Progetto Gestore di libreria personale

# Descrizione del progetto

Il seguente progetto fornisce agli utenti un software per memorizzare e aggiornare la propria libreria virtuale, tramite un database che terrà traccia dei propri libri, con tanto di caratteristiche specifiche e informazioni inerenti allo stato di lettura e alla valutazione inserita dall’utente.

I dati saranno salvati in file JSON o CSV sul dispositivo dell’utente, e potranno coesistere più file di librerie, aperti dall’utente tramite path di salvataggio.

Il progetto comprenderà una suite di test adibita per il controllo delle varie sezioni programmate, come il dialogo con il database, le interazioni con l’utente e le modifiche della libreria.

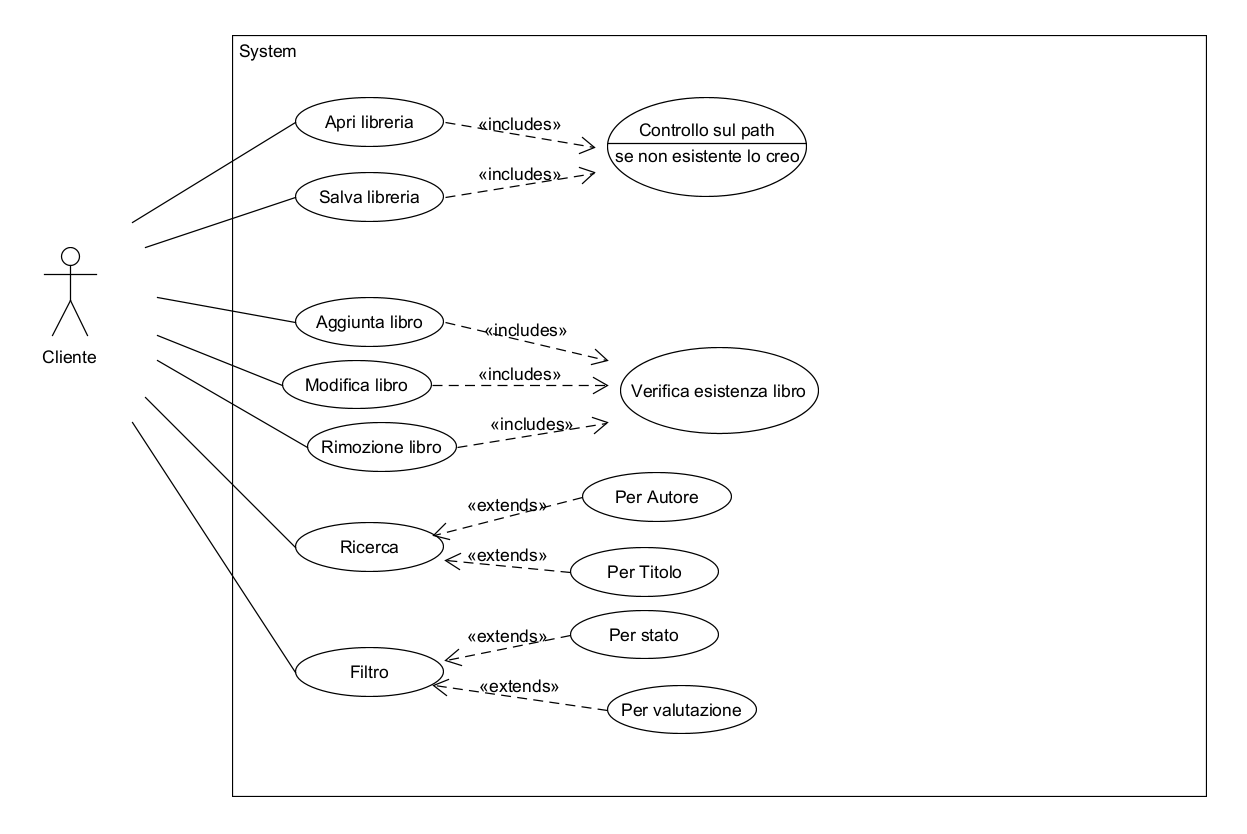
L’utente potrà accedere alla sua libreria tramite una semplice interfaccia, dove avrà a disposizione tutti i comandi per modificarla e una struttura per esplorarla e vedere i libri salvati all’interno.

# Struttura del progetto

I vari elementi fondamentali, le classi per implementare funzioni come la modifica dei libri e la ricerca degli stessi, sono stati fedelmente programmati basandosi sulla struttura standard dei vari design pattern, evidenziando i collegamenti tra le classi unite in package specifici. In aggiunta un package rappresenta tutta la sezione che dialoga con l’utente, la GUI, ed infine si ha un package separato per definire i tipi fondamentali del progetto, usati da tutti gli altri elementi, le classi “Libro” e “Libreria”.

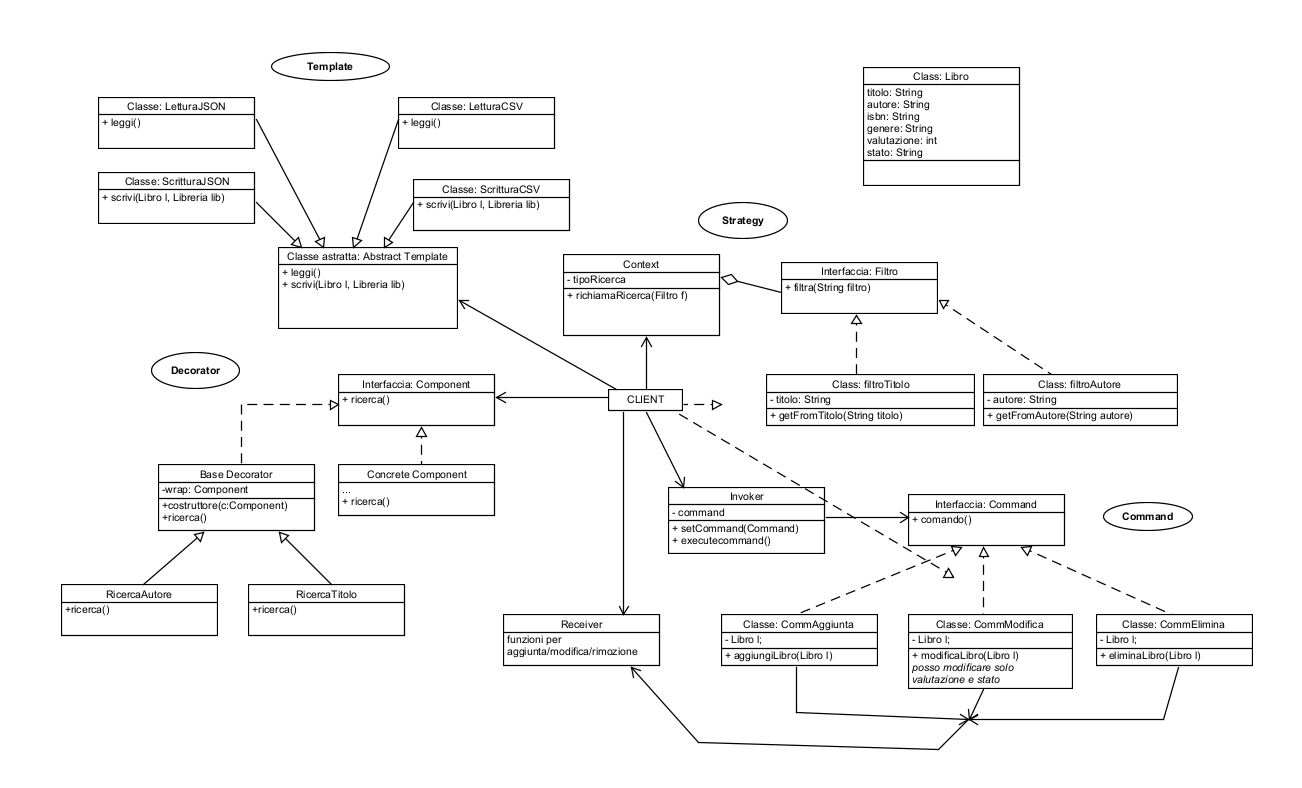
# Use case

Col seguente schema si definiscono le interazioni che l’utente potrà avere col software, gli elementi modificabili e le funzioni disponibili per tenere traccia della propria libreria.



# Implementazioni e design patterns

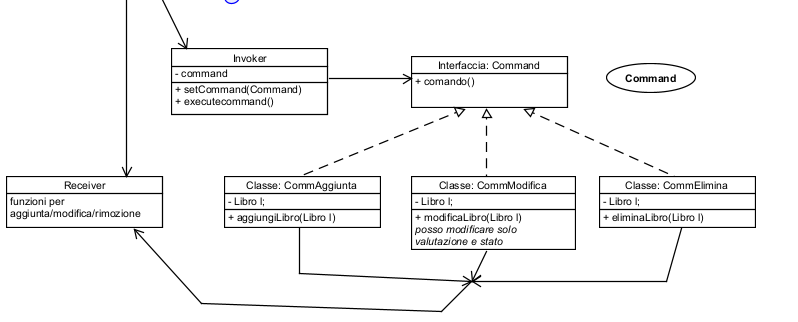
Di seguito è fornita la struttura generale del progetto tramite il suo class diagram, successivamente si andranno ad esporre nello specifico le varie classi e i pattern utilizzati.



Il software è suddiviso principalmente in 4 sezioni: modifica libreria, filtri, ricerche e interazione col database.

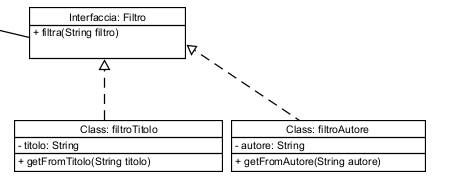
Modifica libreria: Command

La possibilità di aggiungere, modificare e rimuovere libri della libreria è stata inserita tramite il design pattern “command”; data la presenza di diverse istruzioni possibili, che potevano essere gestite in maniera simile, si è scelto di usare questi tipo di pattern, particolarmente utile in casi del genere.



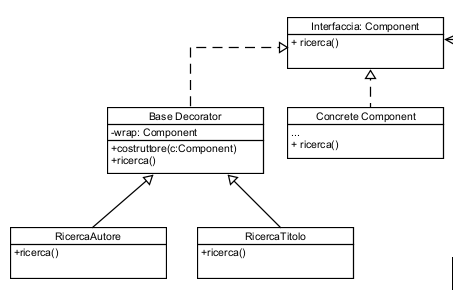
Ricerche: Strategy

Il pattern Strategy permette al client di conoscere un'unica interfaccia dove usare comandi diversi, basati su un concetto comune; in questo caso i vari comandi rappresentano i metodi per ricercare i libri, che poi saranno mostrati in una tabella contenente solo quelli che possiedono qualità specifiche nei campi “Titolo” e “Autore”



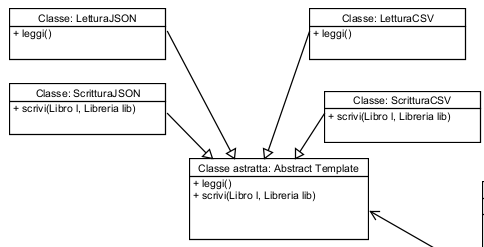
Filtri: Decorator

Dato che più filtri possono coesistere in una sola interfaccia, l’idea di base dietro il pattern decorator è interessante da usare come implementazione; in questo caso si può applicare sulla libreria un oggetto Filtro, che può essere formato a sua volta da più filtri, in base a quanto specifica sarà la nostra ricerca sulla libreria.



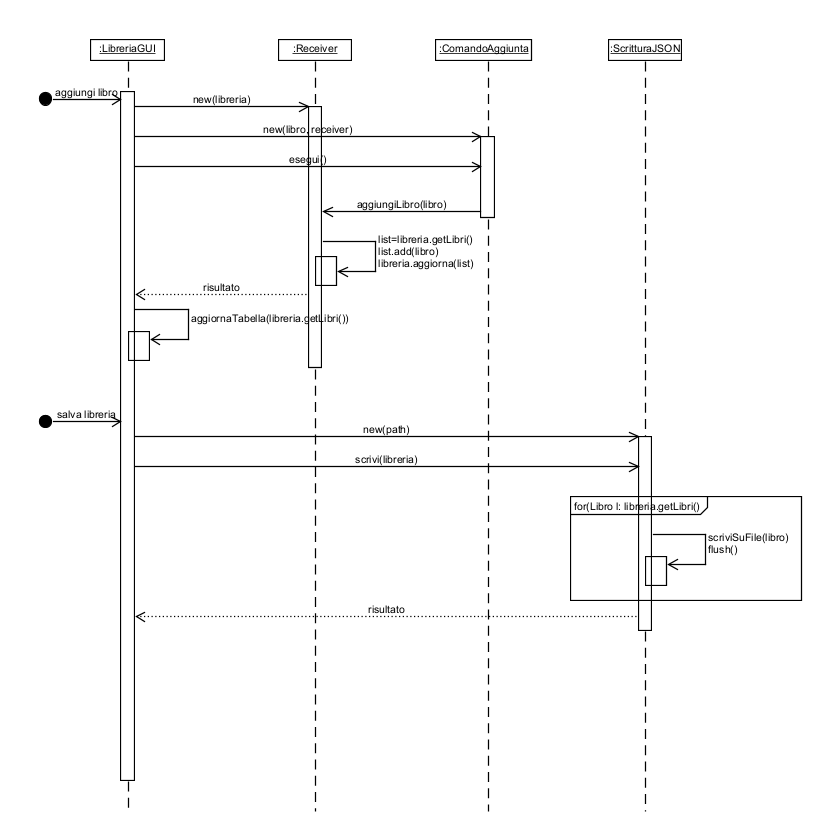
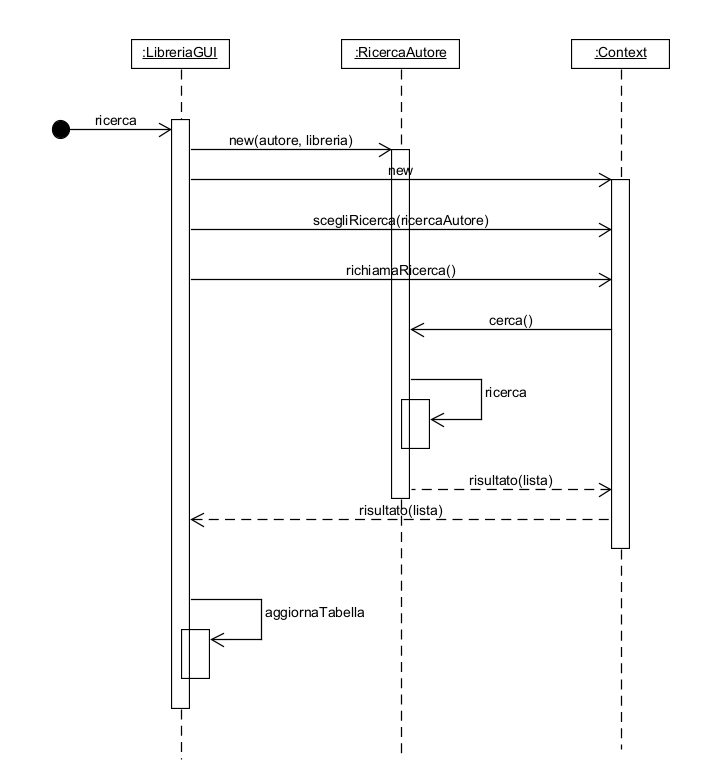
Scrittura su file: Template

L’idea iniziale era di implementare una interfaccia originale con le funzioni di lettura e scrittura generiche, per poi fare in modo che le classi concrete potessero rendere più specifiche quelle operazioni implementando le capacità per interagire con file .json e .csv:



## Sequence diagram e funzionamento del software

Di seguito si trovano i diagrammi di sequenza per mostrare le interazioni che le varie classi hanno in risposta ai comandi dell’utente:

* Modifica libreria e salvataggio in memoria(mostra funzionamento dei pattern command e template)  
  
* Ricerca (strategy, e sezione di GUI):  
  

## Component diagram

Qui si mostra la struttura dei vari elementi e come questi interagiscano tra loro.

