



**Università degli Studi
di L'Aquila**

OBJECT- ORIENTED SOFTWARE

Progetto Biblioteca Digitale

Università degli studi dell'Aquila

Membri del team:

- Andrea Paris

Sommario

Membri del team:	1
Documento dei requisiti:	3
Requisiti non funzionali:	5
Attori:.....	5
1.2 Use Case	7
Scenari:	15
1.3 Modelli di dominio:	18
1.4 Analisi:	19
2.1 Modello dell'architettura del sistema:	29
2.2 Descrizione dell'architettura:	30

2.3.1 Descrizione delle scelte:	31
2.3.2 Design Patterns:	32
Class diagram – GUI:	35
Class diagram – View:	36
Class diagram – Controller:	37
Class diagram – Model:	38
Class diagram – Dao:	40
Class diagram – Database:	42
Class Diagram – Completo:	44
2.3.3 DataBase:	45

Requirements

1.1 Documento dei requisiti

I requisiti del sistema sono i seguenti; a ciascuno di essi è associata una priorità su una scala da 1 a 5 (1: priorità minima; 5 priorità massima).

-Registrazione (5): funzionalità che permettere a ciascun utente di registrarsi al portale, ovvero permette a ciascun utente di crearsi il proprio profilo.

-Login (5): funzionalità che permette a ciascun utente di accedere al proprio profilo

-Consultazione opere digitali (5): funzionalità che permette a ciascun utente di cercare e di visualizzare ciascuna opera presente nella biblioteca

-Download delle opere (3): funzionalità che permette ad utenti con determinati privilegi di poter effettuare il download delle opere scelte

-Digitalizzazione:

- **Visualizzazione pagine in miniature (5):** funzionalità che permette di visualizzare tutte le pagine che compongono un'opera sotto forma di miniature
- **Visualizzazione pagine (5):** funzionalità che permette di visualizzare una per volta, scorrendo con un puntatore, tutte le pagine che compongono un'opera

-Trascrizione (4): funzionalità che permette a ciascuna opera di poter essere “trasformata” in un testo digitale attraverso un text editor TEI (Text Encoding Initiative) integrato

-Assegnazione (5): funzionalità che permette di assegnare parte di un'opera (1 o più immagini) a uno o più utenti trascrittori

-Revisione (4): funzionalità che permette di revisionare ciascuna trascrizione

-Correzioni (4): funzionalità che permette di correggere ciascuna trascrizione

-Validazione (4): funzionalità che permette di validare ciascuna trascrizione

-Riassegnazione (5): funzionalità che permette di riassegnare le pagine di un'opera ad un altro utente trascrittore

- Pubblicazione** (4): funzionalità che permette di pubblicare ciascuna opera e ciascuna trascrizione
- Gestione livelli** (3): funzionalità che permette di gestire il livello di ciascun utente trascrittore
- Supervisione** (4): funzionalità che permette di supervisionare e controllare ciascuna acquisizione delle immagini di ciascuna opera

Requisiti non funzionali

- **Usability**: Il sistema dovrà essere pulito e di facile utilizzo.
- **Reliability**: Il portale dovrà garantire all'utente le funzioni messe a disposizione (vedi Use Case Diagram) senza errori.
- **Availability**: Il portale dovrà essere sempre disponibile e deve poter garantire in qualsiasi momento tutte le funzioni desiderate.

Attori

Il primo attore identificato nel nostro sistema è l'**Amministratore** il cui compito principale è la gestione di tutto il nostro sistema. In particolare i suoi compiti possono essere quelli di controllo sull'efficienza del sistema, il quale deve essere il più pulito possibile e di facile utilizzo: ha il pieno controllo del sistema.

L'amministratore occupandosi della gestione del back-end non interagisce direttamente con l'interfaccia della Biblioteca Digitale ma egli ha il privilegio di poter interagire direttamente con il database del sistema attraverso istruzioni (es. comandi sql) che permettono di avere la completa gestione dei dati relativi alla piattaforma. Inoltre l'amministratore può anche gestire gli utenti registrati sulla piattaforma promuovendoli, dopo apposita richiesta, a trascrittori.

Un altro attore che viene identificato è il **Manager** il quale ha i seguenti compiti:

- 1) Gestione delle assegnazioni ovvero può assegnare una parte (pagina/immagine) di un'opera (insieme di pagine/immagini) a uno o più trascrittori
- 2) Gestione delle riassegnazioni ovvero ha il privilegio di poter assegnare una parte di un'opera ad un trascrittore diverso rispetto a quello scelto inizialmente

3) Gestione delle trascrizioni ovvero ha il privilegio di revisionare, correggere e validare ciascuna trascrizione presente nella Biblioteca. Per trascrizione si intende ciascuna parte di un'opera che è stata convertita in formato testo digitale

Il terzo attore che viene identificato è il **Trascrittore**. Tale attore ha il compito di redigere una trascrizione relativa ad una pagina di un'opera. Con le opportune sincronizzazioni più trascrittori possono lavorare sulla stessa pagina.

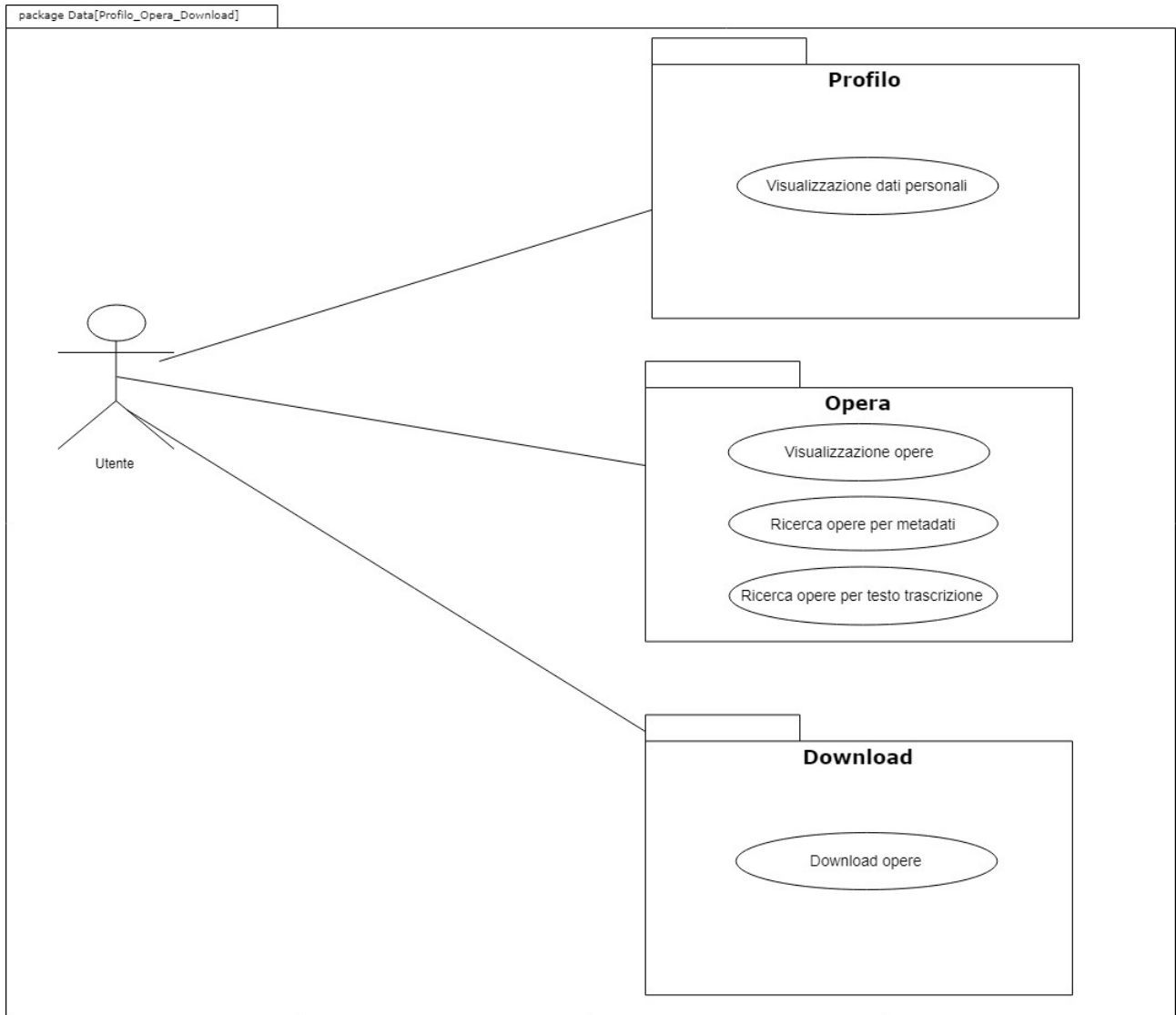
Il quarto attore presente nel nostro sistema è l'**Uploader** il quale ha il compito di caricare le immagini relative a ciascuna opera all'interno della Biblioteca digitale.

Il quinto ed ultimo attore che viene identificato è l'**Utente** ovvero colui che consulta la Biblioteca Digitale. Tale attore è estraneo all'architettura della piattaforma infatti può solamente interagire con l'interfaccia (front-end) del sistema. Un utente che per poter consultare la Biblioteca Digitale deve eseguire la fase di registrazione e successivamente di login. Un utente registrato può: consultare e ricercare tutte le opere presenti nella biblioteca. Tale attore attraverso un'opportuna richiesta può anche diventare

trascrittore del sistema. Inoltre può creare ed interagire con i dati del proprio profilo.

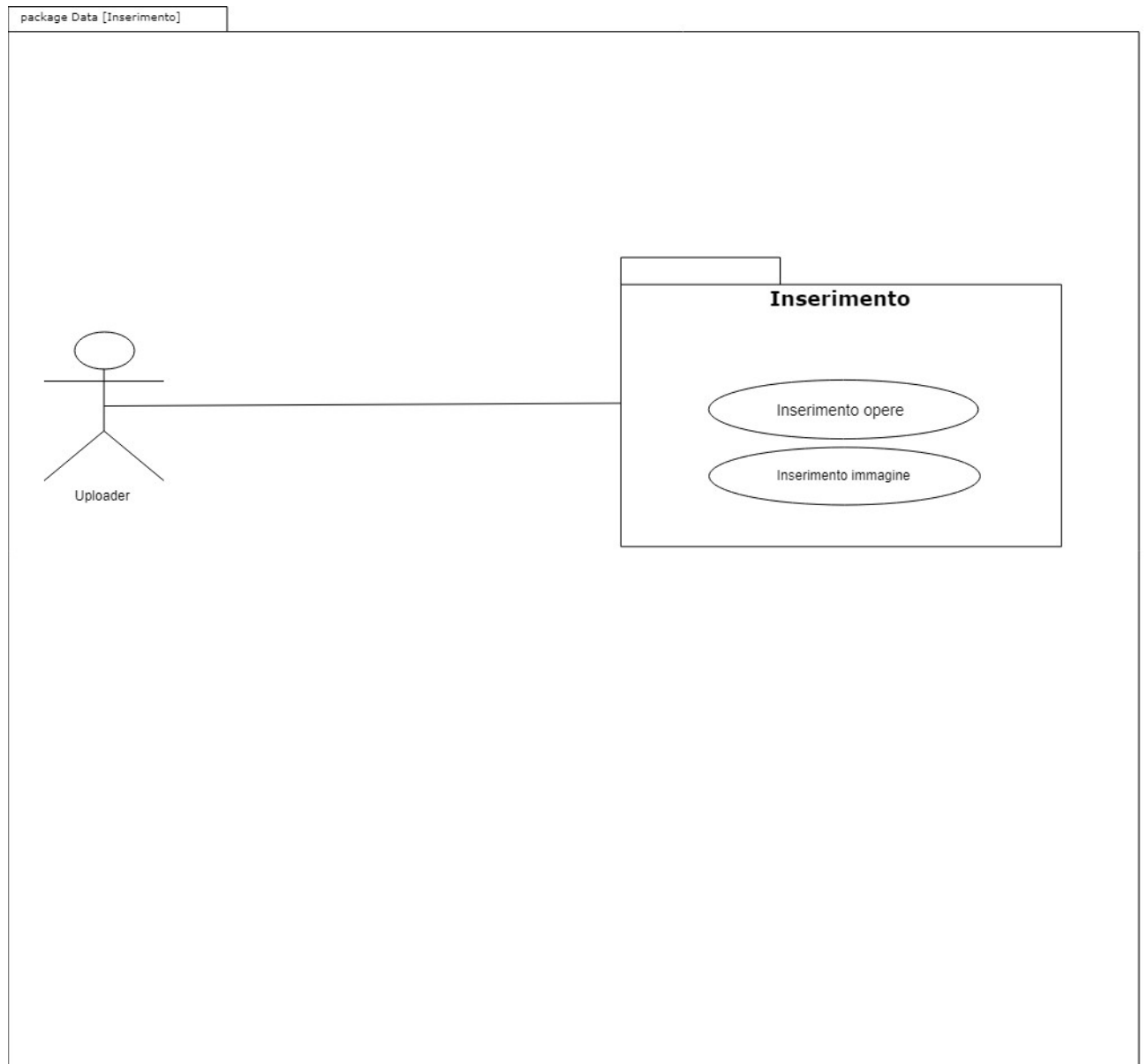
1.2 Use Case

USE CASE UTENTE



Use Case 1 - Utente

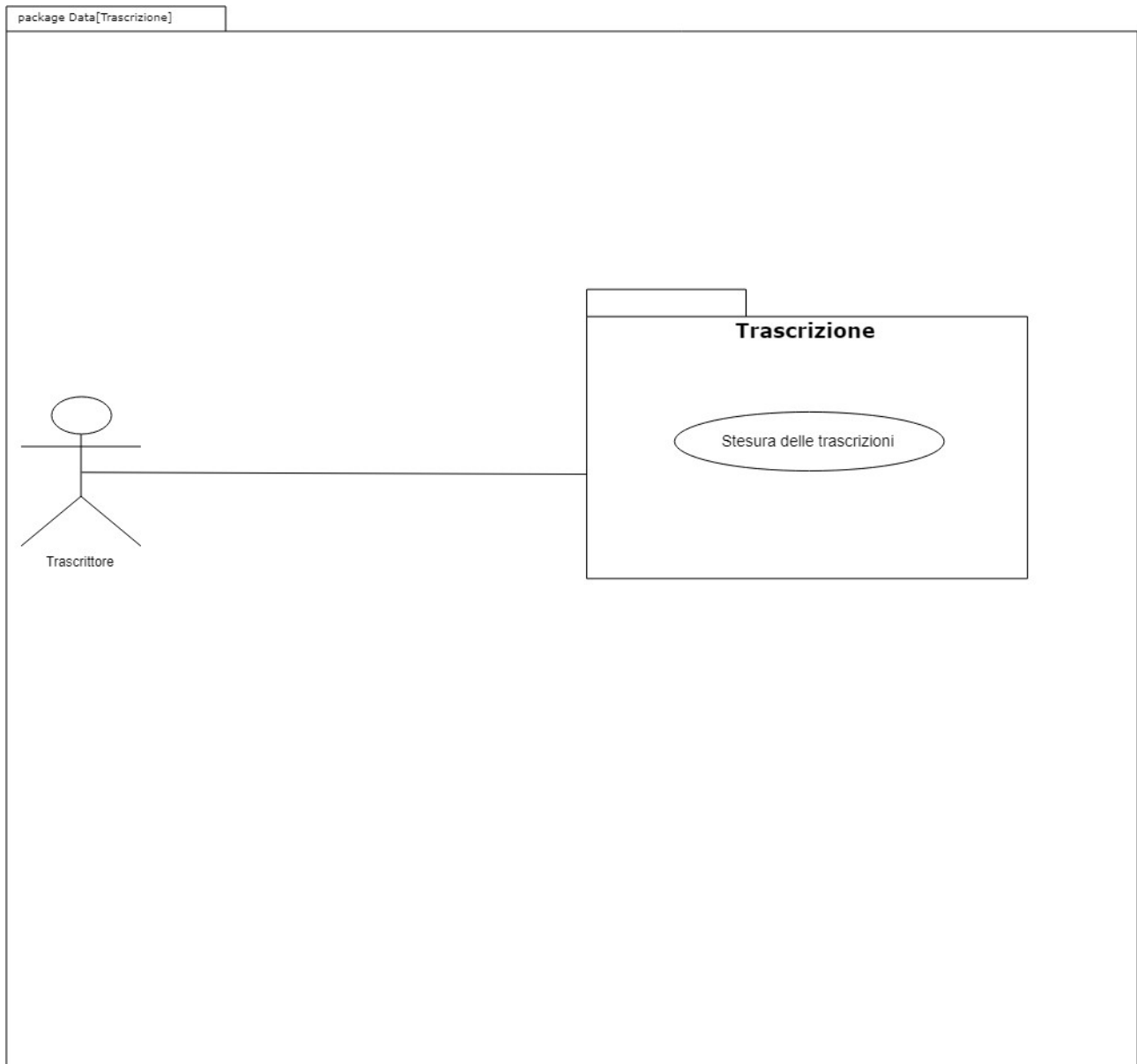
USE CASE UPLOADER



Uploader

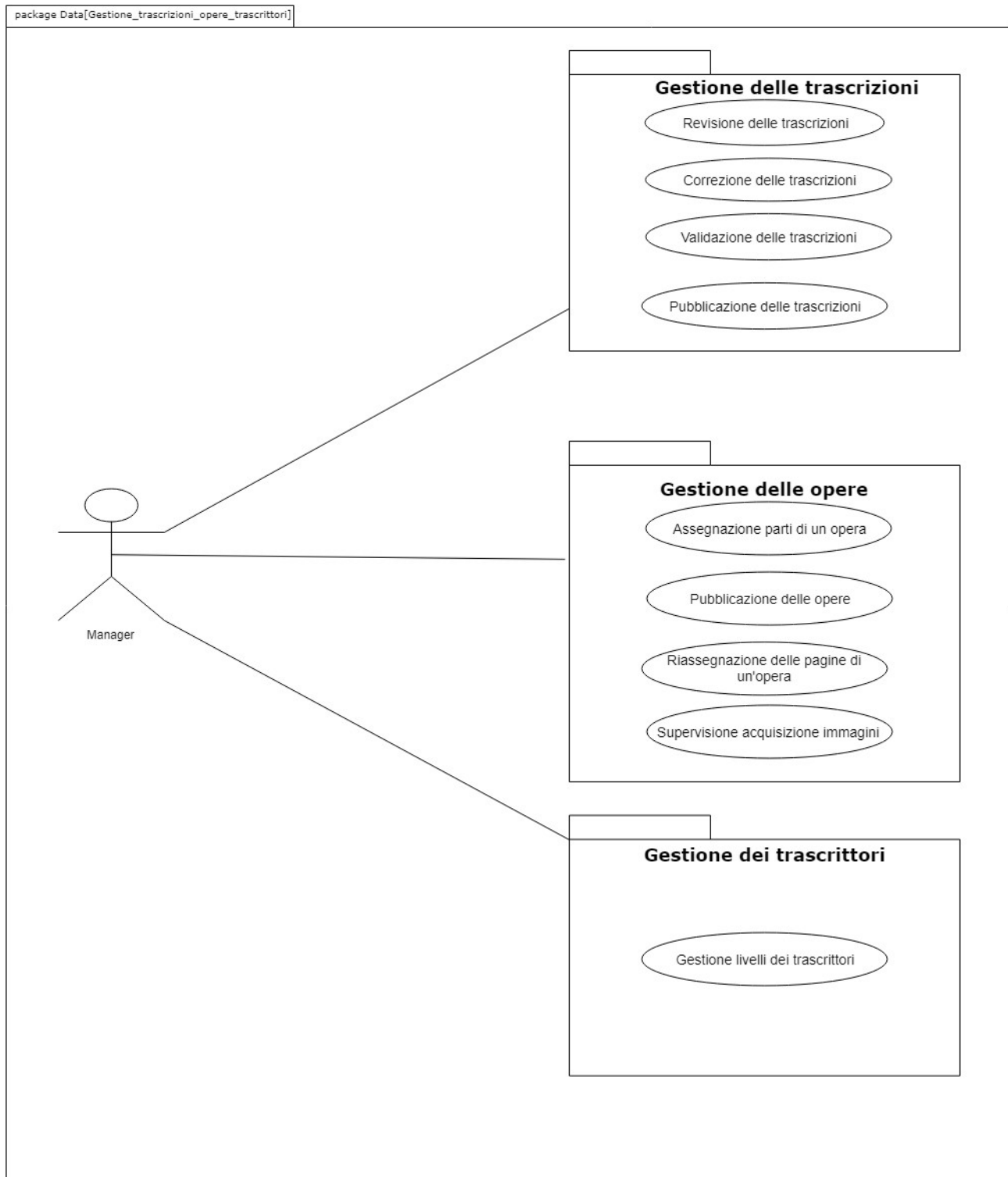
Use Case 2 -

USE CASE TRASCRITTORE



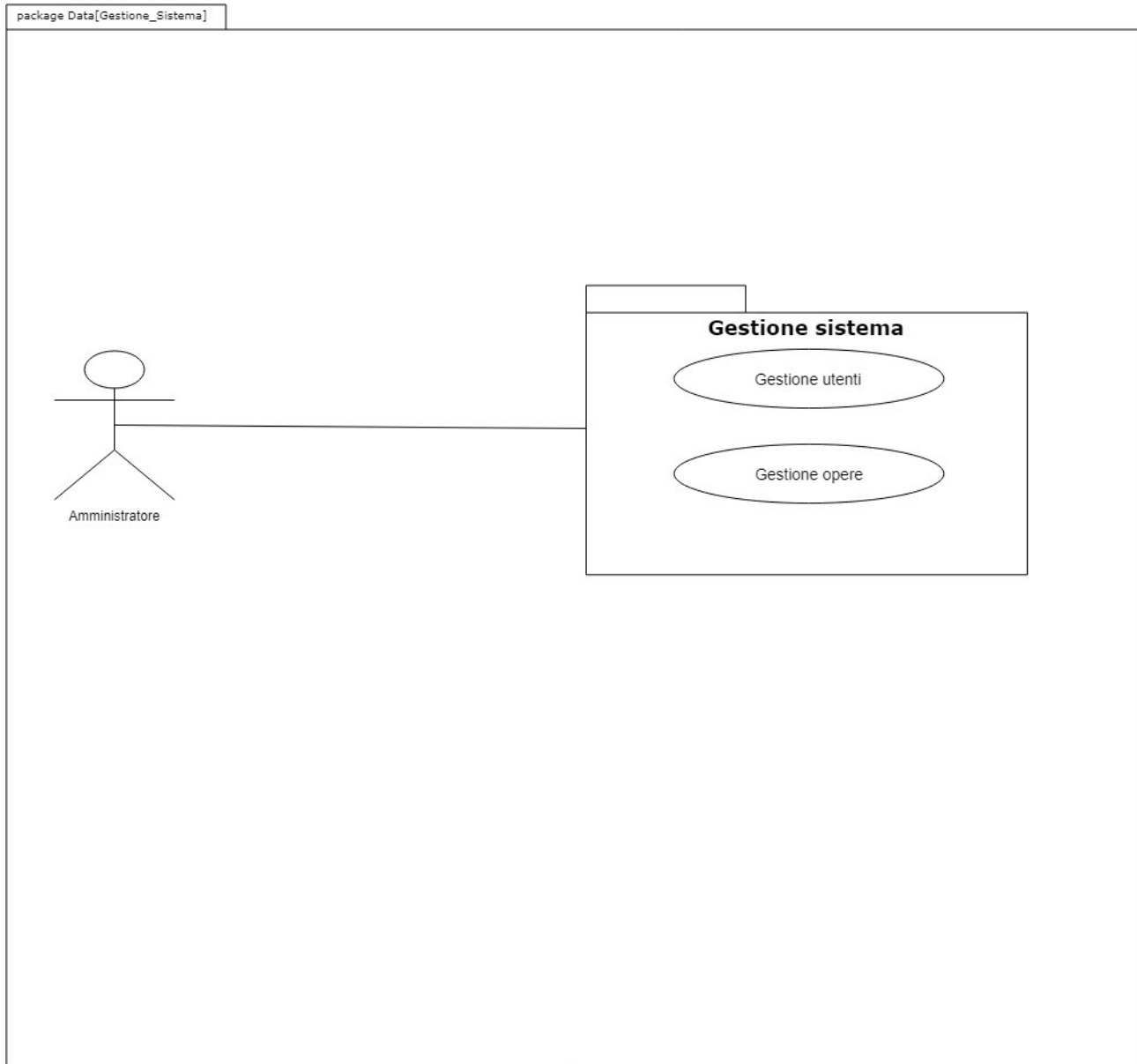
Use Case 3 - Trascrittore

USE CASE MANAGER



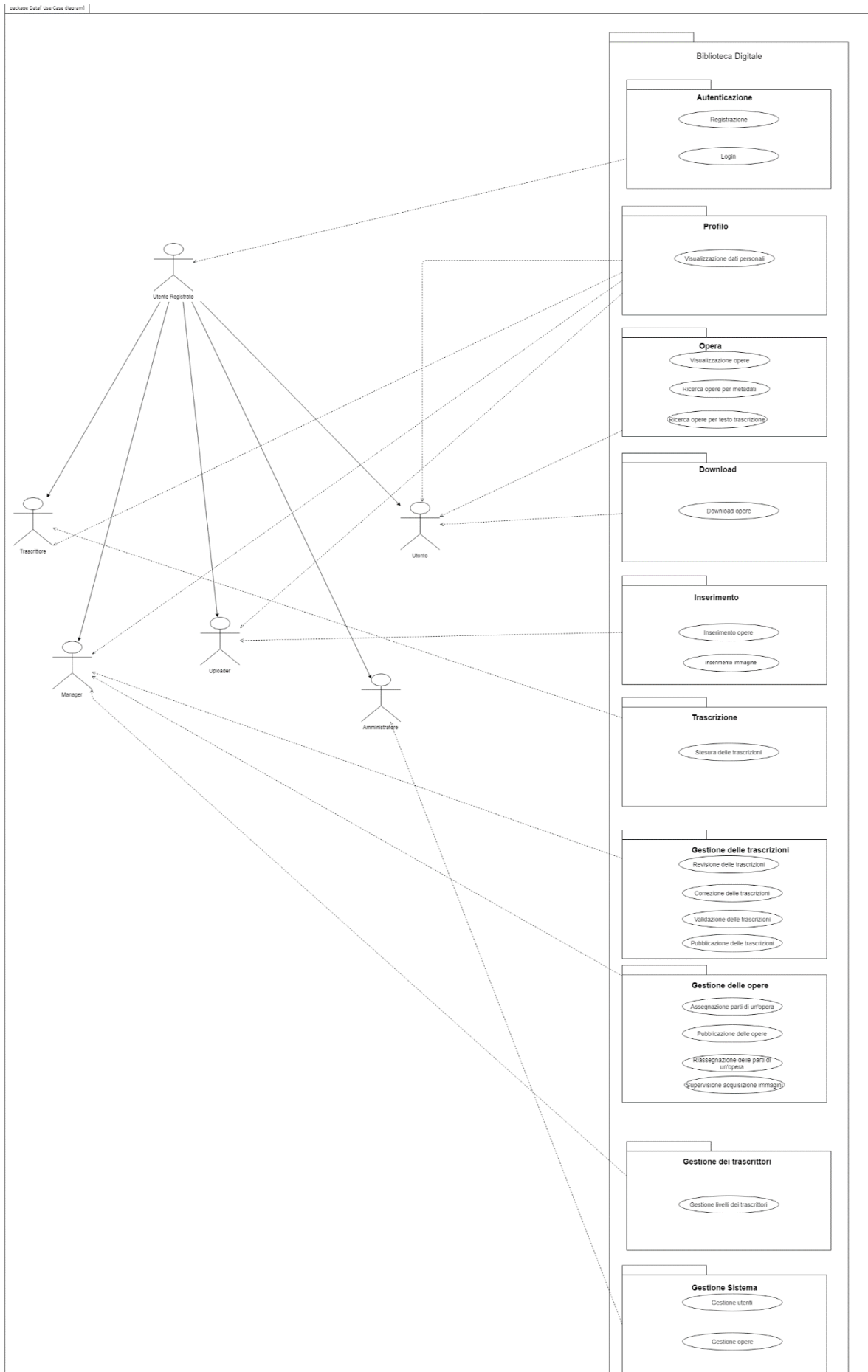
Use Case 4 - Manager

USE CASE AMMINISTRATORE



Use Case 5 –Amministratore

USE CASE GENERALE DEL SISTEMA



Scenari

Autenticazione: L'utente "Giovanni", il manager "Pippo", il trascrittore "Pluto", l'uploader "Andrea" effettuano la registrazione una tantum inserendo i propri dati anagrafici e quelli relativi alla biblioteca digitale (username e password). Questi ultimi dati, permettono a tutti gli utenti di accedere al proprio profilo e svolgere ciò che è a loro consentito.

Profilo: L'utente "Giovanni", il manager "Pippo", il trascrittore "Pluto", l'uploader "Andrea" visualizzano il proprio profilo all'interno della biblioteca digitale. Per l'utente il profilo è suddiviso in due sezioni: nella prima sezione è possibile visualizzare tutte le opere presenti nel catalogo, mentre nella seconda sezione ciascun utente potrà visualizzare i propri dati inseriti durante la registrazione (es. dati anagrafici, esperienze lavorative, ecc.). Anche per il manager, il trascrittore e l'uploader il proprio profilo è suddiviso in due sezioni: nella prima sono presenti tutte le informazioni e tutte le funzionalità riservate per ciascun utente mentre la seconda contiene le informazioni personali.

Opera: L'utente "Giovanni" si reca nell'apposita sezione riservata alle opere (cioè dove vengono visualizzate tutte le opere presenti nella biblioteca) e potrà visualizzarle una ad una oppure potrà ricercare, utilizzando l'apposito spazio di ricerca, le opere che desidera visualizzare.

Download: L'utente "Giovanni", dopo aver fatto richiesta all'amministratore "Paolo" e una volta ottenuto i privilegi, potrà eseguire il download delle opere che desidera. Per eseguire il download basta cliccare sul pulsante DOWNLOAD situato sotto ciascuna opera.

Inserimento: L'uploader "Andrea" quando vuole caricare una nuova immagine appartenente ad un'opera avverte dapprima sia il trascrittore che il manager e poi può inserire dall'apposita area la nuova immagine. Lo stesso vale anche nel caso in cui "Andrea" voglia caricare una nuova opera.

Trascrizione: Il trascrittore “Pluto”, quando viene caricata una nuova parte di un’ opera associa ad essa il corrispondente testo(trascrizione). Scritta la trascrizione il trascrittore “Pluto” avverte il manager “Pippo” così che quest’ultimo può decidere se validarla oppure no. In caso affermativo poi la trascrizione verrà pubblicata. Inoltre quando il trascrittore “Franco” e il trascrittore “Paolo” vogliono lavorare su una stessa trascrizione devono sincronizzarsi tra di loro al fine di evitare incomprensioni.

Gestione delle trascrizioni: Il manager “Pippo” una volta che ha eseguito l’accesso alla propria area può gestire tutto ciò che riguarda ciascuna trascrizione come: la sua revisione, la sua correzione, la sua validazione e la sua pubblicazione.

Gestione delle opere: Il manager “Pippo” una volta che ha eseguito l’accesso alla propria area può gestire tutto ciò che riguarda ciascuna opera come: la sua assegnazione ad uno o più trascrittori, la sua pubblicazione, la sua riassegnazione, la sua acquisizione.

Gestione dei trascrittori: Il manager “Pippo” una volta che ha eseguito l’accesso alla propria area può gestire ciascun trascrittore e ovvero può gestire i livelli di ciascuno di essi. I livelli rappresentano l’esperienza che un trascrittore ha accumulato fino a quel momento. Il livello minimo è 1 e quello massimo è 5.

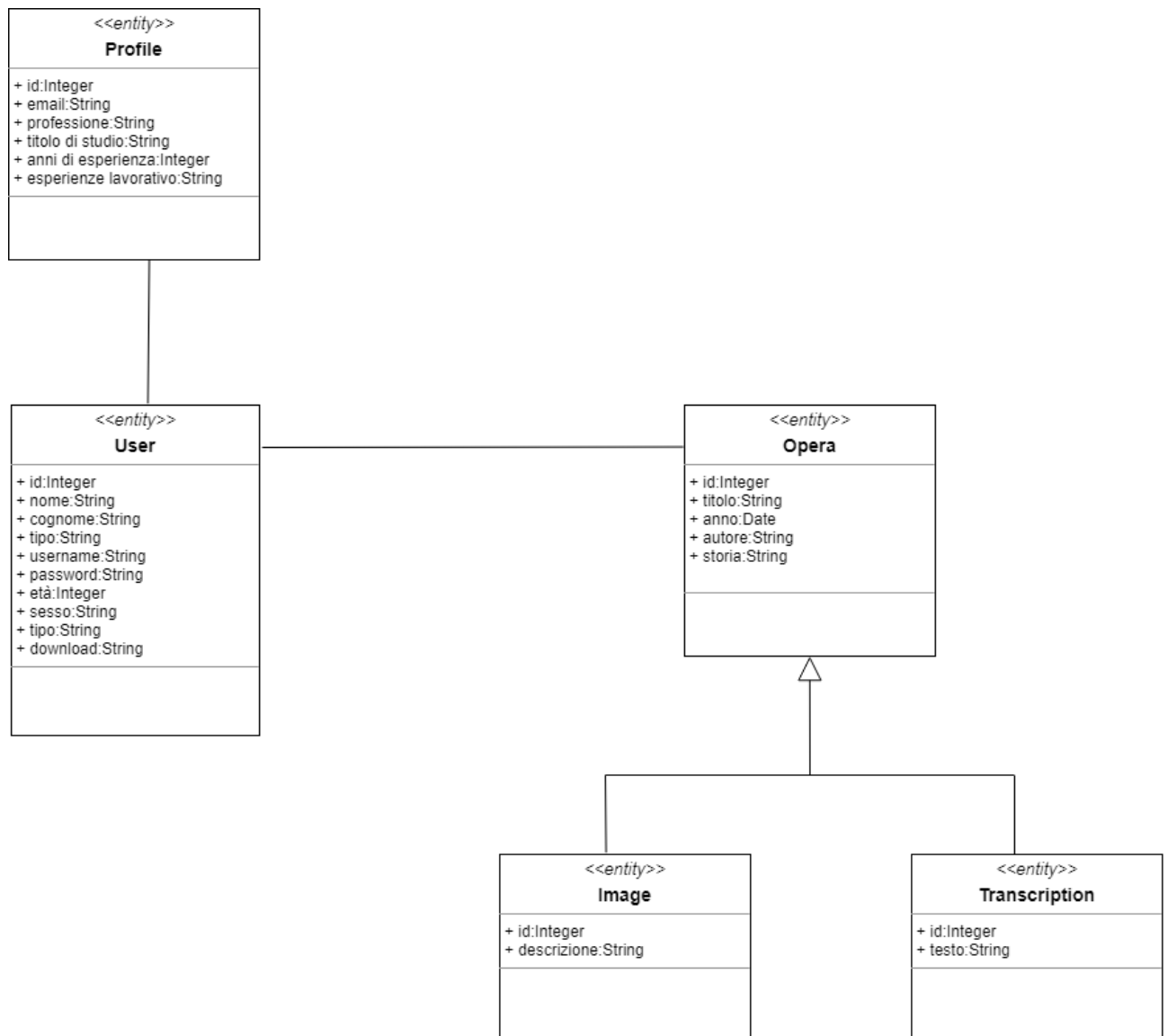
Gestione sistema: L’amministratore “Francesco” ha il compito di garantire il regolare funzionamento della Biblioteca Digitale. Quest’ultima deve essere sempre efficiente e di facile utilizzo sia per quanto riguarda le azioni che può compiere un trascrittore e sia per quanto riguarda le azioni che può compiere un uploader, ma anche per le relative azioni relative all’attività del manager. Un qualsiasi problema, riscontrato da un utente, da un manager, da un trascrittore e da un uploader, deve essere gestito e risolto

da Francesco. Infine Francesco, avendo il completo controllo sul database, può interagire con la tabella relativa a tutti gli utenti registrati sulla piattaforma, e promuovere a suo piacimento, un utente come “Giovanni” a trascrittore. In questo modo Giovanni e Pluto avranno le stesse funzionalità.

1.3 Modelli di dominio

Le classi di dominio che sono state individuati sono le seguenti:

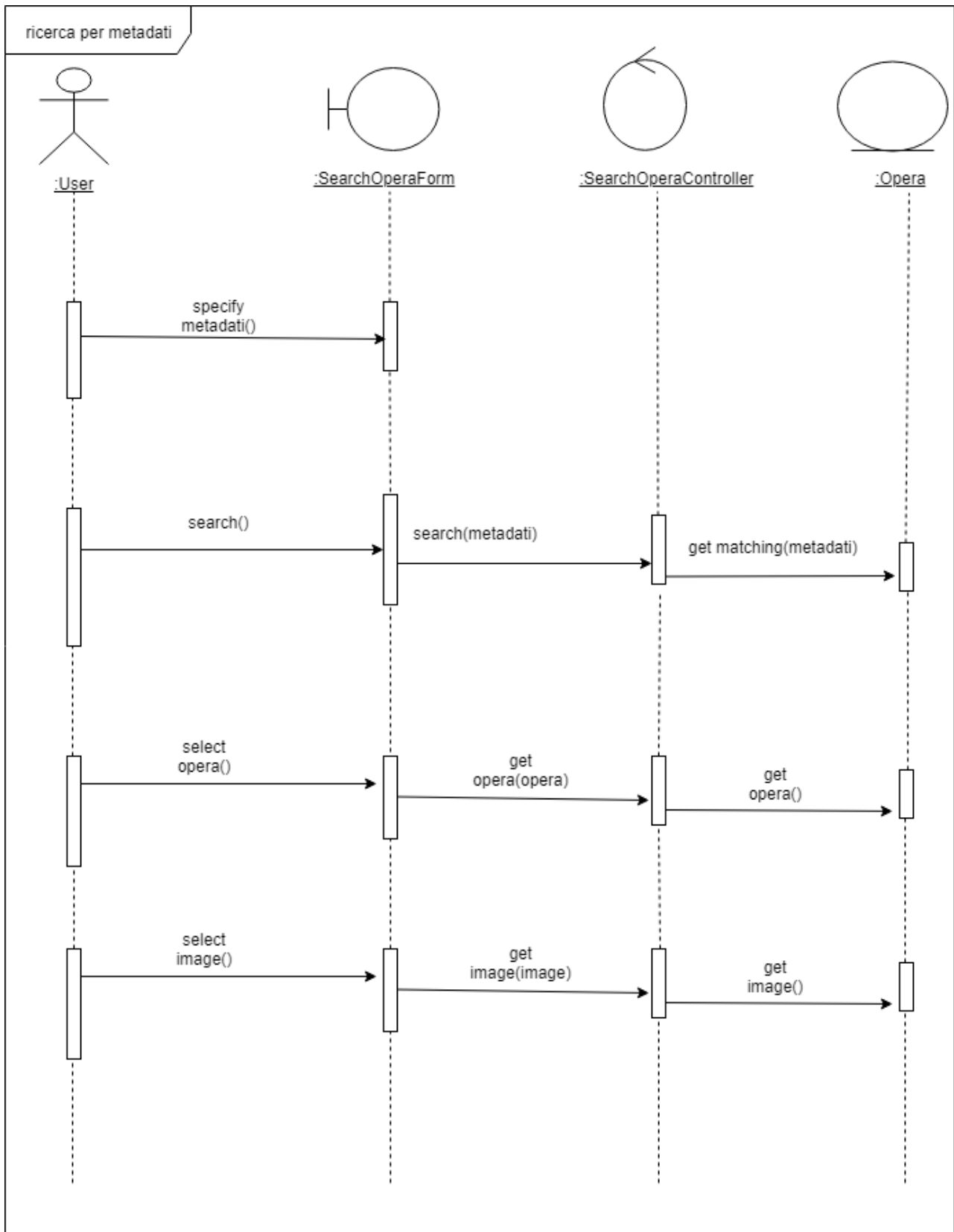
- User
- Profile
- Opera
- Image
- Transcription



1.4 Analisi

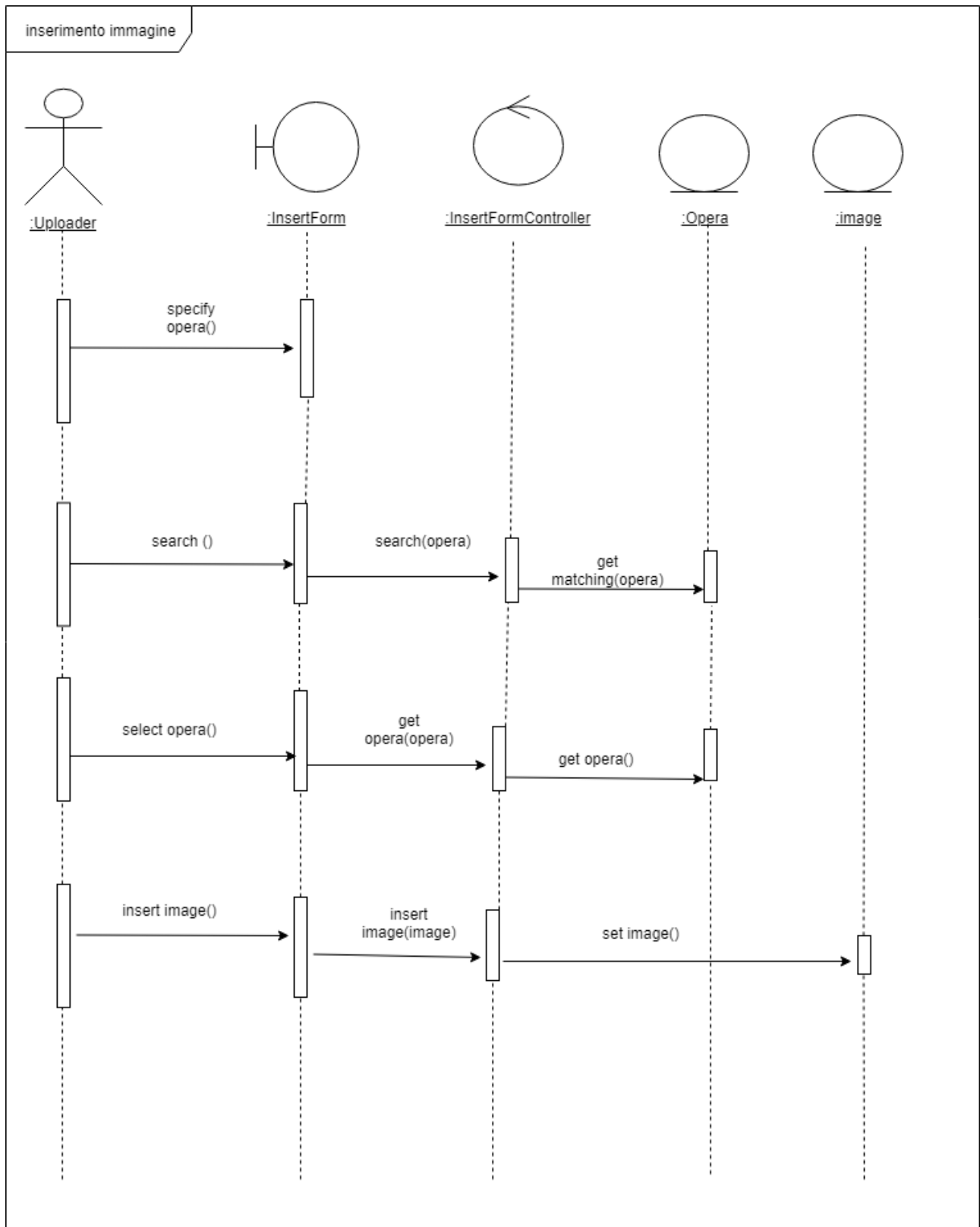
Per una corretta analisi finalizzata all'individuazione di classi entity, boundary e controller si è deciso di realizzare un sequence

diagram per uno degli use case più significativo per ciascun attore.



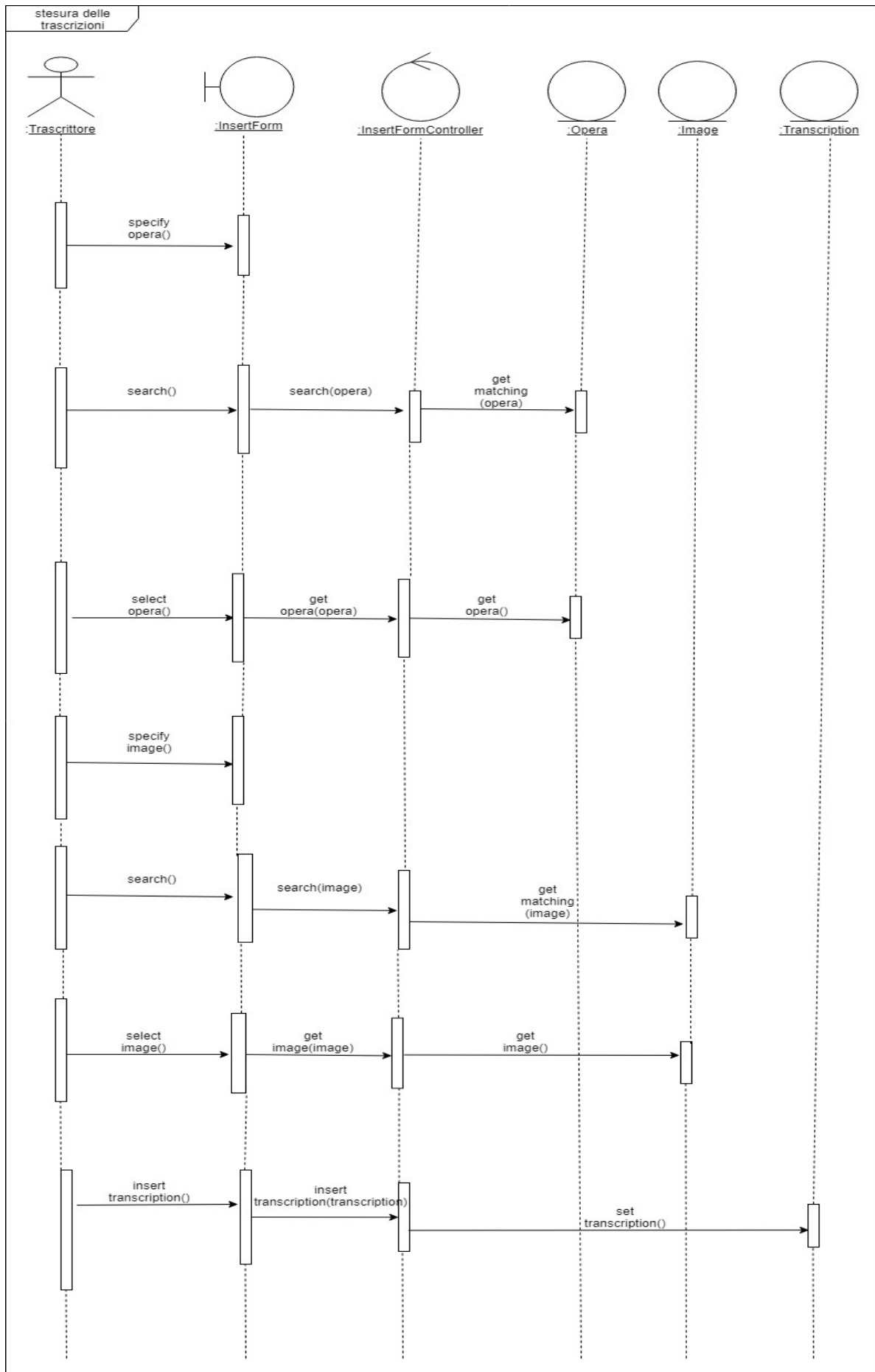
Questo sequence diagram fa riferimento allo use case dell'utente chiamato "ricerca per metadati" e rappresenta lo scenario in cui

un utente intende effettuare una ricerca per metadati al fine di trovare l'opera che desidera e visualizzare l'immagine che preferisce.

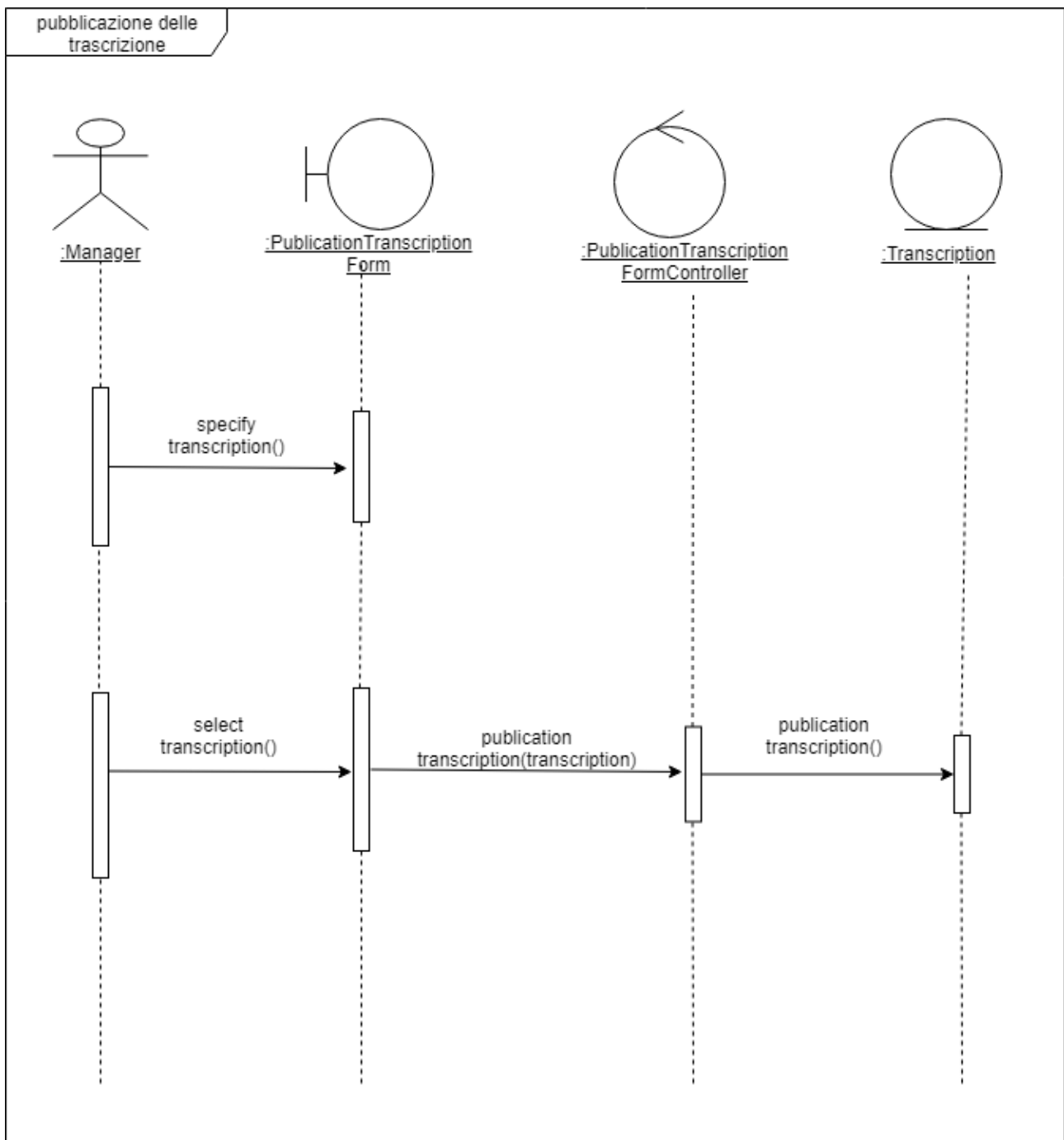


Questo sequence diagram fa riferimento allo use case dell'uploader chiamato "inserimento immagine" e rappresenta lo

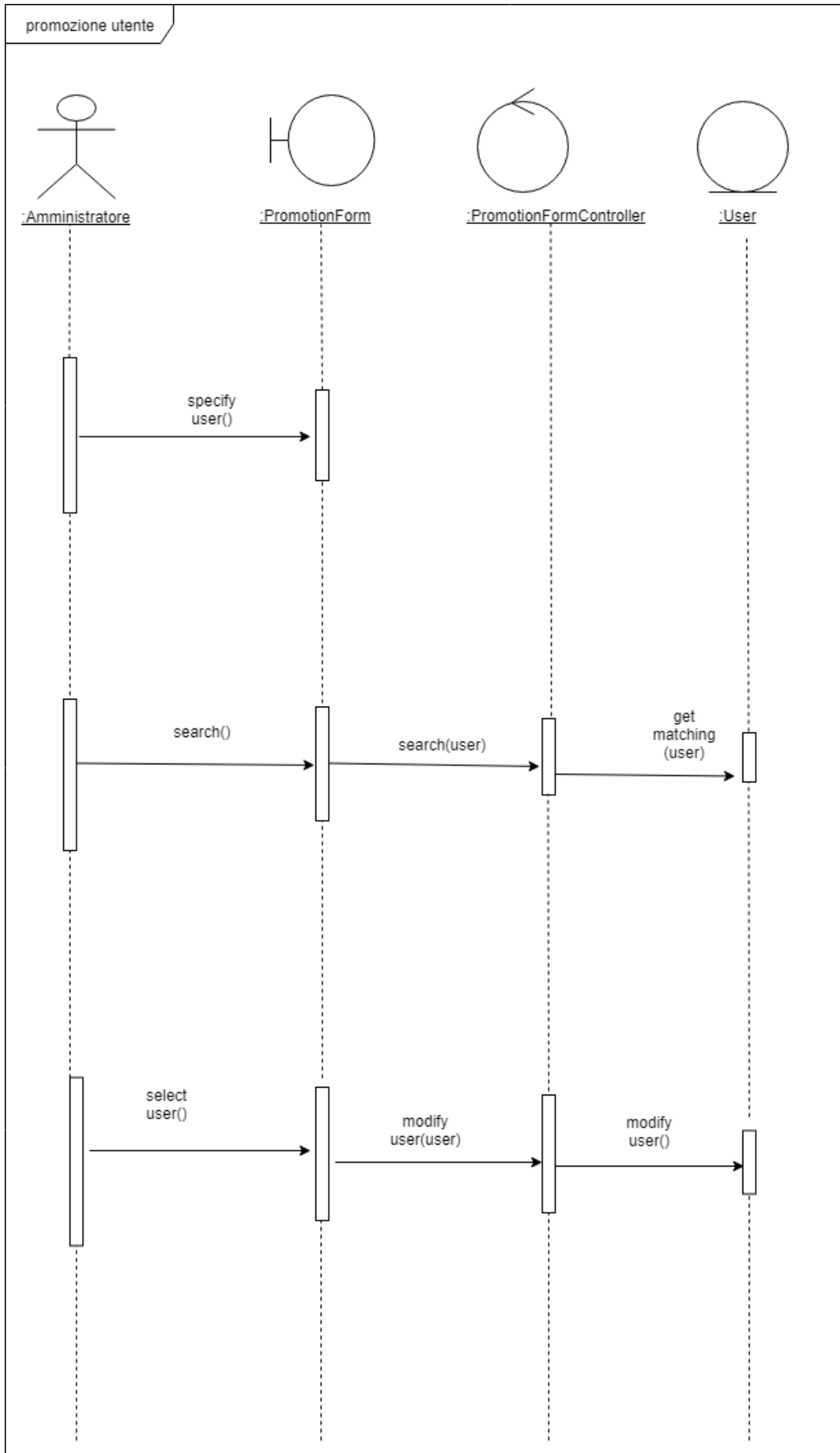
scenario in cui l'uploader intende inserire una nuova immagine inerente ad un'opera.



Questo sequence diagram fa riferimento allo use case del trascrittore chiamato “stesura delle trascrizioni” e rappresenta lo scenario in cui un trascrittore intende scrivere la trascrizione relativa ad una immagine di un’opera stabilita.



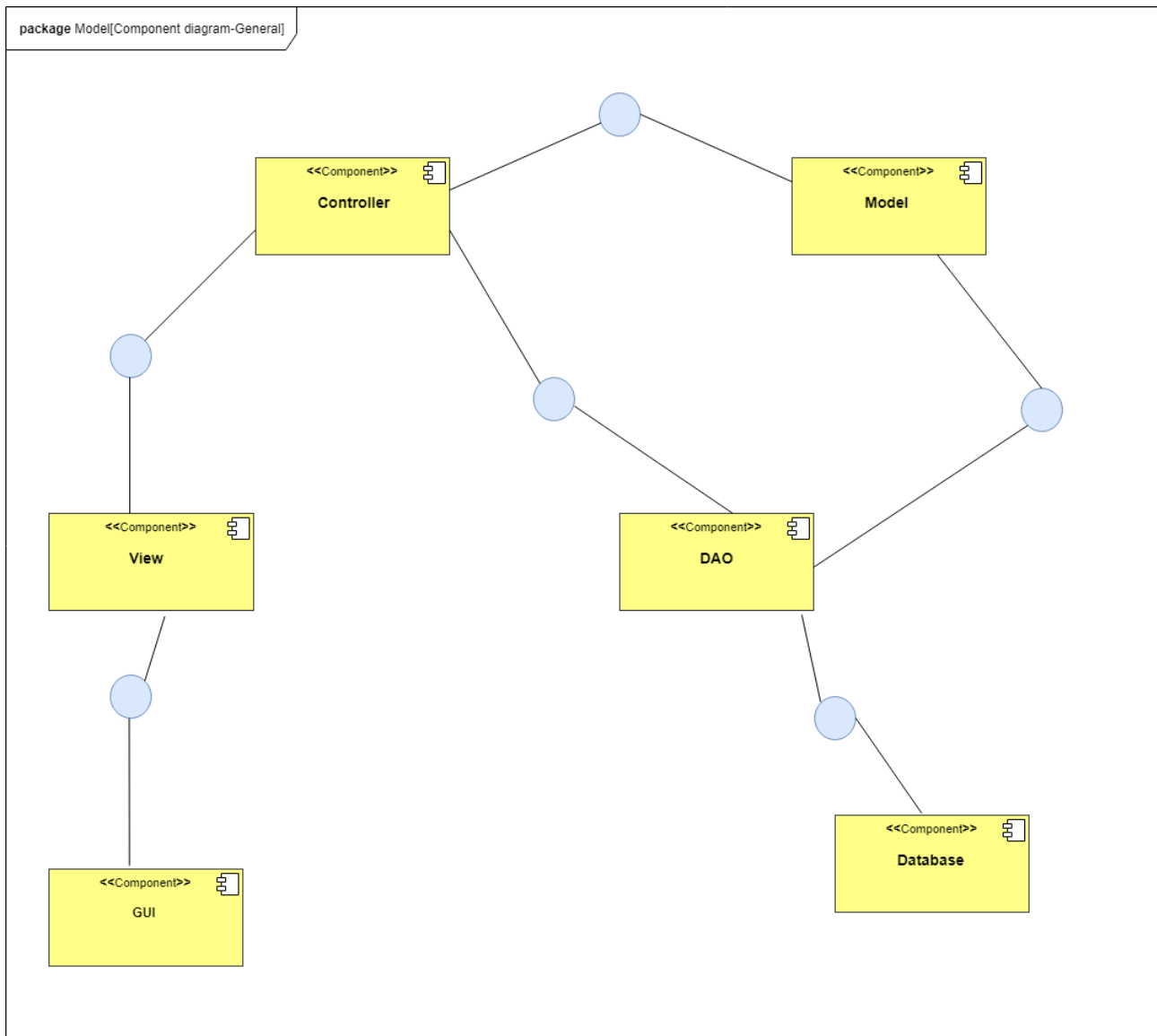
Questo sequence diagram fa riferimento allo use case del manager chiamato “pubblicazione delle trascrizioni” e rappresenta lo scenario in cui il manager intende pubblicare, sempre se la approva, una trascrizione scritta da un trascrittore.



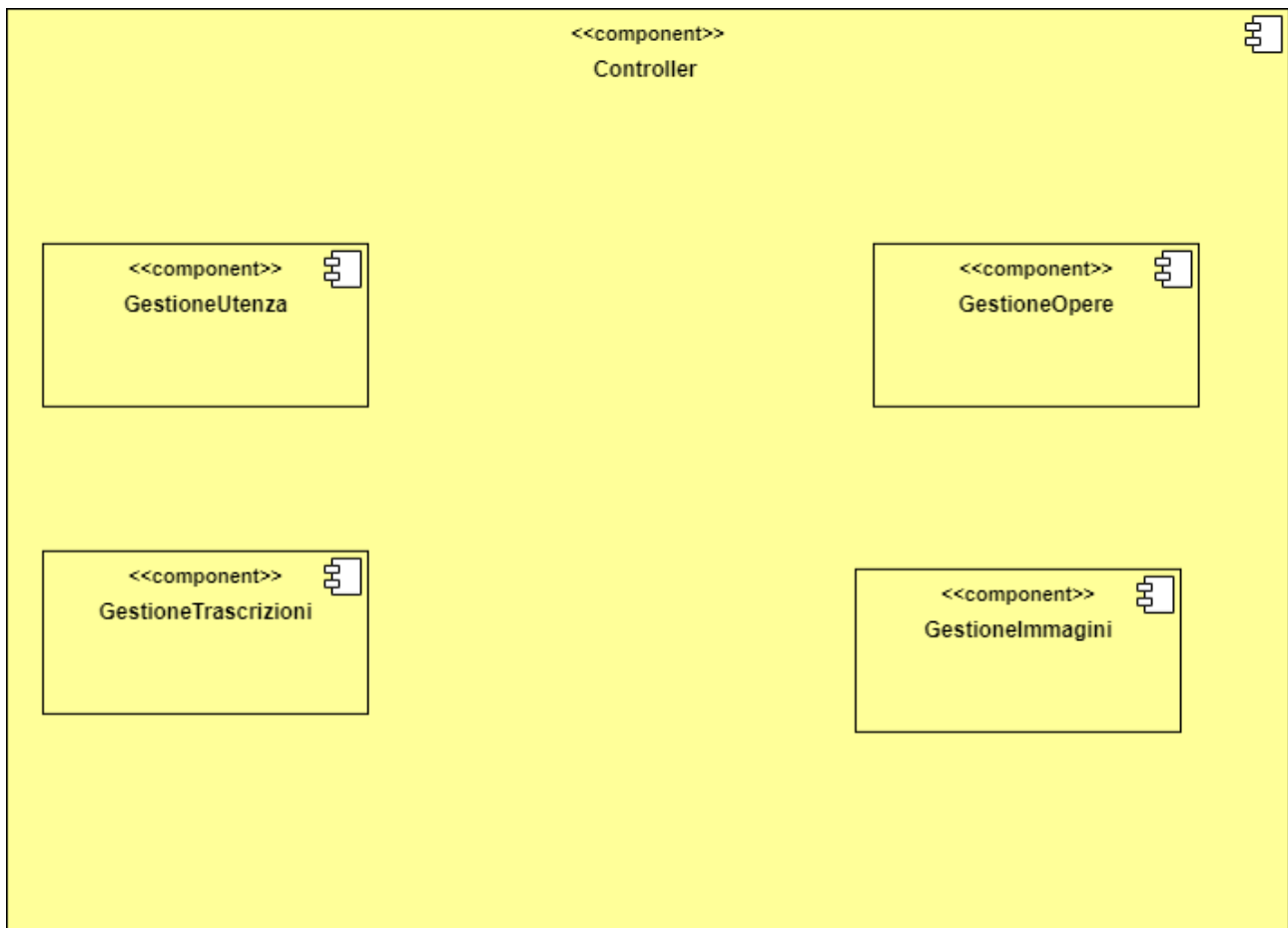
Questo sequence diagram fa riferimento allo use case dell'amministratore rappresentante lo scenario in cui l'amministratore promuove un semplice utente in un trascrittore.

System Design

2.1 Modello dell'architettura del sistema



Component Diagram 1 - Generale



Component Diagram 2 –Controller

2.2 Descrizione dell'architettura

Il component diagram generale presente nella sezione 2.1 ha lo scopo di elencare i componenti principali del nostro sistema software e tutte le relazioni che intercorrono tra essi.

Partendo dal livello più vicino, fino ad arrivare a quello dell'utente finale, individuiamo la componente Gui dedicata alla

gestione dell'aspetto grafico, svolgendo un compito di intermediazione tra user e sistema.

Un elemento connesso a quest'ultimo componente è quello della View il quale costituisce una sorta di “ponte” tra Presentation (questo contiene classi per l'intera interfaccia utente, per consentire all'utente di visualizzare i dati dal sistema e inserire nuovi dati) e la logica vera e propria del sistema, identificata nel Controller. Essenzialmente, quest'ultimo, si occupa della gestione di tutte quelle che sono le funzionalità essenziali. Data la sua complessità è stato suddiviso in sotto componenti quali GestioneUtenza , GestioneOpere, GestioneImmagini e GestioneTrascrizioni. Il primo si occupa dell'elaborazione e del mantenimento delle informazioni relative all'utente finale del sistema; il secondo si occupa di gestire tutte le opere presenti nella biblioteca; il terzo sotto componente del Controller è delegato al caricamento, al mantenimento e all'elaborazione delle immagini corrispondenti a ciascuna opera; ed infine l'ultimo sotto componente si occupa della gestione di tutte le trascrizioni presenti .

Troviamo poi il Model, che contiene dati strutturati in classi con i relativi metodi di accesso, necessari per la manipolazione da parte del Controller. Le stesse verranno organizzate in un Database che ne permette la memorizzazione fisica e la persistenza. Questa funzione viene delegata ad un altro componente del sistema, il DAO, Data Access Object.

2.3.1 Descrizione delle scelte

Il component diagram realizzato è stato sviluppato secondo i principi classici del paradigma OOP, separando logicamente i vari moduli che costituiscono ciascun componente. L'interfaccia, con l'utente, rappresentata dalla componente Presentation, è composta da due parti: quella di front-office dedicata a tutti gli utenti base che permette la visualizzazione delle opere presenti e la visualizzazione del loro profilo sulla biblioteca, e una parte di back-office, utilizzata unicamente dall'amministratore per la gestione del sistema e del relativo database.

2.3.2 Design Pattern

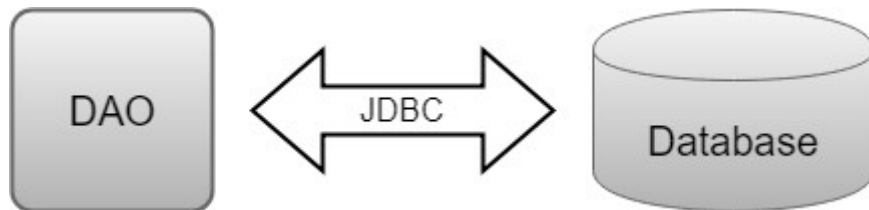
DATA ACCESS OBJECT

È stato scelto di utilizzare il DAO come design pattern per gestire la persistenza dei dati del nostro sistema su Database. Aniché far effettuare le interrogazioni dal componente Model, utilizzando questo pattern, deleghiamo il compito di aggiornamento, di aggiunta e di modifica dei dati del database al DAO.

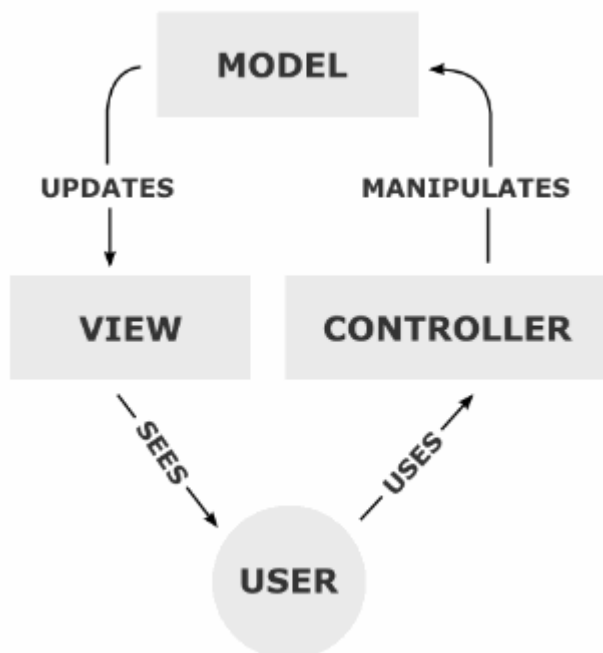
Il pattern si basa su:

- **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 o Value Object:** Un semplice oggetto contenente metodi getters e setters per leggere/memorizzare dati ottenuti attraverso le classi DAO.



MODEL-VIEW-CONTROLLER



Il pattern è basato sulla separazione dei compiti tra tre componenti del sistema:

- **Model:** Uno dei componenti più importanti dell'applicazione è il model, che, incapsulando lo stato dell'applicazione, definisce i

dati e le operazioni che possono essere eseguite su questi. Può contenere dei metodi utilizzati per notificare al controller la modifica di alcuni dati.

- **View:** Si occupa di gestire la presentazione dei dati. In altre parole, tutto quello che è correlato alla creazione/gestione della GUI (Graphical User Interface), ovvero lo strumento che gli utenti utilizzano per interagire con il sistema; ciò viene effettuato dalla View in maniera semplice e user-friendly.
- **Controller:** Questo componente ha la responsabilità di trasformare le iterazioni tra utente e View, in azioni eseguite dal Model. Ma il Controller non rappresenta un semplice "ponte" tra View e Model. Realizzando la mappatura tra input dell'utente e processi eseguiti dal Model e selezionando la schermate della View richieste, il Controller implementa la logica di controllo dell'applicazione. Questa logica può infatti risultare molto complessa tanto da poter essere decomposta in ulteriori sotto componenti proprio come è stato fatto. Utilizzando questo pattern, e quindi la separazione in tre strati, è possibile mantenere separati i concetti avendo un duplice vantaggio ovvero codice di facile manutenzione e modulare.

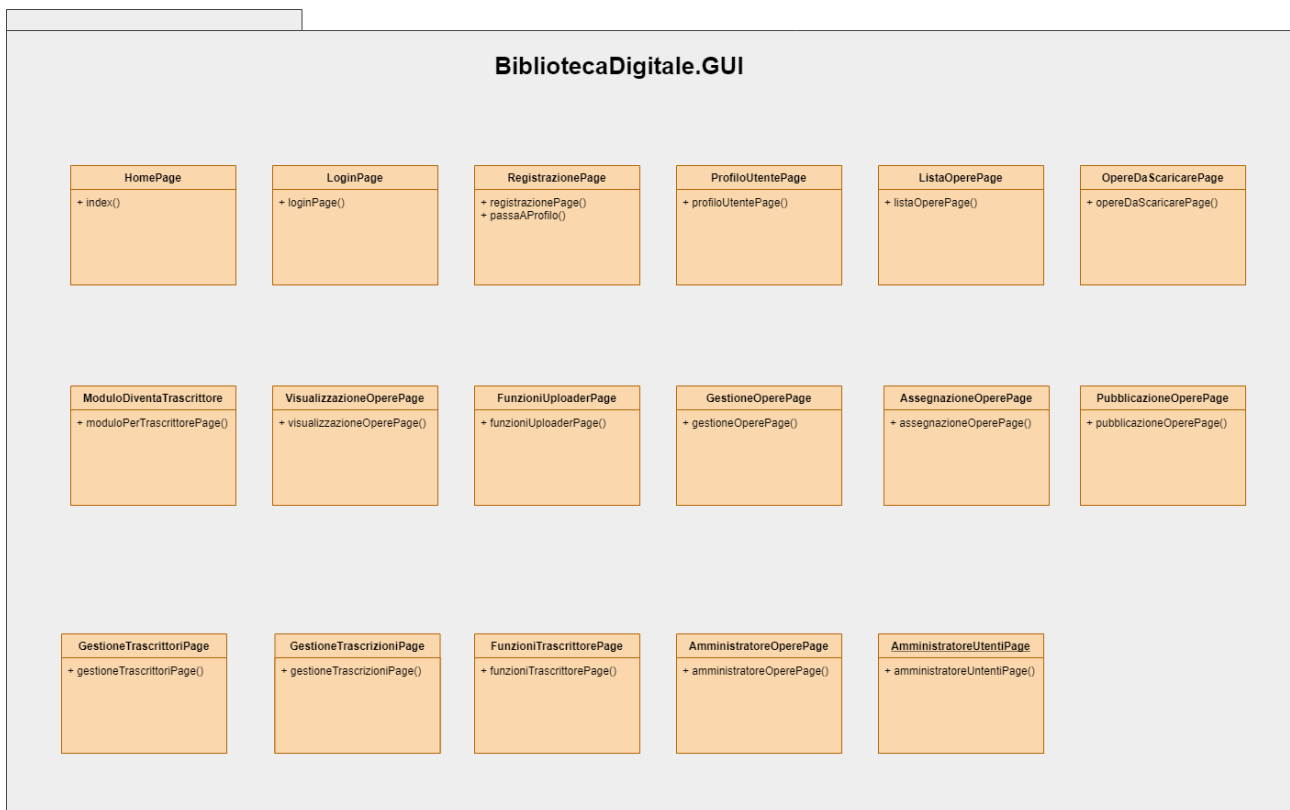
Software Design

La Biblioteca Digitale è stata separata in diversi package, con lo scopo di mantenere una specifica organizzazione del sistema, ma anche per garantire una separazione logica delle varie classi. Sono stati individuati i seguenti package:

- GUI
- View
- Controller
- Model
- DAO
- Database

Di seguito sono elencati i class diagram per ciascun package sopra citato, che hanno il compito di descrivere tipi di entità con relative caratteristiche, operazioni ed eventuali correlazioni logiche tra gli stessi.

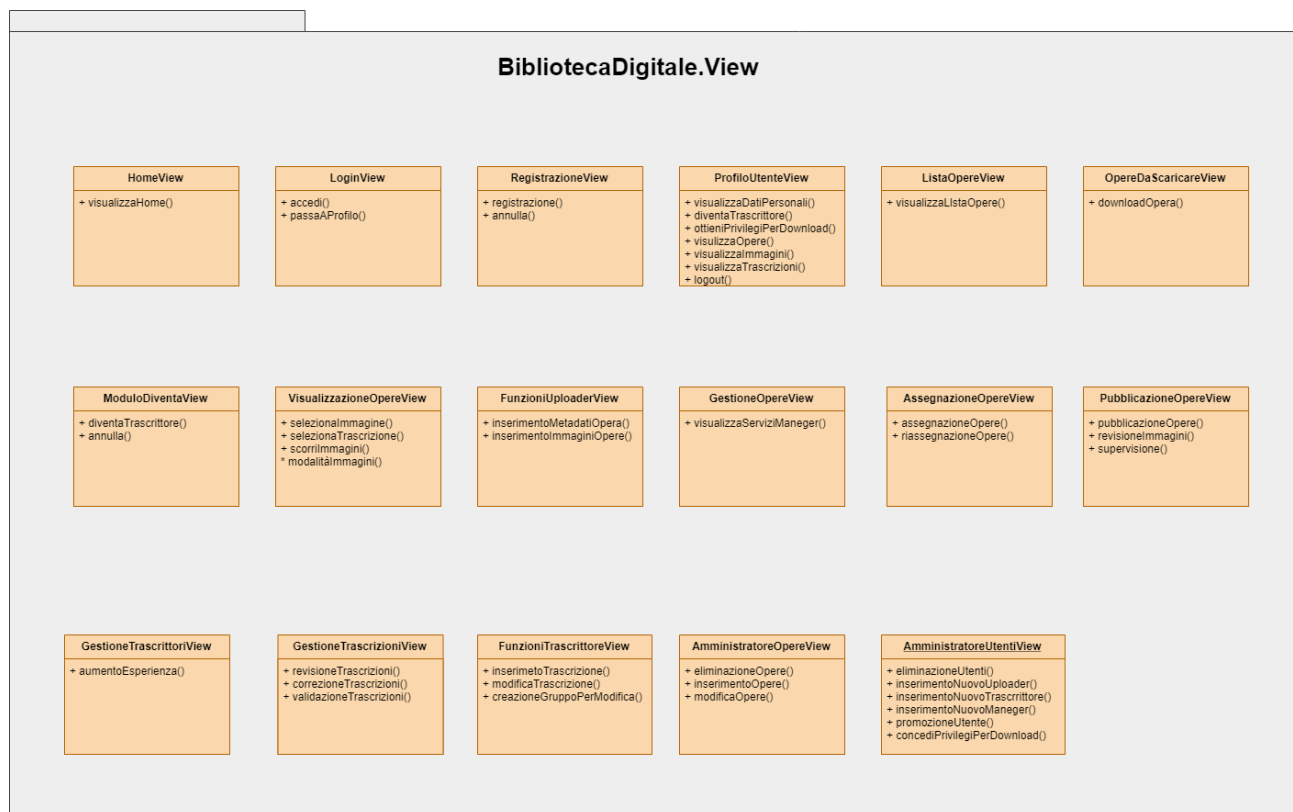
Class diagram - GUI



Class Diagram 1 – GUI

Questo package è costituito da tutte le classi utilizzate per la gestione dell'aspetto grafico e appunto della GUI (Graphical User Interface). Ogni classe del pacchetto costituisce una pagina del sistema.

Class diagram - View

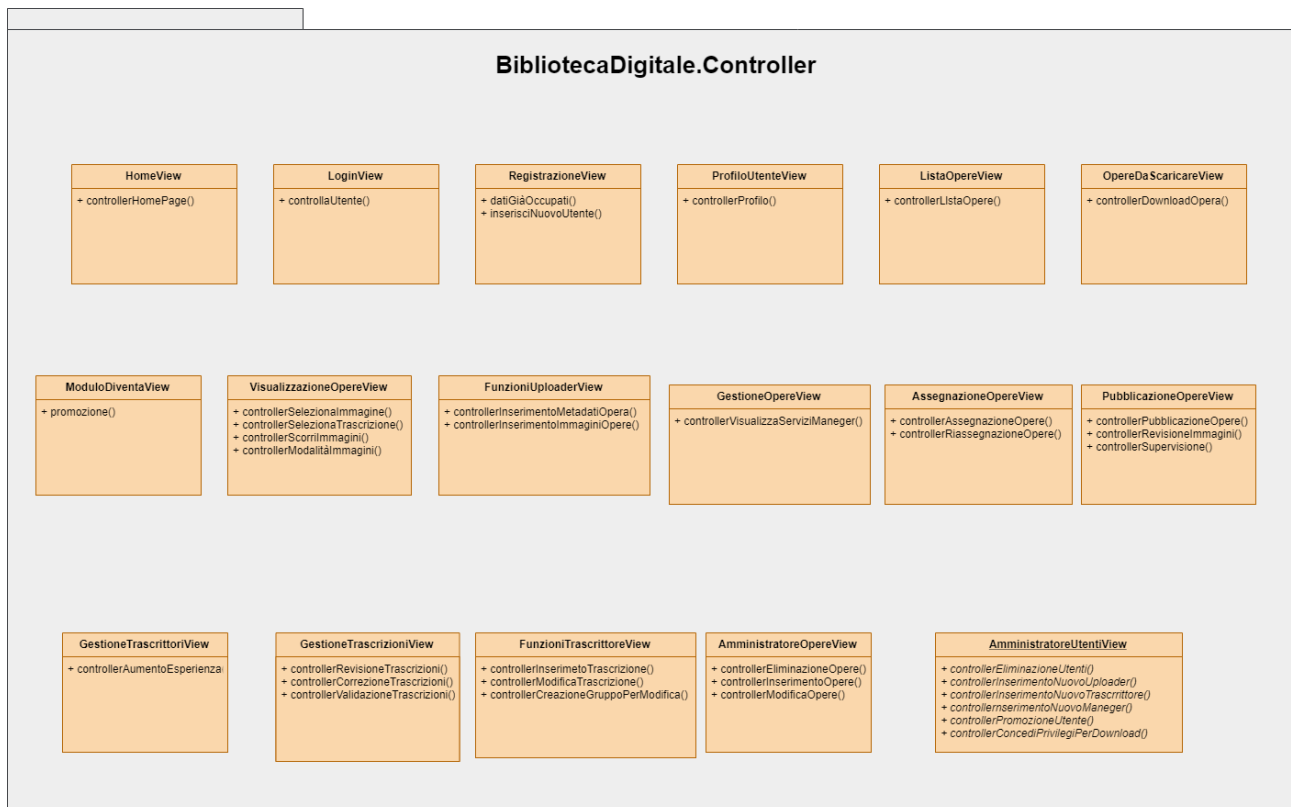


Class Diagram 2 – View

Il package GUI, relativo alla creazione ed alla gestione dell'interfaccia grafica si avvarrà dell'utilizzo di metodi presenti in un altro package del sistema, quello chiamato View. Esso è composto da una serie di classi (una per ogni pagina della GUI) i

cui metodi si occupano della gestione dell'input dell'utente. Infatti secondo il pattern MVC, la view ha il compito di notificare al Controller l'avvenuto input utente.

Class diagram - Controller

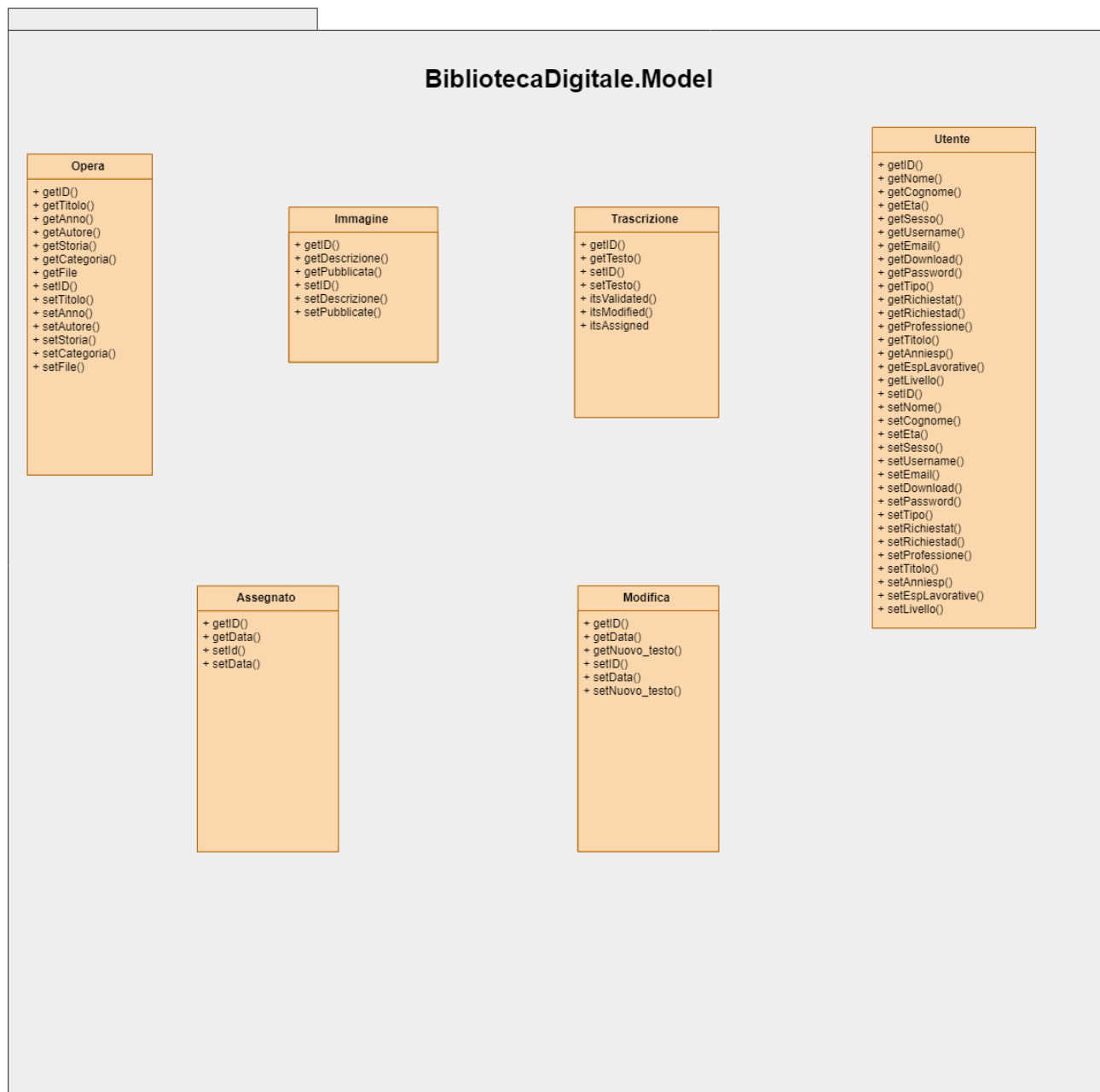


Class Diagram 3 - Controller

Così come accennato nel precedente paragrafo, un altro package del sistema è il Controller. Questo costituisce la logica del sistema occupandosi di “rispondere” alle iterazioni dell'utente con la visualizzazione di risultati.

Per ogni classe del package View (e quindi per ognuna di quella della GUI) è presente una rispettiva classe nel controller.

Class diagram - Model



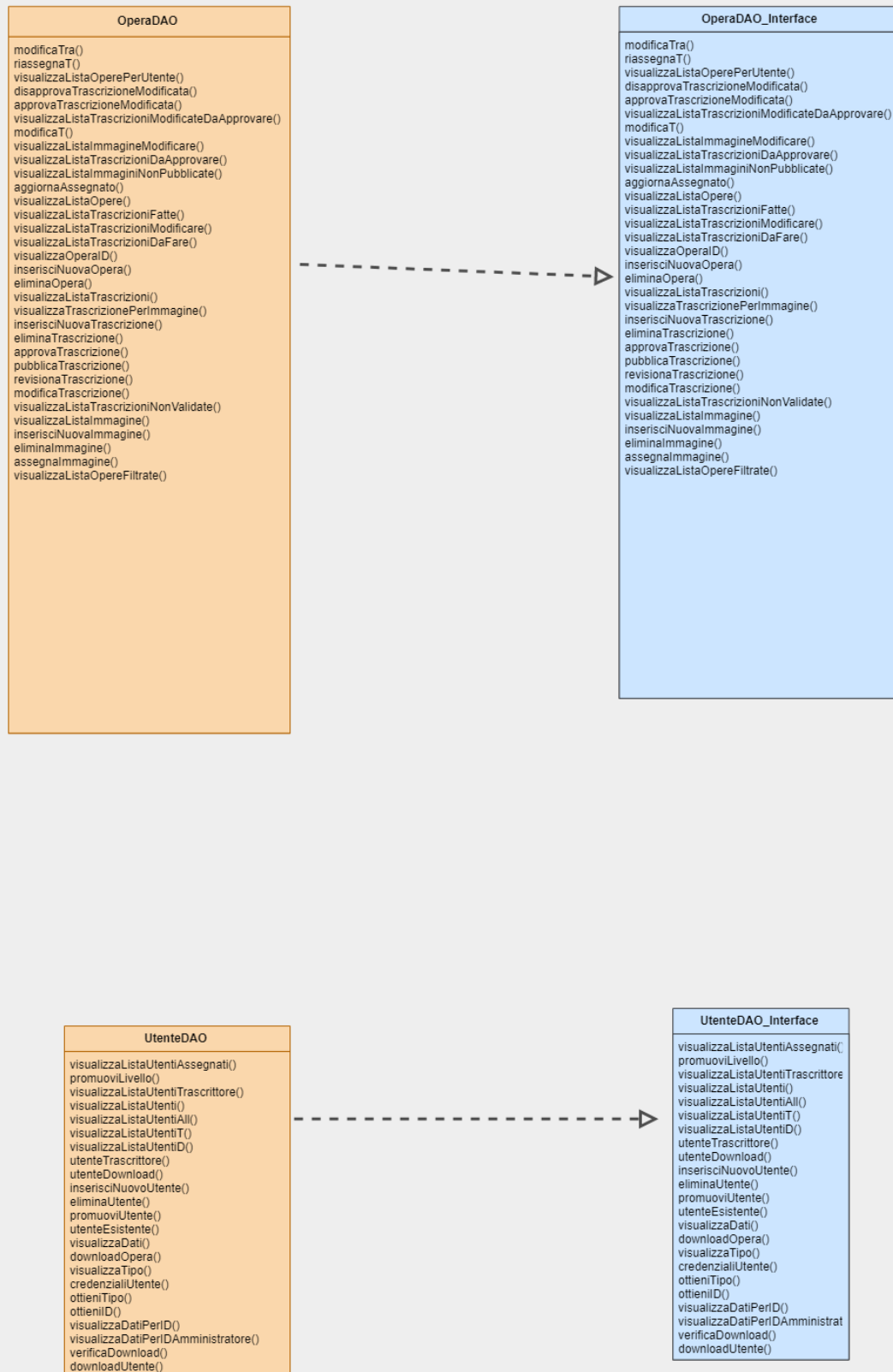
Class Diagram 4 – Model

Nel package Model sono presenti le classi relative alle principali entità del sistema:

- **Opera:** Rappresenta l'opera nel dettaglio, quindi tutte le informazioni che sono di interesse per una particolare opera e che quindi il nostro sistema deve poter memorizzare. Per ciascuna opera appartenente alla biblioteca sono state individuate le caratteristiche più importanti che costituiscono gli attributi della classe Opera. Per questa classe (come per le altre presenti in questo package) sono presenti metodi getters e metodi setters per ogni attributo.
- **Immagine:** Rappresenta le immagini relative a ciascuna opera. Ciascuna immagine è caratterizzata da un ID che la identifica univocamente e da una descrizione.
- **Trascrizione:** Rappresenta le trascrizioni relative a ciascuna opera. Ciascuna immagine è caratterizzata da un ID che la identifica univocamente e da un testo che appunto descrive l'immagine ad essa associata.
- **Utente:** Rappresenta l'utente del sistema, cioè l'utente registrato che può avere accesso a tutte le opere presenti nella biblioteca. Per ciascun utente le informazioni di interesse sono memorizzate sotto forma di attributi
- **Assegnato:** Serve per memorizzare tutte le trascrizioni e i rispettivi utenti trascrittori .
- **Modifica:** Serve per memorizzare tutte le modifiche che ciascun trascrittore effettua su determinate trascrizioni.

Class diagram - DAO

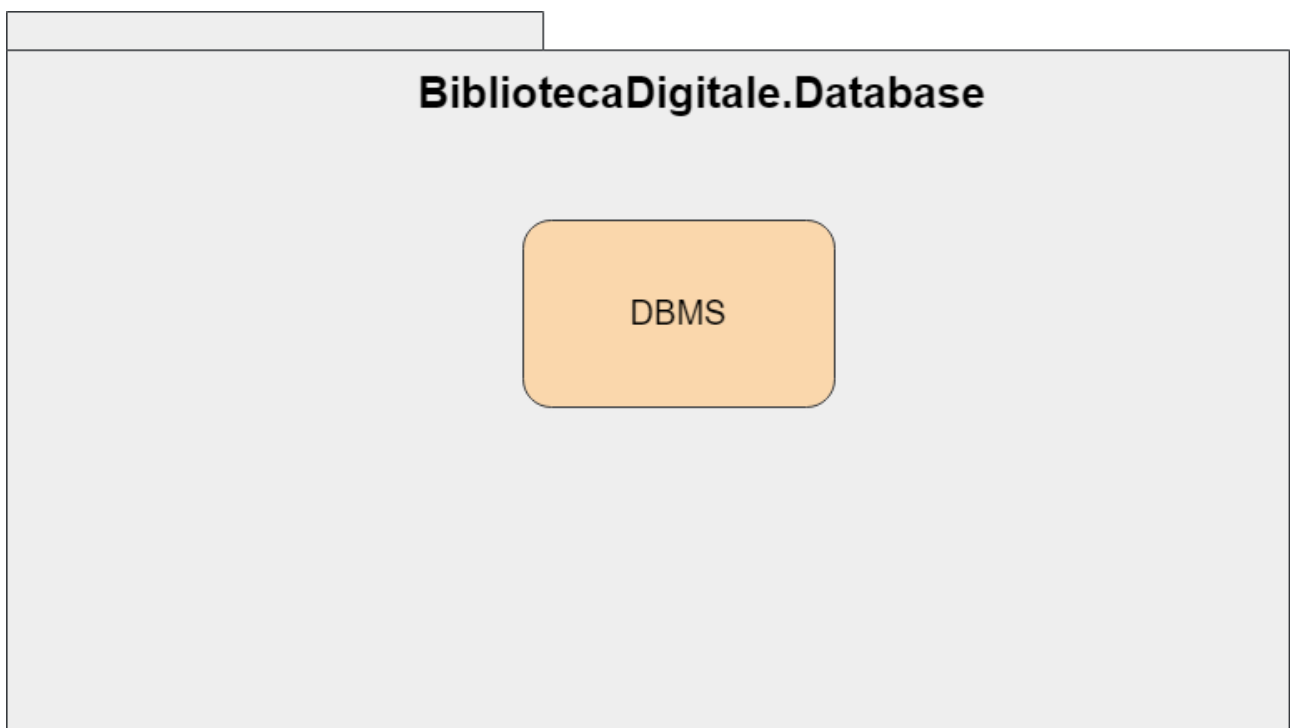
BibliotecaDigitale.DAO



Class Diagram 5 – DAO

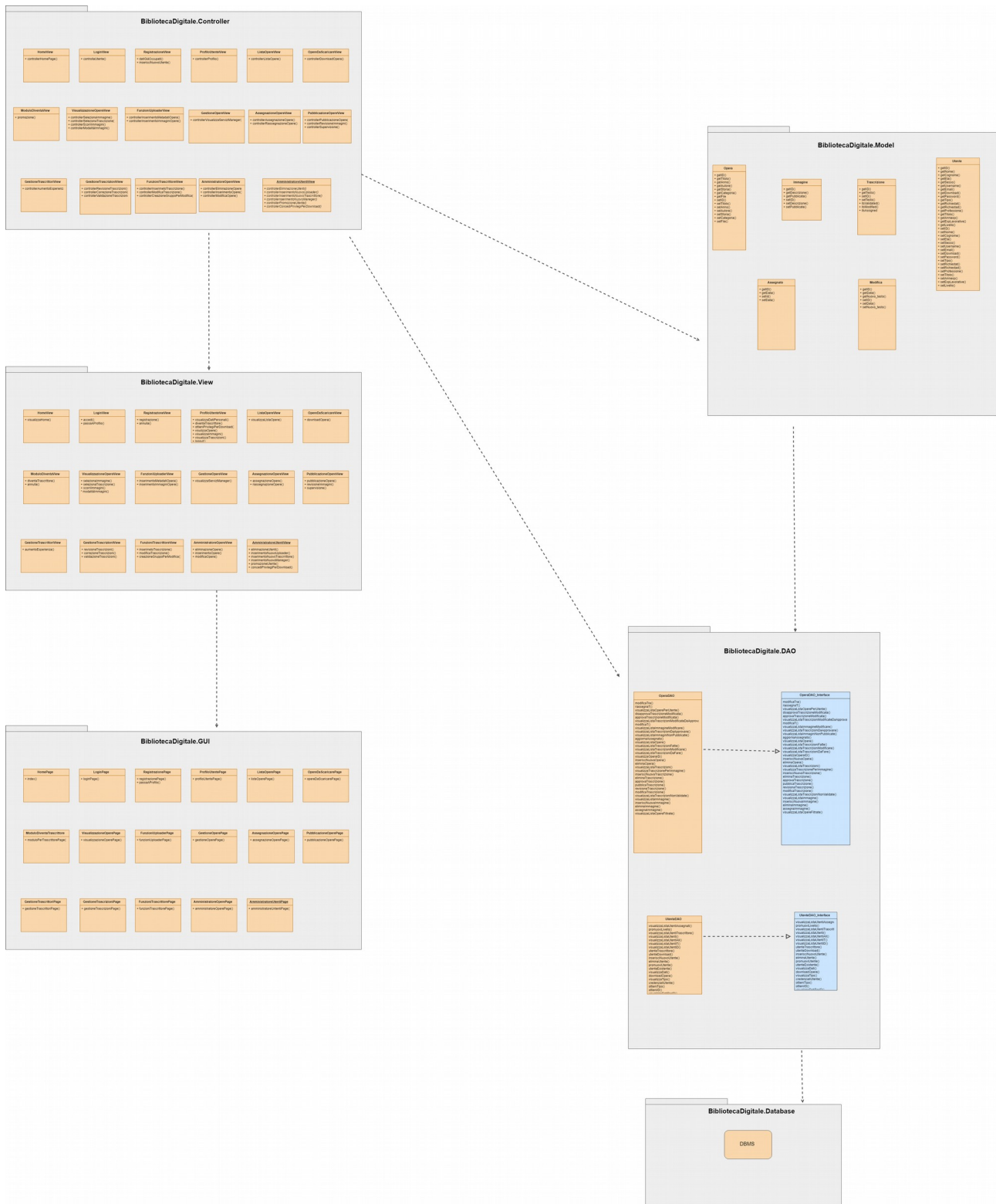
Rispecchiando il pattern DAO, nel sistema troviamo un package chiamato DAO, dedicato all'iterazione con il Database e quindi alla gestione della persistenza dei dati. Per ogni concetto principale del sistema è stata realizzata un'interfaccia contenente la dichiarazione dei metodi implementati dalle rispettive classi. Ogni qualvolta sarà necessario interagire con il database (ad es. quando si dovrà inserire un nuovo utente appena registrato o quando si dovrà inserire trascrizione per una particolare opera) lo si farà tramite l'ausilio di una classe di questo package.

Class diagram - Database



Class Diagram 6 – Database

Class diagram - Completo

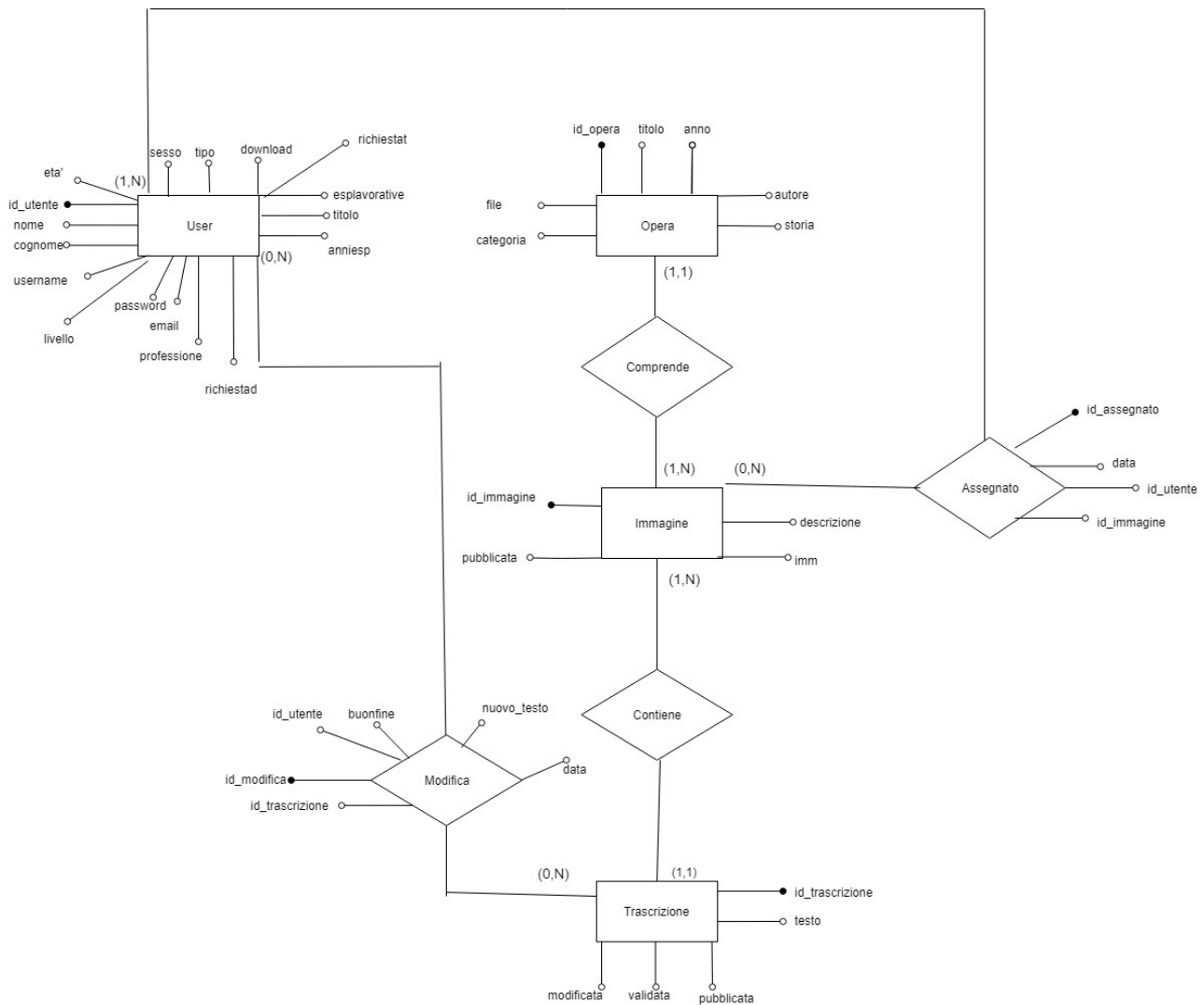


Class Diagram 7 – Completo

Database

Il Database risulta essere una componente molto importante del nostro sistema. Nel suo interno troveremo la tabella che contiene dati relativi agli utenti registrati nella biblioteca; la tabella contenente i dati personali di ciascun utente registrato; la tabella relativa a ciascuna opera presente nelle biblioteca; la tabella relativa ai giochi contenente i dati relativi a ciascuno di essi a cui è possibile giocare nella piattaforma; la tabella relativa allo storico cioè la tabella che conterrà per ciascun utente le opere che ha visualizzato; la tabella relativa alle immagini rappresentative di ciascuna opera; la tabella relativa alle trascrizioni corrispondenti a ciascuna immagine. Il nostro DataBase è fondamentale soprattutto per quanto riguarda la figura dell'amministratore. Esso può agire direttamente sul database; ciò implica che ha un pieno controllo su tutti i dati relativi alla Biblioteca Digitale. In questo modo, l'amministratore può, a suo piacimento, modificare i dati contenuti nelle tabelle del database, permettendo così di modificare i privilegi di un utente: può promuovere un utente (che si è registrato) a trascrittore, permettendo così allo stesso utente di poter scrivere una trascrizione oppure può assegnare ad un utente (sempre registrato) di ottenere i privilegi che permettono il download delle opere.

Modello E-R :



Modello Relazionale :

Utente:

(idutente, nome, cognome, tipo, età, sesso, download, username, password, livello, email, professione, richiestad, richiestat, esplorative, titolo, anniesp)

Opera: (idopera, titolo, anno, autore, storia, file, categoria)

Immagine:

(idimmagine, descrizione, imm, pubblicata, opera, trascrizione)

Trascrizione: (idtrascrizione, testo, publicta, validata, modificata)

Asseganto: (idasseganto,data,utente,immagine)

Modifica:

(idmodifica,nuovo_testo,data,buonfine,utente,trascrizione)

Leggenda:

- Gli attributi sottolineati sono le chiavi che identificano in modo univoco ciascuna istanza di quell'entità.
- Gli attributi sottolineati, in modo tratteggiato, sono le chiavi esterne che ci permettono di realizzare le relazioni tra le varie entità.

Vincoli Di Integrità Referenziale :

- 1) Dall'attributo "opera" dell'entità Immagine verso la chiave "idopera" dell'entità Opera
- 2) Dall'attributo "trascrizione" dell'entità Immagine verso la chiave "idtrascrizione" dell'entità Trascrizione
- 3) Dall'attributo "profilo" dell'entità Utente verso la chiave "idprofilo" dell'entità Profilo
- 4) Dall'attributo "utente" dell'entità Assegnato verso la chiave "idutente" dell'entità Utente
- 5) Dall'attributo "immagine" dell'entità Assegnato verso la chiave "idimmagine" dell'entità Immagine

- 6) Dall'attributo "utente" dell'entità Modifica verso la chiave "idutente" dell'entità Utente
- 7) Dall'attributo "trascrizione" dell'entità Modifica verso la chiave "idtrascrizione" dell'entità Trascrizione