

Tecnologie Web T
15 Gennaio 2020 – Compito

Tempo a disposizione: 3 ore

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

Password.zip file zip contenente il sorgente java/class e pagine Web per punto 1

Sottrazione.zip file zip contenente il sorgente java/class e pagine Web per punto 2

Ricettario.zip file zip contenente il sorgente java/class, XML e 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 (11 punti)

Si realizzi un'applicazione Web, principalmente basata su tecnologie **Java servlet e JSP**, per la gestione della scadenza di **password di utenti appartenenti a gruppi**.

L'applicazione Web deve consentire a un nuovo utente di registrarsi, indicando il proprio nome utente, la propria password e il gruppo di appartenenza (in un set predefinito di gruppi `group0 ... groupN`). Utenti già registrati possono accedere alle funzionalità dell'applicazione dopo avere effettuato il login.

Al momento del login, l'applicazione Web deve controllare server-side se la password utilizzata è corretta e se è stata definita da più di 60 giorni (*scadenza temporale della password*); in caso positivo, è necessario chiedere all'utente di modificarla. Se per un determinato gruppo `group*` ci sono più di 3 password scadute, l'applicazione deve forzare il cambiamento della password per tutti gli utenti di quel gruppo, indipendentemente dal tempo di definizione di quelle password. Inoltre, ad ogni utente deve essere consentito di sbagliare la password al login per non più di 3 volte.

Infine, deve essere data la possibilità all'amministratore (previa autenticazione, pagina **admin.jsp**) di visualizzare lo stato complessivo delle password, gruppo per gruppo, con l'indicazione per ciascuna di quanti giorni residui rimangono prima dell'obbligo di modifica.

ESERCIZIO 2 (11 punti)

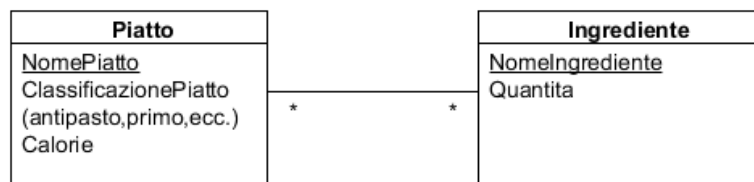
Si realizzi un'applicazione Web, principalmente basata su tecnologie **Java servlet**, **Javascript** e **AJAX**, per il calcolo concorrente della **sottrazione fra due matrici rettangolari A e B**.

L'applicazione Web deve consentire a un utente di inserire i valori interi degli elementi di due matrici A e B con lo stesso numero di righe x e colonne y ; per semplicità si supponga che x e y siano numeri pari. Solo una volta che i valori degli elementi sono stati tutti inseriti, deve comparire una coppia di pulsanti: il primo per comandare il calcolo sequenziale, single-threaded e server-side della sottrazione A-B; il secondo per comandare lo stesso calcolo ma in modo concorrente, multi-threaded e sempre server-side. In particolare, in caso multi-threaded, l'operazione deve essere svolta server-side da 4 thread concorrenti, ciascuno operante su una sottomatrice di $x/2$ righe e $y/2$ colonne.

Infine, deve essere data la possibilità all'amministratore (previa autenticazione, pagina **admin.jsp**) di visualizzare lo stato complessivo delle sessioni utente in atto: se il numero delle sessioni attive è superiore a 10 e il numero delle operazioni di calcolo richieste nell'ultima ora è superiore a 1000, allora deve comparire un pulsante che consenta all'amministratore di forzare la terminazione di tutte le sessioni in corso.

ESERCIZIO 3 (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 Piatto e Ingrediente del diagramma UML** con le corrispondenti **tabelle relazionali derivate dalla progettazione logica del diagramma** stesso.



Nel dettaglio, dopo aver **creato da applicazione Java le tabelle** all'interno del proprio **schema** nel database **TW_STUD** di **DB2** (esplicitando tutti i **vincoli** opportuni di K e FK), 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:

- istanziare alcuni JavaBean "Piatto", e relativi JavaBean "Ingrediente", rendendoli persistenti rispetto alla base di dati associata al diagramma UML al punto sopra;
- aggiungere a tutti i piatti l'ingrediente "*ingrediente segreto*" dello chef, con associata quantità nulla;
- aggiornare la base di dati, aumentando del 2% la proprietà *calorie* dei piatti classificati come "*dolci*";
- restituire *il contenuto della base di dati* producendone una stampa opportunamente formattata sul file **Ricettario.txt**;

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

N.B. L'implementazione **deve limitarsi** al solo **DBMS DB2**. La soluzione Java **deve sfruttare esplicitamente il mapping N-M** specificato nell'UML. **Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.**