

DP2 2023-2024
Analysis report D02

Acme Software Factory



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

Student #1:

- Castillo Cebolla, Rafael rafcasceb@alum.us.es

Other members:

- Flores de Francisco, Daniel danflode@alum.us.es
- Heras Pérez, Raúl rauherper@alum.us.es
- Mellado Díaz, Luis luimeldia@alum.us.es
- Vento Conesa, Adriana adrvencon@alum.us.es

GROUP C1.049

Version 1.0

28-02-24

Content Table

Executive summary	3
Introduction	5
Contents	6
Mandatory task D02-2:	6
Mandatory task D02-3:	7
Mandatory task D02-4:	7
Mandatory task D02-5:	7
Extra task D02-13:	8
Extra task D01-14:	8
Extra task D01-15:	8
Extra task D01-16:	8
Conclusions	9
Bibliography	10

Executive summary

This report will offer a detailed description of those individual task of the second delivery which are deemed in need of further analysis. As stated in the Annexes document, not all tasks are worthy of such distinction; however, the most complex ones may need it to provide an extra clarification to the client and those stakeholders with limited knowledge on the specific matter.

We will follow a precise but accessible approach aiming to promote comprehension and assure a good final product.

Revision Table

Date	Version	Description of the changes	Sprint
28/02/2024	1.0	<ul style="list-style-type: none">• Executive summary• Introduction• Content• Conclusion• Bibliography	2
05/03/2024	1.1	<ul style="list-style-type: none">• Requirements have been reclarified.	2

Introduction

The second delivery comprises eight tasks in total, with four categorized as mandatory and four as optional. Four of these entail coding entities and forms, one is for populating with sample data and the subsequent tasks involve writing reports and a UML diagram.

The document's structure will present for each delivery first the mandatory tasks followed by the optional ones. Each task will be identified by its numerical identifier, accompanied by its official description, and a comprehensive analysis, if deemed necessary, as stated in the Annexes document.

Contents

Mandatory task D02-2:

“A project aggregates several user stories elicited by the same manager. The system must store the following data about them: a **code** (pattern “[A-Z]{3}-[0-9]{4}”, not blank, unique), a **title** (not blank, shorter than 76 characters), an **abstract** (not blank, shorter than 101 characters), an indication on whether it has **fatal** errors, e.g., panics, a **cost** (positive or nought), and an optional **link** with further information. Projects containing fatal errors must be rejected by the system.”

It is worthy of being noted that unless otherwise specified by the word “optional”, all attributes will make use of the @NotNull annotation.

As discussed in: [\[Análisis\] Rangos de atributos no especificados](#) by Rafael Castillo, URLs we will include a maximum length of 255 characters.

A unidirectional ManyToOne relationship with Projects has been implemented in the UserStory entity. To avoid bidirectionality, as we were encouraged, we haven't included a OneToMany in Projects. This will, however, make that we cannot validate in the entity that the projects must have at least two user stories, as the statement says. This will be left for the service. It is important to remark that one user story can not have any user story as long as it is not published (not in draft mode anymore), a validation which must be left for the service.

Another required aspect we cannot implement in this delivery since we shall not use services yet, is that one project can only have user stories of the same manager. We will probably implement it through the User Story service, which owns the relationship, by making sure a user story is not linked to a project with user stories of a different manager.

As discussed in [\[Análisis\] D02-Student#1-002](#) by Jaime Linares, the cost of the project will be predefined and will not derived from the cost of its user stories. However, it must be noted that, as can be read in the same post, this cost can be considered as cost in hours as well.

An important contradiction seemed to be found: the statement asked us to store whether a project contained fatal errors and at the same time we were told that the system should directly reject projects with fatal errors. However, it was, at the end, just luck of context and a bit misleading: indeed a project could be stored having fatal errors, but it could not be published, which is something not mentioned until the next delivery. For this, we contacted our clients (see thread [“\[Análisis\] D02-Student#1-002”](#) by Rafael Castillo). We also agreed on renaming the attribute that indicates if the aggregation is in draft mode in a way closer to the client's terminology.

Since we were encouraged to set limits even when they were not explicit, we consulted about the different ranges in [\[Análisis\] Rangos de atributos no especificados](#). Therefore, I decided to consider a maximum value of 2000 for the estimated cost of the project, based on the decision taken for the user stories.

Mandatory task D02-3:

"A user story is a document that a manager uses to represent the smallest unit of work in a project. The system must store the following data about them: a **title** (not blank, shorter than 76 characters), a **description** (not blank, shorter than 101 characters), an estimated **cost** (in hours, positive, not nought), the **acceptance** criteria (not blank, shorter than 101 characters), a **priority** ("Must", "Should", "Could", or "Won't"), and an optional **link** with further information."

For this entity we have implemented two unidirectional ManyToOne relationships, one with Project and the other with Manager. We made both relationships not optional since, according to the statements, a user story must always belong to one project and one manager. To achieve the second relationship we had to implement, if not fully, at least create the basis, the Manager role which will later be modelled in the extra task 15.

Another important aspect to be analysed is the attribute of the estimated cost in hours. It is never specified if hours will accept fractions or not, allowing for values like an a hour and a half. For this, we contacted our clients (see thread [\[Análisis\] D02-Student#1-003](#) of Rafael Castillo). The agreed solution was modelling it as integer.

As discussed in the aforementioned thread and in the following one, [\[Análisis\] Rangos de atributos no especificados](#) by Rafael Castillo, we must add limits to all attributes even when they are not explicit. After the discussion, I decided to consider a maximum of 200 hours for a user story, considering it says that a user story is the smallest unit of work in a project.

Mandatory task D02-4:

"The system must handle manager dashboards with the following data: total number of "must", "should", "could", and "won't" user stories; average, deviation, minimum, and maximum estimated cost of the user stories; average, deviation, minimum, and maximum cost of the projects."

Being this is dashboard we based our solutions on the example of a dashboard given by the framework (Dashboard.java). This meant allowing null and not using any annotation for these, since it will be a form and data will not be directly injected but they will be calculated through the appropriate service. I used Integer for those for which Double didn't make sense.

Mandatory task D02-5:

"Produce assorted sample data to test your application informally. The data must include two manager accounts with credentials "manager1/manager1" and "manager2/manager2"."

Due to its simplicity, it isn't deemed necessary to further analyse it.

Extra task D02-13:

“There is a new project-specific role called manager, which has the following profile data: degree (not blank, shorter than 76 characters), an overview (not blank, shorter than 101 characters), list of certifications (not blank, shorter than 101 characters), and an optional link with further information.”

The list of certifications mentioned in the statement could be misleading at first; however, the specified annotations make it clear that the attribute must be modelled via a String, not as a List from java.util.

Extra task D01-14:

“Produce a UML domain model.”

Due to its simplicity, it isn’t deemed necessary to further analyse it.

Extra task D01-15:

“Produce an analysis report.”

Due to its simplicity, it isn’t deemed necessary to further analyse it.

Extra task D01-16:

“Produce a planning and progress report.”

Due to its simplicity, it isn’t deemed necessary to further analyse it.

Conclusions

For the delivery number two, we have found some aspects in need of further analysis, either important decisions that had to be taken, implicit information, or contradictions or ambiguities, easily solved by discussing with our clients.

Bibliography

Intentionally blank.