

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 Servizio *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 Servizio *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 Servizio *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 Servizio *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

- $ev.luogo = \underline{luogo}$
- $ev.numServizi = \underline{numServizi}$
- $ev.tipoServizio = \underline{tipoServizio}$
- [se è specificato un noteTipologia] $ev.noteTipologia = \underline{noteTipologia}$

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] $ev.durata = \underline{durata}$
- [se è specificato un luogo] $ev.luogo = \underline{luogo}$
- [se è specificato un numServizi] $ev.numServizi = \underline{numServizi}$
- [se è specificato un tipoServizio] $ev.tipoServizio = \underline{tipoServizio}$
- [se è specificato un noteTipologia] $ev.noteTipologia = \underline{noteTipologia}$

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
 - $ser.fasciaOraria = \underline{fasciaOraria}$
 - $ser.sede = \underline{sede}$
 - $ser.stato = \text{creato}$
 - 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
 - $ser.fasciaOraria = \underline{fasciaOraria}$
 - $ser.sede = \underline{sede}$
 - $ser.stato = \text{creato}$
 - 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 Servizio $turSer$ 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 Servizio *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