Insertando imagen...

ACME-SF

G1.007

**Analysis report**

08/03/2024



# Cover

|  |  |
| --- | --- |
| **Repository:** <https://github.com/DP2-C1-07/Acme-SF-D02.git> | |
| **Student #1**  **ID:** 31878881F  **UVUS:**  pabcabmar3  **Name:** Caballero María, Pablo **Roles:** manager, developer | **Student #2**  **ID Number:**49034820Q  **UVUS:** mararnmon  **Name:** Arnáiz Montero, Marco Antonio  **Roles:** developer, operator |
| **Student #3**  **ID Number:** 77865211E  **UVUS:**  alfalolan  **Name:** Alonso Lanzarán, Alfonso Luis  **Roles:** developer, tester | **Student #4**  **ID Number:** 53932912M  **UVUS:** albsanmim  **Name:** Sánchez Mimbrero, Alberto  **Roles:** developer |
| **Student #5**  **ID Number:** 48123111G  **UVUS:** juagarcar4  **Name:** Garcia Carballo, Juan  **Roles:** developer |  |

Table of contents

[Cover 1](#_Toc2095998540)

[Executive summary 2](#_Toc1985786661)

[Revision table 2](#_Toc1937373377)

[Introduction 2](#_Toc1407709460)

[Contents 3](#_Toc535143832)

[Conclusions 3](#_Toc840296917)

[Bibliography 3](#_Toc578232771)

# Executive summary

This document is an Analysis report, in where the requirements for a certain deliverable are analyzed in case there is ambiguity, or they are not clear. In that case, they will be discussed with the corresponding lecturer, through the EV platform. Otherwise, the “content” section will show the text “intentionally blank”.

# Revision table

|  |  |  |
| --- | --- | --- |
| Number | Date(dd/mm/yyyy) | Description |
| 1.0 | 01/03/2024 | Document done in its entirety, reviewed by peers. No major errors were found. |

# Introduction

The purpose of this document is to provide a detailed analysis report of the functional requirements of the first delivery of the project “Acme-SF-D01”, for the subject “Design and Testing 2”. More specifically, this document conforms to the requirements of the student #1, #2, #3, #4 and #5

This document has an executive summary, a revision table where there are versioned records, this introduction, the main content, and the conclusions of the report.

# Contents

**Student #1:** During the development of the mandatory requirement D02-G01 (create “Claim” entity), a question about whether the entity should have a property to relate it with another one arose. Therefore, I posted a new thread in the “On your tutorials” forum, which was satisfactorily answered: the entity should NOT be related to another one, since the traceability of the creator of the claim was implied in the “email” field, and all the unauthenticated users shared the same account, and no further actions were required.

**Student#3:** During the development of the mandatory requirement D02-G04 (create “Objective” entity), a question about how to implement the duration attribute. Therefore, I posted a new thread in the “On your tutorials” forum, which was satisfactorily answered: the attribute must be implemented with two Date attributes.

Student #5. Information Requirements 2: Code Audit.

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.

* Problem presented.

Implement the "mark" attribute, as it appears to be derived from another entity of a second mandatory requirement, and would need more complex validation, since they ask for a tiebreaker.

* Proposed Solutions:

1. Compute “mark” through a @OneToMany association to the record entity along with a custom method.

This would allow us to calculate the attribute as a derived property. In exchange we would use the @OneToMany association, which is not recommended by the subject criteria.

1. Calculate the attribute “mark” in the service layer of the application.

This would allow us to eliminate the @OneToMany implementation, which is not recommended for use in the context of the subject. In exchange, it would be necessary to postpone the implementation of the correct functionality of the "mark" attribute to the D03 derivable.

* Selected alternative:

Following the instructions given in the "Tutorials" forum, we will decide to opt for the second alternative and the logic will be implemented to calculate the "mark" attribute in the services layer in the following deliverable (D03)

Link to the forum where the problem was discussed: <https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board_entry&conf_id=_405265_1&forum_id=_234042_1&message_id=_405313_1>

NOTE: This conversation was not carried out by any of the members of the group but has been taken into account to select an alternative.

 Information Requirements 3: Audit Record.

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.

* Problem presented.

Implementing the duration attribute offers a problem and that is that it can be understood that it is a numerical value that indicates how much time is assigned to that objective, however, since it mentions that said duration must begin after the moment of instantiation, we cannot create it as a duration attribute. type "int", it would be necessary to create it as an attribute of type "Date".

* Proposed Solutions:

1. Implement the attribute “period” as 2 different attributes of type "Date", one when the period begins and another when it ends.

Thanks to this implementation, it can be compared with the instantiation moment attribute to verify that it is later and does not require the creation of a custom data type. However, this implementation would require the creation of 2 different attributes when it could be solved with just 1 attribute.

1. Use a custom data type that has as attributes an end time and a derived attribute of type double that is the difference in hours between the instantiation time and the end time.

In one hand, it would fulfill the meaning of the word duration, an amount of time, and it can also be compared with the instantiation moment attribute to verify that it is later.

On the other hand, it would be necessary to create a custom data type, increasing the complexity of the solution.

* Selected alternative:

Following the instructions given in the "Tutorials" forum, we will decide to opt for the first alternative, It is a simple and direct solution whose only drawback is that it is not reusable, as a datatype would be. But it's not the right time now to implement it.

Link to the forum where the problem was discussed: <https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board&conf_id=_405265_1&forum_id=_234042_1&message_id=_403018_1>

NOTE: This conversation was not carried out by any of the members of the group but has been taken into account to select an alternative.

# Conclusions

The development of the requirements followed its course with no further inquiries.

# Bibliography

<https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board_entry&conf_id=_405265_1&forum_id=_234042_1&message_id=_404965_1>

https://ev.us.es/webapps/discussionboard/do/message?action=list\_messages&course\_id=\_85092\_1&nav=discussion\_board\_entry&conf\_id=\_405265\_1&forum\_id=\_234042\_1&message\_id=\_400205\_1