1. Contratti UC Gestione Eventi

Scenario principale

Pre-condizioni generali:

-L’utente è un Organizzatore or

Passo 1

* creazioneEvento(noteIniziali: Testo, cl: Cliente, dataInizio: Testo, dataFine: Testo, periodo?: Testo, numOccorrenze?: Numero, dataFineRic?: Testo)

Pre-condizioni: Nessuna

Post-condizioni:

* È stato creato un nuovo Evento Capofila ev
* ev.NoteIniziali = noteIniziali
* ev.Stato = “Creazione”
* ev.DataInizio = dataInizio
* ev.DataFine = dataFine
* or organizza ev
* cl comissiona ev
* [Se è specificato periodo]
  + Ev.periodo = periodo
  + [se è specificato numOccorrenze]
    - Ev.Numero occorrenze = numOccorrenze
    - Sono state create numOccorrenze Occorrenze oc, per ciascuna delle quali vale
      * ev è Composto da oc
      * oc.Note Iniziali = ev.Note Iniziali
      * oc.Stato = “Creazione”
      * Se oc è l’i-esima occorrenza, oc.DataInizio = ev.DataInizio + (periodo \* I) (\*)
      * Se oc è l’i-esima occorrenza, oc.DataFine = ev.DataFine + (periodo \* I)
      * cl comissiona ev
      * or organizza ev
* [Se è specificato dataFineRic]
  + Ev.DataFineRicorrenza = dataFineRic
  + Per ogni data della forma dataInizio + (periodo \* k nell’intervallo [dataInizio, dataFine], è stata creata un’Occorrenza oc, tale che
    - ev è Composto da oc
    - oc.Note Iniziali = ev.Note Iniziali
    - oc.Stato = “Creazione”
    - oc.DataInizio = ev.DataInizio + (periodo \* k)
    - oc.DataFine = ev. DataFine + (periodo \* k)
    - cl comissiona ev
    - or organizza ev

modificaEvento(noteIniziali: Testo, cl: Cliente, dataInizio: Testo, dataFine: Testo, periodo?: Testo, numOccorrenze?: Numero, dataFineRic?: Testo)

annullaEvento(ev: Evento, propMod?: Sì/No)

Pre-condizioni:

* or Organizza ev

Post-condizioni:

* ev.stato = “Annullato”
* Per ogni servizio srv tale che ev prevede srv
  + Per ogni disponibilità disp in srv
    - disp.prenotato = “No”
    - disp.ruolo = “ “
    - l’associazione tra disp ed srv è stata eliminata
* [Se propMod = Sì ed ev è un Evento Capofila]
  + Per ogni Occorrenza oc tale che e composto da oc
    - oc.stato = “Annulato”
    - Per ogni servizio srv’ tale che oc prevede srv
      * Per ogni disponibilità disp’ in srv’
      * disp’.prenotato = “No”
      * disp’.ruolo = “ “
      * l’associazione tra disp’ ed srv’ è stata eliminata

chiusuraEvento(ev: Evento, noteFin?: Testo, propMod?: Sì/No)

Pre-condizioni:

* or Organizza ev

Post-condizioni:

* Ev.stato = “Terminato”
* [Se è specificato noteFin] ev.note finali = noteFin
* [Se propMod è specificato e vale Sì]
  + [Se ev è un Evento Capofila]
    - Per ogni Occorrenza oc da cui è composto ev
      * oc.stato = “Terminato”
      * oc.note finali = noteFin

caricamentoEvento(ev: Evento)

Pre-condizioni:

* or organizza ev

Post-condizioni: nessuna

Passo 2

aggiuntaServizio(tipo: Testo, data: Testo, luogo: Testo, numPersone: Testo, menu: Menù. Propaga: Sì/No)

Pre-condizioni:

* È in corso la modifica di un evento e
* e.Stato = “Creazione” (non elenchiamo proprio gli eventi in corso oppure completati tra quelli ai queli si possa aggiungere un servizio)

Post-condizioni:

* È stato creato un nuovo Servizio srv
* Srv.Tipo = tipo
* Srv.Data = data
* Srv.Luogo = luogo
* Srv.Numero di persone = numPersone
* menu In uso in srv
* e Prevede srv
* Srv.approvato = No
* [Se e.periodo è specificato e Propaga = Sì]
  + Per ogni Occorrenza oc tale che e composto da oc
    - È stato creato un nuovo Servizio srv’
    - srv’.Tipo = tipo
    - Se oc è l’i-esima occorrenza
      * srv’.Data = data + (e.periodo \* i)
    - srv’.Luogo = luogo
    - srv’.NumeroPersone = numPersone
    - menu In uso in srv’
    - oc Prevede srv’
    - srv’.approvato = No

modificaServizio(srv: Servizio, tipo?: Testo, data?: Testo, luogo?: Testo, numPartecipanti?: Testo, evento?: Evento, propagaModifica: Sì/No)

Pre-condizioni:

* Ev prevede srv
* Or Organizza ev
* ev.Stato = “Creazione”

Post-condizioni:

* [Se è specificato tipo] srv.Tipo = tipo
* [Se è specificato data] srv.Data = data
* [Se è specificato luogo] srv.Luogo = luogo
* [Se è specificato numPartecipanti] srv.Numero di persone = numPartecipanti
* [Se è specificato evento] …
* [Se propagaModifica vale Sì e ev è un Evento Capofila e ev.numeroOccorrenze > 1]

Per ogni Occorrenza oc tale che ev composto da oc

eliminaServizio(srv: Servizio, propMod?: Sì/No)

Pre-condizioni:

* Ev prevede srv
* Or organizza ev
* Menu in uso in srv
* Fr contiene compiti di srv
* Pm menù di srv

Post-condizioni:

* Menu non è più associato ad srv
* Fr non è più associato ad srv
* Pm non è più associato ad srv

recuperoElencoServizi(ev: Evento, propMod?: Sì/No)

(Così può vedere solo I servizi dei suoi eventi, ha senso?)

Pre-condizioni:

* Ev prevede srv
* Or organizza ev

Post-condizioni: Nessuna (è un’interrogazione)

Passo 3

assegnazioneChef(ev: Evento, chef: Chef)

Pre-condizioni:

* ev.stato = “Creazione”
* Non esiste un Evento ev’, tale che
  + Ev' assegnato a chef
  + Ev'.dataInizio >= ev.dataInizio ed ev’.dataFine <= ev.dataFine

Post-condizioni:

* ev assegnato a chef

Passo 4

prenotazionePersonale(disp: Disponibilità, srv: Servizio)

Pre-condizioni:

* disp.Prenotato = No
* è in corso la modifica di un evento e
* e prevede srv

Post-condizioni:

* disp.Prenotato = Sì
* Disp in srv
* or prenotazione disp

cancellazionePrenotazione(disp: Disponibilità)

Pre-condizioni

* è in corso la modifica di un evento e
* e prevede srv
* disp in srv
* disp.prenotato = Sì

Post-condizioni:

* disp.prenotato = No
* disp.ruolo = “”
* disp non è più associata a srv

Passo 5

approvazioneMenu(srv: Servizio)

Pre-condizioni:

* Ev prevede srv

Post-condizioni:

* srv.approvato = Sì
* ev.stato = “In corso”

propostaModifica(srv: Servizio, ric: Ricetta, cm: Commento)

Pre-condizioni:

* è in corso la modifica di un evento ev
* ev prevede srv
* ev.stato == “Creazione”

Post-condizioni:

* È stata create una Proposta Modifica pm
* Pm.commento = cm
* Pm.accettata = No
* Pm menù di srv
* Pm riferita a ric

Passo 6

assegnazioneRuolo(disp: Disponibilità, ruolo: Testo)

Pre-condizioni:

* Or prenotazione disp
* Se mp inserimento disp, allora mp è un Membro del personale di servizio

Post-condizioni:

* Disp.Ruolo = ruolo

Cose da fare

* Fare in modo che al passo 1 venga fissato l’evento su cui si lavora
* controllare che l’evento su cui si lavora sia settato: altro (es. Or è l’organizzatore non serve, è già stato verificato al passo 1)
* l’associazione tra organizzatore e prenotazione è ridondante, ma conviene metterla per semplificare e velocizzare la verifica del passo 6
* Anche l’associazione tra prenotazione e membro del personale riservato è ridondante, ma semplifica la verifica delle post-condizioni

Note: (NON CANCELLARE!!!)

* Per definire la semantica delle creazione dell'evento ricorrente, si è assunto che il periodo sia in giorni. È impreciso, perchè potrebbe non esserlo, ma rende l'idea di cosa dovrebbe succedere.