Sprint Retrospective Iteration #2

Issue	Task	Task Assigned to	Estimated Effort per Task	Actual Effort per task	Done	Notes
28	check each time if course is lecturer's and only then get list of candidates	Anna	1 hour	1 hour	yes	
12	for each candidate lecturer can check average rating:					
	1. add average rating to Student class	Anna	10 min	10 min	yes	
	2. find student and get the rating	Anna	20 min	40 min	yes	
10	get recommended list of students	Anna	40 min	30 min	yes	All sorting is done in course, I just need to make a request
7	get the actual number of needed TAs	Anna	10 min	20 min	yes	
13	Authentication/Authorization should work with API-gateway	Efe	6 hours	6hrs	yes	
34	API-Gateway	Efe	2 hours	8 hours	yes	learn gradle better, so you won't suffer with dependencies
4	Declare hours as a TA	Kiril	3 hours	2.5 hours	yes	To reduce network overhead, I made endpoints that allow bulk declaration and approval of hours
6	Approve hours as a Lecturer	Kiril	3 hours	2.5 hours	yes	
38	Add Strategy design pattern for TA Recommendations	Eames	4 hours	3 hours	yes	The querying network overhead needs to be reduced
37	Fix apply and accept methods for Student	Oisín	1 hour	3.5 hours	yes	Also did some refactoring in this time
30	Course hiring TA	Jeff	1 hour	1 hour	yes	only requires integration testing
	Testing all course business logic	Jeff	3 hours	2 hours	no	some last java classes requiring tests, and mocked tests

Main Problems Encountered - Ana

Problem: I was still confused about what service is responsible for some features

Description: For example: number of TAs is apparently done in course microservice, while I aslo did it in lecturer's, which is dublictaion

Reaction: In group meeting we discussed all endpoints and decided where to implement the feature

Adjustments for new sprints - Ana

For all of us the fact that it was exam week was the reason we worked less. We will complete the testing and our tasks in the next week.

I will finish and test lecturer microservice

Main Problems Encountered - Efe

Problem: I had major problems with adjusting gradle dependencies and bundling them together.

Description: spring-cloud-starter-gateway dependency was rejected for various reasons, and figuring it out without adequate gradle knowledge was very hard.

Reaction: read some of the gradle documentation and got help from stackoverflow.

Adjustments for new sprints - Efe

read gradle documentation better and understand how dependencies work.

do the same for spring dependency documentations.

Main Problems Encountered - Kiril

Problem: Parsing array of json objects to ArrayList

Description: Since I had to handle bulk approvals/declarations, I had to find a way to parse the input from my requests

Reaction: I made a research on Gson and the way to test it and implemented a Deserializer and fully tested it

Adjustments for new sprints - Kiril

I plan on finishing up Management microservice and making a design pattern for sending emails

Main Problems Encountered - Oisín

Problem: Custom query method in the repository caused issues

Description: A update method with a custom query made it so the accept and apply funcionality didn't work correctly

Reaction: Spent a long time researching different ways of updating entities in Spring, before falling back on an easier solution

Adjustments for new sprints - Oisín

Next sprint will have more time available, to implement the remaining user stories and discuss moving some functionality of others to the Student microservice

Main Problems Encountered - Eames

Problem: I had issues trying to query other microservices to retrieve specific data

Description: Since each microservice used different naming styles for their endpoints, and returned (often irrelevant data types)

Reaction: Communicated with the rest of our group so that we could agree that certain logic (like compiling a list of ID's) will be handled in their microservice, not ours (this simplified the communication between microservices on both ends.

Adjustments for new sprints - Eames

I plan on fully testing the Course microservice and writing a test suite that has good code coverage

Main Problems Encountered - Jeff

Problem: The main issue I encountered was determining the type of object that would be returned by requests to other microservices and types that should be returned to requests from other microservices

Description: Since each microservice was mostly developed in isolation in the first sprint and that mainly business logic was the focus, each service implemented logic slightly differently.
Therefore, processing and returning of request are as standardized across the microservices causing significant confusion regarding what is required by each service.

Reaction: Communication with the team to standardize request sending and request processing in terms of the data and type of data that is exchanged

Adjustments for new sprints - Jeff

The plan for the next sprint is to fully standardize the aforementioned and to fully test the communications between the Course microservice and all other relevant services