

Documento dei requisiti

1. OBIETTIVO, DEFINIZIONI, PANORAMICA, USECASE

- 1.1. La biblioteca
- 1.2. Definizioni, acronimi e abbreviazioni
- 1.3. Panoramica
 - 1.3.1. Progetto di digitalizzazione
 - 1.3.2. Progetto di trascrizione
- 1.4. Diagramma casi d'uso

2. SOTTOSISTEMI E REQUISITI FUNZIONALI

2.1. Sistema esterno

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

2.2. Sistema viewer: utenti registrati

- 2.2.1. Home
- 2.2.2. Strumento di ricerca di documenti per tag
- 2.2.3. PageView per la consultazione e download di documenti
- 2.2.4. Gestione profilo utente
- 2.2.5. Contatta amministratore

2.3. Sistema uploader: utenti digitalizzatore e revisore scansioni

- 2.3.1. Sistema di digitalizzazione.
- 2.3.2. Sistema di revisione

2.4. Sistema transcriber: utenti trascrittori e revisori trascrizioni

- 2.4.1. Text editor TEI integrato con sistema di sincronizzazione
- 2.4.2. Revisore Trascrizione

2.5. Sistema manager: utenti coordinatori

- 2.5.1. Creazione documento e relativo progetto di digitalizzazione.
- 2.5.2. Selezione dei digitalizzatori e dei revisori delle scansioni.
- 2.5.3. Pubblicazione di documenti nel catalogo
- 2.5.4. Creazione di un progetto di trascrizione
- 2.5.5. Selezione dei trascrittori e dei revisori
- 2.5.6. Pubblicazione di una trascrizione

2.6. Sistema administrator: utenti amministratori

- 2.6.1. Gestione ruoli e permessi
- 2.6.2. Backend

3. REQUISITI NON FUNZIONALI

- 3.1. Facilità d'uso
- 3.2. Flessibilità
- 3.3. Estensibilità

1 OBIETTIVO, PANORAMICA, DEFINIZIONI, USECASE

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 60.0000 carte contenenti memorie storiche della città dell'Aquila.

1.2 Definizioni, acronimi ed abbreviazioni

- Una biblioteca è un insieme di documenti.
- Una collezione è formata da documenti che condividono le stesse caratteristiche.
- Un documento (anche detto opera o manoscritto) è definito come una collezione di pagine più dei metadati, ogni documento possiede una lista di tag utili a catalogarla.

1.3 Panoramica

All'interno del processo di pubblicazione di un documento generalmente sono coinvolti i sistemi Viewer, Uploader, Trascrifer e Manager, l'utente mediante permessi accede alle funzionalità di tali sottosistemi.

Il ruolo di un utente definisce un insieme di permessi.

La gestione di ruoli e permessi è affidata agli utenti amministratori.

Sistema Viewer: accessibile dall'utente registrato

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

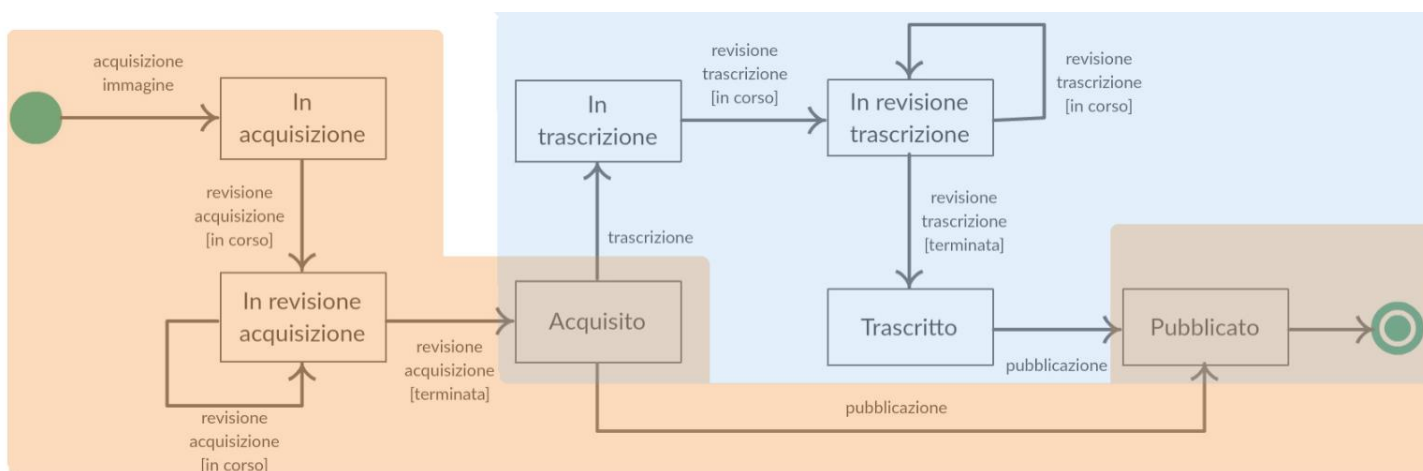
Sistema Transcriber: accessibile dall'utente trascrittore e l'utente revisore delle trascrizioni.

Sistema Manager: coordinatore del progetto.

Sistema Administrator: amministratore del sistema.

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

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



Per la pubblicazione di un documento sono sufficienti immagini e metadati generati dal un processo di digitalizzazione, ma ad essa può essere associata in aggiunta una trascrizione della stessa generata da un processo di trascrizione. Il processo di trascrizione succede sempre il processo di acquisizione del documento (eventualmente già pubblicata in passato).

Il sistema uploader gestisce il processo di acquisizione e revisione.

Il sistema transcriber gestisce il processo di trascrizione e revisione.

Ogni utente trascrittore lavora su un certo insieme di pagine di un documento, inoltre ogni trascrittore ha un livello di esperienza che va da 1 a 5.

L'intero processo di pubblicazione di un documento è gestito da un coordinatore che è l'unico ad avere accesso al sistema Manager che permette di pubblicare l'output prodotto dal sistema Uploader e dal sistema Transcriber.

1.3.1 Processo di digitalizzazione

Il sistema Uploader esegue la digitalizzazione del documento (*in acquisizione*).

Il sistema Uploader supervisiona le scansioni (*revisione acquisizione*).

Il sistema Manager valida l'acquisizione (*acquisito*) e può procedere all'eventuale pubblicazione del documento (*pubblicato*).

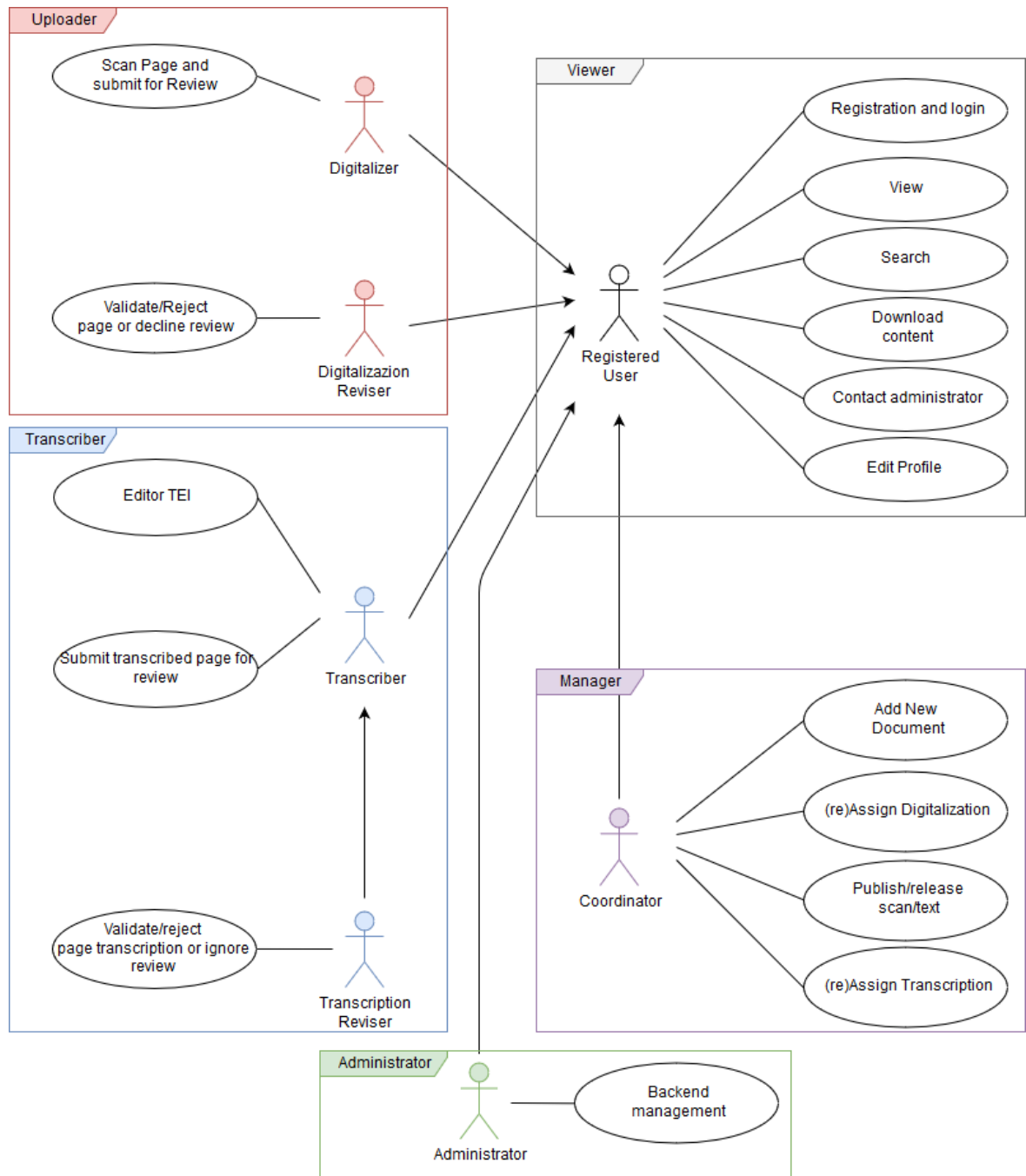
Osservazione: Un documento digitalizzato può essere pubblicato senza trascrizione.

1.3.2 Processo di trascrizione

Il coordinatore attraverso il sistema Manager apre un processo di trascrizione ed assegna le pagine del documento ai rispettivi trascrittori (*in trascrizione*). Le trascrizioni sono digitate manualmente attraverso un text editor TEI integrato. Quindi ogni immagine(pagina) avrà il corrispondente testo associato. Più trascrittori possono lavorare sulla stessa pagina, è necessario sincronizzare le modifiche.

Il sistema Manager supervisiona le scansioni (*revisione trascrizione*).

Il sistema Manager valida l'acquisizione(*acquisito*) e può procedere all'eventuale pubblicazione del documento (*pubblicato*) o semplicemente di un aggiornamento dell stesa se questa è stata già pubblicata precedentemente.



2 SOTTOSISTEMI E REQUISITI FUNZIONALI

2.1 SISTEMA ESTERNO: Form per l'autenticazione e la registrazione degli 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.
- Iscrizione supervisionata: l'utente invia una richiesta di iscrizione e l'amministratore decide se accettare o respingere l'utente.

2.2 Sistema viewer: utenti registrati

Tutti gli utenti registrati sono utenti Viewer.

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 i documenti più recenti.
- La schermata home elenca i segnalibri (ed eventuali progetti) dell'utente.
- La schermata home possiede una bacheca news.

2.2.2 Strumento di ricerca di documenti per tag(searchbar)

- Il sistema possiede uno strumento di ricerca facilmente accessibile che consente di cercare un documento 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 documenti consultabili dall'utente.

2.2.3 PageViewer per la consultazione e download di documenti

- Selezionando un documento si visualizza una vetrina con info e dettagli.
- Avviando il visualizzatore dei documenti, esso si presenta similmente a un lettore pdf che consente di scorrere le pagine contenenti le scansioni del documento.
- Eventualmente se presente una trascrizione il paginatore presenta una doppia schermata che consente di visualizzare e scorrere parallelamente scansioni e trascrizioni in due pannelli distinti.
- E' possibile aggiungere dei segnalibri alle pagine di un documento.
- Solo utenti con particolari privilegi possono scaricare il documento visualizzata.

2.2.4 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.

2.2.5 Contatta amministratore

L'utente viewer può contattare l'amministratore per richiedere particolari ruoli o permessi, e per segnalare eventuali problemi.

2.3 Sistema uploader: utenti digitalizzatore e revisore scansioni

2.3.1 Digitalizzatore

Un qualsiasi utente può diventare un digitalizzatore sotto nomina dell'amministratore e non possono fare 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.

Un digitalizzatore entra a far parte di un progetto su richiesta del coordinatore. Il digitalizzatore mediante apposita schermata può caricare nuove scansioni da aggiungere al documento.

2.3.2 Revisore scansioni

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 il documento può essere segnalata al coordinatore come revisionata.

2.4 Sistema transcriber: utenti trascrittori e revisori trascrizioni

2.4.1 Text Editor integrato

Sistema TEI per la trascrizione delle pagine. Ogni pagina di un documento in trascrizione viene assegnata ad un trascrittore dal coordinatore. Un trascrittore può visualizzare una scansione per volta affiancata dall'editor TEI.

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, essi possono convalidare le pagine della trascrizione o richiederne la modifica. Quando tutte le pagine sono state convalidate la trascrizione del documento può essere pubblicata dal coordinatore.

2.4 Sistema manager: utenti coordinatori

Il sistema manager consente la creazione e la gestione di processi di digitalizzazione e processi di trascrizione di documenti e la gestione del personale che collabora al progetto.

Il ruolo di coordinatore viene assegnato dall'amministratore.

2.5.1 Creazione di un progetto di digitalizzazione

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

2.5.2 Selezione dei digitalizzatori e dei revisori delle scansioni

La selezione del personale per la digitalizzazione e la revisione avviene da una lista di utenti digitalizzatori.

2.5.3 Pubblicazione di documenti nel catalogo

Il coordinatore può accedere a tutto il materiale del progetto e osservarne l'evoluzione.

L'utente coordinatore attenderà la digitalizzazione del documento e la segnalazione del documento come revisionata, a quel punto il coordinatore potrà

decidere se pubblicare il documento oppure respingerla e richiedere una nuova revisione.

2.5.4 Creazione di un progetto di trascrizione

Il coordinatore può avviare la trascrizione di un documento già pubblicata nel sistema attraverso un'apposita interfaccia.

2.5.5 Selezione dei trascrittori e dei revisori

La scelta del personale per la trascrizione avviene da una lista di trascrittori.

1.2 Sistema administrator: utenti amministratori

L'amministratore riceve comunicazioni dagli utenti e gestisce ruoli e permessi.

3 Requisiti non funzionali

3.1 Facilità d'uso

L'applicazione è facilmente accessibile a personale non particolarmente competente in ambito informatico.

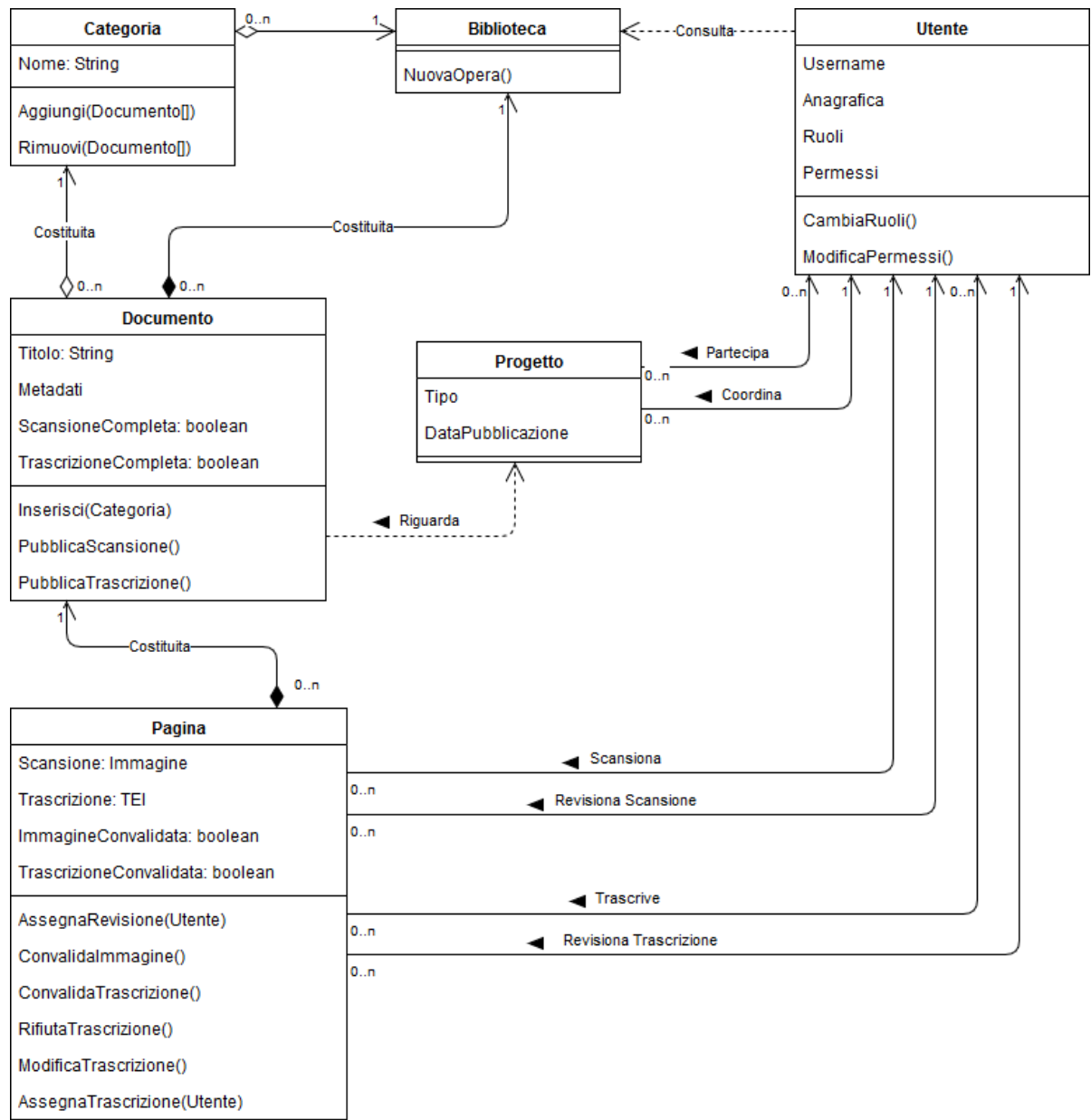
3.2 Flessibilità

L'applicazione è dotata di una barra principale che consente di accedere rapidamente alle diverse schermate secondo i permessi dell'utente.

3.4 Estendibilità

L'applicazione è estendibile al modello client-server.

4 Modello di dominio



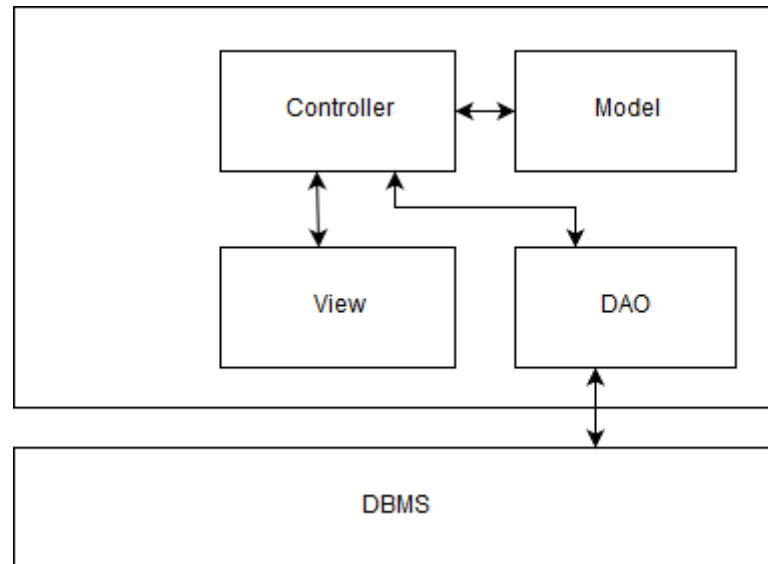
5 Boundary e controller

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> |
|-------------------|--|
| AdminPanel | AdministrationController EditUserController |
| Home | HomePageController PageViewController |
| LoadScan | LocalSession PageScanController |
| LoginScene | LoginController |
| PasswordRecovery | PasswordRecovery |
| NewDocument | createDocumentController |
| PageTranscription | localSession |
| Registration | registrationController |
| ScanRevisor | LocalSession PageScanController ScanningProjectController |
| ShowDocument | LocalSession |
| UserProfile | UserProfileController |
| Container | LocalSession LoginController |
| EditUserProfile | AdministrationController EditUserController LocalSession RoleController |
| Home | HomePageController PageViewController |
| ManageProject | LocalSession PageScanController PageTranscriptionController RoleController ScanningProjectController TranscriptionProjectController |
| Trascription | LocalSession PageScanController PageTrascriptionControlle |
| UserProfile | AdministrationController EditUserController LocalSession |

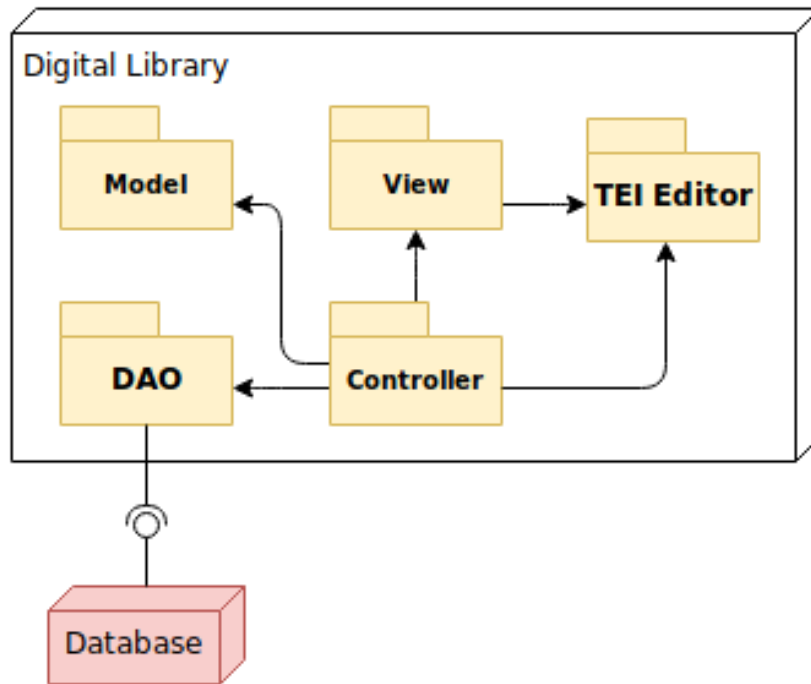
6 Descrizione dell'architettura

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.



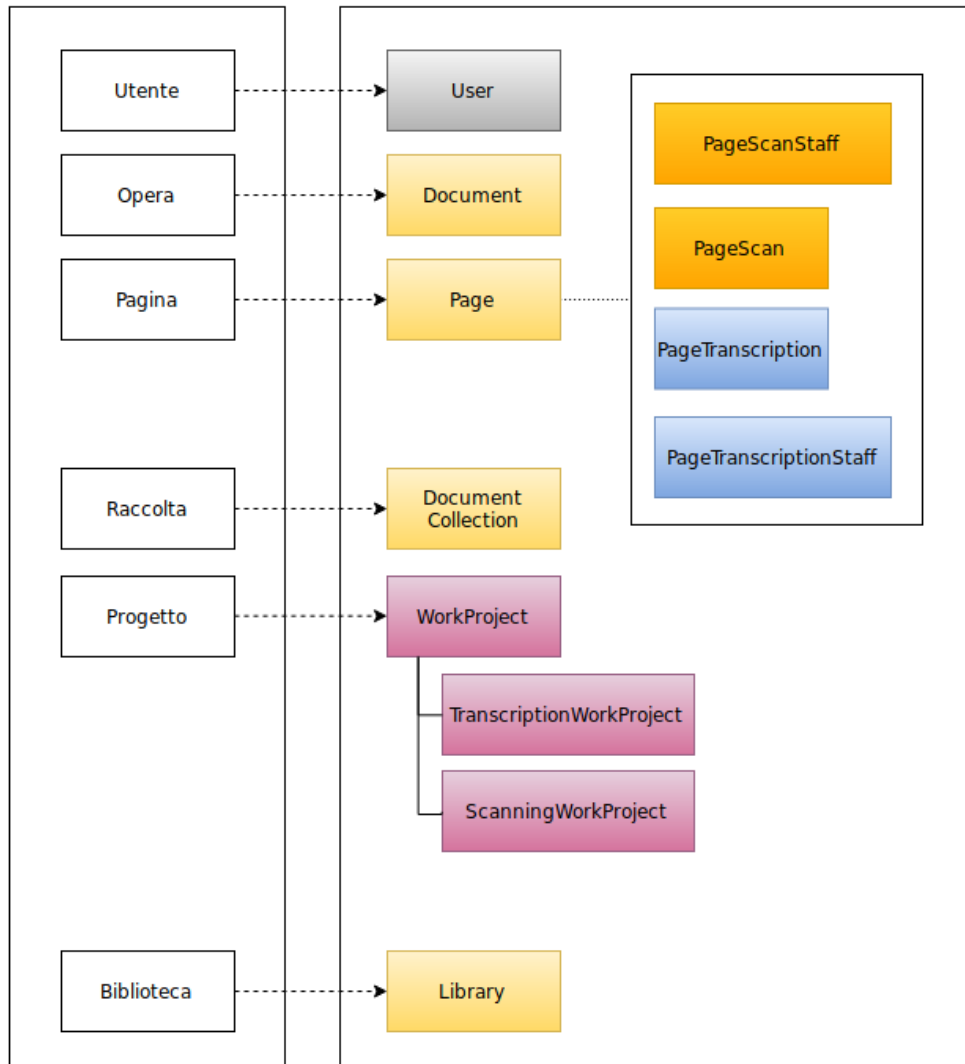
Semplice schematizzazione della suddivisione logica del sistema da cui derivano le scelte di suddivisione in

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.



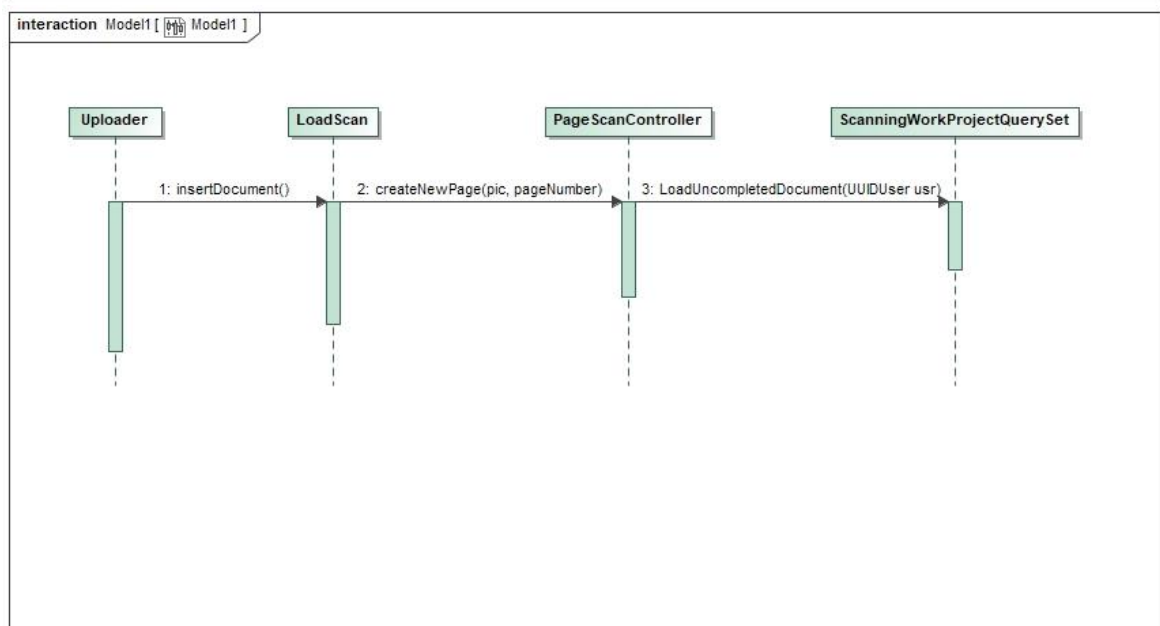
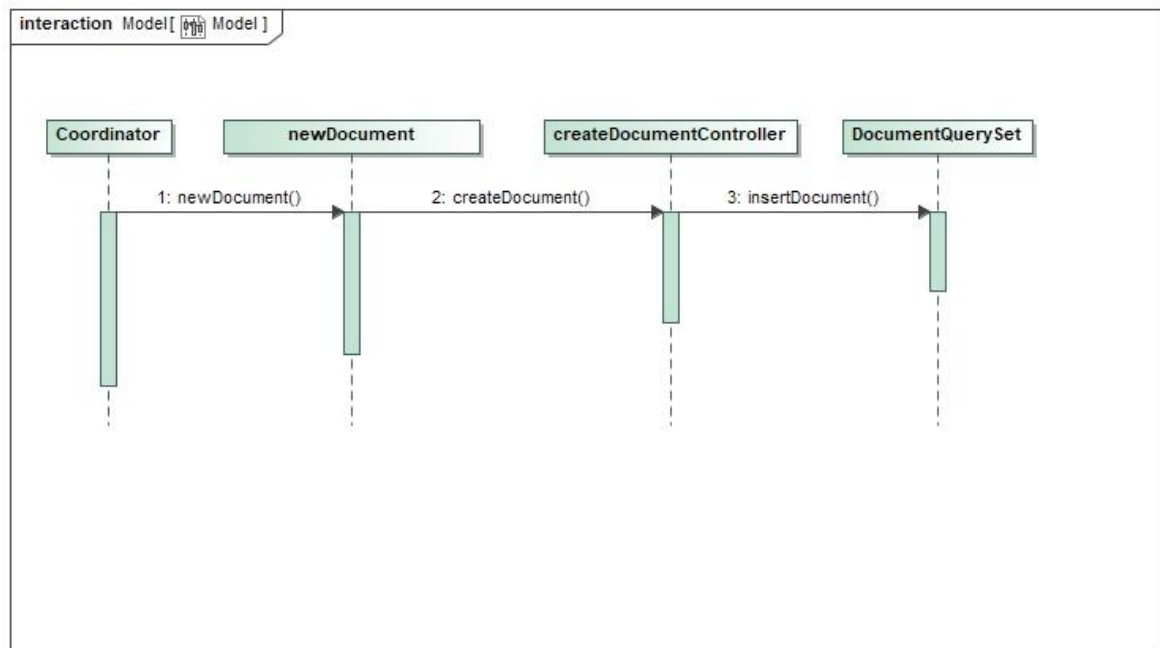
Schema dei package, omettendo il VO

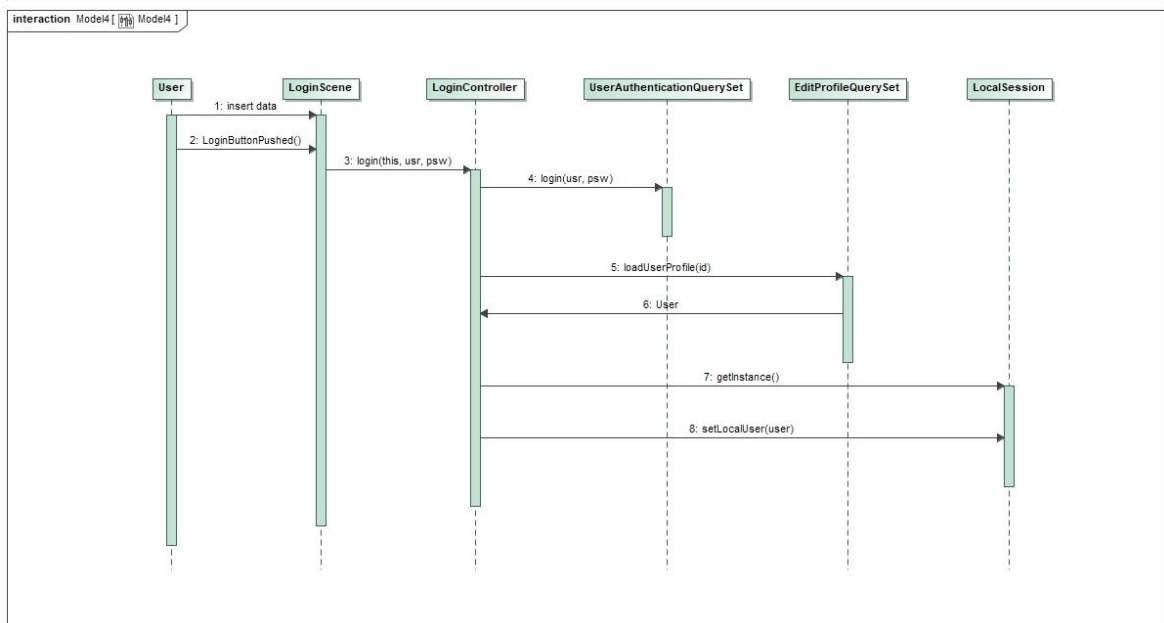
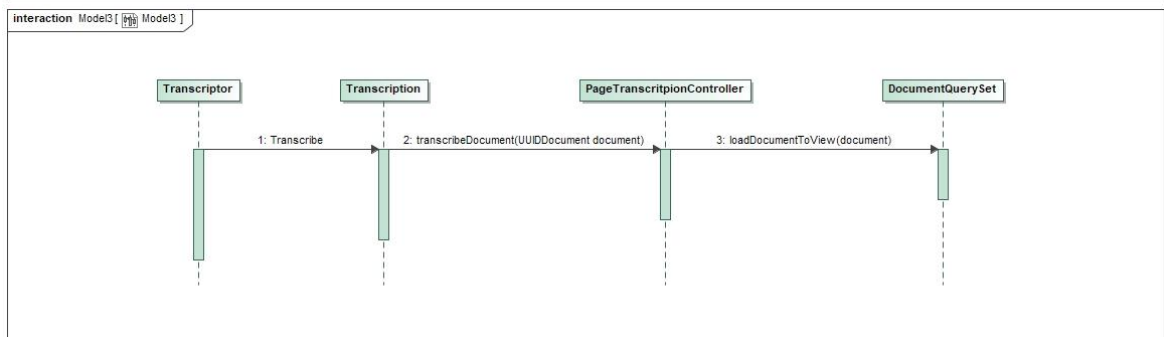
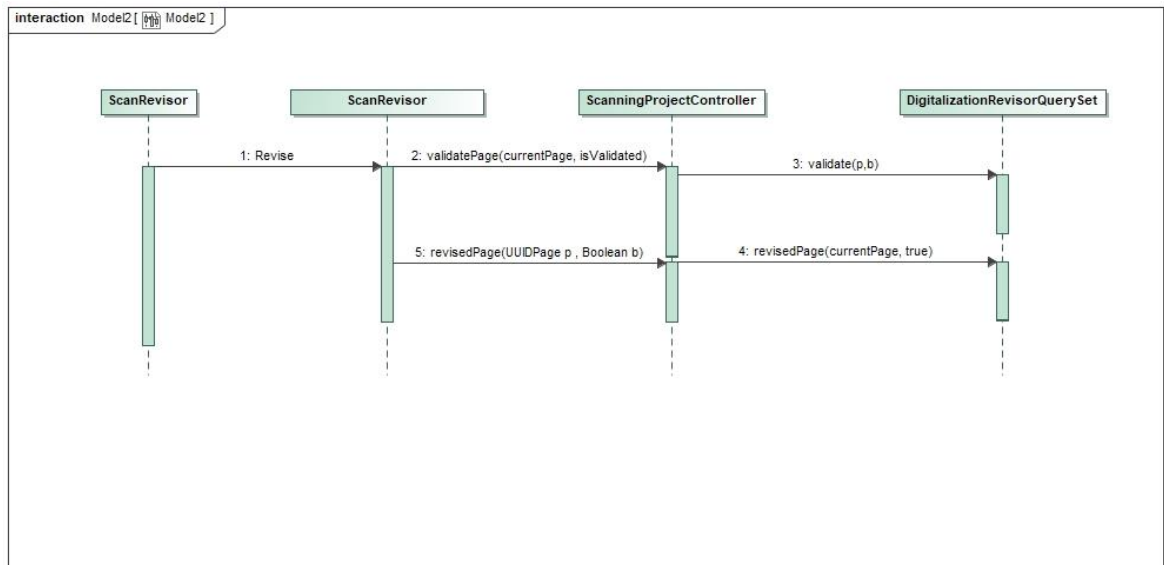
Nel Model sono presenti le Classi derivanti dalle entità emerse nel modello di dominio, con alcune modifiche per evitare classi eccessivamente grandi, fornire una separazione tra parti logicamente distinte delle entità (nel caso delle pagine e dei progetti) e rendere più flessibile l'implementazione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



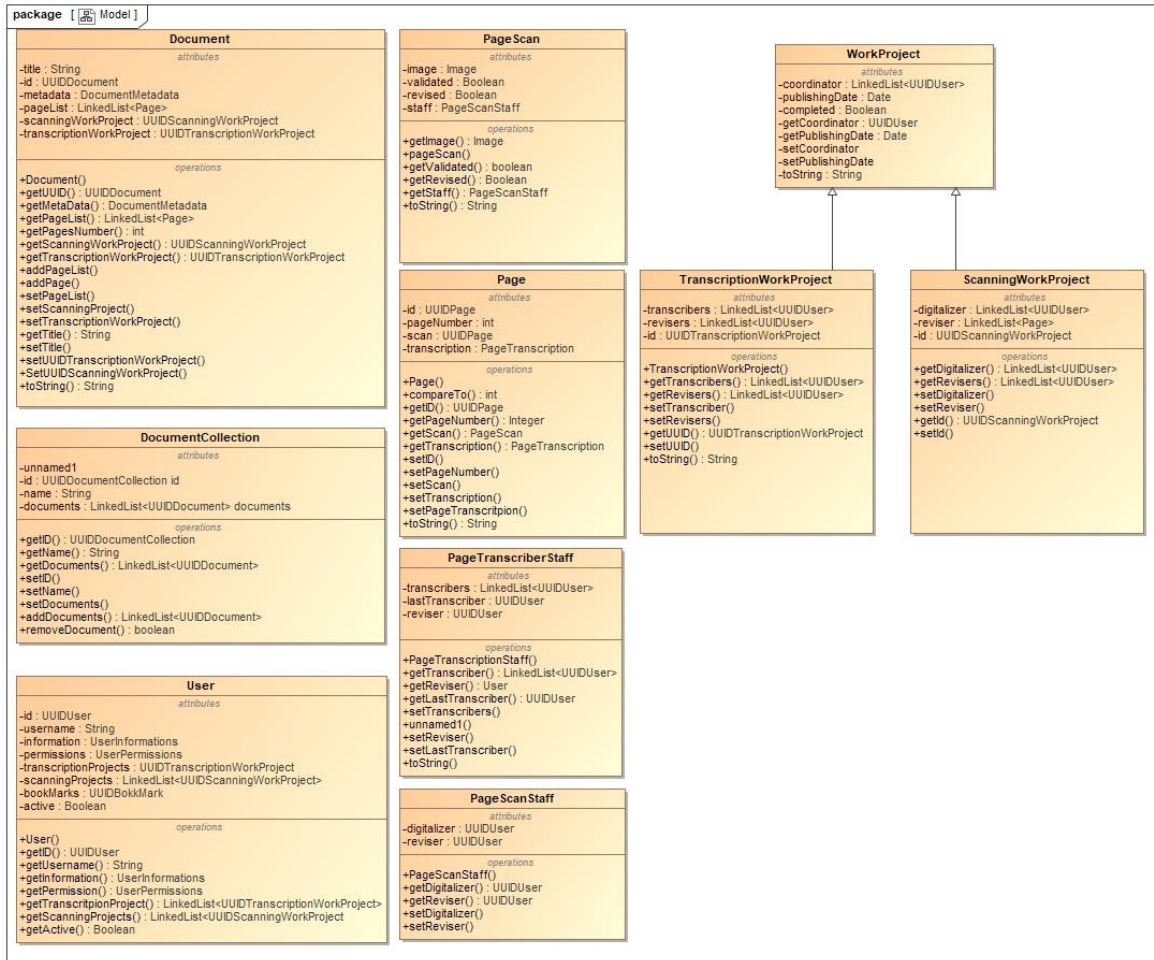
Schema delle conversione delle entità presenti nel modello di dominio nelle classi del package Model


7 Sequence diagram





8 Class diagram



package [ View]

Container

attributes

- newContent: BorderPane
- topbar: TabPane
- content: BorderPane
- logout: Tab
- map: TreeMap<String, Pair<String, String>>

operations

- +genFaviconURL(): String
- +initButtonLink(): void
- +initLogoutLink(): void
- +loadScene(): void
- +initialize(): void

RootCss

operations

- +getCss(): String

Home

attributes

- topbar: AnchorPane
- aProjects: VBox
- lProjects: VBox
- bookmarks: VBox
- news: VBox
- collections: VBox
- transcriptionTabHome: Tab
- scanningTabHome: Tab
- documentCollections: Accordion
- searchBar: TextField

operations

- +initialize(): void
- +initSearchBar(): void
- +loadScanningProjects(): void
- +loadTranscriptionProjects(): void
- +loadBookmarks(): void
- +loadNews(): void
- +loadCollections(): void

LoadScan

attributes

- flechooser: Button
- scanList: VBox
- documentList: ChoiceBox<Entry<UUIDocument, String>>
- pageTable: TableView<Rows>
- number: TableColumn<Rows, String>
- image: TableColumn<Rows, Image>
- revised: TableColumn<Rows, String>
- validated: TableColumn<Rows, String>
- pages: ObservableList<Rows>

operations

- +initialize(): void
- +initLoadScanTitle(): void
- +initFileChooser(): void
- +prepareDocument(): void
- +initPageTable(): void
- +setToOpenDocumentFromScanningProject(): void
- +setToOpenDocument(): void
- +imageUpload(): void
- +askForPageLumber(): Integer
- +isPageLumberAlreadyPresent(): boolean
- +insertDocument(): void
- +initLoadDocument(): void
- +preLoadDocument(): void
- +loadDocument(): void

level

attributes

- transcriberLevel: int
- level: Label
- name: TextField
- surname: TextField
- email: TextField
- label: levelT

ScanningProject

attributes

- topbar: AnchorPane

operations

- +initialize(): void

SceneController

attributes

- scene: Scene
- stage: Stage
- sceneName: String

operations

- +start(): void
- +main(): void
- +loadScene(): void
- +unnamed1(): void

DocumentProperties

attributes

- toShowDocument
- documentInfoController: DocumentInfoController
- title: Label
- author: Label
- composition: Label
- description: TextArea
- scanningComplete: JFXCheckBox
- transcriptionComplete: JFXCheckBox
- categories: TextArea
- tags: TextArea

operations

- +initialize(): void
- +loadInfo(): void
- +loadProjectStatus(): void
- +loadCategories(): void
- +getToShowDocument(): UUIDocument
- +setToShowDocument(): void

ManageProject

attributes

- topbar: AnchorPane
- newDocumentButton: Button
- documentTable: TableView<DocumentRow>
- Document: TableColumn<DocumentRow, String>
- Transcription_PRJ: TableColumn<DocumentRow, String>
- Scanning_PRJ: TableColumn<DocumentRow, String>
- pane: TabPane
- propertiesTab: Tab
- documentProperties: BorderPane
- transcriptionTab: Tab
- scanningTab: Tab
- transcriptionTabPane: TabPane
- scanningTabPane: TabPane
- transcriptionPagesTab: Tab
- scanningPagesTab: Tab
- transcriptionTranscribersTab: Tab
- scanningDigitalizersTab: Tab
- manageContainer: AnchorPane
- transcriptionPagesAnchor: AnchorPane
- scanningPagesAnchor: AnchorPane
- transcriptionTranscriberAnchor: AnchorPane
- scanningDigitalizerAnchor: AnchorPane
- rows: AnchorPane
- transcriptionStaff: TableView<StaffRow>
- usernameTranscriber: TableColumn<StaffRow, String>
- roleTranscriber: TableColumn<StaffRow, String>
- transcriptionProjectStaff: ObservableList<StaffRow>
- transcriptionLog: TableView<PageTranscriptionLog, String>
- pageNumber: TableColumn<PageTranscriptionLog, String>
- transcriptionReviser: TableColumn<PageTranscriptionLog, String>
- transcriptionRevised: TableColumn<PageTranscriptionLog, String>
- transcriptionValidated: TableColumn<PageTranscriptionLog, String>
- transcriptionTranscriber: TableColumn<PageTranscriptionLog, String>
- listLog: ObservableList<PageTranscriptionLog>
- scanningStaff: TableView<StaffRow>
- usernameDigitalizer: TableColumn<StaffRow, String>
- roleDigitalizer: TableColumn<StaffRow, String>
- scanningProjectStaff: ObservableList<StaffRow>
- scanningLog: TableView<PageScanningLog>
- pageScanLumber: TableColumn<PageScanningLog>
- scanningReviser: TableColumn<PageScanningLog, String>
- scanningRevised: TableColumn<PageScanningLog, String>
- scanningValidated: TableColumn<PageScanningLog, String>
- scanningDigitalizer: TableColumn<PageScanningLog, String>
- listScanLog: ObservableList<PageScanningLog, String>
- addDigitalizerButton: Button
- addTranscriberButton: Button
- addReviserButton: Button
- selectedDocumentScanningProject: UUIDScanningWorkProject
- selectedDocumentTranscriptionProject: UUIDTranscriptionWorkProject
- addTranscriberEvent
- loadFC

operations

- +initialize(): void
- +initPane(): void
- +initNewDocumentButton(): void
- +tableDocumentFilter(): void
- +rowClick(): void
- +addDigitalizerEvent(): void
- +addReviserEvent(): void
- +addReviserEvent(): void
- +loadClickedDocumentProjects(): void
- +userRowClick(): void
- +loadTranscriptionPagesLog(): void
- +loadScanningPagesLog(): void
- +loadTranscriptionProjectStaff(): void
- +loadScanningProjectStaff(): void

ShowDocument

attributes

- isEmpty: boolean
- backButton: Button
- pageList: ListView<VBox>
- currentPage: ImageView
- transcription: CodeArea
- previous: Button
- next: Button
- top: Button
- bottom: Button
- number: Label
- message: Label

operations

- +initialize(): void
- +loadPageList(): void
- +viewPage(): void
- +initBackButton(): void
- +initNavigationButton(): void
- +updatePageNumber(): void
- +updatePage(): void
- +updateTranscription(): void
- +setTranscriptionText(): void
- +initKeyboardNavigation(): void
- +next(): void
- +previous(): void
- +top(): void
- +bottom(): void
- +initTranscription(): void

PasswordRecovery

attributes

- email: TextField
- back: Button
- register: Button

operations

- +initialize(): void
- +initRegisterButton(): void
- +initBackButton(): void
- +validEmail(): boolean
- +alert(): void

Registration

attributes

- username: TextField
- password1: PasswordField
- password2: PasswordField
- name: TextField
- email: TextField
- back: Button
- register: Button

LoginScene

attributes

- Font: LoginScene
- x41: Color
- x1: Font
- x2: Color
- loginTitle: Label
- username: TextField
- password: PasswordField
- login: Button
- recoverButton: Button
- registerButton: Button
- unnamed1
- message: Label

operations

- +go(): void
- +loginButtonPushed(): void
- +initialize(): void
- +initEffects(): void
- +displayMessage(): void

PageTranscription

attributes

- isEmpty: bool
- backButton: Button
- saveButton: Button
- pageList: ListView<VBox>
- currentPage: ImageView
- transcription: CodeArea
- previous: Button
- next: Button
- top: Button
- number: Label
- message: Label

operations

- +initialize(): void
- +loadPageList(): void
- +viewPage(): void
- +initBackButton(): void
- +initNavigationButton(): void
- +updatePageLumber(): void
- +updatePage(): void
- +updateTranscription(): void
- +updateTranscriptionText(): void
- +initKeyboardNavigation(): void
- +next(): void
- +previous(): void
- +top(): void
- +bottom(): void
- +initTranscription(): void

AdminPanel

attributes

- users: TableView<UserRow>
- users1: TableView<UserRow>
- userRows: ObservableList<UserRow>
- userRows1: ObservableList<UserRow>
- userID: TableColumn<UserRow, UUIDUser>
- username: TableColumn<UserRow, UUIDUser>
- name: TableColumn<UserRow, UUIDUser>
- surname: TableColumn<UserRow, UUIDUser>
- email: TableColumn<UserRow, UUIDUser>
- userD1: TableColumn<UserRow, UUIDUser>
- surname1: TableColumn<UserRow, String>
- email: TableColumn<UserRow, String>
- pending: Tab
- old: Tab
- requests: TableView<RequestRow>
- requestRows: ObservableList<RequestRow>
- requestID: TableColumn<RequestRow, UUIDRequest>
- request1surname: TableColumn<RequestRow, String>
- requestObject: TableColumn<RequestRow, String>
- requests2: TableView<RequestRow>
- requestRows2: ObservableList<RequestRow>
- requestID2: TableColumn<RequestRow, UUIDRequest>
- requestUserame2: TableColumn<RequestRow, String>
- requestObject2: TableColumn<RequestRow, String>

operations

- +initialize(): void
- +initActiveUserList(): void
- +initInactiveUserList(): void
- +initUserRowClick(): void
- +initUserRowClick1(): void
- +refreshRow(): void
- +initRequests(): void
- +initRequestTab(): void
- +initRequestTab2(): void
- +userRequestDialog(): void

prepareDocument

attributes

- scanList: VBox
- documentList: ChoiceBox<Entry<UUIDocument, String>>
- pageTable: TableView<Rows>
- number: TableColumn<Rows, String>
- image: TableColumn<Rows, Image>
- revised: TableColumn<Rows, String>
- validated: TableColumn<Rows, String>
- Pages: ObservableList<Rows>

operations

- +prepareDocument(): void
- +initPageTable(): void
- +setToDocumentFromTranscriptionProject(): void
- +setToOpenDocument(): void
- +insertDocument(): void
- +initLoadDocument(): void
- +preLoadDocument(): void
- +loadDocument(): void

XmlSyntaxHighlight

attributes

- XML_TAG: Pattern
- ATTRIBUTES: Pattern
- GROUP_OPEN_BRACKET: int
- GROUP_ELEMENT_NAME: int
- GROUP_ATTRIBUTES_SECTION: int
- GROUP_CLOSE_BRACKET: int
- GROUP_ATTRIBUTE_NAME: int
- GROUP_EQUAL_SYMBOL: int
- GROUP_ATTRIBUTE_VALUE: int

operations

- +computeHighlighting(): StyleSpans<Collection<String>>

NewDocument

attributes

- roleList: MenuButton
- label: Label
- intext: TextField
- next: Button
- container: Pane
- definita: Button
- inarea: TextArea
- choiceBox: ChoiceBox<String>
- choiceBox1: ChoiceBox<String>
- choiceBox2: ChoiceBox<String>
- year1: TextField
- year2: TextField
- int c
- title: String
- author: String
- description: String
- yearvar: String
- year1var: String
- year2var: String
- preservState: String
- existingTags: HashMap<Integer, Tag>
- notExistingTags: HashMap<Integer, Tag>

operations

- +initialize(): void
- +initNextButton(): void
- +caseZero(): void
- +caseOne(): void
- +caseTwo(): void
- +caseThree(): void
- +caseFour(): void
- +caseFive(): void
- +caseSix(): void
- +caseSeven(): void
- +caseEight(): void
- +caseNine(): void
- +caseTen(): void
- +finalCase(): void

ScanRevisor

attributes

- scanList: VBox
- documentList: ChoiceBox<Entry<UUIDocument, String>>
- loadDocumentButton: Button
- pageTable: TableView<Rows>
- number: TableColumn<Rows, String>
- image: TableColumn<Rows, ImageView>
- revised: TableColumn<Rows, String>
- validated: TableColumn<Rows, String>
- pages: ObservableList<Rows>
- pageContainer: Pane
- closePageContainerButton: Button
- pageimg: ImageView
- acceptButton: Button
- rejectButton: Button
- confirmButton: Button
- commentArea: TextArea
- currentPage: UUIDPage
- isValidated: boolean

operations

- +initialize(): void
- +initCommentArea(): void
- +initAcceptButton(): void
- +initRejectButton(): void
- +initConfirmButton(): void
- +initClosePageContainerButton(): void
- +initPageContainer(): void
- +insertDocument(): void
- +initLoadDocumentButton(): void
- +initRowClick(): void

HomeController

attributes

- news : HashMap<UUIDocument, String[]>
- categories : HashMap<UUIDocumentCollection, String>
- myCollection : HashMap<UUIDocument, String[]>
- myPrj : HashMap<UUIDocumentCollection, String[]>
- mySPj : HashMap<UUIDocumentCollection, String[]>
- isDocumentInCollections : LinkedList<String[]>

operations

- +loadHomePage() : void
- +loadDocumentInCollections() : void
- +loadCategories() : void
- +loadMyCollection() : void
- +loadMyTranscriptionProjects() : void
- +loadMyScanningProjects() : void
- +getViews() : HashMap<UUIDocument, String[]>
- +getCategories() : HashMap<UUIDocumentCollection, String>
- +getMyCollection() : HashMap<UUIDocument, String[]>
- +setViews() : void
- +getMyTPrj() : HashMap<UUIDocumentCollection, String[]>
- +setMySPRJ() : void
- +checkIsDigitalizer() : boolean
- +checkIsTranscriber() : boolean
- +getListOfDocumentInCollections() : LinkedList<String[]>
- +setListOfDocumentInCollections() : void

PageScanController

attributes

- uncompletedDocument : HashMap<UUIDocument, String>
- currentDocument : UUIDocument
- currentDocumentPages : LinkedList<Page>
- scanningLog : LinkedList<Page>

operations

- +PageScanController() : void
- +openWorkSession() : void
- +loadScanningLog() : void
- +saveDigitalizerModifications() : void
- +saveRevisorModifications() : void
- +closeWorkSession() : void
- +loadUncompletedDocument() : void
- +getUncompletedDocument() : HashMap<UUIDocument, String>
- +setUncompletedDocument() : void
- +loadNewDocumentPages() : void
- +loadUncompletedDocumentPagesFilters() : void
- +getCurrentDocumentPages() : LinkedList<Page>
- +updatePageNumber() : void
- +setCurrentDocument() : void
- +createNewPage() : UUIDPage
- +getScanningLog() : LinkedList<Page>
- +setScanningLog() : void
- +loadNewDocumentPagesOnlyToRevise() : void

SearchController

operations

- +searchByAuthorTitle() : LinkedList<UUIDocument>
- +searchByTag() : LinkedList<UUIDocument>

TranscriptionProjectController

attributes

- Prj : TranscriptionWorkProject
- getCoordinatedTranscriptionProject : HashMap<UUIDTranscriptionWorkProject, String[]>

operations

- +getCoordinatedTranscriptionProject() : HashMap<UUIDTranscriptionWorkProject, String[]>
- +setCoordinatedTranscriptionProject() : void
- +loadCoordinatedTranscriptionProject() : void
- +getTPrj() : TranscriptionWorkProject
- +setTPrj() : void
- +loadTranscriptionProject() : void
- +getUserProfile() : User
- +setTranscriptionComment() : void
- +getAvailableTranscribers() : void
- +getTranscriptionComment() : void
- +getAvailableRevisers() : void

ScanningProjectController

attributes

- SPj : ScanningWorkProject
- coordinatedScanningProject : HashMap<UUIDScanningWorkProject, String[]>

operations

- +validatePage() : void
- +revisedPage() : void
- +getCoordinatedScanningProject() : HashMap<UUIDScanningWorkProject, String[]>
- +loadCoordinatedScanningProject() : void
- +getSPj() : ScanningWorkProject
- +setSPj() : void
- +loadScanningProject() : void
- +setScanningComment() : void
- +getUserProfile() : User
- +addDigitalizerEvent() : void

PageTranscriptionController

attributes

- currentDocumentPages : LinkedList<Page>
- currentDocument : UUIDocument
- transcriptionLog : LinkedList<Page>

operations

- +loadTranscriptionLog() : void
- +getTranscriptionLog() : LinkedList<Page>
- +setTranscriptionLog() : void
- +getCurrentDocumentPage() : LinkedList<Page>
- +setCurrentDocumentPages() : void
- +setCurrentDocument() : UUIDocument
- +setCurrentDocument() : void
- +loadNewDocumentPages() : void
- +loadNewDocumentPagesOnlyNotRevised() : UUIDocument

PageViewController

operations

- +showDocument() : void
- +downloadReadableDocument() : void

CreateDocumentController

operations

- +createDocument() : UUIDocument

ImageUploader

operations

- +ImageUploader() : void
- +uploadImage() : String

getAvailableTag

operations

- +DocumentInfoController() : void
- +getAvailableTags() : void
- +setAvailableTagsToDocument() : void
- +UUIDocumentMetadata getUUIDocumentMetadata() : UUIDocument
- +getAvailableTags() : HashMap<Integer, Tag>

RoleController

operations

- +sendRequestForm() : void
- +controlUserPermission() : boolean
- +addTranscriberUserInTranscriptionProject() : void
- +addReviserUserInTranscriptionProject() : void
- +removeTranscriberUserInTranscriptionProject() : void
- +removeReviserUserInTranscriptionProject() : void
- +addDigitalizerUserInScanningProject() : void
- +addReviserUserInScanningProject() : void
- +removeDigitalizerUserInScanningProject() : void
- +removeReviserUserInScanningProject() : void
- +changeTranscriptionProjectCoordinator() : void
- +changeScanningProjectCoordinator() : void
- +showCoordinatorUsers() : HashMap<UUIDUser, String>
- +removeUserFromUIProject() : void
- +replaceCoordinator() : void

EditUserController

attributes

- editingUser : UUIDUser
- toEditUser : User

operations

- +getEditingUser() : UUIDUser
- +getToEditUser() : User
- +canEdit() : boolean
- +canEditPermissions() : boolean
- +setModifications() : boolean
- +commitModifications() : void
- +setEditingUser() : void
- +setToEditUser() : void
- +setToEditUser() : void
- +startEditing() : void

LoginController

operations

- +login() : void
- +loginDB() : UUIDUser
- +logout() : void

LocalSession

attributes

- localUser : User
- topBarButton : ArrayList<String>
- openedDocument : Document
- instance : LocalSession

operations

- +loadImage() : Image
- +getOpenedDocument() : Document
- +setOpenedDocument() : void
- +getInstance() : LocalSession
- +getLocalUser() : User
- +getLocalUserInfo() : UserInformations
- +loadTopBarButton() : void
- +setLocalUser() : void
- +getTopBarButton() : List<String>
- +clear() : void

RegistrationController

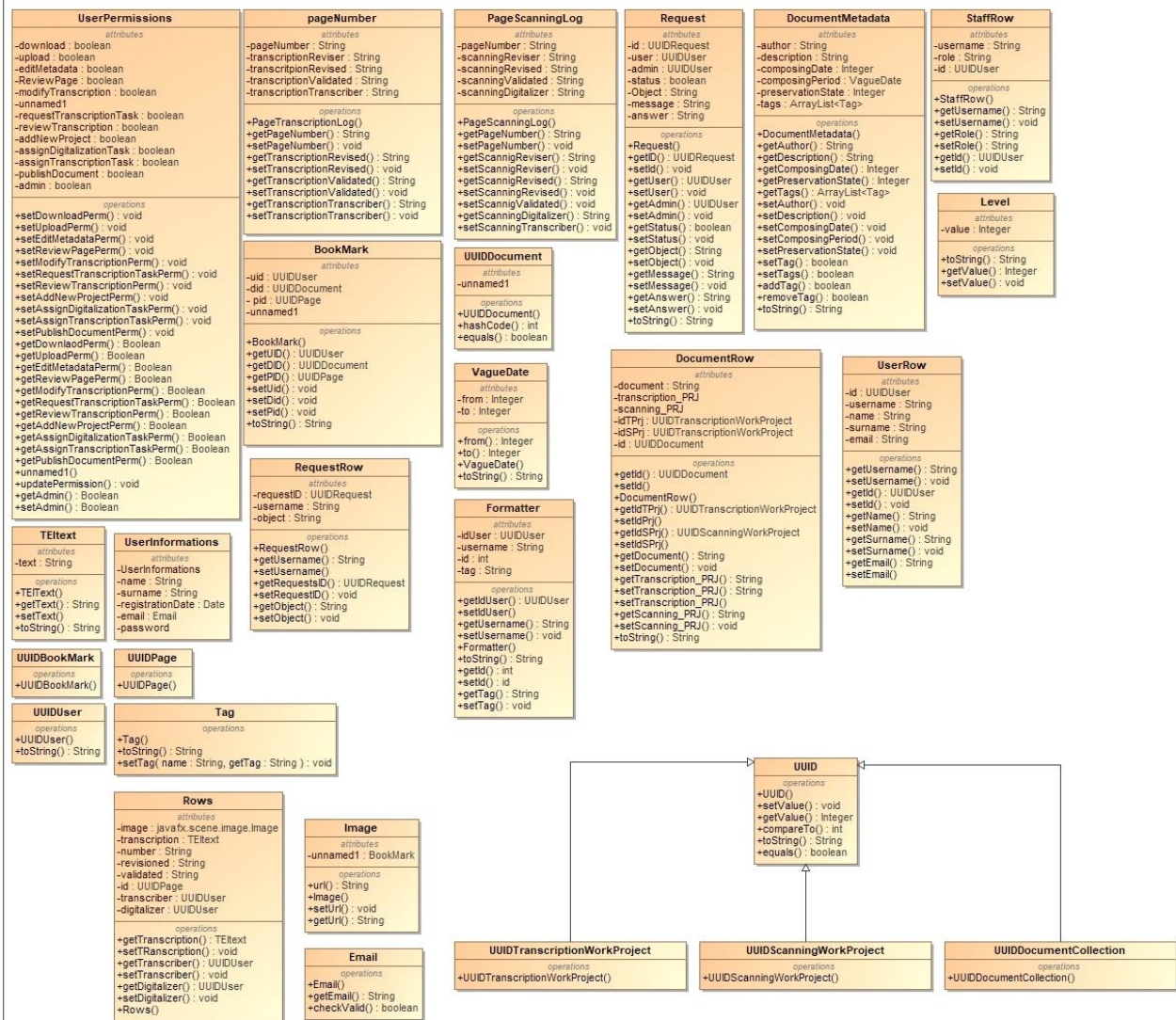
operations

- +register() : String
- +checkValidUsername() : boolean
- +checkUsername() : boolean

Administration Controller

operations

- +getRequest() : LinkedList<Request>
- +loadRequest() : Request
- +loadAllUsers() : HashMap<UUIDUser, String>
- +loadUserList() : LinkedList<User>
- +loadUsername() : String
- +ignoreUserRequest() : void



ScanningWorkProjectQuerySet

operations

```
+insertScanningWorkProject() : UUIDScanningWorkProject
+loadScanningWorkProject() : ScanningWorkProject
+loadUUIDDdocument() : UUIDDdocument
+insertDigitalizerUser() : Boolean
+insertReviserUser() : Boolean
+removeDigitalizerUser() : Boolean
+removeReviserUser() : Boolean
+insertDigitalizerUser() : Integer
+insertReviserUser() : Integer
+removeDigitalizerUser() : Integer
+removeReviserUser() : Integer
+deleteScanningWorkProject() : boolean
+getScanningUncompletedDocument() : HashMap<UUIDDdocument, String>
+loadMyCoordinatedScanningWorkProject() : HashMap<UUIDDScanningWorkProject, String[]>
+ifExistScanningProject() : boolean
+changeScanningProjectCoordinator() : Boolean
```

HomePageQuerySet

operations

```
+loadNews() : HashMap<UUIDUser,String>
+loadTranscriptionNews() : HashMap<UUIDUser,String>
+loadScanningNews() : HashMap<UUIDUser,String>
+loadMyTranscriptionWorkProjectList() : HashMap<UUIDDTranscriptionWorkProject, String[]>
+loadMyScanningWorkProjectList() : hashma<UUIDDScanningWorkProject, String[]>
+loadMyCollectionList() : HashMap<UUIDDdocument, String[]>
+loadMyCollectionList() : HashMap<UUIDDdocument, String[]>
+loadMyCollectionMetaData() : HashMap<Integer, DocumentMetadata>
+loadMyBookMarksList() : HashMap<Integer, BookMark>
+loadLibraryCollectionList() : HashMap<UUIDDdocumentCollection, String>
+loadCollection() : LinkedList<String[]>
+loadLibraryCollectionListWithDocument() : LinkedList<String[]>
```

TranscriptionWorkProjectQuerySet

operations

```
+insertTranscriptionWorkProject() : UUIDDTranscriptionWorkProject
+loadTranscriptionWorkProject() : TranscriptionWorkProject
+insertTranscriberUser() : Boolean
+insertReviserUser() : Boolean
+removeTranscriberUser() : boolean
+removeReviserUser() : boolean
+insertReviserUser() : Integer
+removeTranscriberUser() : Integer
+removeReviserUser() : Integer
+deleteTranscriptionWorkProject() : boolean
+loadUUIDDdocument() : UUIDDdocument
+loadMyCoordinatedTranscriptionWorkProjectList() : HashMap<UUIDDTranscriptionWorkProject, String[]>
+ifExistTranscriptionProject() : boolean
+changeTranscriptionProjectCoordinator() : Boolean
+insertTranscriberUser() : Integer
```

AdministrationQuerySet

attributes

```
-modifyUserStatus : Boolean
-updateUserPermissions : Boolean
-loadRequestList : LinkedList<Request>
-loadRequest : Request
-answerRequest : boolean
-ignoreRequest : Boolean
-loadUserList : LinkedList<User>
-showCoordinatorList : HashMap<UUIDUser,String>
```

loadAvailableTag()

operations

```
+insertDocument() : UUIDDdocument
+loadDocument() : Document
+loadDocumentToView() : Document
+updateDocument() : void
+deleteDocument() : void
+downloadFileDocument() : void
+loadAvailableTag() : HashMap<Integer, Tag>
+addTagToDocument()
```

DBConnection

operations

```
+connect() : Connection
+loadDatabaseException() : void
```

DigitalizerQuerySet

operations

```
+loadDocument() : LinkedList<Page>
+updatePage() : void
+updatePageNumber() : void
+createPage() : UUIDPage
```

PageViewQuerSet

operations

```
+loadOpera() : void
+loadBookmarksDocument() : LinkedList<BookMark>
+removeBookmarksDocument() : void
+addBookmarksDocument() : void
```

RequestToAdminQuerySet

operations

```
+sendRequestToAdmin() : UUIDRequest
+loadAllRequestToAdmin() : LinkedList<Request>
```

ReadProperties

operations

```
+getConnectionProperties() : Properties
```

DigitalizationRevisorQuerySet

operations

```
+validated() : void
+revised() : void
+scanningProcessCompleted() : void
+addScanningRevisionComment() : void
```

UserAuthenticationQuerySet

operations

```
+registration() : boolean
+checkIfUsernameExists() : boolean
+login() : UUIDUser
+getUserPermission() : UserPermissions
+PermissionsControl() : boolean
+control() : boolean
```

TranscriberQuerySet

operations

```
+loadTranscription() : PageTranscription
+updateText() : Boolean
+loadDocument() : LinkedList<Page>
```

SearchQuerySet

operations

```
+searchByAuthorTitle() : LinkedList<UUIDDdocument>
+searchByTag() : LinkedList<UUIDDdocument>
```

DatabaseException

attributes

```
-DatabaseException : void
```

TranscriptionReviserQuerySet

operations

```
+completed() : boolean
+validated() : boolean
+revised() : boolean
+addTranscriptionRevisedComment()
```

EditProfileQuerySet

operations

```
+loadUserProfile() : User
+updateUserProfile() : boolean
+deleteUserProfile() : void
```