

Tecnologie Web T
08 Febbraio 2017 – Compito

Tempo a disposizione: 3 ore

La soluzione comprende la **consegna elettronica** dei seguenti file mediante l'apposito applicativo Web **esamix** (<http://esamix.labx>):

Whiteboard.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Springer.zip	file zip contenente le pagine Web (file html, css e js) per punto 2
Archive.zip	file zip contenente il sorgente java/class e file txt per punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), equamente distribuiti sui tre esercizi, ovvero almeno 6 punti sul primo esercizio, 6 punti sul secondo esercizio e 6 punti sul terzo esercizio

Studenti in debito di Tecnologie Web L-A

Viene richiesto lo svolgimento dei soli esercizi 1 (17 punti) e 2 (16 punti). Tempo a disposizione: 2 ore.

I 18 punti necessari per l'ammissione all'orale sono così distribuiti: almeno 10 punti sul primo esercizio e almeno 8 punti sul secondo

ESERCIZIO 1 – Whiteboard (11 punti)

Si realizzi un'applicazione Web, principalmente basata su tecnologie **JSP**, **Java servlet** e **Javascript**, per la realizzazione di una whiteboard condivisa.

L'applicazione deve consentire ad ogni utente di aprire una propria sessione e di essere riconosciuto; come ipotesi semplificativa, non è possibile per uno stesso utente avere più sessioni correntemente attive, ad esempio da terminali differenti. Una volta aperta la sua sessione, da quel momento in avanti l'utente deve ricevere i nuovi contenuti (scritti da quel momento in avanti) della whiteboard; tali contenuti sono organizzati in righe di testo (sequenze di caratteri separati da newline). Ogni utente dovrà avere la possibilità di

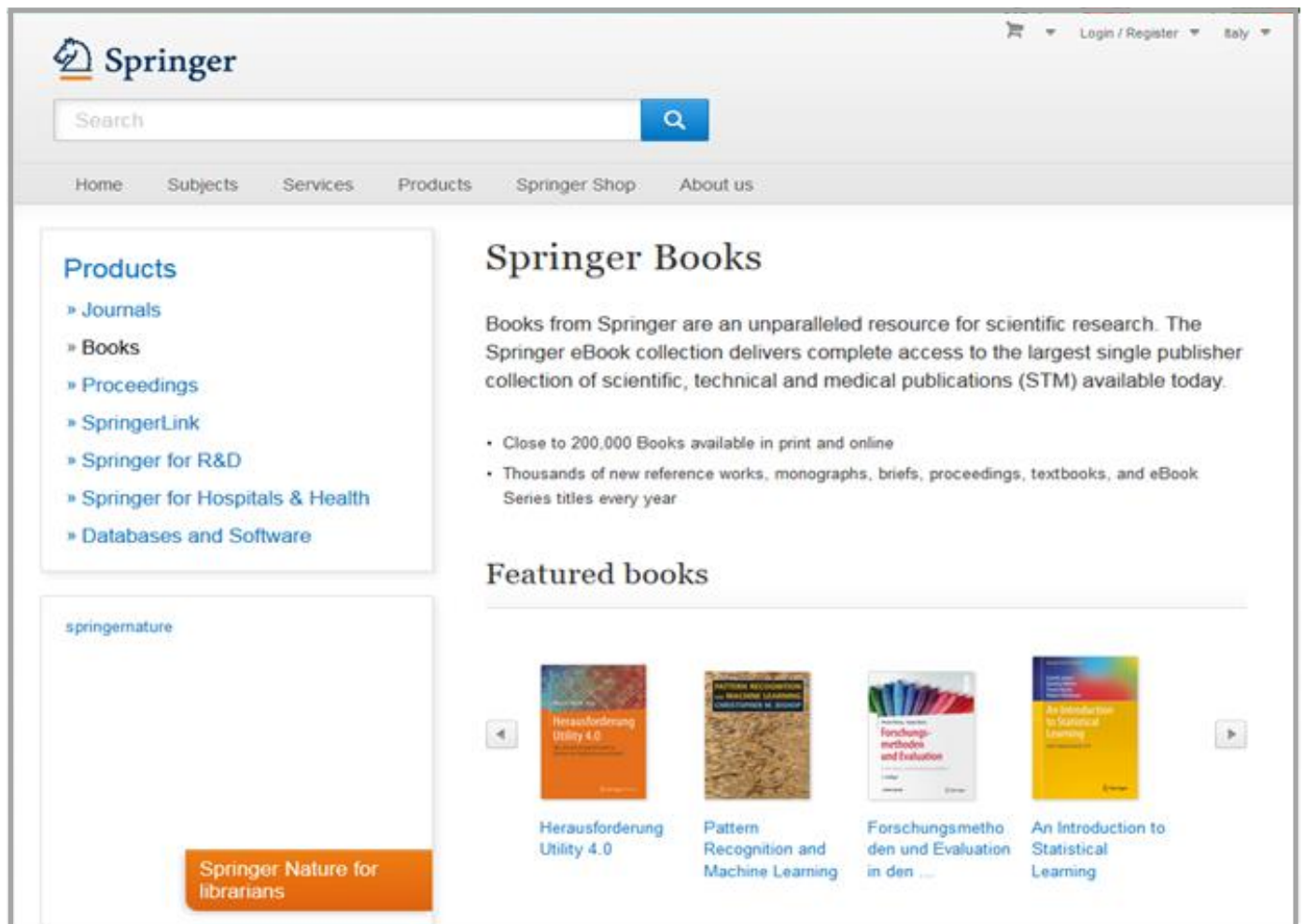
1. contare le occorrenze di una parola, specificata come input, nel suo insieme di righe visibili e
2. aggiungere una nuova riga solo alla fine della whiteboard (append).

In caso di modifiche concorrenti che possano portare a inconsistenze (descrivere quali sono esattamente come commento nel file sorgente), alcune richieste utente devono fallire; i clienti notificati del fallimento devono inoltre ricevere la nuova versione della whiteboard aggiornata in background tramite **AJAX**; conseguentemente deve essere aggiornata la loro pagina di visualizzazione.

Inoltre, deve essere data la possibilità all'amministratore del sistema (username=admin; password=admin) di cancellare qualsiasi riga della whiteboard, con priorità su tutti gli altri utenti, e ovviamente in modo tale da evitare corse critiche.

ESERCIZIO 2 – Springer (11 punti)

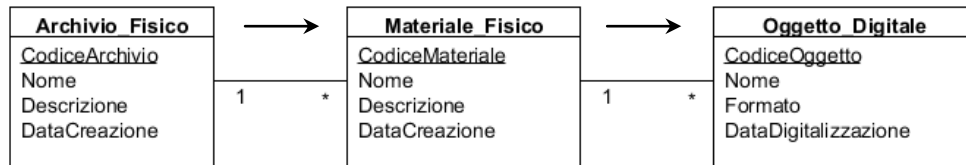
Si realizzino le **pagine Web dinamiche** (basate su **tecnologia HTML, CSS, e Javascript**) in grado di riprodurre il contenuto e il layout grafico del sito Web di seguito riportato. Nell'esempio specifico, il Menù "laterale" espande la voce "Products" del Menù "orizzontale" (generale), mentre la sezione centrale visualizza il contenuto relativo al sottomenù "Books" di "Products". Si preveda inoltre che il campo di ricerca possa ammettere solo stringhe alfabetiche composte di un massimo di due parole (esempio, "*computer science*").



Tecnologie Web T
08 Febbraio 2017 – Compito

ESERCIZIO 3 – Archive (11 punti)

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su tecnologia **Hibernate** in grado di “mappare” efficientemente, e con uso di ID surrogati, il modello di dominio rappresentato dai **JavaBean** del diagramma UML con le corrispondenti **tabelle relazionali derivate dalla progettazione logica** del diagramma stesso.



Nel dettaglio, dopo aver creato da applicazione Java gli schemi delle tabelle all'interno del proprio schema nel database **TW_STUD** di **DB2** (esplicitando tutti i vincoli derivati dal diagramma UML), implementato i **JavaBean**, definiti i **file XML di mapping** e il **file XML di properties**, si richiede la realizzazione di una classe di prova facente uso delle **API Hibernate** in grado di:

- inserire due o più tuple nelle tabelle di interesse;
- restituire **i)** per ogni archivio fisico post 2014, il numero dei formati distinti in cui sono raggruppati i corrispondenti oggetti digitalizzati **ii)** nomi degli archivi con formati digitali di tipo “*jpeg*”;
- produrre una stampa opportunamente formattata e completa dei risultati delle query del punto precedente sul file **Archive.txt**;

il tutto, mediante opportuna gestione delle **transazioni**.

N.B. La soluzione deve sfruttare i mapping specificati nel diagramma UML. Ogni ulteriore scelta fatta dallo studente deve essere opportunamente giustificata mediante commenti nel codice.