Tecnologie Web T
26 Gennaio 2018 – Compito

Tempo a disposizione: 3 ore

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

CipherCounting.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Whtml.zip	file zip contenente il sorgente javascript e pagine Web per punto 2
Cameriere.zip	file zip contenente il sorgente java/class e il 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 (11 punti)

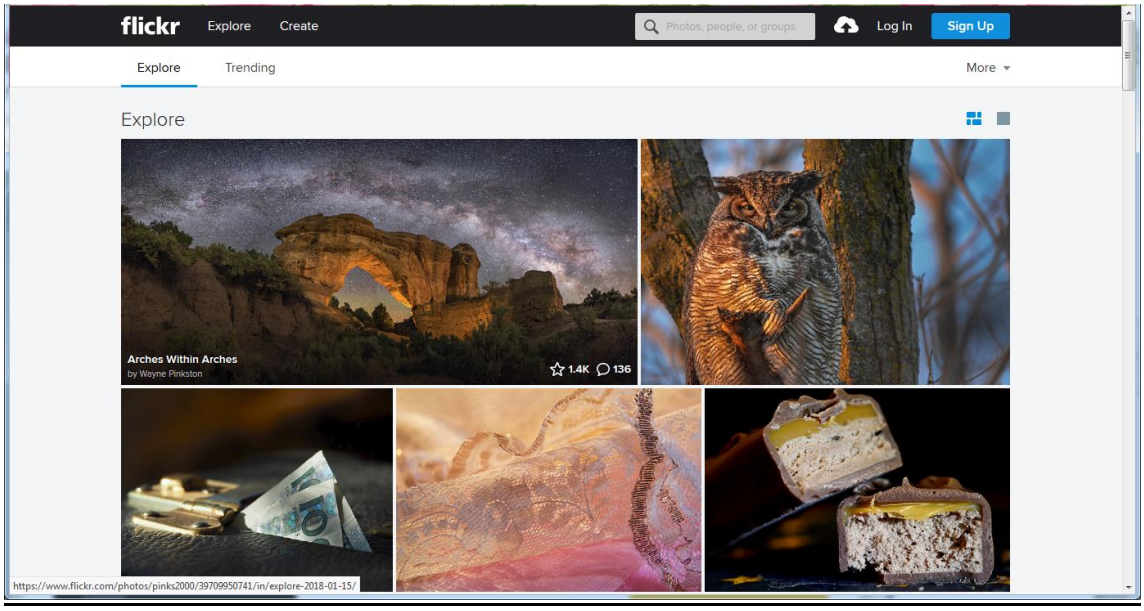
Si realizzi un'applicazione Web, che usi prevalentemente tecnologie **Javascript**, Java **servlet**, **AJAX** e **JSON** per il **conteggio concorrente delle occorrenze di caratteri numerici in x file di testo** mantenuti in URL differenti.
In particolare, l'applicazione Web deve essere costituita da una pagina **home.html** che consente all'utente di inserire un numero naturale **x** che indichi il numero di file di testo da considerare; da home.html si deve essere ri-diretti x volte a una pagina per l'inserimento di una stringa, ciascuna delle quali rappresenta uno degli x URL di file di testo da scaricare. Al termine dell'inserimento stringhe, l'applicazione (lato cliente o lato servitore?) dovrà effettuare il conteggio concorrente delle occorrenze dei caratteri numerici (inclusi fra '0' e '9') in tutti gli x file di testo, nonché il calcolo del numero totale, che dovrà essere restituito al cliente via JSON.
Problemi in caso di accesso concorrente agli stessi file di testo da parte di più clienti? Come evitarli? Si risponda a tali domande tramite commenti nel file sorgente.

Tecnologie Web T
26 Gennaio 2018 – Compito

ESERCIZIO 2 (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** dello "snapshot" del **sito Web flickr** di seguito riportato. Nello specifico, la pagina visualizza il contenuto relativo alla voce "*Explore*" del Menù "orizzontale" nella barra grigio scuro in alto a sinistra, e ancora "*Explore*" tra le sotto-opzioni messe a disposizione nella barra bianca (la scelta è evidenziata dalla linea di sottolineatura azzurra della voce "Explore"), optando per la modalità di visualizzazione di immagini "a mosaico" piuttosto che di una singola immagine alla volta (evidenziata dal color azzurro della prima icona posta sul lato destro della finestra principale di sfondo grigio chiaro). In particolare, si deve prevedere che al passaggio del mouse su ogni immagine visualizzata nella finestra principale, si mostri nell'immagine stessa anche il titolo della fotografia assieme al nome dell'autore, sul lato sinistro, mentre sul lato destro si riporti il numero di *preferenze* ricevute (icona "stella") e il numero di *commenti* inseriti (icona "nuvola") dagli utenti, come evidenziato per la prima immagine della finestra principale.
Infine, si preveda che il campo di ricerca nella barra grigio scuro in alto possa ammettere solo stringhe alfabetiche (tutte maiuscole o minuscole) composte da una sola parola.

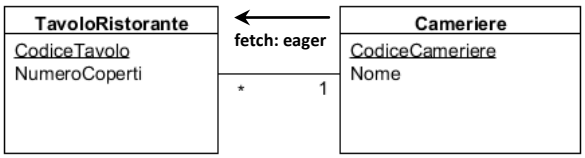
N.B. La soluzione NON deve far uso del costrutto HTML `frame`. Per rappresentare le immagini/icone riportate nello snapshot, si utilizzino figure di esempio a piacere.



Tecnologie Web T
26 Gennaio 2018 – Compito

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 un'apposita classe `IdBroker()` la cui logica è lasciata a scelta libera dello studente.



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 delle seguenti operazioni** (a livello di logica di business) mediante metodi opportuni:

- "Nome del cameriere a cui sono stati assegnati più tavoli";
- "Per ogni cameriere, il numero totale di coperti serviti".

Si crei quindi un **main di prova** che:

(i) inserisca diverse tuple nelle tabelle di interesse;
(ii) faccia uso dei metodi realizzati precedentemente sull'istanza del DB appena creata e produca la stampa dei risultati delle relative operazioni opportunamente formattata sul file **Cameriere.txt**.

N.B. L'implementazione del pattern DAO deve limitarsi al solo **DBMS DB2**. La **soluzione deve sfruttare direttamente i mapping specificati nell'UML e propendere per il caricamento dei dati indicato nello stesso**.