

Progetto di Laboratorio di Basi di Dati 2

A.A. 2010/2011

Andrea Mantoni

Table of Contents

Documento di progetto.....	1
Analisi dei requisiti.....	1
Descrizione sintetica.....	1
Profili utenti ed operazioni principali.....	2
Progettazione concettuale.....	3
Schema E-R.....	3
Documentazione di supporto allo schema E-R.....	3
Progettazione logica.....	4
Relazioni.....	4
Progettazione fisica.....	4
Codice SQL per la creazione delle tabelle.....	4
Codice SQL di alcune operazioni.....	5
Running prototype.....	6
Implementazione finale.....	6
Tecnologie utilizzate.....	6
Ambiente di sviluppo utilizzato.....	7
Descrizione sintetica dei sorgenti.....	7
Guida per l'utente finale.....	9
Installazione/deployment.....	9
Requisiti.....	9
Procedura d'installazione.....	9
Sicurezza.....	10
Personalizzazione GUI.....	10
Funzioni di amministrazione.....	11
Aggiunta di un nuovo prodotto.....	11
Aggiunta caratteristiche.....	11
Aggiunta di un nuovo fornitore o classe merceologica.....	12
Registrazione di una fornitura.....	12
Funzioni per i clienti.....	12
Ricerca nel catalogo prodotti.....	12
Invio di un ordine.....	13
Bugs e limitazioni note.....	13

Documento di progetto

Analisi dei requisiti

Descrizione sintetica

Si vuole realizzare un sito di e-commerce per la vendita di **prodotti** su internet.

L'**amministrazione** deve avere la possibilità di tenere traccia per ciascun prodotto della **disponibilità** in magazzino e del suo **fornitore** attuale.

I **clienti** devono avere la possibilità di consultare il catalogo dei prodotti divisi per **classe merceologica** ed effettuare ricerche.

Dopo la registrazione i clienti sono anche abilitati a inserire i prodotti nel **carrello** ed inviare **ordini**.

Profili utenti ed operazioni principali

Amministratori

caratteristiche: buone conoscenze informatiche. Buone capacità di interpretazione degli errori software. → le loro interfacce di interazione non necessitano di particolari accorgimenti.

Gli amministratori sono abilitati a:

- aggiungere nuovi prodotti nel catalogo (poche volte al mese);
- aggiornare le schede dei prodotti esistenti (poche volte al mese);
- aggiungere/modificare le classi merceologiche e le schede dei fornitori (poche volte al mese);
- monitorare i prodotti in esaurimento (una volta al giorno);
- monitorare gli ordini inviati dai clienti ed aggiornare il loro stato (ad es. “in elaborazione”, “spedito”, etc.) (una volta al giorno).

Clienti

caratteristiche: ampia classe di utenti con conoscenze informatiche eterogenee. Minore capacità di interpretazione degli errori software. → è richiesta una maggiore attenzione nella progettazione delle UI con le quali essi interagiranno (necessità di suggerimenti, visualizzazione di messaggi d'errore semplificati, etc.).

Un cliente può trovarsi in due stati: **anonimo/non loggato** e **loggato** (ognuno con il proprio account personale).

I clienti non loggati possono:

- consultare il catalogo dei prodotti divisi per classe merceologica (molte volte al giorno);
- effettuare ricerche nel catalogo dei prodotti (molte volte al giorno);
- visualizzare dettagli dei singoli prodotti (descrizione, immagini, links, etc.) (molte volte al giorno);
- registrarsi al sito ed immagazzinare i propri dati personali (una sola volta per ogni utente);
- inserire e rimuovere i prodotti nel carrello (molte volte al giorno).

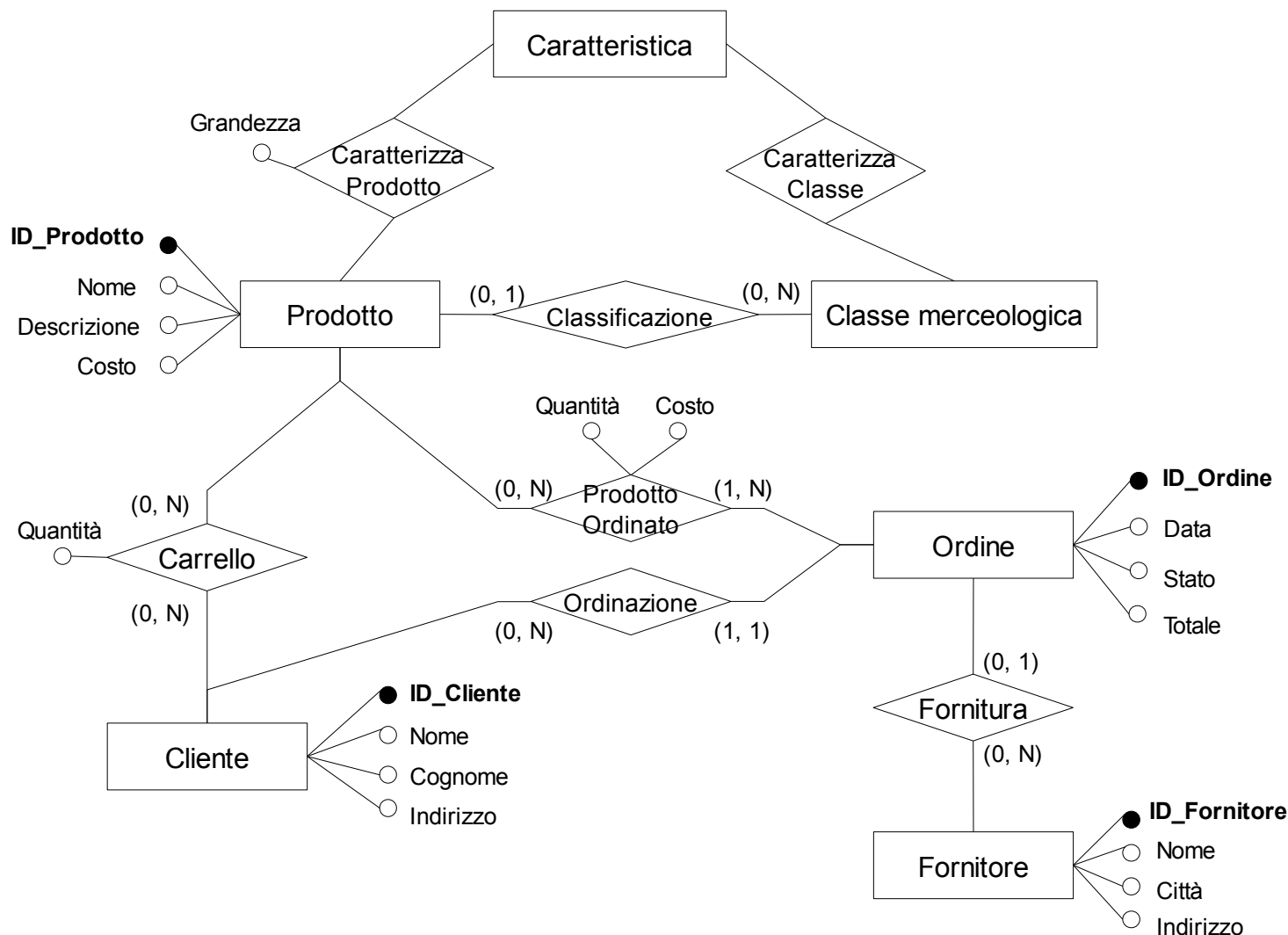
Dopo i login i clienti possono anche:

- modificare il proprio profilo personale (raramente);
- fare logout e salvare i prodotti nel carrello (poche volte al giorno);
- visualizzare il riepilogo e lo stato degli ordini precedenti (una volta al giorno).

OSS.: le operazioni dei clienti hanno una frequenza e dei volumi molto maggiori di quelle degli amministratori. Richiederanno quindi una ottimizzazione maggiore.

Progettazione concettuale

Schema E-R



Documentazione di supporto allo schema E-R

Vincoli non esprimibili nello schema E-R:

- username e password di ciascun utente non possono coincidere;
- username e password di ciascun utente devono avere lunghezza minima di 5 char.

Osservazioni:

- sebbene ciascun prodotto ha un suo costo già memorizzato nella tabella prodotto, è necessario fare una copia di questo valore quando si invia un ordine poiché in seguito il suo costo potrebbe variare. Così si garantisce che il prezzo rimanga “bloccato” dopo l'ordine;
- Il campo “Totale” in “Ordine” è ridondante poiché può essere ottenuto sommando il costo dei prodotti associati;
- L'utente “admin” è una istanza speciale di “Cliente”. Gli “ordini” che esso invia sono le richieste di forniture.

Progettazione logica

Relazioni

```
Cliente(ID_Cliente, Nome, Cognome, Indirizzo, email, username,
        password)
Fornitore(ID_Fornitore, Nome, Indirizzo, Città, Ragione
          Sociale, Immagine, Video)
Classe(ID_Classe, Nome, Descrizione, Immagine, Video)
Prodotto(ID_Prodotto, Nome, Descrizione, Costo, Immagine,
         Video, ID_Fornitore, ID_Classe)

Carrello(ID_Prodotto, ID_Cliente, Quantita)
Ordine(ID_Ordine, ID_Cliente, Stato, Data, Totale)
Prodotto_Ordinato(ID_Ordine, ID_Prodotto, Quantita, Costo)
Fornitura(ID_Ordine, ID_Prodotto, Quantita, Costo)
Carrello(ID_Cliente, ID_Prodotto, Quantita)
Caratteristica(ID_Caratteristica, Nome, Descrizione, Immagine)
CaratterizzaProdotto(ID_Caratteristica, ID_Prodotto,
                    Grandezza)
CaratterizzaClasse(ID_Caratteristica, ID_Classe)
```

Progettazione fisica

Codice SQL per la creazione delle tabelle

```
create table Prodotto
(
    ID_Prodotto      SERIAL          PRIMARY KEY,
    Nome             VARCHAR(255)    NOT NULL,
    Descrizione       VARCHAR(10000),
    Costo            NUMERIC(8,2)    NOT NULL,
    Disponibile      INT             NOT NULL,
    Immagine         VARCHAR(255),
    Video            VARCHAR(255),
    URL              VARCHAR(255),
    ID_Fornitore     BIGINT UNSIGNED REFERENCES
                        Fornitore(ID_Fornitore),
    ID_Classe        BIGINT UNSIGNED REFERENCES
                        Classe_Merceologica(ID_Classe),
    FULLTEXT(Nome, Descrizione)
);
```

OSS.: per velocizzare l'operazione di ricerca nel catalogo dei prodotti (una delle più frequenti), si è creato un indice fulltext sui campi "Nome" e "Descrizione". Questa è una funzionalità proprietaria di MySQL e potrebbe non essere portabile su altri DBMS.

```
create table Cliente
(
    ID_Cliente      SERIAL          PRIMARY KEY,
    Nome            VARCHAR(255)    NOT NULL,
    Cognome         VARCHAR(255)    NOT NULL,
    Indirizzo       VARCHAR(255)    NOT NULL,
```

```

        email          VARCHAR(255)          NOT NULL UNIQUE,
        username       VARCHAR(255)          NOT NULL UNIQUE,
        password        VARCHAR(255)          NOT NULL,
        CHECK ((LENGTH(password) >= 5) AND
               (username <> password))
    );

```

OSS.: La clause “check” è attualmente ignorata da MySQL.

```

create table Fornitore
(
    ID_Fornitore      SERIAL                PRIMARY KEY,
    Nome              VARCHAR(255)          NOT NULL,
    Indirizzo         VARCHAR(255)          NOT NULL,
    Citta             VARCHAR(255)          NOT NULL,
    Ragione_Sociale   VARCHAR(255)
);

create table Classe
(
    ID_Classe         SERIAL                PRIMARY KEY,
    Nome              VARCHAR(255)          NOT NULL UNIQUE,
    Descrizione       VARCHAR(1000),
    Immagine          VARCHAR(255)
);

create table Ordine
(
    ID_Ordine         SERIAL                PRIMARY KEY,
    ID_Cliente        BIGINT UNSIGNED       NOT NULL REFERENCES
                                                Cliente(ID_Cliente),
    Stato             VARCHAR(255)          NOT NULL,
    Data              DATE                  NOT NULL,
    Totale            NUMERIC(8,2)          NOT NULL
);

create table Prodotto_Ordinato
(
    ID_Ordine         BIGINT UNSIGNED NOT NULL REFERENCES Ordine(ID_Ordine),
    ID_Prodotto       BIGINT UNSIGNED NOT NULL REFERENCES
                                                Prodotto(ID_Prodotto),
    Quantita          INT                  NOT NULL,
    Costo             NUMERIC(8,2)         NOT NULL,
    PRIMARY KEY(ID_Ordine, ID_Prodotto)
);

```

Le altre tabelle si possono facilmente dedurre dallo schema E-R e sono quindi omesse in questo documento per brevità.

Codice SQL di alcune operazioni

Elenca tutti i prodotti di una classe merceologica:

```

SELECT ID_Prodotto, Nome, Costo, Disponibile
FROM Prodotto
WHERE ID_Classe = ${param.class};

```

Trova tutti gli ordini effettuati da un utente:

```

SELECT *
FROM Ordine

```

```
WHERE ID_Cliente = ( SELECT ID_Cliente
                     FROM Cliente
                     WHERE username = '${sessionScope.username}'
                   );
```

Lettura del contenuto del carrello di un utente:

```
SELECT ID_Prodotto, Quantita
FROM Carrello
WHERE ID_Cliente = (
    SELECT ID_Cliente
    FROM Cliente
    WHERE username = '${sessionScope.username}'
  );
```

Lettura dei dati dei prodotti inclusi in un ordine:

```
SELECT O.ID_Prodotto, Nome, O.Costo, Quantita
FROM Prodotto_Ordinato AS O, Prodotto AS P
WHERE (ID_Ordine = ${param.id}) AND (O.ID_Prodotto = P.ID_Prodotto);
```

Ricerca di un prodotto (alcune parti sono opzionali):

```
SELECT ID_Prodotto, Nome, Costo, Disponibile
FROM Prodotto
WHERE
  ( MATCH (Nome, Descrizione) AGAINST ('${param.keywords}' IN BOOLEAN MODE )
  AND
  ( ID_Classe = ${param.cat} )
  AND
  ( ID_Prodotto = ANY (
      SELECT ID_Prodotto
      FROM CaratterizzaProdotto
      WHERE ( ID_Caratteristica = ${feature1} ) OR
            ( ID_Caratteristica = ${feature2} ) OR
            ...
    )
  )
ORDER BY Disponibile DESC;
```

OSS.: `match()` è una funzione proprietaria di MySQL. Una sintassi alternativa, sicuramente portabile ma più limitata, è:

```
WHERE Nome LIKE '%${param.keywords}%'
```

Running prototype

La directory “asp_proto” contiene un prototipo eseguibile che mostra alcune funzionalità peculiari del sito.

Il prototipo è realizzato in **ASP Classic**, e richiede un server HTTP che supporta tale tecnologia.

Il database utilizzato è di tipo MDB “embedded”, quindi non richiede un DBMS esterno.

Implementazione finale

Tecnologie utilizzate

La scelta è ricaduta sulla tecnologia **Java Server Pages (JSP)** per il maggiore livello di sicurezza

da essa offerto. La sicurezza infatti rappresenta un aspetto cruciale per i siti di e-commerce. I motivi sono numerosi: necessità di mantenere riservati i dati dei clienti, impedire l'accesso alle funzioni di amministrazione ai non addetti, etc.

Per migliorare la leggibilità del codice si è fatto largo uso della **JSP Standard Tag Library (JSTL)** e del **JSP Expression Language (EL)**. Quest'ultimo richiede un servlet engine compatibile con JSP 2.0.

MySQL è una scelta comune come database di siti internet per le sue buone prestazioni.

Infine si è usato **JavaScript** per il controllo dei form usati dai clienti.

Ambiente di sviluppo utilizzato

- Windows XP
- XAMPP v1.7.3 (Apache v2.2.14, MySQL v5.1.41, Apache Tomcat v6.0.20)
- IBM Eclipse

Descrizione sintetica dei sorgenti

- **[admin]** - directory contenente le pagine accessibili solo dall'amministratore. Contiene al suo interno:
 - **install.jsp** - script di inizializzazione del database, necessario per l'installazione del sito. Contiene il codice SQL per la creazione di tutte le tabelle;
 - **product_edit_form.jsp** – form per la modifica e l'aggiunta dei prodotti;
 - **product_exaus.jsp** - utility per controllare i prodotti in esaurimento e quelli terminati;
 - **stats.jsp** – mostra le statistiche del negozio: bilanci, prodotti più venduti, etc.;
 - **table_del_row.jsp, table_edit_row.jsp, table_edit_row_form.jsp, table_list.jsp, table_view_row.jsp**

Queste pagine consentono di effettuare operazioni su tabelle SQL generiche.

Il nome della tabella è passato mediante il parametro “**table**”.

La riga su cui operare è specificata mediante il parametro “**key**”.

Affinché il codice funzioni l'attributo chiave della tabella deve essere nominato come

“ID_” + \${param.table}

Così, ad esempio:

table_view_row.jsp?table=Cliente&key=3 → visualizza il cliente con ID “3”

table_list.jsp?table=Fornitore → visualizza tutti i fornitori

table_edit_row_form.jsp?table=Fornitore → registra un nuovo fornitore

table_edit_row_form.jsp?table=Fornitore&key=1 → modifica il fornitore con ID “1”

table_del_row.jsp?table=Prodotto&key=2 → cancella il prodotto con ID “2”

Il vantaggio di questa soluzione rispetto a delle pagine specifiche per ciascuna tabella è una maggiore indipendenza dallo schema logico. Infatti se si volessero aggiungere degli attributi nella tabelle non è necessario modificare il codice delle pagine.

- **[images]** - Directory contenente le immagini dei prodotti in vendita sul sito (uploadate dall'amministratore).
- **[theme]** – Directory contenente il tema grafico usato dal sito (personalizzabile). Al suo interno abbiamo:
 - **style.css** – foglio di stile predefinito usato da tutte le pagine del sito;
 - **logo.png** – il logo del negozio, mostrato in tutte le pagine;
 - ***.png** - altre immagini associate al tema grafico.
- **[WEB-INF]** – contiene il deployment descriptor e le librerie Java di supporto;
- **cart.jsp** – visualizza e modifica il contenuto del carrello. Accetta due parametri opzionali:
 - **addpid** → aggiunge un prodotto nel carrello, la quantità è specificata con il parametro “q”
 - **delpid** → rimuove un prodotto dal carrello

I prodotti nel carrello sono memorizzati mediante variabili di sessione così nominate: “cart_pid”+ID_Prodotto. Il valore della variabile corrisponde alla quantità.

- **error.jsp** – pagina di errore predefinita del sito. Cattura le eccezioni sollevate nelle pagine e ne mostra la causa. Può anche visualizzare un messaggio d'errore personalizzato ricevuto mediante il parametro “**errorMessage**”
- **message.jsp** – pagina generica per la visualizzazione di un messaggio (ad es. l'esito positivo di un operazione) inviato con il parametro “**message**”
- **footer.inc.jsp** – footer presente in ogni pagina del sito. Contiene alcuni link a pagine informative a cura dell'amministrazione:
 - **conditions.jsp** – condizioni di vendita
 - **faqs.jsp**
- **header.inc.jsp** – header presente in ogni pagina del sito. Contiene i pulsanti per accedere a tutte le funzioni principali.
- **index.jsp** – pagina iniziale del sito. Contiene le ultime comunicazioni dall'amministrazione (importate dal file “**news.inc.html**”) e la lista degli ultimi prodotti aggiunti.
- **order_save.jsp** – crea un nuovo ordine con il contenuto del carrello attuale.
- **order_view.jsp** – visualizza il riepilogo di un ordine, specificato mediante il parametro “id”.
- **orders_list.jsp** – mostra la lista di tutti gli ordini inviati dall'utente corrente. Se l'utente è admin, mostra tutti gli ordini di tutti i clienti e ne consente la modifica.
- **product_cat_list.inc.jsp** – genera la sidebar presente in tutte le pagine del sito

che mostra le classi merceologiche.

- **product_list.inc.jsp** – scriptlet usato di più pagine per mostrare un elenco di prodotti ottenuto come risultato di una query.
- **product_search.jsp** – effettua la ricerca dei prodotti in base ai parametri: **keywords**, **cat**, **features**.
- **product_view.jsp** – visualizza tutti i dati di un singolo prodotto, specificato mediante il parametro “**pid**”.
- **product_compare.jsp** – effettua il confronto di due o più prodotti, specificati mediante il parametro multiplo “**pid**”.
- **user_edit.jsp** – form per la registrazione di un nuovo utente (se l'utente non è loggato) e per la modifica del profilo. Invoca **user_save.jsp** per il salvataggio.
- **user_login.jsp** – validatore del login. Se ha successo ripristina anche il contenuto del carrello.
- **user_logout.jsp** – termina la sessione corrente e salva il contenuto del carrello.
- **user_save.jsp** – convalida la registrazione o la modifica di un profilo utente.

Guida per l'utente finale

Installazione/deployment

Requisiti

- server HTTP (consigliato: Apache)
- servlet engine con supporto a JSP ≥ 2.0 (consigliato: Tomcat)
- DBMS (consigliato: MySQL)

Procedura d'installazione

1. estrarre il contenuto di “jsp_final” nella context directory destinata all'applicazione (ad esempio: “C:\tomcat\webapps\Myshop”).
2. impostare alcuni parametri nel deployment descriptor (WEB-INF\web.xml):
 - `shopName` = il nome del negozio, viene mostrato come titolo in ogni pagina del sito;
 - `databaseDriver` = il driver del database. Se si usa MySQL può essere lasciato inalterato;
 - `databaseUrl` = l'URL del database. Se è in esecuzione sulla stessa macchina insieme a Tomcat può essere lasciato inalterato;
 - `databaseUsername` = username per la connessione al database;
 - `databasePassword` = password per la connessione al database.
3. aprire la seguente pagina con il browser HTTP per procedere all'inizializzazione del database:
<http://localhost:8080/Myshop/admin/install.jsp>

Al termine dell'installazione verrà creato l'utente “admin” con la password predefinita “admin”.

Sicurezza

La tecnologia Java garantisce un buon livello di sicurezza. Tuttavia alcuni accorgimenti possono aumentare ulteriormente la sicurezza del sito:

- scegliere una buona password di admin e cambiarla regolarmente;
- scegliere una buona password per il DBMS e cambiarla regolarmente;
- impedire l'accesso al DBMS da altri host se non è necessario;
- proteggere l'accesso alla directory “admin” con un meccanismo proprietario del server HTTP (ad es. con un file “.htaccess”).

Per quanto riguarda gli account dei clienti, è richiesta una password di lunghezza minima di 5 caratteri. Dopo 5 login falliti il server impedisce ulteriori tentativi fino allo scadere della sessione corrente (con Tomcat questo avviene dopo 30 minuti dall'ultimo accesso del client). La durata delle sessioni può essere impostata nel deployment descriptor.

Personalizzazione GUI

Molti aspetti della GUI possono essere personalizzati modificando il foglio di stile predefinito in “theme/style.css”.

Al suo interno sono definiti le seguenti classi:

```
.header  
.body  
.cat-menu  
.footer
```

Queste classi corrispondono alle 4 “aree” principali che compongono ogni pagina del sito:



La classe “button” invece definisce lo stile dei pulsanti nell'header.

Per cambiare il logo del negozio e le icone è sufficiente sostituire i files immagine nella directory “theme”.

Funzioni di amministrazione

Le principali funzioni di amministrazione sono accessibili dopo il login dell'utente "admin" mediante i pulsanti nell'header:



NOTA: le funzioni di amministrazione sono evidenziate con dei pulsanti di un colore diverso (nello stile predefinito è "LightSalmon").

Aggiunta di un nuovo prodotto

Loggarsi con l'utente "admin", fare click sul pulsante "Aggiungi nuovo prodotto" e compilare il form.

Il fornitore e la classe merceologica del prodotto devono già essere stati registrati precedentemente, o non saranno selezionabili.

I campi "Immagine", "Video" e "Link esterno" consentono l'inserimento di URLs che possono essere:

- assoluti (ad es. "<http://dominio.com/path/immagine.jpg>")
- relativi alla pagina (ad es. "path/immagine.jpg")
- relativi alla root del server (ad es. "/path/immagine.jpg")

Nel campo "video" è possibile inserire un link a Youtube. Un javascript provvederà ad estrarre una anteprima (statica) del video.

Facendo click su "Salva" il prodotto viene immediatamente messo in vendita ed è visibile da tutti i clienti.

Aggiunta caratteristiche

Per creare o modificare una caratteristica tecnica di un prodotto fare click sul pulsante "Caratteristiche prodotti".

Ogni caratteristica può avere associata una "grandezza" (ad es. una misura come "12cm, 100g, 10GB", etc.)

Una volta creata, una caratteristica può essere associata sia ai prodotti sia alle classi merceologiche.

Una caratteristica associata sia ad un prodotto sia alla classe merceologica a cui esso appartiene

viene visualizzata nel form di ricerca.

Viceversa una caratteristica associata solo al prodotto viene visualizzata nella sua scheda, ma non nel form di ricerca.

Aggiunta di un nuovo fornitore o classe merceologica

Loggarsi con l'utente “admin”, fare click sul pulsante “Dati fornitori” (o “Classe merceologiche”).

Apparirà la tabella con tutti i fornitori (classe merceologiche) già registrati.

Fare click sul pulsante “Aggiungi nuovo” in basso nella pagina.

Compilare il form e procedere al salvataggio.

Se il salvataggio non è andato a buon fine sarà visualizzato un messaggio diagnostico di errore.

NOTA: per queste tabelle non viene fornita la validazione dei campi, che è lasciata al DBMS in base ai vincoli definiti sugli attributi delle tabelle.

Registrazione di una fornitura

Aprire la scheda del prodotto di interesse,

inserire il costo e la quantità della fornitura nel form sotto la voce “Fornitore”,

quindi fare click sul pulsante “Richiedi fornitura”.

Un messaggio comunicherà l'esito dell'operazione.

Nelle sessioni successive sarà possibile aggiornare lo stato delle forniture mediante il pulsante “Lista ordini” nell'header.

Funzioni per i clienti

Ricerca nel catalogo prodotti

E' possibile effettuare ricerche nel catalogo prodotti in diversi modi:

- selezionando una classe merceologica nella lista a lato;
- effettuando una ricerca generale per parole chiave (sempre nella sidebar). E' necessario inserire una o più parole chiave di lunghezza minima di 4 caratteri.

In entrambi i casi i risultati appariranno nell'area centrale, e sarà possibile affinare la ricerca specificando le caratteristiche di interesse e/o altre parole chiave:

hard disk

televisori

videogiochi

Ricerca generale

Categoria: televisori

parole chiave:


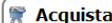

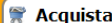

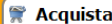
caratteristiche:

☐ diagonale (pollici) : 15
☐ diagonale (pollici) : 20
☐ video input : DVI
☐ video input : HDMI
☐ video input : VGA
☒ rapporto di forma : 4:3

Affina ricerca

Risultati ricerca

Prodotti trovati: 3

	Prodotto	Costo		
	Samsung TV LCD 4/3 20" LW20M21C	601.00€		<input type="checkbox"/>
	Philips LCD 4/3 15" 15PF4121	430.00€		<input checked="" type="checkbox"/>
	LG TV LCD 4/3 20" RZ20LA90	601.00€		<input checked="" type="checkbox"/>

Confronta

condizioni di vendita FAQs

E' possibile eseguire un confronto automatico delle caratteristiche di due o più prodotti selezionandoli mediante i checkbox sul lato destro e facendo click sul pulsante "Confronta".

Le ricerche possono essere salvate nei preferiti e ripetute in seguito.

Invio di un ordine

Accertarsi di essere loggati con il proprio profilo utente.

Inserire i prodotti desiderati nel carrello mediante il pulsante "Acquista".

Aprire la pagina del carrello e fare click sul pulsante "Invia ordine".

Un messaggio comunicherà l'esito dell'operazione e sarà possibile visualizzare il riepilogo dell'ordine.

Nelle sessioni successive sarà possibile visualizzare lo stato del proprio ordine mediante il pulsante "Lista ordini" nell'header.

Bugs e limitazioni note

- I vari tipi di pagamento e spedizione non sono stati modellati. Ciò è accettabile solo se l'amministrazione accetta pagamenti in una sola forma (ad es. PayPal) ed effettua spedizioni a prezzo fisso.
- Alcune pagine hanno dei problemi di scalabilità poiché non c'è limite al numero di elementi visualizzabili (ad es. i risultati delle ricerche);
- Se il DBMS non è configurato per usare la codifica dei caratteri "utf8" potrebbero verificarsi dei problemi con alcuni caratteri "speciali" come le lettere accentate;
- La pagina di confronto diventa particolarmente lenta se i prodotti da confrontare sono molti e/o hanno molte caratteristiche associate (soluzione: ottimizzare la query + mostrare messaggio "confronto in corso").