



Progetto di Ingegneria del software



Sistema informativo
per la gestione di
uno stabilimento
balneare

Ingegneria delle telecomunicazioni

Antonio Esiliato,
Silvana Pizzonia

INDICE

Descrizione	5
Glossario	6
Analisi dei requisiti	8
Requisiti Funzionali	8
Gestione Utilizzatore	9
Gestione Amministratore	9
Gestione Attività	11
Gestione Sistema	14
Requisiti Non Funzionali	15
Matrice di mapping	16
Diagrammi dei casi d'uso	17
Attori	18
Gestione Amministratore	19
AssegnaConPrenotazione	20
AssegnaSenzaPrenotazione	21
CUDRicevuta	23
CUDServizio.....	24
GestisciUscita	25
InserisciAssegnamento	26
ModificaAbbonamento.....	27
ModificaAssegnamento	28
RicercaAssegnamento.....	29
RicercaRicevuta	30
RicercaServizio	31
Gestione Attività	32
ConsegnaServizio	34
DisdettaPrenotazione	35
InserisciAbbonamento	36
InserisciPrenotazione.....	37
ModificaPrenotazione.....	38
RicercaAbbonamento	39
RicercaPrenotazione	40
VisualizzaAbbonamento	41
VisualizzaAssegnamento.....	42
VisualizzaPrenotazione	43
VisualizzaRicevuta	44
VisualizzaServizio	45
Gestione Sistema	46
Backup.....	47
GestisciStatistiche	48
ScadenzaAbbonamento	49
ScadenzaPrenotazione.....	50
VerificaAbbonamentoScaduto.....	51
Gestione Utilizzatore	52
CUDBagnino	53
CUDCliente	54
CUDUtilizzatore	55

RicercaBagnino.....	56
RicercaCliente	57
RicercaUtilizzatore	58
VisualizzaBagnino.....	59
VisualizzaCliente	60
VisualizzaUtilizzatore	61
Package di analisi	62
Attività	63
Servizio	71
Diagrammi di sequenza	76
AssegnaConPrenotazione	76
AssegnaSenzaPrenotazione	77
Backup	79
CUDCliente	80
Consegna Servizio	81
DisdettaPrenotazione	82
GestisciStatistiche	83
GestisciUscita	84
InserisciAbbonamento	85
InserisciPrenotazione	86
ModificaAbbonamento	88
RicercaCliente	89
ScadenzaAbbonamento	90
ScadenzaPrenotazione	91
VerificaAbbonamentoScaduto	92
VisualizzaCliente	93
Diagrammi di attività	94
Assegna senza prenotazione	94
Backup	99
CUD Cliente	100
Gestisci statistiche	101
Gestisci uscita	102
Inserisci prenotazione	105
Ricerca Cliente	110
Classi di progettazione	112
Attività	113
Gestione	130
Servizio	135
Diagramma dei componenti	146
Diagrammi di sequenza raffinati	147
AssegnaConPrenotazione	147
AssegnaSenzaPrenotazione	149
Backup	150
CUDCliente	151
Consegna Servizio	152
DisdettaPrenotazione	153
GestisciStatistiche	154
GestisciUscita	155
InserisciAbbonamento	156
InserisciPrenotazione	157
ModificaAbbonamento	159

RicercaCliente	160
ScadenzaAbbonamento	161
ScadenzaPrenotazione	162
VerificaAbbonamentoScaduto	163
VisualizzaCliente	164
Macchine a stati	165
Abbonamento	165
Cliente	166
Ombrellone	167
Prenotazione	168
Diagramma di deployment	169
Gantt Project	170

DESCRIZIONE

Il progetto proposto consiste nella realizzazione di un sistema informativo per uno stabilimento balneare. Siamo interessati a tutti gli aspetti della gestione di questo tipo di attività: rapporti con i clienti e servizi offerti (ombrelloni, lettini prendisole, sedie sdraio, pedalò e cabine). Per ciascun servizio si dovranno memorizzare codice, disponibilità, il prezzo e due variabili booleane, “consegnato” e “disdetto”, che mantengono aggiornate le informazioni, rispettivamente, sulla consegna o eventuale disdetta di un servizio precedentemente prenotato.

Il costo di ingresso allo stabilimento è di 1 euro a persona.

Lo stabilimento è aperto 120 giorni all'anno, dall' 1 giugno al 30 Settembre.

Le cabine sono tutte uguali e possono essere noleggiate solo ai clienti che hanno preventivamente effettuato la prenotazione.

Gli ombrelloni sono disposti nella spiaggia per fila: il numero della fila è crescente all'aumentare della distanza dal mare.

Il pedalò può essere noleggiato a ore e in tal caso il costo del noleggio viene calcolato in base alle ore di utilizzo.

I dipendenti dello stabilimento balneare sono un cassiere (amministratore) e uno o più assistenti bagnanti (bagnini).

I bagnini sono registrati con i loro dati anagrafici (nome, cognome, data e luogo di nascita, codice fiscale), telefono, indirizzo e-mail e licenza. I bagnini hanno il compito di consegnare i servizi ai clienti.

Per ogni assegnamento è necessario emettere la ricevuta. Si precisa che ogni assegnamento si riferisce ad uno o più servizi. I servizi sono assegnati al massimo per l'intera giornata dal momento in cui il cliente arriva effettivamente allo stabilimento balneare. La ricevuta è identificata da codice e anno e presenta la data e l'ora di emissione.

Il cassiere dovrà gestire l'ingresso e l'uscita dei clienti dallo stabilimento e l'assegnamento dei servizi.

I clienti dovranno usufruire di almeno un servizio per poter accedere allo stabilimento. Per ogni cliente si dovranno memorizzare cognome, nome, codice fiscale, indirizzo e-mail, telefono, età, oltre a un codice identificativo. E', infine, previsto un ulteriore campo per memorizzare eventuali note. Un cliente può, inoltre, sottoscrivere un abbonamento, che può essere mensile o stagionale.

L'abbonamento è identificato da un codice; per ogni abbonamento si dovranno memorizzare la data di rilascio e la data di inizio validità.

I clienti abbonati sono esentati dal pagamento del costo d'ingresso allo stabilimento; inoltre, esclusivamente a questi è consentita la prenotazione dei servizi.

Per ciascuna prenotazione verranno memorizzati un codice, la data e l'ora di inizio. La data di fine prenotazione coincide con la data di scadenza dell'abbonamento associato. La prenotazione si può riferire ad uno o più servizi.

Non è possibile prenotare un pedalò.

Il costo della prenotazione è addebitato al cliente all'atto della prenotazione stessa.

Le prenotazioni devono essere effettuate almeno un giorno prima della data di inizio della prenotazione stessa e possono esser dis dette al più 7 giorni prima della data di inizio; in ogni caso il rimborso è parziale e pari al 50% dell'ammontare del costo del servizio.

Una procedura di backup del sistema viene automaticamente eseguita tutte le sere alle ore 23:30.

L'amministratore, inoltre, può effettuare statistiche sui tutti i dati presenti nel sistema.

GLOSSARIO

Abbonamento

Documento che viene rilasciato al cliente che ne abbia fatto richiesta e che consente di effettuare prenotazioni ed esime il cliente dal costo di ingresso allo stabilimento.

Sinonimi: Tessera

Omonimi: Nessuno

Amministratore

Dipendente dello stabilimento, al quale è conferito il potere di gestione del sistema.

Sinonimi: Cassiere

Omonimi: Nessuno

Assegnamento

Affitto di uno o più servizi ad un cliente per un certo tempo, previo pagamento.

Sinonimi: Noleggio

Omonimi: Nessuno

Backup

Copia di sicurezza dei dati del sistema.

Sinonimi: Nessuno

Omonimi: Nessuno

Bagnino

Dipendente dello stabilimento, ha il compito di tutelare la sicurezza dei clienti e di assegnare i servizi agli stessi.

Sinonimi: Assistente bagnante

Omonimi: Nessuno

Cabina

Spazio chiuso che serve da spogliatoio ai bagnanti.

Sinonimi: Nessuno

Omonimi: Nessuno

Cliente

Colui che fruisce dei servizi offerti dallo stabilimento.

Sinonimi: Bagnante

Omonimi: Nessuno

Codice

Numero assegnato ad un utente per accedere a una sezione riservata.

Sinonimi: ID

Omonimi: Nessuno

Lettino Prendisole

Sorta di letto, su cui il cliente può distendersi e prendere il sole.

Sinonimi: Lettino

Omonimi: Nessuno

Ombrellone

Grande ombrello fisso, che serve a riparare dal sole, sulle spiagge.

Sinonimi: Nessuno

Omonimi: Nessuno

Pedalò

Piccola imbarcazione a pedali.

Sinonimi: Nessuno

Omonimi: Nessuno

Prenotazione

Atto mediante il quale vengono riservati uno o più servizi ad un cliente per un certo tempo.

Sinonimi: Nessuno

Omonimi: Nessuno

Ricevuta

Documento sul quale si attestano i servizi assegnati o prenotati e l'importo versato dai beneficiari di tali servizi.

Sinonimi: Nessuno

Omonimi: Nessuno

Sedia Sdraio

Sedia costruita in modo che ci si possa sdraiare.

Sinonimi: Nessuno

Omonimi: Nessuno

Servizio

Complesso di strutture offerte ai clienti: ombrellone, sedia sdraio, lettino prendisole, pedalò.

Sinonimi: Nessuno

Omonimi: Nessuno

Stabilimento balneare

Edificio opportunamente attrezzato, posto sulla riva del mare.

Sinonimi: Nessuno

Omonimi: Nessuno

Statistiche

Sintesi dei dati di sistema atta ad analizzare l'andamento complessivo dello stabilimento.

Sinonimi: Nessuno

Omonimi: Nessuno

Utente

Chi accede al sistema.

Sinonimi: Nessuno

Omonimi: Nessuno

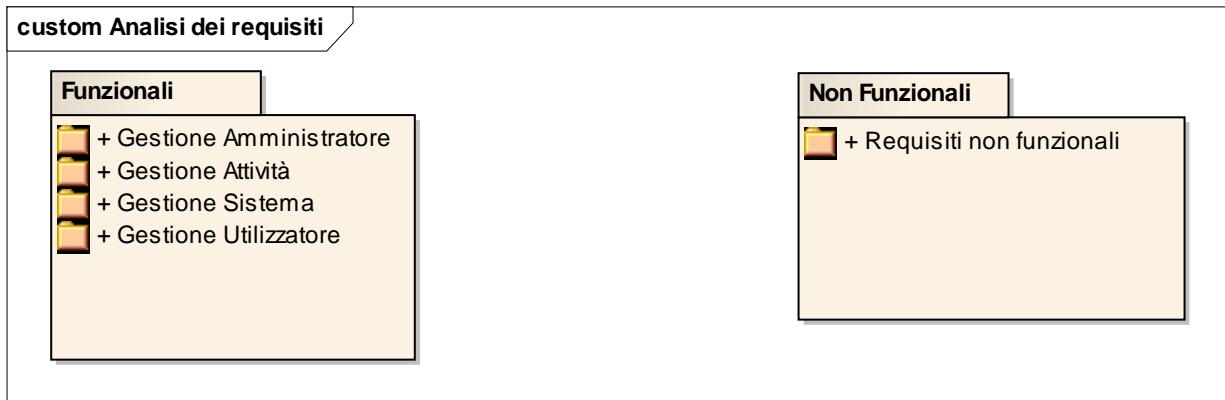
Utilizzatore

Chi usa il sistema software e non ha potere di gestione del sistema.

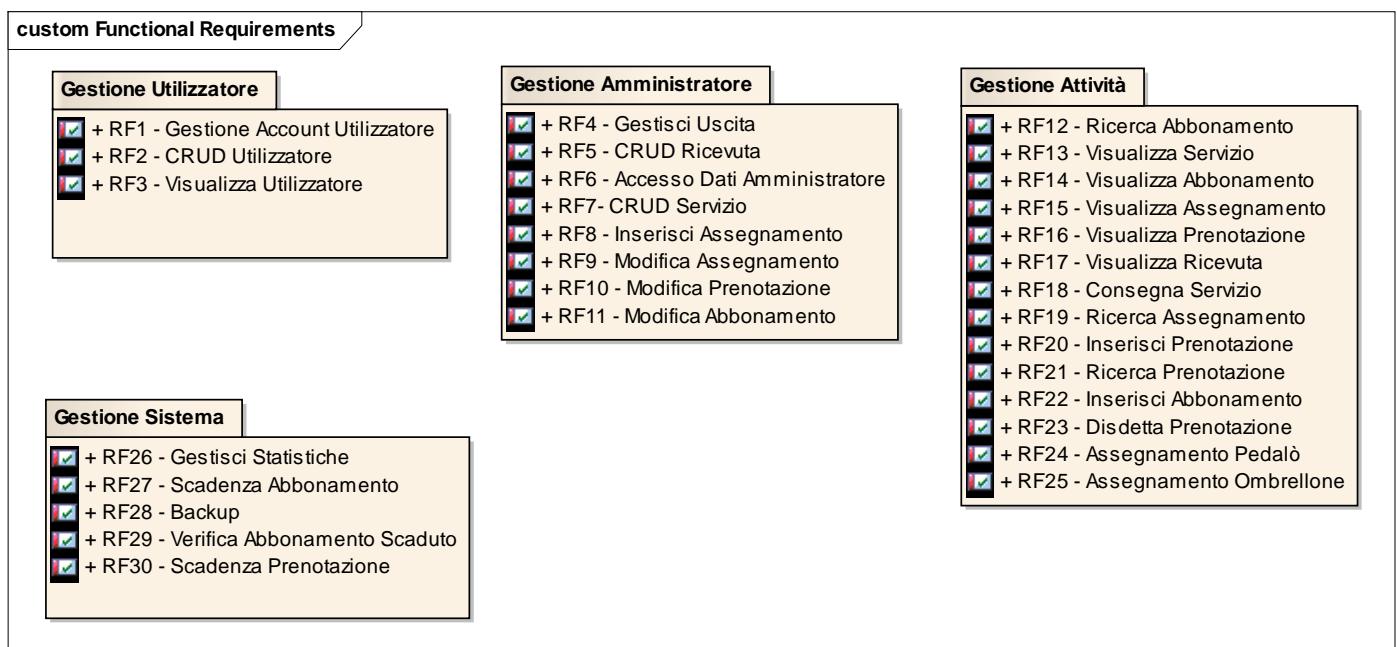
Sinonimi: Nessuno

Omonimi: Nessuno

ANALISI DEI REQUISITI



REQUISITI FUNZIONALI



Gestione Utilizzatore

RF1 - Gestione Account Utilizzatore

Type: **Requirement**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Gestione Utilizzatore

Il sistema dovrà consentire ad un utente generico solo la gestione dei propri dati.

RF2 - CRUD Utilizzatore

Type: **Requirement**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Gestione Utilizzatore

Il sistema dovrà gestire le attività CRUD sugli utilizzatori.

RF3 - Visualizza Utilizzatore

Type: **Requirement**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Gestione Utilizzatore

Il sistema dovrà visualizzare i dati relativi ad un utilizzatore.

Gestione Amministratore

RF4 - Gestisci Uscita

Type: **Requirement**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Gestione Amministratore

Il sistema dovrà gestire l'uscita del cliente dallo stabilimento.

RF5 - CRUD Ricevuta

Type: **Requirement**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Gestione Amministratore

Il sistema dovrà gestire le attività CRUD sulle ricevute.

RF6 - Accesso Dati Amministratore

Type: **Requirement**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Gestione Amministratore

Il sistema dovrà consentire all'amministratore l'accesso, in lettura e scrittura, di tutti i dati presenti nel sistema.

RF7- CRUD Servizio

Type:	Requirement
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Gestione Amministratore

Il sistema dovrà gestire le attività CRUD sui servizi.

RF8 - Inserisci Assegnamento

Type:	Requirement
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Gestione Amministratore

Il sistema dovrà gestire l'inserimento degli assegnamenti dei servizi.

RF9 - Modifica Assegnamento

Type:	Requirement
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Gestione Amministratore

Il sistema dovrà gestire la modifica degli assegnamenti da parte degli amministratori.

RF10 - Modifica Prenotazione

Type:	Requirement
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Gestione Amministratore

Il sistema dovrà gestire la modifica delle prenotazioni da parte dell'amministratore.

RF11 - Modifica Abbonamento

Type:	Requirement
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Gestione Amministratore

Il sistema dovrà gestire la modifica degli abbonamenti da parte dell'amministratore.

Gestione Attività

RF12 - Ricerca Abbonamento

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà ricercare i dati relativi ad un abbonamento.

RF13 - Visualizza Servizio

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà visualizzare i dati relativi ad un servizio.

RF14 - Visualizza Abbonamento

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà visualizzare i dati relativi ad un abbonamento.

RF15 - Visualizza Assegnamento

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà visualizzare i dati relativi ad un assegnamento.

RF16 - Visualizza Prenotazione

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà visualizzare i dati relativi ad una prenotazione.

RF17 - Visualizza Ricevuta

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà visualizzare i dati relativi ad una ricevuta.

RF18 - Consegna Servizio

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà gestire le consegna dei servizi da parte del bagnino ai clienti.

RF19 - Ricerca Assegnamento

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà ricercare i dati relativi ad un assegnamento.

RF20 - Inserisci Prenotazione

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà gestire l'inserimento di una prenotazione da parte dei clienti abbonati.

RF21 - Ricerca Prenotazione

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà ricercare i dati relativi ad una prenotazione.

RF22 - Inserisci Abbonamento

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà gestire l'inserimento di un nuovo abbonamento.

RF23 - Disdetta Prenotazione

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà gestire la disdetta delle prenotazioni, con rimborso del 50% del costo totale del servizio prenotato.

RF24 - Assegnamento Pedalò

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà consentire l'assegnamento del pedalò in base alle ore di noleggio.

RF25 - Assegnamento Ombrellone

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Il sistema dovrà consentire l'assegnamento dell'ombrellone in base alla fila.

Gestione Sistema

RF26 - Gestisci Statistiche

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Il sistema dovrà consentire di effettuare delle statistiche sui dati presenti nel sistema stesso.

RF27 - Scadenza Abbonamento

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Il sistema dovrà gestire la scadenza dell'abbonamento.

RF28 - Backup

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Il sistema dovrà effettuare un backup periodico dei dati.

RF29 - Verifica Abbonamento Scaduto

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Il sistema dovrà verificare se un determinato abbonamento scade nel giorno corrente.

RF30 - Scadenza Prenotazione

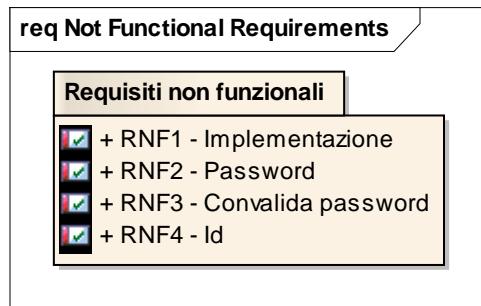
Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Il sistema deve gestire la scadenza delle prenotazioni.

REQUISITI NON FUNZIONALI



RNF1 - Implementazione

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Requisiti non funzionali

Il sistema dovrà essere realizzato in tecnologia Java.

RNF2 - Password

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Requisiti non funzionali

Il sistema dovrà possedere una password di almeno 6 caratteri alfanumerici.

RNF3 - Convalida password

Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Requisiti non funzionali

Il sistema dovrà validare la password entro 3 secondi.

RNF4 - Id

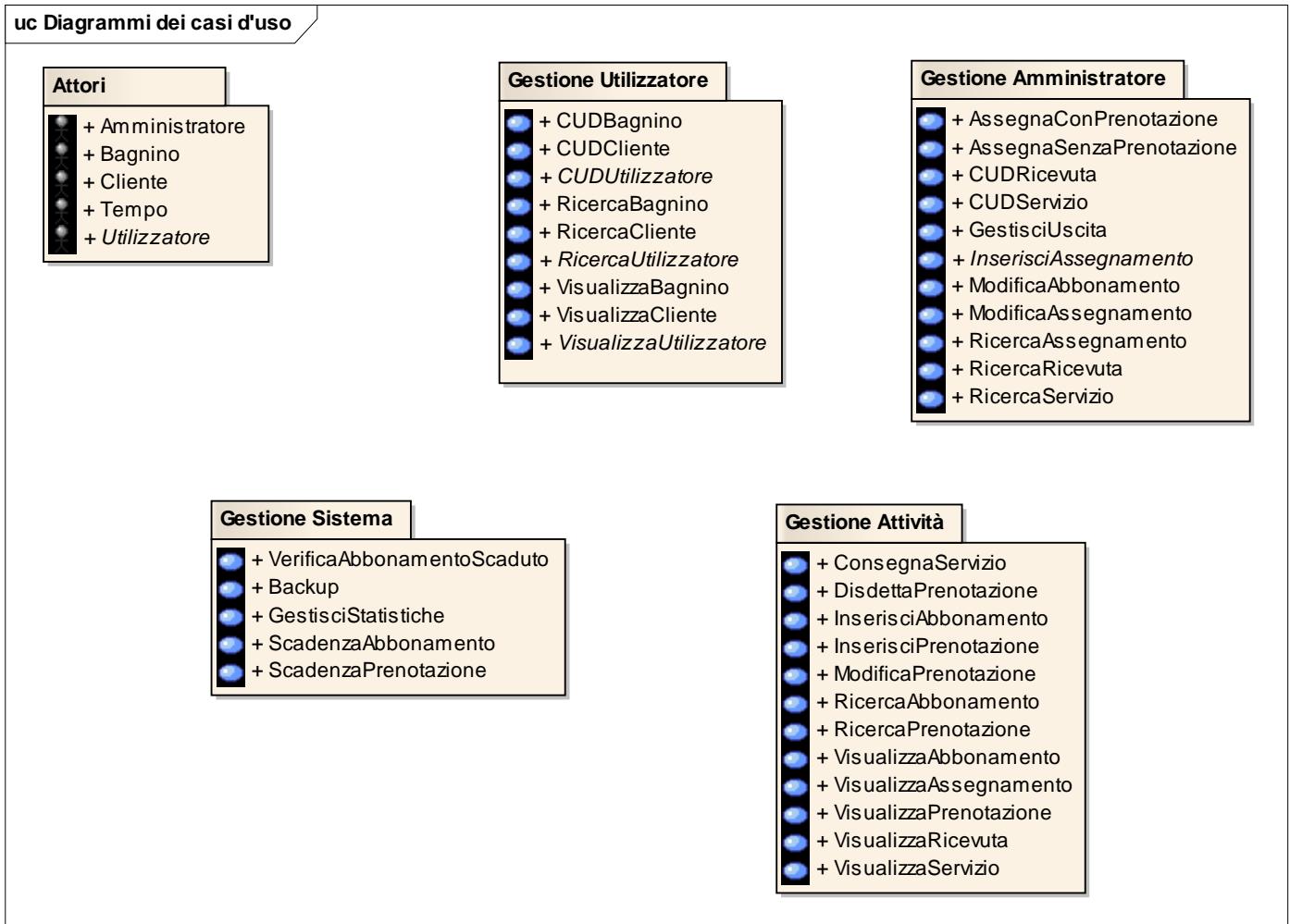
Type: **Requirement**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Requisiti non funzionali

Il sistema non dovrà consentire l'impiego di Id già in uso.

DIAGRAMMI DEI CASI D'USO

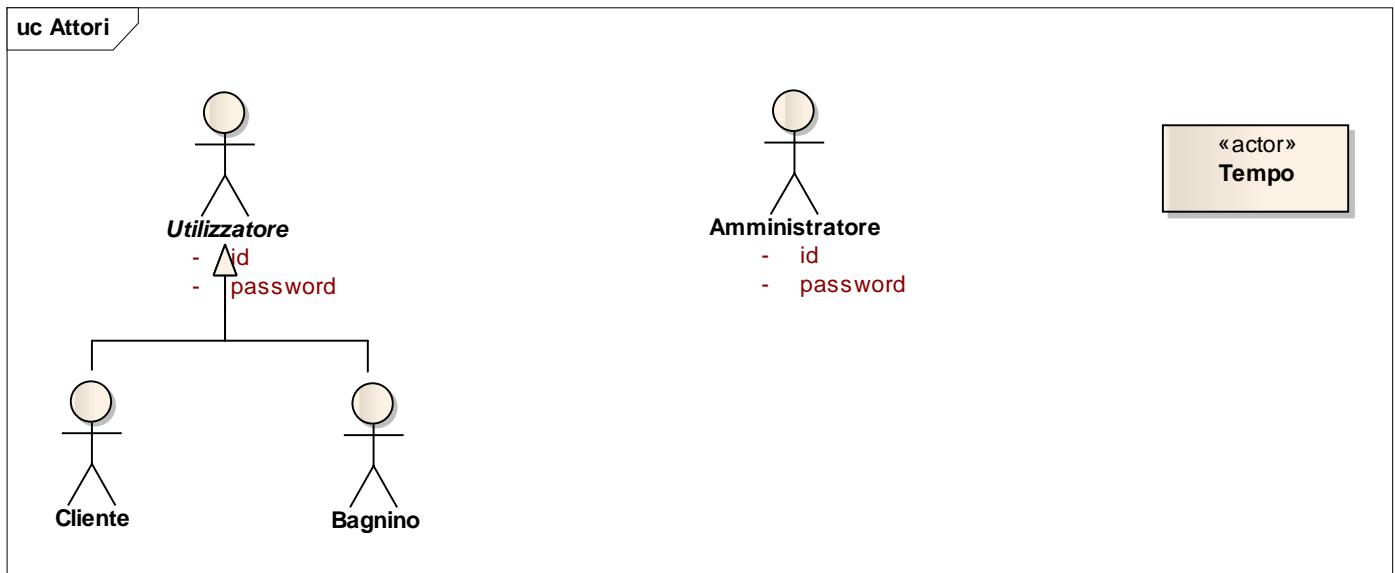


ATTORI

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi dei casi d'uso

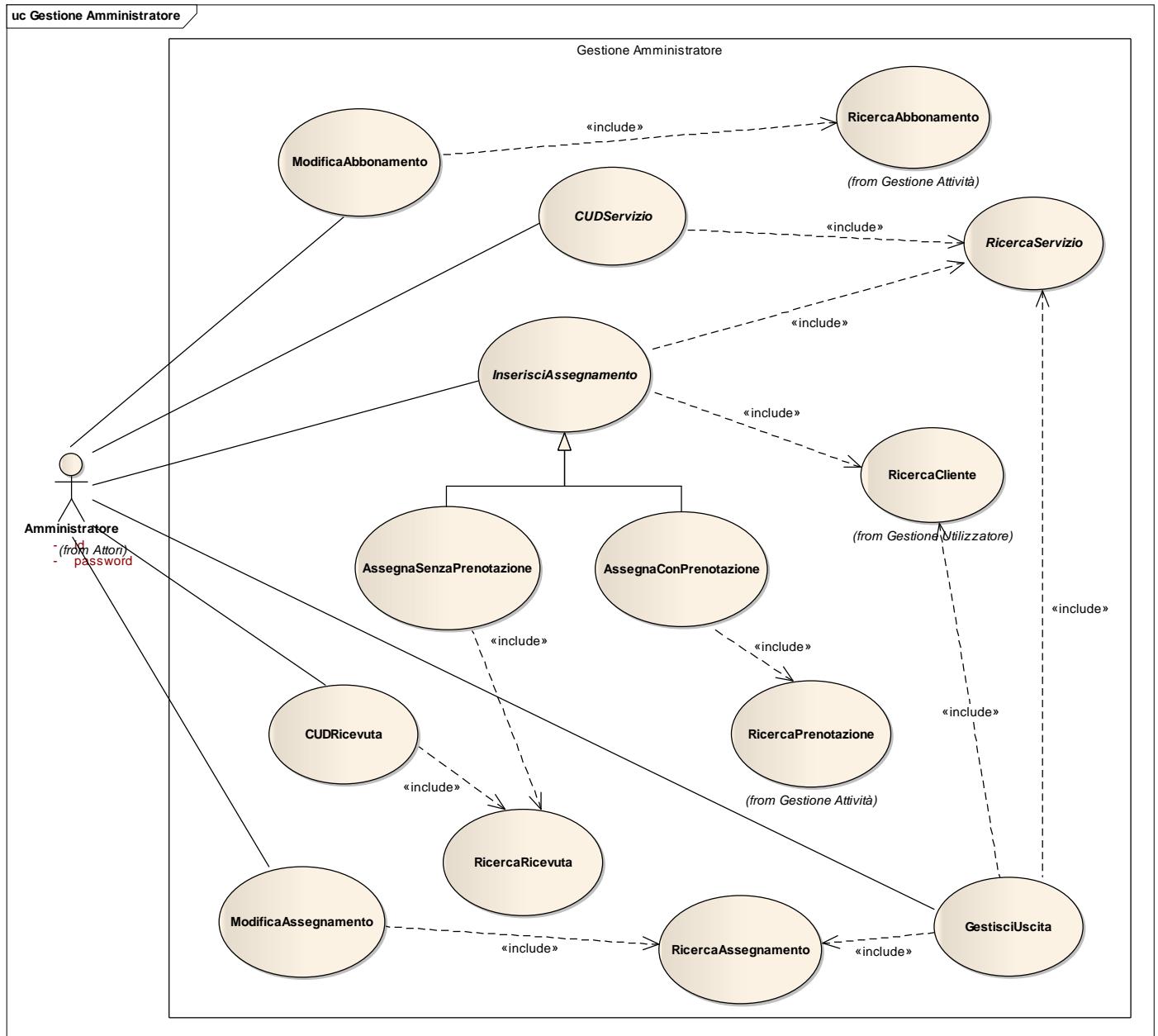


GESTIONE AMMINISTRATORE

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi dei casi d'uso



AssegnaConPrenotazione

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente di inserire nel database un nuovo assegnamento data una prenotazione.

Constraints

- Precondizioni sequenza eventi principale: (*Pre-condition, Status is Approved*)
L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizioni sequenza eventi principale: (*Post-condition, Status is Approved*)
L'assegnamento è stato effettuato.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

- 1.(1.) il caso d'uso inizia quando l'attore primario vuole assegnare uno o più servizi ad un cliente;
- 2.(2.) include (RicercaCliente);
3. *if* il cliente è non abbonato
 - 3.1. il sistema visualizza un messaggio di errore;
4. il sistema preleva la chiave delle prenotazioni associate;
5. *for each* chiave di prenotazione
 - 5.1. include (RicercaPrenotazione);
 - 5.2. *if* la prenotazione è scaduta
 - 5.2.1. il sistema visualizza un opportuno messaggio di errore;
 - 5.3. *else*
 - 5.3.1. (o3.) *for each* servizio prenotato
 - 5.3.1.1. il sistema preleva la chiave del servizio;
 - 5.3.1.2. (3.1.) include (RicercaServizio);
 - 6.(4.) il sistema crea un nuovo assegnamento;
 - 7.(5.) il sistema memorizza i dati inseriti.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

AssegnaSenzaPrenotazione

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente di inserire nel database un nuovo assegnamento senza una precedente prenotazione.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizioni: (*Post-condition, Status is Approved*)
L'assegnamento è stato effettuato.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

- 1.(1.) il caso d'uso inizia quando l'attore primario vuole assegnare uno o più servizi ad un cliente;
- 2.(2.) include(RicercaCliente);
3. il sistema chiede all'attore primario di inserire il servizio da assegnare;
- 4.(3.) *for each* servizio inserito
 - 4.1.(3.1.) include (RicercaServizio);
 - 4.2. il sistema verifica la disponibilità del servizio;
 - 4.3. *if* il servizio non è disponibile o è una cabina
 - 4.3.1. il sistema visualizza un opportuno messaggio di errore;
 - 4.4. *else*
 - 4.4.1. *if* il servizio scelto è un ombrellone
 - 4.4.1.1. *repeat* l'attore primario inserisce il numero della fila
 - 4.4.1.2. *while* non esiste un ombrellone disponibile nella fila richiesta;
 - 4.4.2. *if* il servizio scelto è un pedalò
 - 4.4.2.1. l'attore primario inserisce il numero di ore di noleggio;
 - 4.4.3. il sistema addebita il costo del servizio;
- 5.(4.) il sistema crea un nuovo assegnamento;
- 6.(5.) il sistema memorizza data e ora dell'assegnamento;
7. *for each* ricevuta associata al cliente
 - 7.1. include(RicercaRicevuta);
 8. *if* non esiste una ricevuta relativa allo stesso cliente nello stesso giorno oppure il cliente non risulta abbonato
 - 8.1. il sistema addebita il costo di ingresso allo stabilimento;
 9. il sistema crea una nuova ricevuta;
 10. il sistema memorizza la data e l'ora nella ricevuta;

Scenarios

11. il sistema stampa la ricevuta.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

CUDRicevuta

Type: [UseCase](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente l'inserimento, la modifica o la rimozione di informazioni relative ad una ricevuta.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni: (*Pre-condition, Status is Approved*)

Nessuna

Scenarios

- Sequenza degli eventi principale - [Basic Path](#)

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole effettuare un'operazione CUD sui dati relativi alla ricevuta;
2. *if* l'attore primario vuole inserire una nuova ricevuta
 - 2.1. l'attore primario inserisce i dati relativi alla nuova ricevuta;
 - 2.2. include(RicercaRicevuta);
 - 2.3 *if* la ricevuta è già presente
 - 2.3.1. il sistema visualizza un opportuno messaggio di errore;
 - 2.4. e il sistema memorizza i dati;
3. *else if* l'attore primario vuole aggiornare i dati di una ricevuta
 - 3.1. l'attore primario inserisce la chiave della ricevuta;
 - 3.2. include(RicercaRicevuta);
 - 3.3. *if* la ricevuta ricercata viene trovata
 - 3.3.1. l'attore primario specifica i nuovi dati da inserire;
 - 3.3.2. il sistema procede con l'aggiornamento dei dati;
4. *else if* l'attore primario vuole rimuovere i dati relativi a una ricevuta
 - 4.1. l'attore primario inserisce la chiave della ricevuta;
 - 4.2. include(RicercaRicevuta);
 - 4.3. *if* la ricevuta ricercata viene trovata
 - 4.2.1 il sistema procede con la sua rimozione.

- Sequenza degli eventi alternativa - [Alternate](#)

Notes

Nessuna

CUDServizio

Type: **UseCase** (Abstract)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente l'inserimento, la modifica o la rimozione di informazioni relative a un servizio.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni: (*Post-condition, Status is Approved*)

Nessuna

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole effettuare un'operazione CUD sui dati relativi ad un servizio;
2. *if* l'attore primario vuole inserire un nuovo servizio
 - 2.1. l'attore primario inserisce i dati relativi al nuovo servizio;
 - 2.2. include(RicercaServizio);
 - 2.3. *if* il servizio è già presente
 - 2.3.1. il sistema visualizza un opportuno messaggio di errore;
 - 2.4. *else*
 - 2.4.1. il sistema memorizza i dati;
3. *else if* l'attore primario vuole aggiornare i dati di un servizio
 - 3.1. l'attore primario inserisce la chiave del servizio;
 - 3.2. include(RicercaServizio);
 - 3.3. *if* il servizio ricercato viene trovato
 - 3.3.1. l'attore primario specifica i nuovi dati da inserire;
 - 3.3.2. il sistema procede con l'aggiornamento dei dati;
4. *else if* l'attore primario vuole rimuovere i dati relativi a un servizio
 - 4.1. l'attore primario inserisce la chiave del servizio;
 - 4.2. include(RicercaServizio);
 - 4.3. *if* il servizio ricercato viene trovato
 - 4.2.1 il sistema procede con la sua rimozione.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

GestisciUscita

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso gestisce l'uscita di un cliente dallo stabilimento balneare.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni: (*Pre-condition, Status is Approved*)
 1. Il sistema modifica la disponibilità dei servizi associati al cliente in uscita;
 2. Il sistema setta l'ora di fine assegnamento.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso ha inizio quando l'attore primario vuole terminare l'assegnamento di un servizio relativo ad un cliente;
2. include(RicercaCliente);
3. il sistema preleva la chiave relativa agli assegnamenti associati al cliente;
4. *for each* chiave di assegnamento
 - 4.1. include(RicercaAssegnamento);
 - 4.2. *if* l'assegnamento specificato è scaduto
 - 4.2.1. il sistema visualizza un opportuno messaggio di errore;
 - 4.2. *else*
 - 4.2.1. il sistema sostituisce l'ora di fine assegnamento con l'ora corrente;
 - 4.2.2. il sistema preleva la chiave dei servizi associati all'assegnamento;
 - 4.2.3.. *for each* servizio
 - 4.2.3.1. Include(RicercaServizio);
 - 4.2.3.2. il sistema imposta il servizio come non consegnato;

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

InserisciAssegnamento

Type: **UseCase** (Abstract)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente di inserire nel database un nuovo assegnamento.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
L'assegnamento è stato effettuato oppure si è determinata l'impossibilità di farlo
- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso inizia quando l'attore primario vuole assegnare uno o più servizi ad un cliente;
2. include(RicercaCliente);
3. *for each* servizio inserito
 - 3.1. include (RicercaServizio);
4. il sistema crea un nuovo assegnamento;
5. il sistema memorizza data e ora dell'assegnamento.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

ModificaAbbonamento

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente di modificare un abbonamento.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni: (*Post-condition, Status is Approved*)

L'abbonamento, se esisteva, è stato aggiornato.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso inizia quando l'attore primario vuole modificare i dati di un abbonamento;
2. l'attore primario inserisce la chiave dell'abbonamento;
3. include(RicercaAbbonamento);
4. *if* l'abbonamento è scaduto
 - 4.1 il sistema visualizza un opportuno messaggio di errore;
5. *else*
 - 5.1. l'attore primario fornisce tutte le nuove informazioni relative all'abbonamento;
 - 5.2. il sistema memorizza le nuove informazioni.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna.

ModificaAssegnamento

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente di modificare i dati relativi a un assegnamento precedentemente effettuato.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni: (*Post-condition, Status is Approved*)

L'assegnamento è stato aggiornato.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso inizia quando l'attore primario vuole modificare i dati di un assegnamento precedentemente inserito;
2. l'attore primario inserisce la chiave dell'assegnamento;
3. include(RicercaAssegnamento);
4. l'attore primario fornisce tutte le nuove informazioni relative all'assegnamento;
5. il sistema memorizza le nuove informazioni.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna.

RicercaAssegnamento

Type: [UseCase](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente di ricercare i dati relativi ad un assegnamento.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
La ricerca dell'assegnamento è stata effettuata oppure si è determinata l'impossibilità di farlo
- Precondizioni: (*Pre-condition, Status is Approved*)
Nessuna

Scenarios

- Sequenza degli eventi principale - [Basic Path](#)

Notes

- Il caso d'uso ha inizio quando il sistema deve effettuare la ricerca sui dati relativi ad un assegnamento;
- il sistema preleva la chiave di ricerca;
- il sistema ricerca l'assegnamento che soddisfa i criteri di ricerca specificati;
- if* il sistema non trova nessun assegnamento
 - il sistema visualizza un opportuno messaggio di errore;
- else*
 - il sistema preleva tutte le informazioni relative all'assegnamento.

- Sequenza degli eventi alternativa - [Alternate](#)

Notes

Nessuna

RicercaRicevuta

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente di ricercare i dati relativi ad una ricevuta.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 - Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)
 - La ricerca della ricevuta è stata effettuata oppure si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando il sistema deve effettuare la ricerca sui dati relativi a una ricevuta;
2. il sistema preleva la chiave di ricerca;
3. il sistema ricerca la ricevuta specificata;
4. *if* la ricevuta specificata non è presente
 - 4.1 il sistema visualizza un opportuno messaggio di errore;
5. *else*
 - 5.1. il sistema preleva tutte le informazioni relative alla ricevuta.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

RicercaServizio

Type: UseCase (Abstract)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Amministratore

Questo caso d'uso consente di ricercare i dati relativi ad un servizio.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
La ricerca del servizio è stata effettuata oppure si è determinata l'impossibilità di farlo.
- Precondizioni: (*Pre-condition, Status is Approved*)
Nessuna

Scenarios

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

- Sequenza degli eventi principale - Basic Path

Notes

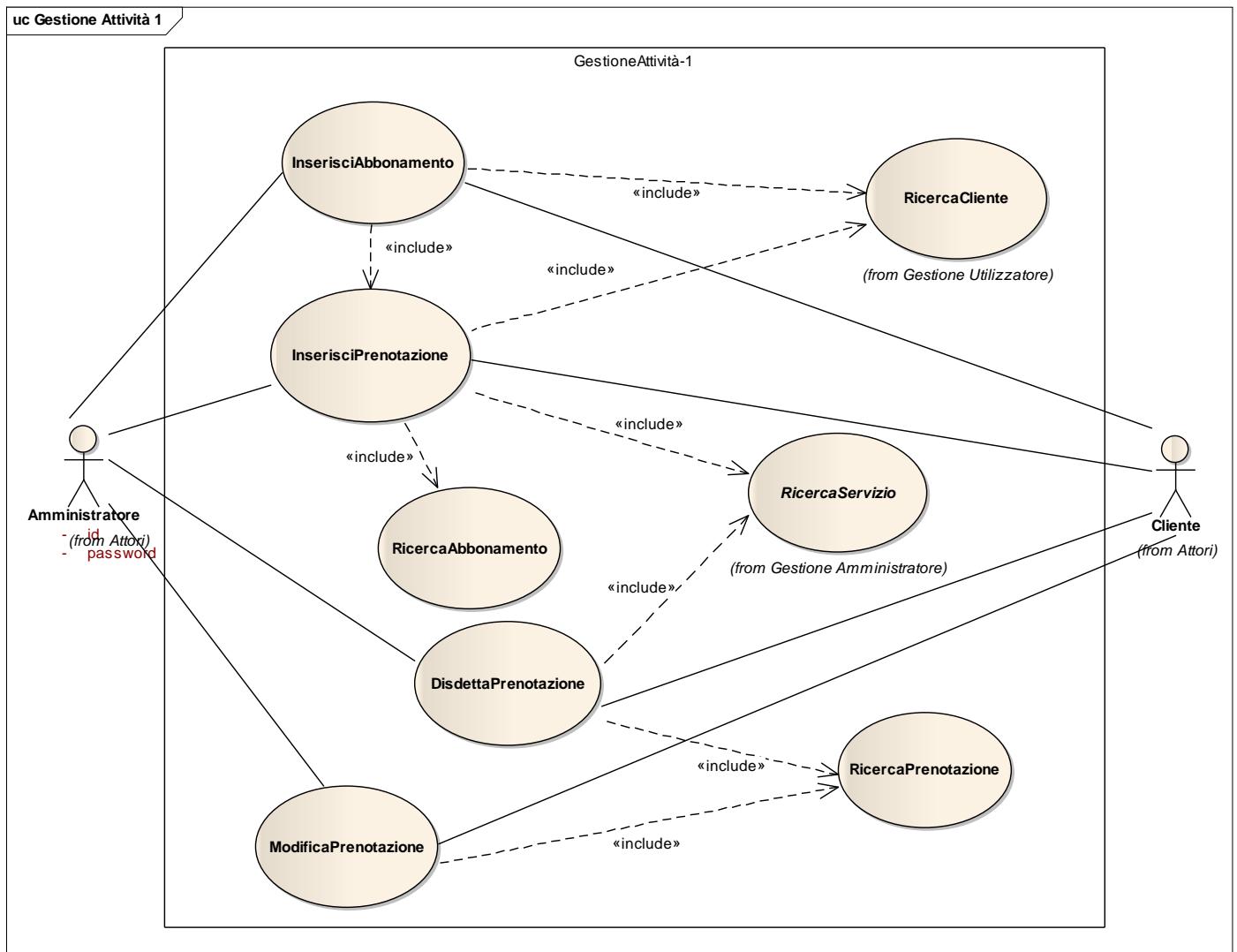
1. Il caso d'uso ha inizio quando il sistema deve effettuare la ricerca sui dati relativi a un servizio;
2. il sistema preleva la chiave di ricerca;
3. il sistema ricerca il servizio specificato;
4. *if* il servizio specificato non è presente
 - 4.1. il sistema visualizza un opportuno messaggio di errore;
5. *else*
 - 5.1. il sistema preleva tutte le informazioni relative al servizio;

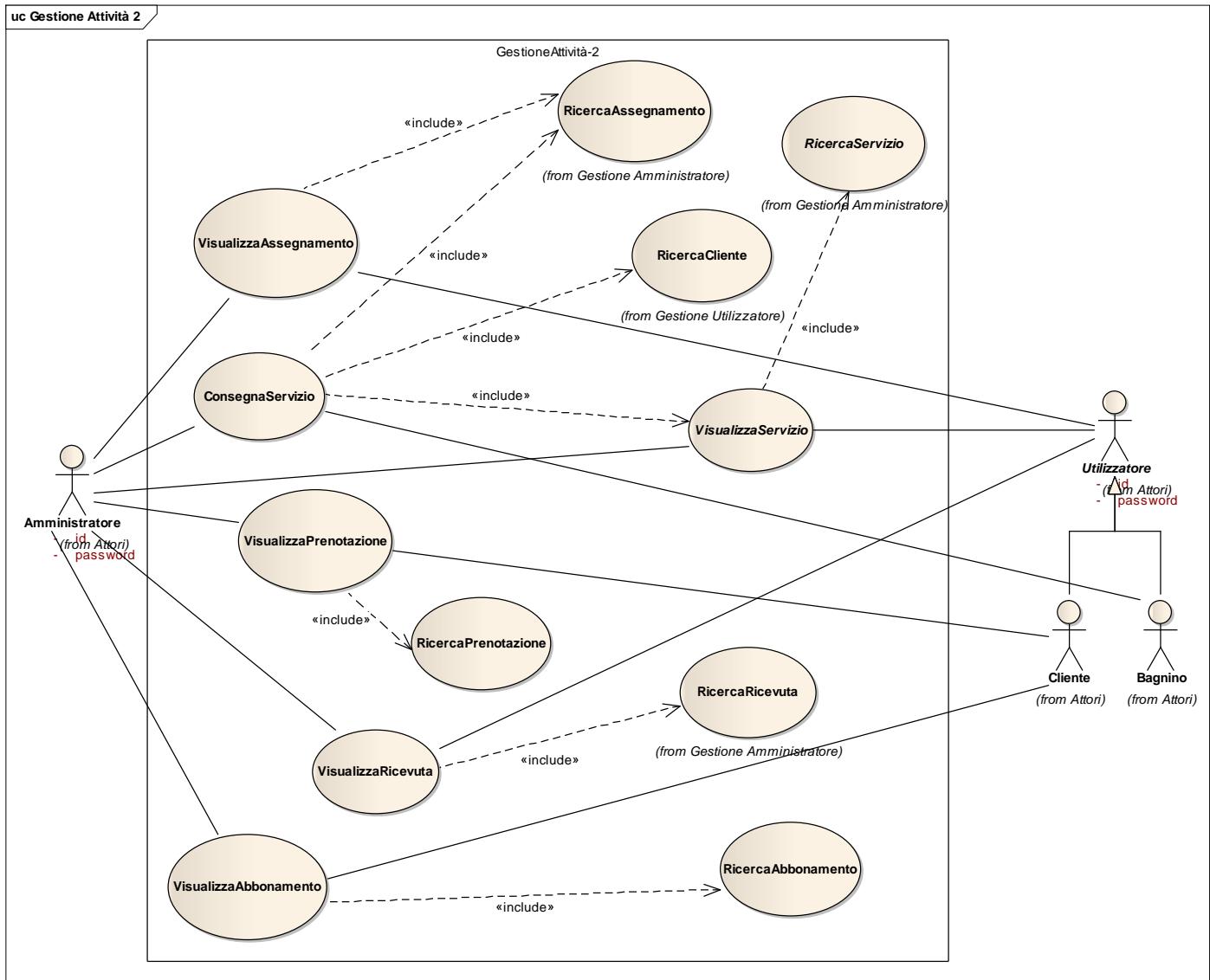
GESTIONE ATTIVITÀ

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi dei casi d'uso





ConsegnaServizio

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di segnalare se il servizio è stato consegnato al cliente.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizioni: (*Post-condition, Status is Approved*)

Il sistema ha modificato i dati relativi al servizio consegnato al cliente oppure si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso ha inizio quando l'attore primario vuole consegnare un servizio assegnato ad un cliente;
2. include (RicercaCliente);
3. il sistema preleva la chiave degli assegnamenti associati al cliente;
4. *for each* assegnamento
 - 4.1. include (RicercaAssegnamento);
 - 4.2. *if* l'assegnamento è scaduto
 - 4.2.1. il sistema visualizza un opportuno messaggio di errore;
 - 4.3. *else*
 - 4.3.1. il sistema preleva la chiave dei servizi associati all'assegnamento;
 - 4.3.2. *for each* servizio
 - 4.3.2.1. include (VisualizzaServizio);
 - 4.3.2.2. *if* il servizio è stato già consegnato
 - 4.3.2.2.1. il sistema visualizza un opportuno messaggio di errore;
 - 4.3.2.3. *else*
 - 4.3.2.3.1. l'attore primario setta come consegnato il servizio;

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

DisdettaPrenotazione

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di gestire la disdetta di una prenotazione.

Constraints

- Precondizioni sequenza eventi principale: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni sequenza eventi principale: (*Post-condition, Status is Approved*)
 1. La prenotazione è stata annullata;
 2. E' stato rimborsato il 50% della prenotazione
- Precondizioni sequenza eventi alternativa: (*Pre-condition, Status is Approved*)

La data di inizio della prenotazione è minore della data corrente più sette giorni
- Postcondizioni sequenza eventi alternativa: (*Post-condition, Status is Approved*)

Nessuna

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso inizia quando l'attore primario vuole disdire una prenotazione precedentemente inserita;
2. l'attore primario inserisce il codice della prenotazione da disdire;
3. include(RicercaPrenotazione);
4. il sistema visualizza i servizi associati alla prenotazione;
5. l'attore primario inserisce la chiave dei servizi che intende disdire
6. *for each* servizio inserito
 - 6.1. include(RicercaServizio);
 - 6.2. il sistema dissdice il servizio prenotato;
 - 6.3. il sistema accredita al cliente un importo pari alla metà del costo del servizio disdetto;
7. il sistema stampa la ricevuta.

- Disdettalimpossibile - Alternate

Notes

1. la sequenza degli eventi alternativa inizia dopo il passo 3 della sequenza degli eventi principale
2. il sistema visualizza un opportuno messaggio di errore.

InserisciAbbonamento

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso gestisce la compilazione dell'abbonamento da parte del cliente.

Constraints

- Precondizione sequenza eventi principale: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizione sequenza eventi principale: (*Pre-condition, Status is Approved*)

L'abbonamento è stato creato oppure si è determinata l'impossibilità di farlo

Scenarios

- Sequenza eventi principale - Basic Path

Notes

1. il caso d'uso ha inizio quando l'attore primario vuole creare un abbonamento;
2. include(RicercaCliente);
3. if il cliente possiede già un abbonamento valido
 - 3.1. il sistema visualizza un opportuno messaggio di errore;
4. il sistema crea l'abbonamento;
5. l'attore primario inserisce il tipo e la data di inizio dell'abbonamento;
6. il sistema modifica lo stato del cliente come abbonato;
7. include (InserisciPrenotazione);
8. il sistema stampa l'abbonamento.

InserisciPrenotazione

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di effettuare una nuova prenotazione presso lo stabilimento balneare

Constraints

- Precondizione sequenza eventi principale: (*Pre-condition, Status is Approved*)
L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizione sequenza eventi principale: (*Post-condition, Status is Approved*)
La prenotazione è stata effettuata o si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso inizia quando l'attore primario vuole prenotare uno o più servizi ad un cliente;
2. include(RicercaCliente);
3. *if* il cliente non è abbonato
 - 3.1 il sistema visualizza un messaggio di errore;
4. il sistema chiede all'attore primario di inserire il servizio;
5. *for each* servizio inserito
 - 5.1. include (RicercaServizio);
 - 5.2. il sistema verifica la disponibilità del servizio;
 - 5.3. *if* il servizio non è disponibile o è un pedalò
 - 5.3.1. il sistema visualizza un opportuno messaggio di errore;
 - 5.4. *if* il servizio scelto è un ombrellone
 - 5.4.1. *repeat* l'attore primario inserisce il numero della fila
 - 5.4.2. *while* non esiste un ombrellone disponibile nella fila richiesta;
 - 5.5. il sistema addebita il costo del servizio sulla ricevuta;
6. il sistema crea una nuova prenotazione;
7. l'attore primario inserisce la data di inizio prenotazione;
8. include(RicercaAbbonamento);
9. il sistema preleva le informazioni dell'abbonamento associato al cliente;
10. il sistema imposta la data di fine prenotazione uguale alla data di fine abbonamento;
11. il sistema memorizza i dati nella prenotazione;
12. il sistema crea una nuova ricevuta;
13. il sistema memorizza la data e l'ora nella ricevuta;
17. il sistema stampa la ricevuta.

Scenarios

- Sequenza eventi alternativa - Alternate

Notes

Nessuna

ModificaPrenotazione

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di modificare i dati relativi a una prenotazione precedentemente effettuata.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni: (*Post-condition, Status is Approved*)

La prenotazione è stata modificata

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

- 1.il caso d'uso inizia quando l'attore primario vuole modificare i dati di una prenotazione precedentemente inserita;
2. include(RicercaPrenotazione);
3. *if* la prenotazione è scaduta

3.1 il sistema visualizza un opportuno messaggio di errore;

4. *else*

- 4.1. l'attore primario fornisce tutte le nuove informazioni relative alla prenotazione;
- 4.2. il sistema memorizza le nuove informazioni.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

RicercaAbbonamento

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente la ricerca di informazioni relative ad un abbonamento

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 - Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)
 - La ricerca dell'abbonamento è stata effettuata oppure si è determinata l'impossibilità di farlo

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole effettuare la ricerca sui dati relativi ad un abbonamento;
2. l'attore primario specifica i criteri di ricerca;
3. il sistema ricerca gli abbonamenti che soddisfano i criteri di ricerca specificati;
4. *if* il sistema non trova nessun abbonamento
 - 4.1 il sistema visualizza un opportuno messaggio di errore;
5. *else*
 - 5.1. il sistema preleva tutte le informazioni relative all'abbonamento.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

RicercaPrenotazione

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente la ricerca di informazioni relative ad una prenotazione.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 - Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)
 - La ricerca della prenotazione è stata effettuata oppure si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando il sistema invoca il caso d'uso;
2. il sistema preleva il codice o la data di scadenza;
3. il sistema ricerca le prenotazioni che soddisfano i criteri di ricerca specificati;
4. *if* il sistema non trova nessuna prenotazione
 - 4.1 il sistema visualizza un opportuno messaggio di errore;
5. *else*
 - 5.1. *forEach* prenotazione
 - 5.1.1. il sistema preleva tutte le informazioni relative.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VisualizzaAbbonamento

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di visualizzare le informazioni relative ad un abbonamento.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizioni: (*Post-condition, Status is Approved*)
La visualizzazione dell'abbonamento è stata effettuata oppure si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi ad un abbonamento;
2. include(RicercaAbbonamento);
3. il sistema visualizza i dati relativi all'abbonamento.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VisualizzaAssegnamento

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di visualizzare le informazioni relative ad un assegnamento.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
La visualizzazione dell'abbonamento è stata effettuata oppure si è determinata l'impossibilità di farlo
- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi ad un assegnamento;
2. include(RicercaAssegnamento);
3. il sistema visualizza i dati relativi all'assegnamento.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VisualizzaPrenotazione

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di visualizzare i dati relativi ad una prenotazione.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizioni: (*Post-condition, Status is Approved*)

La visualizzazione della prenotazione è stata effettuata oppure si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi ad una prenotazione;
2. include(RicercaPrenotazione);
3. *for each* prenotazione il sistema visualizza i dati relativi.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VisualizzaRicevuta

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di visualizzare le informazioni relative a una ricevuta.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
La visualizzazione della ricevuta è stata effettuata oppure si è determinata l'impossibilità di farlo.
- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi a una ricevuta;
2. include(RicercaRicevuta);
3. Il sistema visualizza i dati relativi alla ricevuta.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VisualizzaServizio

Type: **UseCase** (Abstract)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Attività

Questo caso d'uso consente di visualizzare i dati relativi ad un servizio.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)
La visualizzazione del servizio è stata effettuata oppure si è determinata l'impossibilità di farlo

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi a un servizio;
2. l'attore primario inserisce la chiave di ricerca;
2. include(RicercaServizio);
3. il sistema visualizza i dati relativi.

- Sequenza degli eventi alternativa - Alternate

Notes

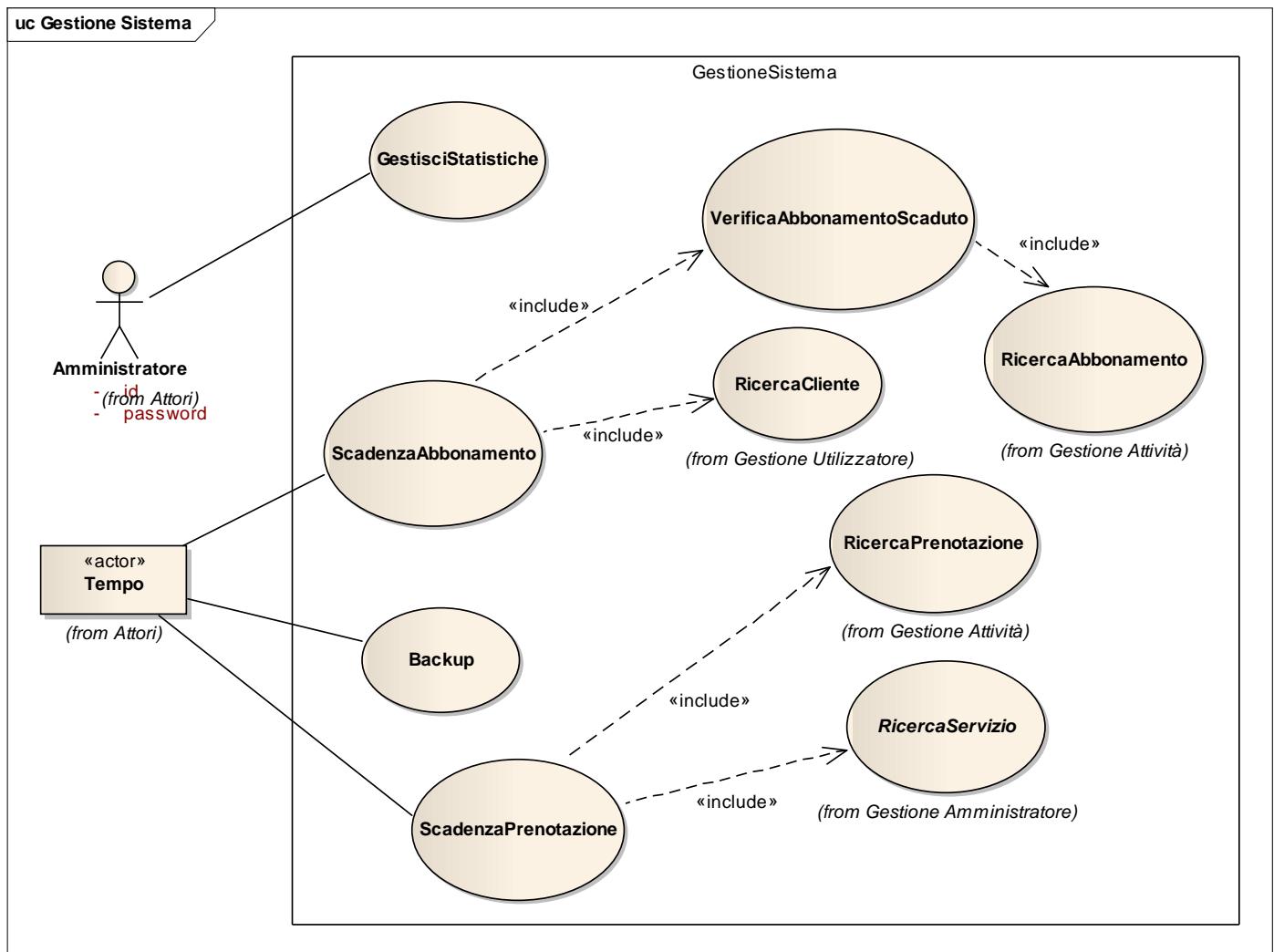
Nessuna

GESTIONE SISTEMA

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi dei casi d'uso



Backup

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
 - I dati sono stati copiati.
- Precondizioni: (*Pre-condition, Status is Approved*)
 - 1. Nessuna

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso inizia ogni giorno alle ore 23.30;
2. il sistema preleva i dati degli utilizzatori;
3. il sistema copia sul disco i dati prelevati;
4. il sistema preleva i dati dei servizi;
5. il sistema copia sul disco i dati prelevati;
6. il sistema preleva i dati degli abbonamenti;
7. il sistema copia su disco i dati prelevati;
8. il sistema preleva i dati delle prenotazioni;
9. il sistema copia sul disco i dati prelevati;
10. il sistema preleva i dati degli assegnamenti;
11. il sistema copia sul disco i dati prelevati;
12. il sistema preleva i dati delle ricevute;
13. il sistema copia sul disco i dati prelevati;

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

GestisciStatistiche

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
Il sistema ha visualizzato le statistiche richieste.
- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account con appropriati diritti d'accesso.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso ha inizio quando l'attore primario richiede statistiche al sistema;
2. *if* l'attore primario richiede statistiche sugli utilizzatori
 - 2.1 il sistema preleva i dati sugli utilizzatori;
 - 2.2 il sistema elabora opportunamente i dati prelevati;
 - 2.3 il sistema visualizza i dati elaborati;
3. *if* l'attore primario richiede statistiche sui servizi
 - 3.1 il sistema preleva i dati sui servizi;
 - 3.2 il sistema elabora opportunamente i dati prelevati;
 - 3.3 il sistema visualizza i dati elaborati;
4. *if* l'attore primario richiede statistiche sugli abbonamenti
 - 4.1 il sistema preleva i dati sugli abbonamenti;
 - 4.2 il sistema elabora opportunamente i dati prelevati;
 - 4.3 il sistema visualizza i dati elaborati;
5. *if* l'attore primario richiede statistiche sulle ricevute
 - 5.1 il sistema preleva i dati sulle ricevute;
 - 5.2 il sistema elabora opportunamente i dati prelevati;
 - 5.3 il sistema visualizza i dati elaborati;
6. *if* l'attore primario richiede statistiche sulle prenotazioni
 - 6.1 il sistema preleva i dati sulle prenotazioni;
 - 6.2 il sistema elabora opportunamente i dati prelevati;
 - 6.3 il sistema visualizza i dati elaborati;
7. *if* l'attore primario richiede statistiche sugli assegnamenti
 - 7.1 il sistema preleva i dati sugli assegnamenti;
 - 7.2 il sistema elabora opportunamente i dati prelevati;
 - 7.3 il sistema visualizza i dati elaborati

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

ScadenzaAbbonamento

Type: [UseCase](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Questo caso d'uso consente di gestire la scadenza di un abbonamento.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 - Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)
 - La scadenza dell'abbonamento è stata gestita.

Scenarios

- Sequenza degli eventi principale - [Basic Path](#)

Notes

1. il caso d'uso inizia ogni giorno alle ore 23.00;
2. *for each* Abbonamento
 - 2.1. include(VerificaAbbonamentoScaduto);
 - 2.2. il sistema preleva il codice del cliente associato all'abbonamento;
 - 2.3. include(RicercaCliente);
 - 2.4. il sistema modifica i dati del cliente titolare dell'abbonamento da abbonato a non abbonato.

- Sequenza degli eventi alternativa - [Alternate](#)

Notes

Nessuna

ScadenzaPrenotazione

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Questo caso d'uso consente di gestire la scadenza di una prenotazione.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 - Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)
 - La scadenza della prenotazione è stata gestita.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. il caso d'uso inizia ogni giorno alle ore 22.30;
2. il sistema inserisce come criterio di ricerca la data di scadenza pari alla data corrente;
3. include(RicercaPrenotazione);
4. il sistema preleva la chiave delle prenotazioni ricercate;
5. *for each* prenotazione
 - 5.2. *for each* servizio associato
 - 5.2.1. include(RicercaServizio);
 - 5.2.2. *if* il servizio è disdetto
 - 5.2.2.1. il sistema modifica il servizio come non disdetto.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VerificaAbbonamentoScaduto

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Sistema

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 - Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)
 - Il sistema visualizza gli abbonamenti scaduti in quel giorno o si è verificata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando viene invocato dal sistema;
2. l'attore primario inserisce la chiave dell'abbonamento interessato;
3. include RicercaAbbonamento;
4. il sistema preleva le informazioni relative all'abbonamento;
5. *if* la data di scadenza coincide con la data corrente
 - 5.1. il sistema preleva i dati relativi all'abbonamento;
 - 5.2. *else* il sistema visualizza un opportuno messaggio di errore.

- Sequenza degli eventi alternativa - Alternate

Notes

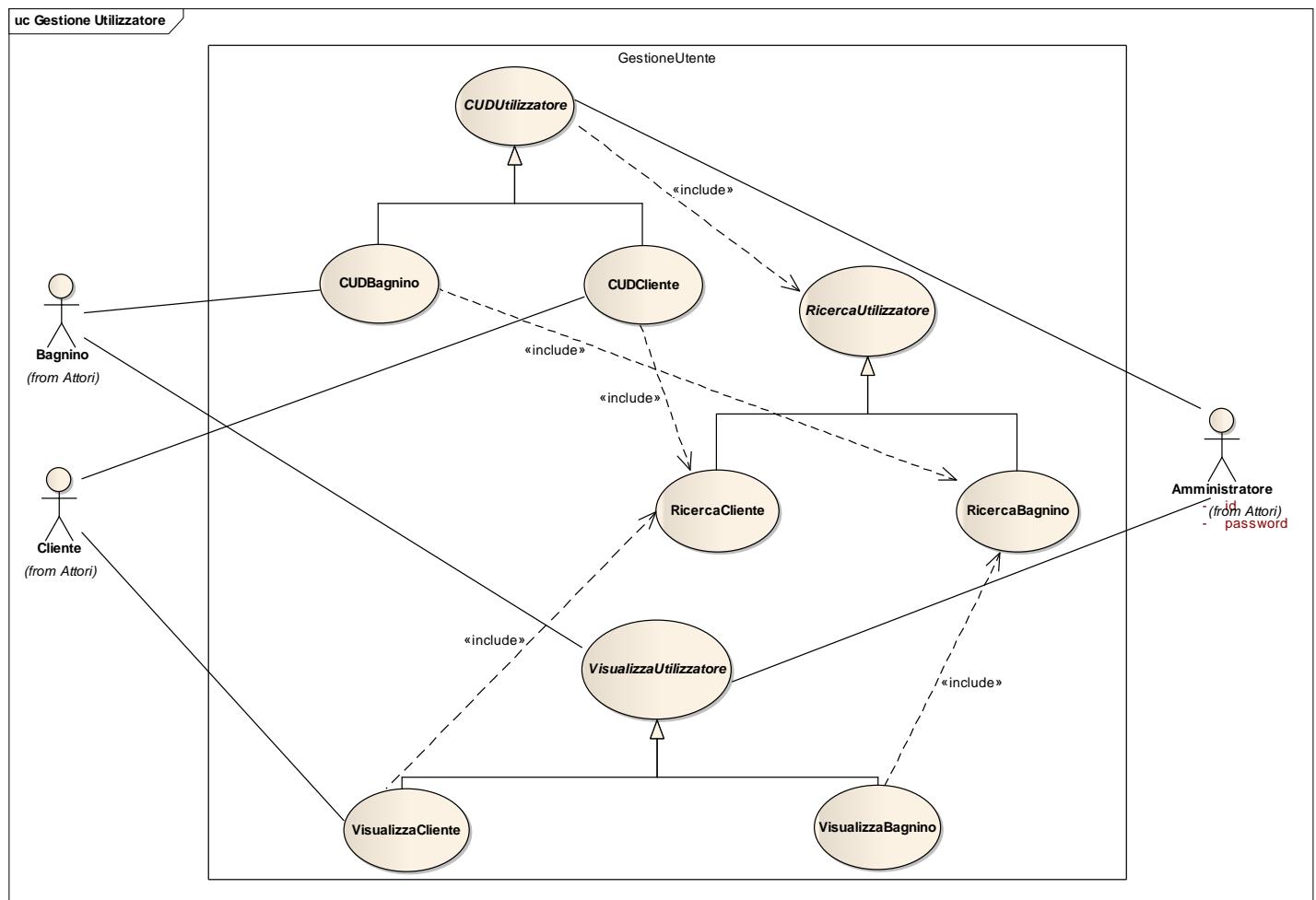
Nessuna

GESTIONE UTILIZZATORE

Type: Package

Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi dei casi d'uso



CUDBagnino

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente l'inserimento, la modifica o la rimozione di informazioni relative a un bagnino.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni: (*Post-condition, Status is Approved*)

Nessuna

Scenarios

- Sequenza eventi principale - Basic Path

Notes

- 1.(o1.) Il caso d'uso ha inizio quando l'attore primario vuole effettuare un'operazione CUD sui dati relativi al bagnino;
2. *if* l'attore primario vuole inserire un nuovo bagnino
 - 2.1.(o2.1.) l'attore primario inserisce i dati relativi al nuovo bagnino;
 - 2.2.(o2.2.) include(RicercaBagnino);
 - 2.3.(2.3.) *if* il bagnino è già presente
 - 2.3.1.(2.3.1.) il sistema visualizza un opportuno messaggio di errore;
 - 2.4.(2.4.) *else*
 - 2.4.1.(2.4.2.) il sistema memorizza i dati;
- 3.(o3.) *else if* l'attore primario vuole aggiornare i dati di un bagnino
 - 3.1.(o3.4.1.1.) include(RicercaBagnino);
 - 3.2.(o3.4.1.2.) *if* il cliente ricercato viene trovato
 - 3.2.1.(3.4.1.3.) l'attore primario specifica i nuovi dati da inserire;
 - 3.2.2.(3.4.1.4.) il sistema procede con l'aggiornamento dei dati;
- 4.(o4.) *else if* l'attore primario vuole rimuovere i dati relativi ad un bagnino
 - 4.1.(o4.1) include (RicercaBagnino);
 - 4.2.(o4.2.) *if* il bagnino ricercato viene trovato
 - 4.2.1.(4.2.1.) il sistema procede con la sua rimozione.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

CUDCliente

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente l'inserimento, la modifica o la rimozione di informazioni relative a un cliente.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
 - Nessuna
- Precondizioni: (*Pre-condition, Status is Approved*)
 - 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

- 1.(o1.) Il caso d'uso ha inizio quando l'attore primario vuole effettuare un'operazione CUD sui dati relativi al cliente;
- 2.(o2.2.) include(RicercaCliente);
 - 2.1.(2.3.) *if* il cliente è già presente
 - 2.1.1.(2.3.1.) il sistema visualizza un opportuno messaggio di errore;
 - 2.2.(2.4.) *else*
 - 2.2.1.(2.4.2.) il sistema memorizza i dati;
- 3.(o3.) *else if* l'attore primario vuole aggiornare i dati di un cliente;
 - 3.1.(o3.4.1.1.) include(RicercaCliente);
 - 3.2.(o3.4.1.2.) *if* il cliente ricercato viene trovato
 - 3.2.1.(3.4.1.3.) l'attore primario specifica i nuovi dati da inserire;
 - 3.2.2.(2.4.1.4.) il sistema procede con l'aggiornamento dei dati;
- 4.(o4.) *else if* l'attore primario vuole rimuovere i dati relativi ad un cliente
 - 4.1.(o4.1.) include (RicercaCliente);
 - 4.2.(o4.2.) *if* il cliente ricercato viene trovato
 - 4.2.1.(4.2.1.) il sistema procede con la sua rimozione.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

CUDUtilizzatore

Type: **UseCase** (Abstract)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente l'inserimento, la modifica o la rimozione di informazioni relative ad un utilizzatore.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizioni: (*Pre-condition, Status is Approved*)

Nessuna

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole effettuare un'operazione CUD sui dati relativi all'utilizzatore;
2. *if* l'attore primario vuole inserire un nuovo utilizzatore
 - 2.1. l'attore primario inserisce i dati relativi al nuovo utilizzatore;
 - 2.2. include(RicercaUtilizzatore);
 - 2.3. *if* l'utilizzatore è già presente
 - 2.3.1. il sistema visualizza un opportuno messaggio di errore;
 - 2.4. *else*
 - 2.4.1. il sistema memorizza i dati;
3. *else if* l'attore primario vuole aggiornare i dati di un utilizzatore
 - 3.4.1.1. include(RicercaUtilizzatore);
 - 3.4.1.2. *if* l'utilizzatore ricercato viene trovato
 - 3.4.1.3. l'attore primario specifica i nuovi dati da inserire;
 - 3.4.1.4. il sistema procede con l'aggiornamento dei dati;
4. *else if* l'attore primario vuole rimuovere i dati relativi a un utilizzatore
 - 4.1. include (RicercaUtilizzatore);
 - 4.2. *if* l'utilizzatore ricercato viene trovato
 - 4.2.1. il sistema procede con la sua rimozione

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

RicercaBagnino

Type: [UseCase](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente di ricercare i dati relativi ad un bagnino.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
La ricerca del bagnino è stata effettuata oppure si è determinata l'impossibilità di farlo
- Precondizioni: (*Pre-condition, Status is Approved*)
Nessuna

Scenarios

- Sequenza degli eventi principale - [Basic Path](#)

Notes

- 1.(o1.) Il caso d'uso ha inizio quando l'attore primario vuole effettuare la ricerca sui dati relativi ad un bagnino;
- 2.(2.) l'attore primario specifica i criteri di ricerca, ovvero codice o codice fiscale o nome e cognome;
- 3.(3.) il sistema ricerca i bagnini che soddisfano i criteri di ricerca specificati;
- 4.(4.)for each bagnino trovato
 - 4.1 (4.1.)il sistema preleva tutte le informazioni relative al bagnino;
5. (5.)if non esiste un bagnino che soddisfi i criteri specificati
 - 5.1.(5.1.) il sistema visualizza un opportuno messaggio di errore.

- Sequenza degli eventi alternativa - [Alternate](#)

Notes

Nessuna

RicercaCliente

Type: [UseCase](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente di ricercare i dati relativi ad un cliente

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 - Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)
 - La ricerca del cliente è stata effettuata oppure si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - [Basic Path](#)

Notes

- 1.(o1.) Il caso d'uso ha inizio quando l'attore primario vuole effettuare la ricerca sui dati relativi ad un cliente;
- 2.(2.) l'attore primario specifica i criteri di ricerca, ovvero codice o codice fiscale o nome e cognome;
- 3.(3.) il sistema ricerca i clienti che soddisfano i criteri di ricerca specificati;
- 4.(4.) for each cliente trovato
 - 4.1 (4.1.)il sistema preleva tutte le informazioni relative al cliente;
 - 5.(5.)if non esiste un cliente che soddisfi i criteri specificati
 - 5.1.(5.1.) il sistema visualizza un opportuno messaggio di errore
 - 5.2. if l'attore primario vuole inserire un nuovo cliente
 - 5.2.1. l'attore primario inserisce i dati relativi al nuovo cliente;
 - 5.2.2. il sistema crea un nuovo cliente;
 - 5.2.3. il sistema memorizza i dati.

- Sequenza degli eventi alternativa - [Alternate](#)

Notes

Nessuna

RicercaUtilizzatore

Type: **UseCase** (Abstract)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente di ricercare i dati relativi ad un utilizzatore.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)

La ricerca dell'utilizzatore è stata effettuata oppure si è determinata l'impossibilità di farlo.

- Precondizioni: (*Pre-condition, Status is Approved*)

Nessuna

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

- Il caso d'uso ha inizio quando l'attore primario vuole effettuare la ricerca sui dati relativi ad un utilizzatore;
- l'attore primario specifica i criteri di ricerca, ovvero codice o codice fiscale o nome e cognome;
- il sistema ricerca gli utilizzatori che soddisfano i criteri di ricerca specificati;
- for each utilizzatore trovato
 - il sistema preleva tutte le informazioni relative all'utilizzatore;
- if non esiste un utilizzatore che soddisfi i criteri specificati
 - il sistema visualizza un opportuno messaggio di errore.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VisualizzaBagnino

Type: **UseCase**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente di visualizzare le informazioni relative ad un bagnino.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizioni: (*Post-condition, Status is Approved*)

La visualizzazione dei dati relativi al bagnino è stata effettuata oppure si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

- 1.(o1.) Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi ad un bagnino;
- 2.(o2.) include(RicercaBagnino);
- 3.(o3.) *for each* bagnino trovato il sistema visualizza i dati relativi.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VisualizzaCliente

Type: UseCase

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente di visualizzare le informazioni relative ad un cliente.

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.
- Postcondizioni: (*Post-condition, Status is Approved*)
La visualizzazione del cliente è stata effettuata oppure si è determinata l'impossibilità di farlo.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

- 1.(o1.) Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi ad un cliente;
- 2.(o2.) include(RicercaCliente);
- 3.(o3.) *for each* cliente trovato il sistema visualizza i dati relativi.

- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

VisualizzaUtilizzatore

Type: **UseCase** (Abstract)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione Utilizzatore

Questo caso d'uso consente la ricerca di informazioni relative ad un utilizzatore.

Constraints

- Postcondizioni: (*Post-condition, Status is Approved*)
La visualizzazione dei dati relativi all'utilizzatore è stata effettuata oppure si è determinata l'impossibilità di farlo.
- Precondizioni: (*Post-condition, Status is Approved*)
 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.

Scenarios

- Sequenza degli eventi principale - Basic Path

Notes

1. Il caso d'uso ha inizio quando l'attore primario vuole visualizzare i dati relativi ad un utilizzatore;
2. include(RicercaUtilizzatore);
3. *for each* utilizzatore trovato il sistema visualizza i dati relativi.

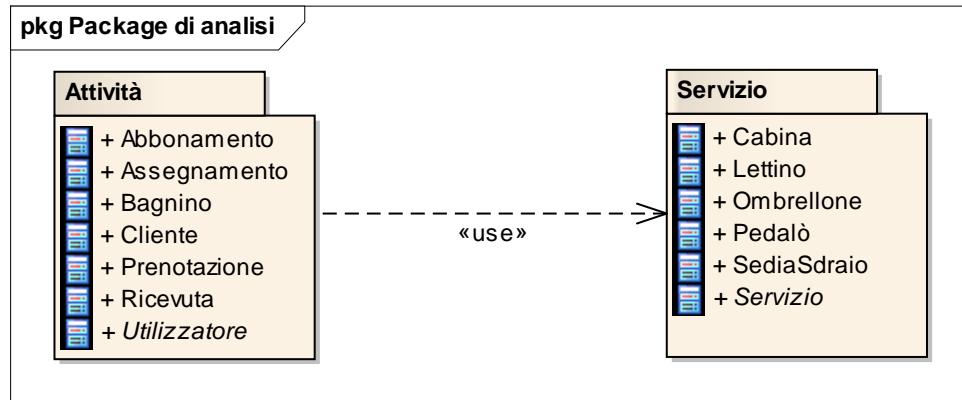
- Sequenza degli eventi alternativa - Alternate

Notes

Nessuna

MATRICE DI MAPPING

PACKAGE DI ANALISI

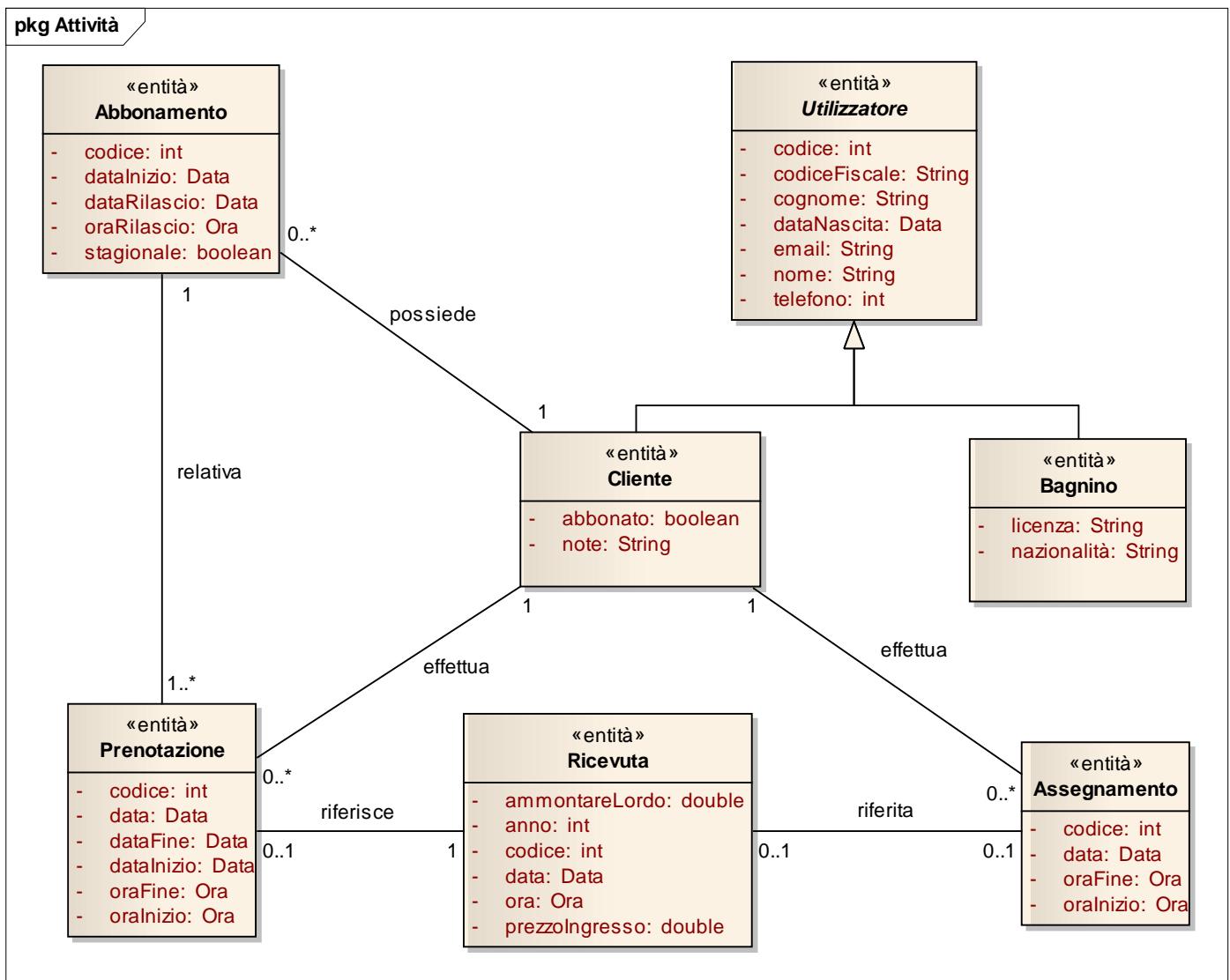


ATTIVITÀ

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Package di analisi



Abbonamento

Type: Class

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> relativa Unspecified	Public Prenotazione	Public Abbonamento	
<u>Association</u> possiede Unspecified	Public Cliente	Public Abbonamento	

Attributes

Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
dataInizio Data Private		<i>Default:</i>
dataRilascio Data Private		<i>Default:</i>
oraRilascio Ora Private		<i>Default:</i>
stagionale boolean Private		<i>Default:</i>

Assegnamento

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> effettua Unspecified	Public Cliente	Public Assegnamento	
<u>Association</u> riferita Unspecified	Public Ricevuta	Public Assegnamento	
<u>Association</u> associato Unspecified	Public Servizio	Public Assegnamento	

Attributes

Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
data Data Private		<i>Default:</i>
oraFine Ora Private		<i>Default:</i>
oraInizio Ora Private		<i>Default:</i>

Bagnino

Type: **Class** **Utilizzatore**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> consegna Unspecified	Public Bagnino	Public Servizio	
<u>Generalization</u> Source -> Destination	Public Bagnino	Public Utilizzatore	

Attributes

Attribute	Notes	Constraints and tags
licenza String Private		<i>Default:</i>
nazionalità String Private		<i>Default:</i>

Cliente

Type: Class Utilizzatore

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Cliente	Public Utilizzatore	
Association possiede Unspecified	Public Cliente	Public Abbonamento	
Association effettua Unspecified	Public Cliente	Public Assegnamento	
Association effettua Unspecified	Public Cliente	Public Prenotazione	

Attributes

Attribute	Notes	Constraints and tags
abbonato boolean Private		<i>Default:</i>
note String Private		<i>Default:</i>

Prenotazione

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> associato Unspecified	Public Servizio	Public Prenotazione	
<u>Association</u> relativa Unspecified	Public Prenotazione	Public Abbonamento	
<u>Association</u> effettua Unspecified	Public Cliente	Public Prenotazione	
<u>Association</u> riferisce Unspecified	Public Prenotazione	Public Ricevuta	

Attributes

Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
data Data Private		<i>Default:</i>
dataFine Data Private		<i>Default:</i>
dataInizio Data Private		<i>Default:</i>

oraFine Ora		<i>Default:</i>
Private		
oraInizio Ora		<i>Default:</i>
Private		

Ricevuta

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> riferita Unspecified	Public Ricevuta	Public Assegnamento	
<u>Association</u> riferisce Unspecified	Public Prenotazione	Public Ricevuta	

Attributes

Attribute	Notes	Constraints and tags
ammontareLordo double Private		<i>Default:</i>
anno int Private		<i>Default:</i>
codice int Private		<i>Default:</i>
data Data Private		<i>Default:</i>
ora Ora Private		<i>Default:</i>
prezzolngresso double Private		<i>Default:</i>

Utilizzatore

Type: **Class** (Abstract)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Cliente	Public Utilizzatore	
Generalization Source -> Destination	Public Bagnino	Public Utilizzatore	

Attributes

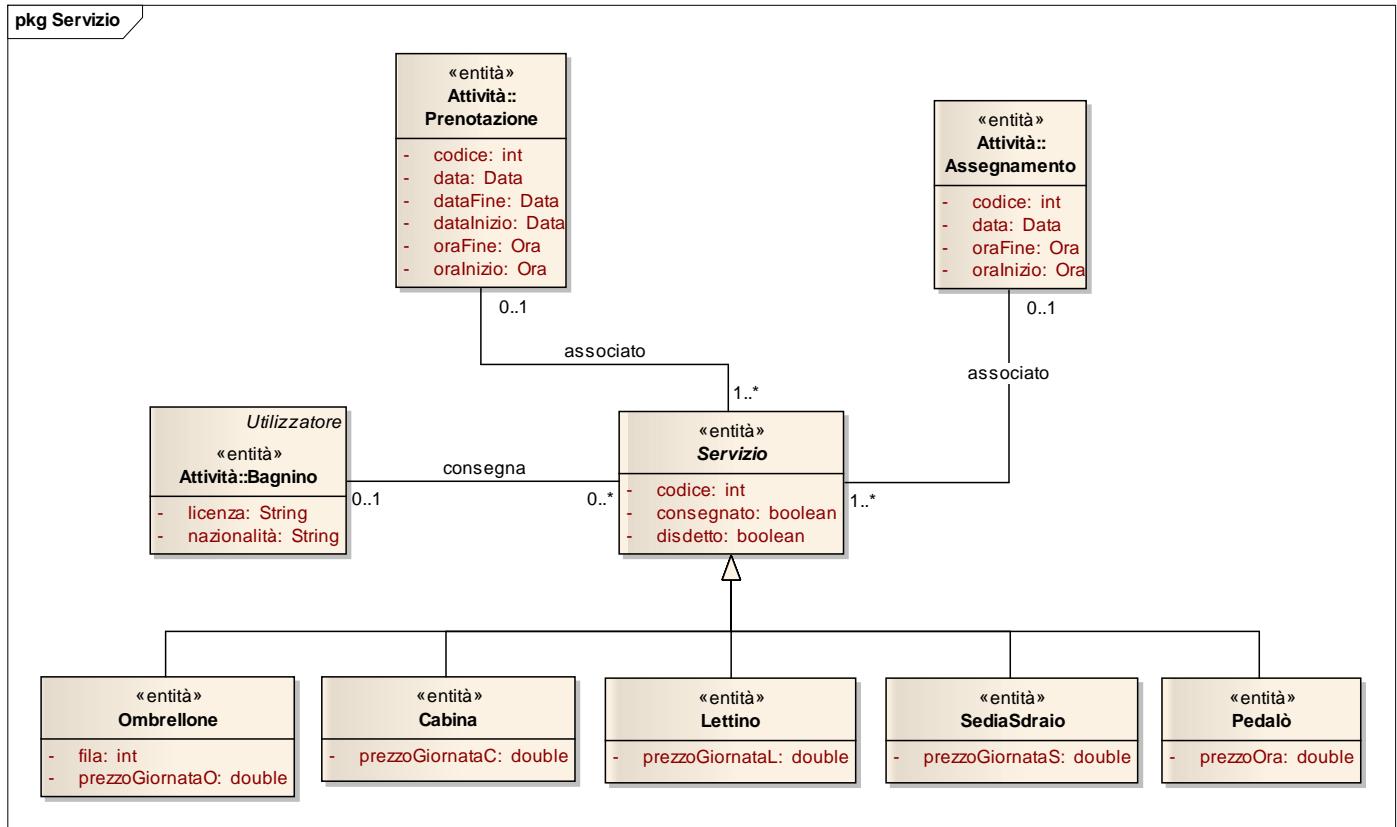
Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
codiceFiscale String Private		<i>Default:</i>
cognome String Private		<i>Default:</i>
dataNascita Data Private		<i>Default:</i>
email String Private		<i>Default:</i>
nome String Private		<i>Default:</i>
telefono int Private		<i>Default:</i>

SERVIZIO

Type: [Package](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Package di analisi



Cabina

Type: [Class Servizio](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Servizio

Connections

Connector	Source	Target	Notes
<u>Generalization</u> Source -> Destination	Public Cabina	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
prezzoGiornataC double Private Static		<i>Default:</i>

Lettino

Type: [Class Servizio](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Servizio

Connections

Connector	Source	Target	Notes
<u>Generalization</u> Source -> Destination	Public Lettino	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
prezzoGiornataL double Private Static		<i>Default:</i>

Ombrellone

Type: **Class Servizio**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Servizio

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Ombrellone	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
fila int Private		<i>Default:</i>
prezzoGiornataO double Private Static		<i>Default:</i>

Pedalò

Type: **Class Servizio**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Servizio

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Pedalò	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
prezzoOra double Private Static		<i>Default:</i>

SediaSdraio

Type: **Class Servizio**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Servizio

Connections

Connector	Source	Target	Notes
<u>Generalization</u> Source -> Destination	Public SediaSdraio	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
prezzoGiornataS double Private Static		<i>Default:</i>

Servizio

Type: **Class** (Abstract)
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Servizio

Connections

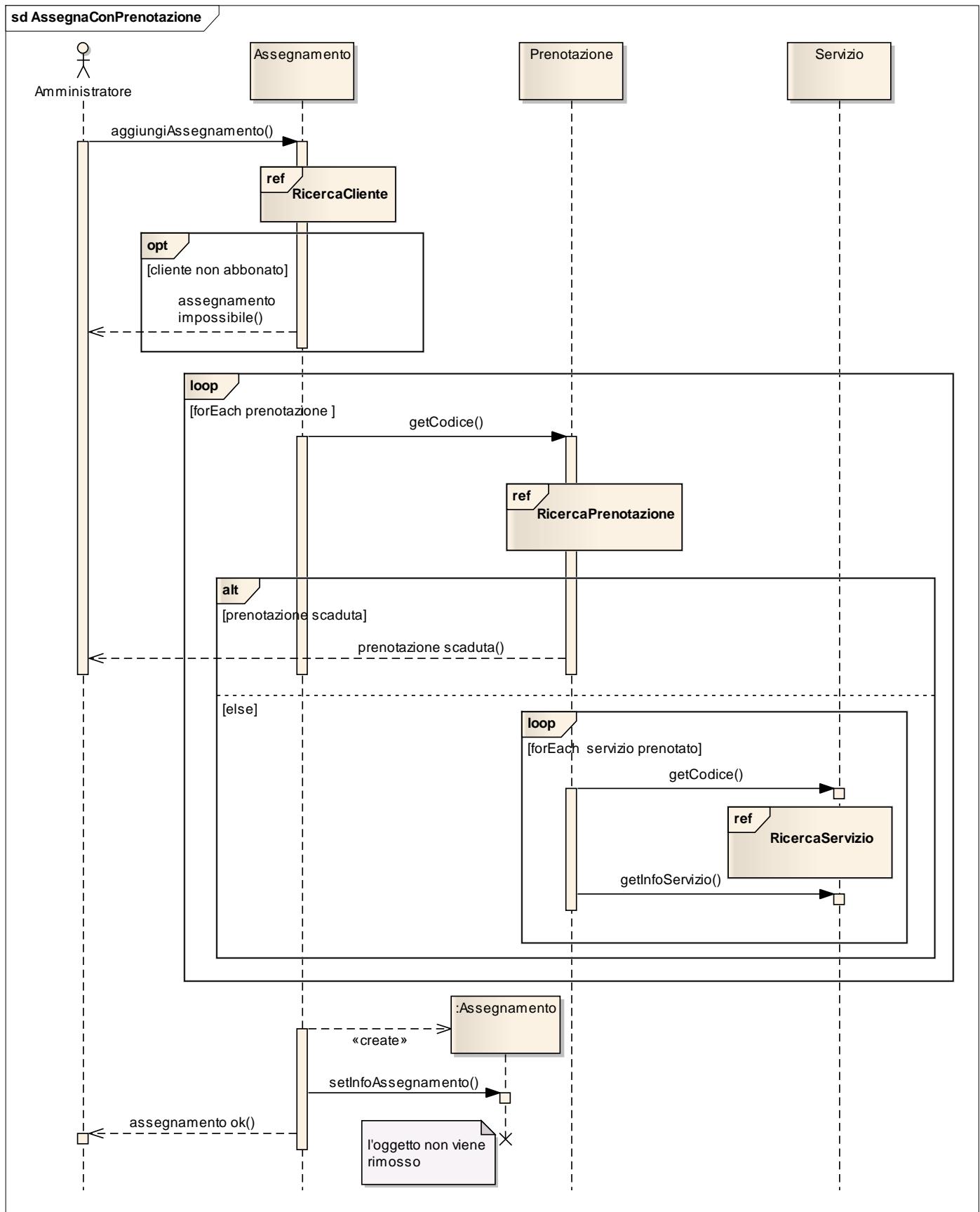
Connector	Source	Target	Notes
Association associato Unspecified	Public Servizio	Public Prenotazione	
Association consegna Unspecified	Public Bagnino	Public Servizio	
Generalization Source -> Destination	Public Pedalò	Public Servizio	
Association associato Unspecified	Public Servizio	Public Assegnamento	
Generalization Source -> Destination	Public SediaSdraio	Public Servizio	
Generalization Source -> Destination	Public Lettino	Public Servizio	
Generalization Source -> Destination	Public Cabina	Public Servizio	
Generalization Source -> Destination	Public Ombrellone	Public Servizio	

Attributes

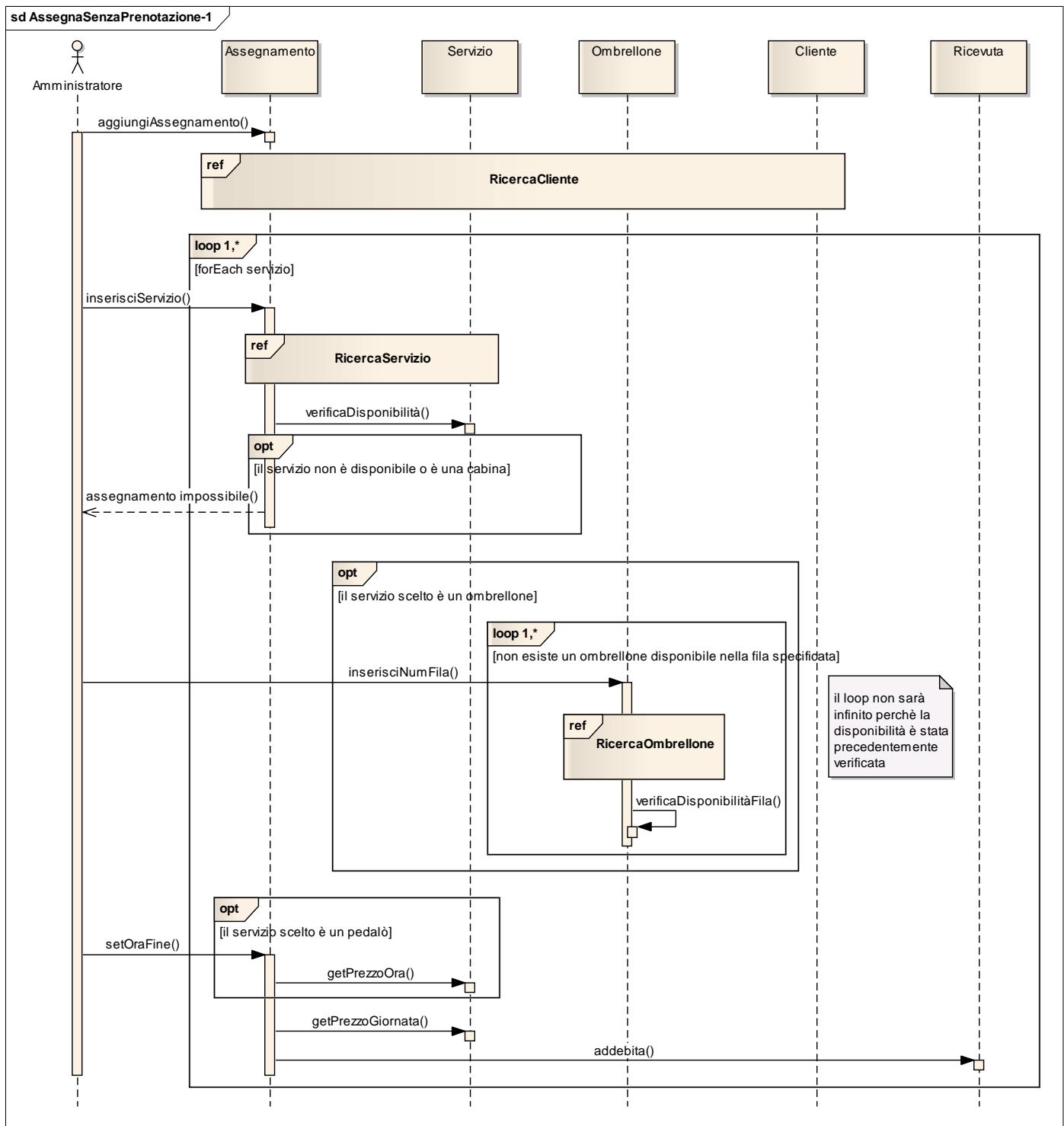
Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
consegnato boolean Private		<i>Default:</i>
disdetto boolean Private		<i>Default:</i>

DIAGRAMMI DI SEQUENZA

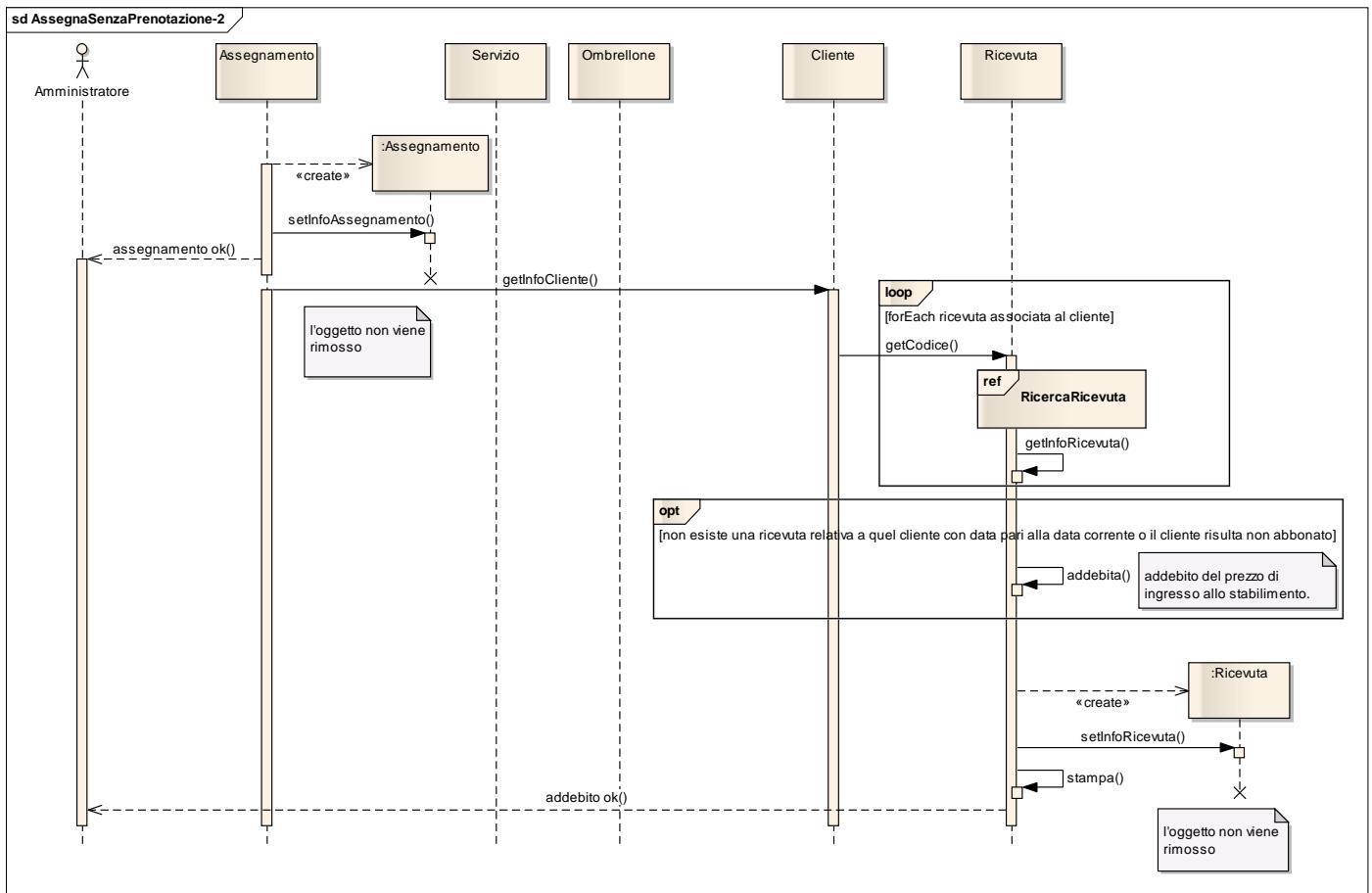
ASSEGNA CON PRENOTAZIONE



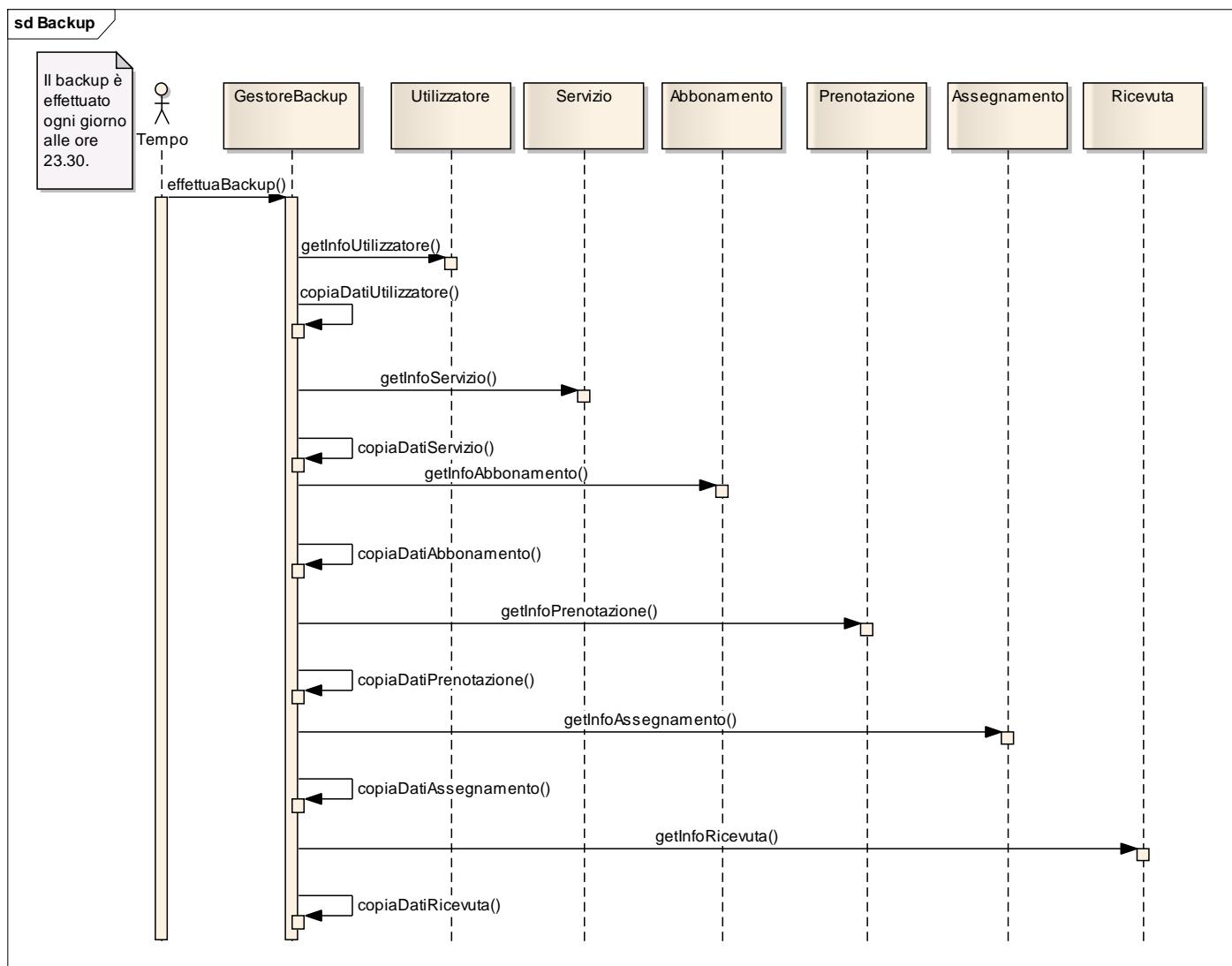
ASSEGNA SENZA PRENOTAZIONE



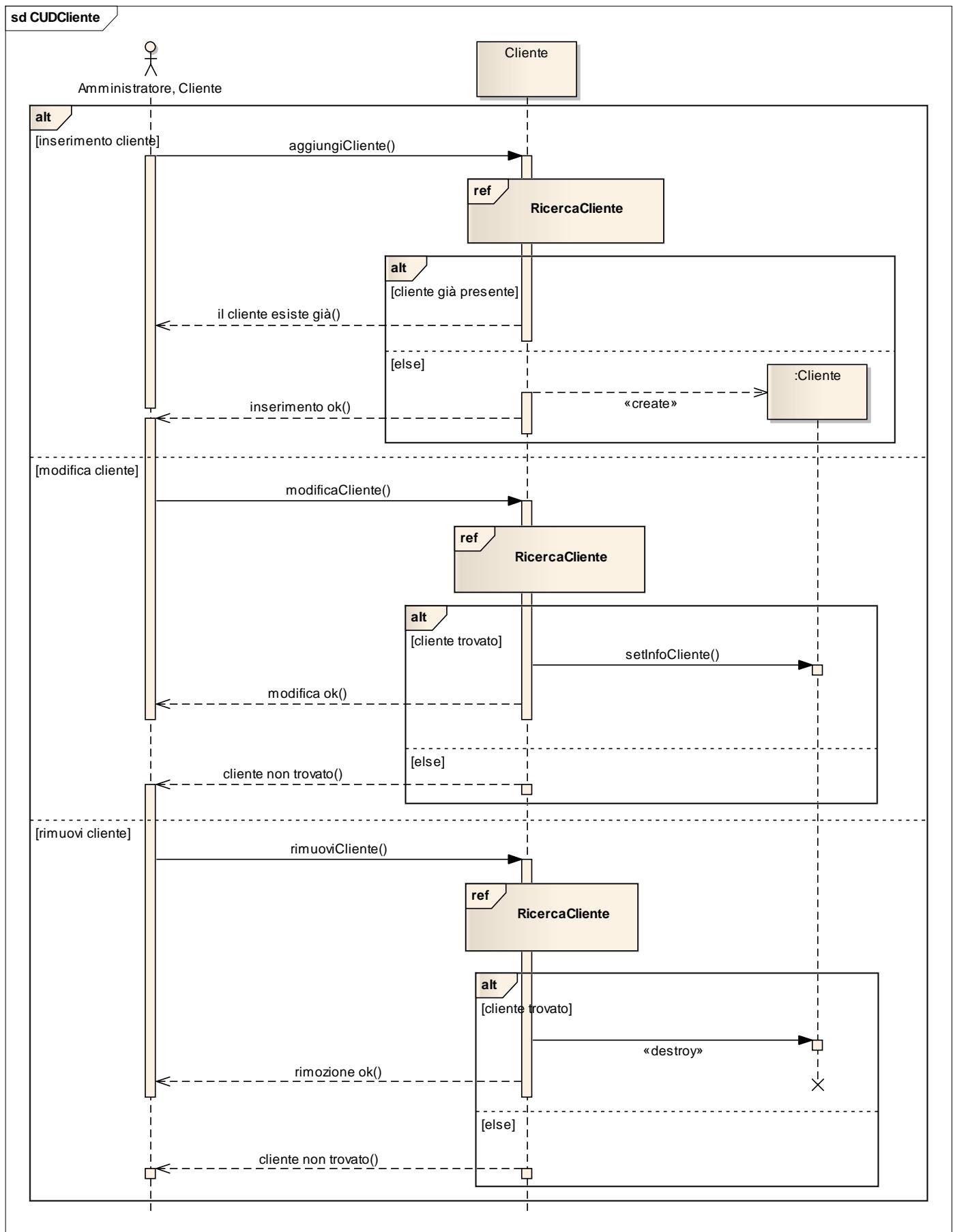
Continua alla pagina successiva



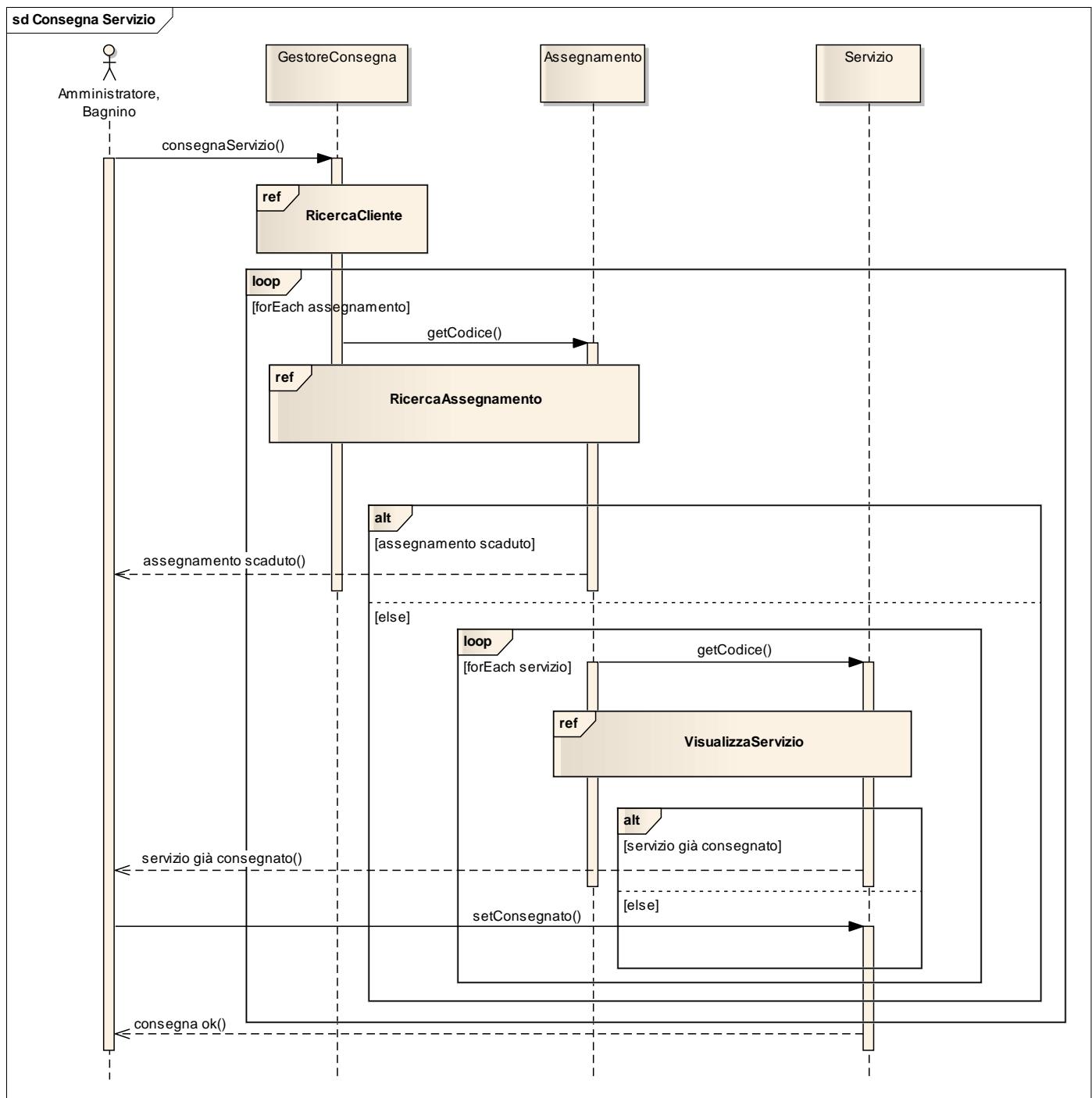
BACKUP



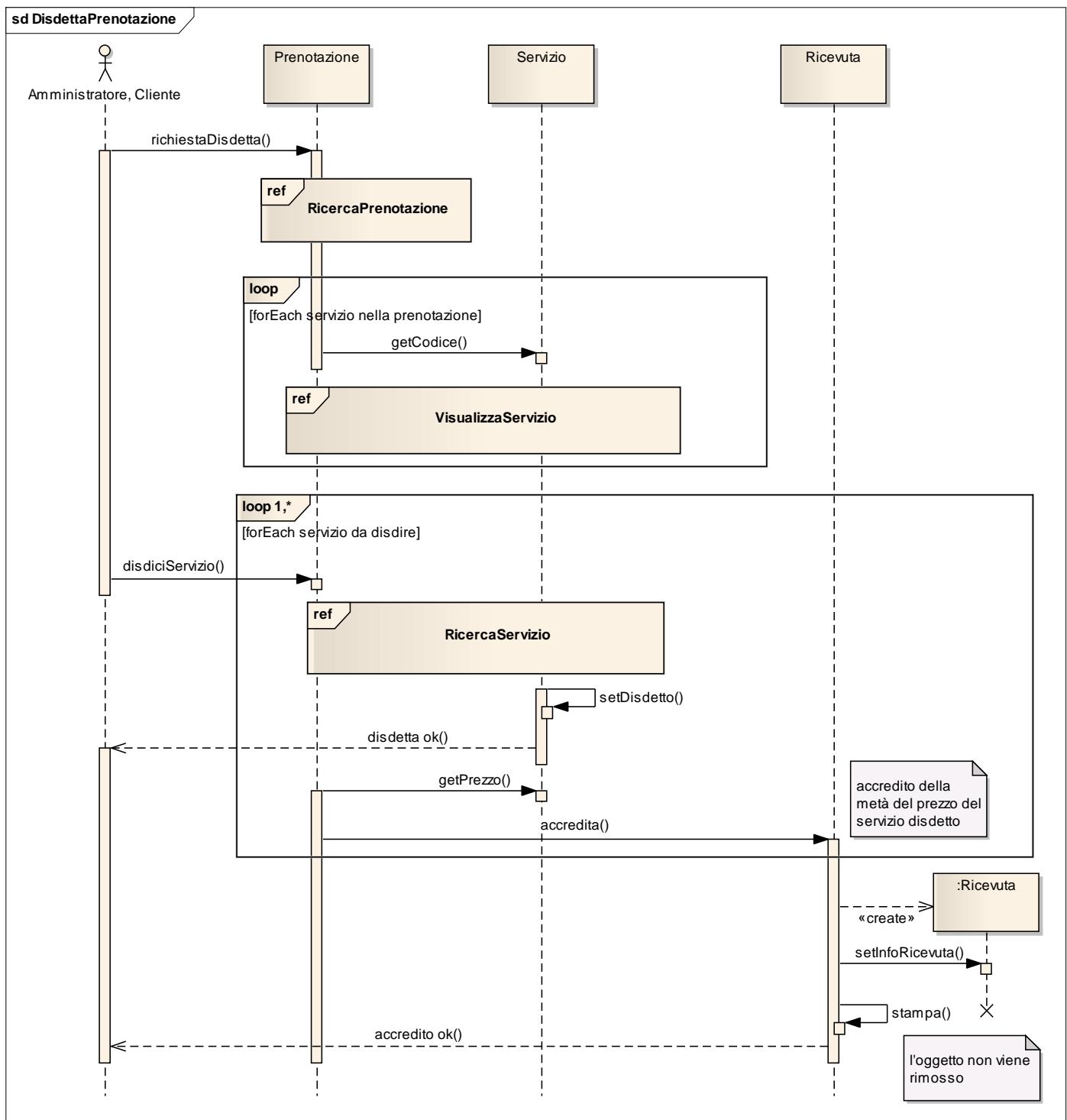
CUD CLIENTE



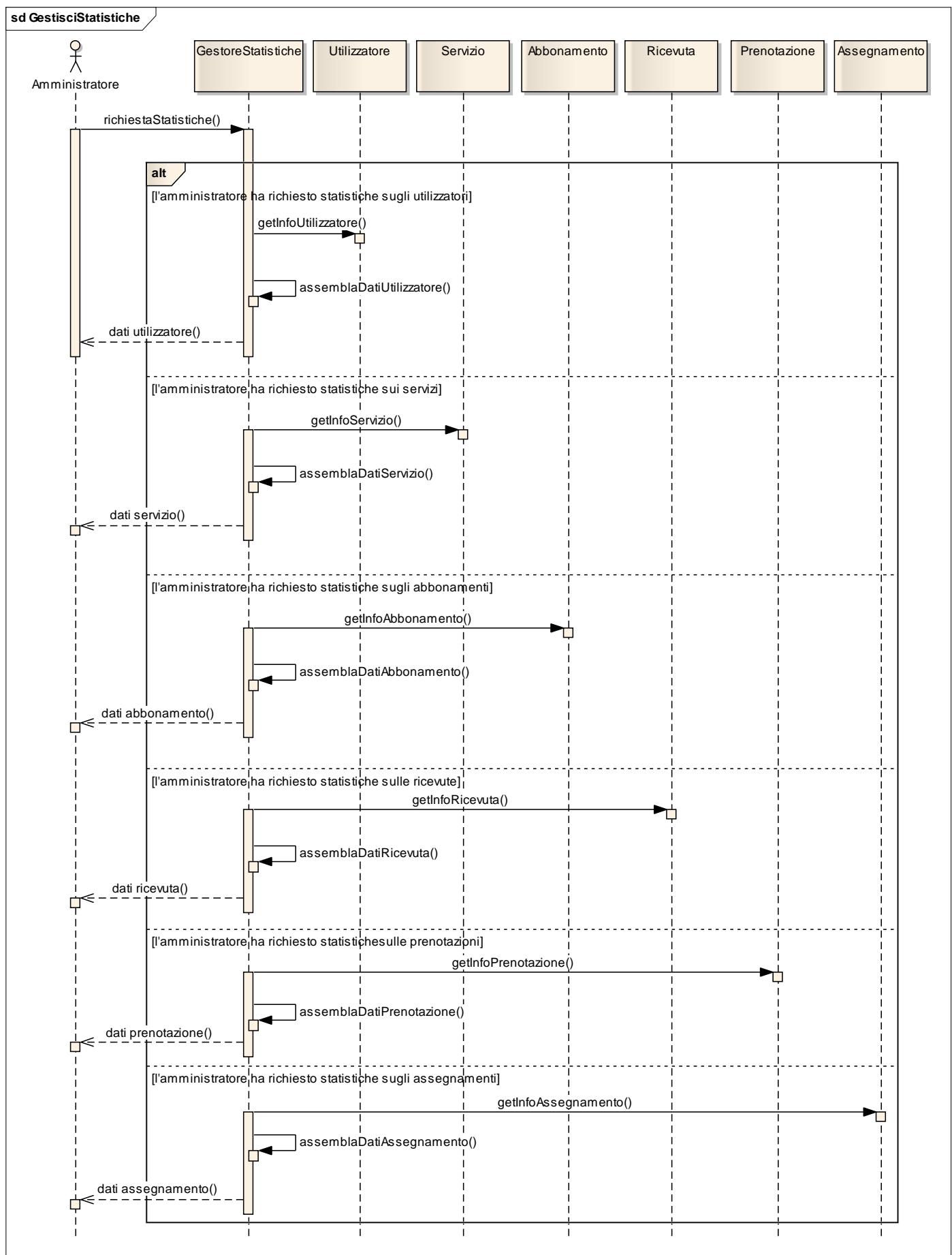
CONSEGNA SERVIZIO



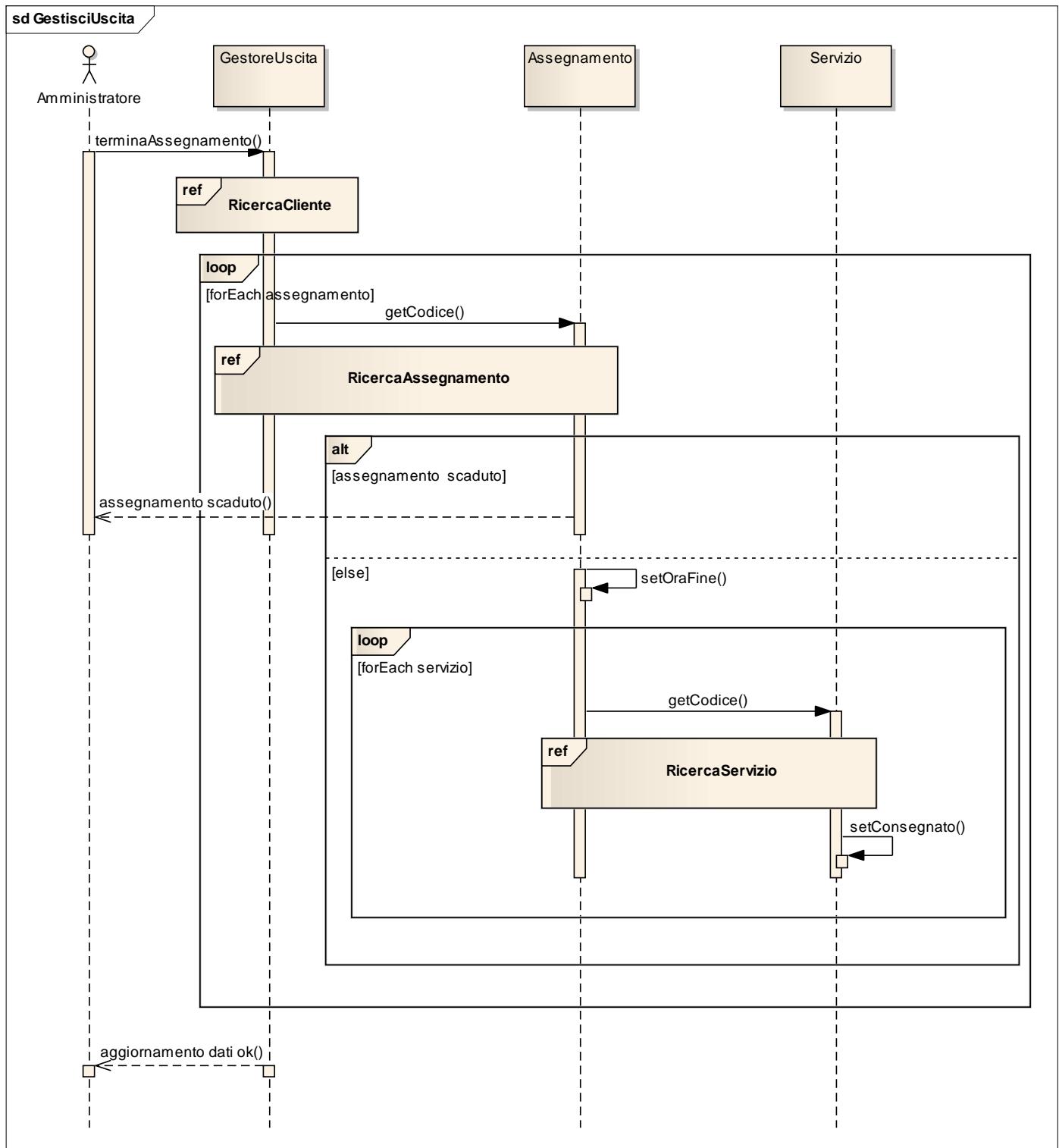
DISDETTA PRENOTAZIONE



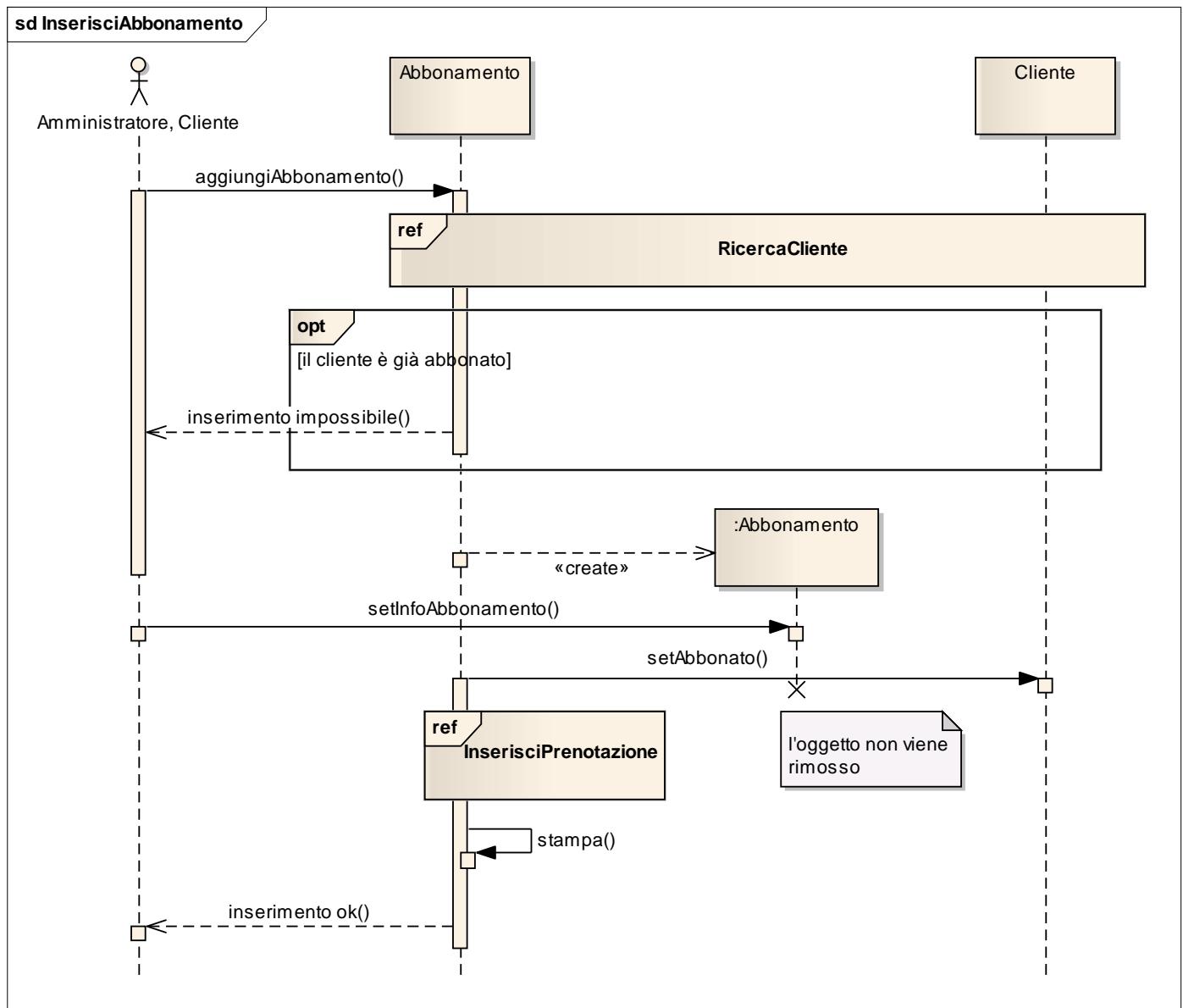
GESTISCI STATISTICHE



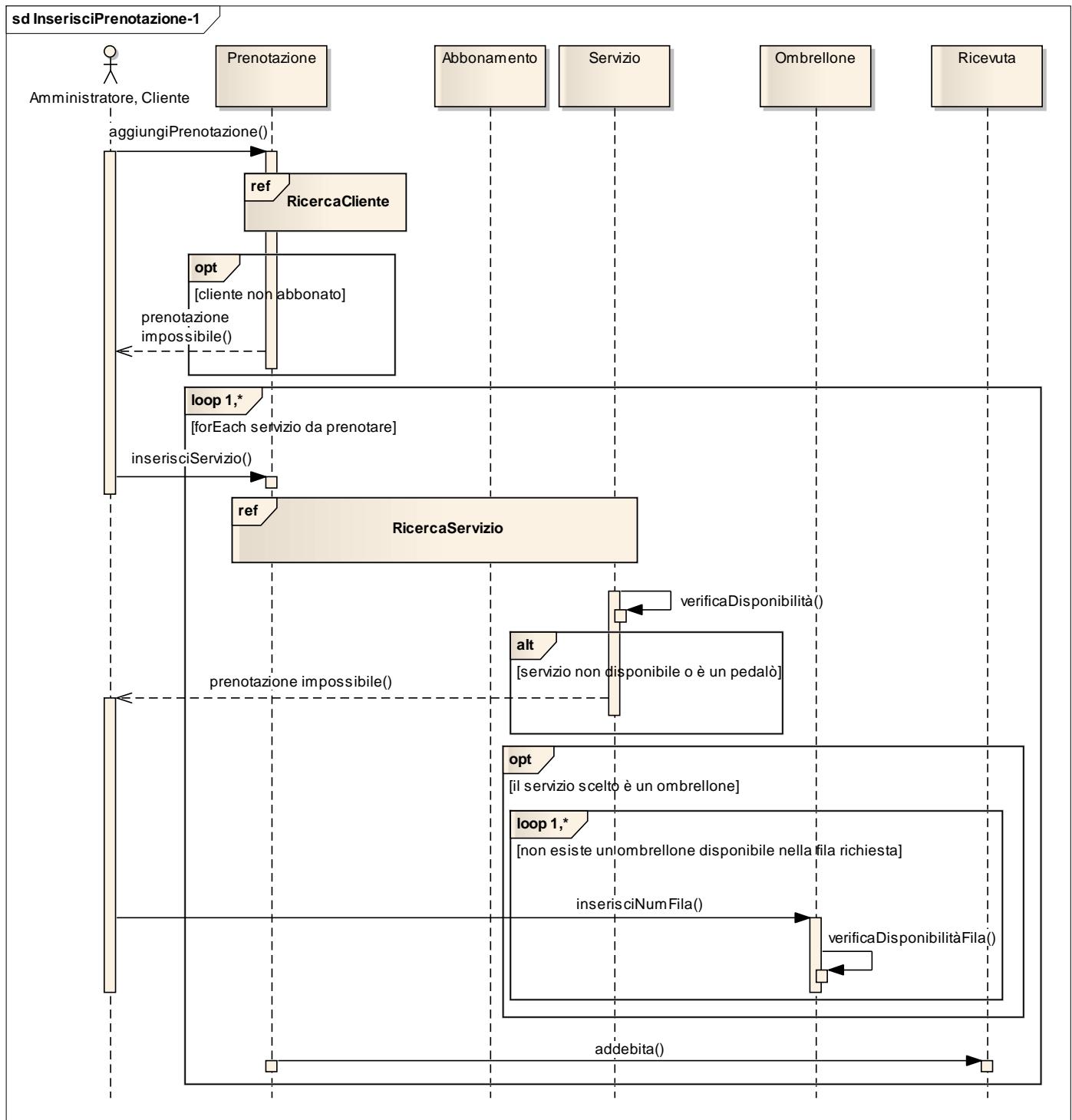
GESTISCI USCITA



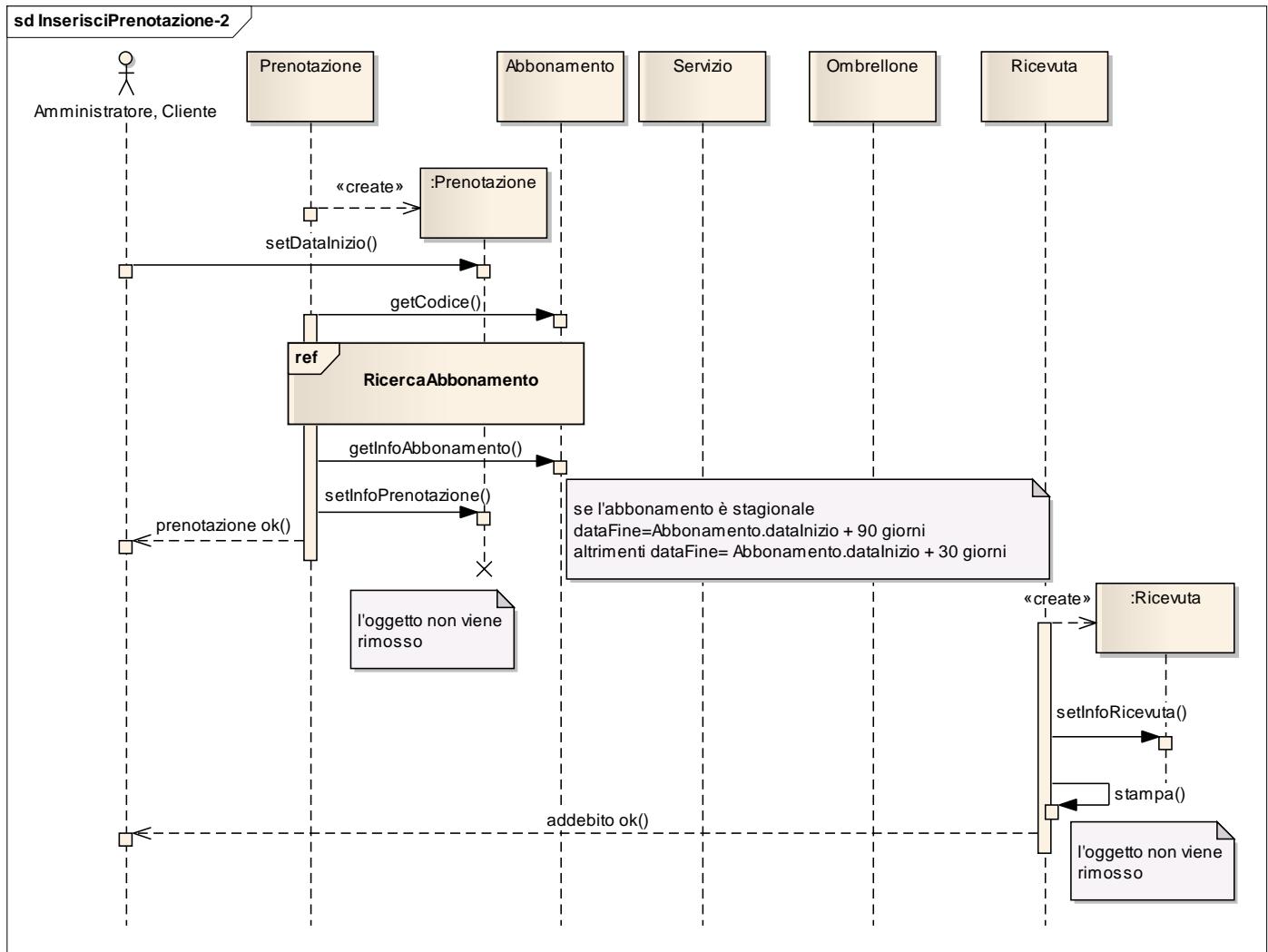
INSERISCI ABBONAMENTO



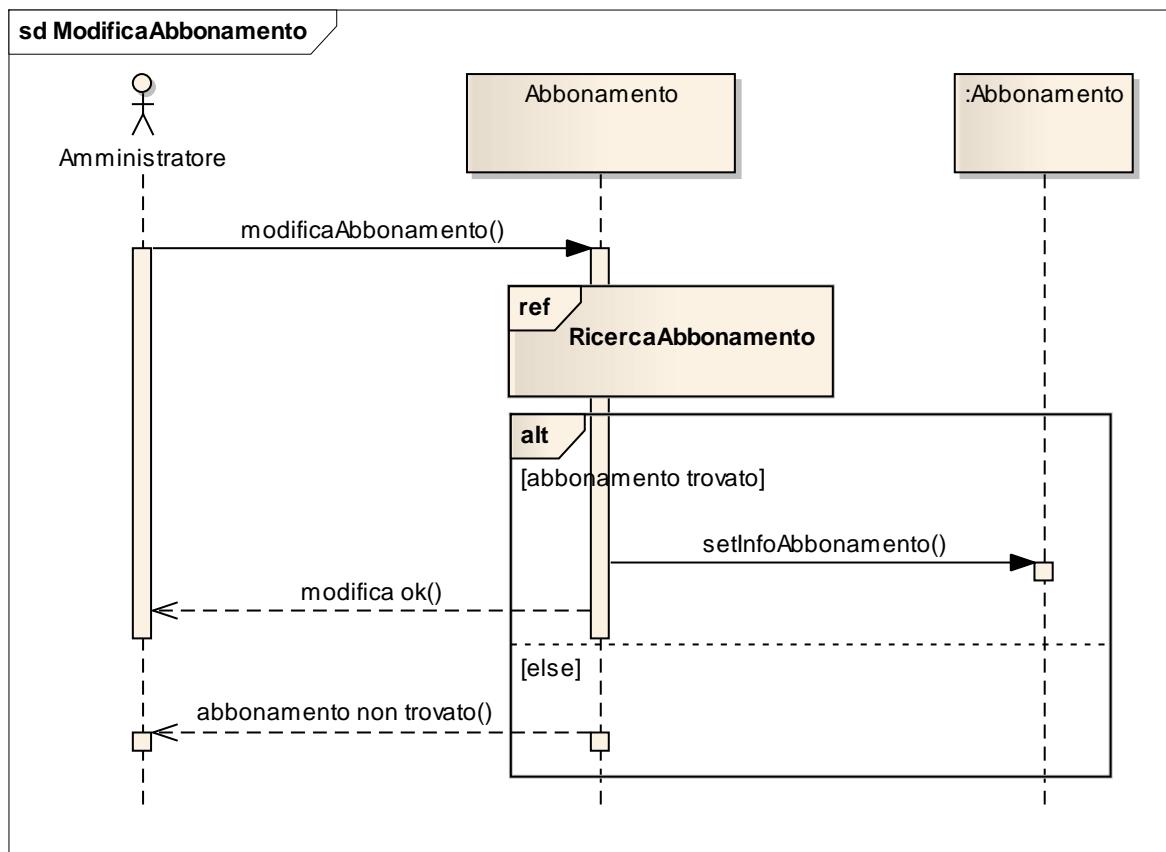
INSERISCI PRENOTAZIONE



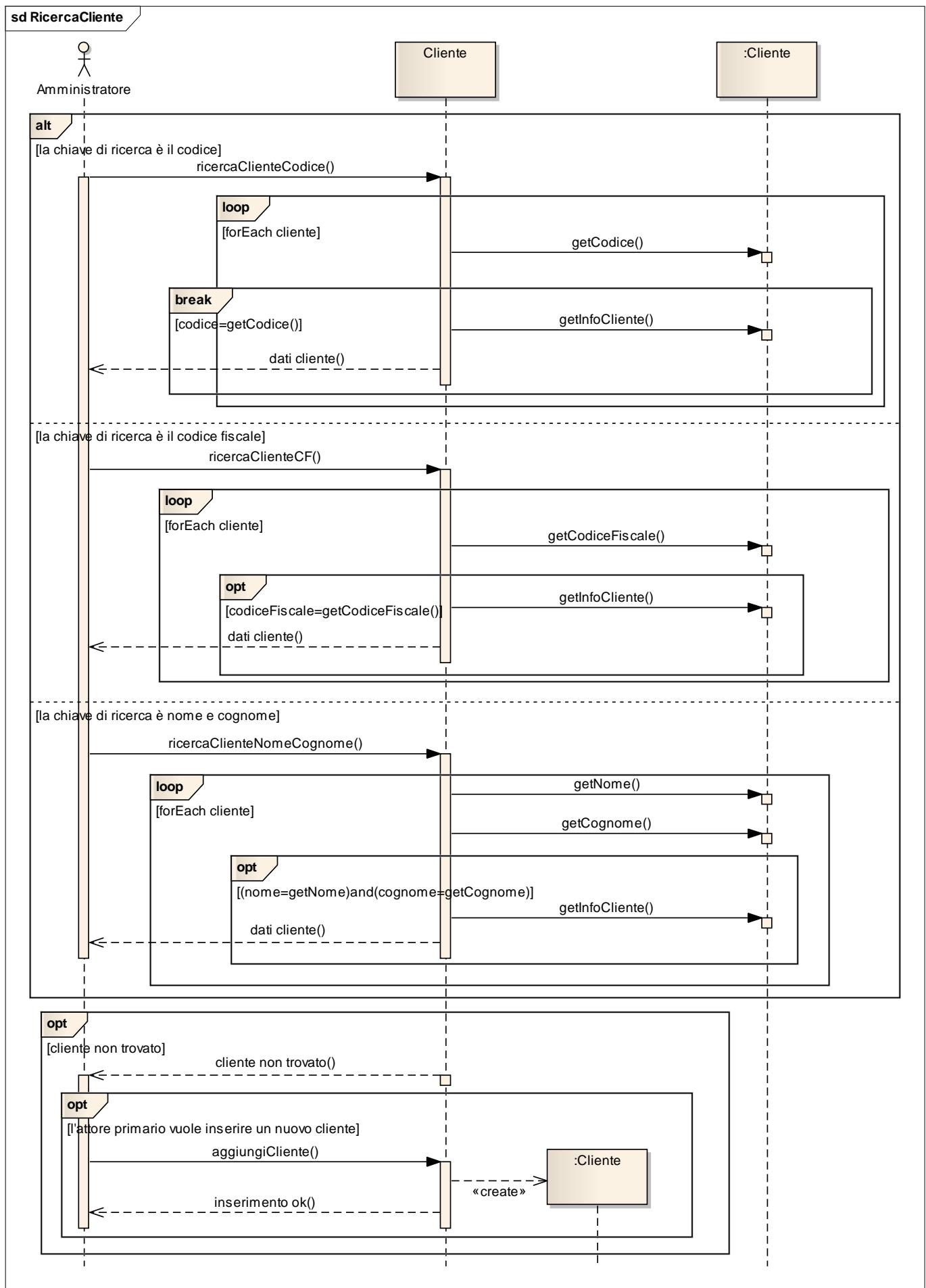
Continua alla pagina successiva



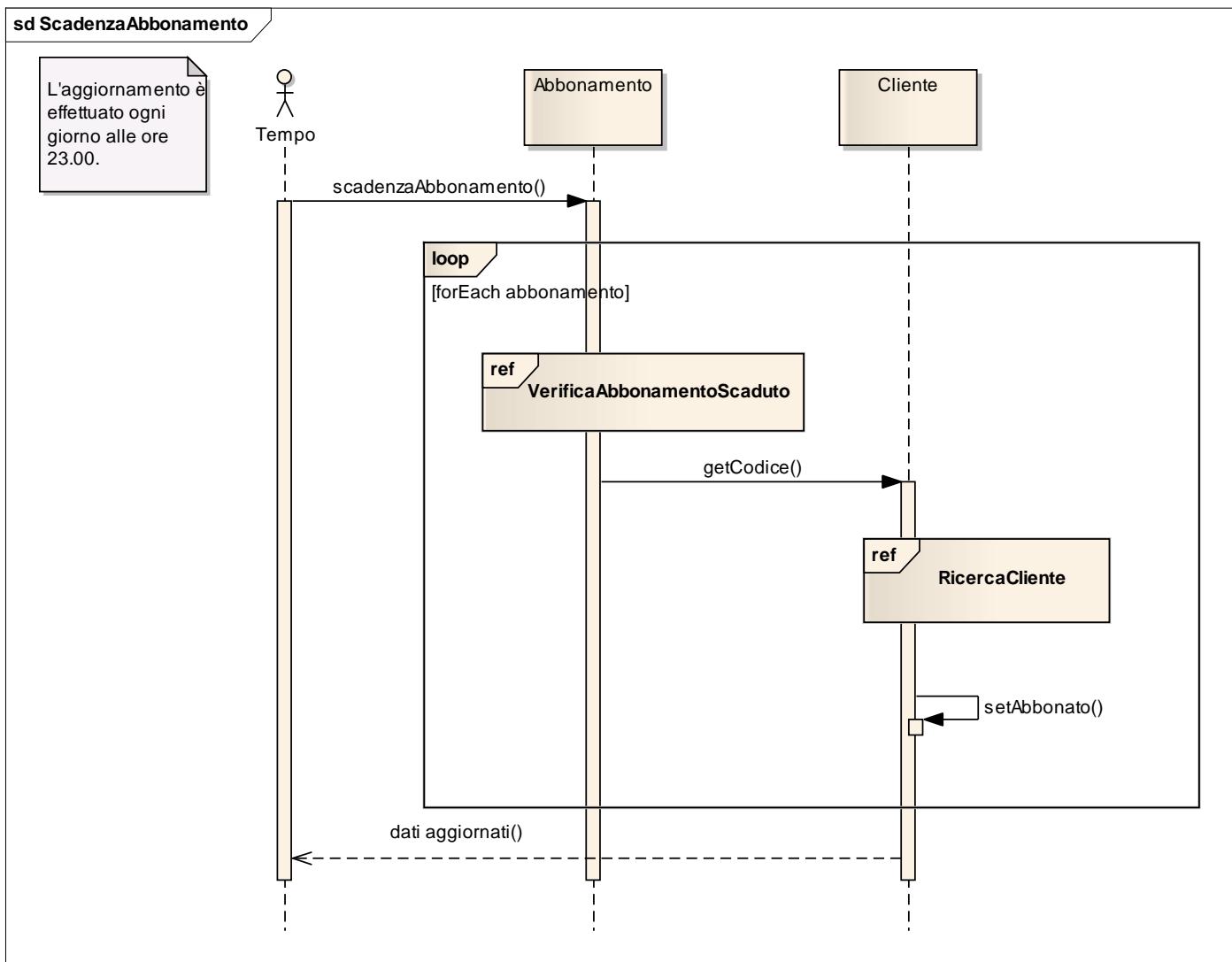
MODIFICA ABBONAMENTO



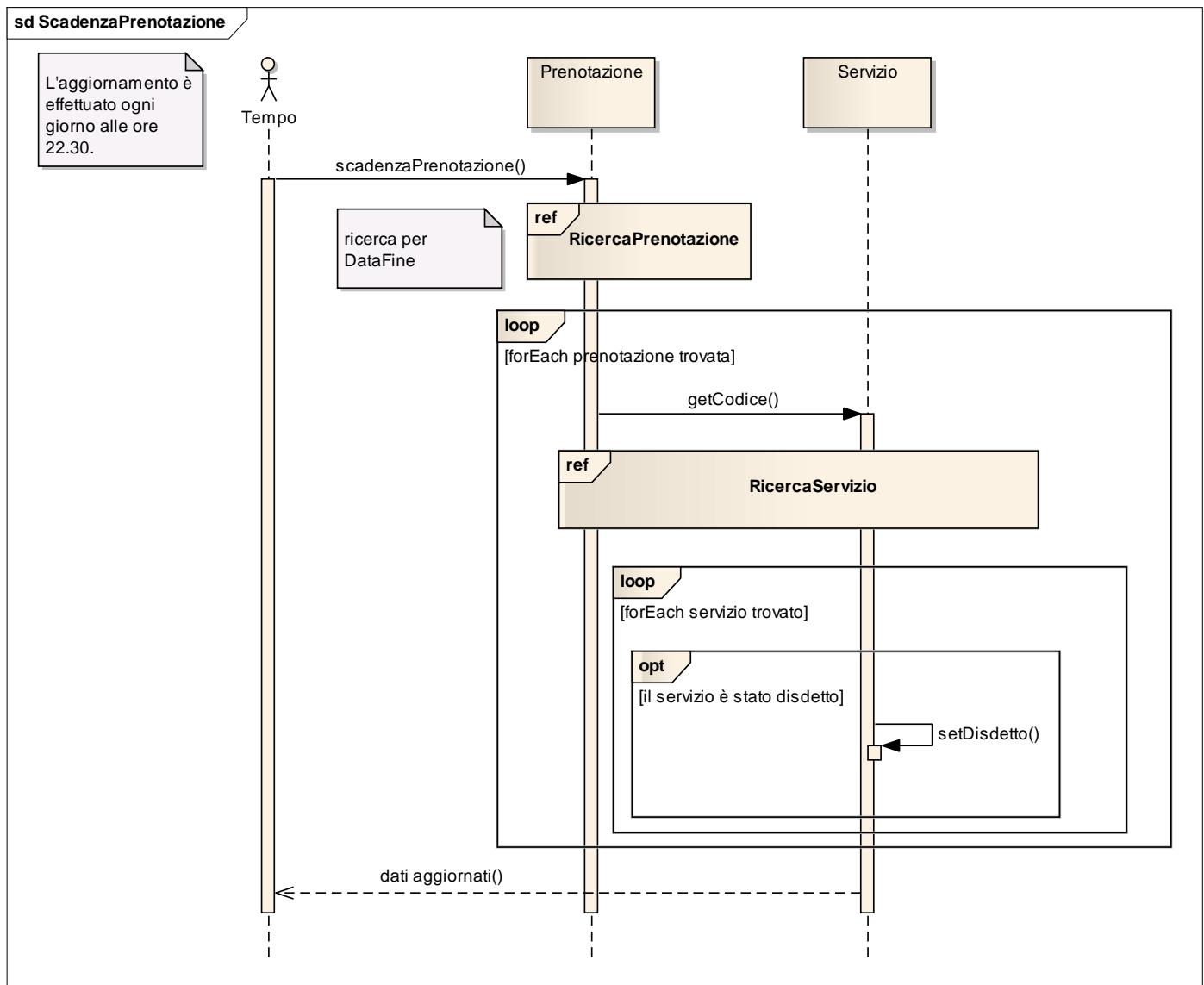
RICERCA CLIENTE



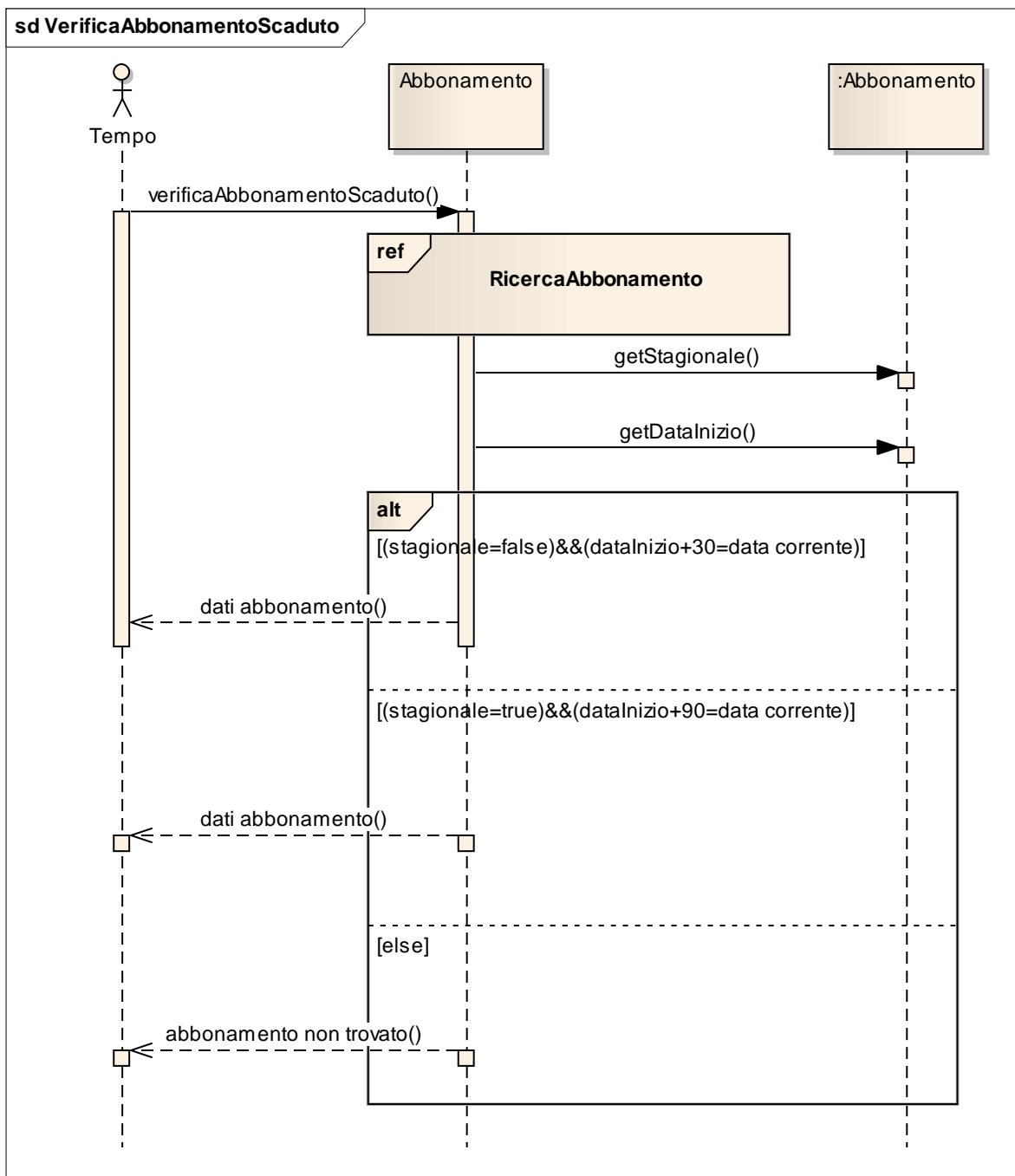
SCADENZA ABBONAMENTO



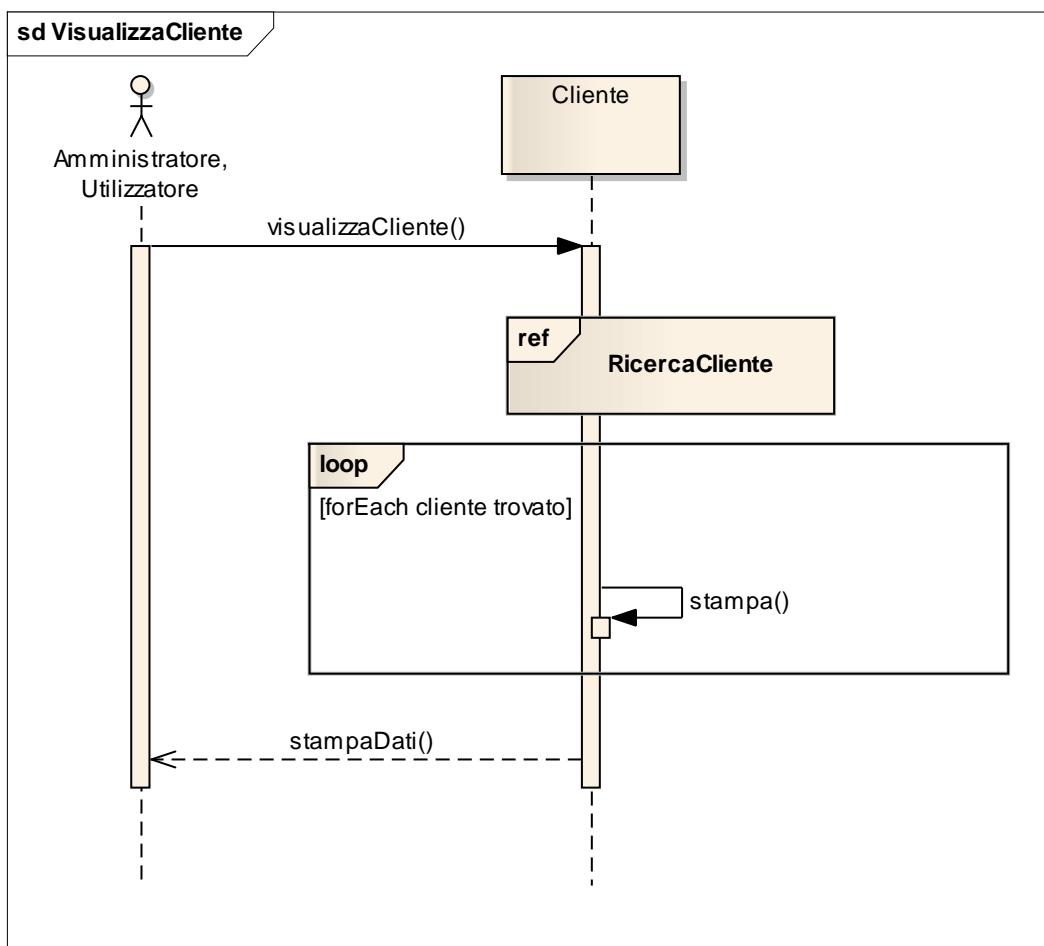
SCADENZA PRENOTAZIONE



VERIFICA ABBONAMENTO SCADUTO



VISUALIZZA CLIENTE



DIAGRAMMI DI ATTIVITA'

ASSEGNA SENZA PRENOTAZIONE

Type: Activity

Status: Proposed. Version 1.0. Phase 1.0.

Package: Assegna senza prenotazione

Constraints

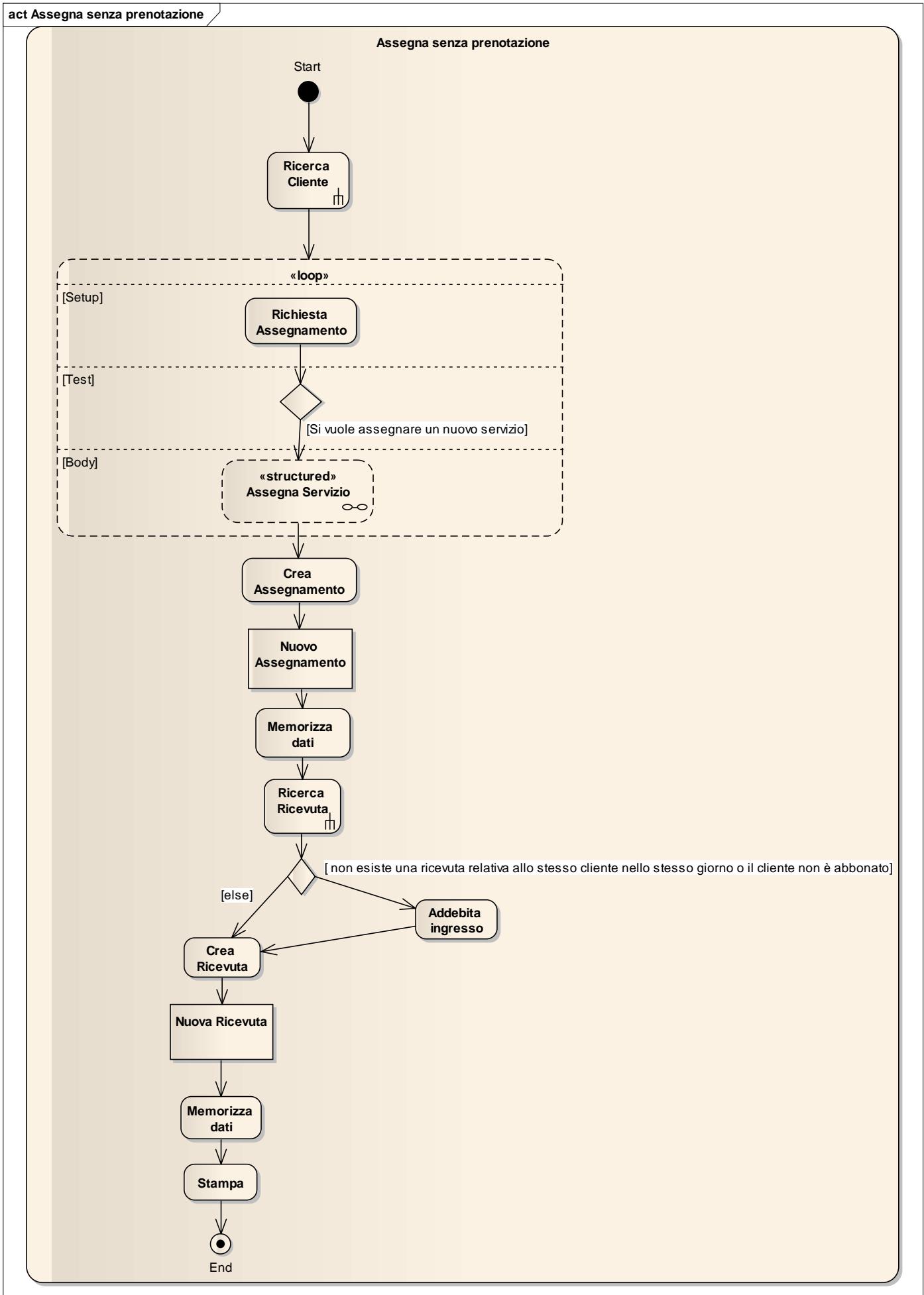
- Postcondizioni: *(Post-condition, Status is Approved)*

L'assegnamento è stato effettuato.

- Precondizioni

: *(Post-condition, Status is Approved)*

L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.



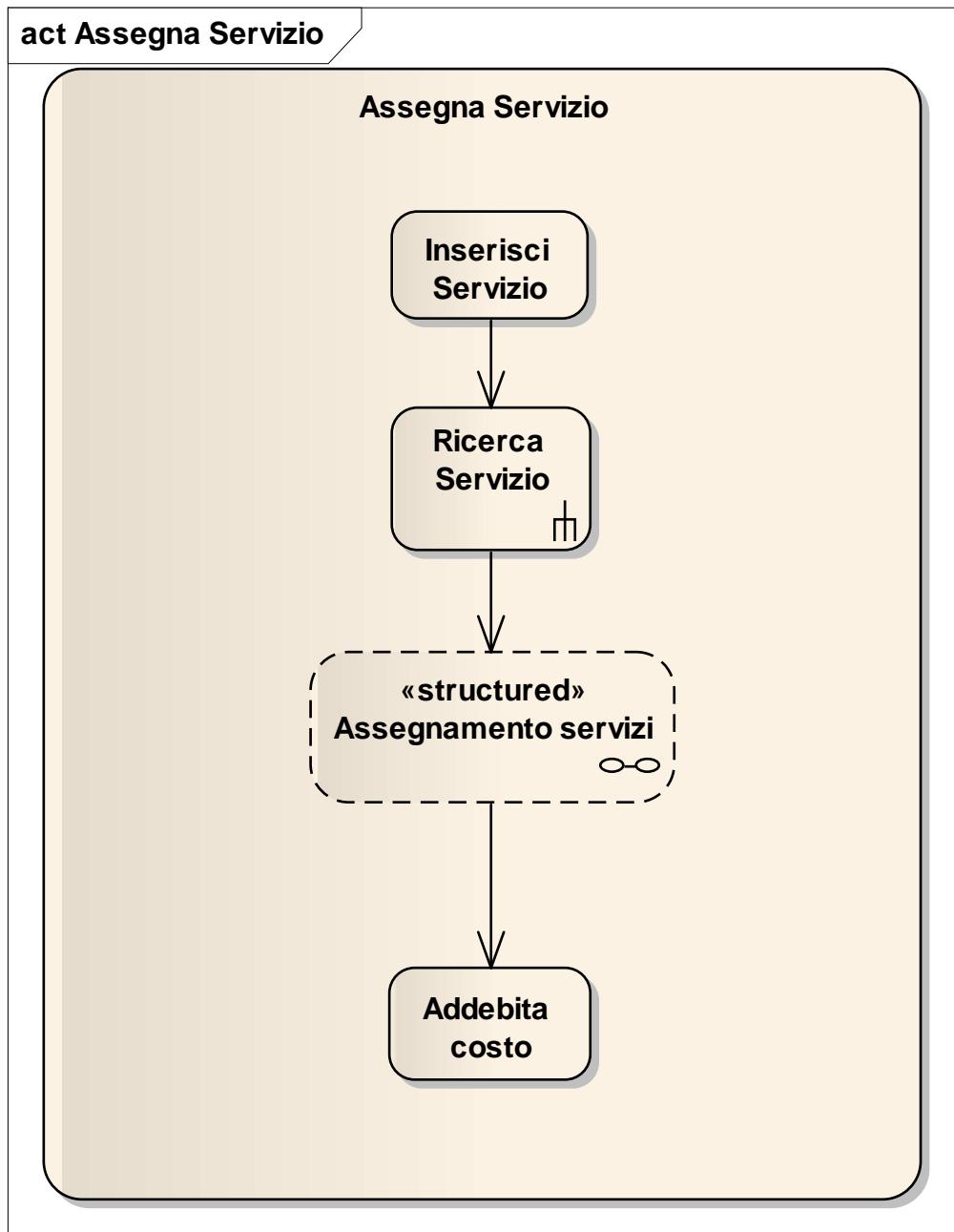
Assegna Servizio

Type: **Activity**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Assegna senza prenotazione

Nodo strutturato Assegna Servizio del diagramma di attività Assegna senza prenotazione



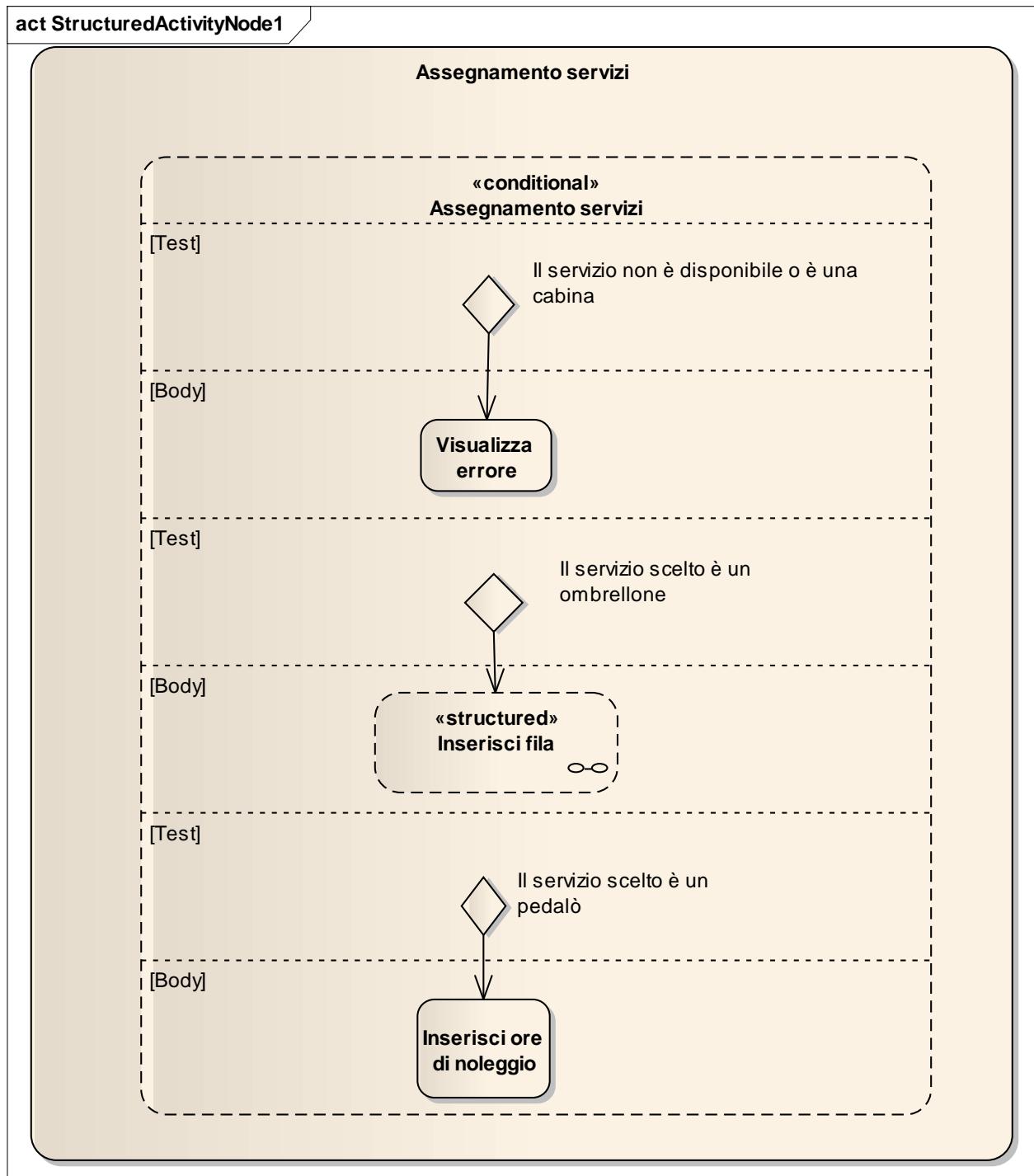
Assegnamento servizi

Type: **Activity**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Assegna senza prenotazione

Nodo strutturato Assegnamento servizi del nodo Assegna Servizio



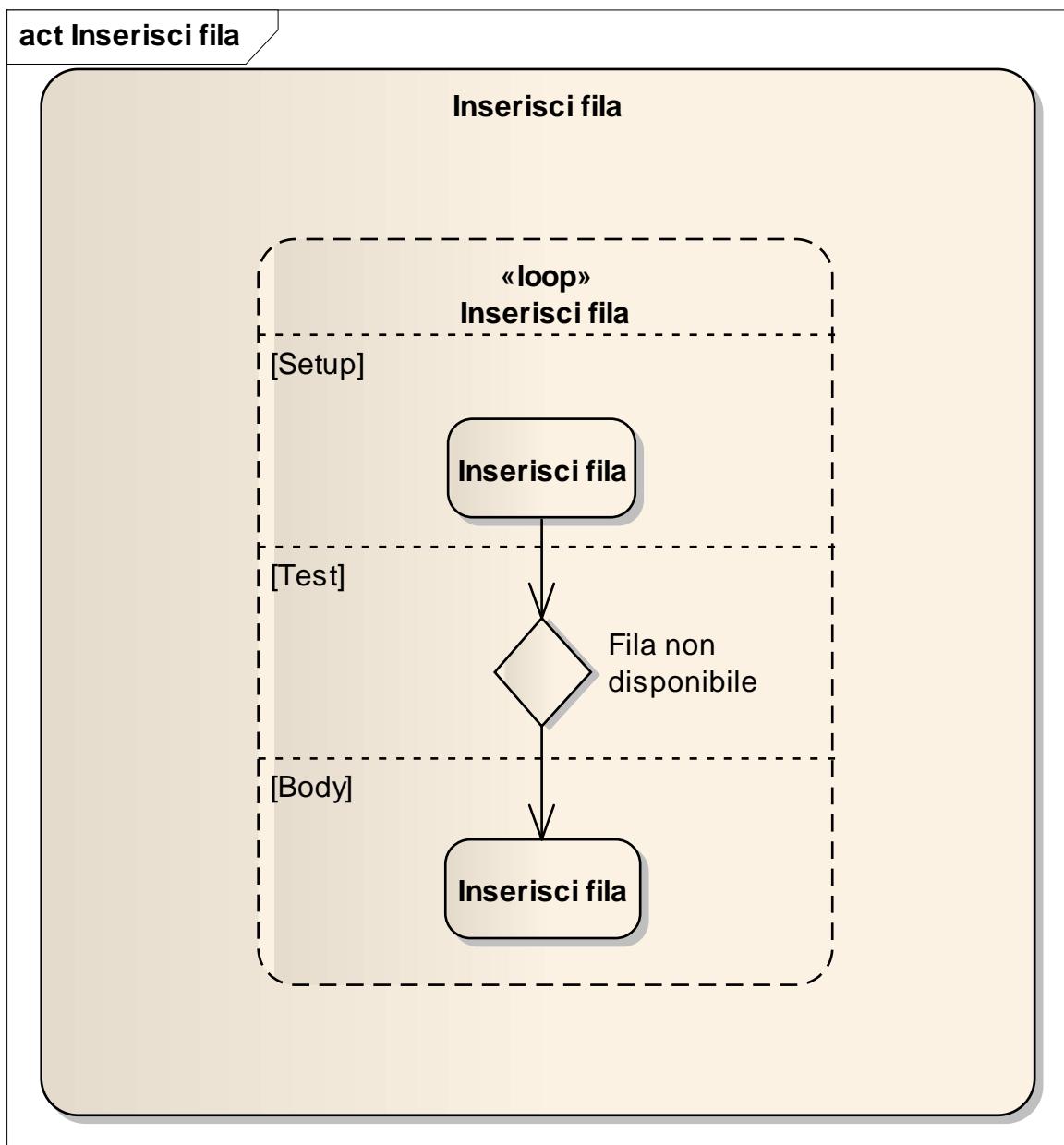
Inserisci fila

Type: **Activity**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Assegna senza prenotazione

Nodo strutturato Inserisci Fila del nodo Assegna Servizio



BACKUP

Type: **Package**

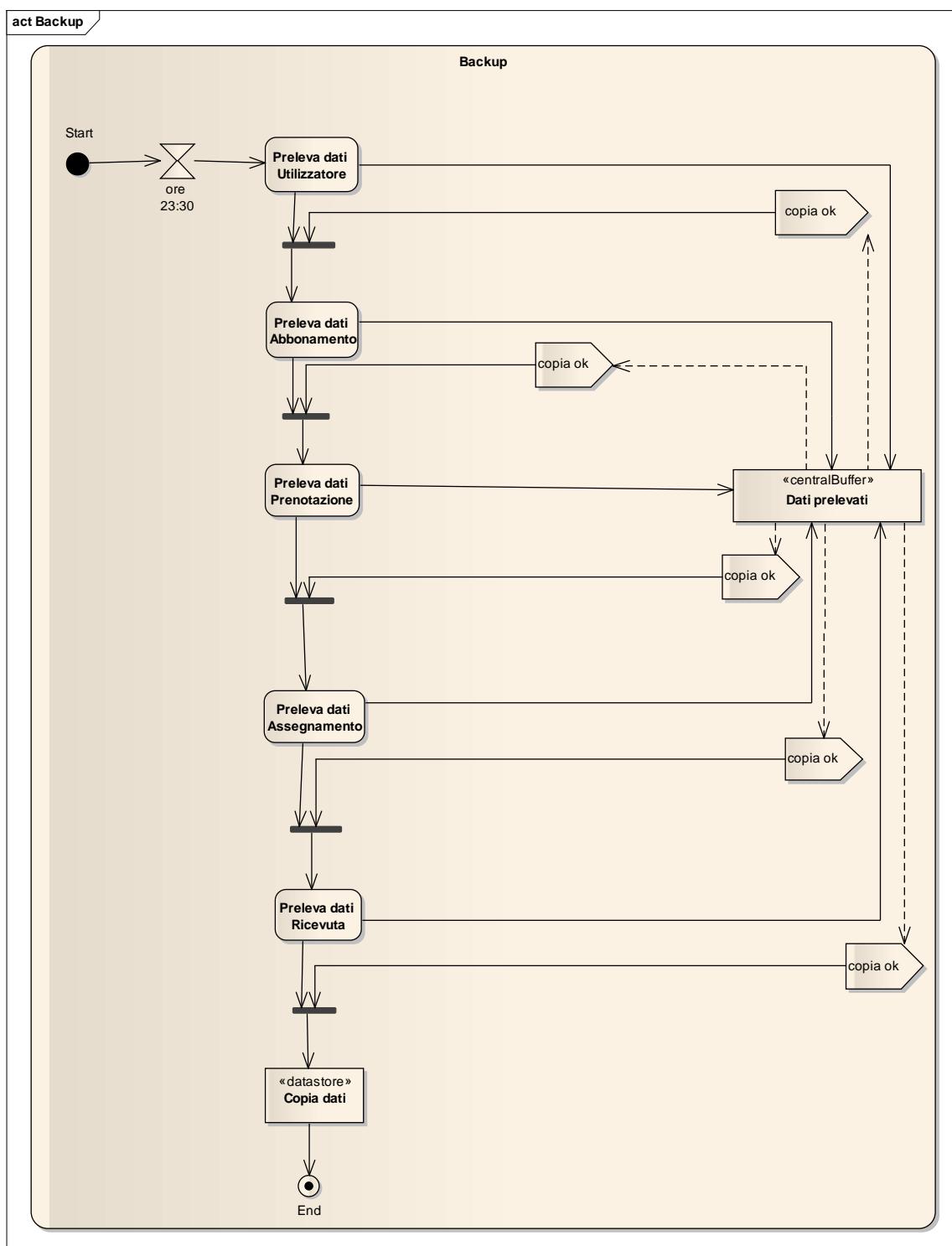
Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi di attività

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
 1. Nessuna
- Postcondizioni: (*Post-condition, Status is Approved*)

I dati sono stati copiati



CUD CLIENTE

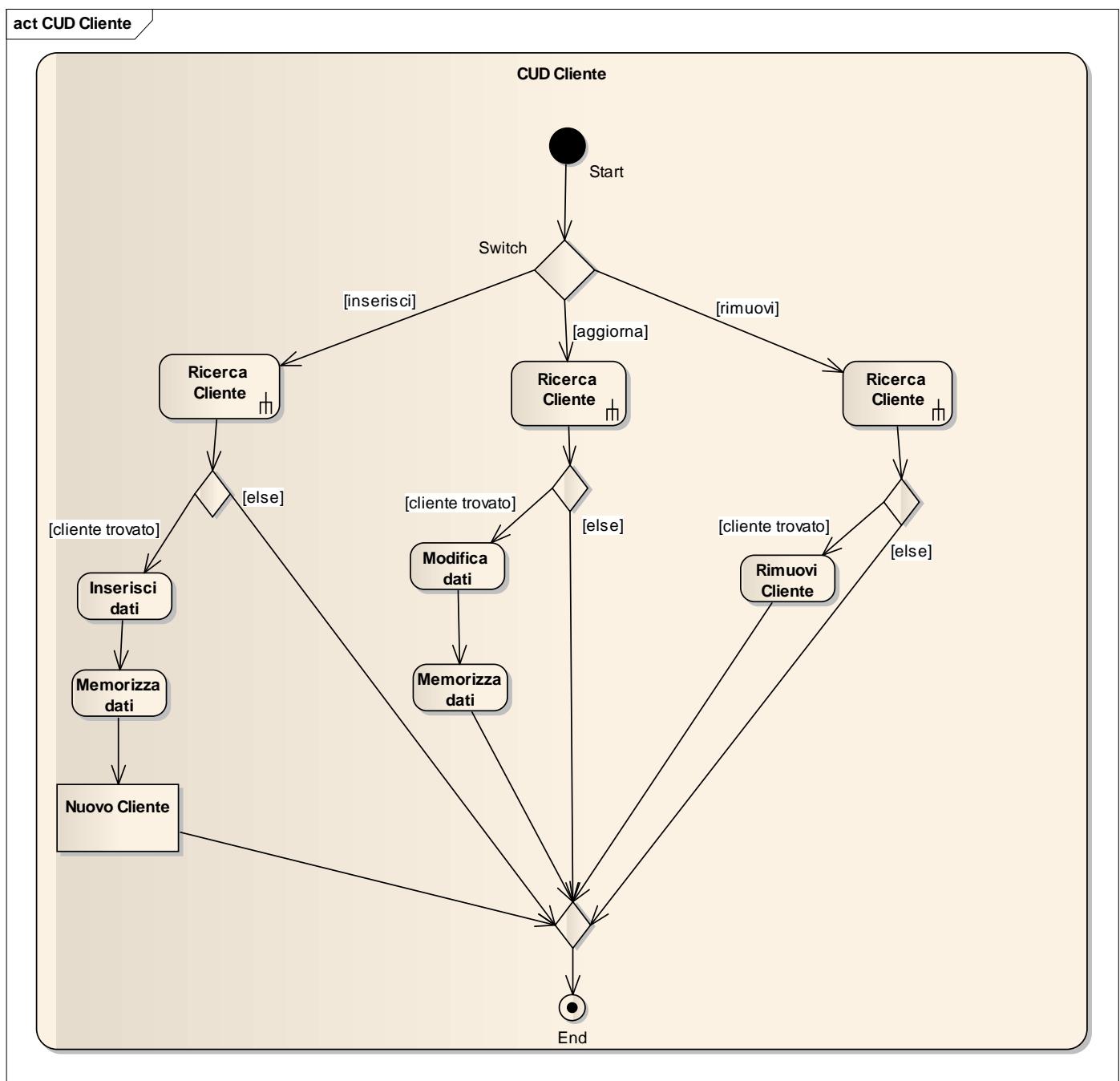
Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi di attività

Constraints

- L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso: (*Pre-condition, Status is Approved*)
- Nessuna: (*Post-condition, Status is Approved*)



GESTISCI STATISTICHE

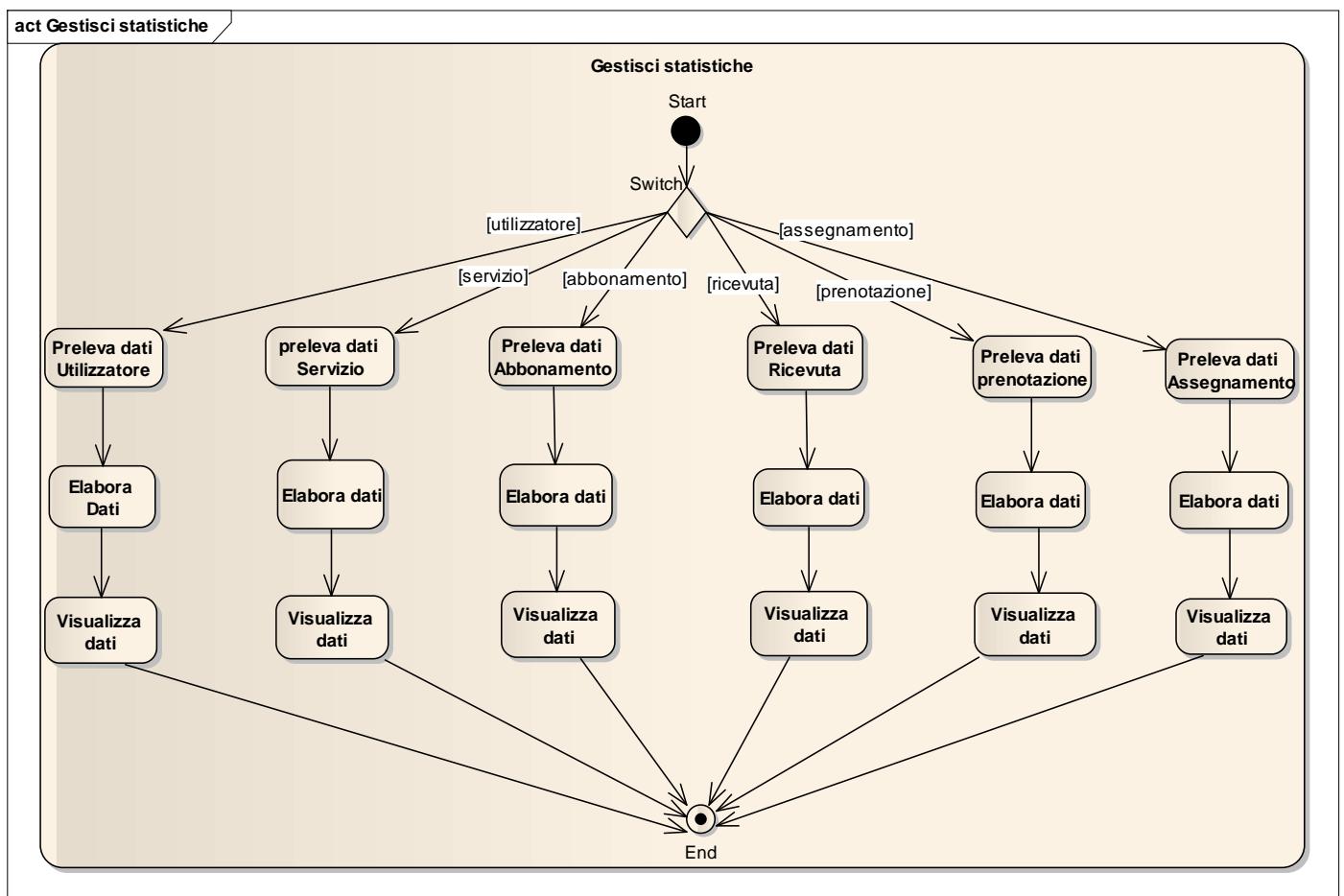
Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi di attività

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
L'attore primario deve disporre di un account con appropriati diritti d'accesso.
- Postcondizioni: (*Post-condition, Status is Approved*)
Il sistema ha visualizzato le statistiche richieste.



GESTISCI USCITA

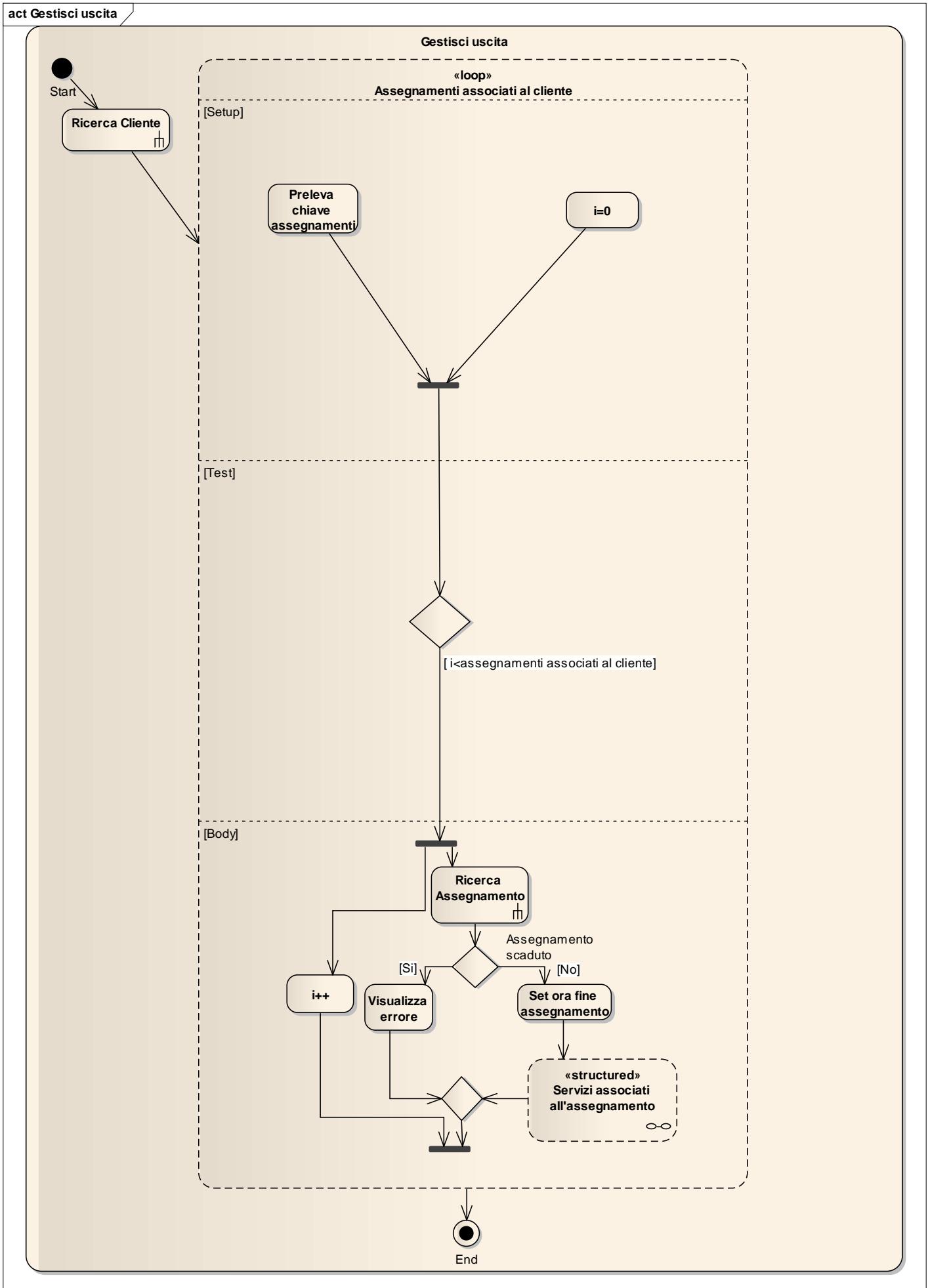
Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Diagrammi di attività

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso
- Postcondizioni: (*Post-condition, Status is Approved*)
 1. Il sistema modifica la disponibilità dei servizi associati al cliente in uscita;
 2. Il sistema setta l'ora di fine assegnamento.



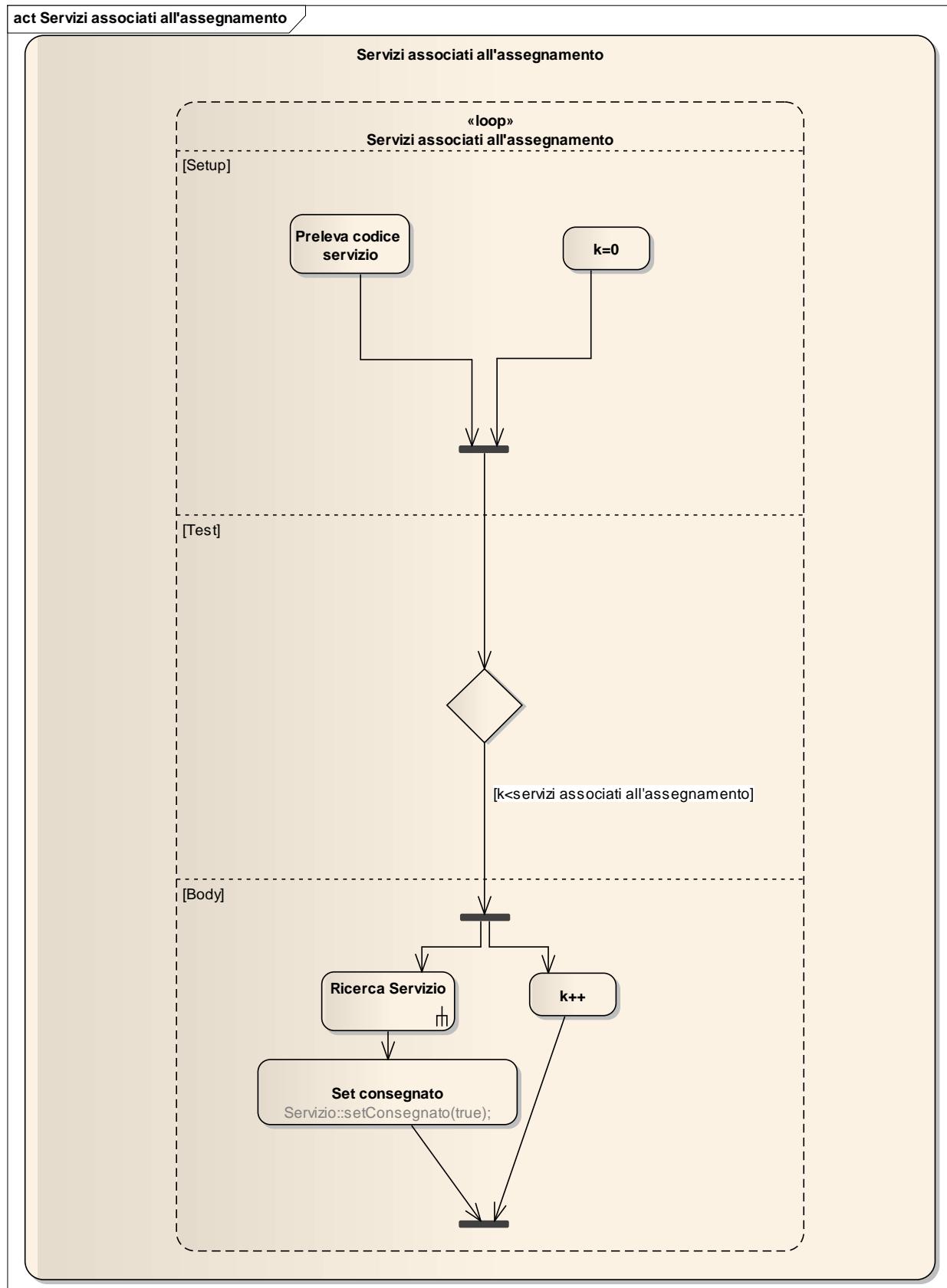
Servizi associati all'assegnamento

Type: **StructuredActivityNode**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestisci uscita

Nodo strutturato Servizi associati all'assegnamento del diagramma di attività Gestisci uscita



INSERISCI PRENOTAZIONE

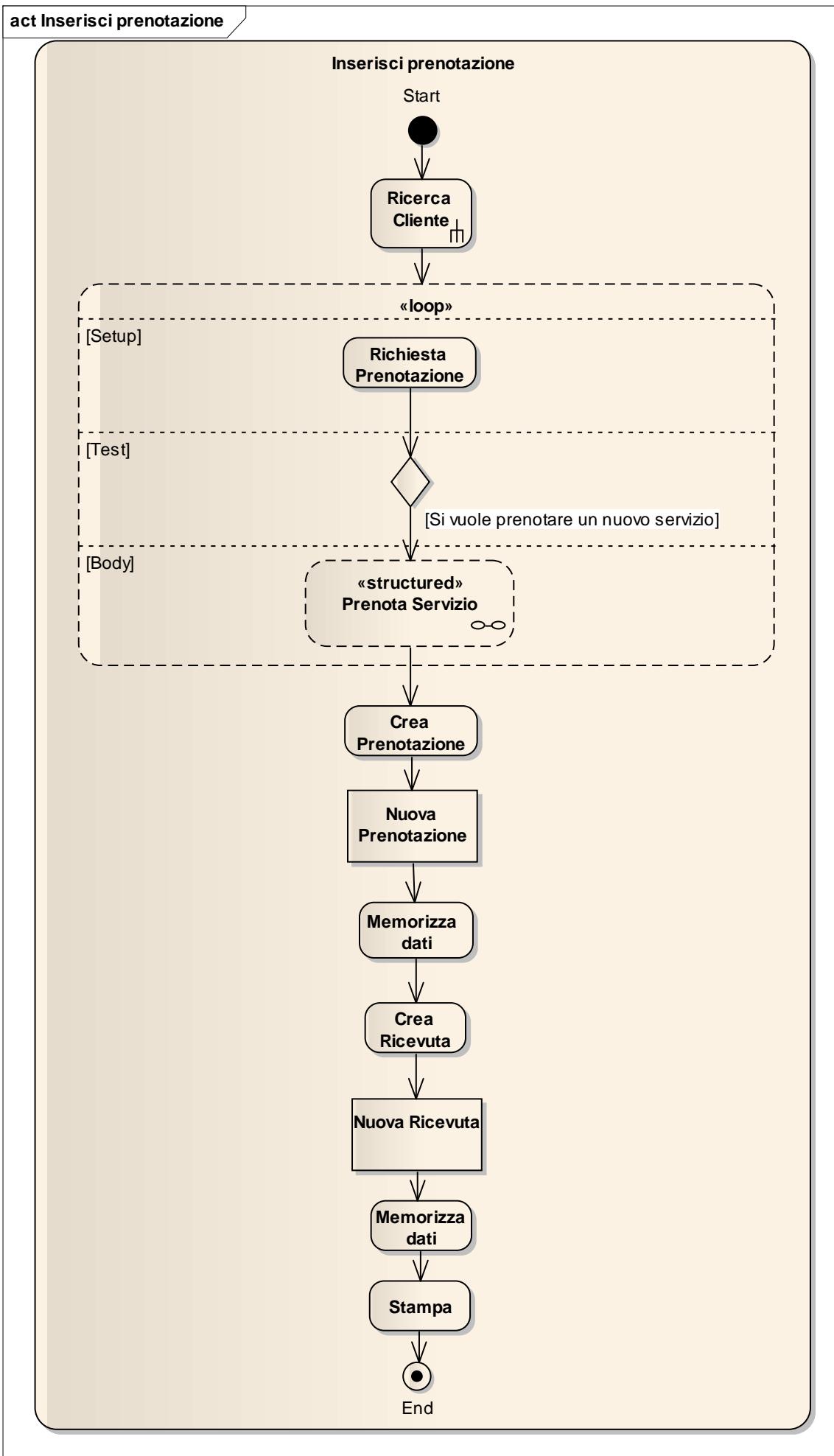
Type: **Activity**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Inserisci prenotazione Keywords:

Constraints

- Postcondizioni: *(Post-condition, Status is Approved)*
La prenotazione è stata effettuata.
- Precondizioni
: (Post-condition, Status is Approved)
L'attore primario deve disporre di un account di utilizzo con appropriati diritti d'accesso.



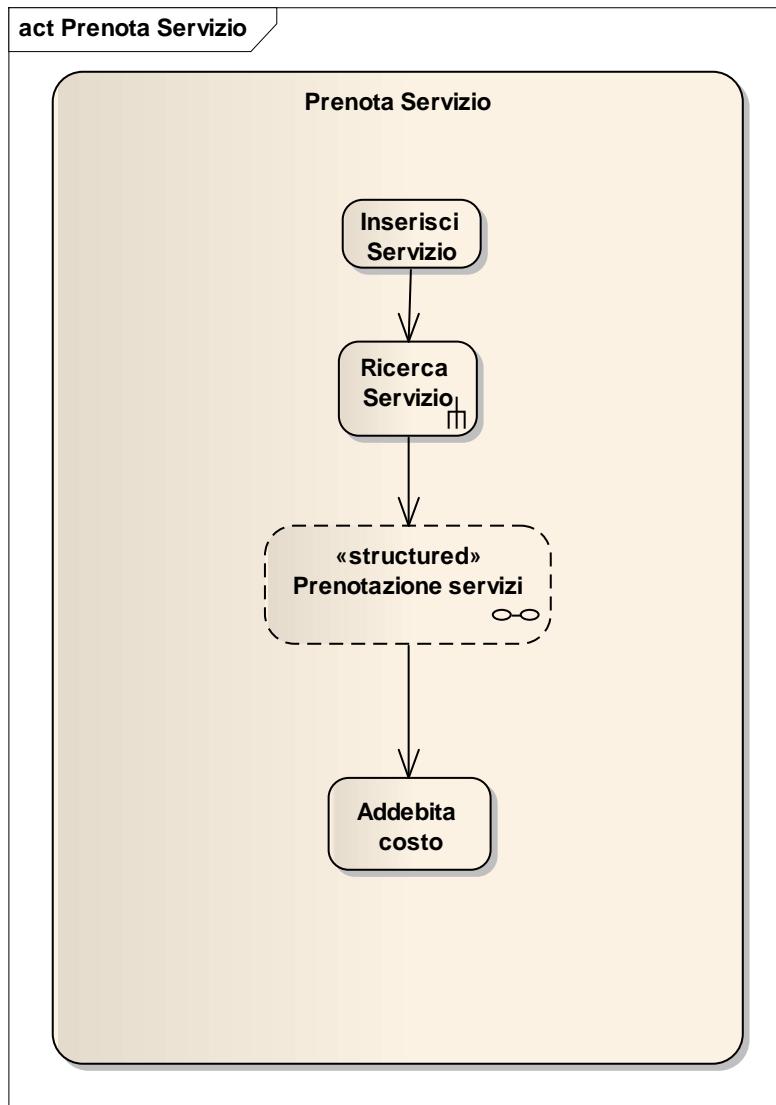
Prenota Servizio

Type: **Activity**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Inserisci prenotazione

Nodo strutturato Prenota servizio del diagramma di attività Inserisci prenotazione



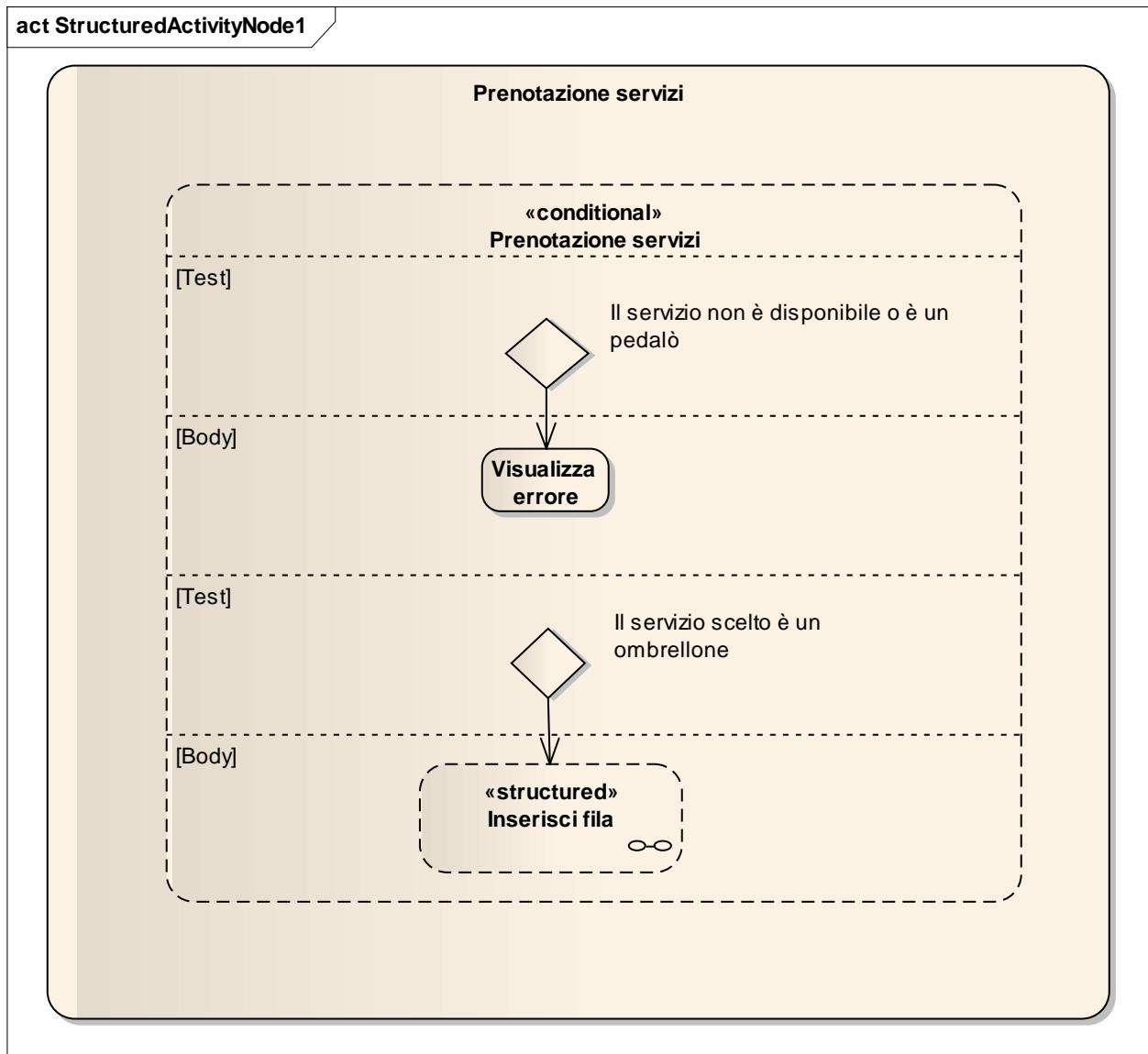
Prenotazione servizi

Type: **StructuredActivityNode**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Inserisci prenotazione

Nodo strutturato Prenotazione servizi del nodo Prenota servizio



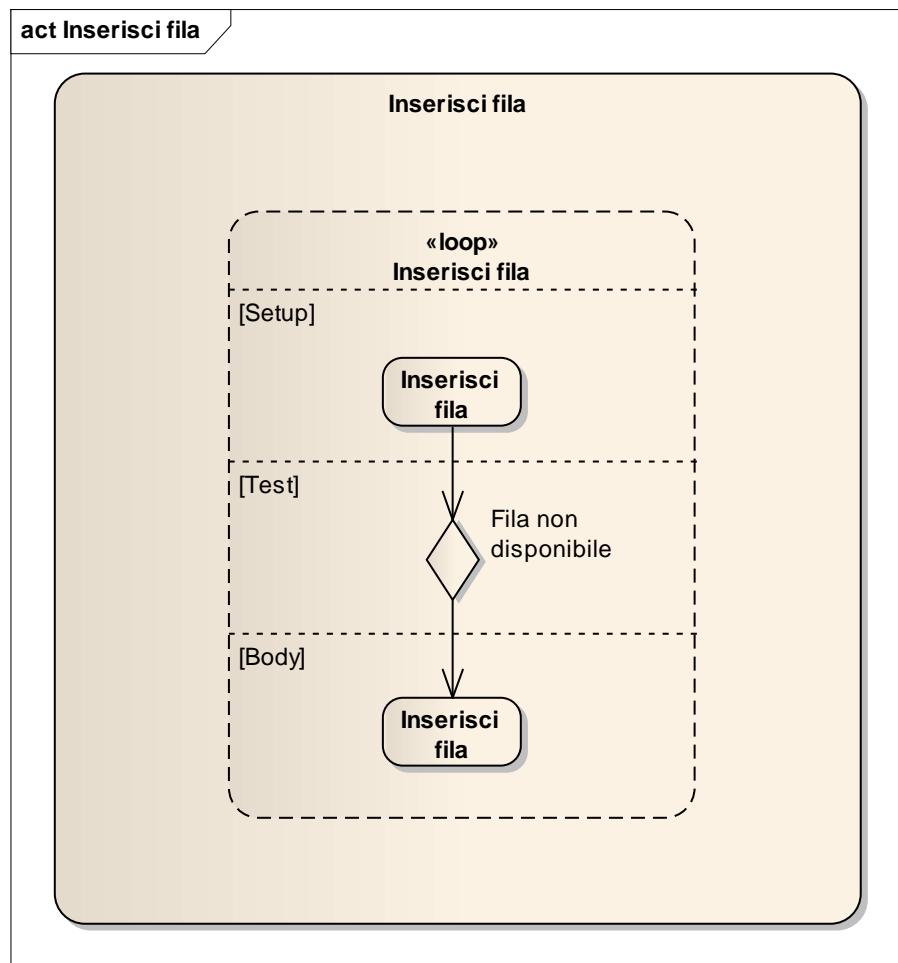
Inserisci fila

Type: **Activity**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Inserisci prenotazione

Nodo strutturato inserisci fila del nodo Prenotazione servizi



RICERCA CLIENTE

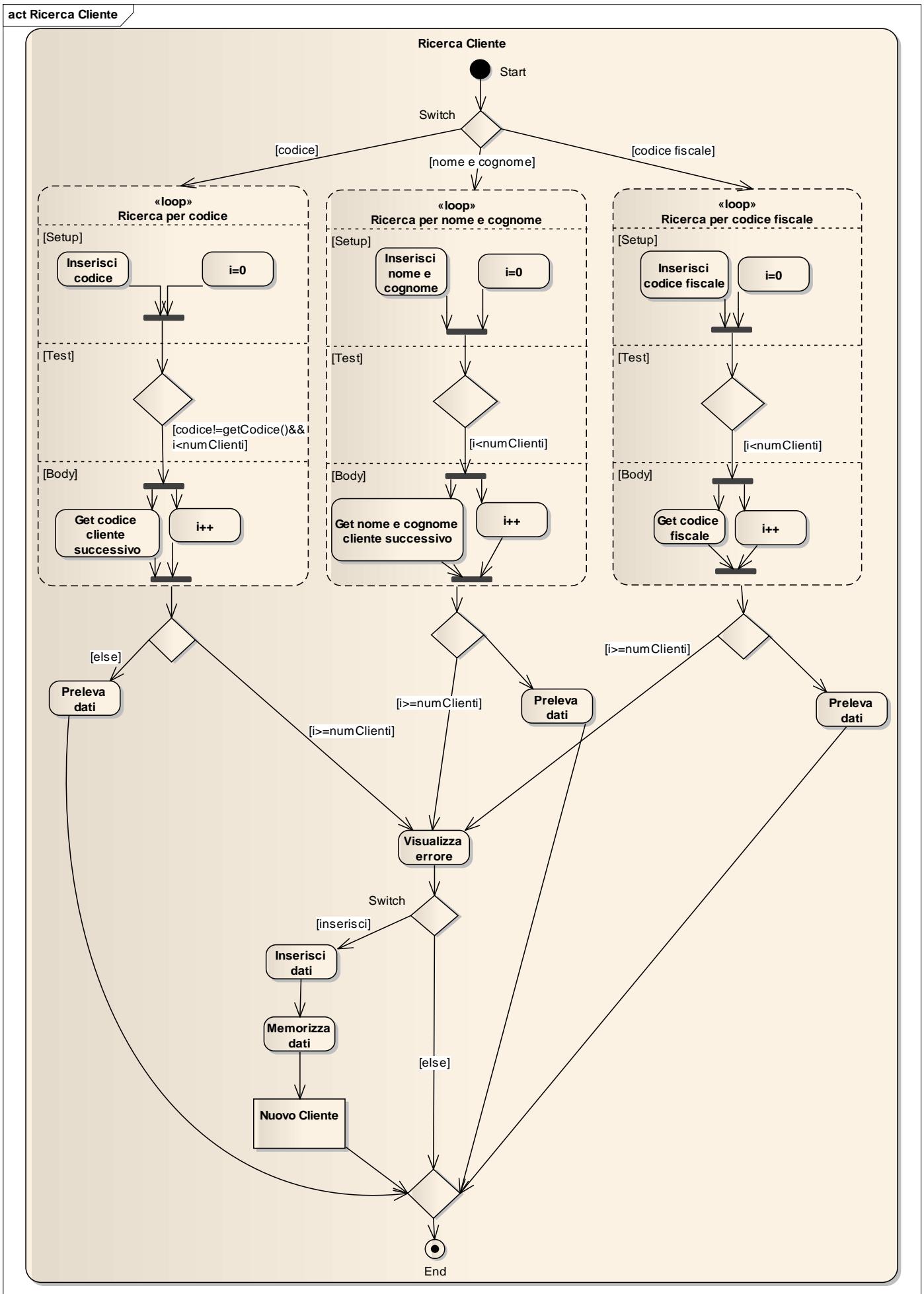
Type: **Activity**

Status: Proposed. Version 1.0. Phase 1.0.

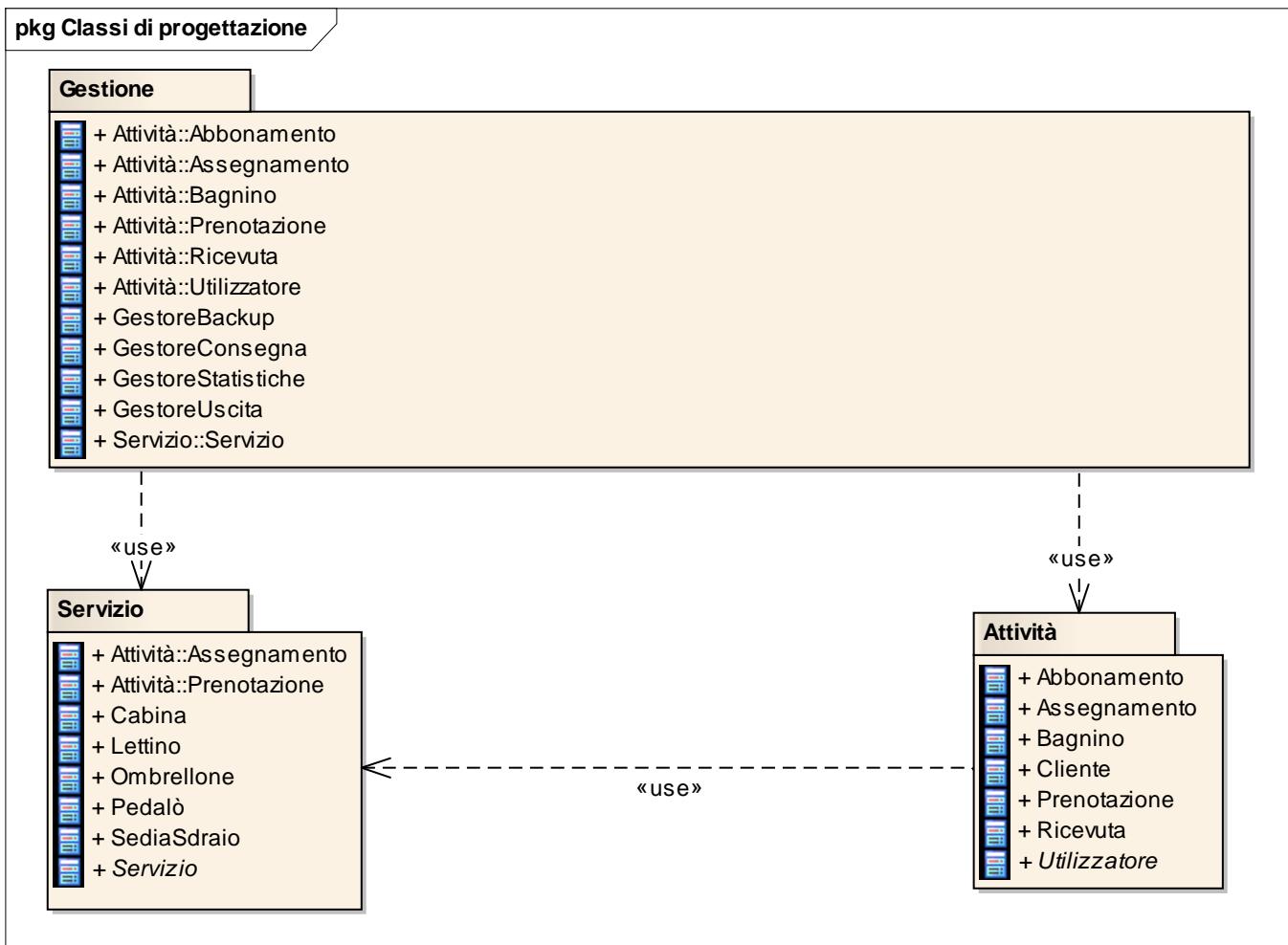
Package: Ricerca Cliente

Constraints

- Precondizioni: (*Pre-condition, Status is Approved*)
Nessuna.
- Postcondizioni: (*Post-condition, Status is Approved*)
La ricerca del cliente è stata effettuata oppure si è determinata l'impossibilità di farlo.



CLASSI DI PROGETTAZIONE

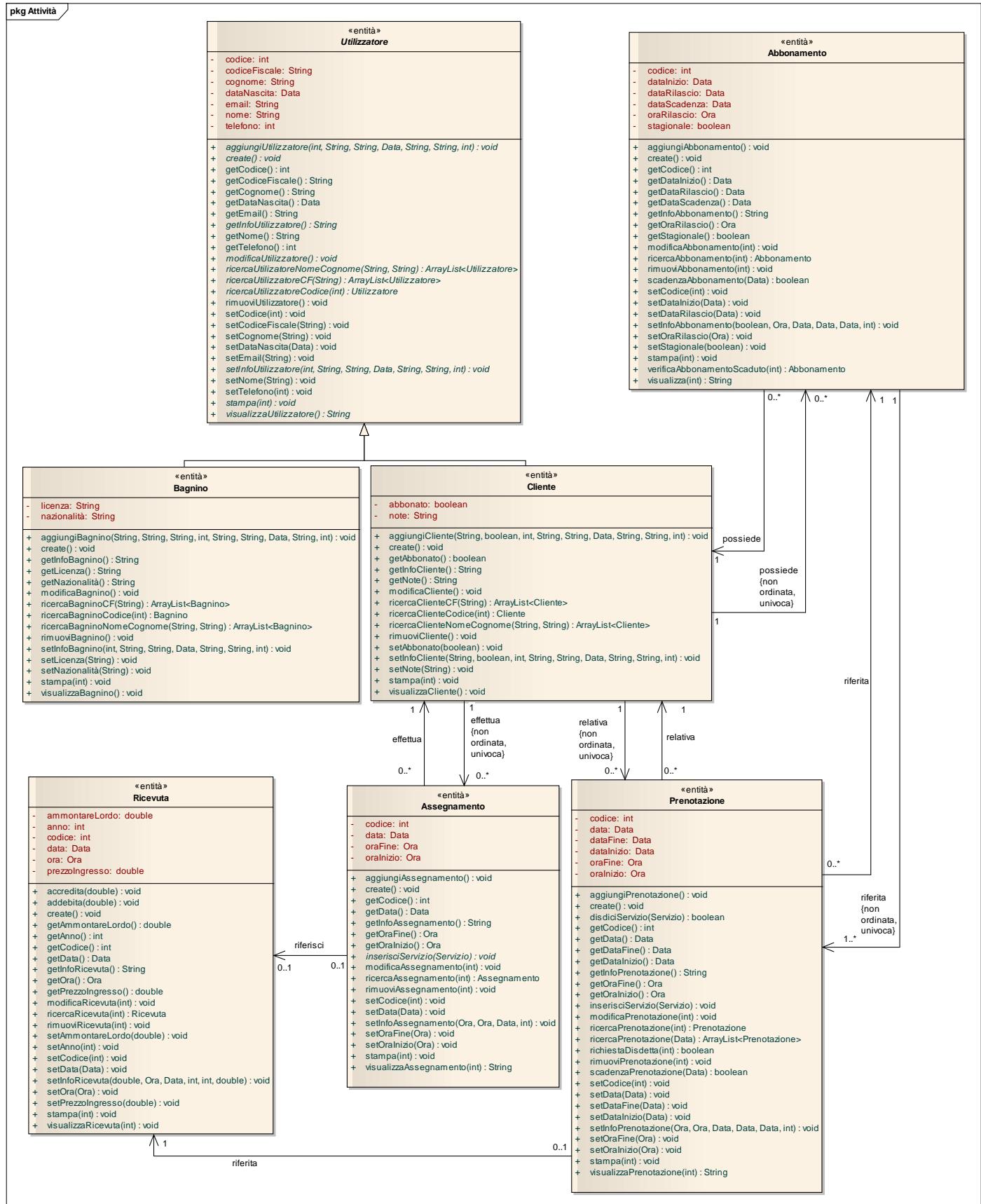


ATTIVITÀ

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Classi di progettazione



Abbonamento

Type: **Class**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> possiede Source -> Destination	Public Abbonamento	Public Cliente	
<u>Association</u> riferita {non ordinata, univoca} Source -> Destination	Public Abbonamento	Public Prenotazione	
<u>Association</u> possiede {non ordinata, univoca} Source -> Destination	Public Cliente	Public Abbonamento	
<u>Association</u> riferita Source -> Destination	Public Prenotazione	Public Abbonamento	

Attributes

Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
datalnizio Data Private		<i>Default:</i>
dataRilascio Data Private		<i>Default:</i>
oraRilascio Ora Private		<i>Default:</i>
stagionale boolean Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiAbbonamento() void Public		
Static create() void Public		
getCodice() int Public		
getDataInizio() Data Public		

getDataRilascio() Data Public		
getDataScadenza() Data Public		
getInfoAbbonamento() String Public		
getOraRilascio() Ora Public		
getStagionale() boolean Public		
Static modificaAbbonamento() void Public		<u>int</u> [in] codice
Static ricercaAbbonamento() Abbonamento Public		<u>int</u> [in] codice
Static rimuoviAbbonamento() void Public		<u>int</u> [in] codice
Static scadenzaAbbonamento() boolean Public		<u>Data</u> [in] dataCorrente
setCodice() void Public		<u>int</u> [in] codice
setDataInizio() void Public		<u>Data</u> [in] dataInizio
setDataRilascio() void Public		<u>Data</u> [in] dataRilascio
setInfoAbbonamento() void Public		<u>boolean</u> [in] stagionale <u>Ora</u> [in] oraRilascio <u>Data</u> [in] dataScadenza <u>Data</u> [in] dataRilascio <u>Data</u> [in] dataInizio <u>int</u> [in] codice
setOraRilascio() void Public		<u>Ora</u> [in] oraRilascio
setStagionale() void Public		<u>boolean</u> [in] stagionale
Static stampa() void Public		<u>int</u> [in] codice
Static verificaAbbonamentoScaduto() Abbonamento Public		<u>int</u> [in] codice
Static visualizza() String Public		<u>int</u> [in] codice

Assegnamento

Type: **Class**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> riferisci Source -> Destination	Public Assegnamento	Public Ricevuta	
<u>Association</u> effettua {non ordinata, univoca} Source -> Destination	Public Cliente	Public Assegnamento	
<u>Association</u> effettua Source -> Destination	Public Assegnamento	Public Cliente	

Attributes

Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
data Data Private		<i>Default:</i>
oraFine Ora Private		<i>Default:</i>
oraInizio Ora Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiAssegnamento() void Public		
Static create() void Public		
getCodice() int Public		
getData() Data Public		
 getInfoAssegnamento() String Public		
getOraFine() Ora Public		
getOraInizio() Ora Public		
inserisciOre() void Public		int [in] ore
Static modificaAssegnamento() void Public		int [in] codice
Static ricercaAssegnamento() Assegnamento Public		int [in] codice
Static rimuoviAssegnamento() void Public		int [in] codice
setCodice() void Public		int [in] codice
setData() void Public		Data [in] data
setInfoAssegnamento() void Public		Ora [in] oraInizio Ora [in] oraFine Data [in] data int [in] codice
setOraFine() void Public		Ora [in] oraFine
setOraInizio() void Public		Ora [in] oraInizio
Static stampa() void Public		int [in] codice
Static visualizzaAssegnamento() String Public		int [in] codice

Bagnino

Type: **Class Utilizzatore**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Bagnino	Public Utilizzatore	

Attributes

Attribute	Notes	Constraints and tags
licenza String Private		<i>Default:</i>
nazionalità String Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiBagnino() void Public		String [in] nazionalità String [in] licenza int [in] telefono String [in] nome String [in] email Data [in] dataNascita String [in] codiceFiscale int [in] codice String [in] cognome
Static create() void Public		
getInfoBagnino() String Public		
getLicenza() String Public		
getNazionalità() String Public		
Static modificaBagnino() void Public		
Static ricercaBagninoCF() ArrayList<Bagnino>		String [in] codiceFiscale

Public		
Static ricercaBagninoCodice() Bagnino Public		int [in] codice
Static ricercaBagninoNomeCo gnome() ArrayList<Bagnino> Public		String [in] cognome String [in] nome
Static rimuoviBagnino() void Public		
setInfoBagnino() void Public		int [in] telefono String [in] nome String [in] email Data [in] dataNascita String [in] cognome String [in] codiceFiscale int [in] codice String [in] licenza String [in] nazionalità
setLicenza() void Public		String [in] licenza
setNazionalità() void Public		String [in] nazionalità
Static stampa() void Public		int [in] codice
Static visualizzaBagnino() void Public		

Cliente

Type: **Class Utilizzatore**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> possiede Source -> Destination	Public Abbonamento	Public Cliente	
<u>Association</u> effettua {non ordinata, univoca} Source -> Destination	Public Cliente	Public Assegnamento	
<u>Association</u> effettua Source -> Destination	Public Assegnamento	Public Cliente	
<u>Association</u> possiede {non ordinata, univoca} Source -> Destination	Public Cliente	Public Abbonamento	
<u>Generalization</u> Source -> Destination	Public Cliente	Public Utilizzatore	
<u>Association</u> relativa {non ordinata, univoca} Source -> Destination	Public Cliente	Public Prenotazione	
<u>Association</u> relativa Source -> Destination	Public Prenotazione	Public Cliente	

Attributes

Attribute	Notes	Constraints and tags
abbonato boolean Private		<i>Default:</i>
note String Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiCliente() void Public		int [in] telefono String [in] nome String [in] email Data [in] dataNascita String [in] cognome String [in] codiceFiscale

		int [in] codice
		String [in] note
		boolean [in] abbonato
Static create() void Public		
getInfoUtilizzatore() String Public		
getAbbonato() boolean Public		
getNote() String Public		
Static modificaCliente() void Public		
Static ricercaClienteCF() ArrayList<Cliente> Public		String [in] codiceFiscale
Static ricercaClienteCodice() Cliente Public		int [in] codice
Static ricercaClienteNomeCognome() ArrayList<Cliente> Public		String [in] cognome String [in] nome
Static rimuoviCliente() void Public		
setAbbonato() void Public		boolean [in] abbonato
setInfoCliente() void Public		String [in] note boolean [in] abbonato int [in] telefono String [in] nome String [in] email Data [in] dataNascita String [in] cognome String [in] codiceFiscale int [in] codice
setNote() void Public		String [in] note
Static stampa() void Public		int [in] codice
Static visualizzaCliente() void Public		

Prenotazione

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> riferita Source -> Destination	Public Prenotazione	Public Ricevuta	
<u>Association</u> riferita {non ordinata, univoca} Source -> Destination	Public Abbonamento	Public Prenotazione	
<u>Association</u> relativa {non ordinata, univoca} Source -> Destination	Public Cliente	Public Prenotazione	
<u>Association</u> relativa Source -> Destination	Public Prenotazione	Public Cliente	
<u>Association</u> riferita Source -> Destination	Public Prenotazione	Public Abbonamento	

Attributes

Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
data Data Private		<i>Default:</i>
dataFine Data Private		<i>Default:</i>
dataInizio Data Private		<i>Default:</i>
oraFine Ora Private		<i>Default:</i>
oraInizio Ora Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiPrenotazione() void Public		
Static create() void Public		
disdiciServizio() boolean Public		Servizio [in] Servizio
getCodice() int Public		
getData() Data Public		
getDataFine() Data Public		
getDataInizio() Data Public		
getInfoPrenotazione() String Public		
getOraFine() Ora Public		
getOraInizio() Ora Public		
inserisciServizio() void Public		Servizio [in] Servizio
Static modificaPrenotazione() void Public		int [in] codice
Static ricercaPrenotazione() Prenotazione Public		int [in] codice
Static ricercaPrenotazione() ArrayList<Prenotazione> Public		Data [in] dataScadenza
Static richiestaDisdetta() boolean Public		int [in] codice
Static rimuoviPrenotazione() void Public		int [in] codice
Static scadenzaPrenotazione() boolean Public		Data [in] dataCorrente
setCodice() void Public		int [in] codice
setData() void Public		Data [in] data
setDataFine() void Public		Data [in] dataFine
setDataInizio() void Public		Data [in] dataInizio
setInfoPrenotazione() void Public		Ora [in] oraInizio Ora [in] oraFine Data [in] dataInizio

		Data [in] dataFine Data [in] data int [in] codice
setOraFine() void Public		Ora [in] oraFine
setOraInizio() void Public		Ora [in] oraInizio
Static stampa() void Public		int [in] codice
Static visualizzaPrenotazione() String Public		int [in] codice

Ricevuta

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Attività

Connections

Connector	Source	Target	Notes
<u>Association</u> riferisci Source -> Destination	Public Assegnamento	Public Ricevuta	
<u>Association</u> riferita Source -> Destination	Public Prenotazione	Public Ricevuta	

Attributes

Attribute	Notes	Constraints and tags
ammontareLordo double Private		<i>Default:</i>
anno int Private		<i>Default:</i>
codice int Private		<i>Default:</i>
data Data Private		<i>Default:</i>
ora Ora Private		<i>Default:</i>
prezzoIngresso double Private		<i>Default:</i>

Operations

Method	Notes	Parameters
accredita() void Public		double [in] importo
addebita() void Public		double [in] prezzo
Static create() void Public		
getAmmontareLordo() double Public		
getAnno() int Public		
getCodice() int Public		
getData() Data Public		

getInfoRicevuta() String Public		
getOra() Ora Public		
getPrezzoIngresso() double Public		
Static modificaRicevuta() void Public		int [in] codice
Static ricercaRicevuta() void Public		int [in] codice
Static rimuoviRicevuta() Ricevuta Public		int [in] codice
setAmmontareLordo() void Public		double [in] ammontareLordo
setAnno() void Public		int [in] anno
setCodice() void Public		int [in] codice
setData() void Public		Data [in] data
setInfoRicevuta() void Public		double [in] prezzoIngresso Ora [in] ora Data [in] data int [in] codice int [in] anno double [in] ammontareLordo
setOra() void Public		Ora [in] ora
setPrezzoIngresso() void Public		double [in] prezzoIngresso
Static stampa() void Public		int [in] codice
Static visualizzaRicevuta() void Public		int [in] codice

Utilizzatore

Type: **Class** (Abstract)
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Attività

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Cliente	Public Utilizzatore	
Generalization Source -> Destination	Public Bagnino	Public Utilizzatore	

Attributes

Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
codiceFiscale String Private		<i>Default:</i>
cognome String Private		<i>Default:</i>
dataNascita Data Private		<i>Default:</i>
email String Private		<i>Default:</i>
nome String Private		<i>Default:</i>
telefono int Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiUtilizzatore() void Public		int [in] telefono String [in] nome String [in] email Data [in] dataNascita String [in] cognome String [in] codiceFiscale int [in] codice
Static create() void Public		
getCodice() int Public		
getCodiceFiscale() String Public		
getCognome() String Public		
getDataNascita() Data Public		
getEmail() String Public		
getInfoUtilizzatore() String Public		
getNome() String Public		
getTelefono() int Public		
Static modificaUtilizzatore() void Public		
Static ricercaUtilizzatoreNomeCognome() ArrayList<Utilizzatore> Public		String [in] cognome String [in] nome
Static ricercaUtilizzatoreCF() ArrayList<Utilizzatore> Public		String [in] codiceFiscale
Static ricercaUtilizzatoreCodice() Utilizzatore Public		int [in] codice
Static rimuoviUtilizzatore() void Public		
setCodice() void Public		int [in] codice
setCodiceFiscale() void Public		String [in] codiceFiscale
setCognome() void Public		String [in] cognome
setDataNascita() void Public		Data [in] dataNascita

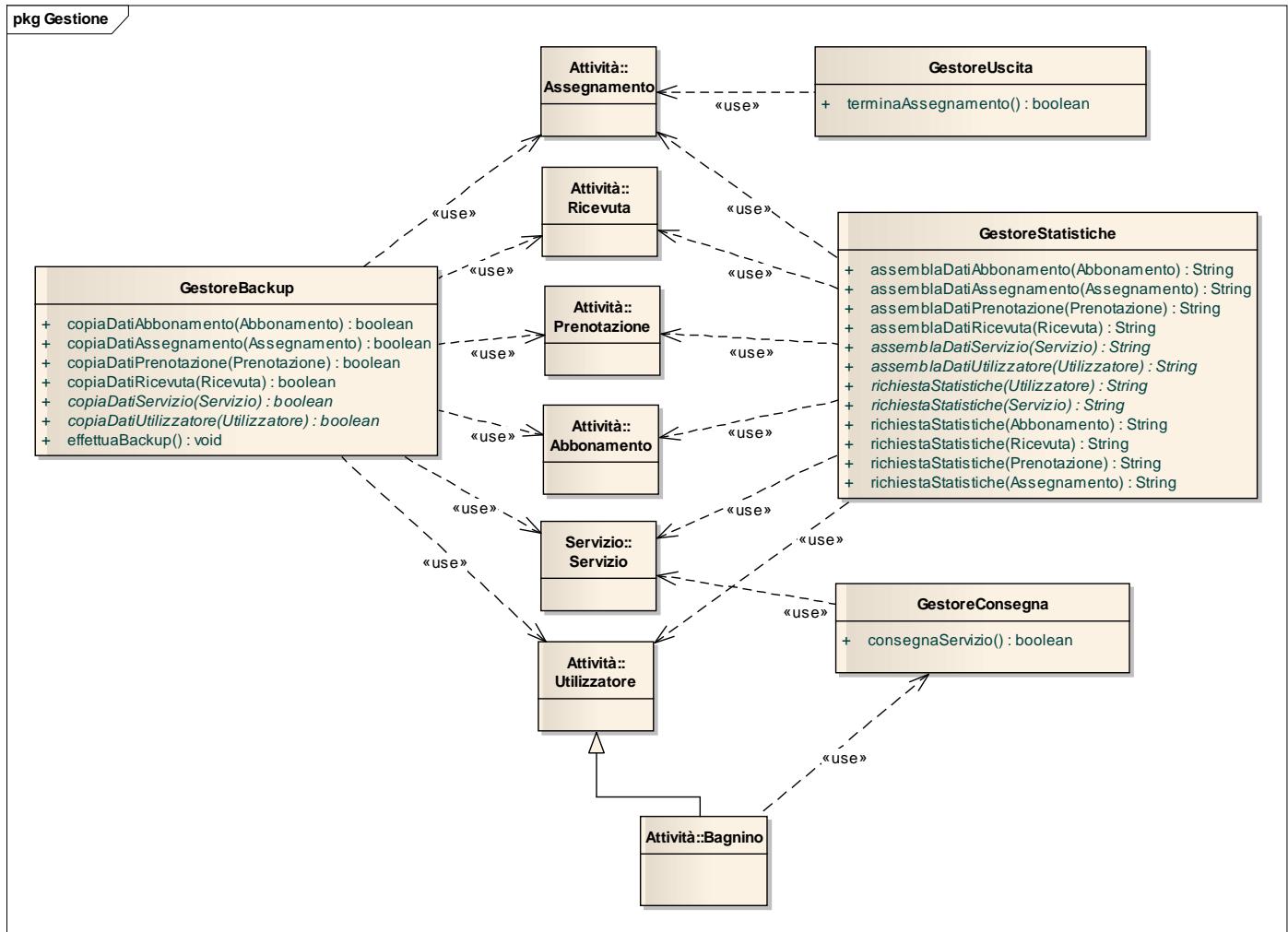
setEmail() void Public		String [in] email
setInfoUtilizzatore() void Public		int [in] telefono String [in] nome String [in] email Data [in] dataNascita String [in] cognome String [in] codiceFiscale int [in] codice
setNome() void Public		String [in] nome
setTelefono() void Public		int [in] telefono
Static stampa() void Public		int [in] codice
Static visualizzaUtilizzatore() String Public		

GESTIONE

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Classi di progettazione



GestoreBackup

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione

Connections

Connector	Source	Target	Notes
Dependency Source -> Destination	Public GestoreBackup	Public Attività::Prenotazione	
Dependency Source -> Destination	Public GestoreBackup	Public Servizio::Servizio	
Dependency Source -> Destination	Public GestoreBackup	Public Attività::Ricevuta	
Dependency Source -> Destination	Public GestoreBackup	Public Attività::Abbonamento	
Dependency Source -> Destination	Public GestoreBackup	Public Attività::Utilizzatore	
Dependency Source -> Destination	Public GestoreBackup	Public Attività::Assegnamento	

Operations

Method	Notes	Parameters
Static copiaDatiAbbonamento() boolean Public		Abbonamento [in] Abbonamento
Static copiaDatiAssegnamento() boolean Public		Assegnamento [in] Assegnamento
Static copiaDatiPrenotazione() boolean Public		Prenotazione [in] Prenotazione
Static copiaDatiRicevuta() boolean Public		Ricevuta [in] Ricevuta
Static copiaDatiServizio() boolean Public		Servizio [in] Servizio
Static copiaDatiUtilizzatore() boolean Public		Utilizzatore [in] Utilizzatore
Static effettuaBackup() void Public		

GestoreConsegna

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Gestione

Connections

Connector	Source	Target	Notes
Dependency Source -> Destination	Public GestoreConsegna	Public Servizio::Servizio	
Dependency Source -> Destination	Public Attività::Bagnino	Public GestoreConsegna	

Operations

Method	Notes	Parameters
Static consegnaServizio() boolean Public		

GestoreStatistiche

Type:

Class

Status:

Proposed. Version 1.0. Phase 1.0.

Package:

Gestione

Connections

Connector	Source	Target	Notes
Dependency Source -> Destination	Public GestoreStatistiche	Public Attività::Utilizzatore	
Dependency Source -> Destination	Public GestoreStatistiche	Public Attività::Ricevuta	
Dependency Source -> Destination	Public GestoreStatistiche	Public Servizio::Servizio	
Dependency Source -> Destination	Public GestoreStatistiche	Public Attività::Assegnamento	
Dependency Source -> Destination	Public GestoreStatistiche	Public Attività::Prenotazione	
Dependency Source -> Destination	Public GestoreStatistiche	Public Attività::Abbonamento	

Operations

Method	Notes	Parameters
Static assemblaDatiAbbonamento() String Public		Abbonamento [in] Abbonamento
Static assemblaDatiAssegnamento() String Public		Assegnamento [in] Assegnamento
Static assemblaDatiPrenotazione() String Public		Prenotazione [in] Prenotazione
Static assemblaDatiRicevuta() String Public		Ricevuta [in] Ricevuta
Static assemblaDatiServizio() String Public		Servizio [in] Servizio
Static assemblaDatiUtilizzatore() String Public		Utilizzatore [in] Utilizzatore
Static richiestaStatistiche() String Public		Utilizzatore [in] Utilizzatore
Static		Servizio [in] Servizio

richiestaStatistiche() String Public		
Static richiestaStatistiche() String Public		Abbonamento [in] Abbonamento
Static richiestaStatistiche() String Public		Ricevuta [in] Ricevuta
Static richiestaStatistiche() String Public		Prenotazione [in] Prenotazione
Static richiestaStatistiche() String Public		Assegnamento [in] Assegnamento

GestoreUscita

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package:

Connections

Connector	Source	Target	Notes
Dependency Source -> Destination	Public GestoreUscita	Public Attività::Assegnamento	

Operations

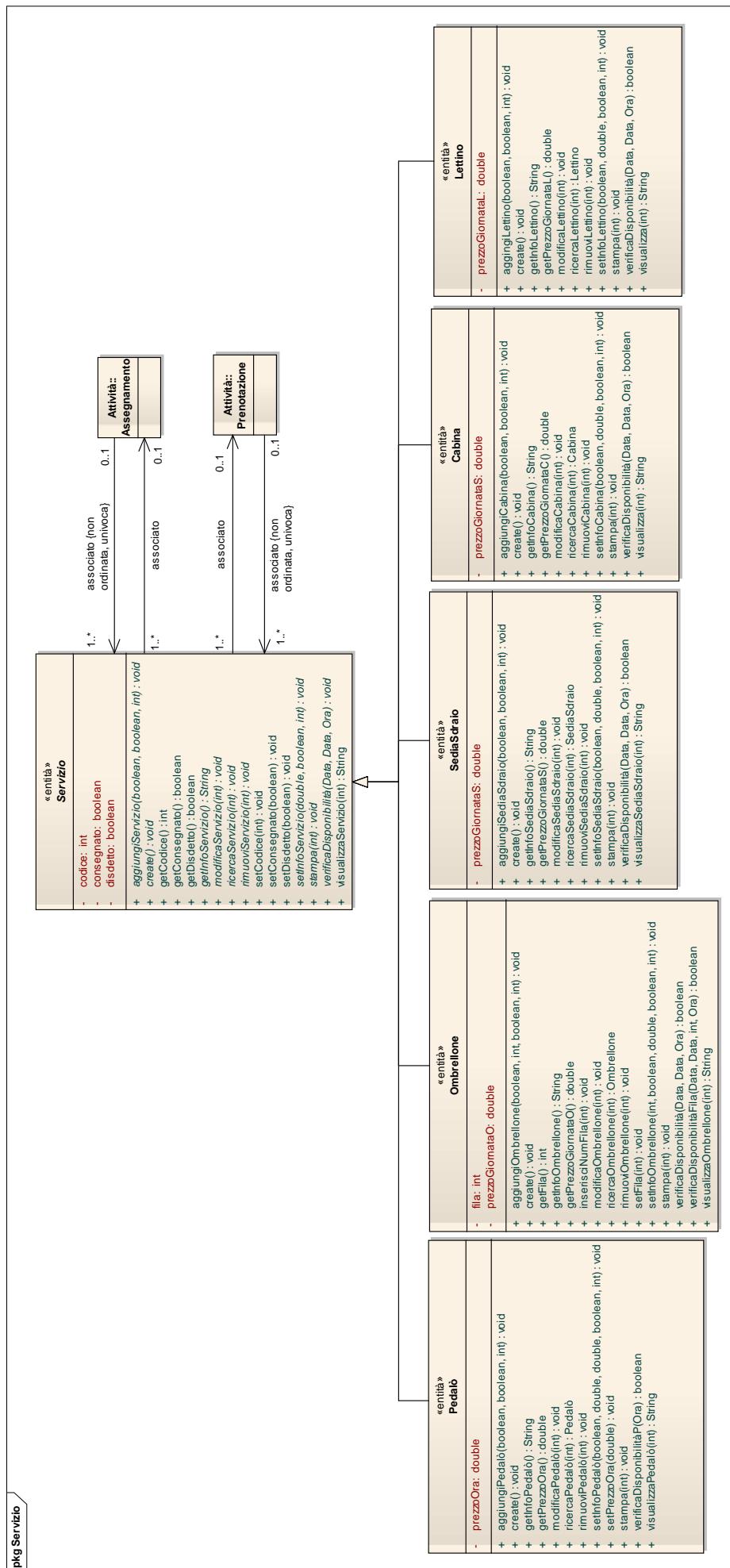
Method	Notes	Parameters
Static terminaAssegnamento() boolean Public		

SERVIZIO

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Classi di progettazione



Cabina

Type: **Class Servizio**

Status: Proposed. Version 1.0. Phase 1.0.

Package:

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Cabina	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
prezzoGiornataS double Private Static		<i>Default:</i>

Operations

Method	Notes	Parameters
aggiungiCabina() void Public		<u>boolean</u> [in] <u>dissetto</u> <u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
Static create() void Public		
getInfoCabina() String Public		
Static getPrezzoGiornataC() double Public		
modificaCabina() void Public		<u>int</u> [in] <u>codice</u>
Static ricercaCabina() Cabina Public		<u>int</u> [in] <u>codice</u>
rimuoviCabina() void Public		<u>int</u> [in] <u>codice</u>
setInfoCabina() void Public		<u>boolean</u> [in] <u>dissetto</u> <u>double</u> [in] <u>prezzoGiornataC</u> <u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
Static stampa() void Public		<u>int</u> [in] <u>codice</u>
Static verificaDisponibilità() boolean Public		<u>Data</u> [in] <u>dataFine</u> <u>Data</u> [in] <u>dataInizio</u> <u>Ora</u> [in] <u>oraInizio</u>
Static visualizza() String Public		<u>int</u> [in] <u>codice</u>

Lettino

Type: **Class Servizio**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Servizio

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Lettino	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
prezzoGiornataL double Private Static		<i>Default:</i>

Operations

Method	Notes	Parameters
aggiungiLettino() void Public		<u>boolean</u> [in] <u>dissetto</u> <u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
Static create() void Public		
getInfoLettino() String Public		
Static getPrezzoGiornataL() double Public		
modificaLettino() void Public		<u>int</u> [in] <u>codice</u>
Static ricercaLettino() Lettino Public		<u>int</u> [in] <u>codice</u>
rimuoviLettino() void Public		<u>int</u> [in] <u>codice</u>
setInfoLettino() void Public		<u>boolean</u> [in] <u>dissetto</u> <u>double</u> [in] <u>prezzoGiornataL</u>

		<u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
Static stampa() void Public		<u>int</u> [in] <u>codice</u>
Static verificaDisponibilità() boolean Public		<u>Data</u> [in] <u>dataFine</u> <u>Data</u> [in] <u>dataInizio</u> <u>Ora</u> [in] <u>oraInizio</u>
Static visualizza() String Public		<u>int</u> [in] <u>codice</u>

Ombrellone

Type: **Class Servizio**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Servizio

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Ombrellone	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
fila int Private		<i>Default:</i>
prezzoGiornataO double Private Static		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiOmbrellone() void Public		boolean [in] <u>disdetto</u> int [in] <u>fila</u> boolean [in] <u>consegnato</u> int [in] <u>codice</u>
Static create() void Public		
getFila() int Public		
 getInfoOmbrellone() String Public		
Static getPrezzoGiornataO() double Public		
inserisciNumFila() void Public		int [in] <u>numFila</u>
Static modificaOmbrellone() void Public		int [in] <u>codice</u>
Static ricercaOmbrellone() Ombrellone Public		int [in] <u>codice</u>
Static rimuoviOmbrellone() void Public		int [in] <u>codice</u>

setFila() void Public		<u>int</u> [in] <u>fila</u>
setInfoOmbrellone() void Public		<u>int</u> [in] <u>fila</u> <u>boolean</u> [in] <u>disdetto</u> <u>double</u> [in] <u>prezzoGiornataO</u> <u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
Static stampa() void Public		<u>int</u> [in] <u>codice</u>
Static verificaDisponibilità() boolean Public		<u>Data</u> [in] <u>dataFine</u> <u>Data</u> [in] <u>dataInizio</u> <u>Ora</u> [in] <u>oraInizio</u>
Static verificaDisponibilitàFila 0 boolean Public		<u>Data</u> [in] <u>dataFine</u> <u>Data</u> [in] <u>dataInizio</u> <u>int</u> [in] <u>fila</u> <u>Ora</u> [in] <u>oraInizio</u>
Static visualizzaOmbrellone() String Public		<u>int</u> [in] <u>codice</u>

Pedalò

Type: **Class Servizio**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Servizio

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Pedalò	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
prezzoOra double Private Static		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiPedalò() void Public		<u>boolean</u> [in] <u>disdetto</u> <u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
Static create() void Public		
getInfoPedalò() String Public		
getPrezzoOra() double Public		
Static modificaPedalò() void Public		<u>int</u> [in] <u>codice</u>
Static ricercaPedalò() Pedalò Public		<u>int</u> [in] <u>codice</u>
Static rimuoviPedalò() void Public		<u>int</u> [in] <u>codice</u>
setInfoPedalò() void Public		<u>boolean</u> [in] <u>disdetto</u> <u>double</u> [in] <u>prezzoOra</u> <u>double</u> [in] <u>prezzoGiornata</u> <u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
setPrezzoOra() void Public		<u>double</u> [in] <u>prezzoOra</u>

Static stampa() void Public		<u>int</u> [in] codice
Static verificaDisponibilitàP() boolean Public		<u>Ora</u> [in] oraInizio
Static visualizzaPedalò() String Public		<u>int</u> [in] codice

SediaSdraio

Type: **Class Servizio**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Servizio

Connections

Connector	Source	Target	Notes
<u>Generalization</u> Source -> Destination	Public SediaSdraio	Public Servizio	

Attributes

Attribute	Notes	Constraints and tags
prezzoGiornataS double Private Static		<i>Default:</i>

Operations

Method	Notes	Parameters
Static aggiungiSediaSdraio() void Public		<u>boolean</u> [in] disdetto <u>boolean</u> [in] consegnato <u>int</u> [in] codice
Static create() void Public		
getInfoSediaSdraio() String Public		
Static getPrezzoGiornataS() double Public		
Static modificaSediaSdraio() void Public		<u>int</u> [in] codice
Static ricercaSediaSdraio() SediaSdraio Public		<u>int</u> [in] codice
setInfoSediaSdraio() void Public		<u>int</u> [in] numDisponibiliS <u>double</u> [in] prezzoGiornata <u>boolean</u> [in] consegnato <u>int</u> [in] codice

Static stampa() void Public		int [in] codice
Static verificaDisponibilità() boolean Public		Data [in] dataFine Data [in] dataInizio Ora [in] oraInizio
Static visualizzaSediaSdraio() String Public		int [in] codice

Servizio

Type: **Class**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Servizio

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Lettino	Public Servizio	
Association associato {non ordinata, univoca} Source -> Destination	Public Attività::Prenotazione	Public Servizio	
Association associato Source -> Destination	Public Servizio	Public Attività::Prenotazione	
Association associato {non ordinata, univoca} Source -> Destination	Public Attività::Assegnamento	Public Servizio	
Generalization Source -> Destination	Public Pedalò	Public Servizio	
Generalization Source -> Destination	Public SediaSdraio	Public Servizio	
Association associato Source -> Destination	Public Servizio	Public Attività::Assegnamento	
Generalization Source -> Destination	Public Cabina	Public Servizio	
Generalization Source -> Destination	Public Ombrellone	Public Servizio	

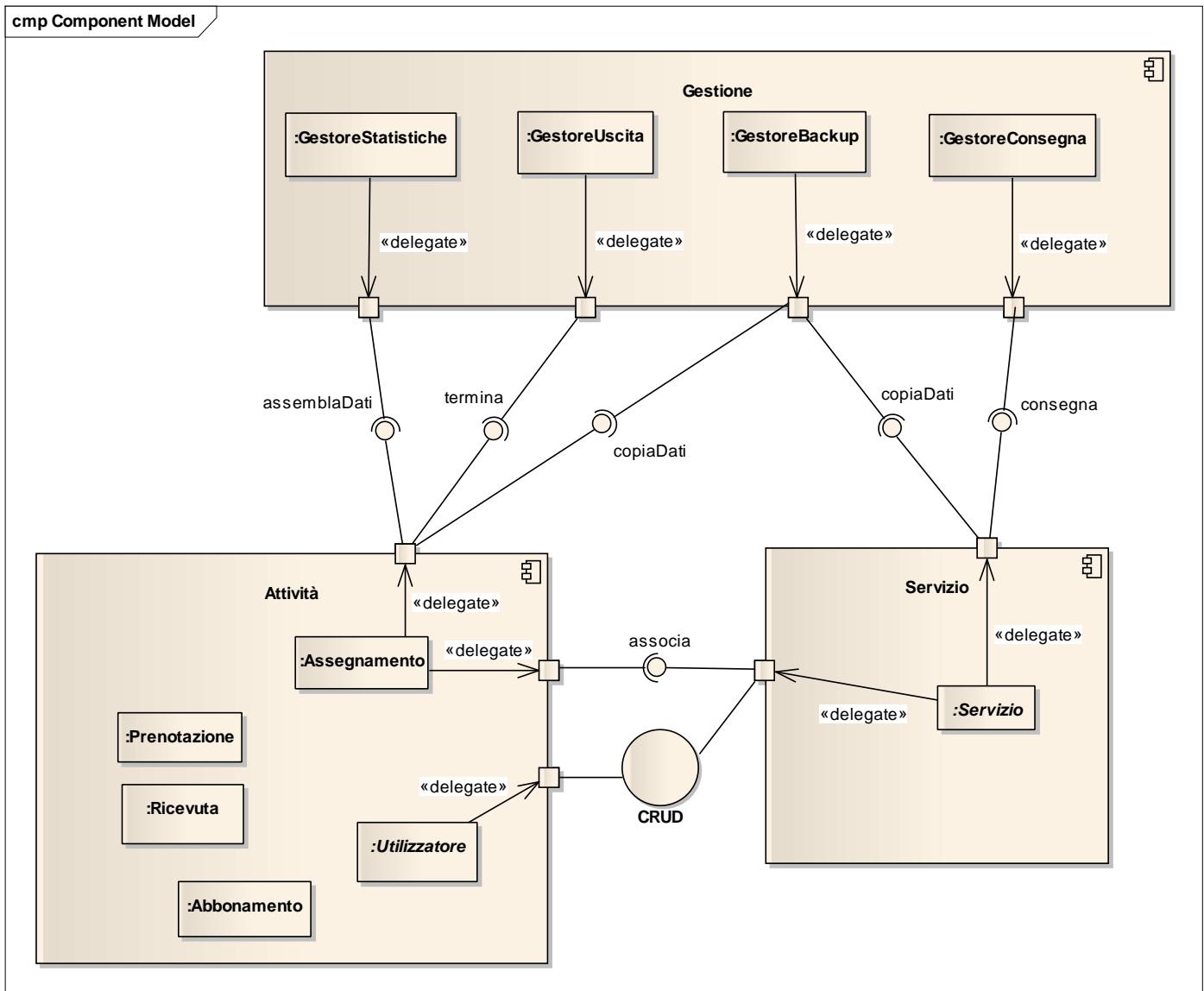
Attributes

Attribute	Notes	Constraints and tags
codice int Private		<i>Default:</i>
consegnato boolean Private		<i>Default:</i>
disdetto boolean Private		<i>Default</i>

Operations

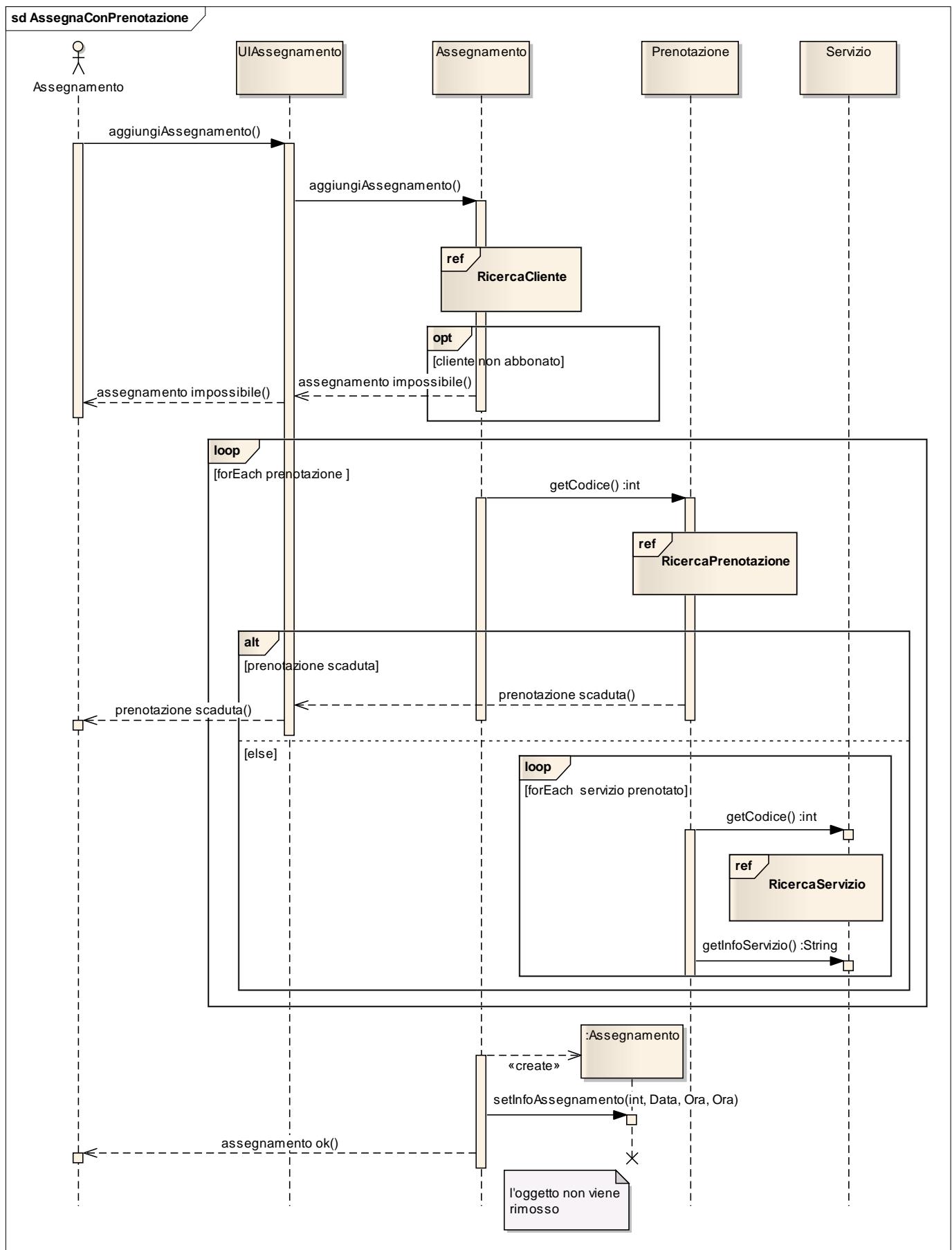
Method	Notes	Parameters
Static aggiungiServizio() void Public		<u>boolean</u> [in] <u>disdetto</u> <u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
Static create() void Public		
getCodice() int Public		
getConsegnato() boolean Public		
getDisdetto() boolean Public		
getInfoServizio() String Public		
Static modificaServizio() void Public		<u>int</u> [in] <u>codice</u>
Static ricercaServizio() void Public		<u>int</u> [in] <u>codice</u>
Static rimuoviServizio() void Public		<u>int</u> [in] <u>codice</u>
setCodice() void Public		<u>int</u> [in] <u>codice</u>
setConsegnato() void Public		<u>boolean</u> [in] <u>consegnato</u>
setDisdetto() void Public		<u>boolean</u> [in] <u>disdetto</u>
Static setInfoServizio() void Public		<u>double</u> [in] <u>prezzoGiornata</u> <u>boolean</u> [in] <u>consegnato</u> <u>int</u> [in] <u>codice</u>
Static stampa() void Public		<u>int</u> [in] <u>codice</u>
Static verificaDisponibilità() void Public		<u>Data</u> [in] <u>dataFine</u> <u>Data</u> [in] <u>dataInizio</u> <u>Ora</u> [in] <u>oraInizio</u>
Static visualizzaServizio() String Public		<u>int</u> [in] <u>codice</u>

DIAGRAMMA DEI COMPONENTI

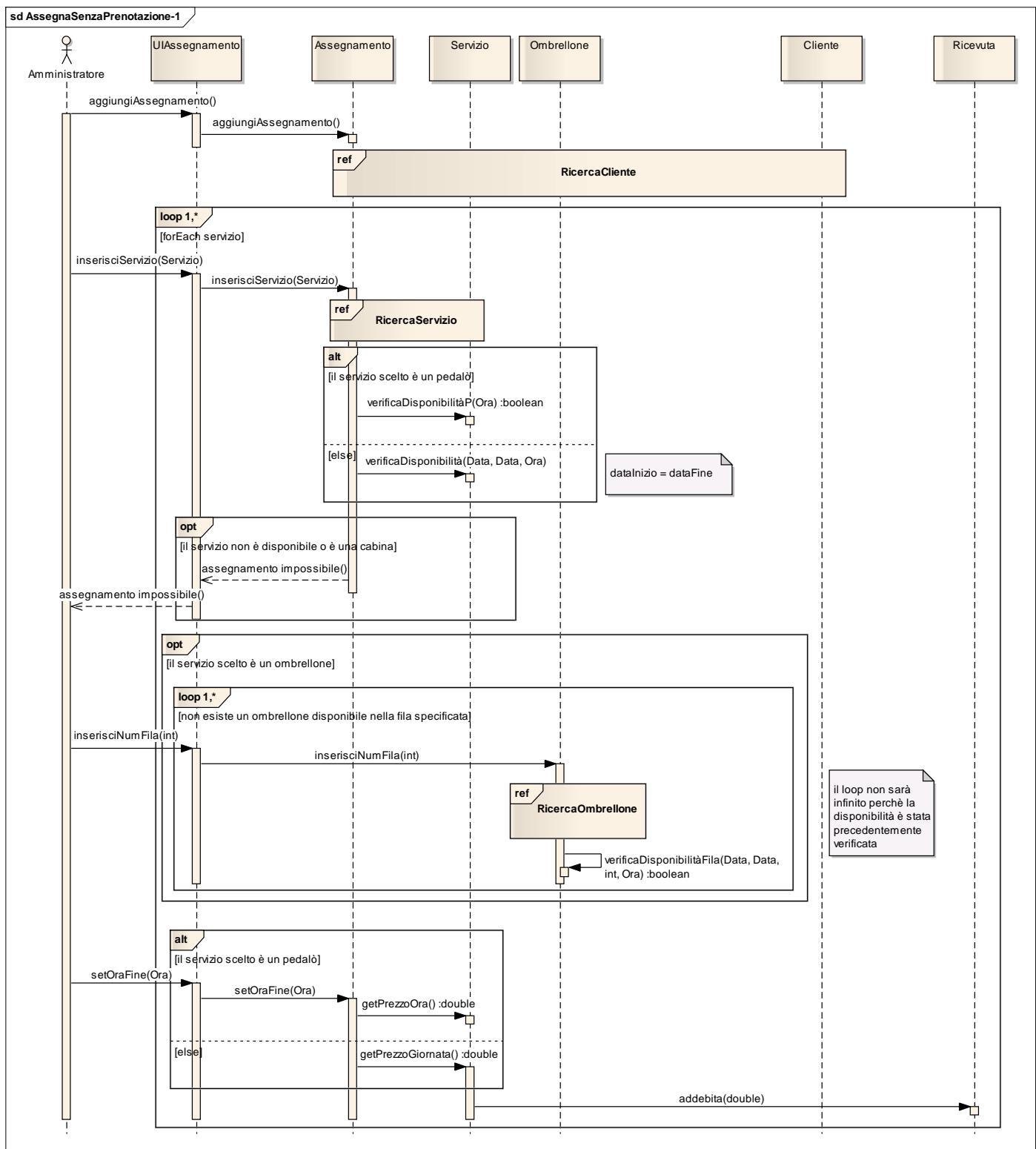


DIAGRAMMI DI SEQUENZA RAFFINATI

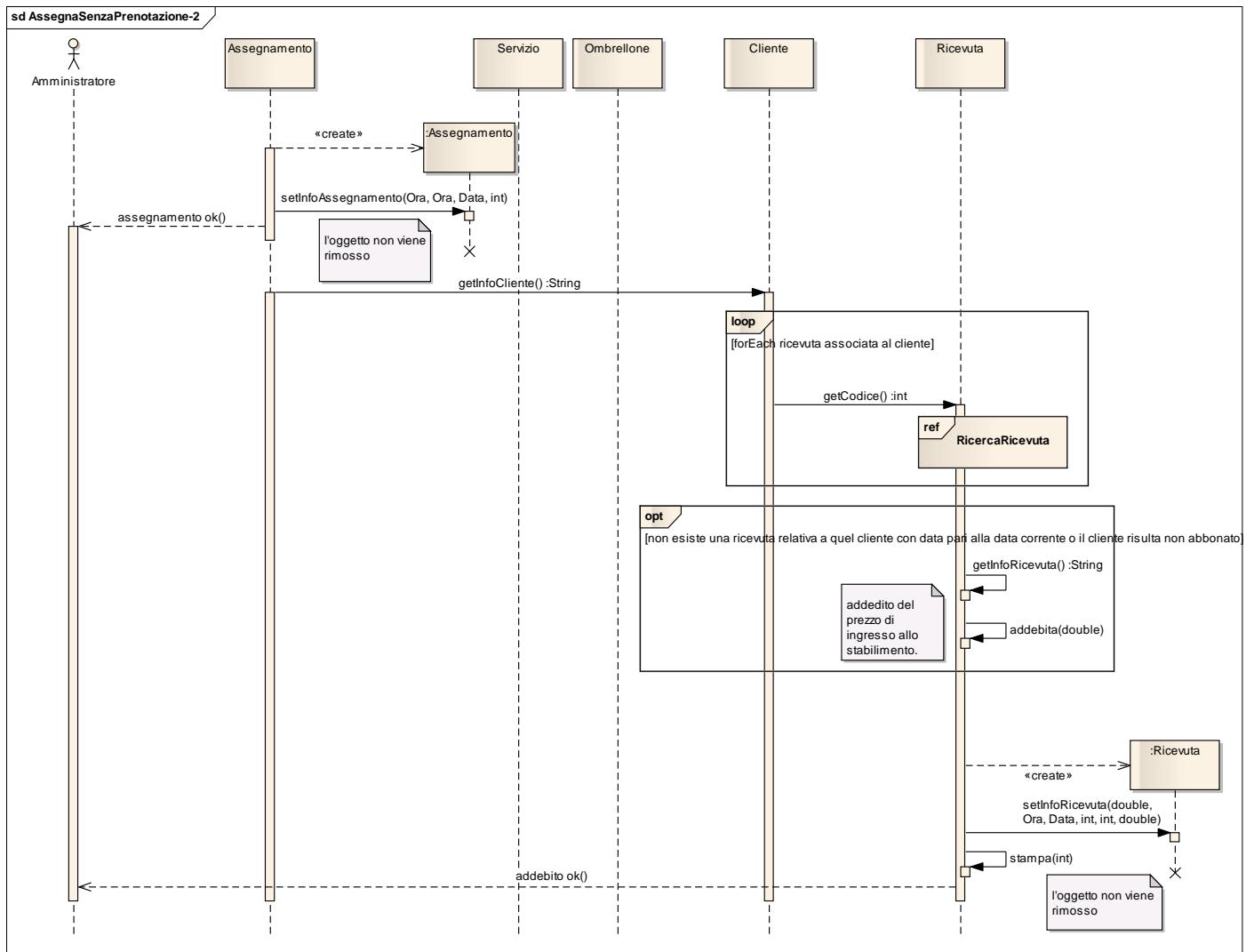
ASSEGNA CON PRENOTAZIONE



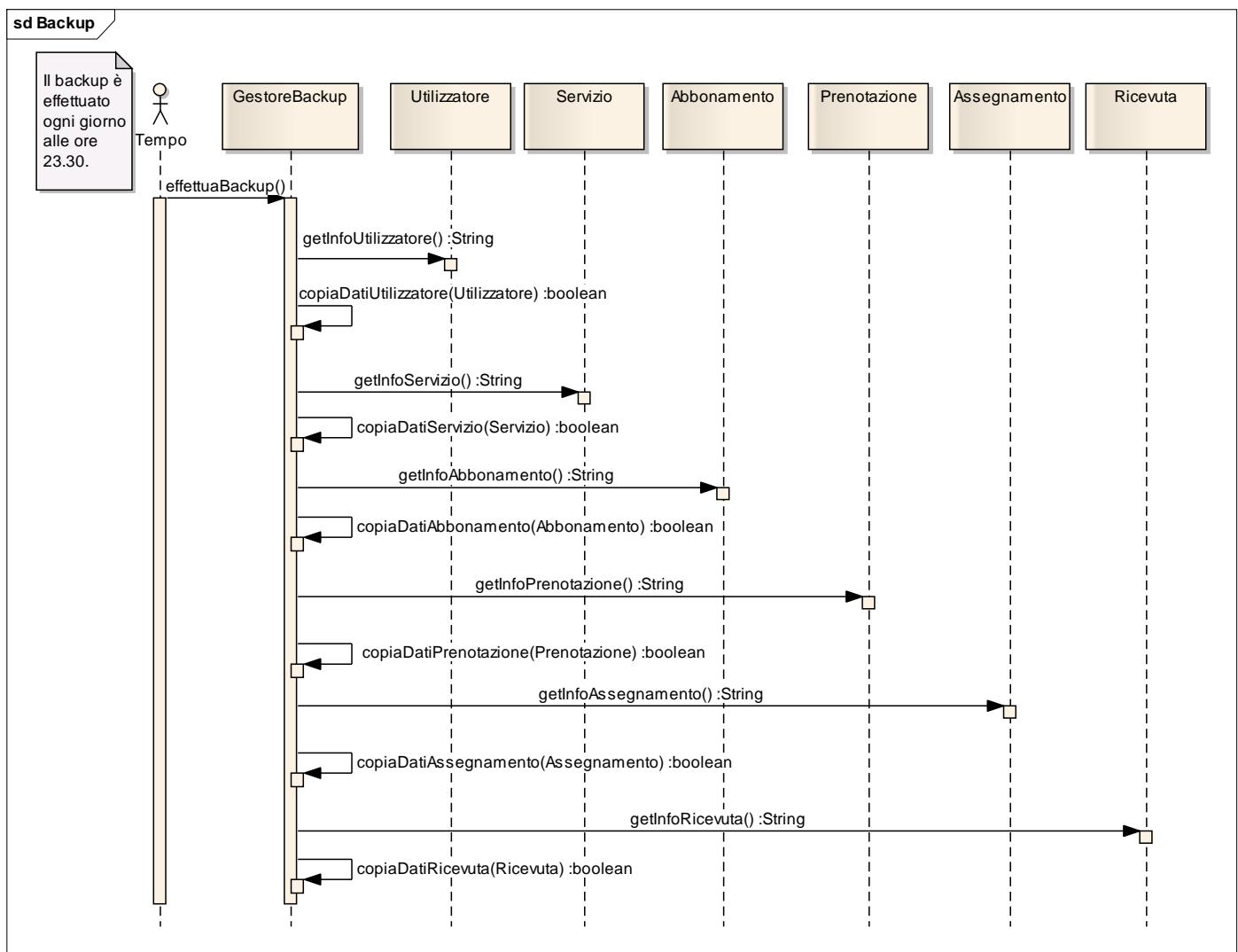
ASSEGNA SENZA PRENOTAZIONE



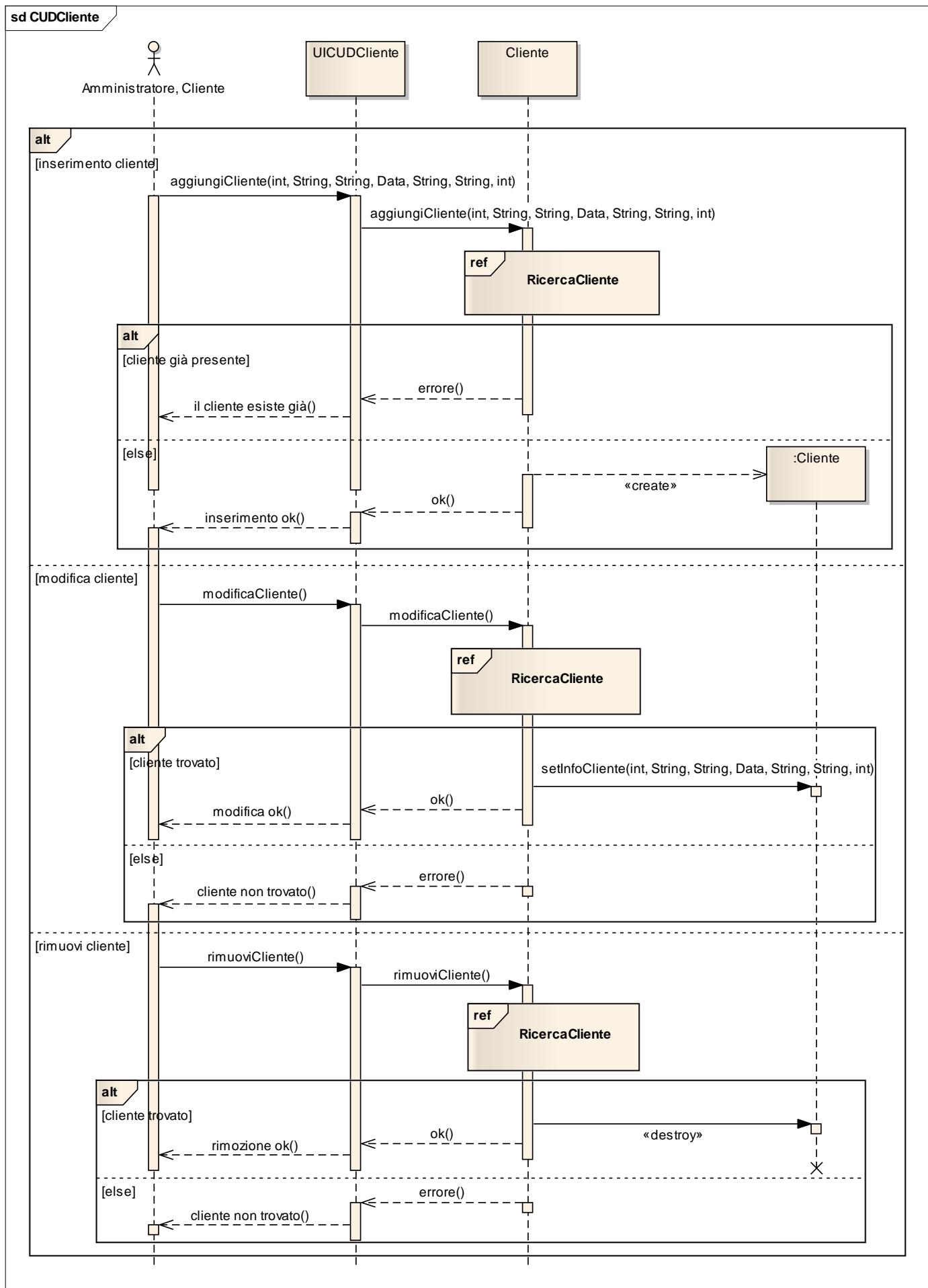
Continua alla pagina successiva



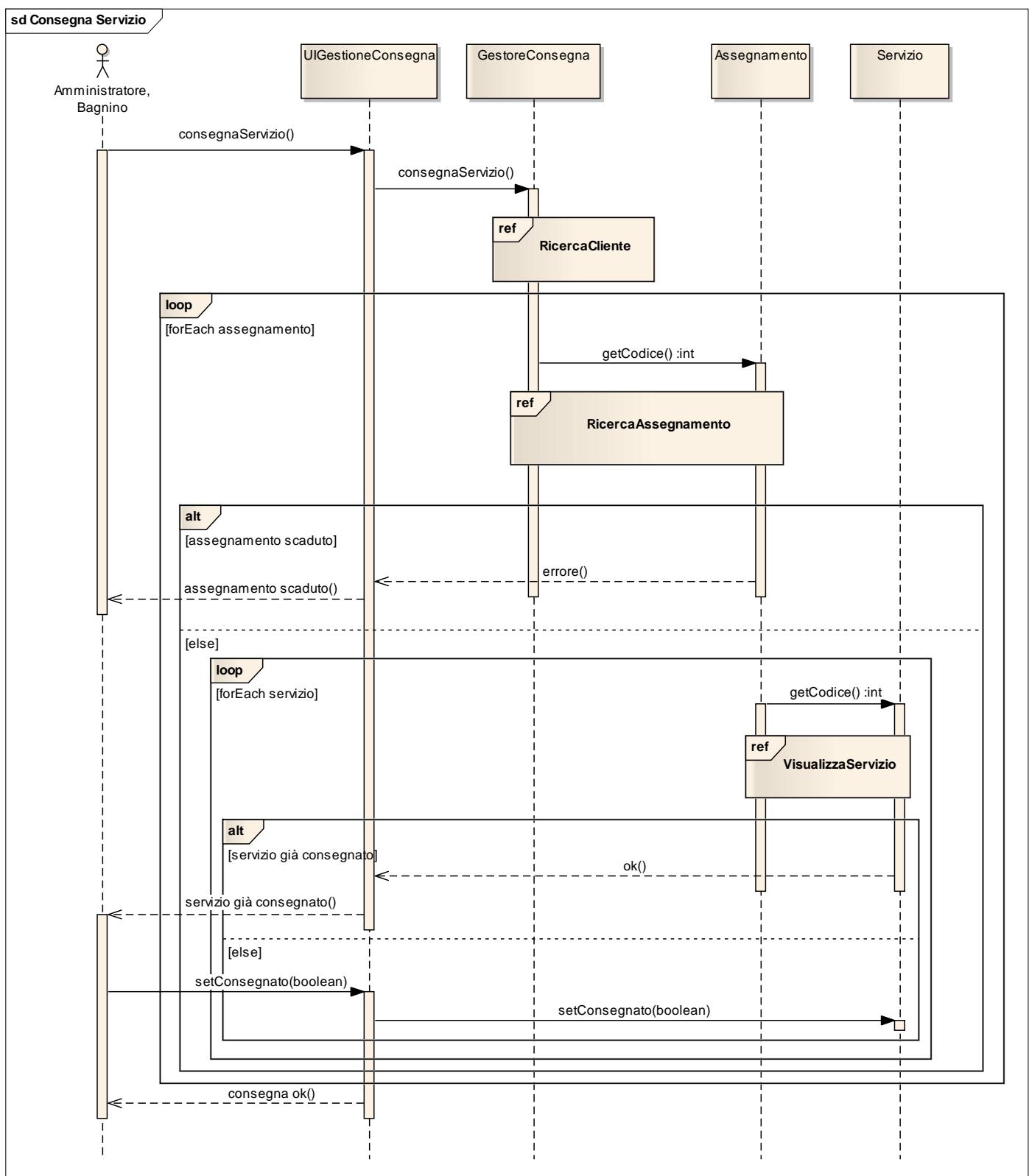
BACKUP



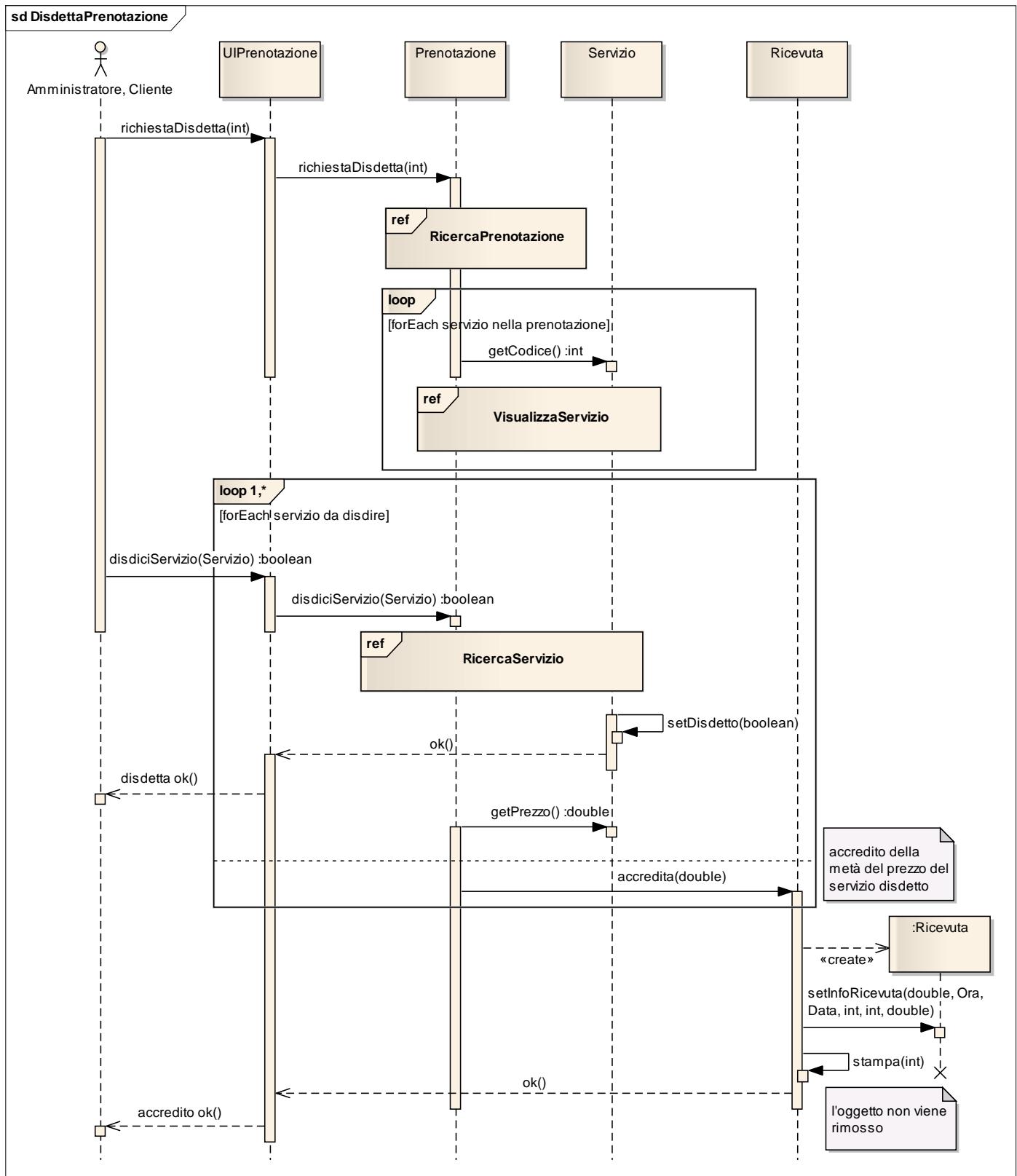
CUD CLIENTE



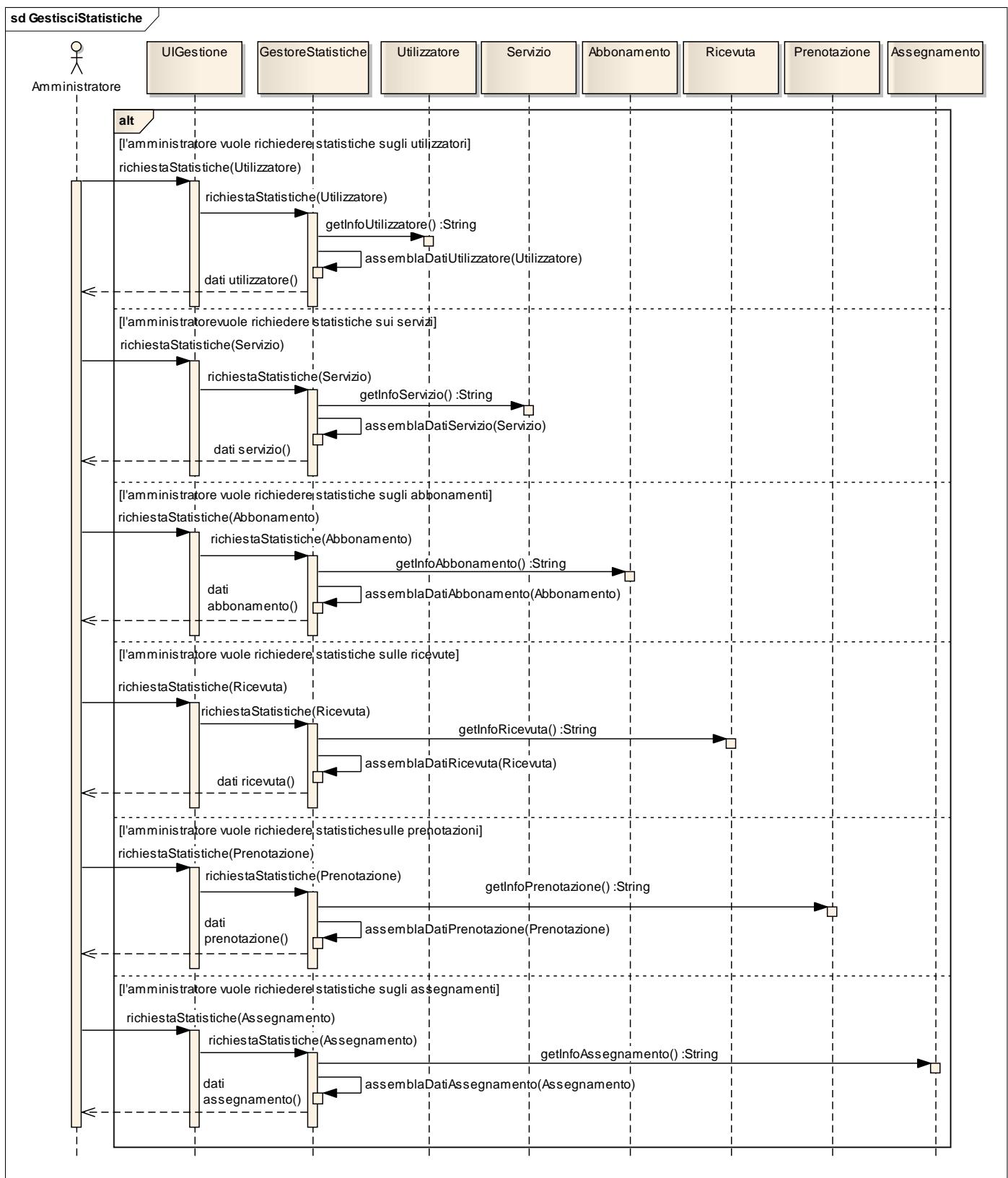
CONSEGNA SERVIZIO



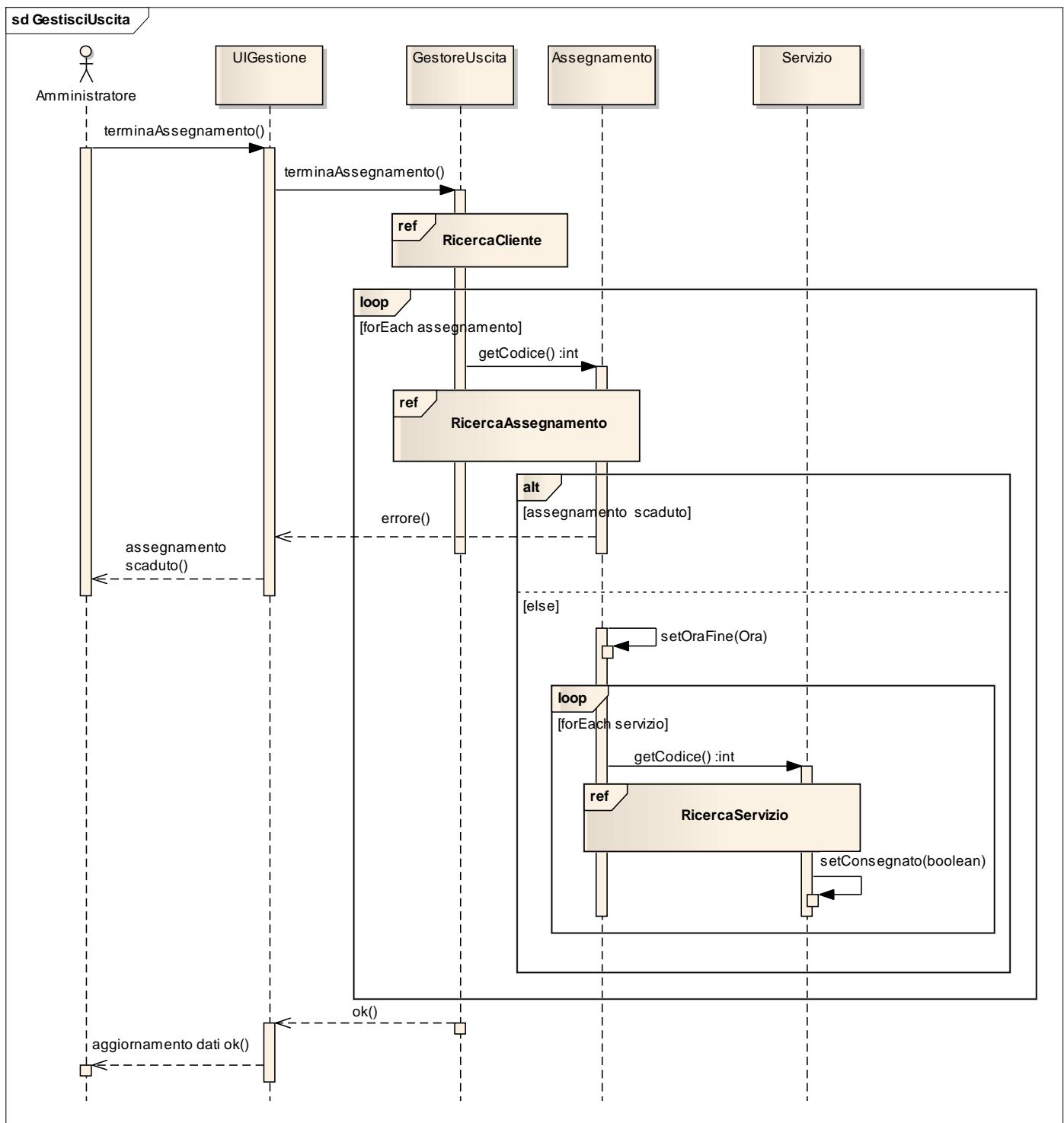
DISDETTA PRENOTAZIONE



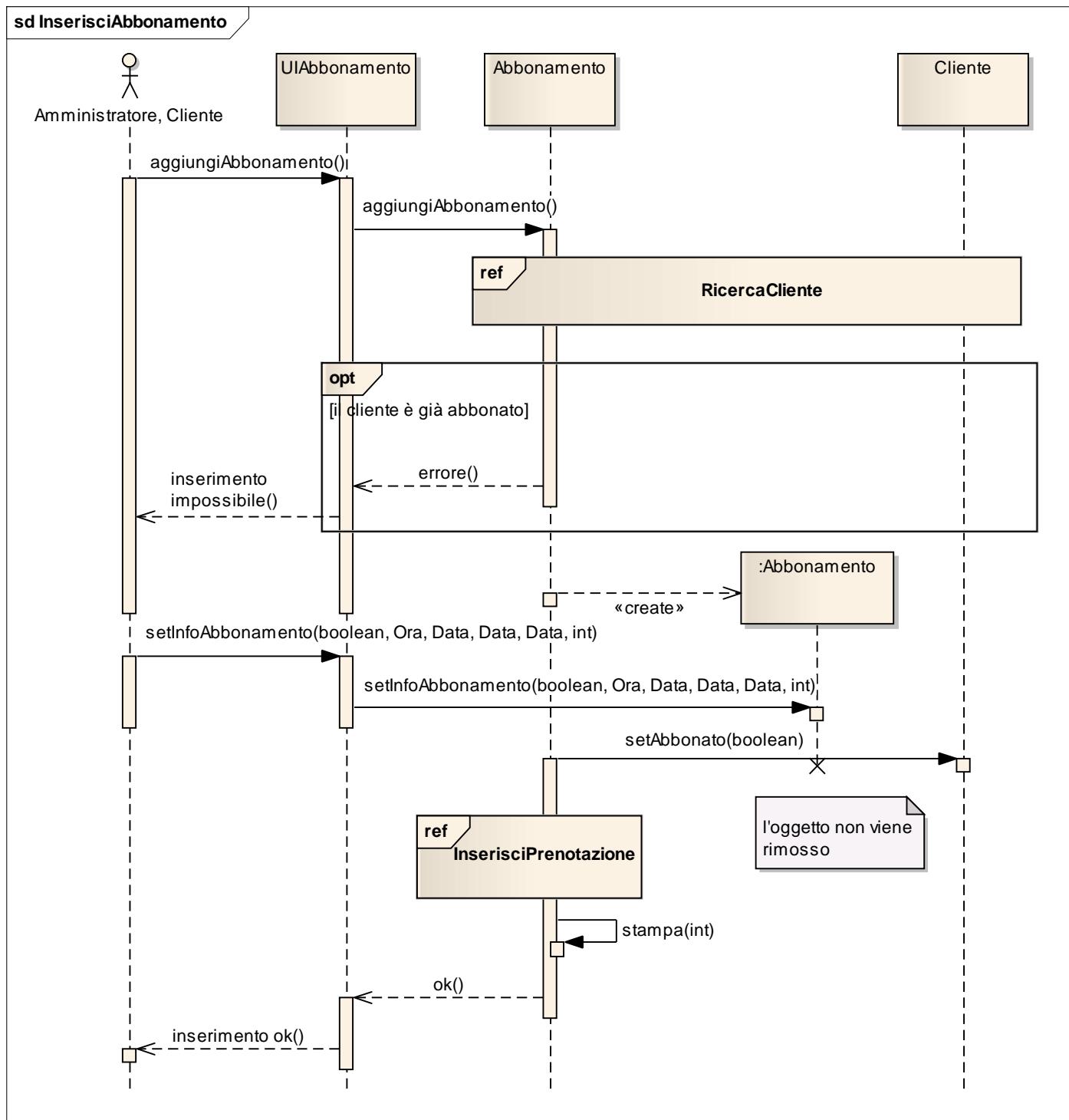
GESTISCI STATISTICHE



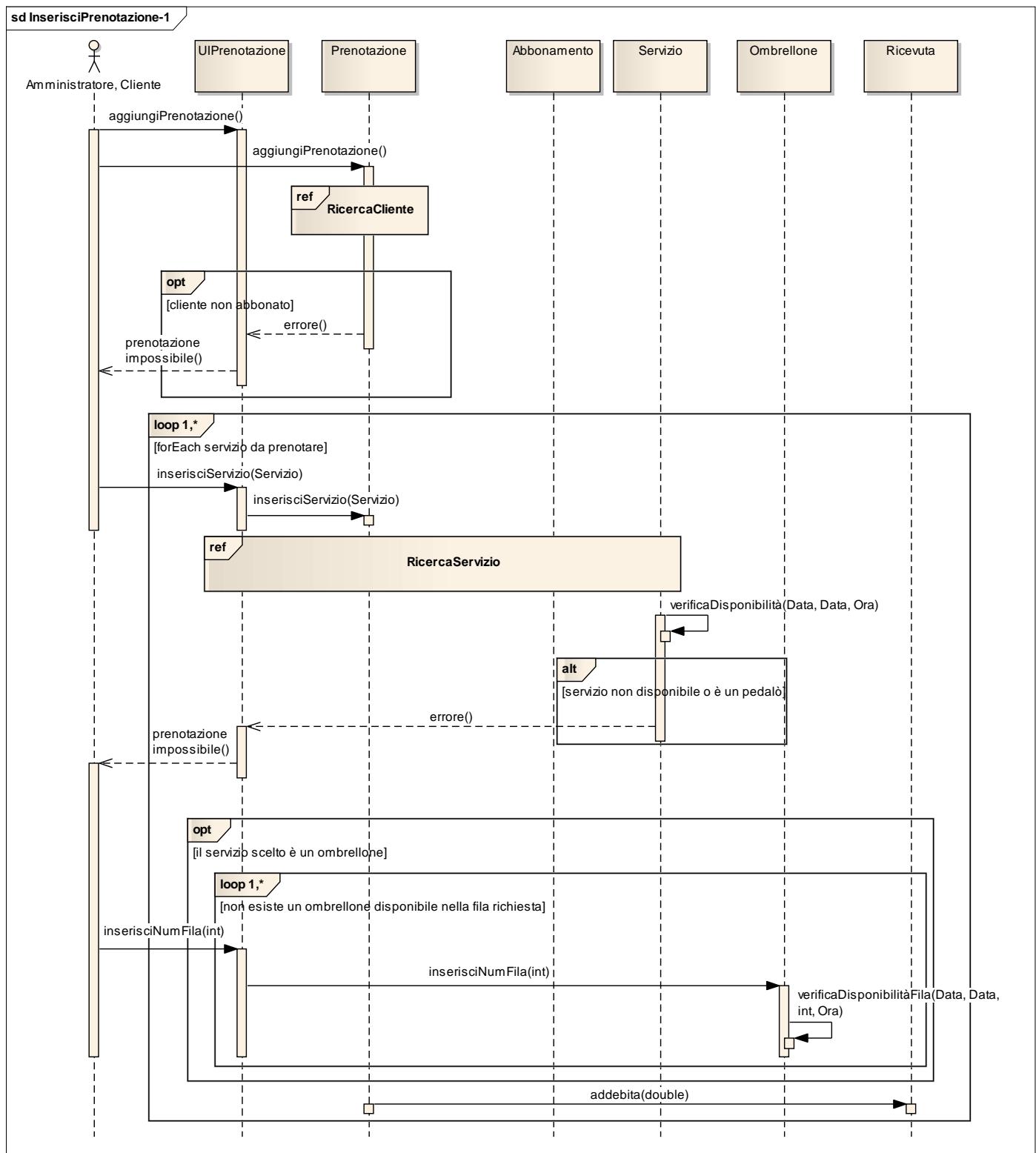
GESTISCI USCITA



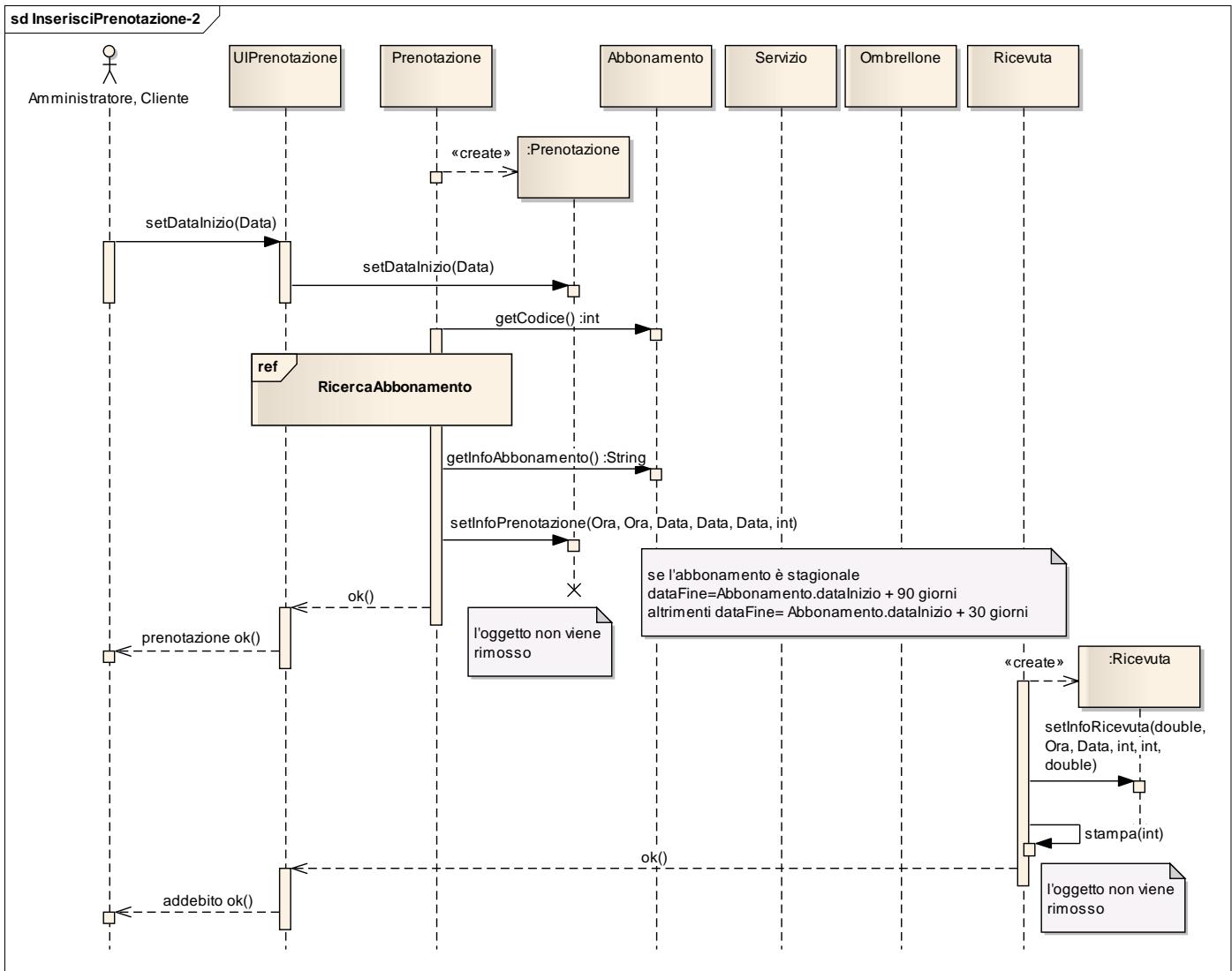
INSERISCI ABBONAMENTO



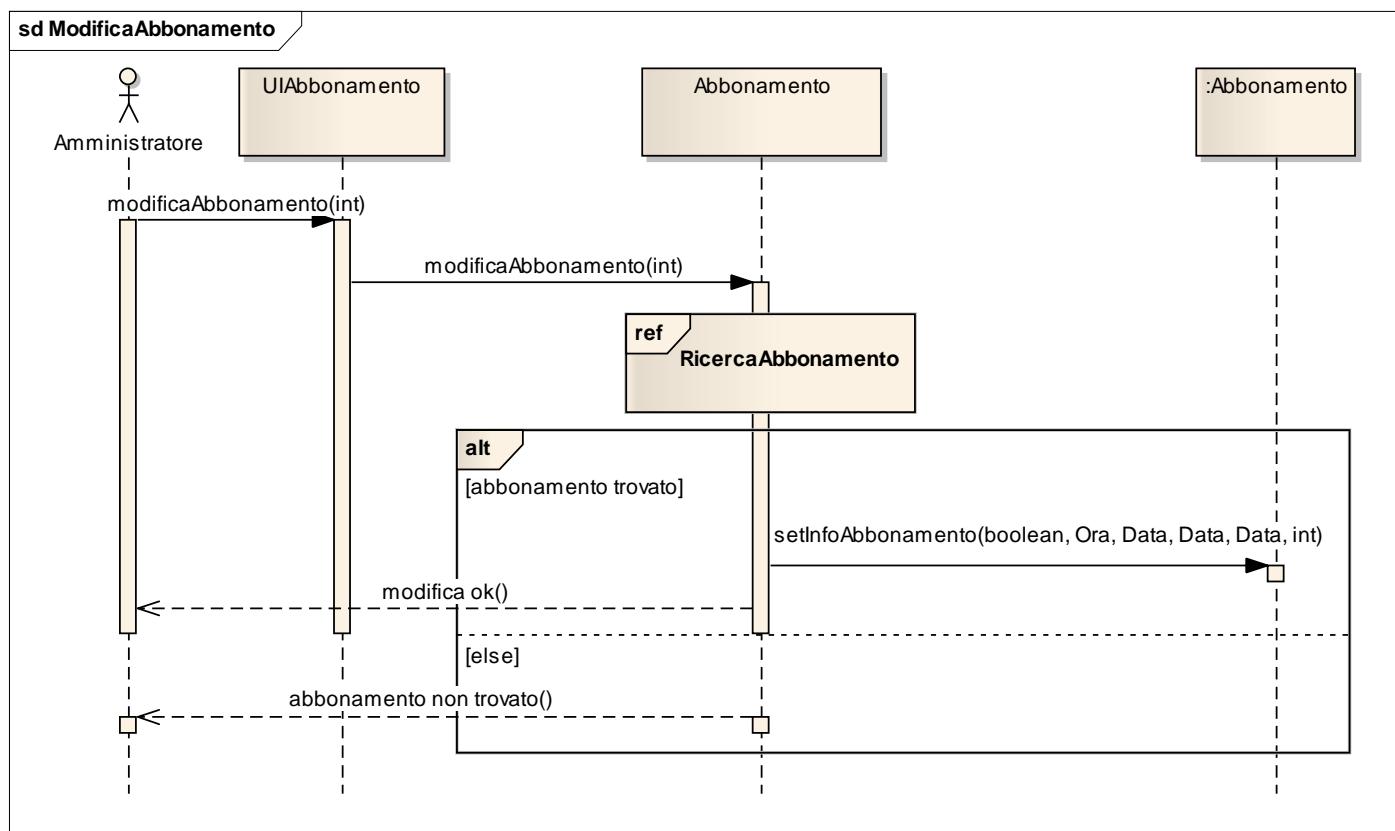
INSERISCI PRENOTAZIONE



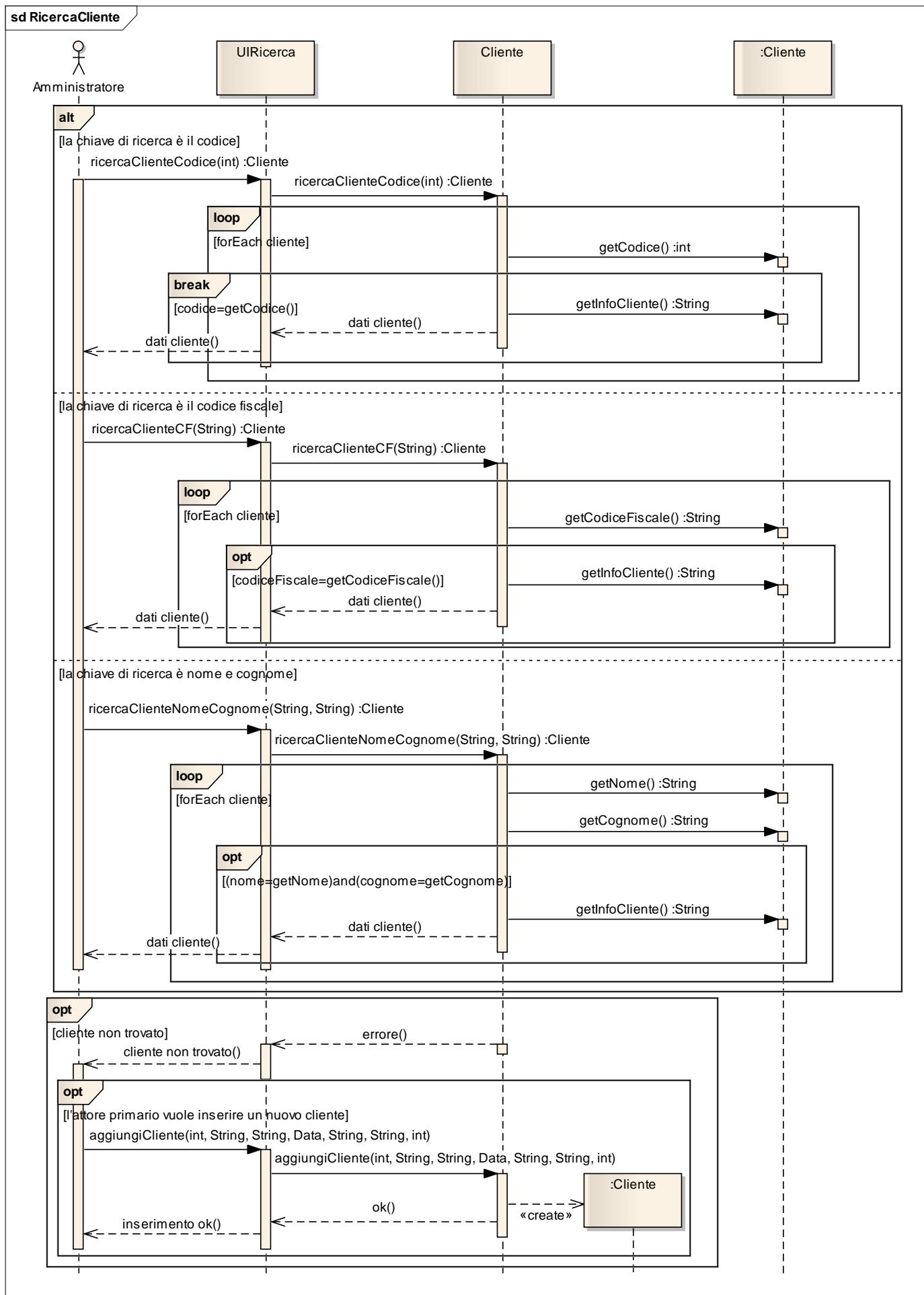
Continua alla pagina successiva



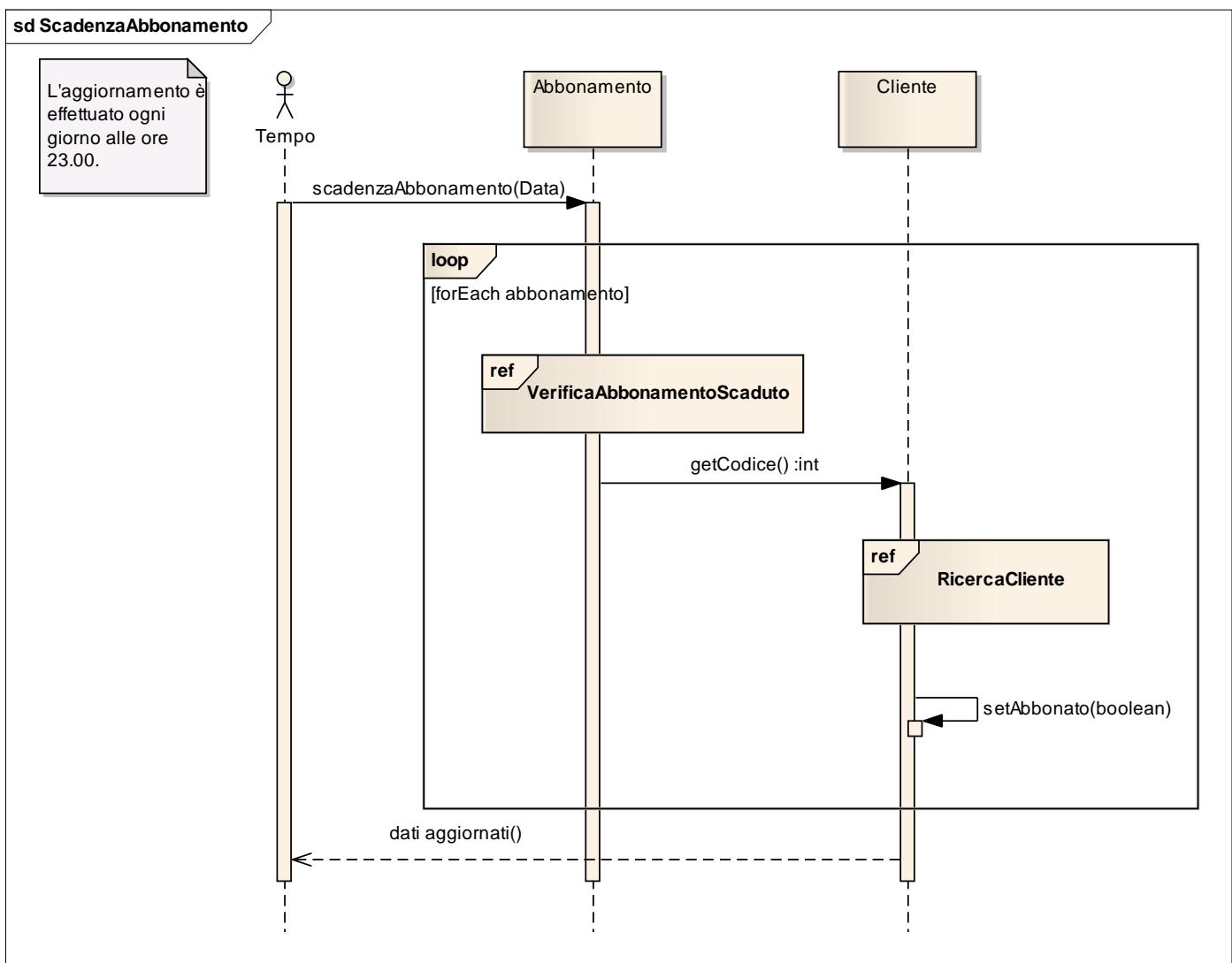
MODIFICA ABBONAMENTO



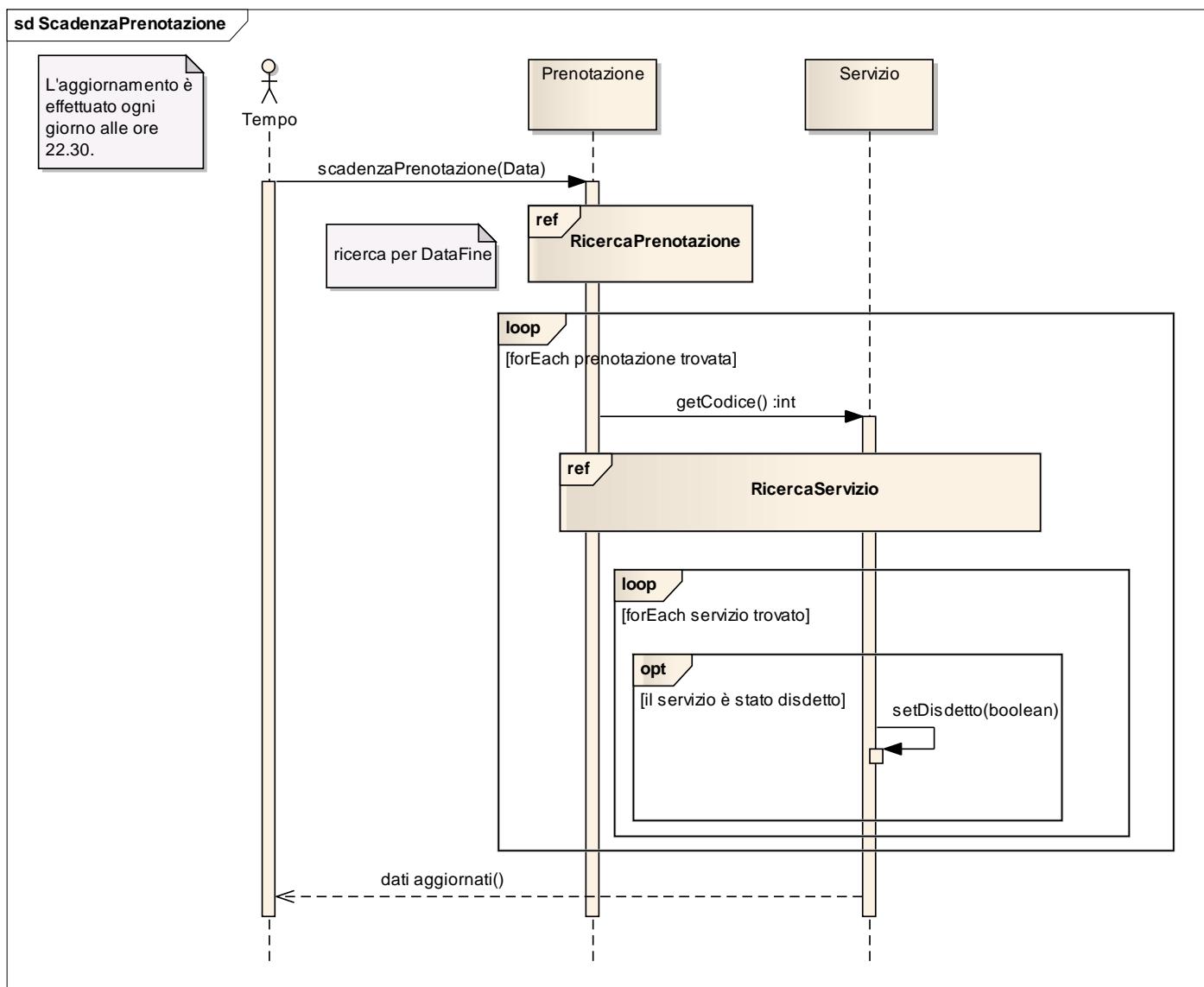
RICERCACLIENTE



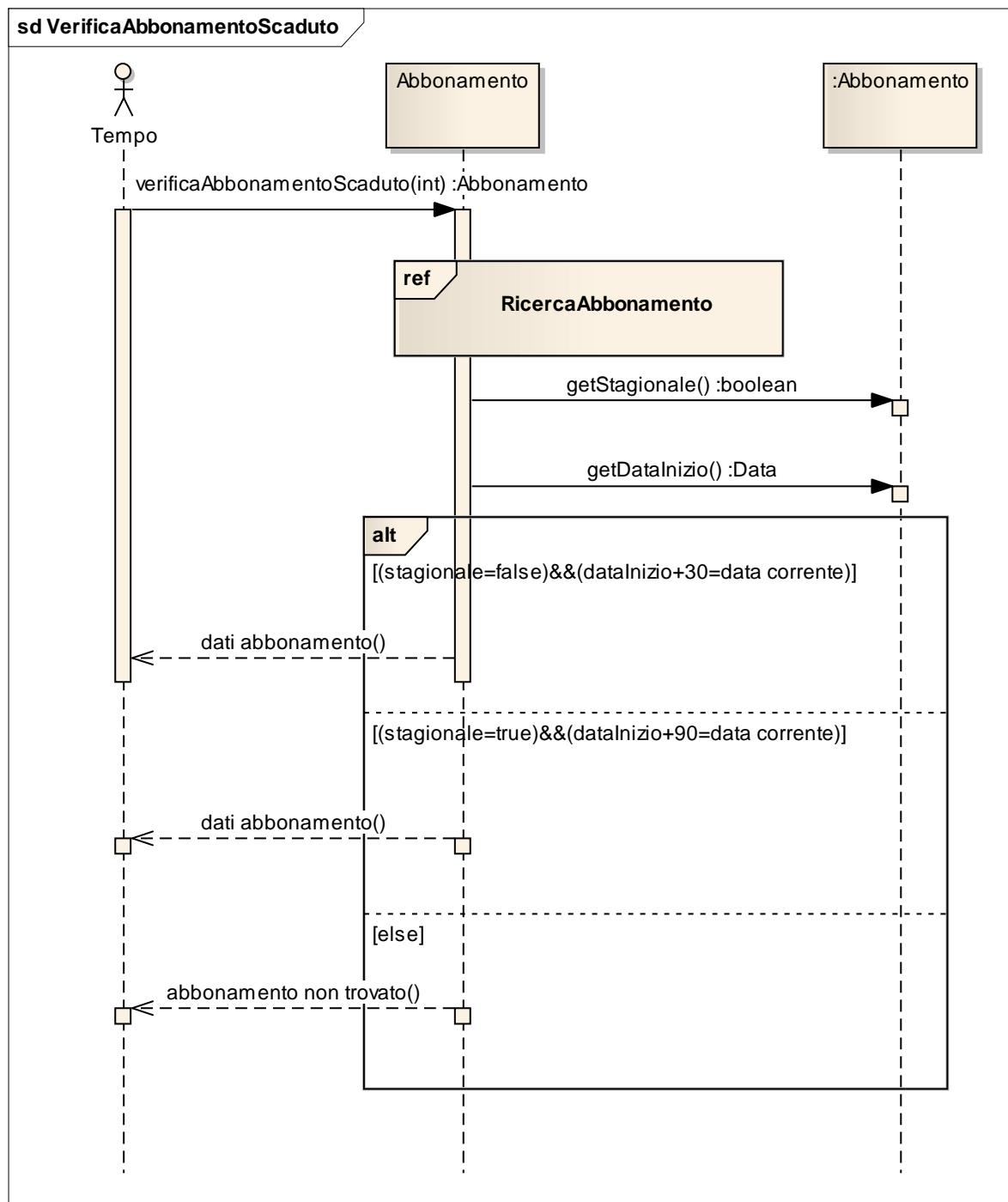
SCADENZA ABBONAMENTO



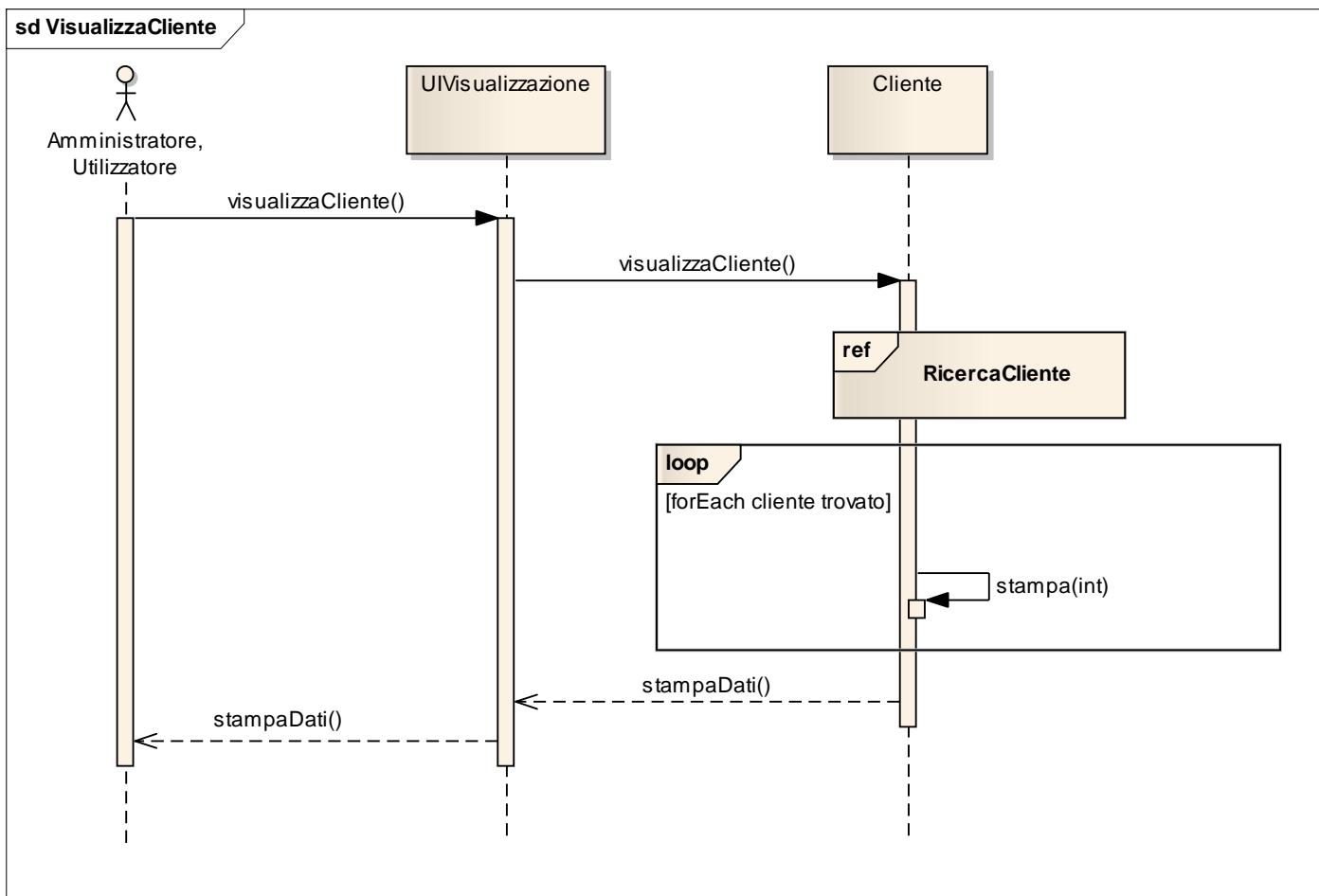
SCADENZA PRENOTAZIONE



VERIFICA ABBONAMENTO SCADUTO



VISUALIZZA CLIENTE



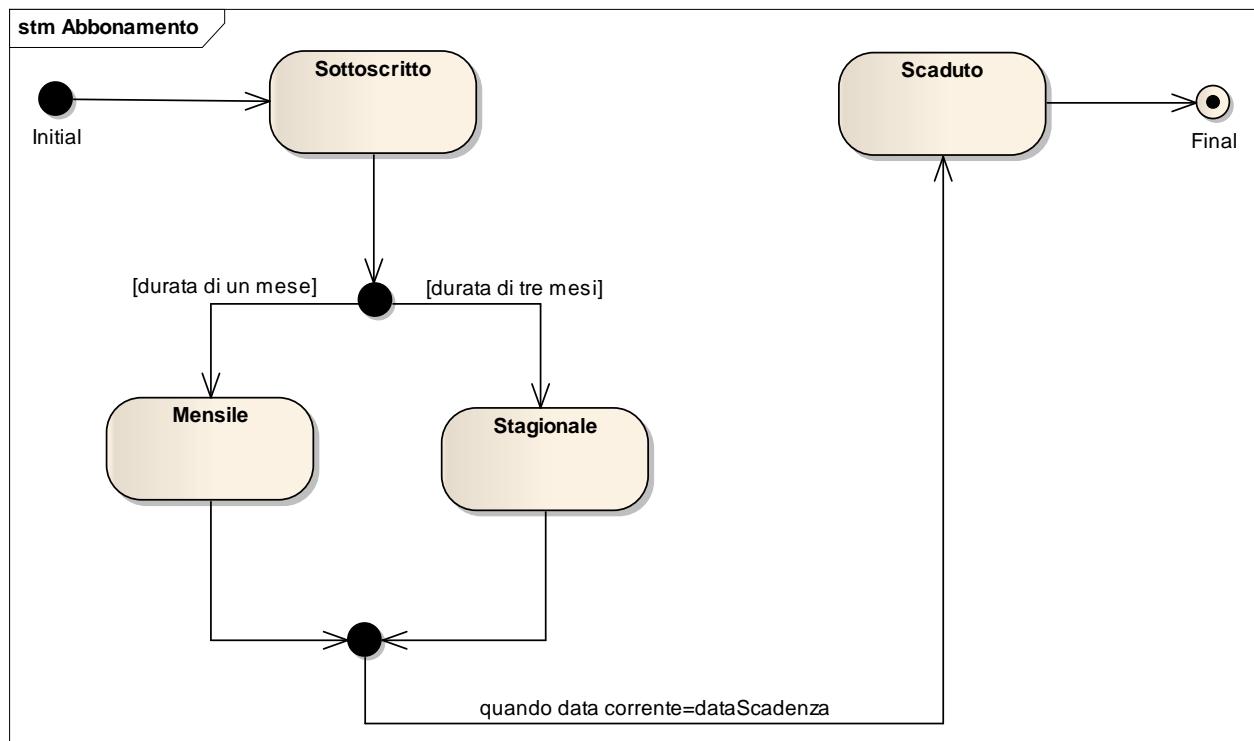
MACCHINE A STATI

Abbonamento

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Macchine a stati

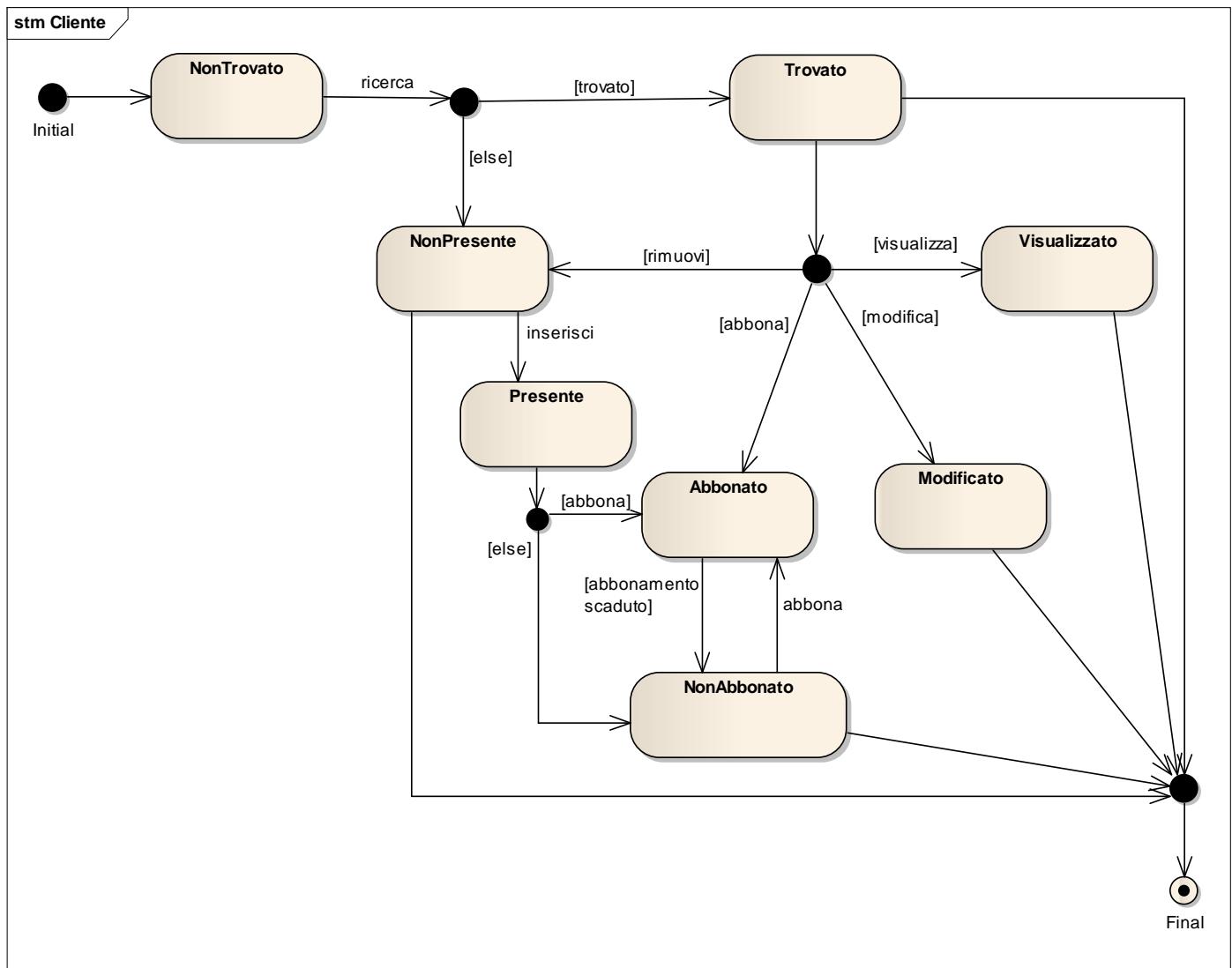


Cliente

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Macchine a stati

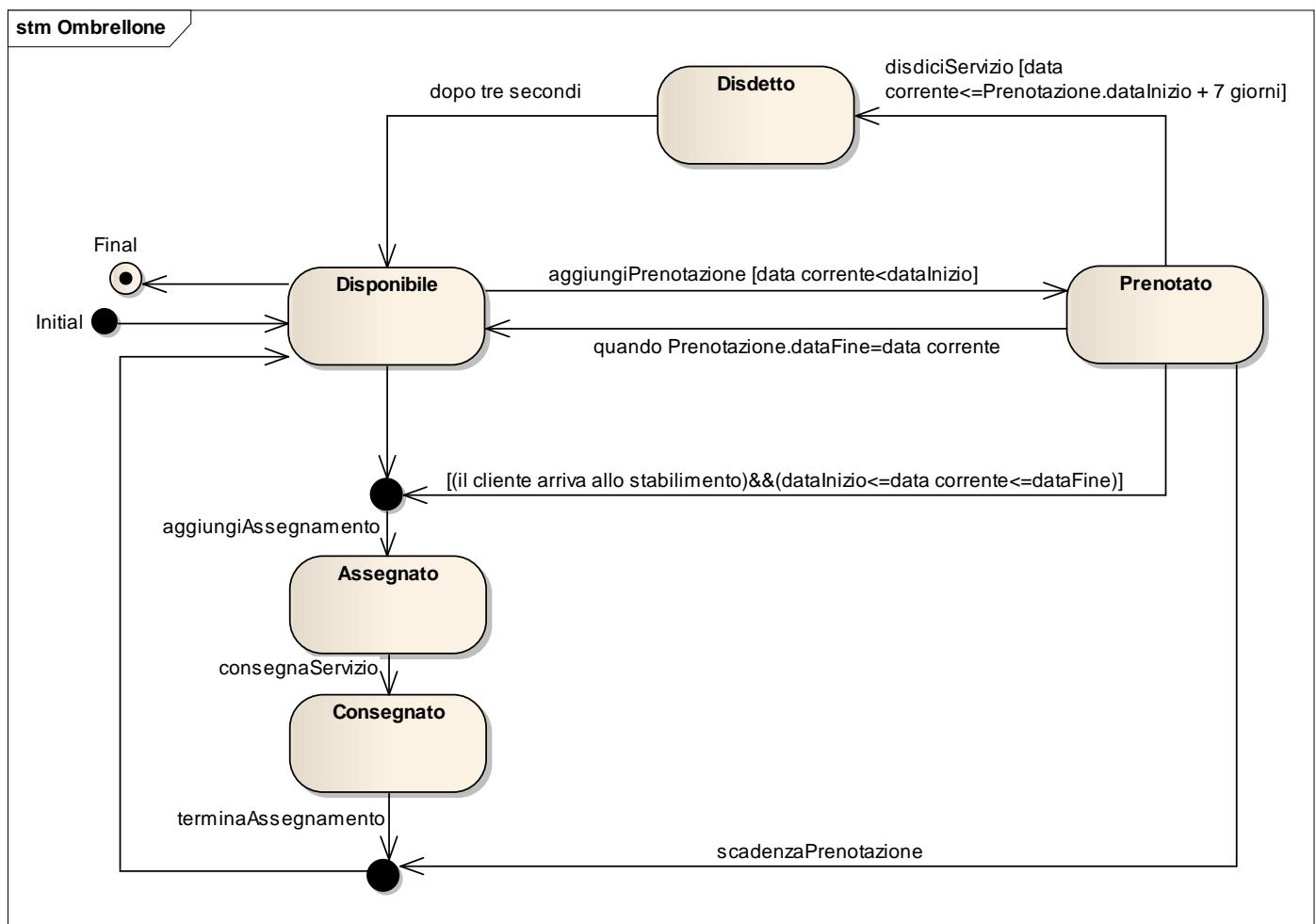


Ombrellone

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Macchine a stati



Prenotazione

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Macchine a stati

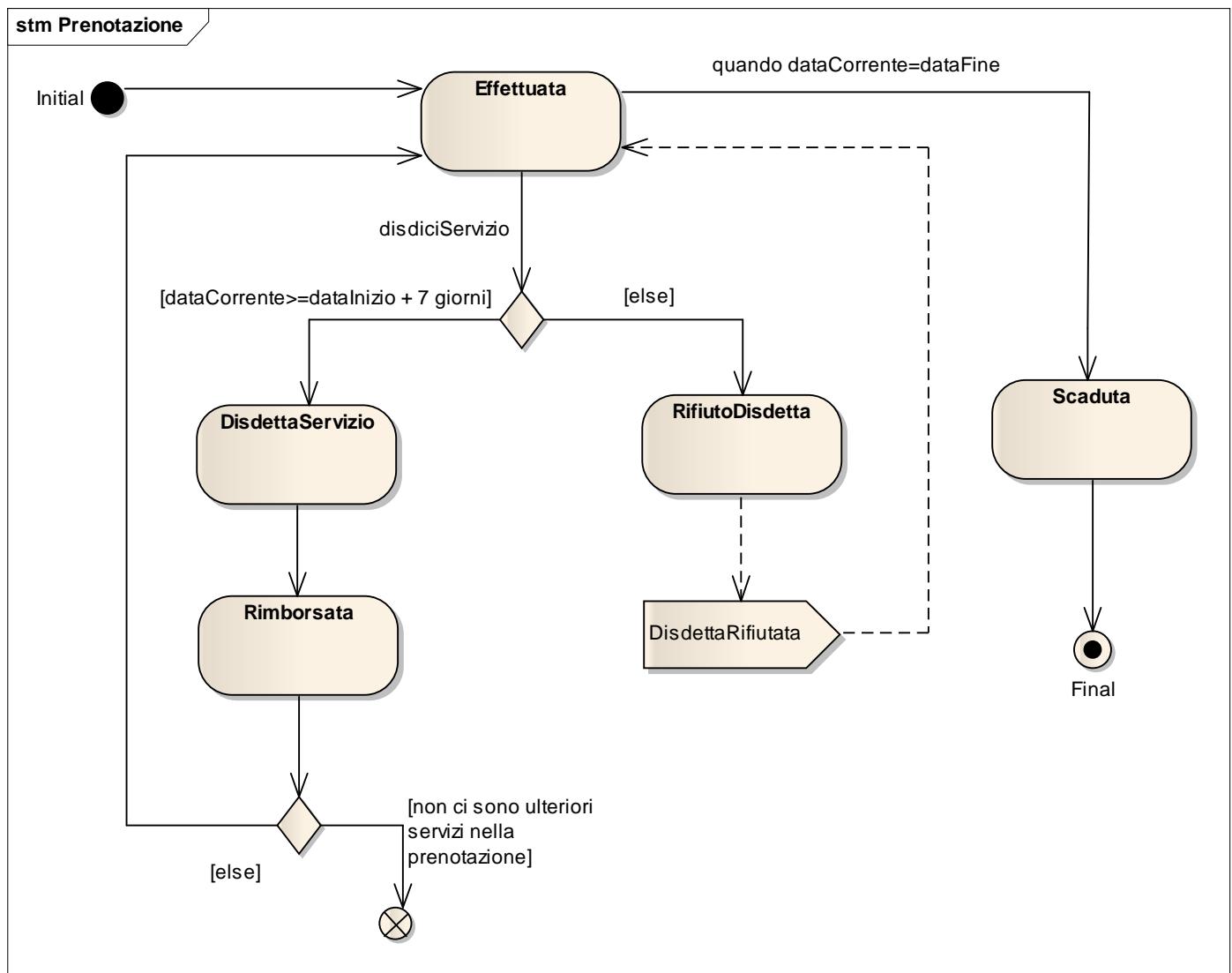


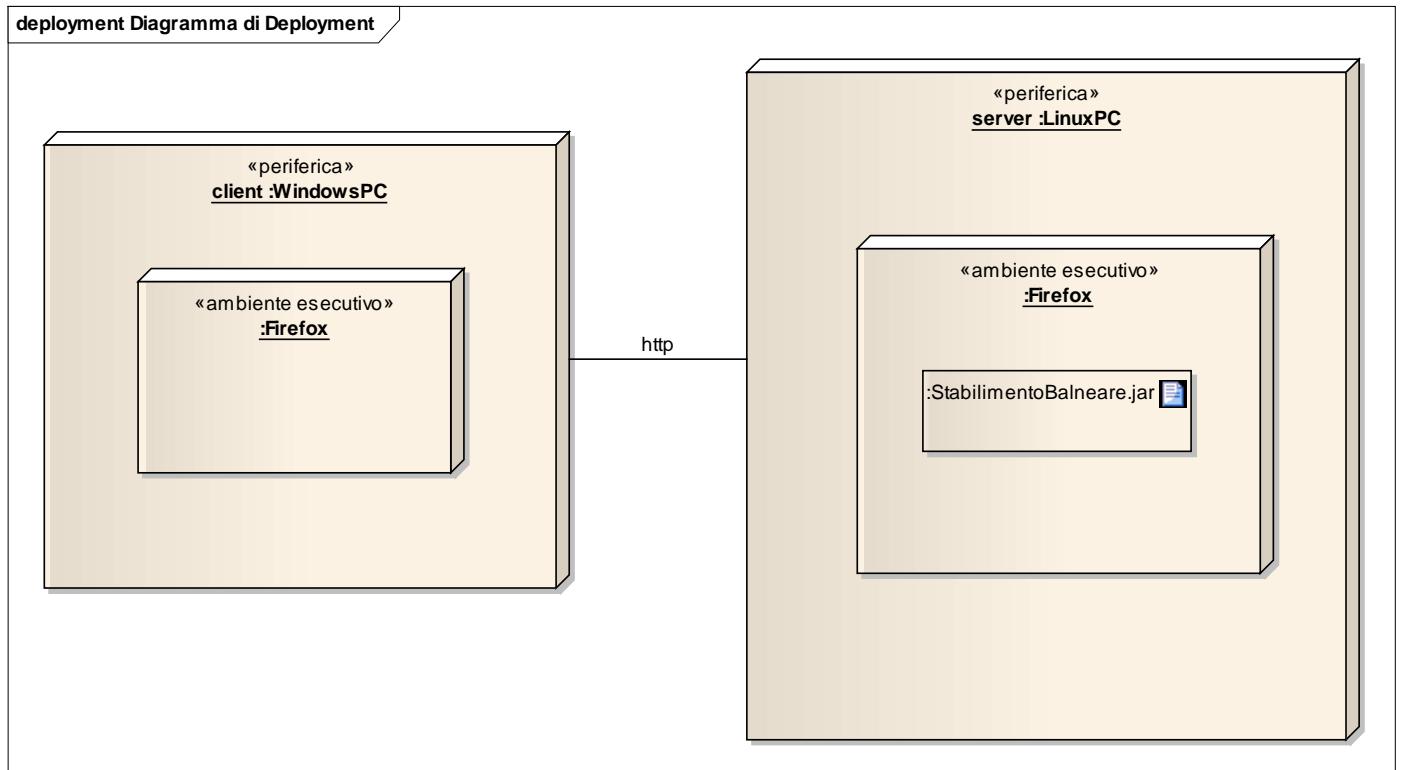
DIAGRAMMA DI DEPLOYMENT

Diagramma di Deployment

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Model



GANTT PROJECT

Ingegneria del software

Project managers:

non assegnato

Dates:

16-ago-2011 - 29-feb-2012

Tasks

Nome	Data d'inizio	Data di fine
Raccolta dei requisiti	16/08/11	20/08/11
Modellazione dei casi d'uso	22/08/11	27/09/11
Individuazione delle classi di analisi	27/09/11	01/10/11
Diagrammi di sequenza	03/10/11	10/11/11
Diagrammi di attività	10/11/11	18/11/11
Classi di progettazione	18/11/11	23/11/11
Diagramma dei componenti	23/11/11	24/11/11
Diagrammi di sequenza raffinati	24/11/11	25/11/11
Macchine a stati	25/11/11	30/11/11
Diagramma di Deployment	01/12/11	02/12/11
Revisione	05/12/11	10/12/11
Implementazione	12/12/11	13/01/12
Testing	13/01/12	29/02/12

Gantt Project

Resources

Nome	Ruolo predefinito	e-mail	Telefono	Ruolo assegnato
Antonio Esiliato	nessuno ruolo specifico			
Silvana Pizzonia	nessuno ruolo specifico			

Gantt Project

Diagramma di Gantt

