

# Distributed Systems Programming

A.Y. 2024/25

## Exam Assignment for Exam Call on 20/01/2025

Deadline for submission: 16/01/2025 EOD

Extend the REST APIs of Laboratory 1, by introducing the restriction for a reviewer who has already completed a review to modify it only after sending a dedicated request to the film owner. This extension should be based on the following specifications:

- After a review has been completed (i.e., the property *completed* has been set to true), if the reviewer wants to modify it, he/she must issue an explicit review modification request to ask for permission to modify the review.
- This request must contain a deadline, composed of date and time (including hour, minute and time zone) by which the answer is expected.
- The answer can be provided by the film owner, who can retrieve the review modification requests issued for the reviews of the owned films, and decide to refuse or accept each one of them.
  - If the film owner refuses a request, the property *completed* of the review remains set to true, and the reviewer cannot modify any field of the review.
  - If the film owner accepts a request, the property *completed* is set to false, and the reviewer can modify the review.
- If the film owner does not provide an answer to the request by the deadline, the request is automatically refused.
- A reviewer can retrieve the status of the requests he/she issued.
- A review modification request can be cancelled by the reviewer who issued it at any time before the film owner provides an answer for it or the deadline has passed.
- When a completed review is retrieved by a user different from the film owner and the reviewer, the possible information about the reviewer requesting a review modification, including the request deadline, must not be visible to that user.
- All other operations provided by the Film Manager service must still work after the extensions of the REST APIs are introduced to support the feature of the review modification requests.

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 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 Linux 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.