

Assignment: Task 1: Refactoring to Adhere to SOLID Principles

Course: Software Refactoring

Student: Olivia Lin

Score: 3.5

Account: ECE

Leonardo da Silva Sousa: You submitted the project without doing the requested refactoring (STEP 5). (Sep 15, 2024 at 3:36pm)

UPDATING FEEDBACK AND SCORE:

Because the step to "Modify the UserService Class to Depend on the Abstraction" is not completed, the violation of the Dependency Inversion Principle persists (-1.5).

As the README outlines, it's necessary to 'use dependency injection to remove the violation.' To complete this part, UserService should have an attribute pointing to DatabaseDriver (instead of PostgresDriver). Then, UserSevice should define a constructor that receives any DatabaseDriver as a parameter. In turn, the Main class should create a concrete reference to PostgresDriver and pass it as a parameter to UserService's constructor.

This last step is what is known as Dependency Injection (DI): to provide UserService with a reference to DatabaseDriver through its constructor, rather than hardcoding a specific implementation (like PostgresDriver). This refactoring offers several benefits that enhance flexibility, maintainability, testability, and adherence to design principles. For example, the DI promotes decoupling between components by ensuring that UserService depends on an abstraction (DatabaseDriver) rather than a concrete implementation (PostgresDriver), which is what we expect from a code that adheres to the dependency inversion principle.

The incomplete refactored version also violates the Single Responsibility principle (SRP). Since UserService is also responsible for creating a concrete dependency on a database, it violates the SRP because it takes on the additional responsibility of knowing how to instantiate a DatabaseDriver with PostgresDriver. With DI, UserService is only responsible for its primary function (e.g., managing user-related operations) and does not concern itself with the details of creating or managing its dependencies. (Sep 16, 2024 at 10:49am)