

Test plan

Introduction and Overview

This project is aimed toward providing a small API service that can handle basic data operations for tasks in a task manager setting. The timeline is 2 weeks, with all deadlines being met towards the end of the timeline.

The application is a small Task Manager API, with local persistence and an HTTP API interface. The User should be able to create new tasks, mark/unmark them as completed, delete a task and read all their existing tasks.

Goals and Objectives

The goal of this project is to create a software product that meets the requirements and is developed within the given timeframe. In this case it is part of the non-functional requirements that we must ensure test coverage of at least 80%.

Scope and Limitations

The test plan covers the entire development process of the project, and is not limited to functional testing, but includes performance testing of the system. The plan doesn't cover security or post-launch testing.

Test coverage is defined as Unit- and Integration test with JUnit. We will only test our own application logic, and this is where 80% of the code should be covered by tests. The testing of API endpoints is done within a closed docker environment, making it so Jacoco does not register classes concerned with server configuration, routing and handlers as tested. These classes will be excluded from the Jacoco report, even though they are still being tested.

Test Environment Requirements

- Java 21
- Maven as package manager
- Correctly installed Junit, JMeter and JacCoCo libraries

Testing Methodology

- *Unit testing*
 - *JUnit*
- *Integration testing*
 - *HTTP*
- *Performance testing*
 - *Jmeter*
- *Jacoco (coverage)*

For the testing methodology we will perform a variety of tests.

Unit testing will be done with JUnit-5 to assess the methods and the outputs they will provide. Furthermore, integration testing will be done with http-files, to test the REST API, the data it provides and its status codes.

JMeter will be used to test the performance of the system during higher loads.

Lastly the test coverage of the code will be monitored using Jacoco.

Scheduling and Milestones

Due to the short timeline of the project, all deliverables are set to be done at the project deadline. No milestones are therefore used, as it is viewed as an all or nothing delivery given the scope of the project.

Resources and Personnel Requirements

The project will be carried out by 2 developers and a dedicated tester.

Risk Assessment and Mitigation Plan

During development of the system, resources will be used to monitor the applications test coverage and strength to make sure the system is as stable as possible.

New risks, like bugs and unexpected behaviors, will be addressed during development to reduce technical debt.

Documentation and Record-Keeping

There is currently no plan to document the testing outside of the reports being generated by various plugins. This is due to documentation being added in a later plan describing workflows for continued integration, which will cover documentation of tests being run upon integration.

Reporting and Approval Processes

Reports will be generated from test resources. This includes a coverage report by JaCoCo library and a performance report. When they reach satisfactory levels, the project is considered satisfactorily tested.

Reports will then be delivered as a product of the testing phase.