# Distributed Systems Programming

A.Y. 2022/23

# Exam Assignment for Exam Call on 01/02/2023

Deadline for submission: 30/01/2023, 12:00 CET

Extend the REST APIs of Laboratory 1, by introducing the possibility to have reviews that must be completed by more than one user in a cooperative way, according to the following specifications:

1. When a film owner assigns a review, the review may be assigned either to a single user, in which case the review process follows the original assignment specifications, or to more users, in which case the review is cooperative and the review process follows the specifications provided here. Note that the assignment of the review to one or more users occurs in a single operation when the review is created.
2. An assignee of a cooperative review can create a review draft, which is like a review data structure, including a proposed review rating, a proposed review textual description, and the id of the draft author (these fields are all mandatory in a draft).
3. When a draft is created, it is said to be open, and each one of the other co-assignees of the review can express either an agreement or a disagreement on the draft. A disagreement must include a text which expresses the reason for disagreement, while an agreement has no associated text. For simplicity, an agreement or disagreement cannot be modified. All the drafts of a review are visible to each assignee of the review, but invisible to all other users.
4. When a draft has received agreement or disagreement from all the review co-assignees (except the author, who agrees by default), the draft becomes closed, and no additional activity is possible on it.
5. Only one draft at a time can be open for each review: a new draft can be created for a review only if no other open draft currently exists for it.
6. If a draft gets closed with the agreement of all co-assignees, the corresponding review is automatically and immediately updated, with the completed flag set to true, the rating and review text taken from the draft, and the review date set to the current date. For a completed review, no new draft can be created.

In summary, a cooperative review lifecycle goes through the following states: 1. Initially, the review is created (with completed set to false). 2. A review assignee creates a new draft, which is initially open. 3. The other co-assignees can express

agreement or disagreement on the open draft (but no other draft can be created). 4. When all the other co-assignees have expressed their agreement or disagreement on the open draft, the draft becomes closed. 5. If all co-assignees have agreed on the draft that has just been closed, the review immediately becomes completed with the rating and the review text taken from the draft. Otherwise, the lifecycle goes back to step 2.

Extend the implementation of the service so that it implements the extended REST APIs. Of course, the database must also be extended accordingly. Note that it is not required to extend the REACT client. Test the extended REST APIs by using a client like Postman.

Submit the updated solution, including all the following items:

- All the json schemas
- The full Open API documentation of the REST APIs, including examples of JSON documents to be used when invoking the operations, and examples of invocations of the API operations, possibly as a Postman collection
- The REST APIs implementation code
- The database file, including some sample data
- README.md files that specify the contents of folders, the instructions on how to run the code from scratch and the main design choices made.

Important:

- Organize the project files as in the solution provided for Laboratory 1.
- The solution must work within the Labinf machines, with the software already installed in those machines.
- The solution will be tested by running it in the Labinf machines.
- The solution must be uploaded to a git repository for which you will soon get the credentials.