****

System Design Document

**Versione <1.0>**

****

Sommario

[1. Requisiti Funzionali 3](#_Toc532804043)

[2. Software Architecture 5](#_Toc532804044)

[2.1 Overview 5](#_Toc532804045)

[2.2 Subsystem decomposition 5](#_Toc532804046)

[3.2.1 Schema Generale 6](#_Toc532804047)

[3.3 Hardware/Software mapping 10](#_Toc532804048)

[3.4 Persistent data management 10](#_Toc532804049)

# 1. Requisiti Funzionali

|  |  |
| --- | --- |
| **CODICE** | DESCRIZIONE |
| RF\_1 | Registrazione Utente |
| RF\_2 | Login |
| RF\_3 | Logout |
| RF\_4 | Visualizza Profilo |
| RF\_5 | Visualizza miei prodotti |
| RF\_6 | Visualizza miei ordini |
| RF\_7 | Elimina Account |
| RF\_8 | Cerca un Prodotto |
| RF\_9 | Inserisci Prodotto |
| RF\_10 | Aggiungi prodotto al Carrello |
| RF\_11 | Acquista Prodotto |
| RF\_12 | Svuota Carrello |
| RF\_13 | Login Admin |
| RF\_14 | Elimina Account by Admin |
| RF\_15 | Elimina Prodotto by Admin |

**RF\_1 – Registrazione Utente:** Il sistema permette la registrazione di un nuovo utente tramite la compilazione di una form.

**RF\_2 – Login:** Il sistema permette all’utente di accedere a tutte le funzionalità.

**RF\_3 – Logout:** Il sistema permette all’utente loggato di scollegarsi dal sito, disabilitando le funzioni

concesse all’ utente che effettua il login.

**RF\_4 – Visualizza Profilo:** Il sistema permette all’utente di visualizzare i proprio dati salvati nel sistema.

**RF\_5 – Visualizza i miei prodotti:** Il sistema permette all’utente di poter prendere visione dei suoi prodotti.

**RF\_6 – Visualizza miei ordini**: il sistema permette all’utente di poter prendere visione dei suoi ordini relativi all’acquisto di prodotti.

**RF\_7 – Elimina account:** il sistema permette all’utente, qualora non sia più interessato al sistema di poter cancellare il suo account.

**RF\_8 – Cerca un Prodotto:** il sistema permette all’utente, qualora non sia più interessato al sistema di poter cancellare il suo account.

**RF\_9 – Inserisci prodotto:** il sistema permette all’utente di poter mettere in vendita sulla piattaforma un suo prodotto.

**RF\_10 – Aggiungi prodotto al carrello:** il sistema permette all’utente di aggiungere un prodotto visualizzato nel carrello.

**RF\_11 – Acquista prodotto:** Il sistema permette all’utente di poter acquistare un prodotto precedentemente aggiunto nel carrello.

**RF\_12 – Svuota Carrello:** Il sistema permette all’utente di svuotare il carrello.

**RF\_13 – Login Admin:** Il sistema permette all’admin di effettuare il login come admin.

**RF\_14 – Elimina Account By Admin:** Il sistema permette all’admin di cancellare un utente dal sistema infiniti

**RF\_15 – Elimina Prodotto By Admin:** Il sistema permette all’utente di poter acquistare un prodotto precedentemente aggiunto nel carrello.

# 2. Software Architecture

## 2.1 Overview

L’architettura scelta per il sistema è Three-layer. L’utente può interagire con l’application layer mediante l’interface layer che offre diverse interfacce in base alle necessità dell’utente.

L’application layer dovrà poi comunicare con il database per la memorizzazione dei dati persistenti. Sul server (diverso dal server che contiene l’application layer), risiede un DBMS che si occupa di recuperare, memorizzare ed interrogare i dati presenti nel database, elaborando, quindi, la richiesta dell’utente.

L’aspetto della concorrenza di accessi multipli al database, sarà pertanto gestito dal DBMS stesso che dovrà evitare eventuali colli di bottiglia. Tale architettura conferisce all’intero sistema una maggiore manutenibilità e permette di gestire il problema della concorrenza degli accessi ai dati in maniera semplice ed efficace.

## 2.2 Subsystem decomposition

I tre livelli relativi all’architettura adottata sono:

* IntefaceLayer
* ApplicationLogicLayer
* DataStorageLayer

**InterfaceLayer:**

* si occupa di gestire l’interfaccia utente; include gli oggetti boundary con cui l’utente interagisce (finestre, form, pagine web, ecc..)

**ApplicationLogicLayer:**

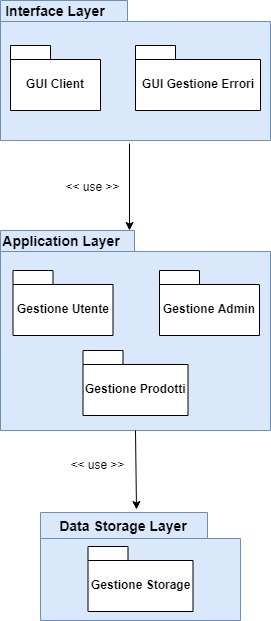
* comprende le componenti logiche (oggetti control e entity), responsabili del corretto funzionamento del sistema, e utilizza il database per effettuare operazioni sui dati.

**DataStorageLayer**:

* comprende un gestore che si occupa di effettuare operazioni di inserimento, cancellazione e aggiornamento e si occupa di rendere disponibili i dati presenti all’interno del database.

### 3.2.1 Schema Generale

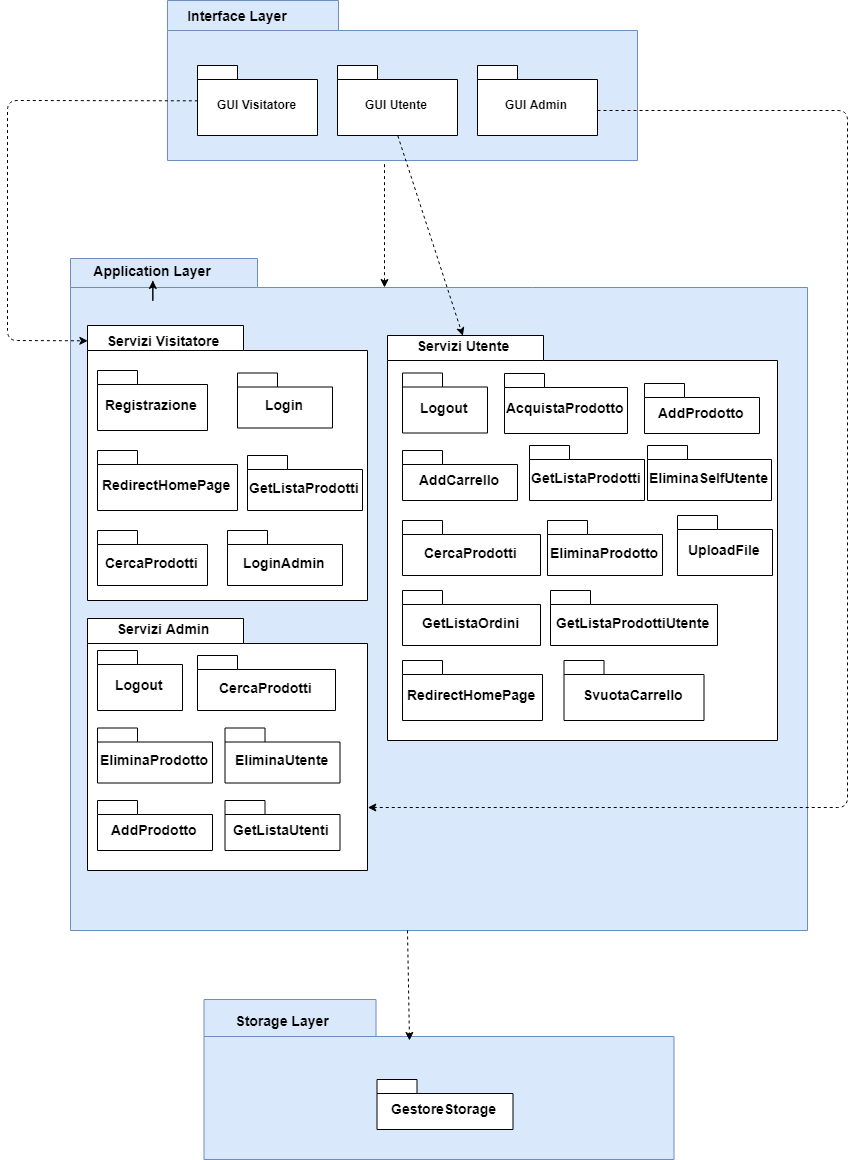
Di seguito è riportato lo schema generale della suddivisione. Successivamente verranno descritti tutti i sottosistemi più nel dettaglio.



**InterfaceLayer:** GUIClient e GUIGestioneErrori, indicano i sottosistemi che contengono tutti gli oggetti boundary;

**ApplicationLogicLayer:** contiene i quattro sottosistemi individuati (Gestione Utente, Gestione Admin e Gestione Prodotto) e per quanto riguarda il Gestore degli Errori è insito in ognuna delle gestioni;

**GestoreStorage:** sottosistema che ha il compito di persistere i dati nel database.



**InterfaceLayer:**

Include tutte le componenti dell’interfaccia grafica del sistema che offrono le funzionalità per l’accesso ad Infiniti accessibili da parte di uno o più utenti.

* **GUIVisitatore:** comprende tutte le interfacce dei servizi offerti all’utente visitatore non loggato.
* **GUIUtente:** comprende tutte le interfacce dei servizi offerti all’utente loggato.
* **GUIAdmin:** tramite questa interfaccia, l’amministratore può accedere alle funzionalità rese solo a lui disponibili.

**ApplicationLogicLayer:**

Il sottosistema è decomposto ulteriormente in altri tre sottosistemi:

* **ServiziVisitatore:** sottosistema che contiene le funzionalità per l’accesso al sistema come:

- ***Registrazione*** : operazione che permette all’utente di registrarsi

- ***Login*** : operazione che permette l’autenticazione al sistema

- ***Reset :*** operazione che permette all’utente di resettare la sua password

- ***EliminaAccount*** : operazione che permette all’utente di eliminare il proprio account

- ***VisualizzaAccount*** : operazione che permette all’utente di visualizzare il suo account

* **ServiziUtente:** sottosistema che comprende le funzionalità dell’amministratore del sistema:

- ***Logout*** : operazione che permette all’utente di scollegarsi dal sistema

- ***DisabilitaAccount*** : operazione che permette di disabilitare un utente

- ***VisualizzaListaUtenti*** : operazione che permette di visualizzare la lista degli utenti

* **ServiziAdmin:** sottosistema che comprende le funzionalità dell’amministratore del sistema:

- ***AbilitaAccount*** : operazione che permette di abilitare un utente

- ***DisabilitaAccount*** : operazione che permette di disabilitare un utente

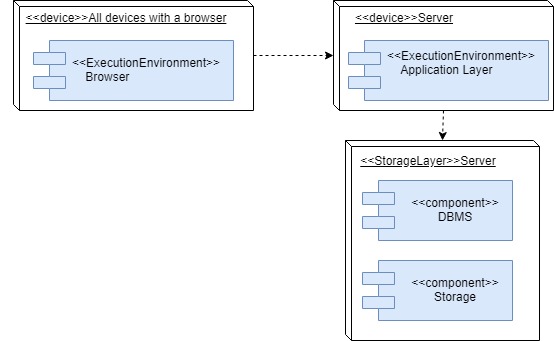
- ***VisualizzaListaUtenti*** : operazione che permette di visualizzare la lista degli utenti

**DataStorageLayer:**

Comprende il ***GestoreStorage*** che permette di effettuare operazioni sul database.

## 3.3 Hardware/Software mapping

Il sistema sviluppato sarà installato su un solo server e utilizzerà un DBMS MySQL stanziato sullo stesso per la gestione dei dati persistenti. Il sistema sarà diviso in un’architettura client e server.



Il Deployment Diagram mostra le componenti che utilizza il sistema; questo diagramma aiuta gli sviluppatori a comprendere le relazioni tra le componenti software e i nodi hardware. Al lato client, l’interfaccia utente verrà mostrata sul browser web ed interagirà con l’application layer, che a sua volta memorizza e interroga i dati presenti sul database. L’application Layer e lo Storage Layer sono collocati su due macchine diverse: L’architettura che si presenta è tree layer – tree tier, ogni layer su una macchina diversa.

## 3.4 Persistent data management

Il sistema userà sia un servizio di storage su file, che servizi di storage su Database. Le risorse saranno opportunamente salvate in cartelle sul server.

La scelta del DBMS è ricaduta su MySQL viste le conoscenze dei team member. La scelta di un database relazionale, rispetto a un database object-oriented, è motivata dalle alte prestazioni offerte dai primi.