HW1: Mid-term assignment report

*Ana Alexandra Antunes [876543]*, v2025-03-26

[1 Introduction 1](#_Toc130550537)

[1.1 Overview of the work 1](#_Toc130550538)

[1.2 Current limitations 1](#_Toc130550539)

[2 Product specification 2](#_Toc130550540)

[2.1 Functional scope and supported interactions 2](#_Toc130550541)

[2.2 System architecture 2](#_Toc130550542)

[2.3 API for developers 2](#_Toc130550543)

[3 Quality assurance 2](#_Toc130550544)

[3.1 Overall strategy for testing 2](#_Toc130550545)

[3.2 Unit and integration testing 2](#_Toc130550546)

[3.3 Functional testing 3](#_Toc130550547)

[3.4 Code quality analysis 3](#_Toc130550548)

[3.5 Continuous integration pipeline [optional] 3](#_Toc130550549)

[4 References & resources 3](#_Toc130550550)

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

This 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…>

# 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.

<briefly introduce your application use case: name the product, if applicable; what is its general purpose, in your words?>

## Current limitations

 <explain the known limitations 🡪 these are unimplemented or faulty, but expected, features>

# Product specification

## Functional scope and supported interactions

<who (actors) will use the application and for what? Explain the main **interactions;** include/explain the experience with a visual summary. >

A screenshot of a smartphone

AI-generated content may be incorrect.

## System implementation architecture

<briefly present the software architecture.>

<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: list meals options, book a meal, check meal reservations/”ticket”,....
* 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

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

[may add some screenshots/code snippets for clarification]

## Functional testing

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

[may add some screenshots/code snippets]

## Non functional testing

[which non-functional test cases did you consider? Expected: performance study with, at least, load tests]

## Code quality analysis

[which tools/workflow did you use to for [static code analysis](https://www.sonarsource.com/learn/static-code-analysis-using-sonarqube/)?

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]

A screenshot of a computer

AI-generated content may be incorrect.

## Continuous integration pipeline [optional]

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

# 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, under /docs> |
| QA dashboard (online) | [**optional**; if you have a quality dashboard available online (e.g.: sonarcloud), place the URL here] |
| CI/CD pipeline | [**optional**; if you have th CI pipeline definition in a server, place the URL here] |
| Deployment ready to use | [**optional**; if you have the solution deployed and running in a 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**>