

InViaggio: Elaborazione - Iterazione 4

G. Messina – S. Squillaci – A. Zarbo

Introduzione

Per l'iterazione 4 sono stati scelti i casi d'uso rimanenti:

- ❖ Implementazione del caso d'uso UC4: Trasferimento Biglietto.
- ❖ Implementazione del caso d'uso UC7: Modifica Prenotazione.
- ❖ Implementazione del caso d'uso UC11: Visualizza storico.
- ❖ Implementazione del caso d'uso UC12: Convalida biglietto.
- ❖ Implementazione del caso d'uso di Avviamento necessario per inizializzare questa iterazione.

In questa iterazione è stato deciso anche di modificare il caso d'uso UC4 poiché originariamente implementava una funzionalità non congrua al programma. Si è pensato quindi di sostituirlo con una nuova funzionalità, la quale permette ad un utente di poter cedere il proprio biglietto prenotato ad un altro utente.

È stato rinominato il caso d'uso UC12, originariamente chiamato "Convalida prenotazione" in "Convalida biglietto", il funzionamento è rimasto invariato.

Aggiornamento elaborati della fase di ideazione

È stato aggiornato il glossario sulla base dei nuovi termini introdotti in questa iterazione, in particolare sono stati aggiunti:

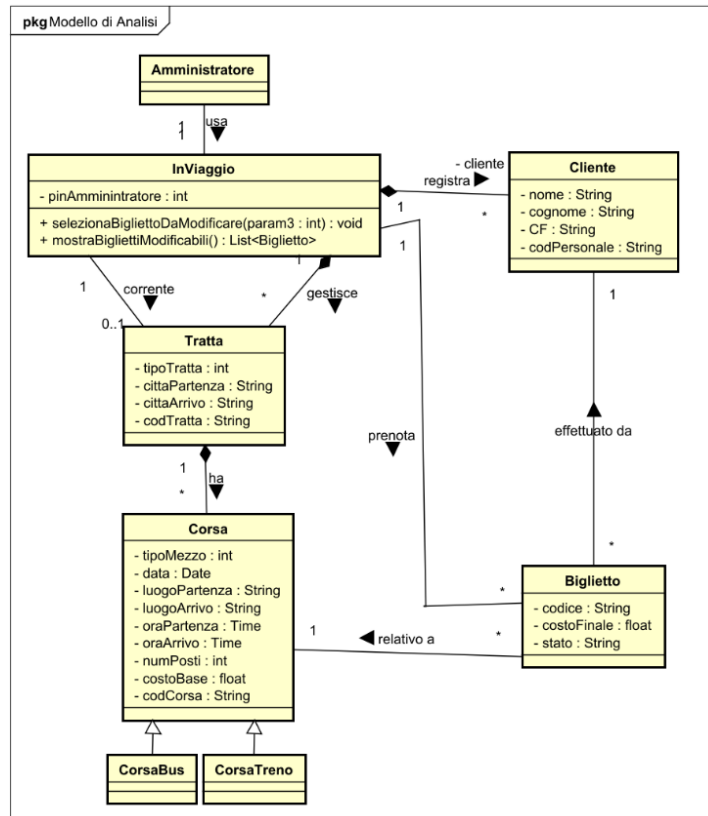
- **Biglietto valido:** Biglietto prenotato e non ancora utilizzato relativo ad una corsa non ancora avvenuta.
- **Biglietto annullato:** Biglietto prenotato ma annullato dall'amministratore a causa della sospensione della tratta relativa alla corsa o a causa della rimozione della corsa
- **Biglietto convalidato:** Biglietto prenotato e convalidato dall'amministratore (Biglietto utilizzato)
- **Biglietto scaduto:** Biglietto prenotato e non utilizzato relativo ad una corsa già avvenuta
- **Biglietto multato:** Biglietto scaduto di cui l'utente ha pagato la penalità durante la convalida di un altro biglietto.

1. Analisi Orientata agli Oggetti

Di seguito, come nelle precedenti iterazioni, vengono riportati: Modello di Dominio, Diagrammi di Sequenza di Sistema e contratti delle operazioni

1.1. Modello di Dominio

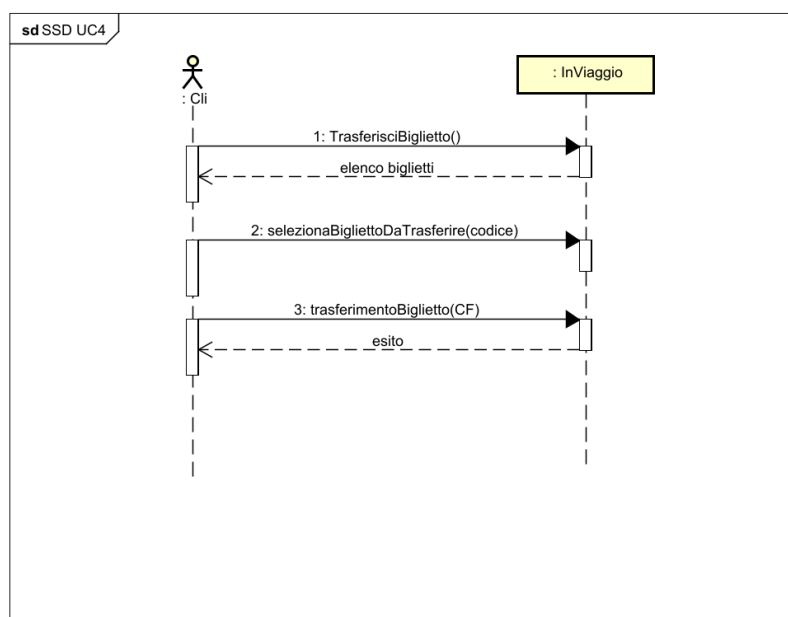
Per l'implementazione del caso d'uso UC11 e UC12 è stato necessario aggiungere l'attributo "stato" alla classe "Biglietto".



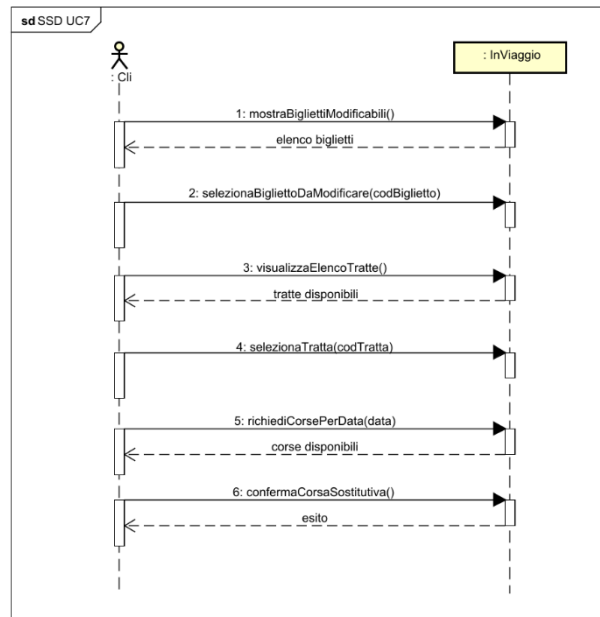
1.2. Diagrammi di Sequenza di Sistema (SSD)

Di seguito vengono riportati i diagrammi di sequenza di Sistema relativi ai casi d'uso presi in esame in questa iterazione.

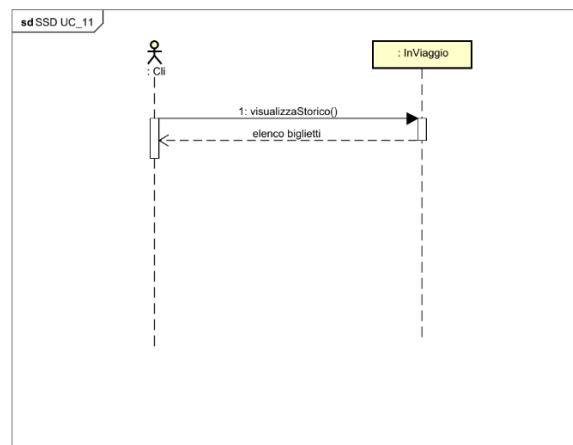
1.2.1. SSD UC4



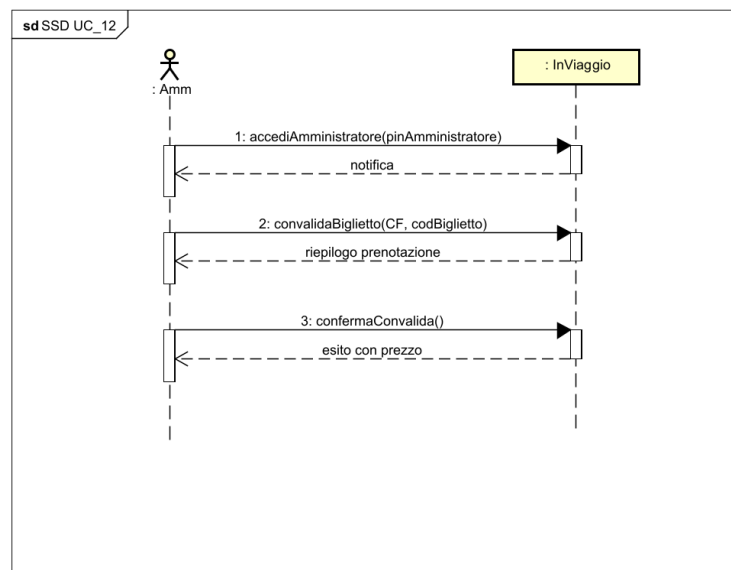
1.2.2. SSD UC7



1.2.3. SSD UC11



1.2.4. SSD UC12



1.3.1 Contratti delle Operazioni

1.3.1.1 UC4 trasferimentoBiglietto

Operazione	trasferimentoBiglietto(CF: String)
Riferimenti	Caso d'uso UC4: Trasferimento Biglietto.
Pre-condizioni	È stata recuperata l'istanza "cl" di Cliente (istanza del cliente che ha effettuato il login) È stata recuperata l'istanza "b" di Biglietto da trasferire
Post-condizioni	<ul style="list-style-type: none">- È stata recuperata l'istanza "cs" della classe Cliente sulla base del CF tramite la relazione registra.- L'istanza "b" di Biglietto, precedentemente recuperata, è stata associata all'istanza "cs" di Cliente tramite la relazione "effettuato da".- L'istanza "b" di Biglietto viene dissociata dall'istanza "cl" di Cliente.

1.3.1.2 UC7 selezionaBigliettoDaModificare

Operazione	selezionaBigliettoDaModificare(codBiglietto:String)
Riferimenti	Caso d'uso UC7: Modifica Prenotazione
Pre-condizioni	-È stata recuperata l'istanza "cl" di Cliente
Post-condizioni	<ul style="list-style-type: none">- È stata recuperata l'istanza "bc" di Biglietto sulla base dell'attributo codBiglietto tramite la relazione "effettuato da" con "cl"

1.3.1.3 UC7 confermaCorsaSostitutiva

Operazione	confermaCorsaSostitutiva()
Riferimenti	Caso d'uso UC7: Modifica Prenotazione
Pre-condizioni	-È stata recuperata l'istanza "cl" di Cliente -È stata recuperata l'istanza "t" di Tratta sulla base del "codTratta" tramite la relazione "gestisce" con inViaggio - È stata recuperata l'istanza "bc" di Biglietto
Post-condizioni	<ul style="list-style-type: none">- È stata recuperata l'istanza "c" di Corsa tramite la relazione "ha" con "t" sulla base del codCorsa.- È stato calcolato il prezzo del biglietto sulla base delle regole di dominio- È stato aggiornato l'attributo "costoFinale" di "bc".- È stata associata l'istanza "c" a "bc" tramite la relazione "relativo a".

1.3.1.4 UC11 visualizzaStorico

Operazione	visualizzaStorico()
------------	---------------------

Riferimenti	Caso d'uso UC11: Visualizza storico
Pre-condizioni	È stata recuperata l'istanza "cl" di Cliente (istanza del cliente che ha effettuato il login)
Post-condizioni	<ul style="list-style-type: none"> - Sono state recuperate tutte le istanze della classe Biglietto relative all'istanza "cl" tramite la relazione "effettuato da".

1.3.1.5 UC12 convalidaBiglietto

Operazione	convalidaBiglietto(CF: String, codBiglietto: String)
Riferimenti	Caso d'uso UC12: Convalida Biglietto
Pre-condizioni	-
Post-condizioni	<ul style="list-style-type: none"> - È stata recuperata l'istanza "cs" (cliente selezionato) di Cliente tramite la relazione "registra" sulla base del CF. - È stata recuperata l'istanza "bc" di Biglietto associata all'istanza "cs" sulla base del codBiglietto tramite la relazione "effettuato da"

1.3.1.6 UC12 confermaConvalida

Operazione	confermaConvalida()
Riferimenti	Caso d'uso UC12: Convalida Biglietto
Pre-condizioni	<ul style="list-style-type: none"> - È stata recuperata l'istanza "cs" di Cliente - È stata recuperata l'istanza "bc" di Biglietto
Post-condizioni	<ul style="list-style-type: none"> - L'attributo "stato" di "bc" viene aggiornato (viene impostato a "convalidato") - È stato aggiornato l'attributo "costoFinale" di "bc" sulla base della regola di dominio R2 - È stato aggiornato l'attributo "stato" in tutte le istanze di Biglietto associate all'istanza "cs"

2. Progettazione Orientata agli oggetti

In questa fase si procede con la definizione degli oggetti software a partire dagli oggetti concettuali individuati nella fase precedente. Inoltre, si definiscono anche le loro responsabilità e le loro interazioni al fine di soddisfare i requisiti individuati nei passi precedenti. A seguire sono riportati i diagrammi di sequenza (SD) ed il diagramma delle classi relativi ai casi d'uso UC4, UC7, UC11 ed UC12.

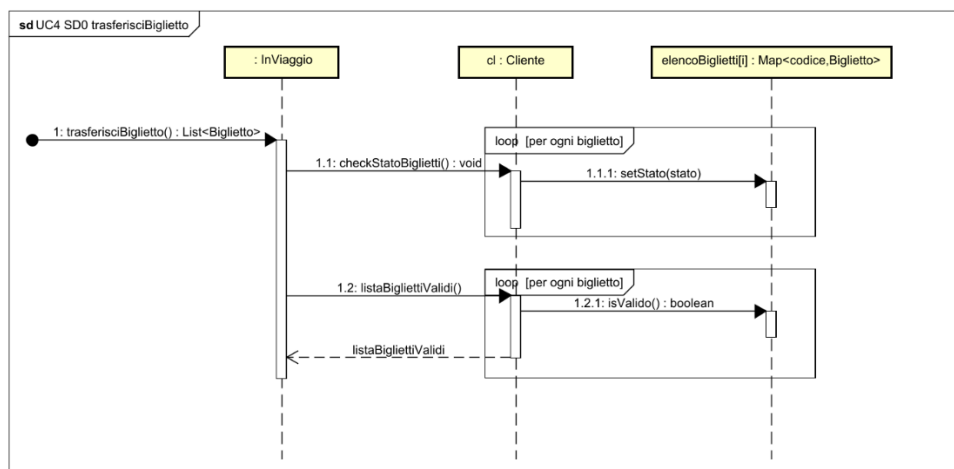
2.1 Pattern applicati

Sono stati applicati i principali pattern GRASP come Controller, Information Expert, basso accoppiamento (low Coupling) e alta coesione (High cohesion). Inoltre, per implementare la regola di dominio R2 per l'UC12 è stato riutilizzato il pattern Strategy.

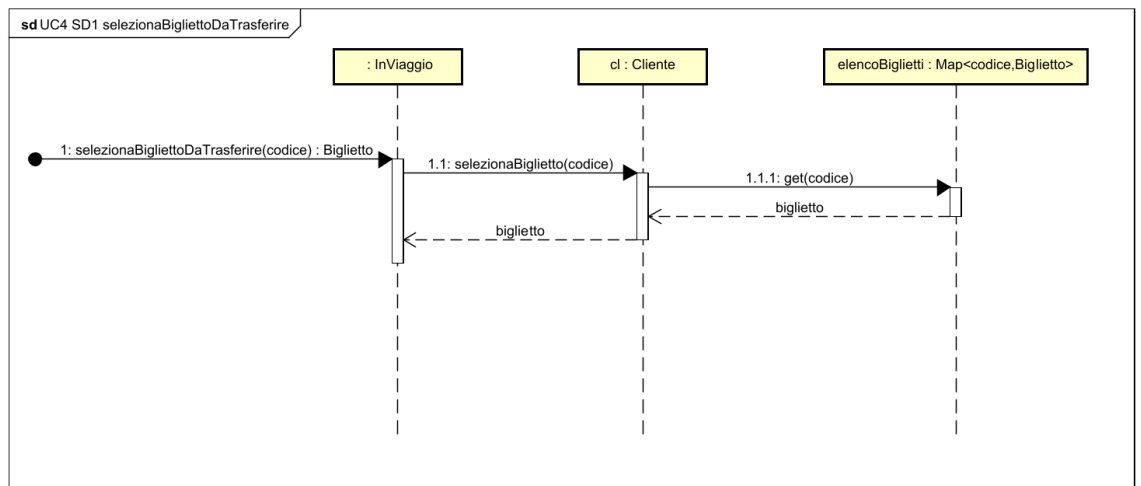
2.2 Diagrammi di Sequenza

Di seguito vengono riportati i diagrammi di sequenza relativi ai casi d'uso presi in esame:

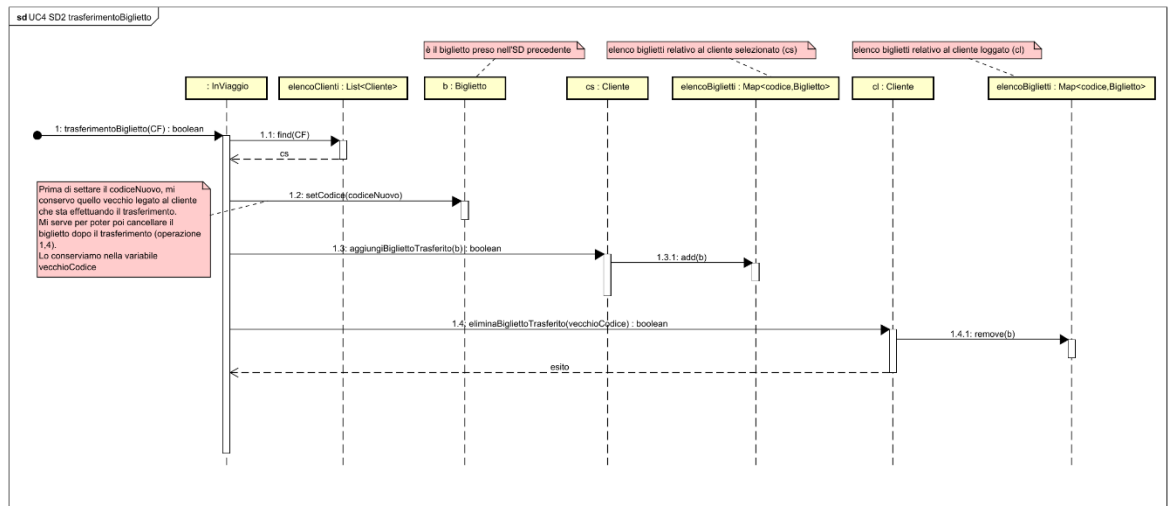
2.2.1 UC4 SD0



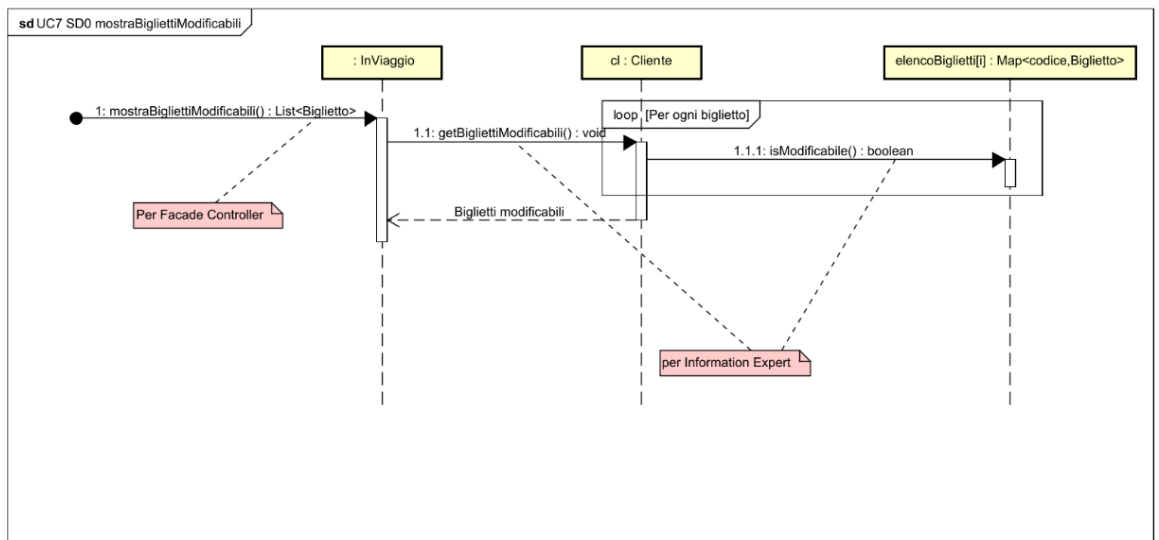
2.2.2 UC4 SD1



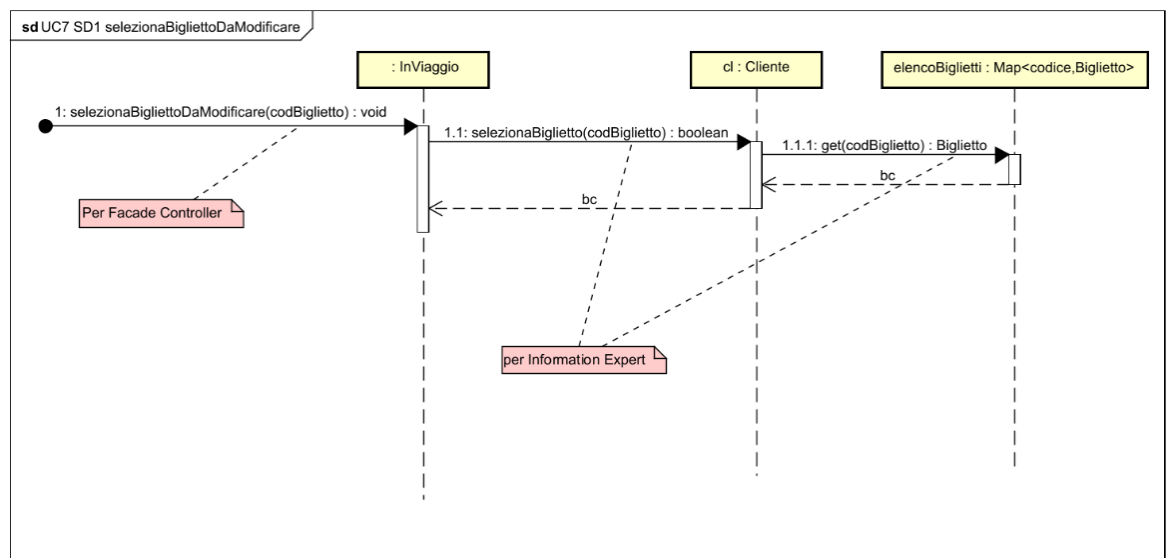
2.2.3 UC4 SD2



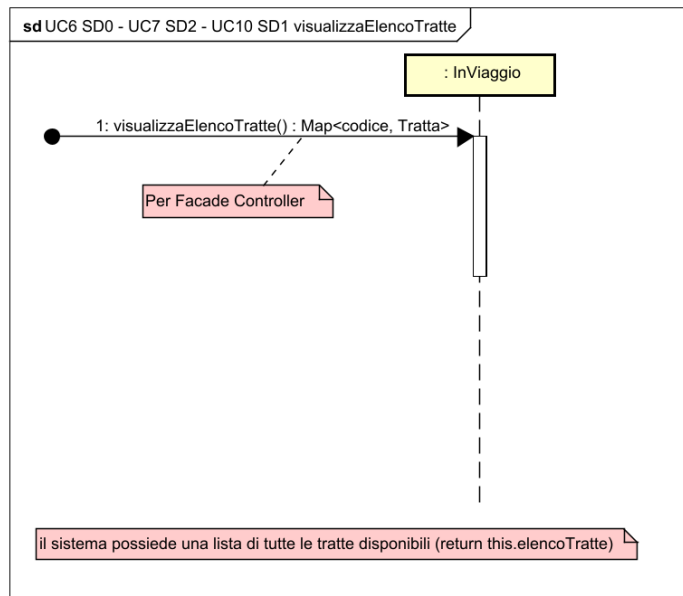
2.2.4 UC7 SD0



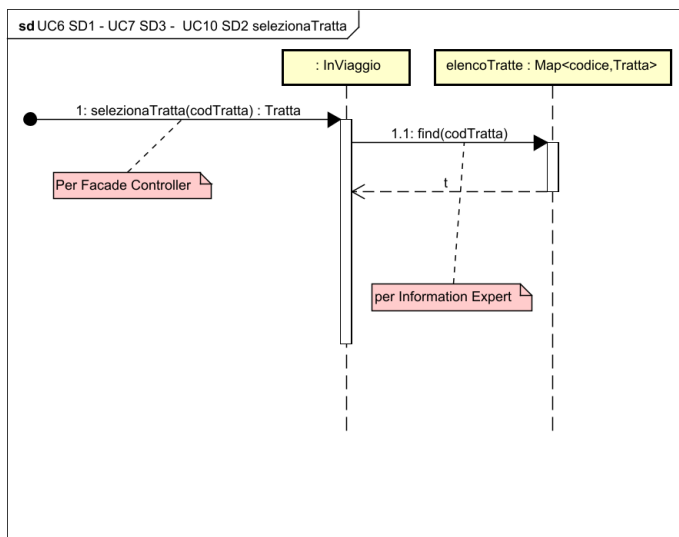
2.2.5 UC7 SD1



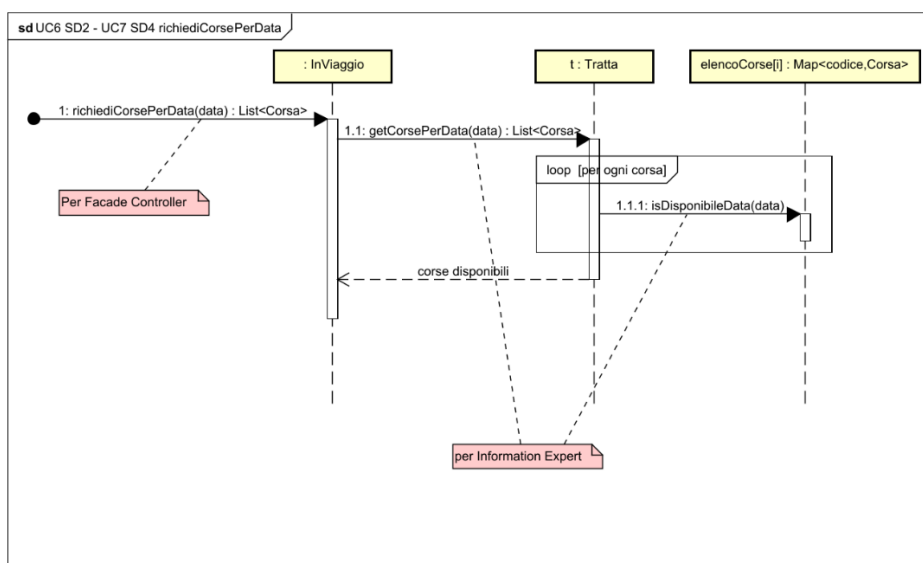
2.2.6 UC7 SD2



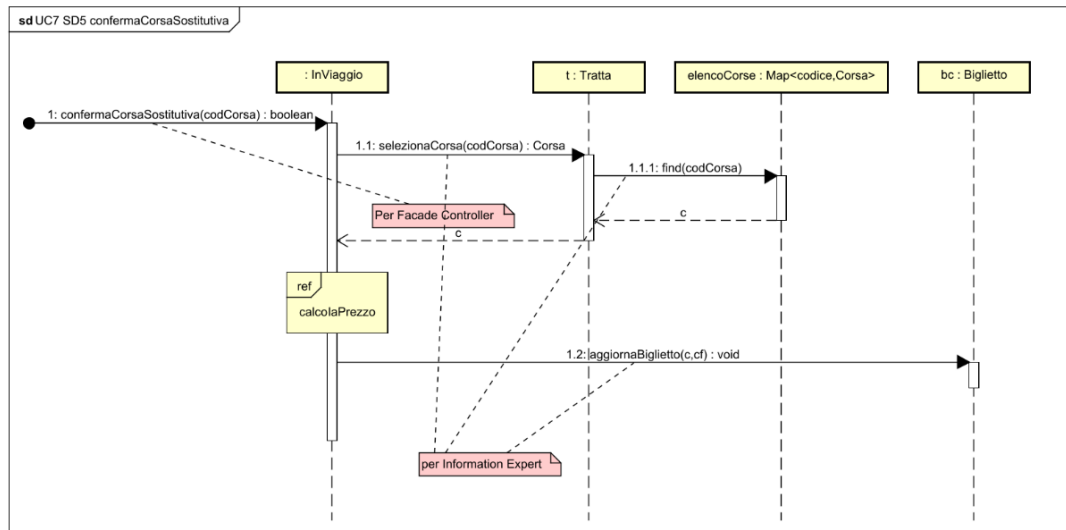
2.2.7 UC7 SD3



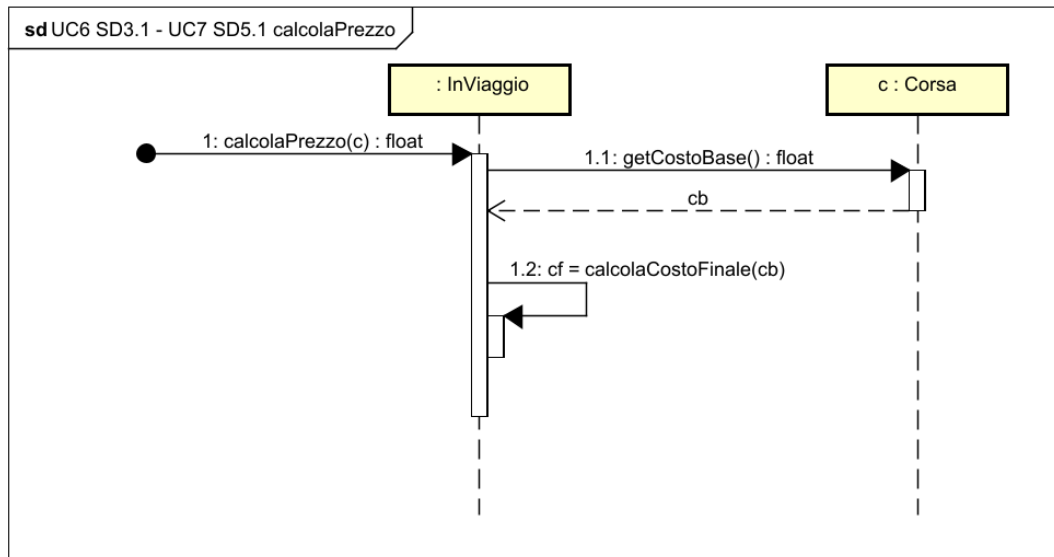
2.2.8 UC7 SD4



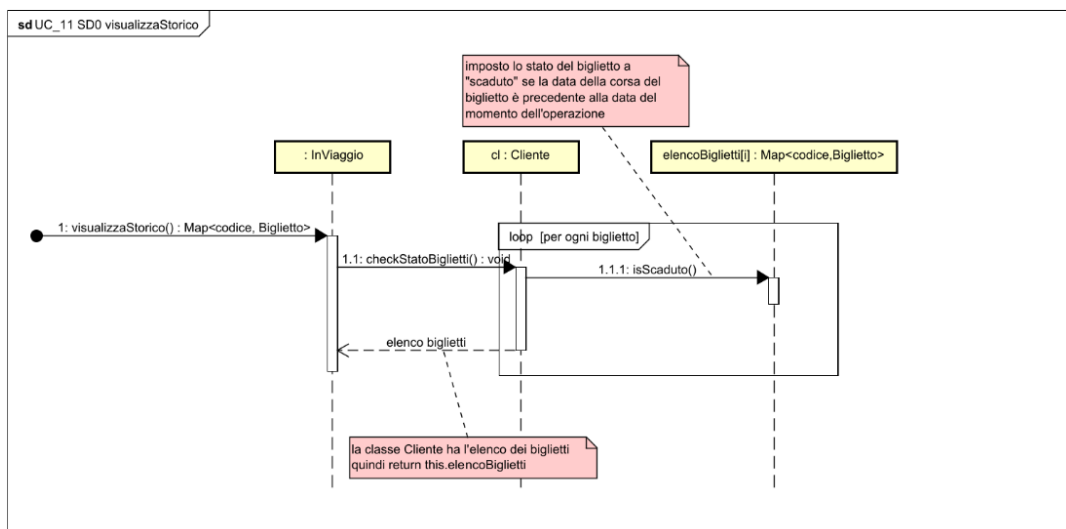
2.2.9 UC7 SD5



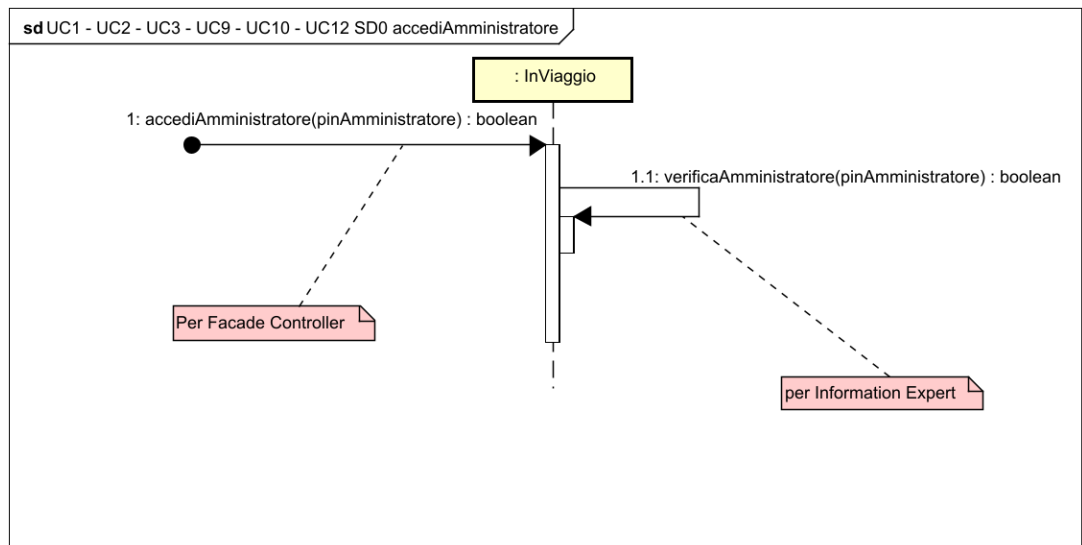
2.2.10 UC7 SD5.1



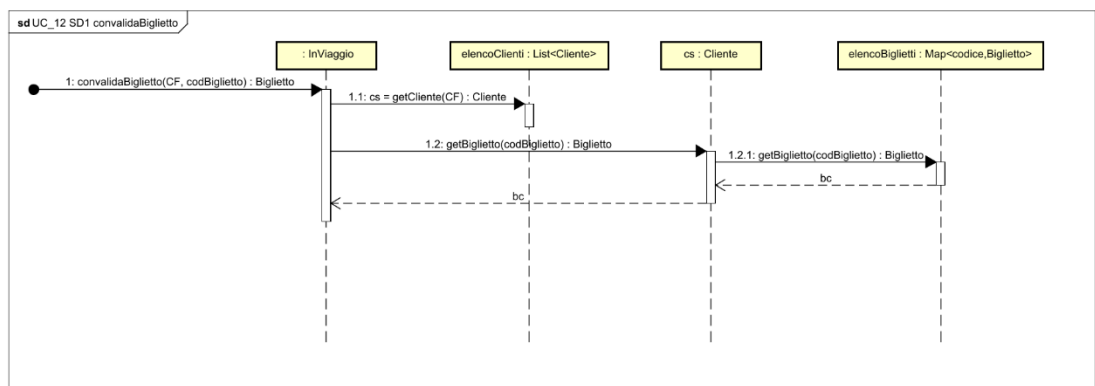
2.2.11 UC11 SD0



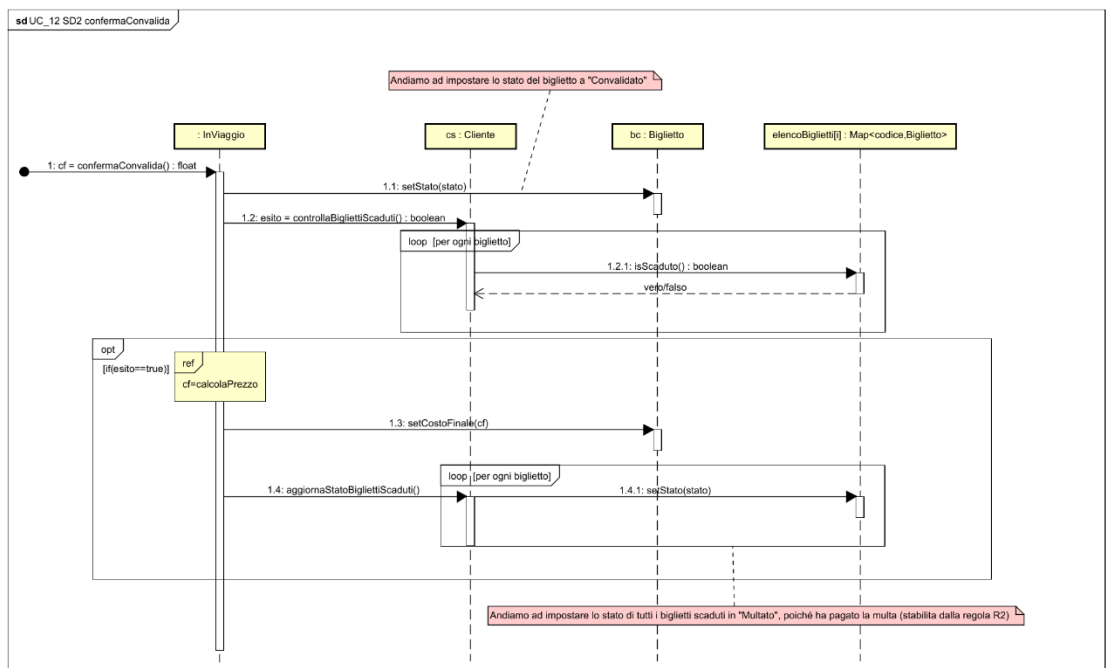
2.2.12 UC12 SD0



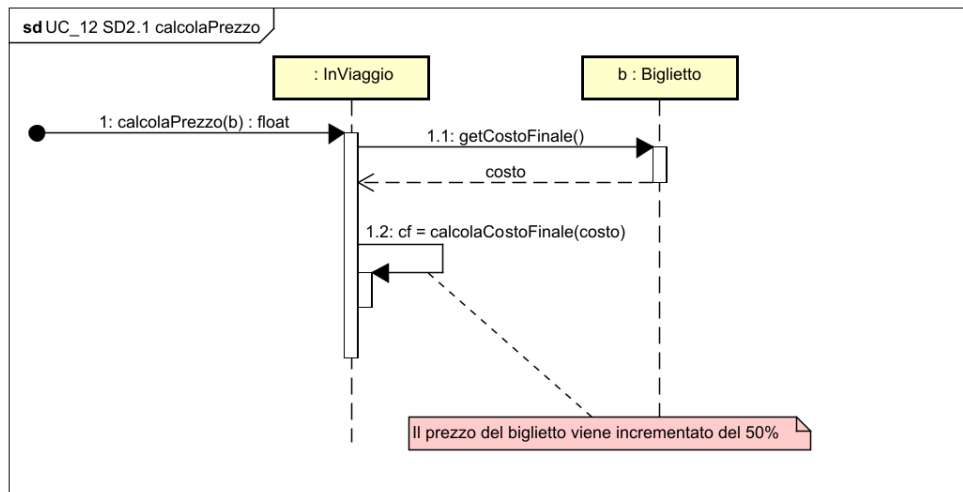
2.2.13 UC12 SD1



2.2.14 UC12 SD2



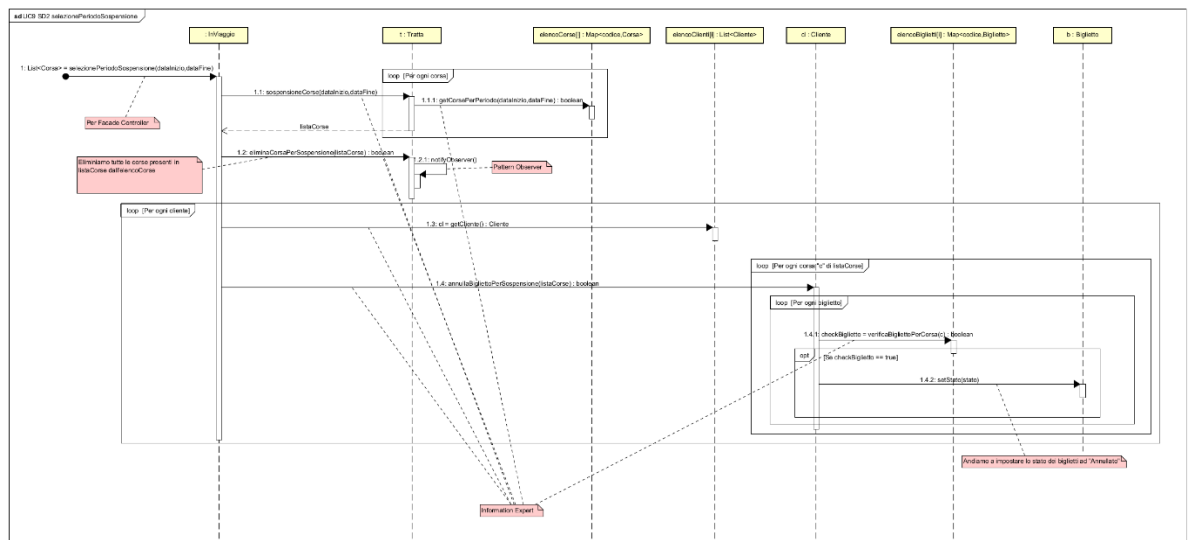
2.2.15 UC12 SD2.1



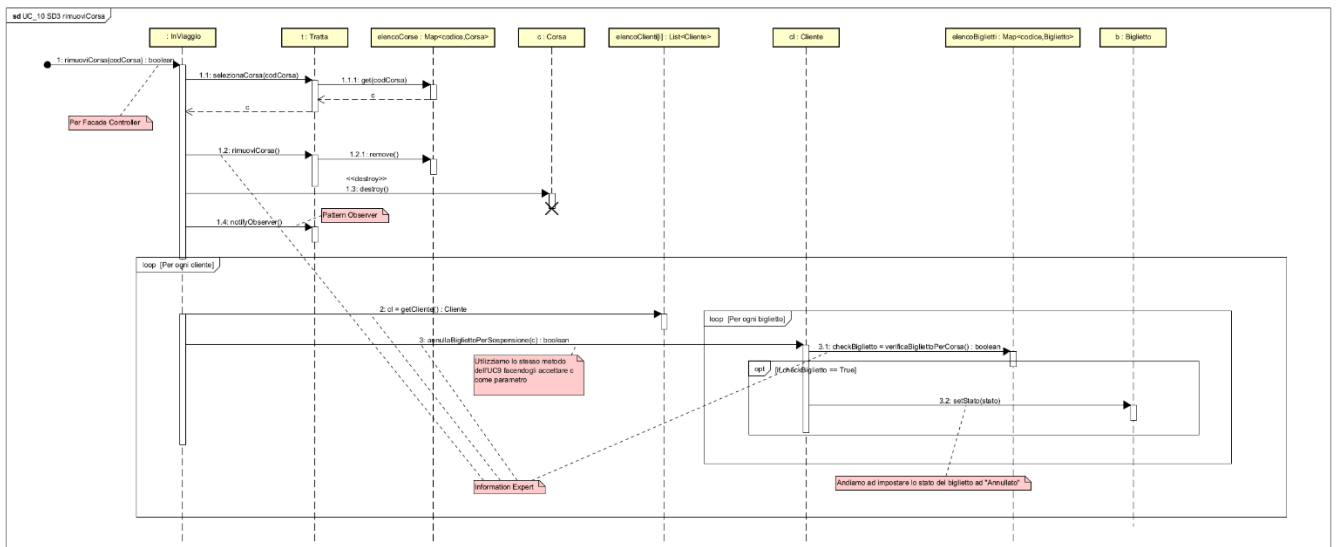
2.3 SD iterazioni precedenti modificati

È stato necessario modificare gli SD dei casi d'uso UC9 ed UC10 realizzati nella scorsa iterazione. In entrambi gli UC al posto di eliminare i biglietti relativi alle corse sospese/rimosse andiamo a modificare lo stato del biglietto, indicandolo come “annullato”. Di seguito sono riportati gli SD aggiornati:

2.3.1 UC9 SD2



2.3.2 UC10 SD3



2.4 Diagramma delle Classi di progetto

Per l'implementazione del caso d'uso UC12 è stata aggiunta la sottoclasse "Multa" alla classe "Prezzo Finale" per applicare la multa al costo del biglietto utilizzando il pattern strategy secondo la regola di dominio R2.
Per la visione del Diagramma delle classi di progetto aprire il file "iterazione4.asta".

3. Testing

Per la quarta iterazione sono stati eseguiti i test per verificare il corretto funzionamento dei casi d'uso UC4, UC7, UC11, UC12.

Si è scelto di continuare ad utilizzare il metodo bottom-up per effettuare i vari test. Vengono riportati i test dei metodi divisi per le relative classi:

- **Biglietto:**

- isModificabile: è stato verificato che il metodo:
 - ritorni "false" se non è modificabile, ossia ci sono meno di 12h tra la partenza della corsa del biglietto e l'istante in cui viene richiamato il metodo;
 - ritorni "true" quando il biglietto è modificabile, ossia ci sono più di 12h dal momento di chiamata del metodo e la partenza della corsa del biglietto.
- isScaduto: è stato verificato che il metodo:
 - ritorni "true" quando lo stato del biglietto è "Scaduto";
 - ritorni "false" quando lo stato del biglietto non è "Scaduto";
- isValido: è stato verificato che il metodo:
 - ritorni "true" quando lo stato del biglietto è "Valido";

- ritorni “false” quando lo stato del biglietto non è “Valido”;

- **Cliente**

- getBigliettiModificabili: è stato verificato che il metodo chiamato ritorni una lista contenente i biglietti che possono essere modificati;
- checkStatoBiglietti: è stato verificato che il metodo vada ad impostare correttamente lo stato “Scaduto” ai biglietti appartenenti a corse effettuate;
- controllaBigliettiScaduti: è stato verificato che:
 - il metodo ritorni “true” se esiste almeno un biglietto con lo stato “Scaduto”;
 - il metodo ritorni “false” se non esistono biglietti con lo stato “Scaduto”;
- aggiornaStatoBigliettiScaduti: è stato verificato che il metodo vada a modificare correttamente lo stato dei biglietti “Scaduto” in “Multato”, verificando che non vengano modificati i biglietti con altri stati;
- listaBigliettiValidi: è stato verificato che il metodo ritorni una lista contenente solamente i biglietti con lo stato “Valido”;

- **InViaggio**

- mostraBigliettiModificabili: è stato verificato che il metodo chiamato ritorni la lista attesa dei biglietti modificabili;
- selezionaBigliettoDaModificare: è stato verificato che il metodo chiamato setti nel bigliettoCorrente il biglietto con il codice richiesto;
- confermaCorsaSostitutiva: è stato verificato che il metodo chiamato torni “true” e il prezzo venga calcolato in maniera congrua alle regole di dominio;
- visualizzaStorico: è stato verificato che il metodo chiamato ritorni mappa aggiornata desiderata;
- convalidaBiglietto: è stato verificato che il metodo chiamato torni:
 - “true” se il codice fiscale esiste e il codice del biglietto passato è associato ad un biglietto posseduto dal cliente.
 - “false” se non esiste nessun cliente con il codice fiscale passato.
 - “false” se non esiste nessun biglietto associato a quel cliente con il codice passato.
- confermaConvalida: è stato verificato che il metodo chiamato imposti correttamente lo stato del biglietto a “convalidato” e calcoli il prezzo in maniera congrua alla regola di dominio R2;
- trasferisciBiglietto: è stato verificato che il metodo chiamato ritorni la lista attesa dei biglietti validi;

- selezionaBigliettoDaTrasferire: è stato verificato che il metodo chiamato setti nel bigliettoCorrente il biglietto con il codice richiesto;
- trasferimentoBiglietto: è stato verificato che il metodo chiamato ritorni:
 - “true” se il codice fiscale passato è associato ad un cliente esistente ed avviene l’effettivo trasferimento del biglietto.
 - “false” se il codice fiscale passato non è associato ad un cliente esistente e quindi non avviene il trasferimento del biglietto.