

Prova d'Esame—Tecnologie Web LA

A.A. 2008/2009

Tempo Disponibile: 3 ore

Ai candidati è consentita la consultazione di manuali e materiale didattico

Esercizio 1. Modellazione

Si progetti il DTD per la gestione di documenti XML che modellano le visite guidate ad una oasi naturale. Ogni visita è caratterizzata da un codice univoco, un luogo di partenza, ed una data di inizio della visita. Inoltre ogni visita è caratterizzata da una lista di uno o più partecipanti. Ogni partecipante è caratterizzato da un nome, da un cognome, eventualmente da un soprannome, e dalla lista di uno o più animali che questi vorrebbe vedere.

Esercizio 2. Programmazione Client Side

Si richiede di implementare la sola parte client side di un applicativo Web in tecnologia AJAX per il reperimento di informazioni sulla disponibilità di posti per le visite turistiche ad una oasi naturale. La pagina è costituita da una form contenente *quattro campi di testo* editabili dall'utente. I campi di testo devono consentire all'utente di inserire le date di suo interesse specificando giorno e mese.

Controllo Disponibilità

Partenza
Giorno:
Mese:

Arrivo
Giorno:
Mese:

Figura 1. Form per l'inserimento delle date

La form è dotata di un bottone che consente di inviare la request al server. Le informazioni devono essere inviate al server in modo asincrono tramite una chiamata GET passando le date di interesse dell'utente. A seguito della pressione del bottone la pagina deve:

- verificare che entrambe le date siano costituite da soli caratteri numerici. Verificare che il giorno sia compreso fra 1 e 31, e che il mese sia compreso fra 1 e 12. Inoltre si deve verificare che la prima data preceda la seconda. Nel caso in cui si verificano errori, deve essere inserito

sotto la form un messaggio che informa l'utente dell'errore che ha commesso. Il messaggio deve specificare il tipo di errore (es. date non numeriche, errore nella specifica del mese, etc.);

- nel caso in cui l'utente abbia inserito i dati correttamente deve essere inviata la richiesta al server. Si richiede che i risultati contenuti nella risposta del server vengano presentati in una tabella che deve essere inserita sotto la form. In particolare, la tabella è organizzata per righe ed ogni riga riporta data, e ora di partenza della visita turistica, i posti ancora disponibili ed il prezzo del biglietto (vedi Tabella1);

Data	Ora	Posti Disponibili	Prezzo
Data	Ora	Posti Disponibili	Prezzo

...

Data	Ora	Posti Disponibili	Prezzo
------	-----	-------------------	--------

Tabella 1: tabella visualizzata all'utente nel caso in cui i dati siano stati inseriti correttamente

Si assume che il server restituisca i dati in formato XML come segue:

```
<?xml version='1.0' encoding='UTF16'?>
<visiteGuidate>
  <visita>
    <data>Data</data>
    <ora>Ora</ora>
    <posti>Posti Disponibili</posti>
    <prezzo>Prezzo</prezzo>
  </visita>
  <visita>
    <data>Data</data>
    <ora>Ora</ora>
    <posti>Posti Disponibili</posti>
    <prezzo>Prezzo</prezzo>
  </visita>
    .....
  <visita>
    <data>Data</data>
    <ora>Ora</ora>
    <posti>Posti Disponibili</posti>
    <prezzo>Prezzo</prezzo>
  </visita>
</visiteGuidate>
```

dove *Data*, ed *Ora* rappresentano rispettivamente la data e l'ora della visita, *Posti Disponibili* i posti ancora disponibili per la prenotazione e *Prezzo* il prezzo del biglietto per la visita. I dati si riferiscono all'intervallo di tempo fra le due date specificate dall'utente. Si assume per semplicità che esista sempre almeno una visita disponibile nel periodo richiesto dall'utente.

Esercizio 3. Programmazione ServerSide

Si richiede di implementare una sezione dell'applicazione Web per la gestione di visite guidate ad una oasi naturale. In particolare, si richiede che il candidato implementi due distinte Java Server Pages (JSP) che consentano di inserire nuove visite all'oasi (*InserisciVisita*) e di visualizzare le visite programmate (*VisualizzaVisite*). Le JSP hanno accesso ad un comune database costituito per semplicità da una sola tabella strutturata come segue (vedi Tabella 2):

Codice Visita	Data	Ora	Posti Disponibili	Prezzo
Codice Visita	Data	Ora	Posti Disponibili	Prezzo

...

Codice Visita	Data	Ora	Posti Disponibili	Prezzo
---------------	------	-----	-------------------	--------

Tabella 2: tabella che realizza la base di dati per la gestione delle visite all'oasi naturalistica

dove *Codice Visita* è un intero che rappresenta univocamente il codice della visita, mentre *Data*, *Ora*, *Posti Disponibili* e *Prezzo* sono interi che rappresentano rispettivamente la data, l'ora, il numero di posti ancora disponibili per la prenotazione ed il prezzo del biglietto della visita.

La JSP *InserisciVisita* consente di inserire una nuova visita guidata nella base dati. La JSP richiede il passaggio di *Codice Visita*, *Data*, *Ora*, *Posti Disponibili* e *Prezzo* della visita via HTTP GET. La JSP, una volta ricevuti i parametri, verifica che non esista nel database alcuna visita con il *Codice Visita* Specificato. In caso esista una visita con lo stesso *Codice Visita* nel database allora l'inserimento non viene effettuato e viene restituita all'utente una pagina che segnala l'errore nell'inserimento del codice. Inoltre, la JSP verifica che non esista nel database un'altra visita prevista nella stessa data ed alla stessa ora. Qualora esistesse una registrazione con medesima data ed ora non deve essere effettuato l'inserimento nel database e deve essere restituita all'utente una pagina che segnala l'errore riscontrato. Nel caso in cui i tutti i controlli vadano a buon fine la JSP inserisce la nuova visita nel database e restituisce all'utente una pagina di conferma dell'avvenuto inserimento.

La JSP *VisualizzaVisite* consente di mostrare all'utente una pagina che include le sole visite guidate in cui è disponibile almeno un posto libero programmate per date successive alla data specificata dall'utente. Per semplicità assumiamo che la data sia modellata come un intero positivo. La JSP richiede il passaggio di una richiesta HTTP GET che specifica la data di interesse. Una volta ricevuta la richiesta la JSP controlla che esista almeno una visita con un posto disponibile dopo la data specificata. In questo caso, viene presenta all'utente una pagina in cui è inclusa la lista delle visite disponibili. In particolare, ogni riga della lista include varie informazioni quali: il codice della visita, il numero di posti disponibili ed il prezzo della visita (vedi esempio sotto).

- Codice Visita, Posti Disponibili, Prezzo
- Codice Visita, Posti Disponibili, Prezzo
- ...
- Codice Visita, Posti Disponibili, Prezzo

Nel caso in cui non esista nel database alcuna visita o che siano esauriti tutti i posti disponibili per le visite programmate dopo la data specificata, deve essere mostrata all'utente una pagina che segnala l'errore.

Si richiede che il candidato implementi l'accesso al database utilizzando il pattern DAO. Si richiede inoltre che il database sia implementato utilizzando un DBMS relazionale (es. hsqldb) oppure, in alternativa, utilizzando semplici file di testo.

Prova d'Esame—Sistemi Operativi LS

A.A. 2008/2009

Tempo Disponibile: 3 ore

Ai candidati è consentita la consultazione di manuali e materiale didattico

Un autobus turistico consente di effettuare il giro panoramico di Bologna. L'autobus può contenere al più N_{max} passeggeri. Per aumentare i profitti si è deciso che l'autobus parta solo quando sono entrati N_{max} passeggeri. Inoltre, i passeggeri possono uscire solo quando il giro turistico è terminato.

L'autobus ha due porte che consentono l'entrata e l'uscita dei passeggeri. I passeggeri possono scegliere di entrare ed uscire indifferentemente da una delle due porte.

I passeggeri entrano nell'autobus in gruppi di al più N_{pas} passeggeri. Si noti che ovviamente $1 \leq N_{pas} \leq N_{max}$. Per semplicità si assume che tutti i passeggeri di un gruppo passino dalla stessa porta in un tempo non nullo. Inoltre, all'ingresso dell'autobus si suppone che i gruppi di passeggeri non siano divisibili, ovvero che non sia possibile caricare nell'autobus solo alcuni membri di un gruppo. Inoltre, per ogni porta dell'autobus, sia la salita, sia la discesa viene effettuata da un gruppo di passeggeri alla volta.

Si richiede che il candidato implementi in Java un monitor che realizza una politica di gestione dell'autobus con i seguenti vincoli:

- I gruppi di passeggeri più numerosi hanno la precedenza all'ingresso rispetto ai gruppi meno numerosi.
- Tutti i gruppi di passeggeri hanno la stessa priorità nell'uscita dall'autobus
- L'ingresso nell'autobus può avvenire solo quando sono scesi tutti i gruppi di passeggeri che lo occupavano nel giro turistico precedente

Suggerimento: i passeggeri che viaggiano da soli possono essere considerati come gruppi composti da un solo passeggero

I commenti al codice sono considerati parte integrante del giudizio