PROPOSTA DI PROGETTO PER LA PROVA FINALE DEL CORSO DI
OBJECT ORIENTED PROGRAMMING
CORSO DI LAUREA IN INFORMATICA
UNIVERSITA' DEGLI STUDI DELL'AQUILA
A.A. 2015/2016

SPECIFICA

Il progetto si propone di realizzare una *biblioteca digitale* di testi e studi che contribuiscono alla formazione della cultura all'interno dell'Università degli Studi dell'Aquila.

Una biblioteca digitale è 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. Lo scopo di questo progetto è la digitalizzazione di manoscritti, che costituiscono un patrimonio bibliografico antico per un totale di 60.000 carte (ms. sec. XV-XIX) contenenti memorie storiche della città dell'Aquila.

Il processo di digitalizzazione dei manoscritti si suddivide in diverse fasi:

- Digitalizzazione
  Il manoscritto è acquisito dal sistema sotto forma di immagini digitali ad alta risoluzione attraverso scanner planetari. Ogni manoscritto è formato da più acquisizioni (ogni immagine rappresenta una singola pagina). La digitalizzazione viene controllata da supervisori all'acquisizione per assicurarne la correttezza (ad esempio, in accordo con standard richiesti) e la qualità. L'immagine acquisita viene memorizzata all'interno del sistema ed assegnata all'opera di riferimento, corredandola di opportuni metadati.
- Trascrizione
  Il manoscritto così acquisito deve essere trasformato in un testo digitale; ciò avviene attraverso operazioni di trascrizioni in formato TEI (Text Encoding Initiative)[1]. Le trascrizioni sono digitate manualmente (la natura del testo rende inutilizzabili strumenti di acquisizione automatica) attraverso un text editor TEI integrato[2].
  Le trascrizioni sono oggetto di revisione da parte di revisori alle trascrizioni.
- Pubblicazione
  I manoscritti, una volta digitalizzati e superata la fase di revisione delle immagini, vengono pubblicati sul sistema e resi accessibili agli utenti del sistema. Le corrispondenti trascrizioni sono pubblicate successivamente, dopo la validazione della stessa.

Attori del sistema:

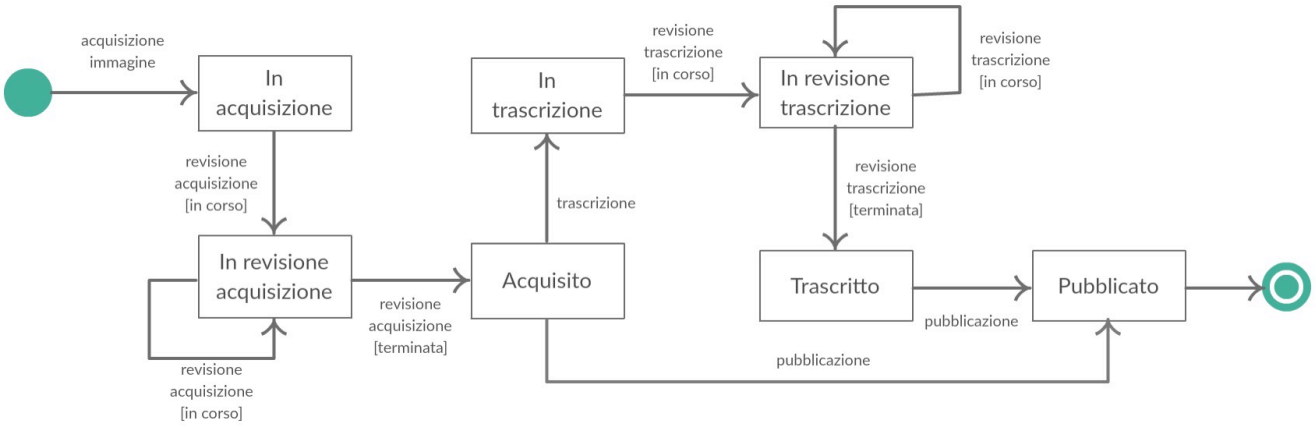
- Amministratore: gestione generale del sistema
- Acquisitore: acquisizione/digitalizzazione immagine
- Revisore acquisizioni: revisione e verifica correttezza dell'acquisizione
- Trascrittore: trascrizione del testo TEI
- Revisore trascrizioni: revisione e validazione della trascrizione
- Utente base: visualizzazione elenco titoli opere
- Utente avanzato: visualizzazione completa delle opere

---

[1] http://www.tei-c.org/index.xml

[2] http://wiki.tei-c.org/index.php/Editors

Processo di pubblicazione:



Links utili:

- Sito per la trascrizione collaborativa del progetto “Bentham”:
  https://www.ucl.ac.uk/Bentham-Project
  con la piattaforma di trascrizione:
  http://www.transcribe-bentham.da.ulcc.ac.uk/td/Transcribe_Bentham
  e il rimando alla nuova piattaforma con l’OCR per manoscritti:
  http://www.transcribe-bentham.da.ulcc.ac.uk/TSX/

- Altro sito collaborativo:
  http://letters1916.maynoothuniversity.ie

- Mirador:
  http://iiif.github.io/mirador/
  usato da Biblissima (http://www.biblissima-condorcet.fr) in Francia per biblioteche digitali di manoscritti. Oltre a permettere di visualizzare le immagini, permette anche l’annotazione (vedi video).

- OpenSeaDragon. Database che registra l’indicizzazione.
  https://openseadragon.codeplex.com

REQUISITI

Parte 1

1. **Requirements**
   Se necessario, riorganizzare la specifica data sotto forma di requisiti.
   Documenti richiesti: (1.1) documento dei requisiti.

2. **System design**
   Definire gli obiettivi di design, decomporre il sistema, selezionare le strategie per la costruzione del sistema, definire l'architettura software del sistema, scegliere design patterns.
   Documenti richiesti: (2.1) modello dell'architettura software (o più modelli con diverso grado di dettaglio), (2.2) descrizione dell'architettura, (2.3) descrizione delle scelte e strategie adottate (compresi design patterns)

3. **Software/Object design**
   Definire gli oggetti del dominio, definire in modo dettagliato componenti, classi, interfacce e membri.
   Documenti richiesti: (3.1) modelli rappresentanti l'object design con classi, interfacce e membri, (3.2) descrizione dei dettagli di design scelti.

4. **Deliverable (deadline 6/5/2016)**
   Il progetto deve essere mantenuto in un repository GIT, che sarà strutturato come segue.

   <nome_progetto> : root del progetto
   - *src :* questa directory conterrà il codice sorgente del progetto
   - *doc :* questa directory conterrà i documenti di design del progetto (in format pdf)
   - *javadoc :* questa directory conterrà il codice javadoc del progetto

   La URL del repository sarà consegnata al docente, la parte di competenza di questo deliverable sarà contenuta nella directory doc.

Note: Gli strumenti a disposizione per svolgere la parte 1 del progetto riguardano il programma del corso relativo alle lezioni 1-13 (OO design, Classi e membri, Package, modificatori di accesso, final/static, Ereditarietà, Polimorfismo, Interfacce)

Parte 2

1. **Refining the design**
   Il design del sistema e il design software devono essere raffinati ed integrati alla base delle nuove conoscenze acquisite nella seconda parte del corso.
   Documenti da raffinare: (2.1), (2.2), (2.3), (3.1)

2. **Implementation**
   Implementazione del sistema software alla base delle scelte di design effettuate. Il lavoro deve essere svolto all'interno di un ambiente di sviluppo (IDE), preferibilmente Eclipse. Il lavoro deve essere sviluppato in modo collaborativo attraverso un repository GIT.

   L'implementazione del sistema deve necessariamente far uso dei seguenti strumenti:
   - Classi, package e modificatori di accesso per una corretta strutturazione del sistema
   - Interfacce

- Ereditarietà
- Polimorfismo e overriding/overloading di metodi con almeno un esempio di dinamic binding
- Uso di final, static
- Eccezioni
- Collection/Map
- JDBC

3. **Documentation**
   La documentazione da allegare al progetto sarà costituita dai documenti ai punti 2.* e 3.* raffinati e inoltre da Javadoc ed eventuali annotazioni (ciò implica opportuna formattazione e commenti).

4. **Deliverable**

   A ultimazione del progetto la comunicazione della URL del repository da aggiorare sarà consegnata al docente, le parti di competenza di questo deliverable saranno contentute nelle directory src, doc, javadoc.

Note: Gli strumenti a disposizione per svolgere la parte 2 del progetto riguardano il programma del corso relativo alle lezioni 14-20 (Eccezioni, Collections/Maps, Generics, I/O, JDBC,Threads, Design patterns).

Tutti i progetti non conformi ai sopra indicati requisiti non saranno accettati.