

De L'Hospital



Studenti:

Gozzo Miriam

Fausto Giovanni

Rondinella Alessia

Prefazione

Il seguente documento presenta una descrizione relativa allo sviluppo dell'applicazione De L'Hospital implementata durante il corso di Ingegneria del Software. Il software è stato implementato nel linguaggio Java sfruttando l'ambiente di sviluppo Eclipse, mentre per la progettazione in UML è stata realizzata mediante il software Astah. Verranno qui presentate solamente le versioni finali di ciascun elaborato ottenute al termine di tutte le fasi di progettazione (per visionare le versioni intermedie si vedano i documenti nelle corrispondenti sottocartelle della documentazione). Infine, vengono descritte le fasi conclusive di testing e di refactoring del programma, con cui sostanzialmente sono stata apportate delle migliorie al codice ed è stato aggiunto un Database per gestire la persistenza dei dati.

Indice

1 Ideazione e analisi dei requisiti	4
1.1 Introduzione.....	4
1.2 Requisiti	4
1.3 Obiettivi e casi d'uso.....	4
1.4 Modello dei casi d'uso	6
UC1: Gestione reparti. CRUD	6
UC2: Gestione pazienti. CRUD	6
UC3: Gestione ricovero. CRUD.....	7
UC4: Gestione servizio. CRUD.....	9
UC5: Gestione sconto. CRUD	10
UC6: Associa sconto.....	10
UC7: Crea Prenotazioni	11
UC8: Consultazione cartella clinica online.....	12
1.5 Documento di visione	12
1.6 Regole di business	12
1.7 Specifiche supplementari	13
1.8 Glossario	14
2 Analisi Orientata agli oggetti	15
2.1 Introduzione.....	15
2.2 Modello di dominio	16
2.3 SSD e Contratti	17
3 Progettazione.....	23
3.1 Diagramma delle classi	23
3.2 Diagramma di sequenza	23
4 Testing.....	29
4.1 Introduzione.....	29
4.2 Individuazione dei casi di test e Testing Unitario	29
4.3 Test di Sistema	30
5 Refactoring e Conclusioni	31
5.1 Database e Refactoring.....	31
5.2 Test di accettazione	31

1 Ideazione e analisi dei requisiti

1.1 Introduzione

La fase di ideazione ha lo scopo di fornire informazioni basilari per la stesura del progetto. Il problema principale che viene trattato riguarda l'analisi degli investimenti, ovvero, si analizzano gli aspetti positivi e negativi al fine di comprendere se vale la pena continuare con la progettazione o meno.

Per analizzare tali aspetti sono stati considerati i seguenti documenti: Modello dei Casi d'Uso, Documento di Visione, Regole di Business, Specifiche Supplementari e Glossario.

1.2 Requisiti

L'Amministrazione del centro ospedaliero richiede la realizzazione di un software che permetta di migliorare le gestioni dei reparti, i ricoveri, i dati dei pazienti, le prenotazioni.

Il software prevede la possibilità di accedere nei panni di segretario ed avere la possibilità di accedere alle operazioni disponibili, che sono diverse. In particolare:

- ✓ Il Segretario deve poter creare un nuovo reparto e aggiungere informazioni riguardanti i costi dei servizi offerti, i posti disponibili.
- ✓ Il Segretario deve poter gestire le informazioni riguardanti i reparti, i relativi servizi offerti e il relativo personale, comprese le modifiche del paziente e le liberazioni/occupazioni dei posti letto;
- ✓ Il Segretario, quando un paziente si reca in clinica, deve poter ricercare il servizio richiesto:
 - a. Se il servizio è disponibile, il paziente può procedere con la richiesta del ricovero;
 - b. Se il servizio non è disponibile, il paziente può lasciare la clinica o richiedere un altro servizio, il segretario può inviare all'amministrazione una richiesta di inserimento del servizio;
- ✓ Nel caso a., dopo essere stato registrato (se non è stato mai registrato), il paziente potrà recarsi in reparto; Il Segretario quindi inserirà le informazioni relative al ricovero nel sistema e il ricovero avrà inizio;
- ✓ Il Segretario deve poter ricercare un paziente all'interno del sistema e modificare le informazioni riguardanti lo stato del paziente, gli interventi effettuati ed eventualmente terminare il ricovero.
- ✓ Il Segretario deve poter creare una o più prenotazioni, in base ai servizi richiesti, per il paziente che vuole effettuare il ricovero.

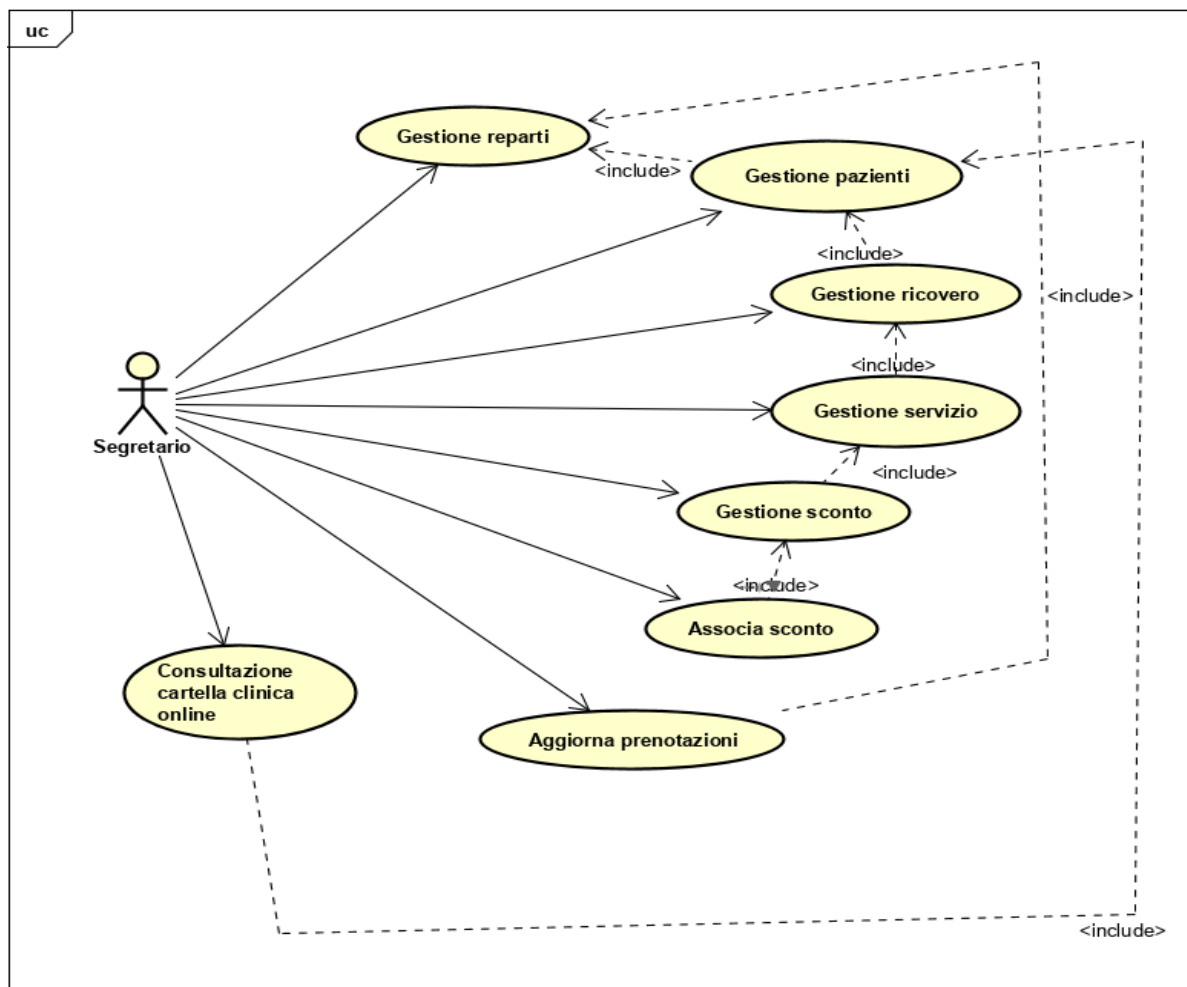
1.3 Obiettivi e casi d'uso

Analizzando i requisiti sono stati individuati gli attori principali e i relativi casi d'uso.

Attore	Obiettivo	Caso d'Uso
Segretario	Gestire le informazioni relative a un reparto	UC1: Gestione reparti
Segretario	Gestire le informazioni relative a un paziente	UC2: Gestione pazienti

Segretario	Inserire servizi di cui vuole usufruire il paziente e aggiornare lo stato del paziente	UC3: Gestione ricovero
Segretario	Gestione nuovi servizi disponibili in uno specifico reparto	UC4: Gestione servizio
Segretario	Gestione di eventuali sconti di cui il paziente ha diritto	UC5: Gestione sconto
Segretario	Ricercare un paziente tramite codice fiscale, per potergli associare uno sconto, se disponibile	UC6: Associa sconto
Segretario	Crea la prenotazioni in base alla disponibilità di posti letto, e in base ai servizi richiesti	UC7: Crea prenotazioni
Segretario	Gestione dati personali online	UC8: Consultazione cartella clinica online

I casi d'uso descritti vengono rappresentati nel seguente schema:



1.4 Modello dei casi d'uso

Si è scelto di fornire una descrizione in formato dettagliato per i casi d'uso UC2: Gestione pazienti e UC3: Gestione ricovero.

Gli altri casi d'uso vengono descritti nel formato breve o informale.

UC1: Gestione reparti. CRUD

1. Il Segretario richiede al sistema **l'inserimento** di un nuovo reparto
2. Il Sistema richiede al segretario le informazioni relative al reparto.
3. Il Segretario inserisce codReparto, medici, infermieri, postiDisponibili e descrizione.

Scenari alternativi

1a. Il segretario richiede al Sistema la **modifica** di un reparto.

1. Il Sistema richiede al segretario di inserire i dati aggiornati.
2. Il Segretario aggiorna i campi di interesse (codReparto, medici, infermieri, postiDisponibili, servizi, descrizione) e conferma.

1b. Il segretario richiede al sistema di **ricercare** un reparto.

1. Il Sistema richiede al segretario di inserire le informazioni da ricercare.
2. Il Segretario inserisce le informazioni a sua disposizione.
3. Il Sistema mostra l'elenco dei reparti che soddisfano i criteri di ricerca.

1c. Il segretario richiede al sistema di **eliminare** un reparto.

1. Il Sistema chiede di confermare l'eliminazione.
2. Il Segretario conferma e il Sistema registra l'avvenuta cancellazione.

UC2: Gestione pazienti. CRUD

Nome del caso d'uso	Registrazione paziente
Portata	Applicazione: De L'Hospital
Livello	Obiettivo paziente
Attore primario	Segretario
Parti interessate e interessi	Segretario: vuole gestire l'intero processo di registrazione
Pre-condizioni	
Garanzia di successo	La registrazione si conclude se i dati sono corretti. L'elenco dei pazienti viene aggiornato
Scenario principale di successo	<ol style="list-style-type: none">1. Il segretario vuole inserire il paziente;2. Il segretario sceglie l'attività "Inserimento paziente";3. Il paziente fornisce i dati;4. Il segretario inserisce CF, nome, cognome, età, percentualeInvalidità e data per registrare il paziente;5. Il Sistema registra le informazioni sul paziente.6. Il Segretario indica di aver finito.

Estensioni	<p>*a. In qualsiasi momento il Sistema fallisce e ha un arresto improvviso</p> <ol style="list-style-type: none"> 1. Il Segretario riavvia il software; 2. Il Sistema ritorna allo stato precedente; <p>1a. Il Segretario richiede al Sistema la modifica di uno stato del paziente.</p> <ol style="list-style-type: none"> 1. Il Sistema richiede al responsabile di inserire i dati aggiornati. 2. Il Segretario aggiorna i campi di interesse e conferma. <p>1b. Il Segretario richiede al sistema di ricercare un paziente esistente.</p> <ol style="list-style-type: none"> 1. Il Sistema richiede il codice fiscale del paziente che intende da cercare. 2. Il Segretario inserisce il codice fiscale e conferma. 3. Il Sistema restituisce il paziente corrispondente. <p>1c. Il Segretario richiede al sistema di eliminare uno stato del paziente.</p> <ol style="list-style-type: none"> 1. Il Sistema chiede di confermare l'eliminazione. 2. Il Segretario conferma e il sistema registra l'avvenuta cancellazione. <p>2a. Il Segretario seleziona l'attività sbagliata</p> <ol style="list-style-type: none"> 1. Il Segretario ritorna alla pagina iniziale e seleziona l'attività inserimento; <p>4a. Il Segretario inserisce informazioni errate</p> <ol style="list-style-type: none"> 2. Il Segretario ricontrolla i dati e li inserisce nuovamente.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	PC
Frequenza di ripetizioni	Legata all'affluenza dei pazienti
Problemi aperti	Se il Segretario non trovasse il servizio, dovrebbe proporre l'inserimento del nuovo servizio?

UC3: Gestione ricovero. CRUD

Nome del caso d'uso	Gestione ricovero
Portata	Applicazione: De L'Hospital
Livello	Obiettivo paziente
Attore primario	Segretario
Parti interessate e interessi	<p>Paziente: usufruisce di uno o più servizi;</p> <p>Segretario: gestisce la cartella clinica comprensiva di eventuali costi;</p>

Pre-condizioni	Il paziente ha richiesto uno o più servizi ed è stato registrato in fase di accettazione;
Garanzia di successo	Il paziente viene curato e regolarmente dimesso;
Scenario principale di successo	<ol style="list-style-type: none"> 1. Un paziente arriva in clinica per iniziare il ricovero. 2. Il Segretario utilizza il software DLH per cercare nell'elenco dei servizi <i>[vedi caso d'uso Gestione Servizio]</i> i servizi di cui necessita il paziente e la disponibilità di posti letto in reparto <i>[vedi caso d'uso Gestione Reparto]</i>. Il sistema mostra l'esito positivo della ricerca, quindi il segretario lo comunica al paziente. 3. Il paziente intende usufruire del servizio. 4. Il Sistema aggiunge il servizio ricercato al preventivo, il Sistema aggiungerà al ricovero il preventivo con i servizi richiesti. <p><i>Ripetere i passi 2,3,4 se ci sono altri servizi di cui il paziente necessita di usufruire.</i></p> <ol style="list-style-type: none"> 5. Il paziente comunica di voler iniziare il ricovero e indica i propri dati al segretario. 6. Il Segretario ricerca il paziente nell'archivio interno <i>[vedi il caso d'uso Gestione Paziente]</i>; il Sistema restituisce il risultato richiesto e lo associa alla prenotazione. 7. Il Segretario conferma l'operazione. 8. Il Sistema calcola il totale del preventivo 9. Il Segretario comunica al paziente l'importo totale e se sono disponibili posti letto in reparto. 10. Il paziente decide proseguire con il ricovero, il Segretario conferma il ricovero e il sistema inoltra alla cassa il preventivo. 11. Il paziente si reca alla cassa, paga e inizia il ricovero in reparto. 12. Il Sistema aggiorna la disponibilità di posti letto in reparto.
Estensioni	<p>*a In qualsiasi momento il Sistema fallisce e ha un arresto improvviso</p> <ol style="list-style-type: none"> 1. Il Segretario riavvia il software; 2. Il Sistema ritorna allo stato precedente; <p>1a. Il paziente sbaglia reparto</p> <ol style="list-style-type: none"> 1. Il Segretario lo indirizza verso il reparto corretto; <p>1b. Il servizio richiesto non è disponibile all'interno della struttura</p> <ol style="list-style-type: none"> 2. Il Segretario comunica al paziente che il servizio non è disponibile; 3. Il Segretario comunica l'assenza del servizio ai suoi superiori;

	<p>6a. Il paziente non è presente nel sistema</p> <ol style="list-style-type: none"> 1. Il Sistema chiede al Segretario di inserire il nuovo paziente nel sistema prima di proseguire. 2. Il Segretario inserisce i dati e conferma [<i>vedi caso d'uso Gestione Paziente</i>]; <p>9a. Il Segretario comunica al paziente che non sono disponibili posti letto al momento.</p> <ol style="list-style-type: none"> 1. Il paziente decide di prenotarsi per la disponibilità futura del posto letto. 2. Il Sistema registra la prenotazione del paziente. 3. Il paziente va via e resta in attesa di una comunicazione da parte dell'ospedale per la notifica della disponibilità di un posto letto.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	PC
Frequenza di ripetizioni	Legata all'affluenza dei pazienti
Problemi aperti	Se non ci fossero posti disponibili, il responsabile di reparto dovrebbe comunicarlo?

UC4: Gestione servizio. CRUD

1. Il Segretario richiede al sistema l'**inserimento** in reparto di un nuovo servizio in reparto.
2. Il Sistema richiede al segretario le informazioni relative al servizio.
3. Il Segretario inserisce codice, medico, tipo, prezzo, e descrizione.

Scenari alternativi

1a. Il Segretario richiede al sistema la **modifica** di un servizio in reparto.

1. Il Sistema richiede al segretario i dati aggiornati.
2. Il Segretario aggiorna i campi di interesse (codice, medico, tipo, prezzo e descrizione) e conferma.

1b. Il Segretario richiede al sistema di **ricercare** un servizio in reparto

1. Il Sistema richiede al libraio di inserire le informazioni nei campi di ricerca (codice, medico, tipo, ecc.).
2. Il Segretario inserisce le informazioni a sua disposizione.
3. Il Sistema mostra l'elenco dei servizi che soddisfano i criteri di ricerca.

1c. Il segretario richiede al sistema di **eliminare** un servizio dal reparto.

1. Il Sistema chiede al libraio di confermare l'eliminazione.
2. Il Segretario conferma.
3. Il Sistema elimina il servizio dal reparto.

UC5: Gestione sconto. CRUD

1. Il Segretario richiede al sistema l'**inserimento** di un nuovo sconto.
2. Il Sistema richiede al Segretario le informazioni relative allo sconto.
3. Il Segretario inserisce codiceSconto, percentuale, descrizione.

Scenari alternativi

1a. Il Segretario richiede al Sistema la **modifica** di uno sconto.

1. Il Sistema richiede al Segretario di inserire i dati aggiornati.
2. Il Segretario aggiorna i campi di interesse (codiceSconto, percentuale, descrizione) e conferma.

1b. Il Segretario richiede al sistema di **ricercare** uno sconto.

1. Il Sistema richiede al segretario di inserire le informazioni da ricercare.
2. Il Segretario inserisce le informazioni a sua disposizione.
3. Il Sistema mostra l'elenco dei reparti che soddisfano i criteri di ricerca.

1c. Il Segretario richiede al sistema di **eliminare** uno sconto.

1. Il Sistema chiede di confermare l'eliminazione.
2. Il Segretario conferma e il sistema registra l'avvenuta cancellazione.

UC6: Associa sconto

Nome del caso d'uso	Associa Sconto
Portata	Applicazione: De L'Hospital
Livello	Obiettivo paziente
Attore primario	Segretario
Parti interessate e interessi	<ul style="list-style-type: none">- Segretario: vuole associare correttamente lo sconto relativo al paziente.- Paziente: desidera ricevere lo sconto in base alle sue esigenze.
Pre-condizioni	Il paziente è stato ricoverato e può usufruire dello sconto.
Garanzia di successo	Assegnazione sconto per il paziente.
Scenario principale di successo	<ol style="list-style-type: none">1. Il Segretario seleziona l'attività "assegna sconto".2. Il Sistema chiede di specificare il codice fiscale relativo al paziente.3. Il Segretario inserisce il codice.4. Il Sistema verifica se ci sono sconti disponibili per il relativo paziente.5. Il Sistema mostra quali sconti sono selezionabili.6. Il Sistema applica lo sconto in automatico. Ripetere i passi 4, 5 e 6 finché ci sono sconti disponibili per quel determinato paziente. <ol style="list-style-type: none">7. Il Sistema provvede all'assegnazione dello sconto al preventivo del paziente.

Estensioni	<p>*a. In qualsiasi momento il Sistema fallisce e ha un arresto improvviso</p> <ol style="list-style-type: none"> 1. Il Segretario riavvia il software; 2. Il Sistema ritorna allo stato precedente; <p>1a. Il Segretario seleziona l'attività sbagliata</p> <ol style="list-style-type: none"> 1. Il Segretario ritorna alla pagina iniziale e seleziona l'attività aggiornamento posti letto disponibili; <p>3a. Il Segretario inserisce informazioni errate</p> <ol style="list-style-type: none"> 1. Il Segretario ricontrolla i dati e li inserisce nuovamente. <p>4a. Il Sistema non trova sconti disponibili.</p> <ol style="list-style-type: none"> 1. Il Segretario ritorna alla pagina iniziale.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	PC
Frequenza di ripetizioni	Legata all'affluenza dei pazienti e alla presenza di prenotazioni
Problemi aperti	Se non è possibile associare lo sconto il segretario dovrebbe comunicarlo?

UC7: Crea Prenotazioni

Nome del caso d'uso	Crea Prenotazioni
Portata	Applicazione: De L'Hospital
Livello	Obiettivo paziente
Attore primario	Segretario
Parti interessate e interessi	<ul style="list-style-type: none"> - Segretario: vuole associare correttamente i posti letto se disponibili alle prenotazioni dei clienti; vuole inoltre informare prontamente i pazienti sull'avvenuta disponibilità di posti letto per iniziare il ricovero se al momento della prenotazione non c'erano posti disponibili. - Paziente: desidera recarsi in ospedale per iniziare il ricovero e quindi fare una prenotazione
Pre-condizioni	Paziente vuole effettuare una prenotazione
Garanzia di successo	Prenotazione effettuata
Scenario principale di successo	<ol style="list-style-type: none"> 8. Il Segretario seleziona l'attività "crea Prenotazione". 9. Il Sistema chiede di specificare il paziente che vuole la prenotazione. 10. Il Segretario inserisce il paziente. 11. Il Sistema chiede i dati per creare il ricovero [<i>vedi il caso d'uso gestione Ricovero</i>]. 12. Il Sistema tramite il paziente e i servizi richiesti può creare la prenotazione, sia che ci siano posti disponibili nei reparti sia che siano tutti occupati.

	13. Il Sistema provvede ad inviare una comunicazione via e-mail al paziente riguardo la disponibilità di un posto letto se al momento della prenotazione non erano disponibili.
Estensioni	<p>*a. In qualsiasi momento il Sistema fallisce e ha un arresto improvviso</p> <p>3. Il Segretario riavvia il software;</p> <p>4. Il Sistema ritorna allo stato precedente;</p> <p>1a. Il Segretario seleziona l'attività sbagliata</p> <p>2. Il Segretario ritorna alla pagina iniziale e seleziona l'attività aggiornamento posti letto disponibili;</p> <p>3a. Il Segretario inserisce informazioni errate</p> <p>2. Il Segretario ricontrolla i dati e li inserisce nuovamente.</p>
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	PC
Frequenza di ripetizioni	Legata all'affluenza dei pazienti e alla presenza di prenotazioni

UC8: Consultazione cartella clinica online

1. Il Segretario richiede al sistema la **consultazione**.
2. Il Segretario inserisce i dati relativi al paziente.
3. Il Sistema mostra la cartella clinica del paziente in questione

1.5 Documento di visione

Il documento di Visione relativo a questo capitolo è presente nell'appendice A. Questo documento è stato redatto nel corso dello sviluppo della fase di ideazione e verrà aggiornato nelle successive interazioni.

1.6 Regole di business

ID	Regola	Modificabilità	Sorgente
R1	<p>Regole di sconto per il paziente:</p> <p>Paziente con invalidità inferiore al 33% – sconto del 20%</p> <p>Paziente con invalidità superiore al 33% – sconto del 40%</p> <p>Paziente con età > 65 – sconto del 20%</p> <p>Paziente con età < 18 – sconto del 20%</p> <p>Paziente che termina</p>	Bassa, queste soglie non possono variare.	Politica interna azienda ospedaliera

	ricovero prima della data prevista – sconto del 20%		
R2	Se non sono presenti posti liberi si crea una prenotazione e in seguito verrà comunicata la disponibilità e in caso un cambiamento di data	Media, le date possono variare.	Politica interna azienda ospedaliera

1.7 Specifiche supplementari

Usabilità

- L'interfaccia grafica deve essere semplice da utilizzare per gli utenti non esperti
- Il testo deve essere visibile da una distanza di almeno 1 metro
- Sono indispensabili caratteristiche quali velocità, facilità ed elaborazione priva di errori

Affidabilità

- In caso di problemi nell'utilizzo del software deve essere necessario eseguire un ripristino in tempi brevi
- Pianificazione di backup periodici del database, al fine di proteggere i dati nel database

Vincoli di implementazione

- Gli sviluppatori hanno scelto una soluzione basata su tecnologie Java, perché ciò garantisce la portabilità a lungo termine, oltre alla facilità di sviluppo
- È necessaria una connessione ad Internet per inviare le comunicazioni ai pazienti

1.8 Glossario

- **Ricovero:** termine riferito a tutto il processo, dall'arrivo del paziente in ospedale fino al pagamento degli interventi e dimissioni
- **Servizio:** termine riferito al tipo di intervento che il paziente deve effettuare.
- **Prenotazione:** prenotazione di un cliente di un servizio qualora non fosse disponibile al momento del suo arrivo
- **Sconto:** indica una riduzione di prezzo che si applica in base a determinati criteri.
- **Preventivo:** contiene i servizi richiesti dal paziente e il relativo costo totale.
- **Paziente:** gli attributi di un paziente da considerare nell'applicazione sono: codice fiscale, nome, cognome, età, percentuale invalidità. Grazie al codice fiscale si evita di aver problemi con pazienti omonimi. Nella tabella sottostante sono indicati i pazienti presenti nel Database.

CF	NOME	COGNOME	ETA	INVALIDITA
1	alessandro	bianchi	25	0
2	andrea	verdi	15	30
3	dario	rossi	15	0
4	caludio	ferrari	75	12
5	giuseppe	marino	25	0
6	pippo	colombo	12	34
7	topolino	bruno	78	89
8	pluto	costa	36	65
9	cesare	villa	15	0
10	gino	grasso	54	78

2 Analisi Orientata agli oggetti

2.1 Introduzione

Seguendo l'approccio iterativo evolutivo suggerito da UP, la realizzazione dell'applicazione è stata articolata su quattro iterazioni. In questo modo è stato possibile implementare in maniera iterativa il nucleo dell'architettura del software, sono state risolte problematiche relative ai rischi maggiori ed è stata affrontata un'analisi dei requisiti graduale in modo da limitare al minimo il danno causato da eventuali errori di progettazione e implementazione.

Per ciascuna iterazione in particolare si ci è occupati di gestire le seguenti problematiche

➤ **Iterazione 1**

- Implementazione scenario principale UC3: Gestione ricovero, senza tener conto, per semplicità, l'associazione di uno sconto al preventivo e la disponibilità dei posti letto
- Implementazione di un caso d'uso Start Up necessario per inizializzare il sistema per poter gestire le esigenze della prima iterazione.

➤ **Iterazione 2**

- Implementazione scenario alternativo 9a del caso d'uso UC3: Gestione Ricovero in cui si prevede che non siano disponibili i posti letto per poter fare il ricovero, ma che comunque il paziente effettui la prenotazione.
- Implementazione del caso d'uso UC7: Crea Prenotazione in quanto completa lo scenario alternativo 9a. Infatti, sarà possibile creare una prenotazione anche se non sono disponibili i posti letto e successivamente verrà aggiornata quando i posti saranno disponibili

➤ **Iterazione 3**

- Implementazione del caso d'uso UC6: Associa Sconto. Ora il segretario può associare uno sconto al ricovero in base agli sconti disponibili per un determinato paziente. Gli sconti sono stati creati precedentemente dal sistema e poi associati.

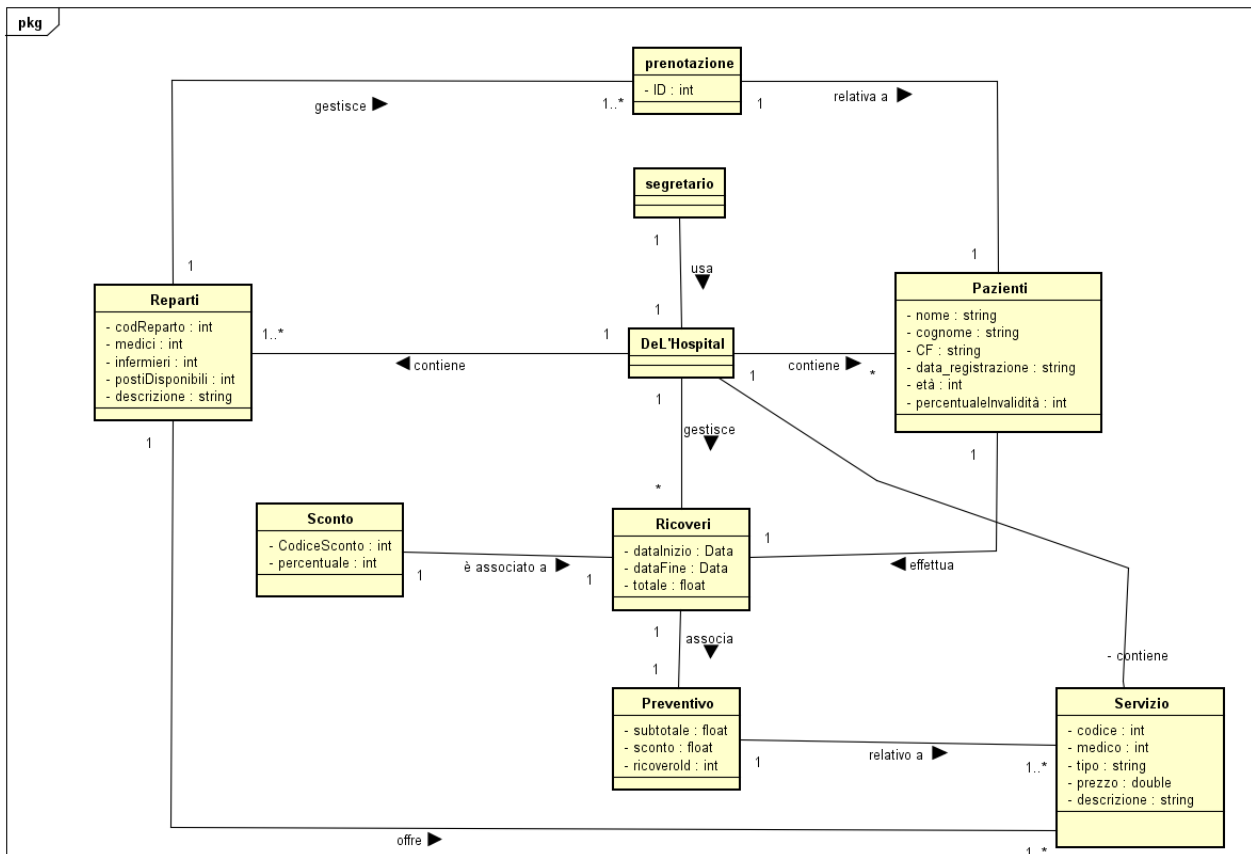
➤ **Iterazione 4**

- Implementazione dello scenario alternativo 6a del caso d'uso UC3: Gestione Ricovero e del relativo caso d'uso UC2: Gestione Pazienti, il cui scopo è quello di inserire, modificare, eliminare un paziente dal Database presente nel sistema.

Il passo iniziale di ciascuna iterazione è stato quello di effettuare un'analisi dei requisiti orientata agli oggetti, cioè un'analisi basata sulla creazione di una descrizione del dominio da un punto di vista degli oggetti. Per fornire tale descrizione sono stati usati diversi strumenti: Modello di dominio, SSD (Sequence System Diagram) e contratti delle operazioni.

2.2 Modello di dominio

La disciplina che in termini di UP si occupa di fornire dettagli sul dominio è la modellazione del Business, in particolare essa comprende la stesura del Modello di Dominio, elaborato grafico in cui vengono identificati i concetti, gli attributi e le associazioni considerati significativi. Tenendo conto del contributo dato da ciascuna iterazione, il modello di dominio è dato da:



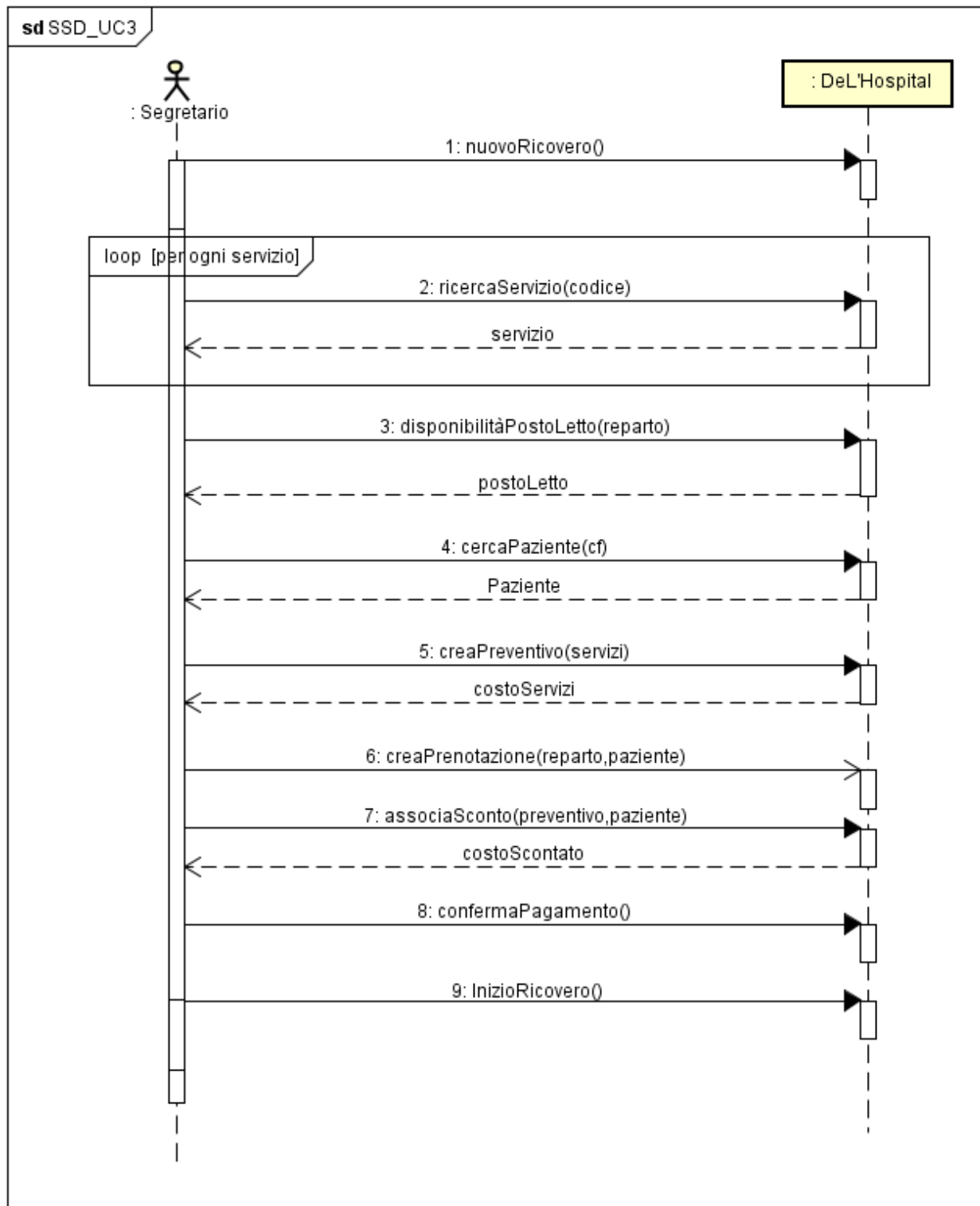
Per un maggiore dettagli si veda l'allegato **Modello di dominio completo**. Come si nota sono identificate le seguenti classi concettuali:

- **Segretario**: rappresenta l'attore principale che interagisce col sistema;
- **DLH**: rappresenta il sistema;
- **Ricoveri**: rappresenta la lista di ricoveri che possono essere effettuati;
- **Preventivo**: contiene i dettagli relativi al pagamento di un particolare ricovero;
- **Servizio**: rappresenta il servizio relativo al preventivo;
- **Pazienti**: paziente che deve effettuare il ricovero;
- **Reparti**: dove il paziente effettua il ricovero;
- **Sconto**: rappresenta lo sconto che verrà associato al preventivo
- **Prenotazione**: rappresenta la prenotazione del paziente per un determinato reparto

2.3 SSD e Contratti

➤ Iterazione 1:

Scenario principale del caso d'uso UC3:



Contratto CO1: nuovoRicovero()

Operazione: nuovoRicovero()
Riferimenti: caso d'uso: Gestisci ricovero
Pre-condizioni: - un paziente vuole usufruire di un servizio
Post-condizioni: - Viene creata una nuova istanza nuovoRicovero di ricovero
- gli attributi di nuovoRicovero sono stati inizializzati
- il nuovoRicovero viene aggiunto all'ArrayList ricoveri
- il ricovero nuovoRicovero è stato associato a del'Hospital tramite l'associazione "gestisce"

Contratto CO2: ricercaServizio(codice)

Operazione: ricercaServizio(codice : integer)
Riferimenti: Caso d'uso: Gestisci ricovero
Pre-condizioni : è in corso un Ricovero *nuovoRicovero*
Post-condizioni: - è noto se il *servizio* ricercato viene offerto dall'azienda ospedaliera

Contratto CO3: disponibilitàPostiLetto(reparto)

Operazione: verificaPostiDisponibili(reparto)
Riferimenti: Caso d'uso: Gestisci ricovero
Pre-condizioni: è in corso un ricovero *nuovoRicovero*
Post-condizioni: -è noto se sono presenti posti liberi

Contratto CO4: cercaPaziente(CF)

Operazione: ricercaPaziente(CF : string)
Riferimenti: Caso d'uso: Gestisci ricovero
Pre-condizioni: - è in corso un ricovero *nuovoRicovero*
Post-condizioni: - è noto se il paziente è già presente nella lista paziente

Contratto CO5: creaPreventivo(servizi)

Operazione: creaPreventivo (servizi)
Riferimenti: Caso d'uso: Gestisci ricovero
Pre-condizioni: è in corso un ricovero *nuovoRicovero*
Post-condizioni: è stato creato un nuovo preventivo che contiene il costo totale dei servizi

Contratto CO6: creaPrenotazione(reparto,paziente)

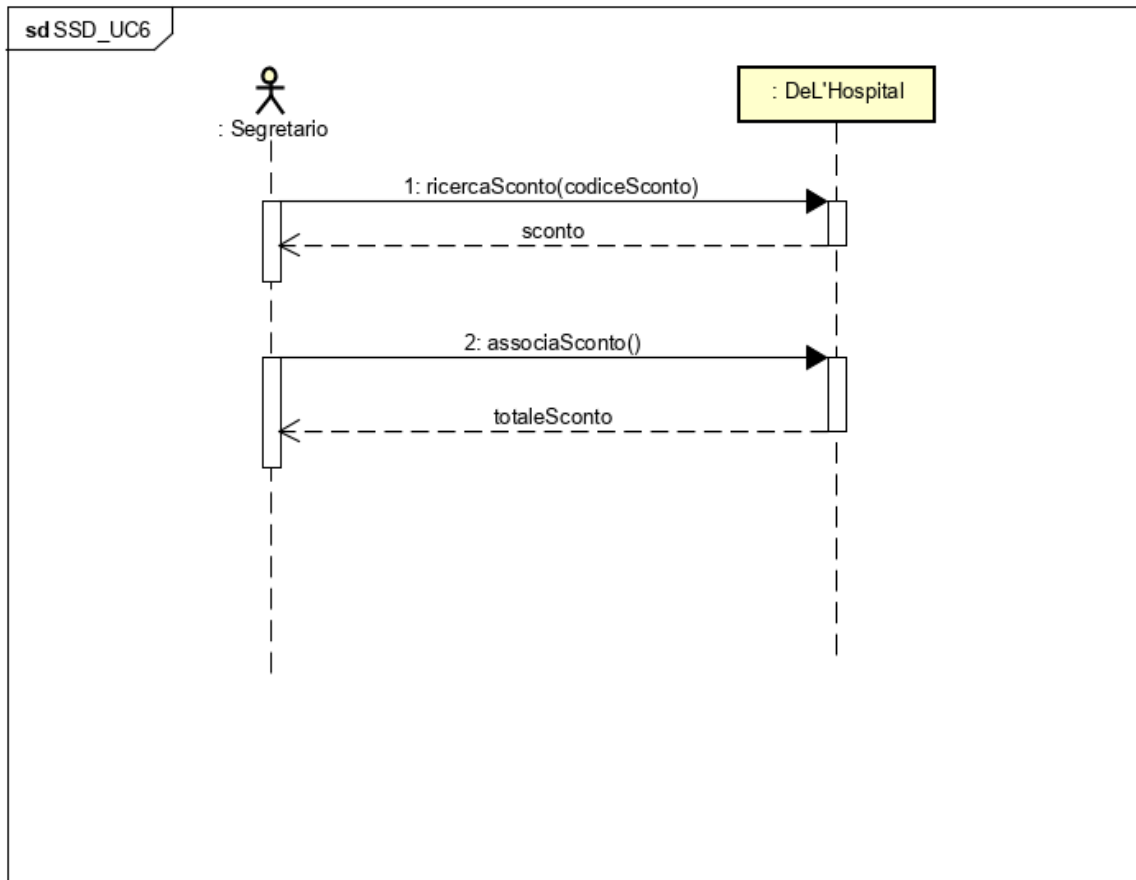
Operazione: creaPrenotazione(reparto,paziente)
Riferimenti: Caso d'uso: Gestisci ricovero
Pre-condizioni: è in corso un ricovero *nuovoRicovero*
Post-condizioni: stata inizializzata una nuova istanza *prenotazione* legata al *paziente* e al *reaprt*

Contratto CO7: associaSconto(preventivo,paziente)

Operazione: associaSconto(preventivo,paziente)
Riferimenti: Caso d'uso: Gestisci ricovero
Pre-condizioni: è in corso un ricovero *nuovoRicovero*
Post-condizioni: è stato associato lo sconto da applicare al totale

➤ Iterazione 3

Scenario principale del caso d'uso UC6:



Contratto CO1: ricercaSconto(codiceSconto)

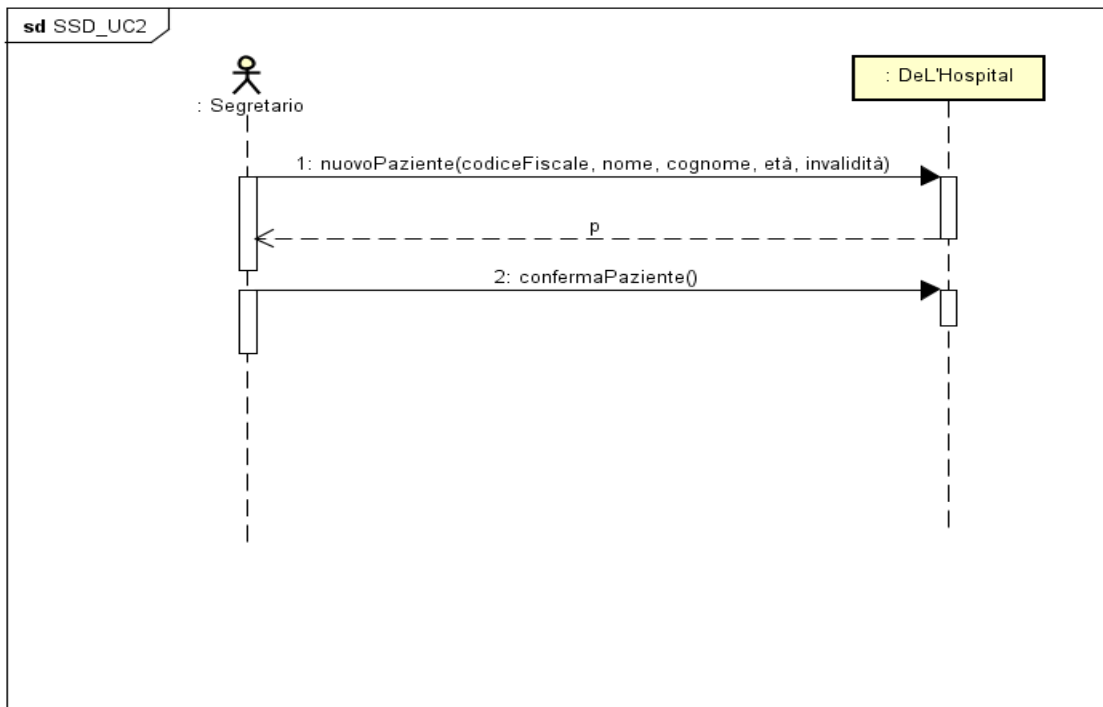
Operazione: ricercaSconto(codiceSconto)
Riferimenti: Caso d'uso: Associa sconto
Pre-condizioni: - è presente un preventivo al quale applicare uno sconto
Post-condizioni: - La classe sconto viene inizializzata

Contratto CO2: associaSconto()

Operazione: associaSconto()
Riferimenti: Caso d'uso: Associa sconto
Pre-condizioni: - è stato trovato uno sconto da associare al preventivo
Post-condizioni: - Lo sconto è stato associato al preventivo del cliente

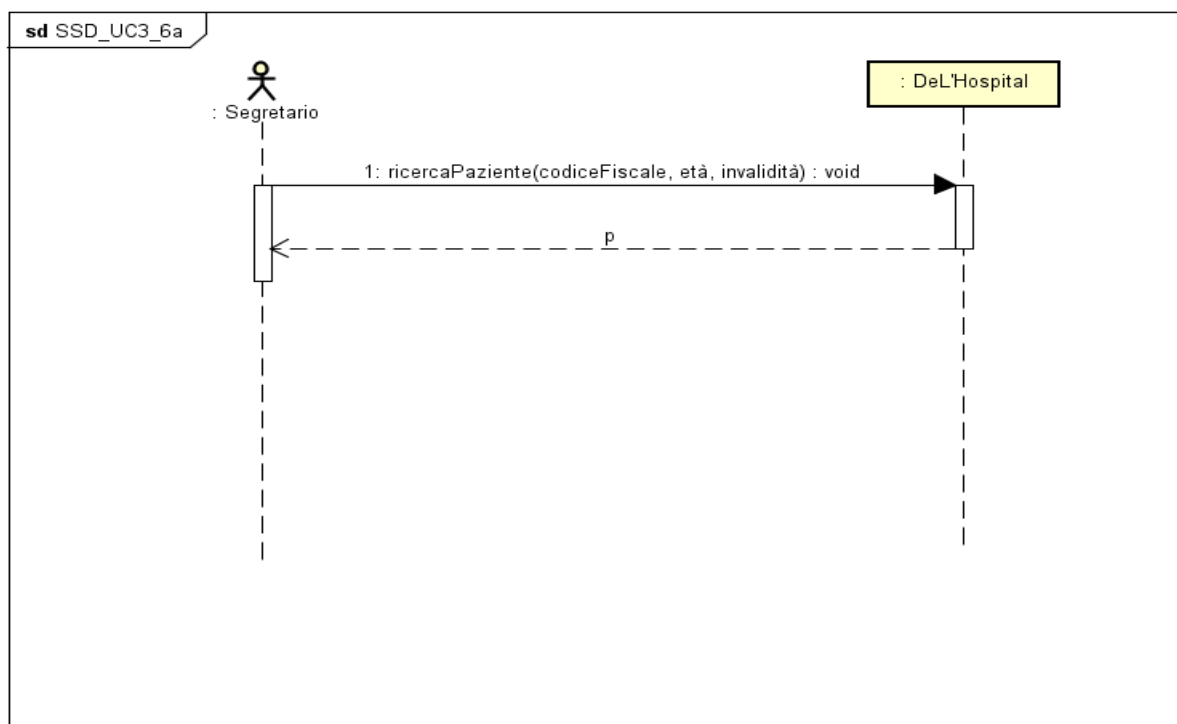
➤ Iterazione 4

Il diagramma di sequenza di sistema relativo al caso d'uso UC2 sarà il seguente:



Si è preso in considerazione lo scenario principale di inserimento di un paziente.

Per quanto riguarda lo scenario alternativo 6a avremo:



Contratto CO1: novoPaziente(codiceFiscale,nome,cognome,età,percentualeInvalidità)

Operazione: novoPaziente(codiceFiscale,nome,cognome,percentualeInvalidità)
Riferimenti: Caso d'uso: Gestione Paziente
Pre-condizioni: - è presente un paziente da registrare
Post-condizioni: - Il paziente è stato registrato

Contratto CO2: ricercaPaziente(codiceFiscale,età,percentualeInvalidità)

Operazione: ricercaPaziente(codiceFiscale,età,percentualeInvalidità)
Riferimenti: Caso d'uso: Gestione Paziente
Pre-condizioni: - c'è un paziente da ricercare
Post-condizioni: - Il paziente è stato trovato

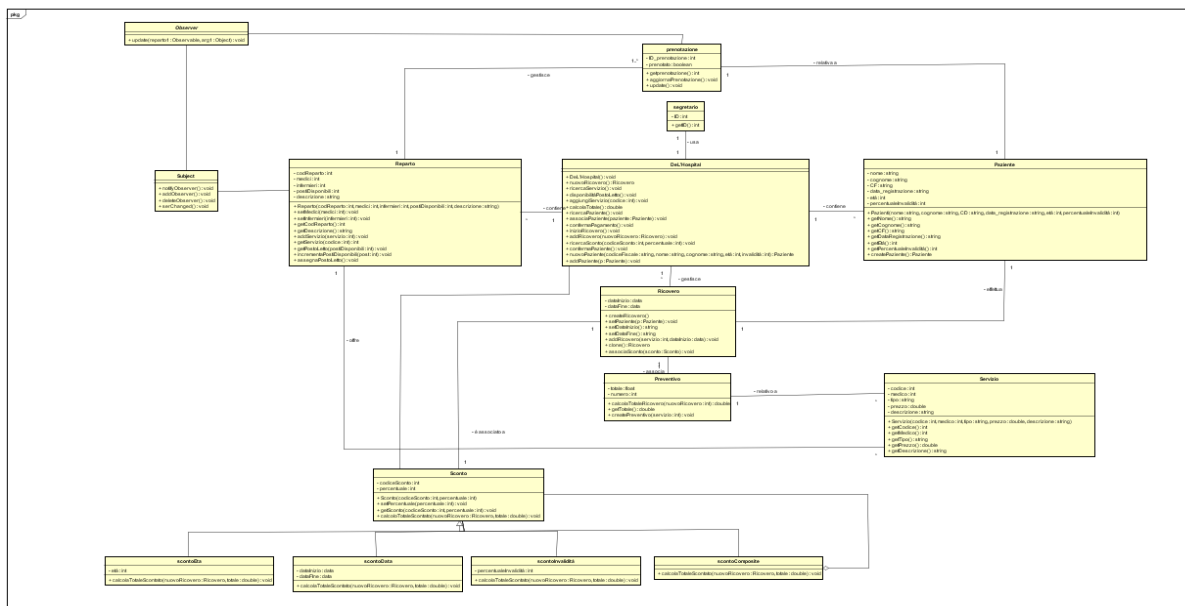
3 Progettazione

3.1 Diagramma delle classi

La progettazione orientata agli oggetti è la disciplina di UP interessata alla definizione degli oggetti software, delle loro responsabilità e a come questi collaborano per soddisfare i requisiti individuati nei passi precedenti.

L'elaborato principale in questa fase che è stato preso in considerazione è il Modello di Progetto, ovvero l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Iterazione) che da un punto di vista statico (Diagramma delle Classi). Come si può notare dal diagramma delle classi, si è fatto uso dei pattern Observer (che in questo caso fa in modo che il reparto comunichi alla prenotazione la disponibilità di posti liberi) e Composite (dato che abbiamo diversi tipi di sconti da poter applicare).

Il diagramma delle classi sarà:

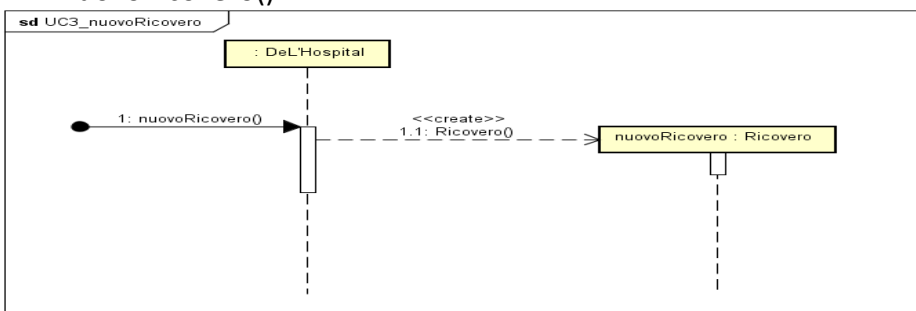


Per un maggiore dettaglio si veda l'allegato **Diagrammi delle classi completo**

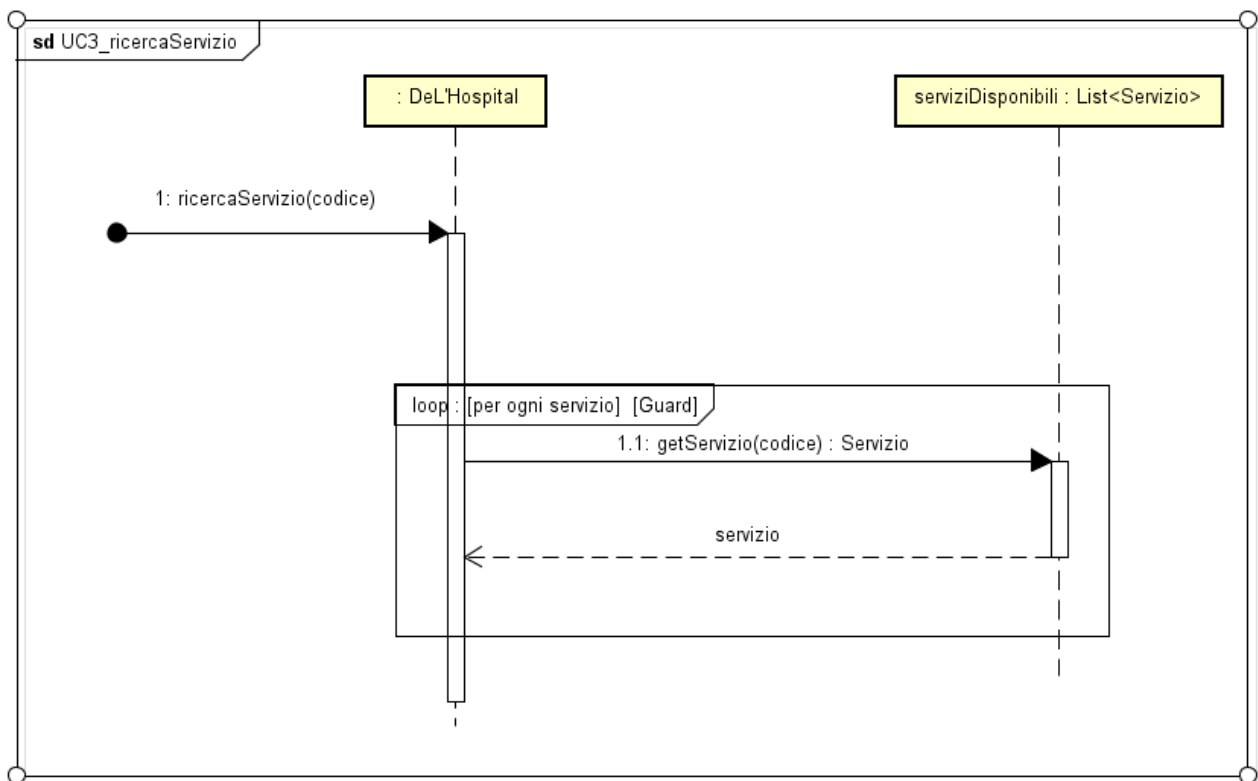
3.2 Diagramma di sequenza

Vengono ora presentati i diagrammi di sequenza che sono stati creati nelle varie iterazioni:

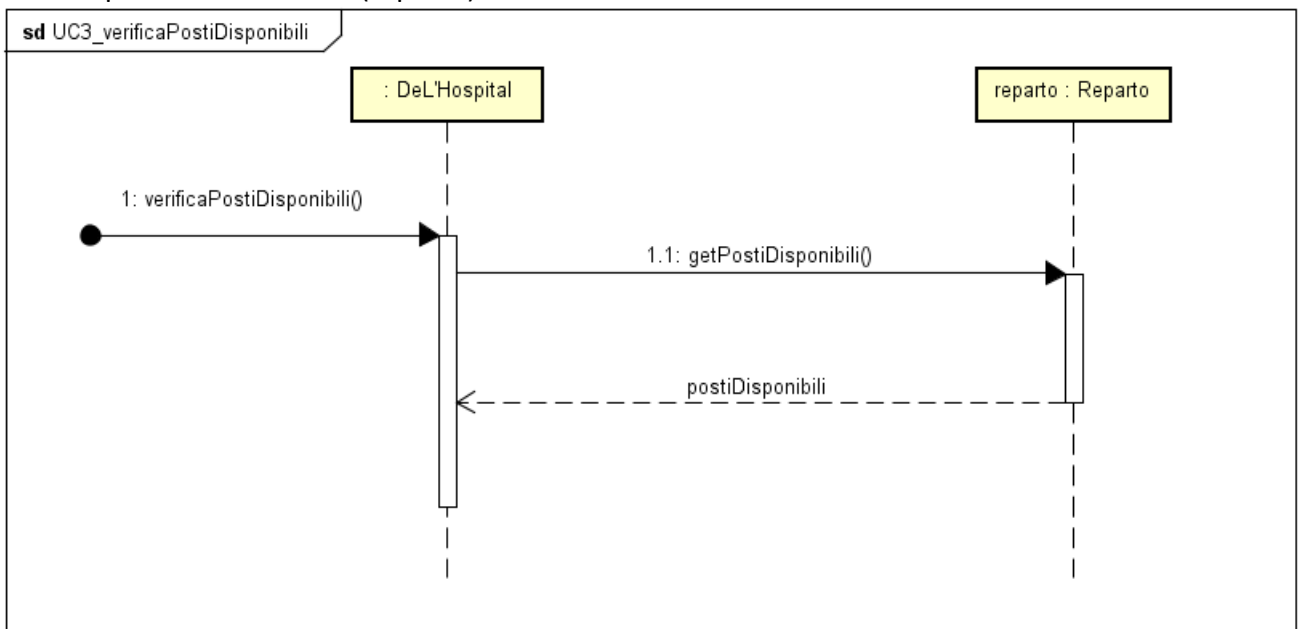
- nuovoRicovero()



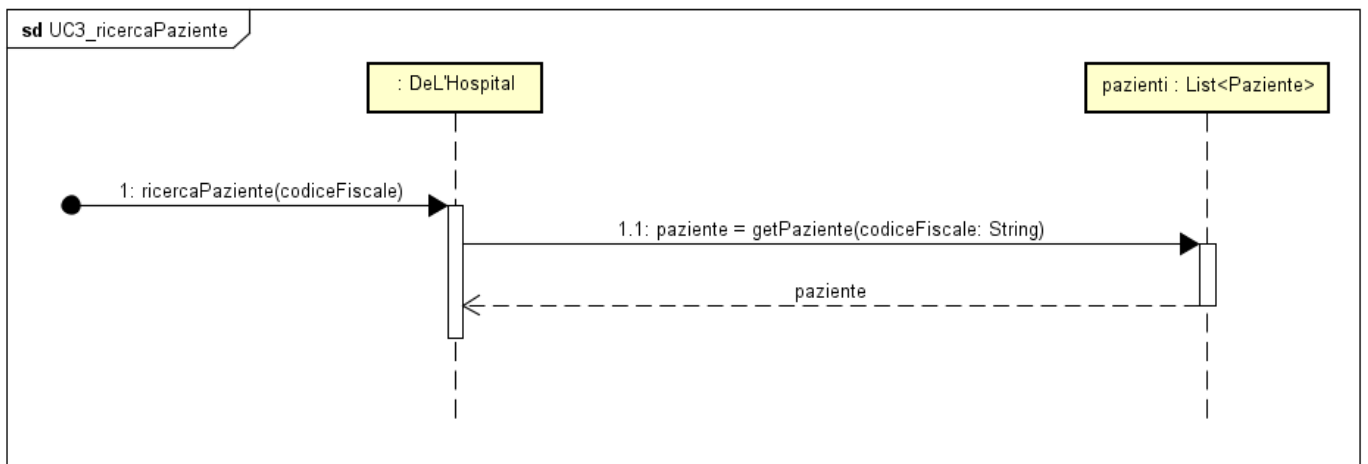
- ricercaServizio(codice: Integer)



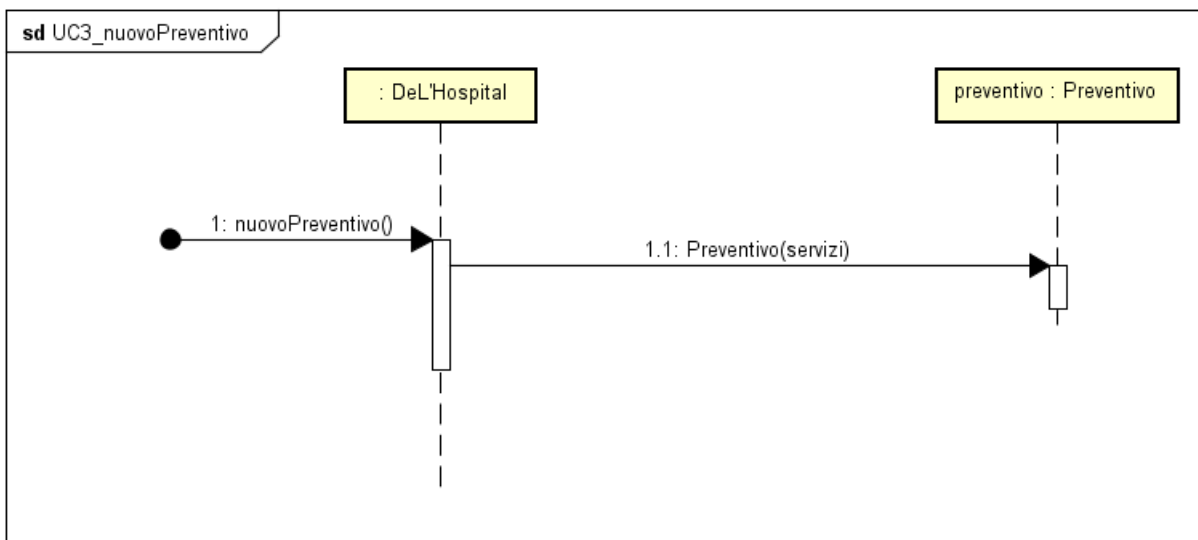
- disponibilitaPostiLetto(reparto)



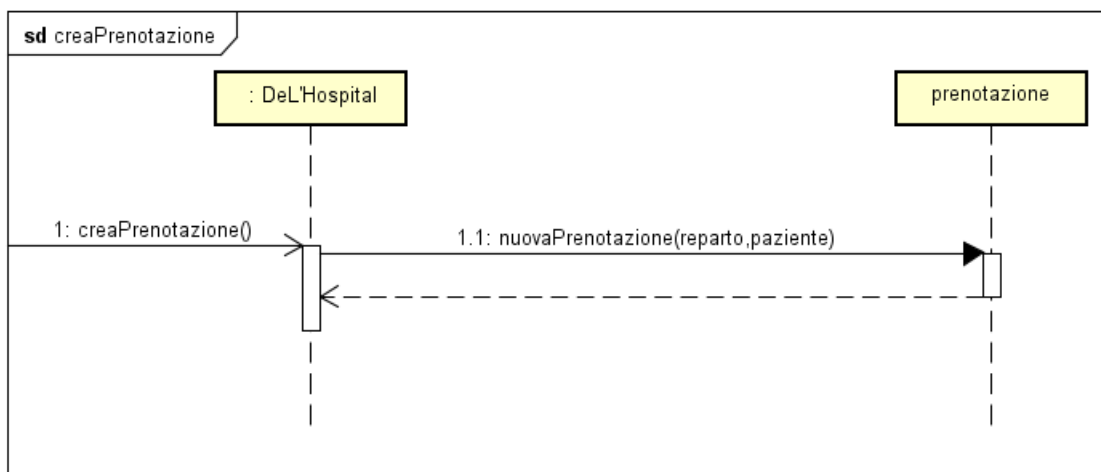
- ricercaPaziente(CF)



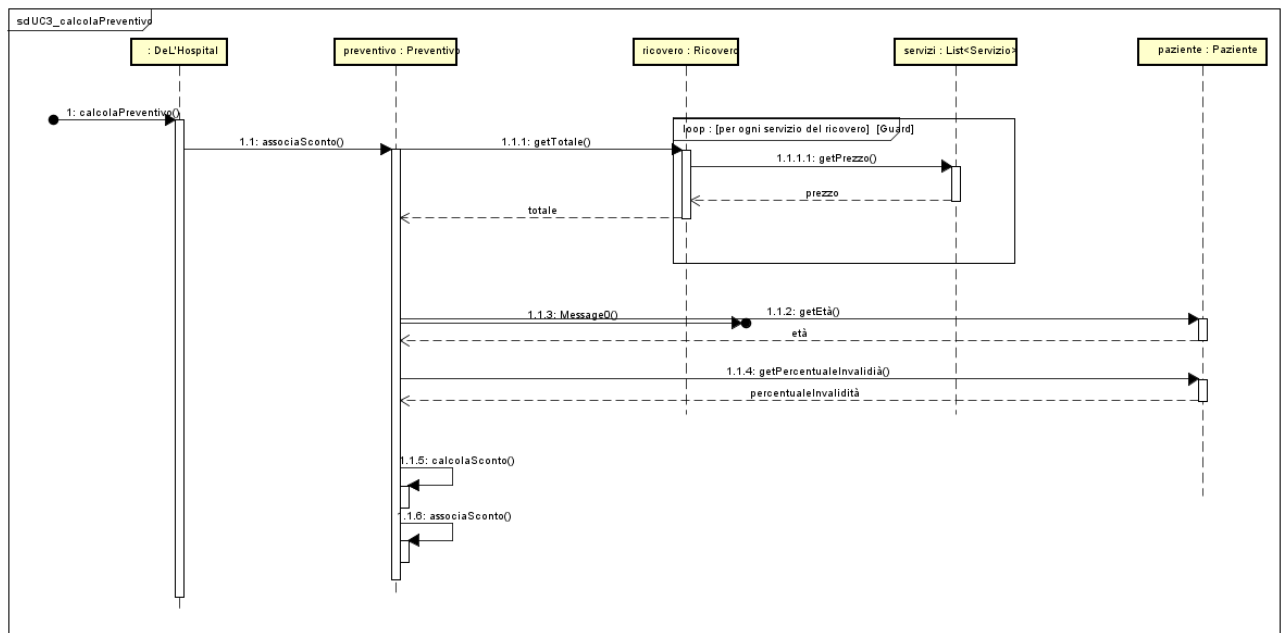
- creaPreventivo()



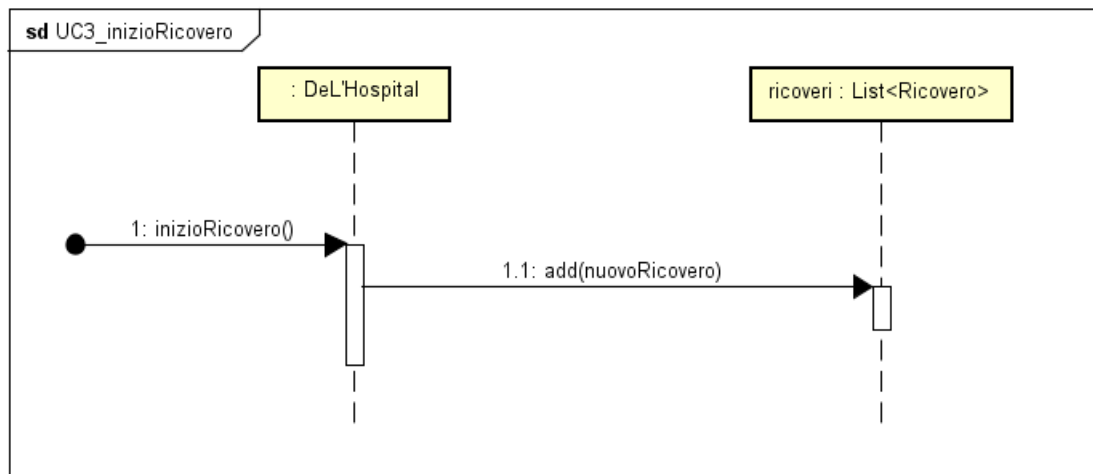
- creaPrenotazione(reparto,paziente)



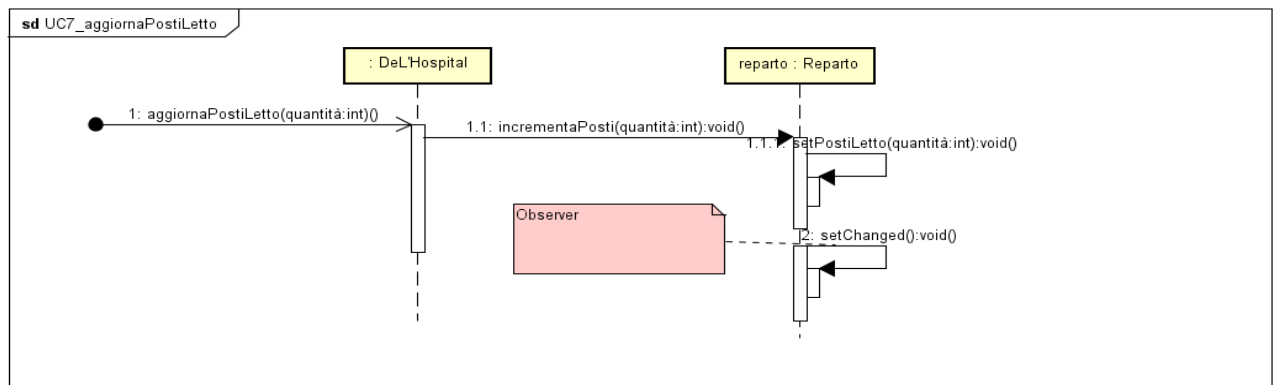
- associaSconto()



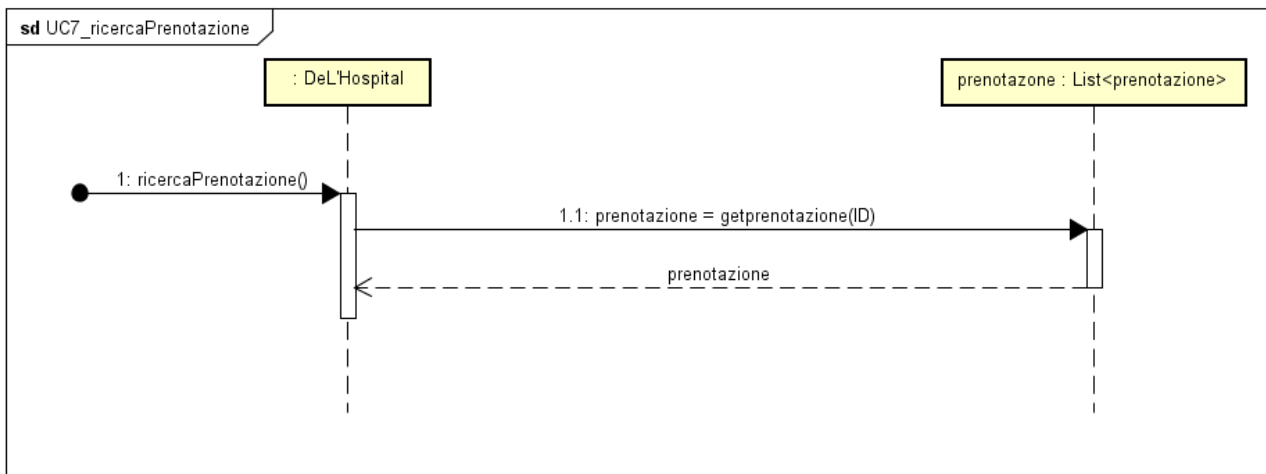
- inizioRicovero()



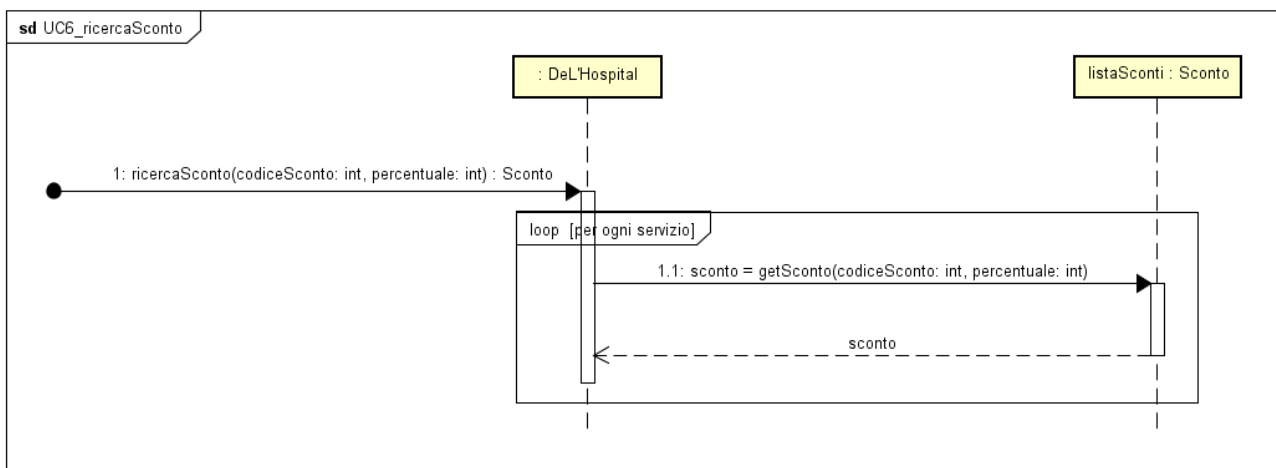
- AggiornaPostiLetto():



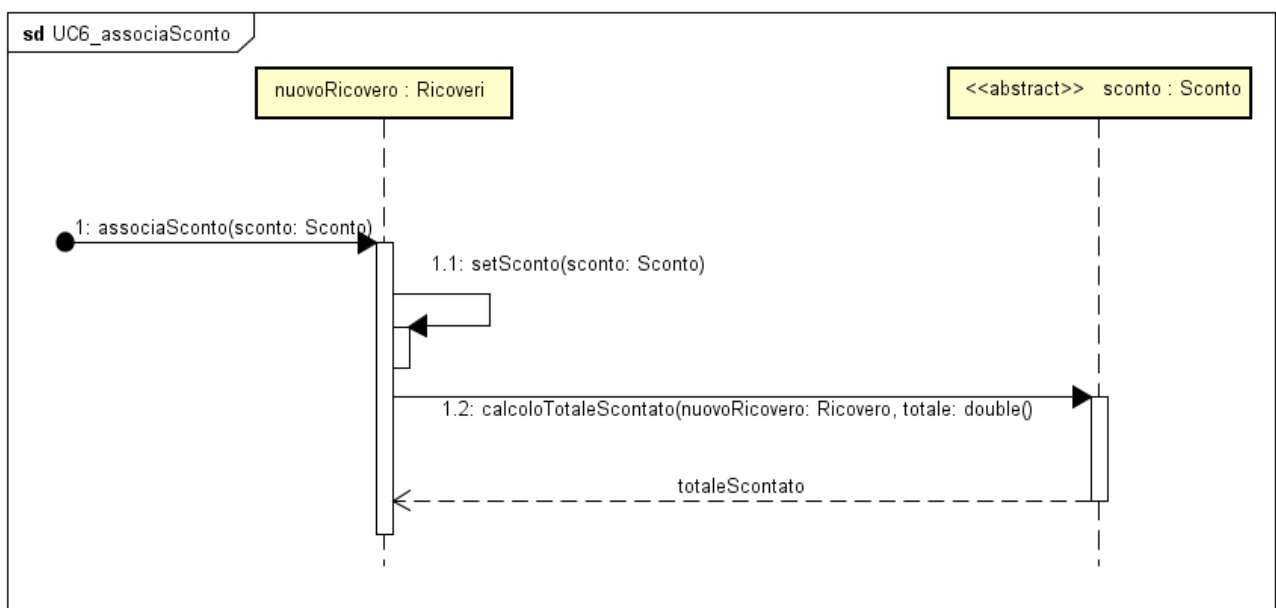
- ricercaPrenotazione():



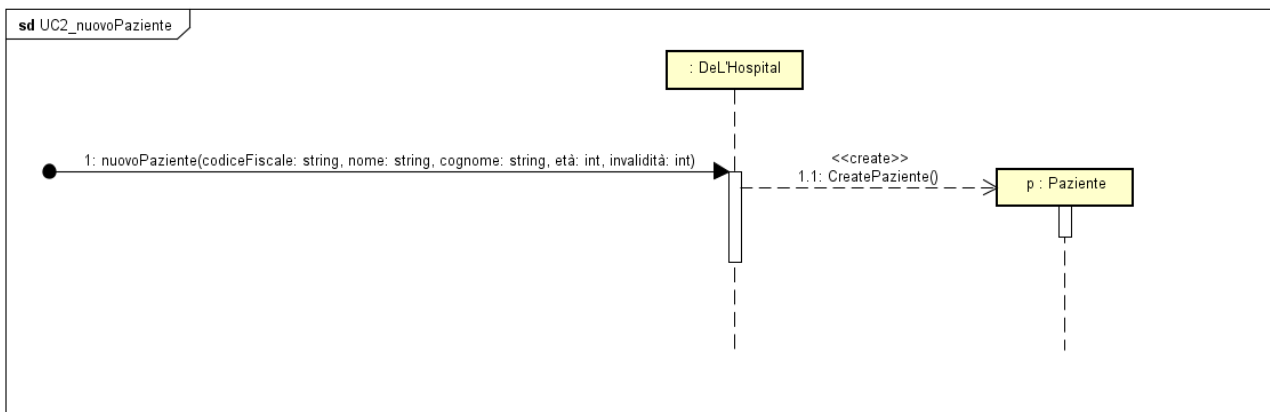
- ricercaSconto



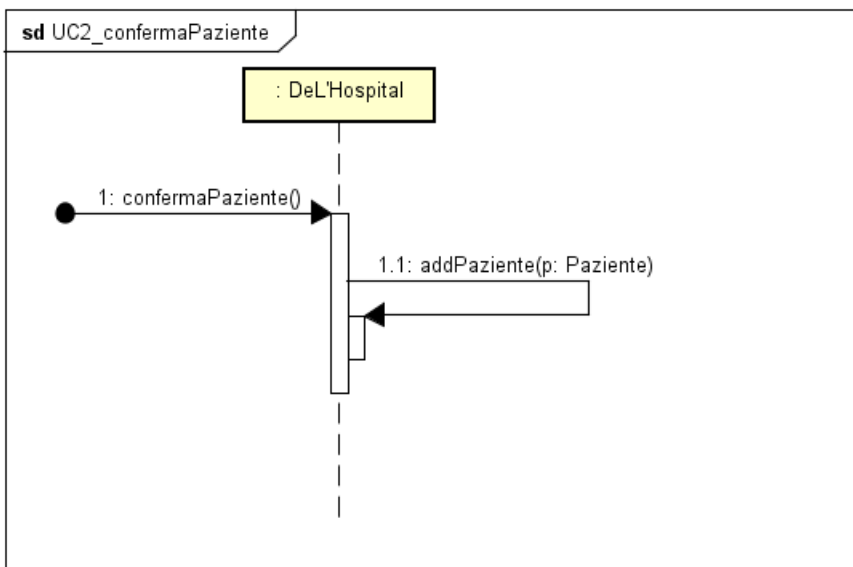
- associaSconto



- nuovoPaziente



- confermaPaziente



4 Testing

4.1 Introduzione

Il **testing** è una fase essenziale del processo di sviluppo di un programma robusto ed efficiente. Permette inoltre di ridurre in modo sostanziale i costi di manutenzione di un applicazione. La probabilità di rilevare malfunzionamenti è proporzionale al numero di test eseguiti, tuttavia è difficile testare un programma completamente. Infatti, le possibili combinazioni di valori di input da prendere in considerazione sono enormi, e non sempre possono essere riprodotte in un tempo ragionevole. Tuttavia, un testing progettato accuratamente può rivelare comportamenti anomali ed indesiderati al fine di rendere il software realizzato funzionante e funzionale secondo le specifiche del committente.

I test si dividono principalmente in due famiglie:

I **test unitari (Unit Test)** → Sono semplici test/prove che vanno a verificare la correttezza direttamente del codice, in ogni sua piccola parte. L'idea dello Unit Test in Java è quella di valutare ogni singolo metodo in funzione dei valori attesi. Esistono diversi tool per Unit Testing per i diversi linguaggi di programmazione, in particolare **JUnit** è il framework più diffuso attualmente per l'automazione del testing di unità di programmi Java.

I **test funzionali** → Sono dei test che vanno a verificare che il sistema software nella sua completezza funzioni correttamente. Questi test trattano il sistema come se fosse una scatola nera alla quale danno degli input e verificano la correttezza degli output.

Per eseguire il testing dell'applicazione realizzata, si è scelto di concentrarsi principalmente su test unitari eseguiti tramite l'ausilio di JUnit.

4.2 Individuazione dei casi di test e Testing Unitario

In una fase preliminare al testing, è stata affrontata una discussione e ispezione del codice sorgente scritto al fine di operare una scelta sulle classi e sui metodi da testare. Il criterio scelto per individuare tali metodi su cui concentrare i test di unità è stato quello di dare priorità a tutti i metodi che si occupano dell'inserimento di nuove istanze, dell'aggiornamento dei posti letto dei reparti e del calcolo dei fattori economici quali: prezzo totale, ed eventuali sconti. Per ciascuno sono state individuate le **classi di equivalenza** tali da poter ridurre il numero di test al minimo insieme necessario, in particolare:

- Prenotazione
 - aggiornaPrenotazione
 - caso normale con posti disponibili e quindi viene aggiornata la prenotazione
 - caso dove non ci sono posti disponibili e quindi non si può assegnare il reparto
 - caso dove inizialmente non ci sono posti disponibili e successivamente incrementati con aggiornamento della prenotazione in automatico
- Reparto
 - incrementaPostiLetto
 - caso dove si incrementa con numero positivo
 - caso dove si incrementa con un numero negativo, ma in questo caso non viene incrementato
 - assegnaPostoLetto
 - caso dove si assegna un posto al reparto se disponibile
- Preventivo

- calcolaTotale
 - caso normale con valori positivi
 - caso con valori negativi e quindi non vengono considerati
- Ricovero
 - associaSconto
 - sono stati testati diverse combinazioni di sconti, che variano in base alle caratteristiche del paziente, per verificare il giusto funzionamento.

A seguito dell'esecuzione di questi test unitari con JUnit, si è passati a correggere le relative porzioni di codice che portavano al fallimento del sistema di alcuni test.

4.3 Test di Sistema

In previsione di una fase di refactoring in cui verrà integrato un Database con l'applicazione esistente, si è scelto di eseguire dei test di sistema manuali lanciando l'applicazione e provando a portare a termine i principali casi d'uso. A seguito di tali test non sono emersi grandi errori nel funzionamento e le prestazioni del sistema si sono mantenute piuttosto elevate, tuttavia sono state rilevate delle imperfezioni e anomalie nell'esecuzione di alcune funzionalità:

- Selezionando i servizi per un ricovero era possibile selezionare più volte lo stesso servizio
- Selezionando il paziente per associare lo sconto era possibile scegliere più volte lo stesso paziente e quindi assegnare più sconti del dovuto
- Terminato un ricovero non veniva liberato il posto letto precedentemente occupato

5 Refactoring e Conclusioni

5.1 Database e Refactoring

In questa fase conclusiva, le operazioni prevalentemente svolte sono state:

- la correzione delle anomalie individuate con il precedente Test di Sistema;
- l'integrazione nel software di un database per gestire la persistenza dei dati. In particolare, il database utilizzato è un DB locale usando phpMyAdmin per creare la tabella Pazienti. L'interfacciamento del DB con l'applicazione è stato realizzato con la libreria Hibernate ORM.
- aggiunta di alcune funzionalità secondarie dell'applicazione, quali la visualizzazione e il salvataggio in locale delle componenti presenti nel sistema, quali: ricoveri, prenotazioni, reparti, sconti, servizi.

5.2 Test di accettazione

A conclusione delle attività necessarie per portare a completamento i passi precedenti, e dopo un'attenta revisione di tutto il codice sorgente, è stato eseguito il test di accettazione conclusivo. Simulando di ritrovarsi nei panni dell'utente (segretario), sono state testate le principali funzionalità del programma al fine di dimostrare che è possibile portare a termine con successo i casi d'uso implementati.