

Relazione di Ingegneria del Software:

Gestione Web di uno Studio Dentistico

Realizzato da:

Raffaele Esposito

Giuseppe Di Lorenzo

Agostino Vallefuego

Contents

- 1 Introduzione
- 2 Attori e Requisiti
 - 2.1 Attori
 - 2.2 Requisiti Funzionali
 - 2.3 Requisiti Non Funzionali
- 3 Diagrammi dei Casi d'Uso
- 4 Specifiche dei Casi d'Uso
- 5 Diagramma delle Classi di Analisi
- 6 Diagramma dei Package d'Analisi
- 7 Diagrammi delle Sequenze di Analisi
- 8 Diagramma delle Attività
- 9 Diagramma delle Transizioni di Stato
- 10 Diagrammi delle Classi di Design
- 11 Pattern
- 12 Diagramma dei Package di Design
- 13 Diagrammi delle Sequenze di Design
- 14 Diagramma dei Componenti
- 15 Diagramma del Deployment
- 16 BPMN

17 Implementazione

17.1 Architettura di Riferimento

17.2 Dettagli di Implementazione

18 Test

1. Introduzione

Si vuole progettare un'applicazione web per la gestione web di uno studio dentistico, utilizzabile sia dai dipendenti che dai pazienti. Il sistema dovrà permettere ai clienti di prenotare un appuntamento, di pagare online e di visualizzare i dati del proprio profilo e per alcuni di essi anche attuare una modifica; mentre ai dipendenti di gestire tutti gli aspetti in merito agli appuntamenti e alle fatture dei clienti oltre alla gestione materiali dello studio dentistico. Il dottore, inoltre, potrà controllare la lista dei dipendenti, la loro registrazione e autorizzare eventuali modifiche effettuate dai dipendenti. Per la realizzazione dell'applicazione web si è deciso di usare il pattern architetturale **Model-View-Controller** che sarà mostrato successivamente. **Il modello di processo scelto è quello iterativo** e tale documento rappresenta il risultato della prima iterazione.

2. Attori e Requisiti

2.1 Attori

- ☐ Dottore
- ☐ Assistente alla Poltrona
- ☐ Segretaria
- ☐ Paziente
- ☐ Tempo
- ☐ Guest
- ☐ Utente

2.2 Requisiti Funzionali

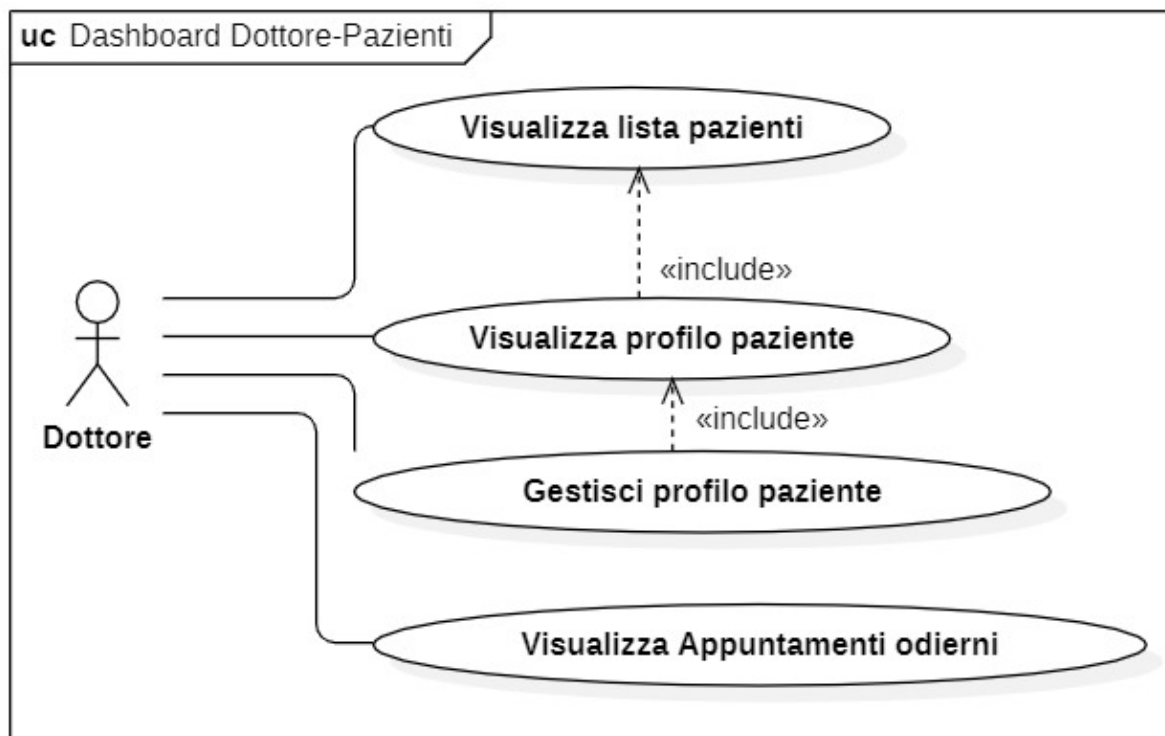
- ☐ Gli utenti non loggati saranno in grado di registrarsi e di effettuare il login
- ☐ Tutti gli utenti loggati saranno in grado di effettuare il logout
- ☐ I dipendenti potranno visualizzare la lista dei pazienti
- ☐ I dipendenti potranno visualizzare il profilo di uno specifico paziente
- ☐ I dipendenti potranno visualizzare la lista degli appuntamenti e la lista di quelli odierni
- ☐ Il dottore potrà eliminare il profilo di un utente
- ☐ Il dottore e l'assistente potranno aggiungere la cartella clinica di un paziente

- ☐ Il dottore e l'assistente potranno effettuare modifiche sulla cartella clinica di un paziente
- ☐ Il dottore e l'assistente potranno eliminare la cartella clinica di un paziente
- ☐ L'assistente potrà richiedere l'approvazione di modifica per alcune azioni in cui vi `e bisogno dell'autorizzazione da parte del dottore
- ☐ Il dottore potrà esaminare le modifiche proposte dall'assistente
- ☐ Il dottore e l'assistente potranno visualizzare l'elenco dei materiali e occuparsi della loro gestione
- ☐ Il dottore potrà generare un link per permettere ai dipendenti di registrarsi
- ☐ Il paziente potrà prenotare un appuntamento verificando la disponibilità e aggiungendo la motivazione della visita
- ☐ Il paziente potrà decidere se attivare la notifica di remind per l'appuntamento
- ☐ Il paziente potrà eliminare la prenotazione precedentemente fissata
- ☐ Il paziente potrà visualizzare tutte le sue prenotazioni attive
- ☐ Il paziente potrà modificare alcuni dei suoi dati dalla sezione profilo
- ☐ Il paziente potrà visualizzare il saldo da pagare per il trattamento
- ☐ Il paziente potrà effettuare il pagamento online per la terapia ricevuta
- ☐ La segretaria potrà aggiungere un appuntamento per uno specifico paziente
- ☐ La segretaria potrà spostare un appuntamento di uno specifico paziente
- ☐ La segretaria potrà eliminare un appuntamento di uno specifico paziente
- ☐ La segretaria potrà aggiungere una fattura nel database dopo il pagamento da parte del paziente
- ☐ La segretaria potrà modificare una fattura nel database
- ☐ La segretaria potrà eliminare una fattura dal database
- ☐ La segretaria potrà inviare una fattura al paziente
- ☐ La segretaria potrà visualizzare il saldo da pagare di un paziente

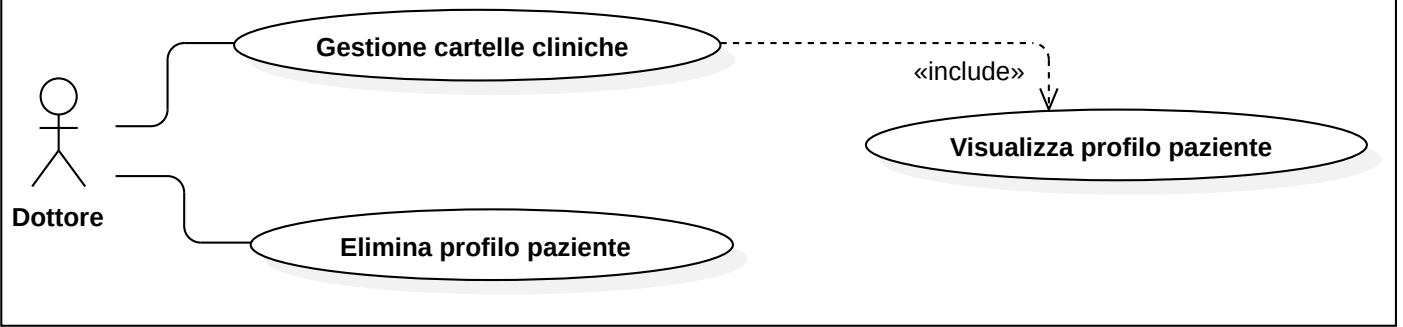
2.3 Requisiti Non Funzionali

- ☐ Il sistema conserverà le password sotto forma di hash
- ☐ Sia i pazienti che i dipendenti potranno utilizzare il sistema attraverso un sito web
- ☐ La connessione al sito web avviene solo mediante HTTPS
- ☐ L'interfaccia web sarà moderna e responsiva

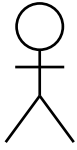
3. Diagrammi dei Casi d'Uso



uc Gestisci profilo paziente



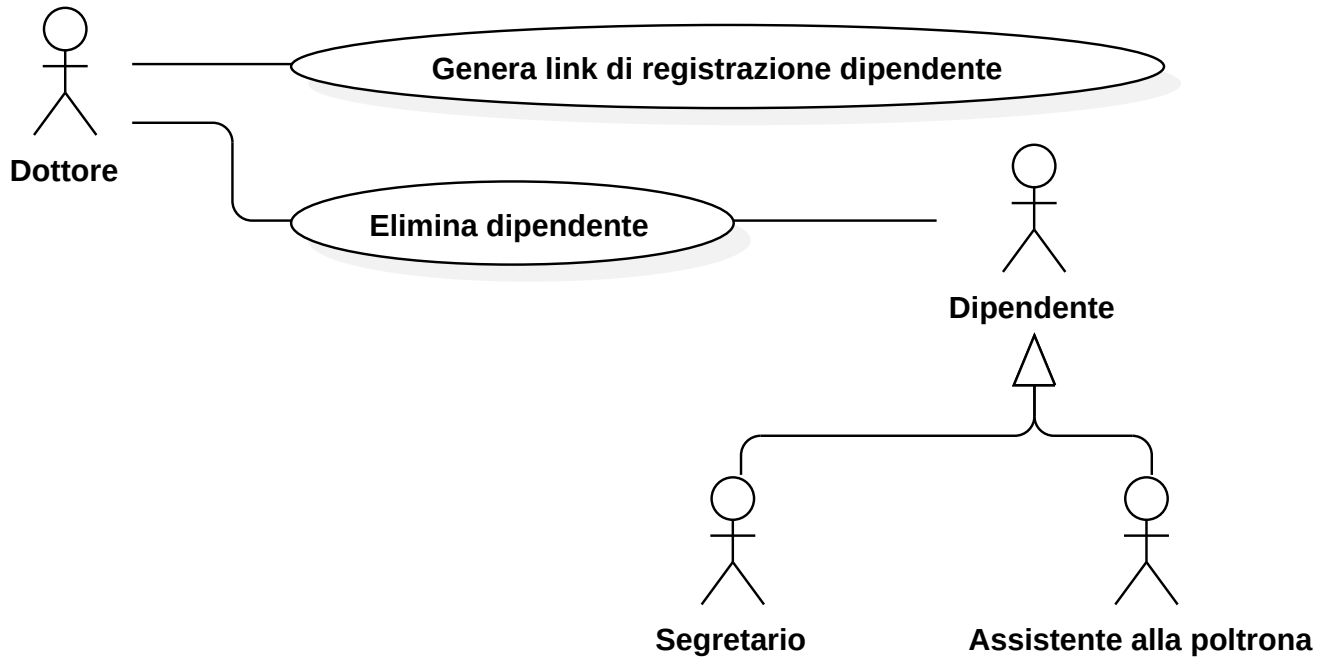
uc Autorizzazione



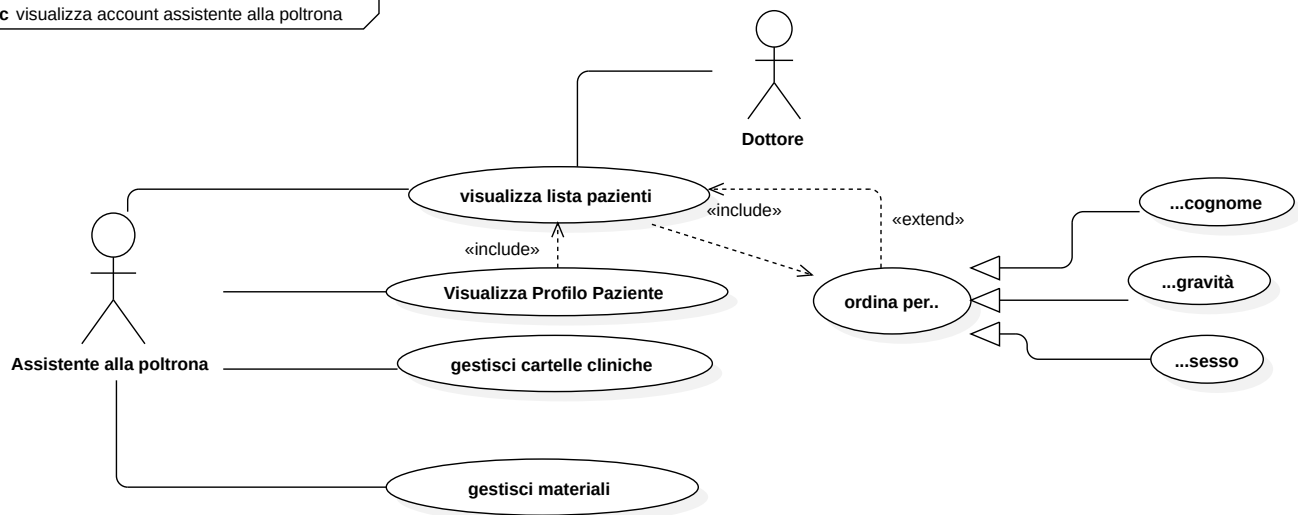
Dottore

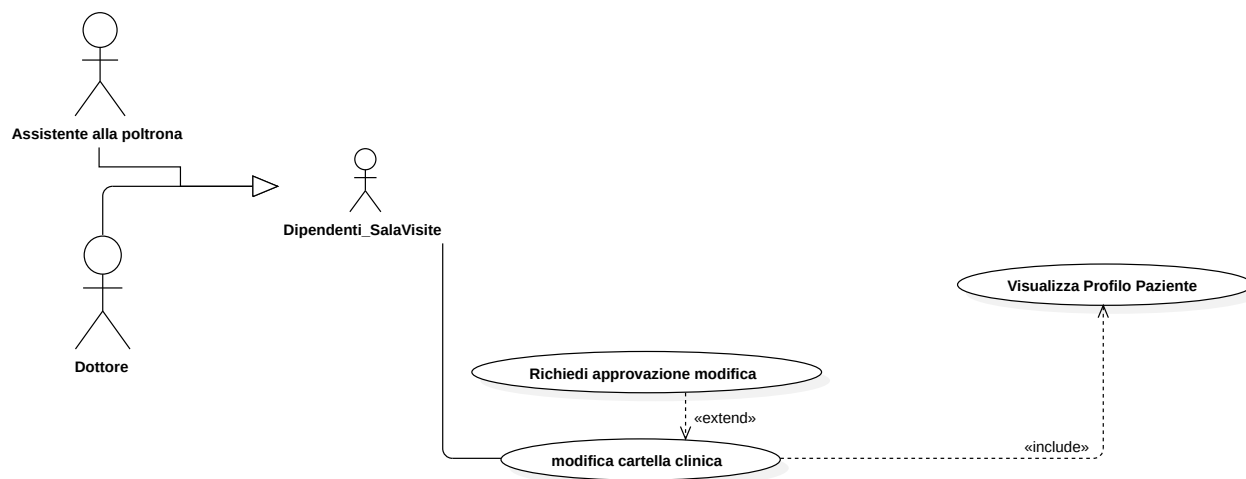
Esamina modifica proposta

uc Nuovo dipendente

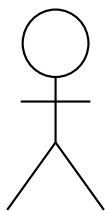


uc visualizza account assistente alla poltrona





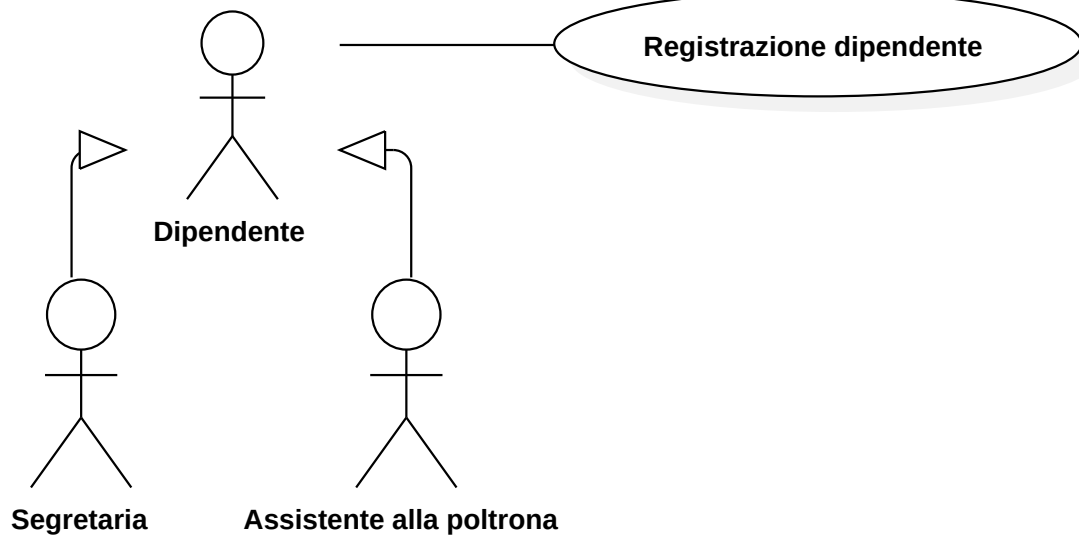
uc Richiedi Approvazione Modifica



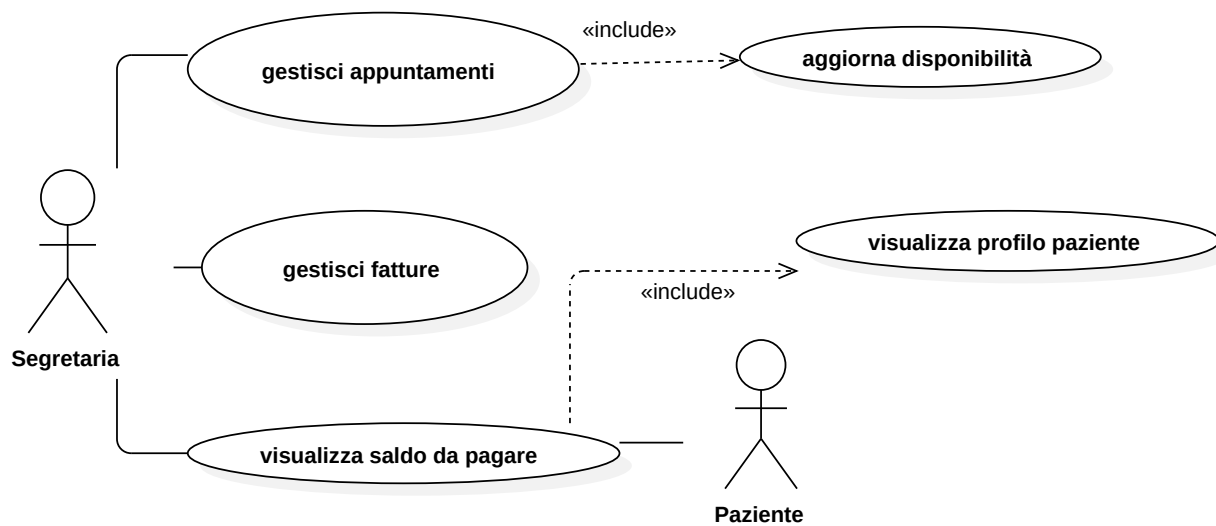
Assistente alla poltrona

Invia Richiesta di modifica al Dottore

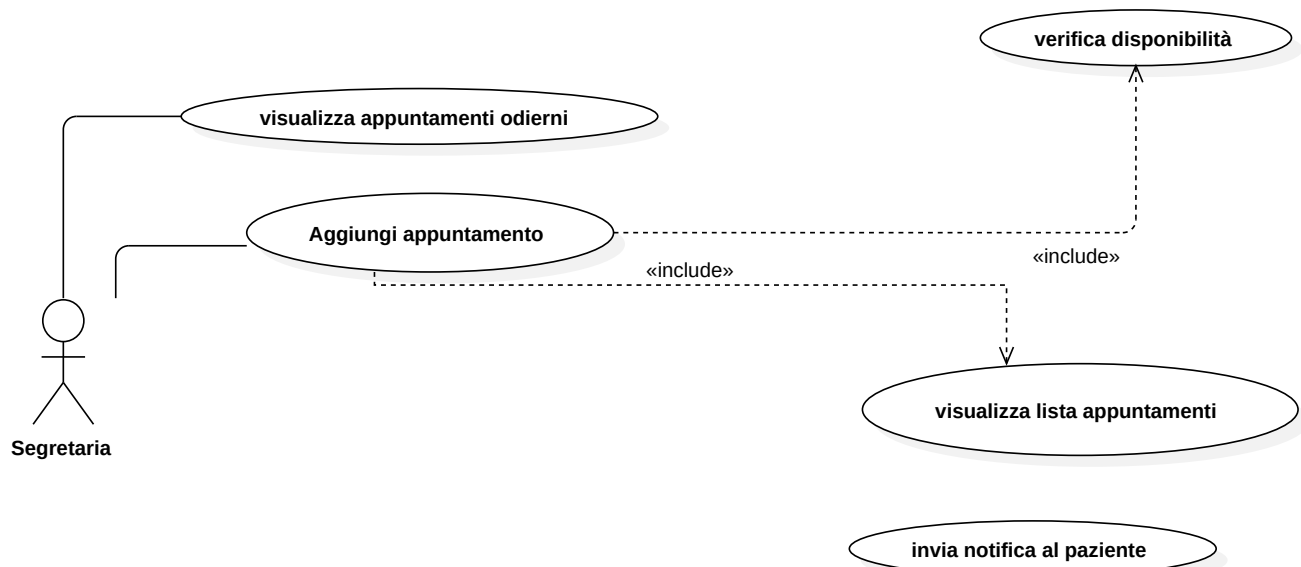
uc Registrazione personale



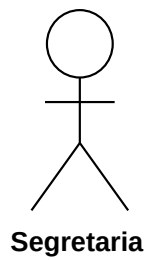
uc visualizza account segretaria



uc gestione appuntamenti

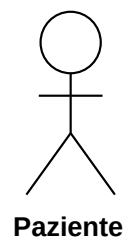


uc gestione pagamenti

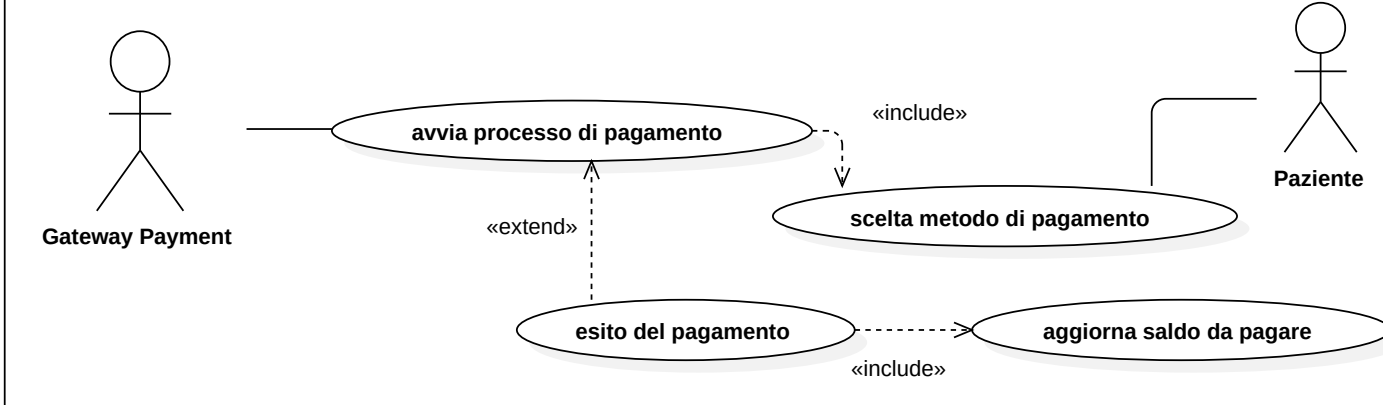


inserisci saldo da pagare

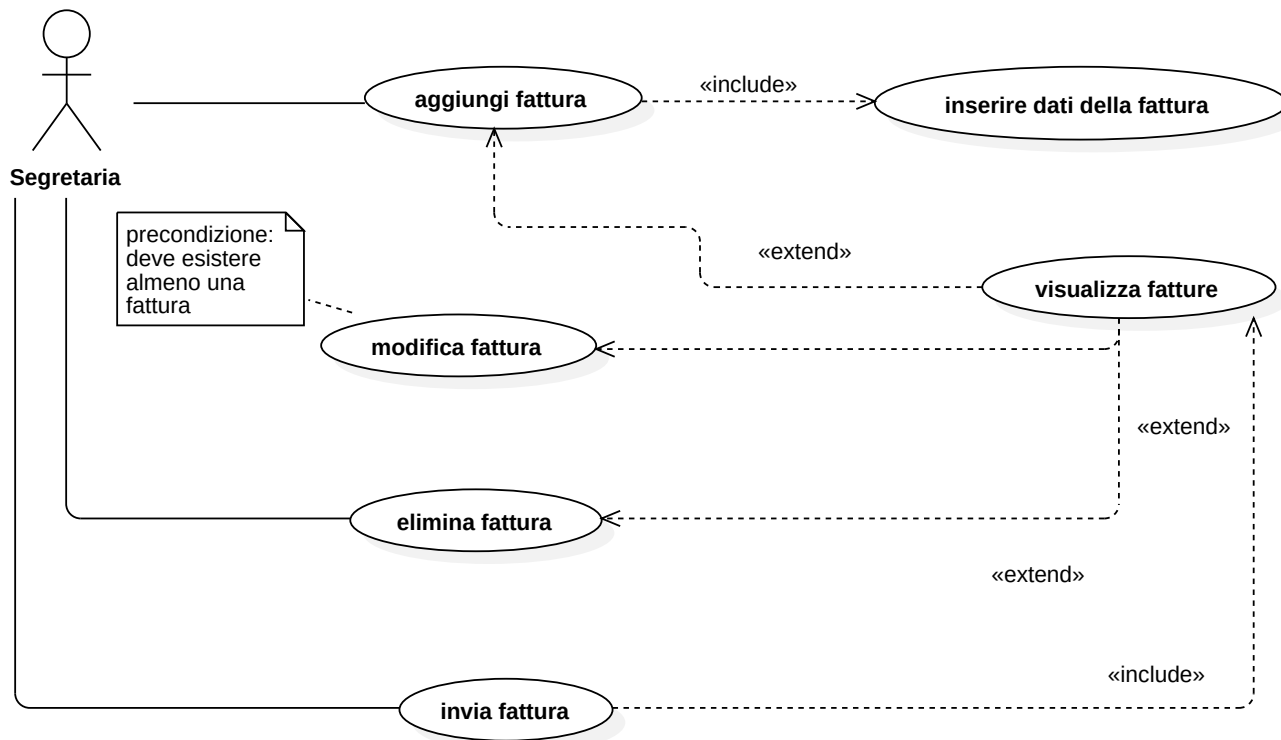
visualizza saldo da pagare

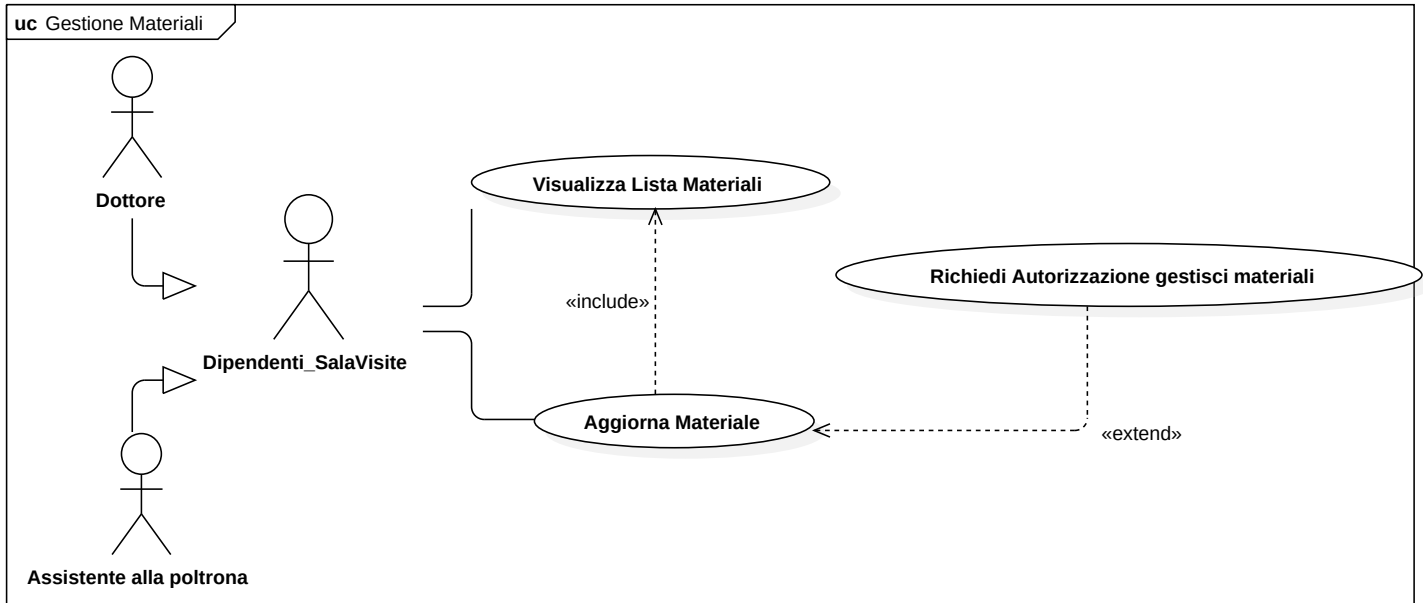


uc verifica pagamento



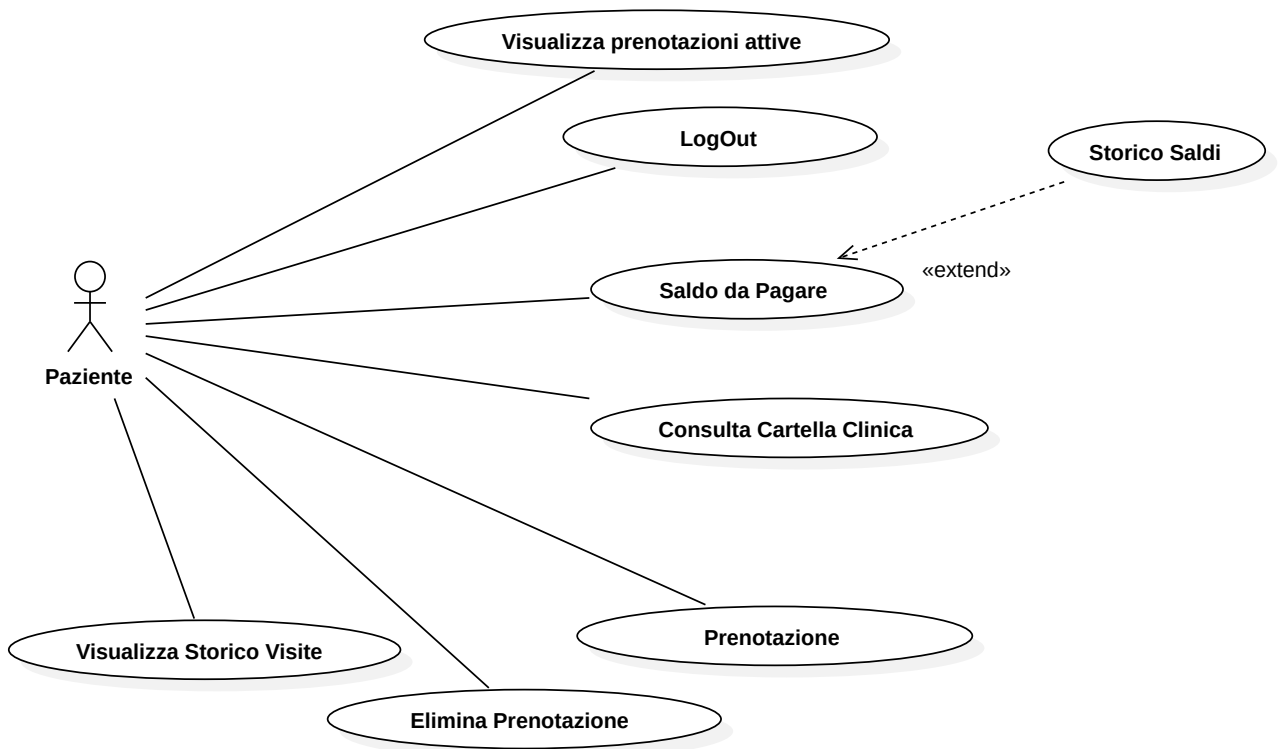
uc gestione fatture

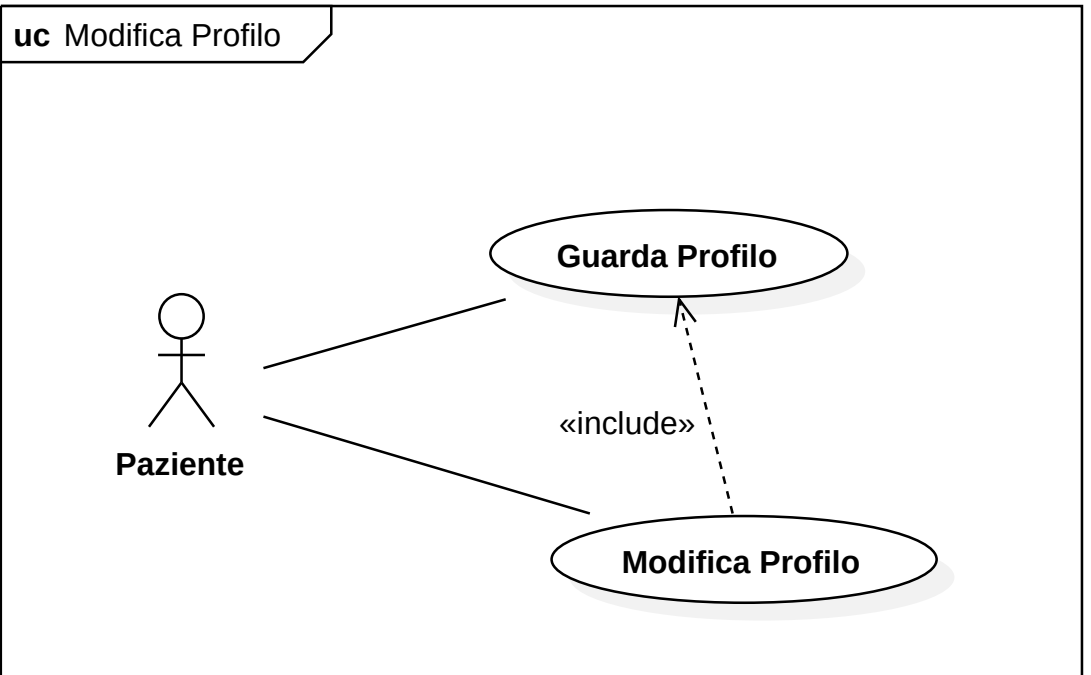




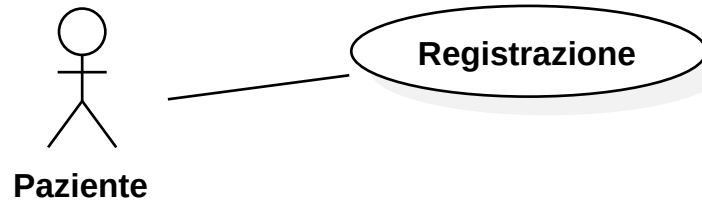
```

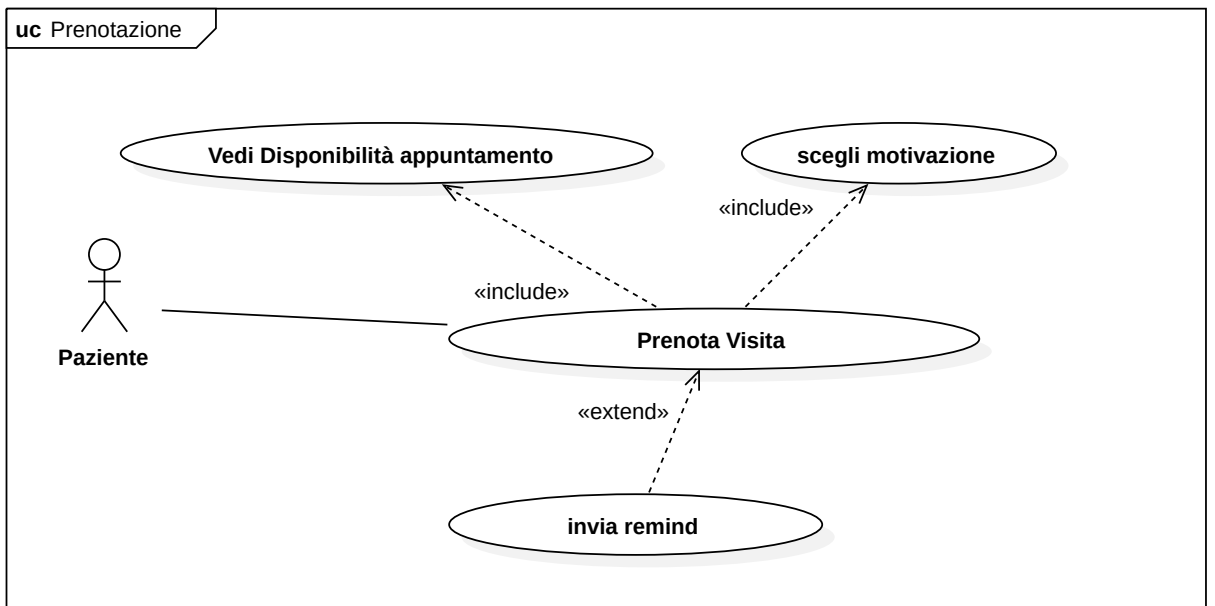
graph LR
    P((Paziente)) --- VPA(Visualizza prenotazioni attive)
    P --- LO(LogOut)
    P --- SDP(Saldo da Pagare)
    P --- CCC(Consulta Cartella Clinica)
    P --- PV(Visualizza Storico Visite)
    P --- EP(Elimina Prenotazione)
    P --- PR(Prenotazione)
    SS(Storico Saldi) -.-> SDP
    SS -- «extend» --> SDP
  
```



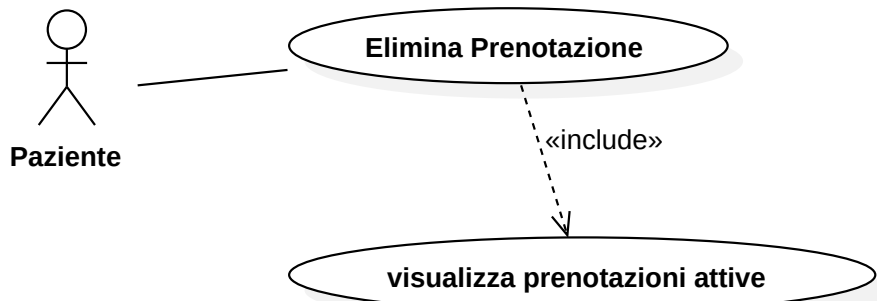


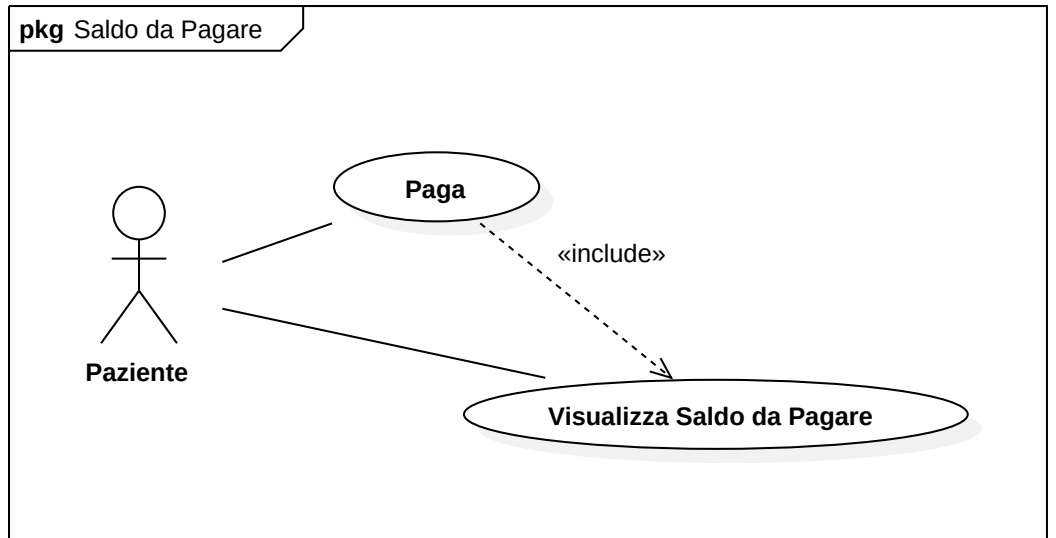
uc Registrazione



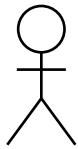


uc Elimina_Prenotazione





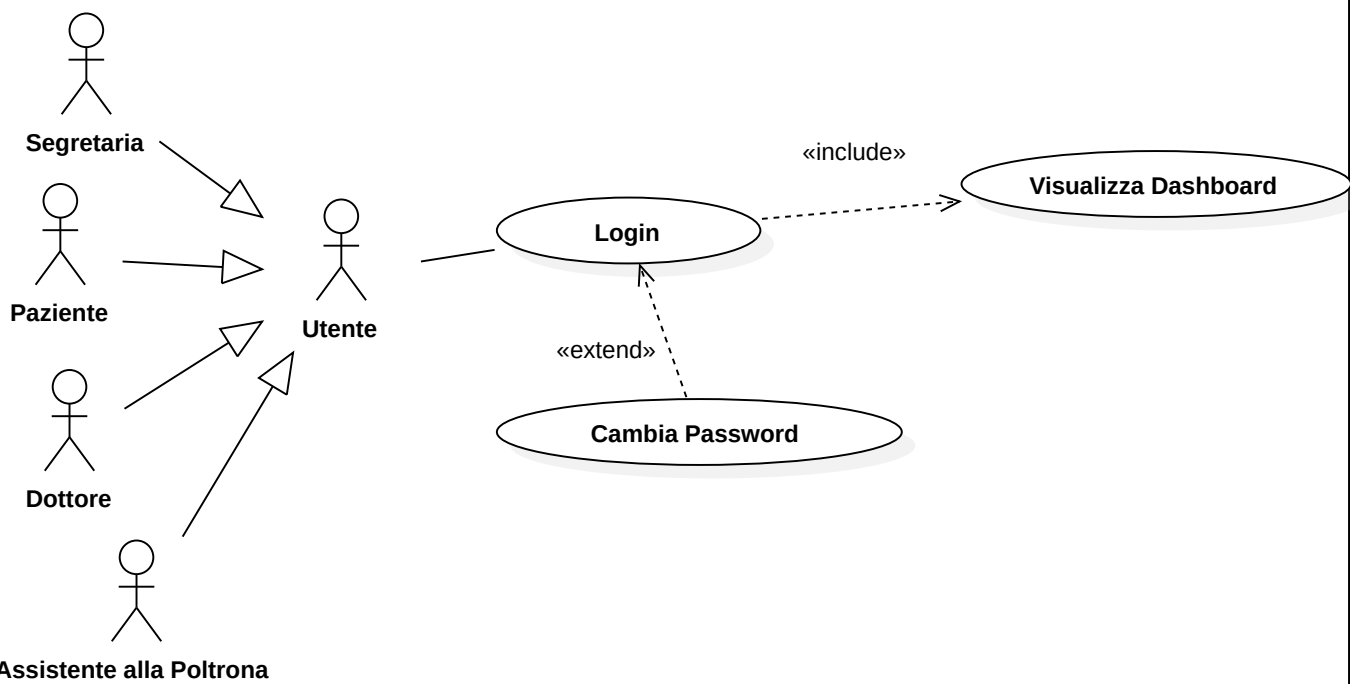
uc Invia Remind

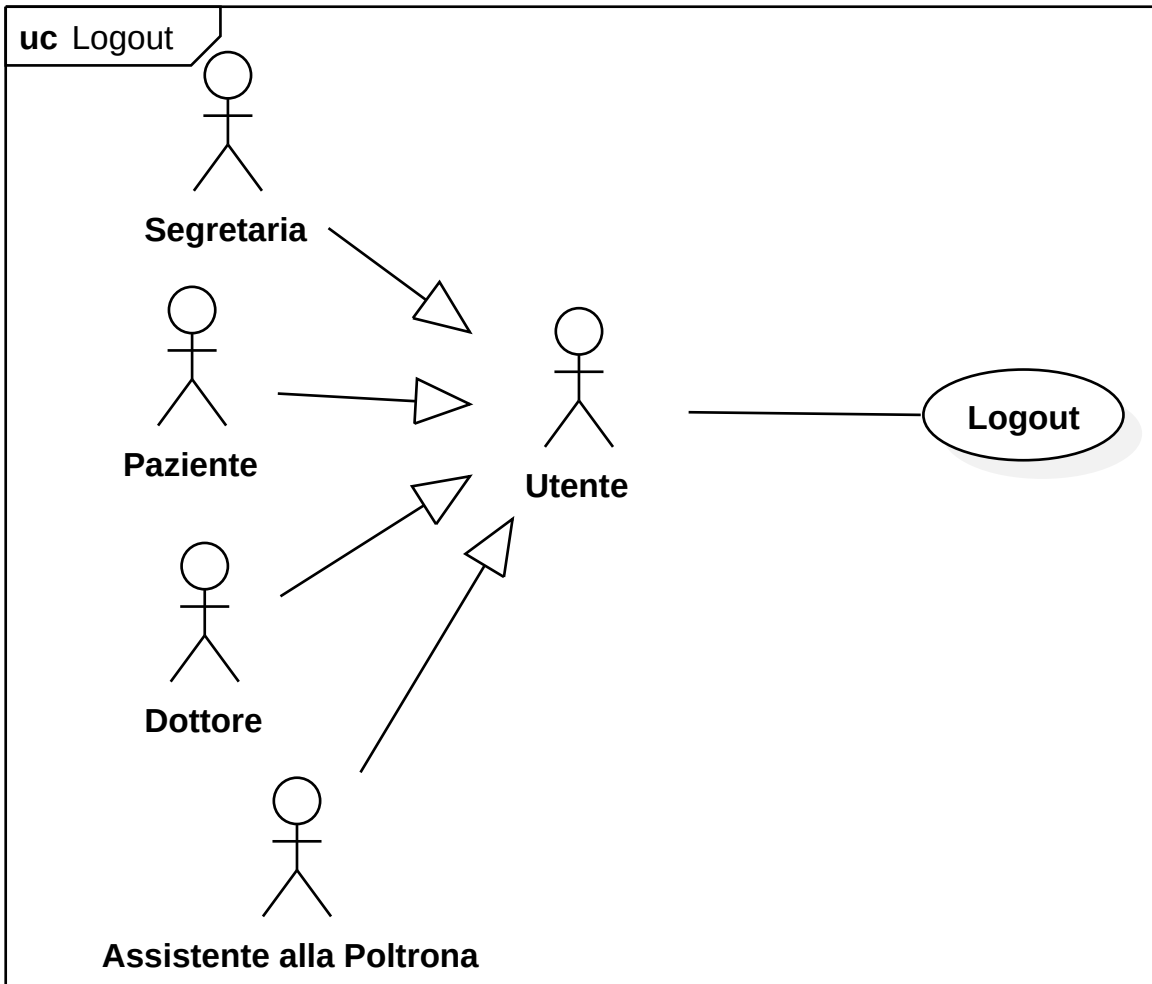


Tempo

Manda Notifica

uc Login





4. Specifiche Dei Casi D'Uso

1 Dashboard Dottore-Paziente

Use Case: Visualizza lista pazienti
ID = 1
Brief Description : Il sistema mostra l'elenco dei pazienti registrati
Primary Actors : Dottore Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore primario è autenticato
Main Flow : <ol style="list-style-type: none">1. Il caso d'uso inizia quando uno degli attori primari seleziona "Visualizza lista pazienti"2. Il sistema mostra la lista del Pazienti3. Il sistema da la possibilità di filtrare l'elenco per Nome completo
Post-Conditions : None
Alternative Flow : <ol style="list-style-type: none">2 Se non ci sono pazienti registrati non mostra nulla

Use Case: Visualizza profilo paziente
ID = 2
Brief Description : Il sistema mostra i dati anagrafici del paziente e la sua cartella clinica
Primary Actors : Dottore Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore è autenticato l'attore si trova sul profilo di un paziente
Main Flow : <ol style="list-style-type: none"> 1. l'attore seleziona "visualizza profilo paziente" su una entry del elenco 2. Il Sistema mostra i dati anagrafici del paziente 3. il sistema mostra la cartella clinica più recente del paziente 4. IF l'attore seleziona "visualizza storico visite" <ol style="list-style-type: none"> 4.1 il sistema mostra lo storico visite nel paziente 4.2 il sistema permette di mostrare una cartella clinica passata 5. IF l'attore seleziona "visualizza prenotazioni attive" <ol style="list-style-type: none"> 5.1 il sistema mostra le prenotazioni attive del paziente
Post-Conditions : None
Alternative Flow : <ol style="list-style-type: none"> 3 Se non c'è una cartella clinica non mostra nulla 4 Se non sono registrate visite passate visualizza un messaggio di avviso 5 Se non sono registrate prenotazioni attive visualizza un messaggio di avviso

Use Case: Visualizza Lista Appuntamenti
ID = 3
Brief Description : L'attore visualizza la lista degli appuntamenti attivi.
Primary Actors : Dottore Segretaria Assistente alla Poltrona
Secondary Actors : None
Pre-Conditions : L'attore si è autenticato correttamente.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore seleziona "Visualizza appuntamenti" 2. L'attore ha la possibilità di filtrare un appuntamento in base al paziente, data o gravità; 3. Il sistema mostra i dettagli di ogni appuntamento visualizzato.
Post-Conditions : None
Alternative Flow : <ol style="list-style-type: none"> 2 Se non sono registrati appuntamenti visualizza un messaggio di avviso

Use Case: Visualizza Appuntamenti odierni
ID = 34
Brief Description : Visualizza gli appuntamenti del giorno ordinati per orario
Primary Actors : Dottore Segretaria Assistente alla Poltrona
Secondary Actors : None
Pre-Conditions : L'attore è autenticato C'è almeno un appuntamento in data odierna
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il dottore seleziona "Visualizza Appuntamenti odierni" 2. Il sistema filtra dall'elenco degli appuntamenti quelli in data odierna 3. L'attore visualizza correttamente gli appuntamenti del giorno corrente per fascia oraria
Post-Conditions : None
Alternative Flow : None

Use Case: Elimina paziente
ID = 4
Brief Description : Una volta nel profilo del paziente permette di eliminare il suo profilo
Primary Actors : Dottore
Secondary Actors : None
Pre-Conditions : L'attore è autenticato l'attore sta visualizzando la lista dei pazienti c'è almeno un paziente nel elenco
Main Flow : <ol style="list-style-type: none"> 1. il caso d'uso inizia quando l'attore dopo aver selezionato un paziente sceglie l'opzione "elimina profilo" 2. il sistema chiede conferma all'attore per la cancellazione del profilo 3. IF cartella clinica non è vuota <ol style="list-style-type: none"> 3.1 invia un email al paziente "allega" la sua cartella clinica
Post-Conditions : L'account del paziente verrà eliminato
Alternative Flow : <ol style="list-style-type: none"> 2 se l'attore non conferma la scelta, annulla l'operazione e il caso d'uso termina

1.1 Caso D'uso in dettaglio: Gestisci Cartella clinica paziente

Use Case: Aggiungi cartella clinica
ID = 5
Brief Description : Da la possibilità al attore di aggiungere una nuova cartella clinica al profilo del paziente
Primary Actors : Dottore Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore deve essere autenticato. l'attore si trova sul profilo di un paziente
Main Flow : <ol style="list-style-type: none">1. il caso d'uso inizia quando l'attore seleziona "aggiungi cartella clinica"2. WHILE l'attore si trova sulla pagina<ol style="list-style-type: none">2.1 l'attore ha la possibilità di inserire la data di caricamento2.2 l'attore ha la possibilità di scrivere l'anamnesi del paziente2.3 l'attore ha la possibilità di scrivere la diagnosi del paziente2.4 trattamento sottoposto2.5 terapia3. all'uscita dalla pagina i nuovi dati saranno salvati
Post-Conditions : Viene creata una nuova cartella clinica se almeno un campo di questa non è vuoto e viene aggiunta al profilo del paziente
Alternative Flow : <ol style="list-style-type: none">3 se tutti i campi da 2.2 a 2.6 sono vuoti non verrà creata/aggiunta nessuna cartella clinica nel profilo del paziente

Use Case: Modifica cartella clinica
ID = 6
Brief Description : Da la possibilità di modificare una cartella clinica del paziente passata
Primary Actors : Dottore Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore è autenticato l'attore si trova sul profilo di un paziente Il paziente ha almeno una cartella clinica nel suo profilo
Main Flow : <ol style="list-style-type: none"> 1. il caso d'uso inizia quando l'attore seleziona "modifica cartella clinica passata" 2. l'attore ha la possibilità di inserire le modifiche che si vogliono applicare 3. il sistema verifica se l'attore ha l'autorizzazione necessaria per modificare la cartella clinica 4. le modifiche vengono applicate al database
Post-Conditions : la cartella clinica del paziente viene modificata
Alternative Flow : <ol style="list-style-type: none"> 4 Modifica Senza Auth C.C.

Alternative Flow : Modifica cartella clinica: Modifica Senza Auth C.C.
ID = 6.1
Brief Description : il sistema, quando l'attore non ha l'auth necessaria ad applicare una modifica, non potendo modificare il database invia invece una richiesta di "approvazione modifica" al dottore
Primary Actors : Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore non ha inviato provato a modificare una cartella clinica
Alternative Flow : 1. L'alternative flow inizia dopo il punto 3 del main flow 2. extend(Richiedi approvazione modifica)
Post-Conditions : La modifica proposta è messa in attesa

Use Case: Elimina cartella clinica
ID = 7
Brief Description : Da a possibilità di eliminare una o più cartelle cliniche di un dato paziente
Primary Actors : Dottore Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'utente è autenticato l'utente si trova sul profilo di un paziente c'è almeno una carte clinica caricata nel profilo
Main Flow : <ol style="list-style-type: none"> 1. il caso d'uso inizia quando l'attore seleziona "elimina cartelle cliniche" 2. il sistema mostra tutti le cartelle cliniche del paziente 3. l'attore può selezionare una o più di queste 4. il sistema controlla se l'attore ha l'autorizzazione necessaria per eliminare la cartella clinica 5. il sistema elimina le cartelle cliniche selezionate
Post-Conditions : Viene eliminata la cartella clinica
Alternative Flow : <ol style="list-style-type: none"> 4 Eliminazione Senza Auth

Alternative Flow : Elimina cartella clinica: Eliminazione Senza Auth
ID = 7.1
Brief Description : il sistema invece di far eliminare direttamente il database al attore invia una richiesta di "approvazione modifica" al dottore
Primary Actors : Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore ha provato ad eliminare una o più cartelle cliniche
Alternative Flow : <ol style="list-style-type: none"> 1. L'alternative flow inizia dopo il punto 4 del main flow 2. extend(Richiedi approvazione modifica)
Post-Conditions : La modifica proposta è messa in attesa

Extension Use Case: Richiedi approvazione modifica
ID = 8
Brief Description : Quando l'attore vuole modificare il database e non se ne ha l'autorizzazione questo caso d'uso fornisce la funzionalità per "mettere in attesa la modifica" e notificare il dottore
Primary Actors : Assistente alla poltrona
Secondary Actors : None
Main Flow : <ol style="list-style-type: none"> 1. Il sistema raccoglie informazioni in merito alle modifiche effettuate 2. Il sistema notifica il dottore e "mette in attesa" la modifica
Post-Conditions : La modifica è messa in attesa
Alternative Flow : None

2 Autorizzazione modifiche

Use Case: Esamina modifiche proposte
ID = 9
Brief Description : l'attore visualizza le modifiche proposte e può decidere se accettarle oppure no
Primary Actors : Dottore
Secondary Actors : Assistente alla poltrona
Pre-Conditions : l'attore primario è autenticato l'attore secondario ha proposto una modifica ad un database
Main Flow : <ol style="list-style-type: none">1. il caso d'uso inizia quando il dottore seleziona "Autorizza"2. l'attore visualizza le modifiche proposte in attesa3. l'attore per ogni una di queste può decidere se accettarle o rifiutarle4. l'attore secondario viene avvisato della decisione del attore primario
Post-Conditions : Sulla base della decisione dell'attore primario la modifica in attesa possono essere ufficializzate o scartate
Alternative Flow : <ol style="list-style-type: none">2 se non ci sono modifiche proposte il sistema mostra un avviso e il caso d'uso termina

3 Materiale

Use Case: Visualizza elenco materiale
ID = 10
Brief Description : permette all'attore di visualizzare l'elenco del materiale
Primary Actors : Dottore Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore è autenticato
Main Flow : <ol style="list-style-type: none">1. il caso d'uso inizia quando l'attore seleziona "mostra materiale"2. Il sistema mostra la lista evidenziando gli elementi con la flag "da compare" spuntata3. sistema permette di filtrare l'elenco per nome del materiale
Post-Conditions : None
Alternative Flow : None

Use Case: Gestione materiale presente
ID = 11
Brief Description : Fornisce le funzionalità di aggiungere elementi e modificare la quantità di un dato materiale nell'elenco materiali presenti
Primary Actors : Dottore Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore è autenticato il sistema sta mostrando l'elenco del materiale presente
Main Flow : <ol style="list-style-type: none"> 1. il caso d'uso inizia quando si seleziona "Modifica elenco" 2. WHILE l'utente si trova nella sezione <ol style="list-style-type: none"> 2.1 può aggiungere un materiale inserendo nome e quantità presente 2.2 Modificare la quantità di un materiale <ol style="list-style-type: none"> 2.2.1 se la quantità è impostata a zero il sistema all'uscita lo elimina 2.3 il sistema verifica se l'attore ha l'auth a effettuate le modifiche 2.4 le modifiche vengono applicate al database
Post-Conditions : Il database viene modificato
Alternative Flow : <ol style="list-style-type: none"> 2.4 Modifica senza auth

Alternative Flow : Gestione elenco materiale : Modifica Senza Auth materiale
ID = 11.1
Brief Description : il sistema, quando l'attore non ha l'auth necessaria ad applicare una modifica, non potendo modificare il database invia invece una richiesta di "approvazione modifica" al dottore
Primary Actors : Assistente alla poltrona
Secondary Actors : None
Pre-Conditions : L'attore ha provato a modificare l'elenco del materiale
Alternative Flow : 1. L'alternative flow inizia dopo il punto 2.4 del main flow 2. extend(Richiedi approvazione modifica)
Post-Conditions : La modifica proposta è messa in attesa

4 Gestione dipendenti

Use Case: Genera link registrazione dipendente
ID = 12
Brief Description : Quando viene assunto un nuovo dipendente è possibile aggiungerlo alla piattaforma generando un link di registrazione e questo sarà inviato via email dal sistema
Primary Actors : Dottore
Secondary Actors : None
Pre-Conditions : L'attore è autenticato
Main Flow : <ol style="list-style-type: none">1. il caso d'uso inizia quando l'attore seleziona "Nuovo Dipendente"2. l'attore può selezionare il ruolo tra assistente alla poltrona e segretario3. l'attore deve scrivere l'email al quale inviare il link di registrazione4. l'attore conferma l'invio della mail5. il sistema genera il link di registrazione valido per 10min e invia la mail al email inserita
Post-Conditions : il sistema salva l'identificativo del link di invito
Alternative Flow : <ol style="list-style-type: none">4 l'attore annulla il processo ed il caso d'uso termina

Use Case: Elimina dipendente
ID = 13
Brief Description : Permette di eliminare un dipendente dalla piattaforma
Primary Actors : Dottore
Secondary Actors : Dipendente
Pre-Conditions : L'attore primario è autenticato c'è almeno un dipendente registrato
Main Flow : <ol style="list-style-type: none"> 1. il caso d'uso inizia quando l'attore primario seleziona "Elimina Dipendente" 2. il sistema mostra l'elenco dei dipendenti mostrando nome,cognome,ruolo 3. il sistema dà la possibilità di filtrare l'elenco 4. l'attore seleziona un dipendente e clicca su elimina 5. il sistema chiede conferma all'attore principale 6. il sistema elimina il profilo dell'attore secondario e lo avvisa con una email
Post-Conditions : l'account viene eliminato dalla piattaforma
Alternative Flow : <ol style="list-style-type: none"> 5 l'attore primario annulla l'eliminazione ed il caso d'uso termina

Use Case: Registrazione dipendente
ID = 14
Brief Description : Gli attori possono registrarsi al portale.
Primary Actors : Assistente alla poltrona Segretaria
Secondary Actors : None
Pre-Conditions : Gli attori giungono nella fase di registrazione in seguito all'apertura del link di invito datogli dal dottore.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore seleziona 'Registrazione' 2. il Sistema riconduce l'attore in un form in cui deve inserire: <ul style="list-style-type: none"> - una propria e-mail valida - una Password che rispetti le condizioni di sicurezza (lunga min:8 max:16 deve contenere dei caratteri speciali e almeno una maiuscola e una minuscola) - Nome - Cognome - Indirizzo - Codice Fiscale - Numero cellulare 3. L'attore, compilati tutti i campi obbligatori, potrà confermare che le informazioni inmesse siano corrette 4. L'attore verrà ricondotto ad una pagina ove inserire il codice di conferma che il sistema invierà alla mail inserita precedentemente 5. La registrazione è andata a buon fine
Post-Conditions : Il Sistema aggiorna il database dei dipendenti, aggiungendo un nuovo dipendente
Alternative Flow : CompilazioneNonCompletata CodiceNonCorretto

Casi D'Uso Paziente

Use Case : registrazione paziente
ID = 15
Brief Description : Il Paziente può registrarsi al portale
Primary Actors : Paziente
Secondary Actors : none
Pre-Conditions : none
Main Flow : <ol style="list-style-type: none">1. Il caso d'uso inizia quando il paziente seleziona 'Registrazione'2. il Sistema riconduce il Paziente in un form in cui deve inserire<ul style="list-style-type: none">- una propria e-mail valida- una Password che rispetti le condizioni di sicurezza (lunga min:8 max:16 deve contenere dei caratteri speciali e almeno una maiuscola e una minuscola)- Nome- Cognome- indirizzo- Codice Fiscale- Numero cellulare3. Il Paziente, compilati tutti i campi obbligatori, potrà confermare che le informazioni inmesse siano corrette4. Il Paziente verrà ricondotto ad una pagina ove immette il codice di conferma che il sistema invierà alla mail inserita precedentemente5. La registrazione è andata a buon fine
Post-Conditions : Il Sistema aggiorna il database dei pazienti, aggiungendo un nuovo paziente
Alternative Flow : CompilazioneNonCompletata CodiceNonCorretto

Alternative Flow : registrazione paziente : compilazione non corretta
ID = 15.1
Brief Description : Il sistema avvisa l'attore che non ha compilato tutti i campi e che non può proseguire se non li completa tutti
Primary Actors : Paziente
Secondary Actors : none
Pre-Conditions : l'attore non ha compilato tutti i campi del form di registrazione, ma ha confermato la corretta inmissione dei dati
Alternative Flow : <ol style="list-style-type: none"> 1. L'alternative flow inizia dopo il punto 3 del main flow 2. Il sistema manda un avviso all'attore, informandolo che non ha compilato correttamente il form
Post-Conditions : none

Alternative Flow : registrazione paziente : CodiceNonCorretto
ID = 15.2
Brief Description : Il sistema avvisa l'attore che il codice immesso non coincide con quello mandato alla mail specificata precedentemenete
Primary Actors : Paziente
Secondary Actors : none
Pre-Conditions : <ul style="list-style-type: none"> - l'attore ha compilato tutti i campi del form di registrazione, confermando con successo - l'attore ha inserito il codice errato
Alternative Flow : <ol style="list-style-type: none"> 1. L'alternative flow inizia dopo il punto 4 del main flow 2. Il sistema manda un avviso all'attore, informandolo che il codice inserito non è corretto 3. l'attore potrà richiedere al sistema di rimandare il codice di conferma alla mail che ha specificato precedentemente
Post-Conditions : il sistema rimanda il codice di conferma

Use Case : Prenotazione
ID = 16
Brief Description : Il Paziente può prenotare una visita
Primary Actors : Paziente
Secondary Actors : none
Pre-Conditions : L'utente deve aver già fatto precedentemente la Registrazione ed effettuato con successo il Login
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il paziente seleziona 'Prenota' 2. include (vedi disponibilità) 3. include (scegli motivazione) 4. if il cliente seleziona 'Attiva Notifica di Remind' <ol style="list-style-type: none"> 4.1 il Paziente, 30 minuti prima dell'orario della visita riceverà una notifica, gestita dal caso d'uso 'Invia Notifica Remind'
Post-Conditions : Il Sistema aggiorna le date disponibili e salva la prenotazione
Alternative Flow : none

Use Case: include(scegli motivazione)
ID = 17
Brief Description: Il Paziente può selezionare tra le motivazioni della sua prenotazione tra quelle di default
Primary Actors: Paziente
Secondary Actors: none
Pre-Conditions: l'utente deve: <ul style="list-style-type: none"> - Aver effettuato correttamente la Registrazione - Aver effettuato con successo il Login - Star iniziando la procedura di prenotazione - Aver scelto correttamente la data della prenotazione tra quelle disponibili
Main Flow: <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il paziente si trova al punto 4. del caso d'uso 'prenotazione' 2. Il Paziente può scegliere una tra le motivazioni di prenotazione disponibili : <ul style="list-style-type: none"> - Controllo mensile - Pulizia Denti - Prima visita generale - Controllo di metà anno - Carie
Post-Conditions: il sistema salverà la motivazione nelle 'informazioni prenotazione'
Alternative Flow: none

Use Case : vedi disponibilità
ID = 18
Brief Description : Il Paziente può selezionare tra gli appuntamenti disponibili
Primary Actors : Paziente
Secondary Actors : none
Pre-Conditions : l'utente deve: <ul style="list-style-type: none"> - Aver effettuato correttamente la Registrazione - Aver effettuato con successo il Login - Star iniziando la procedura di prenotazione
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il paziente si trova al punto 3. del caso d'uso 'prenotazione' 2. Il Paziente può scegliere una tra le date disponibili selezionando <ul style="list-style-type: none"> - Mese - Giorno - Ora 3. Il sistema bloccherà momentaneamente la data di prenotazione scelta
Post-Conditions : Il sistema salverà momentaneamente la data scelta in 'informazioni prenotazione'
Alternative Flow : none

Use Case : Invia Remind
ID = 19
Brief Description : Il Paziente può selezionare tra gli appuntamenti disponibili
Primary Actors : Tempo
Secondary Actors : Paziente
Pre-Conditions : il Paziente deve: <ul style="list-style-type: none"> - Aver effettuato correttamente la Registrazione - Aver effettuato con successo il Login - Aver selezionato 'Attiva Notifica Remind' di prenotazione
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore Tempo capisce che mancano 30 minuti dall'orario visita 2. l'attore Tempo manda la notifica di remind
Post-Conditions : none
Alternative Flow : none

Use Case : Elimina Prenotazione
ID = 20
Brief Description : Il Paziente può eliminare la prenotazione per una visita
Primary Actors : Paziente
Secondary Actors : Segretaria
Pre-Conditions : L'utente deve aver già <ul style="list-style-type: none"> - fatto precedentemente la Registrazione - effettuato con successo il Login - avere almeno una prenotazione all'attivo
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il Paziente seleziona 'Elimina Prenotazione' 2. include (Visualizza Prenotazioni Attive) 3. Il Sistema elimina la prenotazione selezionata 4. Il Sistema manderà una notifica alla Segretaria dell'avvenuta cancellazione
Post-Conditions : Il Sistema aggiorna le date disponibili, la lista delle prenotazioni attive e libera la prenotazione
Alternative Flow : none

Use Case : Visualizza Prenotazioni Attive
ID = 21
Brief Description : Il Paziente può visualizzare le sue prenotazioni
Primary Actors : Paziente
Secondary Actors : none
Pre-Conditions : L'utente deve aver già <ul style="list-style-type: none"> - fatto precedentemente la Registrazione - effettuato con successo il Login - avere almeno una prenotazione all'attivo
Main Flow : <ol style="list-style-type: none"> 1. il Paziente visualizzerà una lista di sue prenotazioni ancora all'attivo 2. il Paziente sceglierà la prenotazione, tra quelle attive
Post-Conditions : none
Alternative Flow : none

Use Case : login
ID = 22
Brief Description : L'attore può effettuare l'accesso all'applicativo
Primary Actors : Paziente,Segretaria,Dottore,Assistente alla poltrona
Secondary Actors : none
Pre-Conditions : L'attore deve aver effettuato precedentemente una corretta registrazione
Main Flow : <ol style="list-style-type: none"> 1. Use Case inizia quando l'attore entra nel sito 2. il Sistema riconduce l'attore in un form in cui deve inserire <ul style="list-style-type: none"> - l'attore seleziona il ruolo (Paziente, dipendente) per il quale sta facendo il login - la e-mail, con la quale si è registrato precedentemente - la propria Password, con la quale si è registrato precedentemente 3. L'attore, compilati tutti i campi obbligatori, potrà confermare che le informazioni inmesse siano corrette 4. Il sistema verificherà la correttezza dei dati inseriti e se coincidono con il ruolo precedentemente selezionato 5. Il login è andata a buon fine
Post-Conditions : Il Sistema riconurrà l'attore alla dashboard relativa al proprio ruolo
Alternative Flow : CompilazioneNonCompletata EmailErrata PasswordErrata

Alternative Flow : login : CredenzialiErrata
ID = 22.2
Brief Description : Il sistema avvisa l'attore che le credenziali inserite sono errate
Primary Actors : Paziente,Segretaria,Dottore,Assistente alla poltrona
Secondary Actors : none
Pre-Conditions : <ul style="list-style-type: none"> - l'attore ha compilato tutti i campi del form di login - l'attore ha confermato di aver inserito le credenziali corrette
Alternative Flow : <ol style="list-style-type: none"> 1. L'alternative flow inizia dopo il punto 4 del main flow 2. Il sistema manda un avviso all'attore, informandolo che le credenziali inmesse non sono corrette e riprovare ad inserire di nuovo mail e password 3. if (attore decide di cambiare password andando a selezionare password dimenticata) <ul style="list-style-type: none"> - extend (CambiaPassword) 4. si riparte dal punto 2 del main flow
Post-Conditions : none

Alternative Flow: Login: CompilazioneNonCompletata
ID = 22.1
Brief Description: Il sistema avvisa l'attore che il login non è andato a buon fine
Primary Actors: Paziente, Dottore, Assistente alla poltrona, Segretaria
Secondary Actors: none
Pre-Conditions: <ul style="list-style-type: none"> - l'attore non ha compilato tutti i campi del login - l'attore ha confermato che le informazioni che ha innesso sono corrette
Alternative Flow: <ol style="list-style-type: none"> 1. L'alternative flow inizia dopo il punto 3 del main flow 2. Il sistema manda un avviso all'attore, informandolo che per poter accedere con successo deve compilare tutti i campi 3. si riparte dal punto 2 del main flow
Post-Conditions: none

Use Case: Modifica Profilo
ID = 23
Brief Description: L'attore può modificare i dati del suo Profilo
Primary Actors: Paziente Assistente alla poltrona Dottore Segretaria
Secondary Actors: none
Pre-Conditions: <ul style="list-style-type: none"> - L'attore deve aver effettuato precedentemente una corretta registrazione - l'attore deve aver effettuato con successo il login - l'attore deve star visualizzando il suo Profilo
Main Flow: <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore seleziona 'modifica Profilo' 2. l'attore verrà ricondotto in una pagina dove potrà modificare: <ul style="list-style-type: none"> - Numero cellulare - Indirizzo - e-mail - Password 3. L'attore, compilati i campi che vuole modificare, potrà confermare le informazioni inmesse 4. La modifica è andata a buon fine
Post-Conditions: Il Sistema aggiorna il database con le nuove informazioni
Alternative Flow: none

Use Case : CambiaPassword
ID = 23.1
Brief Description : l'utente può cambiare la sua password
Primary Actors : Paziente Assistente alla poltrona Dottore Segretaria
Secondary Actors : none
Pre-Conditions : - l'attore deve essersi correttamente registrato
Alternative Flow : <ol style="list-style-type: none"> 1. L'alternative flow inizia quando l'attore seleziona 'password dimenticata' 2. Il sistema riconduce l'attore ad un form nel quale può inserire la mail usata precedentemente per registrarsi 3. l'attore conferma la mail dove vuole che gli sia mandato il link di ripristino 4. il sistema genera un link temporaneo di ripristino e manda una mail all'indirizzo scritto (il link temporaneo deve essere valido per 15 minuti) 5. l'attore, selezionando il link, verrà ricondotto in un form dove può inserire la sua nuova password 6. l'attore conferma la password inserita 7. il sistema avverte l'attore che il processo è andato a buon fine
Post-Conditions : il sistema aggiorna i database salvando la nuova password
Alternative Flow : 3. se la mail inserita non è presente nel databse, il sistema avverte l'attore. Il sistema riconduce l'attore al punto 2.

Use Case: Saldo da Pagare
ID = 24
Brief Description : L'attore può visualizzare il saldo da pagare e pagare
Primary Actors : Paziente
Secondary Actors : none
Pre-Conditions : <ul style="list-style-type: none"> - L'attore deve aver effettuato precedentemente una corretta registrazione - l'attore deve aver effettuato con successo il login - l'attore deve star visualizzando la sezione pagamenti - l'attore deve avere una propria cartella clinica
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore seleziona 'Pagamenti' 2. l'attore verrà ricondotto in una pagina dove potrà visualizzare il saldo da pagare 3. l'attore selezionando 'paga' verrà ricondotto ad un form ove inserire le credenziali della sua carta di credito 4. il sistema avvisa l'attore dell'andata a buon fine del pagamento, ed inserisce la fattura in uno storico saldi 5. l'attore può scaricare la fattura del pagamento
Post-Conditions : Il Sistema aggiorna il saldo da pagare e segna il pagamento come effettuato
Alternative Flow : none

Use Case: Aggiungi appuntamento
ID = 25
Brief Description : L'attore può aggiungere un nuovo appuntamento per un determinato paziente tenendo conto delle disponibilità degli stessi.
Primary Actors : Segretaria
Secondary Actors : None
Pre-Conditions : Deve essere presente almeno un paziente e deve esserci almeno una data disponibile.
Main Flow : <ol style="list-style-type: none">1. Il caso d'uso inizia quando l'attore sceglie l'opzione "Aggiungi appuntamento";2. L'attore seleziona il paziente per cui si vuole aggiungere il suddetto appuntamento;3. L'attore inserisce il giorno e l'ora in cui si vuole fissare l'appuntamento;4. Il sistema controlla la disponibilità della suddetta data;5. Il sistema invia una notifica al paziente in cui viene informato del nuovo appuntamento fissato;6. Viene aggiornato l'elenco delle date disponibili.
Post-Conditions : L'appuntamento è stato fissato correttamente.
Alternative Flow : L'alternative flow inizia dopo il punto 4 del Main flow <ol style="list-style-type: none">1. Il giorno e l'ora scelti non sono disponibili;2. Il sistema verifica quali sono le date libere più vicine a quella scelta;3. L'attore sceglie una delle suddette date ; Ritorno al punto 5 del Main flow

Use Case: Sposta appuntamento
ID = 26
Brief Description : L'attore può spostare l' appuntamento per un determinato paziente tenendo conto delle disponibilità degli stessi.
Primary Actors : Segretaria
Secondary Actors : None
Pre-Conditions : Deve essere presente l'appuntamento per il suddetto paziente e deve esserci almeno una data disponibile.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore sceglie l'opzione "Sposta appuntamento"; 2. L'attore seleziona il paziente per cui si vuole spostare il suddetto appuntamento; 3. L'attore sceglie una delle date disponibili; 4. Il sistema invia una notifica al paziente in cui viene informato dello spostamento dell'appuntamento precedentemente fissato; 5. Viene aggiornato l'elenco delle date disponibili.
Post-Conditions : L'appuntamento è stato spostato correttamente.
Alternative Flow : none

Use Case: Elimina appuntamento
ID = 27
Brief Description : L'attore può eliminare l' appuntamento per un determinato paziente.
Primary Actors : Segretaria
Secondary Actors : None
Pre-Conditions : Deve essere presente l'appuntamento per il suddetto paziente.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore sceglie l'opzione "Elimina appuntamento"; 2. L'attore seleziona il paziente per cui si vuole eliminare il suddetto appuntamento; 3. L'attore inserisce la motivazione della cancellazione; 4. Il sistema invia una notifica al paziente in cui viene informato della cancellazione della prenotazione; 5. Viene aggiornato l'elenco delle date disponibili.
Post-Conditions : L'appuntamento è stato eliminato correttamente.
Alternative Flow : none

Use Case: Aggiungi Fattura
ID = 28
Brief Description : L'attore può aggiungere una fattura nel database per un paziente.
Primary Actors : Segretaria
Secondary Actors : None
Pre-Conditions : Deve essere presente almeno un paziente con una visita effettuata.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore sceglie l'opzione "Aggiungi fattura"; 2. L'attore inserisce i dati della fattura, tra cui: <ul style="list-style-type: none"> • Numero fattura; • Data fattura; • Paziente; • Elementi fattura(descrizione, quantità, prezzo unitario) 3. L'attore inserisce l'importo totale della fattura; 4. Convalida i dettagli della fattura; 5. Il sistema salva la fattura;
Post-Conditions : La fattura è stata aggiunta correttamente
Alternative Flow : L'alternative flow inizia dopo il punto 4 del Main flow <ol style="list-style-type: none"> 4.1 Se i dettagli non sono validi, tornare al passaggio 2 per correggere i dati.

Use Case: Modifica Fattura
ID = 29
Brief Description : L'attore può modificare una fattura nel database prima che venga inviata al paziente.
Primary Actors : Segretaria
Secondary Actors : None
Pre-Conditions : Deve essere presente almeno un paziente con una visita effettuata.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore sceglie l'opzione "Modifica fattura"; 2. L'attore seleziona la fattura da modificare filtrando per pazienti e/o visite effettuate; 3. L'attore modifica i dettagli della fattura ; 4. L'attore convalida i dettagli della fattura; 5. Il sistema aggiorna la fattura;
Post-Conditions : La fattura è stata aggiornata correttamente.
Alternative Flow : L'alternative flow inizia al punto 4: <ol style="list-style-type: none"> 4.1 Se i dettagli non sono corretti tornare al passaggio 3 per correggere i dati. 4.2 Se l'attore non vuole più modificare la fattura ritorna alla sezione "Gestione fatture".

Use Case: Elimina fattura
ID = 30
Brief Description : L'attore può eliminare una fattura nel database in locale prima di inviarla al paziente.
Primary Actors : Segretaria
Secondary Actors : None
Pre-Conditions : Deve esserci almeno una fattura per il suddetto paziente.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore sceglie l'opzione "Elimina fattura"; 2. L'attore seleziona la fattura da eliminare filtrando per pazienti e/o visite effettuate;; 3. Il sistema chiede all'attore se vuole confermare l'operazione scelta; 4. Il sistema elimina la fattura desiderata;
Post-Conditions : La fattura è stata eliminata dal database correttamente.
Alternative Flow : L'alternative flow inizia al punto 3: <ol style="list-style-type: none"> 3.1 Se l'attore non vuole più eliminare la fattura ritorna alla sezione "Gestione fatture".

Use Case: Invia fattura
ID = 31
Brief Description : L'attore può inviare una fattura al paziente.
Primary Actors : Segretaria
Secondary Actors : None
Pre-Conditions : Deve esserci almeno una fattura nel database per il suddetto paziente.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore sceglie l'opzione "Invia fattura"; 2. L'attore seleziona la fattura da inviare filtrando per pazienti e/o visite effettuate; 3. L'attore inserisce la mail, presente nel profilo del paziente, alla quale vuole inviare la suddetta fattura; 4. Il sistema chiede all'attore se vuole confermare l'operazione scelta; 5. Il sistema invia la fattura desiderata;
Post-Conditions : La fattura è stata inviata correttamente.
Alternative Flow : L'alternative flow inizia al punto 3: <ol style="list-style-type: none"> 4.1 Se l'attore non vuole più eliminare la fattura ritorna alla sezione "Gestione fatture".

Use Case: Visualizza Saldo da Pagare
ID = 32
Brief Description : L'attore può visualizzare il saldo da pagare .
Primary Actors : Paziente Segretaria
Secondary Actors : None
Pre-Conditions : L'attore si è correttamente autenticato. C'è almeno un paziente , con relativa cartella clinica, che ha già effettuato una visita.
Main Flow : <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'attore,dal profilo paziente di cui si vuole conoscere il saldo,sceglie l'opzione "Visualizza Saldo da Pagare"; 2. L'attore visualizza il saldo da pagare oltre al costo totale del trattamento pattuito precedentemente col dottore e allo storico visite; 3. L'attore può visualizzare le singole visite osservando il costo della suddetta e verificare se è stata saldata o meno.
Post-Conditions : none
Alternative Flow : none

Use Case: Verifica Pagamento
ID = 33
Brief Description : L'attore avvia il processo di pagamento e ne determina l'esito.
Primary Actors : Gateway Payment
Secondary Actors : Paziente
Pre-Conditions : L'attore secondario ha effettuato almeno una visita e ha deciso di pagare online il saldo.
Main Flow : <ol style="list-style-type: none"> 1. L'attore principale avvia il processo di pagamento; 2. Viene inviata all'attore secondario la richiesta di autorizzazione per il pagamento che si sta andando ad effettuare; 3. L'operazione di versamento è andata a buon fine; 4. Il sistema aggiorna il saldo da pagare del paziente.
Post-Conditions : Il pagamento è stato effettuato correttamente. Il saldo da pagare è stato aggiornato senza errori.
Alternative Flow : <ol style="list-style-type: none"> 2.1 Il pagamento non è andato a buon fine poiché l'attore secondario non ha dato l'autorizzazione. Occorre ritornare nuovamente al passo 1 del Main Flow.

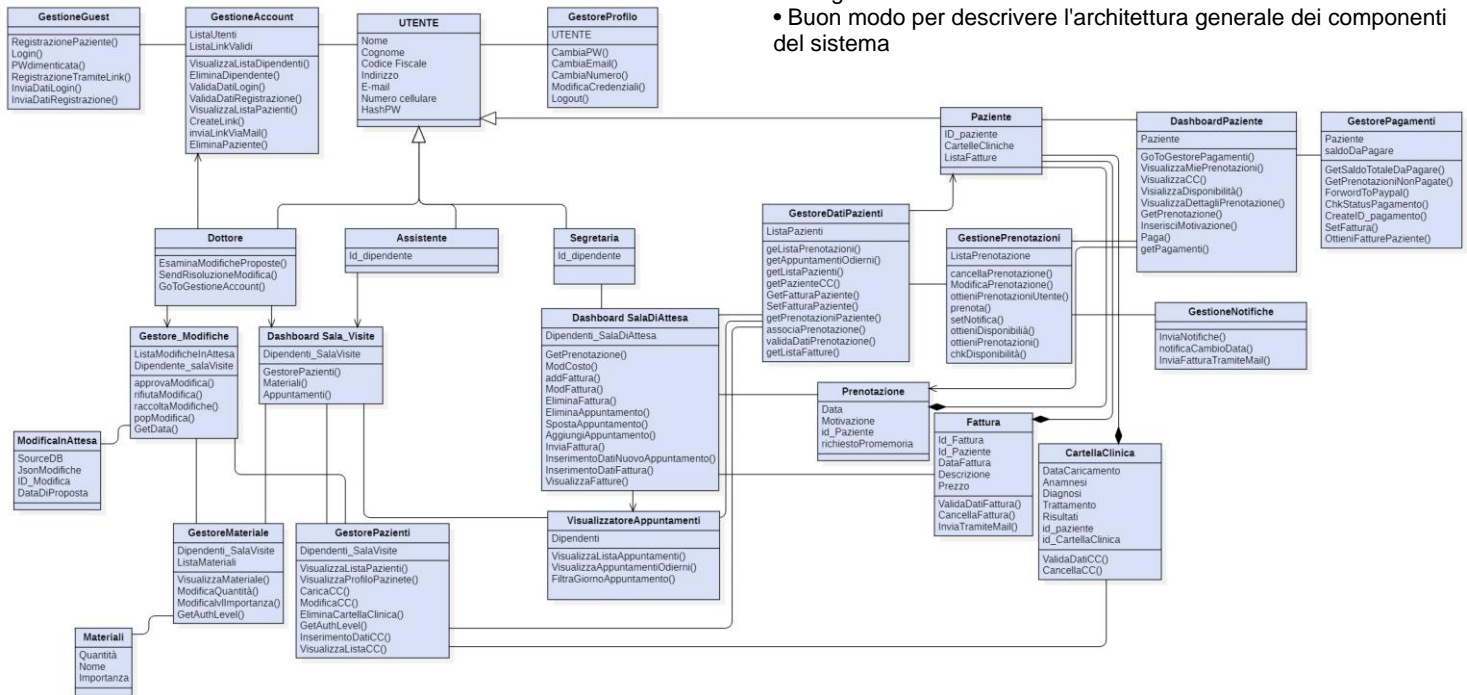
5. Diagramma delle Classi di Analisi ---->

Il diagramma delle classi di analisi fornisce una panoramica di un sistema mostrandone il classi e le relazioni tra di esse.

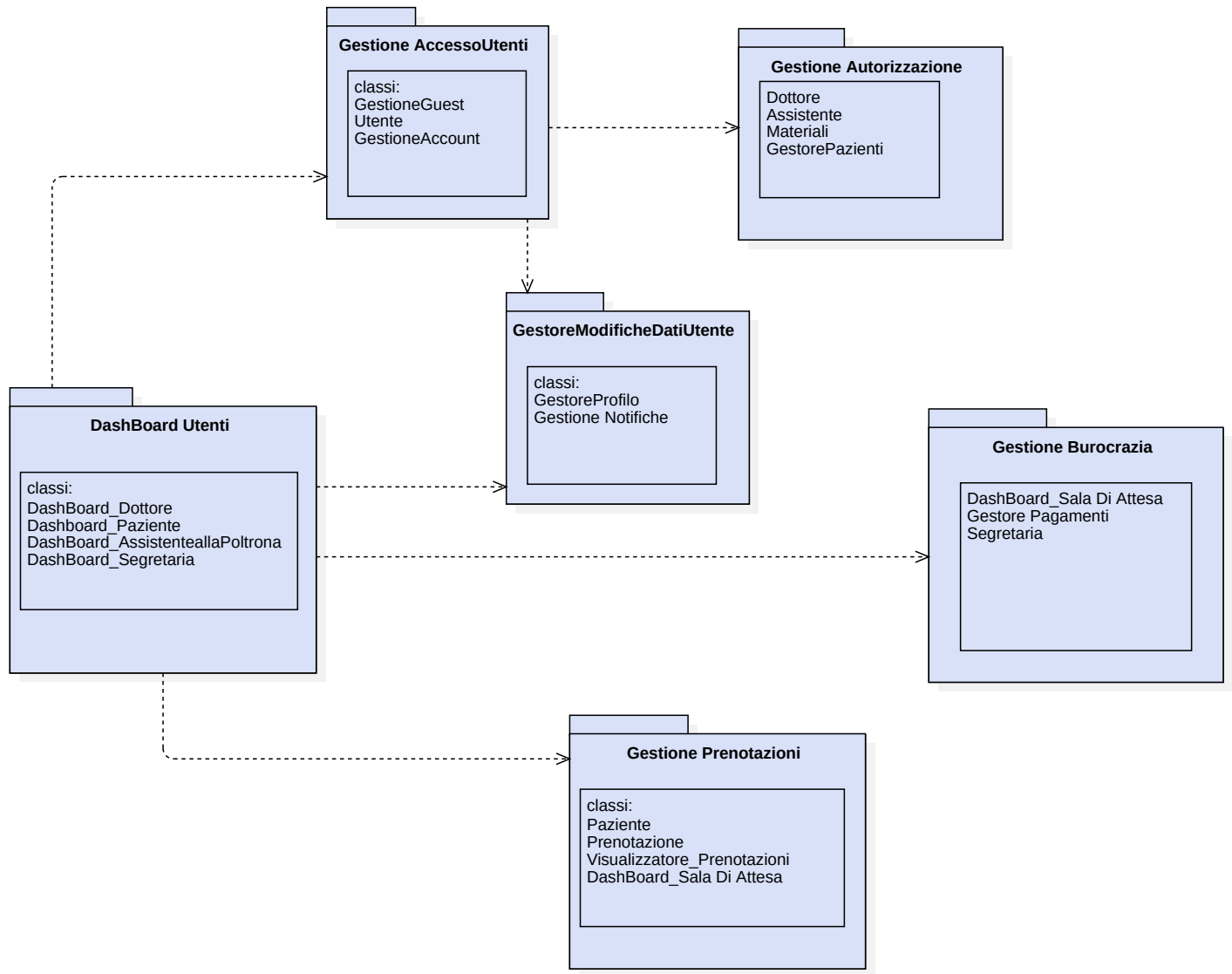
– I diagrammi di classe sono statici

– mostrano ciò che interagisce ma non ciò che accade quando interagiscono

• Buon modo per descrivere l'architettura generale dei componenti del sistema

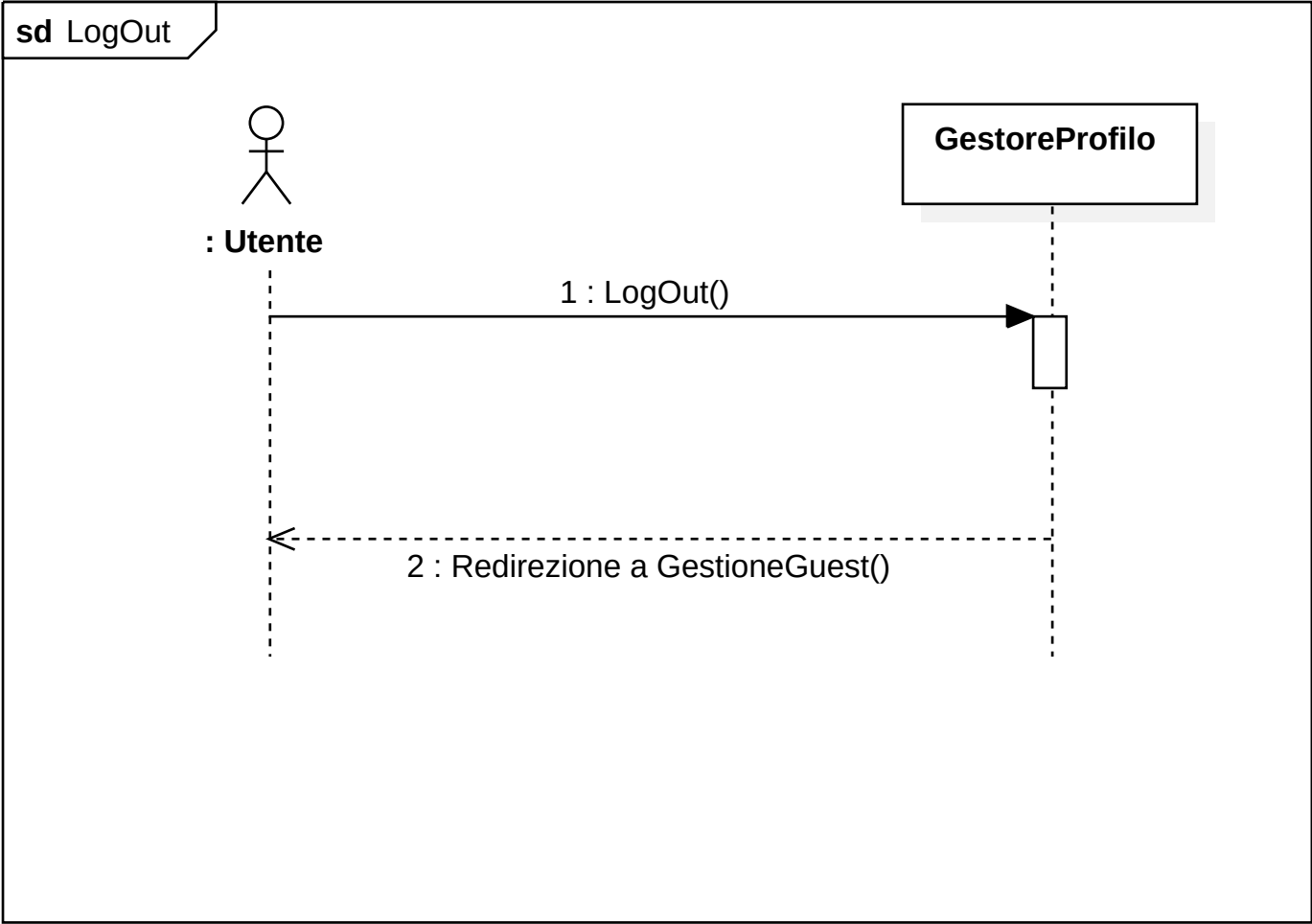


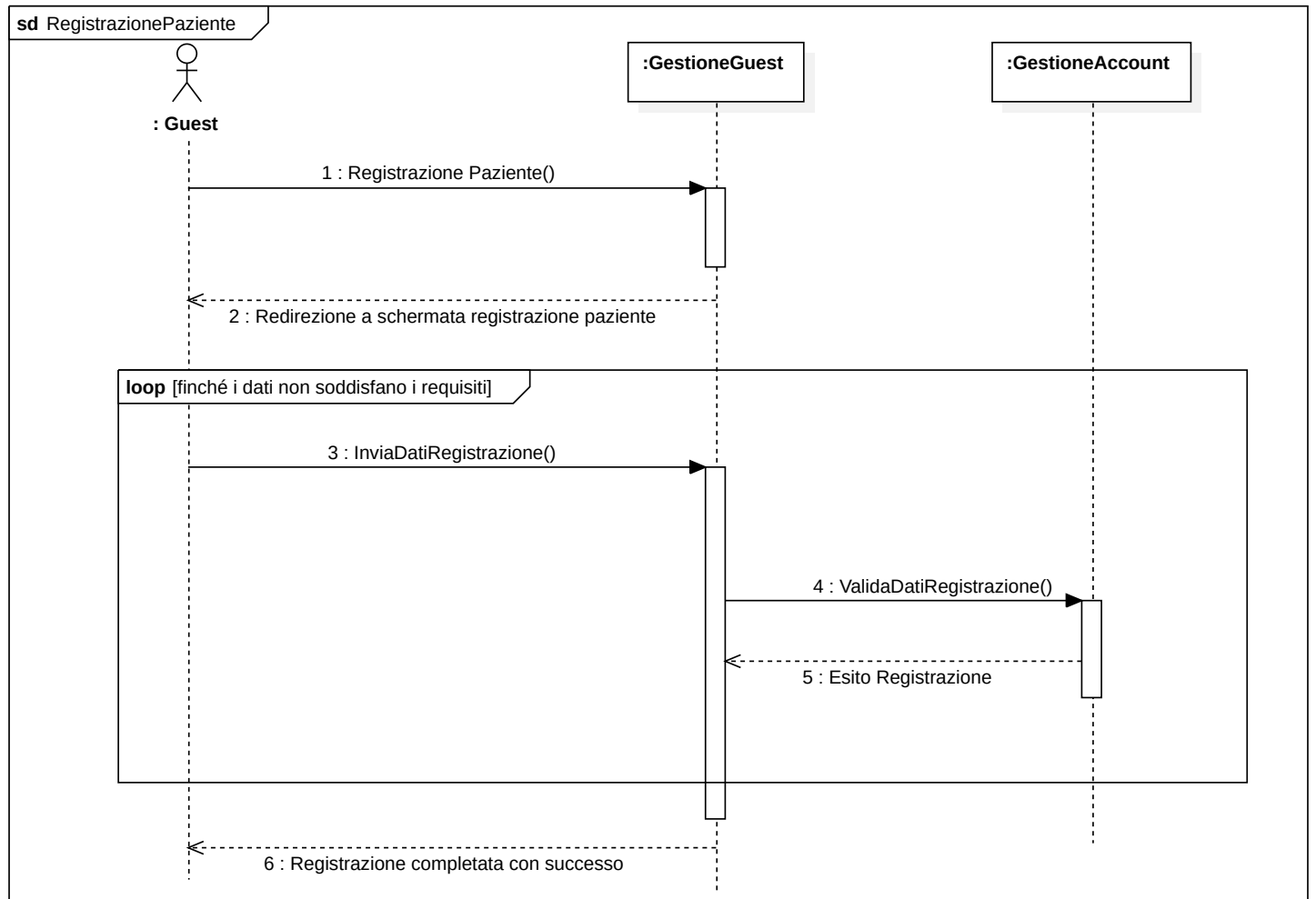
6 Diagramma dei package di analisi



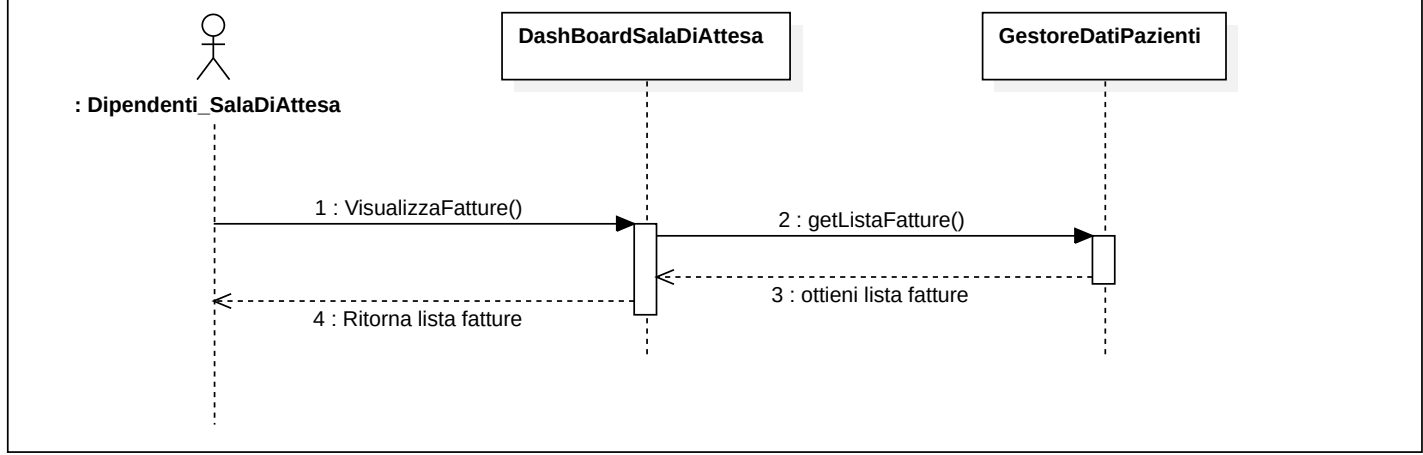
Per organizzare diagrammi di classe complessi, possiamo raggruppare classi in pacchetti. Un pacchetto è una raccolta di elementi UML logicamente correlati.

7. Diagrammi delle sequenze di analisi--> I diagrammi delle sequenze ci aiutano a vedere come il sistema lavora, mostra cosa fanno le classi nel sistema e soprattutto come comunicano tra di loro





sd VisualizzaListaFatture



sd RegistrazioneDipendente


: Guest

:GestioneGuest

:GestioneAccount

1 : RegistrazioneTramiteLink()

2 : Redirezione a schermata registrazione dip.

loop [finché i dati non soddisfano i requisiti]

3 : InviaDatiRegistrazione()

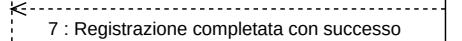
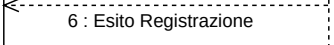
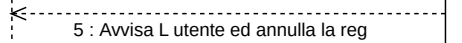
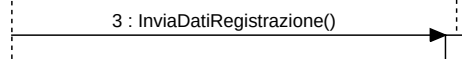
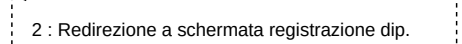
4 : ValidaDatiRegistrazione()

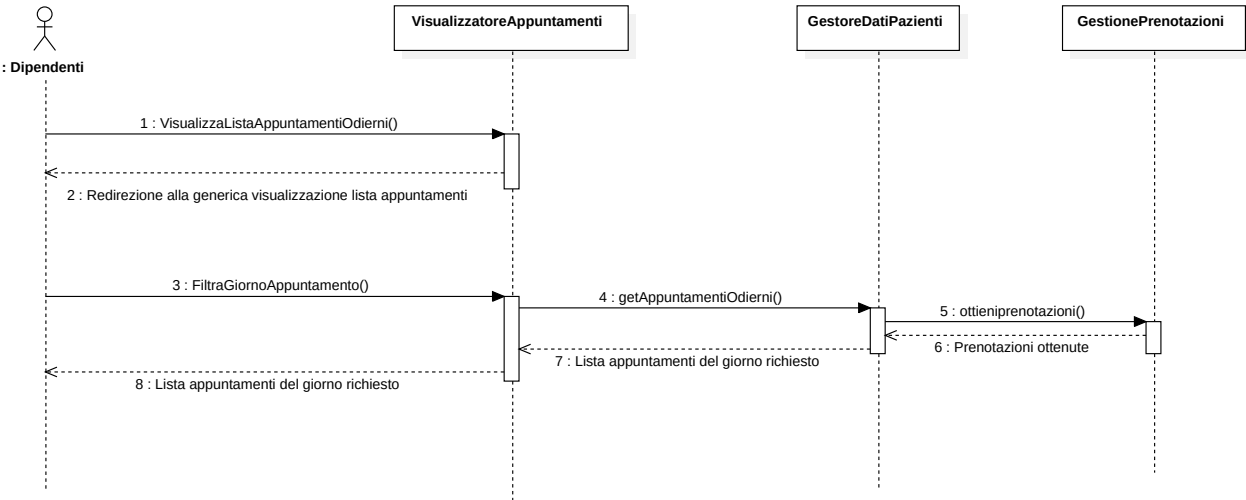
break Link Usato Per La Registrazione non valido

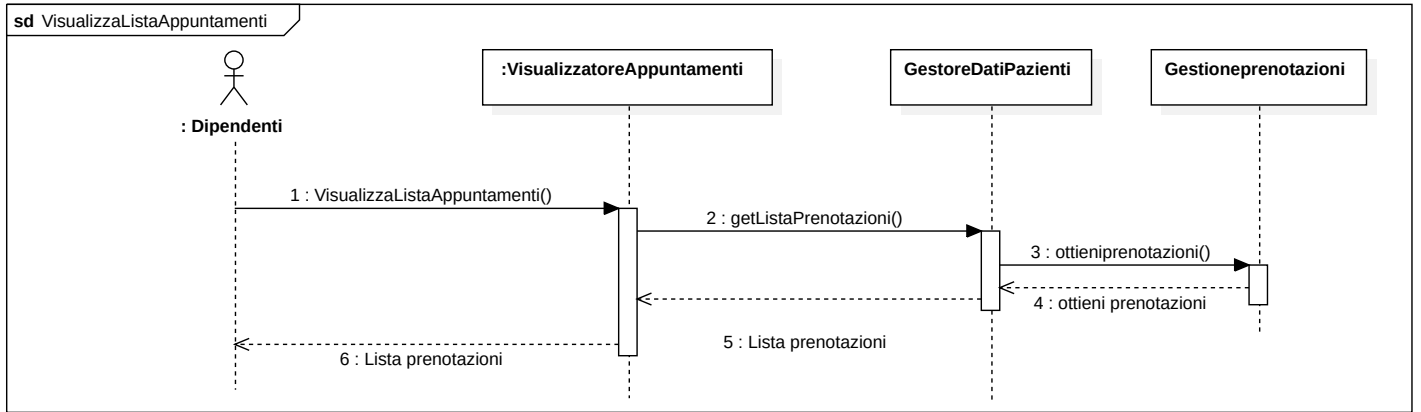
5 : Avvisa L utente ed annulla la reg

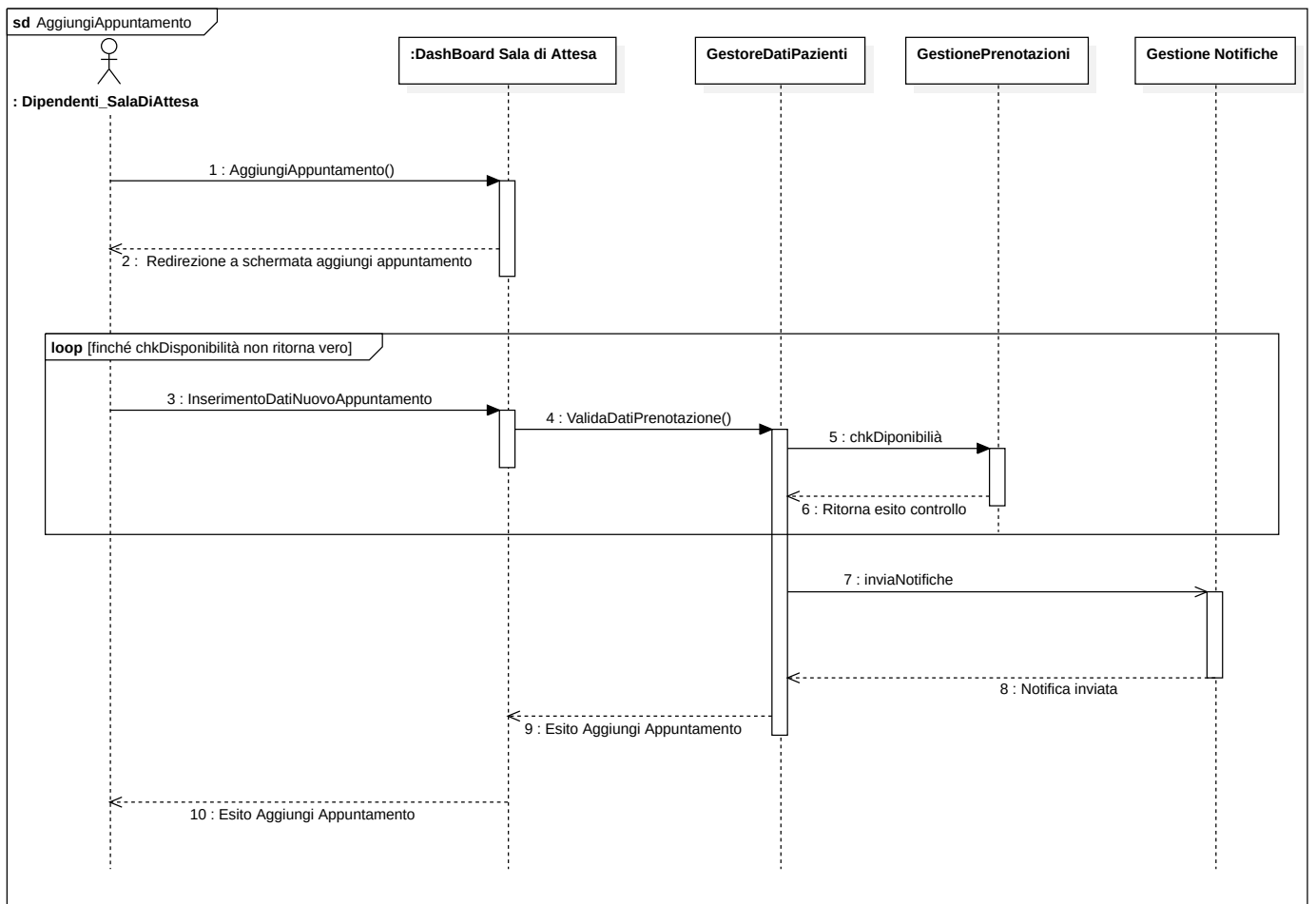
6 : Esito Registrazione

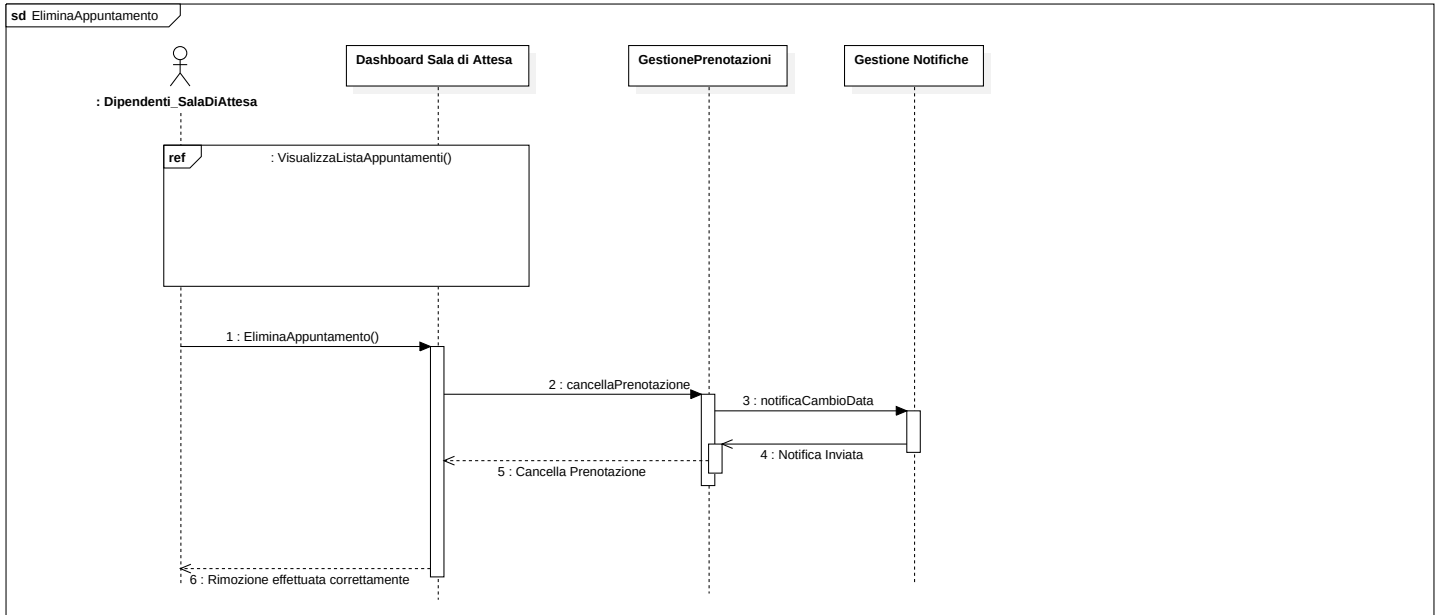
7 : Registrazione completata con successo



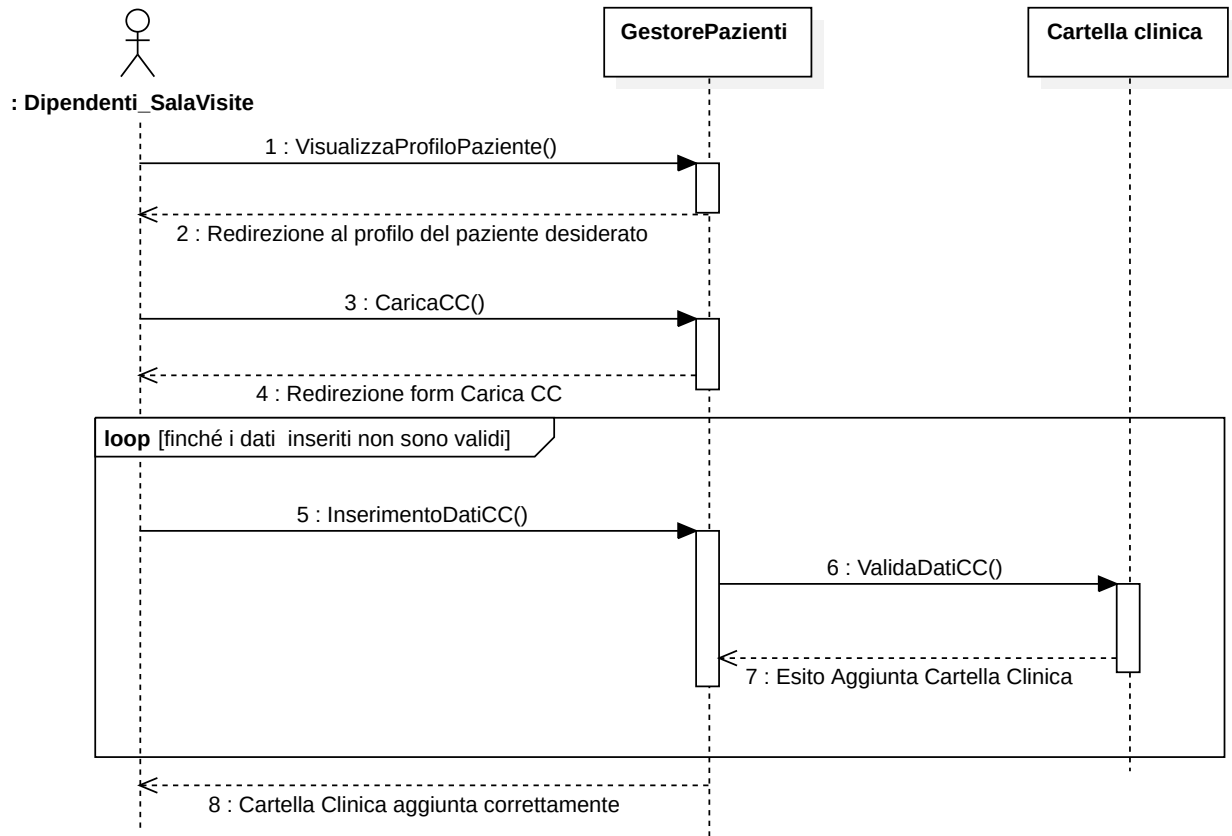








sd Aggiungi Cartella Clinica



sd Elimina Cartella Clinica



: Dipendenti_SalaVisite

GestorePazienti

GestoreDatiPaziente

Cartella clinica

GestoreModifiche

ref

: VisualizzaListaCC()

1 : EliminaCartellaClinica()

2 : getPazienteCC()

3 : Cartella clinica da eliminare

4 : GetAuthLevel

alt

[L'attore è il Dottore=> Ha l'auth per una modifica]

5 : «destroy»

6 : mes. corretta cancellazione CC

7 : Avviso di avvenuta cancellazione

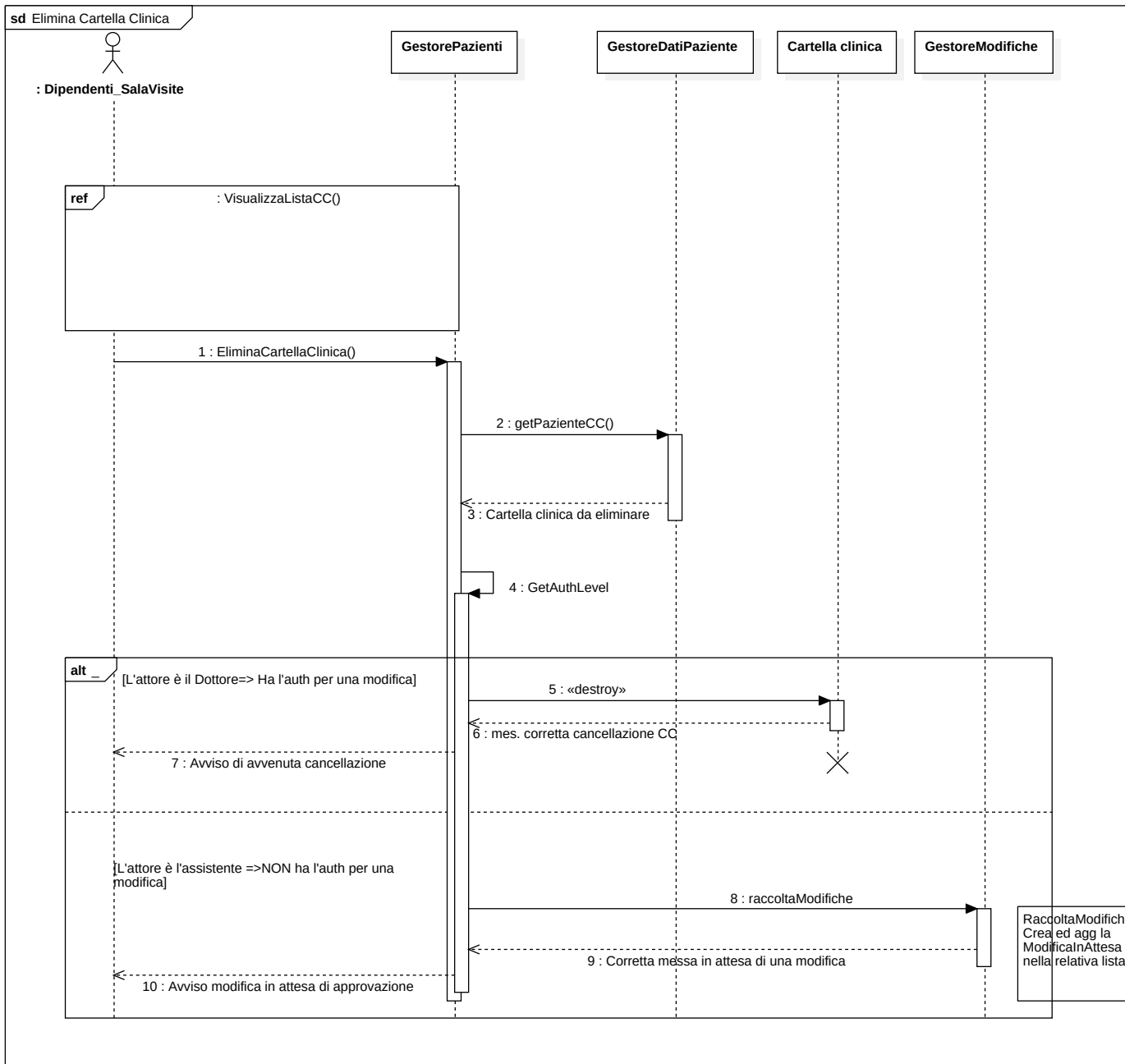
[L'attore è l'assistente => NON ha l'auth per una modifica]

8 : raccoltaModifiche

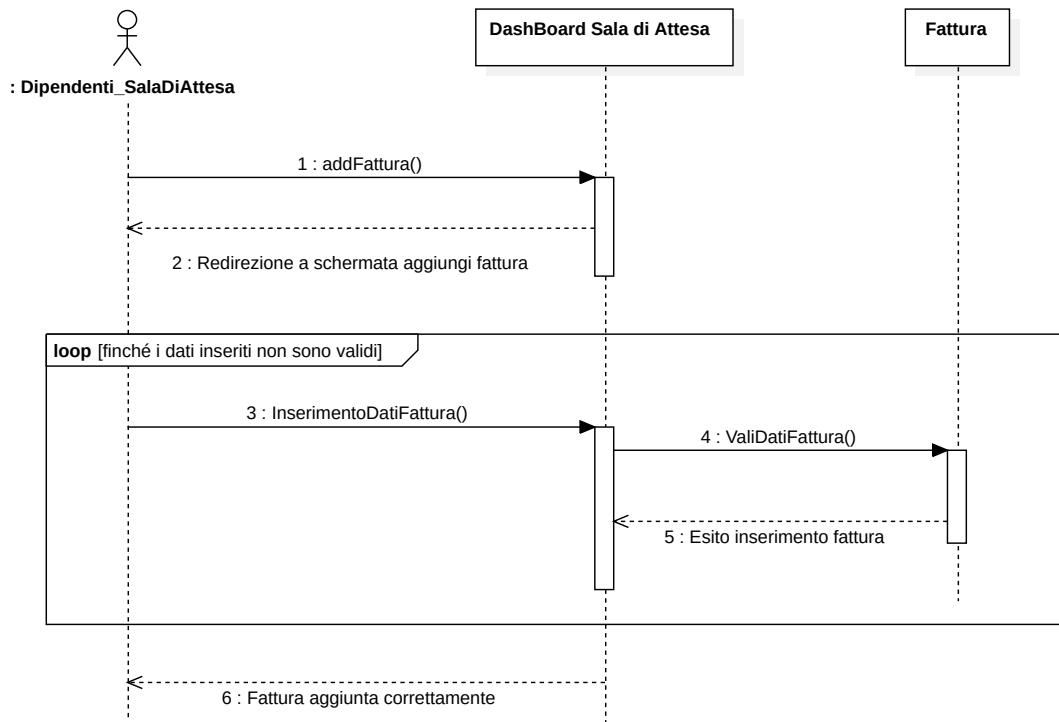
9 : Corretta messa in attesa di una modifica

10 : Avviso modifica in attesa di approvazione

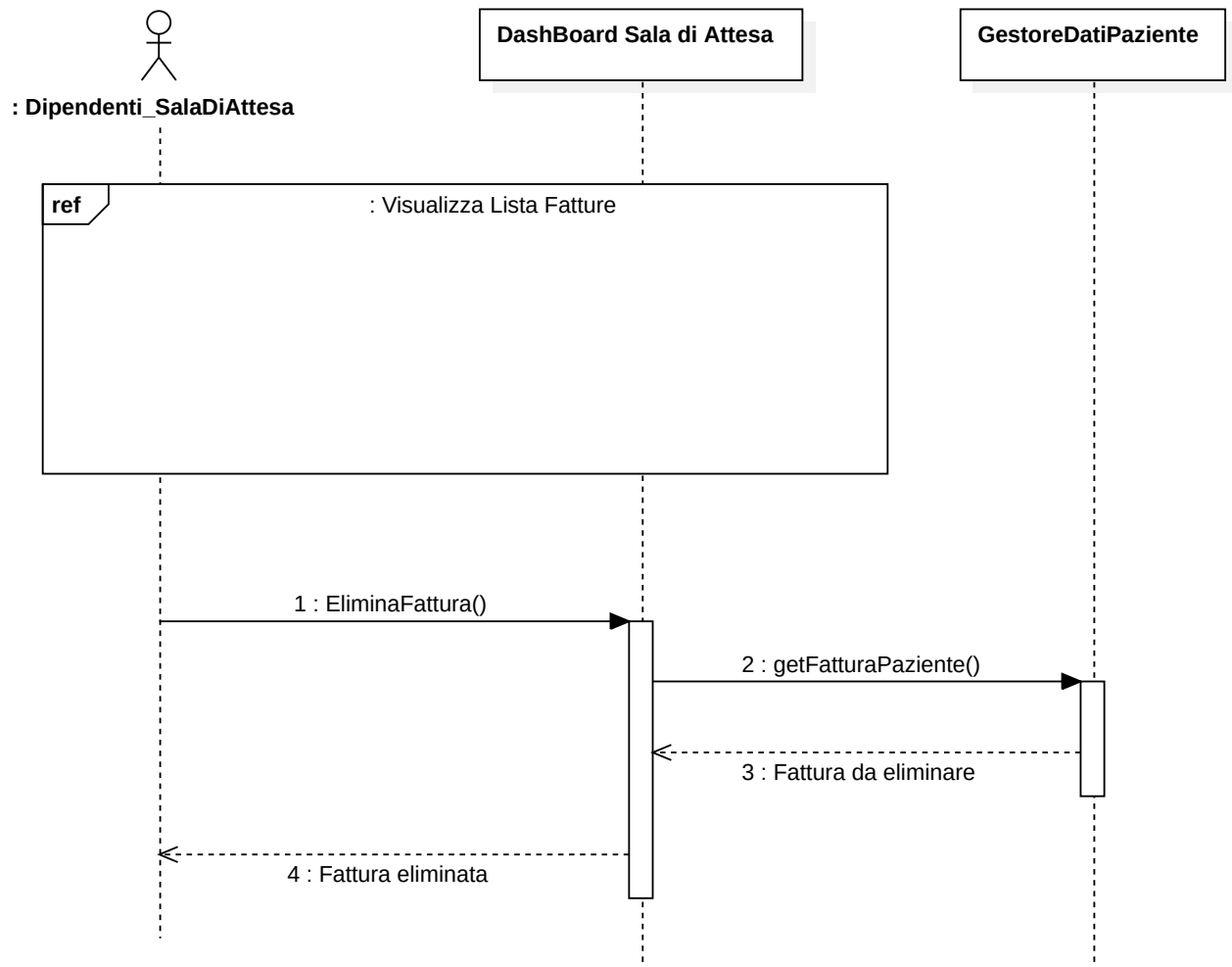
RaccoltaModifiche
Crea ed agg la
ModificaInAttesa
nella relativa lista

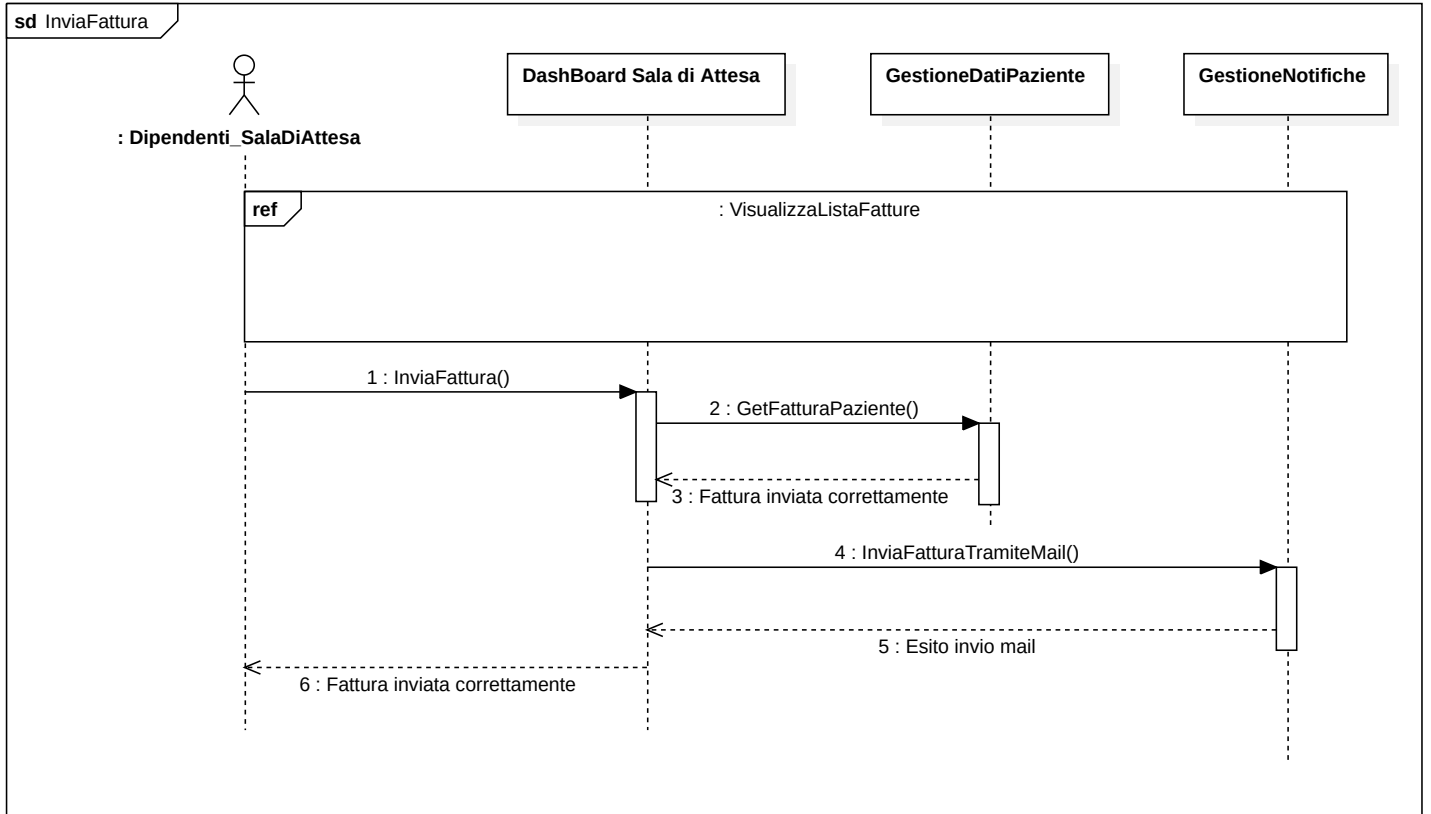


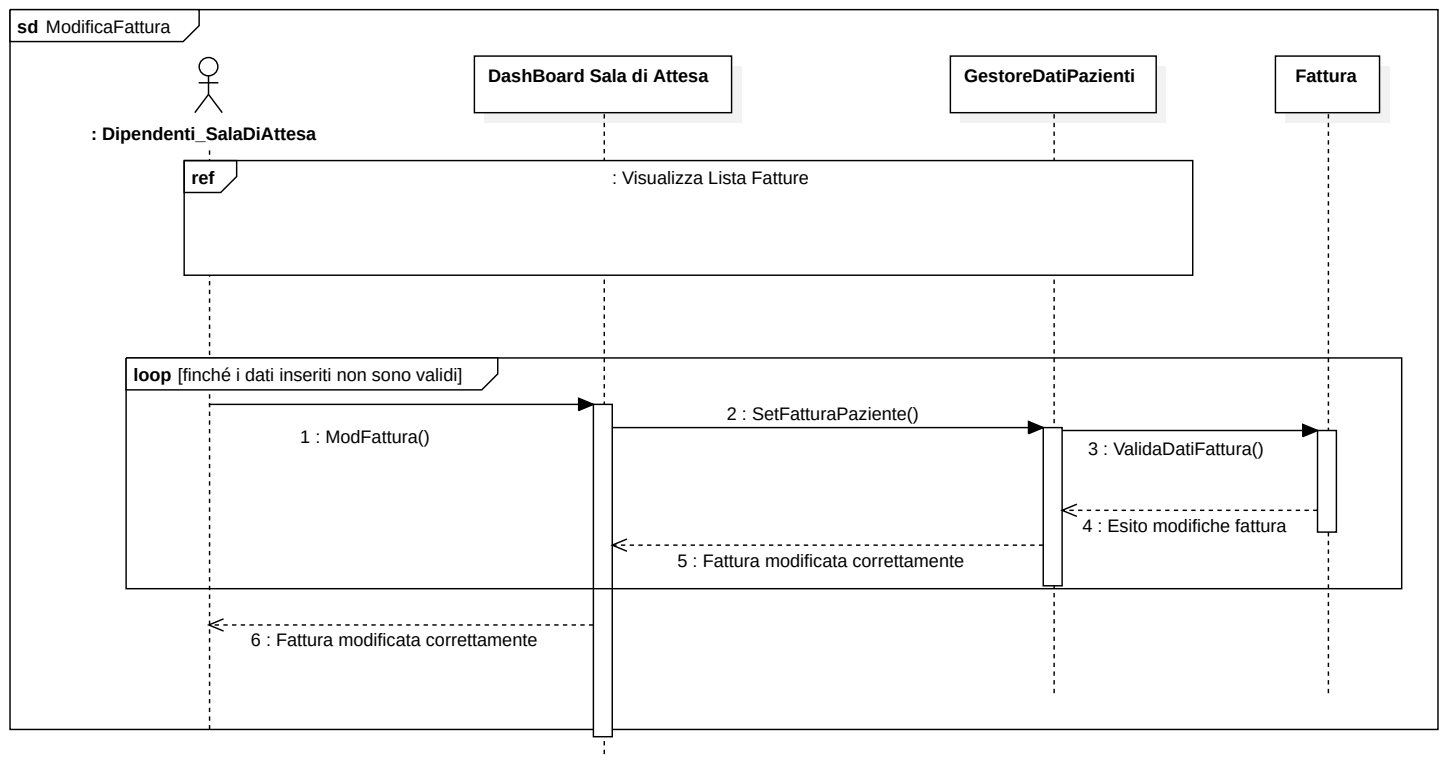
sd Aggiungi Fattura

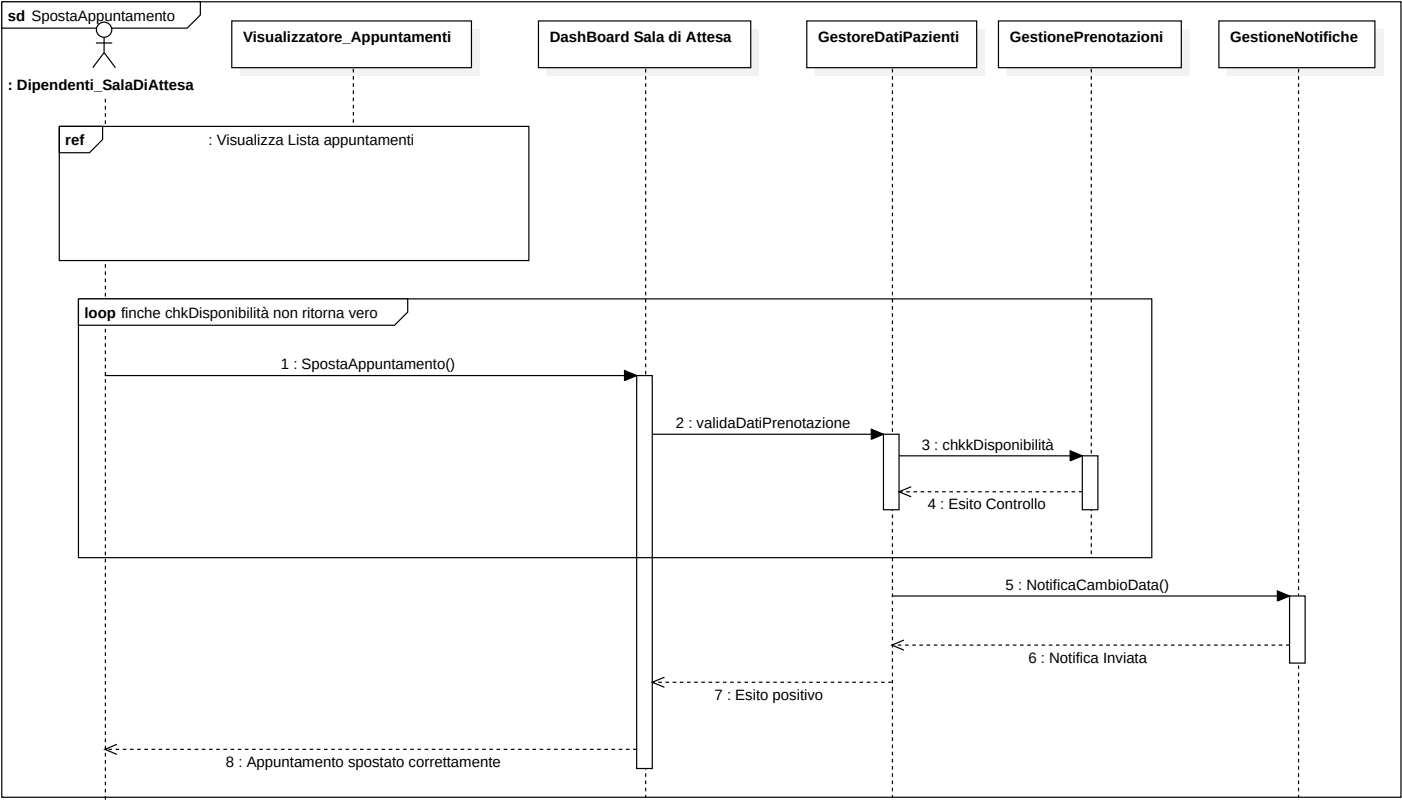


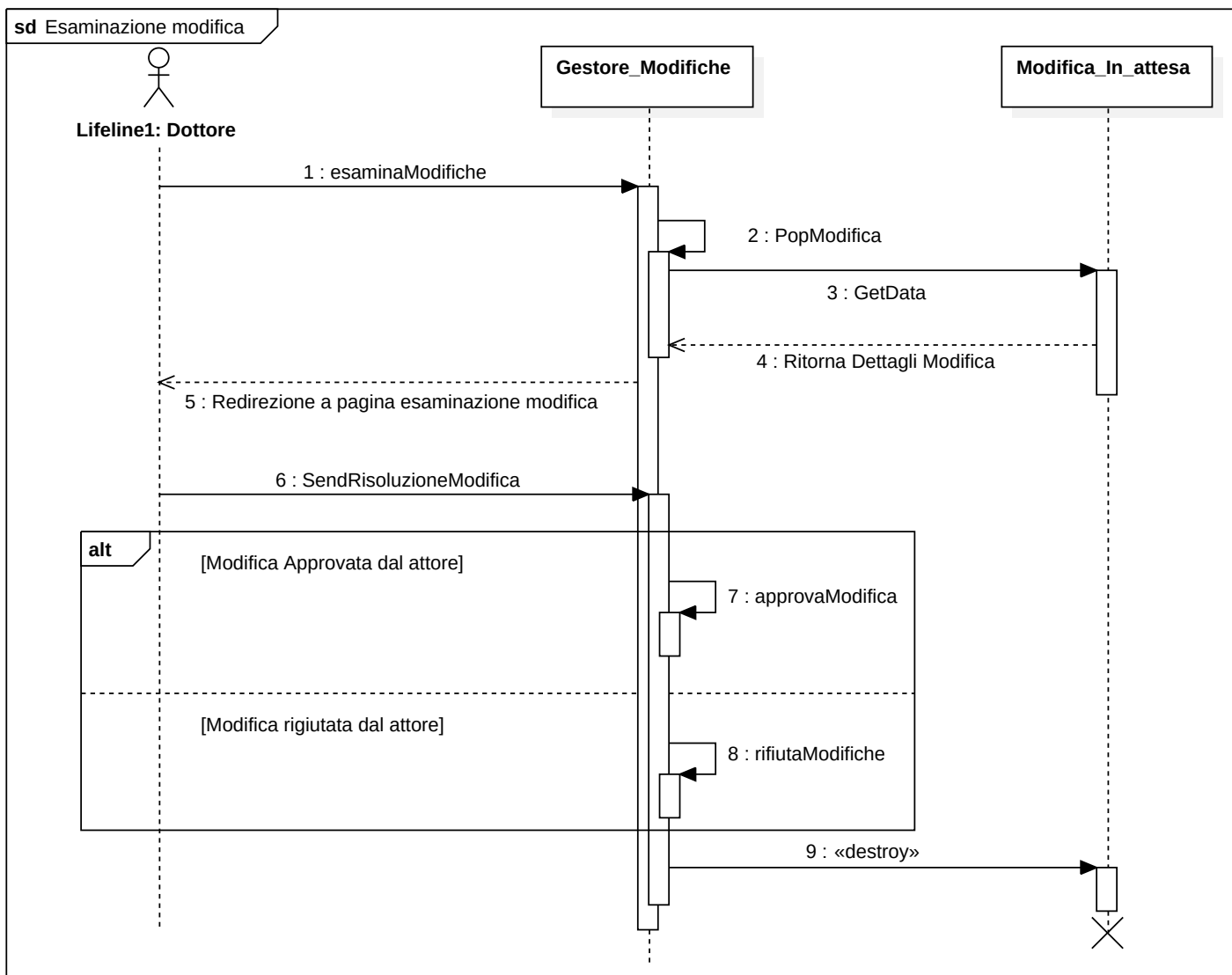
sd EliminaFattura











sd Init per reg dipendente



Lifeline1: Dottore

GestoreAccount

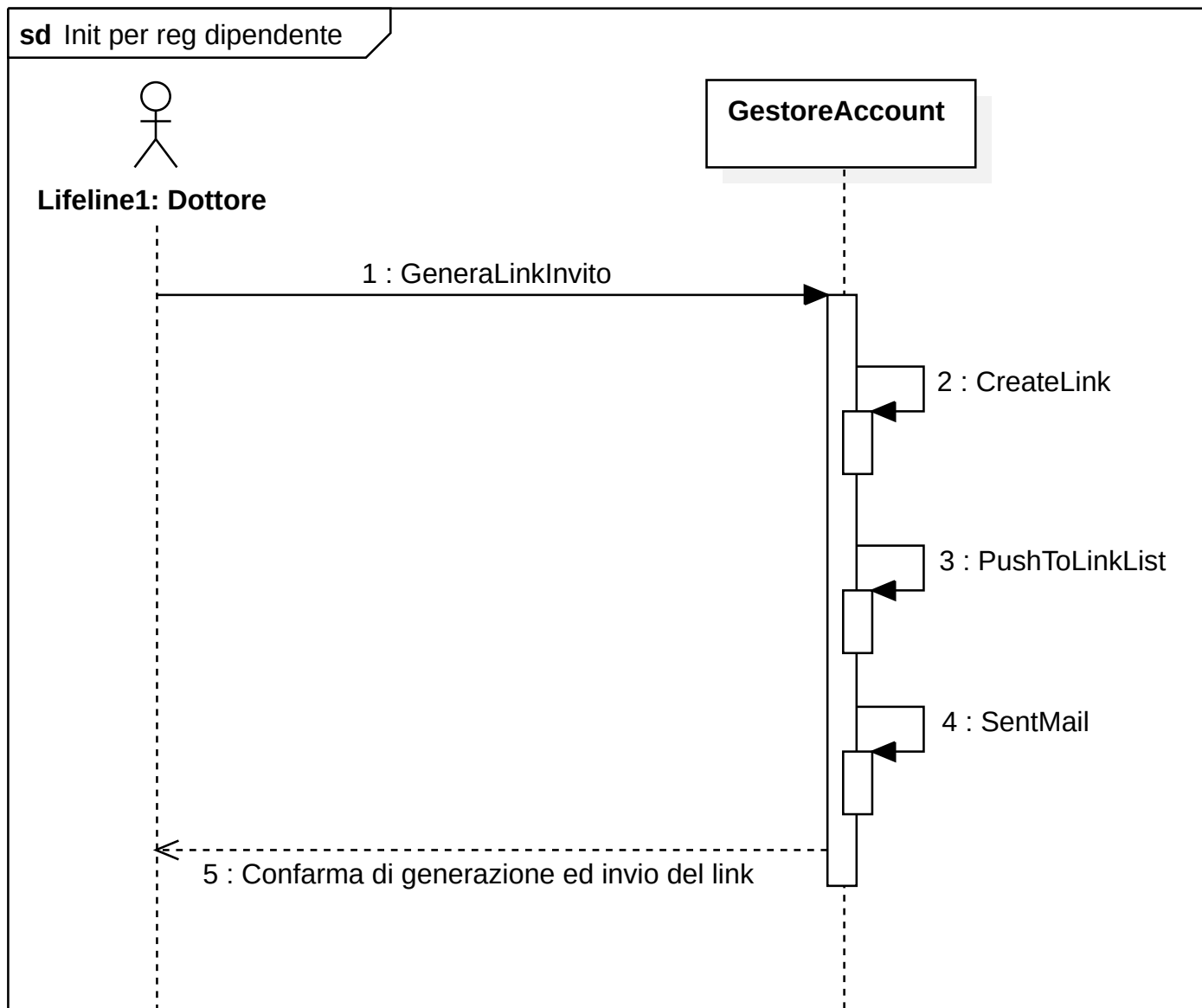
1 : GeneraLinkInvito

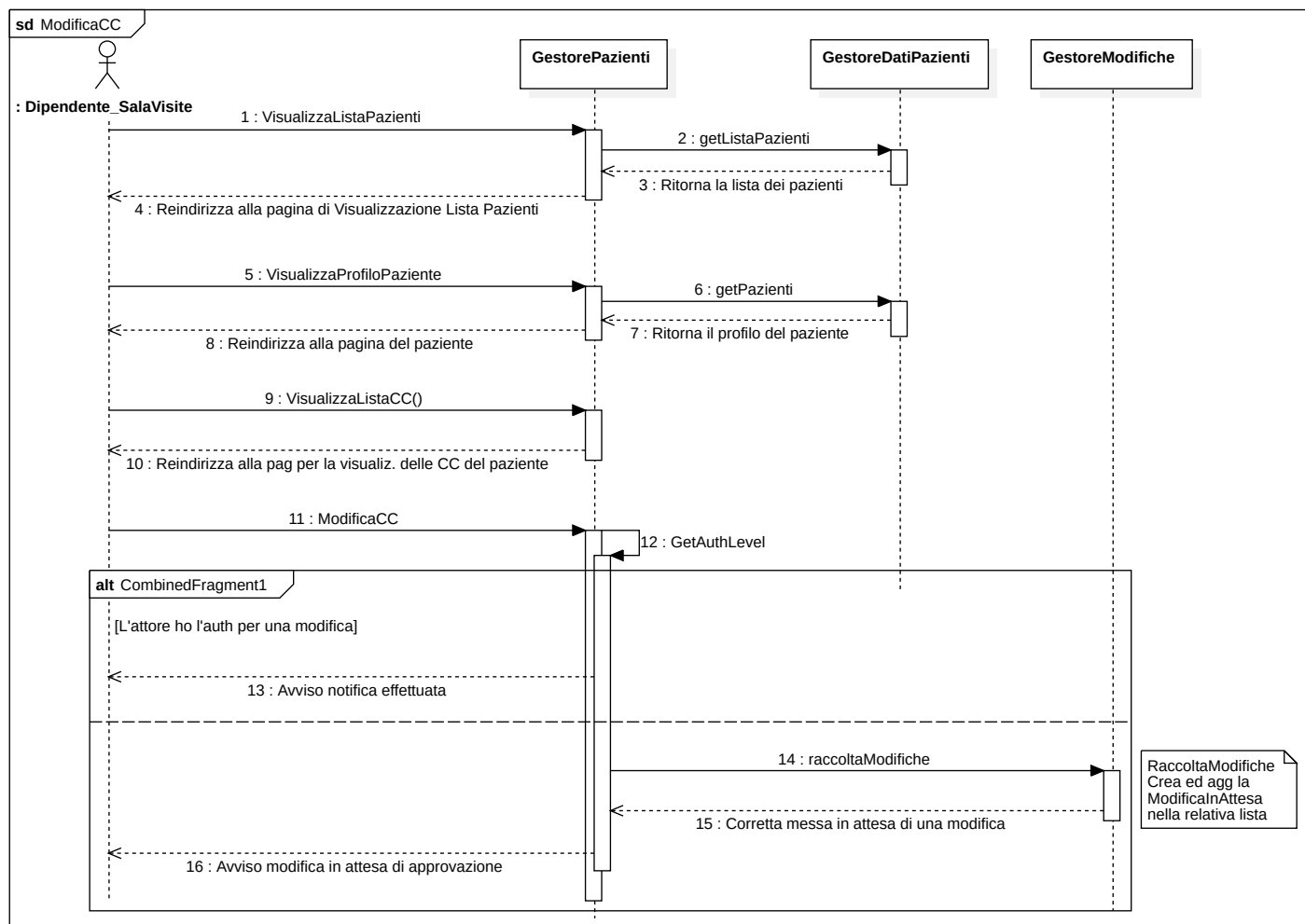
2 : CreateLink

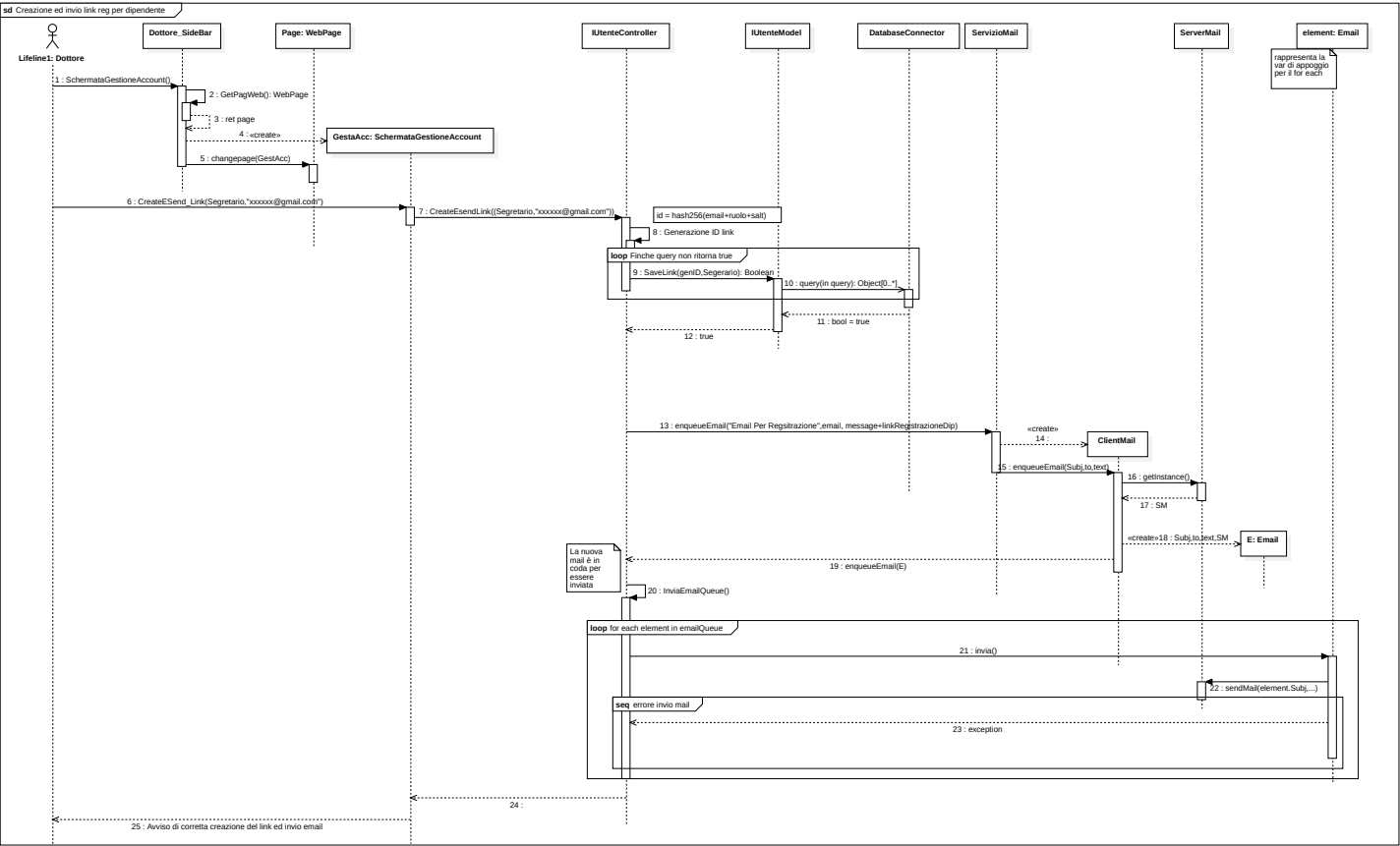
3 : PushToLinkList

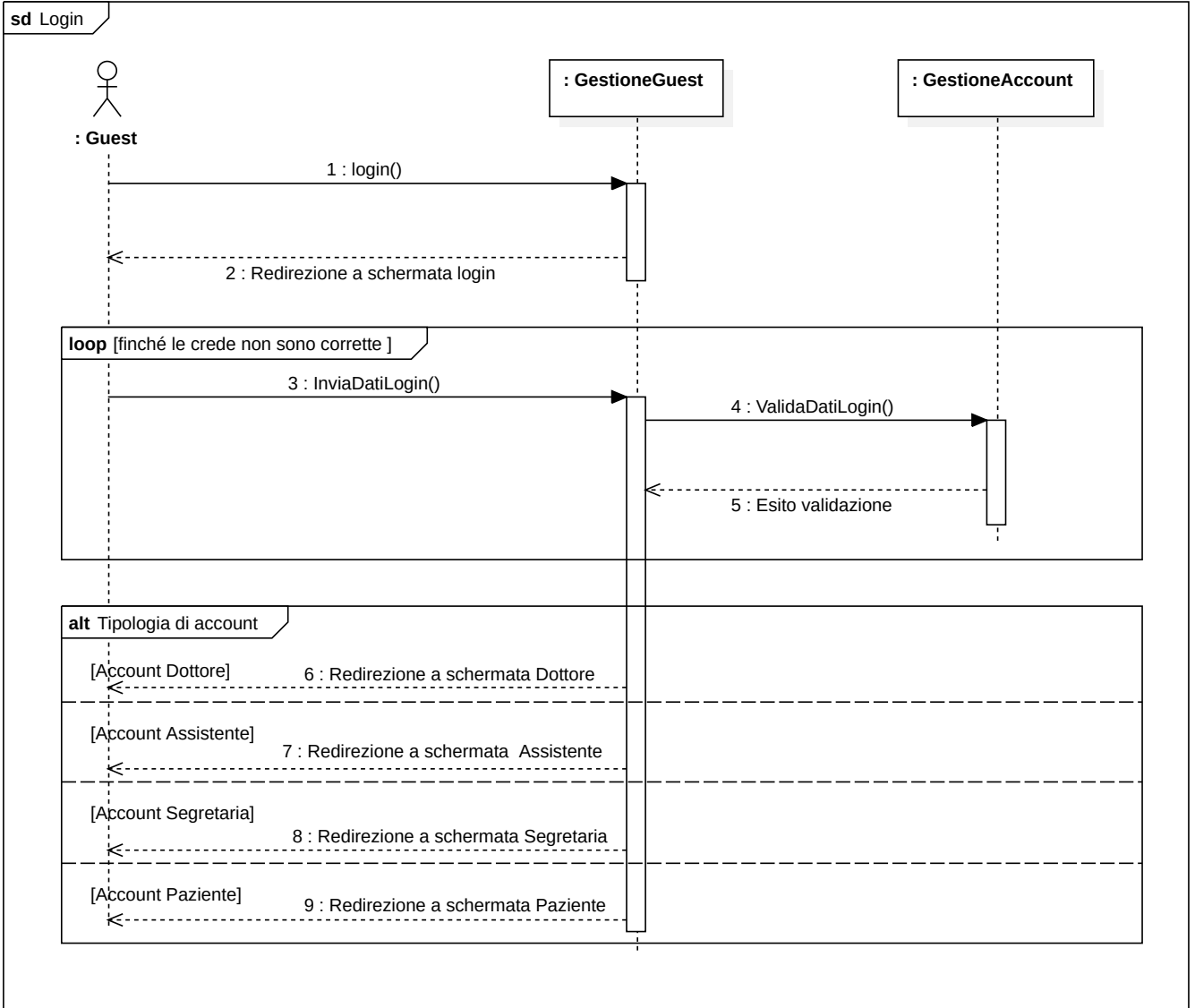
4 : SentMail

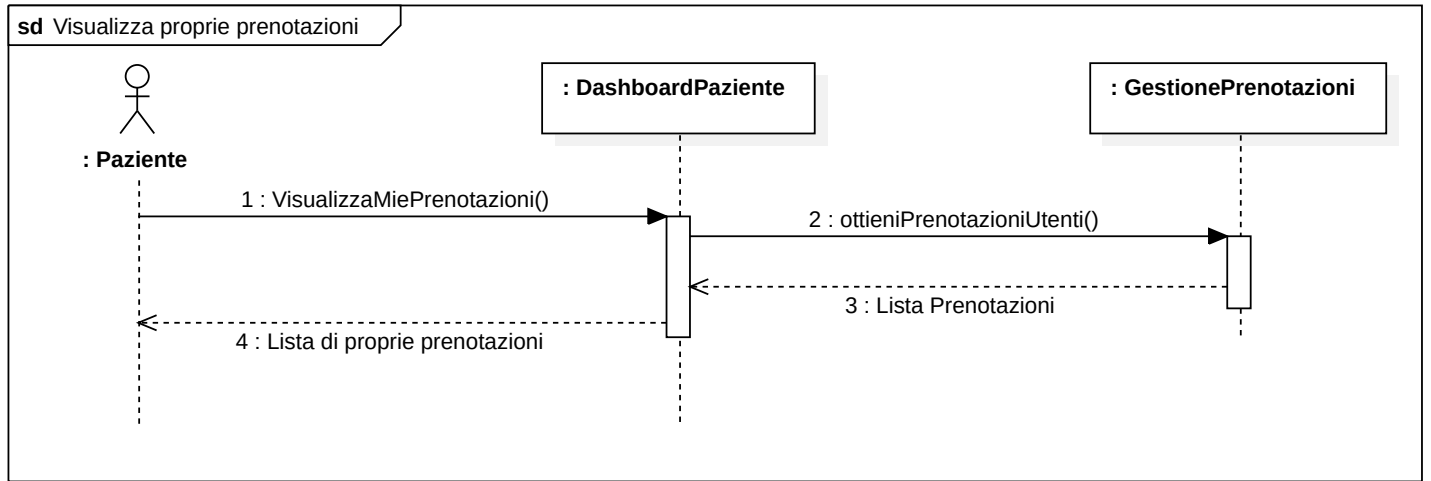
5 : Confarma di generazione ed invio del link



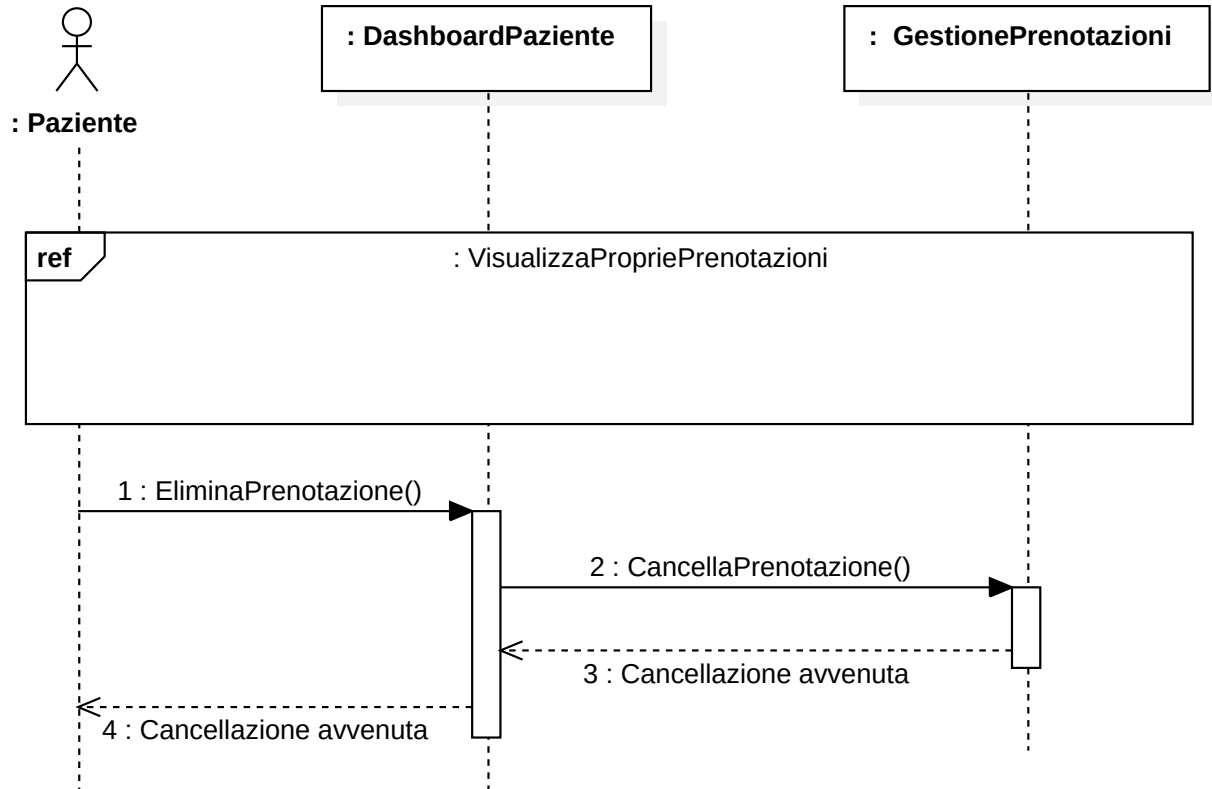




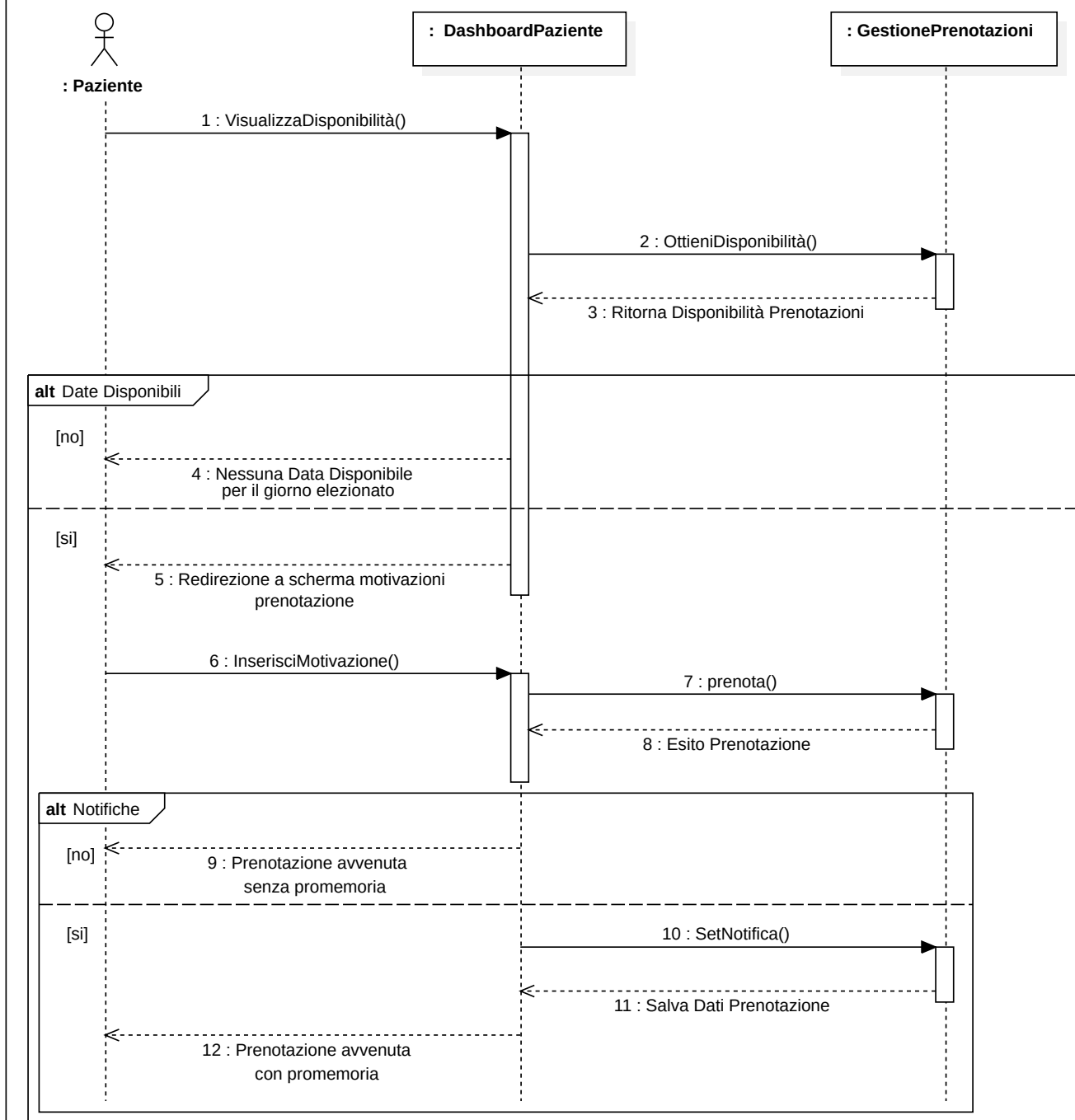




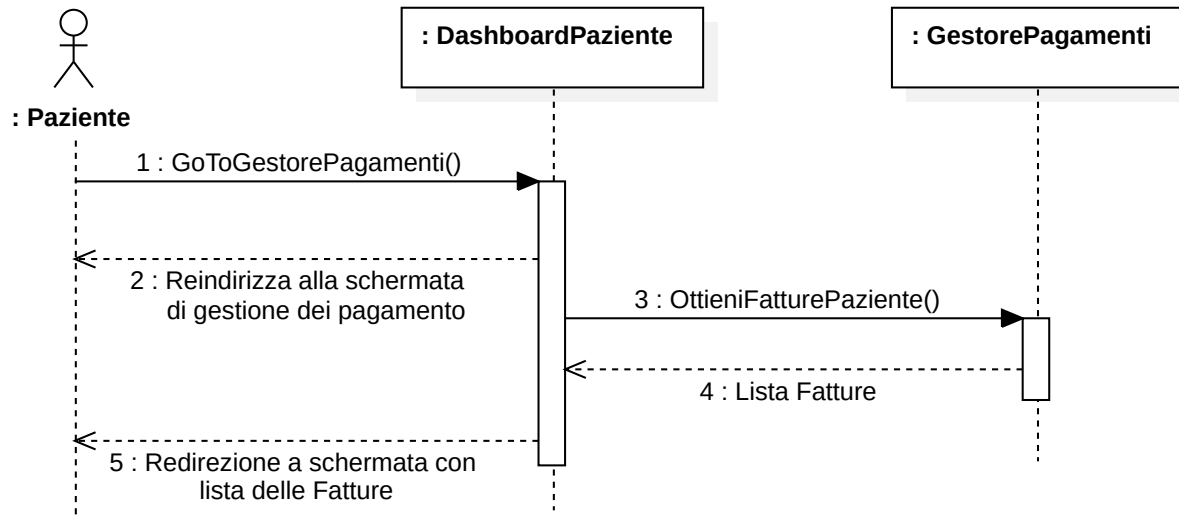
sd CancellaPrenotazione



sd Prenotazione



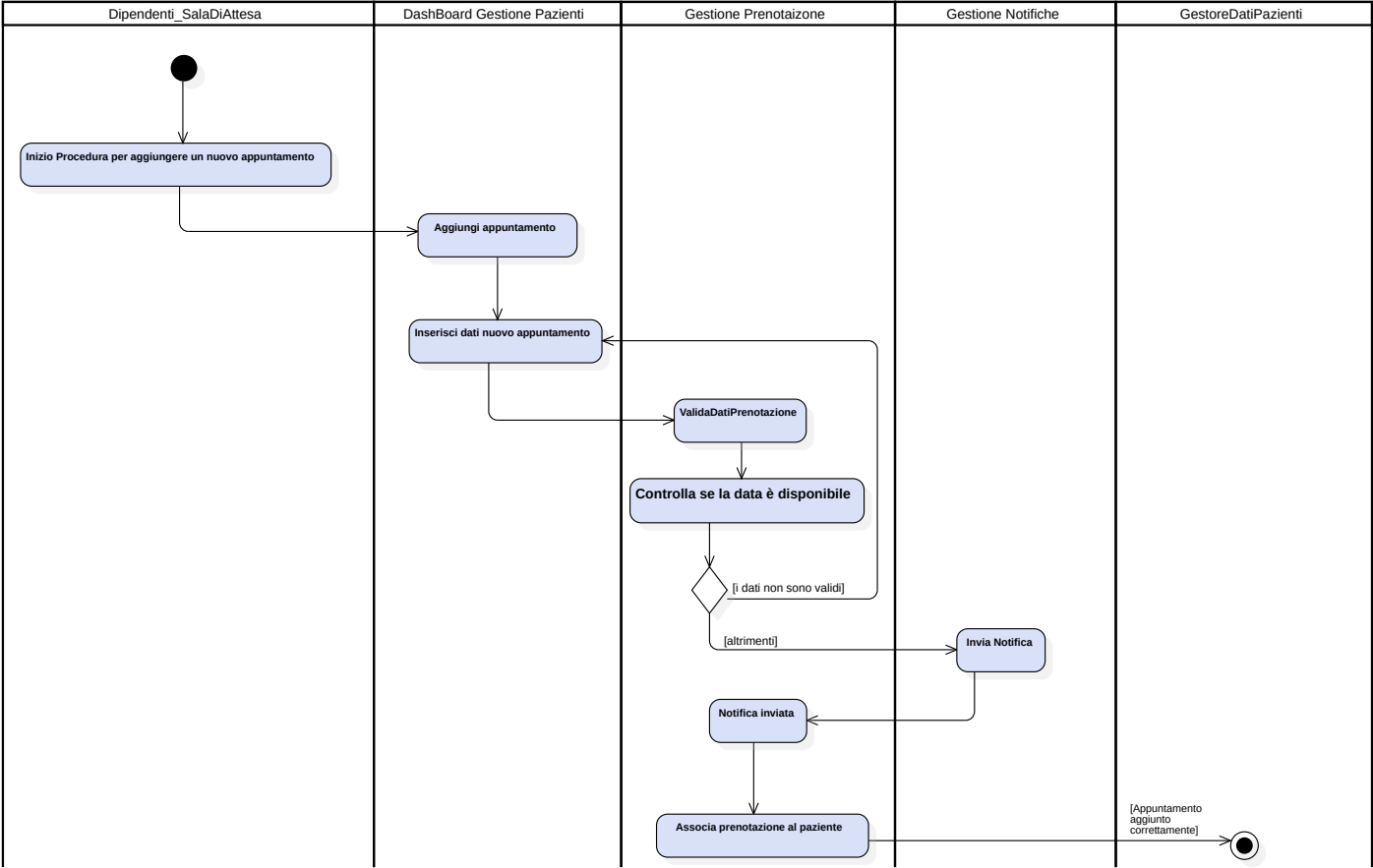
sd Visualizza Fatture



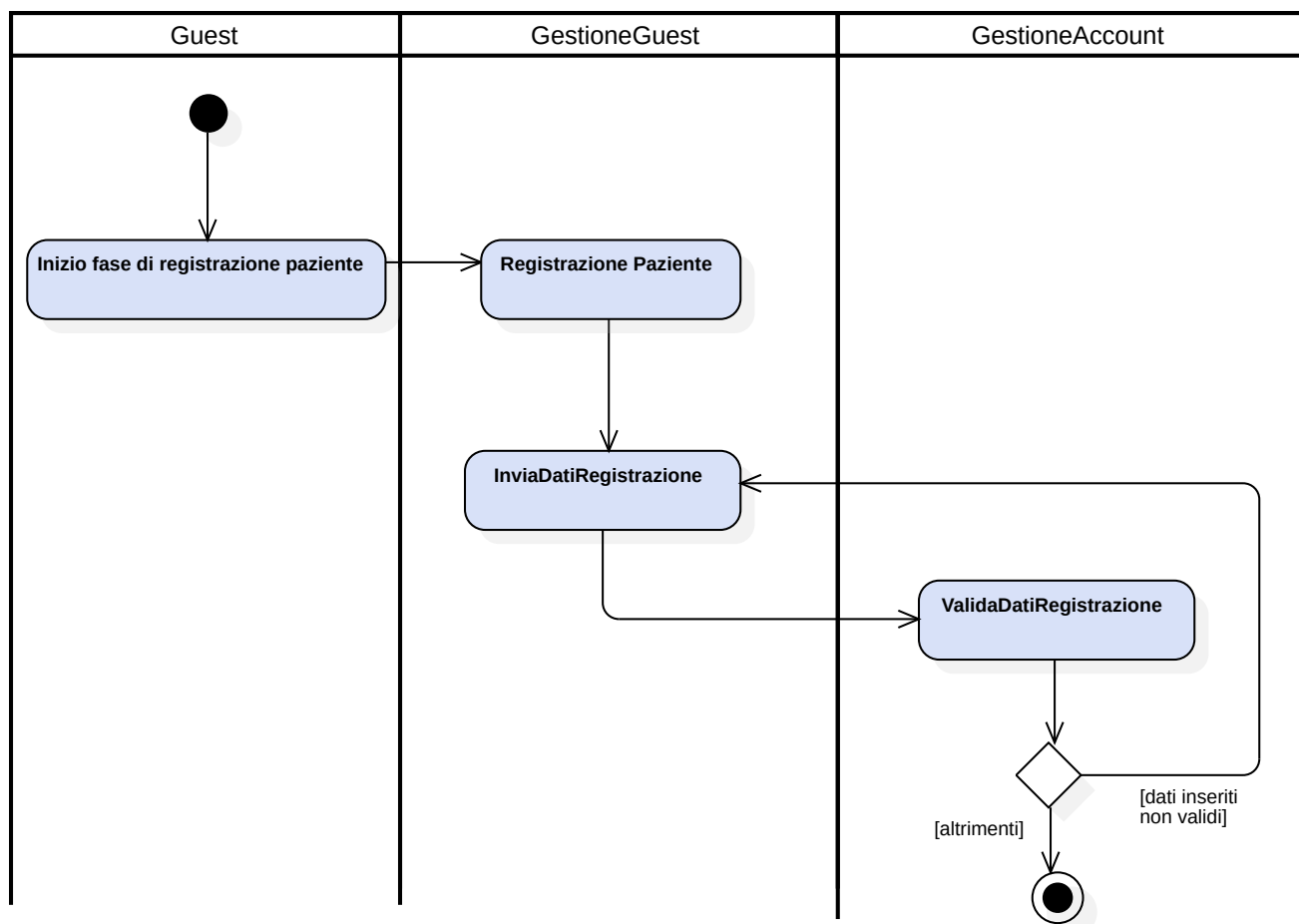
8. Diagramma delle attività--->

descrivono il flusso di eventi in un punto specifico del programma, descrivendo meglio anche le scelte possibili che possono essere percorse

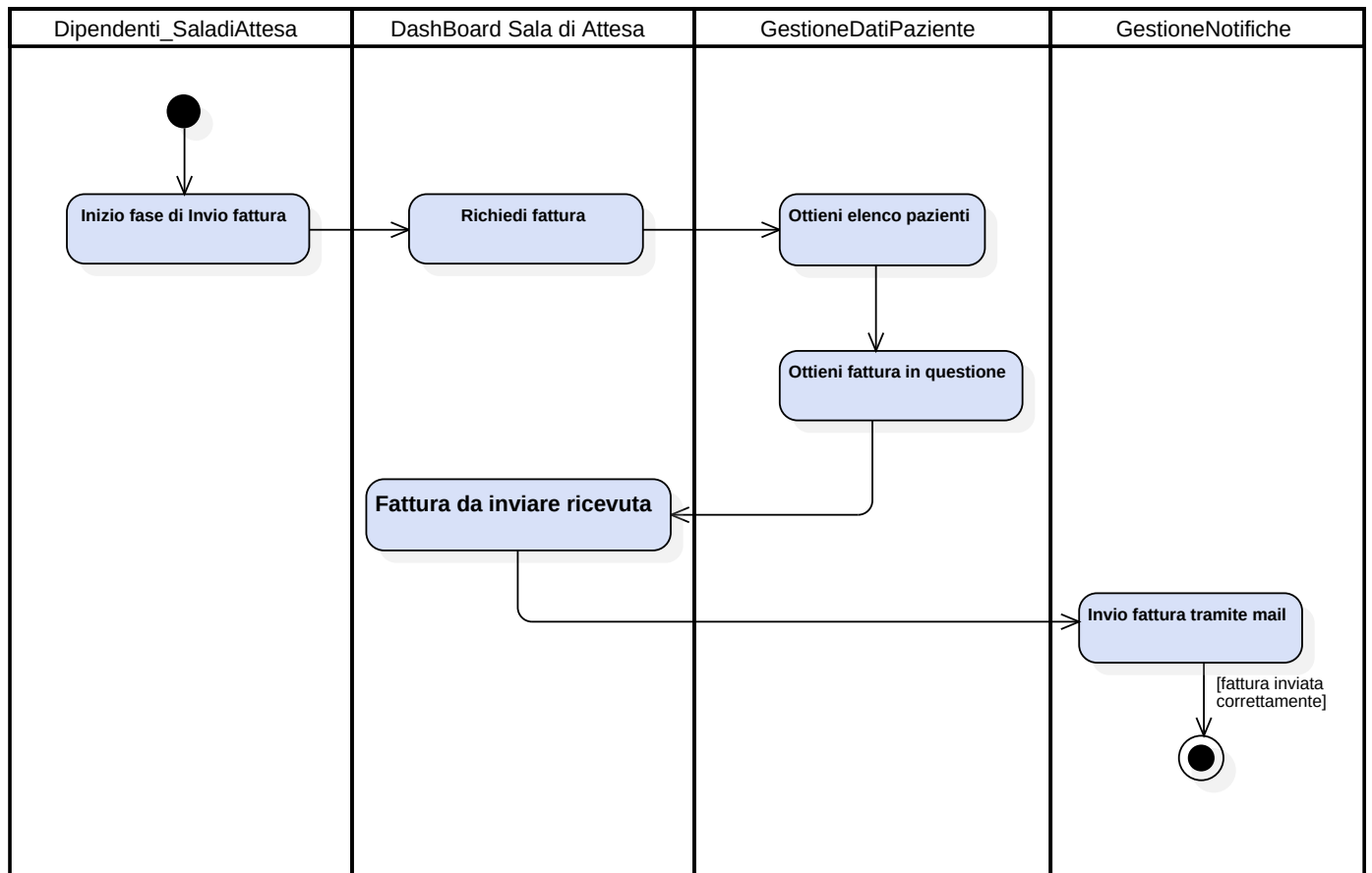
8.1 Aggiungi Appuntamento



8.2 Registrazione Paziente



8.3 Invio Fattura



9. Diagramma delle transizioni di stato---> Mostra i possibili stati dell'oggetto e le transizioni che provocano un cambiamento di stato

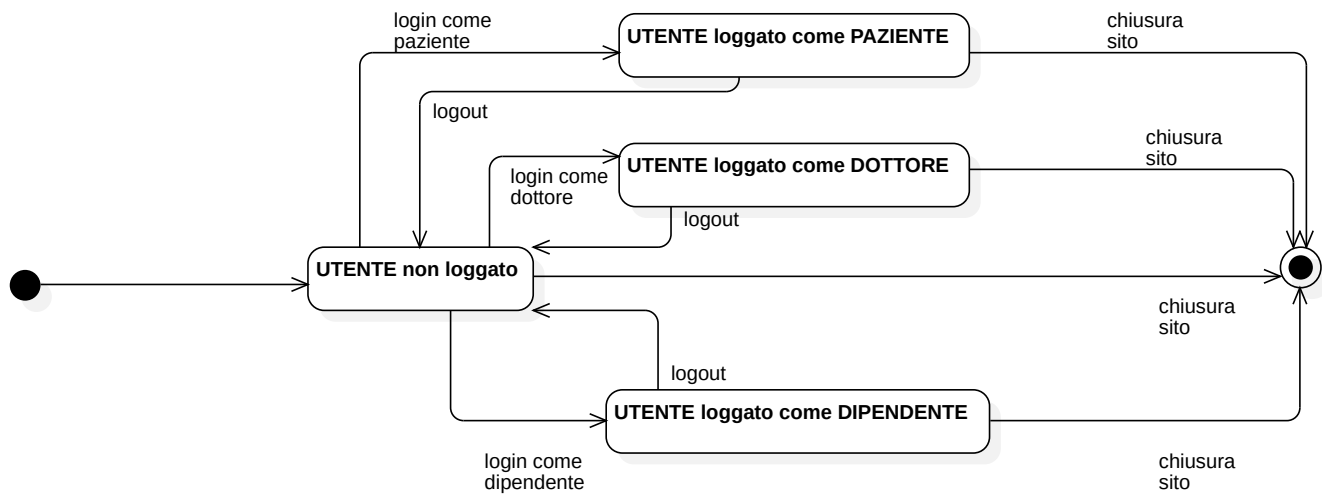


Diagramma delle transizioni di stato che descrive la variazione dello stato di login in base alle azioni dell'utente.

Questi diagrammi forniscono una panoramica del sistema mostrando le sue classi e le loro relazioni. In particolare il diagramma delle classi di un singolo software costituisce la base per la costruzione dell'infrastruttura attorno allo stesso, ed è alla base anche di altri diagrammi, statici e dinamici, che aiutano la comprensione e l'implementazione del prodotto.

[illegible]

Diagramma delle classi di design del model.

10.2 View

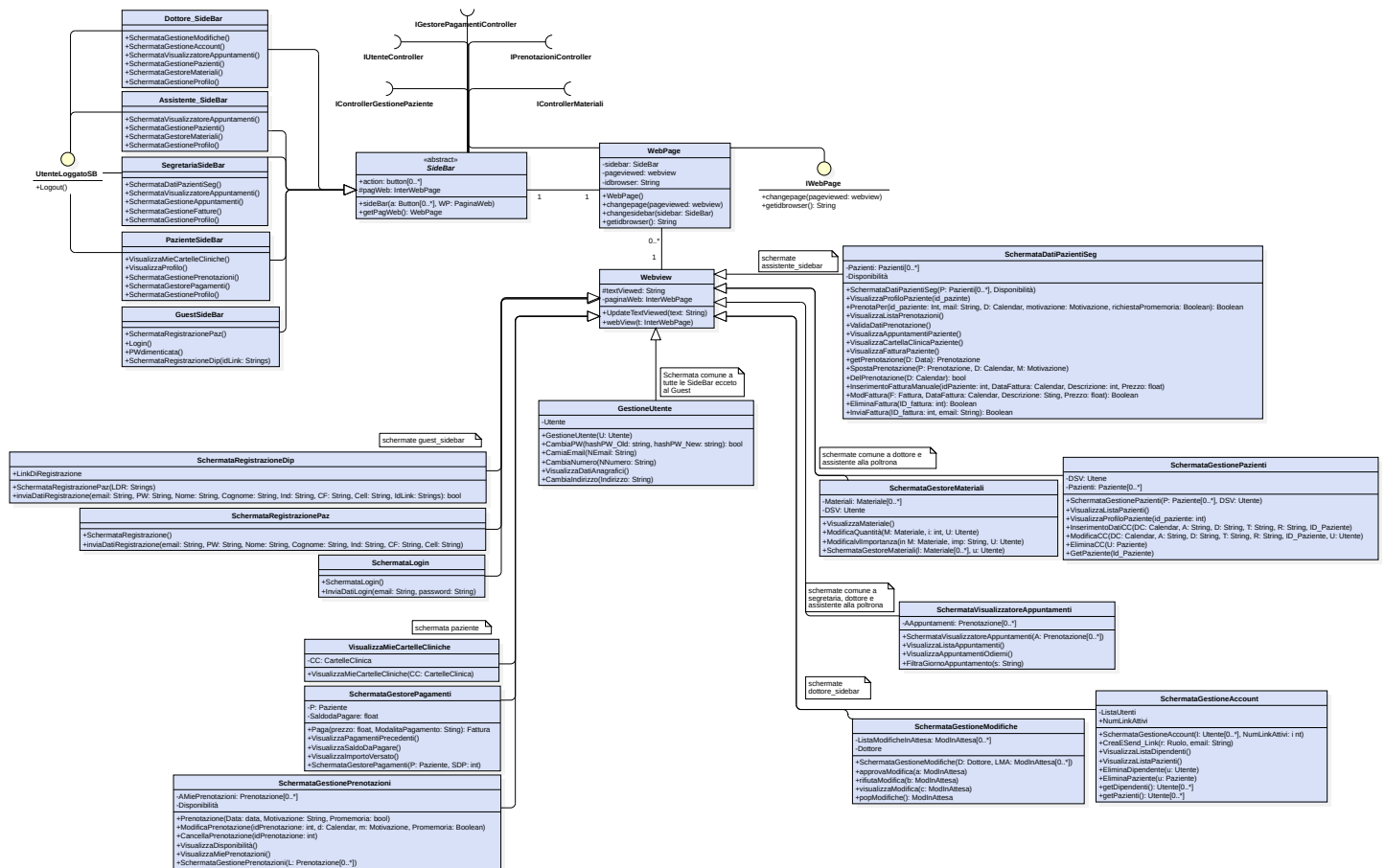


Diagramma delle classi di design del view.

10.3 Controller

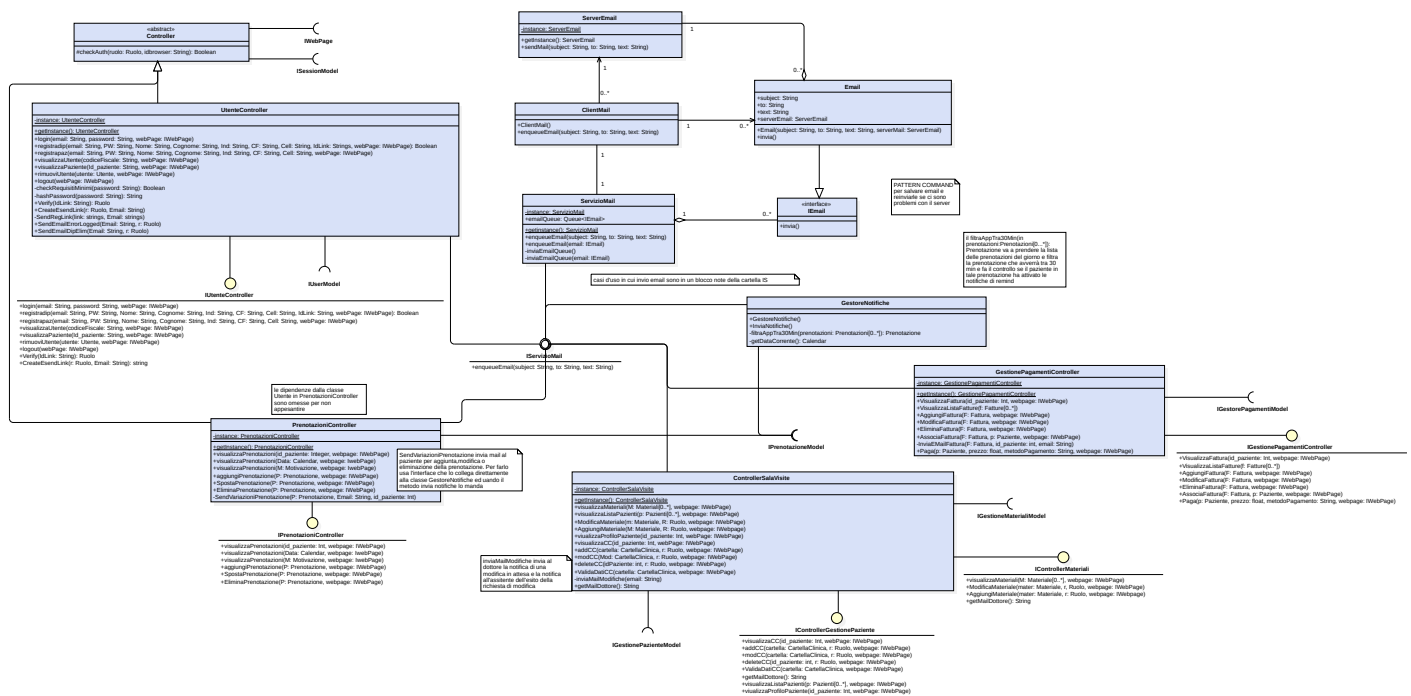


Diagramma delle classi di design del controller.

11. Pattern

Si è fatto uso dei seguenti design pattern.

11.1 Pattern architetturale

Si è scelto di utilizzare il pattern architetturale Model-View-Controller. La pagina web mostrata al client sarà la view, il controller gestirà gli input dell'utente che verranno comunicati attraverso la view e si occuperà di aggiornare o prelevare dati dal model e di aggiornare la view.

11.2 Singleton

Si è fatto uso del pattern Singleton laddove era necessario assicurarsi che fosse sempre presente una sola istanza di una specifica classe, come nel caso della classe ServizioEmail, in quanto la coda delle email deve essere unica.

DatabaseConnection
-instance: DatabaseConnection
+getInstance(): DatabaseConnection
+query(query: String): Object[0..*]

SessionCookies
-instance: SessionCookies
-cookies: cookie[0..*]
+getInstance(): SessionCookies
+setCookie(browser: String, key: String, value: String)
+getCookie(browser: String, key: String)
+unsetCookie(browser: String, key: String)

MediatorAuth
-instance: MediatorAuth
-ListaModinAttesa
-database: DataBaseConnection
+getInstance(): MediatorAuth
+mediateQuery(in query: String, in ruolo: Ruolo): Object[0..*]
+CreateQuery(query: String) ModinAttesa()
+PopModifica(ruolo: Ruolo)
+RisultatoEsaminazione(Ris: Bool, Mod: ModinAttesa)

ServizioMail
-instance: ServizioMail
+emailQueue: Queue<Email>
+getInstance(): ServizioMail
+enqueueEmail(subject: String, to: String, text: String)
+enqueueEmail(email: IEmail)
+inviaEmailQueue()
+inviaEmailQueue(email: IEmail)

UtenteController
-instance: UtenteController
+getInstance(): UtenteController
+login(email: String, password: String, webPage: IWebPage)
+register(email: String, PW: String, Nome: String, Cognome: String, Ind: String, CF: String, Cell: String, IdLink: String, webPage: IWebPage): Boolean
+registrapaz(email: String, PW: String, Nome: String, Cognome: String, Ind: String, CF: String, Cell: String, webPage: IWebPage)
+visualizzaUtente(codiceFiscale: String, webPage: IWebPage)
+visualizzaPaziente(id_paziente: String, webPage: IWebPage)
+rimuoviUtente(utente: Utente, webPage: IWebPage)
+logout(webPage: IWebPage)
+checkRequisitiMinimi(password: String): Boolean
+hashPassword(password: String): String
+VerifyIdLink(String): Ruolo
+CreateEsendLink(r: Ruolo, Email: String)
+SendRegLink(link: String, Email: String)
+SendEmailErrorLogged(Email: String, r: Ruolo)
+SendEmailDipElim(Email: String, r: Ruolo)

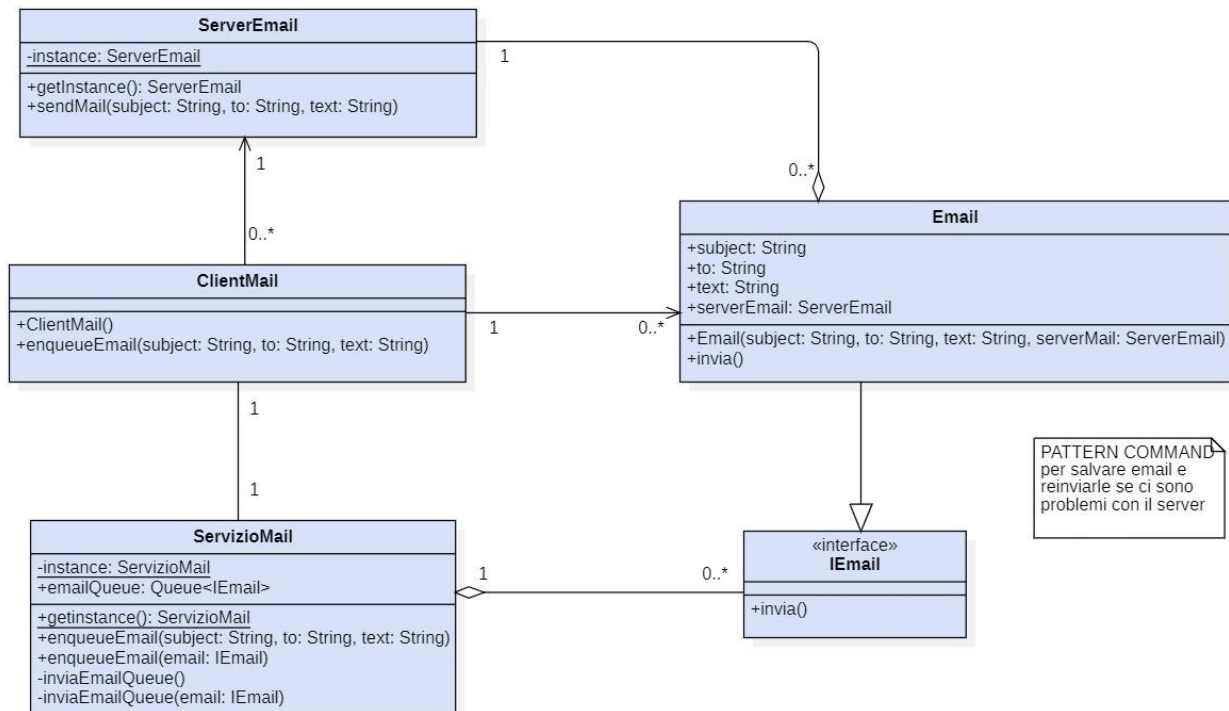
ControllerSalaVisite
-instance: ControllerSalaVisite
+getInstance(): ControllerSalaVisite
+visualizzaMateriali(M: Materiale[0..*], webPage: IWebPage)
+visualizzaListaPazienti(p: Paziente[0..*], webPage: IWebPage)
+ModificaMateriale(m: Materiale, r: Ruolo, webPage: IWebPage)
+AggiungiMateriale(M: Materiale, r: Ruolo, webPage: IWebPage)
+visualizzaProfiloPaziente(id_paziente: Int, webPage: IWebPage)
+visualizzaCC(id_paziente: Int, webPage: IWebPage)
+addCC(cartella: CartellaClinica, r: Ruolo, webPage: IWebPage)
+modCC(Mod: CartellaClinica, r: Ruolo, webPage: IWebPage)
+deleteCC(idPaziente: Int, r: Ruolo, webPage: IWebPage)
+ValidaDatCC(cartella: CartellaClinica, webPage: IWebPage)
+inviaMailModifiche(email: String)
+getMailDottore(): String

PrenotazioniController
-instance: PrenotazioniController
+getInstance(): PrenotazioniController
+visualizzaPrenotazioni(id_paziente: Integer, webPage: IWebPage)
+visualizzaPrenotazioni(Data: Calendar, webPage: IWebPage)
+visualizzaPrenotazioni(M: Motivazione, webPage: IWebPage)
+aggiungiPrenotazione(P: Prenotazione, webPage: IWebPage)
+SpostaPrenotazione(P: Prenotazione, webPage: IWebPage)
+EliminaPrenotazione(P: Prenotazione, webPage: IWebPage)
+SendVariazioniPrenotazione(P: Prenotazione, Email: String, id_paziente: Int)

GestionePagamentiController
-instance: GestionePagamentiController
+getInstance(): GestionePagamentiController
+VisualizzaFatture(id_paziente: Int, webPage: IWebPage)
+VisualizzaListaFatture(f: Fatture[0..*])
+AggiungiFattura(F: Fattura, webPage: IWebPage)
+ModificaFattura(F: Fattura, webPage: IWebPage)
+EliminaFattura(F: Fattura, webPage: IWebPage)
+AssociaFattura(F: Fattura, p: Paziente, webPage: IWebPage)
+InviaEmailFattura(F: Fattura, id_paziente: Int, email: String)
+Paga(p: Paziente, prezzo: float, metodoPagamento: String, webPage: IWebPage)

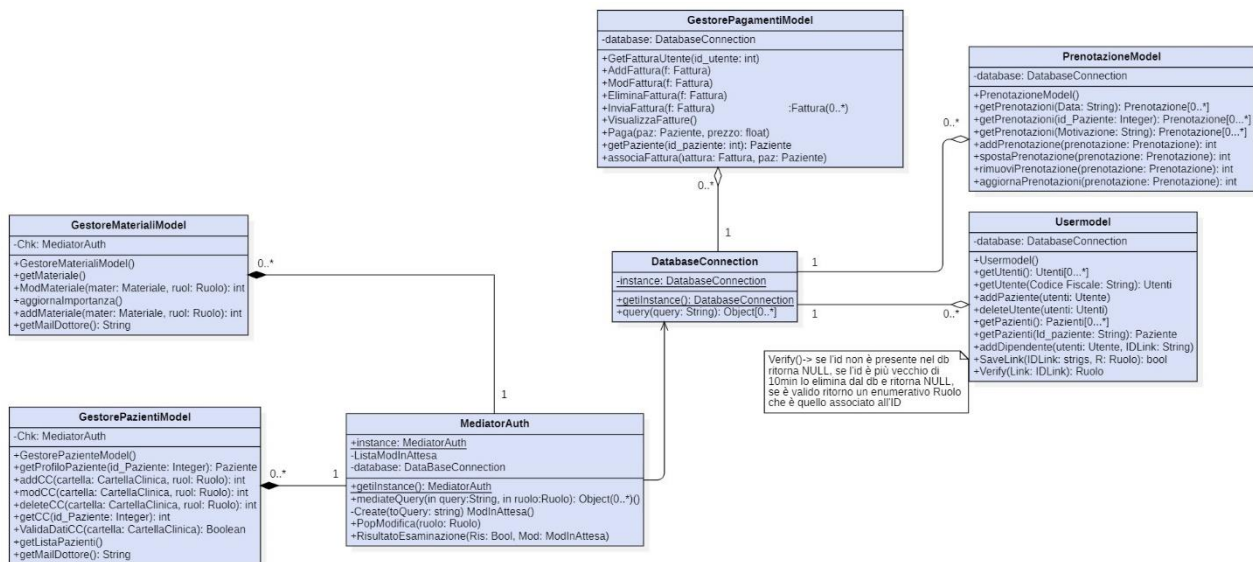
11.3 Command

Si è fatto uso del pattern Command in quanto era necessario trasformare i metodi di invio email in degli oggetti, in modo da conservare una coda di email da continuare a provare a inviare a rimuoverle dalla coda solo quando la classe EmailServer non genera eccezioni.



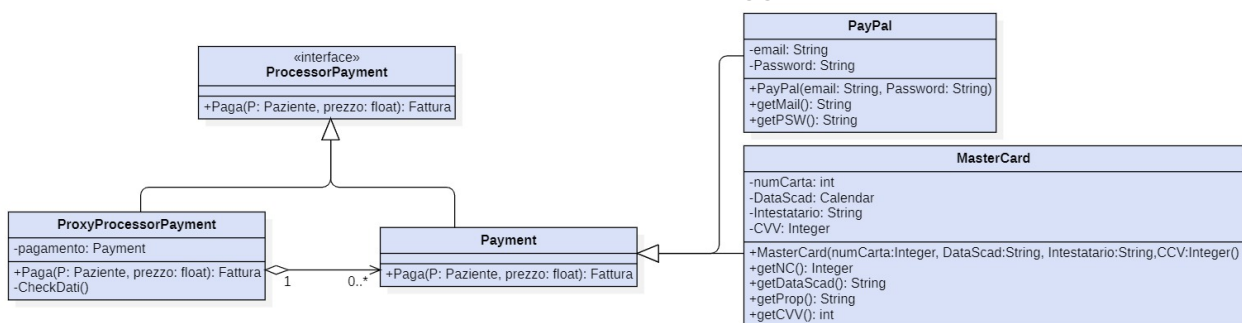
11.4 Facade

Si è fatto uso del pattern Facade in quanto era necessario astrarre una funzionalità complessa (la comunicazione con il database) e fornire al resto del sistema dei metodi per effettuare le operazioni di interesse a più alto livello.



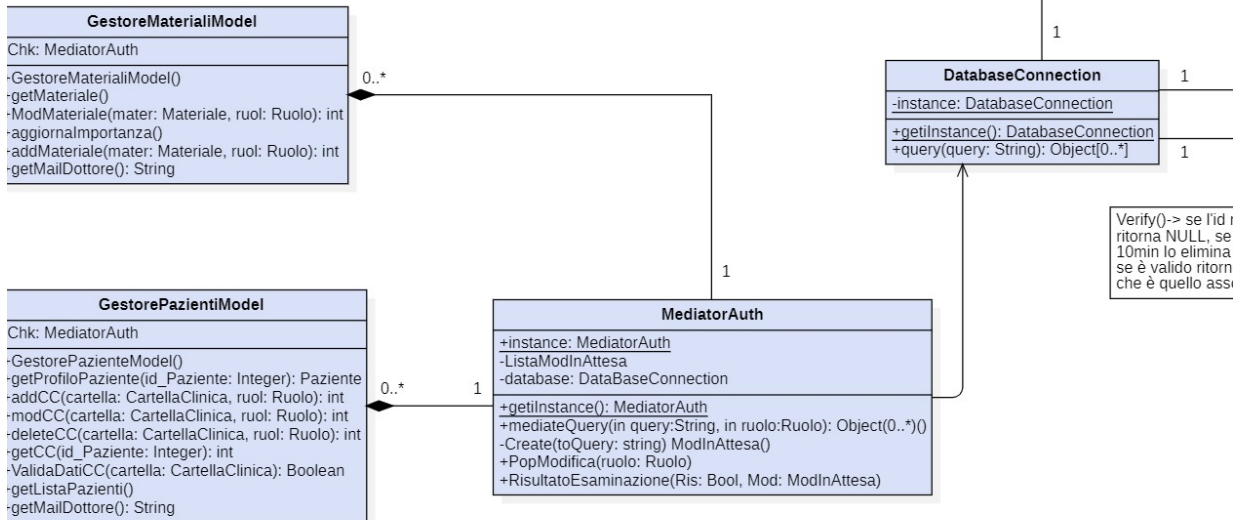
11.5 Proxy

Il pattern Proxy ci permette di avere una maggiore protezione nella fase di pagamento dal sistema; l'utente infatti comunicherà con una infrastruttura posta a metà tra il sistema e il circuito bancario, assicurando una maggior sicurezza.

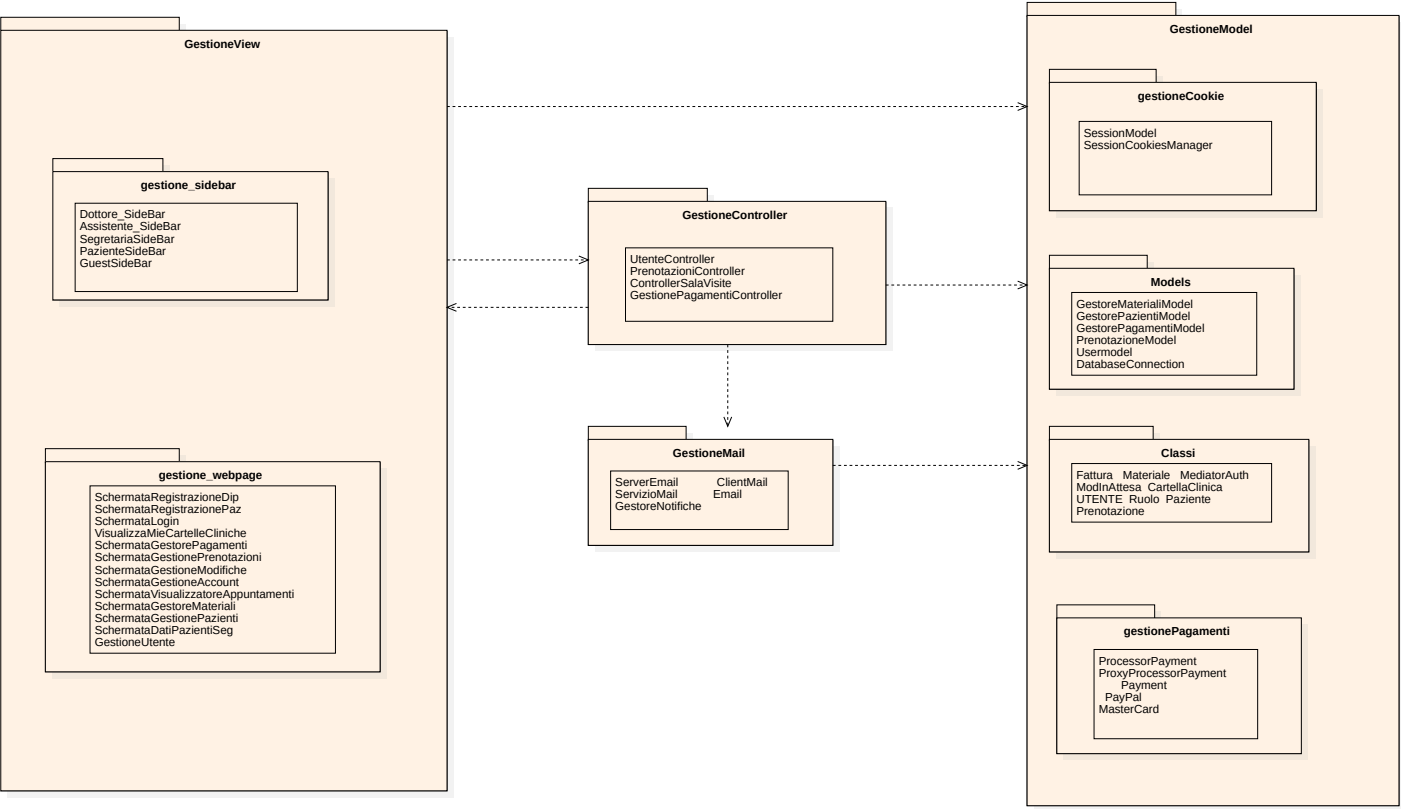


11.6 Mediator

Si tratta di un pattern che permette lo scambio di messaggi tra diversi attori tramite un intermediario. In questo modo gli attori sono collegati indirettamente tramite un intermediario. Disaccoppiare gli attori consente di gestire meglio una serie di problematiche come: gestione delle autorizzazioni e modifiche più rapide.

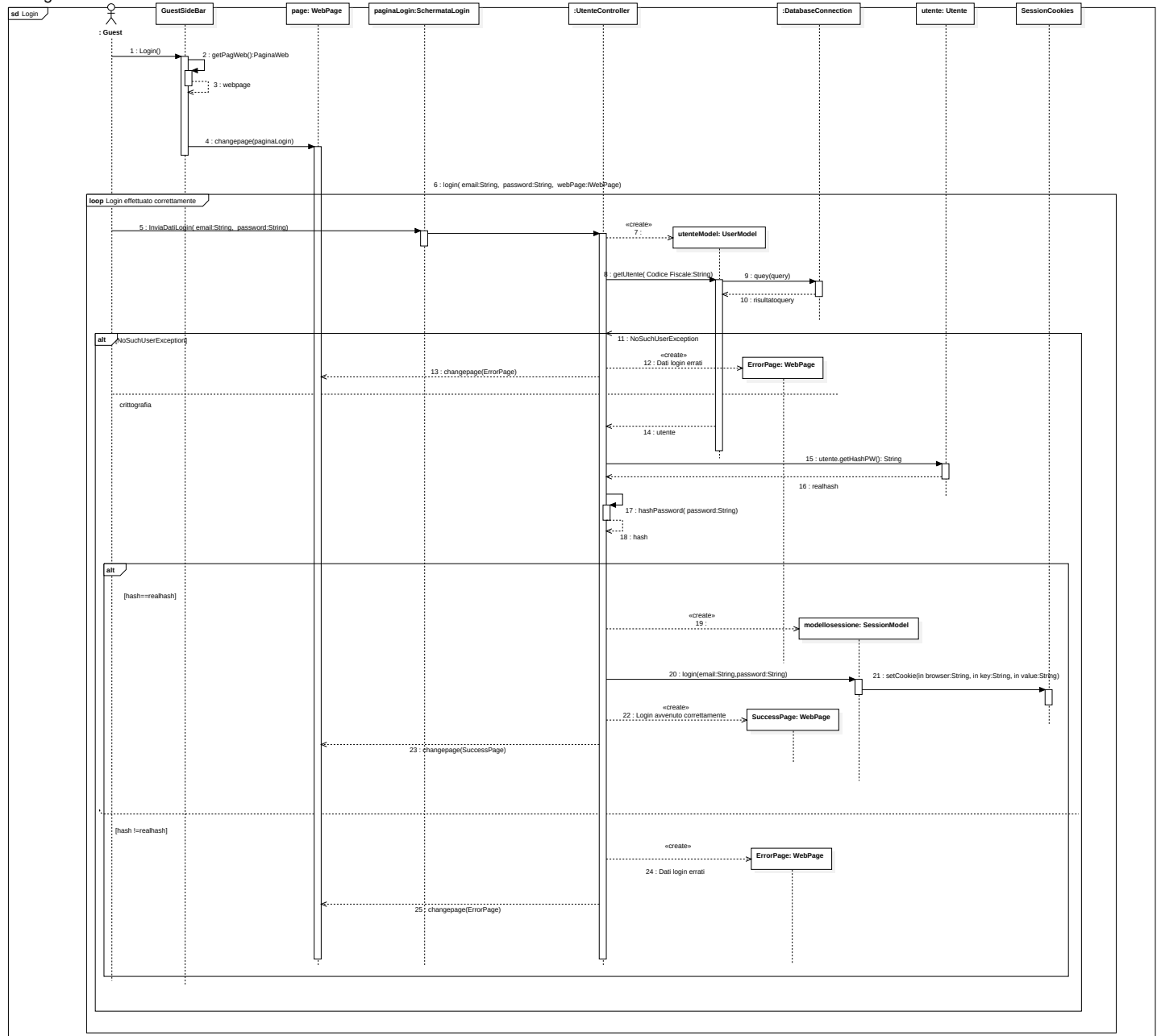


12. Diagramma dei package di design

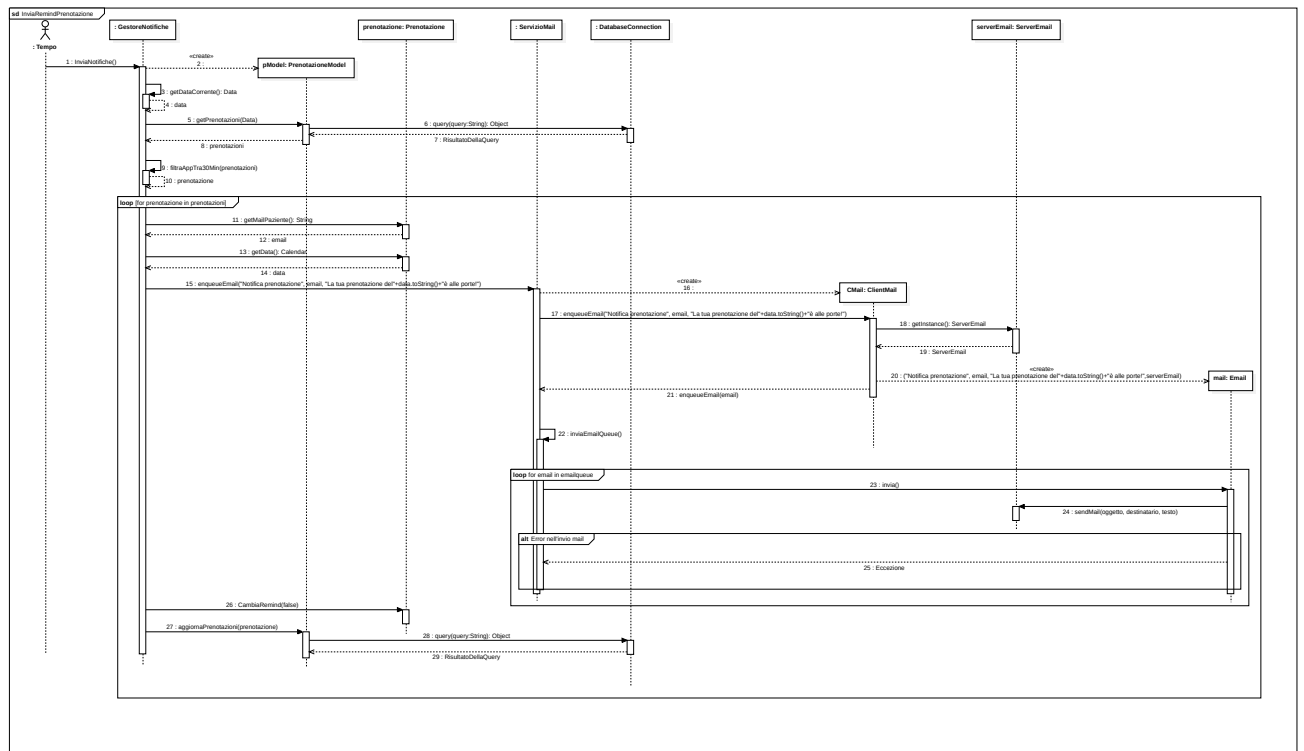


13. Diagrammi delle sequenze di design

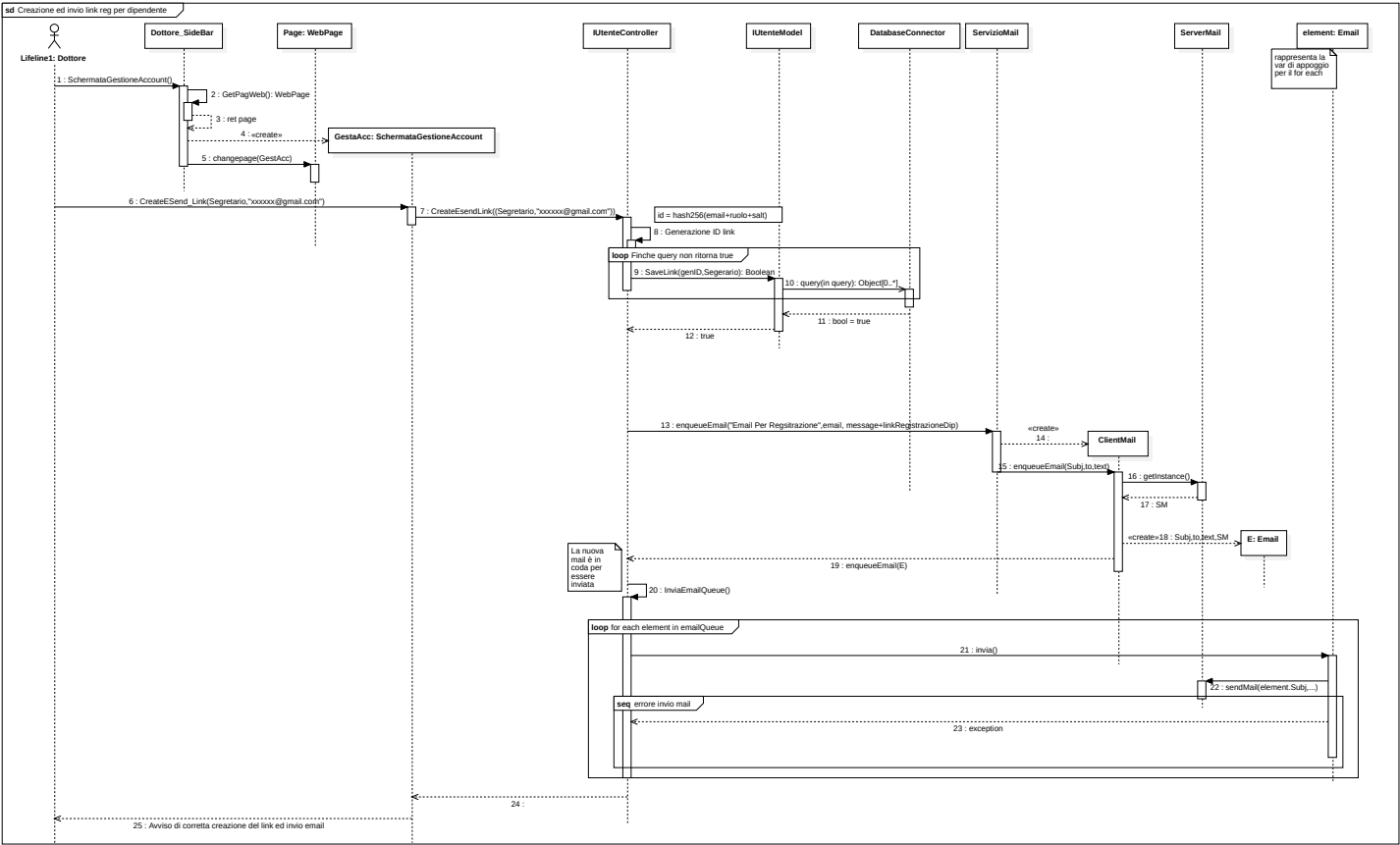
13.1 Login



13.2 InviaRemindPrenotazione

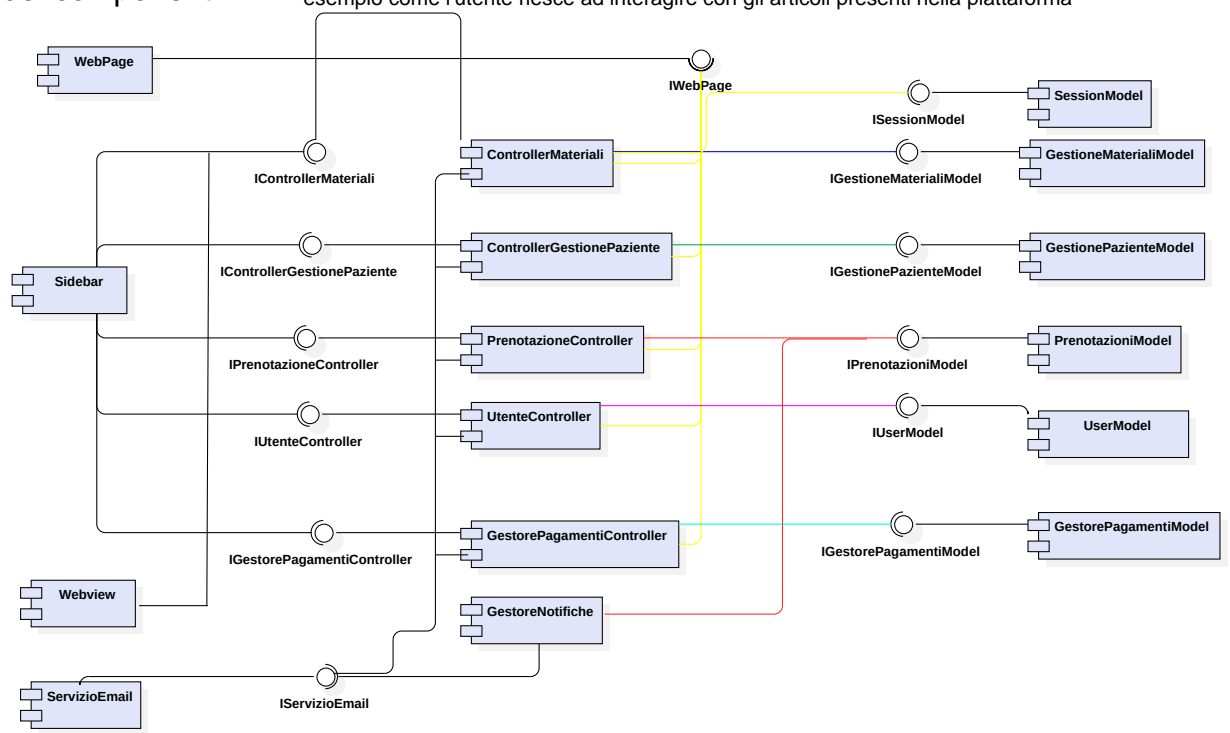


13.3 InvioLinkRegistrazione



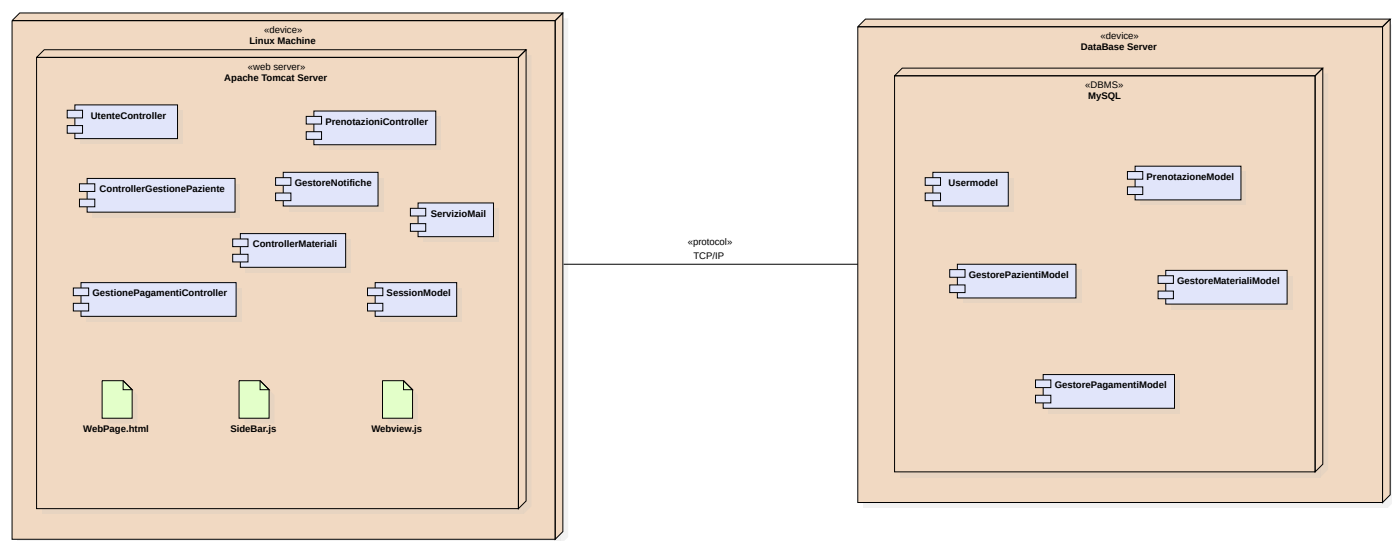
14. Diagramma dei componenti --->

Il diagramma delle componenti mostra come le varie parti del sistema si interfacciano tra di loro, ad esempio come l'utente riesce ad interagire con gli articoli presenti nella piattaforma

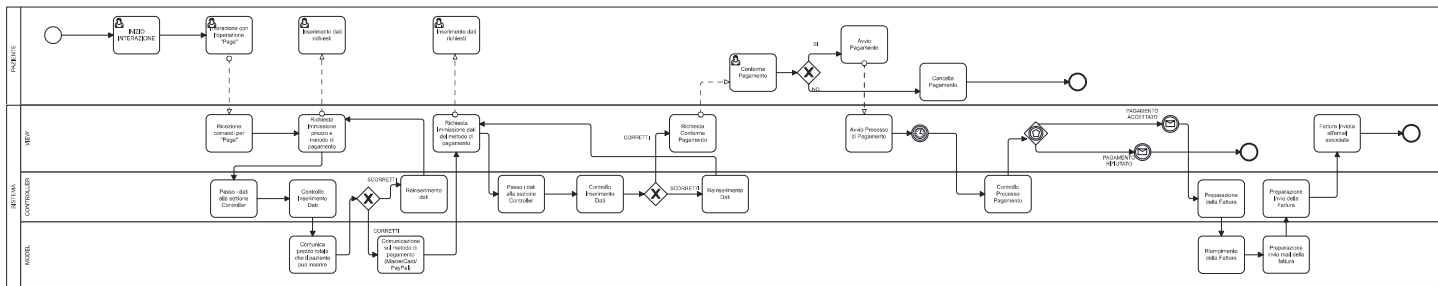


15. Diagramma del deployment--->

mostra l'infrastruttura hardware su cui si poggia il software, e le altre componenti hardware con il quale si può interfacciare e comunicare



aiutano a descrivere un processo all'interno del sistema sotto vari aspetti; infatti il BPMN può essere utilizzato dagli sviluppatori dell'implementazione dei processi o anche da stakeholders che vogliono comprendere i processi che mette in gioco la piattaforma



BPMN per la funzione di pagamento

17. Implementazione

Classe CartellaClinica

```
public class CartellaClinica {
    Calendar DataCaricamento;
    String Anamnesi, Diagnosi, Trattamento, Risultati;
    int id_paziente;

    public CartellaClinica(Calendar string, String anamnesi, String diagnosi, String trattamento, String risultati, int id_paziente) {
        DataCaricamento = string;
        Anamnesi = anamnesi;
        Diagnosi = diagnosi;
        Trattamento = trattamento;
        Risultati = risultati;
        this.id_paziente = id_paziente;
    }

    public Calendar getDataCaricamento() {return DataCaricamento;}
    public String getAnamnesi() {return Anamnesi;}
    public String getDiagnosi() {return Diagnosi;}
    public String getTrattamento() {return Trattamento;}
    public String getRisultati() {return Risultati;}
    public int getId_paziente() {return id_paziente;}

    @Override
    public String toString() {
        return "CartellaClinica [DataCaricamento=" + DataCaricamento + ", Anamnesi=" + Anamnesi + ", Diagnosi="
            + Diagnosi + ", Trattamento=" + Trattamento + ", Risultati=" + Risultati + ", id_paziente="
            + id_paziente + "]\n";
    }
}
```

Classe ModInAttesa

```
public class ModInAttesa {
    private String Json,SourceDB;
    private Calendar data;
    private int IDModifica;

    public ModInAttesa(String json, String sourceDB, Calendar data,int iDModifica) {
        Json = json;
        SourceDB = sourceDB;
        this.data = data;
        IDModifica = iDModifica;
    }
}
```

Classe Paziente extends Utente

```
public class Paziente extends Utente {
    private CartellaClinica CC;
    private ArrayList<Prenotazione> ListaPrenotazioni;
    private ArrayList<Fattura> ListaFatture;
    private int ID_paziente;
    public Paziente(String nome,String cognome,String indirizzo,String email,String numTelefono,String hashPW,int k) {
        super(nome, cognome, indirizzo, email, numTelefono, hashPW, Ruolo.Paziente);
        CC = null;
        ListaPrenotazioni = new ArrayList<Prenotazione>(initialCapacity:0);
        ListaFatture = new ArrayList<Fattura>(initialCapacity:0);
        ID_paziente=k;
    }

    public CartellaClinica getCC(){return CC;}
    public Fattura getF(final int index) {return ListaFatture.get(index);}
    public Prenotazione getP(int index) {return ListaPrenotazioni.get(index);}
    public int getID(){return ID_paziente;}

    /// METODI DI SUPPORTO PER LA SIMULAZIONE DEL DB
    public CartellaClinica setCC(CartellaClinica newCC){return CC=newCC;}
    public CartellaClinica delCC(){return CC=null;}
}
```

Classe GestioneDatiUtenti

```
abstract class GestioneDatiUtenti {
    static ArrayList<Paziente> pazienteList=new ArrayList<Paziente>(initialCapacity:0);

    static void StartInstance(){
        Paziente foo = new Paziente(nome:"Mario",cognome:"Rossi",indirizzo:"Via Rossi",email:"Rossi@gmail.com",numTelefono:"123",hashPW:"HashPW");
        foo.setCC(new CartellaClinica(Calendar.getInstance(),anamnesi:"asd",diagnosi:"asd",trattamento:"asd",risultati:"Ad",foo.getID()));
        Paziente moo = new Paziente(nome:"Carlo",cognome:"Mentone",indirizzo:"Via Neri",email:"Pasquale@alice.eu",numTelefono:"1578",hashPW:"PW");
        moo.setCC(new CartellaClinica(Calendar.getInstance(),anamnesi:"asd",diagnosi:"asd",trattamento:"asd",risultati:"Ad",moo.getID()));
        Paziente roo = new Paziente(nome:"RAmo",cognome:"dicomo",indirizzo:"viale Bianchi",email:"RAmo@gmail",numTelefono:"125478",hashPW:"NOPW");
        roo.setCC(new CartellaClinica(Calendar.getInstance(),anamnesi:"asd",diagnosi:"asd",trattamento:"asd",risultati:"Ad",roo.getID()));
        pazienteList.add(foo);
        pazienteList.add(moo);
        pazienteList.add(roo);
    }

    public static Paziente getPaziente(int id){
        for (Paziente paziente : pazienteList) {
            if(paziente.getID()==id)
                return paziente;
        }
        return null;
    }
}
```

Classe GestionePazienti

```
public class GestionePazienti {
    public Paziente AddCartellaClinica(CartellaClinica c, int IDp){
        Paziente p= GestioneDatiUtente.getPaziente(IDp);
        p.setCC(c);
        return p;
    }

    public int rimuoviCC (int id,Ruolo r) {
        if(r==Ruolo.Dottore){
            Paziente p = GestioneDatiUtente.getPaziente(id);
            p.delCC();
            return 0;
        }
        else if (r==Ruolo.Assistente){
            Gestore_Modifiche.RaccogliModifiche(DBsource:"CartelleCliniche:Eliminazione", (Object)GestioneDatiUtente.getPaziente(id));
            return 1;
        }
        return -1;
    }
    public CartellaClinica getPazienteCC(int i){
        return GestioneDatiUtente.getPaziente(i).getCC();
    }
}
```

Classe Gestore Modifiche

```
import java.util.ArrayList;
import java.util.Calendar;
import com.google.gson.Gson;

public abstract class Gestore_Modifiche {
    static ArrayList<ModInAttesa> list = new ArrayList<ModInAttesa>();

    static public void RaccogliModifiche(String DBsource, Object object){
        Gson gson = new Gson();

        ModInAttesa temp = new ModInAttesa(gson.toJson(object), DBsource, Calendar.getInstance(), iDModifica:0);
        list.add(temp);
    }

    public static int getLen(){return list.size();}
    public static String getString(int index){
        return list.get(index).toString();
    }
}
```

18. Test

I test effettuati riguardano l'aggiunta e la rimozione di una cartella clinica. Nel primo test si può notare l'aggiunta di una cartella clinica che può essere fatta indistintamente sia dal dottore che dall'assistente alla poltrona mentre, nel caso della rimozione (secondo e terzo test), il dottore può effettuarlo senza problemi mentre l'assistente ha bisogno dell'autorizzazione del dottore. Pertanto, nel Test3 andiamo ad effettuare il Test sull'aggiunta di una modifica in attesa nella lista delle modifiche.

```
7      GestionePazienti gestore = new GestionePazienti();
8
9      @Test
10     public void testaddCC() {
11         GestioneDatiUtente.StartInstance();
12         gestore.AddCartellaClinica(CC, IDp:1);
13         assertEquals(CC, gestore.getPazienteCC(i:1));
14     }
15
16     @Test
17     public void testdelCC() {
18         GestioneDatiUtente.StartInstance();
19         assertEquals(0, gestore.rimuoviCC(id:1, Ruolo.Dottore));
20         assertEquals(null, gestore.getPazienteCC(i:1));
21     }
22
23     @Test
24     public void testdel2CC() {
25         GestioneDatiUtente.StartInstance();
26         assertEquals(1, gestore.rimuoviCC(id:1, Ruolo.Assistente));
27         assertEquals(1, Gestore_Modifiche.getLen());
28         System.out.println(Gestore_Modifiche.getString(index:0));
29     }
30 }
```