

Object Oriented Software Design 2017/2018

Progetto: *JLIBRARY*

Gruppo di lavoro:

Francesco Perilli: 250424

Giulio Marcantonio: 249201

Loris Savini: 247293

Raffaele Limongi: 247977

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 di consentire la consultazione dei manoscritti che devono essere digitalizzati e 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 sistema si divide nei seguenti sottoinsiemi:

1. Viewer
2. Uploader
3. Transcriber
4. Manager
5. Administrator

I sottosistemi sono indipendenti tra loro ma comunicano al fine di realizzare i loro scopi. In particolare, vengono descritti come segue:

Viewer

Tale parte del sistema consente la consultazione delle opere digitali a utenti registrati. Consente la ricerca nel catalogo per metadati(descritti in seguito) oppure all'interno del testo della trascrizione. Le opere possono essere suddivise in categorie. Appena si sceglie un'opera, verrà visualizzata con una schermata che avrà sulla destra il testo della trascrizione(se disponibile) e sulla sinistra l'immagine della pagina dell'opera che si sta visualizzando, sarà possibile sfogliare le pagine tramite un paginatore. Alcuni utenti con particolari privilegi possono effettuare il download dell'opera. L'utente può fare richiesta tramite un modulo per essere collaboratore del sistema(trascrittore). Gli utenti possono accedere al loro profilo personale dove saranno visualizzati i dati inseriti nella registrazione, tra cui: titolo di studio, professione , indirizzo, e-mail, etc...

Uploader

Ogni opera è formata da più immagini(scansioni), ognuna delle quali rappresenta una pagina del manoscritto. Per ogni opera vengono caricati dei metadati(titolo,anno,...). Al fine di rendere agevole il caricamento delle immagini e il successivo controllo, per ogni opera si vuole fornire la visualizzazione sia di tutte le pagine in miniatura e sia di una pagina per volta da scorrere con un paginatore. La digitalizzazione viene controllata da supervisori all'acquisizione per assicurarne la correttezza(ad esempio, in accordo con standard richiesti) e la qualità.

Transcriber

Ogni opera acquisita deve essere trasformata in un testo digitale, cio' avviene attraverso operazioni di trascrizioni in formato TEI(Text Encoding Initiative). Le trascrizioni sono digitalizzate manualmente attraverso un text editor TEI integrato. Quindi ogni immagine(pagina) avrà il corrispondente testo associato. Più trascrittori possono lavorare sulla stessa pagina, è necessario sincronizzare le modifiche. Le trascrizioni sono oggetto di revisione da parte di revisori alle trascrizioni.

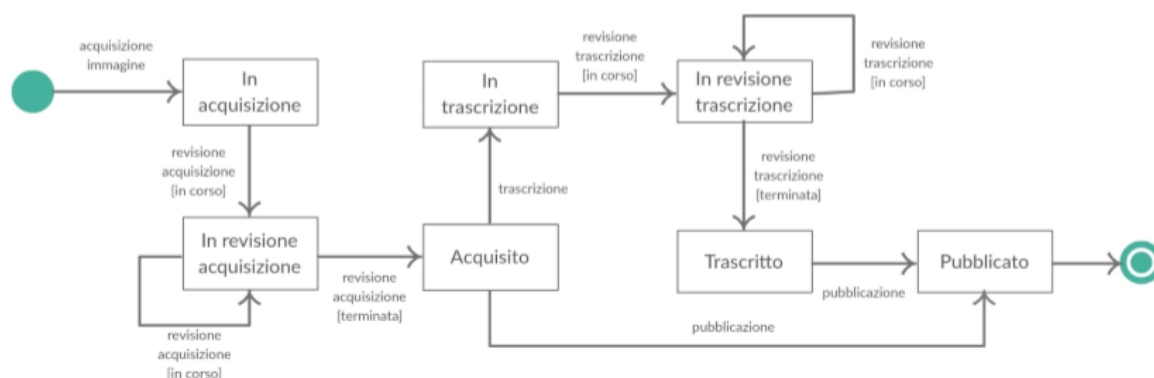
Manager

Questo sottosistema gestisce le assegnazioni, ovvero consente di assegnare parte di un'opera(1 o più immagini) a 1 o più trascrittori. Inoltre, consente di revisionare le trascrizioni concluse, di effettuare correzioni e validazione. E' possibile anche riassegnare delle pagine ai trascrittori. Consente la pubblicazione delle trascrizioni e delle opere(solo immagini). Gestisce i livelli dei trascrittori(ogni trascrittore ha un livello 1-5 in base alla sua esperienza). Consente la supervisione dell'acquisizione immagini.

Administrator

Gestione in back-end di tutto il sistema: anagrafica utenti, opere, etc..

Processo di pubblicazione:



1. Requirements

1.1 Documento dei requisiti

Il cliente richiede la creazione di una biblioteca online per la pubblicazione di manoscritti storici della città dell'Aquila, in cui gli utenti registrati al sito, possono visualizzare tali opere e fare delle ricerche nel relativo catalogo, e in cui dei supervisor, gestiscono e revisionano le trascrizioni caricate da utenti autorizzati. Nel progetto c'è bisogno di implementare un database, per la conservazione dei dati, utilizzando il linguaggio MySQL.

Per quanto riguarda il lato codice e l'interfaccia si userà il linguaggio di programmazione Java, che si interfacerà con il database attraverso specifiche implementazioni.

1.1.2 Requisiti della specifica

- Tutti gli utenti devono avere la possibilità di registrarsi al sito, successivamente potranno accedere al loro profilo personale dove è possibile consultare i dati inseriti nella registrazione, visualizzare le opere disponibili sul sito tramite una ricerca, effettuare una richiesta per essere collaboratore del sistema per caricare delle opere, ed ottenere permessi privilegiati per il download delle opere.
- I supervisor potranno accedere alla loro area personale, tramite login, nella quale potranno visualizzare le richieste da parte degli utenti, che vogliono collaborare all'inserimento delle opere. Ogni opera caricata da un utente autorizzato, verrà visionata dai supervisor per garantirne la corretta trascrizione e la qualità.
- L'amministratore invece gestirà tutta la parte dei dati caricata sul sito come: anagrafica utenti, opere, etc..

1.1.3 Requisiti del sistema

Il sistema dovrà garantire, grazie al corretto funzionamento della piattaforma web, di soddisfare tutti i requisiti appena dichiarati, quindi tutte le varie azioni che un utente loggato può effettuare.

Non sono stati preventivati ulteriori vincoli o funzionalità.

1.1.4 Requisiti interfaccia

- Nella homepage del sito, il sistema dovrà essere in grado di far visualizzare una pagina in cui sarà possibile effettuare il login e la registrazione.
- Il sistema dovrà garantire inoltre una pagina personale ad ogni utente, per permettergli di visualizzare ed eventualmente di modificare il resoconto della sua iscrizione cioè dati anagrafici, dati di accesso, opere scaricate, privilegi aggiuntivi, il loro livello in base alle trascrizioni pubblicate e i vari moduli compilabili e non.
- Ai supervisor invece, dovrà essere garantita una pagina per la gestione delle opere in attesa di approvazione, che potranno essere modificate, accettate, rifiutate e pubblicate dagli stessi supervisor, e una pagina per la gestione dei moduli inviati dagli utenti intenzionati a diventare trascrittori.
- L'amministratore del sistema invece avrà bisogno di una pagina per la gestione dei supervisor, degli utenti e delle opere.

1.1.5 Requisiti Funzionali

1.1.5.1 Registrazione da parte dell'utente alla libreria online

L'utente base, avrà la possibilità di registrarsi alla libreria online inserendo informazioni basilari da poter riutilizzare per effettuare il login e poter iniziare la lettura dell'opera scelta e di altre funzionalità specificate nei vari punti successivi.

Quando un utente si registra gli verrà attribuito di default il livello 0.

1.1.5.2 Visualizzazione del profilo e della homepage

Un utente loggato al sistema, avrà a disposizione una propria area riservata al suo profilo dove sarà in grado di consultare i dati inseriti nella registrazione ed eventualmente di modificarli.

Potrà inoltre consultare il suo livello da trascrittore, effettuare la richiesta tramite un modulo per diventare tale utente e infine attivare un livello vip per poter scaricare le opere.

Un utente, ogni volta che effettua un accesso sarà reindirizzato alla homepage, dove al suo interno potrà decidere se consultare il catalogo delle opere

presenti ed eventualmente effettuare una ricerca e infine potrà andare nella sezione dei trascrittori (nel caso in cui sia abilitato a farlo).

Quando seleziona un'opera verrà reindirizzato alla pagina contenente l'opera con i relativi metadati.

1.1.5.3 Possibilità di uploadare un'opera

Un utente loggato al sistema, indipendentemente dal grado, potrà decidere se caricare, tramite apposito bottone per scegliere i file, le immagini di un'opera sul sito.

Queste immagini verranno conservate nel database in attesa che un supervisor le revisioni e decida, tramite un apposito regolamento, di accettarla e pubblicare tale opera o rifiutarla.

1.1.5.4 Effettuare la visualizzazione di un'opera ed eventualmente scaricarla

Un utente, grazie alla home page, potrà andare a visualizzare un'opera presente nel catalogo della libreria, oppure potrà cercare un'opera tramite apposito form per la ricerca, dove potrà selezionare anche dei filtri per una ricerca più approfondita.

Una volta entrato nella pagina per visualizzare l'opera, l'utente, potrà visionare le immagini dell'opera ed eventualmente, se disponibile, la trascrizione scritta di tale opera.

Soltanto per gli utenti con un account vip sarà possibile visualizzare un bottone per effettuare il download dell'opera che gli permetterà di avere l'opera scaricata, e visualizzabile in offline.

1.1.5.5 Possibilità di trascrivere un'opera

L'utente, con i giusti privilegi da trascrittore dopo la richiesta tramite un modulo, sarà in grado di entrare nella sezione "trascrizione", dove sono presenti le varie trascrizioni in corso, e iniziare a contribuire alla realizzazione della trascrizione delle opere caricate, tutto ciò avviene grazie al lavoro dei supervisor che assegnano le trascrizioni ai vari utenti.

Una volta concluso il lavoro di trascrizione dell'opera, all'utente verrà assegnato un punteggio che sarà sommato al suo punteggio attuale per il livello di trascrittore.

1.1.5.6 Effettuare la revisione delle Immagini caricate

I supervisor, attraverso un pannello presente nella home page e visibile solo per tale utenza, devono revisionare le immagini che sono state caricate dagli utenti: nel caso in cui l'immagine/i risulta essere corretta e leggibile, viene prima approvata, tramite appositi bottoni, e successivamente verrà caricata nell'applicazione nella sezione opere, altrimenti verrà rifiutata e cancellata dal sistema.

Inoltre il supervisor avrà la possibilità di modificare l'immagine/i prima della pubblicazione .

1.1.5.7 Effettuare la Revisione/Modifica delle Trascrizioni

Così come per il punto precedente, i supervisor, attraverso un pannello per le trascrizioni presente nella home page e visibile solo per tale utenza, devono revisionare le trascrizioni che sono state scritte dagli utenti con il privilegio da trascrittore.

Quindi i supervisor, avranno il compito di effettuare tale revisione per queste trascrizioni e decidere se accettarle o meno a seconda del rispetto del regolamento stilato per tali trascrizioni.

Così come nel punto precedente se una trascrizione verrà accettata il supervisor potrà caricarla nella pagina dell'opera collegata, ma prima potrà anche effettuare una modifica al testo se necessario.

1.1.5.8 Gestione dell'User

I supervisor, tramite apposito pannello presente nella loro home page (visibile solo a loro), potranno settare il livello dell'utente relativo alle trascrizioni che tali hanno effettuato ed eventualmente anche modificarlo e potranno anche promuovere o rimuovere un utente base a seconda delle necessità.

1.1.5.9 Gestione delle Assegnazioni

I supervisor, dopo aver effettuato l'accesso, potranno gestire le assegnazioni di un utente per quella determinata trascrizione, tramite un apposito pannello presente nella sezione "trascrizione".

Li potranno decidere quale utente trascrittore assegnare per quella determinata opera, eventualmente modificare il numero dei trascrittori che possono partecipare e revocare l'incarico ad un utente trascrittore già assegnato a l'opera che deve essere trascritta.

1.1.5.10 Gestione dei Moduli

I supervisor, dopo aver effettuato il login, tramite apposito pannello posto nella loro home page e visibile solo a loro, potranno revisionare i vari moduli che gli utenti hanno completato per diventare dei trascrittori.

I supervisor nel caso di accettazione del modulo setteranno automaticamente l'utente base in questione, in un utente trascrittore, che sarà in grado di visualizzare il pannello "trascrizioni" per poter iniziare a trascrivere le opere caricate sull'applicazione, non appena verrà assegnato a una di queste.

Nel caso contrario, il supervisor non setterà l'utente base in un utente trascrittore, ma resterà l'utente base con i permessi di prima.

1.1.5.11 Gestione Back-end del Sistema

Gli utenti appartenenti al gruppo degli admin, possono rendere altri utenti base dei moderatori e viceversa, oppure rimuoverli completamente.

Possono modificare ed eventualmente rimuovere intere opere memorizzate all'interno del sistema e quindi gestire il tutto in back-end.

1.1.6 Requisiti Non funzionali

1.1.6.1 Availability: Il sistema dovrà poter garantire in qualsiasi momento tutte le sue funzionalità

1.1.6.2 Usability: essendo un'applicazione per una libreria di vecchi manoscritti, il sistema dovrà essere di facile utilizzo e di facile comprensione.

1.1.6.3 Reliability: il sistema dovrà garantire all'utente utilizzatore, le varie funzionalità offerte in modo affidabile e senza problemi.

1.1.6.4 Security: il sistema dovrà garantire un livello di sicurezza adeguato sia per quanto riguarda l'accesso e il salvataggio dei dati, e sia per i vari ruoli che un utente può avere, per esempio un utente base non può accedere alle funzionalità riservate ai moderatori o agli amministratori.

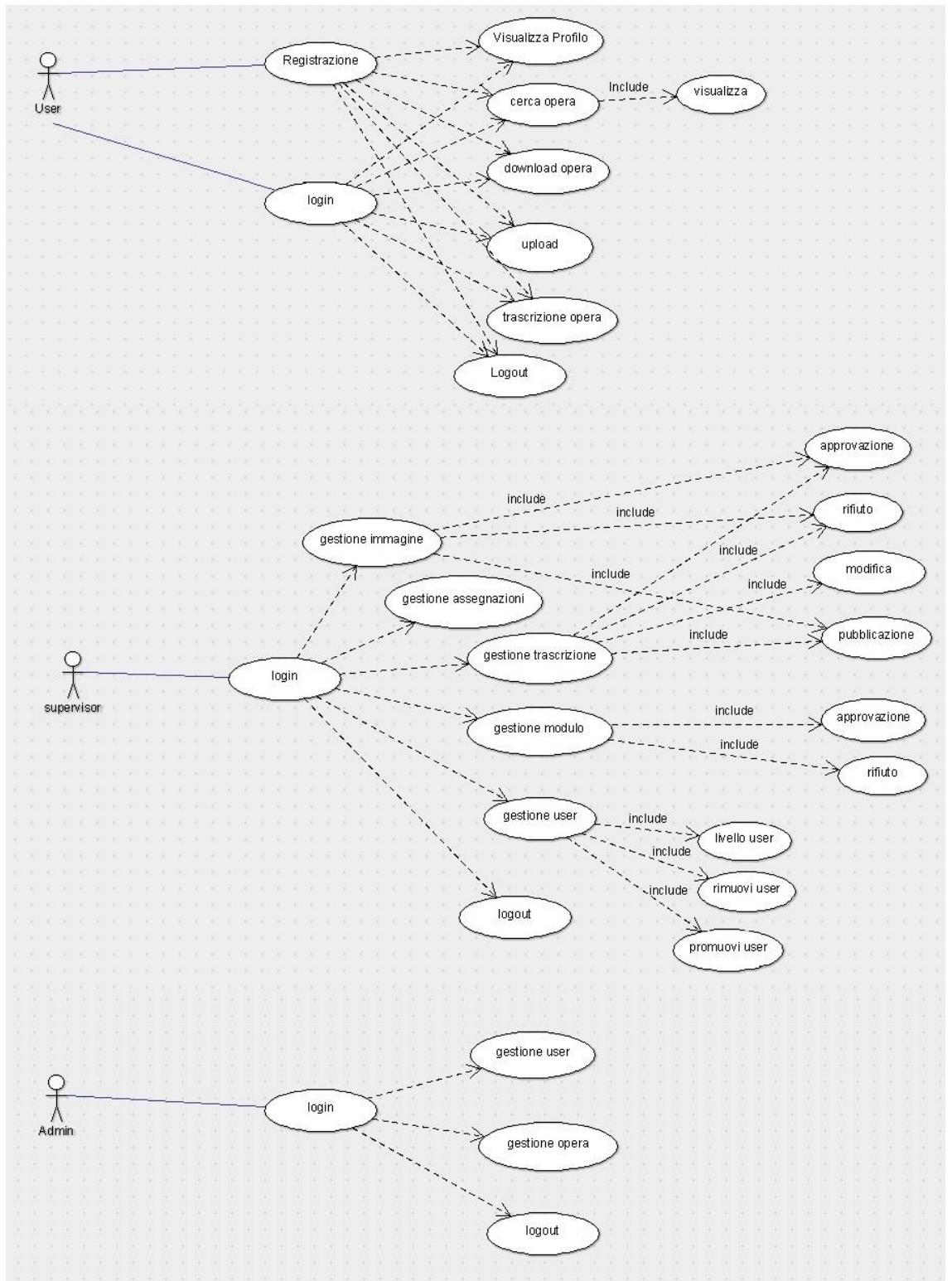
1.1.7 Requisiti DataBase

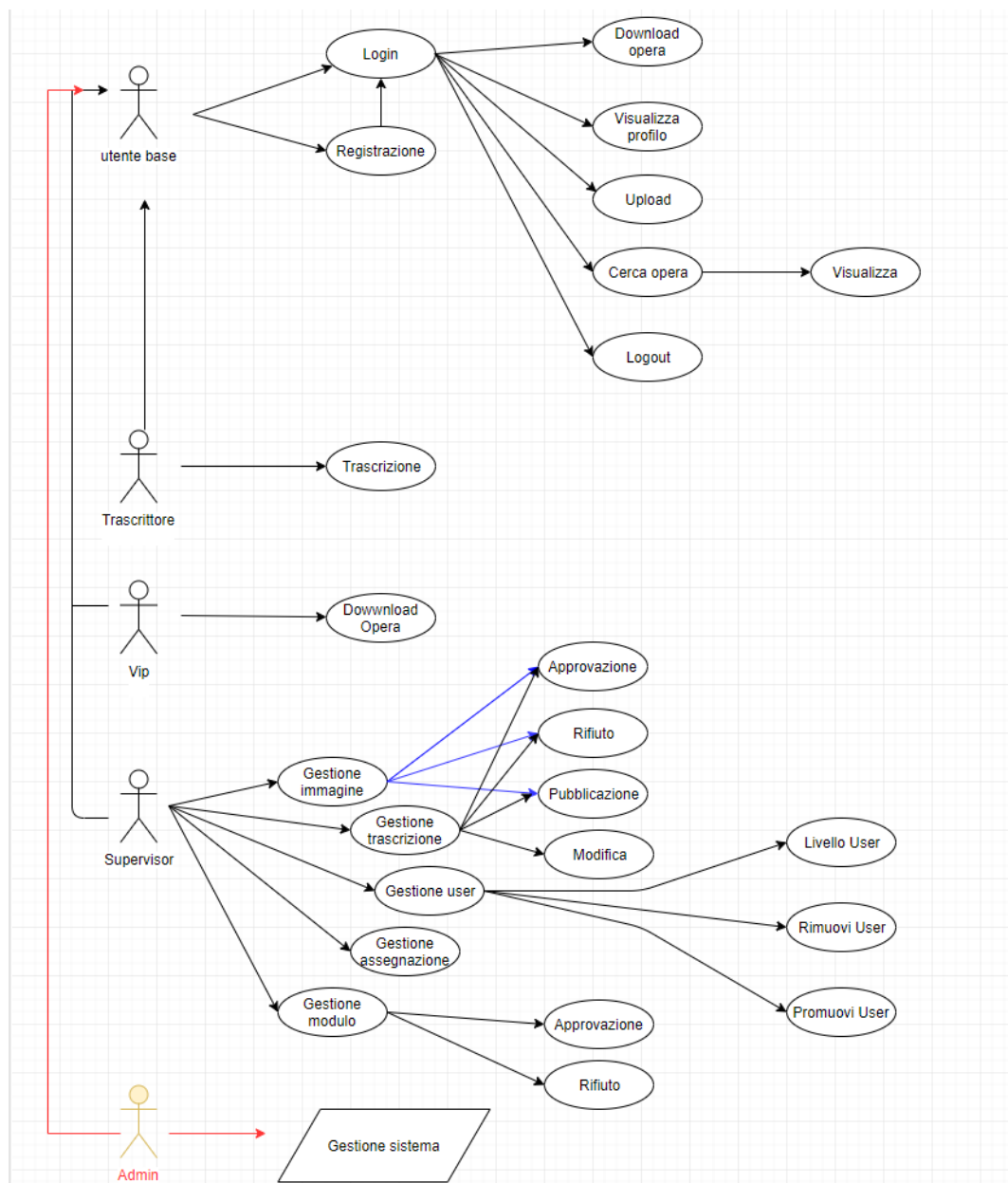
- Il database dovrà mantenere tutti i dati anagrafici di ogni singolo utente registrato al sito.
- Il database dovrà mantenere inoltre anche tutti i dati relativi ai manoscritti caricati e presenti sul sito.

1.2 Requisiti Casi d'uso

Abbiamo sviluppato tutti i possibili attori e casi d'uso mediante uno schema.

Questi casi d'uso ci saranno utili per definire successivamente i metodi utilizzati e per iniziare a capire il problema che successivamente modelleremo con il pattern da noi scelto.





Abbiamo individuato una quindicina di casi d'uso che i 3 attori della applicazione Utente Base, Supervisor, Admin potranno fare.

N.B. Gestione user(admin) è inteso come la rimozione/modifica dell'utente di qualsiasi privilegio.

Vale anche per gestione opera(rimozione/modifica).

Azione	Login
Attori coinvolti	Utente base, Supervisor, Admin

Descrizione	All'avvio dell'applicazione uno dei tre attori coinvolti effettuerà il login inserendo i suoi dati
Esecuzione	L'Utente base/Supervisor/Admin inserisce i suoi username e password
Risultato	L'attore accede alla home page/nella sua pagine del profilo o nella sua pagina gestionale
Errore	L'attore deve nuovamente effettuare il login

Azione	Registrazione
Attori coinvolti	Utente base
Descrizione	All'avvio dell'applicazione l'utente effettuerà una registrazione così da salvare i suoi dati personali all'interno del database
Esecuzione	L'Utente compila il form di registrazione inserendo i suoi dati personali insieme a username e password per accedere in futuro
Risultato	L'utente entra nella sua pagina profilo
Errore	L'attore deve nuovamente effettuare la registrazione

Azione	Visualizza Profilo
Attori coinvolti	Utente

Descrizione	All'interno della sua pagina l'utente potrà visualizzare le sue informazioni personali
Esecuzione	L'Utente tramite il menu a tendina andrà su "visualizza profilo".
Risultato	Vengono visualizzate sullo schermo le informazioni riguardanti l'utente
Errore	Questa azione non dovrebbe ritornare nessun errore

Azione	Cerca Opera
Attori coinvolti	Utente
Descrizione	L'utente sarà in grado di cercare delle opere nella home page tramite un form apposito oppure filtrare le ricerche in base alla categoria.
Esecuzione	L'Utente compila un form nella home page per la ricerca delle opere tramite titolo/nome dell'autore oppure una ricerca filtrata per il genere dell'opera.
Risultato	Viene restituita una lista di opere filtrate tramite i parametri di ricerca inseriti dall'utente.
Errore	Non viene restituito alcun risultato corrispondente.

Azione	Visualizza Opera
Attori coinvolti	Utente

Descrizione	L'utente nella sua pagina profilo potrà accedere alla pagina dell'opera da visualizzare, oppure dopo aver effettuato una ricerca tramite form apposito.
Esecuzione	L'Utente tramite un menù a tendina presente nella sua pagina profilo selezionerà "opere" e gli verrà restituita una lista di opere disponibili dove potrà selezionarne una, oppure tramite ricerca con un form apposito.
Risultato	L'utente visualizzerà correttamente l'opera precedentemente aperta.
Errore	Viene restituito un messaggio di errore e l'utente viene mandato nella sua pagina profilo.

Azione	Download Opera
Attori coinvolti	Utente Vip
Descrizione	L'utente con privilegi vip, sarà in grado di scaricare delle opere nell'apposita pagina di visualizzazione di tali opere.
Esecuzione	L'Utente tramite il bottone "download" presente nella pagina di visualizzazione di un'opera potrà scaricare l'opera che sta visualizzando in quel momento.
Risultato	Viene restituito un messaggio di corretto avvio di download.
Errore	Viene restituito un messaggio di errore e l'utente viene mandato nella sua pagina profilo.

Azione	Upload
Attori coinvolti	Utente base
Descrizione	L'utente all'interno della sua pagina sarà in grado di caricare le immagini delle opere con relative informazioni aggiuntive.
Esecuzione	L'Utente dal menù a tendina presente nella sua pagina profilo selezionerà "Upload opera" tramite un apposito bottone.
Risultato	Viene restituito un messaggio di corretto caricamento delle immagini.
Errore	Viene mostrato un messaggio di errore e l'utente viene fatto tornare nella sua pagina profilo.

Azione	Trascrizione Opera
Attori coinvolti	Utente trascrittore
Descrizione	L'utente con i privilegi da trascrittore potrà digitalizzare in testo le immagini delle opere accettate che gli sono state assegnate da un supervisor.
Esecuzione	L'utente con privilegi da trascrittore accede, tramite il menù a tendina presente nella sua pagina profilo, alla sezione "Trascrizioni opere", dove potrà trascrivere il testo delle immagini già presenti nella sezione.
Risultato	Viene restituita una lista di opere da trascrivere che gli sono state assegnate da un supervisor.

Errore	L'utente viene rimandato nella sua pagina profilo.
--------	--

Azione	Gestione immagine(accept/decline/upload)
Attori coinvolti	Supervisor
Descrizione	Il supervisor sarà in grado di visualizzare la lista delle immagini caricate in attesa di approvazione. Potrà decidere se approvarla o meno ed eventualmente pubblicarla.
Esecuzione	Il Supervisor all'interno della sua pagina profilo selezionerà "Gestione Immagini". Gli verrà restituita una lista di immagini delle opere in attesa di revisione che potrà accettare o meno e successivamente pubblicarla.
Risultato	Viene restituito un messaggio di corretta approvazione/rifiuto/pubblicazione e tornerà nella sua pagina profilo.
Errore	Viene stampato un messaggio d'errore con ritorno alla pagina di profilo.

Azione	Gestione assegnazioni(trascrittori)
Attori coinvolti	Supervisor
Descrizione	Il supervisor sarà in grado di assegnare parte di un opera(1 o più immagini) ad 1 o più trascrittori.

Esecuzione	Il supervisor all'interno della sua pagina profilo selezionerà "Gestione assegnazioni". Gli verrà restituita una lista di utenti disponibili ed abilitati alla trascrizione, a cui assegnerà 1 o più immagini ad uno a diversi user.
Risultato	Viene restituito un messaggio di corretta assegnazione e tornerà nella sua pagina profilo.
Errore	Viene stampato un messaggio di errore con ritorno alla pagina di profilo.

Azione	Gestione trascrizione(accept/decline/modify/upload)
Attori coinvolti	Supervisor
Descrizione	Il supervisor visualizzerà la lista delle trascrizioni aperte/concluse e scegliendone una potrà decidere se accettarla/rifiutarla/modificarla/caricarla.
Esecuzione	Il supervisor all'interno della sua pagina profilo selezionerà "Gestione trascrizioni". Gli verrà restituita una lista delle trascrizioni aperte o terminate. Scegliendo una trascrizione conclusa potrà decidere se accettarla/rifiutarla/modificarla/caricarla.
Risultato	Viene stampato un messaggio di corretta accettazione/rifiuto/modifica/caricamento della trascrizione e il supervisor tornerà nella sua pagina profilo.
Errore	Viene stampato un messaggio di errore e il supervisor tornerà alla pagina di profilo.

Azione	Gestione Modulo(accept/decline)
Attori coinvolti	Supervisor
Descrizione	Il supervisor visualizzerà la lista dei moduli, inviata dagli utenti per diventare dei trascrittori, ancora da approvare e scegliendone una alla volta potrà decidere se accettarla o meno.
Esecuzione	Il supervisor all'interno della sua pagina profilo selezionerà "Gestione moduli". Gli verrà restituita una lista di moduli in attesa di risposta. Il supervisor scegliendone una potrà decidere se accettarla o meno.
Risultato	Viene stampato un messaggio di corretta accettazione/rifiuto del modulo e il supervisor tornerà nella sua pagina profilo.
Errore	Viene stampato un messaggio di errore e il supervisor tornerà alla pagina di profilo.

Azione	Gestione user(remove/promote/modify)
Attori coinvolti	Supervisor
Descrizione	Il supervisor visualizzerà la lista di tutti gli utenti base e potrà promuovere un utente scelto, oppure momentaneamente rimuoverlo(ban). Infine potrà modificare il suo livello di trascrittore.
Esecuzione	Il supervisor all'interno della sua pagina profilo selezionerà "gestione utente". Verrà stampata una lista di utenti e il supervisor scegliendone uno potrà darli i permessi da supervisor o rimuoverlo temporaneamente, oppure modificare il livello da trascrittore

Risultato	Viene stampato un messaggio di corretta promozione/modifica/rimozione(ban) e il moderatore tornerà nella sua pagina profilo
Errore	Viene stampato un messaggio di errore e il supervisor tornerà alla pagina di profilo.

Azione	Gestione user_admin(remove,modify,promote)
Attori coinvolti	Admin
Descrizione	L'amministratore seleziona da una lista di utenti quello da rimuovere/modificare/promuovere dal DB
Esecuzione	Da un menù a tendina presente nella sua pagina profilo selezionerà "gestione utente". Verrà stampata una lista di utenti e l'amministratore scegliendone uno lo rimuoverà/modificherà/promuoverà.
Risultato	Verrà stampato un messaggio di corretta eliminazione/modifica/promozione dell'utente.
Errore	Viene stampato un messaggio di errore e l'admin viene fatto tornare nella sua pagina profilo

Azione	Gestione opera
Attori coinvolti	Admin
Descrizione	L'amministratore seleziona da una lista di opere disponibili quella da

	rimuove dal DB o eventualmente modificarla.
Esecuzione	Da un menù a tendina presente nella sua pagina profilo selezionerà "gestione opera". Verrà stampata una lista di opere e l'amministratore scegliendone una potrà rimuoverla dal DB o eventualmente modificarla.
Risultato	Verrà stampato un messaggio di corretta eliminazione/modifica dell'opera.
Errore	Viene stampato un messaggio di errore e l'admin viene fatto tornare nella sua pagina profilo.

Azione	Logout
Attori coinvolti	Utente Base, Supervisor, Admin
Descrizione	L'attore effettua il logout e termina la sua sessione corrente
Esecuzione	Da un menù a tendina presente nella sua pagina profilo sarà disponibile il logout per terminare la sessione
Risultato	L'Attore verrà rimandato nella pagina iniziale, dove potrà di nuovo effettuare il login
Errore	Non dovrebbero esserci messaggi d'errore in questa azione

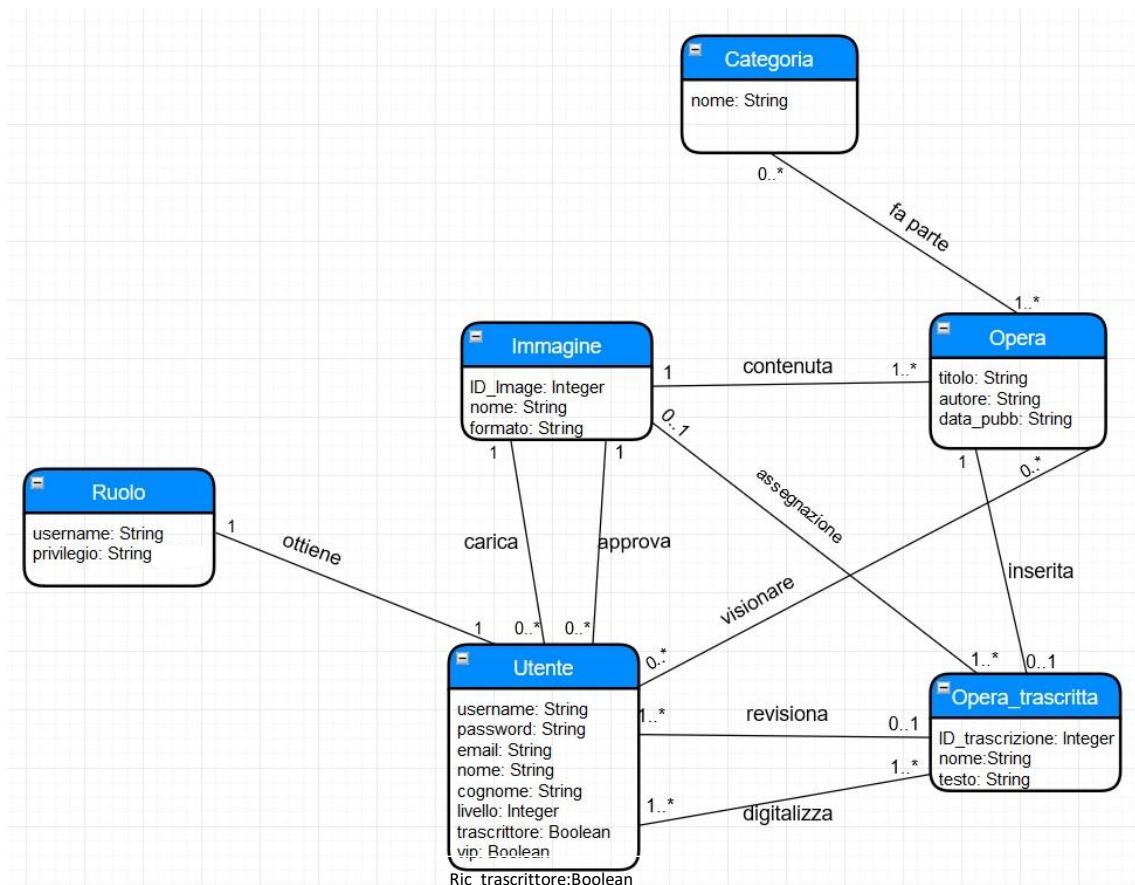
1.3 **Modello di dominio**

Le classi di dominio trovate in questa fase di progettazione sono:

- Utente,
- Opera,

- Immagine,
- Opera-Trascritta,
- Ruolo,
- Categoria.

Il relativo modello è il seguente:

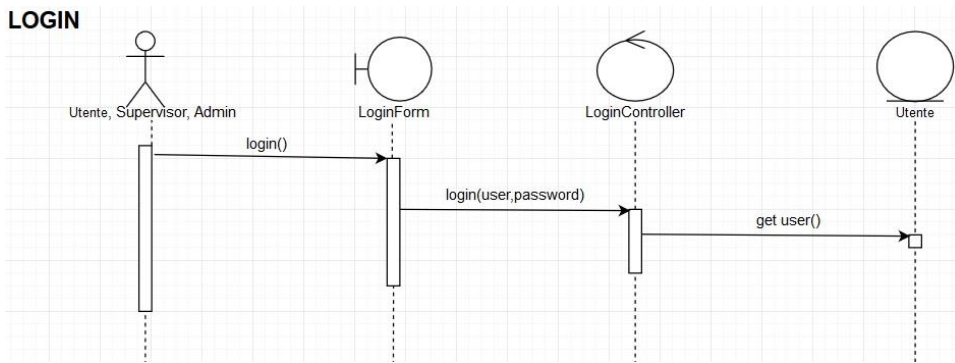


1.4 Individuazione di classi entity, boundary, controller

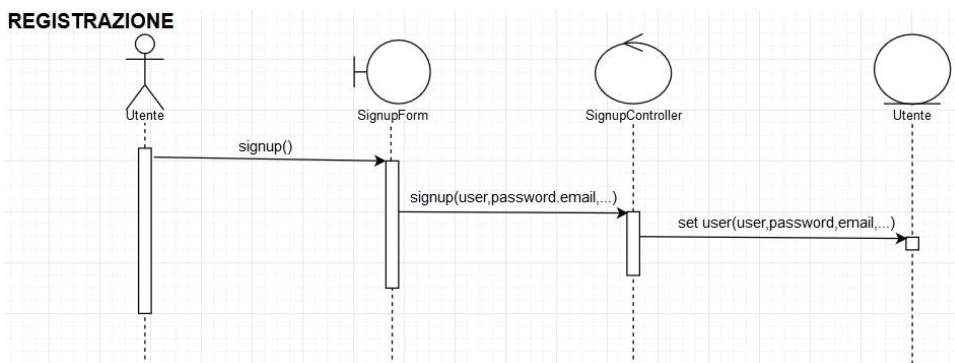
Analizzando i vari use-case trovati e completando il modello di dominio siamo riusciti a dare una prima bozza del funzionamento della nostra applicazione.

Per migliorare ancora tutto ciò fatto fin ora, abbiamo deciso di realizzare dei diagrammi sfruttando il pattern ECB(entity-boundary-control) per ogni use case che coinvolge un attore.

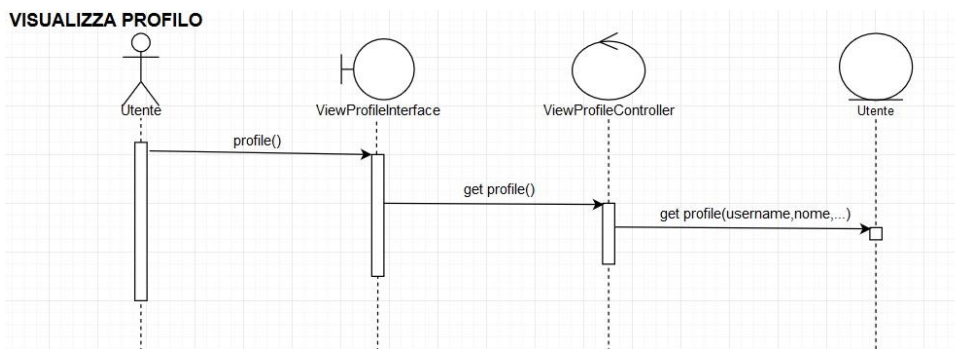
LOGIN



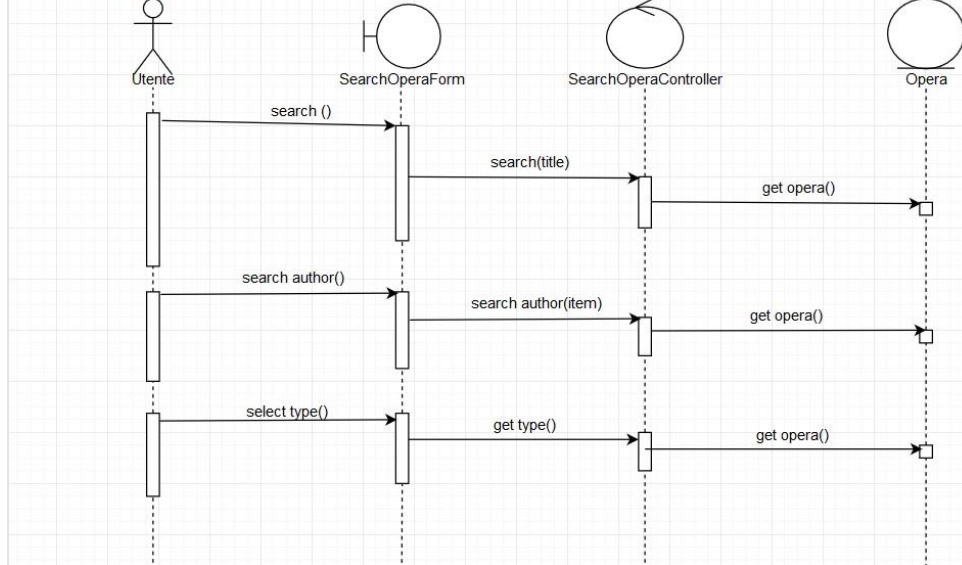
REGISTRAZIONE



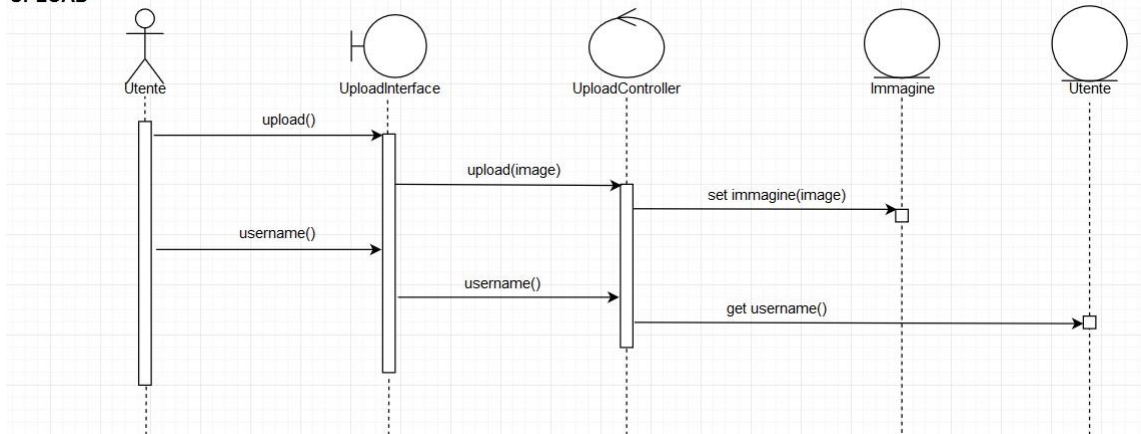
VISUALIZZA PROFILO



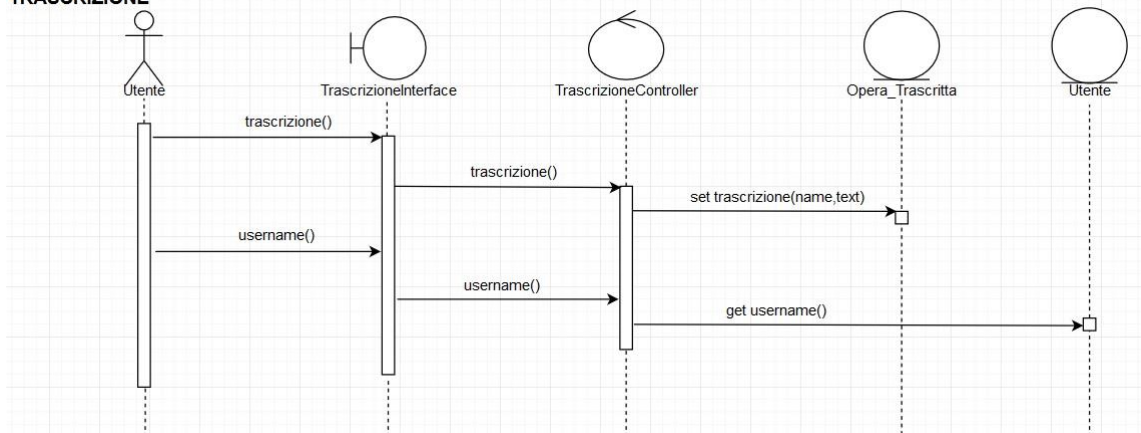
CERCA OPERA



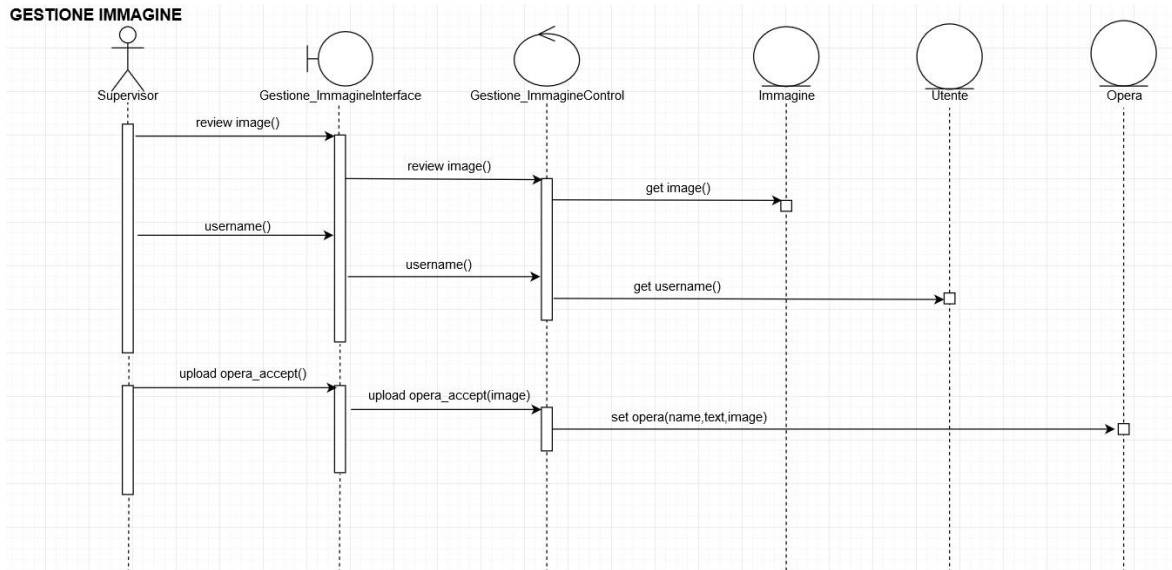
UPLOAD



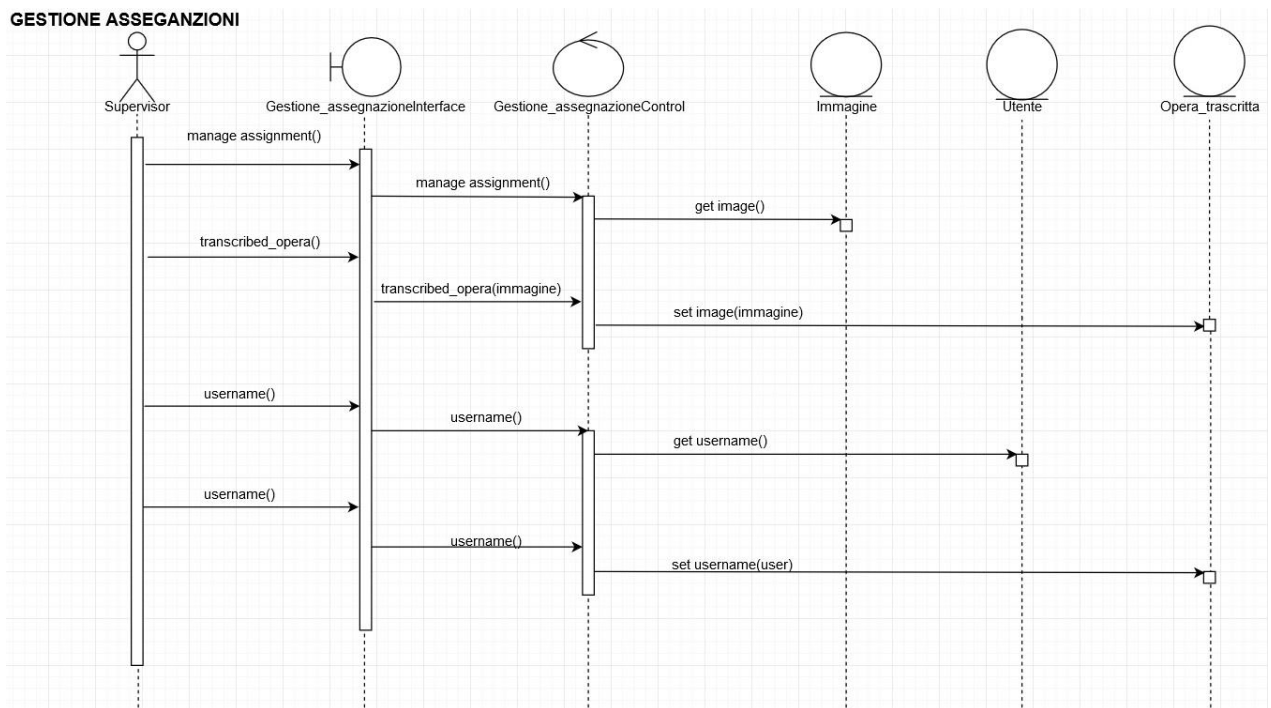
TRASCRIZIONE



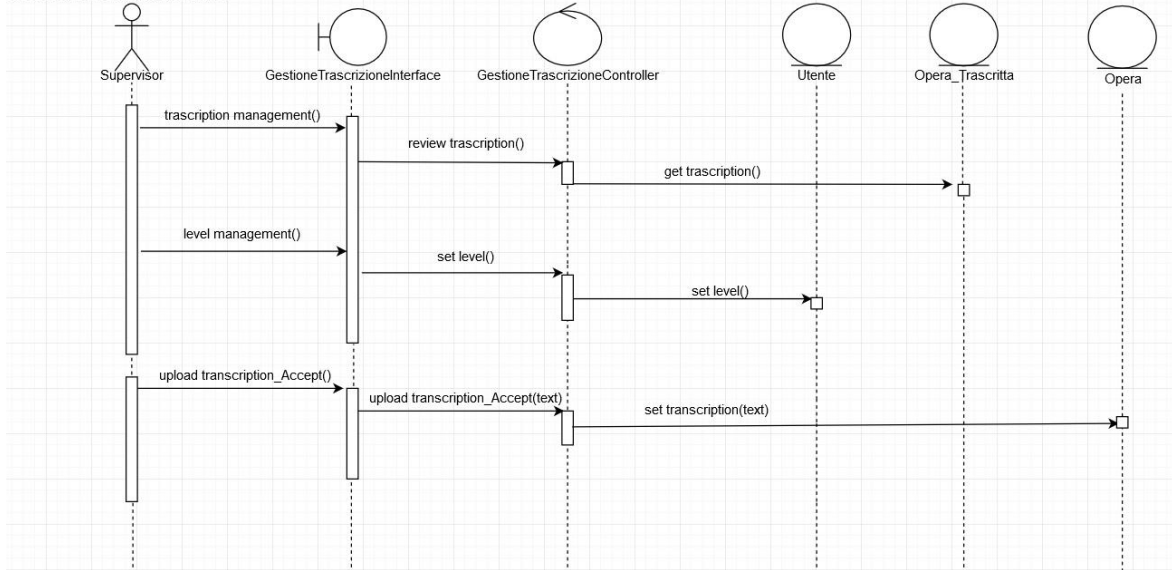
GESTIONE IMMAGINE



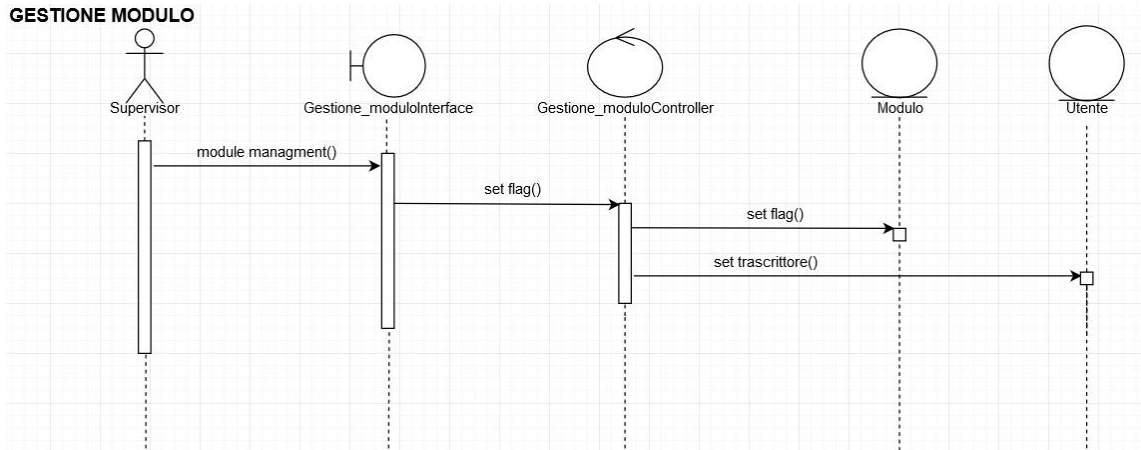
GESTIONE ASSEGNAZIONI



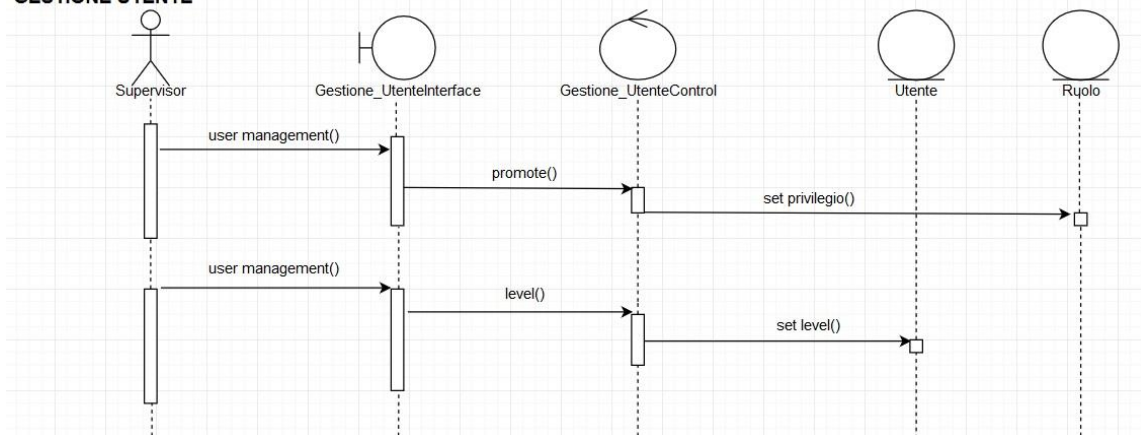
GESTIONE TRASCRIZIONE



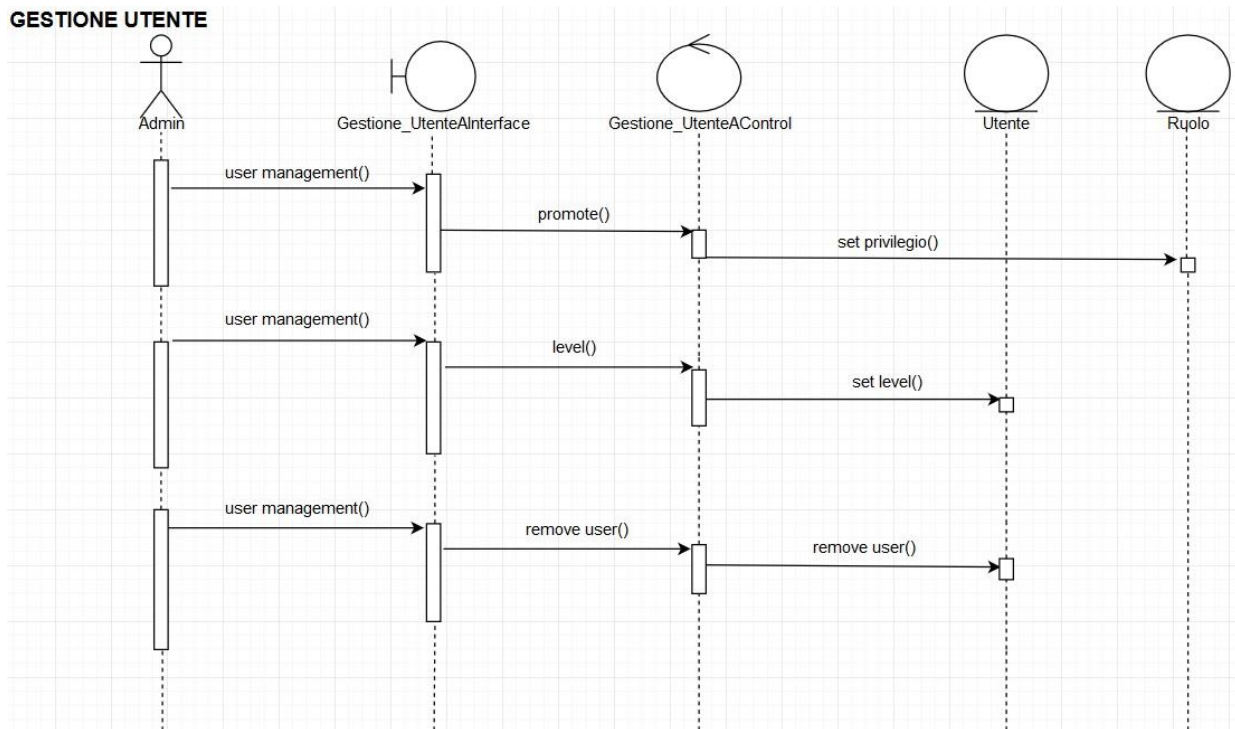
GESTIONE MODULO



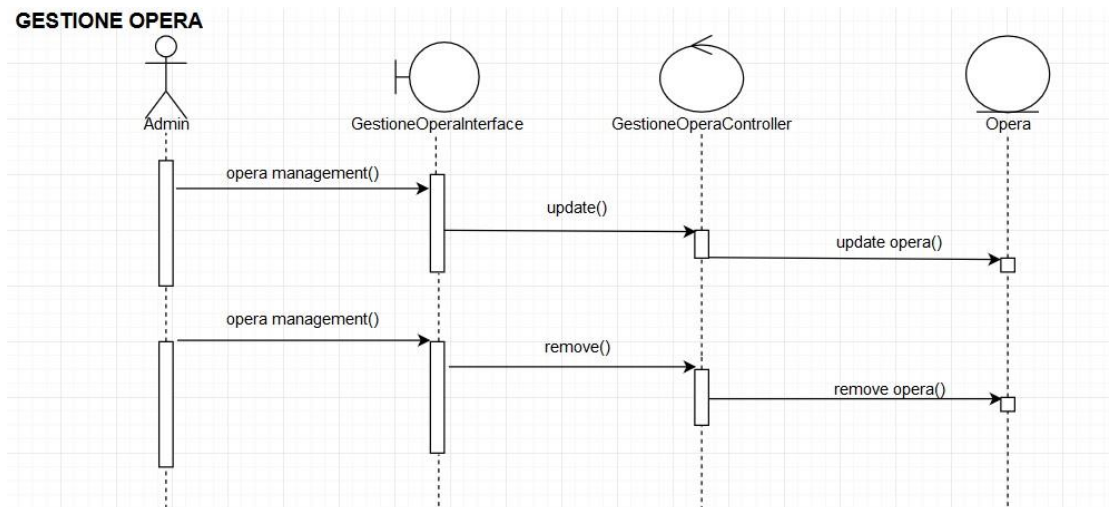
GESTIONE UTENTE



GESTIONE UTENTE

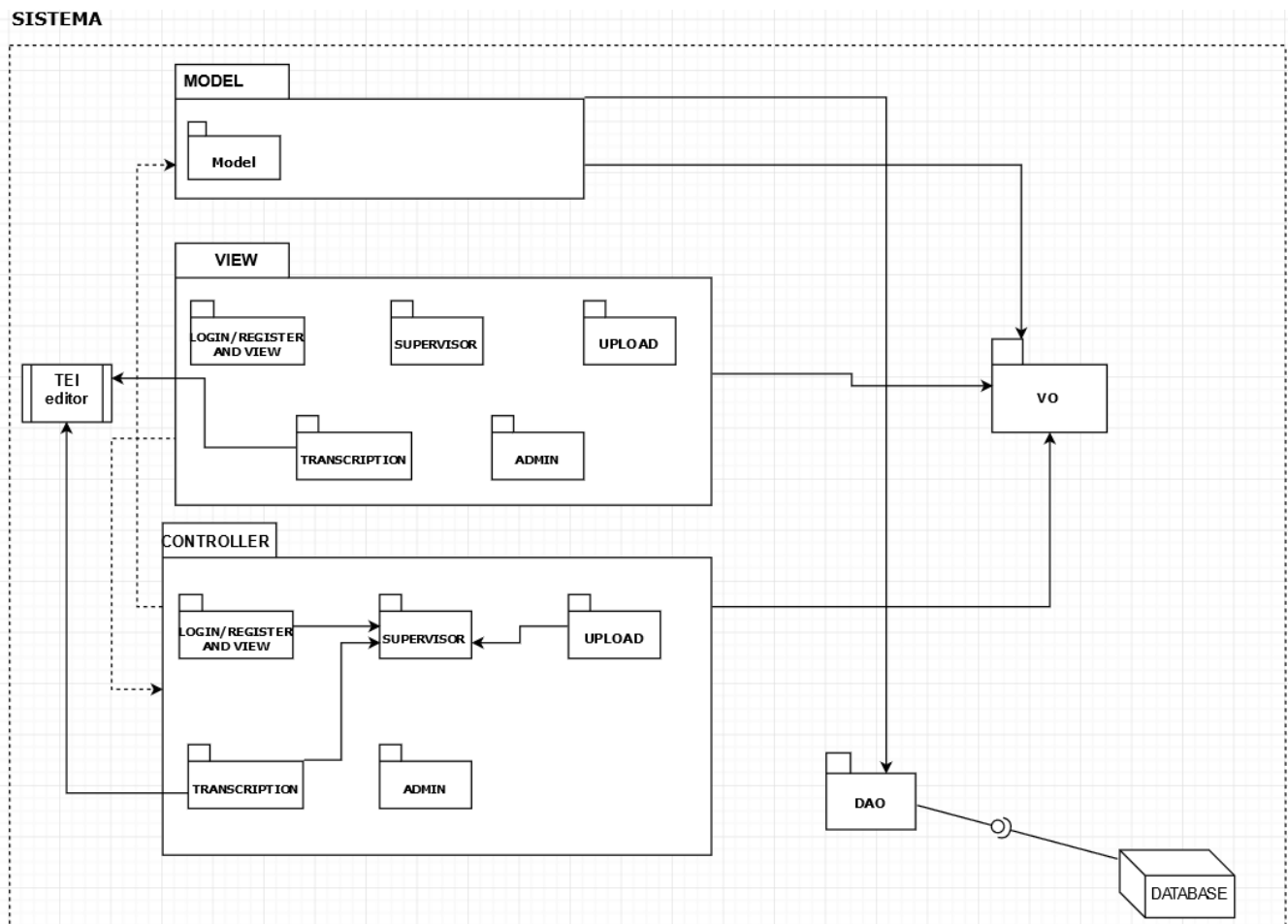


GESTIONE OPERA



2. System Design

2.1 Modello dell'architettura software

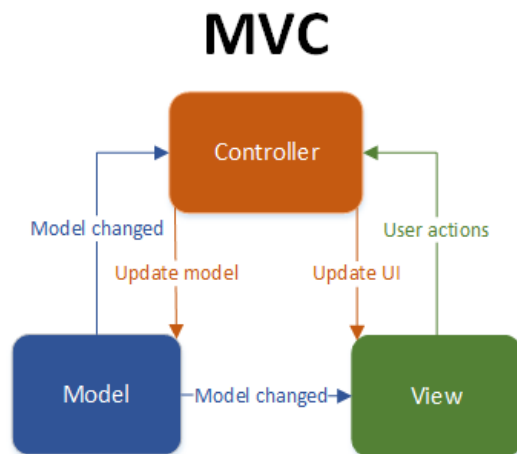


2.2 Descrizione dell'architettura

Come architettura software abbiamo scelto il pattern MVC(Model View Controller), affiancato dai seguenti pattern architetturali DAO(Data Access Object) che è un design pattern che fornisce un'interfaccia astratta ad alcuni tipi di database tra cui SQL, mappando le chiamate della business logic sul modello di persistenza dati, VO(Value Object) che è un design pattern contenente i valori delle classi, dove questi valori sono immutabili cioè invece di modificare gli oggetti dei valori correnti, creiamo nuove

istanze che racchiudono i nuovi valori, e infine il TEI editor, utilizzato per effettuare la trascrizione delle immagini delle opere caricate.

Il pattern MVC è uno dei più utilizzati per la programmazione ad oggetti, in grado di separare in modo efficace la logica di presentazione dei dati dalla logica di business e in grado di adattarsi ad ogni problema.



L'architettura scelta presenta 3 elementi fondamentali che la compongono, ovvero stiamo parlando del:

- **Model:**

E' la parte fondamentale di questa architettura, dove un'applicazione trova il suo **core** principale, ovvero il model grazie all'incapsulamento dello stato dell'applicazione riesce a definire e gestire i dati e le operazioni che possono essere eseguite su questi.

Contiene al suo interno delle classi e degli oggetti che sono manipolati dal controller e che vengono poi inseriti e salvati nel database.

Tramite l'infrastruttura di interfacce **DAO** si ha un'iterazione modulare con il database e di conseguenza l'astrazione del livello dati da quello controller.

- **View:**

Nella view invece vengono gestiti gli insiemi degli elementi per la creazione dell'interfaccia grafica, utilizzata successivamente dagli utenti.

Utilizzeremo **JAVA FX** per la realizzazione grafica della nostra applicazione, che poi linkeremo con le varie funzionalità trovate e che verranno scritte in java.

- **Controller:**

Il controller infine ha il compito di trasformare le interazioni dell'utente della view in azioni eseguite dal model.

Più precisamente realizza la mappatura del input fornito dall'utente e i processi eseguiti dal model, infine il controller implementa la logica di controllo dell'applicazione.

Quindi al suo interno contiene le varie classi e metodi che permettono la gestione del sistema, le quali si occupano della manipolazione dei dati.

Database: L'architettura si conclude con un database relazione che implementa la persistenza dei dati e che quindi serve per memorizzare fisicamente tali i dati relativi al sistema che ci interessa conservare.

2.3 Descrizione delle scelte e strategie adottate

Nel model sono presenti le classi estrapolate grazie al modello di domino costruito al punto 1.3.

Grazie al model, che come detto in precedenza è il core di un'applicazione, potremo comunicare tramite il package **DAO**, usato per il collegamento con il database e che presto verrà definito, con il database per l'inserimento e la modifica dei dati.

Nella view abbiamo inserito vari sotto-package tra cui login/register & view, dove abbiamo messo insieme la fase di login/registrazione e la fase di visualizzazione dell'applicazione, e il sotto-package transcription per la gestione delle trascrizioni che infatti si linkerà con l'editor di testo TEI utilizzato per le trascrizioni.

La view potrà fare riferimento al package VO per l'uso di classi maggiormente utilizzate(generali),così come il model e il controller.

La view infine avrà un matching con il package controller per l'invio delle interazioni dell'utente nella fase di view, dove il controller avrà il compito di realizzare la mappatura dell'input(azioni effettuate) per mandare al model.

Così è strutturato anche il controller che comunque in più sarà in grado di gestire le varie classi all'interno del model per eventuali modifiche o chiamate.

Un altro design pattern utilizzato è il **singleton**.

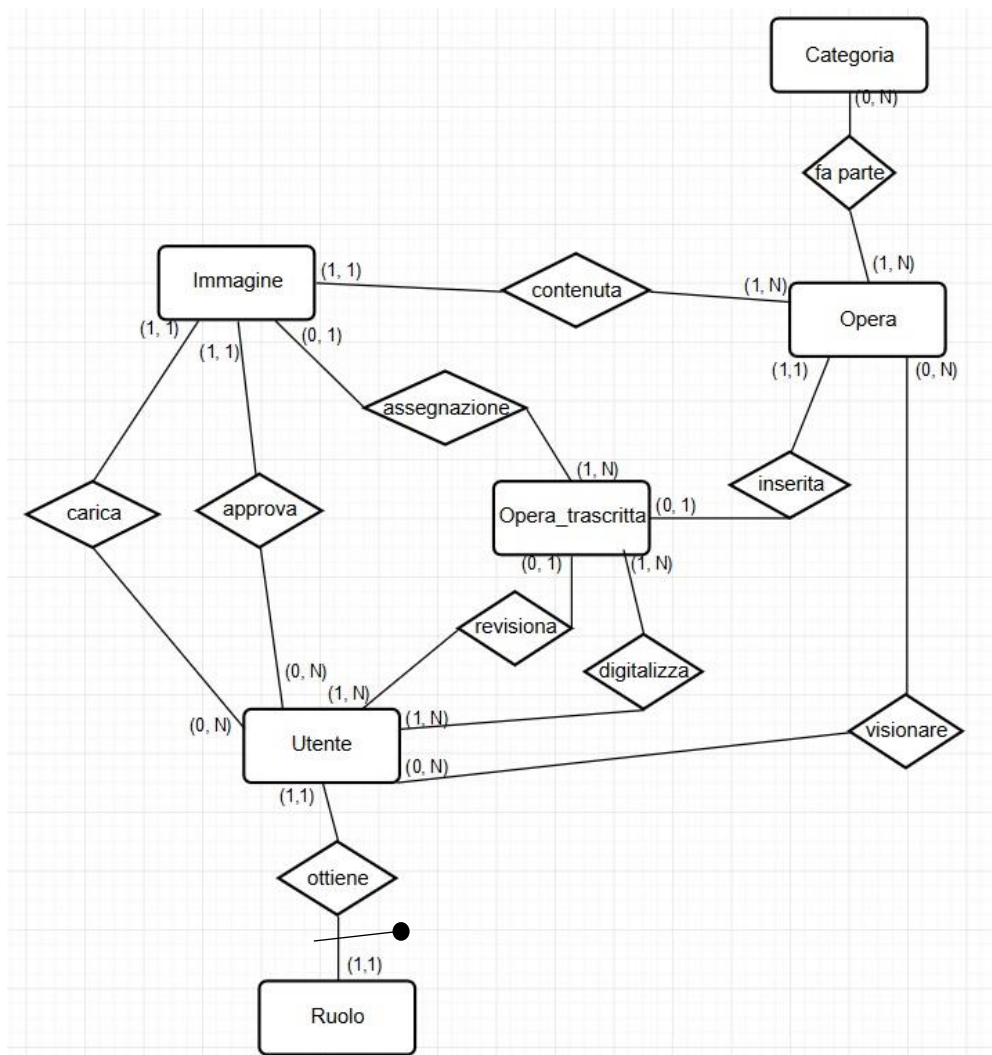
Il **singleton** è un design pattern creazionale che ha lo scopo di assicurarsi che venga creata una sola ed unica istanza di una classe.

Nel dettaglio l'abbiamo utilizzato per la classe **UserModel**, perché conviene avere un'unica istanza dell'utente che si logga al sistema e che quindi dopo il log le informazioni principali di tale utente verranno messe nell'unica istanza dell'UserModel per risparmiare varie operazioni di query in caso servano informazioni come per la visualizzazione del profilo.

Infatti una volta entrati nel profilo non viene effettuata una query ogni volta per selezionare le informazioni e inserirle nei campi appositi, ma invece tali informazioni le andremo a reperire nell'unica istanza creata grazie al singleton.

Solo in un eventuale update andremo ad effettuare una query per poi riaggiornare l'istanza dell'UserModel.

Dopo un'attenta analisi del problema, abbiamo strutturato il seguente modello E-R per facilitare la comprensione dell'intero progetto e una sua successiva strutturazione:

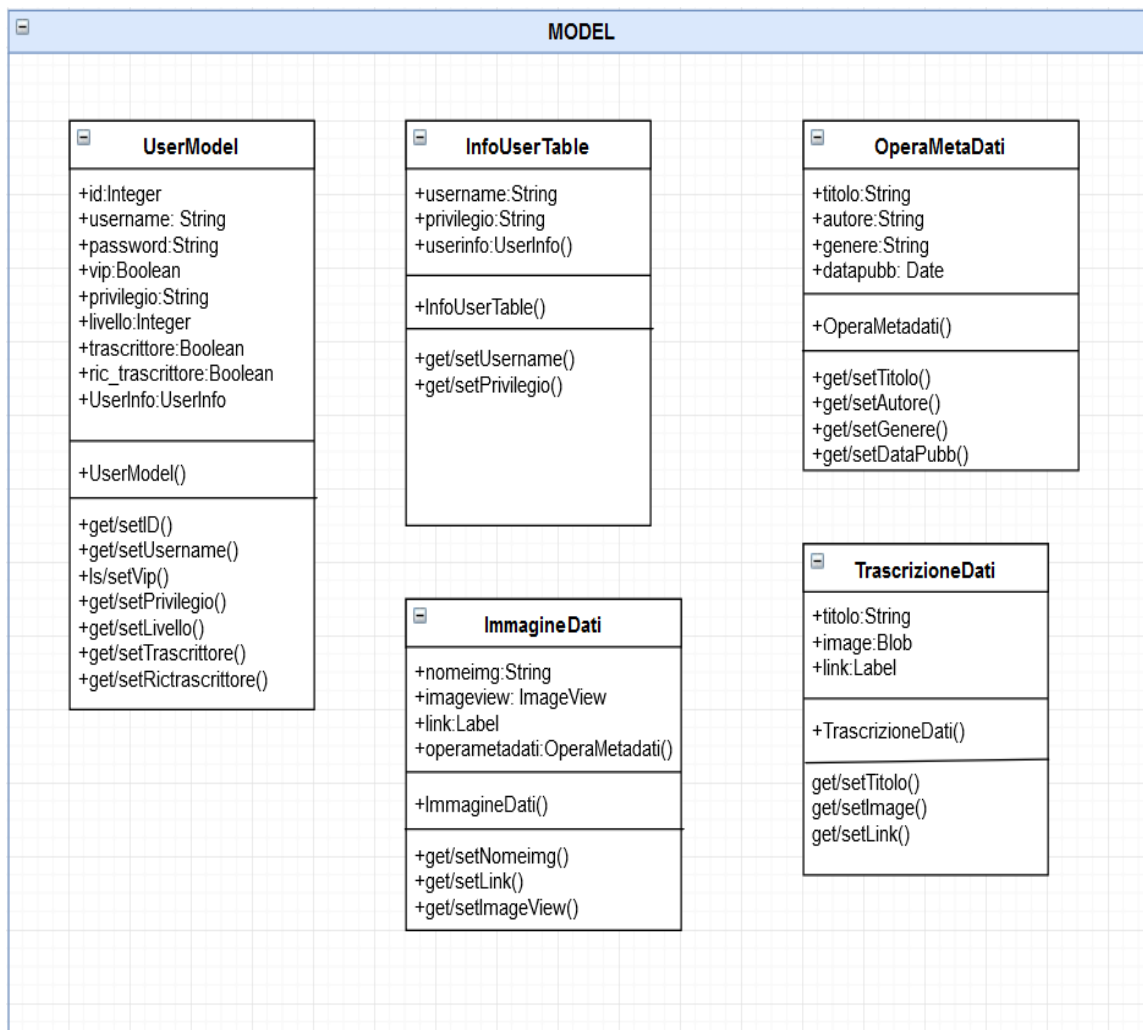


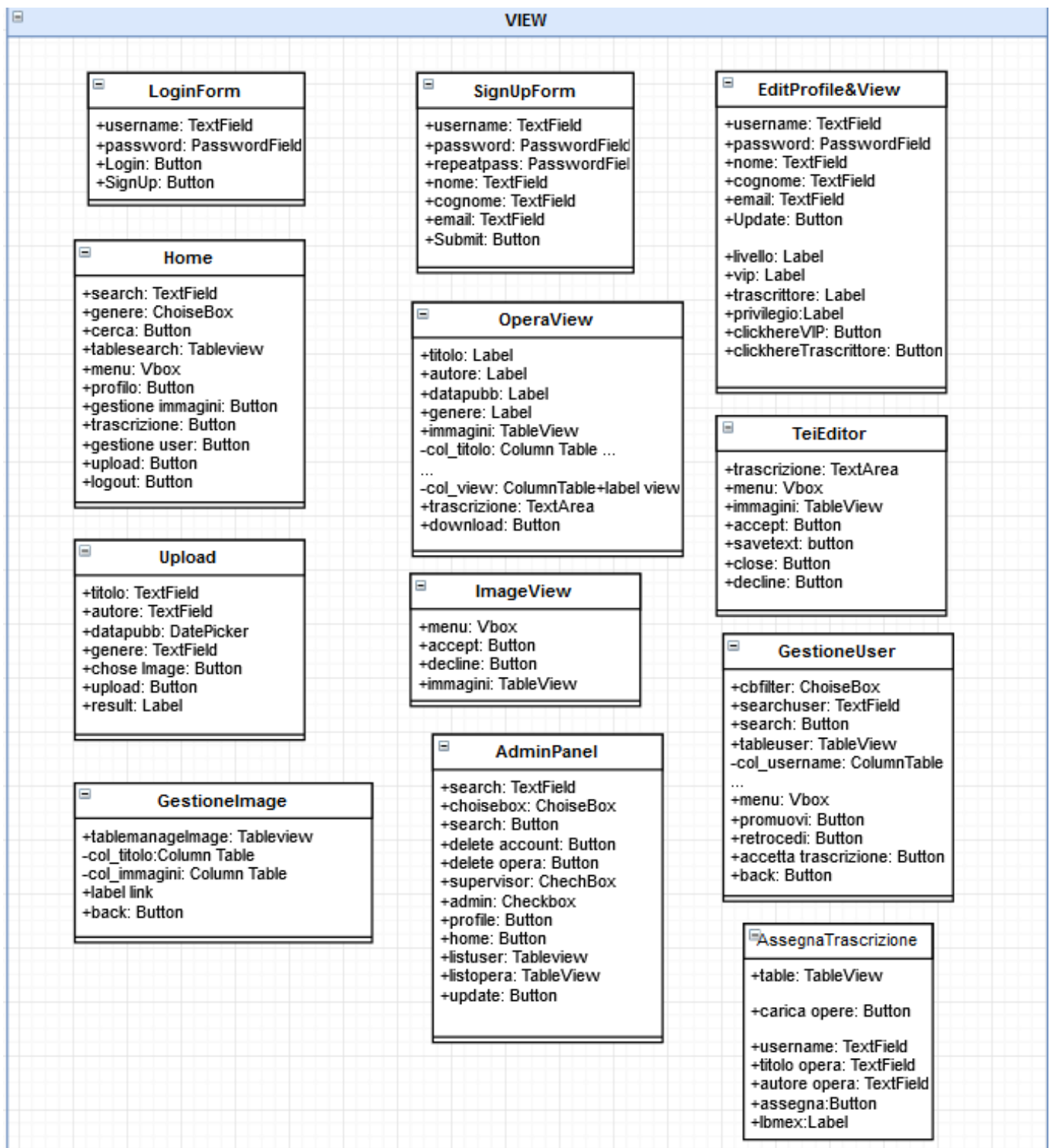
- L'entità **Utente** conserva tutte le persone coinvolte nell'applicazione: Utente Base, supervisor e admin.
- Un **utente base** può visionare zero o più opere e può caricare zero o più immagini.
- Un **utente supervisor** può revisionare zero o più trascrizioni, può approvare zero o più immagini e successivamente se approvate pubblicarle, può retrocedere o promuovere degli utenti.
- L'**Admin** può eliminare utenti, opere, supervisor e gestisce tutti i dati conservati nel DB.
- Un **Immagine** può essere caricata da un unico utente, moderata da un unico supervisor, contenuta in un'unica opera e assegnata ad un'unica trascrizione.
- Un **Opera** può fa parte di nessuna o più categorie, può essere inserita al massimo una trascrizione, può essere visionata da più utenti, vengono contenute più immagini.
- Una **Categoria** può contenere più opere.

- Nell'entità **ruolo** è specificato il ruolo di ogni utente: Base, supervisor, admin.
- **Un Opera trascritta** può essere inserita in un'unica opera, può essere digitalizzata da più utenti, può essere revisionata da più supervisor e può contenere almeno una o più immagini.

3. Software/Object Design

3.1 UML Class Diagram





CONTROLLER

LoginController

```
+LoginController()
+Login
+gotosignup()
```

ViewOperaController

```
+ViewOperaController()
+ViewOpera()
+LoadOpera()
+Download()
+setTable()
+initialize()
```

TeiEditor

```
+TeiEditor()
+setscene()
+LoadText()
+SaveText()
+Close()
+Accept()
+Decline()
+setTable()
+initialize()
```

AssegnazioneTrascController

```
+AssegnazioneTrascController
+setscene
+CaricaOpere()
+Assegna()
+initialize()
+setTable()
```

GestioneUserController

```
+GestioneUserController()
+setscene()
+search()
+PromuoviUser()
+RetrocediUser()
+AcceptTrascrittore()
+gotohome()
+setTable()
+initialize()
```

SignUpController

```
+SignUpController()
+Submit()
+gotologin()
```

GestioneImmaginiController

```
+GestioneImmaginiController()
+setscene()
+gotohome()
+LoadInfoOpera()
+initialize()
+setTable()
```

UploadController

```
+UploadController()
+setscene()
+chosedImage()
+Upload()
+initialize()
```

HomePageController

```
+HomePageController()
+setscene()
+search()
+returnlogin()
+gotoprofile()
+gotomanageuser()
+gotochose()
+gototrascrizione()
+gotomanageimage()
+initialize()
+setTable()
```

ViewImage

```
+ViewImage()
+setscene()
+LoadImage()
+Accept()
+Decline()
+setTable()
+initialize()
```

TrascrizioneController

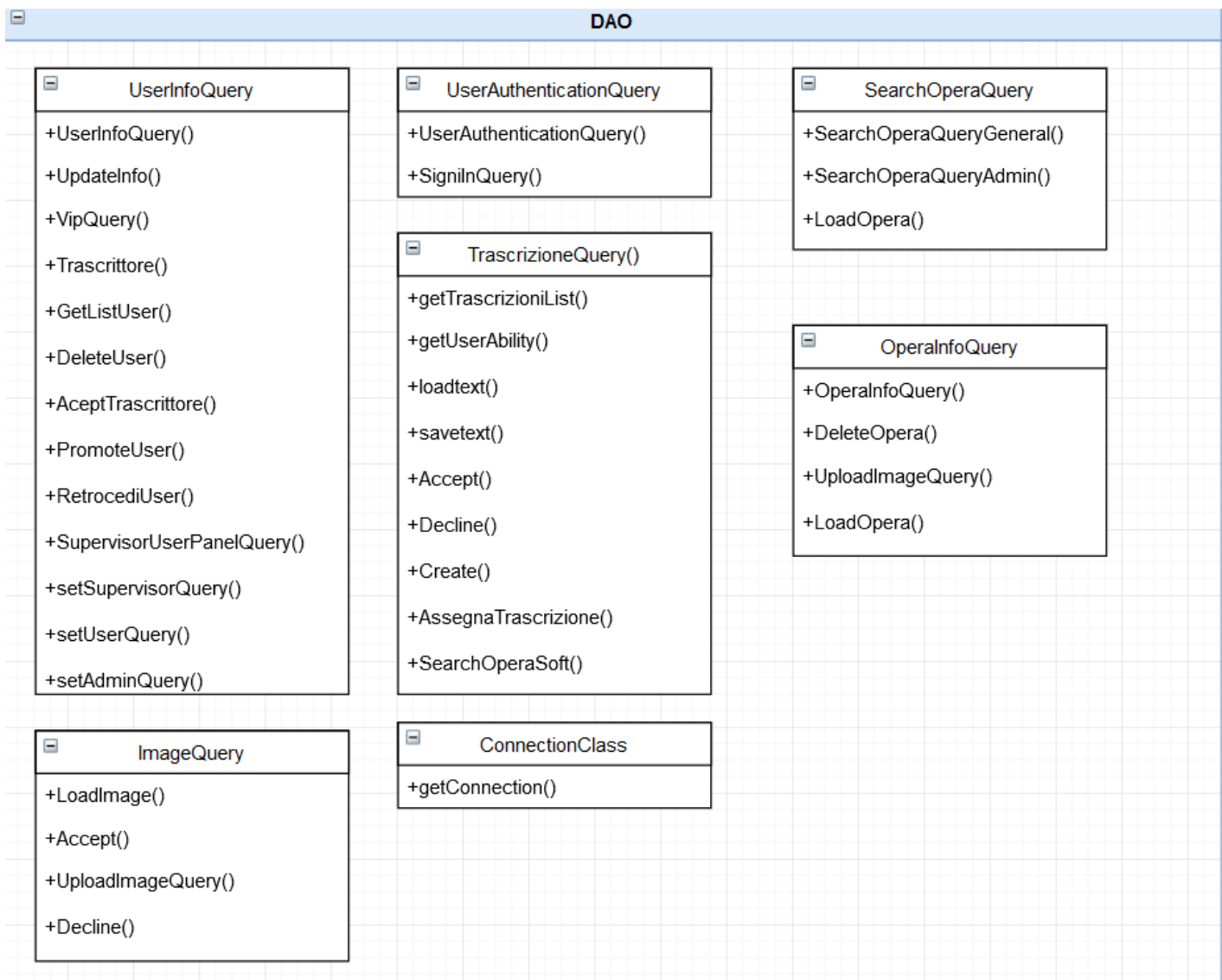
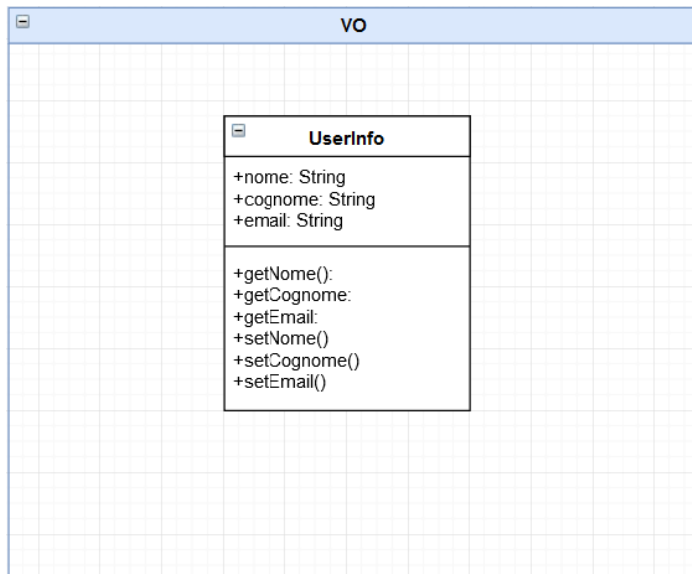
```
+TrascrizioneController()
+setScene()
+gotohome()
+getTitolo()
+initialize()
+setTable()
```

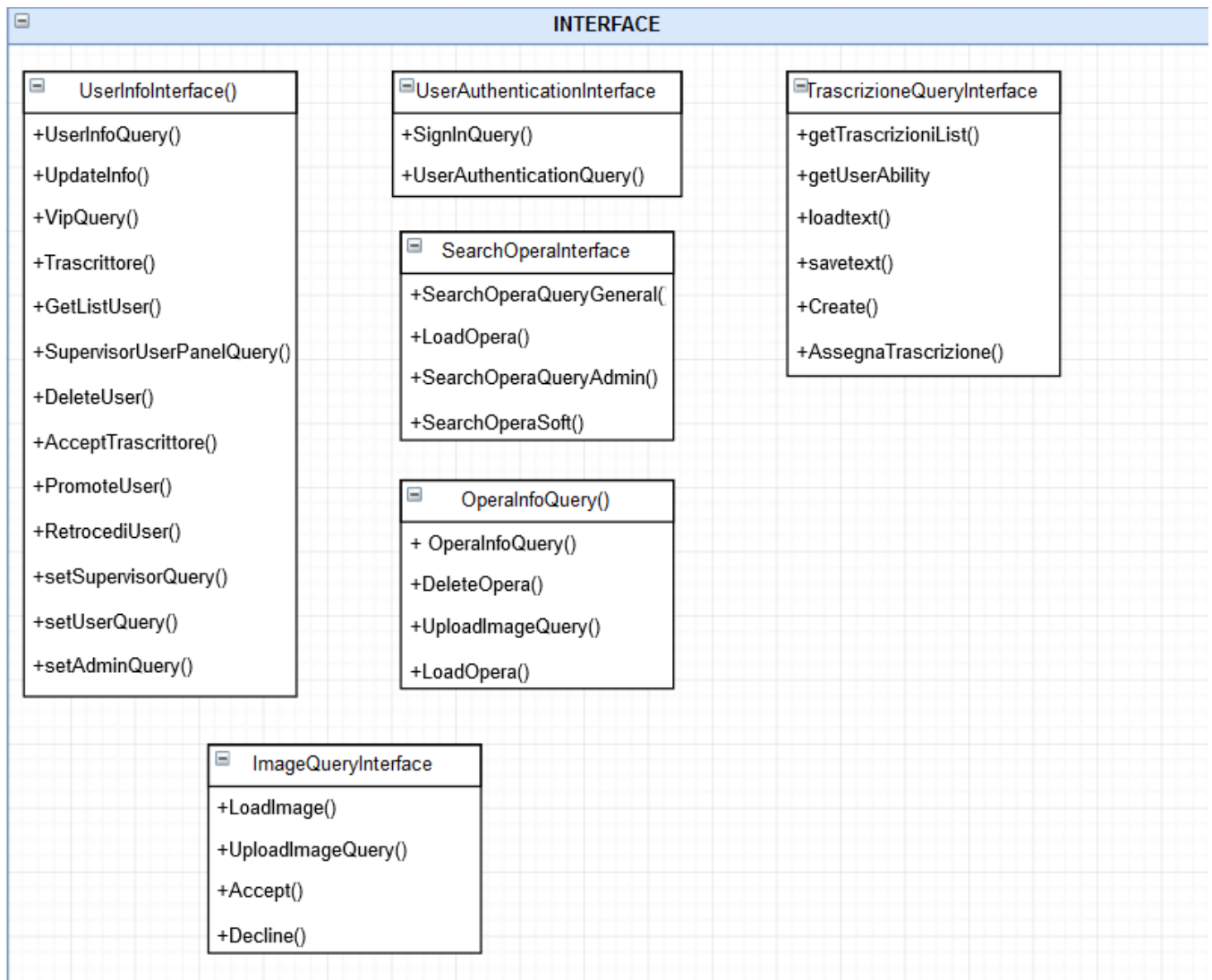
ViewProfileController

```
+ViewProfileController()
+ViewProfile()
+backhome()
+userInfo()
+updateInfo()
+Vip()
+Trascrittore()
+initialize()
```

AdminPannelController

```
+AdminPannelController()
+setsceneAdmin()
+gotoprofile()
+gotoHome()
+searchuser()
+searchOpera()
+search1()
+DeleteAccount()
+DeleteOpera()
+cbSupV()
+cbAdmin()
+initialize()
+setTable()
+setTable1()
```





3.2 Descrizione dei dettagli di design scelti

Model:

La classe **User** contiene, al suo interno, tutte le informazioni principali riguardanti l'utente e i suoi relativi privilegi nel sistema.

Nella classe **Opera** invece, abbiamo tutte le informazioni riguardanti l'opera come i metadati (titolo, autore, data pubblicazione, ecc..) di quella determinata opera caricata, le varie immagini dell'opera che sono state approvate e caricate nel sistema dai supervisor, che può essere una o più di una, in base all'opera e infine il testo trascritto dagli utenti autorizzati, se disponibile.

La classe **Immagine** conserva al suo interno gli oggetti relativi delle classi **ImmagineScan**, che rappresenta la fase della scansione dell'immagine e **ImmagineTrascritta**, che rappresenta la fase di trascrizione di quella determinata pagina o pagine.

Nella classe **ImmagineScan** abbiamo l'immagine caricata in attesa di essere approvata da un supervisor, e ciò viene definito grazie a dei flag che andremo ad implementare chiamati revisione,accettata.

Tramite questi flag andiamo a settare nel sistema se l'immagine è stata revisionata da parte di un supervisor, infatti abbiamo inserito anche il metodo per prelevare l'username del supervisor che ha lavorato alla revisione di tale immagine, e se è stata accettata o meno e quindi pronta per essere caricata nel sistema.

Accade più o meno la stessa cosa nella classe **ImmagineTrascritta**, dove oltre ad avere i soliti due flag accettata e revisione, abbiamo un altro valore booleano chiamato finish, utilizzato per sapere che l'immagine dell'opera in trascrizione è stata conclusa e che quindi i supervisor possono andare a revisionare tale opera e modificarla nel caso ci sia qualche errore, e successivamente decidere se accettarla o meno e infine pubblicarla se l'esito del valore booleano risulta positivo.

Per la trascrizione dell'immagine andremo ad implementare un editor di testo chiamato TEI.

View:

Nella **view** troviamo tutti gli oggetti relativi all'interfaccia del sistema come:

- LoginForm,
- SignupForm,
- TrascrizionePanel,ecc..

Utilizzati quindi per far interfacciare l'utente al sistema.

Li utilizziamo per definire le varie azioni che poi i button,panel,form,textfield ecc... avranno quando l'utente interagirà con tali funzionalità.

Per esempio in LoginForm, mettiamo a disposizione due campi textfield per poter inserire l'username e la password e un pulsante per poter effettuare l'accesso, che invia i dati inseriti confrontandoli con quelli presenti nel db.

Controller:

Le classi definite qui nel controller sono emerse dall'analisi del diagram entity-boundary-control nel punto 1.4.

Sono utilizzate per la gestione delle azioni svolte dall'utente come il login, la registrazione, il caricamento di un'opera, le trascrizioni di tale opere ecc..

Quindi in queste classi verranno definiti i metodi necessari per poter gestire le azioni da parte dell'utente utilizzatore del nostro sistema.

Dao:

Ogni classe definita nel dao contiene le query necessarie al controller per poter effettuare delle operazioni sul database.

Queste operazioni possono essere di vari tipi tra cui per esempio il controllo che avviene nella fase di login, dove i dati inseriti dall'utente passano per il controller, che attraverso le query fornite dal dao, andrà a fare un check con i dati presenti nel database, per poter far accedere l'utente alla homepage dell'applicazione.

Per l'inserimento di nuovi dati nel database, come può essere ad esempio per la pubblicazione di una nuova opera, di un nuovo utente, ecc...

Per la modifica di dati già presenti nel database che si vogliono aggiornare.

Quindi in definitiva nel dao saranno presenti i metodi che permetteranno al controller, tramite le query, di fare interrogazioni al database.