

# Gestire dei turni di cucina

## Informazioni generali

**Nome caso d'uso:** Gestire i turni di cucina

**Portata:** Sistema

**Livello:** Obiettivo utente

**Attore primario:** Organizzatore

**Parti Interessate:** Cuochi e Personale di servizio

**Pre-condizioni:** L'attore deve essere autenticato come Organizzatore

**Garanzie di successo o post-condizioni:** I turni di cucina sono assegnati (è stato indicato chi, quando e cosa fa) e risultano consultabili.

#	Attore	Sistema
1	Aprire il <b>Tabellone</b>	Fornisce il Tabellone e ne Permette sia la Consultazione che la Modifica
2	<b>Opzionalmente</b> Creare un nuovo <b>Orario di Preparazione</b> (vuoto o partendo da uno <b>Schema</b> )	Registra un nuovo Orario di Preparazione associato allo schema
3	<b>Opzionalmente</b> Aggiungere <b>Turno</b> (specificando <i>data, orlnizio, durata, ricorrenza, detttagli</i> ) <i>detttagli</i> contiene <i>luogo</i> [ Se è un turno di Preparazione <i>detttagli</i> contiene <i>evento, stimaTempo,scadenza</i> ]	Registra un nuovo Turno con tutte le informazioni necessarie
	<i>Ripetere 3 finché non si è soddisfatti</i>	
4	<b>Opzionalmente</b> Unisci più turni in un <b>Gruppo di Turni</b>	Registra un nuovo Gruppo di Turni
		Rimuovi i Turni singoli che ora fanno parte di Gruppo di Turni
5	<b>Opzionalmente</b> associa <b>Personale</b>	Registra un elemento del personale nel Turno

		(Specificatamente sarà un <b>Cuoco</b> nei Turni di Preparazione o potrebbe essere un eventuale Cameriere nei Turni di Servizio)
	<i>Ripetere 3 finché non si è soddisfatti</i>	
6	<b>Opzionalmente</b> Salvare lo <b>Schema</b>	Registra lo Schema per utilizzi futuri
	<i>Ripetere 2 finché non si è soddisfatti</i>	
	<i>Termina il caso d'uso</i>	

## Scenario principale di successo

### Eccezione 1.1a

#	Attore	Sistema
1.1a.1	Aprire il <b>Tabellone</b>	L'attore che tenta di modificare il Tabellone non è un Organizzatore

### Estensione 2a

#	Attore	Sistema
2a.1	Aprire un <b>Orario</b> Esistente	Fornisce un Orario e ne Permette la Modifica

### Eccezione 2a.1a

#	Attore	Sistema
2a.1a.1	Aprire un <b>Orario</b> Esistente	L'orario non è presente

### Eccezione 2a.1b

#	Attore	Sistema
---	--------	---------

<b>2a.1b.1</b>	Aprire un <b>Orario</b> Esistente	L'orario non è "in corso" o "in programma"
----------------	-----------------------------------	--

## Estensione 2b

#	Attore	Sistema
<b>2b.1</b>	Elimina un <b>Orario</b> Esistente	Cancella un Orario che non sarà più consultabile
	<i>Riprendere dal passo 2</i>	

## Eccezione 2b.1a

#	Attore	Sistema
<b>2b.1a.1</b>	Elimina un <b>Orario</b> Esistente	L'orario non è presente

## Eccezione 2b.1b

#	Attore	Sistema
<b>2b.1b.1</b>	Elimina un <b>Orario</b> Esistente	L'orario è "in corso"

## Estensione 2c

#	Attore	Sistema
<b>2c.1</b>	Creare un nuovo <b>Orario di Servizio</b> vuoto	Crea un nuovo <b>Schema</b>
		Registra un nuovo Orario di Servizio associato allo schema
<b>2c.2</b>	<b>Opzionalmente</b> Aggiungere <b>Turno</b> (specificando <i>data, orainizio, durata, dettagli</i> ) <i>dettagli</i> contiene <i>evento, stimaTempo, scadenza</i>	Registra un nuovo Turno con tutte le informazioni necessarie

	<i>Ripetere 2c.2 finché non si è soddisfatti</i>	
	<i>Riprendere dal passo 4</i>	

### **Eccezione 3.1a** ( Eccezione 2c.2.1a )

#	Attore	Sistema
<b>3.1a.1</b>	Aggiungere <b>Turno</b>	la posizione nello schema non risulta "libera" nello schema nel momento indicato

### **Estensione 3a** (Estensione 2c.2a)

#	Attore	Sistema
<b>3a.1</b>	Modificare <b>Turno</b>	Permette la modifica di Tutti i Turni e Gruppo di Turni.

### **Eccezione 3a.1a** ( Eccezione 2c.2a.1a )

#	Attore	Sistema
<b>3a.1a.1</b>	Modificare <b>Turno</b>	il Turno non è presente

### **Eccezione 3a.1b** ( Eccezione 2c.2a.1b )

#	Attore	Sistema
<b>3a.1b.1</b>	Modificare <b>Turno</b>	il Turno non "in programma"

### **Eccezione 3a.1c** ( Eccezione 2c.2a.1c )

#	Attore	Sistema
<b>3a.1c.1</b>	Modificare <b>Turno</b>	la posizione nello schema non risulta

		"libera" nello schema nel momento indicato
--	--	--

### Estensione 3b (Estensione 2c.2b)

#	Attore	Sistema
3b.1	Elimina Turno	Elimina di Tutti i Turni o Gruppo di Turni
	<i>Riprendere dal passo 3</i>	

### Eccezione 3b.1a ( Eccezione 2c.2b.1a )

#	Attore	Sistema
3b.1a.1	Elimina Turno	il Turno non è presente

### Eccezione 3b.1b ( Eccezione 2c.2b.1b )

#	Attore	Sistema
3b.1b.1	Elimina Turno	il Turno è "passato" o "in corso"

### Eccezione 4.1a

#	Attore	Sistema
4.1a.1	Unisci più turni in un Gruppo di Turni	uno o più dei turni non esiste

### Estensione 4a

#	Attore	Sistema
4a.1	Separa un Gruppo di Turni	Registra i sottoturni del Gruppo di Turni come Turni Singoli

		Gruppo di Turni non è più registrato
--	--	--------------------------------------

### Eccezione 4a.1a

#	Attore	Sistema
4a.1a.1	Separa un Gruppo di Turni	il Gruppo di Turni non esiste

### Eccezione 4a.1b

#	Attore	Sistema
4a.1b.1	Separa un Gruppo di Turni	il Turno non è un Gruppo di Turni

### Eccezione 5.1a

#	Attore	Sistema
5.1a.1	Associa Personale	il turno non esiste

### Eccezione 5.1b

#	Attore	Sistema
5.1b.1	Associa Personale	il Personale è già associato a questo turno

### Estensione 5a

#	Attore	Sistema
5a.1	Rimuovi Personale	Un elemento del personale non è più registra nel Turno

## Eccezione 5a.1a

#	Attore	Sistema
5a.1a.1	Rimuovi <b>Personale</b>	il turno non esiste

## Eccezione 5a.1b

#	Attore	Sistema
5a.1b.1	Rimuovi <b>Personale</b>	il Personale non è associato a questo turno

## Eccezione 6.1a

#	Attore	Sistema
6.1a.1	Salvare lo <b>Schema</b>	il titolo è già presente nell' <b>Archivio</b>

# Contratti per lo UC “Gestire i Turini”

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

## 1. *apriTabellone():Tabellone*

**Pre-condizioni:** —

**Post-condizioni:**

- è stato fornito il Tabellone *tab* che ora è modificabile

## 2. *aggiungiOrario(tipo:String, schema?: Schema, titolo?: String, dataInizio: Date, dettagli: Map<String, Object>): Orario*

**Pre-condizioni:**

- è in corso la modifica del Tabellone *tab*
- *tipo* deve essere un valore accettabile (essere uguale a “Servizio” o “Preparazione”)
- [se è specificato *schema*]
  - *schema* è uno Schema presente nell'ArchivioSchemi *arc* e *schema.tipo* è uguale a *tipo*
  - il giorno della settimana di *dataInizio* corrisponde con *schema.primoGiorno*

**Post-condizioni:**

- è stata creata un'istanza *or* di Orario **contenuto in** *tab*
- è stato impostato l'orario come aperto assegnando: *tab.orarioAperto* = *or.titolo*
- [se non è specificato *schema*]
  - [se *tipo* è uguale a “Preparazione”]
    - è stata creata un'istanza *schema* di SchemaPreparazione **contenuto in** *or*
    - è stata creata una mapa *turni* di TurnoPreparazione **contentuta in** *schema*
    - [se è specificato *durataTurno*] *schema.durataTurno* = *durataTurno*
    - [altrimenti] *schema.durataTurno* = 2
    - *schema.titolo* = *titolo*
  - [se *tipo* è uguale a “Servizio”]
    - è stata creata un'istanza *schema* di SchemaServizio **contenuto in** *or* e **riferito a** Evento *dettagli.evento*
    - *schema.evento* = *dettagli.evento*
    - è stata creata una mapa *turni* di TurnoServizio **contentuta in** *schema*
    - [se è specificato *durataTurno*] *schema.durataTurno* = *durataTurno*
    - [altrimenti] *schema.durataTurno* = 2
    - *schema.titolo* = *titolo*
- [se *titolo* è specificato]
  - *or.titolo* = *titolo*



- *schema.titolo = titolo*
- [altrimenti] *or.titolo = schema.titolo*
- [altrimenti]
  - *or.schema = schema*
  - *or.titolo = titolo*
  - *schema.titolo = titolo*
- *or.dataInizio = dataInizio*

## 2a. modificaOrario(titolo?:String): Orario

### Pre-condizioni:

- è in corso la modifica del Tabellone *tab*
- *titolo* è uguale a *or.titolo* di un Orario *or* **appartenente** a *tab*

### Post-condizioni:

- Orario *or* **appartenente** a *tab* ora risulta come orario aperto è quindi è modificabile
- *tab.orarioAperto = titolo*

## 2b. rimuoviOrario(): Boolean

### Pre-condizioni:

- è in corso la modifica del Tabellone *tab*
- *tab.orarioAperto* è definito

### Post-condizioni:

- è stato eliminato l'Orario *or* *avente* *or.titolo = tab.orarioAperto*, che **non è più appartenente** a *tab*
- nessun orario risulta come orario aperto e quindi modificabile in *tab*
  - *tab.orarioAperto = NULL*

## 3. aggiungiTurno (data: Data, orInizio: Time, ricorrenza?: Ricorrenza, durata?: int, dettagli: Map<String, Object>): Turno

### Pre-condizioni:

- è in corso la modifica dell'Orario *or*
- *data* è successore di *dataAttuale* e di *or.dataInizio*
- la coppia *<turno.giorno, orInizio>* per *schema* risulta "libero"
- [*ricorrenza* è definito]
  - [*turno* è un TurnoPreparazione]
  - ogni coppia *<turno.giorno, turno.orInizio>* di *ricorrenza* per *schema* risulta "libero"

### Post-condizioni:

- [se *or.schema* è di tipo "Preparazione"] è stata creata un'istanza *turno* di TurnoPreparazione **contenuta nello Schema *schema* contenuto in *or***

- *turno.oralnizio* = *oralnizio*
- [se *durata* è definita] *turno.durata* = *durata*
- [altrimenti] *turno.durata* = *schema.durata*
- *turno.giorno* = [ *or.dataInizio* - *data* ]
- *this.luogo* = *dettagli*["Luogo"]
- [*ricorrenza* è definito] *turno.ricorrenza* = *ricorrenza*
- [se *or.schema* è di tipo "Servizio"] è stata creata un'istanza *turno* di *TurnoServizio* **contenuta nello Schema *schema* contenuto in *or***
  - *turno.oralnizio* = *oralnizio*
  - [se *durata* è definita] *turno.durata* = *durata*
  - [altrimenti] *turno.durata* = *schema.durata*
  - *turno.giorno* = [ *or.dataInizio* - *data* ]
  - *this.stimaTempo* = *dettagli*["StimaTempo"]
  - *this.scadenza* = *dettagli*["Scadenza"]

### 3a. modificaTurno (id: String, data?: Data, oralnizio?: Time, durata?: int, ricorrenza?: Ricorrenza, dettagli?: Map<String, Object>): Turno

#### Pre-condizioni:

- è in corso la modifica dell'Orario *or*
- *id* fa riferimento ad un Turno *turno* **contenuto nello Schema *schema* contenuto in *or***
- [*ricorrenza* è definito]
  - [*turno* è un TurnoPreparazione]
  - ogni coppia <*turno.giorno*,*turno.oralnizio*> di *ricorrenza* per *schema* risulta "libero"

#### Post-condizioni:

- [se *turno* è un GruppoTurni]
  - [per ogni istanza TurnoSingolo *sottoturno* **raggruppata da *turno***]
    - [*data* è definito] *turno.giorno* = [ *or.dataInizio* - *data* ]
    - [*oralnizio* è definito] *turno.oralnizio* = *oralnizio*
    - [*ricorrenza* è definito] *turno.ricorrenza* = *ricorrenza*
    - [se *dettagli* è definito] [per ogni coppia <*key*, *value*> di *dettagli*]
      - *sottoturno.key* = *value*
- [se *turno* è un TurnoSingolo]
  - [*data* è definito] *turno.giorno* = [ *or.dataInizio* - *data* ]
  - [*oralnizio* è definito] *turno.oralnizio* = *oralnizio*
  - [*ricorrenza* è definito] *turno.ricorrenza* = *ricorrenza*
  - [se *dettagli* è definito] [per ogni coppia <*key*, *value*> di *dettagli*]
    - *turno.key* = *value*

### 3b. rimuoviTurno (id: String): Boolean

#### Pre-condizioni:

- è in corso la modifica dell'Orario *or*

- *id* fa riferimento ad un Turno *turno* **contenuto nello** Schema *schema* **contenuto in** *or*

**Post-condizioni:**

- *turno* non è più **contenuto in** *schema*

#### 4. unisciTurni (turni: List<String>): Turno

**Pre-condizioni:**

- è in corso la modifica dell'Orario *or*
- [per ogni *id* in *turni*] *id* fa riferimento ad un Turno *turno* **contenuto nello** Schema *schema* **contenuto in** *or*

**Post-condizioni:**

- è stata creata una nuova istanza *gruppo* di GruppoTurni **contenuto in** *schema*
- [per ogni *istanza turno* di Turno **contenuto in** *schema* corrispondente ad un *id* in *turni*]
  - [se *turno* è un GruppoTurni]
    - [per ogni istanza TurnoSingolo *sottoturno* **raggruppata da** *turno*]
      - *sottoturno* è ora **raggruppato da** *gruppo*
  - [se *turno* è un TurnoSingolo]
    - *turno* è ora **raggruppato da** *gruppo*
- *turno* non è più **contenuto in** *contenuto in* *schema*

#### 4a. separaTurno (id: String): List<Turno>

**Pre-condizioni:**

- è in corso la modifica dell'Orario *or*
- *id* fa riferimento ad un GruppoTurni *gruppo* **contenuto nello** Schema *schema* **contenuto in** *or*

**Post-condizioni:**

- [per ogni *istanza turno* di Turno **raggruppata da** *gruppo*]
  - ora *turno* è **contenuto in** *schema*
- *gruppo* non è più **contenuto in** *schema*

#### 5. associaPersonale (idTurno:String, personale:Personale): Boolean

**Pre-condizioni:**

- è in corso la modifica dell'Orario *or*
- *idTurno* fa riferimento ad un Turno *turno* **contenuto nello** Schema *schema* **contenuto in** *or*
- [personale deve coincidere con la ripologia richiesta in *schema* ]
  - [se *or.schema* è di tipo "Preparazione"]
    - *personale* è un Cuoco
  - [se *or.schema* è di tipo "Servizio"]
    - *personale* è un Cameriere

**Post-condizioni:**

- *personale* ora è **assegnato a** *turno*

## 5.a rimuoviPersonale (idTurno:String, idPersonale:String): Boolean

### Pre-condizioni:

- è in corso la modifica dell'Orario *or*
- *idTurno* fa riferimento ad un Turno *turno* **contenuto nello** Schema *schema* **contenuto in** *or*
- *idPersonale* fa riferimento ad un Personale *personale* **assegnato a** *turno*

### Post-condizioni:

- *personale* no è più **assegnato a** *turno*

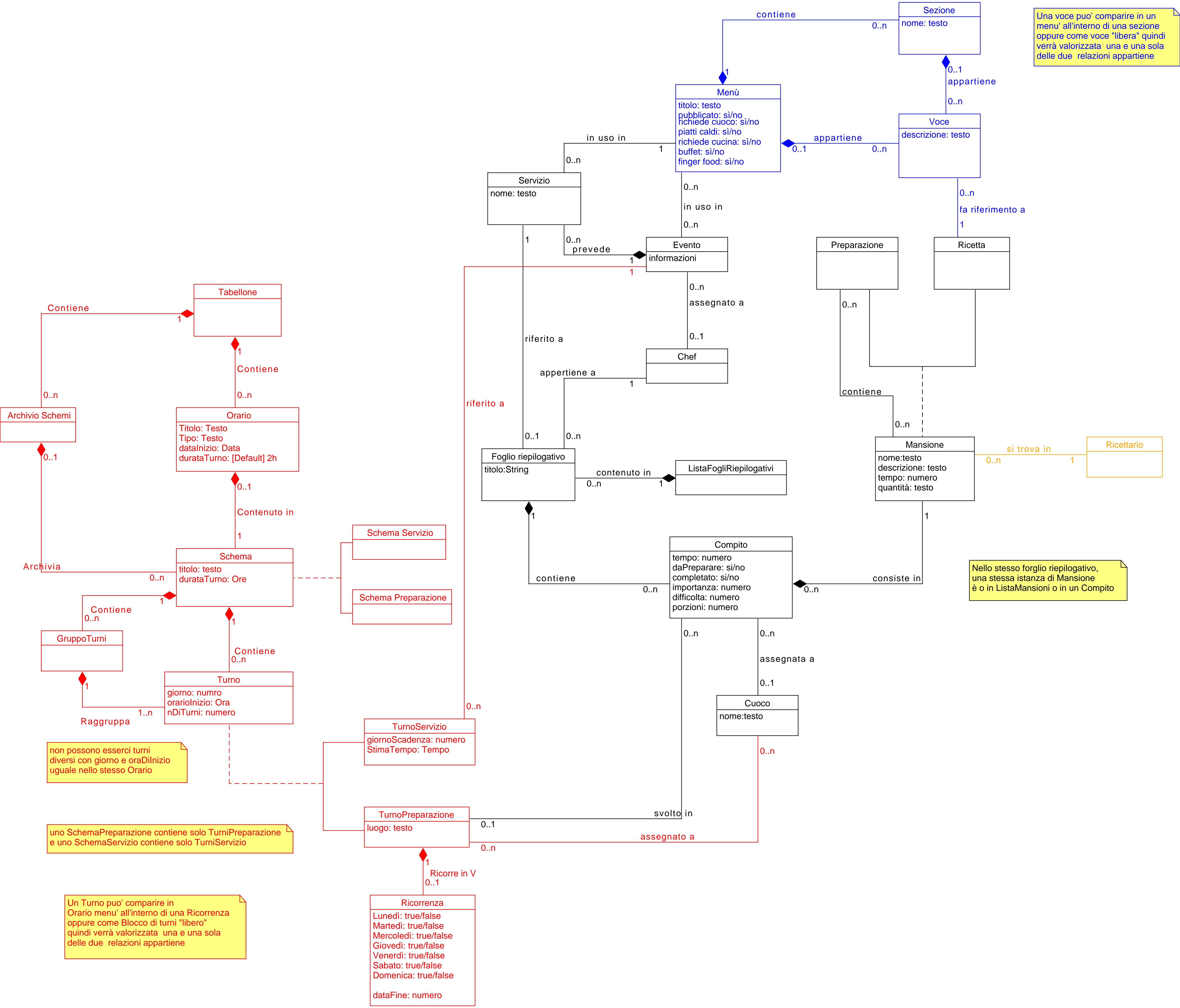
## 6. salvaSchema (titolo?: String): Schema

### Pre-condizioni:

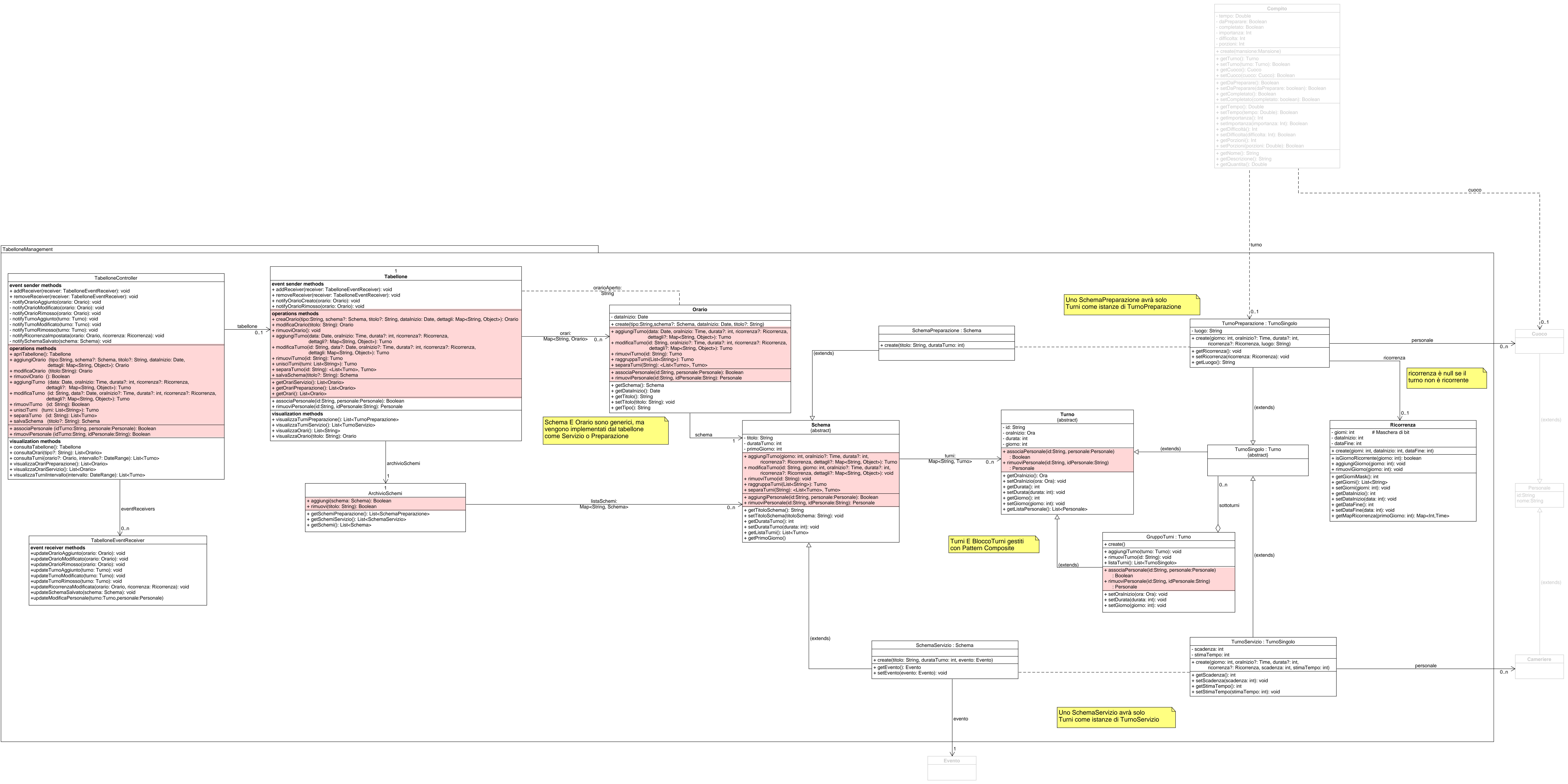
- è in corso la modifica dell'Orario *or*
- non è presente un istanza *schema* di Schema **archiviata in** nell'ArchivioSchemi *archivioSchemi* **contenuto nel** Tabellone *tab* avente *schema.titolo* = *titolo*

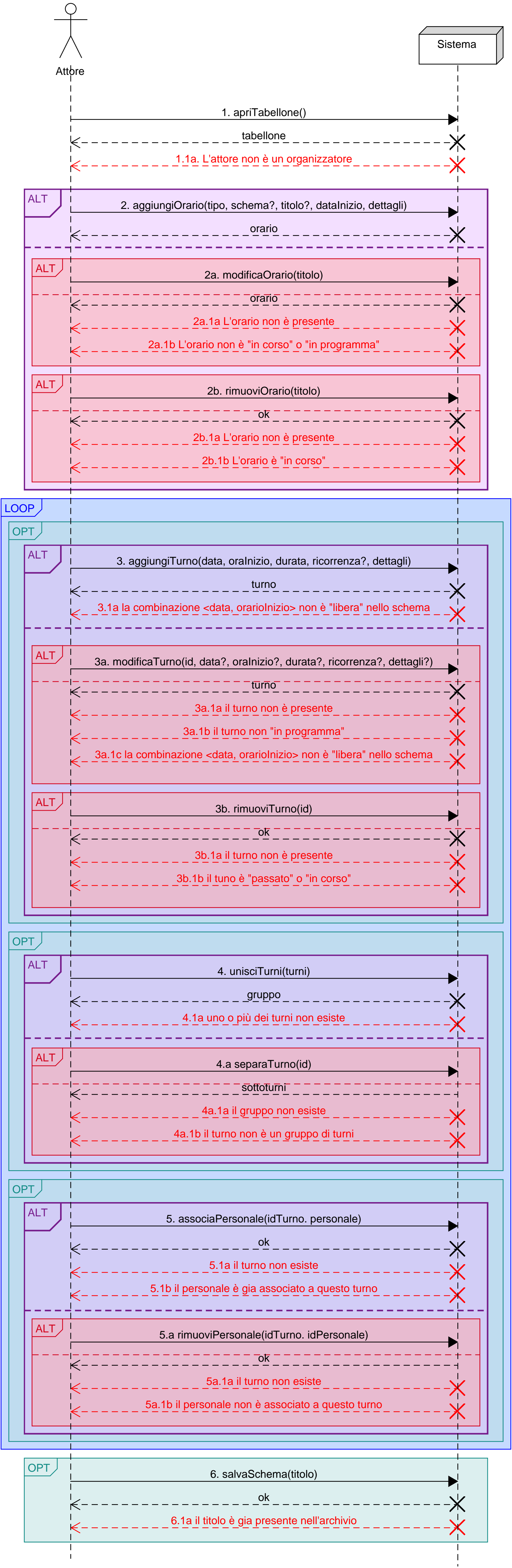
### Post-condizioni:

- è stata creata una copia *nuovoSchema* dell'istanza *schema* di Schema **contenuta in** *or* che ora è **archiviata in** *archivioSchemi*









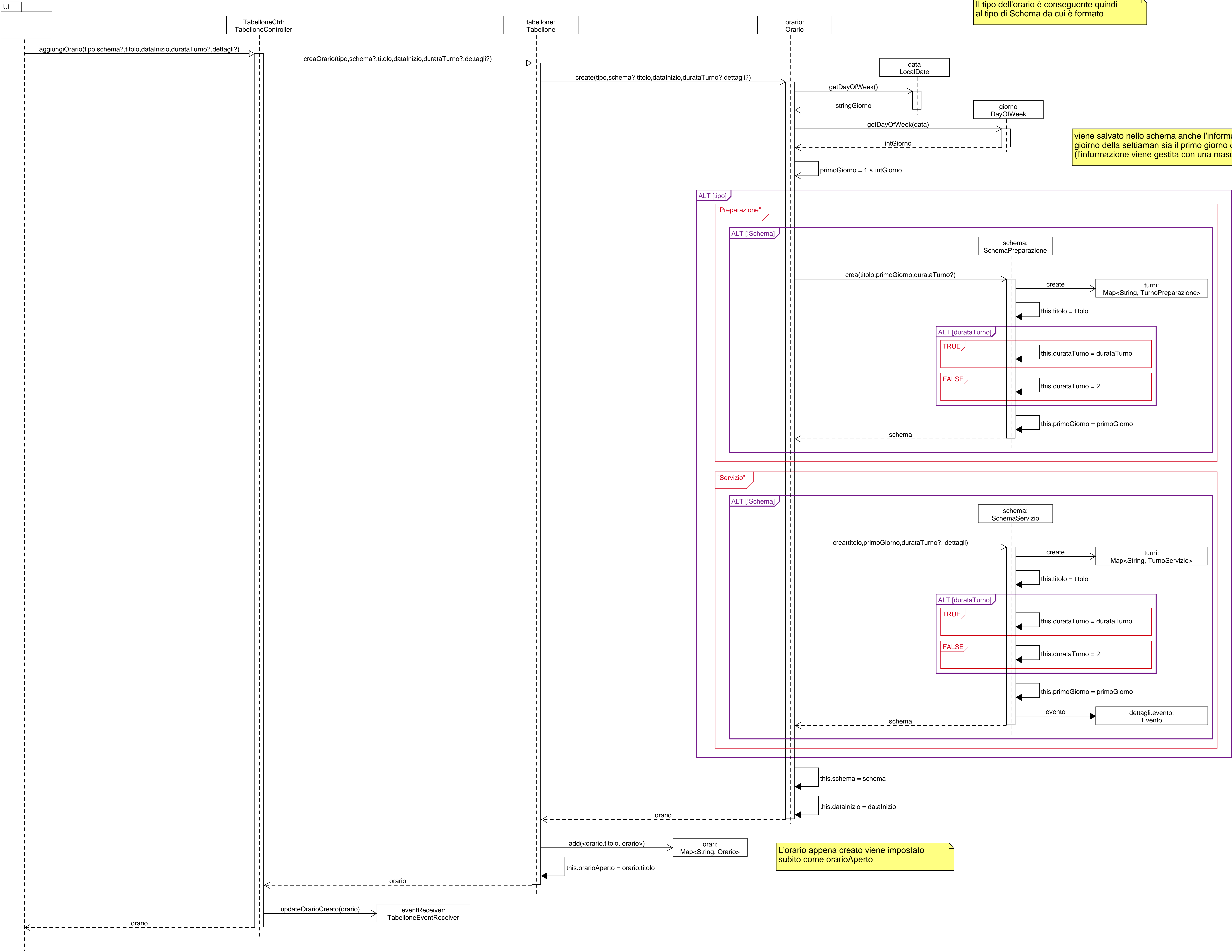


DSD per l'operazione:  
2. aggiungiOrario

Il tipo dell'orario è conseguente quindi  
al tipo di Schema da cui è formato

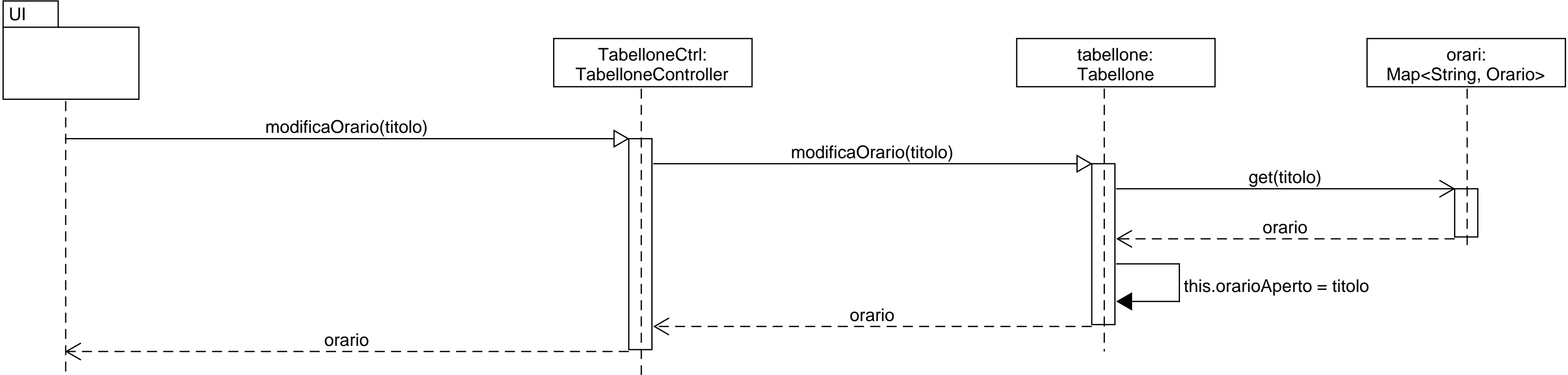
viene salvato nello schema anche l'informazione di che  
giorno della settimana sia il primo giorno dell'orario  
(l'informazione viene gestita con una maschera di bit)

L'orario appena creato viene impostato  
subito come orarioAperto



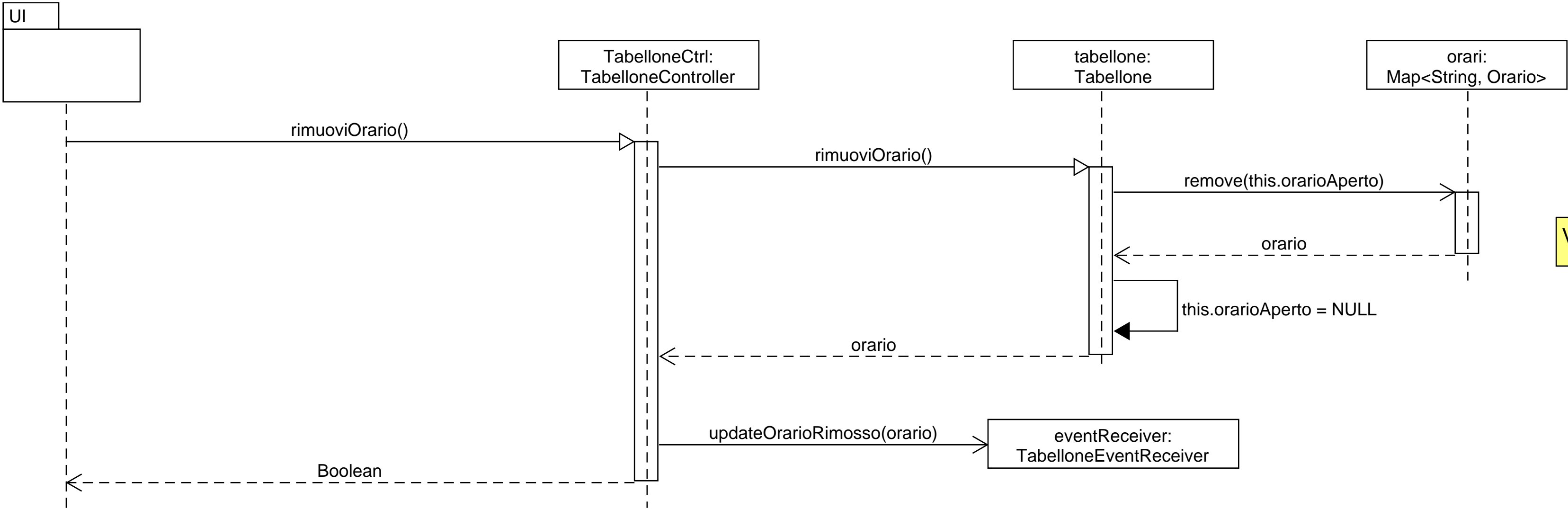


DSD per l'operazione:  
2a. modificaOrario



Per modificare un orario basta impostarlo come aperto,  
e tutte le modifiche successive verranno eseguite su  
di esso fino a quando non ne viene aperto un altro

DSD per l'operazione:  
2b. rimuoviOrario

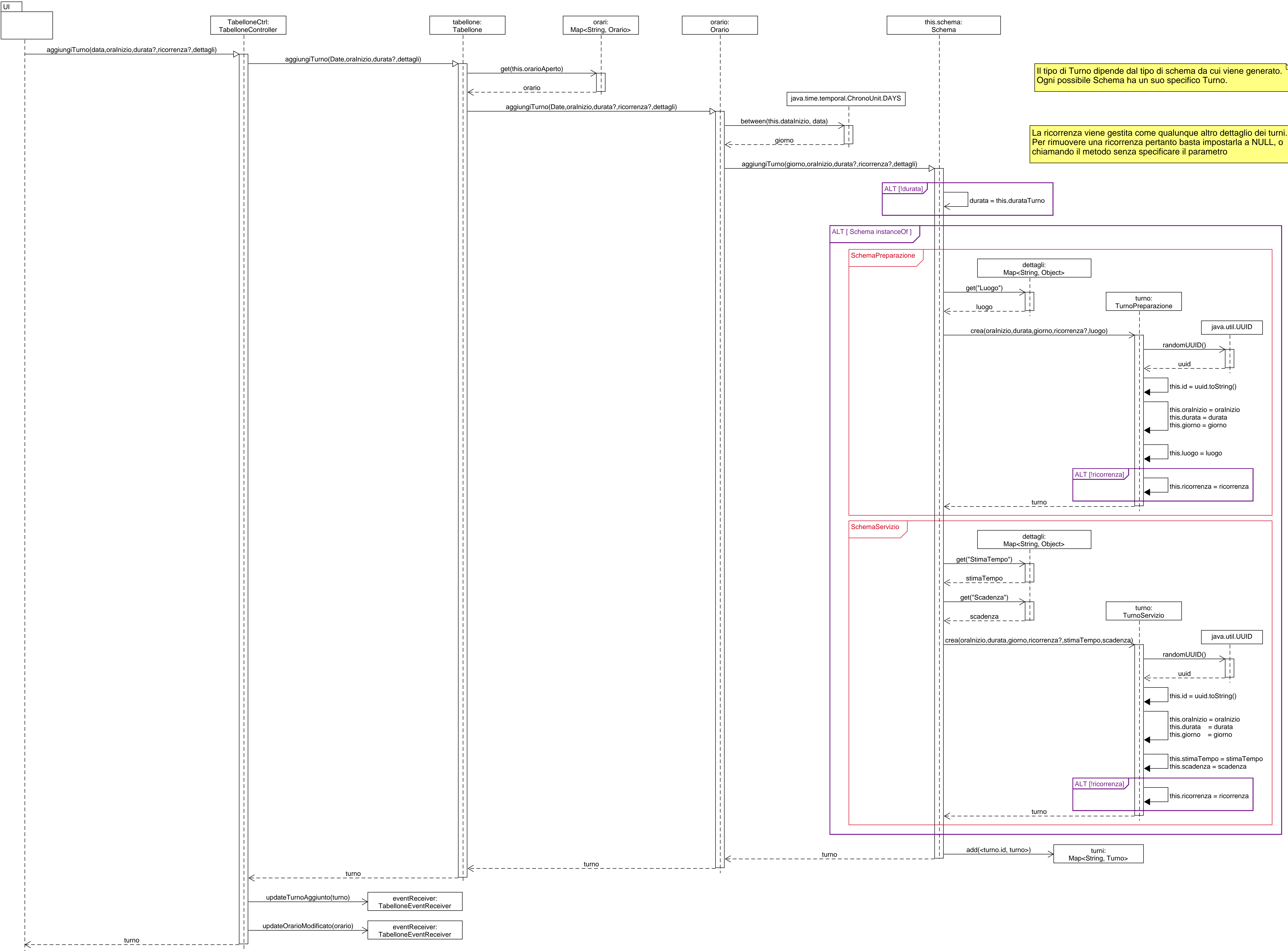


Viene rimosso l'orarioAperto

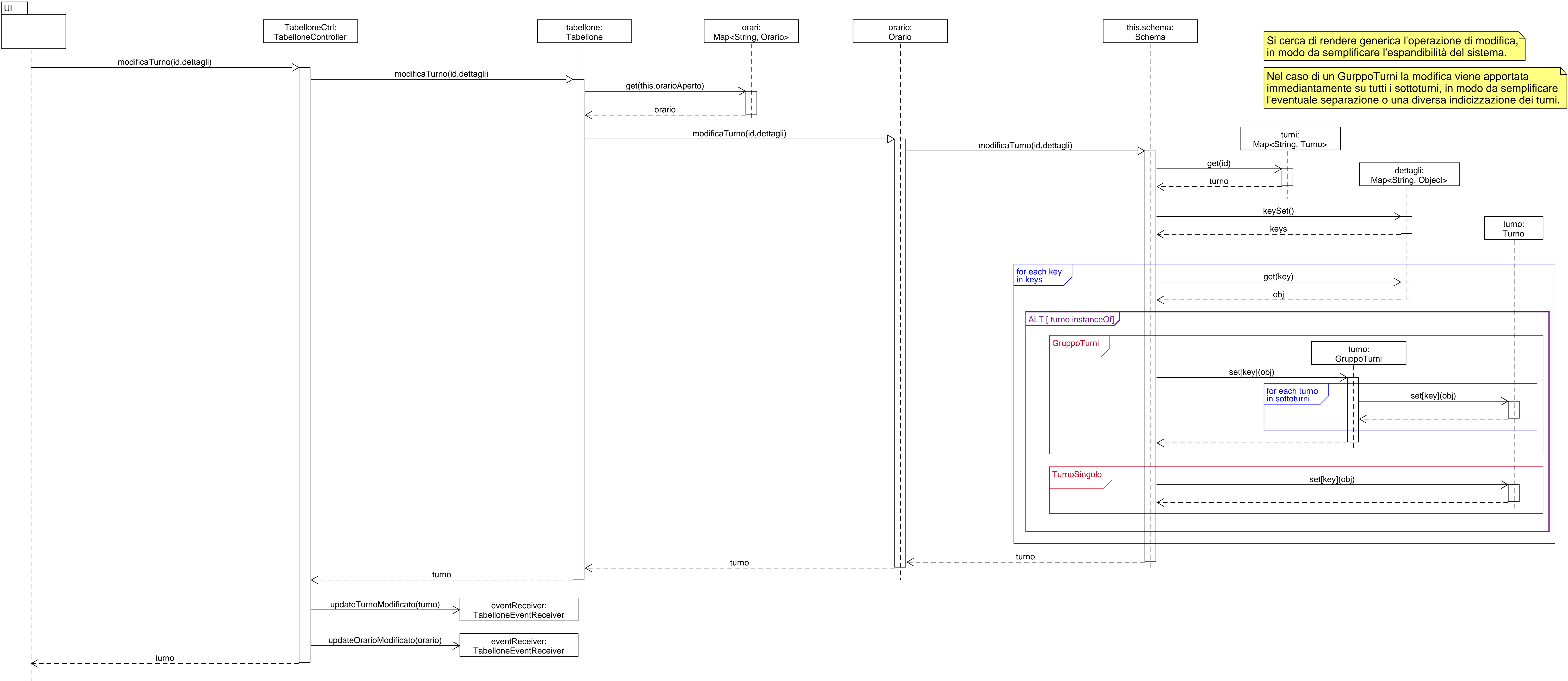
DSD per l'operazione:  
3. aggiungiTurno

Il tipo di Turno dipende dal tipo di schema da cui viene generato.  
Ogni possibile Schema ha un suo specifico Turno.

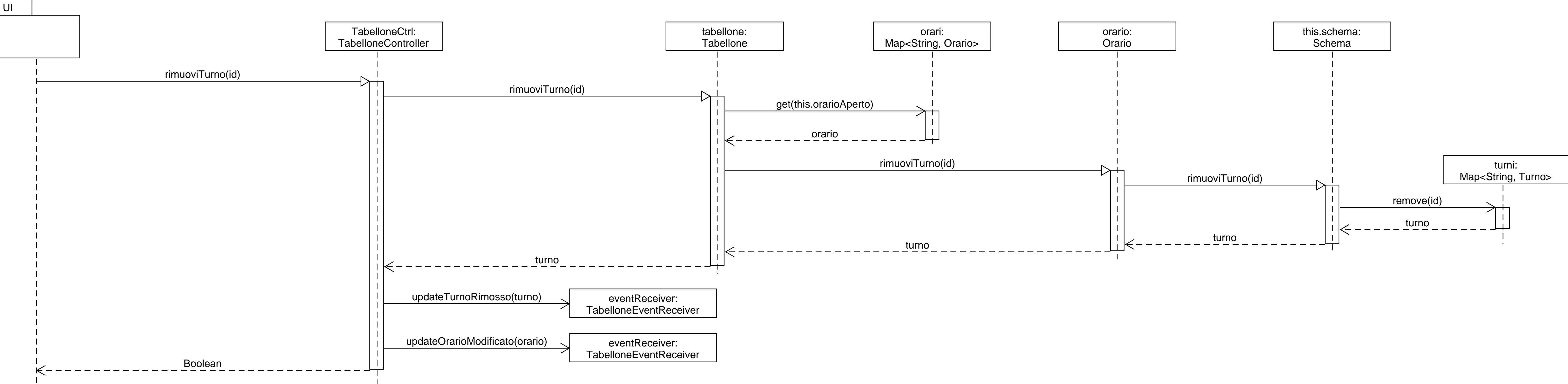
La ricorrenza viene gestita come qualunque altro dettaglio dei turni.  
Per rimuovere una ricorrenza pertanto basta impostarla a NULL, o  
chiamando il metodo senza specificare il parametro



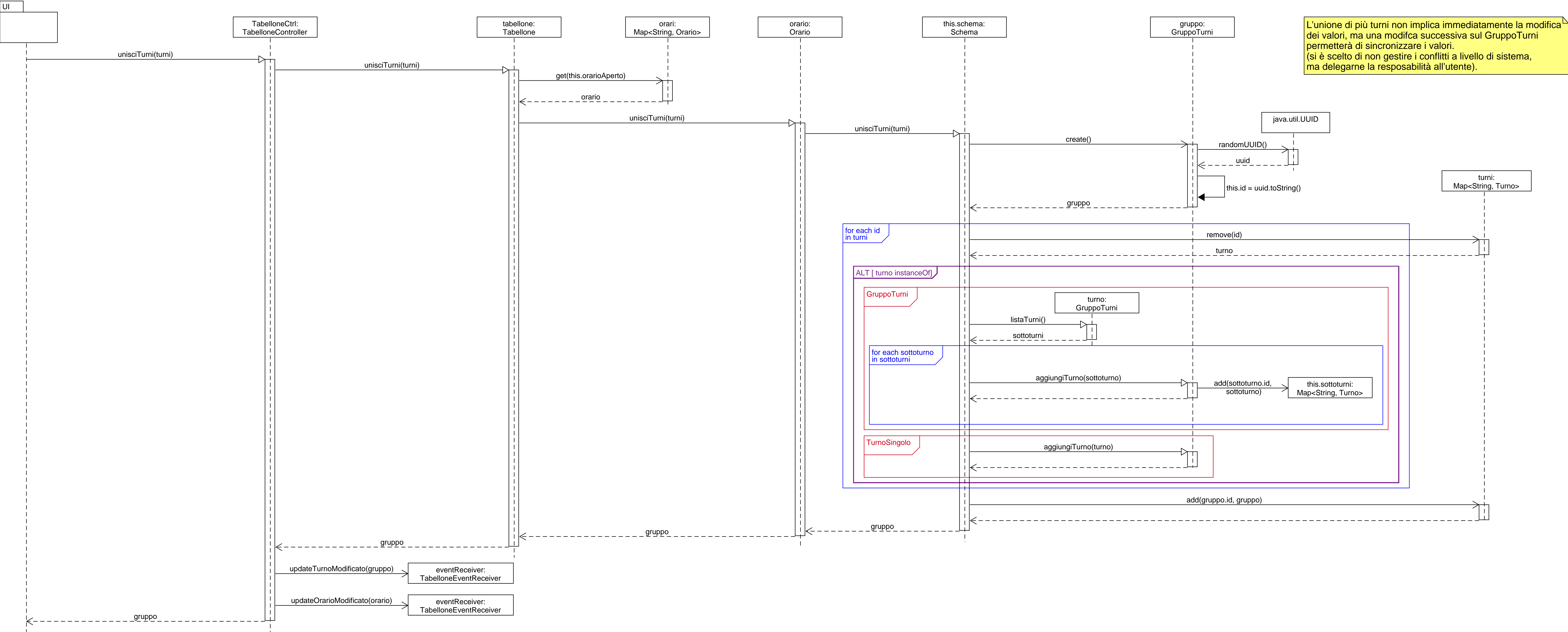
DSD per l'operazione:  
3a. modificaTurno



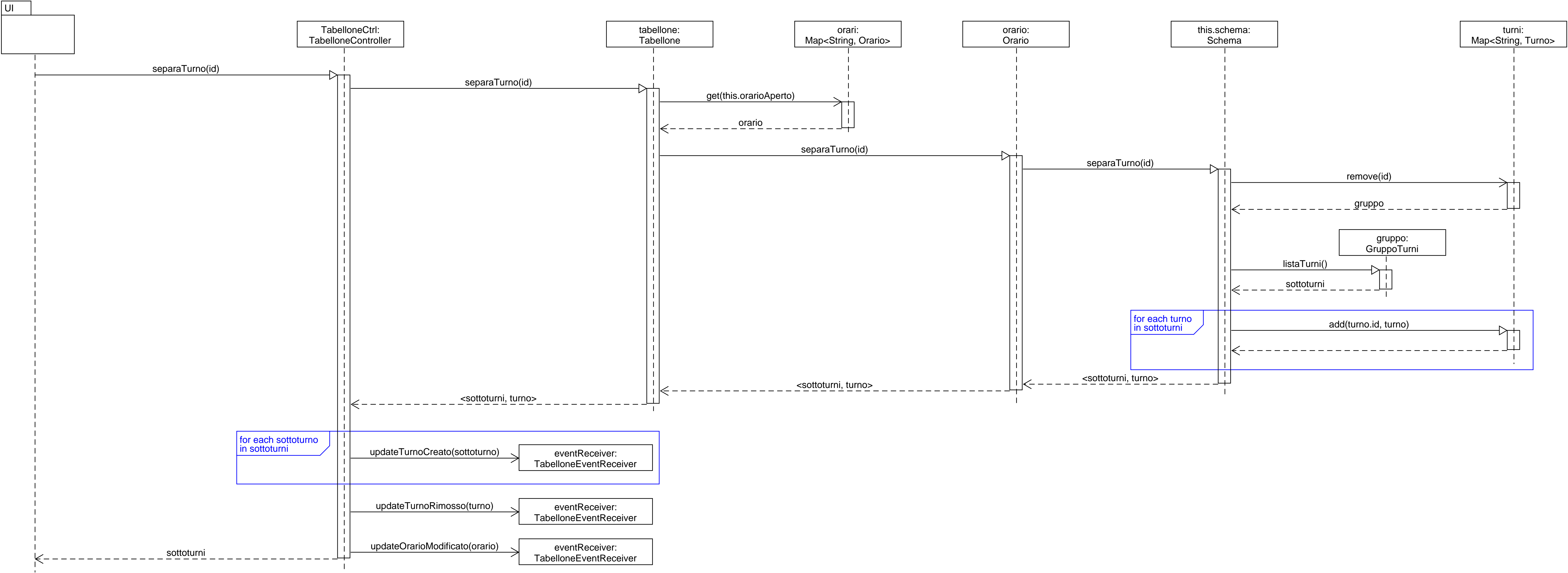
DSD per l'operazione:  
3b. rimuoviTurno



DSD per l'operazione:  
4. unisciTurni



DSD per l'operazione:  
4a. separaTurno



DSD per l'operazione:  
5. assegnaCuoco

