



System Design Document

MedConnect

Riferimento	C05_SDD
Versione	3.0
Data	17/11/2024
Destinatario	Ministero della Salute
Presentato da	Cusati Daniel (Mat. 05121 12422) Medica Vincenzo (Mat. 05121 16808) Varone Giuseppe (Mat. 05121 17831)

Revision History

Data	Versione	Cambiamenti	Autori
18/11/2024	0.1	Aggiunta di Capitolo 1	Cusati, Medica, Varone
18/11/2024	0.1	Aggiunta di Capitolo 2	Cusati, Medica, Varone
18/11/2024	0.2	Aggiunta di Capitolo 3 Paragrafo 3.1	Cusati, Medica, Varone
18/11/2024	0.3	Aggiunta di Capitolo 3 Paragrafo 3.2	Cusati, Medica, Varone
18/11/2024	0.4	Aggiunta di Capitolo 3 Paragrafo 3.3	Cusati, Medica, Varone
18/11/2024	0.5	Aggiunta di Capitolo 3 Paragrafo 3.4	Cusati, Medica, Varone
18/11/2024	0.6	Aggiunta di Capitolo 3 Paragrafo 3.5	Cusati, Medica, Varone
18/11/2024	0.6	Aggiunta di Capitolo 3 Paragrafo 3.6	Cusati, Medica, Varone
18/11/2024	0.6	Aggiunta di Capitolo 3 Paragrafo 3.7	Cusati, Medica, Varone
18/11/2024	0.7	Aggiunta di Capitolo 4	Cusati, Medica, Varone
12/12/2024	0.8	Aggiunta di Capitolo 5	Cusati, Medica, Varone
20/12/2024	0.9	Aggiunta di Capitolo 6	Cusati, Medica, Varone
27/12/2024	1.0	Revisione	Cusati, Medica, Varone
30/12/2024	2.0	Revisione	Cusati, Medica, Varone
06/01/2024	3.0	Aggiunto logo e ultima revisione	Cusati, Medica, Varone

Sommario

Revision History	2
Sommario	3
1 Introduzione	4
1.1 Obiettivi del sistema	4
1.2 Design Goals	4
1.2.1 Trade-off	10
1.3 Definizioni, acronimi e abbreviazioni	10
1.4 Riferimenti	11
1.5 Panoramica	11
2 Architettura del Sistema Corrente	12
3 Architettura del Sistema Proposto	12
3.1 Panoramica	12
3.2 Decomposizione in sottosistemi	12
3.2.1 Component Diagram	13
(sta su draw.io)	13
3.2.2 Architectural Diagram	13
3.2.2.1 Sottosistema Registrazione	13
3.2.2.2 Sottosistema Autenticazione	14
3.2.2.3 Sottosistema Gestione Utente	14
3.2.2.4 Sottosistema Gestione Prenotazione	14
3.2.2.5 Sottosistema Gestione Recensione	14
3.3 Mapping Hardware/Software	14
3.3.1 Deployment Diagram	14
3.4 Gestione dati persistenti	15
3.4.1 Dizionario dei dati	16
3.5 Controllo degli accessi e sicurezza	20
3.6 Controllo globale del software	22
3.7 Condizioni limite	23
4 Servizi dei sottosistemi	23
4.1 Registrazione	23
4.2 Autenticazione	25
4.3 Gestione Utente	25



4.4 Gestione Prenotazione	27
4.5 Gestione Recensione	28
5 Glossario	29
6 Design Pattern	30
6.1 Facade Design Pattern	30
6.2 Proxy Design Pattern	31

1 Introduzione

1.1 Obiettivi del sistema

MedConnect nasce con l'intento di rivoluzionare la gestione delle prenotazioni mediche, puntando a rendere l'interazione tra cittadini e sistema sanitario più semplice, veloce ed efficace. Grazie a un approccio tecnologico avanzato, il sistema ambisce a ottimizzare i processi di prenotazione, migliorando l'accessibilità ai servizi sanitari e promuovendo la trasparenza nell'interazione con i professionisti della salute.

Il progetto si propone di rendere i servizi sanitari più efficienti attraverso l'introduzione di strumenti digitali che automatizzano attività manuali e semplificano l'organizzazione delle risorse.

MedConnect offre le seguenti funzionalità principali:

- Prenotazioni semplificate: consente ai cittadini di fissare appuntamenti in modo rapido e intuitivo, migliorando l'accesso ai servizi medici.
- Valutazioni dei professionisti: permette di lasciare recensioni e feedback sui medici, incentivando la trasparenza e contribuendo al miglioramento della qualità del servizio.
- Ottimizzazione dei tempi di attesa: implementa sistemi automatizzati per ridurre i tempi necessari per ottenere una prenotazione, garantendo una gestione delle risorse più efficiente.

1.2 Design Goals

Nella presente sezione si andranno a presentare i Design Goals, ovvero le qualità sulle quali il sistema deve essere focalizzato, formalizzati esplicitamente così che qualsiasi importante decisione di design può essere fatta consistentemente seguendo lo stesso insieme di design goal. I design goal sono stati suddivisi nelle seguenti categorie:

- **Performance:** includono i requisiti di velocità e di spazio imposti al sistema;
- **Dependability:** determinano quanto sforzo deve essere speso per ridurre al minimo i crash del sistema e le loro conseguenze;
- **Maintenance:** determina quanto sia difficile cambiare il sistema dopo l'implementazione;
- **End User:** includono qualità che sono desiderabili dal punto di vista dell'utente, ma che non sono state coperte dai criteri di Performance e Dependability.

Ciascun design goal è descritto da:

- **Rank**, che ne specifica un valore di priorità compreso tra 1 e 13 (1 massima e 13 minima);
- **ID**, un identificatore e un nome univoco;
- **Descrizione**, una spiegazione;
- **Categoria**, indica la classificazione in base al gruppo di appartenenza;
- **RNF di origine**, specifica il requisito o i requisiti non funzionali da cui è stato derivato.

Rank	ID	Descrizione	Categoria	RNF di origine
4	DG_1 Tempi di risposta	Il sistema deve garantire un tempo di risposta non superiore a 5 secondi.	Performance	RNF_P_2
8	DG_2 Facilità d'uso	Il sistema deve risultare semplice da usare comprensibile con l'uso delle "8 regole d'oro di Shneiderman" per il design delle interfacce grafiche.	End User	RNF_U_1
9	DG_3 Interfaccia Intuitiva	L'interfaccia utente della piattaforma deve consentire di svolgere azioni in modo intuitivo e semplice.	End User	RNF_U_2

13	DG_4 Supporto Utenti	Il sistema dovrebbe aiutare gli utenti a risolvere facilmente i principali dubbi e problemi attraverso una sezione FAQ facilmente accessibile.	End User	RNF_U_3
7	DG_5 Operazioni	Il sistema deve garantire che le operazioni siano eseguite correttamente attraverso la validazione dei dati e i controlli nelle operazioni.	Dependability	RNF_A_1
6	DG_6 Errori di sistema	Il sistema deve gestire gli errori commessi dall'utente, mostrandogli i passi necessari per guidarli verso la risoluzione.	Dependability	RNF_A_2

2	DG_7 Gestione permessi sistema	Il sistema deve consentire agli utente di interagire con esso solamente tramite le operazioni consentite a quest'ultimo impedendogli di andare in stati non gestiti.	Dependability	RNF_A_3
3	DG_8 Disponibilità	Il Sistema deve garantire la massima disponibilità, con un limite di 48 ore all'anno per interventi straordinari.	Dependability	RNF_P_1
5	DG_9 Produttività	Il sistema deve garantire la disponibilità continua anche durante la gestione di un massimo di 1.650 prenotazioni al giorno, prevedendo tolleranza ai guasti e meccanismi di recovery per evitare downtime.	Dependability	RNF_P_3

12	DG_10 Adattabilità e manutenibilità	Il sistema deve essere progettato con un'architettura modulare e flessibile, che supporti facilmente nuove funzionalità e integrazioni, garantendo al contempo una manutenzione efficiente, attraverso l'adozione di best practice come il principio di separazione delle preoccupazioni (SoC) e l'uso di microservizi per favorire la scalabilità e la gestione delle modifiche senza compromettere la stabilità.	Mantenanc e	RNF_S_1 e RNS_S_2
----	---	--	----------------	----------------------

10	DG_11 Compatibilità browser	Il sistema deve essere progettato per garantire la compatibilità con i principali web browser che supportano i principali linguaggi moderni, offrendo un'esperienza utente uniforme e accessibile su diverse piattaforme.	End User	RNF_V_IM_1
11	DG_12 Compatibilità sistemi operativi	Il sistema deve essere progettato per garantire la portabilità, utilizzando tecnologie cross-platform, come framework e librerie compatibili, al fine di assicurare un'esperienza utente consistente su tutte le piattaforme supportate.	End User	RNF_V_IM_2

1	DG_13 Privacy e gestione informazioni	Il sistema deve garantire la protezione dei dati, impiegando protocolli di comunicazione sicuri, archiviando in modo protetto i dati persistenti e assicurando l'accesso ai dati solo agli utenti autorizzati.	Dependability	RNF_V_L_1 e RNF_A_4 e RNF_V_IN_1
---	--	--	---------------	----------------------------------

1.2.1 Trade-off

Trade-off	Descrizione
Sicurezza vs Velocità	Si preferisce investire nella sicurezza complessiva del sistema ciò a discapito della velocità accettando un tempo di risposta non superiore a 10 secondi.
Supporto utenti vs tempi di sviluppo	La creazione di una sezione FAQ completa e accessibile può ritardare il rilascio del sistema, poiché richiede un investimento significativo per anticipare e documentare i problemi più comuni.

1.3 Definizioni, acronimi e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti nel documento corrente:

- **Sottosistema:** un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una relazione funzionale.
- **Design Goal:** le qualità sulle quali il sistema deve essere focalizzato.
- **Dati Persistenti:** dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati.
- **Mapping Hardware/Software:** studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.
- **RAD:** Requirements Analysis Document.
- **SDD:** System Design Document.
- **DG:** Design Goals.

1.4 Riferimenti

Vengono riportati di seguito dei riferimenti ad altri documenti utili durante la lettura:

- C05_RAD
- Libro di Bruegge, Bernd, and Allen H. Dutoit. "Object-oriented software engineering. using uml, patterns, and java."

1.5 Panoramica

Questo documento di System Design è organizzato in quattro sezioni principali:

- **Introduzione:** Fornisce una panoramica dello scopo del sistema e degli obiettivi di design che si intendono raggiungere.
- **Architettura software corrente:** illustra lo stato attuale dell'architettura del software esistente.
- **Architettura software proposta:** Descrive la definizione e la suddivisione del sistema in sottosistemi, il mapping tra hardware e software, e la

gestione dei dati persistenti. Include inoltre la struttura dettagliata dei sottosistemi e le condizioni limite che riguardano l'intero sistema.

- **Glossario:** Presenta un elenco dei termini utilizzati nel documento con le relative definizioni.
- **Design Pattern:** Presentano soluzioni riutilizzabili per problemi comuni di progettazione software.

2 Architettura del Sistema Corrente

Attualmente, esistono diverse piattaforme che consentono agli utenti di prenotare visite mediche e gestire appuntamenti sanitari. Tra le più conosciute troviamo MioDottore, Doctolib e iDoctors, che offrono servizi come la ricerca di medici e la gestione delle prenotazioni. Tuttavia, queste soluzioni sono sviluppate da aziende private e non sono direttamente collegate o supportate dal Ministero della Salute.

MedConnect, invece, si propone come una soluzione commissionata direttamente dal Ministero della Salute, che oltre a offrire un accesso semplificato e rapido ai servizi medici, introduce funzionalità avanzate. Tra queste spiccano un sistema centralizzato per la raccolta e la pubblicazione di recensioni sui medici e meccanismi per ottimizzare l'allocazione delle risorse sanitarie, garantendo così maggiore trasparenza e una significativa riduzione dei tempi di attesa.

Grazie a questa visione innovativa, MedConnect si pone come un ponte diretto tra i cittadini e il sistema sanitario nazionale, migliorandone l'efficienza complessiva.

3 Architettura del Sistema Proposto

3.1 Panoramica

Per il sistema proposto, si adotta un'architettura a tre livelli (three-tier). Questa architettura risulta particolarmente adatta alle nostre esigenze, in quanto consente di suddividere il sistema in modo gerarchico, offrendo così un maggiore controllo e una gestione più efficace delle componenti.

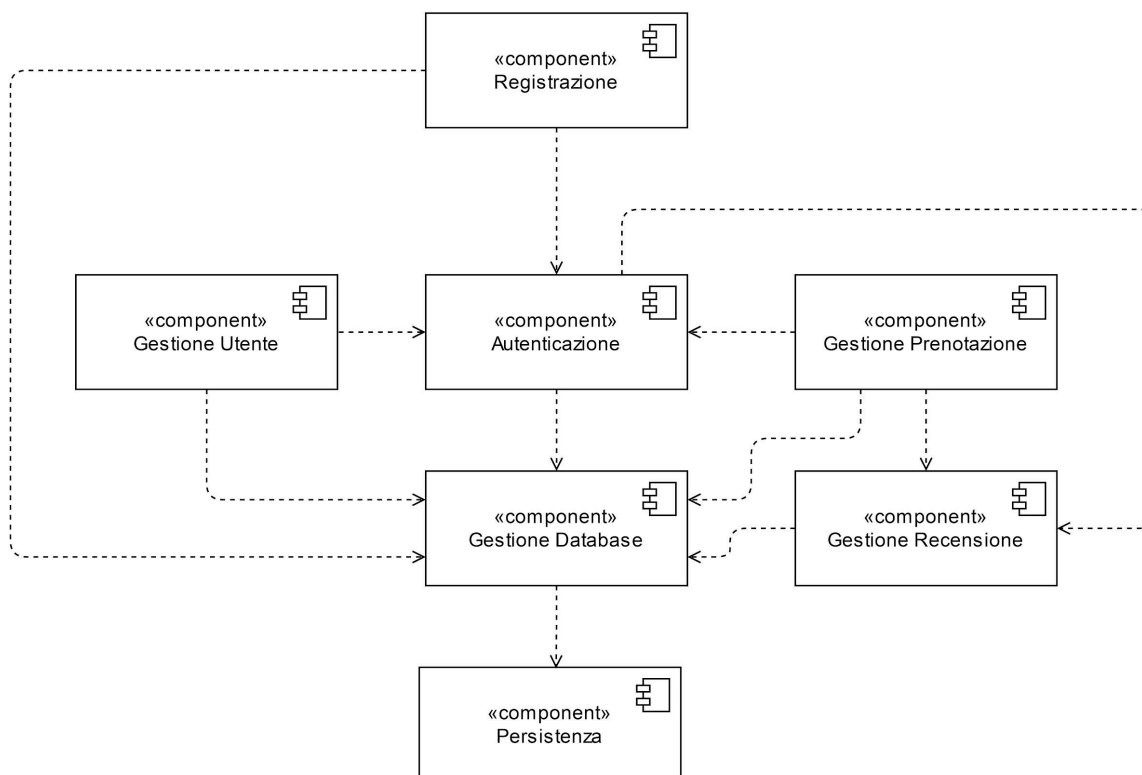
3.2 Decomposizione in sottosistemi

I sottosistemi da noi individuati sono:

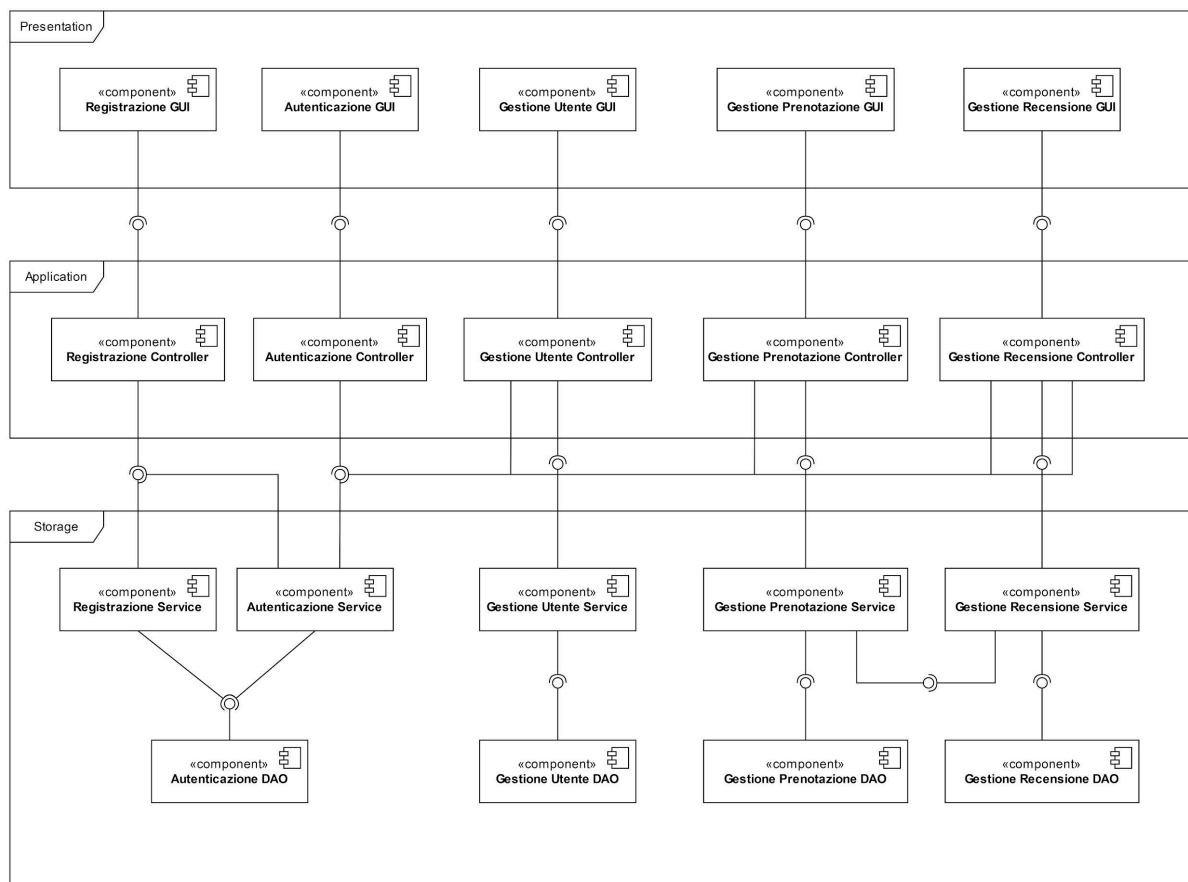
- **Registrazione:** si occupa di gestire la registrazione dei vari utenti;
- **Autenticazione:** si occupa di autenticare gli utenti registrati;
- **Gestione Utente:** si occupa di gestire svariate funzionalità tra cui l'inserimento, la modifica ed eliminazione di informazioni di base, cartelle cliniche e specializzazioni, nonché la ricerca e visualizzazione dei profili medici e dei pazienti.
- **Gestione Prenotazione:** si occupa di gestire svariate funzionalità tra cui ciò che riguarda la prenotazione come visualizzare, effettuare, eliminare e modificare una prenotazione;
- **Gestione Recensione:** si occupa della gestione delle varie recensioni scritte dai pazienti;
- **Gestione Database:** si occupa di gestire le interazioni tra il sistema e il servizio di persistenza;
- **Persistenza:** si occupa di gestire l'archiviazione dei dati attraverso un database.

3.2.1 Component Diagram

Di seguito sono mostrate le dipendenze tra i sottosistemi attraverso un component diagram UML.



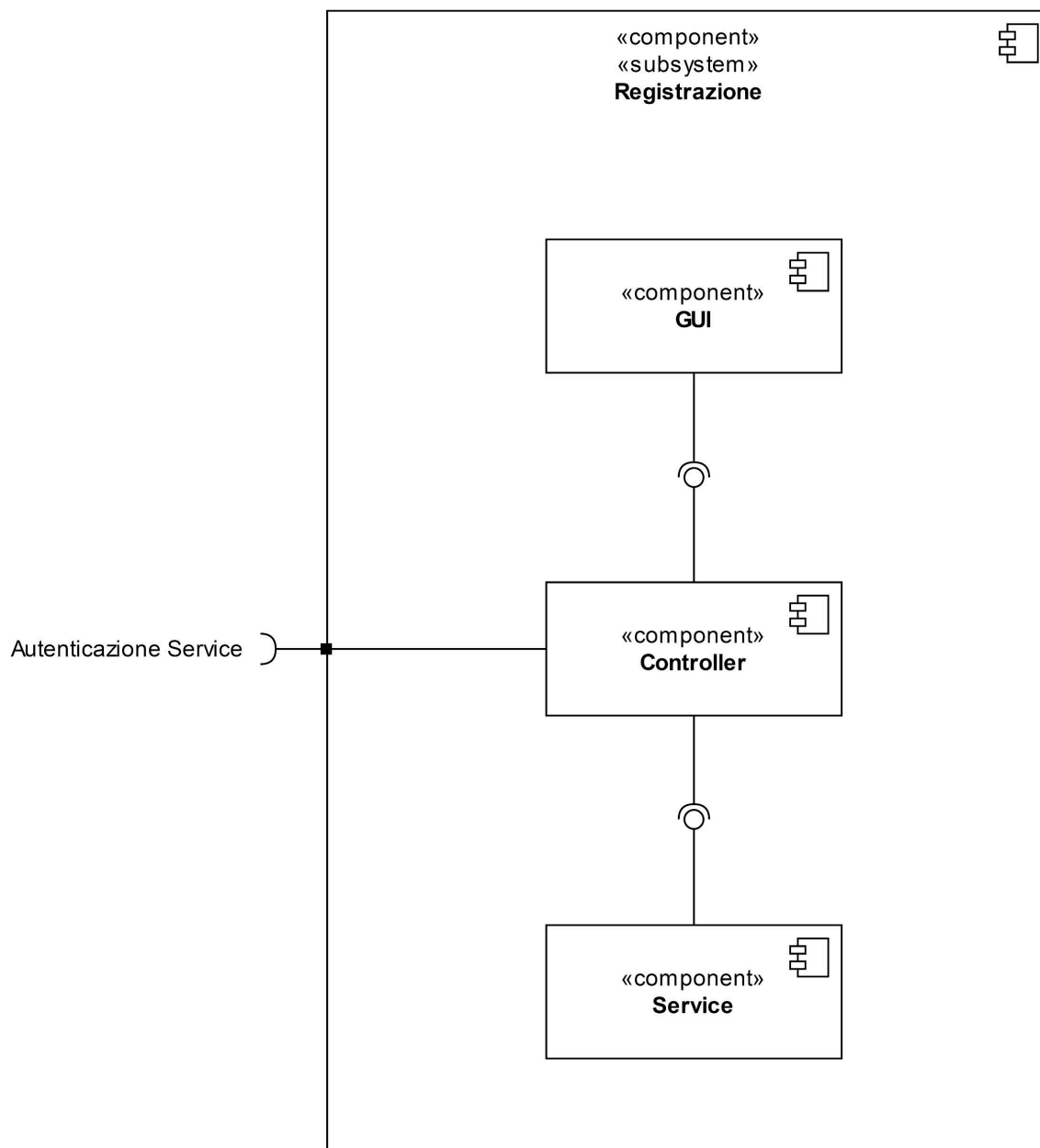
3.2.2 Architectural Diagram



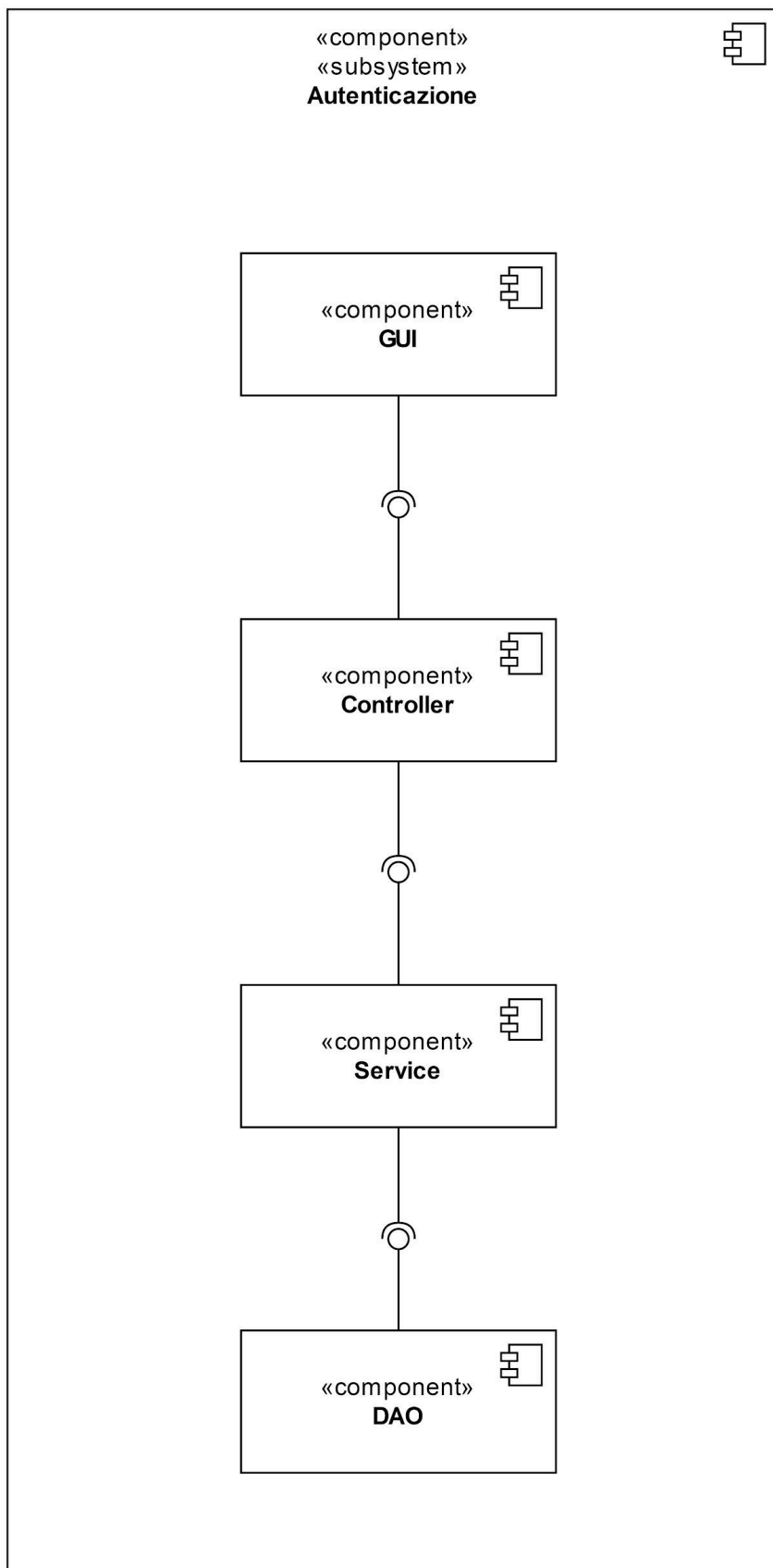
Componenti del diagramma architettuale:

- **GUI (Graphic User Interface)**: rappresenta la logica di presentazione;
- **Controller**: rappresenta la logica di controllo;
- **Service**: rappresenta la logica di business;
- **DAO (Data Access Object)**: rappresenta la logica di accesso ai dati persistenti.

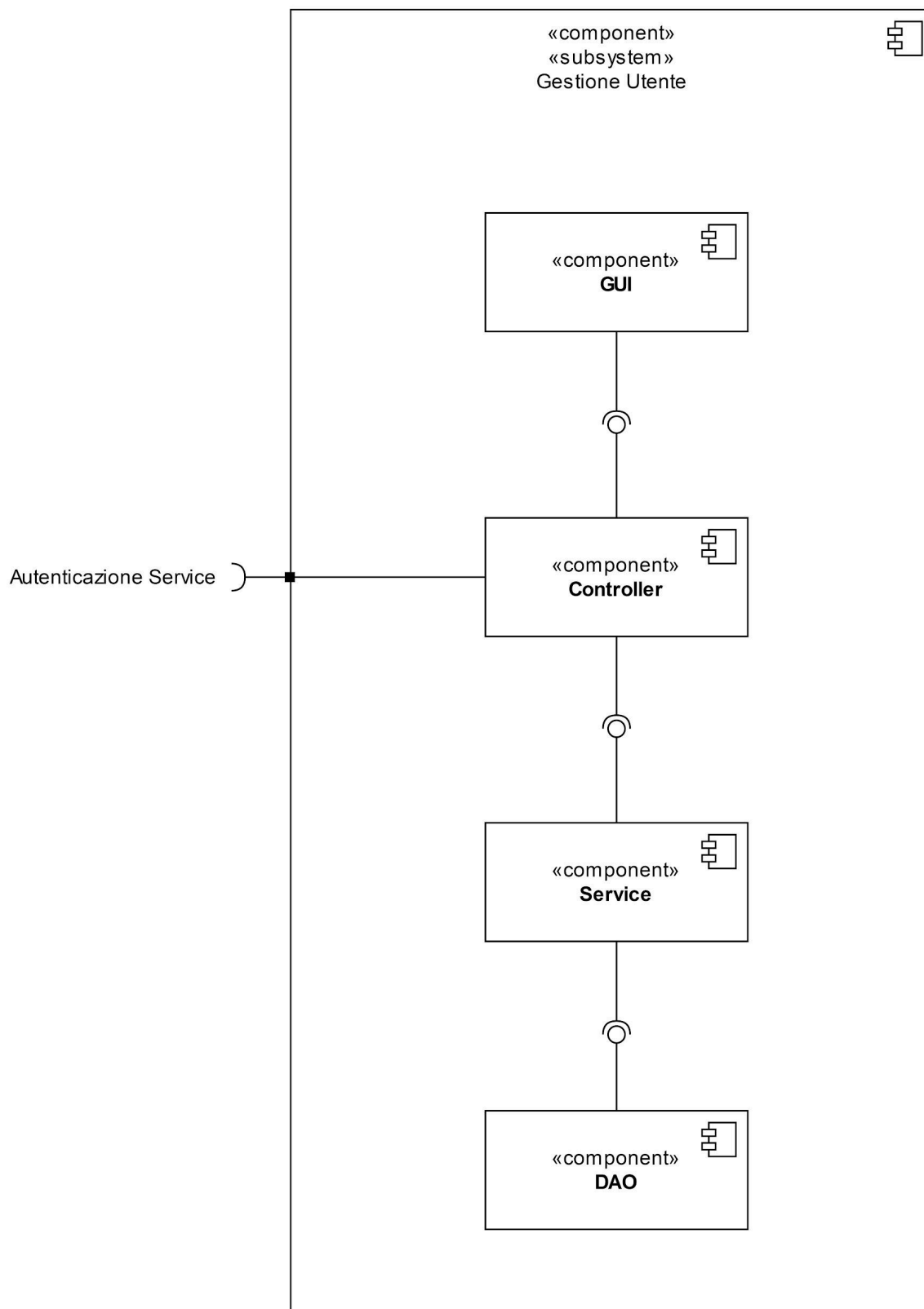
3.2.2.1 Sottosistema Registrazione



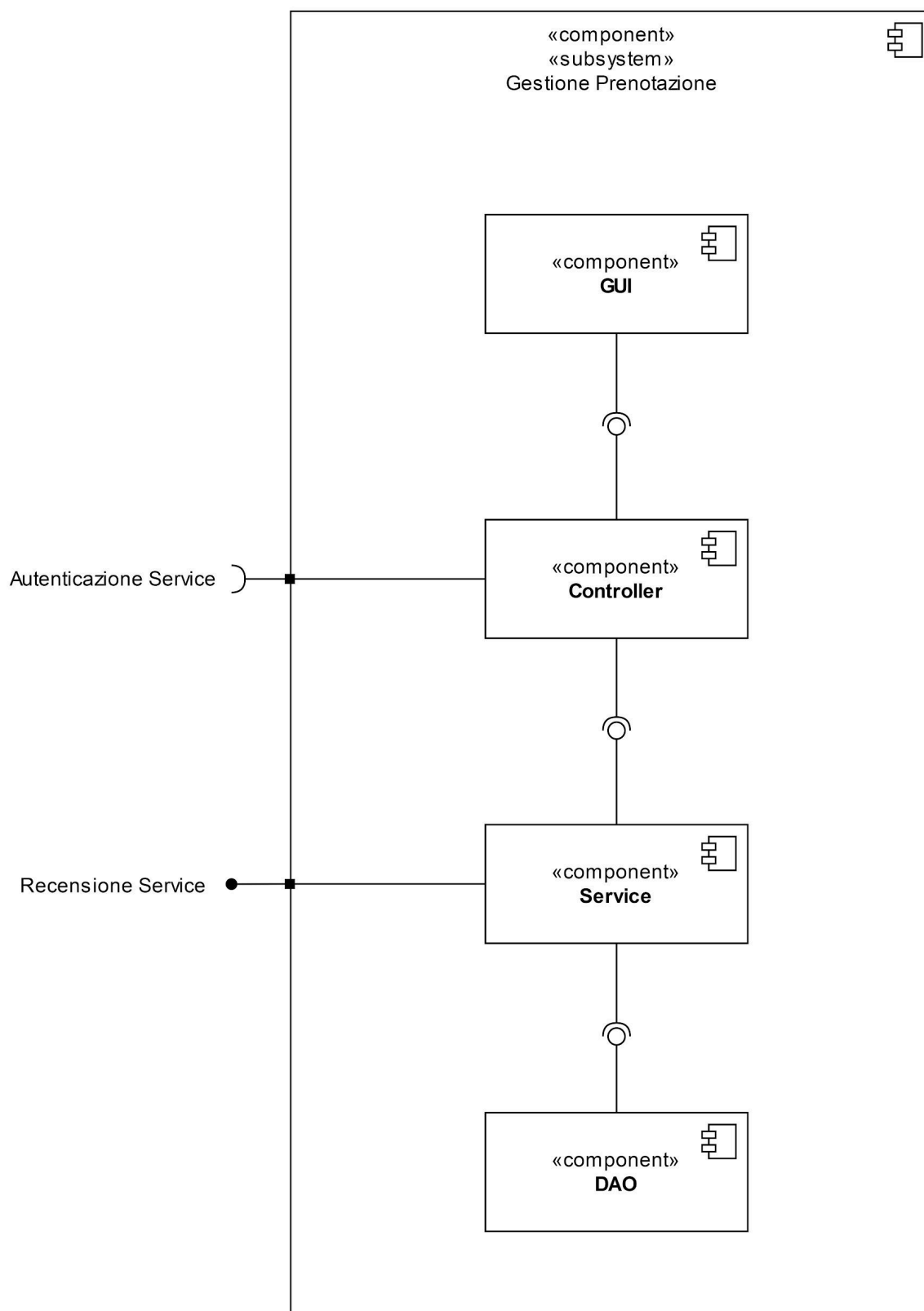
3.2.2.2 Sottosistema Autenticazione



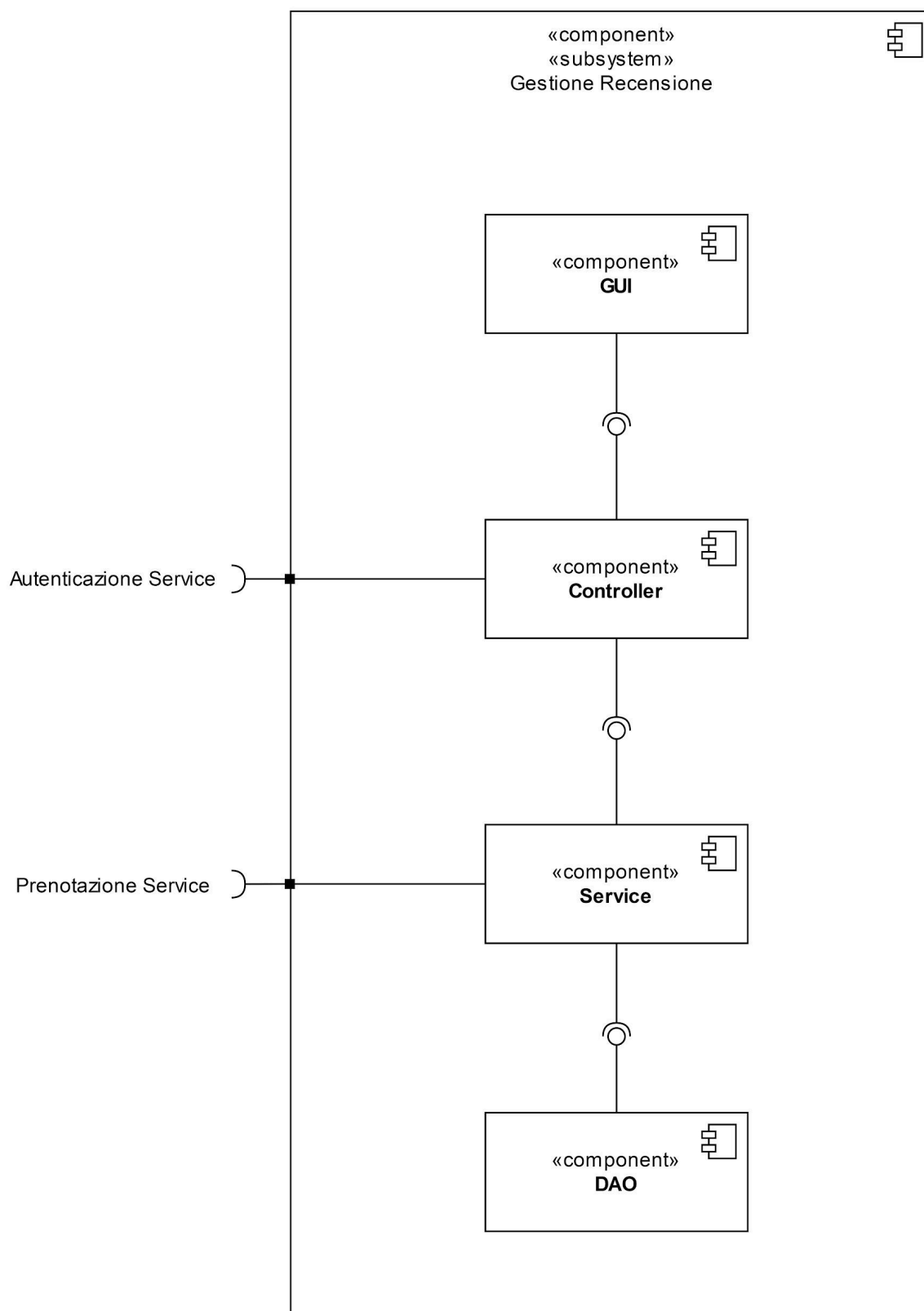
3.2.2.3 Sottosistema Gestione Utente



3.2.2.4 Sottosistema Gestione Prenotazione



3.2.2.5 Sottosistema Gestione Recensione

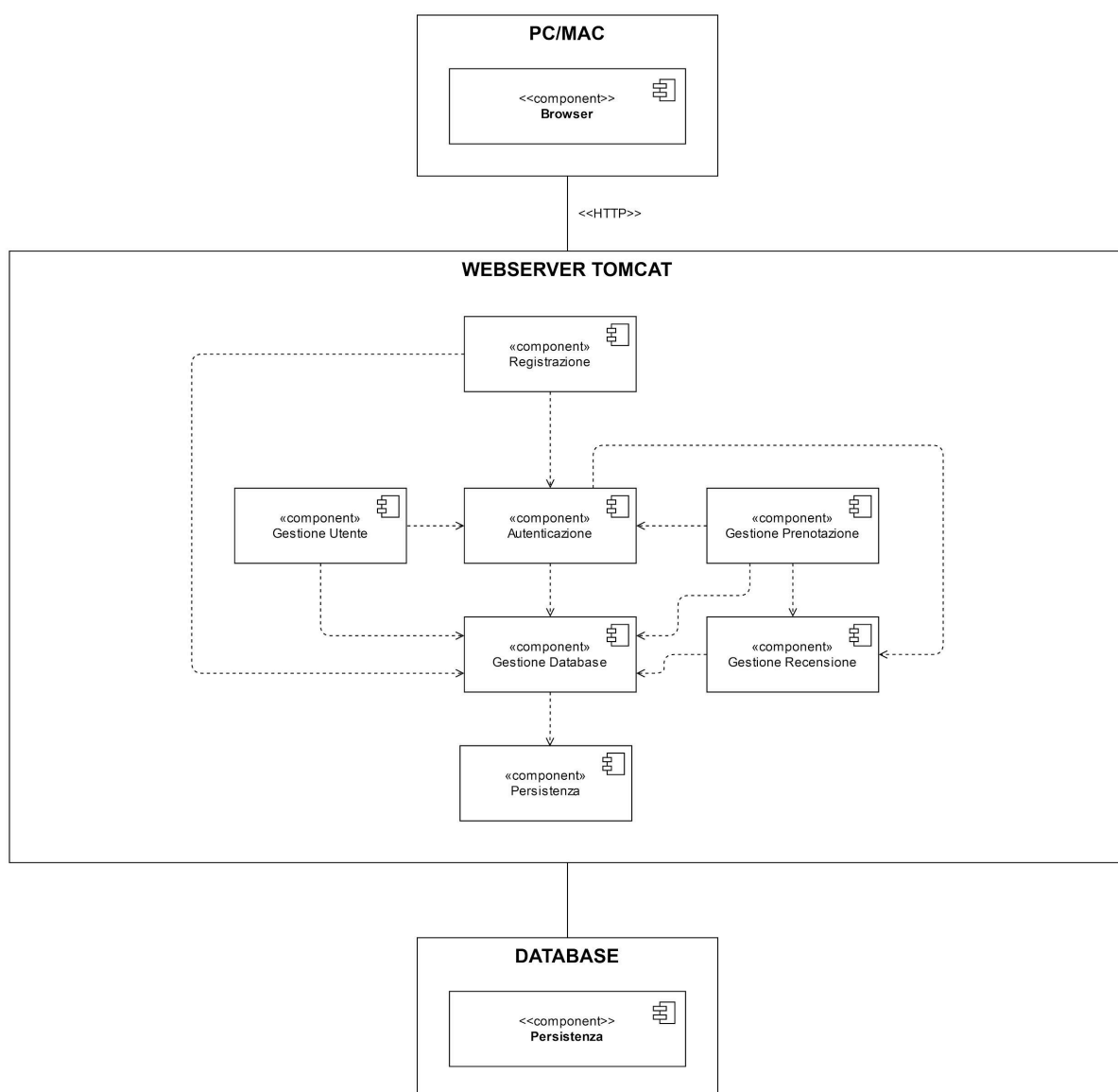


3.3 Mapping Hardware/Software

L'applicazione web in fase di sviluppo sarà ospitata su una piattaforma hardware composta da un server centrale, il quale gestisce le richieste provenienti dai client attraverso un browser.

3.3.1 Deployment Diagram

Di seguito un UML deployment diagram che descrive il mapping hardware/software.



3.4 Gestione dati persistenti

Gli oggetti Entity individuati durante la fase di analisi, specificati nel RAD, sono responsabili della gestione dei dati che devono essere persistenti. Questi dati non solo devono sopravvivere all'esecuzione del sistema, ma anche a eventuali arresti, sia controllati che imprevisti, garantendo la loro disponibilità nel tempo. In MedConnect sono stati identificati i seguenti oggetti persistenti:

- DisponibilitàEntity
- MedicoEntity
- PazienteEntity
- PrenotazioneEntity
- RecensioneEntity
- UtenteRegistratoEntity
- CartellaClinicaEntity
- AmministratoreEntity

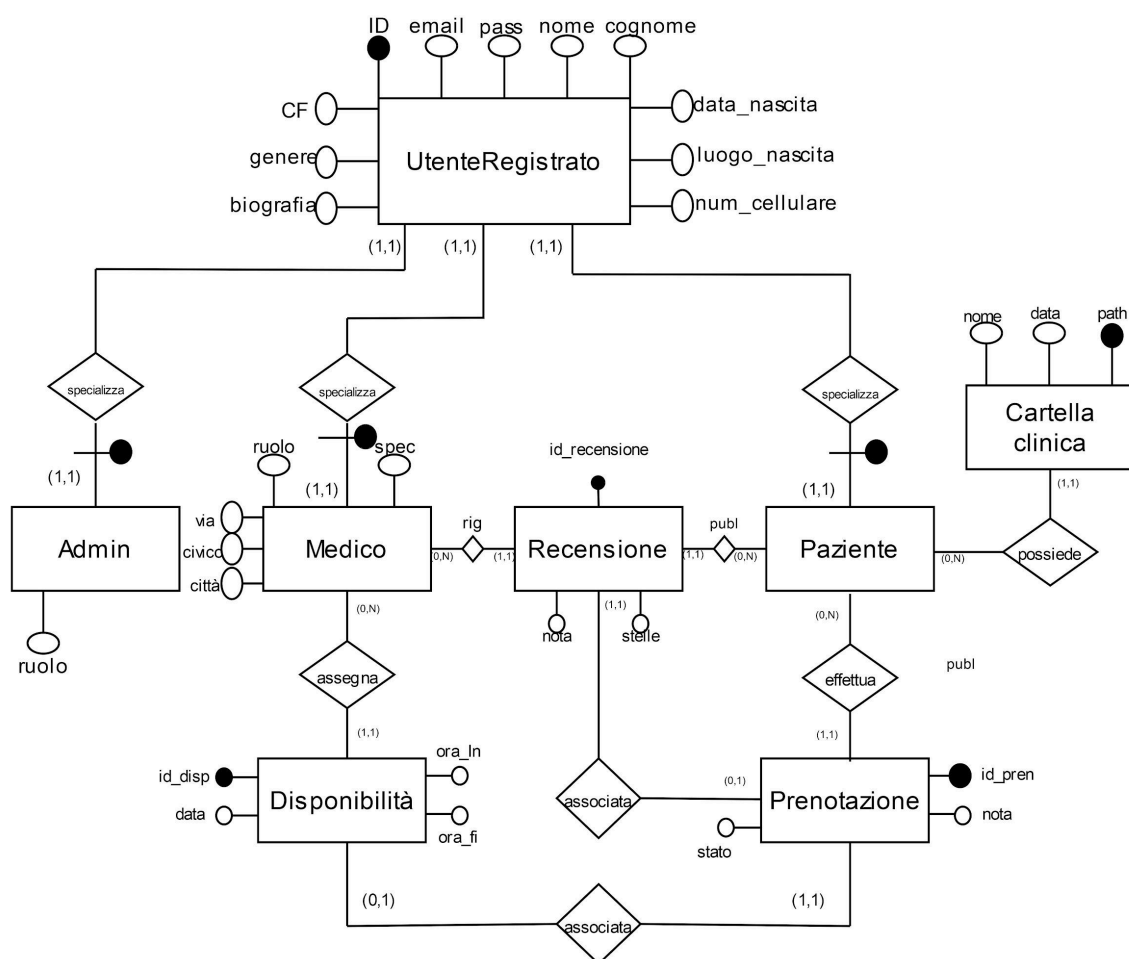
Gli ultimi due oggetti Entity dell'elenco sono stati inseriti per completezza ma non sono stati "ricavati" dagli use case specificati.

La scelta della tecnologia per l'archiviazione di questi oggetti è stata guidata dai requisiti non funzionali e, in particolare, dai design goals precedentemente discussi. Per soddisfare tali requisiti, MedConnect utilizza un database relazionale, che offre i seguenti vantaggi :

- Tempi di risposta rapidi, per un accesso immediato ai dati.
- Facilità di accesso e ampia capacità di archiviazione.
- Accesso concorrente ai dati, garantendo affidabilità e integrità anche in caso di utilizzo simultaneo da parte di più utenti.
- Sicurezza dei dati, con meccanismi di accesso protetto e la possibilità di ripristinare il sistema in caso di guasti o danni.

Utilizzando un database relazionale, i dati nel nostro contesto sono gestiti in modo intuitivo e diretto attraverso tabelle, permettendo una rapida gestione di grandi quantità di dati. Per garantire che i dati siano sempre accurati e accessibili, i database relazionali seguono regole di integrità specifiche. L'integrità dei dati è ulteriormente assicurata da backup

periodici, che in caso di perdita di dati fungono da supporto per il recupero delle informazioni. Inoltre, i dati memorizzati nel database sono protetti tramite un sistema di accesso controllato, con il quale solo un amministratore autorizzato può apportare modifiche, prevenendo accessi non autorizzati e garantendo la sicurezza delle informazioni. Questo approccio assicura che MedConnect possa offrire un servizio efficiente, sicuro e scalabile nel tempo.



SCHEMA LOGICO

UtenteRegistrato (**ID**, email, pass, nome, cognome, cf, genere, biografia, data_nascita, luogo_nascita, num_cellulare)

Admin (**ID_admin**, ruolo) ID_admin **VIR** con id.UtenteRegistrato

Medico (**ID_medico**, ruolo, specializzazione, via, civico, città) ID_medico **VIR** con id.UtenteRegistrato

Paziente (**ID_paziente**) ID_paziente **VIR** con id.UtenteRegistrato

Recensione (**ID_medico**, **ID_paziente**, **ID_recensione**, nota, stelle, ID_prenotazione) ID_prenotazione **VIR** id_pren

Prenotazione (**ID_prenotazione**, stato, nota, **ID_paziente**, **ID_disponibilita**) ID_paz **VIR** + ID_disp **VIR**

Disponibilita (**ID_disponibilita**, data, ora_in, ora_fi, **ID_medico**) ID_medico **VIR**

Cartelle (**Path**, nome, data, **ID_paz**) ID_paz **VIR** ID_paziente

3.4.1 Dizionario dei dati

Di seguito si mostrano gli attributi per ogni entità individuata.

Nome Entità	UtenteRegistrato		
Descrizione	Contiene i dati relativi ad un utente registrato.		
Nome Campo	TIPO	VINCOLO DI CHIAVE	ALTRI VINCOLI
ID	INT	PRIMARY KEY	
email	VARCHAR(255)		NOT NULL UNIQUE
pass	VARCHAR(255)		NOT NULL
nome	VARCHAR(100)		NOT NULL
cognome	VARCHAR(100)		NOT NULL
cf	VARCHAR(16)		NOT NULL UNIQUE
genere	VARCHAR(10)		
biografia	VARCHAR(255)		
data_nascita	DATE		
luogo_nascita	VARCHAR(50)		

num_cellulare	VARCHAR(10)		
---------------	-------------	--	--

Nome Entità	Admin		
Descrizione	Contiene i dati relativi ad un admin.		
Nome Campo	TIPO	VINCOLO DI CHIAVE	ALTRI VINCOLI
ID_admin	INT	PRIMARY KEY FOREIGN KEY	
Ruolo	VARCHAR(50)		

Nome Entità	Medico		
Descrizione	Contiene i dati relativi ad un medico.		
Nome Campo	TIPO	VINCOLO DI CHIAVE	ALTRI VINCOLI
ID_medico	INT	PRIMARY KEY FOREIGN KEY	
Ruolo	VARCHAR(50)		
Specializzazione	VARCHAR(100)		
via	VARCHAR(100)		
civico	VARCHAR(100)		
citta	VARCHAR(100)		

Nome Entità	Paziente		
Descrizione	Contiene i dati relativi ad un paziente.		
Nome Campo	TIPO	VINCOLO DI CHIAVE	ALTRI VINCOLI
ID_paziente	INT	PRIMARY KEY FOREIGN KEY	

Nome Entità	Disponibilità		
Descrizione	Contiene i dati relativi alle disponibilità di un medico.		
Nome Campo	TIPO	VINCOLO DI CHIAVE	ALTRI VINCOLI
ID_disponibilita	INT	PRIMARY KEY	
data	DATE		NOT NULL
ora_in	TIME		NOT NULL
ora_fi	TIME		NOT NULL
ID_medico	INT	FOREIGN KEY	NOT NULL

Nome Entità	Prenotazione		
Descrizione	Contiene i dati relativi ad una prenotazione.		
Nome Campo	TIPO	VINCOLO DI CHIAVE	ALTRI VINCOLI
ID_prenotazione	INT	PRIMARY KEY	
stato	VARCHAR(50)		
nota	VARCHAR(255)		
ID_paziente	INT	FOREIGN KEY	NOT NULL
ID_disponibilita	INT	FOREIGN KEY	NOT NULL

Nome Entità	Recensione		
Descrizione	Contiene i dati relativi ad una recensione.		
Nome Campo	TIPO	VINCOLO DI CHIAVE	ALTRI VINCOLI
ID_recensione	INT	PRIMARY KEY	
ID_prenotazione	INT	FOREIGN KEY	NOT NULL
ID_medico	INT	FOREIGN KEY	NOT NULL

ID_paziente	INT	FOREIGN KEY	NOT NULL
nota	VARCHAR(255)		
stelle	INT		CHECK (stelle BETWEEN 1 AND 5)

Nome Entità	Cartelle		
Descrizione	Contiene i dati relativi ad una cartella clinica di un paziente.		
Nome Campo	TIPO	VINCOLO DI CHIAVE	ALTRI VINCOLI
Path	VARCHAR(255)	PRIMARY KEY	
nome	VARCHAR(100)		NOT NULL
data	DATE		NOT NULL
ID_paz	INT	FOREIGN KEY	NOT NULL

3.5 Controllo degli accessi e sicurezza

Di seguito viene mostrata la matrice degli accessi per poter tenere traccia di quali attori possono accedere ai quali dei servizi offerti dal sistema.

Attore	Paziente	Medico	Utente Registrato
/			
Oggetti			
Registrazione			
Autenticazione			Login Logout
Gestione Utente	InserimentoCartelleCliniche ModificaCartelleCliniche EliminazioneCartelleCliniche RicercaMedico VisualizzazioneMedico	InserimentoSpecializzazioni ModificaSpecializzazioni EliminazioneSpecializzazioni VisualizzazionePaziente	InserimentoInfoBase ModificaInfoBase RiepilogoPrenotazioni EliminazioneInfoBase
Gestione Prenotazione	EffettuaPrenotazione ModificaPrenotazione EliminaPrenotazione	DisdettaPrenotazione	VisualizzazionePrenotazione

Gestione Recensione	InserimentoRecensione ModificaRecensione EliminaRecensione	RecensioneInPrimo Piano	VisualizzazioneRecensione SegnalaRecensione
----------------------------	--	-------------------------	--

Attore / Oggetti	Amministratore	Ospite
Registrazione		RegistrazioneMedico RegistrazionePaziente
Autenticazione	Login Logout	
Gestione Utente	InserimentoInfoBase ModificaInfoBase EliminazioneInfoBase	
Gestione Prenotazione	VisualizzazionePrenotazione	
Gestione Recensione	VisualizzazioneRecensione SegnalaRecensione	

3.6 Controllo globale del software

MedConnect è un sistema interattivo progettato per avviare ogni funzionalità in risposta alle azioni dell'utente tramite un'interfaccia grafica. Gli utenti possono accedere e utilizzare le funzionalità del sistema attraverso l'interfaccia, che attiva il controllo appropriato per gestire l'operazione richiesta.

Ogni azione genera un evento, che viene elaborato da un componente dedicato. Questo componente instrada il flusso verso il sottosistema responsabile della gestione della logica di controllo, che a sua volta comunica con i servizi della logica applicativa per completare l'operazione.

Grazie a questo approccio, MedConnect garantisce un funzionamento efficiente e reattivo, adottando un modello di controllo basato sugli eventi (**event-driven**), ideale per una piattaforma web.

3.7 Condizioni limite

Questa sezione elenca le condizioni limite del sistema, che sono:

- Avvio del sistema,
- Spegnimento del sistema,
- Fallimento del sistema,
- Errore di accesso ai Dati Persistenti.

I dettagli completi di queste condizioni limite descritte tramite casi d'uso, flussi di eventi principali e alternativi, sono documentati nella sezione *Boundary Conditions* del documento **C05_RAD_ver.2.0**, poiché riguardano la definizione dei requisiti. Nel presente documento, ci si concentra sulle implicazioni di progettazione tecnica che derivano da queste condizioni.

4 Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema che in precedenza sono stati elencati.

4.1 Registrazione

Servizio	Descrizione	Interfaccia
Registrazione Paziente	Questa funzionalità del sottosistema permette ai Pazienti di registrarsi all'interno del sistema.	RegistrazioneService
Registrazione Medico	Questa funzionalità del sottosistema permette ai Pazienti di registrarsi all'interno del sistema.	RegistrazioneService

4.2 Autenticazione

Servizio	Descrizione	Interfaccia
Login	Questa funzionalità del sottosistema permette agli utenti registrati di avere accesso al sistema.	AutenticazioneService
Logout	Questa funzionalità del sottosistema permette agli utenti registrati di disconnettersi dal sistema.	AutenticazioneService

4.3 Gestione Utente

Servizio	Descrizione	Interfaccia
Inserimento Cartelle Cliniche	Questa funzionalità del sottosistema permette ai pazienti di inserire la propria cartella clinica.	GestioneUtenteService
Inserimento Specializzazioni	Questa funzionalità del sottosistema permette ai medici di inserire le proprie specializzazioni.	GestioneUtenteService
Inserimento Info Base	Questa funzionalità del sottosistema permette agli utenti registrati di inserire le proprie informazioni di base.	GestioneUtenteService
Modifica Cartelle Cliniche	Questa funzionalità del sottosistema permette ai pazienti di modificare la propria cartella clinica.	GestioneUtenteService
Modifica Specializzazioni	Questa funzionalità del sottosistema permette ai medici di modificare le proprie specializzazioni.	GestioneUtenteService
Modica Info Base	Questa funzionalità del sottosistema permette agli utenti registrati di modificare le proprie informazioni di base.	GestioneUtenteService
Eliminazione Cartelle Cliniche	Questa funzionalità del sottosistema permette ai	GestioneUtenteService

	pazienti di eliminare la propria cartella clinica.	
Eliminazione Specializzazioni	Questa funzionalità del sottosistema permette ai medici di eliminare le proprie specializzazioni.	GestioneUtenteService
Eliminazione Info Base	Questa funzionalità del sottosistema permette agli utenti registrati di eliminare le proprie informazioni di base.	GestioneUtenteService
Riepilogo Prenotazioni	Questa funzionalità del sottosistema permette agli utenti registrati di visualizzare le proprie prenotazioni.	GestioneUtenteService
Ricerca Medico	Questa funzionalità del sottosistema permette ai pazienti di ricercare tramite alcuni parametri un medico.	GestioneUtenteService
Visualizzazione Medico	Questa funzionalità del sottosistema permette ai pazienti di visualizzare il profilo di un medico.	GestioneUtenteService
Visualizzazione Paziente	Questa funzionalità del sottosistema permette ai medici di visualizzare il profilo di un paziente che ha effettuato una prenotazione con quest'ultimo.	GestioneUtenteService

4.4 Gestione Prenotazione

Servizio	Descrizione	Interfaccia
Effettua Prenotazione	Questa funzionalità del sottosistema permette ai pazienti di effettuare una prenotazione per una visita medica.	GestionePrenotazioneService
Modifica Prenotazione	Questa funzionalità del sottosistema permette ai pazienti di modificare una prenotazione per una visita medica.	GestionePrenotazioneService
Elimina Prenotazione	Questa funzionalità del sottosistema permette ai pazienti di eliminare una prenotazione per una visita medica.	GestionePrenotazioneService
Disdetta Prenotazione	Questa funzionalità del sottosistema permette ai pazienti di disdire una prenotazione di un paziente.	GestionePrenotazioneService
Visualizzazione Prenotazione	Questa funzionalità del sottosistema permette agli utenti registrati di visualizzare una prenotazione.	GestionePrenotazioneService

4.5 Gestione Recensione

Servizio	Descrizione	Interfaccia
Inserimento Recensione	Questa funzionalità del sottosistema permette ai pazienti di inserire una recensione per un medico con il quale hanno effettuato una visita.	GestioneRecensioneService
Modifica Recensione	Questa funzionalità del sottosistema permette ai pazienti di modificare una recensione scritta in precedenza.	GestioneRecensioneService
Elimina Recensione	Questa funzionalità del sottosistema permette ai pazienti di eliminare una recensione scritta in precedenza.	GestioneRecensioneService
Recensione In Primo Piano	Questa funzionalità del sottosistema permette ai medici di inserire una recensione in primo piano tra le proprie recensioni.	GestioneRecensioneService
Visualizzazione Recensione	Questa funzionalità del sottosistema permette agli utenti registrati di visualizzare una recensione.	GestioneRecensioneService

Segnala Recensione	Questa funzionalità del sottosistema permette agli utenti registrati di segnalare una recensione.	GestioneRecensioneServizio
--------------------	---	----------------------------

5 Glossario

Termine/Sigla	Definizione
Deployment diagram	In UML rappresenta i componenti in fase di esecuzione e le loro assegnazioni ai nodi hardware. I componenti sono entità autonome che forniscono servizi ad altri componenti o attori. Un nodo è un dispositivo fisico o un ambiente di esecuzione in cui vengono eseguiti i componenti.
Component Diagram	Un diagramma dei componenti rappresenta i componenti di un sistema. Un componente è un modulo autonomo con interfacce ben definite che specifica un'incapsulazione di comportamento e stato. Un componente può essere sostituito da un altro componente con lo stesso comportamento e le stesse interfacce.
Boundary conditions	Le condizioni limite sono situazioni speciali che il sistema deve gestire; includono l'avvio, l'arresto e le eccezioni. I casi d'uso che trattano queste condizioni sono chiamati boundary use cases e sono specificati nel RAD.
Design goals	Gli obiettivi di progettazione, sono le qualità che gli sviluppatori dovrebbero ottimizzare durante la progettazione di un sistema software. I design goal sono spesso derivati dai requisiti non funzionali e sono utilizzati per guidare le decisioni di progettazione.
Event-driven control	"Event-Driven" si riferisce a un sistema in cui il flusso del programma è determinato da eventi. Questi eventi possono essere azioni intraprese da un utente, come un clic del mouse o un input da tastiera, o possono essere segnali provenienti da altri programmi o dispositivi.

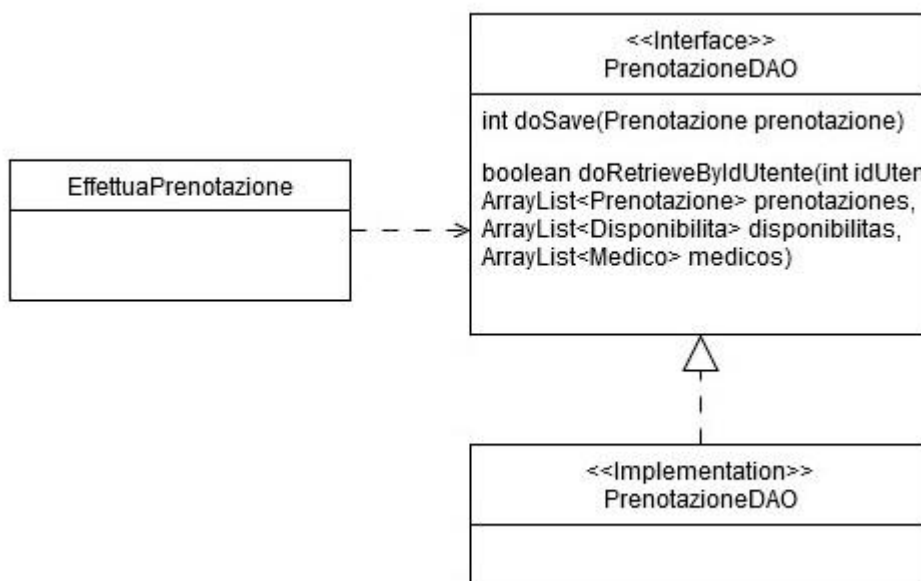
6 Design Pattern

Di seguito, vengono descritti i design pattern utilizzati nello sviluppo del sistema.

6.1 Facade Design Pattern

MedConnect è un sistema caratterizzato da un'elevata complessità, motivo per cui adotta il design pattern Facade per organizzare e semplificare l'accesso alla sua logica di business. In particolare, MedConnect utilizza il pattern Facade per ciascuno dei suoi sottosistemi, implementando interfacce che forniscono accesso ai metodi richiesti.

Di seguito un esempio di Facade utilizzato nel sistema MedConnect:



In questo esempio il sottosistema implementato <<Implementation>> PrenotazioneDAO avrà una propria interfaccia pubblica <<Interface>> PrenotazioneDAO, che esporrà tutte le funzionalità fornite e realizzate dal componente <<Implementation>> PrenotazioneDAO. Tale interfaccia sarà poi implementata da <<Implementation>> PrenotazioneDAO. Questa classe conterrà tutta la logica implementativa dei vari servizi offerti, per cui il chiamante avrà solamente bisogno di istanziarla ed utilizzare il metodo desiderato, senza dover conoscere come è stato implementato

effettivamente.

In generale le interfacce di alto livello espongono i vari servizi offerti da un sottosistema, rendendolo più facile da utilizzare e nascondendo dettagli implementativi o dipendenze a chi deve utilizzarlo. Le interfacce si interpongono quindi tra i vari componenti di un sistema, facilitando la comunicazione e diminuendo l'accoppiamento. Inoltre, ciò apporta vantaggi nell'effettuare l'Integration Testing.

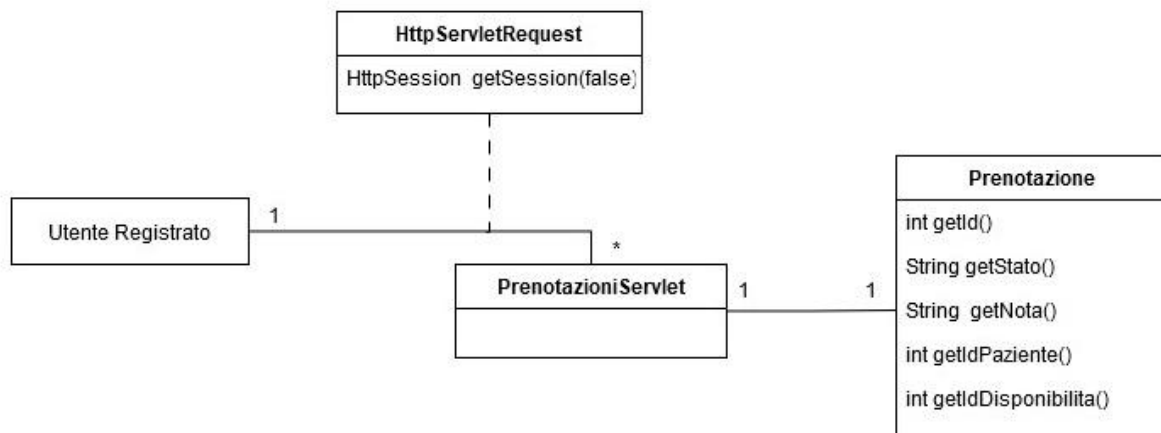
6.2 Proxy Design Pattern

MedConnect per migliorare le prestazioni e la sicurezza del sistema utilizza il Proxy Design Pattern che consente di ritardare i costi dei calcoli e della memoria solo quando necessari o controllando l'accesso prima del caricamento di un oggetto nella memoria.

MedConnect utilizza il proxy design pattern per i seguenti aspetti:

- Proxy di Protezione viene utilizzata per garantire che solo utenti con permessi legittimi possano accedere a dati sensibili, come la sezione le mie prenotazioni. Questo proxy funge da intermediario tra il richiedente e l'oggetto effettivo (es. una prenotazione). I permessi di accesso vengono verificati prima di caricare o eseguire qualsiasi operazione.
- Proxy di Archiviazione viene utilizzata per caricare su richiesta dati pesanti, come le cartelle cliniche, solo quando sono realmente necessari. Questo approccio riduce l'uso della memoria e migliora le performance.

Di seguito un esempio di Proxy di Protezione utilizzato nel sistema MedConnect:



In questo esempio la classe di associazione `HttpServletRequest` contiene un metodo che deve essere utilizzato prima di poter accedere alle Prenotazioni.

Ogni operazione in `Prenotazione` prima di essere richiamata, verifica tramite il proxy `PrenotazioniServlet`, che utilizza il metodo `getSession(false)` della classe `HttpServletRequest`, che l'Utente Registrato che ha effettuato la richiesta possieda i permessi necessari. In caso di accesso autorizzato, `PrenotazioneProxy` delega l'operazione all'oggetto `Prenotazione` effettivo. Se l'accesso viene negato, l'oggetto `Prenotazione` non viene caricato e l'operazione non viene eseguita.