

**PROGETTO DIGITAL LIBRARY**  
**CORSO DI LAUREA IN INFORMATICA**  
**UNIVERSITA' DEGLI STUDI DELL'AQUILA**  
**A.A. 2015/2016**

STUDENTE	MATRICOLA
Gabriele Di Rocco	<b>195037</b>
Valerio Piccioni	<b>223000</b>

## 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*

## REQUISITI

Il sistema Digital Library è finalizzato alla conservazione di testi in formato digitale.

Ogni manoscritto preso in analisi costituirà un artefatto chiamato Opera.

L'opera sarà costituita dall'insieme di una raccolta di immagini raffiguranti le pagine, realizzata dagli **acquisitori**.

In aggiunta ad ogni raccolta potrà essere associata una trascrizione del manoscritto in formato TEI. Tale compito è delegato ai **trascrittori**.

Dunque il sistema sarà predisposto ad accogliere dati testuali relativi alla trascrizione dei manoscritti in tale formato, inseriti tramite text editor esterno.

Saranno messi a disposizione dei supervisor strumenti per il controllo della corretta acquisizione delle immagini e per l'associazione tra queste e l'opera relativa, con annessi metadati. Tali supervisor sono i **revisori delle acquisizioni**.

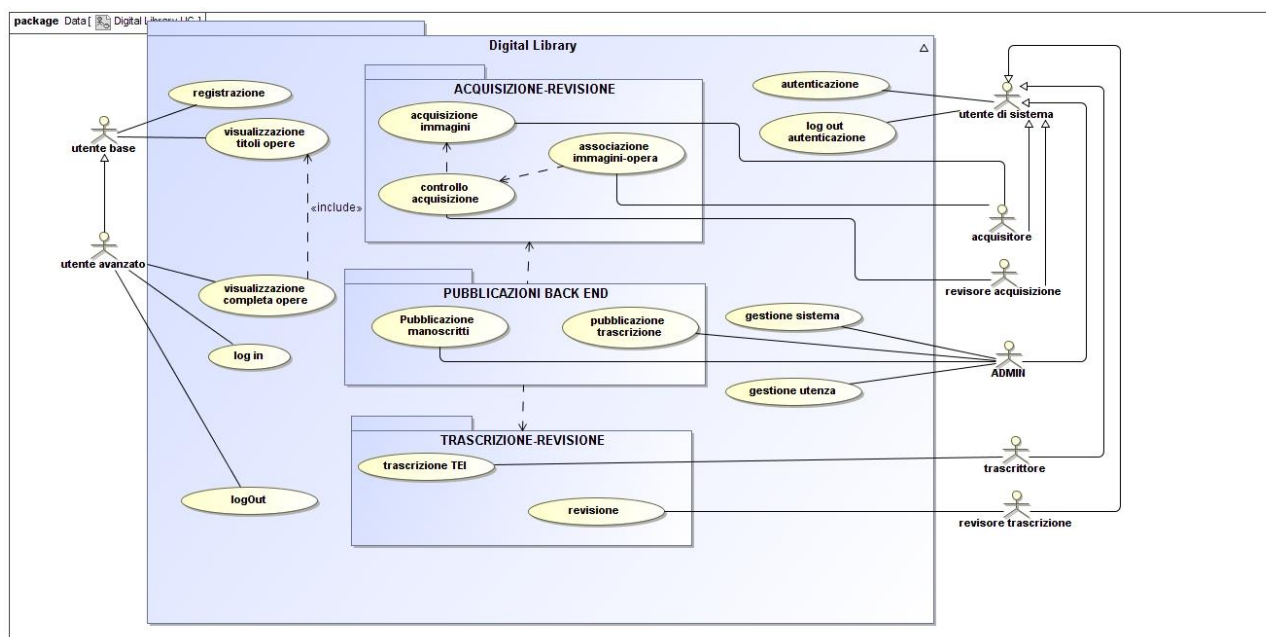
I **revisori delle trascrizioni** avranno la possibilità di avere accesso a tali trascrizioni, validarle e associarle alle immagini corrispondenti già presenti nel sistema.

L'**utente base** avrà accesso parziale alle opere, ossia al solo elenco dei titoli, mentre l'**utente avanzato**, registrato opportunamente nel sistema, avrà accesso alla totalità dell'opera da lui ricercata. Si è deciso di considerare tale suddivisione dando priorità alla possibilità, da parte di un potenziale visitatore del sistema

All'**amministratore** del sistema è permesso di pubblicare le opere, precedentemente revisionate, siano esse costituite dalle sole immagini o arricchite con il relativo file di testo. Inoltre potrà gestire opportunamente gli utenti e i dipendenti del sistema (trascrittori, revisori, acquisitori).

## USE CASE DIAGRAM

Il seguente diagramma mostra in maniera concisa l'insieme delle funzionalità del sistema:



Riassumendo, il sistema mette a disposizione le seguenti attività:

- acquisizione immagini
- revisione immagini
- associazione immagini-opere
- trascrizione manoscritti
- revisione trascrizioni
- pubblicazione manoscritti
- pubblicazione trascrizione
- visualizzazione parziale e totale delle opere
- gestione dell'utenza
- operazioni di registrazione e di login

Gli attori deputati a svolgere le attività di cui sopra sono:

Gli utenti che operano sul sistema:

- Admin
- Acquisitore
- Trascrittore
- Revisore trascrizioni
- Revisore acquisizioni

Gli utenti che utilizzano i servizi:

- Utente base
- Utente avanzato

La registrazione è un'attività a disposizione esclusivamente dell'utente base, mentre la login è esclusiva dell'utente avanzato.

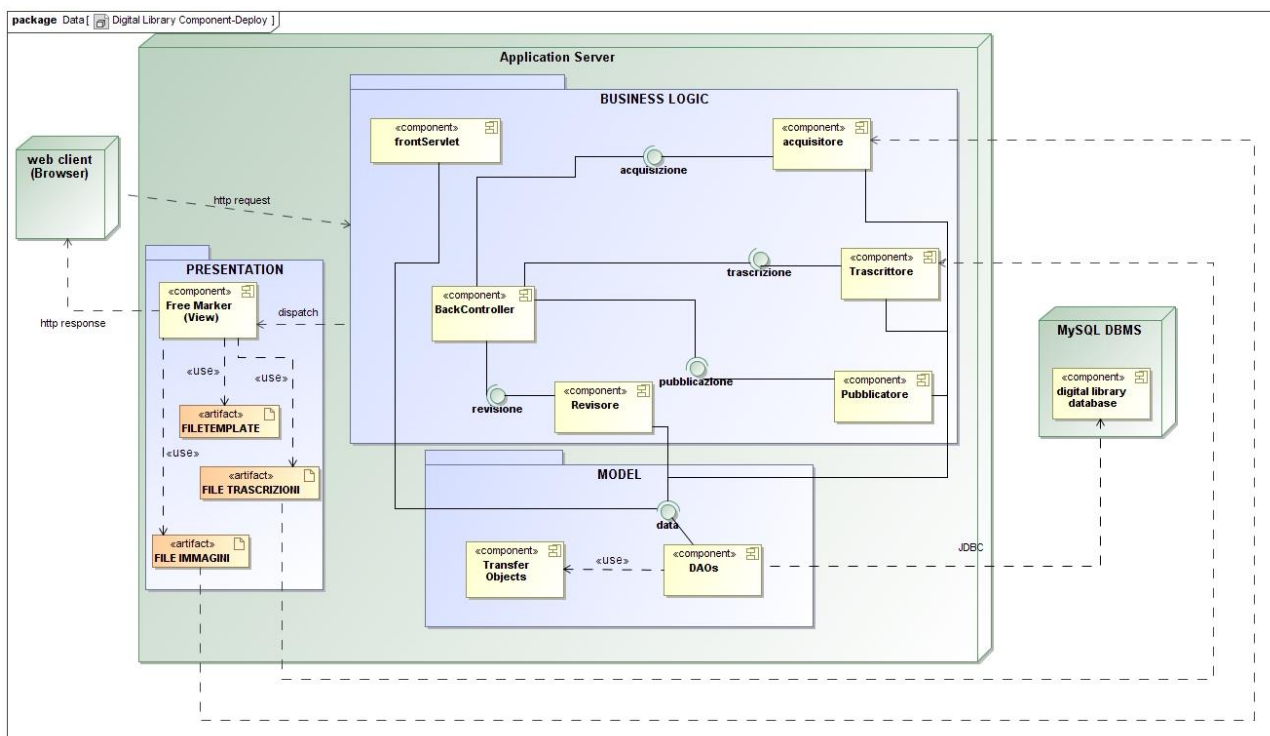
**L'utente di sistema** (amministratori, supervisor ecc..) effettua l'autenticazione, una procedura differente dalla login, dando accesso alle opportune funzionalità messe a disposizione.

Gli use case sono suddivisi in package in base ai differenti processi ai quali appartengono.

La pubblicazione di un'opera dipende ovviamente dalla presenza di manoscritti revisionati, pronti ad essere inseriti nel sistema, dunque il package relativo risulta essere dipendente dai package digitalizzazione e trascrizione.

## SYSTEM DESIGN

Il sistema è stato opportunamente decomposto secondo il principio della separation of concerns, al fine di individuarne le componenti autonome. Si è scelto di optare per un pattern MVC, essendo il più utilizzato e indicato per applicazioni di tale fattura, ove è opportuno separare la logica di presentazione dei dati dalla logica di business.



Il diagramma mostra le componenti del sistema, suddivise nei nodi **web client**, **application server** e **dbms**.

L'**application server**, secondo pattern MVC, si suddivide nei package **Presentation**, **Business Logic** e **Model**.

Il web client effettua le **http request** alla **front Servlet** e al **back controller**.

La View utilizza i file di template, i file di trascrizione e le immagini, il tutto memorizzato nel sistema. Essa restituirà i dati richiesti dal web client.

Il **back controller** si occupa di richiedere i servizi di tutta la business logic, ossia delle componenti **acquisitore**, **trascrittore**, **revisore** e **pubblicatore**, a seconda della richiesta ricevuta precedentemente, restituendo alla View i dati così elaborati.

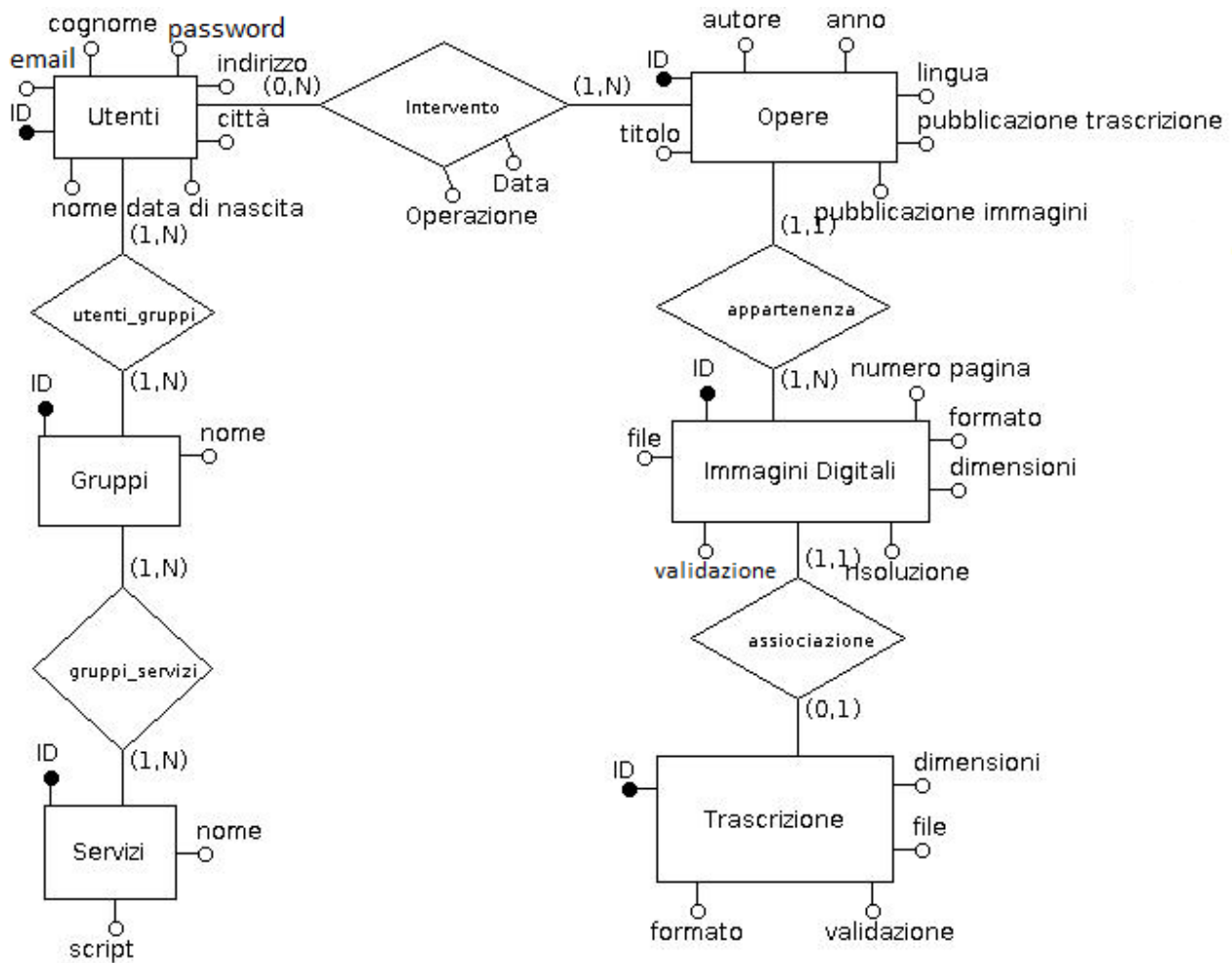
Le componenti di business logic citate poc'anzi si appoggiano al modello del sistema, costituito dai **transfer object**, che realizzano l'astrazione necessaria al trattamento dei dati.

Le informazioni sono incapsulate tramite oggetti **DAO**, in modo da garantire l'opportuno scambio di dati tramite la connessione a database, ove risiedono i dati non volatili della Library.

Una volta ottenuto il dato, la **business logic** restituisce tali informazioni alla View.

## DIGITAL LIBRARY DATABASE

A seguito dell'analisi dei requisiti e della stesura degli use case relativi, il database di sistema è stato organizzato come segue:



Prendiamo in esame le entità:

- Utenti
- Gruppi
- Servizi
- Opere
- Immagini Digitali
- Trascrizioni

Gli **Utenti**, rispettivamente di sistema e non, sono descritti dai seguenti attributi:

- ID
- Nome
- Cognome

- Email
- Password
- Data di nascita
- Indirizzo
- Città

Ogni utente ha associato un gruppo di appartenenza, e tale associazione viene espressa dalla relazione **Utenti\_Gruppi**.

Le cardinalità (1, N) sono dovute alla possibilità di un gruppo di contenere molteplici utenti, mentre un utente di sistema può loggarsi o autenticarsi sia come utente avanzato sia con le credenziali relative al ruolo assegnato all'interno del sistema (revisore, acquirente, ecc.). La decisione di mantenere un unico profilo utente per ruoli diversi, nel caso di utente di sistema, è stata presa guardando alla possibilità di verificare la corretta visualizzazione del contenuto della Digital Library in modalità front office.

Ogni **gruppo** è caratterizzato da:

- ID
- Nome

ed ha associato un servizio caratteristico, per cui la relazione **gruppi\_servizi**, di cardinalità (1, N), esprime questa corrispondenza in quanto un gruppo può avere associato più servizi, mentre un servizio può essere condiviso da più gruppi (pensare all'autenticazione).

I **Servizi** hanno i seguenti attributi:

- ID
- Nome
- Script associato

Gli utenti di sistema sono correlati alle opere sulle quali hanno lavorato, e tale correlazione è espressa tramite la relazione **Intervento**.

Questa è caratterizzata dagli attributi:

- Operazione
- Data

Con questi attributi si esprime il contributo dato da un utente di sistema alle opere, (ovverosia revisionare, pubblicare, trascrivere ecc.).

Le **Opere** presentano i seguenti attributi:

- ID
- Autore
- Anno
- Titolo
- Lingua
- Pubblicazione trascrizione
- Pubblicazione immagini

Questi ultimi due attributi evidenziano l'associazione tra un'opera e le immagini correlate e le trascrizioni, qualora ve ne fossero (ricordiamo che un'opera pubblicata può essere costituita dalle sole immagini scannerizzate).

La relazione **Appartenenza**, che intercorre tra Opere e Immagini Digitali, denota il fatto che ad un'opera possano essere associate l'insieme di immagini che costituiscono il manoscritto originale, ma soprattutto il fatto che un'immagine è parte di una singola opera.

Analizzando nel dettaglio le Immagini Digitali, notiamo i seguenti attributi:

- ID
- File (path)
- Validazione
- Risoluzione
- Dimensioni
- Formato
- Numero pagina

Anzitutto qualsiasi attributo File va ad indicare la presenza nel DB non dell'oggetto BLOB, ma della semplice stringa atta a completare il path del file, nella View.

L'attributo validazione esprime la possibilità di pubblicare o meno una determinata opera ritenuta pronta ad essere messa a disposizione degli utenti.

Ad ogni Immagine può essere associato un file contenente la trascrizione della pagina corrispondente del manoscritto originale, in formato TEI. Espressa dalla relazione **Associazione**, si denota la possibilità di un'immagine di non avere nessuna trascrizione associata o al massimo una, tramite la cardinalità (0, 1).

L'entità **Trascrizione** è rappresentata da questi attributi:

- ID
- Formato
- Dimensioni
- File (path)
- Validazione

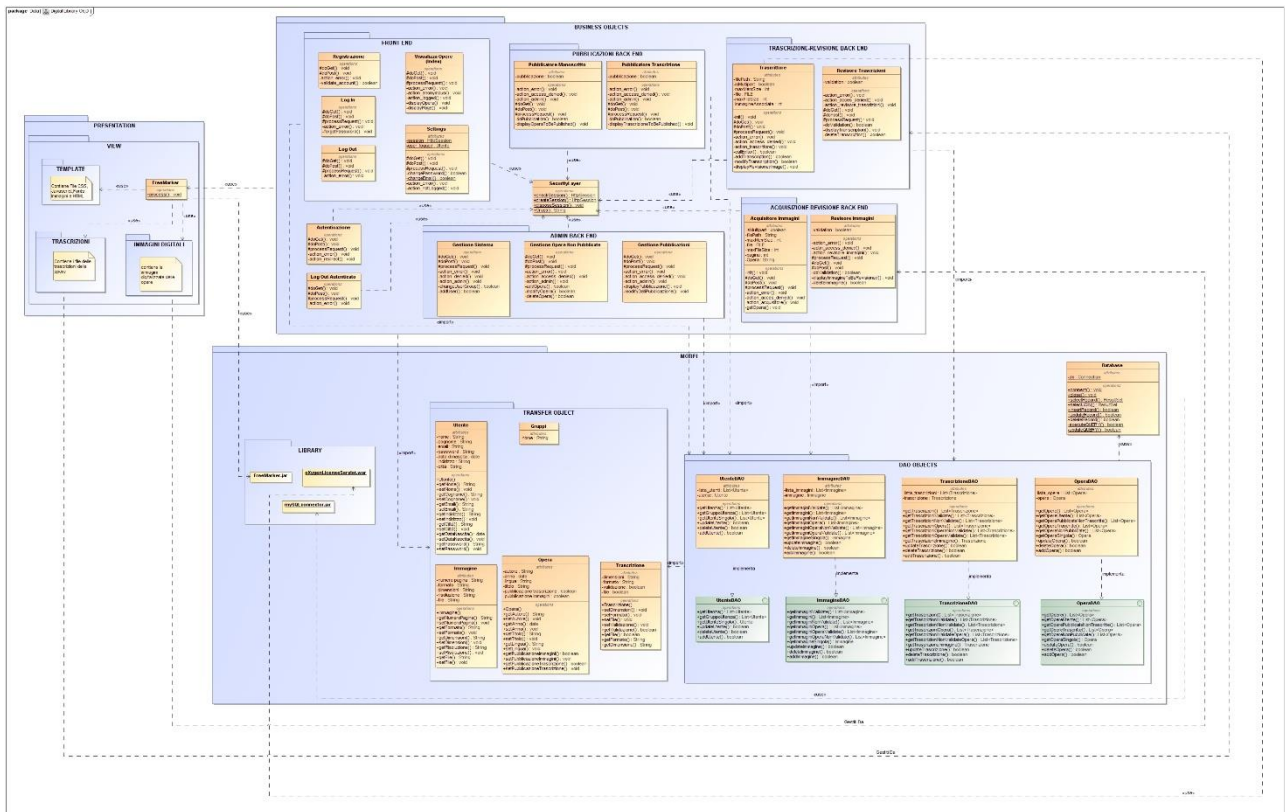
i cui significati sono analoghi agli attributi dell'entità Immagini.



## SOFTWARE OBJECT DESIGN

Dal component-deployment diagram è possibile derivare l'Object Diagram, contenente classi, interfacce e i relativi membri e metodi.

Ovviamente la struttura di base dovrà rispecchiarne le caratteristiche, dunque si ripropone la suddivisione tra i package Presentation, Business Logic e Model, sempre secondo il pattern MVC.



Ogni package è suddiviso in ulteriori sotto-package, così costituiti:

### Presentation

- View, contenente Template, Trascrizioni e Immagini Digitali

### Business logic

- Front end
- Pubblicazioni Back end
- Admin Back end
- Trascrizione-Revisione back end
- Acquisizione-Revisione back end

### Model

- Library
- Transfer Object
- DAO Objects

### *Package Business Logic*

Innanzitutto occorre specificare che tutta la Business Logic si appoggia sul motore di Template Free Marker. Inoltre essa fa uso della classe **security layer**, che si occupa della gestione di sessione.

Il package **front end** contiene le classi:

- Registrazione
- Visualizza Opere,
- Log In
- Log Out
- Settings

Ovviamente gestiscono tutti i metodi relativi al recupero e alla validazione dei dati utente e sessione.

Il package **admin back end** contiene le classi:

- Gestione utenza
- Gestione opere non pubblicate
- Gestione pubblicazioni

Esse implementano i metodi riguardanti il recupero e la modifica dei dati riguardanti l'utenza e la gestione delle opere.

Il package **pubblicazioni back end** contiene le classi:

- Pubblicazione manoscritto
- Pubblicazione trascrizione

I membri di entrambe le classi verificano se l'opera ha associata immagini e/o manoscritti, oltre ai metodi opportuni, e ne effettuano la pubblicazione.

Il package **trascrizione-revisione back end** contiene le classi:

- Trascrittore
- Revisore trascrizioni

In esse sono espresse le caratteristiche delle trascrizioni, come il file, il path, la dimensione, lo stato di validazione, e i metodi di aggiunta, modifica, o eliminazione.

Il package **acquisizione-revisione back end** contiene le classi:

- Acquisitore immagini
- Revisore immagini

Il cui funzionamento è analogo alle classi del trascrittore-revisore, appena descritte.

### *Package Model*

Il Model contiene i seguenti sotto-Package:

**Library**, contenente l'eseguibile di Free Marker, il connettore a database e l'archivio web Oxygen.war.

**Trasfer Object** con all'interno le classi:

- Utente
- Opera
- Immagine
- Trascrizione
- Gruppi

Esse rappresentano la principale implementazione dei dati di interesse, i cui membri e metodi rispecchiano l'organizzazione presente del database del sistema, precedentemente trattato nella documentazione.

**DAO Objects** contenente le classi:

- UtenteDao
- ImmagineDAO
- TrascrizioneDAO
- OperaDAO

E le relative interfacce che implementano:

- UtenteDao
- ImmagineDAO
- TrascrizioneDAO
- OperaDAO

Tutto il package si appoggia alla classe database, deputata alle CRUDE operations. Utilizza i driver JDBC, e dunque nello specifico la libreria MySQLconnector.jar. Inoltre importa l'intero package Transfer Object, del quale si serve per implementare l'incapsulamento dei dati.