***Software Architecture***

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

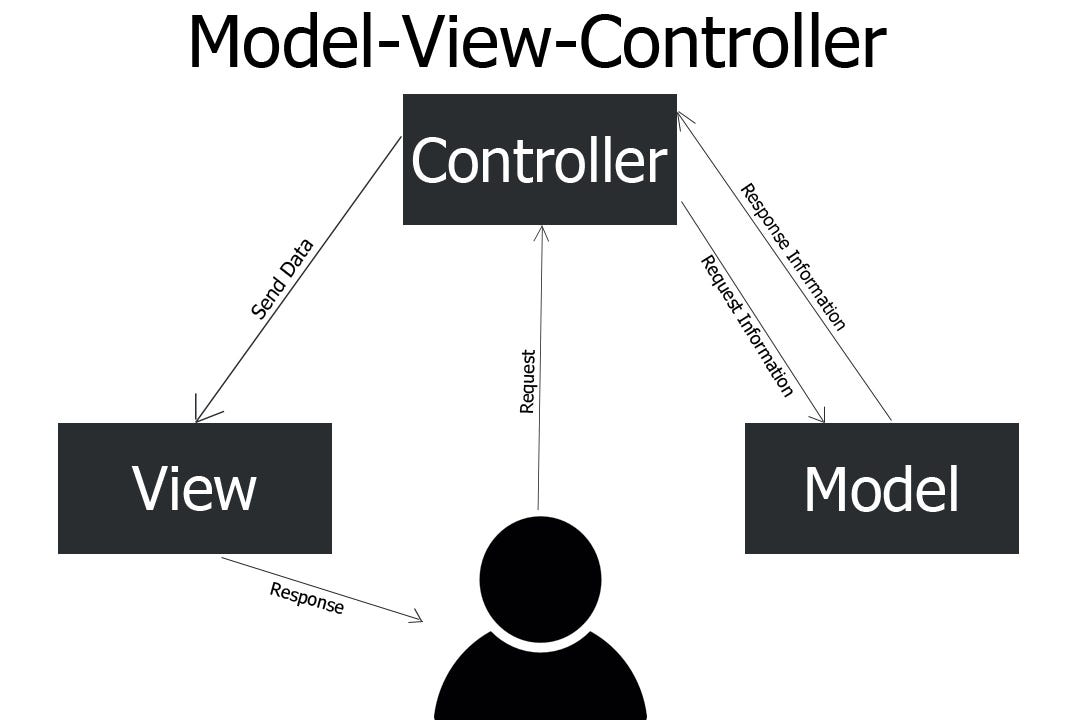
* **Controller:** gestito dal *backend*, si occupa di astrarre e manipolare il contenuto dei dati del software e verificare che ogni funzione sia eseguita nel modo corretto; risponde alle esigenze del *client*, convertendone gli input in comandi per il modello e/o le viste*.*
* **Model:** componente 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; risponde direttamente al controller. Il suo scopo principale è catturare il comportamento dell’applicazione in termini di dominio del problema.
* **View:** gestita dalla parte *frontend,* mostra la visione del software dal punto di vista del *client* (paziente).

Nell’architettura MVC, l’interfaccia utente è separata dall’applicazione, proprio perché non è facile prevedere gli adattamenti previsti dall’utente. L’obiettivo è creare un’architettura a servizi che gestisca le varie richieste (dall’iscrizione al sistema dell’Ospedale fino al verbale medico dell’intervento), il tutto gestito da un solo e unico server; è possibile, tuttavia, che in futuro le varie 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*: dato che bisogna gestire una grande quantità di dati (tra informazioni dei pazienti e dei medici), è necessario fare in modo che l’acceso a queste banche dati sia veloce e non confusionario.

***Componenti****: Memoria*, per ogni utente, che sia medico o paziente, bisogna mantenere una raccolta dati persistenti.

**Connettori:** *Chiamate di Procedure,* il controllo viene trasferito da un componente all’altro, anche attraverso procedure remote

.