HW1: Mid-term assignment report

*Luís Valentim [93989]*, 2021-05-15

Índice

[HW1: Mid-term assignment report 1](#_Toc71576259)

[1 Introduction 1](#_Toc71576260)

[1.1 Overview of the work 1](#_Toc71576261)

[1.2 Current limitations 1](#_Toc71576262)

[2 Product specification 1](#_Toc71576263)

[2.1 Functional scope and supported interactions 1](#_Toc71576264)

[2.2 System architecture 1](#_Toc71576265)

[2.3 API for developers 1](#_Toc71576266)

[3 Quality assurance 2](#_Toc71576267)

[3.1 Overall strategy for testing 2](#_Toc71576268)

[3.2 Unit and integration testing 2](#_Toc71576269)

[3.3 Functional testing 2](#_Toc71576270)

[3.4 Static code analysis 2](#_Toc71576271)

[3.5 Continuous integration pipeline [optional] 2](#_Toc71576272)

[4 References & resources 2](#_Toc71576273)

# Introduction

## 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.

There has been developed a simple webpage to showcase a REST API for air quality information in a specific city, with a Cache service and no long-term storage solution. The main focus however was the quality assurance, mainly through unit tests, integration tests, functional testing and static code analysis however due to time constraints no continuous integration pipeline solution was implemented.

## Current limitations

As previously mentioned there is no CI pipeline implemented, there is only one external API, weatherbit, from which the service retrieves data, meaning the app is reliant on weatherbits API availability in order to work properly.

## Functional scope and supported interactions

## System architecture

<briefly present the software architecture. Include one or more diagrams.>

<detail the specific technologies/frameworks that were used>

## API for developers

<what services/resources can a developer obtain from your project? document your API endpoints>

<note: for the homework, you are expected to expose two “groups” of endpoints:

* Problem domain: get the forecast by region, etc.
* Cache usage statistics: how many hits/misses,… >



# Quality assurance

## Overall strategy for testing

[what was the overall test development strategy? E.g.: did you do TDD? Did you choose to use Cucumber and BDD? Did you mix different testing tools, like REST-Assured and Cucumber?...]

## Unit and integration testing

[which test cases did you considered? How were they implemented?]

[may add some screenshots/code snippets for clarification]

## Functional testing

[which test cases did you considered? How were they implemented?]

[may add some screenshots/code snippets]

## Static code analysis

[which tools/workflow was used to for static code analysis? Show and interpret the results. ]

[you may add some interesting lessons learned, e.g., some code smell reported by the tool that was difficult to spot and otherwise you wouldn’t address it]

## Continuous integration pipeline [optional]

[did you implement a CI pipeline? What was the setup? Illustrate with screenshots, if applicable]

# References & resources

Project resources

* Video demo [where did you place a short video demonstration of your solution; should be in the Git repository]
* Ready to use application: [**optional**; if you have the solution deployed in a public server, place the URL here]
* QA dashboard: [**optional**; if you have a quality dashboard available (e.g.: sonarcloud), place the URL here]

Reference materials

<document the key components (e.g.: libraries, API) or key references (e.g.: blog post) that were helpful and certainly would help other students pursuing a similar work>