

Progetto IS – D&D



ODD – Object Design Document

TEAM – DAVIDE ALBANESE, DAVIDE SCUTIERO

1.Introduzione

1.1 Obbiettivi Design Trade-Offs

Dopo la realizzazione del documento RAD (Requirement Analysis Document) e SDD (System Design Document), abbiamo analizzato e descritto in linea di massima quelli che sono i nostri obiettivi, tuttavia senza trattare il lato implementativo.

Prestazioni VS Costi :

Partendo dal presupposto che il budget a nostra disposizione era pari a zero, abbiamo realizzato il nostro progetto con sistemi open source, raggiungendo delle prestazioni abbastanza soddisfacenti. Il database che abbiamo utilizzato risponde bene e senza alcun tipo di problema .

Interfaccia VS Tempo di risposta :

Il tempo di risposta tra server e interfaccia sono più che sufficienti (rapidi) a soddisfare le

esigenze dei vari utenti collegati al Sistema. Ovviamente maggiore sarà la grandezza del database e maggiore sarà il tempo di risposta e ricerca in esso.

Interfaccia vs. Easy-use :

Uno dei nostri obiettivi principali era quello di realizzare un'interfaccia semplice e intuitiva che potesse essere compresa da chiunque si avvicinasse al nostro sistema per la prima volta, anche se non avesse avuto esperienza con sistemi informatici. Essa si presenta in stile "minimal"; è presente solo ciò che realmente è necessario per il funzionamento del sistema vero e proprio.

Costi vs. Mantenimento :

Per quanto riguarda i costi di gestione e mantenimento sono praticamente pari a zero, poiché il sistema è ancora in fase di evoluzione. Si tratta di materiale open source a partire dall'IDE fino al Server utilizzato. E' sottinteso che se un domani il sistema dovesse crescere avrebbe bisogno di un upgrade necessario a mantenere le funzioni d'uso per ogni singolo utente.

1.2 Linee guida per l'interfaccia

Il sistema è multi utente, può accedervi chiunque, utente registrato o utente non registrato che sia.

- All'utente non registrato il sistema dà pieno accesso al sito. Esso può navigare tranquillamente, consultando il catalogo, visualizzando i dettagli dei prodotti, tuttavia non può accedere ad un'area personale né tantomeno può effettuare acquisti.
- L'utente registrato invece deve avere accesso a una sua area dedicata, nella quale può consultare lo storico degli ordini effettuati. Inoltre gli è consentito consultare il catalogo e procedere all'acquisto di uno o più prodotti a seconda di ciò che si vuole fare. Esso riceve i risultati delle richieste effettuate dal sistema, senza preoccuparsi delle operazioni e di come queste sono state effettuate. Avvenuta l'operazione di vendita di uno o più prodotti il sistema decrementa dal DataBase totale il numero delle quantità vendute del prodotto acquistato (aggiorna quindi il DataBase).

1.3 Definizioni acronimi e abbreviazioni

Acronimi

1. **RAD**: Requirements Analysis Document
2. **SDD**: System Design Document
3. **ODD**: Object Design Document
4. **DB**: DataBase
5. **DBMS**: DataBase Management System

Definizioni

- **Utente** : Ha completato la registrazione al sito ; ha il completo accesso alla piattaforma. Può navigare, visualizzare i prodotti e acquistarli, ha diritto ad un'area personale protetta nella quale può apportare modifiche o rivisionare gli ordini effettuati.
- **Utente non registrato** : A differenza di un utente a tutti gli effetti, un utente non registrato può solamente navigare all'interno del sito . Ha completo accesso al catalogo, ma può esclusivamente visualizzare i prodotti e non acquistarli; per questo tipo di operazione occorre essere registrato in modo tale da consentire al DB di tracciare gli ordini effettuati.
- **Amministratore** : L'amministratore è la figura più importante ; gli è concesso di "amministrare" il sito in tutto e per tutto. Può aggiungere nuovi prodotti al catalogo, inserirne descrizioni e quantità disponibili, può eliminare articoli non più presenti all'interno del catalogo. Ha insomma il potere di gestire ogni aspetto del sito.

- **Catalogo** : E' la sezione nella quale vengono visualizzati i prodotti disponibili per l'acquisto. Anche se un prodotto non è disponibile viene visualizzato per consentire all'utente di visualizzarne i dettagli in caso di ipotetico futuro acquisto.
- **Area Utente** : Rappresenta l'area di gestione per l'utente. Qui possono essere modificati i propri dati personali, ma possono anche essere visualizzati gli ordini effettuati con relativo numero di fattura .
- **Prodotto** : Per prodotto si intende un articolo disponibile o non all'interno del sito. Può essere acquistato o semplicemente visualizzato da un utente; può essere aggiunto al catalogo o eliminato dall'amministratore .

1.4- Riferimenti

- Jim Arlow, Ila Neustadt, UML e Unified Process, McGraw-Hill Italia Ian Sommerville,I.

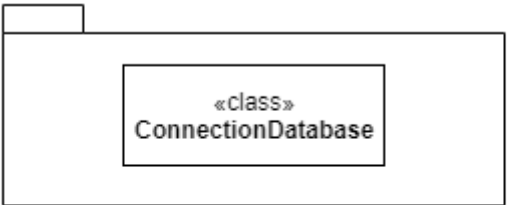
- Bernd Bruegge & Allen H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, (2nd edition), Prentice- Hall, 2003.

2 – Packages

Il nostro sistema è organizzato in tre package principali,ovvero :

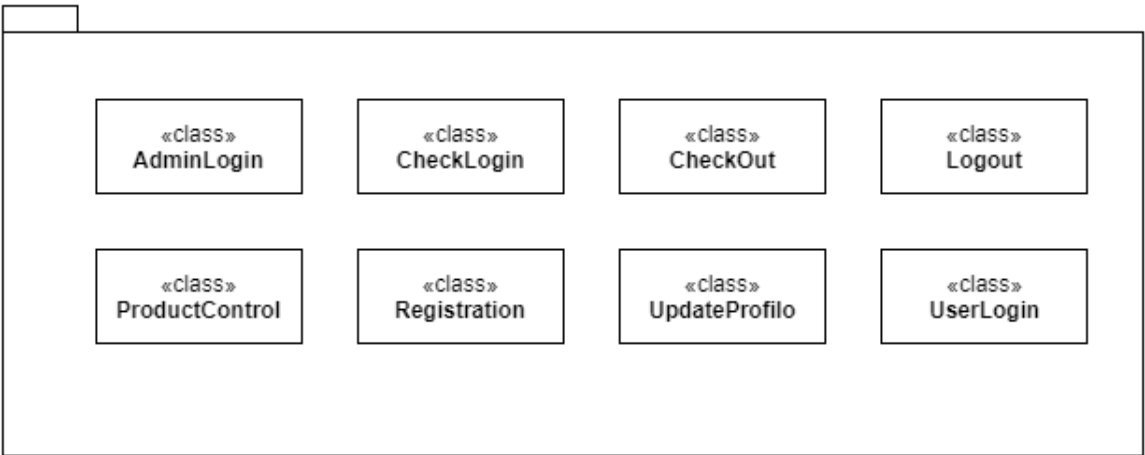
1. ConnectionPool
2. Control
3. Model

2.1 ConnectionPool



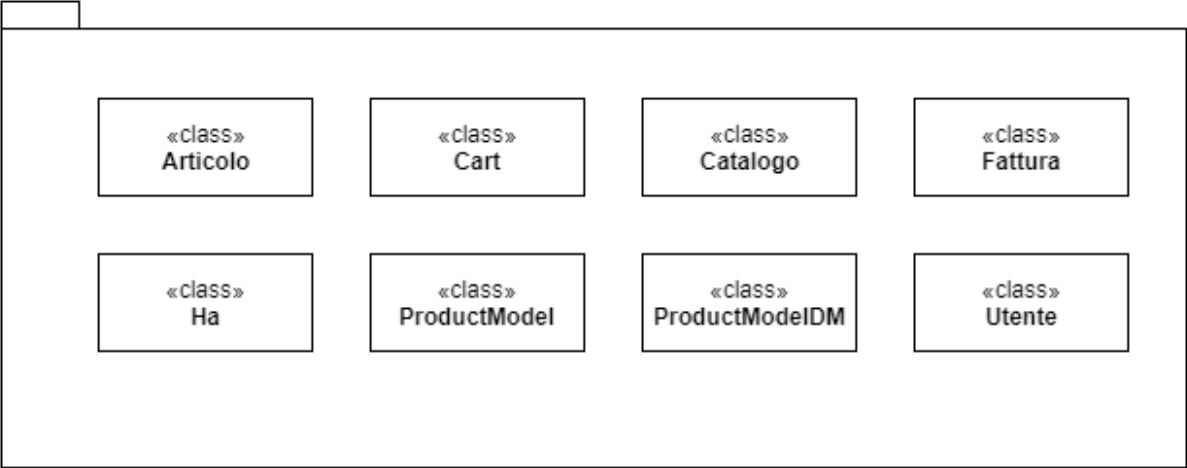
Nome classe	Descrizione
Connection Database	La classe ConnectionDatabase è la classe che si occupa di effettuare la connessione tra server e DB

2.2 Control



Nome classe	Descrizione
AdminLogin	<i>Viene utilizzata per effettuare l'accesso da amministratore all'interno del sito</i>
CheckLogin	<i>Controlla se i dati inseriti al momento del login sono corretti</i>
CheckOut	<i>Consente all'utente di completare un acquisto in modo corretto. Solo un utente registrato utilizza questa classe</i>
LogOut	<i>Serve ad effettuare il logout dal sito</i>
ProductControl	<i>Classe che controlla se un prodotto è disponibile nel catalogo e aggiorna il DB</i>
Registration	<i>Consente a un visitatore di potersi registrare al sito e quindi di diventare un utente a tutti gli effetti. Un amministratore non ha bisogno di registrazione</i>
UpdateProfilo	<i>Consente all'utente di tenere traccia dei propri ordini e delle transazioni effettuate</i>
UserLogin	<i>L'utente deve effettuare l'accesso per poter operare nel sito</i>

2.3 Model

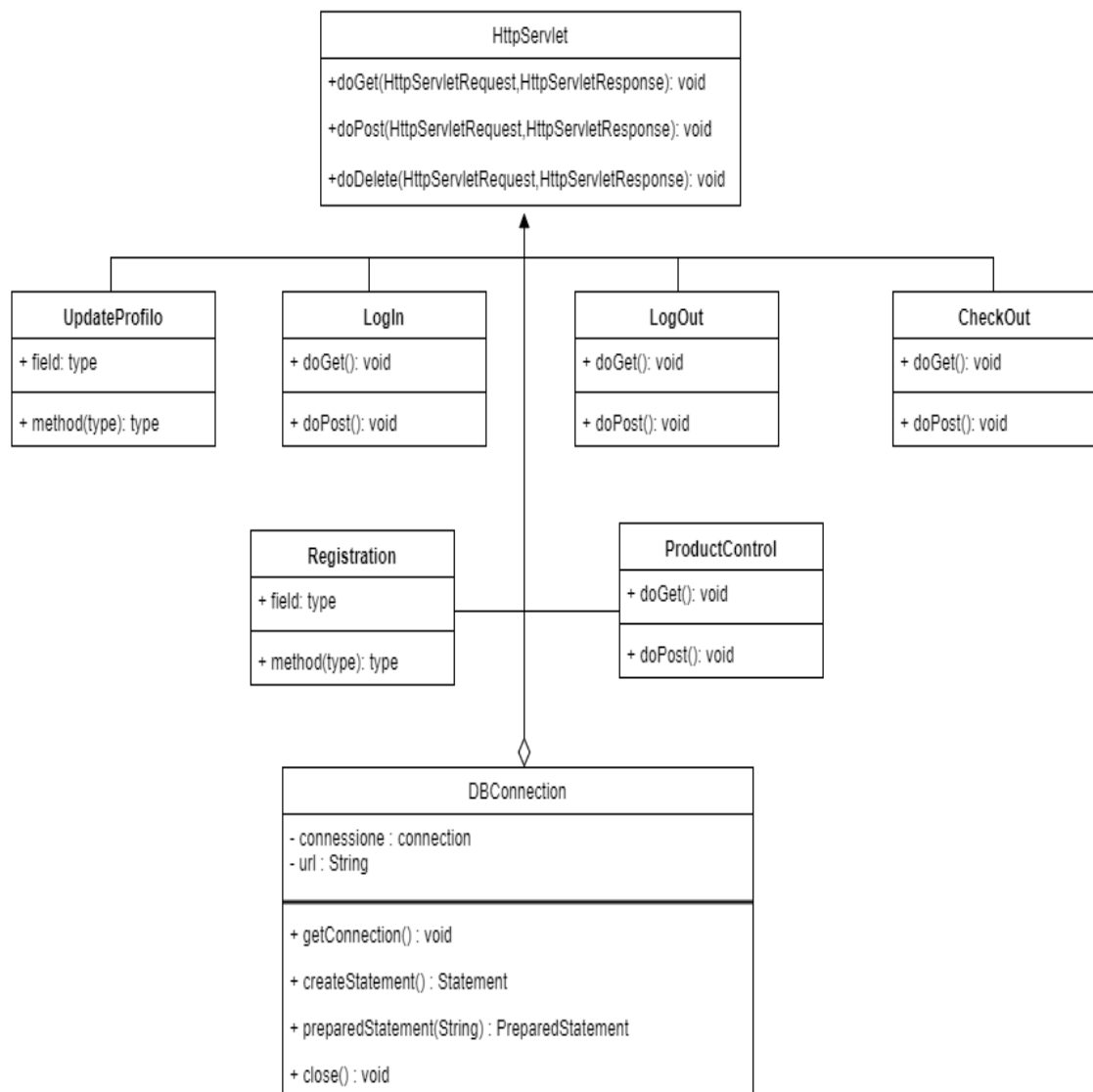


<i>Nome classe</i>	<i>Descrizione</i>
<i>Articolo</i>	<i>Questa classe crea l’oggetto “articolo”, essa rappresenta un prodotto acquistabile situato nel catalogo</i>
<i>Cart</i>	<i>Rappresenta il carrello. Quando un utente acquista uno o più articoli, è qui che vengono depositati temporaneamente ,prima di procedere all’acquisto effettivo</i>

Catalogo	<i>E' forse la sezione più importante. Qui vengono esposti tutti i prodotti presenti e sia un utente non registrato che registrato possono accedere a quest'area</i>
Fattura	<i>Serve all'utente per tenere traccia dell'acquisto effettuato. Il numero della fattura viene generato in base al numero di ordine che il sito ha ricevuto</i>
Ha	<i>Interagisce con la classe Fattura e consente al sistema di associare per ogni utente le proprie fatture</i>
ProductModel	<i>Classe di gestione di un prodotto. Serve a salvare lo stato di un prodotto e ad aggiornarlo, ma anche eliminarlo all'occorrenza</i>
ProductModelDM	<i>Interagisce con ProductModel e serve a gestire tutte le query per poter accedere al DB</i>
Utente	<i>Classe che modella l'oggetto Utente.</i>

3 Class Interfaces

Tutte le classi del sistema estendono la classe HttpServlet ed usano la classe DBManager che si occupa di gestire le connessioni con le basi di date interessate.



❖ **Update Profilo**

Precondizione : Vi deve essere un utente loggato che deve effettuare un acquisto sul catalogo.

Postcondizione : Dopo che l'utente ha effettuato l'acquisto, il profilo viene aggiornato tenendo traccia degli acquisti dello stesso utente; viene memorizzato nel DB.

❖ **Login**

Precondizione : L'utente deve aver inserito i dati negli appositi campi e devono corrispondere ad un utente registrato nel database.

Postcondizione : deve essere creata una sessione con accesso all'area privata dalla quale l'utente può compiere operazioni che a un utente non registrato non sono consentite.

❖ **Logout**

Precondizione : deve esistere già una sessione aperta da un utente.

Postcondizione : la sessione deve essere terminata e il browser deve essere reindirizzato alla home in modo che per accedere all'area privata si deve digitare di nuovo login e password.

❖ **CheckOut**

Precondizione : deve essere stato effettuato un acquisto da parte di un utente.

Postcondizione : l'acquisto deve essere stato completato con successo.

❖ **Registration**

Precondizione : Un visitatore vuole iscriversi al sito; vanno completati i campi richiesti

Postcondizione : I dati inseriti devono essere corretti e non deve esserci un utente già registrato con le stesse credenziali.

❖ **ProductControl**

Precondizione : Deve esserci almeno un prodotto presente all'interno del catalogo

Postcondizione : Il DB deve avere accesso all'elenco dei prodotti disponibili per verificarne la disponibilità .