**Analysis Report**

**Group**: C1.027  
**Repository**: <https://github.com/DP2-C1-027/AirNav-Logistics.git>

### **Group Members:**

1. **Garcia de Tejada Delgado, Jose**
   * Role: Developer
   * Email: josgardel8@alum.us.es
2. **Peñaloza Friqui, Nora**
   * Role: Developer
   * Email: norpennfri@alum.us.es
3. **Niza Cobo, Manuel Jesús**
   * Role: Manager, Developer
   * Email: mannizcob@alum.us.es
4. **Gomez Claraco, Nicolas**
   * Role: Developer
   * Email: nicgomcla@alum.us.es
5. **Campos Diez, Lucia**
   * Role: Tester, Developer
   * Email: luccamdie@alum.us.es

03/12/2025

**Table of Contents**

1. **Executive Summary**
2. **Revision Table**
3. **Introduction**
4. **Contents**
   1. **Analysis of the requirements of D01**
   2. **Analysis of the requirements of D02**
   3. **Analysis of the requirements of D03**
   4. **Analysis of the requirements of D04**
5. **Conclusions**
6. **Bibliography**

**Executive Summary**

This document provides an overview of the version report for the AirNav-Logistics project, summarizing key updates and adjustments made based on the analysis of project requirements. While most tasks were straightforward and didn’t require extensive evaluation, some modifications were necessary to improve the system's accuracy and alignment with real-world scenarios.

Notable updates include the implementation of attribute validation to ensure data consistency and accuracy, along with adjustments to the data model to better represent the relationships between key entities. These changes enhance the system’s integrity and functionality, ensuring it meets the specified requirements.

The report underscores the importance of aligning the system structure with the project's domain needs, while maintaining consistency in data validation and entity relationships.

**Revision Table**

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| 1.0 | 12/03/2025 | Initial report for D02 |
| 2.0 | 26/05/2025 | Last deliverables |

**Introduction**

The purpose of this document is to present an analysis of the tasks performed in the AirNav-Logistics project. However, given that the assigned tasks were straightforward and did not require significant analytical work, no substantial analyses were necessary.

**Contents**

* 1. **Analysis of the requirements of D01:**

No specific analysis records are included in this report as the individual tasks completed did not necessitate detailed requirement evaluations. Therefore, no requirement modifications or validations were needed.

* 1. **Analysis of the requirements of D02**

Upon consideration, the duration property of Legs was designed as a persisted value in the database which will be derived from the departure and arrival dates of the leg, instead of being decided by the user or being a transient property. This was decided upon consulting with the professor on the matter

* 1. **Analysis of the requirements of D03**

It was noted that requirements #8 and #9 share an inconsistency: Flights must be possible to delete up until they are published, but a requirement for publishing them is having at least one published leg. Legs must not be deleted when already published. In deciding on a solution to this and consulting with the professor, the final decision taken was to restrict the deletion for flights with already published legs, as legs should only be getting published when the entire conglomerate is ready to be published. This is a compromise that doesn’t directly contradict one of the requisites explicitly (legs can’t be deleted after being published) even if it somewhat restricts the deletion of flights in special cases (which also goes against the spirit of the requisites, but not the exact wording, and said inconsistency must be solved somehow)

For requirement #8, managers must be able to show the details on their flights. As some of those details are transient (and based on the associated legs) such as departure and arrival dates and cities and the number of layovers, this information is shown as a readonly attribute. But because we must have associated legs, and non-published flights don’t necessarily have associated legs, we find a situation in which no data can be shown (as all the values are null). To fix this awkward situation of showing blank fields to the user, the show feature has been implemented in such a way that only if there’s at least one associated leg will there be any of the transient fields shown. If there’s no associated legs yet, said fields just don’t get rendered, as an attempt to make the app easier to understand and avoid misconceptions. As soon as 1 or more legs are associated to the flight, all data is properly shown upon requesting it.

**Conclusions**

Some decisions have been logged here as they pertained to the interpretation and resolution of conflicts within the requisites, or non-exact wording that left details up to interpretation.

**Bibliography**

Intentionally blank.