***3.2 Decomposizione in sottosistemi***

La decomposizione adottata per il sistema è il pattern MVC composto da tre layers, ognuno dei quali gestisce determinate funzionalità:

* **Controller:** trasforma le interazioni dell’utente nella View in azioni che possono essere eseguite dal Model;
* **Model:** definisce i dati e le operazioni che possono essere eseguite ;
* **View:** gestisce l’implementazione dell’interfaccia grafica , la quale rappresenta il mezzo tramite il quale gli utenti possono interagire col sistema. Dopo attente valutazioni, abbiamo ritenuto opportuno adottare questo tipo di architettura per il nostro sistema poiché lo migliorerebbe sia in termini di efficienza che di manutenibilità.

***Component Diagram***

Nella gestione delle componenti, ogni layer ha in gestione determinate componenti.

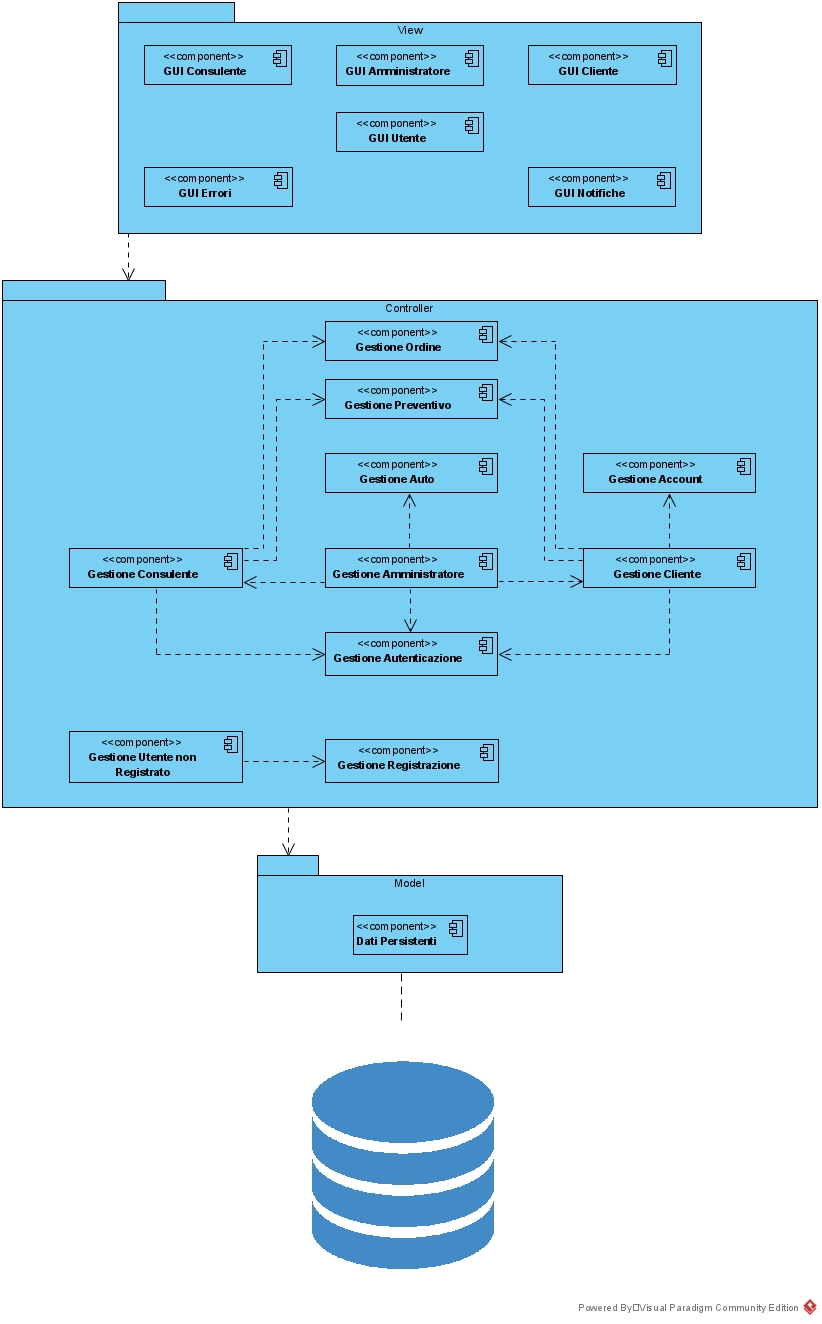
Il layer **View** gestisce 6 componenti:

* GUI Amministratore
* GUI Consulente
* GUI Cliente
* GUI Utente
* GUI Errori
* GUI Notifiche

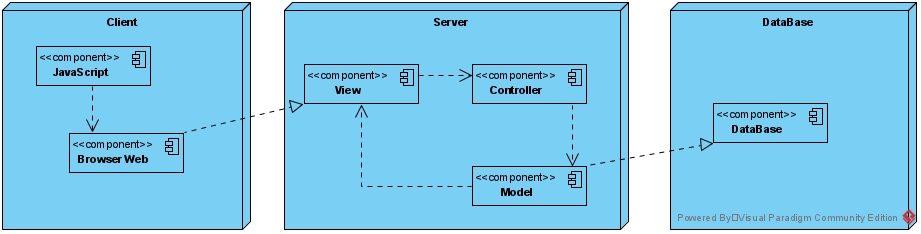
Il layer **Controller** gestisce 10 componenti:

* Gestione Amministratore
* Gestione Consulente
* Gestione Cliente
* Gestione Auto
* Gestione Account
* Gestione Preventivo
* Gestione Ordine
* Gestione Autenticazione
* Gestione Utente non Registrato
* Gestione Autenticazione

Il layer **Model** gestisce una sola componente, cioè quella che si occupa di gestire i dati persistenti nel sistema, interfacciandosi con il DataBase

******

**Deployment Diagram**

L’utente, tramite un browser compatibile con l’interpretazione delle funzionalità di JavaScript, può richiedere le funzionalità del sistema tramite l’interfaccia presente sul Web. Le richieste dell’utente vengono ricevute ed elaborate poi dal sistema tramite il modello MVC, che si occupa di comunicare con il DataBase ed i dati persistenti.