HW1: Mid-term assignment report

*Ana Alexandra Antunes [876543]*, v2020-03-27

[1 Introduction 1](#_Toc36219510)

[1.1 Overview of the work 1](#_Toc36219511)

[1.2 Limitations 1](#_Toc36219512)

[2 Product specification 1](#_Toc36219513)

[2.1 Functional scope and supported interactions 1](#_Toc36219514)

[2.2 System architecture 2](#_Toc36219515)

[2.3 API for developers 2](#_Toc36219516)

[3 Quality assurance 2](#_Toc36219517)

[3.1 Overall strategy for testing 2](#_Toc36219518)

[3.2 Unit and integration testing 2](#_Toc36219519)

[3.3 Functional testing 2](#_Toc36219520)

[3.4 Static code analysis 2](#_Toc36219521)

[3.5 Continuous integration pipeline [optional] 2](#_Toc36219522)

[4 References & resources 3](#_Toc36219523)

# Introduction

## Overview of the work

<contextualize the objectives of this project assignment in the scope of the TQS course>

<introduce your application: brief overview of the application. What is it good for? Introduce the name of the product, if it has one>

## Limitations

 <explain the known limitations/unimplemented (but planned) features>

# Product specification

## Functional scope and supported interactions

<functional (black-box) description of the application: who are the intended **users**? How/what for will they use it (**scenarios)**?>

## System architecture

<briefly present the software architecture. Include diagrams.>

<explain the supporting data models/data structures, i.e., the entities of your problem>

<detail the specific technologies/frameworks that were used>

## API for developers

<what services/resources can a developer obtain from your REST-API?>

<document the support endpoints>



# 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 screenshots/code snippets]

## Functional testing

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

[may add 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

* Git repository: <put URL; be sure that teachers can access>
* Video demo [explain where you placed 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]

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>