# 3 Architettura del sistema proposto

## 3.1 Panoramica

Per realizzare il sistema, proponiamo un’applicazione web, una tipologia di software che meglio rispecchia le idee proposte. L’obiettivo, come ampiamente già discusso più volte è la creazione di un E-commerce per la gestione di un negozio di abbigliamento per la vendita di prodotti online.

Per memorizzare i dati persistenti è stato scelto di utilizzare un database relazionale.

Il sistema è diviso in tre layer, adottando un architettura three-tier: Storage, per la gestione dei dati persistenti, Presentation, per la gestione dell’interfaccia grafica e Controller per la gestione della logica di sistema. I layer sono organizzati secondo il noto pattern “MVC”. Ad un livello di dettaglio maggiore, invece, è possibile identificare ben 11sottosistemi, ognuno dei quali si occupa di una funzionalità specifica, in modo da favorire una alta coesione ed un basso accoppiamento, quindi un cambiamento in un sottosistema non incide sugli altri. Come già più volte detto il sistema prevedrà l’interazione di quattro tipi di utenti: Cliente, Proprietario, Magazziniere e Gestore Marketing, ognuno con le proprie funzionalità specifiche.

## 3.2 Decomposizione in sottosistemi

## 3.2.1 Decomposizione in layer

La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestirne aspetti e funzionalità differenti:

* **View:** gestisce gli elementi dell’interfaccia grafica e gli eventi generati su di essi;
* **Controller:** si occupa della gestione della logica del sistema;
* **Model:** si occupa della gestione e dello scambio dei dati tra i sottosistemi;



## 3.2.2 Decomposizione in sottosistemi

Dopo un’analisi funzionale abbiamo deciso di dividere il sistema nel modo seguente data la necessità di avere un’elevata coesione(poiché le classi del sottosistema effettuano operazioni simili e sono in relazione tra loro) e di un basso accoppiamento(perché eventuali modifiche ad un sottosistema non modifichino gli altri). La suddivisione è stata effettuata secondo la “Decomposizione del Sottosistema in Strati”. In tale suddivisione effettuata abbiamo tre livelli: uno che gestisce la visualizzazione dei dati di un utente, uno che gestisce le operazioni e un altro che immagazzina i dati.

**Primo livello:**

* Interfaccia Cliente: sottosistema che gestisce l’interfaccia di un cliente e che mostra tutte le funzionalità di sua competenza
* Interfaccia Utente non loggato: sottosistema che gestisce l’interfaccia di un utente che non ha ancora effettuato il “Login” e quindi mette a disposizione solo alcune delle funzionalità offerte ad un cliente
* Interfaccia Proprietario: sottosistema che gestisce l’interfaccia del Proprietario e che mostra tutte le funzionalità offerte a quest’ultimo
* Interfaccia Gestore Marketing: sottosistema che gestisce l’interfaccia del Gestore Marketing, in questo caso sono mostrate tutte le operazioni che può compiere questa tipologia di utente
* Interfaccia Magazziniere: sottosistema che gestisce l’interfaccia di un magazziniere, mostrando tutte le funzionalità che gli competono.

**Secondo livello**

* Gestione Cliente: sottosistema che gestisce il profilo di un cliente e le modiche che può apportare ai suoi dati
* Gestione autenticazione: sottosistema che gestisce le regole di autenticazione degli utenti e l’accesso alle varie funzionalità
* Gestione registrazione: sottosistema che gestisce le regole di registrazione degli utenti
* Gestione dipendenti: sottosistema che gestisce le informazioni del negozio, dei prodotti, dei dipendenti e le modifiche su di esse
* Gestione Carrello: sottosistema che gestisce le procedure di acquisto e le eventuali conseguenze

**Terzo livello**

* Database: un singolo sottosistema che gestisce ed immagazzina i dati