***Sprint retrospective, Iteration #3***

| User story | Task | Assignee | Estimated effort | Actual effort | Done | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| Implemented (de)serializer for Timeslot in Activity & fixes for the Activity subsystem | - | Vlad Nitu & Radu Nicolae | 1h30m | 1h30m | Yes. | Self-explanatory; See MR !28 |
| The user that created an activity can also cancel it (Matching Part) | #19 | Vlad Nitu & Radu Nicolae | 1h | 1h30m | Yes | When an user deletes an activity that he/she created, there is an API request made from Activity to Matching that informs the later subsystem to remove all the entities from its DB that have that specific activityId. See MR !28. + added `cancelActivity` API in USER subsystem in order to initiate the deletion (check commit in !33) |
| The user that created an activity can also cancel it (Actvity Part) | #19 | Laimonas Lipinskas | 2h | 3h | Yes | Struggled with testing and had to make some revisions |
| Fixed Jackson errors that occurred as the dependencies were legacy | * VV asda | Laimonas Lipinskas & Vlad Nitu & Radu Nicolae | 30m | 20h | Yes | Updated Gradle jackson dependencies, removed additional dependencies that were causing problems in Matching, Activity and User subystems |
| Implemented additional functionality & fixed Chain of responsibility in Activity | #11, #12 | Vlad Nitu & Radu Nicolae | 3h | 4h | Yes | Fixed design pattern & additional functionality provided (A user should not be matched with a training if the training starts within a half hour). See MR !32 |
| New boat types can be added | #21 | Radu Nicolae | 2h | 1h30m | Yes | See MR !32, we added two methods that allow a user to add a new boat type OR to change the current list of boat types into a new one (so maybe he wants to change the supersiding) |
| [Users should be able to take only the positions that they have the certificates for](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/20) | #20 | Vlad Nitu | 1h | 1h30m | Yes | See MR !32, check InputValidation method where we check for set containing a specific element, where the set currently consists of the positions. In the Chain of Responsability design pattern (CertificateValidator class), we check for supersiding, so that if a user has the certificate to take that specific position |
| [Adding certification to be a cox](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/13) | #13 | Razvan Loghin | 2h | 1h 30 | Yes | Part of the Chain of Resposability design pattern, … |
| [Adding certification hierarchy](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/14) | #14 | Vlad Nitu & Radu Nicolae | 2h | 3h30m | Yes | The certificates are stored in a List of String objects. We modeled it such that a certificate supersedes another certificate if it appears later in that list. As we’ve also documented our code: The latter in the list the more priority it has. The check will be made using “list.indexOf()” method. |
| A user which owns an activity can now decide to accept or decline a potential match for the activity they created | #9 | Lotte Kremer | 1h | 1h | Yes | The endpoint validates whether the owner of an activity accepts or declines (no other inputs allowed). Some extra functionality in basematching was necessary to create the matching result. |
| A user is able to get a list of activities they could take part in based from the matching subsystem by sending the user’s available timeslots and after that choosing the activity | #7#8 | Lotte Kremer | 0,5h | 0,5h | Yes | The logic is mostly in the Matching subsystem and this is just the start request for making a match. The timeslots are collected from the database and can be added through another endpoint. |
| A user can view all the activities he/she is participating in and can unenroll from an activity | #29 | Lotte Kremer | 1h | 1h | Yes | Again the logic is mostly in the Matching subsystem and the part implemented in the User subsystem just sends the request to Matching to see the activities where the user is accepted for and can then make a choice to unenroll based on the activity id |
| A owner cand edit its activity | #25 | Razvan Loghin | 2h | 2h | Yes | An owner of an activity can now edit the fields of the activity. |
| [Implement various improvement and fixes for the User microservice, including additional validation, proper error handling, associating netId with userId (connecting User with Authentication), handling API unhappy paths, refactorings and injections for future development. Ensures 97% branch coverage for User.](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/merge_requests/30) | - | Dawid Plona | 8h | 12h | Yes | Not related to any specific issue, includes general improvement and constitutes a cornerstone for further development. Took longer than expected due to loads of dependency conflicts and required refactorings. Commit list is self-explanatory |
| [Refactor all User microservice’s publisher classes to allow informing users of the potential internal errors occurring after calling another microservice. Fully document the changes and adapt the existing test suites.](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/merge_requests/33) | - | Dawid Plona | 4h | 3h | Yes | Related to multiple issues due to the interconnectedness of the publishers and all the coexisting microservices. Constitutes a basis for reliable and simple error handling while executing API calls between multiple microservices. Took slightly less than expected due to finding a neater solution with Laimonas. Commit list is self-explanatory |
| [Implement all APIs that are yet missing for the User microservice, fully document and test them. Ensure no dependency conflicts are present and resolve all potential merge conflicts.](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/merge_requests/33) | - | Dawid Plona | 8h | 10h | Yes | Related to multiple issues concerning the User microservice functionality. Exceeded the estimate time due to multiple refactorings required for wiring the APIs with other microservices’ publishers and contracts. Commit list is self-explanatory |
| [Augment and finalise the implementation of the Authentication microservice. This includes error parsing, adjusting the error messages for user-friendliness, validating netId (also in this merge request refactored to userId in the entire project) and password, migrating the database to projects, as well as adjusting tests, dependencies and documentation.](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/merge_requests/35) | - | Dawid Plona | 10h | 5h | Yes | Related to all the issues concerning user authentication, as it covers a wide variety of functionality within the Authentication microservice. Took less than expected due to help from Vlad and Radu with migrating the database and usefulness of the available IDE plugins for major code refactorings. Commit list is self-explanatory |
| Updated ReadME file to provide a demo of how to use & tested various flows of the system. | - | Vlad Nitu & Radu Nicolae | 2h | 4h | Yes | See ReadME file from GitLab |
| User can now update their timeslot | #18 | Laimonas Lipinskas | 1h | 1h | Yes |  |
| Tested the majority of the Activity Controller, the ActivitySerivce, User Publisher, TimeSlot and ActivityUtils | - | Razan Loghin | 8h | 8h | Yes | Made sure to achieve 80%+ branch coverage on activity microservice |