

Prova d'Esame

Cognome e Nome _____ Matricola _____

Quesito svolto base ? intermedio ? avanzato ?

Tempo a Disposizione: 4 ore e mezza

Home Banking

Livello Base

E' necessario scrivere un'applicazione Web che consente di gestire un servizio di "home banking".

L'applicazione deve consentire di eseguire il seguente caso d'uso.

“Utente effettua versamento”

Scenario Principale

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per specificare i dati del conto corrente. Nella maschera è necessario specificare il codice del conto corrente (un numero intero)
- Il sistema verifica che nella base di dati esista un conto corrente con il codice specificato. I conti correnti sono memorizzati in una tabella **ContiCorrenti** della base di dati che contiene il numero del conto, il nome e cognome dell'utente, il saldo in Euro.
- Se il conto esiste, l'utente accede ad uno schermo (**versamento.jsp**) nel quale sono riassunti i dati del conto corrente; lo schermo contiene inoltre una maschera attraverso la quale l'utente fornisce l'importo da versare
- Il sistema aggiorna il saldo del conto effettuando il versamento
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati del conto ed importo versato); lo schermo contiene un collegamento allo schermo iniziale nel caso l'utente voglia eseguire una ulteriore operazione

Scenario alternativo: Il conto corrente non esiste

- Se, viceversa, il conto corrente non esiste, 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 che consente di gestire un servizio di "home banking".
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 e il suo codice di conto corrente (NOTA: per semplicità si suppone che ciascun utente possa avere al massimo un conto corrente)
- Se il conto esiste, l'utente accede allo schermo principale del caso d'uso successivo (**versamento.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 effettua versamento”

Scenario Principale

- L'utente accede ad uno schermo (**versamento.jsp**) nel quale sono riassunti i dati dell'utente (nome, cognome, numero di conto corrente, saldo attuale). I dati relativi ai conti correnti sono memorizzati in una tabella **ContiCorrenti** della base di dati che contiene (a) il codice del conto; (b) il saldo in Euro. Lo schermo contiene una maschera attraverso cui l'utente fornisce l'importo in Euro da versare; l'importo minimo che è possibile versare è di 100 Euro
- Il sistema convalida i dati forniti; se i dati sono corretti il sistema aggiorna il saldo del conto effettuando il versamento
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati del conto aggiornati ed importo versato); lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**versamento.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: importo scorretto

- Se la convalida fallisce, l'utente ritorna allo schermo di immissione (**versamento.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 che consente di gestire un servizio di "home banking".
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 e il suo codice di conto corrente (NOTA: per semplicità si suppone che ciascun utente possa avere al massimo un conto corrente)
- Se il conto esiste, l'utente accede allo schermo principale del caso d'uso successivo (**bonifico.jsp**)

Scenario alternativo: dati scorretti

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

“Utente effettua bonifico”

Scenario Principale

- L'utente accede ad uno schermo (**bonifico.jsp**) nel quale sono riassunti i dati dell'utente (nome, cognome, numero di conto corrente, saldo attuale). I dati relativi ai conti correnti sono memorizzati in una tabella **ContiCorrenti** della base di dati che contiene (a) il codice del conto; (b) il saldo in Euro. Lo schermo contiene una maschera attraverso cui l'utente fornisce i dati del bonifico da effettuare, ovvero (a) codice del conto corrente di destinazione (un numero intero); NOTA: il conto corrente deve appartenere ad un altro utente della stessa banca; (b) importo da bonificare in Euro
- Il sistema convalida i dati forniti per verificare che il bonifico sia possibile; in questo caso registra il bonifico nella base di dati; i bonifici sono memorizzati nella tabella **Bonifici** che contiene il codice del conto di partenza, il codice del conto di destinazione e l'importo del bonifico. Inoltre, il sistema aggiorna i saldi dei due conti
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati dell'utente e dati del bonifico effettuato); lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**bonifico.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 per il bonifico

- Se la convalida fallisce, l'utente ritorna allo schermo di immissione (**bonifico.jsp**) in cui vengono visualizzati i necessari messaggi 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**.
- Per la gestione della base di dati è necessario utilizzare **PostgreSQL**, e il relativo driver JDBC

Prova d'Esame

Cognome e Nome _____ Matricola _____

Quesito svolto base ? intermedio ? avanzato ?

Tempo a Disposizione: 4 ore e mezza

Fantacalcio

Livello base

E' necessario scrivere un'applicazione Web che consente gestire il sito del Fantacalcio. L'applicazione deve consentire di eseguire il seguente caso d'uso.

“Utente aggiorna risultato partita”

Scenario Principale

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per specificare i dati della partita di cui bisogna aggiornare il risultato. Nella maschera è necessario specificare il nome della squadra che gioca in casa (es: Milan) e il nome della squadra che gioca in trasferta (es: Roma)
- Il sistema verifica che nella base di dati esista una partita tra le due squadre specificate. I dati delle partite sono memorizzati in una tabella **Partite** della base di dati che contiene il nome della squadra in casa, il nome della squadra in trasferta, i goal fatti dalla squadra in casa, i goal fatti dalla squadra in trasferta
- Se la partita esiste, l'utente accede ad uno schermo (**aggiornamento.jsp**) nel quale sono riassunti i dati della partita (il risultato attuale); lo schermo contiene inoltre una maschera attraverso la quale l'utente fornisce il risultato aggiornato (goal della squadra in casa, goal della squadra in trasferta)
- Il sistema aggiorna il risultato della partita
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale viene riassunto il risultato aggiornato; lo schermo contiene un collegamento allo schermo iniziale nel caso l'utente voglia eseguire una ulteriore operazione

Scenario alternativo: la partita non esiste

- Se, viceversa, la partita non esiste, 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 che consente gestire il sito del Fantacalcio. 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
- Se l'autenticazione ha successo, l'utente accede allo schermo principale del caso d'uso successivo (**inserimento.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 inserisce partita”

Scenario Principale

- L'utente accede ad uno schermo (**inserimento.jsp**) nel quale sono riassunti i dati dell'utente (nome e cognome). Lo schermo contiene una maschera attraverso cui l'utente fornisce i dati di una nuova partita ovvero: codice della partita (un numero intero), nome della squadra in casa, nome della squadra in trasferta, goal della squadra in casa, goal della squadra in trasferta. I dati delle partite sono memorizzati in una tabella **Partite** della base di dati
- Il sistema convalida i dati forniti; se i dati sono corretti aggiorna la base di dati inserendo la partita
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i dati della partita inserita; lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**inserimento.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 della partita

- Se la convalida dei dati fallisce oppure esiste già una partita con lo stesso codice, l'utente ritorna allo schermo di immissione (**inserimento.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 che consente gestire il sito del Fantacalcio. 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
- Se l'autenticazione ha successo, l'utente accede allo schermo principale del caso d'uso successivo (**inserimento.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 inserisce partita”

Scenario Principale

- L'utente accede ad uno schermo (**inserimento.jsp**) nel quale sono riassunti i dati dell'utente (nome, cognome). Lo schermo consente all'utente di inserire i dati di una nuova partita. La base di dati contiene due tabelle: (a) una tabella **Squadre** in cui sono rappresentati il nome della squadra e il punteggio in classifica; (b) una tabella **Partite** in cui sono rappresentati il riferimento alla squadra che ha giocato in casa, il riferimento alla squadra che ha giocato in trasferta; i gol segnati dalla squadra che ha giocato in casa; i gol segnati dalla squadra che ha giocato in trasferta. La maschera per l'inserimento della partita deve consentire di specificare (a) la squadra che gioca in casa (attraverso un menu a tendina); (b) la squadra che gioca in trasferta (attraverso un menu a tendina); (c) i goal fatti dalla squadra in casa; (d) i goal fatti dalla squadra in trasferta
- Il sistema convalida i dati forniti per verificare che l'inserimento sia possibile; in questo caso registra i dati nella base di dati; in particolare, oltre ad inserire la partita, aggiorna i punteggi delle squadre coinvolte (3 punti per la vittoria, 1 punto per il pareggio, 0 per la sconfitta)
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati dell'utente, dati della partita inserita, punteggi aggiornati per le due squadre); lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**inserimento.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 per la partita

- Se la convalida fallisce, l'utente ritorna allo schermo di immissione (**inserimento.jsp**) in cui vengono visualizzati i necessari messaggi 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**.
- Per la gestione della base di dati è necessario utilizzare **PostgreSQL**, e il relativo driver **JDBC**