

Sviluppo delle Applicazioni Software

Laboratorio a.a 2021/2022

Laurentiu Apostol, Chiesa Enrico

Gestire compiti in cucina

Informazioni generali

Nome caso d'uso: Gestire compiti in cucina

Portata: Sistema

Livello: Obiettivo utente

Attore primario: Chef

Parti Interessate: Cuoco, Chef

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

Garanzie di successo o post-condizioni: Gli assegnamenti sono stati registrati

#	Attore	Sistema
1	Genera il foglio riepilogativo per un servizio di un evento (di cui ha ricevuto l'incarico)	Pre-compila il foglio riepilogativo per il servizio specificato in modo che contenga un compito per ogni mansione prevista dal menù associato al servizio
	<i>Se desidera prosegue con il passo 2, altrimenti termina il caso d'uso</i>	
2	Opzionalmente aggiunge una mansione all'elenco delle cose da fare	Aggiorna il foglio riepilogativo con i nuovi compiti derivati dalla mansione (e dalle mansioni in essa contenute)
	<i>Ripete 2 finché non è soddisfatto</i>	
3	Opzionalmente, ordina l'elenco	Aggiorna il foglio riepilogativo con il nuovo ordinamento.
4	Opzionalmente, consulta tabellone dei turni	Fornisce il tabellone dei turni
5	Assegna un compito specificando cosa (compito), quando (turno) e opzionalmente chi (cuoco)	Completa il compito registrando il turno e il cuoco, se specificato
6	Opzionalmente, indica una stima del tempo richiesto per lo svolgimento di un compito, e la quantità/porzioni da preparare	Registra tempo e quantità/porzioni

	Ripete dal passo 5 sinché non è soddisfatto oppure termina il caso d'uso	
--	--	--

Eccezione 1a

#	Attore	Sistema
1a.1	Genera il foglio riepilogativo di un servizio non suo	L'utente che sta generando un foglio riepilogativo non è lo stesso utente al quale appartiene l'evento del servizio specificato, pertanto non si può proseguire
	Termina il caso d'uso	

Estensione 1a

#	Attore	Sistema
1a.1	Apri un foglio riepilogativo, precedentemente compilato, per un servizio di un evento	Fornisce il foglio riepilogativo richiesto

Eccezione 1a.a

#	Attore	Sistema
1a.a.1	Apri un foglio riepilogativo di un servizio non suo	L'utente che sta aprendo un foglio riepilogativo non è lo stesso utente al quale appartiene l'evento del servizio specificato, pertanto non si può proseguire
	Termina il caso d'uso	

Estensione 2a

#	Attore	Sistema
2a.1	Divide un compito opzionalmente fornendo i dosaggi dei due nuovi compiti	Aggiunge al foglio riepilogativo i due nuovi compiti con i dosaggi forniti

Estensione 2b

#	Attore	Sistema
2b.1	Rimuove un compito dall'elenco delle cose da fare	Rimuove il compito dal foglio riepilogativo

Eccezione 5.a

#	Attore	Sistema
5.a.1	E' stata scelto un cuoco e un turno, ma il cuoco non è disponibile per tale turno	L'assegnazione è invalida e verrà rifiutata
	<i>Può decidere di ripetere il passo 5 oppure di terminare il caso d'uso</i>	

Eccezione 5.b

#	Attore	Sistema
5.b.1	Il compito è già stato assegnato	L'assegnazione è invalida e verrà rifiutata
	<i>Può decidere di ripetere il passo 5 oppure di terminare il caso d'uso</i>	

Estensione 5a

#	Attore	Sistema
5a.1	Modifica il cuoco o il turno (o entrambi) di un compito	Registra la modifica

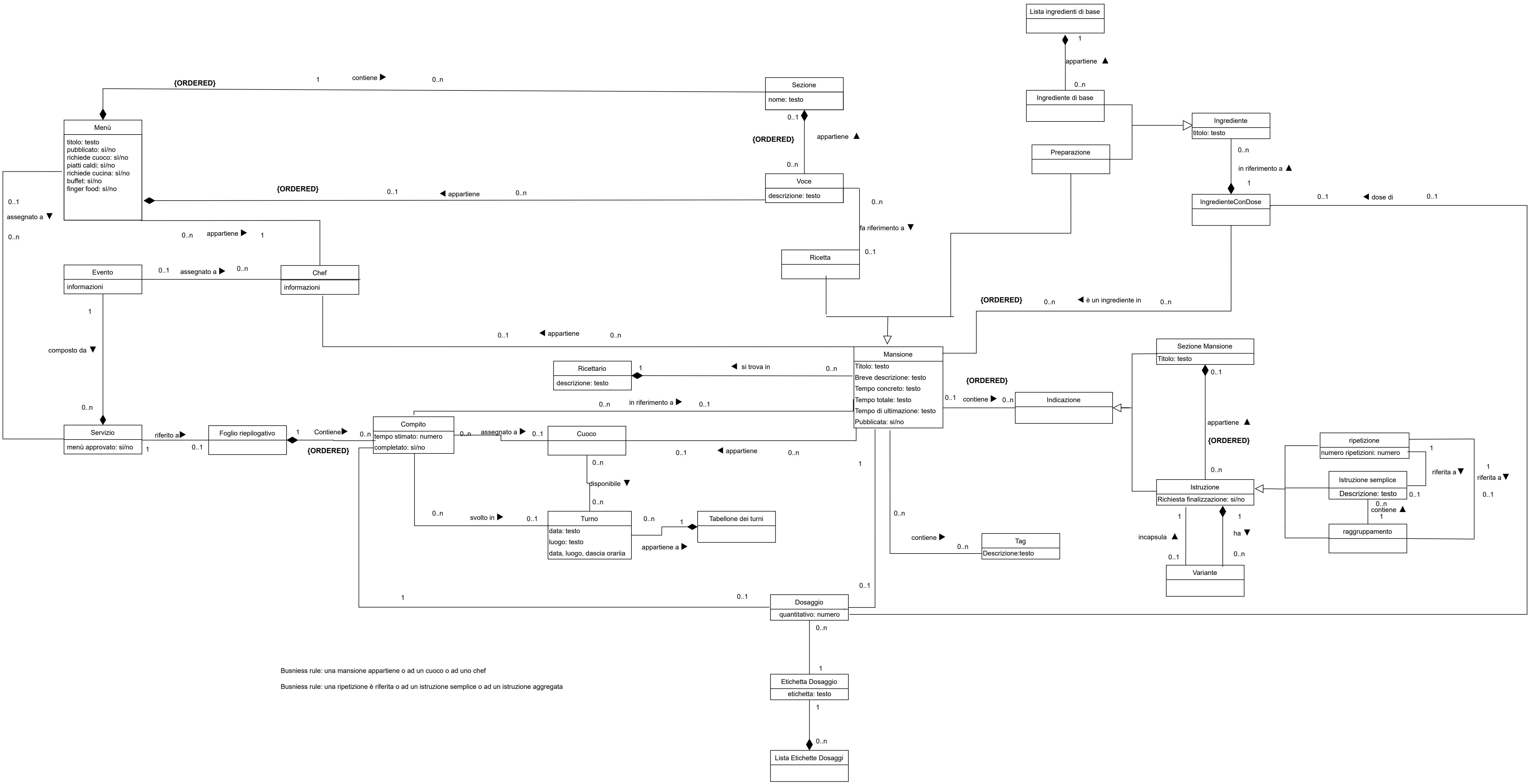
Eccezione 5a.a

#	Attore	Sistema
5a.a.1	E' stato modificato il cuoco o il turno (o entrambi), ma il cuoco non è disponibile per tale turno	La modifica è invalida e verrà rifiutata
	<i>Può decidere di ripetere il passo 5 oppure di terminare il caso d'uso</i>	

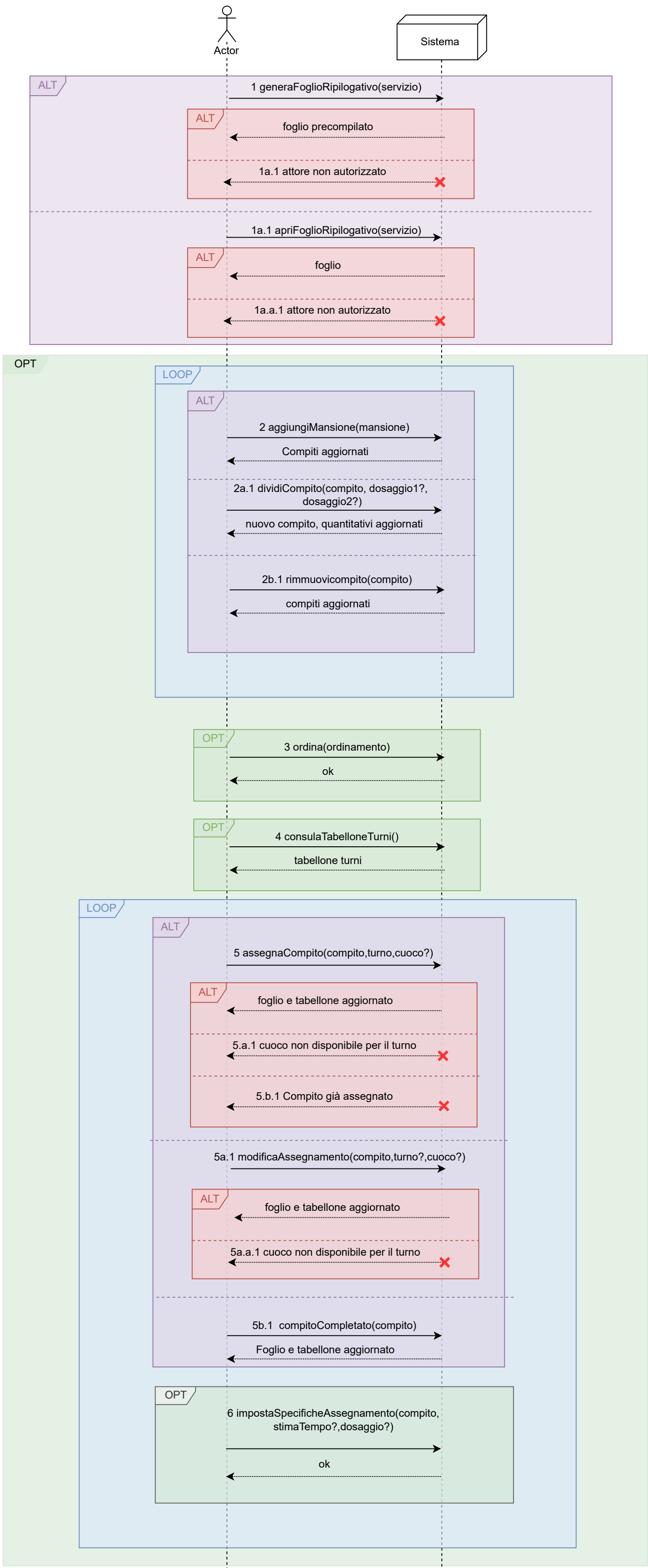
Estensione 5b

#	Attore	Sistema
5b.1	Specifica che una mansione è già completata	Registra l'informazione

Modello di dominio



SSD



Contratti per lo UC “Gestire compiti in cucina”

Pre-condizioni generali:

- l'attore è identificato con un'istanza *ch* di Chef
- esiste un'istanza di TabelloneDeiTurni *tabelloneTurni*

Notazioni generali:

- NECESSARIO(mansione: Mansione): insieme di ogni Preparazione *prep* tale che mansione **contiene** *prep* unito ad ogni *prep* tale che NECESSARIO(*prep*) [le preparazioni e le sottopreparazioni che fanno parte della mansione]
- OGNI_RICETTA(menu: Menu) insieme di ogni Ricetta *ric* tale che menu **contiene** *ric*

1. generaFoglioRipilogativo(servizio: Servizio)

Pre-condizioni:

- esiste un Evento *evento* tale che esiste uno Chef *chef* tale che *evento* **appartiene** a *chef* e servizio **appartiene** a *evento*
- esiste un Menu *menu* tale che *menu* **appartiene** a servizio e *menu* **appartiene** a *evento*
- *evento*.menuApprovato = sì

Post-condizioni: Se *chef* = *ch*:

- è stato creato un FoglioRiepilogativo *folRiep*
- il foglio riepilogativo è di proprietà di *ch*
- ogni Ricetta *ric* tale che *ric* è in OGNI_RICETTA(*menu*):
 - è stato creato un nuovo Compito *comp*
 - [se esiste un Dosaggio *dos* tale che *ric* **contiene** *dos*] è stato creato *dos1*, con *dos1*.quantitativo = *dos*.quantitativo, e esiste EtichettaDosaggio *etDos* tale che *dos* **contiene** *etDos*, *dos1* **contiene** *etDos*, e *comp* **contiene** *dos1*
 - *comp* **contiene** *ric*
 - *folRiep* **contiene** *comp*
 - *comp*.completato = no
- ogni Ricetta *ric* tale che *ric* è in OGNI_RICETTA(*menu*), ogni Preparazione *prep* tale che *prep* è in NECESSARIO(*ric*):
 - è stato creato un nuovo Compito *comp*
 - [se esiste un Dosaggio *dos* tale che *prep* **contiene** *dos*] è stato creato *dos1*, con *dos1*.quantitativo = *dos*.quantitativo, e esiste EtichettaDosaggio *etDos* tale che *dos* **contiene** *etDos*, *dos1* **contiene** *etDos*, e *comp* **contiene** *dos1*
 - *comp* **contiene** *prep*
 - *folRiep* **contiene** *comp*
 - *comp*.completato = no

1a.1 apriFoglioRiepilogativo(servizio: Servizio)

Pre-condizioni:

- esiste un FoglioRiepilogativo *folRiep* tale che servizio è riferito a *folRiep*
- esiste un Evento *evento* tale servizio appartiene a *evento* ed esiste uno Chef *chef* tale che *evento* appartiene a *chef*

Post-condizioni: ---

Non ci sono post-condizioni in quanto si tratta di un'operazione consultiva

2 aggiungiMansione(mansione: Mansione)

Pre-condizioni:

- è in corso la definizione di un FoglioRiepilogativo *folRiep*
- Qualsiasi Mansione può **contenere** un Dosaggio *dos*
- [se esiste *dos*] *dos* **contiene** EtichettaDosaggio *etDos*

Post-condizioni:

- è stato creato un nuovo Compito *comp*
- [se esiste *dos* tale che mansione **contiene** *dos*]
 - è stato creato *dos1*
 - *dos1.quantitativo* = *dos.quantitativo*
 - *dos1* **contiene** *etDos*
 - *comp* **contiene** *dos1*
- *comp* **contiene** mansione
- *folRiep* **contiene** *comp*
- *comp.completato* = no
- ogni tale che *prep* è in NECESSARIO(mansione):
 - è stato creato un nuovo Compito *comp*
 - [se esiste *dos* tale che *prep* **contiene** *dos*]
 - è stato creato *dos1*
 - *dos1.quantitativo* = *dos.quantitativo*
 - *dos1* **contiene** *etDos*
 - *comp* **contiene** *dos1*
 - *comp* **contiene** *prep*
 - *folRiep* **contiene** *comp*
 - *comp.completato* = no

2a.1 dividiCompito(compito: Compito, dosaggio1?: Dosaggio, dosaggio2?: Dosaggio)

Pre-condizioni:

- è in corso la definizione di un FoglioRiepilogativo *folRiep*
- compito si riferisce ad una Mansione *man*

Post-condizioni:

- è stato creato un nuovo Compito *comp*
- [se specificato dosaggio1] *comp* **contiene** dosaggio1
- [se specificato dosaggio2] compito **contiene** dosaggio2
- *man* **contiene** *comp*
- *folRiep* **contiene** *comp*
- *comp.completato* = no

2b.1 rimuoviCompito(compito: Compito)**Pre-condizioni:**

- è in corso la definizione di un FoglioRiepilogativo *folRiep*
- *folRiep* **contiene** compito

Post-condizioni:

- *folRiep* **non contiene** compito

3 ordina(ordinamento)

Pre-condizioni: è in corso la definizione di un FoglioRiepilogativo *folRiep*

Post-condizioni: l'associazione **contiene** fra *folRiep* e i compiti è modificata in accordo a ordinamento

4 consulaTabelloneTurni()

Pre-condizioni:---

Post-condizioni:---

Non ci sono ne pre-condizioni ne post-condizioni in quanto si tratta di un'operazione consultiva e non può andare incontro ad eccezioni

5 assegnaCompito(compito: Compito, turno: Turno, cuoco?: Cuoco)**Pre-condizioni:**

- è in corso la definizione di un FoglioRiepilogativo *folRiep*
- turno **appartiene a** *tabelloneTurni*
- *folRiep* **contiene** compito

Post-condizioni: Se non esiste un Turno *turn* tale che compito **svolto in** *turn* e [Se specificato cuoco] se cuoco **è disponibile per** turno:

- compito **svolto in** turno
- [se specificato cuoco] *comp* **assegnato a** cuoco

(non c'è un'operazione esplicita per aggiornare il tabellone dei turni perché ogni turno deve appartenere al *tabelloneTurni* e ogni turno può essere assegnato al massimo ad un compito)

5a.1 modificaAssegnamento(compito: Compito, turno?: Turno, cuoco?: Cuoco)

Pre-condizioni:

- è in corso la definizione di un FoglioRiepilogativo *folRiep*
- [Se specificato turno] turno **appartiene a** *tabelloneTurni*
- *folRiep* **contiene** compito

Post-condizioni: Se specificato cuoco e se cuoco è disponibile per turno:

- [se esiste *cu* tale che *cu* è assegnato a compito] *cu* **non è assegnato a** compito
- [se esiste *turn* tale che compito è svolto in *turn*] compito **non è svolto in** *turn*
- [se specificato turno] compito **svolto in** turno
- [se specificato cuoco] *comp* **assegnato a** cuoco

5b.1 compitoCompletato(compito: Compito)

Pre-condizioni:

- è in corso la definizione di un FoglioRiepilogativo *folRiep*
- *folRiep* **contiene** compito

Post-condizioni:

- compito.completato = sì

6 impostaSpecificheAssegnamento(compito: Compito, stimaTempo?: numero, dosaggio?: Dosaggio)

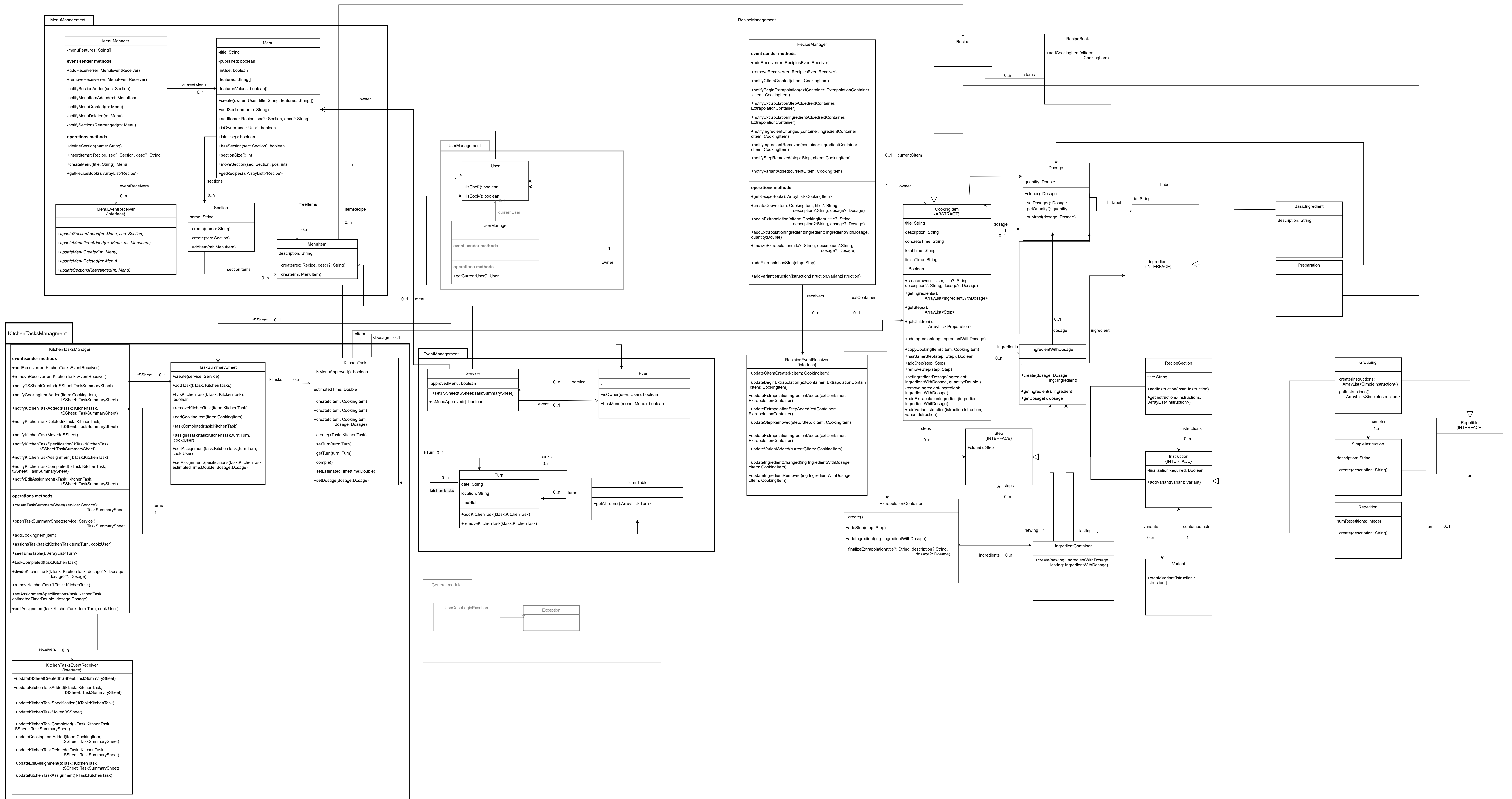
Pre-condizioni:

- è in corso la definizione di un FoglioRiepilogativo *folRiep*
- *folRiep* **contiene** compito

Post-condizioni:

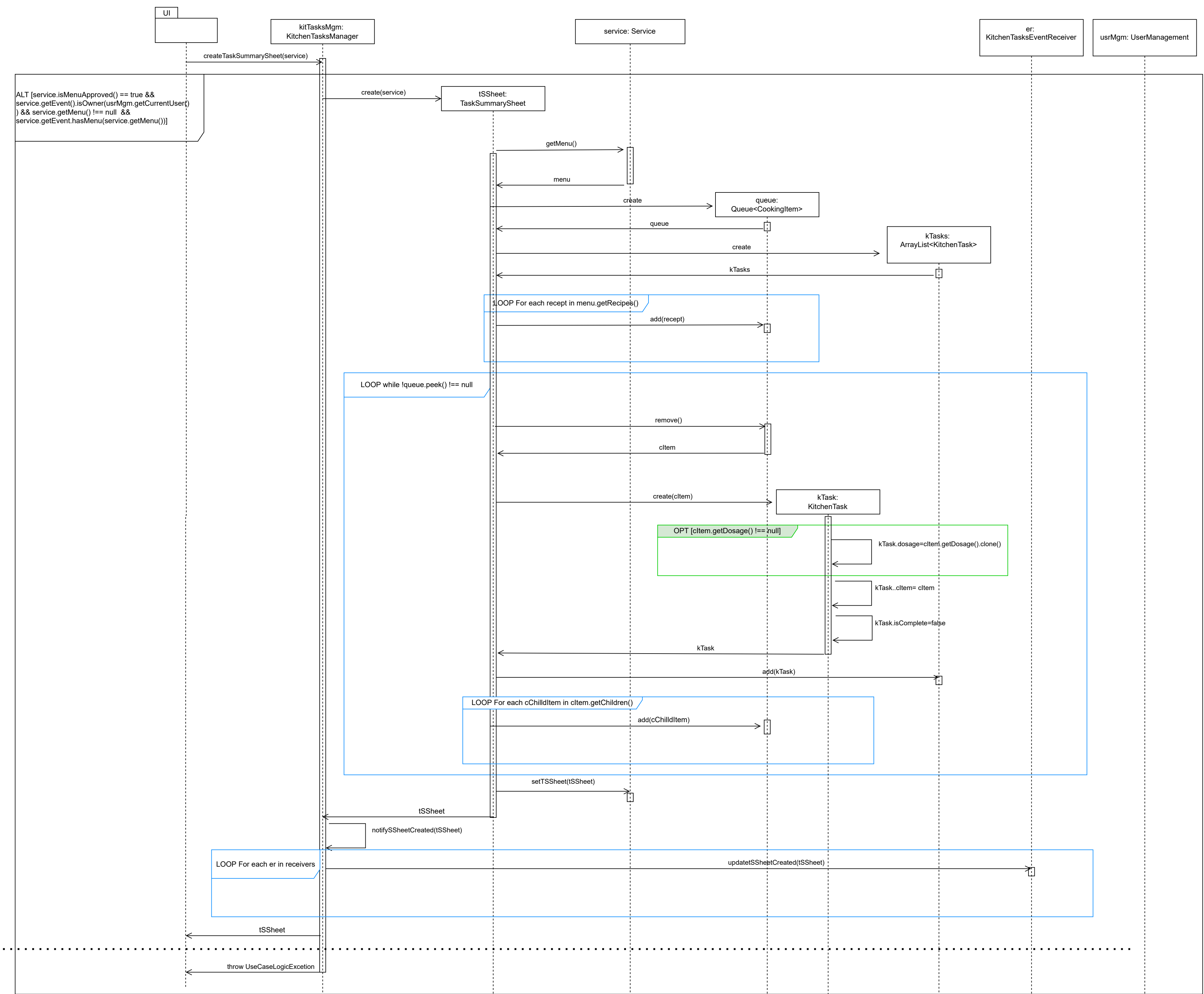
- [se specificato stimaTempo] compito.stimaTempo = stimaTempo
- [se specificato dosaggio] compito **contiene** dosaggio

Design Class Diagram (DCD)

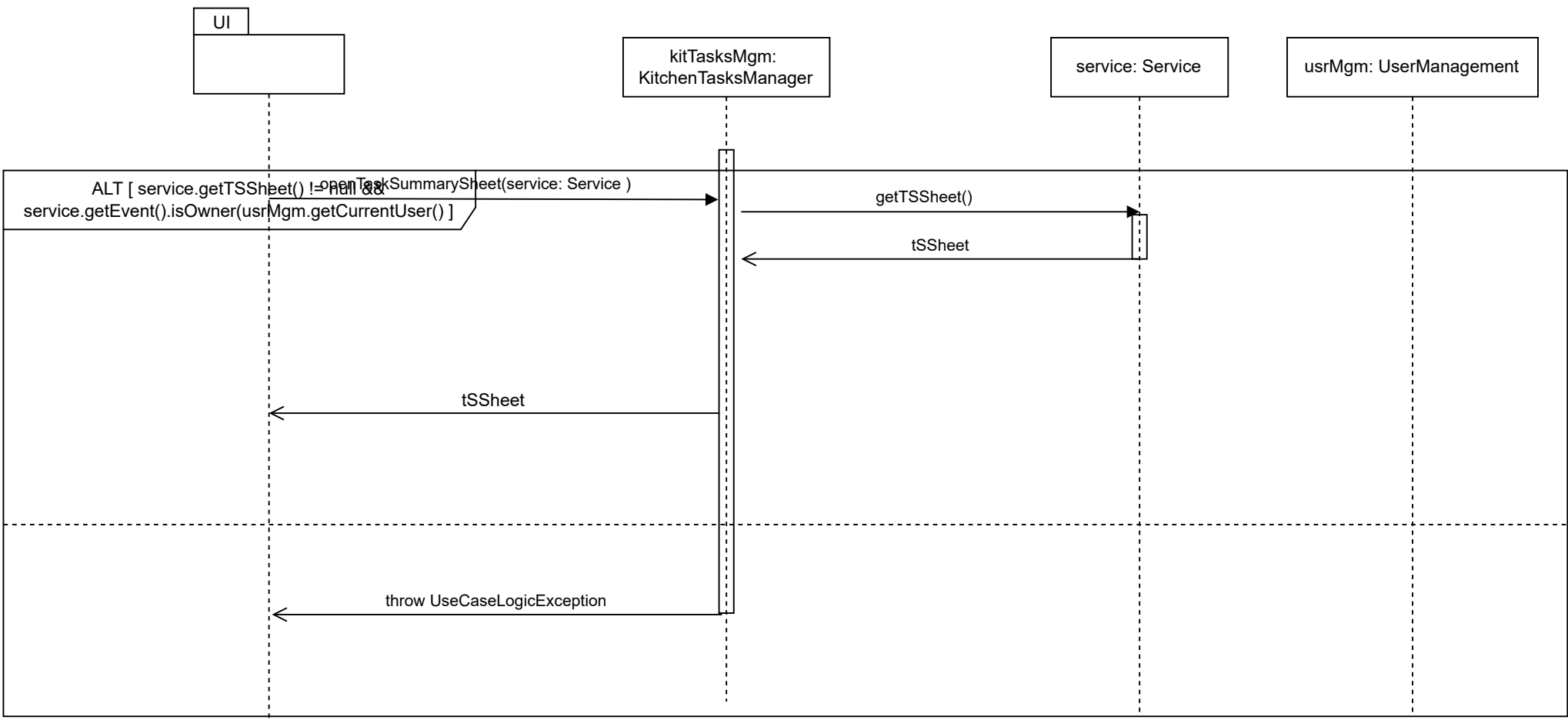


DSD

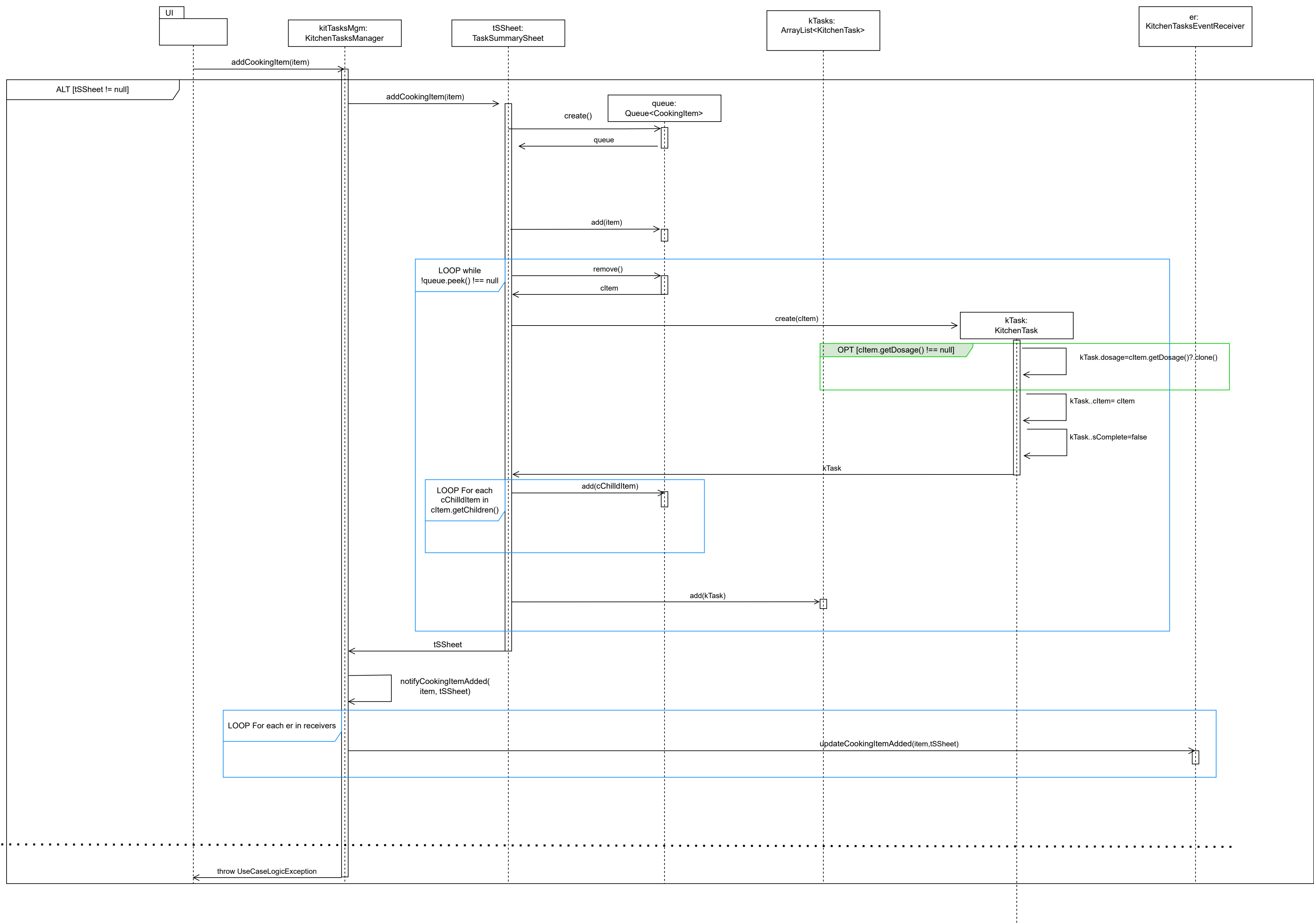
1. createTaskSummarySheet(service: Service)



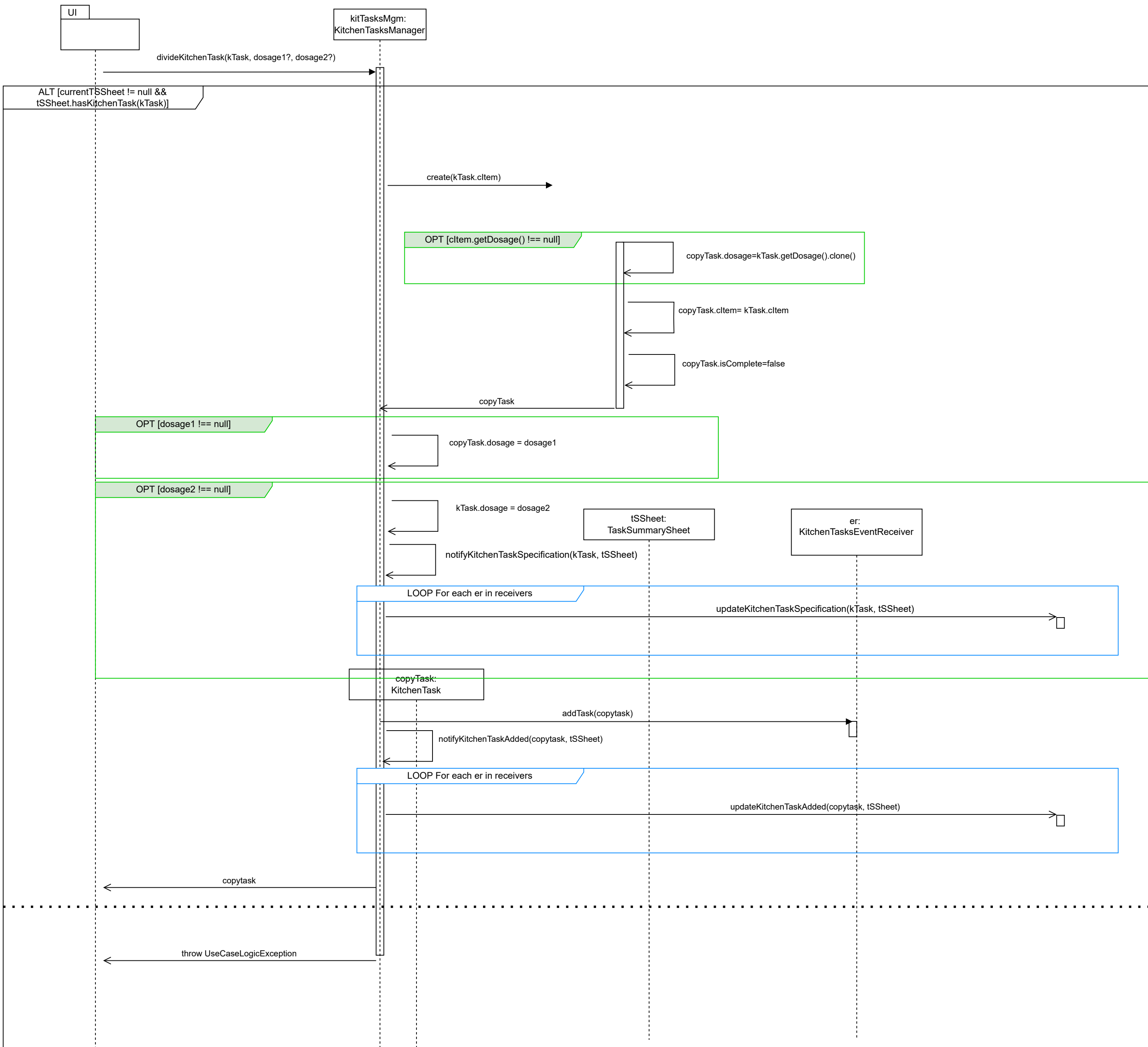
1a.1 openTaskSummarySheet(service: Service)



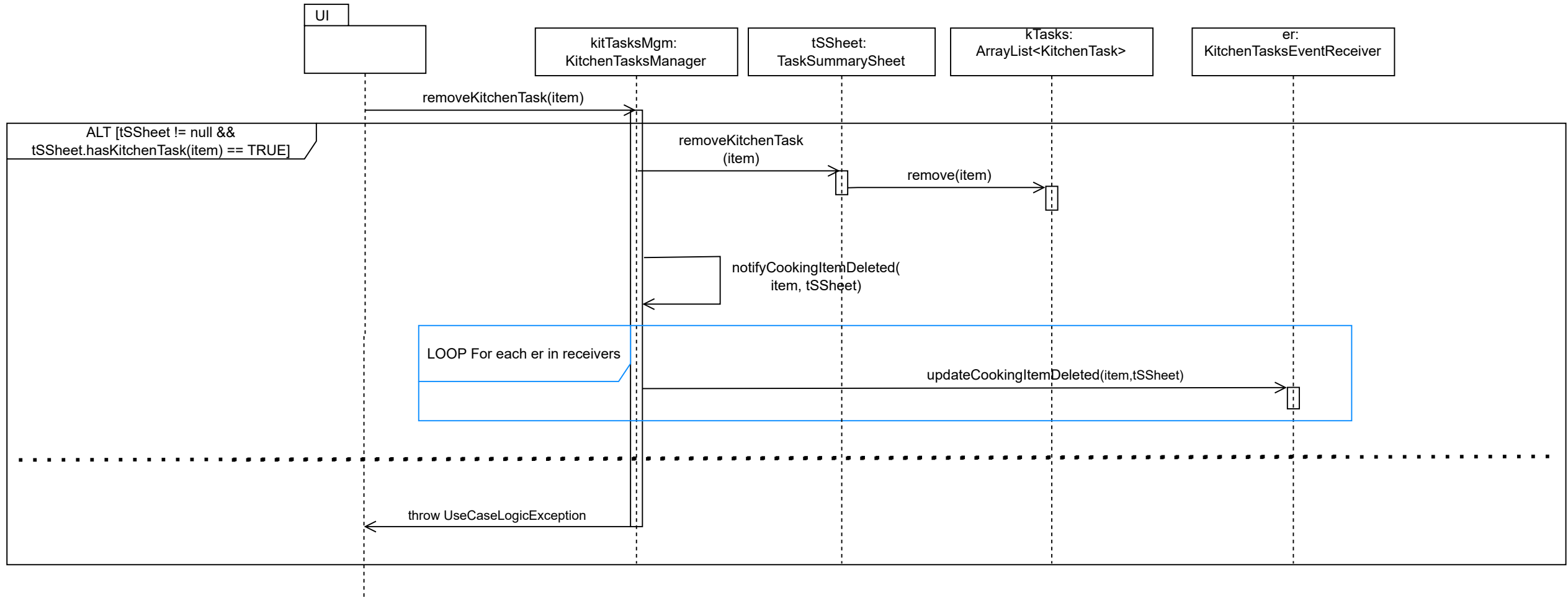
2. addCookingItem(item: CookingItem)



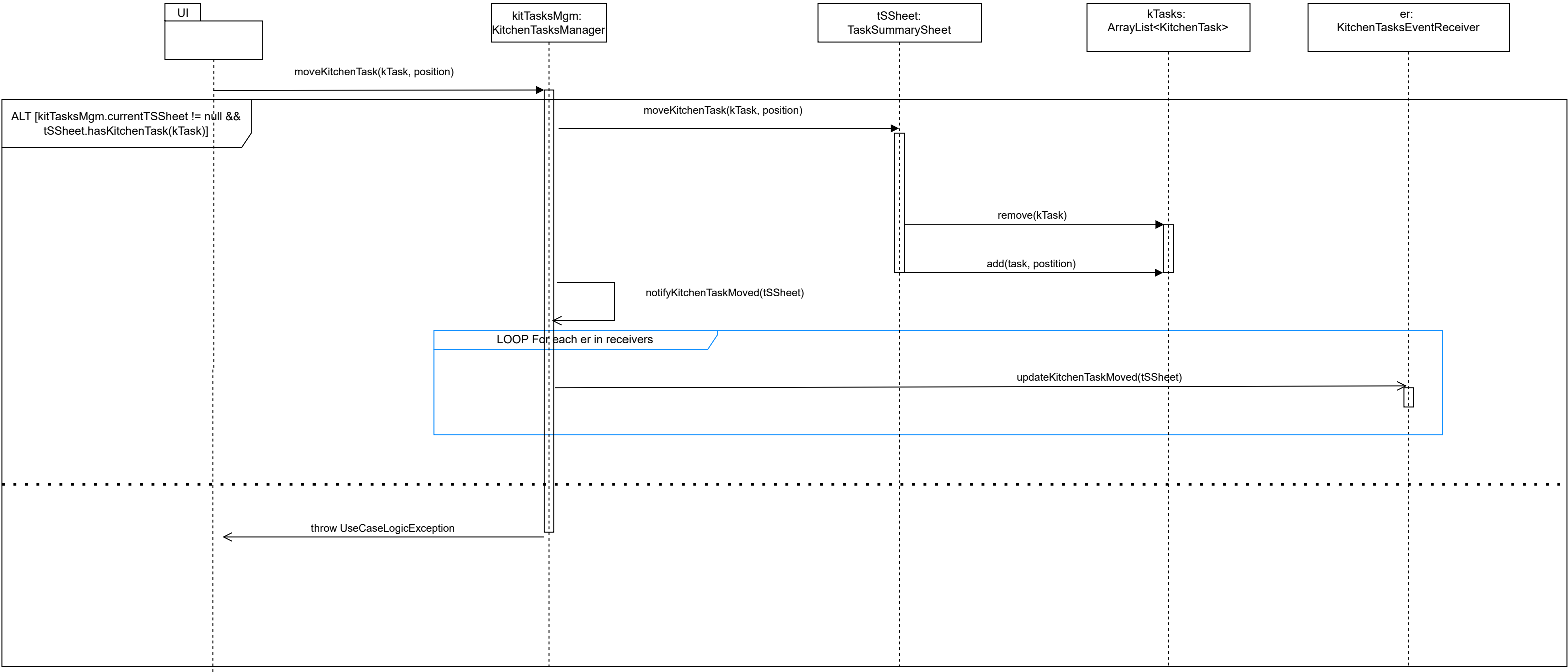
2a.1 divideKitchenTask(kTask: KitchenTask, dosage1?: Dosage, dosage2?: Dosage)



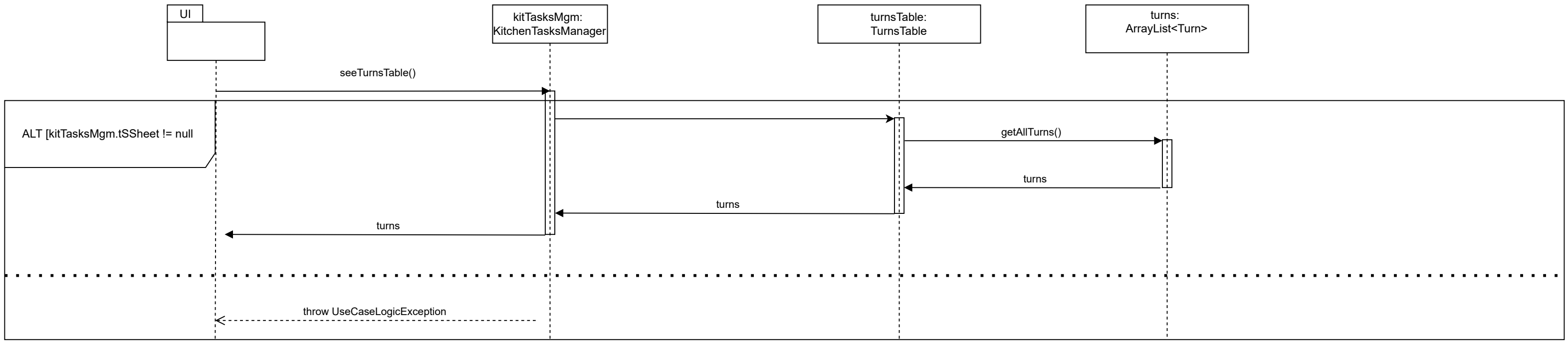
2.b.1 removeKitchenTask(item: KitchenTask)



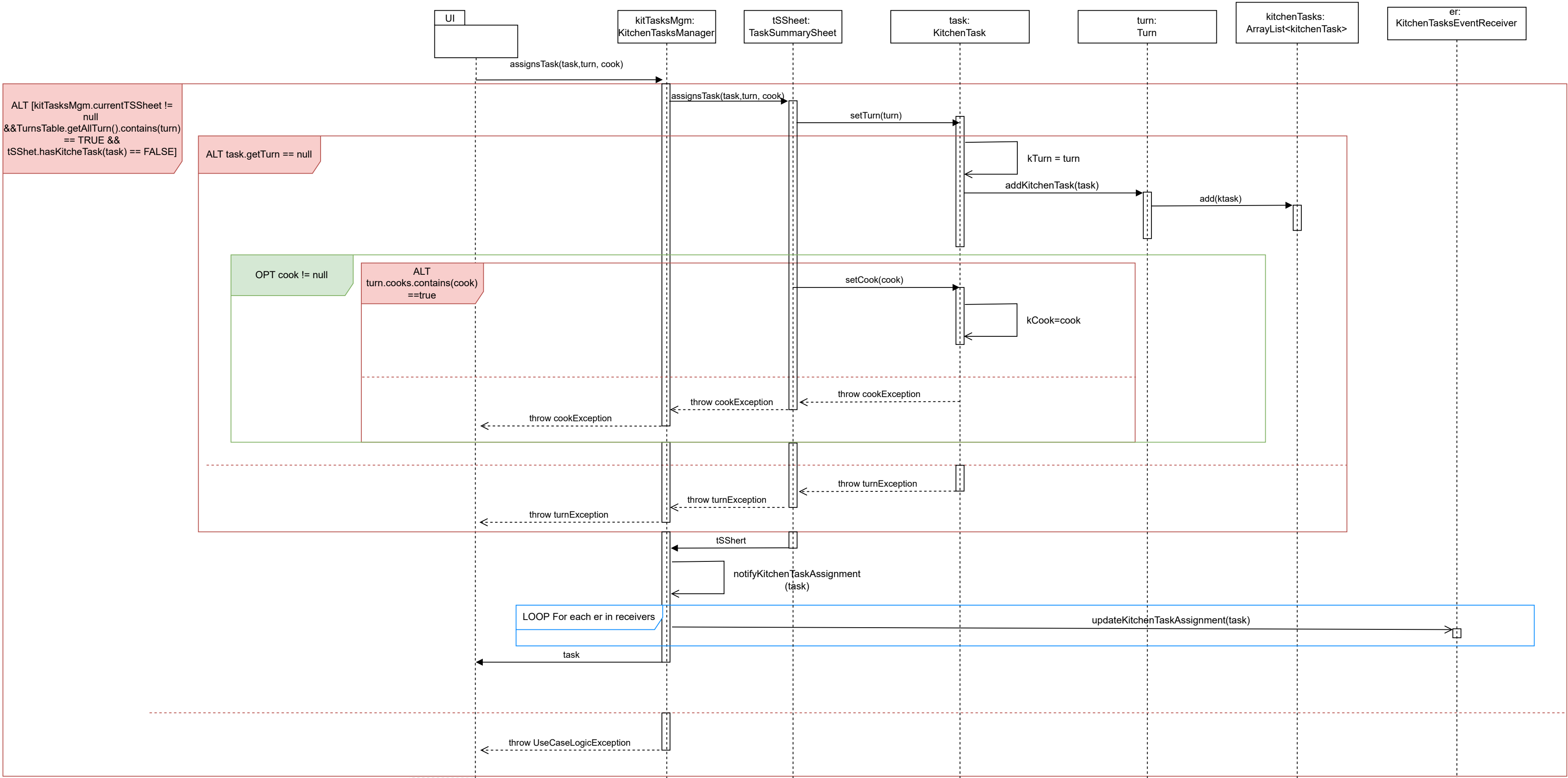
3 moveKitchenTask(kTask: KitchenTask, position: Integer)



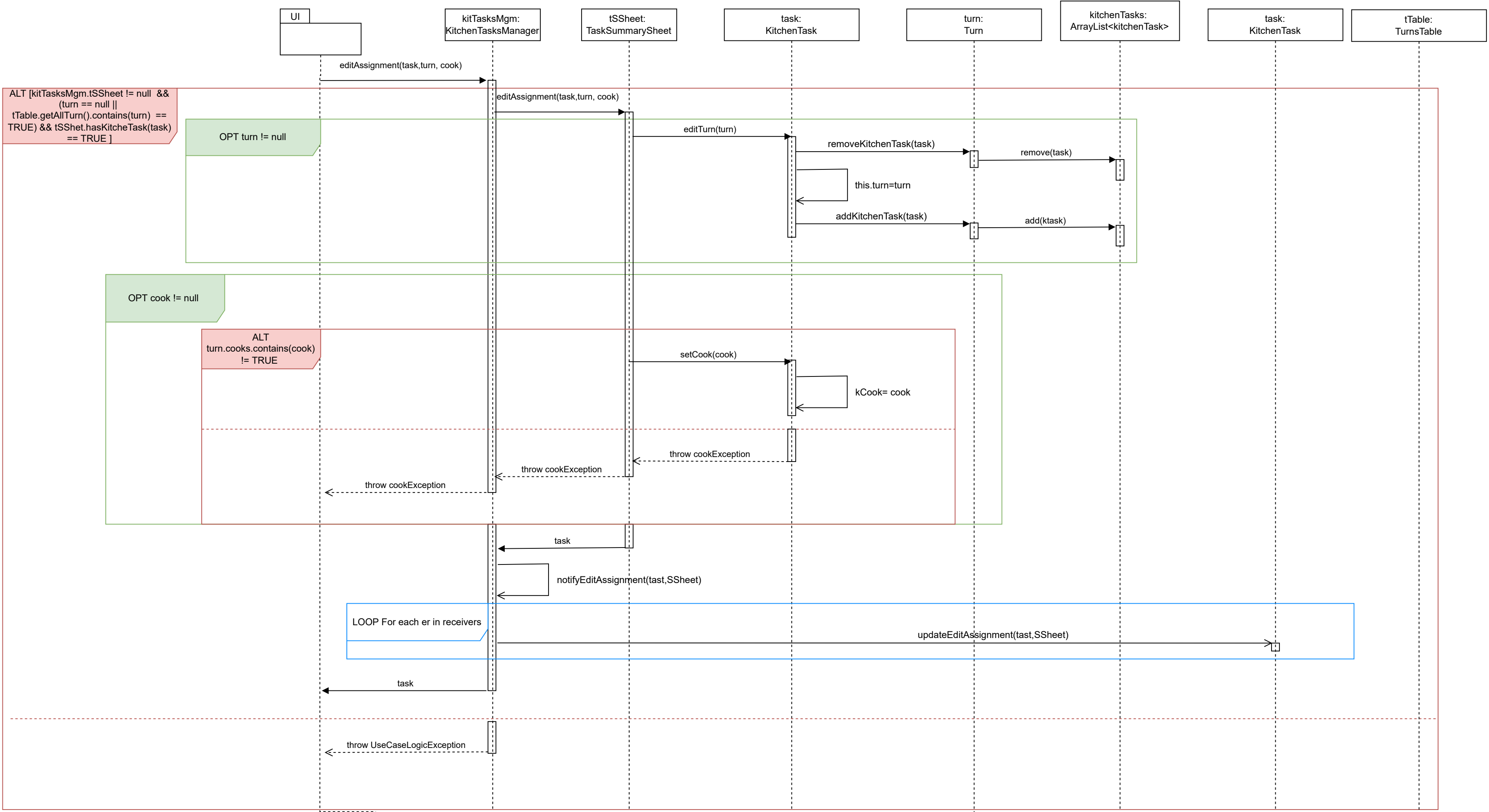
4 seeTurnsTable()



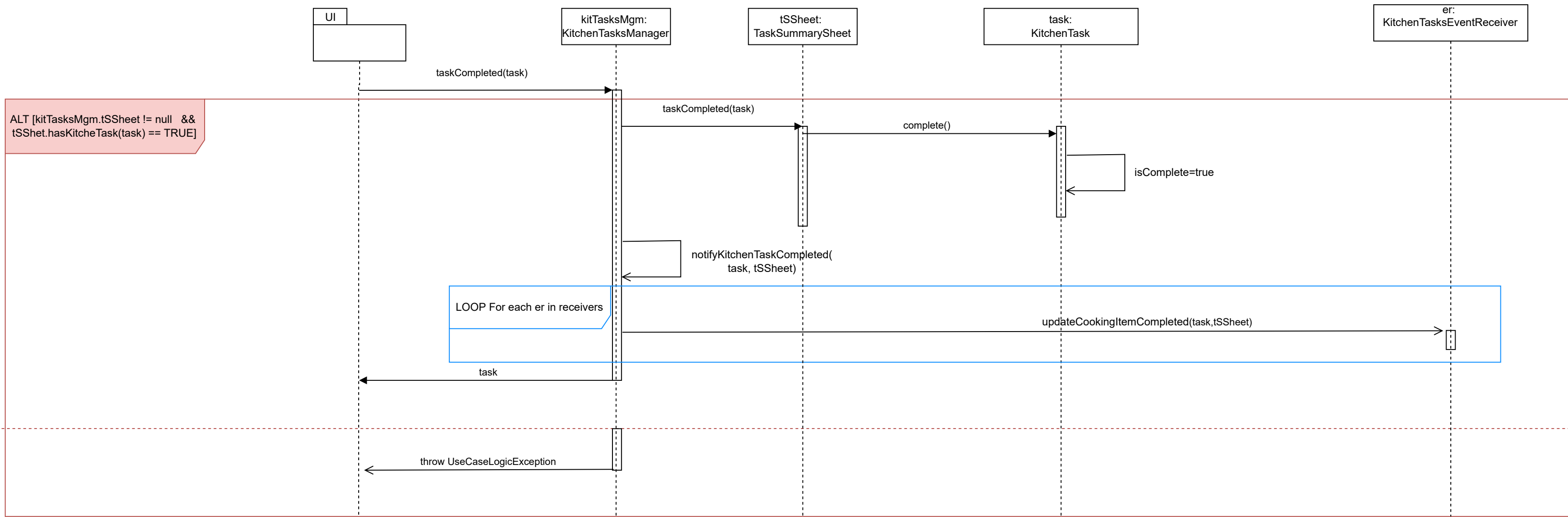
5 assignsTask(task:KitchenTask,turn:Turn, cook:User)



5.a editAssignment(task:KitchenTask,turn:Turn?, cook:User?)



5.b taskCompleted(task:KitchenTask)



6 setAssignmentSpecifications(task:KitchenTask, estimatedTime:Double, dosage:Dosage)

