

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;