Assignment: Task 1: Refactoring to Adhere to SOLID Principles

Course: Software Refactoring
Student: Zizan Gong
Score: 3.5
Account: ECE

Leonardo da Silva Sousa: The refactoring for STEP 5 is incomplete; thus, 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.'

It would help if you had used 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). When UserService is responsible for creating its own dependencies, it violates the SRP because it takes on the additional responsibility of knowing how to instantiate a DatabaseDriver. 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 15, 2024 at 3:34pm)