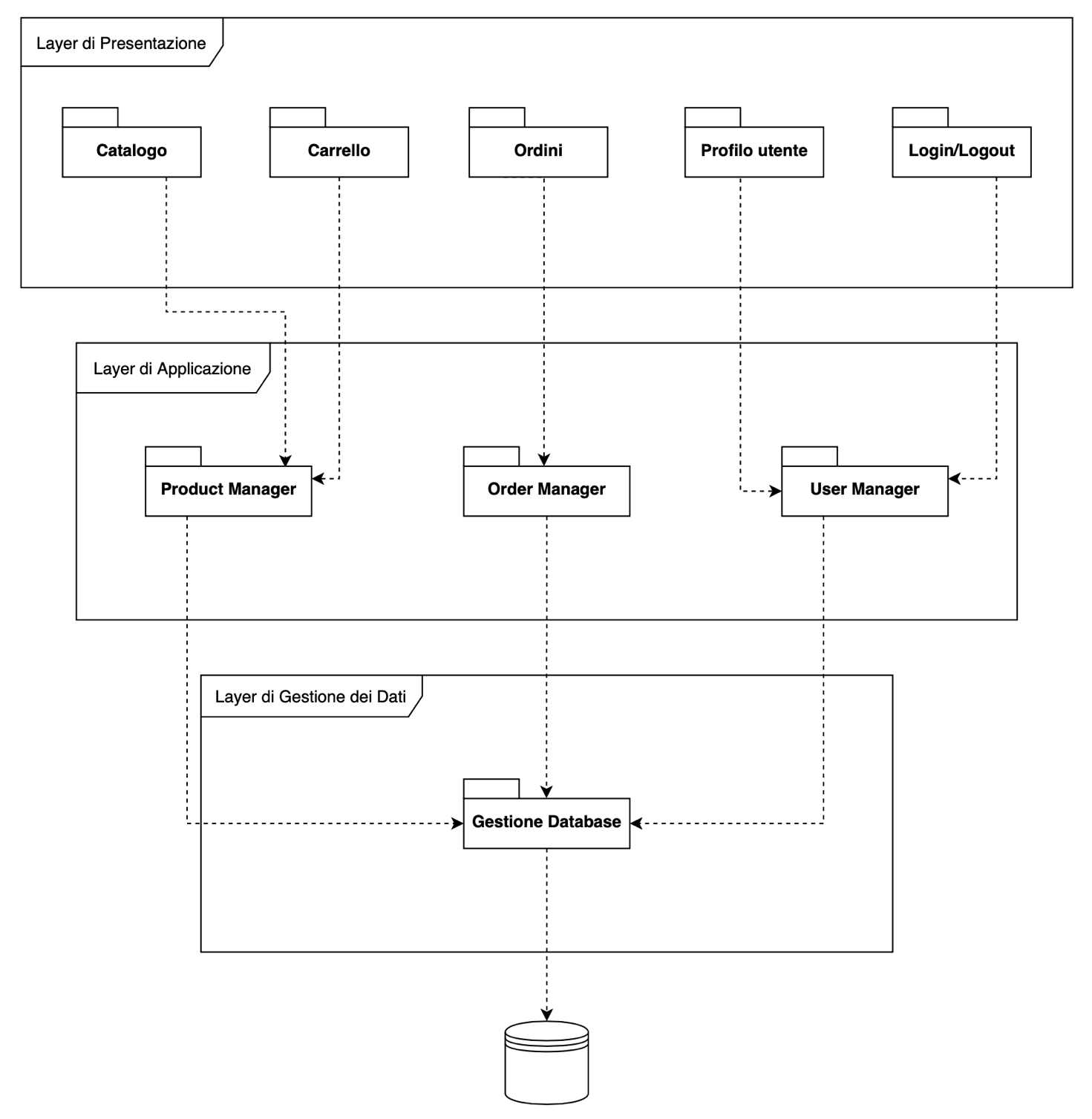
System design document

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 (password) dell’utente tramite crittografia SHA | 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 decomporre il sistema a grana fine |
| 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



Collocazione dell’MVC nel 3-layer: trasversale, ovvero che Model si mappa, in qualche modo, con Business e Data Access, mentre View e Control in Presentation.

[Navigation Path Utente Generico -> NPUG]

[Navigation Path Admin -> NPA]

[Navigation Path Gestore Catalogo -> NPGC]

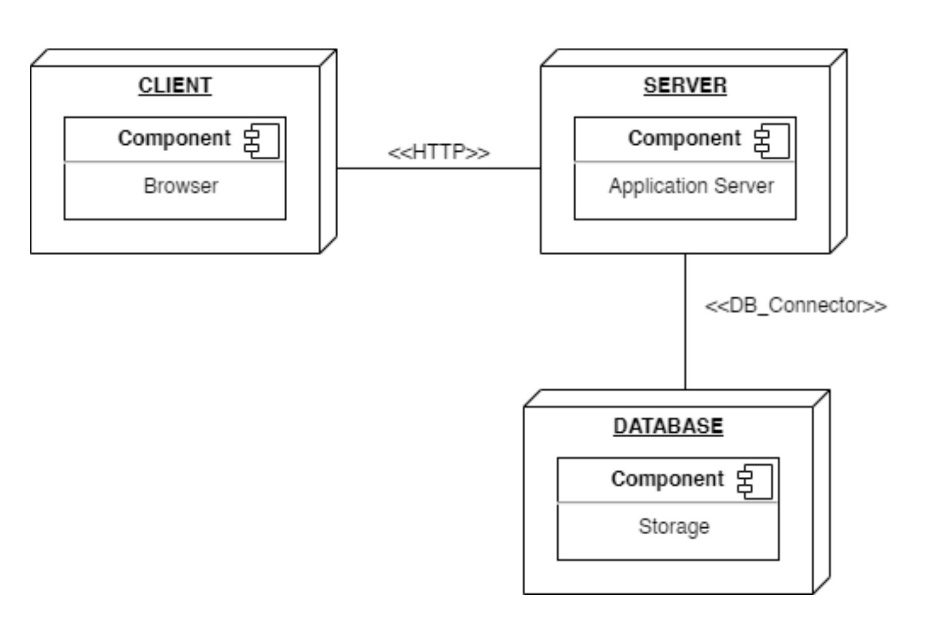
[Navigation Path Gestore Ordini-> NPGO]

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:
     + Inserimento di un prodotto nel carrello
     + Modifica della quantità di un prodotto aggiunto al carrello
     + Rimozione di un prodotto aggiunto al carrello
     + Visualizzazione della suddivisione dei prodotti in categorie
     + Ricerca di un prodotto tramite tipo
     + Ricerca di un prodotto tramite annata
     + Ricerca di un prodotto tramite tipo e annata
     + Acquisto di prodotti
     + Inserimento di un prodotto
     + Cancellazione di un prodotto dal catalogo
     + Modifica di un prodotto
   * **Order manager**: sottosistema che permette:
     + Visualizzazione della fattura di un determinato ordine
     + Visualizzazione dello storico degli ordini di un cliente
     + Visualizzazione del dettaglio di un ordine
     + Visualizzazione ordini complessivi
     + Visualizzazione ordini complessivi dalla data alla data
     + Visualizzazione ordini complessivi dalla data alla data e per cliente
   * **User manager**: sottosistema che permette:
     + Registrazione del cliente
     + Conferma account cliente
     + Registrazione in fase di acquisto
     + Login utente
     + Logout utente
     + Recupero della password utente registrato
     + Eliminazione di un account cliente
     + Inserimento di un nuovo indirizzo di spedizione
     + Modifica indirizzo di spedizione
     + Eliminazione indirizzo di spedizione
     + Inserimento di una nuova carta di credito
     + Eliminazione carta di credito
     + Modifica dei dati di un cliente
     + Login in fase di acquisto
     + Aggiunta nuovo indirizzo di spedizione in fase di acquisto
     + Aggiunta nuova carta di credito in fase di acquisto
     + Inserimento di una nuova carta di credito
     + Eliminazione carta di credito
     + Modifica dei dati di un cliente
     + Login in fase di acquisto
     + Aggiunta nuovo indirizzo di spedizione in fase di acquisto
     + Aggiunta nuova carta di credito in fase di acquisto
     + Registrazione di un nuovo gestore del catalogo da parte di un admin
     + Registrazione di un nuovo gestore ordini da parte di un admin
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



Data Management

Il tipo di DBMS utilizzato è MYSQL.

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)

Global Resource Handling

|  |  |  |  |
| --- | --- | --- | --- |
| Attori | Product Manager | Order Manager | User manager |
| Utente Generico | * Inserimento di un prodotto nel carrello * Modifica della quantità di un prodotto aggiunto al carrello * Rimozione di un prodotto aggiunto al carrello * Visualizzazione della suddivisione dei prodotti in categorie * Ricerca di un prodotto tramite tipo * Ricerca di un prodotto tramite annata * Ricerca di un prodotto tramite tipo e annata |  |  |
| Guest |  |  | * Registrazione del cliente * Conferma account cliente * Registrazione in fase di acquisto |
| Utente registrato |  |  | * Login utente * Logout utente * Recupero della password utente registrato |
| Cliente | * Acquisto di prodotti | * Visualizzazione della fattura di un determinato ordine * Visualizzazione dello storico degli ordini di un cliente * Visualizzazione del dettaglio di un ordine | * Eliminazione di un account cliente * Inserimento di un nuovo indirizzo di spedizione * Modifica indirizzo di spedizione * Eliminazione indirizzo di spedizione * Inserimento di una nuova carta di credito * Eliminazione carta di credito * Modifica dei dati di un cliente * Login in fase di acquisto * Aggiunta nuovo indirizzo di spedizione in fase di acquisto * Aggiunta nuova carta di credito in fase di acquisto |
| Gestore del catalogo | * Inserimento di un prodotto * Cancellazione di un prodotto dal catalogo * Modifica di un prodotto |  |  |
| Gestore degli ordini |  | * Visualizzazione ordini complessivi * Visualizzazione ordini complessivi dalla data alla data * Visualizzazione ordini complessivi dalla data alla data e per cliente |  |
| Admin |  |  | * Registrazione di un nuovo gestore del catalogo da parte di un admin * Registrazione di un nuovo gestore ordini da parte di un admin |

L’autenticazione dell’utente registrato avviene tramite email e password

La password verrà crittografata con tecnologia SHA

Software Control

Il software è caratterizzato da un controllo esplicito centralizzato ed ha un controllo basato su eventi. Il controllo risiede all’interno di un dispatcher che chiama le funzioni tramite callback. Il sistema software è gestito mediante l’uso di Servlet e JSP.

Quindi si ha il layer di applicazione (Controller) che si occupa di rispondere alle richieste del client. Questo layer quindi si comporta da ricevitore di eventi, traduce gli input dell’utente (attraverso le view) in azioni che possono essere eseguite dal model. Successivamente il controller seleziona la view più appropriata per la visualizzazione dei dati stessi.

Il Controller è la componente di Dispatching, di solito si tratta di una servlet che riceve nella request tutti i parametri elaborativi e li distribuisce alle componenti di business, ovvero il Model.

La componente di restituzione del front-end è quella di View, essa prende il risultato dell’elaborazione della componente di Model e la trasforma in stream di output che potrà essere reindirizzata al device chiamante attraverso la stessa servlet che offre servizi di Controller.

Lo schema del modello è il seguente:

Il browser effettua una chiamata alla servlet attraverso una request.

La servlet che offre i servizi di Controller genera oggetti istanze delle classi necessarie a processare la logica di business. Nel frattempo, a causa del servizio richiesto, la servlet è in grado anche di indicare alla componente di View quale visualizzazione utilizzare, ad esempio quale pagina JSP oppure quale file XSL di trasformazione.

La componente di Model effettua i servizi di business ed invia un flusso di dati grezzi alla componente di View, la quale provvederà ad impacchettarli attraverso il framework di visualizzazione indicato dalla servlet di Controller.

Al termine della chiamata sarà la stessa servlet di Controller a restituire, nella response, il flusso trasformato per l’output al device che aveva effettuato la request.