

## Prova d'Esame

Cognome e Nome \_\_\_\_\_ Matricola \_\_\_\_\_

Quesito svolto base ? intermedio ? avanzato ?

Tempo a Disposizione: 4 ore e mezza

### Libri in Linea

#### Livello Base

E' necessario scrivere un'applicazione Web relativa ad un sito per la vendita di libri. L'applicazione deve consentire di eseguire il seguente caso d'uso.

#### “Utente acquista libro”

##### Scenario Principale

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per specificare il titolo del libro da acquistare
- Il sistema verifica che nella base di dati esista un libro con il titolo specificato. I libri venduti sono memorizzati in una tabella **Libri** della base di dati che contiene il codice del libro, il titolo, l'autore, il prezzo e il numero di copie disponibili.
- Se il libro esiste ed è disponibile almeno una copia, l'utente accede ad uno schermo (**acquisto.jsp**) nel quale sono riassunti i dati relativi al libro; lo schermo contiene inoltre una maschera attraverso la quale l'utente fornisce (a) il suo nome e cognome; (b) il codice della sua carta di credito
- Procedendo all'acquisto, il sistema aggiorna il numero di copie disponibili del libro nella base di dati.  
*NOTA: non è necessario registrare nella base di dati i dati relativi all'acquisto.* L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati del libro e dell'acquisto); lo schermo contiene un collegamento allo schermo iniziale nel caso l'utente voglia eseguire un ulteriore acquisto

*Scenario alternativo:* Il libro non esiste oppure il numero di copie è pari a 0

- L'utente accede ad uno schermo (**errore.jsp**) in cui viene visualizzato un messaggio di errore ed un collegamento allo schermo principale per effettuare una nuova operazione

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 1**.
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**.
- Non è necessario provvedere alla convalida dei dati sottomessi dall'utente.
- Non è necessario provvedere alla convalida del codice HTML prodotto dalle pagine JSP.

## Livello Intermedio

E' necessario scrivere un'applicazione Web relativa per ad un sito per la vendita di libri. L'applicazione deve consentire di eseguire i seguenti caso d'uso.

### “Utente effettua login”

#### *Scenario Principale*

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per fornire nome utente e password
- Il sistema autentica l'utente accedendo ad una tabella **Utenti** della base di dati, che contiene il nome utente, la password, il nome e cognome dell'utente (**NOTA:** è necessario prevedere tra gli altri un utente “Mario Rossi” con nome utente **utente** e password **utente**)
- Se l'autenticazione ha successo, l'utente accede allo schermo principale del caso d'uso successivo (**gestione.jsp**)

*Scenario alternativo:* nome utente o password scorretta

- Se l'autorizzazione fallisce, l'utente ritorna allo schermo iniziale (**index.jsp**) in cui viene visualizzato un messaggio di errore

### “Utente gestisce libri”

#### *Scenario Principale*

- L'utente accede ad uno schermo (**gestione.jsp**) nel quale sono riassunti i dati dell'utente (nome, cognome). Lo schermo contiene inoltre un elenco dei libri disponibili. I libri sono memorizzati in una tabella **Libri** della base di dati che contiene il codice del libro, il titolo, l'autore, il prezzo e il numero di copie disponibili. La pagina contiene inoltre le maschere necessarie ad inserire ed eliminare libri.
- Per consentire l'inserimento, la pagina iniziale (**gestione.jsp**) contiene anche una maschera attraverso la quale l'utente può inserire i dati di un nuovo libro (codice, titolo, autore, prezzo e numero di copie). Se l'utente sottomette la maschera, il sistema convalida i dati forniti; se i dati sono corretti il sistema inserisce il nuovo libro nella base di dati e l'utente accede ad uno schermo (**confermaInserimento.jsp**) nel quale vengono riassunti i dati del libro inserito; lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**gestione.jsp**) nel caso l'utente voglia eseguire una ulteriore operazione; (b) un collegamento ad uno schermo di uscita (**uscita.jsp**) attraverso il quale l'utente può abbandonare l'applicazione
- Per consentire l'eliminazione, la pagina contiene un bottone in corrispondenza di ciascun libro visualizzato nell'elenco dei libri disponibili; utilizzando il bottone l'utente può eliminare il libro corrispondente dalla base di dati. Se l'utente decide di eliminare un libro, accede ad uno schermo (**confermaEliminazione.jsp**), in cui vengono riassunti i dati del libro eliminato e viene confermata l'eliminazione; lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**gestione.jsp**) nel caso l'utente voglia eseguire una ulteriore operazione; (b) un collegamento ad uno schermo di uscita (**uscita.jsp**) attraverso il quale l'utente può abbandonare l'applicazione

*Scenario alternativo:* dati scorretti del libro inserito

- Se la convalida fallisce, l'utente ritorna allo schermo di immissione (**gestione.jsp**) in cui viene visualizzato un messaggio di errore

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 1** oppure un'architettura di tipo **Modello 2** basata sull'utilizzo del framework didattico **pinco**
- E' necessario provvedere alla **convalida** dei dati sottomessi dall'utente
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**
- Non è necessario provvedere alla convalida del codice HTML prodotto dalle pagine JSP
- Per la gestione della base di dati è necessario utilizzare **PostgreSQL**, e il relativo driver JDBC

## Livello avanzato

E' necessario scrivere un'applicazione Web relativa ad un sito per la vendita di libri. L'applicazione deve consentire di eseguire i seguenti caso d'uso.

### “Utente acquista libri”

#### *Scenario Principale*

- L'utente accede ad uno schermo (**index.jsp**) che contiene un messaggio di benvenuto, ed un pulsante o un collegamento (link) per entrare nell'applicazione.
- Scegliendo di entrare, si accede ad uno schermo (**iniziale.jsp**) che contiene un elenco dei libri venduti. I libri sono memorizzati in una tabella **Libri** della base di dati che contiene il codice del libro, il titolo, l'autore, il prezzo. L'elenco deve prevedere una maschera attraverso cui l'utente può selezionare uno dei libri da acquistare e il numero di copie richieste (NOTA: si consiglia di prevedere un controllo di tipo “select” o “radio” per specificare il libro per cui effettuare l'offerta; si può supporre che la libreria abbia un numero molto alto di copie disponibili di ciascun libro, per cui non è necessario effettuare verifiche sulla disponibilità di copie)
- A ciascun utente è associato “*carrello degli acquisti*”, inizialmente vuoto. Lo schermo iniziale visualizza il contenuto del carrello degli acquisti dell'utente.
- Se l'utente decide di acquistare un libro, il sistema convalida i dati forniti; se i dati sono corretti, il libro viene inoltre aggiunto al carrello degli acquisti. L'utente accede ad uno schermo (**acquisto.jsp**) in cui sono riassunti i dati del libro selezionato; lo schermo contiene inoltre (a) un collegamento allo schermo iniziale (**iniziale.jsp**) per procedere ad altri acquisti (b) un collegamento ad uno schermo (**pagamento.jsp**) che consente di procedere al pagamento di tutti i libri contenuti nel carrello; (c) un collegamento allo schermo di uscita (**uscita.jsp**) attraverso il quale l'utente può abbandonare l'applicazione senza effettuare l'acquisto
- Se l'utente decide di procedere con il pagamento, accede ad uno schermo (**pagamento.jsp**) in cui è riassunto il contenuto del carrello degli acquisti; per ciascuna voce contenuta del carrello è disponibile un pulsante che consente di eliminare la voce dal carrello; se l'utente decide di eliminare una voce dal carrello, viene aggiornato il contenuto del carrello visualizzato sullo schermo. Lo schermo contiene inoltre una maschera per effettuare il pagamento; attraverso la maschera l'utente fornisce (a) il nome e cognome; (b) il proprio numero di carta di credito (per semplicità di 6 cifre)
- Il sistema convalida i dati forniti; se i dati sono corretti il sistema registra i dati nella base di dati. I dati dell'acquisto vengono registrati in una tabella **CarrelloAcquisti** della base di dati che contiene il codice dell'acquisto, il nome e il cognome dell'acquirente, il codice della carta di credito e l'importo complessivo; il contenuto del carrello degli acquisti viene registrato in una tabella **VoceCarrello** (ciascuna voce si riferisce ad un libro che fanno parte dell'acquisto, con il relativo numero di copie)
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i dati dell'acquisto effettuato; lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**iniziale.jsp**) nel caso l'utente voglia eseguire una ulteriore operazione; (b) un collegamento allo schermo finale (**uscita.jsp**) attraverso il quale l'utente può abbandonare l'applicazione

#### *Scenario alternativo: dati scorretti*

- Se la convalida dei dati fallisce, l'utente ritorna allo schermo di immissione in cui viene visualizzato un messaggio di errore

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 2** basata sul framework didattico **pinco**, ed uno strato di persistenza basato su DAO e gestione delle transazioni
- E' possibile scegliere se utilizzare o meno un pool di connessioni
- E' necessario provvedere alla **convalida** dei dati sottomessi dall'utente
- Il codice HTML prodotto deve essere conforme al **DTD XHTML-Basic**. E' necessario procedere alla convalida del codice prodotto utilizzando il validatore **xmlvalid**
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**.