EAGLELIBRARY

**REQUISITI FUNZIONALI:**

1. Il sistema deve consentire la ricerca opere digitali mediante una serie di parametri. I parametri sono i seguenti :

* Titolo;
* Anno di pubblicazione;
* Autore;
* Genere;
* Parte del testo di una trascrizione;

1. Il sistema deve fornire una classificazione delle opere attraverso il loro genere (Storia, Horror, …).
2. Quando viene completata la ricerca di un’opera, il sistema deve consentire la visualizzazione dell’opera attraverso un widget di paginazione per facilitarne la consultazione.
3. Il sistema deve consentire il download dell’opera se l’utente è munito di permesso speciale.
4. Se l’utente non è munito di opportuni permessi, l’icona che consente di fare download non apparirà sulla GUI.
5. Il sistema deve consentire ad ogni utente che fa uso dell’applicazione di poter diventare trascrittore per mezzo di un modulo opportunamente compilato.
6. Il modulo deve essere controllato da un manager al fine di essere validato o rigettato, qualora quest’ultimo dovesse essere rigettato, l’utente potrà riprovare l’operazione l’anno successivo.
7. Se l’utente prova ad inviare un modulo che era stato precedentemente rigettato senza che un anno sia già passato il sistema emetterà un messaggio di errore per informare l’utente dell’errata procedura.
8. Per ogni utente di EagleLibrary, il sistema inizializzerà un profilo inserendo le informazioni anagrafiche di base, da quel momento esso sarà modificabile in ogni momento.
9. Il sistema deve consentire l’upload delle opere, l’opera sarà caratterizzata da una testata formata dalle informazioni di base quali :

* Autore;
* Titolo;
* Anno di pubblicazione;
* Genere;

e da un insieme di pagine, in formato immagine, che saranno caricate una per volta.  
Anche la procedura di caricamento deve essere agevolata da un paginatore per avere un formato più compatto delle immagini che si stanno caricando.

1. Quando un’opera viene inserita nel sistema, esso sarà soggetto ad una serie di controsaksadnkasdnsandksasasasdssdadsasadk
2. Per ogni opera presente nel sistema bisognerà effettuarne la trasposizione in digitale, ad occuparsi di questo compito potranno essere soltanto i trascrittori, diventati tali per mezzo del modulo di cui sopra.
3. Il sistema deve consentire di effettuare assegnazioni utente 🡨🡪 pagina affinché l’utente possa lavorare alla trascrizione in digitale della pagina/pagine ad esso associata/e.
4. Quando viene completata la fase di trascrizione di una pagina, quest’ultima dovrà essere demandata a dei controlli quali quelli di sintassi e semantica. In virtù di questo passaggio la trascrizione potrà essere accettata senza problemi oppure potrà essere riassegnata all’utente perché non conforme alla pagina originale dell’opera. Seguiranno quindi correzioni e l’iter descritto in precedenza andrà avanti fintantoché non si arriva alla perfetta uguaglianza tra pagina e trascrizione.
5. Quando la trascrizione di una determinata pagina supera tutti i controlli il manager abilita la pubblicazione di quest’ultima. In particolare, quando l’opera verrà visualizzata in corrispondenza della pagina trascritta sarà possibile visualizzare anche la versione digitale.
6. Il sistema consente ai manager di attribuire delle valutazioni in scala ai trascrittori che ne valorizza l’abilità di trascrizione. In particolare, su una scala da 1-5 sarà possibile, ogni utente sarà caratterizzato da un valore appartenente a quel range.
7. In fase di upload di un’opera da parte dell’utente vengono effettuati una serie di controlli al fine di mandare a buon fine l’upload stesso. TBD : Quali sono questi controlli ? Sono manuali o automatici?

**REQUISITI NON FUNZIONALI:**

1. L’implementazione della parte backend dell’applicazione dovrà essere scritta utilizzando il linguaggio Java, in particolare Java 1.8;
2. Garantire che le performance non influiscano negativamente con la user experience
3. Gestione di un DB per la persistenza dei dati.
4. Per il meccanismo di sincronizzazione delle modifiche degli utenti su di un’unica trascrizione si vuole utilizzare un pattern a semaforo, ossia gestire con lock pessimistico le modifiche sulla trascrizione in modo tale da non mandare in overlap le varie versioni.