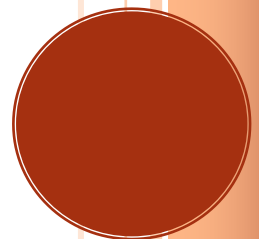


Biblioteca Digitale

Sara Di Berardino 247976
Davide Simboli 246976
Alessio Morgante 246991



Documento dei requisiti

Obiettivi

Sviluppare una biblioteca digitale per consentire la consultazione dei manoscritti che devono essere digitalizzati e che costituiscono un patrimonio bibliografico antico per un totale di 60000 carte, secolo XV-XIX, contenenti memorie storiche della città dell'Aquila.

Requisiti funzionali

1. VIEWER:

- 1.1 Registrazione dell'utente;
- 1.2 Login;
- 1.3 Logout;
- 1.4 Consultazione delle opere digitali a utenti registrati;
- 1.5 Ricerca nel catalogo per metadati oppure all'interno del testo della trascrizione;
- 1.6 Visualizzazione dell'opera che verrà visualizzata con una schermata che avrà sulla destra il testo della trascrizione e sulla sinistra l'immagine della pagina dell'opera che si sta visualizzando;
- 1.7 Possibilità di sfogliare le pagine;
- 1.8 Possibilità per gli utenti privilegiati di effettuare il download dell'opera;
- 1.9 Possibilità per gli utenti di fare richiesta tramite un modulo per essere collaboratore di sistema (trascrittore);
- 1.10 Accesso degli utenti al proprio profilo personale;
- 1.11 Possibilità da parte degli utenti di modificare dati del proprio profilo personale.

2. UPLOADER:

- 2.1 Caricamento delle immagini dell'opera da parte dell'acquisitore;

- 2.2 Caricamento dei metadati dell'opera (titolo, anno, autore, n° pagine totali) da parte dell'acquisitore;
- 2.3 La digitalizzazione viene controllata da supervisori all'acquisizione per assicurarne la correttezza e la qualità;
- 2.4 Pubblicazione dell'opera (solo immagini) da parte del supervisore a seguito del consenso del manager.

3. TRANSCRIBER:

- 3.1 Trascrizione digitale delle opere da parte di trascrittori;
- 3.2 Revisione delle trascrizioni da parte di revisori alle trascrizioni;
- 3.3 Pubblicazione delle trascrizioni da parte dei revisori alle trascrizioni dopo il consenso del manager.
- 3.4 Sincronizzazione delle modifiche.

4. MANAGER:

- 4.1 Assegnazione opere ai trascrittori
- 4.2 Revisione trascrizioni
- 4.3 Trascrizione/correzione trascrizioni
- 4.4 Consenso alla pubblicazione delle trascrizioni e delle opere;
- 4.5 Gestione dei livelli dei trascrittori;
- 4.6 Consenso alla supervisione dell'acquisizione delle immagini;
- 4.7 Accetta/Rifiuta richieste per diventare trascrittore.

5. ADMINISTRATOR:

- 5.1 Eliminazione degli utenti.
- 5.2 Eliminazione delle opere.

Requisiti non funzionali

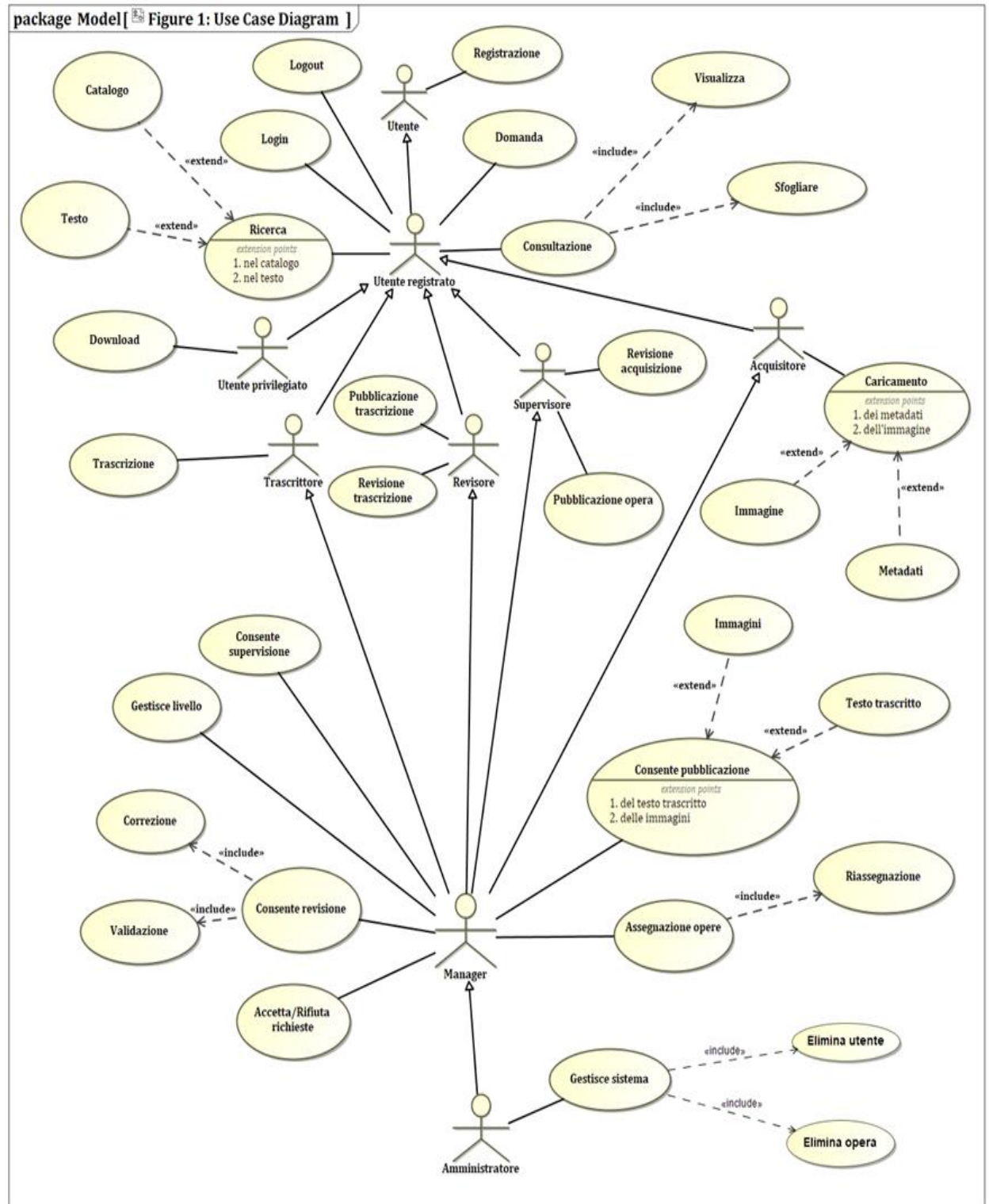
- ✓ **Il software deve prevedere un sistema di protezione per quanto riguarda l'accesso ai ruoli:**
Requisito soddisfatto da un sistema di autenticazione con registrazione e login
- ✓ **Il sistema deve garantire una risposta veloce agli utenti che richiedono la visualizzazione delle opere:**
Requisito soddisfatto dalla velocità del database relazionale di effettuare query di select
- ✓ **Il sistema deve garantire la facilità delle modifiche ed eventuale espansione futura:**
Requisito soddisfatto dall'utilizzo del pattern MVC che permette una facile manutenzione del codice con una conseguente scalabilità
- ✓ **Il sistema deve garantire un'interfaccia user-friendly:**
Requisito soddisfatto dall'utilizzo di JavaFX per realizzare interfacce facili ed intuitive.

Vincoli

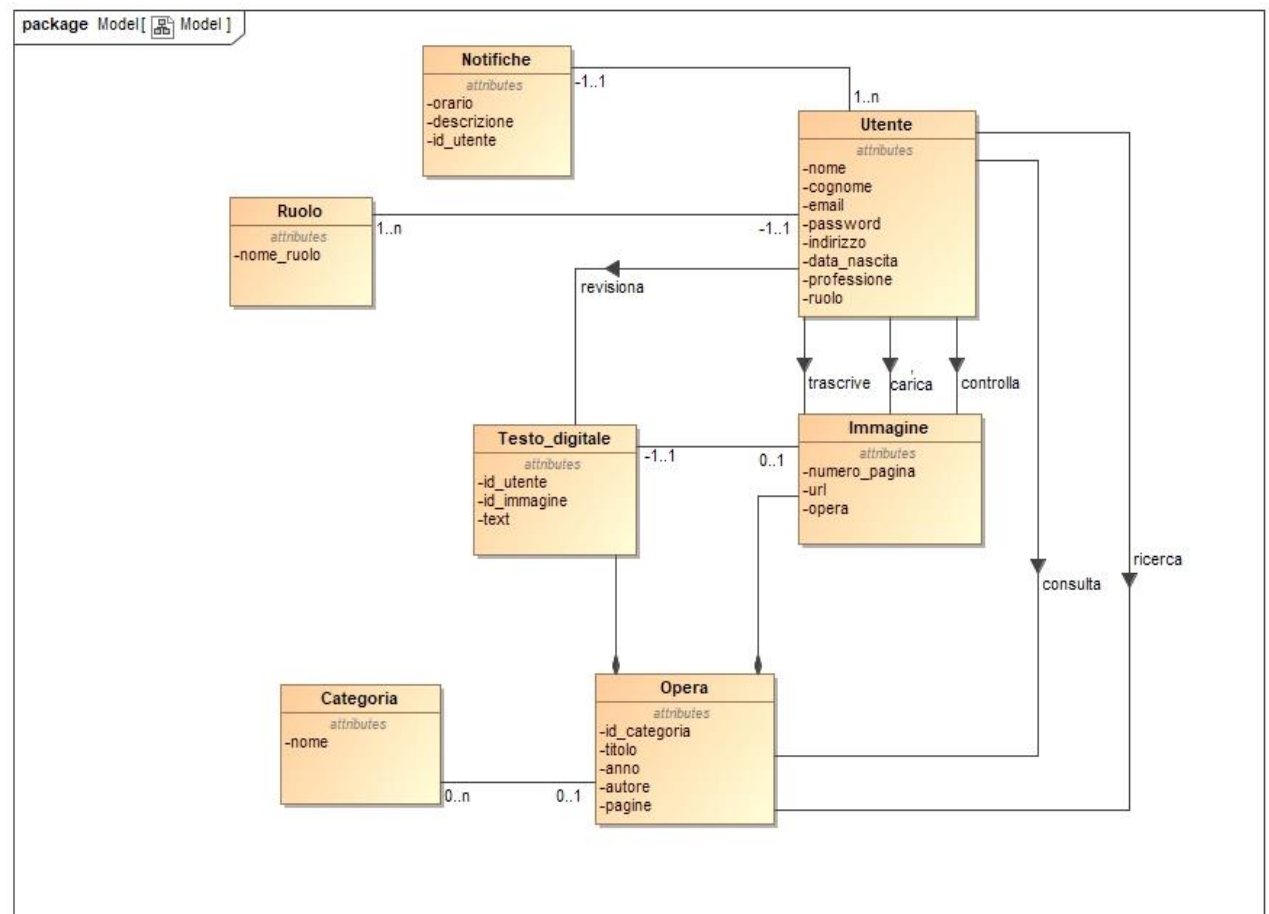
- ✓ Nella registrazione non è possibile usare il ruolo dell'amministratore. Esso è già nel sistema;
- ✓ Nel profilo personale gli utenti possono modificare solo nome, cognome, data di nascita indirizzo, titolo di studio, professione e password;
- ✓ Nel profilo personale vengono visualizzati i dati della registrazione: nome, cognome, data di nascita, email, titolo di studio, professione, indirizzo, password (criptata) e ruolo;
- ✓ Solo l'utente base e l'utente privilegiato possono diventare trascrittori;
- ✓ Se si diventa trascrittori si perde il ruolo precedente;
- ✓ La richiesta per diventare trascrittori può essere mandata una sola volta fino a quando non si riceve una risposta. Se viene accettata si diventa trascrittori, se viene rifiutata può essere mandata di nuovo;
- ✓ Le opere possono essere divise per categorie;
- ✓ L'acquirente può aggiungere le categorie se e solo se non sono già presenti nel database;

- ✓ Ogni opera è formata da più immagini, ognuna delle quali rappresenta una pagina del manoscritto;
- ✓ È possibile sfogliare le pagine dell'opera attraverso un paginatore;
- ✓ In fase di caricamento e di revisione delle immagini si dà la possibilità di avere un riepilogo di esse attraverso una visualizzazione in miniatura e con un paginatore;
- ✓ Si possono acquisire solo 6 pagine alla volta;
- ✓ La pubblicazione delle opere può avvenire solo se tutte le immagini delle stesse sono state acquisite e revisionate;
- ✓ La trascrizione dell'opera in testo digitale avviene attraverso operazioni di trascrizione. Le trascrizioni sono digitate manualmente attraverso un text editor integrato;
- ✓ Più trascrittori possono lavorare sulla stessa pagina;
- ✓ Ogni trascrittore ha un livello 1-5 in base alla propria esperienza;
- ✓ I revisori di trascrizioni possono inserire dei commenti nelle trascrizioni prima di rimandarle ai vari trascrittori;
- ✓ Se disponibile viene visualizzato sulla sinistra il testo della trascrizione e a destra l'immagine dell'opera che si sta visualizzando, altrimenti, se il testo della trascrizione non è disponibile viene visualizzata solo l'immagine dell'opera.
- ✓ Se il manager nega il consenso alla supervisione l'acquirente dovrà acquisire di nuovo quella pagina. Altrimenti, se il consenso è ammesso l'immagine passa al supervisore.
- ✓ Si possono controllare solo 6 immagini acquisite alla volta;
- ✓ Il trascrittore non ha livello quando si iscrive
- ✓ I trascrittori possono trascrivere solo opere che gli vengono assegnate

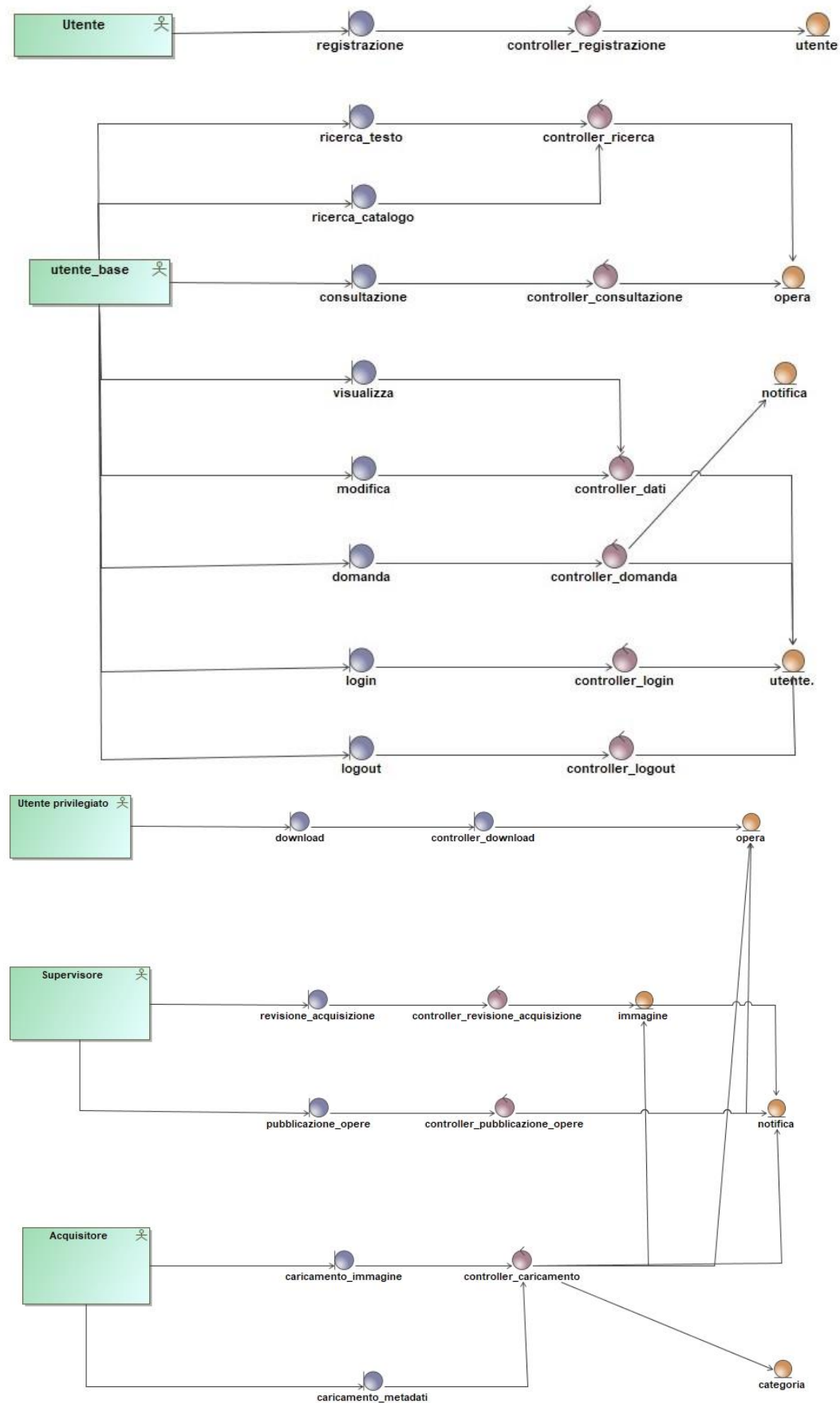
Use case diagram

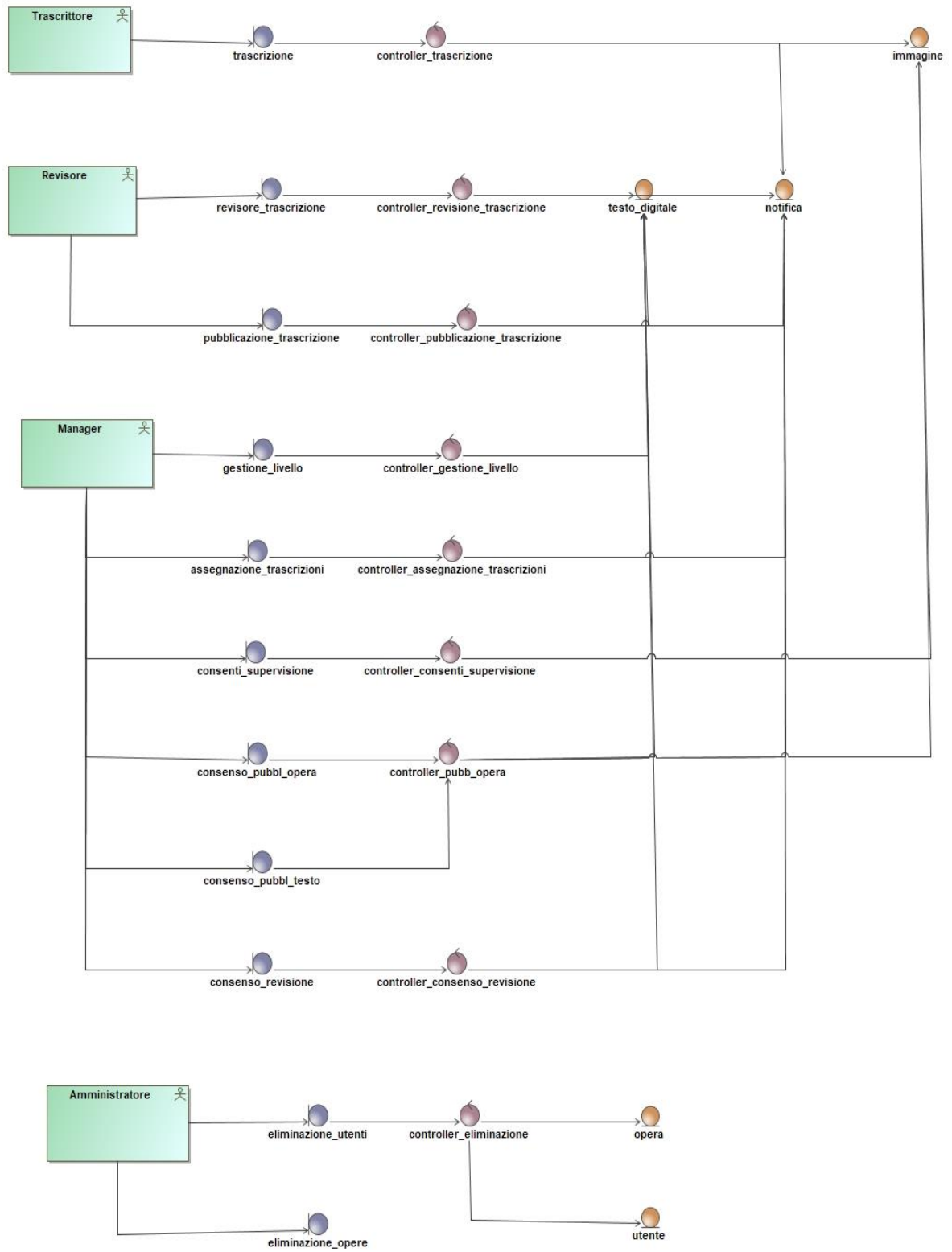


Modello di dominio



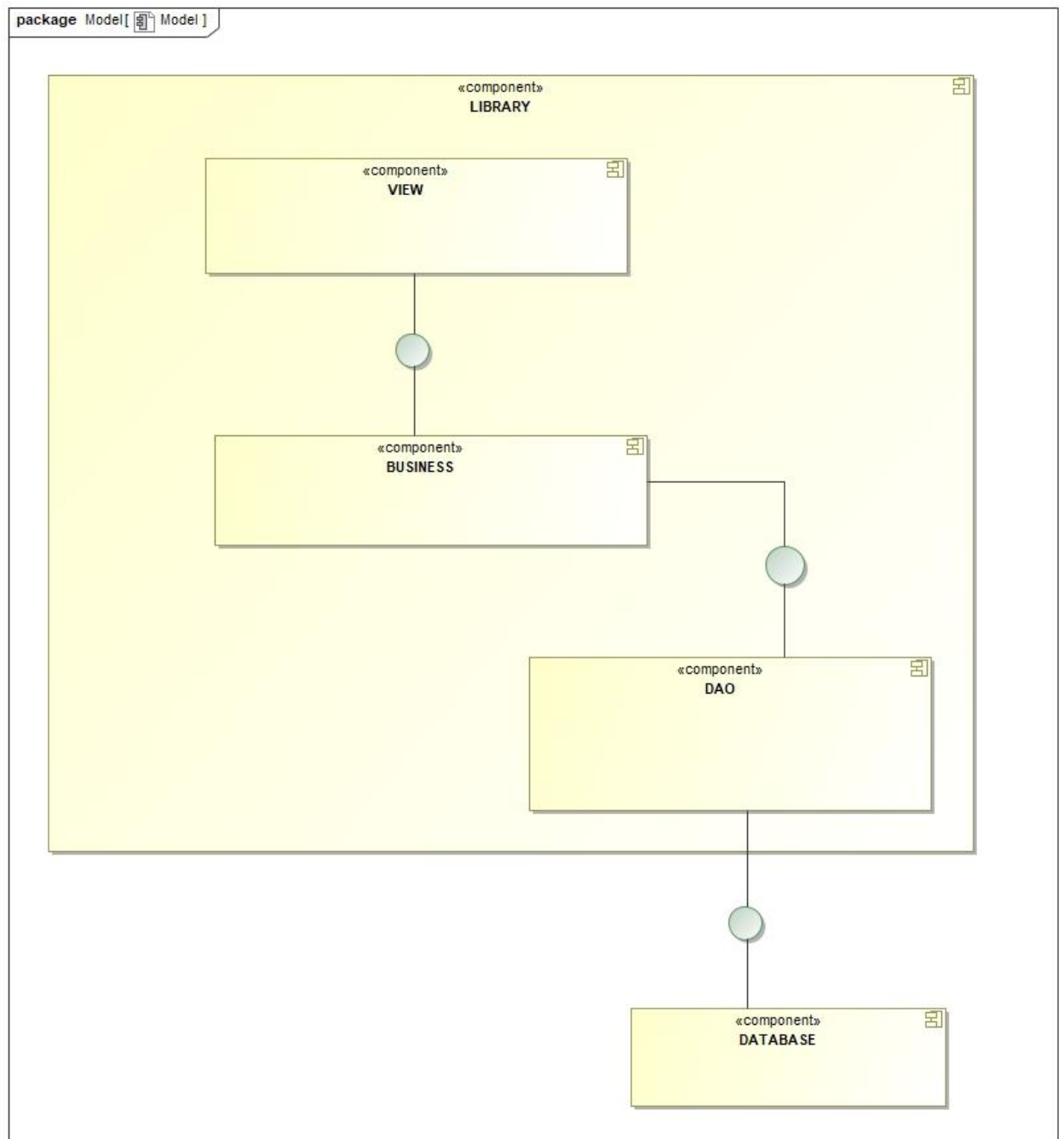
Analisi classi entity,boundary,controller





Modello dell'architettura software

Component diagram



Descrizione dell'architettura

La componente “Library” è suddivisa in diverse sottocomponenti:

- View
- Business
- Dao

La componente “View” contiene l’interfaccia grafica del sistema (realizzata con JavaFX). La componente ‘Business’ è composta da ‘Model’ (con i metodi get() e set()) e da ‘Controller’ (che esegue le varie operazioni sui dati in input). Le varie funzioni per interfacciarsi con i dati del database (e quindi eseguire query) vengono svolte all’interno della componente “Dao”.

Descrizione delle scelte e delle strategie adottate

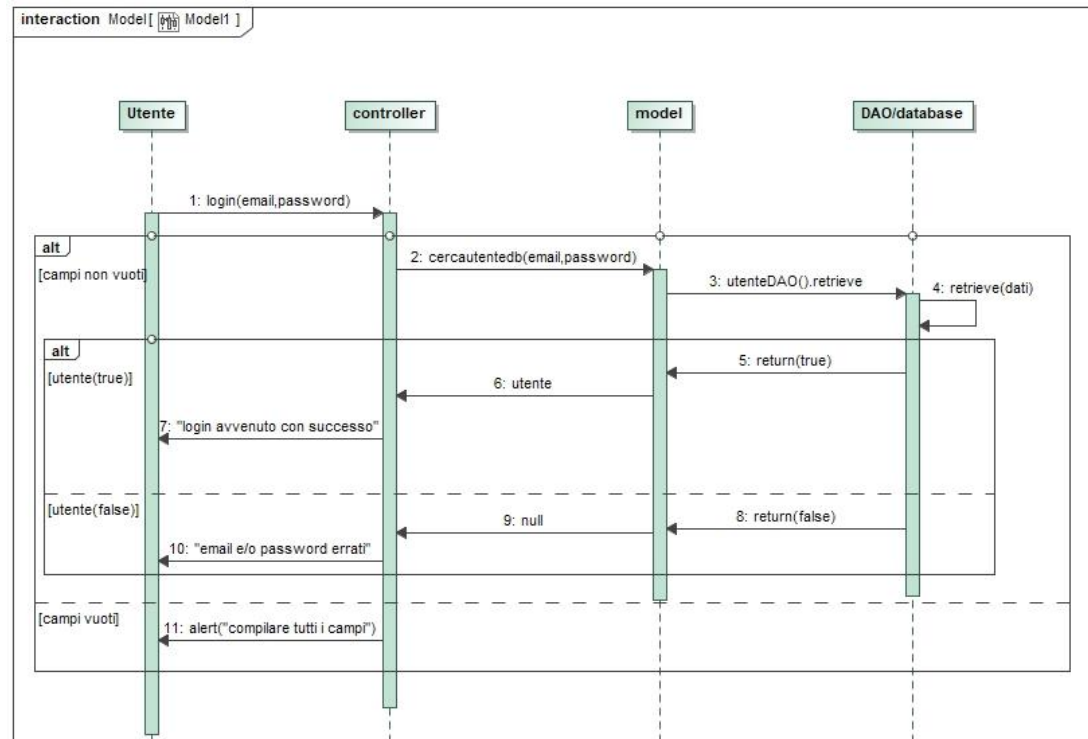
Per interfacciarsi con i dati l’architettura del sistema segue il pattern architetturale DAO. Per quanto riguarda il database la scelta è ricaduta su un database relazionale. Per quanto riguarda la suddivisione della logica il sistema segue il pattern MVC:

- Model: Business, ossia gestione dei dati
- View: Interfaccia, che permette la comunicazione tra utente e sistema
- Controller: che analizza i dati in input e li converte in comandi per Model e/o per View

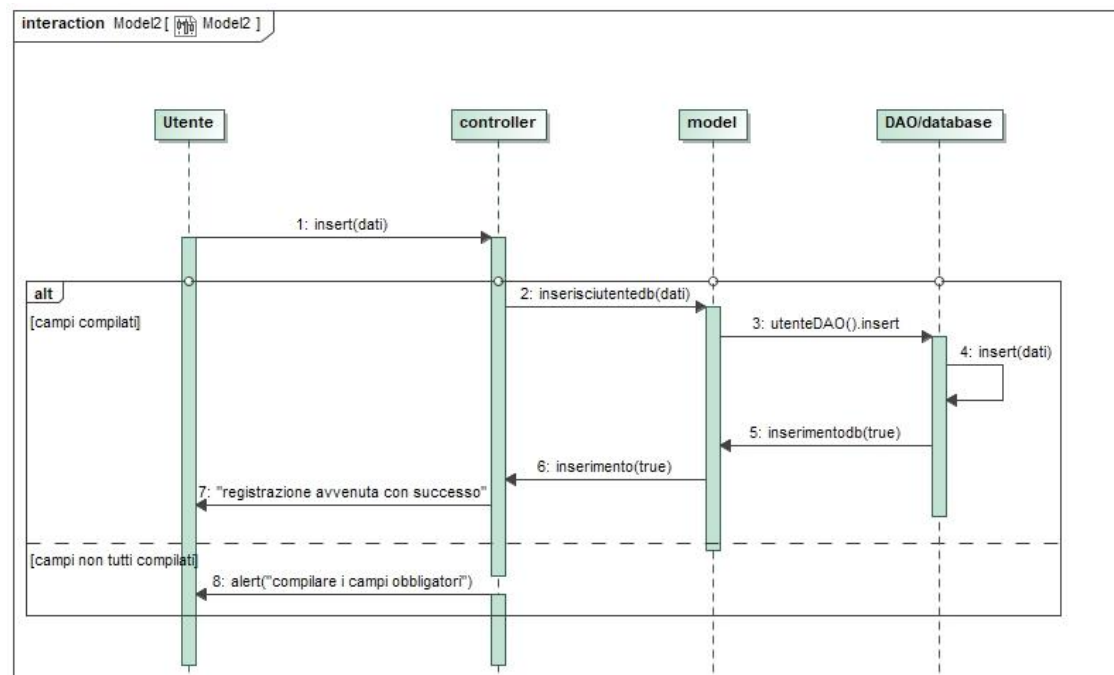
È stato inoltre utilizzato il design pattern relazionale singleton, al fine di garantire che di una classe venga creata una e una sola istanza di tipo utente. È stato implementato nella classe Utente del model.

Sequence diagram

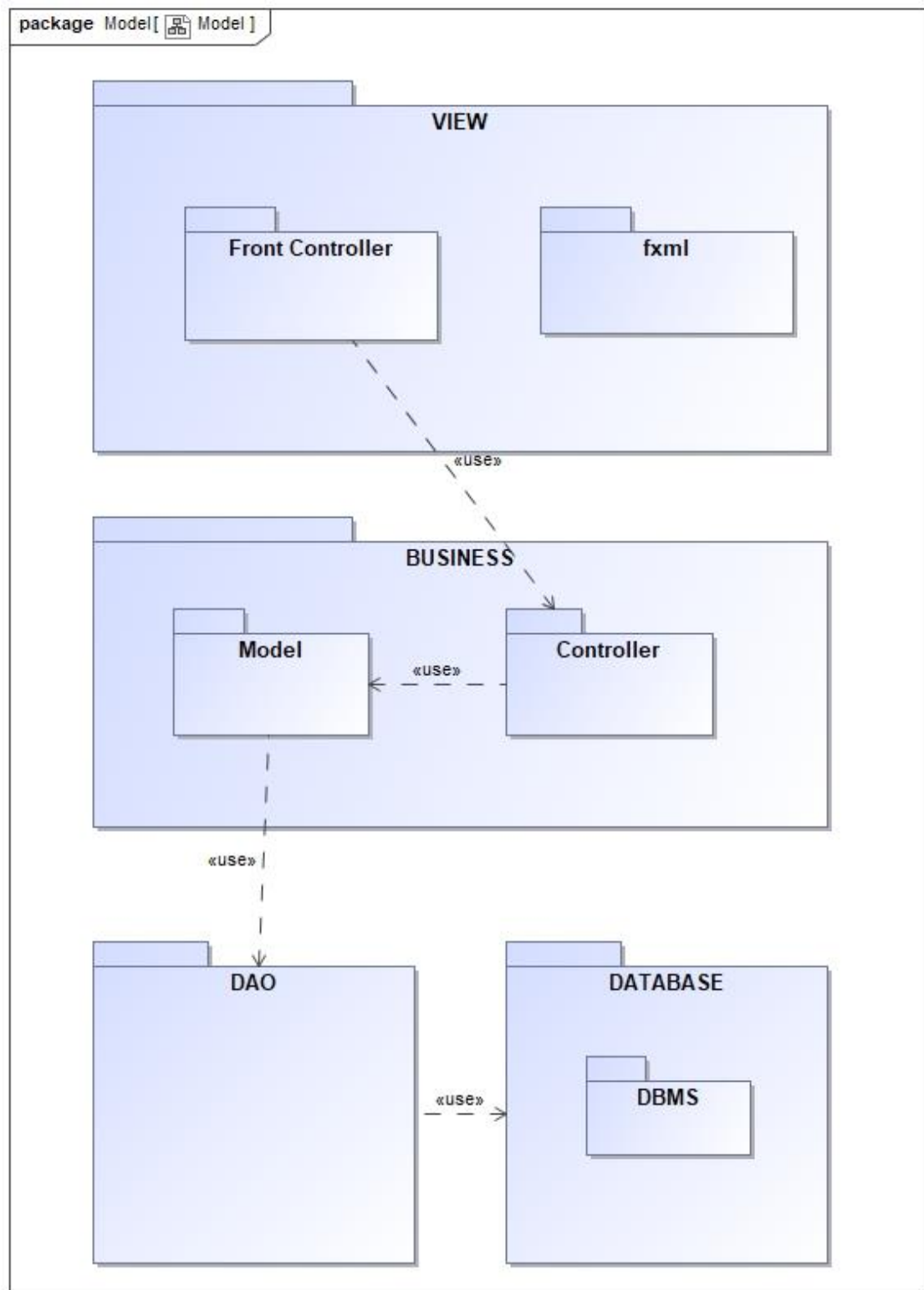
-login

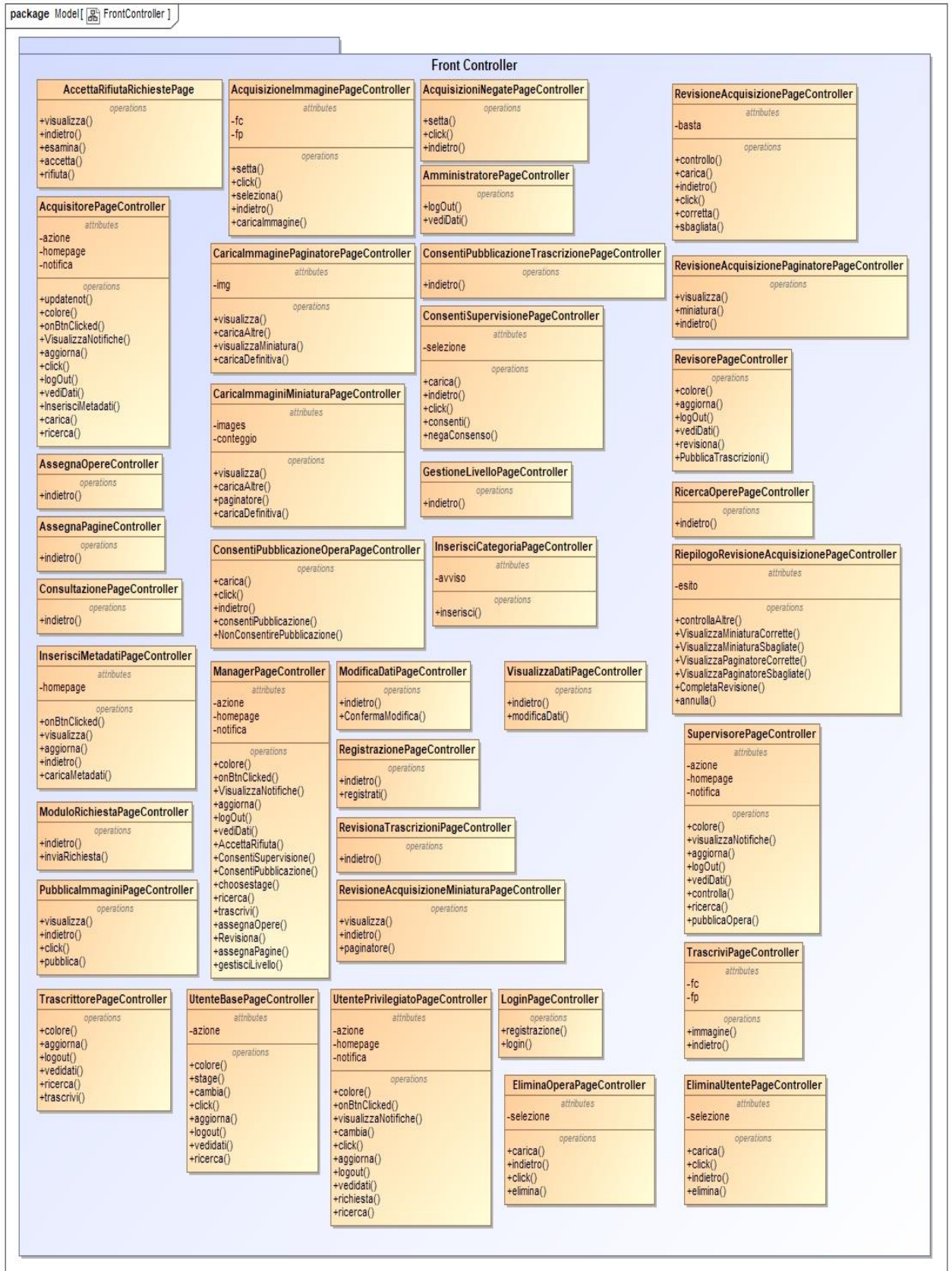



-register



Class diagram






package Model[ JavaFX]

FXML

AccettaRifiutaRichiestePage	CaricalImmaginePaginatorePage	InserisciCategoriaPage
AcquisitorePage	CaricalImmaginiMiniaturaPage	InserisciMetadatiPage
AcquisizioneImmaginePage	ConsentiPubblicazioneOperaPage	LoginPage
AcquisizioniNegatePage	ConsentiPubblicazioneTrascrizionePage	ManagerPage
AmministratorePage	ConsentiSupervisionePage	ModificaDatiPage
AssegnaOpere	ConsultazionePage	ModuloRichiestaPage
AssegnaPagine	GestioneLivelloPage	PubblicalImmaginiPage
RegistrazionePge	RevisionaTrascrizioniPage	RevisioneAcquisizioneMiniaturaPage
RevisioneAcquisizionePage	RevisioneAcquisizionePaginatorePage	RevisorePage
RicercaOperePage	RiepilogoRevisioneAcquisizionePage	SupervisorePage
TrascrittorePage	TrascriviPage	UtenteBasePage
UtentePrivilegiatoPage	VisualizzaDatiPage	EliminaOperaPage
		EliminaUtentePage

package Model [ Controller]

Controller

controller_caricamento

attributes

- categoria
- cercaopera
- n
- url
- titolo
- idutente
- Listaopere
- Listaimmagini
- caricamento

operations

- +prendicategorie()
- +inseriscimetadati()
- +controllonotnull()
- +verificacategoria()
- +associaid()
- +inseriscicategoria()
- +verifica()
- +remove()
- +controlloopera()
- +controllonotnullopera()
- +carica()
- +prendiidmanager()
- +visualizzariepilogo()
- +caricade finitiva()
- +annulla()

controller_consenso_publicazione

attributes

- listaimg
- img
- opere

operations

- +verifica()
- +prendi()
- +visualizza()
- +prendiid()
- +consenso()

controller_consenso_supervisione

attributes

- listaimg
- imgeliminate
- npag
- titolo

operations

- +verifica()
- +prendiurl()
- +prendiid()
- +consenso()
- +eliminate()
- +updatenot()

controller_dati

attributes

- nome
- cognome
- datanascita
- indirizzo
- titolostudio
- professione
- password
- ruolo
- email
- utente

operations

- +visualizza()
- +modifica()
- +visualizzaerrore()
- +modificavisualizza()

controller_domanda

attributes

- nome
- cognome
- ruolo
- utente

operations

- +datirichiesta()
- +prendiidmanager()
- +invia()

controller_notifiche

attributes

- notifiche
- elenco

operations

- +prendinotificheutente()
- +vista1()
- +esistenza()

controller_registrazione

operations

- +inviaRegistrazione()
- +controllonotnull()
- +associaid()

controller_logout

operations

- +disattivautente()

controller_ricerca

attributes

- Listaopere

operations

- +verifica()

controller_revisione_acquisizione

attributes

- imgesaminare
- imgsbagliate
- imgcorrette
- npag
- titolo

operations

- +verifica()
- +prendiurl()
- +prendiid()
- +esaminare()
- +annulla()
- +completa()

controller_login

attributes

- ruolo
- email

operations

- +verificacredenziali()

controller_publicazione_opera

attributes

- Listaimg
- opere

operations

- +prendi()

controller_richieste

attributes

- email
- nome
- cognome
- ruolo
- titolostudio

operations

- +prendiutentidomanda()
- +esaminarichiesta()
- +accettadomanda()
- +rifiutadomanda()
- +cambia()

controller_assegnazione_trascrizioni

attributes

- Listaopere
- img

operations

- +verifica()
- +prendiopere()
- +prendiid()
- +prenditrascrittori()
- +prendipagine()
- +assegna()

controller_consultazione

attributes

- open
- ruoloutente
- img

operations

- +avvia()
- +impostaruolo()
- +prendiimmagini()

controller_trascrizione

attributes

- listaopere
- img
- idimg

operations

- +verifica()
- +prendipagine()
- +prendiid()
- +completa()
- +prenditrascrizioni()
- +aggiorna()

controller_consenso_revisione

attributes

- listatrascrizioni
- idimg
- lista

operations

- +verifica()
- +prenditesto()
- +prendiid()
- +consenso()

controller_eliminazione

attributes

- opere
- utenti

operations

- +prendi()
- +elimina()
- +prendiu()
- +eliminau()

controller_gestionelivello

operations

- +prendiid()
- +prendi()
- +modifica()

controller_revisione_trascrizione

attributes

- traesaminare
- idimg
- lista

operations

- +verifica()
- +prenditesto()
- +prendiid()
- +conferma()


controller_publicazione_trascrizione

attributes

- traesaminare

operations

- +prendi()
- +prenditesto()
- +pubblica()

package Model[ Model]

Model

