Progetto di Sviluppo delle Applicazioni Software

UC: “Gestire gli eventi”

A.A. 2022/2023 - Università degli Studi di Torino

Dipartimento di Informatica

Samuele Perrotta - matr. 943460

Alberto Marino - matr. 948258

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 è registrato ed è consultabile tra le schede degli eventi esistenti

Scenario principale di successo

#	Attore	Sistema
1	Crea una scheda dell'evento con uno stato[1] annotando data e numero di partecipanti	Registra la nuova scheda dell'evento con le informazioni su di esso
2	Segna sulla scheda dell'evento durata, luogo, numero di servizi, tipo servizio, opzionalmente note sulla tipologia	Registra le nuove informazioni sull'evento
3	Crea un servizio con uno stato[1] annotando fascia oraria e sede	Registra il nuovo servizio con le informazioni su di esso
	Ripete 3 finchè non è soddisfatto	
4	Assegna uno chef all'evento	Registra l'assegnamento dello chef
5	Opzionalmente, assegna i ruoli del personale per il/i servizio/i	Registra l'assegnamento del personale
6	Opzionalmente, assegna un cuoco per il servizio	Registra l'assegnazione del cuoco
	Se desidera torna al passo 5 altrimenti prosegue	
7	Opzionalmente, approva un menù	Definisce lo stato dell'evento "in corso"
8	Opzionalmente, segna delle note sull'evento terminato (quale menù,	Registra le note sull'evento terminato

	distribuzione personale, eventuali rimanenze)	
	Termina il caso d'uso	

Eccezione 4.1a

#	Attore	Sistema
4.1a.1	Assegna uno chef all'evento	Lo chef non è disponibile
	Se desidera termina il caso d'uso altrimenti ripete il passo 4	

Eccezione 5.1a

#	Attore	Sistema
5.1a.1	Opzionalmente, assegna i ruoli del personale per il/i servizio/i	Il personale non è disponibile
	Se desidera termina il caso d'uso altrimenti ripete il passo 5	

Eccezione 6.1a

#	Attore	Sistema
6.1a.1	Opzionalmente, assegna un cuoco per il servizio	Il cuoco non è disponibile
	Se desidera termina il caso d'uso altrimenti ripete il passo 6	

Estensione 1a

#	Attore	Sistema
1a.1	Apre una scheda già esistente	Fornisce la scheda dell'evento esistente
	Se desidera termina il caso d'uso altrimenti prosegue altrimenti salta al passo 3 altrimenti salta al passo 4 altrimenti salta al passo 5	

Eccezione 1a.1a

#	Attore	Sistema
1a.1a.1	Apre una scheda già esistente	La scheda dell'evento non è di proprietà dell'organizzatore
	Termina il caso d'uso	

Estensione 1b

#	Attore	Sistema
1b.1	Cancella un evento	Cancella l'evento che non sarà più visibile
	Termina il caso d'uso	

Eccezione 1b.1a

#	Attore	Sistema
1b.1a.1	Cancella un evento	L'evento è in corso e non può essere eliminato
	Termina il caso d'uso	

Eccezione 1b.1b

#	Attore	Sistema
1b.1b.1	Cancella un evento	La scheda dell'evento non è di proprietà dell'organizzatore e quindi non può eliminarlo
	Termina il caso d'uso	

Estensione 1c

#	Attore	Sistema
1c.1	Annulla l'evento	Annulla l'evento
	Salta al passo 4	

Estensione 1d

#	Attore	Sistema

1d.1	Crea un evento ricorrente con uno stato[1] annotando il numero di partecipanti, la frequenza, il numero di ripetizioni e la data di inizio	Registra il nuovo evento ricorrente con le relative informazioni

Estensione 1e

#	Attore	Sistema
1e.1	Apre un evento ricorrente già esistente	Fornisce la scheda dell'evento ricorrente esistente
	Se desidera termina il caso d'uso altrimenti continua	
1e.2	Opzionalmente, modifica durata e/o luogo e/o numero di servizi e/o tipo servizio e/o numero di partecipanti e/o data e/o note sulla tipologia e/o frequenza e/o il numero di ripetizioni della singola istanza o dell'istanza stessa e le successive o dell'intero evento, passandogli un criterio[2]	Registra le modifiche all'istanza o a tutte le istanze
1e.3	Opzionalmente, elimina singola istanza o l'istanza stessa e le successive o l'intero evento, passandogli un criterio[2]	Cancella l'istanza o tutte le istanze che non saranno più visibili a nessuno
	Se desidera termina il caso d'uso altrimenti salta al passo 3 altrimenti salta al passo 4 altrimenti salta al passo 5	

Eccezione 1e.1a

#	Attore	Sistema
1e.1a.1	Apre un evento ricorrente già esistente	La scheda dell'evento ricorrente non è di proprietà dell'organizzatore
	Termina il caso d'uso	

Eccezione 1e.3a

#	Attore	Sistema

1e.3a.1	Opzionalmente, elimina singola istanza o l'istanza stessa e le successive o l'intero evento	L'evento è in corso e non può essere eliminato
	Termina il caso d'uso	

Estensione 2a

#	Attore	Sistema
2a.1	Modifica la durata e/o luogo e/o numero di servizi e/o tipo servizio e/o numero di partecipanti e/o data e/o note sulla tipologia	Registra le modifiche sulla scheda dell'evento
	Se desidera termina il caso d'uso altrimenti prosegue altrimenti salta al passo 4 altrimenti salta al passo 5	

Estensione 3a

#	Attore	Sistema
3a.1	Cancella servizio	Cancella il servizio
	Se desidera termina il caso d'uso altrimenti ripete il passo 3	

Eccezione 3a.1a

#	Attore	Sistema
3a.1a.1	Cancella un servizio	Il servizio è in corso e non può essere eliminato
	Termina il caso d'uso	

Estensione 3b

#	Attore	Sistema
3b.1	Annulla il servizio	Annulla il servizio
	Se desidera termina il caso d'uso altrimenti ripete il passo 3	

Estensione 3c

#	Attore	Sistema
3c.1	Modifica la fascia oraria e/o la sede	Registra le modifiche sul servizio
	Se desidera termina il caso d'uso altrimenti ripete il passo 3 altrimenti prosegue altrimenti salta al passo 5	

Estensione 4a

#	Attore	Sistema
4a.1	Modifica chef	Registra la modifica dello chef
	Se desidera termina il caso d'uso altrimenti prosegue	

Eccezione 4a.1a

#	Attore	Sistema
4a.1a.1	Modifica chef	Lo chef non è disponibile
	Se desidera termina il caso d'uso altrimenti ripete il passo 4	

Estensione 5a

#	Attore	Sistema
5a.1	Modifica assegnazione personale	Registra la modifica del personale
	Se desidera termina il caso d'uso altrimenti prosegue	

Eccezione 5a.1a

#	Attore	Sistema
5a.1a.1	Modifica assegnazione personale	Il personale non è disponibile
	Se desidera termina il caso d'uso altrimenti ripete il passo 5	

Estensione 6a

#	Attore	Sistema
6a.1	Modifica cuoco	Registra la modifica del cuoco
	Se desidera termina il caso d'uso altrimenti prosegue	

Eccezione 6a.1a

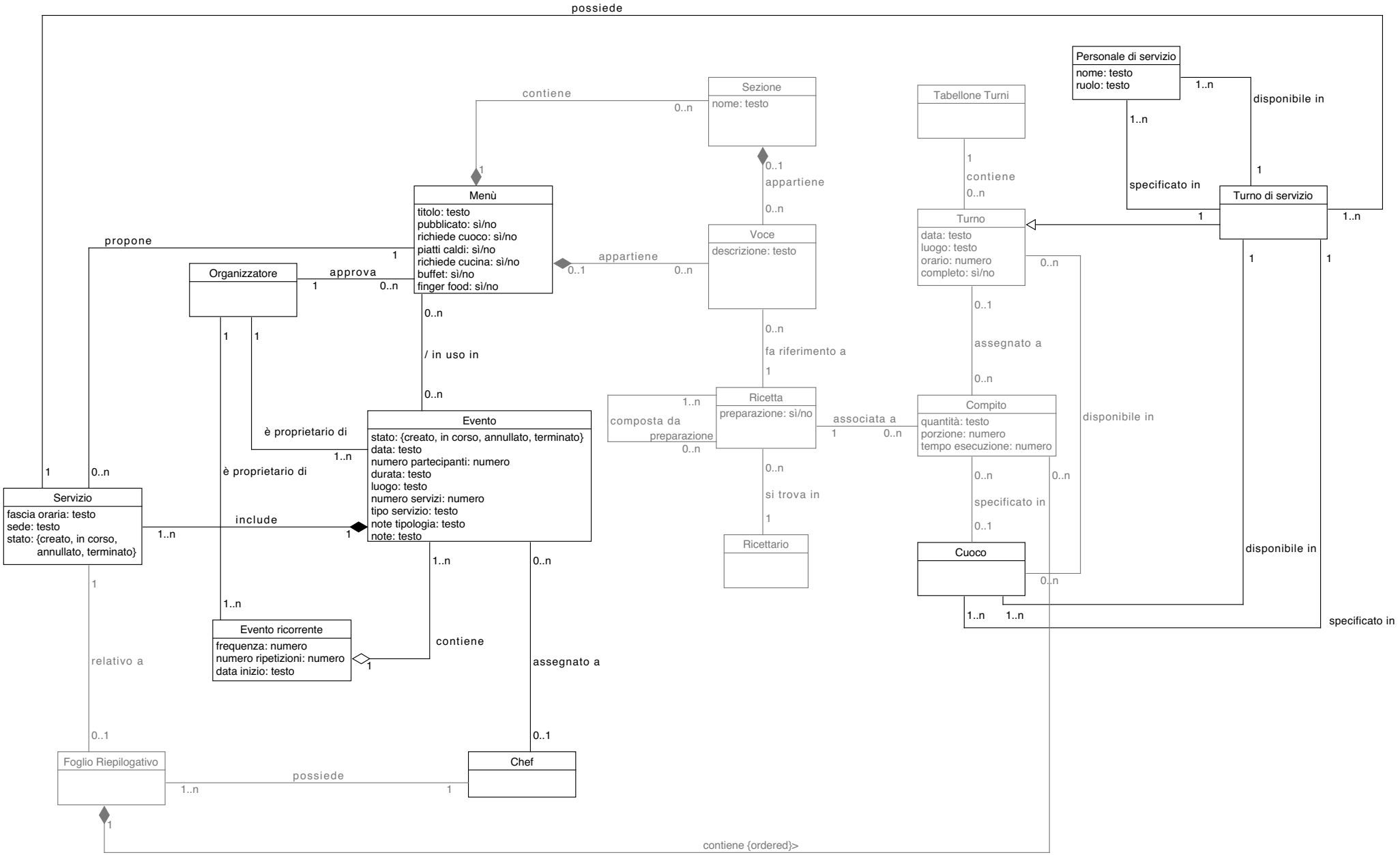
#	Attore	Sistema
6a.1a.1	Modifica cuoco	Il cuoco non è disponibile
	Se desidera termina il caso d'uso altrimenti ripete il passo 6	

---

[1] *Il cambiamento di **stato** (creato, in corso, annullato, terminato) è automaticamente gestito dal sistema*

[2] *Il **criterio** può assumere uno tra questi valori: singola, intero, successive*

Modello di Dominio



Business rules

Quando un Organizzatore è proprietario di un Evento Ricorrente sarà proprietario di tutti gli Eventi contenuti

Quando un Evento viene cancellato il sistema elimina tutti gli elementi a lui associato

L'assegnamento di un cuoco ad un servizio è opzionale

Il cambiamento di stato (creato, in corso, annullato, terminato) è automaticamente gestito dal sistema

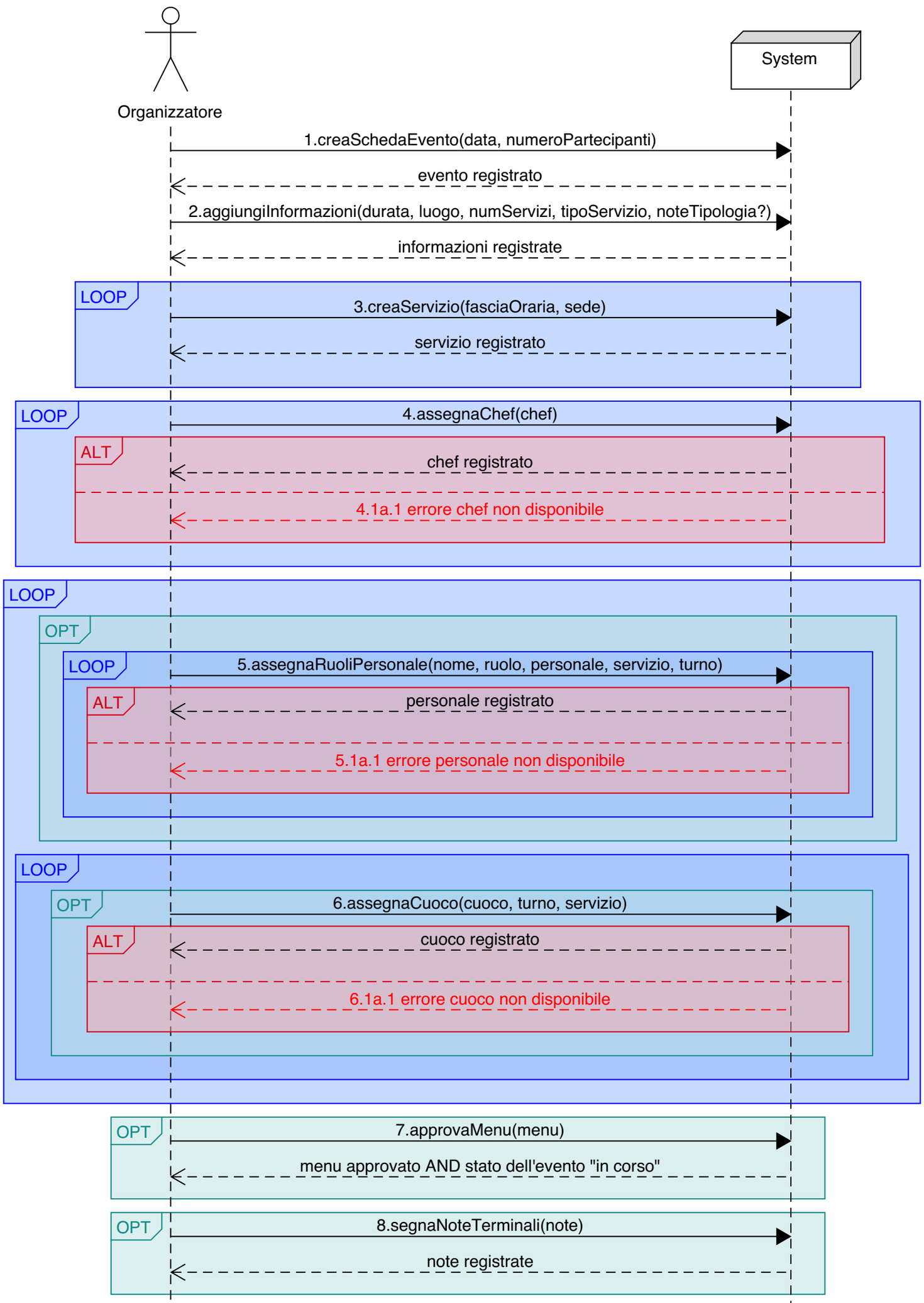
Assegnamento fattibile solo se il cuoco è disponibile nel turno

Una voce puo' comparire in un menu' all'interno di una sezione oppure come voce "libera" quindi verrà valorizzata una e una sola delle due relazioni appartiene

Una ricetta può essere composta da più preparazioni, ma non da più ricette

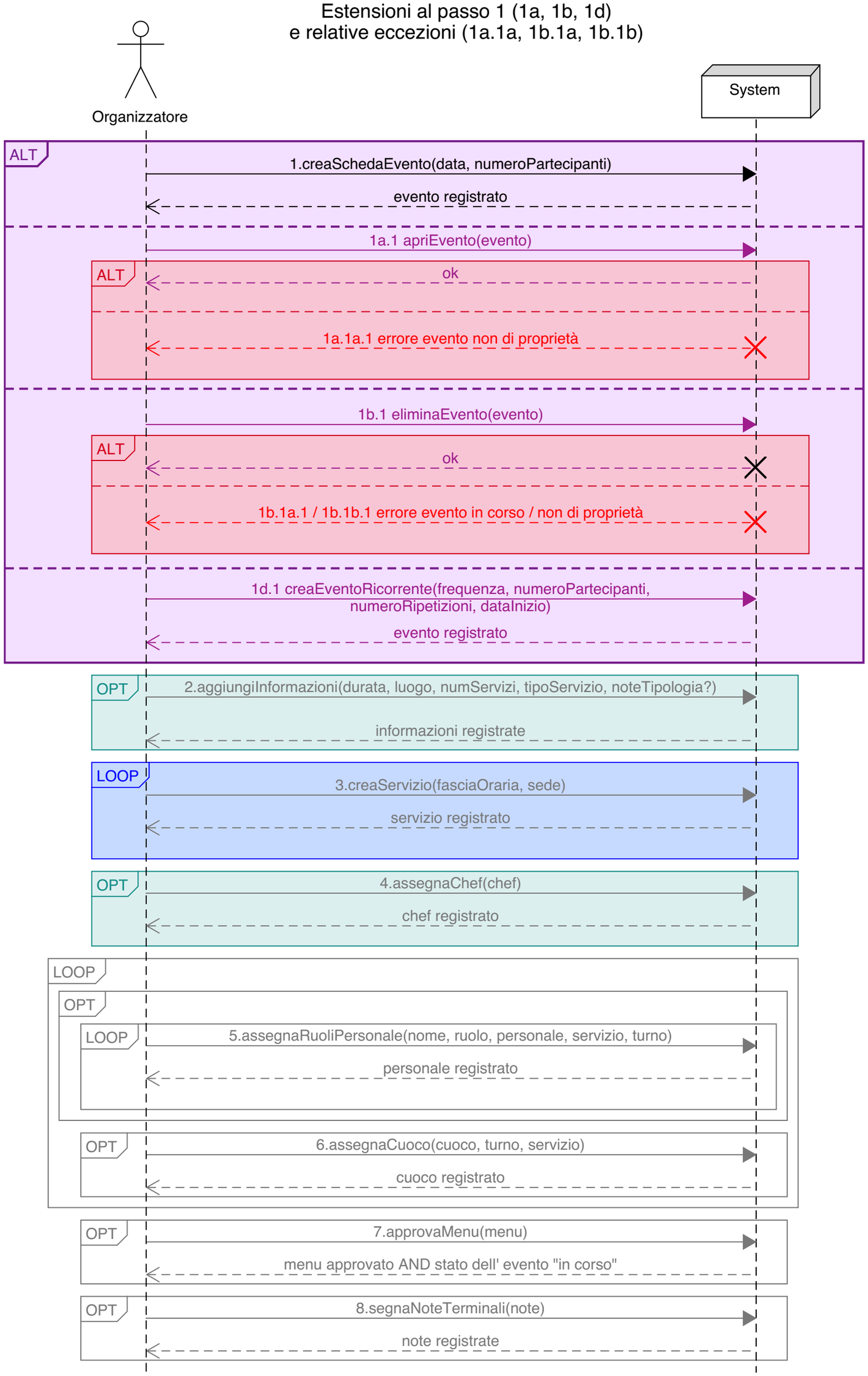
Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

Scenario principale
e relative eccezioni (4.1a, 5.1a, 6.1a)



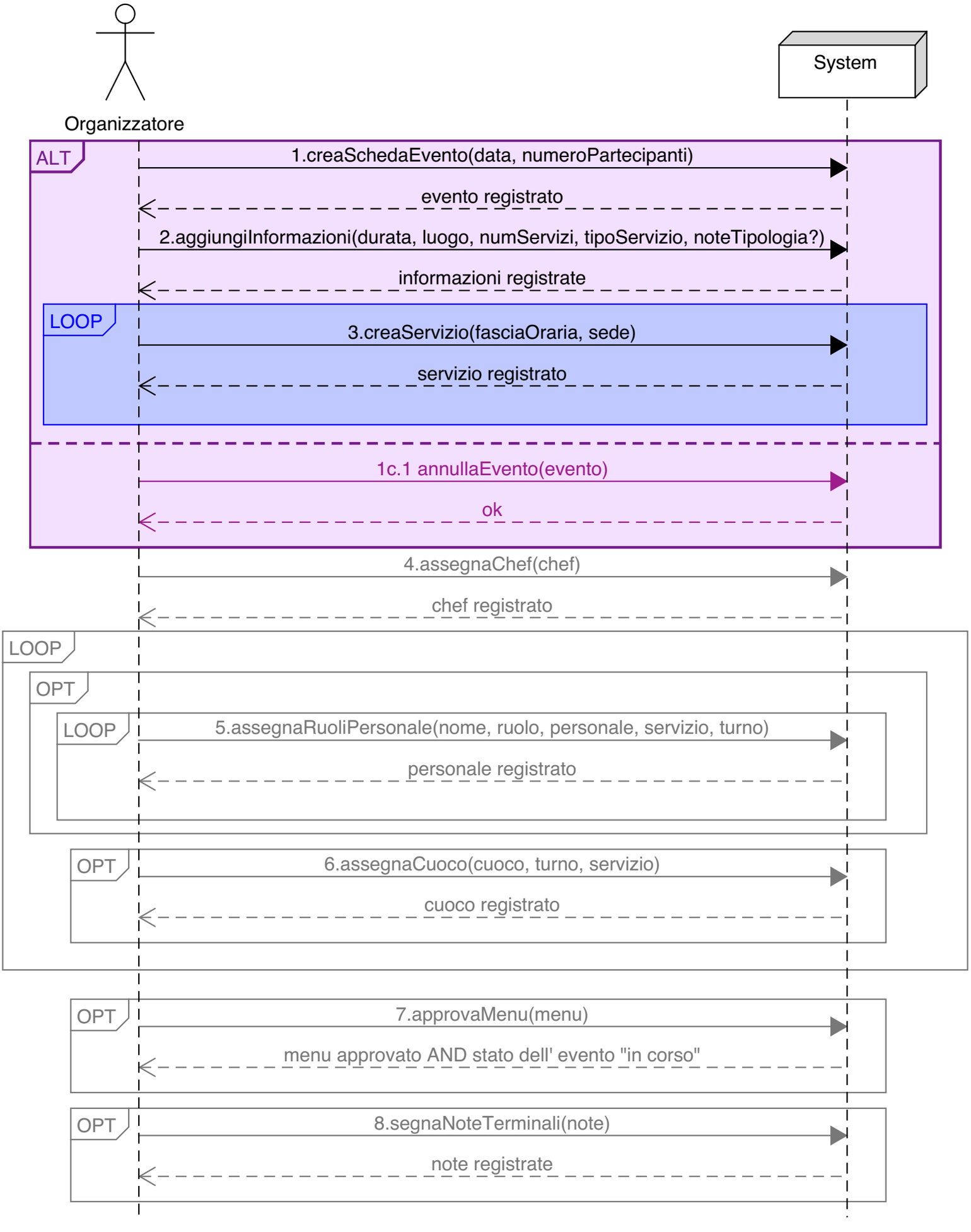
Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

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



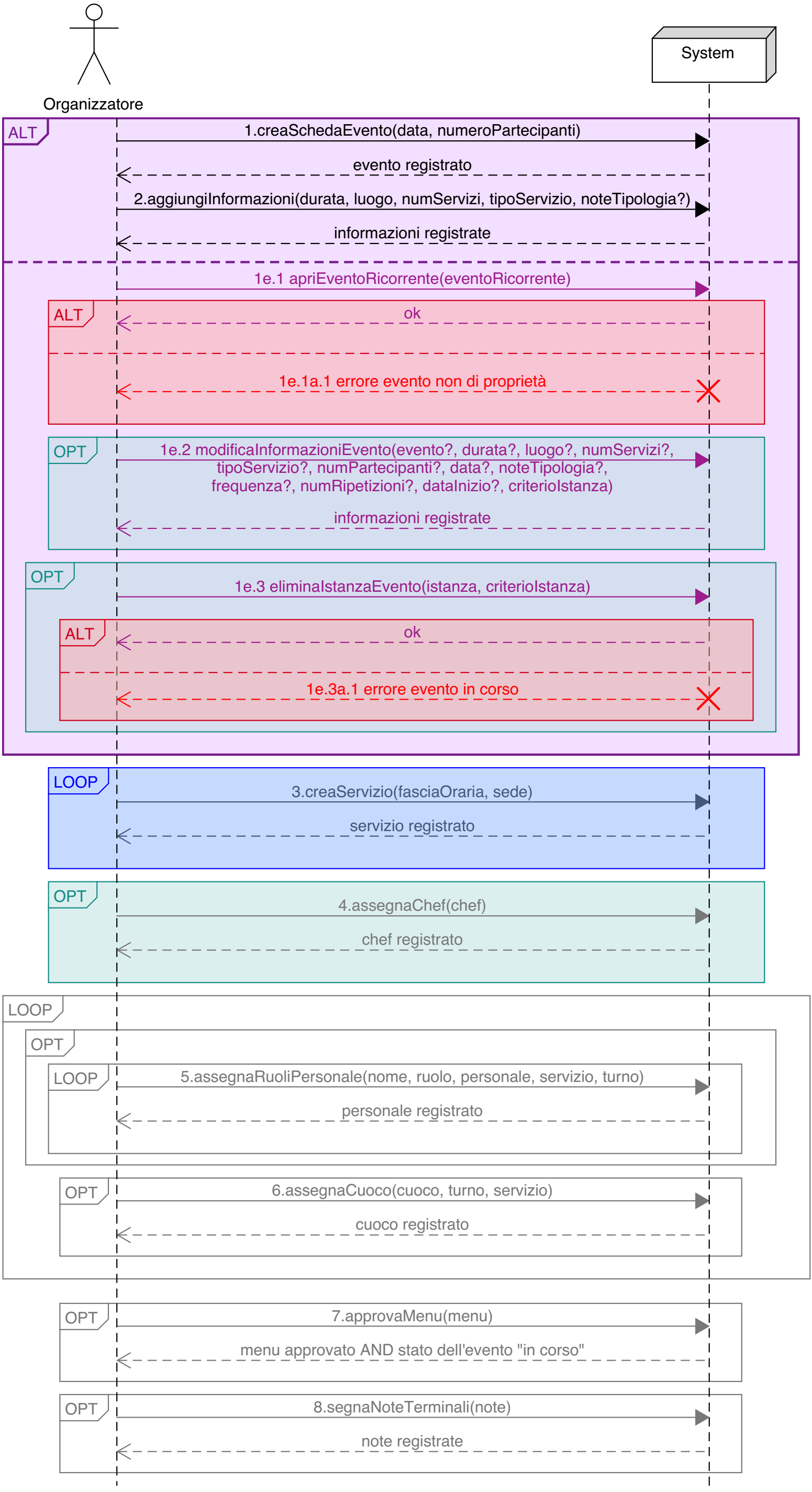
Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

Estensione al passo 1 (1c)



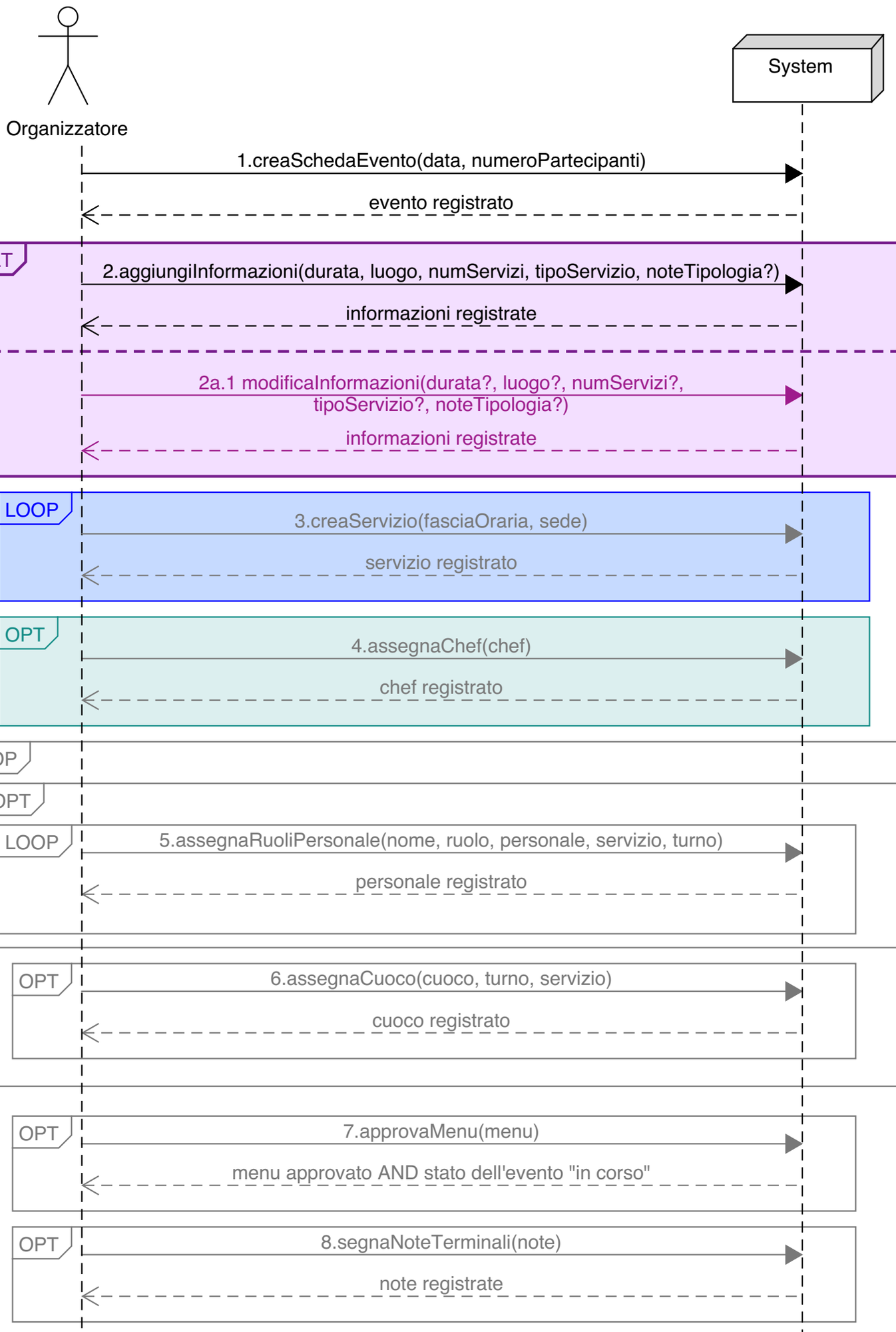
Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

Estensione al passo 1 (1e)
e relative eccezioni (1e.1a, 1e.3a)



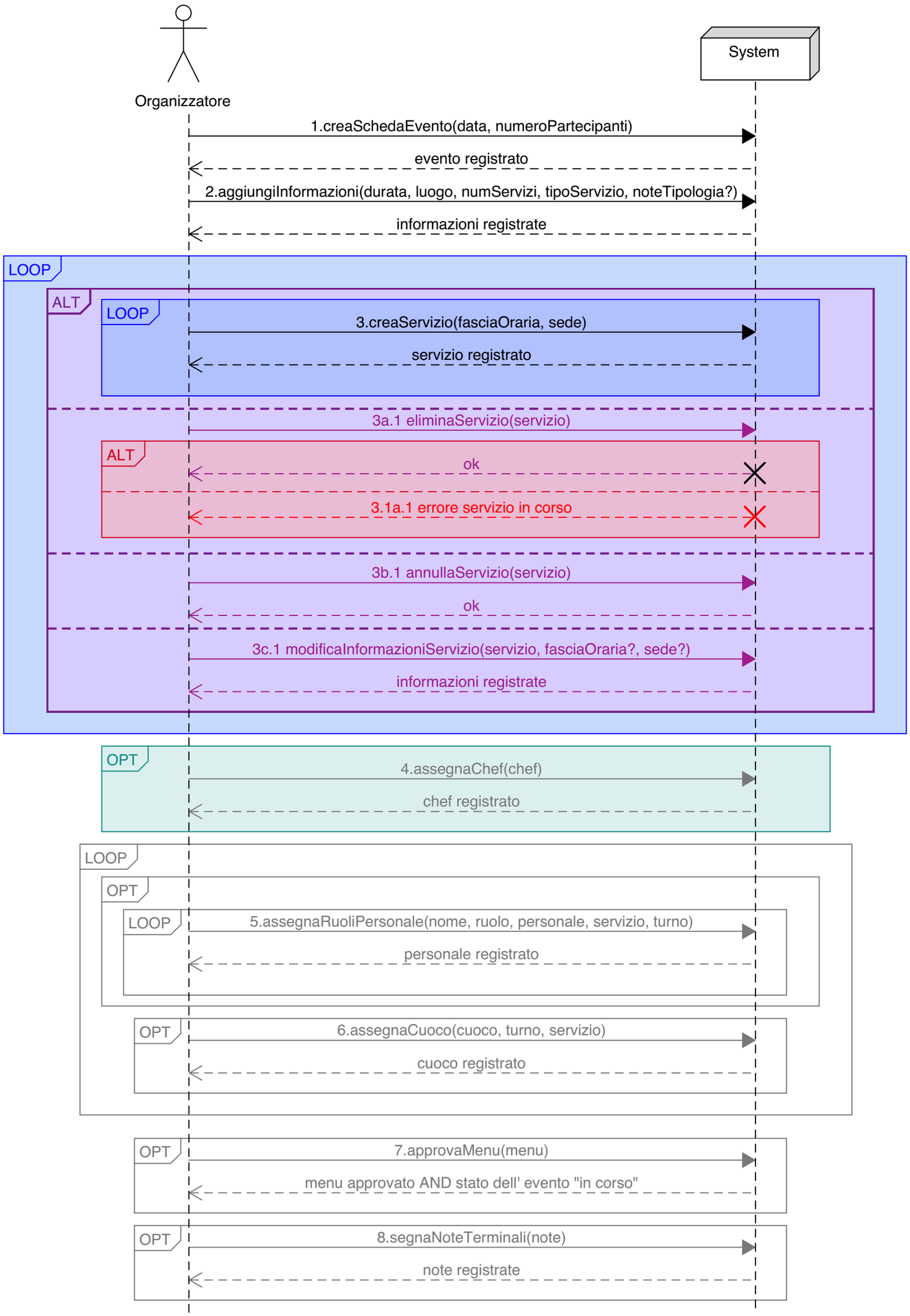
Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

Estensione al passo 2 (2a)



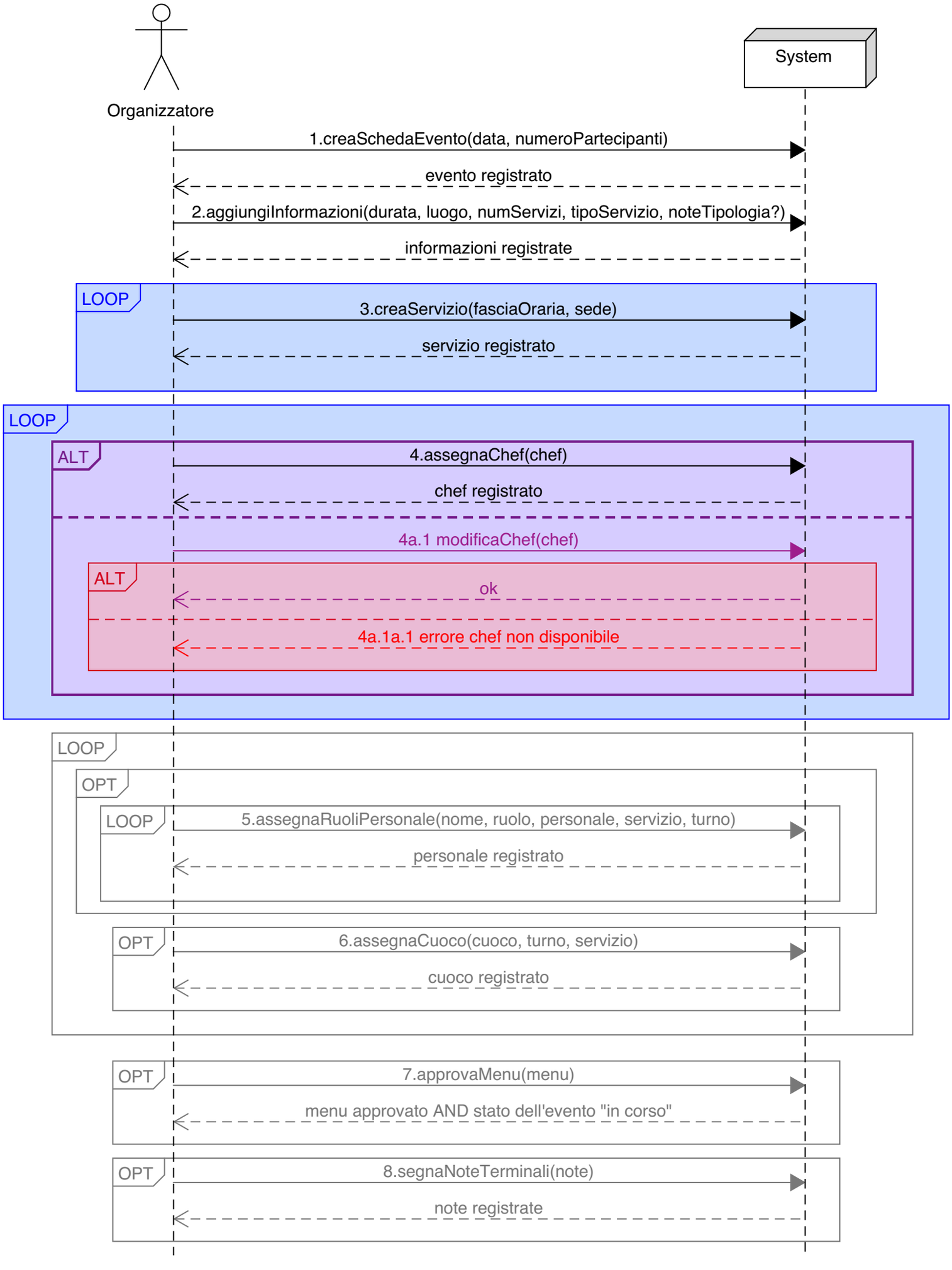
Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

Estensioni al passo 3 (3a, 3b, 3c)
e relativa eccezione (3a.1a)



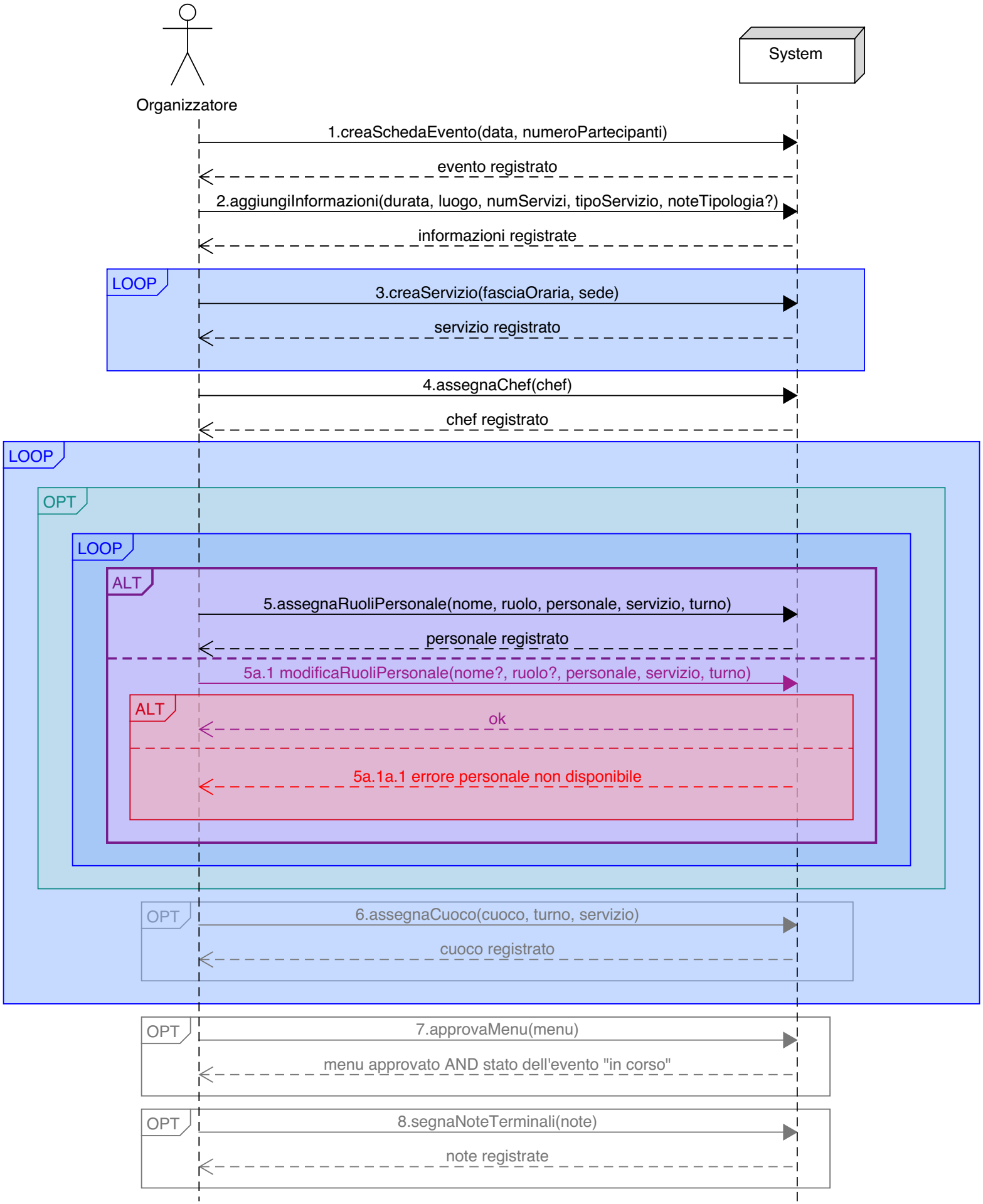
Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

Estensione al passo 4 (4a)
e relativa eccezione (4a.1a)



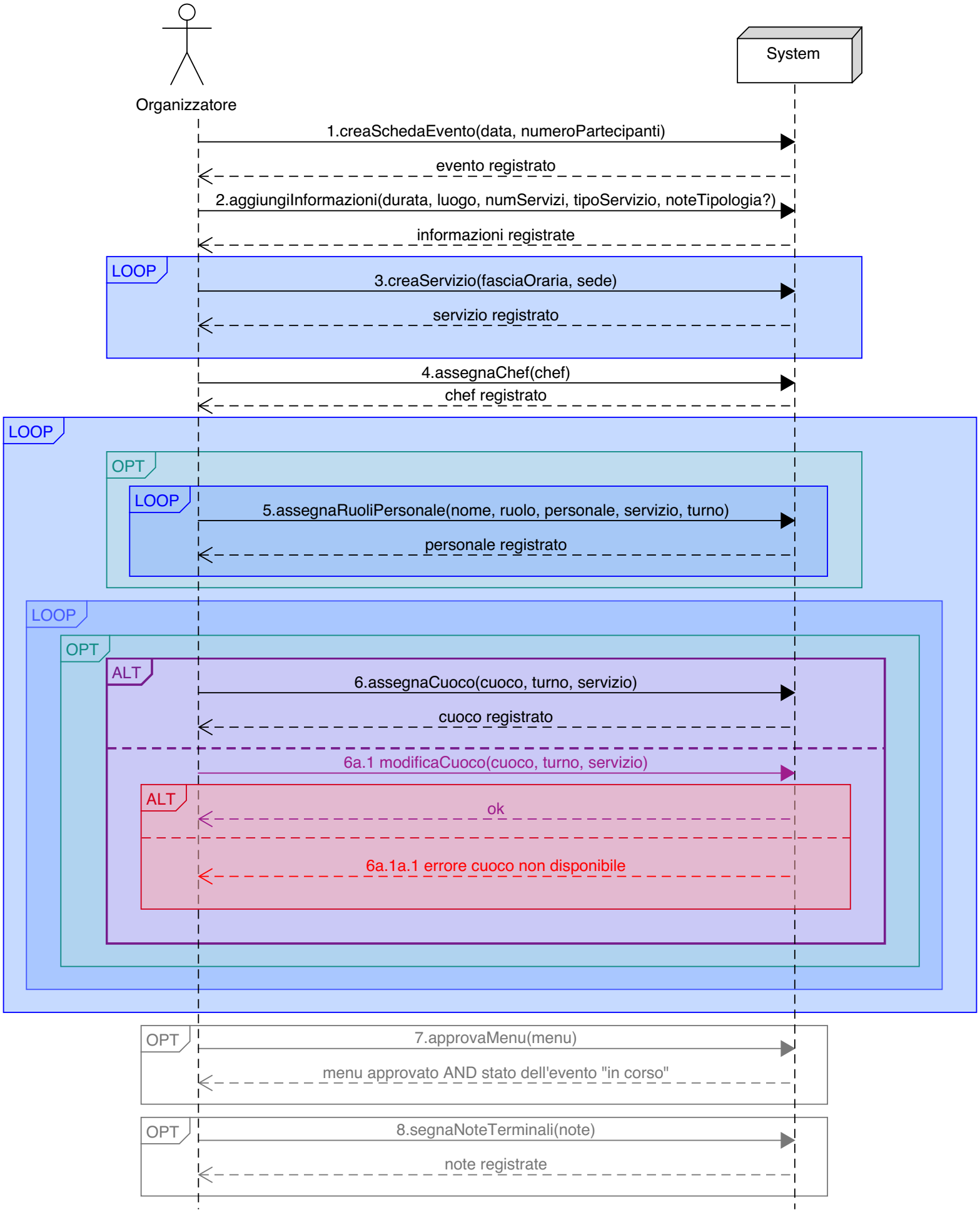
Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

Estensione al passo 5 (5a)
e relativa eccezione (5a.1a)



Diagrammi di sequenza di sistema per l'UC Gestire gli Eventi

Estensione al passo 6 (6a)
e relativa eccezione (6a.1a)



Contratti per lo UC “Gestire gli eventi”

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

1. creaSchedaEvento(data: testo, numeroPartecipanti: numero)

Pre-condizioni: ---

Post-condizioni:
- è stata creata un’istanza *ev* di Evento
- *ev.data* = data
- *ev.numeroPartecipanti* = numeroPartecipanti
- *ev.stato* = creato
- *org* **è proprietario di** *ev*

1a.1 apriEvento(evento: Evento)

Pre-condizioni: *org* **è proprietario di** evento

Post-condizioni: ---

1b.1 eliminaEvento(evento: Evento)

Pre-condizioni: ---

Post-condizioni: Se *org* **è proprietario di** evento e evento.*stato* != inCorso:
- ogni Servizio *ser* tale che evento **include** *ser* è eliminato
- ogni Turno di Servizo *turSer* tale che *ser* **possiede** *turSer* è eliminato
- l’associazione **possiede** tra *turSer* e *ser* è eliminata
- evento è eliminato

1c.1 annullaEvento(evento: Evento)

Pre-condizioni: ---

Post-condizioni:
- evento.*stato* = annullato
- per ogni Servizio *ser* tale che evento **include** *ser:*
    - *ser.stato = annullato*

1d.1 creaEventoRicorrente(frequenza: numero, numeroPartecipanti: numero, numeroRipetizioni: numero, dataInizio: testo)

Pre-condizioni: ---

Post-condizioni:
- è stata creata un’istanza *evR* di Evento Ricorrente
- *evR.numeroRipetizioni* = numeroRipetizioni
- *evR.frequenza* = frequenza
- *evR.dataInizio* = dataInizio
- *org* **è proprietario di** *evR*

- sono create numeroRipetizioni istanze *ev* di Evento
- per ogni istanza *ev* di Evento:
    - *ev.numeroPartecipanti* = numeroPartecipanti
    - *ev.stato* = creato
    - *ev.data* in accordo alla frequenza della ricorrenza
    - *org* **è proprietario di** *ev*
    - *evR* **contiene** *ev*

1e.1 apriEventoRicorrente(eventoRicorrente: Evento Ricorrente)

Pre-condizioni: *org* **è proprietario di** eventoRicorrente

Post-condizioni: ---

1e.2 modificaInformazioniEvento(evento?: Evento, durata?: testo, luogo?: testo, numServizi?: numero, tipoServizio?: testo, numPartecipanti?: numero, data?: testo, noteTipologia?: testo, frequenza?: numero, numRipetizioni?: numero, dataInizio?: testo, criterioIstanza: testo)

Pre-condizioni:
- è in corso la modifica di un Evento Ricorrente *evR*

Post-condizioni:
- Se criterioIstanza = singola ed è specificato evento:
    - [se è specificata una durata] evento.*durata* = durata
    - [se è specificato un luogo] evento.*luogo* = luogo
    - [se è specificato un numServizi] evento.*numServizi* = numServizi
    - [se è specificato un tipoServizio] evento.*tipoServizio* = tipoServizio
    - [se è specificato un numPartecipanti] evento.*numPartecipanti* = numPartecipanti
    - [se è specificata una data] evento.*data* = data
    - [se è specificata una noteTipologia] evento.*noteTipologia* = noteTipologia

- Se criterioIstanza = intero:
    - per ogni Evento *ev* tale che *evR* **contiene** *ev*:
        - [se è specificata una durata] *ev.durata* = durata
        - [se è specificato un luogo] *ev.luogo* = luogo
        - [se è specificato un numServizi] *ev.numServizi* = numServizi
        - [se è specificato un tipoServizio] *ev.tipoServizio* = tipoServizio
        - [se è specificato un numPartecipanti] *ev.numPartecipanti* = numPartecipanti
        - [se è specificata una data] *ev.data* = data
        - [se è specificata una noteTipologia] *ev.noteTipologia* = noteTipologia
        - [se è specificata una frequenza] *evR.frequenza* = frequenza
        - [se è specificato un numRipetizioni] *evR.numRipetizioni* = numRipetizioni
        - [se è specificata una dataInizio] *evR.dataInizio* = dataInizio

- Se criterioIstanza = successive ed è specificato evento:
    - per ogni Evento *ev* tale che eventoRicorrente **contiene** *ev e ev.data* >= evento.*data*:
        - [se è specificata una durata] *ev.durata* = durata
        - [se è specificato un luogo] *ev.luogo* = luogo
        - [se è specificato un numServizi] *ev.numServizi* = numServizi
        - [se è specificato un tipoServizio] *ev.tipoServizio* = tipoServizio

- [se è specificato un numPartecipanti] *ev.numPartecipanti* = numPartecipanti
- [se è specificata una data] *ev.data* = data
- [se è specificata una noteTipologia] *ev.noteTipologia* = noteTipologia
- [se è specificata una frequenza] *evR.frequenza* = frequenza
- [se è specificato un numRipetizioni] *evR.numRipetizioni* = numRipetizioni
- [se è specificata una dataInizio] *evR.dataInizio* = dataInizio

1e.3 eliminaIstanzaEvento(istanza: Evento, criterioIstanza: testo)

Pre-condizioni:
- è in corso la modifica di un Evento Ricorrente *evR*

Post-condizioni: Se istanza.*stato* != inCorso:
- Se criterioIstanza = singola:
  - ogni Servizio *ser* tale che istanza **include** *ser è eliminato*
  - ogni Turno di Servizo *turSer* tale che *ser* **possiede** *turSer* è eliminato
  - l'associazione **possiede** tra *turSer* e *ser* è eliminata
  - istanza è eliminata

- Se criterioIstanza = intero:
  - per ogni Evento *ev* tale che eventoRicorrente **contiene** *ev*:
    - ogni Servizio *ser* tale che *ev* **include** *ser è eliminato*
    - ogni Turno di Servizo *turSer* tale che *ser* **possiede** *turSer* è eliminato
    - l'associazione **possiede** tra *turSer* e *ser* è eliminata
    - *ev* è eliminato
  - eventoRicorrente è eliminato

- Se criterioIstanza = successive:
  - per ogni Evento *ev* tale che eventoRicorrente **contiene** *ev e ev.data* >= istanza.*data*:
    - ogni Servizio *ser* tale che *ev* **include** *ser è eliminato*
    - ogni Turno di Servizo *turSer* tale che *ser* **possiede** *turSer* è eliminato
    - l'associazione **possiede** tra *turSer* e *ser* è eliminata
    - *ev* è eliminato

2. aggiungiInformazioni(durata: testo, luogo: testo, numServizi: numero, tipoServizio: testo, noteTipologia?: testo)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* OPPURE è in corso la definizione di un Evento Ricorrente *evR*

Post-condizioni:
- [se è in corso la definizione di un Evento *ev*]:
  - *ev.durata* = durata
  - *ev.luogo* = luogo
  - *ev.numServizi* = numServizi
  - *ev.tipoServizio* = tipoServizio
  - [se è specificato un noteTipologia] *ev.noteTipologia* = noteTipologia

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
  - *ev.durata* = durata

- *ev.luogo* = luogo
- *ev.numServizi* = numServizi
- *ev.tipoServizio* = tipoServizio
- [se è specificato un noteTipologia] *ev.noteTipologia* = noteTipologia

2a.1 modificaInformazioni(durata?: testo, luogo?: testo, numServizi?: numero, tipoServizio?: testo, noteTipologia?: testo)

Pre-condizioni:
- è in corso la definizione di un Evento *ev*

Post-condizioni:
- [se è specificata una durata] *ev.durata* = durata
- [se è specificato un luogo] *ev.luogo* = luogo
- [se è specificato un numServizi] *ev.numServizi* = numServizi
- [se è specificato un tipoServizio] *ev.tipoServizio* = tipoServizio
- [se è specificato un noteTipologia] *ev.noteTipologia* = noteTipologia

3. creaServizio(fasciaOraria: testo, sede: testo)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* OPPURE è in corso la definizione di un Evento Ricorrente *evR*

Post-condizioni:
- [se è in corso la definizione di un Evento *ev*]:
  - è stata creata un'istanza *ser* di Servizio
  - *ser.fasciaOraria* = fasciaOraria
  - *ser.sede* = sede
  - *ser.stato* = creato
  - *ev* **include** *ser*

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
  - è stata creata un'istanza *ser* di Servizio
  - *ser.fasciaOraria* = fasciaOraria
  - *ser.sede* = sede
  - *ser.stato* = creato
  - *ev* **include** *ser*

3a.1 eliminaServizio(servizio: Servizio)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* che **include** servizio OPPURE è in corso la definizione di un Evento Ricorrente *evR* e ogni suo Evento *ev* **include** servizio

Post-condizioni: Se servizio.*stato* != inCorso:
- [se è in corso la definizione di un Evento *ev*]:
  - ogni Turno di Servizo *turSer* tale che servizio **possiede** *turSer* è eliminato
  - l'associazione **possiede** tra *turSer* e servizio è eliminata
  - servizio è eliminato

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
    - ogni Turno di Servizo *turSer* tale che servizio **possiede** *turSer* è eliminato
    - l'associazione **possiede** tra *turSer* e servizio è eliminata
    - servizio è eliminato

3b.1 annullaServizio(servizio: Servizio)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* che **include** servizio OPPURE è in corso la definizione di un Evento Ricorrente *evR* e ogni suo Evento *ev* **include** servizio

Post-condizioni:
- [se è in corso la definizione di un Evento *ev*]:
    - servizio.*stato* = annullato

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
    - servizio.*stato* = annullato

3c.1 modificaInformazioniServizio(servizio: Servizio, fasciaOraria?: testo, sede?: testo)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* che **include** servizio OPPURE è in corso la definizione di un Evento Ricorrente *evR* e ogni suo Evento *ev* **include** servizio

Post-condizioni:
- [se è in corso la definizione di un Evento *ev*]:
    - [se è specificata una fasciaOraria] servizio.*fasciaOraria* = fasciaOraria
    - [se è specificata una sede] servizio.*sede* = sede

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
    - [se è specificata una fasciaOraria] servizio.*fasciaOraria* = fasciaOraria
    - [se è specificata una sede] servizio.*sede* = sede

4. assegnaChef(chef: Chef)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* OPPURE è in corso la definizione di un Evento Ricorrente *evR*

Post-condizioni: Se chef non è **assegnato a** alcun Evento in cui Evento.*data* = *ev.data*:
- [se è in corso la definizione di un Evento *ev*]:
    - chef è **assegnato a** *ev*

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
    - chef è **assegnato a** *ev*

4a.1 modificaChef(chef: Chef)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* e lo Chef *ch* è **assegnato a** *ev* OPPURE è in corso la definizione di un Evento Ricorrente *evR* e lo Chef *ch* è **assegnato a** ogni suo *ev*

Post-condizioni: Se chef non è **assegnato a** alcun Evento in cui Evento.*data* = *ev*.*data*:
- [se è in corso la definizione di un Evento *ev*]:
  - chef è **assegnato a** *ev*

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
  - chef è **assegnato a** *ev*

5. assegnaRuoliPersonale(nome: testo, ruolo: testo, personale: PersonaleDiServizio, servizio: Servizio, turno: Turno di Servizio)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* ed *ev* **include** servizio e servizio **possiede** turno OPPURE è in corso la definizione di un Evento Ricorrente *evR* e ogni suo Evento *ev* **include** servizio e servizio **possiede** turno

Post-condizioni: Se personale è **disponibile in** turno:
- personale.*nome* = nome
- personale.*ruolo* = ruolo
- personale è **specificato in** turno

5a.1 modificaRuoliPersonale(nome?: testo, ruolo?: testo, personale: PersonaleDiServizio, servizio: Servizio, turno: TurnoDiServizio)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* ed *ev* **include** servizio e servizio **possiede** turno e personale è **specificato in** turno OPPURE è in corso la definizione di un Evento Ricorrente *evR* e ogni suo Evento *ev* **include** servizio e servizio **possiede** turno e personale è **specificato in** turno

Post-condizioni: Se personale è **disponibile in** turno:
- [se è specificato un nome] personale.*nome* = nome
- [se è specificato un ruolo] personale.*ruolo* = ruolo

6. assegnaCuoco(cuoco: Cuoco, turno: TurnoDiServizio, servizio: Servizio)

Pre-condizioni:
- è in corso la definizione di un Evento *ev* ed *ev* **include** servizio e servizio **possiede** turno OPPURE è in corso la definizione di un Evento Ricorrente *evR* e ogni suo Evento *ev* **include** servizio e servizio **possiede** turno

Post-condizioni: Se cuoco è **disponibile in** turno:
- cuoco è **specificato in** turno

6a.1 modificaCuoco(cuoco: Cuoco, turno: TurnoDiServizio, servizio: Servizio)

Pre-condizioni:

- è in corso la definizione di un Evento *ev* ed *ev* **include** servizio e servizio **possiede** turno e il Cuoco *cu* è **specificato in** turno OPPURE è in corso la definizione di un Evento Ricorrente *evR* e ogni suo Evento *ev* **include** servizio e servizio **possiede** turno e il Cuoco *cu* è **specificato in** turno

Post-condizioni: Se cuoco è **disponibile in** turno:

- cuoco è **specificato in** turno

7. approvaMenu(menu: Menu)

Pre-condizioni:

- è in corso la definizione di un Evento *ev* OPPURE è in corso la definizione di un Evento Ricorrente *evR*

Post-condizioni:

- [se è in corso la definizione di un Evento *ev*]:
  - *ev.stato* = inCorso
  - per ogni Servizio *ser* tale che *ev* **include** *ser:*
    - *ser.stato* = *inCorso*
  - *org* **approva** menu
  - menu **in uso in** *ev*

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
  - *ev.stato* = inCorso
  - per ogni Servizio *ser* tale che *ev* **include** *ser:*
    - *ser.stato* = *inCorso*
  - *org* **approva** menu
  - menu **in uso in** *ev*

8. segnaNoteTerminali(note: testo)

Pre-condizioni:

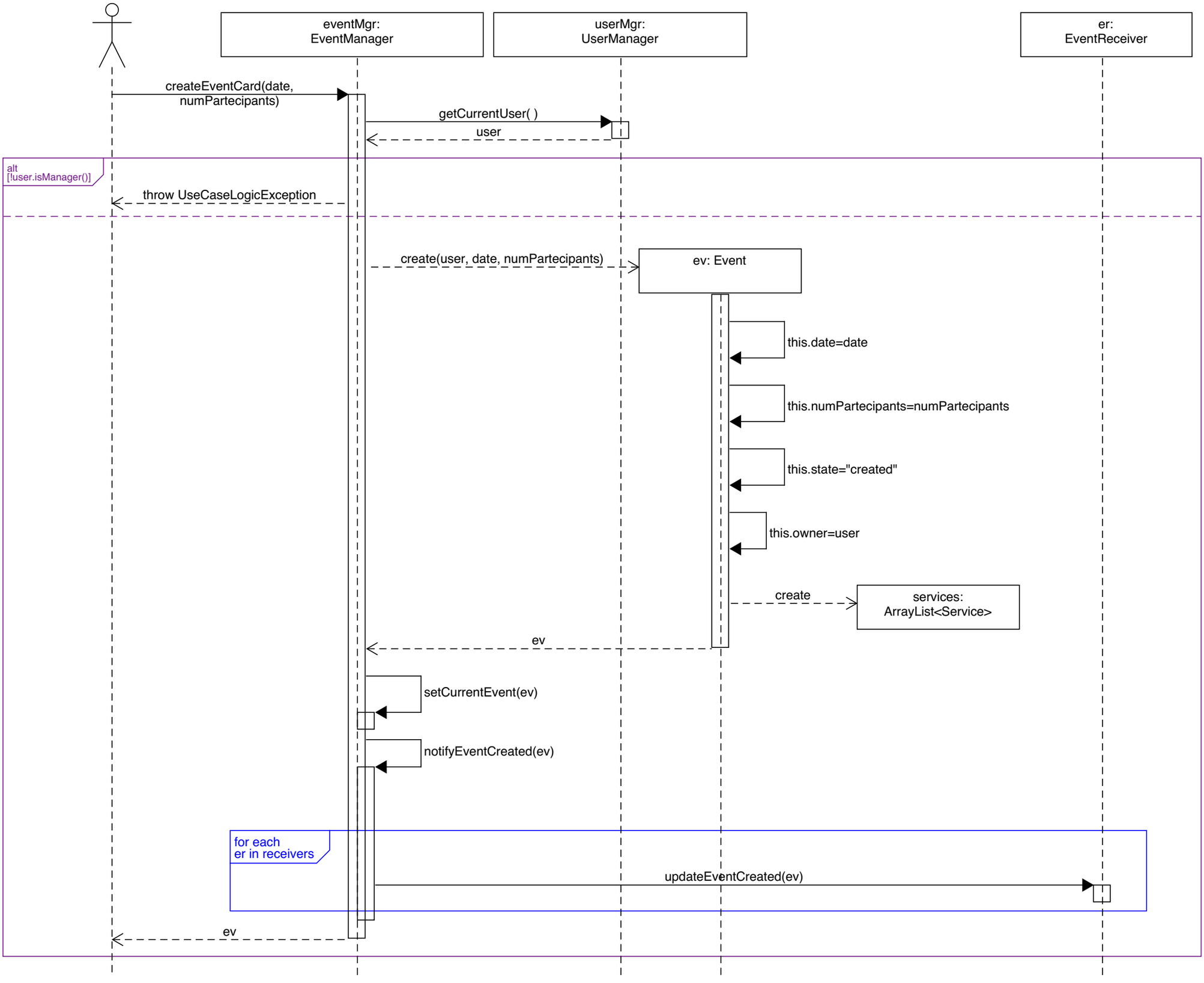
- è in corso la definizione di un Evento *ev* OPPURE è in corso la definizione di un Evento Ricorrente *evR*

Post-condizioni:

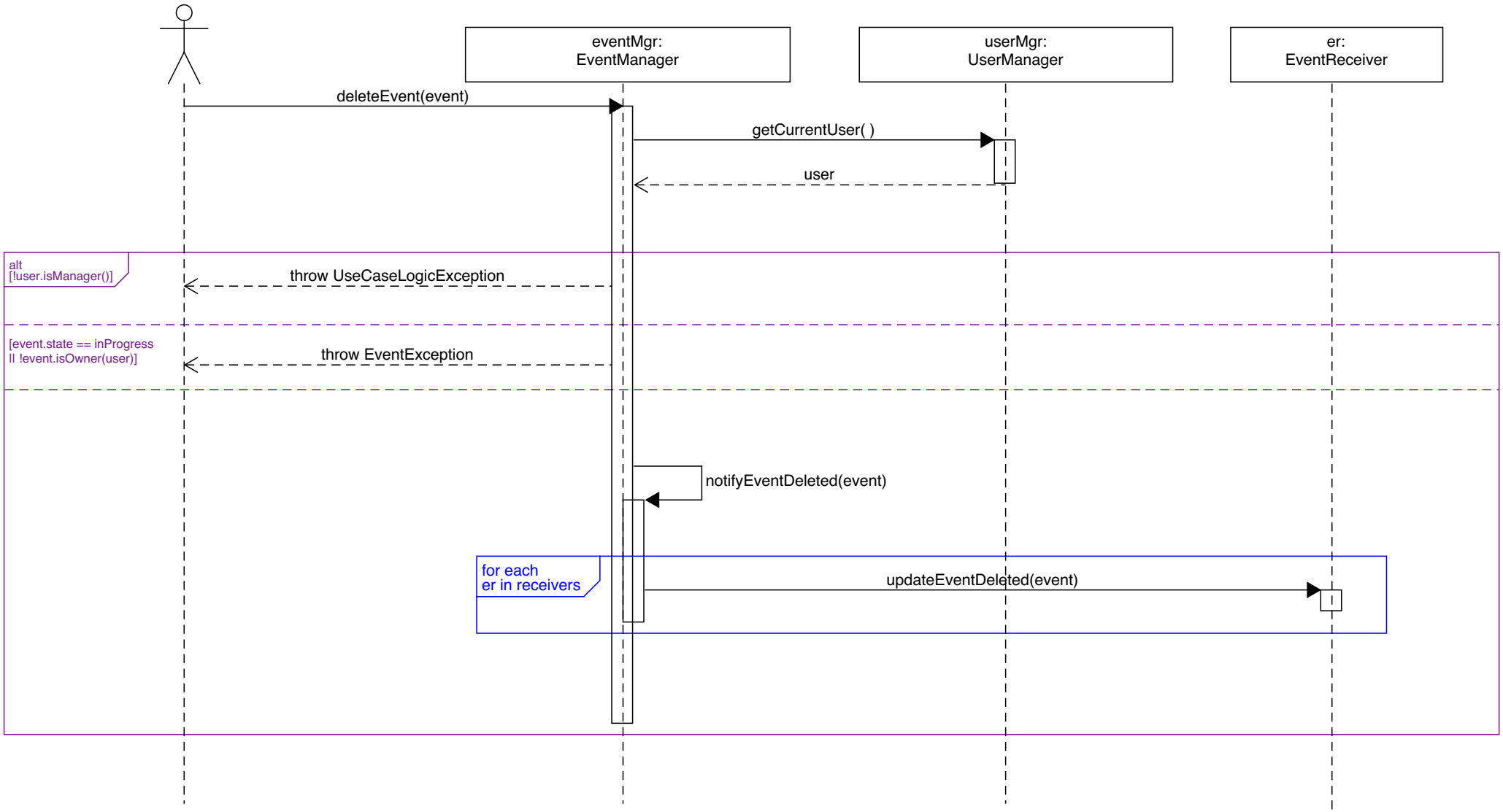
- [se è in corso la definizione di un Evento *ev*]:
  - *ev.stato* = terminato
  - *ev.note* = note
  - per ogni Servizio *ser* tale che *ev* **include** *ser:*
    - *ser.stato* = terminato

- [se è in corso la definizione di un Evento Ricorrente *evR*] per ogni Evento *ev* tale che *evR* **contiene** *ev*:
  - *ev.stato* = terminato
  - *ev.note* = note
  - per ogni Servizio *ser* tale che *ev* **include** *ser:*
    - *ser.stato* = terminato

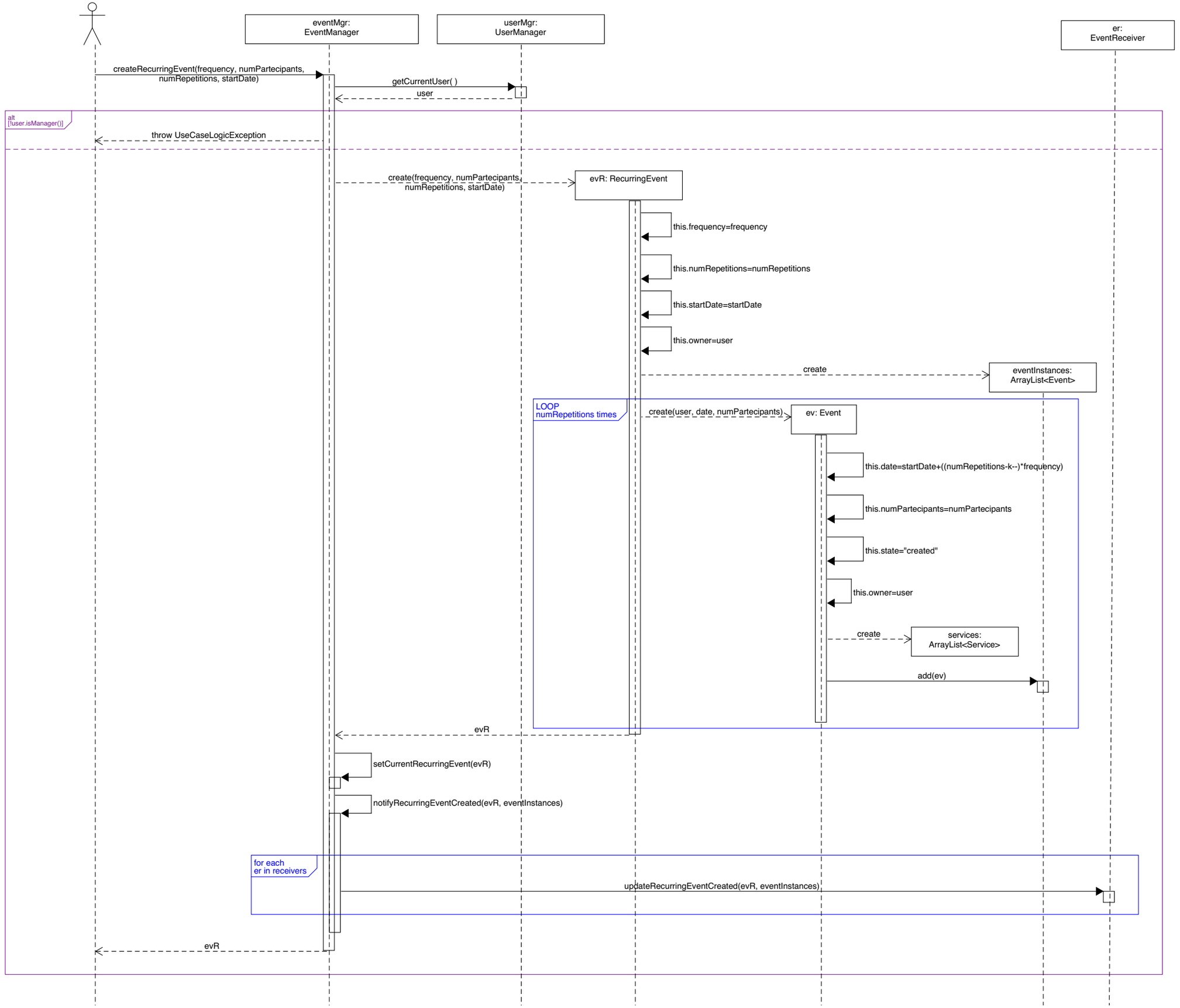
1) createEventCard



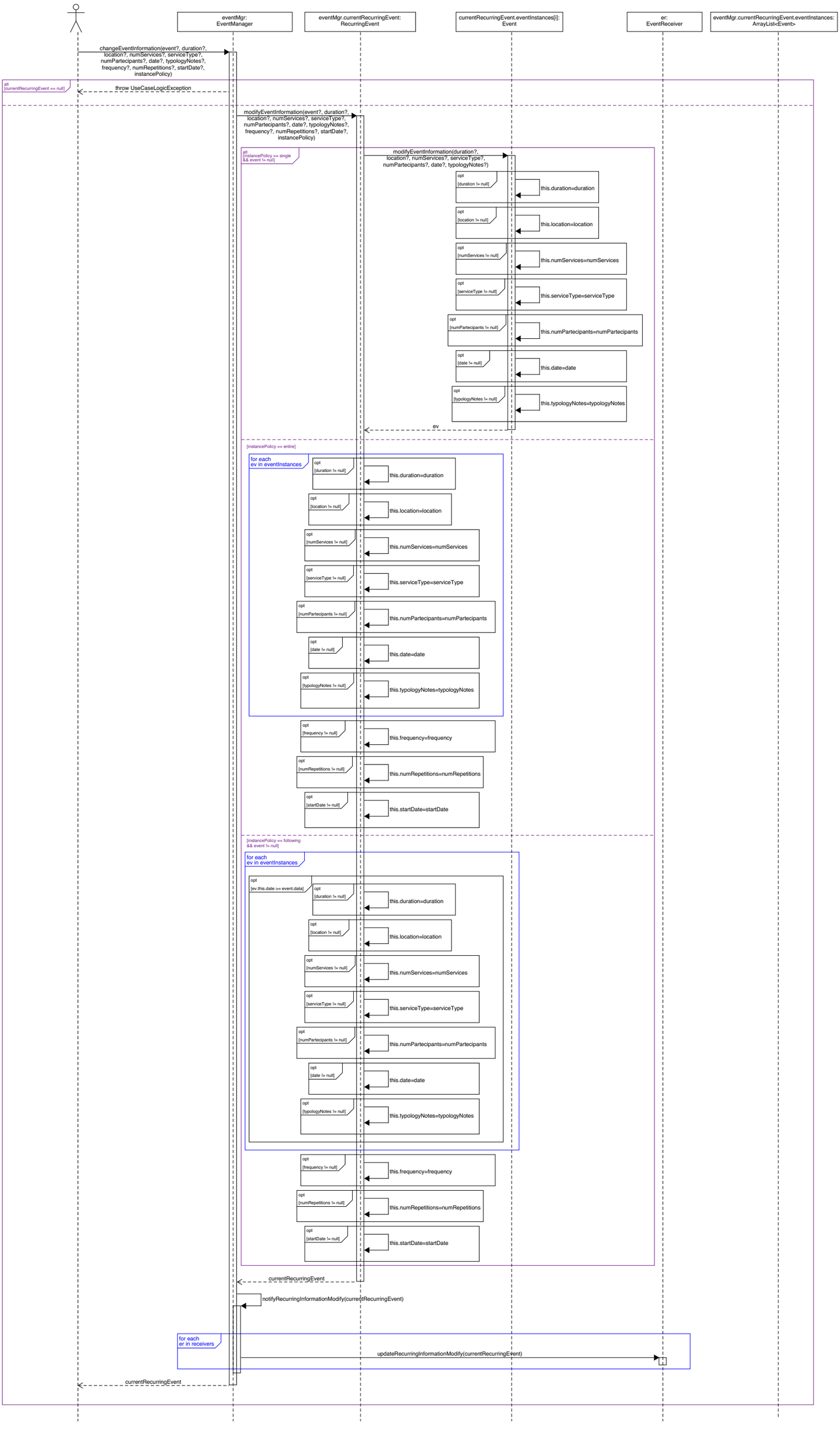
1b.1) deleteEvent



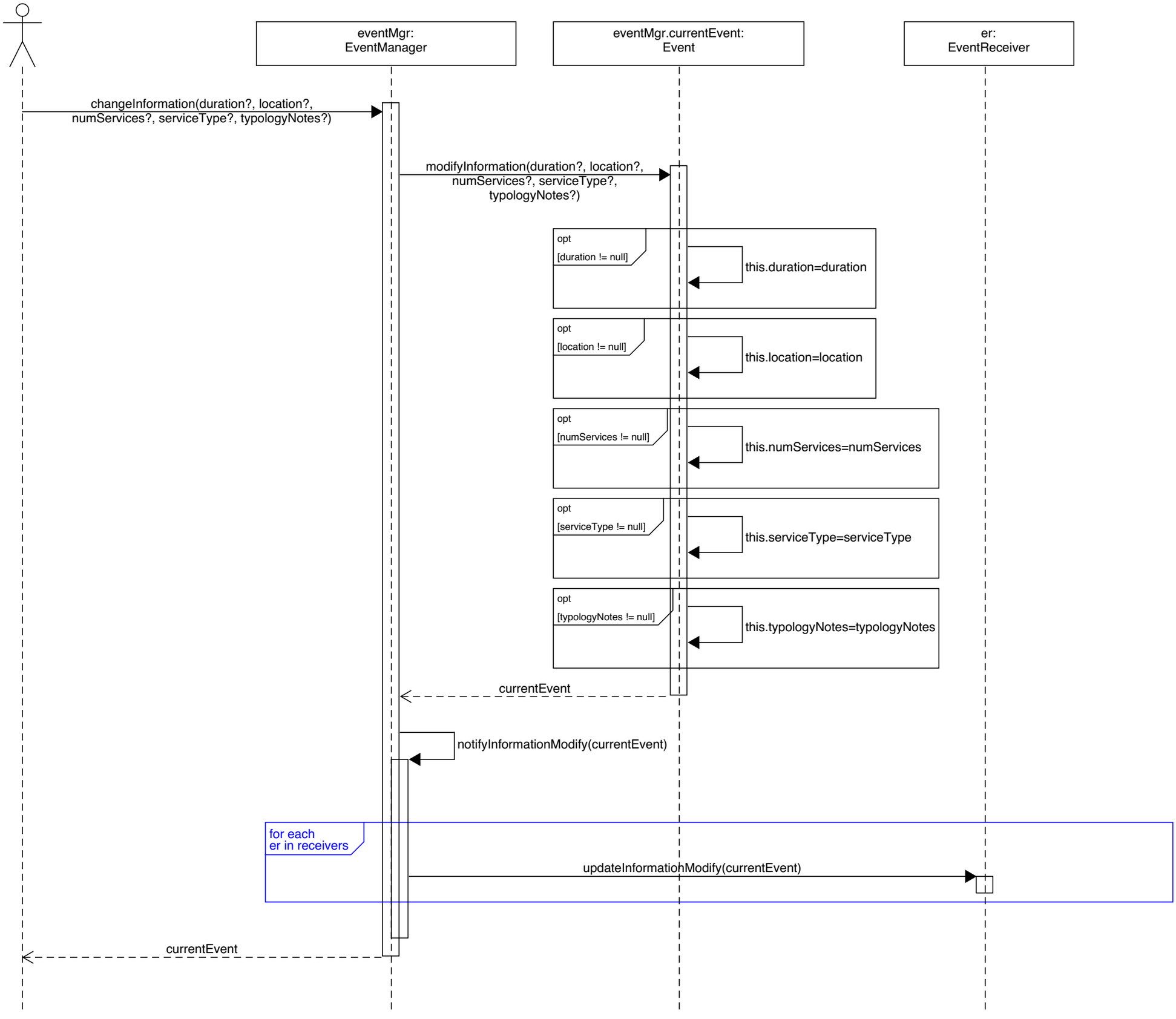
1d.1) createRecurringEvent



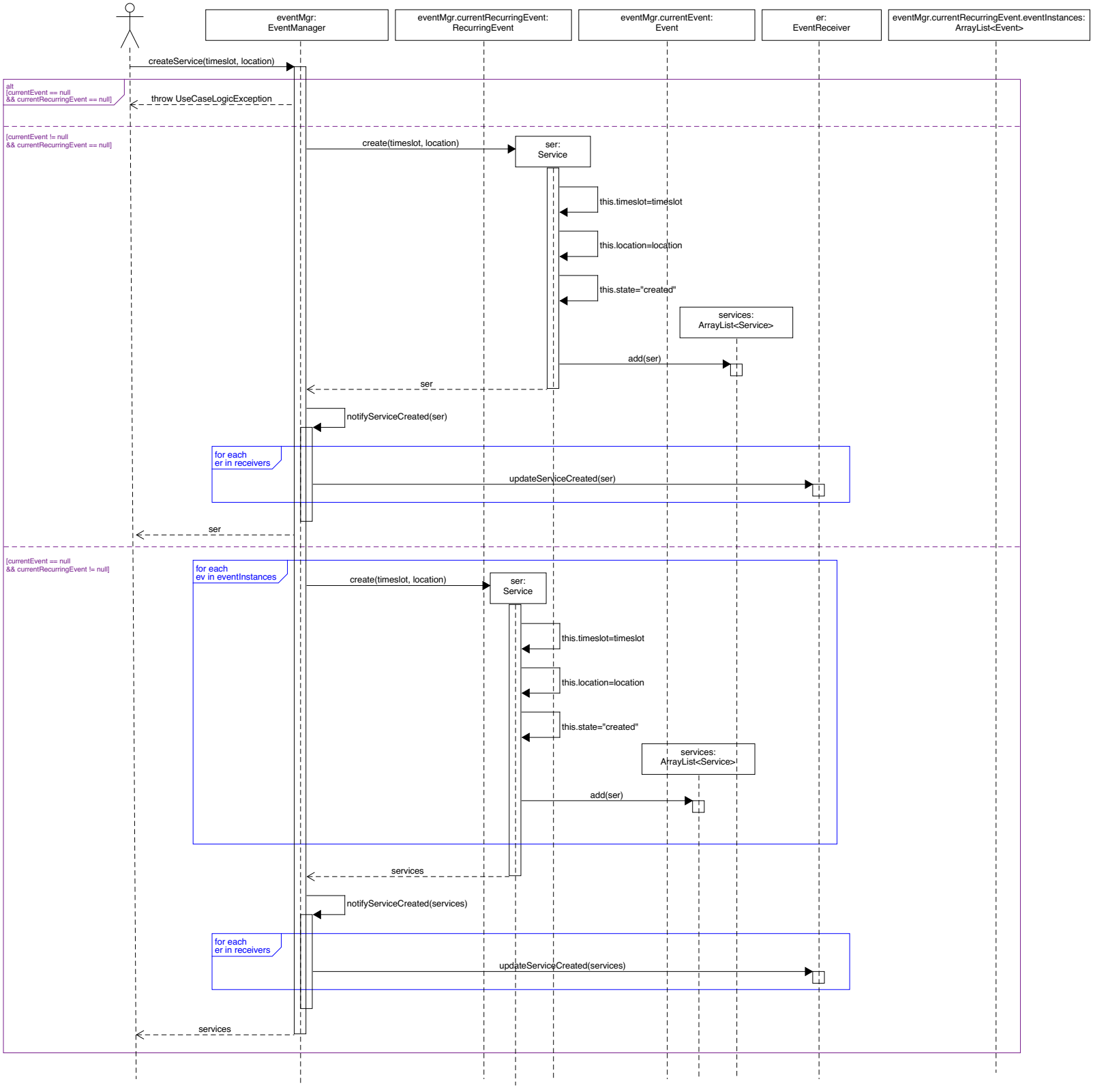
1e.2) changeEventInformation



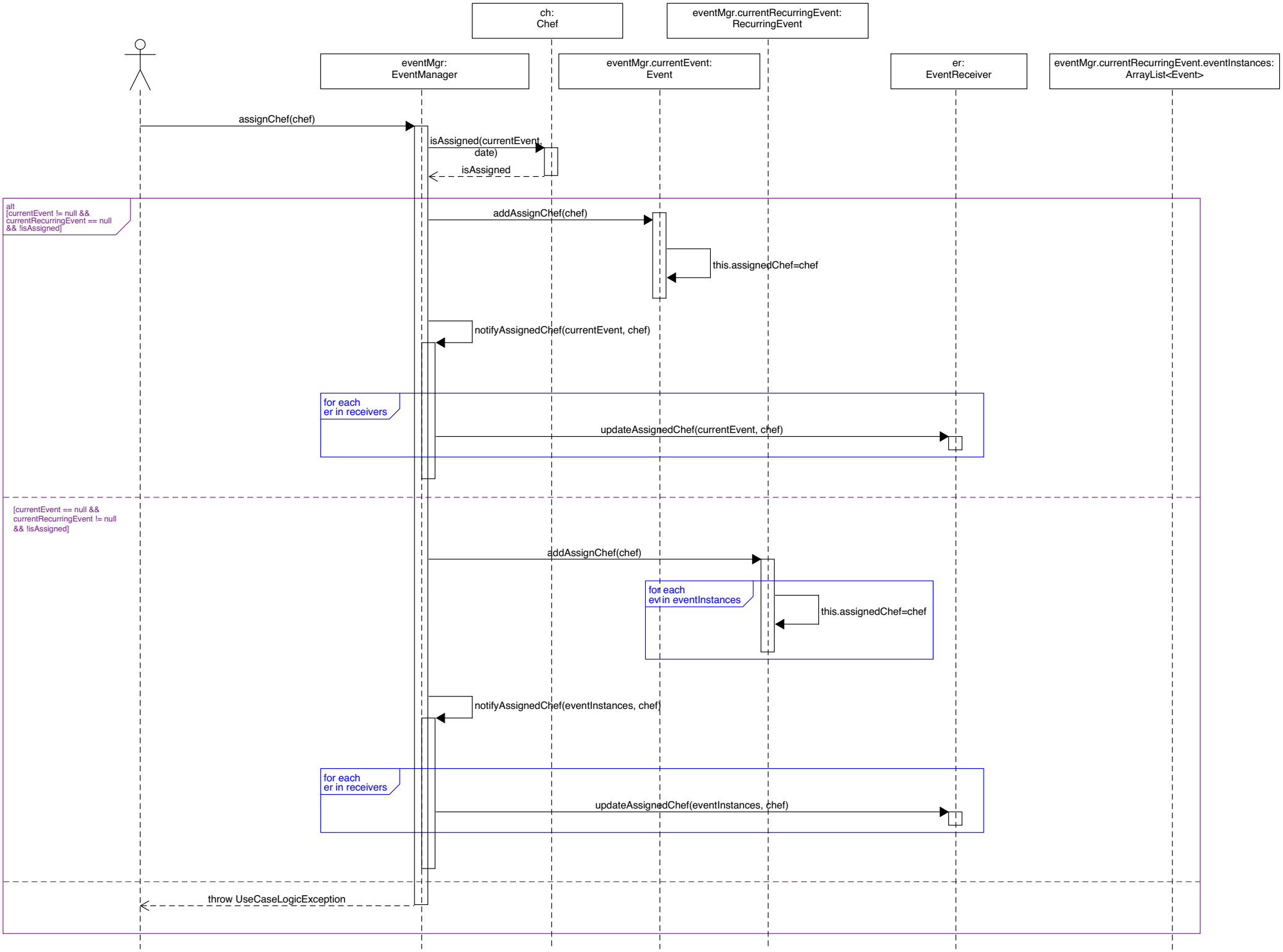
2a.1) changeInformation



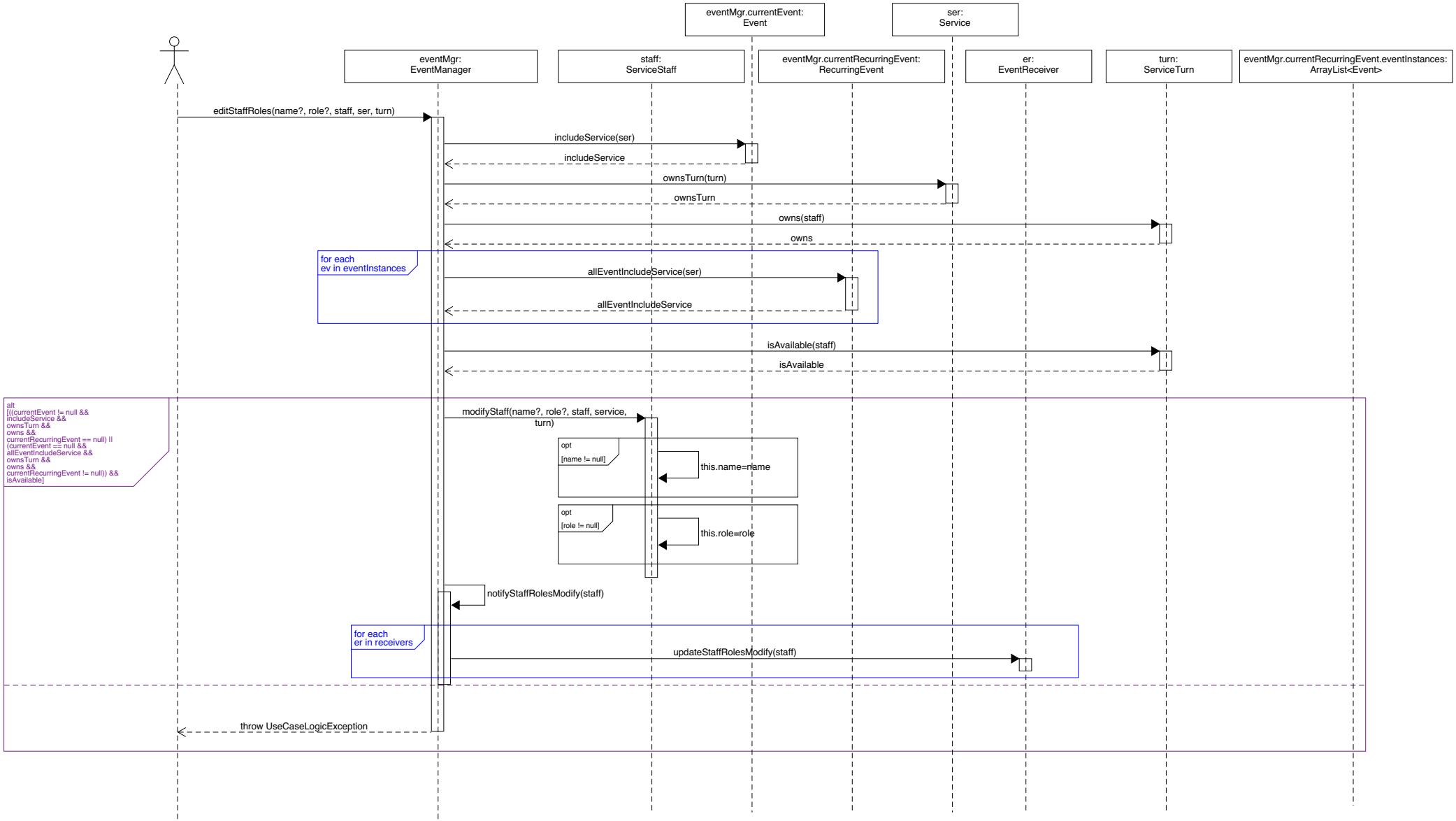
3) createService



4) assignChef



5a.1) editStaffRoles



Design Class Diagram (DCD)