

TDT4242 Spring 2022: Exercise 3 – Code review, analysis, and refactoring (15 points)

Purpose of the exercise

This exercise is designed to let you practice code review, code analysis tools, and code refactoring.

Context

From exercise 1 and exercise 2, you have tested and extended the web application. In this exercise, your group will review, analysis, and refactor the web application to make the code easy to understand and easy to maintain. In this process, you may need to run/update your test scripts to test whether the code functions have been modified after refactoring.

SecFit files to review, analyze and refactor:

- frontend/www/scripts/gallery.js
- frontend/www/scripts/exercise.js
- backend/secfit/workouts/models.py
- backend/secfit/workouts/views.py
- backend/secfit/workouts/serializers.py
- backend/secfit/users/admin.py
- backend/secfit/users/views.py
- backend/secfit/users/urls.py
- backend/secfit/comments/views.py

Exercise tasks

Task 1: Define code review guideline (3 points)

Based on the lecture slides of code review and code refactoring, your group needs to define code review guidelines using a checklist with guided words to identify bad smells of code.

Task 2: Code review, code analysis, and code refactoring (10 + 2 points)

Using your defined guideline to:

- Manually reviewing the code in the listed files above to identify bad smells of code
- Run automatic code analysis tools to identify bad smells of code
- Implement code refactoring and run/update test scripts

One list of possible categories is in the slides of the code refactoring lecture.

Another list is <https://rules.sonarsource.com/python/type/Code%20Smell/RSPEC-1720>

You should identify as many as possible bad smells of code based on the above two categories.

For each bad code smell, you provide the following information in the report:

- Bad code smell category
- How the code small is identified (by manually using review guideline or by running automatic tools)

- Code snippets before and after refactoring
- Link to your test scripts preserving the correctness of the code after refactoring

Evaluation criteria

- Systematic code review guideline (3 points)
- Proper use of code review guideline and code analysis tools to identify bad smells of code and proper code refactoring (10 points)
- Proper test scripts to preserve the correctness of the code (2 points)