

LoginFrame

1. Command Pattern

- **Dove si applica:** gestione delle azioni dei bottoni
- **Motivazione:** Il Command Pattern si utilizza per separare le azioni dei bottoni dalla logica di esecuzione. In LoginFrame, le azioni associate ai bottoni "Login" e "Indietro" possono essere incapsulate in oggetti di comando separati. Questo approccio rende il codice più modulare e facilita l'estensione delle azioni future senza modificare il codice esistente

2. Strategy Pattern

- **Dove si applica:** autenticazione dell'utente
- **Motivazione:** se si prevede di avere diversi algoritmi di autenticazione (ad esempio, autenticazione tramite password, autenticazione a due fattori), il Strategy Pattern può essere utilizzato per definire una famiglia di algoritmi di autenticazione e selezionare quello appropriato. In LoginFrame, la logica di autenticazione tramite DatabaseManager potrebbe essere estesa per supportare ulteriori metodi di autenticazione senza cambiare la logica del frame

3. Singleton Pattern

- **Dove si applica:** gestione della connessione al database
- **Motivazione:** per garantire che ci sia solo una singola istanza di DatabaseManager utilizzata in tutta l'applicazione e per gestire in modo centralizzato la connessione al database, il Singleton Pattern può essere applicato a DatabaseManager. Questo approccio evita conflitti e problemi di accesso concorrente e semplifica la gestione della connessione al database

4. Factory Method Pattern

- **Dove si applica:** creazione di componenti dell'interfaccia utente

- **Motivazione:** se si prevede di avere diverse configurazioni o stili per i componenti dell'interfaccia utente, il Factory Method Pattern può essere utilizzato per centralizzare la creazione di questi componenti. Qui si usa per creare istanze di JButton, JTextField, e JPasswordField con configurazioni personalizzate, migliorando la modularità e la coerenza del design dell'interfaccia