

SPECIFICA

Il progetto si propone di realizzare una *biblioteca digitale* di testi e studi che contribuiscono alla formazione della cultura all'interno dell'Università degli Studi dell'Aquila.

Una biblioteca digitale è uno spazio in cui mettere insieme collezioni, servizi e persone a supporto dell'intero ciclo di vita di creazione, uso, preservazione di dati, informazione e conoscenza. Lo scopo di questo progetto è la digitalizzazione di manoscritti, che costituiscono un patrimonio bibliografico antico per un totale di 60.000 carte (ms. sec. XV-XIX) contenenti memorie storiche della città dell'Aquila. Il processo di digitalizzazione dei manoscritti si suddivide in diverse fasi.

Digitalizzazione:

Il manoscritto è acquisito dal sistema sotto forma di immagini digitali ad alta risoluzione attraverso scanner planetari. Ogni manoscritto è formato da più acquisizioni (ogni immagine rappresenta una singola pagina). La digitalizzazione viene controllata da supervisori all'acquisizione per assicurarne la correttezza (ad esempio, in accordo con standard richiesti) e la qualità. L'immagine acquisita viene memorizzata all'interno del sistema ed assegnata all'opera di riferimento, corredandola di opportuni metadati.

Trascrizione:

Il manoscritto così acquisito deve essere trasformato in un testo digitale; ciò avviene attraverso operazioni di trascrizioni in formato TEI (Text Encoding Initiative).

Le trascrizioni sono digitate manualmente (la natura del testo rende inutilizzabili strumenti di acquisizione automatica) attraverso un text editor TEI integrato.

Le trascrizioni sono oggetto di revisione da parte di revisori alle trascrizioni.

Pubblicazione:

I manoscritti, una volta digitalizzati e superata la fase di revisione delle immagini, vengono pubblicati sul sistema e resi accessibili agli utenti del sistema. Le corrispondenti trascrizioni sono pubblicate successivamente, dopo la validazione della stessa.

Attori del sistema:

- Amministratore: gestione generale del sistema
- Acquisitore: acquisizione/digitalizzazione immagine
- Revisore acquisizioni: revisione e verifica correttezza dell'acquisizione
- Trascrittore: trascrizione del testo TEI

- Revisore trascrizioni: revisione e validazione della trascrizione
- Utente base: visualizzazione elenco titoli opere
- Utente avanzato: visualizzazione completa delle opere

REQUIREMENTS

1.1) Documento dei Requisiti

Scenari

In seguito alla lettura della specifica sono stati individuati i seguenti scenari:

1) Un utente base quindi non registrato vuole utilizzare il sistema

L'utente può utilizzare il sistema senza registrarsi usufruendo della sola funzionalità di visualizzare i titoli delle opere presenti nel sistema. (Utente Base)

L'utente può invece registrarsi, fornendo i propri dati personali, per avere accesso alla visualizzazione completa delle opere presenti nel sistema. (Utente Avanzato)

2) Un utente avanzato quindi registrato vuole utilizzare il sistema

L'utente inserisce le proprie credenziali e accede al sistema se queste sono corrette.

3) Trascrittore, Revisore, Acquisitore devono utilizzare il sistema per svolgere i propri compiti

Essi accedono al sistema fornendo le credenziali fornitegli dall'amministratore.

4) Acquisitore vuole caricare un'immagine da revisionare

Ottenuta l'immagine dallo scanner planetario, l'Acquisitore inserisce l'immagine nel sistema.

5) Revisore delle acquisizioni deve revisionare le immagini caricate dall'Acquisitore

Il Revisore valuta l'immagine e determina se rispetta gli standard previsti. Nel caso in cui essa rispetti gli standard questi la pubblica, altrimenti comunica all'acquisitore di rieseguire la procedura.

6) Il Trascrittore deve trascrivere quanto è riportato nell'immagine pubblicata

Il Trascrittore trascrive il contenuto dell'immagine tramite un editor TEI interno al sistema e carica nel sistema il risultato del suo lavoro.

7) Il Revisore delle trascrizioni deve revisionare le trascrizioni caricate dal trascrittore

Il Revisore valuta la trascrizione. Nel caso in cui essa sia corretta questi la pubblica, altrimenti comunica al trascrittore di effettuare di nuovo la trascrizione.

8) Un utente base o avanzato vuole visualizzare le opere consultabili nel sistema

L'utente effettua una ricerca sui titoli disponibili nel sistema.

9) Un utente avanzato vuole visualizzare un'opera

L'utente effettuata la ricerca, seleziona l'opera dalla lista dei titoli dati come risultato. Il sistema stampa a video le immagini e le trascrizioni (se presenti).

10) L'Amministratore del sistema deve gestire il sistema

L'Amministratore deve fornire le credenziali di accesso ai vari collaboratori, deve poter svolgere tutte le attività, deve poter gestire le opere e deve poter garantire il corretto funzionamento del sistema.

Assunzioni

- 1) L'amministratore comunica attraverso canali esterni (mail) le credenziali di accesso al personale del sistema.
- 2) I Revisori comunicano a Trascrittori e Acquisitori gli esiti delle revisioni attraverso canali esterni (mail).
- 3) La digitalizzazione delle immagini viene svolta esternamente al sistema.
- 4) Si assume che tutti i dati che il sistema deve gestire ed immagazzinare siano memorizzati all'interno di un database.

Requisiti funzionali

A seguito degli scenari esposti sono state evidenziate le seguenti funzionalità

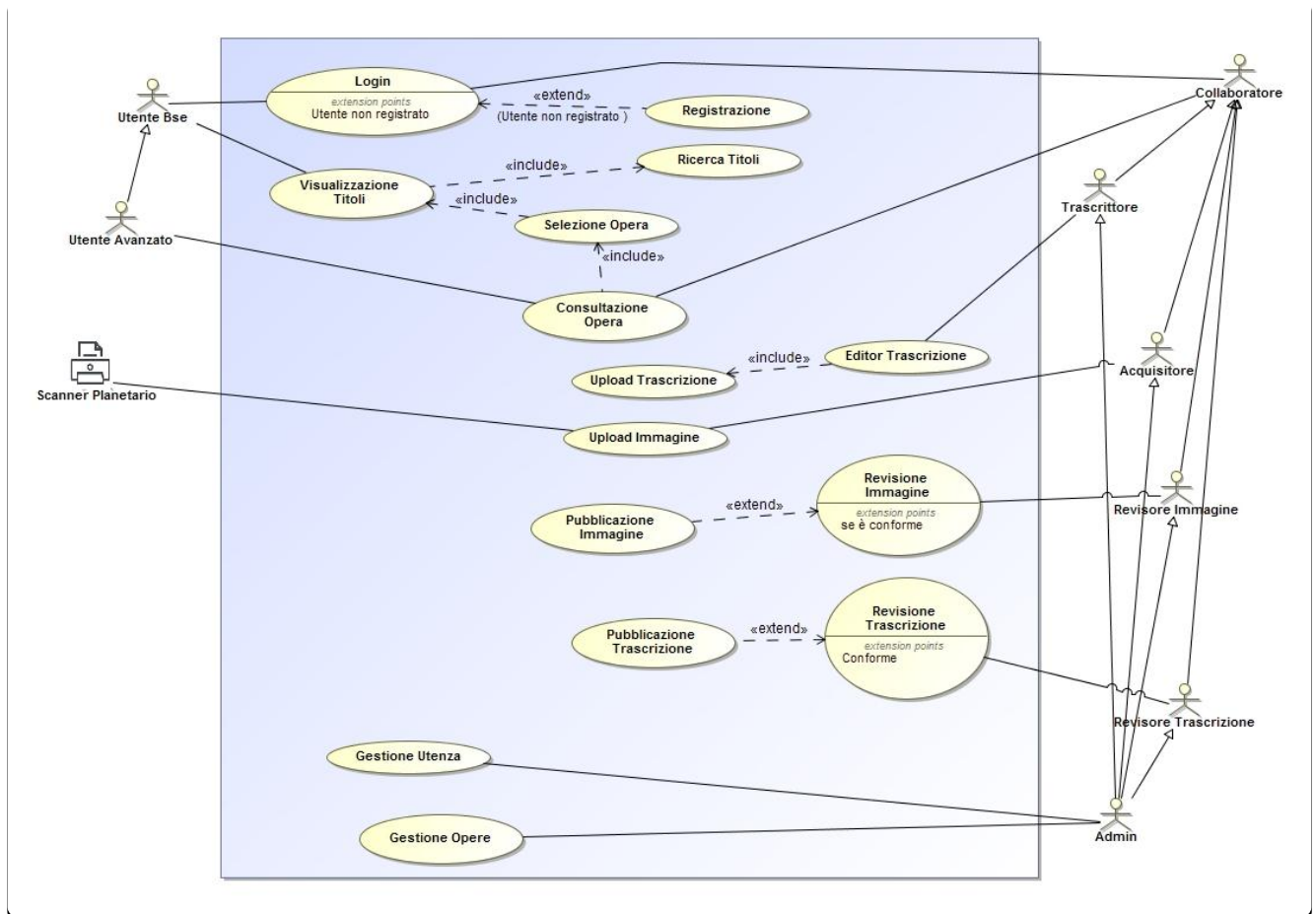
- **Registrazione** Utente (scenario 1)
- **Login** (scenari 2 e 3)
- **Upload immagine** non revisionata (scenario 4)
- **Revisione immagine** (scenario 5)
- **Pubblicazione immagine** revisionata e ritenuta conforme agli standard richiesti (scenario 5)
- **Editor trascrizione** (scenario 6)
- **Upload trascrizione** non revisionata (scenario 6)

- **Revisione trascrizione** (scenario 7)
- **Pubblicazione trascrizione** in seguito a revisione positiva (scenario 7)
- **Ricerca titoli** (scenario 8)
- **Visualizzazione titoli** opere (scenario 8)
- **Selezione opera** (scenario 9)
- **Consultazione opera** (scenario 9)
- **Gestione Utenza** (scenario 10)
- **Gestione Opere** (scenario 10)

Requisiti non funzionali

- **Performance:** il sistema deve permettere all'utente di svolgere le attività in modo veloce ed efficiente. (pochi secondi di attesa)
- **Usability:** il sistema, in particolare l'interfaccia utente, deve essere facile da utilizzare in quanto il target a cui è diretto tale sistema può non avere le competenze tecnologiche necessarie.
- **Reliability:** il sistema deve svolgere le attività preposte senza altri fini.
- **Availability:** dal momento che i dati saranno memorizzati all'interno di un database, questi deve essere sempre accessibile.
- **Security:** il sistema deve proteggere i dati dell'utente.
- **Maintainability:** il sistema deve essere facile da mantenere, aperto a migliorie e nuove funzionalità.

Use Case



Descrizione Use Case

Use Case	Registrazione
Attori partecipanti	Utente base
Descrizione	L'utente base vuole avere pieno accesso alle funzionalità del sistema
Estende	Login
Include / incluso in	-
Condizione in entrata	L'utente avvia la procedura di registrazione fornendo le proprie credenziali
Condizione all'uscita	L'utente è registrato nel sistema

Use Case	Login
Attori partecipanti	Utente base, Utente avanzato, Collaboratori

Descrizione	Gli utenti già registrati inseriscono le proprie credenziali per accedere al sistema
Esteso da	Registrazione
Include /incluso in	-
Condizione in entrata	L'utente avvia la procedura di login fornendo le proprie credenziali
Condizione all'uscita	L'utente è entrato nel sistema

Use Case	Upload immagine
Attori partecipanti	Acquisitore, Scanner Planetario
Descrizione	Acquisitore carica le immagini provenienti dallo scanner planetario
Estende / esteso da	-
Include/ incluso in	-
Condizione in entrata	L'utente avvia la procedura di caricamento dell'immagine
Condizione all'uscita	L'immagine è caricata nel database del sistema

Use Case	Revisione Immagine
Attori partecipanti	Revisore immagini
Descrizione	Il revisore deve visualizzare l'immagine per valutarla.
Esteso da	Pubblicazione immagine
Incluso in/ include	-
Condizione in entrata	Il revisore revisiona l'immagine.
Condizione all'uscita	Il revisore decide se è conforme o meno agli standard

Use Case	Pubblicazione immagine
Attori partecipanti	Revisore immagini
Descrizione	Il revisore pubblica l'immagine in quanto la ritiene conforme agli standard
Estende	Revisione immagine
Include/ incluso da	-
Condizione in entrata	Il revisore ha revisionato l'immagine
Condizione all'uscita	Immagine resa pubblica

Use Case	Revisione trascrizione
Attori partecipanti	Revisore trascrizioni
Descrizione	Il Revisore deve visualizzare la trascrizione per valutarla
Esteso da	Pubblicazione trascrizione
Incluso in / include	-
Condizione in entrata	Il revisore revisiona la trascrizione
Condizione all'uscita	Il revisore decide se è corretta o meno

Use Case	Pubblicazione trascrizione
Attori partecipanti	Revisore trascrizioni
Descrizione	Il revisore pubblica la trascrizione in quanto è esatta

Estende	Revisione trascrizione
Include/ incluso in	-
Condizione in entrata	Il revisore reputa corretta la trascrizione
Condizione all'uscita	Il revisore pubblica la trascrizione

Use Case	Editor trascrizione
Attori partecipanti	Trascrittore
Descrizione	Il trascrittore utilizza un editor interno al sistema per trascrivere
Estende/ esteso da	-
Include	Upload trascrizione
Condizione in entrata	Il trascrittore deve trascrivere e caricare quanto riportato sopra un immagine
Condizione all'uscita	Il trascrittore ha trascritto quanto riportato sopra l'immagine

Use Case	Upload trascrizione
Attori partecipanti	Trascrittore
Descrizione	Il trascrittore carica la trascrizione da valutare nel sistema
Estende/ esteso da	-
Incluso in	Editor trascrizione
Condizione in entrata	Il trascrittore ha trascritto quanto riportato sopra un immagine
Condizione all'uscita	Il trascrittore ha caricato la trascrizione

Use Case	Ricerca titoli
Attori partecipanti	Utente base, Utente avanzato, Collaboratori
Descrizione	Gli utenti e i vari collaboratori utilizzano la funzione di ricerca per trovare i titoli
Estende/ esteso da	-
Incluso in	Visualizzazione titoli
Condizione in entrata	Gli utenti e i vari collaboratori devono ricercare uno o più titoli
Condizione all'uscita	Il sistema ricerca i titoli conformi alla ricerca

Use Case	Selezione opera
Attori partecipanti	Utente avanzato, Collaboratori
Descrizione	Gli utenti e i vari collaboratori selezionano l'opera che gli interessa
Estende/ esteso da	-
Include/ incluso da	Include: Visualizzazione titoli / Incluso da: Consultazione opera
Condizione in entrata	Gli utenti citati visualizzano la lista dei titoli
Condizione all'uscita	Gli utenti citati selezionano il titolo di proprio interesse

Use Case	Visualizza titoli
Attori partecipanti	Utente base, Utente avanzato, Collaboratori
Descrizione	Dopo aver fatto la ricerca, gli utenti visualizzano i titoli disponibili
Estende/ esteso da	-
Include/ incluso da	Incluso da: Selezione opera / Include: Ricerca titoli

Condizione in entrata	Gli utenti hanno effettuato la ricerca
Condizione all'uscita	Il sistema elenca i titoli conformi alla ricerca

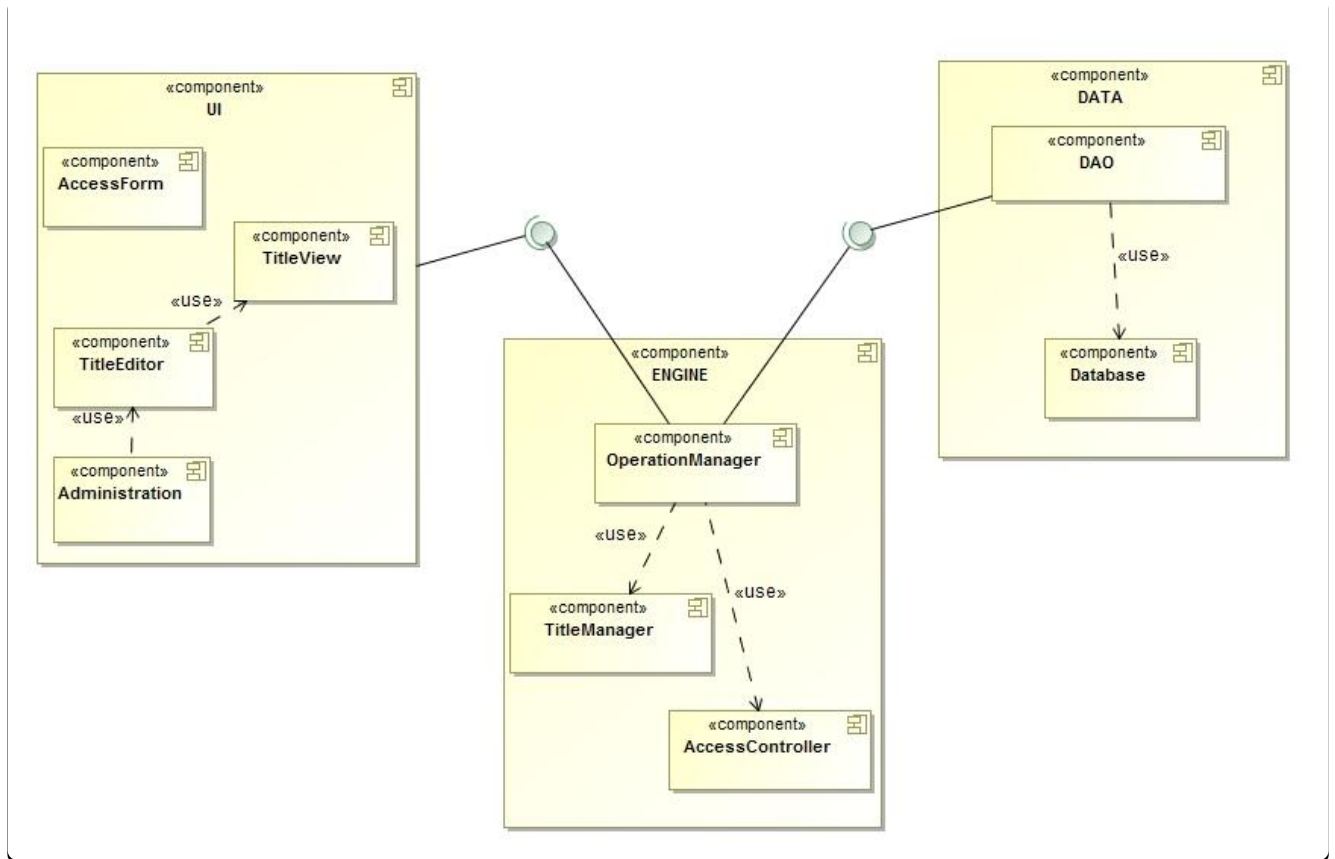
Use Case	Consultazione opera
Attori partecipanti	Utente avanzato, Collaboratori
Descrizione	Gli attori citati vogliono consultare l'opera
Estende/ esteso da	-
Include/ I	Include: Selezione opera / Incluso da: Visualizzazione Immagini, Visualizzazione Trascrizione
Condizione in entrata	L'opera è stata selezionata
Condizione all'uscita	Il sistema permette di consultare l'opera

Use Case	Gestione utenza
Attori partecipanti	Amministratore
Descrizione	L'amministratore può gestire i permessi dell'utenza
Estende/ esteso da	-
Include/ incluso in	-
Condizione in entrata	L'amministratore vuole gestire i permessi
Condizione all'uscita	L'amministratore ha svolto le sue operazioni

Use Case	Gestione opere
Attori partecipanti	Amministratore
Descrizione	L'amministratore può gestire le opere presenti nel sistema
Estende/ esteso da	-
Include/ incluso in	-
Condizione in entrata	L'amministratore vuole gestire le opere
Condizione all'uscita	L'amministratore ha svolto le sue operazioni

SYSTEM DESIGN

2.1) Modello dell'architettura software



2.2) Descrizione dell'architettura

Dopo aver letto la specifica del progetto si è subito cercato di individuare quale fosse l'ambiente di utilizzo del sistema. Dopo varie discussioni si è deciso di optare per la costruzione di un sistema locale.

Al fine di soddisfare i principi di Low Coupling e High Cohesion abbiamo suddiviso il sistema in tre componenti principali: Ui, Engine e Data. I sistemi contengono al loro volta altre sotto componenti tutte strettamente connesse tra di loro.

UI → E' la componente che ha il ruolo di far interfacciare l'utente con il sistema.

- La sottocomponente **AccessForm** permette all'utente di poter inserire, attraverso una form, le proprie credenziali di accesso o di registrazione.
- La sottocomponente **TitleView** permette all'utente di poter ricercare opere, visualizzare titoli e consultare le opere.

- La sottocomponente **TitleEditor** permette ai collaboratori di fare l'upload delle immagini attraverso una form, uploadare ed editare le trascrizioni attraverso l'editor e di revisionare le trascrizioni e le immagini.
- La sottocomponente **Administration** permette all'amministratore di gestire l'utenza e le opere attraverso una sorta di backoffice.

ENGINE→E' il motore del sistema, l'addetto ad elaborare dati, richieste ed accessi.

- La sottocomponente **Operation Manager** ha il ruolo di gestire i dati in arrivo dalla **UI** delegando il lavoro alle altre sottocomponenti. Offre servizi alla componente **UI**.
- La sottocomponente **Title Manager** ha il compito di gestire le richieste riguardanti le opere, le trascrizioni, le immagini e i titoli.
- La sottocomponente **AccessManager** ha la mansione di gestire i dati in arrivo dalle form di accesso e verificarne la correttezza.

DATA→E' la "mente" del sistema, è dove i dati vengono incapsulati e memorizzati.

- La sottocomponente **DAO** ha il ruolo eseguire le queries al database (a seconda di quanto serve alle componenti richiamate dall'**OperationManager**) e di incapsulare i dati in oggetti. Offre servizi all'**OperationManager**.
- La sottocomponente **Database** è addetta al mantenimento di ogni tipologia di dato.

2.3) Descrizione scelte e delle strategie adottate

Come detto nella sezione precedente, si è optato per la creazione di un applicativo desktop che si connetta ad un database locale. Abbiamo quindi che la nostra applicazione avrà una interfaccia grafica attraverso cui l'utente può visualizzare, modificare ed interagire con i dati presenti nel database. Tutto questo grazie ad una componente di controllo (**Engine**) che avrà il ruolo di effettuare le operazioni.

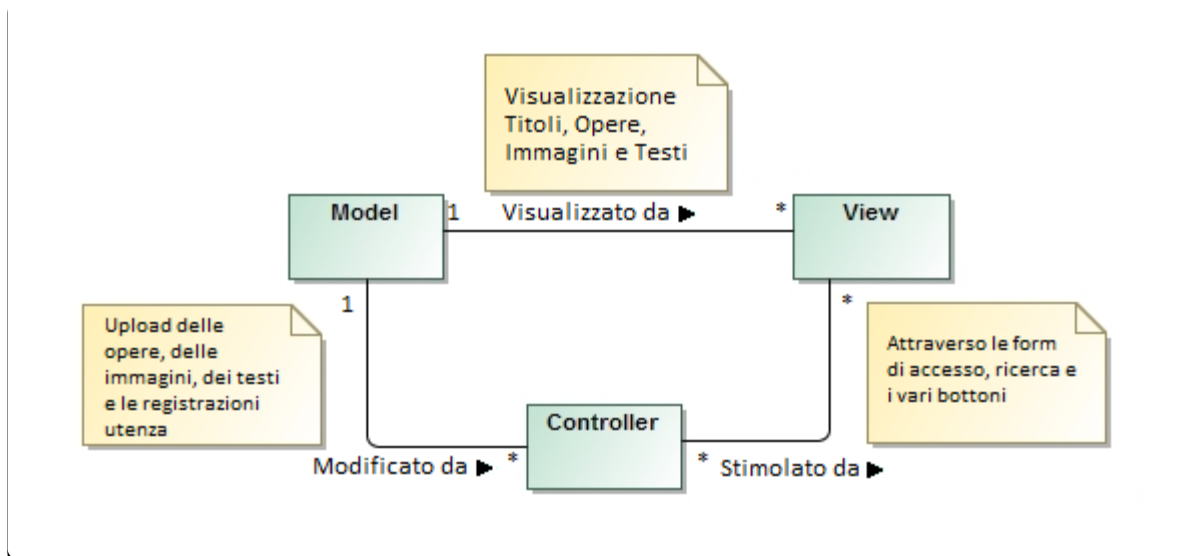
Abbiamo deciso di utilizzare a questo proposito due design pattern:

- MVC design pattern

Perché?

Si è scelto il design pattern MVC per la suddivisione delle logiche che adotta. Le sue componenti rispecchiano a pieno le dinamiche del nostro sistema. L'utente che riempie i campi delle form non fa altro che stimolare attraverso la View quello che è il controller che a sua volta interagisce con il Model. Il visualizzare un'opera non è altro che visualizzare quanto presente nel Model all'interno della View. Il

collaboratore che fa l'upload delle immagini, delle trascrizioni o fa le modifiche non fa altro che apportare cambiamenti al Model e a quanto verrà visualizzato successivamente.

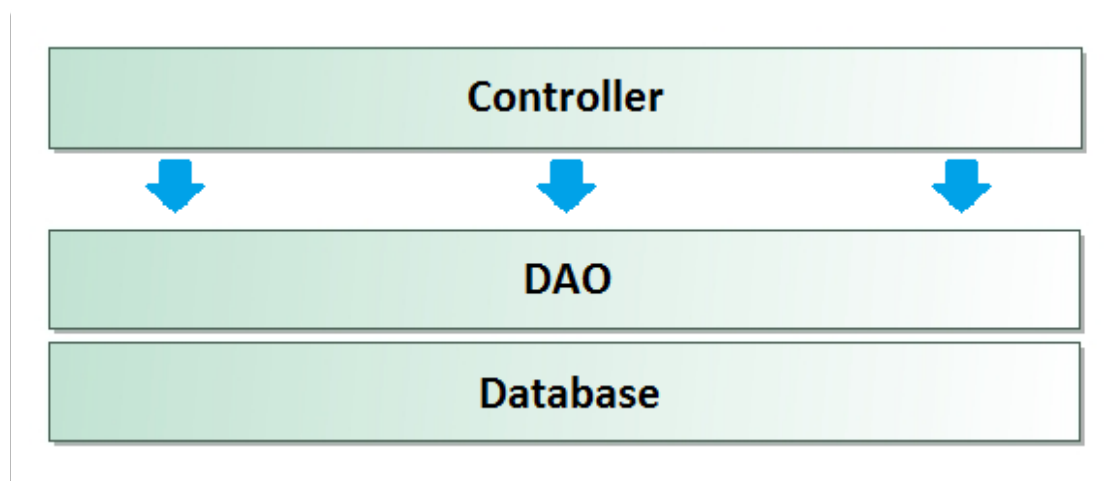


- DAO design pattern

Perché?

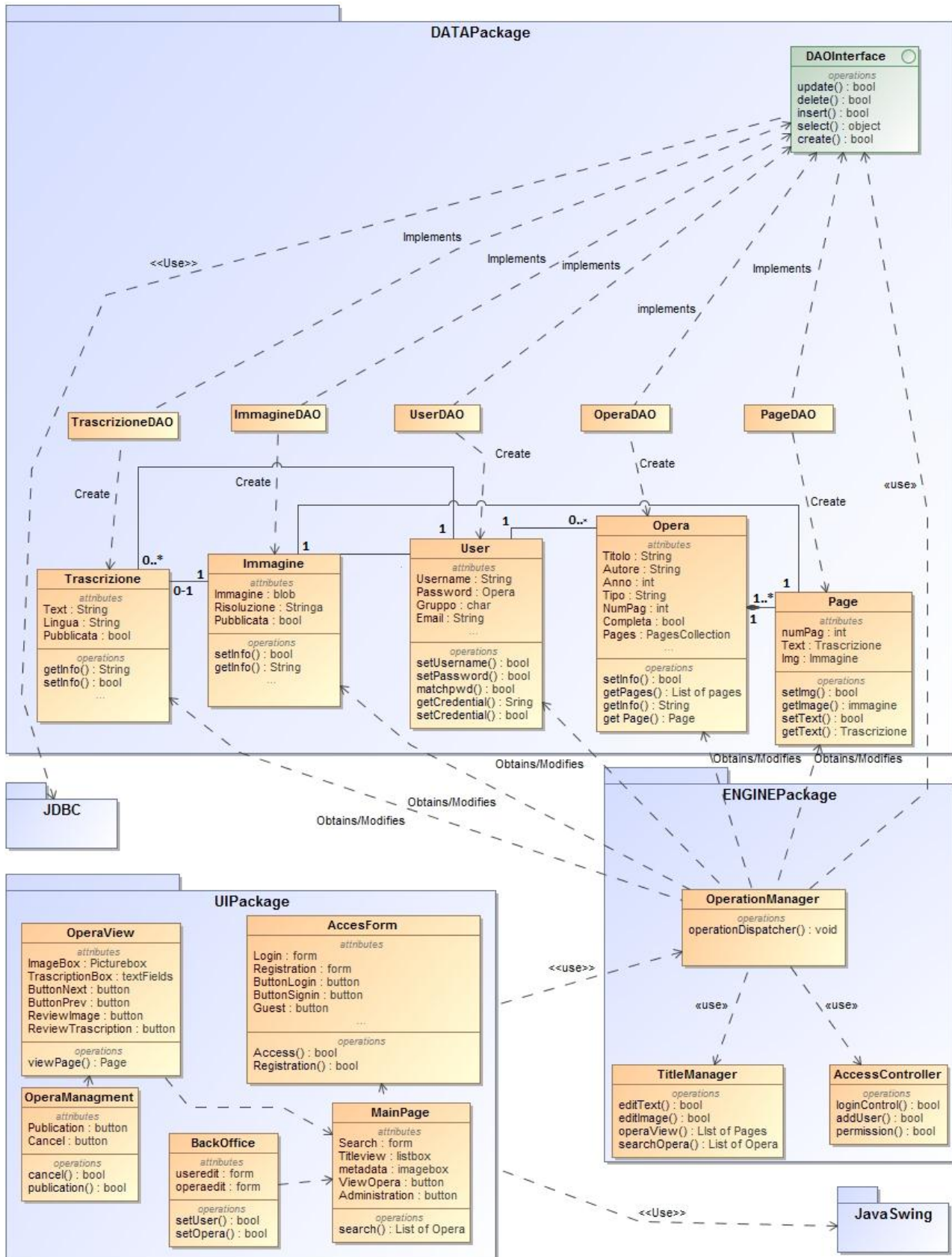
Abbiamo utilizzato il DAO per due motivi:

1. **Separazione delle logiche.** Per isolare l'accesso al database facendo in modo che la nostra logica di controllo non comunichi direttamente con il database, ma con i metodi delle classi DAO addetti ad incapsulare, attraverso le queries, i dati presenti in esso.
2. **Manutenibilità.** In quanto le classi DAO saranno in numero inferiori rispetto alle classi, che costituiscono la logica, che le utilizzano. Quindi in caso di modifiche noi andremo a modificare solo il lato DAO e non tutte le classi che utilizzano i vari dati facilitando il tutto.



SOFTWARE/OBJECT DESIGN

3.1) Modelli rappresentanti l'object design con classi, interfacce e membri



3.2) Descrizione dei dettagli di design scelti

Abbiamo diviso il nostro sistema in 5 package:

1)DataPackage: al fine di seguire il DAO Design Pattern si è costruita una DAOInterface che è un'interfaccia che definisce solo le firme dei metodi che eseguono le operazioni tipiche che si possono effettuare in un database. L'implementazione di tali metodi è rimandata alle ClassiDAO (OperaDAO, PageDAO...) che sono le classi che realmente comunicano con il database ed eseguono le operazioni di selezione, modifica, inserimento e cancellazione. Ognuna di queste operazioni comporta la creazione di oggetti (Opera, Page...) che saranno poi quegli oggetti con cui il resto del sistema interagisce.

2)EnginePackage: questo package contiene al suo interno la classe OperationManager, che interagisce con gli oggetti creati dalle ClassiDAO richiamando altre due classi di questo package, che sono TitleManager (per le operazioni su opere, pagine, immagini, trascrizioni...) e AccessController (per il controllo degli accessi e dei permessi).

3)UIPackage: contiene tutte quelle classi che raffigurano le finestre della nostra GUI, da quella per l'accesso al sistema a quella per la visualizzazione di un'opera. Per gestire i dati in ingresso nelle varie form, i dati da mettere a video e le varie azioni dettate dai bottoni, queste classi istanziano oggetti della classe OperationManager a cui viene demandato il lavoro.

4)JDBC: si tratta di tutte quelle classi che vengono utilizzate per comunicare con il database.

5)JavaSwing: tutte quelle classi che vengono utilizzate per l'implementazione delle interfacce grafiche.