DP2 2023-2024

Testing report D04

Acme Software Factory



Repository: https://github.com/rafcasceb/Acme-SF-D04

Group members:

* Flores de Francisco, Daniel [danflode@alum.us.es](mailto:danflode@alum.us.es)
* Castillo Cebolla, Rafael [rafcasceb@alum.us.es](mailto:rafcasceb@alum.us.es)
* Heras Pérez, Raúl [rauherper@alum.us.es](mailto:rauherper@alum.us.es)
* Mellado Díaz, Luis [luimeldia@alum.us.es](mailto:luimeldia@alum.us.es)
* Vento Conesa, Adriana [adrvencon@alum.us.es](mailto:adrvencon@alum.us.es)

GROUP C1.049

Version 1.0

20-05-24

Content Table

Contenido

[Executive summary 3](#_Toc167228835)

[Revision Table 4](#_Toc167228836)

[Introduction 5](#_Toc167228837)

[Contents 6](#_Toc167228838)

[Functional testing 6](#_Toc167228839)

[Operations by administrator on Banners 6](#_Toc167228840)

[Performance testing 10](#_Toc167228841)

[Performance data 10](#_Toc167228842)

[Hypothesis contrast 11](#_Toc167228843)

[Conclusions 12](#_Toc167228844)

[Bibliography 12](#_Toc167228845)

# Executive summary

This report will provide a comprehensive analysis of the testing procedures and results, featuring distinct sections on functional testing and performance testing. Our approach will be precise yet accessible, aiming to enhance understanding and ensure a high-quality final product.

# Revision Table

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description of the changes | Sprint |
| 20/05/2024 | 1.0 | * Executive summary * Introduction * Functional testing | 4 |

# Introduction

This document will provide a detailed analysis of the testing procedure and results for the following feature:

* Operations by administrators on Banners.

The content of a testing report is organized into two chapters:

* Functional testing: a listing with the test cases implemented, grouped by feature. For each test case, a succinct description plus a clear indication on how effective it was at detecting bugs are provided.
* Performance testing: it provides adequate charts, a 95%-confidence interval for the wall time taken by the application to serve the requests in the functional tests and a 95%- confidence hypothesis contrast.

# Contents

## Functional testing

### Operations by administrator on Banners

*Test case 1: list (list-all)*

For this command, we just selected the button for listing several banners of the administrators many times.

For hacking, we considered accessing the URL by a wrong role. Trying to access it with a good

role but a wrong user doesn’t make sense (any administrator can list the banners). Finally, trying to access with an anonymous user.

It provided a coverage of 90.9 %, covering all instructions except a default assertion, which is logical. No bugs were detected.

*Test case 2: show*

For this command, we selected several banners using an administrator account to see their details.

For hacking, we tried accessing with a wrong role. Trying to access it with a good role but a wrong user doesn’t make sense (any administrator can show the banners). Finally, we tried accessing a contract with an anonymous user.

It provided a coverage of 94.2 %, covering all instructions except a default assertion, which is logical. No bugs were detected.

*Test case 3: create*

For this command, we have tried to create a new banner. For each attribute we have checked the system rejects all different types of invalid data. Later, for each attribute, we have checked the system accepts all different types of valid data.

For hacking, the framework through web browser only supports to test GET hacking operations.

It provided a coverage of 93.6 %, covering all instructions except a default assertion, which is logical. No bugs were detected.

*Test case 4: update*

For this command, we have updated a banner of the data base. For each attribute we have checked the system rejects all different types of invalid data. Later, for each attribute, we have checked the system accepts all different types of valid data.

For hacking, the framework through web browser only supports to test GET hacking operations

It provided a coverage of 93.6 %, covering all instructions except a default assertion, which is logical. No bugs were detected.

*Test case 5: delete*

For this command we tried deleting some banners using an administrator account. There are no exceptions for deleting, except for being an administrator

For hacking, the framework through web browser only supports to test GET hacking operations.

It provided a coverage of 57.1 % since the unbind method is not executed. If we remove the unbind method it would provide an approximate coverage of 90.0% covering all instructions except a default assertion, which is logical. No bugs were detected

## Performance testing

### Performance data

### Hypothesis contrast

# Conclusions

# Bibliography

Intentionally blank.