

Tecnologie Web T
Prova d'Esame di Programmazione – 29 Giugno 2020 – Versione A

Tempo a disposizione: 90 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

Polling.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Pippo.zip	file zip contenente il sorgente java/class e txt per punto 2

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), ben distribuiti sui 2 esercizi, ovvero in ciascuno dei due esercizi si deve raggiungere un punteggio di almeno 9 punti.

ESERCIZIO 1 (16,5 punti)

Si realizzi una applicazione Web capace di realizzare un modello di funzionamento/interazione in **modalità polling – server push**, basandosi principalmente sulle tecnologie **Java Servlet, JSP e Javascript/AJAX**.

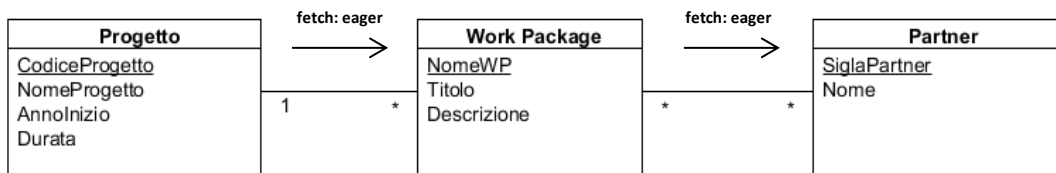
In particolare, l'applicazione Web deve permettere l'accesso ai soli utenti autenticati (tramite username e password), mantenendo server-side l'informazione corrente degli utenti autenticati e con sessioni attive. Dopo l'autenticazione, una servlet deve mostrare a ogni utente la lista degli utenti autenticati con sessioni attive e tale lista deve essere **aggiornata automaticamente ogni 10 secondi**; la servlet deve inviare questa lista al cliente in formato JSON.

Inoltre, si preveda una pagina JSP di amministrazione che permetta di andare a rimuovere in ogni momento righe dalla lista di utenti autenticati descritta sopra, producendo ovviamente il side effect di costringere eventualmente gli utenti cancellati a ri-autenticarsi.

Tecnologie Web T
Prova d'Esame di Programmazione – 29 Giugno 2020 – Versione A

ESERCIZIO 2 (16,5 punti)

Partendo dalla realtà illustrata nel diagramma UML di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **Pattern DAO** in grado di “mappare” efficientemente e con uso di ID surrogati il modello di dominio rappresentato dai **JavaBean** “**Progetto**”, “**Work Package**” e “**Partner**” 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 PK e FK), implementato i **JavaBean** e realizzato le **classi relative al Pattern DAO** per l'accesso **CRUD** alle tabelle, si richiede la realizzazione dei seguenti **metodi**: (i) un metodo che restituisca i nomi dei partner che lavorano nel progetto di nome “Pippo”; (ii) un metodo che restituisca il nome del progetto con associato il numero maggior di partner.

Si crei poi un main di prova in grado di:

- (a) inserire due o più tuple nelle tabelle di interesse;
- (b) fare uso corretto dei metodi realizzati ai punti precedenti al fine di produrre la stampa dei risultati sul file **Pippo.txt**;

N.B. L'implementazione **deve limitarsi** al solo **DBMS DB2**. La soluzione Java **deve sfruttare esplicitamente i mapping N-M e 1-N/N-1** specificati nell'UML. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.