

Università degli studi di Verona

Documentazione progetto Ingegneria del software

Traccia: “Terapia intensiva”

Davide Tiso, VR4 6, informatica

Sommario

Requisiti ed interazione utente-sistema	3
Casi d'uso	3
Utente non autenticato:	4
Utenti autenticati:	5
Personale infermieristico.....	6
Medici	9
Primario	11
Diagrammi delle attività	12
Processo di sviluppo:.....	16
Architettura ed implementazione del sistema:	17
Architettura Client-Server:	17
Server: funzionalità dell'Application Server	19
Client: Sistema informativo gestionale.....	23
Architettura adottata (MVC)	23
View-Controller:	24
Model: classe DataBase.....	25
Diagrammi di sequenza:	26
Test	31
Sistema monitoraggio: generazione segnali	31
Test sviluppatore: sistema gestionale.....	32

Requisiti ed interazione utente-sistema

Il sistema è stato progettato ad uso esclusivo da parte del personale sanitario del reparto di terapia intensiva. Il software client sarà installato solo sui dispositivi autorizzati dall'amministrazione. Ogni utente non autenticato del sistema avrà un accesso limitato al software. Il personale è suddiviso in medici, infermieri e primario del reparto. Ogni utente autorizzato avrà a disposizione delle credenziali (fornite dall'amministrazione) per l'accesso al software completo con funzionalità pertinenti al proprio ruolo all'interno del reparto di terapia intensiva. La possibilità di accedere al sistema di monitoraggio real-time dei parametri vitali è messa a disposizione di tutti gli utenti autenticati così come la possibilità di generare un ricovero. Inoltre, in questa fase iniziale si vuole specificare che il primario nel sistema eredita tutte le risorse messe a disposizione al medico, può inoltre usufruire di ulteriori funzionalità per la gestione delle cartelle cliniche e lo storico di queste ultime.

Casi d'uso



Utente non autenticato:

All'avvio dell'applicativo l'utente verrà introdotto a una schermata standard non completa di tutte le funzionalità implementate, per avere accesso all'intero applicativo l'utente prima dovrà effettuare l'autenticazione. Le operazioni riservate all'utente non autenticato sono principalmente due: visualizzazione dei parametri vitali registrati negli ultimi quindici minuti per ogni paziente ricoverato, possibilità di effettuare l'autenticazione attraverso apposita schermata.

Visualizza parametri paziente:

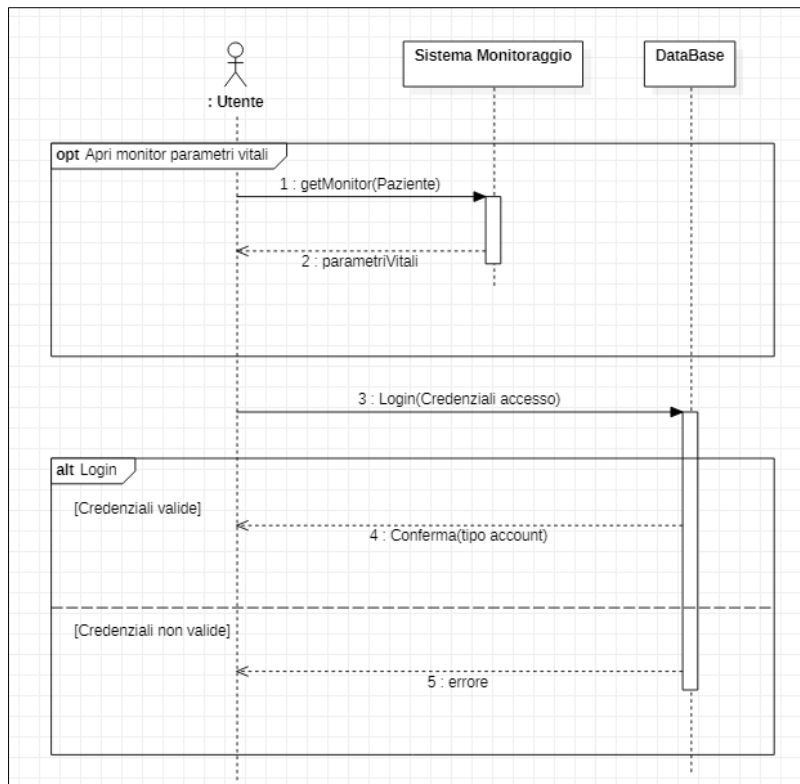
Ogni utente non autenticato deve avere la possibilità di visualizzare l'andamento dei parametri vitali dei pazienti ricoverati registrati negli ultimi quindici minuti. Per fare ciò attraverso un menù a tendina della schermata principale, gli basterà selezionare uno dei pazienti ricoverati.

Attori: <i>Utente non loggato</i>
Precondizioni: Il server deve essere in funzione
Passi: <ol style="list-style-type: none">1. <i>L'utente visualizza l'interfaccia di base</i>2. <i>Se l'utente seleziona il paziente interessato</i><ol style="list-style-type: none">a. <i>Viene visualizzato il grafico relativo all'andamento dei parametri vitali degli ultimi quindici minuti</i>

Login:

Una volta avviato il software, in qualsiasi momento, l'utente può decidere di autenticarsi attraverso le proprie credenziali. Le credenziali sono memorizzate nel database e gestite dall'amministratore del software. In caso di autenticazione fallita il programma verrà chiuso in automatico.

Attori: <i>Utente non loggato</i>
Precondizioni: L'utente deve possedere le credenziali d'accesso
Passi: <ol style="list-style-type: none">1. <i>L'utente seleziona l'opzione d'accesso</i>2. <i>Fintantoché viene visualizzata la schermata di accesso</i><ol style="list-style-type: none">2.1. <i>Se l'utente inserisce le credenziali attendibili</i><ol style="list-style-type: none">2.1.1. <i>Viene effettuata l'autenticazione</i>2.1.2. <i>Viene visualizzata la schermata principale riservata agli utenti autenticati</i>2.2. <i>Se l'utente inserisce credenziali invalide</i><ol style="list-style-type: none">2.2.1. <i>Viene visualizzato un messaggio d'errore</i>
Post condizioni: L'utente viene introdotto alla schermata principale del software completo



Utenti autenticati:

Visualizza monitor real-time e parametri vitali registrati:

Per consultare il monitor real-time dei parametri vitali e i parametri registrati nelle ultime due ore di un determinato paziente, l'utente deve innanzitutto autenticarsi. Una volta effettuato l'accesso verrà introdotto all'interfaccia di base dove potrà selezionare uno dei pazienti ricoverati per accedere al monitor real-time del sistema di monitoraggio. Sarà inoltre possibile visualizzare uno dei grafici relativi all'andamento dei parametri vitali registrati nelle due ore precedenti alla richiesta di visualizzazione.

Attori: Infermiere, Medico, Primario
Precondizioni: L'utente deve essersi autenticato
Passi: <ol style="list-style-type: none"> 1. <i>L'utente seleziona uno dei pazienti ricoverati attraverso l'interfaccia</i> 2. <i>Se l'utente ha selezionato correttamente un paziente ricoverato e il paziente è collegato ai macchinari di rilevazione dei parametri vitali</i> <ol style="list-style-type: none"> 2.1. <i>Visualizzazione del monitor real-time dei parametri vitali</i> 2.2. <i>Visualizzazione del grafico relativo all'andamento dei parametri vitali registrati nelle ultime due ore</i> 2.3. <i>Fintantoché l'utente rimane in questa schermata</i> <ol style="list-style-type: none"> 2.3.1. <i>I dati relativi all'andamento dei parametri vitali vengono aggiornati</i>

Personale infermieristico

Dopo essersi autenticato, l'infermiere viene introdotto alla schermata riservata agli utenti autenticati, che permette di visualizzare i monitor dei pazienti e le somministrazioni effettuate. Inoltre, permette di inserire le somministrazioni effettuate e registrare nuovi pazienti i cui dati anagrafici non sono stati ancora salvati nel sistema.

Somministra farmaci:

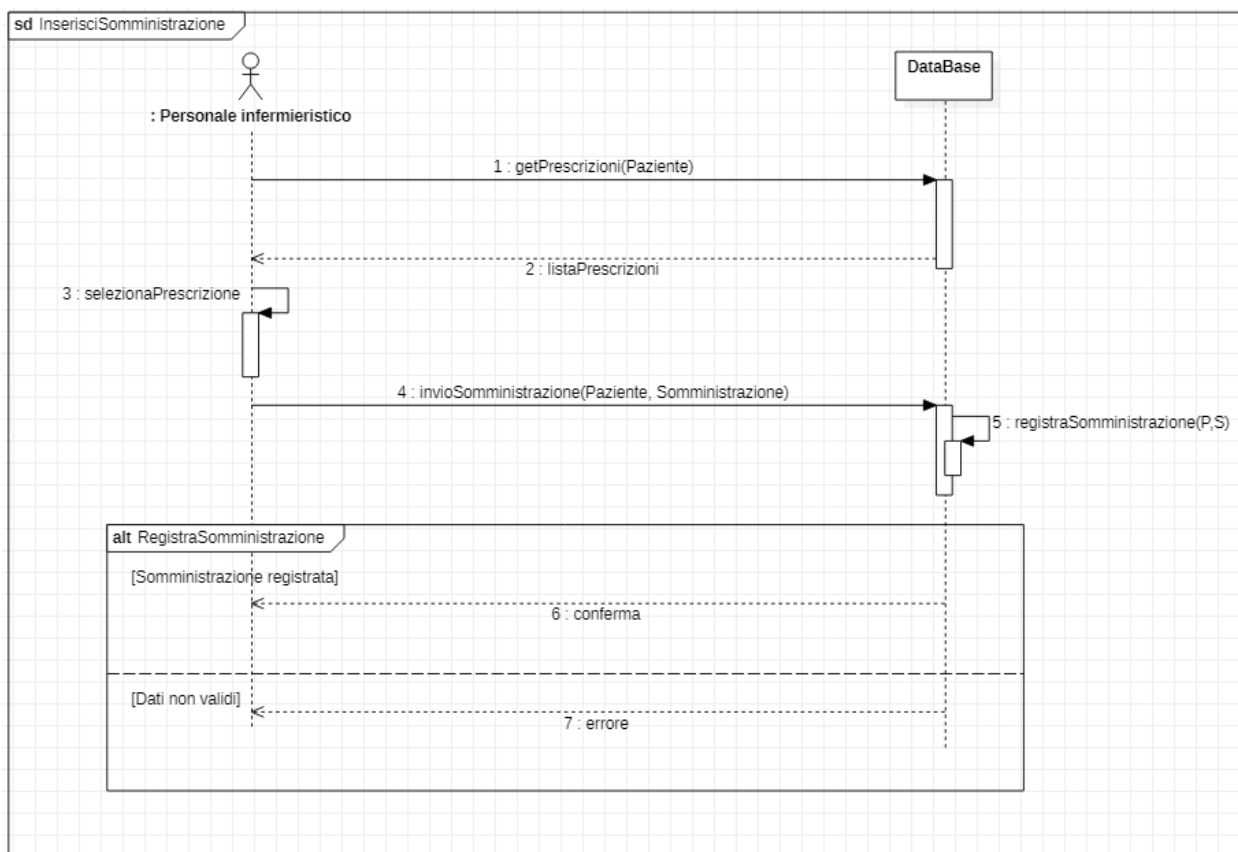
L'infermiere può registrare la somministrazione di un farmaco ad un paziente ricoverato selezionando l'apposito bottone nella stessa interfaccia contenente i parametri vitali del paziente. In questo modo non possono verificarsi errori di somministrazione, in quanto si può accedere a questa opzione solo se la cartella clinica del paziente è ancora aperta, e quindi è ancora in stato di ricovero. Le somministrazioni, per essere registrate, devono essere state prescritte da un medico. I dati relativi alle prescrizioni vengono automaticamente caricati e non possono essere modificati dall'infermiere che può solo modificare la dose somministrata e aggiungere note sullo stato del paziente.

Attori: Infermiere
Precondizioni: <ol style="list-style-type: none">1. L'infermiere ha selezionato dall'interfaccia base un paziente2. Deve essere stata inserita una prescrizione da parte di un medico per poter effettuare la somministrazione
Passi: <ol style="list-style-type: none">1. <i>L'infermiere accede all'interfaccia adibita alla somministrazione del farmaco</i>2. <i>L'infermiere seleziona la prescrizione che intende somministrare</i>3. <i>L'infermiere inserisce eventuali dati relativi alla somministrazione del farmaco e/o note sullo stato del paziente al momento della somministrazione</i>4. <i>L'infermiere conferma la prescrizione</i>
Post condizioni: La somministrazione viene registrata

Registra nuovo paziente:

Un paziente per essere ricoverato deve essere registrato nel database, ovvero i suoi dati anagrafici devono essere opportunamente memorizzati nel sistema. Per questo motivo l'infermiere ha la possibilità di registrare nel sistema nuovi pazienti e i relativi dati anagrafici.

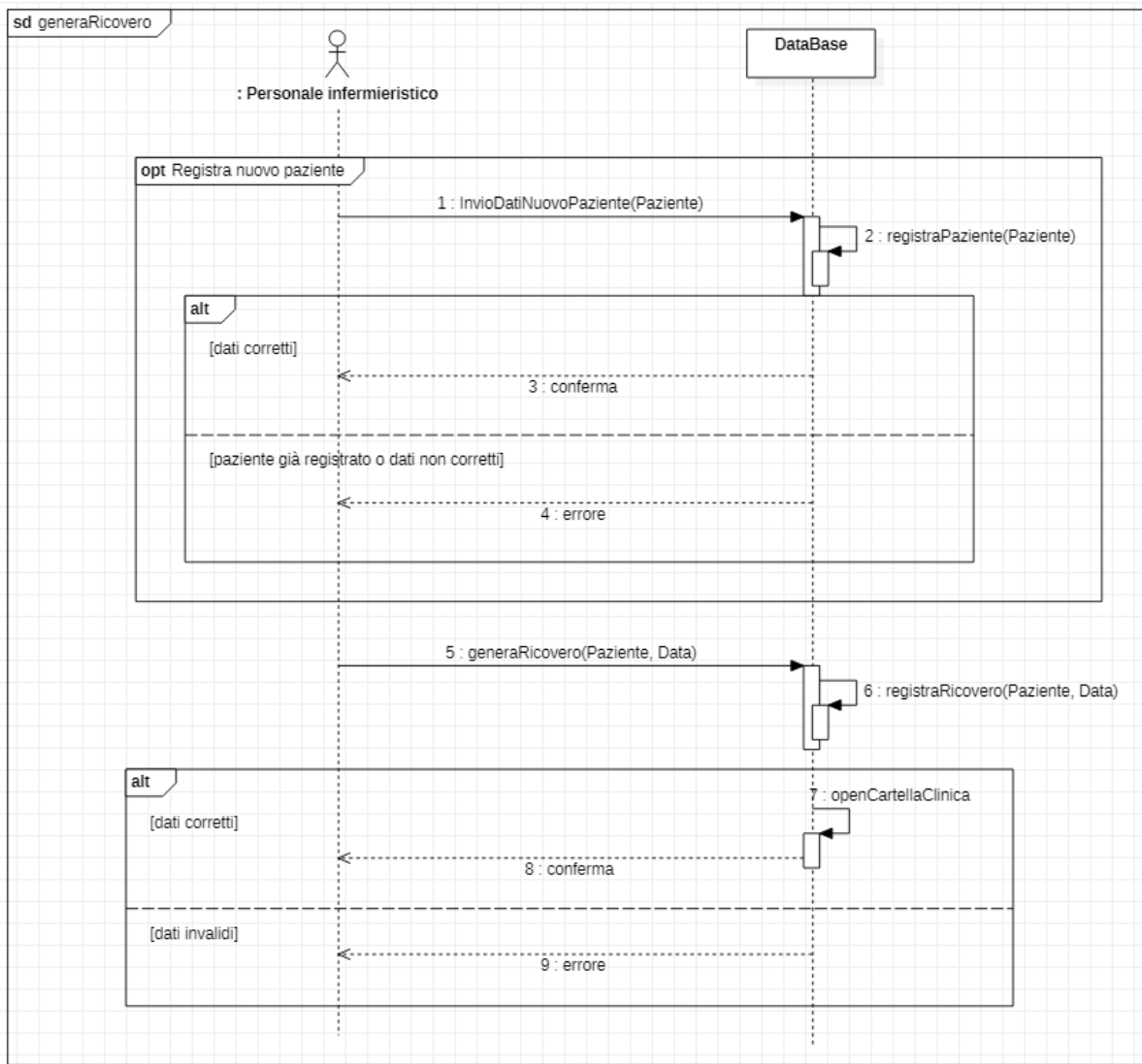
Attori: Infermiere
Precondizioni: Se si vuole registrare un nuovo paziente non dovrà essere già memorizzato nel sistema
Passi: <ol style="list-style-type: none">1. <i>L'infermiere accede all'interfaccia adibita alla generazione di un nuovo ricovero</i>2. <i>L'infermiere accede all'interfaccia per la registrazione al sistema di un paziente</i>3. <i>L'infermiere inserisce correttamente i dati anagrafici del paziente</i>4. <i>L'infermiere conferma la registrazione</i>
Post condizioni: Il paziente viene registrato



Genera ricovero:

L'infermiere è adibito alla generazione del ricovero, deve selezionare uno dei pazienti registrati nel sistema, inserire la data di inizio del ricovero e in tal modo in automatico il sistema genera la cartella clinica del ricovero. I dati del paziente, se selezionato correttamente, non possono essere sbagliati, in quanto sono dati forniti dal sistema.

Attori: Infermiere
Precondizioni: Il paziente da ricoverare deve essere registrato nel sistema
Passi: <ol style="list-style-type: none"> 1. <i>L'infermiere accede all'interfaccia adibita alla generazione di un nuovo ricovero</i> 2. <ol style="list-style-type: none"> a. <i>L'infermiere ricerca il paziente tramite l'interfaccia</i> b. <i>L'infermiere seleziona il paziente da ricoverare</i> c. <i>L'infermiere inserisce la data d'inizio del ricovero</i> 3. <i>L'infermiere conferma il ricovero</i>
Post condizioni: La cartella clinica del ricovero viene registrata nel sistema



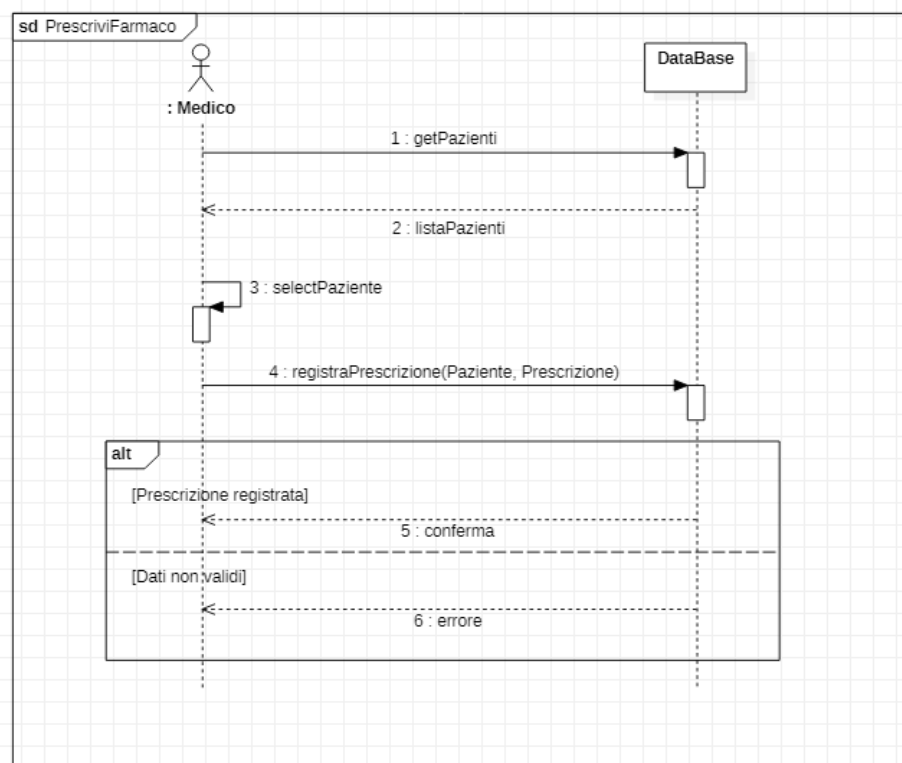
Medici

Dopo aver effettuato l'autenticazione il medico viene introdotto all'interfaccia di base, a differenza del personale infermieristico il medico riceverà uno o più allarmi nel caso in cui un paziente sia in una situazione ritenuta "critica". Come per gli infermieri potrà consultare il monitor real-time dell'andamento dei parametri vitali e i parametri vitali registrati nelle ultime due ore. Sarà anche lui stesso in grado di generare un ricovero e inoltre, per ogni ricovero, sarà in grado di prescrivere farmaci e/o terapie al paziente ricoverato.

Prescrivi farmaco:

Per inserire nel sistema una prescrizione il medico dovrà selezionare un ricovero, una volta introdotto all'interfaccia dell'andamento dei parametri vitali, potrà selezionare l'opzione di registrazione di una nuova prescrizione. Dopo aver selezionato il farmaco, inserito correttamente i dati e dopo averla confermata, la prescrizione sarà registrata nel sistema.

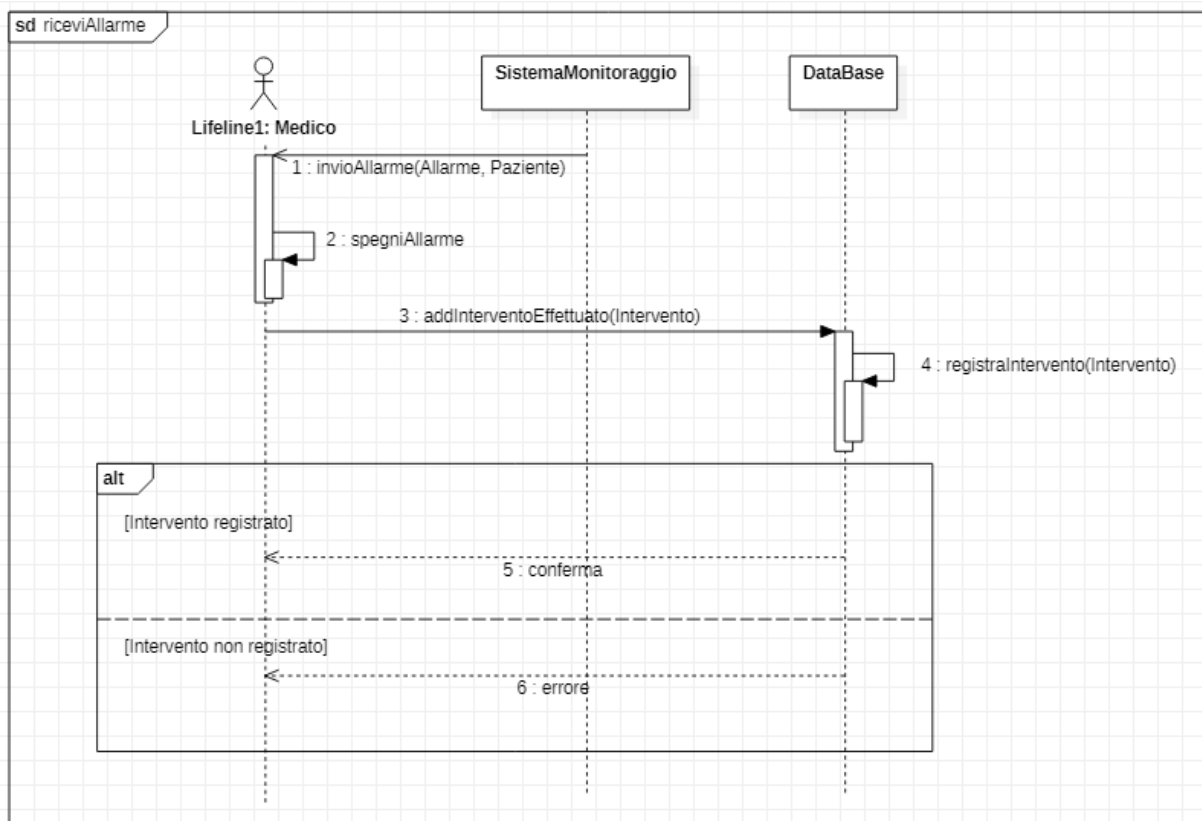
Attori: Medico
Precondizioni: Il medico deve essersi autenticato
Passi: <ol style="list-style-type: none">1. <i>Seleziona uno dei pazienti ricoverati attraverso l'interfaccia</i>2. <i>Il medico viene introdotto all'interfaccia di monitoraggio del paziente</i>3. <i>Seleziona l'opzione relativa alla generazione di una prescrizione</i>4. <i>Il medico viene introdotto alla interfaccia di generazione della prescrizione</i><ol style="list-style-type: none">4.1. <i>Seleziona un farmaco</i>4.2. <i>Ricerca un farmaco</i>5. <i>Inserisce i dati della prescrizione</i>6. <i>Il medico conferma la prescrizione</i>
Post condizioni: la prescrizione viene registrata



Ricevi allarme:

Dopo essersi autenticato il medico in automatico dal sistema di monitoraggio dei parametri vitali riceverà allarmi riguardanti l'andamento dei parametri vitali dei pazienti ricoverati. Si possono rilevare diversi allarmi di diverso tipo in base alla gravità e il medico è tenuto a fermare l'allarme entro un tempo limite stabilito dalle politiche dell'ospedale. Una volta stoppato l'allarme, il medico è tenuto a scrivere la descrizione dettagliata dell'intervento effettuato.

Attori: Medico
Precondizioni: Il medico deve essersi autenticato
Passi: <ol style="list-style-type: none">1. Il medico viene introdotto all'interfaccia iniziale2. Se il sistema di monitoraggio registra parametri vitali critici per il paziente<ol style="list-style-type: none">2.1. Il medico viene introdotto all'interfaccia di gestione dell'allarme2.2. Il medico spegne l'allarme2.3. Il medico inserisce i dati relativi all'intervento effettuato sul paziente
Post condizioni: l'intervento effettuato viene registrato



Primario

Il primario, come da diagramma dei casi d'uso, altro non è che una specializzazione di medico; quindi, tutte le funzionalità messe a disposizione del medico faranno parte del set di funzionalità per il primario. In aggiunta avrà la possibilità di consultare lo storico delle cartelle cliniche dei ricoveri pregressi e potrà chiudere la cartella clinica di un ricovero.

Consulta archivio:

Per consultare l'archivio relativo alle cartelle cliniche di ricoveri pregressi, il primario, dopo essersi autenticato, selezionerà nella schermata iniziale l'opposta opzione per interfacciarsi con il gestionale delle cartelle cliniche.

Attori: Primario
Precondizioni: Il primario deve essersi autenticato
Passi: <ol style="list-style-type: none">1. <i>Il primario seleziona l'opzione "consulta archivio"</i>2. <i>Il primario viene introdotto alla schermata per consultare le cartelle cliniche</i>3. <i>Seleziona l'opzione relativa alla generazione di una prescrizione</i>4. <i>Il primario viene introdotto alla interfaccia di generazione della prescrizione</i><ol style="list-style-type: none">4.1. <i>Seleziona una cartella clinica</i>4.2. <i>Ricerca una cartella clinica</i>5. <i>Il primario visualizza la lettera di dimissioni stilata per la chiusura del ricovero</i>

Chiusura ricovero:

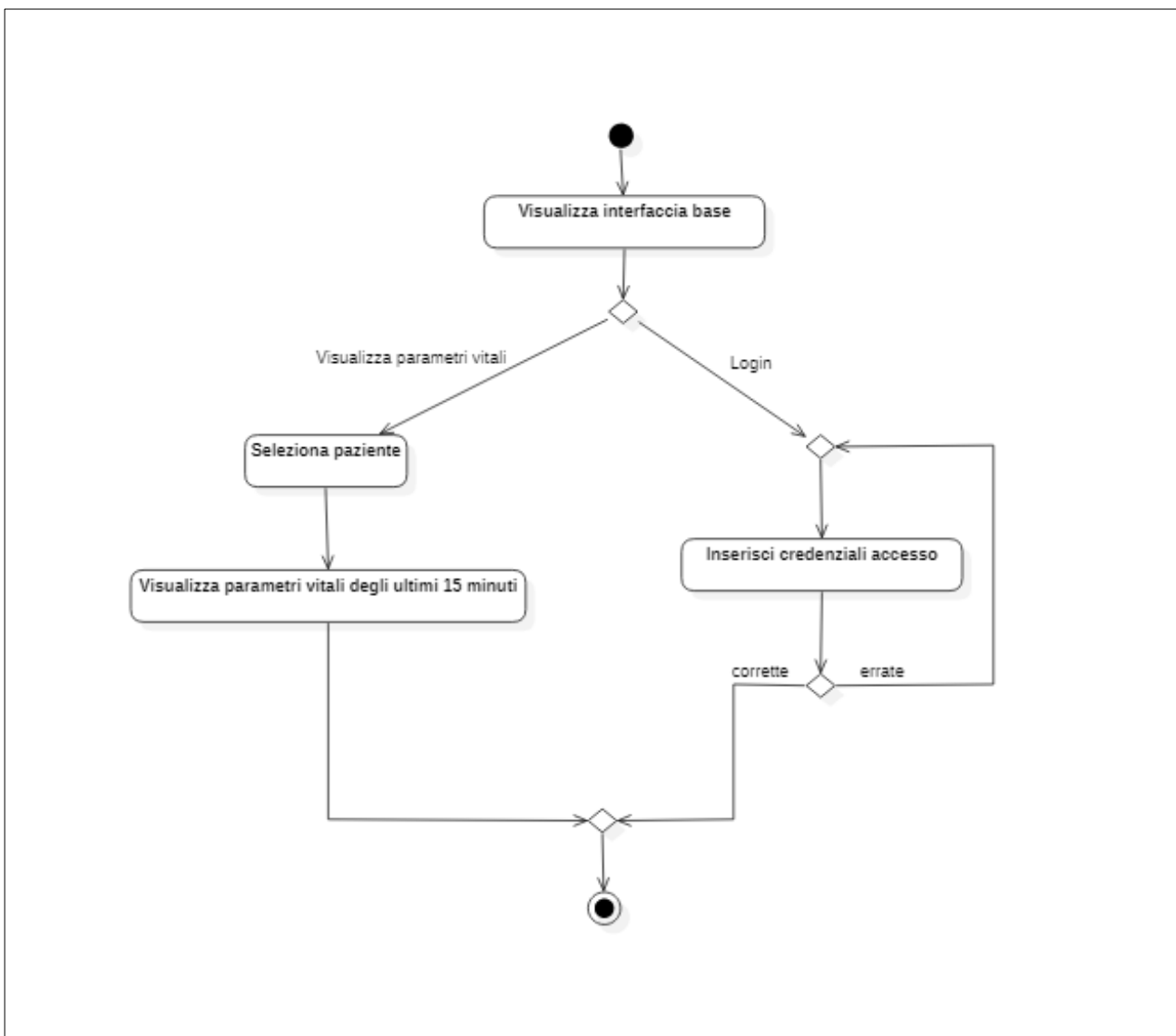
Per poter dimettere un paziente il primario deve stilare l'opportuna lettera di dimissioni. La data viene inserita automaticamente, i restanti campi sono a discrezione del primario. Una copia della lettera di dimissioni verrà salvata in formato pdf nel dispositivo utilizzato e, in caso di errori durante la compilazione, sarà sempre possibile la modifica di quest'ultimo; infatti, il file avrà dei campi "form" compilabili anche dopo la chiusura del ricovero.

NB: la lettera di dimissioni verrà registrata nella cartella \src\documenti\ nel seguente formato "codSanitario-NumeroRicovero-data.pdf", inoltre in caso di errore sarà possibile effettuare modifiche direttamente al file PDF attraverso le form inserite all'interno dello stesso.

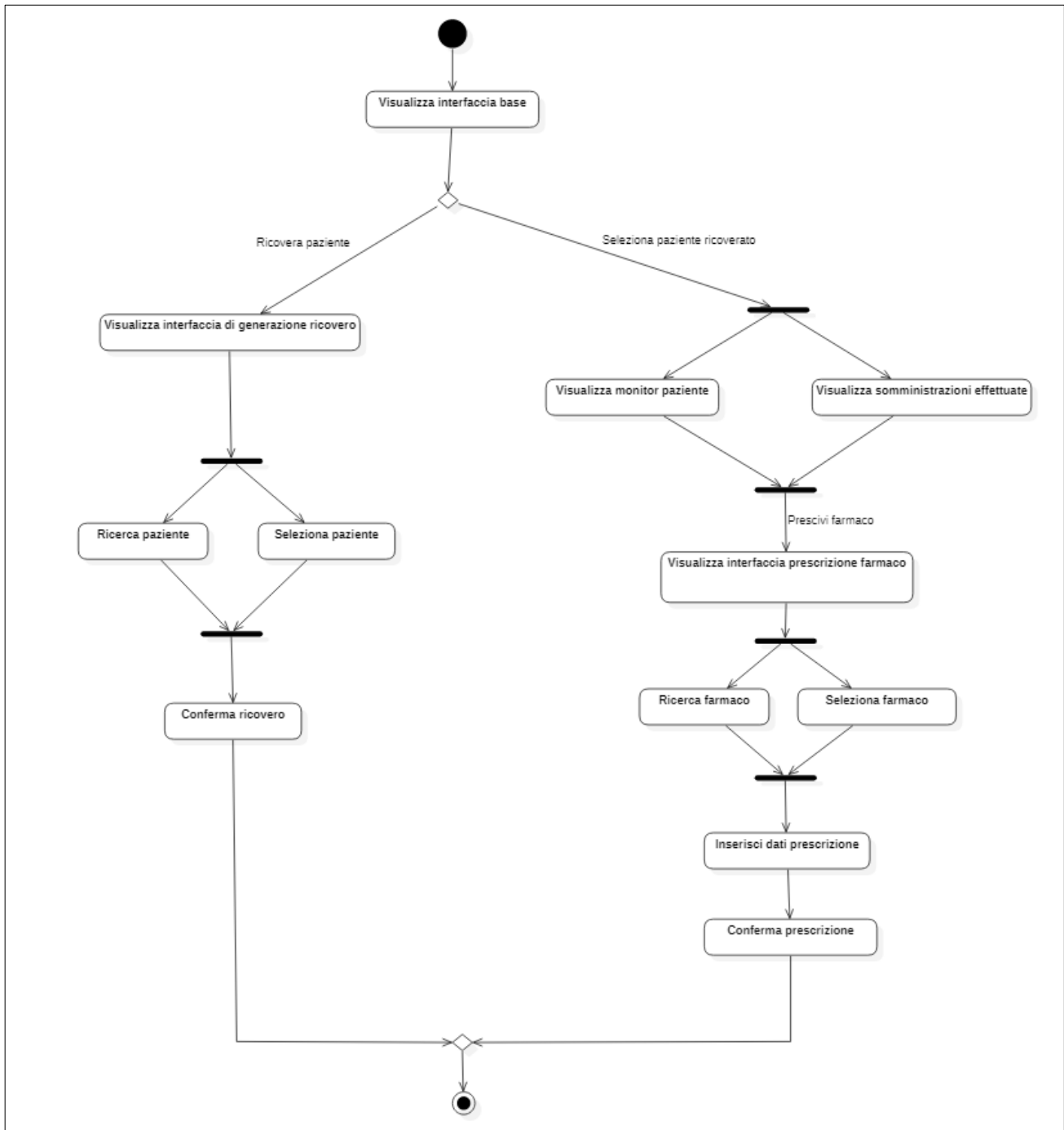
Attori: Primario
Precondizioni: Il primario deve essersi autenticato
Passi: <ol style="list-style-type: none">1. <i>Il primario viene introdotto all'interfaccia iniziale</i>2. <i>Il primario seleziona un paziente ricoverato</i>3. <i>Il primario seleziona l'opzione di chiusura ricovero</i>4. <i>Il primario inserisce opportunamente i dati relativi al ricovero</i>5. <i>Conferma la chiusura del ricovero</i>
Post condizioni: Il paziente viene dimesso e la cartella clinica archiviata, viene generata una copia della lettera di dimissioni in formato Pdf.

Diagrammi delle attività

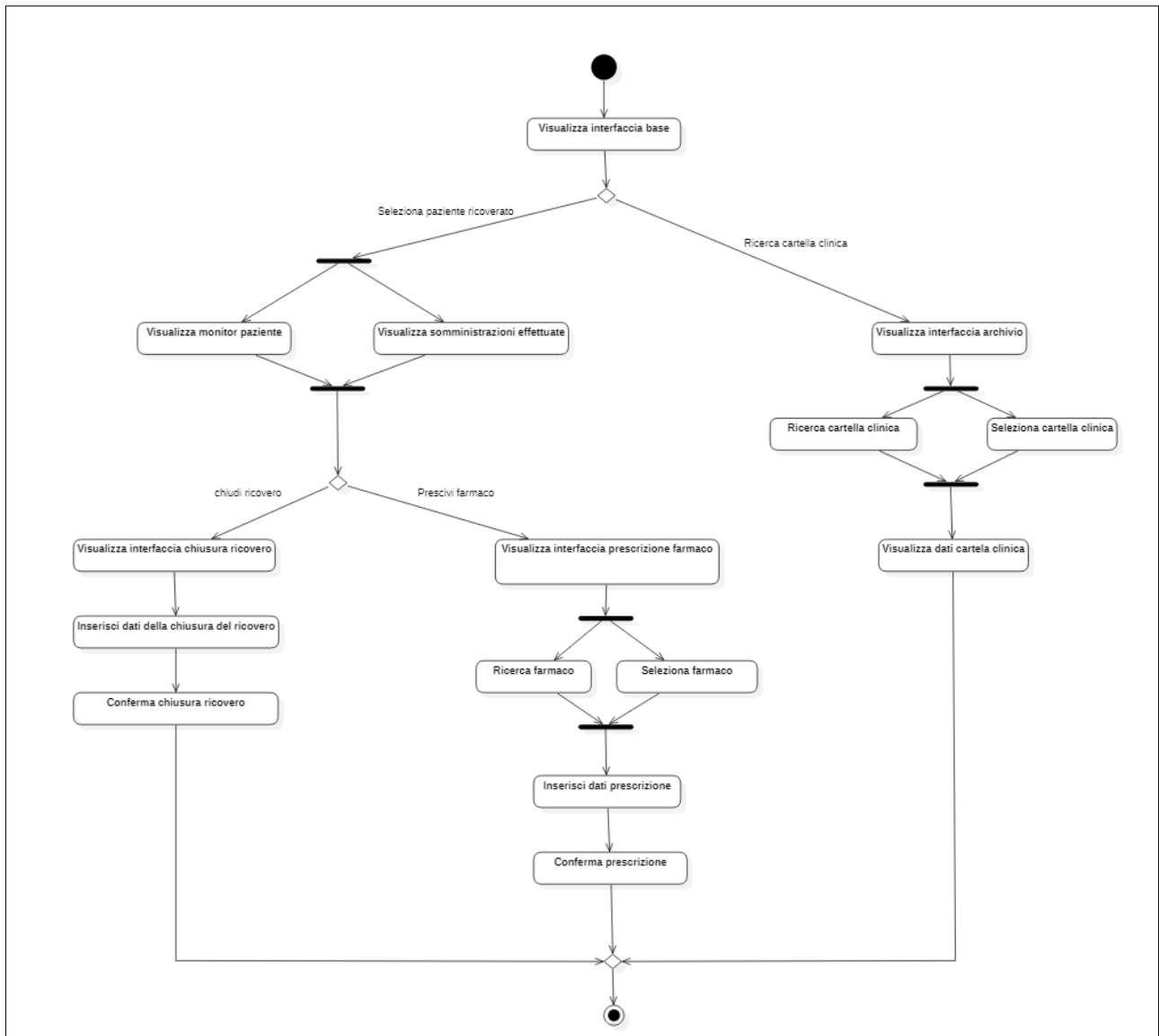
Sono di seguito riportati i diagrammi delle attività con i principali interazioni che ogni utente del sistema può avere con il sistema gestionale.



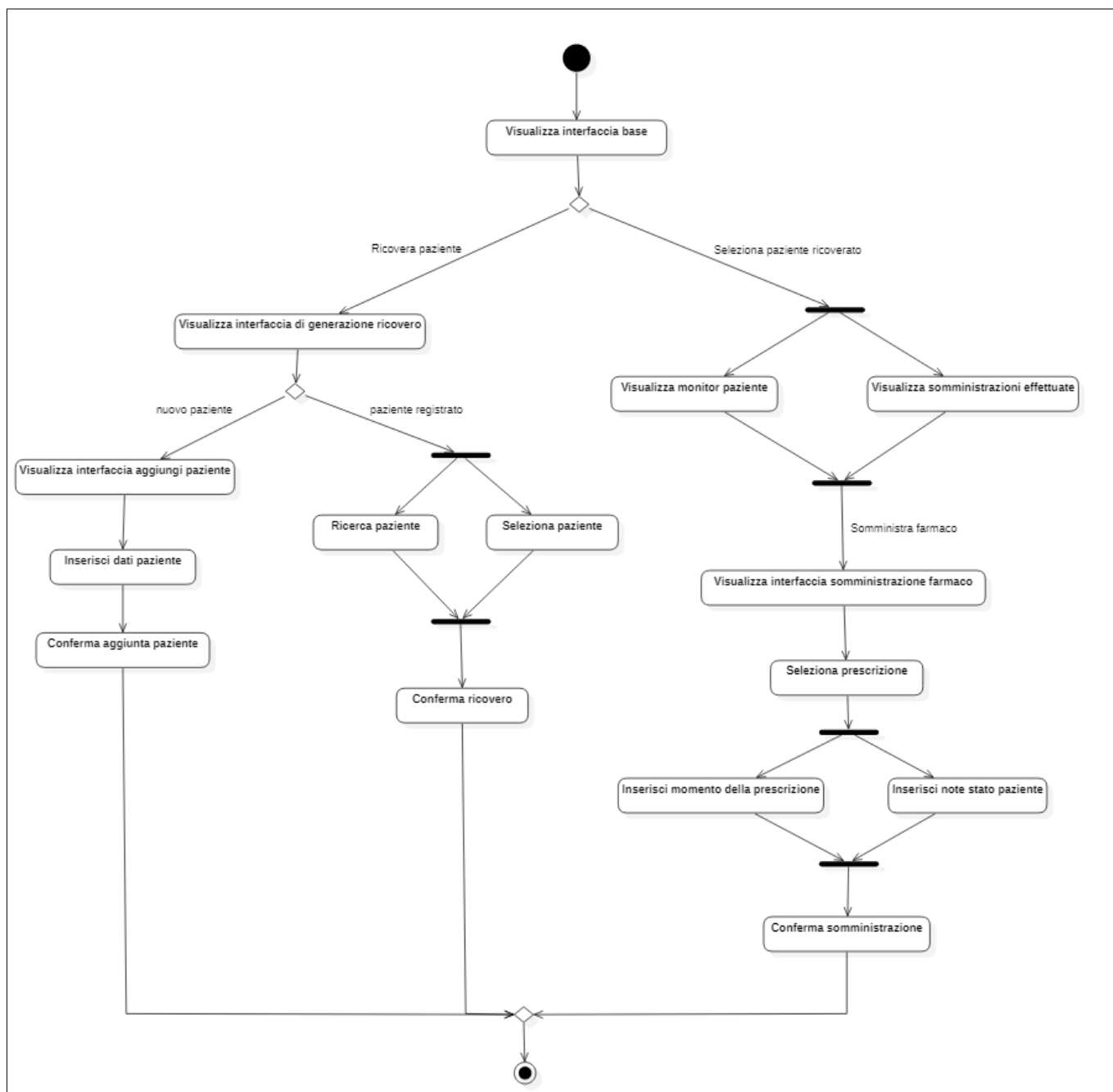
Attività utente non autenticato



Attività medico



Attività primario

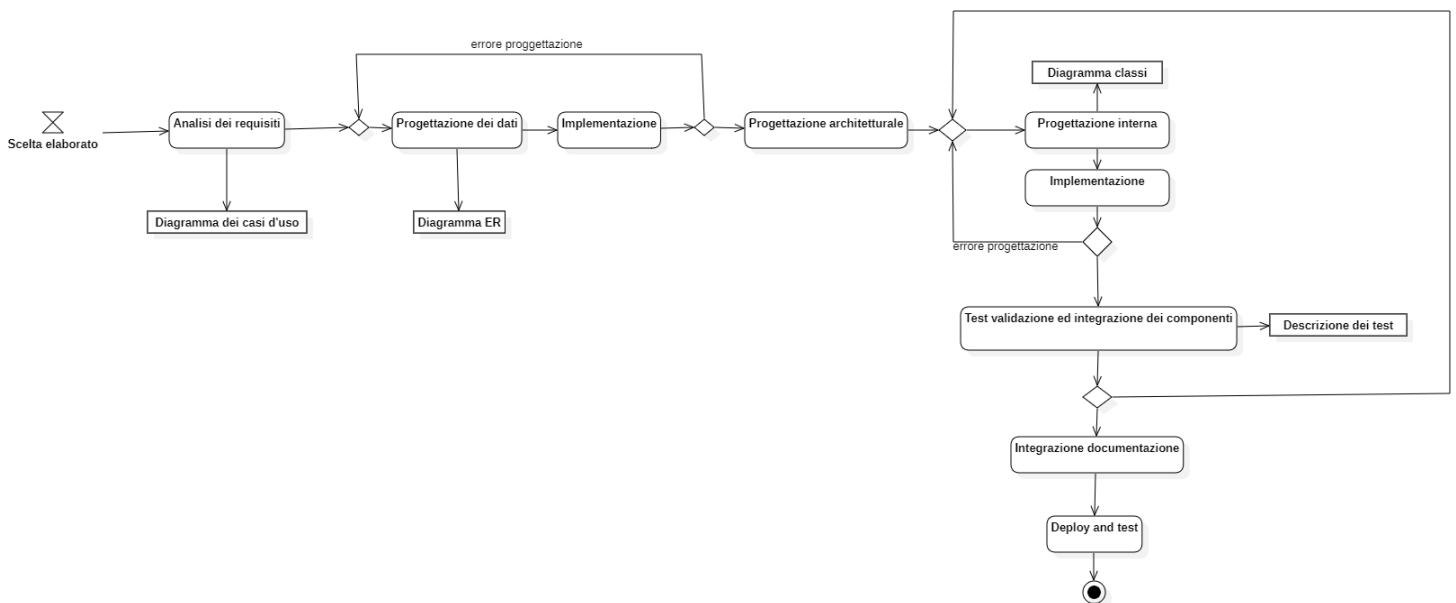


Attività infermiere

Processo di sviluppo:

Il processo di sviluppo utilizzato è stato di tipo agile ed incrementale. Come primo passo del processo di sviluppo è stata effettuata l'analisi dei requisiti della traccia proposta. Questa prima fase ha permesso la creazione della documentazione relativa ai casi d'uso principali che hanno costituito la base per la progettazione dei vari componenti del sistema, oltre che alla creazione dei diagrammi delle attività per gli attori del sistema. Dopo l'analisi dei requisiti si è voluta progettare la base di dati e le strutture dati necessarie alla comunicazione tra le componenti del sistema, oltre che alla memorizzazione dei dati. Le fasi di progettazione e implementazione dei componenti sono state sequenziali, ovvero prima si è cercato di progettare ed ideare il componente, per poi avere un'idea ben strutturata sul quale basarci durante l'implementazione. Alla fine della fase di implementazione per ogni componente è stata svolta una breve fase di test sia a livello grafico che funzionale. Dopo la prima validazione di un componente è avvenuta l'integrazione di quest'ultimo e un'ulteriore fase di test per verificare la corretta comunicazione e integrità dei componenti integrati. Avvenuta la validazione del prototipo si è quindi aggiornata ed integrata la documentazione abbozzata in fase di progettazione/implementazione, sfruttandola anche per verificare un'ultima volta le componenti del prototipo prodotto. Sono stati realizzati in particolare:

- **[aggiornamento] diagrammi delle attività** delle principali interazioni tra utente e sistema,
- **diagrammi di sequenza** relativi ai più significativi casi d'uso,
- **diagrammi delle classi** più significative del prototipo,
- **descrizione dei test.**



Architettura ed implementazione del sistema:

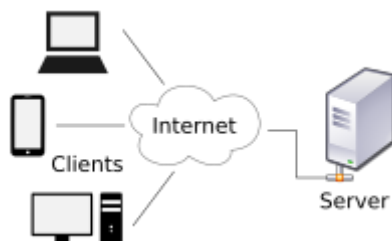
Il sistema informativo realizzato è stato suddiviso in due macro-parti:

- **Sistema informativo gestionale:** sistema dedicato al personale medico e infermieristico che appartiene al reparto di terapia intensiva dell'ospedale. È stato realizzato per permettere al personale dell'ospedale di accedere ai dati relativi ai ricoveri, permetterne la gestione e permettere a dispositivi diversi di avere accesso ai dati registrati dal sistema di monitoraggio.
- **Sistema monitoraggio pazienti:** come da nome, consiste in un sistema di monitoraggio, rilevazione e di memorizzazione real-time dei dati relativi ai pazienti ricoverati.

L'esigenza di poter permettere la comunicazione tra questi due sistemi eterogenei e la possibilità di memorizzare i dati in maniera strutturata e con criterio rispetto ai requisiti del cliente, ci ha portato ad optare per l'implementazione del pattern architetturale client-server.

Architettura Client-Server:

- **Server:** i servizi messi a disposizione dei client sono due
 - Database server: si occupa della memorizzazione dei dati relativo alle cartelle cliniche, ai pazienti al personale medico e di tutti i dati utili al sistema informativo gestionale dell'ospedale,
 - Application server: si occupa della ricezione dei dati relativi all'andamento dei parametri vitali da parte del sistema di monitoraggio e si occuperà di fornire i dati ricevuti dal sistema di monitoraggio agli utenti del sistema gestionale.
- **Client:** i client utilizzeranno i servizi messi a disposizione dal server, in particolare nel sistema possono essere di due tipi:
 - Client del sistema di monitoraggio: consiste in una serie di rilevatori di parametri vitali (battito cardiaco, ossigenazione del sangue, pressione e temperatura sono quelli utilizzati) presenti in ogni posto letto del reparto, i cui dati rilevati verranno inviati, con un time-out definito in fase di analisi, all'AS che li memorizzerà (nel nostro prototipo i rilevatori saranno implementati attraverso la classe monitor-Stanza).
 - Client del sistema gestionale: che richiederanno i dati memorizzati dall'AS.



Ogni utente autenticato (medici, infermieri e primario) attraverso il software del sistema gestionale avrà la possibilità di comunicare con i monitor dei vari pazienti attraverso il servizio messo a disposizione dal server.

Per permettere l'invio di dati in modo strutturato e semplificare la realizzazione della comunicazione tra i client e il server, sono state create delle classi "comuni" in modo tale da permettere una corretta comunicazione fra le due parti; quindi, attraverso il canale di comunicazione socket, verranno inviati direttamente gli oggetti interessati, senza dover effettuare conversioni di tipo e quindi generare complessità inutile.

Classi realizzate per la comunicazione client-server e tra i due sistemi: Allarme, Farmaco, LoggedUser, parametriVitali, Paziente, Prescrizione, Ricovero, Somministrazione (hanno metodi get e set e i loro attributi sono identici alle relative tabelle nel database).

È di seguito riportato uno schema delle classi utilizzate per la comunicazione tra i due sistemi e per il passaggio delle informazioni tra controller all'interno del software client stesso (sono stati tralasciati i metodi get and set e i costruttori per semplificare)

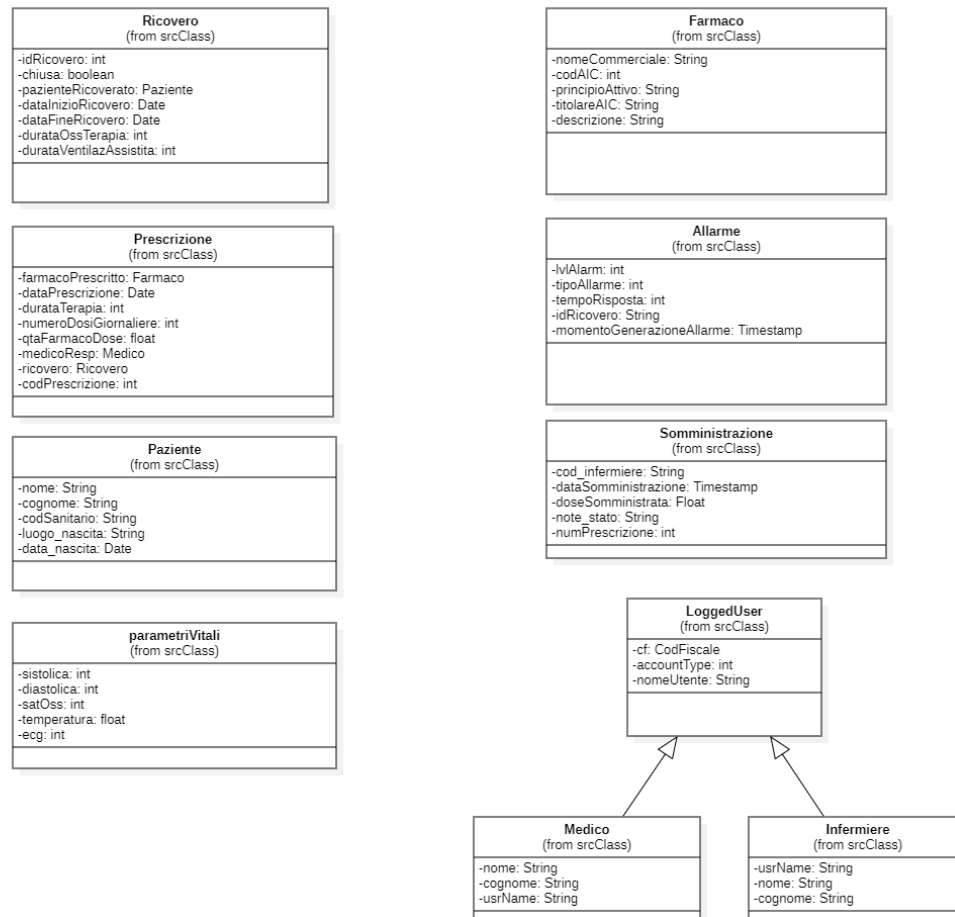


Diagramma delle classi – classi supporto

Server: funzionalità dell'Application Server

L'applicativo Server si divide in due classi principali:

- **ServerCentrale:** accetta in loop connessioni da parte del sistema gestionale (*porta 50512*), inoltre alla sua istanziazione avvia i monitorStanza per i pazienti già ricoverati. Una thread in background si occuperà di gestire l'aggiunta di nuovi monitor a seguito della registrazione di un nuovo ricovero.
- **ServeClient:** rappresenta l'istanza dell'applicativo server per un determinato utente che ne richiede il servizio attraverso l'avvio del sistema gestionale (client). Si occupa appunto di comunicare al client i dati rilevati dal sistema di monitoraggio.

NB: il client del sistema di monitoraggio (rilevatore dei parametri vitali) nel prototipo realizzato non esiste, è stata implementata una classe monitorStanza in grado di generare in modo casuale parametri vitali verosimili. Non è stato dunque realizzato un vero e proprio client, in quanto sarà tutto gestito all'interno del server ([visionare cap. Test](#)).

I principali servizi offerti sono la visualizzazione dei parametri vitali in real-time e l'invio di allarmi relativi all'andamento dei parametri vitali.

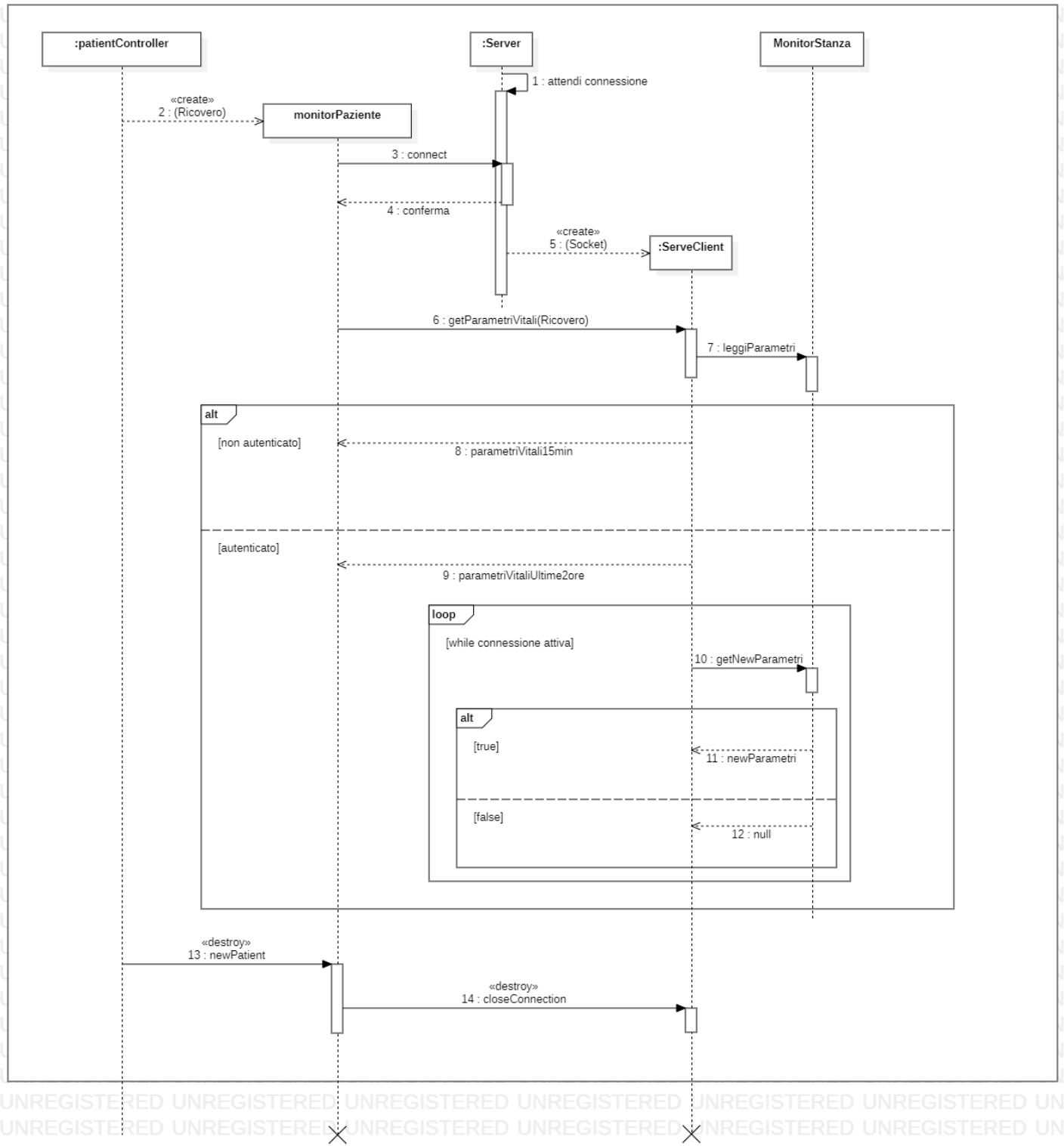
Visualizzazione parametri vitali real-time

Ogni utente autenticato deve avere la possibilità di visualizzare i dati dei parametri vitali dei pazienti ricoverati. L'applicativo non fa altro che accettare connessioni in entrata dalla porta 50512 da parte dei vari utenti autenticati, istaura una connessione e quindi crea un socket di comunicazione. Fatto questo rimane in attesa di sapere a quale ricovero il client è interessato; una volta ricevuto va a leggere i parametri vitali memorizzati dal monitor (su un semplice file) e li invia attraverso la socket al client.

I parametri vitali che il sistema di monitoraggio rileva sono:

- Pressioni sistolica e diastolica (SBP e DBP) ogni 2 minuti,
- Saturazione dell'ossigeno 5 minuti,
- Temperatura ogni 3 minuti,
- Battito cardiaco 1 minuto.

La classe MonitorStanza si occuperà della ricezione dei segnali di un particolare paziente, inoltre viene gestita automaticamente dal server attraverso il metodo controlloNuoviRicoveri() l'aggiunta di monitor per un paziente appena ricoverato.



Interazione utente – sistema monitoraggio

Invio Allarme

Solo i medici e il primario ricevono gli allarmi da parte del sistema di monitoraggio. Gli allarmi hanno diversi livelli di gravità crescente:

- Livello 1: (tempo reazione 3 minuti)
 - a. Tachicardia (Battito cardiaco > 100 bpm), Aritmia;
- Livello 2: (tempo reazione 2 minuti)
 - b. Ipertensione (p. diastolica > 140 e p. sistolica > 95),
 - c. Ipotensione (p. diastolica < 100 e p. sistolica < 60),
 - d. Ipertermia (temperatura > 40),
 - e. Ipotermia (temperatura < 35);
- Livello 3: (tempo reazione 1 minuto)
 - a. Dispnea (saturazione ossigeno < 90%),
 - b. Flutter Ventricolare (battito cardiaco > 160).

Ogni livello di gravità di un allarme ha un tempo di reazione diverso, se l'allarme non verrà spento entro il tempo pre-stabilito, verrà salvata una segnalazione.

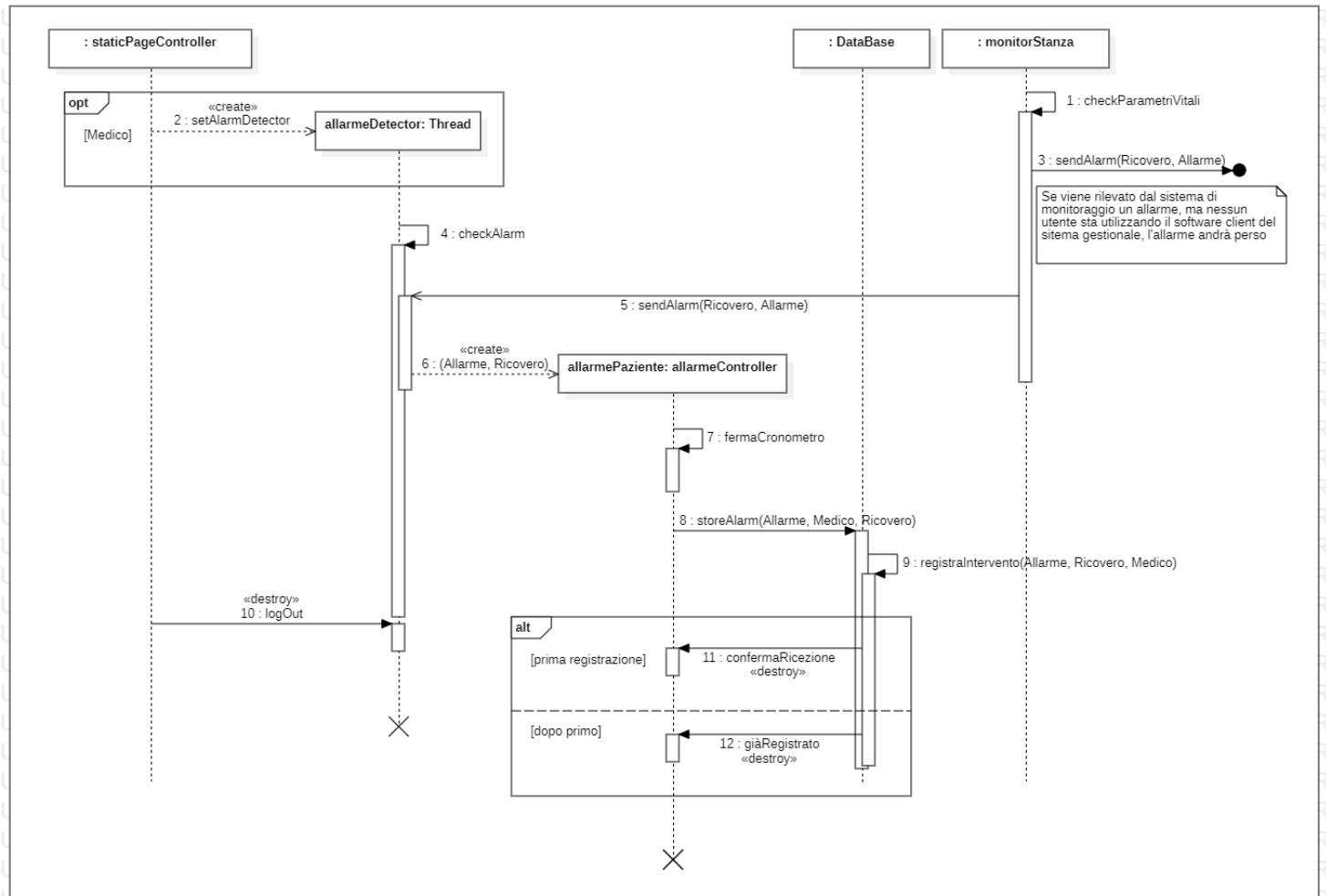
Gli allarmi registrati dal sistema di monitoraggio sono inviati al sistema gestionale attraverso la porta 50153.

Quando un medico si autentica con dall'apposita interfaccia, verrà automaticamente creato un Thread in background che si occuperà della ricezione sulla porta 50153 di messaggi da parte del sistema di monitoraggio.

Gli allarmi sono mandati in modalità **MULTICAST** all'indirizzo 225.0.0.1, quindi tutti i medici autenticati al momento della generazione dell'allarme, lo riceveranno. Al momento della ricezione verrà avviato un cronometro che memorizzerà il tempo di reazione del medico alla ricezione dell'allarme, una volta spento l'allarme il medico potrà attraverso l'apposita interfaccia annotare i passi dell'intervento eseguito sul paziente per riportarlo ad uno stato normale. Solo il primo medico che inserirà l'intervento effettuato e lo confermerà avrà la possibilità di registrare quest'ultimo, gli altri che hanno ricevuto il segnale potranno ignorarlo o, in caso di invio dell'intervento duplicato verrà generato un errore.

Criticità: per ricevere l'allarme necessariamente almeno un medico del reparto deve avere il software del sistema gestionale in esecuzione e si deve essere autenticato, altrimenti il sistema di monitoraggio manderà il segnale a vuoto e quest'ultimo andrà perso.

È di seguito riportato un diagramma di sequenza della ricezione dell'allarme da parte di un client del sistema gestionale



Ricezione allarme

Client: Sistema informativo gestionale

Architettura adottata (MVC)

Il sistema informativo gestionale è stato realizzato mediante le tecniche di modellazione ad oggetti. Per quanto riguarda l'architettura adottata, per la realizzazione di questa prima parte di sistema ci si è fortemente ispirati al pattern architetturale **MVC** (Model View Controller), pattern nativamente implementato nella libreria JavaFX. Il software client sarà dunque suddiviso in tre parti:

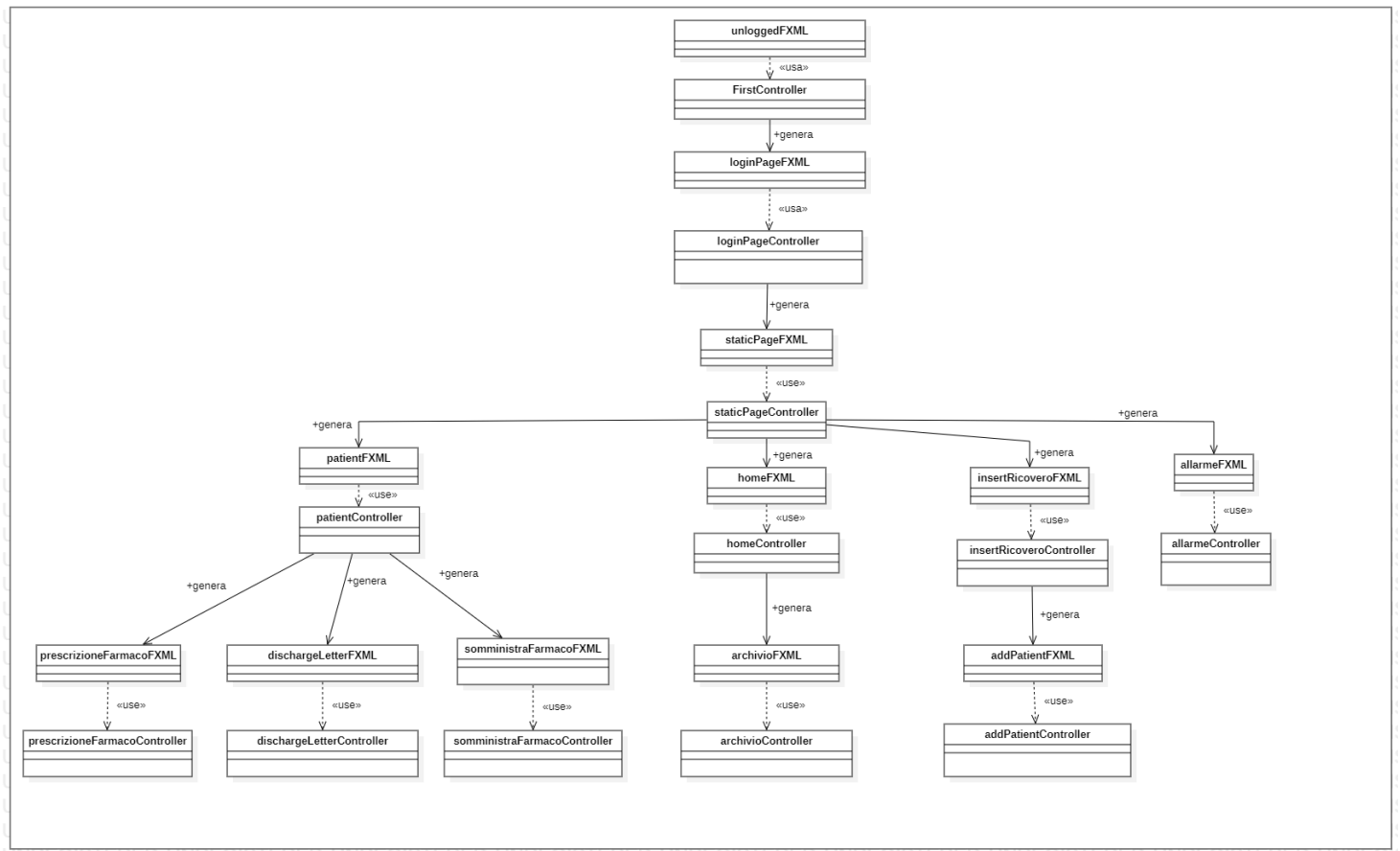
- **Model:** fornisce i metodi per accedere ai dati utili all'applicazione, nel nostro caso in particolare andrà anche ad effettuare la comunicazione con il server per la memorizzazione dei dati inseriti in "View" dal personale medico;
- **View:** visualizza i dati contenuti passati dal controller (o messi a disposizione da esso) altro non è che l'interfaccia grafica messa a disposizione dell'utente;
- **Controller:** riceve i comandi e i dati che l'utente esegue e invia attraverso la View, può manipolarli ed effettuare controlli prima di mandarli al Model.

Il model è stato implementato con la classe Database (di seguito approfondita), e si occupa della gestione dell'invio e della ricezione dei dati richiesti e inviati dall'utente. In particolare, è la classe che gestisce e può manipolare l'intera mole di dati.

Il view-controller come sopra citato è un'architettura nativamente implementata in JavaFX in quanto le interfacce al momento dell'istanziatura sono esplicitamente collegate ad appositi controller. Questi ultimi si occuperanno di intercettare le operazioni/azioni effettuate dall'utente sull'interfaccia e di rispondere all'utente comunicando all'interfaccia i dati o le azioni elaborate. Sono di seguito riportati i diagrammi UML delle classi del view-controller e della classe database. Per una questione di leggibilità non sono stati riportati metodi e attributi delle classi del view-controller.

View-Controller:

In JavaFX, i file FXML sono utilizzati per formattare e generare la vista che permette all'utente di interfacciarsi con il programma. Ogni file FXML al suo interno contiene il collegamento alla classe Controller di riferimento (come da schema), ciò facilita il caricamento del controller e il suo legame con la vista all'inizializzazione.



Class diagram del view-controller

Model: classe DataBase

La classe DataBase funge da Model del pattern architetturale MVC: si occuperà principalmente di fornire l'accesso ai dati presenti nel database sql. Ci si è fortemente ispirati al *DAO pattern* per la realizzazione di tale classe in modo da separare la logica di accesso ai dati dalla logica dell'applicativo realizzato. Solo gli oggetti della classe DataBase potranno effettuare connessioni e interrogazioni verso il server. Per una questione di sicurezza e gestione della connessione tra client e database server si è voluto utilizzare il *pattern Singleton*: mi permette di creare un unico oggetto della classe Database e di richiamarlo di volta in volta attraverso il metodo *getInstance()*: l'utilizzo di tale pattern mi permette infatti di controllare che ogni istanza del software client possa effettuare al più una connessione al database e che, alla chiusura del software, venga opportunamente gestita la chiusura della connessione al server. Vengo qui riportati i metodi più significativi della classe (*per questioni di leggibilità vengono tralasciati i parametri formali*).

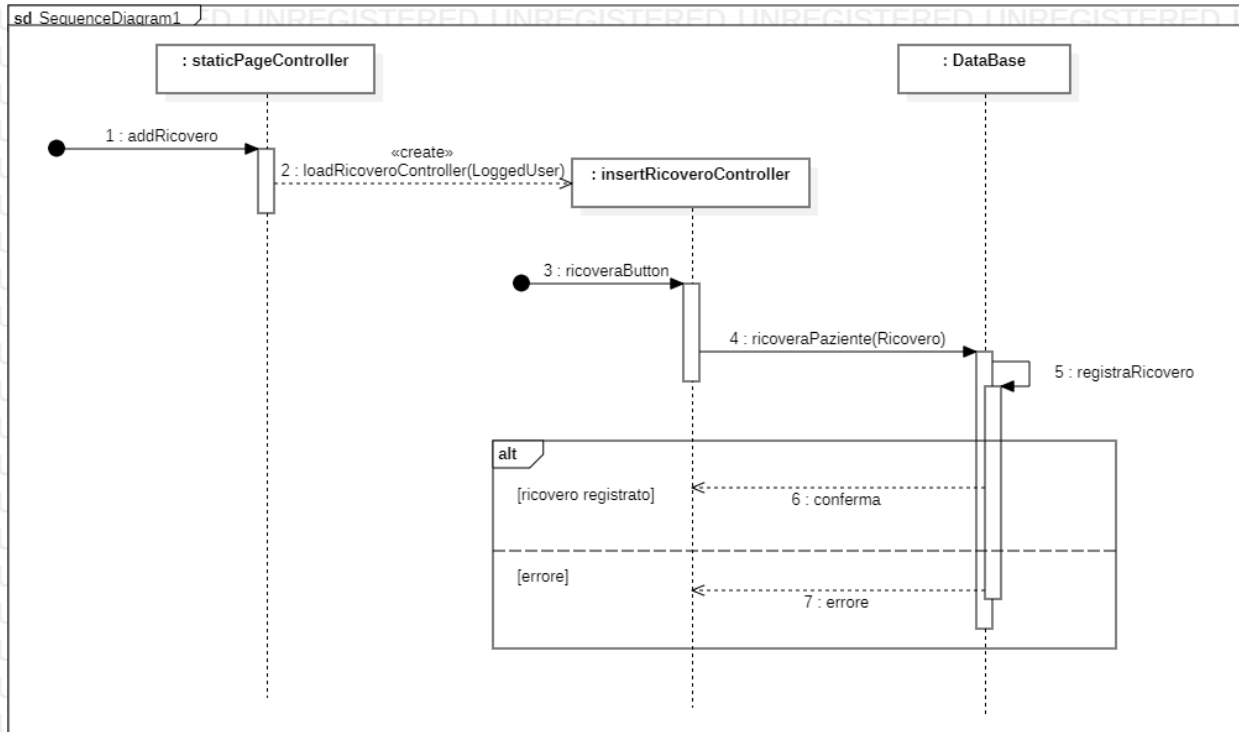
Metodi classe:

- *login()*: controlla se esiste un utente con le credenziali fornite;
- *close()*: effettua la chiusura connessione;
- *registraPaziente()*: registra i dati anagrafici di un nuovo paziente;
- *generaRicovero()*: registra un nuovo ricovero e apre la cartella clinica;
- *registraPrescrizione()*: registra la prescrizione effettuata dal medico;
- *registraSomministrazione()*: registra la somministrazione effettuata da un infermiere;
- *closeRicovero()*: chiude cartella clinica ricovero
- *getPatientList()*: ritorna la lista dei pazienti registrati nel database;
- *getSearchedFarmaco()*: ritorna il farmaco ricercato;
- *getSearchedRicovero()*: ritorna il ricovero ricercato;
- *getSearchedPaziente()*: ritorna il paziente ricercato;
- *getPatientsHospitalized()*: ritorna la lista dei pazienti attualmente ricoverati;
- *getRicoveriArchiviati()*: ritorna la lista delle cartelle cliniche chiuse;
- *getRicovero()*: ritorna un ricovero in base al numero identificativo;
- *getRicoveroArchiviato()*: ritorna la cartella clinica di un ricovero passato in base al numero identificativo;
- *getPrescrizioni()*: ritorna la lista delle prescrizioni per un determinato ricovero;
- *getSomministrazioni()*: ritorna la lista delle somministrazioni per un determinato ricovero;
- *getUserAccount()*: ritorna il tipo di utente che ha effettuato la connessione al database;

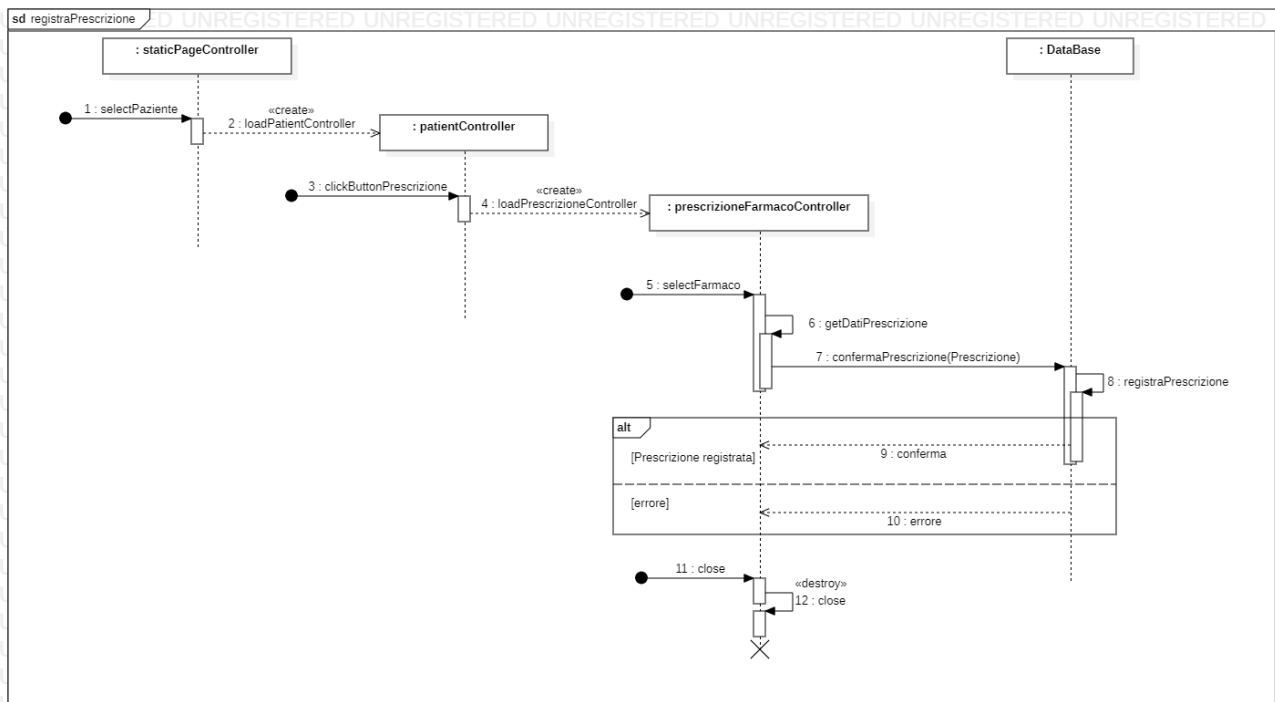
DataBase
-userAccount: LoggedUser -connection: Connection -statement: Statement
-connectToDb(): void -close(): void +login(username: String, password: String): LoggedUser +insertPatient(nome: String, cognome: String, dataNascita, luogoNascita: String, codSanitario: String): boolean +generaRicovero(codSanitarioRicovero: String): boolean +setUserAccount(userAccount): void +registraPrescrizione(prescrizione): boolean +registraSomministrazione(somministrazione): boolean +closeRicovero(ricovero): boolean +getPatientList(): ArrayList<Paziente> +getSearchedFarmaco(denominazione: String, codAIC: String, principioAttivo: String): ArrayList<Farmaci> +getSearchedRicovero(nomePaziente: String, cognomePaziente: String, codiceSanitarioPaziente: String, dataInizio, dataFine): ArrayList<Ricovero> +getSearchedPaziente(nome: String, cognome: String, codSanitario: String): ArrayList<Paziente> +getPatientsHospitalized(): ArrayList<Paziente> +getRicoveriArchiviati(): Ricovero +getRicovero(codRicovero: Integer): Ricovero +getRicoveroArchiviato(codRicovero: Integer): Ricovero +getPrescrizioni(ricovero): ArrayList<Prescrizione> +getSomministrazioni(ricovero): ArrayList<Somministrazione> +getMedicoFromLoggedUser(user): Medico +getUserAccount(): LoggedUser

Diagrammi di sequenza:

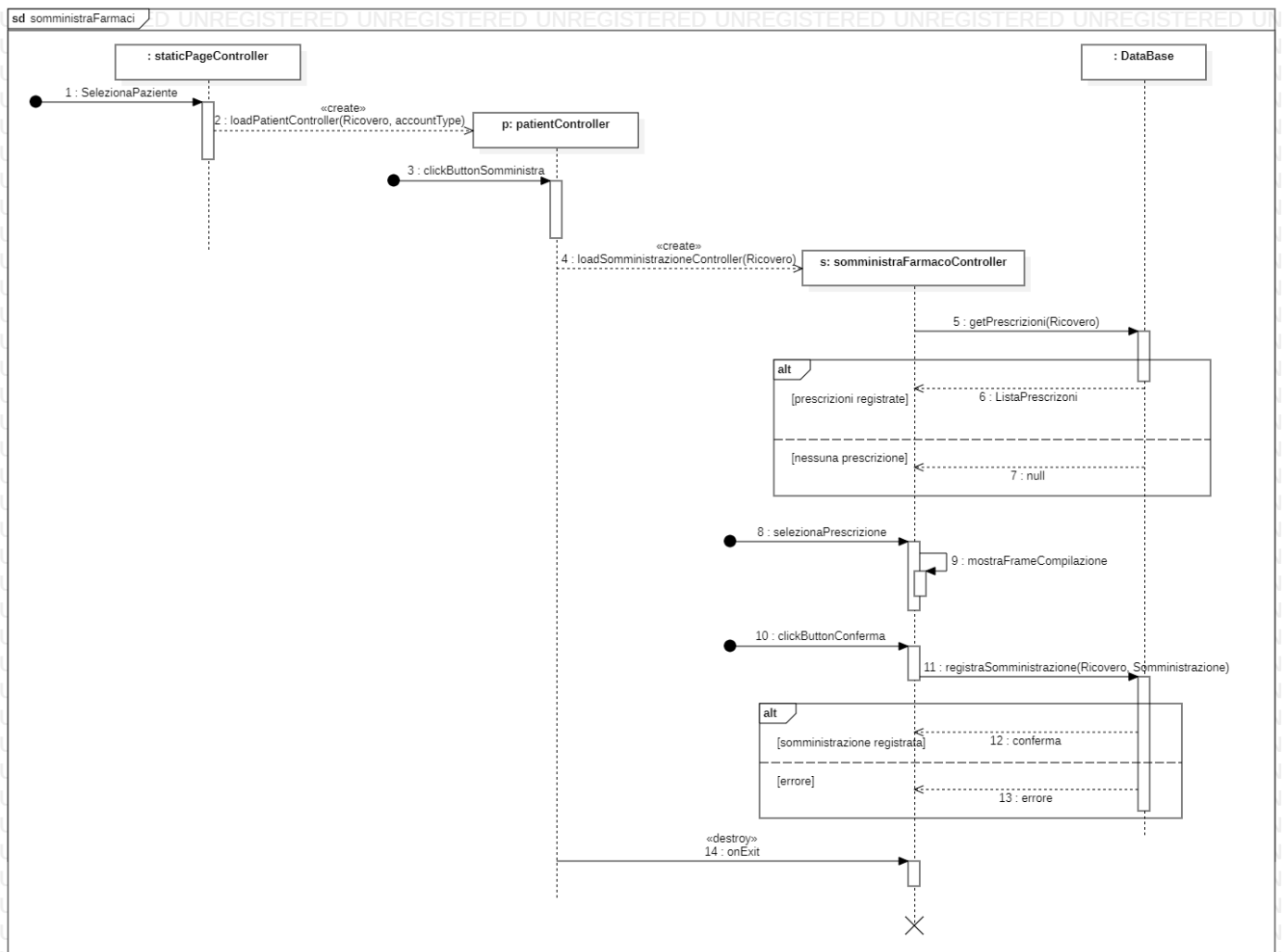
Per semplificare la lettura dei diagrammi di sequenza sono stati tralasciati i passaggi di parametri tra la view e il controller, riportando solo gli input e le azioni dell'utente ritenute più significative per la rappresentazione e la leggibilità dei diagrammi.



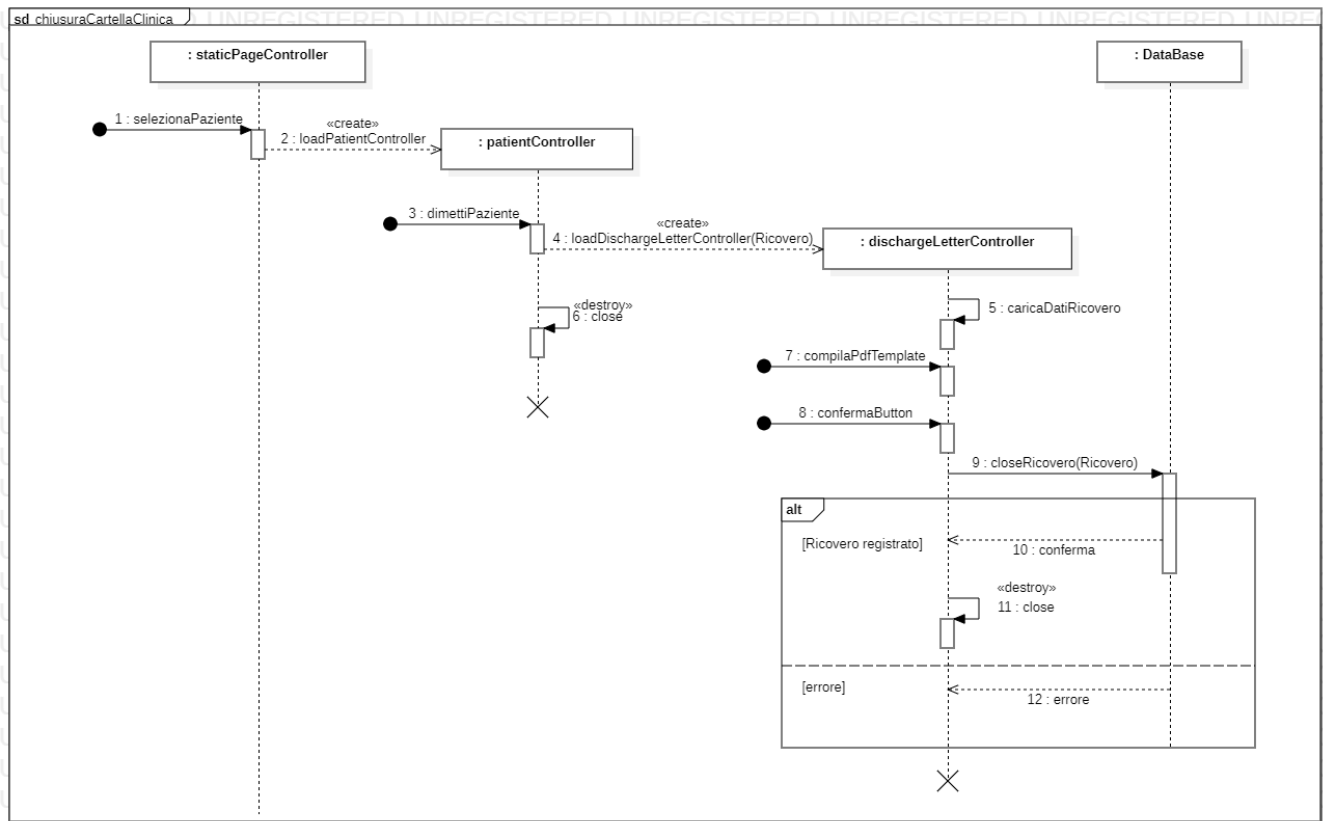
Registra Ricovero



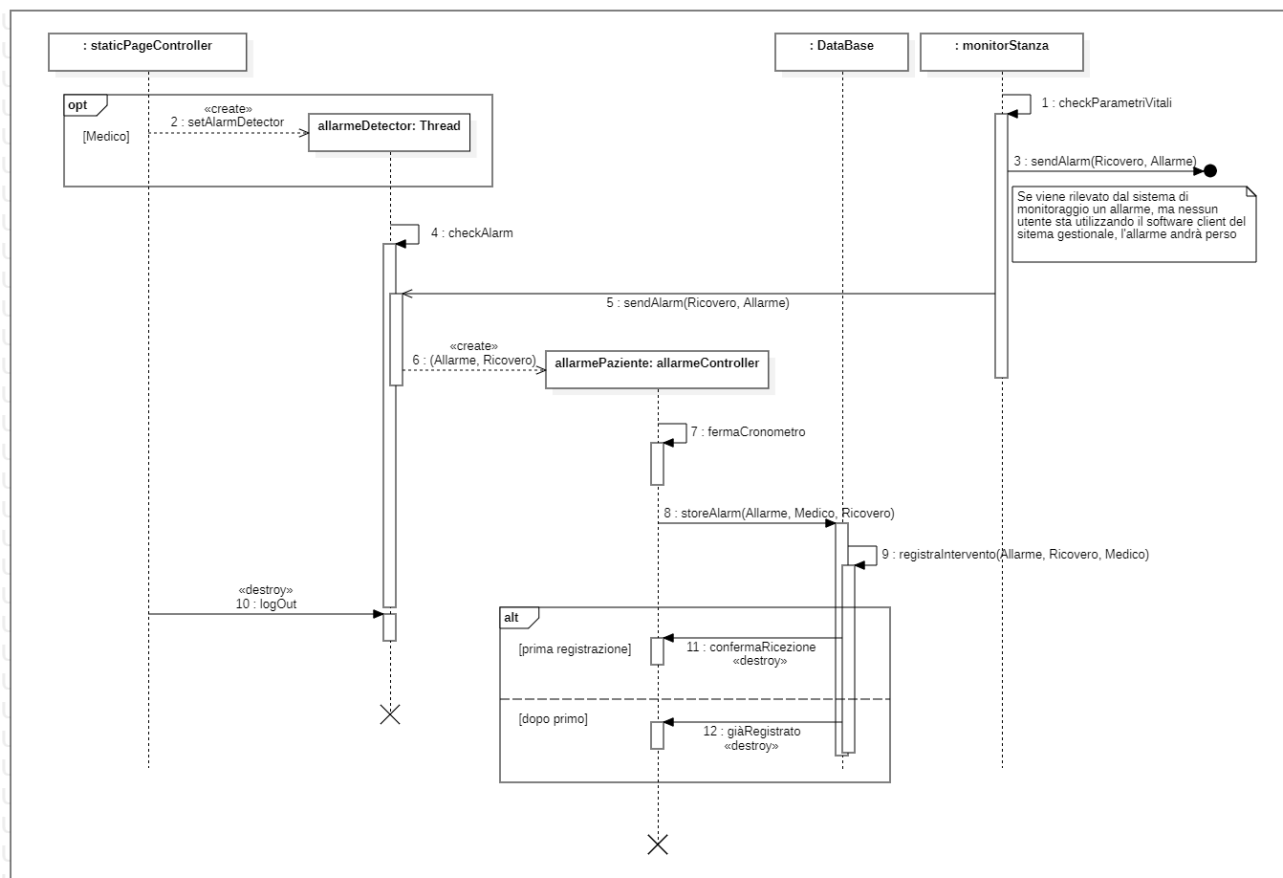
Registra prescrizione



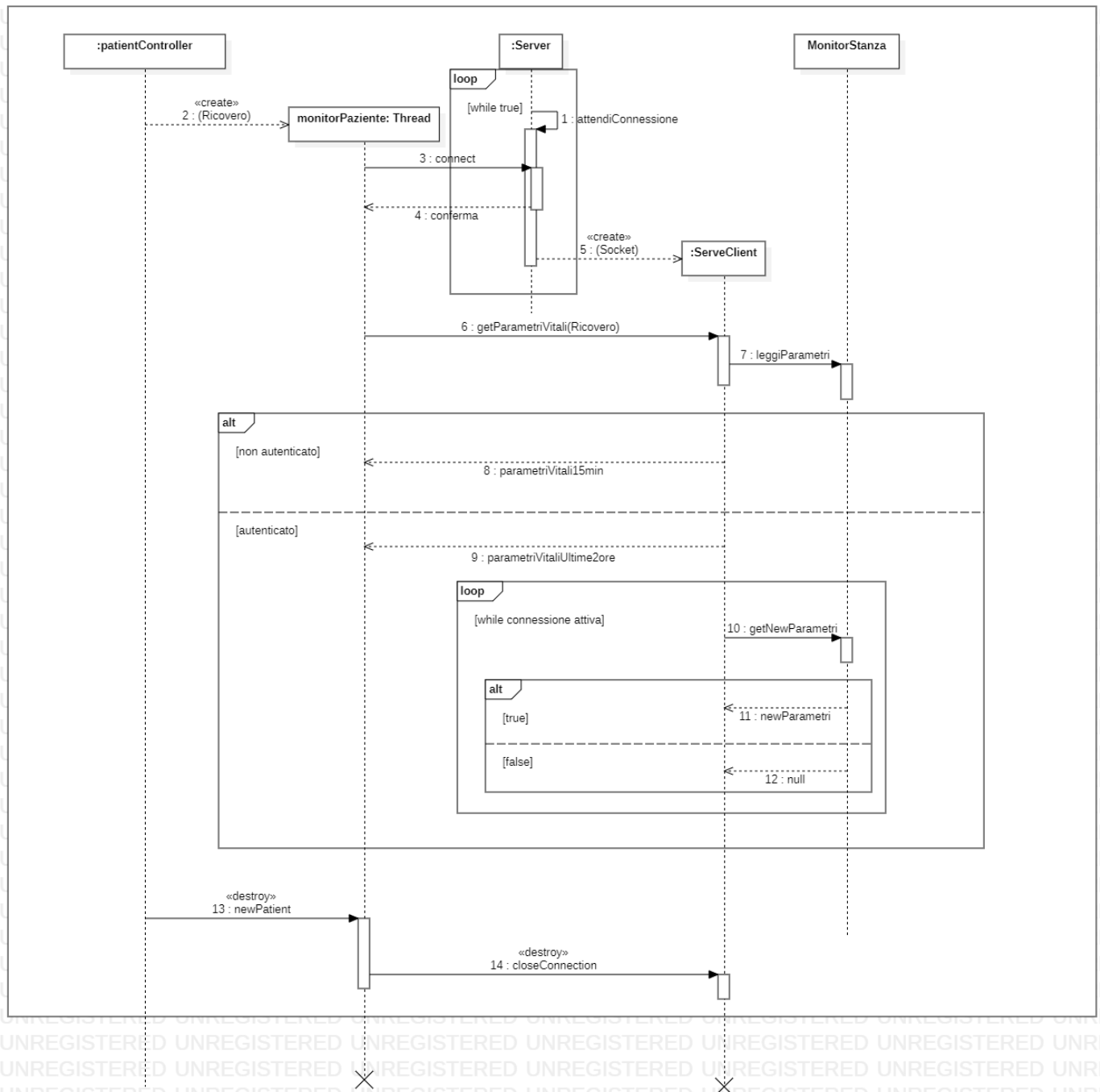
Registra somministrazione



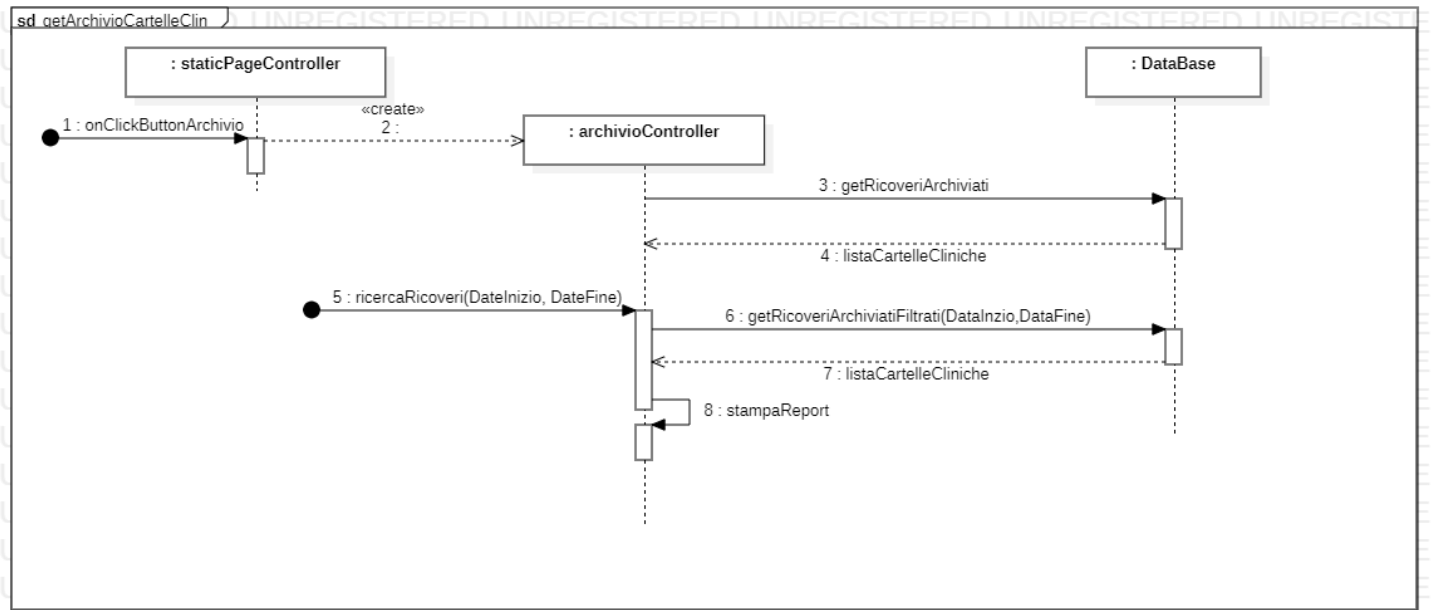
Chiusura cartella clinica



Ricezione allarmi provenienti dal sistema di monitoraggio



Ricezione e visualizzazione dei parametri vitali



Archivio cartelle cliniche e stampa report

Test

Sistema monitoraggio: generazione segnali

Il sistema sviluppato è un prototipo non ancora installato nel sistema informativo dell'ospedale dunque, non avendo a disposizione dati e/o rilevatori di parametri vitali che potessero direttamente generare i segnali richiesti, si è optato per la creazione di un'apposita classe monitorStanza il cui compito principale è quello di generare parametri vitali verosimili e salvarli in locale. Attraverso la classe DataGen viene creato un file così formattato: `"*codRicovero*_codSanitarioPazienteRicoverato*.txt"` per ogni monitorStanza che ne richiama i metodi. In questa classe sono poi stati implementate le verifiche sui parametri vitali generali e inviati gli eventuali allarmi in modalità multicast al gruppo di indirizzi `"225.0.0.1:50153"`. Ogni ricovero registrato genererà implicitamente dunque il proprio monitorStanza il quale nello specifico avvierà quattro Thread, (concettualmente ogni Thread è un rilevatore diverso assegnato ad un determinato posto letto) ognuna delle quali con un diverso campionamento andrà a memorizzare sugli opportuni file i parametri vitali registrati secondo i requisiti del sistema:

- Thread rilevazione per pressioni sistolica e diastolica (SBP e DBP) ogni 2 minuti,
- Thread rilevazione per saturazione dell'ossigeno 5 minuti,
- Thread rilevazione per temperatura ogni 3 minuti,
- Thread rilevazione per battito cardiaco 1 minuto.

Il server per inviare i dati dei parametri vitali generati al sistema gestionale (client), procederà a leggere i dati salvati sul file contenete i *parametri vitali di test*, in ordine cronologico (la prima riga del file conterrà l'ultimo record scritto da un rilevatore di monitorStanza) e li invierà al client quando ne farà richiesta. In questo modo è stata implementata la funzione di visualizzazione e ricezione dei dati in "real-time" (Server e client comunicano sulla porta 50152).

MonitorStanza[Thread]: metodi significativi

- Ricovero getRicovero(): ritorna il ricovero (Cartella clinica del paziente);
- void setRicovero(Ricovero ricovero): al momento del lancio di un monitorStanza il server prima di prenderne il controllo fa il set del ricovero di cui il monitor si dovrà occupare;
- parametriVitali getRandParameter(): metodo per la generazione dei parametri vitali;
- checkAlarm(parametriVitali parametri, int opt): effettua un controllo sui parametri vitali generati, in determinate condizioni lancia degli allarmi ai client.

DataGen: metodi significativi

- void genData(Ricovero ricovero): crea il file che conterrà la registrazione dei parametri vitali generali e li salva man mano che il monitor glieli passa. Questo metodo in particolare si occupa all'avvio dell'applicativo di caricare almeno 2 ore di segnali per ogni paziente ricoverato prima dell'avvio del test stesso.
- void addData(Ricovero ric, parametriVitali param, int option): si occupa di salvare i parametri generati dal monitorStanza.

Test sviluppatore: sistema gestionale

In questa fase sono stati effettuati test funzionali sui vari componenti del sistema gestionale utilizzato dal personale sanitario del reparto di terapia intensiva, i principali test svolti sono stati (segue una minima descrizione):

- **Verifica del corretto funzionamento di autenticazione:** questo test da parte dello sviluppatore ha previsto il controllo delle funzionalità messe a disposizione dell'interfaccia in base all'utente che le ha richieste, in particolare:
 - Utente non autenticato: ha la possibilità solo di effettuare il login o di visualizzare i parametri vitali dei pazienti ricoverati (solo negli ultimi 15 minuti);
 - Utente autenticato: ha l'accesso alla visualizzazione dei parametri vitali registrati nelle ultime due ore;
 - Medico: ricezione allarmi, prescrizione farmaci;
 - Primario: possibilità di navigare nell'archivio delle cartelle cliniche e dimissione di un paziente;
 - Infermiere: somministrazione farmaci prescritti.
- **Verifica generale sulle interfacce:** per ogni interfaccia utilizzata dal sistema si è provato a pigiare ogni tasto alla ricerca di qualche errore/bug del sistema;
- **Inserimento di una prescrizione:** si sono provati ad inserire dati non conformi;
- **Somministrazione di una prescrizione:** sono stati provati dati non conformi e si è provato a duplicare la somministrazione (errore database);
- **Ricezione e spegnimento allarmi:** si è verificato il corretto funzionamento e la corretta comunicazione tra Server-Centrale e il sistema di monitoraggio. Verifica anche della disponibilità della porta per creare il canale di comunicazione tra client e serve nel dispositivo di test. Si è testata la duplicazione di un allarme.
- **Test classe DataBase:** si è principalmente testata la corretta creazione delle query di interrogazione da inviare al database centrale. Si è poi verificata il corretto funzionamento del driver JDBC per il database mariaDB.
- **Verifica server spento:** verificato il NON funzionamento del sistema di monitoraggio qual ora il server sia spento.
- **Verifica salvataggio lettere dimissioni:** in locale vengono salvate ogni qual volta un primario dimetta un paziente delle lettere di dimissioni in formato PDF, è stata controllata la pertinenza dei dati inseriti e quelli salvati.
- **Verifica della visualizzazione dei parametri vitali:** è stata verificata la validità del sistema progettato andando a verificare anche se i grafici e il monitor del software client si aggiornano in maniera automatica alla generazione di nuovi parametri vitali.
- **Verifica delle varie form di ricerca:** è stata verificato il corretto funzionamento delle form di ricerca (in prescrizione farmaco, somministrazione e archivio delle cartelle cliniche).
- **Verifica terminazione del programma:** è stato verificato che alla chiusura del programma client tutte le Thread utilizzate (ad esempio per la ricezione di allarmi da parte dei medici) saranno anch'esse terminate.
- **Verifica corretta registrazione dell'allarme:** è stata verificata l'interfaccia che si occupa di tenere il tempo di reazione del medico, provando a spegnere un allarme e inviando i relativi dati al server.