



# OBJECT ORIENTED SOFTWARE DESIGN

Anno 2018

Valerio Crescia Matricola 236107

[Valerio.crescia@student.univaq.it](mailto:Valerio.crescia@student.univaq.it)

## SPECIFICA PROGETTO

Il progetto si propone di creare un sistema che gestirà una biblioteca virtuale con lo scopo di far fruire ai propri utenti le opere contenute al suo interno, di immagazzinare e digitalizzare le opere dai responsabili del sistema.

## ANALISI REQUISITI

Per ogni requisito è stato assegnato un valore di importanza all'interno del sistema con una scala di valori da 1(Meno importante) a 5(principale)

Dividiamo i requisiti in **Non Funzionali** e **Funzionali**.

## REQUISITI FUNZIONALI

REQUISITO	PRIORITÀ
Il sistema deve poter immagazzinare le opere digitali	3

Il sistema deve essere dotato di una base di dati in modo da immagazzinare tutte le opere che gli operatori del sistema vorranno salvare.

REQUISITO	PRIORITÀ
Il sistema deve far visualizzare le opere digitali e le trascrizioni (se presenti) a utenti registrati	5

Dal requisito si evince che nel sistema deve essere presente una parte di login e registrazione e un'altra per la vista delle opere e delle trascrizioni.

REQUISITO	PRIORITÀ
Il sistema deve consentire la ricerca delle opere dai metadati e nel testo della trascrizione.	4

Dal requisito si nota che il sistema dovrà essere in grado di cercare nei metadati delle opere e nel documento della trascrizione.

Dal requisito non si evince dove la ricerca nel testo della trascrizione avvenga se correlata ad un'opera o in tutte le trascrizioni presenti del sistema.

REQUISITO	PRIORITÀ
La visualizzazione dell'opera singola dovrà avvenire con a destra la trascrizione (se disponibile) e sulla sinistra immagine della pagina dell'opera e sarà possibile sfogliare con un paginatore.	3

Dal requisito si evince come dovrà essere strutturata la vista della singola opera.

REQUISITO	PRIORITÀ
L'utente deve poter fare richiesta per diventare collaboratore del sistema (trascrittore).	3

Dal requisito si evince che nella parte visuale dell'utente che deve essere in precedenza registrato egli possa inviare una richiesta al sistema.

REQUISITO	PRIORITÀ
Il sistema deve far visualizzare i dati inseriti dall'utente nella autenticazione tra cui: titolo di studio, professione, indirizzo, email, etc... .	2

Dal requisito si evince che nella parte visuale del sistema dovrà essere presente una schermata dove l'utente potrà visualizzare le sue credenziali tra cui titolo di studio, professione, indirizzo, email le quali saranno salvate anche nella base di dati.

REQUISITO	PRIORITÀ
Il sistema deve poter far scaricare l'opera solo da utenti specifici.	3

Dal requisito si evince che il sistema dovrà essere in grado di restituire in un formato standard sia le immagini che la trascrizione.

**Requisito è incompleto** assegno che tutti gli utenti che collaborano al sistema possano scaricare l'opera; il formato restituito dall'sistema dovrà essere PDF sia per immagini che per la trascrizione.

REQUISITO	PRIORITÀ
Il sistema deve supportare il caricamento delle opere cioè delle immagini che rappresentano le pagine e dei metadati riguardanti l'opera.	5

Dal requisito si evince che il sistema dovrà essere in grado di immagazzinare le immagini e i metadati inseriti dall'utente.

**Requisito incompleto:** Non è presente chi può immettere le opere nel sistema; Assegno che tutti gli utenti registrati possono caricare le opere nel sistema.

REQUISITO	PRIORITÀ
Il sistema deve far controllare l'opera che si vuole far caricare facendo vedere le miniature delle pagine dell'opera caricata e le singole pagine sfogliabili con un paginatore.	4

Dal requisito si evince che nelle parte visuale dell'sistema dovrà essere presente una parte che sarà adibita alla visualizzazione dell'opera che si intende caricare

REQUISITO	PRIORITÀ
Il sistema deve far controllare il caricamento dell'opera dai supervisori all'acquisizione.	5

Dal requisito si evince che ci sarà un attore del sistema delegato alla validazione delle opere che il sistema vuole acquisire.

REQUISITO	PRIORITÀ
Il sistema deve far digitare e inserire le trascrizioni attraverso un text editor TEI integrato.	4

Dal requisito si evince che nel sistema deve essere presente una sezione dove l'utente può digitare la sua trascrizione e salvarla. Dovrà essere presente anche un'altra che dovrà elaborare le parole e le frasi che l'utente inserisce.

REQUISITO	PRIORITÀ
Il sistema deve far lavorare più trascrittori in contemporanea nella stessa pagina da trascrivere.	4

Dalla trascrizione si evince che il sistema dovrà far lavorare più utenti sullo stesso testo.

**Requisito incompleto:** Il requisito non spiega come gestire l'accesso al documento quindi si dovrà assegnare una politica in base alle librerie TEI trovate in rete.

REQUISITO	PRIORITÀ
Il sistema deve far revisionare le trascrizioni dai revisori delle trascrizioni.	4

Dalla specifica si evince che i revisori possono validare la trascrizione.

**Requisito ambiguo:** Alla operazione "revisione" assegno anche l'operazione oltre alla visione del documento anche l'operazione di modifica della trascrizione.

REQUISITO	PRIORITÀ
Il sistema deve essere in grado di gestire le assegnazioni delle trascrizioni a i trascrittori(cardinalità n m) ed anche gestire i livelli dei trascrittori.	5

Dal requisito si evince che una parte visuale del sistema dovrà far assegnare i trascrittori alle trascrizioni delle singole immagini del opere che rappresentano le singole pagini dell'opera

REQUISITO	PRIORITÀ
Il sistema deve consentire la gestione delle trascrizioni concluse(correzione,validazione)	4

Dal requisito si evince che il sistema deve essere in grado di gestire le trascrizioni che sono già state pubblicate.

REQUISITO	PRIORITÀ
Il sistema deve consentire la pubblicazione delle immagini delle opere	4

Quindi deve essere presente una gestione delle pubblicazioni immagini

REQUISITO	PRIORITÀ
E' possibile anche riassegnare delle pagine ai trascrittori.	4

REQUISITO	PRIORITÀ
Il sistema deve consentire la pubblicazione delle trascrizioni.	4

Quindi deve essere presente una gestione delle pubblicazioni delle trascrizioni

REQUISITO	PRIORITÀ
Il sistema deve assegnare revisori dell'acquisizione alla revisione delle immagini delle opere caricate.	5

Quindi una gestione del personale per assegnare i revisori delle opere a per la validazione delle immagini.

REQUISITO	PRIORITÀ
Gestione in back-end di tutto il sistema: anagrafica utenti, opere, etc.	5

Quindi gestione completa dell'utenza.

## REQUISTI NONFUNZIONALI

**Usability:** Il sistema dovrà essere di facile utilizzo

**Reliability:** Il portala dovrà garantire un'esperienza all'utente di tutte le funzionalità messe a disposizione, senza errori.

**Availability:** Il portale dovrà essere sempre disponibile e deve poter garantire in qualsiasi momento tutte le funzioni desiderate.

## ATTORI DEL SISTEMA

Dall'analisi dei requisiti e dalla rilettura della specifica gli attori che interagiscono nel sistema sono:

**Utente anonimo:** l'utente anonimo è colui che può solo registrarsi ed effettuare il login.

**Utente Registrato:** può consultare le opere immagini e trascrizioni ed caricare le opere nel sistema e fare richiesta per diventare un trascrittore.

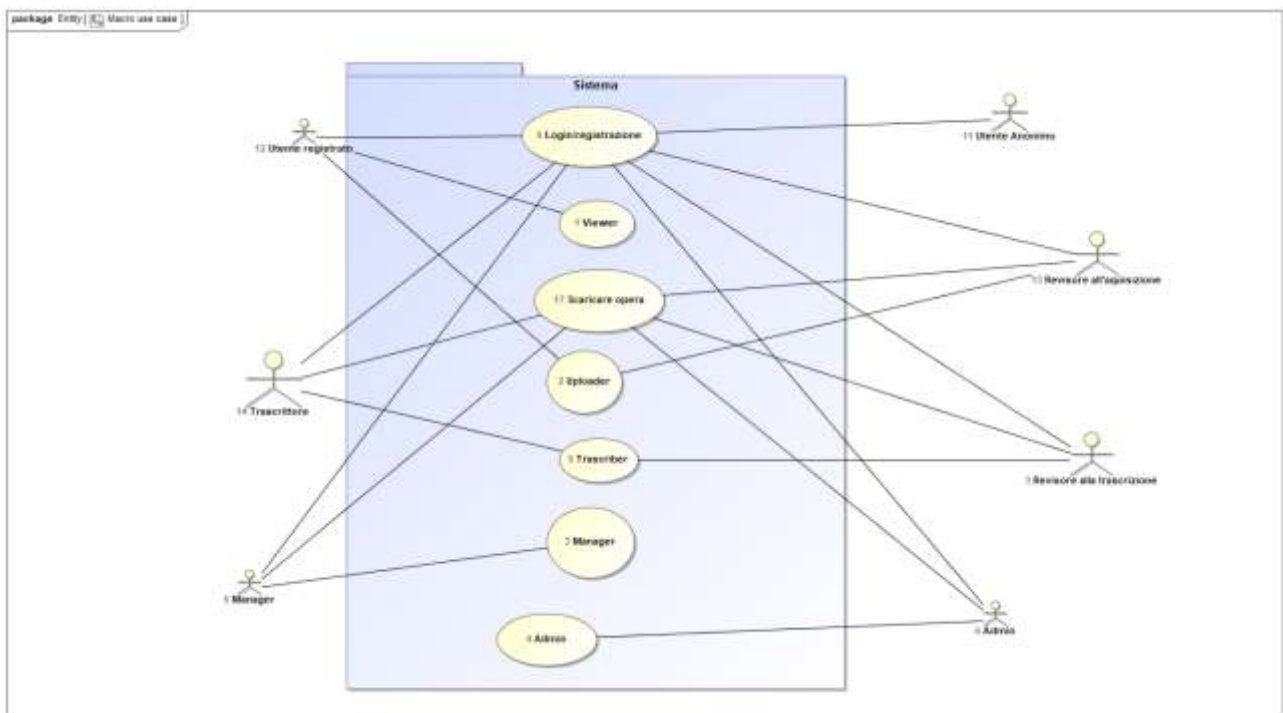
**Trascrittore:** può trascrivere le pagine a lui assegnate ed effettuare il login.

**Revisore alle trascrizioni:** può validare e correggere le opere a lui assegnate ed effettuare il login.

**Revisore alle acquisizioni:** può validare le acquisizioni a lui assegnate ed effettuare un login.

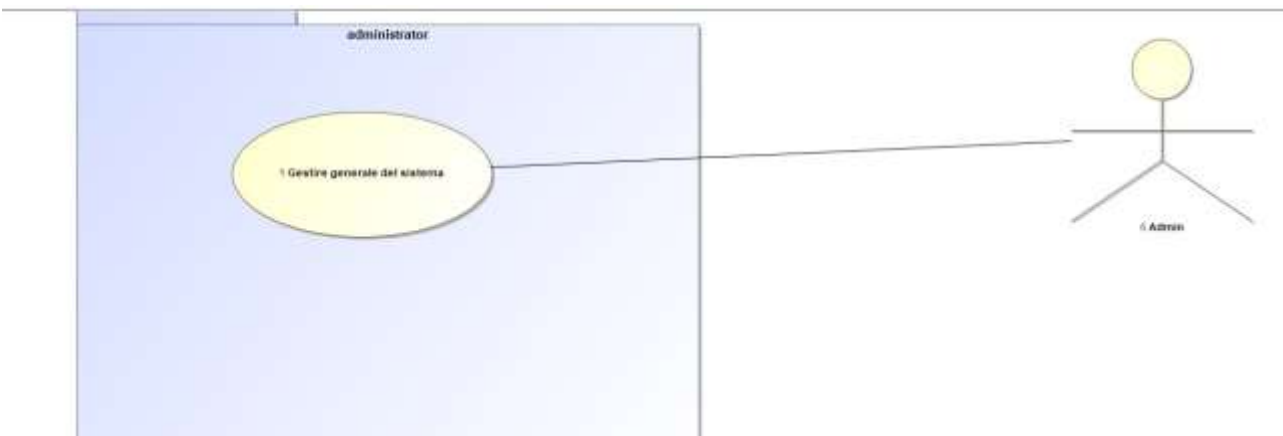
**Manager:** può assegnare più trascrizioni ai trascrittori, revisori alle trascrizioni alle opere da validare anche se pubblicate e assegnare revisori alle acquisizioni alle opere non ancora pubblicate, pubblica le opere le trascrizioni può gestire il livello di tutti i trascrittori.

## USE CASE DIAGRAM



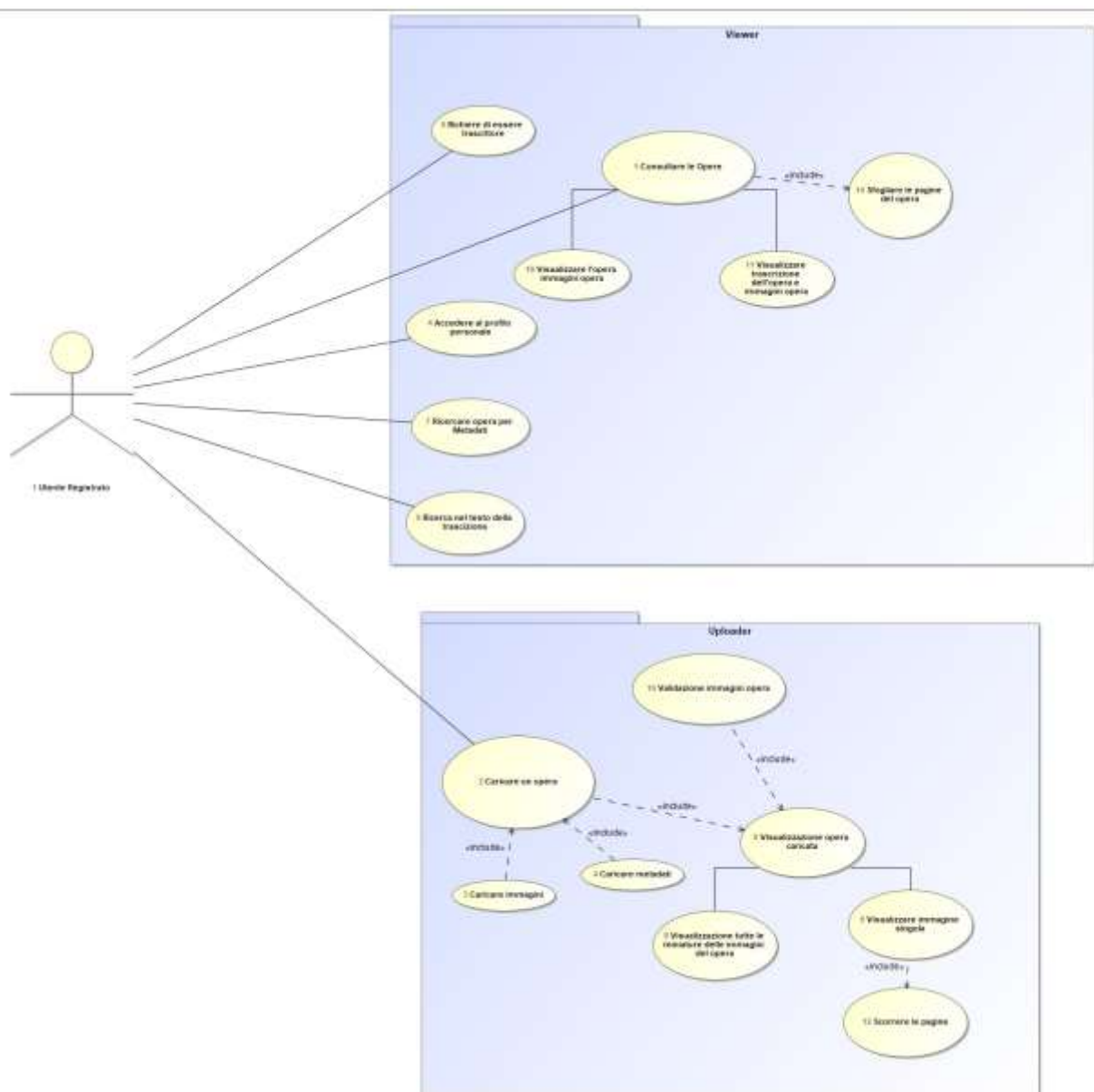
Gli use case specifici per attore hanno all'loro intero tutti gli use case che posso eseguire tranne il login/registrazione e scaricare opera che sono espressi già nel macro use case

**Admin:**

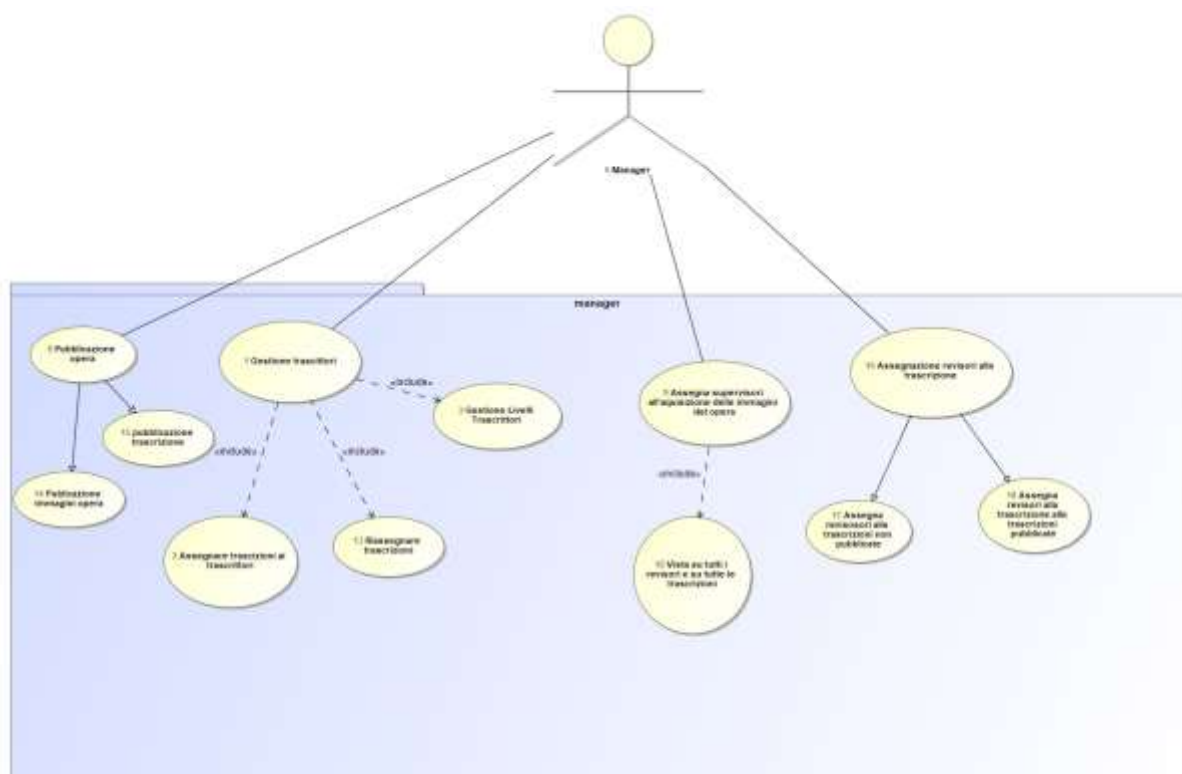




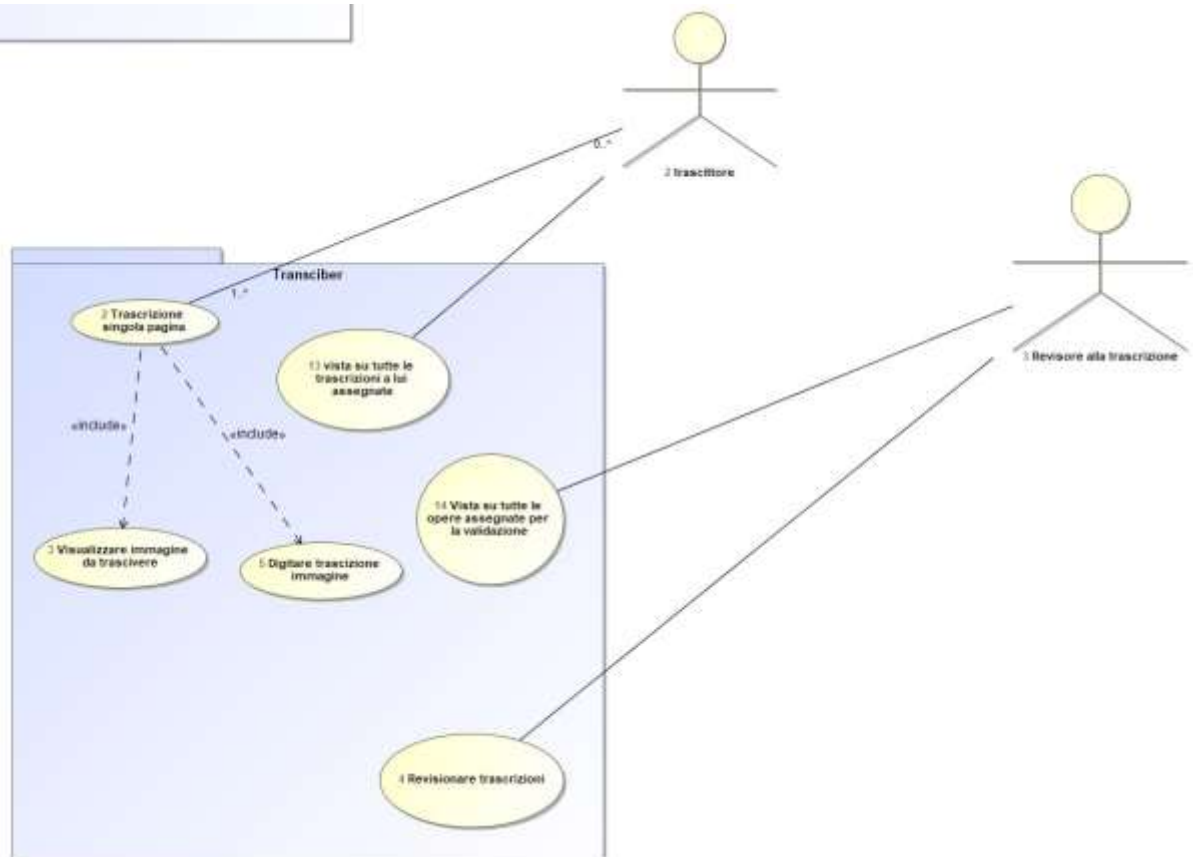
Utente registrato:



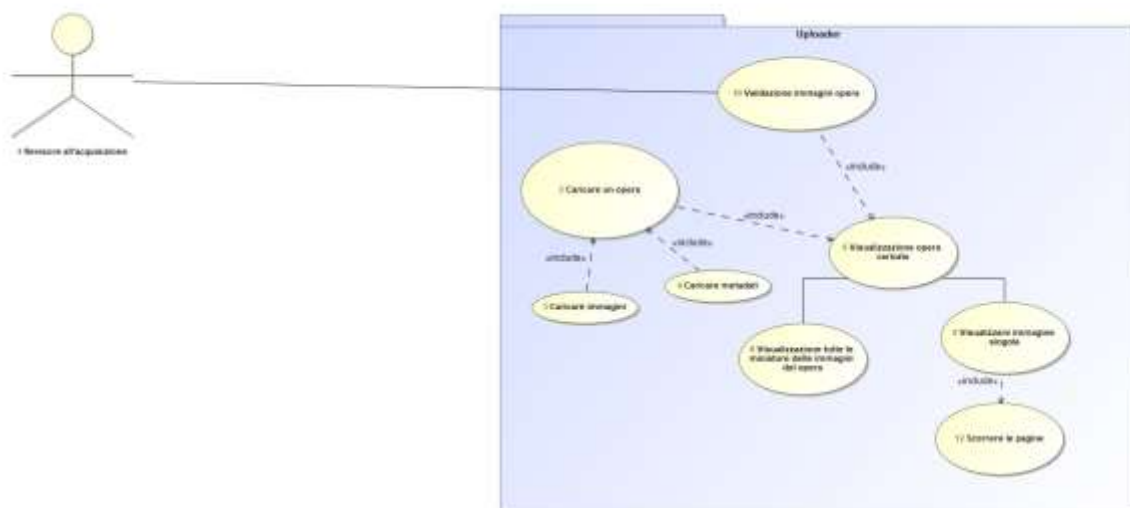
## MANAGER:



## TRACRITTORE E REVISORE ALLA TRASRIZIONE:



## REVISORE ALL'ACQUISIZIONE:



# ENTITY CLASS

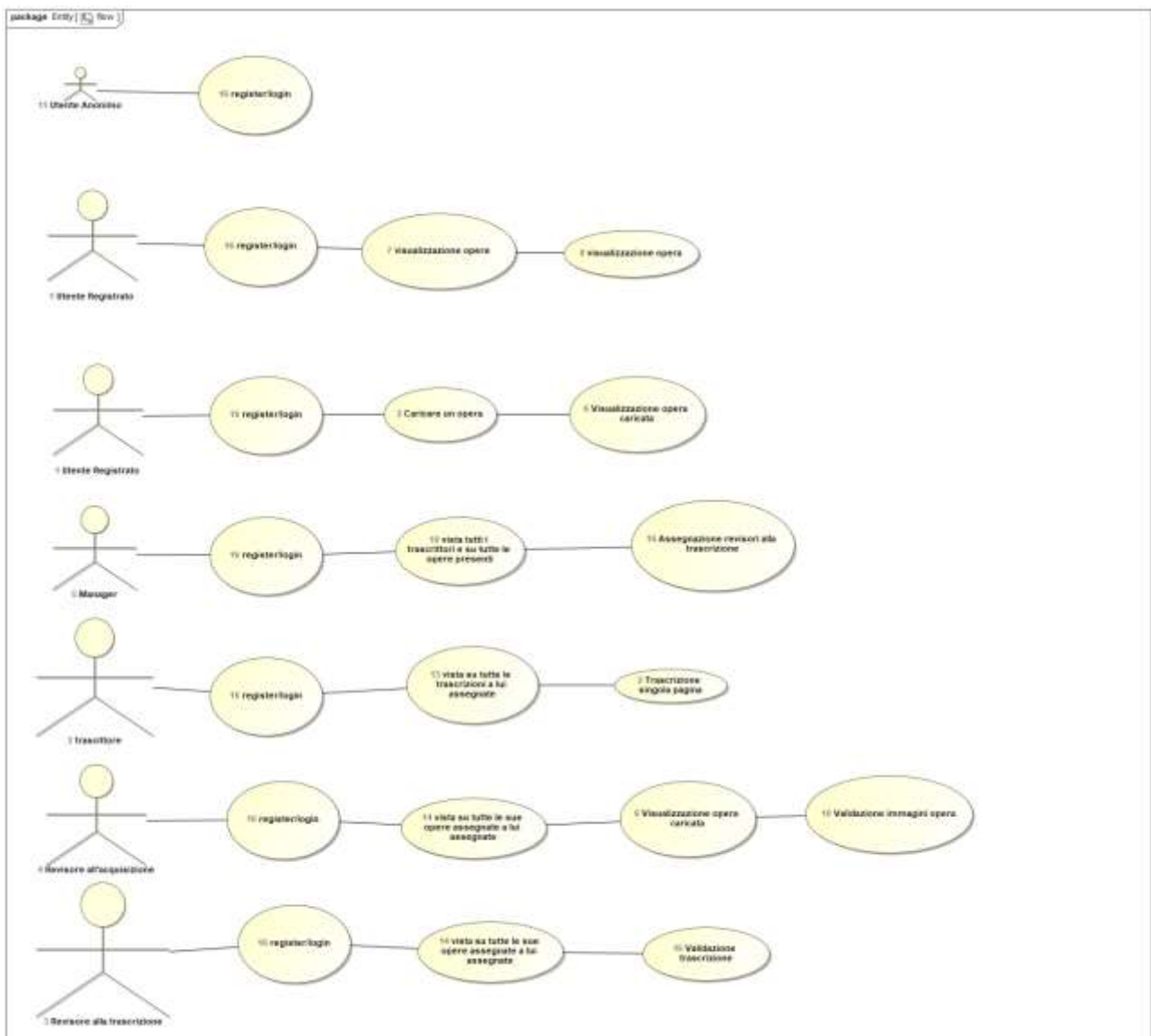
Dallo studio dei requisiti si nota che le entity principali del sistema sono **opera**, **immagini**, **trascrizione** riguardante la singola pagina, **utente**.

Siccome dobbiamo dividere i vari utenti in sotto insiemi che possono fare determinate mansioni allora aggiungo una nuova entity che classificherà gli utenti detta **gruppo**.

La descrizione più approfondita avverrà nel object diagramm.

## SCENARI

Andremo ad analizzare le sequenze di use case delle principali operazioni che gli utenti possono eseguire:



## ANALISI USE CASE PRINCIPALI

Use Case	LOGIN/REGISTRAZIONE
Attori partecipanti	Tutti gli utenti
Descrizione	Fa accedere al sistema tutti gli utenti
Evento scatenante	Click sul pulsante dopo inserimento dei dati
Uses	Login/registrazione effettuati con successo o errore

Use Case	Trascrizione singola pagina
Attori partecipanti	Trascrittore
Descrizione	Permette al trascrittore di salvare il suo testo della trascrizione
Evento scatenante	Click sul pulsante dopo inserimento della trascrizione
Uses	Salvataggio della trascrizione o errore

Use Case	Validazione immagini opera
Attori partecipanti	Revisore all'acquisizione
Descrizione	Permette al revisore di validare le trascrizioni
Evento scatenante	Click sul pulsante della validazione
Uses	Validazione opera o errore

Use Case	Validazione trascrizione
Attori partecipanti	Revisore della trascrizione
Descrizione	Permette di validare le trascrizioni delle opere
Evento scatenante	Click sul pulsante di validazione
Uses	Validazione avvenuta o errore

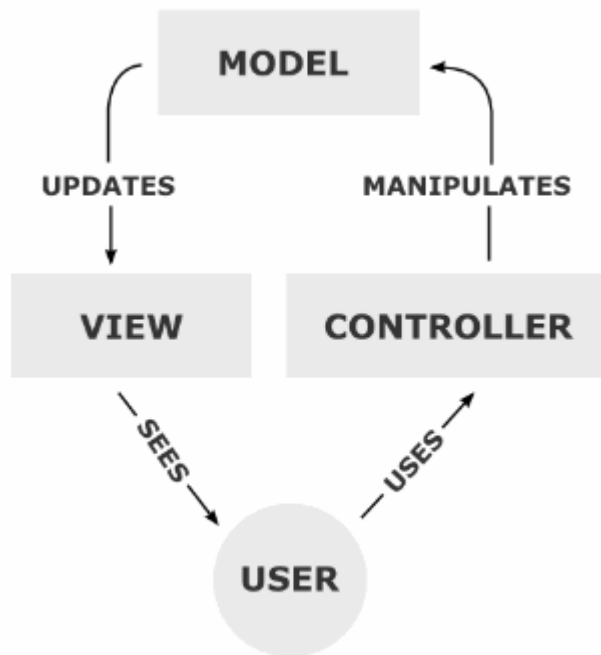
<b>Use Case</b>	<b>Assegna revisori alla trascrizione</b>
Attori partecipanti	manager
Descrizione	Permette al manager di assegnare ai revisori le opere per la validazione
Evento scatenante	Click sul pulsante dopo aver scelto i revisori e le opere
Uses	Assegnazione avvenuta con successo o errore

## DESIGN/ ARCHITECTURAL PATTERN

**Model-View-Controller** (MVC) è un pattern utilizzato in programmazione per dividere il codice in blocchi dalle funzionalità ben distinte.

Il pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- Il **model** fornisce i metodi per accedere ai dati utili all'applicazione;
- La **view** visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti.
- Il **controller** riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.



Il **DAO** (*Data Access Object*) è un pattern architetturale per la gestione della persistenza. Invece di effettuare le interrogazioni dal component Model, utilizzo questo pattern per delegare il compito di aggiornamento di aggiunta e di modifica dei dati del Database al DAO. Il pattern si compone di tre parti:

**Data Access Object Interface:** Un'interfaccia che definisce le operazioni standard da eseguire sui modelli dei nostri oggetti.

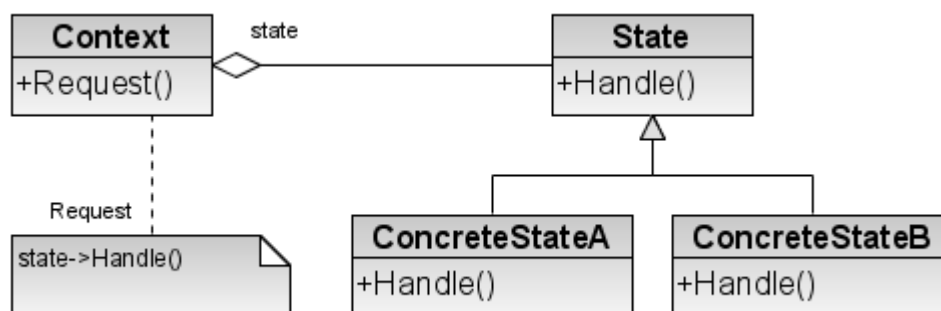
**Data Access Object classe concreta:** Una classe che implementa l'interfaccia definita al punto precedente, i cui compiti sono quelli di prendere i dati dalla sorgente (nel nostro caso un database).

**Model Object:** Un semplice oggetto contenente i metodi *getter* e *setter* per leggere/memorizzare dati ottenuti attraverso le classi DAO.

Dalla lettura delle specifiche e dalla vista dello schema della pubblicazione si è pensato di usare il design pattern **STATE** per la gestione delle opere.

Esso consente ad un oggetto di cambiare il proprio comportamento a run-time in funzione dello stato in cui si trova. I le parti fondamentali di questo design sono:

- **Context:** Definisce l'interfaccia del client e mantiene un riferimento ad un ConcreteState.
- **State:** Definisce l'interfaccia, implementata dai ConcreteState, che incapsula la logica del comportamento associato ad un determinato stato.
- **ConcreteState:** Implementa il comportamento associato ad un particolare stato.



Gli stati sono ricavati dagli posso essere ricavati dello scema preso delle specifiche:



# ARCHITETTURA DEL SISTEMA

Il dalle specifiche si è optato per un sistema desktop ed essendo un sistema client-server l'architettura sarà di tipo MVC.

La nostra architettura è stata divisa in sei componenti quali.

**USER INTERFACE:** Rappresenta una parte del sistema che interagisce direttamente con l'utente. Questa parte corrisponde ad un template generato con il framework SWING.

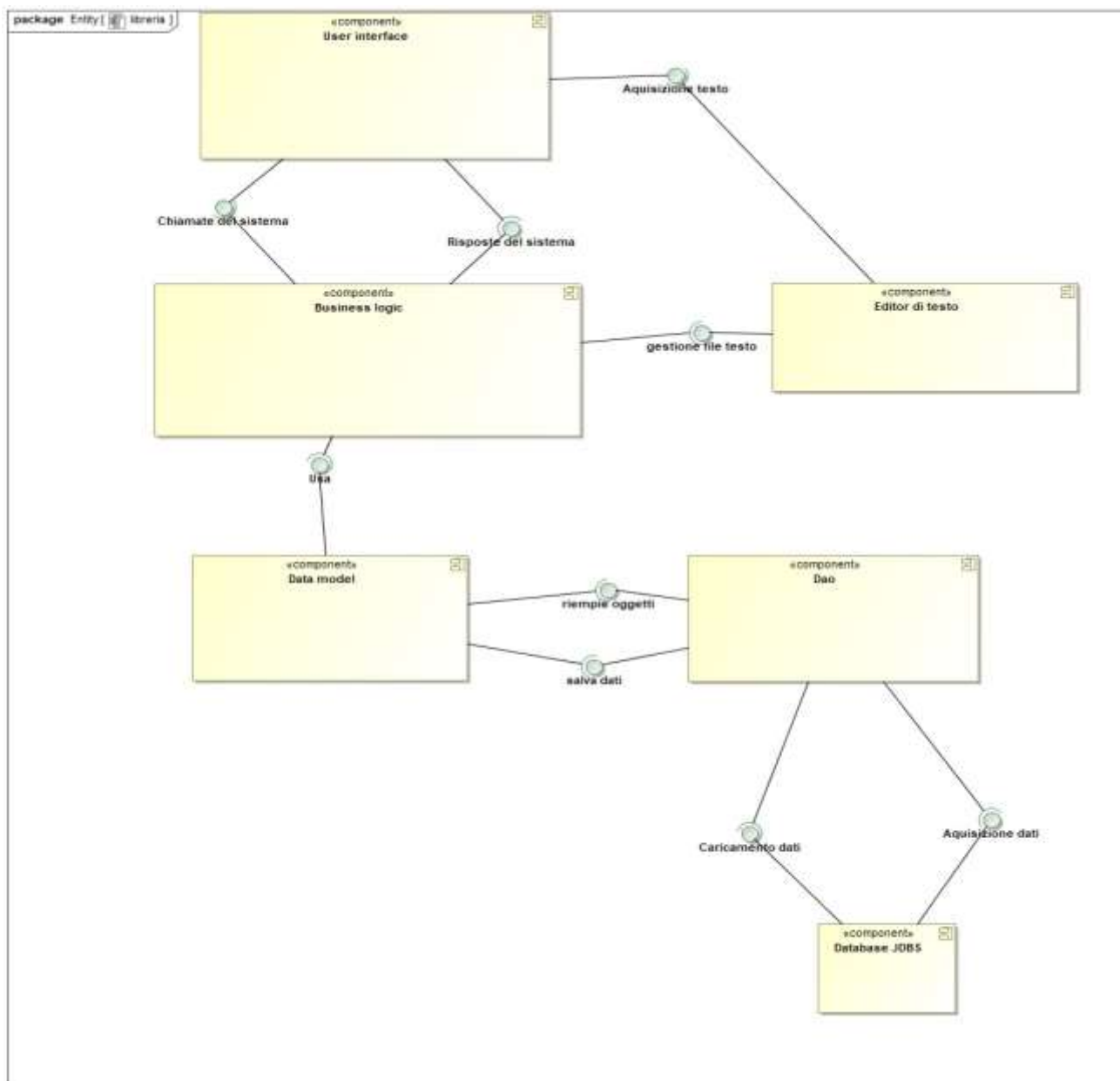
**BUSINESS LOGIC:** E' un componente che racchiude tutte le classi che rappresentano le funzioni del nostro sistema(Controller).

**DATA MODEL:** E' il componente che racchiude al suo interno le classi che rappresentano gli oggetti del sistema che i quali saranno modellati dalle classi della business logic e dalle classi del dao.

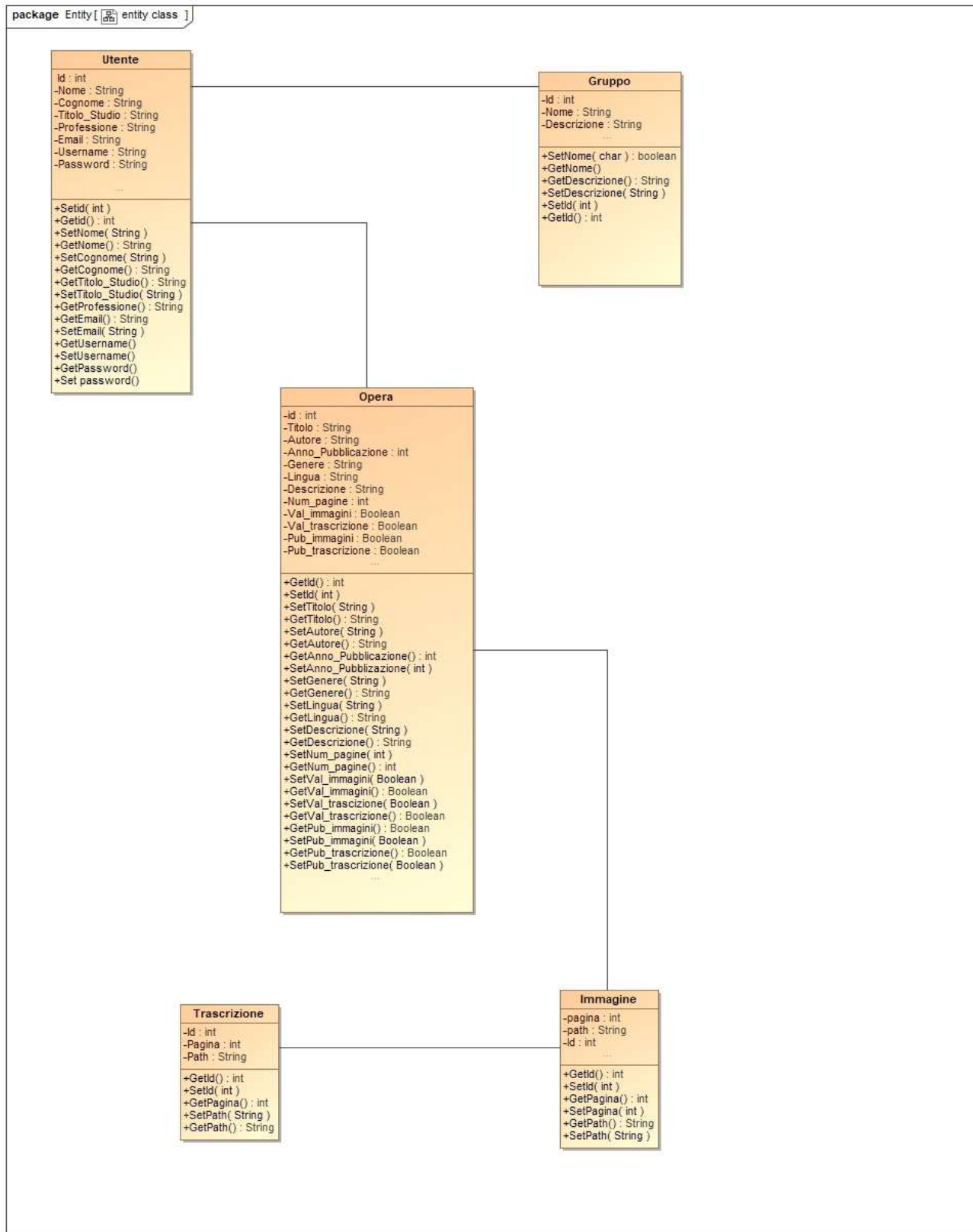
**DAO:** è il componente che si interpone tra il database e il sistema ed ha il compito di estrapolare e immettere dati sul nostro database.DATABASE

**JDBS :** componente che si occupa dello stoccaggio delle informazioni del sistema.

**EDITOR DI TESTO:** è una parte del sistema che è incaricata della gestione della scrittura sul file di testo della trascrizione.



# OBJECT DIAGRAMM



## DESCRIZIONE OBJECT DIAGRAMM

**Opera:** rappresenta l'opera, essa ha come identificata dall'attributo ID ed e caratterizzata dai seguenti attributi: Titolo, Autore, Anno\_publicazione, Genere, Lingua, Descrizione. L'attributo n\_pagine è molto importante per la cardinalità delle relazioni con le immagini e le trascrizioni.

Gli altri attributi quali Val\_immagini, Val\_trascrizione, Pub\_immagini, Pub\_trascrizione sono booleani che rappresentano lo stato dell'opera. I metodi servono per settare i dati e recuperare i dati.

**Image:** rappresenta le pagine dell'opera con l'immagine, gli attributi id lo indentifica, path e pagina lo caratterizzano. I metodi dell'oggetto fanno riferimento a i get/set delle variabili variabile immagine.

**Trascrizione:** riguarda la trascrizione della singola pagina è identificata tramite un ID il path identifica la posizione del file della trascrizione come nella entity I metodi dell'oggetto fanno riferimento al get/set del testo della trascrizione.

**Utente:** Caratterizzato dell' ID rappresenta gli attori che usano il sistema e le loro credenziali.

**Gruppo:** Caratterizzato dell' ID rappresenta gli insiemi di utenti che possono svolgere una determinata azione.