17 Create Membership for Registered User

## Description

As a user

I want to add a membership to a registered user

So that the membership will be stored in the database and is linked to that user

## API

|  |  |
| --- | --- |
| Request | POST |
| URL | http://localhost:8080/users/{userId}/addMembership |
| Path Variable | userId (int) |
| Request Body | A Membership object (JSON) – fields: startDate, endDate, type, freeLoans  All fields are required. |
| Output: happy case | A User object (JSON) – fields: id, name, age, email, password, profile with fields {id, bio, place}, list of memberships with fields {id, startDate, endDate, type, freeLoans} |
| Output: unhappy case | Response 400 BAD REQUEST  One of the following JSON error messages:   * “userId”: “User with id {id} does not exist” * “startDate”: “Start date must be equal or after than today” * “startDate”: “User has already a membership on that date” * “endDate”: “End date must be in the future” * "endDate", "End date must be 1 year after the start date." * “freeLoans”: “Invalid number of free loans for membership type.” * [Optional] “type”: “Type should be chosen from BRONZE, SILVER or GOLD” * [Optional] “startDate”: “Start date must have the format YYY-MM-DD” * [Optional] “endDate”: “End date must have the format YYY-MM-DD” |
| Acceptance Criteria | The user with given userId is updated with the membership with the values of the fields in the request body.   * We refer to story 8-10 for the validation of Membership * A User can have multiple memberships, but not more than one that running is at the same time * A membership is connected to just one user * A membership cannot be removed. * When a user is deleted, all his memberships are deleted from database.   Returns the updated user |

## Technical Requirements

* For requirements of Membership, we refer to story 8-10
* There is no API to create a membership without user
* There is no API to ask for the membership with given membershipId. A membership's information cannot be retrieved on its own.
* [Optional] Remove the field “membershipYears” in the class User.
* Add new functionalities concerning membership to the classes UserService and UserRestController.
* You do not need to validate the format of membership type and start and end dates. In other words, you only need to test cases where the membership type is BRONZE, SILVER or GOLD. The date follows the format YYYY-MM-DD.
* [Optional] Validate the user input for membership type and start and end date.