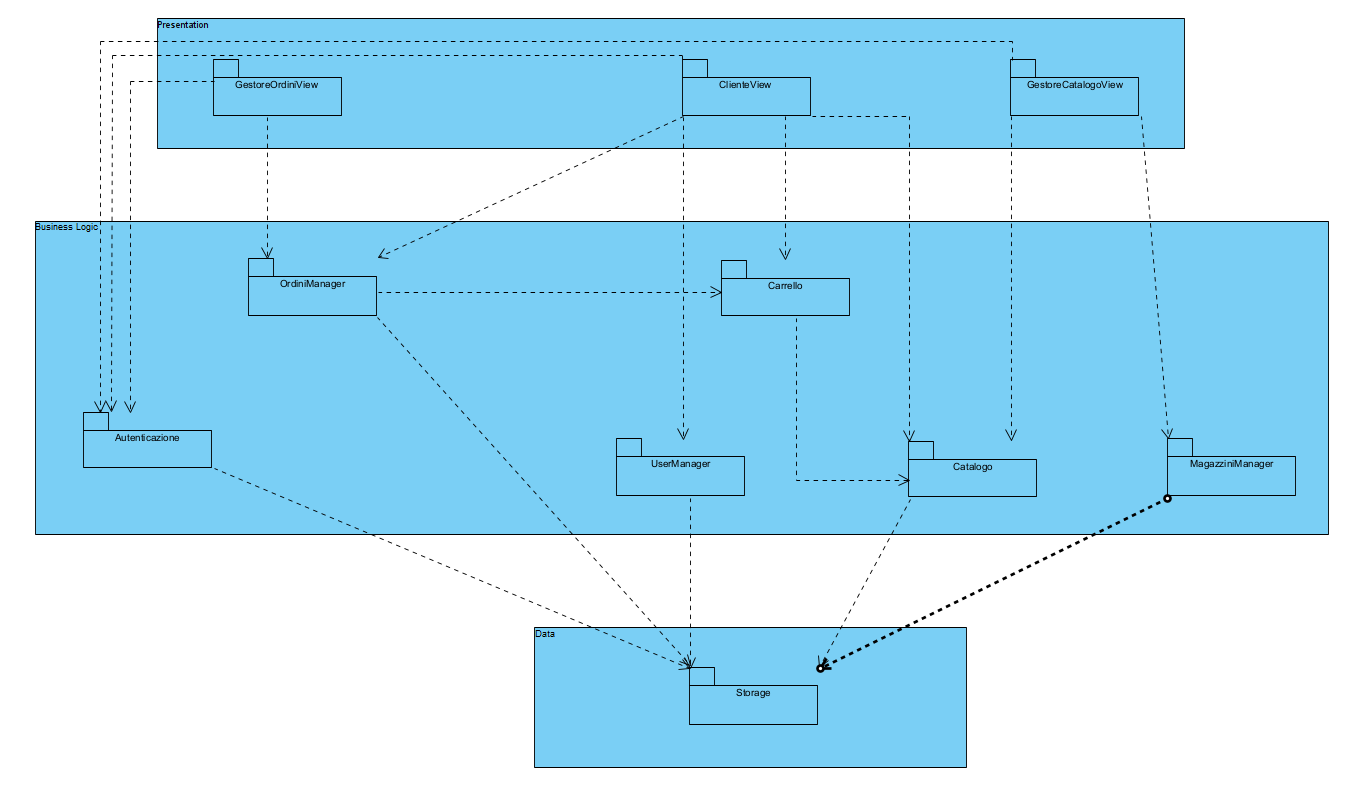
# Architettura del sistema proposto

# 2.1 Decomposizione in sottosistemi

Dividiamo il nostro sistema in 3 layer principali: presentazione, business logic e dati, ogni layer si occuperà di funzionalità diverse. Il layer di presentazione avrà la responsabilità di interfacciare l’utente del sistema alle funzionalità che può utilizzare. La decomposizione in sottosistemi, quindi, è stata fatta in base al ruolo dell’utente finale, dato che a ruoli diversi corrispondono funzionalità diverse.

Il layer di business logic è composto dalle componenti che avranno il compito di gestire il flusso di controllo delle operazioni e di interagire con il database.

Al layer di dati invece troviamo il DBMS che dovrà gestire i dati persistenti presenti all’interno del sistema.

# Sottosistema UserManager

UserManager è il sottosistema che si occupa della gestione delle informazioni dell’utente.

Offre i servizi per la registrazione di un nuovo utente, l’autenticazione di un utente esistente e il servizio che permette di ottenere le informazioni di un utente.

# Sottosistema CatalogoManager

CatalogoManager è il sottosistema che si occupa della gestione delle informazioni dei prodotti presenti nel catalogo.

Fornisce i servizi per la gestione dei prodotti, la gestione dei prodotti nei magazzini e la gestione dei magazzini.

# Sottosistema AcquistoManager

AcquistoManager è il sottosistema che si occupa della gestione delle operazioni relative all’acquisto dei prodotti del catalogo.

Espone quindi il servizio per la gestione del carrello del cliente e quello per la gestione di un ordine.

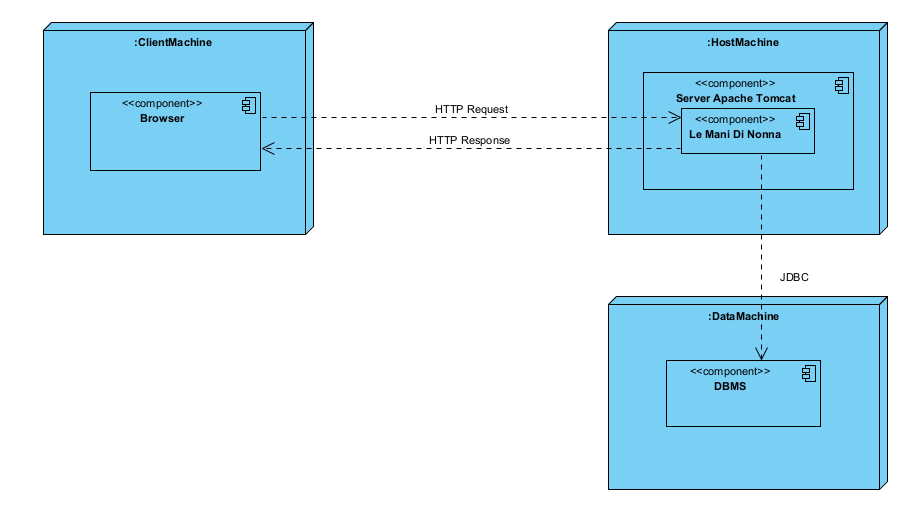
Collabora con il sottosistema UserManager, in quanto ha necessità di conoscere i dettagli del cliente che effettua un ordine e i suoi indirizzi di spedizione, e con il sottosistema CatalogoManager poiché deve conoscere le informazioni dei prodotti acquistati e la loro disponibilità all’interno del magazzino.

# 2.2 Mapping Hardware/Software

L’architettura del sistema è Client-Server con la gestione dei dati persistenti che è stata delegata da un DBMS che può essere installato su una macchina diversa dal Server.

La comunicazione tra il Browser, sulla ClientMachine, e il sistema, sulla ServerMachine, avviene utilizzando delle semplici richieste e risposte HTTP.

Il sistema è distribuito su un server Apache Tomcat presente sulla ServerMachine, e comunica con il DBMS attraverso un driver JDBC.



# Gestione della persistenza

I dati del sistema che devono essere resi permanenti sono:

* Le informazioni degli utenti (compresi gli indirizzi)
* Le informazioni sui prodotti
* Gli ordini
* I magazzini

Per la gestione dei dati persistenti abbiamo scelto di utilizzare un DMBS Relazionale, in particolare MySQL

# Controllo degli accessi

Il controllo degli accessi è illustrato nella seguente matrice degli accessi:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oggetti/  Ruoli | Catalogo | Ordini | Indirizzi | Magazzini | Utente |
| Cliente | Visualizza | Visualizza  Crea | Visualizza  Crea  Rimuovi  Modifica |  | Visualizza  Crea  Rimuovi  Modifica |
| Gestore Ordini |  | Visualizza  Crea  Modifica | Visualizza |  | Visualizza |
| Gestore Catalogo | Visualizza  Crea  Rimuovi  Modifica |  |  | Visualizza  Crea  Rimuovi  Modifica |  |

# Controllo globale del Software

Da chiedere al tutor, il nostro sistema è event-based o procedurale?