



DOCUMENTAZIONE PROGETTO DI INGEGNERIA DEL SOFTWARE

ALESSANDRO DE CARLI – VR500321

MATTIA DANESE – VR502999

MARCO MAZZUCCATO – VR504966

A.A. 2024 – 2025

Sommario

Introduzione al progetto	3
Casi d'uso per l'Admin.....	5
Aggiunta dei diabetologi	6
Eliminazione dei diabetologi.....	7
Casi d'uso per i Diabetologi	8
Aggiungi paziente.....	9
Aggiunta di terapie	10
Modifica delle terapie.....	12
Eliminazione delle terapie	13
Controllo situazione clinica dei pazienti	14
Inserimento dati del paziente	15
Modifica anagrafica del paziente	16
Chat con i pazienti	17
Gestione credenziali	17
Eliminazione dei pazienti.....	18
Casi d'uso per i Pazienti.....	19
Visualizza terapie	19
Aggiunta di livelli di insulina.....	21
Aggiunta di assunzioni farmaci	22
Gestione delle credenziali	23
Chat con medico.....	24
Diagrammi di attività	25
Autenticazione	25
Attività diabetologo.....	26
Attività paziente.....	27
Attività admin	28
Class diagram	29
Class Diagram: Models.....	29
Sviluppo: architettura del progetto	30
Note sul processo di sviluppo e tipologia di approccio	30
Tipologie di pattern utilizzati	30

Progettazione database.....	32
Attività di test e validazione	32
Ispezione del codice e della documentazione	32
Verifica da parte degli sviluppatori	32
Verifica tramite JUnit Test	33
Test TerapiaDAOImpl	33
Test InsulinaDAOImpl	36
Test dell’utente esterno	39

Introduzione al progetto

Il sistema si pone come obiettivo quello di creare una dashboard per pazienti e diabetologi in modo da mettere in comunicazione questi due sotto dal punto di vista medico.

In particolare, dalla dashboard del paziente è possibile verificare l'andamento dell'insulina da lui registrata (ultime dieci). Oltre a questo, sempre nella pagina principale, il paziente è in grado di vedere le terapie che il suo dottore gli ha assegnato.

Il paziente può navigare in altre posizioni nella sua dashboard in modo da poter inserire altri livelli di insulina, dati relativi all'assunzione di medicinali prescritti dal medico, oltre a che poter scambiare dei messaggi con il dottore che lo segue per un contatto più immediato. Il paziente è in grado di modificare la sua password di accesso tramite la sezione delle "Impostazioni".

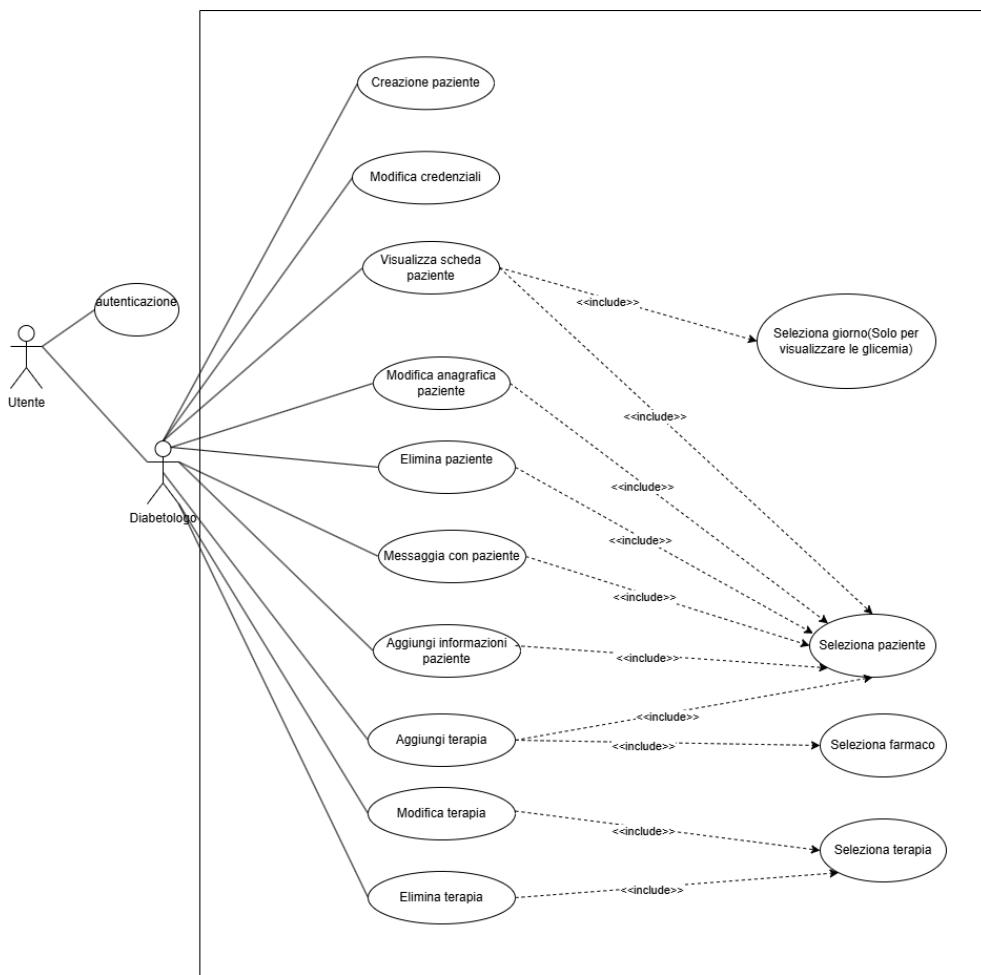
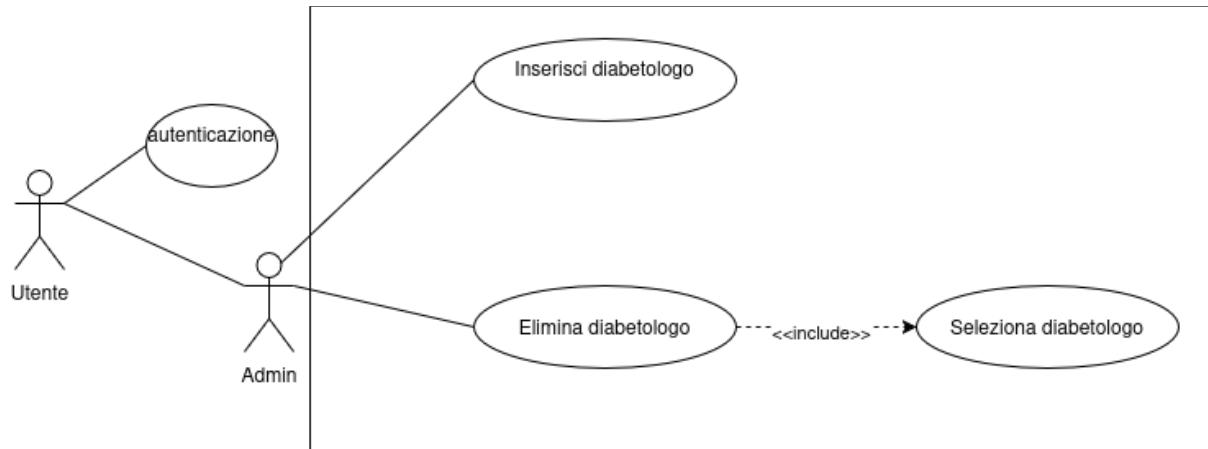
Il medico, nella pagina principale del suo portale, è in grado di aggiungere alla lista dei suoi pazienti uno nuovo. È, inoltre, in grado di controllare lo stato di insulina di ogni suo paziente, il tutto reso semplice da dei grafici, oltre che a visualizzare in modo chiaro le informazioni sulla persona seguita. Può anche vedere, modificare, ed eliminare le terapie già associate ad un determinato paziente.

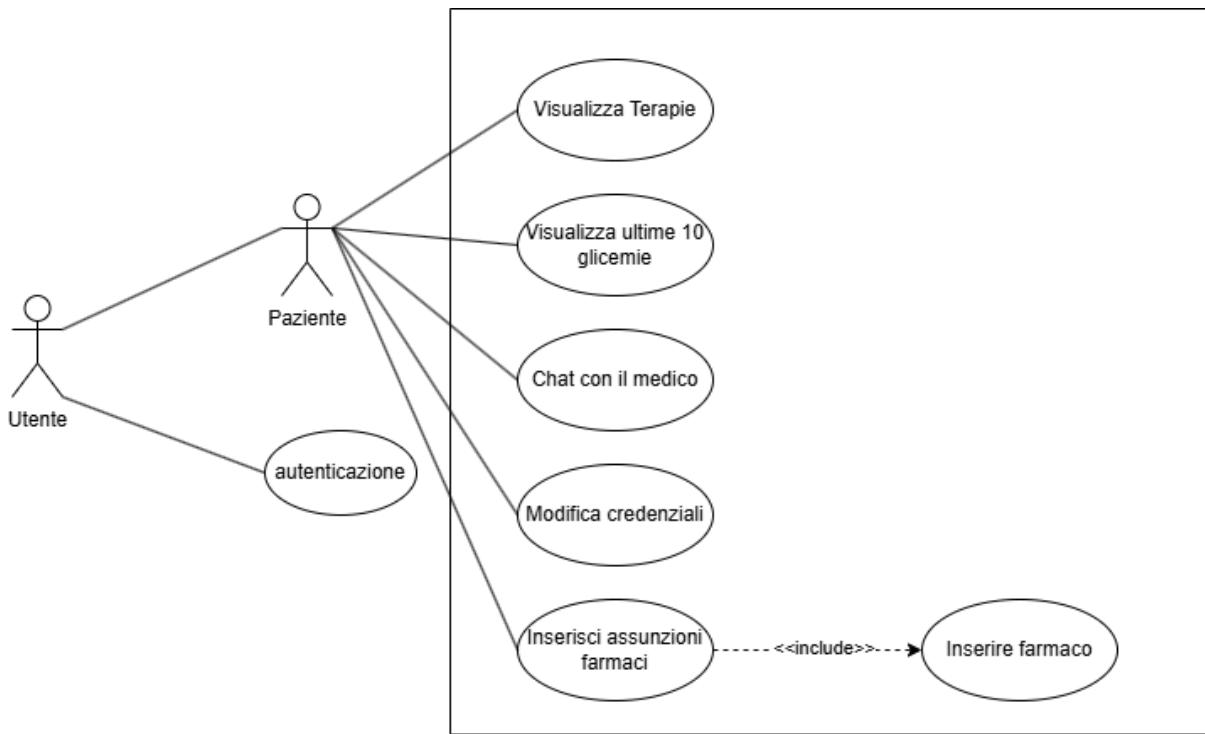
Come ultime funzioni è in grado di aggiungere terapie ad uno dei suoi pazienti, chattare con gli stessi e gestire le sue credenziali di accesso.

Oltre a questo, l'admin del sistema ha una sua dashboard di controllo da dove può creare nuovi diabetologi ed eliminarne esistenti.

Diagramma dei casi d'uso

Per motivi di spazio, abbiamo suddiviso l'intero diagramma dei casi d'uso in tre piccoli diagrammi.





Casi d'uso per l'Admin

Dopo un'autenticazione andata a buon fine, l'admin viene introdotto nella sua interfaccia che gli permette di aggiungere e rimuovere diabetologi.

Autenticazione Admin

Il sistema deve permettere all'admin di autenticarsi per accedere alle funzionalità di gestione.

Attori: Admin

Precondizioni:

1. L'admin deve possedere credenziali valide
2. Il sistema deve essere operativo

Passi:

1. L'admin accede alla schermata di login
2. L'admin inserisce username e password
3. L'admin seleziona "Login"
4. Il sistema verifica le credenziali
5. Se le credenziali sono corrette, l'admin accede al sistema

Postcondizioni: L'admin è autenticato e può accedere alle funzionalità

Aggiunta dei diabetologi

Nella “Home” del admin vi è un form per aggiungere nuovi diabetologi al sistema.

Aggiunta Diabetologo

Il sistema deve permettere all'admin di aggiungere nuovi diabetologi al sistema, inserendo i dati anagrafici e generando le credenziali.

Attori: Admin

Precondizioni:

1. L'admin dev'essersi autenticato
2. Il diabetologo non deve essere già presente nel sistema

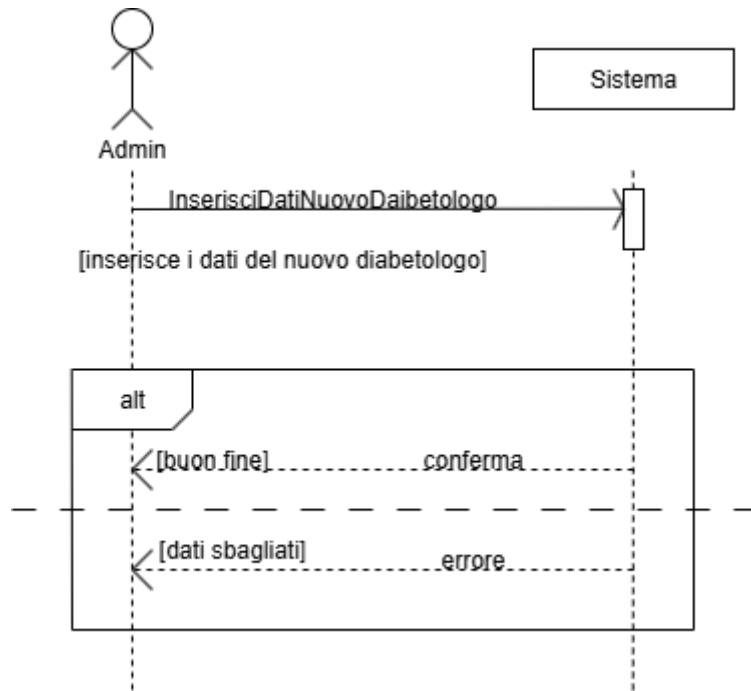
Passi:

1. L'admin accede al sistema al home page
2. L'admin inserisce:
 - Nome
 - Cognome
 - Codice fiscale
 - Email
 - Sesso
3. L'admin seleziona "Aggiungi nuovo diabetologo"
4. Il sistema verifica la validità dei dati
5. Se i dati sono corretti, il diabetologo viene inserito, e vengono resettati i campi

Scenario alternativo: Dati errati

1. Se i dati sono incorretti o incompleti il sistema mostra un messaggio di errore
2. L'Admin può correggere i dati e ripetere l'operazione

Postcondizioni: Il diabetologo è registrato e può accedere al sistema



Eliminazione dei diabetologi

Nella pagina “Diabetologi” nell’interfaccia dell’admin vi è una lista con tutti i dottori salvati nel sistema. L’admin può rimuoverne uno premendo sul pulsante “Elimina” nella riga corrispondente.

Rimozione Diabetologo

Il sistema deve permettere all’admin di rimuovere diabetologi esistenti dal sistema.

Attori: Admin

Precondizioni:

1. L’admin dev’essersi autenticato
2. Il diabetologo deve esistere nel sistema

Passi:

1. L’admin accede al sistema
2. L’admin seleziona “Diabetologi”
3. L’admin seleziona “Rimuovi diabetologo” nella riga del diabetologo che vuole rimuovere
4. Il sistema chiede la conferma dell’operazione
5. L’admin conferma l’operazione
6. Il Diabetologo viene rimosso dal database

Postcondizioni: Il diabetologo è rimosso dal sistema

Casi d'uso per i Diabetologi

Dopo un'opportuna autenticazione, il medico viene introdotto ad una interfaccia principale che gli permette un nuovo inserimento di un nuovo paziente nella sua lista.

Autenticazione Diabetologo

Il sistema deve permettere al diabetologo di autenticarsi per accedere alle funzionalità di gestione pazienti.

Attori: Diabetologo

Precondizioni:

1. Il diabetologo deve possedere credenziali valide
2. Il sistema deve essere operativo

Passi:

1. Il diabetologo accede alla schermata di login
2. Il diabetologo inserisce username e password
3. Il diabetologo seleziona "Login"
4. Il sistema verifica le credenziali
5. Se le credenziali sono corrette, il diabetologo accede al sistema

Postcondizioni: Il diabetologo è autenticato e può accedere alle funzionalità

Aggiungi paziente

Il diabetologo ha l'opportunità di creare un nuovo paziente, inserendone i dati basici (quali nome, cognome, codice fiscale, data di nascita, sesso) per poi seguire clinicamente quella nuova entità.

Aggiungi Paziente

Il diabetologo ha l'opportunità di creare un nuovo paziente, inserendone i dati basici per poi seguire direttamente quella nuova entità.

Attori: Diabetologo

Precondizioni:

1. Il diabetologo dev'essersi autenticato
2. Il paziente non deve essere già presente nel sistema

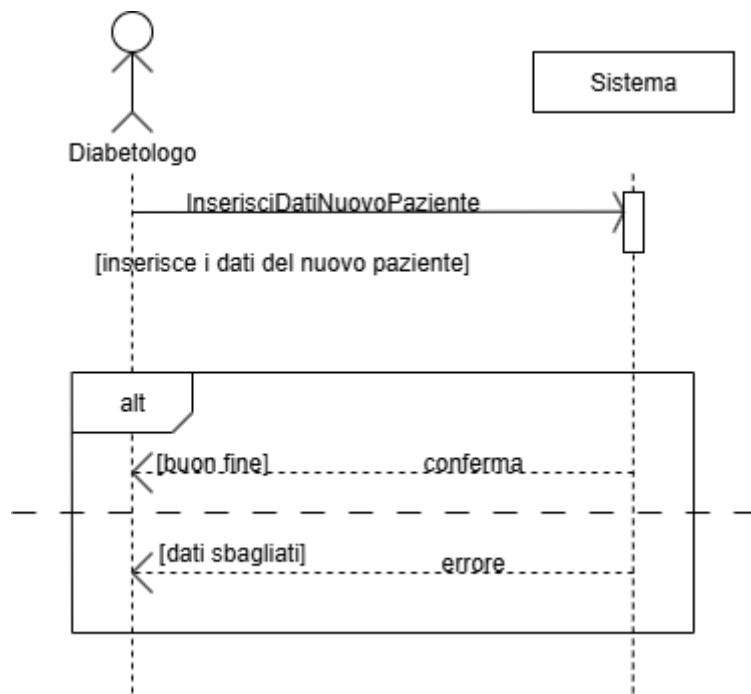
Passi:

1. Il diabetologo accede al sistema al home page
2. Il diabetologo inserisce i dati richiesti:
 - Nome
 - Cognome
 - Codice fiscale
 - Altri dati anagrafici
3. Il diabetologo preme "Crea Paziente"
4. Il sistema verifica la validità dei dati
5. Se i dati sono corretti, il paziente viene registrato e associato al diabetologo

Scenario Alternativo: Dati errati

1. Se i dati sono incompleti o incorretti il sistema mostra un messaggio d'errore
2. Il Diabetologo può correggere i dati e ripetere l'operazione

Postcondizioni: Il paziente è registrato nel sistema e associato al diabetologo



Aggiunta di terapie

Per aggiungere delle terapie, il diabetologo ha una sua interfaccia apposita in cui va ad inserire il codice fiscale del paziente, oltre che a scegliere tra una serie di medicine salvate nel database quella che serve insieme ad altri dati come la quantità e quante volte deve prenderla durante il giorno.

Aggiunta Terapie

Per aggiungere delle terapie, il diabetologo ha una sua interfaccia apposita in cui inserisce il codice fiscale del paziente e sceglie le medicine dal database.

Attori: Diabetologo

Precondizioni:

1. Il diabetologo dev'essersi autenticato
2. Il paziente deve esistere nel sistema
3. Il farmaco deve essere presente nel database

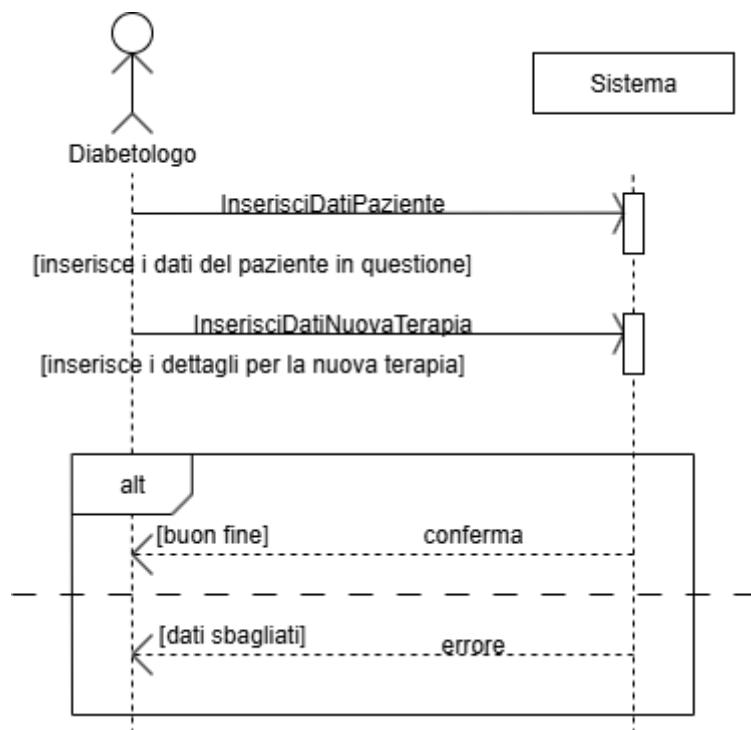
Passi:

1. Il diabetologo accede al sistema
2. Il diabetologo seleziona "Terapie"
3. Il diabetologo inserisce il codice fiscale del paziente
4. Il diabetologo seleziona il farmaco dal database
5. Il diabetologo specifica:
 - Quantità del farmaco
 - Frequenza giornaliera di assunzione
 - Dosaggio
 - Altre indicazioni
6. Il diabetologo conferma la prescrizione

Scenario Alternativo : Dati errati

1. I dati sono incompleti o incorretti, e il sistema mostra un messaggio di errore
2. Il Diabetologo può correggere i dati e ripetere l'operazione

Postcondizioni: La terapia è registrata e associata al paziente



Modifica delle terapie

Per modificare una terapia già esistente di un determinato paziente, il diabetologo deve andare nella scheda dello stesso in cui sono presenti tutti i dettagli. A quel punto, è possibile selezionare, tramite apposito bottone, la terapia da modificare.

Modifica Terapie

Per modificare una terapia già esistente di un determinato paziente, il diabetologo deve accedere alla scheda del paziente.

Attori: Diabetologo

Precondizioni:

1. Il diabetologo dev'essersi autenticato
2. Il paziente deve esistere nel sistema
3. La terapia da modificare deve esistere

Passi:

1. Il diabetologo accede al sistema
2. Il diabetologo seleziona il paziente dalla lista
3. Il diabetologo accede alla scheda terapie del paziente
4. Il diabetologo seleziona la terapia da modificare
5. Il diabetologo modifica i parametri desiderati:
 - Farmaco
 - Quantità
 - Frequenza
 - Dosaggio
 - Indicazioni
6. Il diabetologo conferma le modifiche

Scenario Alternativo : Dati errati

1. I dati che si tenta di modificare sono incompleti o incorretti, il sistema mostra un messaggio di errore
2. Il Diabetologo può correggere i dati e ripetere l'operazione

Postcondizioni: La terapia è aggiornata nel sistema

Eliminazione delle terapie

Per eliminare le terapie occorre andare nella stessa sezione visualizzare i dati clinici di un paziente e, nella tabella riassuntiva, è possibile eliminare una determinata terapia. Di conseguenza, verranno eliminati tutti i dati relativi a database.

Eliminazione Terapie

Per eliminare le terapie occorre andare nella sezione di visualizzazione dei dati clinici di un paziente.

Attori: Diabetologo

Precondizioni:

1. Il diabetologo dev'essersi autenticato
2. Il paziente deve esistere nel sistema
3. La terapia da eliminare deve esistere

Passi:

1. Il diabetologo accede al sistema
2. Il Diabetologo va alla pagina "Pazienti"
3. Il diabetologo seleziona il paziente dalla lista
4. Il diabetologo accede alla scheda terapie del paziente
5. Il diabetologo visualizza la tabella riassuntiva delle terapie
6. Il diabetologo seleziona "Elimina" sulla terapia desiderata
7. Il sistema richiede conferma
8. Il diabetologo conferma l'eliminazione

Postcondizioni: La terapia è rimossa dal sistema

Controllo situazione clinica dei pazienti

Per controllare l'andamento clinico dei pazienti, comprendendo dati su terapie, livelli di insulina, sintomi, comorbita', patologie pregresse e fattori di rischio, il diabetologo ha una sezione apposita in cui, una volta entrato, gli viene presentata una tabella con la lista di tutti i suoi pazienti. Dalla tabella clicca sul bottone "Apri" per aprire una scheda dedicata con tutti i dati specifici sul paziente selezionato.

Controllo Situazione Clinica Pazienti

Per controllare l'andamento clinico dei pazienti, il diabetologo ha una sezione apposita con la lista di tutti i suoi pazienti.

Attori: Diabetologo

Precondizioni: Il diabetologo dev'essersi autenticato

Passi:

1. Il diabetologo accede al sistema
2. Il diabetologo seleziona "Pazienti"
3. Il sistema mostra una tabella con la lista di tutti i pazienti
4. Il diabetologo seleziona "Apri" sul paziente di interesse
5. Il sistema mostra la scheda dedicata con tutti i dati specifici:
 - Dati sulle terapie
 - Sintomi segnalati
 - Dati glicemici
 - Comorbidità
 - Patologie pregresse
 - Fattori di rischio

Postcondizioni: Il diabetologo visualizza i dati clinici completi del paziente

Inserimento dati del paziente

Nella home page sotto il form dell'inserimento del paziente è possibile trovare un altro form con campi di testo in cui il diabetologo ha la possibilità di inserire i dati quali: "Fattori di rischio", "Comorbidità", "Patologie pregresse", scrivendo il codice fiscale del paziente del quale si vogliono aggiungere le informazioni

Inserimento Dati Informazione Paziente

Nel home page , il diabetologo può aggiornare i dati clinici inserendo il codice fiscale del paziente esistente

Attori: Diabetologo

Precondizioni:

1. Il diabetologo dev'essersi autenticato
2. Il paziente deve esistere nel sistema

Passi:

1. Il diabetologo accede al sistema nel home page
2. Il diabetologo inserisci il codice fiscale del paziente
3. Il diabetologo inserisce :
 - Fattori di rischio
 - Comorbidità
 - Patologie pregresse
4. Il diabetologo conferma l'inserimento

Scenario Alternativo: Dati errati

1. Il diabetologo inserisce il codice fiscale di un paziente inesistente o incorretto , oppure dati incorretti
2. Il Diabetologo può correggere i dati e ripetere l'operazione

Postcondizioni: I dati clinici del paziente sono inseriti

Modifica anagrafica del paziente

Il diabetologo può modificare i dati anagrafici di ogni suo paziente.

Modifica Anagrafica Paziente

Il diabetologo può modificare i dati anagrafici di ogni suo paziente.

Attori: Diabetologo

Precondizioni:

1. Il diabetologo dev'essersi autenticato
2. Il paziente deve esistere nel sistema

Passi:

1. Il diabetologo accede al sistema
2. Il diabetologo va alla pagina "Pazienti"
3. Il diabetologo seleziona il paziente dalla lista
4. Il diabetologo seleziona "Modifica anagrafica"
5. Il diabetologo modifica i dati anagrafici:
 - Nome
 - Cognome
 - Codice fiscale
 - Altri dati anagrafici
6. Il diabetologo conferma le modifiche

Scenario Alternativo: Dati errati

1. Se i dati che il diabetologo vuole modificare sono errati o incompleti , il sistema mostra un messaggio di errore
2. Il diabetologo può correggere i dati e ripetere l'operazione

Postcondizioni: I dati anagrafici del paziente sono aggiornati

Chat con i pazienti

Per inviare i messaggi con i pazienti, il diabetologo deve entrare nella sezione apposita, nella barra immediatamente a sinistra il dottore deve scegliere la chat in cui entrare per poi riuscire a mandare dei messaggi.

Chat con i Pazienti

Per inviare messaggi con i pazienti, il diabetologo deve entrare nella sezione apposita.

Attori: Diabetologo

Precondizioni:

1. Il diabetologo dev'essersi autenticato
2. Deve esistere almeno un paziente

Passi:

1. Il diabetologo accede al sistema
2. Il diabetologo seleziona "Chat con i pazienti"
3. Il diabetologo sceglie la chat del paziente dalla lista a sinistra
4. Il diabetologo compone il messaggio
5. Il diabetologo invia il messaggio

Postcondizioni: Il messaggio è inviato al paziente

Gestione credenziali

Il diabetologo può modificare la password di accesso nella pagina dedicata.

Gestione Credenziali

Il diabetologo può modificare la password di accesso nella pagina dedicata.

Attori: Diabetologo

Precondizioni: Il diabetologo dev'essersi autenticato

Passi:

1. Il diabetologo accede al sistema
2. Il diabetologo seleziona "Impostazioni"
3. Il diabetologo inserisce la nuova password
4. Il diabetologo conferma la nuova password
5. Il diabetologo conferma il cambio password

Scenario Alternativo:

1. Se la password nuova e la conferma della nuova password non coincidono , il sistema mostra un messaggio di errore
2. Il diabetologo può correggere i dati e ripetere le operazioni

Postcondizioni: La password del diabetologo è aggiornata

Eliminazione dei pazienti

Il diabetologo può rimuovere un paziente dalla sua lista di persone seguite andando nella pagina dei pazienti dal suo portale e cliccando sul bottone “Elimina” nella riga desiderata. A quel punto, tutti i dati relativi a quel paziente verranno eliminati anche a database.

Eliminazione Pazienti

Il diabetologo può rimuovere un paziente dalla sua lista di persone seguite.

Attori: Diabetologo

Precondizioni:

1. Il diabetologo dev'essersi autenticato
2. Il paziente deve esistere nel sistema

Passi:

1. Il diabetologo accede al sistema
2. Il diabetologo seleziona "Pazienti"
3. Il diabetologo seleziona "Elimina" sul paziente desiderato
4. Il sistema richiede conferma
5. Il diabetologo conferma l'eliminazione

Postcondizioni: Il paziente è rimosso dal sistema

Casi d'uso per i Pazienti

Dopo un'opportuna azione di login da parte dell'utente, il paziente viene introdotto nella sua interfaccia privata che gli permette vedere una sintesi dei suoi dati.

Autenticazione Paziente

Il sistema deve permettere al paziente di autenticarsi per accedere alla propria interfaccia privata.

Attori: Paziente

Precondizioni:

1. Il paziente deve possedere credenziali valide
2. Il sistema deve essere operativo

Passi:

1. Il paziente accede alla schermata di login
2. Il paziente inserisce username e password
3. Il paziente seleziona "Login"
4. Il sistema verifica le credenziali
5. Se le credenziali sono corrette, il paziente accede al sistema

Postcondizioni: Il paziente è autenticato e visualizza la dashboard con sintesi dei dati

Visualizza terapie

È possibile per il paziente visualizzare le terapie a lui assegnate: dalla semplice operazione di login, nella dashboard è in grado di visualizzare nel dettaglio i dati delle stesse inserite appositamente dal diabetologo che lo segue.

Visualizza Terapie

È possibile per il paziente visualizzare le terapie a lui assegnate nella dashboard.

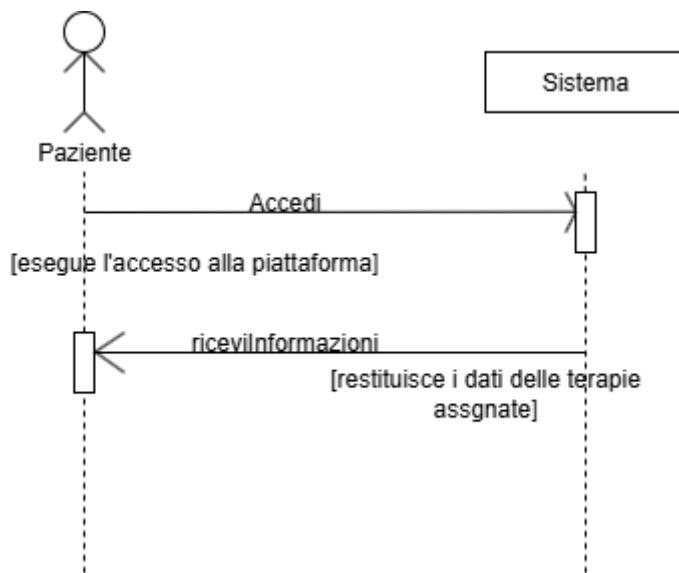
Attori: Paziente

Precondizioni: Il paziente dev'essersi autenticato

Passi:

1. Il paziente accede al sistema
2. Il paziente visualizza la dashboard principale
3. Il sistema mostra i dettagli delle terapie assegnate dal diabetologo
4. Il paziente può visualizzare:
 - Farmaci prescritti
 - Dosaggi
 - Periodo di assunzione del farmaco
 - Indicazioni specifiche

Postcondizioni: Il paziente visualizza le terapie assegnate



Aggiunta di livelli di insulina

Il paziente è in grado di inserire i livelli di insulina registrati nel corso della giornata tramite la sezione apposita in cui deve specificare il livello (in mg), il momento del giorno e l'ora in cui è stata registrata e gli eventuali sintomi.

Aggiunta Livelli di Insulina

Il paziente è in grado di inserire i livelli di insulina registrati nel corso della giornata.

Attori: Paziente

Precondizioni: Il paziente dev'essersi autenticato

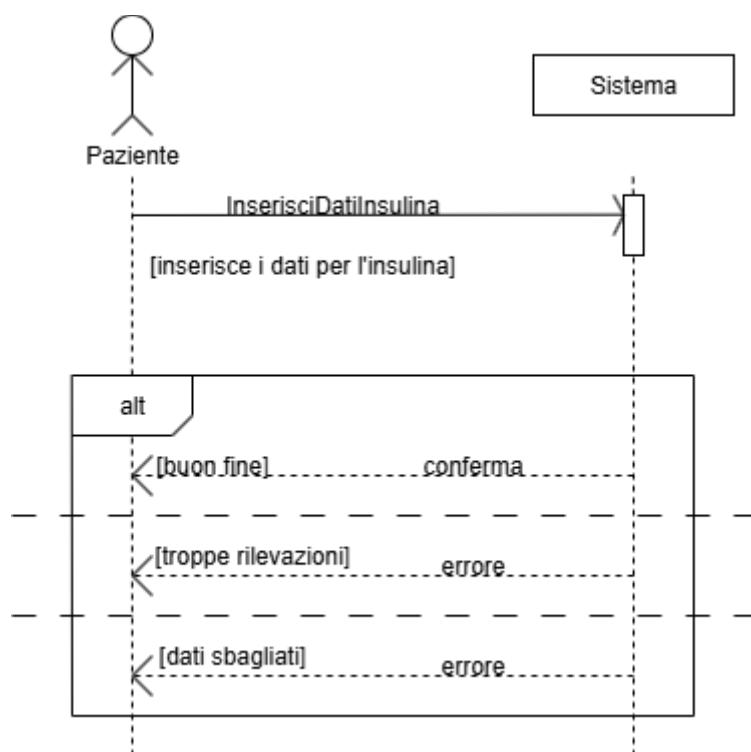
Passi:

1. Il paziente accede al sistema
2. Il paziente seleziona "Insulina"
3. Il paziente inserisce:
 - Livello di insulina (in mg)
 - Eventuali sintomi , se il paziente presenta sintomi
 - Momento del giorno
 - Ora di registrazione
4. Il paziente conferma l'inserimento

Scenario Alternativo: Dati errati

1. Se il paziente inserisce dati errati o incompleti , il sistema mostra un messaggio di errore
2. Il paziente può correggere i dati e ripetere le operazioni

Postcondizioni: Il livello di insulina è registrato nel sistema



Aggiunta di assunzioni farmaci

Il paziente è in grado di inserire i dati relativi all'assunzione dei farmaci prescritti nelle sue terapie nella sezione apposita del suo portale.

Aggiunta Assunzione Farmaci

Il paziente è in grado di inserire i dati relativi all'assunzione dei farmaci prescritti nelle sue terapie.

Attori: Paziente

Precondizioni:

1. Il paziente dev'essersi autenticato
2. Deve esistere una terapia prescritta

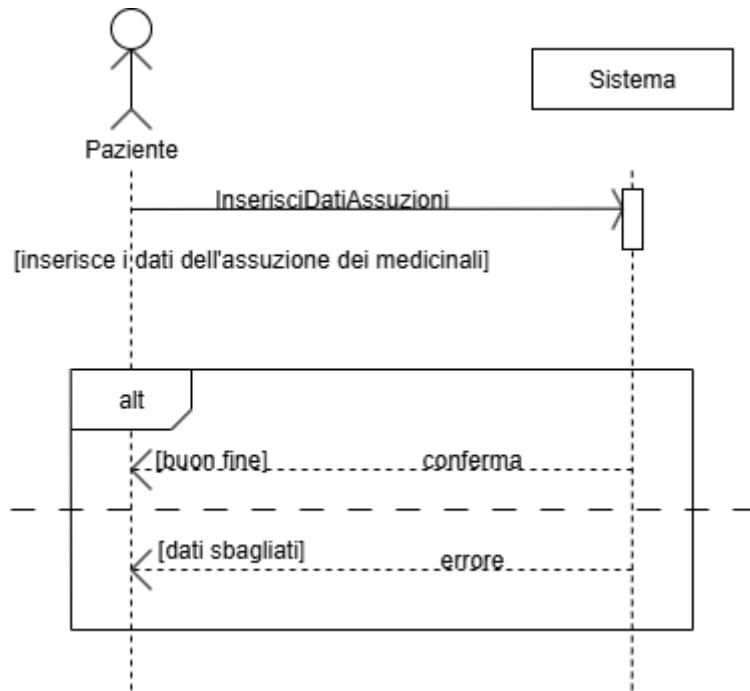
Passi:

1. Il paziente accede al sistema
2. Il paziente seleziona "Assunzioni"
3. Il paziente specifica:
 - Farmaco assunto
 - Dosaggio assunto
 - Data e ora dell'assunzione
4. Il paziente conferma l'inserimento

Scenario Alternativo: Dati errati

1. Se il paziente inserisce dati errati o incompleti , il sistema mostra un messaggio di errore
2. Il paziente può correggere i dati e ripetere le operazioni

Postcondizioni: L'assunzione del farmaco è registrata nel sistema



Gestione delle credenziali

Il diabetologo può modificare la password di accesso nella pagina dedicata.

Gestione Credenziali

Il paziente può modificare la password di accesso nella pagina dedicata.

Attori: Paziente

Precondizioni: Il paziente dev'essersi autenticato

Passi:

1. Il paziente accede al sistema
2. Il paziente seleziona "Impostazioni"
3. Il paziente inserisce la nuova password
4. Il paziente conferma la nuova password

Scenario Alternativo: Dati errati

1. Se il paziente inserisce la nuova password diversa dalla password di conferma, il sistema mostra un messaggio di errore
2. Il paziente può correggere uno dei due campi e ripetere le operazioni

Postcondizioni: La password del paziente è aggiornata

Chat con medico

Il paziente è in grado di scambiare messaggi e comunicare in modo semplice e veloce con il suo dottore tramite la chat diretta con quest'ultimo. È possibile inviare messaggi andando nella sezione apposita.

Chat con Medico

Il paziente è in grado di scambiare messaggi e comunicare in modo semplice e veloce con il suo diabetologo.

Attori: Paziente

Precondizioni:

1. Il paziente dev'essersi autenticato
2. Deve esistere un diabetologo di riferimento

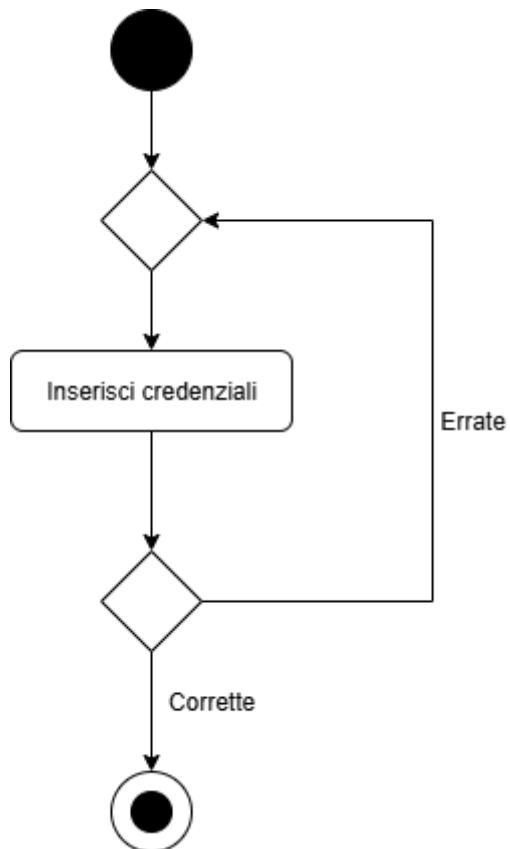
Passi:

1. Il paziente accede al sistema
2. Il paziente seleziona "Chat con il diabetologo"
3. Il paziente scrive il messaggio
4. Il paziente invia il messaggio

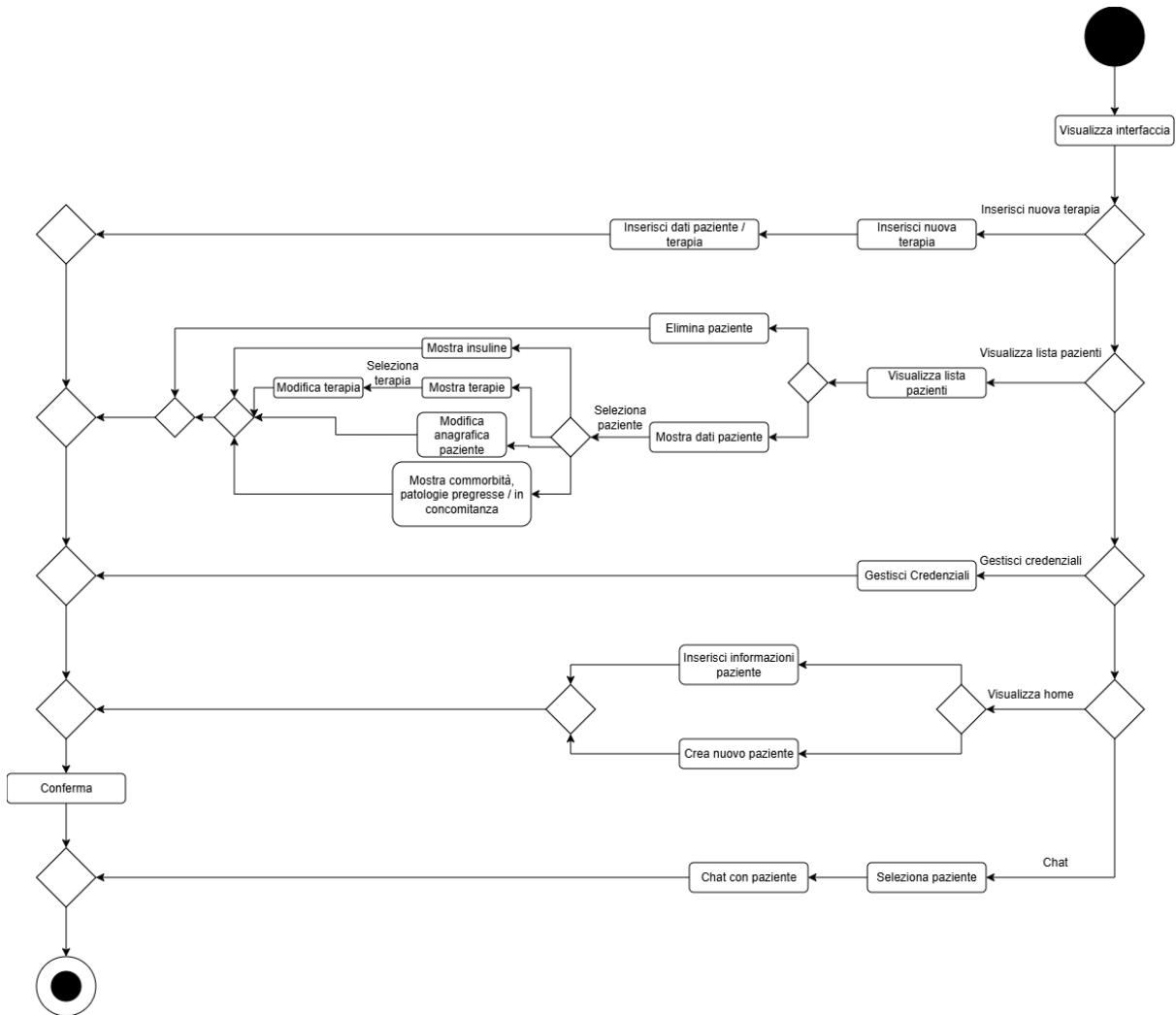
Postcondizioni: Il messaggio è inviato al diabetologo

Diagrammi di attività

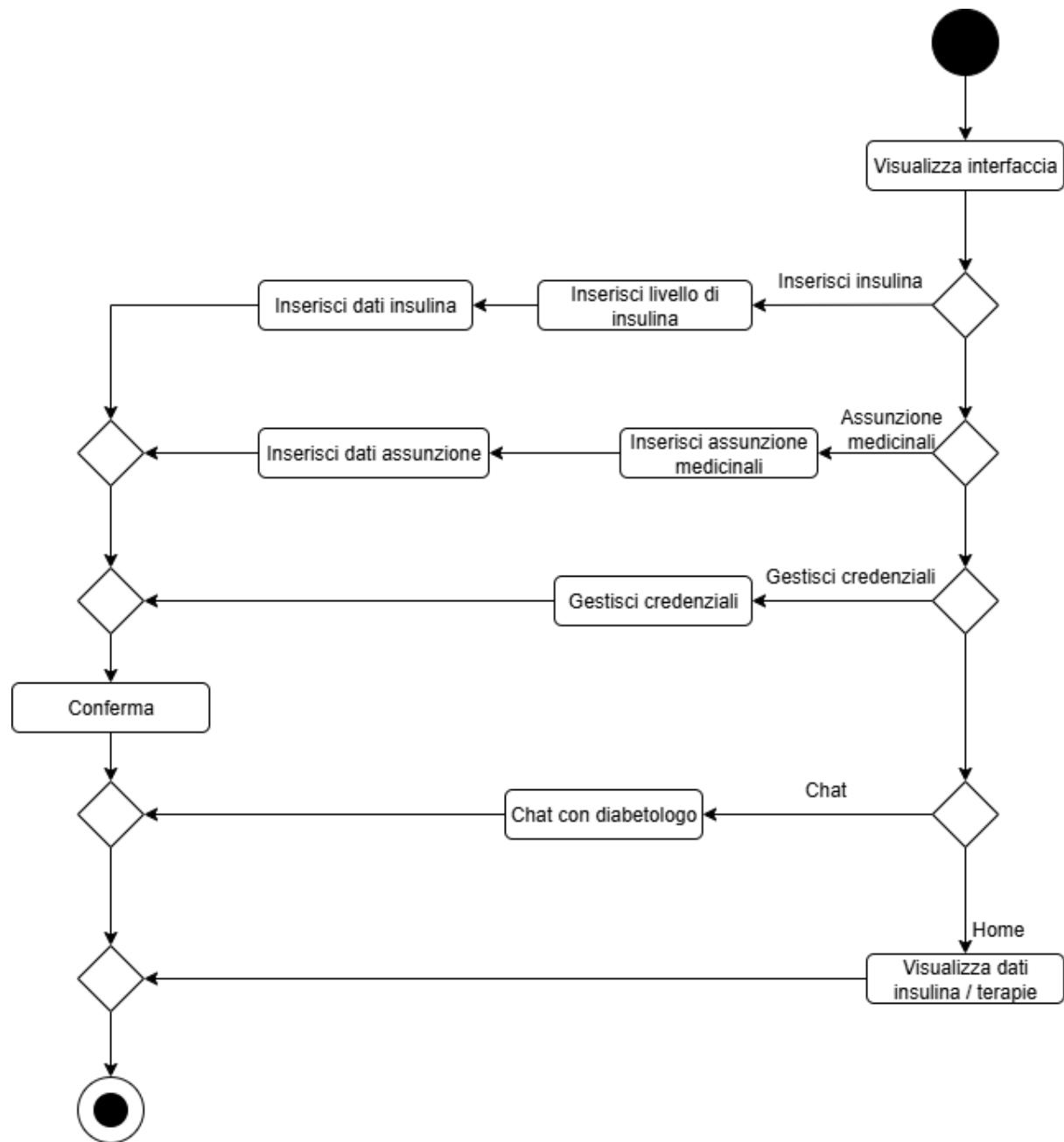
Autenticazione



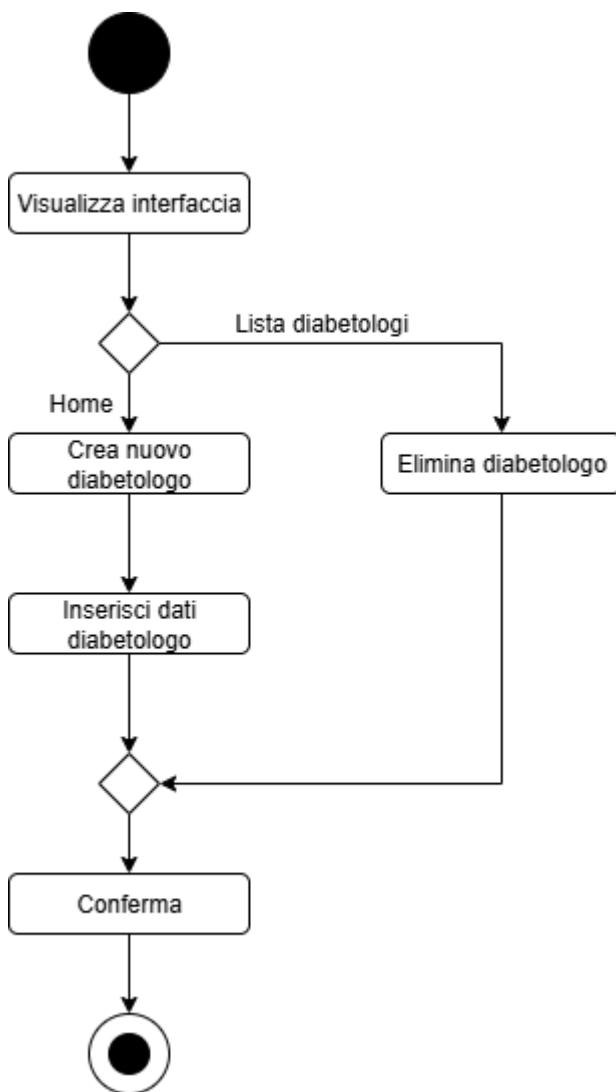
Attività diabetologo



Attività paziente



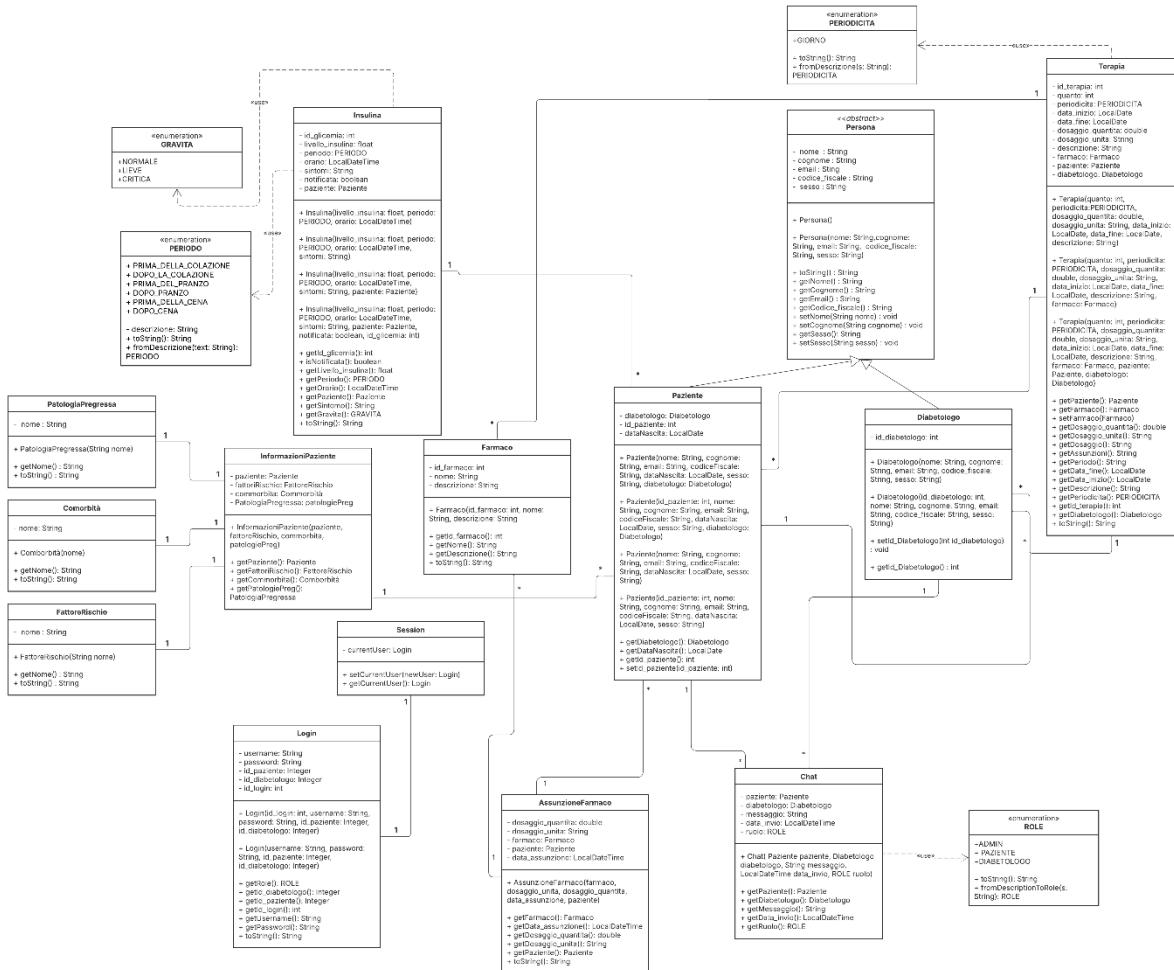
Attività admin



Class diagram

In queste immagini vengono rappresentate le varie classi utilizzate all'interno del programma, considerando anche le relazioni tra di loro.

Class Diagram: Models



Sviluppo: architettura del progetto

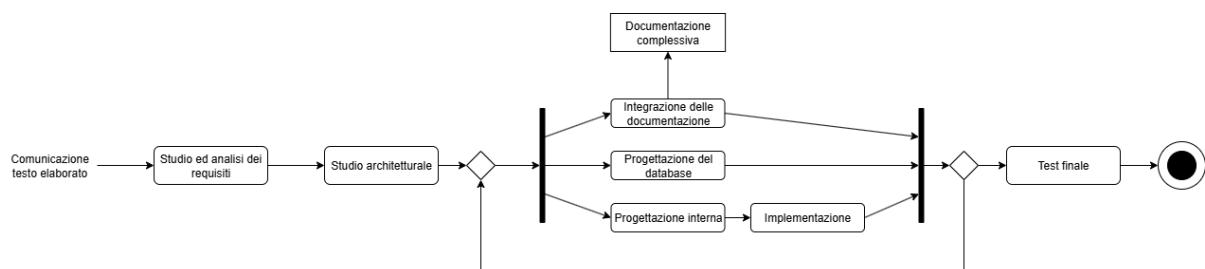
Note sul processo di sviluppo e tipologia di approccio

Il processo di sviluppo che abbiamo adottato è stato principalmente quello di tipo Agile ed Incrementale, in quanto era il modo più semplice e intuitivo per aggiungere nuove funzionalità al nostro software.

Si è cercato di rendere il più ottimale possibile anche la struttura del database, visti i continui cambiamenti dovute alle modifiche logiche e strutturali del software nella sua complessità.

Nonostante questo, si è cercato di rendere il più lineare lo sviluppo dello stesso, tenendo traccia dei dettagli che a mano a mano venivano aggiunti in modo da documentare nel particolare i vari processi di sviluppo.

Prima di iniziare, si è parlato della struttura del database con grande interesse nei dati da salvare, in quanto il resto del progetto si incentrava in questi. Subito dopo si è discusso dei vari use-Cases lato Diabetologo, Admin e Paziente, cercando di unire funzionalità, dati e trovando la soluzione più user-friendly possibile nello sviluppo della UI della nostra applicazione.

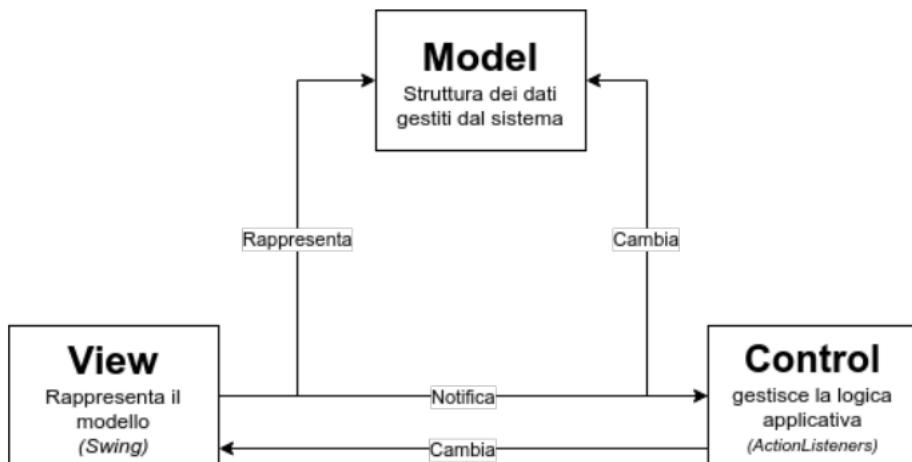


Tipologie di pattern utilizzati

In questo progetto sono stati utilizzati varie tipologie di pattern per rendere il più efficiente possibile l'utilizzo delle classi:

- **DAO:** è stato utilizzato questo genere di pattern in quanto si è rivelato il più versatile e scalabile per il nostro genere di applicazione. In particolare, è stato usato per le assunzioni dei farmaci, chat, diabetologi e pazienti, terapie, informazioni sul paziente, farmaci, insulina e login, ovvero la maggior parte dei models utilizzati all'interno del programma.

- **Singleton**: nella classe DbManager.java, è presente il metodo **`connect()`** che permette una singola istanziazione della classe da utilizzare nel corso di tutto l'utilizzo del software.
- **MVC**: è stato utilizzato il pattern **Modello-Vista-Controllore** in tutto il progetto. In particolare: il modello è quel componente che mantiene i dati e le informazioni memorizzate, oltre che a definire la struttura dell'informazione; la vista è quel componente dell'applicazione che definisce in maniera visiva i dati del sistema; il controllore è quel terzo componente che definisce la parte logica e che gestisce eventi dell'utente.



Progettazione database

Per tenere traccia dei dati e per una maggiore scalabilità dell'applicazione, è stato deciso di utilizzare un supporto database.

Il database è strutturato in modo da poter salvare i dati relativi a pazienti, diabetologi, dati per il login delle due entità e tutti quei dati relativi alla condizione clinica del paziente quali livello di insulina, assunzione di medicinali, terapie prescritte e andamento delle stesse, oltre che ad ulteriori informazioni cliniche necessarie per un corretto trattamento.

Il database inoltre fornisce il supporto per la comunicazione tra le due entità (paziente e diabetologo) funzione opportunamente implementata sotto forma di comunicazione di messaggistica all'interno dei corrispettivi pannelli di controllo.

Attività di test e validazione

Ispezione del codice e della documentazione

In questa fase che ha costituito una grande fetta di progettazione del software, si è rivisto e studiato a fondo il documento delle specifiche assegnate, per poi confrontarlo con l'effettiva struttura del progetto, oltre che a controllare a fondo la corretta implementazione dei pattern indicati precedentemente.

Verifica da parte degli sviluppatori

Per la parte di test lato sviluppatori sono stati i seguenti punti nella fase di test e validazione:

- **Controllo progressivo dei componenti:** man mano che nuovi pezzi venivano aggiunti al progetto, gli stessi venivano controllati a livello pratico per verificarne la correttezza funzionale.
- **Controllo del corretto funzionamento dei componenti visivi:** i componenti della *user-interface* sono stati controllati ogni volta venivano aggiunte delle funzionalità nuove, per accertarsi il corretto funzionamento del flusso di esecuzione.
- **Controllo “stress”:** in questa fase, sono stati presi in considerazione tutti quei componenti che potevano generare degli eventi e sono stati messi sotto “stress”: ogni pulsante cliccabile è stato cliccato e sono stati inseriti input completamente incoerenti con i dati che normalmente accetterebbero. Il tutto

per accertarsi che blocchino il flusso di esecuzione in caso di input mancanti o errati.

- **Controllo dei dati visibili al diabetologo:** è stato controllato che un determinato diabetologo possa visionare solamente i dati relativi ai suo*i* pazienti.
- **Controllo di visione dei dati condivisi:** è stato controllato che le terapie inserite per un determinato paziente dal diabetologo venga visto sia dallo stesso nella scheda dedicata ai dettagli del paziente, oltre che al paziente stesso nella sua dashboard.

Verifica tramite JUnit Test

In questa specifica sezione del percorso di validazione è stato usato un codice particolare con lo scopo di controllare la corretta esecuzione di alcune delle parti fondamentali del programma, il tutto automatizzato con JUnit.

Nel particolare, sono state testate le *implementazioni DAO* di *Insulina* e *Terapia*, due classi fondamentali per quanto riguarda i dati fondamentali su cui si basa tutto il resto nel software.

Test TerapiaDAOImpl

```
import com.dashapp.diabeticsystem.DAO.implementations.TerapiaDaolmpl;
import com.dashapp.diabeticsystem.enums.PERIODICITA;
import com.dashapp.diabeticsystem.models.*;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

import java.sql.SQLException;
import java.time.LocalDate;

import static org.junit.jupiter.api.Assertions.*;

public class TerapiaDaolmplTest {

    private TestDbManager db;
    private TerapiaDaolmpl terapiaDao;

    private Paziente paziente;
    private Diabetologo diabetologo;
    private Farmaco farmaco;

    @BeforeEach
    void setup() throws SQLException {
        db = new TestDbManager();
```

```

terapiaDao = new TerapiaDaolmpl(db);

// Inseriamo diabetologo
int idDiabetologo = db.insertAndGetGeneratedId(
    "INSERT INTO diabetologo(nome, cognome, codice_fiscale, email, sesso) VALUES (?,?,?,?,?,?)",
    "Mario", "Rossi", "RSSMRA80A01H501Z", "mario@example.com", "M"
);
diabetologo = new Diabetologo(idDiabetologo, "Mario", "Rossi", "RSSMRA80A01H501Z",
    "mario@example.com", "M");

// Inseriamo paziente
int idPaziente = db.insertAndGetGeneratedId(
    "INSERT INTO paziente(nome, cognome, codice_fiscale, data_nascita, email, sesso,
id_diabetologo) VALUES (?,?,?,?,?,?)",
    "Luca", "Bianchi", "BNCGLC90A01H501Z", java.sql.Date.valueOf("1990-01-01"),
    "luca@example.com", "M", idDiabetologo
);
paziente = new Paziente(idPaziente, "Luca", "Bianchi", "luca@example.com",
    "BNCGLC90A01H501Z" , LocalDate.of(1990,1,1), "M", diabetologo);

// Farmaco finto
farmaco = new Farmaco(1, "Metformina", "500mg");
}

@Test
void testInsertTherapy() {
    Terapia terapia = new Terapia(
        2,
        PERIODICITA.GIORNO,
        500,
        "mg",
        LocalDate.now(),
        LocalDate.now().plusDays(30),
        "Terapia iniziale",
        farmaco,
        diabetologo,
        paziente
    );

    boolean result = terapiaDao.insertTherapy(terapia);
    assertTrue(result);

    Integer count = db.selectQuery(
        "SELECT COUNT(*) FROM terapia WHERE id_paziente = ?",
        rs -> rs.next() ? rs.getInt(1) : 0,
        paziente.getId_paziente()
    );

    assertEquals(1, count);
}

```

```

}

@Test
void testGetTherapyByFarmacoldAndPatient() {
    // Inseriamo una terapia manualmente
    db.insertAndGetGeneratedId(
        "INSERT INTO
terapia(id_paziente,id_diabetologo,id_farmaco,dosaggio_quantità,dosaggio_unità,quanto,periodicità,da
ta_inizio_terapia,data_fine_terapia,descrizione) VALUES (?,?,?,?,?,?,?,?,?,?)",
        paziente.getId_paziente(), diabetologo.getId_diabetologo(), farmaco.getId_farmaco(),
        500, "mg", 2, "GIORNALIERA",
        java.sql.Date.valueOf(LocalDate.now()),
        java.sql.Date.valueOf(LocalDate.now().plusDays(30)),
        "Test terapia"
    );
}

Terapia terapia = terapiaDao.getTherapyByFarmacoldAndPatient(paziente, farmaco);

assertNotNull(terapia);
assertEquals(2, terapia.getQuanto());
assertEquals("mg", terapia.getDosaggio_unita());
}

@Test
void testRemoveTherapy() {
    int idTerapia = db.insertAndGetGeneratedId(
        "INSERT INTO
terapia(id_paziente,id_diabetologo,id_farmaco,dosaggio_quantità,dosaggio_unità,quanto,periodicità,da
ta_inizio_terapia,data_fine_terapia,descrizione) VALUES (?,?,?,?,?,?,?,?,?,?)",
        paziente.getId_paziente(), diabetologo.getId_diabetologo(), farmaco.getId_farmaco(),
        500, "mg", 2, "GIORNALIERA",
        java.sql.Date.valueOf(LocalDate.now()),
        java.sql.Date.valueOf(LocalDate.now().plusDays(30)),
        "Test terapia"
    );

    boolean removed = terapiaDao.removeTherapy(idTerapia);
    assertTrue(removed);

    Integer count = db.selectQuery(
        "SELECT COUNT(*) FROM terapia WHERE id_terapia = ?",
        rs -> rs.next() ? rs.getInt(1) : 0,
        idTerapia
    );

    assertEquals(0, count);
}
}

```

Test InsulinaDAOImpl

```
import com.dashapp.diabeticsystem.DAO.implementations.InsulinaDaoImpl;
import com.dashapp.diabeticsystem.enums.PERIODO;
import com.dashapp.diabeticsystem.models.Diabetologo;
import com.dashapp.diabeticsystem.models.Insulina;
import com.dashapp.diabeticsystem.models.Paziente;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

import java.sql.SQLException;
import java.time.LocalDate;
import java.time.LocalDateTime;

import static org.junit.jupiter.api.Assertions.*;

public class InsulinaDaoImplTest {

    private TestDbManager testDb;
    private InsulinaDaoImpl dao;
    private Paziente paziente;

    @BeforeEach
    void setup() throws SQLException {
        testDb = new TestDbManager();
        dao = new InsulinaDaoImpl(testDb);

        // Inserisco un diabetologo (FK)
        testDb.updateQuery("""
            INSERT INTO diabetologo(nome, cognome, codice_fiscale, email, sesso)
            VALUES ('Mario', 'Rossi', 'RSSMRA80A01H501Z', 'mario.rossi@example.com', 'M')
        """);

        // Inserisco un paziente (FK)
        testDb.updateQuery("""
            INSERT INTO paziente(nome, cognome, codice_fiscale, data_nascita, email, sesso,
            id_diabetologo)
            VALUES ('Luca', 'Bianchi', 'BNCGLC90A01H501Z', '1990-01-01', 'luca@example.com', 'M', 1)
        """);

        Diabetologo diabetologo = new Diabetologo(1, "Mario", "Rossi", "RSSMRA80A01H501Z",
        "mario.rossi@example.com", "M");
        paziente = new Paziente(1, "Luca", "Bianchi", "luca@example.com", "BNCGLC90A01H501Z",
        LocalDate.parse("1990-01-01"), "M", diabetologo);
    }

    @Test
```

```

void testInsertInsulina() {
    Insulina i = new Insulina(
        120,
        PERIODO.PRIMA_DELLA_COLAZIONE,
        LocalDateTime.now(),
        "nessun sintomo",
        paziente
    );

    boolean ok = dao.insertInsulina(i);
    assertTrue(ok);

    Integer count = testDb.selectQuery(
        "SELECT COUNT(*) AS c FROM insulina",
        rs -> rs.next() ? rs.getInt("c") : 0
    );

    assertEquals(1, count);
}

@Test
void testGetInsulinaByDateAndByPatients() {
    // Inserisco una riga
    testDb.updateQuery("""
        INSERT INTO insulina(id_paziente, valore_glicemia, orario, periodo, sintomi)
        VALUES (1, 150, NOW(), 'prima della colazione', '')
    """);

    var list = dao.getInsulinaByDateAndByPatients(
        LocalDateTime.now().minusDays(1),
        LocalDateTime.now().plusDays(1),
        paziente
    );

    assertEquals(1, list.size());
    assertEquals(150, list.get(0).getLivello_insulina());
}

@Test
void testGetInsulina() {
    // Inserisco due righe
    testDb.updateQuery("""
        INSERT INTO insulina(id_paziente, valore_glicemia, orario, periodo, sintomi)
        VALUES (1, 100, NOW(), 'prima della colazione', '')
    """);

    testDb.updateQuery("""
        INSERT INTO insulina(id_paziente, valore_glicemia, orario, periodo, sintomi)
        VALUES (1, 130, NOW(), 'dopo pranzo', '')
    """);
}

```

```

""");

var list = dao.getInsulina(paziente, 10, 0);

assertEquals(2, list.size());
}

@Test
void testCountDailyInsulinaByPatient() {
    // Inserisco due misurazioni oggi
    testDb.updateQuery("""
        INSERT INTO insulina(id_paziente, valore_glicemia, orario, periodo, sintomi)
        VALUES (1, 110, NOW(), 'prima della colazione', '')
        """);

    testDb.updateQuery("""
        INSERT INTO insulina(id_paziente, valore_glicemia, orario, periodo, sintomi)
        VALUES (1, 140, NOW(), 'dopo pranzo', '')
        """);

    int count = dao.countDailyInsulinaByPatient(paziente);

    assertEquals(2, count);
}

@Test
void testCountDailyMomentOfDay() {
    // Inserisco una misurazione oggi PRIMA COLAZIONE
    testDb.updateQuery("""
        INSERT INTO insulina(id_paziente, valore_glicemia, orario, periodo, sintomi)
        VALUES (1, 110, NOW(), 'prima della colazione', '')
        """);

    int count = dao.coundDailyMomentOfDay(PERIODO.PRIMA_DELLA_COLAZIONE, paziente);

    assertEquals(1, count);
}

@Test
void testmarkAsNotified(){
    testDb.updateQuery("""
        INSERT INTO insulina(id_paziente, valore_glicemia, orario, periodo, sintomi)
        VALUES (1, 110, NOW(), 'prima della colazione', '')
        """);

    dao.markAsNotified(1);

    var list = dao.getInsulinaByDateAndByPatients(
        LocalDateTime.now().minusDays(1),

```

```

        LocalDateTime.now().plusDays(1),
        paziente
    );
    assertTrue(list.get(0).isNotificata());
}

@Test
void testgetNonNotifiedByDateAndPatient(){
    testDb.updateQuery("""
        INSERT INTO insulina(id_paziente, valore_glicemia, orario, periodo, sintomi)
        VALUES (1, 110, NOW(), 'prima della colazione', '')
    """);

    var list = dao.getNonNotifiedByDateAndPatient(
        LocalDateTime.now().minusDays(1),
        LocalDateTime.now().plusDays(1),
        paziente
    );

    assertEquals(1,list.size());
}
}

```

Test dell’utente esterno

Il software è stato testato da un serie di ***utenti esterni***, a cui è stato semplicemente detto lo scopo in generale del programma presentato. Si è cercato di non interferire in maniera eccessiva nell’esperienza di utilizzo, lasciando l’utente il più libero possibile.

Questo genere di test si è rivelato molto utile in quanto i “*tester*” hanno dato luce ad una serie di piccoli errori che per gli sviluppatori non esistevano, oltre che a dare dei consigli su nuove funzionalità e miglioramenti lato UI per una maggiore fruibilità del prodotto finale.