

Università degli Studi di Torino
Corso di Laurea Triennale in Informatica

Corso Sviluppo delle Applicazioni Software (9 CFU)
Anno accademico 2023-2024

Progettazione di Laboratorio

Gestire i Turni

Rinaldi Flavio (m.955750)
Persico Loris (m.949352)



UNIVERSITÀ
DI TORINO

Gestire i turni

Informazioni generali

Nome caso d'uso: Gestire i turni

Portata: Sistema

Livello: Obiettivo utente

Attore primario: Organizzatore

Parti Interessate: Cuoco, Personale di Servizio

Pre-condizioni: L'utente deve essere autenticato come Organizzatore di un evento, e deve esistere almeno un evento a cui lavorare

Garanzie di successo o post-condizioni: Sono stati aggiunti uno o più turni al calendario; eventualmente sono stati modificati o eliminati dei turni; eventualmente sono stati creati dei raggruppamenti di turni

Scenario principale di successo

#	Attore	Sistema
1	Consulta il tabellone dei turni	Restituisce il calendario del tabellone dei turni
	<i>Se vuole prosegue col passo 2, altrimenti salta al passo 5</i>	
2	Inserisce nel calendario un turno preparatorio, specificando data, ora di inizio, ora di fine, ed eventualmente una sede	Registra il nuovo turno nel calendario dei turni
3	<i>Opzionalmente</i> indica per il turno creato una data di scadenza per la modifica delle disponibilità	Registra le informazioni aggiuntive nel calendario dei turni
4	<i>Opzionalmente</i> raggruppa un insieme di turni	Registra il raggruppamento di turni nel calendario dei turni
5	<i>Opzionalmente</i> modifica un turno preparatorio, specificando una nuova data, un nuovo orario di inizio o un nuovo orario di fine, una sede	Registra le modifiche relative al turno nel calendario dei turni

	differente oppure una data di scadenza per la modifica delle disponibilità	
	<i>Ripete 5 finché non è soddisfatto</i>	
6	<i>Opzionalmente</i> elimina un turno dal calendario	Elimina le informazioni relative al turno dal calendario di turni
	<i>Se vuole torna al passo 2, altrimenti termina</i>	

Eccezione 2a

#	Attore	Sistema
2a.1	Inserisce nel calendario un turno preparatorio, specificando data, ora di inizio, ora di fine, ed eventualmente una sede	Sul calendario si verifica una sovrapposizione di turni preparatori

Eccezione 2b

#	Attore	Sistema
2a.1	Inserisce nel calendario un turno preparatorio, specificando data, ora di inizio, ora di fine, ed eventualmente una sede	La data selezionata è nel passato

Estensione 2a

#	Attore	Sistema
2a.1	Consulta la lista di eventi attualmente disponibili	Restituisce la lista di eventi con le relative informazioni
2a.2	Inserisce nel calendario un turno di servizio, specificando data, ora di inizio, ora di fine, e l'evento cui il turno fa riferimento	Registra il nuovo turno nel calendario dei turni

Eccezione 2a.2a

#	Attore	Sistema
2a.2a.1	Inserisce nel calendario un turno di servizio, specificando data, ora di inizio, ora di fine, e l'evento cui il turno fa riferimento	la data selezionata è nel passato

Estensione 2b

#	Attore	Sistema
2b.1	Indica una ricorrenza per il turno creato specificando frequenza, data di inizio e data di fine	Registra le informazioni aggiuntive nel calendario dei turni

Eccezione 3a

#	Attore	Sistema
3a	<i>Opzionalmente</i> indica per il turno creato una data di scadenza per la modifica delle disponibilità	la data selezionata è nel passato

Eccezione 5a

#	Attore	Sistema
5a	<i>Opzionalmente</i> modifica un turno preparatorio, specificando una nuova data, un nuovo orario di inizio o un nuovo orario di fine, una sede	la data selezionata è nel passato

Eccezione 5b

#	Attore	Sistema
5b	<i>Opzionalmente</i> modifica un turno preparatorio, specificando una nuova data, un nuovo orario di inizio o un nuovo orario di fine, una sede	Sul calendario si verifica una sovrapposizione di turni preparatori

Estensione 5a

#	Attore	Sistema
5a	<i>Opzionalmente</i> modifica un turno di servizio, specificando una nuova data, un nuovo orario di inizio o un nuovo orario di fine e una data scadenza disponibilità	Registra le modifiche relative al turno nel calendario dei turni

Estensione 5b

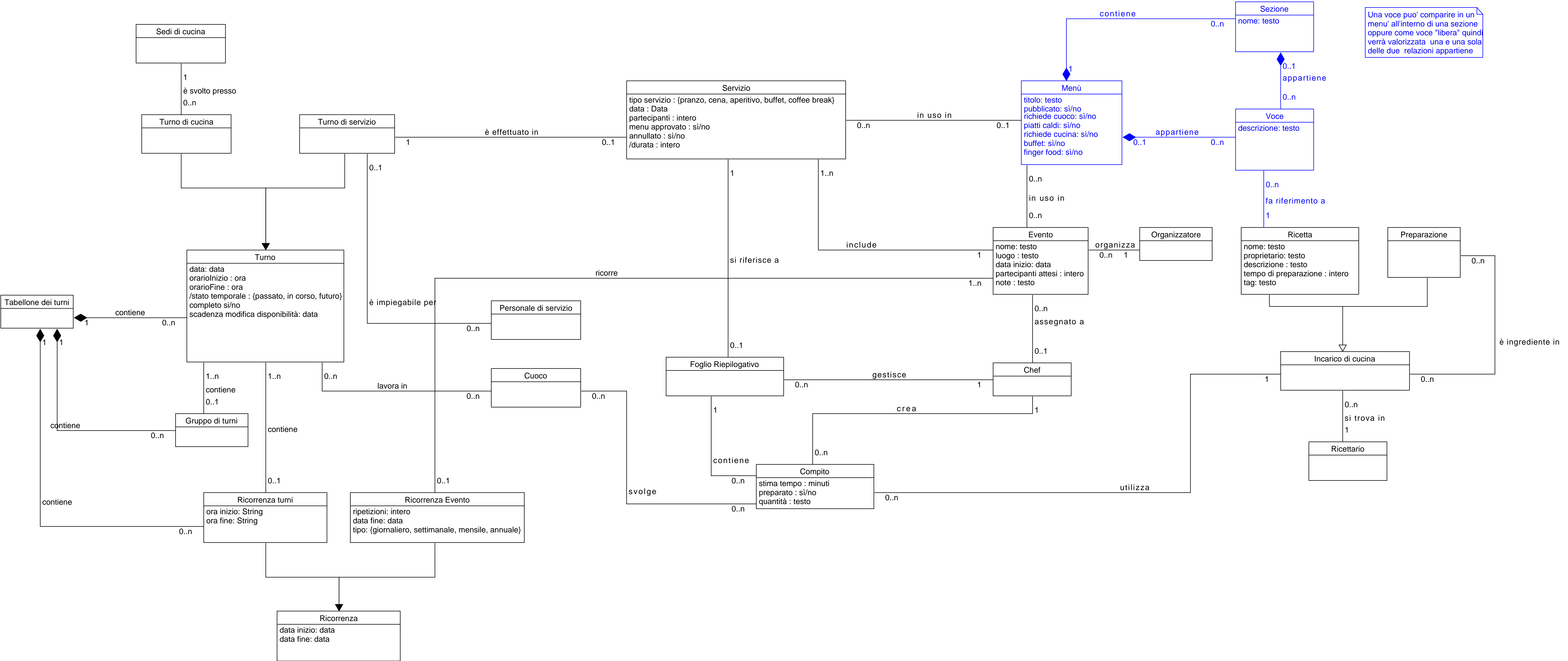
#	Attore	Sistema
5b	Modifica una ricorrenza di turni	Registra le modifiche

Estensione 6a

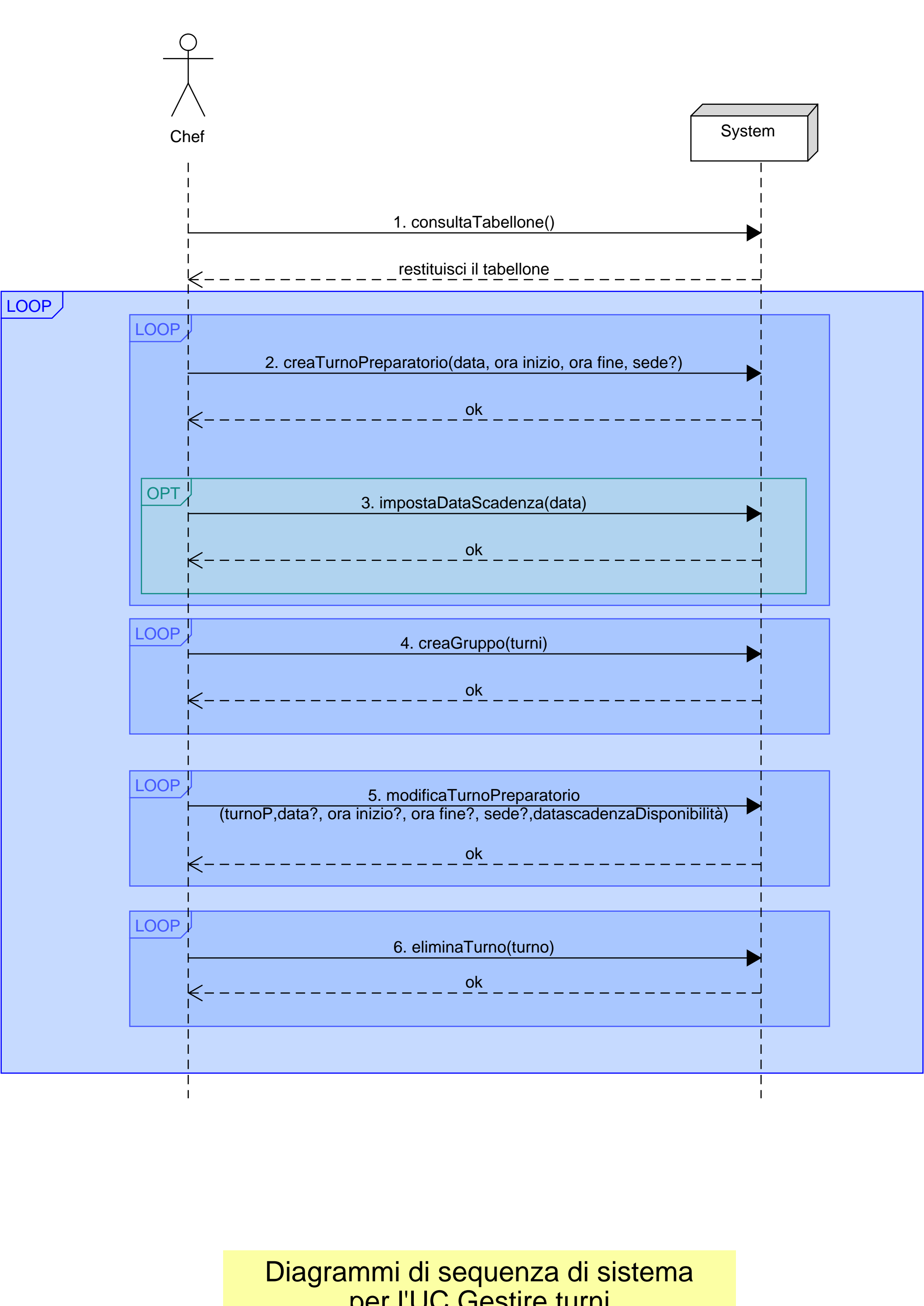
#	Attore	Sistema
6a	Elimina una ricorrenza di turni	Rimuove dal calendario le informazioni relative alla ricorrenza, mantenendo i singoli turni

Estensione 6c

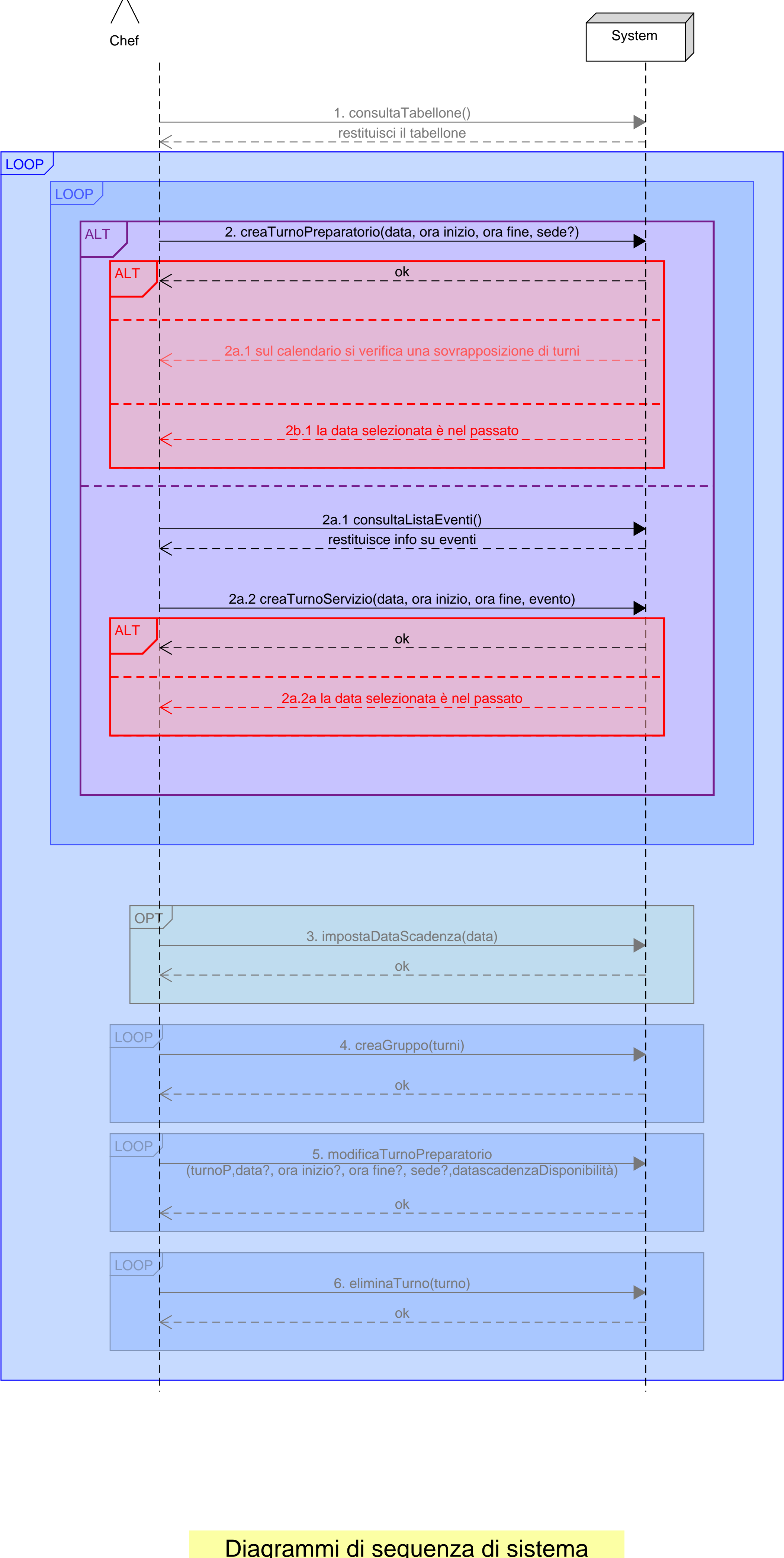
#	Attore	Sistema
6a	Elimina un raggruppamento di turni	Registra le modifiche



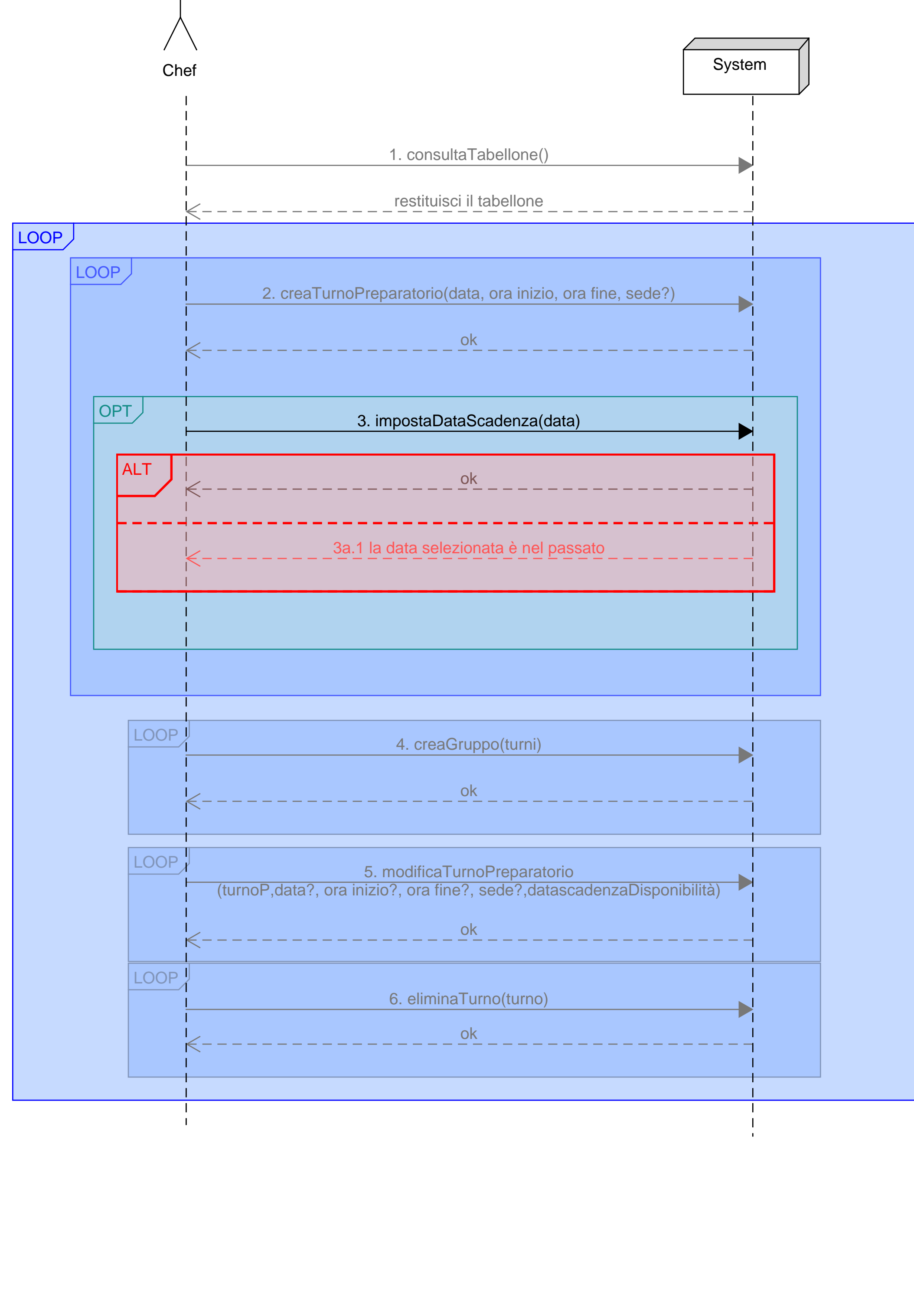
Diagrammi di sequenza di sistema per l'UC Gestire turni



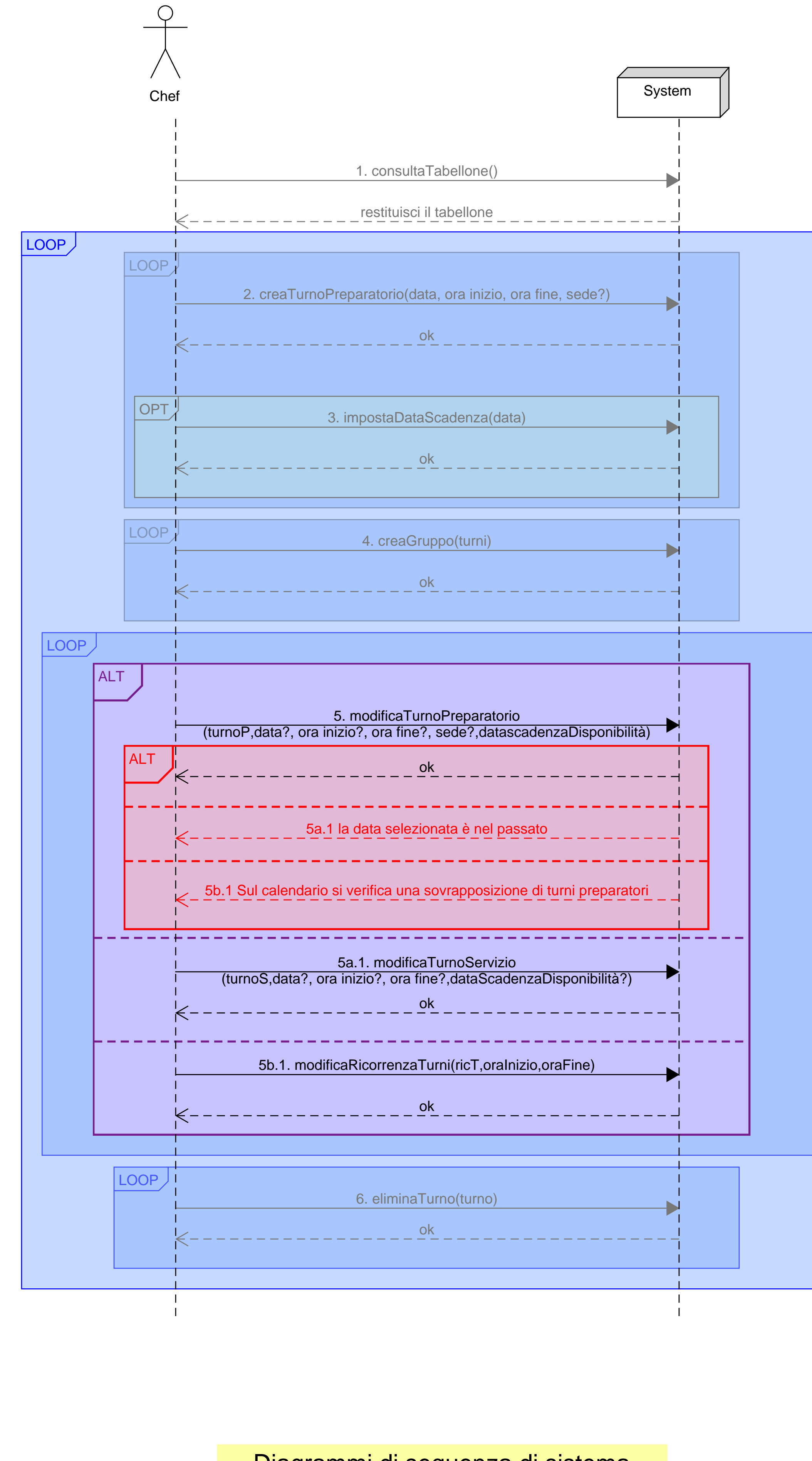
Diagrammi di sequenza di sistema per l'UC Gestire turni



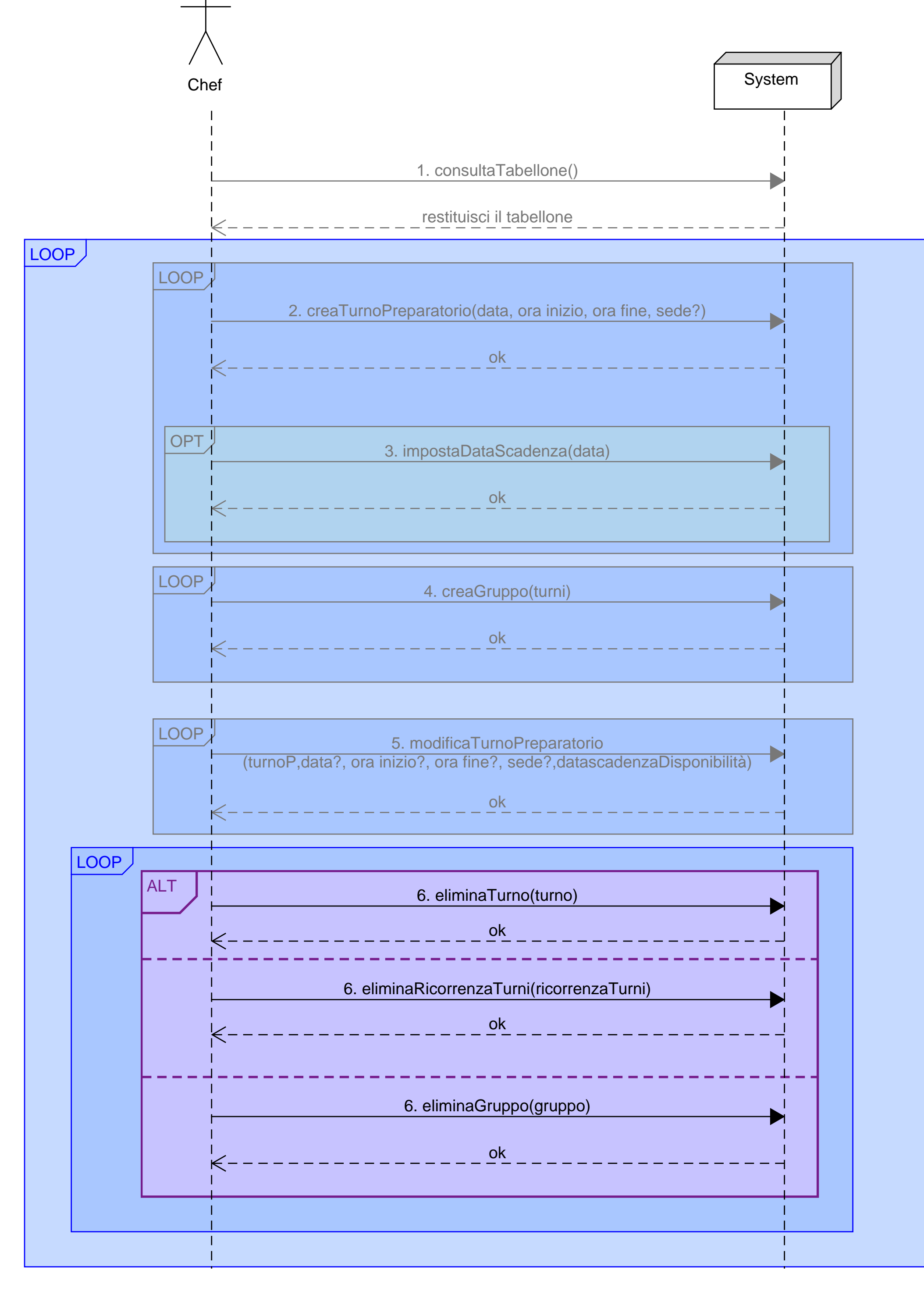
Diagrammi di sequenza di sistema per l'UC Gestire turni



Diagrammi di sequenza di sistema per l'UC Gestire turni



Diagrammi di sequenza di sistema per l'UC Gestire turni



Contratti per l'UC gestire i turni

Pre-condizioni generali : L'attore è identificato con un'istanza org di Organizzatore ed esiste un'unica istanza tabellone di TabelloneTurni

1.consultaTabellone()

Pre-condizioni: - - -

Post-condizioni: - - -

2.creaTurnoPreparatorio(data:data, oraInizio: numero, oraFine: numero, sede?:SedeCucina)

Pre-condizioni:

- Esiste un istanza sedePrinc di SedeCucina

Post-condizioni:

- Se **data** non è nel passato:
 - Se sede è definito e non esiste un'istanza *t2* di TurnoPreparatorio tale che *tabellone* **contiene** *t2*, *t2* **è svolto presso** sede:
 - è stata creata un'istanza *t* di TurnoPreparatorio
 - *t.data* = data
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine
 - *t* è svolto presso sede
 - *tabellone* contiene *t*
 - Se sede non è definito e non esiste un'istanza *t2* di TurnoPreparatorio tale che *tabellone* **contiene** *t2*, *t2* **è svolto presso** *sedePrinc*:
 - è stata creata un'istanza *t* di TurnoPreparatorio
 - *t.data* = data
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine
 - *t* è svolto presso *sedePrinc*
 - *tabellone* contiene *t*

2a.1. consultaListaEventi()

Pre-condizioni: - - -

Post-condizioni: – -

2a.2. creaTurnoServizio(data: data, oraInizio: numero, oraFine: numero, evento: Evento)

Pre-condizioni: - - -

Post-condizioni:

- Se data non è nel passato
 - è stata creata un'istanza *t* di TurnoServizio
 - *t.data* = data
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine
 - *t* è riferito a evento
 - *tabellone* **contiene** *t*

2b.1 impostaRicorrenza(frequenza, dataInizio: data, dataFine: data)

Pre-condizioni:

- è in corso la definizione di un Turno *t*

Post-condizioni:

- è stata creata un'istanza *ric* di RicorrenzaTurni
- *ric.dataInizio* = dataInizio
- *ric.dataFine* = dataFine
- *tabellone* **contiene** *ric*

3.impostaDataScadenza(data:data)

Pre-condizioni:

- è in corso la definizione di un Turno *t*

Post-condizioni:

- **Se data non è nel passato:**
 - *t.scadenzaModificadisponibilità* = data

4.creaGruppi(lista T di TurnoPreparatorio)

Pre-condizioni:

- Per ogni TurnoPreparatorio *t* in T non esiste alcun GruppoTurni *g* tale che *g* **contiene** *t*

Post-condizioni:

- è stata creata un'istanza *gruppo* di GruppoTurni
- Per ogni Turno *t* di T:
 - *gruppo* **contiene** *t*

- *tabellone contiene gruppo*

5.modificaTurnoPreparatorio(turnoP:TurnoPreparatorio, data?:data, oraInizio?:numero, oraFine?:numero, sede?:SedeCucina, dataScadenzaDisponibilita?: data)

Pre-condizioni:

- esiste almeno un'istanza *t* di TurnoPreparatorio

Post-condizioni:

- Se non esiste alcun TurnoPreparatorio *t2*, tali che *t2.data* sia uguale a turnoP.data e *t2.oraFine* sia successivo a turnoP.oraInizio (cioè tali da non creare una sovrapposizione di turni preparatori) :
 - [se data è definito e data non è nel passato]
 - turnoP.data = data
 - [se oraInizio e oraFine sono definiti]
 - turnoP.oraInizio = oraInizio
 - turnoP.oraFine = oraFine
 - [se sede è definito ed esiste una SedeCucina *sc* tale che turnoP è svolto presso *sc*]:
 - l'associazione **svolto presso** tra turnoP e *sc* è eliminata
 - turnoP è svolto presso sede
 - [se dataScadenzaDisponibilita è definito]
 - turnoP.scadenzaModificaDisponibilita = dataScadenzaDisponibilita

5a.1.modificaTurnoServizio(turnoS:TurnoServizio, data?:data, oraInizio?:numero, oraFine?:numero, dataScadenzaDisponibilita?: data)

Pre-condizioni: esiste almeno un'istanza *ts* di TurnoServizio

Post-condizioni:

- [se data è definito e data non è nel passato]
 - turnoS.data = data
- [se oraInizio e oraFine sono definiti]
 - turnoS.oraInizio = oraInizio
 - turnoS.oraFine = oraFine
- [se dataScadenzaDisponibilita è definito]
 - turnoS.scadenzaModificaDisponibilita = dataScadenzaDisponibilita

5b.1.modificaRicorrenzaTurni(ricT:RicorrenzaTurni, oraInizio:numero, oraFine:numero)

Pre-condizioni:- - -

Post-condizioni:

- ricT.oraInizio = oraInizio
- ricT.oraFine = oraFine
- Per ogni Turno t tale che t **ricorre** in ricT:
 - $t.oraInizio$ = oraInizio
 - $t.oraFine$ = oraFine

6. eliminaTurno(turno: Turno)

Pre-condizioni:- - -

Post-condizioni:

- [se esiste una RicorrenzaTurni ric tale che turno **ricorre in** ric]
l'istanza ric viene eliminata
- l'istanza turno viene eliminata

6a.1 eliminaRicorrenzaTurni(ricT: RicorrenzaTurni)

Pre-condizioni:- - -

Post-condizioni:

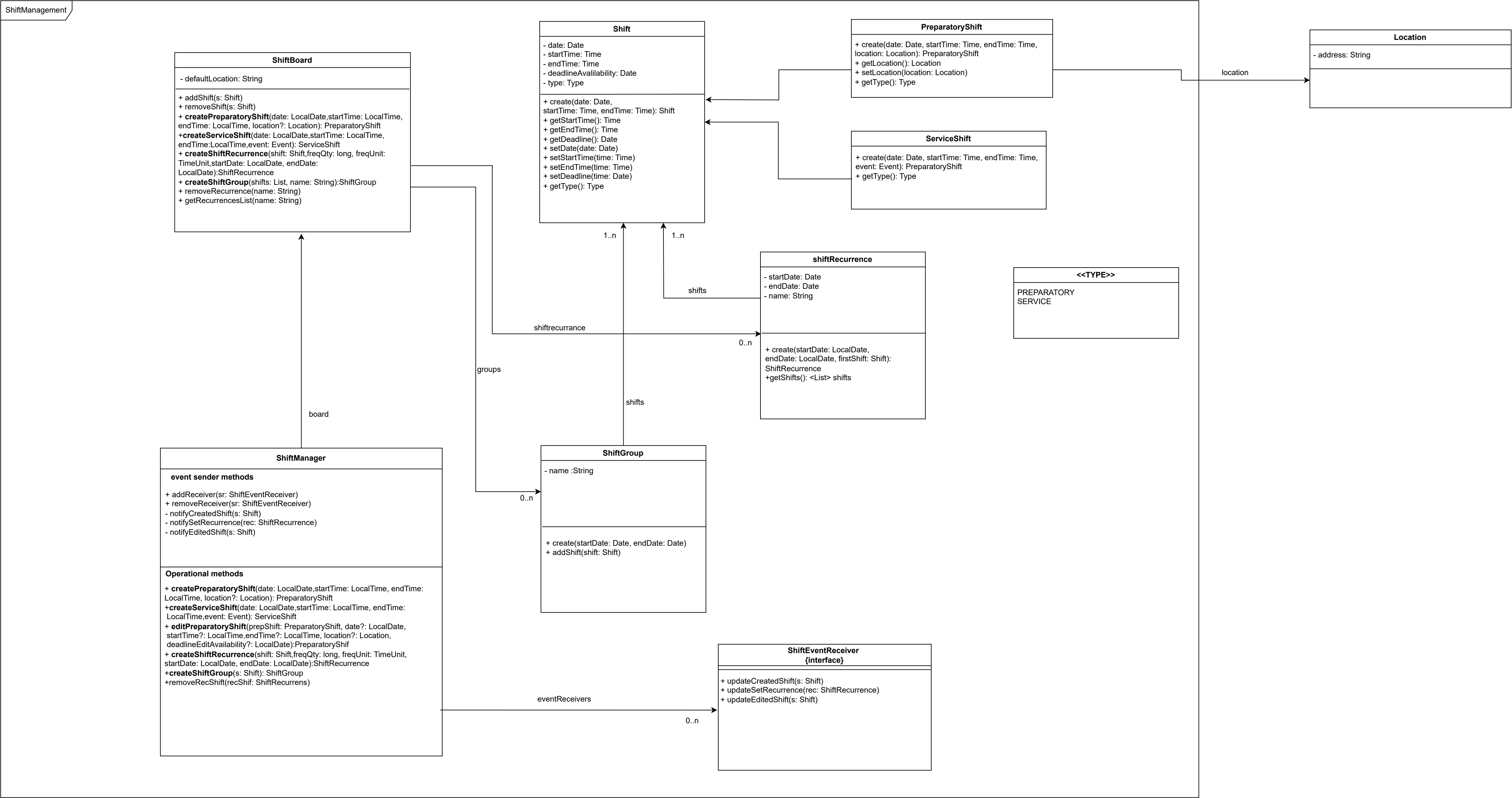
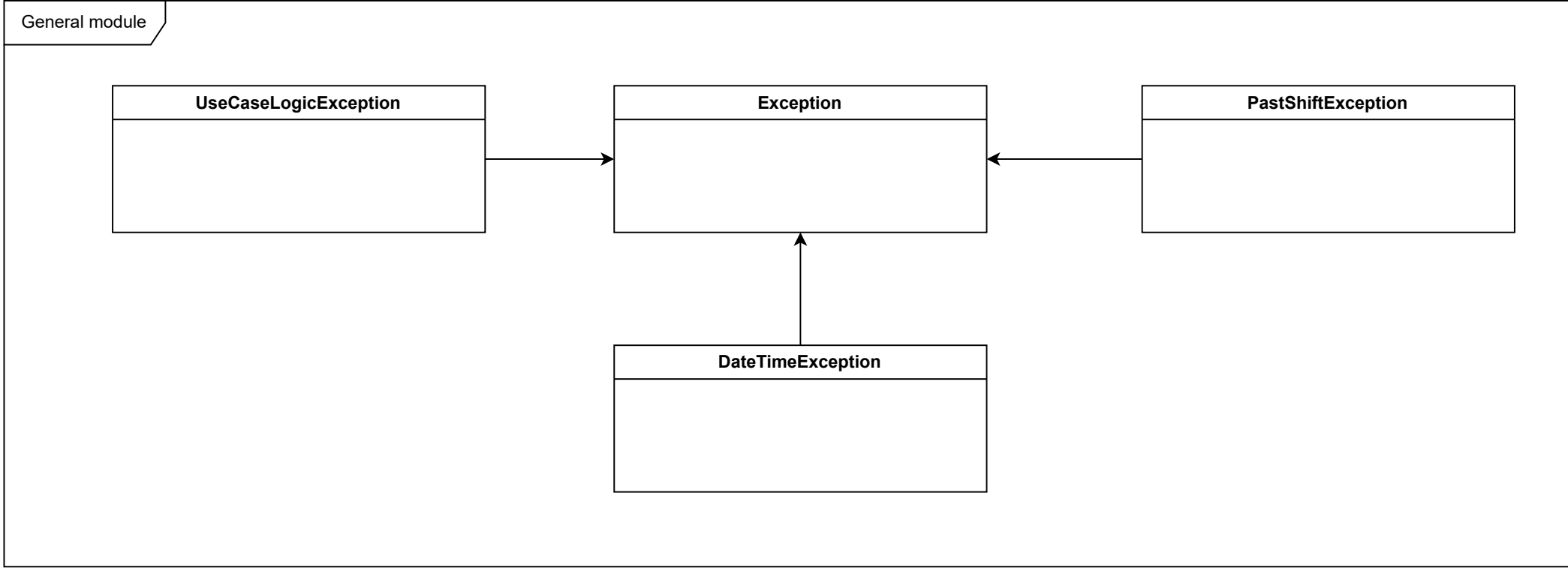
- L'istanza ricT viene eliminata

6b.1 eliminaGruppo(gruppo:GruppoTurni)

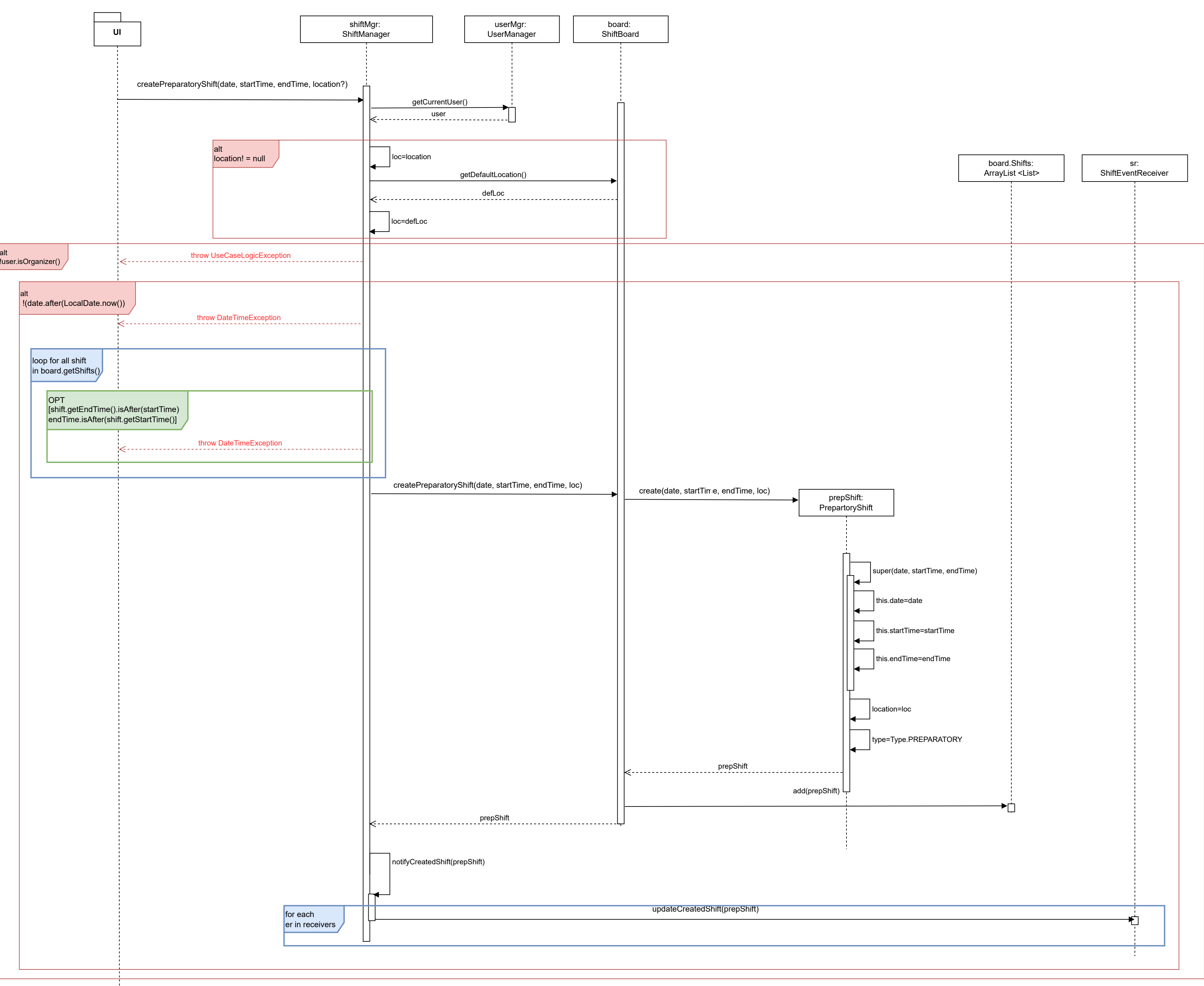
Pre-condizioni:- - -

Post-condizioni:

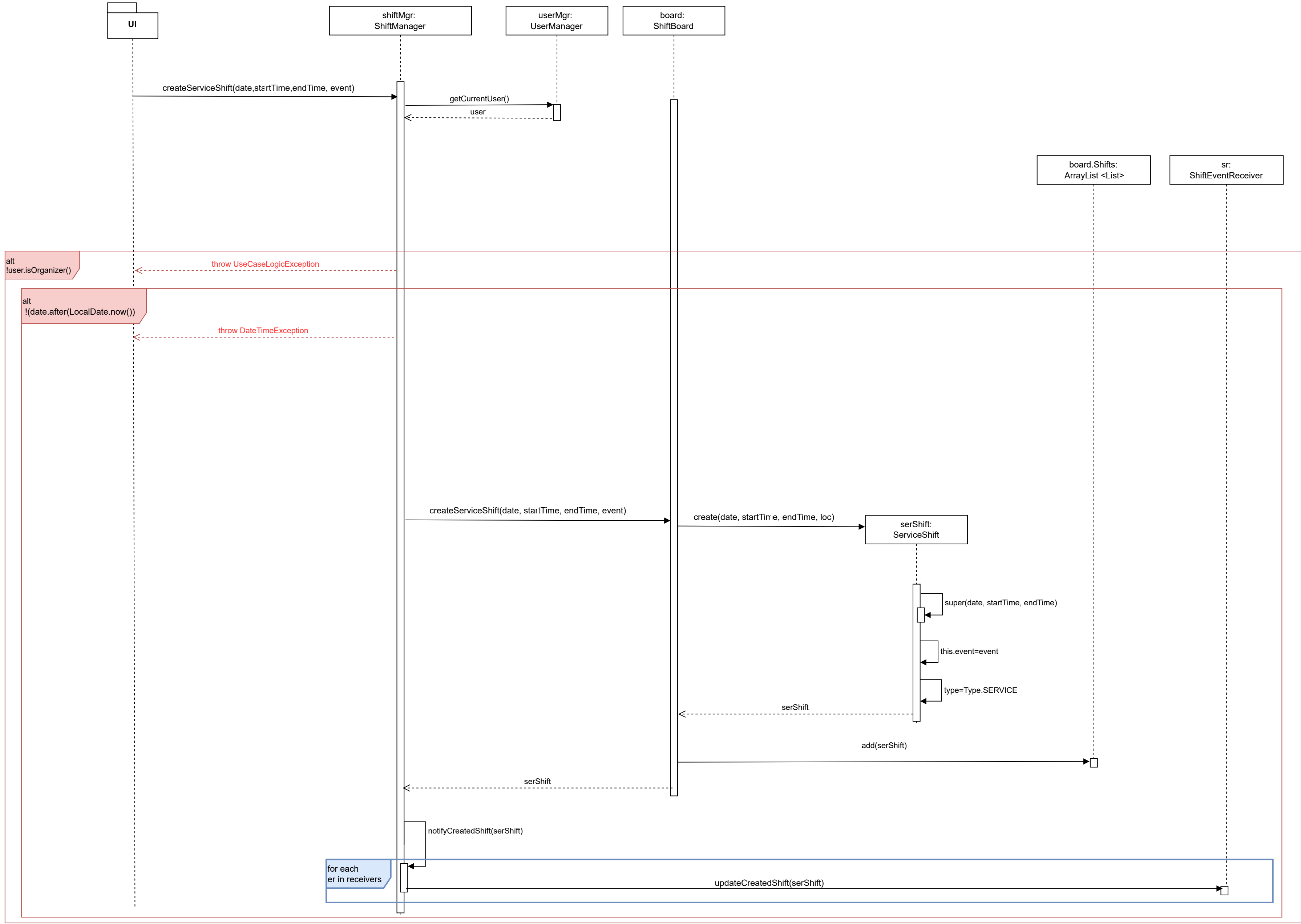
- l'istanza gruppo viene eliminata



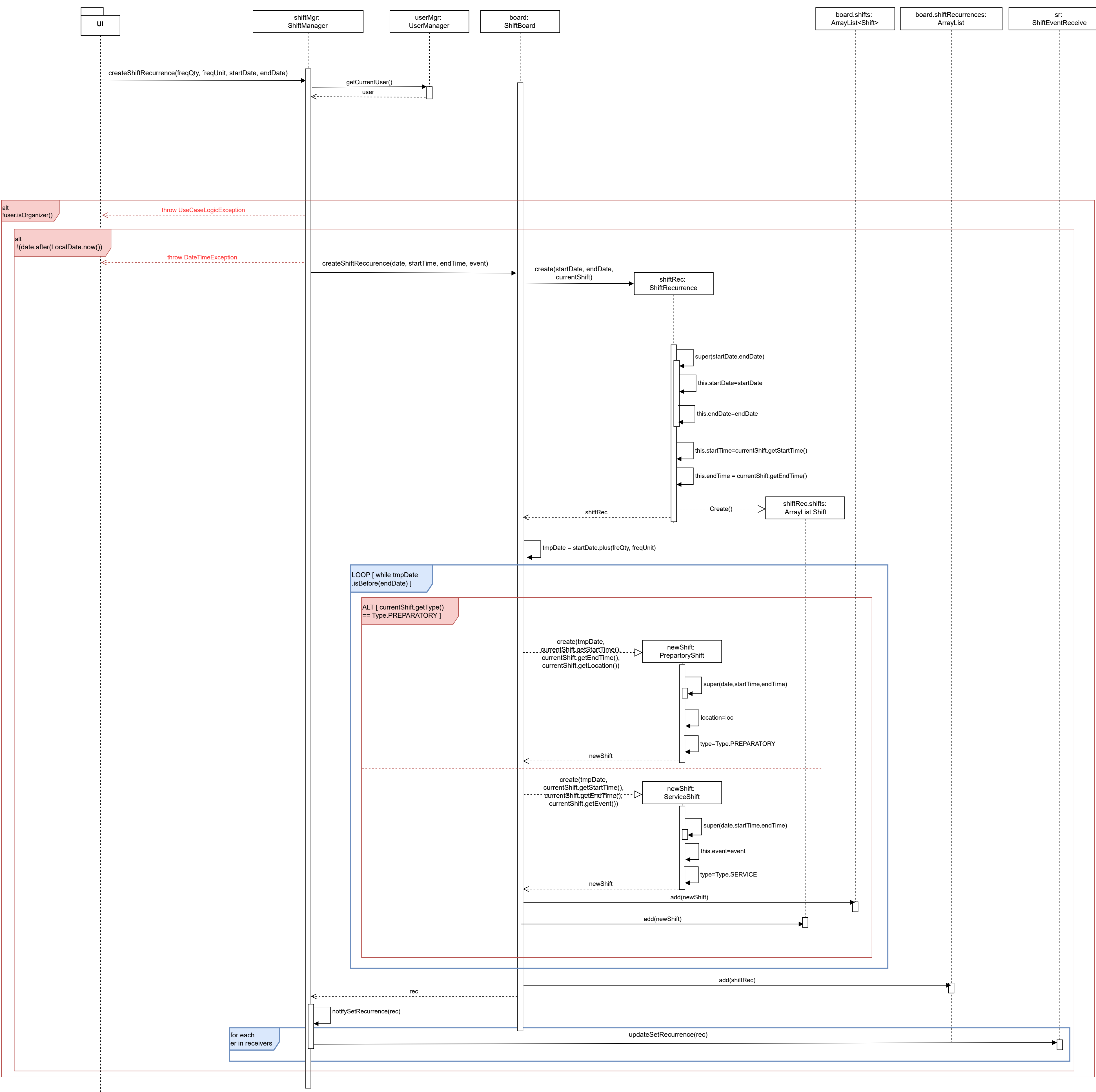
createPreparatoryShift



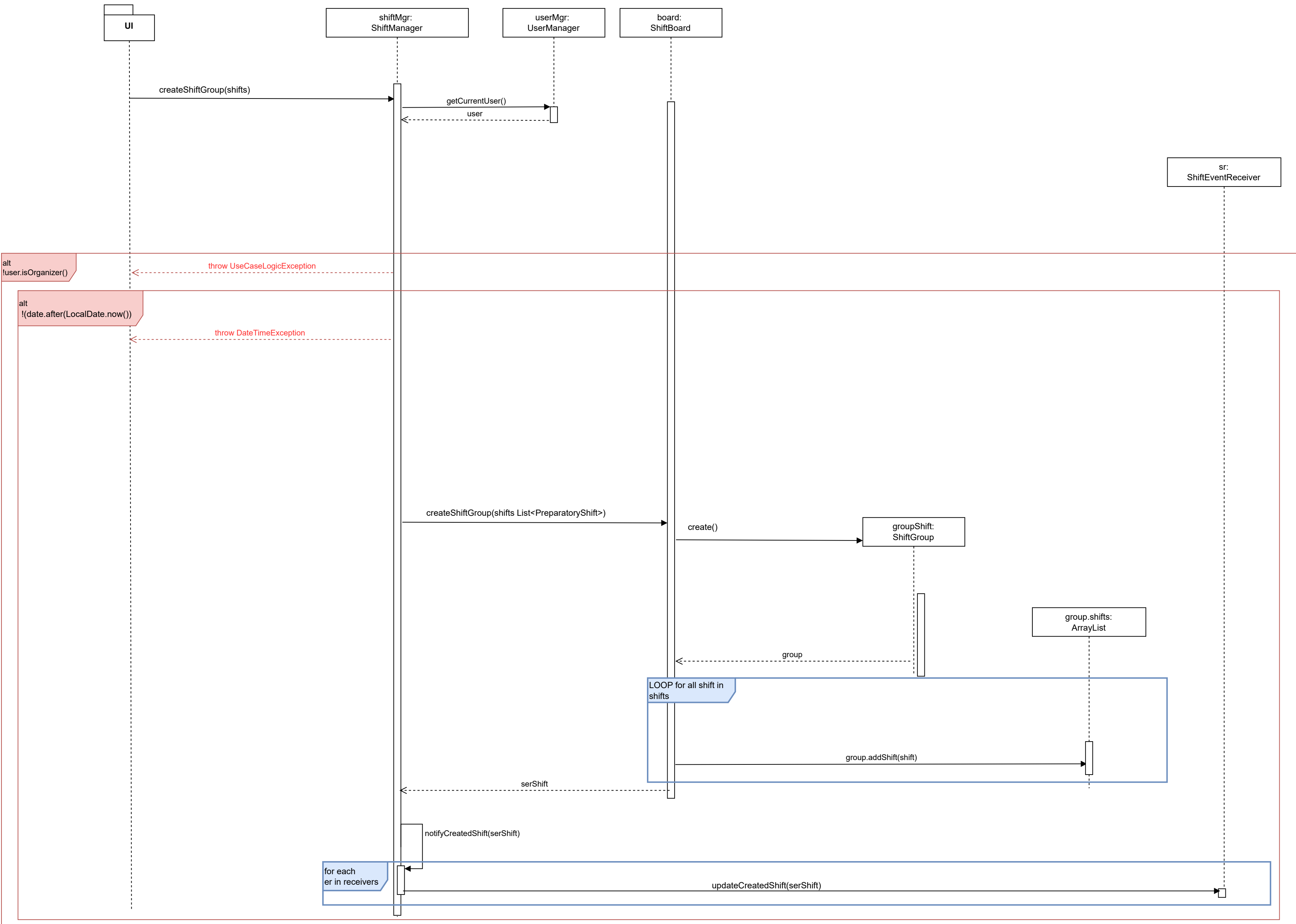
createServiceShift



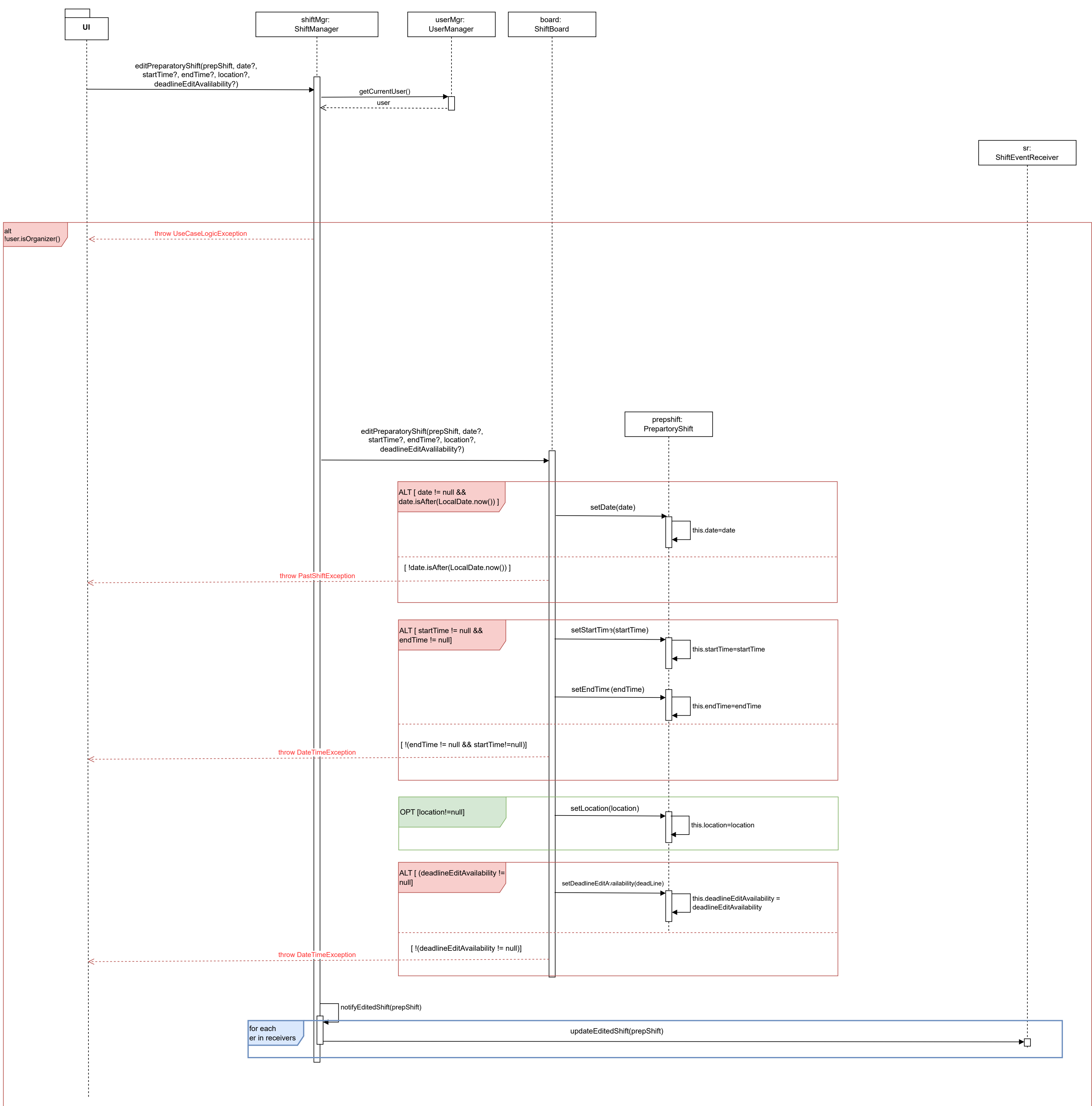
createShiftRecurrence



createShiftGroup



editPreparatoryShift



removeRecShift

