

# PROGETTO SVILUPPO APPLICAZIONI SOFTWARE

*A.S. 22/23*

**GESTIRE GLI EVENTI**

**A CURA DI**

**MAGRÌ DANIELA, MATRICOLA 945982**

**SCICOLONE ALESSANDRO, MATRICOLA 953978**

# INDICE

---

- Use Case dettagliato:
  - Scenario principale 1s - 11s;
  - Estensioni 1a - 3a;
  - Estensioni 3b - 8a;
  - Estensioni 10a - 10ba, Eccezioni 1aa - 3s;
  - Eccezioni 4s - 6b;
  - Eccezioni 8s - 10ba;
- Modello di dominio;
- SSD:
  - Scenario di successo;
  - Estensioni del punto 1;
  - Estensioni del punto 2;
  - Estensioni del punto 3;
  - Estensioni del punto 4;
  - Estensioni del punto 5;
  - Estensioni del punto 6;
  - Estensioni del punto 8;
  - Eccezioni del punto 9;
  - Estensioni del punto 10;
- Contratti delle funzioni generate;
- DCD;
- DSD:
  - 2s creaEventCard, creaSchedaEvento;
  - 7s propostaModifiche;
  - 8s confirmMenu, confermaMenu;
  - 1a.1 createRecursiveEvent, creaEventoRicorrente;
  - 1a.2 modifyRecursiveEvent, modificaEventoRicorrente;
  - 1a.3 modifySingleEvent, modificaSingoloEventoInRicorrenza.

## Use Case dettagliato

---

# UC dettagliato

## Informazioni generali

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

**Portata:** Sistema;

**Livello:** Obiettivo utente;

**Attore primario:** Organizzatore;

**Parti interessate:** Chef, cuochi, personale di servizio;

**Pre-Condizioni:** L'attore deve essere autenticato come organizzatore;

**Garanzie di successo o post-condizioni:** L'evento è terminato.

## Scenario principale caso di successo

#	Attore	Sistema
1	Opzionalmente controlla lo stato della cucina	Fornito stato della cucina
2	Compila una scheda con le informazioni dell'evento	Creata scheda dell'evento
3	Opzionalmente modifica la scheda dell'evento	Modificata scheda dell'evento
4	Sceglie lo chef per l'evento	Chef per l'evento assegnato
5	Sceglie il personale di servizio	Personale di servizio assegnato
	<i>Ripete dal passo 5 fino a che non è soddisfatto</i>	
6	Opzionalmente sceglie un cuoco	Cuoco assegnato
	<i>Ripete dal passo 6 fino a che non è soddisfatto</i>	
7	Opzionalmente propone modifiche ad un menu'	Modifiche al menu' proposte
	<i>Ripete dal passo 7 fino a che non è soddisfatto</i>	
8	Conferma i menù per l'evento	Menu' confermati
9	Opzionalmente modifica la scheda dell'evento	Modificata scheda dell'evento
10	Termina l'evento	Terminato evento
11	Compila delle note riguardante l'evento	Compile note

### Estensione 1a

#	Attore	Sistema
<b>1a.1</b>	Crea l'evento ricorrente	Creato evento ricorrente
<b>1a.2</b>	Opzionalmente modifica i dati dell'evento ricorrente	Modificati dati evento ricorrente
<b>1a.3</b>	Opzionalmente modifica un singolo evento	Modificato singolo evento
	<i>Ripete dal passo 1a.2 fino a che non è soddisfatto</i>	
	<i>Il caso d'uso termina</i>	

### Estensione 1aa

#	Attore	Sistema
<b>1aa.1</b>	Apri l'evento ricorrente	Aperto evento ricorrente

### Estensione 2a

#	Attore	Sistema
<b>2a.1</b>	Apri la scheda dell'evento	Fornisce la scheda dell'evento

### Estensione 2b

#	Attore	Sistema
<b>2b.1</b>	Rifiuta la commessa	Rifiutata commessa
	<i>Il caso d'uso termina</i>	

### Estensione 3a

#	Attore	Sistema
<b>3a.1</b>	L'evento viene cancellato	Evento cancellato
	<i>Il caso d'uso termina</i>	

### Estensione3b

#	Attore	Sistema
3b.1	Rimanda l'evento	Evento rimandato
	<i>Il caso d'uso termina</i>	

### Estensione 4a

#	Attore	Sistema
4a.1	Modifica lo chef per l'evento	Chef per l'evento modificato

### Estensione 5a

#	Attore	Sistema
5a.1	Aggiunge il personale di servizio	Personale di servizio modificato

### Estensione 5b

#	Attore	Sistema
5b.1	Rimuove il personale di servizio	Personale di servizio rimosso

### Estensione 6a

#	Attore	Sistema
6a.1	Rimuove cuoco	Cuoco rimosso

### Estensione 6b

#	Attore	Sistema
6b.1	Modifica cuoco	Cuoco modificato

### Estensione 8a

#	Attore	Sistema
8a.1	Rifiuta i menù per l'evento	Menù rifiutati
	<i>Ripete dal passo 8a.1 fino a che non è soddisfatto</i>	
	<i>Quando lo chef sceglie i nuovi menu' torna al passo 7</i>	

### Estensione 10a

#	Attore	Sistema
10a.1	Disdice l'evento	Evento disdetto

### Estensione 10b

#	Attore	Sistema
10b.1	Modifica servizio	Servizio modificato
	<i>Ripete dal passo 10b.1 fino a che non è soddisfatto</i>	
10b.2	Controlla gli assegnamenti del personale di servizio, chef e cuochi	Forniti assegnamenti

### Estensione 10ba

#	Attore	Sistema
10ba.1	Modifica data	Data modificata

### Eccezione 1aa

#	Attore	Sistema
1aa.1	Apri l'evento ricorrente	L'attore non ha i permessi per poter visualizzare l'evento ricorrente

### Eccezione 2s

#	Attore	Sistema
2s.1	Compila una scheda con le informazioni dell'evento	L'attore che cerca di compilare la scheda informativa dell'evento non è l'organizzatore incaricato a gestire l'evento

### Eccezione 3s

#	Attore	Sistema
3s.1	Opzionalmente modifica la scheda dell'evento	L'attore non ha i permessi per poter modificare la scheda

### Eccezione 4s

#	Attore	Sistema
4s.1	Sceglie lo chef per l'evento	Lo chef richiesto è già occupato con un altro incarico

### Eccezione 4a

#	Attore	Sistema
4a.1	Modifica lo chef per l'evento	Lo chef richiesto è già occupato con un altro incarico

### Eccezione 5s

#	Attore	Sistema
5s.1	Sceglie il personale di servizio per l'evento	Il personale di servizio richiesto è già occupato con un altro incarico

### Eccezione 5a

#	Attore	Sistema
5a.1	Aggiungi il personale di servizio per l'evento	Il personale di servizio richiesto è già occupato con un altro incarico

### Eccezione 5b

#	Attore	Sistema
5b.1	Rimuovi il personale di servizio per l'evento	Il servizio ha un solo personale di servizio

### Eccezione 6s

#	Attore	Sistema
6s.1	Opzionalmente sceglie i cuochi	I cuochi richiesti sono già occupati con un altro incarico

### Eccezione 6b

#	Attore	Sistema
6b.1	Modifica i cuochi	I cuochi richiesti sono già occupati con un altro incarico



### Eccezione 8s

#	Attore	Sistema
<b>8s.1</b>	Conferma il menù per l'evento	Nessun menu' per l'evento è stato scelto

### Eccezione 8a

#	Attore	Sistema
<b>8a.1</b>	Rifiuta il menù per l'evento	Nessun menu' per l'evento è stato scelto

### Eccezione 9s

#	Attore	Sistema
<b>9s.1</b>	Opzionalmente modifica la scheda dell'evento	L'attore non ha i permessi per poter modificare la scheda

### Eccezione 10b

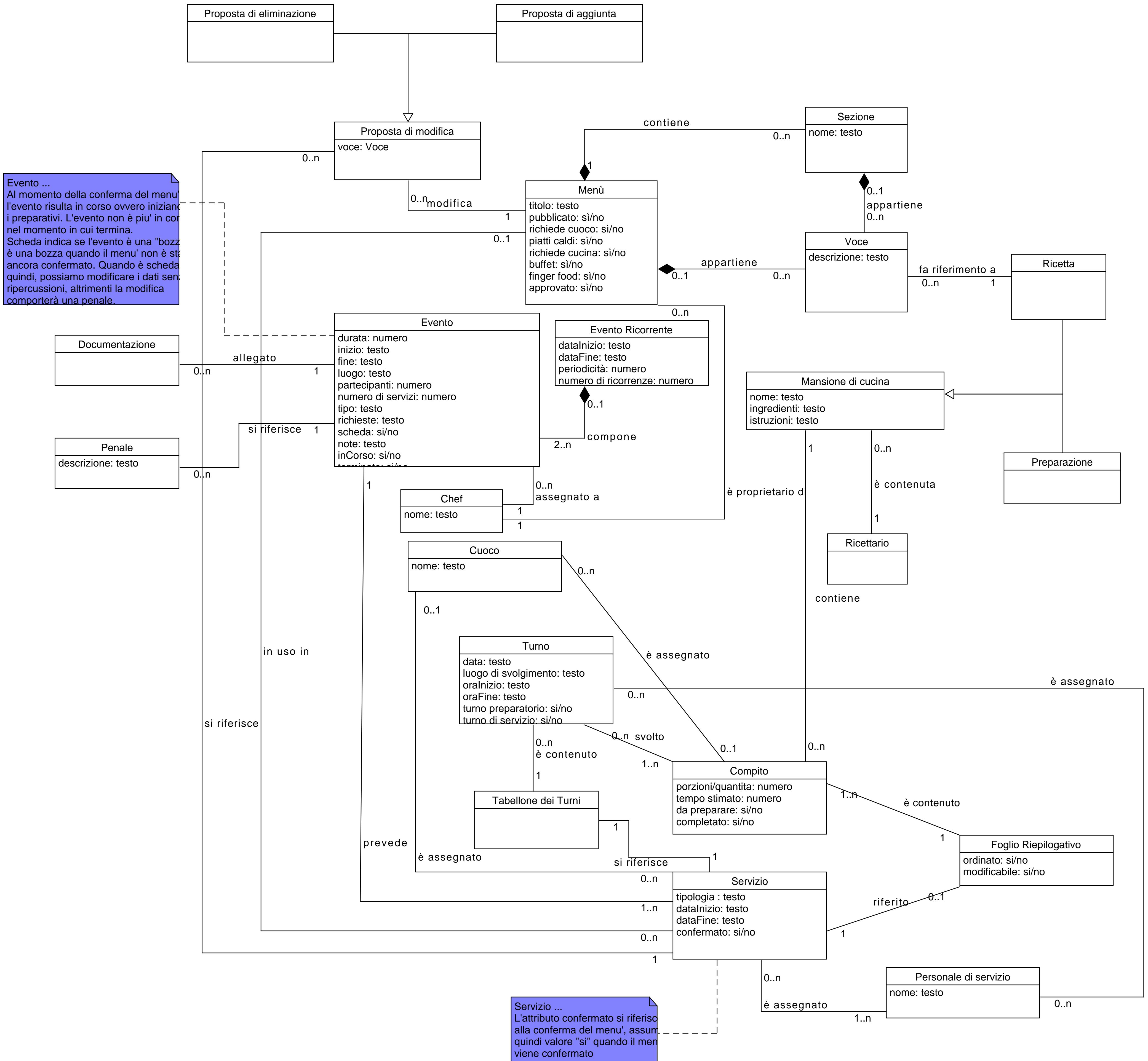
#	Attore	Sistema
<b>10b.1</b>	Modifica servizi	La richiesta di modifica precede meno di 7 giorni l'inizio dell'evento

### Eccezione 10ba

#	Attore	Sistema
<b>10ba.1</b>	Modifica data	La richiesta di modifica precede meno di 7 giorni l'inizio dell'evento

# MODELLO DI DOMINIO

---

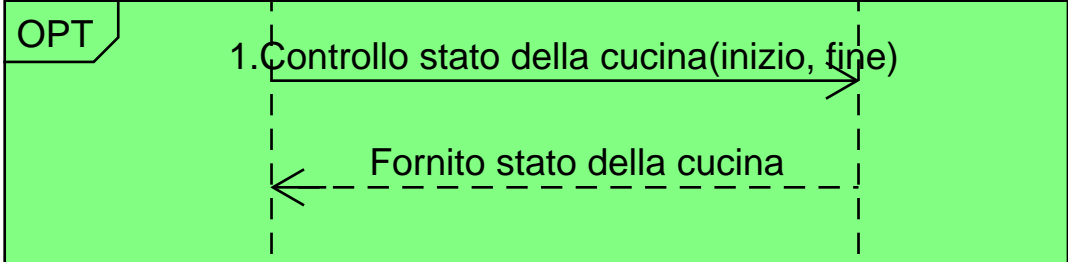


**SSD**

***Diagrammi di sequenza di  
sistema***

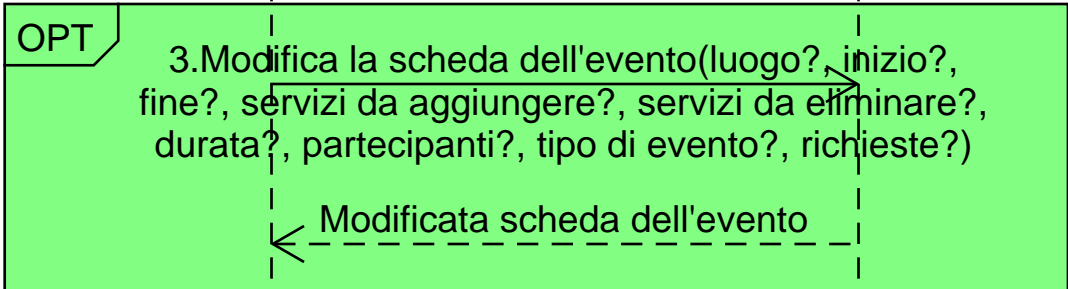
:Utente

:Sistema



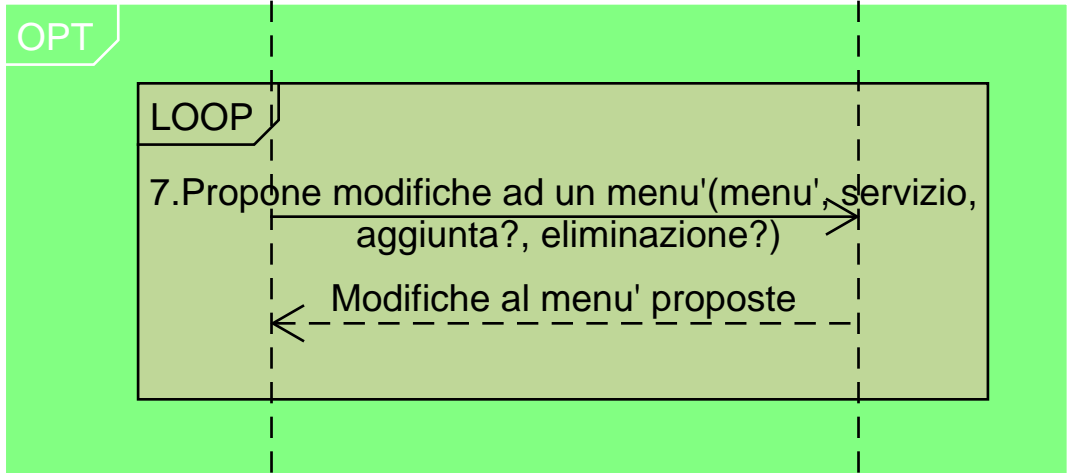
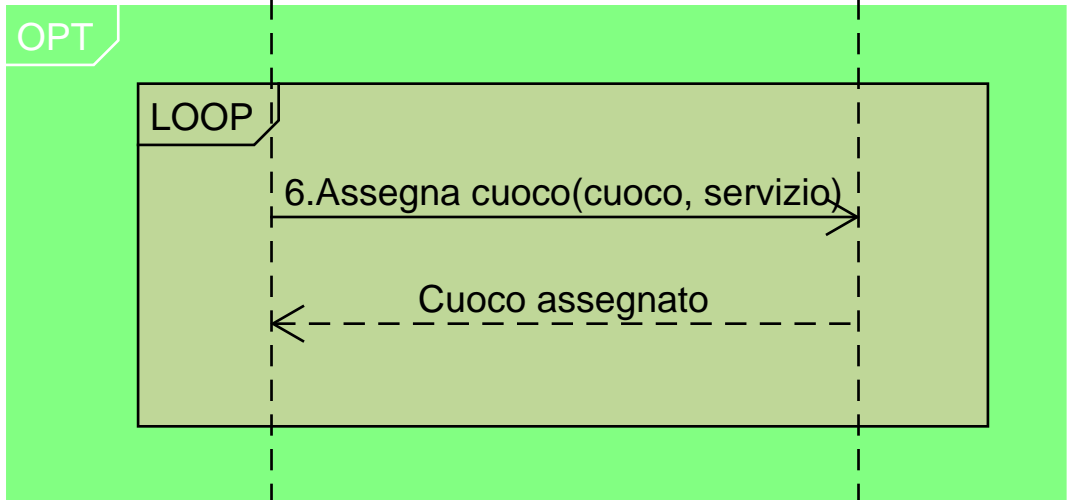
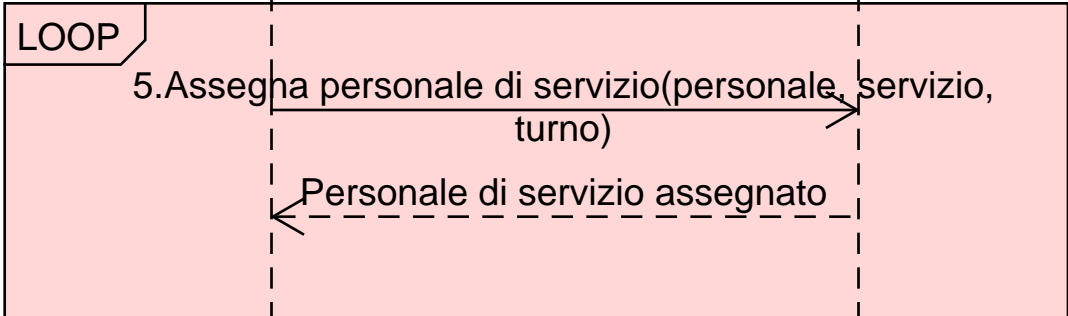
2. Crea la scheda dell'evento(luogo, inizio, fine, servizi, durata, partecipanti, tipo di evento, richieste)

Creato scheda dell'evento



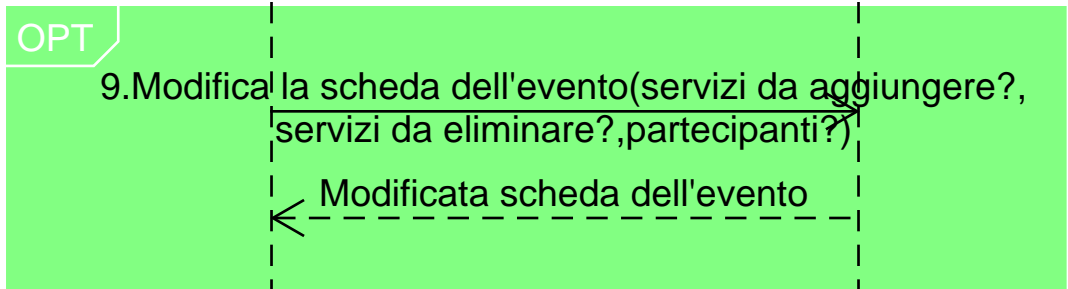
4. Assegna chef all'evento(chef)

Chef per l'evento assegnato



8. Conferma i menu'

Menu' confermati

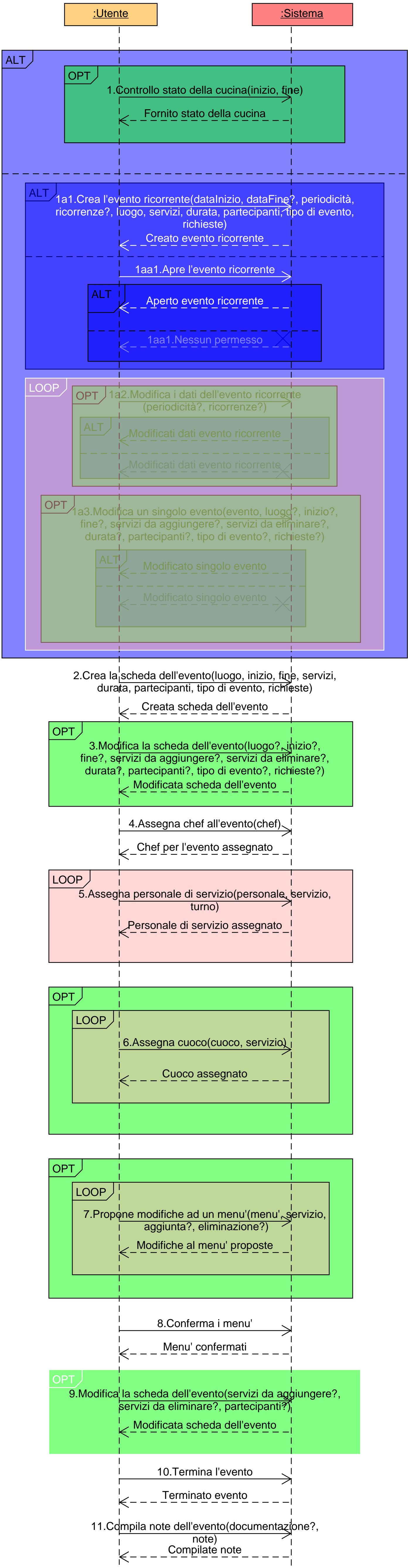


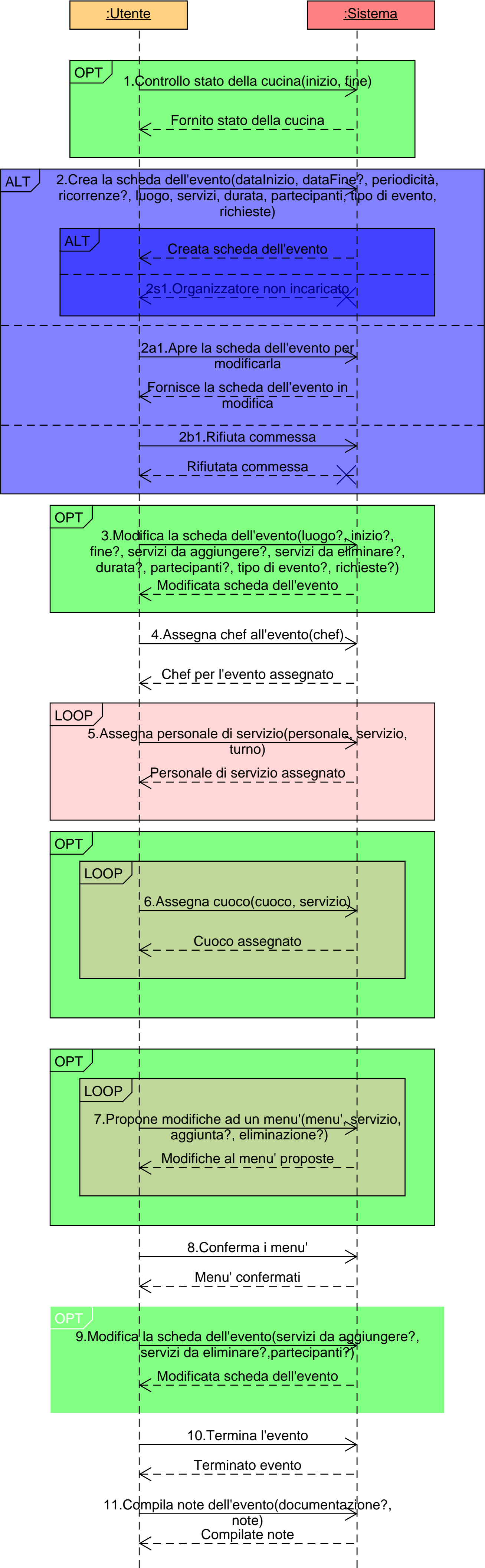
10. Termina l'evento

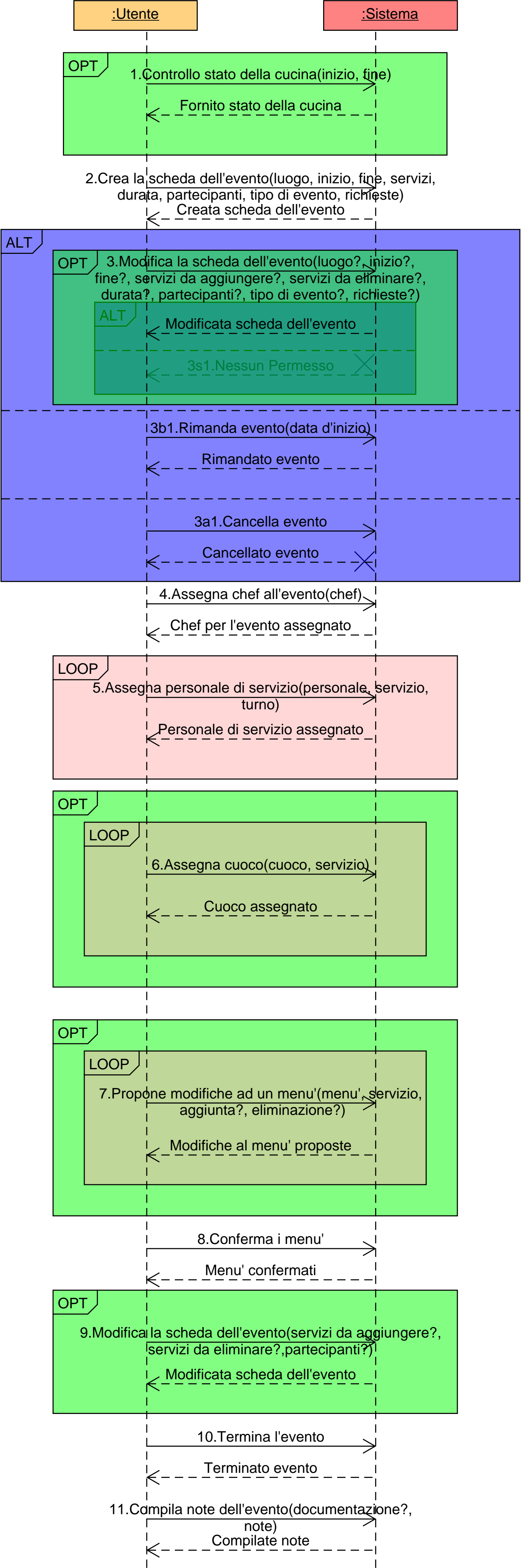
Terminato evento

11. Compila note dell'evento(documentazione?, note)

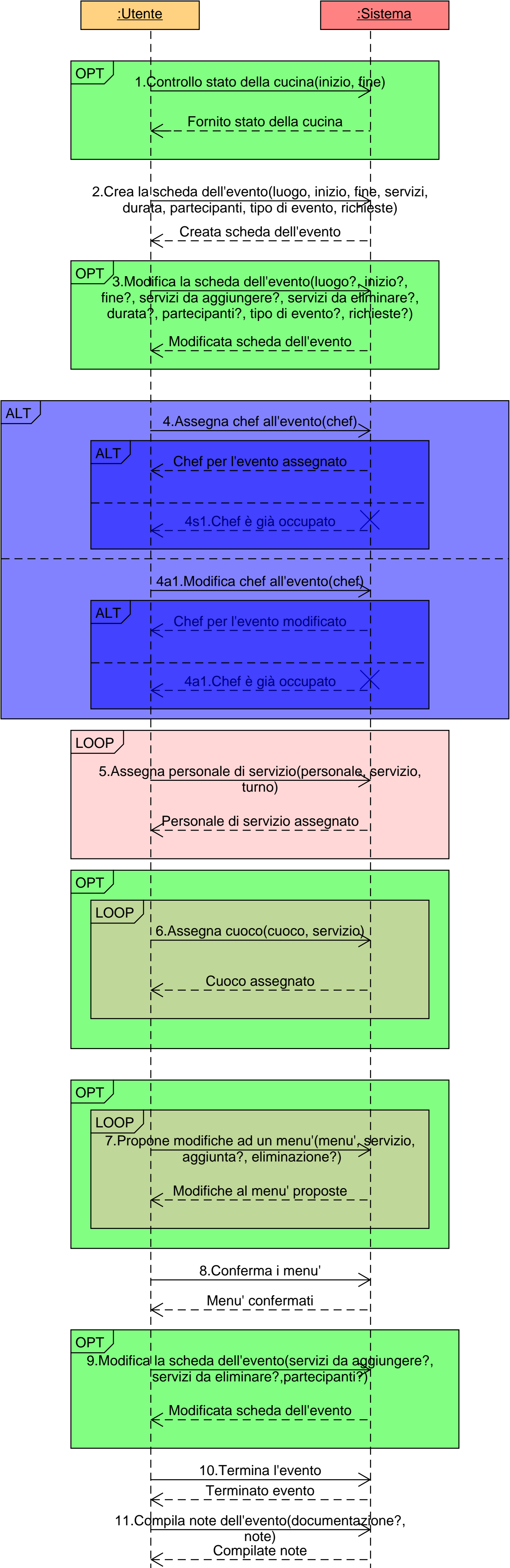
Compilate note

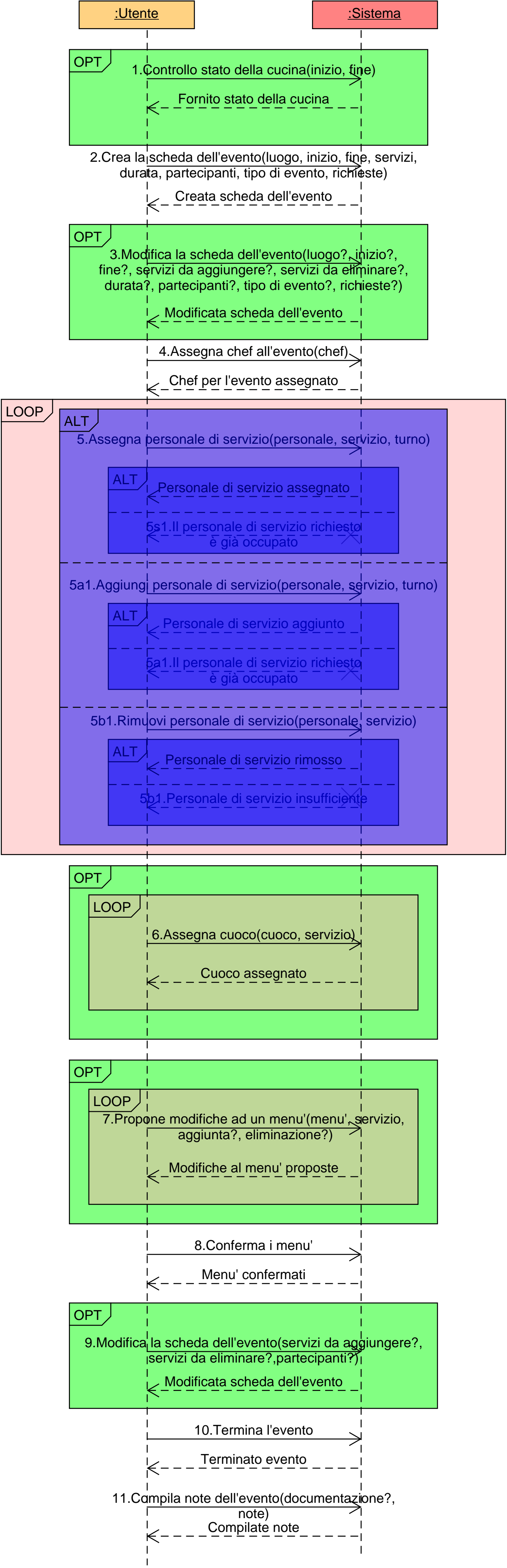


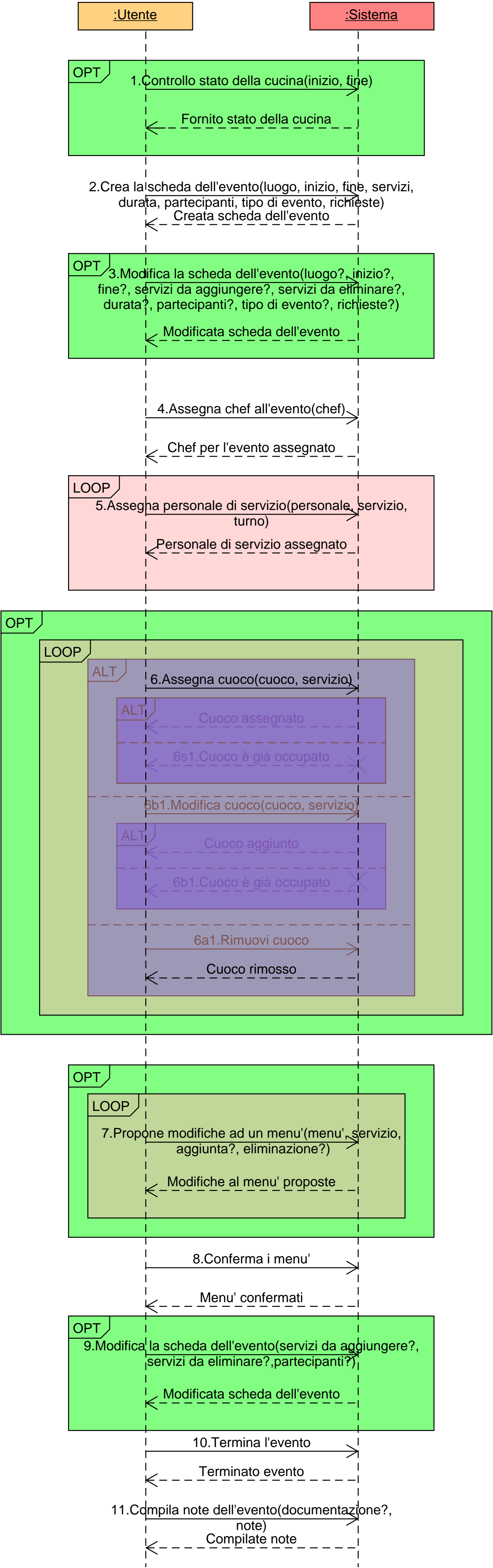


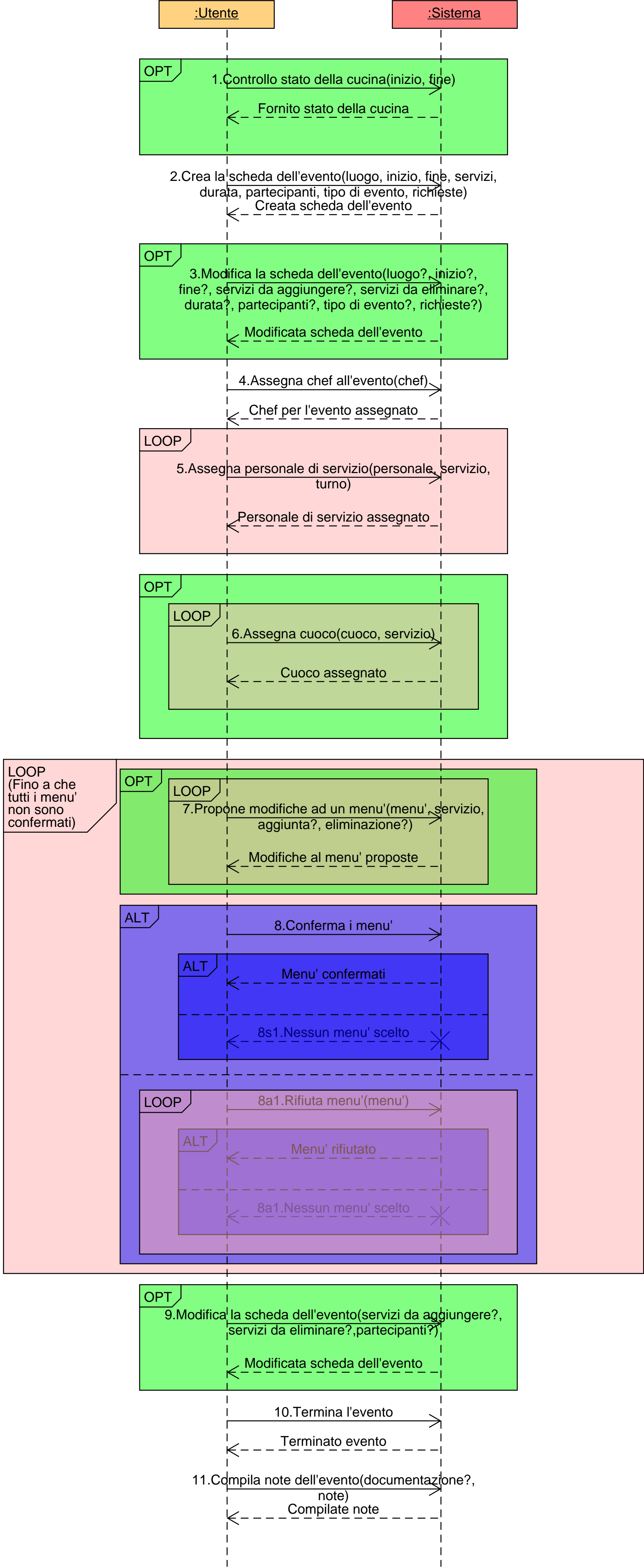


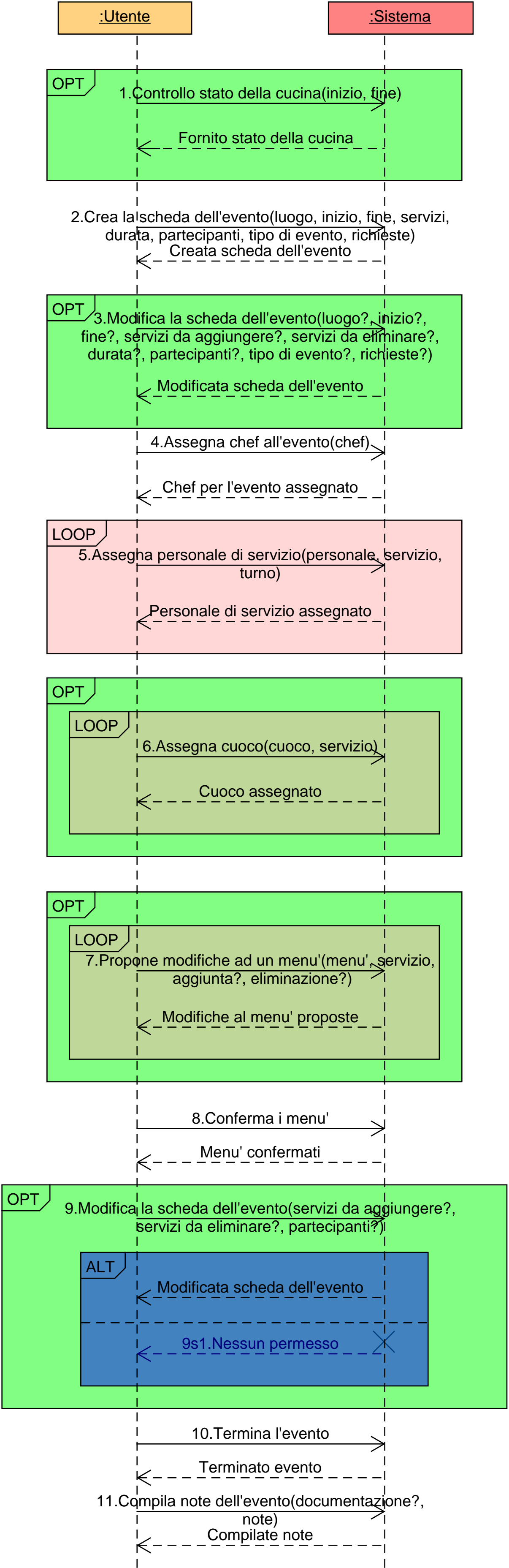


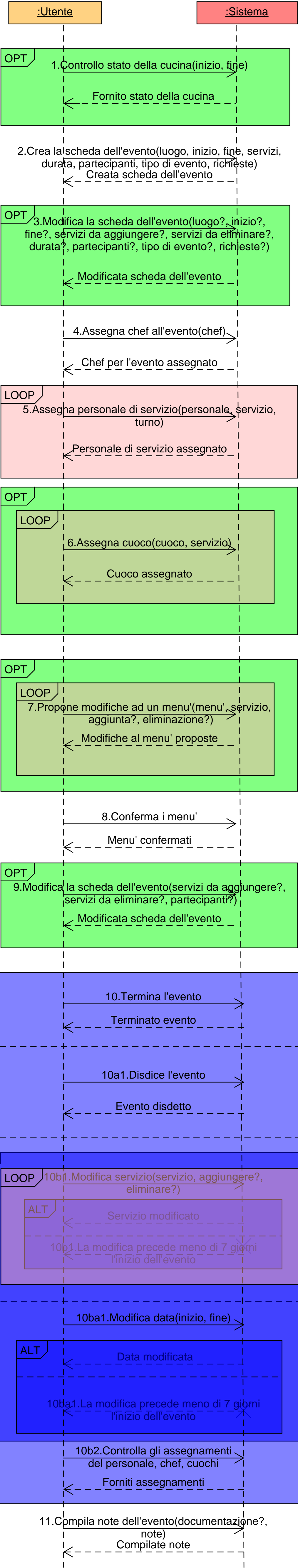












# Contratti

# Contratti

**Pre-condizione generale:** l'attore è identificato con un'istanza *boss* di Organizzatore, e la richiesta dell'evento *ev* è gestita da *boss*.

## Successo 1

**controlloStatoCucina**(inizio: testo, fine: testo)

**Pre-condizioni:** ;

**Post-condizioni:**

- È stata creata una lista *l* di disponibilità di chef, cuochi e personale di servizio.

## Successo 2

**creaSchedaEvento**(luogo: testo, inizio: testo, fine: testo, servizi richiesti: lista, durata: testo, partecipanti: numero, tipo evento: testo, richieste: testo)

**Pre-condizioni:** ;

**Post-condizioni:**

- È stata creata un'istanza *ev* di evento;
- *ev.scheda* = si;
- *ev.luogo* = luogo;
- *ev.inizio* = inizio;
- *ev.fine* = fine;
- *ev.durata* = durata;
- *ev.partecipati* = partecipanti;
- *ev.tipo* = tipo evento;
- *ev.richieste* = richieste;
- *ev.inCorso* = no;
- *ev.terminato* = no;
- Per ogni elemento *s* di servizi richiesti viene creata un'istanza *serv* di servizio ed *ev* prevede *s*.

## Successo 3

**modificaSchedaEvento**(luogo?: testo, inizio?: testo, fine?: testo, aggiungiServizi?: lista, rimuoviServizi?: lista, durata?: testo, partecipanti?: numero, tipo evento?: testo, richieste?: testo)

**Pre-condizioni:**

- [ se rimuoviServizi è specificato ] Per ogni elemento *s* di rimuoviServizi, esiste un'istanza *serv* di servizio dove *s* = *serv*;
- *ev.scheda* = si;
- *ev.inCorso* = no;

**Post-condizioni:**

- [ se luogo è specificato ] *ev.luogo* = luogo;
- [ se inizio è specificato ] *ev.inizio* = inizio;
- [ se fine è specificato ] *ev.fine* = fine;
- [ se durata è specificato ] *ev.durata* = durata;
- [ se partecipanti è specificato ] *ev.partecipati* = partecipanti;



- [ se tipo è specificato ]  $ev.tipo = tipo\_evento$ ;
- [ se rimuoviServizi è specificato ] Per ogni elemento  $s$  di rimuoviServizi,  $ev$  non prevede più  $s$ ;
- [ se aggiungiServizi è specificato ] Per ogni elemento  $s$  di aggiungiServizi, viene creata un'istanza  $serv$  di servizio ed  $ev$  prevede  $s$ ;
- [ se richieste è specificato ]  $ev.richieste = richieste$ .

## Successo 4

**assegnaChef(chef: chef)**

**Pre-condizioni:**

- Esiste un'istanza  $ch$  di chef dove chef =  $ch$ ;
- Per ogni istanza  $ev'$  di evento dove  $ev'.data = ev.data$  chef non è assegnato ad  $ev'$ ;

**Post-condizioni:**

- chef è assegnato ad  $ev$ .

## Successo 6

**assegnaCuoco(cuoco: cuoco, servizio: servizio)**

**Pre-condizioni:**

- Esiste un'istanza  $serv$  di servizio dove servizio =  $serv$ ;
- Esiste un'istanza  $c$  di cuoco dove cuoco =  $c$ ;
- Per ogni istanza  $serv'$  di servizio  $serv'.dataFine < servizio.dataInizio$  oppure  $serv'.dataInizio > servizio.dataFine$ ;

**Post-condizioni:**

- cuoco è assegnato a servizio.

## Successo 5

**assegnaPersonaleDiServizio(personale: personale di servizio, servizio: servizio, turno: turno)**

**Pre-condizioni:**

- Esiste un'istanza  $serv$  di servizio dove servizio =  $serv$ ;
- Esiste un'istanza  $s$  di personale di servizio dove personale =  $s$ ;
- Per ogni istanza  $serv'$  di servizio  $serv'.dataFine < servizio.dataInizio$  oppure  $serv'.dataInizio > servizio.dataFine$ ;
- Esiste un'istanza  $T$  di tabellone dei turni associata a  $serv$ ;
- Esiste un'istanza  $t$  di turno dove turno =  $t$ ;
- turno contenuto in  $T$ ;

**Post-condizioni:**

- personale è assegnato a servizio;
- personale è assegnato a turno.

## Successo 8

confermaMenu()

Pre-condizioni:

- Per ogni istanza *serv* di servizio associata ad *ev*:
  - Esiste un'istanza *m* di menu' associata a *serv*;
  - *m.approvato* = si;

Post-condizioni:

- Per ogni istanza *serv*:
  - *serv.confermato* = si;
- *ev.scheda* = no;
- *ev.inCorso* = si.

## Estensione 8a.1

rifiutaMenu(menu: menu')

Pre-condizioni:

- Esiste un'istanza *m* di menu' dove menu = *m*;
- Esiste un'istanza *serv* di servizio dove:
  - *serv* associata ad *ev*;
  - *serv* associata a menu;

Post-condizioni:

- *serv.confermato* = no;

## Successo 7

propostaModifiche(menu: menu', servizio: servizio, aggiunta?: voce, eliminazione?: voce)

Pre-condizioni:

- Esiste un'istanza *serv* di servizio dove servizio = *serv*;
- Esiste un'istanza *m* di menu' dove menu = *m*;
- servizio associato ad *ev*;
- menu associato a servizio;
- [ se aggiunta è specificato ] aggiunta non è associata a *m*;
- [ se eliminazione è specificato ] esiste un'istanza *v* di voce dove eliminazione = *v* e *v* è associata a *m*;

Post-condizioni:

- [ se aggiunta è specificato ] È stata creata un'istanza *pa* di proposta di aggiunta tale che:
  - *pa.voce* = aggiunta;
  - *pa* è associata a *m*;
  - *pa* è associata a *serv*;
- [ se eliminazione è specificato ] È stata creata un'istanza *pe* di proposta di eliminazione tale che:
  - *pe.voce* = eliminazione;
  - *pe* è associata a *m*;
  - *pe* è associata a *serv*;

## Successo 9

**modificaSchedaEvento(aggiungiServizi?: lista, rimuoviServizi?: lista, partecipanti?: numero)**

**Pre-condizioni:**

- [ se rimuoviServizi è specificato ] Per ogni elemento s di rimuoviServizi, esiste un'istanza *serv* di servizio dove s = *serv*;
- *ev.scheda* = no;
- *ev.inCorso* = si;

**Post-condizioni:**

- *ev.partecipati* = partecipanti;
- [ se aggiungiServizi è specificato ] Per ogni elemento s di aggiungiServizi, viene creata un'istanza *serv* di servizio ed *ev* prevede s;
- [ se rimuoviServizi è specificato ] Per ogni elemento s di rimuoviServizi, *ev* non prevede piu' s;

[ se partecipanti è specificato e differisce piu' del 30% da *ev.partecipati* ]

- È stata creata un'istanza *p* di penale;
- *p* è associata ad *ev*;
- *p.descrizione* = "la modifica dei partecipanti differisce piu' del 30%";

[ se aggiungiServizi o rimuoviServizi è specificato e la data della modifica è antecedente meno di 7 giorni da *ev.Inizio* ]

- È stata creata un'istanza *p'* di penale;
- *p'* è associata ad *ev*;
- *p'.descrizione* = "la data della modifica è antecedente meno di 7 giorni dall'inizio dell'evento";

## Successo 10

**terminaEvento()**

**Pre-condizioni:**

- *ev.inCorso* = si;
- La data corrente *d* in cui viene terminato l'evento *ev* dove *d* = *ev.fine*;

**Post-condizioni:**

- *ev.inCorso* = no;
- *ev.terminato* = si.

## Successo 11

**compilaNoteEvento(documentazione?: lista, note: testo)**

**Pre-condizioni:**

- *ev.inCorso* = no;
- *ev.scheda* = no;

**Post-condizioni:**

- [ se documentazione è specificato ] Per ogni elemento doc di documentazione doc è associata ad *ev*;
- *ev.note* = note.

## Estensione 1a.1

**creaEventoRicorrente**(dataInizio: testo, dataFine?: testo, periodicità: numero, numero\_ricorrenze?: numero, luogo: testo, servizi\_richiesti: lista, durata: testo, partecipanti: numero, tipo\_evento: testo, richieste: testo)

**Pre-condizioni:** ;

**Post-condizioni:**

- È stata creata un'istanza *evR* di evento ricorrente;
- *evR.dataInizio* = dataInizio;
- *evR.periodicità* = periodicità;
- [ se dataFine è specificato ] *evR.dataFine* = dataFine;
- [ se numero\_ricorrenze è specificato ] *evR.numero\_ricorrenze* = numero\_ricorrenze;
- [ se numero\_ricorrenze non è specificato ] viene calcolato il numero di ricorrenze *nr* ed *evR.numero\_ricorrenze* = *nr*;
- Sono state create istanze *ev* quanto è il *evR.numero\_ricorrenze*;
- Per ogni istanza *ev*:
  - *ev* associato ad *evR*;
  - *ev.scheda* = si;
  - *ev.luogo* = luogo;
  - viene calcolato l'inizio *in* ed *ev.inizio* = *in*;
  - viene calcolato la fine *fi* ed *ev.fine* = *fi*;
  - *ev.durata* = durata;
  - *ev.partecipati* = partecipanti;
  - *ev.tipo* = tipo\_evento;
  - *ev.richieste* = richieste;
  - *ev.inCorso* = no;
  - *ev.terminato* = no;
  - Per ogni elemento *s* di servizi\_richiesti viene creata un'istanza *serv* di servizio ed *ev* prevede *s*.

## Estensione 1a.2

**modificaEventoRicorrente**(periodicità?: numero, numero\_ricorrenze?: numero)

**Pre-condizioni:**

- *ev* è un'istanza *evR* di evento ricorrente;

**Post-condizioni:**

[ se periodicità è specificato ] :

- *evR.periodicità* = periodicità;
- Per ogni elemento *ev* associato ad *evR*, sia la data corrente *d*, vengono modificati gli elementi per cui vale *ev.inizio* >= *d*:
  - viene calcolato l'inizio *in* ed *ev.inizio* = *in*;
  - viene calcolato la fine *fi* ed *ev.fine* = *fi*;

[ se numero\_ricorrenze è specificato ] :

[ se *evR.numero\_ricorrenze* > numero\_ricorrenze ]:

- Vengono eliminate le ultime istanze *ev* quanto differisce numero\_ricorrenze da *evR.numero\_ricorrenze*;
- *evR.numero\_ricorrenze* = numero\_ricorrenze;

[ se  $evR.numero\_ricorrenze < numero\_ricorrenze$  ] :

- Sono state create istanze  $ev\_copy$  quanto differisce  $evR.numero\_ricorrenze$  da  $numero\_ricorrenze$  dove, preso l'elemento  $ev$  in  $evR$ , non esiste un elemento  $ev'$  per cui  $ev'.inizio < ev.inizio$ ;
  - Per ogni istanza  $ev\_copy$ :
    - $ev\_copy = ev$ ;
  - $evR.numero\_ricorrenze = numero\_ricorrenze$ ;
- viene calcolata la data fine  $df$  ed  $evR.dataFine = df$ ;

### Estensione 1a.3

**modificaSingoloEventoInRicorrenza(evento: evento, luogo?: testo, inizio?: testo, fine?: testo, aggiungiServizi?: lista, rimuoviServizi?: lista, durata?: testo, partecipanti?: numero, tipo\_evento?: testo, richieste?: testo)**

**Pre-condizioni:**

- [ se rimuoviServizi è specificato ] Per ogni elemento  $s$  di rimuoviServizi, esiste un'istanza  $serv$  di servizio dove  $s = serv$ ;
- $ev$  è un'istanza  $evR$  di evento ricorrente;
- Esiste un'istanza  $ev'$  di evento associata ad  $evR$  dove  $ev' = evento$ ;
- $ev'.scheda = si$ ;
- $ev'.inCorso = no$ ;

**Post-condizioni:**

- [ se luogo è specificato ]  $ev.luogo = luogo$ ;
- [ se inizio è specificato ]  $ev.inizio = inizio$ ;
- [ se fine è specificato ]  $ev.fine = fine$ ;
- [ se durata è specificato ]  $ev.durata = durata$ ;
- [ se partecipanti è specificato ]  $ev.partecipati = partecipanti$ ;
- [ se tipo è specificato ]  $ev.tipo = tipo\_evento$ ;
- [ se aggiungiServizi è specificato ] Per ogni elemento  $s$  di aggiungiServizi, viene creata un'istanza  $serv$  di servizio ed  $ev$  prevede  $s$ ;
- [ se rimuoviServizi è specificato ] Per ogni elemento  $s$  di rimuoviServizi,  $ev$  non prevede più  $s$ ;
- [ se richieste è specificato ]  $ev.richieste = richieste$ .

### Estensione 1aa.1

**apriEventoRicorrente()**

**Pre-condizioni:**

- Esiste un'istanza  $evR$  di evento ricorrente dove  $ev = evR$  ;

**Post-condizioni:** .

### Estensione 2a.1

**apriEvento()**

**Pre-condizioni:**

- Esiste un'istanza  $ev'$  di evento dove  $ev' = ev$ ;

**Post-condizioni:** .

## Estensione 2b.1

rifiutaEvento()

Pre-condizioni: ;

Post-condizioni: .

## Estensione 3a.1

cancellaEvento()

Pre-condizioni:

- $ev.scheda = si$ ;
- $ev.inCorso = no$ ;

Post-condizioni:

- Per ogni istanza  $serv$  di servizio associata ad  $ev$ ,  $serv$  viene eliminata;
- Viene eliminata l'istanza  $ev$ .

## Estensione 3b.1

rimandaEvento(inizio: testo)

Pre-condizioni:

- $ev.scheda = si$ ;
- $ev.inCorso = no$ ;

Post-condizioni:

- $ev.inizio = \underline{inizio}$ ;
- viene calcolato la fine  $\underline{fi}$  ed  $ev.fine = \underline{fi}$ ;

## Estensione 4a.1

modificaChef(chef: chef)

Pre-condizioni:

- Esiste un'istanza  $ch$  di chef assegnato ad  $ev$ ;
- Esiste un'istanza  $ch'$  di chef dove  $\underline{chef} = ch'$ ;
- Per ogni istanza  $ev'$  di evento dove  $ev'.data = ev.data$   $\underline{chef}$  non è assegnato ad  $ev'$ ;

Post-condizioni:

- $ch$  non è piu' associato ad  $ev$ ;
- $\underline{chef}$  è associato ad  $ev$ ;

## Estensione 5a.1

aggiungiPersonaleDiServizio(personale: personale di servizio, servizio: servizio, turno: turno)

Pre-condizioni:

- Esiste un'istanza  $serv$  di servizio dove  $\underline{servizio} = serv$ ;
- Esiste almeno un'istanza  $s$  di personale di servizio associato a  $\underline{servizio}$ ;
- Esiste un'istanza  $s'$  di personale di servizio dove  $\underline{personale} = s'$ ;
- Per ogni istanza  $serv'$  di servizio  $serv'.dataFine < \underline{servizio.dataInizio}$  oppure  $serv'.dataInizio > \underline{servizio.dataFine}$ ;
- Esiste un'istanza  $T$  di tabellone dei turni associata a  $serv$ ;
- Esiste un'istanza  $t$  di turno dove  $\underline{turno} = t$ ;

- turno contenuto in  $T$ ;

**Post-condizioni:**

- personale è assegnato a servizio;
- personale è assegnato a turno.

## Estensione 5b.1

**rimuoviPersonaleDiServizio(personale: personale di servizio, servizio: servizio)**

**Pre-condizioni:**

- Esiste un'istanza  $serv$  di servizio dove servizio =  $serv$ ;
- servizio è associato a due o piu' istanze di personale di servizio;
- Esiste un'istanza  $s$  di personale di servizio dove  $s$  = personale;
- personale è associato a servizio;
- Esiste un'istanza  $T$  di tabellone dei turni associata a servizio;
- Esiste almeno un'istanza  $t$  di turno per cui:
  - $t$  contenuto in  $T$ ;
  - $t$  associato a personale;

**Post-condizioni:**

- personale non è piu' associato a servizio;
- Per ogni  $t$ , personale non è piu' associato a  $t$ ;

## Estensione 6b.1

**modificaCuoco(cuoco: cuoco, servizio: servizio)**

**Pre-condizioni:**

- Esiste un'istanza  $serv$  di servizio dove servizio =  $serv$ ;
- Esiste un'istanza  $c$  di cuoco assegnato a servizio;
- Esiste un'istanza  $c'$  di cuoco dove cuoco =  $c'$ ;
- Per ogni istanza  $serv'$  di servizio  $serv'.dataFine < \underline{servizio.dataInizio}$  oppure  $serv'.dataInizio > \underline{servizio.dataFine}$ ;

**Post-condizioni:**

- $c$  non è piu' associato a servizio;
- cuoco è assegnato a servizio;

## Estensione 6a.1

**rimuoviCuoco(cuoco: cuoco, servizio: servizio)**

**Pre-condizioni:**

- Esiste un'istanza  $serv$  di servizio dove servizio =  $serv$ ;
- Esiste un'istanza  $c$  di cuoco dove  $c$  = cuoco;
- cuoco è associato a servizio;

**Post-condizioni:**

- cuoco non è piu' associato a servizio;

## Estensione 10a.1

**disdiceEvento()**

**Pre-condizioni:**

- *ev.scheda* = no;
- *ev.inCorso* = si;

**Post-condizioni:**

- È stata creata un'istanza *p* di penale;
- *p* è associata ad *ev*;
- *p.descrizione* = "la richiesta di disdetta è avvenuta mentre l'evento è già in corso";
- *ev.terminato* = no;
- *p.descrizione* = "la richiesta di disdetta è avvenuta mentre l'evento è già in corso";

## Estensione 10ba.1

**modificaData(inizio: testo, fine: testo)**

**Pre-condizioni:**

- La data della modifica è antecedente meno di 7 giorni da *ev.Inizio*;
- *ev.scheda* = no;
- *ev.inCorso* = si;

**Post-condizioni:**

- *ev.inizio* = inizio;
- *ev.fine* = fine;
- viene calcolato la durata d ed *ev.durata* = d;

## Estensione 10b.1

**modificaServizio(servizio: servizio, daAggiungere?: si/no, daEliminare?: si/no)**

**Pre-condizioni:**

- La data della modifica è antecedente meno di 7 giorni da *ev.Inizio*;
- *ev.scheda* = no;
- *ev.inCorso* = si;
- [se daEliminare è specificato] Esiste un'istanza *serv* di servizio dove servizio = *serv* e servizio associato ad *ev*;

**Post-condizioni:**

- [se daEliminare è specificato] servizio non è piu' associato ad *ev*;
- [se daAggiungere è specificato] servizio è associato ad *ev*.

## Estensione 10b.2

**controlloAssegnamenti()**

**Pre-condizioni:**

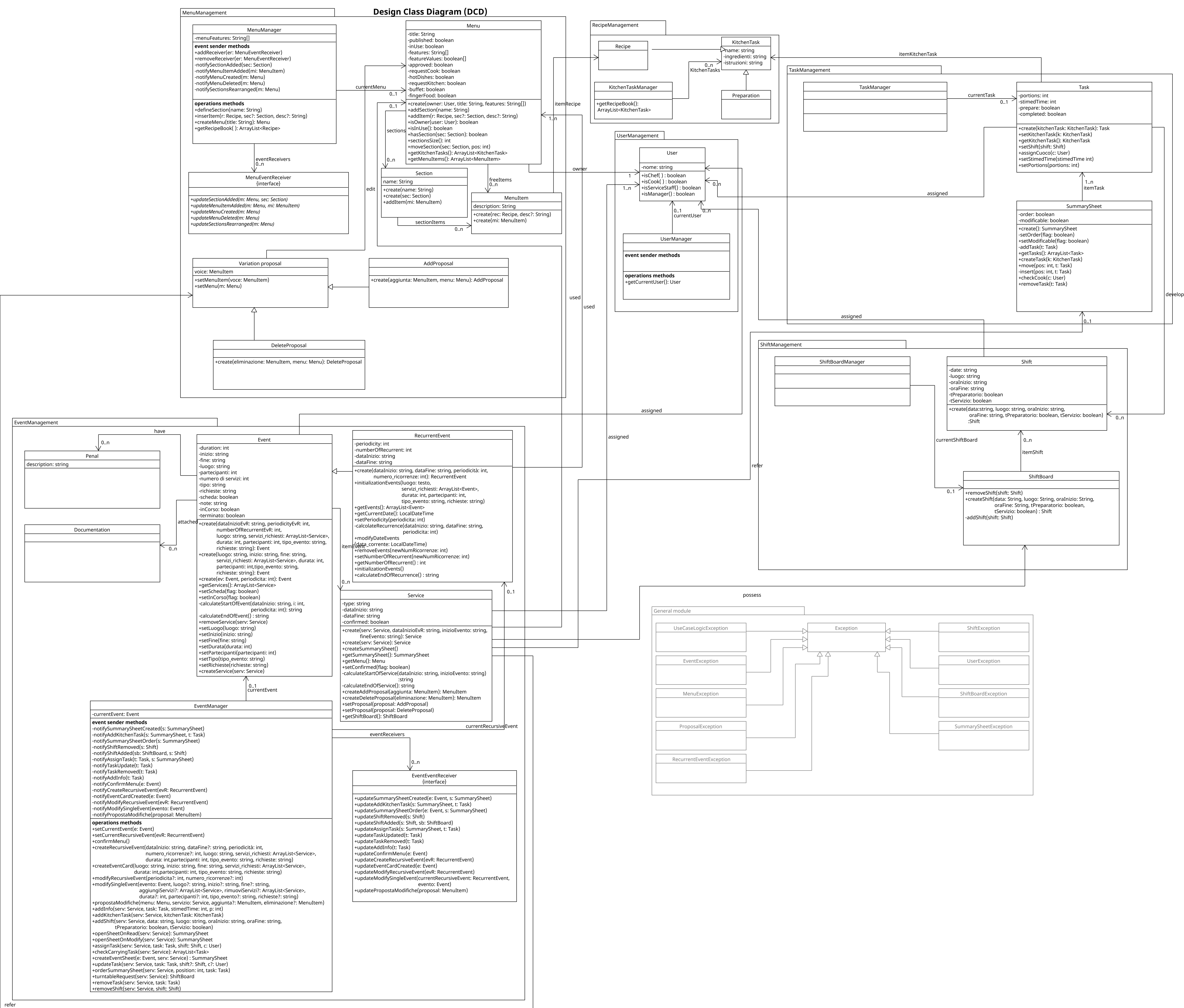
**Post-condizioni:**

- È stata creata una lista *l* di assegnamenti di chef, cuochi e personale di servizio;



DCD

*Diagramma delle classi*



**DSD**

***Diagrammi di sequenza***

