

Progetto
Sviluppo delle Applicazioni Software
2020/2021

Edoardo Grosso 861728

Yassine Amri 862580

Gestire i turni

Informazioni generali

Nome caso d'uso: Gestire i turni

Portata: Sistema

Livello: Obbiettivo utente

Attore primario: Chef

Parti Interessate: Organizzatore, Cuochi e Personale di Servizio.

Pre-condizioni: L'attore deve essere autenticato come Organizzatore

Garanzie di successo o post-condizioni: Creazione e organizzazione dei turni di cucina e i turni di servizio

Scenario principale di successo

#	Attore	Sistema
1	Opzionalmente crea un nuovo turno cucina specificando la data, l'ora di inizio e l'ora di fine, opzionalmente la scadenza di fine disponibilità ed opzionalmente il luogo di svolgimento.	Registra il turno cucina e lo inserisce nel tabellone.
	<i>Se desidera prosegue con il passo 2, altrimenti va al passo 4</i>	
2	Controlla turni	Fornisce i turni cucina.
3	Opzionalmente raggruppa dei turni di cucina	Registra il raggruppamento.
	<i>Ripete il passo 2 finché non soddisfatto</i>	
4	Opzionalmente aggiunge la cadenza e una data di fine ripetizione di un turno.	Registra la ripetizione
	<i>Termina il caso d'uso</i>	

Eccezione 1.1a

#	Attore	Sistema
1.1a.1	Opzionalmente crea un nuovo turno cucina specificando la data, l'ora di inizio e l'ora di fine, opzionalmente la scadenza di fine disponibilità ed opzionalmente il luogo di svolgimento	Sovrapposizione dei turni in cucina
	<i>Termina il caso d'uso</i>	

Estensione 1a

#	Attore	Sistema
1a.1	Opzionalmente scegli il turno per modifica specificando eventualmente la data, ora di inizio, ora di fine, scadenza disponibilità e luogo di svolgimento	Registra la modifica del turno

Eccezione 1a.1a

#	Attore	Sistema
1a.1a.1	Opzionalmente scegli il turno per modifica specificando eventualmente la data, ora di inizio, ora di fine, scadenza disponibilità e luogo di svolgimento	Almeno un cuoco ha già dato la propria disponibilità
	<i>Termina il caso d'uso</i>	

Eccezione 1a.1b

#	Attore	Sistema
1a.1b.1	Opzionalmente scegli il turno per modifica specificando eventualmente la data, ora di inizio, ora di fine, scadenza disponibilità e luogo di svolgimento	La modifica crea una sovrapposizione dei turni cucina
	<i>Termina il caso d'uso</i>	

Estensione 1a.2

#	Attore	Sistema
1a.2	Opzionalmente scegli turno ripetuto per modifica specificando eventualmente, ora di inizio, ora di fine, luogo di svolgimento, scadenza disponibilità e un range di data	Registra la modifica del turno

Eccezione 1a.2a

#	Attore	Sistema
1a.2a.1	Opzionalmente scegli turno ripetuto per modifica specificando eventualmente, ora di inizio, ora di fine, luogo di svolgimento, scadenza disponibilità e un range di data	Almeno un cuoco ha già dato la propria disponibilità ad un turno nel range di data
	<i>Termina il caso d'uso</i>	

Estensione 1b

#	Attore	Sistema
1b.1	Elimina un turno cucina	Elimina il turno cucina
	Termina il caso d'uso	

Eccezione 1b.1a

#	Attore	Sistema
1b.1a.1	Elimina un turno cucina	Almeno un cuoco ha già dato la propria disponibilità
	<i>Termina il caso d'uso</i>	

Eccezione 3.1a

#	Attore	Sistema
3.1a	Opzionalmente raggruppa dei turni di cucina	È già stato data almeno una disponibilità per almeno un turno
	<i>Termina il caso d'uso</i>	

Estensione 3a

#	Attore	Sistema
3a.1	Elimina il raggruppamento: i turni raggruppati diventano turni non raggruppati	Elimina il raggruppamento

Eccezione 3a.1a

#	Attore	Sistema
3a.1a.1	Elimina il raggruppamento: i turni raggruppati diventano turni non raggruppati	È già stata data almeno una disponibilità
	<i>Termina il caso d'uso</i>	

Estensione 3b

#	Attore	Sistema
3b.1	Scollega il turno dal raggruppamento	Registra lo scollegamento del turno dal raggruppamento
	<i>Ripetere finché non soddisfatto</i>	

Eccezione 3b.1a

#	Attore	Sistema
3b.1a.1	Scollega turno dal raggruppamento	È stata già data una disponibilità
	<i>Termina il caso d'uso</i>	

Estensione 3c

#	Attore	Sistema
3c.1	Sceglie un raggruppamento al quale aggiungere il turno.	Registra raggruppamento.

Eccezione 3c.1a

#	Attore	Sistema
3c.1a.1	Sceglie un raggruppamento al quale aggiungere il turno.	È stata già data una disponibilità
	<i>Termina il caso d'uso</i>	

Estensione 4a

#	Attore	Sistema
4a.1	Elimina la ripetizione del turno inserendo le date di inizio e di fine da eliminare.	Elimina la ripetizione.

Eccezione 4a.1a

#	Attore	Sistema
4a.1a.1	Elimina la ripetizione del turno inserendo le date di inizio e di fine da eliminare.	È stata già data una disponibilità ad almeno un turno
	<i>Termina il caso d'uso</i>	

Estensione 4b

#	Attore	Sistema
4b.1	aggiunge la cadenza e una data di fine ripetizione del raggruppamento.	Registra la ripetizione

Eccezione 4b.1a

#	Attore	Sistema
4b.1a.1	aggiunge la cadenza e una data di fine ripetizione del raggruppamento.	Almeno un turno del raggruppamento non è ripetuto
	<i>Termina il caso d'uso</i>	

Eccezione 4b.1b

#	Attore	Sistema
4b.1b.1	aggiunge la cadenza e una data di fine ripetizione del raggruppamento.	La data di fine ripetizione eccede almeno una data di fine ripetizione di un turno
	<i>Termina il caso d'uso</i>	

Estensione (1-4)a

#	Attore	Sistema
(1-4)a.1	Consulta Evento	Fornisce l'evento richiesto
(1-4)a.2	Crea un nuovo turno servizio, specificando il servizio, l'ora di inizio in anticipo e ora di fine aggiuntiva e eventualmente la scadenza di fine disponibilità.	Registra turno e lo inserisce nel tabellone

Estensione (1-4)a.2a

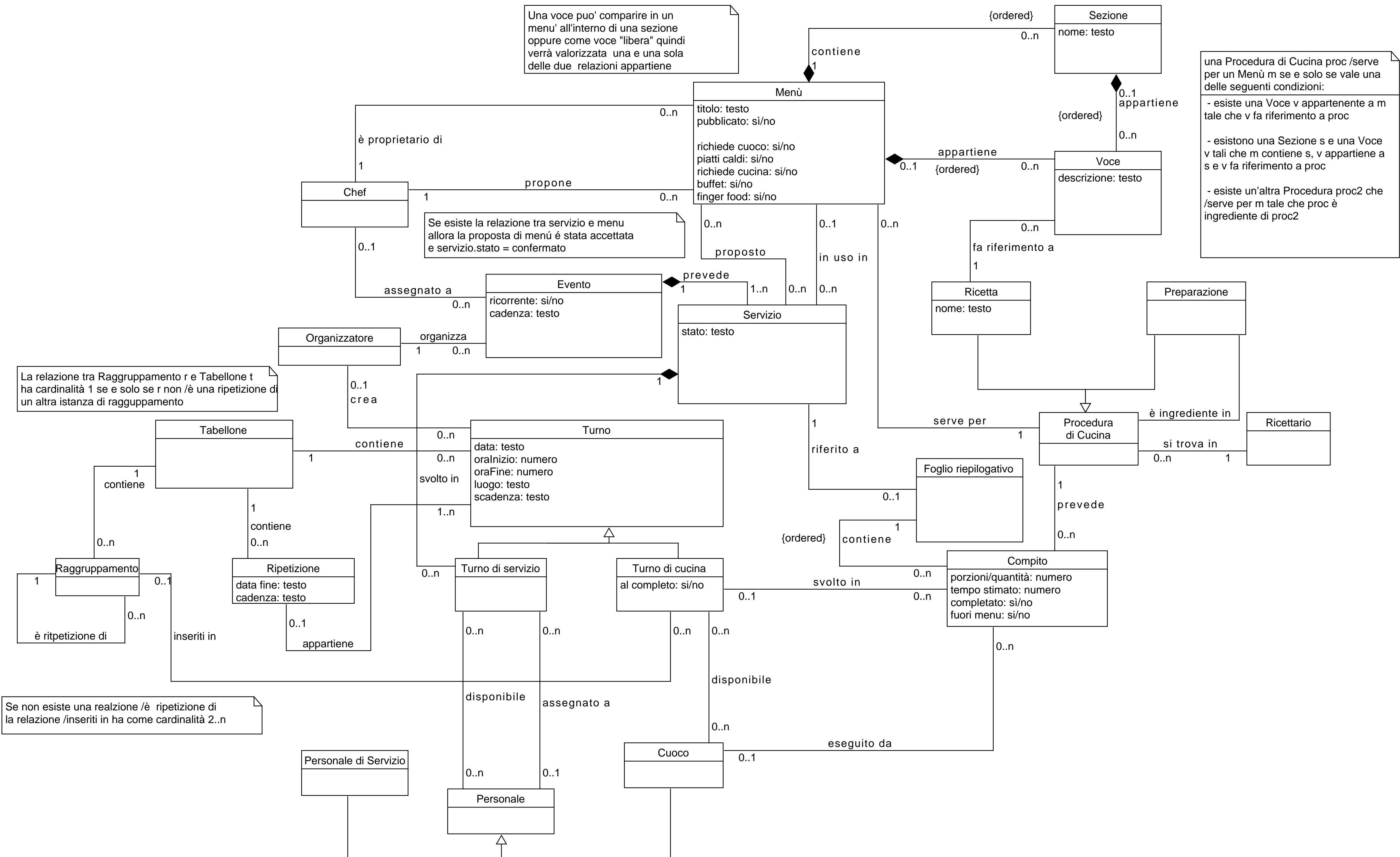
#	Attore	Sistema
(1-4)a.2a.1	Crea un nuovo turno servizio per ogni ricorrenza di un evento, specificando il servizio, l'ora di inizio in anticipo e ora di fine aggiuntiva e eventualmente la scadenza di fine disponibilità.	Registra le ripetizioni dei turni e li inserisce nel tabellone

Estensione (1-4)b

#	Attore	Sistema
(1-4).b.1	Opzionalmente modifica turni di servizio indicando opzionalmente minuti di anticipo e i minuti aggiuntivi	Modifica il turno di servizio

Eccezione (1-4)b.1a

#	Attore	Sistema
(1-4)b.1a.1	Opzionalmente modifica turno di servizio indicando opzionalmente minuti di anticipo e i minuti aggiuntivi	È già stata data almeno una disponibilità
	<i>Termina il caso d'uso</i>	



Gestire Turni - Scenario Principale



Sistema

opt

1. creaTurnoCucina(data, orainizio, oraFine, scadenza?, luogo?)

turno di cucina creato

opt

loop

2. controllaTurniCucina()

turni di cucina forniti

opt

3. creaRaggruppamento(turnoCucina1, turnoCucina2)

raggruppamento creato

opt

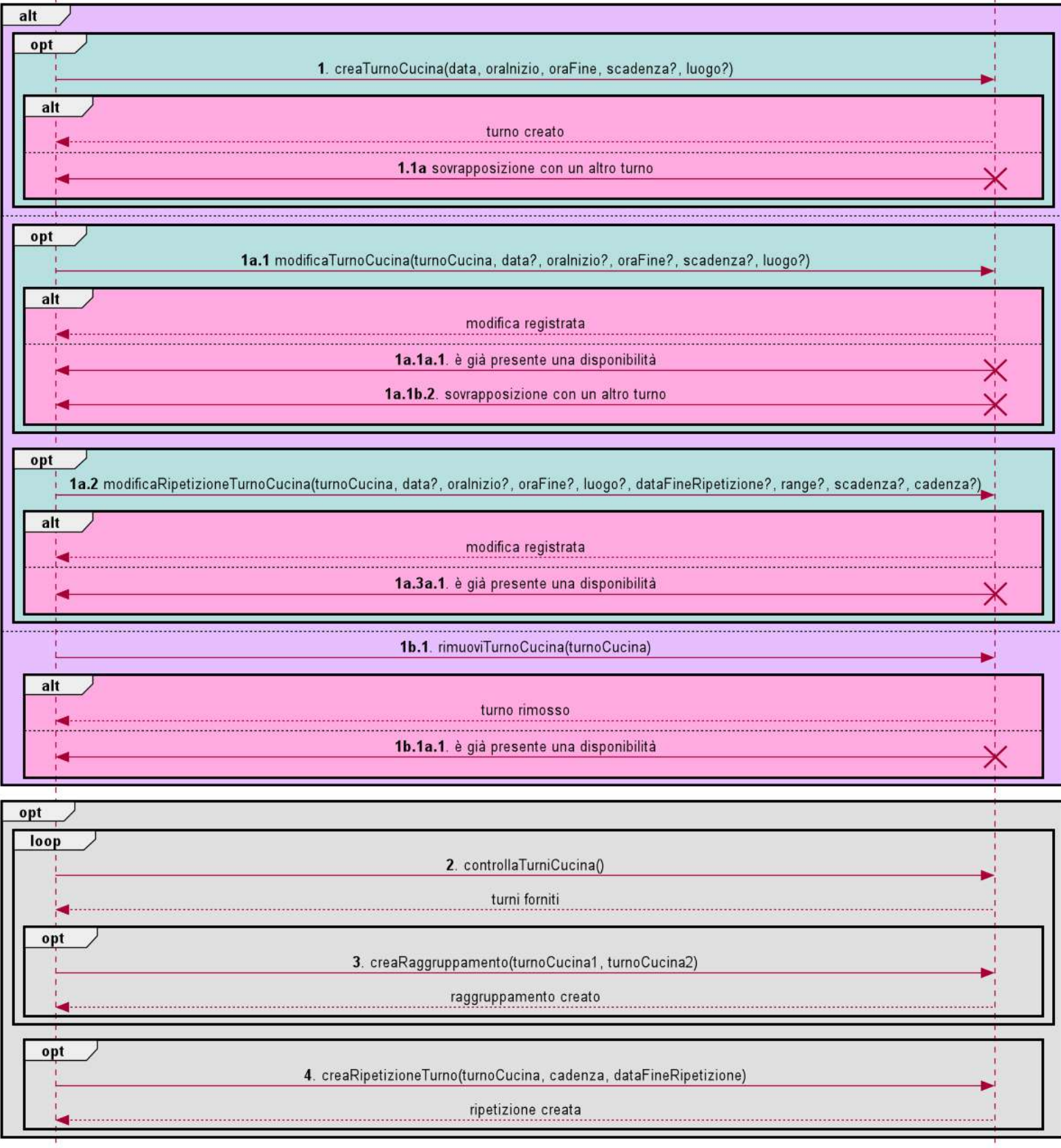
4. creaRipetizioneTurno(turnoCucina, cadenza, dataFineRipetizione)

ripetizione creata

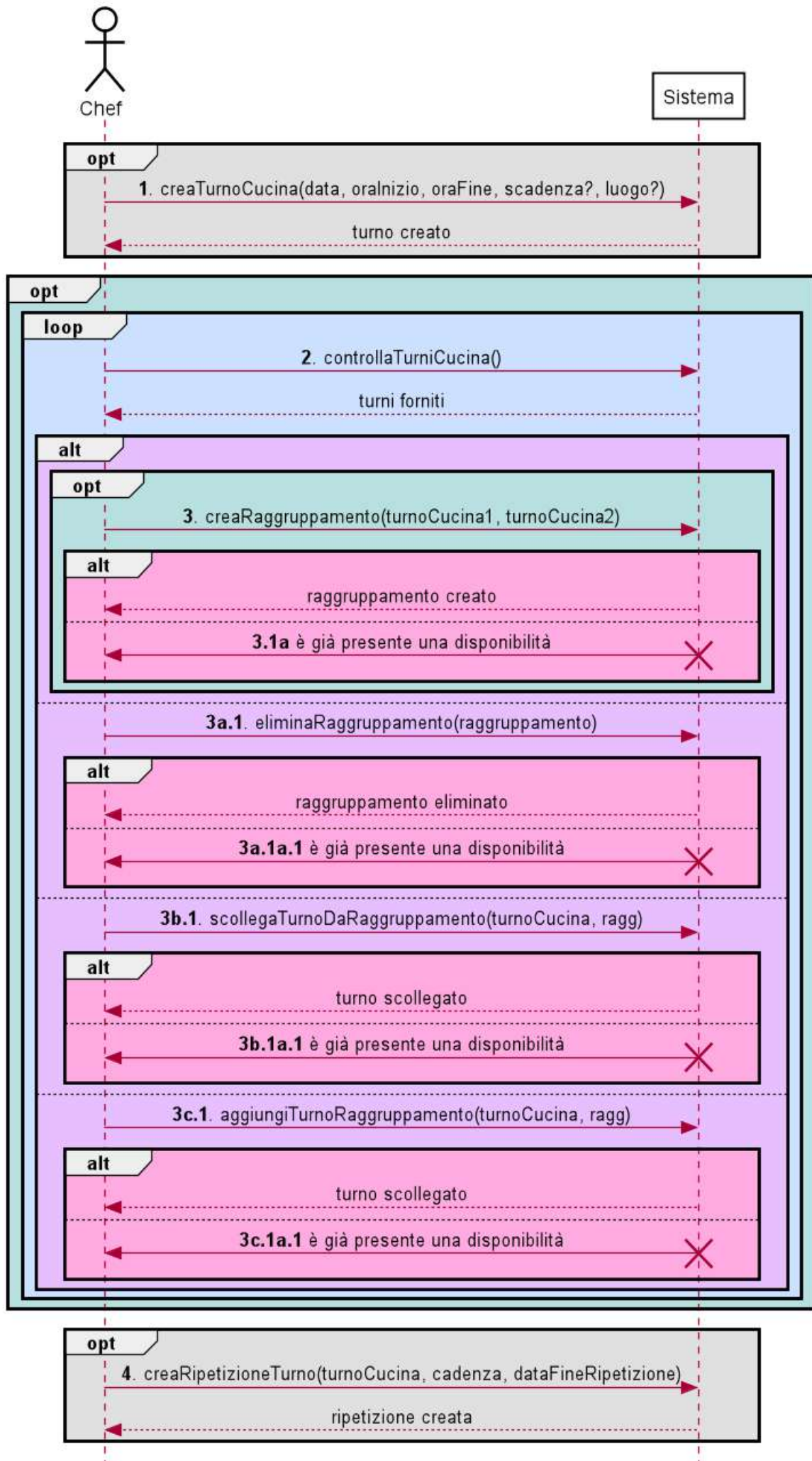
Gestire turni - Passo 1
Estensioni: (1a, 1b)
Eccezioni: (1.1a, 1a.1, 1b.1a)



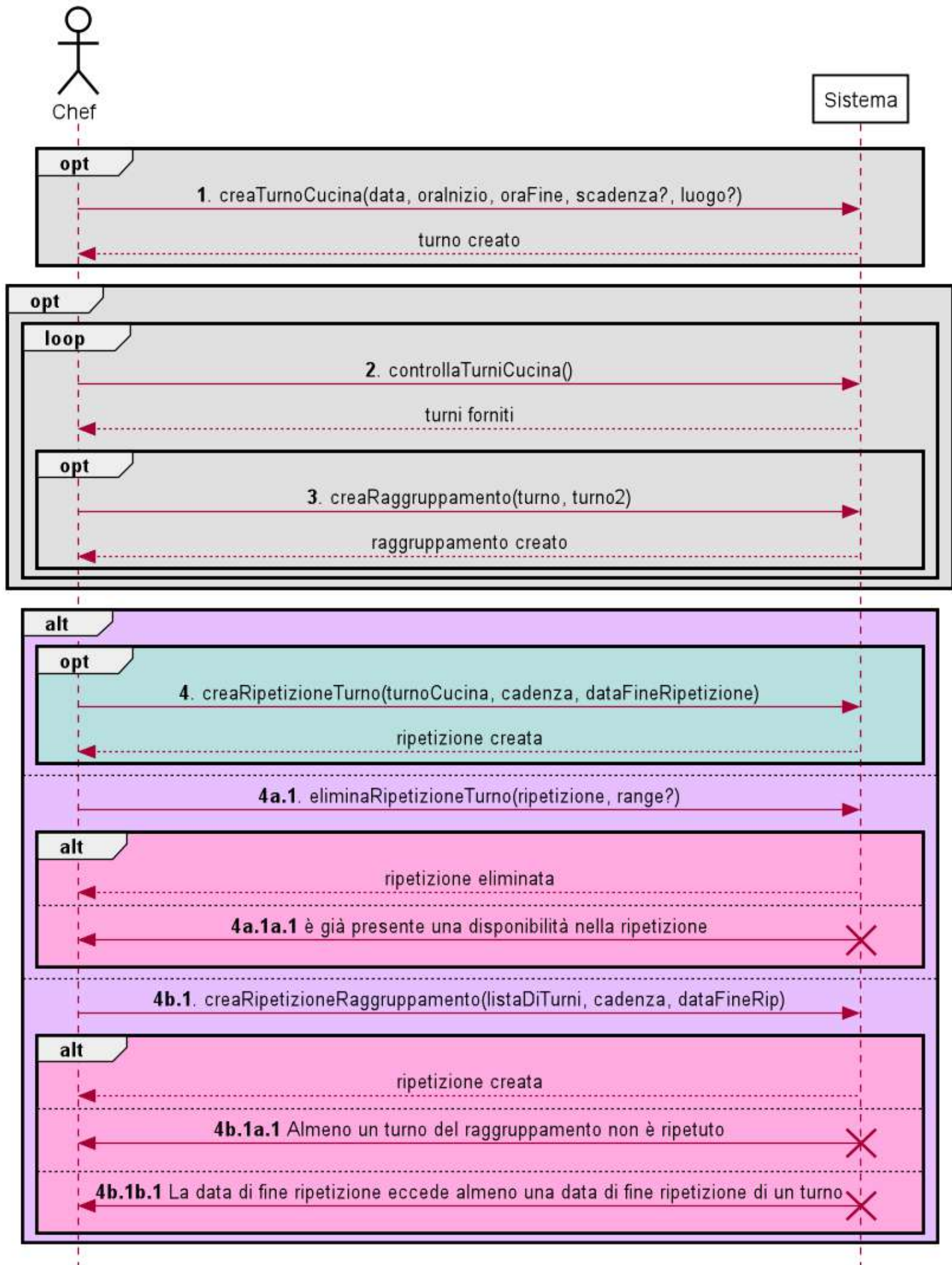
Sistema



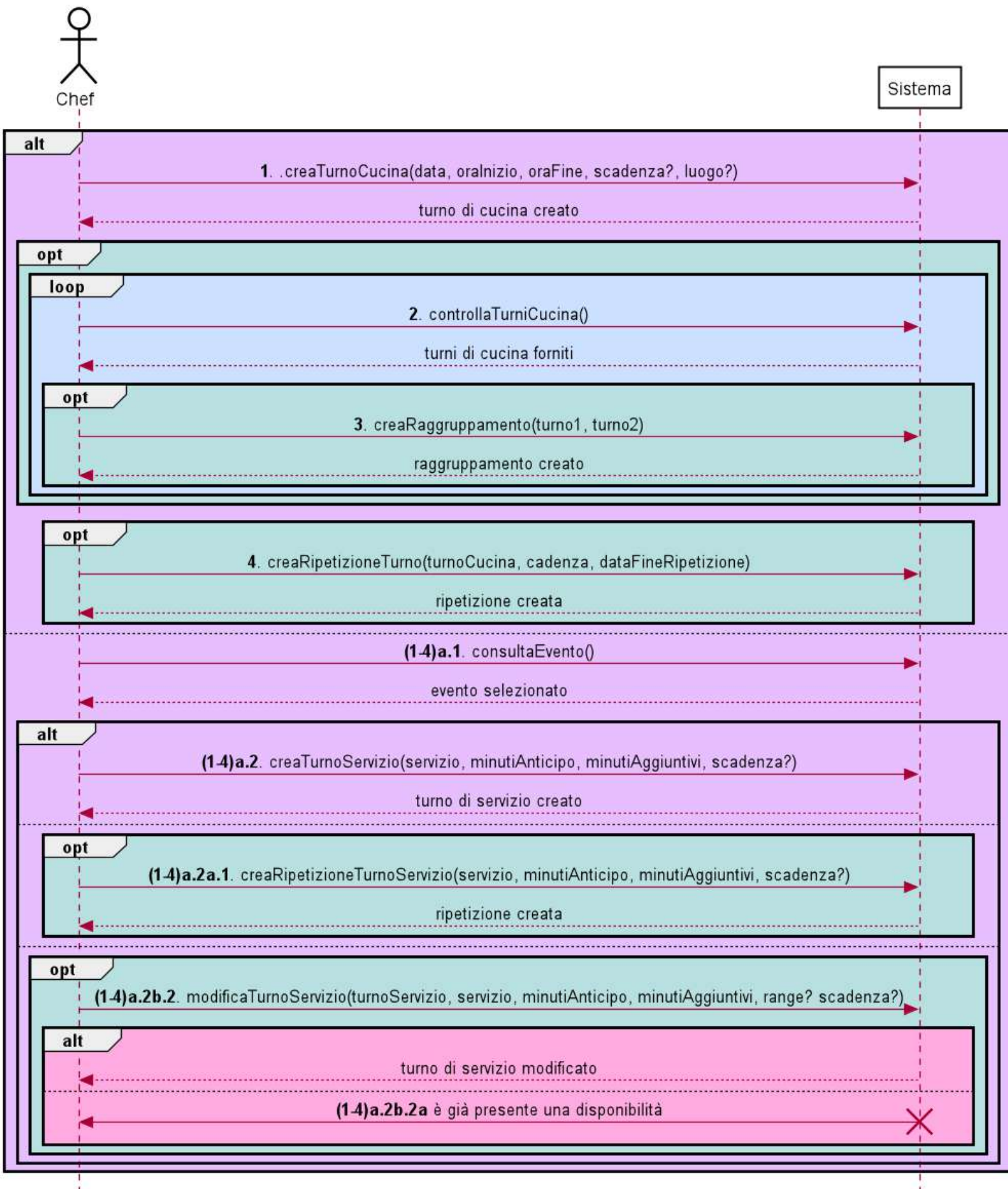
Gestire turni - Passo 2 e 3
Estensioni: (3a.1, 3b.1, 3c.1)
Eccezioni: (3a.1a.1, 3b.1a.1, 3c.1a.1)



Gestire turni - Passo 4
Estensioni: (4a, 4b)
Eccezioni: (4b.1a.1, 4b.1b.1)



Gestire Turni - Passo (1-4)
Estensioni: (1-4)a.1, (1-4)a.2, (1-4)a.2a.1, (1-4)b.1,
Eccezioni: (1-4)b.1a.1



Contratti per lo UC “Gestire Turni”

Pre-condizione generale: l'attore è identificato con un'istanza org di Organizzatore, si sta visualizzando l'istanza tab di Tabellone

1. creaTurnoCucina(data: testo, oraInizio: numero, oraFine: numero, scadenza?:testo, luogo?: testo)

Pre-condizioni: --

Post-condizioni:

- è stata creata un'istanza *trc* di Truno di Cucina
- *trc.data* = data
- *trc.oraInizio* = oraInizio
- *trc.oraFine* = oraFine
- [se scadenza è definito] *trc.scadenza* = scadenza
- [se luogo è definito] *trc.luogo* = luogo
- [se luogo non è definito] *trc.luogo* = cucina
- *tab* **contiene** *trc*

1a.1 modificaTurnoCucina(*trc*: Truno di Cucina, data?: testo, oraInizio?: numero, oraFine?: numero, scadenza?: testo, luogo?: testo)

Pre-condizioni:

- non esiste nessuna istanza *cuoco* che è **disponibile** in *trc*

Post-condizioni:

- [se data è definito] *trc.data* = data
- [se oraInizio è definito] *trc.oraInizio* = oraInizio
- [se oraFine è definito] *trc.oraFine* = oraFine
- [se data è definito] *trc.scadenza* = scadenza
- [se luogo è definito] *trc.luogo* = luogo

1a.2 modificaRipetizioneTurnoCucina(trc: Turno di Cucina,data?: testo, range?: testo, oraInizio?: numero, oraFine?: numero, luogo?: testo, cadenza?:testo, dataFineRip?: testo, scadenza?:testo)

Pre-condizioni:

- trc **appartiene** ad una Ripetizione rip
- [se range è definito] per ogni istanza trcN che **appartiene** a rip con trcN.data, trcN.oraInizio e trcN.oraFine nel range range
 - non esiste nessuna istanza cuoco che è **disponibile** in trcN
- [se range non è definito] per ogni istanza trcN che **appartiene** a rip
 - non esiste nessuna istanza cuoco che è **disponibile** in trcN

Post-condizioni:

- [se cadenza o dataFineRip sono definiti]
 - per ogni istanza trcN di Turno di Cucina in rip
 - trcN non **appartiene** più a rip
 - tab non **contiene** trcN
 - trcN è stato eliminato
 - rip.dataFine = dataFineRip;
 - rip.cadenza = cadenza
 - per ogni giorno della cadenza fino a dataFineRip
 - viene creata un'istanza trcNuovo di Turno di Cucina
 - trcNuovo.oraInizio = oraInizio
 - trcNuovo.oraFine = oraFine
 - trcNuovo.luogo = luogo
 - trcNuovo.scadenza = scadenza in relazione con cadenza
 - trcNuovo.data = data in relazione con cadenza
 - trcNuovo **appartiene** a rip
- per ogni trcN **appartenente** a rip con trcN.data, trcN.oraInizio e trcN.oraFine nel range range
 - [se data è definito] trc.data = data
 - [se oraInizio è definito] trc.oraInizio = oraInizio
 - [se oraFine è definito] trc.oraFine = oraFine
 - [se luogo è definito] trc.luogo = luogo
 - [se scadenza è definito] trc.scadenza = scadenza in relazione con cadenza

1b.1 rimuoviTurnoCucina(trc: Turno di Cucina)

Pre-condizioni:

- non esiste nessuna istanza cuoco che è **disponibile** in trc

Post-condizioni:

- tab non **contiene** trc
- l'istanza trc è stata eliminata

2. controllaTurniCucina()

Pre-condizioni: -

Post-condizioni: -

3. creaRaggruppamento(trc: Turno di Cucina, trc2: Turno di Cucina)

Pre-condizioni:

- trc e trc2 non sono **inseriti in** nessuna istanza di Raggruppamento
- non esiste nessuna istanza *cuoco* che è **disponibile** in trc e/o in trc2

Post-condizioni:

- è stata creata un'istanza *ragg* di Raggruppamento
- trc e trc2 sono **inseriti in** *ragg*
- *tab* **contiene** *ragg*

3a.1 eliminaRaggruppamento(ragg: Raggruppamento)

Pre-condizioni:

- non esiste una istanza *ragg2* di cui ragg è **ripetizione**
- per ogni istanza *trc* di Turno di Cucina che **inserito in** ragg
 - non esiste nessuna istanza *cuoco* che è **disponibile** in *trc*

Post-condizioni:

- per ogni istanza *ragg2* che è ripetizione di ragg
 - per ogni istanza *trc2* di turno di cucina di *ragg2*
 - è stata eliminata la relazione **inserito in** tra *trc2* e *ragg2*
 - l'istanza *ragg2* è stata eliminata
- per ogni istanza *trc* di Turno di Cucina di ragg
 - è stata eliminata la relazione **inserito in** *trc* e ragg
- *tab* **non contiene** ragg
- l'istanza ragg è stata eliminata

3b.1 scollegaTurnoDaRaggruppamento(trc: Turno di Cucina, ragg: Raggruppamento)

Pre-condizioni:

- non esiste una istanza *ragg2* di cui ragg è **ripetizione**
- non esiste una istanza *ragg2* che è **ripetizione** di ragg
- esiste già l'associazione **inserito in** tra trc e ragg
- per ogni istanza *trcN* di Turno di Cucina che **appartiene** a ragg
 - non esiste nessuna istanza *cuoco* che è **disponibile** in *trcN*

- non esiste nessuna istanza *cuoco* che è **disponibile** in *trc*

Post-condizioni:

- è stata eliminata l'associazione tra *trc* e *ragg*

3c.1. aggiungiTurnoRaggruppamento(trc: Turno di Cucina, ragg: Raggruppamento)

Pre-condizioni:

- non esiste una istanza *ragg2* di cui *ragg* è ripetizione
- non esiste una istanza *ragg2* che è ripetizione di *ragg*
- esiste già l'associazione **inserito in** tra *trc* e *ragg*
- per ogni istanza *trcN* di Turno di Cucina che **appartiene** a *ragg*
 - non esiste nessuna istanza *cuoco* che è **disponibile** in *trcN*
- non esiste nessuna istanza *cuoco* che è **disponibile** in *trc*

Post-condizioni:

- *trc* è **inserito** in *ragg*

4. creaRipetizioneTurno(trc: Turno di Cucina, cadenza: testo, dataFineRip: testo)

Pre-condizioni: -

Post-condizioni:

- è stata creata un'istanza *rip* di Ripetizione
- *rip.dataFine* = *dataFineRip*;
- *rip.cadenza* = *cadenza*
- per ogni giorno della *cadenza* fino a *dataFineRip*
 - viene creata un'istanza *trcNuovo* di Turno di Cucina
 - *trcNuovo.oraInizio* = *trc.oraInizio*
 - *trcNuovo.oraFine* = *trc.oraFine*
 - *trcNuovo.luogo* = *trc.luogo*
 - *trcNuovo.data* = *trc.data* in relazione con *cadenza*
 - *trcNuovo.scadenza* = *trc.scadenza* in relazione con *cadenza* 1
 - *tab* **contiene** *trcNuovo*
 - *trcNuovo* **appartiene** a *rip*
 - *trc* **appartiene a** *rip*
- *tab* **contiene** *rip*

1. *cadenza* è riferita su base giornaliera, settimanale, mensile o annuale

4a.1 eliminaRipetizioneTurno(rip:Ripetizione, range?: testo)

Pre-condizioni:

- per ogni istanza *trc* di Turno di Cucina che **appartiene** a *rip*
 - non esiste nessuna istanza *cuoco* che è **disponibile** in *trc*

Post-condizioni:

- per ogni istanza trc di Turno di Cucina in *rip* con *trc.data*, *trc.oraInizio* e *trc.oraFine* nel range range
 - trc non appartiene più a *rip*
 - trc è stato eliminato

4b.1 creaRipetizioneRaggruppamento(listaDiTurni, cadenza: testo, dataFineRip: testo)

Pre-condizioni:

- per ogni istanza *trc* di Turno in listaDiTurni
 - *trc* è un Turno di Cucina
 - *trc* **appartiene** ad un'istanza *rip* di Ripetizione
 - *rip.dataFine* > dataFineRip
 - *rip.cadenza* = cadenza

Post-condizioni:

- è stata creata un'istanza *ragg* di Raggruppamento
- per ogni giorno di cadenza fino a dataFineRip:
 - viene creata un'istanza *raggNuovo* di Raggruppamento
 - per ogni *trc* in lista di turni
 - *trc* è **inserito in** *raggNuovo*
 - *raggNuovo* è **ripetizione di** *ragg*
- *tab* **contiene** *ragg*

(1-4)a.1 controllaEvento()

Pre-condizioni: -

Post-condizioni:-

(1-4)a.2 creaTurnoServizio(ser: Servizio, minutiAnticipo: numero, minutiAggiuntivi: numero, scadenza?: testo)

Pre-condizioni:

- Esiste una istanza *ev* **prevede** ser

Post-condizioni:

- è stata creata un'istanza trs di Truno di Servizio
- *trs.data* = ser.data
- *trs.oraInizio* = ser.oraInizio - (minutiAnticipo)
- *trs.oraFine* = ser.oraFine + (minutiAggiuntivi)
- *trs.luogo* = ser.luogo

- [se scadenza è definito] $trs.scadenza = \underline{scadenza}$
- tab **contiene** trs

(1-4)a.2a.1 creaRipetizioneTurnoServizio(ser: Servizio, minutiAnticipo: numero, minutiAggiuntivi: numero, scadenza?: testo)

Pre-condizioni:

- Esiste una istanza *ev* **prevede** ser
- *ev.ricorrente* = *si*

Post-condizioni:

- è stata creata una istanza *rip* di Ripetizione
- per ogni *ev.cadenza* in relazione con ser.data
 - è stata creata un'istanza trs di Turno di Servizio
 - $trs.data = \underline{ser.data}$
 - $trs.oraInizio = \underline{ser.oraInizio} - (\text{minutiAnticipo})$
 - $trs.oraFine = \underline{ser.oraFine} + (\text{minutiAggiuntivi})$
 - $trs.luogo = \underline{ser.luogo}$
 - [se scadenza è definito] $trs.scadenza = \underline{scadenza}$ in relazione con *cadenza*
 - *trs* **appartiene a** *rip*
 - tab **contiene** *trs*
- tab **contiene** *rip*

1. *cadenza* è riferita su base giornaliera, settimanale, mensile o annuale

(1-4)a.2b.2 modificaTurnoServizio(trs: Turno di Servizio minutiAnticipo: numero, minutiAggiuntivi: numero range?: testo, scadenza?: testo)

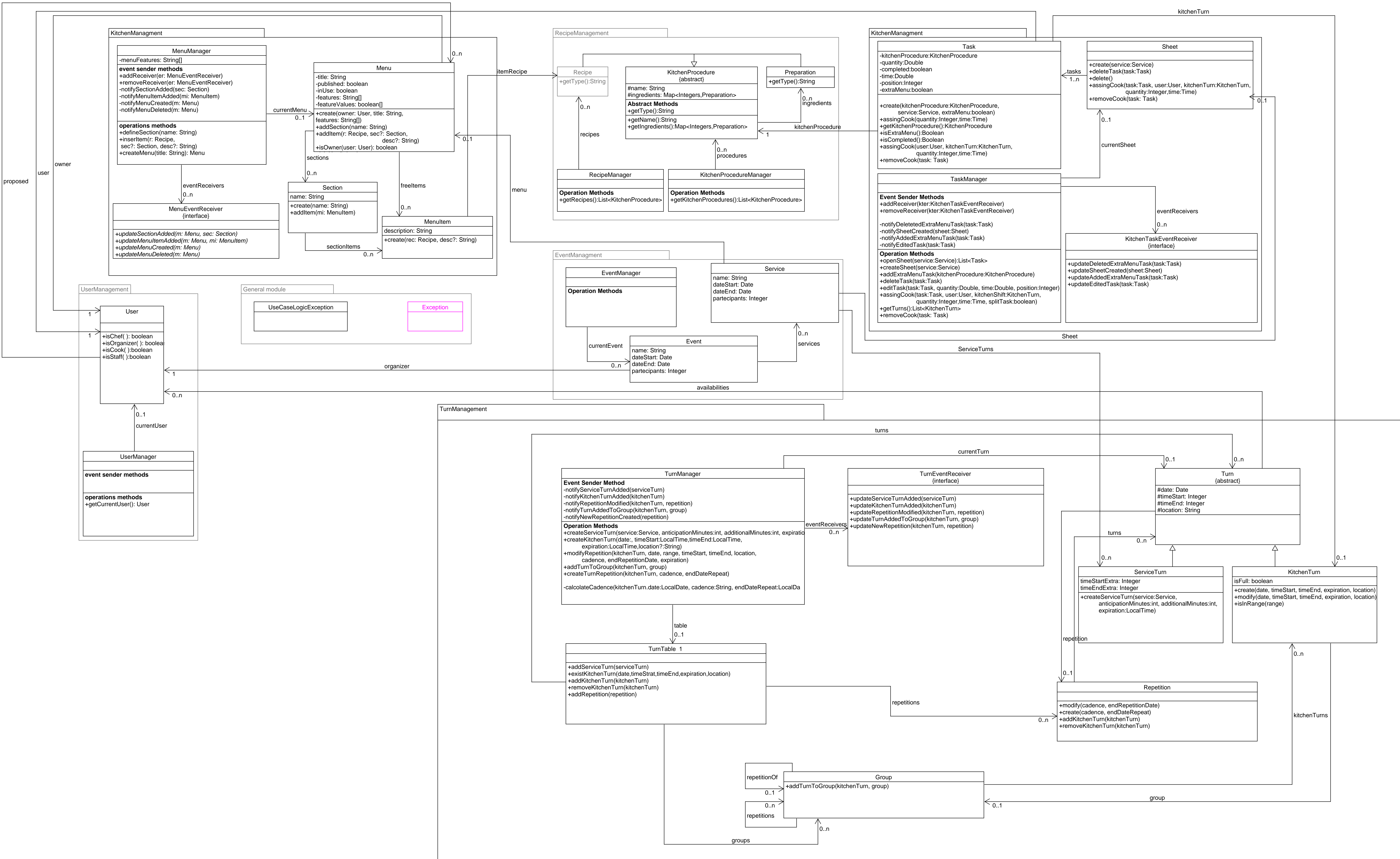
Pre-condizioni:

- trs è **svolto in** una istanza *ser* di Servizio
- per ogni istanza *trs2* di Turno di Servizio che **appartiene a** una istanza *rip* di Ripetizione
 - non esiste nessuna istanza *cuoco* che è **disponibile** in *trs2*
- non esiste nessuna istanza *cuoco* che è **disponibile** in *trs*

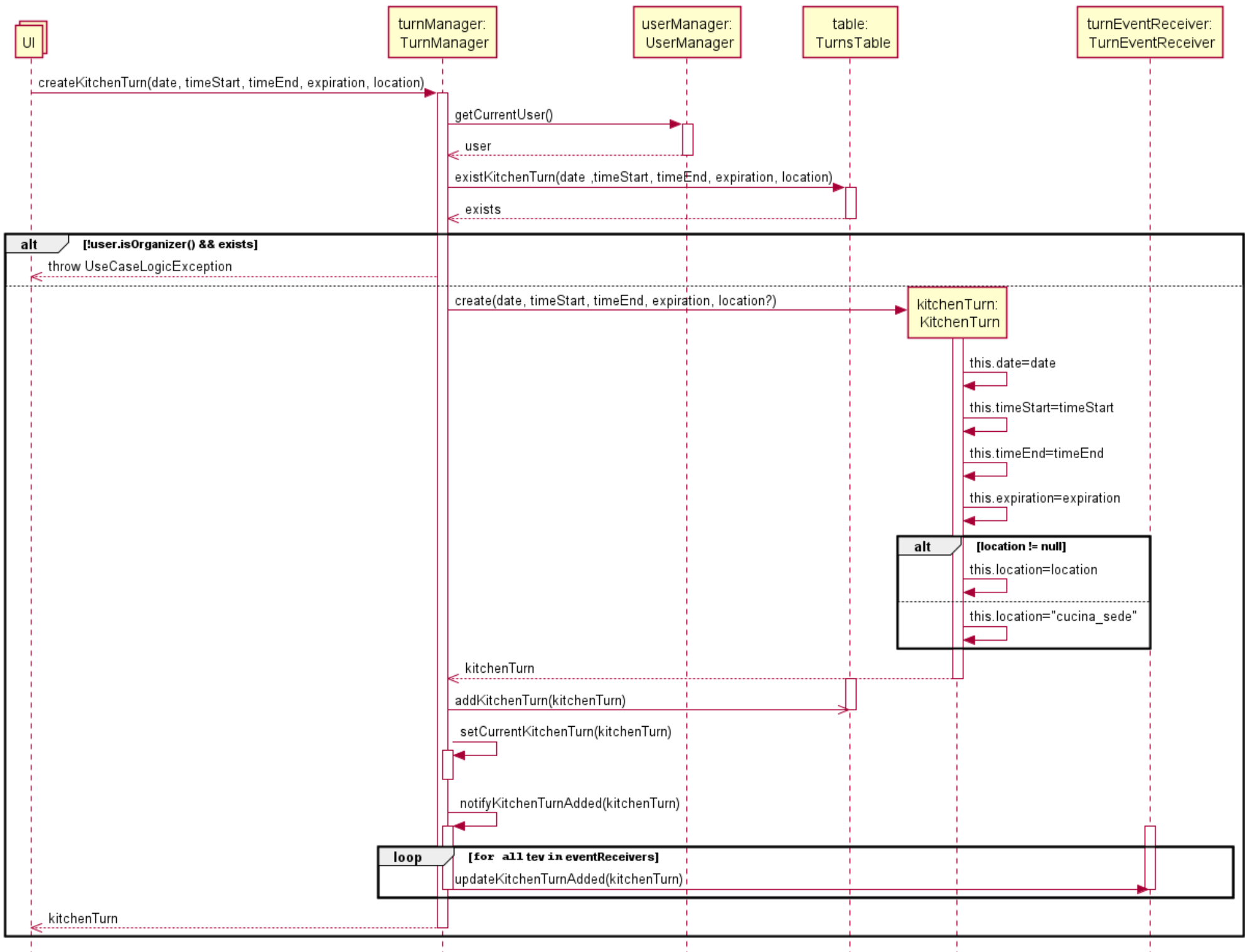
Post-condizioni:

- [se range è definito] per ogni istanza *trs2* di Turno di Servizio che **appartiene a** una istanza *rip* di Ripetizione con *trs2.data*, *trs2.oraInizio* e *trs2.oraFine* nel range range
 - [se minutiAnticipo è definito] $trs2.oraInizio = \underline{ser.oraInizio} - (\text{minutiAnticipo})$
 - [se minutiAggiuntivi è definito] $trs2.oraFineAggiuntiva = \underline{ser.oraFine} + (\text{minutiAggiuntivi})$
 - [se scadenza è definito] $trs.scadenza = \underline{scadenza}$ in relazione con *cadenza*

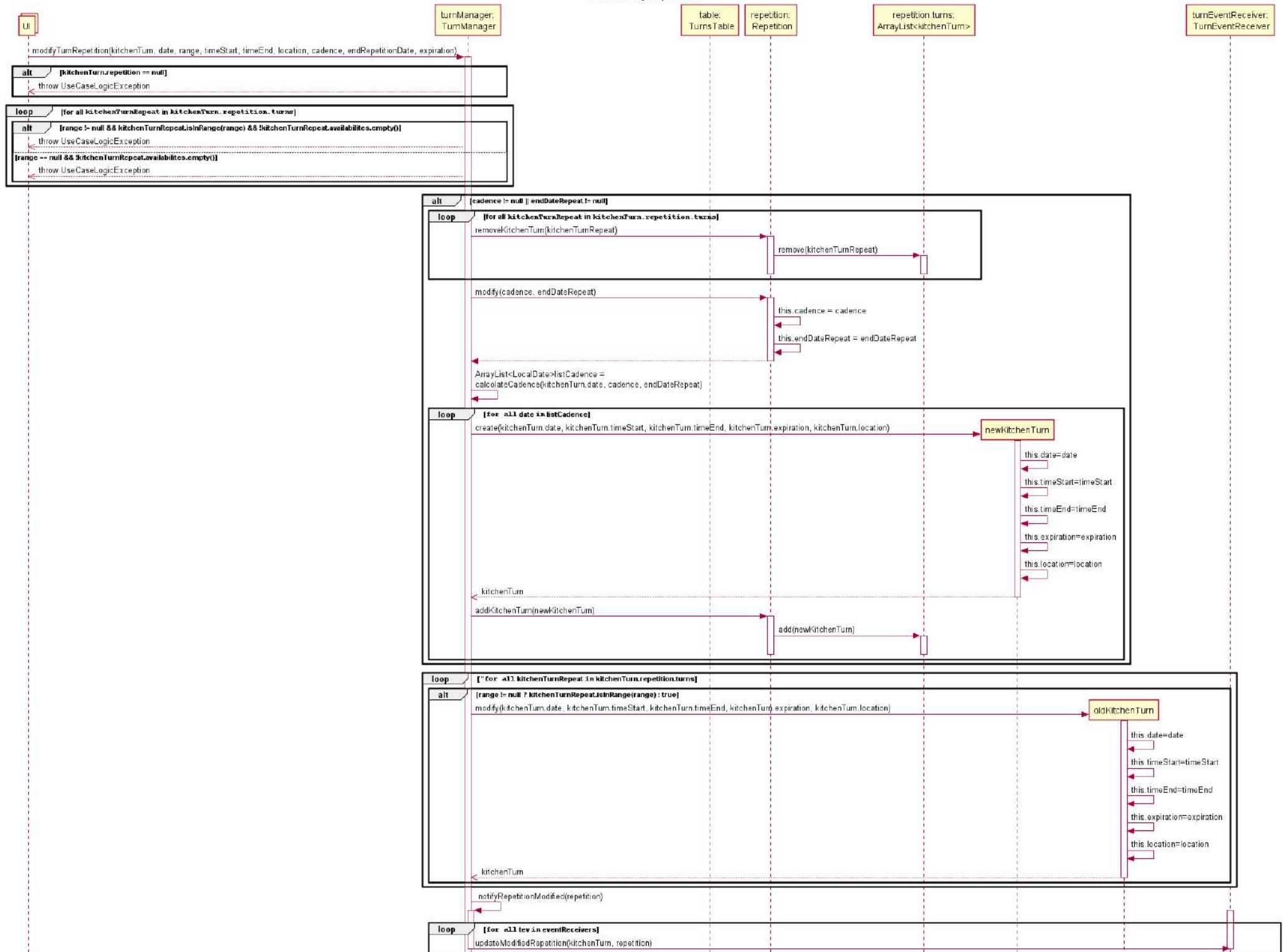
- [se range non è definito] per ogni istanza *trs2* di Turno di Servizio che **appartiene a** una istanza *rip di Ripetizione*
 - [se minutiAnticipo è definito] $trs2.oraInizio = ser.oraInizio - (minutiAnticipo)$
 - [se minutiAggiuntivi è definito] $trs2.oraFineAaggiuntiva = ser.oraFine + (minutiAggiuntivi)$
- [se minutiAnticipo è definito] $trs.oraInizio = ser.oraInizio - (minutiAnticipo)$
- [se minutiAggiuntivi è definito] $trs.oraFineAaggiuntiva = ser.oraFine + (minutiAggiuntivi)$
- [se scadenza è definito] $trs.scadenza = scadenza$



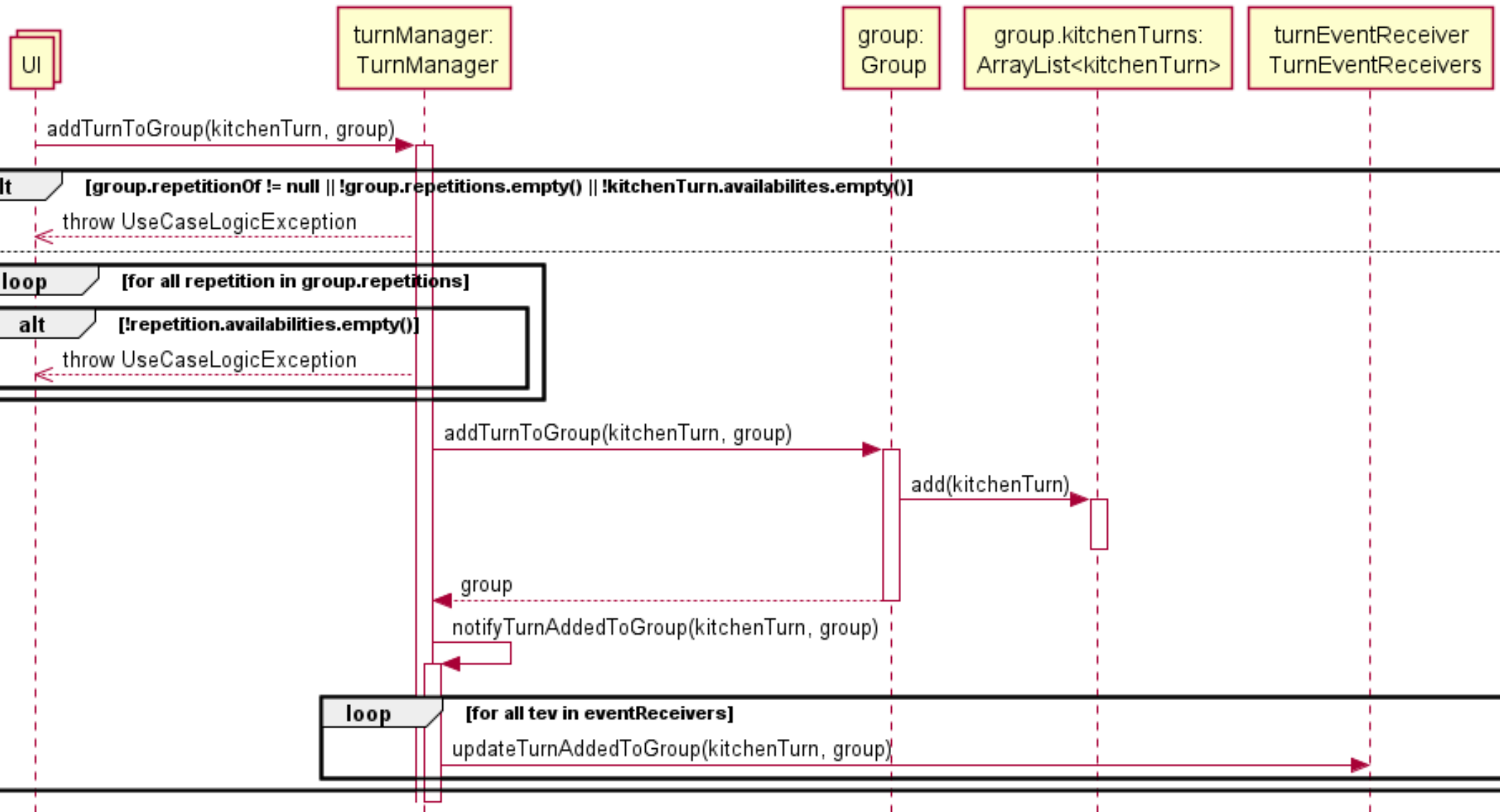
1 createKitchenTurn



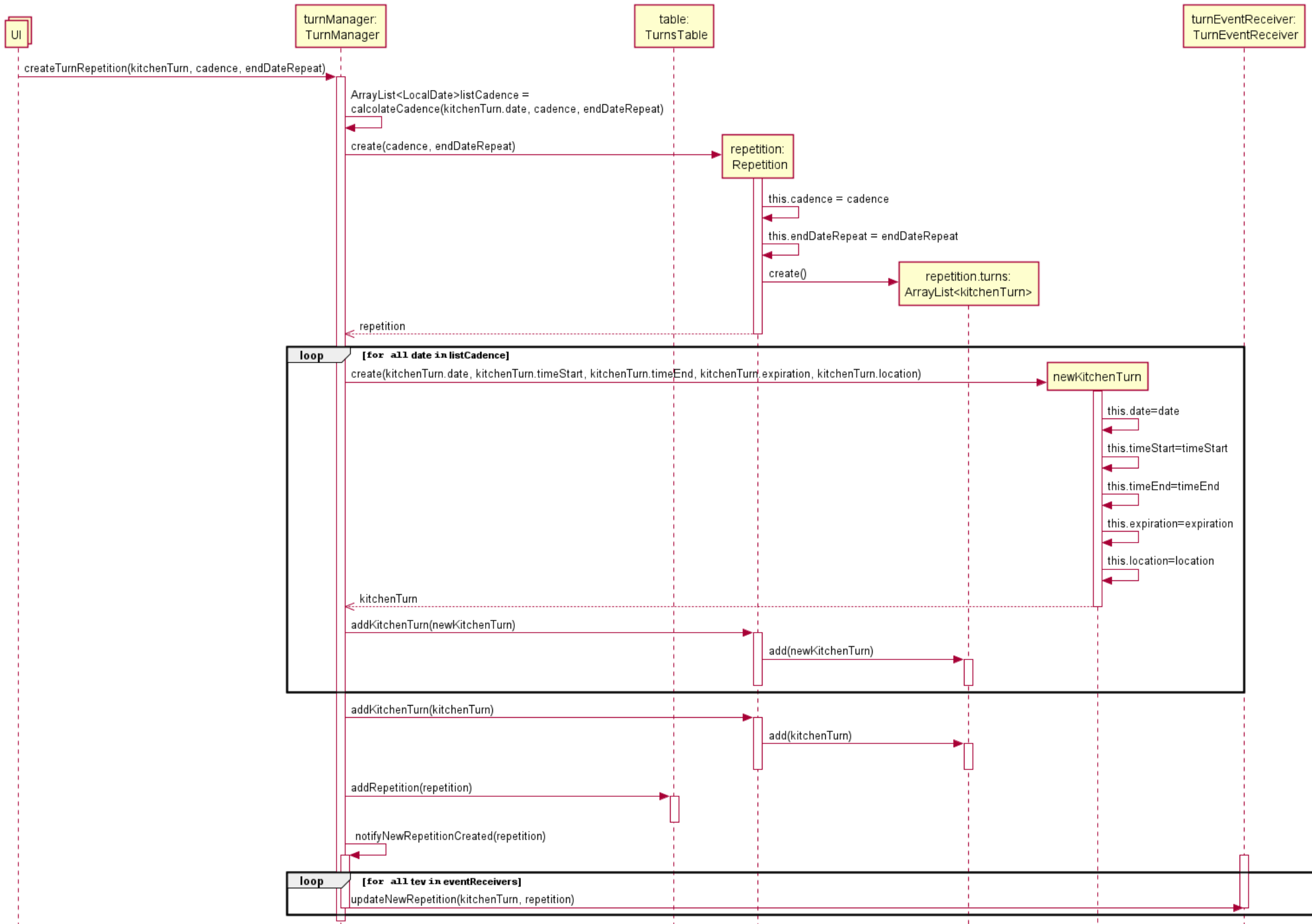
1a.3 modifyRepetition



3c.1 addTurnToGroup



4 createTurnRepetition



(1-4)a.2 createServiceTurn

