Sprint 4: Analysis report

Group: C1.04.14

Repository: https://github.com/marizqlav/Acme-L3-D04

Student #1

Name: Domínguez-Adame, Alberto
email: albdomrui@alum.us.es

Student #2

Name: Herrera Ramírez, Ismael
email: ismherram@alum.us.es

Student #3

Name: Olmedo Marín, Marcos
email: marolmmar1@alum.us.es

Student #4

Name: Izquierdo Lavado, Mario
email: marizqlav @alum.us.es

Student #5

Name: Merino Palma, Alejandro
email: alemerpal@alum.us.es

Table of contents

Summary

Acme Life-Long Learning, Inc. (Acme L3, Inc. for short) is a company that specializes in helping learners get started on a variety of matters with the help of renowned lecturers. The goal of this project is to develop a WIS to help this organization manage their business.

Revision table

Number	Date	Description
1	24/05/2023	Full redaction of the document

<u>Introduction</u>

This document lists analysis records, each one including the following data: a verbatim copy of the requirement to which the record refers; detailed conclusions from the analysis and decisions made to mend the requirement; a link to the validation performed by a lecturer.

This document has the following structure:

- Contents
    - Analysis records 2
    - Analysis records 3
    - Analysis records 4
- Conclusion
- Bibliography

Contents

Analysis records 2

Information Requirement nº7 System Configuration:

Decisions: The problem we have in this requirement was to implement the pattern of the attribute systemCurrency, but it was solved by attending laboratory classes.

Conclusion: To reach this conclusion, we have discussed it among ourselves and then consulted the professor of the laboratory class.

Information Requirement nº10 Offer:

Decisions: I have also had problems implementing the availabilityPeriod attribute, which through a look in the forum I managed to understand that I had to separate it into two attributes, a start date and an end date which I called availabilityPeriodInitial and availabilityPeriodFinal, and taking advantage of the MomentHelper class to do the calculation as a non-persistent attribute, so use the @Transient annotation. In addition, the post also clarifies that the statement says that it can be both in the past and in the future, so the simplest solution was not to put restrictions:

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages &course_id=_63009_1&nav=discussion_board&conf_id=_303964_1&forum_id=_206 215_1&message_id=_358091_1

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages &course_id=_63009_1&nav=discussion_board&conf_id=_303964_1&forum_id=_206 215_1&message_id=_358575_1

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages &course_id=_63009_1&nav=discussion_board&conf_id=_303964_1&forum_id=_206 215_1&message_id=_359274_1

Conclusion: With the help of the previous posts in the subject forum, I have been able to implement the timePeriod attribute in the correct way.

Information Requirement nº13 Administrator Dashboard:

Decisions: To simplify the class attributes, we have decided to use maps, in the case of statisticsOfTheBudgetInOffersPerCurrency, we have decided to use a map in which the key will be the Currency name and the value will be a map with the statistics and their value

Conclusion: To reach this conclusion, we have discussed it among ourselves and then consulted the professor of the laboratory class, who told us that it was valid.

Analysis records 3

Requirement nº18 System Configuration:

Decisions: For this requirement it has been implemented so that only currencies accepted by the API can be added. You can also change the system currency as long as it is in the system currency list.

Conclusion: In conclusion, this requirement has been met by listening in class.

Requirement nº19 Operations by any principals on peeps:

Decisions: This requirement was very clear, therefore there has been no doubt and it has been carried out in accordance with what the statement asks for.

Conclusion: No remarkable conclusion, since there have been no doubts.

Requirement nº20 Operations by authenticated principals on bulletins:

Decisions: This requirement has not been finished because there has not been enough time to fully implement its functionality and views.

Conclusion: In conclusion, it has not been counted as done for this deliverable and is expected to be finished for the next deliverable.

Requirement nº21 Operations by administrators on bulletins:

Decisions: This requirement has not been finished because there has not been enough time to fully implement its functionality and views.

Conclusion: In conclusion, it has not been counted as done for this deliverable and is expected to be finished for the next deliverable.

Requirement nº22 Operations by authenticated principals on offers:

Decisions: This requirement was very clear, so there has been no doubt and it has been carried out in accordance with what the statement requests, therefore there were no doubts when implementing an auxiliary class for the price attribute of type of money that had the type of currency that it was along with the amount of money it has.

Conclusion: In conclusion, I found it necessary to add a helper class so that they can only be added for all project prizes that have the types of coins that are registered in the system configuration along with their amount of money.

Requirement nº23 Operations by administrators on offers:

Decisions: This requirement was very clear, therefore there has been no doubt and it has been carried out in accordance with what the statement asks for.

Conclusion: No remarkable conclusion, since there have been no doubts.

Requirement nº24 Operations by authenticated principals on notes:

Decisions: The only decision to be made was how to implement the author property.

1.- Having the author field be an input text-box and then check that the text follows the appropriate pattern.

2.- Get the value of the author through a method in the service that calls the necessary data (username, name and surname) and then make the input field read only.

Conclusion: The second option is easier and more intuitive, rather than having the user memorize its data in order to write it.

Requirement nº25 Display banners randomly as little intrusively as posible:

Decisions: For this requirement it was specified in class that we shouldn't bring all banners to memory at the same time. Also, their data must be present at every screen. So we tried different approaches:

- We used an auxiliary class for loading banners information in a .txt file inside the resources folder through java's file writer so we can retrieve it on every screen using a js script. This was the first and more clunky approach because it was devised before we were explained how to pass this information properly. I'm still proud of the complex system I designed however.
- We used a controller advice to pass the banner info and retrieve it in jstl view. However, we should only retrieve banners with their display period active. This condition cannot be easily tested with a jsql consult so we had to go another way. We thought there were three approaches:
  - Load all banners in memory, which we were no to do.
  - Load just one banner and check if it's on display. If not, retry. This can be just as inefficient if we have few banners on display.
  - A middle ground approach. Currently we are loading groups of ten banners randomly and checking for any of them with display active. Then we choose randomly from them too. This is more efficient in case we have a middle number of on-display and not-on-display banners, which we think would make the most sense.

Conclusion: The last option is the most efficient one, even if it's the hardest to implement. It wasn't so difficult anyway.

<u>Requirement nº26 Operations by administrators on banners:</u>

This requirement didn't need an analysis, it was pretty straight forward.

<u>Requirement nº27 Moments, money amounts, and Booleans must be internationalised when they are shown:</u>

Decisions: For this requirement, the continuous collaboration of the team has been necessary when writing in the form correctly the internationalization of the dates, the texts, the amount type money and the Booleans, since it had to be fulfilled in this and in all subsequent group or individual deliverables.

Conclusion: In conclusion, the biggest difficulty with this requirement has been when deciding what type of attribute was being placed.

<u>Information Requirement nº28 The system must show money amounts as they are entered by the users:</u>

Decisions: For this requirement it has been necessary to use the same API as AcmeJobs for currency conversion. In addition, all members must also implement a readonly entry where the conversion is shown.

Conclusion: In conclusion, the greatest difficulty of this requirement has been to analyze how currency conversion works, but thanks to the explanations and slides of the theoretical classes we have implemented it without any major problem.

Analysis records 4

Intentionally blank.

Conclusion

This sprint has been less heavy than the previous ones, because we only had to carry out the tests, correct the errors made in the previous sprints and refactor the implemented code. For this very reason and with the slides presented in class and the laboratory classes we have not had much problem carrying out the tests that had to be implemented and to implement the improvements that were proposed to us in the laboratories.

Bibliography

- Document "On Your Derivables" of EV of the subject Design and Testing II.

- Document "So6 - On your follow-ups" of EV of the subject Design and Testing II.

- Document "o8- Annexes" of EV of the subject Design and Testing II.