**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 *evR* di Evento Ricorrente
* *evR.frequenza* = NULL
* *evR.numeroRipetizioni* = NULL
* *evR.dataFine* = NULL
* è stata creata un’istanza *ev* di Evento
* *ev.data =* data
* *ev.numeroPartecipanti =* numeroPartecipanti
* *ev.stato =* NULL
* *org* **è proprietario di** *ev*

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

**Pre-condizioni:** ---

**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 *evC* **include** *ser* è eliminato
* ogni Servizio *ser* tale che *evS* **include** *ser* è eliminato
* ogni Lavoro Preparatorio *lav* tale che *evC* **prevede** *lav* è eliminato
* ogni Lavoro Preparatorio *lav* tale che *evS* **prevede** *lav* è eliminato
* *evS* è eliminato
* *evC* è eliminato
* eventoè eliminato

**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 Lavoro Preparatorio *lav* tale che evento **prevede** *lav* è eliminato
* eventoè eliminato
* L’Evento Ricorrente *evR* che contiene evento è eliminato

**1c.1** **annullaEventoServizio(evento?: Evento, servizio?: Servizio)**

**Pre-condizioni:** ---

**Post-condizioni:**

* [se è specificato un evento] evento*.stato =* annullato
* [se è specificato un servizio] servizio*.stato =* annullato

**1d.1** **creaEventoRicorrente(numeroPartecipanti: numero)**

**Pre-condizioni:** ---

**Post-condizioni:**

* è stata creata un’istanza *evR* di Evento Ricorrente
* è stata creata un’istanza *ev* di Evento
* *ev.numeroPartecipanti =* numeroPartecipanti
* *ev.stato =* NULL
* *org* **è proprietario di** *ev*
* *evR* **contiene** *ev*

**1d.2** **aggiungiInformazioniRicorrenza(frequenza: testo, numeroRipetizioni?: numero, dataFine?: testo)**

**Pre-condizioni:**

* uno tra numeroRipetizioni e dataFine dev’essere definito

**Post-condizioni:**

* *evR.frequenza =* frequenza
* [se è specificato un numeroRipetizioni] *evR.numeroRipetizioni =* numeroRipetizioni
* [se è specificata una dataFine] *evR.dataFine =* dataFine

**1e.1** **apriEventoRicorrente(eventoRicorrente: Evento Ricorrente)**

**Pre-condizioni:** ---

**Post-condizioni:** ---

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

**Pre-condizioni:**

* uno tra numRipetizioni e dataFine dev’essere definito

**Post-condizioni:**

* SecriterioIstanza = singola:
  + [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
  + SecriterioIstanza = intero:
  + per ogni Evento *ev* tale che eventoRicorrente **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] eventoRicorrente.*frequenza* =frequenza
    - [se è specificato un numRipetizioni] eventoRicorrente.*numRipetizioni* =numRipetizioni
    - [se è specificata una dataFine] eventoRicorrente.*dataFine* =dataFine
  + SecriterioIstanza = successive:
  + 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] eventoRicorrente.*frequenza* =frequenza
    - [se è specificato un numRipetizioni] eventoRicorrente.*numRipetizioni* =numRipetizioni
    - [se è specificata una dataFine] eventoRicorrente.*dataFine* =dataFine

**1e.3** **apriEventoRicorrente(istanza: Evento, eventoRicorrente: Evento Ricorrente, criterioIstanza: testo)**

**Pre-condizioni:** ---

**Post-condizioni:** Se*org* **è proprietario di** istanza e istanza*.stato* != inCorso:

* SecriterioIstanza = singola:
* ogni Servizio *ser* tale che istanza **include** *ser* è eliminato
* ogni Lavoro Preparatorio *lav* tale che istanza **prevede** *lav* è eliminato
* istanza è eliminata
* SecriterioIstanza = intero:
* per ogni Evento *ev* tale che eventoRicorrente **contiene** *ev*:
* ogni Servizio *ser* tale che *ev* **include** *ser* è eliminato
* ogni Lavoro Preparatorio *lav* tale che *ev* **prevede** *lav* è eliminato
* *ev* è eliminato
* eventoRicorrente è eliminato
* SecriterioIstanza = 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 Lavoro Preparatorio *lav* tale che *ev* **prevede** *lav* è eliminato
* *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*

**Post-condizioni:**

* *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** **assegnaChef(chef: Chef)**

**Pre-condizioni:**

* è in corso la definizione di un Evento *ev*

**Post-condizioni:**

* chef è **assegnato a** *ev*

**3a.1** **modificaChef(chef: Chef)**

**Pre-condizioni:**

* è in corso la definizione di un Evento *ev*

**Post-condizioni:** Se chef non è **assegnato a** alcun Evento:

* chef è **assegnato a** *ev*

**4** **assegnaRuoliPersonale(nome: testo, ruolo: testo, personale: PersonaleDiServizio)**

**Pre-condizioni:**

* è in corso la definizione di un Evento *ev*
* è in corso la definizione di un Turno di Servizio *turSer*
* è in corso la definizione di un Servizio *ser*

**Post-condizioni:** Se *ser.fasciaOraria* == *turSer.orario*:

* personale*.nome* =nome
* personale*.ruolo* = ruolo
* è stata creata un’istanza *turSer* di Turno di Servizio
* personaleè **specificato in** *turSer*
* è stata creata un’istanza *ser* di Servizio
* *ser* **possiede** *turSer*
* *ev* **include** *ser*

**4a.1** **modificaRuoliPersonale(nome?: testo, ruolo?: testo, servizio: Servizio, personale: PersonaleDiServizio, turno: TurnoDiServizio)**

**Pre-condizioni:**

* è in corso la definizione di un Evento *ev* e *ev* **include** servizio
* servizio **possiede** turno
* personale è **specificato in** turno

**Post-condizioni:** Se servizio*.fasciaOraria* ==turno*.orario*:

* [se è specificato un nome] personale.*nome* = nome
* [se è specificato un ruolo] personale.*ruolo* = ruolo

**5** **assegnaCuoco(cuoco: Cuoco, turno: TurnoDiServizio, servizio: Servizio)**

**Pre-condizioni:**

* è in corso la definizione di un Evento *ev* e *ev* **include** servizio

**Post-condizioni:**

* cuoco è **specificato in** turno
* servizio **possiede** turno

**5a.1** **modificaCuoco(cuoco: Cuoco, turno: TurnoDiServizio, servizio: Servizio)**

**Pre-condizioni:**

* è in corso la definizione di un Evento *ev* e *ev* **include** servizio

**Post-condizioni:** Se servizio*.fasciaOraria* ==turno*.orario*:

* cuoco è **specificato in** turno
* servizio **possiede** turno

**6** **approvaMenu(menu: Menu)**

**Pre-condizioni:**

* è in corso la definizione di un Evento *ev*

**Post-condizioni:**

* *ev*.*stato* = inCorso
* menu.*approvato* = sì

**7** **segnaNoteTerminali(note: testo)**

**Pre-condizioni:**

* è in corso la definizione di un Evento *ev*

**Post-condizioni:**

* *ev*.*stato* = terminato
* *ev.note* = note