sends client to the main page

**GET** <http://localhost:8080/airquality/{city}>

Comments 0

<http://localhost:8080/airquality/{city}>

sends client to corresponding city page where he can see the air quality information for that city

## 3 Quality assurance

### 3.1 Overall strategy for testing

The tests were developed using Junit and Selenium IDE.

### 3.2 Unit and integration testing

Junit was used to test the controller. The focus of these tests were to make sure the controller was returning the correct value.

```
@SpringBootTest
public class ControllerTest {
    @Autowired
    private AQController controller;

    @Test
    public void testController1(){
        controller = new AQController();
        String result = controller.index();
        assertEquals( expected: "redirect:/airquality/Aveiro", result);
    }

    @Test
    public void testController2(){
        controller = new AQController();
        String result = controller.airquality();
        assertEquals( expected: "redirect:/airquality/Aveiro", result);
    }

    @Test
    public void testPostRequest(){
        controller = new AQController();
        String city = "Madrid";
        String result = controller.showPage(city);
        assertEquals( expected: "redirect:/airquality/"+city, result);
    }
}
```

### 3.3 Functional testing

### 3.4 Static code analysis

The tool used for code analysis was Sonar Cloud. This tool showed some code smells that ended up being mostly unused imports.

## 4 References & resources

### Project resources

- Git repository: [https://github.com/DanielJMPinto/AirQuality\\_TQS](https://github.com/DanielJMPinto/AirQuality_TQS)
- Video demo : in the repository

### Reference materials

- <https://howtodoinjava.com/spring-boot2/spring-boot-cache-example/>
- <https://www.weatherbit.io/api/airquality-current>
  - API key: 887e0b117b414dccac1d127a785deab9