

# **Laboratorio Sviluppo Applicazioni Software A.A. 2020/2021**

**Caso d'uso: Gestire i turni**

**Autori:**

Leonardo Magliolo, **Matricola:** 844910

Nicola Pallavidino, **Matricola:** 805732

## Informazioni generali

**Nome caso d'uso:** Gestire i turni

**Portata:** Sistema

**Livello:** Obiettivo utente

**Attore primario:** Organizzatore

**Parti Interessate:** Chef, Cuoco

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

**Garanzie di successo o post-condizioni:**

- I turni di servizio sono assegnati
- I turni di cucina sono stati creati

## Scenario Principale

	Attore	Sistema
1	Sceglie la cucina su cui vuole lavorare	Registra selezione
2	Opzionalmente crea i turni di cucina ricorrentemente, specificando: <ul style="list-style-type: none"><li>• Orario di inizio</li><li>• Orario di fine</li><li>• Giorno della settimana di inizio</li><li>• Giorno della settimana di fine</li><li>• Data di inizio</li><li>• Data di fine</li><li>• Opzionalmente data di scadenza di disponibilità</li></ul>	Registra ricorrenze
3	Opzionalmente raggruppa dei turni di cucina cumulativamente, specificando: <ul style="list-style-type: none"><li>• Dalle ore di inizio</li></ul>	Registra raggruppamenti

	<ul style="list-style-type: none"> <li>• Alle ore di fine</li> <li>• Dal giorno della settimana di inizio</li> <li>• Al giorno della settimana di fine</li> <li>• Dalla data di inizio</li> <li>• Alla data di fine</li> </ul>	
	<i>Nota: se vuole lavorare su più cucine dal passo 1</i>	

## Estensioni ed Eccezioni

### Estensione 2a

#	Attore	Sistema
2a.1	<p>Crea i turni di cucina uno a uno, specificando:</p> <ul style="list-style-type: none"><li>• Orario di inizio</li><li>• Orario di fine</li><li>• Data del turno</li><li>• Opzionalmente data di scadenza aggiunta disponibilità</li></ul>	Registra creazioni

### Eccezione 2a.1a

#	Attore	Sistema
Eccezione 2a.1a.1	<p>Crea i turni di cucina uno a uno, specificando:</p> <ul style="list-style-type: none"><li>• Orario di inizio</li><li>• Orario di fine</li><li>• Data del turno</li></ul> <p>Opzionalmente data di scadenza aggiunta disponibilità</p>	Almeno un turno di cucina indicato si sovrappone temporalmente con un altro turno dichiarato precedentemente per la stessa cucina
	<i>Termina il caso d'uso</i>	

### Estensione 2b

#	Attore	Sistema
2b.1	<p>Elimina i turni della cucina esplicitamente, specificando (si/no) se si vuole eliminare eventualmente tutti i turni appartenenti alla sua ricorrenza</p>	Registra eliminazioni

### Eccezione 2b.1a

#	Attore	Sistema
Eccezione 2b.1a.1	Elimina i turni della cucina esplicitamente, specificando (si/no) se si vuole eliminare eventualmente tutti i turni appartenenti alla sua ricorrenza	Almeno un turno indicato presenta almeno una disponibilità data da un membro del personale autorizzato
	<i>Termina il caso d'uso</i>	

### Estensione 2c

#	Attore	Sistema
2c.1	<p>Modifica i turni di cucina esplicitamente, specificando eventualmente:</p> <ul style="list-style-type: none"><li>• Orario di inizio</li><li>• Orario di fine</li><li>• Data del turno</li><li>• Data di scadenza disponibilità</li><li>• (Obbligatoriamente) Valore che indica se la data di scadenza non inserita (che rappresenta la scadenza nulla) è da considerarsi nelle modifiche o meno (si/no)</li></ul>	Registra modifiche

### Eccezione 2c.1a

#	Attore	Sistema
Eccezione 2c.1a.1	<p>Modifica i turni di cucina esplicitamente, specificando eventualmente:</p> <ul style="list-style-type: none"><li>• Orario di inizio</li><li>• Orario di fine</li><li>• Data del turno</li><li>• Data di scadenza disponibilità</li></ul> <p>(Obbligatoriamente) Valore che indica se la data di scadenza non inserita (che rappresenta la scadenza nulla) è da considerarsi nelle modifiche o meno (si/no)</p>	Almeno un turno di cucina modificato si sovrappone temporalmente con un altro turno dichiarato precedentemente per la stessa cucina
	<i>Termina il caso d'uso</i>	

### Eccezione 2c.1b

#	Attore	Sistema
Eccezione 2c.1b.1	Elimina i turni della cucina esplicitamente, specificando (si/no) se si vuole eliminare eventualmente tutti i turni appartenenti alla sua ricorrenza	Almeno un turno indicato presenta almeno una disponibilità data da un membro del personale autorizzato
	<i>Termina il caso d'uso</i>	

### Estensione 3a

#	Attore	Sistema
3a.1	Raggruppa dei turni della cucina esplicitamente	Registra raggruppamenti

**Eccezione 3a.1a**

#	Attore	Sistema
Eccezione 3a.1a.1	Raggruppa dei turni della cucina esplicitamente	Almeno uno dei turni indicati risulta già appartenente ad un altro gruppo
	<i>Termina il caso d'uso</i>	

**Estensione 3b**

#	Attore	Sistema
3b.1	Elimina raggruppamenti dei turni della cucina esplicitamente, specificando (si/no) se si vuole eliminare eventualmente anche il gruppo di cui fa parte	Registra eliminazioni

**Estensione (1-3)a**

#	Attore	Sistema
(1-3)a.1	Sceglie un evento su cui lavorare	Registra selezione
(1-3)a.2	<p>Opzionalmente genera automaticamente i turni associati all'evento ricorrente selezionato rispetto i suoi servizi, specificando:</p> <ul style="list-style-type: none"> <li>• Minuti in anticipo rispetto all'inizio dell'evento</li> <li>• Minuti in ritardo rispetto alla fine dell'evento</li> <li>• Opzionalmente data di</li> </ul>	Registra assegnazioni

	scadenza aggiunta disponibilità fine dell'evento	
	<i>Se vuole lavorare su più eventi ripete dal passo (1-3)a.1</i>	
(1-3)a.3	Opzionalmente sceglie un servizio su cui vuole lavorare	Registra selezione
(1-3)a.4	Opzionalmente genera un turno per il servizio precedentemente selezionato, specificando: <ul style="list-style-type: none"> <li>• Minuti in anticipo rispetto all'inizio dell'evento</li> <li>• Minuti in ritardo rispetto alla fine dell'evento</li> <li>• Opzionalmente data di scadenza aggiunta disponibilità fine dell'evento</li> </ul>	Registra creazione
(1-3)a.5	Opzionalmente assegna le mansioni di servizio rispetto al servizio precedentemente selezionato specificando il personale di servizio associato	Registra assegnazioni
	<i>Se vuole lavorare su più servizi ripete il passo (1-3)a.3</i>	
(1-3)a.6	Opzionalmente raggruppa dei turni di servizio cumulativamente, specificando: <ul style="list-style-type: none"> <li>• Orario di inizio</li> <li>• Orario di fine</li> <li>• Giorno della settimana di inizio</li> </ul>	Registra raggruppamenti



	<ul style="list-style-type: none"> <li>• Giorno della settimana di fine</li> <li>• Data di inizio</li> <li>• Data di fine</li> </ul>	
--	--	--

### Estensione (1-3)a.2a

#	Attore	Sistema
(1-3)a.2a.1	Elimina tutti i turni associati all'evento ricorrente	Registra eliminazioni

### Estensione (1-3)a.4a

#	Attore	Sistema
(1-3)a.4a.1	Elimina il turno associato ad un servizio	Registra eliminazioni

### Eccezione (1-3)a.4a.1a

#	Attore	Sistema
Eccezione (1-3)a.4a.1a.1	Elimina il turno associato ad un servizio	Il turno indicato presenta almeno una disponibilità data da un membro del personale autorizzato
	<i>Termina il caso d'uso</i>	

### Eccezione (1-3)a.4b.1

#	Attore	Sistema
Eccezione (1-3)a.4b.1	Genera un turno per il servizio precedentemente selezionato, specificando: <ul style="list-style-type: none"> <li>• Minuti in anticipo rispetto all'inizio dell'evento</li> <li>• Minuti in ritardo</li> </ul>	Esiste già un turno di servizio associato al servizio indicato precedentemente

	rispetto alla fine dell'evento <ul style="list-style-type: none"> <li>Opzionalmente data di scadenza aggiunta disponibilità fine dell'evento</li> </ul>	
	<i>Termina il caso d'uso</i>	

### Estensione (1-3)a.5a

#	Attore	Sistema
(1-3)a.5a.1	Elimina le mansioni di servizio	Registra eliminazioni

### Eccezione (1-3)a.5b

#	Attore	Sistema
Eccezione (1-3)a.5b.1	Assegna le mansioni di servizio rispetto al servizio precedentemente selezionato specificando il personale di servizio associato (o cuoco)	Il personale indicato non ha dato la disponibilità per il turno associato al servizio precedentemente selezionato
	<i>Termina il caso d'uso</i>	

### Estensione (1-3)a.6a

#	Attore	Sistema
(1-3)a.6a.1	Raggruppa dei turni di servizio esplicitamente	Registra raggruppamenti

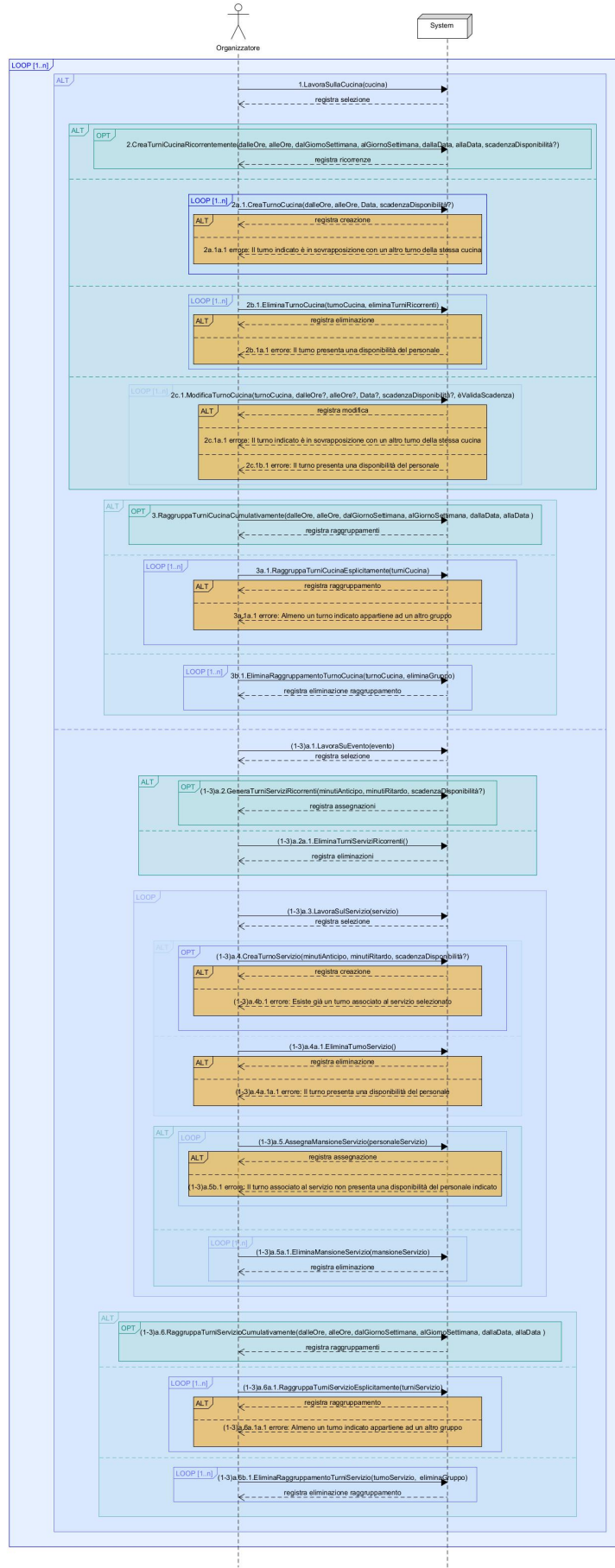
### Eccezione (1-3)a.6a.1a

#	Attore	Sistema
Eccezione (1-3)a.6a.1a.1	Raggruppa dei turni di servizio esplicitamente	Almeno uno dei turni indicati risulta già appartenente ad un altro gruppo
	<i>Termina il caso d'uso</i>	

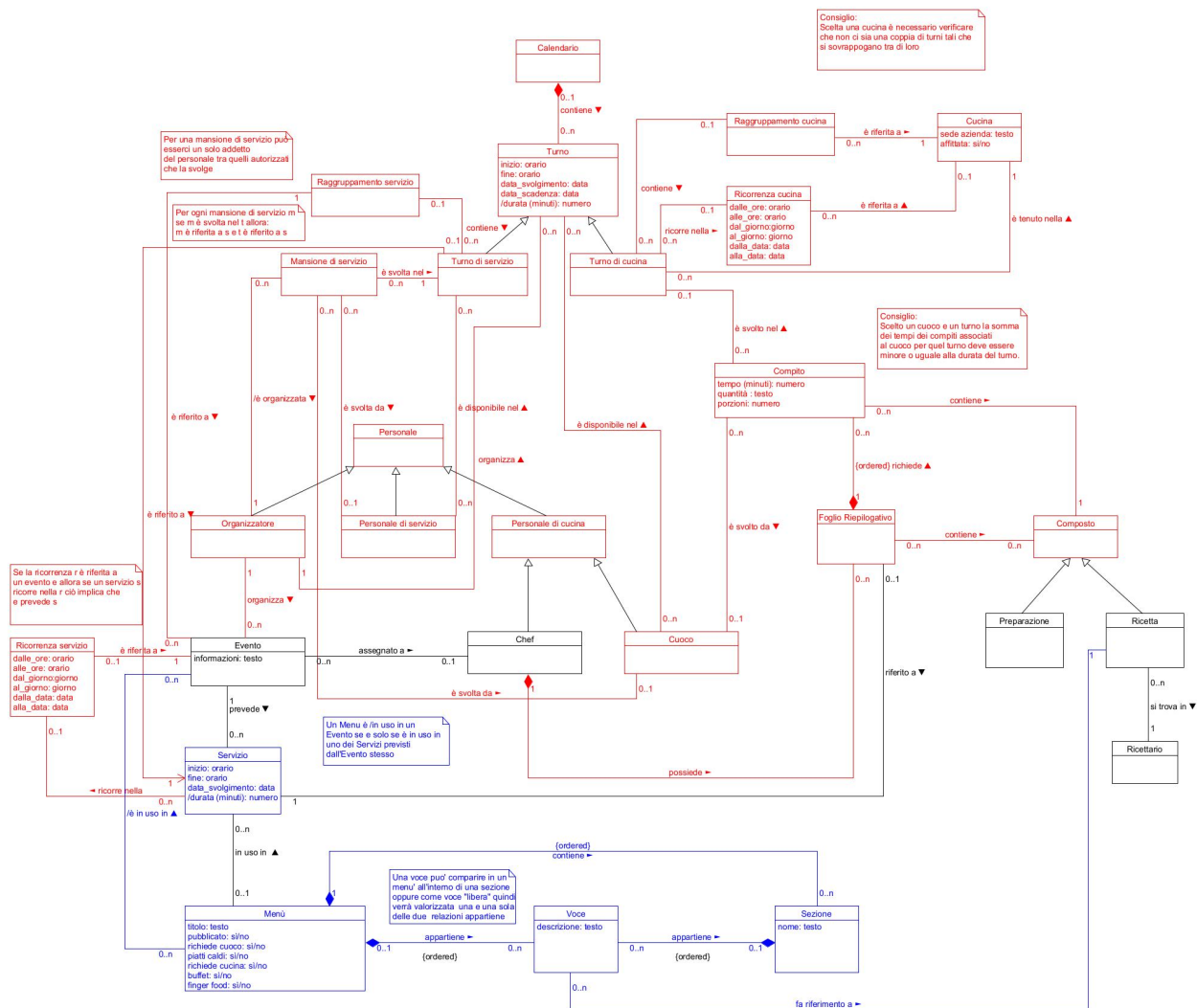
**Estensione (1-3)a.6b**

#	Attore	Sistema
(1-3)a.6b.1	Elimina raggruppamenti dei turni di servizio esplicitamente, specificando (si/no) se si vuole eliminare eventualmente anche il gruppo di cui fa parte	Registra eliminazione

Diagrammi di sequenza di sistema  
per l'UC Gestire i Turni  
Scenario principale, estensioni ed eccezioni



# Modello di dominio



# Contratti

## Pre-condizioni generali:

- L'attore è identificato con un'istanza di *org* di Organizzatore
- E' stata resa nota l'istanza *calendario* di Calendario

## 1 LavoraSullaCucina(cucina: Cucina)

Pre-condizioni: ---

Post-condizioni: ---

## 2 CreaTurniCucinaRicorrentemente(dalleOre: Orario, alleOre: Orario, dalGiornoSettimana: Giorno, alGiornoSettimana: Giorno, dallaData: Data, allaData: Data, scadenzaDisponibilità?: Data)

Pre-condizioni:

- Si sta lavorando sulla *cucina*
- alleOre > dalleOre
- alGiornoDellaSettimana >= dalGiornoDellaSettimana
- dallaData <= allaData

Post-condizioni:

- E' stata creata un'istanza di Ricorrenza cucina *r* tale che:
  - r.dalle\_ore = dalleOre
  - r.alle\_ore = alleOre
  - r.dal\_giorno = dalGiorno
  - r.al\_giorno = alGiorno
  - r.dalla\_data = dallaData
  - r.alla\_data = allaData
  - *r* **referita a** *cucina*
- Per ogni Data *d* tale che:
  - dallaData <= *d* <= allaData
  - dalGiornoSettimana <= *g* <= alGiornoSettimana  
(Dove il giorno della data *d* coincide con *g*)
  - Non esiste un **Turno di cucina** *t* tale che:
    - *t* **è tenuto nella** *cucina*
    - *t.data\_svolgimento* = *d*

- $\underline{\text{dalleOre}} \leq t.\text{inizio} \leq \underline{\text{alleOre}}$  oppure  $\underline{\text{dalleOre}} \leq t.\text{fine} \leq \underline{\text{alleOre}}$

E' stato creato un Turno di cucina  $t$  tale che:

- $t.\text{data\_svolgimento} = d$
- $t.\text{inizio} = \underline{\text{dalleOre}}$
- $t.\text{fine} = \underline{\text{alleOre}}$
- [Se specificata scadenzaDisponibilità]:  
 $t.\text{data\_scadenza} = \underline{\text{scadenzaDisponibilità}}$
- $\text{org organizza } t$
- $t$  è tenuto nella cucina
- calendario contiene  $t$
- $t$  ricorre nella  $r$

## 2a.1 CreaTurnoCucina(dalleOre: Orario, alleOre: Orario, data: Data, scadenzaDisponibilità?: Data)

### Pre-condizioni:

- Si sta lavorando sulla cucina
- $\underline{\text{alleOre}} > \underline{\text{dalleOre}}$
- Non esiste un Turno di cucina  $t$  tale che:
  - $t$  è tenuto nella cucina
  - $t.\text{data\_svolgimento} = \underline{\text{data}}$
  - $\underline{\text{dalleOre}} \leq t.\text{inizio} \leq \underline{\text{alleOre}}$  oppure  $\underline{\text{dalleOre}} \leq t.\text{fine} \leq \underline{\text{alleOre}}$

### Post-condizioni:

- E' stata creata un'istanza  $t$  di Turno di cucina
- $t.\text{data\_svolgimento} = d$
- $t.\text{inizio} = \underline{\text{dalleOre}}$
- $t.\text{fine} = \underline{\text{alleOre}}$
- [Se specificata scadenza]:  $t.\text{data\_scadenza} = \underline{\text{scadenzaDisponibilità}}$
- $\text{org organizza } t$
- $t$  è tenuto nella cucina
- calendario contiene  $t$

## 2b.1.EliminaTurnoCucina(turnoCucina: Turno di cucina, eliminaTurniRicorrenti: si/no)

### Pre-condizioni:

- Si sta lavorando sulla *cucina*
- turnoCucina è **tenuto nella** *cucina*
- *org organizza* turnoCucina
- Non esiste un Cuoco *c* tale che *c* è **disponibile** nel turnoCucina

### Post-condizioni:

[se eliminaTurniRicorrenti = si]:

[se esiste una Ricorrenza di cucina *r* tale che turnoCucina **ricorre nella** *r*]:

- Per ogni Turno di cucina *t* tale che *t* **ricorre nella** *r*:  
[Se non esiste un Cuoco *c* tale che *c* è **disponibile** nel *t*]: *t* è stato eliminato
- *r* è stata eliminata

Altrimenti: turnoCucina è stato eliminato

Altrimenti turnoCucina è stato eliminato

## 2d.1.ModificaTurnoCucina(turnoCucina: Turno di cucina, dalleOre?: Orario, alleOre?: Orario, data?: Data, scadenzaDisponibilità?: Data, èValidaScadenza: si/no)

### Pre-condizioni:

- Si sta lavorando sulla *cucina*
- turnoCucina è **tenuto nella** *cucina*
- *org organizza* turnoCucina
- Non esiste un Cuoco *c* tale che *c* è **disponibile** nel turnoCucina
- Esiste una Data *dataTurnoCucina* tale che turnoCucina è **tenuto nella** *dataTurnoCucina*
- [se specificato alleOre]:
  - [se specificato dalleOre]: alleOre > dalleOre  
altrimenti: alleOre > turnoCucina.inizio
- altrimenti:
- [se specificato dalleOre]: turnoCucina.fine > dalleOre
- Non esiste un Turno di cucina *t* tale che:
  - *t* non è turnoCucina
  - *t* è **tenuto nella** *cucina*
  - [se specificata data]: *t.data\_svolgimento* = data



altrimenti:  $t.data\_svolgimento = dataTurnoCucina$

- [se specificato dalleOre]:
  - [se specificato alleOre]:
    - $\underline{dalleOre} \leq t.inizio \leq \underline{alleOre}$  oppure  $\underline{dalleOre} \leq t.fine \leq \underline{alleOre}$
  - Altrimenti:
    - $\underline{dalleOre} \leq t.inizio \leq \underline{turnoCucina.fine}$  oppure  $\underline{dalleOre} \leq t.fine \leq \underline{turnoCucina.fine}$

*altrimenti:*

- [se specificato alleOre]:
  - $\underline{turnoCucina.inizio} \leq t.inizio \leq \underline{alleOre}$  oppure  $\underline{turnoCucina.inizio} \leq t.fine \leq \underline{alleOre}$
- Altrimenti:
  - $\underline{turnoCucina.inizio} \leq t.inizio \leq \underline{turnoCucina.fine}$  oppure  $\underline{turnoCucina.inizio} \leq t.fine \leq \underline{turnoCucina.fine}$

### Post-condizioni:

- [se specificato dalleOre]:  $\underline{turnoCucina.inizio} = \underline{dalleOre}$
- [se specificato alleOre]:  $\underline{turnoCucina.fine} = \underline{alleOre}$
- [se specificata data]:
  - $t.data\_svolgimento = \underline{data}$
- [se èValidaScadenza = si]
  - [se specificata scadenzaDisponibilità]:
    - $\underline{turnoCucina.data\_scadenza} = \underline{scadenzaDisponibilità}$

*altrimenti:*

$t.data\_svolgimento = *Data\ nulla*$

### 3. RaggruppaTurniCucinaCumulativamente(dalleOre: Orario, alleOre: Orario, dalGiornoSettimana: Giorno, alGiornoSettimana: Giorno, dallaData: Data, allaData: Data)

#### Pre-condizioni:

- Si sta lavorando sulla *cucina*
- $\underline{alleOre} > \underline{dalleOre}$
- $\underline{alGiornoDellaSettimana} \geq \underline{dalGiornoDellaSettimana}$
- $\underline{dallaData} \leq \underline{allaData}$

#### Post-condizioni:

- E' stata creata un'istanza Raggruppamento cucina  $r$  tale che:

*r* è riferito a *cucina*

- Per Turno di cucina *t* tale che:
    - *t* è tenuto nella *cucina*
    - *org organizza t*
    - *t.inizio* >= dalleOre
    - *t.fine* <= alleOre  
dalGiornoSettimana <= *g* <= alGiornoSettimana  
(Dove il giorno della data *t.data\_svolgimento* coincide con *g*)
    - dallaData <= *t.data\_svolgimento* <= allaData
    - Non esiste un Raggruppamento cucina *r* tale che *r* contiene *t*
- allora *r* contiene *t*

### 3a.1.RaggruppaTurniCucinaEsplicitamente(insieme turniCucina di Turni di cucina)

#### Pre-condizioni:

- Si sta lavorando sulla *cucina*
- La cardinalità di turniCucina è maggiore di 1
- Per ogni Turno di cucina *t* appartenente a turniCucina:
  - *t* è tenuto nella *cucina*
  - *org organizza t*
  - Non esiste un Raggruppamento cucina *r* tale che *r* contiene *t*

#### Post-condizioni:

- E' stata creata un'istanza Raggruppamento cucina *r* tale che:
  - Per ogni *t* appartenente a turniCucina: *r* contiene *t*
  - *r* è riferito a *cucina*

### 3b.1.EliminaRaggruppamentoTurnoCucina(turnoCucina: Turno di cucina, eliminaGruppo: si/no)

#### Pre-condizioni:

- Si sta lavorando sulla *cucina*
- *t* è **tenuto nella** *cucina*
- *org organizza t*
- Esiste un Raggruppamento cucina *r* tale che *r* **contiene** turnoCucina

#### Post-condizioni:

[se eliminaGruppo = si]

*r* è stato eliminato

altrimenti: E' stato eliminato il collegamento d'associazione che *r* **contiene** turnoCucina

### (1-3)a.1.LavoraSuEvento(evento: Evento)

Pre-condizioni: *org organizza evento*

Post-condizioni: ---

### (1-3)a.2.GeneraTurniServiziRicorrenti(minutiAnticipo:Numero , minutiRitardo:Numero, scadenzaDisponibilità?:Data)

#### Pre-condizioni:

- Si sta lavorando su *evento*
- Esiste un'istanza di Ricorrenza servizio *r* tale che *r* è **riferito a** *evento*

#### Post-condizioni:

- Per ogni Servizio *s* tale che *s* **ricorre nella** *r*:
  - [Se non esiste un Turno di servizio *t* tale che *t* è **riferito a** *s*]:  
E' stata creata una nuova istanza di Turno di servizio *t* tale che:
    - *t.data\_svolgimento* = *s.data\_svolgimento*
    - *t* è **riferito a** *servizio*  
*t.inizio* = o1  
(Dove o1 è l'orario con minutiAnticipo minuti di anticipo rispetto a *s.inizio*)
    - *t.fine* = o2  
(Dove o2 è l'orario con minutiRitardo minuti di ritardo rispetto a *s.fine*)
    - [Se specificata scadenzaDisponibilità]:

$t.data\_scadenza = \underline{scadenzaDisponibilità}$

- *org organizza t*
- *calendario contiene t*
- *t è riferito a servizio*

### (1-3)a.2a.1.EliminaTurniServiziRicorrenti()

#### Pre-condizioni:

- Si sta lavorando su *evento*
- Esiste un'istanza di Ricorrenza servizio *r* tale che *r* è **riferito a evento**

#### Post-condizioni:

Per ogni Servizio *s* tale che *s* **ricorre nella r**:

Per ogni Turno di servizio *t* tale che *t* è **riferito a r**:

[Se non esiste un'istanza di Personale di servizio *p* tale che *p* è disponibile nel *t* e non esiste un'istanza di Cuoco *c* tale che *c* è disponibile nel *t*]:

- *t* è stato eliminato
- Per ogni istanza di Mansione di servizio *m* tale che *m* è **svolta nel t**: *m* è stata eliminata

### (1-3)a.3.LavoraSulServizio(servizio: Servizio)

Pre-condizioni: *evento prevede servizio*

Post-condizioni: ---

### (1-3)a.4.CreaTurnoServizio(minutiAnticipo:Numero, minutiRitardo:Numero, scadenzaDisponibilità?:Data)

#### Pre-condizioni:

- Si sta lavorando sul *servizio*
- Non esiste un Turno di servizio  $t$  tale che  $t$  è **referito a servizio**

#### Post-condizioni:

- E' stata creata una nuova istanza di Turno di servizio  $t$
- $t$  è **referito a servizio**
- $t.data\_svolgimento = servizio.data\_svolgimento$
- $t.inizio = o1$   
(Dove  $o1$  è l'orario con minutiAnticipo minuti di anticipo rispetto a  $servizio.inizio$ )
- $t.fine = o2$   
(Dove  $o2$  è l'orario con minutiRitardo minuti di ritardo rispetto a  $servizio.fine$ )
- [Se specificata scadenzaDisponibilità]:  
 $t.data\_scadenza = scadenzaDisponibilità$
- *org organizza*  $t$
- *calendario contiene*  $t$

### (1-3)a.4a.1.EliminaTurnoServizio()

#### Pre-condizioni:

- Si sta lavorando sul *servizio*
- Esiste un Turno di servizio  $t$  è referito a *servizio*
- Non esiste un'istanza di Personale di servizio  $p$  tale che  $p$  è disponibile nel  $t$
- Non esiste un'istanza di Cuoco  $c$  tale che  $c$  è disponibile nel  $t$

#### Post-condizioni:

- Per ogni Mansione di servizio  $m$  tale che  $m$  è **svolta nel**  $t$ :  
 $m$  è stata eliminata
- $t$  è stato eliminato

### (1-3)a.5.AssegnaMansioneServizio(personaleServizio: Personale)

#### Pre-condizioni:

- Si sta lavorando sul *servizio*
- personaleServizio è un'istanza di Personale di servizio oppure è un'istanza di Cuoco
- Esiste un Turno di servizio  $t$  tale che  $t$  è **referito a servizio**

#### Post-condizioni:

- E' stata creata una nuova istanza di Mansione di servizio  $m$
- $m$  è svolta da personaleServizio
- $m$  è svolta nel  $t$

### (1-3)a.5a.1.EliminaMansioneServizio(mansioneServizio: Mansione di servizio)

#### Pre-condizioni:

- Si sta lavorando sul *servizio*
- Esiste un Turno di servizio  $t$  tale che mansioneServizio è svolta nel  $t$
- $t$  è **referito a servizio**

Post-condizioni: mansioneServizio è stata eliminata

**(1-3)a.6.RaggruppaTurniServizioCumulativamente(dalleOre: Orario, alleOre: Orario, dalGiornoSettimana: Giorno, alGiornoSettimana: Giorno, dallaData: Data, allaData: Data)**

**Pre-condizioni:**

- Si sta lavorando su *evento*
- alleOre > dalleOre
- alGiornoDellaSettimana >= dalGiornoDellaSettimana
- dallaData <= allaData

**Post-condizioni:**

- E' stata creata un'istanza Raggruppamento di servizio *r* tale che: *r* è **referito a** *evento*
  - Per ogni Turno di servizio *t* tale che:
    - Esiste un Servizio *s* tale che *t* è **referito a** *s*
    - *evento* **prevede** *s*
    - *org* **organizza** *t*
    - *t.inizio* >= dalleOre
    - *t.fine* <= alleOre
    - dalGiornoSettimana <= *g* <= alGiornoSettimana  
(Dove il giorno della data *t.data\_svolgimento* coincide con *g*)
    - dallaData <= *t.data\_svolgimento* <= allaData
    - Non esiste un Raggruppamento di servizio *r* tale che *r* **contiene** *s*
- allora: *r* **contiene** *t*

### **(1-3)a.6a.1.RaggruppaTurniServizioEsplicitamente(insieme turniServizio di Turni di servizio)**

#### **Pre-condizioni:**

- Si sta lavorando su *evento*
- La cardinalità di turniServizio è maggiore di 1
- Per ogni Turno di servizio  $t$  appartenente a turniServizio:
  - Esiste un Servizio  $s$  tale che  $t$  è **riferito a**  $s$
  - *evento* **prevede**  $s$
  - *org* **organizza**  $t$
  - Non esiste un Raggruppamento di servizio  $r$  tale che  $r$  **contiene**  $t$  e  $r$  **riferito a** *evento*

#### **Post-condizioni:**

- E' stata creata un'istanza Raggruppamento di servizio  $r$  tale che:
  - Per ogni  $t$  appartenente a turniServizio:  $r$  **contiene**  $t$
  - $r$  è **riferito a** *evento*

### **(1\_3)a.6b.1.EliminaRaggruppamentoTurnoServizio(turnoServizio: Turno di Servizio, eliminaGruppo: si/no)**

#### **Pre-condizioni:**

- Si sta lavorando su *evento*
- *org* **organizza**  $t$
- Esiste un Raggruppamento di servizio  $r$  tale che  $r$  **contiene** turnoServizio e  $r$  **riferito a** *evento*

#### **Post-condizioni:**

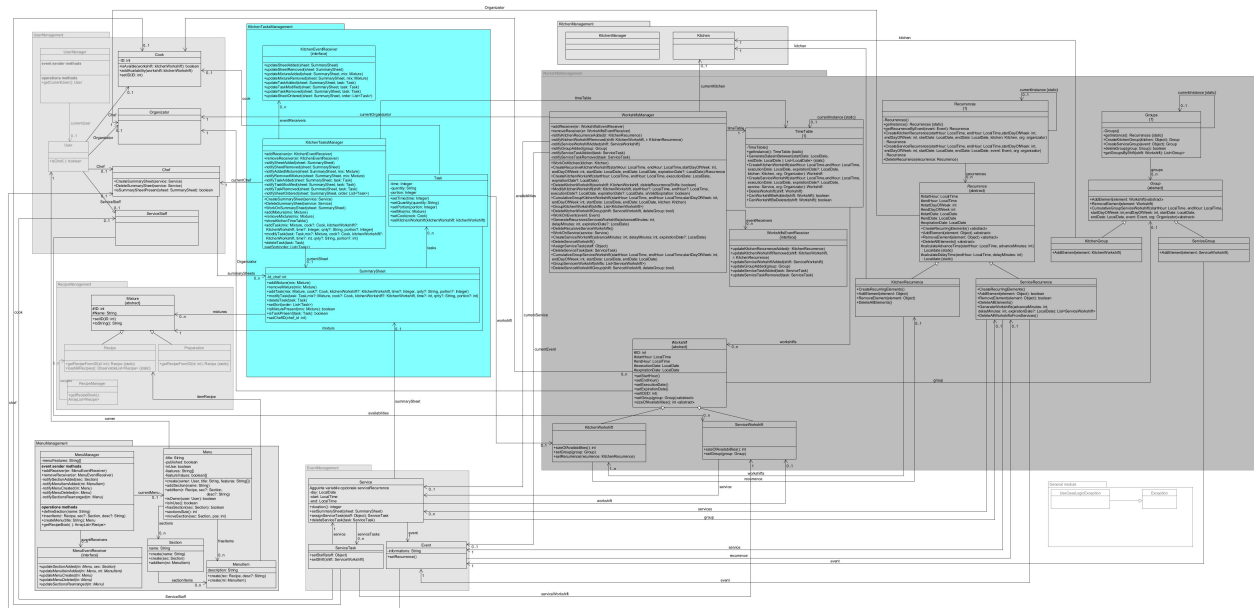
[se eliminaGruppo = si]

$r$  è stato eliminato

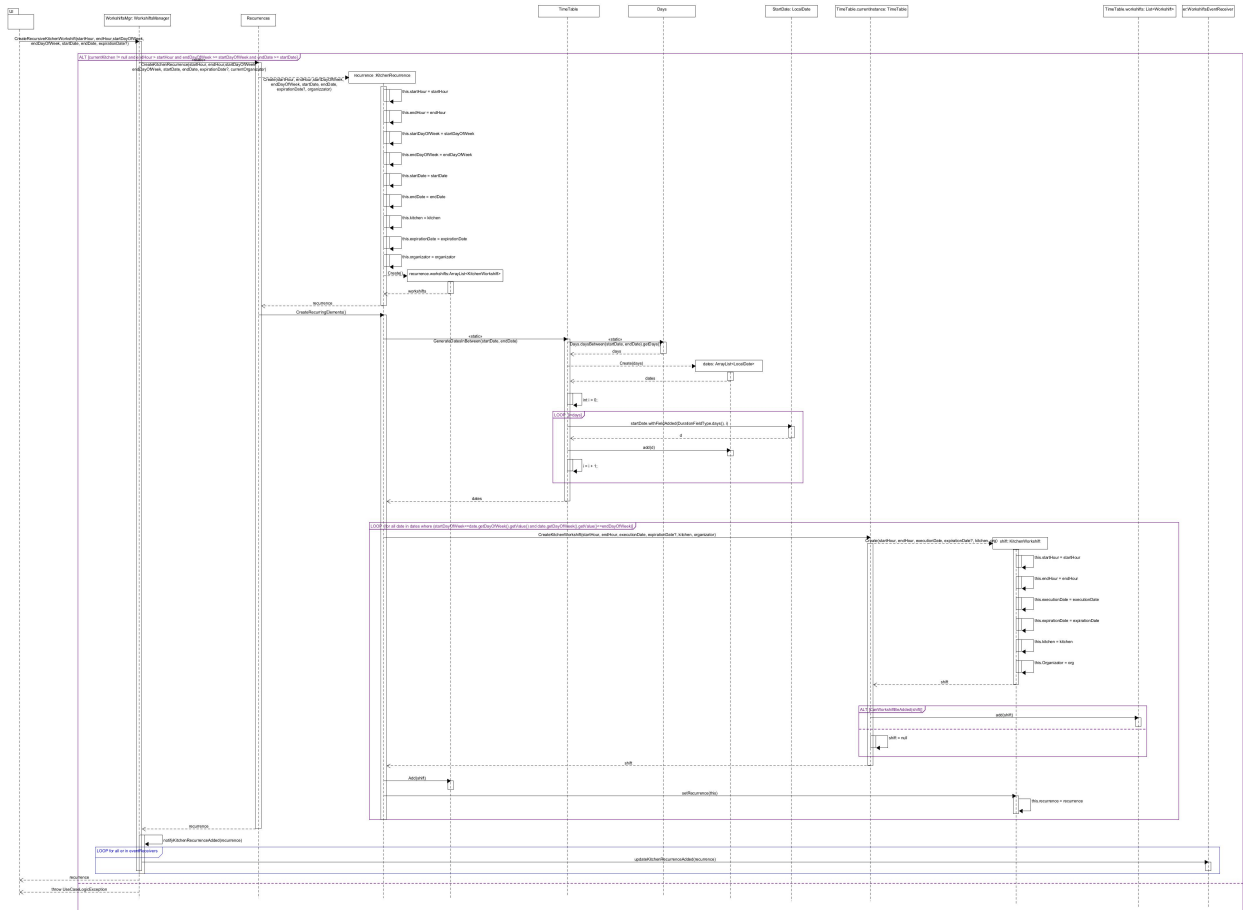
altrimenti: E' stato eliminato il collegamento d'associazione che  
 $r$  **contiene** turnoServizio



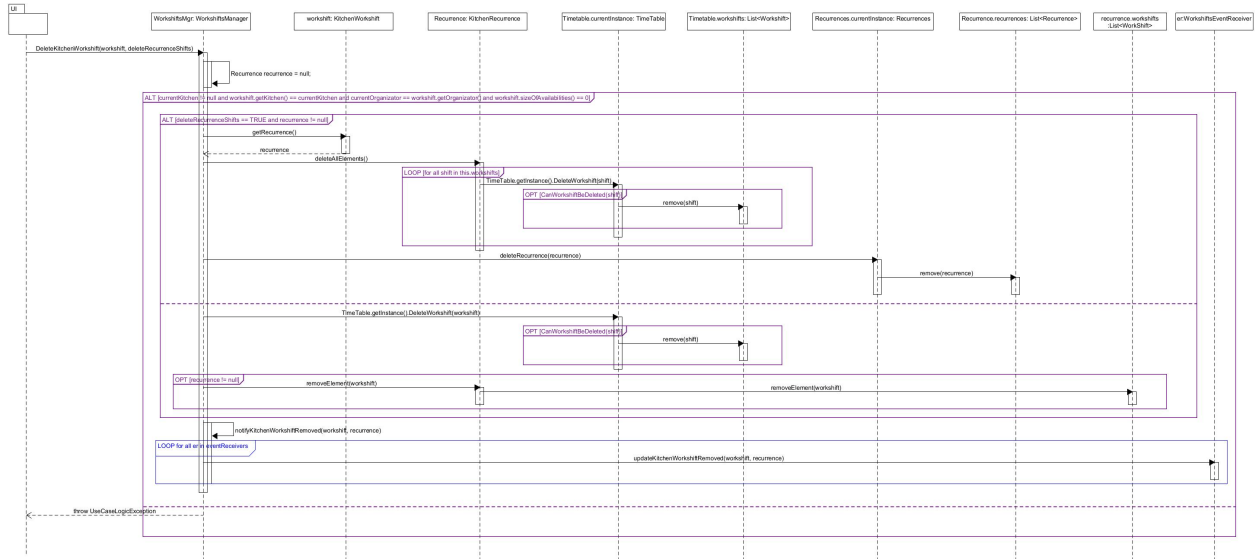
## Detailed Class Diagram



## 2 - CreateKitchenRecurrence

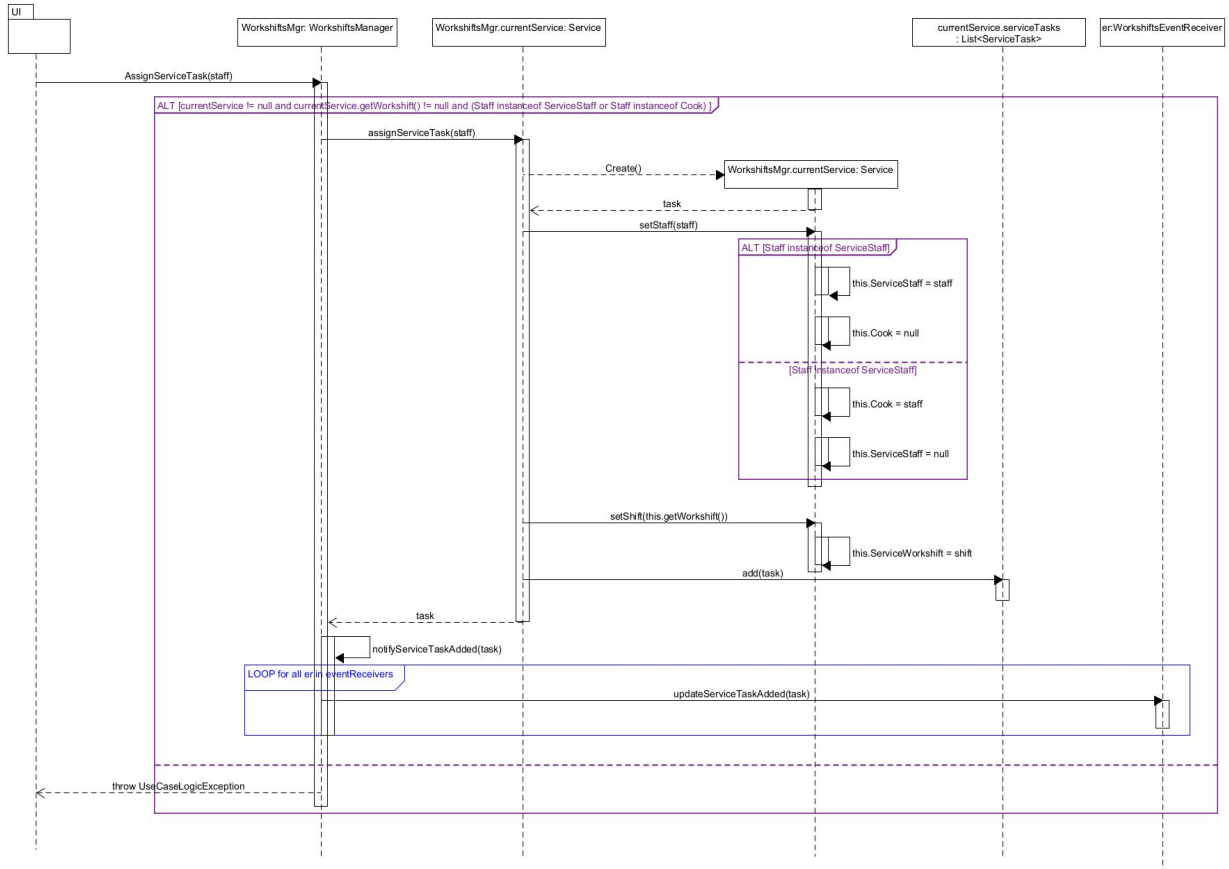


## 2b.1 - DeleteKitchenWorkshift

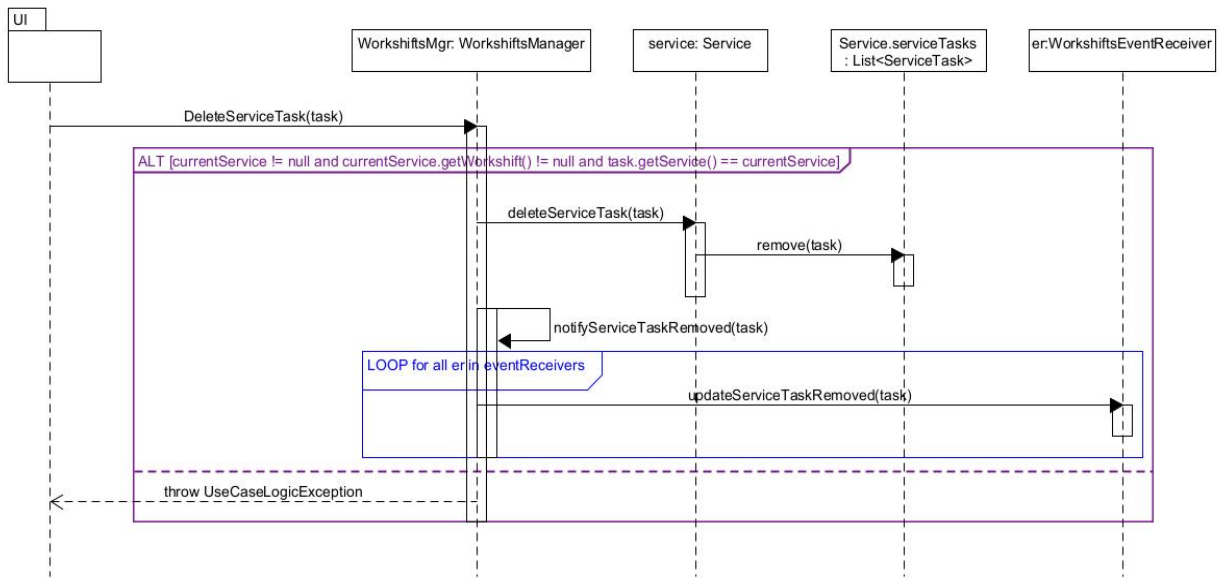


[illegible]

## (1-3)a.5 - AssignServiceTask



### (1-3)a.5a.1 - DeleteServiceTask



**(1-3)a.6 - CumulativeGroupServiceWorkshift**

