



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



System Design Document (SDD) Progetto Chemo

Riferimento	
Versione	1.0
Data	13/02/2023
Destinatario	Prof.ssa Filomena Ferrucci, Prof. Fabio Palomba
Presentato da	Gruppo C13_Chemo
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
03/12/2022	0.1	Prima stesura: scopo del sistema, panoramica del documento e controllo globale del software	Salvatore De Luca
4/12/2022	0.2	Panoramica e introduzione alla scomposizione in sottosistemi	Alfonso Caiazza
5/12/2022	0.3	Mapping HW/SW e Controllo degli accessi e sicurezza	Gianpio Silvestri
6/12/2022	0.4	Inserimento struttura dei sottosistemi, design goals e trade-off	Alfonso Caiazza
9/12/2022	0.5	Revisione documento	Salvatore De Luca, Giuseppe Giugliano, Alfonso Caiazza
23/01/2023	0.6	Modifica ordine sezione “Decomposizione in sottosistemi”	Gianpio Silvestri Salvatore De Luca
28/01/2023	0.7	Modifica sezione “Gestione dei dati persistenti”	Giuseppe Giugliano
30/01/2023	0.8	Eliminazione interazioni con il sistema “HL7/FHIR”	Gianpio Silvestri Salvatore De Luca
13/02/2023	1.0	Revisione finale	Tutto il team



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

Composizione gruppo

Nome	Ruolo	Matricola	E-mail
Claudia Buono	PM	0522501097	c.buono13@studenti.unisa.it
Anna Fulgione	PM	0522501103	a.fulgione4@studenti.unisa.it
Alfonso Caiazza	TM	0512109742	a.caiazza18@studenti.unisa.it
Salvatore De Luca	TM	0512111902	s.deluca39@studenti.unisa.it
Antonio Marcone	TM	0512111101	a.marcone8@studenti.unisa.it
Gianpio Silvestri	TM	0512107818	g.silvestri13@studenti.unisa.it
Giuseppe Giugliano	TM	0512104762	g.giugliano23@studenti.unisa.it



Sommario

Revision History	2
Composizione gruppo	3
1. Introduzione.....	6
1.1 Scopo del sistema	6
1.2 Design goals	6
1.2.1 Trade-off.....	7
1.3 Definizioni, acronimi e abbreviazioni.....	8
1.4 Riferimenti	8
1.5 Panoramica del documento	8
2. Architettura software corrente	9
3. Architettura software proposta	10
3.1 Panoramica.....	10
3.2 Decomposizione in sottosistemi	10
3.2.1 Sottosistema Gestione Autenticazione	11
3.2.2 Sottosistema Gestione Paziente.....	12
3.2.3 Sottosistema Gestione Medicinale	12
3.2.4 Sottosistema Schedulazione.....	13
3.2.5 Diagramma Architetturale	14
3.3 Mapping hardware/software	15
3.4 Gestione dei dati persistenti.....	15
3.4.1 Descrizione delle entità persistenti.....	16
3.4.1.1 Schema E-R.....	16
3.4.1.2 Schema E-R ristrutturato.....	17
3.4.2 Dizionario dei dati	17
3.4.2.1 Utente.....	17
3.4.2.2 Paziente	18
3.4.2.3 Medicinale	18
3.4.2.4 Agenda.....	19
3.4.2.5 Appuntamento.....	19
3.4.2.6 Terapia	20
3.4.2.7 Confezione	20
3.5 Controllo degli accessi e sicurezza.....	20



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

3.6 Controllo globale del software	21
3.7 Boundary conditions	22
3.7.1 Inizializzazione del sistema	22
3.7.2 Arresto del sistema	23
3.7.3 Crash del sistema	24
4. Glossario	26



1. Introduzione

1.1 Scopo del sistema

L'ospedale San Giovanni di Dio Ruggi D'Aragona di Salerno intende fornire al reparto Oncologico un miglioramento dell'organizzazione delle sedute chemioterapiche, utilizzando un modulo di intelligenza artificiale, basato sull'utilizzo ottimale dei farmaci e sul numero di poltrone presenti in sede.

Lo scopo del sistema è, quindi, fornire ai lavoratori del reparto un servizio che migliori drasticamente l'iter da svolgere per effettuare la chemioterapia, lo sviluppo di un calendario per effettuare le terapie. Queste operazioni, essendo svolte da appositi impiegati, al fine di essere migliorati necessitano di una profonda digitalizzazione.

1.2 Design goals

Rank	ID	Descrizione	Categoria	RNF di origine
9	DG_06 Performance elevate	Il sistema dovrà essere performante anche con più di 100 accessi contemporanei	Performance	RNF_P_01
5	DG_07 Tempi di risposta	Il sistema dovrà garantire una risposta all'utente entro i 10 secondi dalla sottomissione della richiesta	Performance	RNF_P_02
8	DG_01 Estendibilità	Il sistema dovrà consentire l'implementazione di successive funzionalità, date le grandi necessità degli utenti	Maintenance	RNF_F_01
1	DG_04 Sicurezza dei dati	Il sistema deve garantire la sicurezza dei dati permanenti, utilizzando protocolli sicuri, e assicurando che la loro gestione sia concessa ai soli utenti opportunamente identificati	Dependability	RNF_A_01, RNF_A_03



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

4	DG_10 Consistenza dei dati	Il sistema dovrà impedire la modifica dei dati, nel caso in cui l'utente commetta errori di inserimento nelle informazioni, avvertendolo e lasciando invariati i dati stessi	Dependability	RNF_A_02
7	DG_05 Notificare gli errori	Il sistema dovrà notificare l'utente tramite appositi messaggi, in caso di errori, che lo aiutino a completare le operazioni	Dependability	RNF_A_04
2	DG_02 Usabilità	Il sistema dovrà implementare un'interfaccia grafica basata sulle “8 regole d'oro di Shneiderman”	End User	RNF_U_01
3	DG_03 Navigazione facilitata	Il sistema dovrà consentire agli utenti finali di navigare facilmente sull'applicazione, utilizzando layout sulla base di una UX comune	End User	RNF_U_02
6	DG_08 Web Browser	Il sistema dovrà essere disponibile sui seguenti web browser: Google Chrome, Safari e Microsoft Edge	End User	RNF_S_01
10	DG_09 Lingua e time zone dell'applicazione	L'applicazione dovrà essere sviluppata in lingua italiana, con riferimento al time zone GTM+1	End User	RNF_L_01

1.2.1 Trade-off

Trade-off	Descrizione
Usabilità vs Portabilità	Il sistema favorirà una maggiore usabilità su alcuni web browser, poiché su altri le funzionalità, conformi alle linee guida sviluppate, potrebbero non essere supportate. L'attenzione sarà infatti maggiormente concentrata su Chrome, Safari e Edge, ovvero i browser ad oggi più utilizzati



Tempi di risposta vs Sicurezza	Il sistema dovrà garantire sempre la sicurezza dei dati a discapito anche del tempo di risposta, utilizzando operazioni di “messa in sicurezza” dei dati che potrebbero richiedere anche più di 10 secondi
Tempi di risposta vs Qualità	Se il software non rispetta i vincoli imposti sul tempo di risposta e, in particolare, di produzione della schedulazione, non dovrà essere intaccata la qualità dei dati, in quanto la produzione della schedulazione e la risposta all’utente deve essere ottimale ed esaustiva

1.3 Definizioni, acronimi e abbreviazioni

- DG: Design Goal
- RNF: Requisito Non Funzionale
- GUI: Graphical User Interface
- DAO: Data Access Object
- UML: Uniform Modeling Language
- DBMS: Database Management System
- SDD: System Design Document

1.4 Riferimenti

- [Requirements Analysis Document](#)

1.5 Panoramica del documento

Questo documento è suddiviso nelle seguenti parti:

- **Introduzione**, contiene la descrizione generale dello scopo del sistema e gli obiettivi di design da raggiungere;
- **Architettura software corrente**, presenta la descrizione dello stato attuale dell’architettura del software già presente;



- **Architettura software proposta**, che contiene la definizione del sistema e il suo partizionamento in sottosistemi, oltre a contenere il mapping Hardware/Software e gestione dei dati persistenti. Viene poi posta l'attenzione sulla struttura dei singoli sottosistemi e sulle boundary conditions riguardanti l'intero sistema;
- **Glossario**, contenente la lista dei termini usati all'interno del documento.

2. Architettura software corrente

Al momento non esiste alcun sistema che metta a disposizione tutte le funzionalità previste nell'applicazione Chemo.



3. Architettura software proposta

3.1 Panoramica

Il sistema proposto riguarda una Web Application che funga da supporto per il personale medico dell'ospedale San Giovanni di Dio Ruggi D'Aragona di Salerno, permettendo una gestione più efficiente dell'intero reparto di Oncologia.

Gli utenti individuati sono il gestore dei medicinali e il membro del personale medico, i quali potranno accedere all'applicazione attraverso un'apposita finestra di login: poiché la password di primo accesso sarà fornita dalla struttura ospedaliera, durante il suo primo accesso l'utente sarà indirizzato ad un form in cui, oltre a poter visualizzare la propria anagrafica, dovrà effettuare la modifica della password.

Il personale medico potrà accedere ai dati dei pazienti presenti nel database della struttura, modificarli o eliminarli; potrà inoltre richiedere la produzione di una schedulazione settimanale delle sedute chemioterapiche, effettuata da un modulo di IA.

Il gestore dei medicinali potrà amministrare i medicinali della struttura, potendo di fatto visualizzare eliminare o aggiungerne di nuovi.

L'architettura scelta per questo sistema è di tipo Three-Tier: la separazione della logica di presentazione da quella di elaborazione porta infatti diversi vantaggi, tra cui sicuramente un maggior riuso e migliore manutenzione del software.

3.2 Decomposizione in sottosistemi

La suddivisione in sottosistemi è stata effettuata considerando due proprietà fondamentali quali basso accoppiamento ed elevata coesione tra le componenti del sistema, per rendere così il sistema manutenibile e portatile. I sottosistemi individuati sono:

- **Gestione account:** fornisce le funzionalità di login, logout e di cambiamento password per tutti gli utenti del sistema;
- **Gestione paziente:** fornisce tutte le funzionalità necessarie per la gestione dei pazienti presenti nel database della struttura, come l'inserimento, la modifica o l'eliminazione di un paziente;
- **Gestione medicinale:** fornisce le funzionalità atte alla gestione dei medicinali, quali inserimento, modifica o eliminazione di un medicinale dal database;
- **Schedulazione:** mette a disposizione una delle funzionalità principali, su cui si fonda lo sviluppo del progetto, ovvero la creazione, tramite il lavoro di un modulo di IA, di una schedulazione

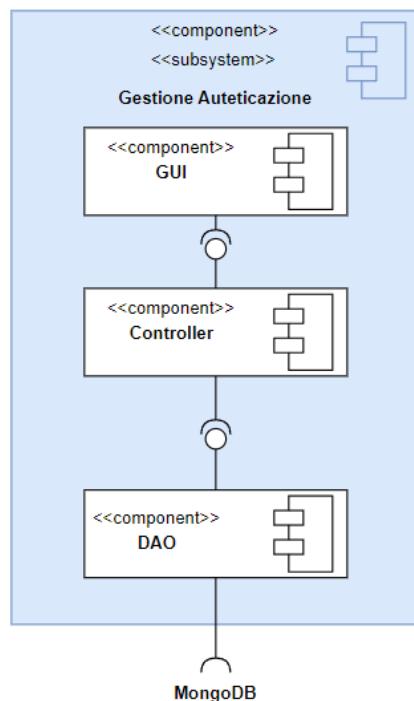


ottimizzata per le sedute chemioterapiche settimanali. Permette anche di visualizzare la schedulazione della settimana corrente.

In particolar modo, le componenti principali di ogni sottosistema saranno:

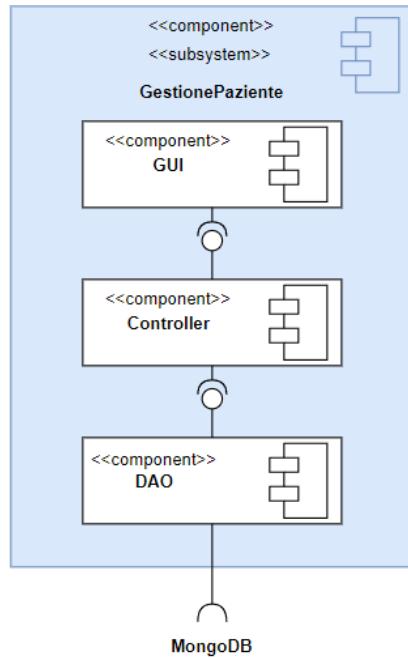
- **GUI**: componente che creerà le pagine da mostrare al client;
- **Controller**: per il controllo del sistema;
- **DAO**: per la gestione della persistenza.

3.2.1 Sottosistema Gestione Autenticazione

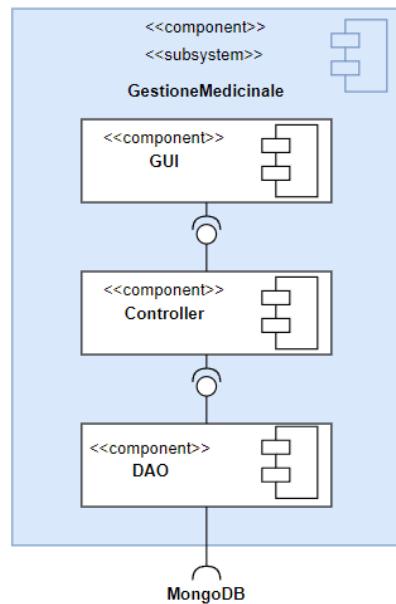




3.2.2 Sottosistema Gestione Paziente

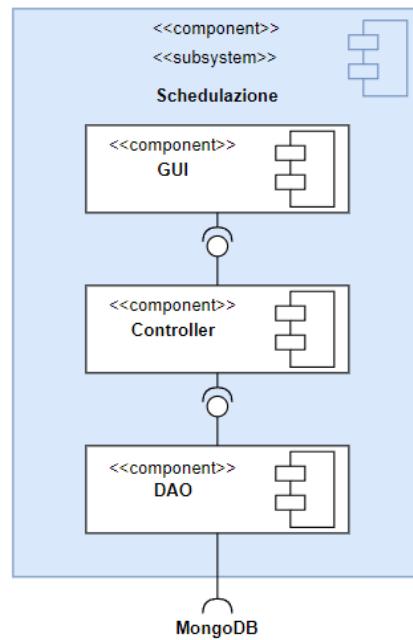


3.2.3 Sottosistema Gestione Medicinale



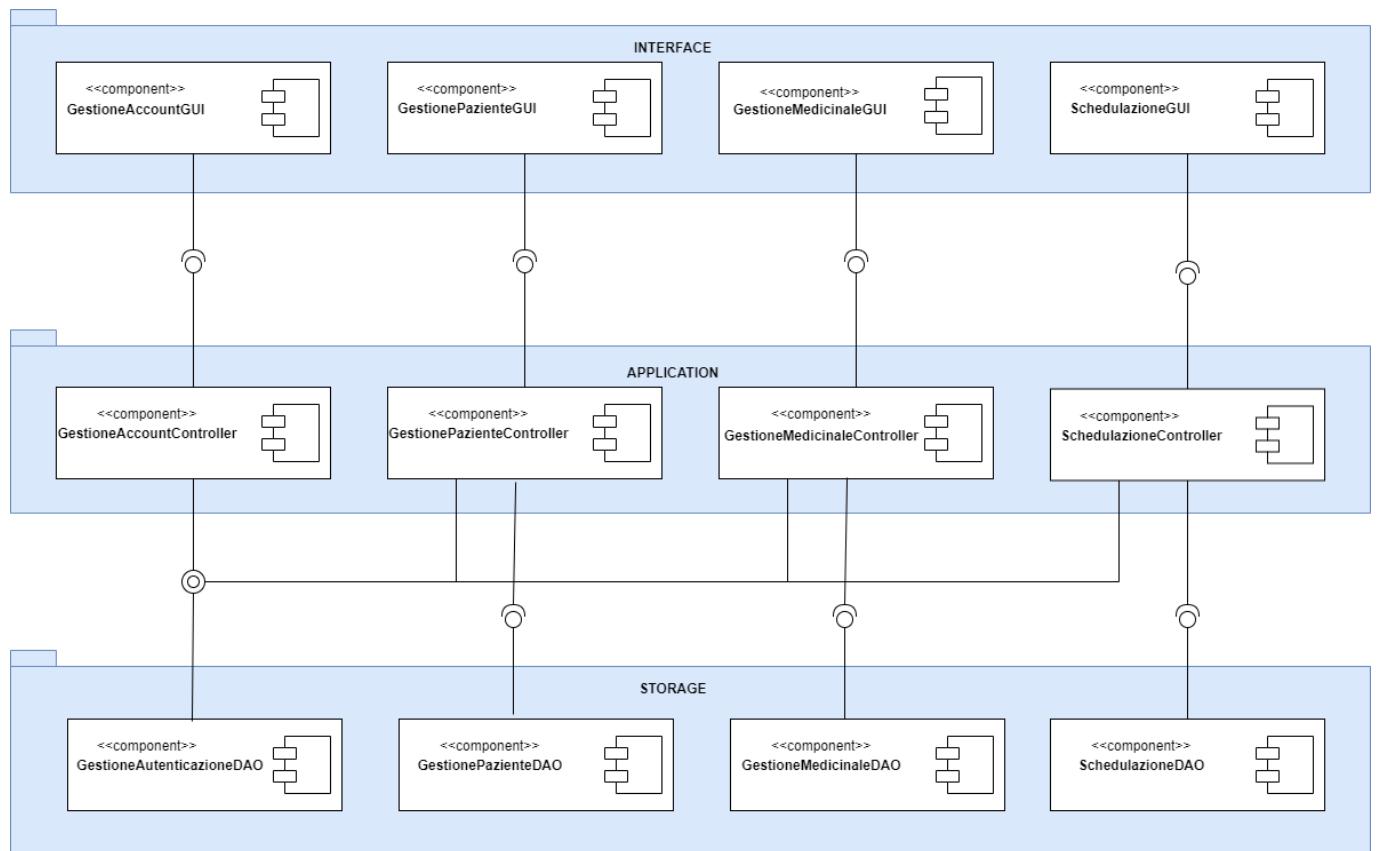


3.2.4 Sottosistema Schedulazione





3.2.5 Diagramma Architetturale

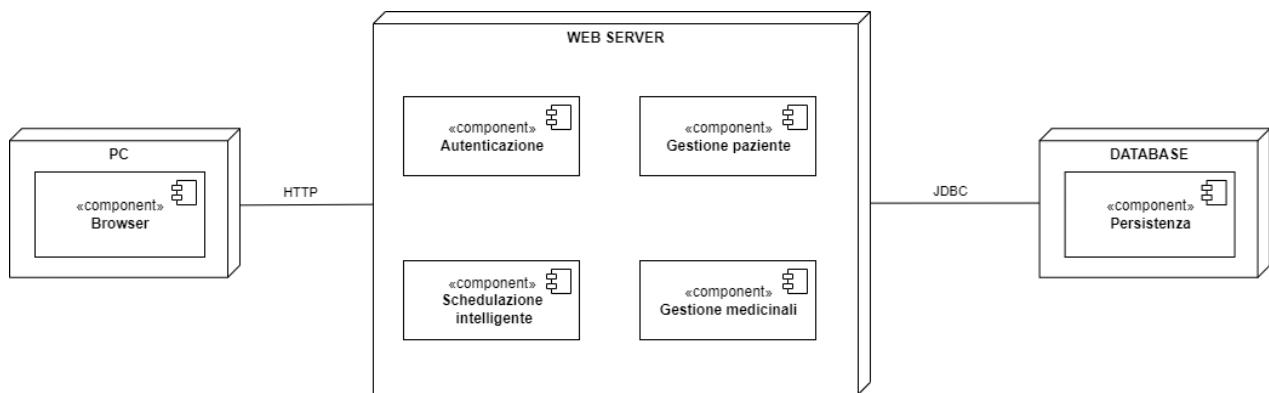


3.3 Mapping hardware/software

L'applicazione web si basa su una piattaforma hardware in cui un server risponde alle richieste effettuate dai computer dell'ospedale.

Essendo che l'applicazione si basa su un'architettura non distribuita, è necessario un solo nodo per il server dell'applicazione.

Di seguito è riportato un UML Deployment Diagram che descrive il mapping hardware/software:

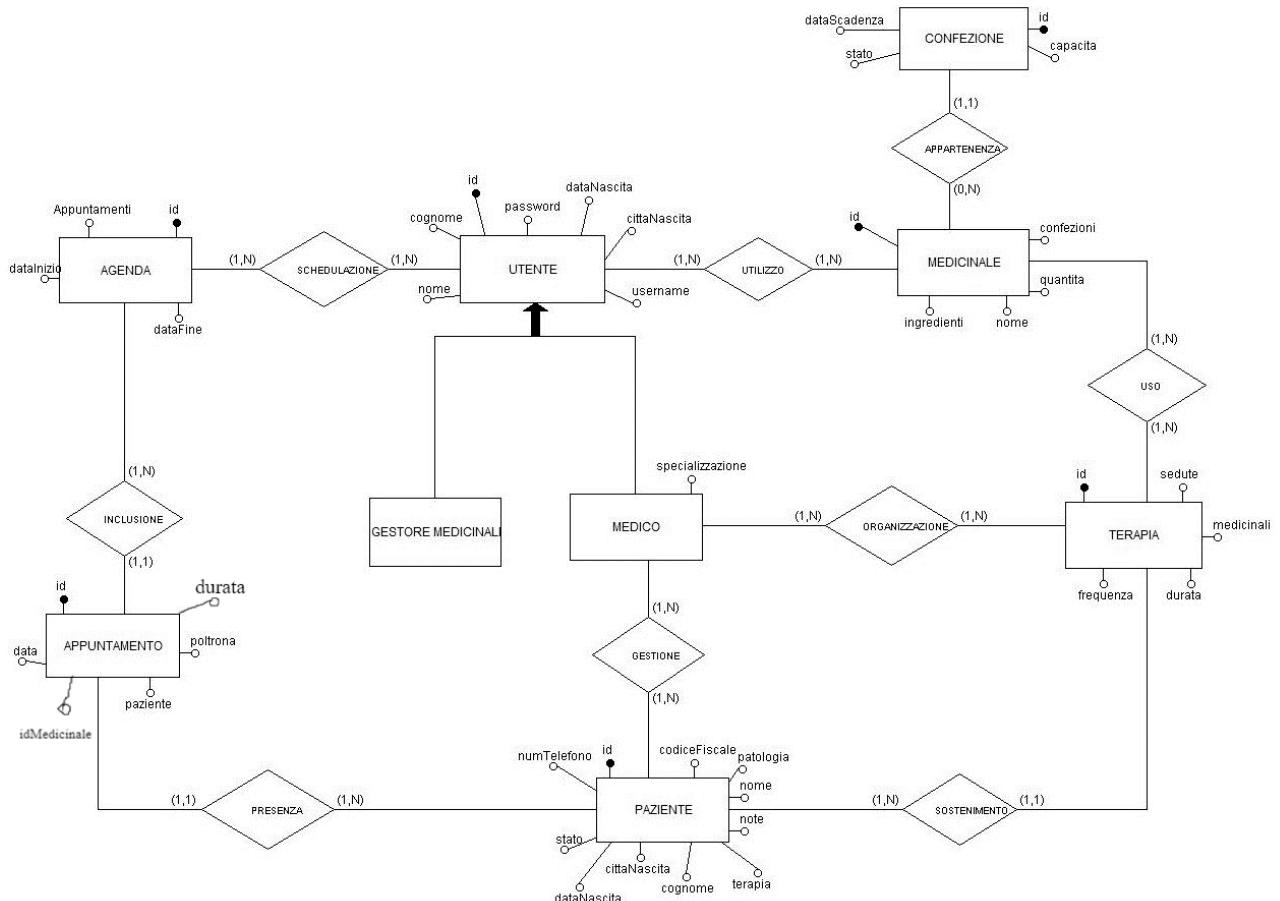


3.4 Gestione dei dati persistenti

Per la memorizzazione dei dati è stato scelto di utilizzare il DBMS non relazionale MongoDB, orientato ai documenti. Il vantaggio della scelta di un database di questo tipo consiste nella possibilità di conservare le informazioni necessarie in documenti JSON, ciò permette di raggruppare in collezioni i documenti, i quali vengono inclusi in esse senza uno schema fisso. Tale caratteristica consente di gestire le entità in modo più efficiente.

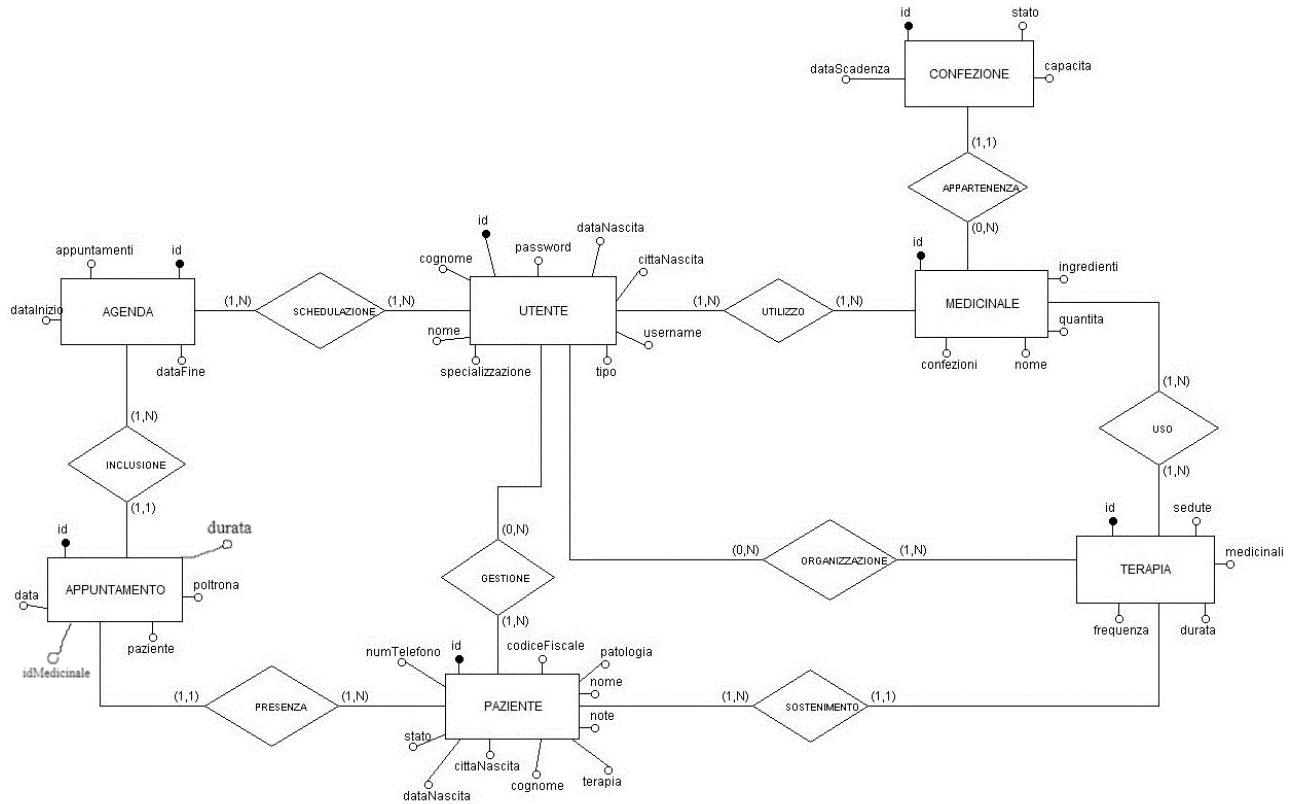
3.4.1 Descrizione delle entità persistenti

3.4.1.1 Schema E-R





3.4.1.2 Schema E-R ristrutturato



3.4.2 Dizionario dei dati

3.4.2.1 Utente

UTENTE	
ATTRIBUTO	TIPO
Id	Object
Username	String
Password	String
Nome	String
Cognome	String
CittàNascita	String
DataNascita	Date
Tipo	Int



Specializzazione

String

La collection Utente include i medici e i gestori dei medicinali, le informazioni anagrafiche relative ad essi e la loro specializzazione.

3.4.2.2 Paziente

PAZIENTE	
ATTRIBUTO	TIPO
Id	Object
CodiceFiscale	String
Nome	String
Cognome	String
CittaNascita	String
DataNascita	Date
NumTelefono	String
Stato	Boolean
Patologia	String
Terapia	Terapia
Note	String

La collection Paziente contiene le informazioni anagrafiche relative ai pazienti e la terapia a cui si stanno sottoponendo, la patologia di cui soffrono e il loro stato, oltre a eventuali note.

3.4.2.3 Medicinale

MEDICINALE	
ATTRIBUTO	TIPO
Id	Object
Nome	String
Ingredienti	String
Quantità	Int



Confezioni | Confezione[]

Questa collection contiene i dati riguardanti i medicinali, una lista e la quantità delle confezioni memorizzate relativi ad essi.

3.4.2.4 Agenda

AGENDA	
ATTRIBUTO	TIPO
Id	Object
DataInizio	Date
DataFine	Date
Appuntamenti	Appuntamento[]

La collection Agenda racchiude i dati riguardanti data di inizio e fine dell'agenda settimanale e una lista degli identificativi degli appuntamenti.

3.4.2.5 Appuntamento

APPUNTAMENTO	
ATTRIBUTO	TIPO
Id	Object
Data	Date
Paziente	String
Poltrona	String
Durata	Int
Id medicinale	String

La collection Appuntamento include le informazioni su data e ora di ogni appuntamento di un dato paziente e la poltrona che occuperà.



3.4.2.6 Terapia

TERAPIA	
ATTRIBUTO	TIPO
Id	Object
Frequenza	Int
Durata	Int
Medicinali	Object[]
Sedute	Int

In questa collection sono presenti i dati relativi alle terapie, in particolare qui vengono forniti il numero delle sedute totali che il paziente dovrà sostenere, i medicinali utilizzati, la frequenza settimanale e la durata di ogni seduta.

3.4.2.7 Confezione

CONFEZIONE	
ATTRIBUTO	TIPO
Id	String
DataScadenza	Data
Capacità	Int
Stato	Boolean

La collection Confezione contiene i dati di ogni confezione di medicinale, tra cui data di scadenza, capacità della confezione e lo stato di essa, che indica se possa essere utilizzata o meno.

3.5 Controllo degli accessi e sicurezza

Di seguito è riportata la matrice degli accessi che ci mostra, per ogni sottosistema, a quali funzionalità può accedere ogni attore:

Oggetti	Autenticazion e	Gestione Paziente	Schedulazion e	Gestione Medicinali
---------	--------------------	-------------------	-------------------	---------------------



Attori				
Gestore Medicinali	Login Logout ModificaPassword		VisualizzaAgen da	AggiungiMedicinale RimuoviMedicinale RicercaMedicinale VisualizzaListaMedicinali AggiungiCoonfezioneMedicinale RimuoviConfezioneMedicinali
Medico	Login Logout ModificaPassword	AggiungiPaziente EliminaPaziente ModificaPaziente RicercaPaziente VisualizzaListaPazienti ModificaStatoPaziente	CreaAgenda VisualizzaAgen da	RicercaMedicinale VisualizzaListaMedicinali

3.6 Controllo globale del software

Il sistema Chemo è altamente iterativo, poiché viene continuamente richiesta un'interazione tra client e server; l'interazione, in particolare, avviene tramite l'uso di un'interfaccia grafica, realizzata con HTML e CSS.

Per il controllo delle richieste, il sistema utilizzerà un meccanismo event-drive, cioè: le richieste vengono generate dal Client e una Servlet, che si occupa di gestire l'evento, avrà il compito di inizializzarle e inoltrarle alle classi opportune. Una volta che la richiesta sarà realizzata ed elaborata, la Servlet di riferimento inoltrerà la risposta al relativo Client.



3.7 Boundary conditions

3.7.1 Inizializzazione del sistema

Identificativo BC_01_InizializzazioneSistema	Inizializzazione del sistema	Data	28/11/22
		Vers.	1.0.0
		Autore	Marcone Antonio
Descrizione	Lo UC descrive l'avvio del sistema		
Attore Principale	Personale Amministrativo		
Attori Secondari	NA		
Entry Condition	Il sistema è pronto per essere avviato		
Exit condition			
On success	Il sistema viene avviato con successo		
Exit condition			
On failure	Il sistema non viene avviato, generando un messaggio di errore		
Rilevanza/User Priority	Bassa		
Frequenza stimata	1/mese		
Extension point	NA		
Generalization of	NA		

FLUSSO DI EVENTI PRINCIPALE

1	Amministratore:	Esegue il comando di avvio del sistema sulla macchina su cui viene eseguito.
2	Sistema:	Avvia la connessione al database
3	Sistema:	Verifica l'integrità dei dati salvati del database
4	Sistema:	Avvia il web server

I Scenario/Flusso di eventi di ERRORE: fallita connessione al database

2.1	Sistema:	Visualizza un messaggio di errore relativo alla fallita connessione con il database
2.2	Amministratore:	Corregge l'errore mostrato dal sistema
2.3	Amministratore:	Riprende dal punto 1

II Scenario/Flusso di eventi di ERRORE: fallita verifica di integrità del database



3.1	Sistema:	Visualizza un messaggio di errore relativo all'integrità dei dati persistenti salvati nel database
3.2	Amministratore:	Corregge l'errore mostrato dal sistema
3.3	Amministratore:	Riprende dal punto 1

III Scenario/Flusso di eventi di ERRORE: fallito avvio del webserver

4.1	Sistema:	Visualizza un messaggio di errore relativo al fallito avvio del web server
4.2	Amministratore:	Corregge l'errore mostrato dal sistema
4.3	Amministratore:	Riprende dal punto 1

3.7.2 Arresto del sistema

Identificativo BC_02_ArrestoSistema	Arresto del sistema	Data	28/11/22	
		Vers.	1.0.0	
		Autore	Marcone Antonio	
Descrizione		Lo UC descrive l'arresto del sistema		
Attore Principale		Personale Amministrativo		
Attori Secondari		NA		
Entry Condition		Il sistema è in esecuzione		
Exit condition On success		Il sistema viene arrestato con successo		
Exit condition On failure		Il sistema non viene arrestato, generando un messaggio di errore		
Rilevanza/User Priority		Bassa		
Frequenza stimata		1/mese		
Extension point		NA		
Generalization of		NA		
FLUSSO DI EVENTI PRINCIPALE				
1	Amministratore:	Esegue il comando di arresto del sistema sulla macchina su cui viene eseguito.		



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

2	Sistema:	Controlla che non ci siano connessioni attive
3	Sistema:	Arresta il web server

I Scenario/Flusso di eventi ALTERNATIVO: presenza di connessioni aperte

2.1	Sistema:	Notifica che ci sono connessioni ancora aperte
2.2	Sistema:	Blocca la creazione di nuove connessioni aspettando che le connessioni aperte vengano chiuse
2.3	Sistema:	Verifica la presenza di connessioni aperte e, se non ci sono, arresta il sistema
2.4	Sistema	Visualizza un messaggio di arresto del sistema avvenuto con successo

II Scenario/Flusso di eventi ALTERNATIVO: terminazione delle connessioni aperte

2.3.1	Sistema:	Termina le connessioni ancora aperte
2.3.2	Sistema	Visualizza un messaggio di arresto del sistema avvenuto la terminazione forzata di alcune connessioni

I Scenario/Flusso di eventi di ERRORE: fallito arresto del webserver

3.1	Sistema:	Visualizza un messaggio di errore relativo al fallito arresto del web server
3.2	Amministratore:	Corregge l'errore mostrato dal sistema
3.3	Amministratore:	Riprende dal punto 1

3.7.3 Crash del sistema

Identificativo BC_03_CrashSistema	Crash del sistema	Data	28/11/22	
		Vers.	1.0.0	
		Autore	Marcone Antonio	
Descrizione		Lo UC descrive il comportamento del sistema in caso di crash del server		
Attore Principale		Personale Amministrativo		
Attori Secondari		NA		
Entry Condition		Il sistema termina improvvisamente		
Exit condition		Il sistema torna operativo con successo		



On success	
Exit condition	
On failure	Il sistema non torna operativo
Rilevanza/User Priority	Bassa
Frequenza stimata	4/anno
Extension point	NA
Generalization of	NA

FLUSSO DI EVENTI PRINCIPALE

1	Amministratore:	Esegue il comando di avvio del sistema sulla macchina su cui viene eseguito.
2	Sistema:	Mostra un messaggio di errore relativo al crash
3	Amministratore:	Ripristina la versione precedente del sistema risolvendo gli errori mostrati dal sistema
4	Amministratore:	Segue gli step per avviare il sistema in BC_01_InizializzazioneSistema



4. Glossario

- **Chemo:** nome dell'applicativo che si andrà a realizzare;
- **UX:** Indica l'esperienza utente, ovvero l'interazione tra sistema e utente;
- **8 regole d'oro di Shneiderman:** sono regole applicabili ad ogni ambiente di interazione al fine di migliorare l'approccio del progettista all'applicazione.
- **Controller:** componente software che riceve i comandi dell'utente (in genere attraverso un'altra componente software) e, sulla base di questi, modifica lo stato di altre componenti.
- **Sottosistema:** sottoinsieme dei servizi del dominio applicativo.