

# Sviluppo delle Applicazioni Software Laboratorio

a.a 2019/2020

Lorenzo Sciandra (859704), Stefano Vittorio Porta (859133)

# Gestire 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, Cliente

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

**Garanzie di successo o post-condizioni:** L'evento viene organizzato assieme ai suoi servizi, vengono scelti lo chef, i cuochi, il personale e i menù per l'evento.

## Scenario principale di successo<sup>1</sup>

#	Attore	Sistema
1	Crea un nuovo evento specificando la data d'inizio e il nome	Registra l'inizio della creazione di un nuovo evento con il nome specificato
2	Opzionalmente aggiunge note relative all'evento <sup>2</sup>	Registra le note riguardo l'evento
3	Opzionalmente verifica lo stato della cucina <sup>3</sup>	Fornisce gli impegni della cucina
4	Opzionalmente modifica i dettagli dell'evento	Registra l'evento con i dettagli specificati
5	Opzionalmente aggiunge un servizio e i suoi dettagli	Registra il nuovo servizio per l'evento che si sta organizzando; applicando le modifiche sulle altre ricorrenze dell'evento, se richiesto
6	Opzionalmente sceglie lo chef	Registra lo chef per l'Evento; applicando le modifiche sulle altre ricorrenze dell'evento, se richiesto
7	Opzionalmente conferma un menù dello chef per un servizio	Registra la conferma; l'evento viene segnato come "in corso/attivo"
8	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 da 2 finchè non soddisfatto</i>	
	<i>Termina il caso d'uso</i>	

<sup>1</sup> Viene creato un nuovo evento partendo da zero

<sup>2</sup> Le note possono comprendere appunti riguardo l'evento o osservazioni, oltre a qualsiasi altro tipo di informazioni.

<sup>3</sup> Verifica quindi quanto è già occupata per altri eventi

## Estensione 1a<sup>4</sup>

#	Attore	Sistema
1a.1	Apri un evento esistente	Visualizza la scheda dell'evento
	<i>Se desidera modificare lo stato dell'evento continua, altrimenti procede al passo 2 dello scenario principale</i>	
1a.2	Opzionalmente annulla l'evento	Registra l'annullamento dell'evento, avvisando della possibilità di una penale. Se desiderato, applica l'annullamento anche ad altre ricorrenze
	<i>Termina il caso d'uso</i>	

## Estensione 1a.2a

#	Attore	Sistema
1a.2a.1	Termina l'evento	Registra la terminazione dell'evento.

## Estensione 1a.2b

#	Attore	Sistema
1a.2b.1	Elimina l'evento	Rimuove l'evento. Se desiderato, applica l'eliminazione anche ad altre ricorrenze
	<i>Termina il caso d'uso</i>	

## Estensione 1b<sup>5</sup>

#	Attore	Sistema
1b.1	Apri un evento annullato o terminato	Visualizza la scheda dell'evento
1b.2	Opzionalmente aggiunge o modifica note	Registra le note
	<i>Termina il caso d'uso</i>	

---

<sup>4</sup> Si apre un evento già esistente, ma che non è stato ancora annullato o terminato. Si può anche gestire lo stato dell'evento.

<sup>5</sup> Si apre un evento terminato/annullato per visionare e/o aggiungere note.

## Estensione 4a

#	Attore	Sistema
4a.1	Modifica la periodicità degli eventi, specificando l'evento da usare come base	Tutti gli eventi programmati sono sostituiti da nuove programmazioni nelle nuove date definite dalla periodicità specificata

## Estensione 5a

#	Attore	Sistema
5a.1	Annulla un servizio	Segna il servizio come annullato e libera i turni dedicati ad esso. Avvisa della possibilità di una penale. Se desiderato, applica l'annullamento anche ad altre ricorrenze

## Estensione 5b

#	Attore	Sistema
5b.1	Elimina un servizio	Rimuove il servizio indicato. Se desiderato, applica l'eliminazione anche ad altre ricorrenze

## Estensione 5c

#	Attore	Sistema
5c.1	Modifica i dettagli di un servizio	Applica al servizio le modifiche effettuate, applicando le modifiche sulle altre ricorrenze dell'evento, se richiesto

## Estensione 7a

#	Attore	Sistema
7a.1	Propone modifiche al menù scelto dallo chef	Registra presso il menù eventuali modifiche, applicando le modifiche sulle altre ricorrenze dell'evento, se richiesto

## Estensione 8a

#	Attore	Sistema
8a.1	Rimuove parte del personale (sala, cuochi servizio)	Registra le modifiche del personale richieste

## Eccezione 1a

#	Attore	Sistema
1a.1	Crea un nuovo evento specificando la data d'inizio	La data d'inizio specificata è già passata

## Eccezione 1a.1a

#	Attore	Sistema
1a.1a.1	Apri un evento esistente	L'evento è terminato o è stato annullato

## Eccezione 1a.1b

#	Attore	Sistema
1a.1b.1	Apri un evento esistente	L'organizzatore non gestisce l'evento indicato

## Eccezione 1a.2b.1a

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

## Eccezione 1b.1a

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

## Eccezione 1b.1b

#	Attore	Sistema
1b.1b.1	Apri un evento annullato o terminato	L'organizzatore non gestisce l'evento indicato

## Eccezione 4a

#	Attore	Sistema
4a.1	Opzionalmente modifica i dettagli dell'evento (periodicità, data di inizio)	La data d'inizio non è modificabile se i lavori sono iniziati.

## Eccezione 4b

#	Attore	Sistema
4b.1	Opzionalmente modifica i dettagli dell'evento (periodicità, data di inizio)	La data d'inizio specificata è già passata

## Eccezione 5a

#	Attore	Sistema
5a.1	Opzionalmente aggiunge un servizio e i suoi dettagli	La data del servizio specificata nei dettagli fa riferimento ad un giorno nel passato

## Eccezione 5a.1a

#	Attore	Sistema
5a.1a.1	Annulla un servizio	Il servizio è stato già annullato o è terminato

## Eccezione 5b.1a

#	Attore	Sistema
5b.1a.1	Elimina un servizio	Il servizio è stato confermato

## Eccezione 5b.1b

#	Attore	Sistema
5b.1b.1	Elimina un servizio	Il servizio è stato annullato o è terminato

## Eccezione 5c.1a

#	Attore	Sistema
5c.1a.1	Modifica i dettagli di un servizio	Il servizio è stato confermato

## Eccezione 5c.1b

#	Attore	Sistema
5c.1b.1	Modifica i dettagli di un servizio	Il servizio è stato annullato o è terminato

## Eccezione 5c.1c

#	Attore	Sistema
5c.1c.1	Modifica i dettagli di un servizio	La data specificata è già passata

## Eccezione 7a

#	Attore	Sistema
7a.1	Opzionalmente conferma il menù dello chef per un servizio	Il servizio è stato già confermato

## Eccezione 7b

#	Attore	Sistema
7b.1	Opzionalmente conferma il menù dello chef per un servizio	Il servizio è stato annullato o è terminato

## Eccezione 7c

#	Attore	Sistema
7c.1	Opzionalmente conferma il menù dello chef per un servizio	Il servizio non ha un menù associato

## Eccezione 7a.1a

#	Attore	Sistema
7a.1a.1	Propone modifiche al menù scelto dallo chef	Il servizio associato al menù è stato annullato o è terminato

## Eccezione 7a.1b

#	Attore	Sistema
7a.1b.1	Propone modifiche al menù scelto dallo chef	Il menù dello chef è già stato confermato per il servizio

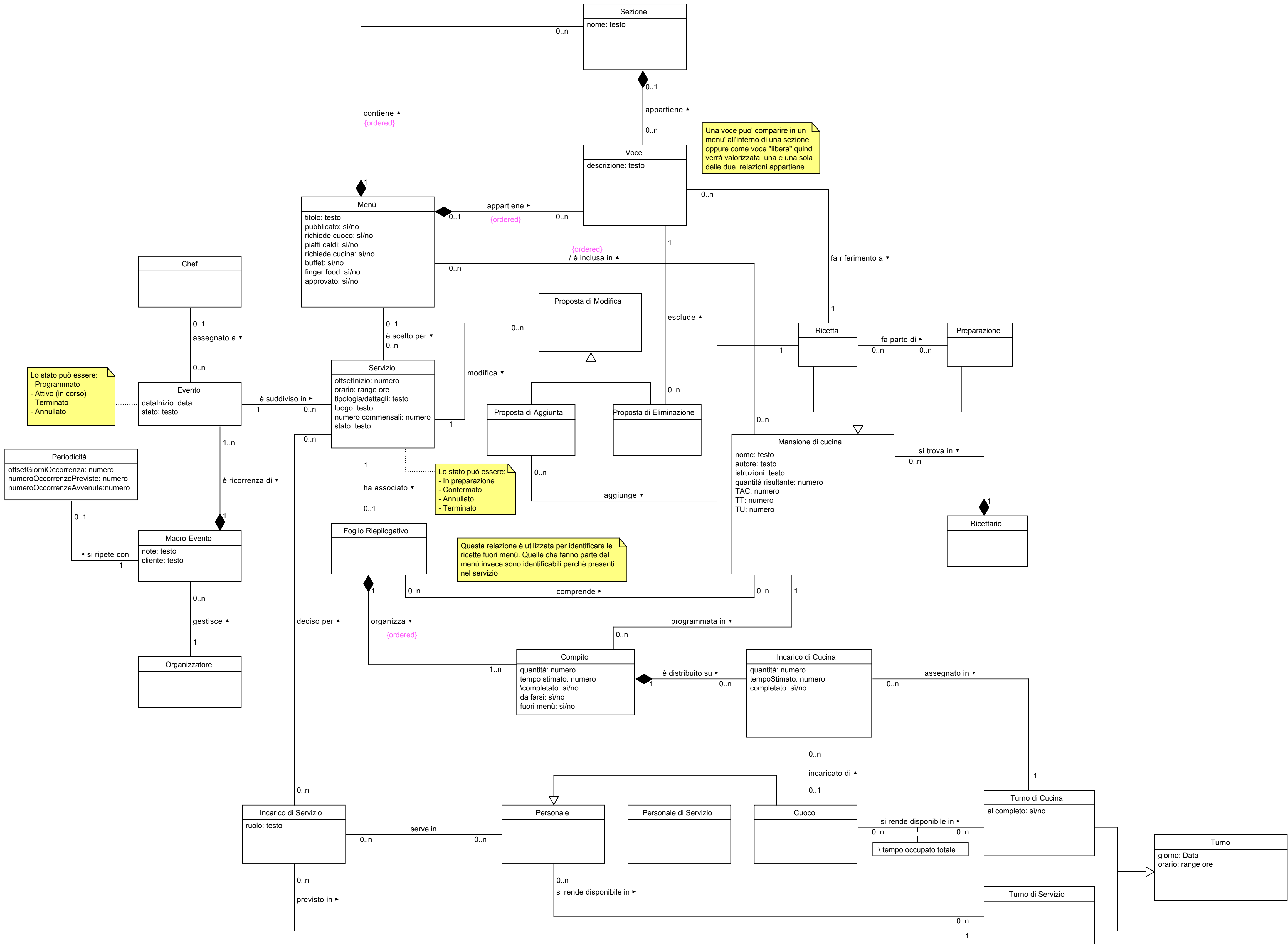
**Eccezione 8a**

#	Attore	Sistema
8a.1	Opzionalmente sceglie il personale (sala, cuochi servizio)	Il servizio è stato annullato o è terminato

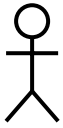
**Eccezione 8a.1a**

#	Attore	Sistema
8a.1a.1	Rimuove parte del personale (sala, cuochi servizio)	Il servizio è stato annullato o è terminato





# Diagrammi di Sequenza di sistema per l'UC Gestire Eventi Scenario principale



Organizzatore

Sistema

1. creaEvento(dataInizio, cliente, nome)

nuovo evento

loop

opt

2. aggiungiNote(note)

ok

opt

3. visualizzaStatoCucina()

stato cucina

opt

4. modificaDettagliEvento(dataInizio?, cliente?)

ok

opt

5. aggiungiServizio(offsetInizio, rangeOrario, numeroCommensali, luogo, dettagli, strategia)

ok

opt

6. assegnaChef(chef, strategia)

ok

opt

7. confermaMenuServizio(servizio)

ok

opt

8. assegnaPersonale(dipendente, turnoServizio, ruolo?)

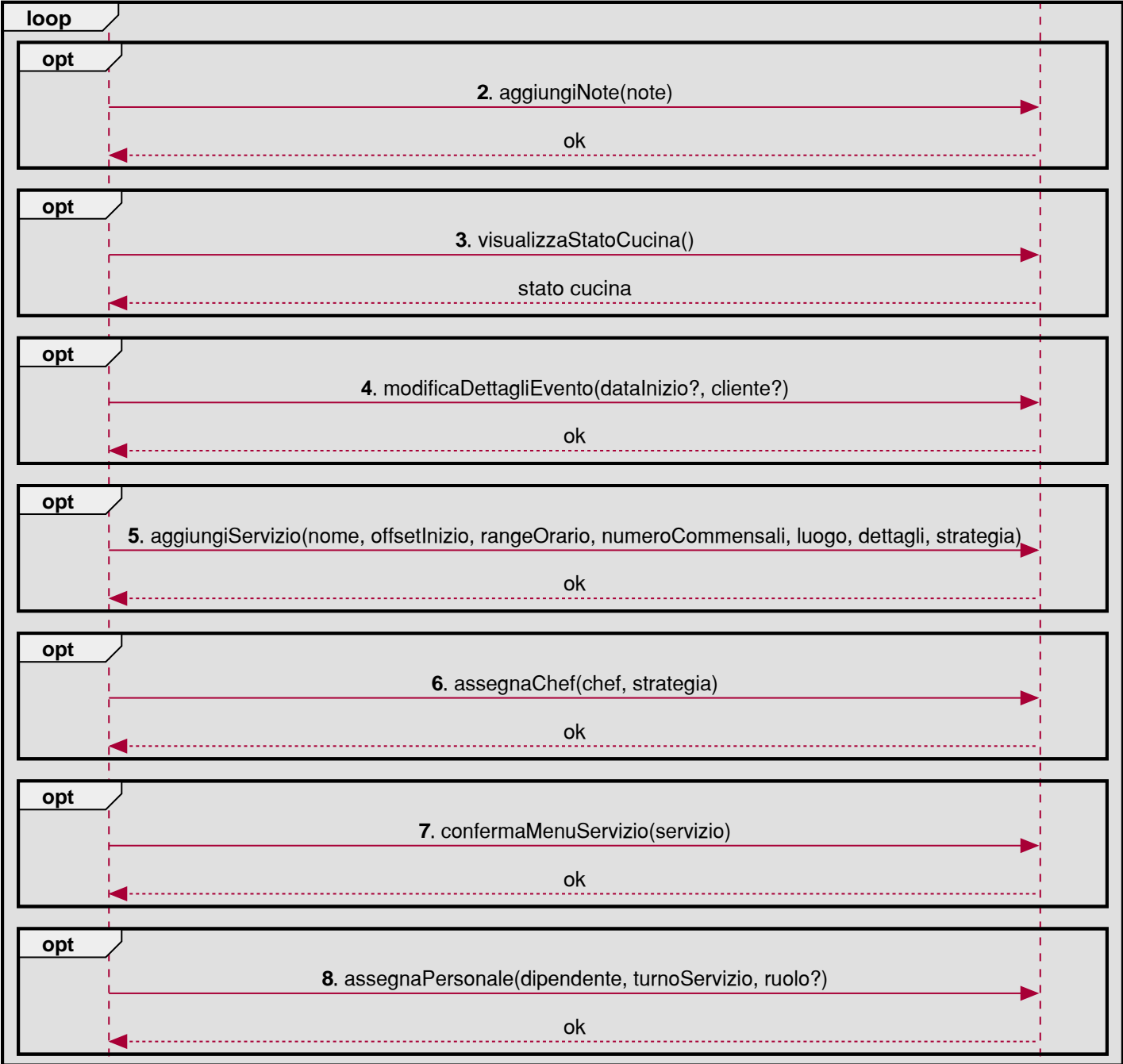
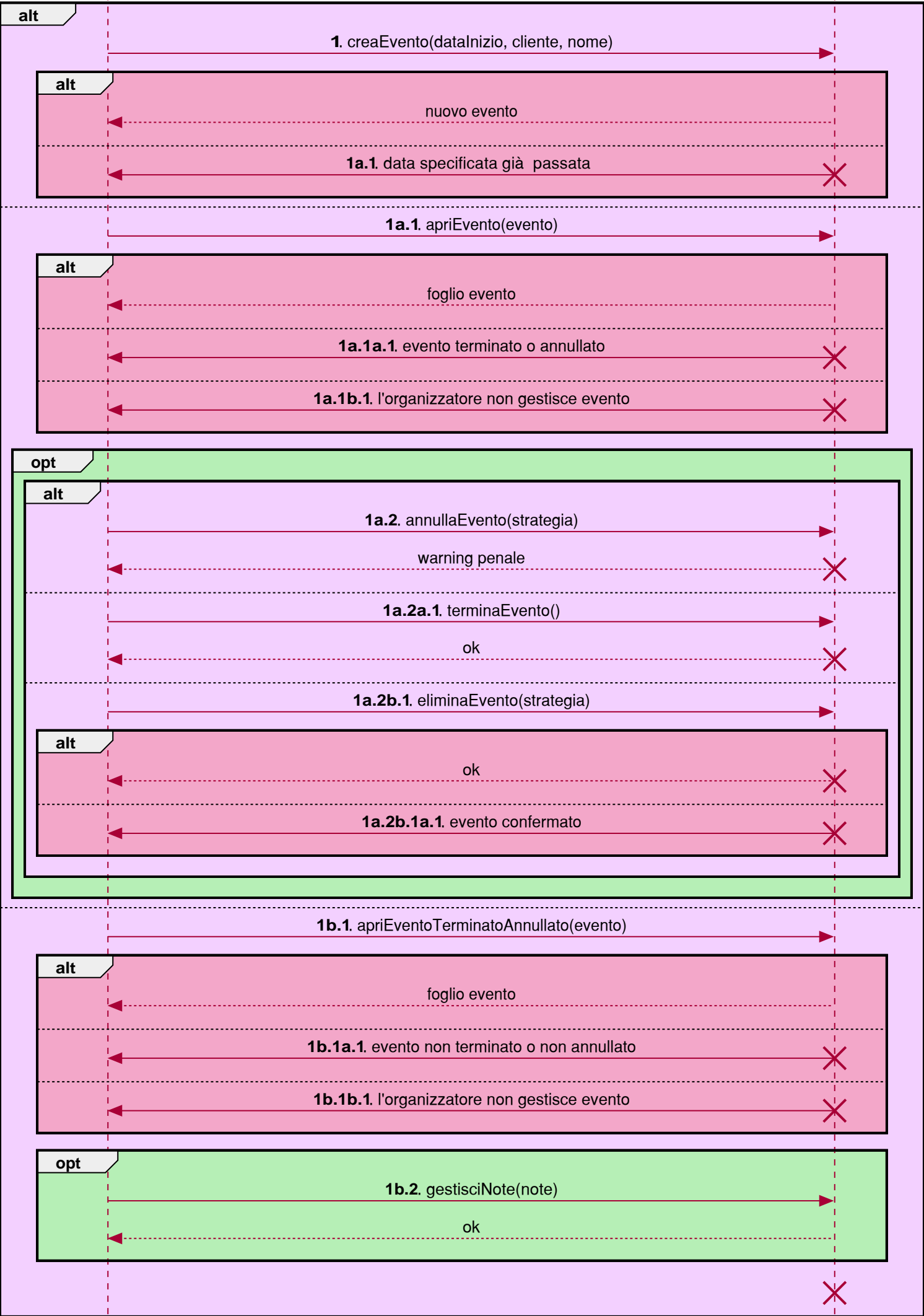
ok



Organizzatore

Diagrammi di Sequenza di Sistema  
per l'UC Gestire Eventi  
Estensioni al passo 1 (1a, 1a.2a, 1a.2b, 1b) e  
relative eccezioni (1a, 1a.1a, 1a.1b, 1a.2b.1a, 1b.1a, 1b.1b)

Sistema



Diagrammi di Sequenza di Sistema  
per l'UC Gestire Eventi

Estensioni al passo 4 (4a) e  
relative eccezioni (4a, 4b)



Organizzatore

Sistema

1. creaEvento(dataInizio, cliente, nome)

nuovo evento

opt

2. aggiungiNote(note)

ok

opt

3. visualizzaStatoCucina()

stato cucina

opt

alt

4. modificaDettagliEvento(dataInizio?, cliente?)

ok

alt

4a.1. lavori avviati: dataInizio non modificabile

4b.1. data specificata già passata

4a.1. modificaPeriodicità (eventoBase, periodicità , dataPartenza)

ok

opt

5. aggiungiServizio(nome, offsetInizio, rangeOrario, numeroCommensali, luogo, dettagli, strategia)

ok

opt

6. assegnaChef(chef, strategia)

ok

opt

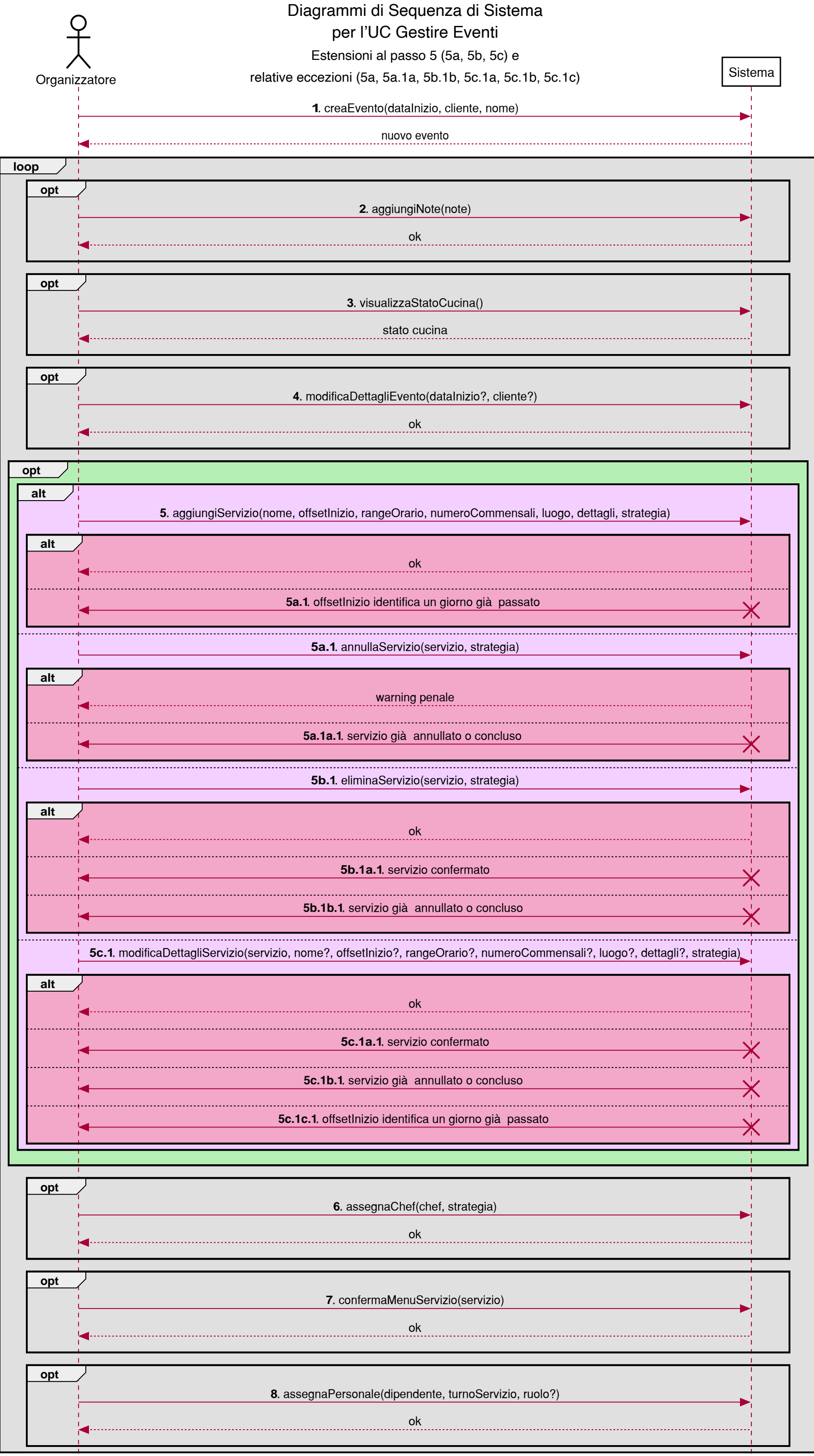
7. confermaMenuServizio(servizio)

ok

opt

8. assegnaPersonale(dipendente, turnoServizio, ruolo?)

ok



opt

6. assegnaChef(chef, strategia)

ok

opt

7. confermaMenuServizio(servizio)

ok

opt

8. assegnaPersonale(dipendente, turnoServizio, ruolo?)

ok

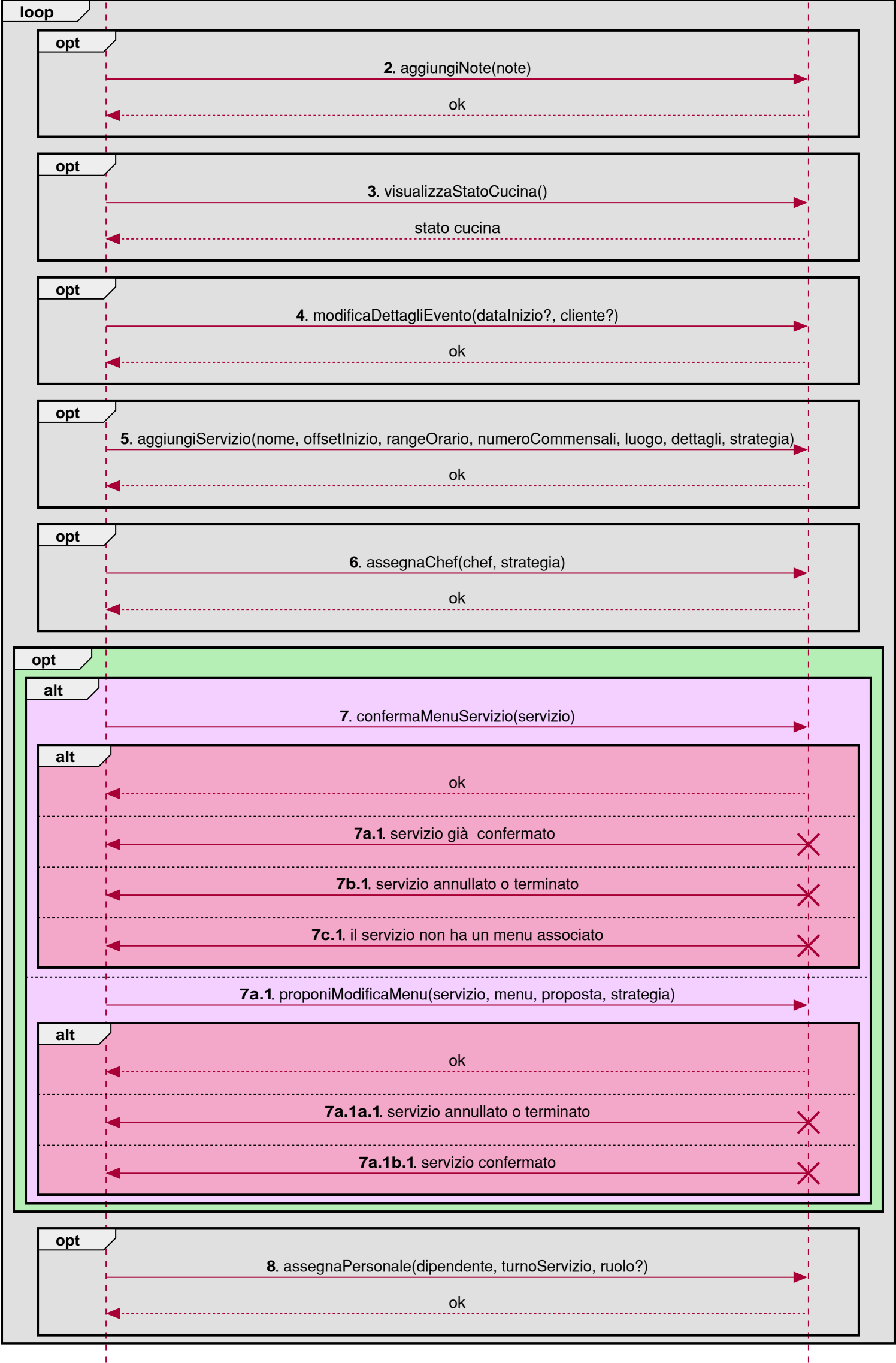


Organizzatore

# Diagrammi di Sequenza di Sistema per l’UC Gestire Eventi

Estensioni al passo 7 (7a) e  
relative eccezioni (7a, 7b, 7c, 7a.1a, 7a.1b)

Sistema



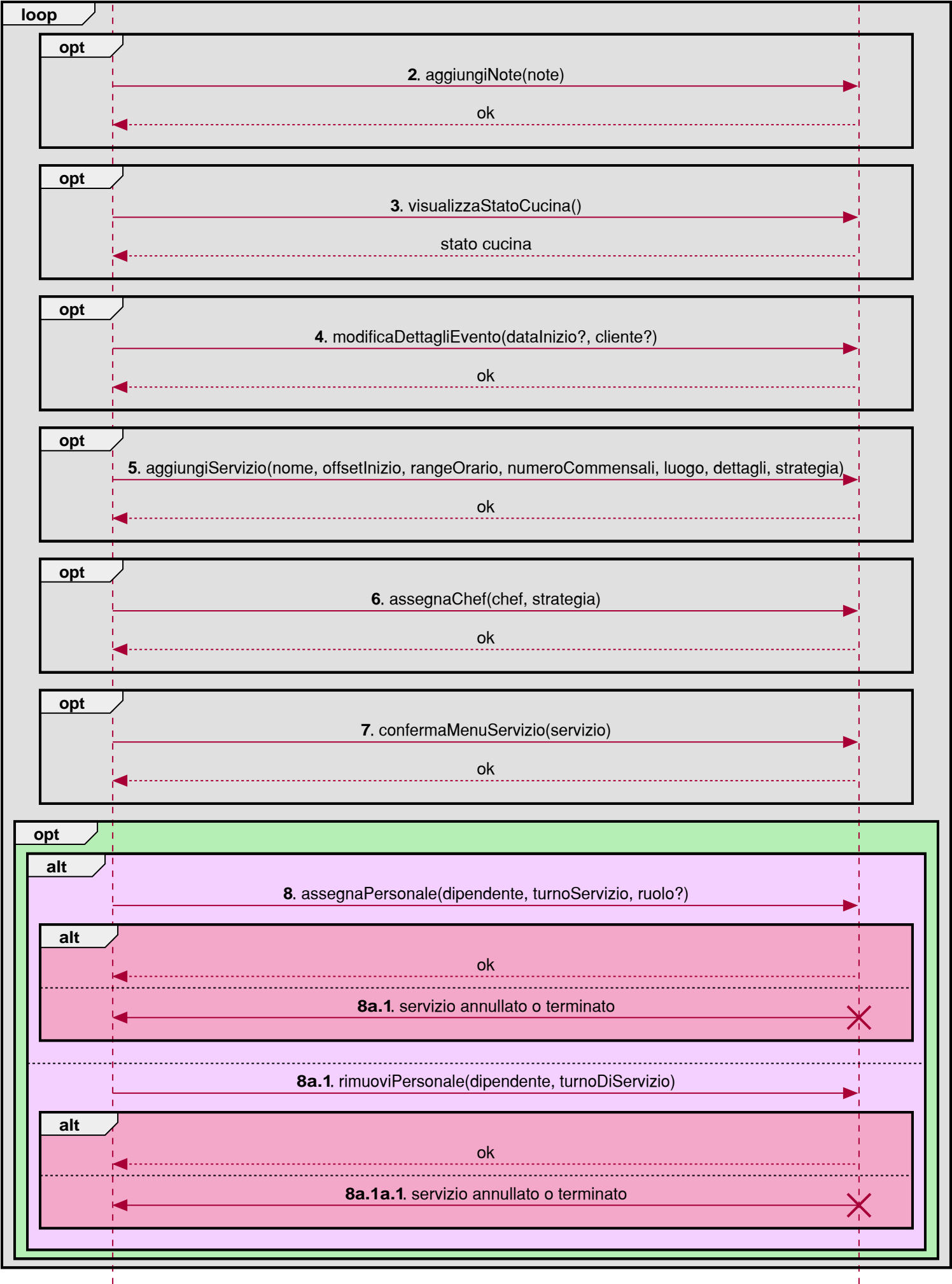


Organizzatore

# Diagrammi di Sequenza di Sistema per l'UC Gestire Eventi

Estensioni al passo 8 (8a) e  
relative eccezioni (8a, 8a.1a)

Sistema



# Contratti per l'UC "Gestire Eventi"

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

## 1. creaEvento(dataInizio: Data, cliente: Testo, nome: Testo):

**Pre-condizioni:** ---

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

- l'istanza *me* di Macro-Evento viene creata
- viene creata l'associazione **gestisce** tra *org* e *me*
- è stata creata l'istanza *ev* di Evento che è **ricorrenza di me**
- *ev.data* = dataInizio
- *me.cliente* = cliente
- *me.nome* = nome

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

**Pre-condizioni:** ---

**Post-condizioni:** Se evento.stato = programmato/attivo e *org* **gestisce** Macro-Evento *me* di cui evento è **ricorrenza**

### 1a.2 annullaEvento(strategia: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza del** Macro-Evento *me*

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

- *evento.stato* = annullato
- per ogni Servizio *s* in cui è **suddiviso evento**:
  - ogni istanza *is* Incarico di Servizio **deciso per** *s* viene eliminata
  - ogni *ic* Incarico di Cucina **distribuito su** *c* Compito **organizzato** dall'istanza *foglio* di Foglio Riepilogativo **associato** a *s*, viene eliminato
- [se strategia = "questo e successivi"] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza di me**, che abbia *evento2.stato* = programmato e che *evento.dataInizio* <= *evento2.dataInizio*
- [se strategia = "tutti"] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza di me** e che abbia *evento2.stato* = programmato



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

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

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

- *evento*.stato = terminato

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

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza** del Macro-Evento *me*

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

- ogni Servizio *s* in cui è **suddiviso** *evento* *s* viene eliminato e:
  - ogni istanza *is* Incarico di Servizio **deciso per** *s* viene eliminata
  - il *foglio* Foglio Riepilogativo **associato a** *s* viene eliminato e:
    - ogni istanza Compito *c* che *foglio* **organizza** viene eliminata e:
      - ogni *ic* Incarico di Cucina **distribuito su** *c* viene eliminato
- l'istanza *evento* viene eliminata
- [se *strategia* = "questo e successivi"] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza** di *me*, che abbia *evento2*.stato = programmato e che *evento*.dataInizio <= *evento2*.dataInizio
- [se *strategia* = "tutti"] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza** di *me* e che abbia *evento2*.stato = programmato

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

**Pre-condizioni:** ---

**Post-condizioni:** Se *evento*.stato = terminato/annullato e *org* **gestisce** Macro-Evento *me* di cui *evento* è **ricorrenza**

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

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza** del Macro-Evento *me*

**Post-condizioni:** *me*.note = *note*

## 2. aggiungiNote(note: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza** del Macro-Evento *me*

**Post-condizioni:** *me*.note = *note*

### 3. visualizzaStatoCucina():

Pre-condizioni: ---

Post-condizioni: ---

### 4. modificaDettagliEvento(dataInizio?: Data, cliente?: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza** del Macro-Evento *me*

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

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

#### 4a.1 modificaPeriodicità(eventoBase: Evento, periodicità: Periodicità, dataPartenza: Data):

**Pre-condizioni:** eventoBase è **ricorrenza** di Macro-Evento *me*

**Post-condizioni:**

- se *me* **si ripete con** Periodicità *p*, questa viene sostituita con la nuova istanza periodicità e vengono create tante istanze *eventoFuturo* di Evento quanto specificato da periodicità.numeroOccorrenze con:
  - *eventoFuturo.dataInizio* = *eventoBase.dataInizio* + (*eventoBase.offsetGiorniOccorrenza* \* *eventoBase.numeroOccorrenze*)
  - per ogni istanza di Servizio *servizio* in cui eventoBase è **suddiviso**, viene creata una copia dell'istanza *servizio* che **suddivide** *eventoFuturo*, assieme a tutte le Proposte Di Modifica che **modificano** *servizio*
  - *eventoFuturo* viene **assegnato** all'istanza Chef *c* a cui eventoBase era **assegnato**
  - la relazione è **ricorrenza di** tra *eventoFuturo* e *me* è creata
  - *eventoFuturo.stato* = programmato
- Inoltre, tutte le istanze di Evento *vecchioEvento* che **sono ricorrenza di** *me*, che non corrispondono più alla nuova periodicità e hanno *vecchioEvento.stato* = Programmato vengono rimosse.

## 5. **aggiungiServizio**(nome: Testo, offsetInizio: Numero, rangeOrario: RangeOrario, numeroCommensali: Numero, luogo: Testo, dettagli: Testo, strategia: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza** del Macro-Evento *me*

**Post-condizioni:** se offsetInizio non identifica un giorno già passato:

- Viene creata una nuova istanza *s* Servizio con:
  - *s.offsetInizio* = offsetInizio
  - *s.orario* = rangeOrario
  - *s.dettagli* = dettagli
  - *s.luogo* = luogo
  - *s.nome* = nome
  - *s.numeroCommensali* = numeroCommensali
  - la relazione è **suddiviso in** tra *s* ed *evento* viene creata
  - [se strategia = “questo e successivi”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza** di *me*, che abbia *evento2.stato* = programmato e che *evento.dataInizio* ≤ *evento2.dataInizio*
  - [se strategia = “tutti”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza** di *me* e che abbia *evento2.stato* = programmato

### 5a.1 **annullaServizio** (servizio: Servizio, strategia: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza** del Macro-Evento *me*, *evento* è **suddiviso in** servizio

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

- per ogni Personale *p* **deciso per** servizio, ogni istanza *is* Incarico di Servizio in cui *p* **serve** viene eliminata
- ogni *ic* Incarico di Cucina **distribuito su** *c* Compito **organizzato** dall'istanza *foglio* di Foglio Riepilogativo **associato** a servizio, viene eliminato.
- [se strategia = “questo e successivi”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza** di *me*, che abbia *evento2.stato* = programmato e che *evento.dataInizio* ≤ *evento2.dataInizio*
- [se strategia = “tutti”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza** di *me* e che abbia *evento2.stato* = programmato

## 5b.1 **eliminaServizio** (servizio: Servizio, strategia: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza del Macro-Evento *me***, *evento* è **suddiviso in servizio**

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

- per ogni Personale *p* **deciso per** *s*, ogni istanza *is* Incarico di Servizio in cui *p* **serve** viene eliminata
- il *foglio* Foglio Riepilogativo **associato a** *s* viene eliminato e:
  - ogni istanza Compito *c* che *foglio* **organizza** viene eliminata e:
    - ogni *ic* Incarico di Cucina **distribuito su** *c* viene eliminato
- servizio viene eliminato
- [se *strategia* = “questo e successivi”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza di *me***, che abbia *evento2*.stato = programmato e che *evento*.dataInizio ≤ *evento2*.dataInizio
- [se *strategia* = “tutti”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza di *me*** e che abbia *evento2*.stato = programmato

## 5c.1 **modificaDettagliServizio**(servizio:Servizio, nome?: Testo, offsetInizio?: Numero, rangeOrario?: RangeOrario, numeroCommensali?: Numero, luogo?: Testo, dettagli?: Testo, strategia: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza del Macro-Evento *me***; *evento* è **suddiviso in servizio**

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

- [se nome è specificato] servizio.nome = nome
- [se offsetInizio è specificato] servizio.offsetInizio = offsetInizio
- [se rangeOrario è specificato] servizio.orario = rangeOrario
- [se dettagli è specificato] servizio.dettagli = dettagli
- [se luogo è specificato] servizio.luogo = luogo
- [se numeroCommensali è specificato] servizio.numeroCommensali = numeroCommensali
- [se *strategia* = “questo e successivi”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza di *me***, che abbia *evento2*.stato = programmato e che *evento*.dataInizio ≤ *evento2*.dataInizio
- [se *strategia* = “tutti”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è **ricorrenza di *me*** e che abbia *evento2*.stato = programmato

## 6. assegnaChef(chef: Chef, strategia: Testo):

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza del** Macro-Evento *me*

**Post-condizioni:**

- chef viene **assegnato in evento**
- [se strategia = “questo e successivi”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è ricorrenza di *me*, che abbia *evento2.stato* = programmato e che *evento.dataInizio* <= *evento2.dataInizio*
- [se strategia = “tutti”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è ricorrenza di *me* e che abbia *evento2.stato* = programmato

## 7. confermaMenuServizio(servizio: Servizio):

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

**Post-condizioni:** Se servizio.stato = in preparazione e un Menu è **scelto per servizio**:

- servizio.stato = confermato

### 7a.1 proponiModificaMenu(servizio: Servizio, menu: Menu, proposta: Proposta di Modifica, strategia: Testo):

**Pre-condizioni:** menu è **scelto per servizio**; è in corso la visualizzazione di un Evento *evento* che è **ricorrenza del** Macro-Evento *me*; *evento* è **suddiviso in servizio**

**Post-condizioni:** Se servizio.stato = in preparazione:

- Viene creata l'associazione **modifica** tra proposta e servizio
- [se strategia = “questo e successivi”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è ricorrenza di *me*, che abbia *evento2.stato* = programmato e che *evento.dataInizio* <= *evento2.dataInizio*
- [se strategia = “tutti”] le modifiche sopra effettuate sono applicate anche ad ogni Evento *evento2* che è ricorrenza di *me* e che abbia *evento2.stato* = programmato

## 8. **assegnaPersonale(dipendente: Personale, turnoServizio: TurnoServizio, ruolo?: Ruolo):**

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza del** Macro-Evento *me*; *evento* è **suddiviso in servizio** Servizio

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

- viene creata l'associazione **deciso per** tra *servizio* e dipendente
- viene creato un *is* Incarico di Servizio e:
  - [se ruolo è specificato] *ic.ruolo* = ruolo
  - viene creata l'associazione **previsto in** tra *is* e turnoServizio
  - viene creata l'associazione **serve in** tra dipendente e *is*

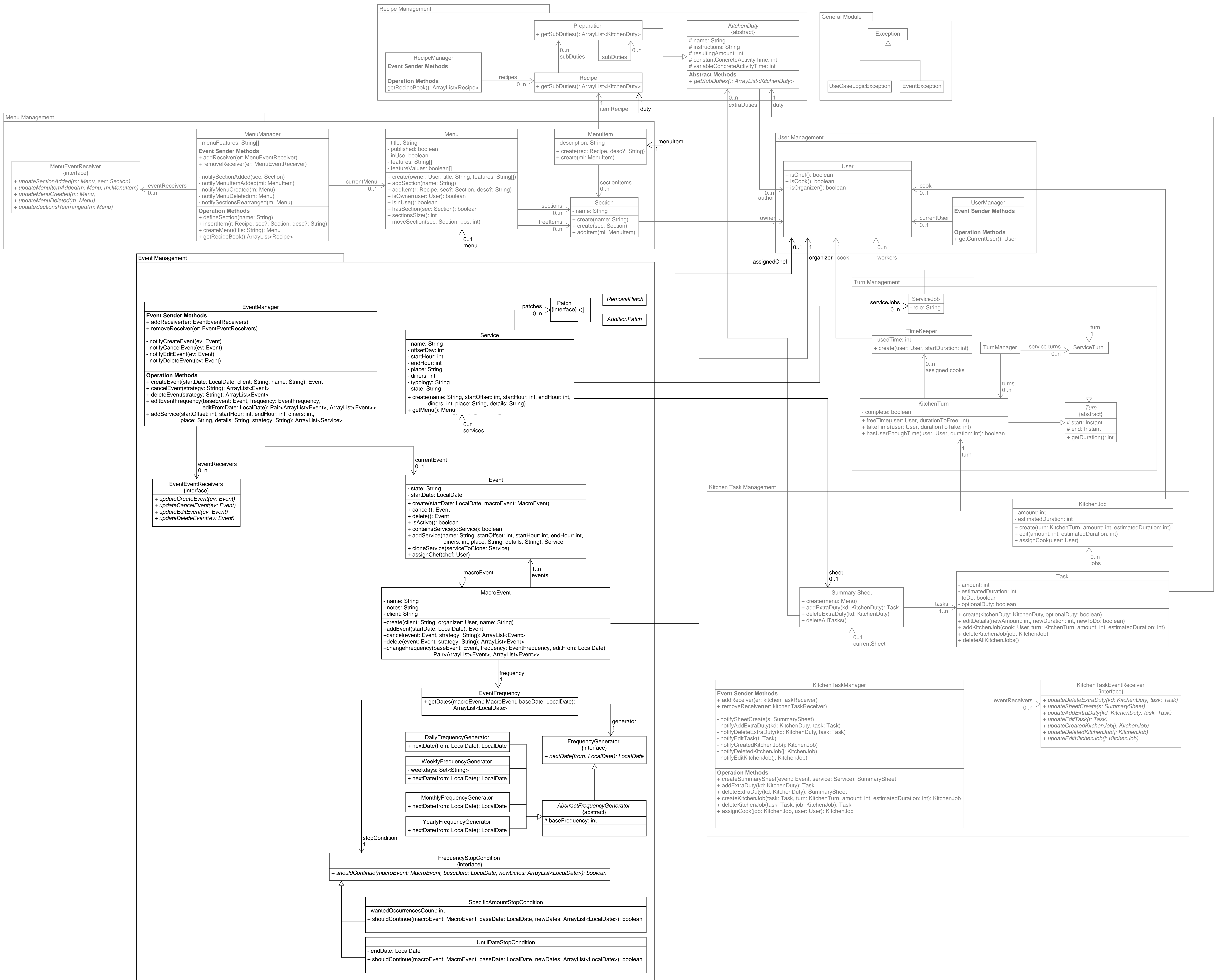
## 8a. **rimuoviPersonale(dipendente: Personale, turno: TurnoServizio):**

**Pre-condizioni:** è in corso la visualizzazione di un Evento *evento* che è **ricorrenza del** Macro-Evento *me*; *evento* è **suddiviso in servizio** Servizio

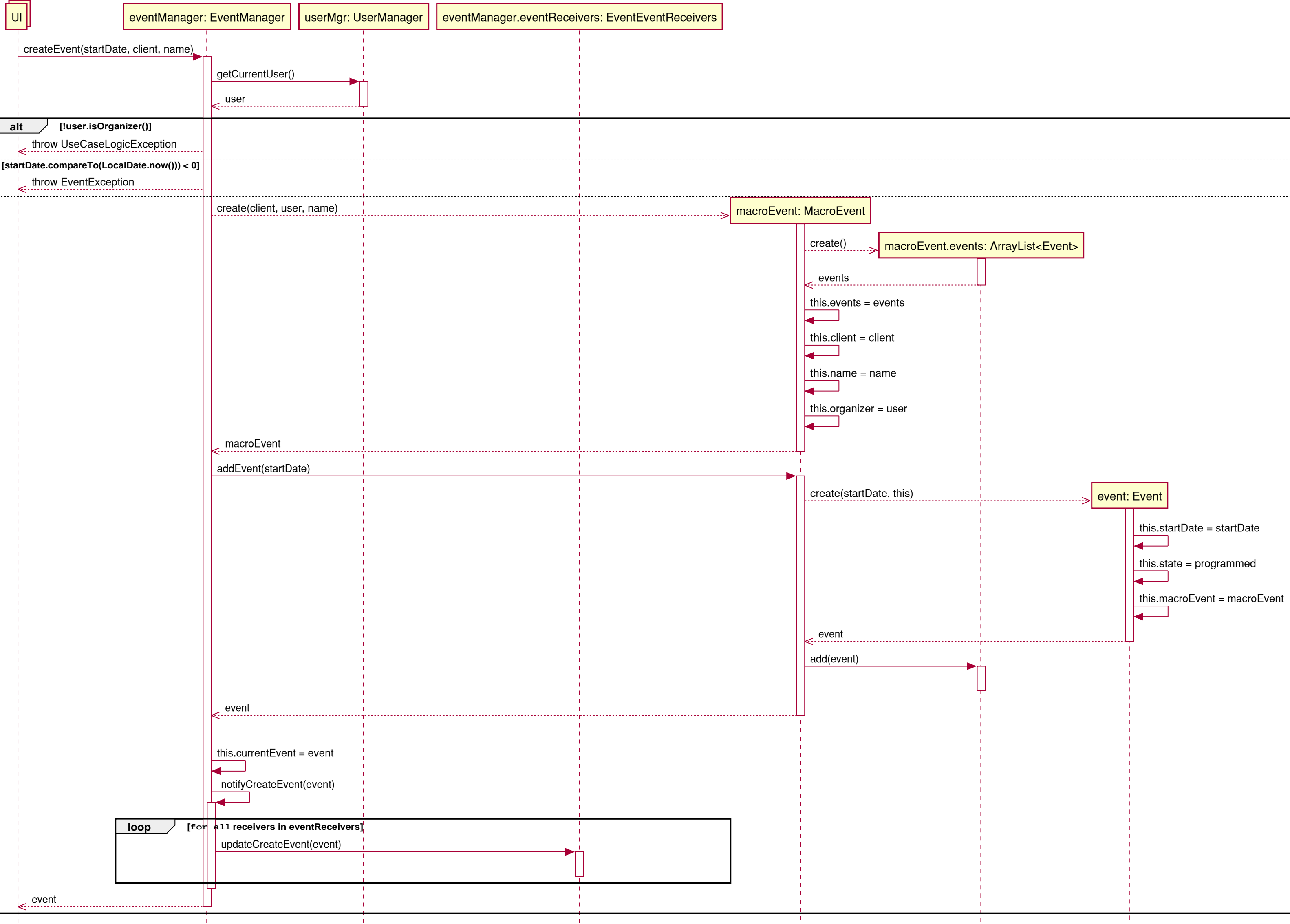
**Post-condizioni:** se *servizio*.stato non è annullato o terminato:

- per ogni *personale* Personale che è **deciso per servizio**
  - viene eliminato ogni *is* Incarico di Servizio in cui *personale* **serve**

## Design Class Diagram (DCD)

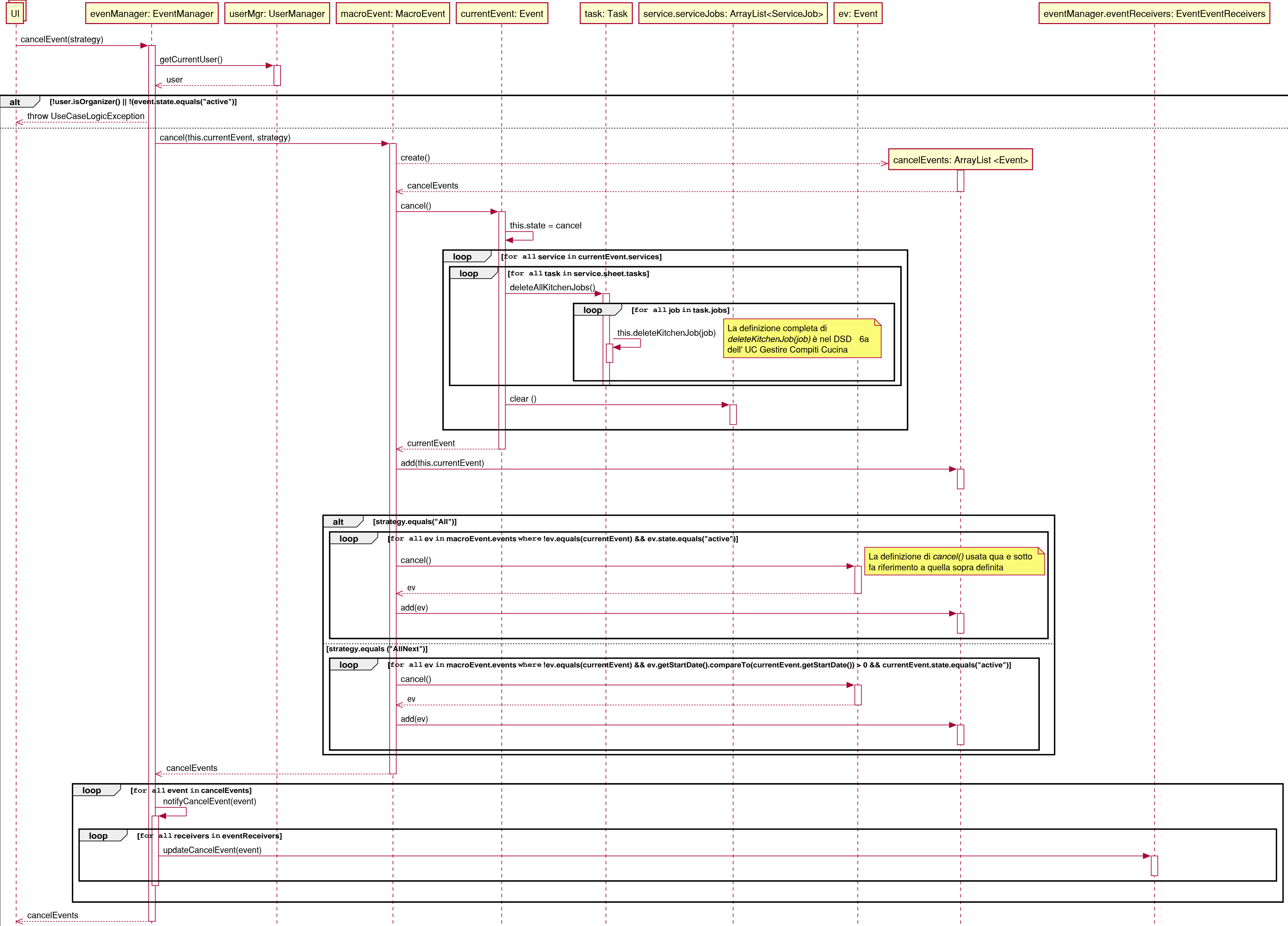


1 createEvent

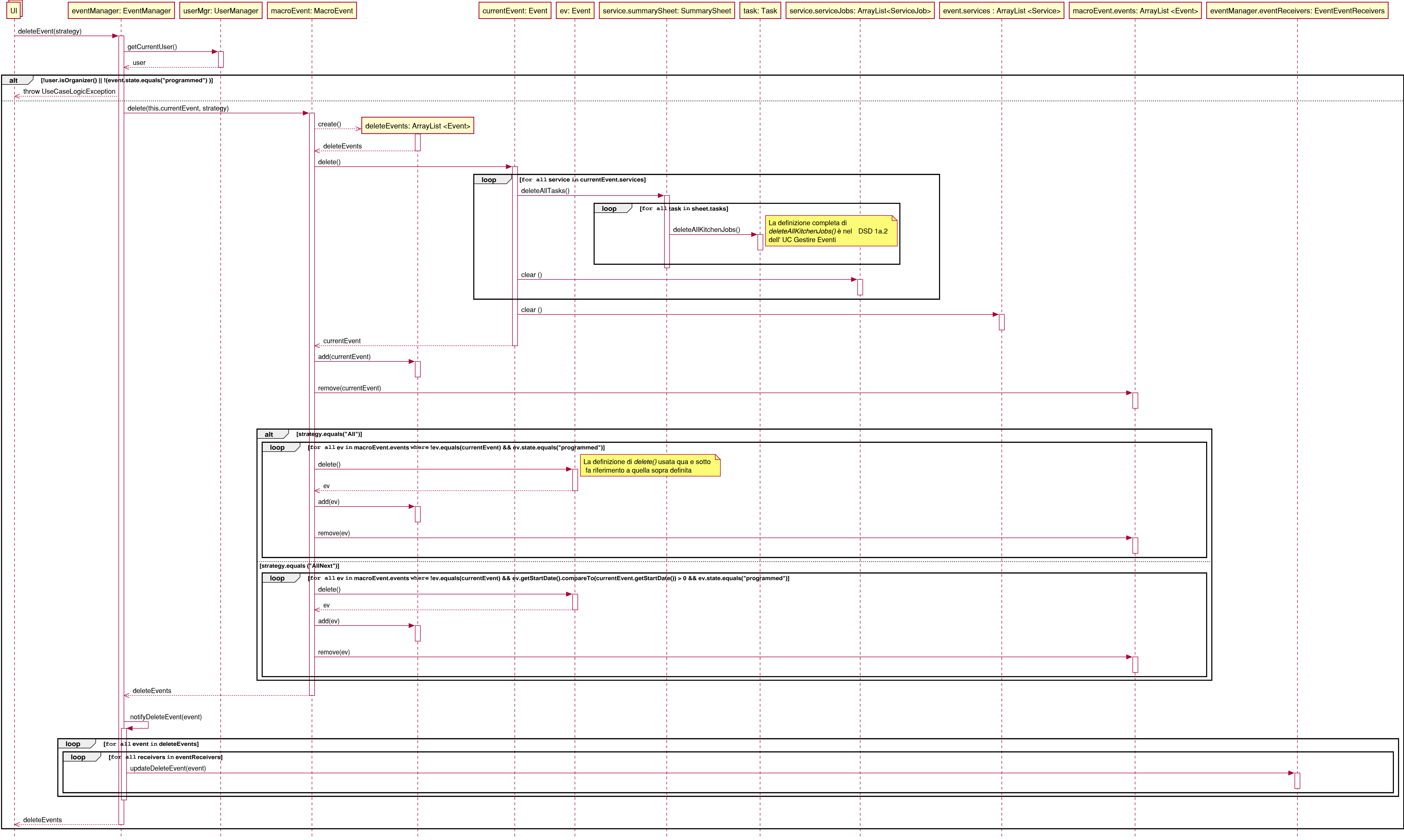




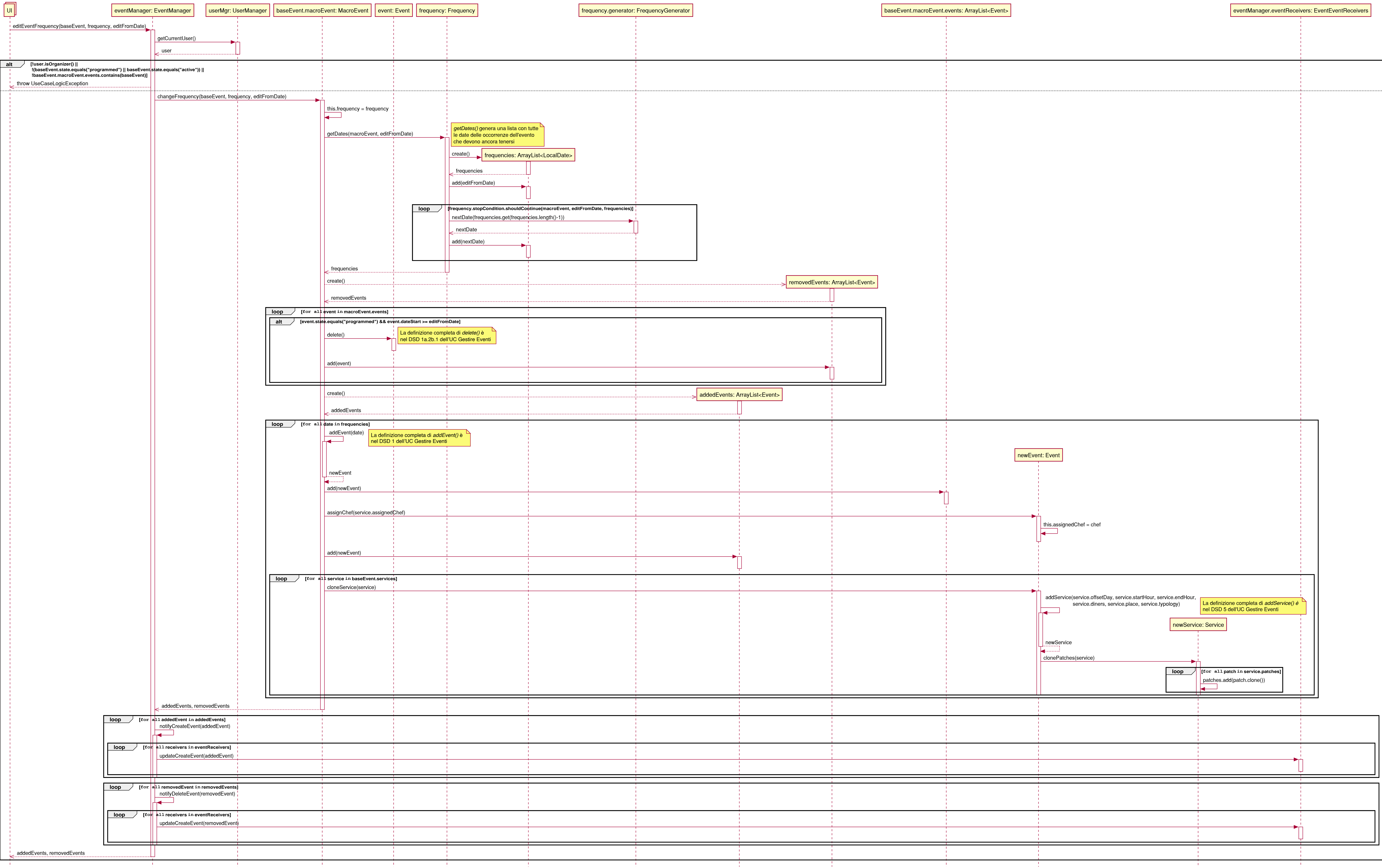
1a.2 cancelEvent



1a.2b.1 deleteEvent



4a.1 editEventFrequency



5 addService

