Gestire i compiti della cucina

Informazioni generali

Nome caso d'uso: Gestire i compiti della cucina

Portata: Sistema

Livello: Obiettivo utente **Attore primario:** Chef

Parti Interessate: Chef, Cuoco

Pre-condizioni:

• L'attore è autenticato come Chef

- Esiste almeno un evento in corso assegnato allo Chef
- È presente il menù per il servizio legato all'evento
- Sono disponibili informazioni sui turni e sulle disponibilità dei cuochi

Garanzie di successo o post-condizioni:

- I compiti sono correttamente registrati nel sistema
- Il foglio riepilogativo del servizio è aggiornato con i nuovi assegnamenti
- Il tabellone dei turni riflette i nuovi compiti assegnati

Scenario principale di successo

#	Attore	Sistema
1	Genera il foglio riepilogativo di un servizio di un evento (di cui ha ricevuto l'incarico)	Presenta il foglio riepilogativo pre- compilato con le ricette e preparazioni previste dal menù associato al servizio
	Se desidera prosegue con il passo 2, altrimenti termina il caso d'uso	
2	Opzionalmente , aggiunge preparazioni e ricette all'elenco delle cose da fare	Aggiorna il foglio riepilogativo con le nuove preparazioni e ricette
3	Opzionalmente, ordina l'elenco	Aggiorna il foglio riepilogativo con il nuovo ordinamento
4	Opzionalmente, consulta tabellone dei turni	Mostra il tabellone dei turni
5	Opzionalmente specifica per un compito il tempo richiesto per il suo svolgimento, e la quantità/porzioni preparate.	Aggiorna il foglio riepilogativo con il compito modificato
6	Opzionalmente assegna un compito ad un cuoco in un determinato turno.	Aggiorna il foglio riepilogativo con la nuova assegnazione
	Ripete dal passo 4 fino al raggiungimento del risultato desiderato	

Estensione 1a

#	Attore	Sistema
1a.1	Lo chef apre un foglio riepilogativo già generato	Mostra il foglio riepilogativo relativo

Eccezione 1a.1a

#	Attore	Sistema
1a.1a.1	Lo chef apre un foglio riepilogativo già generato	Il foglio riepilogativo non è di proprietà dello chef
	Termina il caso d'uso	

Estensione 2a

#	Attore	Sistema
2a.1	Lo chef seleziona una preparazione o ricetta da rimuovere	Il sistema aggiorna il foglio riepilogativo rimuovendo l'elemento selezionato

Estensione 5a

#	Attore	Sistema
5a.1	Lo chef seleziona una preparazione e la marca come già pronta	Aggiorna il foglio riepilogativo indicando che la preparazione non richiede assegnazione

Estensione 5b

#	Attore	Sistema
5b.1	Seleziona un turno e lo marca come saturo	Registra lo stato di saturazione e lo rende visibile agli altri chef

Eccezione 6a

#	Attore	Sistema
6a.1	Opzionalmente assegna un compito ad un cuoco in un determinato turno.	Il turno è stato precedentemente marcato come saturo
	Termina il caso d'uso	

Eccezione 6b

#	Attore	Sistema
6b.1	Opzionalmente assegna un compito ad un cuoco in un determinato turno.	Il cuoco non è disponibile nel turno selezionato
	Termina il caso d'uso	

Eccezione 6c

#	Attore	Sistema
6c.1	Opzionalmente assegna un compito ad un cuoco in un determinato turno.	Il tempo disponibile per lo svolgimento del compito non è sufficiente
	Termina il caso d'uso	

Estensione 6a

#	Attore	Sistema
6a.1	Modifica assegnamento compito	Aggiorna il foglio riepilogativo con la nuova assegnazione

Eccezione (6a.1)a

#	Attore	Sistema
(6a.1)a.1	Modifica assegnamento compito	Il turno è stato precedentemente marcato come saturo
	Termina il caso d'uso	

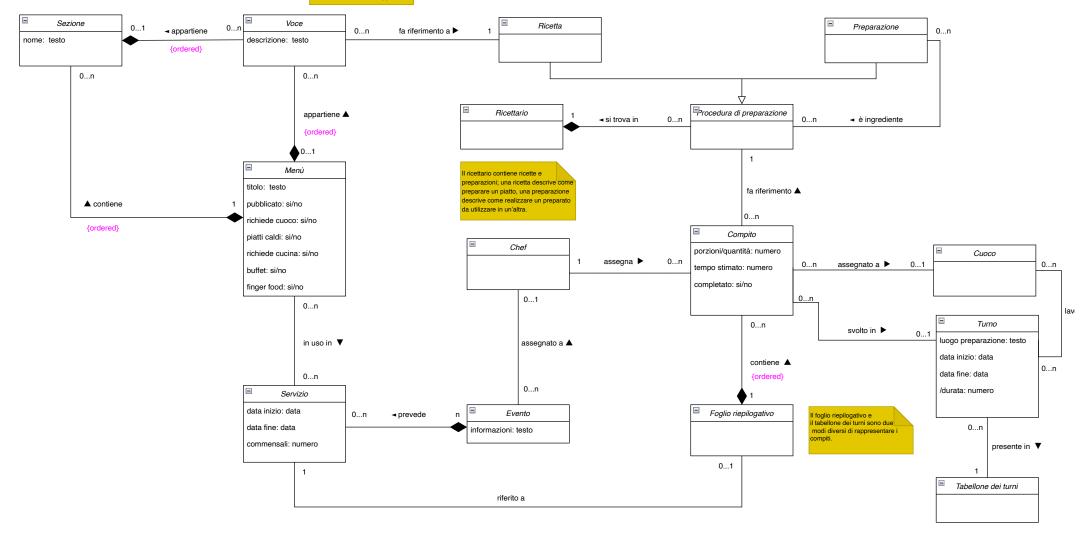
Eccezione (6a.1)b

#	Attore	Sistema
(6a.1)b.1	Modifica assegnamento compito	Il cuoco non è disponibile nel turno selezionato
	Termina il caso d'uso	

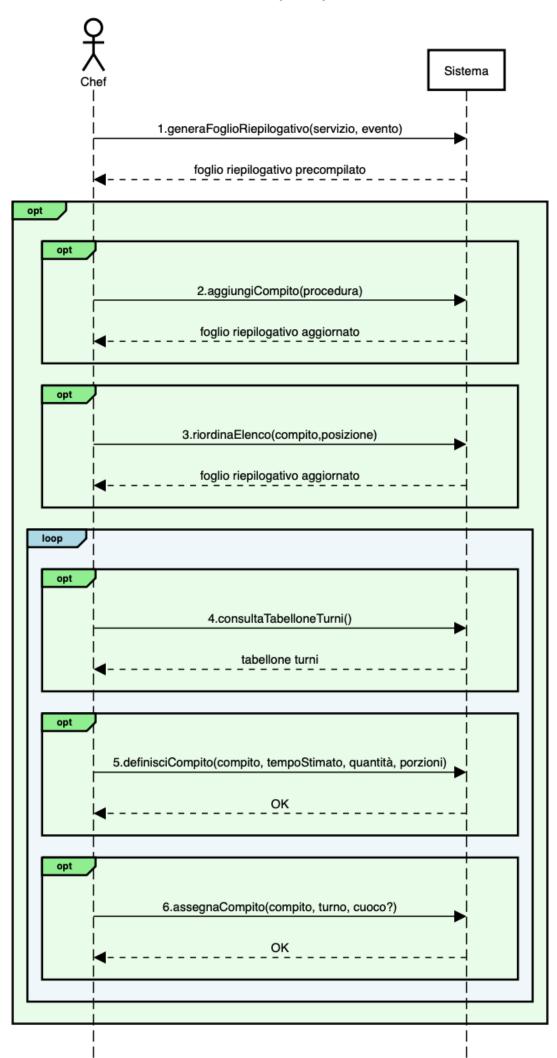
Eccezione (6a.1)c

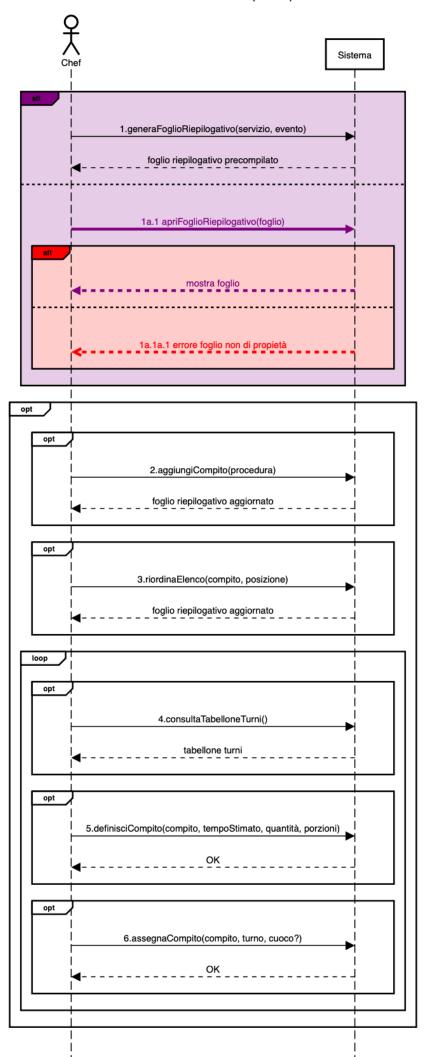
#	Attore	Sistema
(6a.1)c.1	Modifica assegnamento compito	Il tempo disponibile per lo svolgimento del compito non è sufficiente
	Termina il caso d'uso	

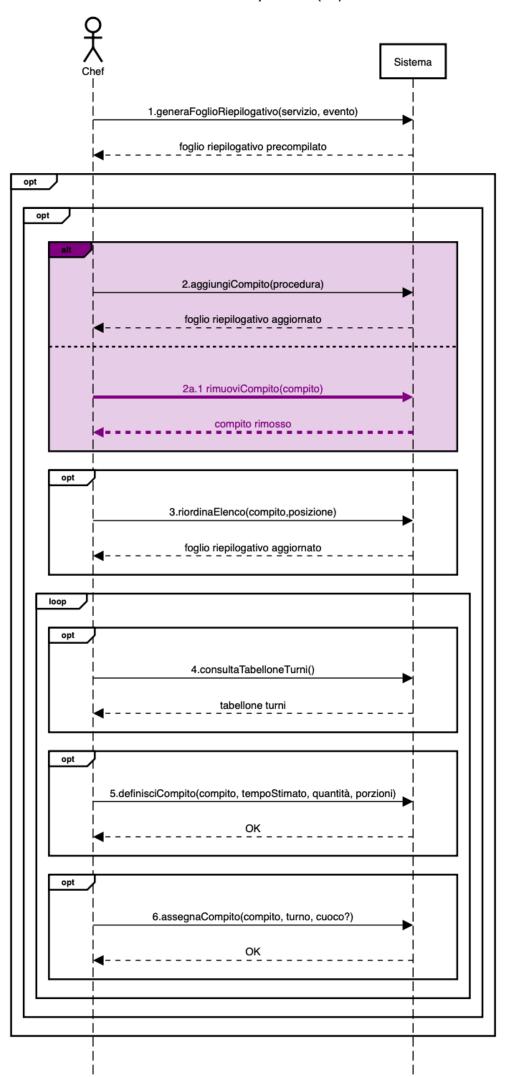
Una voce può comparire in un menù all'interno di una sezioneoppure come voce "libera" quindi verrà valorizzata una e una sola delle due relazioni appartiene

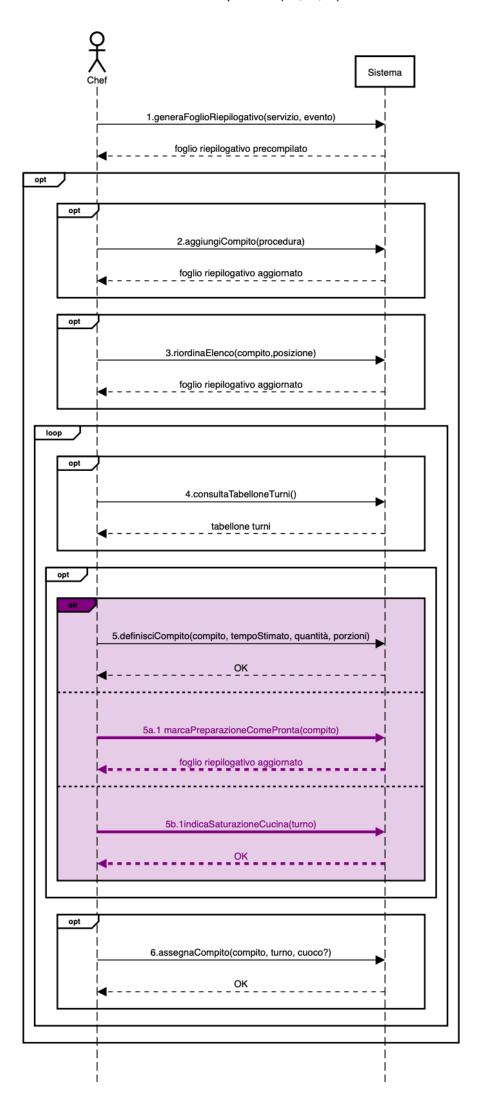


Scenario principale

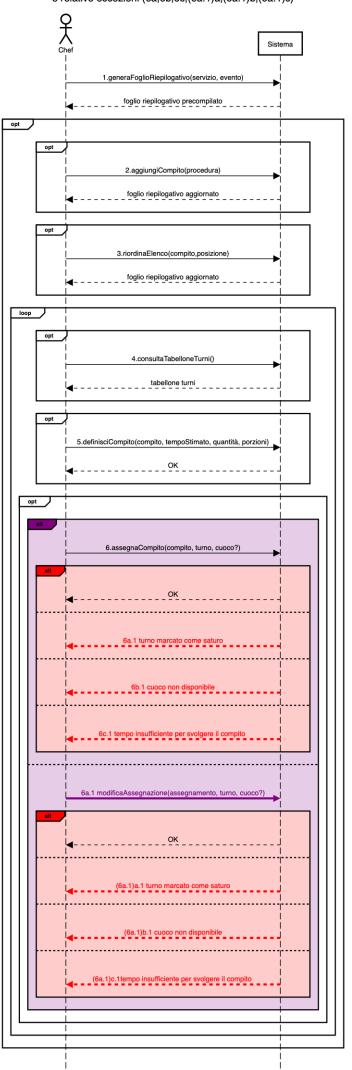








Estensione al passo 6 (6a) e relative eccezioni (6a,6b,6c,(6a.1)a,(6a.1)b,(6a.1)c)



Contratti per lo UC "Gestire i compiti della cucina"

Pre-condizione generale: l'attore è identificato con un'istanza *ch* di Chef, che è proprietario di istanza di Menù *menù.*

1.generaFoglioRiepilogativo(servizio: Servizio, evento: Evento)

Precondizioni:

- servizio è associato a evento
- evento risulta assegnato allo Chef ch

Post-condizioni:

- È stato generato un Foglio Riepilogativo foglio
- foglio si riferisce al servizio e all'evento specificati

1a.1 apriFoglioRiepilogativo(foglio: FoglioRiepilogativo)

Precondizioni:

• esite un'istanza foglio di Foglio Riepilo gativo

Precondizioni:

- <u>servizio</u> è associato a <u>evento</u>
- evento risulta assegnato allo Chef ch

Post-condizioni:

- È stato aperto un Foglio Riepilogativo foglio e reso disponibile per la gestione
- foglio si riferisce al servizio e all'evento specificati
- foglio contiene i compiti associati a servizio

2.aggiungiCompito(procedura: Procedura Preparazione)

Precondizioni:

- È in corso la gestione dei compiti relativi a un Foglio Riepilogativo foglio
- procedura si trova in ricettario
- procedura non è già inclusa nel foglio

Post-condizioni:

- È stata creata un'istanza c di Compito
- *c* fa riferimento a <u>procedura</u>
- foglio contiene c

2a.1 rimuoviCompito(compito): Compito)

Precondizioni:

- compito fa riferimento ad un'istanza procedura di Procedura
- foglio contiene compito

Postcondizioni:

• *foglio* non contiene più *c*

3.riordinaElenco(compito: Compito, posizione: numero)

Precondizione:

- foglio contiene compito
- posizione è numero compreso tra 1 e la lunghezza della lista di compiti presenti in foglio

Post-condizione:

• compito si trova nella posizione posizione della lista di compiti presenti in foglio

4.consultaTabelloneTurni()

Precondizioni: --Post-condizioni: ---

5.definisciCompito(<u>compito</u>:Compito,<u>tempoStimato</u>: numero, <u>quantità</u>: numero, <u>porzioni</u>: numero)

Precondizioni:

- È in corso la gestione dei compiti relativi a un Foglio Riepilogativo foglio
- <u>compito</u> è contenuto in *foglio*

Post-condizioni:

- compito.tempoStimato = tempoStimato
- compito.quantità = quantità
- compito.porzioni = porzioni

5a.1 marcaPreparazioneComePronta(compito: Compito)

Precondizioni:

- È in corso la gestione dei compiti relativi a un Foglio Riepilogativo foglio
- compito è contenuto in foglio
- compito non è marcato come pronto

Postcondizioni:

- <u>compito.completato</u> = si
- compito non è più associato a un Turno o a un Cuoco

5b.1 indicaSaturazioneCucina(turno: Turno)

Precondizioni:

- È in corso la gestione dei compiti relativi a un Foglio Riepilogativo foglio
- turno non è marcato come saturo

Postcondizioni:

• turno è marcato come saturo

6. assegnaCompito(compito: Compito, turno: Turno, cuoco?: Cuoco)

Precondizioni:

• compito è contenuto in foglio

Postcondizioni:

Se compito.tempoRichiesto è minore o uguale a turno.durata.

- viene creata un'istanza ass di Assegnamento
- <u>compito</u> si svolgerà durante <u>turno</u>
- [Se <u>cuoco</u> è specificato ed è disponibile in <u>turno</u>] <u>compito</u> è stato assegnato a <u>cuoco</u>

6a.1 modificaAssegnazione(ass: Assegnamento, turno: Turno, cuoco?: Cuoco)

Precondizioni:

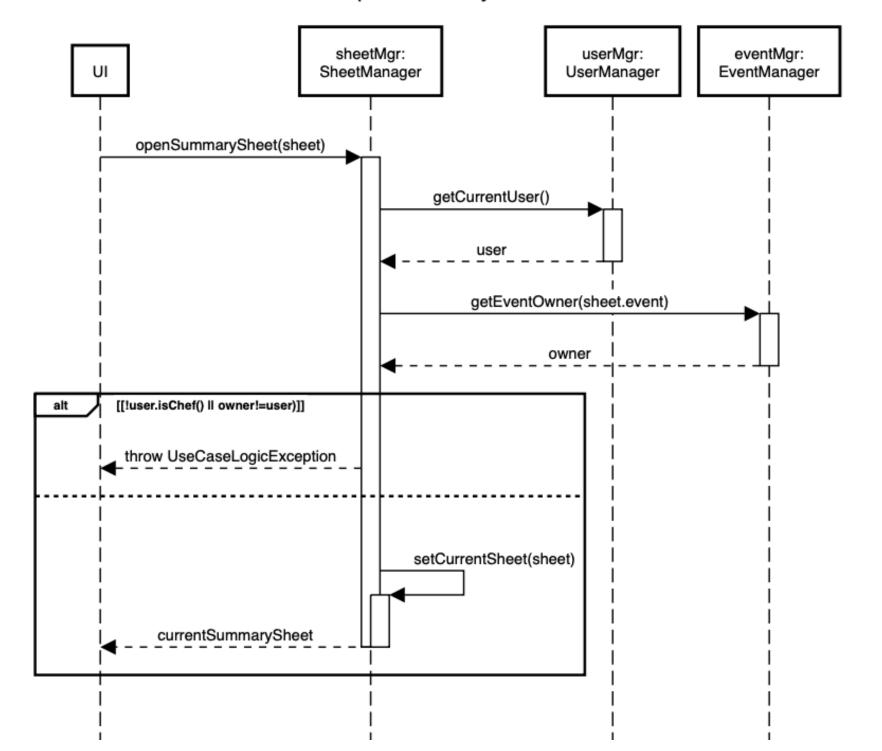
• ass è associato ad un compito presente in foglio

Postcondizioni:

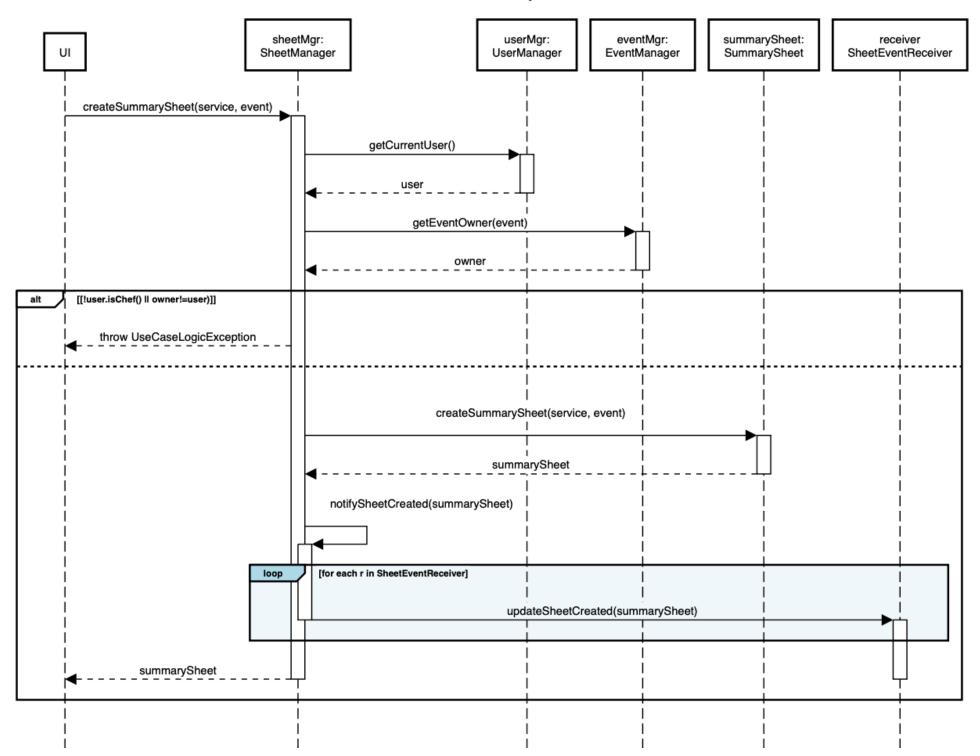
Se compito.tempoRichiesto è minore o uguale a turno.durata.

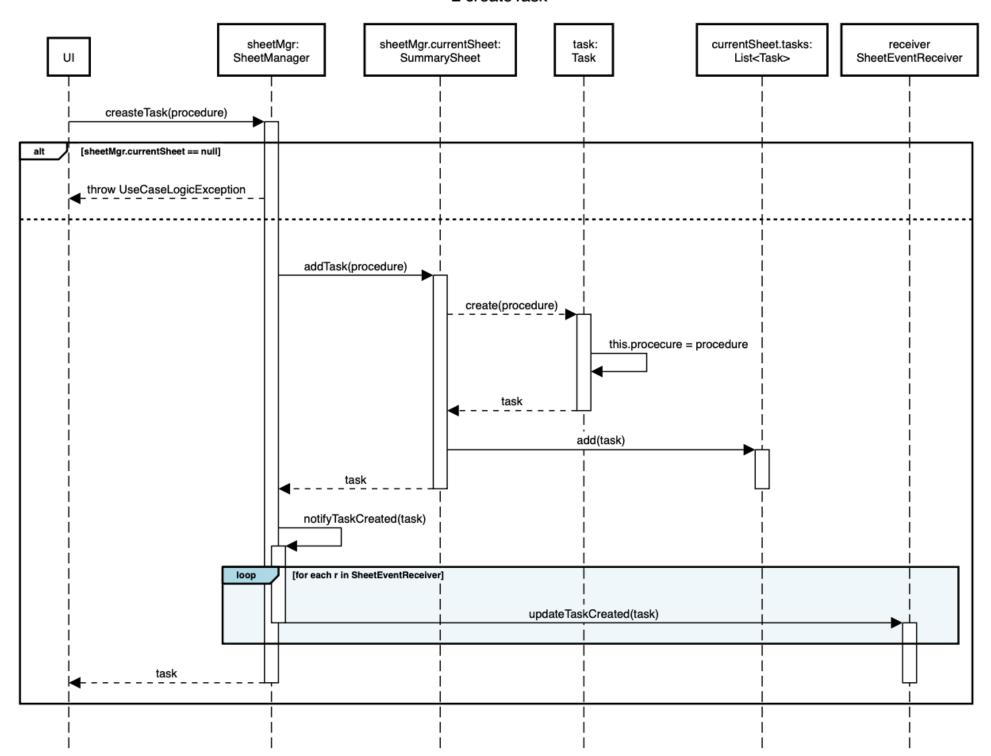
- ass viene modificato
- <u>compito</u> si svolgerà durante <u>turno</u>
- [Se <u>cuoco</u> è specificato ed è disponibile in <u>turno</u>] <u>compito</u> è assegnato a <u>cuoco</u>

1a openSummarySheet

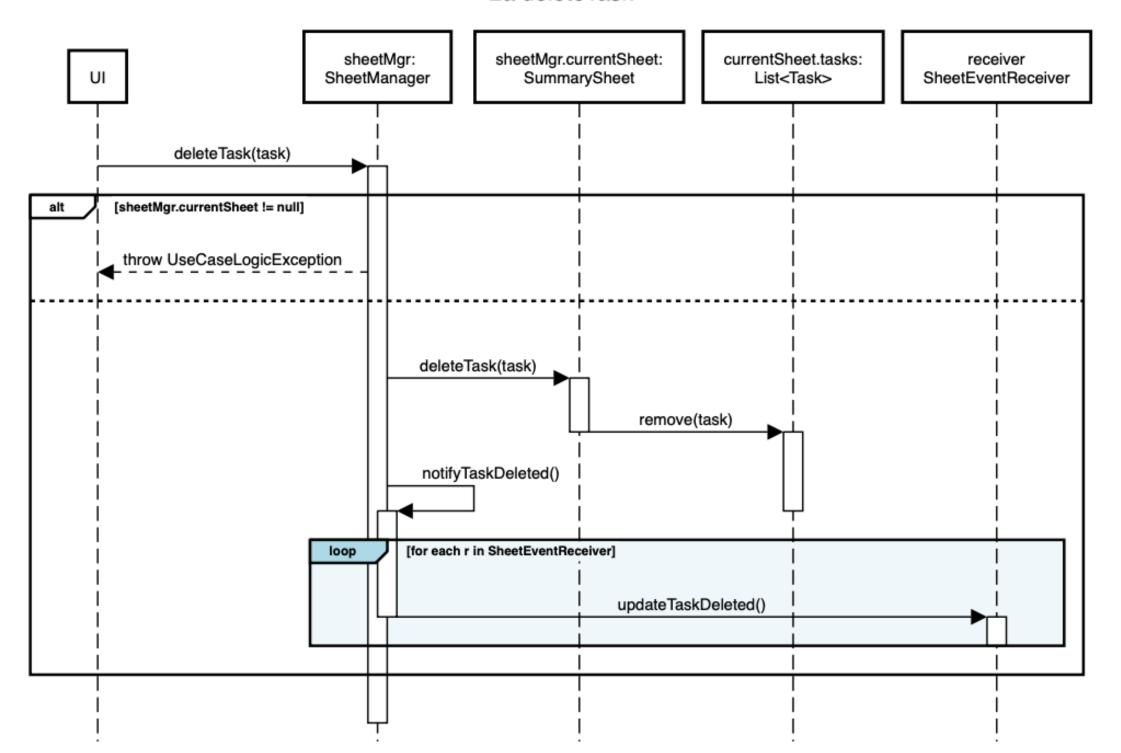


1 createSummarySheet

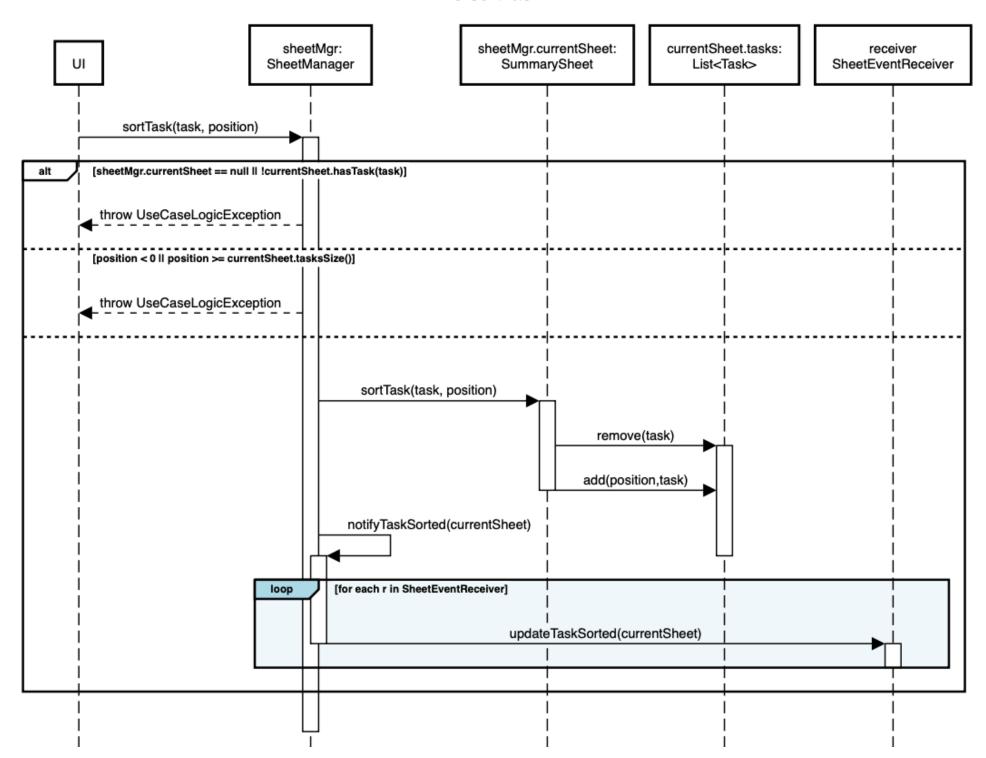




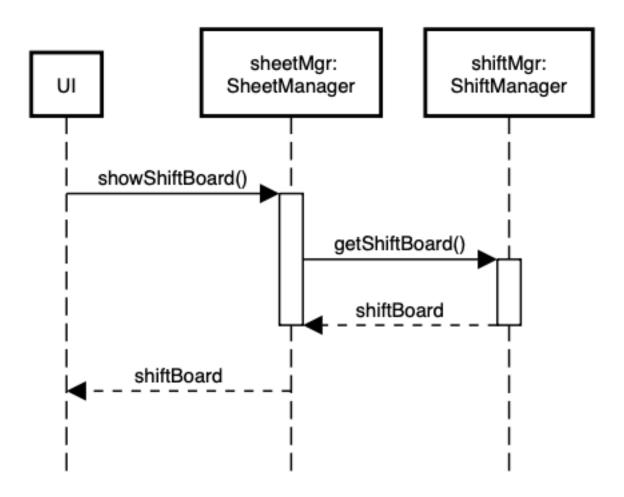
2a deleteTask

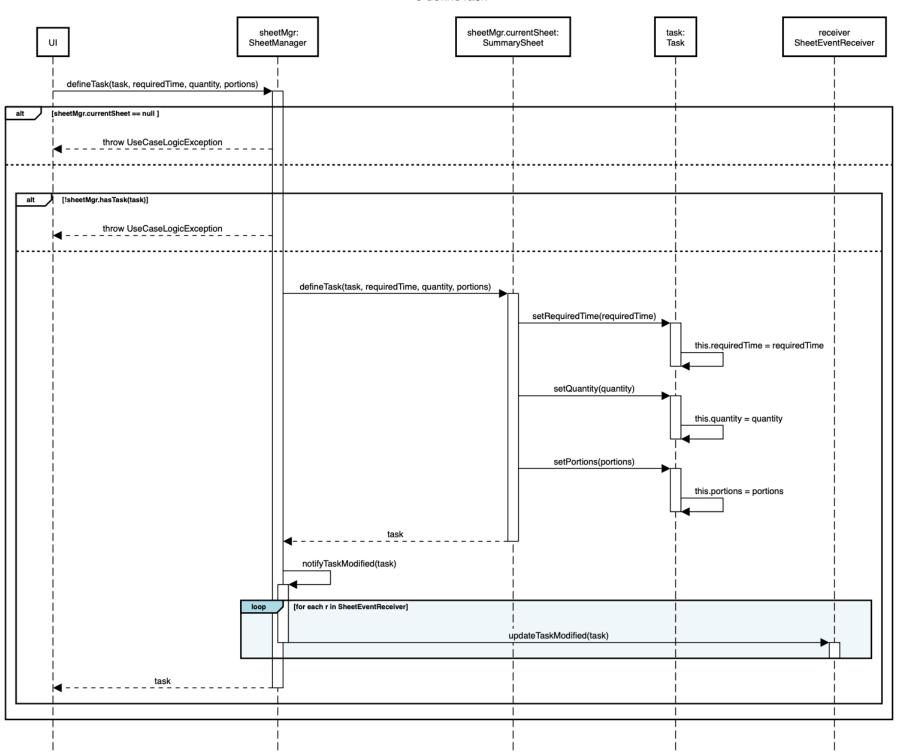


3 sortTask

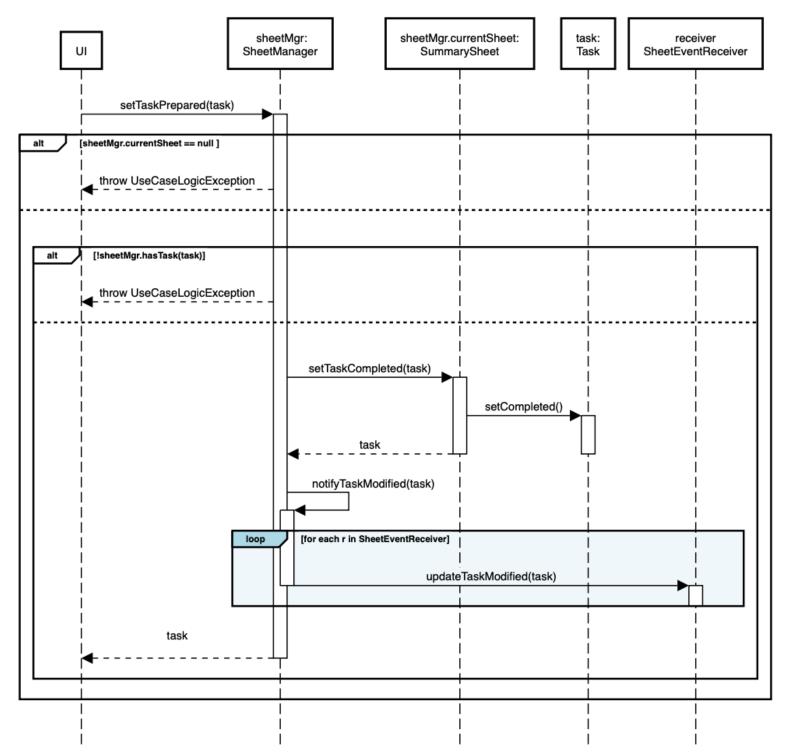


4 showShiftBoard

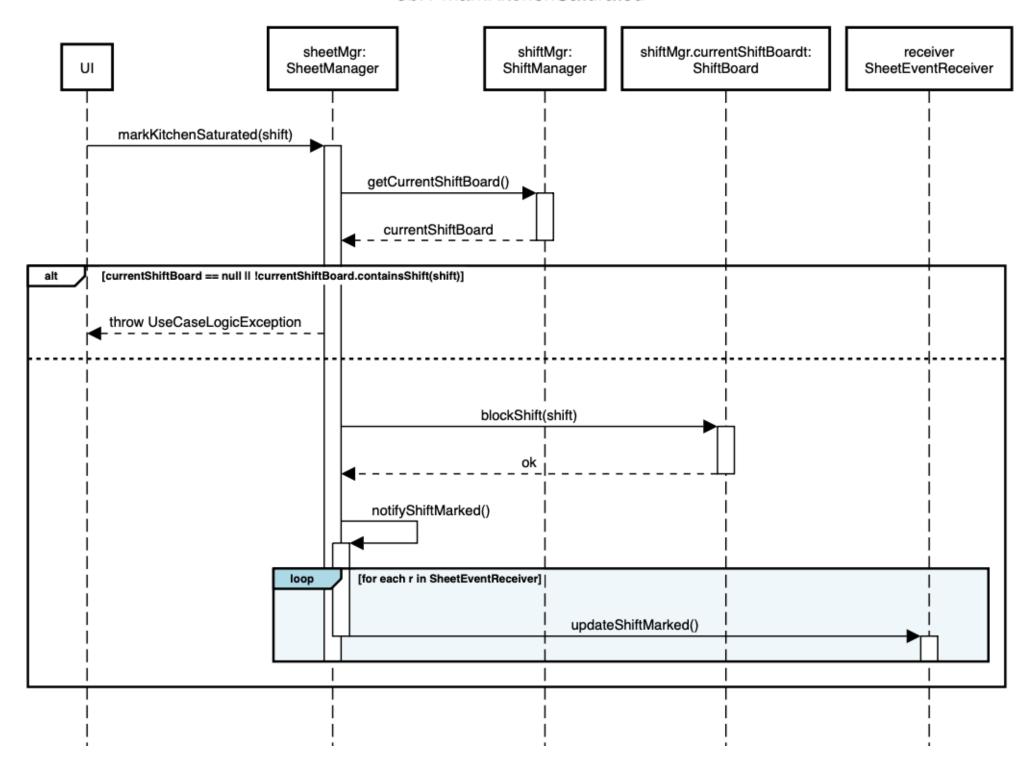




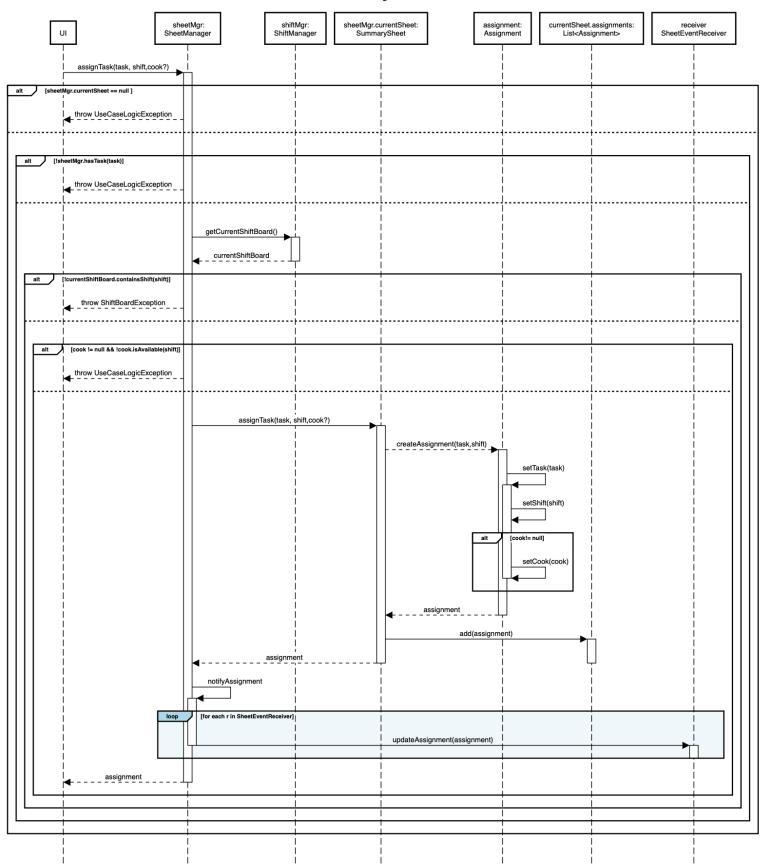
5a.1 setTaskPrepared



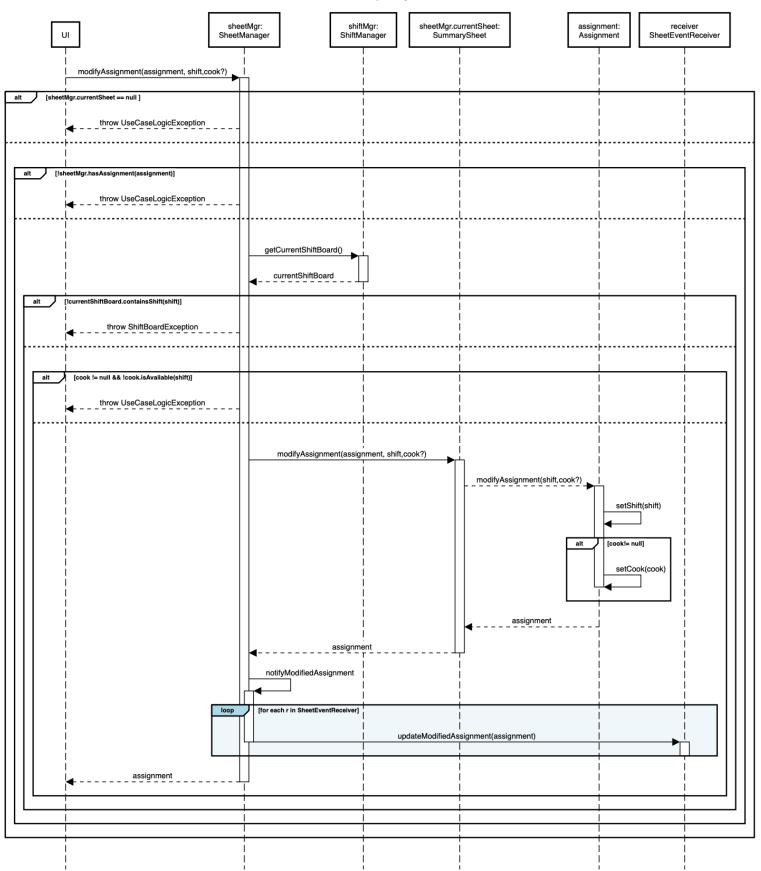
5b.1 markKitchenSaturated



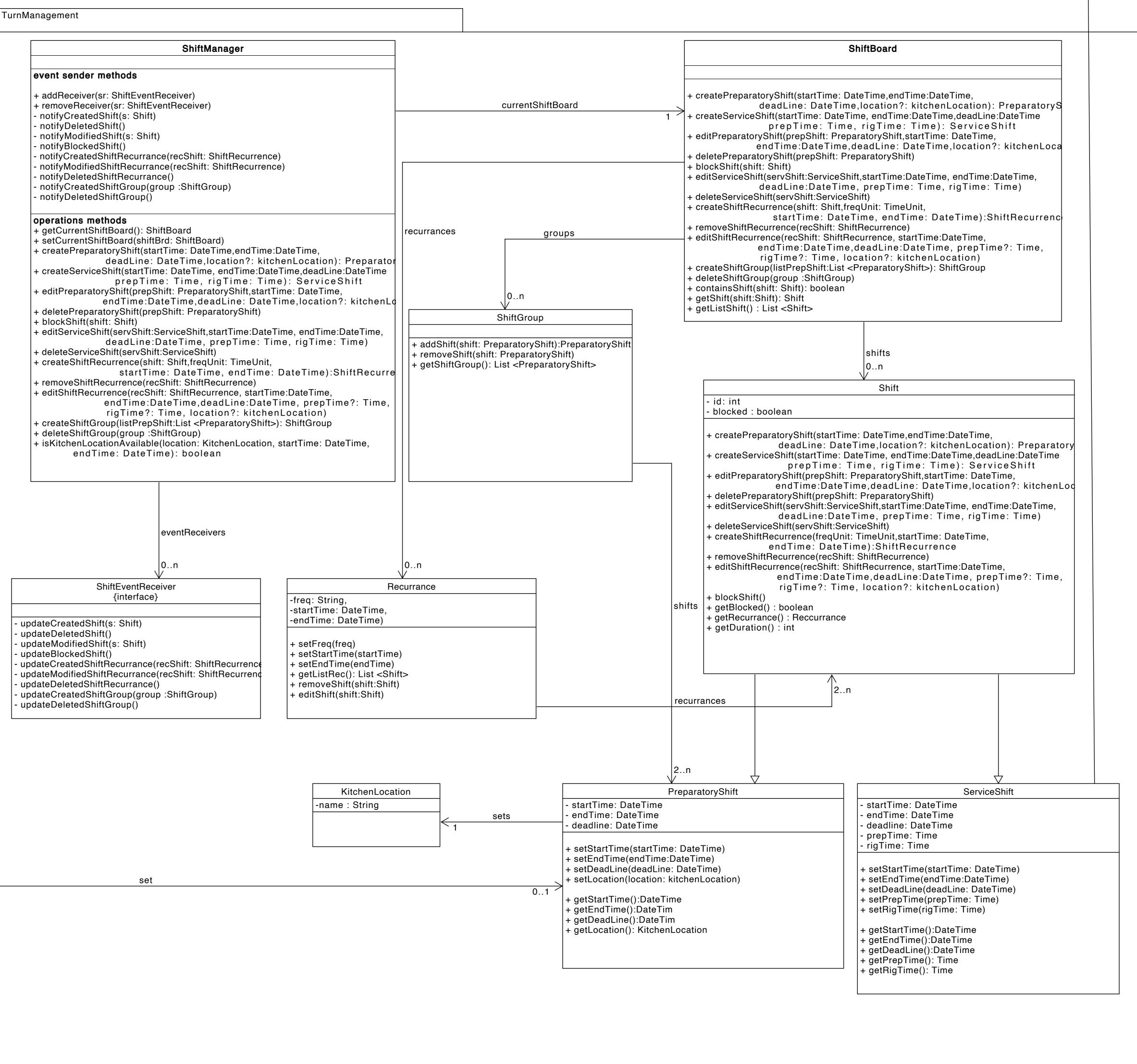
6 assignTask



6a.1 modifyAssignment



Design Class Diagram (DCD) MenuManagement General module UserManagement MenuManager UseCaseLogicException Exception Menu -menuFeatures: String[] -title: String event sender methods -published: boolean UserManager +addReceiver(er: MenuEventReceiver) -inUse: boolean +removeReceiver(er: MenuEventReceiver) -features: String[] -notifySectionAdded(sec: Section) event sender methods -featureValues: boolean[] User -notifyMenuItemAdded(mi: MenuItem) +create(owner: User, title: String, features: String) -notifvMenuCreated(m: Menu) +addSection(name: String) currentUser -notifyMenuDeleted(m: Menu) +isChef(): boolean +addItem(r: Recipe, sec?: Section, operations methods currentMenu -notifySectionsRearranged(m: Menu) +isOrganizer(): boolean +getCurrentUser(): User +isCook(): boolean +isOwner(user: User): boolean +isServiceStaff(): boolean operations methods +isInUse(): boolean +defineSection(name: String) +hasSection(sec: Section): boolean +inserItem(r: Recipe, sec?: Section, desc?: String) +sectionsSize(): int +createMenu(title: String): Menu | sections | +moveSection(sec: Section, pos: int) +getRecipeBook(): ArrayList<Recipe> eventReceivers Section ServiceStuff Chef Cook Organizer name: String MenuEventReceiver +create(name: String) {interface} +isUserAvailable(shift:Shift) +create(sec: Section) : boolean +addItem(mi: MenuItem) Menultem +updateSectionAdded(m: Menu. sec: Section) +updateMenuItemAdded(m: Menu, mi: MenuItem) description: String +updateMenuCreated(m: Menu) -create(rec: Recipe, desc?: String) rupdateMenuDeleted(m: Menu) sectionItems +create(mi: Menultem) menu +updateSectionsRearranged(m: Menu) refers SummarySheet Management EventManagement SheetManager SummarySheet EventManager Event Service -startDate: Date -description: String -id: int -startDate: Date -endDate: Date event sender methods getCurrentEvent() rrentEvent -endDate: Date getEventOwner(event: Event): Ch -notifySheetCreated(sheet: SummarySheet) + addTask(procedure: Procedure) services -numPerson: int -notifySheetDeleted(sheet: SummarySheet) + removeTask(task: Task) -info: String -notifyTaskCreated(task: Task) + sortTask(task: Task, position:int) +getOwner(): Chef notifyTaskDeleted() + modifyTask(task:Task,requiredTime?: int, -notifyTaskModified(task: Task) quantity?: int, portions?: int) -notifyTaskSorted(sheet: SummarySheet) + setTaskCompleted(task: Task) + hasTask(task: Task) : boolean +addReceiver(sr: SheetEventReceiver) +removeReceiver(sr: SheetEventReceiver) + tasksSize(): int operations methods +createSummarySheet(service:Service, event:Event): SummarySheet +setCurrentSheet(sheet: SummarySheet) +openSummarySheet():SummarySheet +deleteSheet(sheet: SummarySheet) +createTask(procedure: Procedure): Task +deleteTask(task: Task) +sortTask(task:Task, position: int) +assignTask(task:Task, cook: Cook, shift:PreparatoryShift, requiredTime: int, quantity: int, portions: int) +modifyTask(task:Task, cook: Cook shift:PreparatoryShift, requiredTime: int, Task quantity: int, portions: int) -quantity: int +setTaskPrepared(task: Task) -portion: int +showShiftBoard() -requiredTime: Time + hasTask(task: Task) : boolean RecipeBookManagement -completed: boolean + setCompleted() + setQuantity(quantity) refers + setPortion(portion) + setRequiredTime(requiredTime) eventReceivers KitchenProcedure RecipeBook SheetEventReceiver -name: String {interface} -itemBooks: List < ItemBook> +updateSheetCreated(sheet: SummarySheet) book +updateSheetDeleted(sheet: SummarySheet) +updateItemAdded(sheet: SummarySheet, item:ItemBook) +updateAssignmentCreated(sheet: SummarySheet,asg:Assignment) +updateAssignmentPortion(asg:Assignment, portion: int) +updateAssignmentTime(sheet: SummarySheet, time: Time) +updateSummarySheetNotes(note: String) +updateCookInAssignment(asg:Assignment, cook: Cook) +updateTurnInAssignment(asg:Assignment, turn: Turn) +updateItemModified(task: Recipe, asg: Assignment) +updateAssignmentDeleted(asg: Assignment) cook +updateTaskSorted(sheet: SummarySheet) Assignment setTask(task: Task) getTask(): Task RecipeBookManager setPreparatoryShift(prepShift: PreparatoryShift) getPreparatoryShift(): PreparatoryShift Recipe Preparation setCook(cook: Cook) event sender methods getCook(): Cook -name: String -name: String preparations operations methods +getRecipes(): ArrayList<ItemBook>



work