



# UNIVERSITÀ DEGLI STUDI DI SALERNO

**Corso di Ingegneria del Software  
System Design Document  
(SDD)**

Quattro<sup>63</sup>hi.it

**Data: 27/12/2017**

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## Coordinatori del progetto:

Nome
De Lucia Andrea
Francese Rita

## Partecipanti:

Nome	Matricola
Palmiero Alberto	0512103454
Piccolo Luigi	0512103964
Reccia Luca	0512103736
Volpe Nicola	0512103556

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

# Sommario

<b>1. INTRODUZIONE .....</b>	<b>4</b>
1.1. OBIETTIVI DEL SISTEMA .....	4
<b>2. DESIGN GOALS .....</b>	<b>5</b>
2.1. CRITERI DI PERFORMANCE .....	5
2.2. CRITERI DI AFFIDABILITÀ .....	5
2.3. MANUTENZIONE.....	5
2.4. CRITERI DI USABILITÀ.....	5
<b>3. DEFINIZIONI, ACRONIMI E ABBREVIAZIONI.....</b>	<b>6</b>
<b>4. RIFERIMENTI .....</b>	<b>6</b>
<b>5. ARCHITETTURA DEL SISTEMA E DEI SOTTOSISTEMI.....</b>	<b>7</b>
5.1. ARCHITETTURA DEL SISTEMA PROPOSTO .....	7
5.1.1. <i>Panoramica</i> .....	7
5.1.2. <i>Decomposizione in Sottosistemi</i> .....	8
5.1.3. <i>Diagramma di Deployment</i> .....	9
5.1.4. <i>Mapping Hardware / Software</i> .....	9
<b>6. GESTIONE DEI DATI PERSISTENTI .....</b>	<b>10</b>
<b>7. GESTIONE E CONTROLLO DELLA SICUREZZA.....</b>	<b>11</b>
<b>8. BOUNDARY CONDITION .....</b>	<b>12</b>
<b>9. SERVIZI DEI SOTTOSISTEMI.....</b>	<b>13</b>
9.1. GESTIONE ORDINI.....	13
9.2. GESTIONE ARTICOLI .....	14
9.3. GESTIONE UTENTI .....	15
<b>10. RILASCIO DEL SISTEMA .....</b>	<b>16</b>

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

# 1. Introduzione

## 1.1. Obiettivi del sistema

Il sistema che si intende realizzare è un sito di commercio elettronico per occhiali da sole, "Quattrocchi.it".

Un paio di **occhiali da sole** è caratterizzato dalla marca, il modello, descrizione, prezzo e tre foto.

Un **cliente** può registrarsi al sito indicando nome, cognome, mail, data di nascita, e inserendo un'opportuna username che lo identifichi univocamente all'interno del sistema, nonché una password di accesso.

Un **cliente registrato** al sito può cercare all'interno del catalogo un prodotto tramite l'inserimento di una singola stringa di testo (Ricerca semplice) o di più parametri (Ricerca avanzata). Dalla pagina specifica dell'articolo è possibile aggiungerlo al **carrello**. Le modifiche apportate al carrello sono immediatamente rese disponibili. Procedendo al checkout, il carrello viene svuotato e il cliente, dopo aver opportunamente indicato il metodo di pagamento e l'indirizzo di spedizione, acquista il prodotto.

Nella pagina profilo del cliente è possibile controllare il proprio storico ordini.

Sul lato amministrativo del sito sono identificati tre tipi di ruoli: **amministratore**, **gestore catalogo** e **gestore ordini**. Ognuno effettua l'accesso dalla pagina login del sito.

L'**amministratore**, le cui credenziali di accesso sono salvate sul database, è colui che aggiunge o rimuove la login di un gestore. È inoltre l'unico che può visualizzare i clienti registrati al sito e, se necessario, disabilitarne l'accesso.

Il **gestore catalogo** ha il compito di aggiungere o rimuovere prodotti dal catalogo, e aggiornare le informazioni di un prodotto.

Il **gestore ordini** coordina e controlla gli articoli da spedire, gli articoli spediti non ancora pervenuti, gli articoli giunti a destinazione.

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## 2. Design Goals

### 2.1. Criteri di performance

#### **Tempi di risposta.**

- Per la visualizzazione del catalogo il tempo di risposta è di massimo 3 secondi.
- Per la visualizzazione del profilo utente è di 2 secondi.
- Per il processing dell'ordine, massimo di 5 secondi.

### 2.2. Criteri di affidabilità

#### **Affidabilità.**

- I dati sensibili dell'utente non sono visibili se non dall'amministratore del sito.

#### **Robustezza.**

- Eventuali input non validi immessi dall'utente saranno opportunamente segnalati attraverso messaggi di errore.

### 2.3. Manutenzione

#### **Supportabilità.**

- Il sistema richiede capacità di adattamento nel caso in cui si decida di trasformarlo in un e-commerce general purpose.

### 2.4. Criteri di usabilità

#### **Leggibilità.**

- Il sistema userà font sans-serif di grandezza non inferiore ai 15px.

#### **Navigazione.**

- Ogni sezione del sito è raggiungibile in non più di 4 click.

#### **Look&Feel.**

- L'interfaccia deve adattarsi a qualsiasi dispositivo che acceda al sito.

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

### 3. Definizioni, acronimi e abbreviazioni

**RAD:** Requirements Analysis Document.

**SDD:** System Design Document.

**ODD:** Object Design Document.

**DB:** Database.

**MVC:** Model Control View.

**Gestore catalogo:** utente del sito che gestisce gli articoli presenti nel catalogo.

**Gestore ordini:** utente del sito che gestisce la spedizione degli ordini effettuati.

**Amministratore:** amministratore del sito. Gestisce tutte le login registrate.

**Greenfield engineering:** Tipologia di sviluppo che comincia da zero, non esiste nessun sistema a priori e i requisiti sono ottenuti dall'utente finale e dal cliente. Nasce, perciò, a partire dai bisogni dell'utente.

**Quattrocchi.it:** il sito di e-commerce di occhiali da sole che si intende sviluppare.

### 4. Riferimenti

- Bernd Bruegge & Allen H. Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns and Java*, (2nd edition), Prentice-Hall, 2003.
- <https://www.amazon.it/>
- <https://www.ray-ban.com>

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## 5. Architettura del sistema e dei sottosistemi

Di seguito verrà presentata l'architettura del sistema proposto in cui gestiremo la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza. Verranno, inoltre, presentati in maggiore dettaglio i servizi dei sottosistemi.

### 5.1. Architettura del sistema proposto

#### 5.1.1. Panoramica

Il sistema proposto è un'applicazione web, distribuita secondo il modello client server.

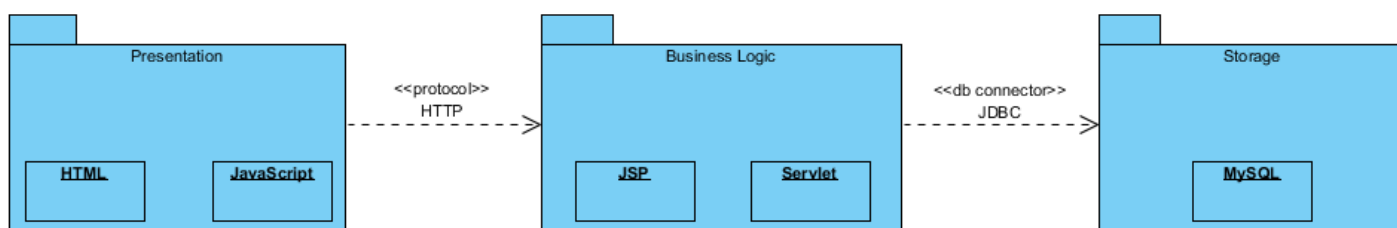
L'obiettivo che si pone è quello di

- permettere a un potenziale cliente del sito di acquistare un paio di occhiali da sole con estrema semplicità.
- snellire il più possibile il lato gestionale della piattaforma.

Il client gestirà la parte di presentazione e la parte di logica direttamente connessa all'interfaccia grafica, il server invece gestirà la parte di logica relativa ai dati e i dati stessi che saranno salvati in un database. L'intenzione è quella di rendere il sistema manutenibile e di permettere facilmente l'aggiunta di nuove funzionalità qualora se ne presentasse la necessità.

I gestori saranno individuati in base alle funzionalità per poter rendere massima la coesione e minimo l'accoppiamento tra i sottosistemi in modo che i cambiamenti in un sottosistema non influiscano sugli altri.

Come decomposizione in sottosistemi utilizzeremo l'architettura three tier layer.



Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

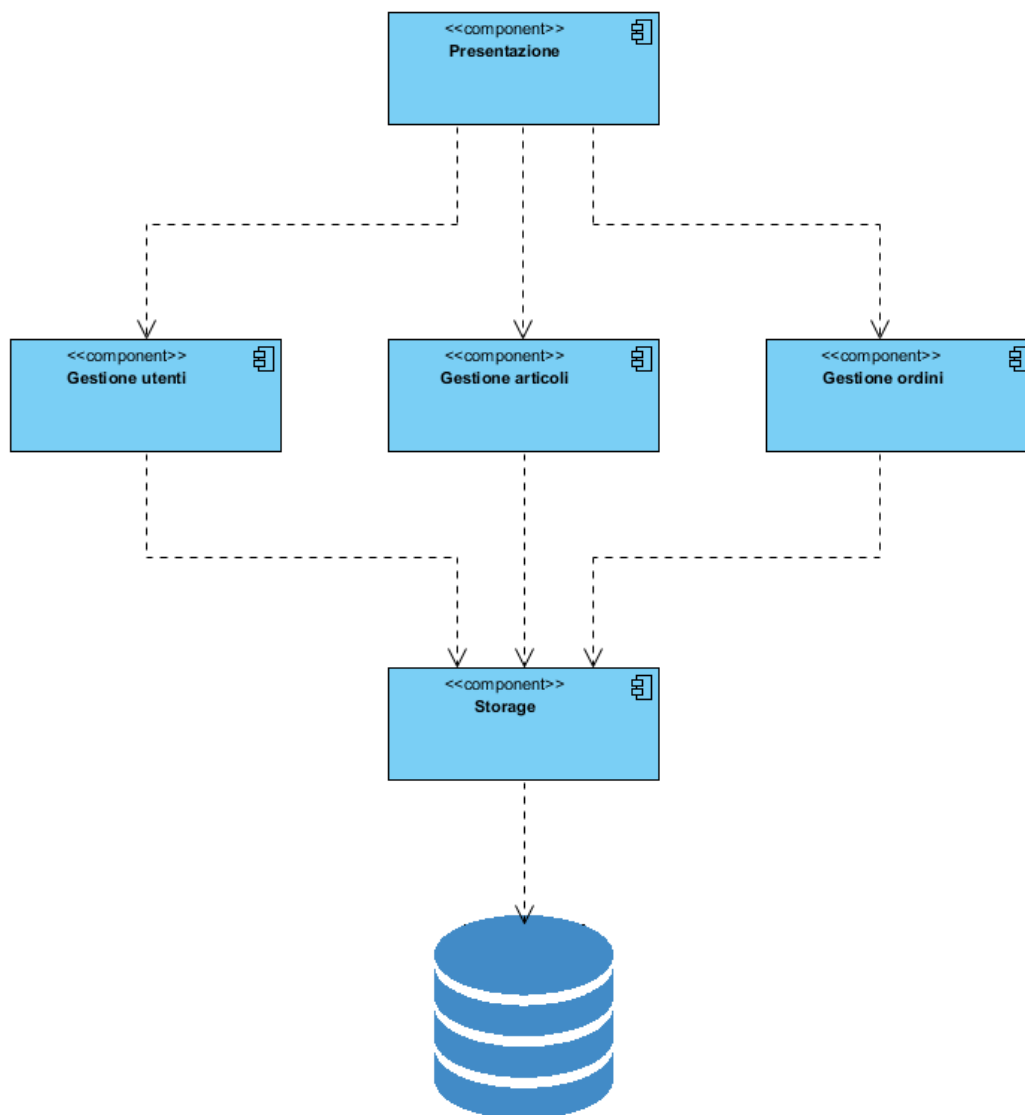
### 5.1.2. Decomposizione in Sottosistemi

Dopo un'attenta analisi abbiamo deciso di suddividere il sistema nel seguente modo, con l'obiettivo di minimizzare l'accoppiamento ed avere un'alta coesione.

Il sistema si compone nei seguenti 5 sottosistemi:

- View
- Gestione utenti
- Gestione articoli
- Gestione ordini
- Model

Il livello Control comprende una divisione in 3 sottosistemi: Gestione articoli, ordini, utenti.





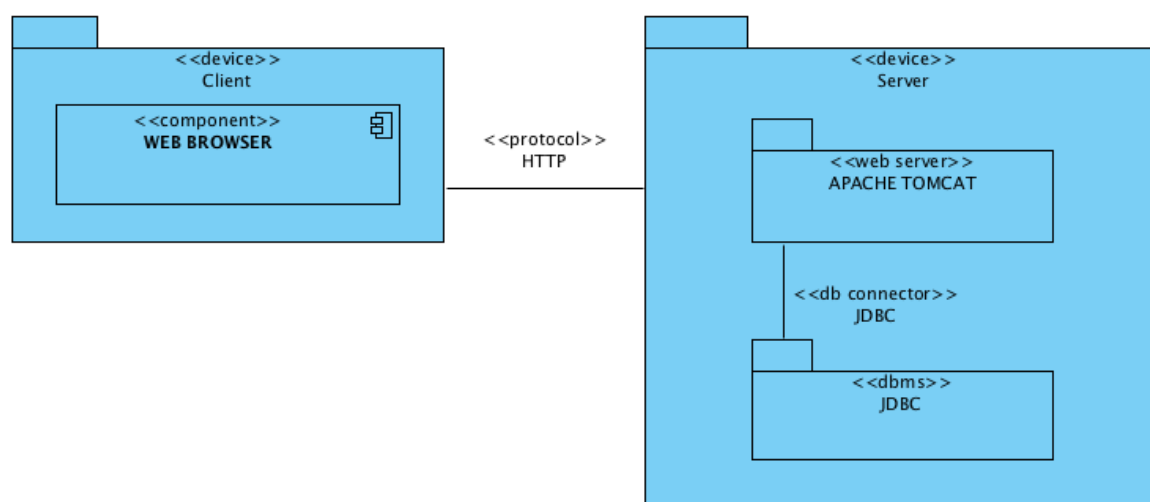
Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

### 5.1.3. Diagramma di Deployment

L'utente richiede le funzionalità tramite un'interfaccia web. Questa costituirà, insieme ad un minimo di logica di business che consiste nel controllo della validità dei dati che l'utente può inserire, tutta la parte che verrà eseguita sul client. Per la parte server includerà la logica di business e la gestione della persistenza.

Sia il client che il server per il corretto funzionamento del sistema devono avere la possibilità di connettersi ad internet.

### 5.1.4. Mapping Hardware / Software



Il sistema che si vuole realizzare sarà installato su una macchina che fungerà da server. Questa dovrà includere un web server, in particolare Apache Tomcat ed un DBMS, useremo MySQL. Il client potrà essere un qualsiasi computer con un browser che supporta Javascript ed una connessione ad Internet.

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## 6. Gestione dei dati persistenti

Nella gestione dei dati persistenti, si è convenuto salvare tutto all'interno di un database MySQL. Il vantaggio di usare un database è dovuto prima di tutto dal fatto che i dati sono logicamente organizzati, nonché dalla maggiore sicurezza garantita nel recupero di dati sensibili.

Il database sarà per la prima implementazione locale, ma se eventualmente si decidesse di renderlo remoto, ciò non costituirebbe un problema. Infatti, tramite l'uso del driver JDBC, il business layer scritto in Java sarà in grado di comunicare con il database, qualunque sia la sua locazione fisica.

La progettazione del database è stata effettuata in un documento esterno chiamato "SCHEMA E-R". Sostanzialmente si tratta della progettazione iniziale dello schema e-r, seguito da una ristrutturazione dove viene specificata solo quella parte di progetto che verrà effettivamente resa pubblica in un primo rilascio.

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## 7. Gestione e controllo della sicurezza

Attori	Sottosistemi	Gestione		
		Ordini	Articoli	Utenti
Gestore ordini		<ul style="list-style-type: none"> <li>Visualizza ordini</li> <li>Inserisci dati di spedizione e gestione dell'ordine</li> </ul>		<ul style="list-style-type: none"> <li>Login</li> </ul>
Gestore catalogo			<ul style="list-style-type: none"> <li>Modifica prodotto</li> <li>Aggiungi prodotto</li> <li>Elimina prodotto</li> </ul>	<ul style="list-style-type: none"> <li>Login</li> </ul>
Amministratore				<ul style="list-style-type: none"> <li>Login</li> <li>Aggiungi login gestore</li> <li>Rimuovi login</li> </ul>
Cliente		<ul style="list-style-type: none"> <li>Aggiungi/Rimuovi prodotto al carrello</li> <li>Modifica quantità prodotto in carrello</li> <li>Visualizza carrello</li> </ul>	<ul style="list-style-type: none"> <li>Ricerca prodotto</li> <li>Ricerca prodotto avanzata</li> <li>Visualizza catalogo</li> <li>Visualizza prodotto</li> </ul>	
Cliente non registrato				<ul style="list-style-type: none"> <li>Effettua registrazione</li> </ul>
Cliente registrato		<ul style="list-style-type: none"> <li>Visualizza storico ordini</li> <li>Checkout</li> </ul>	<ul style="list-style-type: none"> <li>Ricerca prodotto</li> <li>Ricerca prodotto avanzata</li> <li>Visualizza catalogo</li> <li>Visualizza prodotto</li> </ul>	<ul style="list-style-type: none"> <li>Login</li> <li>Inserisci/Rimuovi metodo di pagamento</li> <li>Inserisci/Rimuovi indirizzo di spedizione</li> <li>Visualizza profilo</li> </ul>

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## 8. Boundary condition

<b>Use case name</b>	<b><u>Startup del sistema</u></b>					
<b>Participating actors</b>	Amministratore					
<b>Entry conditions</b>	<ul style="list-style-type: none"><li>• L'amministratore è connesso tramite una shell al server che ospita il web server</li></ul>					
<b>Flow events</b>	<table><tr><td>1. L'amministratore scrive in console <code>"/startup.sh"</code></td><td></td></tr><tr><td></td><td>2. Il sistema inizia l'esecuzione creando manager inizializzandoli.</td></tr></table>		1. L'amministratore scrive in console <code>"/startup.sh"</code>			2. Il sistema inizia l'esecuzione creando manager inizializzandoli.
1. L'amministratore scrive in console <code>"/startup.sh"</code>						
	2. Il sistema inizia l'esecuzione creando manager inizializzandoli.					
<b>Exit conditions</b>	<ul style="list-style-type: none"><li>• Il sito è raggiungibile da qualsiasi web browser connesso alla rete.</li></ul>					

<b>Use case name</b>	<b><u>Shutdown del sistema</u></b>							
<b>Participating actors</b>	Amministratore							
<b>Entry conditions</b>	<ul style="list-style-type: none"><li>• L'amministratore è loggato in una qualsiasi pagina.</li><li>• L'amministratore ha cliccato sul pulsante di shutdown</li></ul>							
<b>Flow events</b>	<table><tr><td>1. L'amministratore clicca su ok per confermare l'azione</td><td></td></tr><tr><td></td><td>2. Il sistema notifica tutti gli utenti del sito che lo shutdown avverrà in 5 minuti.</td></tr><tr><td></td><td>3. Il sistema esegue lo script di shutdown di Tomcat.</td></tr></table>		1. L'amministratore clicca su ok per confermare l'azione			2. Il sistema notifica tutti gli utenti del sito che lo shutdown avverrà in 5 minuti.		3. Il sistema esegue lo script di shutdown di Tomcat.
1. L'amministratore clicca su ok per confermare l'azione								
	2. Il sistema notifica tutti gli utenti del sito che lo shutdown avverrà in 5 minuti.							
	3. Il sistema esegue lo script di shutdown di Tomcat.							
<b>Exit conditions</b>	<ul style="list-style-type: none"><li>• Il sito non è più raggiungibile.</li></ul>							

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## 9. Servizi dei sottosistemi

### 9.1. Gestione ordini

Sottosistema	Descrizione
<b>Gestione Ordini.</b>	Sottosistema che permette la gestione degli ordini e le funzionalità del carrello.

Servizio	Descrizione
<b>inserisciDatiDiSpedizione</b>	Servizio che permette al gestore degli ordini di inserire numero di tracking, corriere e data consegna ad un pacco in stato "da spedire". Questo servizio permette anche di reimpostare i dati di spedizione e lo stato dell'ordine nel caso in cui il gestore degli ordini abbia commesso qualche errore.
<b>aggiungiProdottoAlCarrello</b>	Servizio che permette l'aggiunta di un prodotto al carrello. Se il prodotto è già presente, viene incrementata la sua quantità.
<b>rimuoviProdottoDalCarrello</b>	Servizio che permette la rimozione di un prodotto dal carrello.
<b>modificaQuantitàInCarrello</b>	Servizio che permette di aggiornare la quantità dei prodotti inseriti nel carrello.
<b>visualizzaStoricoOrdini</b>	Servizio che permette a un cliente registrato di visualizzare la lista degli ordini emessi.
<b>visualizzaOrdini</b>	Servizio che permette a un gestore degli ordini di visualizzare la lista di tutti gli ordini.
<b>visualizzaCarrello</b>	Servizio che permette ad un cliente di visualizzare i prodotti inseriti nel carrello e il prezzo totale.
<b>checkout</b>	Servizio che permette ad un cliente registrato di completare l'acquisto dei prodotti che si trovano all'interno del carrello.

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## 9.2. Gestione articoli

Sottosistema	Descrizione
<b>Gestione Articoli.</b>	Sottosistema che permette la modifica e l'aggiunta di prodotti nel catalogo da parte del gestore catalogo, la ricerca dei prodotti nel catalogo.

Servizio	Descrizione
<b>ricercaProdotto</b>	Servizio che permette al cliente di ricercare dei prodotti nel catalogo per via di una parola chiave.
<b>ricercaProdottoAvanzata</b>	Servizio che permette al cliente di ricercare dei prodotti nel catalogo secondo vari parametri.
<b>visualizzaCatalogo</b>	Servizio che permette al cliente di visualizzare i prodotti nel catalogo.
<b>visualizzaProdotto</b>	Servizio che permette al cliente di visualizzare le informazioni inerenti ad un prodotto in catalogo.
<b>aggiungiProdotto</b>	Servizio che permette al gestore catalogo di aggiungere un nuovo prodotto al catalogo.
<b>modificaProdotto</b>	Servizio che permette al gestore catalogo di modificare alcuni parametri di un determinato prodotto.
<b>eliminaProdotto</b>	Servizio che permette al gestore catalogo di eliminare un prodotto dal catalogo.

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

### 9.3. Gestione utenti

Sottosistema	Descrizione
<b>Gestione utenti.</b>	Sottosistema che permette ad un utente di effettuare la login, ad un cliente non registrato di effettuare la registrazione, ad un cliente registrato di visualizzare il proprio profilo e gestire i dati personali, ad un amministratore di creare login gestori e rimuovere login.

Servizio	Descrizione
<b>Login</b>	Servizio che permette ad un cliente già registrato di effettuare l'accesso nel sistema.
<b>Registrazione</b>	Servizio che permette ad un cliente non registrato di creare una login per poter accedere nel sistema.
<b>visualizzaProfilo</b>	Servizio che permette ad un cliente registrato di visualizzare le informazioni inerenti al proprio login.
<b>aggiungiCartaCredito</b>	Servizio che permette ad un cliente registrato di aggiungere un nuovo metodo di pagamento.
<b>rimuoviCartaCredito</b>	Servizio che permette ad un cliente registrato di rimuovere una carta di credito inserita in precedenza.
<b>aggiungiIndirizzoDiSpedizione</b>	Servizio che permette ad un cliente registrato di aggiungere un nuovo indirizzo di spedizione.
<b>rimuoviIndirizzoDiSpedizione</b>	Servizio che permette ad un cliente registrato di rimuovere un indirizzo di spedizione inserito in precedenza.
<b>addLoginGestore</b>	Servizio che permette all'amministratore di creare login per i gestori.
<b>rimuoviLogin</b>	Servizio che permette all'amministratore di rimuovere login.

Progetto: Quattrocchi.it	Versione: 1.1
Documento: System Design Document	Data: 27/12/2017

## 10. Rilascio del sistema

L'approccio utilizzato nell'implementazione del software è di tipo incrementale. Ovvero saranno aggiunte funzionalità in ordine di priorità. L'ordine sarà il seguente:

1. Gestione ordini, accesso al sito e lato acquisti.
2. Gestione catalogo.
3. Pagina amministratore.