The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously. Please read all guidelines carefully before starting the review.
- Please, stay courteous, polite, respectful and constructive in all communications during this review.
- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.

Guidelines

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags in the checklist to report, for example, an “empty work” if repository does not contain the work of the student (or group) in the src folder of the develop branch, or “cheat” in case of cheating or if the student (or group) are unable to explain their work at any time during review as well as if one of the points below is not met. However, except for cheating cases, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the group.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of the check. The final decision on project evaluation remains with the reviewer.

MAIN PART

Exercise 00 - Kirov Reporting

Check that "reporting_server.py" and "reporting_client.py" scripts are present
Check that "*.proto" file(s) are present describing gRPC entities
Check that server implements a response-streaming endpoint that receives space coordinates and returns a stream of Spaceships of random length between 1 and 10
Check that the client prints all the received Spaceships as JSON strings to standard output
Check that every spaceship has fields for Alignment, Name, Class, Length, Crew size, Armed and Officers of appropriate types mentioned in the task

No Yes

Exercise 01 - Data Quality

Check that "reporting_client_v2.py" script is present
Check that Pydantic model with validators is present and implements checks according to p
rovided table
Check that non-valid Spaceship entries are not printed by client

Exercise 02 - Keeping Records

Check that "reporting_client_v3.py" script is present
Check that the database and user in PostgreSQL are created automatically or there is a des
cription in submission on how to do it manually
Check that the client successfully creates valid Spaceship entries in the database upon rece
iving
Check that the client avoids duplicates (by "Name + a set of officers" combination) while in
serting data into a database
Check that the client implements new interfaces "scan" and "list_traitors", as shown in exa
mples
Check that "traitors" are properly detected based on the entries in a database

No Yes

BONUS PART

Bonus section for EX02

Check that running "alembic upgrade head" creates the structure of tables in the database
Check that there is a separate migration for adding a "speed" field to a spaceship-storing t
able

No Yes