



Progetto di Ingegneria del Software – Terapia Intensiva

Agresti Nicola VR407685

Piccinelli Marco VR407621

Università degli Studi di Verona
Dipartimento di Informatica
AA 2017/2018
Corso di laurea in Informatica

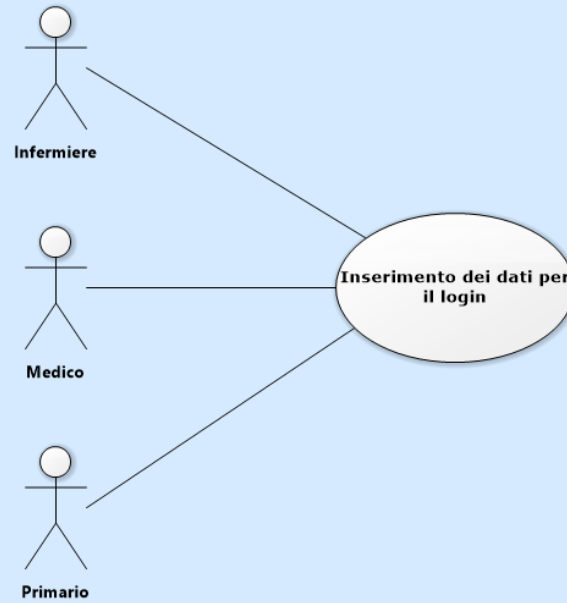
Analisi dei requisiti del nostro sistema

- Scopo
 - Gestire l'acquisizione di dati e segnali in una divisione di terapia intensiva
- Vincoli
 - Le operazioni verranno eseguite da 3 diverse tipologie di dipendenti (infermiere, medico e primario)

- **[R1]** In qualunque momento si possono aggiungere nuovi pazienti
 - **[R1.1]** Per ogni paziente saranno necessari: codice sanitario univoco, nome, cognome, luogo di nascita e data
 - **[R1.2]** Al momento del ricovero in terapia intensiva sarà necessaria una diagnosi di ingresso scritta dal medico
- **[R2]** Durante il ricovero verranno gestite, dai medici, le eventuali prescrizioni di farmaci
 - **[R2.1]** Per ogni prescrizione verrà memorizzato il nome commerciale, la data di prescrizione, la durata della terapia, il numero di dosi giornaliere, la quantità di farmaco per dose e il nome del dottore responsabile della prescrizione
- **[R3]** Durante il ricovero, in base alle prescrizioni, gli infermieri somministreranno i farmaci
 - **[R3.1]** Durante la somministrazione verrà salvata la data, l'orario con precisione al minuto, dose somministrata ed eventuali note

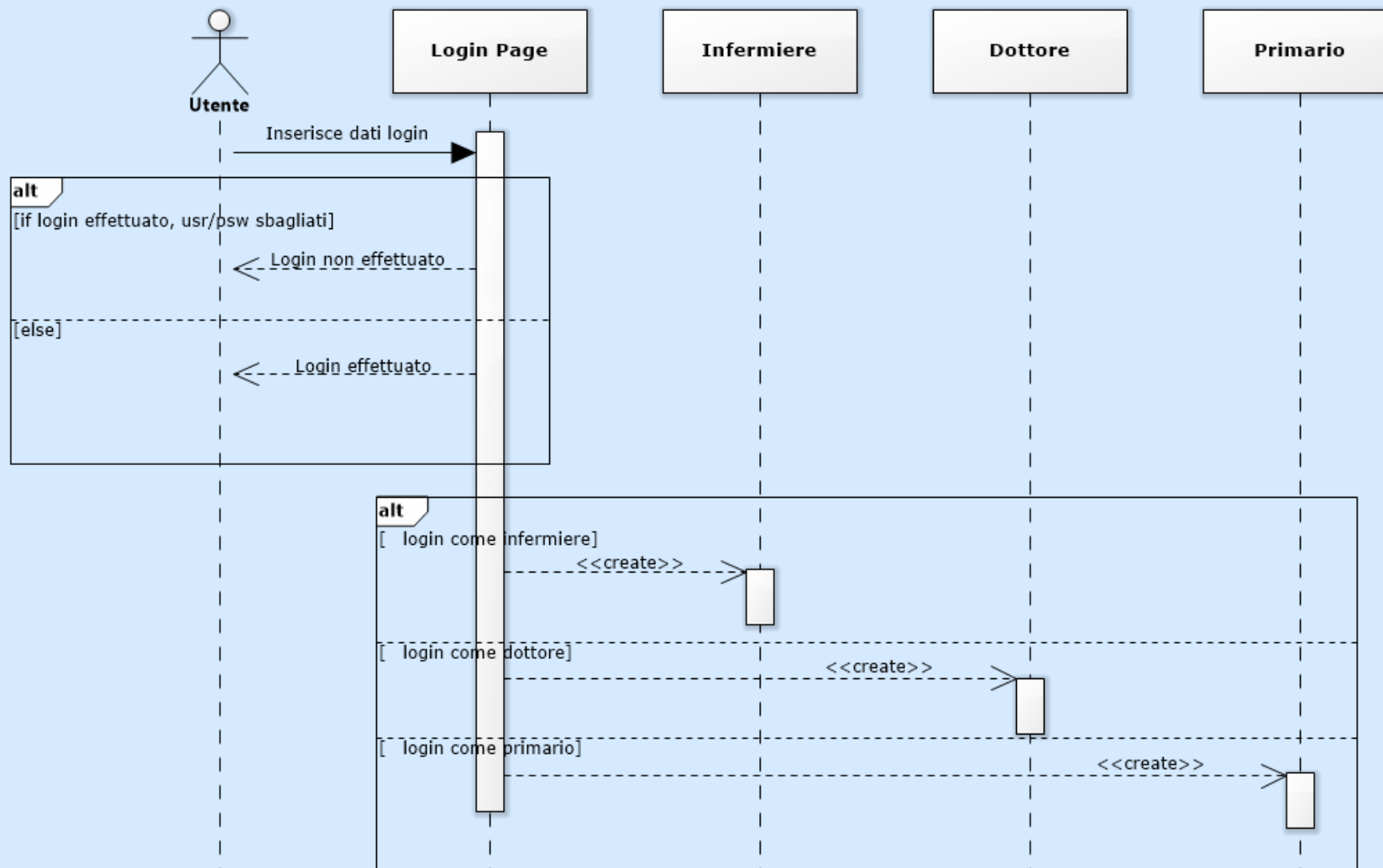
- **[R4]** La gestione dei segnali in arrivo dal sistema di monitoraggio parametri vitali
 - **[R4.1]** Pressione sistolica e diastolica ogni 2 minuti, frequenza cardiaca ogni 5 minuti e temperatura ogni 3 minuti
- **[R5]** In base ai segnali vitali e all'elettrocardiogramma verranno gestiti eventuali allarmi di 3 livelli di gravità
 - **[R5.1]** Aritmia (livello 1), tachicardia (livello 1), flutter (livello 3), ipertensione (livello 2), ipotensione (livello 2), ipertermia (livello 2), ipotermia (livello 2)
- **[R6]** In caso di allarmi i medici dovranno “spegnere l'allarme” e scrivere eventuali note
 - **[R6.1]** In caso di livello 1 dovrà essere spento in 3 minuti, 2 minuti per il livello 2 e 1 minuti per il livello 3

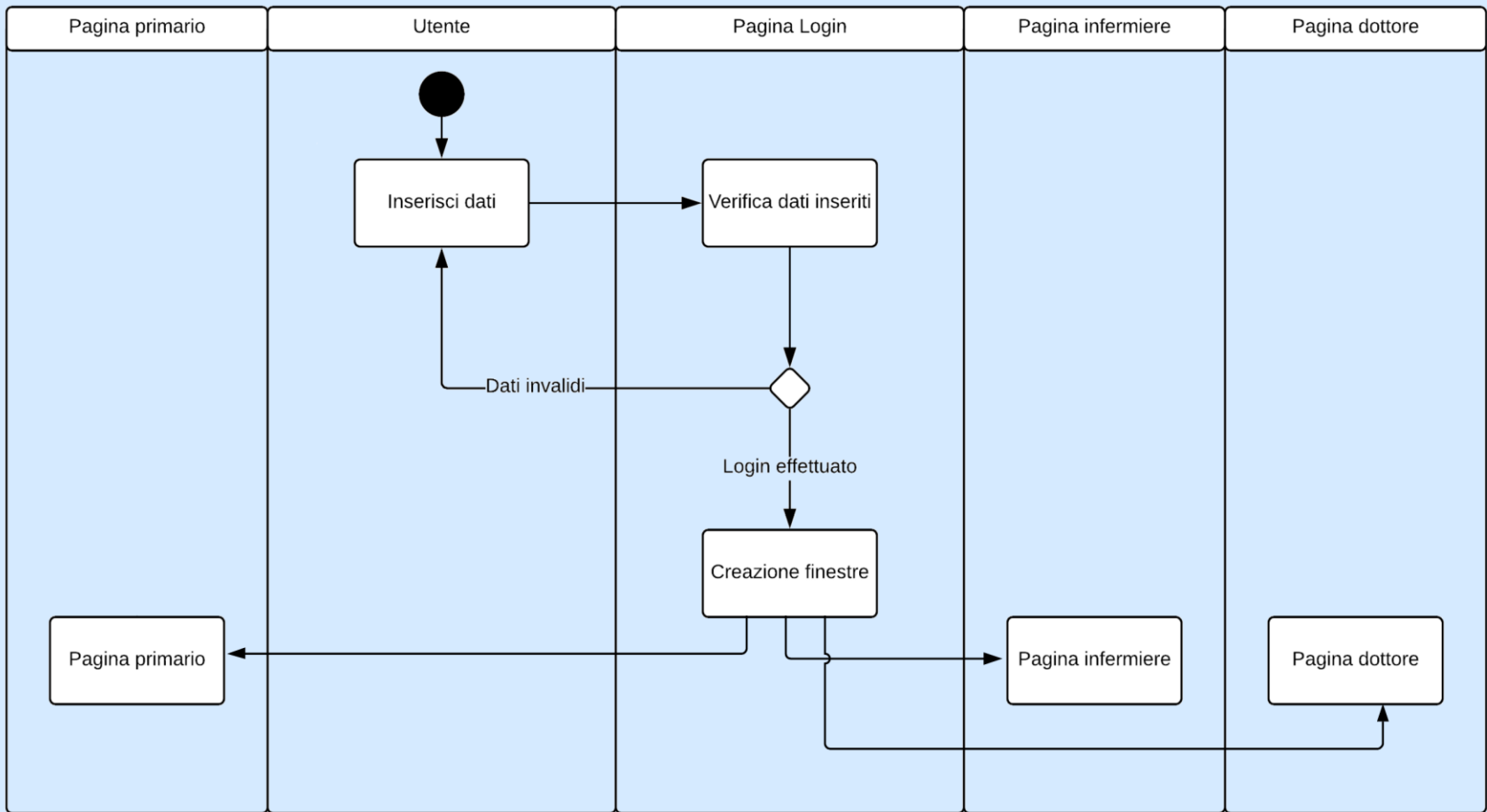
- **[R7]** Per medici ed infermieri sarà possibile visualizzare i parametri monitorati delle ultime 2 ore e le somministrazioni degli ultimi 2 giorni e saranno aggiornati in tempo reale
- **[R8]** Il sistema di visualizzazione parametri vitali per gli ultimi 15 minuti sarà sempre in funzione
- **[R9]** Il primario potrà visualizzare e stampare un report riassuntivo del ricovero e chiudere le cartelle cliniche
- **[R10]** Il sistema permette la visualizzazione dei ricoveri pregressi

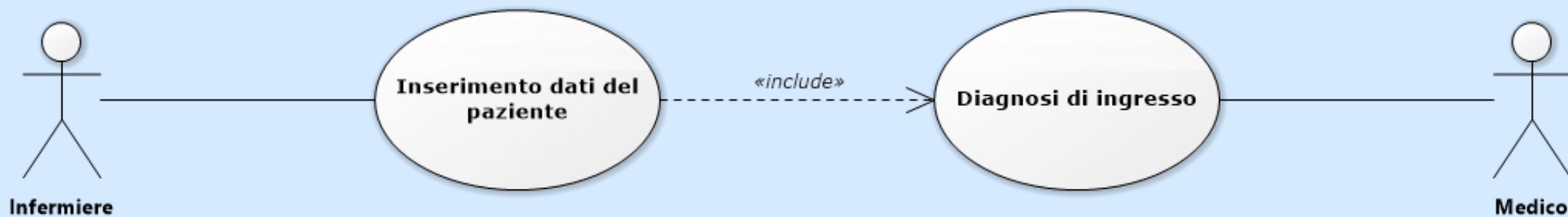


Caso d'uso: autenticazione

- Gli utenti accedono al programma previa autenticazione. Per ogni tipologia di login verrà creata la finestra desiderata

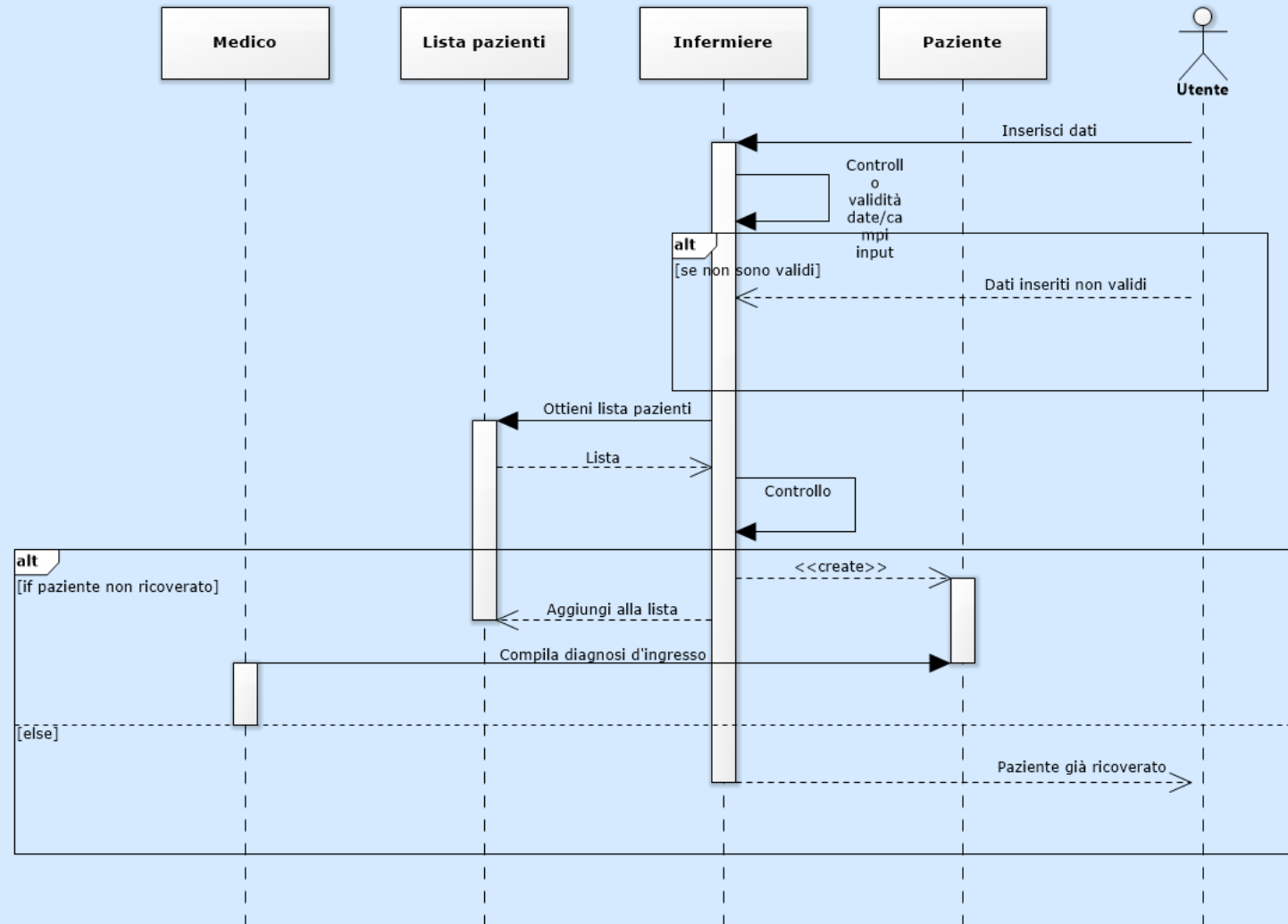


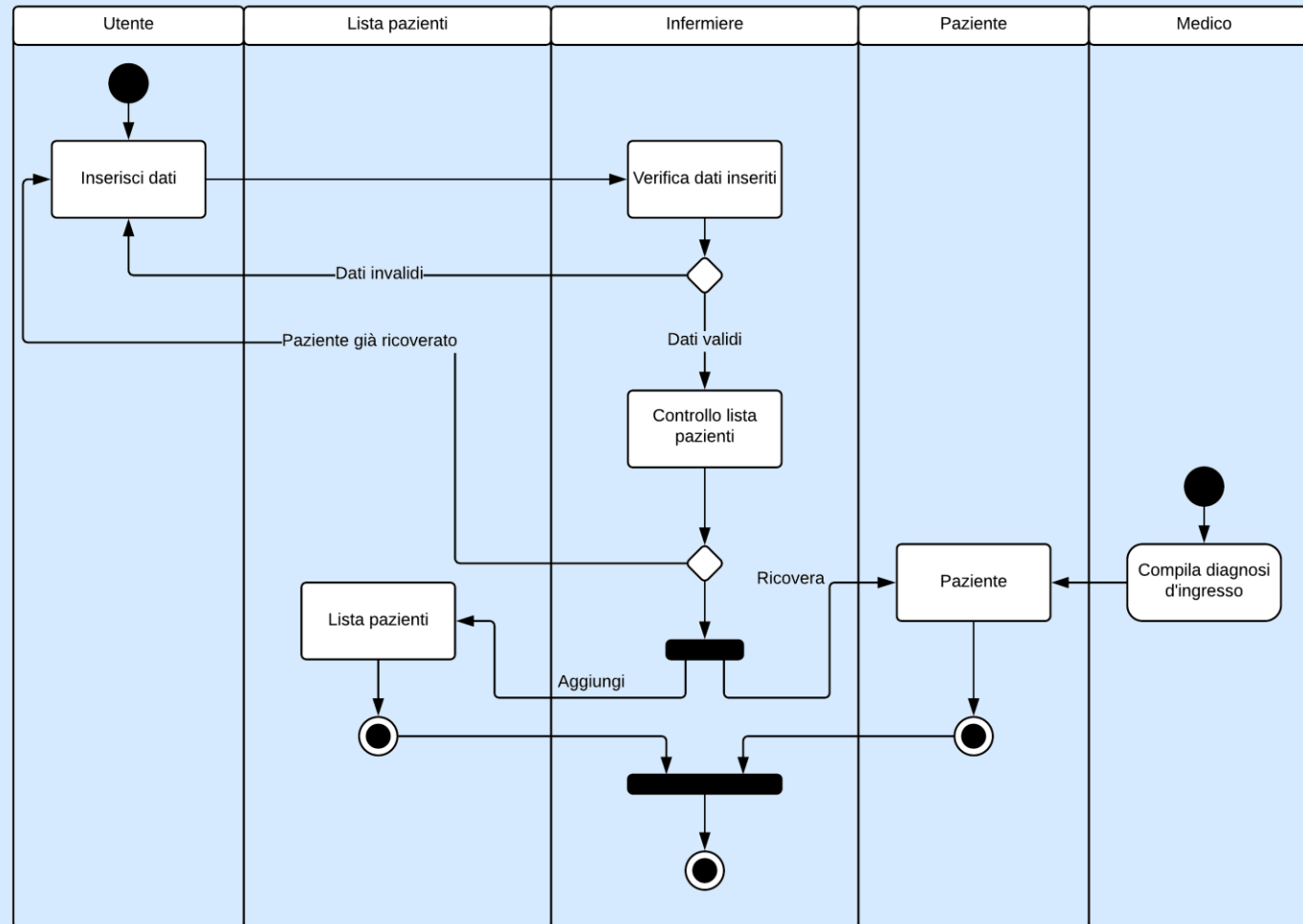




Caso d'uso: ricovero

- Al momento del ricovero, l'infermiere inserisce i dati del paziente (nome, cognome, luogo di nascita, data di nascita)
- Avvenuta la registrazione il medico integra con la diagnosi d'ingresso



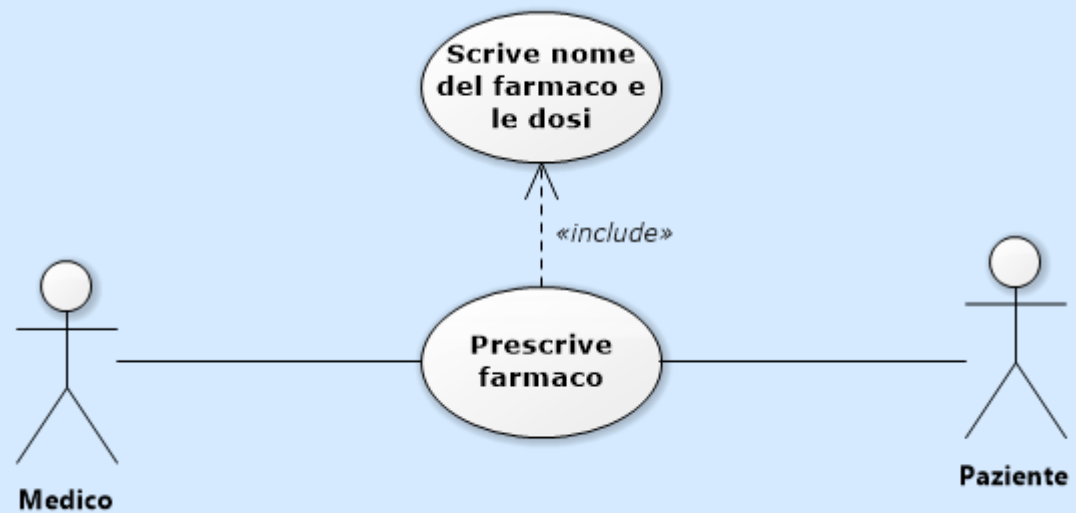
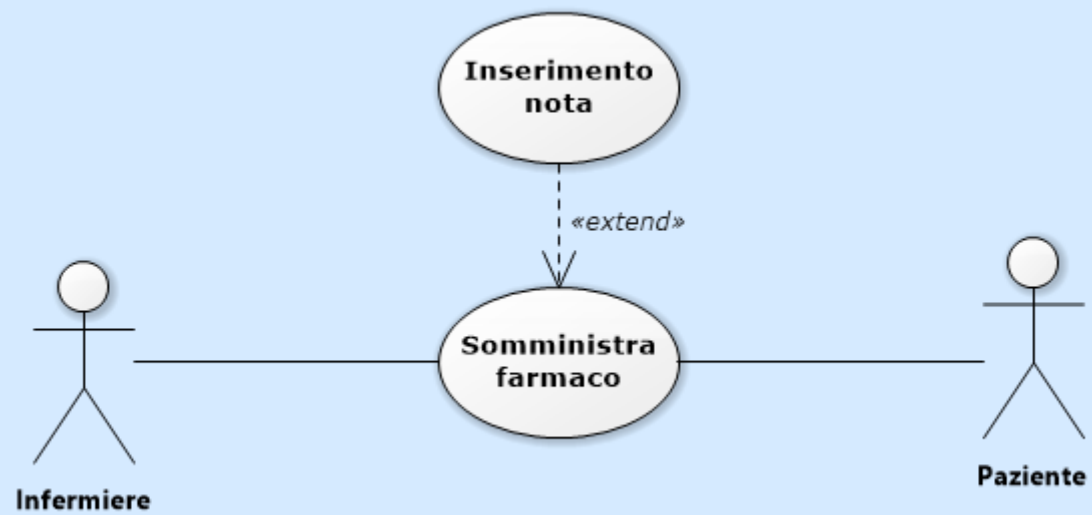


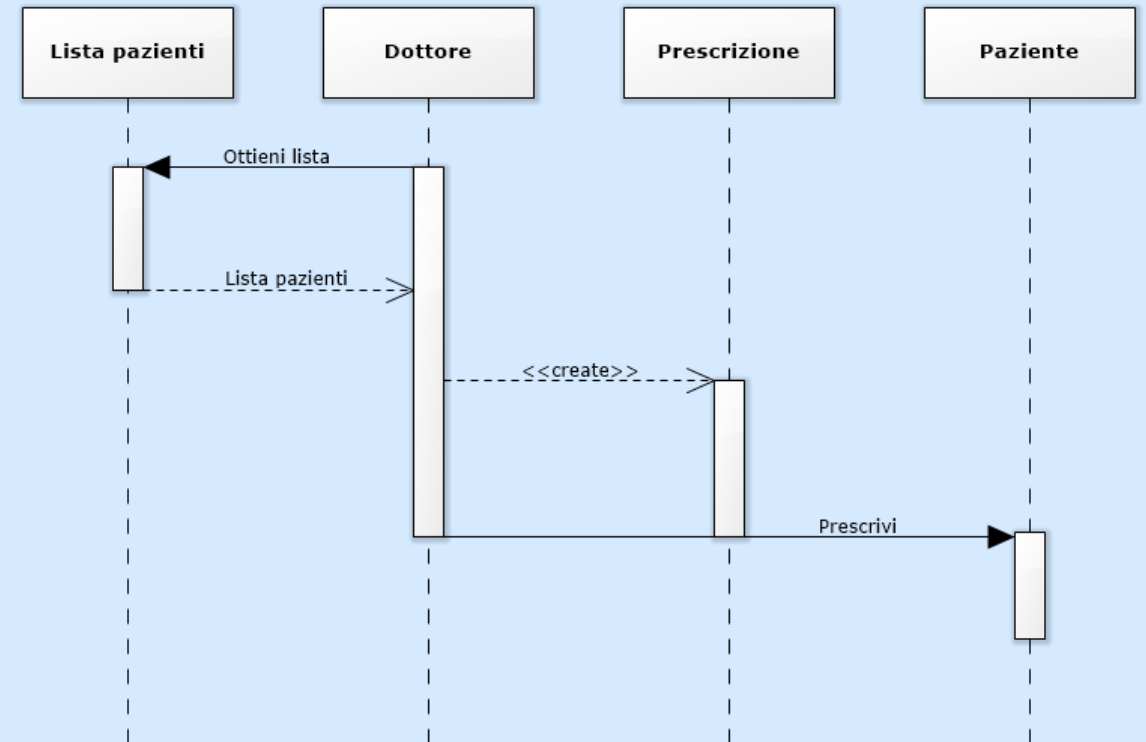
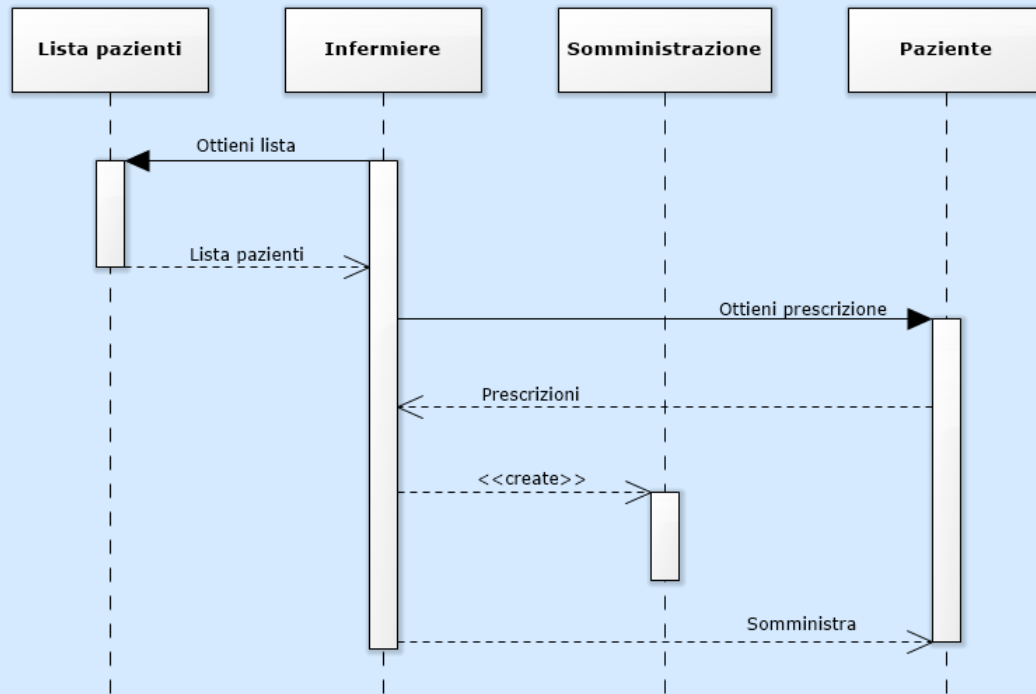
Caso d'uso: prescrizioni e somministrazioni

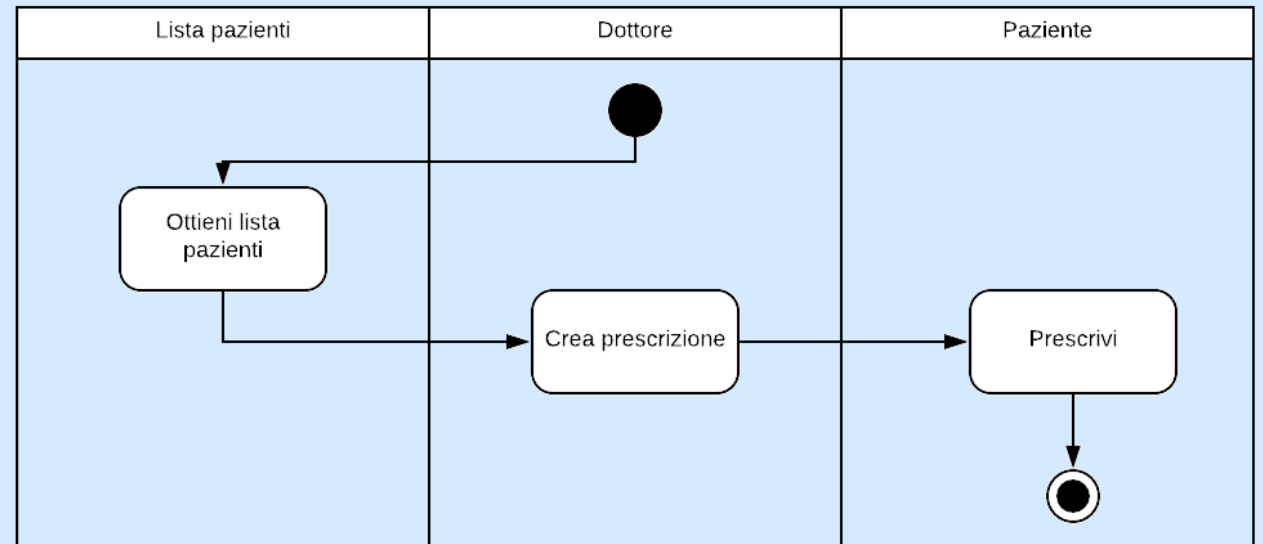
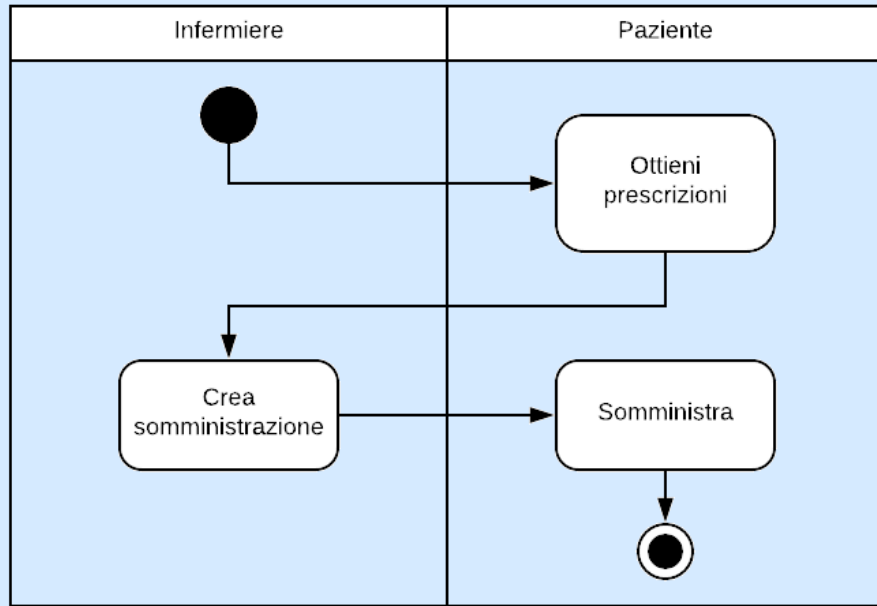
Il medico può sottoscrivere una prescrizione di un farmaco memorizzando i dati necessari (nome farmaco, dose giornaliera, durata terapia e quantità per dose)

L'infermiere procede poi con la somministrazione del farmaco in base alle prescrizioni. Durante questa fase può inserire eventuali note aggiuntive al momento della somministrazione

Entrambi poi potranno consultare le somministrazioni fino a due giorni precedenti per ogni paziente



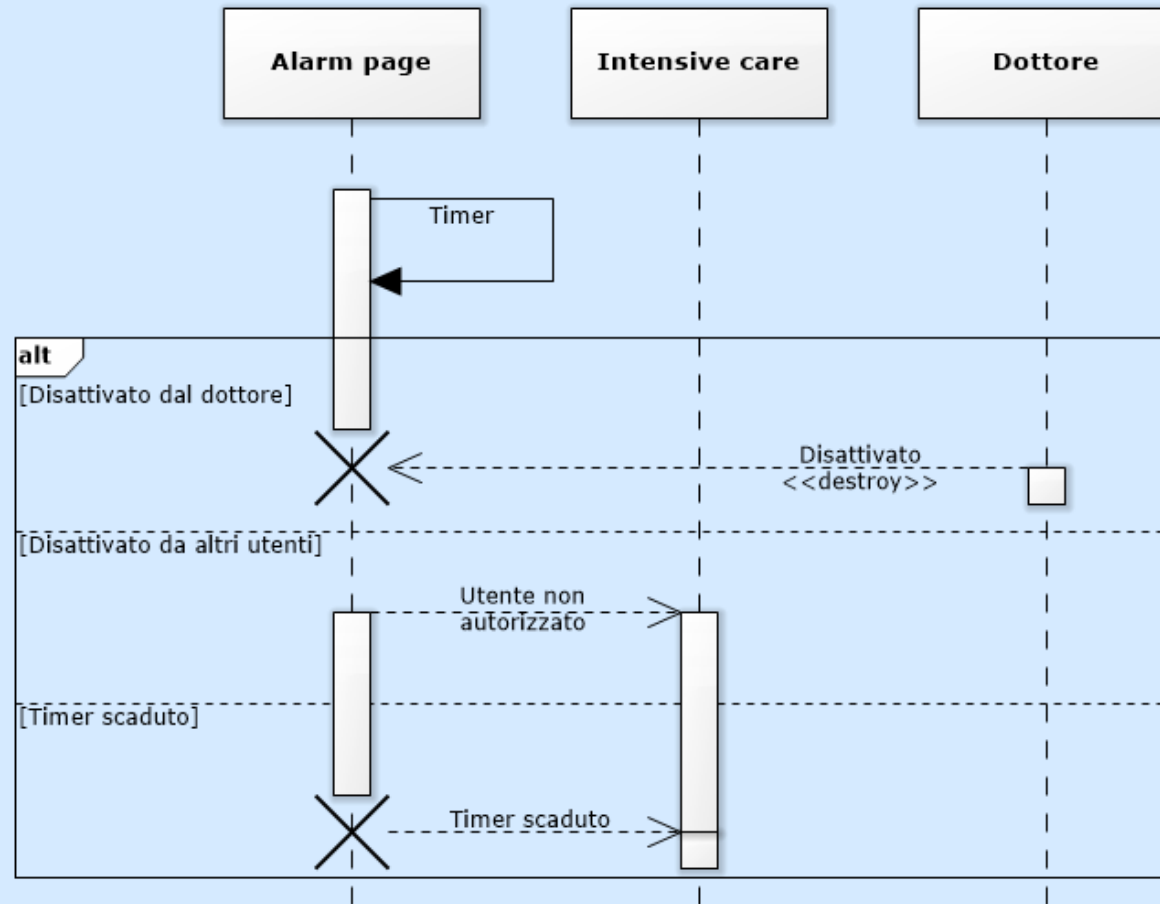


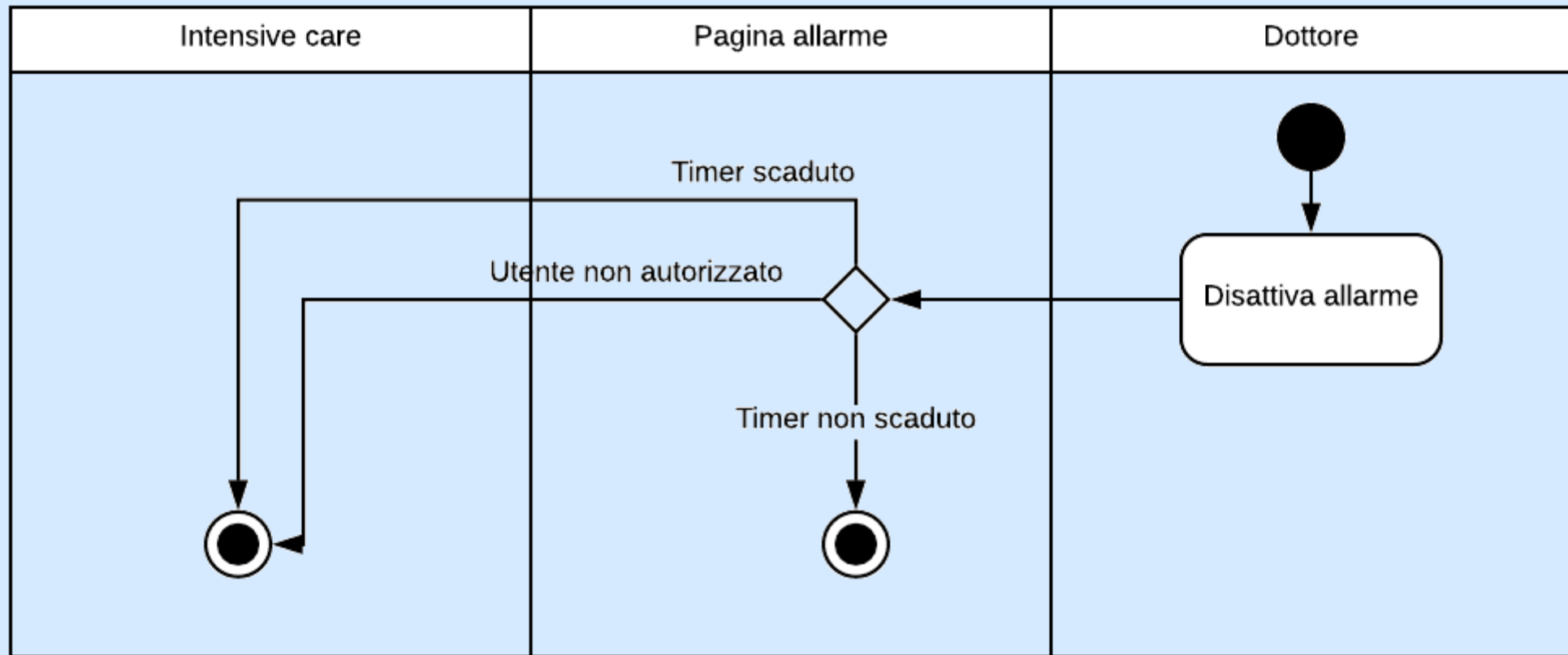


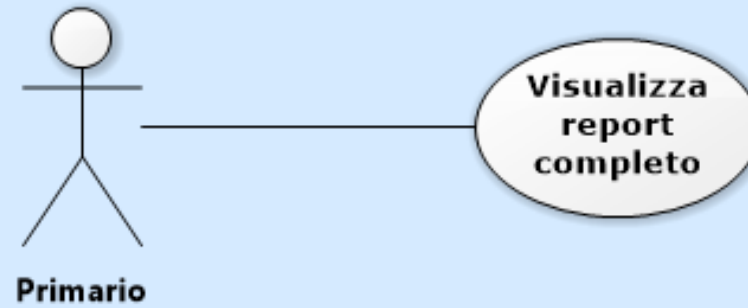


Caso d'uso: allarmi

- Il sistema è progettato per far partire allarmi se i dati vitali dei pazienti sono maggiori o minori rispetto alle soglie consentite. Solo il medico può disattivare tale allarme. Può inoltre aggiungere note aggiuntive su possibili operazioni effettuate sul paziente.

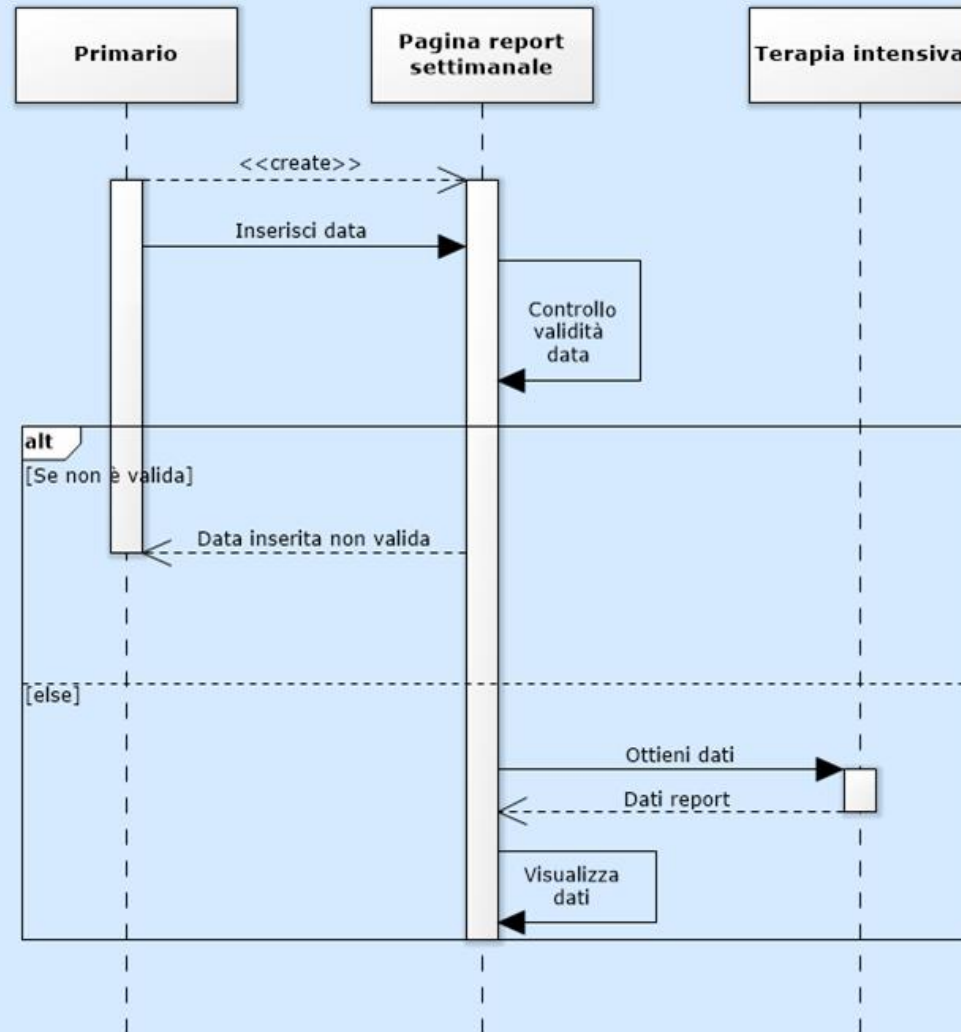


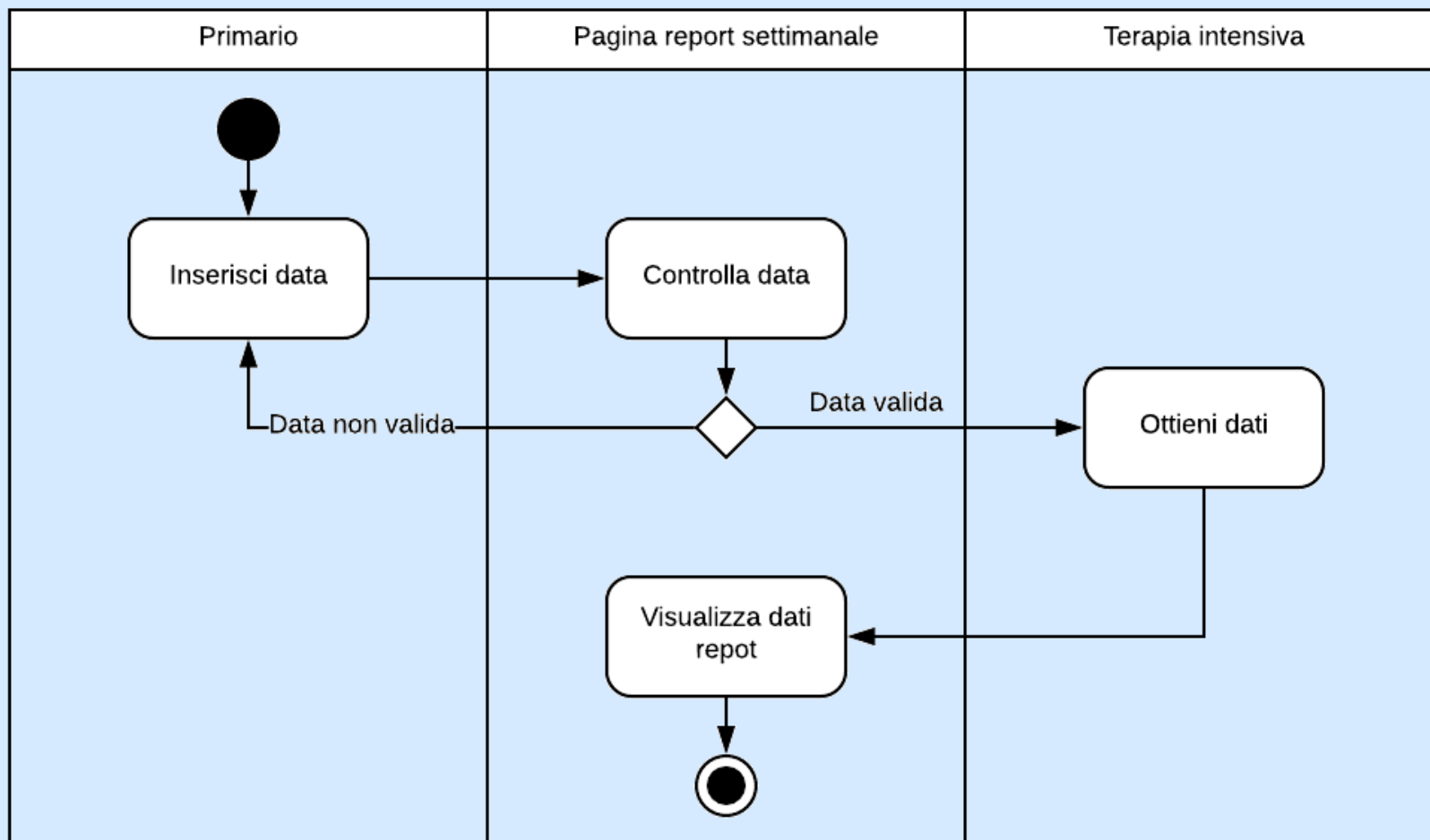


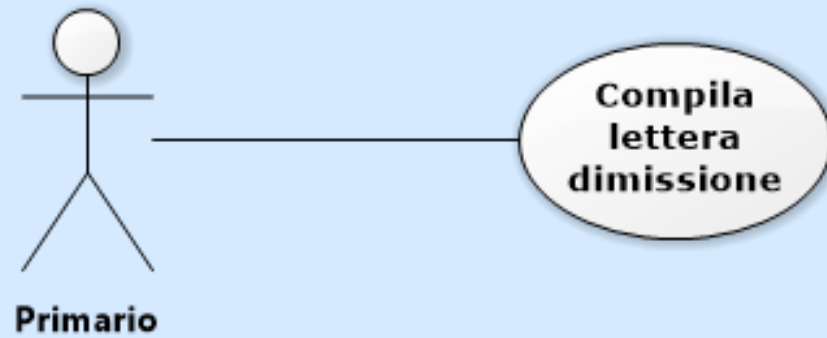


Caso d'uso: report
settimanale

- Il primario può visualizzare un report settimanale della situazione di tutti i pazienti ricoverati

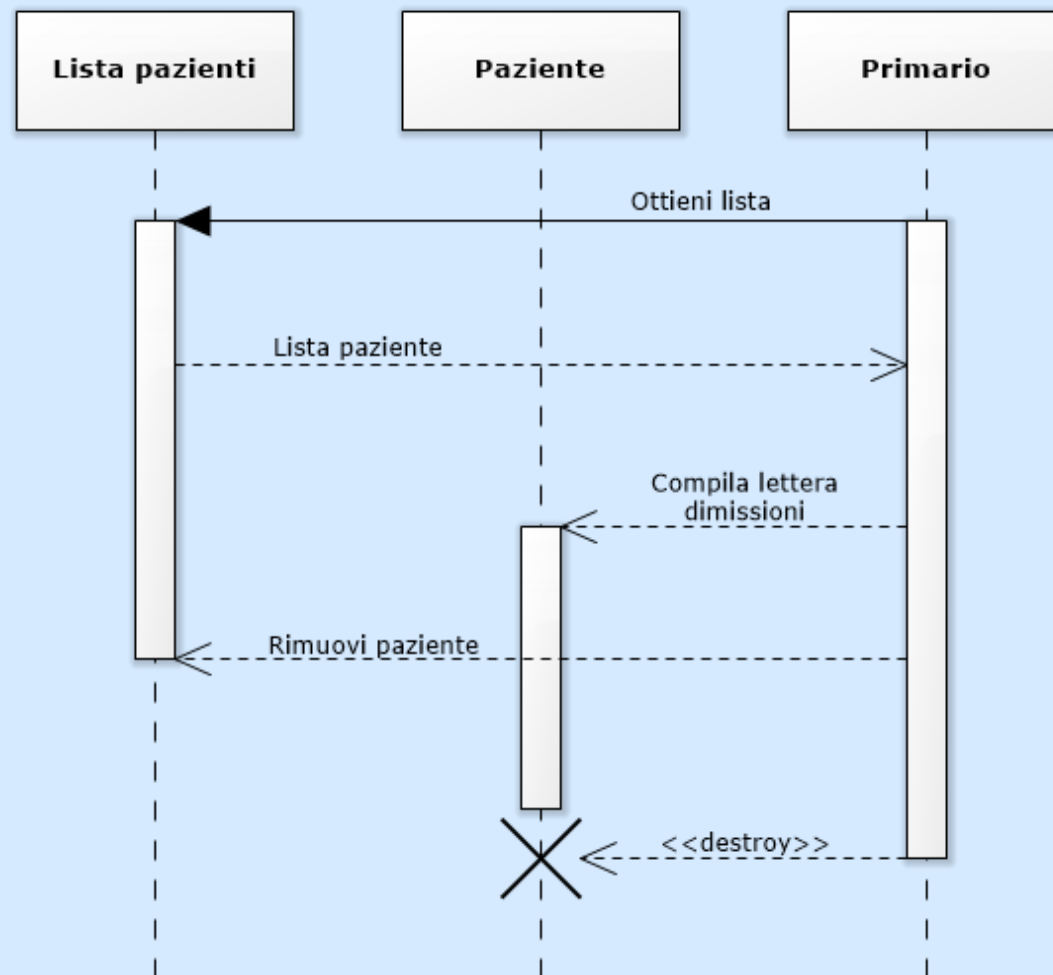


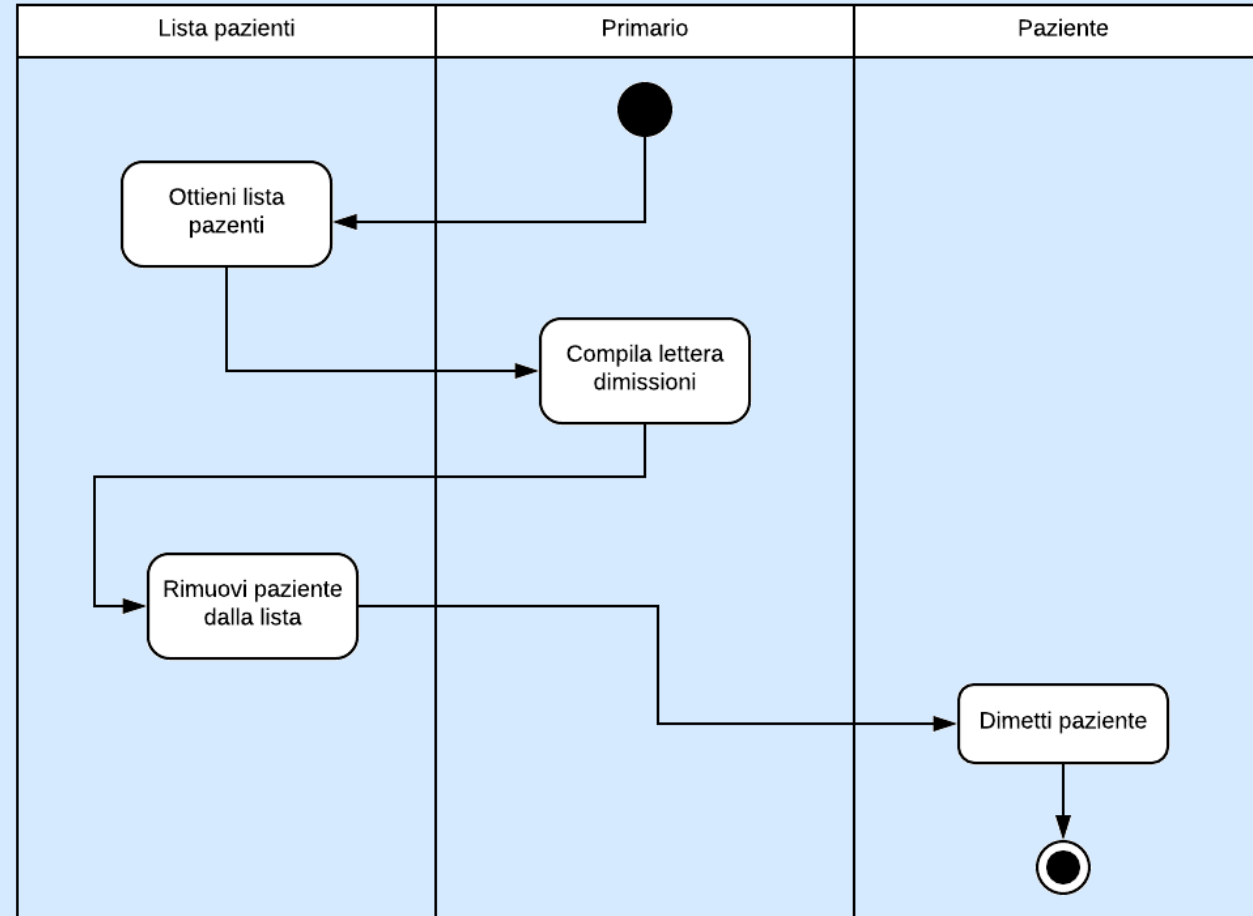


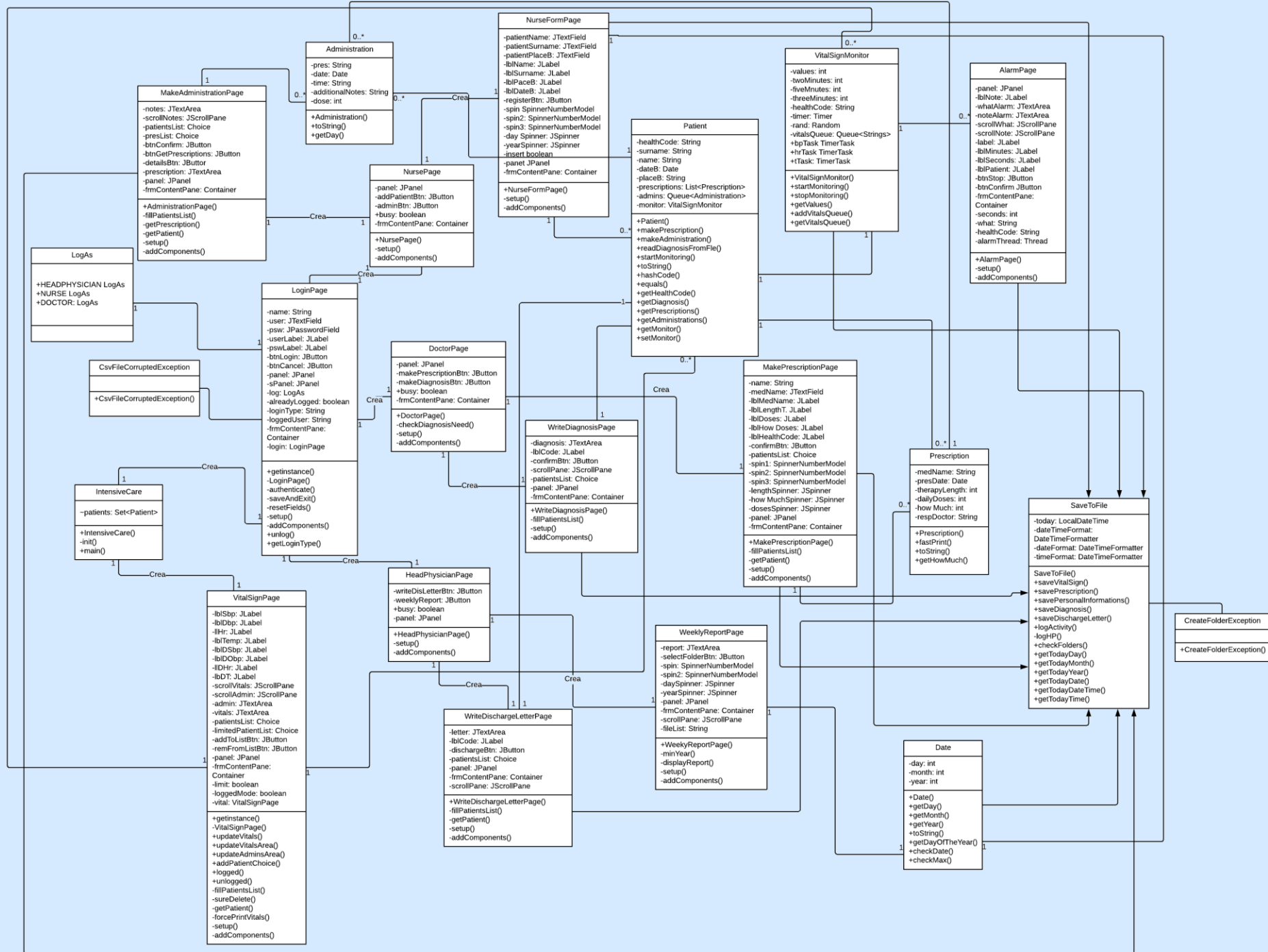


Caso d'uso: dimissione

- Il primario compila la lettera di dimissione portando ufficialmente a termine il ricovero del paziente

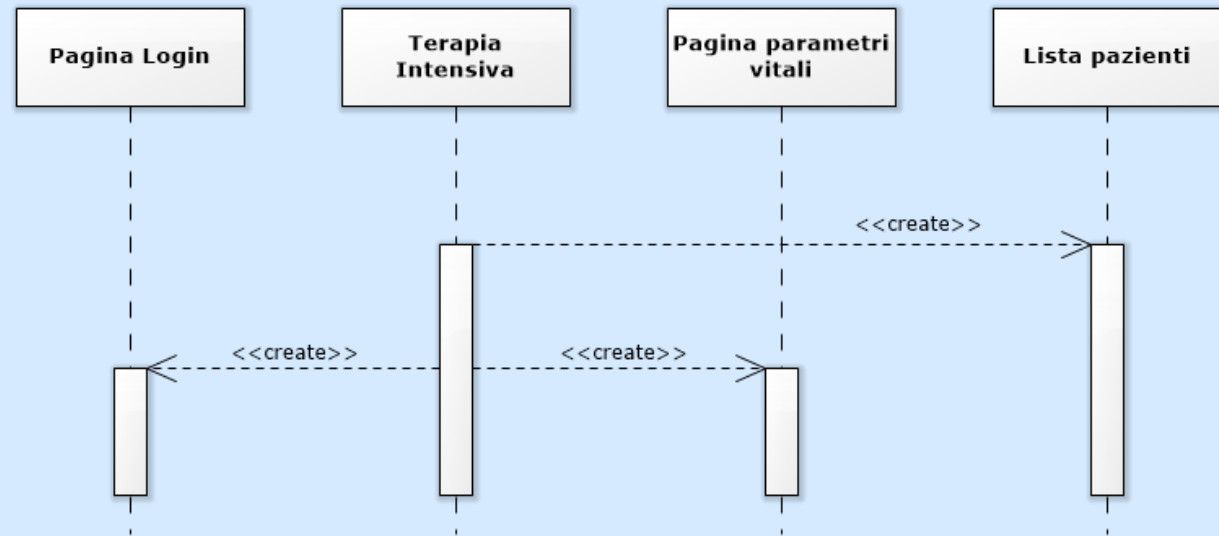




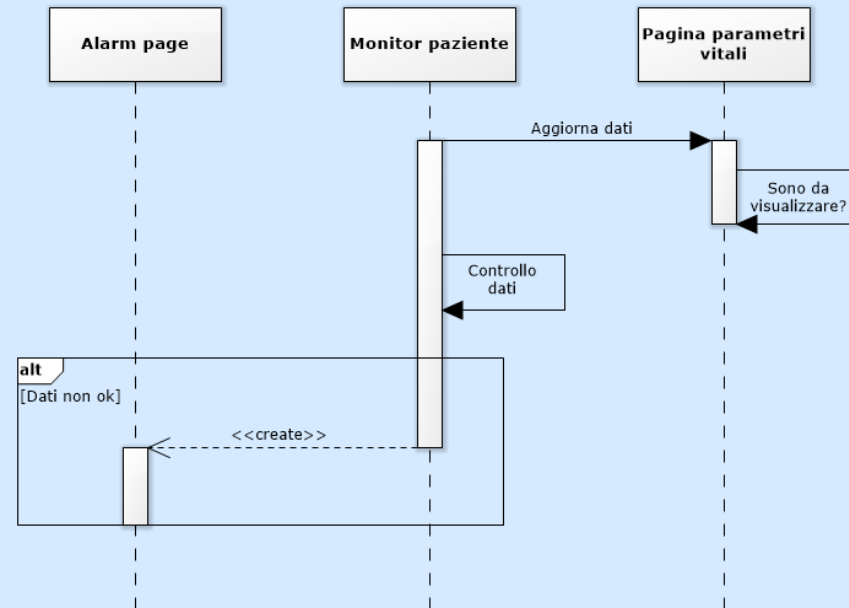


Funzionamento

All'avvio del programma viene creata la lista dei pazienti e viene caricato (se presente) il file di database in cui sono stati serializzati i dati più recenti. Dopodiché vengono create le due finestre di login e di visualizzazione dei parametri vitali. Segue poi la procedura di login precedentemente illustrata dove, in base alla tipologia di login, viene creata la finestra desiderata.



Se i dati vitali di un paziente sfiorano i limiti prestabiliti viene creata una finestra di allarme. Può essere disattivata solamente da un dottore.



Tutti i dati del programma (ricoveri, dati pazienti, allarmi, prescrizioni, somministrazioni, dimissioni) vengono salvati nelle relative cartelle dei pazienti e nella cartella dei file log. Questi ultimi sono utilizzati durante la visualizzazione del report settimanale del primario.

Il programma termina la sua esecuzione se e solo se la finestra di login viene chiusa. Prima della chiusura la lista contenente i pazienti viene serializzata e quindi salvata in un file (.ser). Questo file verrà poi utilizzato per la deserializzazione della lista nel metodo init() della classe IntensiveCare.

Sviluppo

Dopo aver analizzato i requisiti abbiamo deciso che il miglior modo per sviluppare fosse applicare un metodo ***agile*** in quanto ci garantiva un'implementazione rapida delle varie funzionalità con conseguente test per verificarne il corretto funzionamento. Ci è venuto in aiuto il sistema ***git*** in quanto ci ha permesso di condividere il codice in maniera veloce e trasparente. Il tutto è stato scritto utilizzando l'ambiente di sviluppo integrato ***Eclipse*** con il plugin ***EGit***.

Pattern

Abbiamo utilizzato il pattern ***Observer*** tra la pagina di visualizzazione dei dati vitali ed i monitor di ogni singolo paziente. Abbiamo però adattato il pattern alle nostre esigenze in quanto avevamo un solo oggetto che andava notificato (la pagina di visualizzazione). Ogni volta che un nuovo dato viene registrato dal monitor di un paziente viene chiamato il metodo `updateVitals()` ed il metodo `updateVitalsArea()` per aggiornare graficamente i dati visualizzati e l'area in cui sono presenti i dati vitali registrati negli ultimi minuti. Inoltre, quando viene fatta una nuova somministrazione al paziente viene chiamato il metodo `updateAdminsArea()` che si occupa di aggiornare graficamente l'elenco delle somministrazioni. Verranno aggiornati graficamente solamente se il paziente è correntemente visualizzato.

Un altro pattern utilizzato è il ***Singleton***. Le classi che lo adottano sono LoginPage e VitalSignPage. Questo pattern ci garantisce la presenza di una sola istanza. Anche la classe SaveToFile era stata inizialmente pensata come singleton ma, non avendo bisogno di instanziarla come oggetto, abbiamo reso tutti i metodi pubblici e statici.

Fase di test

Avendo utilizzato un approccio agile abbiamo effettuato frequentemente dei test per verificare il giusto funzionamento di ogni funzionalità introdotta. Per alcune funzionalità abbiamo anche scritto appositamente dei file di log fittizi. Siamo anche andati a creare appositamente dei casi limite per sincerarci del corretto comportamento da parte del software e della giusta gestione delle eccezioni. Molto utile è stato anche far testare il programma a persone che non erano a conoscenza del funzionamento e grazie al loro feedback l'abbiamo reso il più intuitivo possibile. Grazie a questi test siamo stati in grado di correggere più volte il codice sia dal punto di vista logico/strutturale sia dal punto di vista grafico.