

Sviluppo delle Applicazioni Software Laboratorio

a.a 2020/21

Lorenzo Botto – 882837

Marco Madeddu - 883153

Gestire turni

Informazioni generali

Nome caso d'uso: Gestire turni

Portata: Sistema

Livello: Obiettivo utente

Attore primario: L'organizzatore

Parti Interessate: Cuoco, Personale di servizio, Chef

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

Garanzie di successo o post-condizioni: I turni sono registrati e si può applicare ad essi la disponibilità

Scenario principale di successo

#	Attore	Sistema
1	Consulta il tabellone dei turni.	Fornisce il tabellone dei turni.
2	Opzionalmente , aggiunge turno di cucina specificando la fascia oraria e la data. Opzionalmente , specifica il luogo del turno.	Registra il turno di cucina e le loro relative informazioni.
3	Opzionalmente , effettua un raggruppamento per più turni di cucina.	Registra il raggruppamento dei turni di cucina.
4	Opzionalmente , consulta le informazioni sugli eventi.	Fornisce le informazioni sugli eventi.
5	Opzionalmente , aggiunge turno di servizio indicando il tempo necessario per preparare l'evento ed il tempo per sistemare quando l'evento è concluso.	Registra il turno di servizio e le relative informazioni.
6	Opzionalmente , aggiunge una scadenza per il ritiro da un turno.	Assegna una scadenza per il ritiro da un turno.
7	Opzionalmente , blocca un turno.	Registra il blocco del turno.
	<i>Torna al passo 2 oppure termina il caso d'uso.</i>	

Eccezione 2a

#	Attore	Sistema
2a.1	Opzionalmente , aggiunge turni di cucina specificando la fascia oraria e se eventualmente sono turni ricorrenti. Opzionalmente , specifica il luogo del turno.	La cucina è già occupata da un altro turno di cucina e quindi non è concessa una sovrapposizione. L'aggiunta non è registrata.

Estensione 2b

#	Attore	Sistema
2b.1	Modifica turno di cucina, indicando almeno un cambiamento tra ora di inizio, ora di fine, luogo e data.	Registra la modifica al turno di cucina.

Estensione 2b.1a

#	Attore	Sistema
2b.1a.1	Modifica una ricorrenza di turni di cucina, indicando almeno un cambiamento tra ora di inizio, ora di fine, luogo.	Registra la modifica ai turni di cucina della ricorrenza.

Eccezione 2b.1a.1a

#	Attore	Sistema
2b.1a.1a.1	Modifica una ricorrenza di turni di cucina, indicando almeno un cambiamento tra ora di inizio, ora di fine, luogo.	Ci sono già disponibilità per il turno da parte dei cuochi, quindi non è più modificabile. La modifica non è registrata.

Eccezione 2b.1a.1b

#	Attore	Sistema
2b.1a.1b.1	Modifica una ricorrenza di turni di cucina, indicando almeno un cambiamento tra ora di inizio, ora di fine, luogo.	La cucina è già occupata da un altro turno di cucina e quindi non è concessa una sovrapposizione. La modifica non è registrata.

Eccezione 2b.1b

#	Attore	Sistema
2b.1b.1	Modifica turno di cucina, indicando almeno un cambiamento tra ora di inizio, ora di fine, luogo e data.	Ci sono già disponibilità per il turno da parte dei cuochi, quindi non è più modificabile. La modifica non è registrata.

Eccezione 2b.1c

#	Attore	Sistema
2b.1c.1	Modifica turno di cucina, indicando almeno un cambiamento tra ora di inizio, ora di fine, luogo e data.	La cucina è già occupata da un altro turno di cucina e quindi non è concessa una sovrapposizione. La modifica non è registrata.

Estensione 2c

#	Attore	Sistema
2c.1	Elimina turno di cucina.	Registra l'eliminazione al turno di cucina.

Estensione 2c.1a

#	Attore	Sistema
2c.1a.1	Elimina una ricorrenza di turni di cucina.	Registra l'eliminazione della ricorrenza.

Estensione 2d

#	Attore	Sistema
2d.1	Inserisce dei turni indicando il tipo della ricorrenza (settimanale, mensile o annuale), la quale inizia dalla data del primo turno indicato fino alla data finale indicata dall'utente, specificando la fascia oraria. Opzionalmente , specifica il luogo del turno.	Registra l'inserimento della ricorrenza e tutti i turni che la compongono.

Eccezione 2d.1a

#	Attore	Sistema
2d.1a.1	Inserisce dei turni indicando il tipo della ricorrenza (settimanale, mensile o annuale), la quale inizia dalla data del primo turno indicato fino alla data finale indicata dall'utente, specificando la fascia oraria. Opzionalmente , specifica il luogo del turno.	La cucina è già occupata da un altro turno di cucina e quindi non è concessa una sovrapposizione. L'aggiunta non è registrata.

Eccezione 3a

#	Attore	Sistema
3a.1	Opzionalmente , effettua un raggruppamento per più turni di cucina.	Ci sono già disponibilità per qualche turno del raggruppamento da parte dei cuochi, quindi non è raggruppabile. Il raggruppamento non è registrato.

Estensione 3b

#	Attore	Sistema
3b.1	Effettua dei raggruppamenti su più turni di cucina ricorrenti, indicando la data di inizio ricorrenza e data di fine di ricorrenza.	Registra i raggruppamenti dei turni di cucina.

Eccezione 3b.1a

#	Attore	Sistema
3b.1a.1	Effettua dei raggruppamenti su più turni di cucina ricorrenti, indicando la data fine di ricorrenza.	Ci sono già disponibilità per qualche turno del raggruppamento da parte dei cuochi, quindi non è raggruppabile. Il raggruppamento non è registrato.

Eccezione 3b.1b

#	Attore	Sistema
3b.1b.1	Effettua dei raggruppamenti su più turni di cucina ricorrenti, indicando la data fine di ricorrenza.	I turni non hanno la stessa tipologia di ricorrenza. Il raggruppamento non è registrato.

Estensione 3c

#	Attore	Sistema
3c.1	Elimina raggruppamento.	Registra l'eliminazione del raggruppamento.

Estensione 5a

#	Attore	Sistema
5a.1	Modifica turno di servizio.	Registra la modifica al turno di servizio.

Eccezione 5a.1a

#	Attore	Sistema
5a.1a.1	Modifica turno di servizio.	Ci sono già disponibilità per il turno da parte del personale di servizio, quindi non è più modificabile. La modifica non è registrata.

Estensione 5b

#	Attore	Sistema
5b.1	Elimina turni di servizio.	Registra l'eliminazione ai turni di servizio.

Estensione 5c

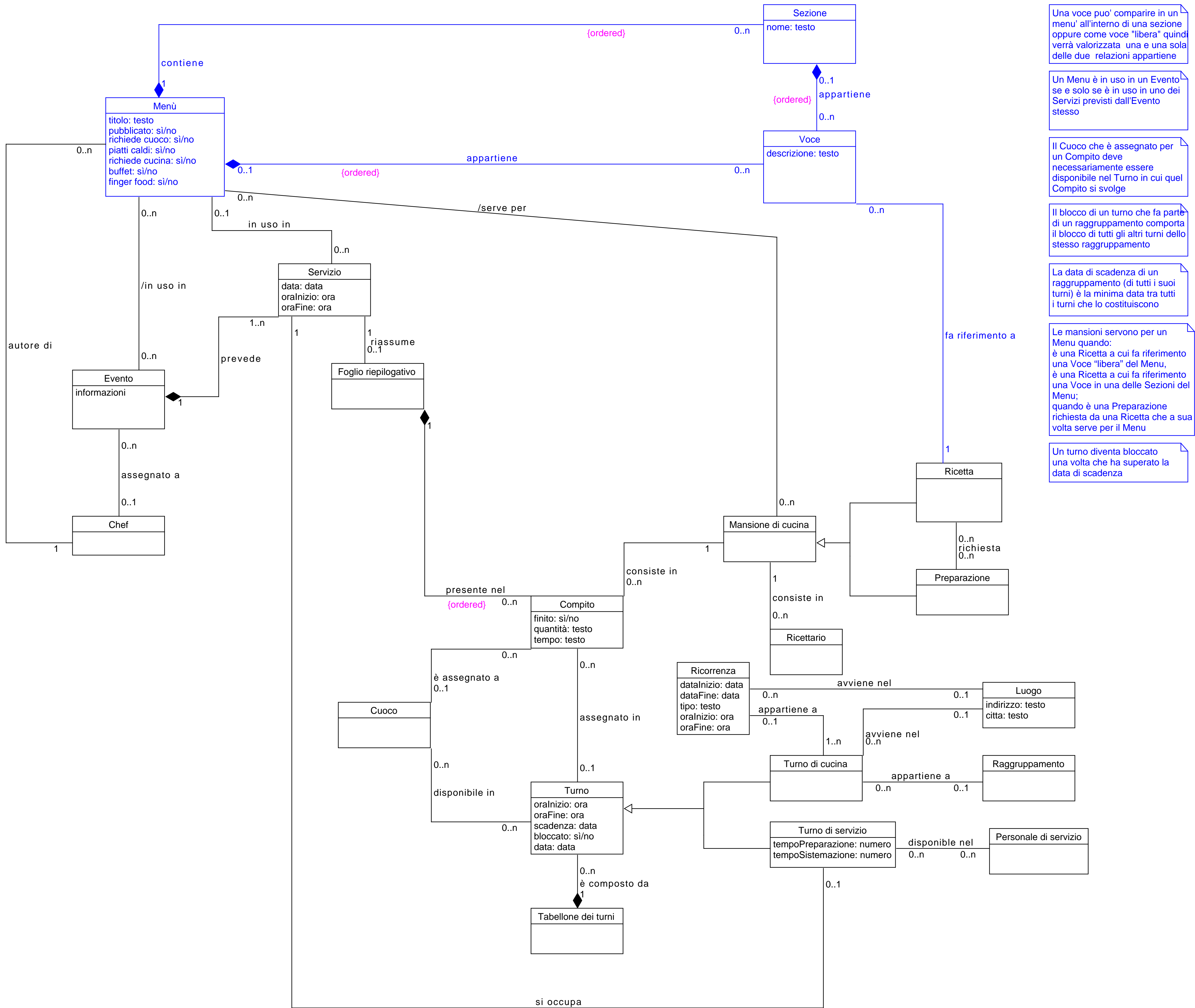
#	Attore	Sistema
5c.1	L'organizzatore precompila in modo automatico i turni di servizio per un evento.	Registra i turni di servizio per il relativo evento.

Estensione 6a

#	Attore	Sistema
6a.1	Modifica della scadenza di un turno.	Registra la modifica della scadenza di un turno o di un raggruppamento di più turni.

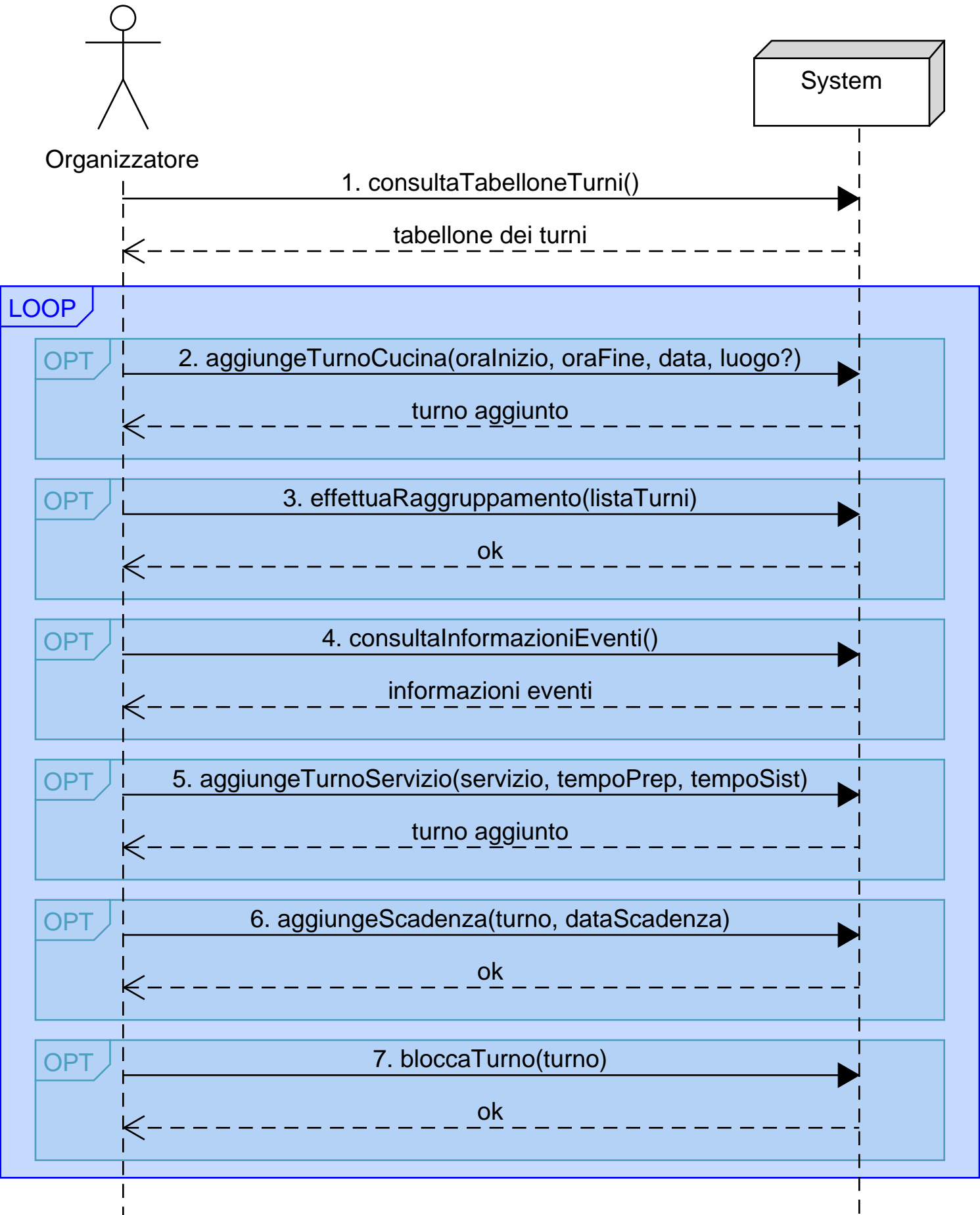
Estensione 7a

#	Attore	Sistema
7a.1	Sblocca un turno.	Registra lo sblocco del turno.



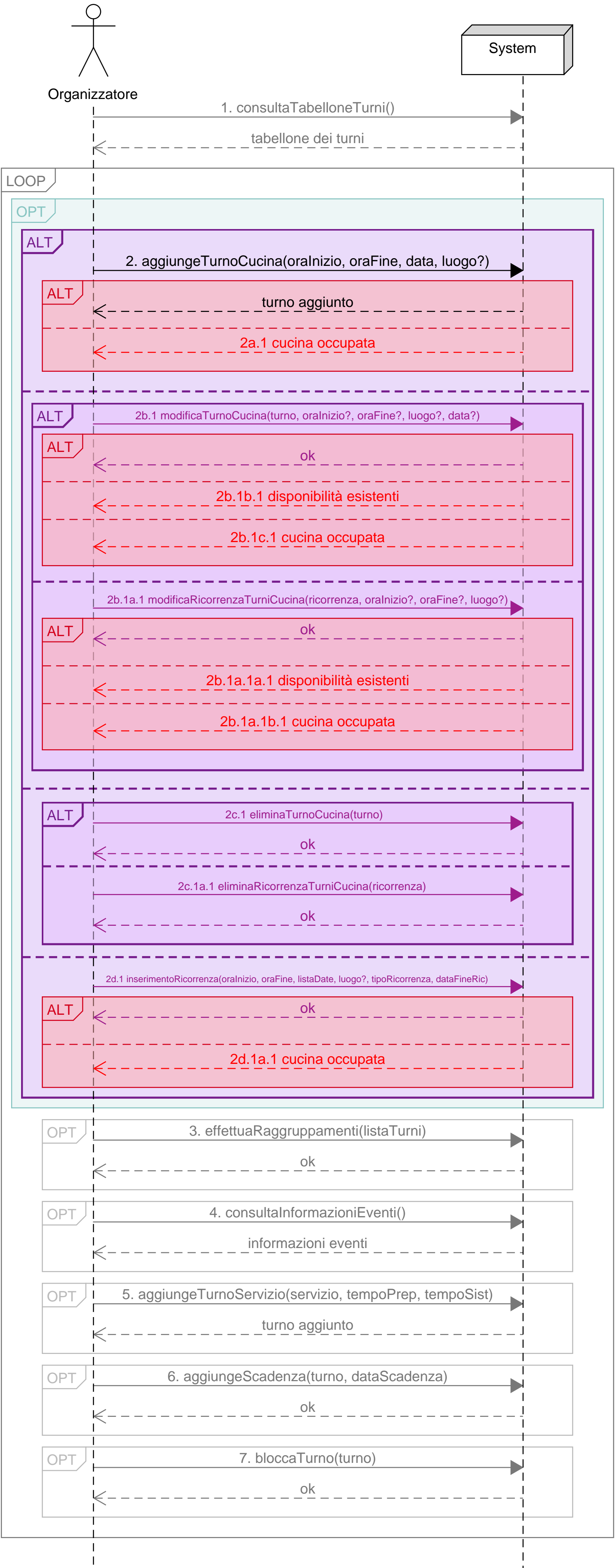
Diagrammi di sequenza di sistema per l'UC Gestire turni

Scenario principale



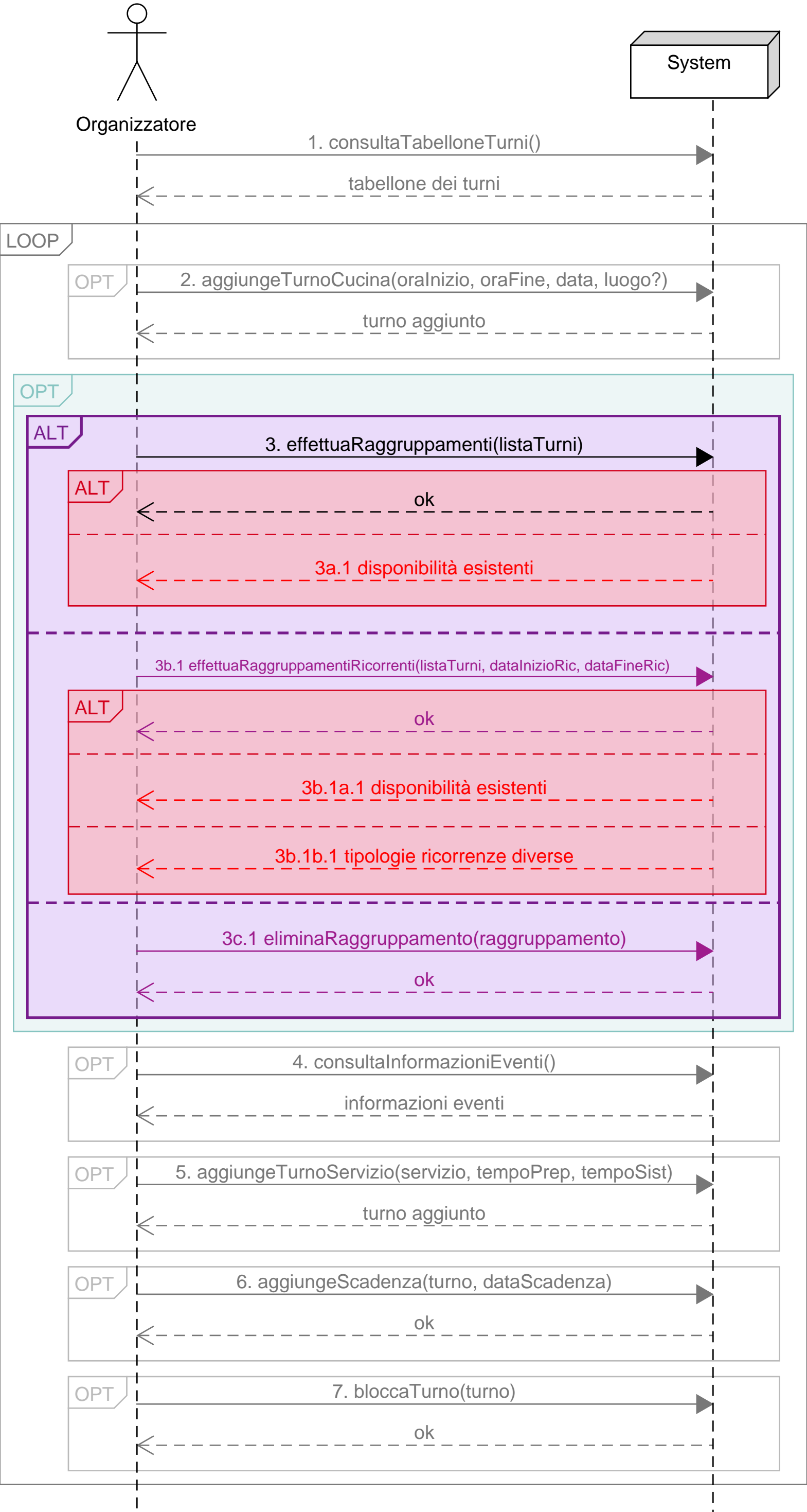
Diagrammi di sequenza di sistema per l'UC Gestire turni

Estensioni al passo 2 (2b, 2b.1a, 2c, 2c.1a, 2d)
e relative eccezioni (2a, 2b.1a.1a, 2b.1a.1b, 2b.1b, 2b.1c, 2d.1a)



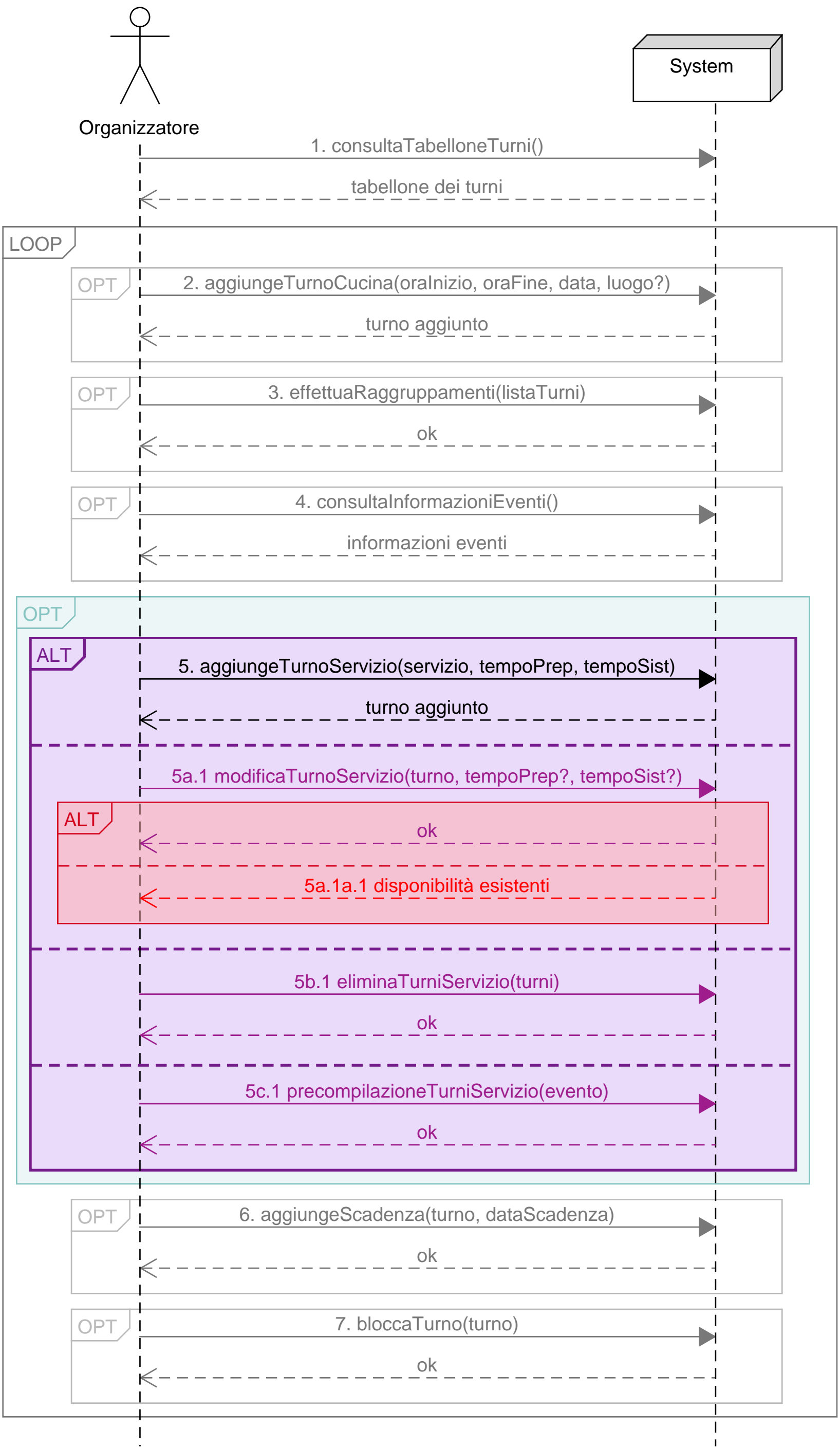
Diagrammi di sequenza di sistema per l'UC Gestire turni

Estensioni al passo 3 (3b, 3c)
e relative eccezioni (3a, 3b.1a, 3b.1b)



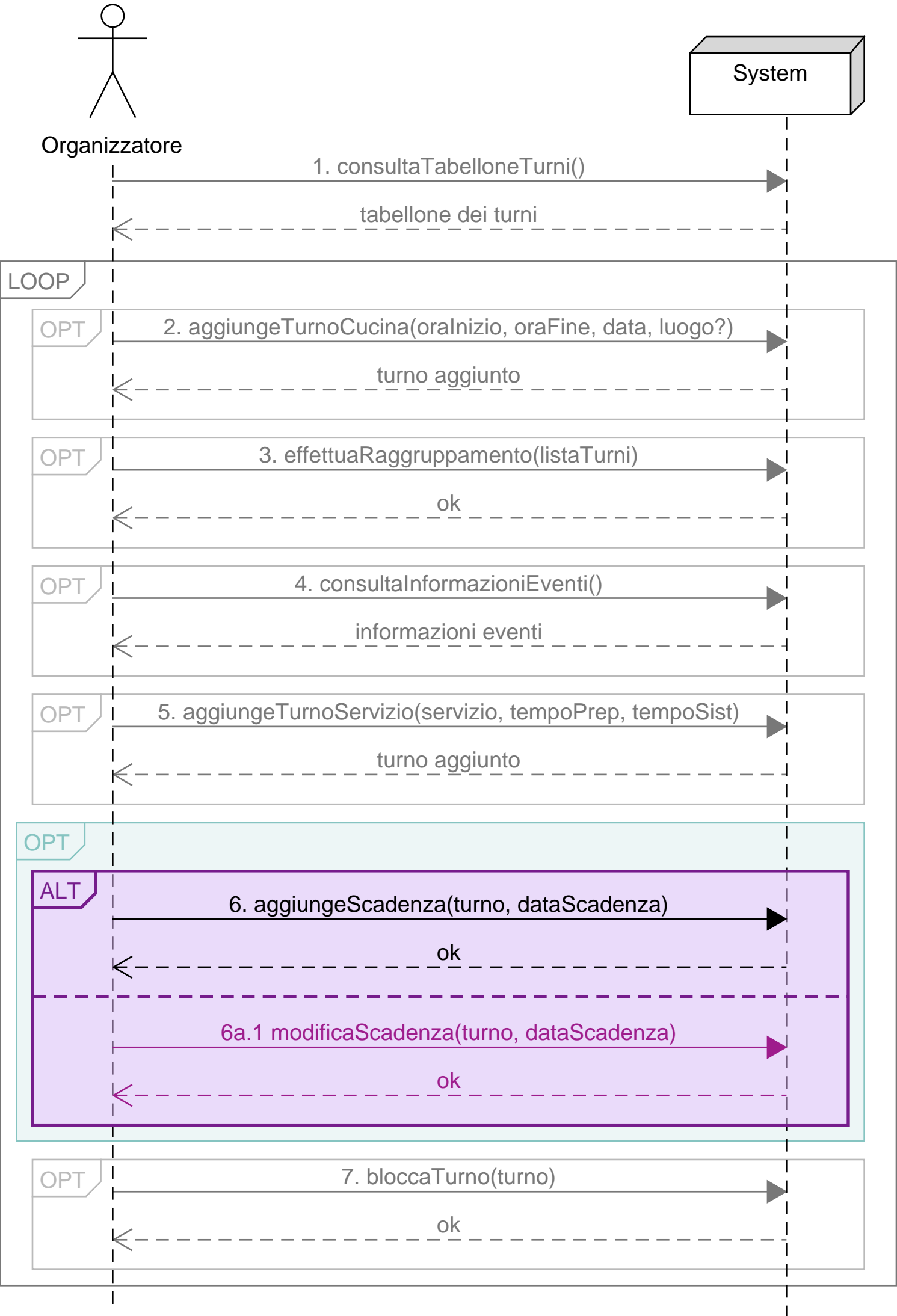
Diagrammi di sequenza di sistema per l'UC Gestire turni

Estensioni al passo 5 (5a, 5b, 5c)
e relative eccezioni (5a.1a)



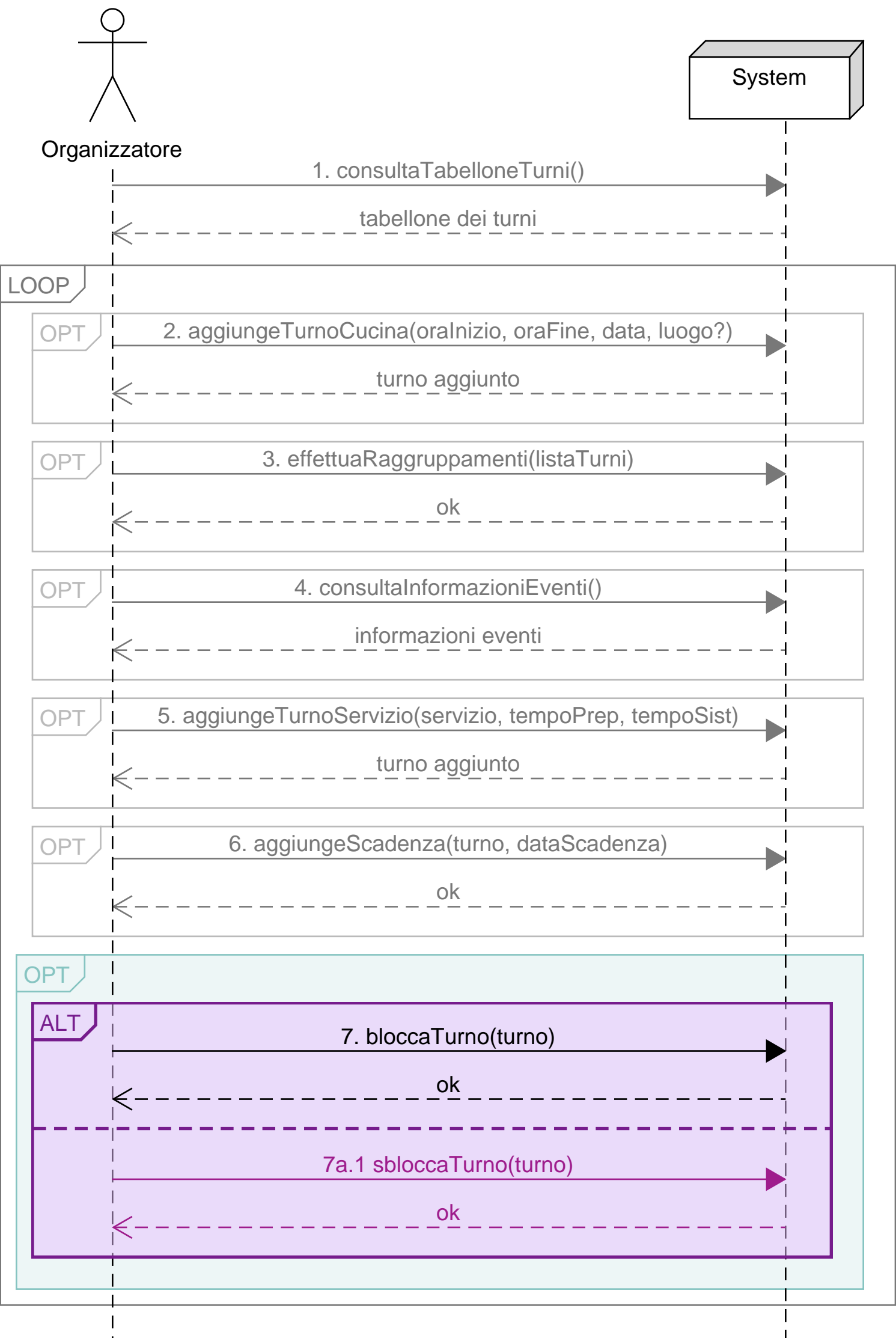
Diagrammi di sequenza di sistema per l'UC Gestire turni

Estensioni al passo 6 (6a)



Diagrammi di sequenza di sistema per l'UC Gestire turni

Estensioni al passo 7 (7a)



Contratti per lo UC “Gestire turni”

Pre-condizione generale: l'attore è identificato con un'istanza *or* di Organizzatore

1. consultaTabelloneTurni ()

Interrogazione al sistema.

Pre-condizioni: ---

Post-condizioni: ---

2. aggiungeTurnoCucina (oraInizio: ora, oraFine: ora, data: Data, luogo?: Luogo)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*
- Esiste un Luogo *ls* standard che identifica la cucina standard

Post-condizioni:

- [se non esiste un Turno *t2* che **avviene nel luogo** (se è specificato, altrimenti in *ls*), con oraInizio <= *t2.oraInizio* <= oraFine oppure *t2.oraInizio* <= oraInizio <= *t2.oraFine* e *t2.data* = data]:
 - è stata creata un'istanza *t* di Turno di cucina
 - [se luogo è specificato] *t* **avviene nel luogo**
 - [se luogo non è specificato] *t* **avviene nel ls**
 - *tab* è **composto** da *t*
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine
 - *t.data* = data

2b.1. modificaTurnoCucina (turno: Turno di cucina, oraInizio?: ora, oraFine?: ora, luogo?: Luogo, data?: Data)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- [se non esiste un Turno *tu* che **avviene nel luogo** (se è specificato, altrimenti in turno.luogo), con oraInizio <= *tu.oraInizio* <= oraFine oppure *tu.oraInizio* <= oraInizio <= *tu.oraFine*, *tu.data* = data, e non esiste un Cuoco *c* **disponibile in turno**]:
 - [se luogo è specificato] l'associazione **avviene nel** tra turno ed un Luogo *l1* è eliminata, turno avviene nel luogo
 - [se oraInizio è specificata] turno.oraInizio = oraInizio
 - [se oraFine è specificata] turno.oraFine = oraFine
 - [se data è specificata] turno.data = data
 - [se turno appartiene a una Ricorrenza *r*] l'associazione **appartiene a** tra *r* e turno è stata eliminata

2b.1a.1. modificaRicorrenzaTurniCucina (ricorrenza: Ricorrenza, oraInizio?: ora, oraFine?: ora, luogo?: Luogo)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- [se per ogni Turno di cucina *t* che **appartiene a ricorrenza** non esiste un Turno *tu* che **avviene nel luogo** (se è specificato, altrimenti in *t.luogo*), con *orario* che si sovrappone all'orario di *t* (o se specificati oraInizio e oraFine), *tu.data* = *t.data*, e non esiste un Cuoco *c* **disponibile in t**]:
 - [se luogo è specificato] ricorrenza avviene nel luogo
 - [se oraInizio è specificata] ricorrenza.oraInizio = oraInizio
 - [se oraFine è specificata] ricorrenza.oraFine = oraFine
 - per ogni *t*:
 - [se luogo è specificato] *t avviene nel luogo*
 - [se oraInizio è specificata] *t.oraInizio* = oraInizio
 - [se oraFine è specificata] *t.oraFine* = oraFine

2c.1. eliminaTurnoCucina (turno: Turno di cucina)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- turno è stato eliminato

2c.1a.1. eliminaRicorrenzaTurniCucina (ricorrenza: Ricorrenza)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- ricorrenza è stato eliminato
- per ogni Turno di cucina *t* che **appartiene a** ricorrenza:
 - *t* è stato eliminato

2d.1. inserimentoRicorrenza (oraInizio: ora, oraFine: ora, lista d1,...,dN di Data, luogo?: Luogo, tipoRicorrenza: testo, dataFineRic: data)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*
- Esiste un Luogo *ls* standard che identifica la cucina standard

Post-condizioni:

- [se per ogni Data *d* che rispetta il pattern definito dalla lista d1,...,dN e tipoRicorrenza non esiste un Turno *tu* che **avviene nel** luogo (se è specificato, altrimenti in *ls*), con oraInizio <= *tu.oraInizio* <= oraFine oppure *tu.oraInizio* <= oraInizio <= *tu.oraFine*]:
 - è stata creata un'istanza *r* di Ricorrenza
 - [se luogo è specificato] *r* **avviene nel** luogo
 - [se luogo non è specificato] *r* **avviene nel** ls
 - *r.oraInizio* = oraInizio
 - *r.oraFine* = oraFine
 - *r.tipo* = tipoRicorrenza
 - *r.dataInizio* = d1
 - *r.dataFine* = dataFineRic
 - per ogni *d*:
 - è stata creata un'istanza *t* di Turno di cucina
 - [se luogo è specificato] *t* **avviene nel** luogo
 - [se luogo non è specificato] *t* **avviene nel** ls
 - *tab* è **composto** da *t*
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine
 - *t.data* = *d*
 - *t* **appartiene a** *r*

3. effettuaRaggruppamenti (lista t1,...,tN di Turni)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- [se non esiste un Cuoco *c* **disponibile in** *t* con *t* appartenente a *t1,...,tN*]:
 - è stata creata un'istanza *r* di Raggruppamento
 - per ogni *t* appartenente a *t1,...,tN*:
 - *t* **appartiene a** *r*

3b.1. effettuaRaggruppamentiRicorrenti (lista t1,...,tN di Turno di Cucina, dataInizioRic: data, dataFineRic: data)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*
- Ogni Ricorrenza *ri* a cui **appartiene** un turno contenuto nella lista *t1,...,tN* ha lo stesso *ri.tipo*

Post-condizioni:

- [se per ogni Turno di cucina *t_i* della lista *t1,...,tN* **appartiene** a una Ricorrenza *ric*, e tutte le *ric* hanno stesso *ric.tipo* e *ric.data* >= dataFineRic e non esiste un Cuoco *c* **disponibile in** alcun turno *tu* che **appartiene** alle varie *ric*]:
 - per ogni settimana, mese, anno (se la ricorrenza è settimanale, mensile o annuale) compreso tra dataInizioRic e dataFineRic:
 - è stata creata un'istanza di Raggruppamento *r*
 - per ogni Turno di cucina *t* che rispetta il pattern di *r* e *t* **appartiene** a una delle *ric*:
 - *t* **appartiene a** *r*

3c.1. eliminaRaggruppamento (raggruppamento: Raggruppamento)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- raggruppamento è stato eliminato

4. consultaInformazioniEventi ()

Interrogazione al sistema.

Pre-condizioni: ---

Post-condizioni: ---

5. aggiungeTurnoServizio (servizio: Servizio, tempoPrep: numero, tempoSist: numero)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- è stata creata un'istanza *ts* di Turno di servizio
- *ts* **si occupa** del servizio
- *ts*.tempoPrep = tempoPrep
- *ts*.tempoSist = tempoSist
- *tab* è **composto** da *ts*

5a.1 modificaTurnoServizio(turno: Turno di servizio, tempoPrep?: numero, tempoSist?: numero)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni: [se non esiste un Cuoco *c* o un Personale di servizio *p* **disponibile in** turno]

- [se tempoPrep è specificato] *ts*.tempoPrep = tempoPrep
- [se tempoSist è specificato] *ts*.tempoSist = tempoSist

5b.1 eliminaTurniServizio(insieme T di Turni di servizio)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- per ogni Turni di servizio *t* appartenenti a *T*:
 - è stato eliminato *t*

5c.1 precompilazioneTurniServizio(evento: Evento)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- per ogni Servizio *s* che è **previsto** in un Evento *e*:
 - è stata creata un'istanza *t* di Turno di servizio
 - *t* **si occupa** del servizio
 - *t.data* = *s.data*
 - *t.oraInizio* = *s.oraInizio*
 - *t.oraFine* = *s.oraFine*
 - *tab* è **composto** da *t*

6. aggiungeScadenza(turno: Turno, dataScadenza: data)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- turno. scadenza = dataScadenza

6a.1. modificaScadenza(turno: Turno, dataScadenza: data)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- turno. scadenza = dataScadenza

7. bloccaTurno(turno: Turno)

Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

Post-condizioni:

- turno. bloccato = si

7a.1. sbloccaTurno(turno: Turno)

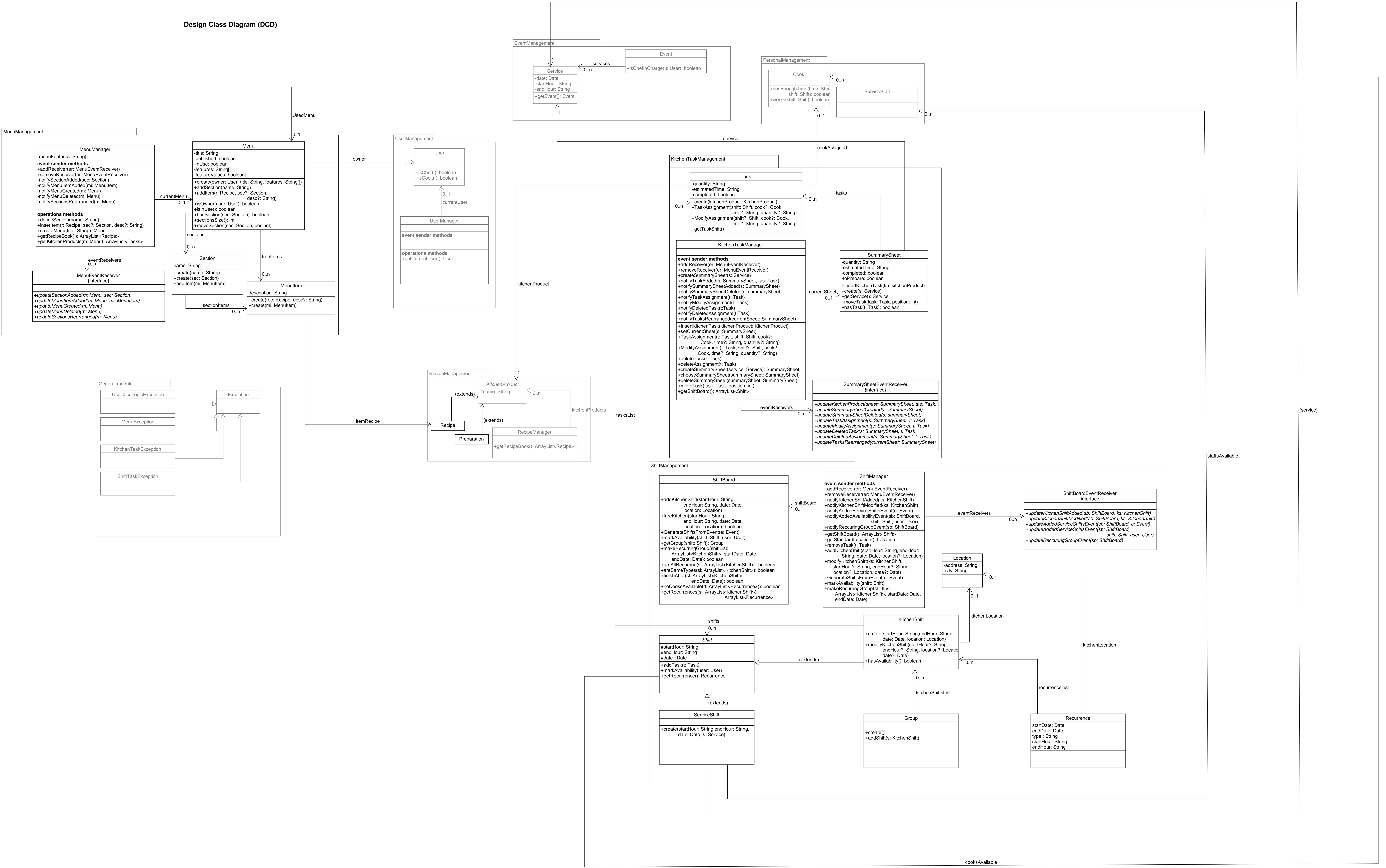
Pre-condizioni:

- Esiste il Tabellone dei turni *tab*

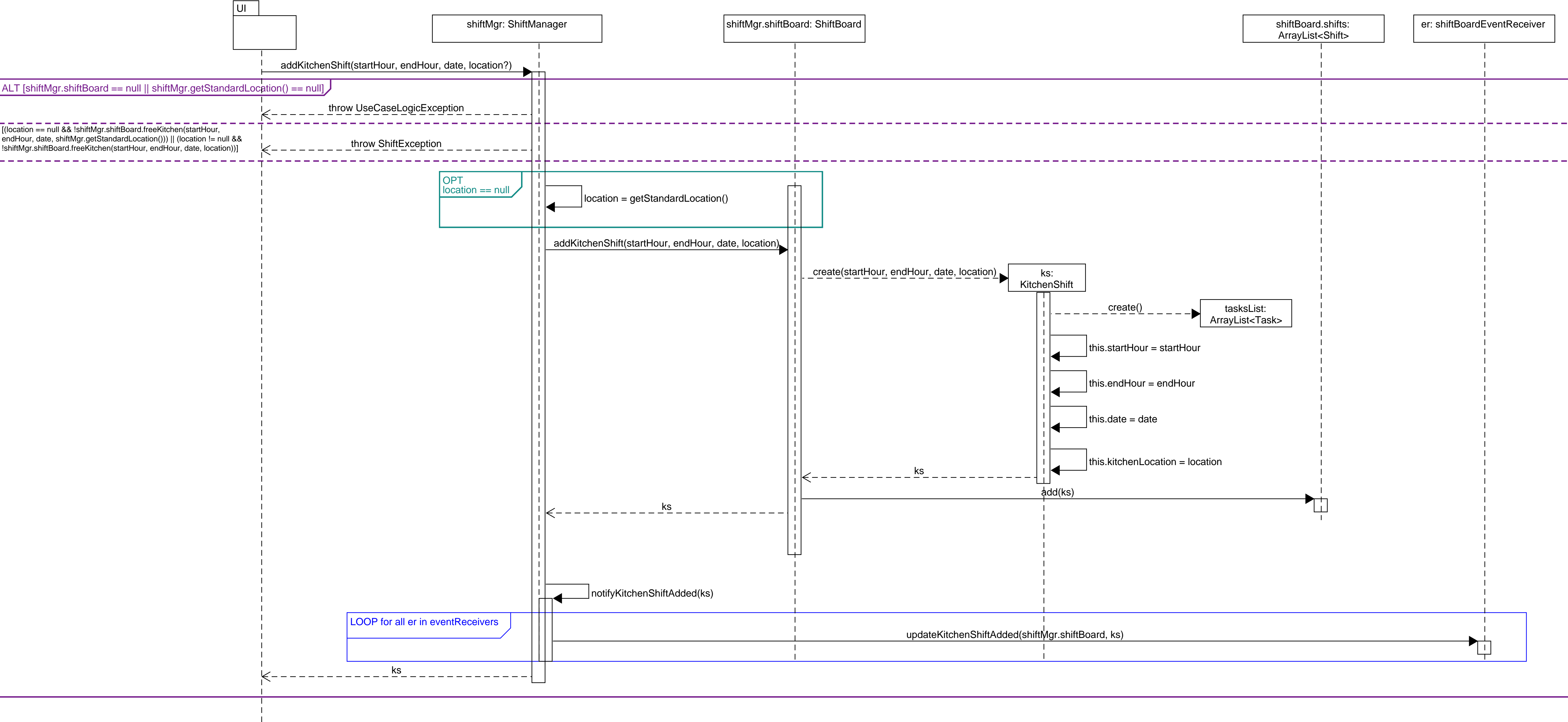
Post-condizioni:

- turno. bloccato = no

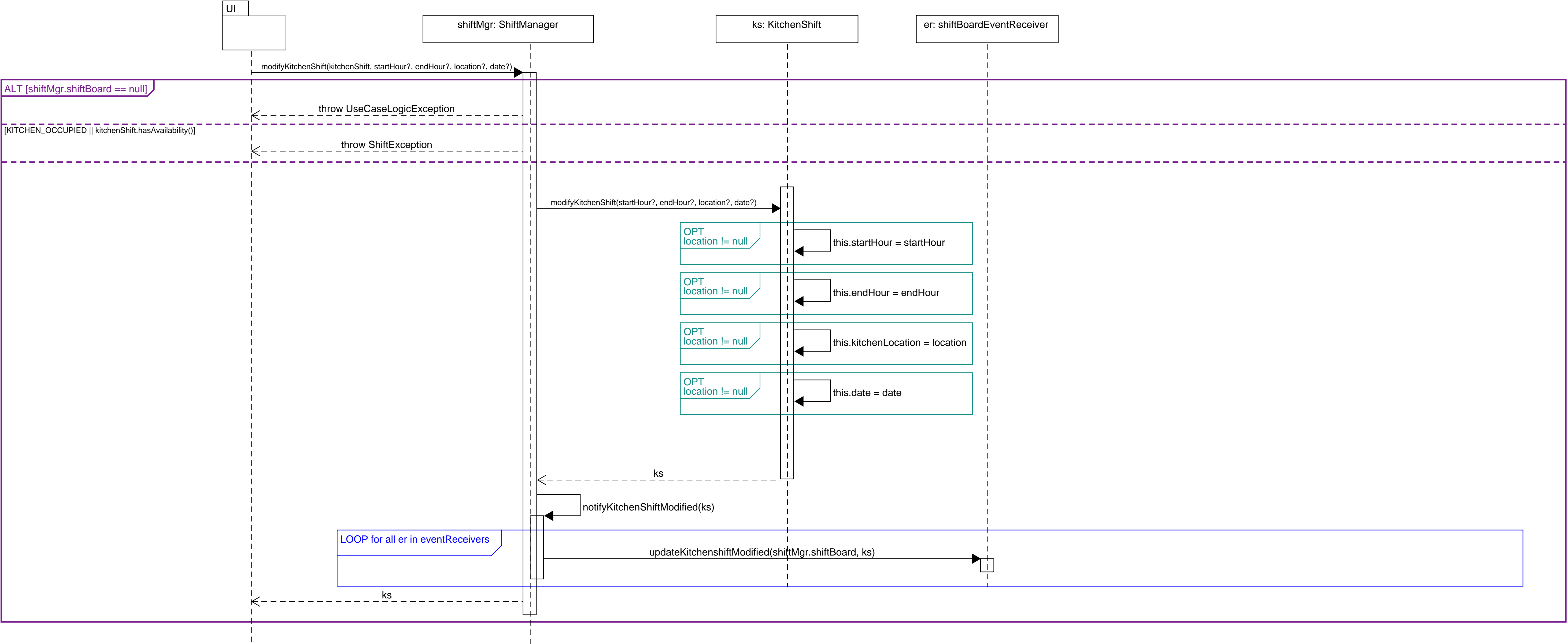
Design Class Diagram (DCD)



2- AddKitchenShift

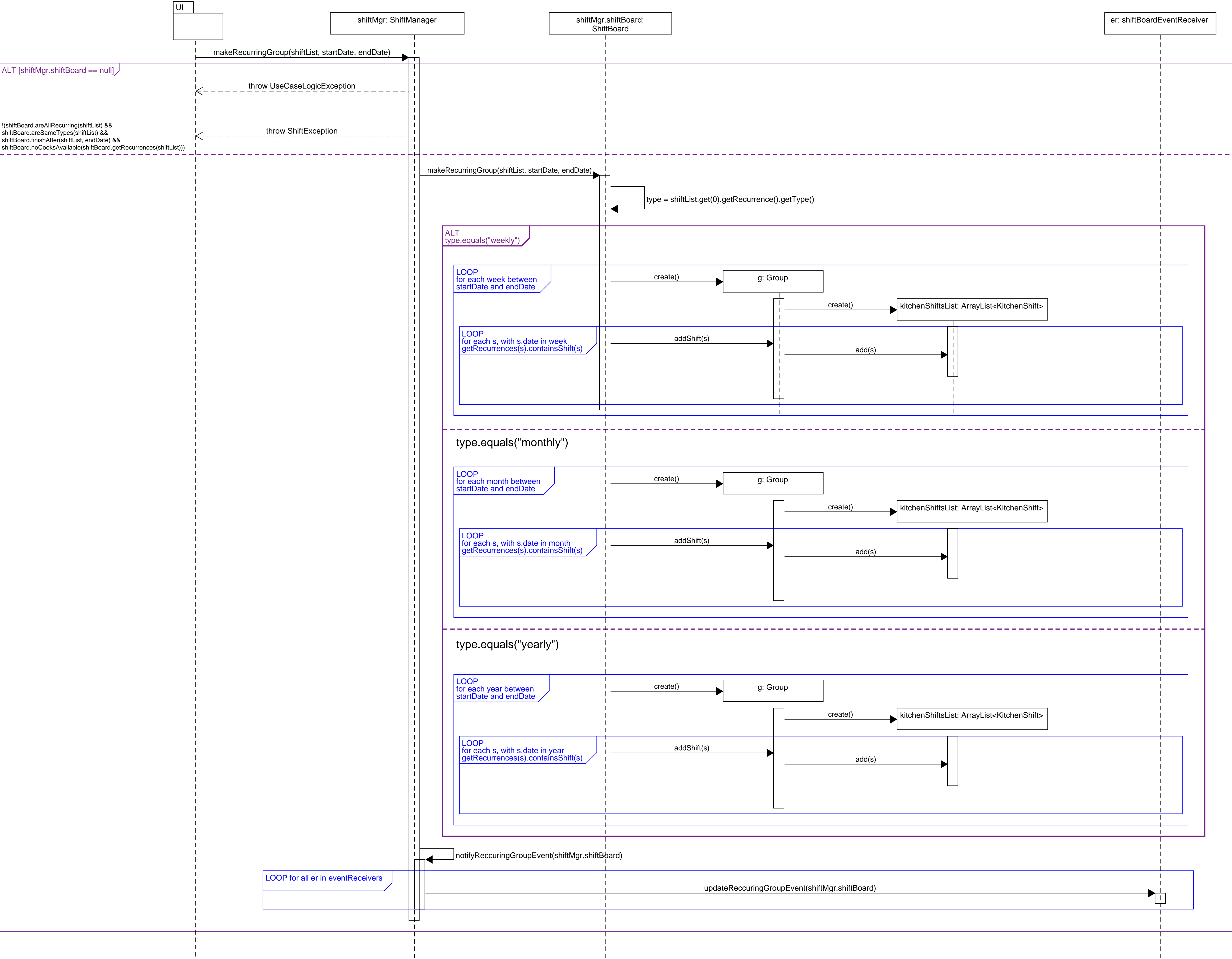


2b - modifyKitchenShift

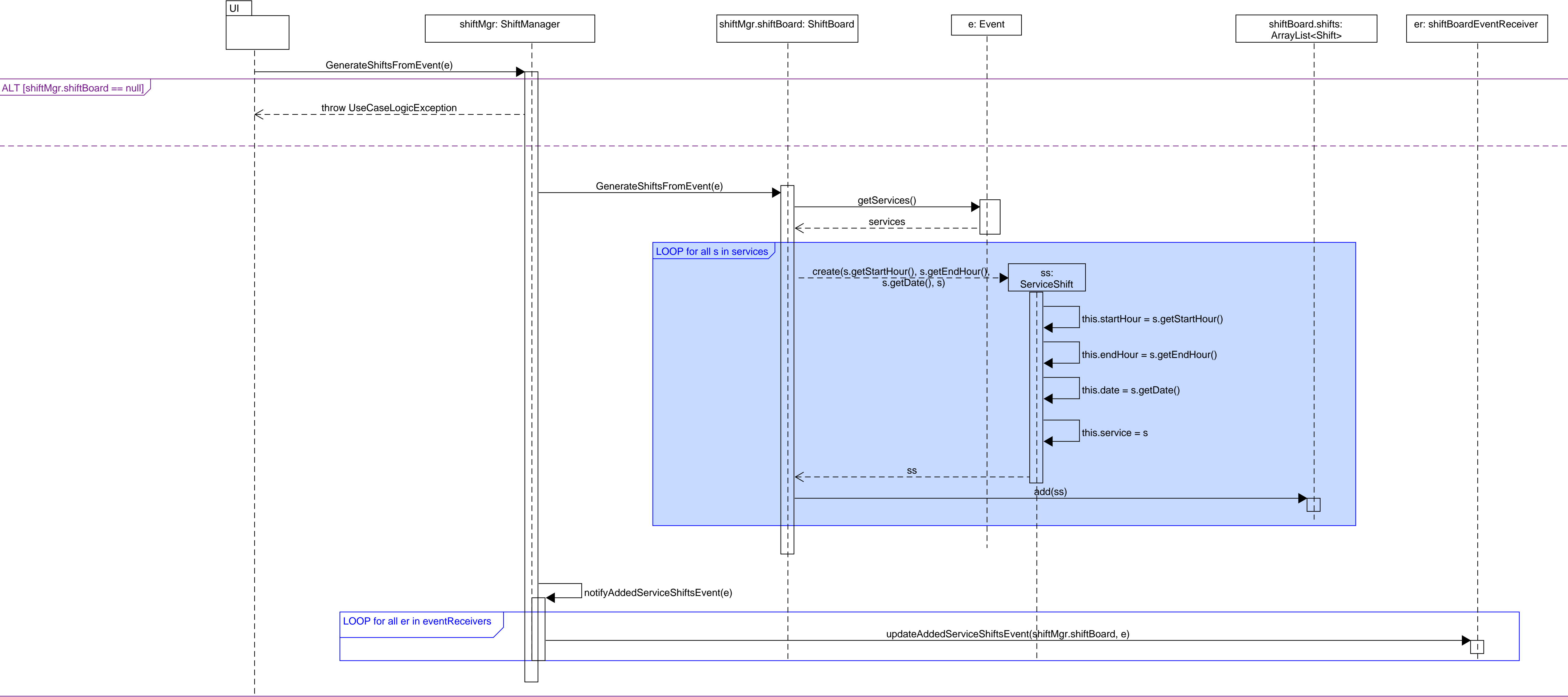


KITCHEN_OCCUPIED = (location == null && !shiftMgr.shiftBoard.freeKitchen(startHour, endHour, date, kitchenShift.getLocation())) || (location != null && !shiftMgr.shiftBoard.freeKitchen(startHour, endHour, date, location))

3b- MakeRecurringGroup



5c- GenerateShiftsFromEvent



Indicare le proprie disponibilità

Informazioni generali

- Nome caso d’uso:** Indicare le proprie disponibilità
- Portata:** Sistema
- Livello:** Obiettivo utente
- Attore primario:** Cuoco, Personale di servizio
- Parti Interessate:** Organizzatore, Chef
- Pre-condizioni:** L’attore deve essere autenticato come Cuoco o Personale di servizio
- Garanzie di successo o post-condizioni:** Le disponibilità sono registrate
- Nota:** I turni forniti all’utente variano in base alla sua autenticazione: il Personale di servizio vedrà solo i turni di servizio, mentre il Cuoco vedrà sia i turni di cucina che i turni di servizio.

Scenario principale di successo

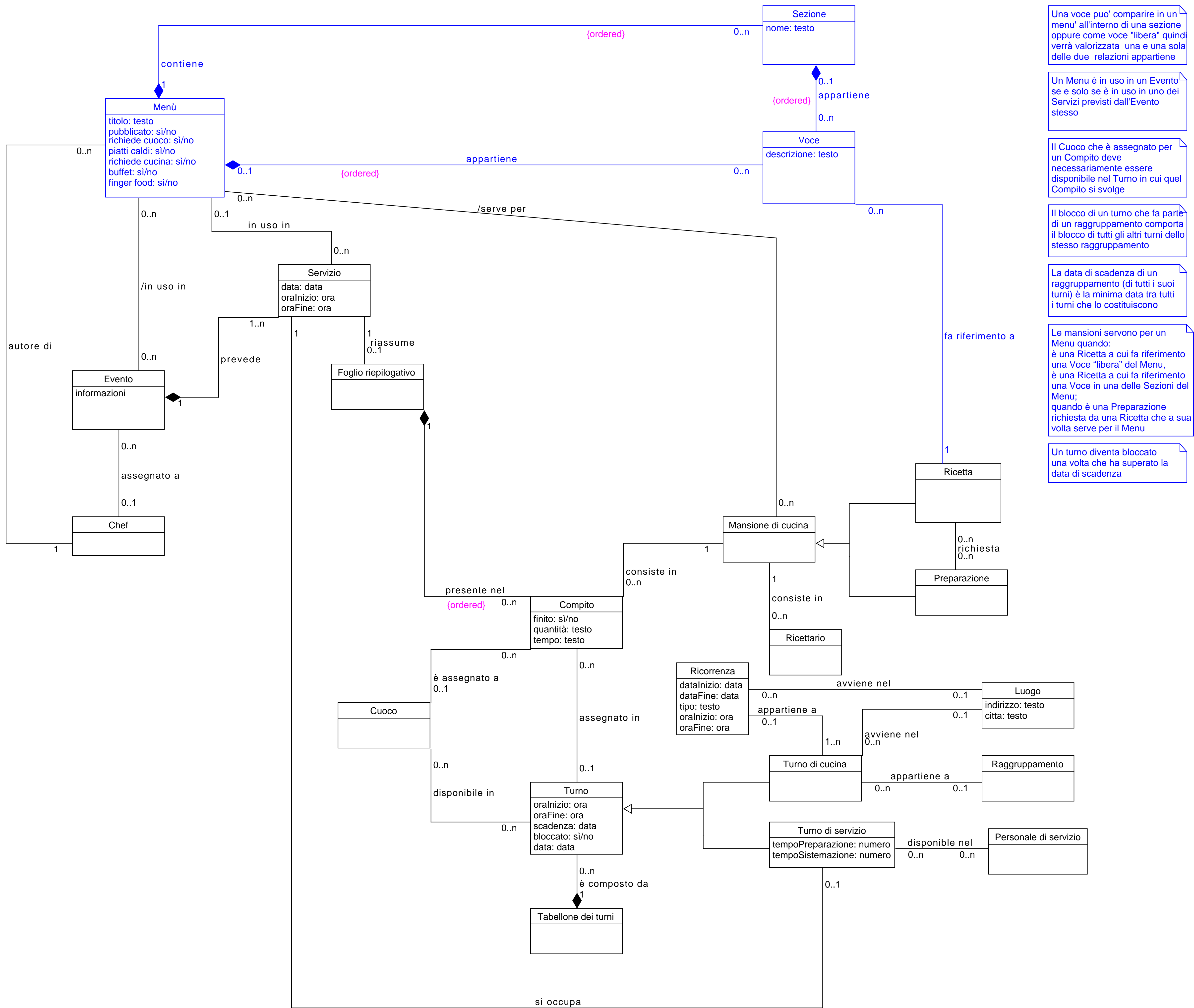
#	Attore	Sistema
1	Consulta il proprio tabellone dei turni.	Fornisce il tabellone dei turni.
2	Segna la propria disponibilità per un turno, che può essere in un raggruppamento.	Registra la disponibilità per un turno ed eventualmente anche agli altri turni del raggruppamento.
	<i>Ripete passo 2 finchè non è soddisfatto oppure termina il caso d’uso.</i>	

Estensione 2a

#	Attore	Sistema
2a.1	Elimina la disponibilità per un turno, che può essere in un raggruppamento.	Registra l'eliminazione della disponibilità per un turno ed eventualmente anche agli altri turni del raggruppamento.

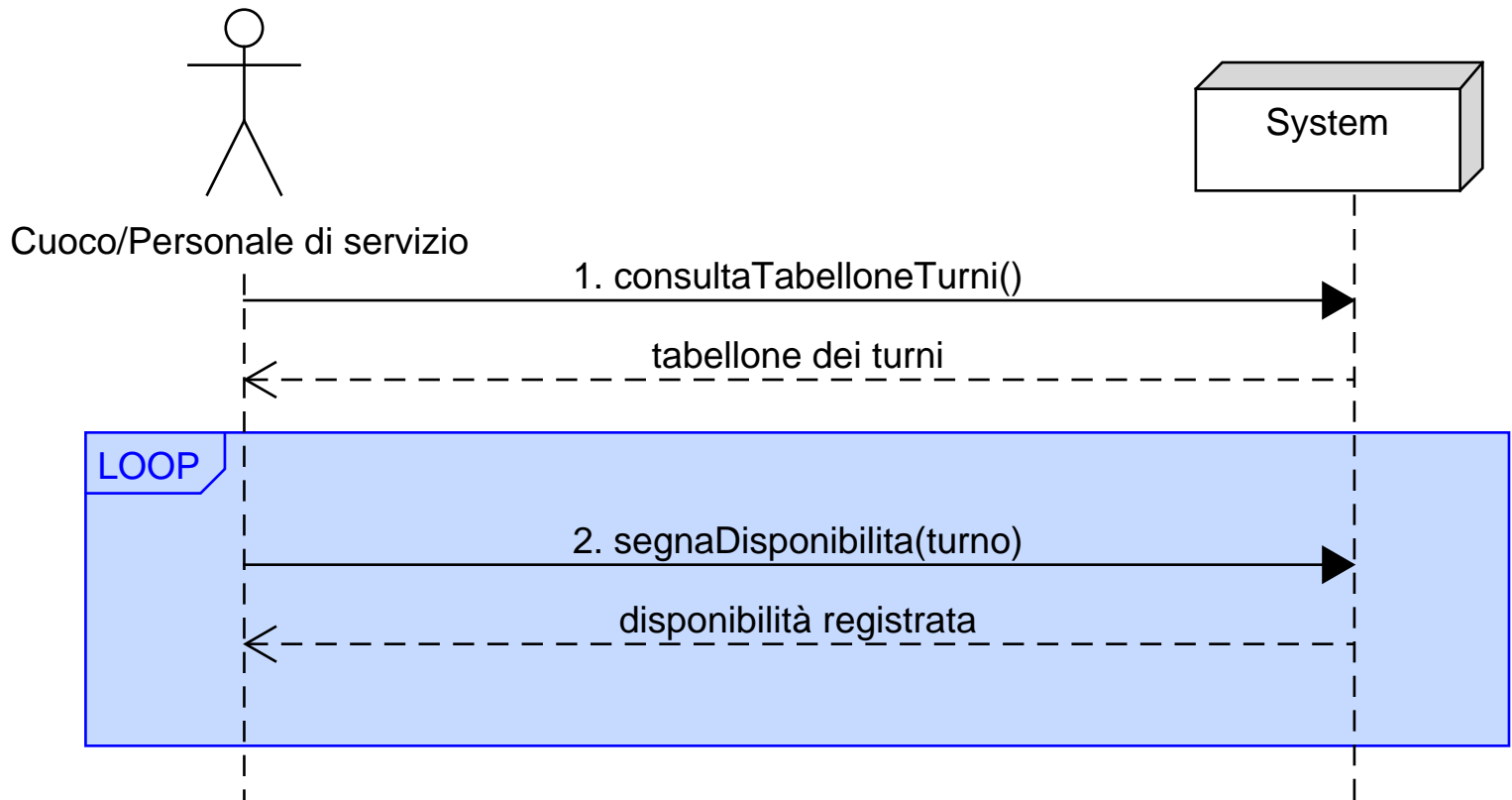
Eccezione 2a.1a

#	Attore	Sistema
2a.1a.1	Elimina la disponibilità per un turno.	Il turno è bloccato perché la data odierna è oltre quella della scadenza. L'eliminazione non è registrata.



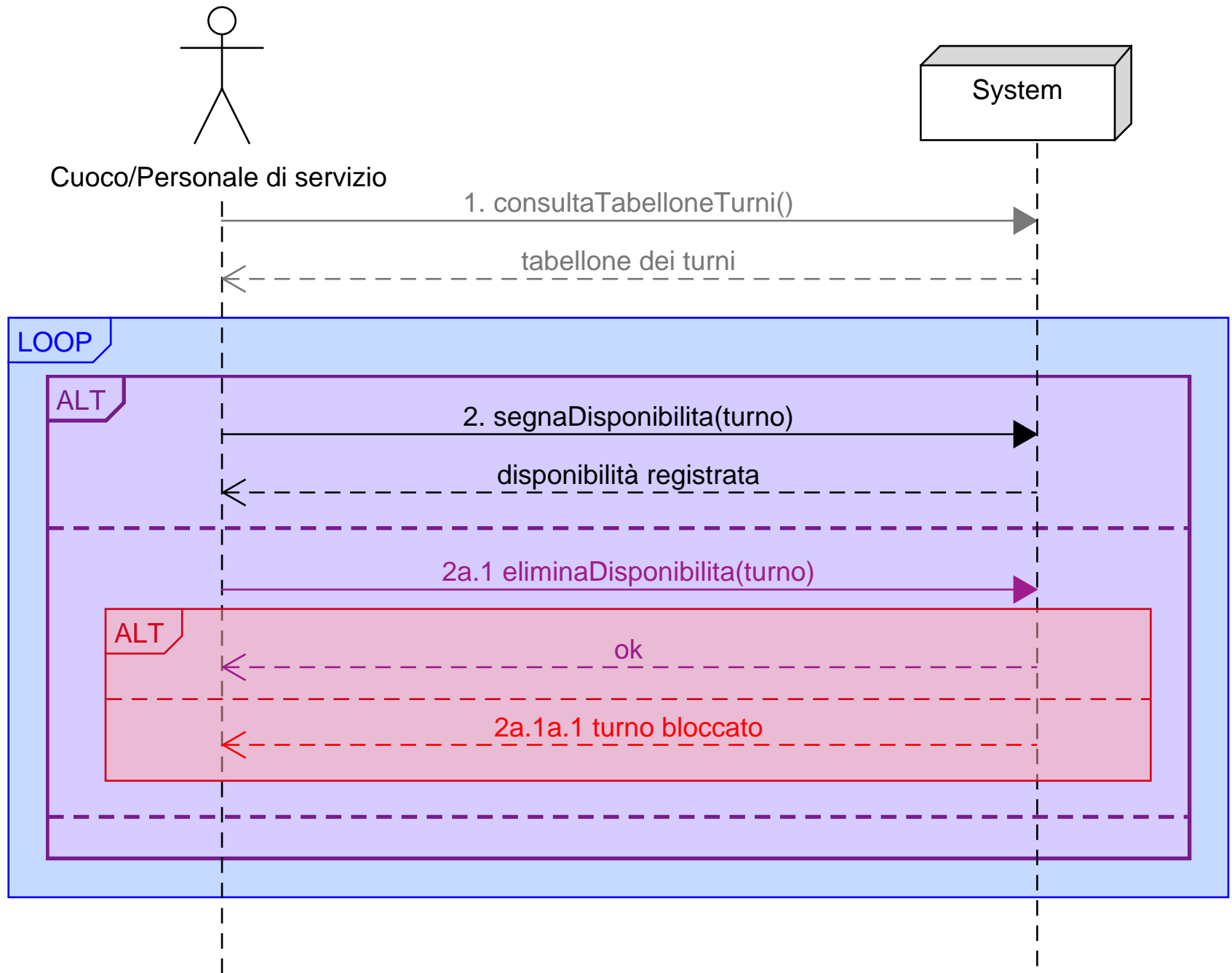
Diagrammi di sequenza di sistema per l'UC Indicare le proprie disponibilità

Scenario principale



Diagrammi di sequenza di sistema per l'UC Indicare le proprie disponibilità

Estensioni al passo 2 (2a)
e relative eccezioni (2a.1a)



Contratti per lo UC “Indicare le proprie disponibilità”

Pre-condizione generale: l'attore è identificato con un'istanza c di Cuoco oppure p di Personale di Servizio

1. consultaTabelloneTurni ()

Interrogazione al sistema.

Pre-condizioni: ---

Post-condizioni: ---

2. segnaDisponibilita (turno: Turno)

Pre-condizioni:

- Esiste il Tabellone dei turni tab

Post-condizioni:

- [se turno **non appartiene** a nessun raggruppamento]:
 - c o p è **disponibile nel** turno
 - [se turno **appartiene** ad un Raggruppamento r]:
Per ogni turno t che **appartiene** a r :
 c o p è **disponibile in** t

2a.1. eliminaDisponibilita (turno: Turno)

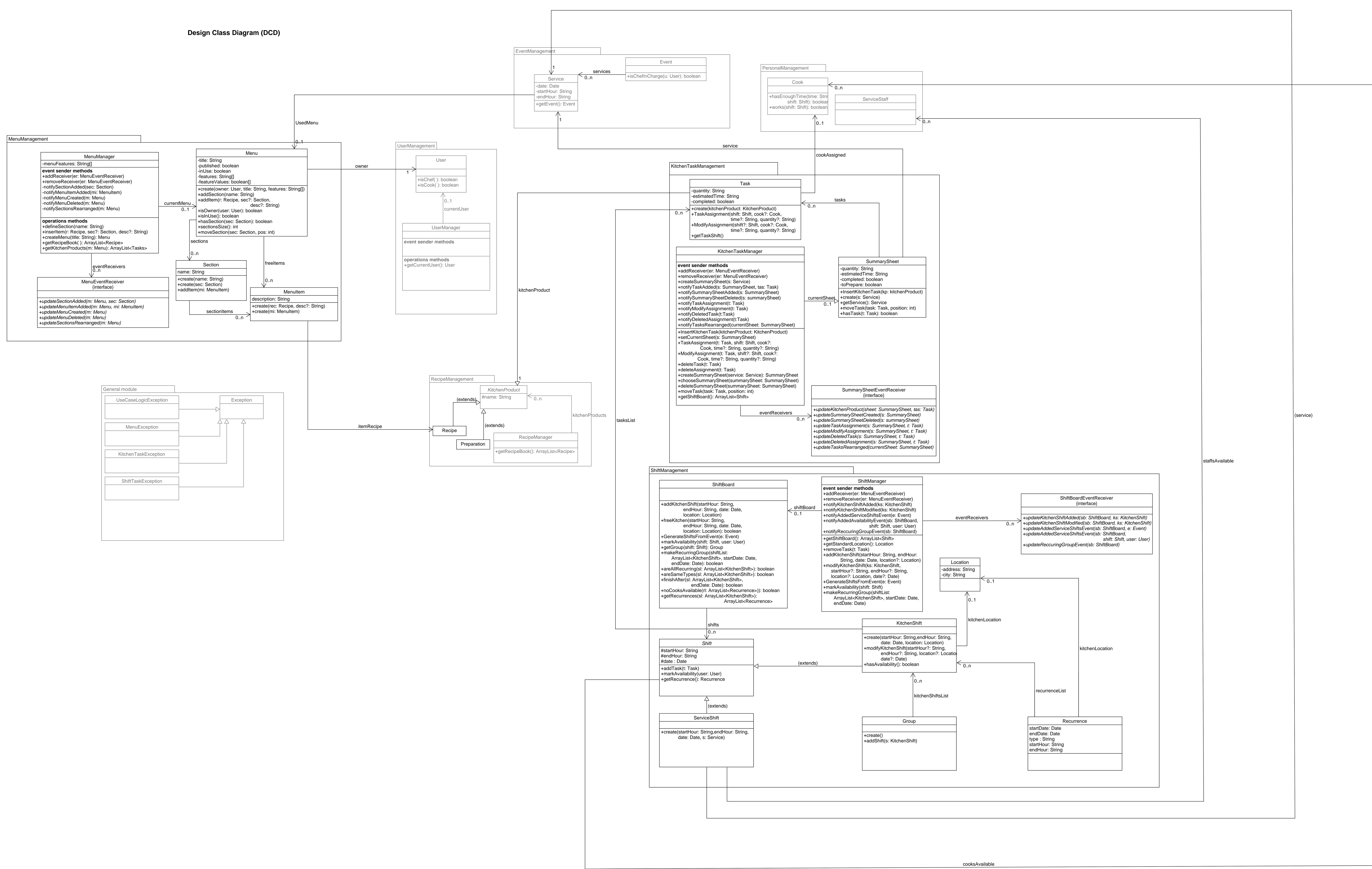
Pre-condizioni:

- Esiste il Tabellone dei turni tab

Post-condizioni:

- [se turno **non appartiene** a nessun raggruppamento e turno non è bloccato] :
 - L'associazione è **disponibile nel** tra c o p e turno è stata eliminata
- [se turno **appartiene** a nessun raggruppamento r e nessun turno t che **appartiene** a r è bloccato]:
 - Per ogni turno t :
 - L'associazione è **disponibile nel** tra c o p e t è stata eliminata

Design Class Diagram (DCD)



2- markAvailability

