Università degli Studi di Salerno
Corso di Ingegneria del Software

NowIWant
Object Design Document
Versione 1.0

LOGO PROGETTO

Data: 18/12/2017

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Ferrazzano Pompeo Alessio	0512102928
Citro Antonio	0512102922
Giovanni Lembo	0512103252
Robertazzi Gennaro Alessio	0512103792

Scritto da:	FPA,CA,RGA,LG

Revision History

Data	Versione	Descrizione	Autore
16/12/2017	0.1	Abbozzo di stesura	FPA
17/12/2017	0.5	Prima stesura completa	CA, LG
18/12/2017	1.0	Controllo errori	FPA, CA, RGA, LG

Indice

1. Introduzione

1.1 Object Design Trade-off

Dopo la finalizzazione del documento RAD (Requirement Analysis Document) e l'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 NowIWant, 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 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. 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 nel progetto stesso. Questo vantaggio comporta però un incremento del tempo per lo sviluppo e la realizzazione dell'intero sistema.

- **Interfaccia vs Tempo di risposta**

  L'interfaccia, grazie all'utilizzo delle form e ad un'implementazione molto semplice, permette un facile utilizzo della gestione della piattaforma all'amministratore e alla visione del sito per un utente iscritto che andrà semplicemente ad acquistare prodotti.

- **Costi vs mantenimento**

  I costi sono stati minimizzati 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 in presenza di errori.

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

- Gestione consistente dei dati;
- Tempo 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 il cui la mole dei dati dovesse essere notevole.

1.2 Linee guida per la documentazione delle interfacce

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 i l'immagazzinamento dei dati.	Performance	Poiché la memoria ha un costo bisogna gestirla diligentemente.
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 utilizzare le funzionalità del sito (salvo per periodi di manutenzione).	Affidabilità	Il sistema deve essere operativo 24h per garantire un maggior utilizzo da parte degli utenti che acquistano i prodotti.
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 ed 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 per la consistenza dei dati.
Bassa	Estendibilità	La piattaforma 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: un “semplice” cliente oppure un utente amministratore).
Al cliente “semplice” il sistema nasconde la logica delle operazioni e del codice fornendogli solamente dei form base utili per poter effettuare operazioni sulla piattaforma (ricerca di un prodotto, aggiunta di prodotti nel carrello).
L’amministratore del sito deve invece avere accesso alle funzionalità avanzate del sito, come i pannelli di gestione ed amministrazione del sito.

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

- Accesso alle variabili (sez. 10.1);
- Utilizzo di spazi bianchi.

1.3 Definizioni, acronimi ed abbreviazioni

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 Froma Normale
API: Application Programming Interface
BROWSER: Chrome-Firefox-IE
WebBrowser: Cliente-Amministratore (Utente che accede al sistema)
WebServer: Server gestore dei Database

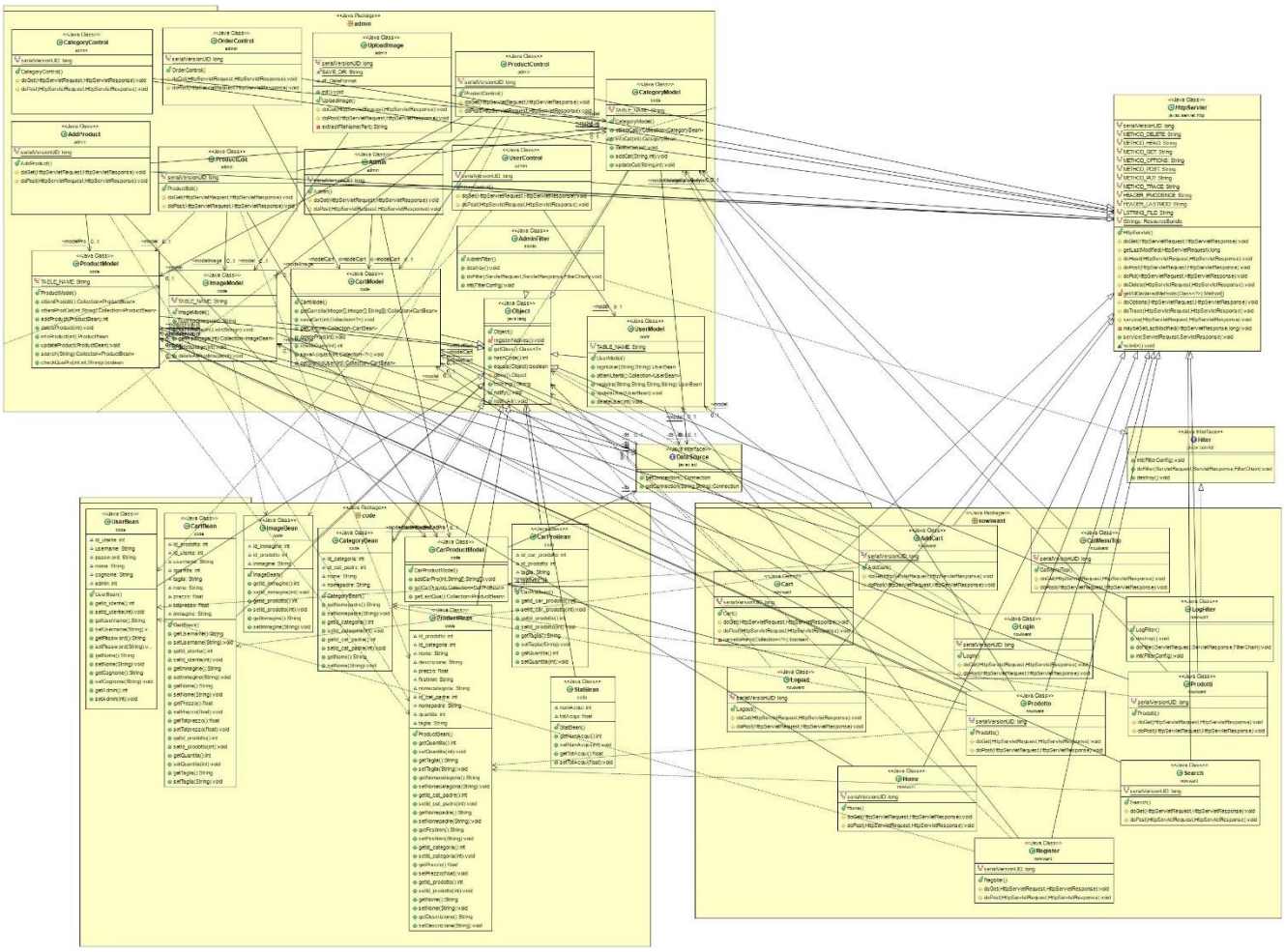
1.4 Riferimenti

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

- RADv1_NOWIWANT;
- SDD_NOWIWANT.

2. Packaging e class interfaces

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



DBMANAGER

NOWIWANT

-URL: String
-CONNESSIONE: String
-st: Statement
-con: Connection

utenti(id_utente,username,password,nome,cognome,admin)
prodotti(id_prodotto,id_categoria,nome,descrizione,prezzo,date_add)
immagini(id_immagine,id_prodotto,immagine)
categorie(id_categoria,id_cat_padre,nome)
car_prodotti(id_car_prodotto,id_prodotto,taglia,quantita)
carrelli(id_carrello,id_utente,id_prodotto,quantita,taglia)
acquisti(id_acquisto,id_utente,id_prodotto,taglia,prezzo,quantita,date_add)

3. Documentazione del riuso

3.1 Design Pattern

Singleton:

Un degli aspetti critici del funzionamento del sistema è l'accesso ai dati persistenti. Per questo motivo si è scelto di delegare ad una singola classe la responsabilità di gestire le connessioni al database. Il problema è la possibilità di ottenere più istanze della classe. Per evitare questo problema si è deciso di applicare nella classe 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.