

# Student #1, Sprint 3: Analysis report

**Group:** C1.04.14

**Repository:** <https://github.com/marizqlav/Acme-L3-D04.git>

**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	.....	5
- 5.-Conclusions	.....	6
- 6.-Bibliography	.....	6

## **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	21/04/2023	Full redaction of the document.

## **Introduction**

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:

- Analysis report

# **Contents**

## **Analysis records**

Requirement nº11 as Anonymous principal: Sign up to the system and become a lecturer.

Decisions: A simple requirement that doesn't require any major decisions to be made.

Conclusions: A button has been added to register as lecturer by entering the corresponding data.

Requirement nº12 as Lecturer: Update their profiles.

Decisions: Again, no mayor had to be made.

Conclusions: The lecturers can now access their data and modify and update it if needed.

Requirement nº13 as Any Principal on courses:

- List the courses in the system that are published.
- Show the details of the courses that they can list (excepting their lectures).

Decisions: This is a simple listing and show of the published courses. The only real decision to be made was how to calculate the CourseType of each course, as it is a derived attribute.

Conclusions: I made an option in the main menu bar that redirects the user (any principal) to the list of courses, from which you can access the individual course data. To calculate the CourseType i have made a function in the Show Service that will get said type according to the number of each LectureType within its associated lectures. This function will be reused for other features.

#### Requirement nº14 as Lecturer on courses:

- List the courses that they have created.
- Show the details of their courses.
- Create, update, or delete their courses. Courses can be updated or deleted as long as they have not been published. For a course to be published, all of its lectures must have been published.

Decisions: This was by far the most complex requirement to implement.

The list and show were easy to implement. In the menu bar you can access the list of your courses as long as you are logged in as a lecturer. From the list you can access the courses show views.

In the list view there is a button to create a course. Both course Code and CourseType are generated in the service automatically so there is no need to add them as fields on the form.

In the show view, if the course is not published you will be able to update, delete, add lecture, remove lecture and publish. You will also always be able to see the list of lectures that are aggregated to the course.

For delete, if a course with lectures associated is deleted, the relation and course will be deleted, but not the lecture, that is to say, the CourseLectures will be deleted where it has said course.

To add lectures, I made a form which lets the lecturer select between the published lectures. This creates a new CourseLecture with the course you were in, and the selected lecture.

To remove lecture, the easiest way I could come up with to implement it is to add a button that takes you to a list view of the lectures of the course, go to the view of said lecture, and then, now that we have both the id of the course and lecture, in the show a button will allow the lecturer to remove the lecture.

Finally the publish button will only appear in the show view of the course if the course has lectures (no need to check if they are published as you can only add published lectures). Once a course is published you won't be able to do any operations on the course, with the exception of listing the lectures of said course.

Conclusions: All these decisions were checked and discussed with our teacher during practice sessions.

#### Requirement nº15 as Lecturer on Lectures:

- List the lectures in their courses.
- Show the details of their lectures.
- Create and publish a lecture.
- Update or delete a lecture as long as it is not published.

#### Decisions:

for the urls, this post was used as a guide:

[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= 363498\\_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= 363498_1)

As explained here, even though the requirement only asks for the listing of lectures on my courses, I have made both that, and the listing of all my courses. To access the listing of the lectures on courses first you have to access said course. To access all you can do it through the menu bar, both as long as you are registered as a lecturer.

The show and create are both simple enough. The create button was added to the listing of all my lecture views. You can also publish a lecture through a button in the show view, as long as it isn't already published, in other words: The lecture's draftmode has value = true.

Again the update and delete buttons are available on the show view of a lecture, as long as it's draftmode has value= true, as asked in the requirements.

Conclusions: This requirement is time consuming but not that difficult to implement. It's difficulties come with its relation with the former requirement, given the relation of aggregation that have course with lecture.

Requirement n°16 as Lecturer on lecturer Dashboards: Show their lecturer dashboards.

Decisions: As explained in the last deliverable i have decided to implement this using maps to store the data of the statistics of courses and lectures, all of which were calculated in the service.

Conclusions: No real decisions to be made, a simple feature to implement.



## **Conclusion**

This deliverable had many requirements that required analysis. Through discussions with our teacher and other classmates we have achieved a more or less satisfying solution to many of the problems encountered. However many of these solutions may not be final, as it may happen that in the next deliverable of testing makes us realize mistakes that we were not aware of. Also said requirements have a disproportionate amount of work compared to the previous deliverables.

## **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.