

System Design Document

Version 1.0

Top Manager

Prof. Andrea De Lucia

Project Manager

Garreffa Giuseppe

Sommario

[Introduzione 5](#_Toc57810905)

[Scopo del sistema 5](#_Toc57810906)

[Obiettivi di progettazione 5](#_Toc57810907)

[Criteri di performance (RNF\_PERFORMANCE) 5](#_Toc57810908)

[Criteri di affidabilità (RNF\_AFFIDABILITÀ) 5](#_Toc57810909)

[Criteri di costo 5](#_Toc57810910)

[Criteri di manutenzione (RNF\_MANUTENIBILITÀ) 5](#_Toc57810911)

[Criteri di sicurezza (RNF\_SICUREZZA) 6](#_Toc57810912)

[Definizioni, acronimi e abbreviazioni 6](#_Toc57810913)

[Riferimenti 7](#_Toc57810914)

[Panoramica 7](#_Toc57810915)

[Architettura del sistema proposto 8](#_Toc57810916)

[Panoramica 8](#_Toc57810917)

[Decomposizione in sottosistemi 9](#_Toc57810918)

[Mappatura hardware / software 9](#_Toc57810919)

[Gestione dei dati persistenti 10](#_Toc57810920)

[Sicurezza e controllo degli accessi 11](#_Toc57810921)

[Controllo del flusso globale del sistema 11](#_Toc57810922)

[Condizioni limite 11](#_Toc57810923)

[Start-up 11](#_Toc57810924)

[Terminazione 11](#_Toc57810925)

[Fallimento 11](#_Toc57810926)

[Servizi dei sottosistemi 13](#_Toc57810927)

[Interfaccia Utente 13](#_Toc57810928)

[Gestione autenticazione 13](#_Toc57810929)

[Gestione paziente 13](#_Toc57810930)

[Gestione laboratorio 13](#_Toc57810931)

[Gestione medico 13](#_Toc57810932)

[Gestione ASL 13](#_Toc57810933)

[Storage 13](#_Toc57810934)

Partecipanti

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Giuseppe Cirillo | 0512102644 |
| Giuseppe Garreffa | 0512103512 |
| Cristian Alfano | 0512102872 |
| Anna Santoro | 0512103458 |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 28/11/2020 | 0.1 | Creazione Documento | Cristian Alfano |
| 01/12/2020 | 0.2 | Inserimento:  Scopo del sistema  Riferimenti  Panoramica  Decomposizione in sottosistemi  Mappatura hardware / software | Cristian Alfano |
| 02/12/2020 | 0.3 | Inserimento:  Obiettivi di progettazione  Definizioni, acronimi e abbreviazioni  Condizioni limite  Servizi dei sottosistemi | Cristian Alfano |
| 03/12/2020 | 0.5 | Revisione del documento con correzione di errori e aggiunte | Giuseppe Garreffa  Anna Santoro  Giuseppe Cirillo |
| 05/12/2020 | 1.0 | Revisione finale ed accettazione | Giuseppe Garreffa  Alfano Cristian  Anna Santoro  Giuseppe Cirillo |

# Introduzione

## Scopo del sistema

Questo sistema, realizzato in particolar modo per diminuire la frammentazione degli attuali sistemi di prenotazioni tra i vari distretti sanitari delle regioni come ad esempio la Campania, sarà il punto di incontro tra pazienti, medici e laboratori. Attraverso **MyMed System**, si avrà una conveniente alternativa rispetto al doversi fisicamente recare in farmacia o al CUP per la prenotazione di un esame, rispondendo dunque all’esigenza di digitalizzare uno dei settori per il rilancio della pubblica amministrazione e sanità italiana.

Il sistema fornirà la possibilità ai Pazienti di poter prenotare i propri esami e visite in autonomia comodamente da pc e smartphone e di visualizzare i propri referti. Il Medico di famiglia potrà supervisionare facilmente i propri Pazienti attraverso il sistema, controllando direttamente i referti per ciascuno di essi. I Laboratori / Specialisti potranno caricare i referti delle visite ed esami svolti sul sistema e renderli così disponibili per i Pazienti e i Medici di famiglia. L’Operatore ASL sarà in grado invece di gestire l’inserimento dei Medici e dei Laboratori dopo averne controllato le richieste di registrazione.

## Obiettivi di progettazione

### Criteri di performance (RNF\_PERFORMANCE)

* **Tempi di risposta: i**l sistema deve garantire rapidità per eseguire le varie operazioni disponibili, con un tempo di latenza di al massimo 1 secondo per la prima risposta da dare al client.
* **Velocità di risoluzione delle query:** le query devono essere risolte nel minor tempo possibile, preferibilmente minore di 1 secondo; sarà possibile testare i tempi di risposta delle query attraverso MySQL Workbench.
* **Throughput: i**l sistema deve essere in grado di gestire molte richieste concorrentemente senza che la qualità dei servizi offerti venga intaccata. Il DBMS sarà in grado dunque di gestire le query in maniera concorrente e l’architettura del Web Server è di per sé già adatta alla gestione delle richieste dei client.
* **Memoria:** inizialmente, ci si aspetta un carico di dati iniziale che occupa spazio di memoria nell’ordine di decine di Gigabytes. Questo carico aumenterà nel tempo, e si stima che per 500000 utenti, aventi ognuno 5 referti caricati nella propria area e altrettante prenotazioni in media, si possa raggiungere uno spazio occupato intorno ai 2.5-3 Terabytes. È dunque importante dotarsi di dischi rigidi organizzati con la tecnica di installazione *RAID 10*, in modo da aumentare le prestazioni in lettura e scrittura e garantire supporto per i futuri fallimenti dei suddetti dischi.

### Criteri di affidabilità (RNF\_AFFIDABILITÀ)

* **Disponibilità:** il Sistema dovrà essere sempre disponibile ed avere un downtime più basso possibile. L’installazione RAID 10 dei dischi rigidi sarà in grado di reagire ai futuri guasti, mentre sarà necessario poter spostare l’operatività dell’applicazione su un’altra macchina in caso di downtime della principale.
* **Robustezza:** il Sistema dovrà impedire l’inserimento di dati errati o non validi e notificare l’utente.

### Criteri di costo

* **Licenze:** il Sistema farà uso di componenti e librerie rilasciate in forma gratuita e preferibilmente open-source, in modo da minimizzare i costi.

### Criteri di manutenzione (RNF\_MANUTENIBILITÀ)

* **Estensibilità:** il Sistema sarà realizzato in maniera modulare, decomponendo lo stesso in package permettendo dunque l’implementazione di funzionalità future in maniera più semplice.
* **Portabilità:** il Sistema dovrà essere operativo e dipendente su qualunque macchina dotata di Java EE 8 e un DBMS basato su SQL; dovrà essere raggiunto da browser realizzati su smartphone e PC.

### Criteri di sicurezza (RNF\_SICUREZZA)

* **Autenticazione:** il Sistema dovrà garantire l’accesso agli opportuni dati di ogni utente, attraverso la procedura di login.
* **Protocollo di trasmissione:** Il Sistema dovrà comunicare coi client in maniera protetta, attraverso HTTPS realizzato preferibilmente con TLS 1.3.

## Definizioni, acronimi e abbreviazioni

|  |  |
| --- | --- |
| Nome | Descrizione |
| CSS | Cascading Style Sheets |
| DBMS | Database Management System |
| HTML | HyperText Markup Language |
| HTTPS | HyperText Transfer Protocol over Secure Socket Layer |
| JS | JavaScript |
| JSP | JavaServer Pages |
| MVC | Model - View - Controller |
| Open Source | Tipo di software o modello di sviluppo che favorisce la modifica, lo studio, la redistribuzione e l’utilizzo dello stesso. [Maggiori informazioni](https://opensource.org/osd) |
| RAID | Redundant Array of Independent Disks |
| RAID 10 | Il sistema RAID 0 divide i dati equamente tra due o più dischi, tipicamente tramite sezionamento (o striping), ma senza mantenere alcuna informazione di parità o ridondanza che aumenti l'affidabilità.  Il sistema RAID 1, detta anche mirror, mantiene una copia esatta di tutti i dati su almeno due dischi.  Un sistema RAID 1+0 è un RAID che viene usato sia per replicare che per condividere dati tra diversi dischi. (fonte: [Wikipedia](https://it.wikipedia.org/wiki/RAID)) |
| RNF | Requisito Non Funzionale |
| TLS | Transport Layer Security |
| TomCat | Apache TomCat è un’implementazione open-source delle tecnologie Java Servlet, JavaServer Pages e Java WebSocket. (fonte: [Apache TomCat](https://tomcat.apache.org/)) |

## Riferimenti

[[1] Problem Statement](https://github.com/PeppeGUNISA/MyMedSystem/blob/master/deliverables/ProblemStatement_MyMedSystem.docx)

[[2] Requisiti e casi d'uso](https://github.com/PeppeGUNISA/MyMedSystem/blob/master/deliverables/Requisiti%20e%20Casi%20d'uso.docx)

[[3] Requirements Analysis Document](https://github.com/PeppeGUNISA/MyMedSystem/blob/master/deliverables/RAD.docx)

[[4] Open Source Initiative](https://opensource.org/)

[[5] Apache TomCat](https://tomcat.apache.org/)

[[6] Certbot EFF](https://certbot.eff.org/)

[[7] Definizione di TLS 1.3](https://tools.ietf.org/html/rfc8446)

## Panoramica

Il documento corrente mostra i dettagli della progettazione del sistema. Vi sono illustrati gli obiettivi di design del sistema ed è descritta l’architettura software proposta.

Nella sezione “*Architettura del sistema proposto*” viene descritta l’architettura del sistema proposto:

* Decomposizione del sistema in sottosistemi
* Mappatura Hardware/Software
* Gestione dei dati persistenti
* Sicurezza e controllo degli accessi per definire gli accessi e le operazioni permesse da ogni tipologia di utenza
* Controllo del flusso globale di sistema, cioè quali operazioni eseguire ed in che ordine
* Condizioni limite, che descrivono l’avvio e lo spegnimento del sistema.

Nella sezione “*Servizi dei sottosistemi*” vengono specificati i servizi forniti da ogni singolo sottosistema

# Architettura del sistema proposto

## Panoramica

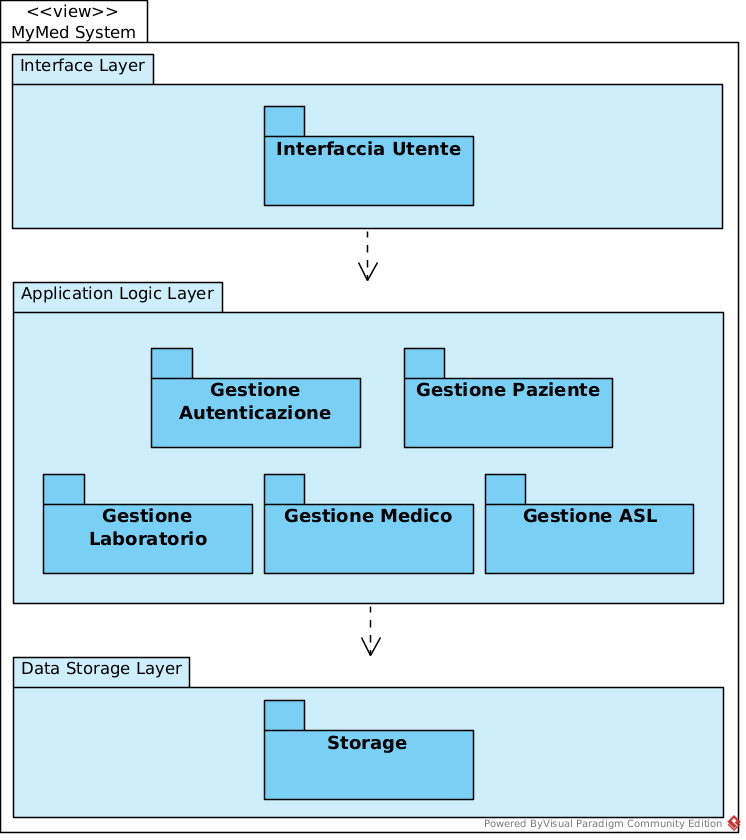
Il Sistema è decomposto in modo tale da poter permettere l’esecuzione in parallelo dei sottosistemi su più macchine. È stata adottato lo stile architetturale Model – View – Controller

I componenti Model rappresentano i sottosistemi a conoscenza del dominio applicativo e forniscono i metodi per accedere ai dati utili dell’applicazione.

I componenti View si occupano di rappresentare le informazioni e i dati richiesti dall’utente.

I componenti Controller informano i componenti View sui cambiamenti del Model e gestiscono le interazioni con l’utente.

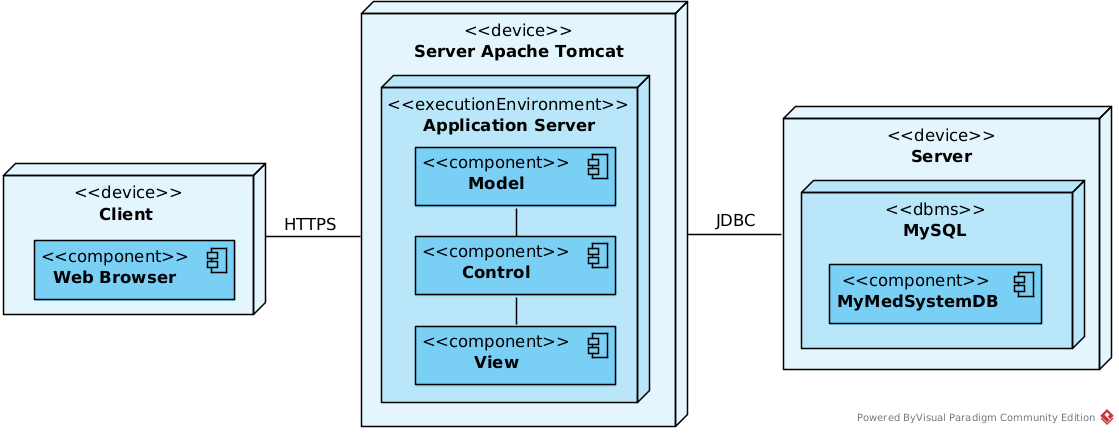
## Decomposizione in sottosistemi



## Mappatura hardware / software

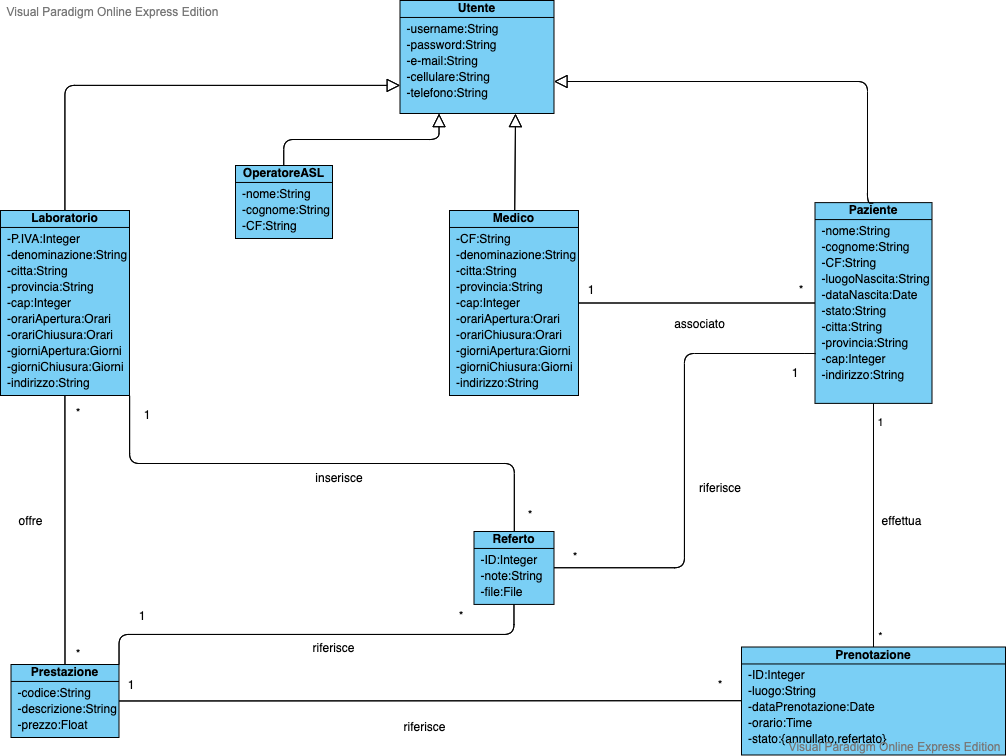
Il sistema sviluppato è sviluppato inizialmente su un solo server e utilizza un DBMS MySql stanziato sullo stesso per la gestione dei dati persistenti che potrà essere separato dal server principale e installato su una seconda macchina; sarà sufficiente definire il link di accesso nel context.xml del server contentente la logica di business. Il client può essere un qualunque dispositivo che abbia un browser installato.

Apache Tomcat 9.0 offrirà la funzione di Web Server, le pagine JSP / HTML realizzeranno le interfacce del sistema e le Java Servlet realizzeranno la logica applicativa. Il protocollo usato per trasmettere dati tra client e server sarà HTTPS, mentre tra il server comunica col database interfacciandosi con le API JDBC.



## Gestione dei dati persistenti

I dati persistenti saranno memorizzati attraverso il DBMS MySql sfruttando il diagramma delle entità.



## Sicurezza e controllo degli accessi

Il protocollo HTTPS sarà certificato da <https://certbot.eff.org/> e sarà abilitato direttamente dai file di configurazione di Tomcat.  
Le connessioni non protette dovranno essere impedite.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Utente | Referto | Prestazione | Prenotazione |
| Paziente | Login()  Registrazione()  Logout() | VisualizzaReferti()  RicercaReferti() |  | CreaPrenotazione()  VisualizzaPrenotazioni()  RicercaPrenotazioni() |
| Medico | Login()  VisualizzaPazienti()  RicercaPazienti()  Logout() | VisualizzaReferti()  RicercaReferti() |  |  |
| Laboratorio | Login()  Logout() | VisualizzaReferti()  InserisciReferto()  ModificaReferto()  RicercaReferto()  EliminaReferto() | InserisciPrestazione()  VisualizzaPrestazioni()  ModificaPrestazione()  EliminaPrestazione() |  |
| Operatore ASL | Login()  Logout()  CreaPersonale()  AssociaPazienti()  VisualizzaPersonale()  RicercaPersonale() |  |  |  |

## Controllo del flusso globale del sistema

Il controllo software globale è di tipo **event-based** ed il controllo risiede in un *dispatcher* che chiama le funzioni mediante *callback*. Il Web Server si occupa di gestire le richieste effettuate dagli utenti (**client**). Il server processa le richieste attraverso i **Control** (dunque Java Servlet), che gestiranno la richiesta eventualmente interagendo con i **Model**. Gli oggetti Control aggiorneranno poi le **View** che saranno visualizzate al client attraverso la generazione di codice HTML dalla pagina JSP.

## Condizioni limite

### Start-up

Per il primo start-up del sistema sono necessari l'avvio del DBMS e il deploy del web server su TomCat 9.

Il deploy consiste nel posizionare il file *MyMedSystem.war* all’interno della cartella *tomcat/webapps* e lanciare il server TomCat.

Dopodiché, il server resterà in ascolto per soddisfare le richieste dei client.

### Fallimento

Possono verificarsi diversi casi di fallimento del sistema:

Nel caso di guasti dovuti al sovraccarico del database con successivo fallimento dello stesso, è prevista come procedura preventiva il salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione del DB.

Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione, non sono previsti metodi che ripristinino lo stato del sistema a uno stato antecedente allo spegnimento inaspettato.

Un altro caso di fallimento potrebbe derivare dal guasto dei dischi rigidi connessi per la gestione del sistema o del database. Il sistema risiederà su dischi configurati con RAID 10 in modo da assicurare le prestazioni e l’eventuale swap di disco in caso di guasto; il DBMS risiederà anch’esso su batterie di dischi rigidi connessi con RAID 10.

# Servizi dei sottosistemi

## Interfaccia Utente

Gestirà l’interfaccia delle pagine presentate all’Utente attraverso l’uso di HTML, JSP, JS e CSS.

## Gestione autenticazione

* **Login:** consente di effettuare l’accesso al Sistema
* **Logout:** consente di effettuare la disconnessione dal Sistema
* **Registrazione:** Consente ad un Paziente di registrarsi al Sistema

## Gestione paziente

* **Crea prenotazione:** consente al Paziente di effettuare una nuova prenotazione
* **Visualizza prenotazione:** consente al Paziente di visualizzare le prenotazioni effettuate
* **Ricerca prenotazione:** consente al Paziente di ricercare le prenotazioni effettuate
* **Ricerca referto:** consente al Paziente di ricercare i propri referti
* **Visualizza referto:** consente al Paziente di visualizzare i propri referti
* **Visualizza info medico:** consente al Paziente di visualizzare le informazioni del proprio medico
* **Modifica dati:** consente al Paziente di modificare i propri dati

## Gestione laboratorio

* **Inserisci referto:** consente al Laboratorio di inserire un nuovo referto
* **Visualizza referto:** consente al Laboratorio di visualizzare i referti inseriti
* **Modifica referto:** consente al Laboratorio di modificare i referti inseriti
* **Cancella referto:** consente al Laboratorio di eliminare un referto inserito
* **Ricerca referto:** consente al Laboratorio di ricercare i referti inseriti
* **Inserisci prestazione:** consente al Laboratorio di inserire una nuova prestazione offerta
* **Elimina prestazione:** consente al Laboratorio di eliminare una prestazione offerta
* **Visualizza prestazioni:** consente al Laboratorio di visualizzare le prestazioni offerte
* **Modifica prestazione:** consente al Laboratorio di modificare una prestazione offerte
* **Modifica dati:** consente al Laboratorio di modificare i propri dati

## Gestione medico

* **Ricerca paziente:** consente al Medico di ricercare i propri pazienti
* **Visualizza paziente:** consente al Medico di visualizzare i propri pazienti
* **Visualizza referto:** consente al Medico di visualizzare i referti dei propri pazienti
* **Ricerca referto:** consente al Medico di ricercare i referti dei propri pazienti
* **Modifica dati:** consente al Medico di modificare i propri dati

## Gestione ASL

* **Inserimento personale:** consente all’Operatore di inserire un nuovo Laboratorio o Medico
* **Visualizza personale:** consente all’Operatore di visualizzare il personale presente nel Sistema
* **Ricerca personale:** consente all’Operatore di ricercare il personale presente nel Sistema
* **Associa pazienti:** consente all’Operatore di associare i pazienti registrati al Sistema ad un Medico

## Storage

Si occuperà di offrire i servizi per la memorizzazione dei dati persistenti ai sottosistemi sopraindicati.