

SSAD Report - 4th Assignment

Description and Reasoning for the used Design patterns:

The provided code implements a bookstore system with features like managing books, users, subscriptions, and book updates. Several design patterns are utilized in this implementation:

Factory Method Pattern (Creational Design Pattern):

The Factory Method pattern is employed in the UserFactory class to create instances of User subclasses (StandardUser and PremiumUser) based on the provided user type. This pattern promotes loose coupling by delegating the responsibility of object creation to a separate factory class, making it easier to add new user types in the future without modifying existing client code.

Proxy Pattern (Structural Design Pattern):

The Proxy pattern is implemented through the BookProxy class. It acts as a surrogate for the Book class, providing controlled access to its methods based on the user's premium access. This pattern is chosen to add an extra layer of control over book access, allowing different behaviors for users with premium access compared to standard users.

Observer Pattern (Behavioral Design Pattern):

The Observer pattern is implemented using the SubscriptionNotifier interface and its concrete implementation, BookUpdateNotifier. This pattern facilitates the notification of subscribed users about book updates. Subscribers (users) are decoupled from the subject (book updates), allowing for efficient broadcasting of updates to multiple subscribers without them being aware of each other.

