Testing report

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

DELIVERABLE 2

DESING AND TESTING 2

2023-2024

MIGUEL GONZÁLEZ ORTIZ

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

|  |  |
| --- | --- |
| 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.

**Sponsorship**

**Texto

Descripción generada automáticamente**

* List-All

The test of the list-all functionality consisted of viewing the list of published sponsorships. For the hacking, an attempt was made to access this list without having sufficient permissions. The coverage obtained was 91.7%.

* List-mine

The test of the list-mine functionality consisted of viewing the list of sponsorships whose sponsor 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 92.6%.

* Show

The test of the show functionality consisted of viewing the details of a sponsorships. 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 sponsorship of which you are not the sponsor and is not published, as well as trying to make a show of a sponsorship with an id that does not exist. The coverage obtained was 97.1%.

* Create

The test of the create functionality consisted in creating sponsorships 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 95.7%.

* Update

The test of the update functionality consisted in updating sponsorships 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 sponsorship and it was also tested to try to update an unpublished sponsorship with a user logged in as sponsor but who is not the owner of that sponsorship. The coverage obtained was 95.9%.

* Delete

The test of the update functionality consisted in deleting a sponsorship. For the hacking we tried to delete a sponsorship without being logged in, also a non-existing sponsorship and we tried to delete an unpublished sponsorship while logged in as sponsor without being owner of that sponsorship. The coverage obtained was 92.7%.

* Publish

The test of the update functionality consisted in publishing a sponsorship. For the hacking we tried to publish a sponsorship that was already published, also a non-existing sponsorship and we tried to publish an unpublished sponsorship while logged in as sponsor without being owner of that sponsorship. The coverage obtained was 96.6%.

**Invoices**

**Texto

Descripción generada automáticamente**

* List-mine

The test of the list-mine functionality consisted of viewing the list of invoices whose sponsor 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 93.8%.

* Show

The test of the show functionality consisted of viewing the details of an invoice. 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 an invoice of which you are not the sponsor and is not published, as well as trying to make a show of an invoice with an id that does not exist. The coverage obtained was 96.5%.

* Create

The test of the create functionality consisted in creating invoices 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 94.9%.

* Update

The test of the update functionality consisted in updating invoices 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 invoice and it was also tested to try to update an unpublished invoice with a user logged in as sponsor but who is not the owner of that invoice. The coverage obtained was 94.8%.

* Delete

The test of the update functionality consisted in deleting an invoice. For the hacking we tried to delete an invoice without being logged in, also a non-existing invoice and we tried to delete an unpublished invoice while logged in as sponsor without being owner of that invoice. The coverage obtained was 90.9%.

**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, Gráfico en cascada

Descripción generada automáticamente**

Average of the times obtained after the implementation of indexes:

Gráfico, Gráfico en cascada

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Z-test analysis:

Tabla

Descripción generada automáticamente

After the z-test the p-value (Valor critico de z (dos colas) ) is in the range (0 - alpha), where alpha = 0.05. Therefore, my changes we can clearly see that it has not improved.

* Profiling software

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

We see that, for example, the first one, which is sponsorshipPublishService validate(), is consuming a lot. But in Self time, we see that all the data is at 0.0ms, which means that it’s not the method that is consuming too much time, but the methods that it invokes

* Profiling hardware

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

The performance monitor screenshot indicates that while running the tests, the processor, disk, and network resources are being utilized efficiently without any major bottlenecks or resource contention.

# Conclusion

Everything went as expected.

# Bibliography

Intentionally blank