

**Tecnologie Web T**  
**11 Giugno 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>):

<b>Portale.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 1
<b>Word.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 2
<b>Accertamento.zip</b>	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 permetta a utenti autenticati di accedere a contenuti Web scaricati server-side da siti multipli e di segnalare il proprio interesse per i contenuti considerati più rilevanti. L'applicazione deve essere basata principalmente su **tecnologie Java servlet e JSON**.

In particolare, l'applicazione dovrà avere una pagina iniziale **home** che richieda all'utente nome e cognome (considerati identificatori univoci) e che gli permetta di aprire una sessione di interazione. Dopo l'apertura della sessione verrà richiesto all'utente di inserire tre URL in una pagina **portale**; la pressione di un pulsante, solo dopo l'inserimento dei tre URL, dovrà scatenare l'invocazione di una servlet che si occuperà di scaricare i 3 documenti relativi. La servlet dovrà prendere i contenuti di tali documenti  $x_i$  e aggregarli in un unico documento complessivo da trasferire verso il browser in formato JSON; il browser dovrà visualizzare tale contenuto, ponendo una casella di selezione a fianco di ogni documento  $x_i$  per dare la possibilità di segnalare il proprio interesse verso  $x_i$ ; i documenti selezionati dovranno essere visualizzati nella porzione bassa di **portale** fino alla terminazione della sessione utente.

Inoltre, si preveda che, per alleggerire il carico del server, nel caso in cui vi siano più di 50 utenti con sessioni attive, la funzionalità di stato di sessione sia disabilitata e tutti gli utenti accedano al servizio in modo stateless; in altri termini, alla richiesta di apertura della 51<sup>a</sup> sessione utente, tutti gli utenti (inclusi quelli che avevano sessione attiva in precedenza) devono accedere solo in modalità stateless fino alla terminazione della loro sessione.

**Tecnologie Web T**  
**11 Giugno 2015 – Compito**

**ESERCIZIO 2 (11 punti)**

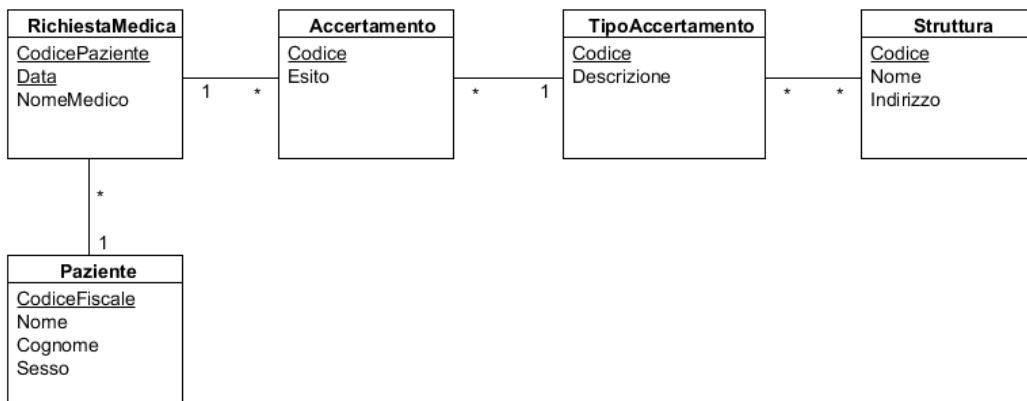
Si realizzi un'applicazione Web per il word processing parallelo AJAX-based di un file di testo, chiamato file.txt e memorizzato server-side nella cartella accessibile /resources/file.txt. In particolare, l'applicazione dovrà avere una pagina iniziale Javascript in cui l'utente possa inserire un numero x fra 10 e 99 (input da validare; il numero x rappresenta il numero di righe di testo che ogni richiesta AJAX dovrà ottenere concorrentemente).

Automaticamente, al termine dell'inserimento e della validazione di x, l'applicazione deve invocare 5 volte, tramite AJAX in modo asincrono e concorrente, la stessa servlet **wordProcessing**; la servlet si occuperà semplicemente di andare a considerare il blocco di righe successive  $[i*x, i*x+x)$ , dove i è il numero dell'invocazione compreso in  $[0,4]$ , e di restituire tali righe **in formato txt** dopo avere sostituito ogni occorrenza di carattere MAIUSCOLO nel corrispondente minuscolo; per semplicità si supponga che file.txt contenga sempre un numero sufficiente di righe, ad esempio maggiore di 500.

Si commenti nel codice quale è il modello di esecuzione concorrente utilizzato e se possono sussistere problemi di interferenza fra le eventuali istanze multiple/singole della classe servlet messe in esecuzione concorrentemente dalle chiamate AJAX.

**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** con le corrispondenti **tabelle relazionali derivate dalla progettazione logica del diagramma** stesso.



Nel dettaglio, **dopo aver creato da applicazione Java gli schemi delle tabelle** all'interno del proprio schema nel database **TW\_STUD** di **DB2** (esplicitandone tutti i vincoli derivati dal diagramma UML), 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:

- inserire due o più tuple nelle tabelle di interesse;
- determinare i) il nome del medico che ha richiesto più accertamenti di tipo “risonanza” per pazienti di sesso femminile; ii) il nome delle strutture in cui possono essere eseguiti accertamenti prescritti dal medico “Mario Rossi” a partire dal 1 giugno 2014;
- stampare i risultati ottenuti al punto precedente sul file **Accertamento.txt**;

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

**N.B.** La soluzione deve sfruttare i mapping M-N e 1-N specificati nel diagramma UML. Ogni ulteriore scelta fatta dallo studente deve essere opportunamente giustificata mediante opportuni commenti nel codice.