***Software Architecture***

Il software progettato si basa su un’architettura *Client-Server,* a sua volta implementata con uno stile architetturale *Model-View-Controller* (MVC):

* **CONTROLLER**

Gestito dal *backend*, si occupa di astrarre e manipolare il contenuto dei dati del software, oltre a verificare che ogni funzione sia eseguita nel modo corretto; risponde alle esigenze del *client*, convertendone gli input in comandi per il modello e/o le viste*.*

Nel nostro progetto la componente Controller fa riferimento alla classe DataService (situata in GestioneSale\_business\_logic): al suo interno sono contenute tutte le funzioni che verranno poi invocate per eseguire le diverse mansioni del medico (Compilazione e Modifica di Pagina Anagrafica/Operazione/Verbale Medico). Grazie alle funzioni presenti nel DataService, si sono poi potuti definire i casi di test, per lo studio del software.

* **MODEL**

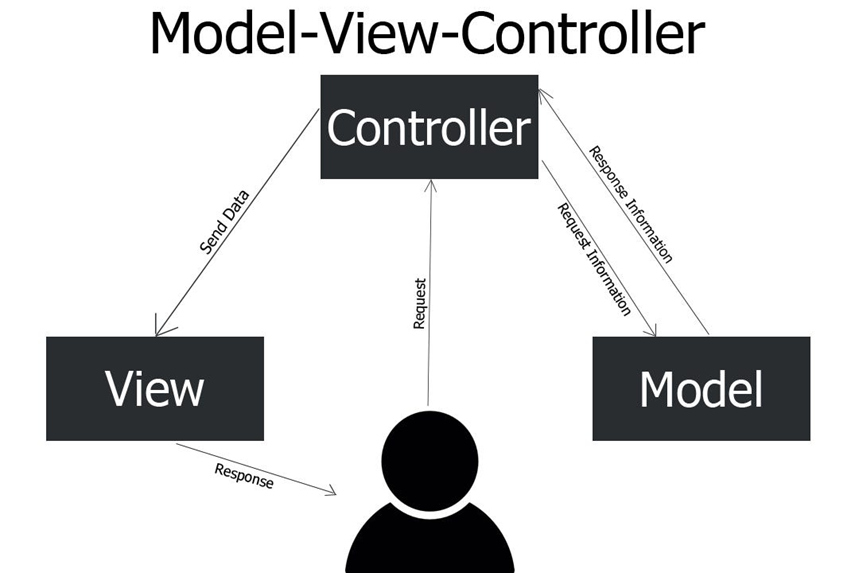
Componente centrale sempre gestito dal *backend*, permette un confronto con il modello preposto al funzionamento dell’intero sistema: lo scopo è creare un’applicazione che risponda a tutti i requisiti di partenza definiti, in termini di dominio del problema; risponde direttamente al *controller*.

Nel progetto il model corrisponde al DataBase, creato con SQLite e situato in GestioneSale\_db\_sqlite: l’accesso ai dati è sicuro e quando viene invocata una funzione del controller, il database deve fornire tutti i dati interessati rispetto all’obiettivo dell’operazione; il recupero di tali dati deve essere veloce e non contradditorio.

* **VIEW**

Gestita dalla parte *frontend,* mostra la visione del software dal punto di vista del *client* (medico).

La view deve usufruire dei servizi proposti da controller e model: quando un utente (medico) entra all’interno della pagina di sistema e richiede di effettuare una nuova compilazione/modifica (di pagine anagrafiche/operazioni/verbali medici), il sistema deve recuperare le informazioni per le azioni dal controller e i singoli dati dal Model



Nell’architettura MVC, l’interfaccia utente è separata dall’applicazione, proprio perché non è facile prevedere gli adattamenti richiesti dall’utente. L’obiettivo è creare un’architettura a servizi, che gestisca le varie richieste (dall’inserimento dei dati anagrafici nel sistema dell’Ospedale fino alla compilazione del verbale medico), il tutto gestito da un solo e unico server; è possibile, tuttavia, che in futuro alcune parti del progetto possano essere utilizzate singolarmente, per rispondere ad uno dei vari servizi proposti dal sistema.

I componenti e connettori del progetto sono legati da *dati condivisi*: poiché bisogna gestire una grande quantità di dati (tra informazioni dei pazienti, dati di operazione e voci del verbale), è necessario fare in modo che l’acceso a queste banche dati sia veloce e non confusionario.