

# HW1: Mid-term assignment report

Ana Alexandra Antunes [876543], v2022-04-07

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Overview of the work .....	1
1.2	Current limitations .....	1
<b>2</b>	<b>Product specification.....</b>	<b>2</b>
2.1	Functional scope and supported interactions .....	2
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 .....	3
3.4	Code quality analysis.....	3
3.5	Continuous integration pipeline [optional] .....	3
<b>4</b>	<b>References &amp; resources.....</b>	<b>3</b>

<All remarks like this should be removed from the final document!

This is a template for the expected **content/structure**. You may use any editing tool to prepare the report (LaTeX included).

Feel free to write in Portuguese or English, but do not mix languages between headings and body...>

## 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.

<briefly introduce your application: name the product, if applicable; what is its purpose?>

### 1.2 Current limitations

<explain the known limitations → unimplemented or faulty (but expected) features>

## 2 Product specification

### 2.1 Functional scope and supported interactions

<functional description of the application: who (actors) will use the application and for what? Briefly explain the main **usage scenarios**. >

### 2.2 System architecture

<briefly present the software architecture. Include one or more diagrams.>  
<detail the specific technologies/frameworks that were used>

### 2.3 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 COVID tracking data by region, etc.
- Cache usage statistics: how many hits/misses,... >

[ Base URL: localhost:8080/weather ]

client Regular user of the weather forecast API

GET

/now/{latitude},{longitude} get weather forecast of the current day for the given coordinates

GET

/recent/{latitude},{longitude}/{days} get weather forecast of the next days starting from today until the given number of days for the given coordinates

GET

/period/{latitude},{longitude}/{start},{end} get weather forecast of the given time period for the given coordinates

GET

/cached get weather forecasts previously requested and still present in cache

## 3 Quality assurance

### 3.1 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?...]

### 3.2 Unit and integration testing

[where did you use unit and integration test? for what? which was the implementation strategy?]

[may add some screenshots/code snippets for clarification]

### 3.3 Functional testing

[which user-facing test cases did you considered? How were they implemented?]  
[may add some screenshots/code snippets]

### 3.4 Code quality analysis

[which tools/workflow did you use 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]

### 3.5 Continuous integration pipeline [optional]

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

## 4 References & resources

### Project resources

Resource:	URL/location:
Git repository	<link to your TQs repo>
Video demo	< short video demonstration of your solution; consider including in the Git repository>
QA dashboard (online)	[ <b>optional</b> ; if you have a quality dashboard available (e.g.: sonarcloud), place the URL here]
CI/CD pipeline	[ <b>optional</b> ; if you have th CI pipeline definition in a server, place the URL here]
Deployment ready to use	[ <b>optional</b> ; if you have the solution deployed in server, 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**>