

Ingegneria del Software

Progetto OspiteSocial

Prof. Orazio Tomarchio

Anno Accademico: 2019/2020

Corso di Laurea Magistrale – Ingegneria Informatica

Impalà Antonio – Matricola 1000012375

Di Blasi Kristian – Matricola 1000012372

Germano Giuseppe – Matricola 1000008784

1. Ideazione ed Analisi dei Requisiti

1.1. Introduzione

In fase di realizzazione del software verranno individuati i servizi che si vogliono offrire tramite l'implementazione del software OspiteSocial. Quest'ultimo si propone come obiettivi: l'ottimizzazione della trattativa tra un ospitante ed un ospite durante il proprio viaggio. Verranno prodotti diversi documenti tra cui: documento di visione, diagramma e modello dei casi d'uso.

1.2. Requisiti

Si richiede la realizzazione di un applicativo il cui fine è migliorare la gestione della trattativa tra ospitanti e ospitati. Dopo un'accurata analisi preliminare dei requisiti, sono state individuate le seguenti funzionalità necessarie:

- L'applicativo deve gestire le iscrizioni delle utenze (ospitanti/ospitati), conservandone: i dati anagrafici, i quali devono essere consultabili, e le preferenze riguardanti la tipologia di ospitalità.
- Deve consentire l'aggiornamento o l'aggiunta di un servizio di ospitalità che l'utente vuole mettere a disposizione, relativamente al numero di utenti ospitabili e deve inoltre permettere di settare un periodo di tempo nel quale l'utente non accetti nessun ospite, senza doverne dare motivazione esplicita.
- Deve essere fruibile, a qualunque utente, un elenco dei possibili ospitanti, previa definizione di un filtro indicante il periodo di alloggio desiderato. Tale filtro permetterà di mostrare un elenco con tutti i possibili ospitanti che nel periodo specificato accettano ospiti (indicando anche il numero ospitabile). Di ogni soluzione mostrata l'utente può accedere ai dettagli.
- Scelto il piano di alloggio, il sistema permetterà la relativa prenotazione, indicando informazioni quali: date, persone, scelte dei posti letto e altro. Durante la procedura un utente può includere nel piano di alloggio degli accompagnatori, purché questi ultimi siano registrati alla piattaforma.
- L'ospitante, tramite la piattaforma, può accettare o rifiutare la richiesta di alloggio, indicando una motivazione in caso di rifiuto.
- Il sistema deve garantire la persistenza dei dati relativi a tali procedure.
- Procedure di feedback per indicare, con un intero da 0 a 5, il gradimento dell'ospitante nel caso dell'ospitato (obbligatorio), e dell'ospitato nel caso dell'ospitante (facoltativo).
- Il sistema deve prevedere una sezione che permetta ad un utente la visualizzazione di tutti i soggiorni effettuati dallo stesso.

- Il sistema deve implementare una politica a crediti: ogni giorno in cui un ospitato usufruisce del servizio l'ospitante otterrà dei crediti, contestualmente l'ospitato li cede. In fase di iscrizione, il sistema, assegna dei crediti all'utente, così da permettergli, preliminarmente, di usufruire del servizio.
- Il sistema deve permettere all'ospite di effettuare cambiamenti alla prenotazione entro un tempo scelto dall'ospitante. Questi cambiamenti verranno notificati all'ospitante tramite l'arrivo di una richiesta di aggiornamento.

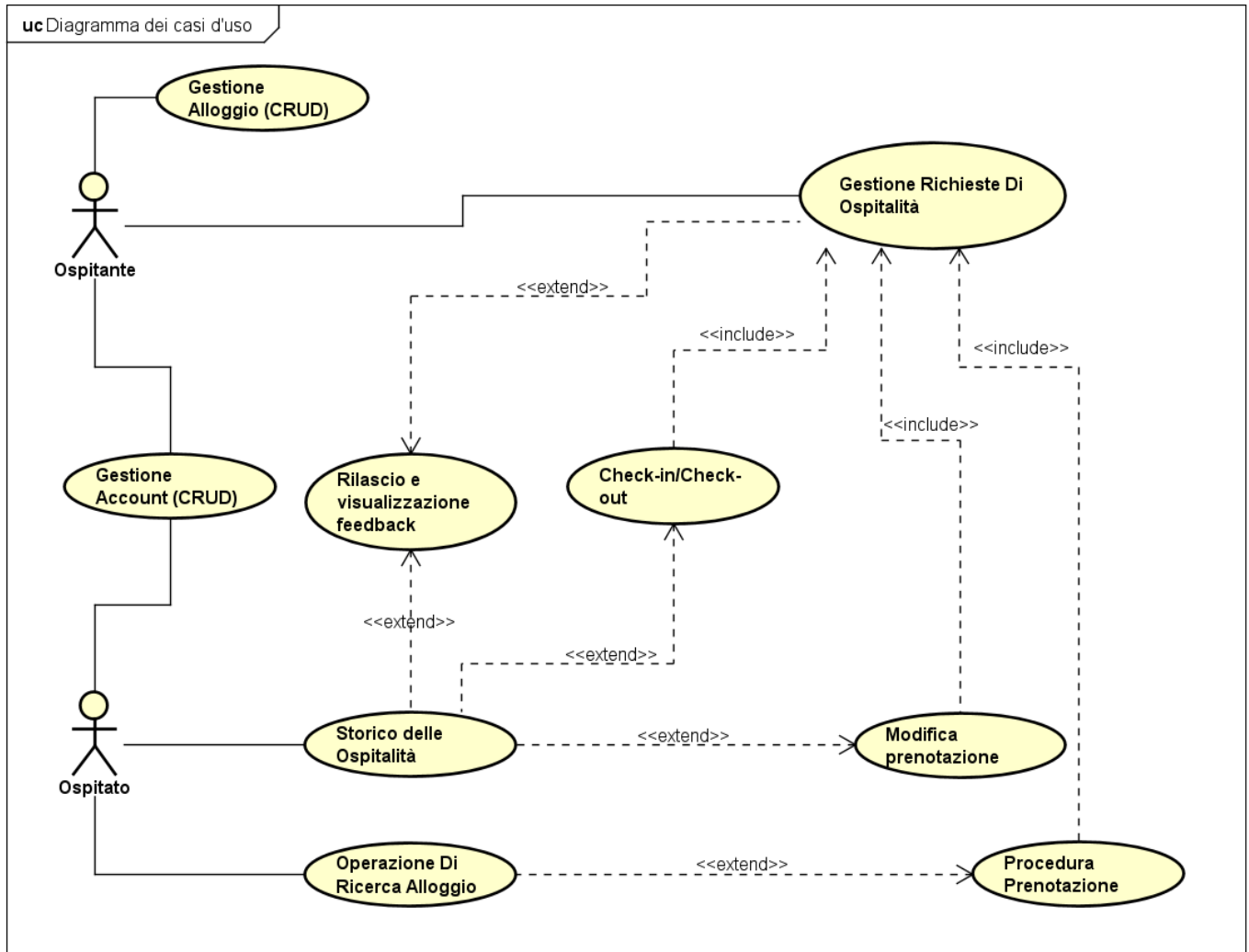
Dall'analisi dei requisiti sono stati individuati i seguenti casi d'uso:

1.3. Casi d'uso

ATTORE	OBIETTIVO	CASO D'USO
Ospitante/Ospitato	L'utente può creare, visualizzare, modificare o eliminare il proprio account.	UC1: Gestione Account (CRUD)
Ospitante	L'ospitante può definire le informazioni relative alla struttura d'alloggio, comprendendo una possibile eliminazione, visualizzazione o modifica dello stesso.	UC2: Gestione alloggio(CRUD)
Ospitato	L'ospitato può prenotare il soggiorno presso una struttura disponibile	UC3: Procedura prenotazione
Ospitato	L'ospitato può ricercare gli alloggi disponibili specificando la città, il periodo di permanenza e il numero di posti letto desiderati.	UC4: Operazione di ricerca alloggio
Ospitante	L'ospitante può accettare o rifiutare le richieste di ospitalità.	UC5: Gestione richieste di ospitalità

Ospitato	L'ospitato può visualizzare lo storico di tutte le prenotazioni in cui è coinvolto.	UC6: Storico delle ospitalità
Ospitante/Ospitato	L'ospitante può scegliere di valutare (con un intero compreso tra 0 e 5) l'ospitato. L'ospitato deve necessariamente valutare il servizio (con un intero compreso tra 0 e 5) dell'ospitante.	UC7: Rilascio feedback
Ospitato	L'ospitato può modificare i dati relativi ad una prenotazione entro il giorno di inizio soggiorno.	UC8: Modifica prenotazione
Ospitato	L'ospitato durante i propri soggiorni deve notificare la data effettiva di arrivo e partenza dall'abitazione	UC9: Check-in/Check-out

1.4. Diagramma dei casi d'uso



1.5. Modello dei casi d'uso

Caso d'uso UC1: Gestione Account (CRUD)	
Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Utente
Parti interessate e interessi	Utente: vuole poter iscriversi alla piattaforma per usufruire dei servizi offerti.
Pre-condizioni	Nessuna pre-condizione.
Garanzie di successo (post-condizioni)	L'utente viene registrato/modificato/ricercato/eliminato con successo nel sistema.
Scenario principale di successo (flusso base)	<p>Creazione account (C):</p> <ol style="list-style-type: none">1. L'utente richiede al sistema la registrazione di un nuovo account.2. Il sistema chiede all'utente di inserire i propri dati anagrafici: codice fiscale, nome, cognome, sesso, età, numero di telefono, città di residenza.3. L'utente inserisce i dati richiesti e conferma.4. Se i dati non sono presenti, il sistema acquisisce i dati immessi.5. Il sistema crea il nuovo account assegnando all'utente un numero iniziale di crediti (numero arbitrario definito dal sistema ed equivalente per ogni utente). <p>Visualizzazione account (R):</p> <ol style="list-style-type: none">1. Un utente richiede al sistema di visualizzare il proprio profilo.2. Il sistema ricerca le informazioni relative all'utente.3. Il sistema mostra le informazioni all'utente. <p>Modifica account (U):</p> <ol style="list-style-type: none">1. L'utente chiede la modifica dei dati immessi.2. Il sistema mostra i dati attuali con la possibilità di effettuare modifiche.3. L'utente modifica i dati desiderati.4. Il sistema verifica la validità dei dati e aggiorna quelli modificati.

	<p>Eliminazione account (D):</p> <ol style="list-style-type: none"> 1. L'utente richiede al sistema l'eliminazione del proprio profilo. 2. Il sistema chiede conferma dell'operazione. 3. L'utente conferma la procedura. 4. Il sistema rimuove il profilo utente.
Estensioni (scenari alternativi)	<p>I dati inseriti dall'utente sono già presenti nel sistema: (C)</p> <ol style="list-style-type: none"> 1. L'utente richiede al sistema di registrarsi. 2. Il sistema richiede all'utente di inserire i dati. 3. L'utente inserisce i propri dati e conferma. 4. Il sistema notifica all'utente che non è possibile effettuare la registrazione. <p>I nuovi dati immessi non sono validi: (U)</p> <ol style="list-style-type: none"> 1. Il sistema informa l'utente sulla non validità dei dati immessi, chiede di inserirli nuovamente.

Caso d'uso UC2: Gestione alloggio (CRUD)

Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Ospitante
Parti interessate e interessi	<ul style="list-style-type: none"> ▪ Ospitante: <ul style="list-style-type: none"> - Vuole che l'operazione di inserimento, modifica o rimozione dell'alloggio venga effettuata correttamente affinché riceva le richieste. - Vuole che i dati dell'alloggio da visualizzare siano corretti affinché possa valutare correttamente la modifica degli stessi. ▪ Ospitato: vuole che le informazioni riguardanti un alloggio siano corrette e complete in modo da effettuare una scelta mirata.
Pre-condizioni	L'ospitante deve appartenere al sistema.
Garanzie di successo (post-condizioni)	L'alloggio risulti disponibile al possibile impegno da parte di un ospitato.
Scenario principale di successo (flusso base)	<p>Inserimento alloggio (C) :</p> <ol style="list-style-type: none"> 1. L'ospitante chiede al sistema di poter inserire un nuovo alloggio. 2. Il sistema chiede all'ospitante l'inserimento dei dati: città, indirizzo, distanza dal centro città, distanza dalla stazione autobus/metro/treno più vicina e periodo di disponibilità alloggio. 3. L'ospitante inserisce i dati. 4. Il sistema richiede l'inserimento dei dati per le singole stanze: tipologia di posto letto (letto singolo, doppio o divano letto), se si tratta di stanza privata o comune (ad esempio divano letto in soggiorno) e in numero di posti letto della stanza. 5. L'ospitante inserisce i dati. 6. Il sistema verifica la validità dei dati. 7. Il sistema inserisce il nuovo alloggio. <p>Visualizza alloggio (R):</p> <ol style="list-style-type: none"> 1. L'ospitante chiede al sistema di visualizzare i dati relativi ad un alloggio. 2. Il sistema chiede all'utente di selezionare l'alloggio da visualizzare.

	<p>3. L'ospitante seleziona l'alloggio.</p> <p>4. Il sistema visualizza i dati dell'alloggio selezionato.</p> <p>Modifica alloggio (U):</p> <p>1. L'ospitante chiede al sistema di poter modificare i dati relativi ad un alloggio.</p> <p>2. Il sistema chiede all'utente di selezionare l'alloggio da modificare.</p> <p>3. L'ospitante seleziona l'alloggio.</p> <p>4. Il sistema chiede all'ospitante l'inserimento dei nuovi dati dell'alloggio.</p> <p>5. L'ospitante inserisce i dati.</p> <p>6. Il sistema verifica la validità dei dati e aggiorna le informazioni relative l'alloggio.</p> <p>Eliminazione alloggio (D):</p> <p>1. L'ospitante chiede al sistema di poter eliminare un alloggio.</p> <p>2. Il sistema chiede all'utente di selezionare l'alloggio da eliminare.</p> <p>3. L'ospitante seleziona l'alloggio.</p> <p>4. Il sistema chiede all'ospitante la conferma di eliminazione</p> <p>5. L'ospitante conferma.</p> <p>6. Il sistema elimina l'alloggio.</p>
Estensioni (scenari alternativi)	<p>L'alloggio inserito dall'ospitante risulta già memorizzato (C/U):</p> <p>1. L'ospitante inserisce l'alloggio</p> <p>2. Il sistema blocca l'inserimento avvisando l'ospitante di tale insuccesso.</p>
Requisiti speciali	<p>Modifica: l'aggiornamento delle informazioni sulla piattaforma deve avvenire entro 5 minuti dalla richiesta.</p>
Frequenza di ripetizione	<p>Legata al numero di alloggi da inserire sulla piattaforma</p>
Varie	<p>L'ospitante potrebbe non visualizzare l'alloggio o visualizzare informazioni non aggiornate.</p>

Caso d'uso UC3: Procedura prenotazione

Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Ospitato
Parti interessate e interessi	<ul style="list-style-type: none"> ▪ Ospitante: vuole che l'operazione di prenotazione dell'alloggio venga effettuata correttamente affinché arrivi la richiesta. ▪ Ospitato: vuole che l'operazione di prenotazione vada a buon fine affinché possa soggiornare durante il proprio viaggio.
Pre-condizioni	<ul style="list-style-type: none"> ▪ Tutti gli ospitati devono appartenere al sistema. ▪ L'alloggio deve essere prenotabile relativamente al periodo richiesto dall'ospitato. ▪ Devono essere disponibili sufficienti posti letto per tutti gli ospiti della prenotazione. ▪ L'utente deve aver rilasciato i feedback relativi alle abitazioni dei soggiorni effettuati. ▪ L'utente deve possedere un saldo crediti positivo.
Garanzie di successo (post-condizioni)	<ul style="list-style-type: none"> ▪ La prenotazione dell'alloggio avviene con successo, con relativa richiesta all'ospitante. ▪ L'alloggio non è più presente in una ricerca contenente lo stesso periodo di tempo.
Scenario principale di successo (flusso base)	<p>Prenotazione alloggio:</p> <ol style="list-style-type: none"> 1. L'ospitato sceglie il piano d'alloggio, aggiungendo eventualmente anche gli altri ospiti che soggiornano insieme a lui. 2. Il sistema effettua la prenotazione. 3. Il sistema invia la richiesta all'ospitante e lo notifica via email.
Estensioni (scenari alternativi)	Nessuno.
Requisiti speciali	<ul style="list-style-type: none"> ▪ Tempi di prenotazioni brevi
Frequenza di ripetizione	<ul style="list-style-type: none"> ▪ Legata al numero di prenotazioni degli ospitati

Caso d'uso UC4: Operazione di ricerca alloggio

Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Ospitato
Parti interessate e interessi	<ul style="list-style-type: none"> ▪ Ospitante: vuole che l'operazione di ricerca dell'alloggio venga effettuata correttamente affinché la propria abitazione venga rilevata. ▪ Ospitato: vuole che l'operazione di ricerca sia semplice, intuitiva ed efficiente affinché possa trovare una soluzione di soggiorno adatta alle proprie esigenze.
Pre-condizioni	L'ospitato deve appartenere al sistema.
Garanzie di successo (post-condizioni)	Il sistema mostra all'utente i risultati della ricerca.
Scenario principale di successo (flusso base)	Ricerca alloggio: <ol style="list-style-type: none"> 1. L'ospitato inserisce i dati relativi ai filtri in uso: data di inizio soggiorno, data di fine soggiorno, città, numero di posti letto liberi. 2. Il sistema prende in carico la richiesta e mostra gli alloggi che presentano le proprietà richieste.
Estensioni (scenari alternativi)	Non sono presenti alloggi che soddisfano le richieste della ricerca: <ol style="list-style-type: none"> 1. Il sistema notifica l'assenza di alloggi che rispettano le richieste.
Requisiti speciali	<ul style="list-style-type: none"> ▪ Tempi di ricerca brevi. ▪ Interfaccia grafica intuitiva.
Frequenza di ripetizione	<ul style="list-style-type: none"> ▪ Legata al numero di ricerche effettuate dagli utenti che cercano alloggio.

Caso d'uso UC5: Gestione richieste di ospitalità

Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Ospitante
Parti interessate e interessi	<ul style="list-style-type: none"> ▪ Ospitante: vuole essere informato costantemente e in tempo reale di tutte le richieste e le notifiche relative ai propri alloggi. ▪ Ospitato: vuole che la richiesta venga visionata dall'ospitante e che la sua decisione a riguardo venga notificata nel più breve tempo possibile.
Pre-condizioni	L'ospitante deve appartenere al sistema.
Garanzie di successo (post-condizioni)	<ul style="list-style-type: none"> ▪ All'utente che ha fatto la prenotazione viene notificata la decisione dell'ospitante. ▪ L'ospitante riceve notifiche riguardo le varie funzionalità del sistema.
Scenario principale di successo (flusso base)	<p>Visualizzazione richieste e prenotazioni (B):</p> <ol style="list-style-type: none"> 1. L'ospitante interroga il sistema per ottenere l'elenco delle richieste di ospitalità e delle prenotazioni. 2. Il sistema mostra l'elenco richiesto, eventualmente vuoto. <p>Accettazione richiesta di ospitalità (A):</p> <ol style="list-style-type: none"> 1. L'ospitante sceglie la richiesta dall'elenco visualizzato e la accetta. 2. Il sistema pone lo stato della prenotazione in "Confermata", notificando all'utente l'accettazione della richiesta. <p>Rifiuto richiesta di ospitalità (A):</p> <ol style="list-style-type: none"> 1. L'ospitante sceglie la richiesta dall'elenco visualizzato e la rifiuta, specificando il motivo di tale rifiuto. 2. Il sistema pone lo stato della prenotazione in "Rifiutata", notificando all'utente il rifiuto della richiesta e riportando il motivo associato.
Estensioni (scenari alternativi)	Nessuna.
Requisiti speciali	Interfaccia di gestione semplice e intuitiva.

Frequenza di ripetizione	Legata al numero di richieste ricevute e di visualizzazioni richieste dall'utente.
---------------------------------	--

Caso d'uso UC6: Storico delle ospitalità

Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Ospitato
Parti interessate e interessi	Ospitato: vuole poter risalire a tutte le prenotazioni che lo interessano.
Pre-condizioni	-L'ospitato deve aver effettuato delle prenotazioni. -L'ospitato deve far parte del sistema.
Garanzie di successo (post-condizioni)	L'ospitato visualizza la lista delle prenotazioni che lo interessano.
Scenario principale di successo (flusso base)	Visualizza prenotazioni (A): 6. L'ospitato richiede al sistema di visualizzare le prenotazioni. 7. Il sistema mostra tutte le prenotazioni in cui è presente l'utente come ospite.
Estensioni (scenari alternativi)	Nessuno scenario alternativo.
Requisiti speciali	<ul style="list-style-type: none">▪ Interfaccia grafica semplice ed intuitiva.▪ Il sistema presenta tempi di risposta brevi ed accettabili.
Frequenza di ripetizione	<ul style="list-style-type: none">▪ Legata al numero di visualizzazioni richieste dagli utenti.

Caso d'uso UC7: Rilascio e visualizzazione feedback

Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Utente
Parti interessate e interessi	<ul style="list-style-type: none"> ▪ Ospitante: Vuole inserire una valutazione complessiva sul comportamento dell'utente ospitato. ▪ Ospitato: Deve inserire una valutazione complessiva sul servizio offertogli.
Pre-condizioni	<ul style="list-style-type: none"> -L'ospitato deve appartenere al sistema. -L'ospitato deve aver effettuato e concluso un soggiorno. -L'ospitato deve aver effettuato il check-out. -L'ospitante deve appartenere al sistema. -L'ospitante deve aver ospitato in una sua abitazione.
Garanzie di successo (post-condizioni)	Il feedback è visibile agli utenti del sistema.
Scenario principale di successo (flusso base)	<p>Rilascio feedback per l'ospitante (A):</p> <ol style="list-style-type: none"> 7. L'ospitato chiede al sistema di poter rilasciare un feedback relativo al soggiorno effettuato. 8. Il sistema chiede all'utente di inserire il valore della valutazione. 9. L'ospitato inserisce il valore. 10. Il sistema verifica la validità dei dati e registra il feedback associandolo all'abitazione. <p>Rilascio feedback per l'ospitato (B):</p> <ol style="list-style-type: none"> 3. L'ospitante chiede al sistema di poter rilasciare un feedback relativo all'utente ospitato. 4. Il sistema chiede all'utente di inserire il valore della valutazione. 5. L'ospitante inserisce il valore. <p>Il sistema verifica la validità dei dati e registra il feedback associandolo all'ospitato.</p> <p>Visualizzazione feedback abitazione (A):</p>

	<ol style="list-style-type: none"> 1. L'utente chiede al sistema di poter visualizzare i feedback relativi ad una abitazione. 2. Il sistema calcola e restituisce una media dei feedback rilasciati dagli ospitati per l'abitazione richiesta. <p>Visualizzazione feedback ospitato (B):</p> <ol style="list-style-type: none"> 1. L'ospitante chiede di poter visualizzare i feedback relativi ad un utente che ha richiesto alloggio. 2. Il sistema calcola e restituisce una media dei feedback rilasciati dagli ospitanti all'utente.
Estensioni (scenari alternativi)	Nessuna.
Requisiti speciali	Nessun requisito speciale.
Frequenza di ripetizione	Legata al numero di visualizzazione dei feedback e numero di prenotazioni.

Caso d'uso UC8: Modifica prenotazione

Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Ospitato
Parti interessate e interessi	<ul style="list-style-type: none"> ▪ Ospitante: vuole che l'operazione di prenotazione dell'alloggio venga effettuata correttamente affinché arrivi la richiesta. ▪ Ospitato: vuole che l'operazione di prenotazione vada a buon fine affinché possa soggiornare durante il proprio viaggio.
Pre-condizioni	<ul style="list-style-type: none"> ▪ L'alloggio deve essere prenotabile relativamente al periodo richiesto dall'ospitato. ▪ Devono essere disponibili sufficienti posti letto per tutti gli ospiti della prenotazione. ▪ La modifica deve essere effettuata solo da utenti presenti nella lista degli ospiti. ▪ La prenotazione deve trovarsi in stato "Da confermare" o "Confermata".
Garanzie di successo (post-condizioni)	<ul style="list-style-type: none"> ▪ La modifica della prenotazione avviene con successo, con relativa richiesta all'ospitante. ▪ L'alloggio non è più presente in una ricerca contenente lo stesso periodo di tempo.
Scenario principale di successo (flusso base)	<p>Modifica Prenotazione soggiorno:</p> <ol style="list-style-type: none"> 1. L'ospitato richiede al sistema la modifica di una determinata prenotazione. 2. Il sistema verifica lo stato e richiede all'utente di inserire le modifiche: data inizio e fine soggiorno. 3. L'utente inserisce le relative modifiche. 4. Il sistema pone lo stato della prenotazione in "Da confermare", infine invia la nuova richiesta all'ospitante e lo notifica via email. <p>Annullamento prenotazione:</p> <ol style="list-style-type: none"> 1. L'utente richiede al sistema l'annullamento di una relativa prenotazione.

	2. Il sistema prende in carico l'annullamento ponendo lo stato della prenotazione a "Annullata". Infine notifica l'ospitante di tale annullamento tramite invio email.
Estensioni (scenari alternativi)	Nessuno.
Requisiti speciali	<ul style="list-style-type: none"> ▪ Tempi di prenotazioni brevi
Frequenza di ripetizione	<ul style="list-style-type: none"> ▪ Legata al numero di prenotazioni degli ospitati

Caso d'uso UC9: Check-in/Check-out

Portata	Piattaforma OspiteSocial
Livello	Obiettivo utente
Attore primario	Ospitato
Parti interessate e interessi	<ul style="list-style-type: none"> ▪ Ospitante: vuole sapere se l'utente ospitato ha effettivamente iniziato/concluso il soggiorno. ▪ Ospitato: deve poter indicare data e ora di arrivo e partenza.
Pre-condizioni	<ul style="list-style-type: none"> -L'ospitato deve appartenere al sistema. -L'ospitato deve aver effettuato una prenotazione. -L'ospitante deve aver accettato la prenotazione.
Garanzie di successo (post-condizioni)	Il sistema ha aggiornato i dati della prenotazione.
Scenario principale di successo (flusso base)	<p>Check-in (I):</p> <ol style="list-style-type: none"> 1. Gli ospitati, appena arrivano nell'abitazione del soggiorno, richiedono al sistema di effettuare il check-in. 2. Il sistema aggiorna i dati della prenotazione registrando la data di arrivo effettiva e impostando lo stato della prenotazione in "In corso". 3. Il sistema calcola i crediti che ogni ospite dovrà versare per il soggiorno. 4. Il sistema avvia la transazione dei crediti, indirizzati dagli ospiti verso l'ospitante. <p>Check-out (O):</p> <ol style="list-style-type: none"> 1. Gli ospitati, appena termina il soggiorno, richiedono al sistema di effettuare il check-out. 2. Il sistema aggiorna i dati della prenotazione registrando la data di fine alloggio effettiva e impostando lo stato della prenotazione in "Terminata".
Estensioni (scenari alternativi)	<p>Crediti insufficienti:</p> <ol style="list-style-type: none"> 1. Gli ospitati/o non possiedono abbastanza crediti da saldare il soggiorno. 2. Il sistema detrae i crediti, portando il saldo in negativo. (il saldo in negativo impedirà all'utente future prenotazioni).

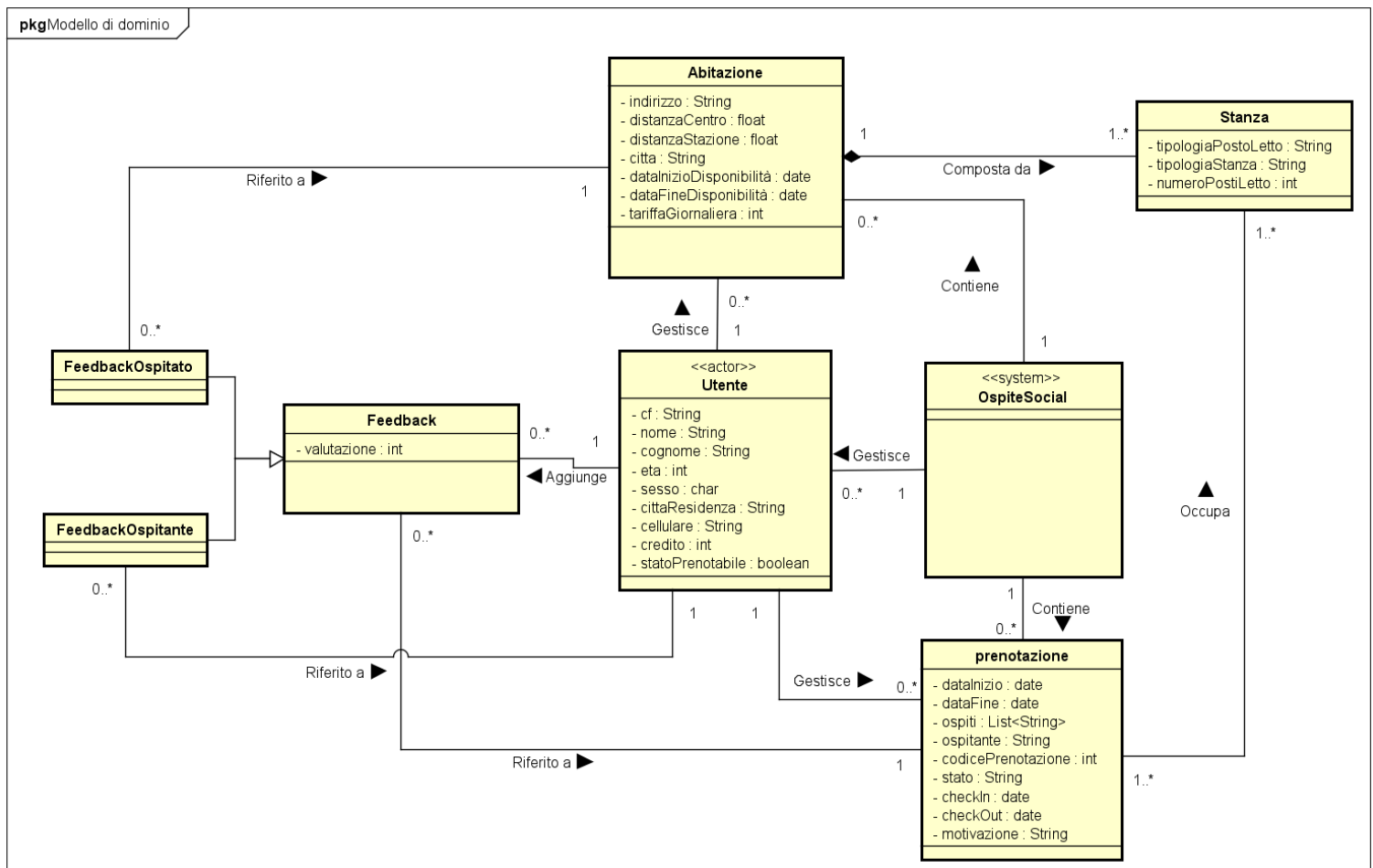
	<p>Mancato check-in (A):</p> <ol style="list-style-type: none"> 1. Il sistema verifica allo scadere di un giorno (12:00 AM), che tutti i check-in previsti, relativi al giorno appena terminato, siano stati effettuati. 2. Nel caso in cui uno o più check-in non sono stati effettuati, il sistema annullerà la prenotazione, cambiando lo stato da “Confermata” ad “Annullata”. Tale procedura renderà le abitazioni prenotabili da nuovi utenti. <p>Check-out anticipato (O):</p> <ol style="list-style-type: none"> 1. L’utente/i richiede di terminare il soggiorno in anticipo, effettuando il check-out. 2. Il sistema termina il soggiorno e rende l’abitazione nuovamente prenotabile.
Requisiti speciali	<ul style="list-style-type: none"> ▪ Tempi di registrazione/aggiornamento brevi. ▪ Interfaccia grafica intuitiva.
Frequenza di ripetizione	<ul style="list-style-type: none"> ▪ Legata al numero di soggiorni.

GLOSSARIO:

- **Soggiorno:** richiesta di ospitalità di un ospitato, accettata dall'ospitante, con avvenuta terminazione del periodo di ospitalità.
- **Ospitante:** utente che ospita un altro utente nel proprio alloggio.
- **Ospitato:** utente soggiornante in un alloggio.
- **Richiesta:** interesse di un utente ad essere ospitato in un determinato alloggio.
- **Feedback:** valutazioni riguardo il gradimento di un servizio, basati su una scala intera estesa da 0 a 5.
- **Credito:** moneta virtuale utilizzata dal sistema per lo scambio del servizio di ospitalità fra utenti del sistema.
- **Check-in:** notifica per l'ospitante dell'arrivo dell'ospite nella struttura.
- **Check-out:** notifica per l'ospitante della partenza dell'ospite dalla struttura.

2. Fase di elaborazione

2.1. Modello di dominio

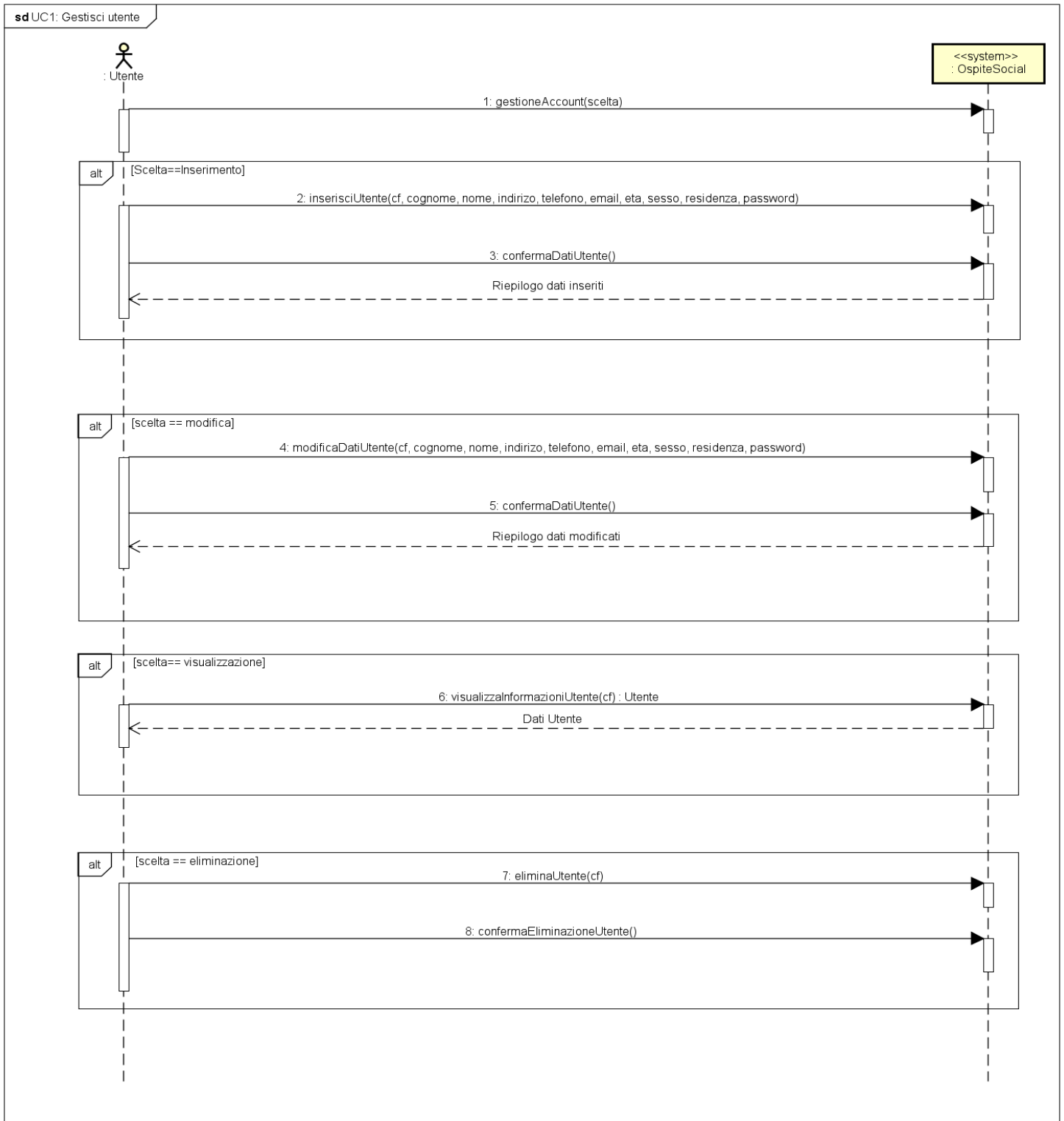


Sono state individuate le seguenti classi concettuali:

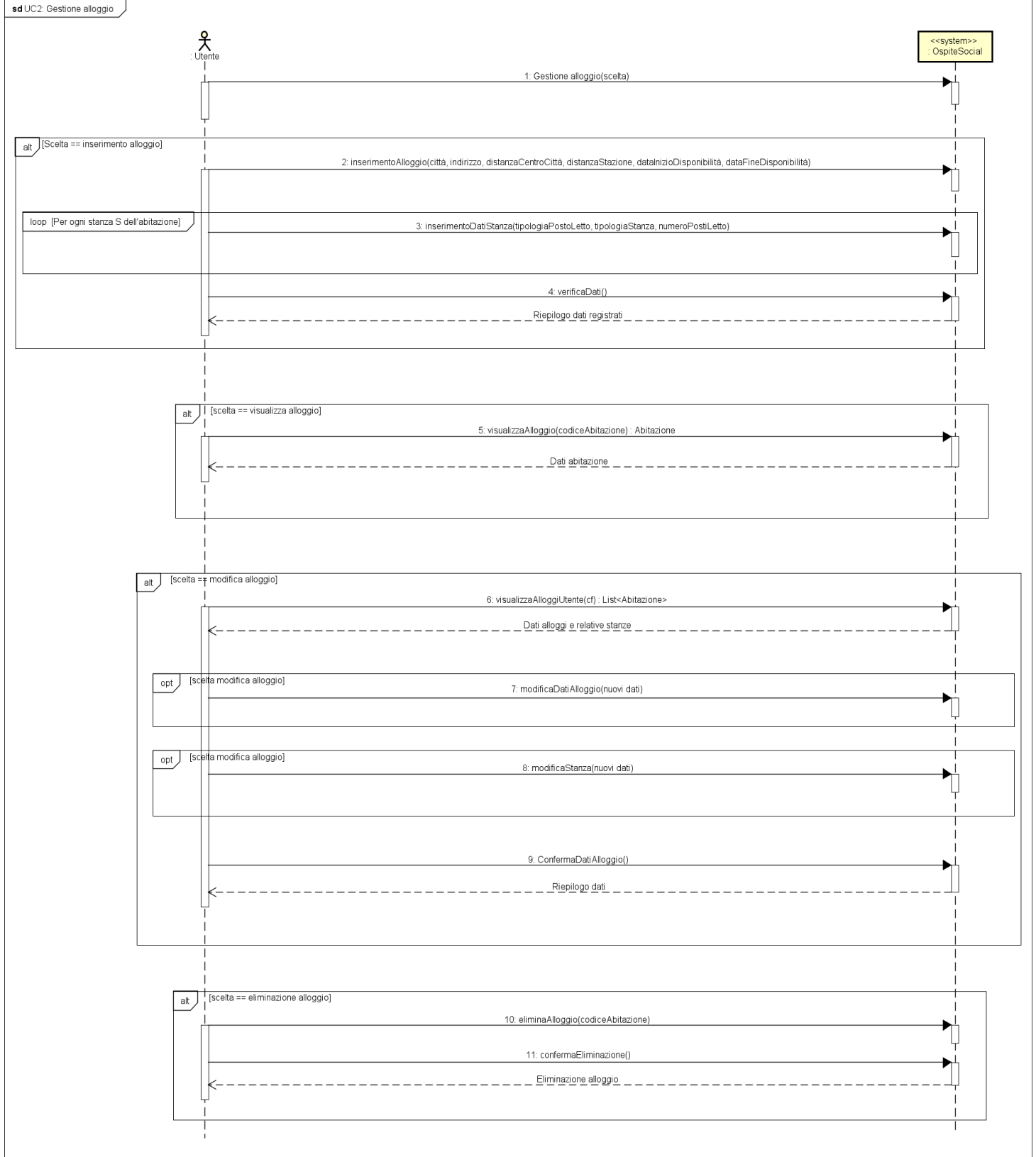
- **OspiteSocial**: rappresenta l'intero sistema
- **Utente**: rappresenta l'utilizzatore del sistema ed i dati ad esso correlati
- **Abitazione**: rappresenta la struttura dove è possibile effettuare un soggiorno ed i dati ad essa correlati
- **Stanza**: compone l'abitazione e contiene i dati relativi al numero di utenti ospitabili
- **Prenotazione**: rappresenta la richiesta, da parte di un utente ospite, di essere ospitato da un utente ospitante; con i vari dati ad essa correlati.
- **Feedback**: rappresenta il giudizio sulla qualità del servizio da parte di ospiti e ospitanti

2.2. Diagramma di sequenza di sistema (SSD)

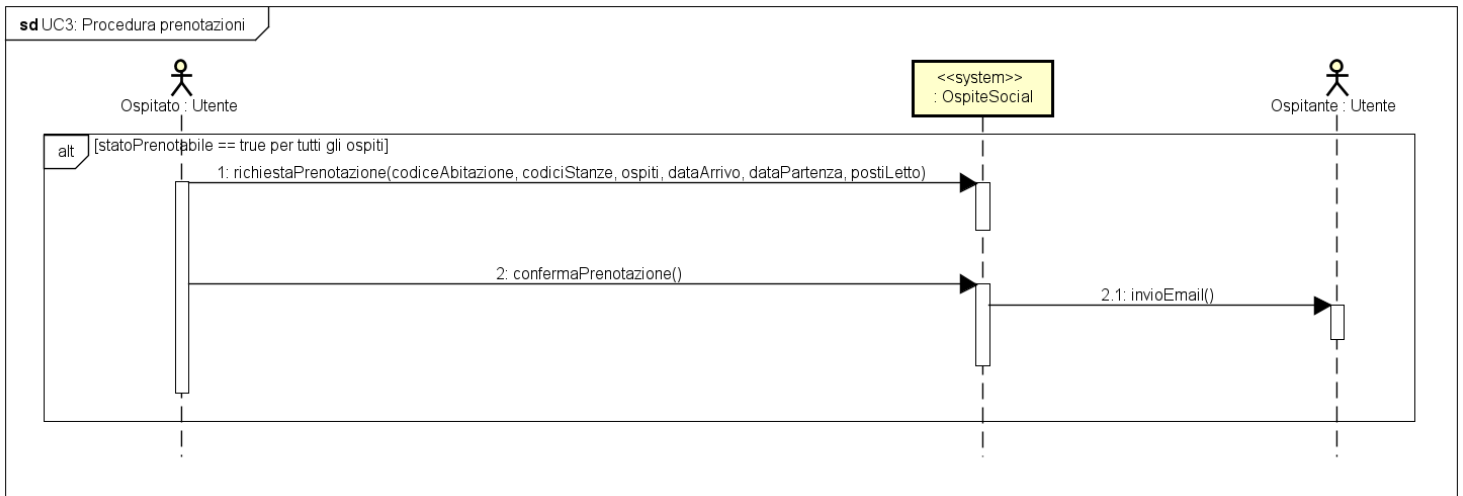
UC1: Gestione utente (CRUD)



UC2: Gestione alloggio

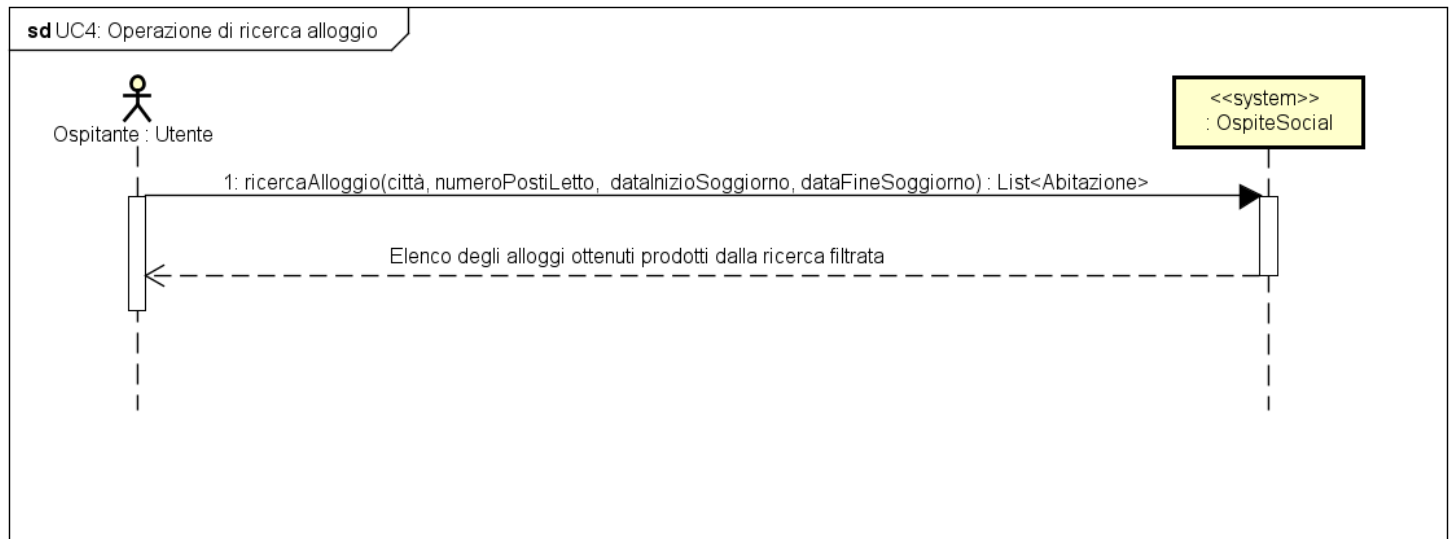


UC3: Procedura prenotazioni

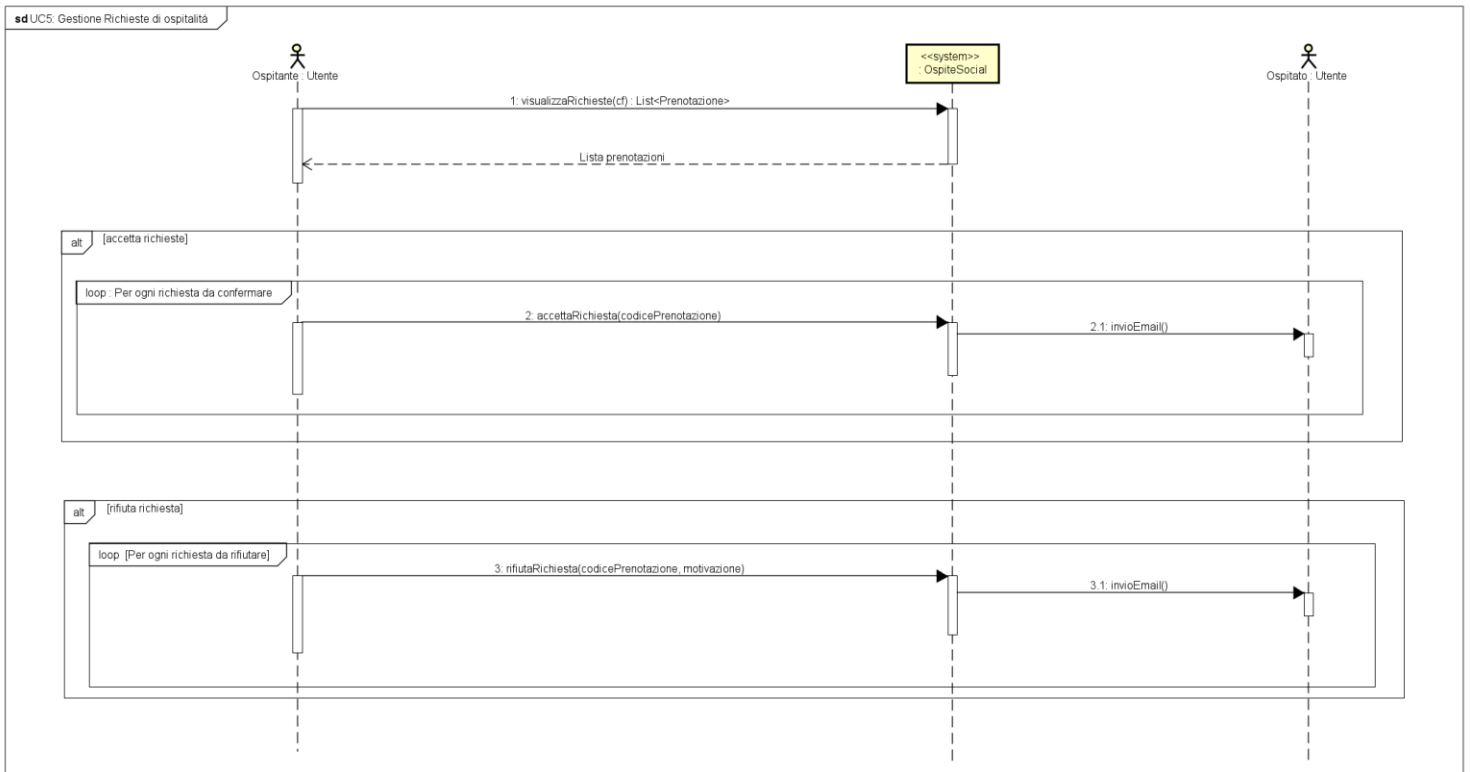


Nella procedura di prenotazione è stato inserito un controllo sullo stato dell'utente ospitato affinché quest'ultimo non possa effettuare prenotazioni se non ha rilasciato tutti i feedback relativi ai soggiorni effettuati.

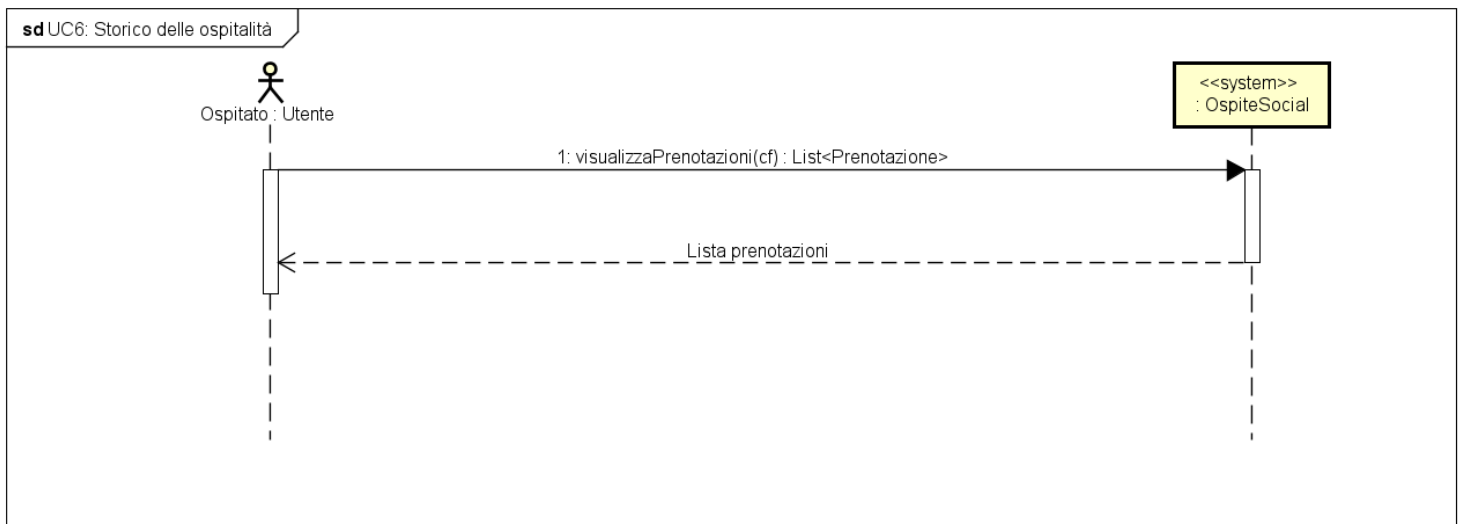
UC4: Operazione di ricerca alloggio



UC5: Gestione richieste di ospitalità



UC6: Storico delle ospitalità



UC7A: Rilascio e visualizzazione feedback



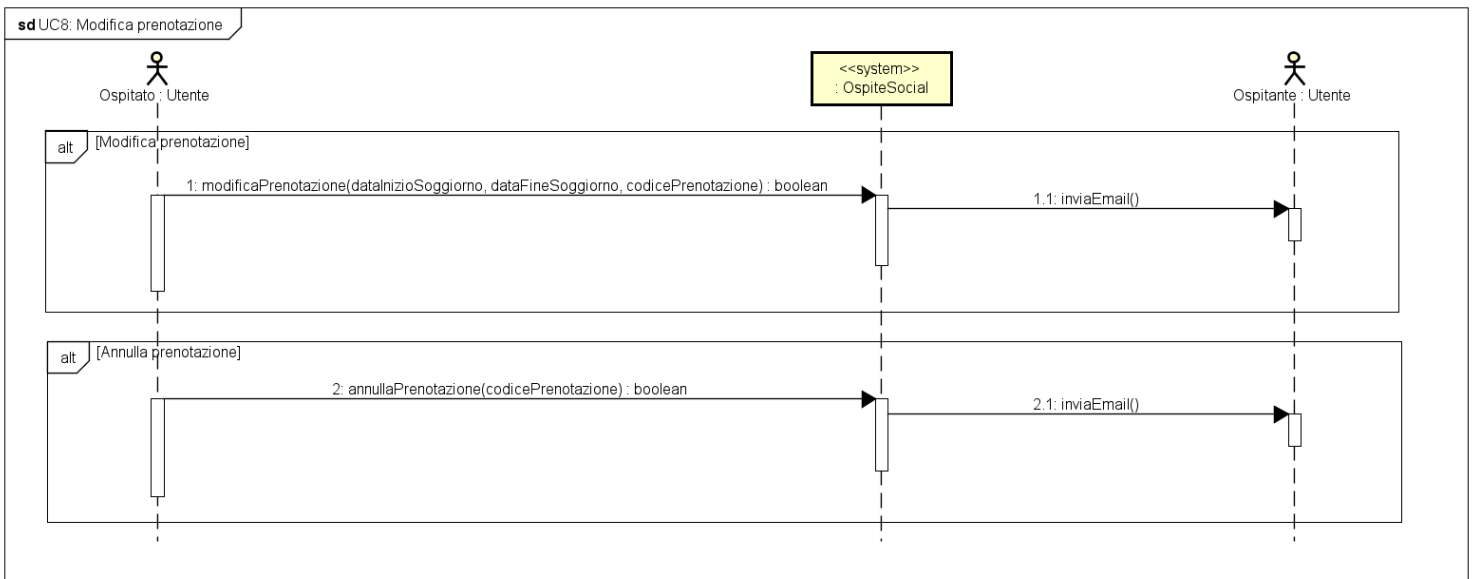
La media dei feedback rilasciati sull'abitazione saranno visualizzati nel contesto della prenotazione, contestualmente ai dettagli relativi all'abitazione.

UC7B: Rilascio e visualizzazione feedback



La visualizzazione dei feedback relativi all'utente che chiede ospitalità saranno visualizzati nel contesto della gestione delle richieste, dove l'ospitante ha la possibilità di accettare o rifiutare tali richieste.

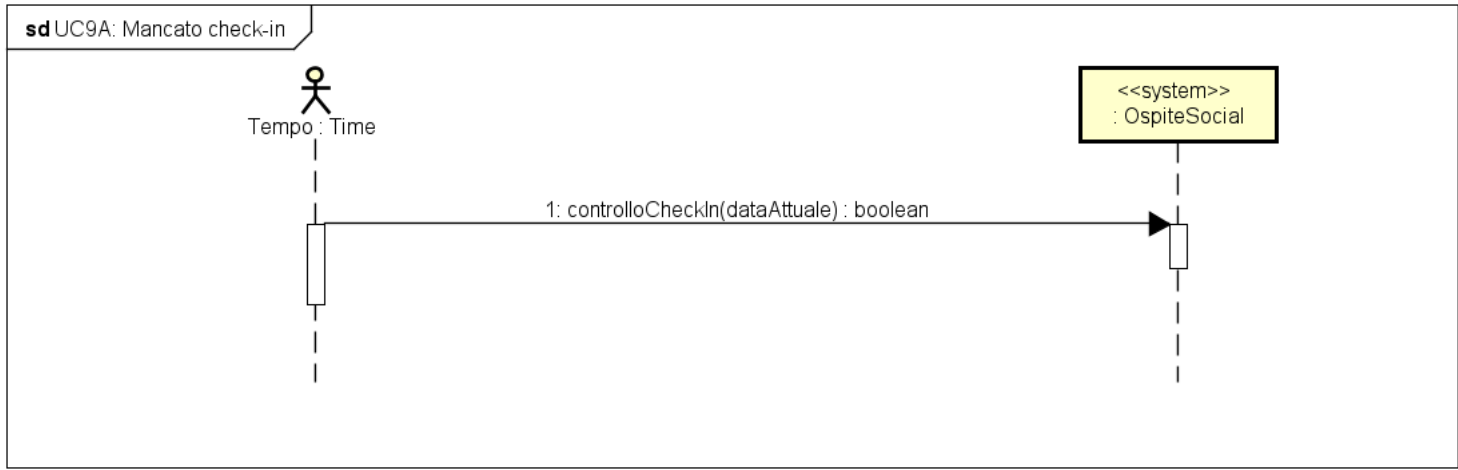
UC8: Modifica prenotazione



UC9: Check-in/Check-Out



UC9: Mancato check-in



2.3. Contratti delle operazioni

Dagli SSD realizzati possiamo ora definire le seguenti operazioni. Per semplicità sono stati sviluppati solo un numero ridotto di contratti delle operazioni.

Contratti operazioni UC3

Contratto C01: RichiestaPrenotazione

Operazione	richiestaPrenotazione(codiceAbitazione, numeroPostiLetto, dataArrivo, dataPartenza)
Riferimenti	Caso d'uso UC3 Procedura Prenotazione.
Pre-condizioni	<ul style="list-style-type: none">-Tutti gli ospitati presenti nella prenotazione devono essere registrati nel sistema.-L'alloggio desiderato deve essere prenotabile.-Tutti gli ospitati devono avere sufficienti crediti per la prenotazione.-Devono essere disponibili sufficienti posti letto per tutti gli ospiti della prenotazione.-L'utente deve aver rilasciato i feedback relativi alle abitazioni dei soggiorni effettuati.-L'utente deve possedere un saldo crediti positivo.
Post-condizione	<ul style="list-style-type: none">-E' stata notificata la richiesta all'ospitante.-L'alloggio è stato reso non prenotabile se è al completo per il determinato periodo di alloggio.-Lo stato della richiesta è impostato su "Da confermare"

Contratti operazioni UC5

Contratto C02: AccettaRichiesta

Operazione	accettaRichiesta(codicePrenotazione)
Riferimenti	Caso d'uso UC5 Gestione richieste di ospitalità
Pre-condizioni	<ul style="list-style-type: none">-L'ospitante deve essere registrato nel sistema.-L'alloggio deve essere disponibile per le date di soggiorno.-Lo stato della richiesta deve essere "Da confermare".
Post-condizione	<ul style="list-style-type: none">-E' stato notificato all'ospitato che l'ospitante ha accettato il soggiorno.-Lo stato della richiesta passa da "Da confermare" a "Confermata".

Contratto C03: RifiutaRichiesta

Operazione	rifiutaRichiesta(codicePrenotazione, motivazione)
Riferimenti	Caso d'uso UC5 Gestione richieste di ospitalità
Pre-condizioni	<ul style="list-style-type: none">-L'ospitante deve essere registrato nel sistema.-Lo stato della richiesta deve essere "Da confermare".
Post-condizione	<ul style="list-style-type: none">-E' stato notificato all'ospitato che l'ospitante ha rifiutato il soggiorno, allegando la motivazione del rifiuto.-L'alloggio è stato reso prenotabile.-Lo stato della richiesta passa da "Da confermare" a "Rifiutata".

Contratti operazioni UC7

Contratto C01: AggiungiFeedbackOspitante

Operazione	aggiungiFeedbackOspitante (codicePrenotazione, valore, autore, ospite)
Riferimenti	Caso d'uso UC7 Rilascio e visualizzazione feedback
Pre-condizioni	-L'ospitante deve essere registrato nel sistema. -L'ospitato deve aver terminato il soggiorno.
Post-condizione	Il feedback è stato memorizzato nel sistema.

Contratto C01: AggiungiFeedbackOspitato

Operazione	aggiungiFeedbackOspitato(codicePrenotazione, valore, autore, ospite, codiceAbitazione)
Riferimenti	Caso d'uso UC7 Rilascio e visualizzazione feedback
Pre-condizioni	-L'ospitante deve essere registrato nel sistema. -L'ospitato deve aver terminato il soggiorno.
Post-condizione	-Il feedback è stato memorizzato nel sistema. -L'attributo statoPrenotabile dell'ospitato è stato impostato a "true".

Contratti operazioni UC8

Contratto C02: annullaPrenotazione

Operazione	annullaPrenotazione(codicePrenotazione)
Riferimenti	Caso d'uso UC8 Modifica prenotazione
Pre-condizioni	-L'ospite deve essere registrato nel sistema. -L'utente che effettua l'annullamento deve essere un ospite della prenotazione. -Lo stato della prenotazione deve essere "Da confermare" o "Confermata".
Post-condizione	-E' stato notificato all'ospitante che l'ospitato ha annullato il soggiorno. -L'alloggio è stato reso prenotabile. -Lo stato della prenotazione passa da "Da confermare"/"Confermata" a "Rifiutata".

Contratto C01: modificaPrenotazione

Operazione	modificaPrenotazione(codicePrenotazione, dataInizioSoggiorno, dataFineSoggiorno)
Riferimenti	Caso d'uso UC8 Modifica prenotazione
Pre-condizioni	-L'ospitato deve essere registrato nel sistema. -L'utente che effettua la modifica deve essere un ospite della prenotazione. -Lo stato della prenotazione deve essere "Da confermare" o "Confermata".
Post-condizione	La prenotazione è stata aggiornata apportando le modifiche richieste dall'ospitato.

Contratti operazioni UC9

Modifica Contratto C01: *checkIn*

Operazione	checkIn(codicePrenotazione, data)
Riferimenti	Caso d'uso UC9 Check-in/Check-out
Pre-condizioni	<ul style="list-style-type: none">-L'ospitante deve essere registrato nel sistema.-L'ospitato/i deve aver prenotato il soggiorno.-L'ospitante deve aver accettato la richiesta.-Il giorno di check-in deve coincidere con il giorno di inizio soggiorno.
Post-condizione	<ul style="list-style-type: none">-E' stata memorizzata la data esatta di arrivo degli ospitati/o.-I crediti sono stati trasferiti dagli ospiti/e all'ospitante.-Lo stato della prenotazione è stato impostato "In corso".-L'attributo statoPrenotabile dell'utente, che effettua il check-in, è stato impostato a "False".

Modifica Contratto C02: *checkOut*

Operazione	checkOut(codicePrenotazione, data)
Riferimenti	Caso d'uso UC9 Check-in/Check-out
Pre-condizioni	<ul style="list-style-type: none">-L'ospitante deve essere registrato nel sistema.-L'ospitato/i deve aver prenotato il soggiorno.-L'ospitato/i deve aver effettuato il check-in.
Post-condizione	<ul style="list-style-type: none">-E' stata memorizzata la data esatta di fine soggiorno.-L'abitazione è stata resa nuovamente prenotabile.

Contratto C01: *controlloCheckIn*

Operazione	controlloCheckIn(dataAttuale)
Riferimenti	Caso d'uso UC9 Check-in/Check-out
Pre-condizioni	<ul style="list-style-type: none">-L'ospitato deve essere registrato nel sistema.-L'ospitato/i deve aver prenotato il soggiorno.-L'ospitato/i non deve aver effettuato il check-in.
Post-condizione	<ul style="list-style-type: none">-Sono state annullate tutte le prenotazioni in cui il campo checkIn è "null" e la data di inizio soggiorno è passata.-L'abitazione è stata resa nuovamente prenotabile.

Durante la fase di progettazione sono stati realizzati, per ogni iterazione, il diagramma delle classi di progetto e i diagrammi di sequenza di sistema. Tali diagrammi sono stati arricchiti con i casi d'uso implementati e con i pattern utilizzati.

pkgModello di progetto

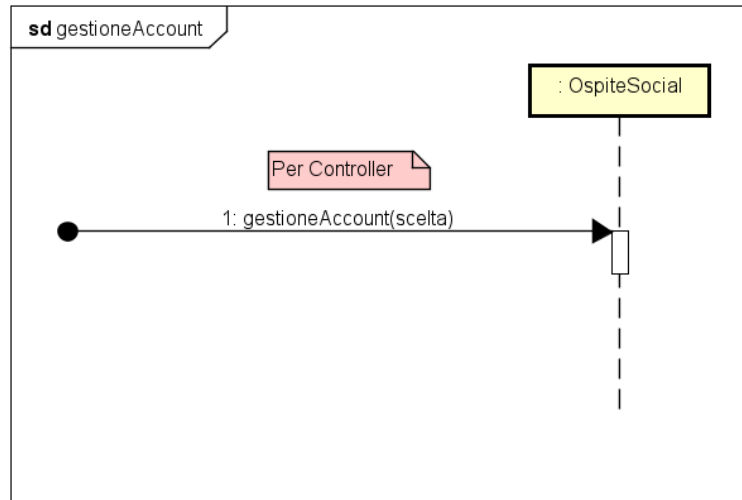
```
classDiagram
    class AbitazioneDAO {
        <<interface>>
        +deleteAbitazione() boolean
        +createAbitazione() boolean
        +updateAbitazione() boolean
        +getAbitazione() Abitazione
        +getAllAbitazioni() List<Abitazione>
    }
    class Abitazione {
        -indirizzo String
        -distanzaCentro float
        -distanzaStazione float
        -citta String
        -dataInizioDisponibilita date
        -dataFineDisponibilita date
        -listaStanze List<Stanza>
        -tariffaGiornaliera int
    }
    class Stanza {
        -tipologiaPostoLetto String
        -tipologiaStanza String
        -numeroPostiLetto int
        -codiceStanza int
    }
    class UtenteDAO {
        <<interface>>
        +deleteUtente() boolean
        +createUtente() boolean
        +updateUtente() boolean
        +getUtente() Utente
        +addCrediti() boolean
        +removeCrediti() boolean
        +setStatoPrenotabile() boolean
    }
    class Utente {
        -cf String
        -nome String
        -cognome String
        -eta int
        - Sesso char
        -cittaResidenza String
        -cellulare String
        -credito int
        -statoPrenotabile boolean
    }
    class OspiteSocial {
        +inserisciUtente() Utente
        +eliminaUtente() boolean
        +modificaDatiUtente() boolean
        +visualizzaInformazioniUtente() Utente
        +inserimentoAlloggio() boolean
        +modificaDatiAlloggio() boolean
        +visualizzaAlloggio() Abitazione
        +eliminaAlloggio() boolean
        +ricercaAlloggio() List<Abitazione>
        +inserimentoDatiStanza() boolean
        +accettaRichiesta() boolean
        +richiestaPrenotazione() boolean
        +modificaDatiStanza() boolean
        +visualizzaAlloggiUtente() List<Abitazione>
        +visualizzaRichieste() List<Prenotazione>
        +rifiutaRichiesta() boolean
        +visualizzaPrenotazioni() List<Prenotazione>
        +aggiungiFeedbackOspitato() boolean
        +aggiungiFeedbackOspitante() boolean
        +checkin() boolean
        +checkout() boolean
        +visualizzaFeedbackAbitazione() float
        +visualizzaFeedbackOspitato() float
        +modificaPrenotazione() boolean
        +annullaPrenotazione() boolean
    }
    class Prenotazione {
        -dataInizio date
        -dataFine date
        -ospiti List<String>
        -ospitante String
        -codicePrenotazione int
        -codiceAbitazione int
        -stato String
        -codiciStanze List<int>
        -checkin date
        -checkout date
        -motivazioni String
    }
    class FeedbackDAO {
        <<interface>>
        +createFeedbackOspitato() boolean
        +createFeedbackOspitante() boolean
        +getMediaFeedbackAbitazione() float
        +getMediaFeedbackOspitato() float
    }
    class Feedback {
        -valutazione int
        -codiceFeedback int
    }
    class FeedbackOspitato {
    }
    class FeedbackOspitante {
    }
    class StanzaDAO {
        <<interface>>
        +createStanza() boolean
        +updateStanza() boolean
        +getAllStanze() List<Stanza>
    }
    class PrenotazioneDAO {
        <<interface>>
        +setStato() boolean
        +makePrenotazione() boolean
        +searchAlloggio() List<Abitazione>
        +showRichieste() List<Prenotazione>
        +setMotivazione() boolean
        +showPrenotazioni() List<Prenotazione>
        +setCheckin() boolean
        +setCheckout() boolean
        +makeCrediti() int
        +getOspitante() String
        +getOspiti() List<String>
        +setPeriodoPrenotazione() boolean
    }

    AbitazioneDAO <|-- Abitazione
    UtenteDAO <|-- Utente
    FeedbackDAO <|-- Feedback
    StanzaDAO <|-- Stanza
    PrenotazioneDAO <|-- Prenotazione

    Abitazione "1" -- "0..*" Stanza : - codiceAbitazione
    Abitazione "1" -- "0..*" OspiteSocial : - proprietario
    OspiteSocial "1" -- "0..*" Utente
    OspiteSocial "1" -- "0..*" Stanza : - codiciStanze
    OspiteSocial "1" -- "0..*" Prenotazione
    Utente "1" -- "0..*" Feedback : - autore
    Utente "1" -- "0..*" Prenotazione
    Feedback "0..*" -- "1..*" FeedbackOspitato : - codicePrenotazione
    Feedback "0..*" -- "1..*" FeedbackOspitante : - ospite
    FeedbackOspitato <|-- FeedbackOspitante
    FeedbackOspitato <|-- FeedbackOspitante
```

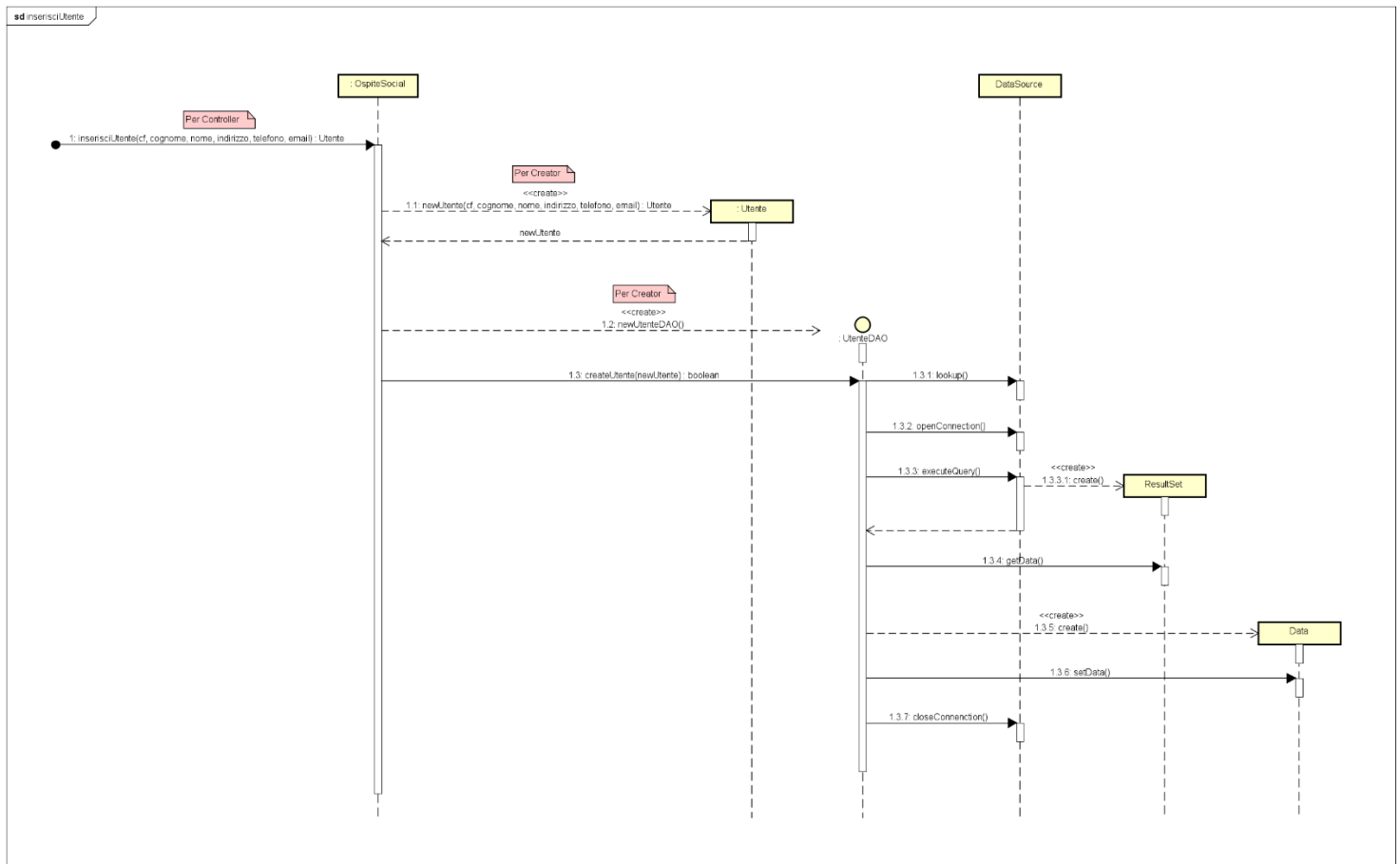
3.2. Diagramma di sequenza (SD)

UC1 – gestioneAccount



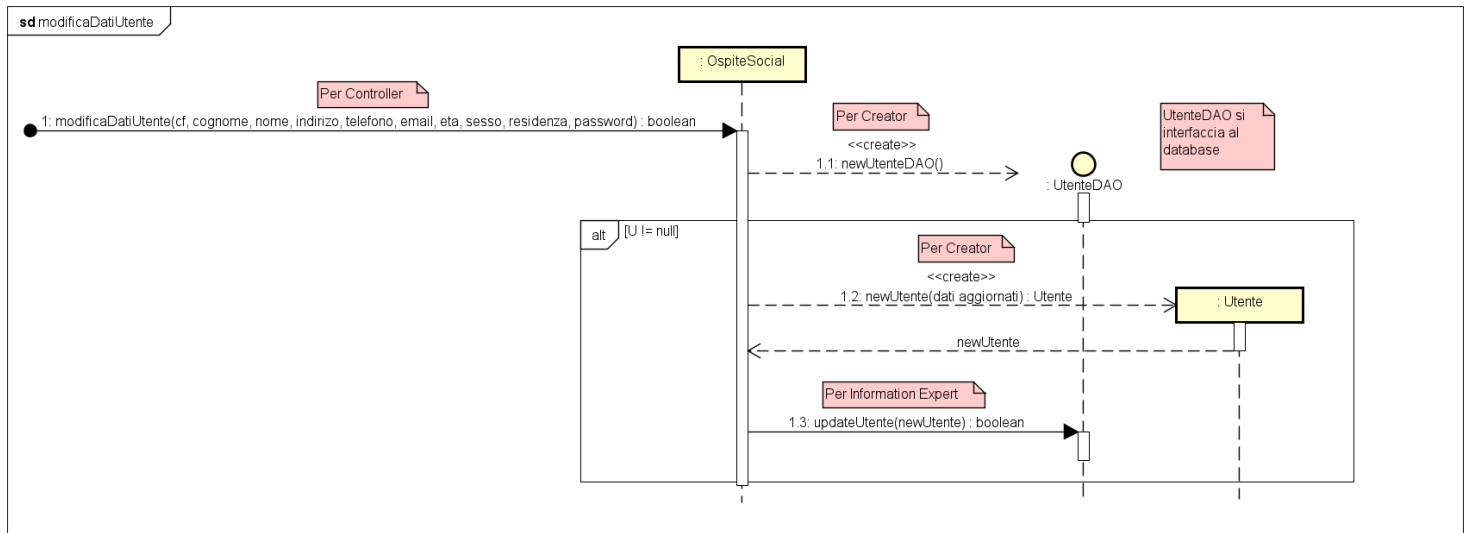
Per il Pattern **Controller**, OspiteSocial è il primo “oggetto” a ricevere e coordinare le operazioni. Tale Pattern si applicherà in tutti i successivi diagrammi di sequenza, sarà sempre il sistema a catturare i messaggi che provengono dalle varie UI degli attori.

UC1 – inserisciUtente

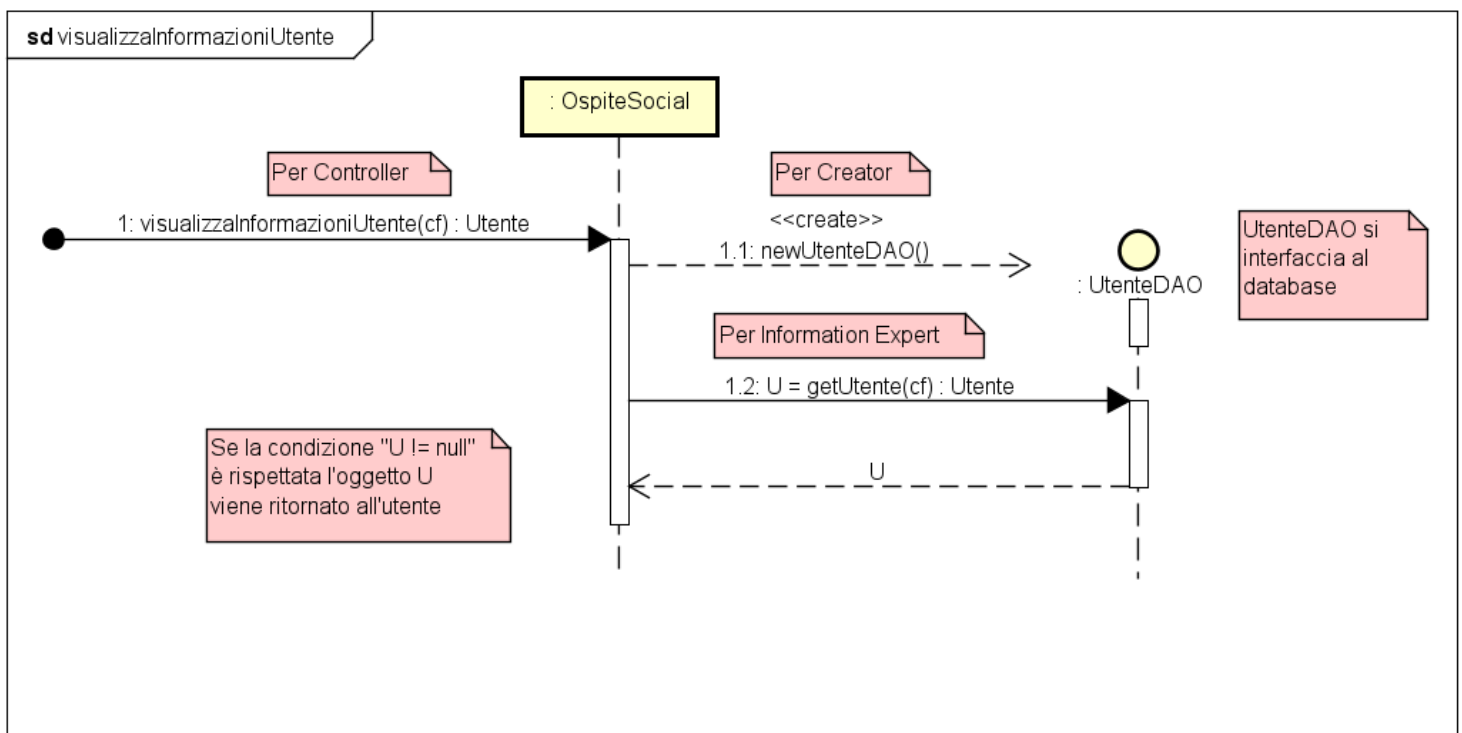


Poiché `OspiteSocial` possiede i dati per inizializzare un oggetto di tipo `Utente` allora, per il Pattern **Creator** sarà `OspiteSocial` a creare la nuova istanza. Inoltre, si è scelto d'utilizzare un Database per fornire Persistenza dei dati. A tal proposito, abbiamo applicato il pattern DAO ed in questo diagramma viene mostrata l'interazione tra l'interfaccia DAO ed il Database. Nei successivi diagrammi tale interazione sarà sottintesa.

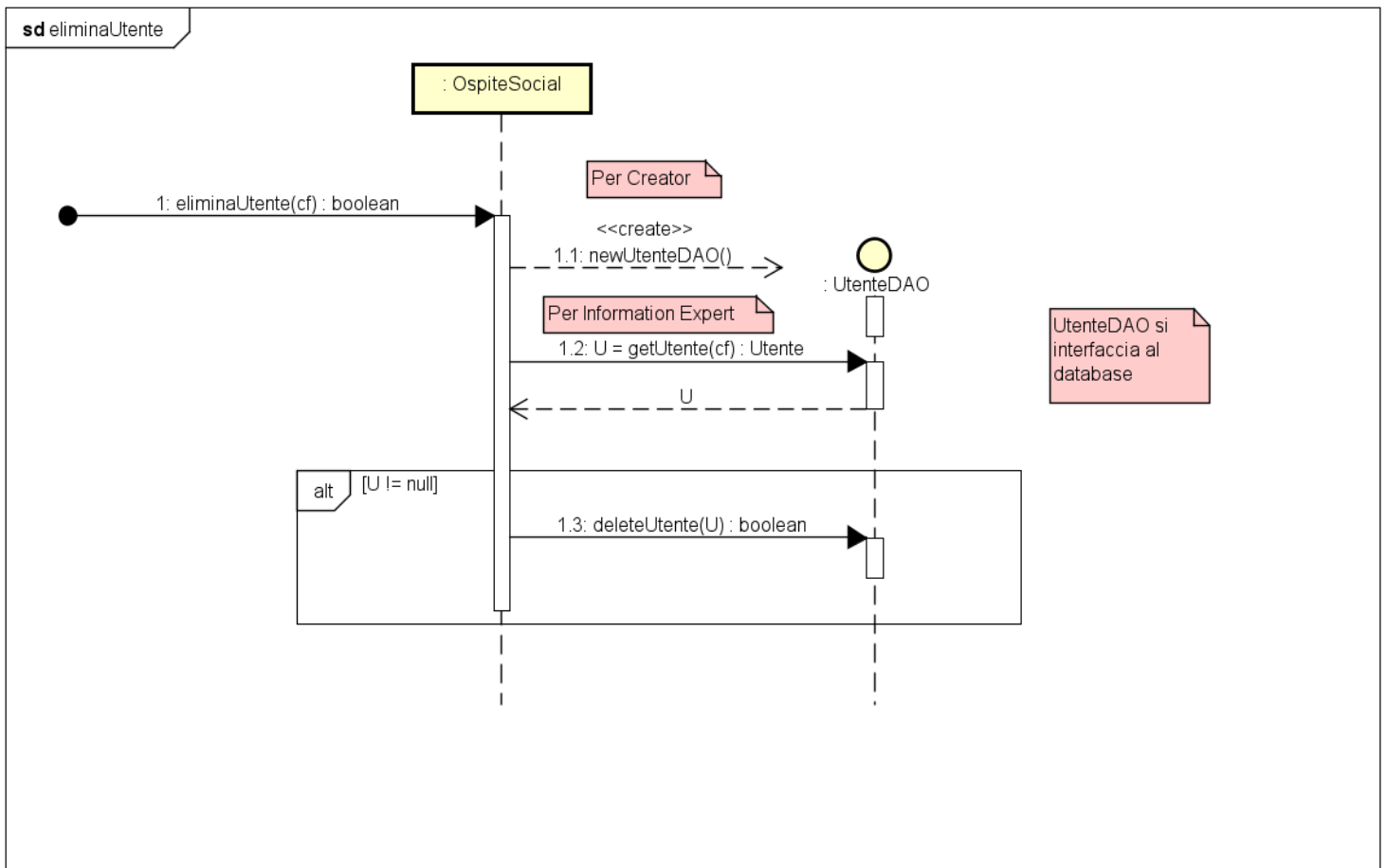
UC1 – modificaDatiUtente



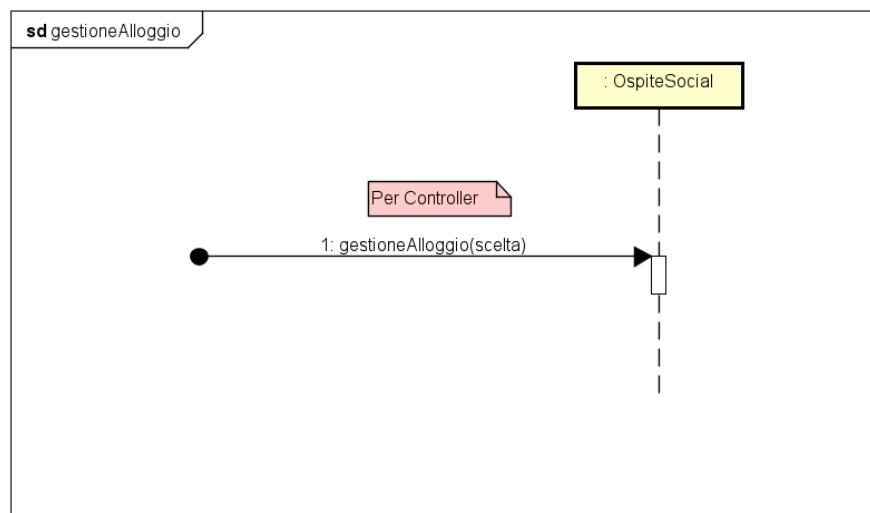
UC1 – visualizzaInformazioniUtente



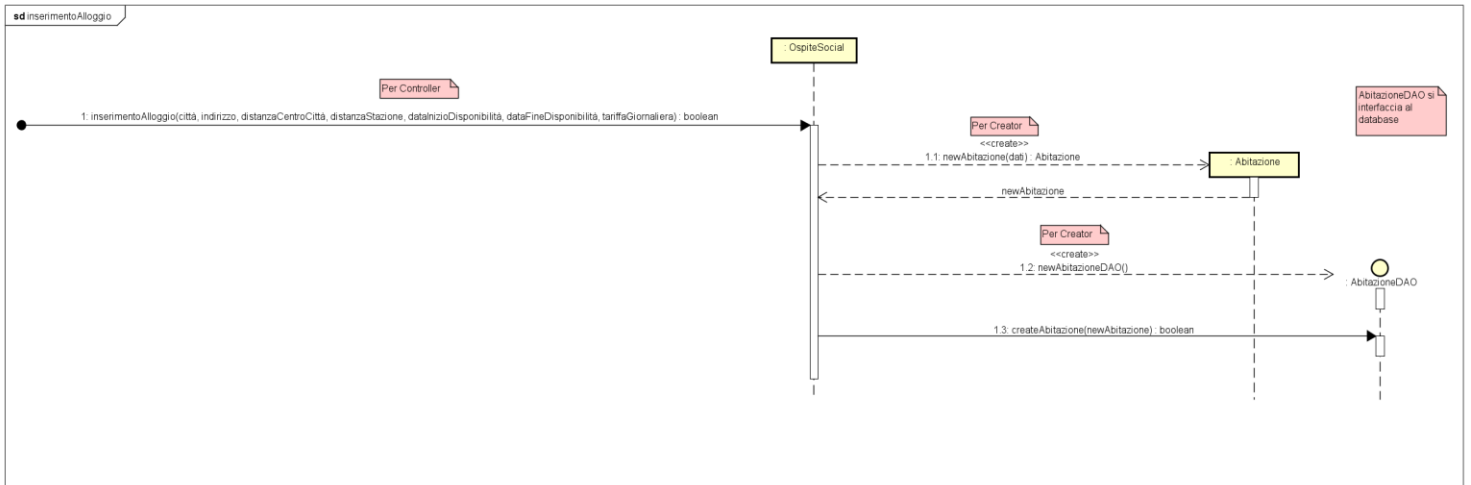
UC1 – eliminaUtente



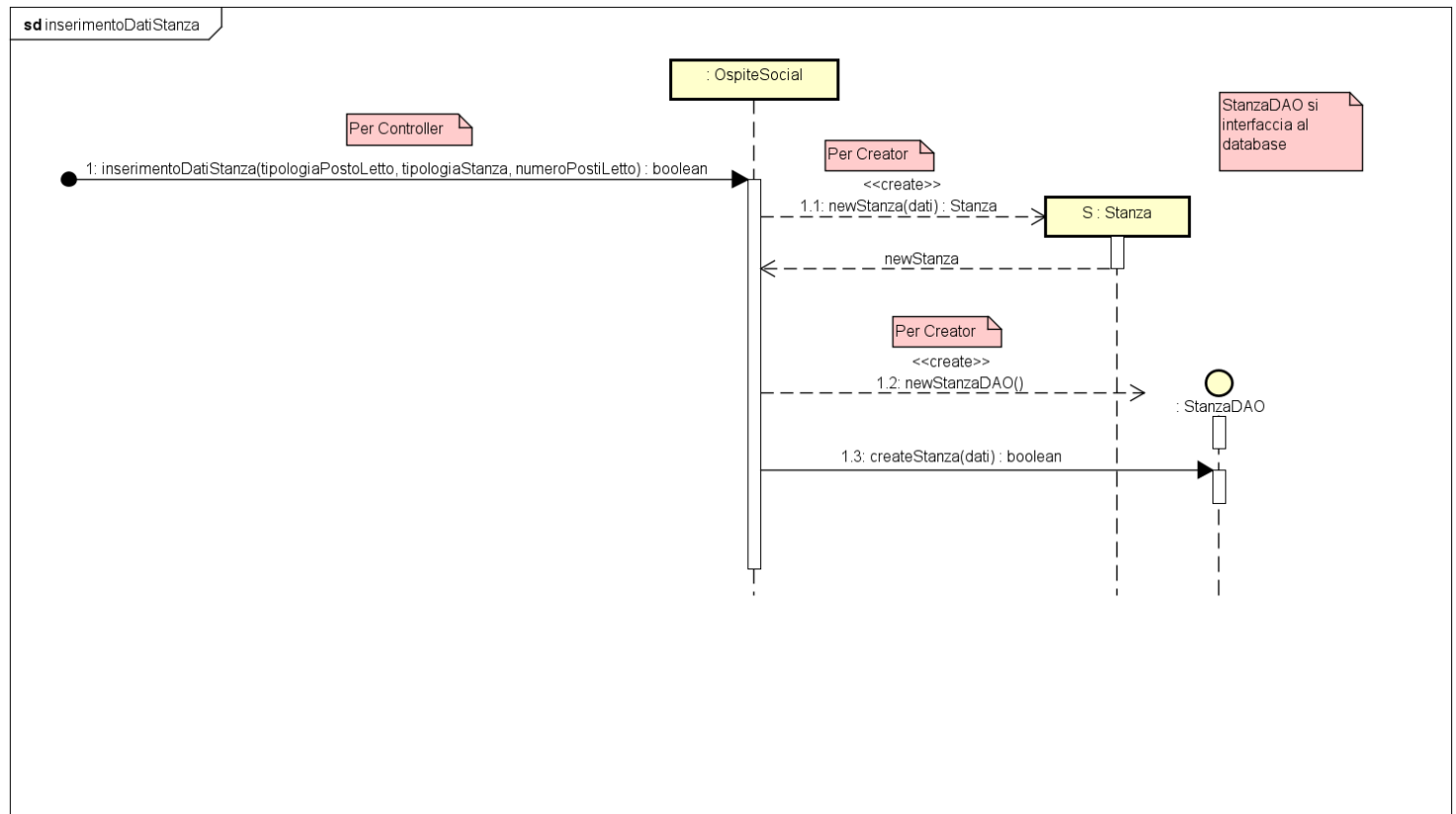
UC2 – gestioneAlloggio



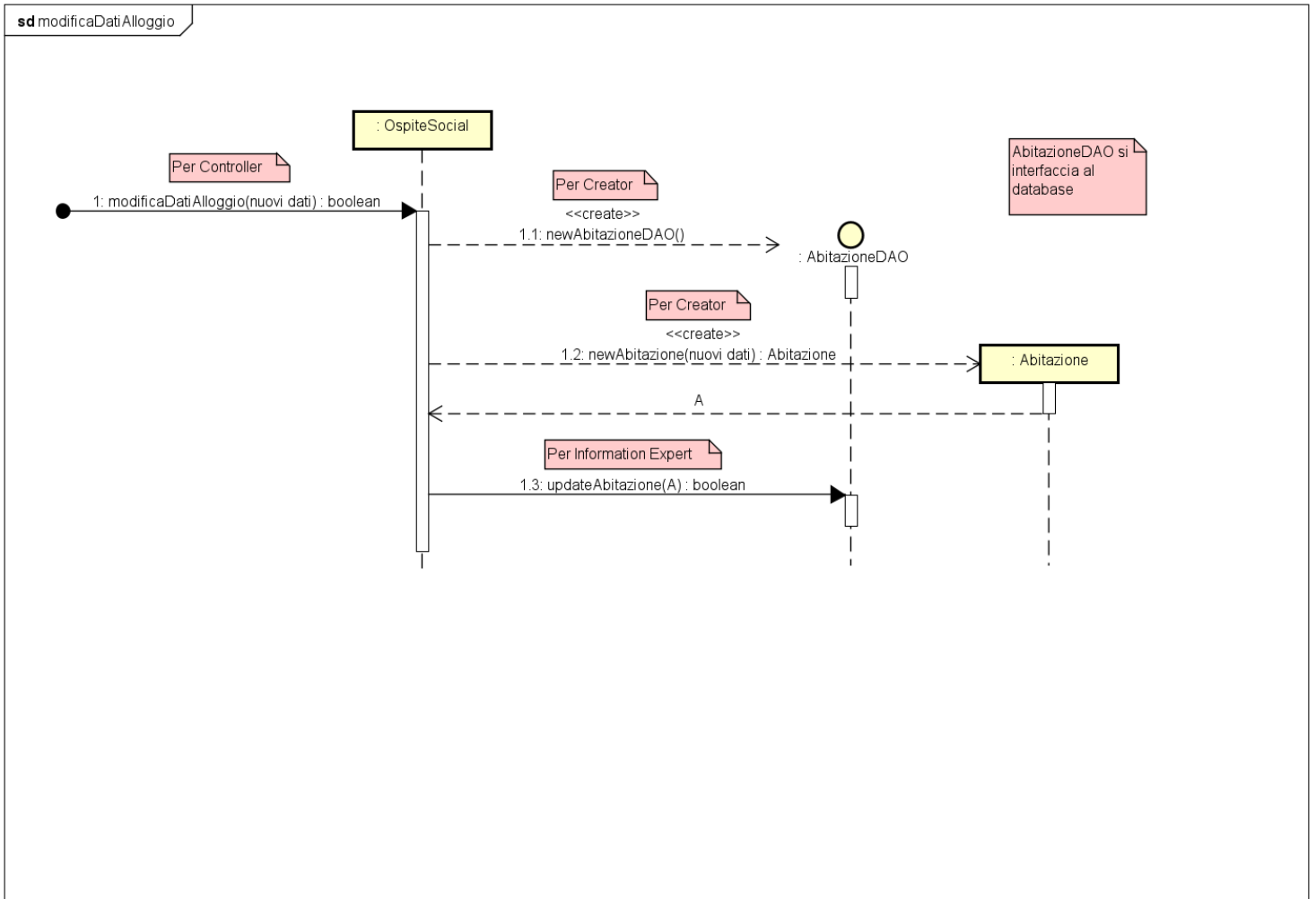
UC2 – inserimentoAlloggio



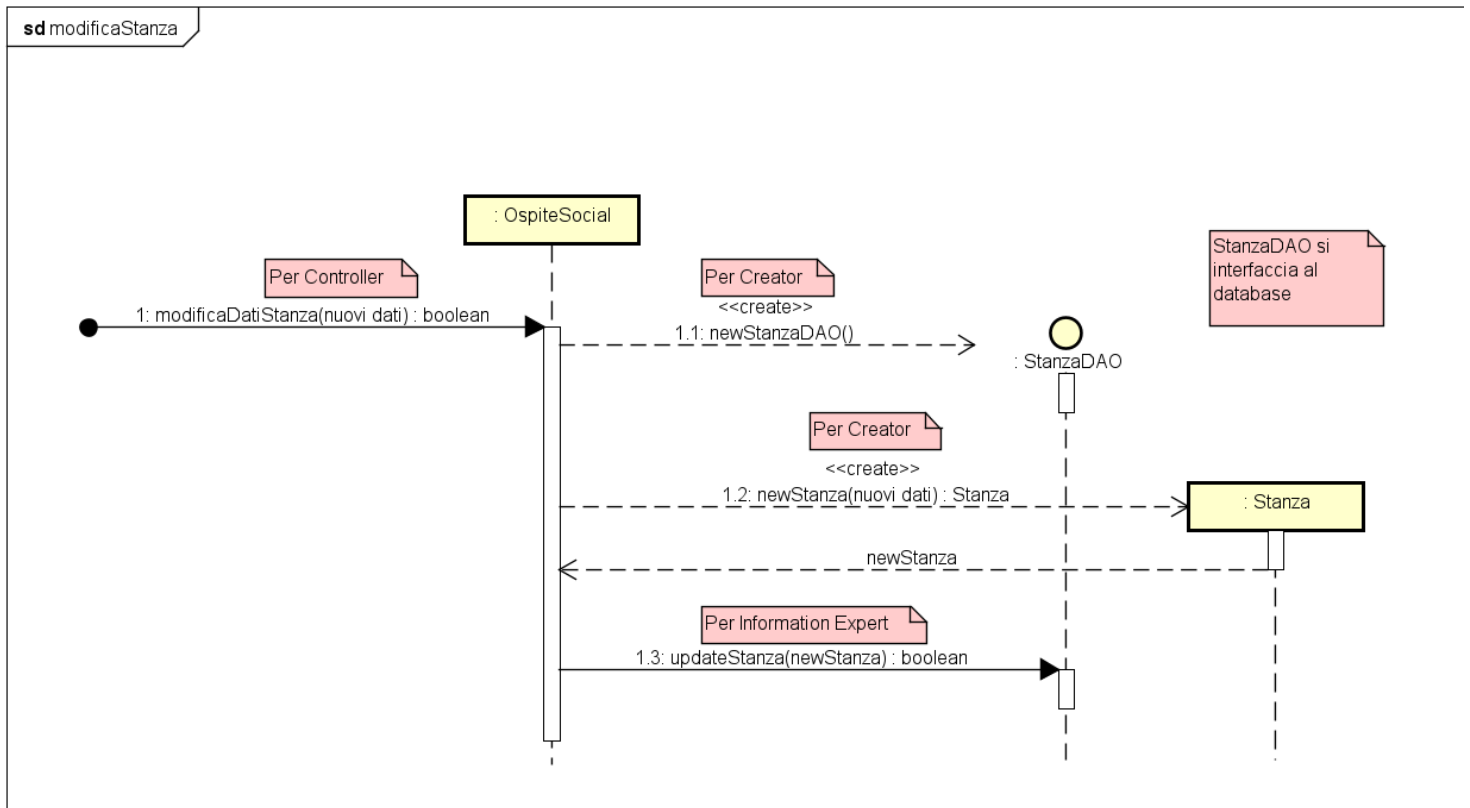
UC2 – inserimentoDatiStanza



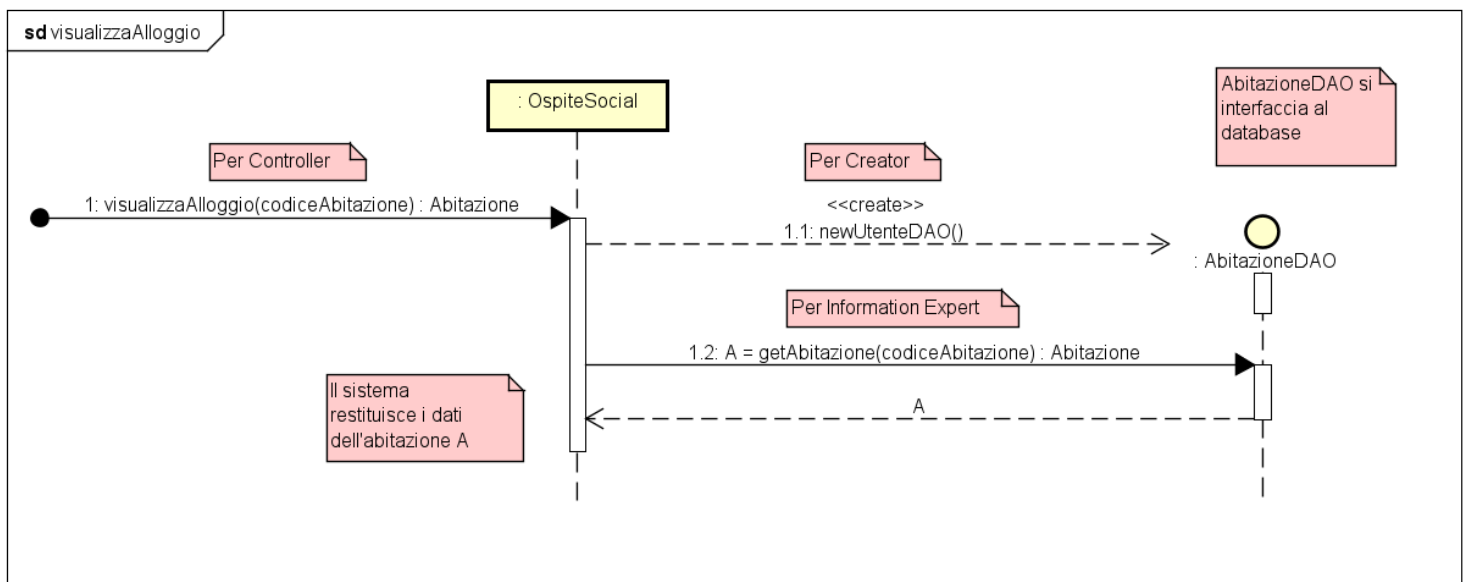
UC2 – modificaDatiAlloggio



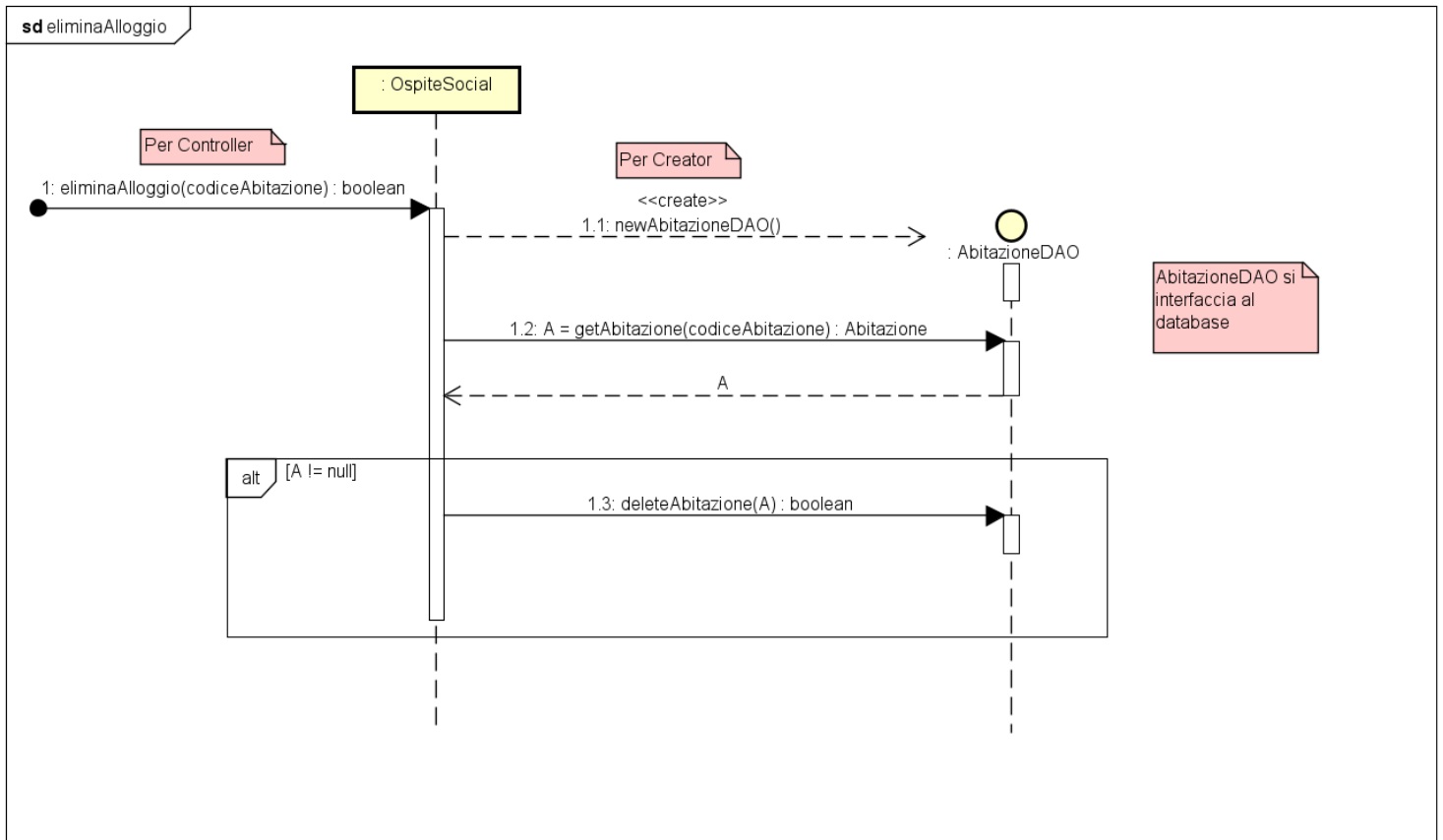
UC2 – modificaStanza



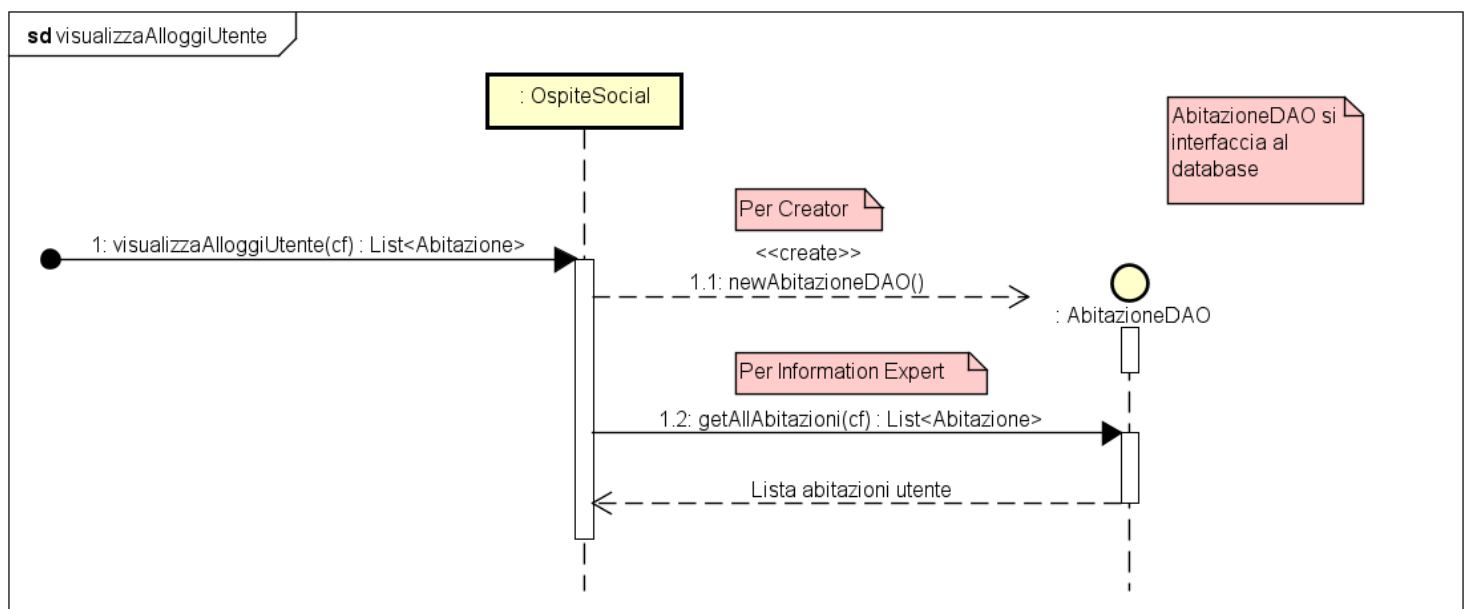
UC2 – visualizzaAlloggio



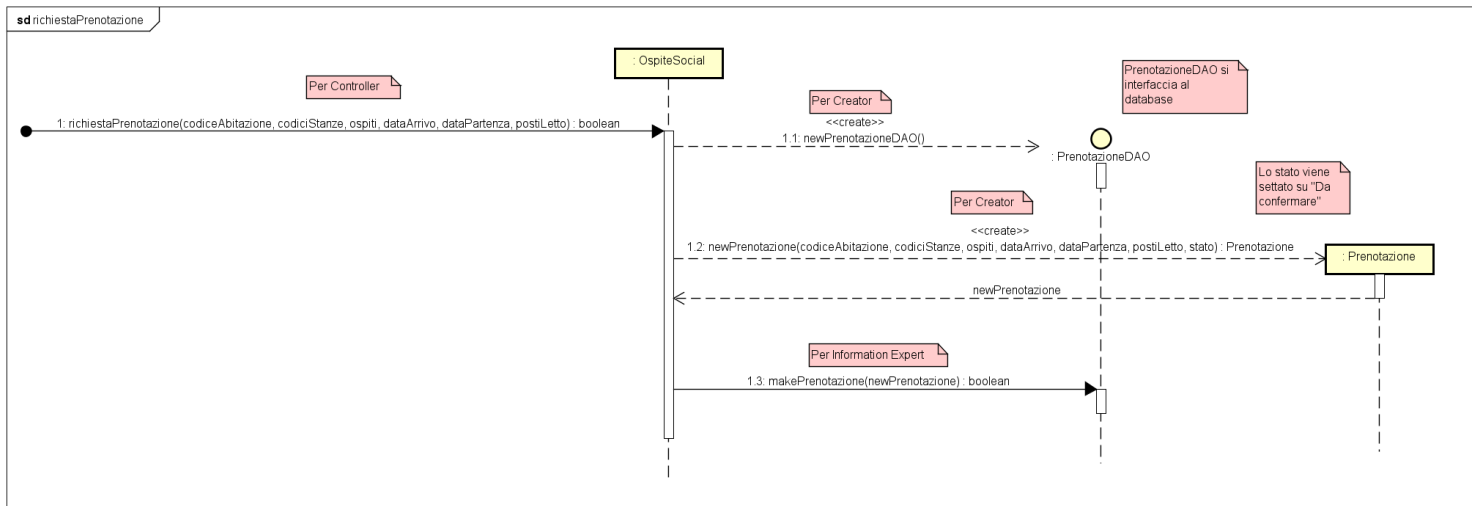
UC2 – visualizzaAlloggiUtente



UC2 – eliminaAlloggio

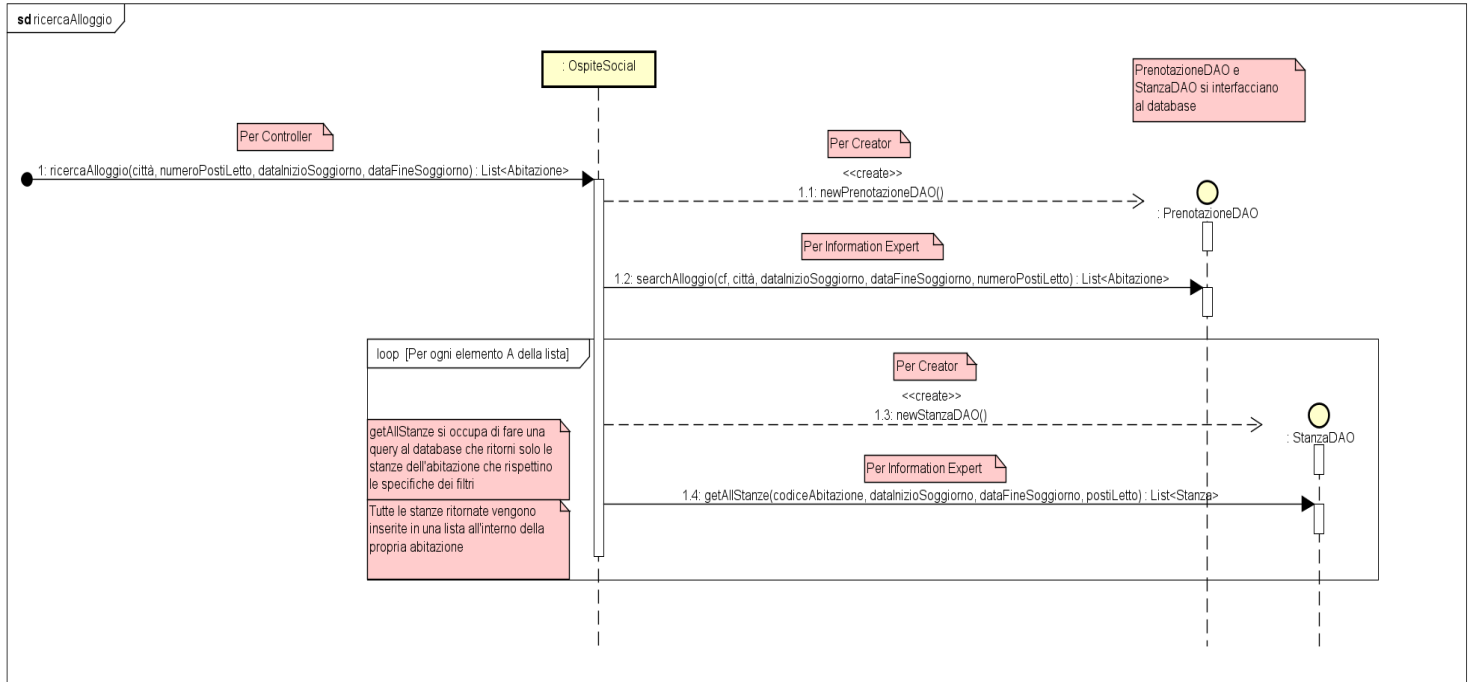


UC3 – richiestaPrenotazione



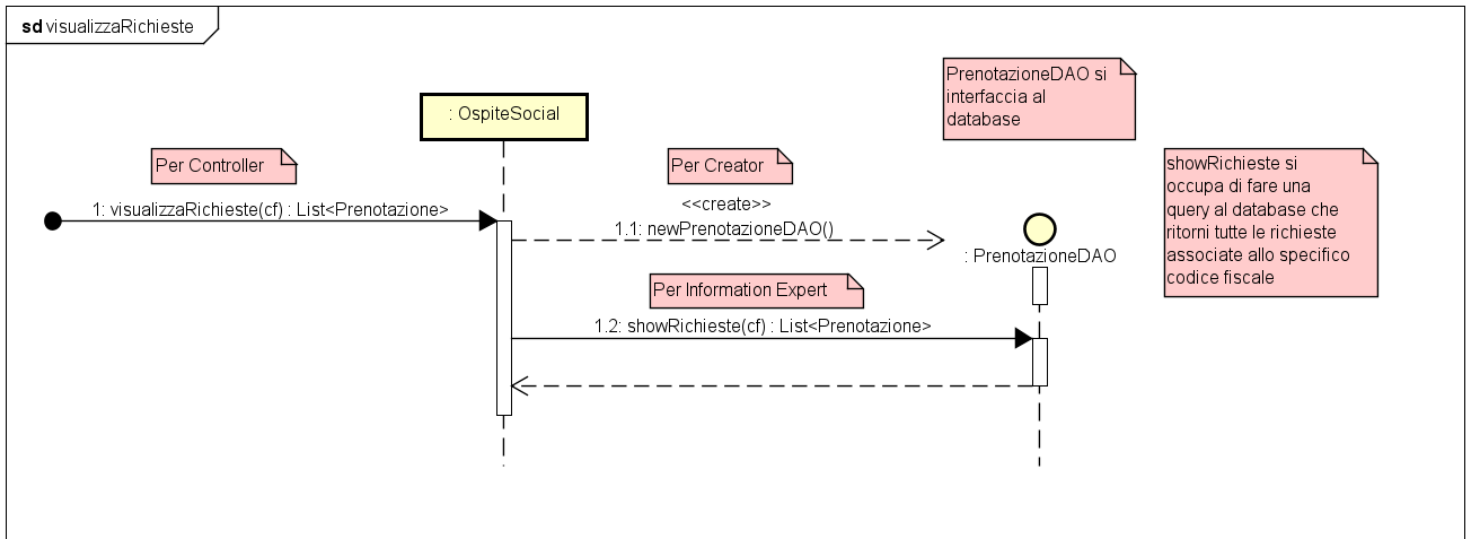
L'attributo "stato" della classe *Prenotazione*, monitora lo stato della prenotazione, acquisendo come possibili valori: "Da confermare", "Confermata", "In corso", "Annullata", "Terminata".

UC4 – ricercaAlloggio

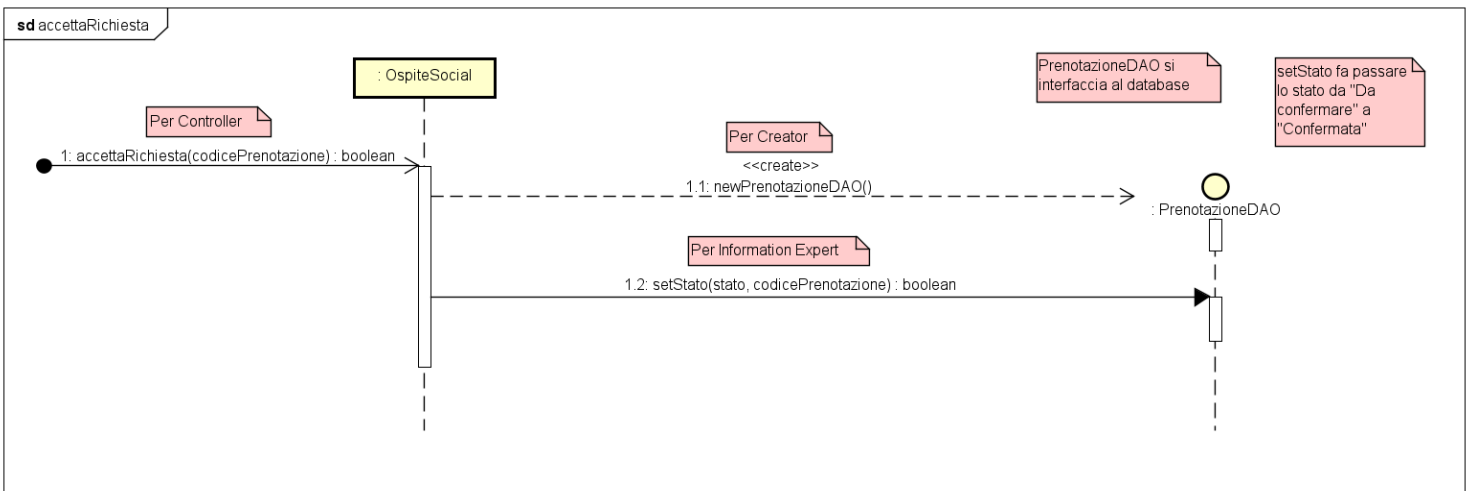


In generale le classi *DAO*, per il Pattern Information Expert, possono recuperare e gestire le relative informazioni.

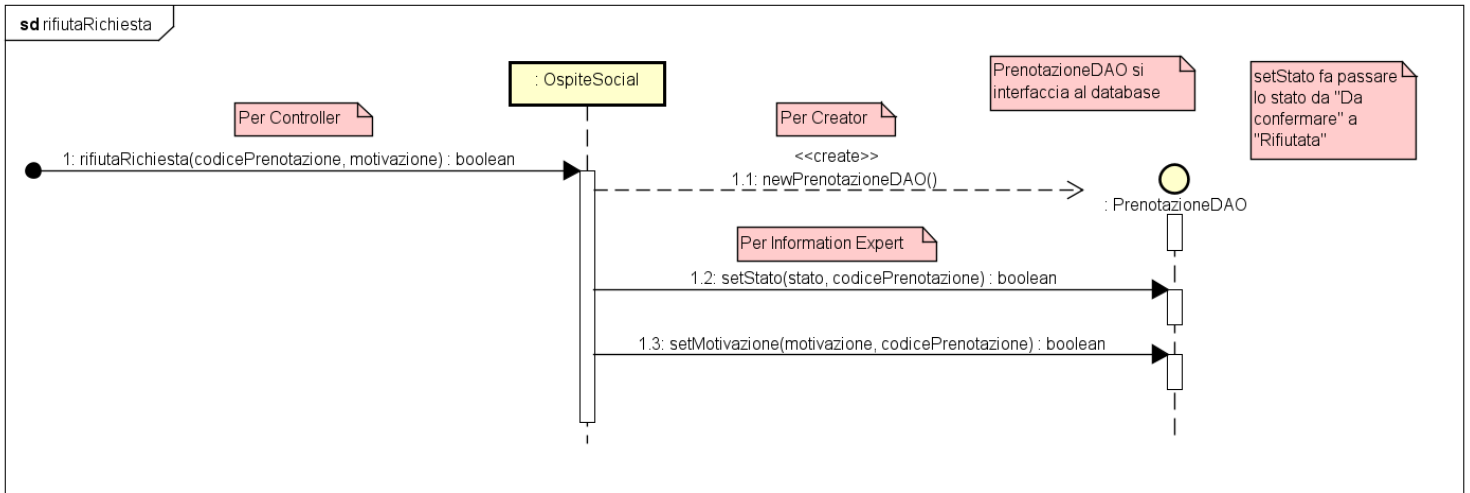
UC5 – visualizzaRichieste



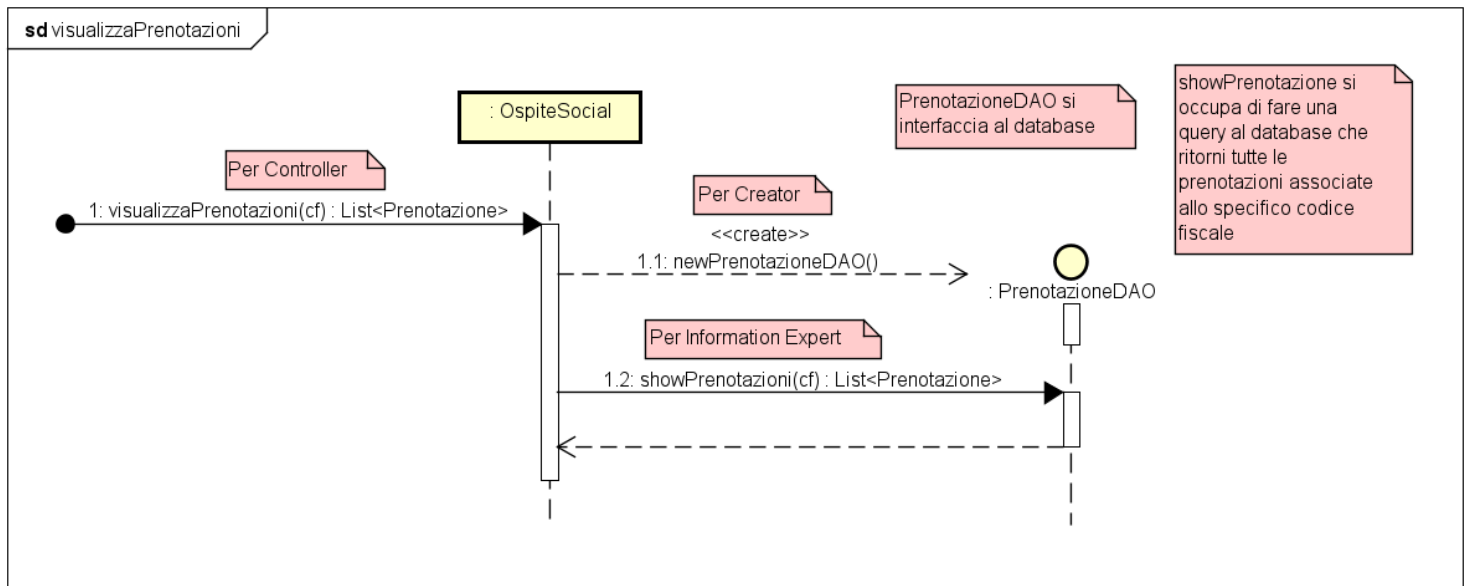
UC5 – accettaRichiesta



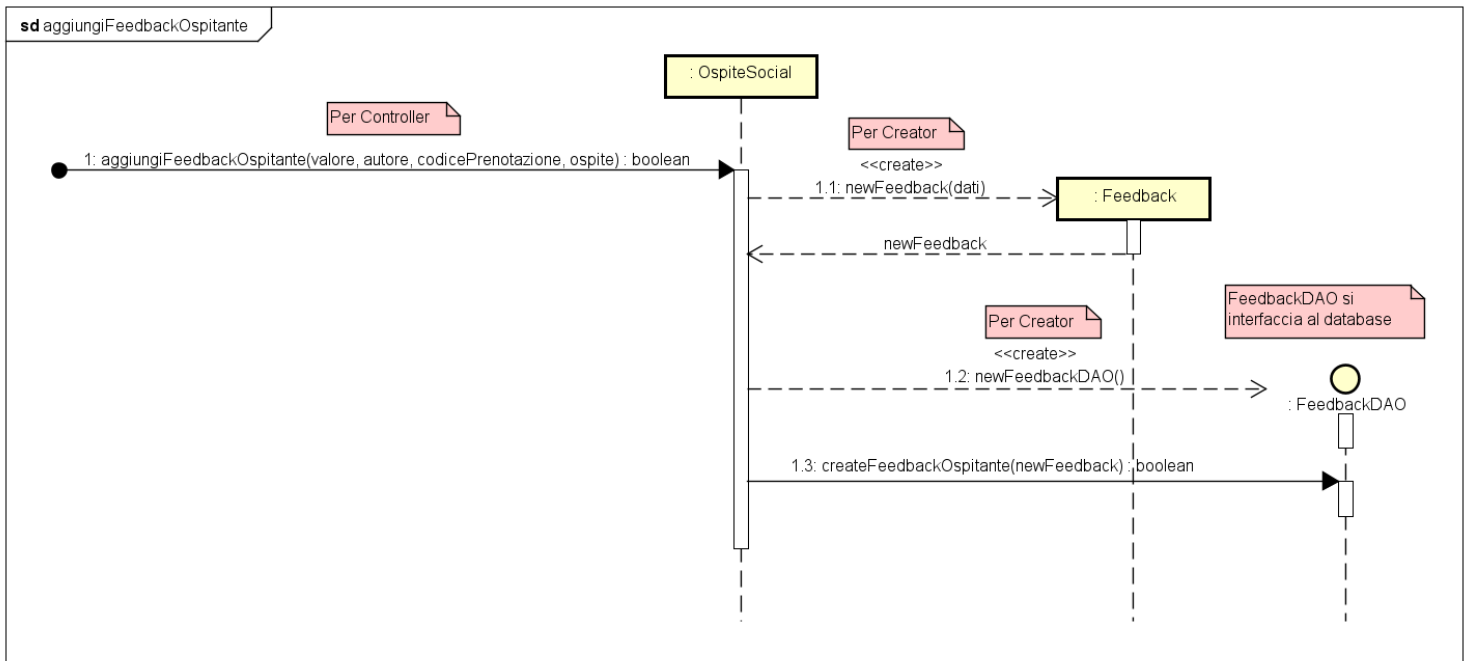
UC5 – rifiutaRichiesta



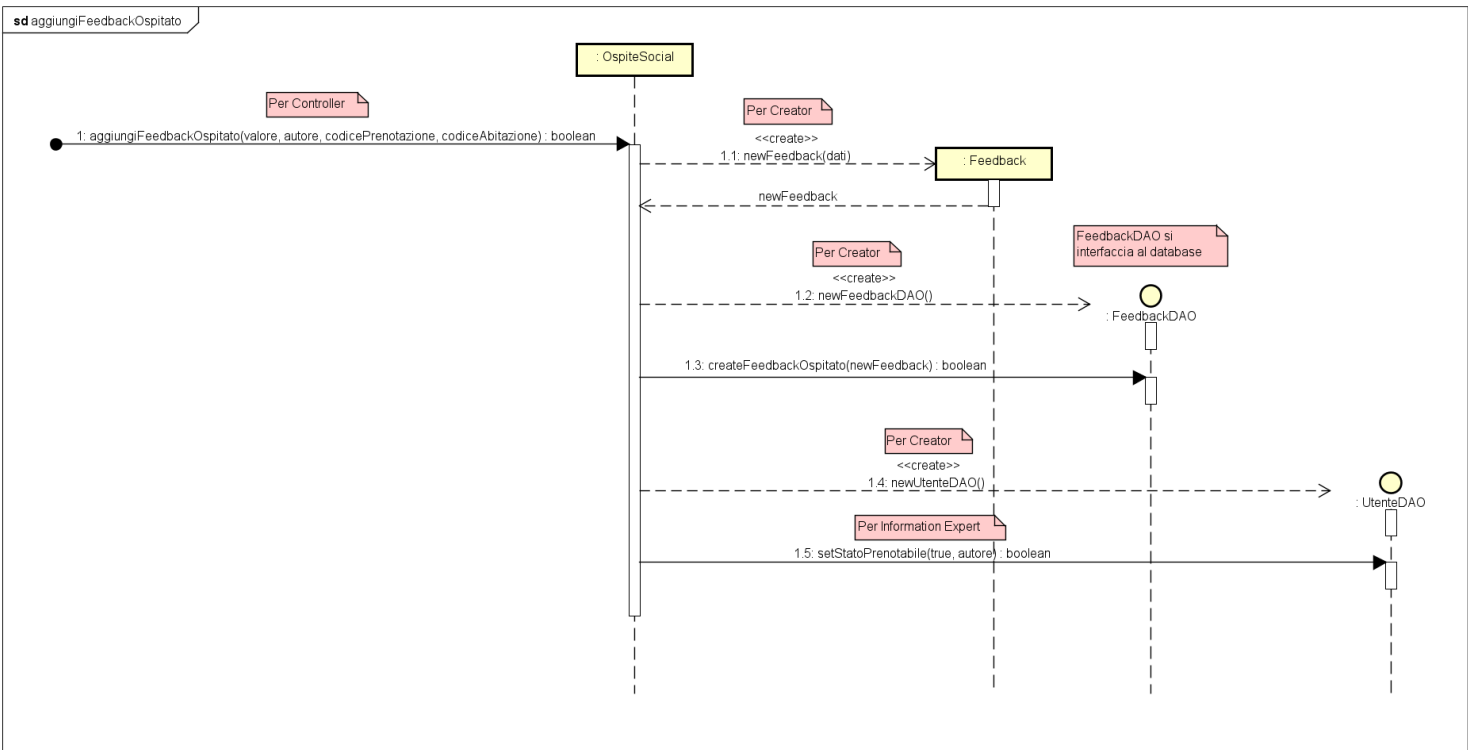
UC6 – visualizzaPrenotazioni



UC7 – aggiungiFeedbackOspitante

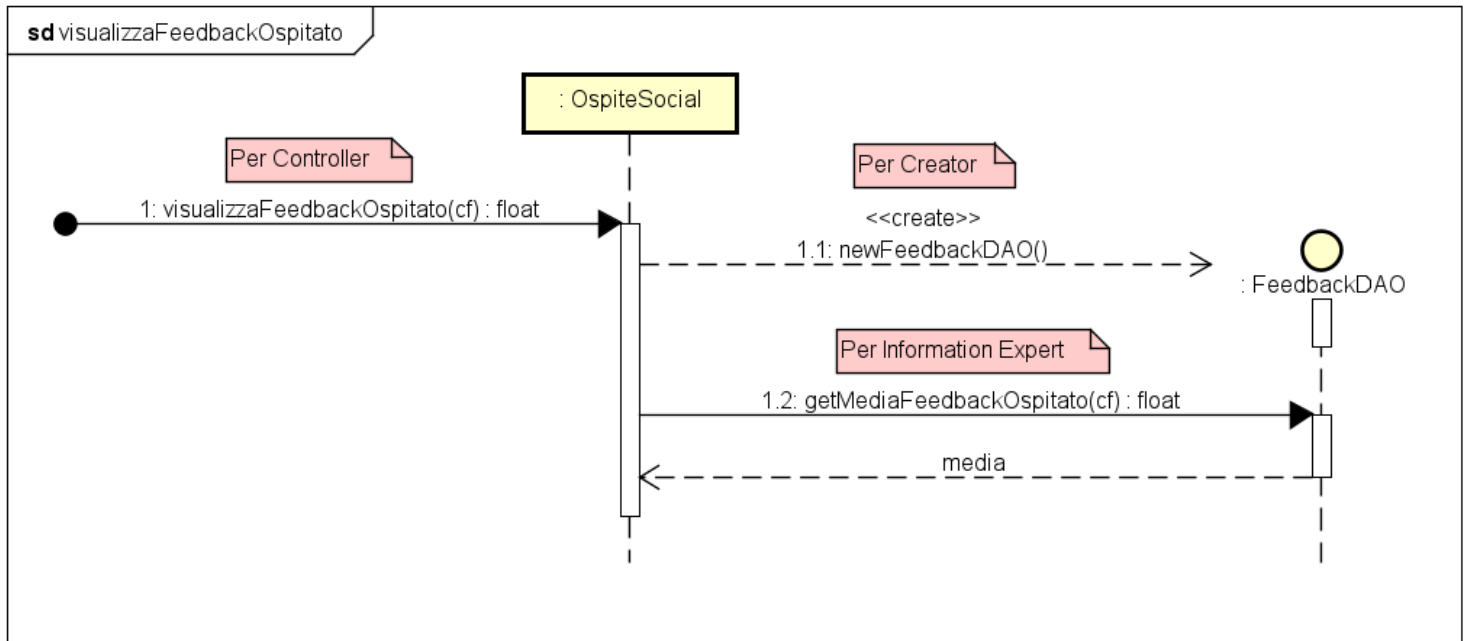


UC7 – aggiungiFeedbackOspitato

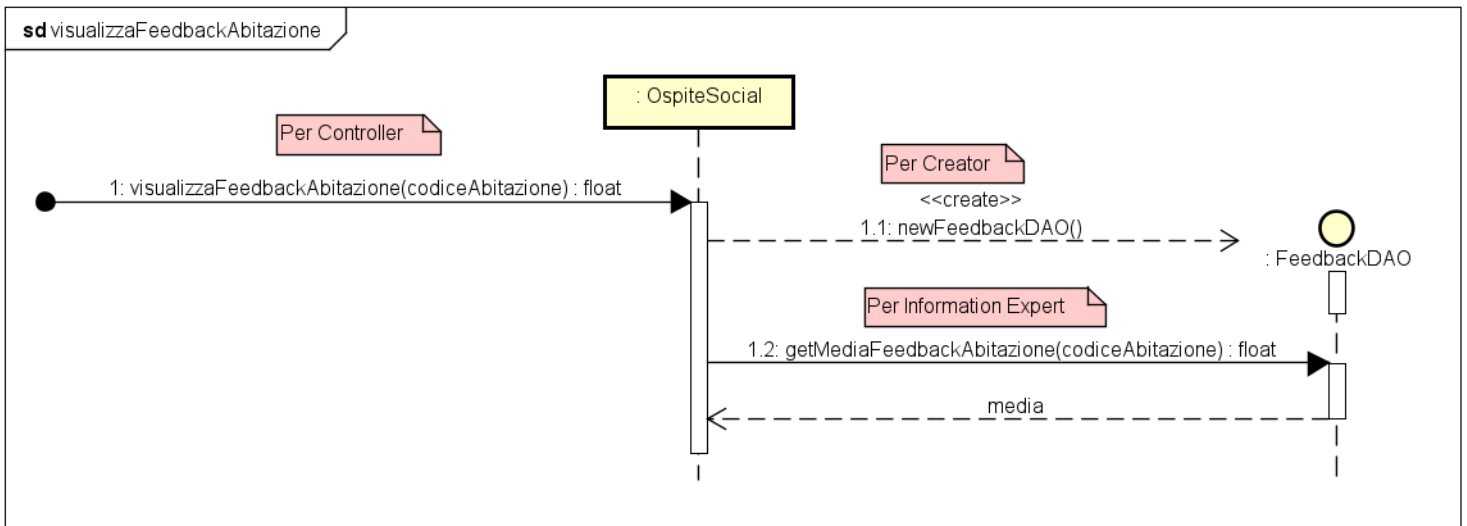


Nel momento in cui l'ospitato aggiunge il feedback il sistema, dopo l'inserimento, imposta l'attributo "statoPrenotabile" dello stesso a "true" dandogli così la possibilità di effettuare nuove prenotazioni.

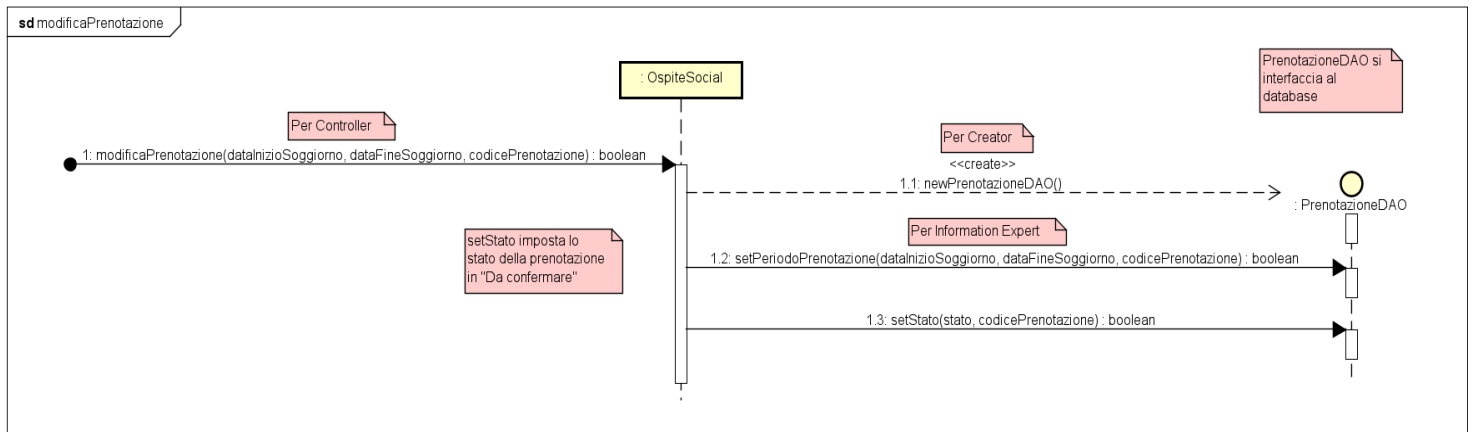
UC7 – visualizzaFeedbackOspitato



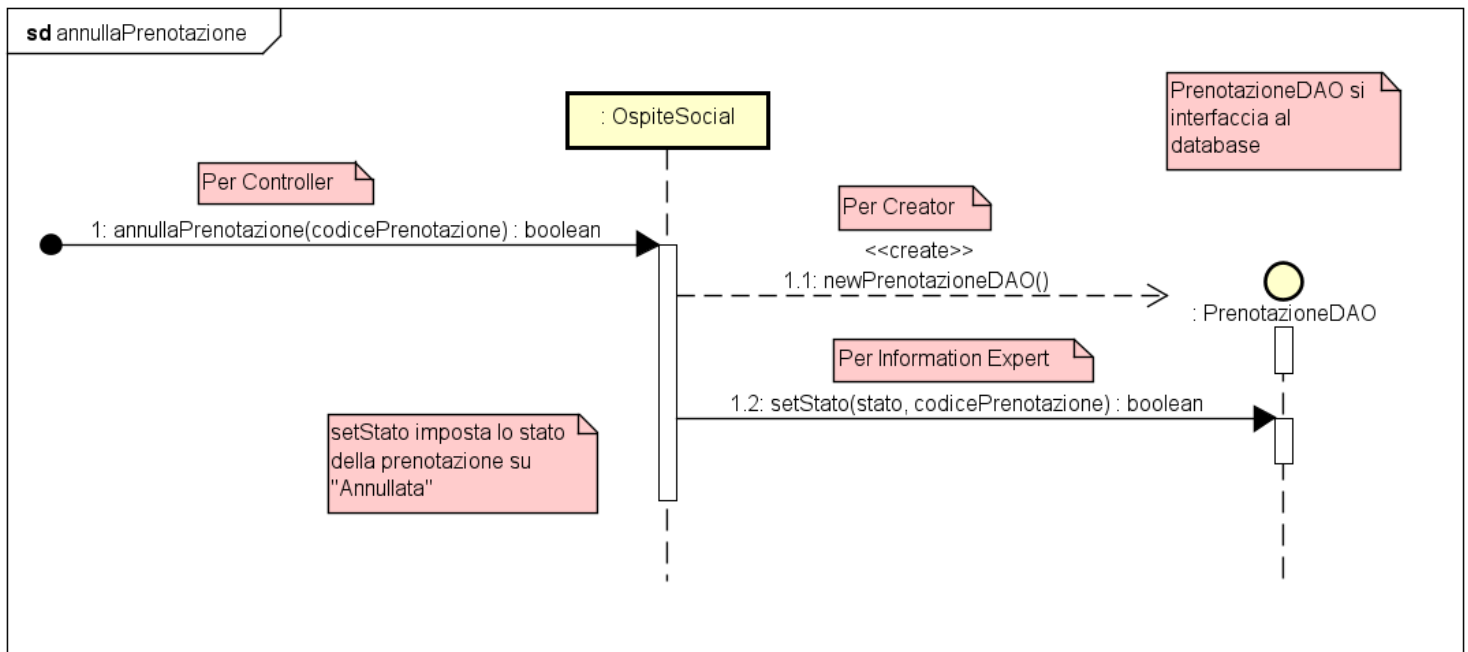
UC7 – visualizzaFeedbackAbitazione



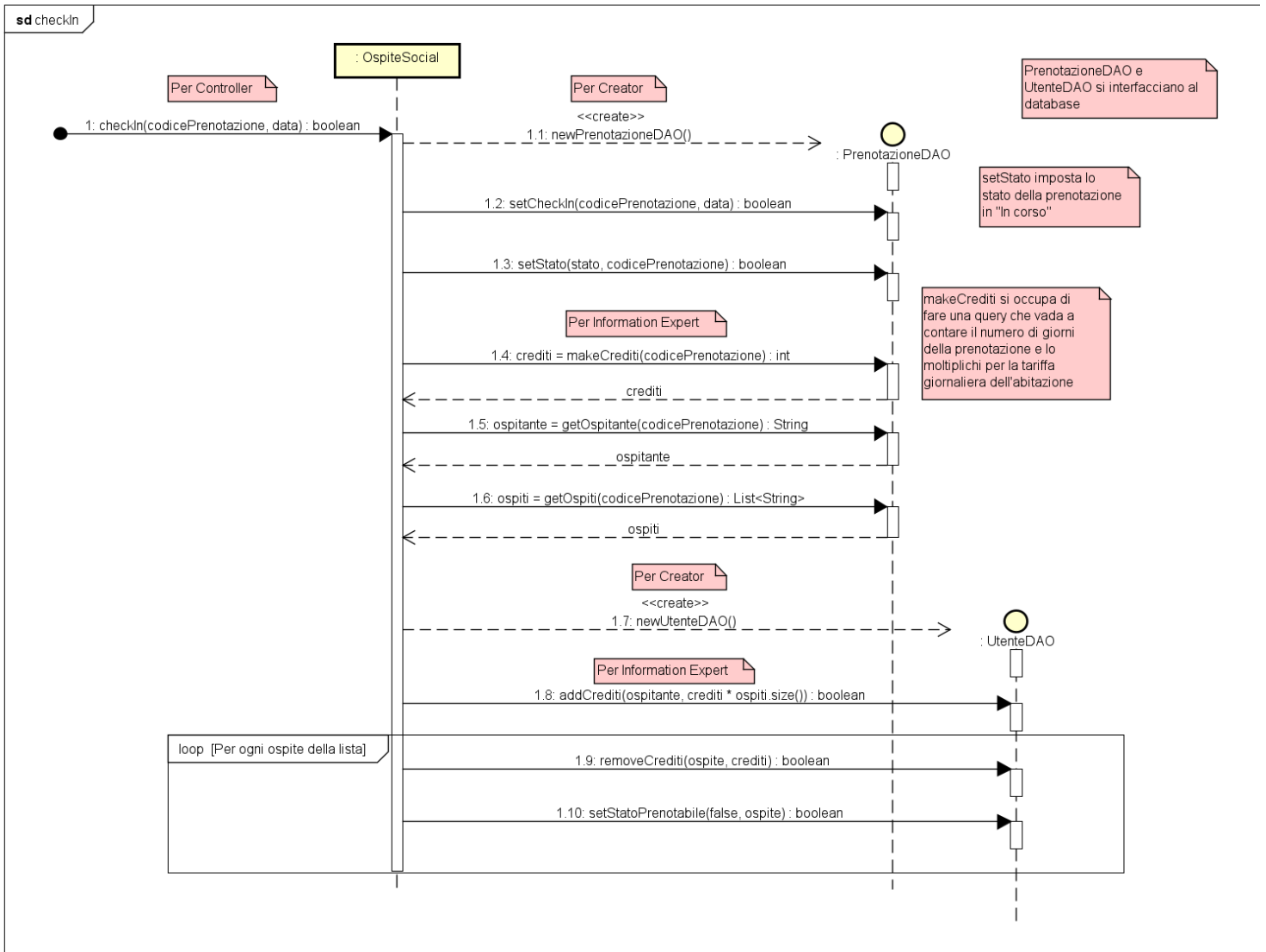
UC8 – modificaPrenotazione



UC8 – annullaPrenotazione

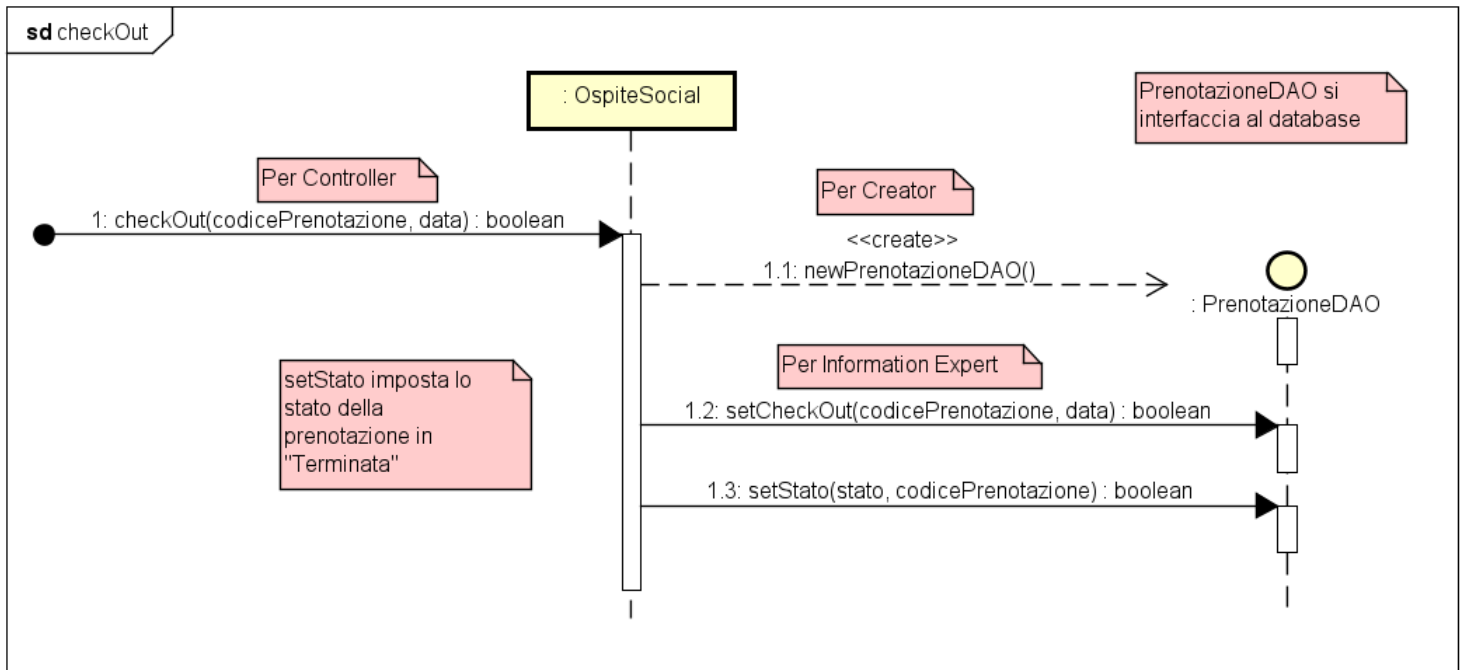


UC9 – CheckIn

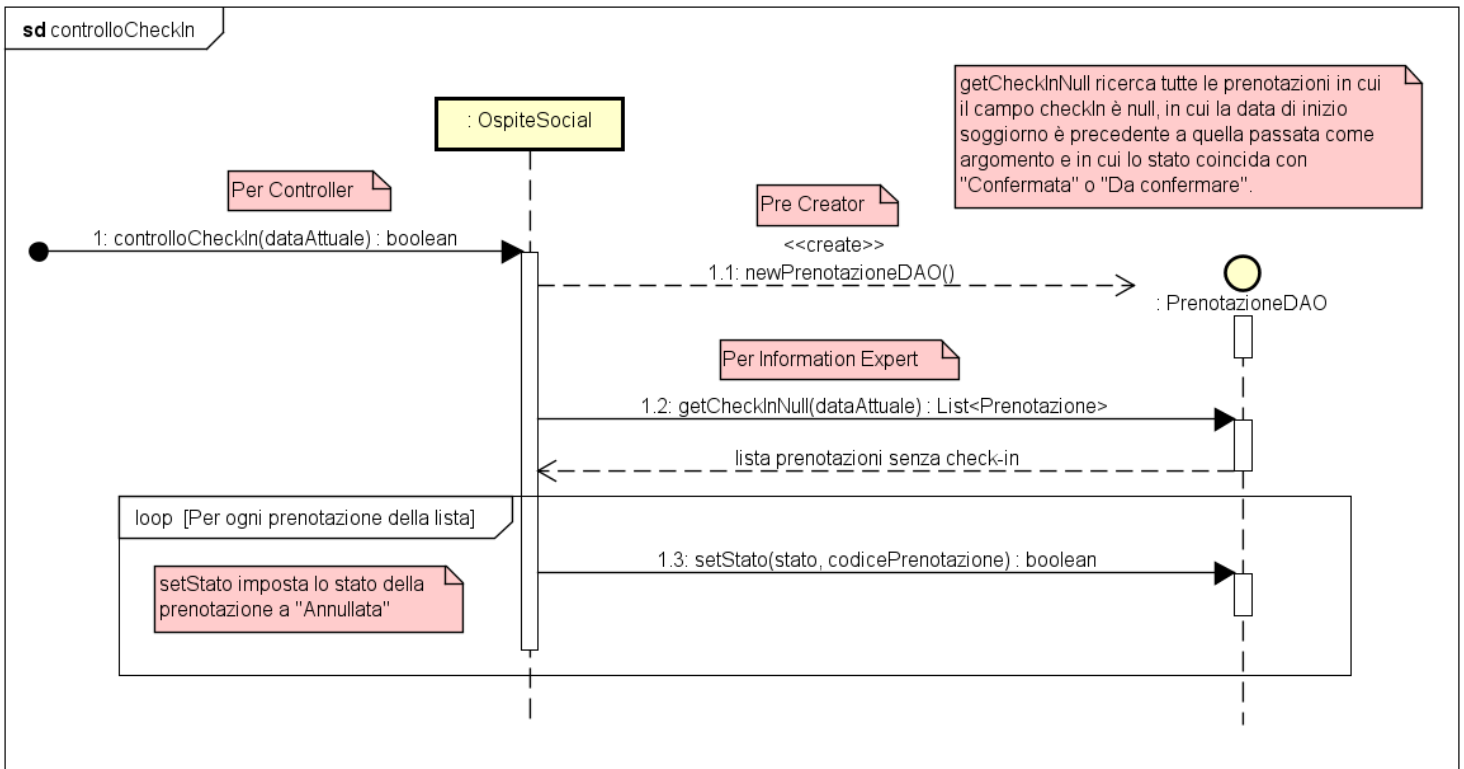


Durante il check-in l'attributo "statoPrenotabile" di tutti gli utenti che figurano come ospiti viene settato a "false" in modo da rendere necessarie le procedure di check-out e rilascio feedback ai fini di effettuare una nuova prenotazione. Per semplicità abbiamo considerato che tutti gli ospiti di una prenotazione arrivano e vanno via insieme, di conseguenza l'operazione di check-in sarà effettuata solo da un ospite e sarà valida per tutti gli ospiti della prenotazione. La stessa considerazione va fatta per l'operazione di check-out.

UC9 – CheckOut



UC9 – ControlloCheckIn



3.2. Pattern applicati

- Creator
- Information Expert
- Singleton (per la Classe OspiteSocial)
- DAO

4. Codice prodotto e test

L'implementazione del software è stata effettuata tramite la programmazione orientata agli oggetti in JAVA, in ambiente IDE Netbeans.

La strategia di sviluppo, atta alla suddivisione del sistema da realizzare in gerarchia di moduli e componenti, è stata una strategia intermedia tra top-down e bottom-up.

Inizialmente è stata seguita una strategia prevalentemente top-down (al fine di trovare una soluzione del problema). Nel mentre si è tenuto in considerazione sempre l'eventuale riutilizzo del software già

implementato o moduli per i quali si premeditava un possibile riutilizzo. Di conseguenza sono stati confrontati i moduli ottenuti con quelli esistenti, e nel caso in cui si trovavano moduli

simili o già realizzati, è stato modificato il progetto per riutilizzarli. Infine il progetto è stato visto più volte in fase di progettazione.

Il testing di tale codice è stato condotto tramite l'utilizzo del framework open source JUNIT.

Nelle varie iterazioni sono stati eseguiti dei test: l'approccio utilizzato è stato Black box, al fine di verificare un corretto funzionamento del comportamento del sistema.

Per quanto riguarda i test effettuati mediante JUnit sono stata testata tutte le funzionalità delle operazioni CRUD, le quali ritenute in questo progetto fondamentali, e le operazioni relative alle prenotazioni, ricerca degli alloggi, gestione delle richieste e visualizzazione di richieste e prenotazioni.

Il codice è anche disponibile sul repository GitHub al seguente URL:
<https://github.com/AntonioImp/OspiteSocial>.