Testing report

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

DELIVERABLE 2

DESING AND TESTING 2

2023-2024

JESUS FERNÁNDEZ RODRÍGUEZ

|  |  |
| --- | --- |
| Date | Version |
| 05/26/2024 | V1.0 |
| 07/07/2024 | V2.0 |

|  |  |
| --- | --- |
| Group: 21 | |
| Members | Email |
| Fernández Rodríguez, Jesús | [jesferrod1@alum.us.es](mailto:jesferrod1@alum.us.es) |
| García Rodríguez, Javier | [javgarrod5@alum.us.es](mailto:javgarrod5@alum.us.es) |
| González Ortiz, Miguel | [miggonort1@alum.us.es](mailto:miggonort1@alum.us.es) |
| Palomo García, Miguel | [migpalgar1@alum.us.es](mailto:migpalgar1@alum.us.es) |
| Periáñez Franco, Luis Javier | [luiperfra1@alum.us.es](mailto:luiperfra1@alum.us.es) |

GitHub repository: <https://github.com/JesusFern/Acme-SF-D04>

TABLE OF CONTENTS

[Executive Summary 3](#_Toc167735894)

[Introduction 3](#_Toc167735895)

[Contents 3](#_Toc167735896)

[Conclusion 11](#_Toc167735897)

[Bibliography 11](#_Toc167735898)

# Executive Summary

This report contains the information obtained through the execution of functional and performance tests. In order to show that the indicated methodology has been followed and the conclusions we have reached after the tests.

**Revision Table**

|  |  |  |
| --- | --- | --- |
| Revision Number | Date | Description |
| v1r0 | 05/26/2024 | First version |

# Introduction

The content of this report is divided into two parts, one with the functional testing analysis and the other with the performance testing analysis.

During the tests I had to make changes in the authorization of a method that was not well implemented, also after the performance tests I have implemented indexes as indicated in the theory slides.

# Contents

**Functional testing**

The tests have been performed following the methodology proposed in the theory slides, obtaining as much coverage as possible.

In my test case I have performed additional tests on the intermediate entity and its functionalities.

During the execution of the tester#replayer launcher it is observed that a lot of phantom requests are produced, however these do not affect the result of the testing or the coverage obtained.

Throughout the testing, 100% coverage has never been achieved because the lines “assert object != null;” can never encounter a null object. However, it is good practice to keep these lines of code as recommended.

Additionally, in some specific cases, there are lines of code where the coverage did not cover all possibilities because they are impossible scenarios. These lines have been intentionally kept, and their details will be explained further when appropriate.

**Project**

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja**

* List-All

The test of the list-all functionality consisted of viewing the list of published projects. For the hacking, an attempt was made to access this list without having sufficient permissions, in this case trying to log in a non-logged in user. The coverage obtained was 91.7%.

* List-mine

The test of the list-mine functionality consisted of viewing the list of projects whose manager is the logged-in person. For the hacking, an attempt was made to access this list without having sufficient permissions, in this case trying to log in a non-logged in user. The coverage obtained was 92.6%.

* Show

The test of the show functionality consisted of viewing the details of a project. For the hacking, we tried to access this list without having sufficient permissions, in this case trying to log in with a user of another role and make a show of a project of which you are not the manager and is not published, as well as trying to make a show of a project with an id that does not exist. The coverage obtained was 96.3%.

* Create

The test of the create functionality consisted in creating projects with all possible variations in their attributes including not allowed values to check that the appropriate error message is triggered. For the hacking we tried to create with a non-logged user. The coverage obtained was 90.4%.

* Update

The test of the update functionality consisted in updating projects with all possible variations in their attributes including not allowed values to check that the appropriate error message is triggered. For the hacking we tried to update with a non-logged user, also a non-existing project and it was also tested to try to update an unpublished project with a user logged in as manager but who is not the owner of that project. The coverage obtained was 92%.

* Delete

The test of the update functionality consisted in deleting a project. For the hacking we tried to delete a project without being logged in, also a non-existing project and we tried to delete an unpublished project while logged in as manager without being owner of that project. The coverage obtained was 90.8%.

* Publish

The test of the update functionality consisted in publishing a project. For the hacking we tried to publish a project that was already published, also a non-existing project and we tried to publish an unpublished project while logged in as manager without being owner of that project. The coverage obtained was 93.1%.

**User Story**

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja**

* List-in-project

The test of the list-in-project functionality consisted of viewing the list of user stories of a projects. For the hacking, an attempt was made to access this list without having sufficient permissions, also we also tried to access the user stories of an unpublished project from a user logged in as a manager but who is not the owner of that project and try to access the user stories of a non-existing project. The coverage obtained was 92.3%.

* List-mine

The test of the list-mine functionality consisted of viewing the list of user stories whose manager is the logged-in person. For the hacking, an attempt was made to access this list without having sufficient permissions, in this case trying to log in with a user of another role or a non-logged in user. The coverage obtained was 92%.

* Show

The test of the show functionality consisted of viewing the details of a user story. For the hacking, we tried to access this list without having sufficient permissions, in this case trying to log in with a user of another role and make a show of a user story of which you are not the manager and is not published, as well as trying to make a show of a user story with an id that does not exist. The coverage obtained was 96.1%.

* Create

The test of the create functionality consisted in creating user stories with all possible variations in their attributes including not allowed values to check that the appropriate error message is triggered. For the hacking we tried to create with a non-logged user. The coverage obtained was 89.5%.

* Update

The test of the update functionality consisted in updating user stories with all possible variations in their attributes including not allowed values to check that the appropriate error message is triggered. For the hacking we tried to update with a non-logged user, also a non-existing user story and it was also tested to try to update an unpublished user story with a user logged in as manager but who is not the owner of that user story. The coverage obtained was 91.3%.

* Delete

The test of the update functionality consisted in deleting a user story. For the hacking we tried to delete a user story without being logged in, also a non-existing user story and we tried to delete an unpublished user story while logged in as manager without being owner of that user story. The coverage obtained was 90.8%.

* Publish

The test of the update functionality consisted in publishing a user story. For the hacking we tried to publish a user story that was already published, also a non-existing user story and we tried to publish an unpublished user story while logged in as manager without being owner of that user story. The coverage obtained was 91.4%.

**Project User Story (Intermediate entity)**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

* List-in-project

The test of the list-in-project functionality consisted of viewing the list of user stories related to a project. For the hacking, an attempt was made to access this list without having sufficient permissions, also we also tried to access the relations of an unpublished project from a user logged in as a manager but who is not the owner of that project and try to access the relations of a non-existing project. The coverage obtained was 93%.

* Show

The test of the show functionality consisted of viewing the details of the relation of user stories in a project. For the hacking, we tried to access this list without having sufficient permissions, in this case trying to log in with a user of another role and make a show of a relation of which project you are not the manager and is not published, as well as trying to make a show of a relation with an id that does not exist. The coverage obtained was 96.3%.

* Create

The test of the create functionality consisted in creating user story – project relations with all possible variations in their attributes including not allowed values to check that the appropriate error message is triggered. For the hacking we tried to create with a non-logged user , also we tried to create a relationship of an unpublished project by logging in as a manager who is not the owner of that project. The coverage obtained was 92.4%.

In the validate method, it can be observed that the line of code: “if (!super.getBuffer().getErrors().hasErrors()) { “ , never encounters a scenario where the buffer has errors. This is because when you create a project-user-story from the screen for creating a relationship between a user story and a project, you select its attributes from a dropdown menu where all possible values are valid, so it will never produce errors. However, it has been decided to keep this line as a good practice.

* Delete

The test of the update functionality consisted in deleting the relation of a user story to a project. For the hacking we tried to delete a relation without being logged in, also a non-existing user story and we tried to delete an unpublished relation while logged in as manager without being owner of that project. The coverage obtained was 88.2%.

**Performance testing**

During the performance analysis, the performance before and after the indexes were implemented was analyzed.

* Analysing performance

Average of the times obtained before the implementation of indexes:

**Gráfico

Descripción generada automáticamente**

As can be seen, the grand average is 7.17 milliseconds before the implementation of indexes. It can also be seen that the methods with the highest average milliseconds are manager/project/create and manager/project/publish.

Average of the times obtained after the implementation of indexes:

Gráfico

Descripción generada automáticamente

As can be seen, the grand average is 7.33 milliseconds after the implementation of indexes. After index addition, the methods with the highest average milliseconds are manager/project/delete and manager/project/publish.

Comparison of the analyses obtained before and after the implementation of the indexes:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

As can be seen both intervals meets our 1-second sample requirement comfortably.

Z-test analysis:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

After the z-test the p-value (Valor critico de z (dos colas) ) is in the range (alpha – 1), where alpha = 0.05. Therefore, my changes have not obtained a relevant improvement, although the values obtained are different, globally they are the same.

* Profiling software

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

This was the result obtained after profiling software. As can be seen, the execution time of the validate method of the project publish is notably superior to the rest of the methods. However, its self time is 0 so it’s not that method that consumes too much time, but the method that it invokes.

* Profiling hardware

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

This was the result obtained after profiling hardware.

The memory of this computer is being moderately used, but it is not a clear bottleneck, the CPU and the network are little used so they are far from a bottleneck as well as the other components.

# Conclusion

Everything went as expected.

# Bibliography

Intentionally blank