# **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 20, 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 | 20/05/2025 | Intial version – all sections added |

# **Introduction**

The purpose of this document is to provide an analysis of the various tests that have been carried out to assess code coverage related to Student 1, as well as an analysis of the system's performance with respect to the requirements associated with that student.

# **Contents**

## **Functional Testing**

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

* List-show.safe: This test checks the functionality of listing the flights of a manager and displays the specific content of all these entities belonging to manager1, testing the show functionality of the entity.
* Create.safe: Several flights are created using valid test data accepted by the system. An attempt is made to create a flight without providing any information (returning the corresponding error messages), and finally, some valid data is added along with various invalid values to test fields of type Money (negative values) and String (overly long messages).
* Update.safe: Data from several flights belonging to manager1 are updated using both valid inputs accepted by the system and invalid inputs, similarly to the tests in create.safe.
* Delete.safe: All flights of manager2 are removed, and it is verified that the entities related to those flights are also deleted.
* Publish.safe: Attempts are made publish all unpublished flights of manager2. Some are successfully published, while others return the appropriate error as they are not in a valid state to be published.
* Ids.hack: Tests are conducted using invalid IDs in the URL, such as 9999999, 0000000, or an empty character, expecting the corresponding error. This is tested across multiple operations, including show, update, delete, and publish.
* List-show.hack: GET hacking tests are performed by accessing URLs associated with managers from other users and from a non logged in user. Attempts are also made to access both unpublished and published flights of other managers while logged in as one of them.
* Create-update.hack: POST hacking tests are conducted by modifying values (relationship Ids like Airline) with F12, and attempts are made to create and update a flight with these modified values, returning the expected error response.

The test coverage achieved for the entity Flight is 98%.

Texto

El contenido generado por IA puede ser incorrecto.

All lines of this entity have been tested either fully or partially, except for one line in the Delete functionality. This is because that specific line is responsible for deleting all Bookings associated with the corresponding flight. However, this action is not strictly necessary, as there should not be any bookings linked to a flight that can be deleted, since such a flight would not be published. Nonetheless, the line has been included as a precautionary measure, given that most of the project is developed individually and it is not possible to fully verify the extent to which the student responsible for this entity has implemented the functionality correctly. The line is therefore included to prevent potential errors when deleting a flight.

Texto

El contenido generado por IA puede ser incorrecto.

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

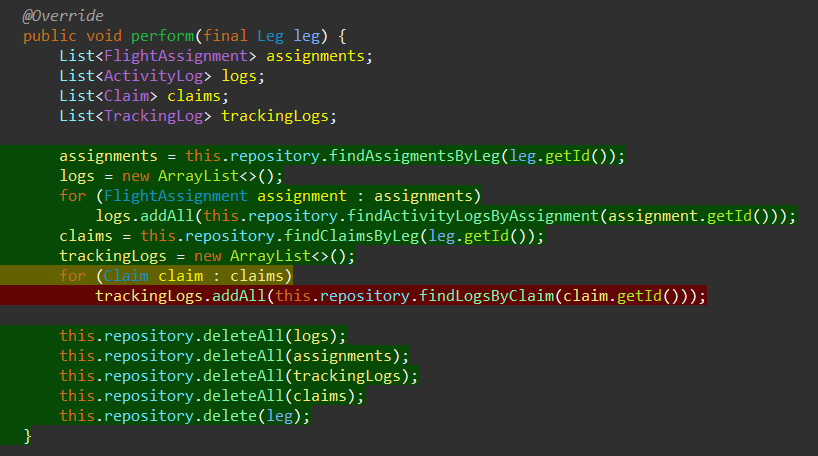
* List-show.safe: This test checks the functionality of listing the legs of a manager and displays the specific content of all these entities belonging to manager1, testing the show functionality of the entity.
* Create.safe: Several legs are created using valid test data accepted by the system. An attempt is made to create a leg without providing any information (returning the corresponding error messages), and finally, some valid data is added along with various invalid values to test fields of type Double (negative values of duration) and triying to add a duplicate flight number (returning the corresponding error message).
* Update.safe: Data from several legs belonging to manager1 are updated using both valid inputs accepted by the system and invalid inputs, similarly to the tests in create.safe.
* Delete.safe: All legs of manager2 are removed, and it is verified that the entities related to those legs are also deleted.
* Publish.safe: Attempts are made publish all unpublished legs of manager2.
* Ids.hack: Tests are conducted using invalid IDs in the URL, such as 9999999, 0000000, or an empty character, expecting the corresponding error. This is tested across multiple operations, including show, update, delete, and publish.
* List-show.hack: GET hacking tests are performed by accessing URLs associated with managers from other users and from a non logged in user. Attempts are also made to access both unpublished and published legs of other managers while logged in as one of them.
* Create-update.hack: POST hacking tests are conducted by modifying values (relationship Ids like Aircraft and Flight) with F12, and attempts are made to create and update a leg with these modified values, returning the expected error response.

The test coverage achieved for the entity Leg is 99.2%.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

All lines of this entity have been tested either fully or partially, except for one line in the Delete functionality with the same meaning of the entity Flight previously mentioned.



## **Performance Testing**

A performance analysis of the system will now be carried out through the execution of the previously mentioned functional tests. The tests have been executed under two different 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.

As observed, the most time-consuming operation on average is the list of flights, which takes 100 milliseconds in average.

Below are some statistics regarding the operations:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

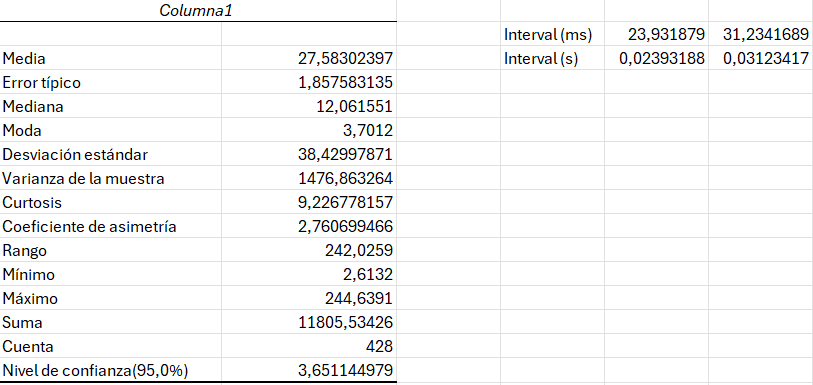
El contenido generado por IA puede ser incorrecto.

As we can observe, the system without indexes gives a confidence interval ranging from 24 milliseconds to 31.5 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.

As we can see, the most time consuming operation on average is still the flight listing, which takes less than 95 milliseconds in average. All operations have decreased their response time compared to the previous analysis except of the update of a leg. However, there doesn't appear to be a significant change in their performance.

Below are some statistics regarding the operations:



As we can observe, the system with indexes gives a confidence interval ranging from 23.9 milliseconds to 31.2 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:

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

The pvalue obtained from the z-test is 0.934762193. 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:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

As we can see, the performance analysis results have significantly improved, as evidenced by the reduction in the confidence interval from (23.9, 31.2) to (9.2, 12.2) milliseconds. This improvement is due to the increased processing power of the machine on which the tests were executed. As observed, the flight listing operation continues to be the most time consuming among the operations analyzed, with its average response time being more than twice the average of the other operations.

# **Conclusion**

Over 90% of the code related to Student 1 has been tested, enabling the detection and correction of bugs that were not initially apparent. Additionally, it has been statistically demonstrated that the requirements implemented by this student meet the non-functional requirement that the system, on average, takes less than one second to perform operations on the entities Leg and Fligh, improving the user experience when interacting with the system.

# **Bibliography**

Intentionally blank