

INDICE

1. REQUIREMENTS

1.1. DOCUMENTO DEI REQUISITI

1.1.1. Obiettivo, definizioni, panoramica, usecase

- 1.1.1.1. La Biblioteca
- 1.1.1.2. Definizioni, acronimi e abbreviazioni
- 1.1.1.3. Panoramica
 - 1.1.1.3.1. Progetto di digitalizzazione
 - 1.1.1.3.2. Progetto di trascrizione
- 1.1.1.4. Diagramma casi d'uso

1.1.2. Sottosistemi e requisiti funzionali

1.1.2.1. Sistema esterno

- 1.1.2.1.1. Form per l'autenticazione e la registrazione di utenti

1.1.2.2. Sistema viewer: utenti registrati

- 1.1.2.2.1. Home
- 1.1.2.2.2. Strumento di ricerca opere per tag
- 1.1.2.2.3. PageViewer per la consultazione e download di opere
- 1.1.2.2.4. Modulo per la candidatura come Transcriber
- 1.1.2.2.5. Gestione profilo utente
- 1.1.2.2.6. Segnalazione problemi

1.1.2.3. Sistema uploader: utenti digitalizzatore e revisore scansioni

- 1.1.2.3.1. Sistema di digitalizzazione
- 1.1.2.3.2. Sistema di revisione

1.1.2.4. Sistema transcriber: utenti trascrittori e revisori trascrizioni

- 1.1.2.4.1. Text Editor TEI integrato con sistema di sincronizzazione
- 1.1.2.4.2. Revisore Scansione

1.1.2.5. Sistema Manager: utenti coordinatori

- 1.1.2.5.1. Creazione opera e relativo progetto di digitalizzazione
- 1.1.2.5.2. Selezione di digitalizzatori e di revisori delle scansioni
- 1.1.2.5.3. Pubblicazione di opere nel catalogo
- 1.1.2.5.4. Creazione di un progetto di Trascrizione
- 1.1.2.5.5. Selezione dei Trascrittori e dei Revisori
- 1.1.2.5.6. Pubblicazione di una trascrizione

1.1.2.6. Sistema Administrator: utenti amministratori

- 1.1.2.6.1. Gestione ruoli e permessi
- 1.1.2.6.2. Back-end

1.1.3. Requisiti non funzionali

- 1.1.3.1. Facilità d'uso
- 1.1.3.2. Flessibilità
- 1.1.3.3. Estensibilità

1.1.4. Attori del sistema e permessi

- 1.1.4.1. Utente Registrato
 - 1.1.4.1.1. Uploader
 - 1.1.4.1.2. Upload Reviser
 - 1.1.4.1.3. Transcriber
 - 1.1.4.1.3.1. Transcription Reviser
 - 1.1.4.1.4. Coordinator
 - 1.1.4.1.5. Administrator
 - 1.1.4.1.6. Utente Esterno

1.2. UML USE CASE

1.3. MODELLI DI DOMINIO

1.4. ANALISI FINALIZZATA ALL'INDIVIDUAZIONE DELLE CLASSI ENTITY, BOUNDARY E CONTROLLER

2. SYSTEM DESIGN

2.1. MODELLO DELL'ARCHITETTURA SOFTWARE

2.2. DESCRIZIONE DELL'ARCHITETTURA

2.3. DESCRIZIONE DELLE SCELTE E DELLE STRATEGIE ADOTTATE

3. SOFTWARE/OBJECT DESIGN

3.1. UML CLASS DIAGRAMS

3.2. DESCRIZIONE DEI DETTAGLI DI DESIGN SCELTI

CONTENUTI

1. REQUIREMENTS

1.1. DOCUMENTO DEI REQUISITI

1.1.1. *Obiettivo, definizioni, panoramica, usecase*

1.1.1.1. La Biblioteca

Progettazione di una biblioteca digitale: 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. Il sistema deve consentire la consultazione dei manoscritti che devono essere digitalizzati e che costituiscono un patrimonio bibliografico antico per un totale di 60.000 carte contenenti memorie storiche della città de L'Aquila.

1.1.1.2. Definizioni, acronimi e abbreviazioni

- Una biblioteca è un insieme di documenti
- Una collezione è formata da documenti che condividono le stesse caratteristiche
- Un documento (opera o manoscritto) è definita come una collezione di pagine più dei metadati, ogni opera possiede una lista di **tag** (attributi) utili a catalogarla
- Un catalogo è formato da collezioni di documenti scelte dall'amministratore del sistema, queste particolari collezioni sono chiamate categorie

1.1.1.3. Panoramica

All'interno del processo di pubblicazione di un'opera generalmente sono coinvolti i sistemi Viewer, Uploader, Transcriber e Manager, l'utente mediante dei ruoli accede a tali sottosistemi. La gestione dei ruoli è affidata agli utenti amministratori.

Sistema Viewer: accessibile dall'utente registrato

Sistema Uploader: accessibile dall'utente digitalizzatore e dall'utente revisore delle digitalizzazioni

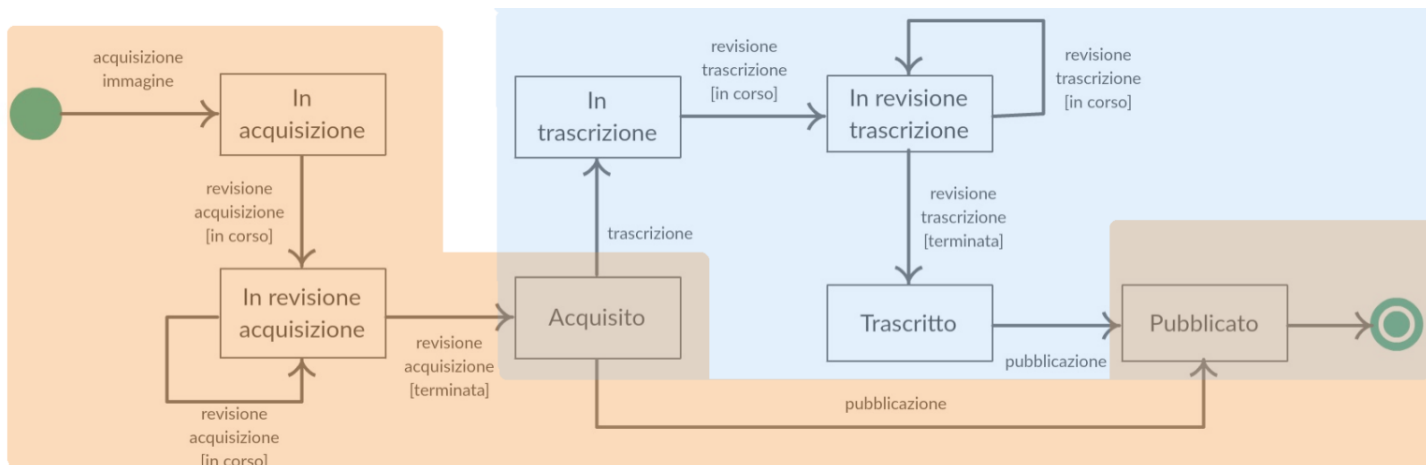
Sistema Transcriber: accessibile dall'utente tracciatore e dall'utente revisore delle trascrizioni

Sistema Manager: coordinatore del progetto

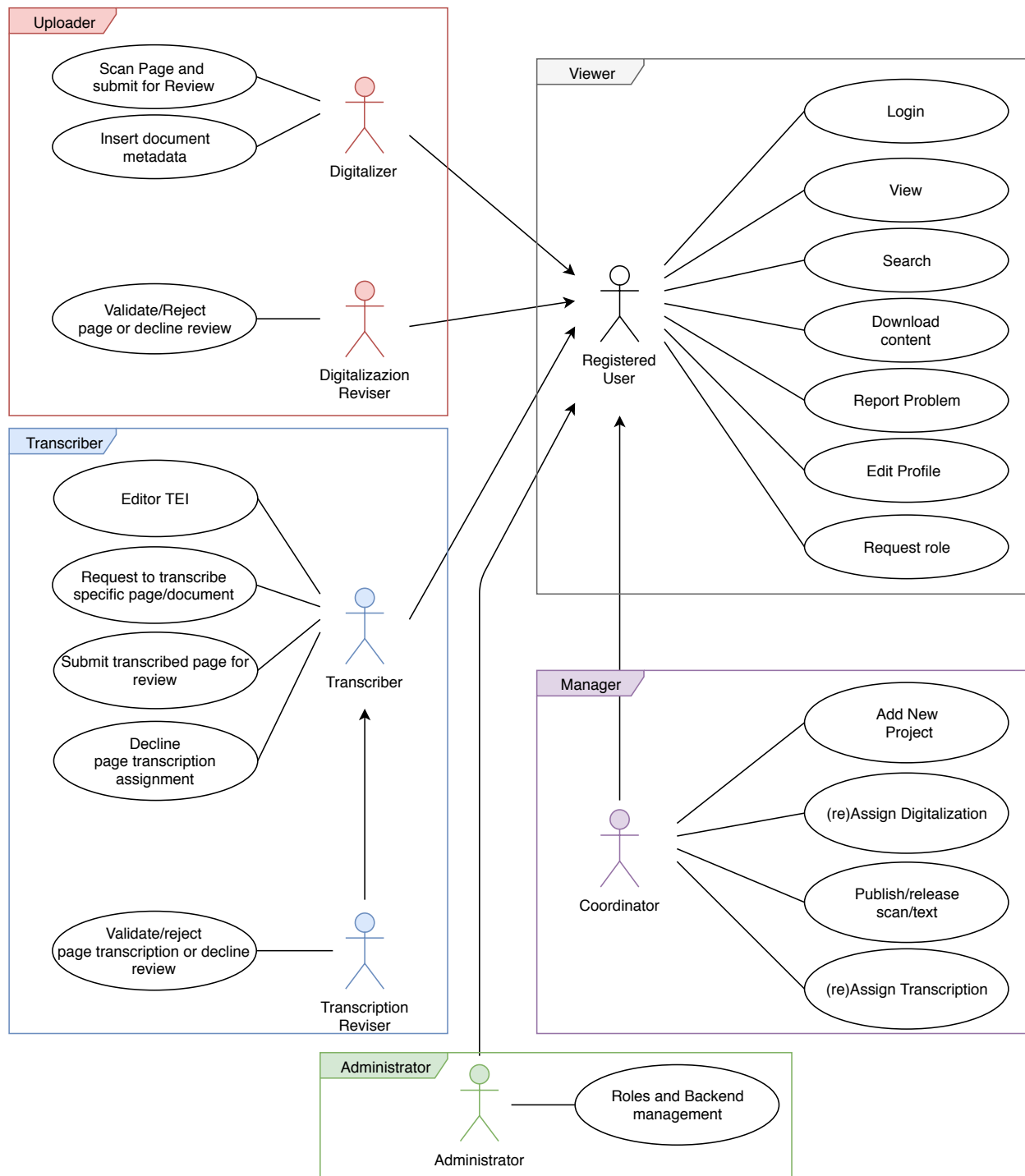
Sistema Administrator: amministratore del sistema

Tutti questi sottosistemi lavorano ad un unico fine che è la pubblicazione di un'opera (scansione ed eventualmente trascrizione) all'interno della biblioteca.

Possiamo distinguere due tipi di processi: un processo di digitalizzazione e un processo di trascrizione.



1.1.1.4. Diagramma casi d'uso



1.1.2. Sottosistemi e requisiti funzionali

1.1.2.1. Sistema esterno

1.1.2.1.1. Form per l'autenticazione e la registrazione di utenti

Il sistema esterno si presenta come un'interfaccia pubblica che consente l'accesso all'interno dell'applicazione mediante un form di login o di registrazione che proietterà l'utente nella schermata home del sistema.

La registrazione all'interno della piattaforma può essere configurata in due modalità a discrezione dell'amministratore del sistema:

- Iscrizione libera: l'utente inserisce i propri dati e si registra automaticamente nel sistema mediante mail di conferma.
- Iscrizione supervisionata: l'utente invia una richiesta di iscrizione e l'amministratore decide se accettarla o meno.

1.1.2.2. Sistema viewer

Tutti gli utenti registrati sono anche utenti viewer

1.1.2.2.1. Home

- La schermata home è composta da un catalogo dove sono mostrate le categorie principali accessibili e consultabili.
- La schermata home presenta una vetrina con le opere più recenti.
- La schermata home elenca i segnalibri dell'utente.
- La schermata home contiene una bacheca con gli avvisi più importanti

1.1.2.2.2. Strumento di ricerca opere per tag

- Il sistema possiede uno strumento di ricerca facilmente accessibile che consente di cercare un'opera all'interno della biblioteca.
- La ricerca avviene mediante l'inserimento di parole chiave e tag all'interno della barra di input.
- Il risultato della ricerca consiste in una lista di opere consultabili dall'utente.

1.1.2.2.3. PageViewer per la consultazione e download di opere

- Selezionando un'opera si visualizza una vetrina con info e dettagli.
- Avviando il paginatore si accede al contenuto dell'opera, esso si presenta similmente a un lettore PDF che consente di scrollare le pagine contenenti le scansioni dell'opera.
- Eventualmente se presente una trascrizione il paginatore presenta una doppia schermata che consente di visualizzare e scrollare parallelamente scansioni e trascrizioni in due frame distinti.
- I due frame possono essere scambiati di posizione o minimizzati dall'utente consentendogli di visualizzare solo i contenuti di suo interesse.
- Inoltre il paginatore fornisce uno strumento di ricerca interno al testo dell'opera.
- È possibile aggiungere dei segnalibri alle pagine di un'opera.
- Solo utenti con particolari privilegi possono scaricare l'opera visualizzata

1.1.2.2.4. Modulo per la candidatura come Transcriber

- L'utente iscritto può compilare un modulo di richiesta per diventare un trascrittore.
- L'utente amministratore può accettare o meno la richiesta degli utenti che intendono diventare trascrittori, l'amministratore

inoltre può delegare questo suo compito ad utenti fidati tramite la gestione dei permessi dal pannello di controllo.

1.1.2.2.5. Gestione profilo utente

L'identità reale dell'utente viene memorizzata nell'anagrafica nel momento della registrazione più un username, una password e un indirizzo email. L'utente può modificare il campo username, mail e il campo password. Le informazioni anagrafiche rimangono immutate. Un utente può accedere ai diversi sottosistemi del sistema solo se gli sono stati assegnati i rispettivi ruoli.

1.1.2.2.6. Segnalazione problemi

1.1.2.3. Sistema uploader

Gli utenti possono avere accesso al sistema uploader solo se hanno il ruolo di digitalizzatore.

1.1.2.3.1. Sistema di digitalizzazione

Un qualsiasi utente può diventare un digitalizzatore sotto nomina dell'amministratore e di norma non possono farne richiesta per diventarlo, dato che i digitalizzatori probabilmente sono un personale competente e responsabile che ha accesso fisicamente ai manoscritti e solitamente sono a stretto contatto nella realtà con l'amministratore. Per mantenere una certa flessibilità nel sistema abbiamo deciso di aggiungere nel pannello dei permessi un flag che consente di abilitare i moduli di richiesta per diventare digitalizzatore. Un digitalizzatore può entrare a far parte di un progetto mediante una richiesta o un invito da parte di un coordinatore. Il digitalizzatore mediante apposita schermata può caricare nuove scansioni da aggiungere all'opera. Una volta caricata l'intera opera il digitalizzatore deve settare la scansione come completata e inviarla per una revisione.

1.1.2.3.2. Sistema di revisione

Un revisore delle scansioni è un utente del sistema nominato revisore delle scansioni all'interno di uno specifico progetto, è il coordinatore a nominare il revisore. Una scansione completata viene controllata da un revisore delle scansioni che convalida le pagine o ne richiede una nuova scansione, quando tutte le pagine sono state convalidate l'opera può essere segnalata al coordinatore come revisionata.

1.1.2.4. Sistema transcriber: utenti trascrittori e revisori trascrizioni

1.1.2.4.1. Text Editor TEI integrato con sistema di sincronizzazione

Sistema TEI per la trascrizione delle pagine. Ogni pagina può avere più trascrittori i quali accedono in maniera mutualmente esclusiva al testo, pagine diverse della stessa opera possono essere accedute contemporaneamente. Un trascrittore può visualizzare una scansione per volta affiancata dall'editor TEI. Ogni qualvolta il trascrittore cambia pagina il sistema chiede al trascrittore se vuole effettuare il salvataggio con possibilità di automatizzare il progetto di salvataggio per tutte le pagine. Inoltre un qualsiasi trascrittore che lavora su una pagina può segnalare la trascrizione ad un revisore.

1.1.2.4.2. Sistema di revisione

I revisori delle trascrizioni si occupano di controllare la qualità delle trascrizioni e di assicurarsi che esse rispettino il formato TEI, egli può convalidare le pagine della trascrizione o richiederne la modifica. Quando tutte le pagine sono state convalidate la trascrizione dell'opera può essere segnalata al coordinatore come

revisionata.

1.1.2.5. **Sistema Manager**

Il sistema Manager consente la creazione e la gestione di processi di digitalizzazione e di processi di trascrizione di opere e la gestione del personale che collabora al progetto. Il ruolo di coordinatore viene nominato dall'amministratore. Un coordinatore può essere specializzato nelle digitalizzazioni o nelle trascrizioni o in entrambi, a seconda dei permessi dati dall'amministratore.

1.1.2.5.1. **Creazione opera e relativo progetto di digitalizzazione**

Il coordinatore delle digitalizzazioni attraverso un'apposita schermata può creare un document e quindi avviare un progetto di digitalizzazione, durante la creazione egli deve inserire i metadati dell'opera più dettagli aggiuntivi sul progetto.

1.1.2.5.2. **Selezione di digitalizzatori e di revisori delle scansioni**

La selezione del personale per la digitalizzazione e la revisione avviene su una lista di utenti digitalizzatori. Inoltre il coordinatore può anche pubblicare un avviso in bacheca invitando gli utenti registrati ad avanzare una richiesta per partecipare al progetto come revisori.

1.1.2.5.3. **Pubblicazione di opere nel catalogo**

Il coordinatore può accedere a tutto il materiale del progetto e osservarne l'evoluzione. L'utente coordinatore attenderà la digitalizzazione dell'opera e la segnalazione dell'opera come revisionata, a quel punto il coordinatore potrà decidere se pubblicare l'opera oppure respingerla e richiedere una nuova revisione. Se il coordinatore si occupa anche della trascrizione può decidere di rimandare la pubblicazione dell'opera direttamente al termine della trascrizione. Unico form con doppia funzione

1.1.2.5.4. **Creazione di un progetto di Trascrizione**

Il Coordinatore può avviare la trascrizione di un'opera già pubblicata nel sistema attraverso un'apposita interfaccia stabilendone durata e termine del progetto.

1.1.2.5.5. **Selezione dei Trascrittori e dei Revisori**

La scelta del personale per la trascrizione avviene su una lista di trascrittori, i revisori sono scelti sempre tra i trascrittori. Inoltre il coordinatore può anche pubblicare un avviso in bacheca invitando i trascrittori competenti ad avanzare una richiesta per partecipare al progetto, oppure invitando gli altri utenti a fare richiesta per diventare trascrittori e poi avanzare una richiesta per quello specifico progetto.

1.1.2.5.6. **Pubblicazione di una trascrizione**

1.1.2.6. **Sistema Administrator**

L'amministratore gestisce le registrazioni degli utenti, i ruoli e le configurazioni del sistema.

1.1.2.6.1. **Gestione ruoli e permessi**

1.1.2.6.2. **Back-end**

1.1.3. Requisiti non funzionali

1.1.3.1. Facilità d'uso

1.1.3.2. Flessibilità

1.1.3.3. Estensibilità

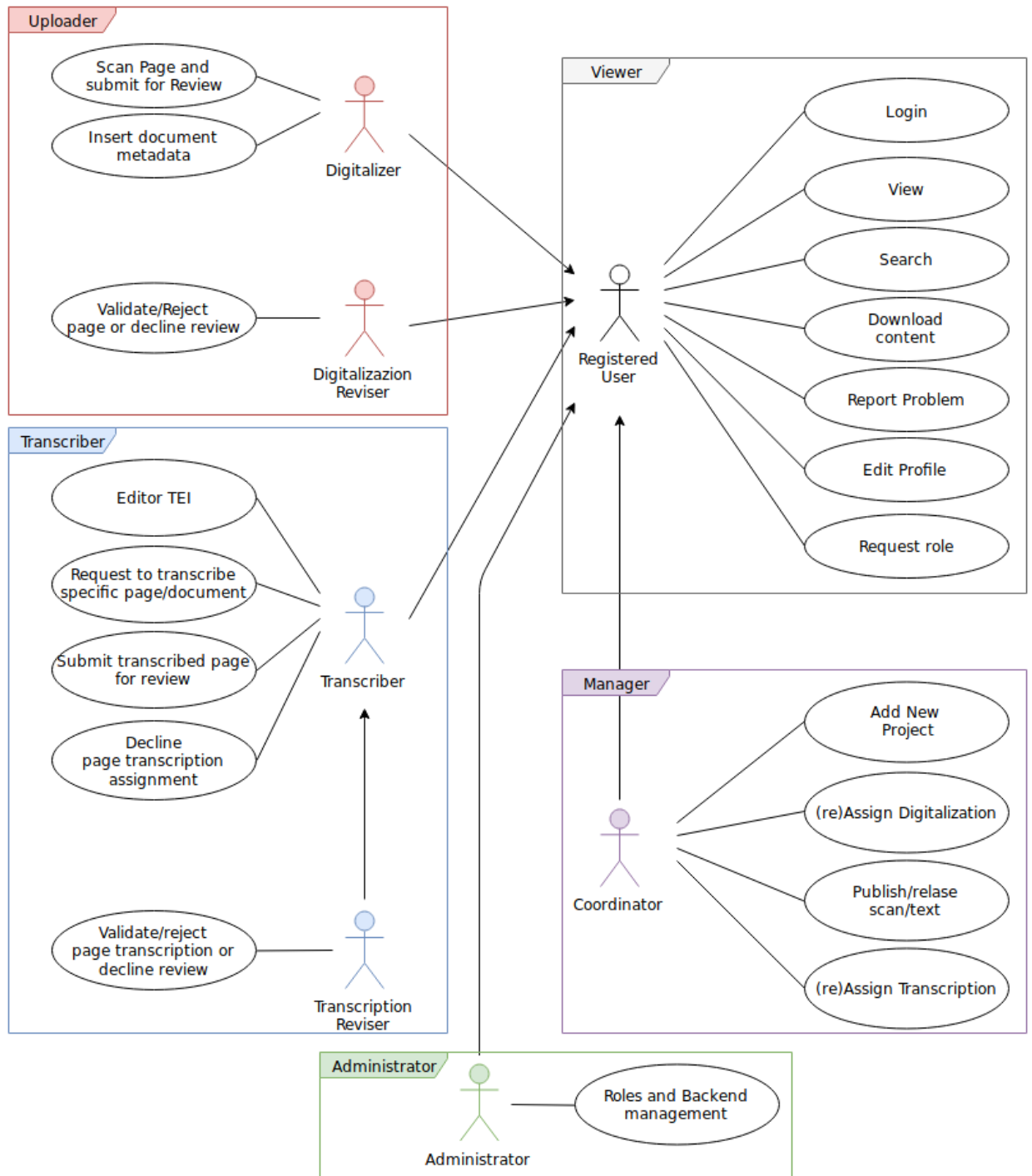
1.1.4. Attori del sistema e permessi

1.1.4.1. Utente Registrato

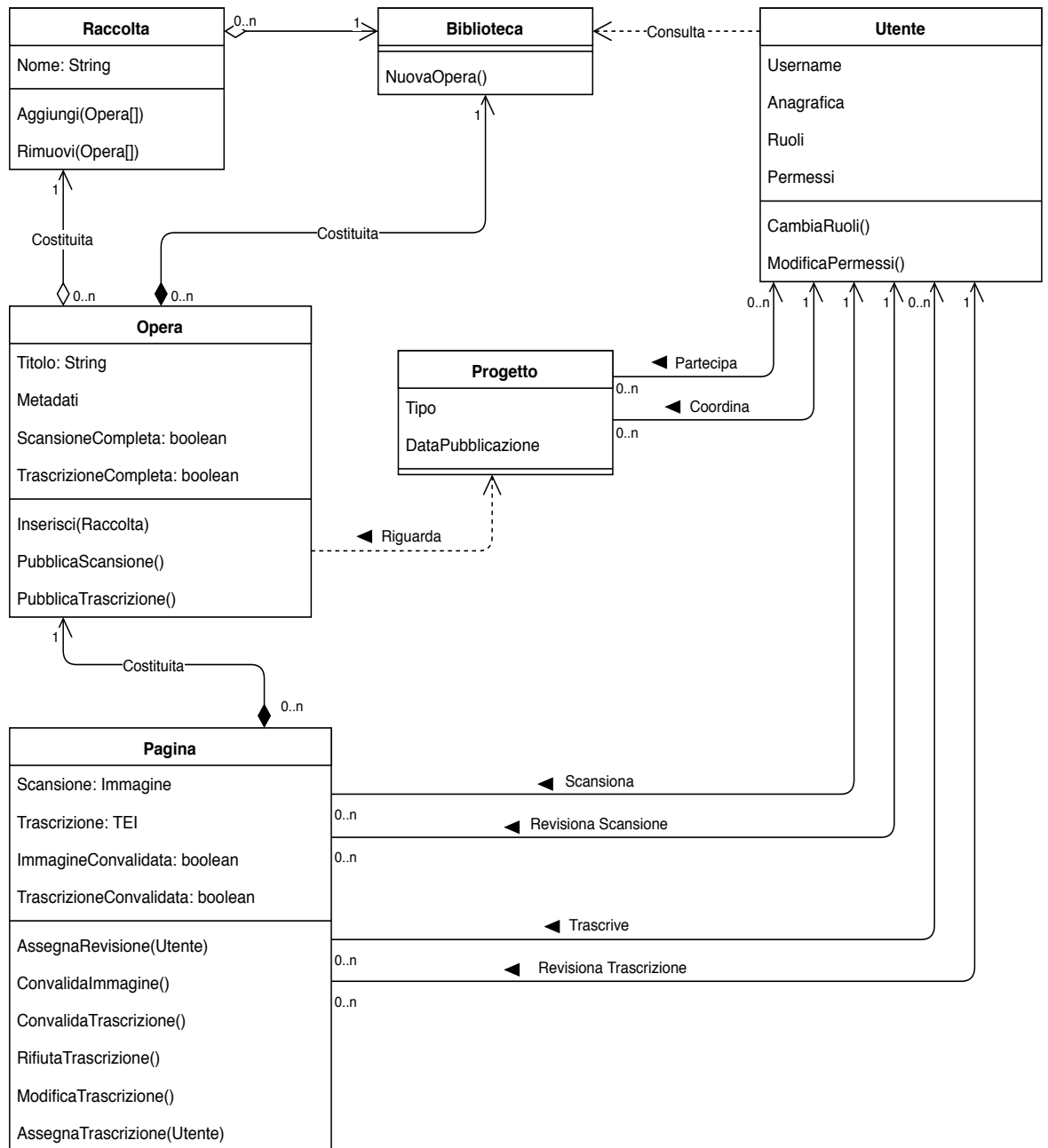
1.1.4.1.1. Uploader

- 1.1.4.1.2. Upload Reviser
- 1.1.4.1.3. Transcriber
 - 1.1.4.1.3.1. Transcription Reviser
- 1.1.4.1.4. Coordinator
- 1.1.4.1.5. Administrator
- 1.1.4.1.6. Utente Esterno

1.2. UML USE CASE



1.3. MODELLI DI DOMINIO

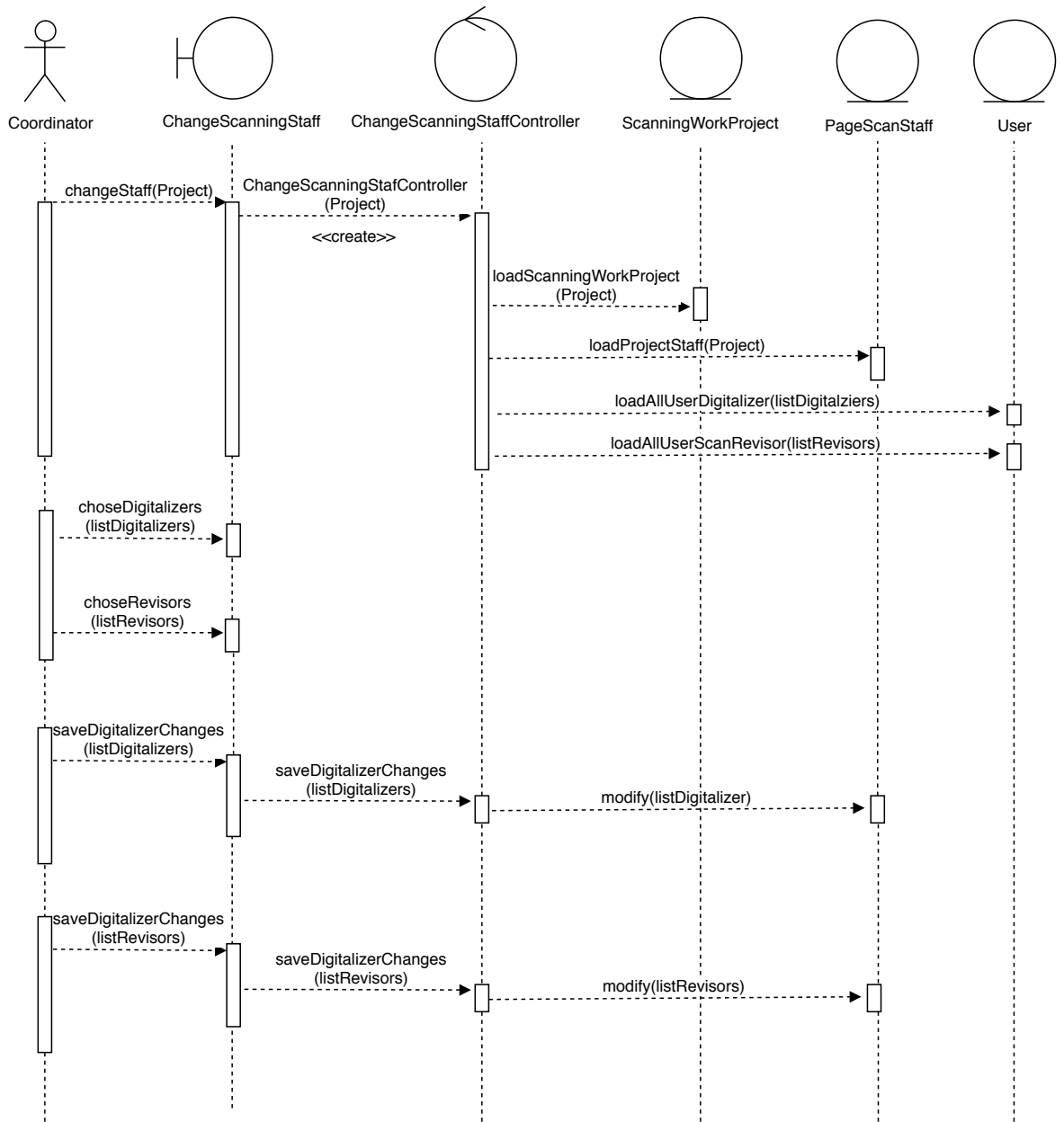


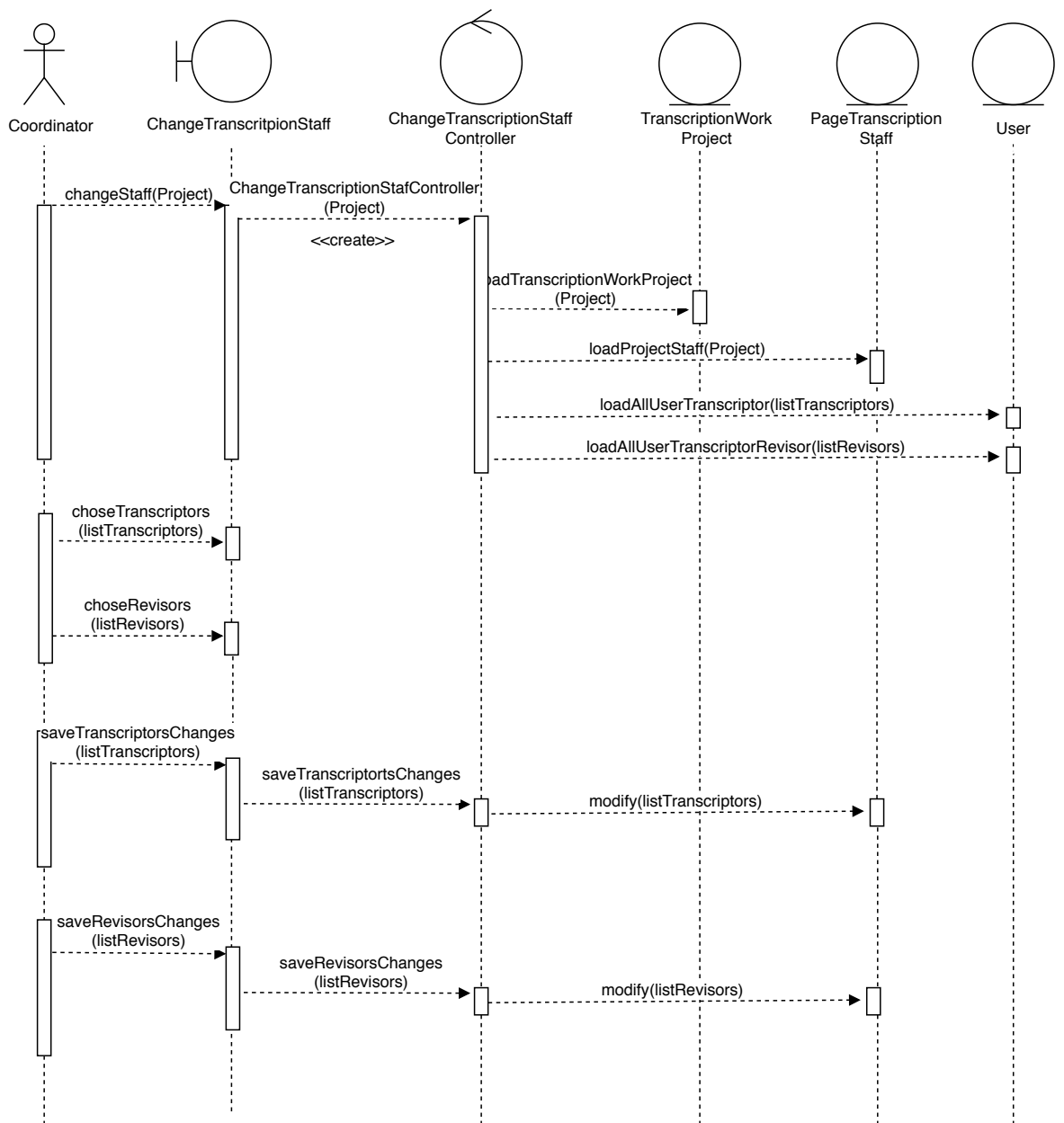
1.4. ANALISI FINALIZZATA ALL'INDIVIDUAZIONE DELLE CLASSI ENTITY, BOUNDARY E CONTROLLER

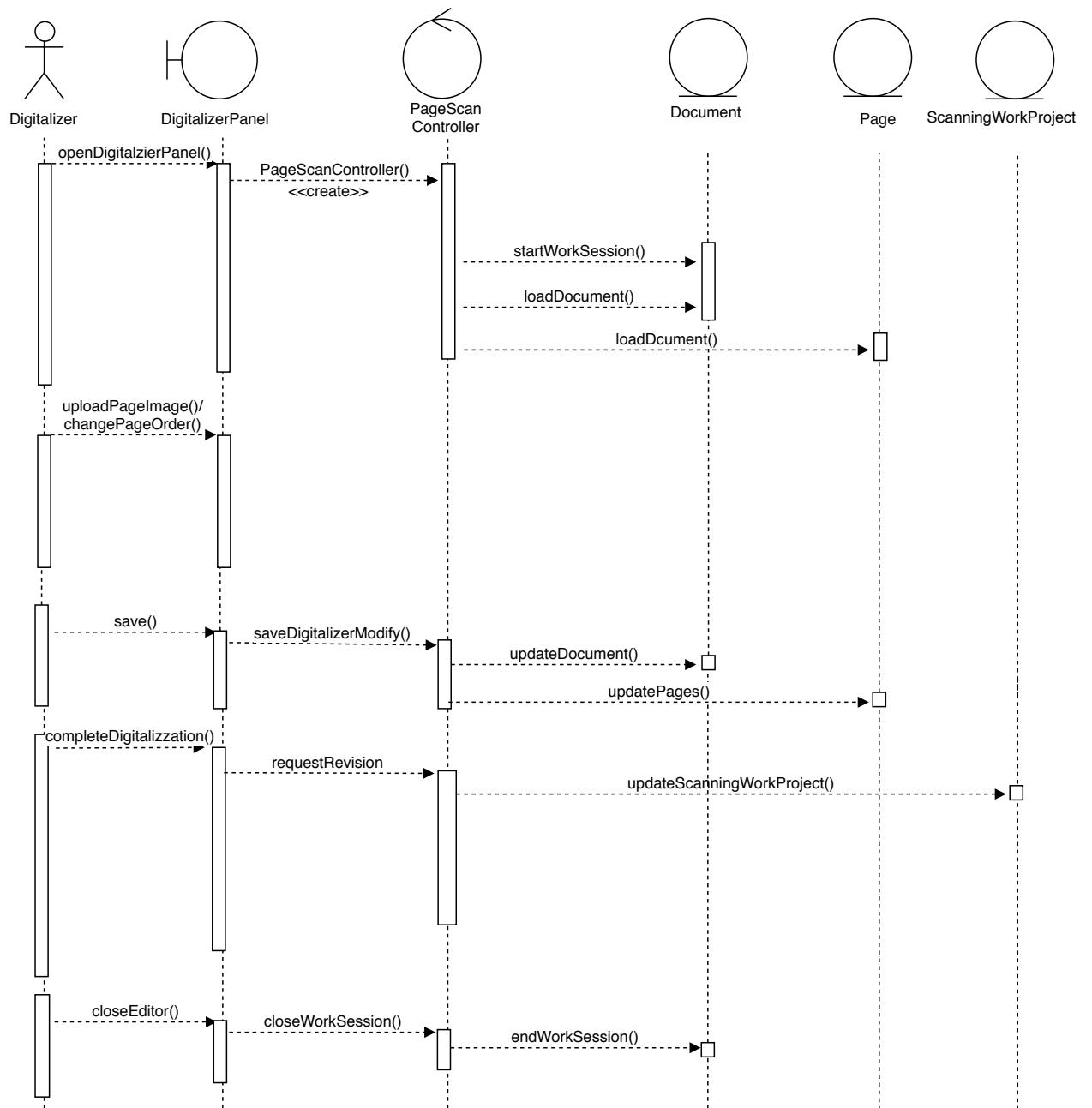
BOUNDARY E CONTROLLER

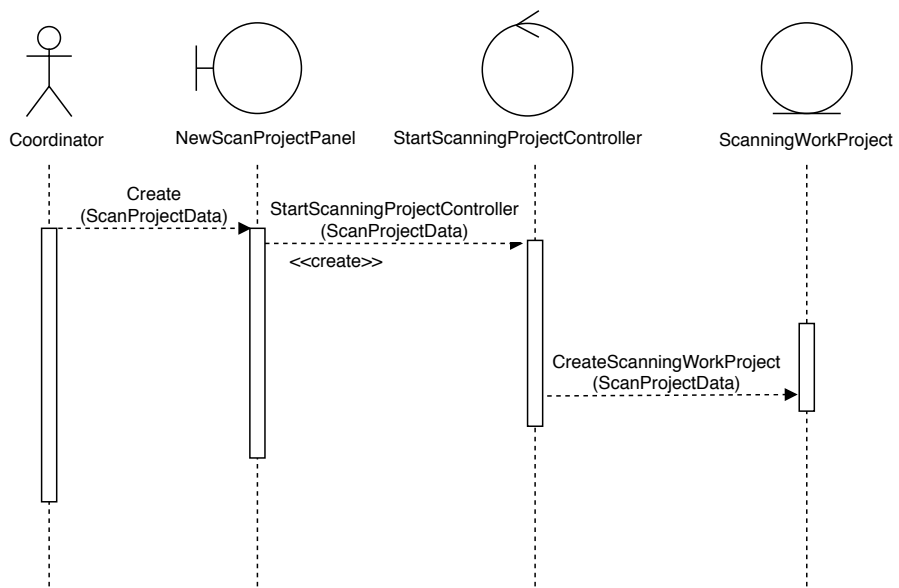
Gli oggetti entity corrisponderanno alle classi del package model nel class diagram, derivanti dalle entità esplicitate nel modello di dominio, secondo quanto verrà chiarito nel documento di descrizione dell'architettura.

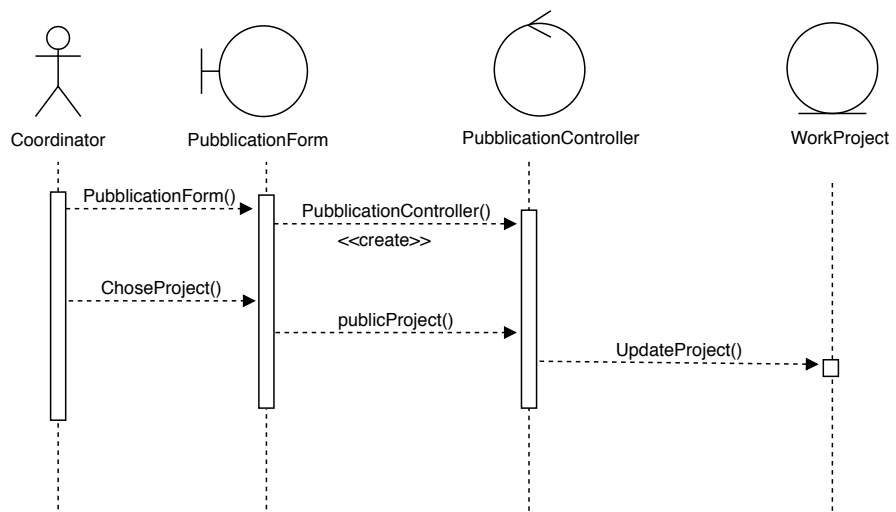
<i>Boundary</i>	<i>Controller</i>
LoginForm	LoginController
RegistrationForm	RegistrationController
HomePage	HomePageController
EditProfileForm	ProfileController
PageView	PageViewController
RoleRequestForm	RoleRequestController
DocumentSearchForm	DocumentSearchController
DigitalizerPanel	PageScanController
DigitalizerRevisorPanel	PageScanController
ExternalTeiTextEditor	PageTranscriptionController
TranscriptionRevisorPanel	PageTranscriptionController
ChangeScanningStaff	ChangeScanningStaffController
CreateEditDocument	CreateDocumentController EditDocumentController
ChangeTranscriptionStaff	ChangeTranscriptionStaffController
PubblicationForm	PubblicationController
RoleRequestAnswer	AdministrationController
EditRoles	AdministrationController
StartTranscriptionProjectForm	StartTranscriptionProjectController
StartScanningProjectForm	StartScanningProjectController

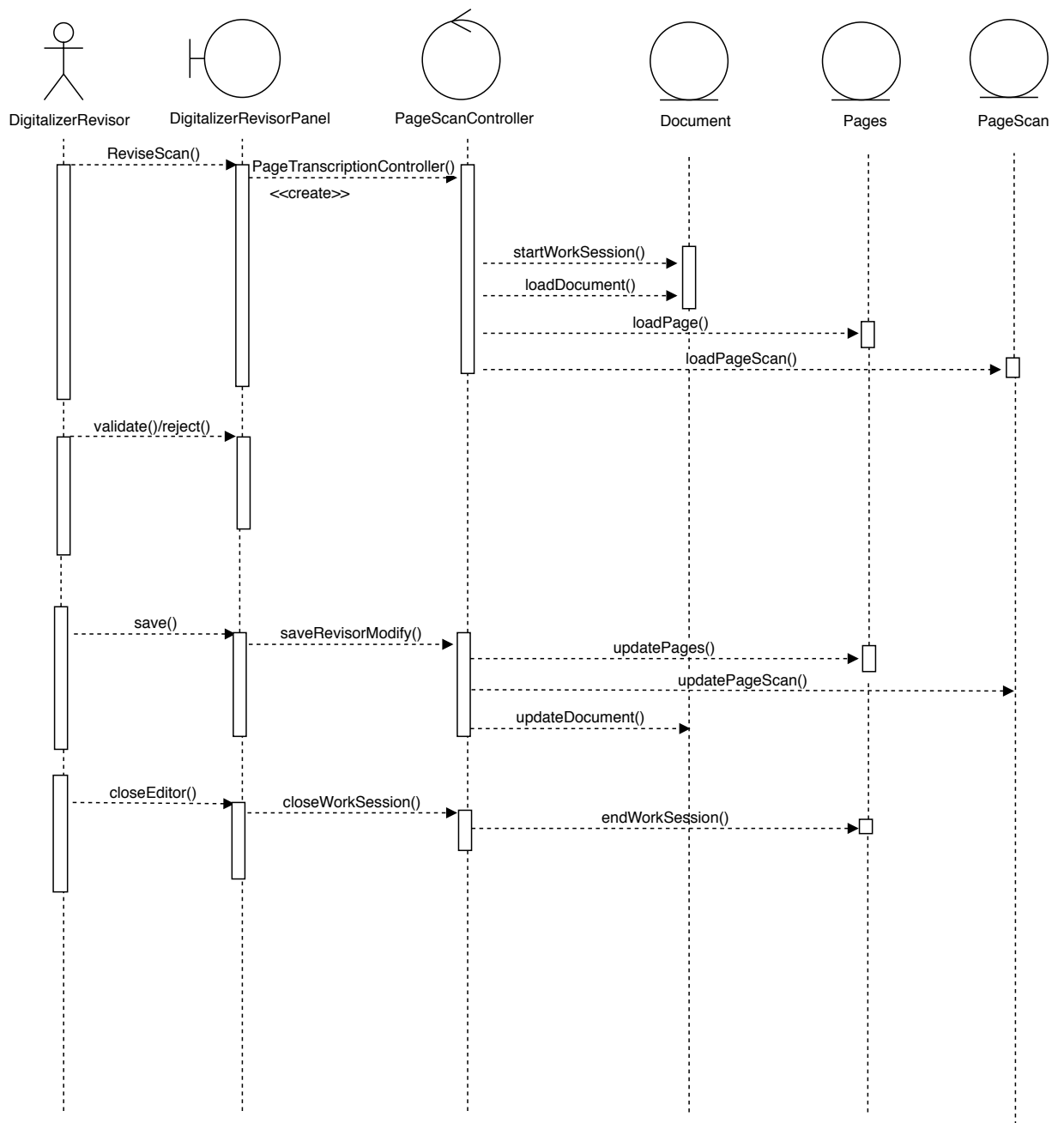


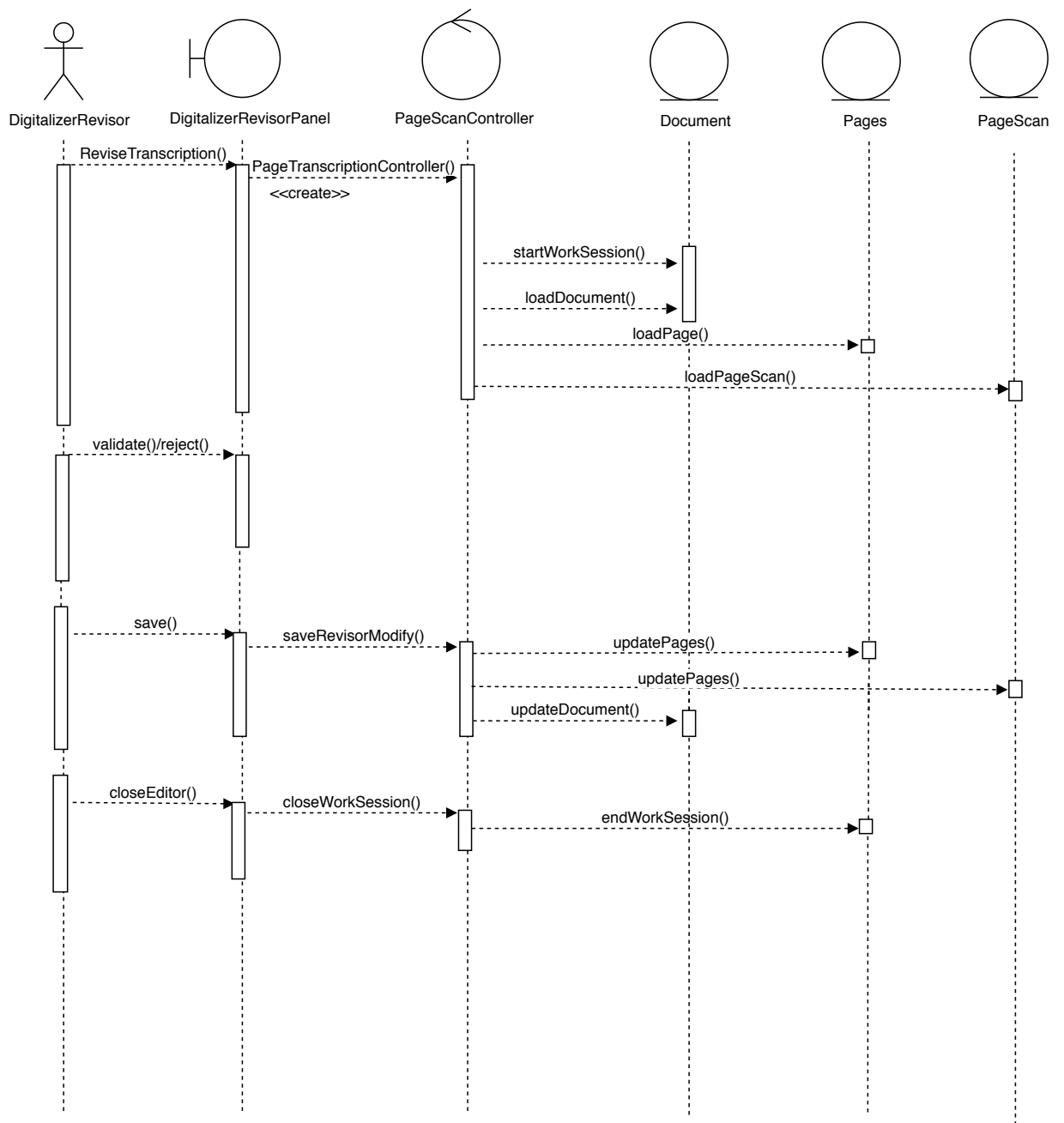


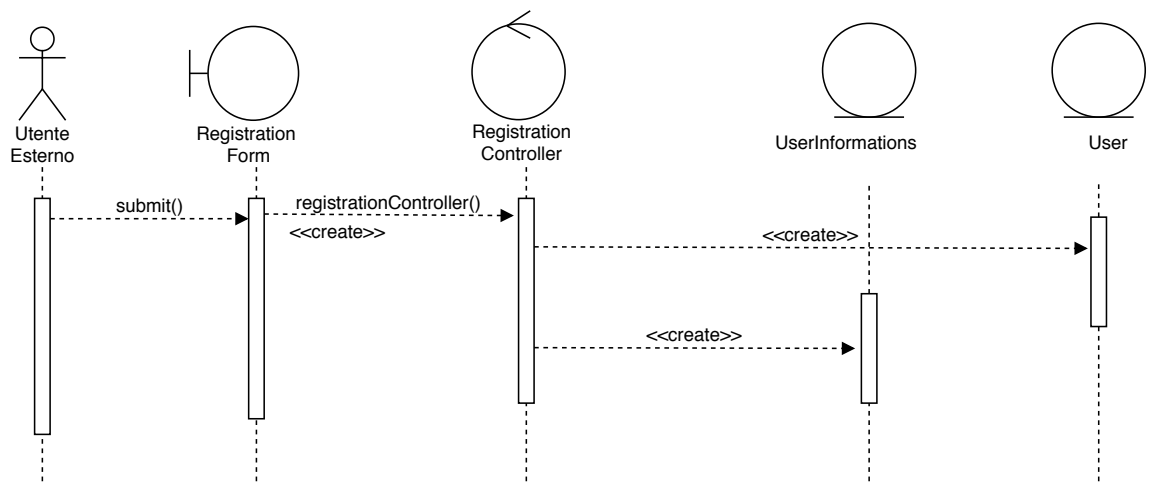






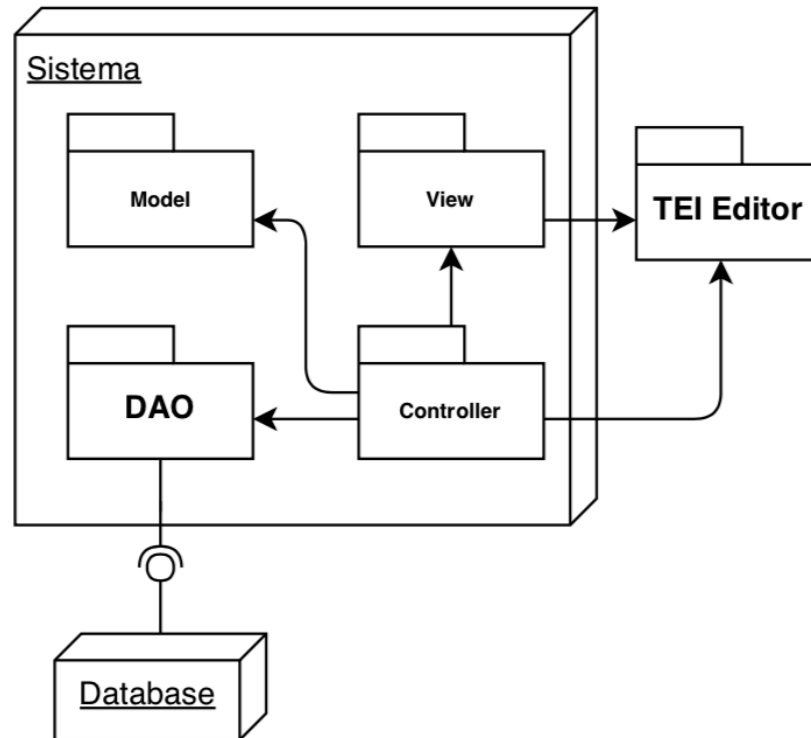


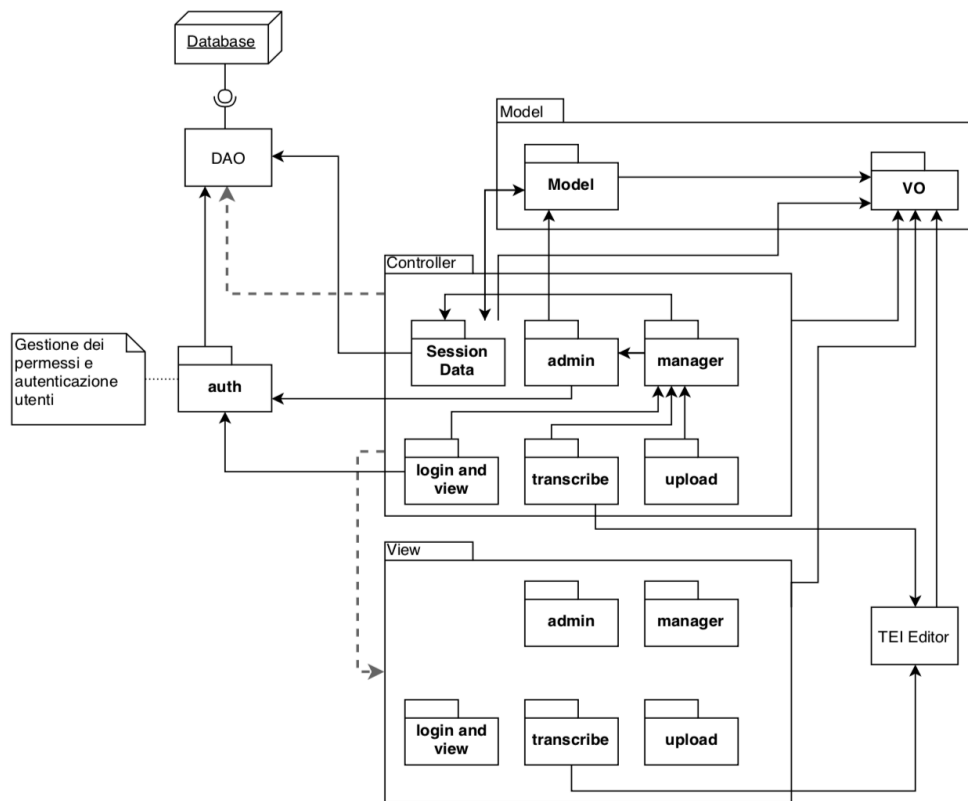




2. SYSTEM DESIGN

2.1. MODELLO DELL'ARCHITETTURA SOFTWARE





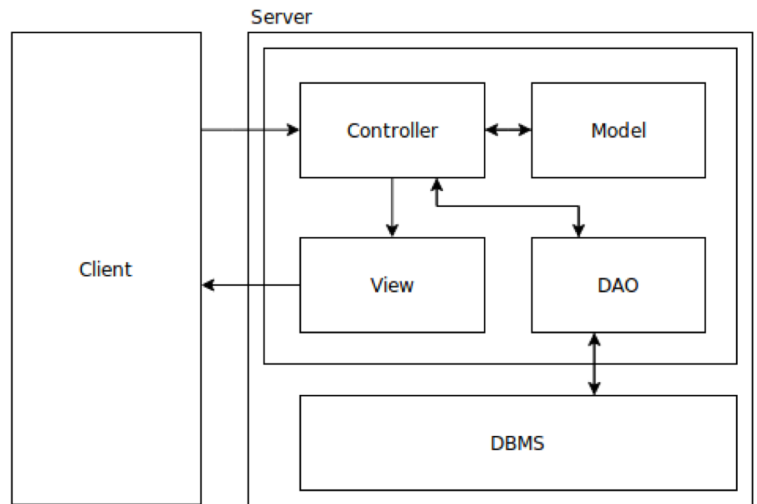
2.2. DESCRIZIONE DELL'ARCHITETTURA

Il sistema è composto di due parti, un applicativo server e un client. La parte client è stata per ora tralasciata e verrà studiata nel dettaglio al momento dell'implementazione del server. Perciò d'ora in avanti ci limiteremo a considerare unicamente la parte server.

Il sistema è stato progettato tenendo a mente i requisiti di scalabilità, flessibilità ed estensibilità. La scelta è stata quella di seguire un pattern architetturale Model-View-Controller.

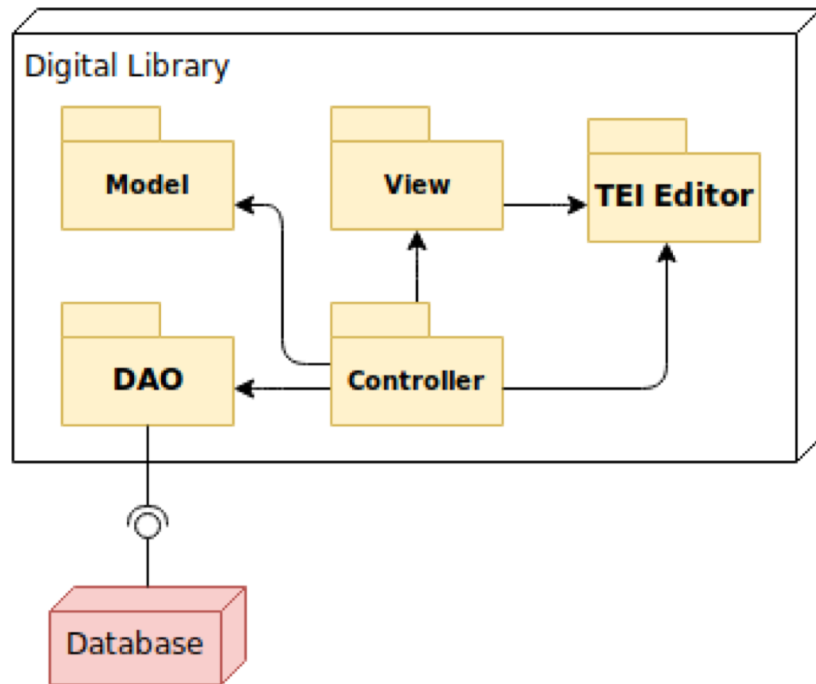
Oltre ai tre package Model, View e Controller sono presenti VO, DAO e TEI Editor; il package VO contiene semplici classi utilizzate frequentemente nel sistema, DAO si occupa di astrarre l'accesso al DBMS, TEI Editor sfrutta una libreria esterna per implementare l'editor TEI per le trascrizioni (quindi per ora non è stato considerato).

Si era pensato di aggiungere un package per la gestione dell'autenticazione e dei permessi degli utenti sfruttando delle librerie esterne ma si è ritenuto che a questo livello di progettazione avrebbe aggiunto complessità eccessiva al sistema.



Semplice schematizzazione della suddivisione logica del sistema da cui derivano le scelte di suddivisione in package.

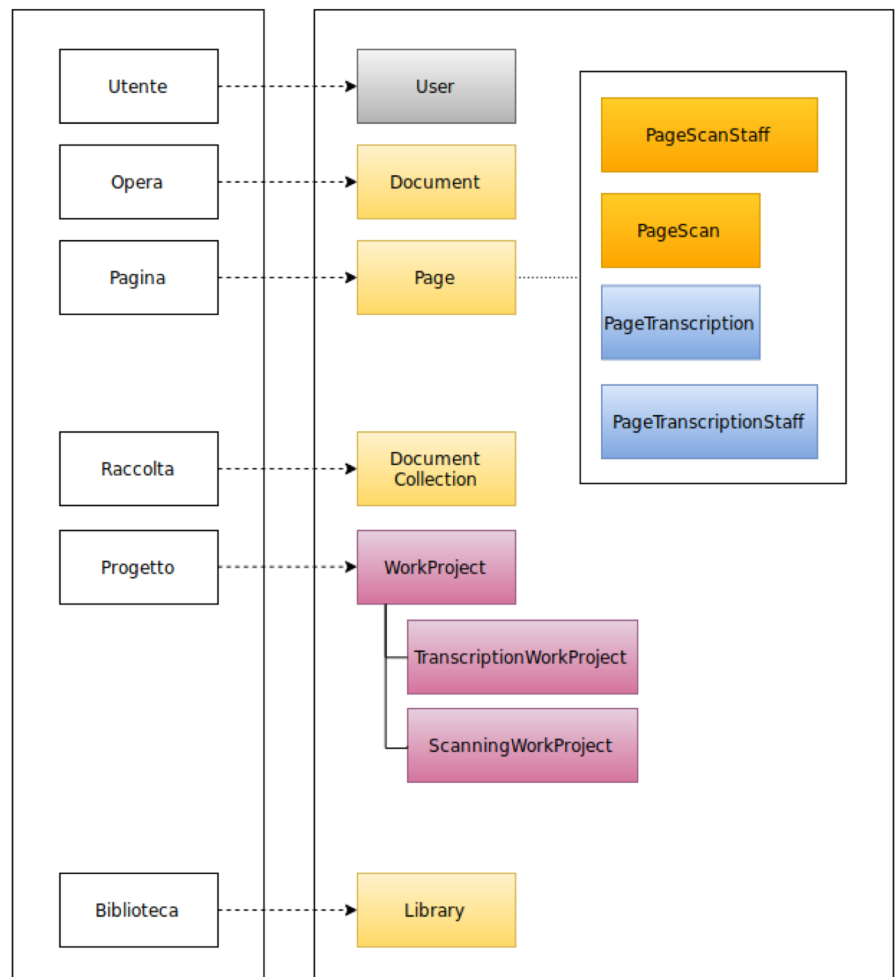
Scendendo più nel dettaglio, si possono raggruppare le funzionalità di Controller e View per sottosistemi e farvi corrispondere dei subpackages. Le funzionalità di registrazione e autenticazione sono state accorpate al sottosistema Viewer. Nel package Controller è inoltre presente un subpackage *Session Data* che permette l'astrazione del caricamento in



Schema dei package, omettendo il VO

memoria delle classi nel Model, in modo da non complicare eccessivamente la struttura del Model ma rendere il sistema scalabile.

Nel Model sono presenti le Classi derivanti dalle entità emerse nel modello di dominio, con alcune modifiche per evitare classi eccessivamente grandi, fornire una separazione tra parti logicamente distinte delle entità (nel caso delle pagine e dei progetti) e rendere più flessibile l'implementazione (anche in vista di modifiche future).

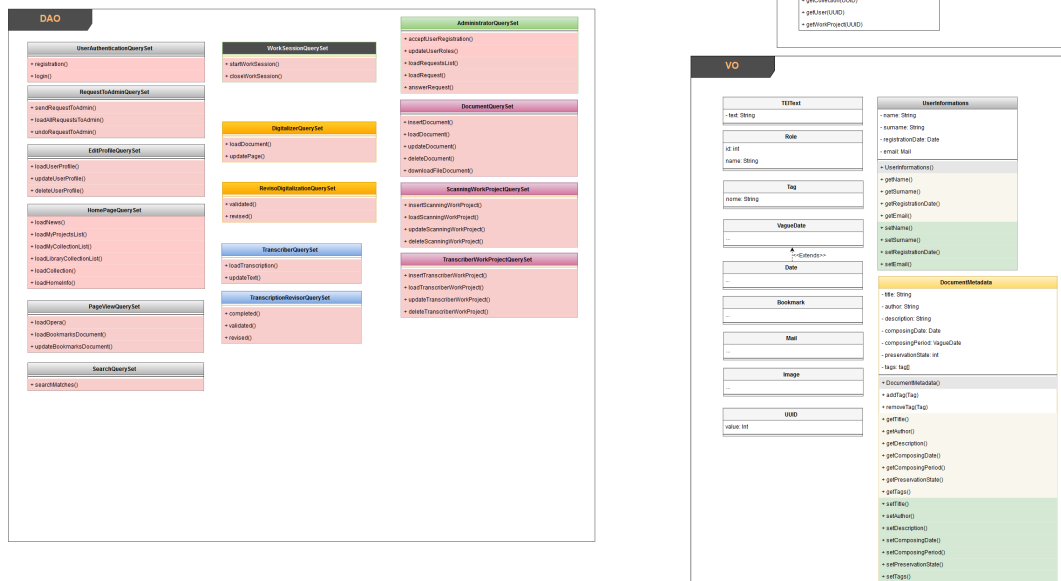
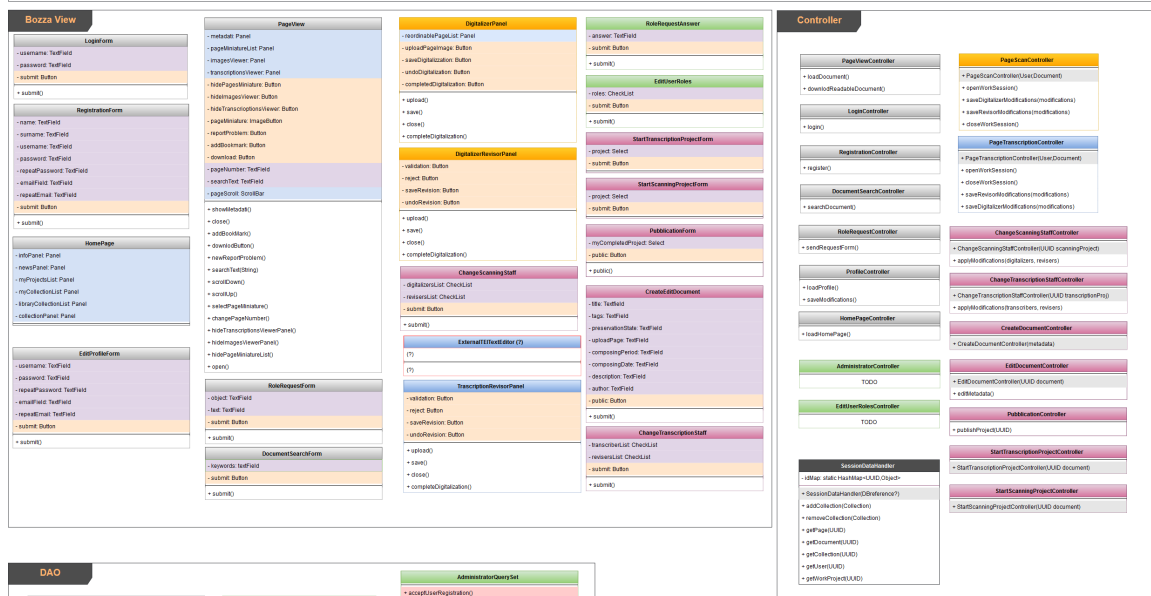
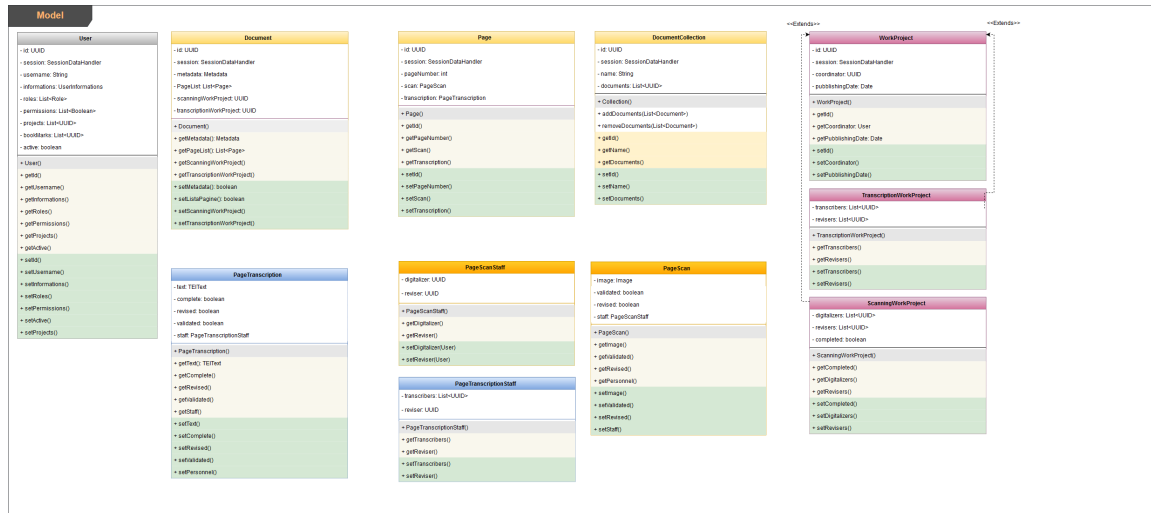


Schema delle conversione delle entità presenti nel modello di dominio nelle classi del package Model

2.3. DESCRIZIONE DELLE SCELTE E DELLE STRATEGIE ADOTTATE (Si veda 2.2)

3. SOFTWARE/OBJECT DESIGN

3.1. UML CLASS DIAGRAMS



3.2. DESCRIZIONE DEI DETTAGLI DI DESIGN SCELTI

Legenda:

Classe

Variabile istanza

Affermazioni importanti

1. Model

La classe **Document** include dentro sé tutte le informazioni riguardanti l'opera, le pagine ed i relativi progetti di digitalizzazione e trascrizione.

Nella classe **Document** infatti troviamo un oggetto **metadata** che incapsula le informazioni che caratterizzano l'opera, un oggetto **ScanningWorkProject** che include dentro sé tutti gli oggetti riguardanti il progetto di digitalizzazione dell'opera e una classe **TranscriptionWorkProject** che riassume tutte le informazioni riguardanti il progetto di trascrizione. Essendo l'opera costituita da un numero variabile di pagine, essa include dentro una lista di riferimenti ad oggetti di classe Page. Inizialmente un nuovo Document contiene solamente i metadati.

Per associare delle pagine al **Document** è necessario creare un nuovo progetto di digitalizzazione.

La classe **WorkProject** rappresenta un generico progetto. Essa comprende un attributo *coordinator* che rappresenta il responsabile del progetto ed un attributo *PublishingDate* che rappresenta la data di pubblicazione del progetto. Di default la data è inizializzata a null stando ad indicare che la relativa digitalizzazione o trascrizione non è stata ancora pubblicata, solo quando la data verrà inserita il lavoro sarà visibile al generico utente registrato.

Le classi **ScanningWorkProject** e **TranscriptionWorkProject** sono estensioni della classe **WorkProject** e si differenziano per le loro applicazioni.

Un progetto di digitalizzazione prevede la creazione di uno **ScanningWorkProject** da parte di un coordinatore e la scelta di una lista di digitalizzatori e di revisori.

I digitalizzatori creano nuove **Page** da aggiungere al **Document**. La classe **Page** incapsula al suo interno un oggetto della classe **PageScan** che rappresenta la scansione di quella pagina e una classe **PageTranscription** che rappresenta la trascrizione di quella pagina. Il digitalizzatore oltre alla **Page** crea anche la relativa **PageScan** che contiene l'immagine della scansione.

Le classi **PageScan** e **PageTranscription** sono dotate delle rispettive liste **PageScanStaff** e **PageTranscriptionStaff** che rappresentano il personale che ha lavorato a quella particolare scansione o trascrizione. Inoltre queste due classi possiedono dei flag per segnalare lo stato della revisione e l'esito della revisione. Il digitalizzatore può riordinare le **Page** cambiando il loro numero di pagine (*Page Number*), può anche eliminare e ricaricare scansioni fino a quando tutto il lavoro non è completato.

Quando i digitalizzatori hanno terminato l'intera scansione dell'opera possono aggiornare l'attributo *completed* dell'oggetto **ScanningWorkProject**, solo dopo il caricamento di tutte le immagini i revisori

potranno incominciare la revisione delle scansioni, ogni singola **PageScan** possiede un attributo *revised* per indicare se è stata revisionata ed un attributo *validated* per indicare se è stata validata da un revisore. Quando tutte le scansioni sono state validate allora l'opera potrà essere pubblicata dal coordinatore impostando automaticamente la data di pubblicazione nello **ScanningWorkProject**.

Un progetto di trascrizione prevede la creazione di un **TranscriptionWorkProject** da parte di un coordinatore sulla base di una scansione delle pagine già esistente e la scelta di una lista di trascrittori e di revisori. La classe **PageTranscription** possiede un attributo **TEIText** ed un attributo **PageScanStaff** che rappresenta il personale che ha lavorato alla singola trascrizione. Inoltre sono presenti degli attributi che segnalano lo stato della trascrizione, lo stato di revisione e l'esito della revisione. I trascrittori del progetto accedono in maniera mutualmente esclusiva alle trascrizioni delle pagine, quando una singola trascrizione viene completata viene settato l'attributo *completed* consentendone l'accesso ai revisori.

Quando un revisore termina la revisione cambia il valore dell'attributo *revised* (che indica il completamento della revisione) e quello dell'attributo *validated* (che specifica l'esito della revisione)

La trascrizione dell'intero **Document** è pubblicabile quando tutte le revisioni di ogni trascrizione sono state validate. La pubblicazione avviene impostando la data di pubblicazione.

La classe **DocumentCollection** rappresenta una raccolta di **Document** (opere).

La classe **SessionDataHandler** si occupa di gestire gli oggetti caricati dall'applicazione e di far corrispondere ad essi i corrispondenti **UUID**, gli **UUID** permettono di rappresentare oggetti all'interno di altri oggetti.

Le classi **User** e **UserInformations** rappresentano l'utente e le relative informazioni.

2. Controller

2.1 La classe **SessionDataHandler**[Controller] fa da contenitore per tutti gli oggetti del Model. **SessionDataHandler** incapsula una **HashMap<UUID,Object>**, ogni accesso ai membri della **HashMap** avviene attraverso metodi specifici per le varie classi Model (Page, Document , ecc.). **SessionDataHandler** si interfaccia con le classi del DAO per caricare automaticamente in memoria gli oggetti richiesti non presenti nella **HashMap** e rimuovere quelli non più necessari.

Nelle classi del Model ogni riferimento ad altri oggetti Model è contenuto come **UUID**, fornendo però dei metodi per restituire i riferimenti diretti in memoria (che vengono forniti attraverso il **SessionDataHandler** direttamente dai metodi delle classi, in modo completamente trasparente per le classi che invocano i metodi).

L'uso di **HashMap** in **SessionDataHandler** garantisce l'accesso in tempo costante, in modo da non compromettere eccessivamente le prestazioni del sistema ma permettendo al contempo di mantenere solo parte degli oggetti in memoria.

2.2 Il resto delle classi appartenenti al package Controller sono emersi dall'analisi entity-boundary-controller

3. Dao

Ogni classe del DAO fornisce le query per l'accesso ai dati di un particolare sotto-sistema dell'applicazione, spesso per ogni controller corrisponde una classe del DAO.

I controller infatti si servono delle query messe a disposizione dai metodi delle classi per svolgere le loro operazioni sul database. I nomi dei metodi spesso contengono parole che richiamano le operazioni tipiche che possiamo svolgere sul database, per esempio:

- insert per i metodi che aggiungono nuovi record al db.
- load per i metodi con query che caricano record dal db.
- update per i metodi che aggiornano record nel db.
- delete per i metodi che cancellano record dal db.

Corrispondenze tra i sottosistemi e le classi del DAO:

VIEWER:

UserAuthenticationQuerySet per login e registrazione.

RequestToAdminQuerySet per richiedere ruoli all'amministratore o per altre esigenze.

EditProfileQuerySet per l'aggiornamento del proprio profilo utente.

HomePageQuerySet per caricare le informazioni principali da visualizzare nella homepage.

PageViewQuerySet per caricare le pagine di un Document da visualizzare nella PageView.

SearchQuerySet per la ricerca di Document all'interno del db.

UPLOADER:

DigitalizerQuerySet per caricare e aggiornare le scansioni di un'opera.

RevisorDigitalizationQuerySet per caricare gli esiti delle revisioni delle scansioni.

TRANSCRIBER:

TranscriberQuerySet per caricare e aggiornare le trascrizioni delle pagine.

TranscriptionRevisorQuerySet per caricare gli esiti delle revisioni delle trascrizioni.

MANAGER:

DocumentQuerySet per inserire nuove opere nel db.

ScanningWorkProjectQuerySet per inserire e aggiornare i progetti di digitalizzazione.

TranscriberWorkProjectQuerySet per inserire e aggiornare i progetti di trascrizione.

ADMINISTRATOR:

AdministratorQuerySet per inserire/rimuovere/aggiornare utenti e ruoli.

WORK SESSION HANDLER

WorkSessionQuerySet contiene in particolare due metodi `startQuerySession(user, entities)` ed `endQuerySession()`. Sostanzialmente ci sono delle entità nel database che possono essere modificate solo da un utente per volta, per esempio solo un trascrittore per volta può avere accesso alla trascrizione di una pagina, o solo un digitalizzatore per volta può avere accesso all'intero set di scansioni dell'opera, stesso discorso vale per i rispettivi revisori. Dunque abbiamo scelto di inserire dei flag all'interno del db per garantire la mutua esclusione per gli accessi. Il metodo `startQuerySession(user, entities)` restituisce `true` al controller di una certa attività se l'utente può accedere a tali entities, viceversa restituisce `false` nel caso in cui la sessione è occupata già da un altro utente. Il controller comunicherà l'esito all'utente nella maniera più opportuna.