

02- Elaborazione – Iterazione 2

Introduzione

Durante questa seconda iterazione la nostra decisione è stata quella di analizzare i successivi tre casi d'uso:

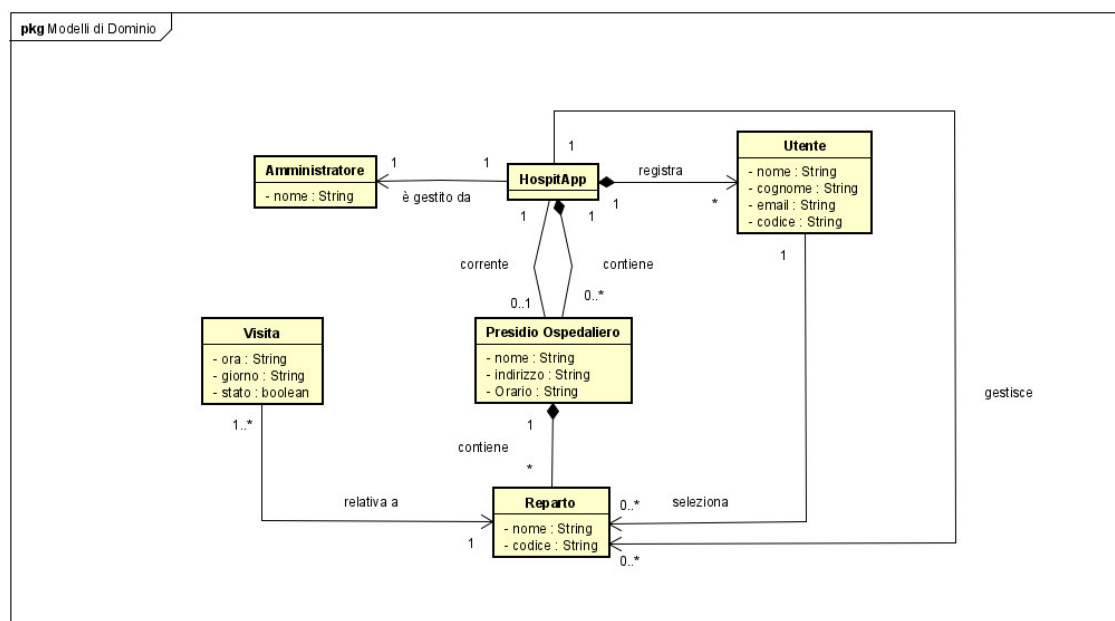
- Caso d'uso UC2: RICHIEDI PRENOTAZIONE. In questo caso d'uso l'attore è il paziente che seleziona un presidio e un reparto tra quelli disponibili nel presidio stesso e manda una richiesta di prenotazione di una prestazione medica.
- Caso d'uso UC3: GESTISCI PRENOTAZIONE. In questo caso d'uso l'attore, il presidio ospedaliero, assegna un giorno alla visita effettuata precedentemente dal paziente in base alle disponibilità del presidio stesso.
- Caso d'uso UC4: VISUALIZZA PRENOTAZIONE. In questo caso l'utente ha la possibilità di visualizzare le proprie prenotazioni e vedere lo stato della richiesta.

Analisi Orientata agli oggetti

Per delineare il dominio da una prospettiva orientata agli oggetti e affrontare ulteriori requisiti, impiegheremo nuovamente gli stessi strumenti utilizzati nella precedente iterazione, ossia il Modello di Dominio, il Diagramma di Sequenza di Sistema (SSD) e i Contratti delle Operazioni. Nello specifico, i paragrafi seguenti mettono in luce le modifiche apportate a tali documenti rispetto alla fase precedente.

Modello di Dominio

Il modello di Dominio del caso d'uso UC2 fa ovviamente riferimento al caso d'uso precedente e, tenendo conto di associazioni e attributi, è così modellato:



Analizzando il caso d'uso UC2, inoltre, emergono altre nuove classi concettuali:

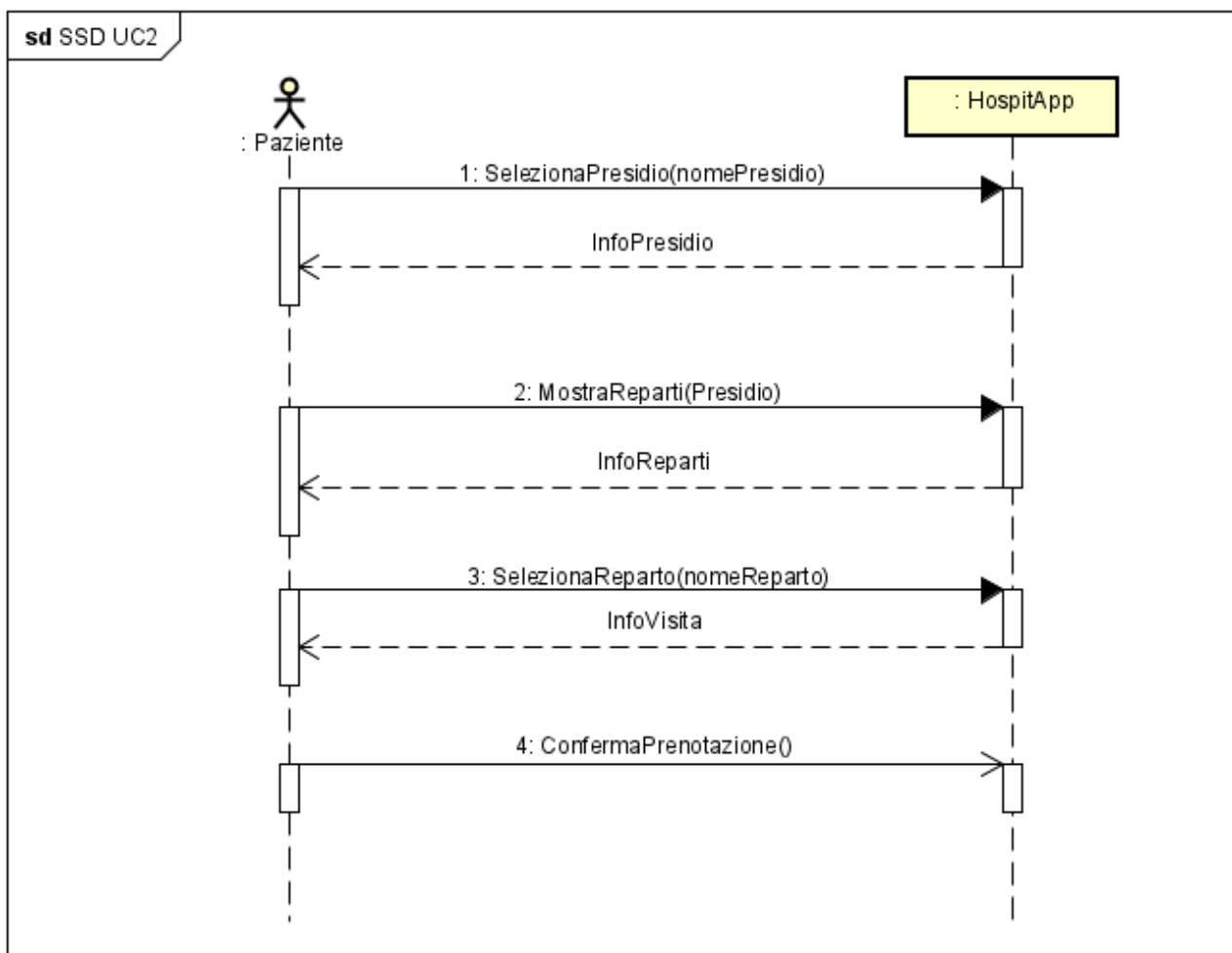
Utente: Rappresenta l'utente che effettua le operazioni all'interno del software HospitApp.

Visita: Rappresenta la richiesta dell'Utente di avere una prenotazione per un determinato presidio e reparto per una determinata prestazione medica.

Progettazione UC2

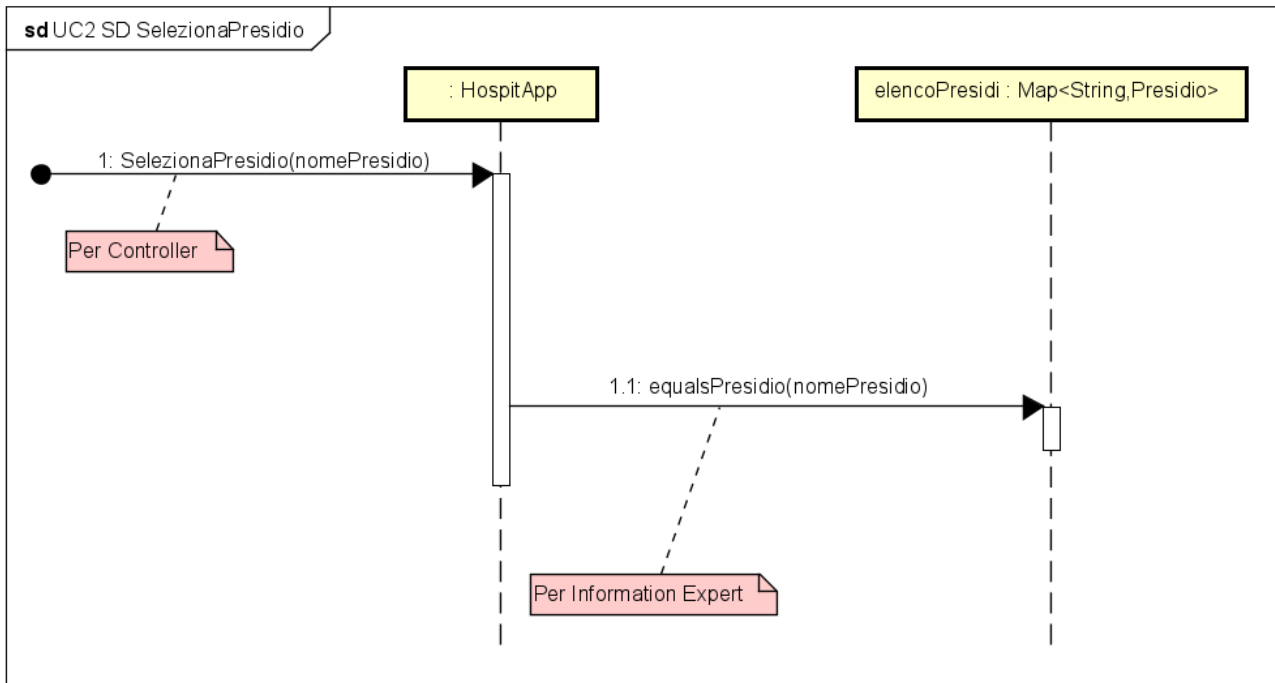
Nuovamente, l'elaborato principale di questa fase che è stato preso in considerazione è il Modello di Progetto, ovvero l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagramma delle Classi). Seguono dunque i diagrammi di Sequenza di sistema, diagrammi di Sequenza relativi al caso d'uso UC2.

Diagrammi di Sequenza di sistema UC2:

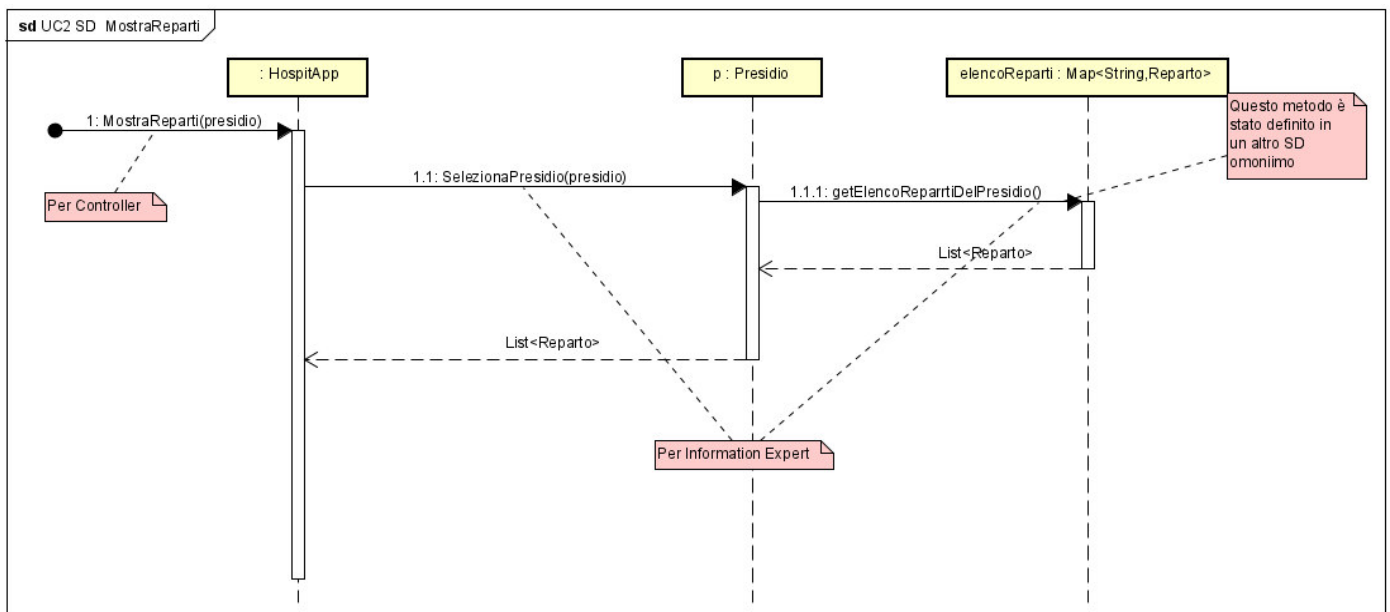


Diagrammi di Sequenza UC2:

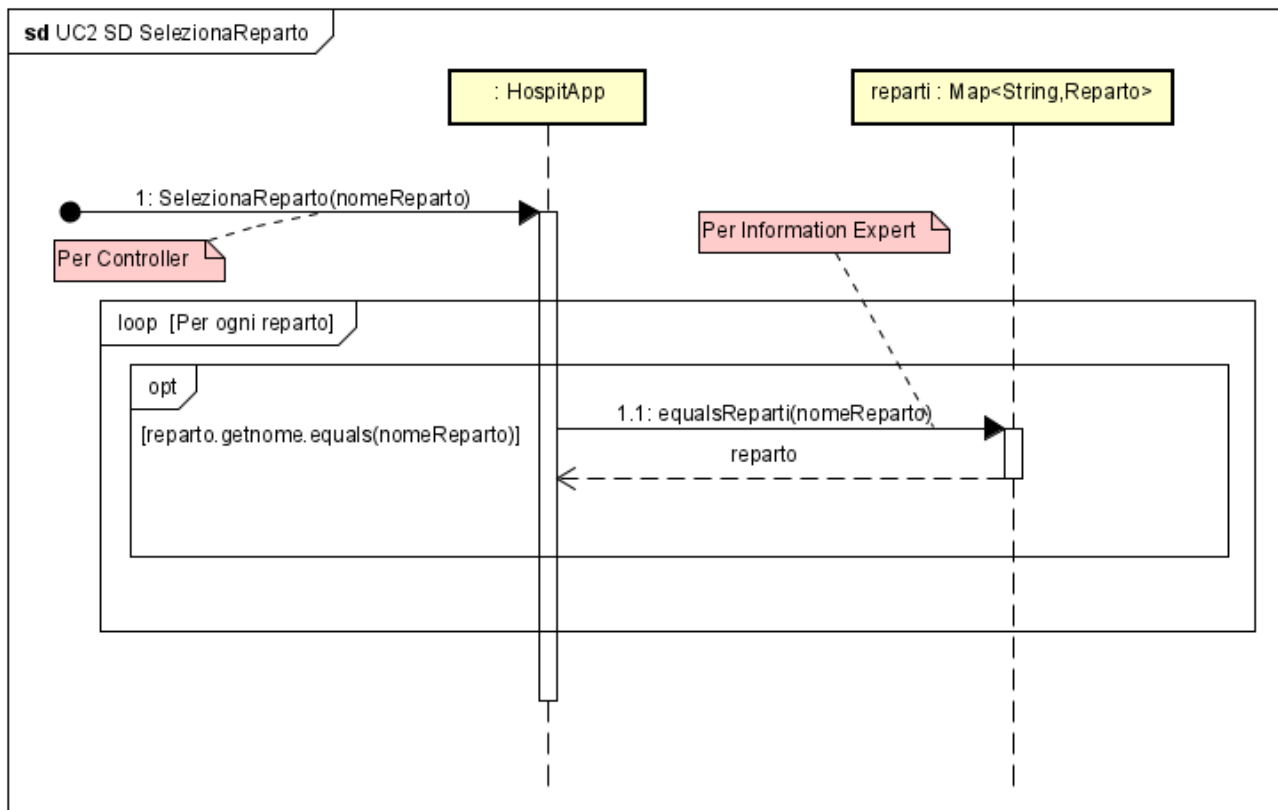
- **SelezionaPresidio** permette all'utente di selezionare un presidio per il quale vuole esprimere un interesse riguardo la visita che ha intenzione di prenotare.



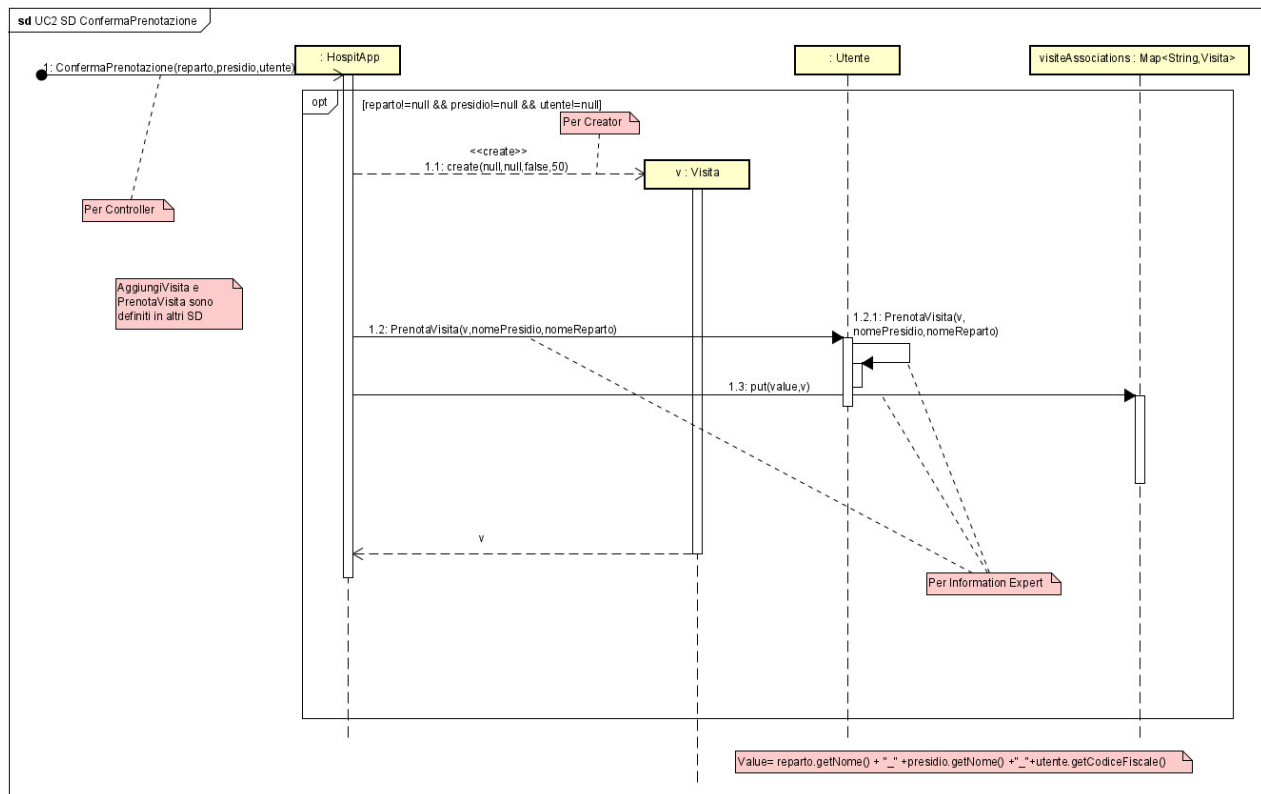
- **MostraReparti(presidio)** mostra i Reparti di un determinato Presidio presidio;



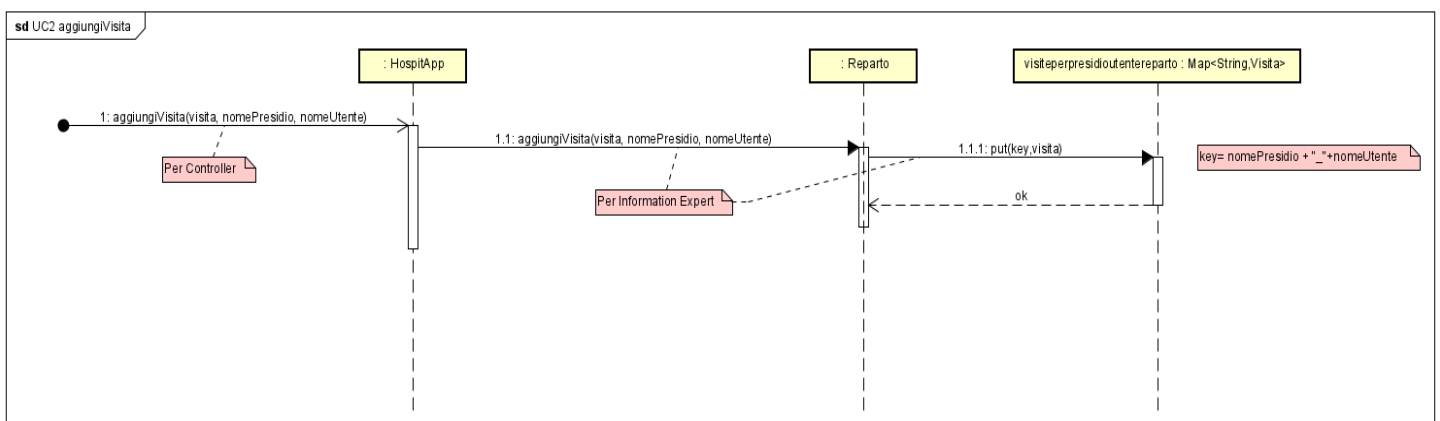
- **SelezionaReparto** permette all'utente di selezionare un determinato Reparto selezionato dall'utente.



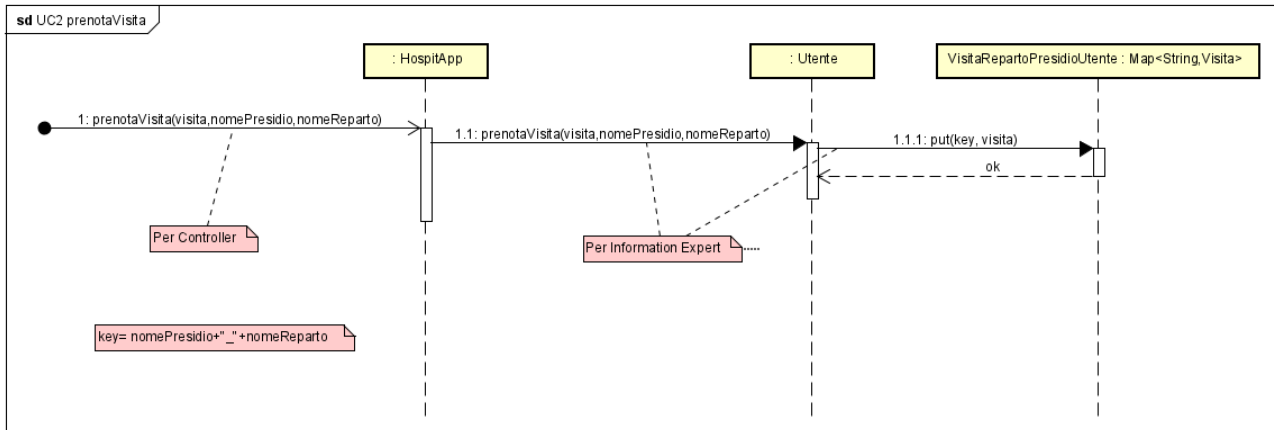
- **ConfermaPrenotazione** fa in modo che una volta che l'utente ha selezionato un presidio e un reparto di quello specifico presidio, viene creato un oggetto di tipo Visita, viene associata la specifica visita all'utente che ha effettuato la prenotazione e al reparto scelto e infine la visita viene aggiunta ad una Mappa (visiteAssociations) che contiene come key la concatenazione del nomeReparto, del nomePresidio e del codiceFiscaleUtente e come value la visita.



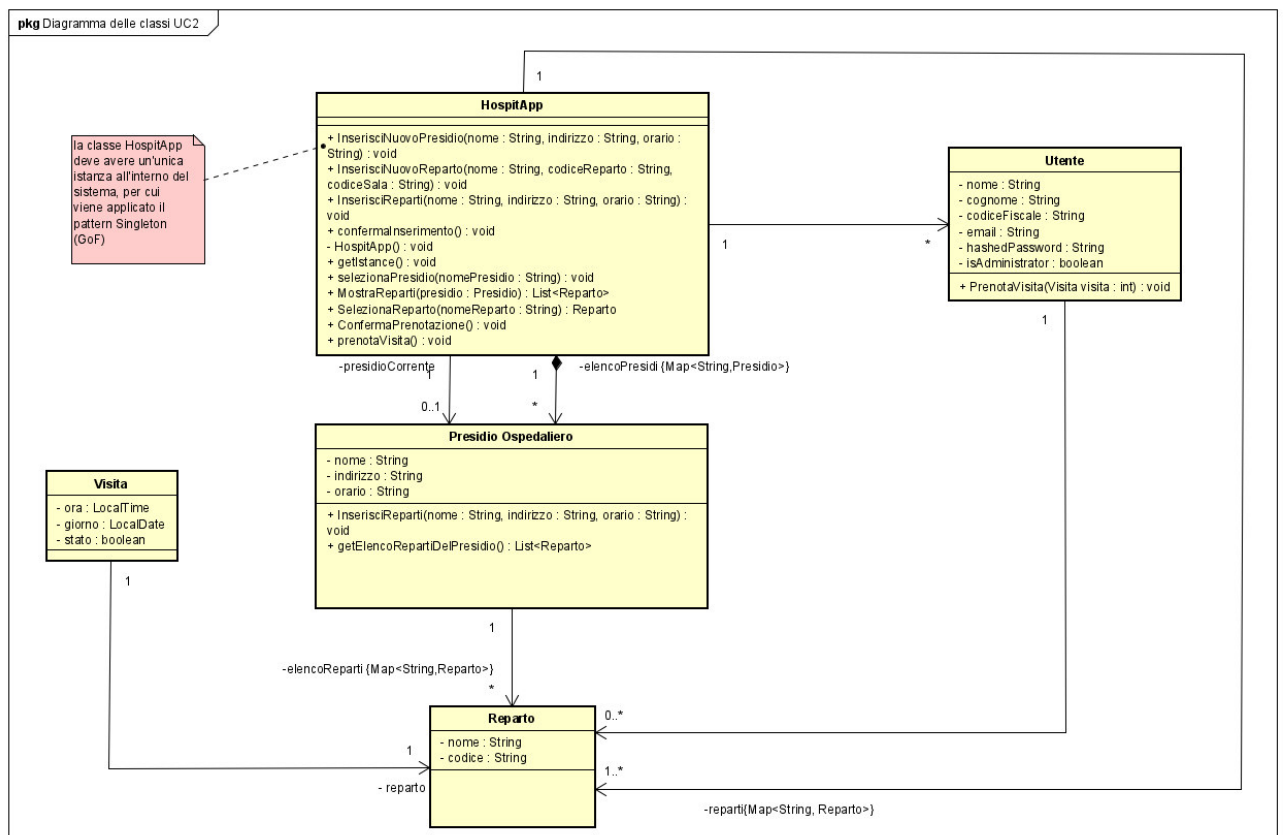
- Come scritto in precedenza, il metodo **aggiungiVisita** permette al sistema di salvare la visita specifica nel reparto selezionato dall'utente.



- Come scritto in precedenza, il metodo **prenotaVisita** permette al sistema di salvare la visita specifica nella classe dell'Utente che ha prenotato quella specifica visita..



• Diagramma delle classi UC2



Contratti delle operazioni UC2

1: Prenota Visita

Operazione:

PrenotaVisita(NomePresidio)

Riferimenti:

Caso d'uso: Richiedi Prenotazione;

Pre-condizioni:

- L'utente deve avere eseguito l'accesso ad HospitApp;
- L'utente ha bisogno di prenotare una visita;

Post-condizioni:

-

2: Mostra Reparti

Operazione:

MostraReparti(Presidio)

Riferimenti:

Caso d'uso: Richiedi Prenotazione;

Pre-condizioni:

- È stato inserito il Presidio p;

Post-condizioni:

- Sono stati restituiti tutti i reparti r del presidio p;

3: Selezione Reparto

Operazione:

SelezionaReparto(nomeReparto)

Riferimenti:

Caso d'uso: Richiedi Prenotazione;

Pre-condizioni:

- È stato inserito almeno un reparto r;
- L'utente ha visualizzato almeno un reparto;

Post-condizioni:

- Il Reparto r richiesto dall'utente è stato restituito;

4: Conferma Prenotazione

Operazione:

ConfermaPrenotazione(nomeReparto, nomePresidio)

Riferimenti:

Caso d'uso: Richiedi Prenotazione;

Pre-condizioni:

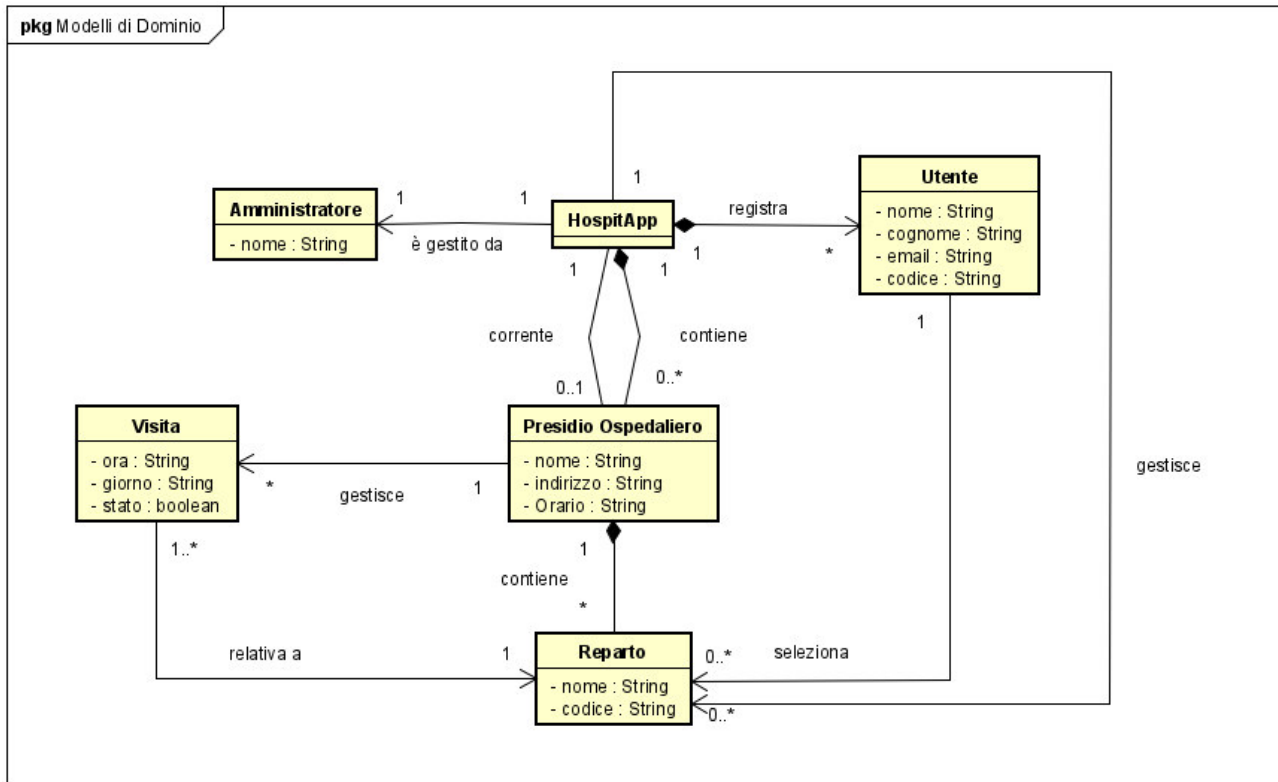
- È in corso l'inserimento di un reparto r (e quindi di un presidio p) da parte dell'utente;

Post-condizioni:

- È stata creata la nuova istanza v di visita;
- Gli attributi di visita sono stati inizializzati a "null";
- La visita v è stata associata a Reparto tramite l'associazione "relativa a";
- La classe v è stata associata a Presidio tramite l'associazione "gestisce";
- L'istanza v è stata aggiunta ad una mappa Map<String,Visita>, dove laVisita è associata all'utente u;
- La visita v viene ritornata;

Progettazione UC3

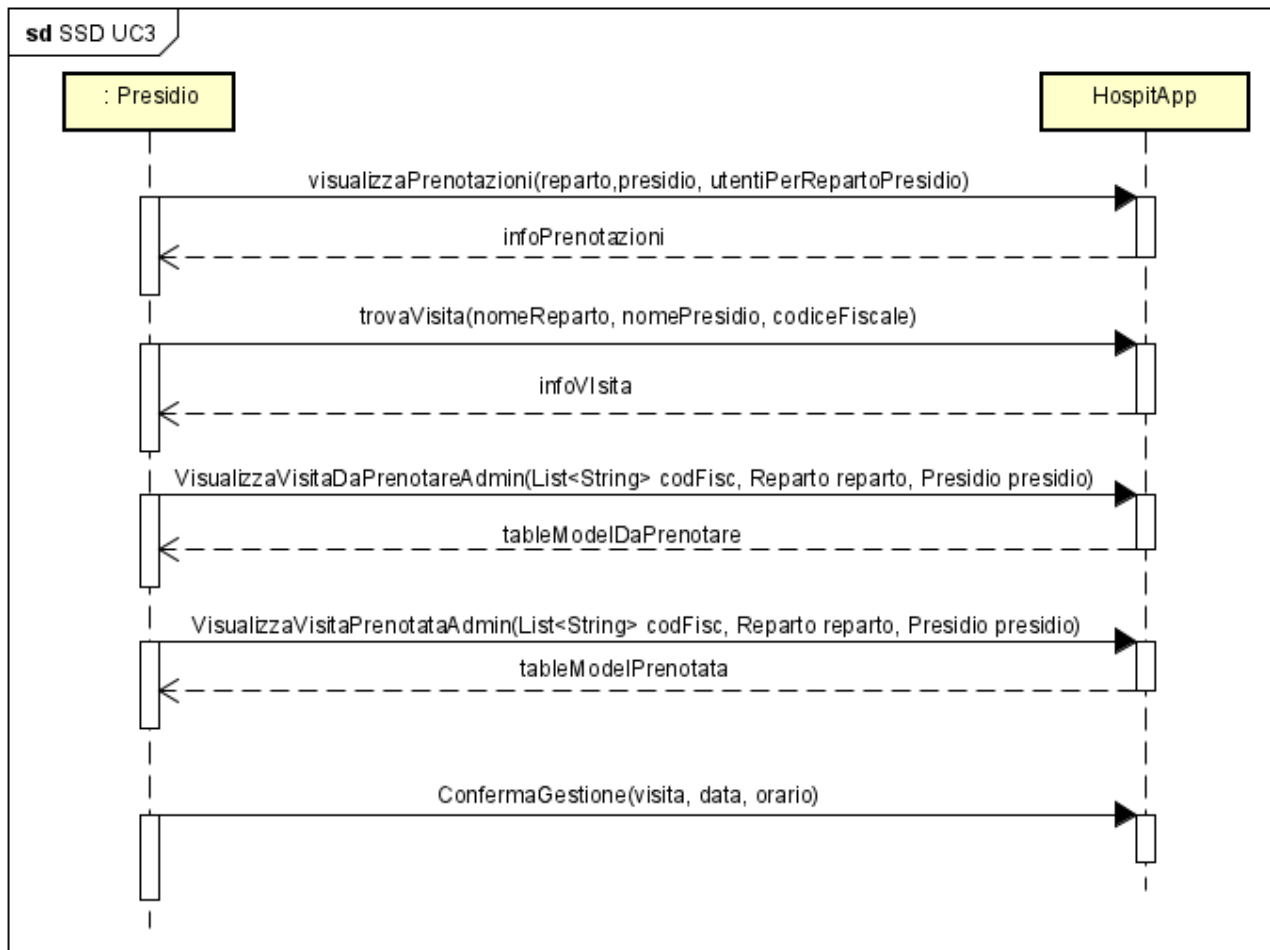
Il modello di Dominio UC3 riprende il modello di dominio UC2:



In questo caso non è stata necessaria l'introduzione di nuove classi, bensì è stato aggiunto il collegamento logico "gestisce" tra Presidio Ospedaliero e Visita. Grazie a questo, il presidio ospedaliero è in grado di accedere alle Visite inoltrate dagli Utenti ed è anche in grado di gestire, tramite le operazioni che verranno definite nel diagramma di sequenza di sistema (SSD) di gestirle.

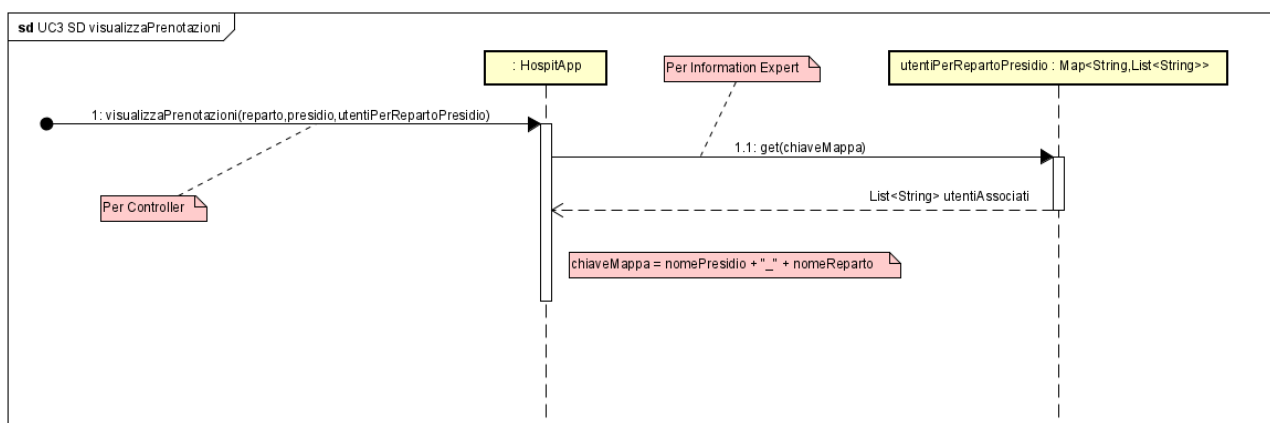
L'elaborato principale di questa fase che è stato preso in considerazione è il Modello di Progetto, ovvero l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagramma delle Classi). Seguono dunque i diagrammi di Sequenza di sistema, diagrammi di Sequenza relativi al caso d'uso UC3.

Diagrammi di Sequenza di sistema UC3:

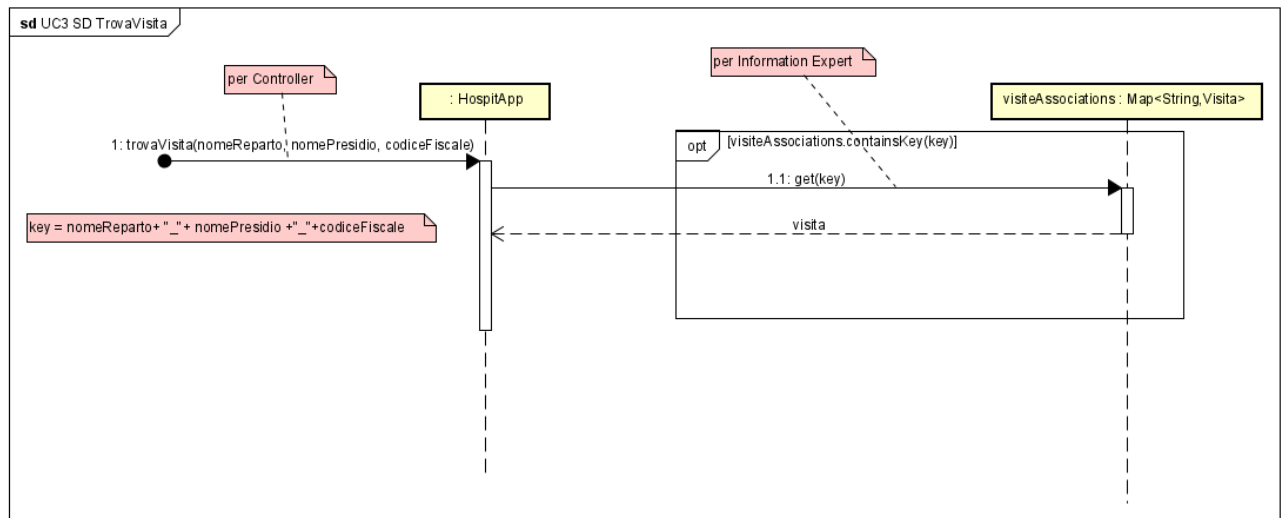


Diagrammi di Sequenza UC3:

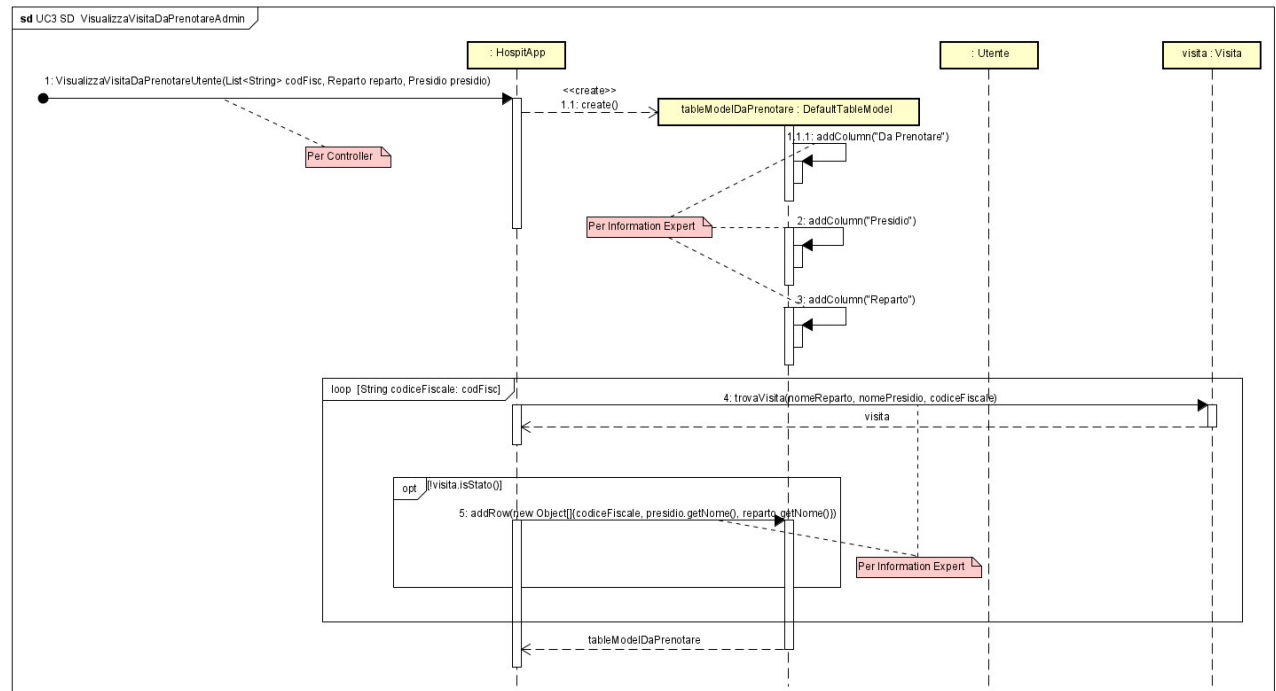
- Il metodo **VisualizzaPrenotazioni** permette di visualizzare tutti gli utenti che hanno effettuato una Prenotazione per quel Reparto;



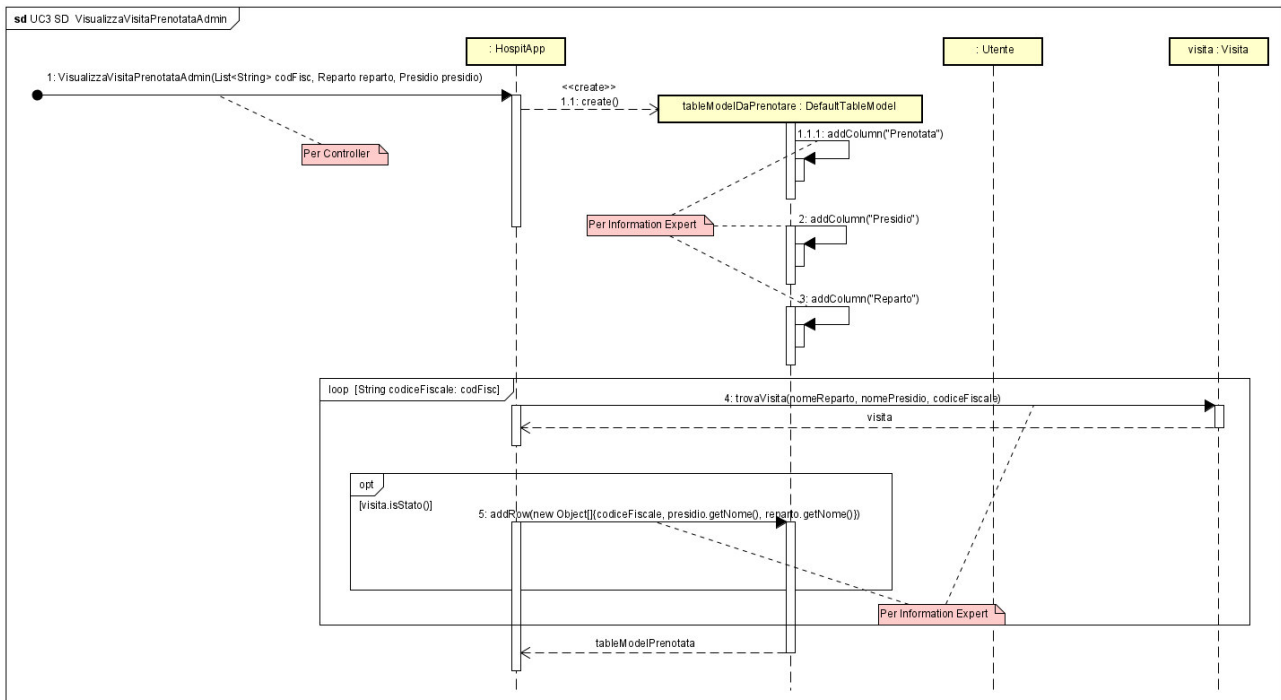
- Il metodo **TrovaVisita** richiama la specifica Visita in un determinato Reparto di un determinato Utente;



- VisualizzaVisitaDaPrenotareAdmin** questa funzione permette al presidio di visualizzare tutte le visite da prenotare degli utenti in un dato reparto associato a quel Presidio.



- **VisualizzaVisitaPrenotataAdmin** questa funzione permette al presidio di visualizzare tutte le visite prenotate degli utenti in un dato reparto associato a quel Presidio.



- Tramite il metodo **ConfermaGestione** setto il Giorno e l'orario della Visita per quel determinato Utente;

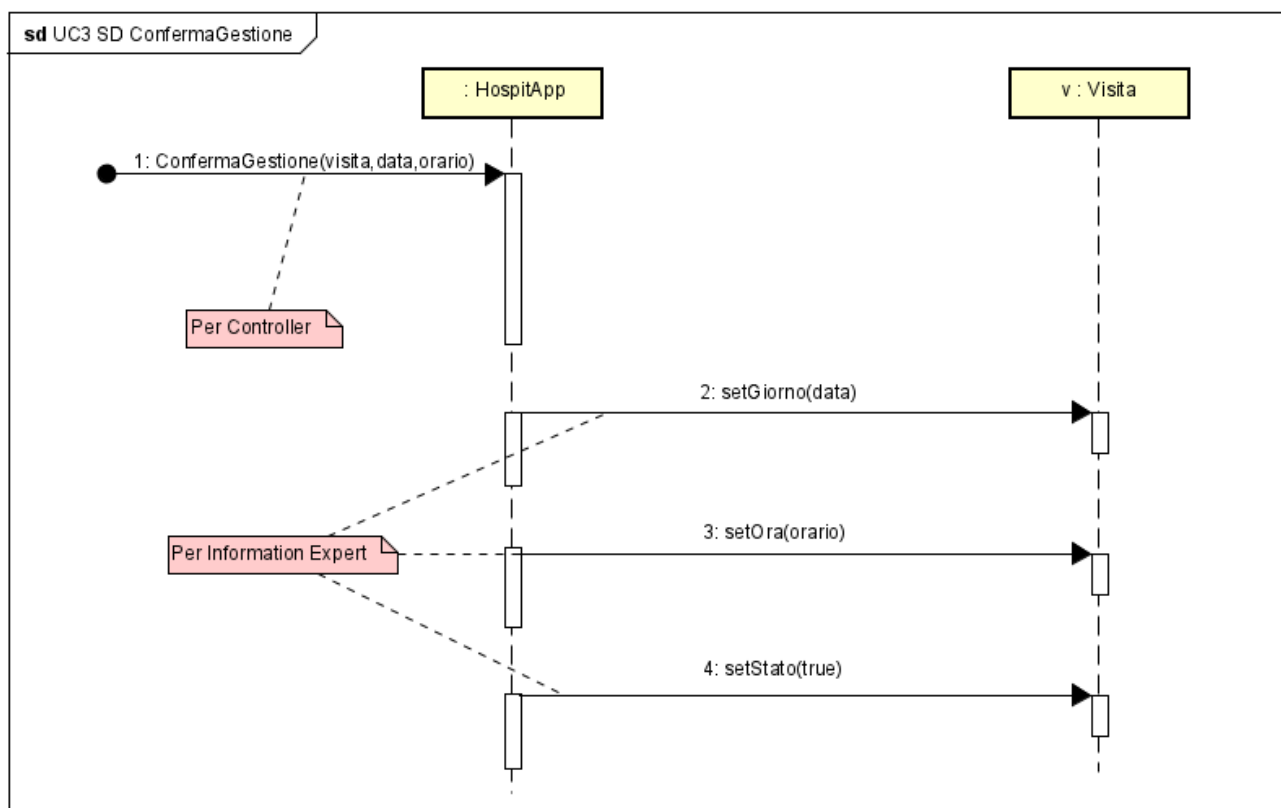
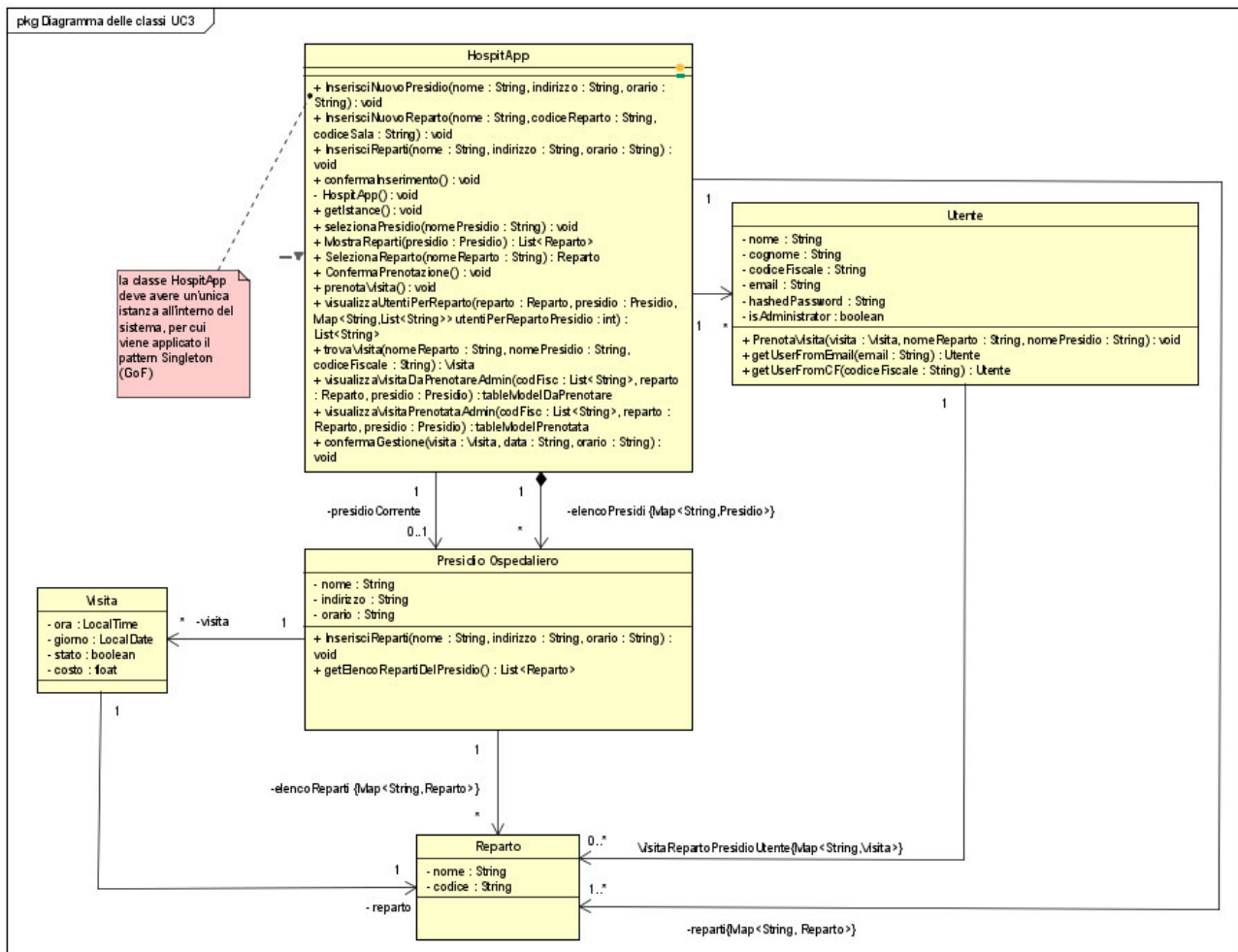


Diagramma delle classi UC3



Contratti delle operazioni UC3

1: visualizza Utenti per Reparto

Operazione:

visualizzaUtentiPerReparto(reparto, presidio, utentiPerRepartoPresidio)

Riferimenti:

- Caso d'uso: Gestisci Prenotazione;

Pre-condizioni:

- L'utente deve avere fatto richiesta di una visita per un determinato Reparto;
- La Map<String, List<String>> utentiPerRepartoPresidio è stata inizializzata e riempita con tutti gli utenti che hanno richiesto una Visita in uno specifico Reparto di un determinato Presidio;

Post-condizioni:

- La `List<String> utentiAssociati` viene inizializzata e riempita con i value di `utentiPerRepartoPresidio` per una specifica key (key: `presidio.getNome() + "_" + reparto.getNome()`);
- Viene restituita la lista `utentiAssociati`

2: Trova Visita

Operazione:

`trovaVisita(nomeReparto, nomePresidio, codiceFiscale)`

Riferimenti:

Caso d'uso: Gestisci Prenotazione;

Pre-condizioni:

- E' stata inizializzata e riempita la mappa `Map<String,Visita> visiteAssociations`;

Post-condizioni:

- Viene restituita la visita che nella mappa `visitaAssociations` viene associata alla key (key; `nomeReparto + "_" + nomePresidio + "_" + codiceFiscale`);

3: Visualizza Visita Da Prenotare Admin

Operazione:

`VisualizzaVisitaDaPrenotareAdmin(List<String> codFisc, Reparto reparto, Presidio presidio)`

Riferimenti:

Caso d'uso: Gestisci Prenotazione;

Pre-condizioni:

- È stata inizializzata una `List<String> utentiAssociati`;

Post-condizioni:

- È stata creata e riempita tante volte quanti sono gli utenti in utentiAssociati per quel Presidio e Reparto una tableModelDaPrenotare che contiene: “Da Prenotare”, “Presidio”, “Reparto”;

4: Visualizza Visita Prenotata Admin

Operazione:

VisualizzaVisitaPrenotataAdmin(List<String> codFisc, Reparto reparto, Presidio presidio)

Riferimenti:

Caso d’uso: Gestisci Prenotazione;

Pre-condizioni:

- È stata inizializzata una List<String> utentiAssociati;

Post-condizioni:

- È stata creata e riempita tante volte quanti sono gli utenti in utentiAssociati per quel Presidio e Reparto una tableModelPrenotata che contiene: “Da Prenotare”, “Presidio”, “Reparto”;

5: Conferma Gestione

Operazione:

confermaGestione(visita,data,orario)

Riferimenti:

Caso d'uso: Gestisci Prenotazione;

Pre-condizioni:

- È stata trovata una Visita visita in trovaVisita(nomeReparto, nomePresidio, codiceFiscale);

Post-condizioni:

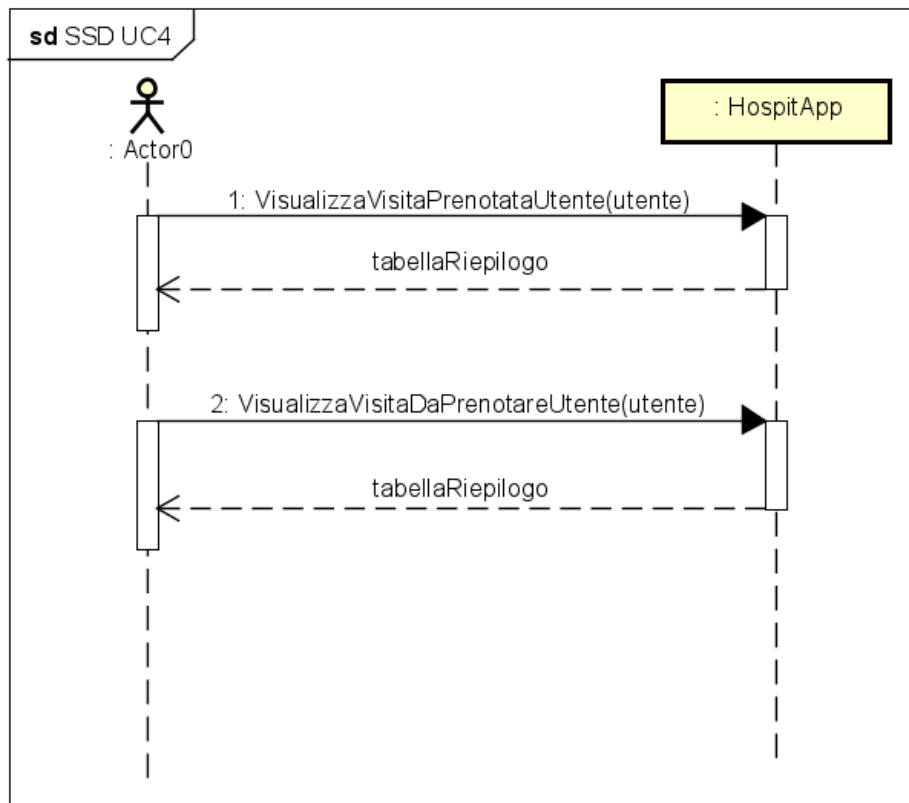
- È stato settato il giorno di visita uguale al valore scelto dal presidio;
- È stato settato l'orario di visita uguale al valore scelto dal presidio;
- È stato settato lo stato di visita a true;

Progettazione UC4

Il modello di Dominio UC4, essendo un caso d'uso di visualizzazione, non introduce nuove classi al modello di dominio già esistente. Piuttosto, si occupa di definire un nuovo metodo di visualizzazione evidenziato negli SSD e nel diagramma delle classi

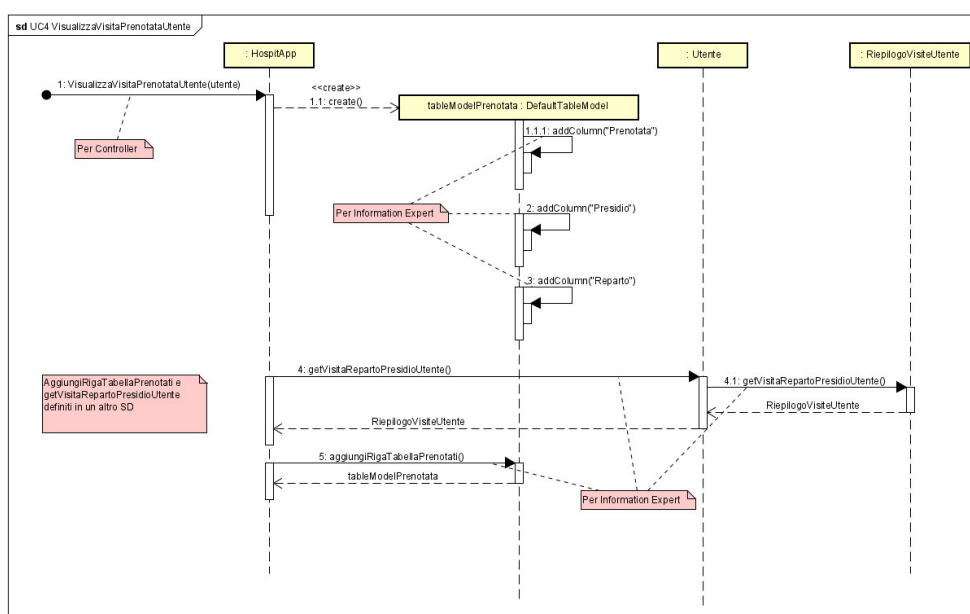
L'elaborato principale di questa fase che è stato preso in considerazione è il Modello di Progetto, ovvero l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagramma delle Classi). Seguono dunque i diagrammi di Sequenza di sistema, diagrammi di Sequenza relativi al caso d'uso UC4.

Diagrammi di Sequenza di sistema UC4:

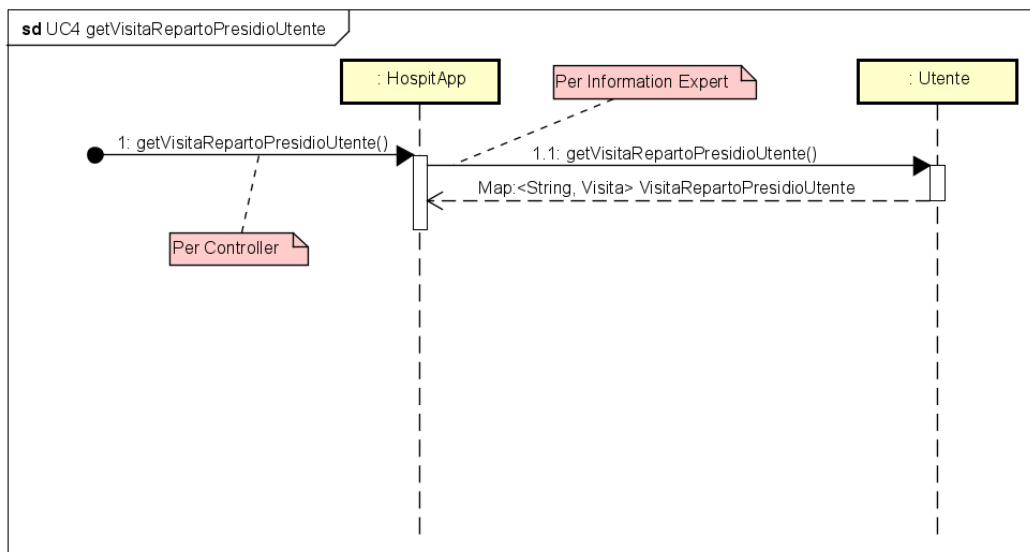


Diagrammi di Sequenza UC4:

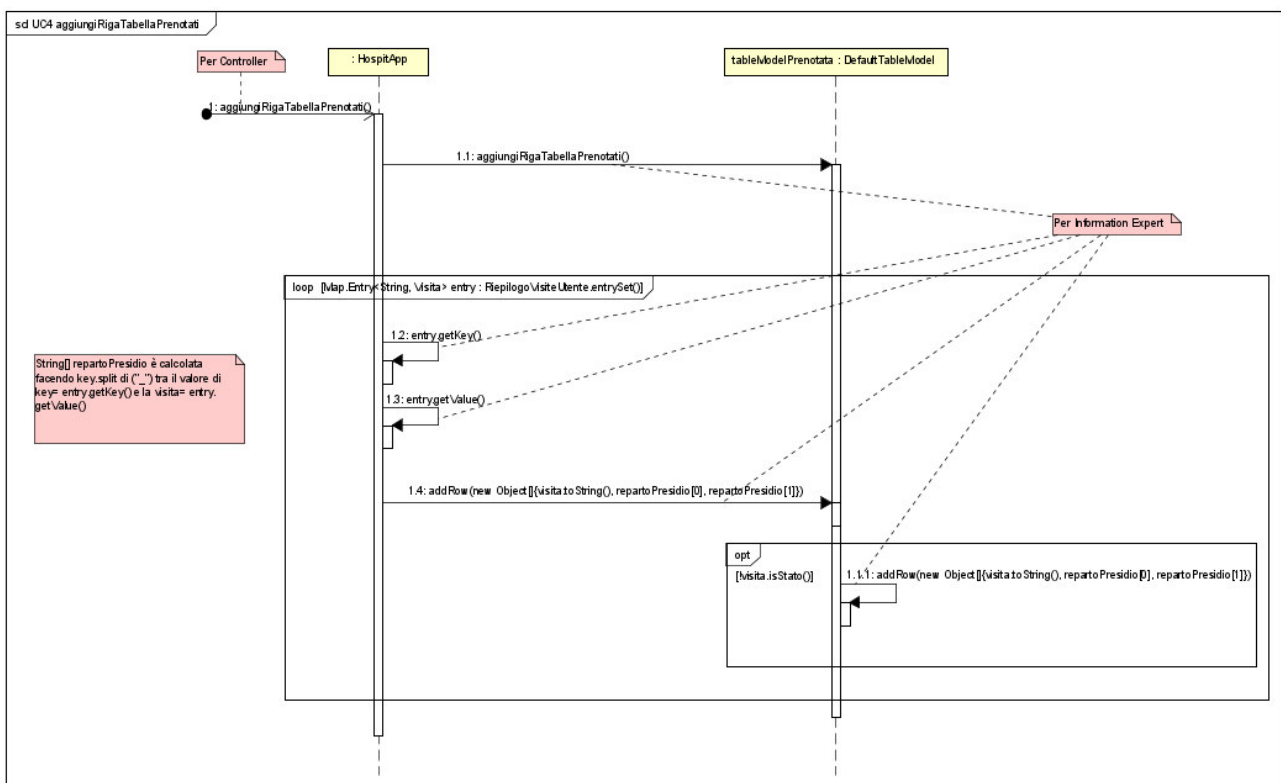
- **VisualizzaVisitaPrenotataUtente** permette di visualizzare una tabella contenente tutte le visite (con le relative informazioni) per cui l'utente ha inviato una prenotazione per le quali l'amministratore ha settato un giorno e un orario specifico.



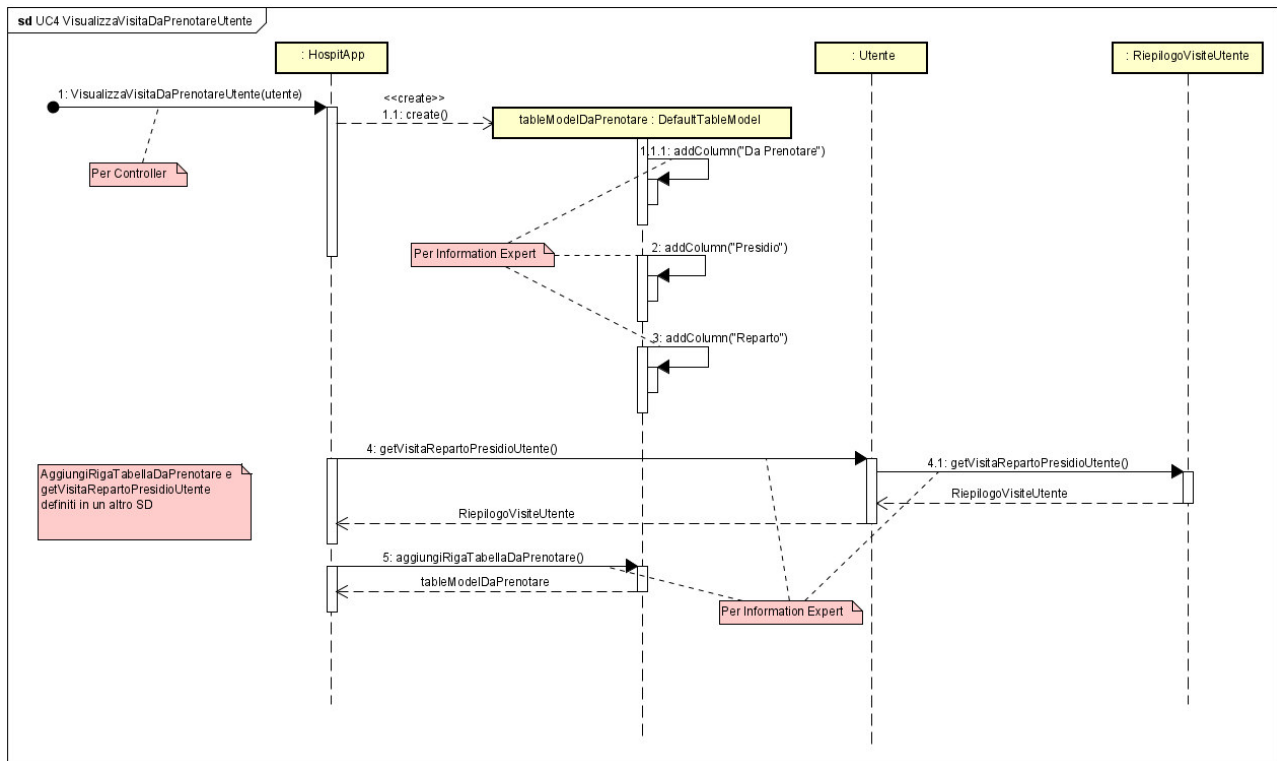
- Il metodo **getVisitaRepartoPresidioUtente()** ritorna la mappa in cui sono contenute le informazioni riguardanti la Visita, il Reparto e il Presidio della prenotazione.



Anche il metodo **AggiungiRigaTabellaPrenotata()** è un metodo definito in un altro SD per una questione di ordine. Non fa altro che aggiungere tutte le righe alla tabella già creata in modo da poter visualizzare correttamente rispettivamente la visita, il presidio e il reparto corrispondente (solo delle visite per cui sono stati settati un orario e un giorno dall'amministratore).



- **VisualizzaVisitaDaPrenotareUtente** permette di visualizzare anch'esso una tabella contenente tutte le visite (con le relative informazioni) per cui l'utente ha inviato una prenotazione per le quali l'amministratore non ha ancora settato un giorno e un orario specifico.



AggiungiRigaTabellaDaPrenotare(): è praticamente lo stesso metodo di AggiungiRigaTabellaPrenotata ma al posto di aggiungere le righe delle visite che sono state settate dall'amministratore, aggiunge le righe delle visite per cui l'amministratore non ha ancora settato un orario e una data.

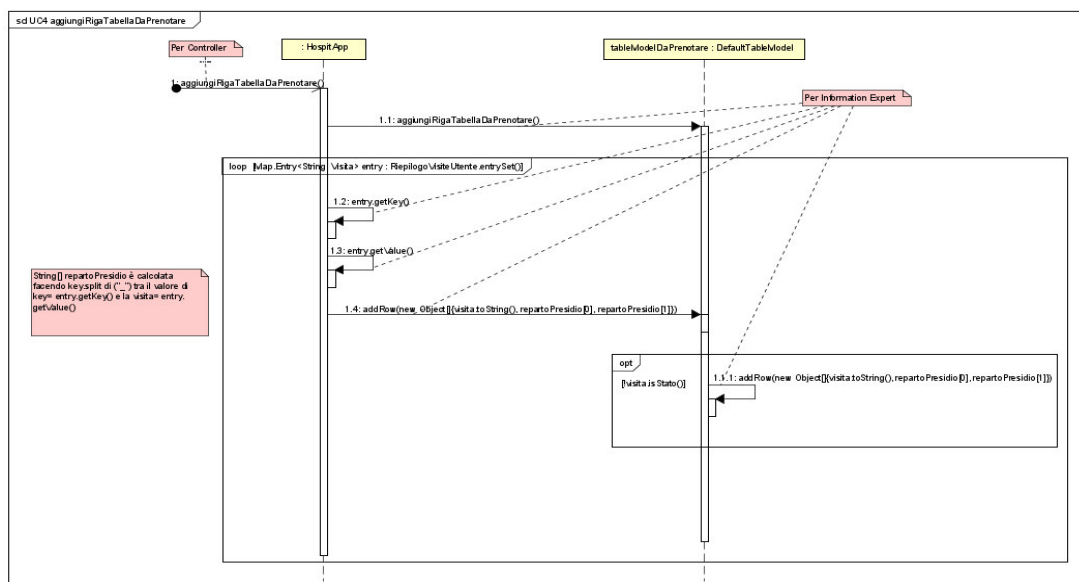
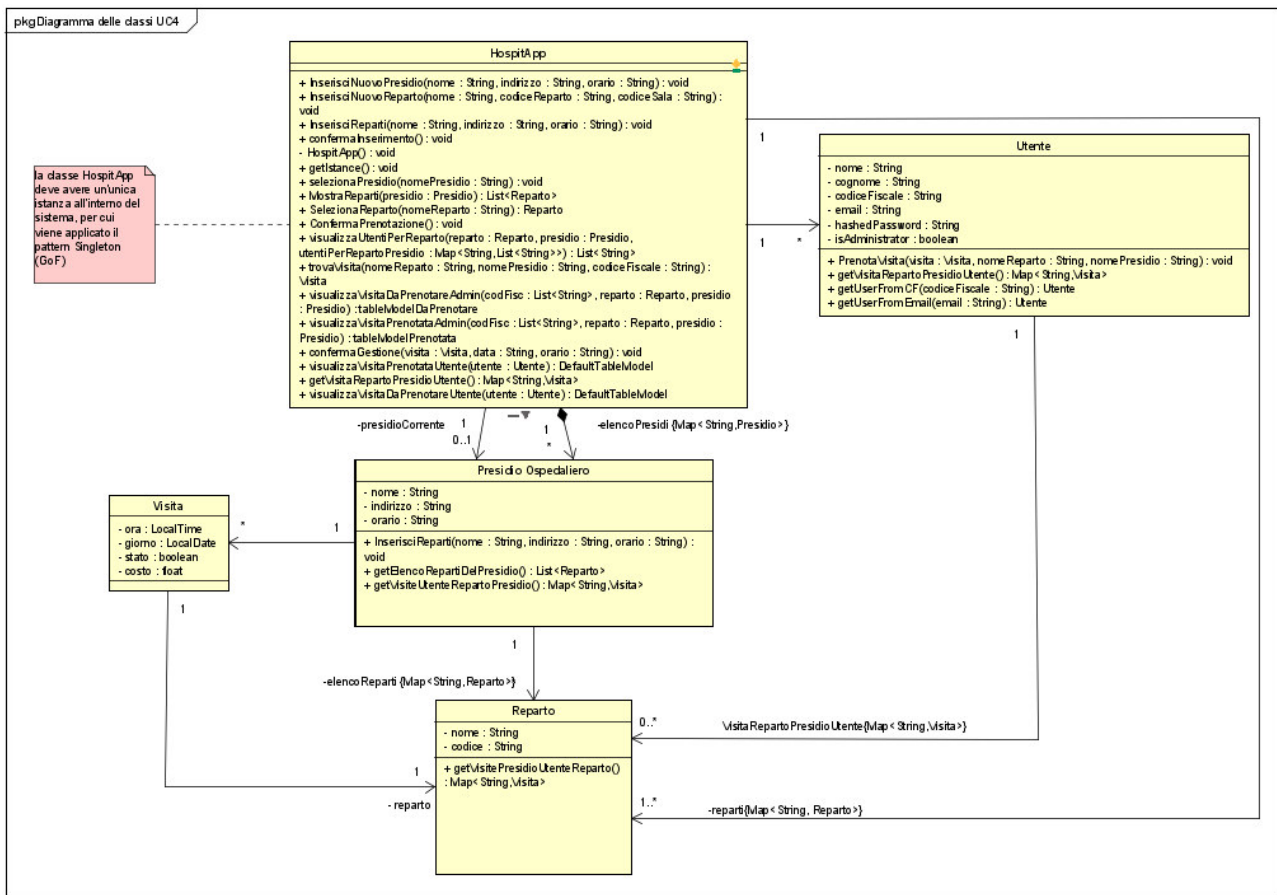


Diagramma delle classi UC4



Contratti delle operazioni UC4

1: Visualizza Visita Prenotata Utente

Operazione:

visualizzaVisitaPrenotataUtente (Utente utente)

Riferimenti:

Caso d'uso: Visualizza Visita Prenotata Utente;

Pre-condizioni:

-

Post-condizioni:

- È stata creato un oggetto di DefaultTabelModel;
- Le righe sono state riempite con i valori della Visita v, del Reparto r e del Presidio p associate a v;

2: Visualizza Visita Da Prenotare Utente

Operazione:

visualizzaVisitadaPrenotareUtente (Utente utente)

Riferimenti:

Caso d'uso: Visualizza Prenotazione Utente;

Pre-condizioni:

-

Post-condizioni:

- È stata creato un oggetto di DefaultTabelModel;
- Le righe sono state riempite con i valori della Visita v, del Reparto r e del Presidio p associate a v;