

Programmazione per il Web

Specifiche di progetto

A.A. 2015/2016

Igor Melatti

June 20, 2016

1 Funzionalità Richieste

Si vuole costruire un sito di commercio elettronico, usando le Servlet Java e le pagine JSP. Nel seguito, si supporrà che la Web Application che realizza il sito sia `ecommerce_2016_matricola` (dove al posto di `matricola` occorre sostituire la matricola dello studente), e che quindi il sito venga visualizzato all'indirizzo `http://localhost:8080/ecommerce_2016_matricola`.

1.1 Dati e Schemi

Il contenuto del sito (ovvero, ciò che dovrà essere visualizzato e potenzialmente modificato dal sito) è descritto, in modo equivalente, sia dal file XML `market.xml` (conforme allo schema XSD `market.xsd`) che dal file `market.sql`, che contiene il dump di un database PostgreSQL (il cui DDL è dato nel file `market_ddl.sql`). Una descrizione semplificata è la seguente. Occorre gestire sia utenti che categorie di prodotti. Gli utenti possono essere *amministratori* oppure *normali*. Le categorie sono identificate da un numero, e contengono prodotti messi in vendita. A loro volta, i prodotti sono identificati da un numero, e possono essere o no disponibili per la vendita. Nel seguito si parlerà dei dati memorizzati nel file XML o nel database come *base di dati*.

1.2 Pagine HTML da Costruire

Nel sito devono essere presenti le seguenti (categorie di) pagine.

1. Una pagina che mostri le categorie di prodotti (con tutte le informazioni contenute negli schemi, ma con in più l'indicazione del numero totale dei prodotti in ciascuna categoria¹). Tale pagina costituisce la pagina iniziale dell'intero sito (corrispondente all'indirizzo

¹Attenzione: se per un prodotto il campo `show` è falso, non va incluso nel conteggio

http://localhost:8080/ecommerce_2016_matricola/). Nel seguito, verrà chiamata *pagina delle categorie*.

2. Una pagina che, dato un insieme non vuoto di categorie, mostri i prodotti (con tutte le informazioni contenute negli schemi) in vendita in tali categorie. Nel seguito, verrà chiamata *pagina dei prodotti* e dovrà essere raggiungibile tramite l'indirizzo http://localhost:8080/ecommerce_2016_matricola/products.jsp.
3. Una pagina che mostri il contenuto del carrello, ovvero i prodotti attualmente selezionati per l'acquisto (con la relativa quantità). Deve riportare anche il totale della spesa. Nel seguito, verrà chiamata *pagina del carrello* e dovrà essere raggiungibile tramite l'indirizzo http://localhost:8080/ecommerce_2016_matricola/trolley.jsp.
4. Una pagina che mostri tutti gli utenti registrati, con tutte le informazioni contenute negli schemi. Nel seguito, verrà chiamata *pagina degli utenti* e dovrà essere raggiungibile tramite l'indirizzo http://localhost:8080/ecommerce_2016_matricola/users.jsp.
5. Una pagina per registrarsi al sito (ovvero, diventare un utente). Deve essere possibile inserire tutte le informazioni presenti nello schema (alcune sono opzionali, esattamente come nello schema), con l'eccezione del tipo utente (che sarà sempre **regular**). Nel seguito, verrà chiamata *pagina della registrazione* e dovrà essere raggiungibile tramite l'indirizzo http://localhost:8080/ecommerce_2016_matricola/signup.html.
6. Una pagina per effettuare il login (ovvero, che permette ad un utente registrato di autenticarsi). Nel seguito, verrà chiamata *pagina di login* e dovrà essere raggiungibile tramite l'indirizzo http://localhost:8080/ecommerce_2016_matricola/login.html.
Se si prova a visualizzare questo indirizzo dopo che un login è già stato effettuato, si viene rediretti alla pagina iniziale delle categorie.

1.3 Specifiche per le Pagine

Le pagine elencate in Sezione 1.2 dovranno essere visualizzabili diversamente a seconda che l'utente che le visualizza sia amministratore, normale (**regular**) oppure non loggato. Per ogni caso di errore non esplicitamente trattato nel seguito, occorre rimandare alla pagina delle categorie con il messaggio di errore "Incorrect Request". Nel seguito, tali casi verranno esaminati separatamente.

1.3.1 Utente Non Loggato

Delle pagine elencate in Sezione 1.2, sono visualizzabili solo le seguenti: quella delle categorie, quella dei prodotti, quella del login e quella della registrazione. Per ogni singola pagina valgono le seguenti specifiche.

- La pagina delle categorie deve mostrare tutte le categorie presenti nella base di dati, ordinate per `id` (ovvero: nell'HTML ritornato, la categoria con un identificatore minore deve trovarsi più in alto di quella con un identificatore maggiore). Qui come altrove, nel momento di inserire le immagini (quando sono presenti), occorre usare tutte le informazioni presenti nella base di dati per la categoria (o, più avanti, per il prodotto) in esame, che vanno codificate negli opportuni attributi del tag HTML `img`. Pertanto, gli attributi `src` ed `alt` del tag `image` di `market.xsd` diventano gli omonimi attributi del tag HTML `img`, e analoga sorte deve toccare gli attributi `image_path` ed `image_descr` delle tabelle `item` e `category` della base di dati, rispettivamente. Inoltre, in tale pagina:
 - deve essere presente un link alla pagina di login;
 - ogni immagine e ogni nome di categoria è un link alla pagina dei prodotti di quella categoria;
 - ad ogni categoria deve essere associato un input HTML di scelta multipla, e deve essere presente un bottone HTML che rimandi alla pagina dei prodotti delle categorie scelte.
- La pagina dei prodotti deve mostrare solo i prodotti relativi ad un certo numero di categorie (quelle scelte nella pagina delle categorie), per i quali l'attributo `show` sia vero (nota: se nell'XML non viene specificato, allora è vero). Tali categorie vanno specificate nella *query string* con il parametro `categories`, avente come valore la lista degli identificatori (ordinata per valore degli identificatori stessi), separati da trattini `-`. Ulteriori parametri della query string dovranno essere `page` (numero di pagina) e `numItems` (numero di prodotti per pagina), aventi come valore un intero positivo. Siano p e k tali valori, e sia $\langle P_1, \dots, P_n \rangle$ la sequenza dei prodotti da mostrare, ordinata lessicograficamente prima sugli identificatori delle categorie di appartenenza, e poi su quelli dei prodotti. Si immagini di dividere questa sequenza in gruppi da k prodotti l'uno; allora, dovranno essere mostrati solamente i prodotti del p -esimo gruppo, ovvero $P_{(p-1)k+1}, \dots, P_{pk}$. Se $p > \lceil \frac{n}{k} \rceil$ (maggiore del numero massimo di pagine), occorre scrivere il messaggio d'errore "There is no page p " (al posto di p , mettere il valore) e non mostrare nessun prodotto. Se p non viene dato, assumere che sia 1. Se k non viene dato, assumere che sia il valore del parametro di configurazione della Web application denominato `itemsPerPage`. Se non viene dato il parametro `categories`, e non si è nemmeno arrivati alla pagina dei prodotti scegliendo delle categorie dalla pagina delle categorie, allora occorre tornare alla pagina delle categorie, mostrando all'inizio della pagina stessa il messaggio d'errore "Please select a category in order to show products". Se il parametro `categories` contiene un identificatore di una categoria non esistente, tornare alla pagina delle categorie, mostrando all'inizio della pagina stessa il messaggio d'errore "The category identifier I does not exist. Please select existing categories in order to show products" (dove I è il primo identificatore non

trovato). Inoltre, la visualizzazione deve raggruppare i prodotti in categorie (mostrando, all'inizio di ogni gruppo di prodotti di una categoria, il nome della categoria stessa). Infine, nella pagina dei prodotti:

- deve essere presente un link alla pagina di login;
 - deve essere presente un link alla pagina delle categorie;
 - devono essere presenti $\lceil \frac{n}{k} \rceil$ link alla stessa pagina dei prodotti, con **page** settato di volta in volta a $1, \dots, \lceil \frac{n}{k} \rceil$; inoltre, devono essere presenti i link alla pagina precedente (la precedente della prima è la prima), quella successiva (la successiva dell'ultima è l'ultima), la prima e l'ultima;
 - deve essere presente un bottone HTML associato ad un form che permetta di scegliere il valore di k ; una volta settato, deve rimanere fissato anche nelle successive visite alla pagina dei prodotti fatte dallo stesso browser (finché il browser stesso non viene chiuso);
 - deve essere presente un bottone HTML associato ad un form che permetta di scegliere un ordinamento diverso: per **Id**, **Price**, **Available** o **Title**. Una volta cliccato su tale bottone, si ritorna alla pagina dei prodotti riordinata (ma sempre suddivisa in categorie), con **page** settato ad 1. Inoltre, il valore della chiave dell'ordinamento deve rimanere fissato anche nelle successive visite alla pagina dei prodotti fatte dallo stesso browser (finché il browser stesso non viene chiuso). Infine, nel caso dell'ordinamento per **Price**, fare sempre riferimento al prezzo in Euro (assumere che, se la valuta non è l'Euro, allora c'è la conversione **toEUR**);
- La pagina di login permette ad un utente di fare login: tale azione avrà durata finché il browser non viene chiuso. Nella pagina di login:
 - deve essere presente un link alla pagina di registrazione;
 - deve essere presente un bottone HTML che permetta di autenticare la coppia (nome utente, password) immessa. In caso di autenticazione fallita, si ritorna alla pagina di login con un messaggio d'errore; altrimenti, si continua a partire dalla pagina dalla quale la pagina di login era stata raggiunta.
 - Nella pagina di registrazione:
 - deve essere presente un bottone HTML che permetta di aggiungere l'utente con le informazioni immesse. In caso di registrazione fallita (o perché il nome utente scelto è già presente, o perché non tutte le informazioni obbligatorie sono state fornite), si ritorna alla pagina di registrazione con un messaggio d'errore; altrimenti, si continua a partire dalla pagina dalla quale la pagina di login era stata raggiunta.

1.3.2 Utente Normale (regular)

Delle pagine elencate in Sezione 1.2, sono visualizzabili solo le seguenti: quella delle categorie, quella dei prodotti e quella del carrello. Per ogni singola pagina valgono le seguenti specifiche.

- Per la pagina delle categorie valgono le stesse specifiche della Sezione 1.3.1. Oltre a ciò, dev'essere sempre riportato o il nome e cognome utente reale (se almeno uno dei due è presente nella base di dati) o il nome utente. Inoltre, deve essere mostrato anche il sommario del carrello: numero di prodotti scelti e valore totale della spesa. Infine, la pagina deve contenere gli stessi link della Sezione 1.3.1, ma *senza* quello alla pagina di login e con in più i seguenti link:
 - “Logout”, che fa tornare le richieste dal browser attuale a quelle di utente non loggato;
 - “Cart”, che rimanda alla pagina del carrello per l'utente attuale.
- Per la pagina dei prodotti valgono le stesse specifiche della Sezione 1.3.1. Tuttavia, se un prodotto è già presente nel carrello, non va mostrato nella pagina dei prodotti (a meno che in seguito non venga cancellato dal carrello). Oltre a ciò, dev'essere sempre riportato o il nome e cognome utente reale (se almeno uno dei due è presente nella base di dati) o il nome utente. Inoltre, deve essere mostrato anche il sommario del carrello: numero di prodotti scelti e valore totale della spesa. Il valore della spesa dev'essere tradotto in EUR (assumere che, se la valuta non è l'Euro, allora c'è la conversione toEUR) e riportato con 2 sole cifre decimali (assumere che i prezzi saranno sempre a 2 cifre decimali). Infine, la pagina deve contenere gli stessi link della Sezione 1.3.1, ma *senza* quello alla pagina di login e con in più i seguenti:
 - un link “Logout”, che fa tornare le richieste dal browser attuale a quelle di utente non loggato;
 - un link “Cart”, che rimanda alla pagina del carrello per l'utente attuale;
 - ogni immagine e ogni nome di prodotto è un link alla stessa attuale pagina dei prodotti, dove il carrello virtuale è aggiornato con l'aggiunta del prodotto scelto (con quantità pari ad 1);
 - ad ogni prodotto deve essere associato un input HTML di scelta multipla ed un campo numerico per la quantità (inizialmente 1), e deve essere presente un bottone HTML che rimandi alla stessa attuale pagina dei prodotti, dove il carrello virtuale è aggiornato con l'aggiunta del prodotto scelto con la quantità scelta. Quando un prodotto viene scelto per l'acquisto, non deve più essere mostrato nella pagina dei prodotti (a meno che non venga poi cancellato dal carrello, vedere più avanti). Affinché un prodotto possa essere scelto per l'acquisto,

è necessario che l'attributo `available` sia meno recente della data della richiesta (vale anche per i link sulle immagini e sui nomi dei prodotti); in caso di errore, i soli prodotti non ancora disponibili non vanno aggiunti al carrello.

- La pagina del carrello deve mostrare tutti e soli i prodotti scelti finora dall'utente, visualizzati come nella pagina dei prodotti (paginazione ed ordinamento compresi), ma con la quantità inizialmente fissata a quella scelta al momento dell'aggiunta (e il prezzo modificato di conseguenza). Anche qui, le scelte per paginazione ed ordinamento vanno mantenute finché il browser non viene chiuso. Inoltre, la scelta di paginazione/ordinamento fatta per i prodotti dev'essere inizialmente valida anche per il carrello (e viceversa). Deve essere inoltre riportato il totale della spesa (di nuovo, il totale dev'essere riportato in EUR); nel caso ci sia la paginazione, dev'essere fatto su tutte le pagine. Di nuovo, la somma dev'essere a 2 sole cifre decimali. Se il carrello è vuoto, occorre che sia presente la scritta "The cart is empty". Infine, nella pagina del carrello:
 - ad ogni prodotto deve essere associato un input HTML di scelta multipla ed un campo numerico per la quantità (inizialmente quella scelta in precedenza), e deve essere presente un bottone HTML che rimandi alla stessa attuale pagina del carrello, dove la quantità è stata modificata;
 - deve essere presente un unico bottone HTML "Delete Selected", che rimuove dal carrello solo i prodotti selezionati dalla scelta multipla di cui sopra;
 - ad ogni prodotto deve essere associato un bottone HTML "Delete", che rimuove il prodotto dal carrello;
 - deve essere presente un link "Logout", che fa tornare le richieste dal browser attuale a quelle di utente non loggato, e rimanda direttamente alla pagina iniziale delle categorie;
 - deve essere presente un link "Back To Products", che rimanda alla pagina dei prodotti che era visualizzata nel momento in cui si è selezionato il link "Cart";
 - deve essere presente un link "Back To Categories", che rimanda alla pagina delle categorie;
 - devono essere presenti due bottoni HTML "Buy" e "Delete All", che svuotano il carrello stesso. Fintantoché uno di questi bottoni non viene premuto, il carrello deve mantenere il suo contenuto, anche attraverso cicli di logout e di login.

1.3.3 Amministratore

Delle pagine elencate in Sezione 1.2, sono visualizzabili solo le seguenti: quella delle categorie, quella dei prodotti e quella degli utenti. Per ogni singola pagina valgono le seguenti specifiche.

- Per la pagina delle categorie valgono le stesse specifiche della Sezione 1.3.1, ma tutte le informazioni mostrate (tranne l'identificatore) sono modificabili. Oltre a ciò, dev'essere sempre riportato o il nome e cognome utente reale (se almeno uno dei due è presente nella base di dati) o il nome utente. Inoltre, la pagina deve contenere gli stessi link della Sezione 1.3.1, ma *senza* quello alla pagina di login, e con in più i seguenti:
 - deve essere presente un link “Manage Accounts”, che manda alla pagina degli utenti;
 - deve essere presente un link “Logout”, che fa tornare le richieste dal browser attuale a quelle di utente non loggato;
 - all'inizio della pagina, prima di riportare le varie categorie, deve essere possibile immettere tutte le informazioni per una nuova categoria (compresa la possibilità di uploadare un'immagine). L'identificatore non va immesso, ma va poi settato al massimo identificatore attualmente presente per una categoria, aumentato di 1. Ai campi appena descritti va associato un bottone HTML che modifica la base di dati con l'aggiunta della categoria, e poi torna alla stessa pagina delle categorie;
 - ad ogni categoria deve essere associato un bottone “Modify”, che modifica la categoria nella base di dati, considerando i nuovi valori immessi;
 - ad ogni categoria deve essere associato un bottone “Delete”, che apre una nuova pagina contenente la lista (non paginata, e ordinata sugli identificatori) degli identificatori e nomi dei prodotti in quella categoria. Questa nuova pagina, denominata nel seguito *pagina di cancellazione di una categoria*, deve avere un link che modifica la base di dati con la cancellazione della categoria e di tutti i prodotti in essa presenti, e poi torna alla pagina delle categorie. Inoltre, deve anche contenere un link che torna alla pagina delle categorie, senza effettuare alcuna modifica.
- Per la pagina dei prodotti valgono le stesse specifiche della Sezione 1.3.1, ma tutte le informazioni mostrate (tranne l'identificatore) sono modificabili. Inoltre, vanno mostrati anche i prodotti dove `show` è falso (anche tale valore dev'essere modificabile). Oltre a ciò, dev'essere sempre riportato o il nome e cognome utente reale (se almeno uno dei due è presente nella base di dati) o il nome utente. Inoltre, la pagina deve contenere gli stessi link della Sezione 1.3.1, ma *senza* quello alla pagina di login. Infine, in tale pagina:
 - deve essere presente un link “Manage Accounts”, che manda alla pagina degli utenti;
 - deve essere presente un link “Logout”, che fa tornare le richieste dal browser attuale a quelle di utente non loggato;

- all’inizio della pagina, prima di riportare i vari prodotti nelle categorie scelte, deve essere possibile immettere tutte le informazioni per un nuovo prodotto (compresa la possibilità di uploadare un’immagine). La categoria del nuovo prodotto va scelta da un dropdown. L’identificatore non va immesso, ma va poi settato al massimo identificatore attualmente presente per un prodotto della stessa categoria, aumentato di 1². Ai campi appena descritti va associato un bottone HTML che modifica la base di dati con l’aggiunta del prodotto, e poi torna alla stessa pagina dei prodotti;
 - ad ogni prodotto deve essere associato un bottone “Modify”, che modifica il prodotto nella base di dati, considerando i nuovi valori immessi³
 - ad ogni prodotto deve essere associato un bottone “Delete”, che modifica la base di dati con la cancellazione del prodotto, e poi torna alla stessa pagina dei prodotti.
- La pagina degli utenti deve mostrare tutti gli utenti presenti nella base di dati, ordinati per tipo (prima gli amministratori, poi i **regular**) e poi per **username**. I due gruppi di utenti devono essere mostrati in 2 tabelle diverse, visualizzate consecutivamente. Inoltre, in tale pagina:
 - deve essere presente un link “Back To Categories” alla pagina delle categorie;
 - deve essere presente un link “Logout”, che fa tornare le richieste dal browser attuale a quelle di utente non loggato e torna alla pagina delle caregorie;
 - all’inizio della pagina, prima di riportare i vari utenti, deve essere possibile immettere tutte le informazioni per un nuovo utente (permettendo anche di scegliere se amministratore o no). Ai campi appena descritti va associato un bottone HTML che modifica la base di dati con l’aggiunta del nuovo utente, e poi torna alla stessa pagina dei prodotti;
 - ad ogni utente deve essere associato un bottone “Modify”, che modifica l’utente nella base di dati, considerando i nuovi valori immessi;
 - ad ogni utente (tranne quello attuale) deve essere associato un bottone “Delete”, che modifica la base di dati con la cancellazione dell’utente, e poi torna alla stessa pagina degli utenti.

²Questo vale solo nel caso in cui si legga e scrive il file XML; nel caso del database, è solo il massimo identificatore aumentato di 1

³Una precisazione: il campo “currency” si comporta diversamente nel caso in cui la base di dati sia il file XML e nel caso in cui invece sia il DBMS PostgreSQL. *Non* è richiesto di sanare questa discrepanza. Pertanto, ad esempio, modificare una currency c di un prodotto p , cambiandone una o più conversioni, ha l’effetto di modificare la conversione per tutti gli altri prodotti che hanno c come currency, ma solo nel caso del DBMS PostgreSQL (parametro di configurazione XML a **false**). Come ulteriore esempio, immettere un nuovo prodotto con una currency già usata ma specificando un diverso cambio, ha l’effetto di modificare la conversione per tutti gli altri prodotti che hanno c come currency, ma solo nel caso del DBMS PostgreSQL.

1.4 Nomi di Tag

Table 1: Tabella dei nomi dei tag (da intendersi come attributo “name” del tag stesso).

Nome	Pagina	Commento
<code>insertCategory</code>	Categorie	Form che permette di aggiungere una nuova categoria (solo amministratori)
<code>insertCategoryName</code>	Categorie	Input del form <code>insertCategory</code> per l'immissione del nome di una nuova categoria (solo amministratori)
<code>insertCategoryImage</code>	Categorie	Input del form <code>insertCategory</code> per l'immissione dell'immagine di una nuova categoria (solo amministratori)
<code>selectCategory</code>	Categorie	Form che permette di scegliere una categoria (per gli amministratori, anche di modificarla o cancellarla)
<code>selectCategoryInput</code>	Categorie	Input del form <code>selectCategory</code> , per scegliere di quali categorie mostrare i prodotti; i loro valori devono coincidere con gli id delle rispettive categorie
<code>selectCat</code>	Categorie	Pulsante HTML del form <code>selectCategory</code> , per scegliere di quali categorie mostrare i prodotti
<code>updateCategory<i>i</i>Name</code>	Categorie	Input del form <code>selectCategory</code> per la modifica del nome della categoria con id <i>i</i> (solo amministratori)
<code>updateCategory<i>i</i>Image</code>	Categorie	Input del form <code>selectCategory</code> per la modifica dell'immagine della categoria con id <i>i</i> (solo amministratori)
<code>updateCat<i>i</i></code>	Categorie	Pulsante HTML del form <code>selectCategory</code> , per scegliere di modificare la categoria con id <i>i</i> (solo amministratori)
<code>deleteCat<i>i</i></code>	Categorie	Pulsante HTML del form <code>selectCategory</code> , per scegliere di cancellare la categoria con id <i>i</i> (solo amministratori)
<code>goToCategory<i>i</i></code>	Categorie	Link HTML sull'immagine della categoria con id <i>i</i> , per scegliere di vedere direttamente i prodotti di tale categoria
<code>login</code>	Categorie	Link HTML per effettuare il login
<code>logout</code>	Categorie	Link HTML per effettuare il logout
<code>accounts</code>	Categorie	Link HTML per la pagina degli utenti (solo amministratori)
<code>delete</code>	Cancellazione Categorie	Link per cancellare la categoria (solo amministratori)

Continuazione di Tabella 1.

Nome	Pagina	Commento
abort	Cancellazione Categorie	Link per tornare alla pagina delle categorie, senza cancellare la categoria (solo amministratori)
insertItem	Prodotti	Form che permette di aggiungere un nuovo prodotto (solo amministratori)
insertItemCategoryId	Prodotti	Input del form <code>insertItem</code> per l'immissione dell'id della categoria di un nuovo prodotto (solo amministratori)
insertItemTitle	Prodotti	Input del form <code>insertItem</code> per l'immissione del titolo di un nuovo prodotto (solo amministratori)
insertItemDescription	Prodotti	Input del form <code>insertItem</code> per l'immissione della descrizione di un nuovo prodotto (solo amministratori)
insertItemPrice	Prodotti	Input del form <code>insertItem</code> per l'immissione del prezzo di un nuovo prodotto (solo amministratori)
insertItemCurrency	Prodotti	Input del form <code>insertItem</code> per l'immissione della valuta di un nuovo prodotto (solo amministratori)
insertItemCurrencyToEUR	Prodotti	Input del form <code>insertItem</code> per l'immissione della conversione in EUR del prezzo di un nuovo prodotto (solo amministratori)
insertItemCurrencyToDollar	Prodotti	Input del form <code>insertItem</code> per l'immissione della conversione in Dollari di un nuovo prodotto (solo amministratori)
insertItemAvailable	Prodotti	Input del form <code>insertItem</code> per l'immissione della data di disponibilità di un nuovo prodotto (solo amministratori; il formato è quello dell'ISO 8601: yyyy-mm-ddThh:mm:ss)
insertItemImage	Prodotti	Input del form <code>insertItem</code> per l'immissione dell'immagine di un nuovo prodotto (solo amministratori)
selectItem	Prodotti	Form che permette di scegliere un prodotto per l'acquisto (solo utenti regolari), o di modificarlo/cancellarlo (solo amministratori)
selectItemInput	Prodotti	Input del form <code>selectItem</code> , per scegliere quali prodotti aggiungere al carrello; i loro valori devono consistere nella stringa $i-j$, dove i è l'id della categoria del prodotto e j è l'id del prodotto (solo utenti regolari)

Continuazione di Tabella 1.

Nome	Pagina	Commento
<code>selectItemQInput$i-j$</code>	Prodotti	Input del form <code>selectItem</code> , per scegliere le quantità dei prodotti da aggiungere al carrello; nuovamente, i è l'id della categoria del prodotto e j è l'id del prodotto (solo utenti regolari)
<code>selectIt</code>	Prodotti	Pulsante HTML del form <code>selectItem</code> , per scegliere quali prodotti aggiungere al carrello (solo utenti regolari)
<code>selectItem$i-j$</code>	Prodotti	Link HTML sull'immagine del prodotto con id j nella categoria con id i , per aggiungere direttamente tale prodotto al carrello (con quantità 1)
<code>updateItem$i-j$Title</code>	Prodotti	Input del form <code>selectItem</code> per la modifica del titolo del prodotto con id j nella categoria con id i (solo amministratori)
<code>updateItem$i-j$Description</code>	Prodotti	Input del form <code>selectItem</code> per la modifica della descrizione del prodotto con id j nella categoria con id i (solo amministratori)
<code>updateItem$i-j$Price</code>	Prodotti	Input del form <code>selectItem</code> per la modifica del prezzo del prodotto con id j nella categoria con id i (solo amministratori)
<code>updateItem$i-j$Currency</code>	Prodotti	Input del form <code>selectItem</code> per la modifica della valuta del prodotto con id j nella categoria con id i (solo amministratori)
<code>updateItem$i-j$CurrencyToEUR</code>	Prodotti	Input del form <code>selectItem</code> per la modifica della conversione in EUR del prezzo del prodotto con id j nella categoria con id i (solo amministratori)
<code>updateItem$i-j$CurrencyToDollar</code>	Prodotti	Input del form <code>selectItem</code> per la modifica della conversione in Dollari del prodotto con id j nella categoria con id i (solo amministratori)
<code>updateItem$i-j$Available</code>	Prodotti	Input del form <code>selectItem</code> per la modifica della data di disponibilità del prodotto con id j nella categoria con id i (solo amministratori; il formato è quello dell'ISO 8601: yyyy-mm-ddThh:mm:ss)
<code>updateItem$i-j$Image</code>	Prodotti	Input del form <code>selectItem</code> per la modifica dell'immagine del prodotto con id j nella categoria con id i (solo amministratori)
<code>updateItem$i-j$</code>	Prodotti	Pulsante HTML del form <code>selectItem</code> per la modifica del prodotto con id j nella categoria con id i (solo amministratori)

Continuazione di Tabella 1.

Nome	Pagina	Commento
<code>deleteItem$i-j$</code>	Prodotti	Pulsante HTML del form <code>selectItem</code> per la cancellazione del prodotto con id j nella categoria con id i (solo amministratori)
<code>selectOrder</code>	Prodotti	Form che permette di scegliere un ordinamento per i prodotti
<code>selectOrderInput</code>	Prodotti	Input del form <code>selectOrder</code> ; il suo valore deve essere uno tra <code>id</code> , <code>Price</code> , <code>Available</code> o <code>Title</code>
<code>selectPageSize</code>	Prodotti	Form che permette di scegliere quanti prodotti mostrare per ogni pagina
<code>selectPageSizeInput</code>	Prodotti	Input del form <code>selectPageSize</code>
<code>goToPagei</code>	Prodotti	Link per andare alla pagina i nella paginazione dei prodotti
<code>goToPredPage</code>	Prodotti	Link per andare alla pagina precedente nella paginazione dei prodotti
<code>goToNextPage</code>	Prodotti	Link per andare alla pagina successiva nella paginazione dei prodotti
<code>cart</code>	Prodotti	Link per andare alla pagina del carrello
<code>backToCategories</code>	Prodotti	Link per tornare alla pagina delle categorie
<code>login</code>	Prodotti	Link HTML per effettuare il login
<code>logout</code>	Prodotti	Link HTML per effettuare il logout
<code>accounts</code>	Prodotti	Link HTML per la pagina degli utenti (solo amministratori)
<code>changeItem</code>	Carrello	Form che permette di selezionare un prodotto (o per la modifica della quantità o per la cancellazione)
<code>changeItemInput</code>	Carrello	Input del form <code>changeItem</code> , per scegliere quali prodotti modificare (limitatamente alla quantità) o cancellare; i loro valori devono consistere nella stringa $i-j$, dove i è l'id della categoria del prodotto e j è l'id del prodotto
<code>changeItemQInput$i-j$</code>	Carrello	Input del form <code>changeItem</code> , per scegliere le nuove quantità dei prodotti; nuovamente, i è l'id della categoria del prodotto e j è l'id del prodotto
<code>changeIt</code>	Carrello	Pulsante HTML del form <code>changeItem</code> , per modificare i prodotti selezionati
<code>deleteIt</code>	Carrello	Pulsante HTML del form <code>changeItem</code> , per cancellare dal carrello i prodotti selezionati
<code>buyIt</code>	Carrello	Pulsante HTML del form <code>changeItem</code> , per comprare tutto il contenuto attuale del carrello (in pratica, cancellandone il contenuto)

Continuazione di Tabella 1.

Nome	Pagina	Commento
<code>deleteAll</code>	Carrello	Pulsante HTML del form <code>changeItem</code> , per cancellare tutto il contenuto attuale del carrello
<code>selectOrder</code>	Carrello	Form che permette di scegliere un ordinamento per i prodotti
<code>selectOrderInput</code>	Carrello	Input del form <code>selectOrder</code> ; il suo valore deve essere uno tra <code>id</code> , <code>Price</code> , <code>Available</code> o <code>Title</code>
<code>selectPageSize</code>	Carrello	Form che permette di scegliere quanti prodotti mostrare per ogni pagina
<code>selectPageSizeInput</code>	Carrello	Input del form <code>selectPageSize</code>
<code>goToPage<i>i</i></code>	Carrello	Link per andare alla pagina <i>i</i> nella paginazione dei prodotti
<code>goToPredPage</code>	Carrello	Link per andare alla pagina precedente nella paginazione dei prodotti
<code>goToNextPage</code>	Carrello	Link per andare alla pagina successiva nella paginazione dei prodotti
<code>backToProducts</code>	Carrello	Link per tornare alla pagina dei prodotti (con la paginazione presente al momento di cliccare su "Cart")
<code>backToCategories</code>	Carrello	Link per tornare alla pagina delle categorie
<code>login</code>	Login	Form che permette di effettuare il login
<code>username</code>	Login	Input del form <code>login</code>
<code>password</code>	Login	Input del form <code>login</code>
<code>signup</code>	Login	Link alla pagina di registrazione
<code>signup</code>	Signup	Form che permette di effettuare la registrazione
<code>username</code>	Signup	Input del form <code>signup</code>
<code>password</code>	Signup	Input del form <code>signup</code>
<code>realName</code>	Signup	Input del form <code>signup</code>
<code>realFamilyName</code>	Signup	Input del form <code>signup</code>
<code>insertAccount</code>	Utenti	Form che permette di aggiungere un nuovo utente
<code>insertAccountName</code>	Utenti	Input del form <code>insertAccount</code> per l'immissione dello username di un nuovo utente
<code>insertAccountPassword</code>	Utenti	Input del form <code>insertAccount</code> per l'immissione della password di un nuovo utente
<code>insertAccountRealName</code>	Utenti	Input del form <code>insertAccount</code> per l'immissione del nome di un nuovo utente

Continuazione di Tabella 1.

Nome	Pagina	Commento
insertAccountFamilyName	Utenti	Input del form <code>insertAccount</code> per l'immissione del cognome di un nuovo utente
insertAccountType	Utenti	Input del form <code>insertAccount</code> per l'immissione del tipo di un nuovo utente; potrà assumere come valori <code>regular</code> oppure <code>administrator</code>
updateAccount <i>x</i>	Utenti	Form che permette di modificare/cancellare l'account con username <i>x</i> ⁴
updateAccount <i>x</i> Name	Utenti	Input del form <code>updateAccount<i>x</i></code> per la modifica dello username dell'utente con username <i>x</i>
updateAccount <i>x</i> Password	Utenti	Input del form <code>updateAccount<i>x</i></code> per la modifica della password dell'utente con username <i>x</i>
updateAccount <i>x</i> RealName	Utenti	Input del form <code>updateAccount<i>x</i></code> per la modifica del nome dell'utente con username <i>x</i>
updateAccount <i>x</i> FamilyName	Utenti	Input del form <code>updateAccount<i>x</i></code> per la modifica del cognome dell'utente con username <i>x</i>
updateAcc <i>x</i>	Utenti	Pulsante HTML del form <code>updateAccount<i>x</i></code> , per effettuare le modifiche
deleteAcc <i>x</i>	Utenti	Pulsante HTML del form <code>updateAccount<i>x</i></code> , per cancellare l'utente con username <i>x</i>
backToCategories	Utenti	Link alla pagina delle categorie
logout	Utenti	Link HTML per effettuare il logout (torna alla pagina delle categorie)

Occorre che gli attributi “name” di alcuni tag siano fissati come descritto in Tabella 1.

1.5 Modalità di Funzionamento

Quanto detto in Sezione 1.3 va realizzato nel modo seguente:

- se si parte con l'indirizzo `http://localhost:8080/ecommerce_2016_matricola`, il tutto deve poter funzionare anche se i cookie sono disabilitati
- se si parte dall'indirizzo `http://localhost:8080/ecommerce_2016_matricola/cookies`, il tutto deve funzionare solo se i cookie sono abilitati⁵;

⁴Assumere che non esistano due utenti con uguale username e tipo diverso.

⁵Ovvero, in caso di cookie disabilitati, è corretto che un utente loggato non sia più considerato tale non appena si muove dalla pagina corrente, o che non si riesca a tornare alla pagina precedente dopo un login.

Table 2: Parametri di configurazione

Nome	Visibilità	Significato
XML	Locale alla servlet	Se vale <code>true</code> , la servlet o la pagina JSP deve leggere (e potenzialmente modificare) il file XML, altrimenti deve leggere e modificare il database
DBName	Intera Web Application	Contiene il nome del database
DBUser	Intera Web Application	Username per l'autenticazione al database
DBPassword	Intera Web Application	Password per l'autenticazione al database
XMLName	Intera Web Application	Percorso relativo alla Web Application del file XML (si assume che sia accessibile in lettura e scrittura da servlet e pagine JSP della Web Application). Deve cominciare con uno slash, corrispondente alla directory della Web Application ⁶ . Attenzione: il separatore di directory deve essere <code>/</code> ; nel codice, sostituirlo con <code>File.separator</code>
ImagesDir	Intera Web Application	Percorso relativo al server delle immagini da usare nomi di file di immagini nella base di dati si dovranno trovare dentro tale directory). Deve cominciare con uno slash, corrispondente alla directory di tutte le Web Application. Attenzione: il separatore di directory deve essere <code>/</code> ; nel codice, sostituirlo con <code>File.separator</code>
itemsPerPage	Intera Web Application	Valore di default per la paginazione dei risultati nella pagina dei prodotti

- devono essere gestite opportunamente i parametri di configurazione listati in Tabella 2;
- viene lasciata libera scelta su cosa implementare con servlet e cosa implementare con pagine JSP, con l'unica imposizione che le pagine delle categorie vanno realizzate con servlet. Dove possibile, le pagine di login e registrazione devono essere realizzate con HTML statico;
- è possibile usare il database anche nel caso in cui l'input sia il file XML, ma senza far riferimento alle tabelle del file `market.ddl.sql`;
- per gli utenti che cercano di visualizzare pagine per le quali non si ha l'autorizzazione (ad esempio, la pagina del carrello per utenti non loggati), occorre reindirizzare alla pagina iniziale, con il messaggio d'errore: "Incorrect request".

⁶Usare `getServletContext().getRealPath` per ottenere il path del file su disco.

1.6 Tag Personalizzati

Scrivere il codice ed il deployment descriptor per un tag `item`, con le seguenti specifiche. Il tag dovrà avere i seguenti attributi:

- `XML`: booleano, se settato a `true` legge le informazioni dal file XML, altrimenti da PostgreSQL (opzionale, di default è vero)
- `XMLName`: come in Tabella 2 (opzionale, ma obbligatorio se `XML` è vero)
- `DBName`: come in Tabella 2 (opzionale, ma obbligatorio se `XML` è falso)
- `DBUser`: come in Tabella 2 (opzionale, ma obbligatorio se `XML` è falso)
- `DBPassword`: come in Tabella 2 (opzionale, ma obbligatorio se `XML` è falso)
- `idCat`: intero, è l'identificatore di una categoria (obbligatorio)
- `idProd`: intero, è l'identificatore di un prodotto all'interno della categoria (obbligatorio)

Se il tag ha un corpo, al suo interno deve essere possibile accedere al prodotto di identificatore `idProd` della categoria `idCat`, sotto forma di un Java Bean di tipo `ItemBean` e di identificatore `idItem` (se non esiste un tale prodotto, allora il Java Bean dovrà essere `null`). Tale Java Bean dovrà essere implementato nel file `WEB-INF/classes/bean/ItemBean.class` e avere le seguenti proprietà con i seguenti tipi:

- `idProd`: intero; dev'essere inizializzato a zero dal costruttore di default
- `idCat`: intero; dev'essere inizializzato a zero dal costruttore di default
- `title`: stringa
- `description`: stringa
- `price`: double
- `currencyName`: stringa
- `currencyToEUR`: double
- `currencyToDollar`: double
- `available`: `java.util.Date`
- `imageSrc`: stringa
- `imageAlt`: stringa
- `show`: booleano

Tale bean dovrà avere un carattere persistente, e dovrà essere accessibile a qualsiasi utente acceda alla pagina. Questo comportamento dovrà essere abilitato solo se i vincoli di obbligarietà "condizionale" sugli attributi `XMLName`, `DBName`, `DBUser` e `DBPassword` sono rispettati, altrimenti il Java Bean dovrà essere `null`.

2 Modalità di Consegna

Verrà approntato un sito Internet dove uploadare un singolo file ZIP. Tale file dovrà contenere l'intero context root della Web application. Oltre a contenere le directory (ed eventualmente le librerie) richieste per il normale funzionamento, occorre mettere in una directory `WEB-INF/source` tutti i sorgenti delle servlet, che dovranno essere compilabili con Java 1.6. Nella stessa directory, deve essere presente anche un `Makefile` per compilare tutti i sorgenti. Lo stesso `Makefile` deve avere almeno i seguenti target:

- `all`: compila tutti i sorgenti Java
- `clean`: cancella tutti i `.class`
- `copy`: copia i file `.class` ottenuti con la compilazione nelle giuste directory all'interno di `WEB-INF`.

Inoltre, la compilazione non deve usare direttamente il comando `javac`, ma deve usare una variabile `JC`, definita ad esempio come `JC = javac`. Per un esempio di come scrivere un `Makefile` per Java, vedere l'indirizzo <https://www.cs.swarthmore.edu/~newhall/unixhelp/javamakefiles.html>. Tutti i file di HTML statico vanno messi in una directory `html` del context root. Tutte le pagine JSP vanno messe in una directory `jsp` del context root. Il deployment descriptor del tag personalizzato si deve chiamare `tld.xml` e va messo nella stessa directory del `web.xml`. Non è richiesto che lo ZIP contenga le immagini.

Infine, *non* usare il tag `url-pattern` vuoto (ovvero, `<url-pattern></url-pattern>` o `<url-pattern />`) nel `web.xml`.

3 Valutazione del Progetto

Il progetto viene valutato sommando i punteggi ottenuti come segue:

- se passano tutti i test sulla parte che lavora sull'XML e ha i cookie disabilitati: 2/12
- se passano tutti i test sulla parte che lavora sull'XML e ha i cookie abilitati: 1/12
- se passano tutti i test sulla parte che lavora sul database e ha i cookie disabilitati: 3/12
- se passano tutti i test sulla parte che lavora sul database e ha i cookie abilitati: 2/12
- se passano tutti i test sulla parte che definisce il tag personalizzato, con XML a true: 2/12
- se passano tutti i test sulla parte che definisce il tag personalizzato, con XML a false: 2/12