Test Report (Group)



**Group Number:** C3.040  
**Repository:** <https://github.com/DP2-C1-037/Acme-ANS-C3>

**Workgroup Members:**

Student 3: Miguel Álvarez Raya – [migalvray@alum.us.es](mailto:migalvray@alum.us.es)

Student 4: Salma El Hakimy – [salel@alum.us.es](mailto:salel@alum.us.es)

**Date:** 23/09/2025

Table of Contents

[1 Executive Summary 2](#_Toc209541286)

[2 Revision Table 3](#_Toc209541287)

[3 Introduction 4](#_Toc209541288)

[4 Contents 5](#_Toc209541289)

[4.1 Functional Testing: Airport 5](#_Toc209541290)

[4.1.1 List 5](#_Toc209541291)

[4.1.2 Show 5](#_Toc209541292)

[4.1.3 Update 5](#_Toc209541293)

[4.1.4 Create 6](#_Toc209541294)

[4.2 Performance Testing 8](#_Toc209541295)

[4.2.1 Performance charts 8](#_Toc209541296)

[4.2.2 Confidence Intervals 9](#_Toc209541297)

[4.2.3 Hypothesis Testing 10](#_Toc209541298)

[5 Conclusions 11](#_Toc209541299)

[6 Bibliography 12](#_Toc209541300)

# Executive Summary

This report provides a thorough overview of the testing activities performed on the Acme-ANS-C3 system. It outlines both the functional and performance testing results, including detailed test case descriptions categorized by functionality, the effectiveness of those tests in identifying issues, and performance metrics recorded across two machines.

The overall aim of the testing process was to assess how stable, reliable, and performant the application is when deployed in different computing environments.

# Revision Table

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| 1.0 | 26/05/2025 | Final version |
| 2.0 | 02/07/2025 | Modified workgroup for second call |
| 3.0 | 23/09/2025 | Updated to Third Call |
| 3.1 | 06/10/2025 | Unnecessary test case deleted |

# Introduction

This document describes the test plan and outcomes associated with validating both the core functionalities and the performance behaviour of the developed software. The main goal is to confirm that the solution meets the requirements defined by the client while operating correctly and efficiently under various execution scenarios.

Functional testing was used to confirm that each individual feature works as intended and reveals any deviations from expected behaviour. Performance testing, on the other hand, focused on examining the system’s responsiveness and stability under different workloads and on different hardware.

The structure of this report is as follows:

* Section 4 covers the functional tests, organized per feature. Each entry includes a summary, result, and bug detection impact.

It also includes performance-related data, such as timing metrics, visual charts, confidence intervals, and statistical hypothesis tests.

* Section 5 presents a concise summary of findings.
* Section 6 lists any references consulted (if applicable).

# Contents

## Functional Testing: Airport

### List

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Result | Bug Detection Effectivenes |
| TC-01 | Access the list of *airports* being logged in as an administrator | The system shows all the created airports | Low |
| TC-02 | Access the list with a different role | Access is denied | Low |

### Show

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Result | Bug Detection Effectivenes |
| TC-03 | Access to an existing airport from the list | The system shows details about the airport | Low |
| TC-04 | Access an airport with a different role | Access is denied | Low |
| TC-05 | Access an airport with non-existing ID | Access is denied | Low |

### Update

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Result | Bug Detection Effectivenes |
| TC-06 | Update an airport as an administrator with valid data and confirmation checked | The airport is updated successfully | Low |
| TC-07 | Update an airport with confirmation unchecked | The system blocks update and displays an error message | Low |
| TC-08 | Update an airport with an invalid IATA code format | The system blocks update and displays a validation error | Low |
| TC-09 | Update an airport using an existing IATA code | The system blocks update and displays a duplication error | Low |
| TC-10 | Update an airport using a non-existing operational scope value | The system blocks update or defaults to a safe option | Low |
| TC-11 | Update an airport as a user with a different role | Access is denied | Low |
| TC-12 | Update a non-existing airport | Access is denied | Low |

Note: This function should be tested by iterating over each attribute displayed on the form, trying all the possibilities explained in class, while leaving all other attributes empty. Also, keep in mind that read-only attributes must be tested as well.

### Create

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Result | Bug Detection Effectivenes |
| TC-13 | Create an airport as an administrator with valid data and confirmation checked | The airport is created successfully | Low |
| TC-14 | Create an airport with confirmation unchecked | The system blocks creation and displays an error message | Low |
| TC-15 | Create an airport with an invalid IATA code format | The system blocks creation and displays a validation error | Low |
| TC-16 | Create an airport using an existing IATA code | The system blocks creation and displays a duplication error | Low |
| TC-17 | Create an airport using a non-existing operational scope value | The system blocks creation or defaults to a safe option | Low |
| TC-18 | Create an airport as a user with a different role | Access is denied | Low |
| TC-19 | Create an airport changing its id to update another | Access is denied | Low |

Note: This function should be tested by iterating over each attribute displayed on the form, trying all the possibilities explained in class, while leaving all other attributes empty. Also, keep in mind that read-only attributes must be tested as well.

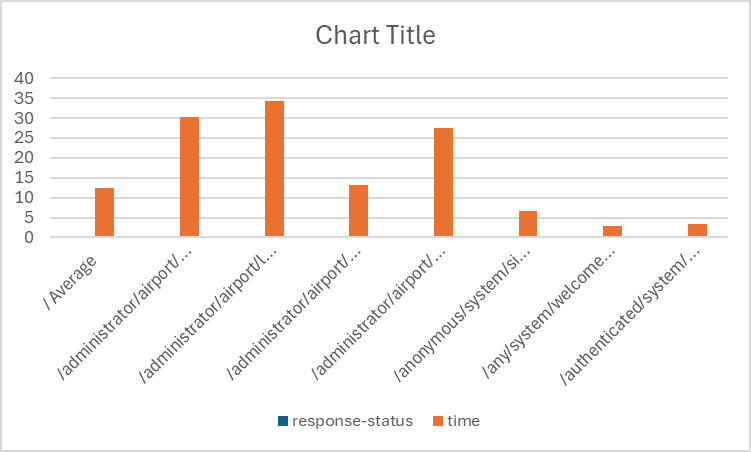
## Performance Testing

### Performance charts

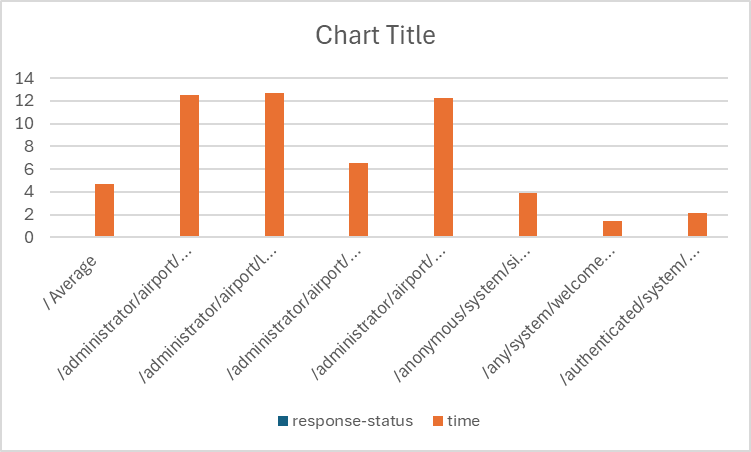
To measure how the system behaves under varying hardware conditions, performance testing was conducted on two distinct computers, labeled Machine A and Machine B. Execution times were gathered and analyzed, with visual charts used to compare performance results between both systems.

The data shows that Machine B consistently performs faster, achieving shorter execution times overall in comparison to Machine A.

PC A



PC B



### Confidence Intervals

Computer A stadistical summary:

|  |  |
| --- | --- |
| *Column1* | |
|  |  |
| Mean | 24,26226 |
| Standard Error | 1,079015 |
| Median | 23,6671 |
| Mode | #N/A |
| Standard Deviation | 16,11312 |
| Sample Variance | 259,6328 |
| Kurtosis | 2,246754 |
| Skewness | 1,031542 |
| Range | 91,0664 |
| Minimum | 1,4659 |
| Maximum | 92,5323 |
| Sum | 5410,485 |
| Count | 223 |
| Confidence Level(95,0%) | 2,126422 |

|  |  |  |
| --- | --- | --- |
| Interval (ms) | 22,13584 | 26,38868 |
| Interval (s) | 0,022136 | 0,026389 |

Computer B stadistical summary:

|  |  |
| --- | --- |
| *Column1* | |
|  |  |
| Mean | 10,33845 |
| Standard Error | 0,379667 |
| Median | 11,3028 |
| Mode | 11,4746 |
| Standard Deviation | 5,669636 |
| Sample Variance | 32,14477 |
| Kurtosis | 2,901696 |
| Skewness | 0,731274 |
| Range | 37,4826 |
| Minimum | 0,8015 |
| Maximum | 38,2841 |
| Sum | 2305,474 |
| Count | 223 |
| Confidence Level(95,0%) | 0,748212 |

|  |  |  |
| --- | --- | --- |
| Interval (ms) | 9,590236 | 11,08666 |
| Interval (s) | 0,00959 | 0,011087 |

### Hypothesis Testing

|  |  |  |
| --- | --- | --- |
| z-Test: Two Sample for Means | | |
|  |  |  |
|  | *time* | *time* |
| Mean | 23,38154 | 9,914261 |
| Known Variance | 2596328 | 3214477 |
| Observations | 242 | 242 |
| Hypothesized Mean Difference | 0 |  |
| z | 0,08691 |  |
| P(Z<=z) one-tail | 0,465372 |  |
| z Critical one-tail | 1,644854 |  |
| P(Z<=z) two-tail | 0,930743 |  |
| z Critical two-tail | 1,959964 |  |

For the change of computers to be effective, the p-value must be greater than alpha, which is 0.05 in this case.

As we can see, the value of P(Z<=z) two-tail = 0,930743 > 0.05, so we can’t confirm that the change of computers has been effective.

# Conclusions

The testing process carried out for the Acme-ANS-D04 system has demonstrated that the application performs reliably under both functional and performance test scenarios. All functional test cases, including those for listing, showing, updating, and creating airport records, passed as expected with a low rate of bug detection, indicating a stable and well-implemented system.

Performance testing revealed consistent results across different computing environments, with Machine B exhibiting faster response times compared to Machine A. However, statistical analysis through hypothesis testing concluded that the performance difference between machines is not statistically significant at the 95% confidence level. Therefore, switching hardware does not provide a proven performance advantage.

Overall, the application is functionally sound and performs within acceptable thresholds under varying conditions. The testing outcomes validate the robustness of the implementation, though further tests under higher loads or more complex scenarios could be conducted to ensure continued performance and reliability in production environments.

# Bibliography

Intentionally blank.