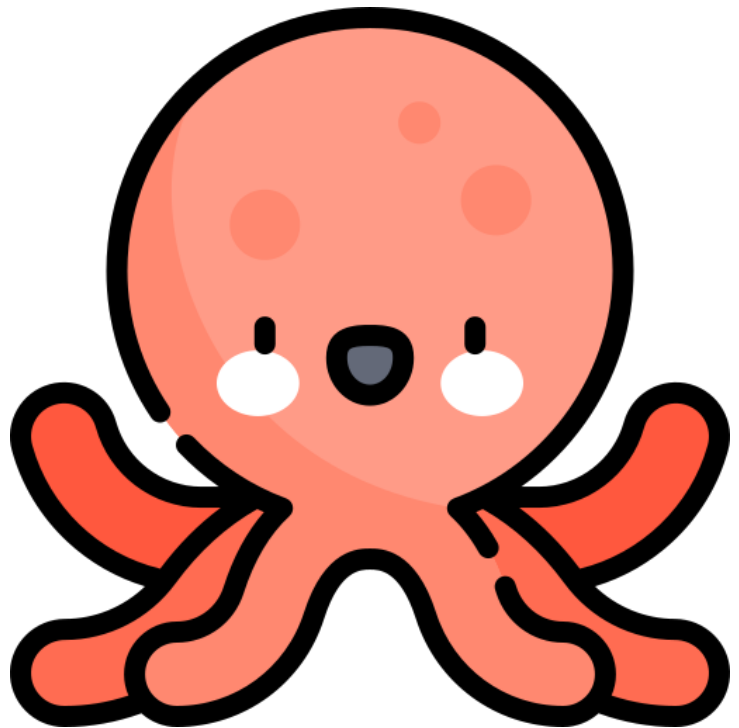


*System
Design Document
Progetto
OctoPlus*



Revision History

Data	Versione	Descrizione	Autore
22/11/2023	0.1	Prima stesura	Tutto il team
25/11/2023	0.2	Aggiunte Referenze a RNF	Donnarumma Salvatore
28/11/2023	0.3	Aggiunti Component e Deployment Diagram	Donnarumma Salvatore
29/11/2023	0.4	Correzioni Matrice secondo le direttive del tutor	Tomeo Orlando
09/12/2023	0.5	Revisione documento	Tomeo Orlando
12/12/2023	0.6	Aggiunta sezione 4	Donnarumma Salvatore
13/12/2023	0.7	Aggiunta sezione crittografia, corretta matrice degli accessi, aggiunto ulteriore boundary condition	Tutto il team
22/01/2024	0.8	Correzioni e aggiornamento sdd	Tutto il team
04/02/2024	0.9	Aggiunto sommario	Orlando Tomeo
08/02/2024	1.0	Revisione documento	Tutto il team

Sommario

1.	Introduzione.....	3
1.1	Scopo del sistema.....	3
1.2	Obiettivi di progettazione.....	3
1.2.1	Trade-off	5
1.4	Riferimenti.....	5
1.5	Overview	5
2.	Architettura del sistema corrente	5
3.	Architettura Proposta	6
3.1	Overview	6
3.2	Scomposizione in sottosistemi.....	6
3.3	Mapping Hardware e Software.....	7
3.4	Gestione della persistenza	7
3.5	Controllo degli accessi e sicurezza	8
3.6	Controllo global del software	8
3.7	Boundary Condition.....	9
4	Servizi dei sottosistemi	10

1. Introduzione

1.1 Scopo del sistema

OctoPlus ha come obiettivo primario di offrire una piattaforma e-commerce per l'acquisto di attrezzatura da scuba-diving e immersioni che sia all'altezza dei competitor esistenti. Il sistema consente agli acquirenti di esplorare una vasta gamma di prodotti, visualizzare dettagli, confrontare prezzi e effettuare transazioni online in modo sicuro.

1.2 Obiettivi di progettazione

Categoria	ID	Descrizione	Reference
Perfomance	OP_1Persistenza	Per garantire la persistenza dei dati relativi a utenti, prodotti e ordini effettuati, è necessario l'impiego di un database relazionale in verranno memorizzate queste informazioni.	RNF3.3.7
Performance	OP_2Capacity	Il sistema deve essere in grado di gestire picchi di utenza e deve essere in grado di lavorare correttamente anche con un afflusso medio-alto di utenti.	RNF3.3.8
Performance	OP_3LoadingTime	Il sistema non deve avere soffrire di caricamenti troppo lunghi per offrire una navigazione fluida e piacevole all'utente. Una pagina deve caricarsi entro un tempo massimo di 5 secondi.	RNF3.3.9
Performance	OP_4ResponseTime	Il sistema deve garantire un tempo di risposta agli input degli utenti di massimo 10 secondi.	RNF3.3.9
Affidabilità	OP_5Robustezza	Il sistema non deve permettere utilizzi pericolosi e impropri.	RNF3.3.10
Affidabilità	OP_6SystemSecurity	Il sistema deve disporre di meccanismi di protezione che non permettono ai comuni utenti di accedere alle aree personali dei gestori.	RNF3.3.6
Affidabilità	OP_7ErrorManagement	Il sistema deve fornire messaggi di errore chiari e suggerimento per aiutare gli utenti a risolvere eventuali	RNF3.3.5

		problemi.	
Affidabilità	OP_8DataSecurity	Il sistema implementa meccanismi di crittografia dei dati sensibili degli utenti per la loro protezione.	RNF3.3.1
End User	OP_9FriendlyUsage	Il sistema offre un'interfaccia intuitiva con struttura di navigazione chiara. Consente agli utenti di trovare rapidamente i prodotti desiderati offrendo una ricerca mirata e filtrata e il carrello è accessibile in qualunque momento.	RNF3.3.2
End User	OP_10Accessibility	Il sistema è supportato dai dispositivi mobile e desktop garantendo la responsività del layout.	RNF3.3.3
End User	OP_11Compatibly	Il sistema deve garantire la compatibilità con diversi browser e sistemi operativi.	RNF3.3.4
Mantenimento	OP_12Three-TierModel	L'architettura del sistema deve essere ispirata al modello Three-Tier per favorire la modularità, manutenibilità e estendibilità. Il sistema permetterà l'aggiunta di nuove funzionalità per mantenere il sito aggiornato e permettere un'esperienza migliore agli utenti.	RNF3.3.11

1.2.1 Trade-off

Trade-off	Razionale
Tempo di distribuzione vs Funzionalità	Per ridurre al minimo i tempi di sviluppo ci limiteremo ad implementare solo le funzionalità a priorità elevata come definito nelle specifiche iniziali ai fini di rientrare nei tempi prestabiliti per il rilascio e con meno bug.
Costo vs Robustezza	Abbiamo deciso di concentrarci sulla robustezza per garantire che il sistema sia in grado di operare in modo affidabile e resistente anche in situazioni impreviste. Questa decisione è stata guidata dal nostro desiderio di offrire un'esperienza utente ottimale e di ridurre al minimo il rischio di malfunzionamenti che potrebbero compromettere la soddisfazione del cliente.

1.3 Definizioni Acronimi e Abbreviazioni

Nel documento vengono utilizzate i seguenti acronimi e abbreviazioni:

- OP_X: obiettivo di progettazione numero X
- RNF X: requisito non funzionale numero X
- DB: database

1.4 Riferimenti

- Problem Statement (PS)
- Requirements Analysis Document (RAD)
- Testing Plan (TP)
- Test Case Specification (TCS)

1.5 Overview

Il documento prevede le seguenti sezioni:

- **Introduzione:** Presentazione generale dello scopo del sistema e degli obiettivi di design che si intende rispettare.
- **Architettura del sistema corrente:** viene descritto lo stato attuale dell'architettura dei sistemi già sul mercato.
- **Architettura proposta:** presenta una overview generale di come il sistema sarà strutturato e scomposto in sottosistemi; a seguire vi è il mapping Hardware/Software rispetto ai sottosistemi precedentemente definiti, e una descrizione di come i dati persistenti saranno gestiti. Infine vi è la definizione e le motivazioni del flusso di controllo che caratterizza il sistema, oltre che gli use case legali alle boundary conditions.

2. Architettura del sistema corrente

Al momento della scrittura di questo documento non sono pochi i siti e-commerce che, come OctoPlus, sono dedicati alla vendita di attrezzature per snorkeling e scuba diving. È possibile dunque fare una comparazione diretta con molti e-commerce già presenti sul mercato che probabilmente utilizzeranno un'architettura Three-tier, scelta molto gettata grazie ai vantaggi che offre.

3. Architettura Proposta

3.1 Overview

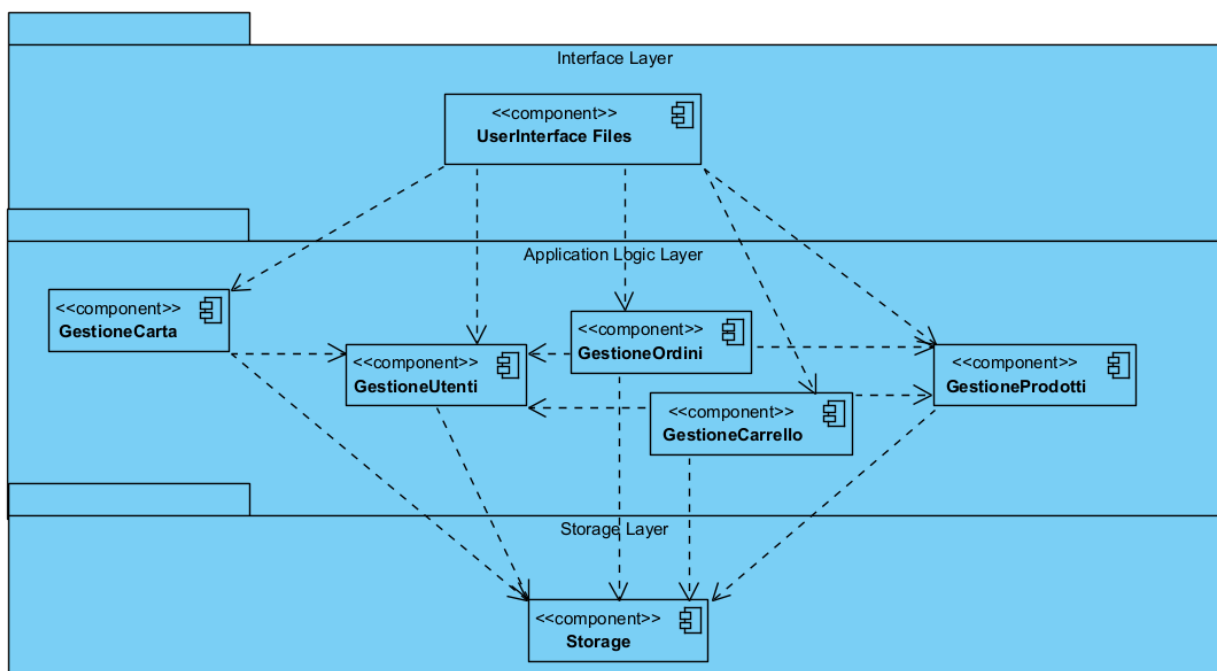
Il sistema si basa sul modello Three-Tier, presentato come una architettura suddivisa in tre layer principali inerenti a data, presentation e application logic. Il modello THREE-TIER è dunque una architettura che organizza il sistema in 3 layers principali: interface layer che include oggetti boundary che cui l'utente interagisce, application logic layer che include oggetti control e entity, infine storage layer che si occupa della gestione dei dati persistenti. L'utilizzo di un'architettura three-tier in un sito e-commerce fornisce vantaggi chiave come la separazione chiara delle responsabilità, la scalabilità efficace, la manutenibilità semplificata, la riusabilità del codice, la flessibilità tecnologica, una migliore gestione degli errori, la sicurezza migliorata, la facilità di integrazione e l'adattabilità a futuri sviluppi.

3.2 Scomposizione in sottosistemi

Di seguito tutti i sottosistemi individuati con relative funzionalità offerte:

- **UserInterfaceFiles**: sottosistema che si occupa della gestione delle interfacce utente. Rappresentano le componenti che permettono l'interazione degli utenti con il sistema.
- **GestioneUtenti**: fornisce le funzionalità di registrazione e autenticazione di un Utente/Gestore e modifica dei dati personali di quest'ultimi.
- **GestioneProdotti**: fornisce le funzionalità per la gestione dei prodotti del catalogo.
- **GestioneOrdini**: fornisce le funzionalità per la gestione degli ordini.
- **GestioneCarrello**: fornisce le funzionalità per la gestione del carrello (persistenza sul db).
- **GestioneCarta**: fornisce le funzionalità per la gestione della carta come per il pagamento (modifica dati, inserimento dati e persistenza).
- **Storage**: sottosistema che si occupa della gestione e accesso ai dati persistenti tramite l'uso di un database.

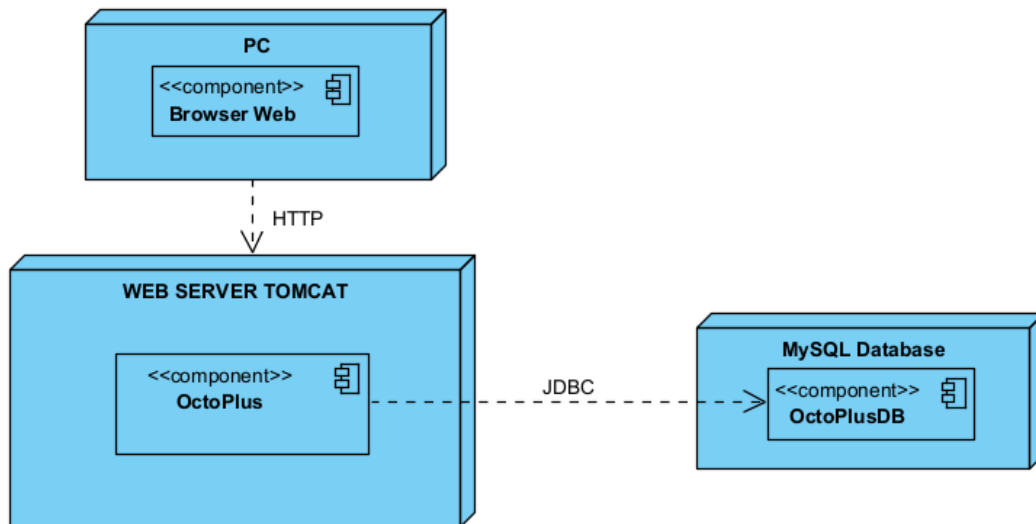
A seguire l'UML Component Diagram:



3.3 Mapping Hardware e Software

Il sistema che sarà sviluppato, trattandosi di un applicativo web, possiamo rappresentarlo attraverso tre nodi:

- **PC:** dove sarà installato un web browser con il quale sarà possibile inoltrare richieste http al server e su cui sarà installato l'applicativo.
- **WebServer Tomcat:** dove avverrà il deployment dell'applicativo. Qui il sistema interagisce con un DBMS mediante l'utilizzo di driver JDBC.
- **MySQL Database:** ambiente che ospiterà il database del sistema e si occuperà della sua gestione.



3.4 Gestione della persistenza

La gestione dei dati persistenti viene effettuata tramite un Database MySQL. I dati da rendere persistenti sono:

- Informazioni relative agli utenti che si registrano al sito [nome, cognome, email, password, numero telefono, tipo di utente] ed eventualmente il carrello, una volta registrati alla piattaforma.
- Informazioni relative ai prodotti che sono presentati nel catalogo [nome, categoria, prezzo, descrizione, foto, statistiche].
- Informazioni relative agli ordini effettuati dagli utenti [utente che ha effettuato l'ordine, data, stato, prezzo totale, indirizzo, data acquisto, data consegna]
- Informazioni relative alla carta usata come pagamento [proprietario, data_scadenza, numero_carta]

3.5 Controllo degli accessi e sicurezza

Oggetti Attori	Utente	Prodotto	Carrello	Ordine	Catalogo	Carta
Utente	registrazione login cambioPassword visualizzaProfilo logout	acquistoProdotto visualizzaProdotto	visualizzaCarrello eliminaProdotto aggiungiProdotto aumentareQntProd diminuireQntProd acquistoDaCarrello	visualizzaOrdine	visualizzaCatalogo filtraPrezzo filtraCategoria ricercaProdotto	salvaCarta modificaCarta cancellaCarta
Gestore Utenti	login cambioPasswordGestore registraGestore cancellaGestore visualizzaProfilo logout					
Gestore Ordini	login visualizzaProfilo logout			rimozioneOrdine ricercaOrdine cambioStatoOrdine filtroDataOrdine visualizzaOrdini		
Gestore Prodotti	login visualizzaProfilo logout	modificaProdotto visualizzaProdotto creaProdotto cancellaProdotto			visualizzaCatalogo ordinaProdotti aggiungiProdotto rimuoviProdotto	

Si definisce autenticazione il processo che consiste nell'associazione tra l'identità di un utente ad il sistema. Nel nostro caso, il processo di autenticazione consiste nell'inserimento da parte dell'utente di una email, che funge da username, e di una password, memorizzata nel sistema. Con questo meccanismo di autenticazione, possiamo assumere che un utente è legittimo se è a conoscenza della combinazione email-password. Per rafforzare la sicurezza degli accessi, prevenendo accessi non autorizzati, viene introdotto un meccanismo di crittografia. Tale sistema viene introdotto tramite l'utilizzo dell'algoritmo SHA-512 che consiste nel processamento di una stringa in un'altra stringa di lunghezza variabile. La lunghezza dipenderà dalla complessità e numero di caratteri utilizzati per la password.

3.6 Controllo global del software

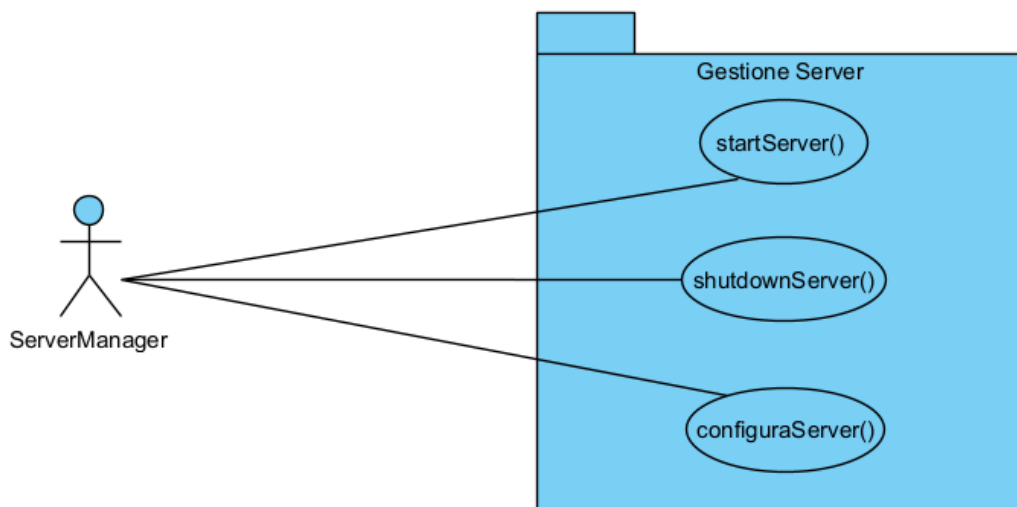
Il nostro sistema avrà un controllo del flusso globale di tipo Event-Driven. Tutte le richieste http verranno gestite da un Web Server che fungerà da sottosistema dispatcher: quest'ultimo si occuperà del delivery delle richieste verso le apposite Servlet che si occuperanno dell'elaborazione di response http ed eventualmente di eseguire il codice di business dell'applicazione. Si utilizza dunque un Design centralizzato dove sono presenti diversi control objects (le Servlets) che controllano il flusso. Grazie a questa implementazione i cambiamenti alla struttura di controllo sono facili da apportare, tuttavia c'è lo svantaggio che il singolo control object può diventare un collo di bottiglia per le performance dell'intero sistema.

3.7 Boundary Condition

Boundary condition analizzate:

- **Inizializzazione (start-up):** il sistema viene avviato rendendo operativi il Server Tomcat e il server MySQL. Viene successivamente stabilita una connessione al database mediante driver JDBC. Quest'ultimo verifica la sanità dei dati persistenti. Se tutto il processo è andato a buon fine, il sistema rende infine disponibili le sue funzionalità e servizi agli utenti.
- **Terminazione (shutdown):** viene data la possibilità di spegnere completamente il sistema per permettere le operazioni di manutenzione. Il sistema verifica prima dello spegnimento che non ci siano connessioni aperte verso l'esterno; in caso affermativo termina l'esecuzione dell'applicativo e verrà chiusa la connessione aperta. L'orario designato per permettere la manutenzione sarà una fascia orario in cui si registreranno il minor numero di accessi al sito.
- **Failure (gestione delle eccezioni):** in caso di failure del sistema si potrebbe pensare ad introdurre meccanismi di ripristino automatico, ad esempio se l'eccezione è causata da problemi di connessioni col database, il sistema potrebbe tentare automaticamente di ristabilire la connessione. Si prevedono pagine di errore utente chiare ma che evitano di fornire dettagli tecnici sensibili agli utenti. In casi estremi il sistema verrà reso offline (shutdown) finché non verranno risolti i problemi dal team dei developer.

User task per le boundary condition:



4 Servizi dei sottosistemi

GestioneUtenti

Servizio	Descrizione	Interfaccia
Salva un utente sul DB	Servizio che permette di salvare un nuovo utente nel DB.	IUserDao
Autenticazione di un utente	Servizio che permette di verificare se un utente che si sta autenticando è già registrato al sito e le sue credenziali sono già presenti nel DB.	IUserDao
Ottieni elenco utenti salvati nel DB	Servizio che permette di ottenere l'elenco degli utenti presenti nel DB.	IUserDao
Cambio password di un utente	Servizio che permette di cambiare la password associata ad un utente salvata sul DB.	IUserDao
Controllo vecchia password	Servizio che permette di verificare se una password inserita corrisponde a una password già registrata sul DB.	IUserDao
Ottieni un utente salvato nel DB.	Servizio che permette di ottenere un utente salvato nel DB.	IUserDao
Salva un gestore sul DB.	Servizio che permette di salvare un nuovo gestore nel DB.	IUserDao
Cancella un gestore nel DB.	Servizio che permette di cancellare un Gestore presente nel DB.	IUserDao
Cambio password di un Gestore	Servizio che permette di cambiare la password associata ad un Gestore salvato sul DB.	IUserDao

GestioneProdotti

Servizio	Descrizione	Interfaccia
Salva un prodotto sul DB	Servizio che permette di salvare un nuovo prodotto nel DB.	IProductDao
Cancella un prodotto nel DB	Servizio che permette di cancellare un prodotto nel DB.	IProductDao
Ottieni elenco di tutti i prodotti salvati nel DB	Servizio che permette di ottenere l'elenco dei prodotti presenti nel DB.	IProductDao
Ottieni un prodotto salvato nel DB.	Servizio che permette di ottenere un prodotto salvato nel DB.	IProductDao
Modifica un prodotto presente nel DB	Servizio che permette di modificare un prodotto presente nel DB.	IProductDao
Ottieni elenco filtrato di prodotti	Servizio che permette di ottenere un elenco filtrato di prodotti salvati nel DB.	IProductDao
Carica foto di un prodotto nel DB.	Servizio che permette di salvare la foto/immagine relativa a uno specifico prodotto nel DB.	IPhotoDao
Ottieni foto di un prodotto salvato sul DB.	Servizio che permette di ottenere al foto/immagine relativa a uno specifico prodotto salvato nel DB.	IPhotoDao

GestioneOrdini

Servizio	Descrizione	Interfaccia
Salva un ordine sul DB	Servizio che permette di salvare un nuovo ordine nel DB.	IOrderDao
Cancella un ordine nel DB	Servizio che permette di cancellare un ordine nel DB.	IOrderDao
Ottieni elenco di tutti gli ordini salvati nel DB	Servizio che permette di ottenere l'elenco degli ordini presenti nel DB.	IOrderDao
Ottieni elenco ordini relativi a uno specifico utente.	Servizio che permette di ottenere un elenco di ordini relativi a uno specifico utente, entrambi presenti nel DB.	IOrderDao
Ottieni elenco prodotti ordinati relativi a uno specifico ordine.	Servizio che permette di ottenere l'elenco dei prodotti ordinati di uno specifico ordine.	IOrderDao
Ottieni un ordine salvato nel DB.	Servizio che permette di ottenere un ordine salvato nel DB.	IOrderDao

GestioneCarrello

Servizio	Descrizione	Interfaccia
Salva il carrello nel DB	Servizio che permette di salvare il carrello di uno specifico utente nel DB.	ICarrelloDao
Cancella il carrello nel DB	Servizio che permette di cancellare un carrello nel DB.	ICarrelloDao
Recupera Carrello dal DB	Servizio che permette di recuperare l'elenco dei prodotti salvati nel carrello di uno specifico utente	ICarrelloDao

GestioneCarta

Servizio	Descrizione	Interfaccia
Salva i dati della carta nel DB	Servizio che permette di salvare i dati di una carta di specifico utente nel DB.	ICartaDao
Cancella i dati della carta nel DB	Servizio che permette di cancellare i dati di una carta nel DB.	ICartaDao
Recupera dati di pagamento di un utente dal DB	Servizio che permette di recuperare i dati di pagamento di uno specifico utente	ICartaDao