



# System Design Document

## MediCare

Riferimento	C14_SDD.pdf
Versione	2.0
Data	12/13/2023
Destinatario	F. Ferrucci, F. Palomba
Presentato da	Primo Vinicio Calabrese Giovanni Casaburi Matteo Avella Gianluca Palumbo Salvatore Basilicata Domenico Alessandro Urciuoli
Approvato da	Luca Contrasto, Matteo Cicalese



## Team Members

Ruolo	Nome	Acronimo	Contatto
Project Manager	Matteo Cicalese	MC	m.cicalese18@studenti.unisa.it
Project Manager	Luca Contrasto	LC	l.contrasto@studenti.unisa.it
Team Member	Primo Vinicio Calabrese	PVC	p.calabrese17@studenti.unisa.it
Team Member	Giovanni Casaburi	GC	g.casaburi16@studenti.unisa.it
Team Member	Matteo Avella	MA	m.avella17@studenti.unisa.it
Team Member	Gianluca Palumbo	GP	g.palumbo40@studenti.unisa.it
Team Member	Salvatore Basilicata	SB	s.basilicata@studenti.unisa.it
Team Member	Domenico Alessandro Urciuoli	DAU	d.urciuoli2@studenti.unisa.it

## Revision History

Data	Versione	Descrizione	Autori
29/11/2023	0.1	Prima stesura del documento	Gianluca Palumbo
1/12/2023	0.3	Architettura sistema corrente, Decomposizione in sottosistemi,	Domenico Alessandro Urciuoli, Gianluca Palumbo
1/12/2023	0.4	Scopo del sistema, Design Goal,	Matteo Avella
2/12/2023	0.5	Trade – offs, Mapping	Giovanni Casaburi
2/12/2023	0.6	Gestione dati persistenti, Hardware/Software,	Primo Vinicio Calabrese, Salvatore Basilicata, Domenico Alessandro Urciuoli
2/12/2023	0.7	Condizioni Boundary	Gianluca Palumbo, Primo Vinicio Calabrese
2/12/2023	0.8	Panoramica Architettura sistema proposto	Giovanni Casaburi
3/12/2023	0.9	Aggiunta Controllo degli accessi	Giovanni Casaburi, Salvatore Basilicata



Laurea Triennale in informatica-Università di Salerno  
Corso di Ingegneria del Software-Prof.ssa F. Ferrucci - Prof. F. Palomba

12/12/2023	1.0	Revisione	Tutti i team members
22/01/2024	2.0	Preparazione alla consegna	Giovanni Casaburi, Salvatore Basilicata, Matteo Avella, Gianluca Palumbo, Primo Vinicio Calabrese



## Sommario

1. Introduzione .....	6
1.1 Scopo del Sistema .....	6
1.2 Design Goal.....	6
1.3 Trade-offs .....	9
1.4 Definizioni, acronimi e abbreviazioni .....	9
1.5 Riferimenti.....	10
1.6 Panoramica.....	11
2. Architettura del sistema corrente .....	12
3. Architettura del sistema proposto .....	13
3.1 Panoramica.....	13
3.2 Decomposizione in sottosistemi .....	13
3.3 Mapping Hardware / Software .....	16
3.4 Gestione dati persistenti .....	18
3.5 Controllo degli accessi e sicurezza .....	20
3.6 Controllo del flusso globale del sistema .....	21
3.7 Condizioni Boundary .....	21
4. Servizi dei sottosistemi .....	23
5. Glossario .....	26

# 1. Introduzione

## 1.1 Scopo del Sistema

Il software da noi proposto, MediCare, si pone come obiettivo quello di fornire una piattaforma digitale che facilita la gestione e la fornitura di servizi di assistenza sanitaria, migliorando l'efficienza, la qualità e l'accessibilità delle cure mediche.

MediCare ridurrà i tempi di attesa per le visite mediche, oltre a semplificare i processi amministrativi e burocratici di cui si sente molto parlare, soprattutto nel contesto della sanità italiana. I criteri per la protezione dei dati sensibili utilizzati in questo software rispettano quelli del Regolamento Generale sulla Protezione dei Dati (GDPR).

Il nostro sistema si propone non solo come intermediario tra il cittadino e il Servizio Sanitario Nazionale Italiano, ma anche come contributo alla riduzione del carico su quest'ultimo e alla semplificazione delle pratiche amministrative che i cittadini devono affrontare.

Sebbene esistano già web-app simili, come MioDottore (<https://www.miodottore.it/>) o IDoctors (<https://www.idoctors.it/>) nel settore privato, o Sinfonia (<https://sinfonia.regione.campania.it/>) nel settore pubblico, il nostro sistema si differenzierà offrendo la possibilità di interfacciare gli utenti con entrambi gli ambienti, pubblici e privati.

## 1.2 Design Goal

In questo paragrafo vengono esposti i Design Goals (“Obiettivi di Design”) che il sistema punta a raggiungere, i quali consentiranno di delineare il percorso di sviluppo che il progetto seguirà. I design goals sono caratterizzati da una serie di campi, ognuno dei quali corrisponde ad una colonna della tabella, questi sono:

- **Rank:** indica la priorità del design goal, più il numero è basso e più la priorità è alta;
- **ID:** un identificatore univoco;
- **Descrizione:** descrive le caratteristiche del design goal;
- **Categoria:** raggruppa i design goals che descrivono qualità che rientrano nella stessa macroarea;
- **Origine:** il requisito funzionale da cui il design goal è stato generato;
- **Trade-off:** indica una scelta fra due o più parametri, dove la diminuzione di uno costituisce l'aumento degli altri;

Rank / Priorità	ID Design Goal	Descrizione design goal	Categoria	Origine
1	<b>DG_1</b> Privacy	Il sistema garantirà rispetto in materia della legge sul trattamento dei dati personali e privacy, specificatamente del regolamento UE 2016/679 in materia di protezione e trattamento dei dati personali.	Dependability	<b>RNF_LEG.1</b>
2	<b>DG_2</b> Trattamento dei dati	Il sistema chiede l'obbligo del rispetto della legge CDM art.12 Riguardo il trattamento dei dati del cliente da parte del medico.	Dependability	<b>RNF_LEG.3</b>
3	<b>DG_3</b> Riservatezza dei dati	Il sistema chiede l'obbligo del segreto professionale al medico e riservatezza dei dati personali, come da CDM art.10 e art. 11(Codice Deontologico Medico)	Dependability	<b>RNF_LEG.2</b>
4	<b>DG_4</b> Persistenza	Il sistema deve garantire la memorizzazione dei dati tramite un database relazione MySQL	Dependability	<b>RNF_IN.2</b>
5	<b>DG_5</b> Separazione dei ruoli	Il sistema garantisce la distinzione dei ruoli, permettendo alle	End User	<b>RNF_AFF.3</b>

		diverse categorie di utenti di accedere solo alle funzionalità e ai dati per i quali sono autorizzati.		
6	<b>DG_6</b> User Friendly	Il sistema deve essere intuitivo, di facile utilizzo e non ambiguo	End User	<b>RNF_U.1</b>
7	<b>DG_7</b> Manutenibilità	Il sistema deve essere facilmente modulare per garantire la manutenibilità.	Maintenance	<b>RNF_S.1</b>
8	<b>DG_8</b> Reperibilità	Il sistema deve risultare funzionante 24/24, fatta eccezione per eventuali manutenzioni	Dependability	<b>RNF_AFF.4</b>
9	<b>DG_9</b> Piattaforma Web	Il sistema deve essere reso disponibile come piattaforma web e compatibile con tutti i browser	End User	<b>RNF_IM.1</b>
10	<b>DG_10</b> Multiplatforma	Il sistema deve essere facilmente accessibile a tutti i dispositivi con una connessione a Internet.	End User	<b>RNF_PA.1</b>
11	<b>DG_11</b> Estensibilità	Il sistema deve essere estensibile per consentire future aggiunte di nuove funzionalità.	Maintenance	<b>RNF_S.2</b>
12	<b>DG_12</b> Modulo IA	Il sistema deve fornire un modulo d'intelligenza artificiale che riesca a	End User	<b>RNF_IM.6</b>



		prevedere la presenza di malattie cardiache nel paziente		
--	--	--	--	--

## 1.3 Trade-off

Trade-off	Descrizione
Tempo di rilascio Vs Funzionalità	In caso di mancanza di tempo, le funzionalità con priorità più bassa potrebbero non essere implementate, dando maggiore importanza alle funzionalità con una priorità alta.
Tempi di risposta vs Sicurezza	Considerato che il sistema deve gestire dati sensibili e deve preoccuparsi di offrire un servizio con un adeguato livello di sicurezza, i tempi di risposta potrebbero risentirne, andando a rendere alcuni processi più lunghi, per esempio l'autenticazione tramite 2FA.
Tempo di rilascio vs Qualità	Per via della presenza di una scadenza inderogabile, verrà data una maggiore importanza ai tempi di consegna rispetto alla qualità della piattaforma, ciononostante la suddetta qualità rimarrà comunque conforme a determinati standard.
Efficienza vs Compatibilità	Si mira a rilasciare un sistema che non sia altamente performante per fare in modo che sia compatibile con più browser e dispositivi e che sia in grado di adattarsi meglio.
Costi vs Affidabilità	Il sistema presenta costi più elevati rispetto allo standard stabilito, ma punta a garantire una maggiore affidabilità.

## 1.4 Definizioni, acronimi e abbreviazioni

Si elencano tutte le definizioni, acronimi e abbreviazioni presenti all'interno del documento:

- **XHTML:** eXtensible Markup Language
- **HTML:** Hyper Text Markup Language
- **CSS:** Cascading Style Sheets

- **JS:** JavaScript
- **HTTP:** Hyper Text Transfer Protocol
- **GA:** Gestione Autenticazione
- **GAD:** Gestione Admin
- **GF:** Gestione Farmaci
- **GP:** Gestione Prenotazioni
- **GU:** Gestione Utente
- **GAI:** Gestione AI
- **ER:** Entità-Relazione
- **DG:** Design Goal
- **MVC:** Model View Controller
- **SOA:** Service Oriented Architecture
- **ACID:** Atomico, Coerente, Isolato, Durevole
- **MySQL:** My Structured Query Language
- **SQL:** Structured Query Language
- **ASL:** Azienda Sanitaria Locale
- **GDPR:** Regolamento generale sulla protezione dei dati
- **IA:** Intelligenza artificiale
- **FSE:** Fascicolo Sanitario elettronico
- **2FA:** Autenticazione a due fattori

## 1.5 Riferimenti

---

I riferimenti sono:

- Libro, “Object Oriented Software Engineering using UML, Patterns and Java”

Edizione: 3rd

Anno: 2014

Autori: Bernd Bruegge, Allen H. Dutoit

- [Statement of work](#)
- [Requirement Analysis Document](#)

- [Matrice di tracciabilità](#)
- [Test Plan](#)
- [Test Case Specification](#)
- [Spiegazione cosa sia un “cup”](#)
- [Dataset](#)
- <https://regione.campania.it/assets/documents/allegato-1a.pdf>

## 1.6 Panoramica

---

Il presente documento è strutturato in quattro sezioni principali:

- **Introduzione:** in questa sezione del documento troviamo una descrizione dello scopo del sistema, i design goal, i trade – offs ed informazioni circa il linguaggio ed i riferimenti utilizzati.
- **Architettura del sistema corrente:** in questa sezione del documento troviamo l’architettura del sistema attualmente esistente.
- **Architettura del sistema proposto:** in questa sezione del documento troviamo una descrizione accurata dell’architettura del sistema da noi proposto. Tale sezione è a sua volta suddivisa in:
  - decomposizione dei sottosistemi;
  - mapping hardware / software;
  - Gestione dei persistenti;
  - Controllo degli accessi e sicurezza, del flusso globale del sistema;
  - Condizioni boundary
- **Servizi dei sottosistemi:** in questa sezione del documento troviamo la descrizione dei servizi di ogni sottosistema proposto
- **Glossario:** sezione del documento che associa ad ogni sigla / termine menzionato all’intero del documento una descrizione.

## 2. Architettura del sistema corrente

Un sistema attualmente attivo nell'ambito sanitario è la piattaforma Web Sinfonia della regione Campania, qui sotto si illustra la sua architettura:

Il Sistema Sinfonia si colloca, in Regione Campania, in un contesto di informatizzazione regionale in cui l'insieme delle soluzioni applicative ed infrastrutturali per il governo dei processi sanitari risulta essere "variegato" (e complesso) per tecnologie e pluralità di sistemi adottati nell'ambito dei diversi domini ed organizzazioni sanitarie e con una elevata tendenza alla distribuzione dei sistemi sul territorio.

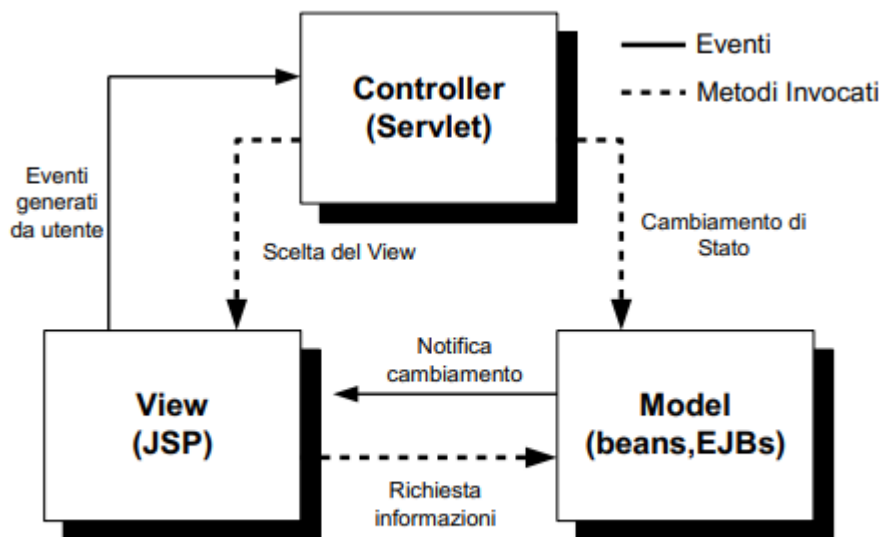
La risposta a tali obiettivi è rappresentata dalla implementazione di soluzioni architetturali orientate ai servizi (SOA), che consentono di realizzare nuovi applicativi ed integrare applicativi esistenti all'interno di un processo di ridisegno complessivo del sistema IT.

Il software applicativo è distribuito sui seguenti tier:

- Presentation Tier
- Business Tier

La generazione dinamica delle pagine XHTML di risposta ed il controllo del flusso delle pagine web generate dinamicamente avviene seguendo il modello definito dal pattern MVC (Model View Controller).

<https://regione.campania.it/assets/documents/allegato-1a.pdf>



**Figura 1 – Il Pattern MVC**

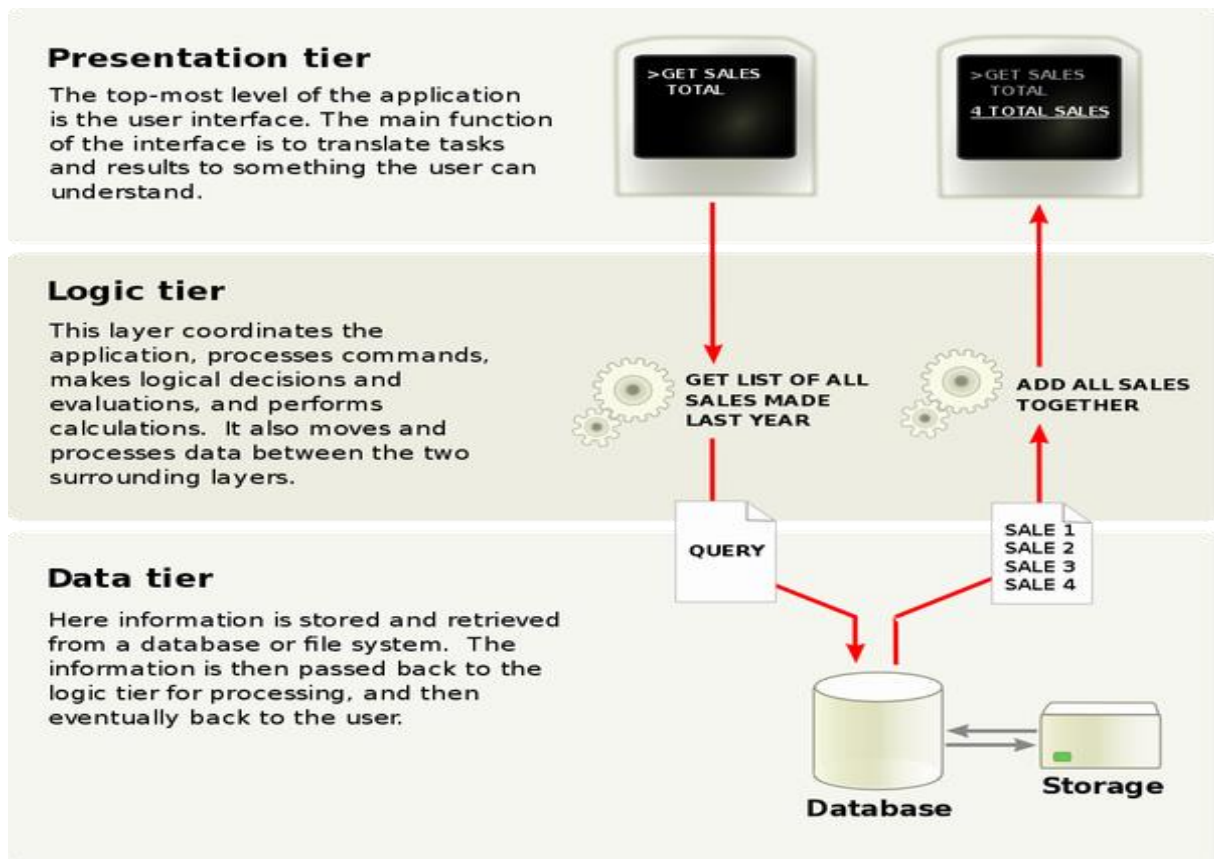
## 3. Architettura del sistema proposto

### 3.1 Panoramica

Medicare ha come obiettivo fornire una piattaforma web dove sia studi privati che enti sanitari pubblici possano gestire liberamente il proprio flusso lavorativo, per gli utenti normali si mira a fornire un'adeguata gestione della propria storia sanitaria e vaccinale con la considerazione delle eventuali esenzioni da reddito.

Il sistema proposto avrà un'architettura a **three-tier** con la classica suddivisione tra Interface layer, Application logic layer e Storage layer. Per quanto la parte di front-end e dell'Interface layer, verrà utilizzato Python, HTML, CSS e JS. La logica di business della piattaforma sarà implementata completamente in Python tramite l'utilizzo dei framework Flask e Jinja.

Per quanto riguarda la persistenza dei dati, fornita dallo Storage layer, verrà utilizzato un database relazionale in particolare il database MySQL al quale sarà possibile collegarsi tramite sqlalchemy.



a

### 3.2 Decomposizione in sottosistemi

Il sistema è stato decomposto nei seguenti sottosistemi:

**Gestione Utente:** questo sottosistema si interessa delle funzionalità della visione di un fascicolo elettronico sanitario, cancellazione dell'account (per il medico privato), inserimento e gestione dell'ISEE.

**Gestione Autenticazione:** questo sottosistema si interessa delle funzionalità di Login (per il paziente tramite 2FA, per il medico (pubblico e privato) e l'Ente sanitario tramite credenziali e), Logout e registrazione sulla piattaforma (per il paziente tramite 2FA, per l'ente e per il medico privato con credenziali proprie).

**Gestione Prenotazione:** il sottosistema si interessa delle funzionalità relative all'inoltro di richieste di prenotazioni di visite mediche, analisi e vaccini e visualizzazione degli storici delle stesse. Si interessa anche delle funzionalità di pagamento delle prestazioni sanitarie.

**Gestione Ente:** il sottosistema si interessa delle funzionalità relativa alla creazione ed eliminazione di un account di reparto.

**Gestione AI:** il sottosistema si interessa delle funzionalità relative alla visualizzazione di una previsione riguardante la diagnosi di una malattia cardiaca.

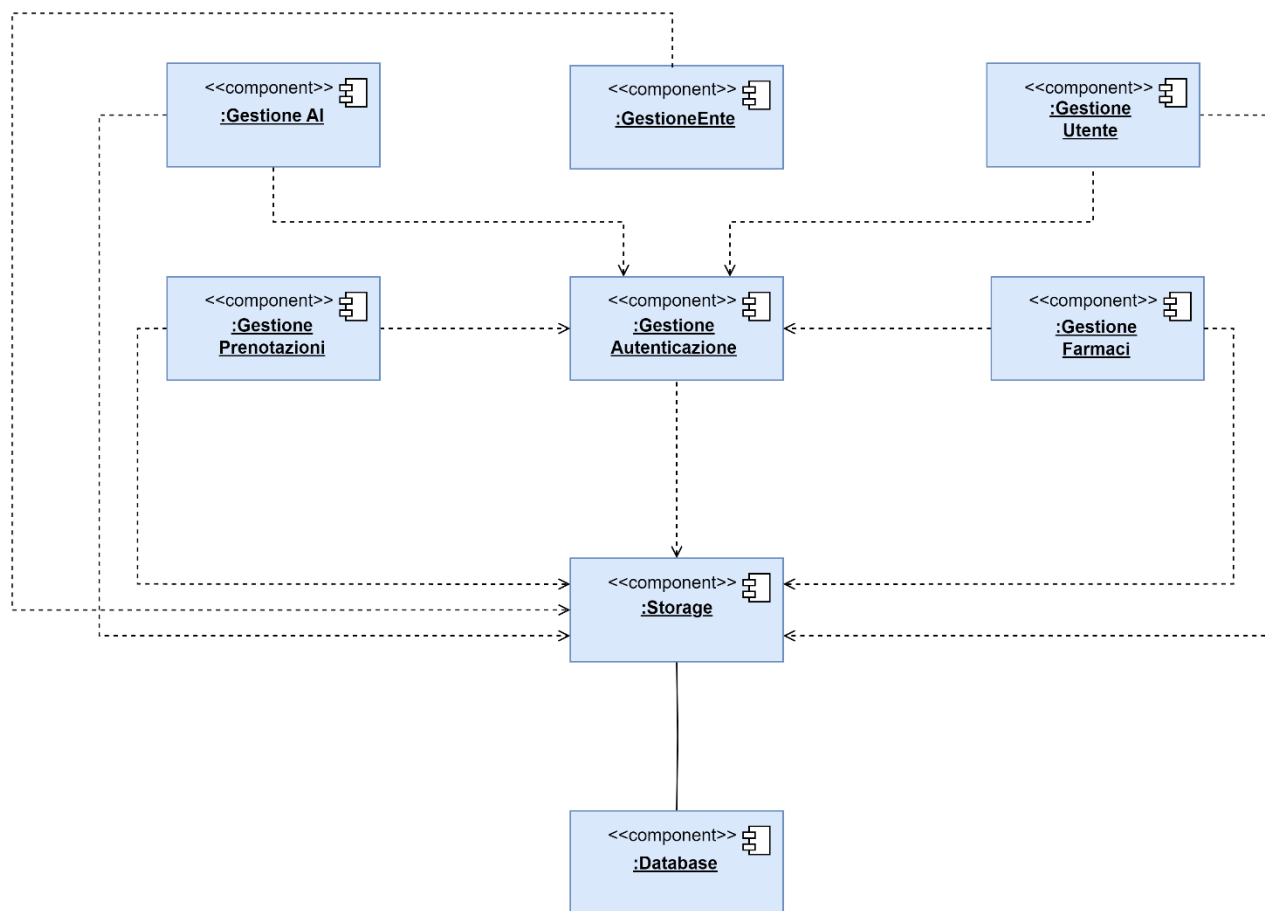
**Gestione Farmaci:** il sottosistema si interessa delle funzionalità relative alla visualizzazione dei farmaci, filtraggio e ricerca degli stessi. Si interessa inoltre della visione dei dettagli di un certo farmaco.

**Gestione Storage:** questo sottosistema si interpone tra i sottosistemi e il sottosistema Database

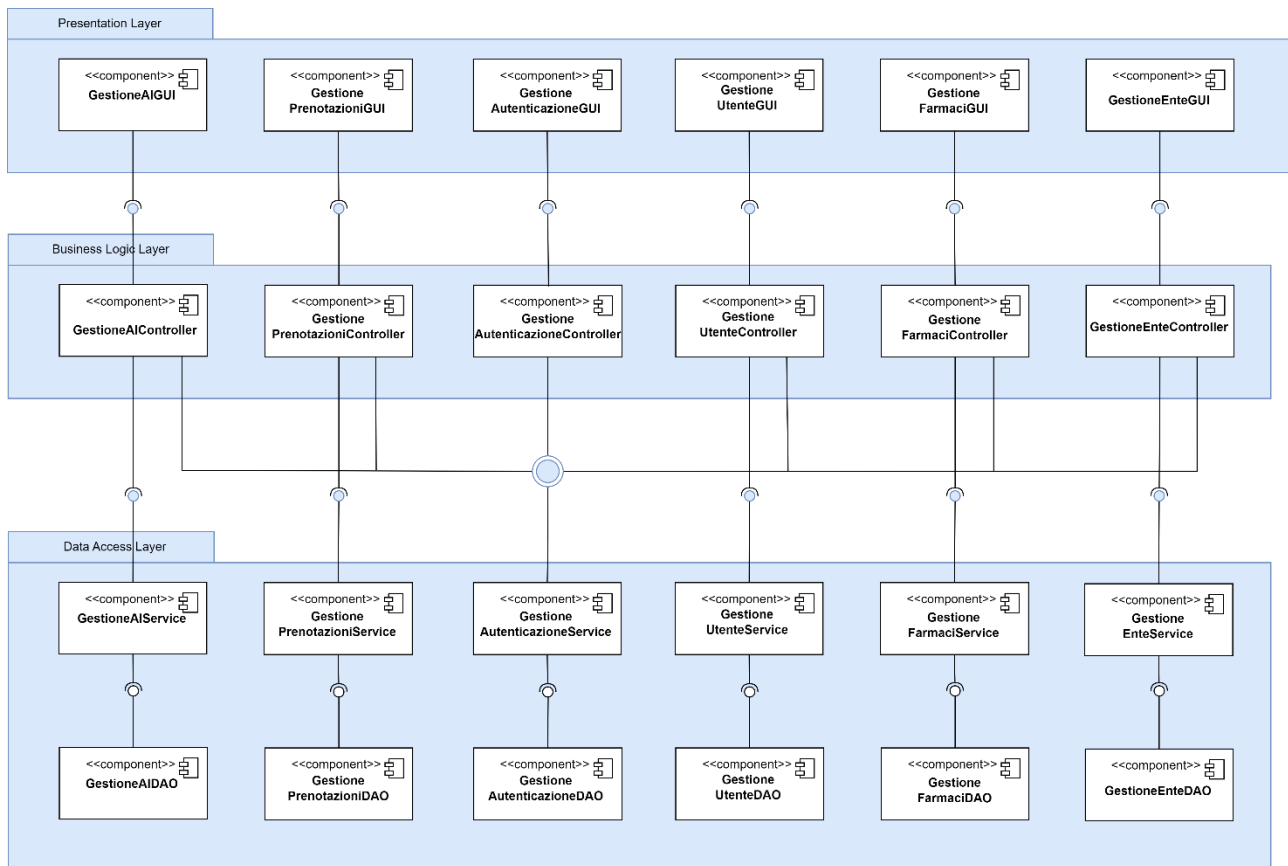
**Gestione Database:** il sottosistema permette di garantire la conservazione dei dati in maniera persistente.

Per ridurre l'accoppiamento tra questi sottosistemi, decidiamo di creare un nuovo sottosistema chiamato Storage, che protegge il database dagli altri sottosistemi, se decidessimo di cambiare il fornitore del database o di utilizzare un meccanismo di archiviazione diverso, è sufficiente cambiare il sottosistema Storage. In questo modo, l'accoppiamento complessivo della decomposizione del sottosistema è stato ridotto.

## Component Diagram UML



## Diagramma Architeturale



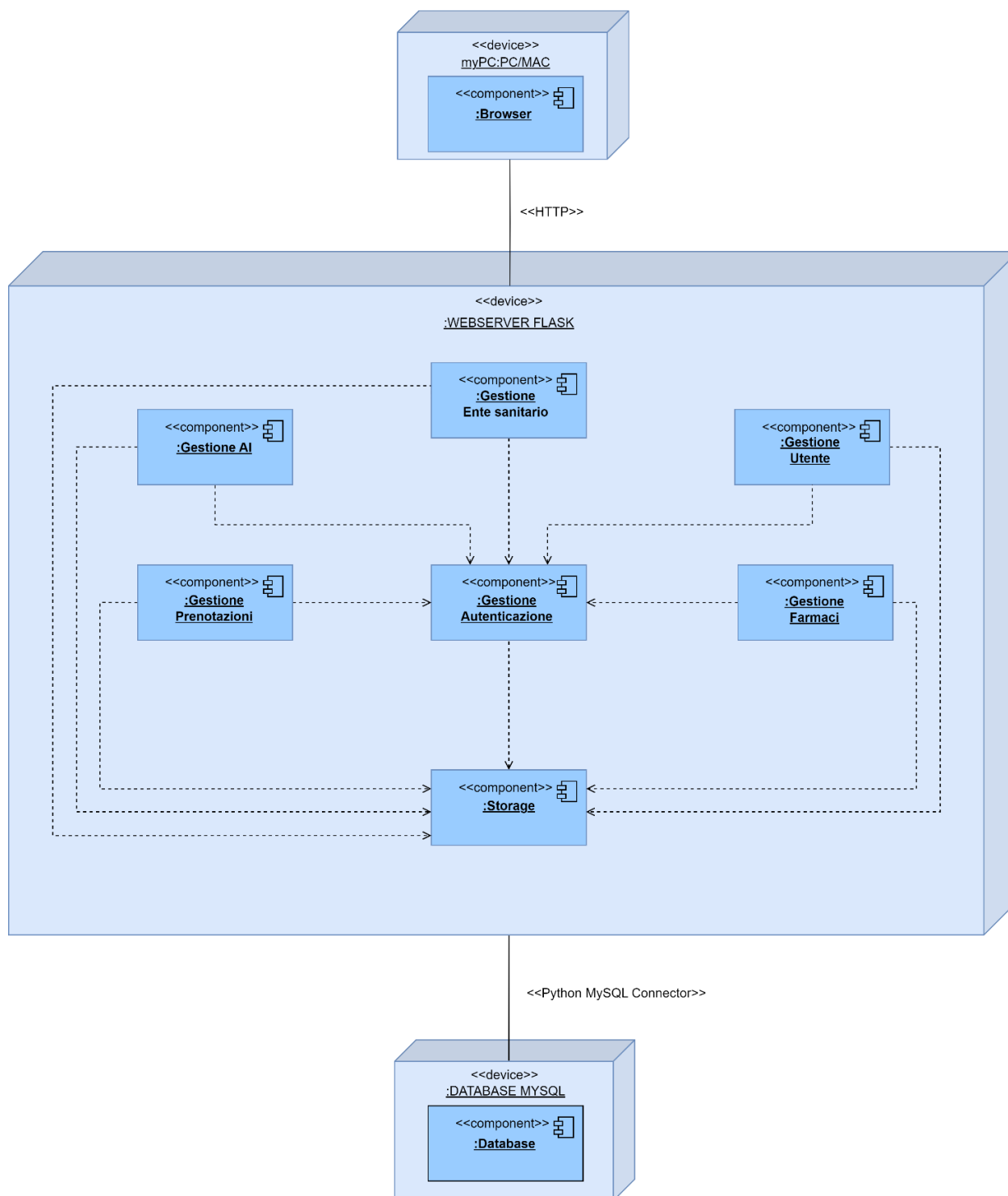
Dal diagramma architettuale si evince che l'architettura è di tipo chiusa, tale scelta è stata presa per raggiungere più facilmente i design goal di sicurezza, garantendo ai vari sottosistemi di accedere alle sole interfacce necessarie per svolgere le loro funzionalità. Un'altra caratteristica di questa architettura è che abbassa significativamente il coupling tra i sottosistemi, favorendo il design goal di manutenibilità.

### 3.3 Mapping Hardware / Software

La piattaforma utilizzerà una struttura basata su architettura di tipo Client-Server. Il dispositivo connesso alla rete internet e che si occupa di erogare i servizi rappresenta il Server, che gestisce la logica di business e la comunicazione con i diversi client. Il dispositivo che, disponendo di un browser e di una connessione internet, può collegarsi al server tramite il protocollo HTTP, rappresenta il Client. I



dati persistenti correlati al dominio di applicazione sono gestiti mediante l'utilizzo del Database Management System MySQL.



## 3.4 Gestioni dati persistenti

---

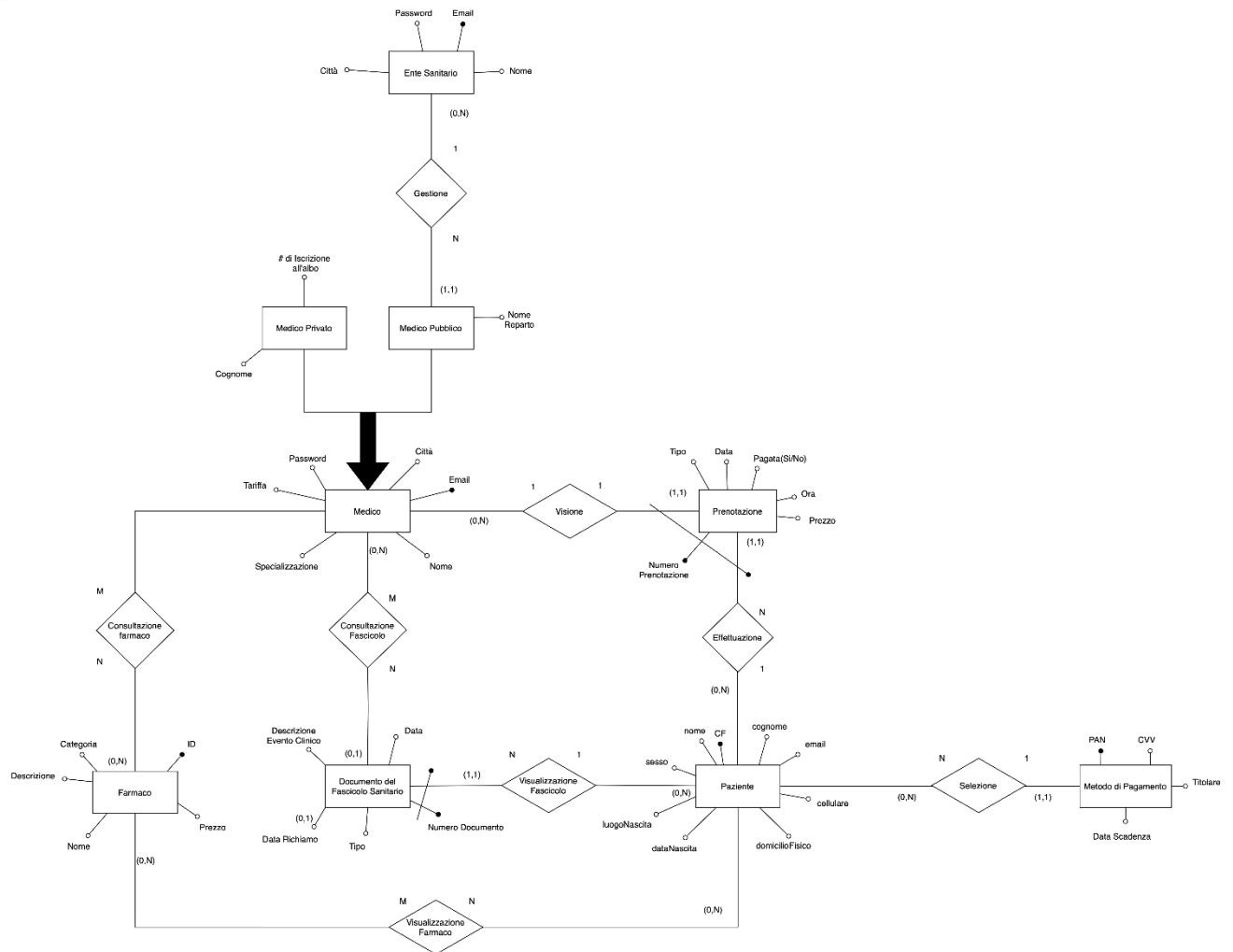
Per la gestione affidabile e persistente dei dati nel nostro sistema, abbiamo adottato un approccio robusto basato su un database relazionale, che offre robustezza e familiarità. Questa scelta è stata motivata da diversi aspetti chiave, mirati a garantire una gestione dei dati efficiente e coerente.

Le ragioni chiave dietro l'adozione di MySQL includono:

- **Vincoli e Integrità Referenziale:** L'uso di vincoli e l'integrità referenziale in MySQL è prezioso per mantenere l'integrità dei dati e garantire che le relazioni tra le tabelle siano gestite correttamente.
- **Sicurezza dei Dati:** Utilizziamo le funzionalità di accesso e controllo degli utenti di MySQL per limitare l'accesso ai dati, garantendo la protezione delle informazioni sensibili.
- **Transazioni ACID:** MySQL segue il modello ACID (Atomicità, Consistenza, Isolamento, Durabilità), garantendo la coerenza e l'affidabilità dei dati. Questa caratteristica è essenziale in contesti come il nostro, in cui la precisione delle informazioni è cruciale.
- **Affidabilità e Backup:** Sfruttiamo le capacità di backup di MySQL per creare copie di sicurezza regolari, consentendo il ripristino in caso di malfunzionamenti hardware o software.
- **Flessibilità nella Query Language:** L'uso di SQL (Structured Query Language) rende la gestione delle query molto flessibile e potente, consentendo di eseguire interrogazioni complesse in modo efficiente.

La preferenza per MySQL è guidata dalla sua intuitiva usabilità e dalla familiarità del team con la piattaforma. L'esperienza pregressa di ogni membro semplifica l'integrazione e accelera il processo di sviluppo.

Per quanto riguarda la definizione delle entità e delle relazioni nel database, facciamo riferimento al modello ER (mostrato di seguito):



## 3.5 Controllo degli accessi e sicurezza

Oggetti	Attori	Paziente	Medico Privato	Medico Pubblico	Ente Sanitario
<b>Gestione Autenticazione e</b>		Login 2FA Logout Registrazione	Login Logout Registrazione	Login Logout	Login Logout Registrazione
<b>Gestione Utente</b>		Gestione FSE Cancellazione Account Richiesta Esenzioni	Gestione FSE Cancellazione Account	Gestione FSE	
<b>Gestione Prenotazioni</b>		Prenotazione vaccino Prenotazione visite mediche Prenotazione analisi Visualizzazione storico prenotazioni Visualizzazione dettagli prenotazione	Visualizzazione storico prenotazioni Visualizzazioni dettagli prenotazioni	Visualizzazione storico prenotazioni Visualizzazione dettagli prenotazioni	
<b>Gestione Farmaci</b>		Esplorazione database di farmaci Filtro catalogo Ricerca Farmaco Visualizzazione dettagli farmaco	Esplorazione database di farmaci Filtro catalogo Ricerca Farmaco Visualizzazione dettagli farmaco	Esplorazione database di farmaci Filtro catalogo Ricerca Farmaco Visualizzazione dettagli farmaco	Esplorazione database di farmaci Filtro catalogo Ricerca Farmaco Visualizzazione dettagli farmaco

<b>Gestione Ente</b>				Creazione account medico pubblico  Eliminazione account medico pubblico
<b>Gestione AI</b>	Diagnostica malattia cardiaca			

## 3.6 Controllo del flusso globale del sistema

Il sistema MediCare è un sistema software interattivo dove l'utente potrà utilizzare le funzionalità fornite dai vari sottosistemi tramite dei comandi presenti nell'Interfaccia Grafica. Da questo si evince che il controllo del flusso è di tipo event-driven dove l'utente con l'utilizzo dell'interfaccia grafica potrà richiedere, mediante la creazione di un evento, l'utilizzo di un servizio da parte del sistema che si occuperà di affidare all'appropriato handler di gestire l'evento che a sua volta indirizzerà il controllo del flusso al sottosistema che è fornito di tale funzionalità.

## 3.7 Condizioni Boundary

### Avvio del sistema

Identificativo <i>UCBC_1</i>	<i>Avvio del sistema</i>	<i>Data</i> 02/12/2023	
		<i>Vers.</i>	1.0
		<i>Autore</i>	Palumbo Gianluca
<b>Descrizione</b>	<i>Questo use case permette l'avvio del sistema</i>		
<b>Attore Principale</b>	<b>Nome</b> Admin		
<b>Attori secondari</b>	<b>Nome</b> NA		
<b>Entry Condition</b>	Il sistema è spento AND L'admin accede al Server		
<b>Exit condition</b> On success	Il sistema è avviato correttamente		

<b>Exit condition</b> On failure		Il sistema non è avviato
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>		
1	Admin:	Attraverso un comando avvia il Server
2	Sistema:	Stabilisce una connessione al Database, recupera i dati e mostra all'utente una pagina Browser Web con le operazioni che gli è consentito svolgere.
<b>I Flusso di eventi Alternativo: Connessione al Database fallita</b>		
2.a1	Sistema:	Invia un segnale di errore all'utente notificandolo della mancata connessione al Database
2.a2	Admin:	Controlla il collegamento al Database
2.a3	Admin:	Con un comando stabilisce la connessione al Database
2.a4	Admin:	Riesegue il passo 1
<b>II Flusso di eventi Alternativo: Dati danneggiati</b>		
2.b1	Sistema:	Invia un segnale di errore all'utente notificandolo di problemi ai dati
2.b2	Admin:	Controlla l'integrità dei dati
2.b3	Admin:	Corregge i dati danneggiati
2.b4	Admin:	Riesegue il passo 1

### Spegnimento del sistema

Identificativo UCBC_2		Spegnimento del sistema	Data	02/12/2023
			Vers.	1.0
			Autore	Calabrese Primo Vinicio
Descrizione		Questo use case permette lo spegnimento del sistema		
Attore Principale		Nome Admin		
Attori secondari		Nome NA		
Entry Condition		Il sistema è acceso AND L'admin vuole spegnere il sistema		
Exit condition On success		Il sistema viene spento correttamente		
Exit condition On failure		Il sistema non viene spento		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Admin:	Attraverso un comando spegne il Server		
2	Sistema:	Il sistema completa le ultime richieste, libera le risorse e si spegne		
I Flusso di eventi Alternativo: Connessioni attive				

2.a1	Sistema:	Invia un segnale di errore all'utente notificandolo di connessioni ancora attive
2.a2	Sistema:	Controlla le richieste ancora in corso
2.a3	Sistema:	Controlla che non ci siano più connessioni attive
2.a4	Admin:	Riesegue il passo 1
<b>Il Flusso di eventi Alternativo: Connessioni attive</b>		
2.b1	Sistema:	Invia un segnale di errore all'utente notificandolo di connessioni ancora attive
2.b2	Sistema:	Recide tutte le connessioni
2.b3	Admin:	Riesegue il passo 1

## 4. Servizi dei sottosistemi

### Sottosistema GestioneAutenticazione

Servizio	Descrizione	Interfaccia
<b>Registrazione</b>	Permette al medico e all'ente di registrarsi	GestioneAutenticazioneService
<b>Login</b>	Permette agli attori medico pubblico, medico privato e ente sanitario di accedere alla piattaforma e di usufruire di tutti i servizi	GestioneAutenticazioneService
<b>Logout</b>	Permette a tutti gli attori del sistema di disconnettersi dalla piattaforma	GestioneAutenticazioneService
<b>Login 2FA</b>	Permette al paziente di accedere alla piattaforma e di usufruire di tutti i servizi tramite autenticazione a due fattori	GestioneAutenticazioneService

### Sottosistema GestioneFarmaci

Servizio	Descrizione	Interfaccia
<b>Visualizzazione e dettagli farmaci</b>	Permette a tutti gli attori di visualizzare i dettagli dei farmaci	GestioneFarmaciService
<b>Filtro catalogo</b>	Permette a tutti gli attori di filtrare i farmaci per categoria	GestioneFarmaciService
<b>Ricerca di un farmaco</b>	Permette a tutti gli attori di cercare un farmaco	GestioneFarmaciService

### Sottosistema GestioneUtente

Servizio	Descrizione	Interfaccia
<b>Visualizzazione e fascicolo sanitario elettronico</b>	Permette a tutti gli attori di visualizzare il fascicolo sanitario elettronico	GestioneUtenteService
<b>Cancellazione account</b>	Permette al paziente e al medico privato di eliminare il proprio account	GestioneUtenteService
<b>Inserimento ISEE</b>	Permette al paziente di inserire e gestire il proprio ISEE	GestioneUtenteService

### Sottosistema GestionePrenotazioni

Servizio	Descrizione	Interfaccia
----------	-------------	-------------



<b>Prenotazione Visite Mediche</b>	Permette al paziente di prenotare una visita medica specialistica	GestionePrenotazioneService
<b>Prenotazione Analisi</b>	Permette al paziente di riservare una visita diagnostica strumentale	GestionePrenotazioneService
<b>Prenotazione Vaccini</b>	Permette al paziente di prenotarsi ad un richiamo vaccinale	GestionePrenotazioneService
<b>Visualizza dettagli</b>	Permette al paziente di vedere i dettagli di una prenotazione	GestionePrenotazioneService
<b>Visualizza storico</b>	Permette al paziente di vedere tutte le prenotazioni che ha effettuato	GestionePrenotazioneService
<b>Scelta metodo di pagamento</b>		Permette al paziente di scegliere quale metodologia di pagamento usare per pagare la prenotazione GestionePrenotazioneService

### Sottosistema GestioneAI

Servizio	Descrizione	Interfaccia
<b>Diagnostica cardiaca</b>	Permette a tutti gli attori di visualizzare una previsione che indica che un paziente potrebbe essere soggetto ad una malattia cardiaca	GestioneAIService

### Sottosistema Gestione Ente

Servizio	Descrizione	Interfaccia
<b>Registrazione ente</b>	Permette all'ente sanitario di registrarsi	GestioneEnteService
<b>Creazione account medico pubblico</b>	Permette all'ente di creare un account per un suo reparto sanitario	GestioneEnteService
<b>Eliminazione account reparto</b>	Permette all'ente di cancellare un account di un suo reparto sanitario	GestioneEnteService

## 5. Glossario

Sigla / Termine	Definizione
<b>ASL</b>	L'azienda sanitaria locale (A.S.L.) è un ente dotato di personalità giuridica pubblica e di autonomia imprenditoriale che opera nel quadro del servizio sanitario nazionale.
<b>2FA</b>	L'autenticazione a due fattori è un metodo di autenticazione che si basa sull'utilizzo congiunto di due metodi di autenticazione individuali.
<b>GDPR</b>	Il Regolamento generale sulla protezione dei dati (GDPR, dall'inglese General Data Protection Regulation) è un regolamento dell'Unione Europea che disciplina il modo in cui le aziende e le altre organizzazioni trattano i dati personali.
<b>Servizio Sanitario Nazionale</b>	Il Servizio Sanitario Nazionale, altrimenti conosciuto con la sigla SSN, è l'insieme delle funzioni, delle attività e dei servizi sanitari gestiti dallo Stato italiano per tutelare la salute dei cittadini.

<b>Diagnosi</b>	In medicina, la determinazione della natura o della sede di una malattia in base alla valutazione dei sintomi.
<b>SOA</b>	L'architettura orientata ai servizi (SOA) è un metodo di sviluppo del software che utilizza componenti chiamati servizi per creare applicazioni aziendali. Ogni servizio fornisce una funzionalità aziendale e i servizi possono comunicare tra loro attraverso piattaforme e lingue diverse.
<b>MVC</b>	Model-View-Controller (MVC) è un modello di architettura del software. MVC prevede un'architettura composta da tre parti diverse: i dati (Model), la visualizzazione dei dati (View) e la gestione degli input (Controller).
<b>HTTP</b>	Sigla di hypertext transfer protocol, usata in informatica, e in particolare nella tecnica della rete Internet, che indica il protocollo di accesso alle informazioni.
<b>Modello ER</b>	Un modello ER per database consente di visualizzare il modo in cui le entità all'interno di un database si relazionano tra loro e con gli attributi di ciascuna entità. Questo esempio illustra le entità coinvolte nel sistema scolastico.
<b>FSE</b>	Il Fascicolo Sanitario Elettronico è l'insieme dei tuoi dati e documenti digitali di tipo sanitario e sociosanitario generati da eventi clinici presenti e trascorsi.
<b>ISEE</b>	L'ISEE è l'indicatore che serve per valutare e confrontare la situazione economica dei nuclei familiari che intendono richiedere una prestazione sociale agevolata.
<b>JINJA</b>	Jinja permette la creazione di file HTML, XML o in altri formati di markup, che vengono restituiti all'utente mediante una risposta HTTP.
<b>FLASK</b>	Flask è un micro-framework Web scritto in Python, è chiamato "micro-framework" perché ha un nucleo semplice ma estendibile. Contiene server e debugger per lo sviluppo e supporto integrato per il test d'unità.
<b>SqlAlchemy</b>	SQLAlchemy è una libreria per Python che fornisce un set di strumenti flessibili e potenti per interagire con database relazionali
<b>MYSQL</b>	MySQL è un sistema di gestione di database relazionali open source ampiamente utilizzato per la creazione, la gestione e l'interrogazione di basi di dati. Rappresenta una soluzione affidabile e veloce per lo storage e il recupero efficiente delle informazioni.
<b>SQL</b>	SQL, acronimo di Structured Query Language, è un linguaggio di programmazione utilizzato per gestire e manipolare database relazionali.