



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

CHEMO SMART

SCHEDULER FOR CHEMOTERAPY



S.D.D. System Design Document ChemoSmart

Versione	2.1
Data	27/12/2022
Destinatario	Prof.ssa F. Ferrucci, Prof. F. Palomba
Presentato da	C. Troiano, M. Purice, L. Miranda, A. Nappi, G. Basile, C. De Palma
Approvato da	A. Bergamo, F. P. Ianuzziello



Revision History

Data	Versione	Descrizione	Autori
24/11/2022	0.1	Prima stesura	LM, CT, CDP, GB, AN, MP
29/11/2022	0.2	Aggiunta divisione in sottosistemi, diagramma architetturale del sistema	LM, CT, CDP, GB, AN, MP
02/12/2022	0.3	Aggiunto mapping hardware/software, gestione dati persistenti, controllo degli accessi e sicurezza, condizioni limite e servizi dei sottosistemi e glossario	LM, CT, CDP, GB, AN, MP
05/12/2022	0.4	Revisione generale del documento	LM, CT, CDP, GB, AN, MP
09/12/2022	0.5	Revisione generale del documento	LM, CT, CDP, GB, AN, MP
12/12/2022	1.0	Revisione del Documento per Consegna Intermedia	AB, FPI
27/12/2022	2.0	Revisione documento, aggiornato diagramma architetturale, mapping hw/sw	LM, CT, CDP, GB, AN, MP
21/02/2022	2.1	Revisione Documento per Consegna Finale	LM, CT, CDP, GB, AN, MP



Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Alessandro Bergamo	Project Manager	AB	a.bergamo2@studenti.unisa.it
Francesco Pio Iannuzziello	Project Manager	FPI	f.ianuzziello1@studenti.unisa.it
Luigi Miranda	Team Member	LM	l.miranda11@studenti.unisa.it
Ciro Troiano	Team Member	CT	c.troiano17@studenti.unisa.it
Antonio Nappi	Team Member	AN	a.nappi47@studenti.unisa.it
Giuseppe Basile	Team Member	GB	g.basile36@studenti.unisa.it
Mihail Purice	Team Member	MP	m.purice@studenti.unisa.it
Claudio De Palma	Team Member	CDP	c.depalma5@studenti.unisa.it



Sommario

Revision History	2
Team members	3
Sommario	4
1 Introduzione	5
1.1 Scopo del sistema	5
1.2 Design Goals.....	5
1.3 Definizioni, acronimi e abbreviazioni	8
1.4 Riferimenti.....	8
1.5 Organizzazione del documento.....	9
2 Architettura del sistema corrente	9
3 Architettura del sistema proposto	9
3.1 Panoramica sulla sezione	9
3.2 Decomposizione in sottosistemi	10
3.3 Mapping hardware/software	16
3.4 Gestione dei dati persistenti.....	17
3.5 Controllo degli accessi e sicurezza	19
3.6 Controllo globale del software	20
3.7 Condizioni limite	20
4 Servizi dei sottosistemi	25
5 Glossario.....	27



1 Introduzione

1.1 Scopo del sistema

ChemoSmart permetterà una migliore organizzazione degli appuntamenti nel reparto chemioterapico, grazie ad una WebApp con una interfaccia intuitiva ed un modulo di IA che gestirà la schedulazione degli appuntamenti.

Il fulcro del sistema è basato su:

- Uno **scheduler**, gestito da un modulo di IA che permetterà la schedulazione di appuntamenti di terapia chemioterapica in modo semplice ed efficiente a seconda della priorità assegnata ad ogni paziente sulla base del quadro clinico di quest'ultimo.
- Un **calendario**, che permetterà al personale ospedaliero e di segreteria di visualizzare gli appuntamenti schedulati, inoltre, in casi di necessità, permetterà ai medici di inserire un appuntamento manualmente.

1.2 Design Goals

Nella seguente sezione si andranno ad illustrare i Design Goals, ovvero le caratteristiche sulle quali il sistema si dovrà basare.

Seguendo le linee guida del libro “Object Oriented Software Engineering” di Bernd Bruegge, i design goals sono stati suddivisi nelle seguenti categorie:

- **Performance**, includono tutti i design goal riguardo velocità e requisiti di spazio, sicurezza, efficienza del sistema;
- **Dependability**, determinano lo sforzo necessario per la corretta gestione dei dati e di integrità del sistema e il comportamento del sistema in caso di failure e le sue conseguenze;
- **Maintenance**, determinano lo sforzo necessario per modificare e gestire il sistema dopo il suo rilascio;
- **End User**, qualità desiderabili per l'utente finale.

Ciascun design goal è descritto da:

- **Rank**, specifica un valore di priorità da 1 a 12 (1 massimo 12 minimo);
- **ID**, un identificatore univoco ed un nome esplicativo;
- **Descrizione**, una descrizione formale ed esplicita del design goal;



- **Categoria**, specifica a quale categoria tra quelle individuate in precedenza appartiene;
- **RNF di origine**, specifica da quale RNF deriva il design goal;
- **Autore**, autore del design goal;

Di seguito la tabella dei design goal ordinata per valore di priorità:

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine	Autore
1	DG_1 Integrità delle operazioni	Il sistema, che tratta con dati di pazienti con malattie gravi e farmaci a breve scadenza, dovrà garantire una percentuale di successo delle operazioni superiore al 96%.	Performance	RNF_A_1	GB
2	DG_2 Scheduler Calendario Intelligente	Il sistema dovrà comprendere un modulo di intelligenza artificiale, per la schedulazione intelligente degli appuntamenti di una terapia chemioterapica di un paziente sulla base della sua condizione clinica. Il sistema dovrà garantire una percentuale di successo superiore al 85%.	Performance	RNF_IM_1	CDP
3	DG_3 Sicurezza e riservatezza dei dati	Il sistema dovrà garantire la massima sicurezza dei dati conservati in merito alla privacy, come dettato dal Regolamento UE 2016/679 del Parlamento italiano. Il sistema dovrà assicurare la visualizzazione dei dati persistenti solo agli utenti che hanno diritto ad accedervi.	Performance	RNF_LE_1	LM
4	DG_4 Coerenza delle operazioni	Il sistema dovrà fornire una divisione netta e chiara delle operazioni possibili in base alla categoria di utente che utilizza il sistema.	Dependability	RNF_A_2	CT
5	DG_5 Feedback esplicito	Il sistema dovrà fornire un feedback esplicito per ogni operazione compiuta così da rendere trasparente e immediato il comportamento del sistema.	Dependability	RNF_U_2	AN



6	DG_6 Standard di manutenibilità	Il sistema dovrà essere sviluppato seguendo i principali standard per garantire manutenibilità, perché seguirà lo standard ISO/IEC 9126*	Maintenance	RNF_S_1	GB
7	DG_7 Integrità del sistema in caso di errore	Il sistema in caso di errore dovrà essere in grado di mantenere integrità nelle operazioni ancora attive e bloccare tempestivamente le operazioni già in programma, notificando l'errore tramite messaggi diretti.	Dependability	RNF_A_2	MP
8	DG_8 Manutenibilità del sistema	Il sistema dovrà garantire la manutenibilità da parte dell'amministratore come la correzione di problemi o il miglioramento di una componente software, in media 2 volte al mese.	Maintenance	RNF_OP_1	LM
9	DG_9 Persistenza dei dati	Il sistema dovrà prevedere l'inserimento dei dati persistenti all'interno di un database non relazionale tramite "MongoDB".	Performance	RNF_IN_1	CDP
10	DG_10 Usabilità del sistema	Il sistema deve presentare all'utente un'interfaccia chiara e facile da comprendere, rendendo ogni azione ben esplicita.	End User	RNF_U_1	CT
11	DG_11 Semplicità e reattività del sistema	Il sistema dovrà prevedere una Single Web Application presentando contenuti dinamici e adattandosi ad ogni tipologia di device.	End User	RNF_PA_1 RNF_P_1	MP
12	DG_12 Tempi di risposta	Il sistema deve garantire un tempo di risposta non superiore a 30 secondi.	Performance	RNF_P_2	AN



TRADE-OFF

Trade-off	Descrizione
Affidabilità modulo IA VS Tempi di risposta	Il sistema ha come priorità l'ottimizzazione del modulo di IA per la schedulazione, il modulò dovrà avere un tasso di precisione che oscilla tra l'85% e il 90% ed un tasso di recall fra il 75% e 80%, anche a costo di allungare i tempi di risposta.



1.3 Definizioni, acronimi e abbreviazioni

Nella seguente sezione verranno riportati gli acronimi individuati:

- **DG:** Design Goal
- **DB:** Database
- **NRDBMS:** Non-Relational DBMS
- **DBMS:** Database Management System
- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document
- **TPD:** Test Plan Document
- **TCS:** Test Case Specification Document

1.4 Riferimenti

Di seguito una lista dei documenti progettuali:

- [Requirements Analysis Document;](#)
- [System Design Document;](#)
- [Test Plan Document;](#)
- [Test Case Specification Document;](#)



1.5 Organizzazione del documento

Introduzione: Viene descritto in generale lo scopo del sistema, gli obiettivi di design che il sistema propone di raggiungere.

Architettura software corrente: Viene descritto lo stato attuale dell'architettura del software già presente.

Architettura software proposta: Viene descritto come il sistema sarà definito e partizionato in sottosistemi, il loro mapping Hardware/Software, la gestione dei dati persistenti. Verranno poi presentate la struttura dei singoli sottosistemi e le boundary conditions riguardanti l'intero sistema.

Glossario: Contiene la lista dei termini usati nel documento con annessa spiegazione.

2 Architettura del sistema corrente

Al momento, non esiste alcun sistema software ad-hoc che racchiuda tutte le funzionalità che ChemoSmart propone di offrire. Le alternative presenti sul mercato, ad oggi, sono sviluppate per contesti generici e potrebbero risultare molto inefficienti e, fondamentalmente, poco sicure per la gestione di un sistema ospedaliero.

3 Architettura del sistema proposto

3.1 Panoramica sulla sezione

Il sistema sarà realizzato tramite un'architettura a Microservizi. Tale architettura ci offre la possibilità di realizzare un sistema efficace ed efficiente con alta scalabilità, requisiti fondamentali per l'obiettivo a lungo termine del sistema.

Nello sviluppo del sistema verranno utilizzati **HTML5**, **EJS**, **Bootstrap** per la realizzazione del front-end.

Per il back-end verrà utilizzato **Node.js** ed **Express.js**.

Il login alla piattaforma sarà gestito esternamente dal **Sistema Pubblico di Identità Digitale (SPID)** tramite l'API fornito.

Lo scheduler "intelligente" degli appuntamenti verrà realizzato tramite un modulo IA basato su modelli di Machine Learning tramite linguaggio **Python**, in particolare mediante l'utilizzo delle librerie **Scikit-Learn** e **Pandas**.

Per la gestione della persistenza, e quindi di un database, verranno utilizzati:

- **Mongoose**, per l'interfacciamento tra database ed i Microservizi;
- **MongoDB** per lo storage di dati.

3.2 Decomposizione in sottosistemi

I sottosistemi individuati sono:

- **Persistenza**, si occupa della gestione dei dati persistenti;
- **Mongoose**, si occupa di gestire l'interazione di tutti i microservizi con i relativi database ad essi associati;
- **Autenticazione**, si occupa del corretto funzionamento del login e del logout.
- **Gestione Appuntamento**, si occupa della creazione, modifica, cancellazione e visualizzazione di un appuntamento.
- **Gestione Paziente**, si occupa della gestione della Cartella Clinica come i dati del paziente, diagnosi passate e attuali, stato di salute, informazioni di contatto e recapiti familiari, quadro di salute generale ed inoltre permette di creare, modificare, cancellare e visualizzare Terapie.
- **Gestione Terapia**, si occupa della gestione delle terapie mostrando e permettendo una gestione rapida delle terapie;
- **Gestione Farmaci**, si occupa della gestione dei farmaci richiesti.

Di seguito un diagramma componenti in cui sono illustrate le dipendenze dei sottosistemi.

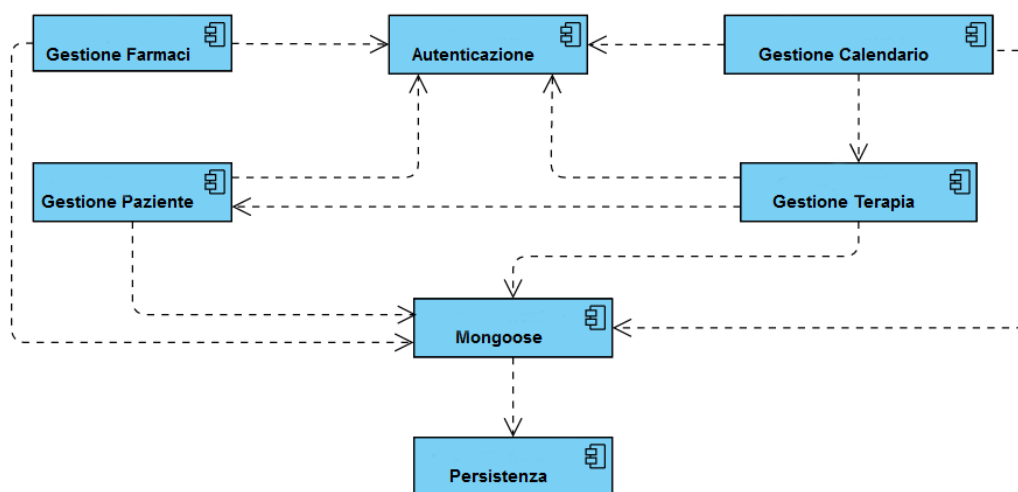
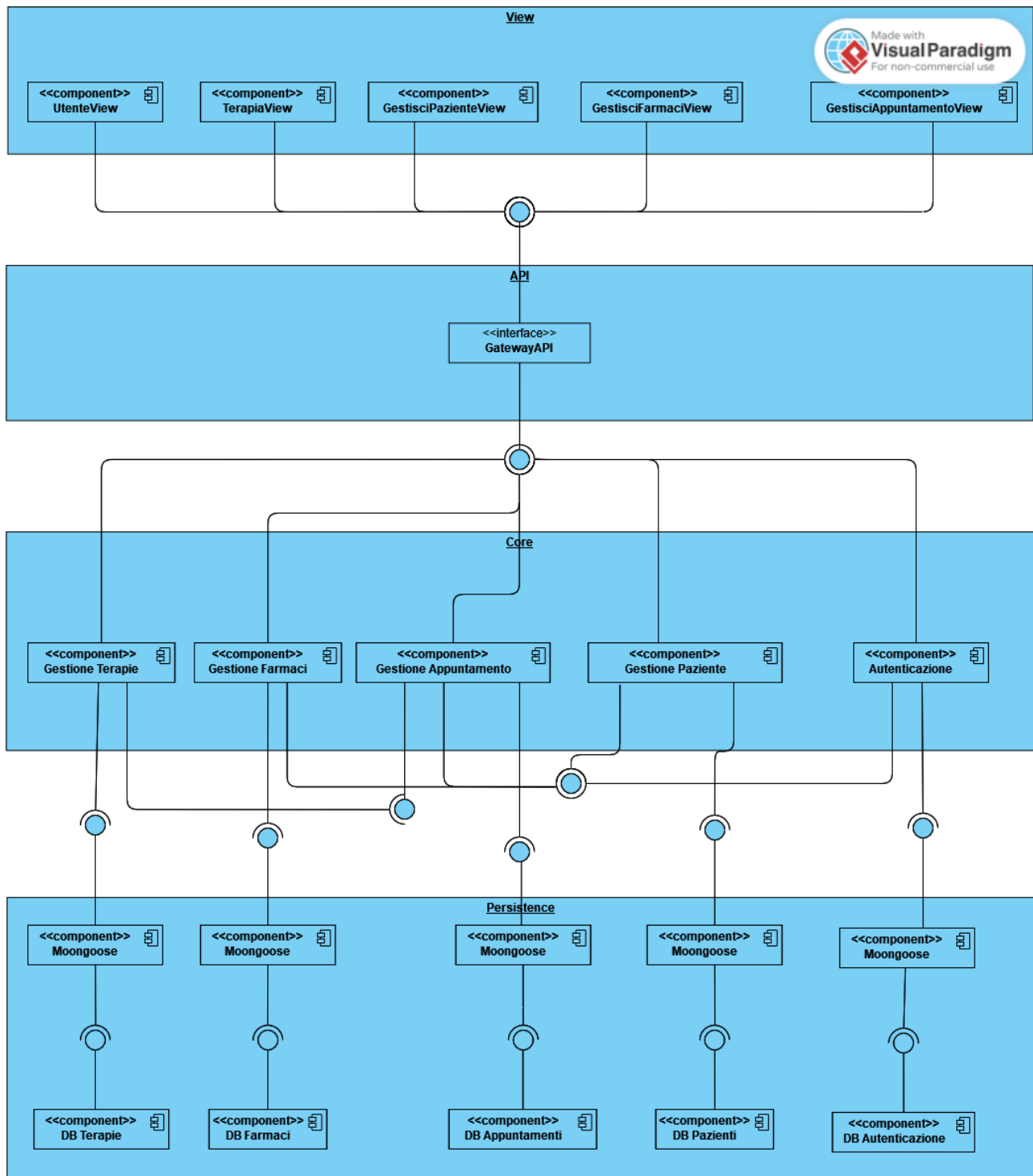
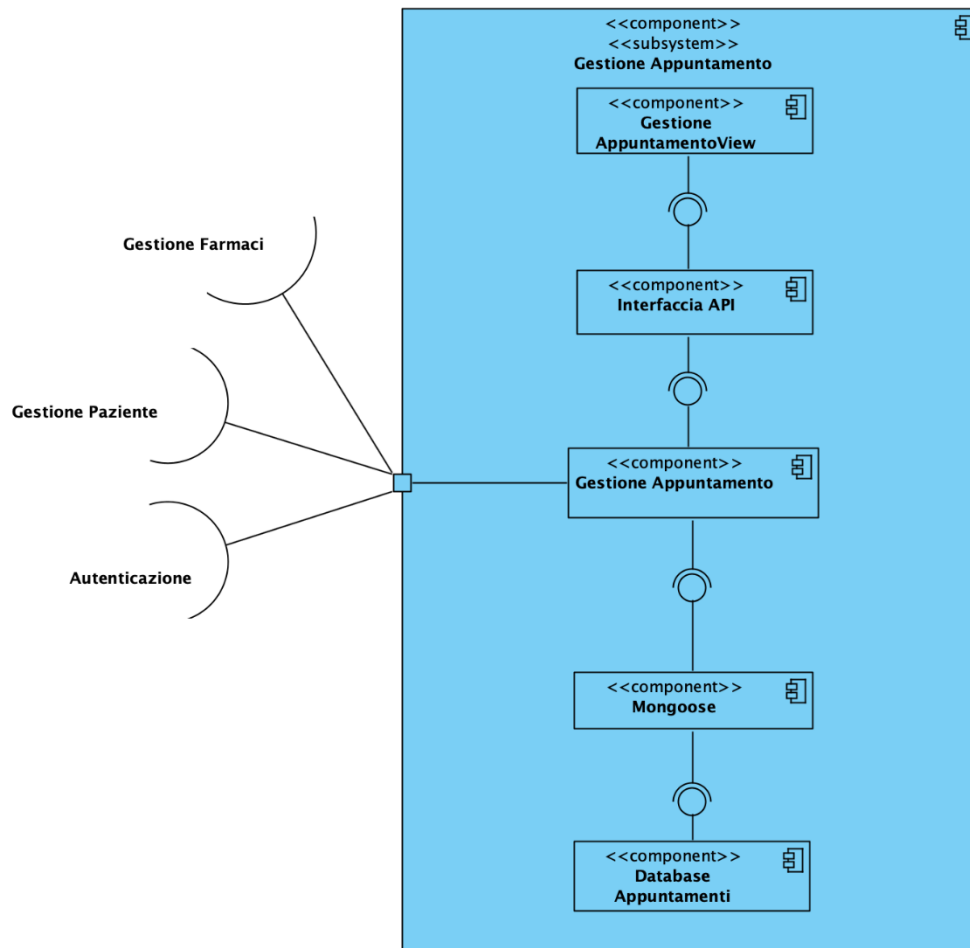


Diagramma architetturale



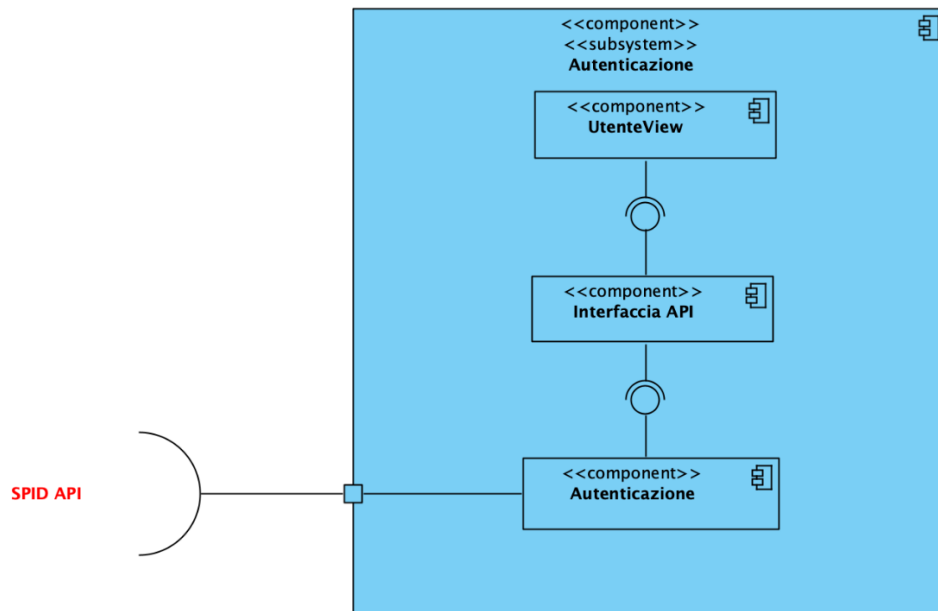
Di seguito vengono illustrati i vari sottosistemi identificati:

Sottosistema Gestione Appuntamento

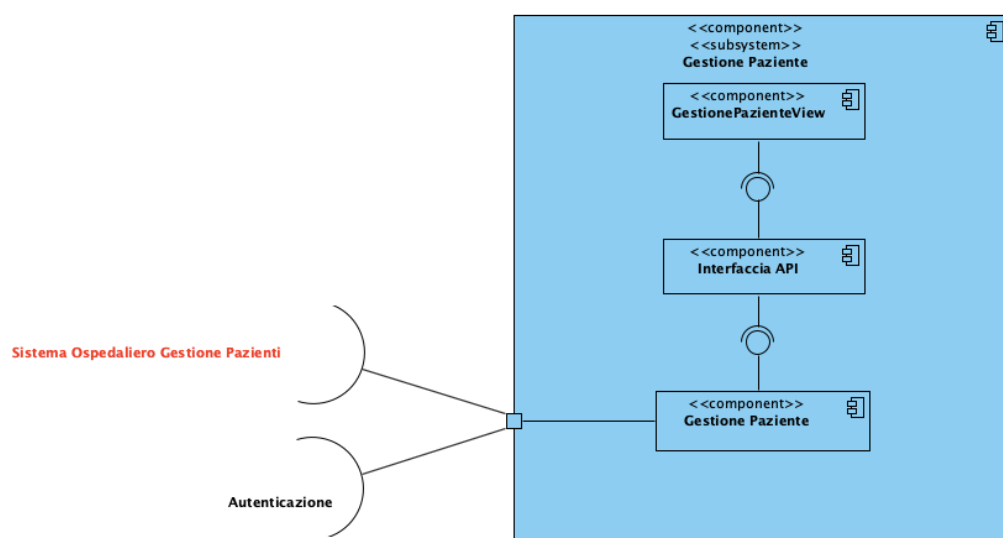




Sottosistema Autenticazione

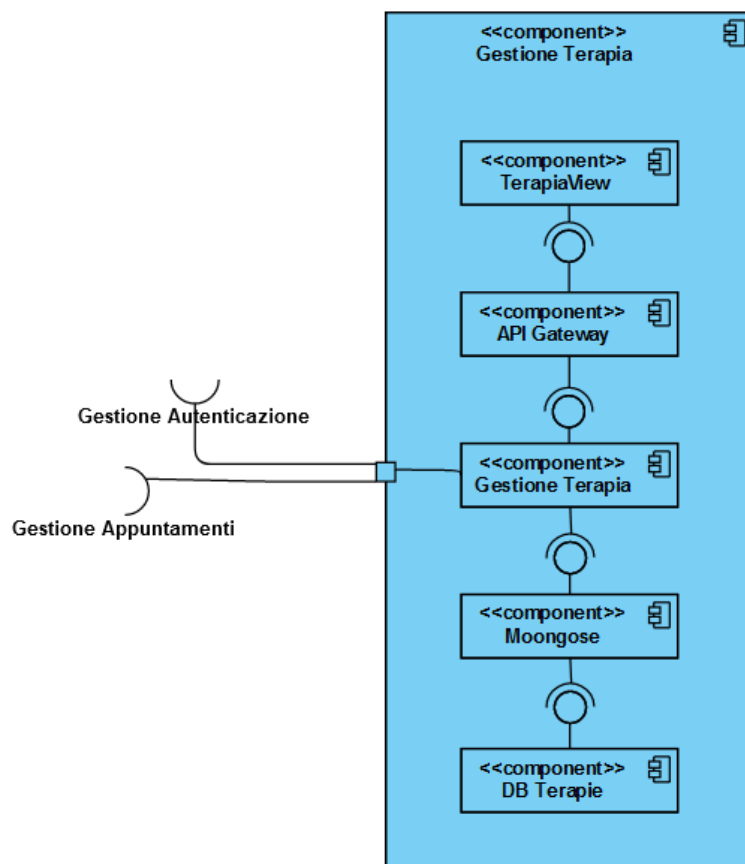


Sottosistema Gestione Paziente



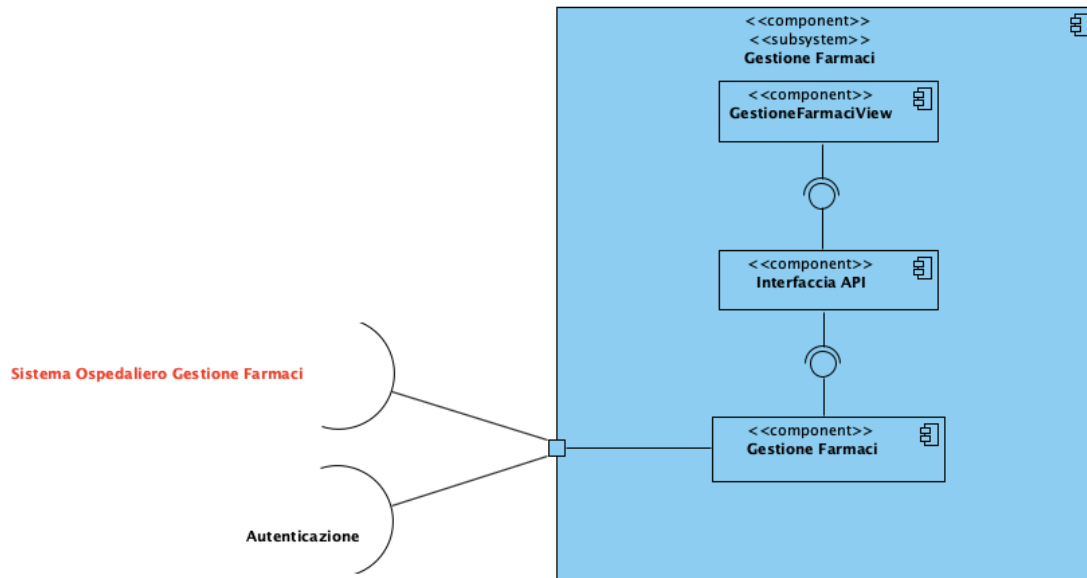


Sottosistema Gestione Terapia



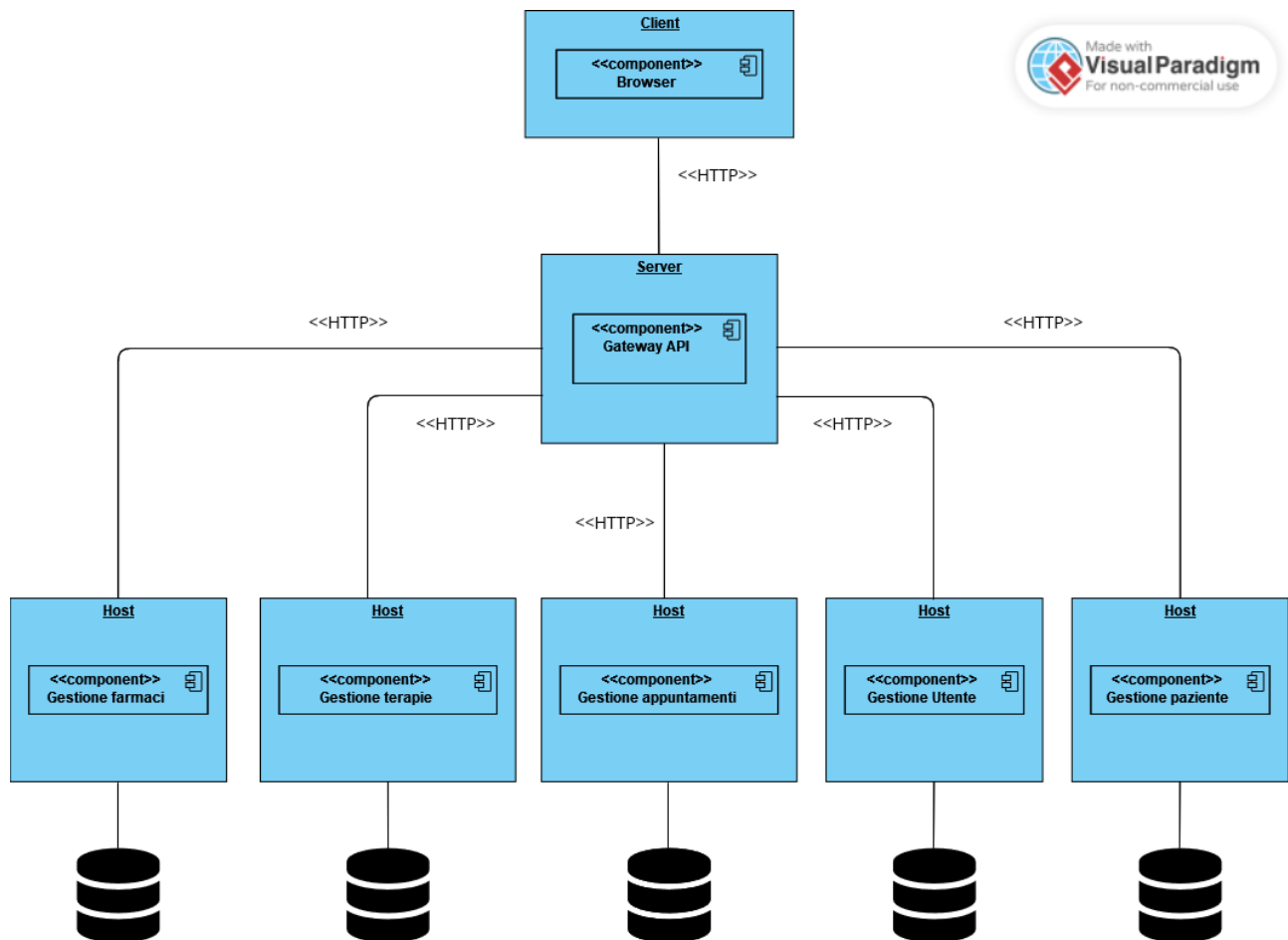


Sottosistema Gestione Farmaci



3.3 Mapping hardware/software

Il sistema sarà realizzato tramite un server che fungerà da nodo centrale per lo scambio di informazioni da un client ad un Microservizio e viceversa. Di seguito una rappresentazione tipica di come sarà realizzato il sistema.





3.4 Gestione dei dati persistenti

Introduzione

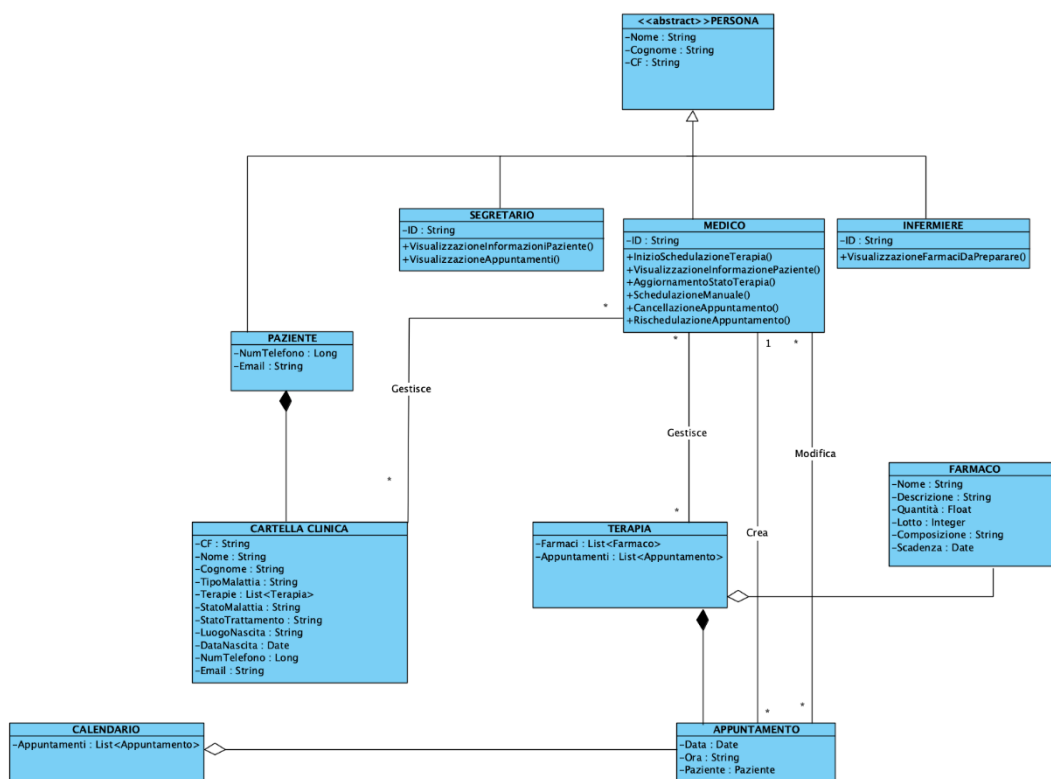
Per la gestione dei dati persistenti è stato scelto di utilizzare un database non relazionale per agevolare l'accesso ai dati e velocizzare tutte le operazioni canoniche di inserimento, modifica ed eliminazione.

In particolare, l'utilizzo di un NRDBMS permette:

- **Ottima scalabilità**, peculiare caratteristica dei NRDBMS. Permette di poter inserire nuove informazioni, nuovi dati da memorizzare in maniera rapida e completamente invisibile all'utente che utilizza il servizio. Questa caratteristica risulta fondamentale per le operazioni che il sistema prevede di fornire in futuro.
- **Performance**, permette di ottenere dati e trasmetterli in maniera rapida e sicura senza dipendere dalle classiche join di un RDBMS che appesantirebbero il servizio che ne fa uso e, di conseguenza, l'intero sistema.
- **Semplicità**, altra caratteristica fondamentale dei NRDBMS, permette di gestire il database in maniera esplicita ed esplicativa.
- **Object-Oriented**, un NRDBMS offre la possibilità di mappare in maniera più efficiente l'oggetto del proprio applicativo ai dati che esso conserva, semplificando e riducendo notevolmente l'effort richiesto per la realizzazione di interfacce per lo scambio di dati dal database all'applicativo e viceversa.

Questa scelta, inoltre, è risultata fondamentale per la gestione dei dati nel contesto di un'architettura a microservizi, poiché quest'ultimi potrebbero risiedere su nodi differenti, molto distanti fra di loro e l'efficienza comunicativa tra le parti del sistema risulta essere uno dei fulcri del sistema stesso.

CD_SDD: Class Diagram del sistema



Dizionario dei dati

Nome Entità	Appuntamento	
Descrizione	Contiene i dati relativi ad un appuntamento	
Nome campo	Tipo	Altri vincoli
Id_Appuntamento	ObjectId	NOT NULL AUTO INCREMENT
DataInizio	Date	NOT NULL
DataFine	Date	NOT NULL
Codice Fiscale	String	NOT NULL Length == 16
Nome	String	NOT NULL
Cognome	String	NOT NULL



FarmaciAppuntamento	String	NOT NULL
TipoSchedulazione	Int	Può assumere: <ul style="list-style-type: none"> • 0: Automatico • 1: Manuale
Stato	Int	Può assumere: <ul style="list-style-type: none"> • 0: Schedulato • 1: In esecuzione • 2: Terminato

Nome Entità	Farmaco	
Descrizione	Contiene i dati relativi ad un farmaco	
Nome campo	Tipo	Altri vincoli
Id_Farmaco	ObjectId	NOT NULL AUTO INCREMENT
nome	String	NOT NULL
descrizione	String	NOT NULL
dose	String	NOT NULL
stock	Int	NOT NULL

Nome Entità	Paziente	
Descrizione	Contiene i dati relativi ad un paziente	
Nome campo	Tipo	Altri vincoli
Id_Paziente	ObjectId	NOT NULL AUTO INCREMENT
nome	String	NOT NULL
cognome	String	NOT NULL
cf	String	NOT NULL Length == 16
sex	String	NOT NULL ['M', 'F']
dataNascita	Date	NOT NULL
età	Int	NOT NULL



telefono	Int minimum 12 maximum 12	NOT NULL Length = 12
email	String	NOT NULL
Indice_inquinamento_ambientale	Int	NOT NULL Range Val[0,10]
Indice_uso_alcolici	Int	NOT NULL Range Val[0,10]
grado_allergia	Int	NOT NULL Range Val[0,10]
Grado_rischio_lavorativo	Int	NOT NULL Range Val[0,10]
Indice_fattori_rischio_familiare	Int	NOT NULL Range Val[0,10]
Indice_malattie_croniche	Int	NOT NULL Range Val[0,10]
Indice_alimentazione_scorretta	Int	NOT NULL Range Val[0,10]
Indice_obesità	Int	NOT NULL Range Val[0,10]
Grado_esposizione_fumo_attivo	Int	NOT NULL Range Val[0,10]
Grado_esposizione_fumo_passivo	Int	NOT NULL Range Val[0,10]
Indice_dolori_localizzati	Int	NOT NULL Range Val[0,10]
Indice_emottisti	Int	NOT NULL Range Val[0,10]
Indici_astenia	Int	NOT NULL Range Val[0,10]
Indice_perdita_peso	Int	NOT NULL Range Val[0,10]
Indice_dispnea	Int	NOT NULL Range Val[0,10]
Indice_respiro_sibilante	Int	NOT NULL Range Val[0,10]
Indice_disfagia	Int	NOT NULL Range Val[0,10]
Stato_dita_di_Ippocrate	Int	NOT NULL Range Val[0,10]
Stato_immunodepressione	Int	NOT NULL Range Val[0,10]
Indice_tosse_secca	Int	NOT NULL Range Val[0,10]
Indice_russamento	Int	NOT NULL Range Val[0,10]
priorità	String	NOT NULL ['Alta', 'Media', 'Bassa']



Nome Entità	Terapia	
Descrizione	Contiene i dati relativi ad una terapia	
Nome campo	Tipo	Altri vincoli
Id_terapia	ObjectId	NOT NULL
Cf paziente	String	NOT NULL Length = 16
farmaco	String	NOT NULL
dataInizio	Date	NOT NULL
numAppuntamenti	Int	NOT NULL
Frequenza appuntamenti	Int	NOT NULL [7,14,21]
stato	String	NOT NULL ['Non schedulata','In corso','Terminata']



3.5 Controllo degli accessi e sicurezza

Di seguito è riportata la matrice degli accessi che mostra per ogni attore del sistema tutti i servizi a cui può accedere.

Attore Oggetto	Medico	Segretario	Infermiere
Autenticazione	Login Logout	Login Logout	Login Logout
Gestione Appuntamento	IniziaSchedulazioneManualeA ppuntamento CancellaAppuntamento ModificaAppuntamento VisualizzaInfoAppuntamento		
Gestione Paziente	IniziaNuovaTerapia VisualizzaCartellaClinica CreaCartellaClinica VisualizzaAppuntamenti VisualizzaInfoAppuntamento	VisualizzaAnagraficaP aziente VisualizzaAppuntame nti VisualizzaInfoAppunta mento	
Gestione Terapia	AggiornaTerapia TerminaTerapia CreaTerapia		
Gestione Farmaci	VisualizzaFarmaci ComunicazioneFarmaciPronti		VisualizzaFarmaci ComunicazioneFarma ciPronti



3.6 Controllo globale del software

Il sistema ChemoSmart presenterà ogni funzionalità offerta tramite un'interfaccia grafica e permetterà l'esecuzione di quest'ultima tramite comandi impartiti dall'utente. Quando un utente seleziona una funzionalità, quest'ultima verrà gestita dal nodo centrale che passerà la richiesta al nodo del microservizio corrispondente.

3.7 Condizioni limite

In questa sezione vengono presentate le boundary conditions inerenti all'avvio del sistema, spegnimento del sistema, fallimento del sistema ed errore di accesso ai dati persistenti.

Avvio del sistema

Identificativo UCBC_1	Avvio del sistema	Data	01/12/2022
		Vers.	1.0
		Autore	Giuseppe Basile
Descrizione	Lo Use Case permette all'amministratore di avviare il sistema.		
Attore Principale	Amministratore È interessato ad avviare il sistema.		
Attori secondari	NA		
Entry condition	L'amministratore accede al Server AND Il sistema si trova nello stato 'spento'		
Exit condition On success	L'amministratore avvia correttamente il sistema.		
Exit condition On failure	Il sistema non viene avviato correttamente.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			



1	Amministratore	Inizializza il sistema eseguendo un comando sulla macchina.
2	Sistema	Verifica l'integrità dei dati persistenti, e in caso positivo consente di utilizzare i suoi servizi e le sue funzionalità destinati agli utenti. Lo stato del sistema passa da 'spento' ad 'avviato'
I Scenario/Flusso di eventi Alternativo: Dati persistenti corrotti		
2.1	Sistema	Il sistema notifica l'amministrazione che alcuni dati sono corrotti e non avvia il sistema.
2.2	Amministratore	Corregge i dati corrotti utilizzando una copia di Backup.
2.3	Amministratore	Esegue il passaggio 1.

Arresto del sistema

Identificativo UCBC_2	Arresto del sistema	Data	01/12/2022
		Vers.	1.0
		Autore	Luigi Miranda
Descrizione	Lo Use Case permette all'amministratore di arrestare il sistema.		
Attore Principale	Amministratore È interessato ad arrestare il sistema.		
Attori secondari	NA		
Entry condition	L'amministratore accede al Server AND Il sistema si trova nello stato 'avviato'		
Exit condition On success	L'amministratore arresta correttamente il sistema.		



Exit condition		Il sistema non viene arrestato correttamente.
On failure		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Amministratore	Esegue sulla macchina un comando per arrestare il sistema.
2	Sistema	Verifica che tutti le componenti siano completamente disconnesse, in caso positivo arresta correttamente il sistema. Lo stato passa da 'attivo' a 'spento'.
I Scenario/Flusso di eventi Alternativo: Connessione con componenti ancora aperte.		
2.1	Sistema	Il sistema notifica l'amministrazione che ci sono ancora connessioni aperte.
2.2	Sistema	Resta in attesa che ogni componente termini le sue operazioni, evitando di generare nuove connessioni se non per le operazioni ancora in corso.
2.3	Sistema	Controlla che tutte le operazioni siano terminate e che non ci siano connessioni aperte, in caso positivo arresta correttamente il sistema.
2.4	Sistema	Notifica l'amministratore che l'arresto del sistema è avvenuto con successo.

Fallimento del sistema

Identificativo UCBC_3	Fallimento del sistema	Data	01/12/2022
		Vers.	1.0
		Autore	Giuseppe Basile
Descrizione	Lo Use Case fornisce all'amministratore la possibilità di rispondere ad un caso di fallimento accidentale del sistema.		
Attore Principale	Amministratore È interessato a risolvere una situazione in cui il sistema fallisce.		



Attori secondari	NA	
Entry condition	L'amministratore accede al Server AND Il sistema inaspettatamente viene terminato.	
Exit condition On success	L'amministratore risolve correttamente il fallimento.	
Exit condition On failure	Non è possibile risolvere il fallimento.	
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Sistema	Notifica con un messaggio di errore che si è verificato un fallimento del sistema, nel caso di fallimento non risolvibile il sistema avvia la procedura di arresto (UCBS_2).
I Scenario/Flusso di eventi Alternativo: Una funzionalità ha causato un fallimento.		
1.1	Amministratore	La funzionalità che ha causato un errore viene sospesa temporaneamente, nell'attesa che tale funzionalità venga corretta così da evitare nuovi fallimenti.

Errore accesso ai dati persistenti

Identificativo <i>UCBC_4</i>	Errore di Accesso ai Dati Persistenti	<i>Data</i>	<i>01/12/2022</i>
		<i>Vers.</i>	<i>1.0</i>
		<i>Autore</i>	<i>Luigi Miranda</i>
Descrizione	Lo Use Case presenta come il sistema deve comportarsi nel caso in cui è impossibile accedere ai dati persistenti.		
Attore Principale	Amministratore È interessato a risolvere problemi di accesso ai dati persistenti.		
Attori secondari	NA		



Entry condition		L'amministratore accede al Server AND Il sistema non è in grado di accedere ai dati persistenti.
Exit condition On success		Il sistema non è stato intaccato e riprende il suo funzionamento correttamente.
Exit condition On failure		Il sistema è stato intaccato e non riprende il suo funzionamento correttamente.
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Sistema	Comunica all'amministratore con apposito messaggio che non è stato possibile accedere ai dati persistenti.
2	Sistema	Blocca tutte le funzionalità che desiderano accedere ai dati persistenti e comunica a tutte le richieste in sospeso l'impossibilità di accedere a tali dati.
3	Amministratore	Include UCBC_1
4	Sistema	Ripristina l'accessibilità ai dati persistenti.



4 Servizi dei sottosistemi

In questa sezione vengono descritti tutti i servizi dei sottosistemi realizzati.

Sottosistema Autenticazione

Servizio	Descrizione	Interfaccia
Login	Questa funzionalità consente l'accesso alla Piattaforma attraverso il sistema di identità digitale SPID.	AutenticazioneService
Logout	Questa funzionalità consente la disconnessione dalla Piattaforma.	AutenticazioneService

Sottosistema Gestione Appuntamento

Servizio	Descrizione	Interfaccia
IniziaSchedulazioneManualeAppuntamento	Questa funzionalità consente di inserire un nuovo appuntamento scegliendo manualmente la data e l'orario.	GestioneAppuntamentoService
CancellaAppuntamento	Questa funzionalità consente di eliminare un appuntamento dal calendario.	GestioneAppuntamentoService
ModificaAppuntamento	Questa funzionalità consente di modificare la data e l'orario di un appuntamento.	GestioneAppuntamentoService
VisualizzaInfoAppuntamento	Questa funzionalità consente di visualizzare le informazioni di un singolo appuntamento.	GestioneAppuntamentoService
VisualizzaAppuntamenti	Questa funzionalità consente di visualizzare il calendario con tutti gli appuntamenti di uno specifico mese.	GestioneAppuntamentoService
VisualizzaInfoAppuntamento	Questa funzionalità consente di visualizzare le informazioni di un singolo appuntamento.	GestioneAppuntamentoService

Sottosistema Gestione Paziente

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



IniziaNuovaTerapia	Questa funzionalità consente di inserire una nuova terapia di un determinato paziente.	GestionePazient eService
VisualizzaCartellaClinica	Questa funzionalità consente di visualizzare tutte le informazioni della cartella clinica di un paziente.	GestionePazient eService
VisualizzaAnagrafica	Questa funzionalità consente di visualizzare i dati anagrafici di un paziente.	GestionePazient eService

Sottosistema Gestione Terapia

Servizio	Descrizione	Interfaccia
AggiornaTerapia	Questa funzionalità consente di aggiornare lo stato di una terapia di un paziente.	GestioneTerapia Service
TerminaTerapia	Questa funzionalità consente di terminare una terapia.	GestioneTerapia Service
CreaTerapia	Questa funzionalità consente di creare una nuova terapia.	GestioneTerapia Service

Sottosistema Gestione Farmaci

Servizio	Descrizione	Interfaccia
VisualizzaFarmaci	Questa funzionalità consente di visualizzare tutti i farmaci della piattaforma.	GestioneFarmaci Service
ComunicazioneFarmaciPronti	Questa funzionalità consente di comunicare i farmaci che sono stati preparati.	GestioneFarmaci Service



5 Glossario

Termine	Descrizione
ChemoSmart	Nome del sistema che si andrà a realizzare.
Architettura a microservizi	Architettura software/hardware per la gestione dei servizi offerti tramite la suddivisione del sottosistema in tanti piccoli sottosistemi indipendenti posti anche su macchine differenti.
COTS	Commercial Off The Shelf, si riferisce a prodotti software e hardware presenti sul mercato per l'acquisto da parte di qualunque azienda interessata ad utilizzarla nei propri progetti e artefatti.
DBMS	Database Management System, applicativo per la gestione di un database relazionale.
NRDBMS	Non Relational DBMS, applicativo per la gestione di un database di tipo non relazionale.
Node.js	Framework basato su JavaScript che permette di utilizzare quest'ultimo per la gestione del backend.
Express.js	Framework basato su JavaScript per Node.js che fornisce componenti di backend per la realizzazione di API e web services che, ad oggi, rappresenta lo standard per la gestione del backend per Node.js
Bootstrap	Bootstrap è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web. Essa contiene modelli di progettazione basati su HTML e CSS, sia per la tipografia, che per le varie componenti dell'interfaccia, come moduli, pulsanti e navigazione, così come alcune estensioni opzionali di JavaScript.
TensorFlow	È una libreria software open source per l'apprendimento automatico, che fornisce moduli sperimentati e ottimizzati, utili nella realizzazione di algoritmi.
Scikit-Learn	È una libreria open source di apprendimento automatico per il linguaggio di programmazione Python.
Pandas	È una libreria open source per la manipolazione e analisi di dati, disponibile per Python.
MongoDB	NRDBMS disponibile al pubblico, sia in versione freeware per piccoli progetti che in versione
Mongoose	Libreria disponibile per Node.js che permette il collegamento tra database MongoDB e applicativo.