

# **Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

**EdenJewelry**  
**ODD\_EdenJewelry**  
**Versione 1.1**

Data: 16/12/2024

Progetto: EdenJewelry	Versione: 1.1
Documento: SDD_EdenJewelry	Data: 16/12/2024

**Coordinatore del progetto:**

Nome	Matricola

**Partecipanti:**

Nome	Matricola
Alessandro Di Palma	0512115939
Gaetano D'Alessio	0512110836
Luigi Montuori	0512117799
Miriam Eva De Santis	0512117121

<b>Scritto da:</b>	Luigi Montuori
--------------------	----------------

**Revision History**

Data	Versione	Descrizione	Autore
04/12/2024	1.0	Prima stesura del documento	Luigi Montuori
16/12/2024	1.1	Specifiche interfacce dei sottosistemi	Miriam Eva De Santis

Indice

1. INTRODUZIONE.....4

1.1. Object design trade offs.....4

1.2. Interface documentation guidelines .....4

1.3. Definitions, acronyms, and abbreviations .....4

1.4. References .....4

4. Specifiche interfacce dei sottosistemi .....4

# 1. INTRODUZIONE

## 1.1. Object Design Trade Offs

Funzionalità Vs Usabilità	Abbiamo preferito creare uno shop semplice da usare, piuttosto che concentrarci su implementare molte funzionalità ridondanti.
Costo Vs Robustezza	Un giusto compromesso tra un sito robusto, ma che non richieda troppo tempo per essere implementato.
Sviluppo rapido Vs Funzionalità	Favorire e facilitare lo sviluppo, riducendolo ad una serie di funzionalità essenziali.

## 1.2. Interface documentation guidelines

## 1.3. Definitions, acronyms, and abbreviations

## 1.4. References

- SDD\_EdenJewlry

## 4. Specifiche interfacce dei sottosistemi

### 4.1 Carrello\_interface

Carrello_interface	
Descrizione	Questa interfaccia permette di interagire con l'entità Carrello.
Pre- condizione	<pre>context Carrello_interface::addToCart(idProdotto: Integer) pre: idProdotto &lt;&gt; null and idProdotto &gt; 0  context Carrello_interface::deleteItem(idProdotto: Integer) pre: idProdotto &lt;&gt; null and idProdotto &gt; 0  context Carrello_interface::modificaQuantita(idProdotto: Integer) pre: idProdotto &lt;&gt; null and idProdotto &gt; 0</pre>

Post- condizione	<pre>context Carrello_interface::addToCart(idProdotto: Integer)   Carrello -&gt;includes(prodotto)  context Carrello_interface::deleteItem(idProdotto: Integer)   Carrello -&gt;not includes(prodotto)  context Carrello_interface::modificaQuantita(userId: Integer)   Carrello.(quantita aumentata)    Carrello.(quantita diminuita)</pre>
Invariante	

## 4.2CatalogoUtenti\_interface

CatalogoUtenti_interface	
Descrizione	Questa interfaccia permette di interagire con l'entità CatalogoUtenti
Pre- condizione	<p>Context</p> <p>CatalogoUtenti_interface::addUtente(idUtente: Integer) pre:</p> <ul style="list-style-type: none"><li>idUtente &lt;&gt; null</li><li>and idUtente &gt; 0</li></ul> <p>context</p> <p>CatalogoUtenti_interface::removeUtente(idUtente: Integer) pre:</p> <ul style="list-style-type: none"><li>idUtente &lt;&gt; null</li><li>and idUtente &gt; 0</li></ul> <p>context</p> <p>CatalogoUtenti_interface::UtenteById(idUtente: Integer) pre:</p> <ul style="list-style-type: none"><li>idUtente &lt;&gt; null</li><li>and idUtente &gt; 0</li></ul>

## Post- condizione

context

CatalogoUtenti\_interface::addUtente

(idUtente: Integer) post:

Catalogo->includes(Utente)

context

CatalogoUtenti\_interface::removeUte

nt(idUtente:Integer) post:

Catalogo->not includes(Utente)

context

UserIDS::UtenteById(idUtente:

Integer) post:

return result=Utente.allinstance->(SELECT \* FROM Utenti p where  
p.idUtente:idUtente)

#### 4.2Catalogo\_interface

Catalogo_interface	
Descrizione	Questa interfaccia permette di interagire con l'entità Catalogo
Pre- condizione	Context Catalogo_interface::addProdotto(idProdotti: Integer) pre: idProdotti <> null and idProdotti > 0
	context Catalogo_interface::deleteProdotto(idProdotti:: Integer) pre: idProdotti <> null and idProdotti > 0  context CatalogoUtenti_interface::validate(idProdotti: User) pre: idProdotti <> null and idProdotti > 0



Post- condizione	<p>Context</p> <p>Catalogo_interface::addProdotto(idProdotti: Integer) post:</p> <p>Catalogo-&gt;includes(Prodotto)</p> <p>context</p> <p>Catalogo_interface::deleteProdotto(idProdotti: Integer) post:</p> <p>Catalogo-&gt;not includes(Prodotto)</p> <p>context</p> <p>CatalogoUtenti_interface::validate(idProdotti: User) post:</p> <p>int id= Prodotto.allinstance-&gt;(SELECT * FROM Prodotto p where p.idprodotto: idprodotto)</p> <p>if(id&lt;&gt;null) return true; else return false;</p>
------------------	---

#### 4.3Wishlist\_interface

Wishlist_interface	
Descrizione	Questa interfaccia permette di interagire con l'entità Wishlist
Pre- condizione	<p>Context</p> <p>Wishlist_interface::addToList(idProdotti: Integer) pre:</p> <ul style="list-style-type: none"><li>idProdotti &lt;&gt; null and</li><li>idProdotti &gt; 0</li></ul> <p>context</p> <p>Wishlist_interface::removeFromList(idProdotti: Integer) pre:</p> <ul style="list-style-type: none"><li>idProdotti &lt;&gt; null and</li><li>idProdotti &gt; 0</li></ul>
Post- condizione	<p>Context</p> <p>Catalogo_interface::addToList(idProdotti: Integer) post:</p> <ul style="list-style-type: none"><li>Wishlist-&gt;includes(Prodotto)</li></ul> <p>context</p> <p>Catalogo_interface::deleteProdotto(idProdotti: Integer) post:</p>

	Wishlist-> not includes(Prod to)
--	--