**Contratti per lo UC “Gestire i turni”**

Pre-condizione generale: l’attore è identificato con un’ istanza *org* di Organizzatore

1. mostraCalendario()

Interrogazione al sistema: non ha post-condizioni

Nemmeno precondizioni, potrebbe essere fatta in qualsiasi momento

1a.1 visualizzazioneFiltrata(schemaTurni: CalendarioTurni, filtraData?:Data, filtraOrario?:Ora, filtraCuoco?:Cuoco, filtraEvento?:Evento)

Da fare

Precondizioni:

* E’ in corso la compilazione del Calendario dei Turni *ct*
* schemaTurni *st* che appartiene a *ct*

Post-condizioni:

* [se è specificato filtraData] shemaTurni.data == filtraData
* [se è specificato filtraEvento] shemaTurni.evento == filtraEvento
* [se è specificato filtraOrario] shemaTurni.orario == filtraOrario
* [se è specificato filtraCuoco] shemaTurni.cuoco == filtraCuoco
* [se è specificato filtraData] shemaTurni.data == filtraData

1b.1 apriSchema(schemaTurni: CalendarioTurni)

Precondizioni:

* E’ in corso la compilazione del Calendario dei Turni *ct*
* schemaTurni *st* che appartiene a *ct*

Post-condizioni:

* Viene mostrato il *ct* in cui è applicato *st. ?????????*

2. aggiungiTurnoCucina( cuoco?: Cuoco, data?:Data, orarioInizio?:Ora, orarioFine?:Ora, evento?: Evento, tabelloneDeiTurni: TabelloneTurni)

Precondizioni:

* E’ in corso la compilazione del Calendario dei Turni *ct*

Post-condizioni:

* **è stata creata** una istanza di Turno *t*
* [se è specificato cuoco] *t.cuoco* **=** cuoco
* [se è specificato evento] *t.evento* **=** evento
* [se è specificato oraInizio] *t.oraInizio* **=** orInizio
* [se è specificato oraFine] *t.oraFine* **=** oraFine
* [se è specificato data] *t.data* **=** data
* Turno *t*  **appartiene a** *ct*
* Turno *t*  **è parte del** tabelloneDeiTurni *tdt*

2a.1 eliminazioneTurno(turno:Turno)

Precondizioni:

* E’ in corso la compilazione del Calendario dei Turni *ct*
* turno *t* **appartiene a** *ct*

Post-condizioni:

* È stata elilminata la relazione appartiene tra *t*  e *ct*
* È stata elilminata la relazione appartiene tra *t*  e cuoco
* È stata elilminata la relazione appartiene tra *t*  e *tabellone dei turni*

2b.1 applicoTurnoRicorrente(turno:Turno, insieme G di giorni)

Precondizioni:

* E’ in corso la compilazione del Calendario dei Turni *ct*
* turno **appartiene a** *ct*
* G **appartiene a** *ct*

Post-condizioni:

* Se ogni l’elemento di G non ha un turno, ad ogni elemento appartiene *t*

3. raggruppaTurni(insieme T di Turni, cuoco: Cuoco)

Precondizioni:

* è in corso **la compilazione** Calendario dei turni *ct*
* **è richiesta** una lista di turni T *lt* in cui ogni turno ha la stessa data
* *lt* **appartiene a**  Calendario dei Turni *ct*
* cuoco *c*  appartiene a *lt*

Post-condizioni:

* E’ stata aggiunta una istanza di Turni *t* **che appartiene** a *ct*

4. modificaTurno(orarioInizio?: Ora, orarioFine?: Ora, turno:Turno, data?:Data, cuoco?: Cuoco)

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* *t*  appartiene a *ct*

Post-condizioni:

* [se è specificato orarioInizio] *t.orarioInizio =*  orarioInizio
* [se è specificato orarioFine] *t.orarioFine =*  orarioFine
* [se è specificato cuoco] *t.cuoco =* cuoco
* [se è specificato data] *t.data =* data
* Turno tè stato modificato

4a.1 modificaTurni(orarioInizio?: Ora, orarioFine?: Ora, insieme T di Turni, data?:Data, cuoco?: Cuoco)

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* *T*  appartiene a *ct*

Post-condizioni:

* [se è specificato orarioInizio] *T.orarioInizio =*  orarioInizio
* [se è specificato orarioFine] *T.orarioFine =*  orarioFine
* [se è specificato cuoco] *T.cuoco =* cuoco
* [se è specificato data] *T.data =* data
* *T* è stato modificato

4b. eliminaTurni(insieme T di Turni)

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* T **appartiene a** *ct*

Post-condizioni:

* [se un cuoco *c* appar**tiene a** T] viene eliminata la relazione appartiene tra *c* e T
* Viene eliminata la relazione **appartiene**  tra T e *ct*
* Viene eliminata la relazione **è parte del** tra T e tabellone dei turni

5. aggiungiScadenza( insieme T di Turni, dataScadenza: Data)

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* T **appartiene a** *ct*

Post-condizioni:

* *T.data Scadenza* = data Scadenza

5a.1 modificaScadenza( insieme T di Turni, dataScadenza: Data)

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* T **appartiene a** *ct*

Post-condizioni:

* *T.data Scadenza* = data Scadenza

7. aggiungoTurnoServizio(servizio: Servizio, oraInizio?: Orario, oraFine?: Orario, addettoServizio?: testo, luogo?: testo, data?: Data, tabelloneDeiTurni: TabelloneTurni)

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* sevizio *s* appartiene a Evento *e,* deve esistere un *e* tale per cui *s* si riferisce a *e*

Post-condizioni:

* **è stata creata** una istanza di turno di servizio *ts*
* [se è specificato orarioInizio] *ts.orarioInizio =*  ora Inizio
* [se è specificato orarioFine] *ts.orarioFine =*  oraFine
* [se è specificato data] *ts.data =* data
* [se è specificato addettoServizio] *ts.addettoServizio =* addettoServizio
* [se è specificato luogo] *ts.luogo =* luogo
* *ts* **appartiene a** servizio *s*
* *ts* **è parte del** tabellone dei turni *tdt*

7a.1 modificaTurnoServizio(turnoServizio: TurnoServizio, )

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* sevizio *s* appartiene a Evento *e,* deve esistere un *e* tale per cui *s* si riferisce a *e*

Post-condizioni:

* [se è specificato orarioInizio] *ts.orarioInizio =*  ora Inizio
* [se è specificato orarioFine] *ts.orarioFine =*  oraFine
* [se è specificato data] *ts.data =* data
* [se è specificato addettoServizio] *ts.addettoServizio =* addettoServizio
* [se è specificato luogo] *ts.luogo =* luogo

7b.1 eliminaTurnoServizio(turnoServizio: TurnoServizio, servizio: Servizio, tabelloneDeiTurni: TabelloneTurni )

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* turnoServizio *ts* **appartiene a** sevizio *s, s* **appartiene a** Evento *e,* deve esistere un *e* tale per cui *s* si riferisce a *e*
* turnoServizio **è parte del** tabelloneDeiTurni *tdt*

Post-condizioni:

* viene eliminata la relazione **appartiene a** tra *ts*  e addetto al servizio
* viene eliminata la relazione **appartiene a** tra *ts*  e s
* viene eliminata la relazione **è parte del** tra *ts*  e *tdt*
* viene eliminata la relazione **appartiene** tra *ts* e *ct*
* [se è specificato luogo] *ts.luogo =* luogo

7c.1 impostaTempoPrePostTurno(turnoServizio: TurnoServizio, tempoPre: numero, tempoPost: numero)

Precondizioni:

* è in corso la compilazione del Calendario dei Turni *ct*
* turnoServizio *ts* **appartiene** a Servizio *s*

Post-condizioni:

* [se è specificato tempoPre] *ts.tempoPre =*  tempoPre
* [se è specificato tempoPost] *ts.tempoPost =*  tempoPost

8. salvaSchemaTurni()

Precondizioni:

* È richiesto uno schema dei turni

Post-condizioni:

* Lo schema è stato salvato e pubblicato