The following Design Patterns used in this application:

Singleton (`SecurityConfig`)**:** Ensures a single instance of security configuration, promoting centralized configuration and efficient resource management.

Repository Pattern (`AppUserRepository, TransactionHistoryRepository`): Abstracts the data access logic and provides a way to manage database operations. It helps in separating the business logic from data access logic, making the codebase cleaner, more maintainable, and easier to test.

Dependency Pattern: Dependency Injection (DI), which can be implemented using various techniques such as constructor injection, setter injection, and field injection. It helps to ensure Loose Coupling, Testability, Maintainability, and Scalability.

Builder Pattern: For Readability, Immutability, Flexibility, and Validation.

Decorator Pattern (`JwtUsernameAndPasswordAuthenticationFilter`): Adds additional responsibilities to HTTP request handling, enabling flexible and modular enhancement of functionalities.