

DP2 2023-2024
Analysis report D02

Acme Software Factory



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

Student #2:

- Flores de Francisco, Daniel danflode@alum.us.es

Other members:

- Castillo Cebolla, Rafael rafcasceb@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 1.1

29-02-24

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
29/02/2024	1.0	<ul style="list-style-type: none">• Executive summary• Introduction• Content• Conclusion• Bibliography	2
07/03/2024	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 contract is one or several agreements between the stakeholders involved in the development 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 instantiation moment (in the past), a provider name (not blank, shorter than 76 characters), a customer name (not blank, shorter than 76 characters), some goals (not blank, shorter than 101 characters), and a budget (less than or equal to the corresponding project cost).”

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 and Client has been implemented in the Contract entity. To avoid bidirectionality, as we were encouraged, we haven't included a OneToMany in Projects.

Another required aspect we cannot implement in this delivery since we shall not use services yet, is that one contract can only have progress logs of the same client. We will probably implement it through the Progress Log service.

As discussed in [\[Análisis\] D02-Student#2-002](#) by Daniel Flores, the budget of a contract will be modeled using Money datatype. It must be noted that, as can be read in the previous post ([\[Análisis\] Rangos de atributos no especificados](#)), this budget range goes from 0.00 to 1,000000.0 no matter what currency is. In addition, we can not implement that the budget of a contract must be lower or equal to the corresponding cost of the project associated with it. We left this constraint for services. In the same post, our client aclares to us that Contract entity is a conglomerate composed of Progress Logs.

Mandatory task D02-3:

“Every contract has an evolution that is composed of progress logs. The system must store the following data about them: a record id (pattern “PG-[A-Z]{1,2}-[0-9]{4}”, not blank, unique), a percentage of completeness (positive), a comment on the progress (not blank, shorter than 101 characters), a registration moment (in the past), and a responsible person for the registration (not blank, shorter than 76 characters).”

For this entity we have implemented two unidirectional ManyToOne relationships, one with Contract and the other with Client. To achieve the second relationship we had to implement, if not fully, at least create the basis, the Client role which will later be modelled in the extra task 15.

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.

Another aspect I can not implement yet is the fact that the registration moment should be after the creation of the contract associated with it. I left this for the services as a custom validation.

Mandatory task D02-4:

“The system must handle client dashboards with the following data: total number of progress logs with a completeness rate below 25%, between 25% and 50%, between 50% and 75%, and above 75%; average, deviation, minimum, and maximum budget of the contracts.”

The only remarkable aspect was having to choose between primitive and wrapper types. The calculations will be left for the service.

Mandatory task D02-5:

“Produce assorted sample data to test your application informally. The data must include two client accounts with credentials “client1/client1” and “client2/client2”

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 clients, which has the following profile data: identification (pattern “CLI-[0-9]{4}”, not blank, unique), a company name (not blank, shorter than 76 characters), type (not blank, either “company” or “individual”), an email (not blank), and an optional link with further information.”

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

Extra task D02-14:

“Produce a UML domain model.”

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

Extra task D02-15:

“Produce an analysis report.”

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

Extra task D02-16:

“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.