

**Nome Progetto: DressMe**  
**Titolo Documento: System design document**



Sommario

|  |    |
|--|----|
| MANAGER:                                   | 3  |
| REVISION HISTORY:                          | 3  |
| SOMMARIO                                   | 2  |
| 1 OBIETTIVI DESIGN                         | 4  |
| 1.2 Compromessi                            | 5  |
| 1.3 DEFINIZIONI, ACRONIMI ED ABBREVIAZIONI | 5  |
| 1.4 RIFERIMENTI                            | 5  |
| 1.5 PANORAMICA                             | 5  |
| 2.2Decomposizione del sistema              | 6  |
| 2.2.1 Gestione Sottosistema                | 6  |
| 2.3 HARDWARE/SOFTWARE MAPPING              | 7  |
| 2.4 GESTIONE DEI DATI PERSISTENTI          | 8  |
| 2.5 CONTROLLO DEGLI ACCESSI E SICUREZZA    | 9  |
| 3 Subsystem service                        | 12 |

# SYSTEM DESIGN DOCUMENT

Manager:

| Professori            |
|-----------------------|
| Prof. De Lucia Andrea |

Partecipanti:

| Nome                | Matricola  |
|---------------------|------------|
| Luigi Emanuele Sica | 0512109540 |
| Francesco Ciccone   | 0512108238 |
| Emanuele Riccardi   | 0512107254 |

Revision History:

| Data       | Versione | Descrizione                    | Autore  |
|------------|----------|--------------------------------|---|
| 20/10/2021 | 1.0      | Prima stesura del documento    | Team members  |
| 04/11/2021 | 2.0      | Stesura del documento          | Luigi Emanuele Sica<br>Francesco Ciccone<br>Emanuele Riccardi                     |
| 16/12/2021 | 3.0      | Revisione del documento        | Luigi Emanuele Sica<br>Francesco Ciccone<br>Emanuele Riccardi                     |
| 20/1/2021  | 3.1      | Ultima revisione del documento | Luigi Emanuele Sica<br>Francesco Ciccone<br>Emanuele Riccardi<br>Emanuele Iannone |
| 02/02/2021 | 3.2      | Ultima revisione del documento | Luigi Emanuele Sica<br>Francesco Ciccone<br>Emanuele Riccardi<br>Emanuele Iannone |
| 15/02/2021 | 3.3      | Ultima revisione del documento | Luigi Emanuele Sica<br>Francesco Ciccone<br>Emanuele Riccardi<br>Emanuele Iannone |

## 1 OBIETTIVI DESIGN

Illustriamo nella seguente tabella gli obiettivi di design per il sistema e le relative priorità (a numeri più bassi corrispondono priorità più elevate). Per ogni obiettivo riportiamo anche l'origine, facendo riferimento, in particolare, all'identificativo del requisito non funzionale ad esso associato.

| Priorità | ID   | Descrizione   | Categoria      | Origine                          |
|----------|------|---|----------------|----------------------------------|
| 1        | DG_1 | <b>Robustezza:</b> Il sistema deve sopravvivere agli input errati degli utenti. In un caso del genere il sistema non deve accettare l'input e notificare l'utente invitandolo a correggere l'errore   | Affidabilità   | RNF_13                           |
| 3        | DG_2 | <b>Sicurezza:</b> Il sistema deve comunicare tramite protocollo HTTPS in modo da assicurare una maggiore sicurezza.   | Sicurezza      | RNF_12                           |
| 2        | DG_3 | <b>Usabilità:</b> Il sistema deve essere facile da utilizzare senza necessariamente consultare la documentazione anche tramite l'utilizzo di un menu contestuale che permette di muoversi all'interno del sito in qualsiasi momento. I contenuti dovranno essere fruibili attraverso dispositivi sia desktop che mobile ed accessibili attraverso un numero ridotto di interazioni. | Usabilità      | RNF_1<br>RNF_2<br>RNF_3<br>RNF_4 |
| 2        | DG_4 | <b>Costi di sviluppo:</b> Bisogna abbattere i costi di sviluppo del sistema   | Costo          | Management                       |
| 1        | DG_5 | <b>Tempi di sviluppo:</b> Bisogna sviluppare il sistema in tempi ridotti  | Costo          | Top Management                   |
| 3        | DG_6 | <b>Prestazioni:</b> Il sistema deve essere in grado di supportare fino a 1000 utenti simultaneamente e conservando le loro informazioni in una struttura dati.  | Prestazioni    | RNF_8<br>RNF_9                   |
| 3        | DG_7 | <b>Tempi di risposta:</b> Il sistema deve elaborare le richieste e produrre output in meno di 2 secondi.  | Prestazioni    | RNF_7                            |
| 2        | DG_8 | <b>Leggibilità:</b> Il codice prodotto dev'essere semplice da comprendere. Ogni metodo e campo non banale dev'essere documentato opportunamente al fine di aumentarne la comprensione   | Sopportabilità | RNF_15                           |

## 1.2 Compromessi

### Tempo di rilascio vs Funzionalità

Il team si impegna nel consegnare il sistema completo di tutte le sue funzionalità nei tempi stabiliti tenendo conto di un possibile ritardo nella consegna.

### Prestazioni vs Costi

Il team cercherà di ottenere le migliori prestazioni per non sforare il budget a disposizione.

### 1.3 Definizioni, acronimi ed abbreviazioni

- Piattaforma = Applicazione web
- CRUD = Create, Read, Update, Delete

### 1.4 Riferimenti

- Requisiti funzionali: Sezione 3.1 del Problem statement
- Requisiti non funzionali: Sezione 3.2 del Problem statement

## 1.5 Panoramica

Nel documento verranno affrontati l'analisi delle architetture di sistemi simili, la scomposizione in sottosistemi del sistema proposto. Verranno quindi definiti i servizi esposti da ciascun sottosistema.

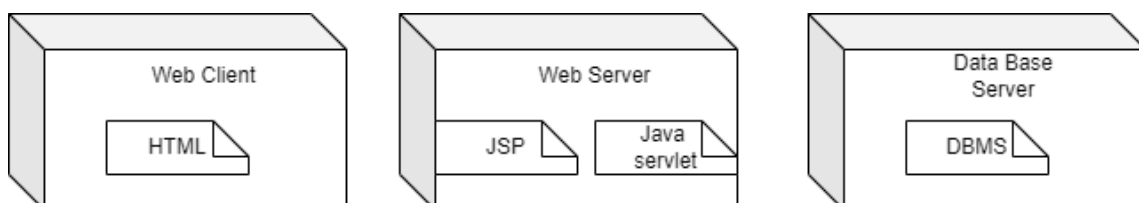
Nel nostro sviluppo il sistema utilizzeremo un'architettura di tipo MVC. IL pattern Model-View-Controller prevede che il software venga diviso in tre parti ognuna delle quali ha un compito diverso.

- **Il model** si occupa della di gestire i dati e quindi sarà responsabile dell'interazione con il database sottostante
- **La view** si occupa di curare l'interazione con l'utente e quindi avrà il compito di gestire la formattazione dei dati che verranno visualizzati.
- **Il controller** dopo aver ricevuto i comandi forniti dall'utente si occuperà di elaborare i dati, passarli al model se necessario e inviare la risposta al view appropriato.

L'utilizzo del modello MVC comporta numerosi vantaggi, tra i quali ricordiamo la possibilità di suddividere il lavoro più facilmente tra i vari componenti del team e la maggiore agilità negli interventi di manutenzione.

All'interno del nostro sistema:

- Il model verrà realizzato utilizzando classi Java appropriate.
- La parte di view verrà implementata utilizzando pagine HTML e JavaScript.
- I control saranno realizzati tramite Servlet.

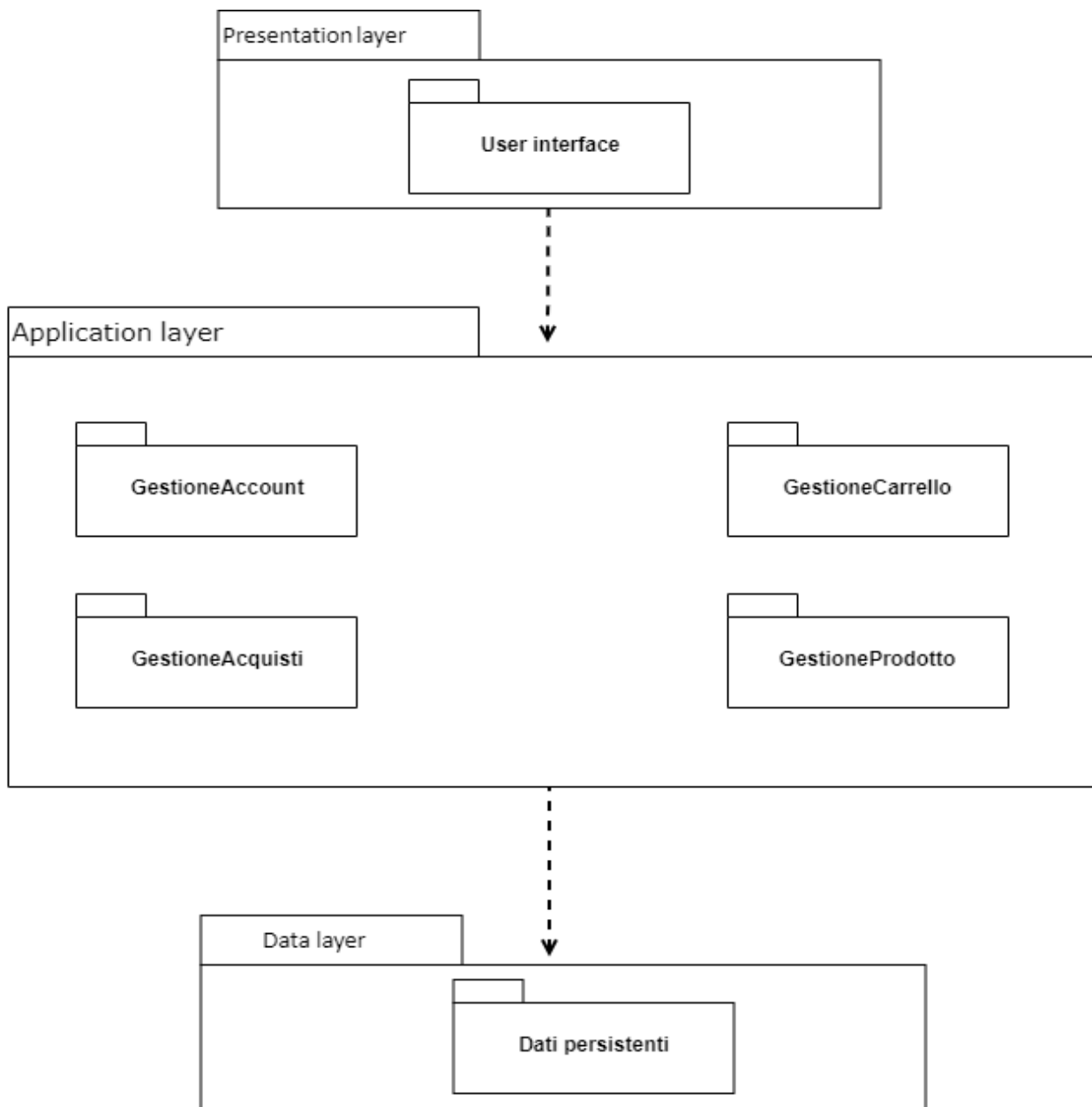


## 2.2 Decomposizione del sistema

Per realizzare il sistema è stata usata un'architettura three-tier. Questo è un caso particolare di un'architettura multi-tier in cui la logica dell'applicazione viene suddivisa in tre layer:

- Presentation layer: composto dalle interfacce grafiche e in particolare dai boundary object come le form che vengono compilate dall'utente (Frontend, ovvero l'interfaccia che gli utenti utilizzeranno per interagire col sistema).
- Application layer: composto dagli oggetti che si occupano della gestione del controllo, dell'elaborazione dati e di notificare i cambiamenti al Presentation layer. Questo strato interagisce con il database sottostante tramite lo data layer.
- Data layer: si occupa della memorizzazione dei dati persistenti e del loro recupero dal database ad esempio eseguendo delle query.

### 2.2.1 Gestione Sottosistema



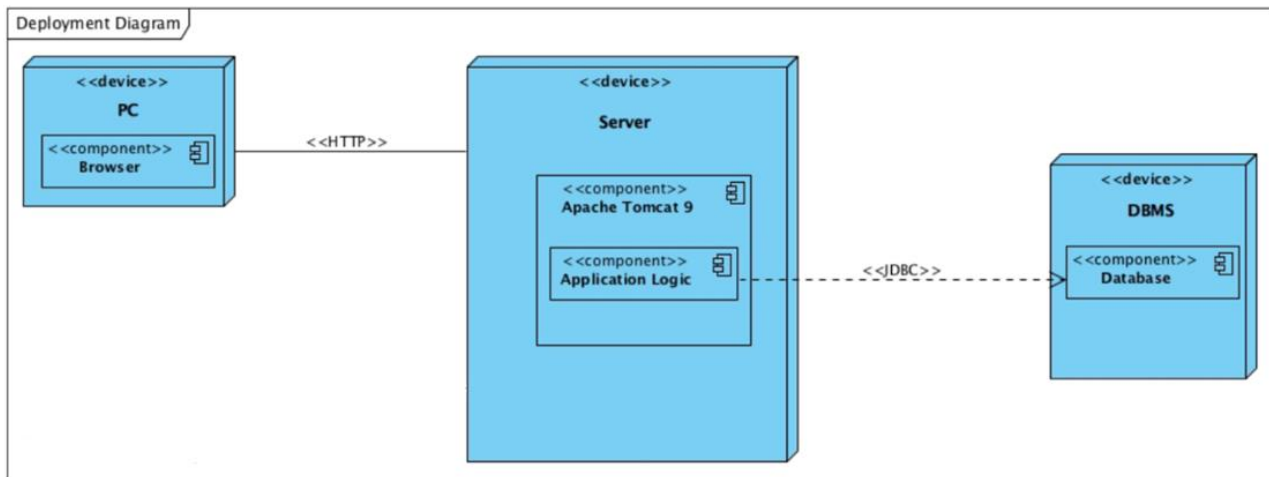
- **User Interface:**
  1. Gestisce le interazioni utente-sistema attraverso le pagine JSP e servlet per i vari servizi offerti dal sistema.
- **GestioneAccount:**
  1. Il cliente può effettuare l'accesso al sistema utilizzando le proprie credenziali e può effettuare il logout. Inoltre, può visualizzare lo storico dei suoi ordini.
  2. l'utente non registrato può effettuare la registrazione al sistema.
  3. l'utente Gestore ordine può effettuare l'accesso al sistema utilizzando le proprie credenziali e può effettuare il logout.
  4. l'utente Gestore prodotti può effettuare l'accesso al sistema utilizzando le proprie credenziali e può effettuare il logout.
  5. l'utente Direttore può effettuare l'accesso al sistema utilizzando le proprie credenziali e può effettuare il logout.
  6. Permette al direttore di aggiungere o rimuovere un dipendente (Gestore ordine o Gestore prodotti).
- **GestioneCarrello:**
  1. Permette al cliente di visualizzare, aggiungere e rimuovere un prodotto al carrello. Consente inoltre di procedere all'ordine.
  2. Permette all'utente non registrato può visualizzare, aggiungere e rimuovere un prodotto al carrello.
- **GestioneAcquisti:**
  1. Permette al cliente di inserire i dati di spedizione, i dati della carta e di confermare l'acquisto.
  2. Permette al gestore ordini di visualizzare o confermare un ordine.
- **GestioneProdotti:**
  1. Il cliente può visualizzare i prodotti presenti nel sito.
  2. L'utente non registrato può visualizzare i prodotti presenti nel sito.
  3. Il Gestore prodotti può visualizzare, inserire, rimuove e modificare un prodotto.

## 2.3 Hardware/software mapping

L'architettura del sistema DressMe è di tipo client/server. Il server riceve le richieste da parte del client, e risponde in tempo utile. I motivi di questa scelta sono:

- **Portabilità:** il sistema potrà essere utilizzato su una varietà di macchine e sistemi operativi, da computer fissi a dispositivi mobili;
- **Flessibilità:** per ogni tipologia di utente che effettua l'accesso al sistema, vi sarà un'interfaccia grafica apposita, tramite la quale ogni attore potrà eseguire le operazioni ad esso riservate.
- **Affidabilità:** entrambi i componenti client e server devono essere affidabili ed essere in grado di mantenere i propri dati anche in seguito a guasti, quindi deve essere possibile effettuare dei backup periodici al database con cadenza

trimestrale.



## Web Server

Il server utilizzato è Apache Tomcat.

## Interface layer

L'utente utilizza il sistema mediante un Browser installato all'interno del suo dispositivo (ad es. Chrome, Firefox).

## Application Logic layer

Il sistema, e quindi le funzionalità, sono implementate in linguaggio Java (Servlet). Il codice in JavaServlet verrà tradotto in linguaggio HTML e JSP, inoltre, il codice risultante viene inviato al browser del client.

## Database Server

Il DBMS usato è MySQL, il quale presenta molte API che permettono l'interazione tra sistema e database.

## 2.4 Gestione dei dati persistenti

Per la gestione dei dati persistenti si è deciso di utilizzare MySQL, questo per due motivi principali:

- I dati sono ben strutturati e per questo si prestano particolarmente ad essere memorizzati in questo DBMS
- Per abbattere ulteriormente i costi, strutturare dei file avrebbe richiesto maggior lavoro e quindi più costi; inoltre, MySQL è gratuito e di conseguenza comporta un costo in meno.

Infine, ovviamente MySQL ci permette di trattare i dati ed effettuare operazioni con estrema semplicità e di aggiungere un ulteriore livello di sicurezza per l'accesso ai dati.

**Le seguenti entità, provenienti dal Class Diagram verranno rese persistenti:**

- Prodotti
- Cliente
- Carrello
- Ordini
- Personale



## 2.5 Controllo degli accessi e sicurezza

Il controllo degli accessi viene gestito attraverso l'inserimento di e-mail e password personali, che garantiranno l'accesso alle operazioni permesse in base al tipo di utente registrato.

Si ricorrerà all'utilizzo della sessione del server per tenere traccia dell'utente loggato. Per questioni di efficienza, la sessione sarà attiva per soli 10 minuti dopo l'ultima interazione dell'utente col sistema. Il salvataggio delle password nel database sarà criptato.

Di seguito viene elencata la Global Access Table che descrive per ogni cella della matrice quali operazioni può effettuare ogni attore su un oggetto.

|                        | Account  | Prodotti  | Carrello  | Ordini   | Acquisti   | Direttore |
|------------------------|--|---|---|--|--|-----------|
| Cliente non registrato | - Registrazione  | -Visualizzare prodotti<br>-Visualizzare dettagli prodotto                             | -Aggiungere prodotto al carrello<br>- Rimuovere prodotto dal carrello                         |  |  |           |
| Cliente registrato     | -Login<br>-Modificare password<br>-Visualizzare storici ordini effettuati<br>-Logout | -Visualizzare prodotti<br>-Visualizzare dettagli prodotto                             | -Aggiungere prodotto al carrello<br>-Rimuovere prodotto dal carrello<br>-Procedere all'ordine | -Generare un ordine  | -Inserisce dati di spedizione<br>-Inserisci dati per il pagamento<br>-Conferma pagamento |           |
| Gestore Ordini         | -Login<br>-Logout  |   |   | - Visualizzare tutti gli ordini<br>- Confermare ordine<br>-Modificare ordine |  |           |
| Gestore prodotti       | -Login<br>-Logout  | -Aggiungi prodotto<br>-Rimuovi prodotto<br>-Modifica prodotto<br>-Visualizza prodotti |   |  |  |           |

|           |                   |                           |  |  |  |   |
|-----------|-------------------|---------------------------|--|--|--|---|
| Direttore | -Login<br>-Logout | -Visualizza<br>dipendenti |  |  |  | -Aggiunta<br>nuovo<br>gestore<br>ordini<br>-Aggiunta<br>nuovo<br>gestore<br>prodotti<br>-Visualizza<br>dipendenti |
|-----------|-------------------|---------------------------|--|--|--|---|

## 2.6 Global software control

Il controllo software globale è di tipo event-driven. Il controllo risiede in un dispatcher che chiama le funzioni mediante callback. Il Web Server si occupa di gestire le richieste effettuate dagli utenti (client). Il server ordina le richieste ai Control (dunque Java Servlet), che gestiranno la richiesta eventualmente interagendo con i Model. Il Web Server aggiornerà poi le View che saranno visualizzate al client attraverso la creazione di codice HTML dalla pagina JSP.

## 2.7 Boundary conditions

Le condizioni limite riguardano l'inizializzazione, la terminazione e le failure del sistema.

### -Inizializzazione del sistema:

**Identificativo:** Start Server

**Attori:** Amministratore

**Flusso di eventi:**

- 1) L'amministratore decide di avviare il server, accede alla macchina Server e preme su 'Start'
- 2) Il sistema, con le opportune procedure di avvio, attiva il server e i relativi servizi in remoto rendendosi disponibile ad eventuali richieste.
- 3) Il sistema notifica il successo della procedura.

**Condizione di uscita:** L'amministratore visualizza la notifica di successo

### -Terminazione del sistema:

**Identificativo:** Stop Server

**Attori:** Amministratore

**Flusso di eventi:**

- 1) L'amministratore decide di arrestare il server, accede alla macchina Server e preme su 'Stop'
- 2) Il sistema riceve la richiesta, effettua le operazioni ancora in corso e spegne il server

**Condizione di uscita:** L'amministratore visualizza la notifica di successo.

### **-Fallimenti del sistema:**

Nel caso di fallimento hardware/software, si procede ad un riavvio del sistema. Per quanto riguarda invece la gestione dei dati persistenti, si procederà ad un backup periodico per minimizzare il rischio di perdita dovuto a guasti al DBMS.

**Identificativo:** Restart Server

**Attori:** Amministratore

**Flusso di eventi:**

- 1) L'amministratore decide di riavviare il server, accede alla macchina Server e preme su 'Reset'
- 2) Il sistema riceve la richiesta, effettua le operazioni ancora in corso e riavvia il server

**Condizione di uscita:** L'amministratore visualizza la notifica di successo.

### 3 Subsystem service

Di seguito verranno descritti I servizi offerti dai vari sottosistemi, con relativa descrizione.

| Sottosistema | Descrizione  |
|--------------|--|
| Account      | Consente ad un Utente non registrato di effettuare la registrazione e a un Utente registrato di effettuare il login.<br>Permette l'inserimento e la rimozione del personale (Gestore ordine o Gestore prodotti) compreso login e logout. |

| Servizio  | Descrizione  |
|---|--|
| login   | Consente ad un Utente di accedere al sistema tramite username e password                           |
| logout  | Consente un Utente, Gestore ordini, Gestori prodotti e Direttore di rimuovere l'accesso al sistema |
| Login Gestore ordine                                | Consente ad un Gestore Ordini di accedere alle rispettive funzionalità protette                    |
| Registrazione Utente                                | Permette ad un Utente non registrato di creare un account nel sistema                              |
| Login Gestore Prodotti                              | Consente ad un Gestore prodotti di accedere alle rispettive funzionalità protette                  |
| Modifica password Utente                            | Consente ad un Utente registrato di modificare la password relativa al proprio account             |
| Modifica dati personale (Gestore ordine e prodotto) | Consente ad un Direttore di modificare le informazioni relative all'account di un personale.       |
| Login Direttore                                     | Consente ad un Direttore di accedere alle rispettive funzionalità protette                         |
| Registrazione personale                             | Consente ad un Direttore di inserire le informazioni relative all'account di un personale          |
| Rimuovi personale                                   | Consente ad un Direttore di rimuovere le informazioni relative all'account di un personale         |

|                                |   |
|--------------------------------|---|
| Visualizzazione storico ordini | Questa funzionalità permette al cliente di visualizzare lo storico degli ordini effettuati. |
|--------------------------------|---|

| Sottosistema | Descrizione   |
|--------------|---|
| Carrello     | Permette di visualizzare il carrello, aggiungere e rimuovere un servizio da quest'ultimo. |

| Servizio                      | Descrizione   |
|-------------------------------|---|
| Visualizza carrello           | Consente di visualizzare i prodotti aggiunti nel carrello e il prezzo totale  |
| Aggiungi prodotto al carrello | Consente di aggiungere un prodotto al carrello  |
| Rimuovi Prodotto dal carrello | Consente di rimuovere un prodotto dal carrello  |
| Procedere All'acquisto        | Permette ad un Utente registrato di iniziare la procedura per completare l'acquisto di un prodotto. Dopo tale operazione il carrello verrà svuotato |

| Sottosistema | Descrizione   |
|--------------|---|
| Prodotto     | Permette l'aggiunta, la modifica e la rimozione di un prodotto da parte di un Gestore prodotti. Consente inoltre la visualizzazione di tutto il catalogo dei prodotti |

| Servizio              | Descrizione  |
|-----------------------|--|
| Aggiunta prodotto     | Permette ad un Gestore prodotti di inserire un nuovo prodotto alla categoria selezionata                       |
| Rimuove prodotto      | Permette ad un Gestore prodotti di rimuovere un prodotto.  |
| Modifica prodotto     | Permette ad un Gestore prodotti di modificare le informazioni relative ad un prodotto precedentemente inserito |
| Visualizzare Prodotti | Permette di visualizzare i prodotti presenti nel sito  |

|  |   |
|--|---|
| Visualizzare Prodotti in base alla categoria | Permette al cliente (registrato e non )di visualizzare i prodotti della categoria selezionata |
|--|---|

| Sottosistema | Descrizione   |
|--------------|---|
| Acquisti     | Consente di inserire i dati di spedizione e i dati di pagamento e ad un Utente registrato per completare l'acquisto, generando un ordine.<br>Permette la conferma degli ordini effettuati dai clienti.<br>Consente inoltre la visualizzazione di tutto il catalogo dei prodotti |

| Servizio                  | Descrizione   |
|---------------------------|---|
| Inserisci dati spedizione | Permette di inserire i dati di spedizione con le relative informazioni del destinatario.  |
| Inserisci Dati Pagamento  | Permette di inserire i dati della carta di credito, necessari per effettuare il checkout  |
| Pagare/genera ordine      | Permette ad un utente registrato di acquistare i prodotti inseriti nel carrello. Dopo tale operazione il carrello verrà svuotato e genererà l'ordine. |
| Confermare ordine         | Permette ad un Gestore ordine di poter confermare un ordine.  |
| Modifica prodotto         | Permette ad un Gestore ordine di modificare le informazioni relative ad un ordine se non ancora spedito   |
| Visualizzare ordini       | Permette di visualizzare tutti gli ordini effettuati non ancora confermati.   |