

DM Shop
Object Design Document
Versione 2.0



Data: 04/02/2017

Progetto: DM Shop	Versione: 2.0
Documento: Object Design Document	04/02/2017

Partecipanti:

Nome	Matricola
Antonucci Gaetano	05121/02960
Carusone Francesco	05121/02252
Pagliarulo Salvatore	05121/03188

Scritto da:	Antonucci Gaetano, Carusone Francesco, Pagliarulo Salvatore
--------------------	---

Revision History

Data	Versione	Descrizione	Autore
30/01/2017	1.1	Revisione package prodotti	Pagliarulo Salvatore
31/01/2017	1.2	Revisione package acquisti	Antonucci Gaetano
04/02/2017	2.0	Aggiunta e revisione package	Pagliarulo Salvatore

Progetto: DM Shop	Versione: 2.0
Documento: Object Design Document	Data: 04/02/2017

Indice

1.	Introduction	5
1.1.	Object design trade-offs	5
1.1.1.	Comprare vs Costruire	5
1.1.2.	Funzionalità vs Tempo di consegna	5
1.1.3.	Linee guida e convenzioni	5
1.2.	Meccanismi gestione delle eccezioni	5
1.3.	Interface documentation guidelines	6
1.4.	Definitions, acronyms, and abbreviations	6
1.5.	References	6
2.	Packages	6
3.	Class interfaces	6
3.1.	Package utenti	7
3.1.1	Class Utente	7
3.1.2	Class LoginControl	7
3.1.3	Class LogoutControl	7
3.1.4	Class RegistrazioneControl	7
3.1.8	Class GestioneUtenti	7
3.2	Package prodotti	7
3.2.1	Class Prodotto	7
3.2.2	Class ButtonAggiungiProdottoControl	7
3.2.3	Class AggiungiProdottoControl	7
3.2.4	Class EliminaProdottoControl	7
3.2.5	Class ButtonModificaProdottoControl	8
3.2.6.	Class VisualizzaProdottoControl	8
3.2.7.	Class ButtonElencoProdottiControl	8
3.2.7	Class ModificaProdottoControl	8
3.2.8	Class GestioneProdotti	8
3.3	Package acquisti	8
3.3.1	Class ButtonAcquistaControl	8
3.3.2	Class AcquistoCarrelloControl	8
3.3.3	Class StoricoAcquistiUtenteControl	8
3.3.4	Class AggiungiProdottoCarrelloControl	8
3.3.5	Class RimuoviProdottoCarrelloControl	8
3.3.6.	Class Carrello	8
3.4	Package ordini	9
3.4.1	Class Ordine	9
3.4.2	Class StatiOrdine	9
3.4.2.	Class DettagliOrdine	9
3.4.3	Class ButtonGestioneOrdiniControl	9
3.4.4	Class ButtonEvadiOrdineControl	9
3.4.5	Class VisualizzaDettagliOrdineControl	9
3.4.6	Class GestioneOrdini	9
3.5.	Package database	9
3.5.1.	Class Database	9

Progetto: DM Shop	Versione: 2.0
Documento: Object Design Document	Data: 04/02/2017

Progetto: DM Shop	Versione: 2.0
Documento: Object Design Document	Data: 04/02/2017

1. Introduction

Il documento descrive gli obiettivi del design imposti dagli sviluppatori e le linee guida per lo sviluppo delle interfacce dei sottosistemi e la suddivisione dei sottosistemi in package e classi.

1.1.Object design trade-offs

Gli obiettivi principali dello sviluppo dal punto di vista degli sviluppatori sono:

- L'applicazione non deve richiedere molte risorse al sistema in termini di memoria e CPU in modo tale da rendere possibile l'utilizzo del sistema anche su macchine non recenti.
- Il tempo di risposta del sistema non dev'essere troppo lento, in particolar modo la sezione che dovrà interagire con l'utente, anche se ci sarà comunque un tempo di tolleranza non molto ristretto.
- Il sistema dovrà sfruttare al meglio la rete senza sovraccaricarla inutilmente, quindi lo spazio occupato dai file dell'applicazione non è fondamentale visto il rapporto costo/spazio dei supporti.

1.1.1. Comprare vs Costruire

Tutte le funzionalità dell'applicazione verranno integrate dagli sviluppatori in modo da ridurre i costi, evitando l'acquisto di librerie.

1.1.2. Funzionalità vs Tempo di consegna

La fase di sviluppo del progetto tenderà di coprire tutte le funzionalità del sistema allungando però i tempi consegna.

1.1.3. Linee guida e convenzioni

I meccanismi per la gestione delle eccezioni sono racchiusi maggiormente nella gestione della connessione tra applicazione e database.

1.1.3.1. Casi limite (Boundary Cases)

Il database che verrà utilizzato, conterrà un valore massimo di lunghezza per i campi in modo tale che ci sia abbastanza spazio in ciascuno di essi.

1.2.Meccanismi gestione delle eccezioni

La gestione delle eccezioni verrà eseguita in modo tale da garantire l'integrità del sistema, infatti le eccezioni verranno sollevate per affrontare qualsiasi tipo di problema riscontrato per permettere una corretta esecuzione del programma. Ad esempio:

- a) Nome utente e password errate:

L'utente inserisce username e password nei campi di testo e clicca il button per accedere al sistema. L'interfaccia utente comunicherà quindi con il sottosistema di gestione degli utenti per verificare i dati inseriti. Nel caso esistano già questi dati oppure la password inserita non è corretta, verrà visualizzato un messaggio di errore durante l'identificazione specificando che l'username o la password non sono corretti.

Progetto: DM Shop	Versione: 2.0
Documento: Object Design Document	Data: 04/02/2017

1.3. Interface documentation guidelines

Le regole utilizzate per classi, metodi, variabili e costanti sono specificate nella tabella sottostante.

Tipo identificato	Regole per Naming	Esempi
Packages	Tutto scritto in minuscolo.	utenti
Classi	La prima lettera del nome della classe è scritta in maiuscolo, il resto in minuscolo. Eventualmente una seconda parola per descrivere meglio la classe inizierà con la lettera maiuscola.	class Utente; class GestioneUtenti;
Metodi	I metodi sono spesso verbi, la prima lettera è in minuscolo e la prima lettera delle parole successive sono in maiuscolo.	acquistaProdotto();
Variabili	Le variabili sia interne che esterne alle classi sono scritte in minuscolo, fatta eccezione per le costanti che sono scritte tutte in maiuscolo.	int prezzo; String nome; static final VARIABILE;

1.4. Definitions, acronyms, and abbreviations

Di seguito sono riportate alcune abbreviazioni usate a volte dagli sviluppatori durante la scrittura del codice:

- obj = object
- id = User ID
- pwd = password

1.5. References

2. Packages

- utenti: contiene tutte le classi che servono a gestire i dati degli utenti del sistema.
- prodotti: contiene tutte le classi che servono a gestire i prodotti.
- ordini: contiene tutte le classi che servono a gestire gli ordini.
- acquisti: contiene tutte le classi che servono a gestire gli acquisti di un utente.

3. Class interfaces

Per avere una descrizione completa delle classi verrà realizzata la documentazione javadoc in un file separato, in modo da avere una documentazione del codice sorgente corretta e dettagliata.

Progetto: DM Shop	Versione: 2.0
Documento: Object Design Document	Data: 04/02/2017

3.1.Package utenti

3.1.1 Class Utente

Questa classe rappresenta un utente del sistema.

3.1.2 Class LoginControl

Questa classe si occupa dell'operazione di login.

3.1.3 Class LogoutControl

Questa classe si occupa dell'operazione di logout.

3.1.4 Class RegistrazioneControl

Questa classe si occupa dell'operazione di registrazione.

3.1.5 Class EliminaUtenteControl

Questa classe si occupa dell'operazione di eliminazione di un utente.

3.1.6 Class VisualizzaElencoUtentiControl

Questa classe si occupa dell'operazione di visualizzazione dell'elenco degli utenti.

3.1.7 Class VisualizzaDatiUtenteControl

Questa classe si occupa dell'operazione di visualizzazione dei dati di un utente.

3.1.8 Class GestioneUtenti

Questa classe si occupa di gestire gli utenti e le loro operazioni ed il relativo collegamento col database.

3.2 Package prodotti

3.2.1 Class Prodotto

Questa classe rappresenta un prodotto.

3.2.2 Class ButtonAggiungiProdottoControl

Questa classe si occupa di caricare la jsp contenente il form per l'inserimento di un prodotto.

3.2.3 Class AggiungiProdottoControl

Questa classe si occupa dell'operazione di aggiunta di un prodotto al catalogo.

3.2.4 Class EliminaProdottoControl

Questa classe si occupa dell'operazione di rimozione di un prodotto dal catalogo.

Progetto: DM Shop	Versione: 2.0
Documento: Object Design Document	Data: 04/02/2017

3.2.5 Class ButtonModificaProdottoControl

Questa classe si occupa di caricare la jsp contenente il form per la modifica di un prodotto.

3.2.6. Class VisualizzaProdottoControl

Questa classe si occupa della visualizzazione di un prodotto.

3.2.7. Class ButtonElencoProdottiControl

Questa classe si occupa della visualizzazione dell'elenco dei prodotti.

3.2.7 Class ModificaProdottoControl

Questa classe si occupa dell'operazione di modifica di un prodotto.

3.2.8 Class GestioneProdotti

Questa classe si occupa di gestire i prodotti e le loro operazioni ed il relativo collegamento col database.

3.3 Package acquisti

3.3.1 Class ButtonAcquistaControl

Questa classe viene chiamata nel momento in cui l'utente vuole acquistare i prodotti e visualizzerà la pagina di riepilogo di tutti i prodotti nel carrello.

3.3.2 Class AcquistoCarrelloControl

Questa classe viene chiamata nel momento in cui l'utente vuole procedere all'acquisto dei prodotti all'interno del suo carrello.

3.3.3 Class StoricoAcquistiUtenteControl

Questa classe si occupa dell'operazione di visualizzazione degli acquisti di un utente per quest'ultimo.

3.3.4 Class AggiungiProdottoCarrelloControl

Questa classe si occupa di aggiungere un prodotto all'interno del carrello dell'utente che l'ha richiesto.

3.3.5 Class RimuoviProdottoCarrelloControl

Questa classe si occupa di rimuovere un prodotto all'interno del carrello dell'utente che l'ha richiesto.

3.3.6. Class Carrello

Questa classe si occupa di gestire il carrello dell'utente.

Progetto: DM Shop	Versione: 2.0
Documento: Object Design Document	Data: 04/02/2017

3.4 Package ordini

3.4.1 Class Ordine

Questa classe rappresenta un ordine.

3.4.2 Class StatiOrdine

Questa classe rappresenta lo stato di un ordine.

3.4.2. Class DettagliOrdine

Questa classe specifica i dettagli di un ordine.

3.4.3 Class ButtonGestioneOrdiniControl

Questa classe si occupa di caricare la jsp contenente l'elenco degli ordini da evadere.

3.4.4 Class ButtonEvadiOrdineControl

Questa classe si occupa dell'operazione di evasione di un ordine e verrà chiamata nel momento in cui il gestore cliccherà sul button "EvadiOrdine" e consisterà nel cambio di stato dell'ordine relativo.

3.4.5 Class VisualizzaDettagliOrdineControl

Questa classe si occupa di visualizzare i dettagli di un ordine.

3.4.6 Class GestioneOrdini

Questa classe si occupa di gestire gli ordini e le loro operazioni ed il relativo collegamento col database.

3.5.Package database

3.5.1. Class Database

Questa classe si occupa della gestione del database.