# Sprint Retrospective week 5 (Iteration #2)

| User story | Task | Assignee | Estimated effort | Actual effort | Done | Notes |
|---|---|---|---|---|---|---|
| *The system shall allow users to publish trainings and competitions with positions that still need to be filled* | Finishing up the efforts from the last sprint. | Adomas | 1h | 1h | yes | |
| *The system shall allow 5 different positions to be filled: cox, coach, port side rower, starboard side rower, sculling rower* | Finishing up the efforts from the last sprint. | Adomas | 2h | 2h | yes | |
| *Publishers shall be able to cancel activities.* | Add functionality for deleting an activity. | Adomas | 8h | 10h | no | Ready for review. |
| *Publishers shall be able to edit activities.* | Add functionality for updating an activity. | Adomas | 3h | 2h | no | The code is not pushed, but it will be up for review after activity deletion is done. |
| *Users shall not be able to edit/cancel other users' activities.* | - | Adomas | - | - | no | The tasks for deletion and editing contain this requirement. |

| | | | | | | |
|---|---|---|---|---|---|---|
| *All users of the system shall be able to authenticate themselves to determine who they are and what they can do on the platform.* | Add /authenticate post mapping and handlers. | Mitchell | 10h | 14h | yes | |
| *All matches to modified activities shall be erased* | When an Activity is changed/erased all the matches to that activity should be discarded and the user shall be notified of changes | Diana | 2h | 2h | yes | Implemented the whole flow regarding activity modification/erasal by doing an api request called by the Activity microservice and then calling the Notification microservice |
| *A user shall not be matched with a training if it starts within the next 30 minutes* | Apply a time constraint filtering based on the time slot given by the client and the timeslot of the activities, whilst also taking into consideration the type of the activity | Lucian | 1h | 1h | yes | Added one more level of validation for Chain of responsibility pattern by creating a TimeConstraint handler that would take care of this issue. |
| *A user shall not be matched with a competition if it starts within a day.* | | Lucian | 1h | 1h | yes | |
| *The system shall allow publishers to accept or deny participants in training sessions* | The user requests activities based on time slot, then chooses one/more of the activities (now it's status is Pending), then the publisher pulls all the | Diana Lucian | 2h | 2h | yes | We made it to also apply for competitions. The Notification micro service is the one that emails the participants of the decision made, thus we communicate with that one. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | requests made to his activities and for each user accepts/declines the request. | | | | | |
| *Test the whole matching micro service* | Do unit and integration test | Diana Lucian | 12h | 7h | no | The current branch coverage is 68% but the controller and some of the handlers still need to be tested |
| *Test the notification microservice* | Do unit and integration testing | Madalin a | 4h | 8h | yes | Instruction coverage 95%, branch coverage 82% |