26 DE MAYO DE 2024

TESTING REPORT

GROUP C-1.047

Acme SF, Inc.

Repository link: https://github.com/Cargarmar18/Acme-SF

Manuel Castillejo Vela
Email: mancasvel@alum.us.es
Carlos García Martínez
Email: cargarmar18@alum.us.es
María C. Rodríguez Millán
Email: marrodmil@alum.us.es

Table of contents

1. Executive summary

The functional testing chapter provides a detailed listing of tests implemented by feature, each with a complete description of the function. Overall, the functional tests were highly effective, revealing mainly minor bugs that significantly enhanced the application.

The performance testing chapter presents an analysis of the time taken by the project to serve requests (being one 10% more than the other) on a computer, including detailed charts and 95%-confidence intervals. These results are crucial for optimizing and analyzing the performance of the system.

2. Revision table

Revision number	Date	Description
1.0.0	26-05-2024	Creation and development of the document.

3. Introduction

This document provides a comprehensive analysis of the testing and performance evaluation of the mandatory features develope. It aims to ensure that the implemented functionalities meet the required standards of correctness, security, and efficiency. By systematically examining each test case and performance metric.

The analysis is divided into two main sections: Functional Testing and Performance Testing. The Functional Testing section is further broken down into two subsections. The first subsection, Test Case Analysis, delves into the specifics of individual test cases for various features such as project

and user story management. Each test case is described in detail, outlining the scenarios tested, the methodology used, and the effectiveness in identifying bugs. This includes testing both normal and the limit cases, as well as security measures to ensure that only authorized roles can perform certain actions.

The second subsection, Coverage Analysis, provides a quantitative assessment of the test coverage across different classes. It includes a detailed table showing the coverage percentages for various services related to project and user story management.

The Performance Testing section focuses on evaluating the efficiency of the system by analyzing the time it takes to perform various manager-related actions. The analysis identifies the most time-consuming actions and provides insights into why certain operations are less efficient.

Further, the Confidence Interval subsection presents the computations for the 95%-confidence interval for the time taken by the system to serve requests. The document also includes a Hypothesis Contrast subsection that simulates a performance improvement scenario by assuming a 10% increase in efficiency. Through Z-testing analysis, the document evaluates whether the observed performance changes are statistically significant.

4. Contents

4.1Testing analysis:

In this section, I will analyze the test suite generated for the mandatory banner features within the administrator role. This analysis is divided into two parts: the first part focuses on examining each individual test case, while the second part addresses the coverage analysis.

4.1.1 Test case analysis:

- List banner:

This test case involves accessing the banner listing as an administrator to verify if the feature is correctly implemented. This was done by interacting with that listing as the correct role.

Additionally, to test the security of this feature, we attempted to access the listings without an administrator role for many roles. As expected, we received an error indicating that other roles are not authorized.

- Show banner:

This test case involves accessing the specific data of a banner as an administrator to verify if the feature is correctly implemented. This was done accessing the specific data of banners.

Additionally, to test the security of this feature, we attempted to access the show without an administrator role and testing for an id that doesn't correspond to a banner. As expected, we received an error indicating that other roles are not authorized and that user neither.

- Create banner:

This test case involves creating a banner as an administrator to verify if the feature is correctly implemented. This was done by trying many combinations in the form to check if it doesn't panic when sending an empty form or wrong values (including the custom constraints) and checking how the feature handles conflictive values such as the first and last possible value, SQL injection, start and end of moment…

Additional testing for hacking cannot be done due to the framework not supporting POST hacking testing.

- Delete banner:

This test case involves deleting a specific banner as an administrator to verify if the feature is correctly implemented. This was done by interacting with the specific banner and deleting it.

Additional testing for hacking cannot be done due to the framework not supporting POST hacking testing.

- Update banner:

This test case involves updating a banner as an administrator to verify if the feature is correctly implemented. This was done by trying many combinations in the form to check if it doesn't panic when sending an empty form or wrong values (including the custom constraints) and checking how the feature handles conflictive values such as the first and last possible value, SQL injection, start and end of moment…

Additional testing for hacking cannot be done due to the framework not supporting POST hacking testing.

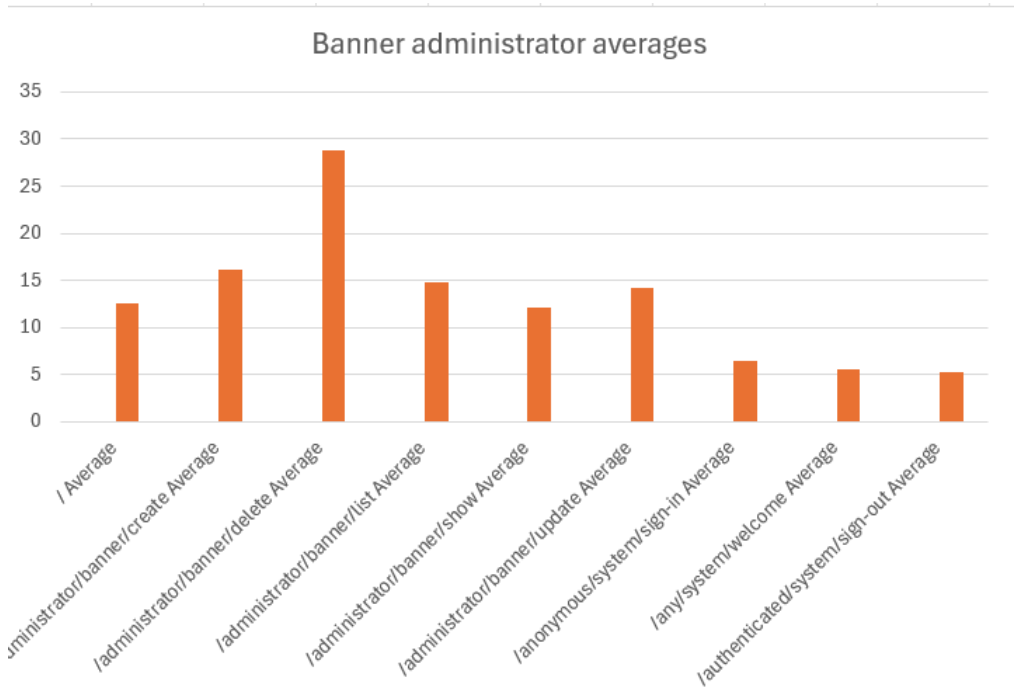
4.1.2 Coverage analysis:

Class Name	Coverage
AdministratorBannerUpdateService	93,8%
AdministratorBannerCreateService	93,7%
AdministratorBannerDeleteService	85,7%
AdministratorBannerListService	93,3%
AdministratorBannerShowService	100%

Providing the table with the data, we can conclude that the coverage is overall complete knowing that the non-covered code, is mainly code provided by the ACME framework or unreachable.

4.2Performance analysis:

After executing the tests, the following chart shows the results of the average performance of each banner-administrator interaction.



Looking into this graph we can conclude that the most time-consuming action is the deletion of a banner. After further analysis, the delete function overall seems to be the most inefficient along the project. Although, banner related functionalities seem to be the most time consuming of the project as well.

- Confidence interval:

Let us now see the computations for the confidence interval:

Column1					
			Interval (ms)	12,1125228	15,07998515
Mean	13,59625398		Interval (s)	0,012112523	0,015079985
Standard Error	0,75383836				
Median	8,7885				
Mode	11,2607				
Standard Deviation	12,81525212				
Sample Variance	164,2306869				
Kurtosis	35,27947183				
Skewness	4,565244504				
Range	136,9415				
Minimum	2,9828				
Maximum	139,9243				
Sum	3929,3174				
Count	289				
Confidence Level(95,0%)	1,483731175				

Even if we don't have any specific requirement for the confidence interval in our project, a confidence interval of [0,012 , 0,015] assuming the 95% confidence level is an overall great interval time.

- Hypothesis contrast:

We don't have any requirement regarding the performance of the system and indexes are already implemented so to simulate a hypothesis contrast let us assume an increase of 10% over the time.

Keeping this in mind, this is the confidence interval of the generated data in contrast.

Before				*After*		
Mean	13,59625398			Mean	14,95587938	
Standard Error	0,75383836			Standard Error	0,829222196	
Median	8,7885			Median	9,66735	
Mode	11,2607			Mode	12,38677	
Standard Deviation	12,81525212			Standard Deviation	14,09677733	
Sample Variance	164,2306869			Sample Variance	198,7191312	
Kurtosis	35,27947183			Kurtosis	35,27947183	
Skewness	4,565244504			Skewness	4,565244504	
Range	136,9415			Range	150,63565	
Minimum	2,9828			Minimum	3,28108	
Maximum	139,9243			Maximum	153,91673	
Sum	3929,3174			Sum	4322,24914	
Count	289			Count	289	
Confidence Level(95,0%)	1,483731175			Confidence Level(95,0%)	1,632104293	
Interval (ms)	12,1125228	15,07998515		Interval (ms)	13,32377508	16,58798367
Interval (s)	0,012112523	0,015079985		Interval (s)	0,013323775	0,016587984

Now, the Z-testing analysis for both cases is as follows:

z-Test: Two Sample for Means		
	Before	*After*
Mean	13,59625398	14,95587938
Known Variance	164,2306869	198,7191312
Observations	289	289
Hypothesized Mean Difference	0	
z	-1,21323491	
P(Z<=z) one-tail	0,11252001	
z Critical one-tail	1,644853627	
P(Z<=z) two-tail	0,22504002	
z Critical two-tail	1,959963985	

To assess the impact of the change, we must focus on the "P(Z<=z) two-tail" row, which indicates a value of 0.22. Given that α = 1 - confidence level = 0.05, we can ascertain that, since P(Z<=z) exceeds 0.05, the changes did not significantly affect our performance. Although the sample times differ, overall, they remain consistent.

5. Conclusions

In conclusion, the comprehensive testing and performance analysis performed provide valuable insights into the system's efficiency. By examining test cases, we have confirmed that critical functionalities adhere to the required standards of correctness and security. Regarding performance, key areas for optimization have been identified. The observed variations in performance metrics, along with the computation of confidence intervals and hypothesis contrasts, offer a profound understanding of the system's efficiency under various scenarios.

6. Bibliography

Intentionally blank