****

**INDICE**

**Project-Oriented-Software-Design 2017/2018**

**1. REQUIREMENTS**

Il sistema offre una biblioteca digitale di testi e studi che contribuiscono alla formazione della cultura all’interno dell’Università degli Studi dell’Aquila. Lo scopo del progetto è di consentire la consultazione di manoscritti che devono essere digitalizzati. Gli utenti possono consultare il catalogo digitale dei manoscritti e visualizzare un’opera di loro interesse a seguito di una ricerca. Gli utenti con privilegi possono scaricare le opere. Ogni utente può richiedere di diventare trascrittore, che può aggiungere il testo in formato digitale di una immagine.

* 1. **DOCUMENTO DEI REQUISITI:**
     1. **Viewer**
* Consultazione opere

Gli Utenti possono visualizzare le opere presenti nel catalogo.

* Ricerca nel catalogo

Gli Utenti possono cercare le opere nel catalogo per: titolo, autore, contenuto del testo.

* Download dell’opera

Utenti con Privilegi possono scaricare le opere.

* Richiesta per essere collaboratore del sistema (Trascrittore)

Un Utente può far richiesta per diventare Trascrittore.

* Accesso dati personali

Gli Utenti possono visualizzare il proprio profilo contente i propri dati.

* + 1. **Uploader**
* Upload opera nel sistema (immagini)

Gli Uploader possono caricare immagini nel catalogo dopo che sono state revisionate da un Revisore Upload.

* Caricamento metadati opera

Gli Uploader definiscono i dati della/e pagina/e caricata/e.

* + 1. **Transcriber**
* Trasformazione opera in un testo digitale (formato TEI)

I trascrittori trascrivono le opere in formato cartaceo sul catalogo attraverso un editor di testo (TEI) presente nel programma.

* + 1. **Manager**
* Gestione assegnazioni

I revisori delle trascrizioni assegnano delle parti di opera a uno o più trascrittori.

* Riassegnazione pagine

Le pagine assegnate ai Trascrittori possono essere riassegnate.

* Revisione delle trascrizioni

I Revisori delle trascrizioni controllano le trascrizioni e a seguito le validano se adatte alla pubblicazione.

* Gestione livelli di esperienza

I Revisori delle trascrizioni gestiscono i livelli di esperienza dei trascrittori. Il livello va da 1 a 5 e viene scelto in base alla sua esperienza.

* Revisiona acquisizione immagini

I Revisori degli Upload validano le immagini caricate prima di aggiungerle al catalogo.

* + 1. **Administrator**
* Gestione del sistema

L’Amministratore gestisce tutto il sistema.

* 1. **MODELLI UML USE CASE**

***Sono stati identificati i seguenti attori del sistema:***

* **Amministratore**

Gestione del sistema

* **Utente base**

Consultazione opere (visualizzazione e ricerca)

* **Utente privilegiato**

Utente base con possibilità di download dell’opera

* **Trascrittore**

Utente che può trascrivere opere nell’editor (TEI)

* **Revisori Trascrizioni**

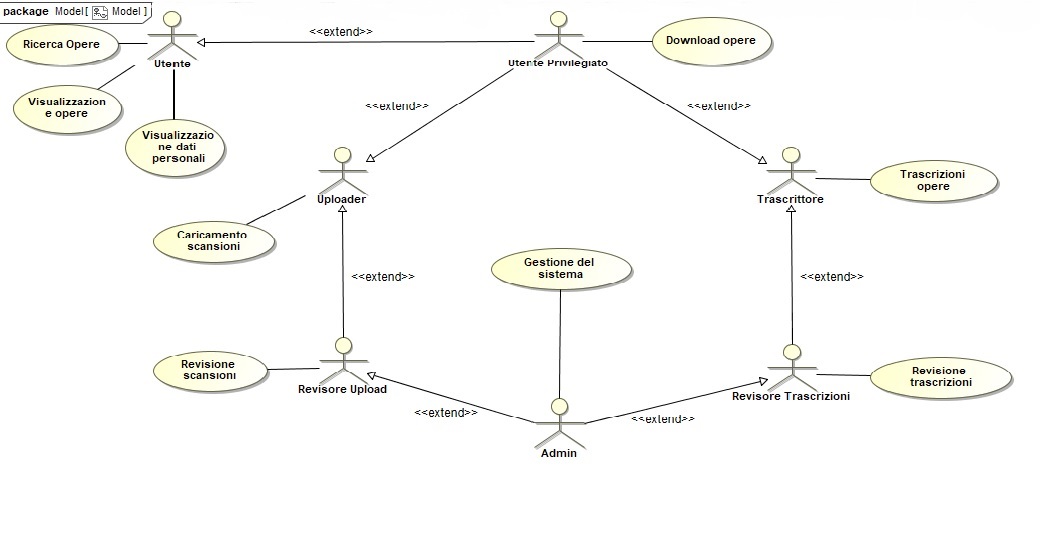
Revisione e validazione della trascrizione

* **Uploader**

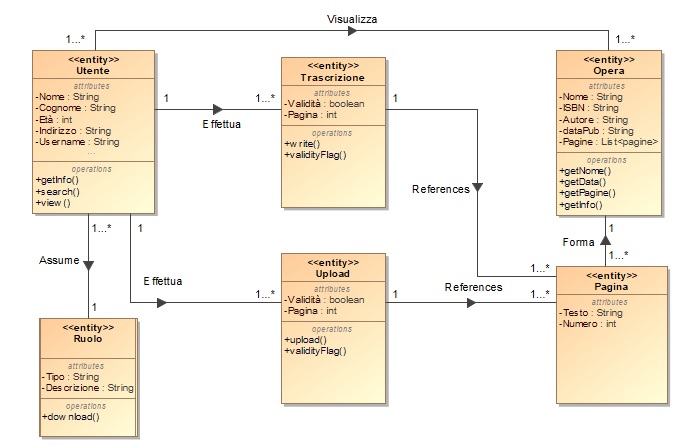
Caricamento immagini

* **Revisori Upload**

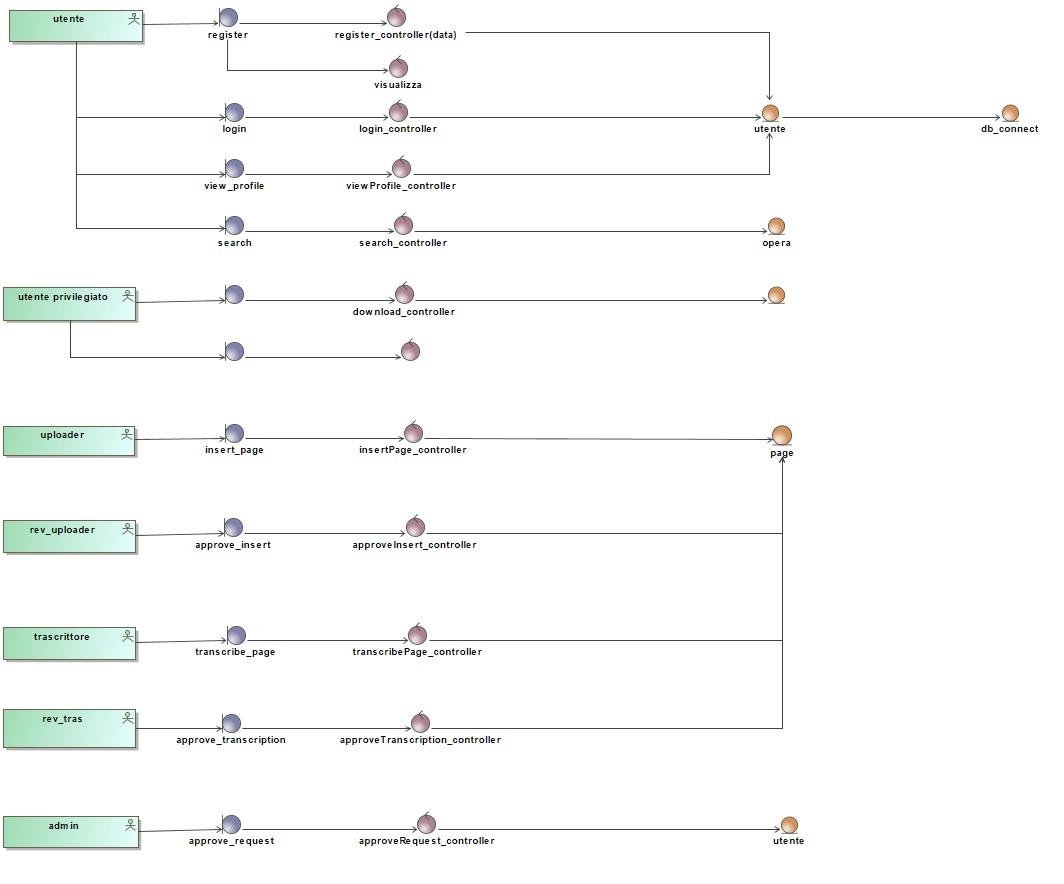
Revisione immagini caricate

******

* 1. **MODELLI DI DOMINIO**

****

* 1. **ANALISI FINALIZZATA ALL’INDIVIDUAZIONE DELLE CLASSI ENTITY, BOUNDARY E CONTROLLER**



**2. SYSTEM DESIGN**

**2.1 Modello dell’architettura software**

**MVC:**

Il modello architetturale scelto è stato il pattern **MVC**, il quale utilizza un design pattern che suddivide il sistema in più moduli indipendenti e renderà più facili le modifiche e gli aggiornamenti.

Abbiamo usato il pattern **MVC** perché:

* È più facile riutilizzare il codice, quindi lo sviluppo è più veloce
* Il codice è più organizzato, quindi è più facile da capire e mantenere
* È più facile testare il codice.
* è più sicuro

Il pattern architetturale, è diviso in tre parti:

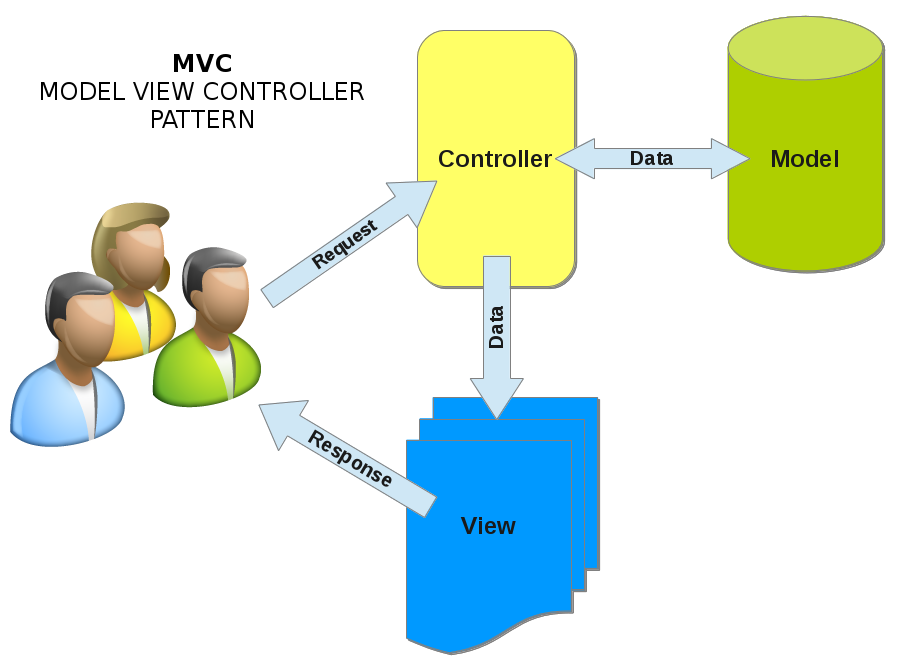
1. **Model**

* Incapsula lo stato dell’applicazione
* Deve permettere di accedere ai dati
* Deve notificare i cambiamenti di stato

1. **View**

* Mostra il modello
* Gestisce l’interazione con l’utente

1. **Controller**

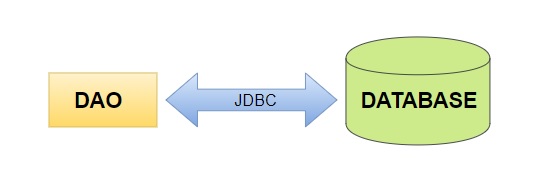
* Rappresenta la logica applicativa
* Collega le azioni dell’utente con modifiche dello stato
* Sceglie cosa essere mostrato

**2.2 DESCRIZIONE DELL’ARCHITETTURA**

**MODEL**

Nel package model sono inoltre stati utilizzati altri due design pattern per la gestione dei dati:

Il **DAO** (Data Access Object) servirà per collegare un Database al sistema che gestisce tutti i dati degli utenti e delle opere nel sistema. Il DAO incapsula tutti gli accessi ai dati conservandoli in un posto unico (DB). Per gestire al meglio i dati da inserire nel DB è stato utilizzato il **VO** (Value Object) che incapsula i **business data** (il modello business coopera con il database con una relazione associativa) per migliorare la loro gestione limitando il passaggio di oggetti pesanti attraverso il sistema. I VO sono rappresentazioni “leggere” delle classi di dominio dell’analisi del modello.



**2.2 Descrizione dell’architettura**

**View:**

Sarà composta da una schermata iniziale di login in cui verranno inseriti i dati di accesso e a seconda del suo ruolo avrà a disposizione delle operazioni. L’utente base ha a disposizione una barra di ricerca dove potrà inserire i dati per cercare l’opera nel catalogo e visualizzare quindi le opere. Gli altri utenti in aggiunta avranno a disposizione delle funzionalità aggiuntive come il download e la modifica di alcune parti dell’opera a seconda del proprio ruolo.

**Model:**

**Control:**

In base ai dati di accesso dell’utente garantirà l’uso di determinate funzioni che potrà utilizzare. Gestisce le operazioni effettuate da GUI (Graphic User Interface) attraverso varie funzioni corrispondenti a ogni parte dell’interfaccia. Tramite questo modulo l’amministratore può modificare il ruolo di ogni utente,