# **Testing Report**



|  |
| --- |
| **Group:** C1.031 |
| **Repository:** https://github.com/aaronma300604/DP2-C01-31 |
| **Members:**  **Student #1:** López Ramos, Daniel ([danlopram@alum.us.es](mailto:danlopram@alum.us.es))  **Student #2:** Ángel Postigo, Estrella ([estangpos@alum.us.es](mailto:estangpos@alum.us.es))  **Student #3:** Miranda Balastegui, Iván ([ivamirbal@alum.us.es](mailto:ivamirbal@alum.us.es))  **Student #4:** Terrón Hernández, Diego ([dieterher@alum.us.es](mailto:dieterher@alum.us.es))  **Student #5:** Mayoral Ansias, Aarón ([aarmayans@alum.us.es](mailto:aarmayans@alum.us.es)) |
| **Date:** Seville May 25, 2025 |

Table of Contents

[**Testing Report** 1](#_Toc199081253)

[**Revision Table** 3](#_Toc199081254)

[**Introduction** 3](#_Toc199081255)

[**Contents** 3](#_Toc199081256)

[**Functional Testing** 3](#_Toc199081257)

[**Performance Testing** 5](#_Toc199081258)

[**Conclusion** 10](#_Toc199081259)

[**Bibliography** 10](#_Toc199081260)

# **Revision Table**

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| 1.0 | 25/05/2025 | Intial version – all sections added |
| 2.0 | 30/06/2025 | Second version – changes in excel screenshots added |

# **Introduction**

This document aims to present an analysis of the various tests conducted to evaluate code coverage for Student 3, along with an assessment of the system’s performance in relation to the requirements assigned to that student.

# **Contents**

## **Functional Testing**

The following is a list of functional tests performed for the entity Flight Assignment:

-**Create-fa.safe**: This test verifies the functionality of creating new flight assignments for a crew member. It includes various scenarios such as empty forms, invalid inputs, and valid data to ensure proper entity creation. It was particularly useful in identifying null pointer issues in the code.

-**Delete-fa.safe**: This test assesses the correct operation of the delete feature by removing a flight assignment. It helped uncover an issue related to cascade deletions involving flight assignments and activity logs.

-**Update-fa.safe**: This test evaluates the update functionality of a flight assignment through several cases, including empty fields, invalid data, and valid inputs. It aimed to prevent runtime errors like null pointers.

-**Publish-fa.safe**: This test validates the publishing functionality of a flight assignment by considering the scenarios defined in the student’s requirements. It also includes empty, invalid, and valid input cases. The test helped detect potential issues in the code ensuring that assignments were only published under the correct conditions.

-**List-show-fa.safe**: This test ensures that the list of created flight assignments is correctly displayed, including the details of each entry. No bugs were identified during its execution.

-**Create-fa.hack**: This test examines how the system handles unauthorized actions using methods like POST manipulation in the flight assignment creation service. Fields were altered via browser developer tools (e.g., modifying relationship IDs like Leg) to simulate unauthorized input. These tests proved particularly effective at uncovering logic flaws in conditional statements.

-**Delete-fa.hack**: This test focuses on the system's response to unauthorized deletion attempts through POST hacking. Similar to the previous test, developer tools were used to inject forbidden data such as altered relationship IDs.

-**Update-fa.hack**: Closely related to Create-fa.hack, this test targets the update service, attempting unauthorized modifications via POST requests. It also used developer tools to alter relationship fields like Leg, revealing weaknesses in conditional validation logic. The test was absolutely useful for detecting a problem in the drop-down of the legs attribute.

-**Publish-fa.hack**: This test checks the system’s resilience against unauthorized publishing actions. It involved modifying fields using browser tools to introduce invalid relationships, effectively uncovering issues in the service's logical flow.

- Show-list-fa.hack: This test simulates hacking attempts to access the flight assignment information of a crew member (including the list and details of each assignment) from other roles or unauthorized crew members by manipulating the URL. The test helped improve the validation logic to prevent unauthorized access to this information.

The test coverage achieved for the entity Flight Assignment is 97,9%.

Texto

El contenido generado por IA puede ser incorrecto.

All lines of this entity have been tested either fully or partially, except for two lines, one in the services Create, Update, Publish and Delete that prevents to return a null value in a function when retrieving data from the database in the case that the database had no upcoming legs.

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

And other one in the Publish Service that prevents a null point exception in case that the flight crew member id were not correctly taken from the database:

Texto

El contenido generado por IA puede ser incorrecto.

The following is a list of functional tests performed for the entity Activity Log:

**List-show-al.safe**: This test verifies the correct behaviour of the application to list and show activityLogs. The test did not find any errors but was helpful to assure that everything is under control.

**Create-al.safe**: This test verifies the correct creation of activity logs by evaluating different input scenarios including empty forms, invalid values, and valid data. It ensures that the system correctly handles user input and stores activity logs as expected. The test was useful in detecting potential null pointer exceptions during the creation process.

**Delete-al.safe**: This test checks the proper behaviour of the deletion functionality for activity logs. It ensures that activity logs can be successfully removed from the system and that no residual data or orphaned references remain.

**Update-al.safe**: This test evaluates the update mechanism for activity logs by simulating various editing scenarios. It includes empty fields, incorrect data formats, and valid updates to confirm that the system properly validates input and prevents runtime issues.

**Publish-al.safe**: This test checks the logic for publishing an activity log, in accordance with the specific requirements defined for this feature. It tests edge cases such as missing or invalid values and verifies that only eligible activity logs can be published (activity logs related to flight assignments that are published). This test helped identify logic errors related to publication constraints.

**Create-al.hack**: This test examines the system’s behavior when attempting unauthorized creation of activity logs through methods such as POST request manipulation. Fields were modified using browser developer tools to insert unauthorized data (e.g., incorrect IDs or user assignments). This helped reveal insufficient validation and poor handling of unexpected inputs.

**Delete-al.hack**: This test evaluates the system’s defense against unauthorized deletion attempts using hacking techniques such as direct POST requests. It includes cases where unauthorized users attempt to delete logs by altering parameters. The test proved effective in identifying vulnerabilities in access control.

**Publish-al.hack**: This test simulates hacking attempts to publish activity logs under unauthorized or invalid conditions using browser manipulation tools. By injecting invalid data, it was possible to test whether the system properly blocks unauthorized publishing actions. This test helped uncover flaws in permission checks and data validation.

**Update-al.hack**: Similar to the create and publish hack tests, this test targets the update functionality of activity logs. It involved altering form values and relationships using browser tools to inject invalid or unauthorized data. The test revealed weak points in conditional logic and validation layers, particularly in services lacking robust input sanitization.

**List-show-al.hack**: This test simulates hacking attempts to access the list of activity logs and their content by manipulating the URL of a flight crew member while logged in with other roles, such as managers or agents. The test helped confirm that the validation method in these two services correctly prevents unauthorized roles from viewing the information.

The test coverage achieved for the entity Activity Log is 98.8%.

Texto

El contenido generado por IA puede ser incorrecto.

All lines of this entity have been tested either fully or partially.

## **Performance Testing**

A performance evaluation of the system will now be conducted by executing the functional tests previously described. These tests have been performed under two distinct scenarios:

* Without indexes for query optimization: The Excel file “tester-performance-clean-no-index.xlsx” contains the average performance results of the test operations as illustrated in the following chart.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

As observed, the most time-consuming operation on average is the creation of activity logs, which takes 37.34 milliseconds in average.

Below are some statistics regarding the operations:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

As we can observe, he system without indexes gives a confidence interval ranging from 11.108 milliseconds to 12.75 milliseconds.

* With indexes for query optimization: The Excel file “tester-performance-clean-index.xlsx” contains the average performance results of the test operations, as illustrated in the following chart.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

As we can see, the most time-consuming operation on average is still the creation of activity logs, which takes 36.10 milliseconds in average. Almost all operations have slightly decreased their response time compared to the previous analysis. However, there does not appear to be a significant change in their overall performance.

Below are some statistics regarding the operations:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

As we can observe, the system with indexes gives a confidence interval ranging from 10,73 milliseconds to 12.34 milliseconds.

Next, using both confidence intervals, a hypothesis test will be conducted using a z-test. This test is documented in the file “z-test.xlsx”, with supporting screenshots provided below:

Tabla

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

The pvalue obtained from the z-test is 0.99928977. This value is greater than alpha, which in this case is 0.05. We can conclude that the changes are not significant enough in terms of system performance.

The results of executing these tests on another group member's computer will now be presented to compare them with the previously reported results:

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

This graph illustrates the performance improvement compared to the previous machine. Although the creation of activity logs remains the most time-consuming task, its execution time has decreased by more than 10 milliseconds. Overall, the response time of all requests has been reduced.

Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Based on this information, we can conclude that the performance analysis results have improved significantly, as demonstrated by the reduction in the confidence interval from (12.24, 14.48) to (8.45, 10.05) milliseconds. This improvement can be attributed to the enhanced processing power of the machine on which the tests were conducted.

# **Conclusion**

More than 95% of the code developed by Student 3 has undergone testing, allowing for the identification and resolution of previously undetected issues. Furthermore, statistical analysis confirms that the features implemented by this student satisfy the non-functional requirement that the system, on average, completes operations involving the Flight Assignment and Activity Log entities in under one second, thereby enhancing the overall user experience.

# **Bibliography**

Intentionally blank