



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

CORSO DI
INGEGNERIA, GESTIONE ED EVOLUZIONE DEL SOFTWARE

Modification Report

HeartCare

DOCENTE

Prof. Andrea De Lucia

Università degli Studi di Salerno

AUTORI

Annamaria Basile

Matricola: 0522501844

Paolo Carmine Valletta

Matricola: 0522501828

Alessandro Zoccola

Matricola: 0522501804

Anno Accademico 2023-2024

Indice

Elenco delle Tabelle	iv
1 Introduzione	1
1.1 Panoramica del Sistema HeartCare	1
1.2 Change Request	2
1.3 Struttura documento	2
2 Analisi di HeartCare	4
2.1 Architettura del sistema	4
2.1.1 Panoramica	4
2.1.2 Decomposizione in sottosistemi	4
2.1.3 Dipendenze tra sottosistemi	6
2.2 Class Diagram	8
2.3 Use Case	9
2.4 Sequence Diagram	13
3 Cambiamenti Proposti	15
3.1 Obiettivi CR_1	15
3.2 Obiettivi CR_2	16
3.3 Obiettivi CR_3	16

4	Impact Analysis	17
4.1	CR_1 Migrazione dati	18
4.1.1	Starting Impact Set - SIS	18
4.1.2	Candidate Impact Set - CIS	18
4.2	CR_2 Aggiunta nuova tipologia di utente	18
4.2.1	Starting Impact Set - SIS	18
4.2.2	Analisi delle dipendenze	19
4.2.3	Candidate Impact Set - CIS	20
4.3	CR_3 Modifica GUI	23
4.3.1	Starting Impact Set - SIS	23
4.3.2	Analisi delle dipendenze	23
4.3.3	Candidate Impact Set - CIS	23
5	Implementazione dei Cambiamenti	25
5.1	Metodologia	25
5.2	CR_1	26
5.2.1	Parte Implementativa	26
5.2.2	Discovered Impact Set - SIS	28
5.2.3	Actual Impact Set - AIS	28
5.2.4	False Positive Impact Set - FPIS	28
5.2.5	Metriche di processo	28
5.3	CR_2	29
5.3.1	Parte Implementativa	29
5.3.2	Discovered Impact Set - SIS	29
5.3.3	Actual Impact Set - AIS	30
5.3.4	False Positive Impact Set - FPIS	33
5.3.5	Metriche di processo	33
5.4	CR_3	34
5.4.1	Parte Implementativa	34
5.4.2	Discovered Impact Set - SIS	34
5.4.3	Actual Impact Set - AIS	35

5.4.4	False Positive Impact Set - FPIS	35
5.4.5	Metriche di processo	36
6	Conclusioni	38
6.1	Sintesi delle Modifiche	38
6.2	Lesson learned	38

Elenco delle tabelle

1.1	La tabella delle change requests	3
2.1	Use case: Avvio misurazione	10
2.2	Use case: Assegnamento caregiver	11
2.3	Use case: Flusso di eventi assegnamento caregiver	12
4.1	Candidate Impact sets CR_2	21
4.2	Candidate Impact sets CR_2	22
4.3	Candidate Impact sets CR_3	24
5.1	Actual Impact sets CR_2	31
5.2	Actual Impact sets CR_2	32
5.3	Actual Impact sets CR_3	37

CAPITOLO 1

Introduzione

1.1 Panoramica del Sistema HeartCare

HeartCare è una piattaforma che fornisce uno strumento di telemonitoraggio ai pazienti affetti da malattie cardiache, permettendo loro di tenere sotto osservazione i propri valori e di ricevere notifiche preventive in caso di possibili problemi grazie all'applicazione dell'Intelligenza Artificiale. Il paziente può, tramite le misurazioni, aggiornare il proprio Fascicolo Sanitario Elettronico (FSE), che è accessibile al medico.

HeartCare è stata sviluppata per offrire un supporto online alle problematiche delle persone cardiopatiche. La piattaforma consente al paziente di accedere al proprio FSE, di aggiungere note destinate al medico, di registrare un nuovo dispositivo di monitoraggio e di eseguire misurazioni. Per il medico, HeartCare offre l'accesso ai fascicoli di ogni suo paziente, la possibilità di inviare note destinate al paziente e di stabilire il giorno della prossima visita. Inoltre, un amministratore gestisce la presenza di medici e pazienti sulla piattaforma e le relazioni tra di loro.

Ogni paziente può assegnarsi autonomamente un caregiver inserendo la sua email, nome e cognome. Quest'ultimo è una persona esterna al sistema che riceverà comunicazioni tramite mail qualora una misurazione effettuata dal paziente avrà prodotto dei valori pericolosi.

1.2 Change Request

Uno dei limiti maggiori di Heartcare è la mancanza di una sezione dedicata al caregiver. Attualmente il caregiver è un attributo del paziente e non esiste una tipologia di utente specifico. L'unica funzionalità disponibile per il caregiver è ricevere via email una notifica di conferma per essere diventato un caregiver e successivamente una notifica in caso di misurazioni anomale del paziente da cui è stato scelto.

L'obiettivo è quello di aggiungere una nuova tipologia di utente al sistema e applicare le modifiche che ne conseguono. In particolare verrà aggiunta una nuova entità, effettuando una migrazione dei dati pre esistenti nel database, compresi tutti i record, le tabelle e le relazioni verso l'entità appena creata nel sistema. Questo processo include l'analisi e la trasformazione dei dati, l'assegnazione corretta dei nuovi identificativi e l'assicurazione che tutte le informazioni siano correttamente integrate e accessibili nella nuova struttura dati.

Inoltre è necessario lo sviluppo di una nuova sezione a cui il caregiver potrà accedere per ed usare la funzionalità messa a disposizione. Nella tabella ?? viene riportata la lista delle Change Request approvate.

1.3 Struttura documento

Il documento sarà strutturato in 6 capitoli: Analisi di Heartcare (2), Cambiamenti Proposti (3), Impact Analysis (4), Implementazione dei cambiamenti (5) e Conclusioni (6). Nel secondo capitolo si effettua un'analisi del vecchio sistema proposto, sulla quale verranno applicate le Change Request. Nel terzo capitolo sarà presente una Sezione dove proponiamo i cambiamenti proposti e gli obiettivi correlati per ogni CR. Nel quarto capitolo, viene definita una sezione per ogni change requests, che documenterà l'Impact Analysis effettuata. Nel quinto capitolo saranno presenti le informazioni relative alle vere e proprie modifiche effettuate e i relativi risultati delle modifiche attuate, come ad esempio Actual Impact Set. Infine, nel capitolo di Conclusione sono stati riportati: l'esito finale delle modifiche e le lezioni apprese durante lo svolgimento di questo progetto.

CR_ID	Titolo	Descrizione	Priorità	Impatto	Difficoltà
CR_1	Migrazione dati	Effettuare la migrazione dei dati pre esistenti nel database, verso la nuova entità creata.	Alta	Medio	Media
CR_2	Aggiunta nuova tipologia di utente	Viene aggiunto una nuova tipologia di utente al sistema, il caregiver.	Alta	Alto	Difficile
CR_3	Modifica GUI	Aggiungere una nuova pagina dedicata al caregiver in modo che possa visualizzare le informazioni dei pazienti a cui è associato.	Alta	Alto	Media

Tabella 1.1: La tabella delle change requests

Analisi di HeartCare

2.1 Architettura del sistema

Il sistema Heartcare utilizza un'architettura Three-tier. Questo significa che è strutturato in tre livelli distinti. Come illustrato nella Figura 2.1, i tre tier sono:

2.1.1 Panoramica

- **Persistence Tier:** responsabile del recupero e della persistenza dei dati,
- **Business Tier:** che gestisce la logica di business e le funzionalità dell'applicazione,
- **Presentation Tier:** incaricato di presentare l'interfaccia utente.

2.1.2 Decomposizione in sottosistemi

Inoltre nel sistema così progettato, sono stati identificati i seguenti sottosistemi:

- **Gestione Utente:** si occupa delle funzionalità di login e logout. Inoltre si occuperà anche della gestione dei dati personali degli account.
- **Registrazione:** si occupa di gestire la registrazione degli utenti alla piattaforma.

- **Gestione Misurazione:** si occupa della gestione delle misurazioni e delle funzionalità ad esse legate, si occupa anche delle funzioni di inserimento, modifica e cancellazione di un dispositivo.
- **Gestione Visita:** si occupa della gestione delle visite.
- **Comunicazione:** si occupa della gestione delle note tra paziente e medico e delle notifiche.
- **Storage JPA:** modulo di Java Spring che si interpone tra l'insieme dei sottosistemi e il sottosistema di persistenza.
- **Persistenza:** si occupa di gestire la persistenza dei dati con un database relazionale.

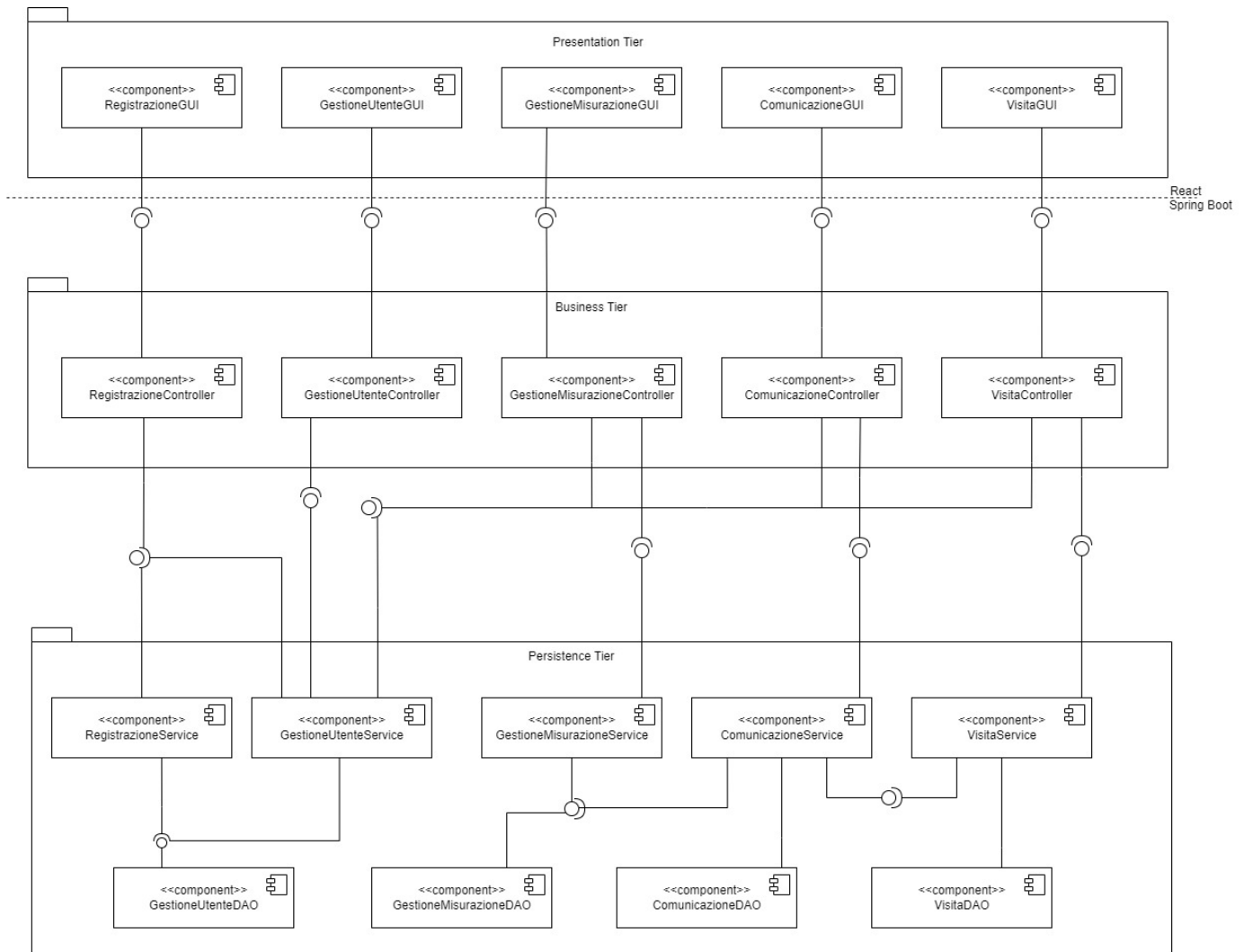


Figura 2.1: Diagramma architetturale

2.1.3 Dipendenze tra sottosistemi

Di seguito sono riportate, come mostrato in Figura 2.2, le varie dipendenze tra i sottosistemi di cui è composto HeartCare.

- **Registrazione:** utilizza funzionalità dei sottosistemi Gestione Utente e Storage JPA,
- **Gestione Utente:** utilizza funzionalità del sottosistema Storage JPA,
- **Gestione Visita:** utilizza funzionalità dei sottosistemi Comunicazione, Gestione Utente e Storage JPA,
- **Comunicazione:** utilizza funzionalità dei sottosistemi Gestione Utente e Storage JPA,
- **Gestione Misurazione:** utilizza funzionalità dei sottosistemi Comunicazione, Gestione Utente e Storage JPA,
- **Storage JPA:** utilizza funzionalità del sottosistema Persistenza,
- **Persistenza:** utilizza il servizio di hosting per l'archiviazione dei dati.

Alcuni sottosistemi sono gestiti da componenti COTS (Commercial Off The Shelf), come Storage JPA che viene fornito da Java Spring e Persistenza che è gestita attraverso un DBMS relazionale. Anche il sottosistema Comunicazione farà riferimento a componenti COTS.

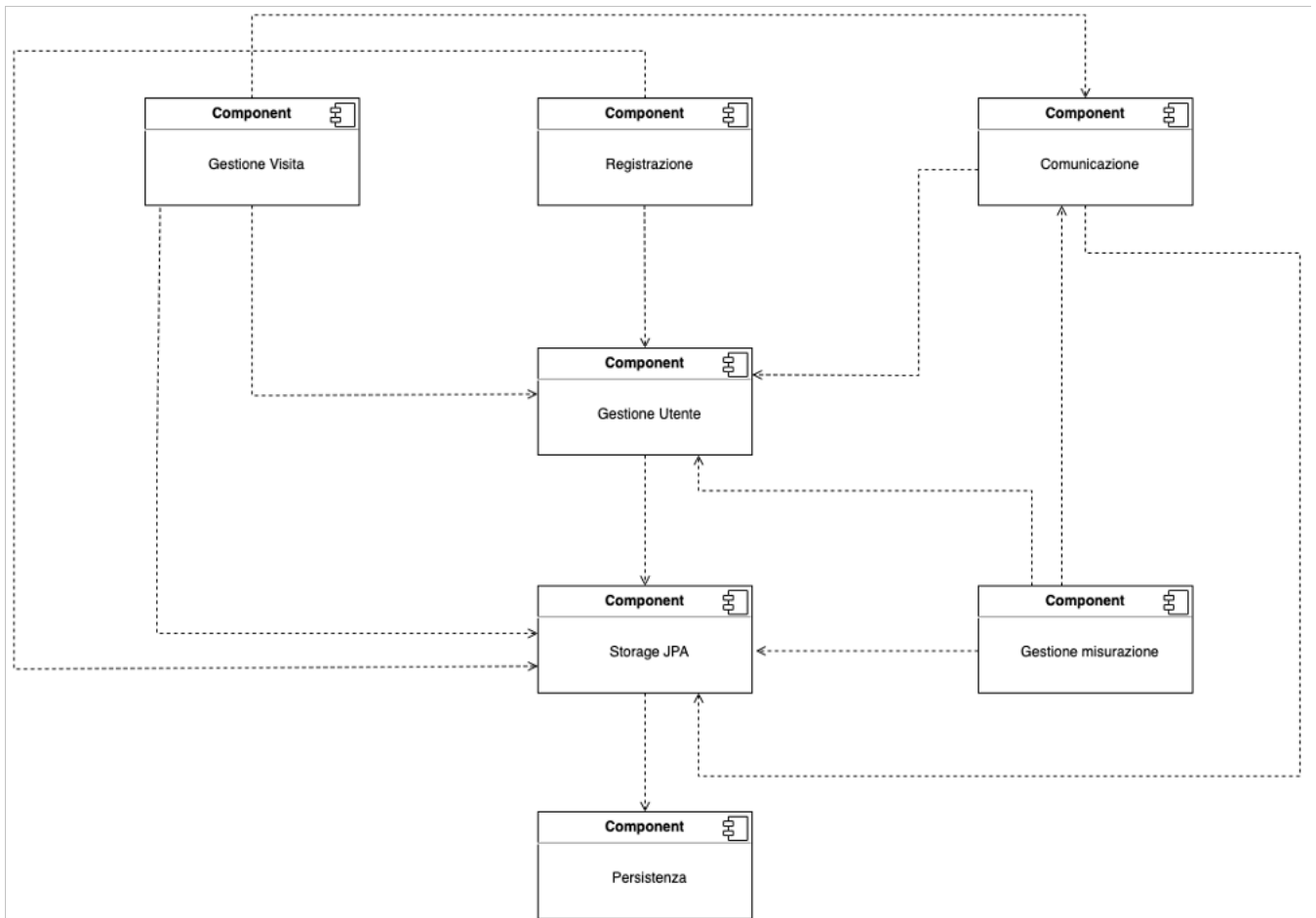


Figura 2.2: Dipendenze dei sottosistemi

2.2 Class Diagram

Per comprendere meglio l'architettura del sistema riportiamo nella figura 2.3 il class diagram.

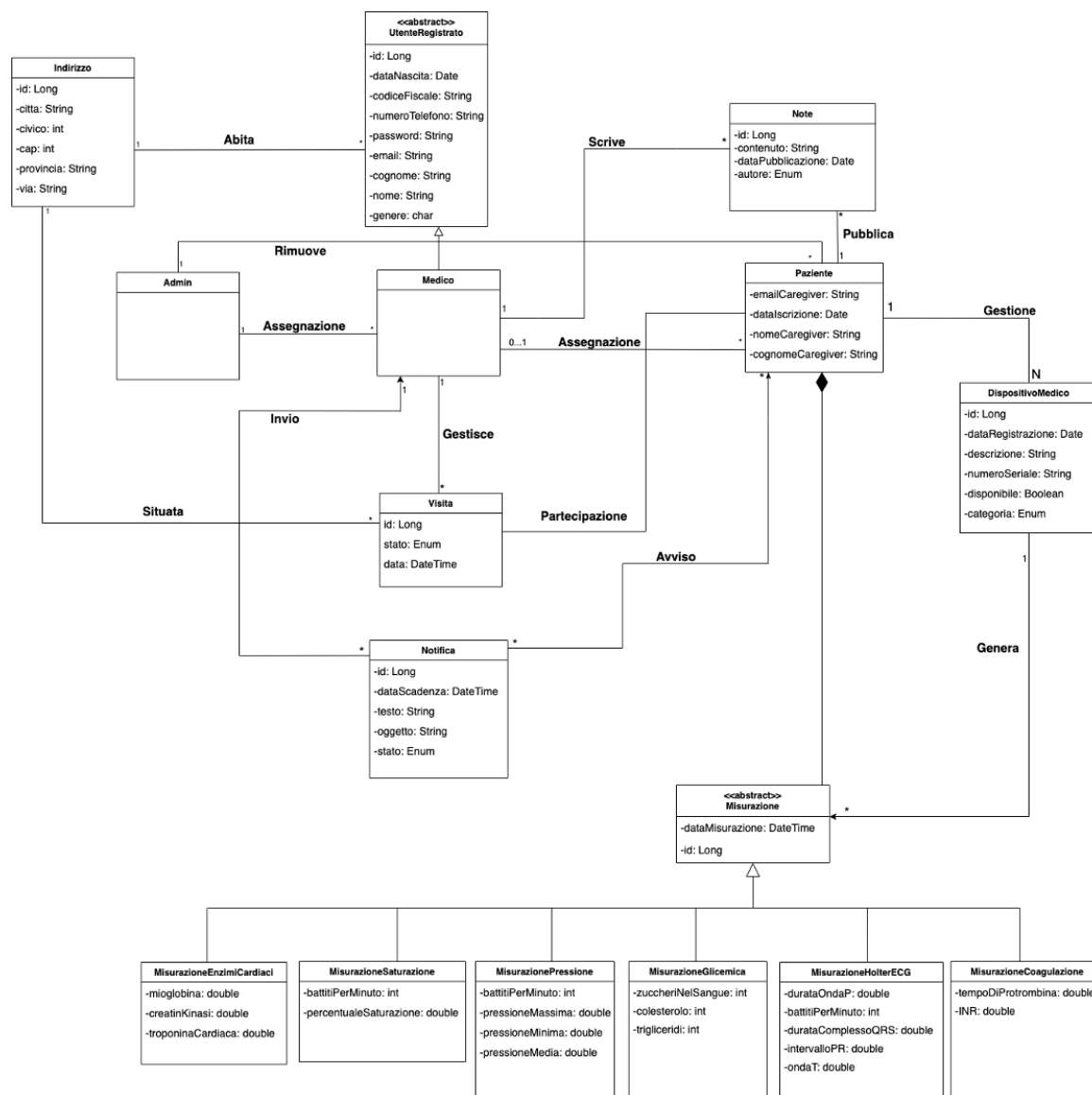


Figura 2.3: Class diagram

2.3 Use Case

Nella tabella 2.1 è presentato il nostro use case chiave, che rappresenta il requisito funzionale fondamentale del sistema e descrive il processo di misurazione effettuato da un paziente.

In aggiunta, nella tabella 2.3, troviamo il dettaglio relativo all'assegnazione di un caregiver da parte di un paziente. Questo use case sarà particolarmente influenzato dalle modifiche che intendiamo apportare al sistema.

Campo	Descrizione
Nome	Avvio monitoraggio
Descrizione	Questo UC descrive la funzionalità di avvio di una sessione di monitoraggio da parte di un paziente.
Attore principale	Paziente: vuole effettuare una misurazione
Attore secondario	NA
Entry condition	Un paziente vuole effettuare una misurazione.
Exit condition (on success)	La misurazione che si genera entra nel Fascicolo Sanitario Elettronico.
Exit condition (on failure)	Viene mostrato un messaggio di errore.
Rilevanza/User priority	Alta
Frequenza stimata	1/giorno
Scenari alternativi	Altri percorsi possibili del Use Case
Eccezioni	Gestione delle eccezioni o errori
Flusso di eventi principale/Main scenario	
Attore	Descrizione
Paziente	1: Preme il bottone "Avvia Misurazione" presente nella Home
Sistema	2: Mostra una scheda in cui mostra i dispositivi assegnati al paziente da cui far partire la misurazione.
Paziente	3: Sceglie "Saturimetro" e conferma.
Sistema	4.1: Il sistema avvia il monitoraggio ed invia una notifica per avvisare il paziente circa l'avvenuto salvataggio della misurazione.
Flusso di eventi alternativo	
Sistema	4.2: In caso di misurazione con valori anomali il sistema invia una notifica al caregiver e al paziente.

Tabella 2.1: Use case: Avvio misurazione

Campo	Descrizione
Nome	Assegnamento Caregiver
Descrizione	Questo UC descrive la funzionalità di assegnazione di un proprio caregiver da parte di un paziente.
Attore principale	Paziente: è interessato ad assegnare un proprio caregiver.
Attore secondario	NA
Entry condition	Un utente è loggato come paziente AND Un paziente è interessato all'assegnamento di un proprio caregiver AND Il sistema deve fornire al paziente una pagina per l'inserimento del caregiver.
Exit condition (on success)	Al paziente viene assegnato il caregiver e viene inviata una mail al suo caregiver.
Exit condition (on failure)	Il caregiver non viene assegnato al paziente.
Rilevanza/User priority	Alta
Frequenza stimata	3/anno
Scenari alternativi	Altri percorsi possibili del Use Case
Eccezioni	Gestione delle eccezioni o errori

Tabella 2.2: Use case: Assegnamento caregiver

Flusso di eventi principale/Main scenario	
Attore	Descrizione
Paziente	1: Richiede l'assegnazione del caregiver cliccando su "Profilo"
Sistema	2: Restituisce la pagina del profilo del paziente contenente le informazioni personali tra cui il caregiver.
Paziente	3: Clicca sul pulsante "modifica".
Sistema	4: Fa comparire un form con tutte le informazioni personali tra cui: Nome, Cognome ed E-mail Caregiver
Paziente	5: Inserisce i dati richiesti e clicca "Conferma"
Sistema	6: Verifica la correttezza dei campi
Sistema	7.1: Fa apparire un messaggio di corretta assegnazione e invia una mail di conferma al caregiver.
Flusso di eventi alternativo: il paziente non inserisce una mail nel formato corretto	
Sistema	7.2a: Mostra un messaggio di errore sotto al form informando al paziente di scrivere un testo nel formato nome@dominio.
Sistema	7.3a: Rimane in attesa che il paziente riscriva la mail.
Flusso di eventi alternativo: il paziente non inserisce nome e/o cognome	
Sistema	7.2b: Mostra un messaggio di errore sotto al form informando il paziente di inserire nome e/o cognome nei campi lasciati vuoti.
Sistema	7.3b: Rimane in attesa che il paziente compili i campi vuoti.

Tabella 2.3: Use case: Flusso di eventi assegnamento caregiver

2.4 Sequence Diagram

In questa sezione riporteremo i sequence diagram dei casi d'uso descritti precedentemente. Attraverso questi diagrammi, potremo comprendere in dettaglio come gli attori del sistema interagiscono con esso.

Sequence diagram - Avvio monitoraggio

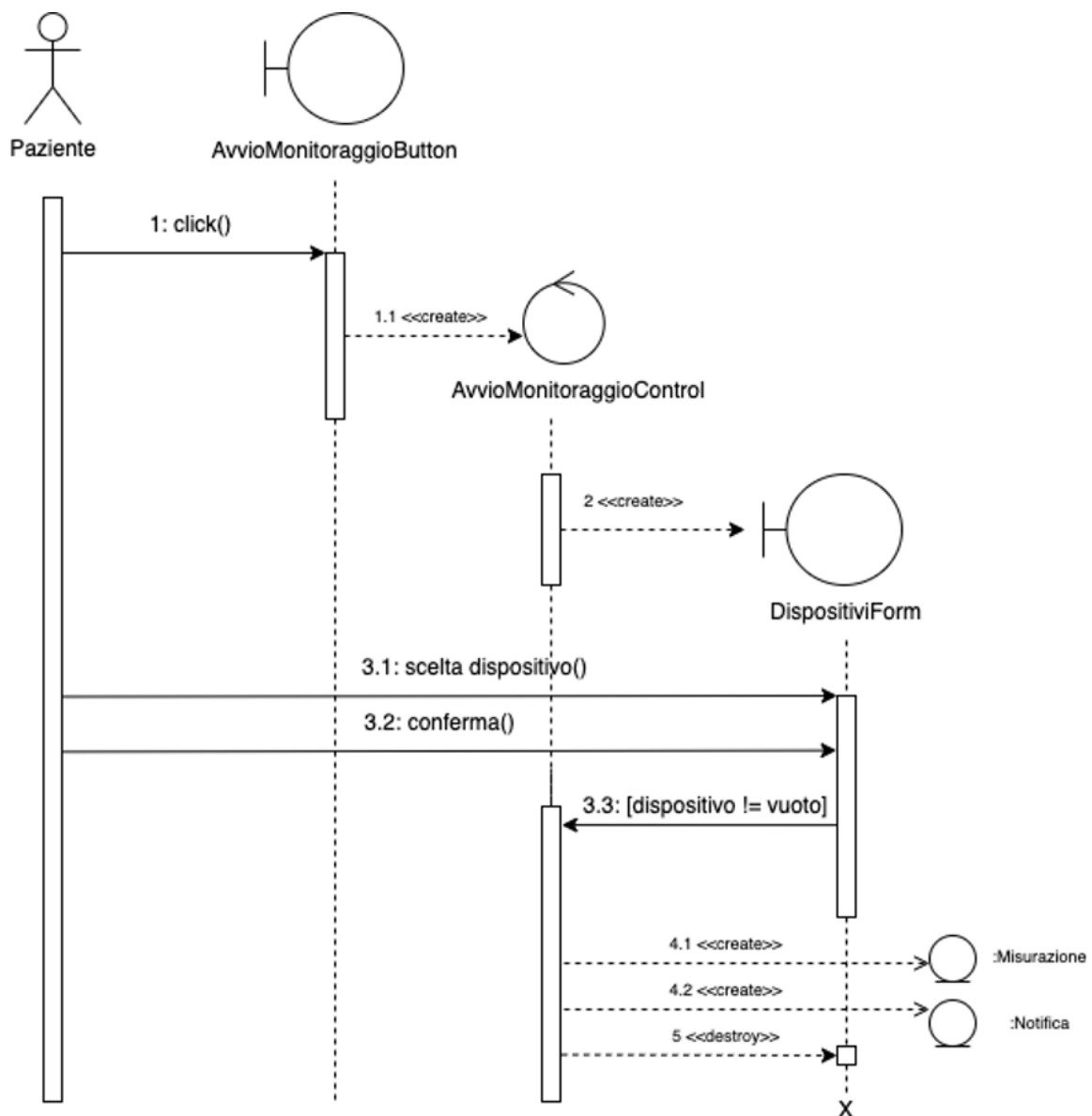


Figura 2.4: Class diagram

Sequence diagram - Avvio monitoraggio

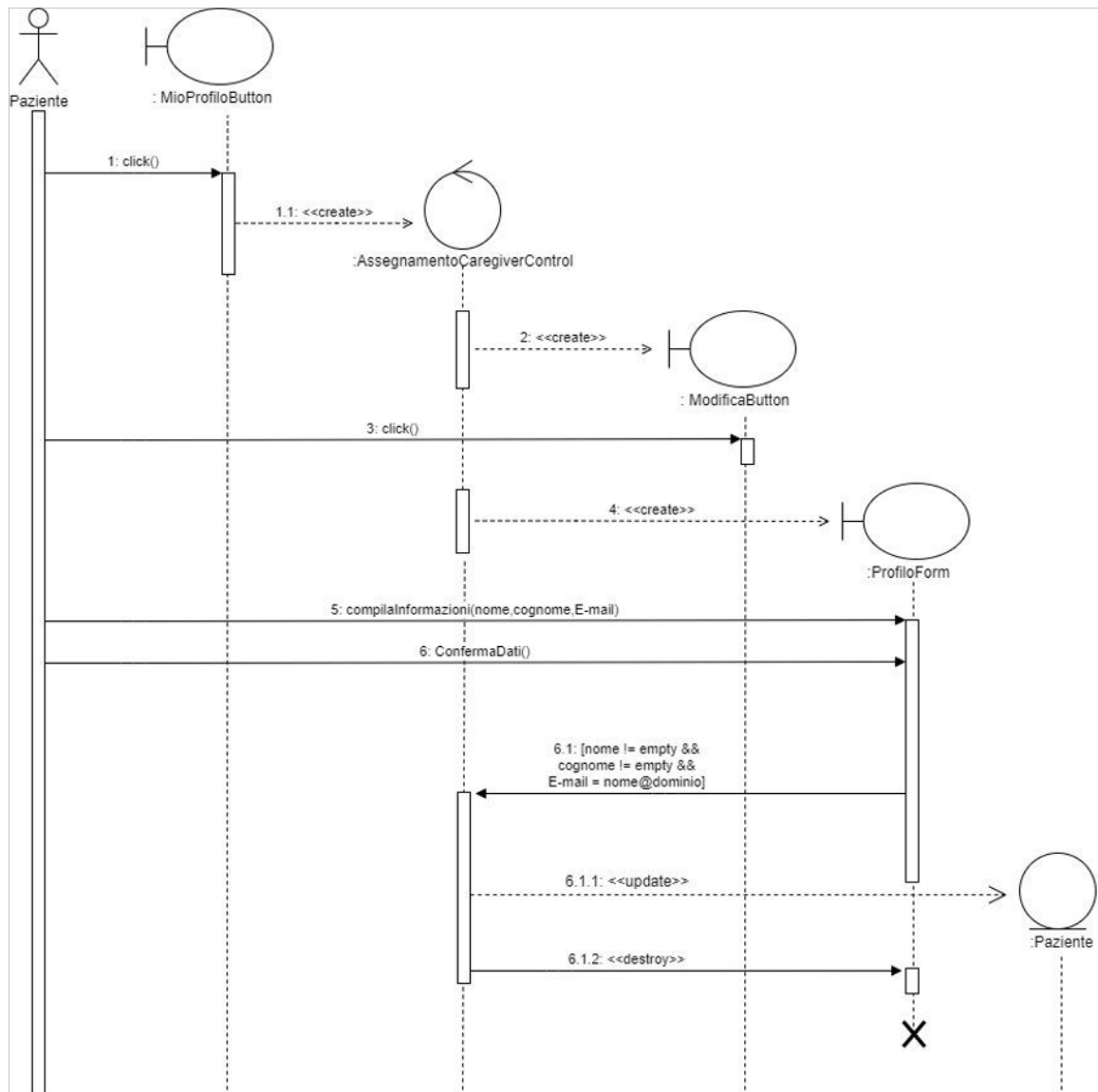


Figura 2.5: Sequence diagram - Assegnamento caregiver

Cambiamenti Proposti

3.1 Obiettivi CR_1

Attualmente, il sistema conserva già informazioni associate a ciascun paziente riguardanti il proprio caregiver, tuttavia il caregiver non è considerato un utente del sistema. L'obiettivo principale di questa CR è rendere il sistema *retrocompatibile*, iscrivendo automaticamente il caregiver utilizzando i dati già presenti sulla piattaforma. La migrazione dei dati viene quindi effettuata per consentire la creazione di questi nuovi utenti registrati a partire dai dati già esistenti e per permettere loro di utilizzare il sistema. L'obiettivo può essere suddiviso in due sotto-obiettivi:

- **Compatibilità retroattiva dei dati:** Garantire che il sistema funzioni senza problemi con i dati archiviati prima dei cambiamenti. Questo obiettivo si concentra sulla necessità di assicurare che i dati esistenti siano compatibili con il nuovo schema di dati.
- **Coerenza dei dati:** Assicurarsi che durante la migrazione dei dati non si verifichino perdite o corruzioni di dati. È essenziale garantire che i dati mantengano la loro integrità e precisione durante il processo di migrazione.

3.2 Obiettivi CR_2

L'implementazione di questa CR mira a potenziare le funzionalità e la completezza del sistema, rendendolo più utilizzabile. L'introduzione del 'caregiver' come nuovo attore all'interno del sistema apporta significativi miglioramenti, consentendo l'implementazione di funzionalità cruciali. Questa aggiunta estende la gamma di interazioni e possibilità all'interno della piattaforma, arricchendo le dinamiche esistenti e offrendo nuove opportunità di interazione.

Inizialmente, il 'caregiver' non era un utente riconosciuto dal sistema, limitando così le possibilità di utilizzo dello stesso, cioè, il caregiver non aveva effettivamente un ruolo, poteva soltanto ricevere delle e-mail inviate quando una misurazione del paziente era sballata, ma non poteva intervenire nè inviare lui stesso messaggi al medico. Grazie a questa CR, il sistema diventa effettivamente accessibile e utilizzabile dai caregiver dei pazienti, migliorando la qualità e l'efficienza del sistema.

3.3 Obiettivi CR_3

- **Miglioramento dell'esperienza utente:** rendere il sistema migliore dal punto di vista dell'interazione con l'utente, lavorando sulla coerenza delle funzionalità e la soddisfazione dell'utente.
- **Aggiunta di nuove funzionalità:** permettere al caregiver di avere una sezione dedicata con delle proprie funzionalità, in modo da avere sotto controllo le informazioni che riguardano i pazienti a cui è stato assegnato.

CAPITOLO 4

Impact Analysis

L'impact analysis è il processo che permette di identificare le componenti che verranno impattate da una change request. Questo processo è necessario per comprendere se porzioni critiche del sistema verranno intaccate dalla modifica, analizzare quali saranno le componenti del sistema che devono essere sottoposte a testing di regressione al termine delle modifiche e per comprenderne i costi.

Prima di procedere è necessario introdurre alcune definizioni fondamentali:

- La ***Starting Impact Set (SIS)*** è l'insieme iniziale di oggetti (componenti) che presumibilmente verranno impattati dalla CR.
- La ***Cadidate Impact Set (CIS)*** è l'insieme di oggetti (componenti) che si stima possano essere impattati secondo un certo approccio di impact analysis.
- La ***Discovered Impact Set (DIS)*** è l'insieme di nuovi oggetti (componenti), non contenuti nel CIS, che si scopre essere influenzato durante l'implementazione di una CR.
- La ***Actual Impact Set (AIS)*** è l'insieme di oggetti (componenti) che è effettivamente cambiato in seguito all'esecuzione di una CR.

- La *False Positive Impact Set (FPIS)* è l'insieme di oggetti (componenti) che si stima siano influenzati da un'implementazione di una CR ma che non sono effettivamente influenzati dalla CR. Precisamente, $FPIS = CIS \setminus AIS$.

4.1 CR_1 Migrazione dati

4.1.1 Starting Impact Set - SIS

Nello Starting Impact Set, sono stati individuate ed elencate di seguito le tabelle del database che si presume saranno modificate.

- Tabella "paziente"
- Tabella "utente_registrato"

4.1.2 Candidate Impact Set - CIS

Per il Candidate Impact Set è stato analizzato lo schema relazionale del database, a seguito dell'analisi sono stati definite le componenti che lo costituiscono.

- Tabella "paziente"
- Tabella "utente_registrato"

4.2 CR_2 Aggiunta nuova tipologia di utente

4.2.1 Starting Impact Set - SIS

Per l'aggiunta del nuovo utente al sistema abbiamo presumibilmente questi cambiamenti:

- Package *utils*:
 - *c15.dev.utils.Role*: Classe enum che attualmente considera solo Medico, Paziente e Admin.
- Package *registrazione*:

- *c15.dev.registrazione*: Modifica dell'intero package registrazione per aggiungere la pagina dedicata alla registrazione del caregiver.
- Package *entity*:
 - *c15.dev.model.entity.Paziente*: In questa classe, rimuoveremo i tre attributi che attualmente identificano staticamente il caregiver, sostituendoli con un riferimento all'entità *caregiver* che verrà creata successivamente.
- Package *gestioneUtente*:
 - *c15.dev.gestioneUtente.controller*: Modificare i metodi presenti nelle classi del package 'controller' che si occupano poi di chiamare il service per ricevere dati sui pazienti (e quindi caregiver) per poi fornirli al frontend.
 - *c15.dev.gestioneUtente.service*: Modificare i metodi presenti nei package 'service' che vengono chiamati dal controller e forniscono poi i dati necessari richiesti.
- Package *gestioneComunicazione*:
 - *c15.dev.gestioneComunicazione.service*: Modificare i metodi presenti nei package 'service' che riguardano la comunicazione.
- Package *model.listener*:
 - *c15.dev.model.listener.MisurazioneEventiListener*: Modificare i metodi eseguiti quando accadono eventi anomali sulle misurazioni.

4.2.2 Analisi delle dipendenze

L'approccio selezionato per la stima del Candidate Impact Set (CIS) è basato sull'analisi delle dipendenze del sistema. Per questo motivo, si è scelto di sviluppare una matrice di dipendenze, la quale si propone di offrire una rappresentazione visiva e strutturata delle connessioni esistenti tra le varie componenti del sistema in esame.

Ogni cella (i, j) della matrice descrive il numero di chiamate ai metodi della componente della riga i dalla componente della colonna j.

L'analisi di questa matrice consente di individuare gruppi di componenti strettamente dipendenti, le quali modifiche potrebbero avere un impatto significativo su altre parti del sistema. Questo processo semplifica così l'individuazione delle componenti da inserire all'interno del CIS. La matrice delle dipendenze è disponibile nell'appendice online¹.

Per la realizzazione di questa tabella, abbiamo preso in considerazione tutti i principali package e le dipendenze che essi hanno con gli altri package del progetto. In particolare, ci siamo concentrati sull'analisi delle dipendenze tra le diverse componenti, escludendo quelle interne alla stessa componente, cioè le dipendenze tra le classi all'interno del package. Questo approccio ci ha permesso di ottenere una visione chiara delle relazioni tra i diversi moduli del sistema, evidenziando le aree di maggiore interazione tra le componenti. Sono state calcolate come dipendenze gli usi, le chiamate ai metodi di una componente e le istanziazioni delle classi, mentre non sono state valutate le operazioni di *import*.

4.2.3 Candidate Impact Set - CIS

Dopo aver condotto le analisi descritte nella Sezione 4.2.2, è stato elaborato il Candidate Impact Set, il quale è descritto dettagliatamente nelle Tabelle 4.1 4.2. Durante questo processo, è stata analizzata la matrice delle dipendenze disponibile nell'appendice online¹, focalizzandosi sulla righe specifiche di ciascuna classe inclusi nel nostro System Impact Set (SIS). Questo approccio ci ha consentito di individuare con precisione le dipendenze dirette e indirette causate dal modulo coinvolto nel SIS e di selezionare i moduli rilevanti per il Candidate Impact Set. Il primo passo è stato analizzare la classe *entity.Paziente*, poiché la nostra nuova entità si basa sugli attributi esistenti di questa classe, sono state identificate tutte le dipendenze dirette e indirette associate a questa classe, successivamente, sono state analizzate anche tutte le dipendenze associate alle altre classi e gli altri package inseriti nel SIS (Sezione 4.2.1), quindi da tenere in considerazione. Nelle tabelle 4.1 e 4.2 sono elencate tutte le classi che si prevede saranno coinvolte nelle modifiche.

¹https://github.com/Annamariabasile/HeartCare_IGES/tree/CR2/docs/docsIGES

	Starting Impact Set	Candidate Impact Set
Paziente	X	X
GestioneMisurazioneController		X
GestioneUtenteController	X	X
GestioneVisitaController		X
MisurazioneEventiListener	X	X
GestioneComunicazioneServiceImpl	X	X
RegistrazioneController	X	X
GestioneVisitaServiceImpl		X
RegistrazioneServiceImpl	X	X
RegistrazioneService	X	X
ControlloMisurazione		X
DispositivoMedicoAdapter		X
GestioneMisurazioneServiceImpl		X
GestioneUtenteServiceImpl	X	X
GestioneUtenteService	X	X
PazienteDAO		X
MisurazioneCoagulazione		X
MisurazioneEnzimiCardiaci		X
MisurazioneGlicemia		X
MisurazioneHolterECG		X
MisurazionePressione		X
MisurazioneSaturazione		X

Tabella 4.1: Candidate Impact sets CR_2

	Starting Impact Set	Candidate Impact Set
Nota		X
Visita		X
Medico		X
Misurazione		X
DispositivoMedicoStub		X
GestioneComunicazioneController		X
GestioneComunicazioneService	X	X
NotaDAO		X
GestioneVisitaService		X
VisitaDAO		X
Indirizzo		X
MedicoDAO		X
IDispositivoMedico		X
MisurazioneDTO		X
MisurazioneDAO		X
Role	X	X
UtenteRegistrato		X
Admin		X
JwtService		X
AdminDAO		X
ApplicationConfig		X
UtenteRegistratoDAO		X
JwtAuthenticationConfig		X

Tabella 4.2: Candidate Impact sets CR_2

4.3 CR_3 Modifica GUI

4.3.1 Starting Impact Set - SIS

Per lo Starting Impact Set presumibilmente si hanno questi cambiamenti:

- *app.js*
- *ListaUtenti.js*
- *LoginForm.js*
- *ModificaDatiForm.js*
- *RegistrazioneForm.js*
- *VisitaCard.js*

4.3.2 Analisi delle dipendenze

L'approccio scelto per stimare il Candidate Impact Set (CIS) si basa sull'analisi delle dipendenze del sistema. A tal fine, è stata sviluppata una matrice delle dipendenze che offre una rappresentazione visiva e strutturata delle connessioni tra le varie componenti del sistema in esame.

L'analisi di questa matrice consente di individuare gruppi di componenti strettamente interconnesse, le cui modifiche potrebbero avere un impatto significativo su altre parti del sistema. La matrice delle dipendenze è disponibile nell'appendice online².

Per la realizzazione di questa matrice, sono state considerate le componenti appartenenti al front-end.

4.3.3 Candidate Impact Set - CIS

Dopo aver eseguito le analisi descritte nella Sezione 4.3.2, è stato sviluppato il Candidate Impact Set. Durante questo processo, è stata esaminata la matrice delle dipendenze disponibile nell'appendice online². Per ciascuna delle componenti individuate nello Starting Impact

Set (SIS), sono state identificate tutte le dipendenze dirette e indirette associate. Nella Tabella 4.3 sono elencate tutte le componenti che si prevede saranno coinvolte nella modifica.

	Starting Impact Set	Candidate Impact Set
App.js	X	X
ListaUtenti.js	X	X
LoginForm.js	X	X
ModificaDatiForm.js	X	X
RegistrazioneForm.js	X	X
VisitaCard.js	X	X
Login.js		X
index.js		X
App.test.js		X
profilo.js		X
Registrazione.js		X
HomeMedico.js		X
ListaVisite.js		X
Schedules.js		X

Tabella 4.3: Candidate Impact sets CR_3

²https://github.com/Annamariabasile/HeartCare_IGES/blob/CR3/app/depMatrixCr3.pdf

Implementazione dei Cambiamenti

5.1 Metodologia

Nelle sezioni a seguire sarà discussa l'implementazione delle CR proposte nei capitoli precedenti, in particolare sarà posta attenzione al processo implementativo e per ciascuna change request sarà definito l'Actual Impact Set (AIS), il Discovered Impact Set (DIS) e il False Positive Impact Set (FPIS). Inoltre, per ciascuna CR, sono state calcolate alcune metriche chiave per valutare l'efficacia dell'approccio di impact analysis. Queste metriche sono utili per valutare quanto l'approccio sia stato di supporto al processo di implementazione della CR. Per il calcolo delle metriche, è stato scelto di non considerare anche le componenti di test modificate e di cui era stata prevista la modifica per concentrare l'attenzione sulle componenti principali del progetto e per migliorare la comprensibilità dei risultati.

5.2 CR_1

5.2.1 Parte Implementativa

Per prima cosa abbiamo clonato il nostro database in produzione "heartcare" in un database di test "heartcare_test".

Successivamente abbiamo inserito nella tabella "utente_registrato" gli n caregiver presenti nella piattaforma. Lo abbiamo fatto utilizzando i dati relativi ai caregiver presenti nella tabella paziente, con alcuni valori predefiniti o generati dinamicamente per i campi mancanti. In particolare il ruolo di questi caregiver sarà quello di "CAREGIVER NON REGISTRATO". Il ruolo "CAREGIVER NON REGISTRATO" è un ruolo temporaneo associato solo ai caregiver creati dalla migrazione dei dati e verrà modificato in fase di primo accesso al sistema quando al caregiver saranno richieste le informazioni mancanti per completare la registrazione.

```
insert into utente_registrato (codice_fiscale, cognome,
data_di_nascita, email, genere, nome, numero_telefono,
password, ruolo)
SELECT
    SUBSTRING(UUID(), 1, 16) AS codice_fiscale,
    cognome_caregiver,
    '1970-01-01' AS data_di_nascita,
    email_caregiver,
    'M' AS genere,
    nome_caregiver,
    CONCAT('+39',LPAD(FLOOR(RAND()*10000000000),10,'0')) AS telefono,
    'password' AS password,
    'CAREGIVER_NON_REGISTRATO' AS stato
FROM paziente;
```

Passiamo poi alla creazione della tabella "caregiver" e all'inserimento degli id dei caregiver appena registrati.

```
CREATE TABLE caregiver (
```

```
    id BIGINT PRIMARY KEY,  
    FOREIGN KEY (id) REFERENCES utente_registrato(id)  
);
```

```
INSERT INTO caregiver (id)  
SELECT id  
FROM utente_registrato  
WHERE ruolo = 'CAREGIVER NON REGISTRATO';
```

A questo punto aggiungiamo un campo alla tabella paziente, tale campo consentirà l'associazione tra il paziente ed il suo caregiver.

```
ALTER TABLE paziente  
ADD COLUMN id_caregiver BIGINT,  
ADD CONSTRAINT fk_paziente_id_caregiver FOREIGN KEY (id_caregiver)  
REFERENCES utente_registrato(id);  
  
UPDATE paziente  
SET id_caregiver = (  
    SELECT id  
    FROM utente_registrato  
    WHERE email = paziente.email_caregiver  
);
```

Infine, dopo aver completato la migrazione dei dati, è possibile modificare lo schema della tabella "paziente" in modo da eliminare le colonne "nome_caregiver", "cognome_caregiver" ed "email_caregiver".

```
ALTER TABLE paziente  
DROP COLUMN nome_caregiver,  
DROP COLUMN cognome_caregiver,  
DROP COLUMN email_caregiver;
```


5.2.2 Discovered Impact Set - SIS

Non ci sono moduli, non presenti nel CIS, che sono stati poi modificati per l'implementazione della CR_1.

5.2.3 Actual Impact Set - AIS

Di seguito sono elencate le tabelle del database che costituiscono l'Actual Impact Set per la CR_1:

- Tabella "paziente"
- Tabella "utente_registrato"

5.2.4 False Positive Impact Set - FPIS

Non ci sono moduli che si pensava venissero impattati dalla CR_1 e che poi non sono stati impattati.

5.2.5 Metriche di processo

Le metriche di processo sono riportate di seguito:

$$\text{Recall} = \frac{|\text{CIS} \cap \text{AIS}|}{|\text{AIS}|} = \frac{2}{2} = 1$$

Abbiamo calcolato la metrica di Recall, che rappresenta la proporzione degli elementi correttamente identificati nel CIS che sono anche presenti nell'AIS. Il risultato suggerisce che il 100% degli elementi effettivamente influenzati dalle modifiche è stato correttamente individuato nel CIS.

$$\text{Precision} = \frac{|\text{CIS} \cap \text{AIS}|}{|\text{CIS}|} = \frac{2}{2} = 1$$

La Precision è stata valutata per misurare la proporzione degli elementi effettivamente modificati del CIS rispetto a tutti gli elementi inclusi nel CIS. Questo risultato, ovvero il 100%, indica che tutti gli elementi identificati nel CIS sono effettivamente influenzati dal cambiamento proposto.

5.3 CR_2

5.3.1 Parte Implementativa

Nel corso dell'implementazione della CR_2, sono state apportate diverse modifiche al sistema, andando a considerare gli obiettivi proposti nel Capitolo 3.2. Le modifiche eseguite sono riportate nell'elenco sottostante:

- **Creazione della classe Caregiver:** abbiamo creato la nuova entità *Caregiver* e la relativa classe *CaregiverDAO* per recuperare i dati dal database.
- **Modifica della classe paziente:** sono stati eliminati dalla classe paziente gli attributi "nome_caregiver", "cognome_caregiver" ed "email_caregiver". Inoltre è stato aggiunto il riferimento al relativo caregiver.
- **Modifica funzionalità di Login:** è stato modificato il metodo di login affinché anche le nuove tipologie di utenti potessero eseguire tale funzionalità.
- **Modifica funzionalità di registrazione:** è stato aggiunto il metodo per la registrazione di un caregiver.
- **Modifica funzionalità gestione del caregiver:** sono stati realizzati i metodi per assegnazione, modifica e rimozione di un caregiver ad un paziente.
- **Aggiunta possibilità di invio note per il caregiver:** sono stati aggiunti i metodi che permettono al caregiver di inviare delle note e visualizzare le note dei propri pazienti.
- **Aggiunta possibilità di visualizzazione FSE:** aggiunti metodi per permettere al caregiver di visualizzare il fascicolo sanitario elettronico (FSE) e le visite il programma dei propri pazienti.

5.3.2 Discovered Impact Set - SIS

Dopo aver implementato le modifiche delineate nella Sezione 5.3.1, è stata derivato il Discovered Impact Set. Questo processo ha rilevato che sono state apportate modifiche alle componenti non precedentemente elencate nel CIS delle Tabelle 4.1 e 4.2, di conseguenza, i moduli risultanti nel DIS sono:

- *SecurityConfig*: In questa classe è stato aggiunto l'endpoint per registrare un caregiver.
- *GestioneMisurazioneService*: In questa classe è stato aggiunta la firma del metodo *getAllMisurazioniByCaregiver* che restituisce tutte le misurazioni dei pazienti di un caregiver.

5.3.3 Actual Impact Set - AIS

Nella Tabelle 5.1 5.2, sono state riportate le componenti modificate durante lo sviluppo della CR_2.

	Candidate Impact Set	Actual Impact Set
Paziente	X	X
GestioneMisurazioneController	X	X
GestioneUtenteController	X	X
GestioneVisitaController	X	X
MisurazioneEventiListener	X	X
GestioneComunicazioneServiceImpl	X	X
RegistrazioneController	X	X
GestioneVisitaServiceImpl	X	
RegistrazioneServiceImpl	X	X
RegistrazioneService	X	X
ControlloMisurazione	X	
DispositivoMedicoAdapter	X	
GestioneMisurazioneServiceImpl	X	X
GestioneUtenteServiceImpl	X	X
GestioneUtenteService	X	X
PazienteDAO	X	
MisurazioneCoagulazione	X	
MisurazioneEnzimiCardiaci	X	
MisurazioneGlicemia	X	
MisurazioneHolterECG	X	
MisurazionePressione	X	
MisurazioneSaturazione	X	
Nota	X	
Visita	X	

Tabella 5.1: Actual Impact sets CR_2

	Candidate Impact Set	Actual Impact Set
Medico	X	
Misurazione	X	
DispositivoMedicoStub	X	
GestioneComunicazioneController	X	X
GestioneComunicazioneService	X	X
NotaDAO	X	X
GestioneVisitaService	X	
VisitaDAO	X	
Indirizzo	X	
MedicoDAO	X	
IDispositivoMedico	X	
MisurazioneDTO	X	
MisurazioneDAO	X	
Role	X	X
UtenteRegistrato	X	
Admin	X	
JwtService	X	
AdminDAO	X	
ApplicationConfig	X	
UtenteRegistratoDAO	X	X
JwtAuthenticationConfig	X	
SecurityConfig		X
GestioneMisurazioneService		X
Totale	45	19

Tabella 5.2: Actual Impact sets CR_2

5.3.4 False Positive Impact Set - FPIS

Dall'analisi delle Tabelle 5.1 e 5.2, è possibile estrarre il False Positive Impact Set. Le componenti che fanno parte del FPIS sono:

- Le classi per la **gestione delle misurazioni**: *MisurazioneCoagulazione, MisurazioneEnzimi-Cardiaci, MisurazioneGlicemia, MisurazioneHolterECG, MisurazionePressione, Misurazione-Saturazione, Misurazione, ControlloMisurazione, MisurazioneDTO*;
- Le **entity**: *Nota, Visita, Medico, Indirizzo, Admin, UtenteRegistrato*;
- Le classi DAO: *VisitaDAO, MedicoDAO, MisurazioneDAO, PazienteDAO, AdminDAO*;
- Le classi per la **gestione dei dispositivi medici**: *DispositivoMedicoAdapter, DispositivoMedicoStub, IDispositivoMedico*;
- Le classi per la **gestione delle visite**: *GestioneVisitaService, GestioneVisitaServiceImpl*;
- Le classi per la **gestione dell'autenticazione e della sicurezza**: *JWTService, JwtAuthenticationConfig, ApplicationConfig*.

5.3.5 Metriche di processo

Le metriche di processo sono riportate di seguito:

$$\text{Recall} = \frac{|\text{CIS} \cap \text{AIS}|}{|\text{AIS}|} = \frac{17}{19} = 0.8947$$

Abbiamo calcolato la metrica di Recall, che rappresenta la proporzione degli elementi correttamente identificati nel CIS che sono anche presenti nell'AIS. Il risultato suggerisce che il 89.47% degli elementi effettivamente influenzati dalle modifiche è stato correttamente individuato nel CIS (ovvero 17 componenti su 19).

$$\text{Precision} = \frac{|\text{CIS} \cap \text{AIS}|}{|\text{CIS}|} = \frac{17}{45} = 0.3778$$

La Precision è stata valutata per misurare la proporzione degli elementi effettivamente modificati del CIS rispetto a tutti gli elementi inclusi nel CIS. Questo risultato, ovvero il 37.78%, indica che non tutti gli elementi identificati nel CIS sono effettivamente influenzati dal cambiamento proposto.

5.4 CR_3

5.4.1 Parte Implementativa

L'implementazione della CR_3 ha comportato diverse modifiche alle componenti del nostro sistema, in particolare nell'elenco seguente vengono riportate i cambiamenti fatti al sistema:

- **Login Caregiver:** aggiunta funzionalità di login per l'utente con ruolo "CAREGIVER".
- **Creazione HomePage Caregiver:** aggiunta pagine home page e del menu per gli utenti con ruolo "CAREGIVER".
- **Visualizzazione pazienti di un caregiver:** aggiunta funzionalità di visualizzazione dei pazienti di un caregiver.
- **Visualizzazione visite in programma dei pazienti di un caregiver:** aggiunta funzionalità visualizzazione delle visite dei pazienti del caregiver.
- **Invio note caregiver:** aggiunta funzionalità di lettura e invio note da parte del caregiver.
- **Creazione pagine per i caregiver:** creazione delle pagine accessibili dagli utenti con ruolo "CAREGIVER" in cui poter visualizzare i propri pazienti, le visite in programma dei propri pazienti e il profilo dove poter modificare i suoi dati.
- **Aggiunta e modifica caregiver:** modificata funzionalità di aggiunta e modifica del caregiver da parte del paziente.
- **Registrazione caregiver:** creazione pagina registrazione per utenti con ruolo "CAREGIVER_NON_REGISTRATO".
- **Visualizzazione Fascicolo Sanitario Elettronico:** funzionalità che permette al caregiver di visualizzare il fascicolo sanitario elettronico dei propri pazienti.

5.4.2 Discovered Impact Set - SIS

Dopo aver implementato le modifiche descritte nella Sezione 5.4.1, è stato derivato il Discovered Impact Set. Questo processo ha rivelato che sono state effettuate delle modifiche

alle componenti precedentemente elencate nel CIS della Tabella 4.3. Di conseguenza, i moduli risultanti nel DIS sono:

- *GestioneComunicazioneController*
- *GestioneUtenteController*
- *GestioneUtenteService*
- *GestioneUtenteServiceImpl*
- *RegistrazioneController*
- *RegistrazioneServiceImpl*
- *SecurityConfig*
- *NoteContainerPaziente*
- *NoteContainer*
- *CardPaziente*

5.4.3 Actual Impact Set - AIS

Nella tabella 5.3 sono riportate le componenti modificate durante lo sviluppo della CR_3.

5.4.4 False Positive Impact Set - FPIS

Dall'analisi della tabella 4.3, è possibile estrarre il False Positive Impact Set. Le componenti che fanno parte del FPIS sono:

- *Login.js*
- *index.js*
- *profilo.js*
- *Registrazione.js*

5.4.5 Metriche di processo

Le metriche di processo sono riportate di seguito:

$$\mathbf{Recall} = \frac{|CIS \cap AIS|}{|AIS|} = \frac{8}{18} = 44.44\%$$

$$\mathbf{Precision} = \frac{|CIS \cap AIS|}{|CIS|} = \frac{8}{13} = 61.54\%$$

	Candidate Impact Set	Actual Impact Set
GestioneComunicazioneController.java		X
GestioneUtenteController.java		X
GestioneUtenteService.java		X
GestioneUtenteServiceImpl.java		X
RegistrazioneController.java		X
RegistrazioneServiceImpl.java		X
SecurityConfig.java		X
App.js	X	X
ListaUtenti.js	X	X
LoginForm.js	X	X
ModificaDatiForm.js	X	X
RegistrazioneForm.js	X	
VisitaCard.js	X	X
Login.js	X	
index.js	X	
profilo.js	X	
Registrazione.js	X	
HomeMedico.js	X	X
ListaVisite.js	X	X
Schedules.js	X	X
NoteContainer.js		X
NoteContainerPaziente.js		X
CardPaziente.js		X
Totale	13	18

Tabella 5.3: Actual Impact sets CR_3

6.1 Sintesi delle Modifiche

Le Change Requests implementate hanno riguardato vari aspetti del sistema per migliorarne l'efficienza, l'accessibilità e la funzionalità. La prima Change Request ha coinvolto la migrazione dei dati, con significative modifiche alla struttura del database e alla gestione dei dati, garantendo una migrazione sicura e senza perdite di informazioni. La seconda modifica ha introdotto una nuova tipologia di utente, il caregiver, per supportare meglio i pazienti, richiedendo la creazione di nuove pagine e funzionalità dedicate, come la visualizzazione delle visite programmate e la gestione delle note tra caregiver e pazienti. Infine, l'ultima Change Request ha comportato una revisione dell'interfaccia utente, rendendo il sistema più intuitivo e accessibile, con modifiche al design delle pagine principali e alle modalità di interazione degli utenti con il sistema.

6.2 Lesson learned

Durante la realizzazione di questo progetto, le lezioni apprese sono le seguenti:

- **Utilizzo strutturato di GitHub.** Abbiamo imparato a sfruttare alcune funzionalità di GitHub, tra cui la gestione dei branch, l'uso efficace di issues e pull request. Questo ha permesso di gestire il progetto avendo una maggiore visibilità dello stato delle attività e permettendo una pianificazione dei task.
- **Importanza del pair programming.** Abbiamo sperimentato i vantaggi del pair programming, riconoscendo la sua importanza nella risoluzione efficiente dei problemi, nell'apprendimento reciproco e nella condivisione delle conoscenze tra i membri del team di sviluppo.

Queste lezioni apprese hanno arricchito la nostra esperienza di sviluppo e di lavoro in team. Siamo riusciti a consolidare le nostre capacità nella gestione di progetti software complessi e siamo determinati a mettere in pratica queste lezioni in futuri progetti.