

Corso di Laurea Magistrale in Ingegneria Informatica
Ingegneria del Software

Progetto finale: AnaLab

Fase di Elaborazione
Iterazione 2

Giulia Silvestro

Contents

1	Introduzione	2
2	Caso d'uso UC1: Registra prenotazione	3
2.1	Analisi	3
2.1.1	Modello di Dominio	3
2.1.2	Diagramma di sequenza di sistema	4
2.1.3	Contratti delle operazioni	5
2.2	Progettazione	8
2.2.1	Diagrammi di Interazione	8
2.2.2	Diagramma delle Classi	12
3	Caso d'uso UC8: Visualizza prenotazioni di una data	13
3.1	Analisi	13
3.1.1	Formato Dettagliato	13
3.1.2	Modello di Dominio	14
3.1.3	Diagramma di sequenza di sistema	14
3.1.4	Contratti delle operazioni	14
3.2	Progettazione	14
3.2.1	Diagrammi di Interazione	14
3.2.2	Diagramma delle Classi	15
4	Implementazione	16
4.1	Testing	16
4.1.1	Prenotazione	16

1 Introduzione

Conclusa la prima iterazione, si passa alla seconda. Durante questa iterazione si è deciso di concentrarsi sulle seguenti attività:

- Progettazione dell'UC1: Registra prenotazione,
- Progettazione dell'UC8: Visualizza prenotazioni di una data
- Ampliamento alle nuove classi del caso d'uso di avviamento, con la creazione di istanze delle classi identificate,
- Mantenere i dati momentaneamente solo in memoria principale.

2 Caso d'uso UC1: Registra prenotazione

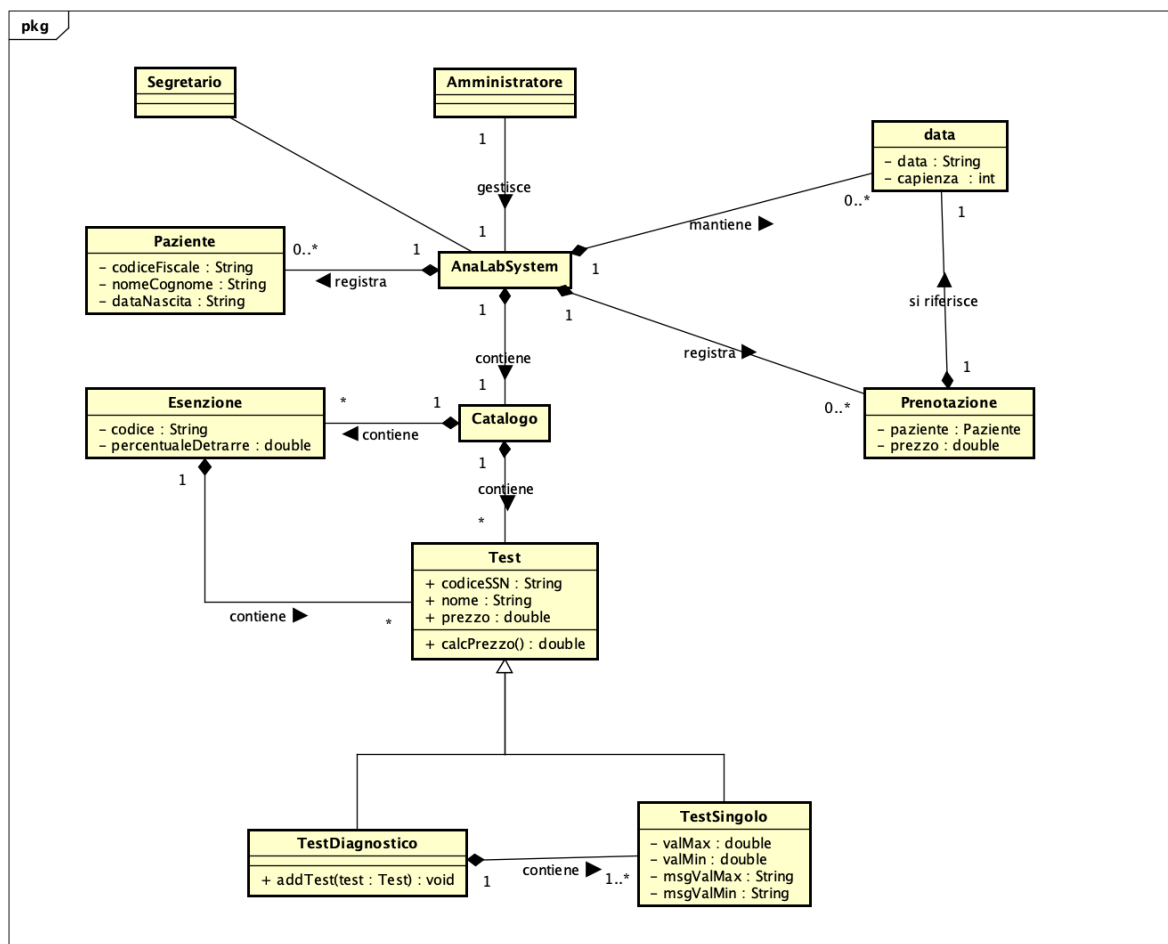
2.1 Analisi

2.1.1 Modello di Dominio

Dalla descrizione del caso d'uso in formato dettagliato possiamo identificare le seguente classe concettuale aggiuntiva:

- **Prenotazione**: una prenotazione che è possibile registrare nel sistema.
- **Data**: racchiude il concetto di data che è possibile associare ad una prenotazione ed inserire nel calendario, tenendo traccia del numero di prenotazioni per quel giorno.

Da cui, tenendo conto di associazioni e attributi, è stato ricavato il seguente Modello di Dominio:



Modello di dominio

2.1.2 Diagramma di sequenza di sistema

Il diagramma di sequenza di sistema per il caso d'uso UC1 è il seguente:

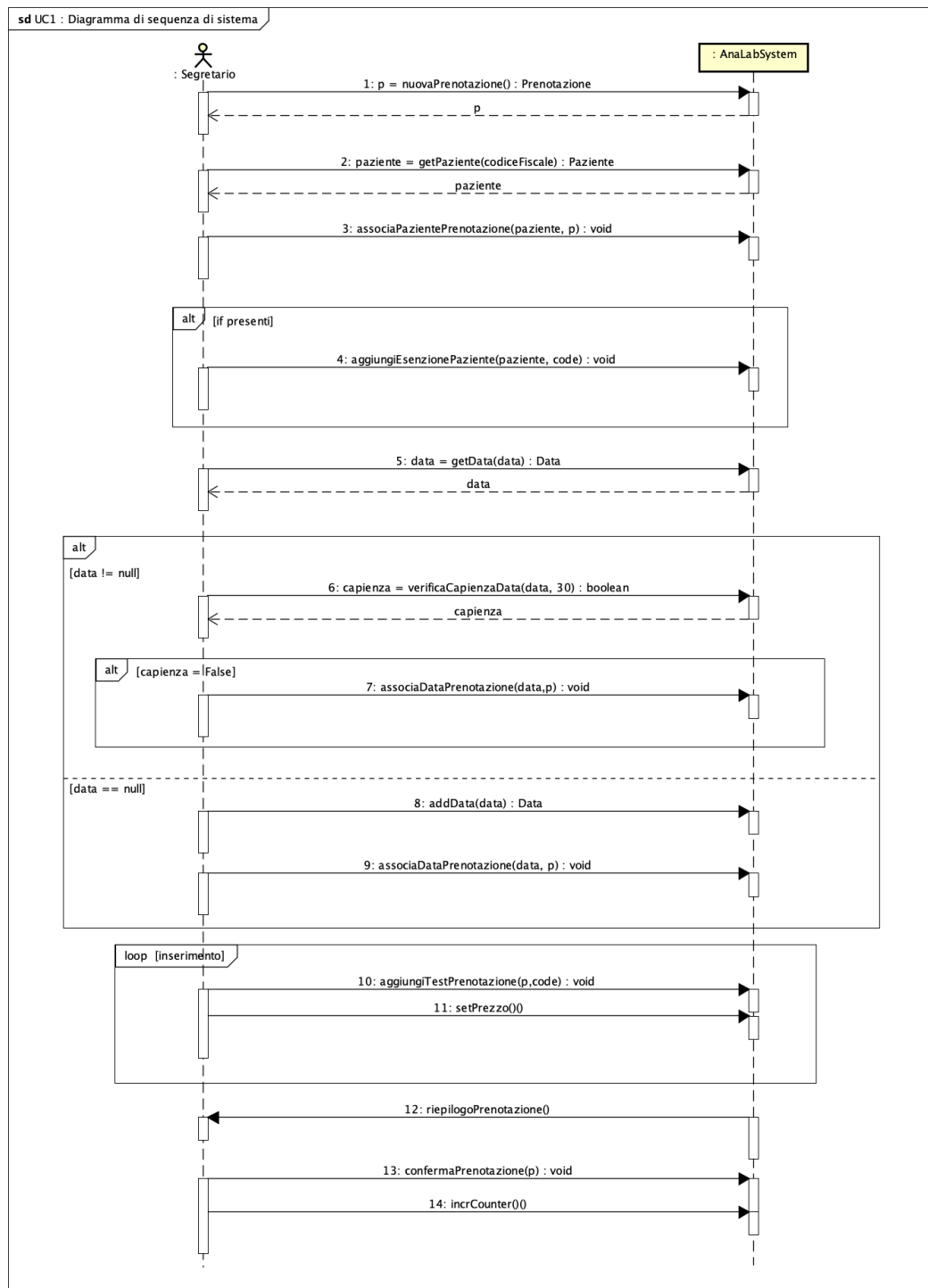


Diagramma di sequenza di sistema UC1

2.1.3 Contratti delle operazioni

nuovaPrenotazione()

Descrizione:	L'operazione consente di iniziare la creazione di una nuova prenotazione.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	-
Post-condizioni:	è stato creato una nuova istanza <i>p</i> di Prenotazione, gli attributi di <i>p</i> sono stati inizializzati correttamente <i>p</i> è associato a AnaLabSystem tramite l'associazione corrente

getPaziente(codiceFiscale)

Descrizione:	L'operazione consente di trovare l'istanza di Paziente dal codice fiscale dello stesso.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	-
Post-condizioni:	è stato trovato il paziente <i>paziente</i> corrispondente al codice fiscale inserito (oppure, in uno scenario alternativo, non è stato trovato il paziente e l'operazione ritorna un valore <i>null</i>)

associaPazienteprenotazione(*p*, *paziente*)

Descrizione:	L'operazione consente di associare il paziente <i>paziente</i> a prenotazioneCorrente.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	- è in corso la registrazione di una prenotazione <i>prenotazioneCorrente</i> , il paziente è già registrato nel sistema.
Post-condizioni:	<i>paziente</i> è stato associato a <i>prenotazioneCorrente</i> .

getData(getData)

Descrizione:	L'operazione consente di trovare l'istanza di Data dalla data scritta in formato dd/mm/yyyy (String) nel calendario gestito dal sistema.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	-
Post-condizioni:	è stato trovata la data <i>data</i> corrispondente alla stringa inserita. (oppure, in uno scenario alternativo, non è stata trovata la data e l'operazione ritorna un valore <i>null</i>)

verificaCapienzaData(data)

Descrizione:	L'operazione consente di verificare che la data selezionata non abbia raggiunto la capienza massima di prenotazioni.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	-
Post-condizioni:	l'operazione ritorna un tipo boolean: <i>true</i> se per quella data è stato raggiunto il numero massimo di prenotazioni, <i>false</i> se c'è ancora spazio per nuove prenotazioni.

addData(data)

Descrizione:	L'operazione consente di aggiungere una nuova data <i>data</i> nel calendario, se non esistono già prenotazioni per quella data.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	-
Post-condizioni:	è stata aggiunta la data <i>data</i> al calendario del sistema.

associaDataPrenotazione(p, data)

Descrizione:	L'operazione consente di associare la data <i>data</i> alla prenotazione corrente <i>p</i> .
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	è in corso la registrazione di una prenotazione <i>p</i> .
Post-condizioni:	<i>data</i> è stata associata alla prenotazione corrente <i>p</i> .

aggiungiTestPrenotazione(p, code)

Descrizione:	L'operazione consente di aggiungere un test all'istanza p corrente.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	è in corso la registrazione di una prenotazione p .
Post-condizioni:	un test è stato associato all'istanza di Prenotazione p tramite l'associazione "contiene"

setPrezzo()

Descrizione:	L'operazione consente di determinare il prezzo della prenotazione p corrente considerando l'attuale lista di test prenotati e le eventuali esenzioni del paziente.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	è in corso la registrazione di una prenotazione p . Un nuovo test è stato associato a p .
Post-condizioni:	l'attributo <i>prezzo</i> di p è stato aggiornato correttamente.

confermaPrenotazione(p)

Descrizione:	L'operazione consente di terminare e inserire una nuova prenotazione nell'elenco delle prenotazioni del sistema, al termine dell'inserimento di tutte le informazioni.
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	è in corso la registrazione di una prenotazione p .
Post-condizioni:	è stata correttamente inserita l'istanza corrente p di Prenotazione nell'elenco di prenotazioni del sistema.

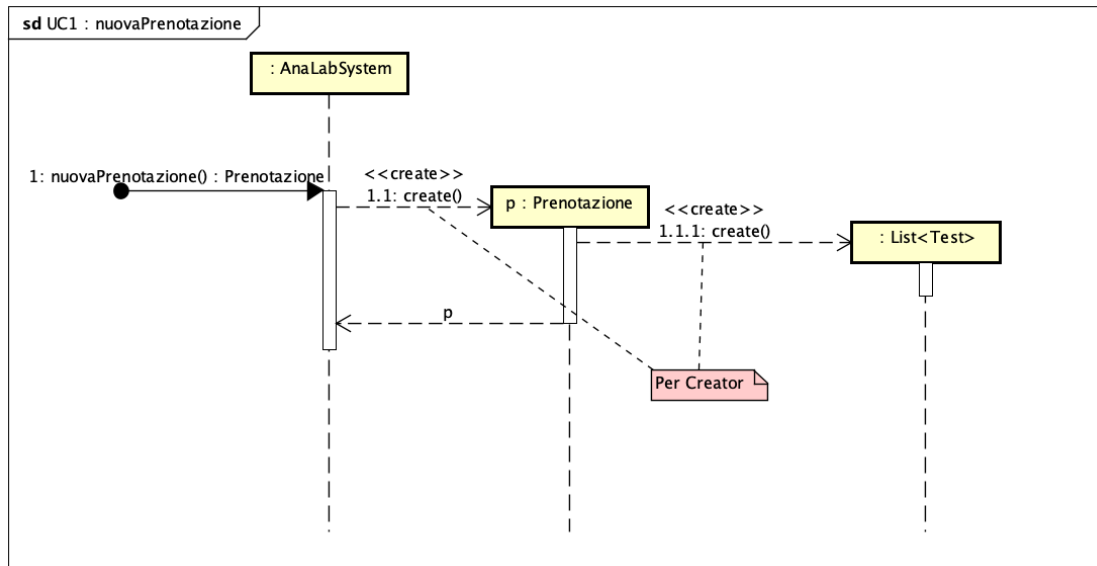
incrCounter

Descrizione:	L'operazione consente di incrementare di 1 il numero di prenotazioni della data associata alla prenotazione corrente p .
Riferimenti:	Caso d'uso: UC1 Registra nuova prenotazione
Pre-condizioni:	è in corso la registrazione di una prenotazione p .
Post-condizioni:	l'attributo <i>counter</i> di <i>data</i> è stato incrementato di 1.

2.2 Progettazione

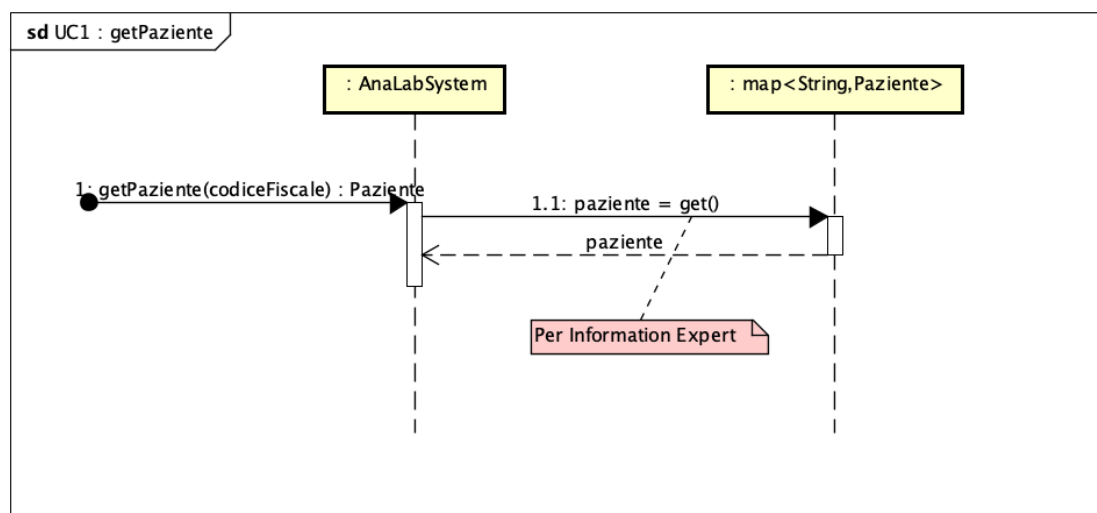
2.2.1 Diagrammi di Interazione

nuovaPrenotazione Parametri:



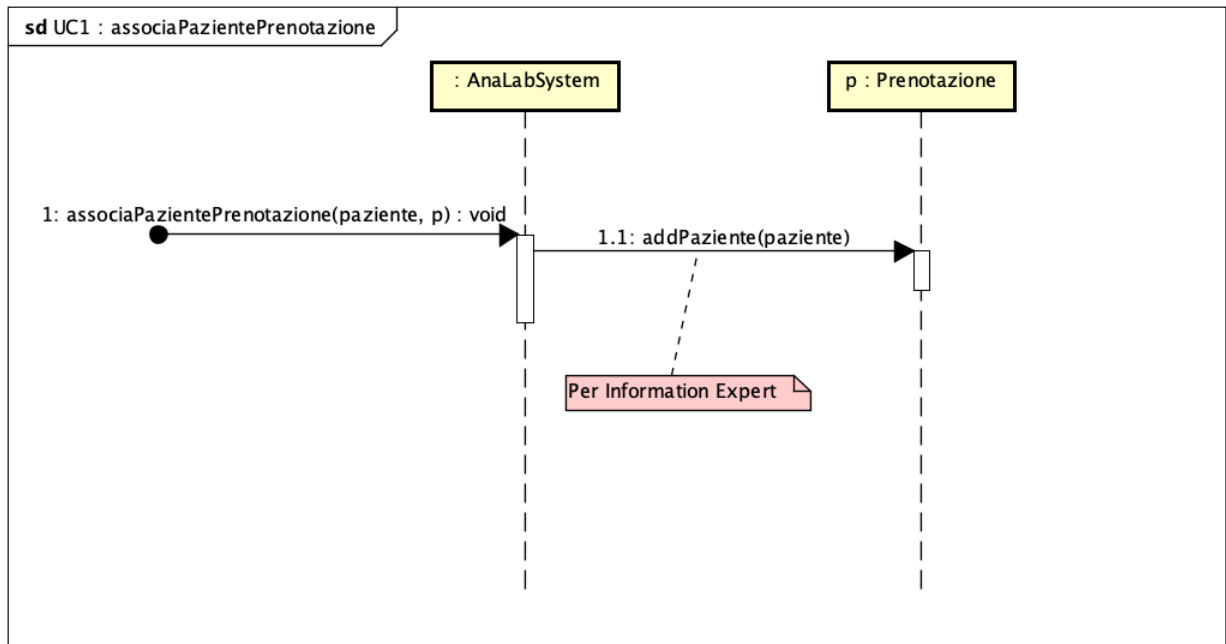
Diagrammi di Interazione: nuovaPrenotazione

getPaziente Parametri: (*codiceFiscale : String*)



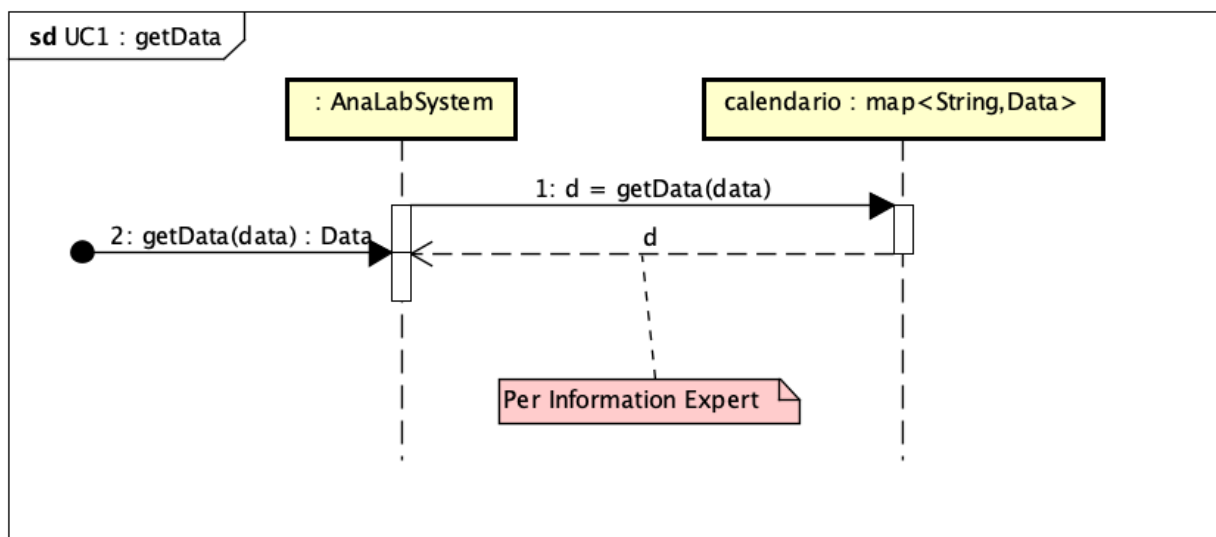
Diagrammi di Interazione: getPaziente

associaPazienteprenotazione Parametri: (*p : Prenotazione, paziente Paziente*)



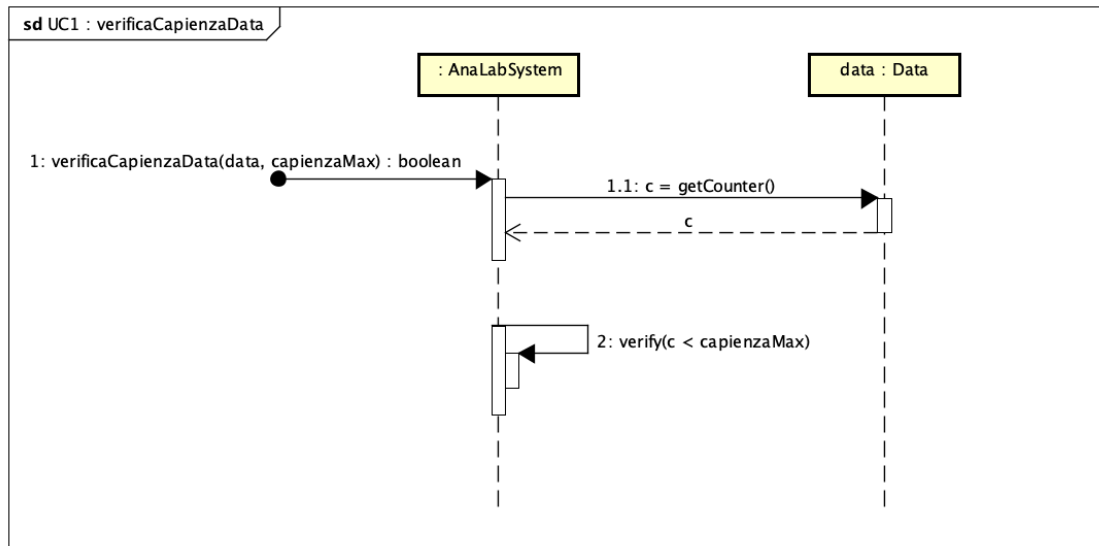
Diagrammi di Interazione: associaPazienteprenotazione

getData Parametri: (*data*: String



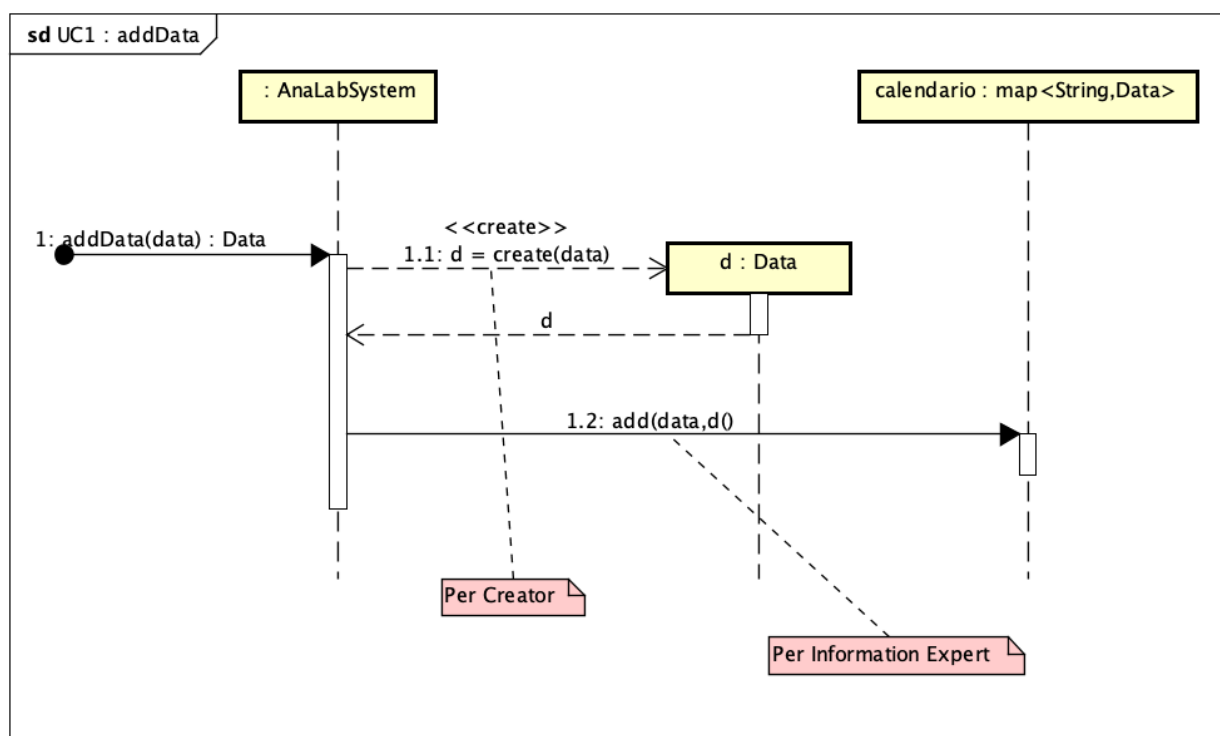
Diagrammi di Interazione: getData

verificaCapienzaData Parametri: (*data* : Data)



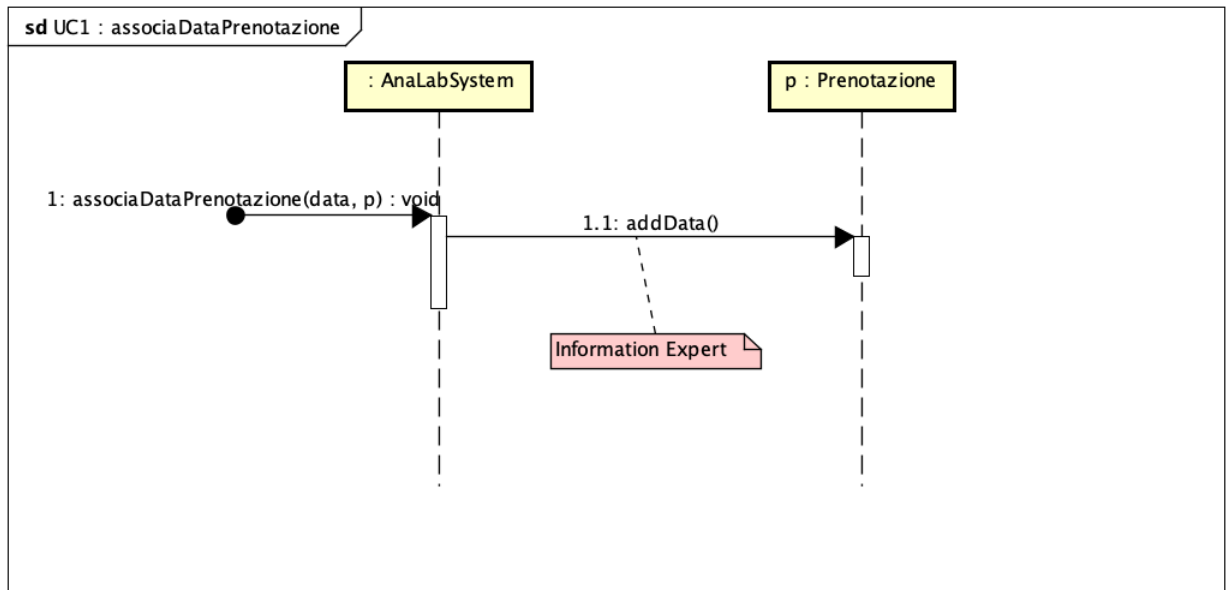
Diagrammi di Interazione: `verificaCapienzaData`

addData Parametri: *(data : String)*



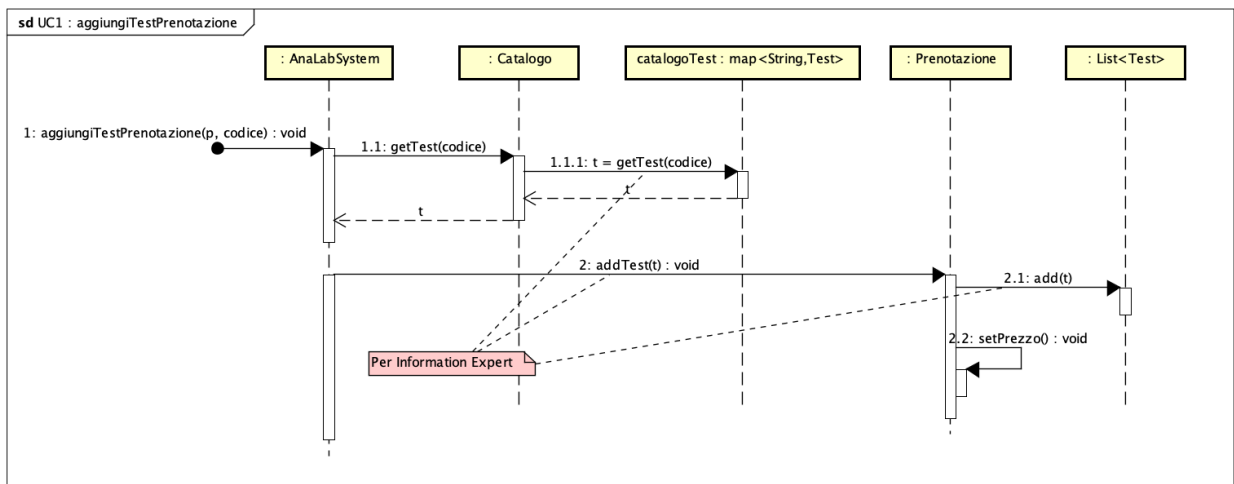
Diagrammi di Interazione: `addData`

associaDataPrenotazione Parametri: *(d : Data, prenotazione Prenotazione)*



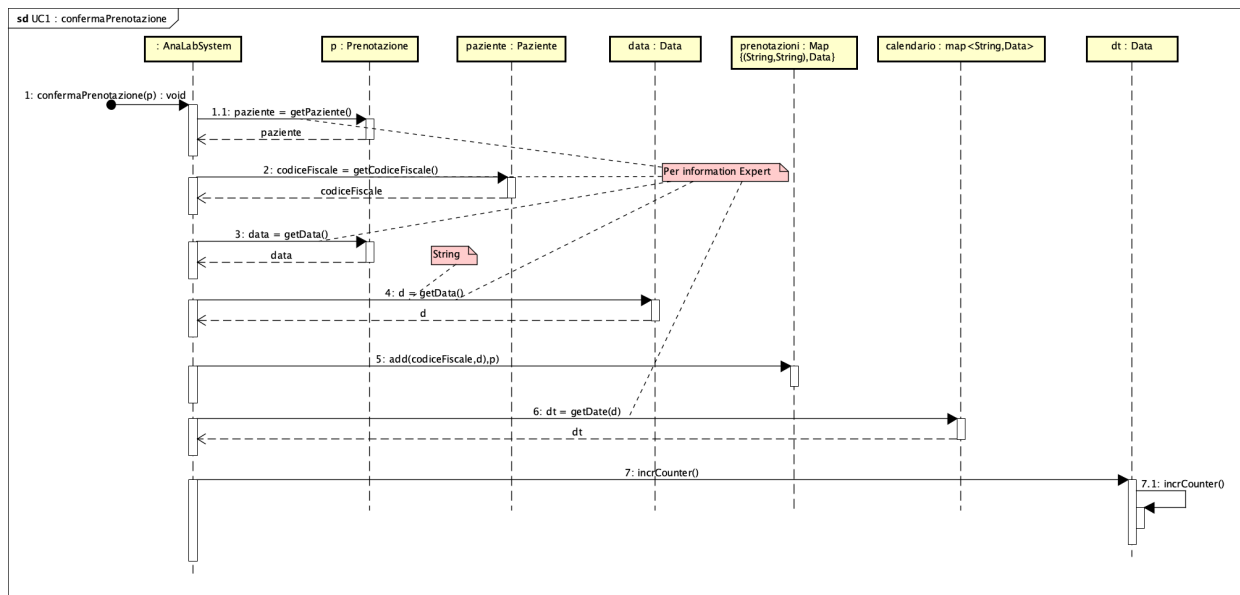
Diagrammi di Interazione: associaDataPrenotazione

aggiungiTestPrenotazione Parametri: (*p : Prenotazione, String code*)



Diagrammi di Interazione: aggiungiTestPrenotazione

confermaPrenotazione Parametri: (*p : Prenotazione*)



Diagrammi di Interazione: confermaPrenotazione

2.2.2 Diagramma delle Classi

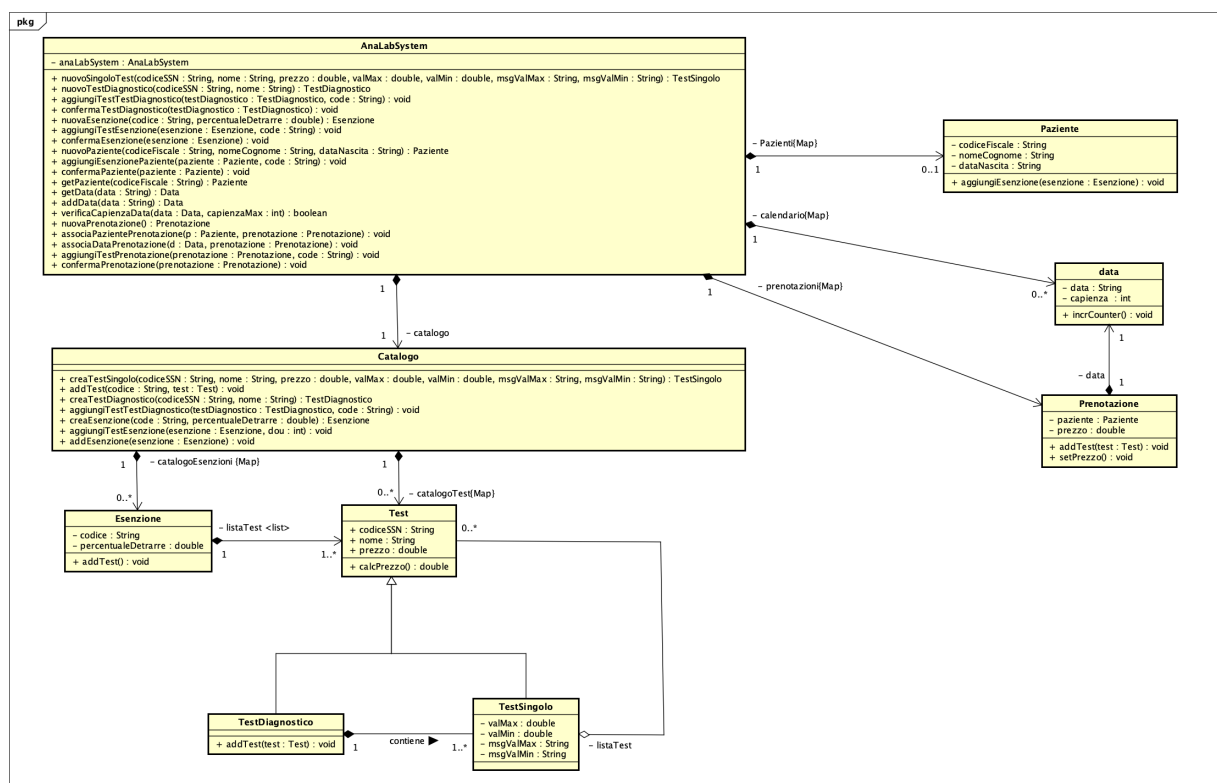


Diagramma delle Classi

3 Caso d'uso UC8: Visualizza prenotazioni di una data

3.1 Analisi

Prima di tutto, scriviamo il caso d'uso in formato dettagliato:

3.1.1 Formato Dettagliato

- **Nome:** Visualizza prenotazioni di una data
- **Livello:** Obiettivo utente
- **Attore primario:** Segretario
- **Parti interessate e interessi:**
 - **Segretario:** potrà visualizzare per ogni giorno la lista di prenotazioni, e quindi di pazienti da aspettarsi.
- **Pre-condizioni:**
 - Esistono prenotazioni per la data inserita
- **Post-condizioni:**
 - Sono visualizzate con successo le prenotazioni del giorno inserito.
- **Scenario principale di successo:**
 1. Il segretario inizia la procedura di visualizzazione prenotazioni.
 2. Il segretario inserisce una data.
 3. Il sistema restituisce la lista di prenotazioni per quella data..
- **Estensioni:**
 -
- **Frequenza di ripetizioni:** Frequente.

3.1.2 Modello di Dominio

Non vi è la necessità di aggiungere ulteriori classi di dominio per la realizzazione del caso d'uso.

3.1.3 Diagramma di sequenza di sistema

Il diagramma di sequenza di sistema per il caso d'uso UC8 è il seguente:

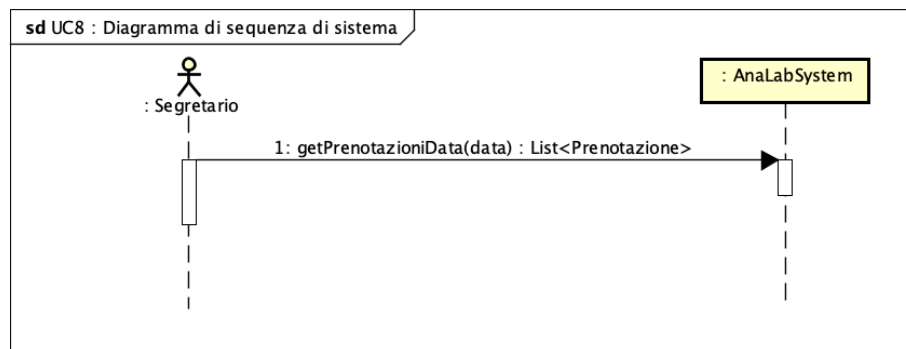


Diagramma di sequenza di sistema UC8

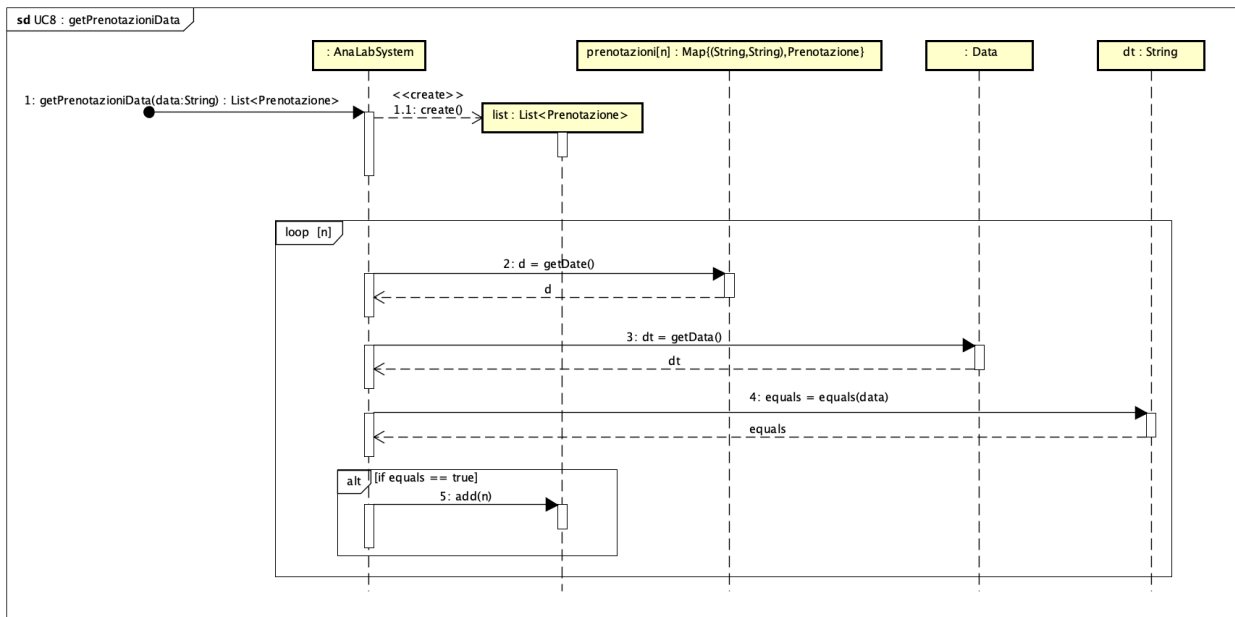
3.1.4 Contratti delle operazioni

Descrizione:	L'operazione consente di trovare le prenotazioni di una data inserita dall'attore (Segretario).
Riferimenti:	Caso d'uso: UC8: Visualizza prenotazioni data
Pre-condizioni:	-
Post-condizioni:	sono state trovate le prenotazioni p corrispondente alla data inserita

3.2 Progettazione

3.2.1 Diagrammi di Interazione

getPrenotazioniData Parametri: (*data* : String)



Diagrammi di Interazione: getPrenotazioniData

3.2.2 Diagramma delle Classi

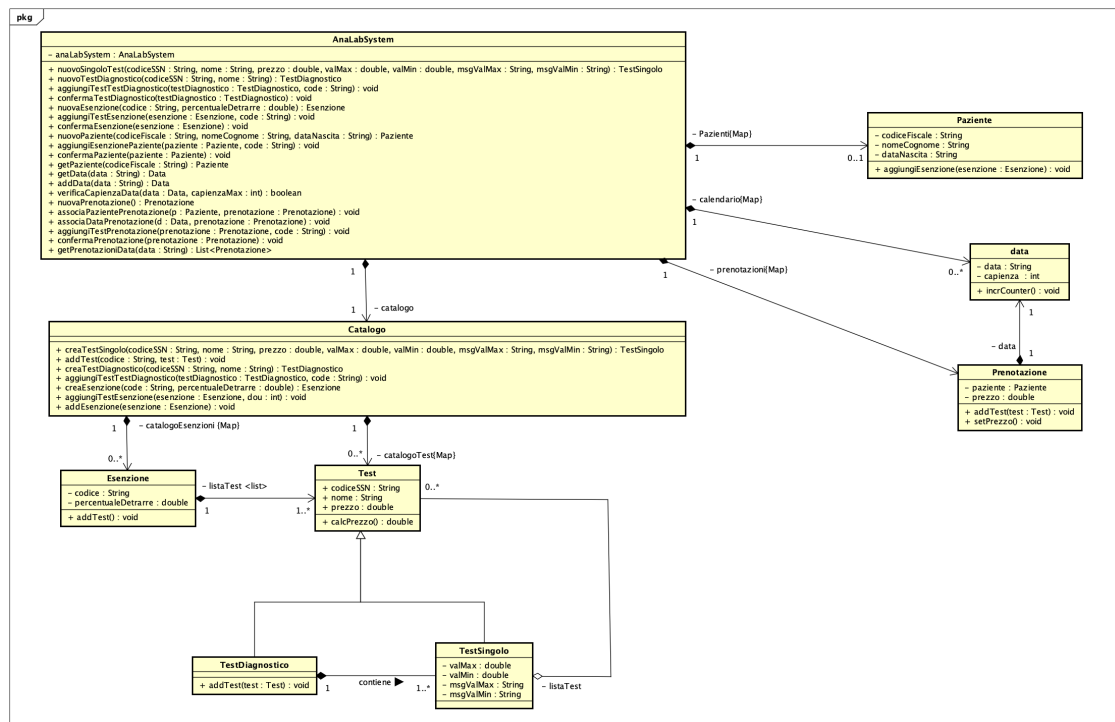


Diagramma delle Classi

4 Implementazione

Tutti i casi d'uso sono stati implementati e alcuni brevi test per valutare i metodi con una più alta probabilità di errore.

4.1 Testing

Si è scelto di implementare test a scatola nera, testando in particolare i metodi che calcolano i prezzi, i quali possono presentare le maggiori criticità. L'approccio utilizzato in questa iterazione è di bottom-up; si testano, quindi, i metodi più interni al sistema (metodi che operano solo sugli attributi pubblici e privati della classe stessa).

4.1.1 Prenotazione

E' stato testato il metodo di *Prenotazione setPrezzo* che permette di calcolare il prezzo di una prenotazione applicando le opportune detrazioni in base alle esenzioni possedute dal paziente prenotante. Sono stati creati, nel metodo di setUp test, esenzioni e tre pazienti: uno senza esenzioni, uno con una esenzione e uno con due esenzioni i cui test su cui si applica si sovrappongono. Sono stati realizzati tre test:

Prenotazione da paziente senza esenzioni: ci si aspetta che il prezzo sia semplicemente la somma dei prezzi dei test prenotati.

Prenotazione da paziente con una esenzione: ci si aspetta che il prezzo sia la somma dei prezzi dei test, alcuni dei quali sono scontati della *percentualeDetrarre* dell'esenzione a cui fanno riferimento, se il paziente la detiene.

Prenotazione da paziente con due esenzioni sovrapposte: ci si aspetta che il prezzo sia la somma dei prezzi dei test, alcuni dei quali sono scontati della *percentualeDetrarre* maggiore delle esenzioni a cui fanno riferimento, se il paziente le detiene.

Durante l'esecuzione del test ci si è accorti di un errore nella costruzione del metodo, per cui il prezzo calcolato nel caso di due esenzioni sovrapposte era quello dell'esenzione che si trovava nell'ultima posizione nella lista di esenzioni del Paziente. Tale errore è stato corretto immediatamente dopo l'individuazione.