

# **Object Oriented Software Design**

## **Progetto 2017-2018**

Fabio Capitanio

Mat. 236632

[fabio.capitanio@student.univaq.it](mailto:fabio.capitanio@student.univaq.it)

Marco Zucconi

Mat. 236261

[marco.zucconi@student.univaq.it](mailto:marco.zucconi@student.univaq.it)

# **Indice**

## **1. Requisiti**

- 1.1 Documento dei requisiti
  - 1.1.1 Requisiti funzionali
  - 1.1.2 Requisiti non funzionali
- 1.2 Modello use case
- 1.3 Classi entity-complete
- 1.4 Classi entity, boundary e controller

## **2. System Design**

- 2.1 Modello dell'architettura software
- 2.2 Descrizione dell'architettura
- 2.3 Descrizione delle scelte

## **3. Software/Object Design**

- 3.1 Modello object design
- 3.2 Descrizione dei design scelti

## **1.1 Requisiti**

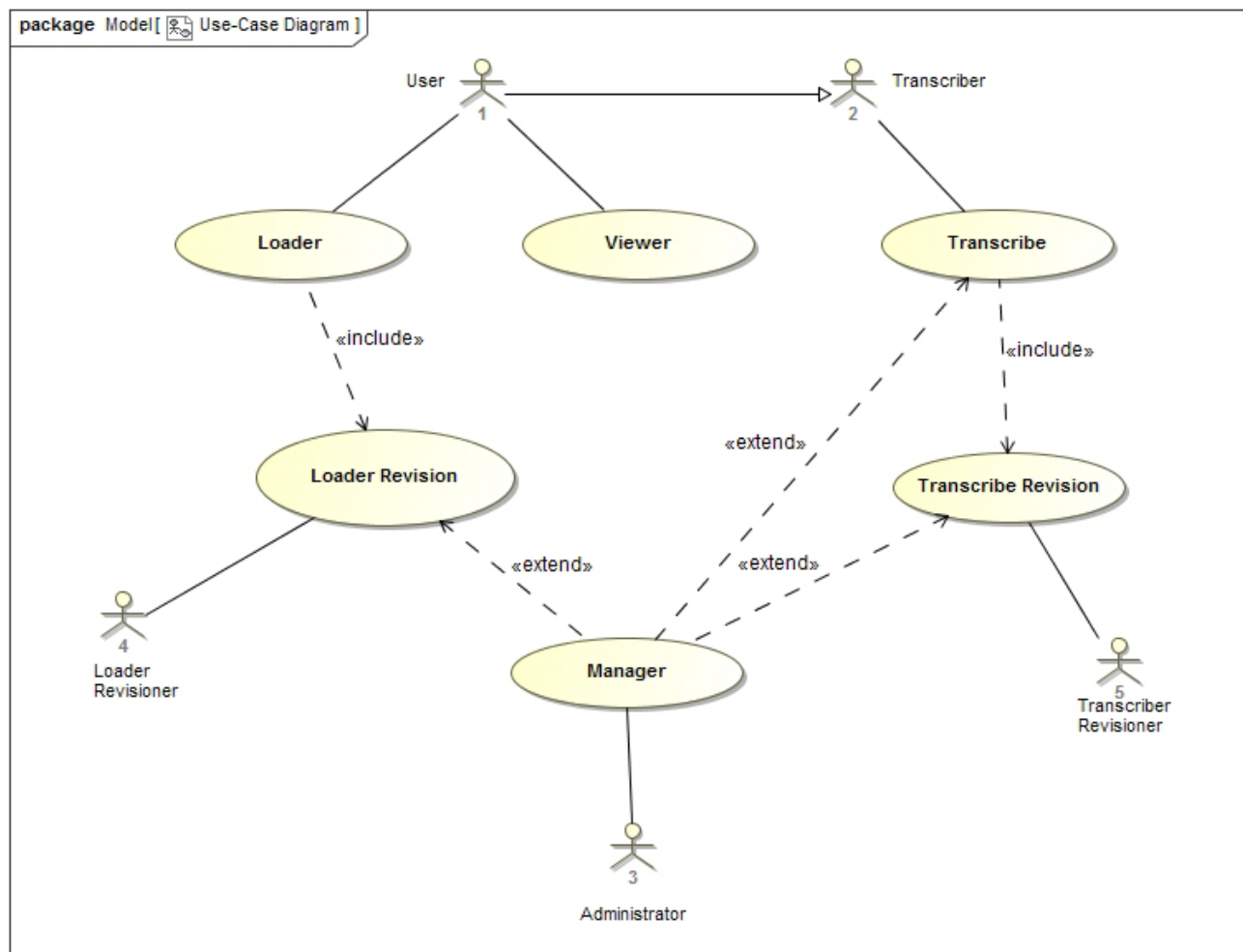
### **1.1.1 Requisiti Funzionali**

- Login/Registrazione: Funzionalità che permette agli utenti di accedere al sistema.
- Visualizzazione opere: Funzionalità che permette agli utenti di visualizzare l'immagine della pagina dell'opera e se presente anche la sua trascrizione.
- Ricerca libro/parole nel testo: Funzione che permette la ricerca di opere all'interno del catalogo per metadati o all'interno del testo della trascrizione.
- Download Opere: Funzione che permette ad alcuni utenti con particolari privilegi di effettuare il download delle opere.
- Visualizzazione Profilo Personale: Funzione che permette agli utenti di visualizzare i dettagli del proprio profilo.
- Promozione: Funzione che permette ai soli admin di promuovere un utente a transcriber
- Correzione e Validazione: Funzione che permette ai soli admin di poter revisionare le trascrizioni concluse, e di effettuare correzioni e validazioni.

### **1.1.2 Requisiti non Funzionali**

- Sincronizzazione modifiche: Funzione che permette a più trascrittori di lavorare su una stessa pagina, quindi garantire la sincronizzazione delle modifiche.
- Usability: Il sistema dovrà essere pulito e di facile utilizzo.
- Reliability: il sistema dovrà garantire all'utente le funzioni messe a disposizione senza errori.
- Availability: il sistema dovrà essere disponibile in qualsiasi momento.

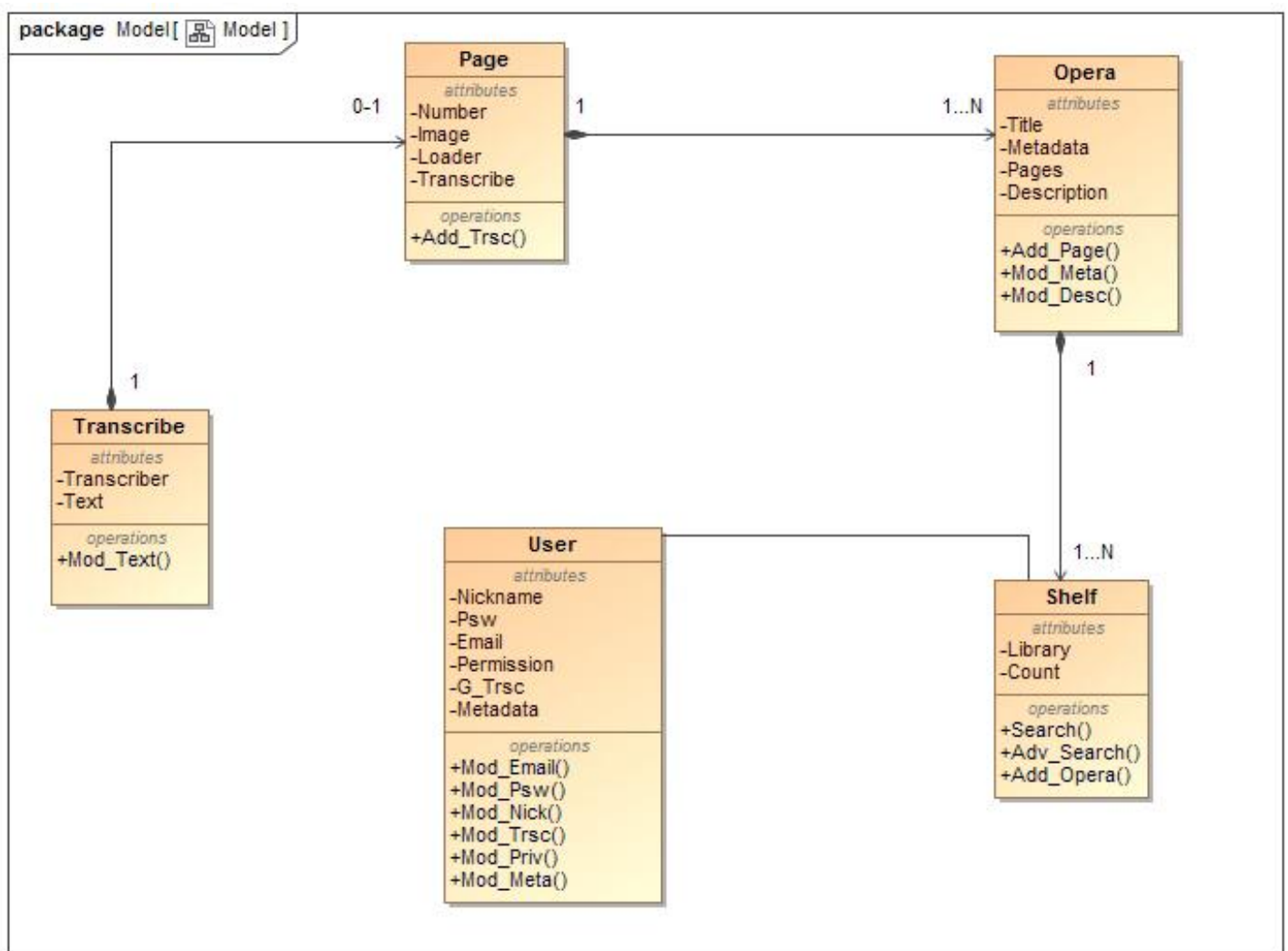
## 1.2 Modello Use-Case



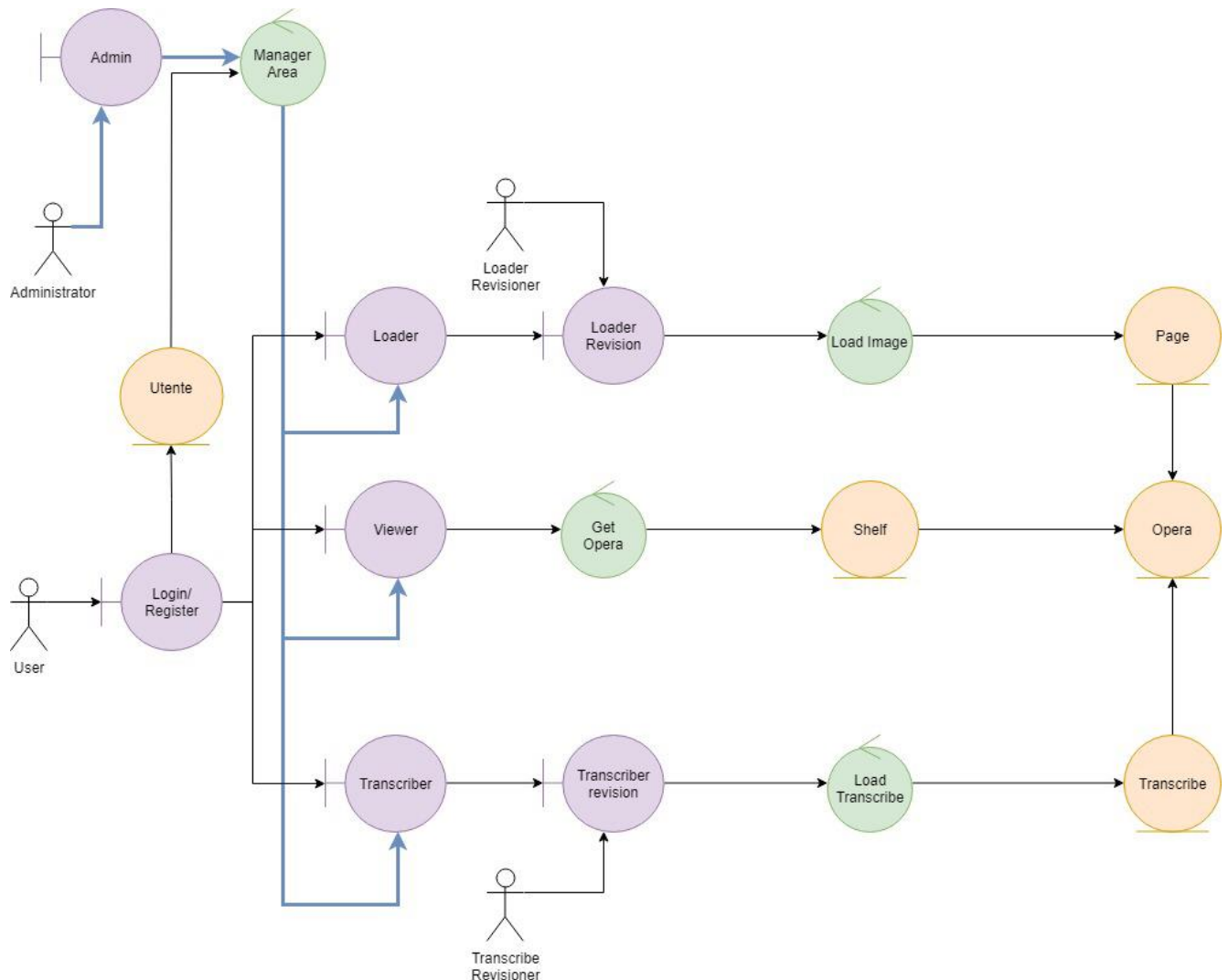
- User: L'utente del sistema, può visualizzare le opere e se ha i permessi eseguire le trascrizioni.
- Transcriber: Utente con permessi speciali, che trascrive pagine di opere in versione digitale.
- Loader/Transcriber Revisioner: Utenti con permessi speciali che effettuano la revisione delle pagine caricate (loader) e delle trascrizioni digitali (transcriber).
- Administrator: Amministratori del sistema, che gestiscono i permessi degli utenti e il funzionamento del sistema.
- Viewer: componente di visualizzazione delle opere all'interno della libreria.

- Loader: Componente di caricamento delle immagini digitali all'interno della libreria.
- Transcribe: Componente di trascrizione delle immagini digitali in formato testo da parte dei transcriber.
- Loader revision: Componente di controllo per il caricamento delle immagini, utilizzato dai loader revisioner.
- Transcribe revision: Componente di controllo per la trascrizione delle immagini, utilizzato dai transcriber revisioner.
- Manger: Componente di controllo per la gestione degli utenti, della libreria e dei permessi.

### 1.3 Classi Entity-Complete



## 1.4 Boundary Entity Controller



### Boundary Object

- Login/Register: Tramite questa interfaccia gli utenti potranno registrarsi nella piattaforma e accedere tramite il login ai servizi offerti.
- Viewer: È l'interfaccia che permette agli utenti di visualizzare le opere presenti all'interno della libreria.
- Loader: È l'interfaccia che permette agli utenti di caricare nuove opere e farle approvare per la pubblicazione all'interno della libreria.

- Loader Revision: È l'interfaccia attraverso la quale i loader revisioner controllano le opere caricate dagli utenti e ne permettono la pubblicazione.
- Transcriber: È l'interfaccia che permette all'utente di trascrivere testualmente una pagina di un'opera all'interno della libreria.
- Transcriber Revision: Attraverso questa interfaccia i transcriber revisioner possono esaminare le trascrizioni effettuate sulle opere e decidere di approvarle o rifiutarle.
- Admin: L'interfaccia di gestione del sistema, permette agli amministratori di operare sulle diverse funzionalità offerte, tra cui privilegi degli utenti, i permessi sulle varie opere per i trascrittori e la gestione delle opere presenti.

## **Controller Object**

- Get Opera: ricevuti i filtri di ricerca dall'interfaccia di visualizzazione, va a trovare l'opera interessata all'interno del database e la restituisce all'interfaccia affinché sia visualizzata.
- Load Image: una volta approvata, le varie immagini dell'opera vengono unite e inserite all'interno del database per poter essere utilizzate dagli utenti.
- Load Transcribe: una volta approvata, acquisisce la trascrizione della pagina dell'opera e la collega all'immagine corrispondente nel database.
- Manager area: permette all'interfaccia di Admin di gestire e comunicare con le altre interfacce del sistema.

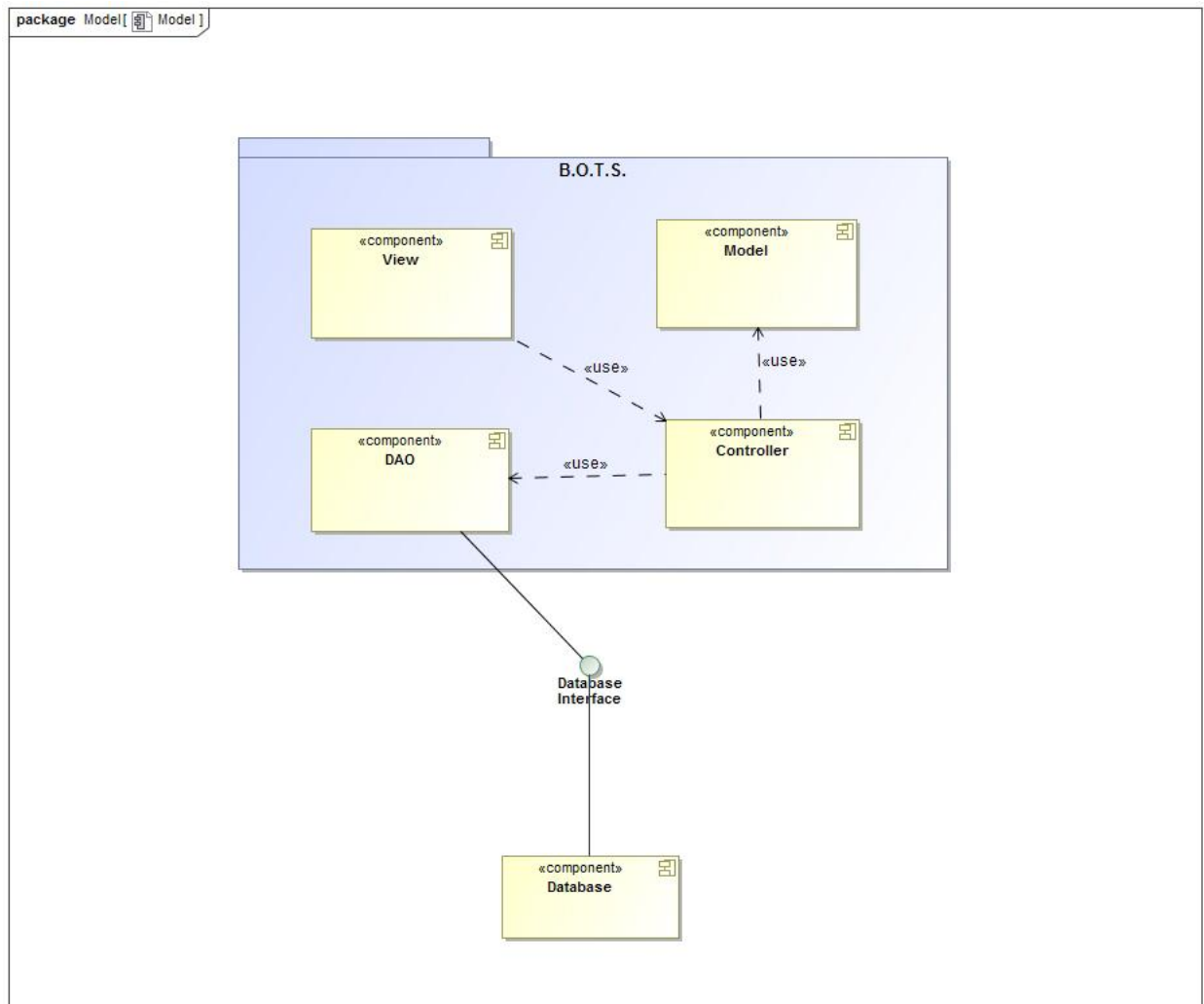
## **Entity Object**

- Users: Contiene tutti i dati sugli utenti, tra cui informazioni personali, privilegi e autorizzazioni.
- Shelf: Contiene l'insieme delle opere all'interno del sistema.
- Opera: Contiene le informazioni relative all'opera, l'autore e le pagine di cui è composta.

- Page: Contiene l'immagine della pagina, il numero sequenziale all'interno dell'opera e l'utente che l'ha caricata.
- Transcribe: Contiene la trascrizione testuale della pagina a cui è associata.

## 2. System Design

### 2.1 Modello dell'architettura software





## **2.2 Descrizione dell'architettura software**

Il nostro software si basa su un database SQL.

L'architettura è divisa in View, Model, Controller e DAO.

La componente View contiene l'insieme delle interfacce utilizzate dal software.

Il componente Model si occupa di contenere le classi utilizzate per tutte le operazioni all'interno del software.

Il componente controller contiene tutte le funzioni necessarie al corretto funzionamento del software.

Il componente DAO permette al controller di interfacciarsi al database.

## **2.3 Descrizione delle scelte**

Sono stati seguiti i principi del paradigma Object-Oriented .

Per gestire le diverse funzione che possono svolgere gli utenti con privilegi speciali(transcribe, loader revisioner e transcribe revisioner) abbiamo optato per una gestione dinamica delle interfacce.

Appena effettuato il login, all'utente verrà mostrata una finestra di scelta in cui potrà visitare le sezione del software in base ai suoi permessi.

Gli amministratori, effettuato il login con delle apposite credenziali, verranno reindirizzati nell'interfaccia di gestione del software, attraverso la quale potranno amministrare l'intero sistema.

### 3. Software/Object Design

