

**INGEGNERIA DEL SOFTWARE 2016-2017**

**Documento di Object Design 1.0**

**Fiorazon**

**15/12/2016**

**Sommario**

[**1.** **Introduction** 3](#_Toc469562348)

[**I.** **Object design trade-offs** 3](#_Toc469562349)

[**II.** **Interface documentation guidelines** 4](#_Toc469562350)

[**III.** **Definitions, acronyms, and abbreviations** 4](#_Toc469562351)

[**IV.** **References** 5](#_Toc469562352)

[**2.** **Packages** 5](#_Toc469562353)

[**3.** **Class Interfaces** 6](#_Toc469562354)

[**4.** **Team** 8](#_Toc469562355)

[**5.** **Storia delle versioni** 8](#_Toc469562356)

### **Introduction**

### **Object design trade-offs**

La nostra piattaforma sarà completamente sviluppata senza l’ausilio di codice scritto da terzi.

I motivi di questa scelta ricadono sul fatto che il riutilizzo di codice di terzi:

* Appesantirebbe il caricamento delle pagine web e aumenterebbe il tempo di computazione dei servizi offerti dal server, in quanto il codice avrà molto probabilmente funzioni inutilizzate nel normale utilizzo dell’applicazione stessa;
* Utilizzare codice preconfezionato da terzi produce l’inconveniente di doverlo modificare in caso di un bug o di una modifica su qualche funzioni, il che non sarebbe tanto semplice e veloce come soluzione considerando che il codice non è stato prodotto da nessun membro del progetto.

Nell’ applicazione esistono alcune situazioni che potrebbero provocare errori nel sistema e vengono gestiti nei seguenti modi:

* **Username o password sbagliata:**

riscontrando questo problema il sistema mostra un messaggio d’ errore che identifica il problema.

* **Inserimento data nascita che identifica un utente minorenne nella fase di registrazione oppure nella fase della modifica dell’anagrafica dell’utente:**

Il sistema riscontrando il problema in primo luogo impedisce la modifica o la registrazione e inoltre nella pagina in cui si trova a seconda del caso compare un messaggio di errore che lo identifica.

* **Utilizzo di un username già utilizzata da un altro utente nella fase di registrazione:**

L’utilizzo di un username già utilizzata non è permesso, quindi di conseguenza il sistema non permette di continuare la registrazione e quindi mostra un messaggio di errore che ne spiega i motivi e mostra alcuni username no utilizzate che potrebbero andare bene.

* **Inserimento di un prodotto già esistente nel catalogo:**

Per evitare problemi di incongruenza nei dati dei prodotti è fatto assoluto divieto inserire un prodotto già esistente. Il sistema riscontrando il problema non permette di continuare l’operazione e mostra un messaggio di errore nella pagina.

* **Procedere con l’ordine del carrello senza essere loggati:**

L’ applicazione per permettere di effettuare un ordine obbliga l’utente ad effettuare il login oppure la registrazione se non l’ha fatto ancora. Quindi il sistema trovandosi nel punto in cui un utente non registrato voglia produrre un ordine, gli mostra un massaggio di errore e contemporaneamente anche la pagina che permetta di fare il login oppure la registrazione.

### **Interface documentation guidelines**

|  |  |  |
| --- | --- | --- |
| **Identifier Type** | **Rules for Naming** | **Examples** |
| **Package** | Il nome di ogni package deve essere scritto tutto in minuscolo, inoltre deve essere un nome molto sintetico e al contempo deve essere rispecchiare il suo contenuto | package beans |
| **Classe** | Il nome di ogni classe deve iniziare per lettera maiuscola, mentre le restanti lettere dovranno essere minuscole ad eccezione di un nome composto da più parole, in cui e consentito differenziare la parola mettendo la prima lettera di essa in maiuscolo. E infine il nome deve essere fortemente significativo in modo da facilitare la comprensione del ruolo della suddetta classe. | Class GestisciCarrello |
| **Interfaccia** | I nomi delle interfacce seguono le stesse regole delle classi. | Interface ModificaCampo |
| **Metodo** | I nomi dei metodi devono iniziare per un verbo scritto in minuscolo, che rappresenti l’effetto del suo utilizzo.  Esso può essere succeduto da altre parole, le quali devono iniziare con lettera maiuscola e continuare fino alla fine in minuscolo. | inserisciProdotto(); |
| **Variabile** | I nomi delle variabili devono essere brevi e coincisi, e anch’esse devono iniziare per lettera minuscola, mentre le parole interne per maiuscola | Int contaUtenti; |
| **Costante** | I nomi delle costanti devono essere scritti tutti in maiuscolo e separati da “\_” se composti da più parole. | COSTANTE\_PI\_GRECO |

### **Definitions, acronyms, and abbreviations**

**Package:** collezione di classi e interfacce correlate.

**Classe:** è identificabile come un tipo di dato astratto.

**Interfaccia:** è un tipo analogo alla classe, ma vincolato a non definire l'implementazione dei propri metodi.

**Metodo:** è un sottoprogramma associato in modo esclusivo ad una classe e che rappresenta.

**Variabile:** è un insieme di dati modificabili situati in una porzione di memoria destinata a contenere dei dati **.**

**Costante:** identifica una porzione di memoria il cui valore non varia nel corso dell'esecuzione di un programma.

### **References**

* Requirements Analysis Document
* System Design Document

### **Packages**

Individuiamo 4 packages in cui dividere il codice, uno per ogni sotto-sistema:   
-carrelloPackage  
-prodottiPackage  
-utentiPackage  
-acquistoPackage

Il package carrelloPackage contiene i file “Carrello.jsp”, “Carrello.java”, “ControlloAggiuntaProdottoAlCarrello.java”, “CarrelloManager.java”, “RimuoviProdottoCarrello.java” responsabili dell’interazione con il carrello.  
Il file ”carrello.jsp” interagisce con ControlloAcquistaCarrello.java del package acquistoPackage durante l’operazione di acquisto del carrello

Il package prodottiPackage “AggiungiProdottoCatalogo.jsp”, “ControlloAggiungiProdottiCatalogo.java”, “ProdottiManager.java”, “Prodotto.java”, “CancellaProdotto.jsp”, “ControlloCancellaProdotto.java”, “modificaProdotto.jsp”, “ControlloModificaProdotto.java”, “Catalogo.jsp”, “Risultati.jsp”, “RicercaProdotti.java”.   
Esso contiene tutte quelle pagine .jsp e .java responsabili delle operazioni riguardanti i prodotti.  
Il file “Catalogo.jsp” interagisce con “ControlloAggiuntaProdottoAlCarrello.java” di carrelloPackage durante l’operazione di aggiunta di un prodotto al carrello.

Il package utentiPackage contiene i file “PaginaAmministratore.jsp”, “Utente.java”, “PaginaUtente.jsp”, “ControlloCancellazioneUtenti.java”, “UtentiManager.java”, “PaginaLogin.html”, “ControlloLoginAmministratore.java”, “ControlloLoginUtente.java”, “ControlloLogoutAmministratore.java”, “ControlloLogoutUtenti.java”, “ControlloModificaDatiUtente.java”, “PaginaRegistrazione.jsp”, “ControlloRegistrazione.java”.  
Esso contiene tutte quelle pagine .jsp e .java responsabili di quelle operazioni riguardanti gli utenti e l’amministratore.  
Il file “PaginaAmministratore.jsp” interagisce con “aggiungiProdottoCatalogo.jsp” del package prodottiPackage durante l’operazione di aggiunta di un prodotto al catalogo, con “OrdiniAmministratore.jsp” del package acquistoPackage durante l’operazione di aggiornamento dello stato di un ordine.  
Il file “PaginaUtente.jsp” interagisce con “OrdiniUtente.jsp” del package acquistoPackage durante l’operazione di conferma dell’arrivo di un ordine.

Il package acquistoPackage contiene i file “pagamento.jsp”, “ControlloAcquistaCarrello.java”, “ControlloPagamento.java”, “OrdiniManager.java”, “Ordine.java”, “OrdiniAmministratore.jsp”, “ControlloAvanzamentoOrdini.java”, “OrdiniUtente.jsp”, “ControlloConfermaOrdine.java”.   
Esso contiene quei file responsabili di eseguire l’acquisto e di gestire la vita di un ordine.

### **Class Interfaces**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Interfaccia** | **Descrizione** |
| Carrello.java | Metodi get e set | Questa è la classe java bean del carrello |
| ControlloAggiunta ProdottiAlCarrello.java | Metodi get e post | Questa è la servlet responsabile dell’aggiunta di un prodotto al carrello |
| RimuoviProdotto Carrello.java | Metodi get e post | Questa è la servlet responsabile della rimozione di un prodotto dal carrello |
| CarrelloManager.java | aggiungiProdottoAlCarrello() rimuoviProdottoDalCarrello()  getCarrello() | Questa è la classe manager del carrello |

|  |  |  |
| --- | --- | --- |
| **Nome** | **Interfaccia** | **Descrizione** |
| Prodotto.java | Metodi get e set | Questa è la classe java bean del prodotto |
| AggiungiProdotto  Catalogo.java | Metodi get e post | Questa è la servlet responsabile dell’aggiunta di un prodotto al catalogo |
| ControlloCancella Prodotto.java | Metodi get e post | Questa è la servlet responsabile della rimozione di un prodotto al catalogo |
| ControlloModifica Prodotto.java | Metodi get e post | Questa è la servlet responsabile della modifica di un prodotto al catalogo |
| RicercaProdotti.java | Metodi get e post | Questa è la servlet responsabile della ricerca dei prodotti |
| ProdottiManager.java | aggiungiProdottoCatalogo()  cancellaProdottoCatalogo()  controlloModificaProdotto()  ricercaProdotti() | Questa è la classe manager del catalogo |

|  |  |  |
| --- | --- | --- |
| **Nome** | **Interfaccia** | **Descrizione** |
| Utenti.java | Metodi get e set | Questa è la classe java bean dell’utente |
| ControlloLogin Amministratore.java | Metodi get e post | Questa è la servlet responsabile del login dell’amministratore |
| ControlloLogin  Utente.java | Metodi get e post | Questa è la servlet responsabile del login dell’utente |
| ControlloLogout  Utente.java | Metodi get e post | Questa è la servlet responsabile del logout di un utente |
| ControlloLogout  Amministratore.java | Metodi get e post | Questa è la servlet responsabile del logout dell’amministratore |
| ControlloModifica DatiUtente.java | Metodi get e post | Questa è la servlet responsabile della modifica dei dati di un utente |
| Controllo Registrazione.java | Metodi get e post | Questa è la servlet responsabile della registrazione di un utente |
| Controllo  Cancellazione.java | Metodi get e post | Questa è la servlet responsabile della cancellazione di un utente |
| UtentiManager.java | isUser()  isAdmin()  modificaDatiUtente()  registraNuovoUtente()  cancellaUtente() | Questa è la classe manager dell’utente |

|  |  |  |
| --- | --- | --- |
| **Nome** | **Interfaccia** | **Descrizione** |
| Ordine.java | Metodi get e set | Questa è la classe java bean dell’ordine |
| ControlloAcquisto Carrello.java | Metodi get e post | Questa è la servlet responsabile dell’acquisto del carrello |
| Controllo Pagamento.java | Metodi get e post | Questa è la servlet responsabile del pagamento del carrello |
| ControlloAvanzamento  Ordini.java | Metodi get e post | Questa è la servlet responsabile dell’avanzamento dello stato di spedizione dell’ordine |
| ControlloConferma  Ordini.java | Metodi get e post | Questa è la servlet responsabile della conferma dell’arrivo di un ordine |
| OrdiniManager.java | creaOrdine()  ModficaOrdine()  AvanzaStatoOrdine()  ConferemaArrivoOrdine() | Questa è la classe manager dell’ordine |

### **Team**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Giuseppe Siani | 0512102958 |
| Carmine D’ Alessandro | 0512103402 |
| Aleandro Giuseppe Libano | 0512102976 |

### **Storia delle versioni**

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | VERSIONE | DESCRIZIONE | AUTORE |
| 14/12/2016 | 1.0 | Completamento ODD | Giuseppe Siani, Carmine D’Alessandro |
| 26/12/2016 | 1.1 | Correzione errori | Carmine D’Alessandro |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |