|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

**Edil CommerceDesign  
Object Design Document  
Versione 1.0**

****

Data: 18/01/2023

Partecipanti:

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| De Chiara Luigi | 0512109483 |
| Ferrare Alex | 0512106300 |

Sommario

[**1.Introduzione** 2](#_Toc125703479)

[**2.Object Design Trade-Offs** 3](#_Toc125703480)

[**3. Packeges** 3](#_Toc125703481)

[**3.1. Organizzazione Packeges** 4](#_Toc125703482)

[**3.2. Back-end Packeges** 5](#_Toc125703483)

[**3.3. Front-end Packeges** 7](#_Toc125703484)

# **1.Introduzione**

L’object design document è stato realizzato per aggiungere dettagli all’analisi dei requisiti e per prendere decisioni implementative.

In particolare, nel documento saranno definite le interfacce delle classi, le operazioni, i tipi, gli argomenti e le signatures dei sottosistemi definiti nel System Design Document.

Verrà effettuata l’integrazione di tutte le funzionalità individuate nelle fasi precedenti attraverso la produzione di un modello.

# **2.Object Design Trade-Offs**

**Costi di sviluppo Vs. Portabilità**

Gli sviluppatori implementeranno il sistema in modo tale che possa essere fruibile tramite una lunga lista di dispositivi, impattando sul budget stabilito.

**Sicurezza Vs. Costo**

Gli sviluppatori realizzeranno una logica di sicurezza per la privacy, che impatterà sul budget prefissato.

# **3. Packeges**

Back-end

* Control
* Model
* Utils

Front-end

* Admin
* CSS
* Script
* User
* WebContent

## **3.1. Organizzazione Packeges**

Immagine che contiene testo

Descrizione generata automaticamenteImmagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamenteImmagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

## **3.2. Back-end Packeges**

|  |  |
| --- | --- |
| **Control** | |
| **Classe** | **Descrizione** |
| AggiornaQ | Servlet che gestisce la variazione di quantità di un articolo nel carrello. |
| AggiornaSerach | Servlet che gestisce l’ordinamento degli articoli in base a vari criteri. |
| Aggiungi | Servlet che gestisce l’aggiunta di un articolo al carrello. |
| AggiungiArticolo | Servlet che gestisce l’aggiunta di un articolo al catalogo. |
| Autocomplete | Servlet che gestisce l’autocompletamento del nome degli articoli nella SearchBar. |
| ComputaOrdine | Servlet che gestisce l’ordine di un utente |
| Login | Servlet che gestisce l’accesso alla piattaforma |
| Logout | Servlet che gestisce il logout |
| Modifica | Servlet che gestisce la modifica dei dati di un utente. |
| ModificaArticolo | Servlet che gestisce la modifica di un articolo al catalogo. |
| OperaRecensione | Servlet che gestisce le recensioni degli articoli. |
| Registrazione | Servlet che gestisce la registrazione alla piattaforma |
| Rimuovi | Servlet che gestisce la rimozione di un articolo dal carrello. |
| Search | Servlet che gestisce la ricerca degli articoli del catalogo. |

|  |  |
| --- | --- |
| **Model** | |
| **Classe** | **Descrizione** |
| ArticoloBean | Classe che descrive le caratteristiche di un Articolo. |
| ArticoloModelDS | Descrive l’interazione col database per Articolo. |
| Carrello | Classe che utilizzata per la gestione del carrello. |
| CartaBean | Classe che descrive le caratteristiche di una Carta di credito. |
| CartaModelDS | Descrive l’interazione col database per la carta di credito. |
| CategoriaBean | Classe che descrive le caratteristiche di una categoria. |
| CategoriaModelDS | Descrive l’interazione col database per le categoria. |
| ComponeBean | Classe che descrive le caratteristiche di compone ordine. |
| ComponeModelDS | Descrive l’interazione col database per definire la tipologia di prodotti che compongono un ordine. |
| ContrassegnoBean | Classe che descrive le caratteristiche di contrassegno. |
| ContrassegnoModelDS | Descrive l’interazione col database per il contrassegno. |
| InfoFatturazioneBean | Classe che descrive le caratteristiche le informazioni di fatturazione. |
| InfoFatturazioneModelDS | Descrive l’interazione col database per le informazioni di fatturazione. |
| OrdineBean | Classe che descrive le caratteristiche di un ordine. |
| OrdineModelDS | Descrive l’interazione col database per l’ordine. |
| RecensisceBean | Classe che descrive le caratteristiche di una recensione. |
| RecensiceModelDS | Descrive l’interazione col database per la recensione. |
| RuoloUserBean | Classe che descrive le caratteristiche dei ruoli per gli user. |
| RuoloUserModelDS | Descrive l’interazione col database per Ruolo User. |
| UserBean | Classe che descrive le caratteristiche di un utente. |
| UserModelDS | Descrive l’interazione col database per User. |

|  |  |
| --- | --- |
| **Utils** | |
| **Classe** | **Descrizione** |
| MainContext | File di configurazione dell’accesso al database tramite i driver JDBC. |
| MainSession | Classe che gestisce la sessione. |
| ValidazioneInput | Classe che gestisce gli input esterni. |

## **3.3. Front-end Packeges**

|  |  |
| --- | --- |
| **Admin** | |
| **Classe** | **Descrizione** |
| Admin | Pagina per la manutenzione del catalogo. |

|  |  |
| --- | --- |
| **User** | |
| **Classe** | **Descrizione** |
| Carrello | Pagina dedicata al carrello. |
| Chechout | Pagina dedicata alla conferma dell’ordine |
| ordineEffettuato | Pagina dedicata all’ordine effettuato. |
| Profilo | Pagina dedicata alle informazione dell’utente. |

|  |  |
| --- | --- |
| **WebContent** | |
| **Classe** | **Descrizione** |
| Articolo | Pagina dedicata alla visualizzazione dell’articolo. |
| Home | Pagina dedicata all’homepage. |
| Login | Pagina dedicata all’autenticazione degli utenti. |
| Registrazione | Pagina dedicata alla registrazione degli utenti. |
| Result | Pagina dedicata alla visualizzazione del catalogo. |

# **4. Class Interface**

(private =”-“, protected=”#”, public=”+”)

|  |  |
| --- | --- |
| **Nome classe** | AggiornaQ |
| **Descrizione** | Servlet che gestisce la variazione di quantità di un articolo nel carrello. |
| **Signature dei metodi** | # doGet (HttpServletRequest, HttpServletResponse): void  Action:   * Carrello |
| **Pre-condizione** |  |
| **Post-condizione** |  |
| **invariante** |  |

|  |  |
| --- | --- |
| **Nome classe** | AggiornaQ |
| **Descrizione** | Servlet che gestisce la variazione di quantità di un articolo nel carrello. |
| **Signature dei metodi** | # doGet (HttpServletRequest, HttpServletResponse): void  Action:   * Carrello |
| **Pre-condizione** |  |
| **Post-condizione** |  |
| **invariante** |  |