Sprint Retrospective week 4 (Iteration #1)

User story	***Task***	***Assignee***	***Estimated effort***	***Actual effort***	***Done***	***Notes***
The ID and a password shall serve as credentials for authentication.	Implement Registration	Mitchell	8 hours	12 hours	yes	Shares a lot of code with the login.
Users shall be stored in the system	Implement User Storage	Eigard	15 hours	10 hours	yes	Only very basic functionality at this point, but set up for extension
System should notify users in case they are accepted/denied for an activity	Implement notification microservice and basic functionality	Madalina	8 hours	8 hours	yes	Ready for use, might improve the format of the email message this sprint
System should notify users in case an activity they signed up for is cancelled/modified	Add corresponding methods to the director class and another API endpoint to the controller	Madalina	3 hours	3 hours	yes	Same as above; might also separate the cases (cancelled and modified) after discussion with peers.
System should notify the publisher of an activity when a user signs up for it	Add corresponding methods to the director class and another API endpoint to the controller	Madalina	3 hours	3 hours	yes	Same as above
Tests for the notification microservice	Get the test coverage over 90%	Madalina	4 hours	30 min	no	Currently only the tests for the authentication directory are in place, should be completed at the beginning of the second sprint.

The system shall match a user's availability with an activity time slot.	Implement matching microservice and basic functionality.	Diana	20h	15h	yes	We've implemented in parallel the whole skeleton of the matching microservice (issue **#39**) by following the UML and User Story we created. We started by structuring the code to match the ports and adapter low level architecture we created by doing a MatchingController and Communication Components in the application layer and implementing the core domain. *Note that for the next sprint we aim to increase the overall test coverage of these classes and carefully analyse the flow of the app whilst making sure we sanitise the inputs given by the client.* *On the way, we also managed to solve some bugs, detailed in issues **#38 and #37** allocated to this sprint.*
The user shall be able to specify positions he is capable of filling	Implement sanitization for the received position, and create API Endpoint for this functionality.	Lucian			yes	
The user shall be able to specify availability.	Implement a timeslot value object and API Endpoint for specifying availability.	Diana, Lucian			yes	
Certain competitions shall only be available for competitive users.	This blocks issue **#35**, also solved in this sprint that implies implementing the chain of responsibility design pattern for the filtering of activities, in order to do that, we created handlers for each constraint the activity shall have. This task also required to	Diana	4h	3h	yes	Decided to implement this in the first sprint even if they are should haves in order to create a good structure of the filtering functionality (main functionality of the app) in the first place. To do that we make use of a FilteringHandler interface and instantiate it in our MicroService class (core domain class of the microservice implemented)
Users competing in competitions shall be of the same gender and organization.		Diana			yes	
The boat types shall supersede each other in the following order: C4, 4+, 8+.		Diana, Lucian			yes	
The cox position shall be allowed to be filled only if the		Diana, Lucian			yes	

user has earned a certificate for the boat type.	solve ***#36***, which meant implementing functionality of certificates.					
The system shall allow users to publish trainings and competitions with positions that still need to be filled	Create an entity and a repository for saving an Activity.	Adomas	4h	4h	no	Miscommunications with teammates about the Activity class design. Also, the lack of tests. Will fix this whenever I can.
The system shall allow users to publish trainings and competitions with positions that still need to be filled	Add a controller and data models to enable communication using Rest API.	Adomas	3h	4h	no	Disagreements with teammates regarding implementation. Also, the lack of testing. Will be discussed with teammates ASAP.