



## SDD

# System Design Document

Versione	1.0
Data	10/12/2019
Destinatario	Prof. Andrea De Lucia
Presentato da	Riccardo Martiniello Davide Cresci Alessio Rizzolo Giuseppe Caiazzo

## Revision History

data	versione	descrizione	autori
30/11/2019	1.0	Introduzione e architettura del sistema corrente	Tutti i membri
3/12/2019	1.0	Mapping hardware-software	Tutti i membri
5/12/2019	1.0	Panoramica e decomposizione in sottosistemi	Tutti i membri
6/12/2019	1.0	Gestione persistente dei dati	Tutti i membri
6/12/2019	1.0	Controllo accessi e sicurezza	Tutti i membri



## Sommario

1. Introduzione .....	4
1.1 Obiettivo del sistema.....	4
1.2 Design goal .....	4
1.3 Riferimenti.....	5
1.4 Organizzazione del contenuto.....	5
2. Architettura del sistema corrente .....	5
3. Architettura del sistema proposto .....	5
3.1 Panoramica.....	5
3.2 Decomposizione in sottosistemi .....	6
3.2.1 Decomposizione in layer.....	6
3.2.2 Decomposizione in sottosistemi.....	6
3.3 Mapping Hardware-Software.....	8
3.4 Gestione persistente dei dati .....	9
3.5 Controlli accesso e sicurezza .....	12
3.6 Controllo globale del sistema.....	14
3.7 Boundary Conditions .....	14
4. Glossario .....	15

# 1. Introduzione

## 1.1 Obiettivo del sistema

L'obiettivo del sistema è quello di realizzare un e-commerce per la gestione di un negozio online specializzato nella vendita di capi di abbigliamento. Dress-Store è un e-commerce semplice e facile da utilizzare con tante funzionalità per migliorare sempre di più l'esperienza di coloro che decidono di visitarlo. Il sito, in particolare, offrirà diverse funzionalità quali: permettere ad un addetto di monitorare lo stato del magazzino che si occuperà dell'approvvigionamento dello stesso, inserimento di offerte da parte di un gestore e la possibilità di effettuare controlli statistici sui prodotti venduti.

Il sito è rivolto a persone di ogni età. Il nostro obiettivo è garantire semplicità a coloro che hanno intenzione di acquistare prodotti in maniera rapida e senza molte pretese e allo stesso tempo offrire un'ampia scelta a chi cerca prodotti più specifici.

## 1.2 Design goal

<b>PRIORITA'</b>	<b>DESIGN GOAL</b>	<b>DESCRIZIONE</b>
<b>Media</b>	<b>Usabilità</b>	L'utente deve essere in grado di muoversi all'interno del negozio di e-commerce; questo è possibile grazie all'utilizzo di menù contestuali che consentono di spostarsi agevolmente e di comprendere il funzionamento delle varie attività offerte.
<b>Alta</b>	<b>Affidabilità</b>	Il sito deve garantire all'utente una certa sicurezza in caso di inserimenti di input non validi, in questo modo la navigazione risulta più immediata e sicura.
<b>Alta</b>	<b>Performance</b>	Il tempo di risposta del sito deve essere rapido. Se il sito deve essere sottoposto a manutenzione, l'utente deve essere avvisato con 24h di anticipo. Se l'utente chiede dei chiarimenti, deve ottenere una risposta entro un paio di ore.
<b>Media</b>	<b>Sostenibilità</b>	Il sistema deve essere adattabile ai cambiamenti del dominio dell'applicazione e deve essere mantenibile per poter gestire nuove tecnologie e correggere difetti esistenti.

## 1.3 Riferimenti

Per realizzare il progetto sono stati utilizzati:

- Slide del docente;
- B. BRUEGGE, A.H. DUTOIT, OBJECT ORIENTED SOFTWARE ENGINEERING – USING UML, PATTERNS AND JAVA, PRENTICE HALL, 3D EDITION, 2009;
- RAD di DRESS-STORE;

## 1.4 Organizzazione del contenuto

Il presente documento illustra la struttura richiesta per la piattaforma Dress-Store. Nella prima parte del documento vengono esposti gli obiettivi di progettazione. Al secondo e al terzo punto del documento vengono espresse l'architettura software corrente e quella da noi proposta. Poi, viene mostrata la decomposizione del sistema in sottosistemi, delle componenti hardware/software off-the-shelf che verranno utilizzate ai fini della progettazione e di come verranno gestiti i dati persistenti.

La sezione riguardante l'architettura del sistema proposto si conclude con i dettagli su sicurezza e gestione degli accessi e sulla gestione delle condizioni limite. Il documento prosegue con una sezione che illustra i dettagli della decomposizione in sottosistemi spiegando, in modo più dettagliato, i servizi che ogni sottosistema offre agli altri sottosistemi. Il documento si conclude infine con un glossario.

## 2. Architettura del sistema corrente

Il progetto Dress-Store è un progetto “greenfield engineering”. Il sistema non sostituirà nessun sistema precedente.

Tuttavia, trovare architetture simili per un e-commerce è davvero semplice, sul web girano ormai migliaia di siti riguardanti lo shop-online (ASOS, Zalando, ecc.) il nostro intento è quello di rendere il sito molto semplice all'uso dell'utente e arrivare il prima possibile a DARE quello che il cliente VUOLE e che sta cercando.

## 3. Architettura del sistema proposto

### 3.1 Panoramica

Per realizzare il sistema, proponiamo un'applicazione web, una tipologia di software che meglio rispecchia le idee proposte. L'obiettivo, come ampiamente già discusso più volte è la creazione di un E-commerce per la gestione di un negozio di abbigliamento per la vendita di prodotti online.

Per memorizzare i dati persistenti è stato scelto di utilizzare un database relazionale. Il sistema è diviso in tre layer, adottando un'architettura three-tier: Storage, per la gestione dei dati persistenti, Presentation, per la gestione dell'interfaccia grafica e Controller per la gestione della logica di sistema. I layer sono organizzati secondo il noto pattern “MVC”. Ad un livello di dettaglio maggiore, invece, è possibile

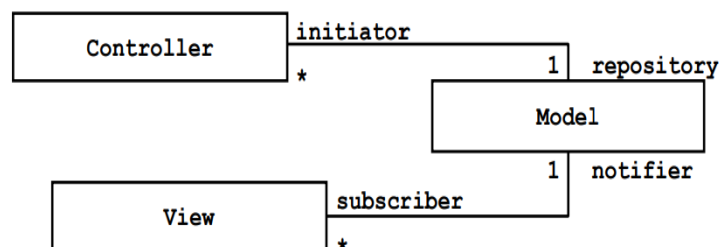
identificare ben 11 sottosistemi, ognuno dei quali si occupa di una funzionalità specifica, in modo da favorire una alta coesione ed un basso accoppiamento, quindi un cambiamento in un sottosistema non incide sugli altri. Come già più volte detto il sistema prevedrà l'interazione di quattro tipi di utenti: Cliente, Proprietario, Magazziniere e Gestore Marketing, ognuno con le proprie funzionalità specifiche.

## 3.2 Decomposizione in sottosistemi

### 3.2.1 Decomposizione in layer

La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestire aspetti e funzionalità differenti:

- **View:** gestisce gli elementi dell'interfaccia grafica e gli eventi generati su di essi;
- **Controller:** si occupa della gestione della logica del sistema;
- **Model:** si occupa della gestione e dello scambio dei dati tra i sottosistemi;



### 3.2.2 Decomposizione in sottosistemi

Dopo un'analisi funzionale abbiamo deciso di dividere il sistema nel modo seguente: elevata coesione (poiché le classi del sottosistema effettuano operazioni simili e sono in relazione tra loro) e basso accoppiamento (perché eventuali modifiche ad un sottosistema non modifichino gli altri). La suddivisione è stata effettuata secondo la "Decomposizione del Sottosistema in Strati". In tale suddivisione effettuata abbiamo tre livelli: uno che gestisce la visualizzazione dei dati di un utente, uno che gestisce le operazioni e un altro che immagazzina i dati.

*Primo livello:*

- Interfaccia Cliente: sottosistema che gestisce l'interfaccia di un cliente e che mostra tutte le funzionalità di sua competenza
- Interfaccia Utente non loggato: sottosistema che gestisce l'interfaccia di un utente che non ha ancora effettuato il "Login" e quindi mette a disposizione solo alcune delle funzionalità offerte ad un cliente
- Interfaccia Proprietario: sottosistema che gestisce l'interfaccia del Proprietario e che mostra tutte le funzionalità offerte a quest'ultimo
- Interfaccia Gestore Marketing: sottosistema che gestisce l'interfaccia del Gestore Marketing, in questo caso sono mostrate tutte le operazioni che può compiere questa tipologia di utente

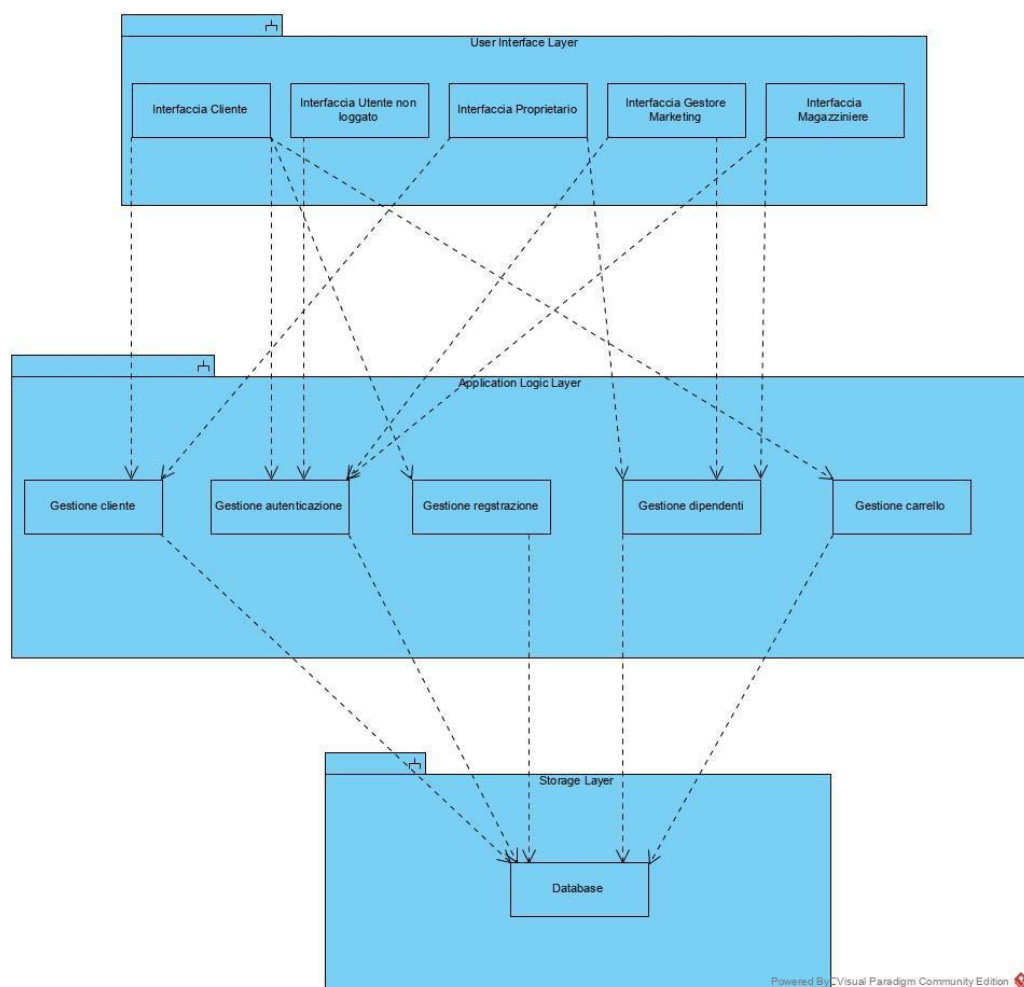
- Interfaccia Magazziniere: sottosistema che gestisce l'interfaccia di un magazziniere, mostrando tutte le funzionalità che gli competono.

## Secondo livello

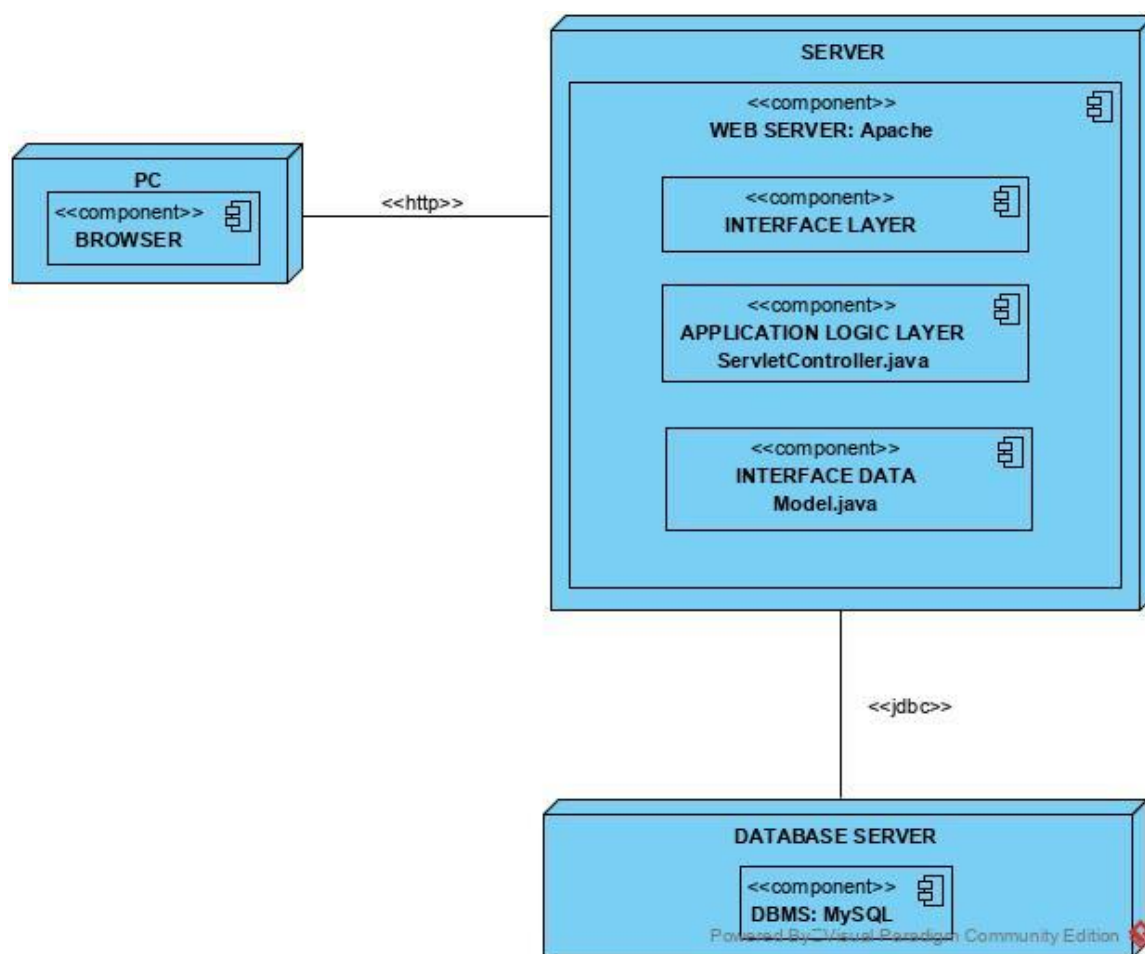
- Gestione Cliente: sottosistema che gestisce il profilo di un cliente e le modifiche che può apportare ai suoi dati.
- Gestione autenticazione: sottosistema che gestisce le regole di autenticazione degli utenti e l'accesso alle varie funzionalità.
- Gestione registrazione: sottosistema che gestisce le regole di registrazione degli utenti.
- Gestione dipendenti: sottosistema che gestisce le informazioni del negozio, dei prodotti, dei dipendenti e le modifiche su di esse.
- Gestione Carrello: sottosistema che gestisce le procedure di acquisto e le eventuali conseguenze.

## Terzo livello

- Database: un singolo sottosistema che gestisce ed immagazzina i dati



### 3.3 Mapping Hardware-Software



Il sistema che si desidera sviluppare utilizzerà una struttura hardware costituita da un server che risponderà ai servizi richiesti dai client. Il client è una qualsiasi macchina attraverso la quale un utente può collegarsi al sistema, utilizzando una connessione ad Internet.

La macchina server gestisce la logica e i dati persistenti contenuti nel database.

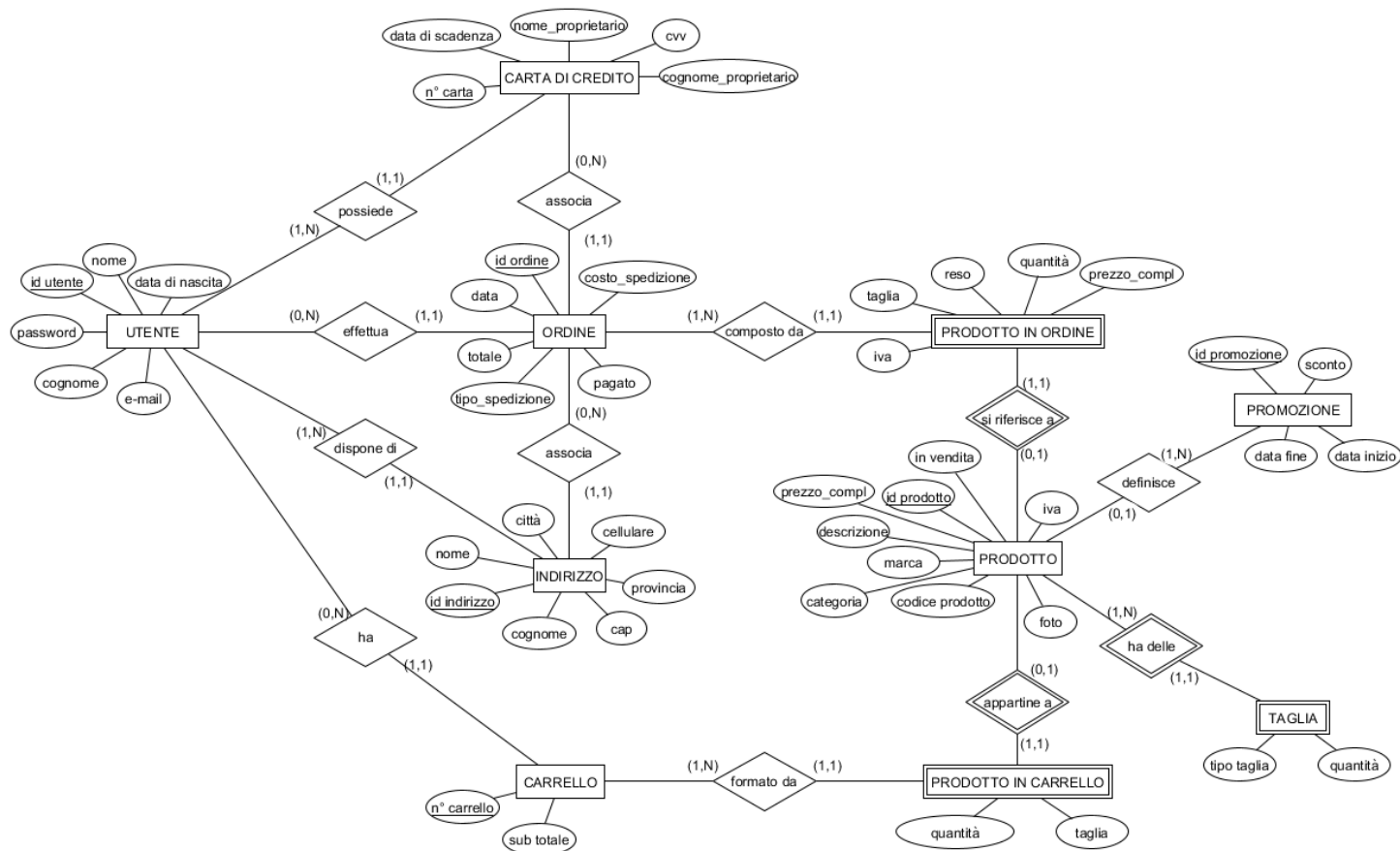
Il client ed il server saranno collegati tramite protocolli "HTTP", con il quale il client inoltra le proprie richieste.

Le componenti necessarie al client sono: una macchina con una connessione ad Internet con installato un browser web.

Per il server, invece, è richiesta (oltre una connessione ad Internet) una macchina capace di immagazzinare una grande quantità di dati. La parte software richiesta è un DBMS, che consente la connessione di più client.



### 3.4 Gestione persistente dei dati



#### Utente

NOME	TIPO	VINCOLI	KEY
<b>id_utente</b>	INT(10)	NotNull, Auto Incremental	PRIMARY KEY
<b>nome</b>	VARCHAR(30)	NOT NULL	
<b>cognome</b>	VARCHAR(30)	NOT NULL	
<b>e-mail</b>	VARCHAR(30)	NOT NULL	
<b>password</b>	VARCHAR(200)	NOT NULL	
<b>tipo</b>	TINYINT(1)	NOT NULL	
<b>data_nascita</b>	DATE	NOT NULL	

### Prodotto

NOME	TIPO	VINCOLI	KEY
<b>id_prodotto</b>	INT(10)	NOT NULL, AUTO INCREMENTAL	PRIMARY KEY
<b>codice_prodotto</b>	VARCHAR(30)	NOT NULL	
<b>descrizione</b>	VARCHAR(200)	NOT NULL	
<b>marca</b>	VARCHAR(20)	NOT NULL	
<b>modello</b>	VARCHAR(20)	NOT NULL	
<b>prezzo</b>	FLOAT	NOT NULL	
<b>iva</b>	INT(10)	NOT NULL	
<b>in_vendita</b>	TINYINT(1)	NOT NULL	
<b>categoria</b>	VARCHAR(20)	NOT NULL	
<b>foto</b>	VARCHAR(50)	NOT NULL	

### Prodotto in carrello

NOME	TIPO	VINCOLI	KEY
<b>id_prodotto</b>	INT(10)	NOT NULL, AUTO INCREMENTAL	PRIMARY KEY
<b>n° carrello</b>	INT(11)	NOT NULL	PRIMARY KEY
<b>quantità</b>	INT(3)	NOT NULL	
<b>taglia</b>	VARCHAR(1)	NOT NULL	

### Prodotto in ordine

NOME	TIPO	VINCOLI	KEY
<b>id_prodotto</b>	INT(10)	NOT NULL, AUTO INCREMENTAL	PRIMARY KEY
<b>id_ordine</b>	INT(11)	NOT NULL	PRIMARY KEY
<b>prezzo</b>	FLOAT	NOT NULL	
<b>iva</b>	INT(5)	NOT NULL	
<b>taglia</b>	VARCHAR(5)	NOT NULL	
<b>quantità</b>	INT(11)	NOT NULL	
<b>reso</b>	TINYINT(1)	NOT NULL	

## Carta di credito

NOME	TIPO	VINCOLI	KEY
<b>num_carta</b>	INT(20)	NOT NULL	PRIMARY KEY
<b>data_scadenza</b>	DATE	NOT NULL	
<b>cvv</b>	INT(3)	NOT NULL	
<b>nome_proprietario</b>	VARCHAR(30)	NOT NULL	
<b>cognome_proprietario</b>	VARCHAR(30)	NOT NULL	

## Indirizzo

NOME	TIPO	VINCOLI	KEY
<b>id_indirizzo</b>	INT(20)	NOT NULL, AUTO INCREMENTAL	PRIMARY KEY
<b>nome</b>	VARCHAR(30)	NOT NULL	
<b>cognome</b>	VARCHAR(30)	NOT NULL	
<b>cap</b>	INT(5)	NOT NULL	
<b>provincia</b>	CHAR(2)	NOT NULL	
<b>città</b>	VARCHAR(30)	NOT NULL	
<b>via</b>	VARCHAR(50)	NOT NULL	
<b>cellulare</b>	VARCHAR(13)	NOT NULL	

## Ordine

NOME	TIPO	VINCOLI	KEY
<b>id_ordine</b>	INT(11)	NOT NULL, AUTO INCREMENTAL	PRIMARY KEY
<b>data</b>	DATE	NOT NULL	
<b>pagato</b>	TINYINT(1)	NOT NULL	
<b>totale</b>	FLOAT	NOT NULL	
<b>tipo_spedizione</b>	VARCHAR(10)		
<b>costo_spedizione</b>	FLOAT	NOT NULL	

### Promozione

NOME	TIPO	VINCOLI	KEY
<b>id_promozione</b>	INT(20)	NOT NULL, AUTO INCREMENTAL	PRIMARY KEY
<b>data_inizio</b>	DATE	NOT NULL	
<b>data_fine</b>	DATE	NOT NULL	
<b>sconto</b>	INT(11)	NOT NULL	

### Taglia

NOME	TIPO	VINCOLI	KEY
<b>id_prodotto</b>	INT(20)	NOT NULL	PRIMARY KEY
<b>taglia</b>	VARCHAR(5)	NOT NULL	
<b>quantità</b>	INT(11)	NOT NULL	

## 3.5 Controlli accesso e sicurezza

Per il controllo dell'accesso viene fornito un meccanismo di autenticazione, il login, che prevede l'inserimento dell'indirizzo e-mail e della password. Nel caso in cui un utente non loggato tenta di accedere ad una area riservata che richiede l'autenticazione, gli sarà notificato l'impossibilità di accedere a tale area e successivamente sarà indirizzato alla pagina di login.

Gli utenti si dividono in: utente non registrato, cliente, magazziniere, gestore marketing e proprietario.

L'utente non registrato può registrarsi, può vedere il catalogo ma senza fare acquisti.

Il cliente aggiunge alle funzionalità dell'utente non registrato, modificare il proprio profilo, effettuare il login/logout, vedere il catalogo, effettuare un reso, inserire/eliminare prodotti nel carrello, visualizzare il carrello, visualizzare il proprio storico acquisti, modificare le modalità di pagamento (aggiungere o eliminare carte di credito).

Il magazziniere invece, può controllare il numero di scorte di un determinato prodotto, modificare il numero di prodotti nel database.

Il gestore marketing può scegliere il prezzo di un prodotto, effettuare sconti promozionali, cambiare prezzo ai prodotti.

Il proprietario può visualizzare statistiche di vendite, controllare il guadagno, controllare i dati di tutti gli utenti.

*Matrice degli accessi:*

ATTORE	GESTIONE UTENTE	GESTIONE PAGAMENTO	GESTIONE PRODOTTO	GESTIONE CARRELLO	GESTIONE ORDINE	GESTIONE PROMOZIONE
Utente non registrato	Registrazione		Ricerca prodotti, visualizza catalogo			
Cliente	Login utente, logout utente, modifica profilo utente	Aggiungi CC, Rimuovi CC	Visualizza catalogo, acquista prodotto	Aggiungi al carrello, Modifica quantità dal carrello, Rimuovi dal carrello, Visualizza carrello	Visualizza storico acquisti	
Magazziniere	Login magazziniere, logout magazziniere, modifica profilo utente		Visualizza quantità, modifica quantità, aggiungi prodotto			
Gestore marketing	Login gestore marketing, logout gestore marketing, modifica profilo utente, visualizza utente		Aggiungi prezzo, modifica prezzo			Aggiungi promozione
Proprietario	Login proprietario, logout proprietario, modifica profilo utente, visualizza lista utenti, elimina utente		Visualizza storico vendite			

### 3.6 Controllo globale del sistema

Il sistema ha un flusso guidato dagli eventi, pertanto richiede un'iterazione continua da parte dell'utente.

Quindi non ci sarà nessun evento prestabilito ma sarà l'utente a decidere le operazioni da compiere.

### 3.7 Boundary Conditions

- *Start-up*

Per rendere accessibile il sistema è necessario avviare un web server che fornisca i servizi di un Database. Una volta fatto ciò, per un client sarà possibile effettuare l'accesso al sistema tramite l'area di login (o se non si è ancora registrato, potrà procedere alla registrazione tramite l'apposita area).

Successivamente, il sistema provvederà a far accedere l'utente alla propria area personale.

- *Terminazione*

Per consentire la terminazione del server, un amministratore del sistema dovrà effettuare questa procedura. Dopo aver effettuato ciò, nessun client si potrà più connettere al sito fino a quando non verrà rifatta la procedura di Start-up precedentemente descritta.

Lato client, invece, la terminazione avviene dopo aver effettuato il Log-out.

- *Fallimento*

Il fallimento del sistema si può verificare a causa di diversi fattori:

- Sovraccarico del database server: in questo caso, si procederà al salvataggio periodico dei dati.
- Interruzione dell'alimentazione: il sistema comunicherà agli utenti l'impossibilità di accedere al sistema.
- Errore dell'hardware: non è prevista nessuna contromisura
- Errore dell'avvio del server a causa di errori di implementazione: in questo caso si procederà con l'ispezione del codice e con la correzione degli errori.

## 4. Glossario

Nome	Descrizione
<b>Deployment Diagram</b>	schema che descrive la struttura dinamica del sistema
<b>Server</b>	programma di gestione di un servizio che invia informazioni in un particolare formato e che viene ricevuto e interpretato da un client
<b>DBMS</b>	Database Management System, un programma informatico progettato per gestire un database, cioè di numerose strutture dati.
<b>Layer</b>	insieme di classi con funzionalità simili
<b>Start-up</b>	processo di accensione di un dispositivo di un sistema
<b>Design Goals</b>	obiettivi di design
<b>DB</b>	Database (sistema di memorizzazione per l'archiviazione dei dati permanenti)
<b>HTML</b>	HyperText Markup Language (linguaggio di markup utilizzato per la definizione della struttura della pagina Web)
<b>Java</b>	linguaggio orientato agli oggetti
<b>Layout</b>	impaginazione struttura grafica del portale e dell'applicazione
<b>RAD</b>	Requirements Analysis Document
<b>SDD</b>	System Design Document