Design Patterns Used

This document discusses the refactoring process of a Flask-based stock analysis application with two design patterns: Singleton and Strategy.

Singleton Pattern

The Singleton Pattern ensures that only one instance of a class is created, and it provides a global point of access to that instance.

Where and Why the Singleton Pattern Was Used:

A class `DatabaseConnection` was created by implementing the Singleton pattern in the method `get_connection()`, meaning only one instance of the database connection is used across the application.

Why:

In the perspective of a web application, opening and closing database connections on each request is usually inefficient. Due to the Singleton pattern, this can ensure that this instance of a database connection reuses it across requests rather than the continuous overhead of opening and closing them.

Strategy Pattern

The Strategy Pattern provides a way to define a family of algorithms and encapsulates each, ensuring they can be used interchangeably. In this example, various technical analysis strategies will be selected dynamically.

Where and Why the Strategy Pattern Was Used:

Strategy pattern was implemented on `TechnicalAnalysisStrategy` class and its subclasses (`SMAStrategy`, `EMAStrategy`, `RSIStrategy`). User will be able to select the strategy with which to perform the stock analysis.

Accordingly, the Strategy design pattern allows the application to easily extend its technical analysis strategies without including any changes to existing code. This, in turn, makes an application more flexible and easier to maintain.

Refactoring Changes

Singleton Pattern:

The `get_db_connection()` function was refactored into a `DatabaseConnection` class with a static method `get_connection()`, ensuring only one database connection is used throughout the application.

Strategy Pattern:

We introduced an abstract `TechnicalAnalysisStrategy` class with concrete strategy classes for each technical indicator. The strategy pattern allows the user to select which analysis method to apply.

In a nutshell, both the Singleton and Strategy patterns were selected for better management of database connections and extending the application for new enhancements in technical analysis strategies.