Tecnologie Web T
16 Giugno 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**):

Group_Purchase.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Navigator.zip	file zip contenente il sorgente java/class e pagine Web per punto 2
Players.zip	file zip contenente il sorgente java/class 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 JSP, Java servlet e Javascript, per ***l'acquisto di gruppo*** di prodotti da un sito di e-commerce.
L'applicazione deve consentire ad ogni utente di autenticarsi tramite username e password; come ipotesi semplificativa, non è possibile per uno stesso utente avere più sessioni correntemente attive, ad esempio da dispositivi differenti. Ogni utente appartiene ad uno e un solo gruppo, staticamente noto lato server-side (ad esempio, id_gruppo come info contenuta in struttura dati che mantiene anche username e password).
Superata l'autenticazione, l'utente deve ricevere, via JSON, il catalogo dei prodotti disponibili, descritti come id_prodotto, breve testo di presentazione, costo, numero di item disponibili. Ogni utente potrà selezionare prodotti da inserire nel proprio ***carrello di gruppo*** e potrà richiedere la finalizzazione dell'acquisto. La finalizzazione dell'acquisto sarà effettuata (prodotti venduti nell'acquisto di gruppo non più disponibili per altri gruppi d'acquisto) solo quando tutti i clienti con sessioni attive di quel gruppo avranno premuto il pulsante "Finalizza". Se tutte le sessioni utente di un gruppo terminano senza essere giunti al completamento della finalizzazione, i dati contenuti nel carrello di gruppo sono persi.
In caso di finalizzazioni di gruppo "in confitto" da parte di gruppi differenti, ad esempio se gruppo1 vuole acquistare n item di un prodotto di cui rimangono solo k<n item, la richiesta di gruppo1 deve fallire; agli utenti di tale gruppo deve essere inviata la nuova versione del catalogo JSON tramite AJAX alla prossima operazione richiesta.

Tecnologie Web T
16 Giugno 2017 – Compito

ESERCIZIO 2 (11 punti)

Si realizzi un'applicazione Web di ***assistenza a turisti pedoni***, principalmente basata su tecnologie Javascript e AJAX, per il prefetching di informazioni geolocalizzate.

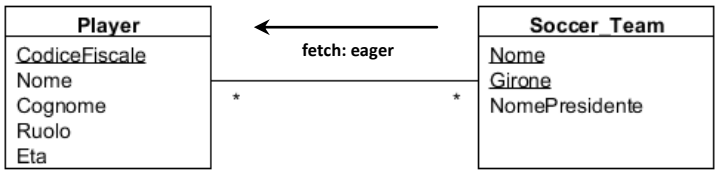
L'applicazione Web deve consentire ad ogni utente di indicare la sua posizione corrente e di vedersi visualizzare, in risposta, le info relative a tutte le eventuali attrazioni turistiche visibili dalla sua posizione corrente. Le informazioni mantenute lato servitore sulle attrazioni turistiche includono coordinate cartesiane, nome e semplice testo di descrizione. Si supponga che l'informazione di posizione dell'utente sia nota in ogni momento all'applicazione (viene inserita tramite form; coordinate cartesiane (x,y) espresse in metri); ogni attrazione turistica è visibile se e solo se la sua distanza dall'utente è minore di k metri (con k parametro configurabile a livello di applicazione) e se ci troviamo in situazione non affollata (meno di 10 altri utenti nel raggio di 100 metri).

Per ottenere un comportamento fortemente responsive dell'applicazione Web, si faccia in modo che il browser Web faccia il download in background, in modo asincrono e concorrente, delle informazioni relative alle attrazioni che saranno eventualmente visibili nella posizione futura più probabile per l'utente. Per semplicità, si supponga che tale posizione futura più probabile sia sempre $(x_{t2},y_{t2}) = (x_{t1}+50, y_{t1})$.

ESERCIZIO 3 (11 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*** del diagramma UML con le corrispondenti ***tabelle relazionali*** derivate dalla progettazione logica del diagramma stesso.

Nel dettaglio, la gestione degli ID surrogati è demandata a una apposita classe `IdBroker()` la cui logica si basa sull'uso del costrutto `SEQUENCE`.



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*** opportuni), ***implementato i JavaBean*** e ***realizzato le classi relative al pattern DAO per l'accesso CRUD*** alle tabelle, si richiede ***l'implementazione della seguente operazione*** (a livello di logica di business):

- "Formazione delle squadre di calcio in capo al presidente *Rossi Mario*"; in particolare, per ogni squadra (di cui si richiedono gli attributi *Nome,Girone*), la formazione dei giocatori del team deve essere esportata mediante i soli attributi ordinati *Cognome, Nome, Eta*.

Si crei un ***main di prova*** che: (i) inserisca due o piu' tuple nelle tabelle di interesse; (ii) faccia uso corretto dell'operazione realizzata al punto precedente al fine di produrre una stampa opportunamente formattata del risultato sul file **Players.txt**.

N.B. L'implementazione del pattern DAO deve limitarsi al solo DBMS DB2. La soluzione deve sfruttare direttamente i mapping M-N specificati nell'UML (sulla base dei versi di navigazione esplicitati) e propendere per il caricamento dei dati indicato nello stesso. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.