

Group: C1.010

García Rivero, Andrés Francisco -[andgarriv@alum.us.es](mailto:andgarriv@alum.us.es)

López Cubiles, Antonio José - [antlopcub@alum.us.es](mailto:antlopcub@alum.us.es)

Nieto Vicioso, Javier - [javnievic@alum.us.es](mailto:javnievic@alum.us.es)

Díaz Ordoñez, Pablo -  [pabdiaord@alum.us.es](mailto:%20pabdiaord@alum.us.es)

Sevillano Barea, Alejandro - [alesevbar@alum.us.es](mailto:alesevbar@alum.us.es)

March 8th, 2024

Analysis Report

**Repository: https://github.com/C1-010/Acme-SF-D02**

Contenido

[1. Executive Summary 2](#_Toc158936944)

[2. Revision Table 2](#_Toc158936945)

[3. Introduction 2](#_Toc158936946)

[4. Contents 2](#_Toc158936949)

4.1. Information requirements……………………………………………………………………………………..2

4.2. Functional requirements……………………………………………………………………………………….2

4.3. Non-functional requirements………………………………………………………………………………..2

4.4. Testing requirements……………………………………………………………………………………..……..2

4.5. Managerial requirements……………………………………………………………………………….……. 3

[5. Conclusions 3](#_Toc158936950)

[6. Bibliography 3](#_Toc158936951)

# 1. Executive Summary

This document will gather an analysis of the most complex requirements included in this deliverable. Within the analysis of each requirement, the requirement itself will be quoted verbatim, and the decisions made to fulfil said requirement will be included.

# 2. Revision Table

|  |  |  |
| --- | --- | --- |
| Revision number | Date | Description |
| 1 | 08/03/2024 | Document created and completed |

# 3. Introduction

# The purpose of this document is to perform an analysis of the requirements deemed necessary, providing a detailed description of each. Additionally, different available options will be identified, if any, to address each requirement, along with the associated advantages and disadvantages of each. Finally, the chosen alternative will be presented.

# 4. Contents

**4.1. Information requirements**

**Mandatory**

*2) Code audits are essential pieces to ensure the quality of a project. The system must store the following data about them: a code (pattern “[A-Z]{1,3}-[0-9]{3}”, not blank, unique), an execution date (in the past), a type (“Static”, “Dynamic”), a list of proposed corrective actions (not blank, shorter than 101 characters), a mark (computed as the mode of the marks in the corresponding auditing records; ties must be broken arbitrarily if necessary), and an optional link with further information.*

*3) The result of each code audit is based on the analysis of their audit records. The system must store the following data about them: a code (pattern “AU-[0-9]{4}-[0-9]{3}”, not blank, unique), the period during which the subject was audited (in the past, at least one hour long), a mark (“A+”, “A”, “B”, “C”, “F”, or “F-”), and an optional link with further information.*

*4) The system must handle auditor dashboards with the following data: total number of code audits for “Static” and “Dynamic” types; average, deviation, minimum, and maximum number of audit records in their audits; average, deviation, minimum, and maximum time of the period lengths in their audit records.*

Requirements 2 and 3 are quite similar since they involve creating two entities with their respective relationships and attributes, some of which, such as mark or period, will be developed later in subsequent installments.

Requirement 4 was to create an auditor dashboard, which was easy since I was able to guide myself through the transparency.

**Supplementary**

*13) There is a new project-specific role called auditor, which has the following profile data: firm (not blank, shorter than 76 characters), professional ID (not blank, shorter than 26 characters), a list of certifications (not blank, shorter than 101 characters), and an optional link with further information.*

It has been an easy requirement to create since it has been the same as if I created an entity but instead of the entities folder in the roles folder.

**4.2.** **Functional requirements**

Intentionally blank.

**4.3. Non-functional requirements**

Intentionally blank.

.**4.4. Testing requirements**

**Mandatory**

1. Produce assorted sample data to test your application informally. The data must include two **auditor** accounts with credentials “**auditor1/auditor1**” and “**auditor2/auditor2**”.

I have had to create an auditor csv giving values to the attributes previously created for the auditor role.

**4.5. Managerial requirements**

**Supplementary**

1. Produce a UML domain model.
2. Produce an analysis report.
3. Produce a planning and progress report.

# Having already created the "analysis report" and "planning and progress report" templates in deliverable 1, I have developed them quite quickly using those templates. In the UML using UMLET it is much easier but it did take me a little longer than expected by having to create several entities and the form with their respective attributes.

# 5. Conclusions

# In conclusion, the analysis of the requirements for this deliverable has been simple but it took me longer on the individual tasks than on the group tasks that I had assigned. For some other requirement I have had to search the forum for information.

# 6. Bibliography

Intentionally blank.