

Gestire i compiti della cucina

Informazioni generali

Nome caso d'uso: Gestire i compiti della cucina
Portata: Sistema
Livello: Obiettivo utente
Attore primario: Chef
Parti Interessate: Cuochi
Pre-condizioni: L'attore deve essere autenticato come Chef
Garanzie di successo o post-condizioni: I compiti della cucina sono assegnati (è stato indicato chi, quando e cosa fa) e risultano consultabili.

#	Attore	Sistema
1	Crea un FoglioRiepilogativo per un Servizio di un Evento	Registra un nuovo FoglioRiepilogativo
2	Opzionalmente aggiunge Ricette e Rpreparazioni.	Registra una Ricetta o Preparazione come Compito
3	Opzionalmente ordina i Compiti per importanza	Presenta i Compiti registrati nel FoglioRiepilogativo ordinate per importanza
4	Opzionalmente imposta il Turno in cui deve essere svolto un Compito	Assegna il Turno al Compito specificato
5	Opzionalmente assegna ad un Cuoco un Compito	Assegna il Cuoco al Compito specificato
	Ripetere 2 finché non si è soddisfatti	

Scenario principale di successo

Eccezione 1.1a

#	Attore	Sistema
1.1a.1	Crea un FoglioRiepilogativo per un Servizio di un Evento	L'evento non è assegnato all'Attore
	<i>Termina il caso d'uso</i>	

Eccezione 1.1b

#	Attore	Sistema
1.1b.1	Crea un FoglioRiepilogativo per un Servizio di un Evento	È già presente un foglioRiepilogativo per questo Servizio
	<i>Termina il caso d'uso</i>	

Estensione 1a

#	Attore	Sistema
1a.1	Modifica un foglio riepilogativo	Fornisce un foglio riepilogativo e ne permette la modifica
	<i>Termina il caso d'uso</i>	

Eccezione 1a.1a

#	Attore	Sistema
1a.1a.1	Modifica un foglio riepilogativo	Il FoglioRiepilogativo non appartiene all'Attore
	<i>Termina il caso d'uso</i>	

Eccezione 1a.1b

#	Attore	Sistema
1a.1b.1	Modifica un foglio riepilogativo	foglioRiepilogativo non esiste
	<i>Termina il caso d'uso</i>	

Estensione 2a

#	Attore	Sistema
2a.1	Modifica un Compito specificando Opzionalmente: <ul style="list-style-type: none">- Stima del tempo- Se è da preparare- Se è stato completato- La difficoltà- L'importanza- Quante porzioni	Modifica il Compito registrando i nuovi valori inseriti

Eccezione 2a.1a

#	Attore	Sistema
2a.1a.1	Modifica un Compito	Il Compito non è registrato nel foglioRiepilogativo

Eccezione 2a.1b

#	Attore	Sistema
2a.1b.1	Modifica un Compito	il tempo del Compito supera la durata del Turno in cui è svolto

Estensione 2b

#	Attore	Sistema
2b.1	Rimuovi un Compito	È stato rimosso il Compito che non è più registrato nel foglioRiepilogativo

Eccezione 2b.1a

#	Attore	Sistema
2b.1a.1	Rimuovi un Compito	Il Compito non è registrato nel foglioRiepilogativo

Eccezione 3.1a

#	Attore	Sistema
3.1a.1	ordina i Compiti per <i>importanza</i>	Non sono presenti Compiti registrati nel foglioRiepilogativo

Estensione 3a

#	Attore	Sistema
3a.1	ordina i Compiti per <i>difficoltà</i>	Presenta i Compiti registrati nel FoglioRiepilogativo ordinate per difficoltà

Eccezione 3a.1a

#	Attore	Sistema
3a.1a.1	ordina i Compiti per <i>difficoltà</i>	Non sono presenti Compiti registrati nel foglioRiepilogativo

Eccezione 4.1a

#	Attore	Sistema
4.1a.1	imposta il Turno in cui deve essere svolto un Compito	Il Compito non è registrato nel foglioRiepilogativo

Eccezione 4.1b

#	Attore	Sistema
4.1b.1	imposta il Turno in cui deve essere svolto un Compito	Il Cuoco assegnato al compito non è assegnato al Turno

Eccezione 4.1c

#	Attore	Sistema
4.1c.1	imposta il Turno in cui deve essere svolto un Compito	Il tempo necessario per il Compito è superiore alla durata del Turno

Eccezione 5.1a

#	Attore	Sistema
5.1a.1	assegna ad un Cuoco un Compito	Il Compito non è registrato nel foglioRiepilogativo

Eccezione 5.1b

#	Attore	Sistema
5.1b.1	assegna ad un Cuoco un Compito	Il Cuoco non è assegnato al Turno a cui è svolto il Compito

Contratti per lo UC “Gestire i compiti della cucina”

Pre-condizione generale:

- l'attore è identificato con un'istanza *chef* di Chef
- ListaFogliRiepilogativi *listaFogliRiepilogativi* contiene tutti i fogli riepilogativi creati

1. creaFoglioRiepilogativo(servizio: Servizio): FoglioRiepilogativo

Pre-condizioni:

- il *servizio* è previsto da un Evento *evento*
- *evento* è **assegnato a chef**
- il Menu *menu* è **in uso** nel *servizio*
- [Non esiste *foglioRiepilogativo* FoglioRiepilogativo **contenuto in** *listaFogliRiepilogativi*]
 - tale per cui *foglioRiepilogativo*.titolo = *servizio*.titolo

Post-condizioni:

- è stata creata un'istanza *foglio* di FoglioRiepilogativo che **appartiene a chef** e **contenuta in listaFogliRiepilogativi**
- *foglio*.servizio = *servizio*
- *foglio*.titolo = *servizio*.nome
- *chef* è **proprietario** di *foglio*
- [per ogni Voce *voce* **appartenente a menu**]
 - [per ogni Mansione *mansione* **riferita a voce**]
 - è stata creata un'istanza *compito* di Compito **contenuta in foglio** che **consiste in mansione**
 - *compito*.completato = FALSE
 - *compito*.daPreparare = TRUE
 - *compito*.tempo=*mansione*.tempo
 - [per ogni Preparazione *preparazione* **di cui cosiste mansione**]
 - è stata creata un'istanza *compito* di Compito **contenuta in foglio** che **consiste in preparazione**
 - *compito*.completato = FALSE
 - *compito*.daPreparare = TRUE
 - *compito*.tempo= *preparazione*.tempo

1a. modificaFoglioRiepilogativo(titolo: String): FoglioRiepilogativo

Pre-condizioni:

- [esiste *foglio* FogliRiepilogativi è **contenuto in listaFogliRiepilogativi**]
 - tale per cui *foglio*.titolo = *titolo*
- *foglio* **appartiene a chef**

Post-condizioni:

- *foglio* ora risulta come foglioo aperto è quindi è modificabile
- *foglio*Aperto = *titolo*

1b. rimuoviFoglioRiepilogativo(): Boolean

Pre-condizioni:

- [*foglioAperto* è definito] è in corso la modifica di FoglioRiepilogativo *foglio contenuto in listaFogliRiepilogativi*
- *foglio appartiene a chef*

Post-condizioni:

- è stato rimosso *foglio* che ora non è più **contenuto in listaFogliRiepilogativi**
- *foglioAperto* = NULL

2. aggiungiCompito(mansione: Mansione, tempo?: int, daPreparare?: boolean): Compito

Pre-condizioni:

- [*foglioAperto* è definito] è in corso la modifica di FoglioRiepilogativo *foglio contenuto in listaFogliRiepilogativi*
- *foglio appartiene a chef*

Post-condizioni:

- è stata creata un'istanza *compito* di Compito **contenuta in foglio** che **consiste in mansione**
- *compito.completato* = FALSE
- [se *tempo* è definito] *compito.tempo* = *tempo*
- [altrimenti] *compito.tempo* = *mansione.tempo*
- [se *daPreparare* è definito] *compito.daPreparare* = *daPreparare*
- [altrimenti] *compito.daPreparare* = TRUE

2a. modificaCompito(id: String, tempo?: int, daPreparare?: boolean, completato?: boolean, difficolta?:Int, importanza?:Int, porzioni?:Int): Compito

Pre-condizioni:

- [*foglioAperto* è definito] è in corso la modifica di FoglioRiepilogativo *foglio contenuto in listaFogliRiepilogativi*
- *foglio appartiene a chef*
- Esiste il Compito *compito* **contenuto in foglio** associato ad *id*
- [se *tempo* è definito] [se *compito* è **svolto in turno** Turno]
 - *turno.tempo* <= *tempo*

Post-condizioni:

- è stata modificato *compito*:
 - [se *tempo* è definito] *compito.tempo* = *tempo*
 - [se *daPreparare* è definito] *compito.daPreparare* = *daPreparare*
 - [se *completato* è definito] *compito.completato* = *completato*
 - [se *difficolta* è definito] *compito.difficolta* = *difficolta*
 - [se *importanza* è definito] *compito.importanza* = *importanza*
 - [se *porzioni* è definito] *compito.porzioni* = *porzioni*

2b. rimuoviCompito(id: String): Boolean

Pre-condizioni:

- [*foglioAperto* è definito] è in corso la modifica di FoglioRiepilogativo *foglio* **contenuto in listaFogliRiepilogativi**
- *foglio* **appartiene a chef**
- Esiste il Compito *compito* **contenuto in foglio** associato ad *id*

Post-condizioni:

- *compito* non è più **contenuto in foglio**

3. ordinaCompitiImportanza(metodo: int): ArrayList<Compiti>

Pre-condizioni:

- [*foglioAperto* è definito] è in corso la modifica di FoglioRiepilogativo *foglio* **contenuto in listaFogliRiepilogativi**
- *foglio* **appartiene a chef**
- Esiste almeno 2 istanze *compito* Compito **contenuto in foglio**

Post-condizioni:

- [*metodo* < 0] è stata restituita una lista ordinata delle istanze *compito* Compito in base a *compito.Importanza* CRESCENTE
- [*metodo* > 0] è stata restituita una lista ordinata delle istanze *compito* Compito in base a *compito.Importanza* DECRESCENTE

3a. ordinaCompitiDifficolta(metodo: int): ArrayList<Compiti>

Pre-condizioni:

- [*foglioAperto* è definito] è in corso la modifica di FoglioRiepilogativo *foglio* **contenuto in listaFogliRiepilogativi**
- *foglio* **appartiene a chef**
- Esiste almeno 2 istanze *compito* Compito **contenuto in foglio**

Post-condizioni:

- [*metodo* < 0] è stata restituita una lista ordinata delle istanze *compito* Compito in base a *compito.difficolta* CRESCENTE
- [*metodo* > 0] è stata restituita una lista ordinata delle istanze *compito* Compito in base a *compito.difficolta* DECRESCENTE

4. impostaTurno(id: String, turno: Turno): Boolean

Pre-condizioni:

- [*foglioAperto* è definito] è in corso la modifica di FoglioRiepilogativo *foglio* **contenuto in listaFogliRiepilogativi**
- *foglio* **appartiene a chef**
- Esiste il Compito *compito* **contenuto in foglio** associato ad *id*
- [se *compito.cuoco* è definito] *compito.cuoco* è **assegnato a turno**

Post-condizioni:

- *compito* è **svolto in turno**

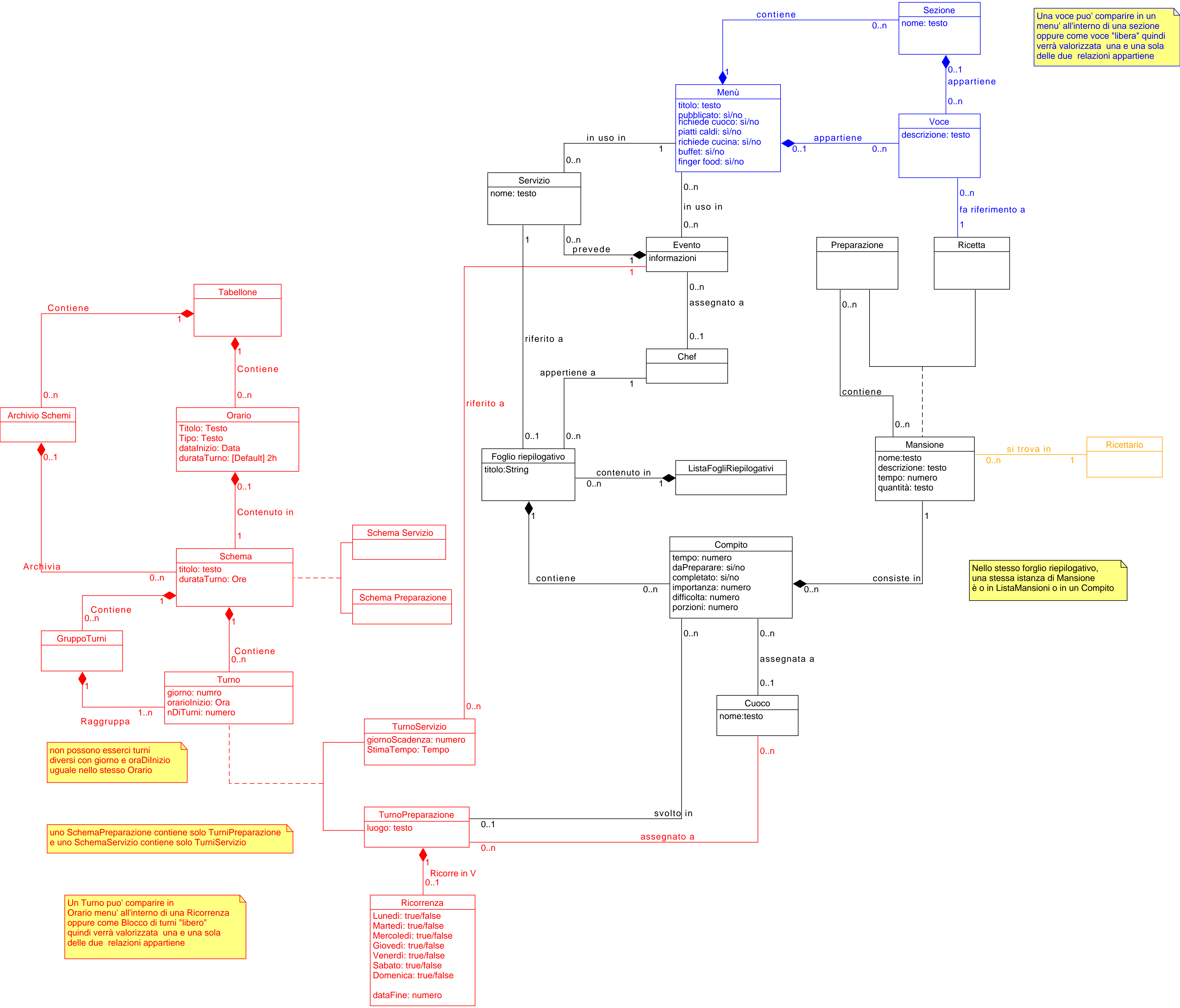
5. assegnaCuoco(id: String, cuoco: Cuoco):Boolean

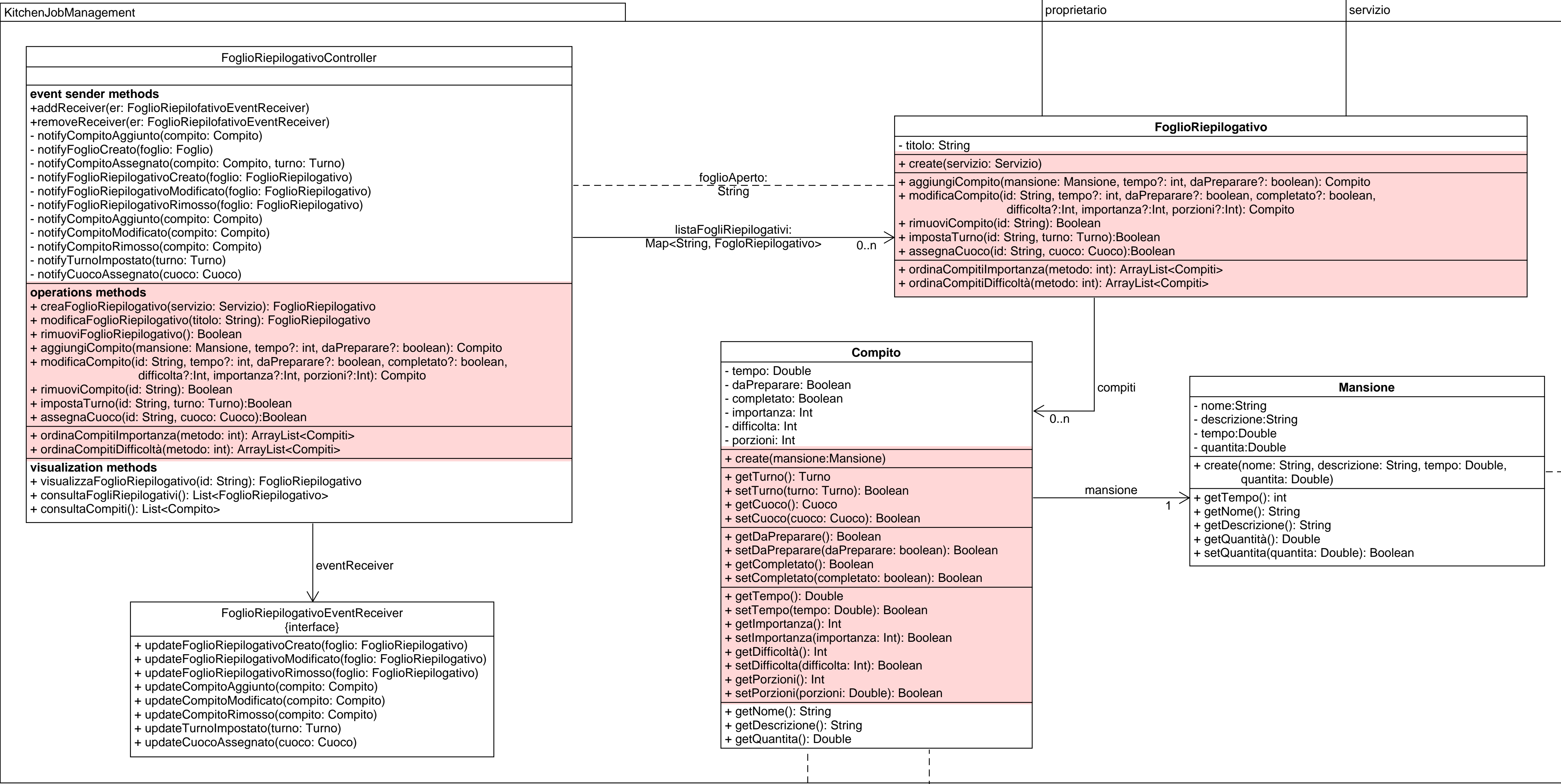
Pre-condizioni:

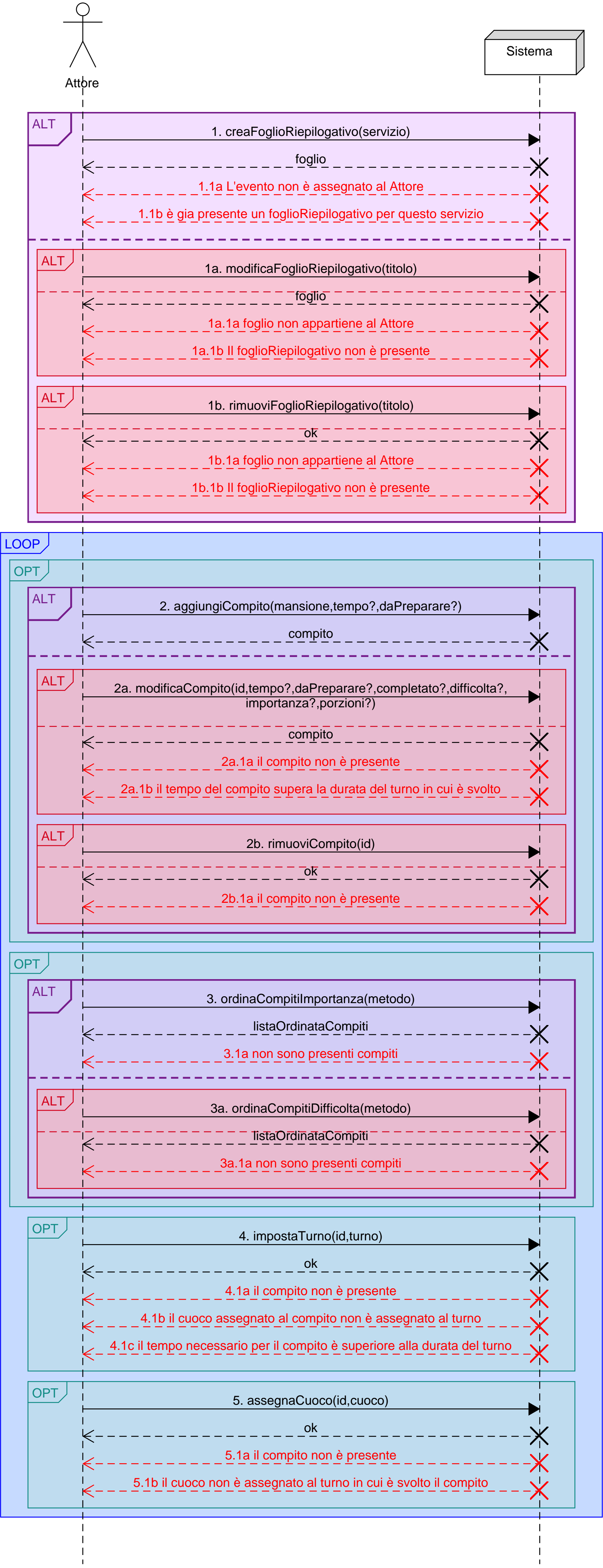
- [*foglioAperto* è definito] è in corso la modifica di *FoglioRiepilogativo* *foglio* **contenuto in listaFogliRiepilogativi**
- *foglio* **appartiene a chef**
- Esiste il Compito *compito* **contenuto in foglio** associato ad *id*
- [se *compito.turno* è definito] *cuoco* è **assegnato a compito.turno**

Post-condizioni:

- *cuoco* è stato **assegnato a compito**

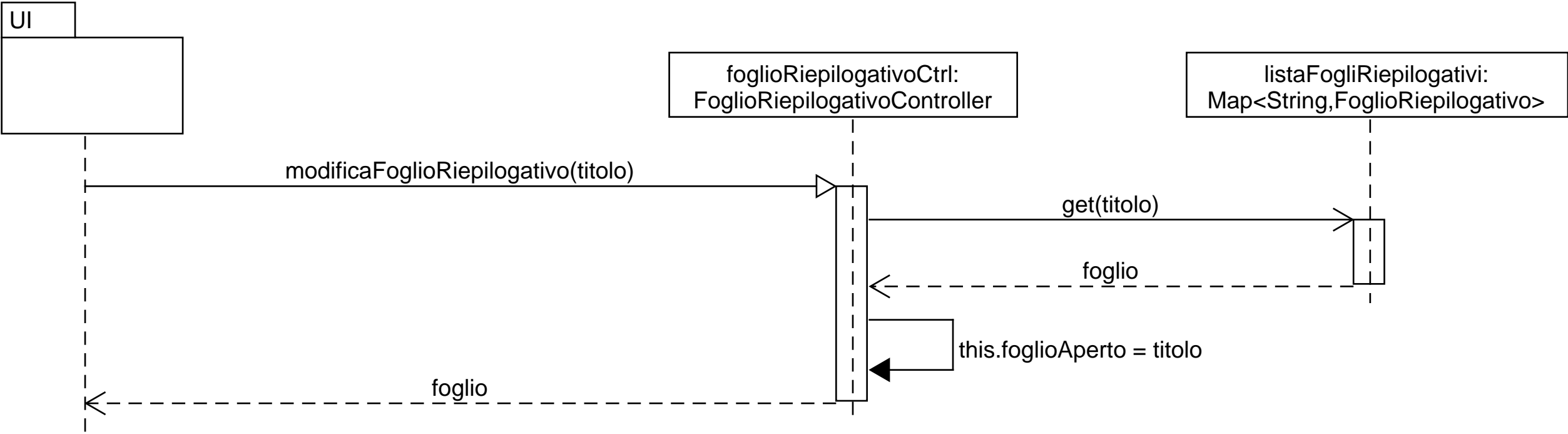




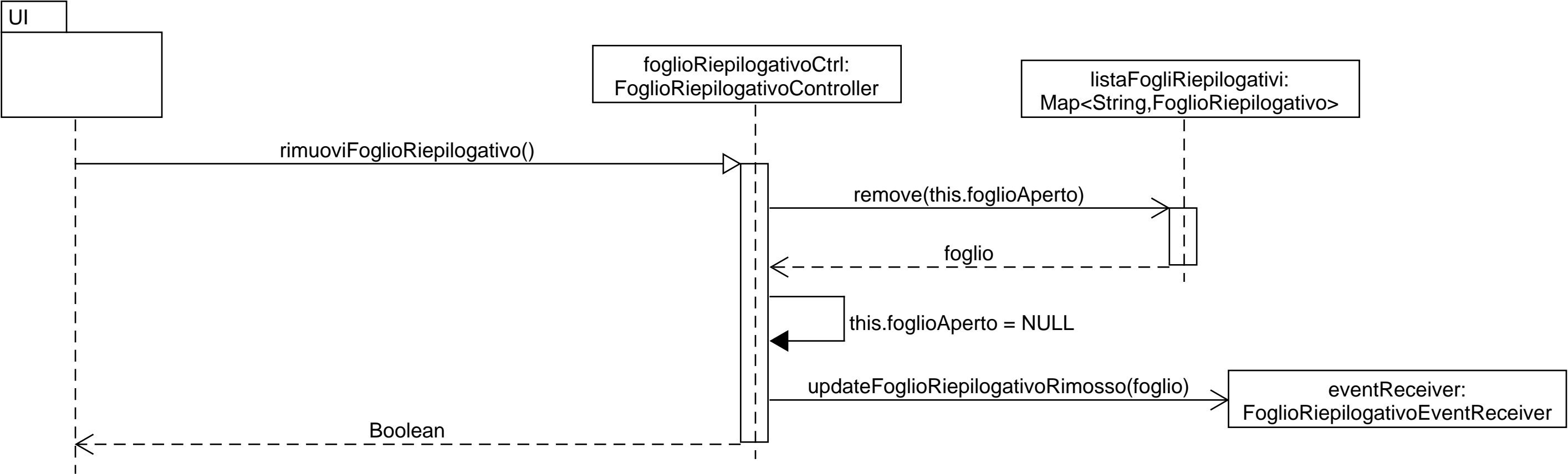


DSD per l'operazione:
1. creaFoglioRiepilogativo

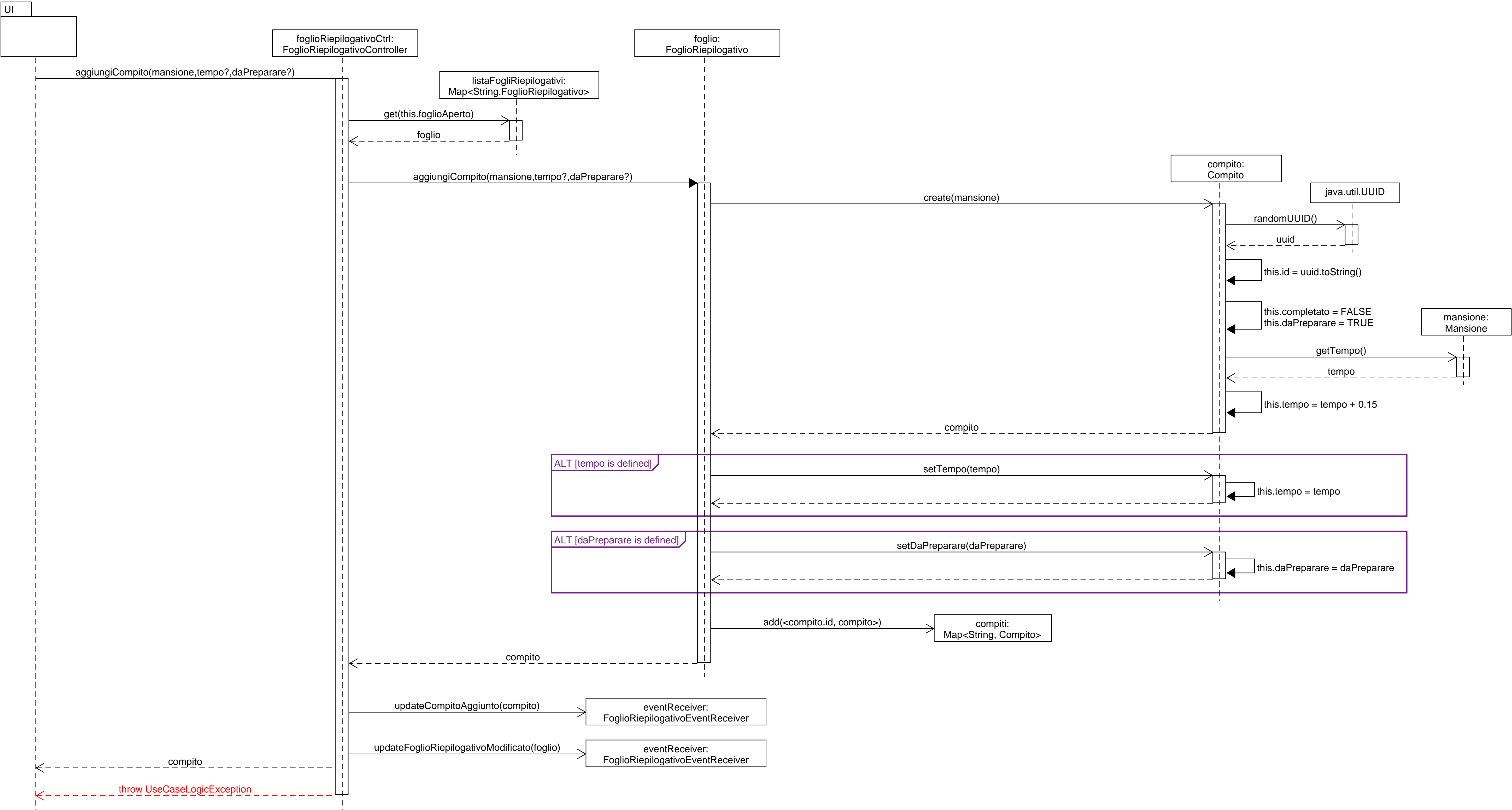
DSD per l'operazione:
1a. modificaFoglioRiepilogativo



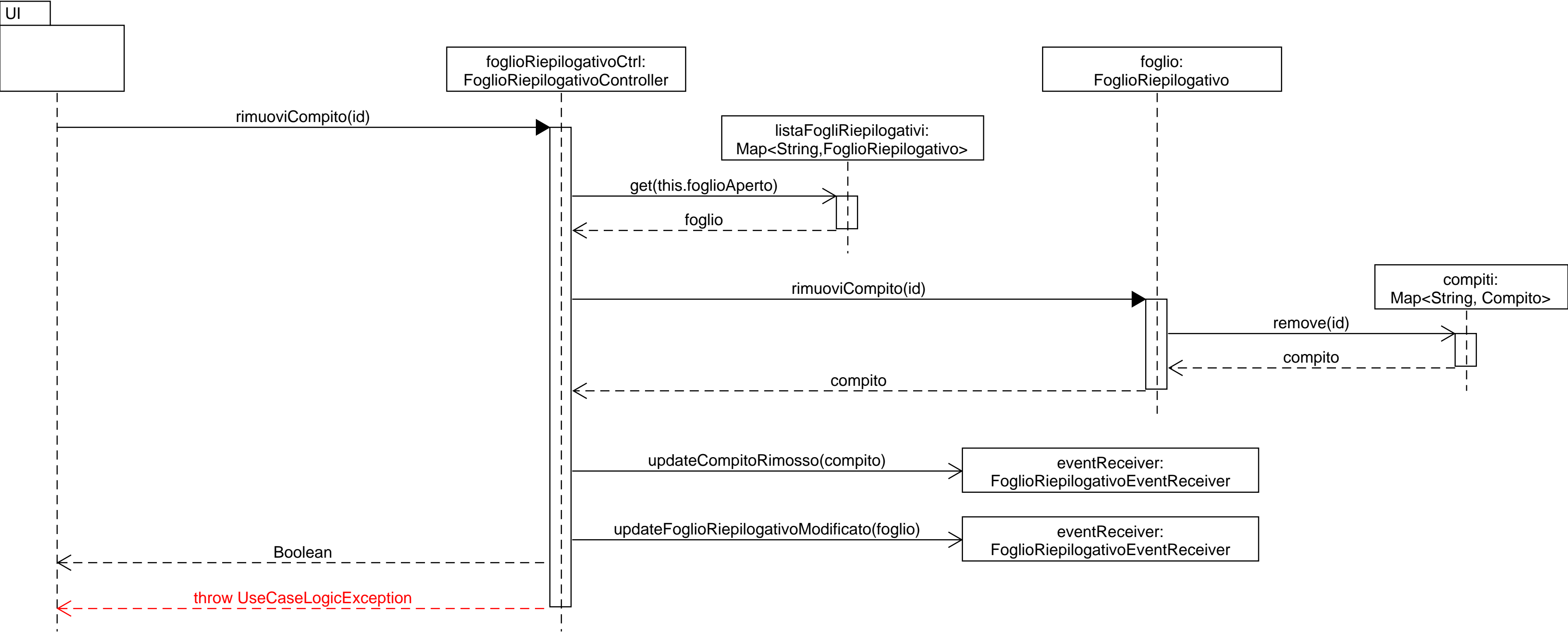
DSD per l'operazione:
1b. rimuoviaFoglioRiepilogativo



DSD per l'operazione:
2. aggiungiCompito



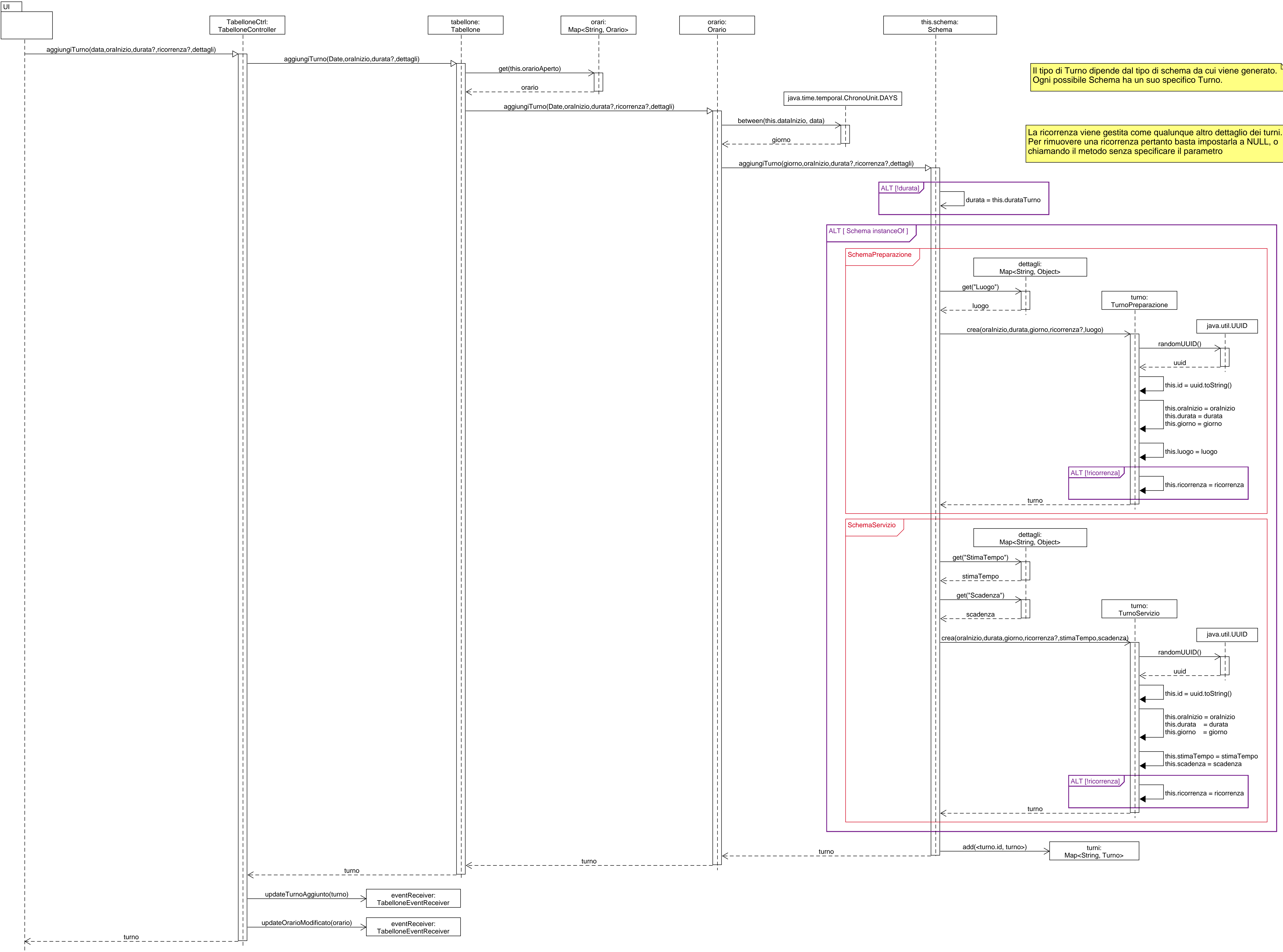
DSD per l'operazione:
2b. rimuoviCompito



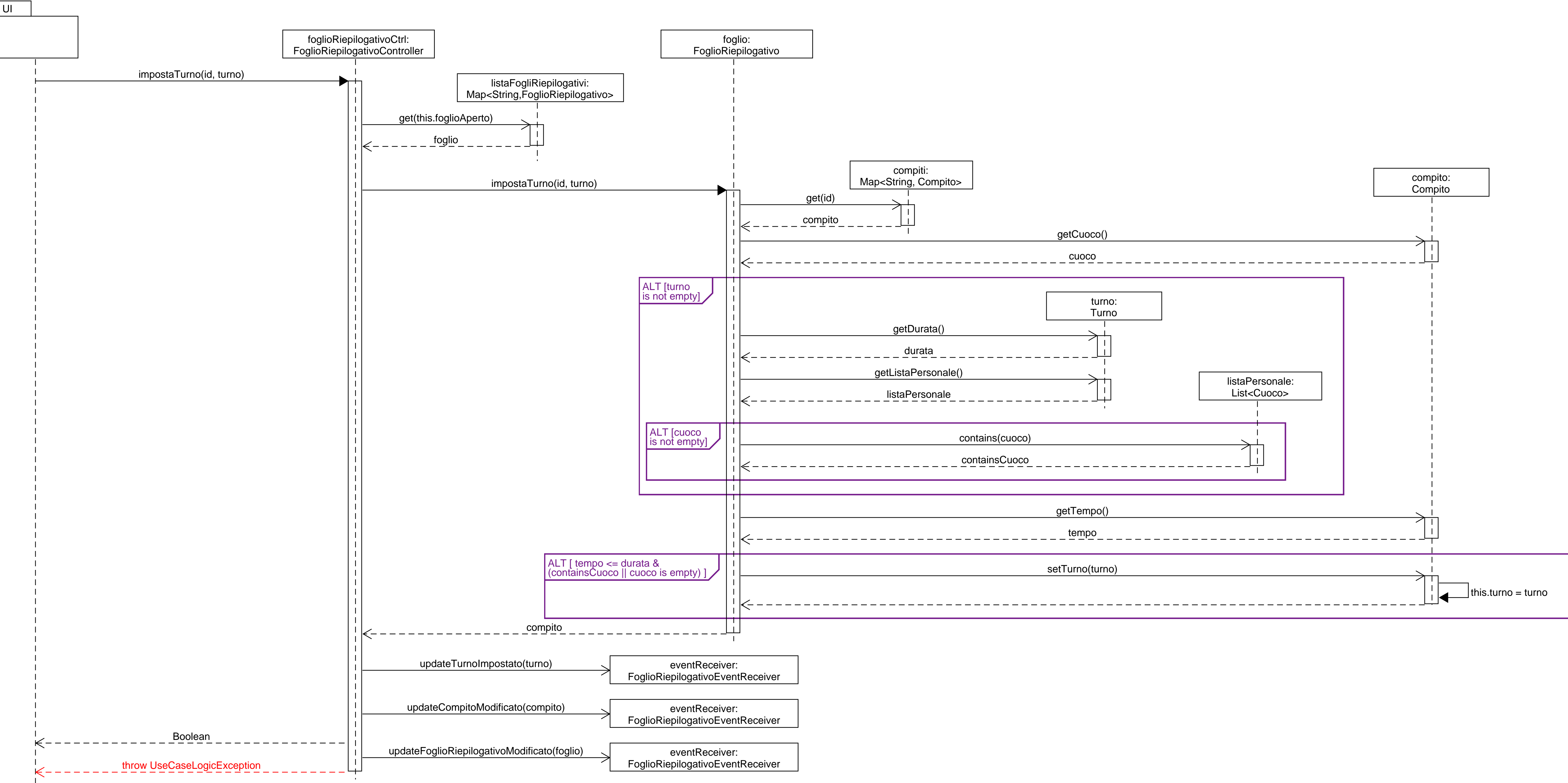
DSD per l'operazione:
3. aggiungiTurno

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:
4. impostaTurno



DSD per l'operazione:
5. assegnaCuoco

