



**MedConnect**

MedConnect - C05

# Progetto Ingegneria del software

Laurea Triennale in Informatica - Università di Salerno

**Cusati Daniel**

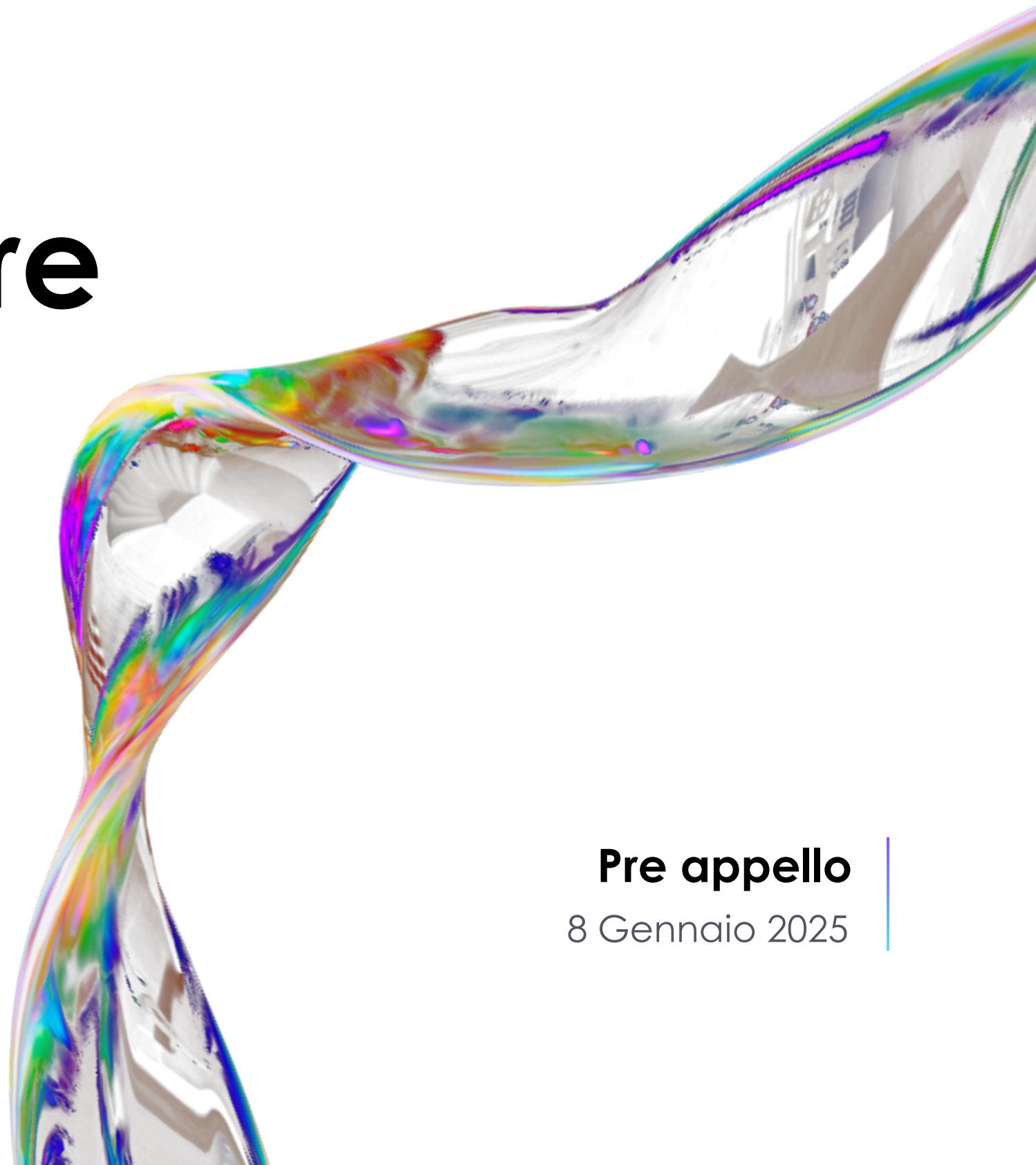
(Mat. 05121 12422)

**Medica Vincenzo**

(Mat. 05121 16808)

**Varone Giuseppe**

(Mat. 05121 17831)



**Pre appello**

8 Gennaio 2025

# Outline

Introduzione

---

RAD

---

SDD

---

Testing

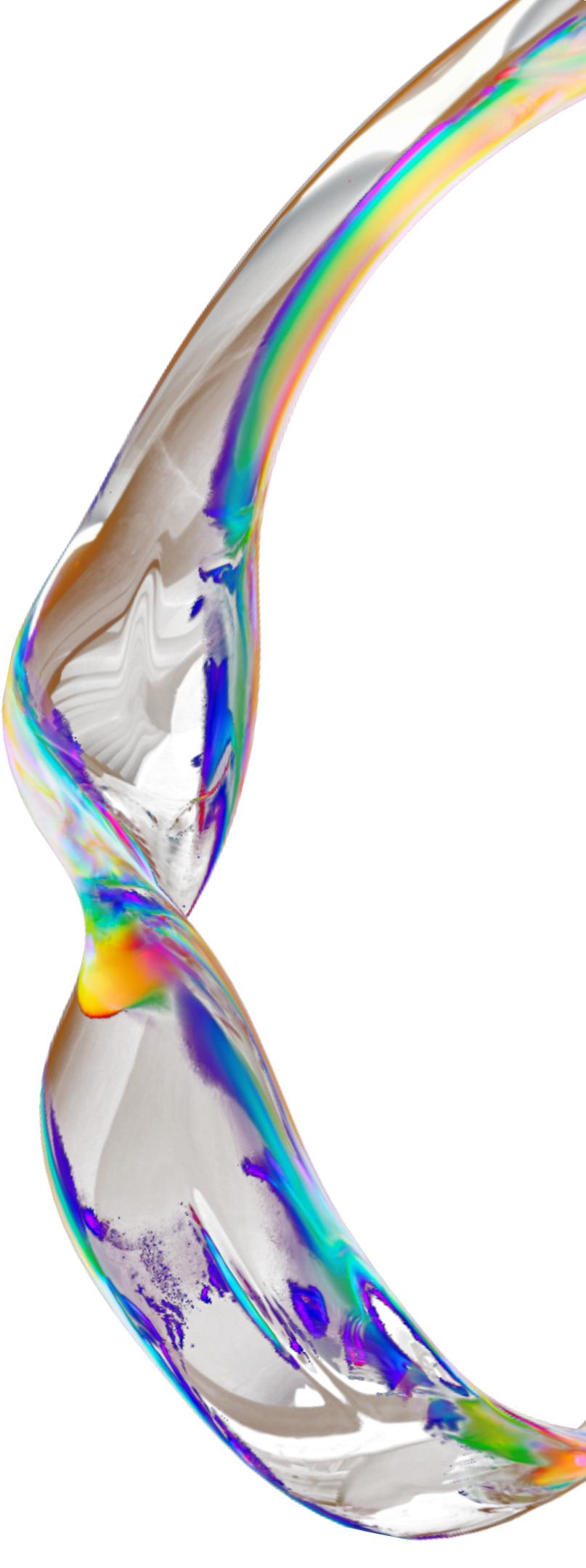
---

Lezioni imparate

---

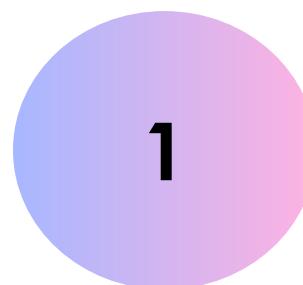
Domande

---



# Introduzione

MedConnect è una web app innovativa, commissionata dal Ministero della Salute, progettata per rivoluzionare la gestione delle prenotazioni mediche. La piattaforma rappresenta un ponte diretto tra pazienti e medici e mira a semplificare l'accesso ai servizi sanitari, ridurre i tempi di attesa e promuovere un mercato sanitario più trasparente e professionale.



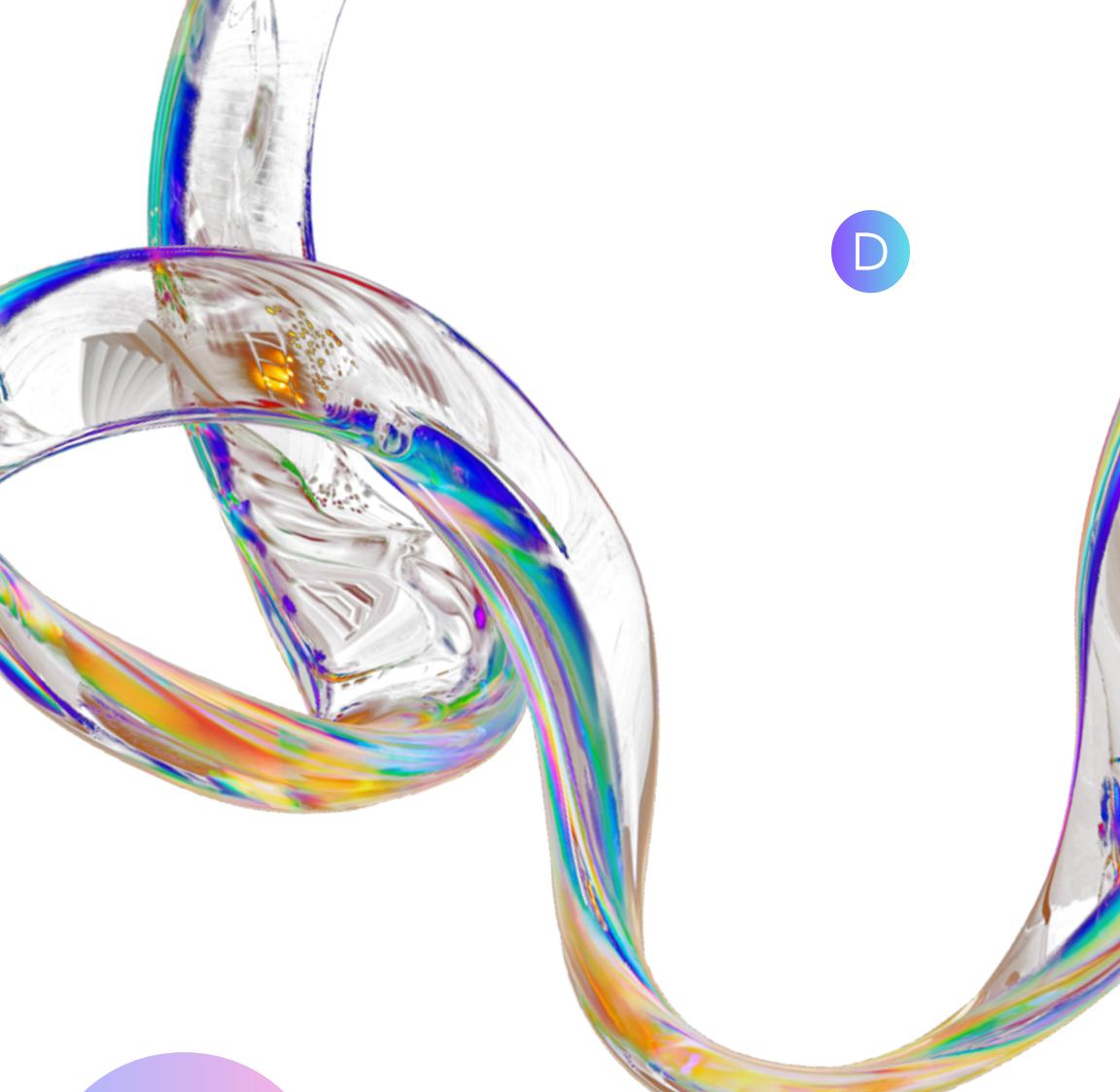
## Prenotazioni

gestione delle prenotazioni in modo semplice e intuitivo.



## Recensioni

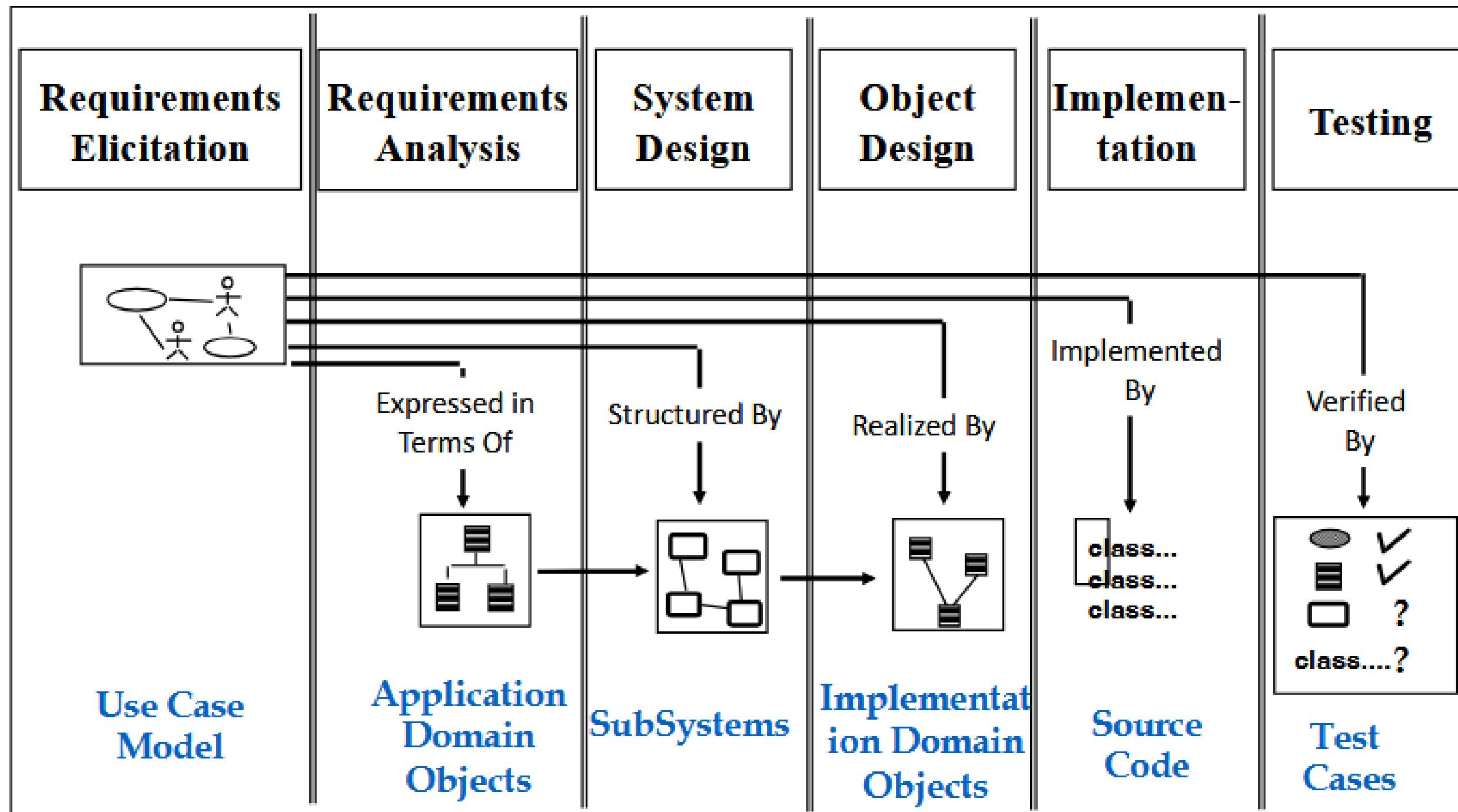
I pazienti possono lasciare una recensione ai medici che li hanno visitati.



## Profilo personale

Gli utenti possono condividere informazioni personali per arricchire il profilo.

# flusso del lavoro



D

# RAD

## Requirements Analysis Document

Sistema corrente

---

Attori

---

## Sistema proposto

---



Requisiti funzionali



Requisiti non funzionali



Vincoli

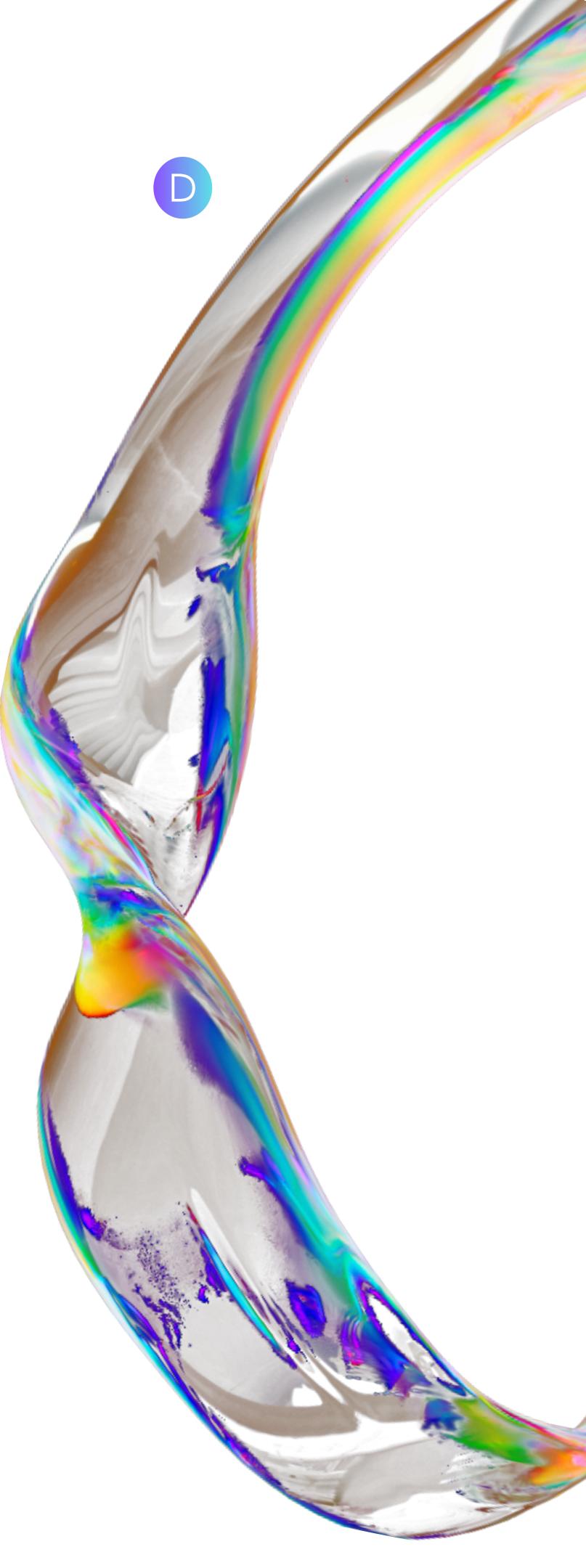


Modelli di sistema

Interfaccia Utente : mock-ups e navigational path

---

D

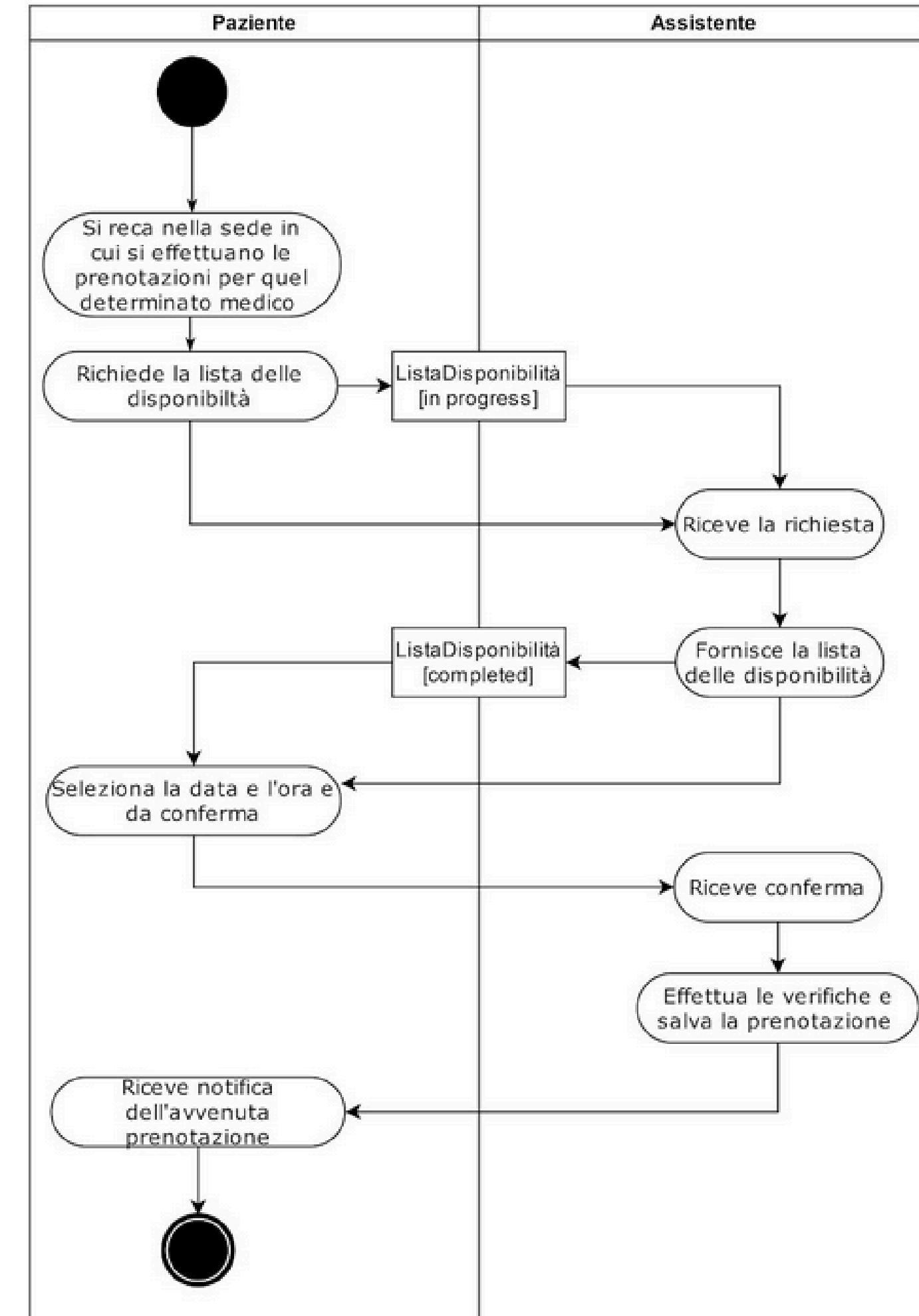


# Sistema corrente

Nel sistema sanitario tradizionale, la gestione delle prenotazioni mediche si basava su processi manuali e non centralizzati.

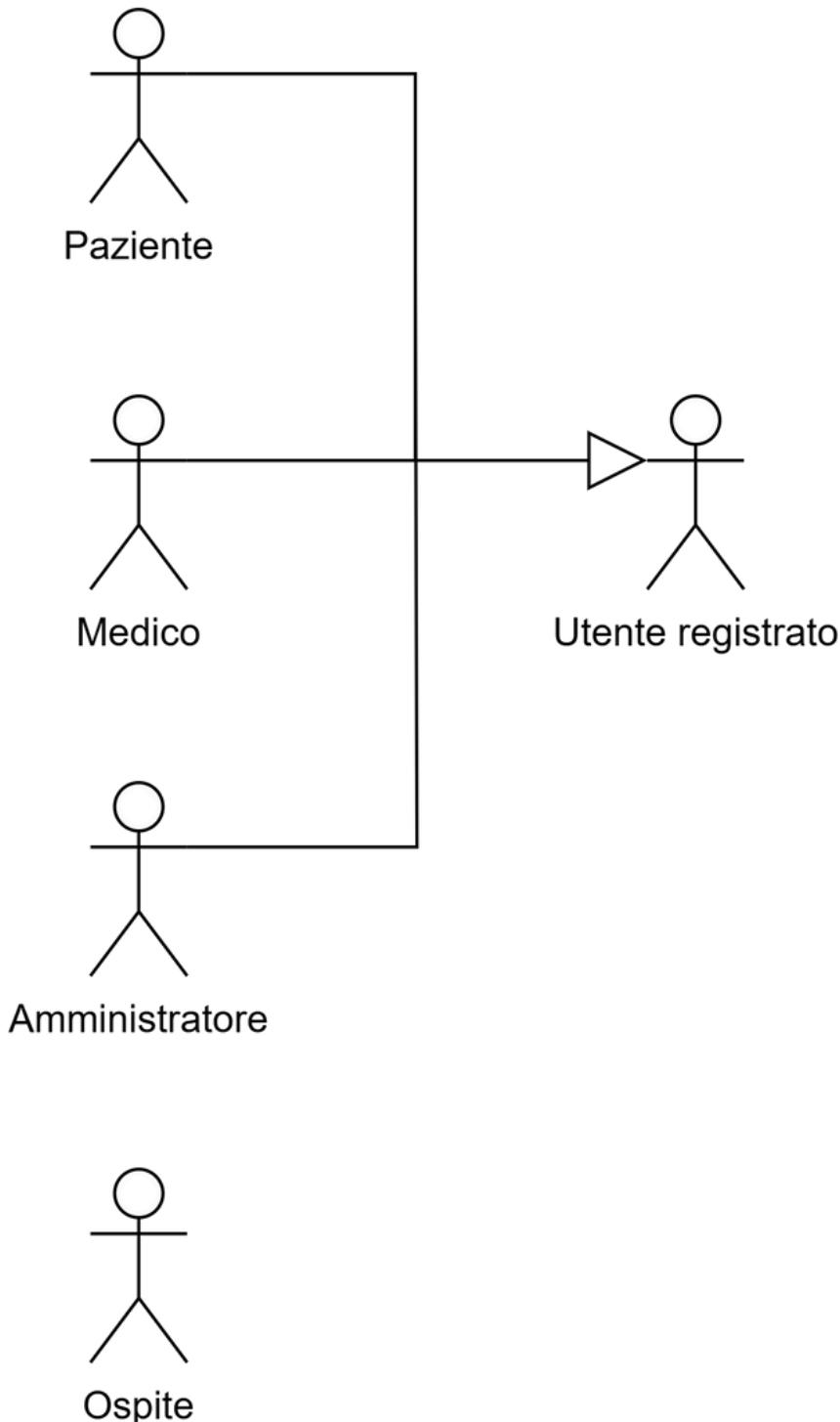
I pazienti erano spesso costretti a recarsi di persona presso ambulatori o a contattare telefonicamente strutture sanitarie, spesso sovraccaricate, con limitate opzioni nella scelta dei medici disponibili.

Con l'introduzione di MedConnect, il sistema sanitario è stato digitalizzato e centralizzato. Questo ha eliminato la necessità di vincoli fisici o temporali, consentendo a medici e pazienti di gestire appuntamenti in modo più efficiente e flessibile.



D

# Attori



## Paziente :

Persona affetta da una malattia, e più genericamente chi è affidato alle cure di un medico o di un chirurgo. Nel contesto di sviluppo di questa web app questo termine non indica necessariamente una persona affetta da una malattia.

## Medico :

Persona che professa la medicina, e dunque si occupa di studiare, curare e prevenire le malattie delle persone.

## Amministratore :

Persona responsabile della gestione del sistema in vari aspetti, a cui è assegnato un ruolo specifico.

## Ospite :

Persona che non è registrata alla piattaforma e può accedere ad operazioni di visualizzazione.

# Requisiti funzionali

I requisiti funzionali individuati e definiti del sistema sono organizzati in categorie chiamate 'gestioni', ciascuna delle quali raggruppa funzionalità correlate per una migliore comprensione e organizzazione. Ad ogni requisito è stato associato un ID, una descrizione ed una priorità.

Gestione Prenotazione



Gestione Utente



Gestione Recensione



0

5

10

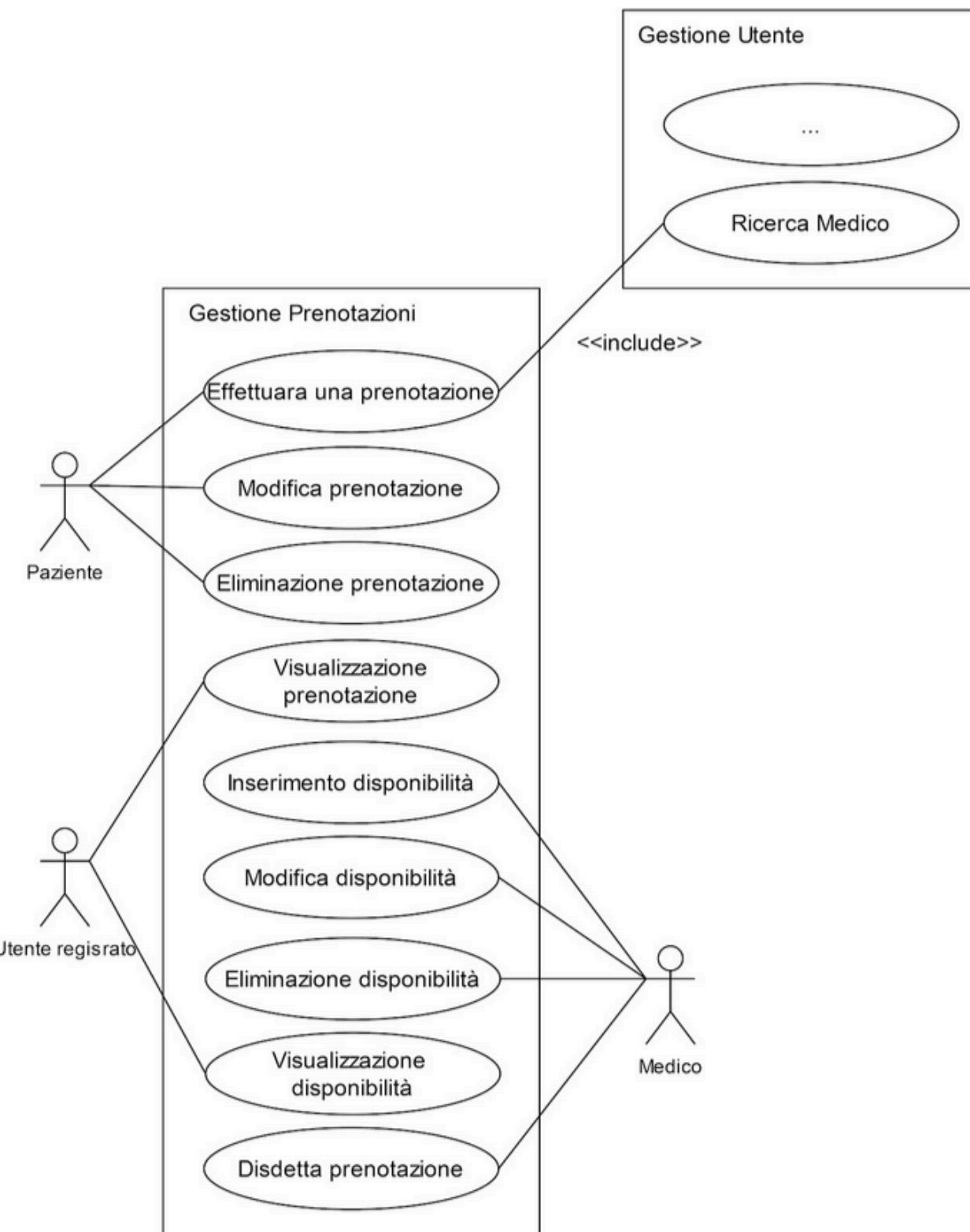
15

20

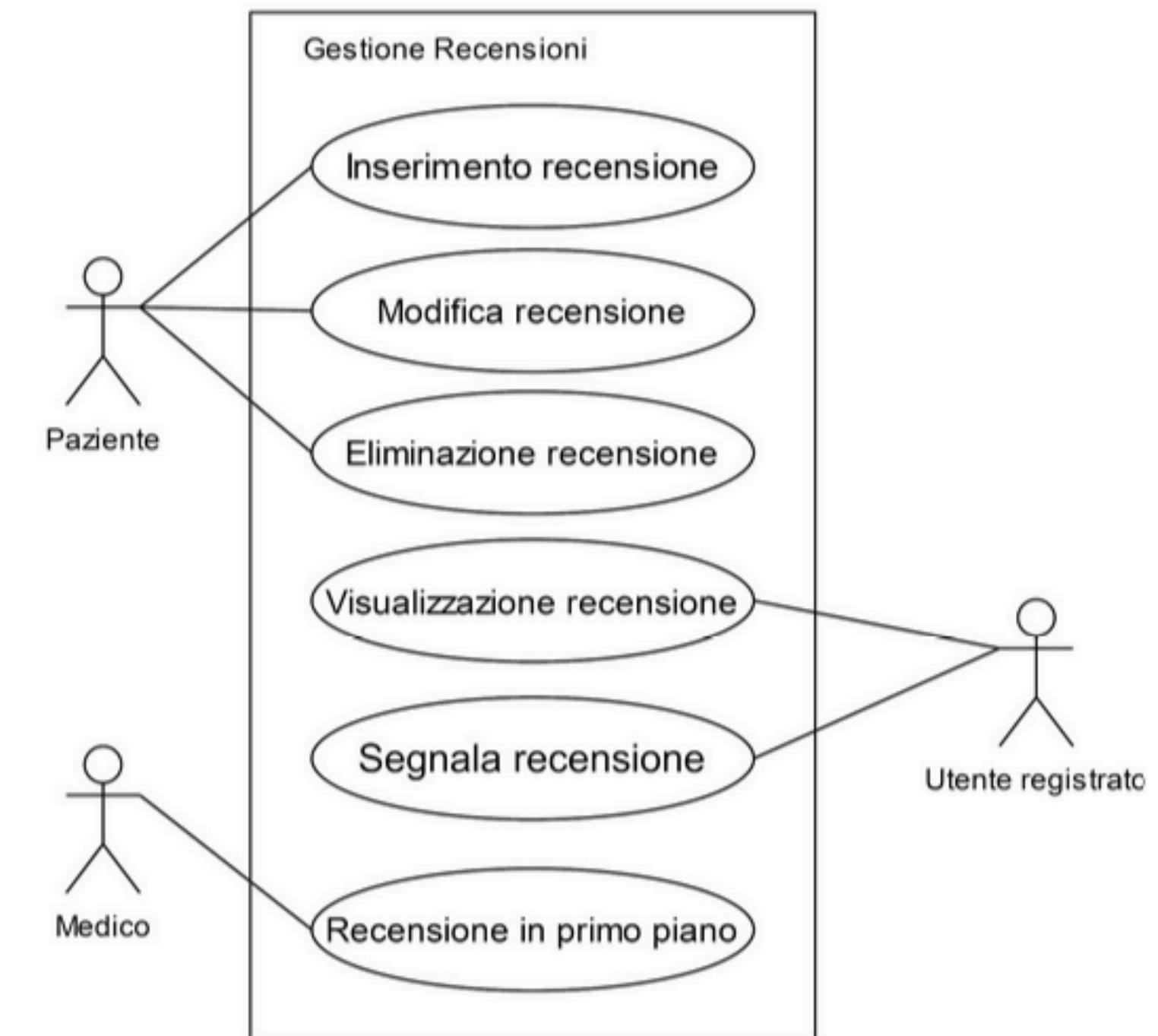
D

# Use case diagram

## GESTIONE PRENOTAZIONI

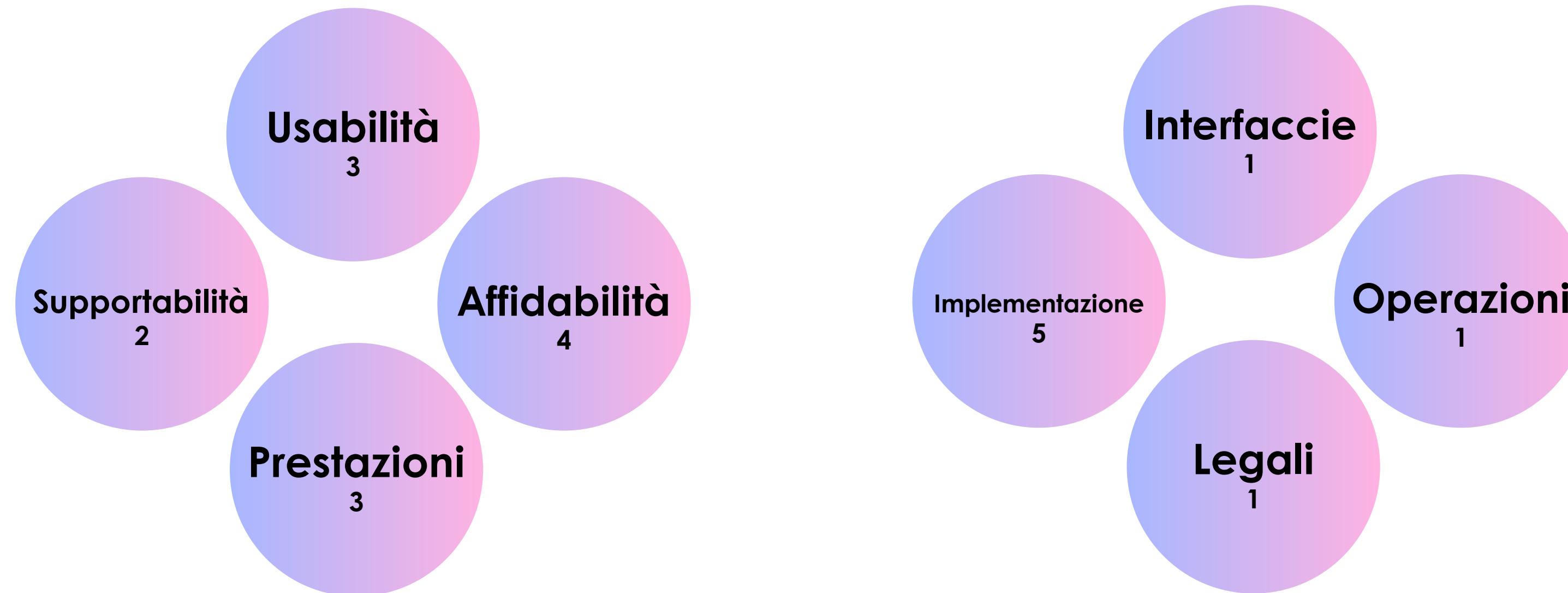


## GESTIONE RECENSIONI



# Requisiti non funzionali

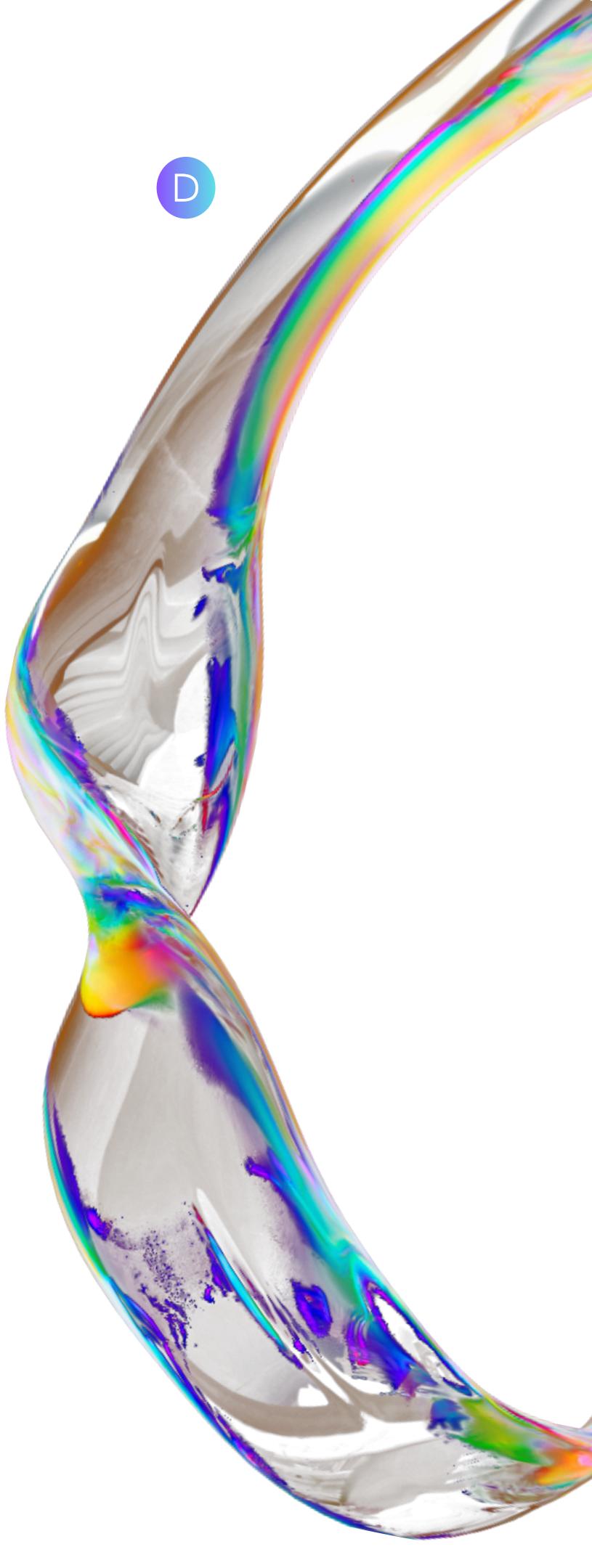
I requisiti non funzionali individuati sono stati suddivisi in categorie secondo il modello FURPS+, che include: usabilità, affidabilità, prestazioni, supportabilità e altri vincoli. Ad ogni requisito è stato associato un ID, una descrizione ed una priorità.



# Modelli di sistema sviluppati

- Quattro scenari per membro del team.
- Uno use case per membro del team.
- Uno use case diagram per membro del team.
- Un class diagram.
- Due sequence diagram.
- Due statechart diagram.

D



# Scenari

Di seguito trovate l'elenco dei quattro scenari selezionati dalla gestione **prenotazione** e sviluppati da **Varone Giuseppe**. Lo scenario in grassetto è stato sviluppato come use case.



## **Effettuare una prenotazione**

Il sistema dovrà permettere al paziente di effettuare una prenotazione medica in base ai medici disponibili.



## Visualizzazione prenotazione

Il sistema dovrà permettere all'utente registrato di visualizzare le sue prenotazioni.



## Modifica Prenotazione

Il sistema dovrà permettere al paziente di modificare le informazioni della prenotazione medica.



## Eliminazione Prenotazione

Il sistema dovrà permettere al paziente di eliminare una prenotazione.

# Scenari

Di seguito trovate l'elenco dei quattro scenari selezionati dalla gestione **utente** e sviluppati da **Cusati Daniel**. Lo scenario in grassetto è stato sviluppato come use case.



## Inserimento informazioni di base

Il sistema dovrà permettere all'utente di inserire le sue informazioni di base.



## Inserimento cartella clinica

Il sistema dovrà permettere al paziente di inserire le sue cartelle cliniche.



## Inserimento specializzazione

Il sistema dovrà permettere al medico di inserire le sue competenze e specializzazioni.



## Eliminazione specializzazione

Il sistema dovrà permettere al medico di eliminare le sue specializzazioni.

# Scenari

Di seguito trovate l'elenco dei quattro scenari selezionati dalla gestione **recensione** e sviluppati da **Medica Vincenzo**. Lo scenario in grassetto è stato sviluppato come use case.



## Inserimento recensione

Il sistema dovrà consentire ai pazienti di scrivere una recensione a visita completata.



## Modifica recensione

Il sistema dovrà consentire ai pazienti di modificare le recensioni precedentemente scritte.



## Eliminazione recensione

Il sistema dovrà consentire ai pazienti di eliminare una recensione precedentemente inserita.



## Visualizzazione recensione

Il sistema dovrà consentire ad un utente registrato di visualizzare una recensione fatta ad un medico.

# Use case sviluppati

- Effettuare una prenotazione.
- Inserimento informazioni di base.
- Inserimento recensione.

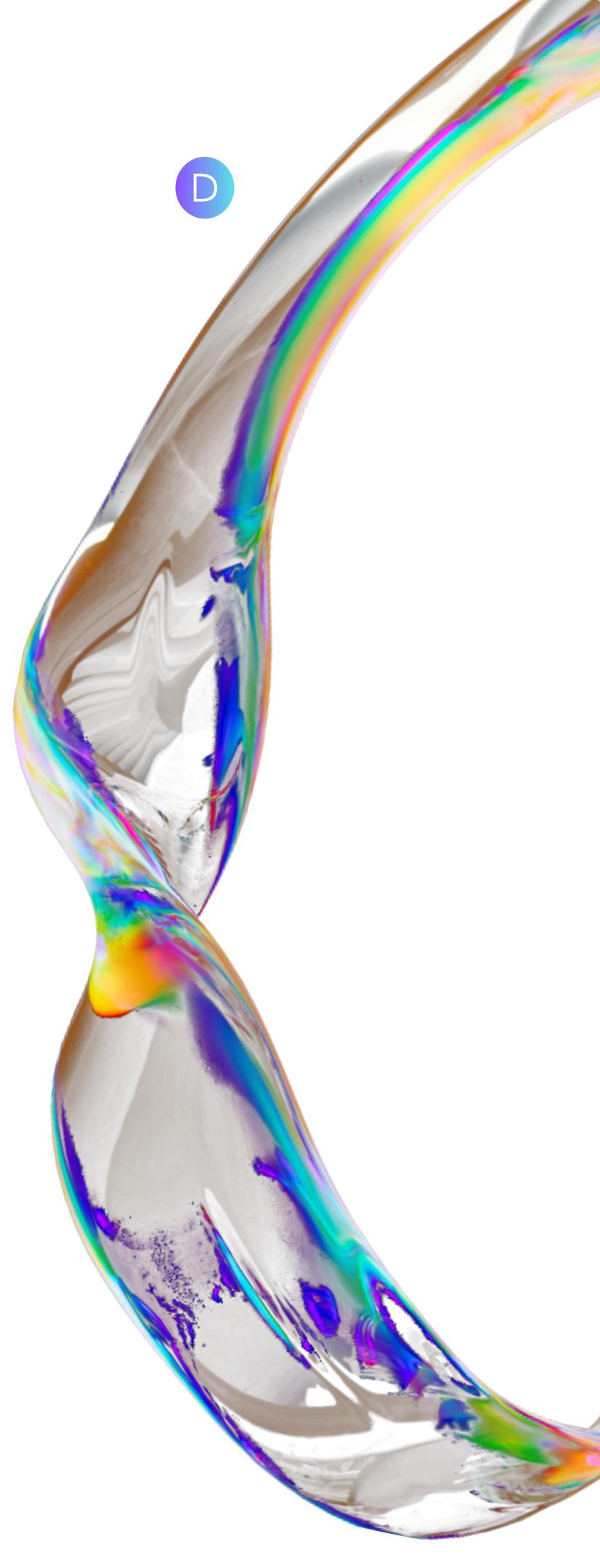
Per ciascun caso d'uso, sono stati identificati gli oggetti che sono stati suddivisi in tre categorie:

**Entity**: rappresentano i dati persistenti gestiti dal sistema.

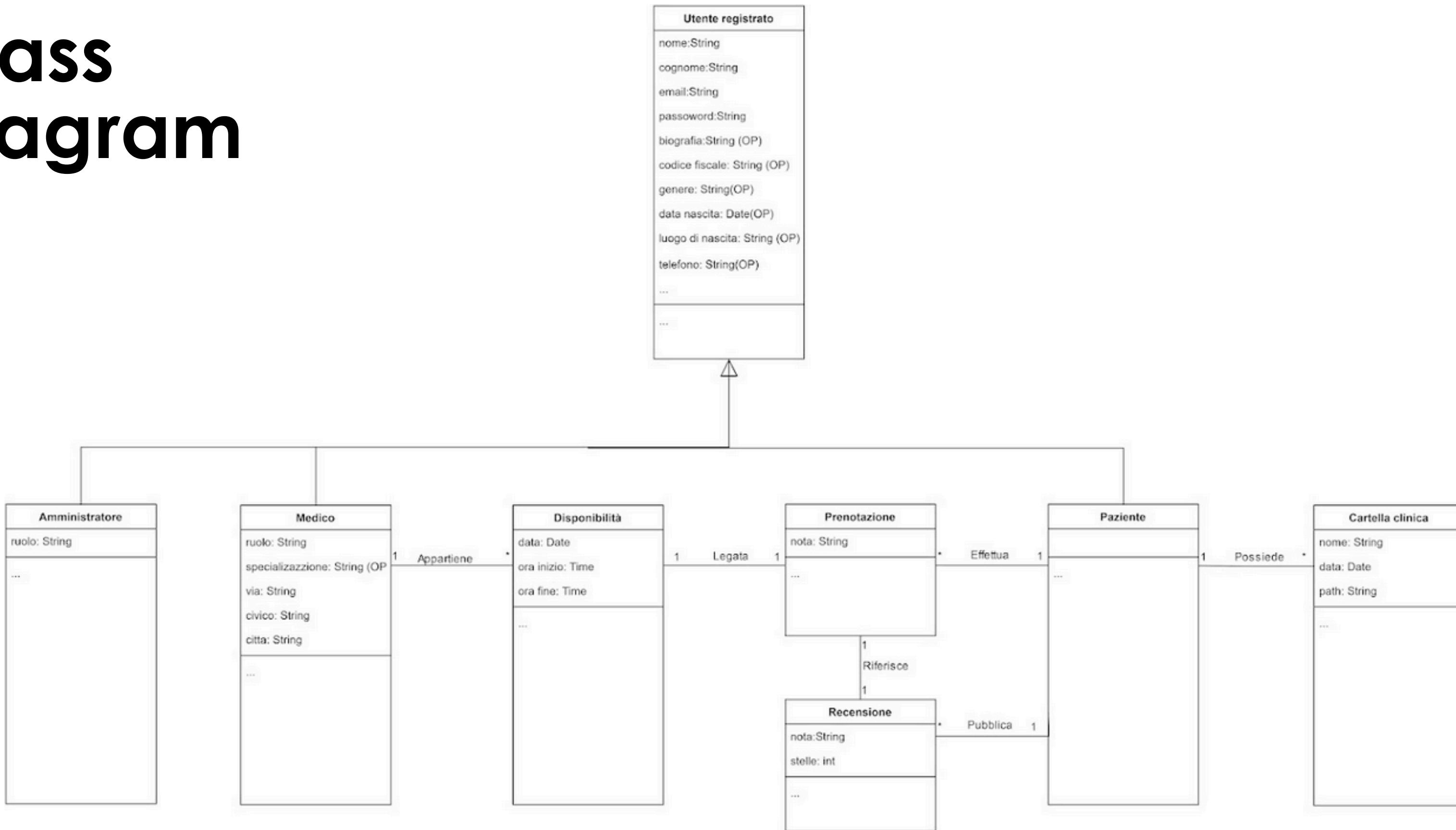
**Boundary**: rappresentano le interazioni tra gli utenti (attori) e il sistema.

**Control**: sono responsabili dell'esecuzione dei casi d'uso e della loro logica.

Successivamente abbiamo utilizzato un **class diagram** per rappresentare la struttura **statica** del sistema in termini di oggetti, classi, attributi, operazioni e associazioni per cercare di formalizzare la conoscenza del dominio applicativo.



# Class Diagram



# Modello dinamico

Per analizzare e documentare il comportamento **dinamico** del sistema, abbiamo utilizzato sequence diagram e statechart diagram.

I protagonisti di questi diagrammi sono gli oggetti “prenotazione” e “recensione”, poiché rivestono un ruolo fondamentale e critico nello sviluppo del sistema.



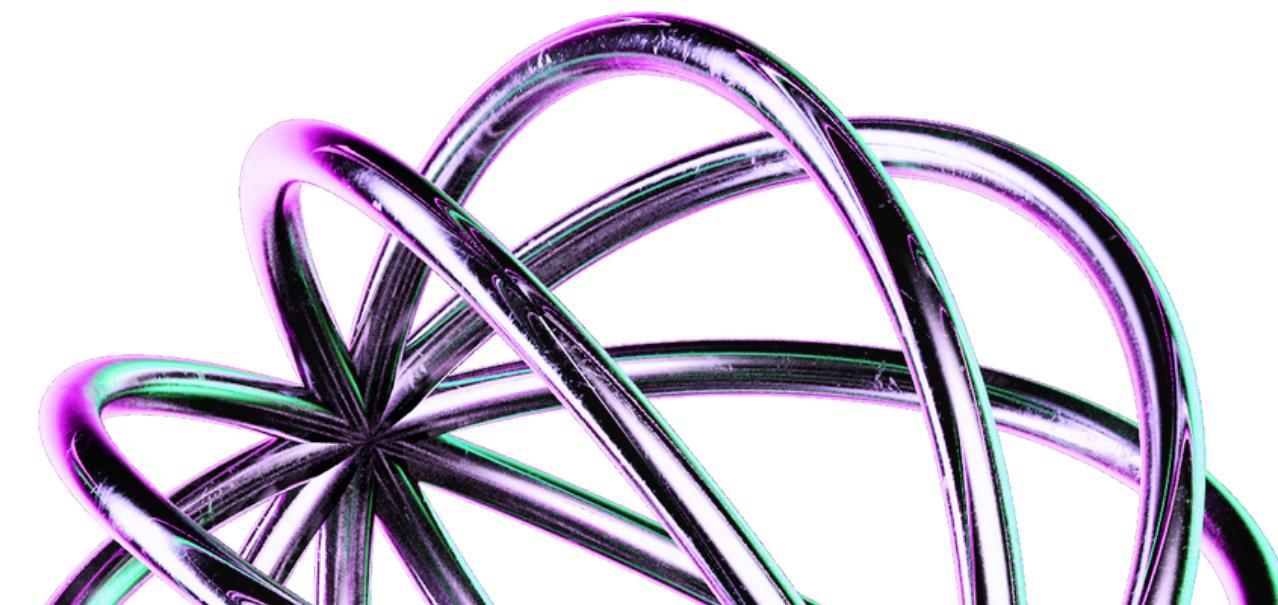
## Sequence Diagram

sono utilizzati per formalizzare il comportamento dinamico del sistema e per visualizzare la comunicazione tra gli oggetti.



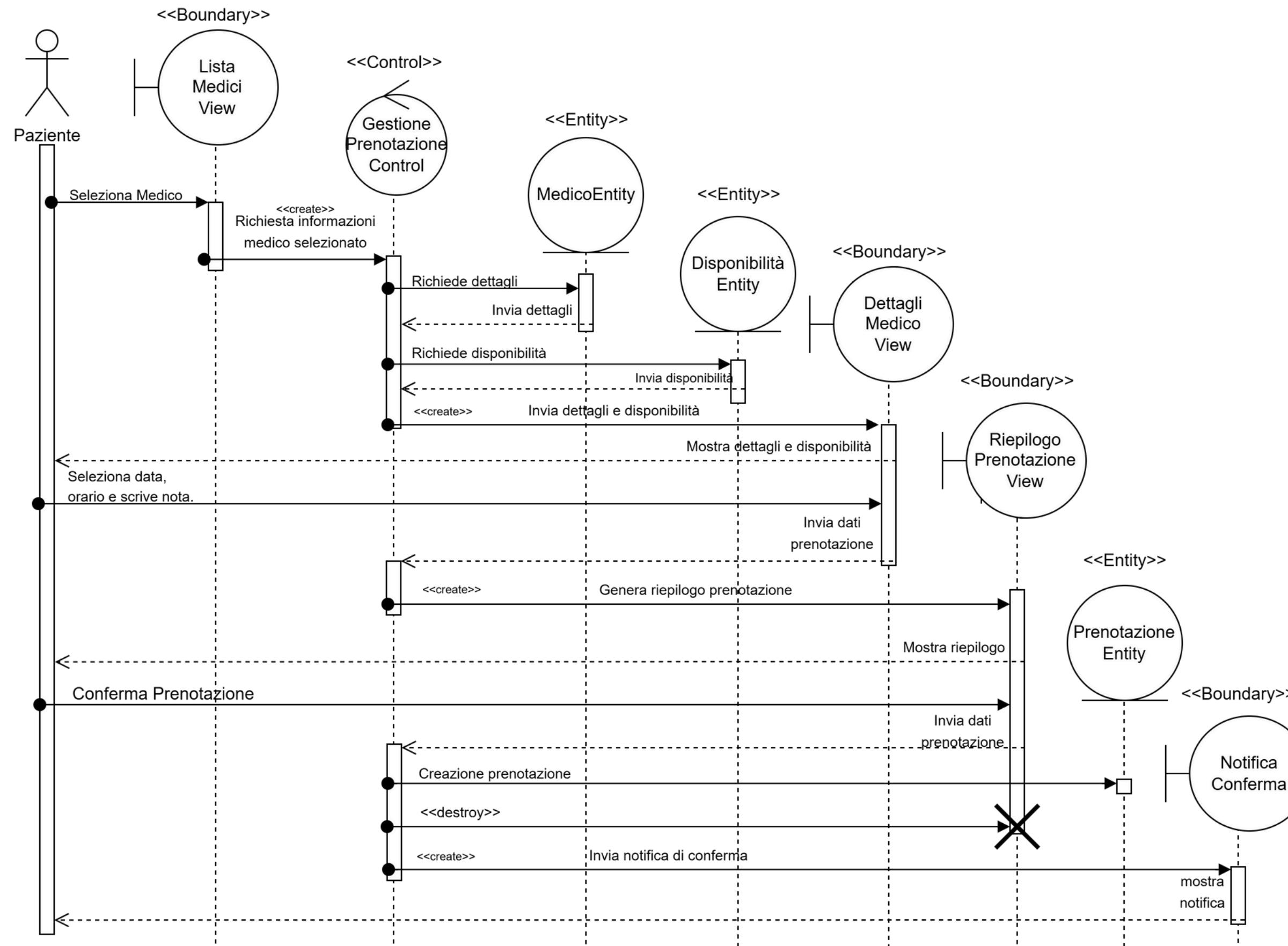
## Statechart Diagram

descrivono il comportamento dinamico di un singolo oggetto come un numero di stati e transizioni tra questi stati. Uno stato rappresenta un particolare insieme di valori per un oggetto.

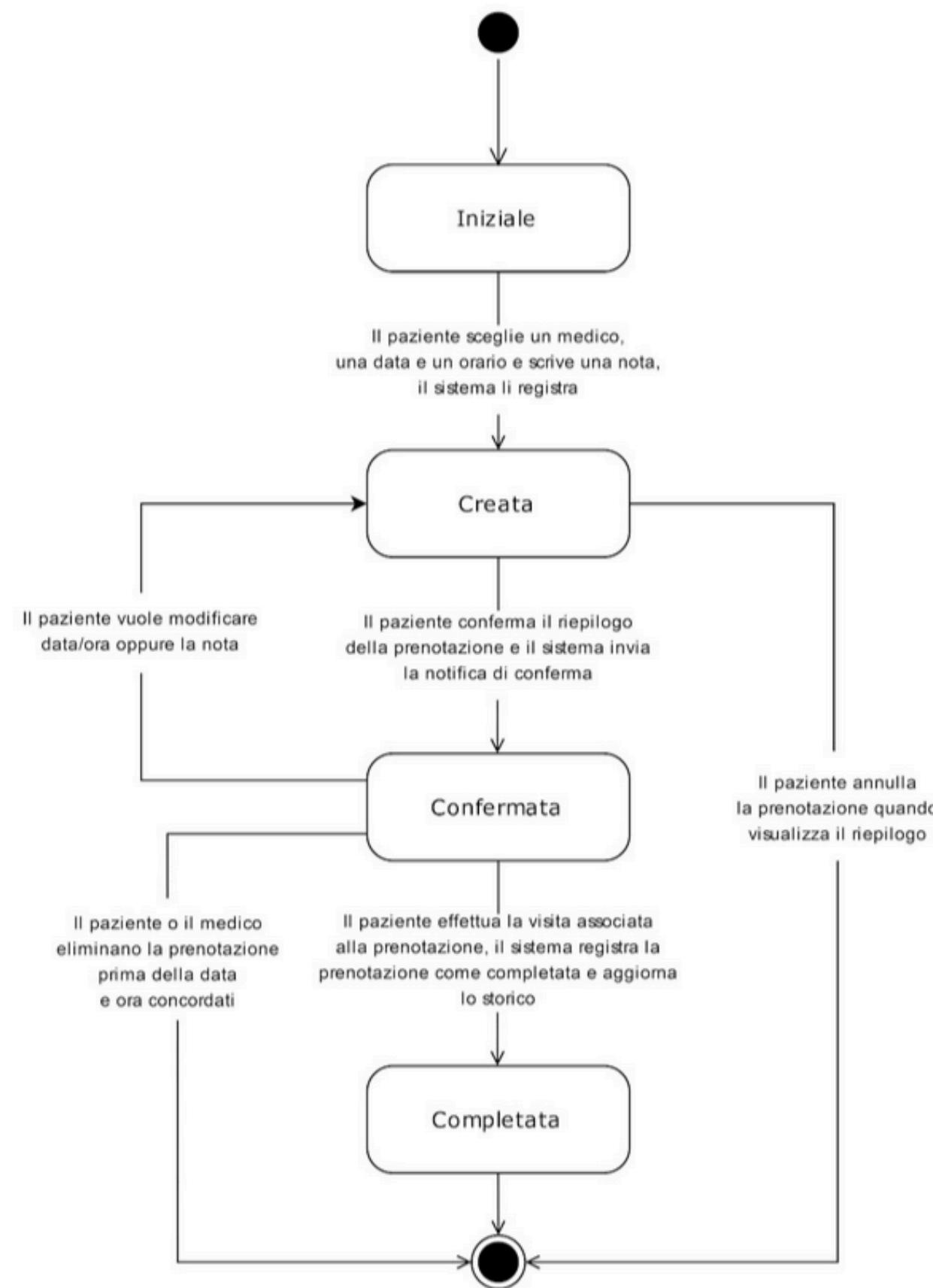


# Sequence Diagram: Effettuare una prenotazione

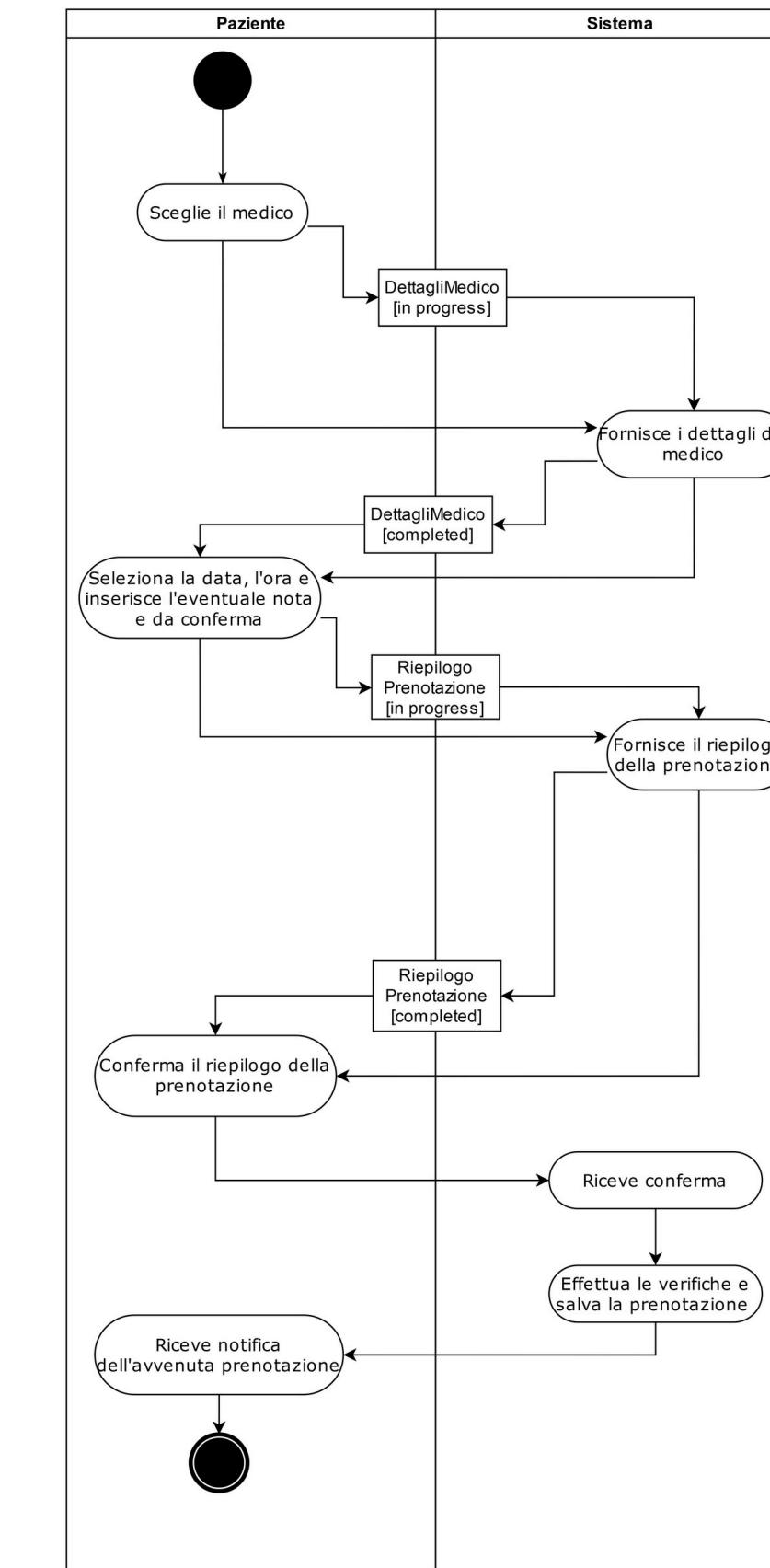
D



# Statechart Diagram: Prenotazione

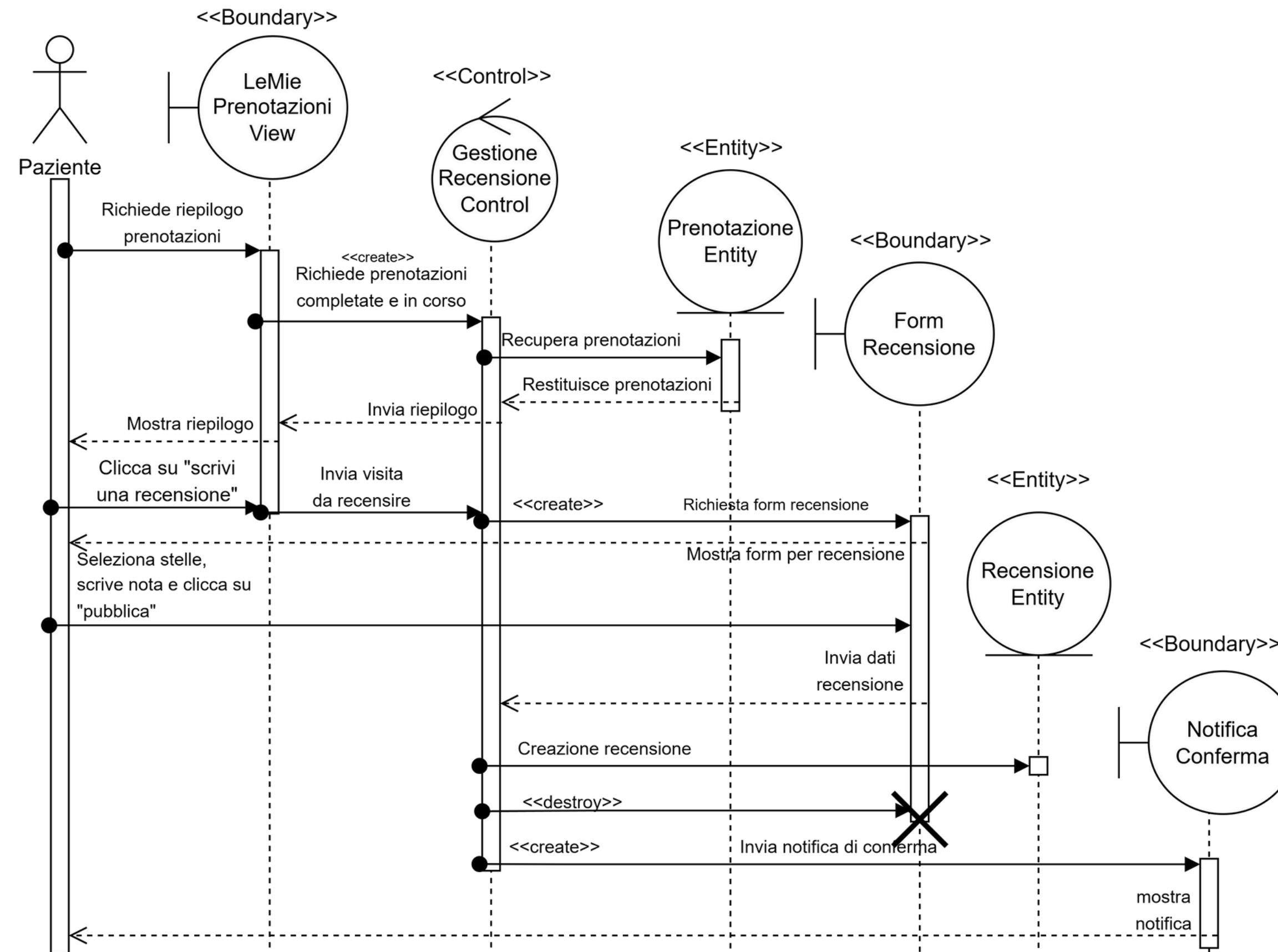


# Activity diagram: Effettuare prenotazione



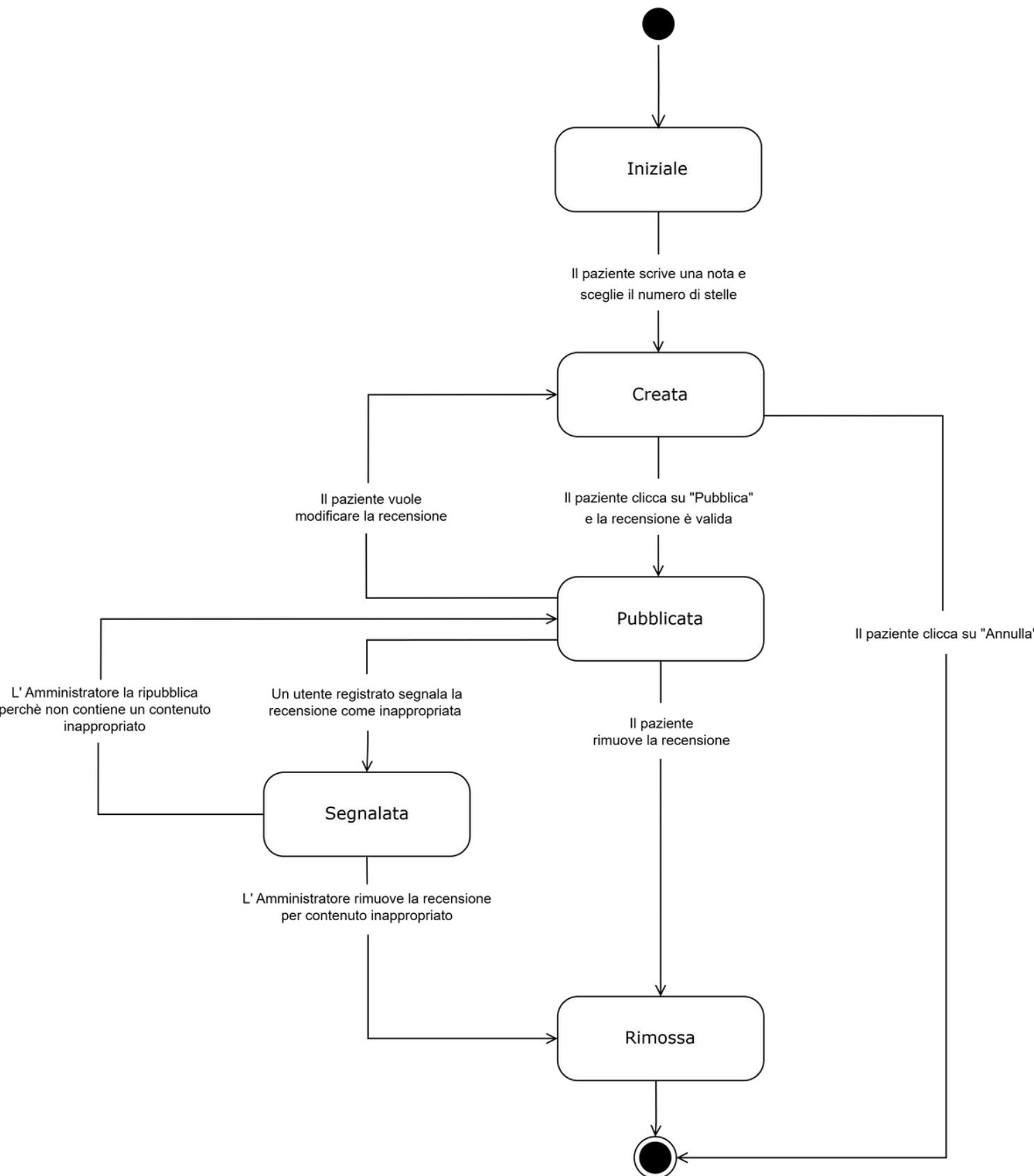
# Sequence Diagram: Inserimento recensione

V



# Statechart Diagram: Recensione

V



# Mock-ups

**MedConnect**

Effettua il login al tuo account

E-mail  
Password

Login

Non hai ancora un account? [Registrati!](#)

**MedConnect**

**FAQ**

**AREA UTENTE**

- Informazioni Account
- Le mie prenotazioni
- (...)
- Logout

Cerca il medico e prenota la tua visita!

Ruolo / Nome      Città

Es : Oculista, Mario Rossi ..      Es: Napoli ..

Ricerca

(Footer)

**MedConnect**

**Area Utente**  
Benvenuto,  
**Mario Rossi**

Path: [Home](#) > [Area Utente](#) > [Le mie prenotazioni](#)

**Prossimi appuntamenti (1)**

Studio privato di oculistica  
Dottor. Guido La Vespa  
Aprilia, via dei Motorini 125

16 Febbraio 2025  
16:00, Domenica

[Dettagli della prenotazione](#)      [Modifica](#)      [Cancella prenotazione](#)

**Visite completeate (2)**

Studio privato di oculistica  
Dottor. Guido La Vespa  
Aprilia, via dei Motorini 125

10 Novembre 2024  
11:00, Domenica

[Dettagli della prenotazione](#)      [Scrivi una recensione](#)

Studio privato di oculistica  
Dottor. Guido La Vespa

03 Novembre 2024  
18:00, Domenica

**MedConnect**

**Area Utente**  
Benvenuto,  
**Mario Rossi**

Path: [Home](#) > [Area Utente](#) > [Informazioni](#) > [Informazioni di base](#)

**Informazioni**

Informazioni sull'Account  
Informazioni di base  
Cart. cliniche / Specializzazioni

**Le mie prenotazioni**

**Le mie recensioni**

**Informazioni di base**

**Biografia**  
Inserisci qui la tua biografia (max x parole)

**Data di nascita**  gg/mm/aaaa      **Luogo di Nascita**  Napoli

**Codice Fiscale**  Ese: RSSMR447P03B615M

**Sesso**  
 Maschio       Femmina       Altro

**Contatto telefonico**  Ese: +393141592654

**Firma**  Paziente/Medico

[Salva](#)      [Annulla](#)

X Elimina il mio Account

**Informazioni**

Informazioni sull'Account  
Informazioni di base  
Cart. cliniche / Specializzazioni

**Le mie prenotazioni**

**Le mie recensioni**

**Le mie recensioni**

★★★★★  
Professionale, cortese e attento alle esigenze del paziente.

[Visualizza](#)      [Modifica](#)      [Elimina](#)

★★★★★  
Ottima esperienza, medico competente e disponibile.

[Visualizza](#)      [Modifica](#)      [Elimina](#)

★★★★★  
Empatico, preciso e sempre pronto ad aiutare.

[Visualizza](#)      [Modifica](#)      [Elimina](#)

# SDD

## System Design Document

Introduzione

---

Architettura del sistema corrente

---

Architettura del sistema proposto

---

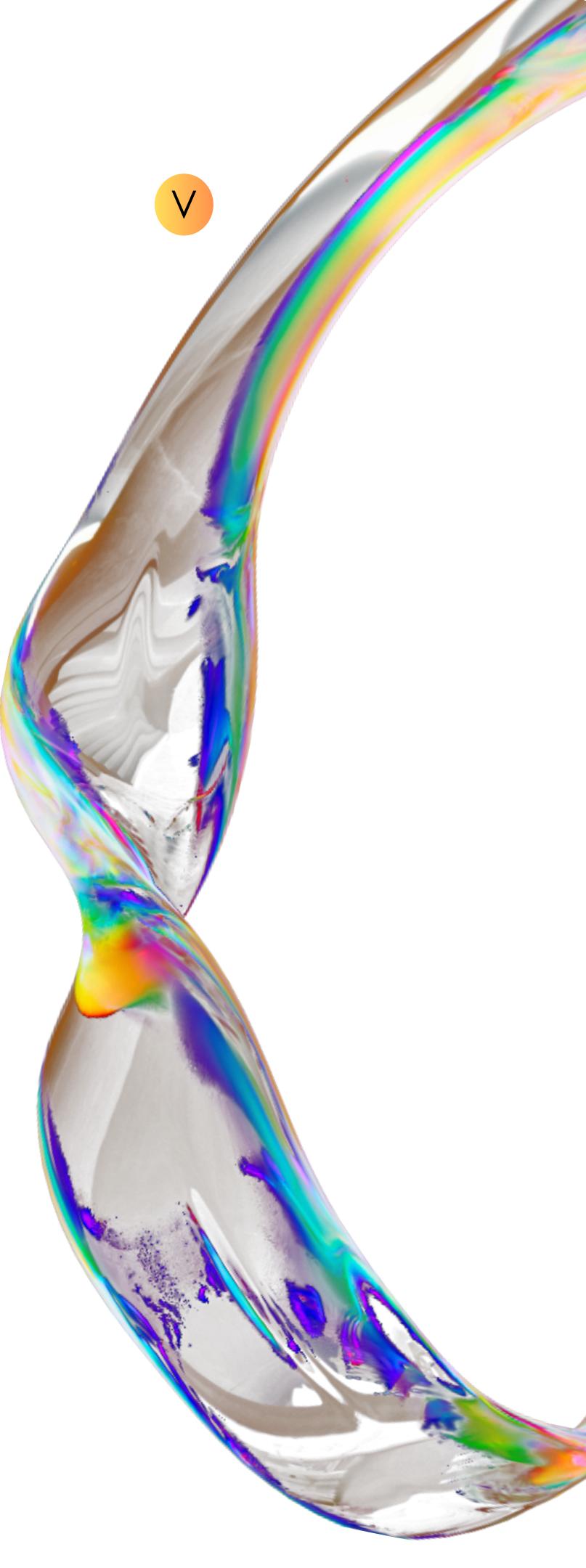
Servizi dei sottosistemi

---

**Extra** : Design pattern

---

V



# Introduzione : Design Goal

Analizzando i requisiti non funzionali, abbiamo identificato i design goal, ovvero qualità che aiutano a stabilire le priorità nello sviluppo del sistema e che guidano gli sviluppatori a prendere decisioni di progettazione.

Ogni design goal è caratterizzato da:  
rank (1-13), ID, descrizione, categoria (Performance,  
Dependability, Maintenance, EndUser), RNF di origine.

Trade-off	Razionale
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.

Rank	ID	Descrizione	Categoria	RNF di origine
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
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

# Scelta dell'architettura

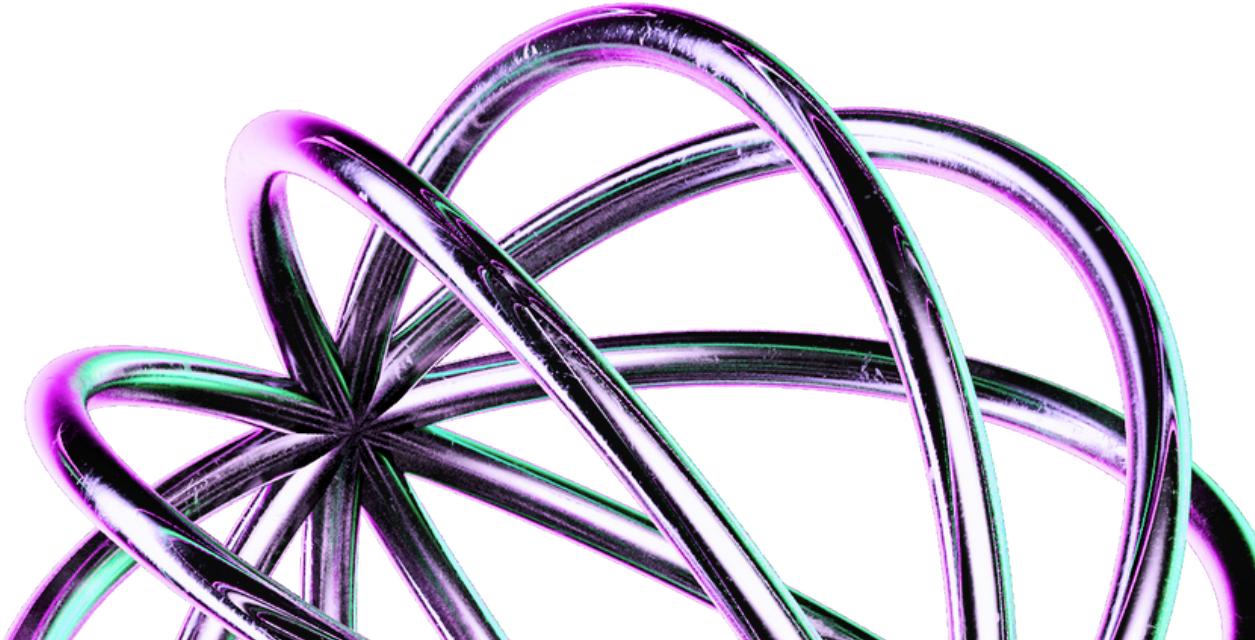
MedConnect adotta un'architettura a tre livelli (**three-tier**), organizzata in:

**Livello di interfaccia:** gestisce l'interazione con l'utente tramite oggetti boundary (package **View**).

**Livello di logica applicativa:** comprende oggetti controller e entity per elaborazioni, controllo delle regole e notifiche. (package **Controller**)

**Livello di archiviazione:** si occupa di archiviazione, recupero e interrogazione dei dati persistenti. (package **Model**)

Questa struttura garantisce **scalabilità e riduce l'accoppiamento** tra i sottosistemi.



## Nota bene

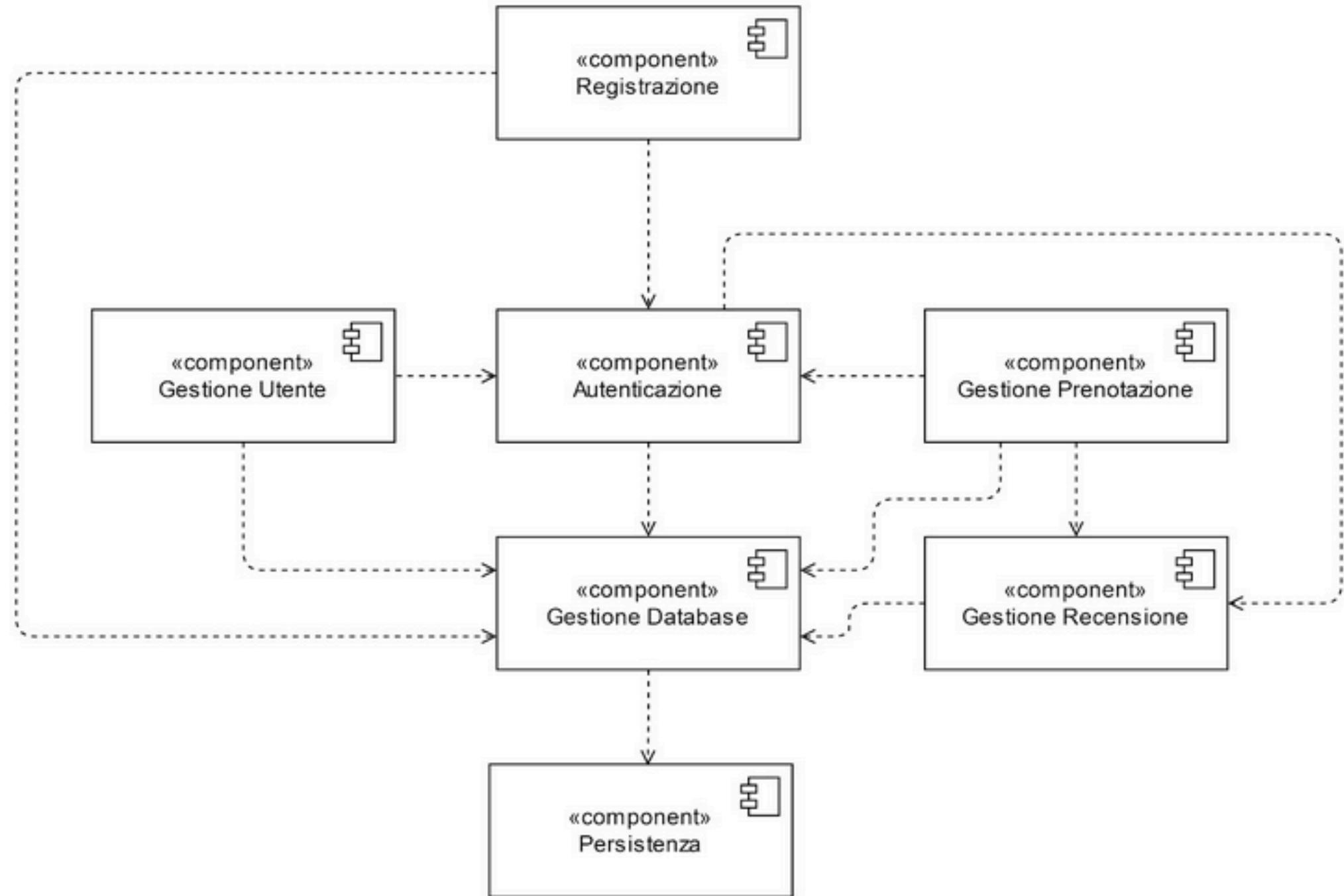
nell'architettura MVC, la View dipende dal Model grazie al design pattern dell'Observer.  
nell'architettura **three-tier non c'è l'observer** quindi i livelli sono più indipendenti.

# Decomposizione in sottosistemi

## Component Diagram

I sottosistemi individuati sono:

1. **Registrazione**: gestione della registrazione degli utenti.
2. **Autenticazione**: verifica degli utenti registrati.
3. **Gestione Utente**: gestione di informazioni, cartelle cliniche, specializzazioni e profili di medici e pazienti.
4. **Gestione Prenotazione**: gestione delle prenotazioni (creazione, modifica, eliminazione, visualizzazione).
5. **Gestione Recensione**: gestione delle recensioni dei pazienti.
6. **Gestione Database**: interazione con il sistema di persistenza.
7. **Persistenza**: archiviazione dei dati nel database.

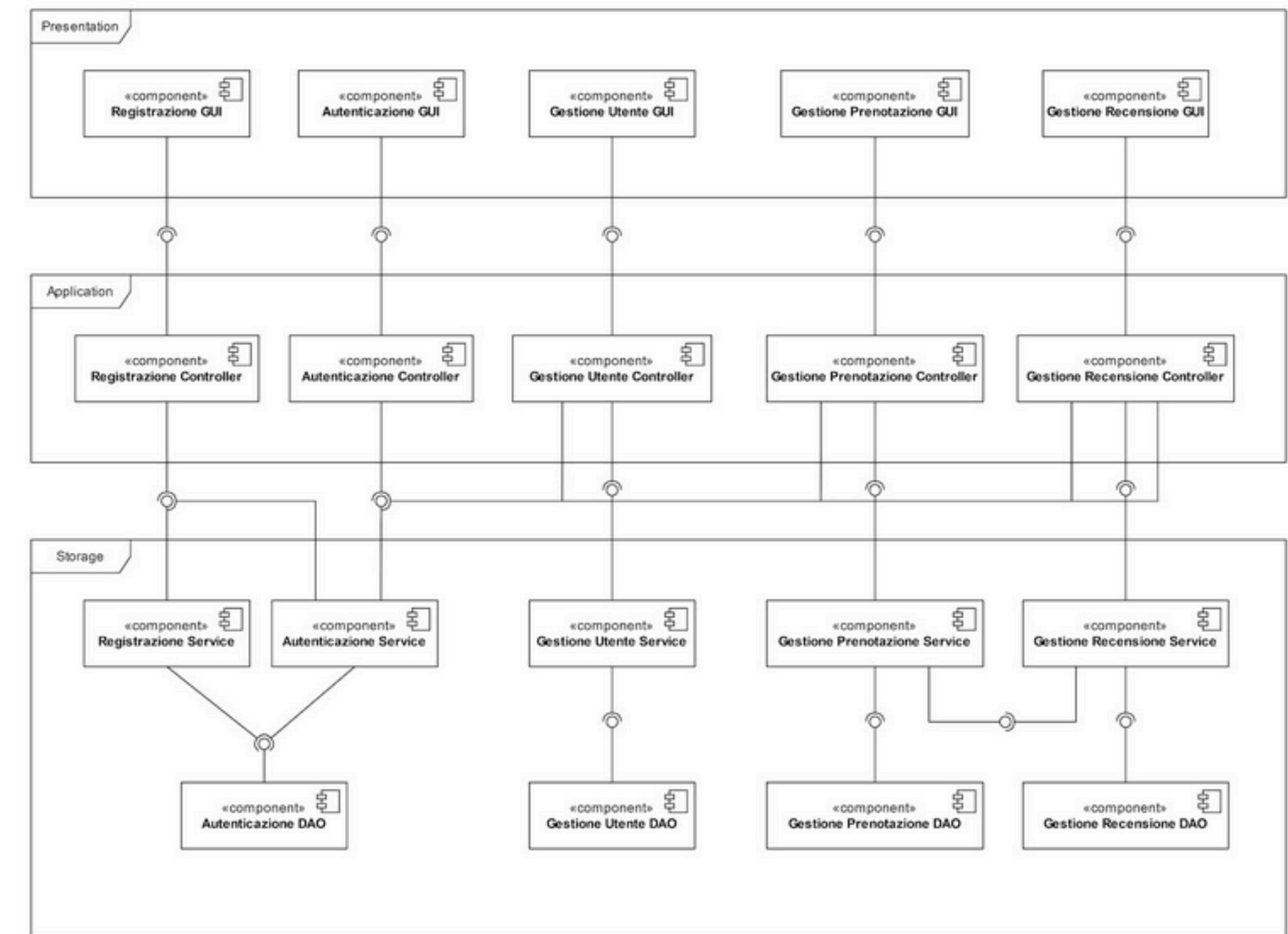


# Decomposizione in sottosistemi

## Architectural Diagram

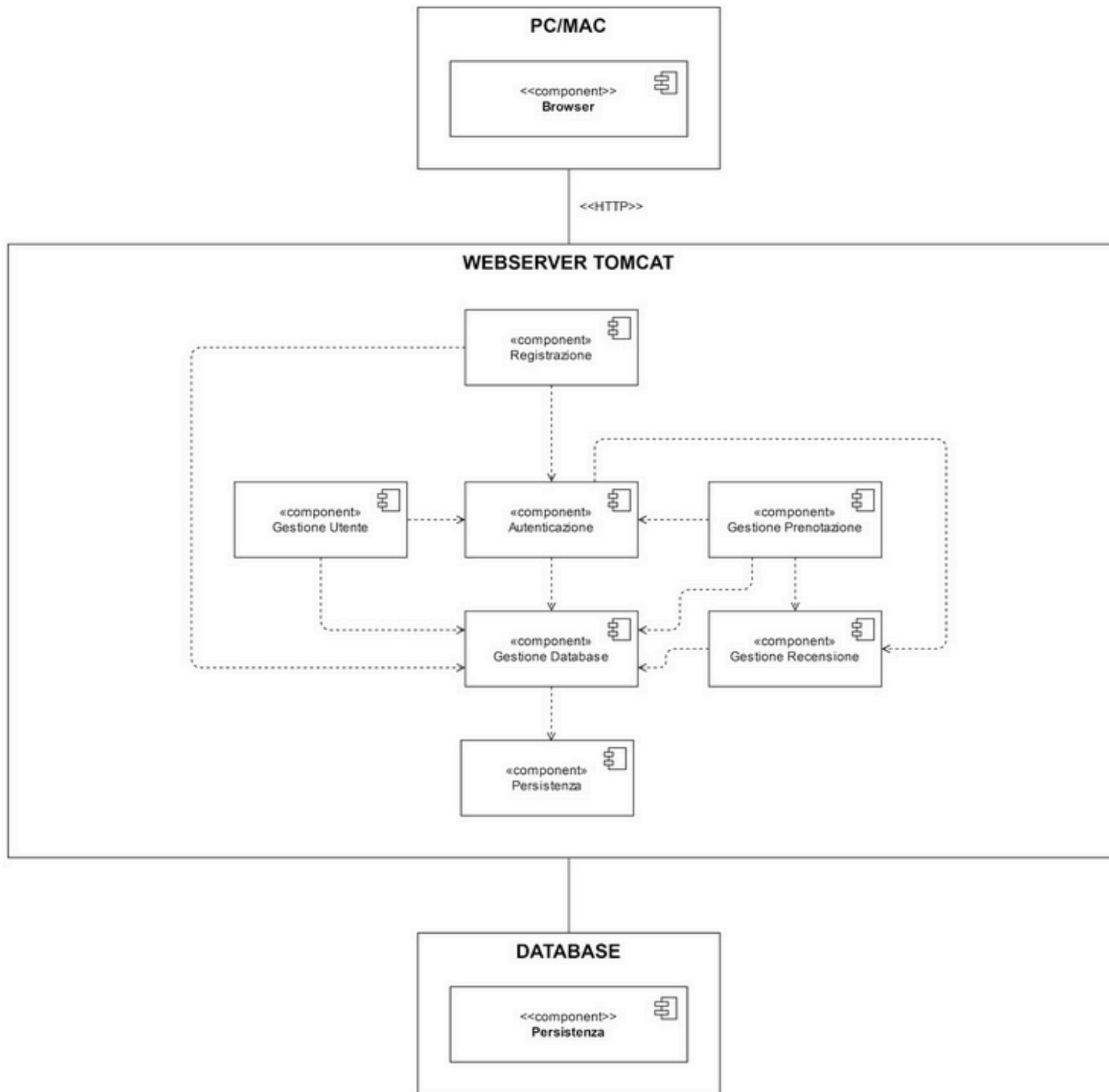
Componenti del diagramma architetturale:

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



# Decomposizione in sottosistemi

Deployment Diagram

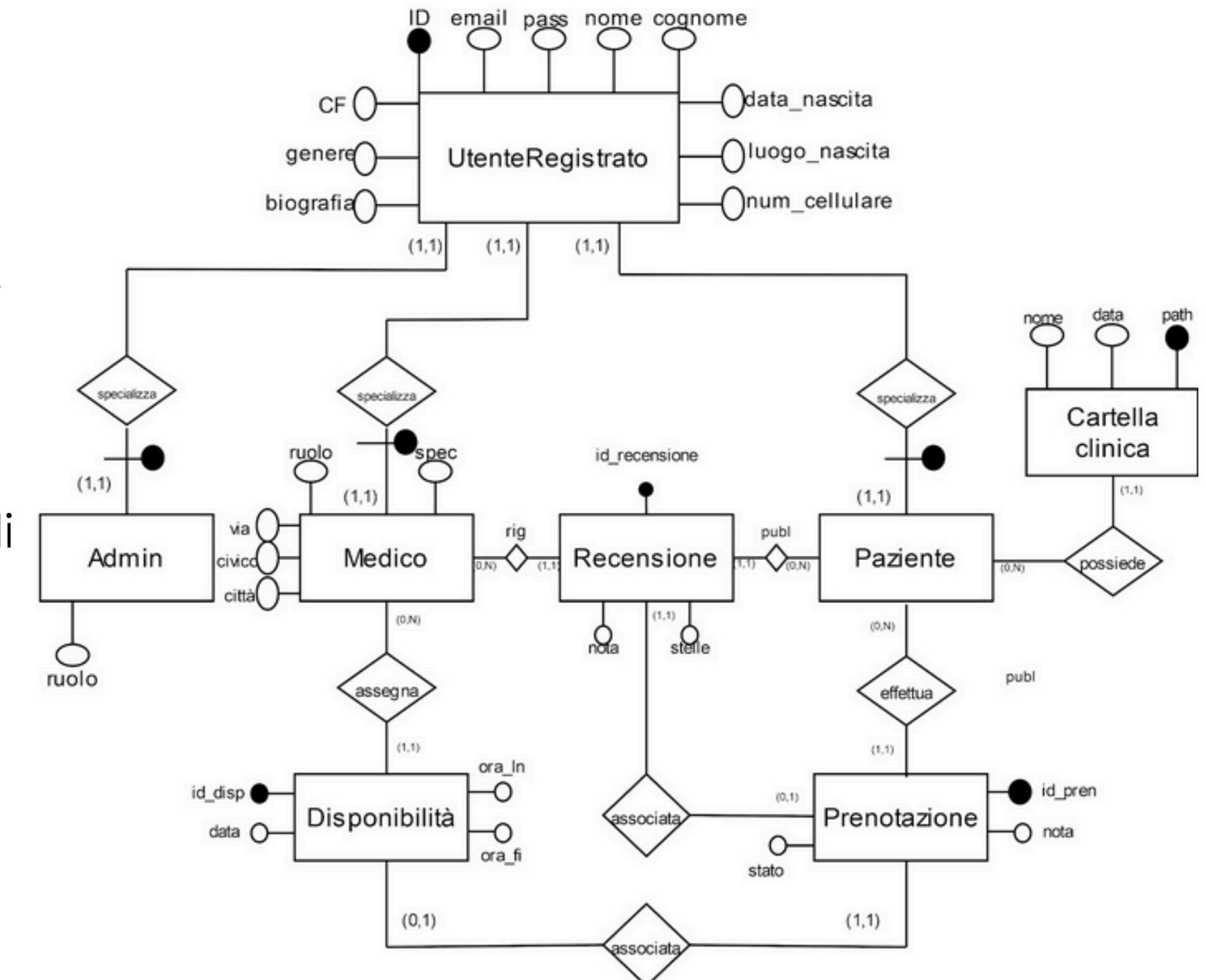


# Gestione dati persistenti

I dati persistenti sono informazioni che devono essere mantenute anche dopo la chiusura del sistema. MedConnect richiede:

1. L'esecuzione di **query complesse** sugli attributi dei dati;
2. La gestione di **grandi volumi** di dati;
3. La garanzia dell'integrità dei dati anche in caso di **accessi simultanei**.

Per soddisfare queste esigenze, si è deciso di utilizzare un **database relazionale** per la memorizzazione dei dati persistenti.



# Controllo degli accessi e sicurezza

Per monitorare quali attori possono accedere ai servizi offerti dal sistema, abbiamo utilizzato una **matrice di accesso**, strutturata come segue:

**Righe**: ogni riga rappresenta una funzionalità del sistema per cui si desidera controllare l'accesso.

**Colonne**: ogni colonna rappresenta un attore del sistema.

**Celle**: ogni cella indica i diritti di accesso dell'attore (colonna) alla funzionalità corrispondente (riga).

# Flusso di controllo globale

MedConnect utilizza un modello di controllo basato sugli eventi (**event-driven**), in cui ogni funzionalità è attivata dalle azioni degli utenti tramite un'interfaccia grafica.

Ogni azione genera un evento, elaborato da un componente dedicato che instrada il flusso verso il sottosistema responsabile della logica di controllo.

Questo, a sua volta, comunica con i servizi della logica applicativa per completare l'operazione, garantendo un funzionamento efficiente e reattivo, ideale per una piattaforma web.

# Condizioni limite

Le condizioni limite del sistema includono:

- avvio;
- spegnimento;
- **fallimento**;
- errori di accesso ai dati persistenti.

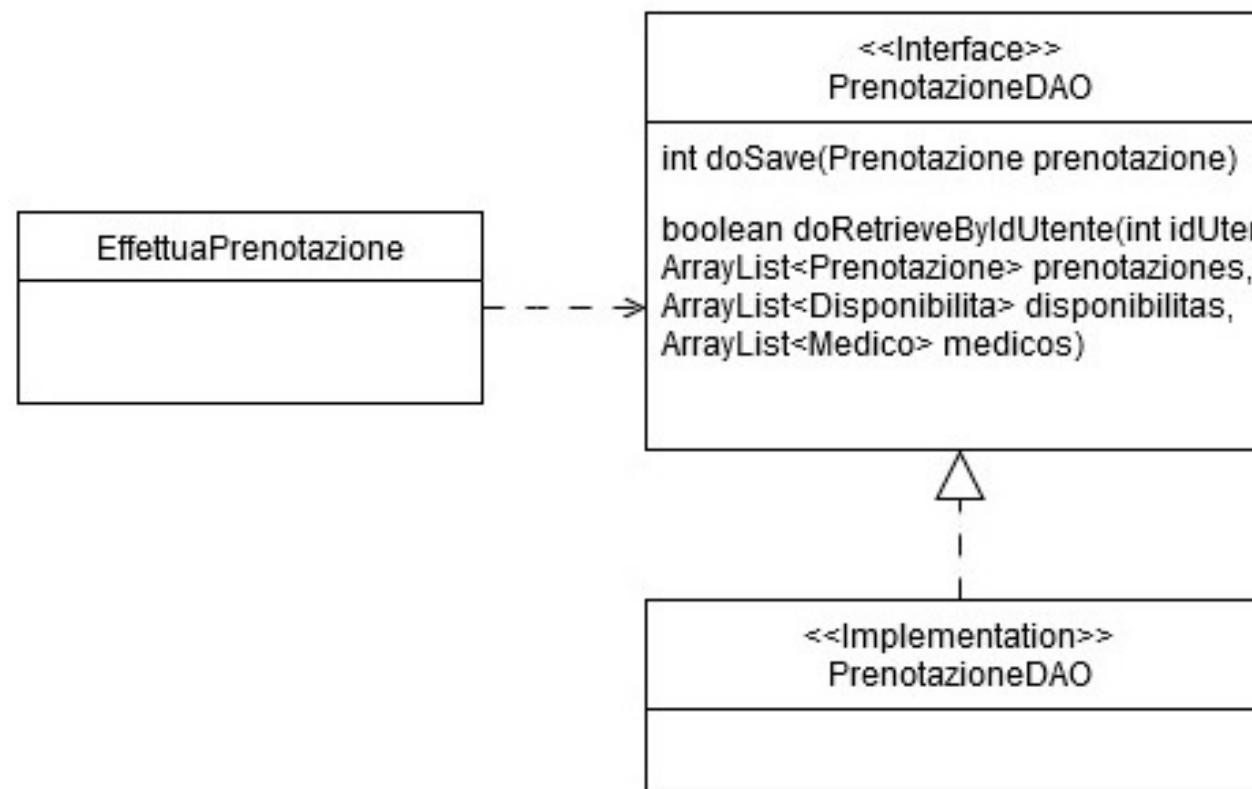
I dettagli completi, descritti tramite **Boundary Use Case**, sono documentati nella sezione Boundary Conditions del RAD.

<b>Identificativo</b>	<i>Fallimento del sistema</i>	Data	11/12/2024
UCBC_3		Vers.	1.0
		Autore	
<b>Descrizione</b>	Lo UC definisce il comportamento del Sistema in caso di fallimento		
<b>Attore Principale</b>	<b>Amministratore</b>		
<b>Attori secondari</b>	<b>NA</b>		
<b>Entry Condition</b>	Il sistema si arresta inaspettatamente		
<b>Exit condition</b>	Il sistema viene riavviato correttamente		
<b>On success</b>			
<b>Exit condition</b>	Il sistema non viene riavviato correttamente		
<b>On failure</b>			
<b>FLUSSO DI EVENTI PRINCIPALE</b>			
1	<b>Amministratore:</b>	<b>Include</b> USBC_1 (Avvio del sistema)	

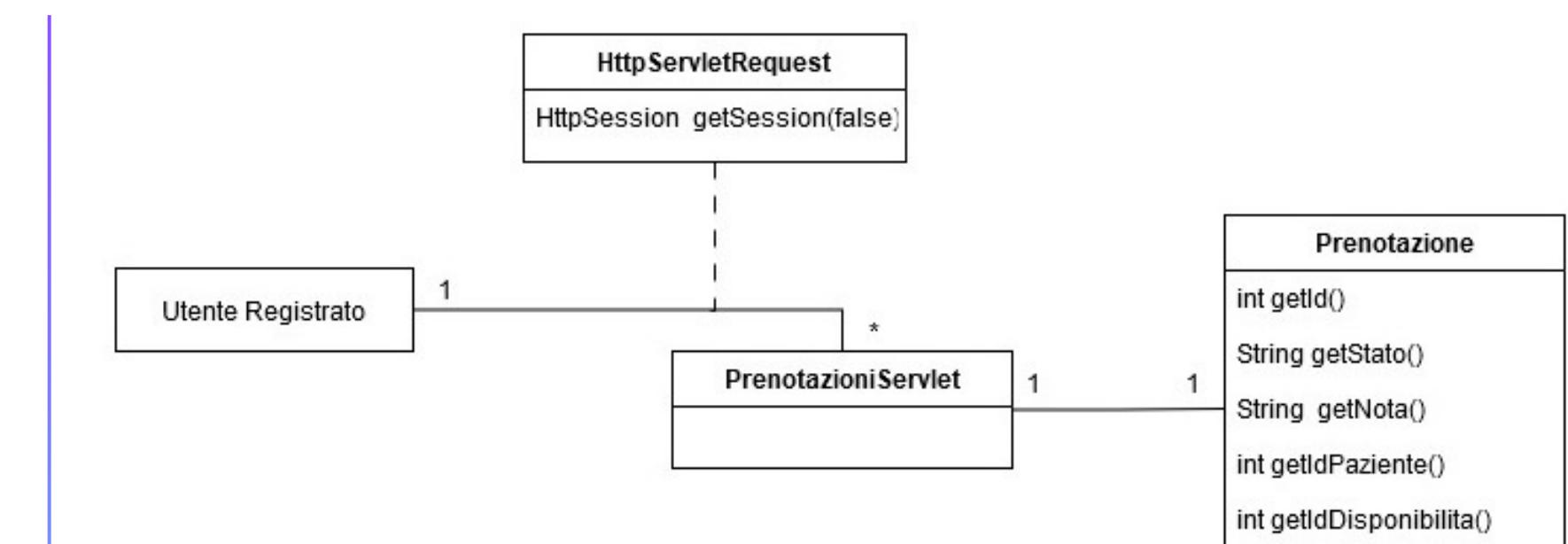
# Design Pattern

## Facade

## Proxy



In questo esempio il sottosistema implementato PrenotazioneDAO avrà una propria interfaccia pubblica PrenotazioneDAO, che esporrà tutte le funzionalità fornite e realizzate dal componente implementato PrenotazioneDAO. Tale interfaccia sarà poi implementata da implementazione 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.



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.

# Testing

- Ha successo se il sistema evidenzia una failure.
- Fallisce se l'output generato dal sistema coincide con quello previsto dall'oracolo.

Test Plan

---

Test Case Specification

---

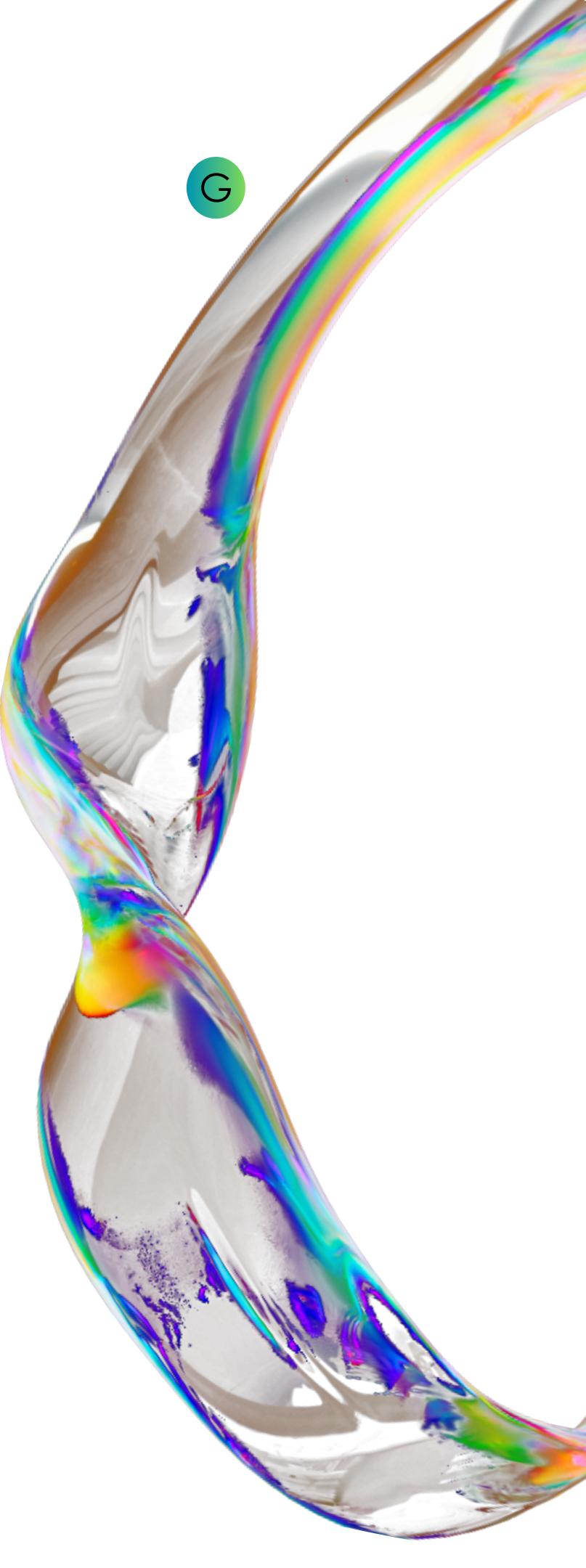
Test Incident Report

---

Test Summary Report

---

G



# Test Plan

Lista delle funzionalità di cui si effettuerà il testing:

- Gestione Utente: Inserimento informazioni di base
- Gestione Prenotazione: Effettuare prenotazione
- Gestione Recensione: Inserimento recensione

Articolato in:

- Testing di Unità (tramite le classi di test, utilizzando JUnit, Mockito e stimando i risultati con JaCoCo)
- Testing di Integrazione (badget terminato)
- Testing di Sistema (tramite Selenium IDE)

Definiti i **Test Case** utilizzando il category partition, con il quale gli input verranno suddivisi in classi di equivalenza per ridurre al minimo il numero di test case.

Affianco un'es. di test case di Gestione Recensione: Inserimento recensione.

## 9.3 Gestione Recensione

### 9.3.1 Inserimento recensione

Parametro : Stelle	
FORMATO:	
Nome Categoria	Scelta per categoria
Formato [FS]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FS_OK]
Parametro : Nota	
Nome Categoria	Scelta per categoria
Lunghezza [LN]	1. Lunghezza > 255 = false [error] 2. Lunghezza <= 255 = true [PROPERTY LN_OK]

Test Case ID	Test Frame	Esito
TC_3.1	FS1	<b>Errato</b> : numero stelle errato.
TC_3.2	FS2, LN1	<b>Errato</b> : lunghezza nota della recensione non corretta.
TC_3.3	FS2, LN2	<b>Corretto.</b>

# Test Case Specification

Descrivere in modo dettagliato i test case progettati per verificare il funzionamento del sistema rispetto ai requisiti specificati.

Nel Test Case Specification i test case vengono istanziati utilizzando dati reali e relativo oracoli.

## 1.3 Gestione Recensione

### 1.3.1 Inserimento recensione

Test Case ID	TC_3.1	Test Frame	FS1		
<b>Pre-condizione</b>					
1. L'utente si trova nella sezione "Le mie prenotazioni".					
<b>Flusso di eventi</b>					
2. L'utente compila il form:					
Input	Valore				
Stelle	0				
Nota	Visita eccellente: medico professionale, chiaro e cortese. Mi ha messo a mio agio e fornito soluzioni efficaci. Consigliato!				
3. L'utente clicca sul pulsante "Pubblica"					
<b>Oracolo</b>					
La pubblicazione della recensione non va a buon fine poiché il campo "Stelle" non rispetta il formato.					

Test Case ID	TC_3.2	Test Frame	FS2, LN1		
<b>Pre-condizione</b>					
1. L'utente si trova nella sezione "Le mie prenotazioni".					
<b>Flusso di eventi</b>					
2. L'utente compila il form:					
Input	Valore				
Stelle	5				
Nota	La visita con il dottor Fanti è stata eccellente. Professionale, cortese e chiaro nelle spiegazioni, mi ha fatto sentire a mio agio. Ha ascoltato con attenzione i miei sintomi e proposto un piano di cura efficace. Studio pulito e ben organizzato. Consiglio vivamente!				
3. L'utente clicca sul pulsante "Pubblica"					
<b>Oracolo</b>					
La pubblicazione della recensione non va a buon fine poiché il campo "Nota" non rispetta la lunghezza.					

# Test Incident Report

Fornisce un report dettagliato sugli esiti dei test eseguiti, seguendo le indicazioni di quanto definito nel TP e nella TCS.

E' stato utilizzato il **Plugin Selenium IDE per Mozilla Firefox**.

E' stata creata la tabella **Test Incident Report Table** la quale contiene il report dell'esecuzione dei test di sistema relativi alla piattaforma MedConnect:

<b>Test Incident Report</b>		<b>Esecuzione: 1</b>	<b>Data Iterazione: 05/01/2025</b>			<b>Note</b>	<b>Test Passati</b>	<b>Test non passati</b>	<b>Esito</b>
<b>Test Case ID</b>	<b>Test Incident ID</b>	<b>Tester</b>	<b>Data</b>	<b>Esito</b>					
TC_1.1	TI_1.1	Tutti	05/01/2025	Not Passed	L'id del paziente risultava essere nullo		5	1	Esecuzione rifiutata
TC_1.2	TI_1.2	Tutti	05/01/2025	Passed					
TC_1.3	TI_1.3	Tutti	05/01/2025	Passed					
TC_3.1	TI_3.1	Tutti	05/01/2025	Passed					
TC_3.2	TI_3.2	Tutti	05/01/2025	Passed					
TC_3.3	TI_3.3	Tutti	05/01/2025	Passed					

<b>Test Incident Report</b>		<b>Esecuzione: 2</b>	<b>Data Iterazione: 05/01/2025</b>			<b>Note</b>	<b>Test Passati</b>	<b>Test non passati</b>	<b>Esito</b>
<b>Test Case ID</b>	<b>Test Incident ID</b>	<b>Tester</b>	<b>Data</b>	<b>Esito</b>					
TC_1.1	TI_1.1	Tutti	05/01/2025	Passed			6	0	Esecuzione accettata
TC_1.2	TI_1.2	Tutti	05/01/2025	Passed					
TC_1.3	TI_1.3	Tutti	05/01/2025	Passed					
TC_3.1	TI_3.1	Tutti	05/01/2025	Passed					
TC_3.2	TI_3.2	Tutti	05/01/2025	Passed					
TC_3.3	TI_3.3	Tutti	05/01/2025	Passed					

# Test Summary Report

Fornisce un riepilogo che include l'esito complessivo dei test di sistema, i cui dettagli sono contenuti in questo report.

## Report di JaCoCo:

Branch Coverage	Line Coverage
70,7%	55,9%

**Branch Coverage:** misura la percentuale di ramificazioni condizionali (come if, switch, ecc.) che sono state eseguite nei test.

**Line Coverage:** misura la percentuale di righe di codice che sono state effettivamente eseguite durante i test.

## Report test di Sistema tramite Selenium:

Esecuzione	Data	#Fallimenti	#Successi
Esecuzione 1	05/01/25	1	5
Esecuzione 2	05/01/25	0	6



**Lezioni  
imparate**

# Cusati Daniel



Cosa hai imparato?

- L'importanza di stabilire obiettivi chiari e completarli entro le scadenze.
- A concentrarmi meno sulla revisione continua e a evitare la ricerca ossessiva della perfezione.
- A non generare un Javadoc direttamente sul desktop...



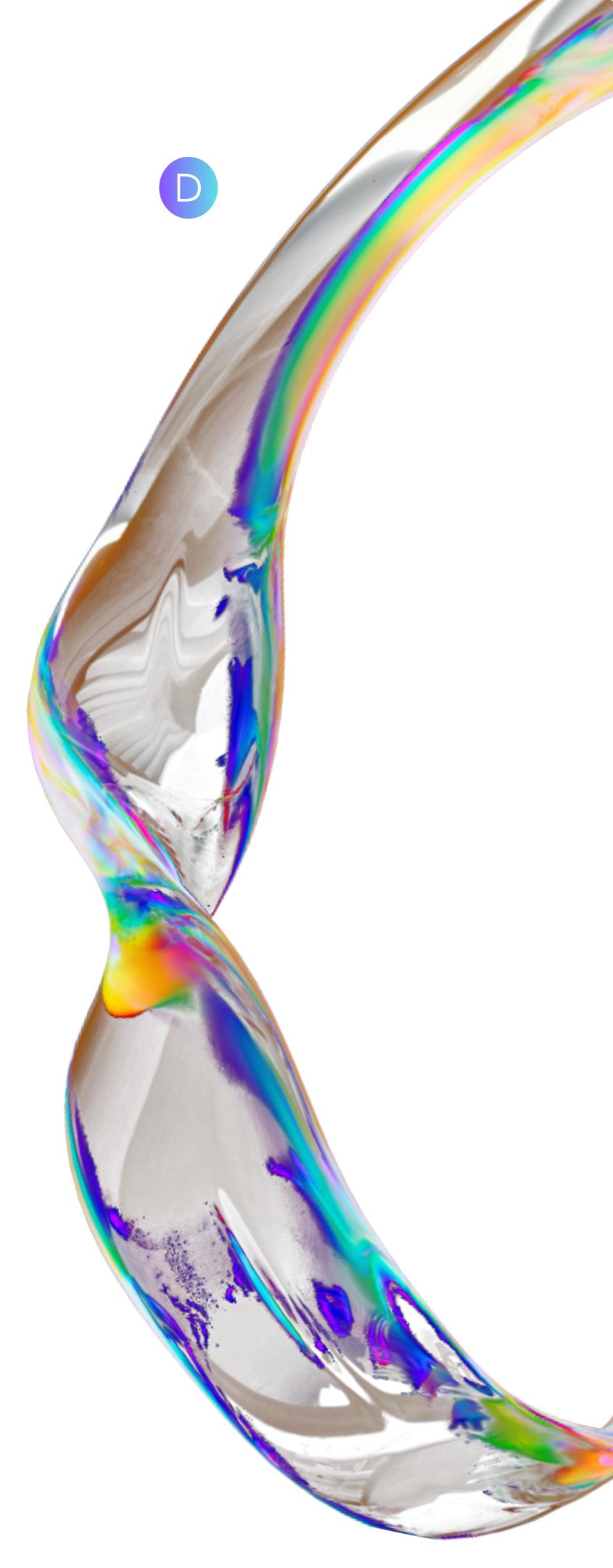
Qual è stata la lezione più difficile?

- Mantenere una coerenza generale dal primo paragrafo del RAD fino alla fase di implementazione finale.



Qual è stato il tuo contributo?

- Mi sono occupato di coordinare alcune attività chiave del progetto, promuovendo l'introduzione delle scadenze. Inoltre, ho fornito supporto al team sia dal punto di vista tecnico che organizzativo.



# Medica Vincenzo



Cosa hai imparato?

- Ho sviluppato una maggiore capacità di collaborazione all'interno di un team.
- Ho compreso l'importanza di una pianificazione e un'organizzazione dettagliata per il successo del progetto.



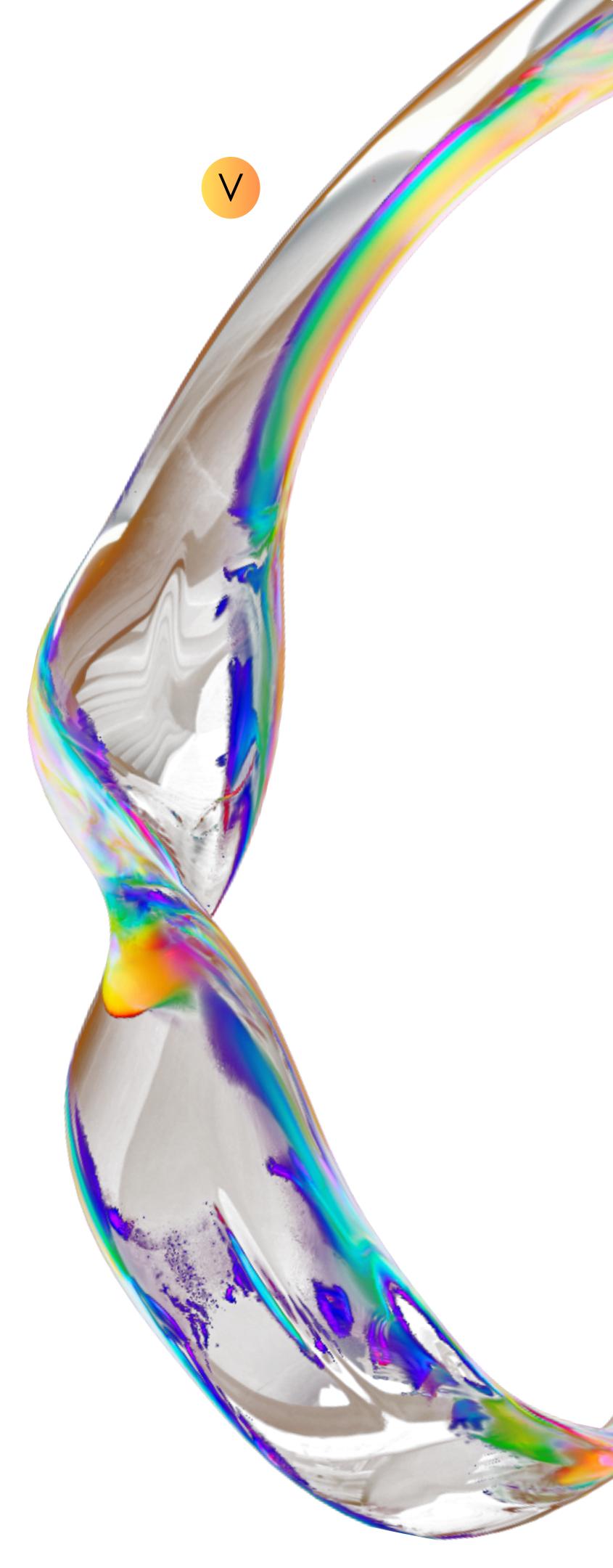
Qual è stata la lezione più difficile?

- Affrontare e gestire il testing non funzionale, superando le complessità tecniche e metodologiche che legano il testing all'implementazione.



Qual è stato il tuo contributo?

- Ho dato un contributo significativo al progetto, curando sia i piccoli dettagli che gli aspetti di alto livello e supportando il team nel raggiungimento degli obiettivi prefissati.



# Varone Giuseppe



Cosa hai imparato?

- Ho sviluppato una maggiore capacità di collaborazione all'interno di un team.



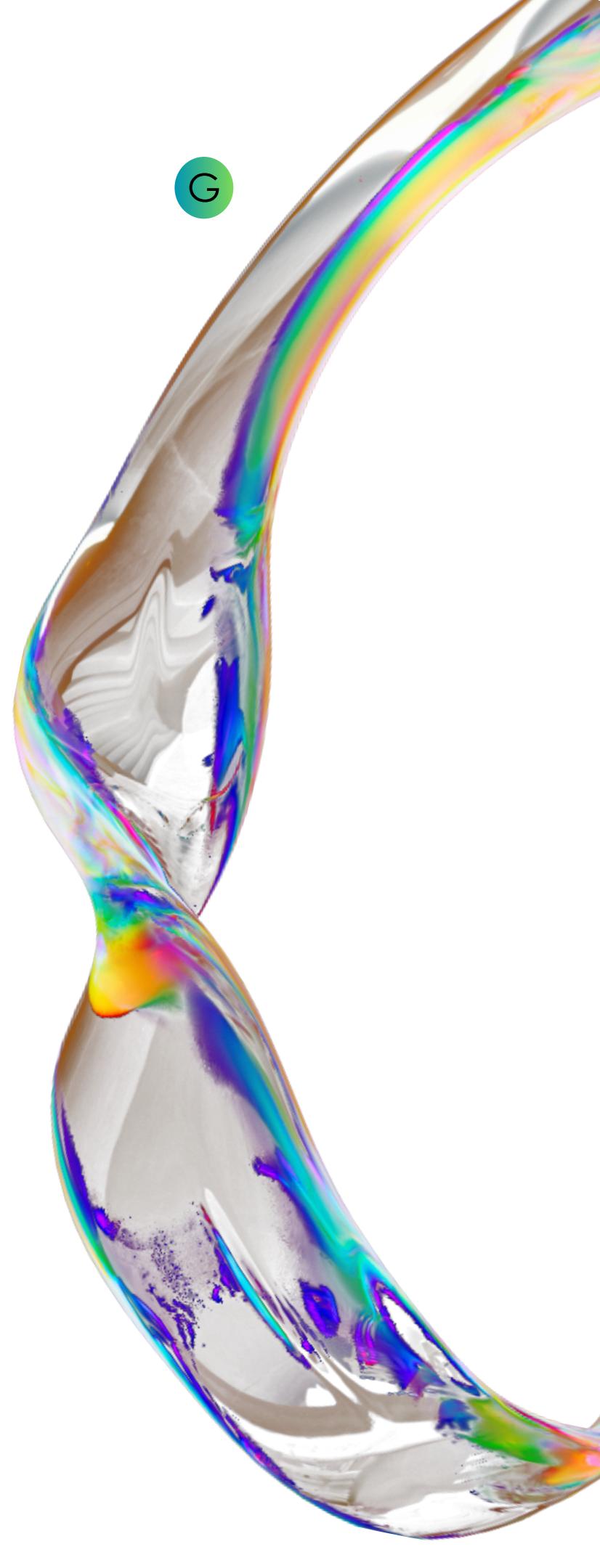
Qual è stata la lezione più difficile?

- Imparare a gestire le scadenze.



Qual è stato il tuo contributo?

- Ho dato un contributo significativo al progetto, curando sia i piccoli dettagli che gli aspetti di alto livello e supportando il team nel raggiungimento degli obiettivi prefissati.



# Team



Cosa è andato bene:

- Aver concluso il progetto con un buon risultato



Cosa è andato male:

- L'errata pianificazione di alcune scelte



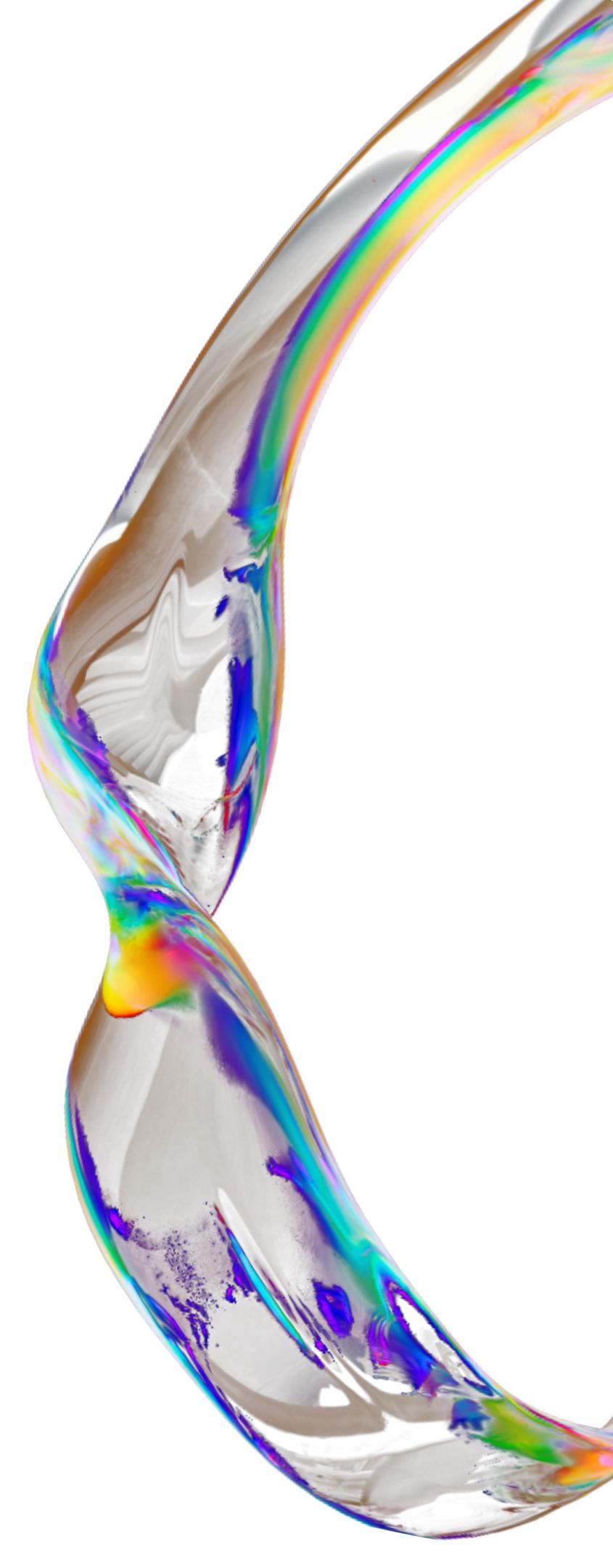
Cosa faresti diversamente:

- Una pianificazione più accurata di molte scelte avrebbe permesso di risparmiare tempo prezioso.



Quanto è buono il tuo lavoro:

- Molto buono paragonato al tempo e alle risorse impiegate.





# Domande

MedConnect - C05

# Grazie per l'attenzione

**Cusati Daniel**

(Mat. 05121 12422)

**Medica Vincenzo**

(Mat. 05121 16808)

**Varone Giuseppe**

(Mat. 05121 17831)

**Pre appello**

8 Gennaio 2025



**MedConnect**