

Progetto: WHATSHOP	Versione: 2.0
Documento: Object Design Document	Data: 20/01/2017

WHATSHOP
ODD
Object Design Document
Versione 2.0



Data: 20/01/2017

Progetto: WHATSHOP	Versione: 2.0
Documento: Object Design Document	Data: 20/01/2017

Organizzazione del progetto

Partecipanti:

Nome	Matricola
Barone Gerardo	0512103264
Luigi Cerreto	0512102940
Giuseppe D'Avanzo	0512102892
Alessandro Manganiello	0512103246

Scritto da:	Gerardo Barone, Luigi Cerreto, Giuseppe D'avanzo, Alessandro Manganiello
--------------------	--

Revision History

Data	Version e	Descrizione	Autore
16/10/2016	0.5	Abbozzo di stesura	G.B, A.M, G.D.
17/10/2016	1.0	Prima stesura completa	G.B, A.M, G. D, L.C
21/10/2016	2.0	Controllo errori sintattici e grammaticali	G.B, G.D

Sommario

Organizzazione del progetto	2
Revision History.....	3
Sommario	4
1. Introduzione	5
1.1 Object Design Trade-off.....	5
1.2 Linee guida per la documentazione delle interfacce	6
1.3 Definizioni, acronimi e abbreviazioni	10
1.4 Riferimenti	10
2. Packaging e Class interfaces	12
3. Documentazione del riuso	13
3.1 Design Pattern	13
3.2 Component Off-The-Shelf (COTS).....	13
4. Glossario	15

1. Introduzione

1.1 - Obiettivi Design Trade-off

Dopo la finalizzazione del documento RAD (Requirement Analysis Document) e SDD (System Design Document), abbiamo descritto quello che sarà il nostro sistema e dunque i nostri obiettivi, tralasciando gli aspetti di implementazione. Per quanto riguarda la realizzazione del sistema Whatshop, sono stati individuati i seguenti trade-off.

- **Prestazioni VS Costi**

Considerando il nostro sistema, possiamo dire che il budget non eccessivo a nostra disposizione ci ha permesso di realizzare il nostro prodotto utilizzando materiale open source per permetterci di minimizzare i costi e rendendo il sistema molto soddisfacente.

- **Comprensibilità VS Tempo**

Parte fondamentale della progettazione è costituita da codice comprensibile anche per eventuali persone esterne al progetto. Per avere tale requisito è necessario prevedere l'utilizzo di classi con metodi ben identificabili e facilmente interpretabili, utilizzando l'indentazione e una documentazione appropriata del codice sorgente come i commenti. Ciò favorisce anche la comprensibilità, agevolando il processo di mantenimento e di modifica del progetto anche per futuri sviluppatori che non hanno lavorato dall'inizio al progetto stesso. Questo vantaggio tuttavia comporta un incremento del tempo per lo sviluppo e la realizzazione dell'intero sistema.

- **Interfaccia VS Tempo di risposta**

I tempi di risposta tra il nostro server e l'interfaccia non troppo minimale del sito web sono molto rapidi e quindi più che sufficienti per soddisfare le richieste dei vari utenti connessi alla piattaforma.

I tempi di risposta potranno, però, variare a seconda della quantità dei negozi presenti sulla piattaforma e quindi nel database e a seconda della quantità di prodotti inseriti. Dunque maggiore sarà la grandezza del database e maggiori saranno i tempi di inoltro delle richieste nel database.

- **Interfaccia VS Easy-use**

La nostra interfaccia, grazie all'utilizzo delle form e grazie ad una impostazione molto semplice ed intuitiva, permette un facile utilizzo della gestione della piattaforma sia ad un rivenditore esperto che ad un neofita, fino ad arrivare ad un facile utilizzo anche per l'utente iscritto che andrà semplicemente ad acquistare prodotti sul sito web.

- **Costi VS Mantenimento**

I nostri costi sono stati ridotti al minimo grazie all'utilizzo di materiale open source e l'utilizzo di linguaggi Java. Grazie a determinate implementazioni, il sistema può essere facilmente modificato in futuro ed implementato di nuove funzioni, oppure corretto se in presenza di errori.

I costi di mantenimento aumenteranno in rapporto alla quantità di negozi aperti sul nostro sito web e al personale impiegato per la gestione dell'intera piattaforma.

La scelta di utilizzare un database è scaturita dai diversi vantaggi che se ne derivano:

- Gestione consistente dei dati;
- Tempo di risposta basso rispetto all'utilizzo di un file system;
- Accesso veloce e concorrente ai dati;

Ovviamente l'utilizzo di un database comporta l'utilizzo di una grande quantità di memoria nel momento in cui la mole di dati dovesse essere notevole.

1.2 Design Goals

Priorità	ID Design	Descrizione	Categoria	Motivazione
Alta	Tempi di Risposta	La piattaforma deve garantire tempi di risposta molto rapidi(in ordini di secondi) alle richieste degli utenti	Performance	I tempi di risposta brevi sono fondamentali per soddisfare le esigenze degli utenti.
Alta	Memoria	E' necessario avere sufficiente memoria per la gestione e l'immagazzinamento dei dati	Performance	Poiché la memoria ha un costo bisogna gestirla diligentemente

Priorità	ID Design	Descrizione	Categoria	Motivazione
Alta	Robustezza	Il sistema deve essere capace di riconoscere errori commessi dagli utenti e segnalarli tramite opportuni messaggi di errore	Affidabilità	L'utente può non accorgersi di aver commesso un errore ed è quindi compito del sistema segnalarlo
Alta	Disponibilità	Il sistema dovrà essere disponibile ogni qualvolta un utente o un amministratore voglia ricorrere alle funzionalità del sito(salvo per periodi di manutenzione, opportunamente comunicati)	Affidabilità	Il sistema deve essere operativo h24 per garantire un maggior utilizzo da parte degli utenti che acquistano i prodotti ivi presenti.

Priorità	ID Design	Descrizione	Categoria	Motivazione
Alta	Tolleranza ai guasti	Il sistema deve essere operativo anche in caso di malfunzionamenti parziali	Affidabilità	La tolleranza ai malfunzionamenti impedisce all'intero sistema di andare offline e di preservare la consistenza dei dati.
Alta	Sicurezza	L'utente interessato a registrarsi alla nostra piattaforma, avrà un nome utente e una password univoci; la consistenza dei dati è garantita dal fatto che solo gli amministratori del sito possono accedere al database	Affidabilità	La sicurezza è fondamentale sia per tutelare la privacy degli utenti registrati, sia la consistenza dei dati
Bassa	Estendibilità	Il Sistema WhatShop deve essere estendibile per la crescita e la miglioria del sito	Manutenzione	Il sito potrà essere implementato con nuove funzioni in futuro ed è quindi utile garantirne l'estendibilità

Il sistema è multi – utente (cioè può accedervi chiunque, sia un semplice cliente, un rivenditore con un negozio oppure un amministratore del sito web).

Al cliente semplice ed al rivenditore, il sistema nasconde la logica delle operazioni e del codice, fornendogli solamente dei form base utili per effettuare funzioni nel sito web, come ad esempio ricercare prodotti, aggiungere prodotti ad un carrello o creare un nuovo negozio per vendere i propri oggetti.

L'amministratore del sito deve invece avere accesso alle funzionalità avanzate del sito, come ai pannelli di gestione ed amministrazione della piattaforma, al database per verificare errori, ai log per verificare acquisti ed al pannello di gestione degli utenti iscritti, per gestire allontanamenti dei suddetti in casi specifici di utilizzo scorretto del sito web.

Nella scrittura di codice per le classi Java ci si atterrà allo standard [CCJPL] nella sua interezza, con particolare attenzione a:

1. Convenzioni sui nomi (cap. 9 dello standard);
2. Struttura dei file (cap. 3 dello standard);
3. Accesso alle variabili (sez. 10.1 dello standard);
4. Commenti speciali (sez. 10.5.4 dello standard);
5. Utilizzo degli spazi bianchi

1.2 - Definizioni, acronimi e abbreviazioni

ACRONIMI & DEFINIZIONI

CCJPL: Code Conventions for the Java Programming

RAD: Requirements Analysis Document

SDD: System Design Document

ODD: Object Design Document

SQL: Structured Query Language

DB: DataBase

DBMS: DataBase Management System

JSP: Java Server Pages

HTML: HyperText Markup Language

CSS: Cascading Style Sheets

3NF: Terza forma normale

API: Application Programming Interface

BROWSER: IExplorer, Edge, Chrome, Firefox

WebBrowser: Cliente/Rivenditore/Amministratore (utente che accede al sistema)

WebServer: Server gestore dei DataBase.

Riferimenti

Il presente ODD riferisce alle ultime versioni dei precedenti documenti rilasciati, in particolare:

PRADv1_WHATSHOP

SDD_WHATSHOP

2.Packaging e Class interfaces

Il packing e class interface è stato realizzato tramite l'applicativo javadoc utilizzato per la generazione automatica del codice sorgente scritto in linguaggio Java.

3.Documentazione del Riuso

3.1 Design Pattern

Singleton

Uno degli aspetti critici nel funzionamento del sistema è l'accesso ai dati persistenti, per questo motivo si è scelto di delegare ad una singola classe (DBManager) la responsabilità di gestire le connessioni al database. Il problema, causato dall'utilizzo di questa soluzione, è la possibilità di ottenere più istanze di DBManager con conseguente perdita di efficienza dovuta alla creazione e al mantenimento di connessioni non utilizzate. Per evitare questo problema si è deciso di applicare alla classe DBManager il design pattern Singleton.

4. GLOSSARIO

Account: e-mail e password dell'utente registrato; obbligatorie ai fini della registrazione.

Prodotto: descrizione, corredata da immagine, di un prodotto che un utente della piattaforma ha messo in vendita.

DBMS: Database Management System, software progettato per la creazione e la manipolazione di database.

Design Pattern: Un design pattern (traducibile in lingua italiana come schema progettuale, schema di progettazione, schema architetturale), è un concetto che può essere definito "una soluzione progettuale generale ad un problema ricorrente". Si tratta di una descrizione o modello logico da applicare per la risoluzione di un problema che può presentarsi in diverse situazioni durante le fasi di progettazione e sviluppo del software.

HTML: L'Hyper Text Markup Language (HTML; traduzione letterale: linguaggio a marcatori per ipertesti), in informatica è il linguaggio di markup solitamente usato per la formattazione e impaginazione di documenti ipertestuali disponibili nel World Wide Web sotto forma di pagine web.

Javadoc: Javadoc è un applicativo incluso nel Java Development Kit della Sun Microsystems, utilizzato per la generazione automatica della documentazione del codice sorgente scritto in linguaggio Java.

ODD: Object Design Document.

Profilo: insieme delle informazioni (residenza, foto profilo, descrizione, contatti) che un utente può aggiungere, facoltativamente, in seguito alla registrazione.

RAD: Requirements Analysis Document, documento che tratta nel dettaglio l'analisi dei requisiti.

SDD: System Design Document, documento che tratta nel dettaglio della progettazione del sistema, e dei suoi obiettivi.