SDD

System Design Document



|  |  |
| --- | --- |
|  |  |
| Versione | 2.0 |
| Data | 04/02/2022 |
| Destinatario | Prof. Carmine Gravino |
| Presentato da | Abbatiello Giuseppe, Ancona Marco, Boudad El Mehdi, De Palma Marco |
|  |  |

## RevisionHistory

Di seguito è riportata la revision history del documento, per maggiori dettagli vedere il file excel del report contributi.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 15/01/2022 | 0.1 | Prima stesura | Team |
| 16/01/2022 | 0.2 | Introduzione | Team |
| 17/01/2022 | 0.3 | Stesura sezione 1.1 e 1.2 | Team |
| 18/01/2022 | 0.4 | Stesura sezione 1.3 | EB |
| 19/01/2022 | 0.5 | Stesura sezione 2 | GA |
| 20/01/2022 | 06 | Stesura sezione 3.1 | MDP |
| 21/01/2022 | 0.7 | Stesura sezione 3.2 | MDP, GA |
| 22/01/2022 | 0.8 | Stesura sezione 3.3 | EB, GA |
| 23/01/2022 | 0.9 | Stesura sezione 3.4 | MA, EB |
| 24/01/2022 | 1.0 | Stesura sezione 3.5 | GA |
| 25/01/2022 | 1.1 | Stesura sezione 3.6 | GA |
| 26/01/2022 | 1.2 | Stesura sezione 3.7 | Team |
| 27/01/2022 | 1.3 | Stesura sezione 4 | MDP, GA |
| 28/01/2022 | 1.4 | Stesura sezione 5 | MDP |
| 04/02/2022 | 2.0 | Revisione | Team |

# Membri del team

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Ruolo nel progetto | Acronimo | Informazioni di contatto |
| Giuseppe Abbatiello | Team Member | GA | g.abbatiello@studenti.unisa.it |
| El Mehdi Boudad | Team Member | EB | e.boudad@studenti.unisa.it |
| Marco De Palma | Team Member | MDP | [m.depalma6@studenti.unisa.it](mailto:m.depalma6@studenti.unisa.it) |
| Marco Ancona | Team Member | MA | m.ancona2@studenti.unisa.it |

Sommario

[RevisionHistory 2](#_Toc95680421)

[Membri del team 2](#_Toc95680422)

[1 Introduzione 5](#_Toc95680423)

[1.1 Scopo del sistema 5](#_Toc95680424)

[1.2 Obiettivi di design 5](#_Toc95680425)

[Design Goals 6](#_Toc95680426)

[Trade-off 7](#_Toc95680427)

[1.3 Definizioni, acronimi e abbreviazioni 8](#_Toc95680428)

[2 Architettura del sistema corrente 8](#_Toc95680429)

[3 Architettura del sistema proposto 9](#_Toc95680430)

[3.1 Panoramica sulla sezione 9](#_Toc95680431)

[3.2 Decomposizione del sistema 9](#_Toc95680432)

[Diagramma architetturale 10](#_Toc95680433)

[Sottosistema Autenticazione 11](#_Toc95680434)

[Sottosistema Acquisto 12](#_Toc95680435)

[Sottosistema Segnalazione 13](#_Toc95680436)

[3.3 Mapping hardware/software 13](#_Toc95680437)

[3.4 Gestione dati persistenti 15](#_Toc95680438)

[Introduzione 15](#_Toc95680439)

[Schema ER Database 15](#_Toc95680440)

[Dizionario dei dati 16](#_Toc95680441)

[3.5 controllo degli accesi 19](#_Toc95680442)

[3.6 Controllo globale del software 20](#_Toc95680443)

[3.7 Condizioni limite 20](#_Toc95680444)

[Avvio del sistema 20](#_Toc95680445)

[Spegnimento del sistema 21](#_Toc95680446)

[Fallimento del sistema 21](#_Toc95680447)

[Errore di accesso ai Dati Persistenti 22](#_Toc95680448)

[4 Servizi dei sottosistemi 23](#_Toc95680449)

[Sottosistema Registrazione 23](#_Toc95680450)

[Sottosistema Autenticazione 23](#_Toc95680451)

[Sottosistema Gestione Utente 23](#_Toc95680452)

[Sottosistema Gestione Prodotto 24](#_Toc95680453)

[Sottosistema Gestione Acquisto 24](#_Toc95680454)

[Sottosistema Lista dei desideri 25](#_Toc95680455)

[Sottosistema Segnalazione 25](#_Toc95680456)

[5 Glossario 25](#_Toc95680457)

# 1 Introduzione

## Scopo del sistema

PartnerShop è una piattaforma online destinata al mercato italiano il cui scopo è quello di semplificare l’interazione tra le persone che posseggono un negozio e gli utenti che vogliono acquistare dei prodotti. Le persone possono visualizzare tutti i prodotti del sito senza effettuare una registrazione, mentre per acquistare o salvare o caricare dei prodotti essa è necessaria, per quanto riguarda i venditori invece essi dovranno registrarsi al sito fornendo la partita IVA del negozio oltre ai dati utente. Il sito viene gestito da uno o più amministratori che si occupano della rimozione degli utenti e dei prodotti che possono essere pericolosi dopo aver ricevuto delle segnalazioni.

## Obiettivi di design

In questa sezione vengono descritti i Design Goals, ovvero le qualità sulle quali il sistema si dovrebbe focalizzare.

Seguendo le linee guide linee guida del libro Bernd Bruegge – Object Oriented Software Engineering i design goal sono statti suddivisi nelle seguenti categorie:

* Performance: includono I requisiti di spazio e tempo imposti dal sistema
* Dependability: determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema e le loro conseguenze.
* Maintenance: determina quanto sforzo è necessario per modificare il sistema dopo il suo rilascio.
* End user criteria: includono qualità che sono desiderabili dal punto di vista dell’utente, ma che non sono state coperte dai criteri di Performance e Dependability.

### Design Goals

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rank | ID design Goal | Descrizione | Categoria | RNF di origine |
| 4 | DG\_1 Tempo di risposta | Il sistema deve garantire che le richieste dell’utente saranno elaborate entro massimo 5 ms. | Performance | RNF\_P\_1 |
| 3 | DG\_2 Navigazione concorrente | Il sistema deve garantirà che il sistema funzioni fino a 5000 utenti connessi contemporaneamente. | Performance | RNF\_P\_4 |
| 1 | DG\_3  Affidabilità del sistema | Il sistema deve garantire che almeno il 90% delle operazioni vengano effettuate con successo. | Dependability | RNF\_A\_1 |
| 5 | DG\_4  Robustezza del sistema | Il sistema notificherà l’utente tramite messaggi di errore per guidare l’utente nella risoluzione | Dependability | RNF\_A\_2 |
| 2 | DG\_5  Sicurezza dei dati | Il sistema gestirà tutte le informazioni sensibili ad alto livello per garantire la sicurezza degli utenti. | Dependability | RNF\_A\_3  RNF\_L\_1 |
| 6 | DG\_6  Disponibilità del sistema | Il sistema garantirà la massima disponibilità con un limite di 50 ore all’anno di possibile manutenzione. | Dependability | RNF\_P\_2 |
| 7 | DG\_7  Estensibilità | Il sistema sarà sviluppato seguendo standard che permettono in maniera semplice di aggiungere nuove funzionalità. | Maintenance | RNF\_S\_1 |
| 12 | DG\_8  Modificabilità | Il sistema sarà sviluppato seguendo standard che permettono in maniera semplice di cambiare le funzionalità. | Maintenance | RNF\_S\_2 |
| 8 | DG\_9  Leggibilità | Il sistema sarà sviluppato seguendo dei design pattern che faciliteranno la leggibilità del codice. | Maintenance | RNF\_IM\_1 |
| 9 | DG\_10  Semplicità di utilizzo | Il sistema deve permettere agli utenti di essere utilizzato da chiunque e non solo da esperti. | End User | RNF\_U\_1 |
| 10 | DG\_11 Interfaccia intuitiva | L’interfaccia della piattaforma deve essere semplice e deve utilizzare componenti familiari per l’utente. | End User | RNF\_U\_2 |
| 11 | DG\_12 Immediatezza d’utilizzo | Gli utenti del sistema dovranno essere capaci di interagire senza l’uso di un manuale. | End User | RNF\_U\_3 |

### Trade-off

|  |  |
| --- | --- |
| Trade-off | Descrizione |
| Affidabilità del sistema vs Tempi di risposta | Si punta a massimizzare l’affidabilità del sistema anche a discapito di tempi di attesa più lunghi di massimo 10ms. |
| Sicurezza vs Tempo di risposta | Per garantire una maggiore sicurezza dei dati degli utenti si punta anche alla penalizzazione delle prestazioni con tempi di attesa più lunghi di massimo 10ms. |
| Robustezza vs Leggibilità | Il sistema sarà in grado di notificare l’utente in caso di malfunzionamenti, a discapito della leggibilità del codice. |

## Definizioni, acronimi e abbreviazioni

Vengono riportate alcune definizioni presenti nel corrente documento.

• Sottosistema: un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una relazione funzionale.

• Design Goal: le qualità sulle quali il sistema deve essere focalizzato.

• Dati Persistenti: dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati.

• Mapping Hardware/Software: studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.

• SDD: System Design Document

• RAD: Requirements Analysis Document

# Architettura del sistema corrente

Un sistema che presenta delle funzionalità simile al nostro è quello di eBay in cui le persone possono registrarsi come venditori e aggiungere dei prodotti e chiunque può effettuare degli acquisti tramite una semplice registrazione. Per permettere la separazione dei servizi del sistema viene utilizzata un’architettura di tipo Three Tier.

# Architettura del sistema proposto

## Panoramica sulla sezione

Il sistema che si vuole realizzare rientra nel campo della Green-field Engineering.

Il sistema proposto sarà basato sul pattern architetturale Three Tier.  
La motivazione risiede nel fatto che tale architettura facilita lo sviluppo di un applicazione web.

Nello sviluppo del sistema verranno utilizzati HTML5, CSS3, Javascript e JSTL per la parte di front-end e generazione della view.

Per il back-end verrà utilizzato Java.

Per la gestione del database verrà utilizzato MySQL.

## 3.2 Decomposizione del sistema

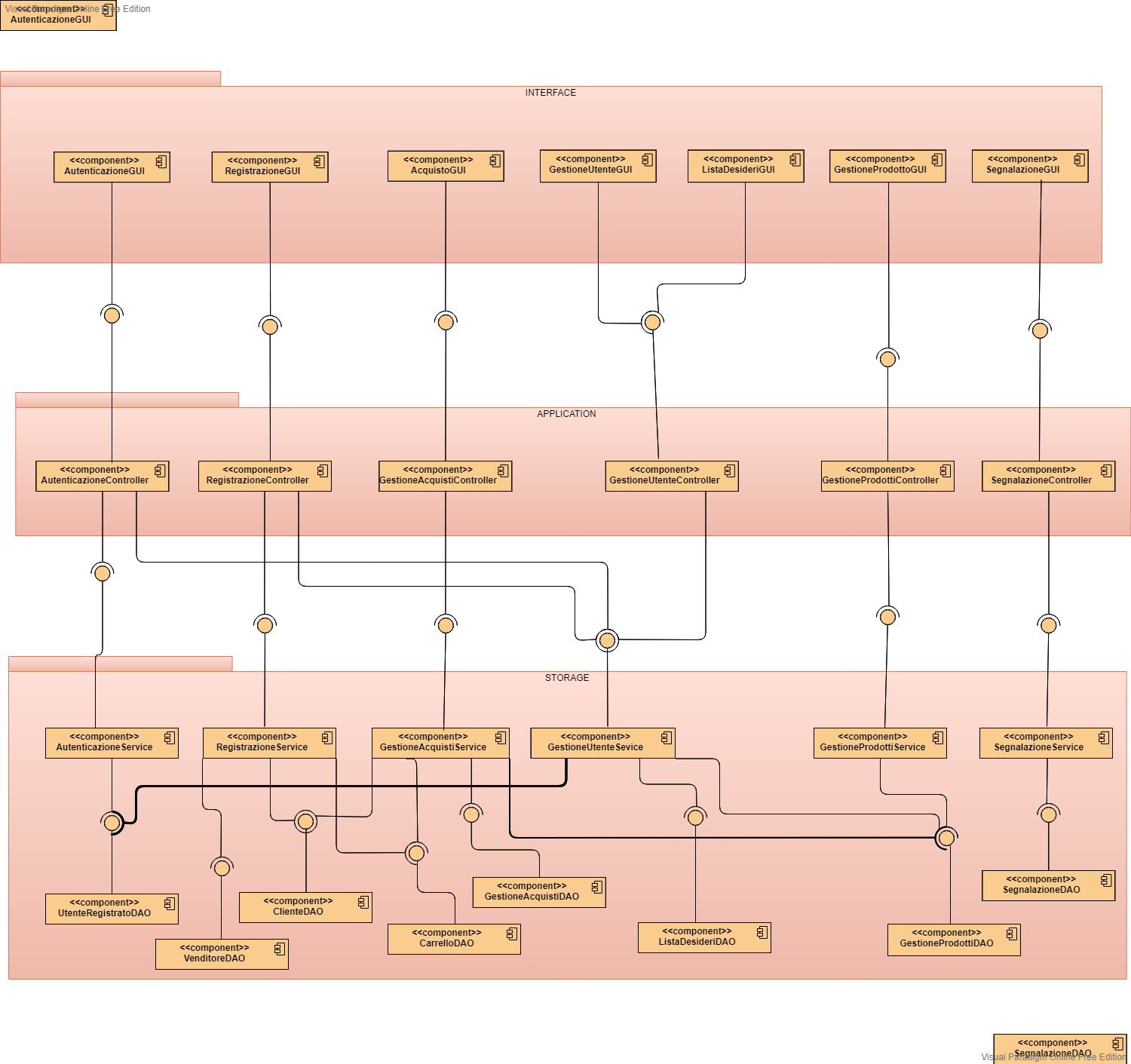
I sottosistemi individuati sono:

* **Registrazione**: si occupa della registrazione delle varie tipologie di utenti: venditore e cliente.
* **Autenticazione**: si occupa delle funzionalità di login e logout.
* **Gestione utente**: si occupa delle funzionalità di visualizzazione area utente, modifica dati account e cancella dati account.
* **Gestione prodotto**: si occupa delle funzionalità riguardanti la visualizzazione, la creazione, modifica e rimozione dei prodotti.
* **Gestione Acquisto**: si occupa delle funzionalità relative all’acquisto sono aggiungi e rimuovi al carrello, acquista prodotti.
* **Lista desideri**: si occupa delle funzionalità che riguardano la lista desideri e sono gestione e rimozione dalla lista dei desideri.
* **Segnalazione**: si occupa delle funzioni di creazione, visualizzazione e chiusura di una segnalazione.
* **Storage**: si occupa di fare da intermediario tra il livello di persistenza e il livello di controllo.
* **Persistenza**: si occupa di gestire la persistenza dei dati con il database.

Il sistema sarà composto dai seguenti componenti:

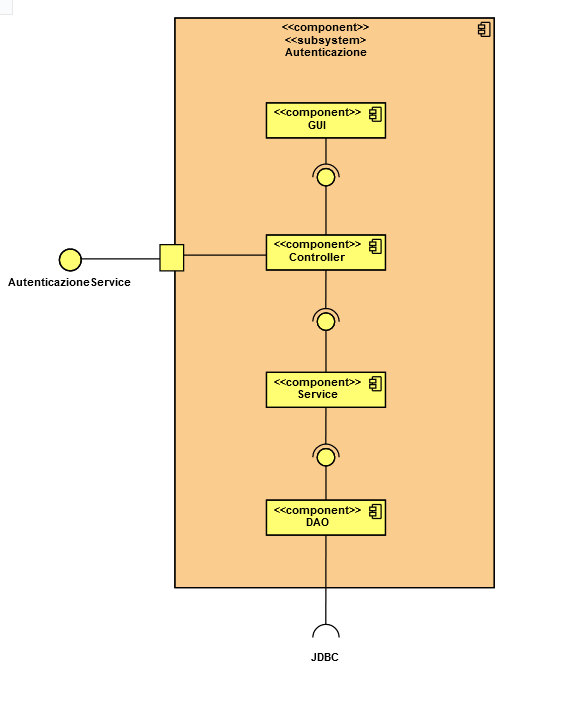
* **GUI**: “Graphic User Interface” - Interfaccia grafica che contiene le view che saranno visualizzate dagli utenti del sito.
* **Controller**: ne fanno parte i componenti che si occupano della logica del sistema.
* **Service**: ne fanno parte i componenti che si occupano della logica di business.
* **DAO**: “Data Access Object”, che si occupa di fornire accesso ai dati persistenti.

### Diagramma architetturale

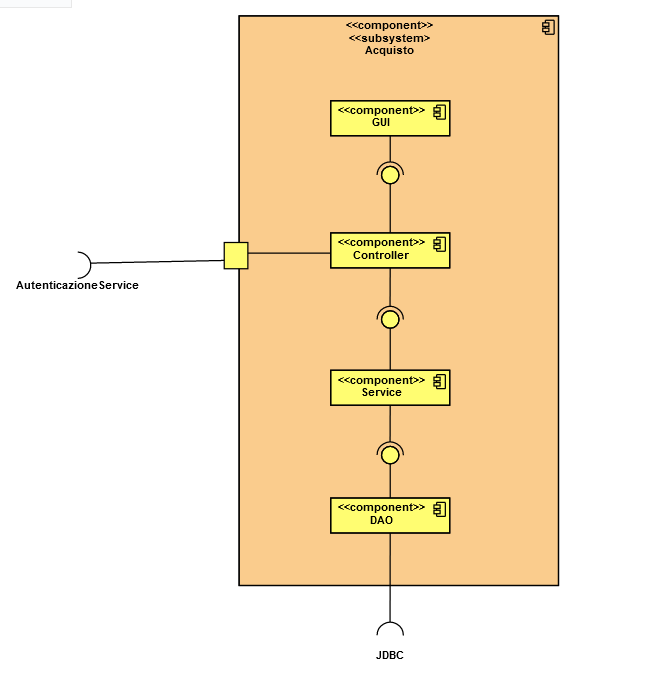


A seguito di alcune decisioni prese all’interno dell’ODD si è deciso di eliminare il controller per la Lista dei desideri che era stato deciso in precedenza.

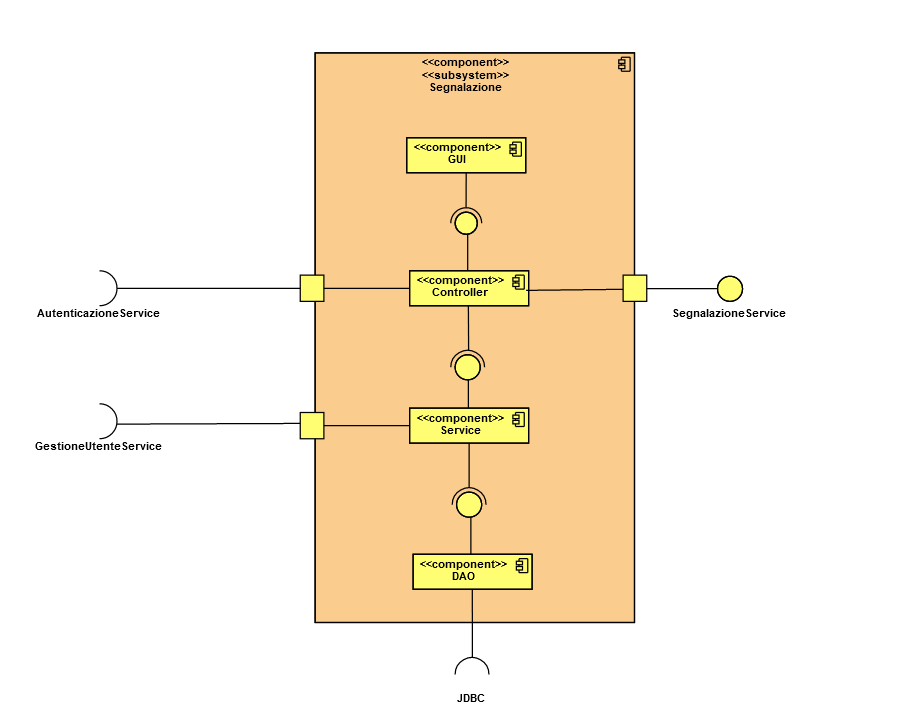
#### Sottosistema Autenticazione



#### Sottosistema Acquisto



#### Sottosistema Segnalazione

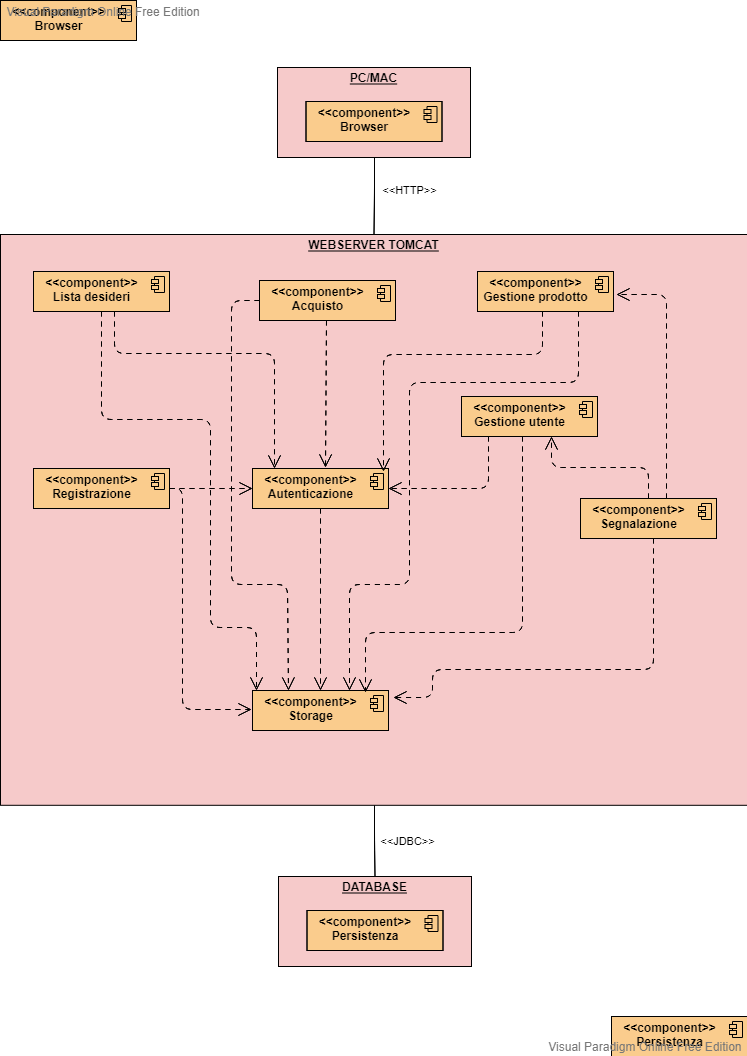


## 3.3 Mapping hardware/software

L’applicazione web sarà sviluppata da una piattaforma hardware che è formata da un server Tomcat che risponde alle richieste dei clienti che utilizzano un browser.

Poiché la nostra architettura è basata su un web server senza un'architettura distribuita, esso risiede su un solo nodo.

Quello che segue è un deployment diagram che descrive il mapping hardware/software.



## 3.4 Gestione dati persistenti

### Introduzione

Per la gestione del salvataggio si è scelto di usare un Database di tipo relazionale che permette un accesso concorrente ai dati e allo stesso tempo garantisce la loro consistenza.

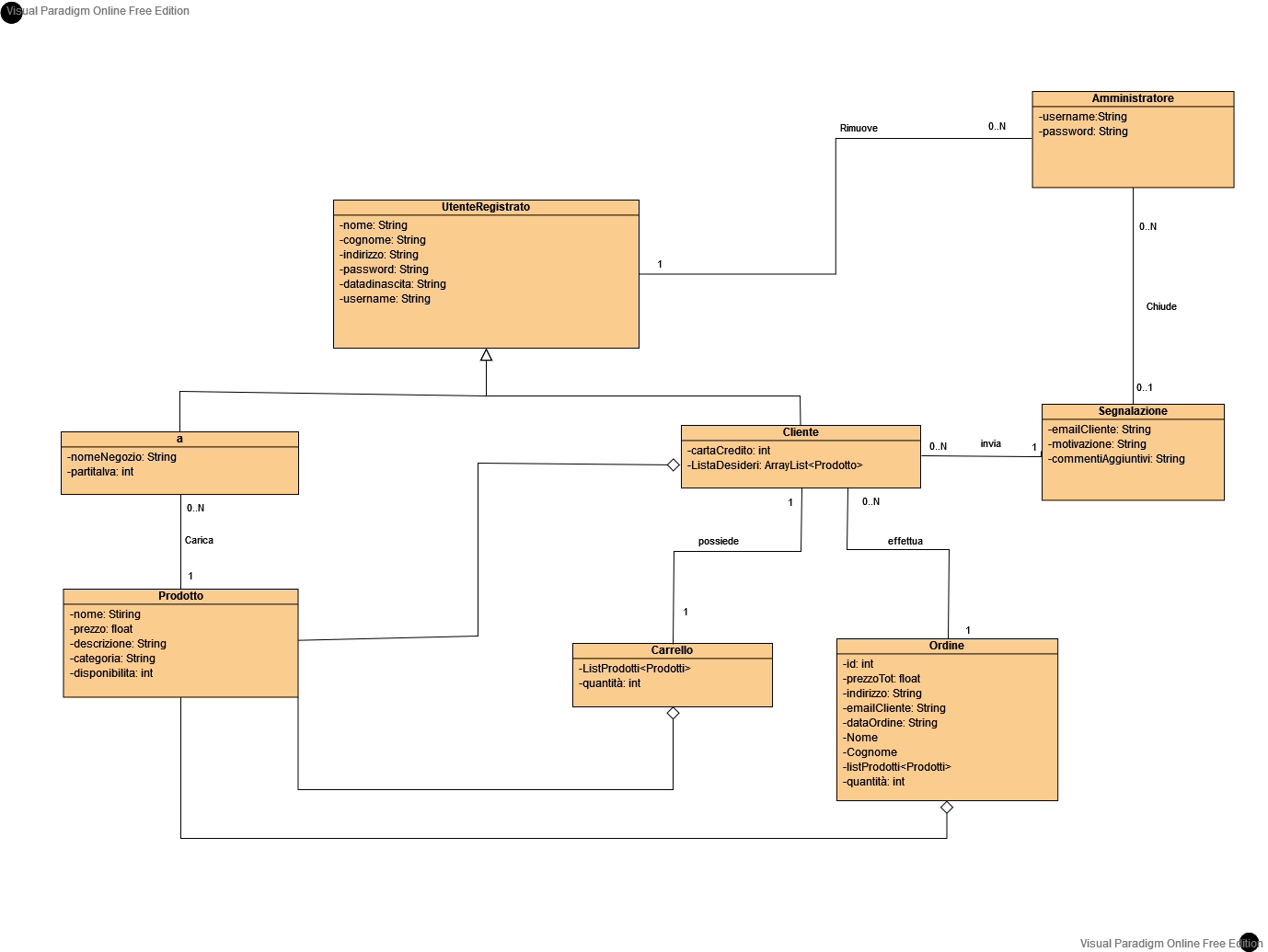
La scelta di un DBMS è stata presa per essere coerenti con alcuni vincoli stabili nei design goals, come:

* **Sicurezza dei dati**, perché assicura l’accesso protetto ai dati e per la criptazione delle password attraverso l’algoritmo SHA1.
* **Affidabilità dei dati**, perché permette di salvare copie dei dati e rispristinarli in caso di guasti.

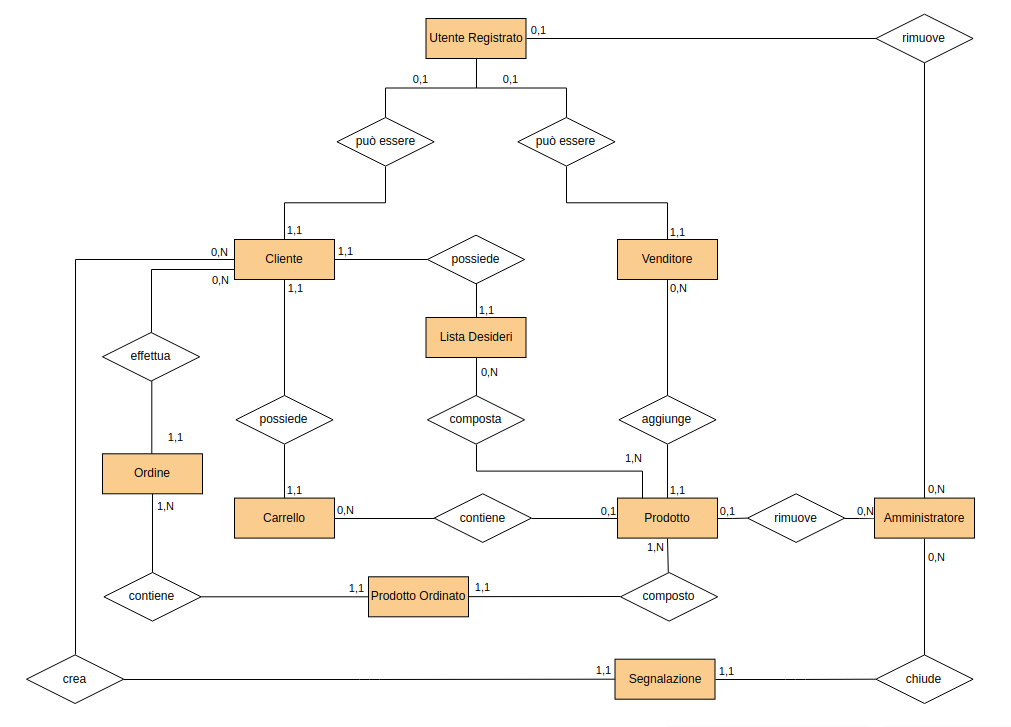
Ogni entità del Class Diagram presente nel RAD è stata tradotta in una tabella del Database, fatta eccezione per Prodotto Ordinato che deriva dalla relazione N a N presente tra le tabelle Prodotto e Ordine.

### CD\_SDD: Entity Class Diagram ristrutturato

Inoltre per motivi di efficienza l’entità lista desideri e stata eliminata ed inserita come attributo all’interno di cliente poiché non era conveniente implementarla come oggetto in quanto non l’unica classe che vi accede è proprio cliente.



### Schema ER Database



### Dizionario dei dati

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Entità | *Utente\_Registrato* | | |
| *Descrizione* | *Contiene i dati relativi ad un utente registrato.* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *nome* | *Varchar(50)* |  | *NOT NULL* |
| *cognome* | *Varchar(50)* |  | *NOT NULL* |
| *ddn* | *Date* |  | *NOT NULL* |
| *email* | *Varchar(50)* | *PRIMARY KEY* | *NOT NULL*  *UNIQUE* |
| *indirizzo* | *Varchar(500)* |  | *NOT NULL* |
| *username* | *Varchar(50)* |  | *NOT NULL*  *UNIQUE* |
| *password* | *Varchar(50)* |  | *NOT NULL* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *Cliente* | | |
| *Descrizione* | *Contiene i dati relativi ad un cliente.* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *email* | *Varchar(50)* | *PRIMARY KEY*  *FOREIGN KEY* | *NOT NULL*  *UNIQUE* |
| *creditcard* | *Varchar(50)* |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *Venditore* | | |
| *Descrizione* | *Contiene i dati relativi ad un venditore.* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *email* | *Varchar(50)* | *PRIMARY KEY*  *FOREIGN KEY* | *NOT NULL*  *UNIQUE* |
| *nomeNegozio* | *Varchar(50)* |  | *NOT NULL* |
| *partitaIVA* | *Varchar(50)* |  | *NOT NULL* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *Amministratore* | | |
| *Descrizione* | *Contiene i dati relativi ad un amministratore* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *nome* | *Varchar(50)* |  | *NOT NULL* |
| *cognome* | *Varchar(50)* |  | *NOT NULL* |
| *email* | *Varchar(50)* | *PRIMARY KEY* | *NOT NULL*  *UNIQUE* |
| *username* | *Varchar(50)* |  | *NOT NULL*  *UNIQUE* |
| *passwordhash* | *Varchar(50)* |  | *NOT NULL* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *Prodotto* | | |
| *Descrizione* | *Contiene i dati relativi ad un prodotto.* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *idProdotto* | *Int* | *PRIMARY KEY* | *NOT NULL*  *UNIQUE*  *AUTO\_INCREMENT* |
| *nome* | *Varchar(50)* |  | *NOT NULL* |
| *descrizione* | *Varchar(500)* |  | *NOT NULL* |
| *categoria* | *Varchar(50)* |  | *NOT NULL* |
| *quantità* | *Int* |  | *NOT NULL* |
| *emailVenditore* | *Varchar(50)* | *FOREIGN KEY* | *NOT NULL*  *UNIQUE* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *Carrello* | | |
| *Descrizione* | *Contiene i dati relativi ad un carrello* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *emailCliente* | *Varchar(50)* | *PRIMARY KEY*  *FOREIGN KEY* | *NOT NULL* |
| *idProdotto* | *Int* | *FOREIGN KEY* | *NOT NULL* |
| *quantità* | *Int* |  | *NOT NULL* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *prodottiOrdinati* | | |
| *Descrizione* | *Contiene i dati relativi ad un prodotto ordinato* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *idOrdine* | *Int* | *PRIMARY KEY*  *FOREIGN KEY* | *NOT NULL* |
| *idProdotto* | *Int* | *PRIMARY KEY*  *FOREIGN KEY* | *NOT NULL* |
| *quantità* | *Int* |  | *NOT NULL* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *Ordine* | | |
| *Descrizione* | *Contiene i dati relativi ad un ordine.* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *idOrdine* | *Int* | *PRIMARY KEY* | *NOT NULL*  *UNIQUE*  *AUTO\_INCREMENT* |
| *emailCliente* | *Varchar(50)* | *FOREIGN KEY* | *NOT NULL*  *UNIQUE* |
| *dataOrdine* | *Date* |  | *NOT NULL* |
| *indirizzo* | *Varchar(500)* |  | *NOT NULL* |
| *prezzoTot* | *Float* |  | *NOT NULL* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *Segnalazione* | | |
| *Descrizione* | *Contiene i dati relativi ad una segnalazione.* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *idSegnalazione* | *Int* | *PRIMARY KEY* | *NOT NULL*  *UNIQUE*  *AUTO\_INCREMENT* |
| *emailCliente* | *Varchar(50)* | *FOREIGN KEY* | *NOT NULL*  *UNIQUE* |
| *motivazione* | *Varchar(500)* |  | *NOT NULL* |
| *commento* | *Varchar(500)* |  | *NOT NULL* |
| *stato* | *Boolean* |  | *NOT NULL*  *DEFAULT\_FALSE* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Nome Entità* | *ListaDesideri* | | |
| *Descrizione* | *Contiene i dati relativi ad una lista desideri.* | | |
| *Nome campo* | *Tipo* | *Vincolo di chiave* | *Altri vincoli* |
| *emailCliente* | *Varchar(50)* | *PRIMARY KEY*  *FOREIGN KEY* | *NOT NULL* |
| *idProdotto* | *Int* | *FOREIGN KEY* | *NOT NULL* |

## 3.5 controllo degli accesi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attori  Oggetti | Cliente | Venditore | Amministratore | Ospite |
| Registrazione |  |  |  | RegistrazioneCliente  RegistrazioneVenditore |
| Autenticazione | Login  Logout | Login  Logout | Login  Logout |  |
| Gestione Utente | VisualizzaArea ModificaDati  CancellaDati | VisualizzaArea ModificaDati  CancellaDati | VisualizzaArea ModificaDati  CancellaDati |  |
| Gestione  Prodotto | VisualizzaProdotto  RicercaProdotto | VisualizzaProdotto  ModificaDatiProdotto  RimozioneProdotto  RicercaProdotto  AggiuntaProdotto | VisualizzaProdotto  RicercaProdotto  RimozioneProdotto | VisualizzaProdotto  RicercaProdotto |
| Gestione Acquisto | AggiungiAlCarrello  RimuoviDalCarrello VisualizzaOrdine  AcquistaProdotto |  |  |  |
| Gestione  ListaDesideri | AggiungiAllaListaDes  RimuoviDallaListaDes |  |  |  |
| Segnalazione | EffettuaSegnalazione |  | VisualizzaSegnalazione  ChiusuraSegnalazione |  |

## 3.6 Controllo globale del software

Il sistema ParterShop è una piattaforma con un'interfaccia grafica, con la quale l’utente può interagire e generare degli eventi. Per ogni evento che viene generato è richiamato l’oggetto di controllo relativo alla funzionalità che si occuperà della logica di controllo per poi rivolgersi ai servizi per la logica applicativa.

Per tali motivi il sistema utilizzerà un meccanismo di controllo del flusso di tipo ***event-driven*** essendo esso una web-application.

## 3.7 Condizioni limite

### Avvio del sistema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Identificativo | | UC\_BC1 –  Avvio del sistema | | Data | 26/01/2022 |
| **Versione** | **0.1** |
| **Autori** | Giuseppe Abbatiello |
| Descrizione | | Lo UC permette l’avvio del sistema. | | | |
| Attore principale | | Amministratore | | | |
| Attori secondari | | NA | | | |
| Entry Condition | | L’amministratore accede al server | | | |
| Exit Condition  On Success | | Il sistema si avvia correttamente. | | | |
| Exit Condition  On Failure | | Il sistema non si avvia. | | | |
| Flusso di eventi principale | | | | | |
| 1 | **Amministratore** | | Esegue il comando che avvia il sistema. | | |
| 2 | **Sistema** | | Il sistema verifica che in precedenza era stato spento correttamente, e verifica che l’integrità dei dati persistenti. Una volta verificata l’integrità rende tutte le funzionalità disponibili agli utenti. | | |
| Flusso di eventi alternativo: dati persistenti dannegiati | | | | | |
| 2.a1 | Sistema | | Notifica l’amministratore la non integrità dei dati e non effettua l’avvio. | | |
| 2.a2 | Amministratore | | Corregge il problema dei dati. | | |
| 2.a3 | Amministratore | | Avvia il sistema. | | |

### Spegnimento del sistema

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Identificativo | | UC\_BC2 -  Spegnimento del Sistema | | Data | 26/01/2022 | |
| **Versione** | 0.1 | |
| **Autori** | Marco Ancona | |
| Descrizione | | Lo UC permette lo spegnimento del sistema. | | | | |
| Attore principale | | Amministratore | | | | |
| Attori secondari | | NA | | | | |
| Entry Condition | | L’amministratore è loggato nel Server  AND  Il Sistema è stato avviato, non è stato spento. | | | | |
| Exit Condition  On Success | | Il sistema viene spento correttamente. | | | | |
| Exit Condition  On Failure | | Il sistema non viene spento. | | | | |
| Flusso di eventi principale | | | | | |
| 1 | **Amministratore** | | Invia l’input per lo spegnimento del sistema. | | | |
| 2 | **Sistema** | | Controlla se sono presenti connessioni attive, se non ci sono spegne il sistema. | | | |
| Flusso di eventi alternativo: Sono presenti connessioni ancora attive | | | | | |
| 2.a1 | **Sistema** | | Forza la chiusura delle connessioni attive. | | | |
| 2.a2 | **Sistema** | | Comunica all’Amministratore il successo nello spegnimento del sistema. | | | |

### Fallimento del sistema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Identificativo  UC\_BC3 – Fallimento del Sistema | | | Data | 26/01/2022 | |
|  | | | **Versione** | **0.1** |
|  | | | **Autori** | El Mehdi Boudad |
| Descrizione | | Lo UC definisce il comportamento del sistema in caso di fallimento. | | | |
| Attore principale | | Amministratore | | | |
| Attori secondari | | NA | | | |
| Entry Condition | | Il sistema viene sospeso inaspettatamente. | | | |
| Exit Condition  On Success | | Il sistema è riavviato correttamente. | | | |
| Exit Condition  On Failure | | Il sistema non è riavviato. | | | |
| Flusso di eventi principale | | | | | |
| 1 | **Amministratore** | | Include UC\_BC1 | | |

### Errore di accesso ai Dati Persistenti

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Identificativo | | UC\_BC4 –  Errore di accesso ai dati persistenti | | Data | 26/01/2022 |
| **Versione** | **0.1** |
| **Autori** | Marco De Palma |
| Descrizione | | Lo UC descrive il comportamento del sistema qualora fosse impossibile accedere ai dati persistenti o questi risultassero corrotti. | | | |
| Attore principale | | Amministratore | | | |
| Attori secondari | | NA | | | |
| Entry Condition | | Il Sistema non può accedere ai dati persistenti  OR  I dati persistenti risultano corrotti | | | |
| Exit Condition  On Success | | Il Sistema riprende il normale funzionamento | | | |
| Exit Condition  On Failure | | Il Sistema non riprende il normale funzionamento | | | |
| Flusso di eventi principale | | | | | |
| 1 | **Sistema:** | | Comunica all’amministratore l’impossibilità di accedere ai dati persistenti. | | |
| 2 | **Amministratore:** | | Il sistema verifica che in precedenza era stato spento correttamente, e verifica che l’integrità dei dati persistenti. Una volta verificata l’integrità rende tutte le funzionalità disponibili agli utenti. | | |
| Flusso di eventi alternativo: | | | | | |
| 2.a1 | Sistema | |  | | |
|  | | | | | |

# Servizi dei sottosistemi

In questa sezione sono descritti i servizi sei sottosistemi precedentemente elencati.

### Sottosistema Registrazione

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Registrazione cliente | Questa funzionalità permette la registrazione di un cliente. | RegistrazioneService |
| Registrazione venditore | Questa funzionalità permette la registrazione di un venditore. | RegistrazioneService |

### Sottosistema Autenticazione

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Login | Questa funzionalità permette di effettuare l’accesso al sistema tramite le proprie credenziali e ottenere le funzionalità che offre. | AutenticazioneService |
| Logout | Questa funzionalità permette di disconnettersi dal sistema. | AutenticazioneService |

### Sottosistema Gestione Utente

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Visualizza area utente | Permette di visualizzare i propri dati relativi all’area utente. | GestioneUtenteService |
| Modifica area utente | Permette di modificare i propri dati relativi all’area utente. | GestioneUtenteService |
| Cancellazione dati account | Permette di cancellare il proprio account dal sistema. | GestioneUtenteService |

### Sottosistema Gestione Prodotto

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Visualizzazione prodotto | Permetti di visualizzare i dettagli di un prodotto. | GestioneProdottoService |
| Modifica prodotto | Permette di modifica i dati di un prodotto. | GestioneProdottoService |
| Rimozione prodotto | Permette di rimuovere un prodotto. | GestioneProdottoService |
| Ricerca Prodotto | Permette di effettuare una ricerca di un prodotto. | GestioneProdottoService |
| Aggiunta Prodotto | Permette di inserire un prodotto | GestioneProdottoService |

### Sottosistema Gestione Acquisto

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Aggiungi a carrello | Permette di aggiungere un prodotto al carrello. | GestioneAcquistoService |
| Rimuovi dal carrello | Permette di rimuovere un prodotto al carrello. | GestioneAcquistoService |
| Acquista prodotto | Permette di effettuare l’acquisto di un prodotto. | GestioneAcquistoService |
| Visualizza ordine | Permette di visualizzare i dettagli di un ordine. | GestioneAcquistoService |

### Sottosistema Lista dei desideri

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Aggiungi alla Lista dei desideri | Questa funzionalità permette di aggiungere dei prodotti alla lista dei desideri. | ListaDesService |
| Rimuovi dalla Lista dei desideri | Questa funzionalità permette di rimuovere dei prodotti dalla lista dei desideri. | ListaDesService |

### Sottosistema Segnalazione

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Effettua segnalazione | Permette di effettuare una segnalazione. | SegnalazioneService |
| Visualizza segnalazione | Permette di visualizzare i dettagli di una segnalazione. | SegnalazioneService |
| Chiusura Segnalazione | Permette di chiudere una segnalazione. | SegnalazioneService |

# 5 Glossario

Nella presente sezione sono raccolti le sigle o i termini del documento che necessitano di una definizione.

|  |  |
| --- | --- |
| Sigla/Termine | Definizione |
| Event driven | Event driven è un linguaggio di programmazione orientato agli eventi. Per evento si intende una qualsiasi azione computa dall'utente intercettabile dal programma al cui verificarsi si scatena una routine. |
| DBMS | un Database Management System è un sistema software progettato per consentire la creazione, la manipolazione e l'interrogazione efficiente di database. |
| PartnerShop | Nome dell’applicativo che si andrà a realizzare. |
| Piattaforma | Base software o hardware su cui sono sviluppate o eseguite applicazioni. |