

Student #2, Sprint 4: Analisis 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

- 1.-Summary	3
- 2.-Revision table	3
- 3.-Introduction	4
- 4.-Contents	5
- 4.1.-Analysis records 2	5
- 4.2.-Analysis records 3	7
- 4.3.-Analysis records 4	10
- 5.-Conclusions	11
- 6.-Bibliography	11

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	25/05/2023	Full redaction of the document

Introduction

This document lists the analysis records for each of the sprints, each of the analysis records includes the following data: a verbatim copy of the requirement to which the record refers; detailed conclusions of the analysis and decisions made to amend the requirement; a link to the validation carried out 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 nº4 Student:

Decisions: This is a simple requirement. The only real decision to be made is how to implement the list of strong features and the list of weak features, which we could implement as a list of string or a long single string.

Conclusion: Through a discussion in class with the teacher and the other groups we have determined that given the restrictions (not blank, shorter than 101 characters) the easiest method was to implement it with a single String. Also, this is how it was explained in the forum:

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=63009_1&nav=discussion_board&conf_id=303964_1&forum_id=206215_1&message_id=356955_1

Information Requirement nº5 Enrolment:

Decisions: Several doubts arose in this requirement, the first about the code attribute, since it had a bug, it has simply been corrected as discussed in the forum:

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=63009_1&nav=discussion_board&conf_id=303964_1&forum_id=206215_1&message_id=360129_1

In addition, also having a one-to-one relationship, as explained in the laboratory class, the best solution was to implement the Workbook attributes within the Enrollment class. I have also had doubts about how to implement the workTime attribute, but it was also discussed in class and has been discussed in the forum, I will implement it as a derived attribute, so it will be implemented in the service:

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=63009_1&nav=discussion_board&conf_id=303964_1&forum_id=206215_1&message_id=358091_1

Regarding the relationships, I have interpreted and corroborated in the laboratory classes that both the relationship with Student and that of Course should be Many-To-One.

Conclusion: As already explained, we have managed to solve all our problems and doubts by attending laboratory classes and the help of the subject forum.

Information Requirement nº6 Activities:

Decisions: Implementing the timePeriod 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, 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=206215_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=206215_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=206215_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. The Many-To-One relationship with Enrolment was explained in lab class so it hasn't been a problem.

Information Requirement nº7 Student Dashboard:

Decisions: For this requirement the only problem has been to decide whether to use a map to simplify the model or to have an attribute for each statistic.

Conclusion: During the laboratory classes it has been concluded that using a map is more elegant and simplifies the model.

Analysis records 3

Functional Requirement nº11 Sign up to the system and become a student:

Decisions: This has been a simple requirement, so it doesn't require any major decisions.

Conclusion: A button has been added where you can register as a student by entering the corresponding data.

Functional Requirement nº12 Students can update their profiles:

Decisions: This requirement is also very simple, it is only about updating the data of a student, checking before that only one student can update their profile.

Conclusion: There is nothing remarkable in this requirement beyond what has already been commented.

Functional Requirement nº13 Operations by any students on courses:

Decisions: For this requirement it has been necessary to list the published courses, in addition it has to be verified that the user who wants to access the list must be a student.

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=63009_1&nav=discussion_board&conf_id=303964_1&forum_id=206215_1&message_id=364648_1

In addition, the details of the course show the data of the lecturer. I have also added a button that shows the list of lectures for a specific course. Details of these lessons are also displayed.

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=63009_1&nav=discussion_board&conf_id=303964_1&forum_id=206215_1&message_id=365480_1

Conclusion: In conclusion, this has not been the most difficult requirement to implement, just a couple of shows and listings, but it is necessary for a student to have all the necessary information to be able to enroll in a course. Following the

recommendations of the laboratory class and the forum, everything has been implemented without problems

Functional Requirement nº14 Operations by students on enrollments:

Decisions: This has been the most complex requirement in my opinion, to begin with the student will be able to list their own enrollments, they will also be able to see their details and create new enrollments. You can also update and delete the enrollment if it has not been finalized.

An enrollment can be finalized when the data of a credit card is entered, that is to say holder name, the number of the credit card, the date of expiration and the CVC. Of these data, the database will only store the holder name and the last four digits of the credit card number. "lowerNibble" will be of type string because the input will allow "-" and spaces to be entered, then it will be parsed to an integer in the service.

A question that arose during the laboratory classes was how a user enrolls in a course, there were two options, or see the list of the course, select the course in which you want to enroll and from a button go to fill in the data of the enrollment with the id of the course passed by parameter, or from the list of enrollments, a form was accessed and the course was selected through a drop-down with the code and the title of the course. Finally, and on the recommendation of the laboratory professor, I have decided to implement it in both ways, because the more facilities we give to the user, the better their experience will be.

[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=364576_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=206215_1&message_id=364576_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=365854_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=206215_1&message_id=365854_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=363294_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=206215_1&message_id=363294_1)

Conclusion: In conclusion, this requirement has been the most complex but also gratifying, following the forum's recommendations and the laboratory teacher I have been able to implement everything so there have been no problems.

Functional Requirement nº15 Operations by students on workbooks:

Decisions: The only difficulty with this requirement has been to verify that you can create, modify and delete activities from a workbook as long as the associated enrollment has been completed. As already mentioned in the previous deliverable, in our database there are no workbooks as such, therefore the activities are directly related to the enrollments.

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_63009_1&nav=discussion_board&conf_id=_303964_1&forum_id=_206215_1&message_id=363300_1

Another very common question both in the forum and in the laboratory class has been when entities can be created, edited and deleted, in my case, having been advised by the laboratory class teacher and also as explained in the forum and in the requirements, the activities will begin to be created, edited or deleted once the enrollment has finished, that is to say that its draftmode is false because it would have already been charged. Therefore, you could start receiving the activities service after entering your credit card details. This is because applying the logic of the requirements, you can only start creating activities after the enrollment has finished. As the laboratory professor told us, in real life you usually pay first and then receive the service. Due to this, the list activities button will not be shown until the enrollment is finished

Another decision has been to publish or not an activity, in which I have chosen that it is not necessary since the activities can only be registered after completing the enrollment and are private to each student, therefore publishing the activities or not, It is not relevant in this context since they serve to keep a record of the work that the student is doing for the same.

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_63009_1&nav=discussion_board&conf_id=_303964_1&forum_id=_206215_1&message_id=364573_1

Conclusion: In conclusion, this requirement has not been very complex but if it has raised doubts, as always following the advice of the laboratory teacher and the forum, it has been implemented in the aforementioned way.

Functional Requirement nº16 Operations by students on student dashboards:

Decisions: This requirement will only have to calculate the required properties, for this the queries will be created in the repository, some will be stored in a map. Then these properties will be displayed on the Student Dashboard.

Conclusion: In conclusion, no important decision had to be made on this requirement.

Analysis records 4

Functional Requirement nº19 Produce a test suite for Requirements #14 and #15:

Decisions: For this deliverable it has only been necessary to do the tests themselves, so no decision has been necessary.

Conclusion: In conclusion, following the advice of the theory classes, I have done all the tests without problem.

Functional Requirement nº20 Produce assorted testing data for your test suite:

Decisions: The only decision for the CSV has been to choose the amount of data to test, in my case there are always at least 5 cases.

Conclusion: In conclusion, this requirement has been very simple since it is only about entering data that allows the tests to be executed.

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.