***Sprint retrospective, Iteration #1***

| User story | Task | Assignee | Estimated effort | Actual effort | Done | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| [Users must be able to specify availability time slots](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/7) | #30 | Vlad Nitu | 1h30m | 2h | Yes. | Implemented TimeSlot logic in order to check for overlaps and other unhappy paths. Users are asked to provide a list of timeslots when requesting through the API provided by ‘Matching’. |
| [A user must be allowed to review all the trainings and competitions that they are matched with](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/8) | #8 | Radu Nicolae & Vlad Nitu | 3h | 2h | Yes | Through ‘findAllMatchings’ API endpoint. Required multiple helper API endpoints to allow Matching microservice to communicate with other components, such as ‘getAvailableActivites’, ‘saveMatching’ and ‘chooseActivity’ |
| [The activity owner must be able to accept or decline users’ requests to join the trainings](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/9) | #9 | Radu Nicolae & Vlad Nitu | 2h30m | 4h | Yes | Through ‘chooseMatchAccept’ and ‘chooseMatchDecline’ API endpoints from MatchingController |
| [The User should be notified when accepted for an activity.](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/23) | #23 | Radu Nicolae & Vlad Nitu | 5h | 3h30m | Ongoing, done part required on Matching comp. | Through ‘notifyUser’ method that retrieves information via the API endpoint that ‘Notification’ component will provide later on in order to continue the flow of user notification. |
| [The matching should be done on a first come first serve basis](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/28) | #28 | Radu Nicolae & Vlad Nitu | 3h | 5h | Yes | Setting up the microservice and creating the first API endpoint. ‘saveMatching’ API endpoint guarantees First-come first-serve allocation |
| [Matching microservice should achieve full branch coverage](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/30) | #30 | Radu Nicolae & Vlad Nitu | 6h | 6h | Yes | Achieving (almost) 100% branch coverage, as JaCoCo report shows. |
| [A user should be able to unenroll from an activity he/she was matched to](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/29) | #29 | Radu Nicolae | 2h | 2h | Yes | Through ‘unenroll’ API endpoint provide in MatchingController, which also forwards the request to activity via ‘activiiyPublisher’ in order to re-post the position as available |
| [User must authorize in order to use the Matching microservice](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/32) | #32 | Radu Nicolae & Vlad Nitu | 2h30m | 2h | Yes | Check commit: #**e3e3249c** |
| [As the system owner, I want to make sure users' data is secure and no user can overstep their privileges.](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/1) | #1, #2,  #3,  #4,  #5,  #6 | Dawid Plona | 12h | 6h | Ongoing | Uploaded the project template containing the authentication skeleton, configured the dependencies and created the database for Authentication. Took less than expected, as the template turned out to contain a nearly-finished Authentication implementation. However, a lot of not easily quantifiable time was spent on figuring out and managing the template dependencies and its overall flow. |
| [A user can create a training activity and specify which positions still need to be filled](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/16) | #16 | Razvan Loghin | 2h 30 | 3h | Ongoing | The User is able to create a Training Activity but still cannot edit it afterwards |
| [User can create a competition activity and specify which positions still need to be filled](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/17) | #17 | Razvan Loghin | 1h 30 | 2h | Ongoing | The User is able to create a Competition Activity but still cannot edit it afterwards |
| Added the User publisher for the activity microservice | - | Razvan Loghin | 3h | 3h | Yes | Using the UserPulisher, the Activity microservice can obtain the informations about users. Essential, for the purpose of the microservice. |
| [A competition should be able to have an entry requirement of competitiveness level, gender and organization](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/27) | #27 | Razvan Loghin | 2h | 2h | Yes | A competition has the following extra attributes besides training: competitiveness level, gender and organisation. |
| [Notification microservice persists incoming notification requests and returns them for user when queried](https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM33a/-/issues/34) | #34 | Laimonas Lipinskas | 6h | 12h | Yes | The major time difference was mainly due to me having to recreate everything multiple times so that it would be in the same gradle project |
| User microservice is set-up to store all the personal information about the users of the platform and being able to send or get information to/from other subsystems |  | Lotte Kremer | 5h | 15h | Yes | Took lots of unanticipated time due to the fact that I was not yet familiar with gradle, spring etc. |