Tecnologie Web T
11 Settembre 2015 – Compito

Tempo a disposizione: 3 ore

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

CoppiaSessione.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
ShowRoom.zip	file zip contenente le pagine Web per punto 2
123.zip	file zip contenente il sorgente java/class e file xml 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 (11 punti)
Si realizzi un'**applicazione Web** che "**emuli**" il concetto di **sessione per una coppia di utenti**; il dominio applicativo sia il semplice calcolo di somme e moltiplicazioni fra operandi reali; l'applicazione deve essere basata principalmente su **tecnologie Java servlet, JSP e JSON**.
In particolare, l'applicazione dovrà essere costituita da una pagina iniziale di **login** che permetta di inserire username e password; tali informazioni di autenticazione verranno controllate server-side facendo uso di un file di testo (**password.txt**) costituito da una riga per ogni utente, contenente username e password separate da uno spazio bianco. Se due utenti sono contemporaneamente connessi all'applicazione e le loro righe in password.txt sono consecutive (in posizione i e i+1 con i pari, ovvero 0 e 1, oppure 8 e 9, ad esempio), allora quei due utenti devono condividere lo stesso stato di sessione applicativa.
Dopo essersi autenticato, un utente ha accesso a una pagina Javascript **calcola** che permette di inserire due operandi e di scegliere un'operazione (somma o moltiplicazione). La pagina deve controllare che i due operandi inseriti siano numeri positivi con meno di 4 cifre decimali; superati i controlli, gli operandi devono essere trasferiti in formato JSON con metodo POST verso una JSP **calcolatrice** che deve semplicemente calcolare e mostrare il risultato, oltre che visualizzare i risultati delle operazioni precedenti eseguite all'interno della sessione di interazione di coppia.
Infine, si commenti testualmente nel sorgente della soluzione se possono riscontrarsi situazioni di inconsistenza temporanea fra i dati di sessione mostrati ai due utenti di una coppia e in quali casi. È possibile evitare tali inconsistenze e con quali linee guida di soluzione (NON da includere nella soluzione da consegnare, ma solo da descriversi in linea di principio)?

Tecnologie Web T
11 Settembre 2015 – Compito

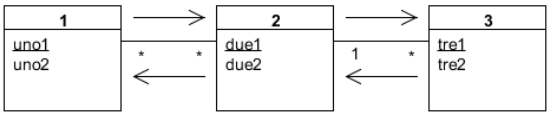
ESERCIZIO 2 (11 punti)

Si realizzino le **pagine dinamiche HTML del sito Web ShowRoom** basate su tecnologia **Javascript** che rispettino le seguenti specifiche (il layout grafico delle pagine deve essere confinato nell'opportuno file **ShowRoom.css**)

- la pagina principale è divisa verticalmente in due parti di diversa larghezza: **A)** parte sinistra (30% della larghezza totale); nel dettaglio, ha sfondo grigio, testo *Arial, Bold, 11pt* e riporta *i)* nella parte superiore il logo dello show room che espone, cliccando il quale deve essere possibile accedere al sito "*http://www.logomaker.it*" (il logo è rappresentato dall'ipotetica immagine "**show.jpg**" che deve essere visualizzata in modo centrato); *ii)* la parte centrale ospita un menù verticale con l'elenco delle classi di prodotti in mostra, ovvero *classe A*, *classe B*, *classe C*, *classe D*; *iii)* infine, nella parte in basso è previsto il link "*Registrazione Utente*" per la registrazione al sito da parte di nuovi utenti (il link punta ad apposito form di registrazione); **B)** parte destra (restante 70% della larghezza totale); nel dettaglio, è a sua volta divisa orizzontalmente in due sotto-parti *1)* parte alta (15% dell'altezza totale): ha sfondo verde e riporta il titolo "*Let's go shopping!*" (testo *Comics, Bold, 14pt* di colore grigio) *2)* parte restante: ha sfondo bianco, contiene nell'estremità superiore un piccolo form di login per utenti già registrati (campi form: **login** e **password utente** (caselle di testo editabili lunghe 12 e 8 caratteri, rispettivamente, e obbligatorie) più **bottone invia**, il tutto disposto orizzontalmente e allineato a sinistra; segue una linea di separazione tra il form di login e la finestra di visualizzazione sequenziale delle immagini dei prodotti delle classi selezionate dal menù da parte dell'utente (per utenti autenticati) oppure del form di registrazione per nuovi utenti. A titolo di esempio, per i prodotti di classe *A*, verranno visualizzate un numero costante **N** di immagini (*A1.jpeg, A2.jpeg,…, AN.jpeg*) rappresentative della categoria *A*.
- *contenuto form di registrazione*: **nome**, **cognome**, **email**, **login, password, password bis** (caselle di testo editabili lunghe 40, 40, 30, 12, 8 e 8 caratteri, rispettivamente, e tutte obbligatorie); **bottone invia**;
- controllo eventi: **A) mentre l'utente edita i campi dei form** i) subito dopo che l'utente ha editato i campi **nome**, **cognome** e **login**, deve verificare che tali campi non contengano numeri; ii) subito dopo che l'utente ha editato il campo **password**, deve verificare che tale campo contenga almeno 4 lettere (di cui 1 maiuscola), 3 numeri e un carattere speciale tra *!*, *$* e *?*; **B) al momento dell'invio del contenuto di un form** deve verificare che **tutti** i campi richiesti non siano vuoti.

ESERCIZIO 3 (11 punti)

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



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**, definito 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;
- ottenere i) per ogni tupla di *2* il numero di tuple di *1* ad essa associata e ii) per ogni tupla di *1* l'elenco delle coppie di 3 (*tre1*, *tre2*) ad essa associate; produrre una stampa opportunamente formattata del risultato dei punti precedenti sul file **123.txt**;

il tutto, mediante uso di **ID surrogati** e opportuna **gestione delle transazioni**.

N.B. La soluzione **deve sfruttare i mapping M-N** specificati. Tutti gli **attributi** dati sono di tipo **stringa**. **Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata** con commenti nel codice.