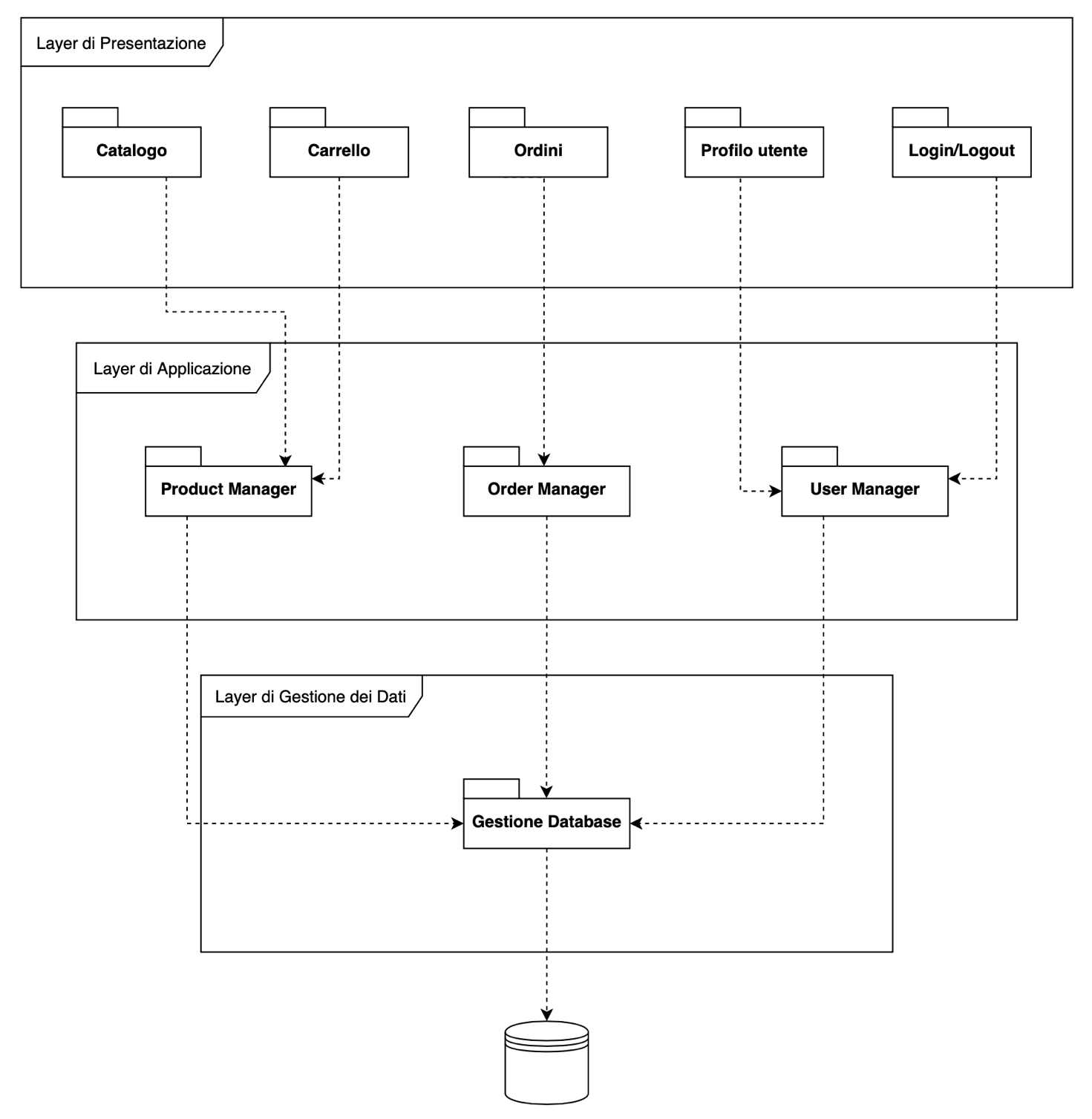
System design document

Design goals

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Priorità | Id design goal | Descrizione | Categoria | Requisito non funzionale | Trade off |
| 1 | DG-1 | Il sito dovrà essere responsive e dovrà adattarsi a diversi dispositivi e risoluzioni | Usabilità | NFR01 | Funzionalità vs usabilità  Trattandosi di un sito di e-commerce si preferisce dare priorità all’usabilità rispetto alla funzionalità |
| 1 | DG-2 | Il sistema dovrà essere in grado di gestire input errati dall’utente, cercando di far correggere l’immissione dei dati | Affidabilità | NFR02 | Affidabilità vs costo  Nonostante risulti essere più dispendioso, si preferisce utilizzare meccanismi di controllo dell’inserimento dei dati in input al fine di evitare errori |
| 1 | DG-3 | Il sistema dovrà essere in grado di gestire manomissioni tramite sql injection | Affidabilità | NFR03 |  |
| 2 | DG-4 | Il sistema utilizzerà un protocollo per la comunicazione sicura attraverso una rete di computer utilizzato su Internet (https) | Affidabilità | NFR04 |  |
| 3 | DG-5 | Il sistema provvederà a crittografie i dati sensibili dell’utente tramite crittografia MD5 | Affidabilità | NFR05 |  |
| 2 | DG-6 | Il sistema dovrà garantire un tempo di risposta relativamente basso (<4s) anche in condizioni di traffico elevato | Prestazioni | NFR06 | Prestazioni vs modificabilità  Trattandosi di una piattaforma web si preferisce dare priorità alla performance piuttosto che alla logica del software |
| 2 | DG-7 | Per inserire un nuovo prodotto l’amministratore deve impiegare meno di 3 min | Prestazioni | NFR07 |  |
| 3 | DG-8 | Il sistema dovrebbe sopportare la connessione di 300 utenti simultaneamente, senza subire crolli | Prestazioni | NFR08 |  |

System Decomposition



[Navigation Path Utente Generico -> NPUG]

[Navigation Path Admin -> NPA]

[Navigation Path Gestore Catalogo -> NPGC]

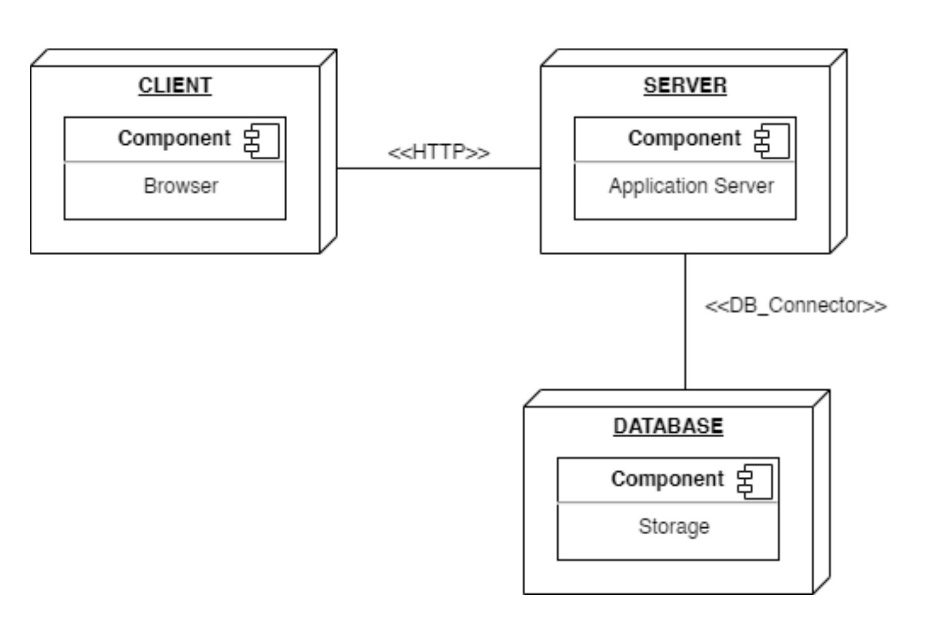
[Navigation Path Gestore Ordini-> NPGO]

Decomposizione del sistema in sottosistemi utilizzando i 3 layer forniti dall’architettura MVC.

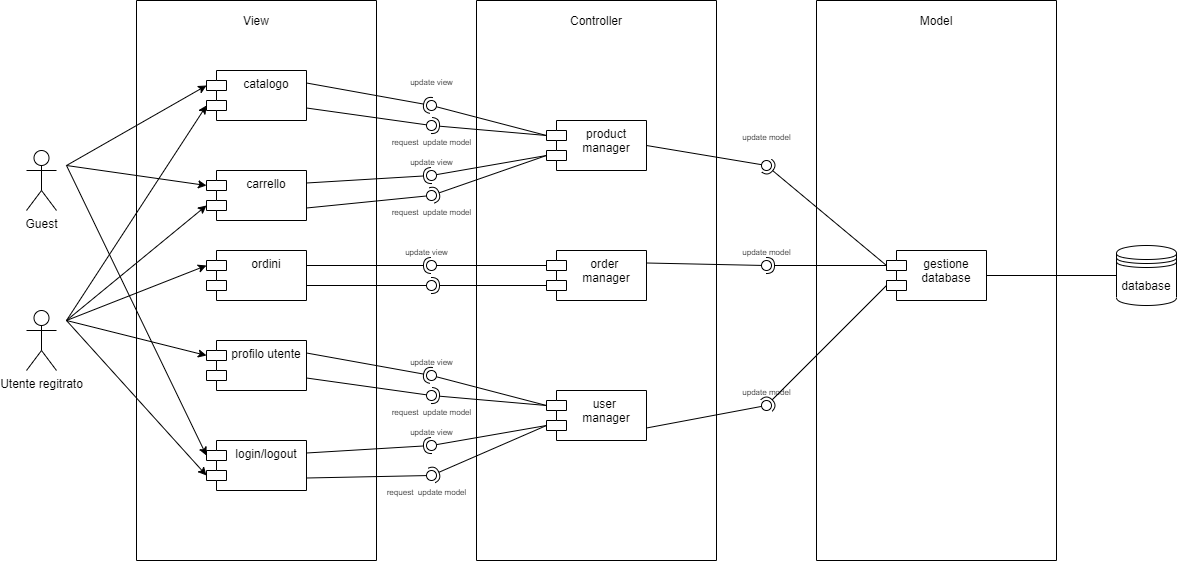
1. **Layer di Presentazione**: livello che si occupa dell’interazione tra utente e sistema attraverso componenti grafiche che rappresentano le funzionalità del sistema
   * **Catalogo**: Sottosistema che raggruppa le interfacce grafiche per:
     + La visualizzazione dei prodotti nel catalogo da parte del **guest** o **cliente**; [NPUG -> Home]
     + La visualizzazione del prodotto singolo da parte del **guest** o **cliente**; [NPUG -> Prodotto]
     + Ricerca dei prodotti da parte del **guest** o **cliente**; [NPUG -> Cerca]
     + Gestione da parte del **gestore del catalogo** della lista di prodotti; [NPGC -> Prodotti]
   * **Carrello**: Sottosistema che raggruppa le interfacce grafiche per:
     + Visualizzazione, modifica della quantità, cancellazione di un prodotto nel carrello
     + L’inserimento dei prodotti scelti dal **guest** o **cliente** nel carrello; [NPUG -> Carrello]
   * **Ordine**: Sottosistema che raggruppa le interfacce grafiche per:
     + La visualizzazione degli ordini effettuati dal **cliente**; [NPUG->Ordini]
     + Visualizzare il dettaglio di un ordine effettuato dal **cliente**; [NPUG->Dettaglio ordine]
     + Stampare la fattura di un ordine effettuato da un **cliente**; [NPUG->Fattura]
     + Gestione da parte del **gestore degli ordini** della lista ordini;   
       [NPGO->Ordini]
   * **Profilo utente**: Sottosistema che raggruppa le interfacce grafiche per:
     + Gestire i dati del **cliente**, carte di credito e indirizzi di spedizione; [NPUG->Profilo utente]
   * **Login/Logout**: Sottosistema che raggruppa le interfacce grafiche per:
     + Login e logout usufruibili da tutti gli **utenti registrati**; [NPUG->Accedi]
     + Recupero password per il **cliente**; [NPUG->Accedi]
     + Registrazione di un nuovo cliente da parte del **guest**; [NPUG->Registrati]
     + Registrazione di un nuovo gestore del catalogo da parte di un **admin**; [NPA->form inserimento nuovo gestore del catalogo]
     + Registrazione di un nuovo gestore degli ordini da parte di un **admin**;[NPA->form inserimento nuovo gestore degli ordini]
2. **Layer di Applicazione**: livello che si occupa della gestione della logica di business e al fine di produrre i risultati da inoltrare al Layer di Presentazione
   * **Product manager**: sottosistema che permette:
     + La gestione dei prodotti (inserimento, modifica, cancellazione);
     + La visualizzazione dei prodotti;
   * **Order manager**: sottosistema che permette:
     + La gestione degli ordini;
     + La visualizzazione degli ordini;
   * **User manager**: sottosistema che permette:
     + Il login e logout;
     + La registrazione di un nuovo cliente;
     + La registrazione di un nuovo gestore ordini e gestore catalogo;
     + Recupero e cambio password;
     + Visualizzare i dati personali del cliente;
     + Gestione dei dati personali del cliente;
3. **Layer di Gestione dei Dati**: Livello che gestisce i dati persistenti necessari al funzionamento di tutto il sistema
   * **Gestione Database**: Sottosistema che ha il ruolo di interagire con la base di dati. DAO:
     + **Prodotto**
       - doSave(product);
       - updateProduct(product)
       - getProduct(int id);
       - deleteProduct(int id)
       - updateQuantità(int id, int q);
       - doRetriveByTipo\_Annata(tipo, annata);
       - doRetriveByTipo(tipo);
       - doRetriveByAnnata(annata);
       - doRetiveByCategory(string cat);
     + **Ordini**
       - doRetiveAllOrderByClient(int id);
       - doRetriveOrderByData(data1, data2);
       - doRetriveOrderByData\_Cliente(data1, data2, int idCliente);
       - doSave(order);
     + **Voce** **Ordine**
       - getFattura(int idOrder);
       - getDettaglioOrdine(int idOrder);
       - doSave(voceOrdine);
     + **Cliente**
       - newCliente(cliente);
       - deleteAccount(int idCliente);
       - passwordRecovery(string email);
       - confirmAccount(int idCliente);
     + **Admin**
       - newGestoreCatalogo();
       - newGestoreOrdini();
     + **Carte di credito** 
       - deleteCreditCard(creditCard);
       - newCreditCard(creditCard);
     + **Indirizzi di spedizione** 
       - newIndirizzoSpedizione(indirizzoSpedizione);
       - deleteIndirizzoSpedizione(int id);
       - updateIndirizzoSpedizione(indirizzoSpedizione);

Hardware software mapping

- Deployment diagram



- Component Diagram



Per una questione di leggibilità del diagramma, invece di utilizzare le specializzazioni dell’utente registrato, abbiamo preferito utilizzare direttamente l’utente registrato. Nel punto precedente (System Decomposition/Layer di presentazione) viene illustrato quale attore interagisce con quale interfaccia grafica.

Data Management

Il tipo di DBMS utilizzato è MYSQL Workbench.

I dati che sono stati resi persistenti sono i seguenti:

1. **Prodotto** (IDProdotto, prezzo, disponibilità, nome, categoria, descrizione, immagine, tipo, annata)
2. **DettaglioOrdine** (IDProdotto, IDOrdine, quantità, prezzo unitario, Iva)
3. **Ordine** (IDOrdine, totale ordine, data, metodo di pagamento, IDIndirizzoSpedizione, IDCartaDiCredito)
4. **IndirizzoSpedizione** (IDIndirizzoSpedizione, nome, cognome, indirizzo, CAP, città, provincia, alias, IDAccount)
5. **CartaDiCredito** (IDCartaDiCredito, nome, cognome, numero, scadenza, CCV, IDAccount)
6. **Account** (IDAccount, nome, cognome, e-mail, password, Stato, tipo)

Tutti i dati sono salvati nel DataBase, tranne l’attributo che indica l’IVA presente nella tabella “Dettaglio Ordine” che è salvato su file.

[Clicca qui](Schema%20relazionale.pdf)