

Gestire i turni

Informazioni generali

Nome caso d'uso: Gestire i turni

Portata: Sistema

Livello: Obiettivo utente

Attore primario: Organizzatore

Parti Interessate: Organizzatore, Chef, Cuoco, Personale di servizio

Pre-condizioni:

- L'attore è autenticato come Organizzatore
- Esistono eventi pianificati nel sistema per cui definire i turni

Garanzie di successo o post-condizioni:

- I turni (preparatori e di servizio) sono correttamente creati, modificati o eliminati nel sistema
- I raggruppamenti di turni preparatori sono definiti (se applicabile)
- I turni di servizio includono i tempi aggiuntivi specificati
- Le ricorrenze dei turni sono correttamente impostate (se applicabile)
- Il personale (cuochi e personale di servizio) può visualizzare i turni disponibili

Scenario principale di successo

#	Attore	Sistema
1	Consulta tabellone dei turni	Mostra tabellone dei turni esistente
2	Opzionalmente crea un nuovo turno di preparazione specificando inizio e fine del turno, data scadenza per la modifica ed eventualmente una sede.	Registra il turno sul tabellone dei turni
3	Opzionalmente crea un nuovo turno di servizio specificando inizio e fine del turno, data scadenza per la modifica, tempo di preparazione e di rigoverno.	Registra il turno sul tabellone dei turni
4	Opzionalmente imposta ricorrenza su turno presente.	Registra la ricorrenza di turni sul tabellone dei turni
5	Opzionalmente , seleziona almeno due turni preparatori esistenti e crea un raggruppamento.	Registra il raggruppamento sul tabellone dei turni
	<i>Opzionalmente ripete dal passo 2 se no continua</i>	
	Termina il caso d'uso	

Eccezione 2a

#	Attore	Sistema
2a.1	Opzionalmente crea un nuovo turno di preparazione specificando inizio e fine del turno, data scadenza per la modifica ed eventualmente una sede.	La sede indicata è satura
	<i>Termina il caso d'uso</i>	

Eccezione 2b

#	Attore	Sistema
2b.1	Opzionalmente crea un nuovo turno di preparazione specificando inizio e fine del turno, data scadenza per la modifica ed eventualmente una sede.	Le date inserite sono inconsistenti
	<i>Termina il caso d'uso</i>	

Estensione 2a

#	Attore	Sistema
2a.1	Modifica turno di preparazione specificando inizio e fine del turno, data scadenza per la modifica ed eventualmente una sede.	Registra il turno modificato sul tabellone dei turni

Eccezione (2a.1) a

#	Attore	Sistema
(2a.1) a.1	Modifica turno di preparazione specificando inizio e fine del turno, data scadenza per la modifica ed eventualmente una sede.	La sede indicata è satura per quell'orario
	<i>Termina il caso d'uso</i>	

Eccezione (2a.1) b

#	Attore	Sistema
(2a.1) b.1	Modifica turno di preparazione specificando inizio e fine del turno, data scadenza per la modifica ed eventualmente una sede.	Le date inserite sono inconsistenti
	<i>Termina il caso d'uso</i>	

Estensione 2b

#	Attore	Sistema
2b.1	Elimina turno di preparazione	Elimina il turno dal tabellone dei turni

Estensione 2c

#	Attore	Sistema
2c.1	Congela turno di preparazione per impedire il ritiro della disponibilità	Registra il turno modificato sul tabellone dei turni

Eccezione 3a

#	Attore	Sistema
3a.1	Opzionalmente crea un nuovo turno di servizio specificando inizio e fine del turno, data scadenza per la modifica, tempo di preparazione e di rigoverno.	Le date inserite sono inconsistenti
	<i>Termina il caso d'uso</i>	

Eccezione 3b

#	Attore	Sistema
3b.1	Opzionalmente crea un nuovo turno di servizio specificando inizio e fine del turno, data scadenza per la modifica, tempo di preparazione e di rigoverno.	Presenza di turni sovrapposti
	<i>Termina il caso d'uso</i>	

Estensione 3a

#	Attore	Sistema
3a.1	Modifica turno di servizio specificando inizio e fine del turno, data scadenza per la modifica, tempo di preparazione e di rigoverno.	Registra il turno modificato sul tabellone dei turni

Eccezione (3a.1) a

#	Attore	Sistema
(3a.1) a.1	Modifica turno di servizio specificando inizio e fine del turno, data scadenza per la modifica, tempo di preparazione e di rigoverno.	Le date inserite sono inconsistenti
	<i>Termina il caso d'uso</i>	

Eccezione (3a.1) b

#	Attore	Sistema
(3a.1) b.1	Modifica turno di servizio specificando inizio e fine del turno, data scadenza per la modifica, tempo di preparazione e di rigoverno.	Presenza di turni sovrapposti
	<i>Termina il caso d'uso</i>	

Estensione 3c

#	Attore	Sistema
3c.1	Congela turno di servizio per impedire il ritiro della disponibilità	Registra il turno modificato sul tabellone dei turni

Eccezione 4a

#	Attore	Sistema
4a.1	Opzionalmente imposta ricorrenza su turno presente.	Presenza di turni sovrapposti
	<i>Termina il caso d'uso</i>	

Estensione 4a

#	Attore	Sistema
4a.1	Elimina una ricorrenza di un turno	Elimina tutte le ricorrenze di un turno.

Estensione 4b

#	Attore	Sistema
4b.1	Modifica una ricorrenza di un turno	Modifica tutte le ricorrenze di un turno.

Eccezione (4b.1)a

#	Attore	Sistema
(4b.1)a.1	Modifica una ricorrenza di un turno	Presenza di turni sovrapposti
	<i>Termina il caso d'uso</i>	

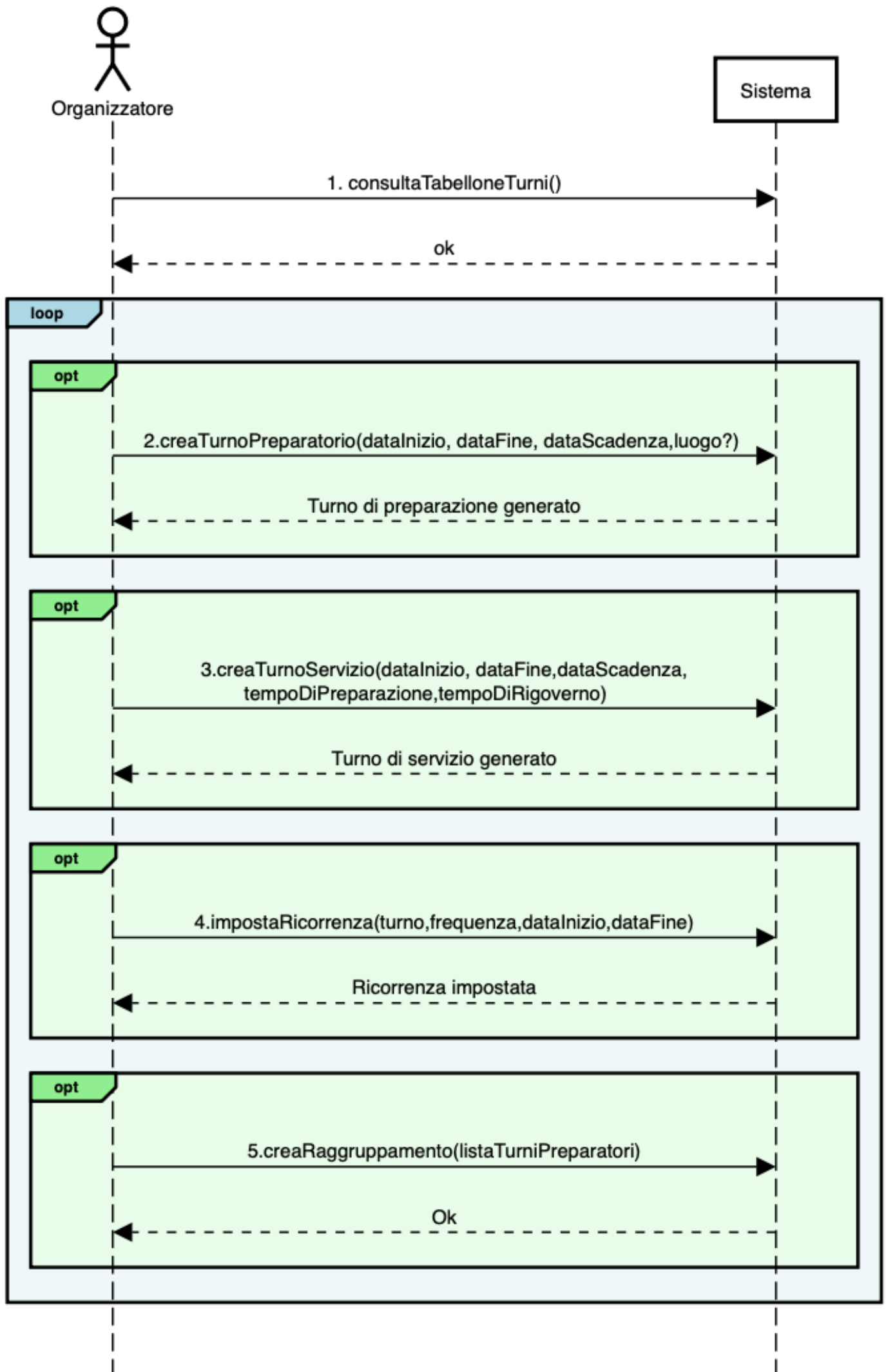
Eccezione (4b.1)b

#	Attore	Sistema
(4b.1)b.1	Modifica una ricorrenza di un turno	Le date inserite sono inconsistenti
	<i>Termina il caso d'uso</i>	

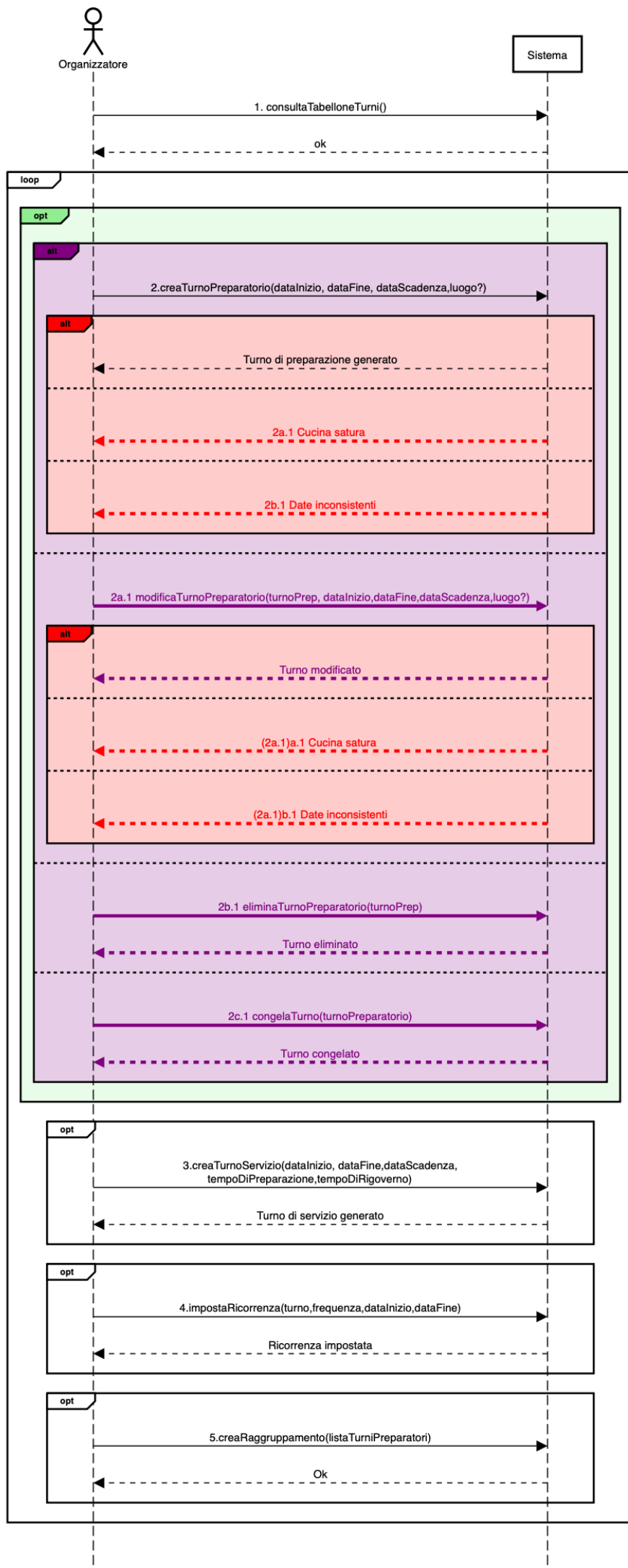
Estensione 5a

#	Attore	Sistema
5a.1	Scioglie un raggruppamento di turni	Elimina il raggruppamento mantenendo i singoli turni.

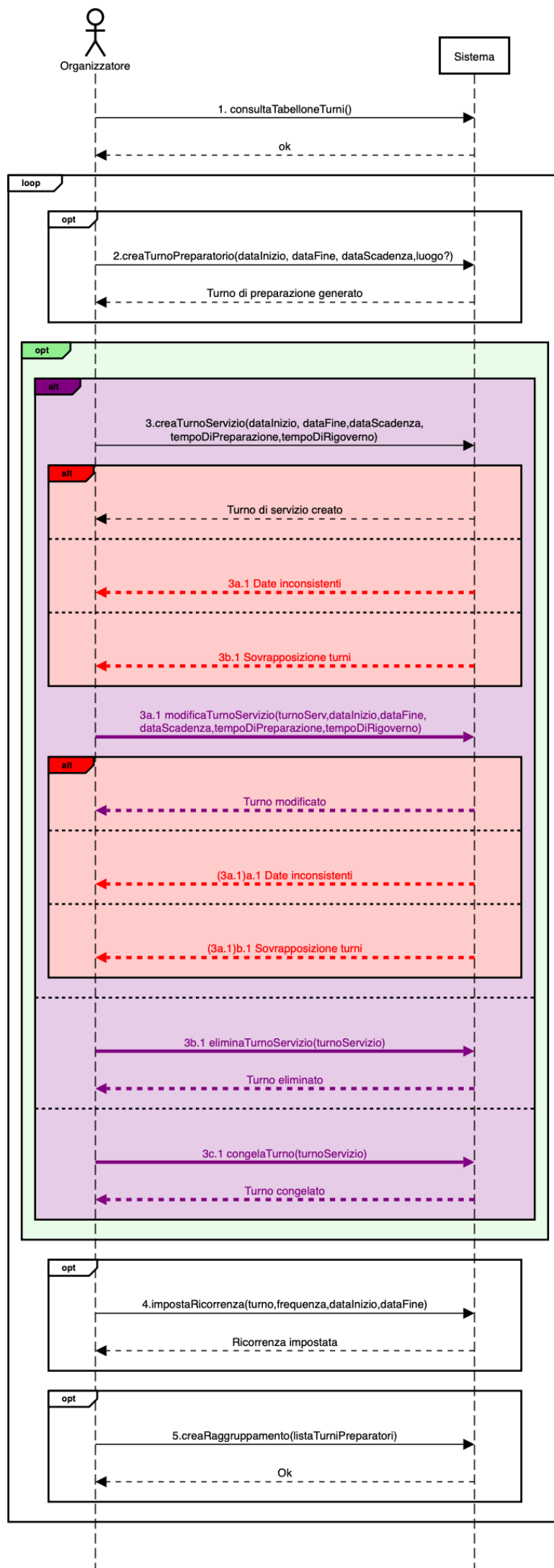
Scenario principale



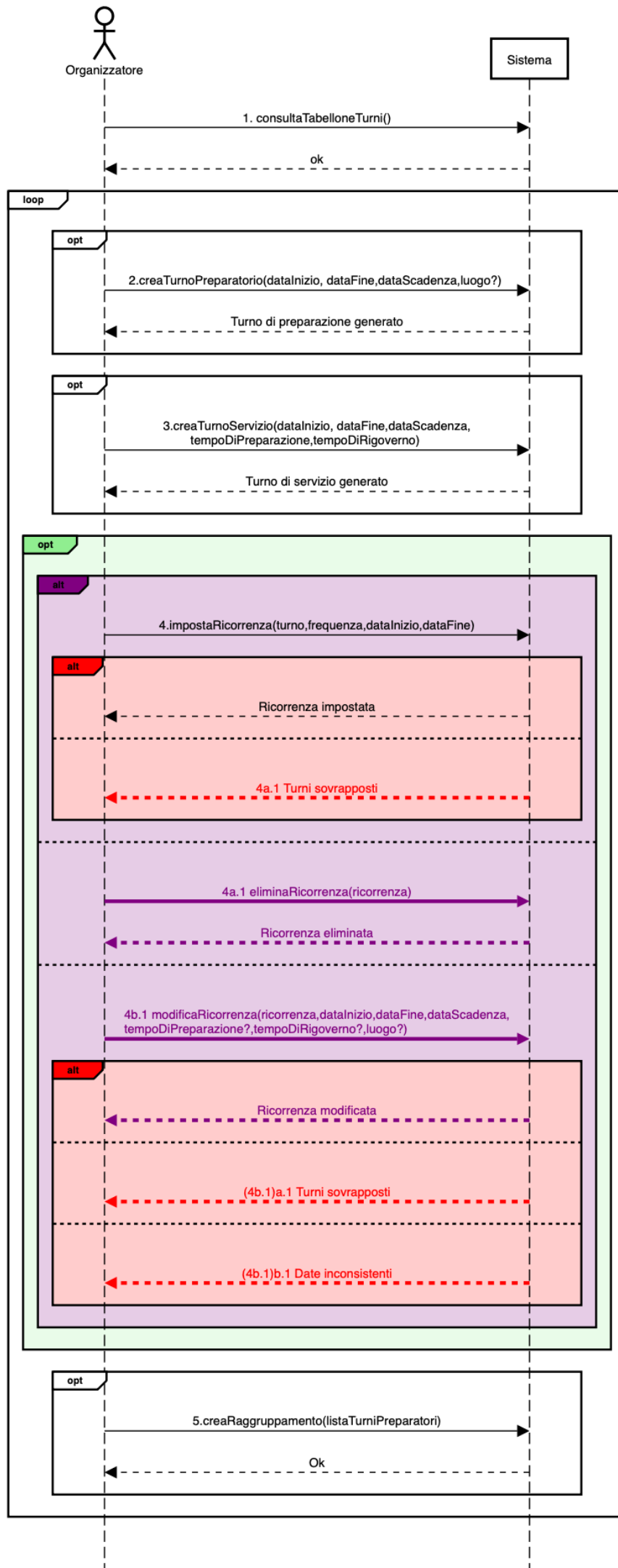
Estensioni al passo 2(2a,2b,2c)
e relative eccezioni (2a,2b,(2a.1)a,(2a.1) b)



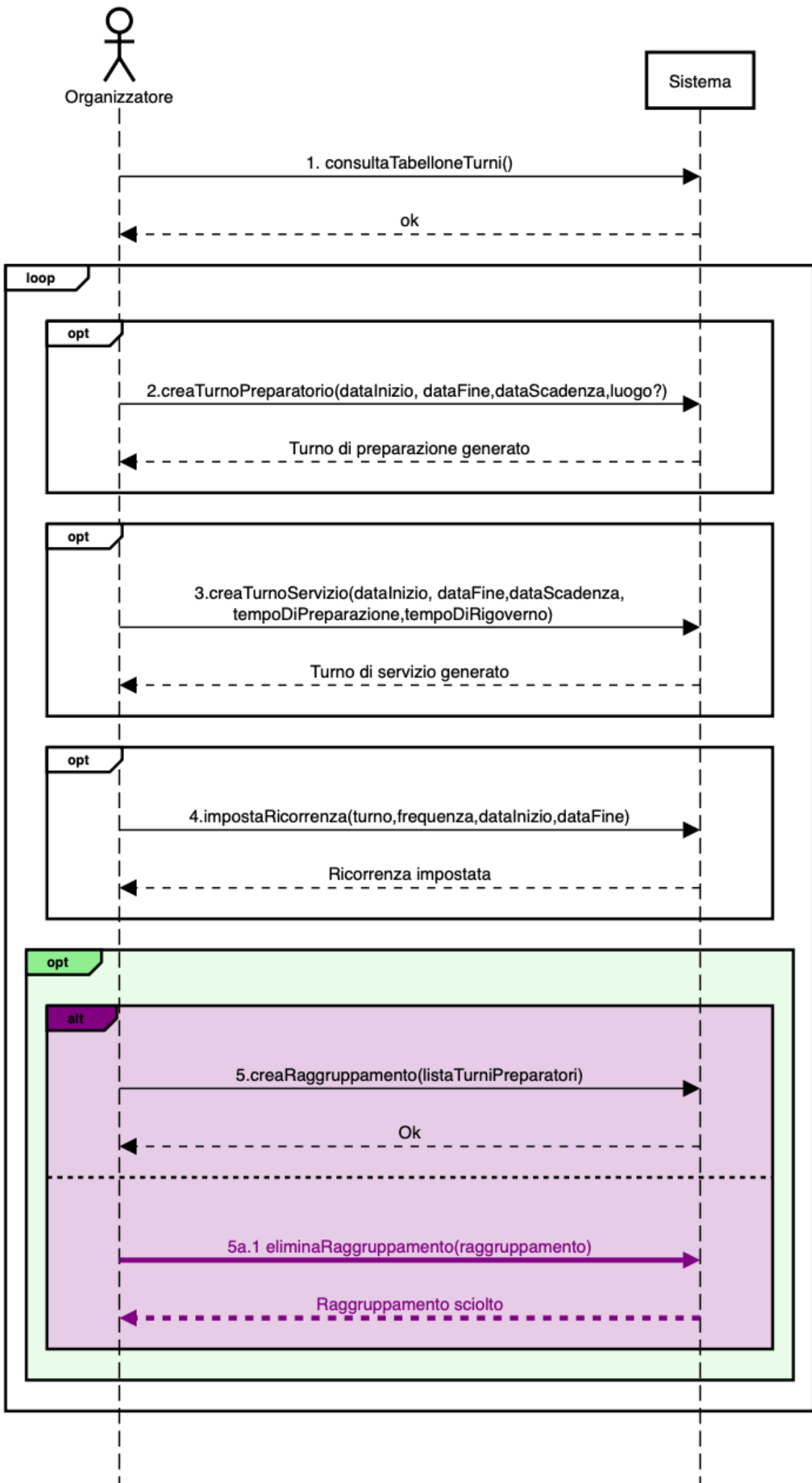
Estensioni al passo 3(3a,3b,3c)
e relative eccezioni (3a,3b,(3a.1)a,(3a.1)b)



Estensioni al passo 4(4a,4b)
e relative eccezioni (4a,(4b.1)a,(4b.1)b)



Estensioni al passo 5(5a)



Contratti per lo UC "Gestire i turni"

Pre-condizione generale: l'attore è identificato con un'istanza *org* di Organizzatore, a *org* è assegnato evento a cui appartiene servizio

1.consultaTabelloneTurni()

Precondizioni:

- esiste un'istanza *tabellone* di Tabellone turni

Post-condizioni: ---

2.creaTurnoPreparatorio(dataInizio:data, dataFine:data,dataScadenza:data,luogo?:SedeCucina)

Pre-condizioni:

Esiste un'istanza *cucinaPred* di SedeCucina

Post-condizioni:

Se dataInizio, dataFine e dataScadenza non sono date passate, dataInizio è antecedente a dataFine e dataScadenza è compresa tra dataInizio e dataScadenza:

Se luogo è specificato e non esiste in *tabellone* un'istanza *turno_i* di TurnoPreparatorio svolto in luogo tale che l'intervallo tra *turno_i.dataInizio* e *turno_i.dataFine* si sovrapponga all'intervallo dataInizio e dataFine:

- è stata creata un'istanza *turnoPreparatorio* di Turno Preparatorio
- *turnoPreparatorio* è presente in *tabellone*
- *turnoPreparatorio.dataInizio* = dataInizio
- *turnoPreparatorio.dataFine* = dataFine
- *turnoPreparatorio.dataScadenza* = dataScadenza
- *turnoPreparatorio.congelato* = no
- *turnoPreparatorio* è svolto in luogo

Se luogo non è specificato e non esiste in *tabellone* un'istanza *turno_i* di TurnoPreparatorio svolto in *cucinaPred* tale che l'intervallo tra *turno_i.dataInizio* e *turno_i.dataFine* si sovrapponga all'intervallo dataInizio e dataFine:

- è stata creata un'istanza *turnoPreparatorio* di Turno Preparatorio
- *turnoPreparatorio* è presente in *tabellone*
- *turnoPreparatorio.dataInizio* = dataInizio
- *turnoPreparatorio.dataFine* = dataFine
- *turnoPreparatorio.dataScadenza* = dataScadenza
- *turnoPreparatorio.congelato* = no
- *turnoPreparatorio* è svolto in *cucinaPred*

2a.1 modificaTurnoDiPreparazione(turnoPreparatorio:TurnoDiPreparazione, dataInizio:data, dataFine:data, dataScadenza:data, luogo?:SedeCucina)

Pre-condizioni:

- turnoPreparatorio è presente in *tabellone*
- esiste un'istanza di SedeCucina associata a turnoPreparatorio.

Post-condizioni:

Se dataInizio, dataFine e dataScadenza non sono date passate, dataInizio è antecedente a dataFine dataScadenza è compresa tra dataInizio e dataScadenza e *turnoPreparatorio.congelato* = no:

Se luogo è specificato e non esiste in *tabellone* un'istanza *turno_i* di TurnoPreparatorio svolto in luogo tale che l'intervallo tra *turno_i.dataInizio* e *turno_i.dataFine* si sovrapponga all'intervallo dataInizio e dataFine:

- è stata modificata l'istanza turnoPreparatorio
- turnoPreparatorio è presente in *tabellone*
- *turnoPreparatorio.dataInizio* = dataInizio
- *turnoPreparatorio.dataFine* = dataFine
- *turnoPreparatorio.dataScadenza* = dataScadenza
- *turnoPreparatorio* è svolto in luogo

Se luogo non è specificato e non esiste in *tabellone* un'istanza *turno_i* di TurnoPreparatorio, associata alla stessa istanza SedeCucina di turnoPreparatorio, tale che l'intervallo tra *turno_i.dataInizio* e *turno_i.dataFine* si sovrapponga all'intervallo dataInizio e dataFine:

- è stata modificata l'istanza turnoPreparatorio
- turnoPreparatorio è presente in *tabellone*
- *turnoPreparatorio.dataInizio* = dataInizio
- *turnoPreparatorio.dataFine* = dataFine
- *turnoPreparatorio.dataScadenza* = dataScadenza

Se turnoPreparatorio è contenuto in un'istanza *ric* di Ricorrenza questa viene modificata.

2b.1 eliminaTurnoDiPreparazione(turnoPreparatorio:TurnoDiPreparazione)

Pre-condizioni:

- turnoPreparatorio è presente in *tabellone*

Post-condizioni:

- l'istanza turnoPreparatorio viene eliminata
- se turnoPreparatorio era contenuta in un'istanza *ric* di Ricorrenza questa viene eliminata

2c.1 congelaTurnoPreparatorio(turnoPreparatorio:TurnoDiPreparazione)

Pre-condizioni:

- turnoPreparatorio è presente in *tabellone* e *turnoPreparatorio.congelamento* = no

Post-condizioni:

- Il turnoPreparatorio viene congelato e non è più possibile ritirare una disponibilità
- *turnoPreparatorio.congelamento* = si
- Se turnoPreparatorio è contenuto in un'istanza *ric* di Ricorrenza questa viene congelato.

3.creaTurnoServizio(dataInizio:data, dataFine:data, dataScadenza:data, tempoDiPreparazione:numero, tempoDiRigoverno:numero)

Pre-condizioni: ---

Post-condizioni:

Se dataInizio, dataFine e dataScadenza non sono date passate, dataInizio è antecedente a dataFine e dataScadenza è compresa tra dataInizio e dataScadenza:

Se non esiste in *tabellone* un'istanza *turno_i* di TurnoDiServizio tale che l'intervallo tra *turno_i.dataInizio* e *turno_i.dataFine* si sovrapponga all'intervallo dataInizio e dataFine:

- è stata creata un'istanza *turnoDiServizio* di TurnoDiServizio
- *turnoDiServizio* è presente in *tabellone*
- *turnoDiServizio.dataInizio* = dataInizio
- *turnoDiServizio.dataFine* = dataFine
- *turnoDiServizio.dataScadenza* = dataScadenza
- *turnoDiServizio.tempodiPreparazione* = tempoDiPreparazione
- *turnoDiServizio.tempodiRigoverno* = tempoDiRigoverno
- *turnoPreparatorio.congelato* = no

3a.1 modificaTurnoServizio(turnoDiServizio:TurnoDiServizio, dataInizio:data, dataFine:data, dataScadenza:data, tempoDiPreparazione:numero, tempoDiRigoverno:numero)

Pre-condizioni:

- turnoDiServizio è presente in *tabellone*

Post-condizioni:

Se dataInizio, dataFine e dataScadenza non sono date passate, dataInizio è antecedente a dataFine e dataScadenza è compresa tra dataInizio e dataScadenza:

Se non esiste in *tabellone* un'istanza *turno_i* di TurnoDiServizio tale che l'intervallo tra *turno_i.dataInizio* e *turno_i.dataFine* si sovrapponga all'intervallo dataInizio e dataFine:

- è stata modificata l'istanza turnoDiServizio
- turnoDiServizio è presente in *tabellone*
- *turnoDiServizio.dataInizio* = dataInizio
- *turnoDiServizio.dataFine* = dataFine
- *turnoDiServizio.dataScadenza* = dataScadenza
- *turnoDiServizio.tempoDiPreparazione* = tempoDiPreparazione
- *turnoDiServizio.tempoDiRigoverno* = tempoDiRigoverno
- *turnoPreparatorio.congelato* = no

Se turnoDiServizio è contenuto in un'istanza *ric* di Ricorrenza questa viene modificata.

3b.1 eliminaTurnoServizio (turnoServizio: TurnoDiServizio)

Pre-condizioni:

- turnoServizio è presente in *tabellone*

Post-condizioni:

- l'istanza turnoServizio viene eliminata
- se turnoServizio era contenuta in un'istanza *ric* di Ricorrenza questa viene eliminata

3c.1 congelaTurnoDiServizio(turnoServizio: TurnoDiServizio)

Pre-condizioni:

- turnoServizio è presente in *tabellone* e *turnoServizio.congelato* = no

Post-condizioni:

- Il turnoServizio viene congelato e non è più possibile ritirare una disponibilità
- *turnoServizio.congelato* = si

Se turnoDiServizio è contenuto in un'istanza *ric* di Ricorrenza questa viene congelata.

4.impostaRicorrenza(turno:Turno,frequenza:testo,dataInizio:data,dataFine:data)

Pre-condizioni:

turno è presente in *tabellone*, dataInizio e dataFine non sono date passate, dataInizio è antecedente a dataFine e frequenza è una frequenza valida (giornaliera, settimanale, mensile).

Post-condizioni:

Se non esiste un Turno *t* tale che si sovrapponga ad una ricorrenza di turno:

- È stata creata una Ricorrenza *r* associata a turno
- *r.frequenza* = frequenza
- *r.dataInizio* = dataInizio
- *r.dataFine* = dataFine

- Per ogni ricorrenza di *turno* da dataInizio a dataFine crea un'istanza *turno_i* dove:
 - *turno_i.dataInizio* = *turno.dataInizio*
 - *turno_i.dataFine* = *turno.dataFine*
 - *turno_i.dataScadenza* = *turno.dataScadenza*
 - [se turno è un Turno di preparazione] *turno_i. luogoPreparazione* = *turno. luogoPreparazione*
 - [se turno è un Turno di servizio] *turno_i. tempoDiPreparazione* = *turno. tempoDiPreparazione*
 - [se turno è un Turno di servizio] *turno_i. tempoDiRigoverno* = *turno. tempoDiRigoverno*

4.a eliminaRicorrenza(ric:Ricorrenza)

Pre-condizioni:

Post-condizioni:

Ogni *turno_i* che ricorre in ric viene eliminato
L'istanza *ric* viene eliminata

4.b modificaRicorrenza(ric:Ricorrenza, dataInizio:data, dataFine:data, dataScadenza:data, tempoDiPreparazione?:numero, tempoDiRigoverno?:numero, luogo?:SedeCucina)

Pre-condizioni:

Se si vogliono modificare dei turni di servizio occorre definire tempoDiPreparazione e tempoDiRigoverno e lasciare non definito luogo. In alternativa se si vogliono modificare dei turni di preparazione occorre non definire tempoDiPreparazione e tempoDiRigoverno mentre luogo è opzionale.

Post-condizioni:

Se dataInizio, dataFine e dataScadenza non sono date passate, dataInizio è antecedente a dataFine e dataScadenza è compresa tra dataInizio e dataScadenza
Se non esiste un Turno *t* tale che si sovrapponga ad un *turno_i* che ricorre in *ric* dopo la modifica:

Ogni *turno_i* che ricorre in ric viene modificato:

- *turno_i.dataInizio* = *turno.dataInizio* + *ric.frequenza* * numero della ricorrenza
- *turno_i.dataFine* = *turno.dataFine* + *ric.frequenza* * numero della ricorrenza
- *turno_i.dataScadenza* = *turno.dataScadenza*
- [se turno è un Turno di preparazione] *turno_i. luogoPreparazione* = *turno. luogoPreparazione*
- [se turno è un Turno di servizio] *turno_i. tempoDiPreparazione* = *turno. tempoDiPreparazione*
- [se turno è un Turno di servizio] *turno_i. tempoDiRigoverno* = *turno. tempoDiRigoverno*

5. creaRaggruppamento(listaTurniPreparatori: lista TurnoDiPreparazione)

Precondizioni:

- Tutti i turni in listaTurniPreparatori sono istanze di TurnoDiPreparazione
- listaTurniPreparatori contiene almeno 2 istanze di TurnoDiPreparazione

Post-condizioni:

- È stato creato un nuovo Raggruppamento r
- Tutti i turni in listaTurniPreparatori sono contenuti in r

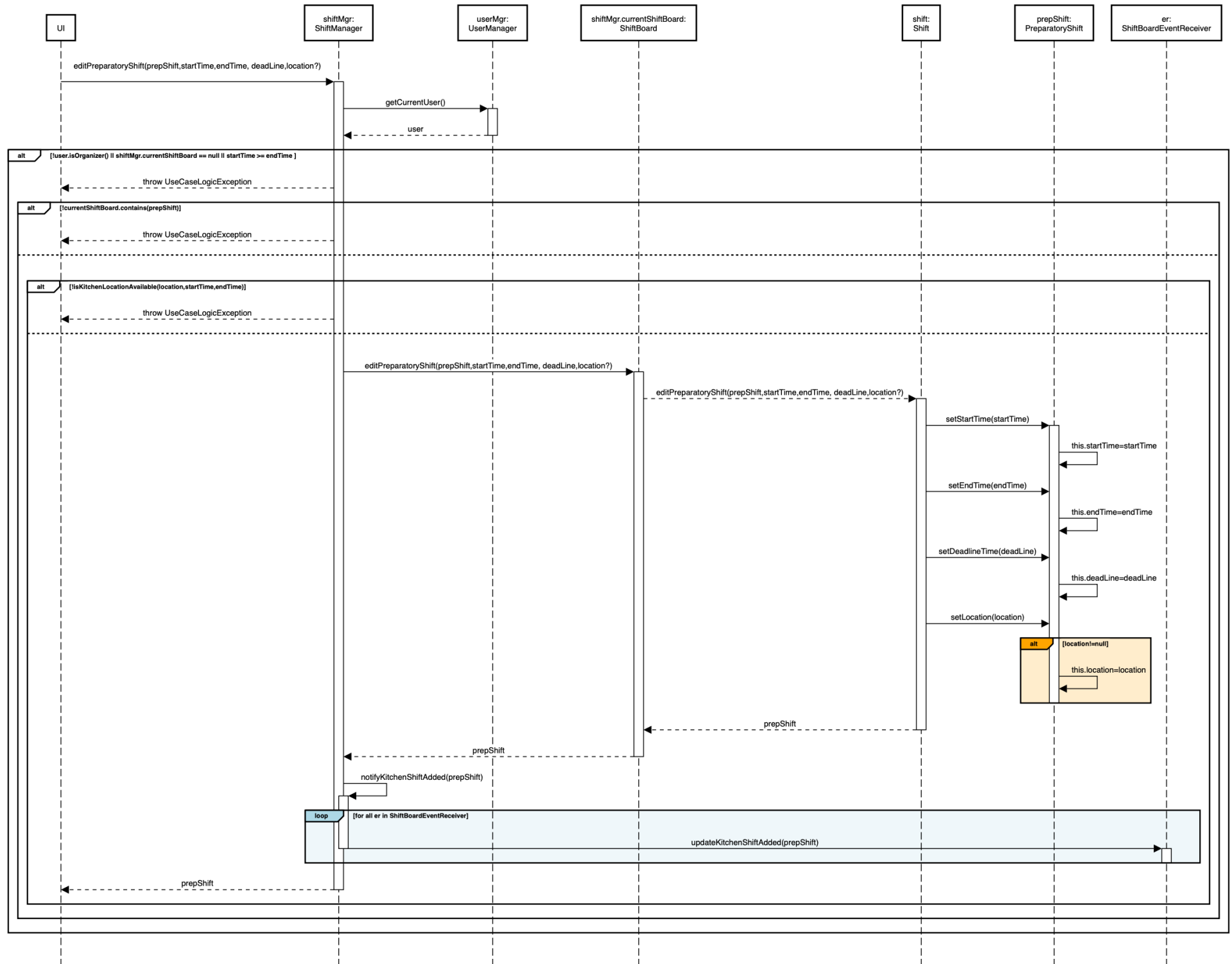
5a.1 sciogliRaggruppamento(raggruppamento: Raggruppamento)

Precondizioni:

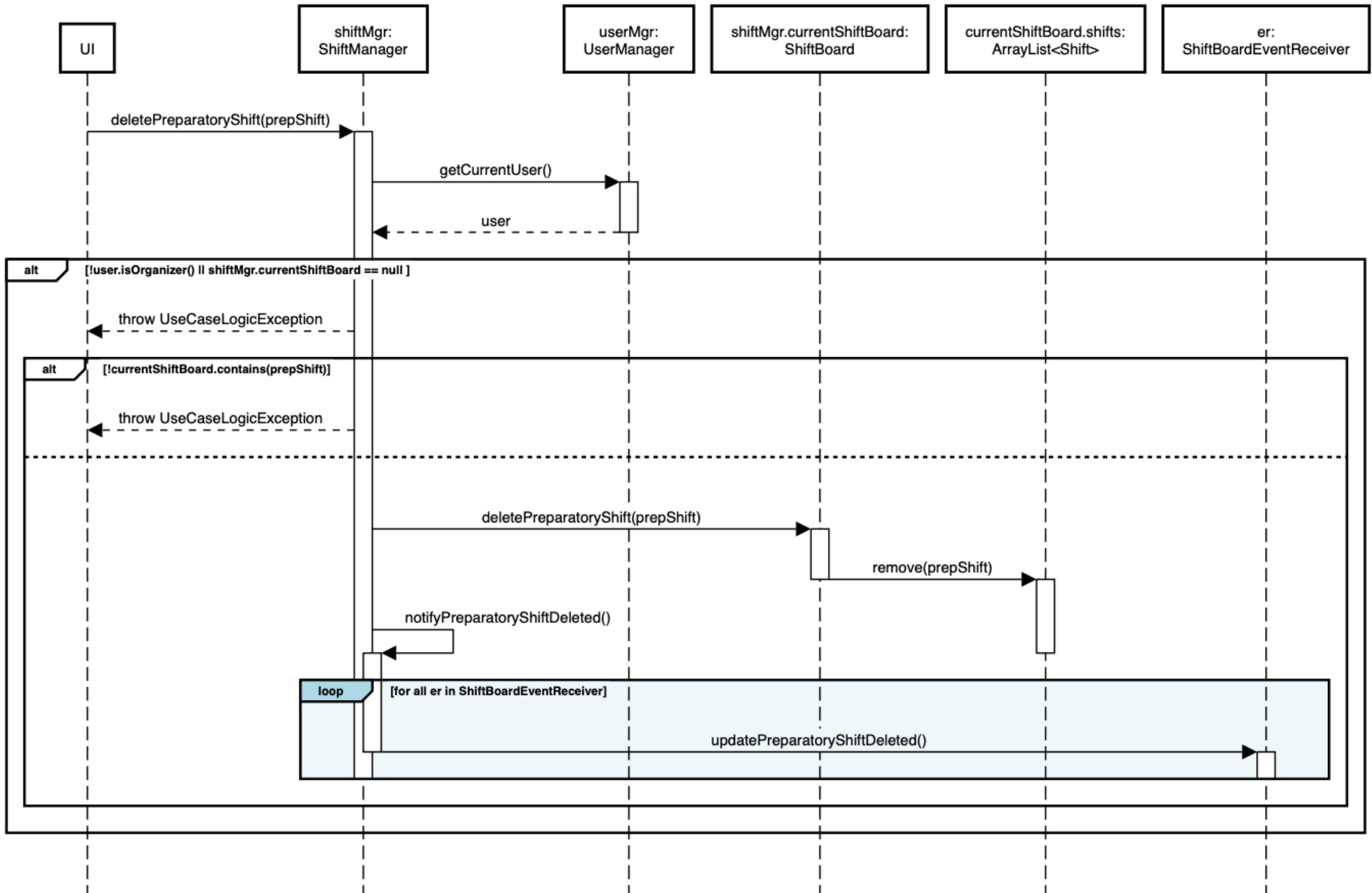
Post-condizioni:

- Per ogni TurnoDiPreparazione t che è contenuto in raggruppamento, l'associazione tra raggruppamento a t è eliminata.
- raggruppamento è eliminata

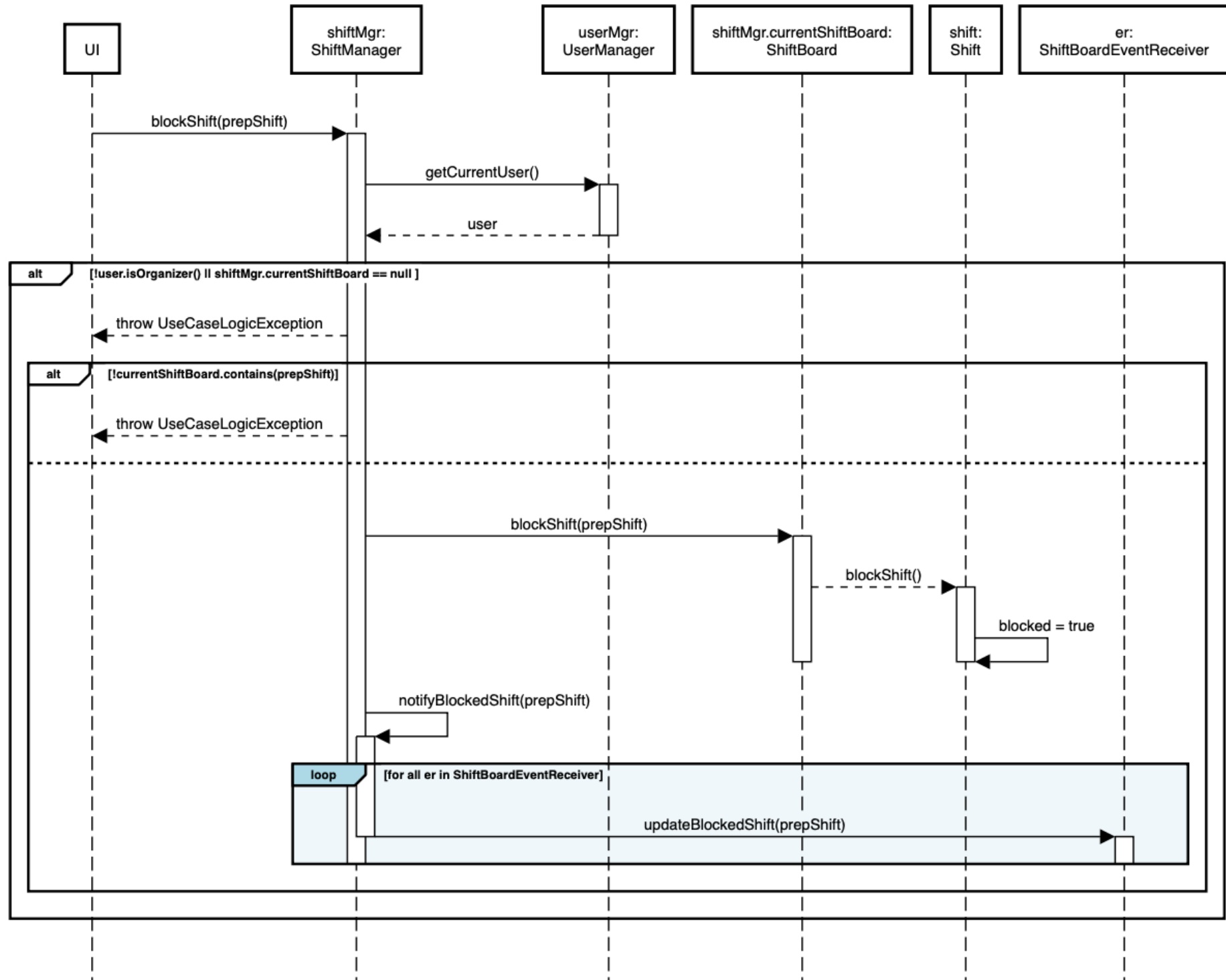
2a.1 editPreparatoryShift



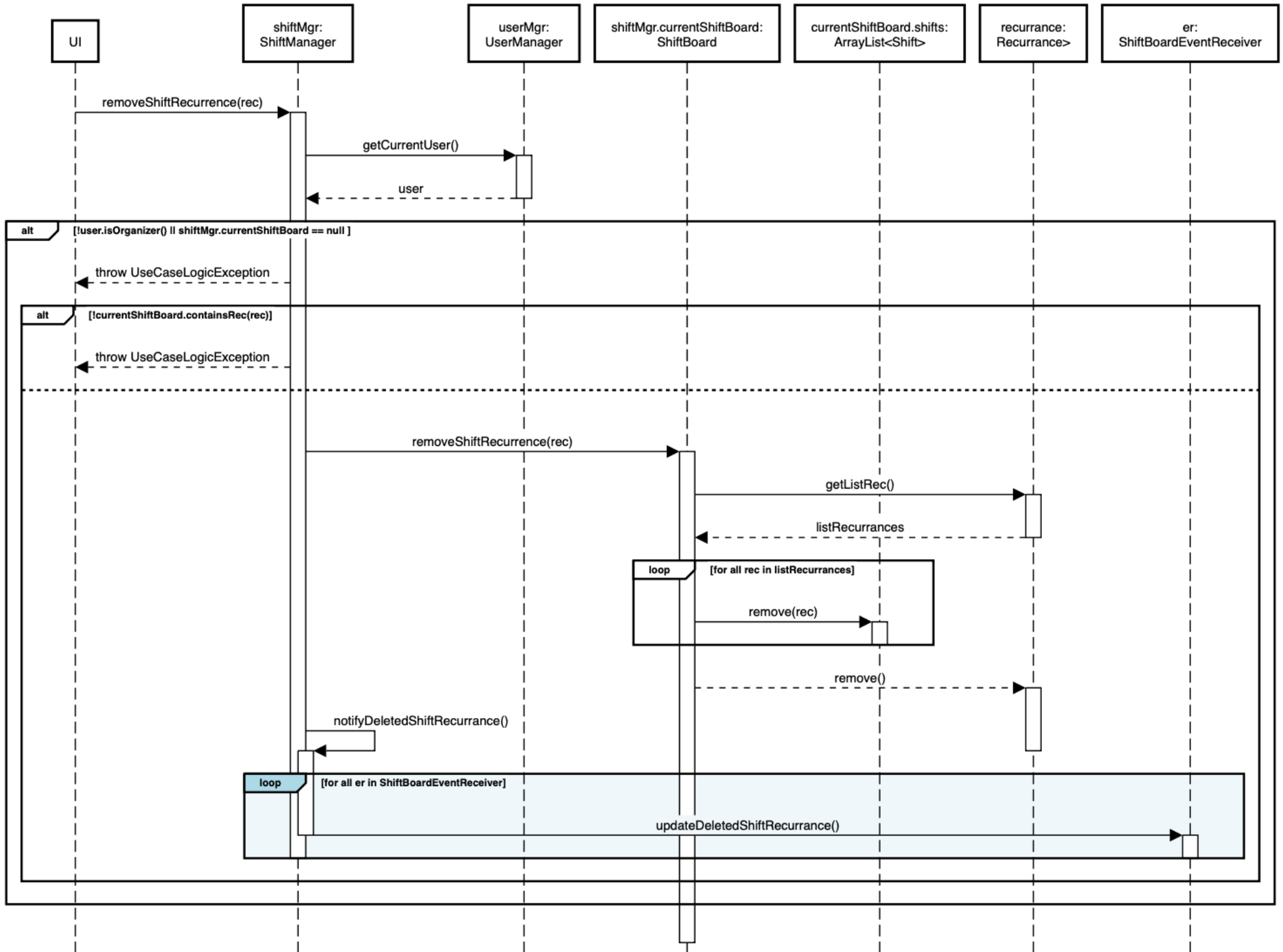
2b.1 deletePreparatoryShift



2c.1 blockPrepShift



4a.1 removeShiftRecurrence



```

classDiagram
    class UseCaseLogicException
    class Exception
    UseCaseLogicException --|> Exception

```

