DP2 2023-2024
Analysis Report

Acme Software Factory

Repository: https://github.com/rafcasceb/Acme-SF-D03

Student #5:

- Vento Conesa, Adriana adrvencon@alum.us.es

Other members:

- Castillo Cebolla, Rafael rafcasceb@alum.us.es
- Flores de Francisco, Daniel danflode@alum.us.es
- Heras Pérez, Raúl rauherper@alum.us.es
- Mellado Díaz, Luis luimeldia@alum.us.es

GROUP C1.049

Version 1.0

15-04-24

Content Table

Abstract

This report encompasses a detailed description of each task undertaken by student five for deliverable D03 of the project. As noted in the annexes, not all tasks require commentary, emphasizing selective analysis where necessary. This analysis is particularly valuable in situations where the client possesses limited technological expertise.

Revision Table

Date	Version	Description of the changes	Deliverable
15/04/2024	V1	• Abstract. • Introduction. • Contents: task detailing. • Conclusions.	3

Introduction

In this third phase of delivery, our focus is on meeting a set of obligatory and supplementary criteria. The obligatory segment comprises three functional requirements, while the supplementary scope includes three additional functional requirements and three managerial requirements.

The document structure unfolds as follows: an introductory section that sets the context of the report, followed by an analysis of each task in the contents section, where applicable. Finally, a concise conclusion will wrap up the report, summarizing key findings and outlining next steps.

Contents

Mandatory Deliverables

Functional requirements:

6. Operations by **auditors** on **code audits**:
    - List the **code audits** that they have created.
    - Show the details of their **code audits**.
    - Create, update, or delete their **code audits**. **Code audits** can be updated or deleted as long as they have not been published. For a **code audit** to be published, the mark must be, at least, "C".

    The listed points were implemented in a straightforward manner. There were, however, a few design considerations to be taken into account:

    1. The listing, as explained in theory classes, does not include all attributes regarding code audits. Thus, this applies to the rest of requirements listed in this document.
    2. In the details of the code audit the attribute "published" was not displayed, as it is only relevant to the user for functional purposes.
    3. As it holds a conglomerate relationship with audit records (see functional requirement seven), particularly a composition, a delete cascade was applied for all the "children" entries of the parent.
    4. Branching from the previous point, it was also necessary to determine whether the "children" entries of a code audit would have to be published for a code audit to be published as well.

        This question was proposed in the forum and the conclusion was that the children entries of a code audit would have to be published for a code audit to be published as well.

        Link to the validation: The post can be found on the discussion board, under the title "[Análisis] D03 Student#5-06-07" Created by Adriana Vento Conesa.

    5. The attribute "Mark" described in the previous delivery (which, as explained in the previous analysis document, could not be implemented directly in the entity) was computed in the service. An auxiliary class was used for this task. This allowed to

implement the remark: “For a **code audit** to be published, the mark must be, at least, “C”.”

7. Operations by **auditors** on **audit records**:
    - List the audit records in their **code audits**.
    - Show the details of their **audit records**.
    - Create and publish an **audit record**.
    - Update or delete an **audit record** as long as it is not published.

It must be noted that this requirement introduces a new information requirement to audit records: similarly to code audits, they must be able to be published. This, analogous to what was implemented in this delivery, meant the inclusion of a Boolean attribute “published” in the entity of this requirement. This wasn’t inferred from the previous delivery’s requirement, which is why we remark it now. This requirement will, however, not appear in the representation of the entity, as it is only for functional purposes.

The listed points were implemented in a straightforward manner. There were, however, a few design considerations to be taken into account:

1. Due to the relationship between this entity and code audits (see functional requirement six) it was necessary to evaluate whether a code audit being published would forbid a user from creating further audit records for it.

   This question was proposed in the forum and the conclusion was, although not explicitly stated, inferred from the response: it would not be possible to create further audit records in an already published code audit.

   Link to the validation: The post can be found on the discussion board, under the title “[Análisis] D03 Student#5-06-07” Created by Adriana Vento Conesa.

2. A separate section for the audit records created by an auditor was created, although not explicitly stated. This was done in order to offer a more complete implementation.

8. Operations by **auditors** on **auditor** dashboards:
    - Show their **auditor** dashboards.

Given the self-explanatory nature of the requirement, further analysis was deemed unnecessary.

Optional Deliverables

Functional requirements:

17. Operations by anonymous principals on user accounts:
    - Sign up to the system and become an **auditor**.

    Given the self-explanatory nature of the requirement, further analysis was deemed unnecessary.

18. Operations by **auditors** on user accounts:
    - Update their profiles.

    Given the self-explanatory nature of the requirement, further analysis was deemed unnecessary.

19. Operations by any principals on **code audits**:
    - List the **code audits** in the system that are published.
    - Show the details of the **code audits** that they can list (including their **audit records**).

    Similarly to previous requirements, the attribute published wasn't included in the form. In this case, it wasn't included in the listing either, as it is implicit that all shown code audits and therefore, audit records, must be published.

Managerial requirements:

20. Produce an analysis report.

    This report constitutes the task under examination; thus, an in-depth analysis wasn't deemed necessary.

21. Produce a planning and progress report.

    Given the self-explanatory nature of the requirement, further analysis was deemed unnecessary.

22. Produce a lint report.

    Given the self-explanatory nature of the requirement, further analysis was deemed unnecessary.

Conclusions

In conclusion, the document underscores the complexities and challenges encountered during the analysis and implementation of various requirements. Throughout the process, appropriate consultations were conducted with the client to address and amend requirements as necessary, ensuring alignment with project objectives and client expectations.

Bibliography

Intentionally blank.