

# Sviluppo delle Applicazioni Software

## Laboratorio

a.a 2022/23

Emanuele Carlucci, Federico di Geronimo

## Gestire gli eventi

### Informazioni generali

**Nome caso d'uso:** Gestire gli eventi

**Portata:** Sistema

**Livello:** Obiettivo utente

**Attore primario:** Organizzatore

**Parti Interessate:** Chef, cuoco, personale di servizio

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

**Garanzie di successo o post-condizioni:** L'evento viene creato, vengono creati i servizi relativi e vengono assegnati chef, cuochi, personale e vengono confermati i menù.

#	Attore	Sistema
1	Crea un nuovo evento specificando nome, data d'inizio e cliente	Registra un nuovo evento con il nome specificato
2	Opzionalmente verifica lo stato della cucina	Fornisce lo stato della cucina
3	Opzionalmente compila i dettagli dell'evento	Registra l'evento con i dettagli specificati
4	Opzionalmente aggiunge un servizio e i suoi dettagli	Registra il nuovo servizio per l'evento che si sta organizzando
5	Opzionalmente decide lo chef	Registra lo chef per l'Evento
6	Opzionalmente conferma un menù dello chef per un servizio	Registra la conferma; l'evento viene segnato come "in corso/attivo"
7	Opzionalmente sceglie il personale (di sala, cuochi per servizio) e specifica il ruolo per il turno	Registra il personale selezionato per il turno dell'evento
	<i>Ripete dal passo 2 finché non è soddisfatto</i>	
	<i>Termina il caso d'uso</i>	

### Estensione 1a

#	Attore	Sistema
1a.1	Apri evento esistente	Fornisce scheda evento richiesta
	<i>Se desidera modificare la scheda evento procede al passo 2</i>	
1a.2	Opzionalmente annulla l'evento	Segna l'evento come annullato
	<i>Termina il caso d'uso</i>	

### Estensione 1a.2a

#	Attore	Sistema
1a.2a.1	Termina l'evento	Segna l'evento come terminato
1a.2a.2	Opzionalmente aggiunge o modifica note	Aggiunge le note alla scheda evento
	<i>Termina il caso d'uso</i>	

### Estensione 1a.2b

#	Attore	Sistema
1a.2b.1	Elimina l'evento	Elimina l'evento
	<i>Termina il caso d'uso</i>	

### Estensione 1b

#	Attore	Sistema
1b.1	Apri evento terminato	Fornisce la scheda evento
1b.2	Opzionalmente aggiunge/modifica note	Aggiunge le note alla scheda evento
	<i>Termina il caso d'uso</i>	

### Estensione 1c

#	Attore	Sistema
1c.1	Apri evento annullato	Fornisce la scheda evento
	<i>Termina il caso d'uso</i>	

### Estensione 3a

#	Attore	Sistema
3a.1	Modifica la periodicità degli eventi	Modifica la periodicità di tutti gli eventi precedentemente programmati

### Estensione 4a

#	Attore	Sistema
4a.1	Modifica i dettagli di un servizio	Modifica i dettagli del servizio selezionato

### Estensione 4b

#	Attore	Sistema
4b.1	Annulla un servizio	Segna il servizio come annullato

### Estensione 4c

#	Attore	Sistema
4c.1	Elimina un servizio	Rimuove il servizio selezionato

### Estensione 6a

#	Attore	Sistema
6a.1	Propone modifiche al menù	Aggiunge una proposta di modifica al servizio

### Estensione 7a

#	Attore	Sistema
7a.1	Rimuove personale(di sala, cuochi di servizio)	Registra le modifiche

### Eccezione 1a

#	Attore	Sistema
1a.1	Crea un nuovo evento specificando nome, data d'inizio e cliente	La data d'inizio è già passata

### Eccezione 1b

#	Attore	Sistema
1b.1	Crea un nuovo evento specificando nome, data d'inizio e cliente	Esiste già una scheda evento con nome, data d'inizio e cliente specificati

### Eccezione 1a.1a

#	Attore	Sistema
1a.1a.1	Apri evento esistente	L'attore non è l'organizzatore dell'evento specificato

### Eccezione 1a.1b

#	Attore	Sistema
1a.1b.1	Apri evento esistente	L'evento specificato è terminato/annullato

### Eccezione 1a.2b.1a

#	Attore	Sistema
1a.2b.1a.1	Elimina l'evento	L'evento specificato è confermato

### Eccezione 1b.1a

#	Attore	Sistema
1b.1a.1	Apri evento terminato o annullato	L'evento specificato non è terminato

### Eccezione 1b.1b

#	Attore	Sistema
1b.1b.1	Apri evento terminato o annullato	L'attore non è l'organizzatore dell'evento specificato

### Eccezione 1c.1a

#	Attore	Sistema
1b.1a .1	Apri evento terminato o annullato	L'evento specificato non è annullato

### Eccezione 1c.1b

#	Attore	Sistema
1b.1b .1	Apri evento terminato o annullato	L'attore non è l'organizzatore dell'evento specificato

### Eccezione 3a

#	Attore	Sistema
3a.1	Opzionalmente compila i dettagli dell'evento	L'evento specificato è in corso

### Eccezione 3b

#	Attore	Sistema
3b.1	Opzionalmente compila i dettagli dell'evento	La data specificata è già passata

### Eccezione 3a.1a

#	Attore	Sistema
3a.1a.1	Modifica la periodicità degli eventi	La data di fine è già passata

### Eccezione 4a

#	Attore	Sistema
4a .1	Opzionalmente aggiunge un servizio e i suoi dettagli	La data specificata del servizio è già passata

### Eccezione 4a.1a

#	Attore	Sistema
4a.1a .1	Modifica i dettagli di un servizio	La data specificata è già passata

### Eccezione 4a.1b

#	Attore	Sistema
4a.1b .1	Modifica i dettagli di un servizio	Il servizio è confermato

### Eccezione 4a.1c

#	Attore	Sistema
4a.1c .1	Modifica i dettagli di un servizio	Il servizio è stato annullato o è terminato

### Eccezione 4b.1a

#	Attore	Sistema
4b.1a .1	Annulla un servizio	Il servizio è stato già annullato o terminato

### Eccezione 4c.1a

#	Attore	Sistema
4c.1a .1	Elimina un servizio	Il servizio è confermato

### Eccezione 4c.1b

#	Attore	Sistema
4c.1b .1	Elimina un servizio	Il servizio è stato annullato o è terminato

### Eccezione 6a

#	Attore	Sistema
6a.1	Opzionalmente conferma un menù dello chef per un servizio	Il servizio associato è stato annullato/terminato

### Eccezione 6b

#	Attore	Sistema
6b.1	Opzionalmente conferma un menù dello chef per un servizio	Il servizio è stato già confermato

### Eccezione 6c

#	Attore	Sistema
6c.1	Opzionalmente conferma un menù dello chef per un servizio	Il servizio non ha ancora un menù associato

### Eccezione 6a.1a

#	Attore	Sistema
6a.1a .1	Propone modifiche al menù	Il menù è già stato confermato per il servizio

### Eccezione 6a.1b

#	Attore	Sistema
6a.1b .1	Propone modifiche al menù	Il servizio associato è stato annullato/terminato

### Eccezione 7a

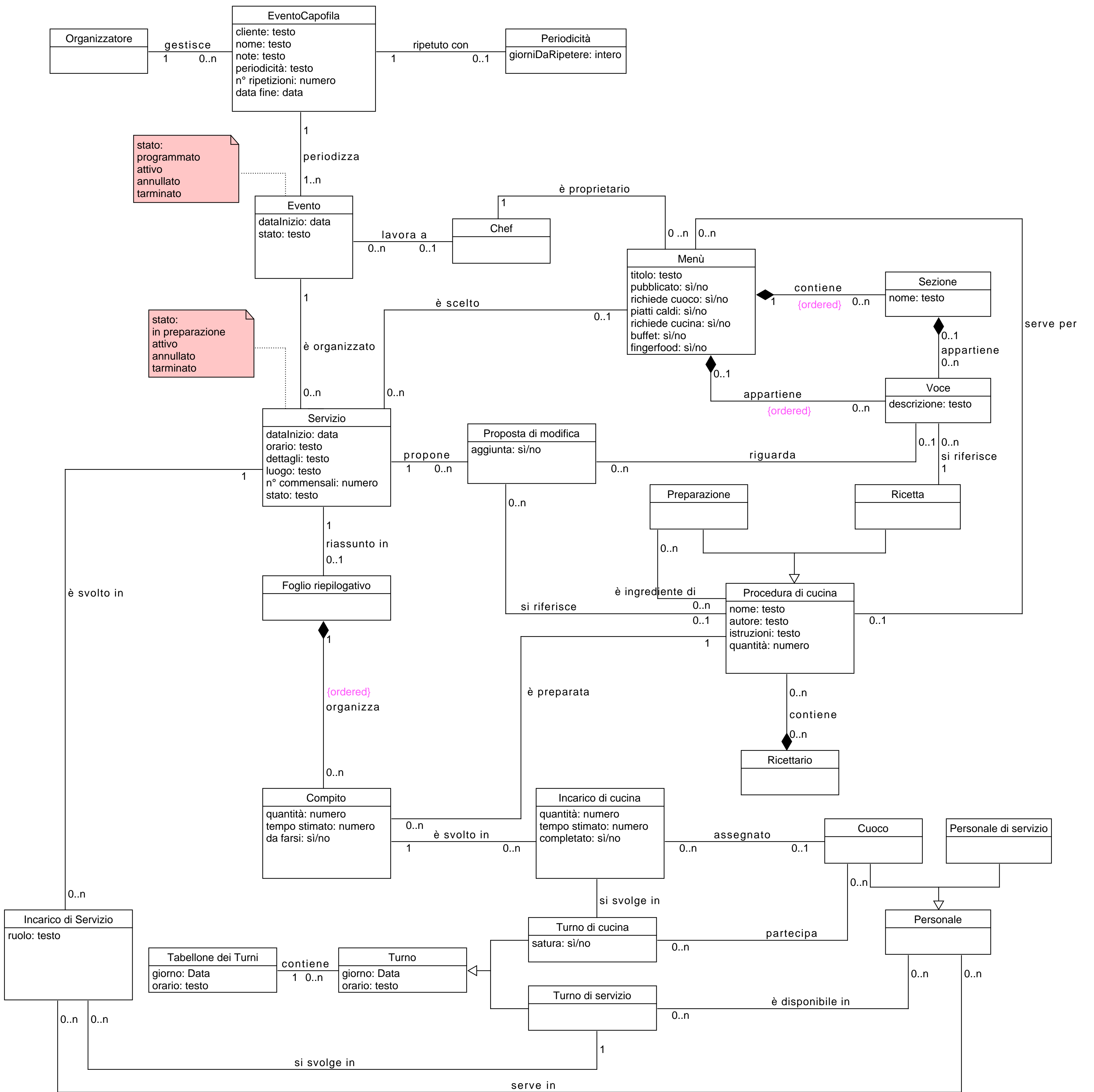
#	Attore	Sistema
7a.1	Opzionalmente sceglie il personale (di sala, cuochi per servizio) e specifica il ruolo per il turno	Il servizio associato è stato annullato/terminato

### Eccezione 7a.1a

#	Attore	Sistema
7a.1a.1	Rimuove personale(di sala, cuochi di servizio)	Il servizio associato è stato annullato/terminato

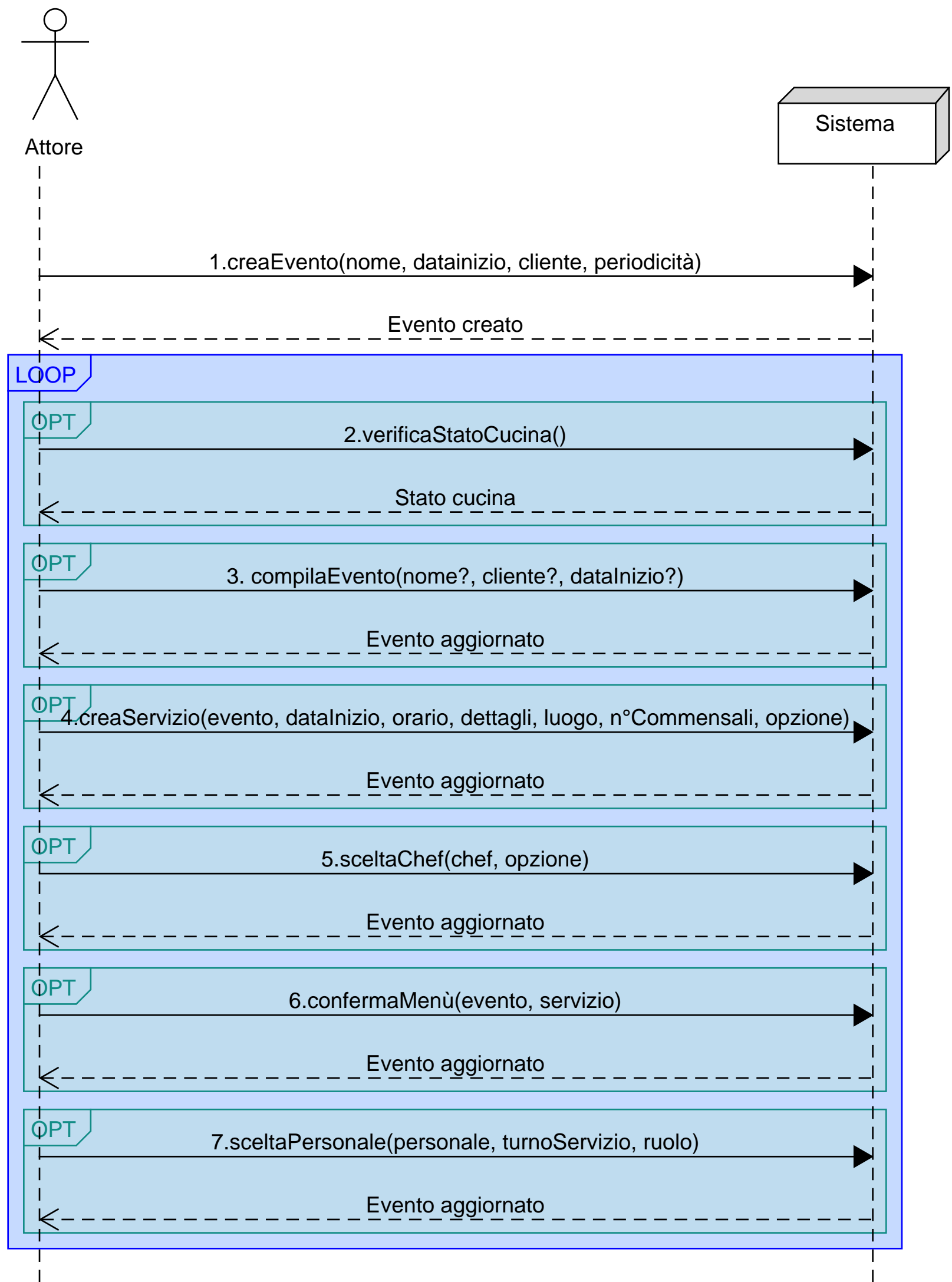


MODELLO DI DOMINIO

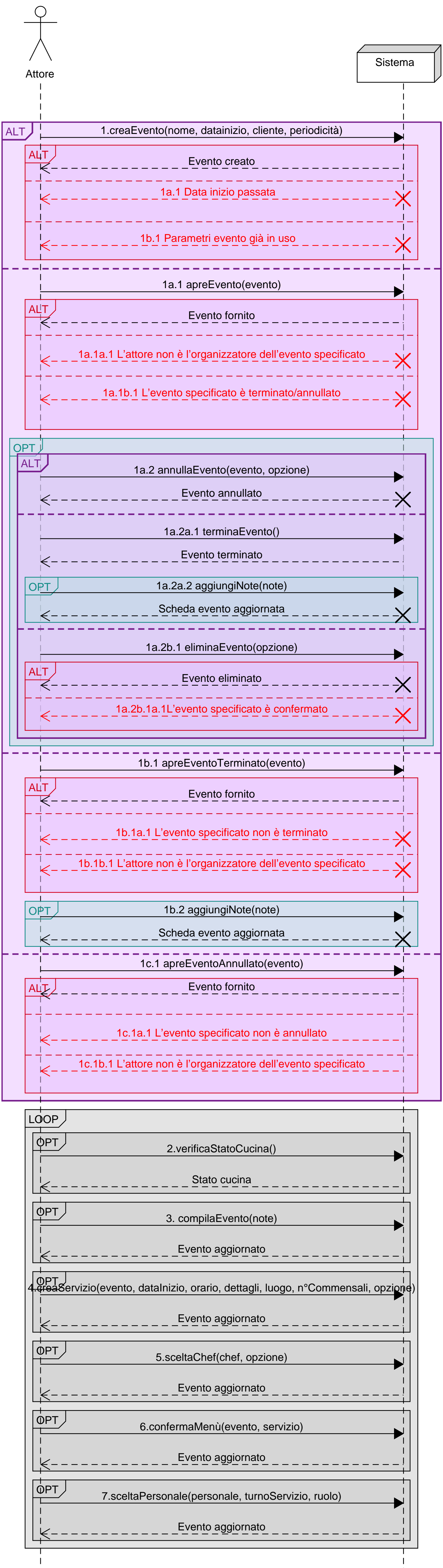


# Diagrammi di sequenza di sistema per l'UC Gestire compiti di cucina

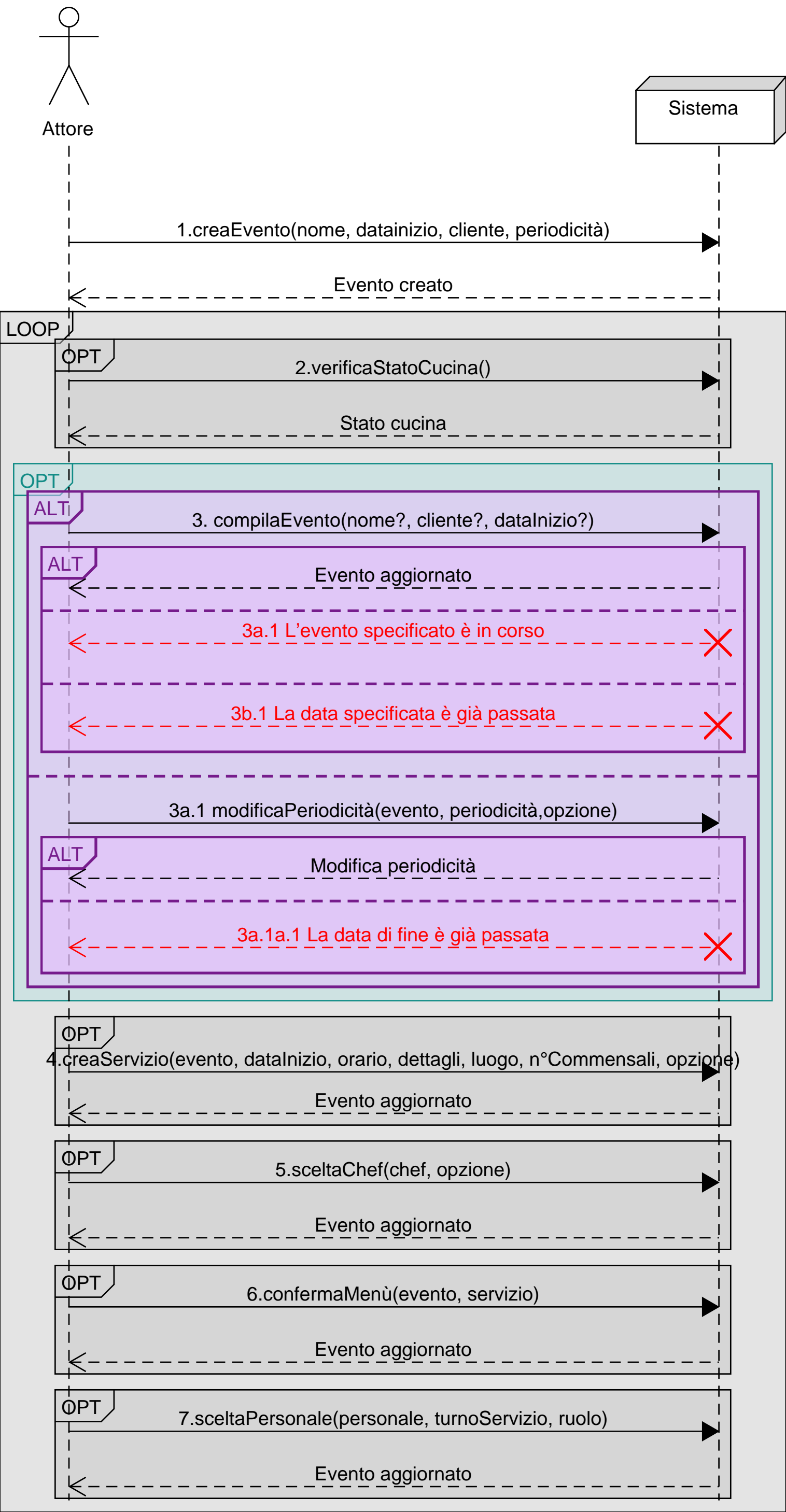
## Scenario principale



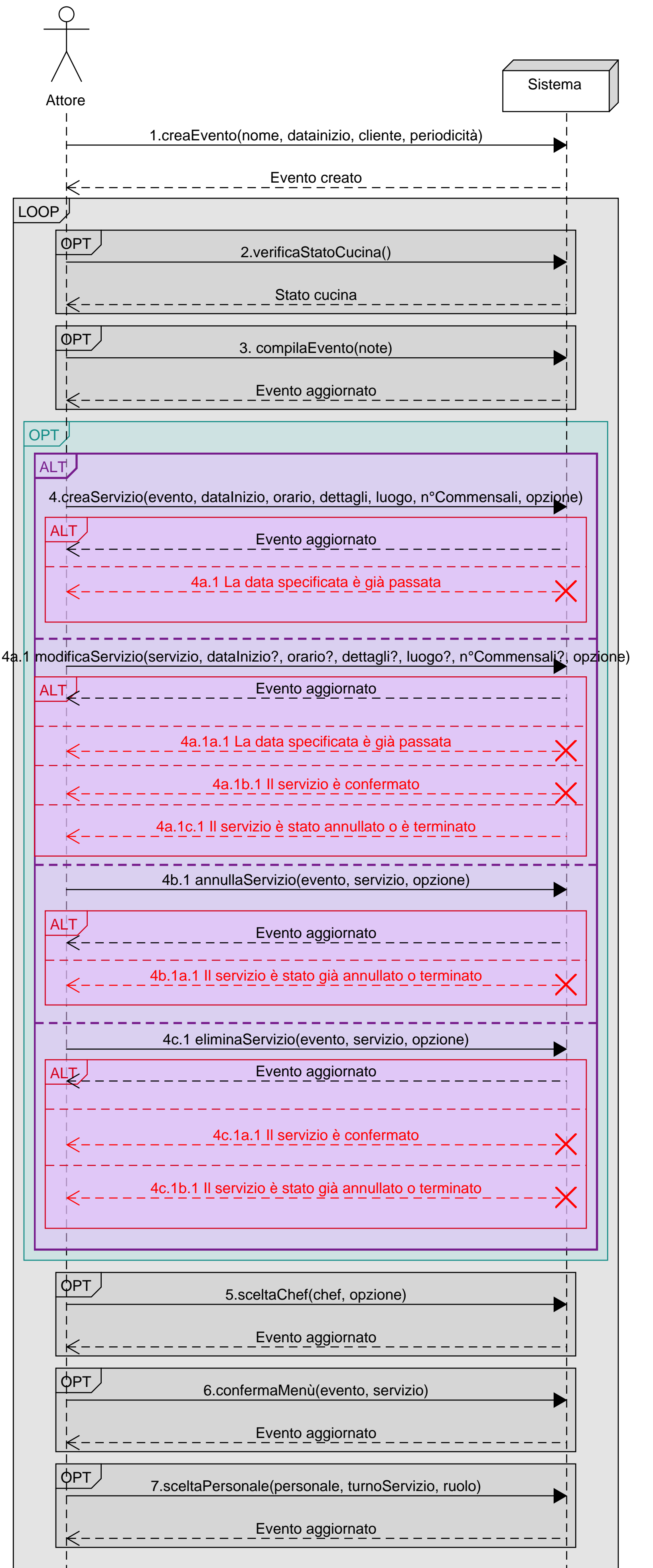
SSD Estensioni al passo 1 (1a, 1b, 1c) e relative eccezioni(1a, 1b, 1a.1a, 1a.1b, 1a.2a.1a, 1b.1a, 1b.1b, 1c.1a, 1c.1b)



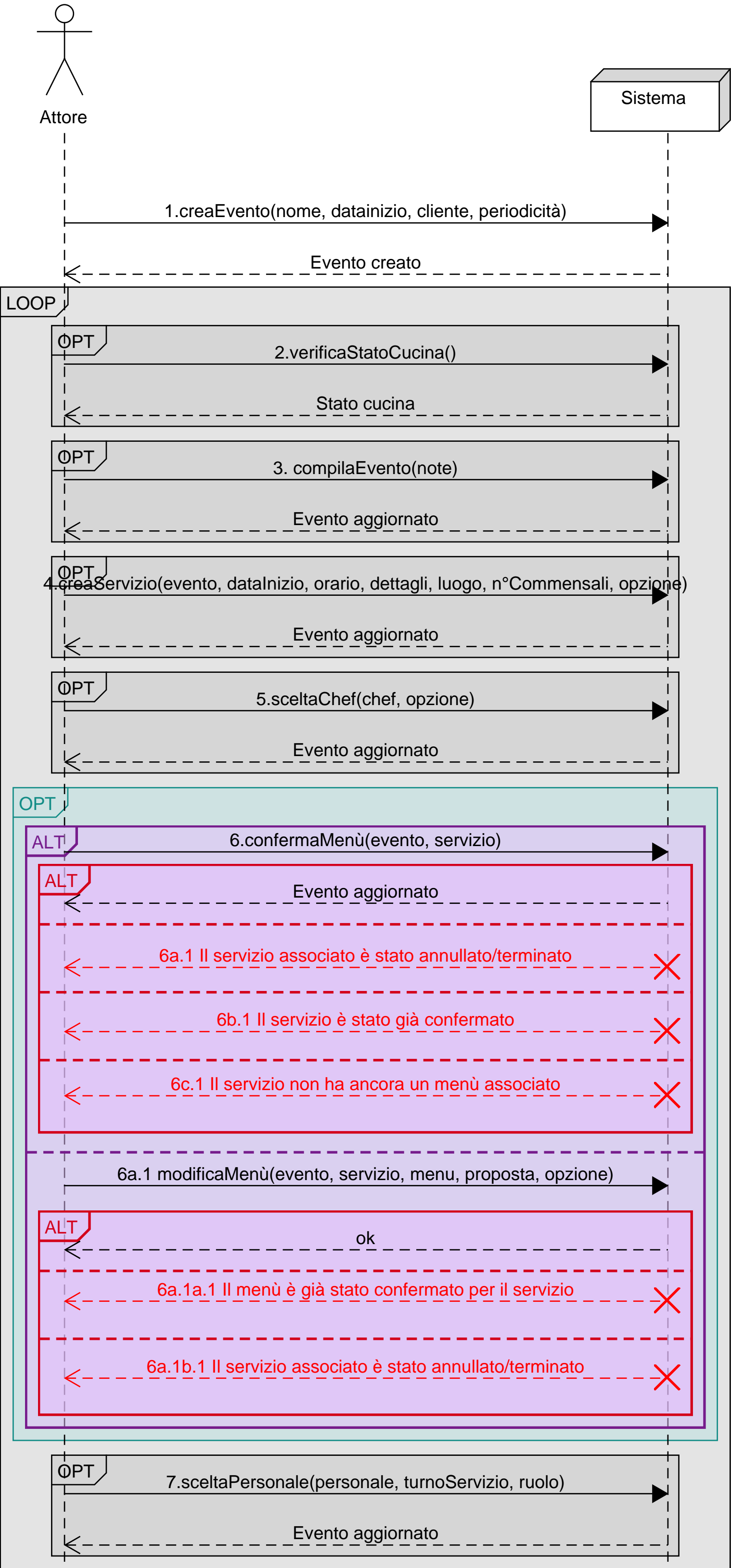
# SSD Estensioni al passo 3 (3a) e relative eccezioni (3a, 3b, 3a.1a)



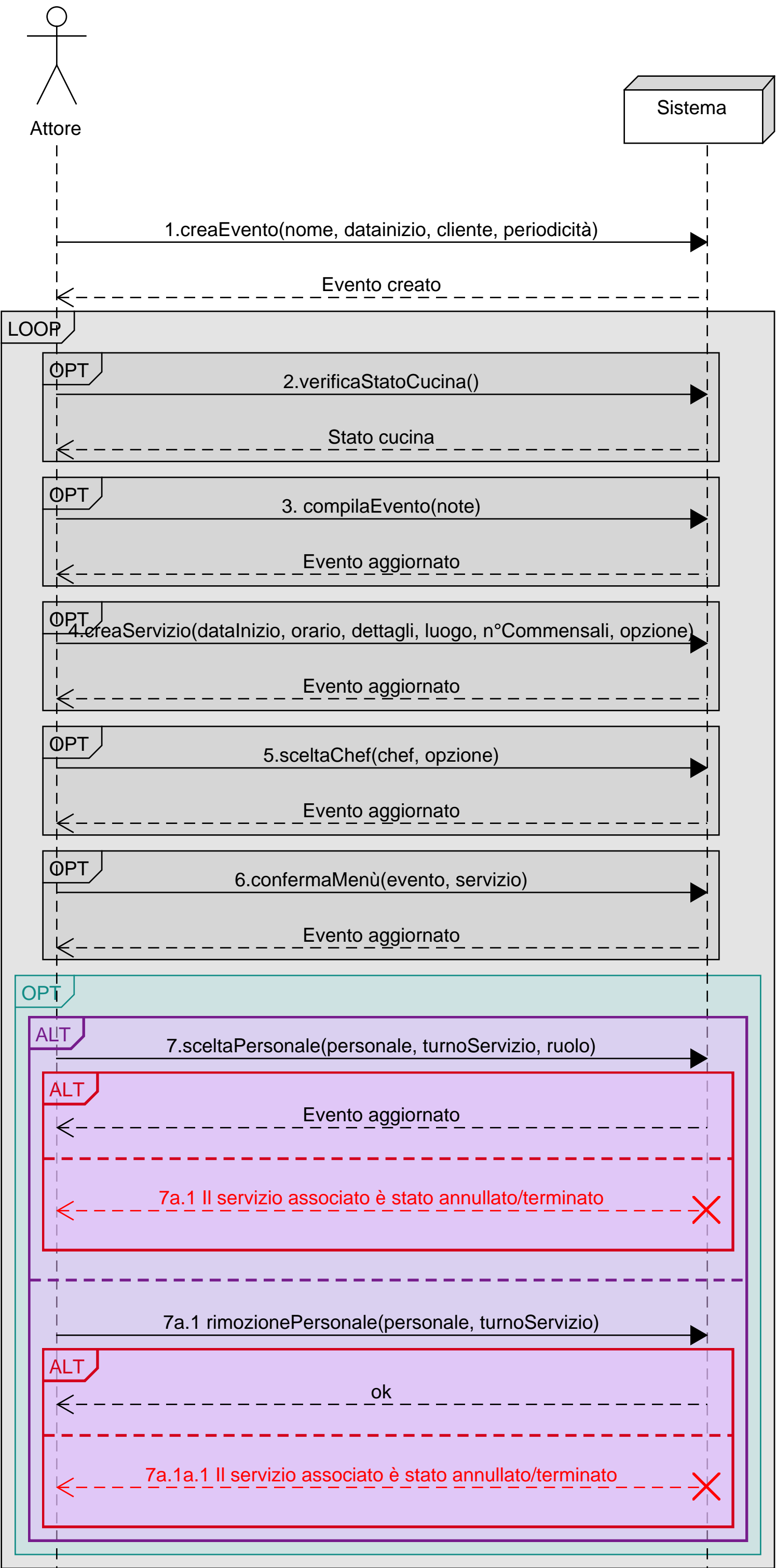
SSD Estensioni al passo 4 (4a, 4b, 4c) e relative eccezioni (4a, 4a.1a, 4a.1b, 4a.1c, 4b.1a, 4c.1a, 4c.1b)



# SSD Estensioni al passo 6 (6a) e relative eccezioni (6a, 6b, 6c, 6a.1a, 6a.1b)



# SSD Estensioni al passo 7 (7a) e relative eccezioni (7a, 7a.1a)



# Contratti per lo UC “Gestire gli eventi”

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

## 1. creaEvento(nome: testo, cliente: Cliente, dataInizio: testo, periodicità?: Periodicità):

**Pre-condizioni:** ---

**Post-condizioni:** se la dataInizio non è ancora passata:

- viene creata un'istanza di *ec* di EventoCapofila
- *ec.cliente* = cliente
- *ec.nome* = nome
- *ec.periodicità* = periodicità
- è stata creata l'associazione **gestisce** tra *org* e *ec*
- vengono create tante istanze *eventoF* di Evento in base a quanto specificato in periodicità

### 1a.1 apriEvento(evento: Evento):

**Pre-condizioni:** ---

**Post-condizioni:** Se evento.stato = programmato/attivo e *org* **gestisce** evento

- è stata aperta la scheda evento

### 1a.2 annullaEvento(evento: Evento, opzione: Testo):

**Pre-condizioni:** evento è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:** se *evento*.stato è attivo:

- *evento*.stato = annullato
- per ogni Servizio *ser* in cui è **organizzato** *evento*:
  - ogni istanza *is* Incarico di Servizio che è **svolto in** *ser* viene eliminata
  - ogni *ic* Incarico di Cucina che **svolge** *c* Compito **organizzato** dall'istanza *foglio* di Foglio Riepilogativo che **riassume** *s*, viene eliminato
- [se opzione = “questo evento e quelli successivi”] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2*.stato = programmato e che *evento*.dataInizio ≤ *ev2*.dataInizio
- [se opzione = “tutti gli eventi”] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2*.stato = programmato

#### 1a.2a.1 terminaEvento():

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento*

**Post-condizioni:** se *evento*.stato è attivo:

- *evento*.stato = terminato



### 1a.2a.2 aggiungiNote(note: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:** se *evento.stato* è terminato:

- *ec.note* = *note*

### 1a.2b.1 eliminaEvento(opzione: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:** se *evento.stato* è programmato:

- per ogni Servizio *s* in cui è **organizzato** *evento*:
  - ogni istanza *is* Incarico di Servizio che è **svolto in** *s* viene eliminata
  - il *foglio* Foglio Riepilogativo che **riassume** *s* viene eliminato
    - per ogni istanza Compito *c* che *foglio* **organizza**:
      - ogni *ic* Incarico di Cucina che **svolge** *c* viene eliminato
    - *c* viene eliminato
- *s* viene eliminato
- l'istanza *evento* viene eliminata
- [se opzione = "questo evento e quelli successivi"] le operazioni effettuate sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
- [se opzione = "tutti gli eventi"] le operazioni effettuate sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

### 1b.1 apreEventoTerminato(evento: Evento):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento*

**Post-condizioni:** se *evento.stato* è terminato:

- è stata aperta la scheda evento

### 1b.2 aggiungiNote(note: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:**

- *ec.note* = *note*

### 1c.1 apreEventoAnnullato(evento: Evento):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento*

**Post-condizioni:** se *evento.stato* è annullato:

- è stata aperta la scheda evento

### 2. verificaStatoCucina():

**Pre-condizioni:** ---

**Post-condizioni:** ---

### 3. compilaEvento(nome?: Testo, cliente?: Cliente, dataInizio?: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:** se dataInizio non è ancora passata e se *evento.stato* è programmato:

- [se è specificata dataInizio] *evento.dataInizio* = dataInizio
- [se è specificato il cliente] *ec .cliente* = cliente

### 3a.1 modificaPeriodicità(evento: Evento, periodicità: Periodicità, opzione: testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:**

- *ec.periodicità* = periodicità
- vengono create tante istanze *eventoF* di Evento in base a quanto specificato in periodicità
- [se opzione = "questo evento e quelli successivi"]: tutte le istanze di Evento *eventoObsoleto* che sono ricorrenza di *ec* , che non corrispondono più alla nuova periodicità e hanno *eventoObsoleto.stato*=programmato e *eventoObsoleto.dataInizio*<=*evento.dataInizio* vengono eliminate
- [se opzione = "tutti gli eventi"]: tutte le istanze di Evento *eventoObsoleto* che sono ricorrenza di *ec* , che non corrispondono più alla nuova periodicità e hanno *eventoObsoleto.stato*=programmato vengono eliminate

#### **4. creaServizio(evento: Evento, dataInizio: Testo, orario: Testo, dettagli?: testo, luogo: Testo, n°Commensali: numero, opzione: testo):**

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:** se dataInizio non è ancora passata:

- Viene creata una nuova istanza *s* di Servizio:
  - *s.dataInizio* = dataInizio
  - *s.orario* = orario
  - [se è specificata dettagli] *s.dettagli* = dettagli
  - *s.luogo* = luogo
  - *s.n°Commensali* = n°Commensali
  - *s.stato* = in preparazione
- Viene creata la relazione è **organizzato** tra *evento* e *s*
- [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
- [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

#### **4a.1 modificaServizio(servizio: Servizio, dataInizio?: Testo, orario?: Testo, dettagli?: testo, luogo?: Testo, n°Commensali?: numero, opzione: testo):**

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:**

- se dataInizio non è ancora passata, se servizio.stato non è attivo, annullato o terminato:
  - [se è specificata dataInizio] servizio.dataInizio = dataInizio
  - [se è specificata orario] servizio.orario = orario
  - [se è specificata dettagli] servizio.dettagli = dettagli
  - [se è specificata luogo] servizio.luogo = luogo
  - [se è specificata n°Commensali] servizio.n°Commensali = n°Commensali
- se servizio.stato è attivo:
  - se non è specificato nessun parametro opzionale eccetto n°Commensali e n°Commensali < (servizio.n°Commensali + 30%) e n°Commensali > (servizio.n°Commensali - 30%):
    - servizio.n°Commensali = n°Commensali
  - [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
  - [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

## 4b.1 annullaServizio(evento: Evento, servizio: Servizio, opzione: testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:** se servizio.stato non è annullato o terminato:

- servizio.stato = annullato
- per ogni istanza *com* Compito **organizzato** in *foglio* Foglio Riepilogativo che **riassume** servizio:
  - ogni istanza *ic* Incarico di Cucina che **svolge** *com* viene eliminata
- ogni istanza *is* Incarico di Servizio che **svolge** servizio viene eliminata
- [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da ec**, che abbia *ev2*.stato = programmato e che *evento*.dataInizio <= *ev2*.dataInizio
- [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da ec**, che abbia *ev2*.stato = programmato

## 4c.1 eliminaServizio(evento: Evento, servizio: Servizio, opzione: testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:** se servizio.stato non è annullato o terminato o attivo:

- ogni istanza *is* Incarico di Servizio che **svolge** servizio viene eliminata
- per ogni istanza *foglio* Foglio Riepilogativo che **riassume** servizio:
  - per ogni istanza *c* Compito **organizzato** da *foglio*:
    - ogni *ic* Incarico di Cucina che **svolge** *c* viene eliminato
  - *c* viene eliminato
- *foglio* viene eliminato
- servizio viene eliminato
- [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da ec**, che abbia *ev2*.stato = programmato e che *evento*.dataInizio <= *ev2*.dataInizio
- [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da ec**, che abbia *ev2*.stato = programmato

## 5. sceltaChef(chef: Chef, opzione: testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:**

- chef **lavora a** un *evento*
- [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
- [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

## 6. confermaMenù(evento: Evento, servizio: Servizio):

**Pre-condizioni:** : è in corso la visualizzazione di un Evento *evento* che è **suddiviso** in servizio

**Post-condizioni:** se servizio.stato non è annullato o terminato o attivo e un Menu è **scelto per** servizio:

- servizio.stato = attivo
- *evento.stato* = attivo

### 6a.1 modificaMenù(evento: Evento, servizio: Servizio, menu: Menu, proposta: Proposta di Modifica, opzione: testo):

**Pre-condizioni:** : è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

**Post-condizioni:** se servizio.stato non è annullato o terminato o attivo e un Menu è **scelto per** servizio:

- Viene creata l'associazione **propone** tra proposta e servizio
- [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
- [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

## **7. sceltaPersonale(personale: Personale, turnoServizio: TurnoServizio, ruolo?: Ruolo):**

**Pre-condizioni:** : è in corso la visualizzazione di un Evento *evento* che è **suddiviso** in *servizio* Servizio

**Post-condizioni:** se servizio.stato non è annullato o terminato:

- viene creato un *is* Incarico di Servizio e:
  - [se ruolo è specificato] *is.ruolo* = ruolo
  - viene creata l'associazione **si svolge in** tra *is* e turnoServizio
  - viene creata l'associazione **serve in** tra personale *is*
- viene creata l'associazione **è svolto in** tra *is* e *servizio*

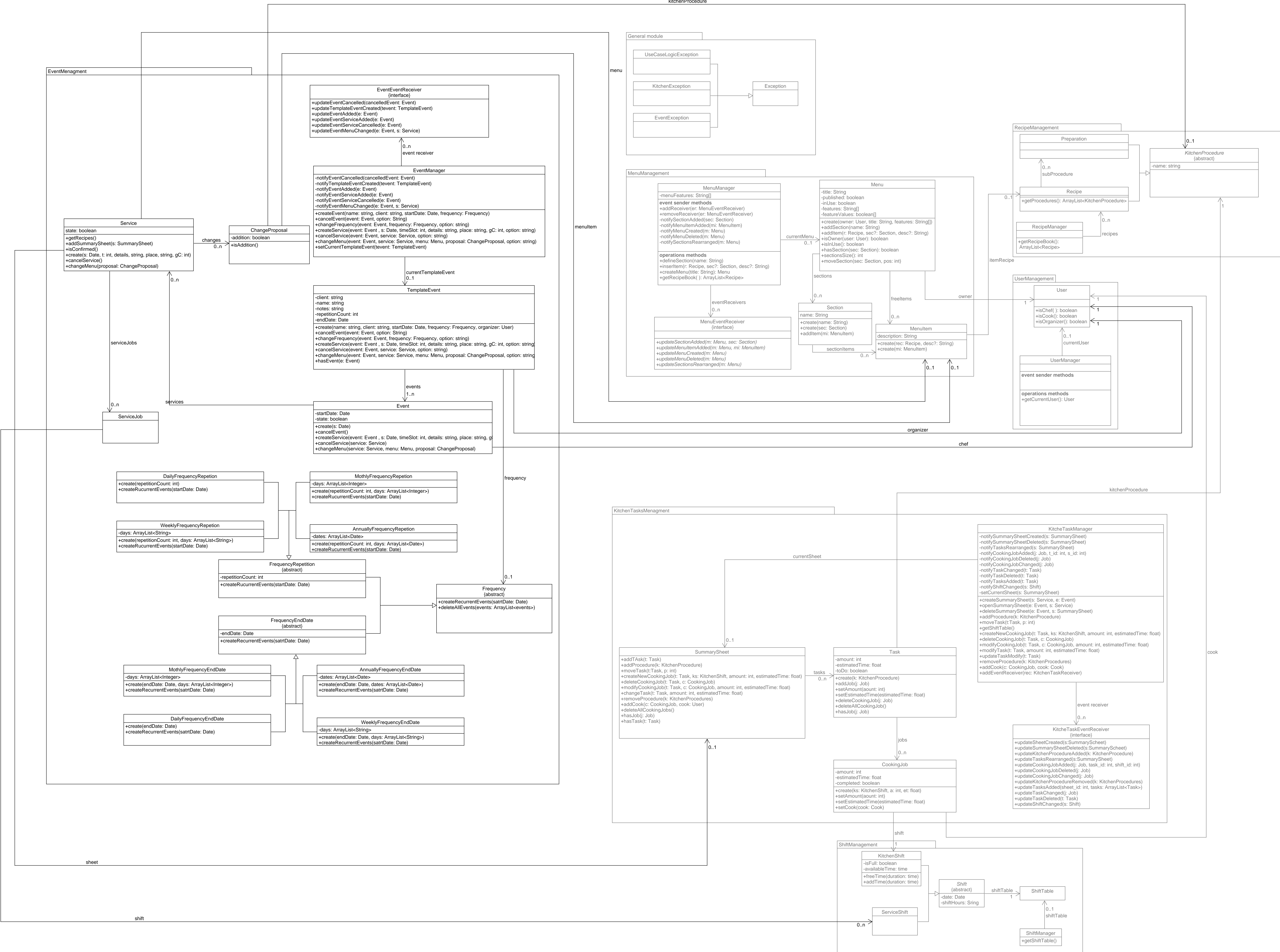
### **7a.1 rimuoviPersonale(personale: Personale, turnoServizio: TurnoServizio):**

**Pre-condizioni:** : è in corso la visualizzazione di un Evento *evento* che è **suddiviso** in *servizio* Servizio

**Post-condizioni:** se servizio.stato non è annullato o terminato:

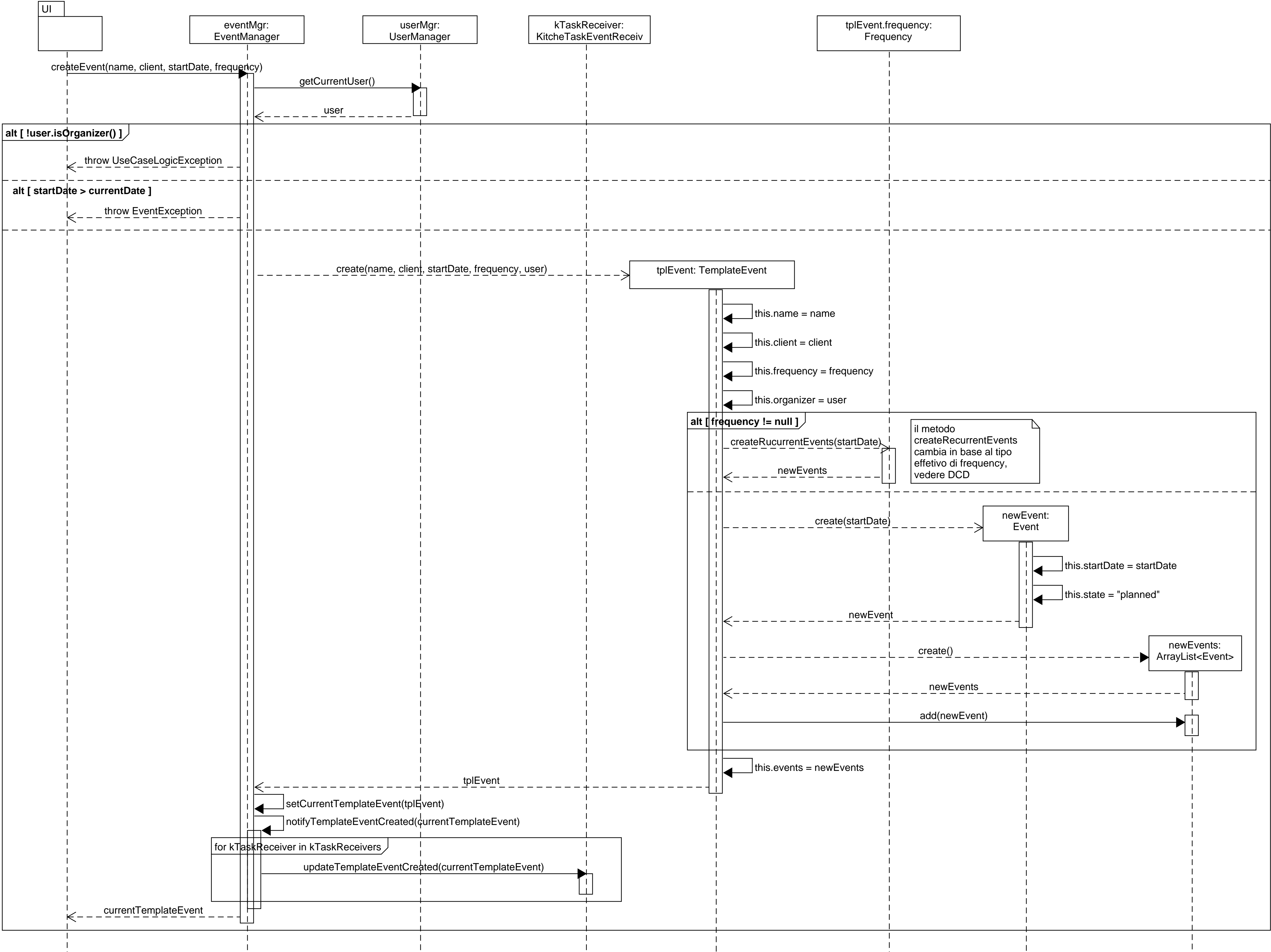
- per ogni *is* Incarico di Servizio **deciso per servizio**:
  - viene eliminata ogni istanza di personale che **serve in** *is*

Design Class Diagram (DCD)



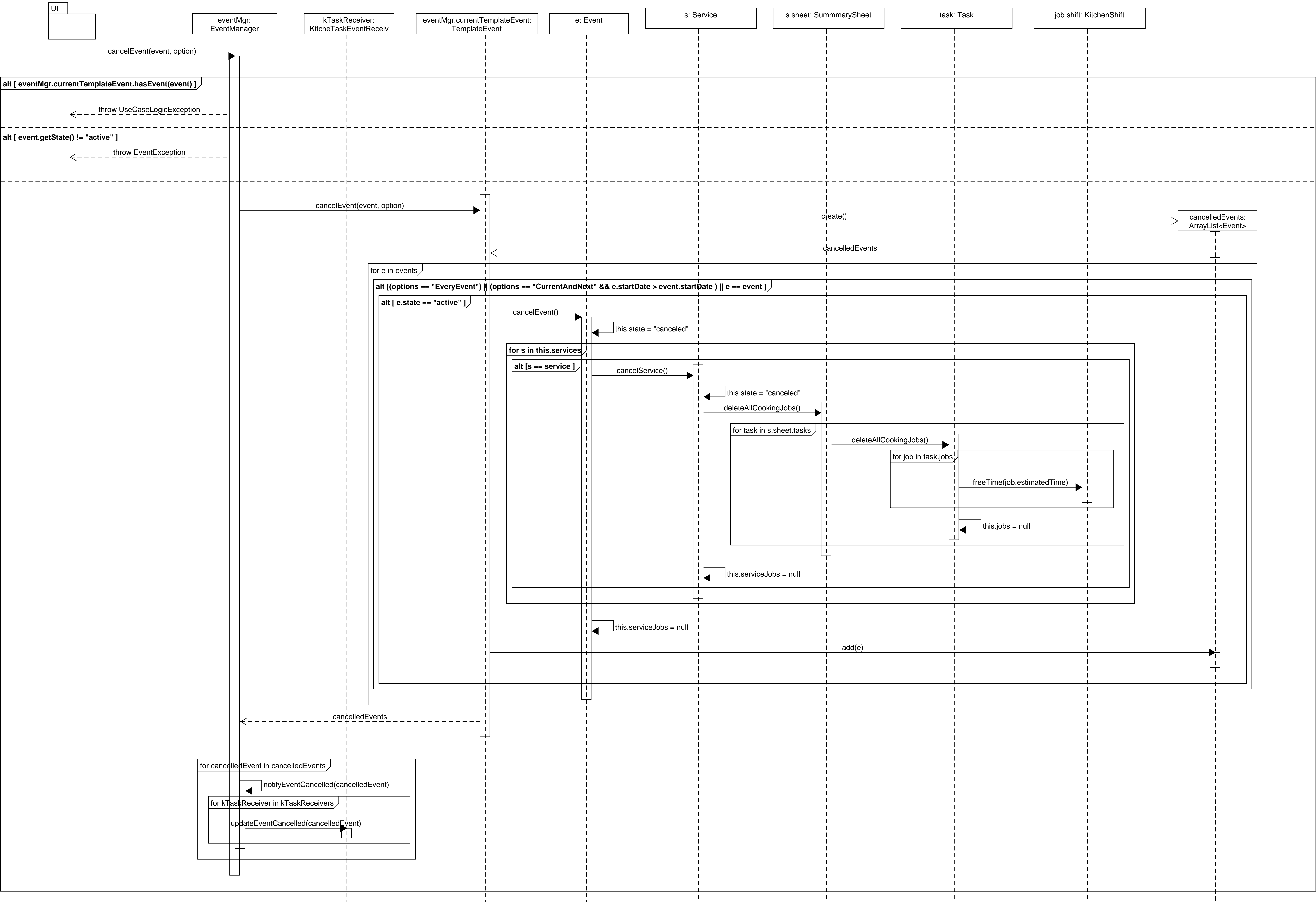
1 - createEvent

currentDate viene ricavata dipendentempente dal linguaggio di programmazione

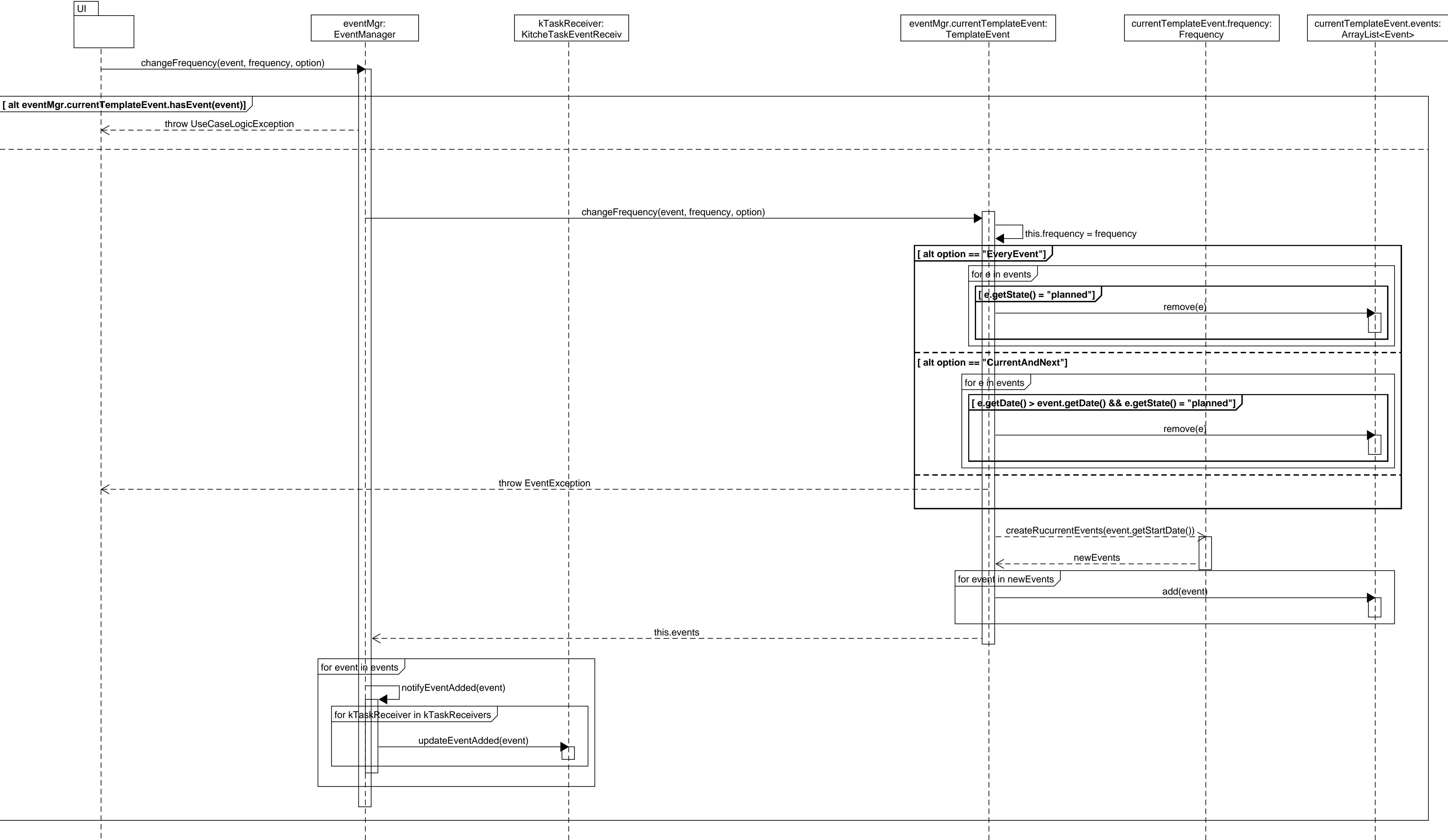




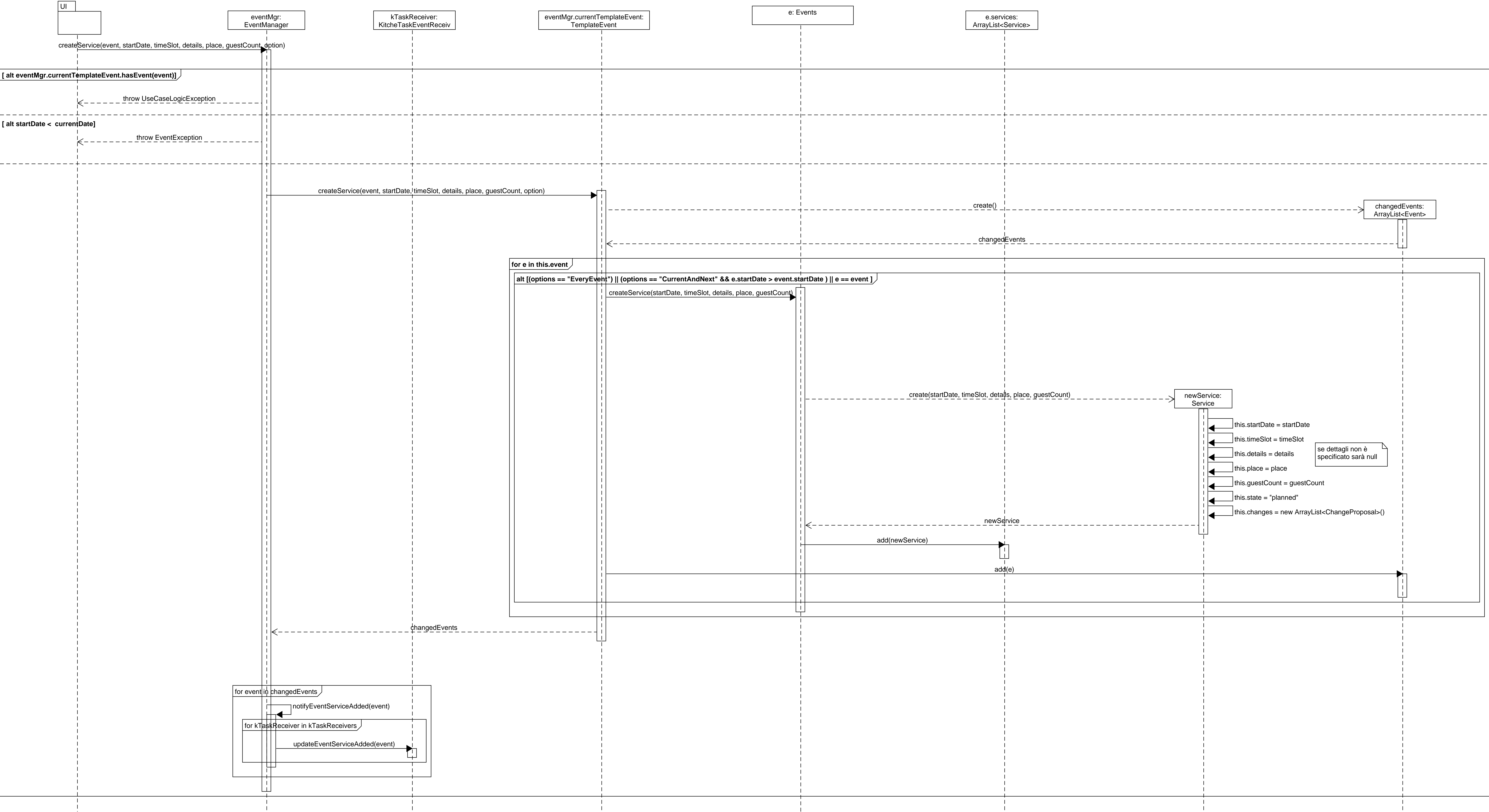
1a.2 cancelEvent



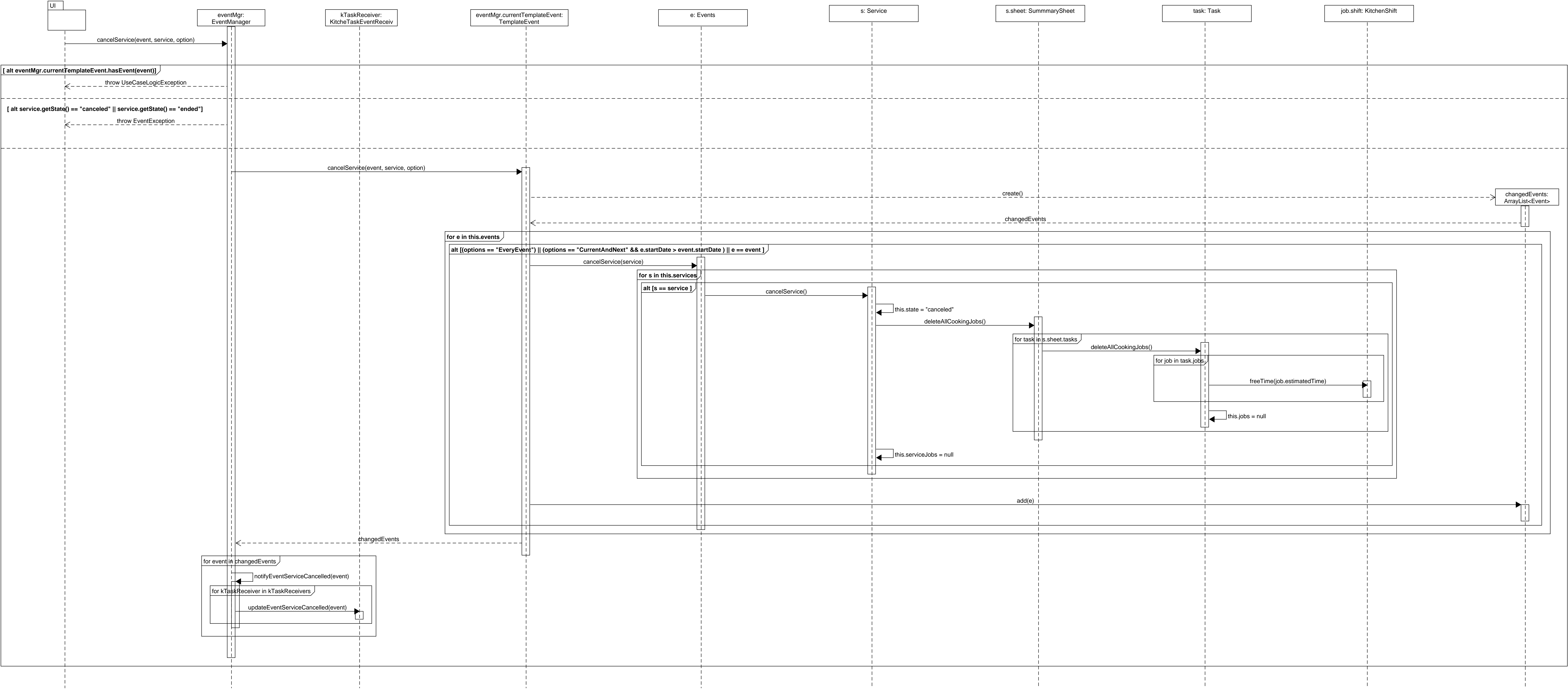
3a changeFrequency



4 createService



4b cancelService



6a changeMenu

