DP2 2023-2024

Analysis Report

Acme Software Factory

Repository: https://github.com/rafcasceb/Acme-SF-D02

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

02-03-24

Content Table

Abstract

This report encompasses a detailed description of each task undertaken by student five for deliverable D02 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
02/03/2024	V1	• Abstract. • Introduction. • Contents: task detailing. • Conclusions.	2

Introduction

In this second delivery phase, our attention is divided into fulfilling a set of obligatory and optional requirements. The obligatory segment encompasses three information requirements and one testing requirement. Concurrently, within the optional scope, there is one additional information requirement and three managerial requirements. Given the diverse nature of these tasks, particularly of the obligatory information requirements, a meticulous analysis was deemed necessary. Consequently, we required consultations with the client to gain deeper insights into their expectations.

The document unfolds in the following manner: this introductory section establishes the context of the report, followed by an analysis of each task in the contents section, if appropriate. Lastly, a brief conclusion will conclude the report.

Contents

Mandatory Deliverables

Information requirements:

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.

   Refer to analysis notes in task three.

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.

   Requirement three, along with requirement two, raised several uncertainties due to its incomplete specification in various aspects.

   The first aspect, independent of requirement two, solely pertained to requirement three. The implementation of the "period" attribute lacked clarity as it could be interpreted and implemented in various ways. Moreover, enforcing the "at least one hour long" constraint using standard entity constraints proved unfeasible, necessitating its postponement for implementation in the corresponding service. This challenge extended to several other entities. Consequently, when I intended to initiate a consultation on the discussion board, I discovered that another student had already raised a question very similar to mine. The final recommendation to the issue was to create two attributes that represented this period, alongside an added set of (advanced) constraints. This attribute was implemented as such, and the implementation of the advanced constraints were postponed.

   Link to the validation: The post can be found on the discussion board, under the title “Ambigüedad en un requisito de información” Created by Alfonso Luis Alonso Lanzarán.

   The second aspect was regarding the implementation of the “mark” attribute (a **mark**, computed as the mode of the marks in the corresponding auditing records; ties must be broken arbitrarily if necessary) in requirement two. This logic was too complex to be implemented in the entity without a @OneToMany relationship with the entity of requirement 3, which is heavily discouraged by the good practices of the course. Therefore, a post in the discussion board was created. The conclusions drawn indicated that the relationship between the entities was that of a conglomerate. The recommendation was to establish this relationship for the time being, as the computation of this derived attribute would be covered in subsequent lectures. Additionally, the requirement was amended to specify that the tie must be broken in favor of the smaller mark.

Link to the validation: The post can be found on the discussion board, under the title “[Análisis] D02-Student#5 - 002” Created by Adriana Vento Conesa.

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**.

   Considering the self-explanatory nature of the requirement, additional analysis was deemed unnecessary. We followed the guidelines provided in the course to execute the task.

Testing requirements:

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

   Considering the self-explanatory nature of the requirement, additional analysis was deemed unnecessary. We followed the guidelines provided in the course to execute the task.

Optional Deliverables

Information requirements:

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.

    While each attribute is inherently self-explanatory, the "optional link" attribute sparked debate due to the absence of a specified maximum limit from the client's requirements, among other attributes. Consequently, this issue, which also impacted group tasks, prompted one of our team members to seek clarification via the forum. After thorough discussion, it was collectively agreed that the maximum length of a link should be set at 255 characters, among other established limits. These conclusions apply as well to requirements two and three as needed.

Link to the validation: The post can be found on the discussion board, under the title “[Análisis] Rangos de atributos no especificados” Created by Rafael Castillo Cebolla.

Managerial requirements:

14. Produce a UML domain model.

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

15. Produce an analysis report.

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

16. Produce a planning and progress 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.