
DOCUMENTAZIONE PROGETTO INGEGNERIA DEL SOFTWARE

AUTORI

ANGELICO ENRICO
BAZZOTTI EDOARDO
CERULLO DAVIDE
PORTARO RICCARDO

Università degli studi di Verona

GIUGNO 2023

Indice

Requisiti ed interazioni utente-sistema	2
Specifiche	3
Casi d'uso relativi agli utenti	3
Home page	5
Dati pressione	5
Dati farmaci	6
Modifica profilo	7
Casi d'uso relativi ai medici	8
Home page	8
Ricerca paziente	8
Analisi dottore	9
Diagrammi attività	11
Diagramma attività paziente	13
Diagramma attività paziente	15
Sviluppo: progetto dell'architettura ed implementazione del sistema	17
Progettazione e pattern architetturali utilizzati	18
Implementazione e design pattern usati	20
Diagrammi di sequenza del software implementato	21
Attività di test e validazione	25
Codice Test	25
Funzionalità aggiuntive	26

Requisiti ed interazioni utente-sistema

Il sistema proposto implementa un prototipo di applicazione a supporto delle persone che soffrono di malattie ipertensive. Gli "attori" all'interno dell'applicazione sono i Pazienti(Patient) e i Dottori(Doctor) ed entrambi possono accedere al sistema tramite una schermata di login. I pazienti hanno la possibilità di registrarsi al sistema per poi loggarsi con le loro credenziali; I dottori invece sono registrati al sistema dagli amministratori con una password standard la quale potrà essere cambiata a seguito del primo login effettuato.

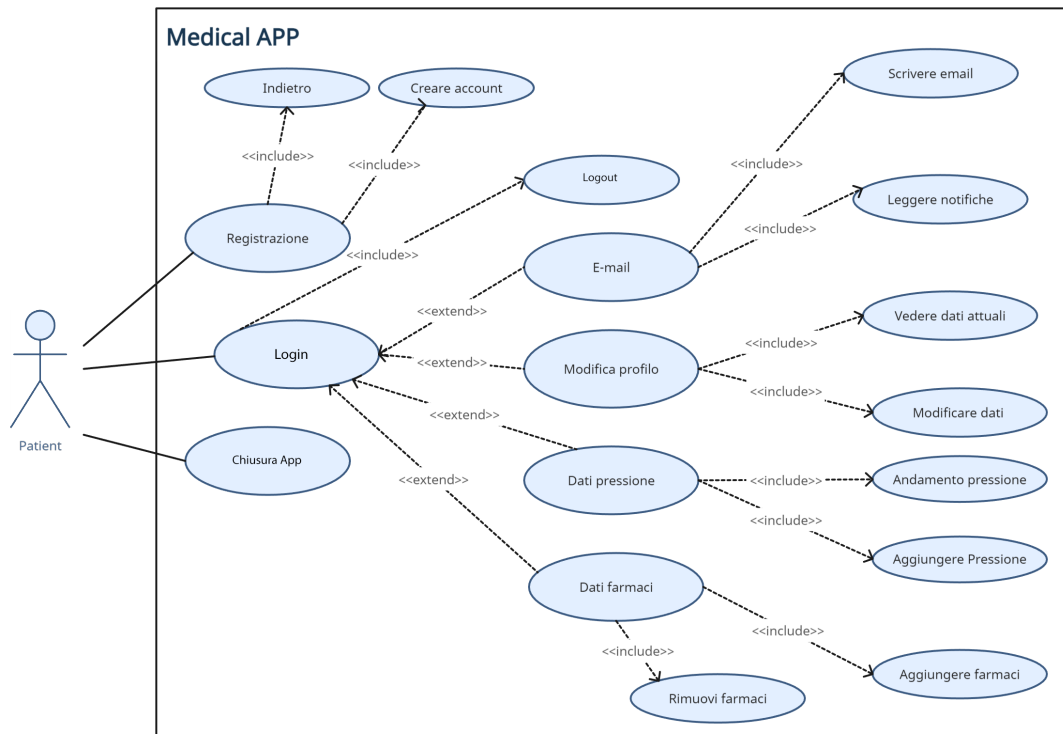


Figure 1: Casi d'uso paziente

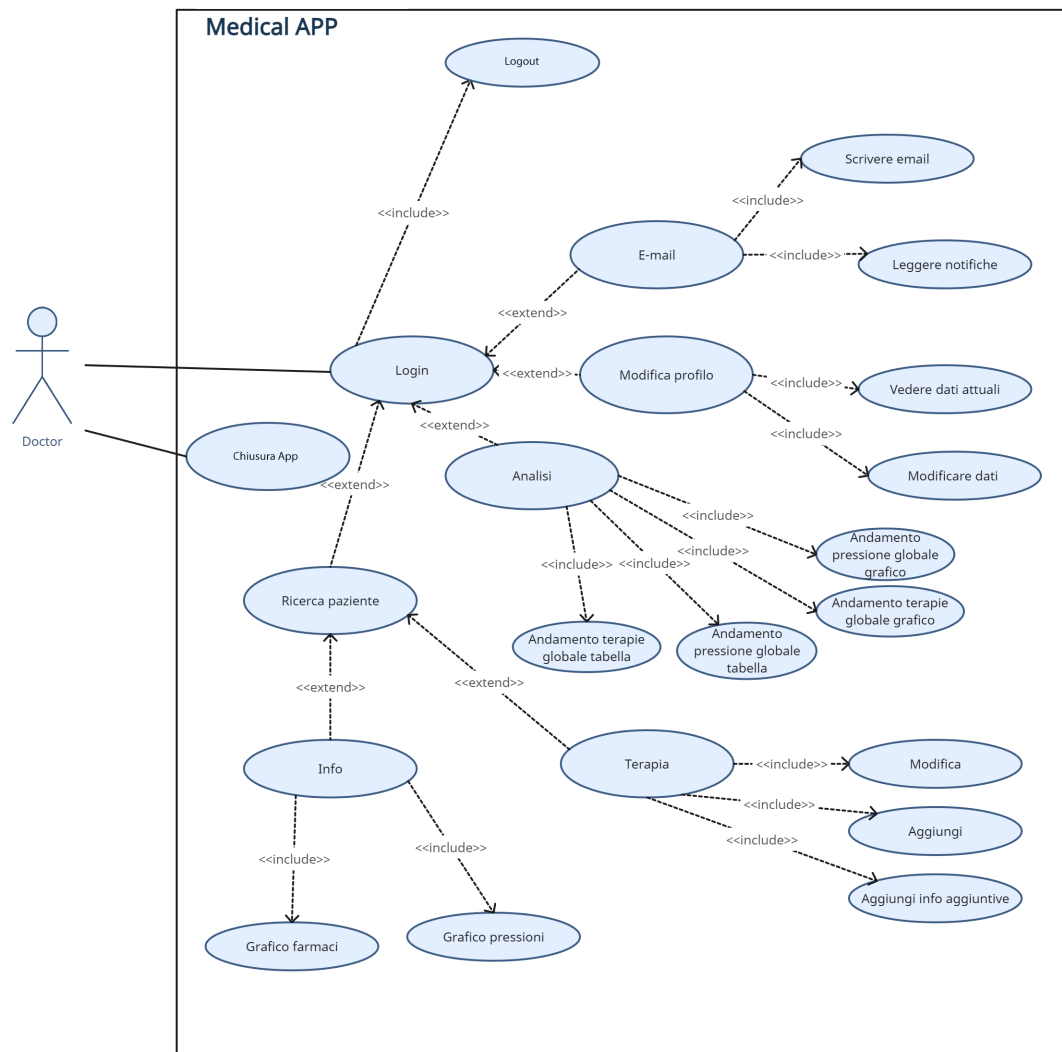


Figure 2: Casi d'uso dottore

Specifiche e casi d'uso

All'avvio dell'applicazione l'utente deve identificarsi in uno dei due attori (paziente o dottore), altrimenti sarà impossibilitato a svolgere una qualsiasi azione.

Casi d'uso relativi agli utenti

Dopo essersi identificato, il paziente dovrà scegliere se effettuare il login o registrarsi al sistema a seguito di un possibile incontro avvenuto con il suo nuovo dottore.

Fase di registrazione

Per poter effettuare il **primo accesso al sistema**, il paziente deve registrarsi inserendo i suoi dati anagrafici, nome utente, email, password e medico referente al quale è stato assegnato. Se il paziente dovesse lasciare dei campi vuoti, sbagliare ad inserire i dati o inserire dati anagrafici già presenti nel sistema, verrà visualizzato un messaggio di errore ed i campi errati verranno evidenziati. A seguito di una corretta registrazione, il paziente verrà riportato alla pagina di login e sarà pronto per accedere al sistema.

Attori: *Paziente*

Precondizioni: *Il paziente deve accedere alla schermata di registrazione*

Passi:

1. User accede al sistema
2. User si identifica in un paziente
3. Il paziente seleziona il pulsante di registrazione
4. Immette dati anagrafici, email, nome utente, password e medico referente
5. Conferma la registrazione

Sequenza alternativa: *Il paziente torna alla schermata di login annullando la registrazione*

Post-condizione: *Il paziente viene riportato alla schermata di login e potrà autenticarsi*

Di seguito il **Sequence diagram** per il caso della registrazione del paziente:

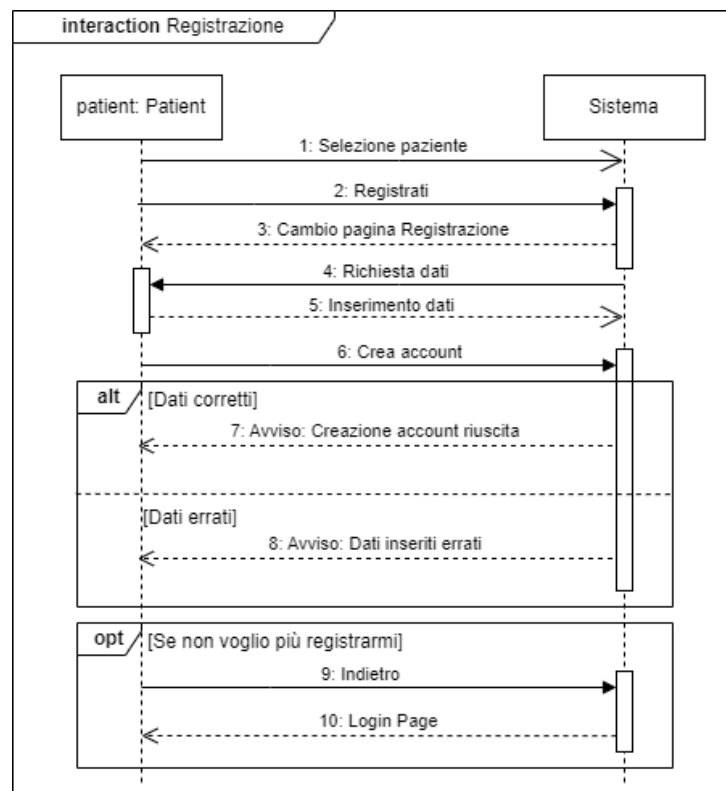


Figure 3: Registrazione paziente

Fase post-login

NOTA: A seguito del login, il paziente verrà portato nella **home page** dove a lato potrà visualizzare il suo nome con sotto un menù che offre la possibilità di navigazione tra le varie pagine : *Home page, Dati pressione, Dati farmaci, Modifica profilo*. Il paziente potrà accedere a questa sezione in ogni pagina dove è possibile la navigazione.

Home page

È la prima pagina visualizzata a seguito del login del paziente. In questa schermata viene visualizzato un prospetto generale che l'utente dovrebbe monitorare: l'ultima pressione registrata, lo stato attuale delle sue terapie, i farmaci che deve assumere (prescritti dal medico referente), il suo medico referente e la data di ultima rilevazione della pressione. L'unica azione che il paziente può compiere in questa schermata è la gestione delle proprie notifiche: il paziente potrà comunicare con il proprio medico referente per segnalare casi particolari relativi alle terapie, reazioni collaterali che i vari farmaci stanno causando, in modo da permettere al dottore di modificarle in base alle esigenze.

Attori: *Paziente*

Precondizioni: *Il paziente deve aver effettuato il login*

Passi:

1. *Il paziente si sposta sulla schermata relativa alle notifiche*
2. *Naviga tra le notifiche medico/sistema*

Sequenza alternativa: *Il paziente naviga tra le altre pagine*

Post-condizione: *Nessuna*

Dati pressione

Nella seguente schermata, il paziente avrà la possibilità di:

- **Registrare una nuova pressione:** dovrà inserire i dati relativi alla pressione *Sistolica, Diastolica, Sintomi rilevati*. Il paziente **potrà selezionare più sintomi** tra quelli in elenco o andare ad aggiungerne di nuovi cliccando sulla sezione altro (al paziente viene consigliato di effettuare una misurazione coerente della pressione tramite la media di tre registrazioni). Il paziente avrà la possibilità di **rimuovere una pressione precedentemente registrata** ma nel caso in cui vengano inseriti valori fuori norma (es. data rilevazione oltre la data odierna, pressione fuori dai "naturali" valori standardizzati) verrà notificato dell'errore.
- **Visualizzare la tabella delle precedenti registrazioni:** in questa tabella il paziente potrà monitorare le precedenti registrazioni con annessi i relativi stati individuati automaticamente dal sistema
- **Grafico andamento pressioni:** il paziente può scegliere se visualizzare un grafico lineare che gli mostra un andamento più generale sulle pressioni registrate.

Attori: *Paziente*

Precondizioni: *Il paziente deve aver effettuato il login*

Passi:

1. *Il paziente si sposta alla schermata "Dati pressione"*
2. *Inserisce i dati relativi alla pressione con sintomi annessi*
3. *Conferma o rimuove i dati relativi alla pressione inserita*

Sequenza alternativa: *Il paziente accede ad altre schermate disponibili nel menù a tendina*

Post-condizione: *Viene aggiornata la tabella e il grafico con la nuova pressione rilevata/rimossa*

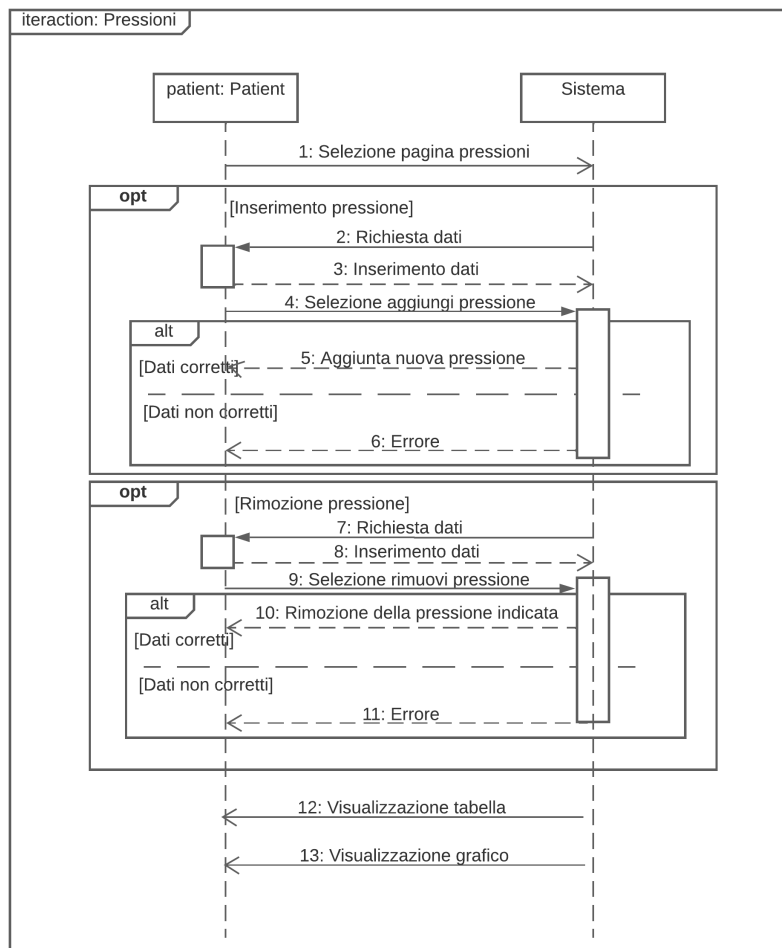


Figure 4: Inserimento rilevazione

Dati farmaci

Sezione in cui l'utente può registrare i farmaci presi, si ha la possibilità di:

- **Registrare un nuovo farmaco:** dovrà inserire i dati relativi al *farmaco assunto*, *quantità assunte*, *data di assunzione del farmaco*, *ora di assunzione*. Anche in questa fase il paziente avrà la possibilità di rimuovere un farmaco, nel caso in cui il farmaco che si vuole rimuovere non è presente nel sistema, l'azione non avrà alcun effetto.

- **Visualizzare l'elenco dei farmaci assunti precedentemente:** utile per auto-monitorare se si sta seguendo in maniera corretta la terapia, assumendo i farmaci nell'ora consigliata dal medico.

Attori: *Paziente*

Precondizioni: *Il paziente deve aver effettuato il login*

Passi:

1. *Il paziente si sposta alla schermata "Dati farmaci"*
2. *Inserisce i dati relativi ai farmaci assunti*
3. *Conferma o rimuove i dati relativi ai farmaci*

Sequenza alternativa: *Il paziente accede ad altre schermate nel menù a tendina*

Post-condizione: *Viene aggiornata la tabella con il farmaco assunto/rimosso*

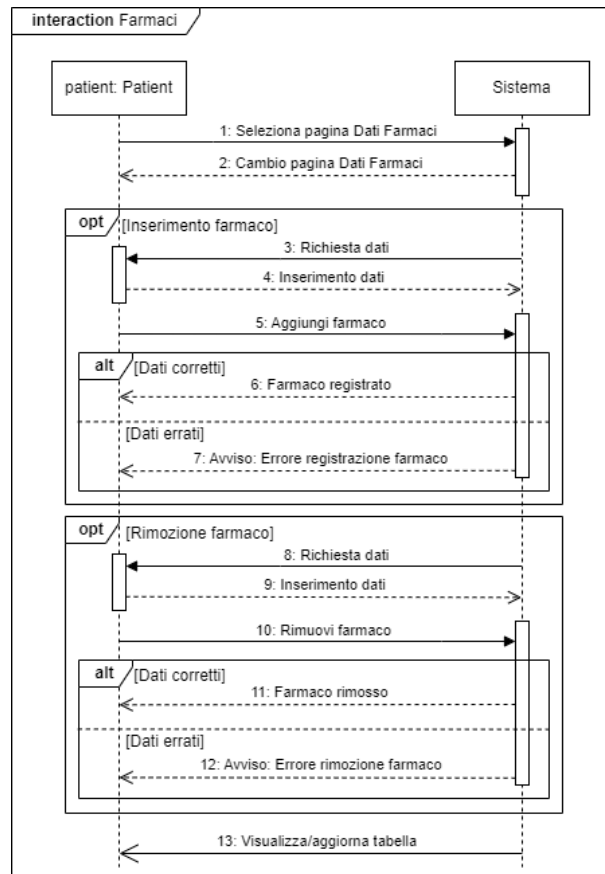


Figure 5: Inserimento farmaci

Modifica profilo

Sezione relativa alla modifica dei dati del paziente, i dati personali non sono mutabili (nome, cognome, codice fiscale, data di nascita, luogo di nascita, provincia di nascita), i dati relativi all'account invece possono essere modificati come lo *username*, *password*, *domicilio*, *email*, *numero di telefono*. Anche in questo caso i dati fuori norma verranno notificati.

Attori: *Paziente*

Precondizioni: *Il paziente deve aver effettuato il login*

Passi:

1. *Il paziente si sposta alla schermata "Modifica profilo"*
2. *Cambia i dati relativi all'account*
3. *Conferma i nuovi dati al sistema*

Sequenza alternativa: *Il paziente accede ad altre schermate nel menù a tendina*

Post-condizione: *Viene aggiornati i dati dell'account del paziente*

Casi d'uso relativi ai medici

Nella fase di login il medico avrà già un account prefabbricato dal sistema, le credenziali al primo accesso equivarranno al proprio nome utente e una password standard la quale potrà essere cambiata al primo login per una maggiore sicurezza dell'account.

Fase post-login

NOTA: A seguito del login, il medico verrà portato nella **home page** dove a lato potrà visualizzare il suo nome con sotto un menù che offre la possibilità di navigazione tra le varie pagine : *Home page, Ricerca paziente, Analisi, Modifica profilo*. Il medico potrà accedere a questa sezione in ogni pagina dove è possibile la navigazione.

Home page

Nella seguente pagina il medico può visualizzare la propria crescita lavorativa: il totale dei pazienti che affianca vedendo anche se sono in terapia e un grafico che mostra in base al periodo selezionato i nuovi pazienti che si sono registrati da lui. Potrà inoltre visualizzare la sezione delle notifiche per poter comunicare ai clienti le varie variazioni riguardanti la terapia da seguire.

Attori: *Medico*

Precondizioni: *Il medico deve aver effettuato il login*

Passi:

1. *Il medico si sposta alla schermata relativa alle notifiche*
2. *Naviga tra le notifiche paziente/sistema*

Sequenza alternativa: *Il medico accede ad altre schermate nel menù a tendina*

Post-condizione: *Nessuna*

Ricerca paziente

Sezione nella quale il medico può cercare tutte le informazioni relative ad un suo paziente, vedere un andamento delle pressioni, dei farmaci presi, assegnare una nuova terapia o modificare/rimuovere una già esistente.

Attori: *Medico*

Precondizioni: *Il medico deve aver effettuato il login*

Passi:

1. *Il medico si sposta alla schermata relativa "Ricerca paziente"*
2. *Il medico seleziona il paziente su cui avere più informazioni*
3. *Il medico visualizza la tabella delle pressioni*
4. *Assegna le terapie in base all'anamnesi delle pressioni e alle condizioni di salute del paziente*

Sequenza alternativa:

- *Il medico accede ad altre schermate del menù a tendina*
- *Il medico nella schermata "Ricerca paziente" chiede più informazioni e visualizza lo stato di salute del paziente, con un grafico che in base al periodo selezionato mostra i farmaci assunti e le pressioni registrate*

Post-condizione: *La terapia del paziente viene aggiornata/modificata/eliminata*

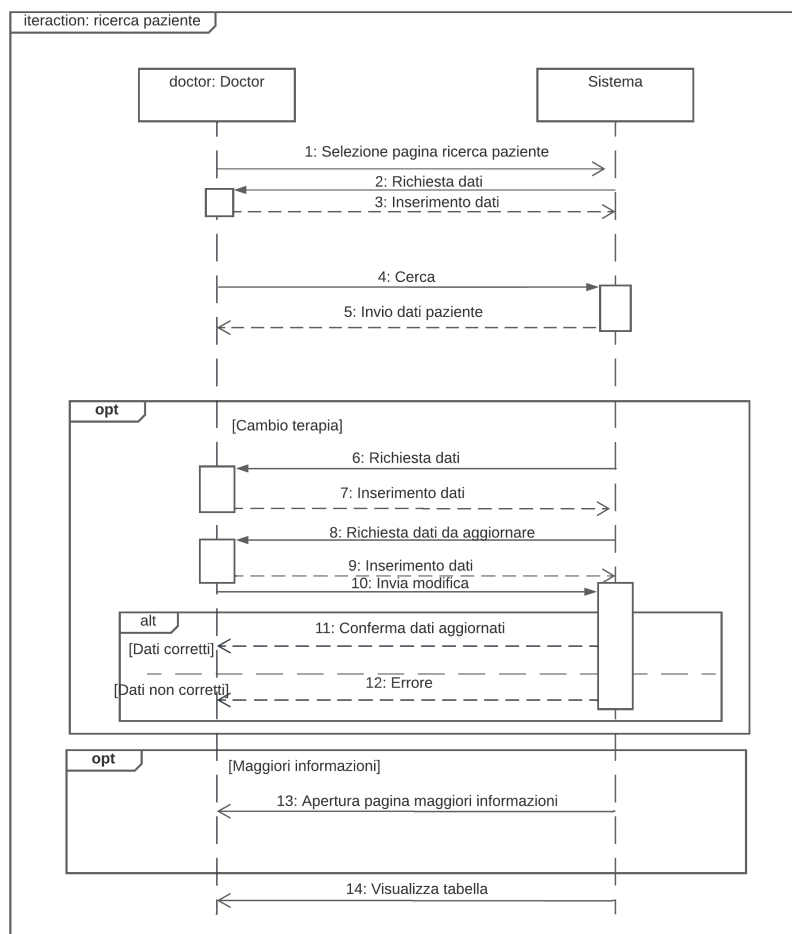


Figure 6: Cambio terapia

Analisi dottore

Schermata in cui il dottore può avere un prospetto generale dei suoi pazienti tramite grafici. Nella parte sottostante è presente un menù a tendina dove sarà possibile scegliere se analizzare le pressioni o i pazienti in terapia. Per le pressioni, in base al periodo selezionato, è possibile verificare quanti pazienti hanno registrato una determinata tipologia di pressione, utile per rendere meno pesante il lavoro del dottore nel caso in cui tutti i suoi pazienti fossero in condizioni di salute ottimale. Per le terapie invece, il medico potrà seguire quanti pazienti stanno seguendo una determinata terapia. Questo potrebbe risultare utile per capire quale terapia funziona al meglio su buona parte dei pazienti, in modo da indirizzare tutti sulle terapie più diffuse.

Attori: *Medico*

Precondizioni: *Il medico deve aver effettuato il login*

Passi:

1. *Il medico si sposta alla schermata relativa "Analisi Dottore"*
2. *Il medico decide se vedere i pazienti in base alle specifiche pressioni oppure secondo le terapie prescritte*
3. *Il medico visualizza la tabella delle pressioni o delle terapie globali*

Sequenza alternativa: *Il medico accede ad altre schermate del menù a tendina*

Post-condizione: *La terapia del paziente viene aggiornata/modificata/eliminata*

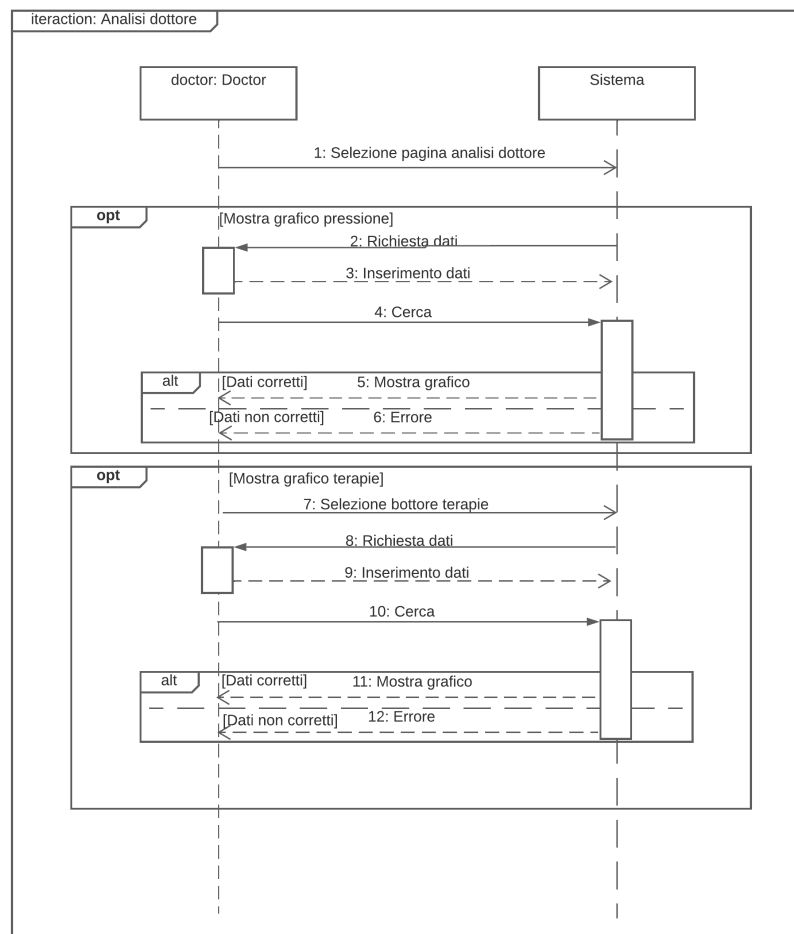


Figure 7: Analisi dottore

Diagrammi attività

Nota: Le "attività" indicate nei diagrammi sono considerate come eseguite una sola volta, anche se potrebbero essere ripetute. Questa semplificazione è stata adottata per favorire la chiarezza e la comprensione dei diagrammi. Inoltre per rendere più chiaro il flusso di attività, si è scelto di tornare sempre nella home page, anche se è possibile entrare in una delle altre schede dal menu di sinistra.

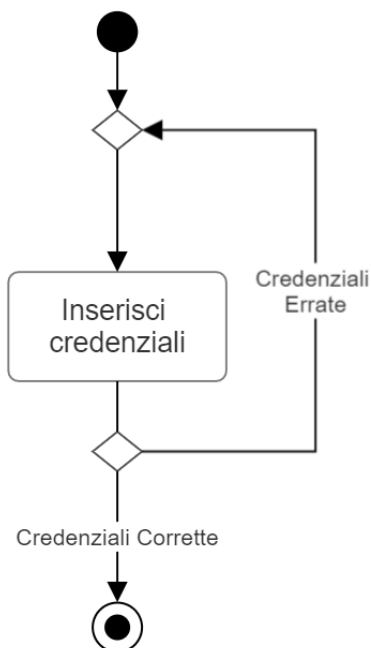


Figure 8: Effettua login

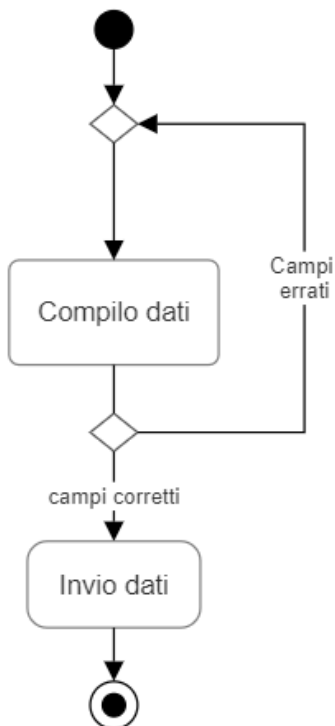


Figure 9: Effettua registrazione (solo paziente)

Di seguito due activity diagram che sono comuni ad entrambi gli utenti: **Modifica profilo** e **Chat del profilo**

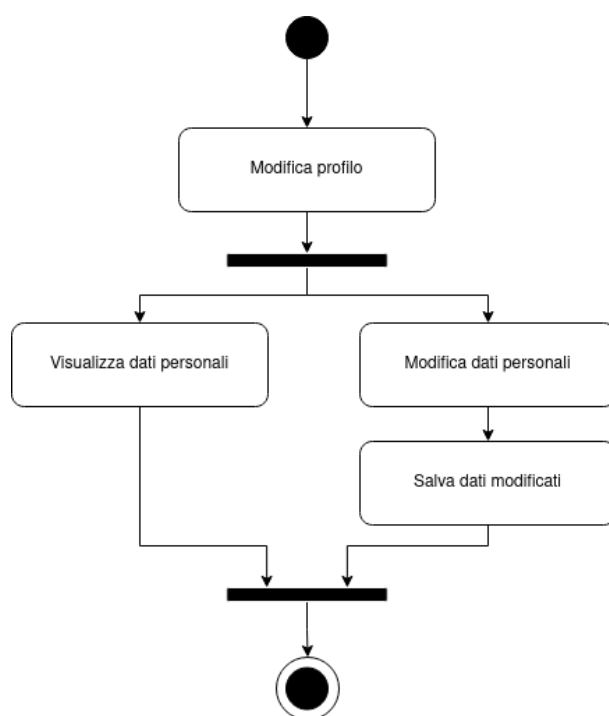


Figure 10: Modifica profilo Utente

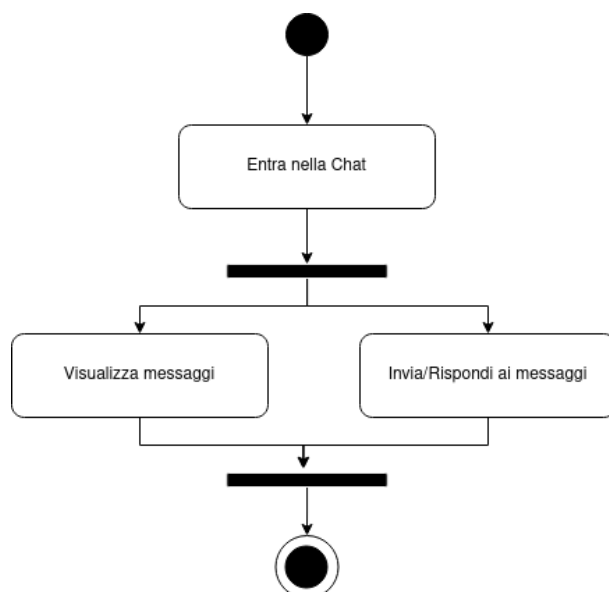


Figure 11: Chat Medico/Paziente/Sistema

Diagramma attività paziente

Di seguito riportato un diagramma generale del flusso di attività del paziente, con a seguire la vista in dettaglio di ogni attività.

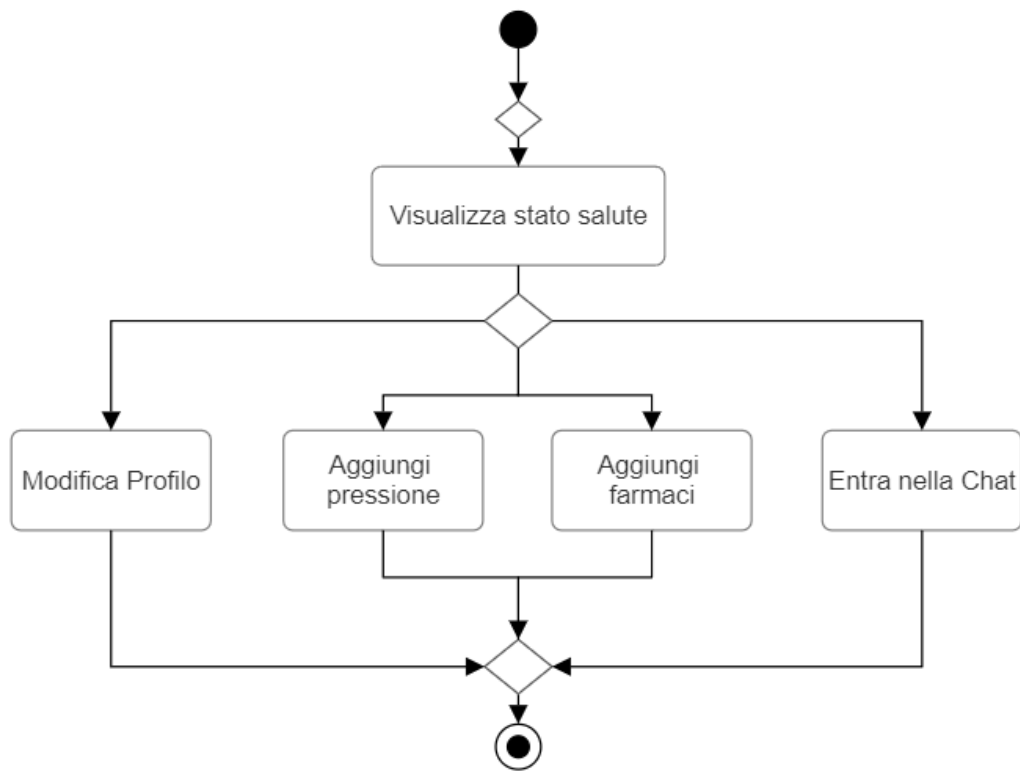


Figure 12: Activity diagram paziente

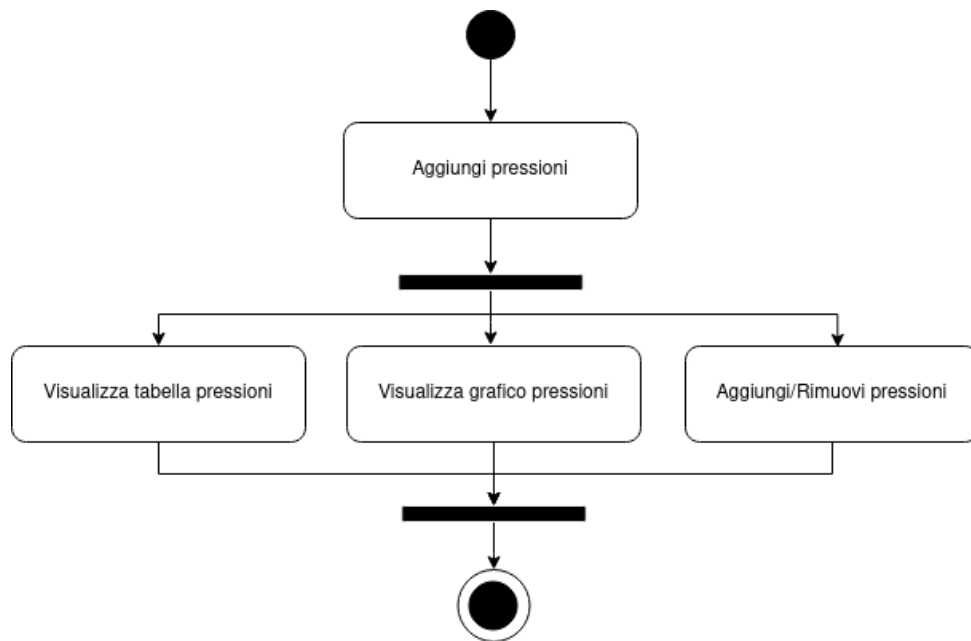


Figure 13: Aggiunta/Rimozione pressione paziente

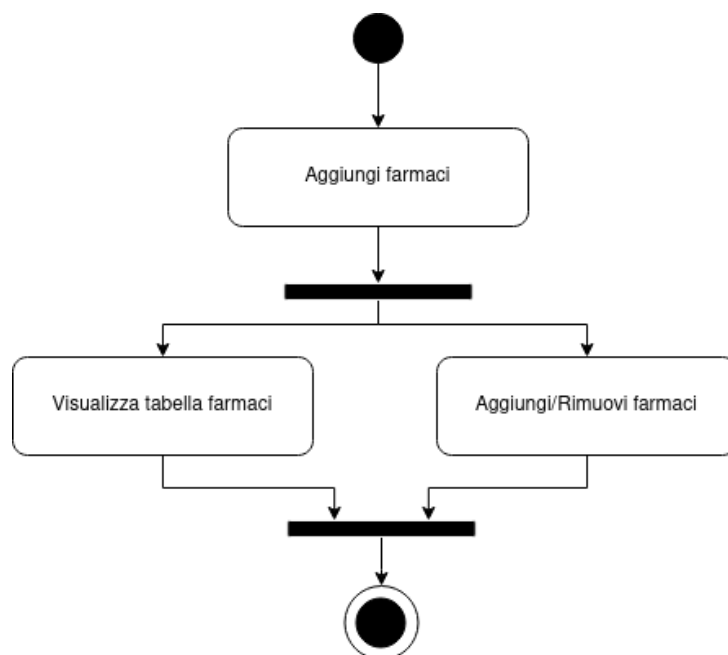


Figure 14: Aggiunta/Rimozione farmaco paziente

Diagramma attività dottore

Di seguito riportato un diagramma generale del flusso di attività del dottore, con a seguire la vista in dettaglio di ogni attività che può essere svolta.

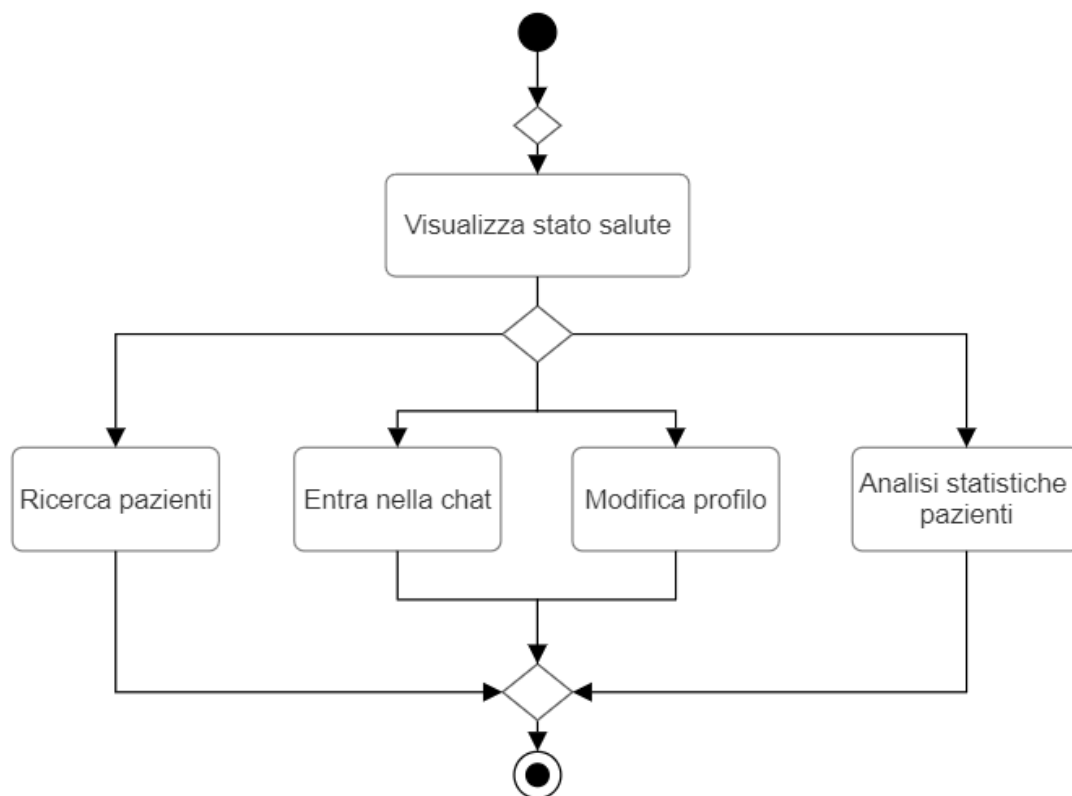


Figure 15: Activity diagram dottore

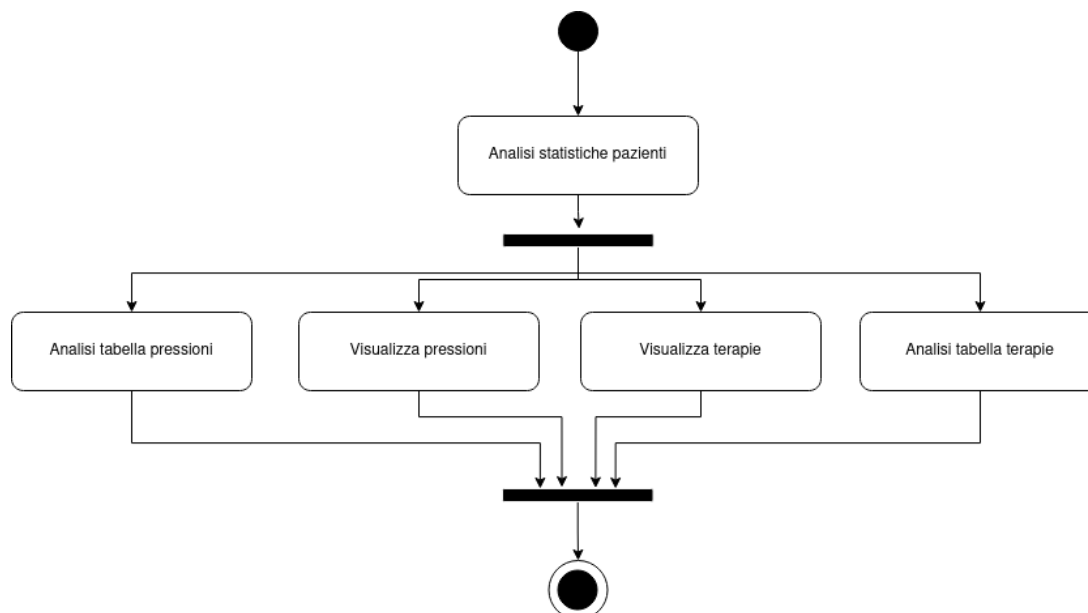


Figure 16: Analisi statistiche pazienti

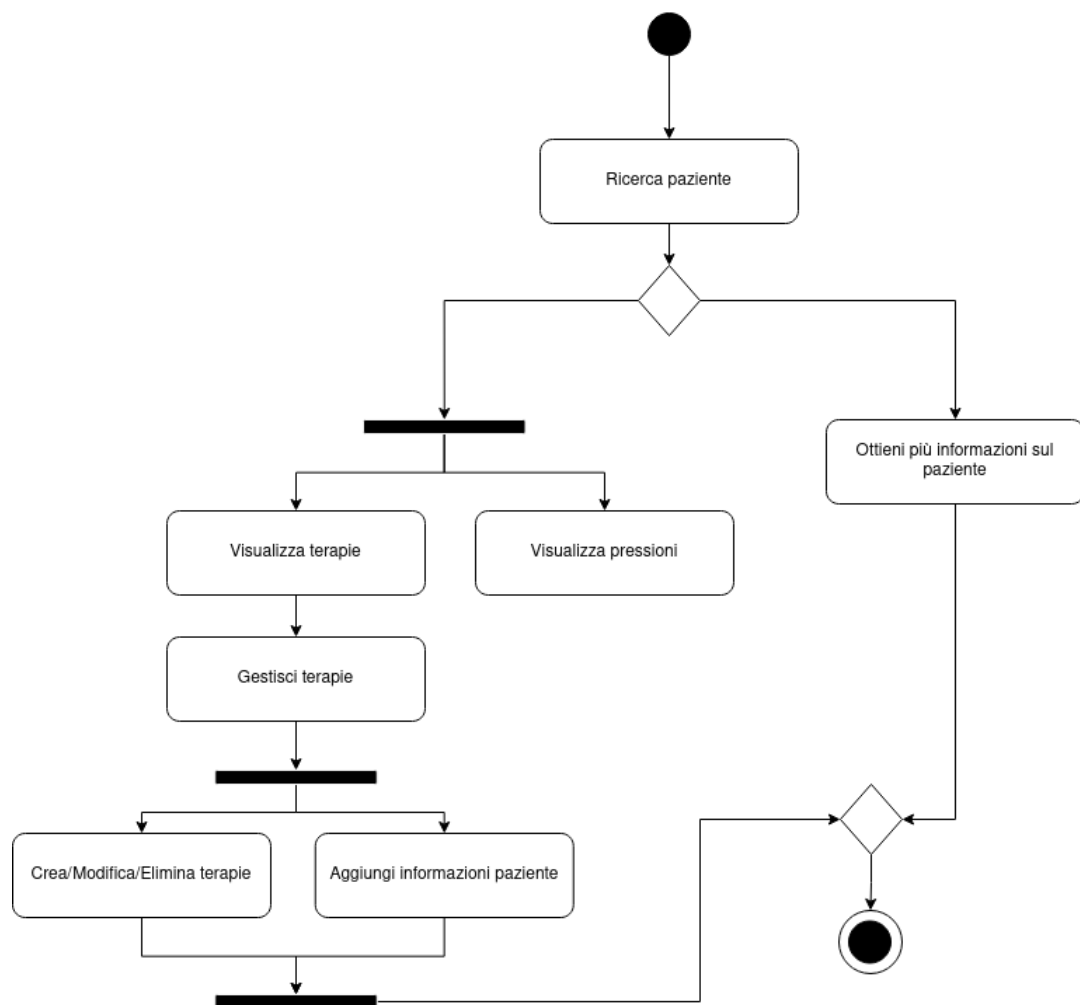


Figure 17: Analisi per paziente con terapie, pressioni e farmaci assunti


Sviluppo: progetto dell'architettura ed implementazione del sistema

Per il progetto si è scelto di seguire un approccio *Plan-Driven e Incrementale* che mette l'accento sulla pianificazione, l'analisi e la gestione delle attività di sviluppo del sistema. Come prima fase abbiamo raccolto i requisiti che il sistema avrebbe dovuto soddisfare tramite un'approfondita lettura delle specifiche richieste analizzando in particolare le parti più critiche di comunicazione tra le due tipologie di utenti. Dopo aver stabilito in dettaglio la fase di analisi, si è passato ad una fase di progettazione che includeva diagrammi di alto livello e descrizione delle componenti principali. Essendo il progetto sviluppato con un approccio incrementale si è deciso suddividere le varie *fasi di sviluppo* del software nel seguente modo:

- Fase di registrazione
- Implementazione paziente
- Implementazione dottore
- Implementazione comunicazione Paziente/Dottore/Sistema

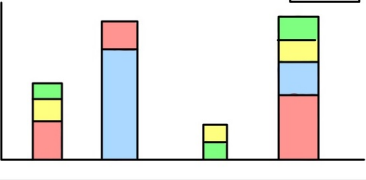
per ciascuna fase abbiamo valutato le varie alternative con annessi i possibili rischi e le varie GUI, scartando successivamente quelle che risultavano complesse e poco intuitive. Dopo aver stabilito le specifiche siamo passati alla fase di sviluppo dando in primo luogo maggiore attenzione alla costruzione degli **use-cases** principali e dei **diagrammi di attività** relativi ad ogni fase e parallelamente al design del sistema in modo da ottenere una versione effettiva dell'applicazione così da evitare fasi consistenti di re-coding. A fine dell'attività di sviluppo si è svolta una fase di testing e verifica del prodotto ponendo una maggiore attenzione per le fasi più critiche.

Esempio di analisi e sviluppo nella fase "implementazione del dottore", pagina analisi dottore



Dott. M. ROSSI


[Home](#)
[Pazienti](#)
[Analisi](#)
[Modifica Profilo](#)

GRAFICO



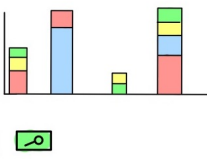
Tabella

Pressione	Terapie
<div><div>● Ipertensione I° GRADO</div><div>● Normale</div><div>● Ottimale</div><div>● Ipertensione III° GRADO</div></div>	<div>DATA INIZIO</div> <div>DATA FINE</div> <div></div>



Dott. M. ROSSI

[Home](#)
[Pazienti](#)
[Analisi](#)
[Modifica Profilo](#)

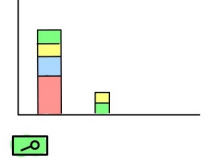
Pressioni




SCEGLI MESE



Terapie



SCEGLI MESE



Rischi:

- Inconsistenza dati grafico
- Possibili date errate con i mesi scelti da start a end

Rischi:

- Difficoltà di lettura
- Pesantezza del codice per una singola pagina

Per la parte di progettazione, è stato deciso di creare un progetto JavaFX e di lavorare con questa libreria, utilizzando come IDE IntelliJ. Per un maggiore parallelismo all'interno del team, abbiamo preferito creare una repository su GitHub per mantenere il codice sempre aggiornato a tutti i collaboratori al

progetto andando a caricare ogni versione con commento sulle implementazioni effettuate nel seguente giorno.

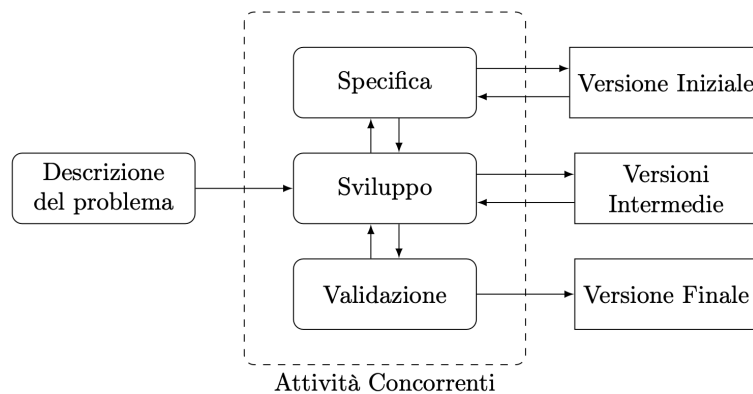


Figure 18: Schema riassuntivo adottato per ogni fase di sviluppo citata precedentemente

Con *versione finale* viene intesa la conclusione della fase con le classi annesse, che successivamente non è sono più state modificate se non per alcune sottigliezze.

Progettazione e pattern architetturali utilizzati

Il sistema progettato richiama due pattern architetturali standardizzati:

- **Model View Controller (MVC):** L'architettura Model View Controller è stata scelta ed utilizzata per favorire una separazione chiara sul lavoro che una classe deve andare a svolgere.

La componente **Model** contiene i metodi di accesso ai dati. Nel nostro caso le classi che appartengono a questa componente sono Therapies, Pressures e User, classe astratta che viene implementata dal Dottore o dal Paziente

La componente **View** gestisce la vista con l'utente, sono stati implementati vari file *.fxml* tra cui i principali : "User page" e "Doctor page". Questi, a loro volta, implementano altri file *.fxml* che richiamano tutte le pagine che l'utente può selezionare. Abbiamo preferito adottare questa metodologia in modo da rendere più leggero il codice della view ma allo stesso tempo mantenere un interfaccia grafica simile a quelle comuni ai giorni d'oggi (evitando così la chiusura della precedente schermata e l'apertura della nuova)

L'ultima componente **Controller** gestisce l'interazione tra il Model e la View, all'interno del progetto ogni file *.FXML* è associata ad una classe controller che cattura le azioni dell'utente all'interno del sistema e le gestisce di conseguenza.

L'utilizzo dell'architettura MVC ha reso il codice più modulare, facilmente mantenibile e facilmente testabile. Tutti questi vantaggi hanno permesso una migliore organizzazione del progetto ed un design più pulito.

Ogni membro del Model view controller è stato inserito all'interno di un package in modo da mantenere separate a livello logico le tre componenti utile per aggiornare l'applicazione con più facilità.

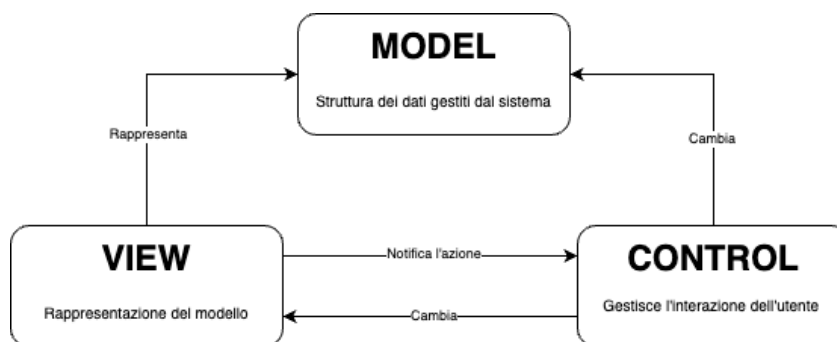


Figure 19: Architettura MVC

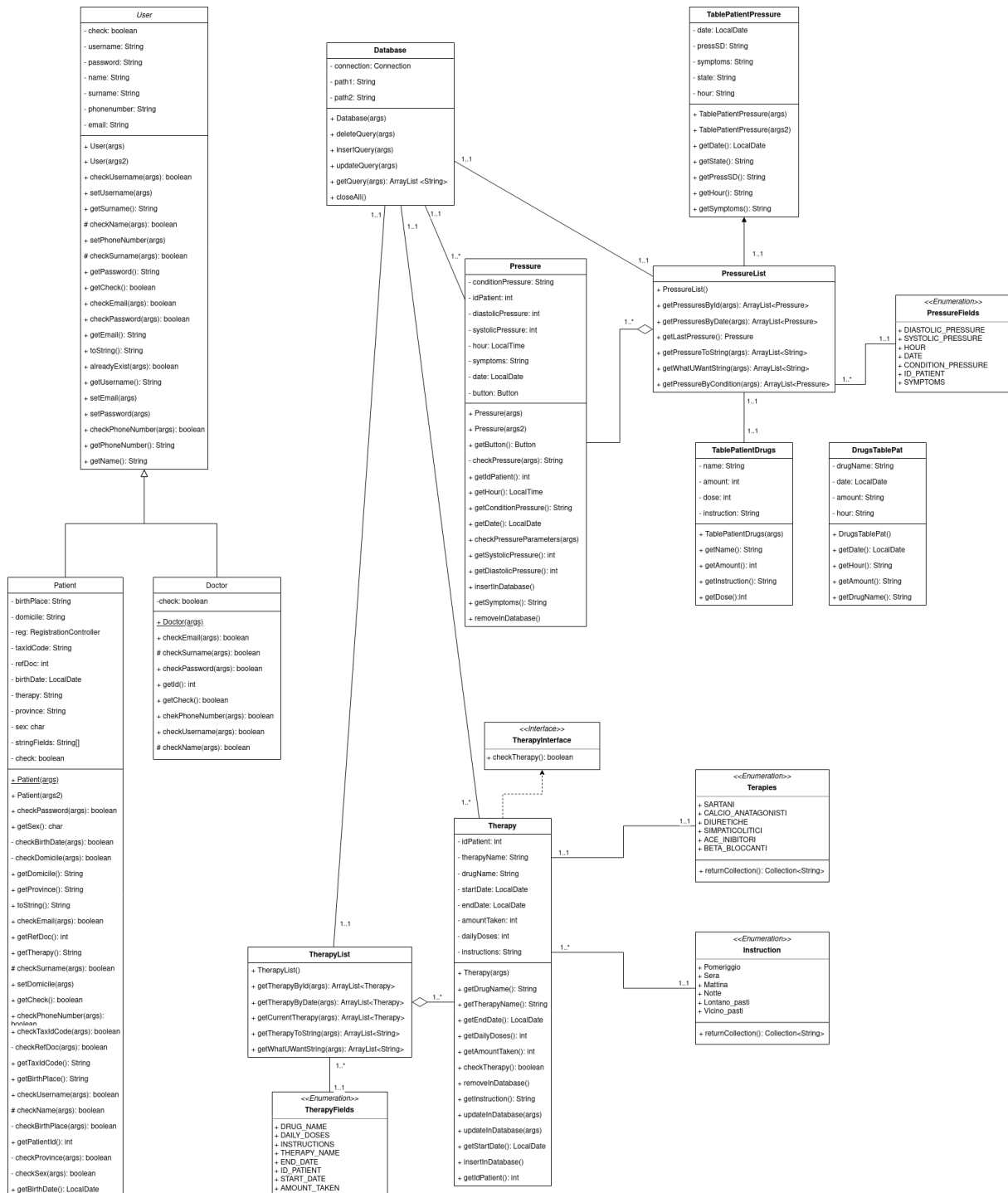


Figure 20: Diagramma delle classi del modello

- **Architettura a repository:** L'architettura a repository è stata scelta in modo intuitivo per soddisfare diverse esigenze nel progetto. La creazione di un database contenente tabelle pertinenti ha facilitato l'interrogazione e l'ottenimento delle informazioni necessarie, consentendo la comunicazione e l'elaborazione di dati tra componenti anche distinte. Inoltre, l'utilizzo dell'architettura a repository ha semplificato la gestione coerente dell'accesso e della persistenza dei dati. Questo ha reso il progetto più scalabile, in grado di adattarsi alle esigenze future senza dover apportare modifiche significative.

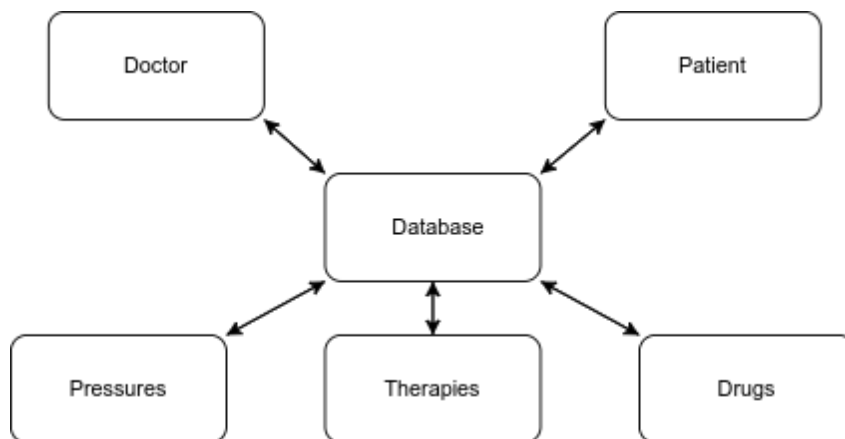


Figure 21: Architettura a repository utilizzata

L'implementazione dei due modelli non va intesa nella sua interezza, ad esempio il modello *MVC* viene "violato" con la nostra idea di implementazione dell'architettura a repository. A volte, abbiamo preferito infatti andare a modificare i dati all'interno del database direttamente dal controller, questo perché a nostro parere con alcuni valori (es. i farmaci con le quantità da assumere) non si aveva bisogno di effettuare operazioni ma soltanto gestirne l'aggiunta e la rimozione.

Implementazione e design pattern usati

Il sistema progettato richiama N design pattern:

- **pattern Iterator:** È stato utilizzato nel progetto per semplificare l'iterazione e l'accesso agli elementi di una collezione
- **pattern Facade:** È stato implementato nel progetto per semplificare l'interazione con un sottosistema complesso. Invece di creare una classe specifica che funge da maschera, il programma è stato interfacciato direttamente con alcune classi che collegano implicitamente quelle di livello inferiore. Questo ha permesso di nascondere all'utente la complessità sottostante del sistema, offrendo un'interfaccia più semplice e intuitiva. Un chiaro esempio della seguente implementazione è nella pagina "Analisi dottore". Il controller di questa classe si interfaccia con altre classi come "*Pressure list*", "*Therapy list*", "*Patient*" utilizzando i metodi delle seguenti fini al calcolo del diagramma che il dottore vorrà visualizzare.
- **pattern Template:** È stato implementato nella gestione delle classi "*Doctor*" e "*Patient*", con l'estensione alla classe "**abstract User**". Questa implementazione ha fornito un'interfaccia semplificata per l'interazione con le funzionalità legate ai medici e ai pazienti. La classe abstract User, implementa i metodi comuni ad entrambe le tipologie di utenti che possono accedere al sistema come ad esempio tutte le "*get*", o il controllo per dati già esistenti quali: pazienti già registrati, numeri di telefoni ed email già presenti nel sistema...
- **pattern Observer:** È il pattern più utilizzato all'interno del sistema nella sezione paziente. È stato utilizzato all'interno delle tabelle e grafici che mostrano i farmaci assunti e le pressioni registrate. Ad ogni **registrazione** [Soggetto] (aggiunta pressione / farmaco) tutti i **grafici e le tabelle**[Observer] vengono notificati in tempo reale e aggiornati di conseguenza. Questo è stato fondamentale per far visualizzare al paziente l'avvenuta dell'ultimo dato inserito e per poter visualizzare anche un andamento generale del suo stato salute attuale.

Diagrammi di sequenza del software implementato

Per ogni diagramma spiegheremo brevemente il loro funzionamento:

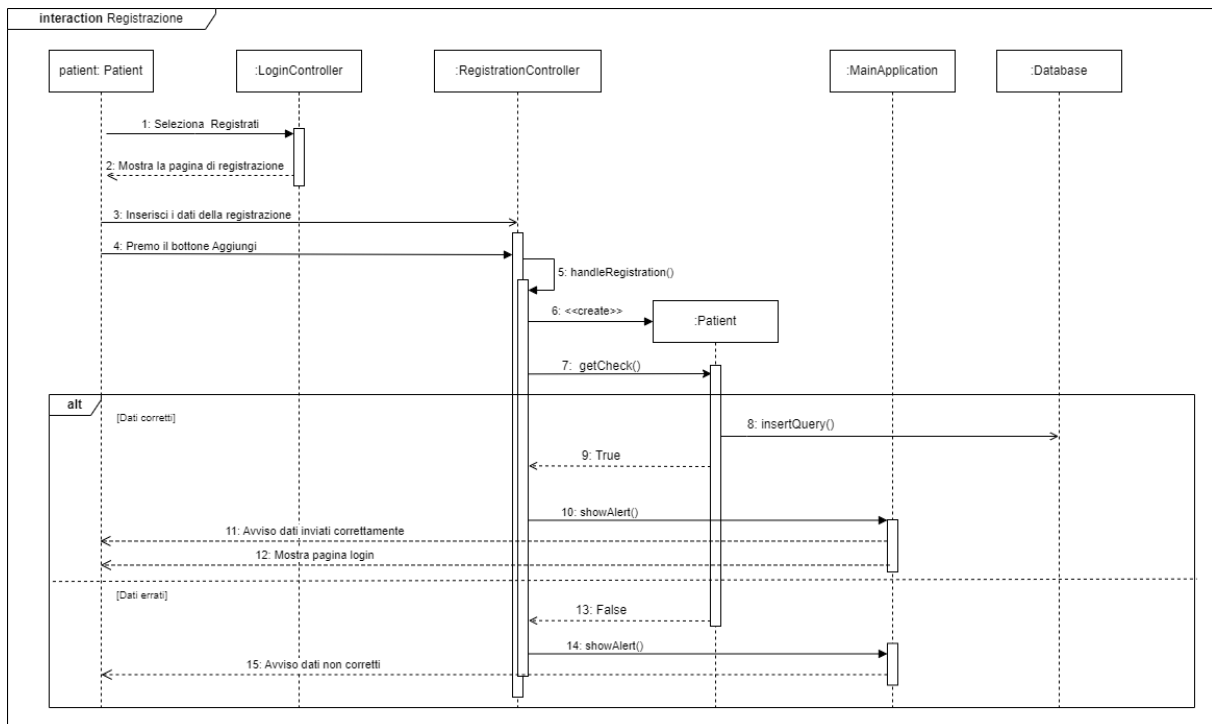


Figure 22: Registrazione del paziente

1. Il paziente appena aperta l'applicazione decide di effettuare una nuova registrazione
2. Il LoginController risponde mostrando la pagina di registrazione
3. Il paziente inserisce tutti i dati richiesti
4. Viene premuto il testo "Aggiungi"
5. Il RegistrationController gestisce la richiesta del paziente
6. Viene creato un nuovo paziente
7. Viene restituito il paziente
8. Viene effettuato il controllo sui dati inseriti, se già presente all'interno del database e se i dati inseriti sono leciti
9. Se c'è successo, il paziente viene registrato correttamente e tramite query viene inserito all'interno del database
10. Il paziente ritorna il valore di "getCheck"
11. Il RegistrationController chiede di mostrare un pop-up dove viene mostrato allo user la corretta avvenuta della registrazione
12. Viene mostrato il pop-up
13. Il paziente viene riportato alla pagina di login
14. Nel caso di dati errati Patient ritorna falso
15. Il controller chiede di mostrare il pop-up di incorretta avvenuta registrazione
16. Viene mostrato il pop-up

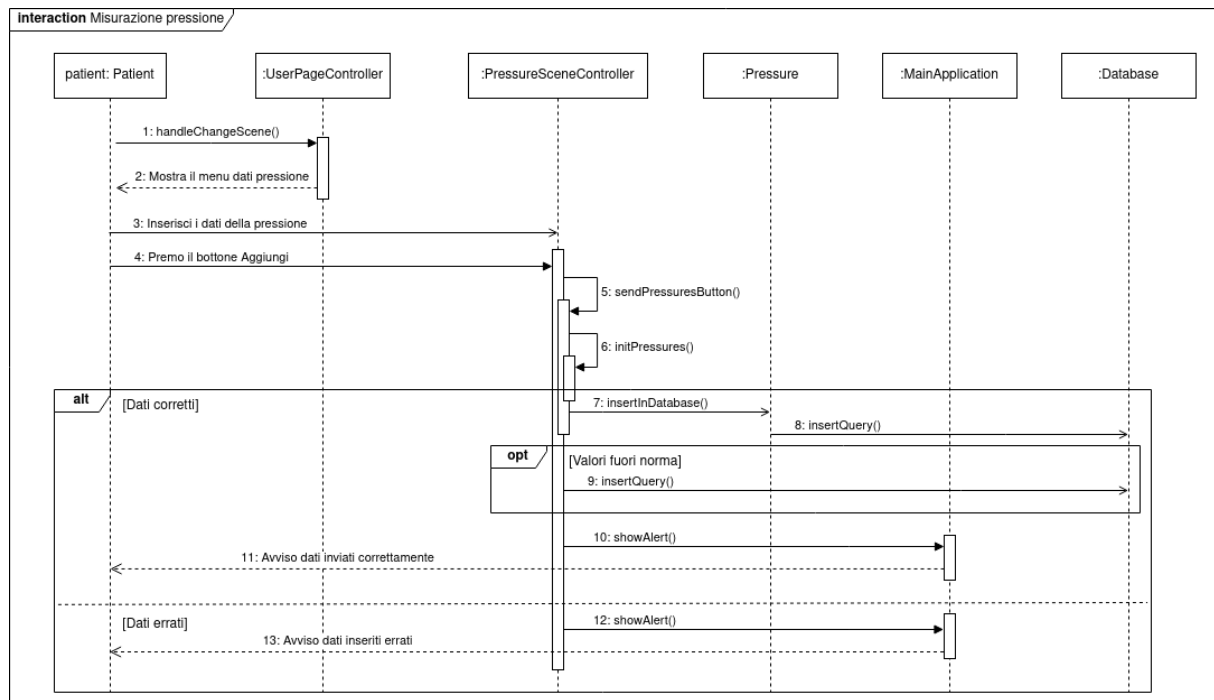


Figure 23: Misurazione pressione

1. Il paziente chiede di voler cambiare pagina
2. UserPageController risponde mostrando la pagina delle pressioni da registrare
3. Il paziente inserisce i dati relativi alla pressione
4. Viene schiacciato il tasto di aggiunta della nuova rilevazione della pressione
5. La pressione viene inserita nel database, se troppo alta, va nell'opzionale al punto "9"
6. Prende i dati della pressione appena inseriti e li restituisce in un nuovo modello di pressione
7. Viene richiesto l'inserimento all'interno del database
8. Nella classe pressure viene effettuato un inserimento della pressione appena registrata
9. Nel caso in cui i valori fossero alti, vengono inseriti nel database e viene notificato il dottore tramite una mail
10. A seguito della rilevazione viene chiesto di mostrare un pop-up
11. Viene mostrato il pop-up per la corretta avvenuta della registrazione
12. Nel caso i dati inseriti fossero inconsistenti (esempio un inserimento di pressione in un giorno al futuro) viene richiesto di mostrare un pop-up per l'incorretta registrazione della pressione
13. Il pop-up viene mostrato

Per quanto riguarda la registrazione dell'assunzione **farmaci**, il sequence diagram corrisponde in grosso modo alla registrazione della pressione, le uniche differenze sono le classi che vengono richiamate.

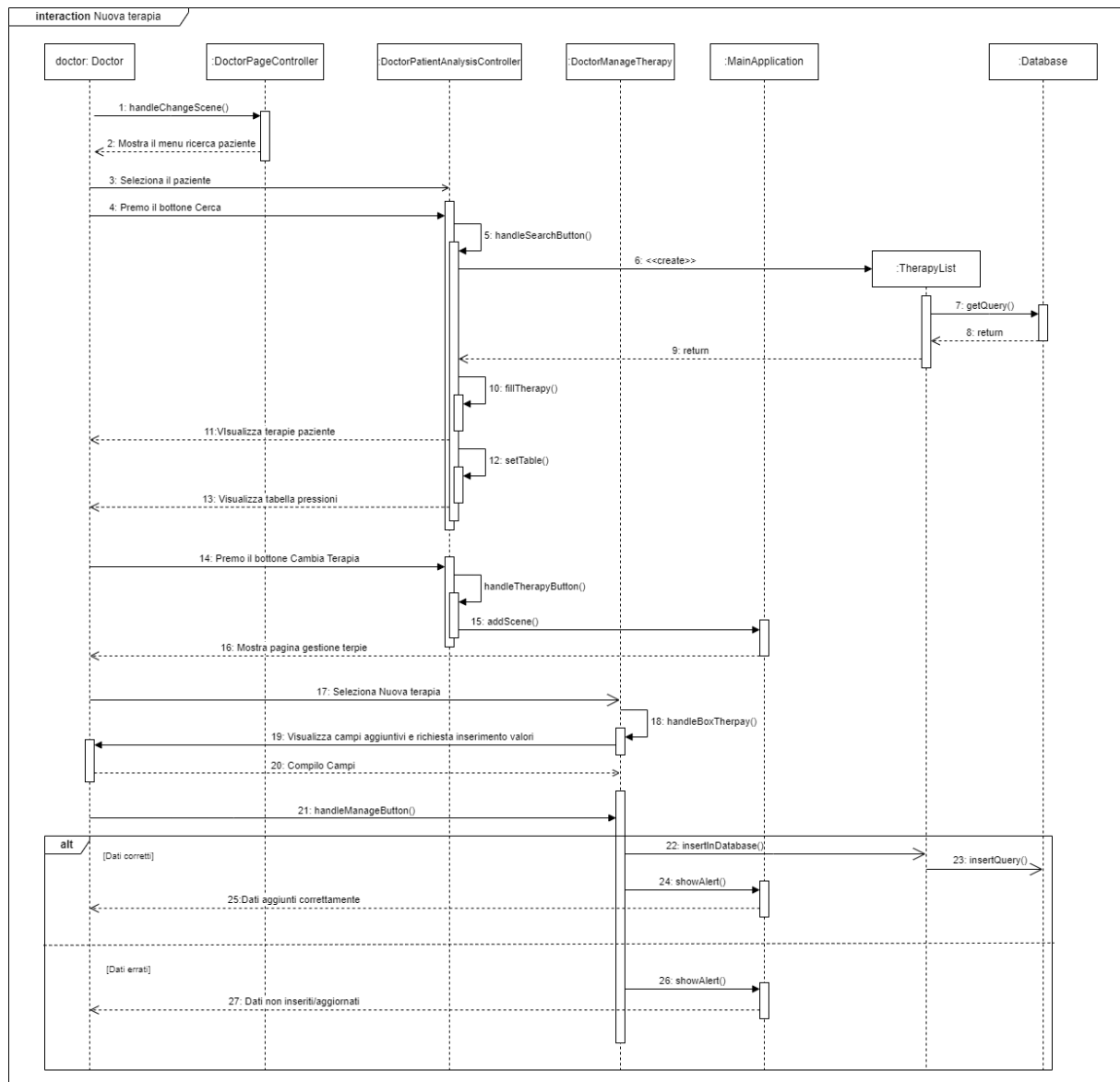


Figure 24: Nuova terapia dottore

1. Il dottore chiede di voler cambiare la schermata
2. Il DoctorPageController risponde mostrando la pagina di ricerca del paziente
3. Il dottore tramite una combobox, sceglie il paziente su cui vuole ottenere informazioni
4. Il dottore preme il bottone per la ricerca del paziente
5. Cerco il paziente selezionato
6. Instanzio una nuova classe TherapyList
7. Tramite una query vengono prese le terapie che il paziente ha già attive
8. Vengono ritornate le terapie
9. Le terapie vengono ritornate alla classe DoctorPatientAnalysisController
10. *fillTherapy()* Le terapie del paziente selezionato vengono sistemate in base alla rappresentazione che dovrebbe avere nella view
11. Vengono mostrate le terapie
12. Viene settata la tabella che successivamente dovrà essere mostrata nella view

13. Viene mostrata la tabella delle pressioni
14. Il dottore dopo aver visualizzato le terapie già presenti, esprime la facoltà di volerne aggiungere/cambiarne una
15. Il bottone schiacciato viene gestito
16. Viene aggiunta schermata aggiuntiva in modo che il dottore possa verificare in quella sottostante le terapie già presenti e nella nuova decidere quale nuova terapia inserire
17. La nuova schermata viene mostrata in view
18. Il medico tramite una combobox potrà scegliere se modificare le terapie già presenti o crearne una nuova
19. A seconda dell'azione intrapresa gestisco la view in modo differente
20. Vengono mostrati tutti i campi aggiuntivi che possono completare una terapia: data, nome terapia e farmaci da assumere
21. Il dottore compila i campi mancanti per poter registrare una nuova terapia
22. Il dottore preme il pulsante cambia terapia
- A: Se i dati inseriti sono corretti:
 23. Viene richiamata la funzione insert nel database da parte di therapyList
 24. Aggiungo i parametri nel database
 25. Viene richiesto di mostrare un pop-up
 26. Viene mostrato il pop-up di corretto cambiamento/aggiunta della terapia
- B: Se i dati inseriti non sono corretti:
 27. Viene richiesto di mostrare un pop-up
 28. Viene mostrato il pop-up di errore

Attività di test e validazione

Durante lo sviluppo del prototipo, sono state effettuate prove e test su ogni componente del sistema che è stato modificato o aggiunto alla fine di ogni fase precedentemente citata. Dopo il completamento del progetto, i membri del gruppo di progettazione hanno eseguito test per verificare il corretto funzionamento complessivo del programma e dei meccanismi specifici. Successivamente, sono stati coinvolti anche utenti esterni, come parenti e amici, per individuare eventuali errori non gestiti, particolari non considerati durante lo sviluppo o modifiche necessarie alle schermate del programma. La fase di testing condotta da individui non tecnici è stata fondamentale per rendere il programma il più user-friendly possibile.

Le attività di test si sono basate principalmente sull'esecuzione ripetuta del programma, svolgendo diverse azioni. In caso di errori o bug, è stata effettuata un'analisi del codice per risolvere il problema. Durante questa fase, l'uso del debugger è stato fondamentale per individuare e correggere eventuali problemi.

Infine, il software è stato sottoposto a un test condotto tramite Junit un framework di testing per il linguaggio java, il test tramite questo framework ha testato che potevano provocare più problemi a livello software. Tutte le fasi di test sono state svolte eliminando tutti gli errori così da avere un programma senza nessun tipo di problema software e logico

Codice Test

```
class UserTest {
    Patient p = new Patient(null, "alessia", "de benedittis", "pazzoini@gmail.com", "3417029110",
        "ilPazzo", "PAzzio23", "bari", "ba", LocalDate.parse("2001-08-01"),
        "Via piazza flavio", 'F',"DBNLSS01M41A662P" , 2);
    @Test
    void getTestPatient() {
        assertEquals("alessia", p.getName());
        assertEquals("Pazzini", p.getSurname());
        assertEquals("pazzoini@gmail.com", p.getEmail());
        assertEquals("3417029110", p.getPhoneNumber());
        assertEquals("ilPazzo", p.getUsername());
    }

    @Test
    void testTaxIDCode() {
        assertTrue(p.checkTaxIdCode(p.getTaxIDCode()));
    }

    @Test
    void testEmail(){
        assertTrue(p.checkEmail(p.getEmail()));
    }

    // DOCTOR TEST

    Doctor d = new Doctor("Giacomo", "Mandrillo", "Mandri@gmail.com", "345698741", "Ilmandri",
        "123Ilmandri!");

    @Test
    void checkPhoneNumber() {
        assertTrue(d.checkPhoneNumber(d.getPhoneNumber()));
    }

    @Test
    void checkEmail(){
        assertTrue(d.checkEmail(d.getEmail()));
    }
}
```

Funzionalità aggiuntive

Nella fase di sviluppo del progetto, sono state implementate diverse parti aggiuntive per migliorare l'applicazione e arricchire l'esperienza degli utenti finali. Le funzionalità aggiuntive sono:

- **Registrazione paziente:** Abbiamo deciso di implementare la funzionalità di registrazione da parte dei pazienti. Il paziente per potersi registrare deve inserire i propri dati personali come nome, cognome, residenza, ecc.. Ogni campo viene sottoposto ad un controllo specifico sulla consistenza del dato, l'email deve contenere una "@" e un ".", la password deve essere sicura quindi lunghezza minima 8 e deve contenere caratteri speciali. Vorremo porre particolare attenzione sul controllo scrupoloso che viene effettuato sul campo CodiceFiscale, permettendo un maggiore controllo sui valori inseriti, nello specifico: codice fiscale, nome, cognome, data di nascita, luogo di nascita e sesso. Per rendere possibile questo controllo ci siamo affidati all'utilizzo di un database, preso dal sito dell'Agenzia Delle Entrate, dove risiedono tutti i comuni con associato il proprio Codice Comune. Con l'assemblamento e l'elaborazione di tutti i dati forniti il risultato finale dichiara se c'è un'anomalia o meno nei dati inseriti dall'utente.
- **Filtraggio dei pazienti da parte del dottore (Accordati con il professore):** Il dottore dopo essersi autenticato ha la possibilità di controllare i propri pazienti in due modi: Il primo modo consiste nel cercare un paziente singolo attraverso il nome ed il cognome. Tuttavia questa modalità ci è sembrata troppo limitativa per l'esigenza del progetto, pertanto abbiamo deciso di implementare un secondo modo che offrisse maggiori opzioni e flessibilità nella ricerca dei pazienti. Il secondo modo permette al dottore di controllare la situazione in modo generale, controllando se le proprie terapie stanno avendo un buon riscontro o meno, tutto questo viene mostrato tramite l'utilizzo di grafici, rappresentando le situazioni delle pressioni effettuate dai pazienti. Inoltre, è possibile visualizzare le stesse informazioni in formato tabella, facilitando così la ricerca di pazienti con pressioni specifiche o terapie in corso. L'implementazione di questa seconda modalità amplia le opzioni di monitoraggio e analisi delle informazioni dei pazienti da parte del dottore. Ciò permette una valutazione più approfondita delle terapie e semplifica la gestione globale del trattamento.
- **Implementazione di una chat dottore/paziente:** abbiamo pensato fosse utile anche una chat tra medico e paziente in modo da comunicare in maniera facile, veloce e professionale informazioni utili al fine di migliorare l'user-experience di entrambi gli user. Questa idea è arrivata dopo aver pensato ad un possibile scambio di messaggi tra parte dei due user: magari il paziente ha bisogno di una prescrizione per il determinato farmaco citato dal dottore o deve comunicare la mancanza ad un possibile appuntamento prefissato, quindi normalmente avrebbe dovuto mandare una mail e aspettare una risposta. A noi è sembrato più intuitivo implementare una sorta di casella postale all'interno dell'applicazione in cui il paziente dopo aver effettuato le sue registrazioni giornaliere, comunica con il suo medico senza cambiare alcuna applicazione. Stessa cosa è possibile da parte del dottore, implementata per lo stesso motivo. Un esempio di utilità è sicuramente il medico che deve rimuovere un ipotetico appuntamento per una qualsiasi causa e lo comunica quindi tramite chat.