EagleLibrary - System Design

1. Requisiti

La progettazione del sistema EagleLibrary inizia dalla stilatura dei requisiti funzionali e non. Per poter individuare quelli che, secondo noi, erano i requisiti che caratterizzavano l'applicazione da implementare abbiamo in prima istanza riportato tutti quei requisiti che si evincevano facilmente dalle specifiche del progetto, in più, ci siamo immedesimati nell'utente finale dell'applicazione per cercare di formalizzare altri requisiti che rendessero il sistema quanto più semplice ed user friendly possibile.

1.1 REQUISITI FUNZIONALI:

1. Ricerca opere digitali tramite filtri (metadati, parte del testo)
2. Classificazione delle opere
3. Consultazione opera
4. Download opera
5. Registrazione come trascrittore (con modulo)
6. Profilo utente
7. Caricamento opere in pdf con metadati (titolo, categoria, ecc…)
8. Consentire di supervisionare le opere caricate
9. Ogni opera acquisita deve essere trasformata in un testo digitale dai transcriber
10. Assegnazioni delle pagine ai transcriber
11. Revisionare, effettuare correzioni, validare e riassegnare le pagine trascritte
12. Gestione delle pubblicazioni per versioni digitali e immagini
13. Gestione dei livelli dei trascrittori
14. Supervisione delle acquisizioni delle immagini
15. Gestione in back-end di tutto il sistema.

1.2 REQUISITI NON FUNZIONALI:

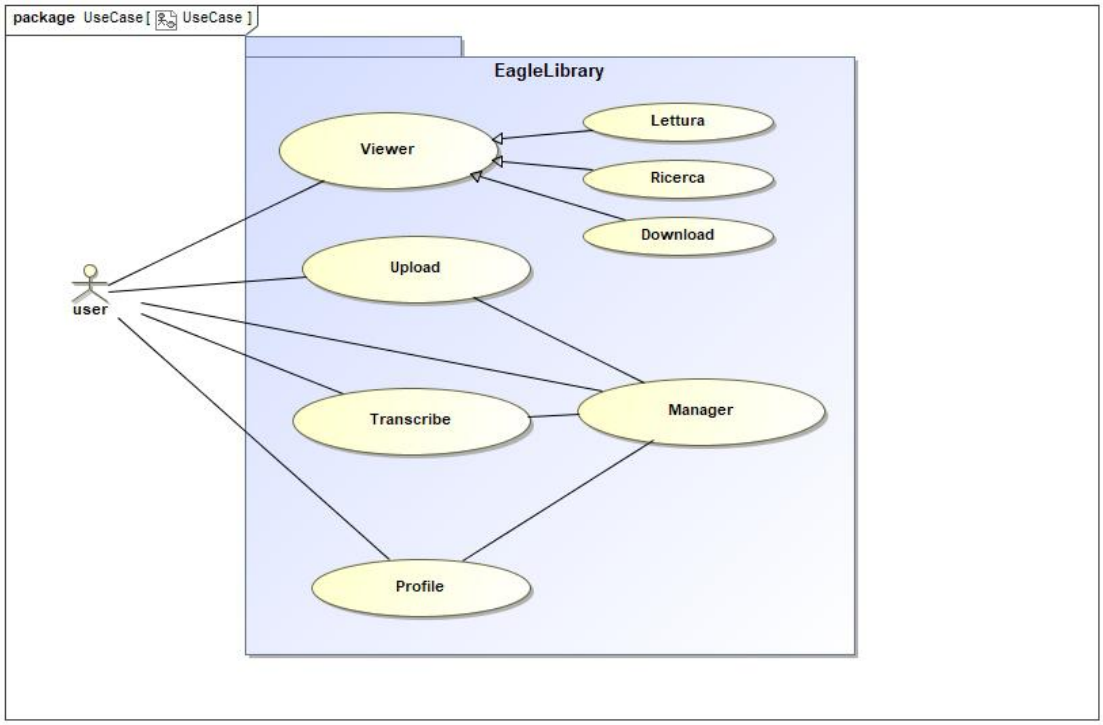
1. Utilizzare il linguaggio java
2. Garantire che le performance non influiscano negativamente con la user experience
3. Gestione di un DB per la persistenza dei dati.

1.3 USE CASE

Il secondo step è stato quello di partire dai requisiti e dalle specifiche di progetto per tirare fuori dei casi d'uso che iniziassero a farci capire ad alto livello quale sarebbe state le interazioni tra :

- user & system;
- subSystem & subSystem.

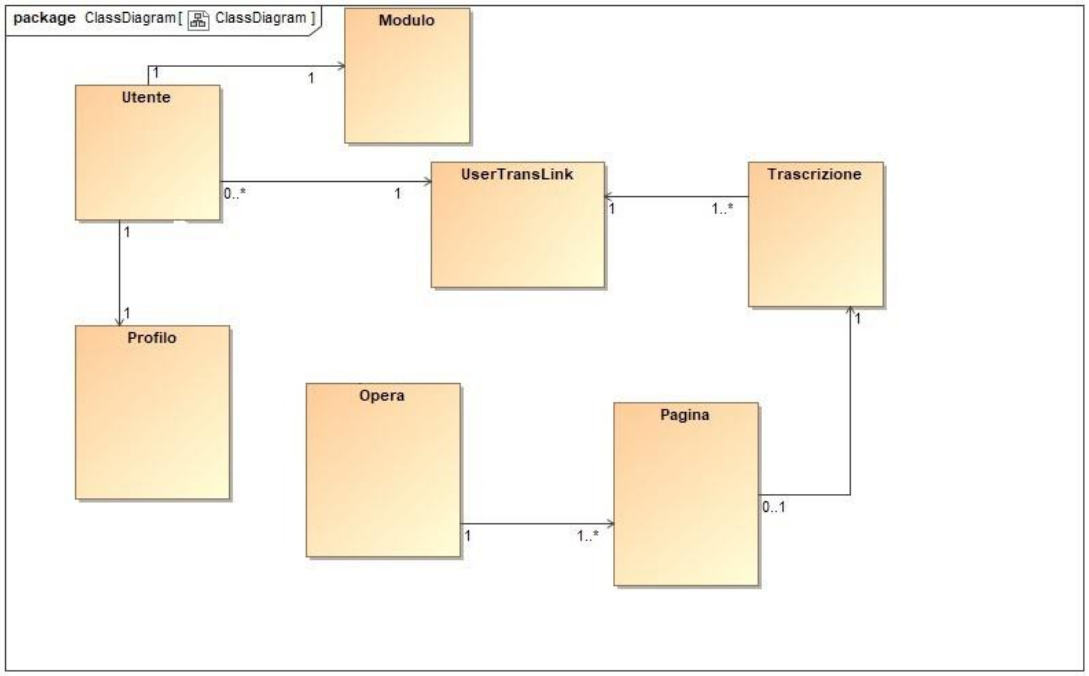
Ricavando dunque il seguente use case:



1.4 ENTITÀ

A questo punto ci siamo concetrati sull'estrazione delle entità che caratterizzano il sistema e abbiamo formalizzato tutto in un class diagram. In questa fase abbiamo dovuto fare delle assunzioni che ci hanno permesso di semplificare alcuni comportamenti del sistema e degli attori che vi interagiscono.

1.4.1 Entity Diagram



1.4.2 Assunzioni Entity Diagram:

1. Supponiamo che un utente può inviare un singolo modulo che, in base alla revisione del manager, può rinviare oppure dovrà aspettare un tot di tempo (es. 1 anno) prima di poter rinviare ancora il modulo; caso eccezionale per alcuni utenti che non potranno più rinviare il modulo.
2. Una pagina può avere una sola trascrizione
3. Assumiamo che la tabella di user_trans_link possa essere intesa come il concetto di Assegnazione.
4. Il modulo è un form online da compilare.
5. Supponiamo che l'immagine di una pagina è semplicemente un attributo della tabella Pagina.
6. Supponiamo che solo l'administrator possa effettuare upload delle varie opere.
7. Assumiamo che il modulo da parte di un utente può essere accettato o rifiutato solo dal manager.

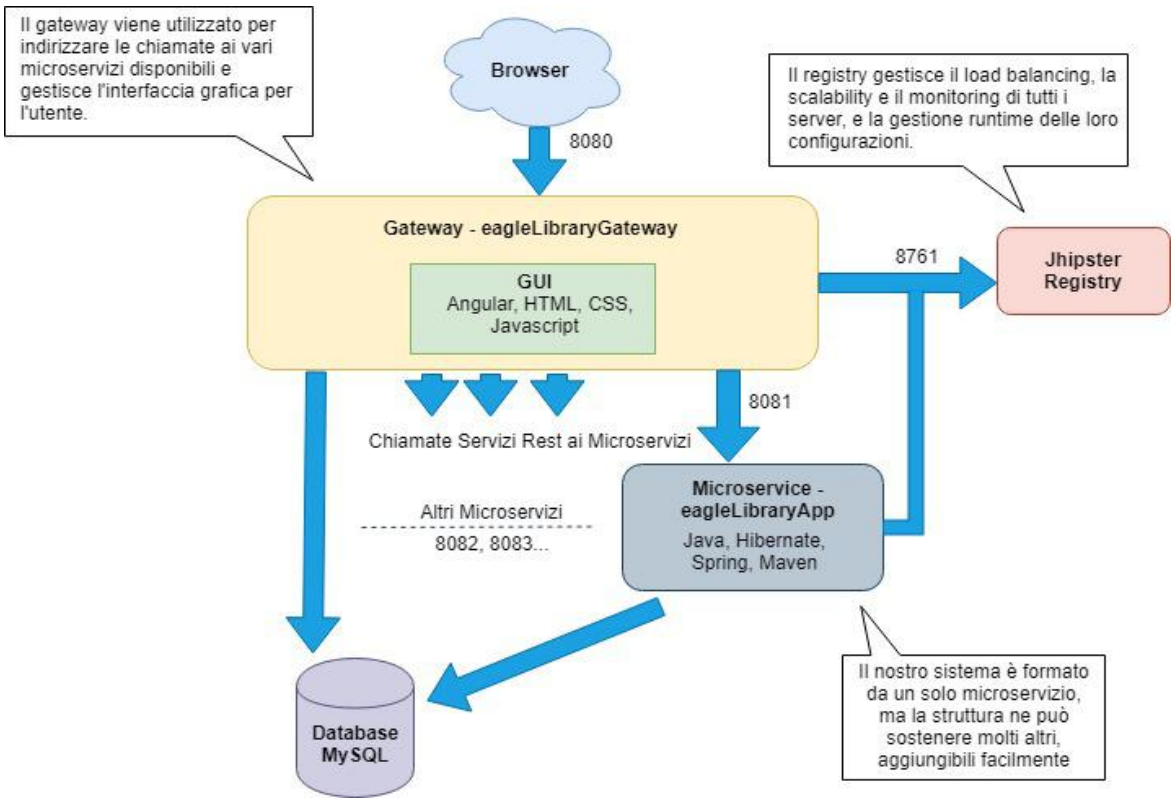
2. ARCHITETTURA

Una volta che sono diventati chiari quali dovessero essere le entità e come avrebbero dovuto interagire tra loro ad alto livello abbiamo iniziato a pensare al tipo di architettura da utilizzare. Fin da subito abbiamo voluto scegliere qualcosa che divergesse dal classico monolite così siamo andati verso una soluzione più modulare che non richiedesse aver presente già tutta l'architettura del sistema. Senza precluderci nulla quindi, abbiamo optato per un'architettura a microservizi.

2.1 MICROSERVIZI

Nella fattispecie di implementare (almeno per la prima versione del programma) 3 microservizi deployabili indipendentemente con Spring Boot. I tre microservizi sono i seguenti:

- Registry - modulo che agevola il monitoring, server balance, application's performance e così via.
- Gateway - modulo che gestisce la parte grafica e che comunica con il microservizio che verrà presentato al punto successivo per mezzo di chiamate rest.
- Application - modulo che gestisce tutta la parte back end della nostra applicazione.



2.2 SCAFFOLDING

La piattaforma che ci ha aiutato a realizzare l'architettura del nostro progetto è stata Jhipster.

Come prerequisito all'utilizzo di Jhipster abbiamo dovuto installare sui nostri terminali :

- Java 8;
- NodeJs 8.11.3 LTS (le versioni non LTS non sono supportate);
- Yeoman 2.0.5;
- Yarn 1.13.0;

Jhipster ci ha aiutato molto per la generazione automatica della DTS e di conseguenza di tutta la struttura del progetto. Inoltre abbiamo utilizzato la piattaforma anche per generare una prima versione delle JPA, dei POJO e dei CRUD per le entità principali dell'applicazione.

Lo start up dei vari moduli con l'ausilio di Jhipster ha incluso nel progetto features che potremmo decidere di non tenere e/o customizzare a seconda delle esigenze del cliente e del team di sviluppo.

2.2 FRAMEWORKS

Tra i framework che abbiamo utilizzato ci sono anche i seguenti :

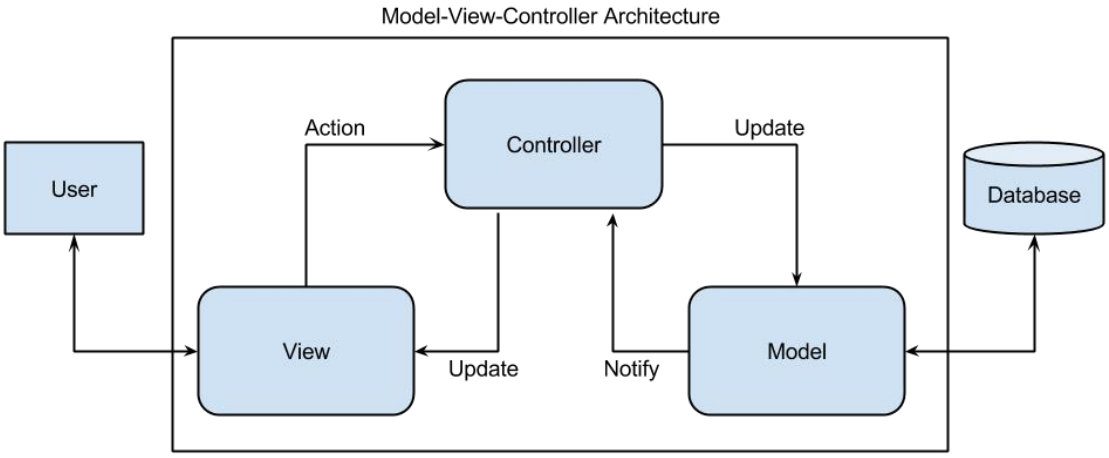
- Maven per la gestione del progetto, build automation e la gestione delle dipendenze;
- Spring, nella fattispecie l'utilizzo di @Annotation per lo scanning e il detecting dei package nel progetto. In più lo utilizziamo per l'injection.
- Hibernate per la gestione del DB lato backend.

2.3 DATABASE

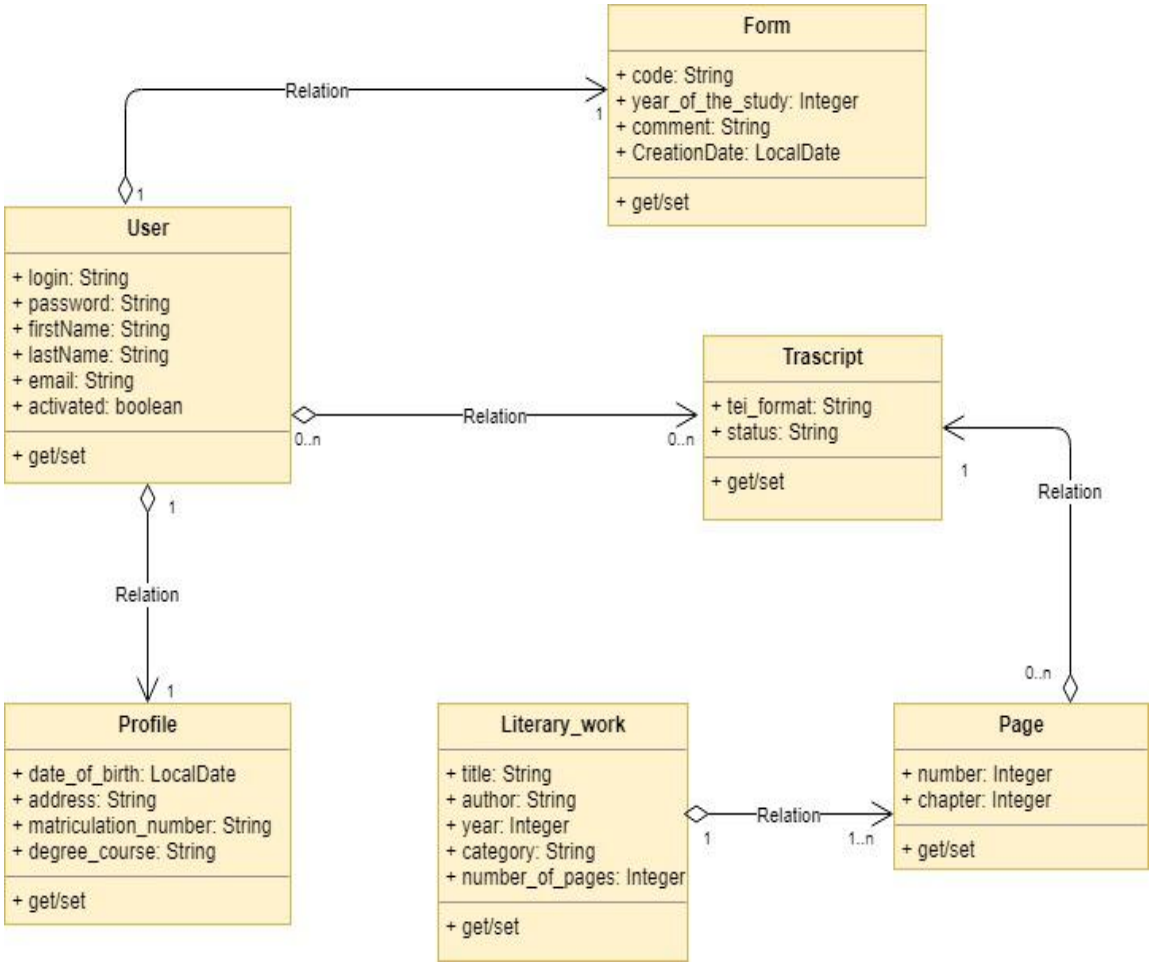
La scelta sul RDBMS da utilizzare è ricaduta su MySQL, un database open source molto semplice da utilizzare, gestibile con i vari tool messi a disposizione da Oracle, con il quale abbiamo già avuto esperienza in passato.

3. SYSTEM DESIGN

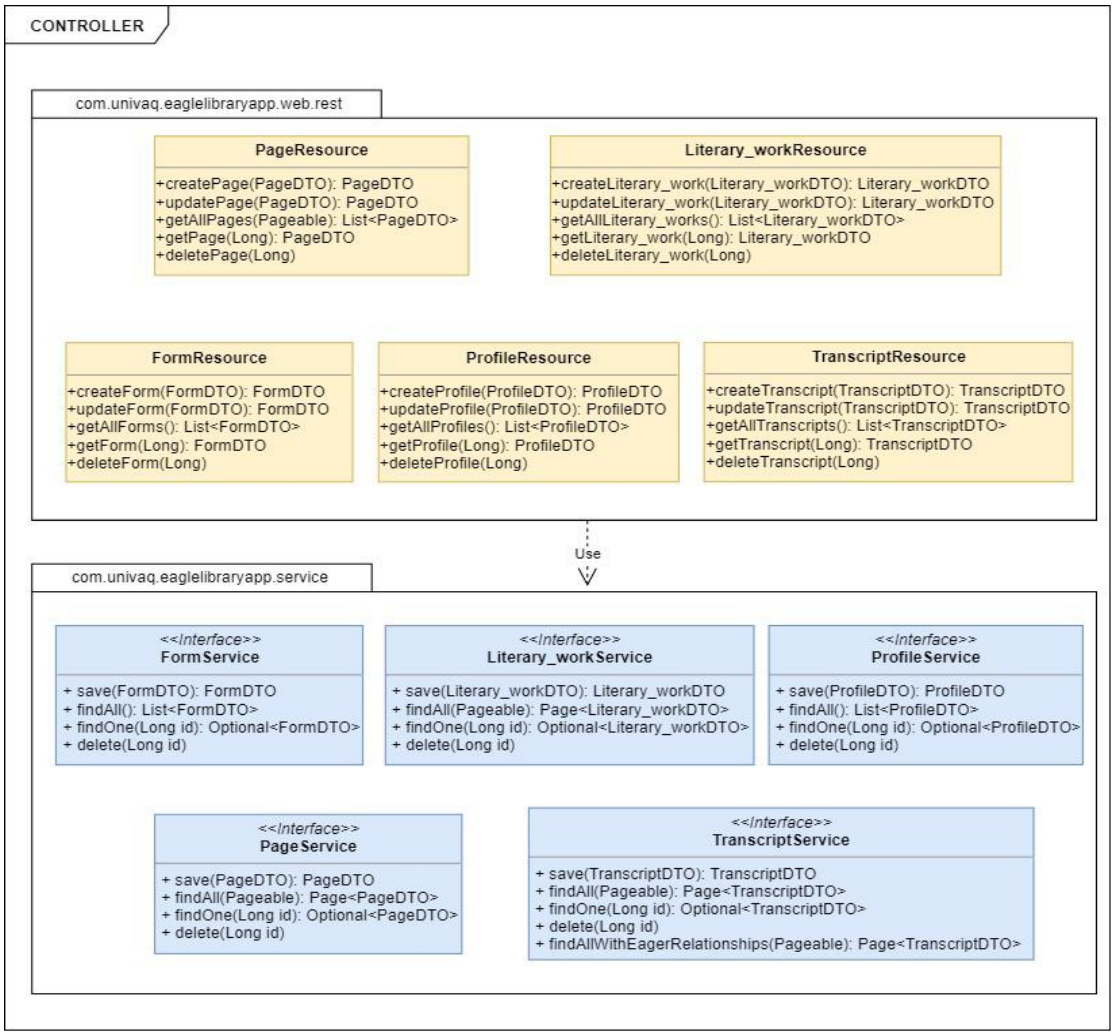
3.1 PATTERN MVC



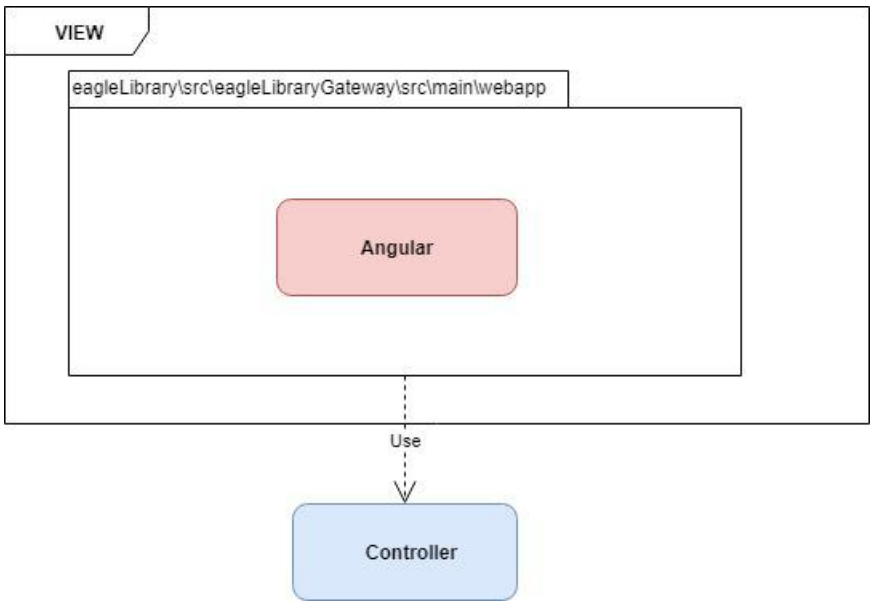
3.2 MODEL DIAGRAM



3.3 CONTROL DIAGRAM



3.4 VIEW DIAGRAM



4. SOFTWARE/OBJECT DESIGN

4.1 ENTITY SUB-SYSTEM RELATIONS

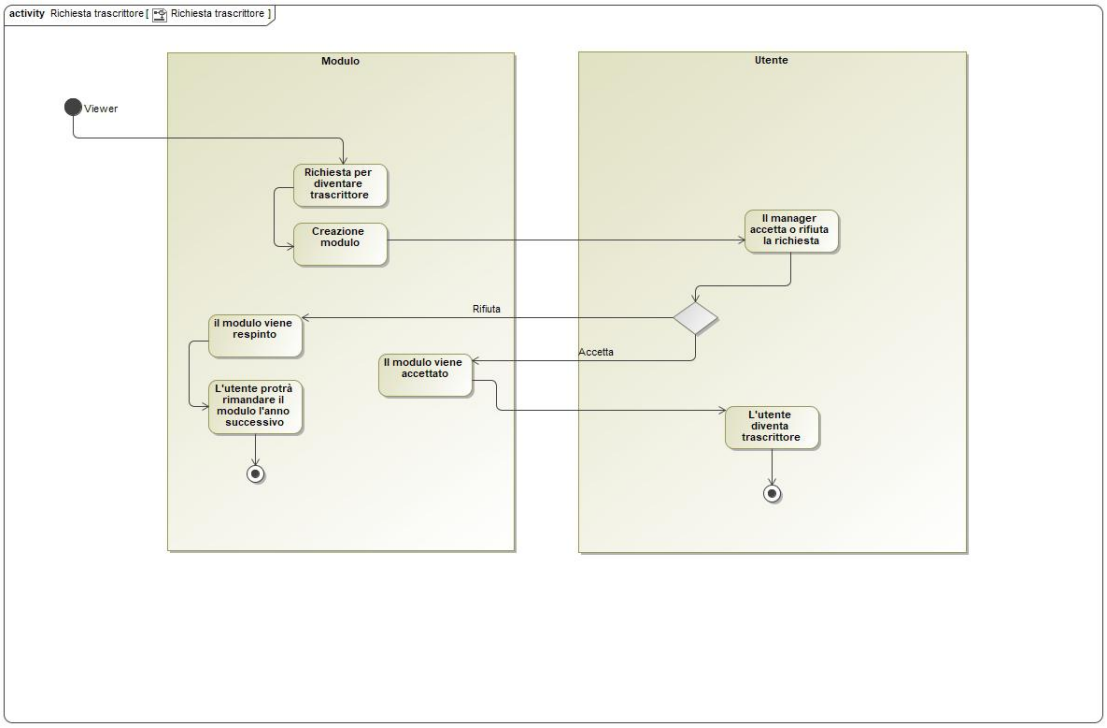
Attraverso questi passaggi abbiamo quindi consolidato la parte architetturale della nostra applicazione, a questo punto abbiamo fatto un passo indietro e siamo tornati sulle specifiche del progetto per guardare i requisiti da un punto di vista differente, invece di elencare cosa potesse/oppure no fare un subSystem [mi riferisco quindi a Viewer, Transcriber, etc...] abbiamo implementato uno schema che descrivesse tutte e sole le operazioni che vengono effettuate dai subSystem sulle entità che abbiamo individuato.

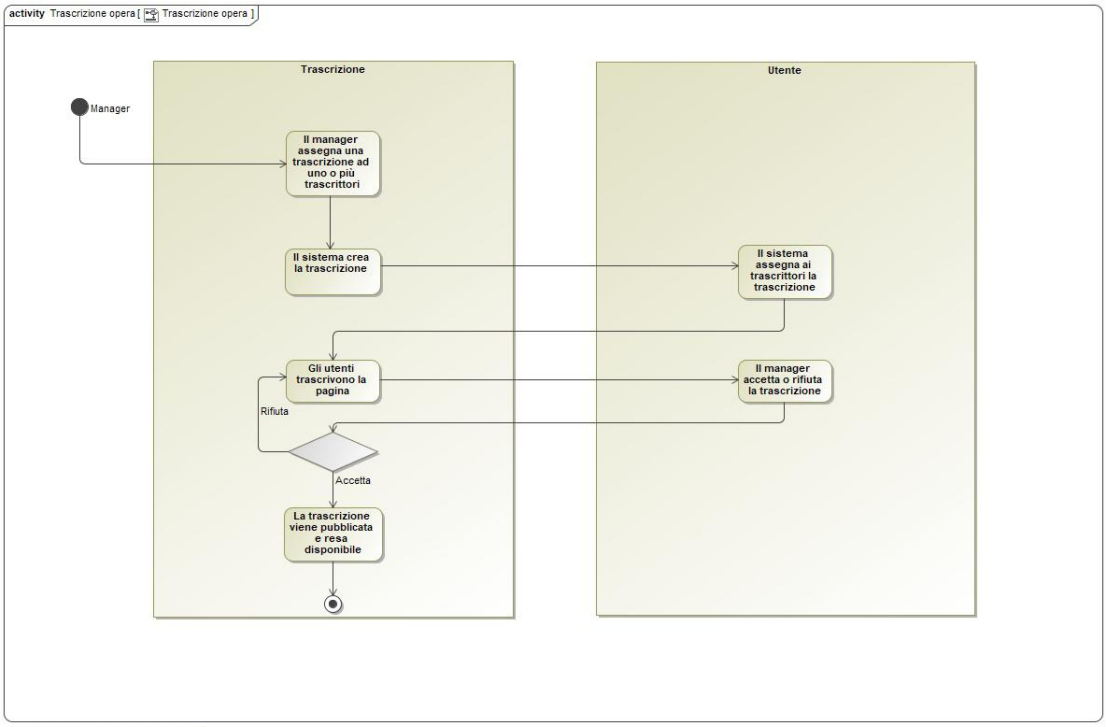
```
*******************************************************************************
Modulo :                                                                      *
        Viewer :                                                              *
            - Attraverso il modulo, l'utente può fare richiesta per diventare *
              un trascrittore.                                                *
                                                                              *
        Manager :                                                             *
            - Può accettare o rifiutare il modulo per la nomina di trascrittore; *
                                                                              *
*******************************************************************************
Utente :                                                                      *
        Viewer :                                                              *
            - Consente la visualizzazione, la ricerca                         *
              e il download (se si gode di alcuni privilegi) di un'opera;     *
            - La possibilità di far richiesta di diventare                    *
              trascrittore attraverso un modulo;                              *
            - Consente di accedere al proprio profilo;                        *
                                                                              *
        Uploader :                                                            *
            - Se un utente è anche un supervisore, allora l'utente può        *
              valutare la qualità di una trascrizione                         *
              (vedi ultima riga uploader);                                    *
                                                                              *
        Transcriber :                                                         *
            - Se un utente è trascrittore allora può trascrivere le parti di  *
              opera ad esso associati;                                        *
        N.B. : (ultima riga di transcriber == ultima riga di uploader, in cui *
                viene spiegata questa cosa);                                  *
                                                                              *
        Manager : (solo per utenti con alcuni privilegi!!!)                   *
            - Può assegnare 1 o più parti di un opera/e a più trascrittori;   *
            - Consente di effettuare correzioni e validazioni;                *
            - Può riassegnare pagine ai trascrittori;                         *
            - Consente la pubblicazione di trscrizioni e o immagini;          *
            - Consente la supervisione dell'acquisizione delle immagini;      *
            - Gestisce i livelli dei trascrittori;                            *
                                                                              *
*******************************************************************************
Opera :                                                                       *
    Viewer :                                                                  *
            - Consente la consultazione dell'opera;                           *
            - Consente la ricerca di un'opera per mezzo dei metadati (...);   *
            - Le opere possono essere suddivise per categoria;                *
            - Alcuni utenti possono fare il download dell'opera;              *
                                                                              *
    Uploader :                                                                *
            - Ogni opera è formata da più immagini (scansioni);               *
            - Per ogni opera vengono caricati dei metadati (titolo, anno, ...); *
                                                                              *
    Transcriber :                                                             *
            - Ogni opera acquisita deve essree trasformata in opera digitale; *
                                                                              *
    Manager :                                                                 *
            - Consente di associare parte di un'opera a 1 o più trascrittori; *
            - Consente la pubblicazione delle opere;                          *
                                                                              *
*******************************************************************************
Pagina :                                                                      *
N.B. per me la pagina è un immagine (Viewer rigo 3, Uploader rigo 1);         *
                                                                              *
    Viewer :                                                                  *
            - E' possibile sfogliare le pagine per mezzo di un paginatore;    *
                                                                              *
    Transcriber :                                                             *
            - Trascrive la pagina usando editor di testo formato TEI;         *
                                                                              *
    Manager :                                                                 *
            - Consente di assegnare parte di un'opera (1 o più immagini       *
              a 1 o più trascrittore);                                        *
            - Consente la pubblicazione di trscrizioni e o immagini;          *
            - Consente di riassegnare un'immagine ad un trascrittore          *
              qualora la trascrizione necessiti di una nuova revisione;       *
            - Consente la supervisione dell'acquisizione delle immagini;      *
                                                                              *
*******************************************************************************
Profilo :                                                                     *
        Viewer :                                                              *
            - Gli utenti possono accedere al proprio profilo o al profilo     *
              di altri per la visualizzazione dei dati inerenti all'utente    *
              associato al profilo;                                           *
                                                                              *
*******************************************************************************
```

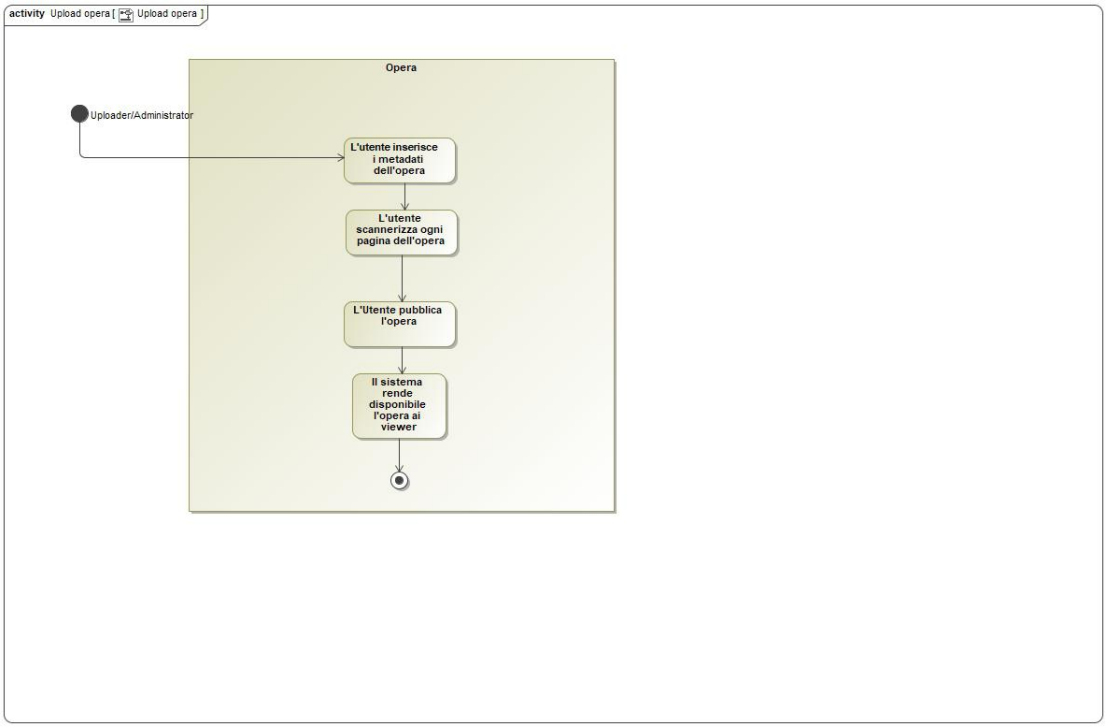
```
Trascrizione :                                                                  *
            Viewer :                                                            *
                - Può essere ricercato per mezzo di metadati;                   *
                - Può essere visualizzato da utenti registrati;                 *
                                                                                *
            Uploader :                                                          *
                - L'uente può caricarla per sottometterla a revisione;          *
                                                                                *
            Transcriber :                                                       *
                - Mediante un editor TEI effettua(crea) la trascrizione;        *
                                                                                *
            Manager :                                                           *
                - Revisione, corregge e valida una trascrizione;                *
                - Pubblica una trascrizione;                                    *
                                                                                *
********************************************************************************
```

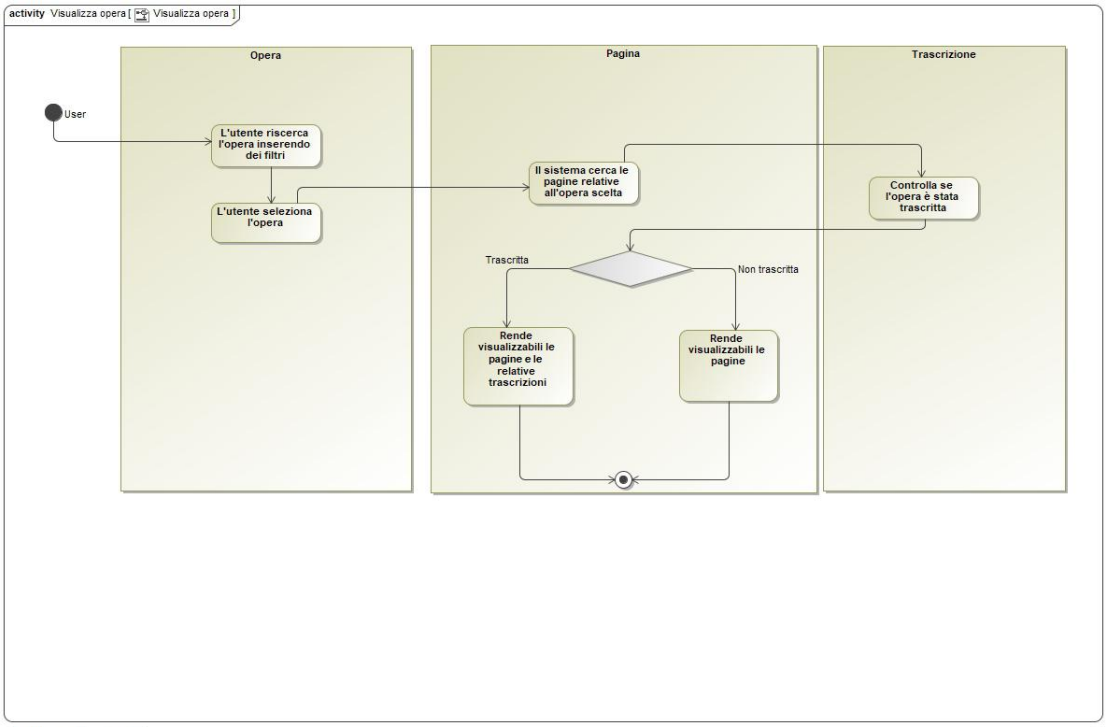
4.2 ACTIVITY DIAGRAMS

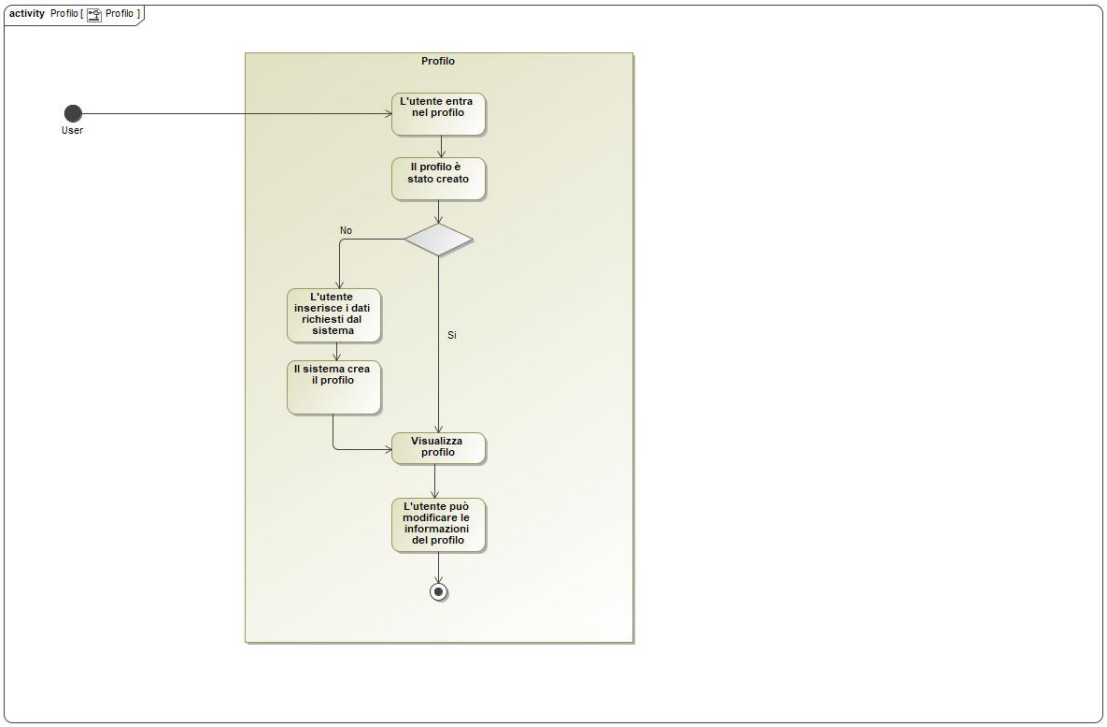
Questa nuova prospettiva ci ha offerto spunti per iniziare a buttare giù i primi high level design (che da qui denoterò con HLD) intesi come activity diagram (almeno in questa prima fase, solo activity) che appunto spiegano come interagiscono le entità con i vari subSystem.




activity Trascrizione opera [Trascrizione opera]
Trascrizione
Utente
Manager
Il manager assegna una trascrizione ad uno o più trascrittori
Il sistema crea la trascrizione
Il sistema assegna ai trascrittori la trascrizione
Gli utenti trascrivono la pagina
Il manager accetta o rifiuta la trascrizione
Rifiuta
Accetta
La trascrizione viene pubblicata e resa disponibile


activity Upload opera [Upload opera]
Opera
Uploader/Administrator
L'utente inserisce i metadati dell'opera
L'utente scannerizza ogni pagina dell'opera
L'Utente pubblica l'opera
Il sistema rende disponibile l'opera ai viewer


activity Visualizza opera [Visualizza opera]
Opera
Pagina
Trascrizione
User
L'utente riscerca l'opera inserendo dei filtri
L'utente seleziona l'opera
Il sistema cerca le pagine relative all'opera scelta
Controlla se l'opera è stata trascritta
Trascritta
Non trascritta
Rende visualizzabili le pagine e le relative trascrizioni
Rende visualizzabili le pagine


activity Profilo [Profilo]
Profilo
User
L'utente entra nel profilo
Il profilo è stato creato
No
Si
L'utente inserisce i dati richiesti dal sistema
Il sistema crea il profilo
Visualizza profilo
L'utente può modificare le informazioni del profilo