

# Sistema Gestione ZTL

Riccardo Maria Pesce

Anno Accademico 2019-2020

## **Abstract**

In questa terza iterazione verranno implementati i restanti casi d'uso, rifinito il software per tenere in considerazione gli utenti senza ID. Vedremo di più dopo il resoconto dell'incontro con il cliente.

# Contenuti

<b>1</b>	<b>Resoconto Iterazione 3</b>	<b>3</b>
1.1	Richiesta Gestione Sanzioni automatica . . . . .	3
1.2	Richiesta Riconoscimento Targhe automatico . . . . .	3
<b>2</b>	<b>Requisiti</b>	<b>4</b>
2.1	Gestione Utenti senza Dispositivo . . . . .	4
2.2	Emissione Sanzioni . . . . .	5
<b>3</b>	<b>Modello dei Casi D'Uso</b>	<b>5</b>
3.1	Caso D'Uso UC1 . . . . .	5
3.2	Caso D'Uso UC2 . . . . .	6
3.3	Caso D'Uso UC5 . . . . .	7
<b>4</b>	<b>Regole di Business</b>	<b>8</b>
<b>5</b>	<b>Modello di Dominio</b>	<b>9</b>
<b>6</b>	<b>Diagrammi di Sequenza di Sistema</b>	<b>10</b>
6.1	UC1 . . . . .	10
6.2	UC2 . . . . .	13
6.3	UC5 - Emetti Sanzione . . . . .	14
<b>7</b>	<b>Contratti delle Operazioni</b>	<b>16</b>
7.1	CO5 - notificaIngressoTarga() . . . . .	16
7.2	CO6 - notificaUscitaTarga() . . . . .	16
7.3	CO7 - emettiSanzione() . . . . .	16
<b>8</b>	<b>Diagrammi di Interazione</b>	<b>17</b>
8.1	UC1 - Registra Ingresso . . . . .	17
8.2	UC2 - Registra Uscita . . . . .	18
8.3	UC3 - Gestisci Terminali . . . . .	19
8.4	UC4 - Gestisci Residenti . . . . .	20
8.5	UC5 - Emetti Sanzione . . . . .	21
<b>9</b>	<b>Diagrammi delle Classi Progettuali</b>	<b>21</b>

<b>10 Motivazioni progettuali</b>	<b>22</b>
10.1 Riduzione dell'accoppiamento tramite <i>Controller</i> . . . . .	22
10.2 Incapsulare richieste d'accesso/uscita . . . . .	22
10.3 Interfaccia Utente migliorata . . . . .	22
<b>11 Casi di Test</b>	<b>22</b>
11.1 Test Unitari della classe <i>GestoreAccessi</i> . . . . .	23
11.2 Test d'integrazione di <i>GestoreSanzioni</i> . . . . .	23
11.3 Test d'integrazione di <i>Dispositivo</i> . . . . .	24

# 1 Resoconto Iterazione 3

Durante il meeting con il cliente, ci è stato fatto notare che:

- Gli utenti senza dispositivo, che commettono un'infrazione, non vengono sanzionati.
- Gli utenti che possiedono un dispositivo, possono essere multati agevolmente, in quanto un'infrazione verrebbe associata al codice ID del dispositivo.

Questo ci porterà a definire dei requisiti nuovi, che si andranno ad aggiungere a quelli passati, ed integreranno/chiariranno quelli definiti in fase di ideazione. Da un punto di vista software, le dipendenze sono ridotte rispetto a prima, dove il sistema centrale era responsabile di tutto. Tuttavia, si vedono chiaramente alcune migliorie che possono essere apportate, e che verranno ampiamente discusse nelle motivazioni progettuali.

## 1.1 Richiesta Gestione Sanzioni automatica

Come accennato nel paragrafo precedente, si vuole implementare nel sistema la funzionalità di sanzionare automaticamente gli utenti carico-scarico che commettono un'infrazione. Questa logica era prima delegata al comune, in quanto il software notificava (nella riga di comando), il transito irregolare. Poi, chi di competenza multava l'utente. Adesso, dobbiamo implementare la logica di sanzionamento per gli utenti-carico scarico.

## 1.2 Richiesta Riconoscimento Targhe automatico

Per quanto concerne l'ingresso di utenti sprovvisti di dispositivo, il cliente richiede che il riconoscimento targa venga effettuato dal sistema, e non dalla revisione delle registrazioni, che è una gran perdita di tempo inutile. L'implementazione dell'algoritmo verrà eseguita in parallelo dal dipartimento di ingegneria artificiale. Noi qui vogliamo aggiungere la funzione di gestione accessi anche per utenti senza terminale, e quindi tale funzione, per adesso, sarà solamente simulata qui.

## 2 Requisiti

### 2.1 Gestione Utenti senza Dispositivo

Riprendendo i requisiti definiti in fase di ideazione, andiamo ad aggiungere la nuova richiesta:

- Se l'utente è residente, il sistema, dopo aver acquisito il codice identificativo, conferma l'ingresso ritornando quest'ultimo e lascia passare il veicolo.
- Se l'utente è un utente Carico-Scarico, il sistema:
  1. Dovrà controllare che il valico dalla quale si sta entrando sia consentito al transito di tale tipologia di utenti.
  2. Dovrà controllare che l'utente non abbia già usufruito dei due intervalli di tempo consentiti.

Solo in caso positivo, il sistema ritrasmetterà il codice di identificazione come conferma del transito. In caso negativo, verrà ritrasmesso -1 e verrà sanzionato con una sanzione di tipo *Transito Irregolare*.

- Se l'utente non ha alcun dispositivo di identificazione, viene registrata la sua targa e notificato il transito irregolare a schermo.

Per quanto concerne l'uscita:

- Se l'utente residente, il terminale ritorna l'ID e l'utente esce senza ulteriori azioni necessarie.
- Se l'utente è carico-scarico, il sistema controlla l'eventuale presenza di infrazioni all'uscita, e se presenti queste verranno notificate all'utente ed aggiunte al suo dispositivo (identificato da codice ID).
- Se l'utente non ha alcun dispositivo, verrà registrata la targa e notificata l'uscita al sistema centrale, dove chi di dovere prenderà le giuste misure.

## 2.2 Emissione Sanzioni

Ribadiamo che mentre per gli utenti senza targa le sanzioni vengono eseguite dai dipendenti (in quanto esse possono prevedere procedimenti al di fuori della portata del sistema), per gli utenti carico-scarico le infrazioni vengono generate automaticamente dal sistema ed in seguito associate al dispositivo che ha eseguito l'infrazione. Alla luce di ciò, definiamo:

- Il sistema deve sanzionare opportunamente l'utente carico-scarico colto a infrangere.
- Il sistema deve accumulare per un utente le sanzioni che spettano ad esso, e deve quindi provvedere l'ammontare totale ed un sommario sulle irregolarità commesse.

## 3 Modello dei Casi D'Uso

A tal proposito i casi d'uso che descriveremo qui saranno *UC1* e *UC2* opportunamente modificati per accomodare gli utenti privi di dispositivo (e quindi di ID), ed infine il caso d'uso *UC5 - Sanziona Utente*.

### 3.1 Caso D'Uso UC1

- **UC1:** Registra Ingresso
- **Portata:** Sistema Gestion ZTL
- **Livello:** Obiettivo utente
- **Pre-condizioni:** l'utente sta varcando il valico d'ingresso
- **Garanzie di successo (Post-condizioni):** l'utente (identificato) entra nella zona a traffico limitato
- **Scenario principale di successo**
  - L'utente, avvicinatosi al varco, innesta il sistema attraverso l'invio, da parte del telepass, di un codice identificativo
  - Se l'utente è residente, il telepass lo lascia passare senza ulteriori azioni

- Se l'utente non è residente, il terminale è abilitato all'ingresso di utenti carico-scarico e l'utente non ha già usufruito dei suoi due intervalli consentiti, il sistema registra l'orario d'ingresso e ritorna alla vettura il codice identificativo come conferma, e lo lascia passare

- **Scenari alternativi**

- **Il varco non è abilitato all'accesso degli utenti carico-scarico**
  1. Viene registrata l'infrazione *Ingresso irregolare* all'utente (attraverso il suo ID).
- **Il transito sta avvenendo in un intervallo non consentito**
  1. Viene registrata l'infrazione *Transito Irregolare* all'utente attraverso il suo ID.
- **L'utente non possiede alcun dispositivo telepass**
  1. Attraverso un'apposita telecamera viene registrata la targa dell'utente.
  2. Viene notificata l'infrazione *Transito non consentito*

- **Requisiti speciali:** Bassa latenza

- **Frequenza:** Ogni qualvolta un utente si presenta al varco

### 3.2 Caso D'Uso UC2

- **UC2:** Registra Uscita
- **Portata:** Sistema Gestion ZTL
- **Livello:** Obiettivo utente
- **Pre-condizioni:** l'utente si trova all'interno della zona a traffico limitato
- **Garanzie di successo (Post-condizioni):** l'utente esce dalla zona a traffico limitato
- **Scenario principale di successo**

- L’utente, avvicinandosi al varco d’uscita, attiva il sistema attraverso l’invio, da parte del telepass, di un codice identificativo
- Se l’utente è residente, il telepass lo lascia uscire senza ulteriori azioni.
- Se l’utente è carico-scarico, il varco di uscita è quello abilitato a tali utenti e non ha sostato per più di un’ora, il sistema registra lo rimuove semplicemente da una determinata lista di utenti carico-scarico all’interno della ZTL e ritorna alla vettura il codice identificativo come conferma, e lo lascia passare
- Se l’utente è sprovvisto di dispositivo, viene lasciato passare, in quanto comunque il suo transito è stato registrato. Viene solamente notificata l’uscita.

- **Scenari alternativi**

- **L’utente è rimasto per più del tempo consentito**
  1. Viene registrata una multa di tipo *Transito in Eccesso* all’utente (attraverso il suo ID).

- **Requisiti speciali:** Bassa latenza

- **Frequenza:** Ogni qualvolta un utente si presenta al varco d’uscita

### 3.3 Caso D’Uso UC5

- **UC5:** Sanziona Utente
- **Portata:** Sistema Gestion ZTL
- **Livello:** Obiettivo utente
- **Pre-condizioni:** l’utente ha commesso un’irregolarità.
- **Garanzie di successo (Post-condizioni):** l’utente viene sanzionato, associando al suo dispositivo una sanzione.
- **Scenario principale di successo**
  - Se l’utente sta eseguendo l’ingresso:



- \* Se l'ingresso è fuori intervallo, il sistema eroga una sanzione "Transito Irregolare"
- \* Se l'ingresso è già avvenuto, e l'utente quindi sta eseguendo di nuovo l'accesso, il sistema eroga una sanzione "Transito Extra".
- \* Se l'ingresso avviene da terminale residente, viene erogata una multa di tipo "Accesso irregolare".
- Se l'utente sta eseguendo l'uscita:
  1. Se l'uscita avviene dopo lo scattare dell'ora, il sistema registra la sanzione con l'ammontare determinato secondo la relativa regola di business definita sotto. Verrà erogata una multa di tipo "Transito Extra".
- **Scenari alternativi**
  - **L'utente non ha dispositivo ID**
    - \* Viene registrata la targa attraverso la telecamera e viene notificato il transito, sia esso in ingresso o uscita, al sistema centrale da terminale.
- **Requisiti speciali:** Bassa latenza
- **Frequenza:** Sempre potenzialmente (ogni volta che viene commessa un'infrazione).

## 4 Regole di Business

Si rende necessario, alla luce dei requisiti e dei casi d'uso formulati, aggiungere ulteriori regole di business, specie per quanto riguarda il computo delle sanzioni. Definiamo quindi:

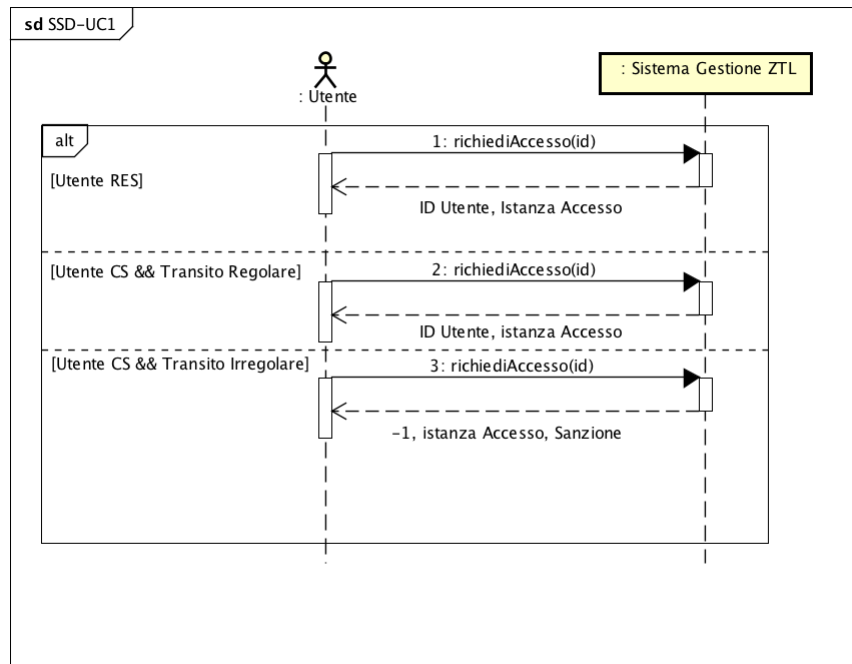
1. **Regola R6:** Il sistema sanziona esegue sanzioni relative al transito nella zona a traffico limitato.
2. **Regola R7:** L'entrata al di fuori dell'intervallo concesso agli utenti carico-scarico verrà multata con una sanzione "TRANSITO IRREGOLARE" dal valore di 50 euro.

3. **Regola R8:** L'entrata extra oltre a quella concessa giornalmente viene multata con una sanzione "TRANSITO EXTRA" dal valore di 30 euro.
4. **Regola R9:** L'uscita al di fuori dell'ora concessa ammonta a 15 euro e viene descritta come "TRANSITO ECCESSIVO".
5. **Regola R10:** Le multe che concernono gli utenti senza targa sono a carico degli impiegati comunali in quanto la sanzione, essendo grave, può portare a conseguenze fuori dalla portata dell'applicazione.
6. **Regola R11:** L'infrazione che commette l'utente carico-scarico che entra, anche nel regolare intervallo, da un terminale per i soli residenti viene indicata con "ACCESSO IRREGOLARE" e viene multata con 10 euro.

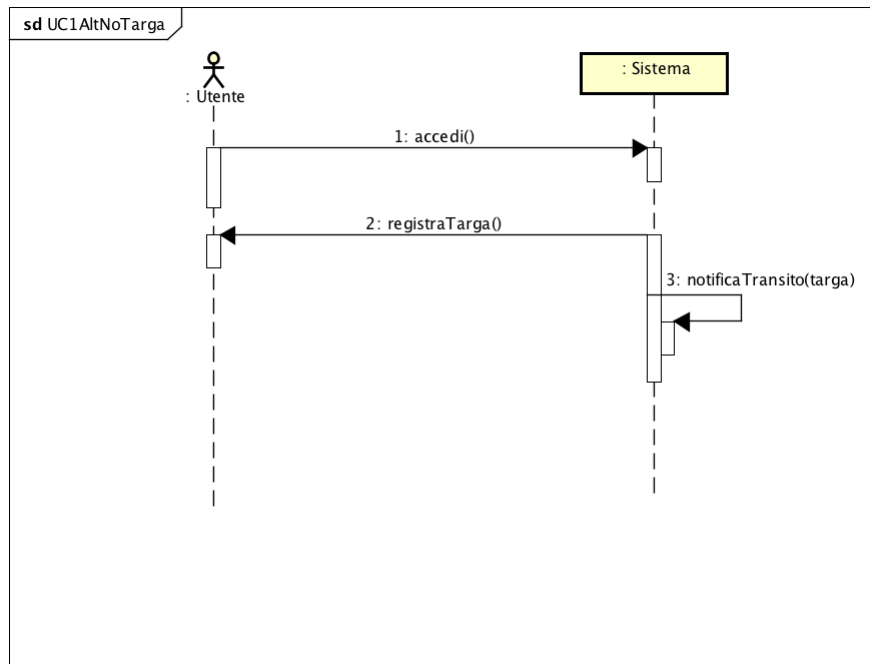
## 5 Modello di Dominio

Il modello di dominio per questa iterazione si può definire in termini di quello dell'iterazione precedente, con l'aggiunta della classe concettuale *Sanzione*, ed il raffinamento di alcune associazioni: per esempio, *Terminale* autentica *Autenticatore* attraverso la creazione di un'istanza di *Accesso* e sanziona essa attraverso la creazione di un'istanza *Sanzione*. Queste associazioni vengono rappresentate attraverso una linea che punta alla linea che connette le due classi da associare.



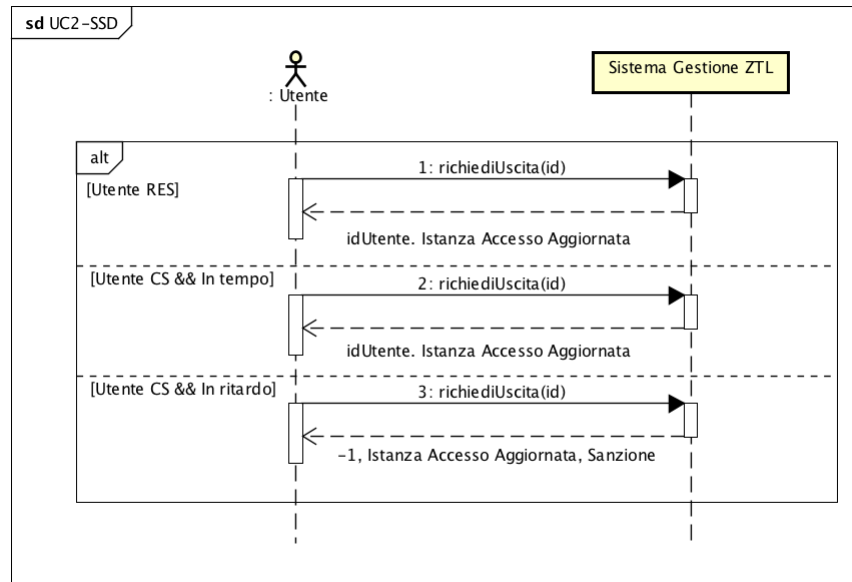


Introduciamo il caso in cui l'utente è senza dispositivo.

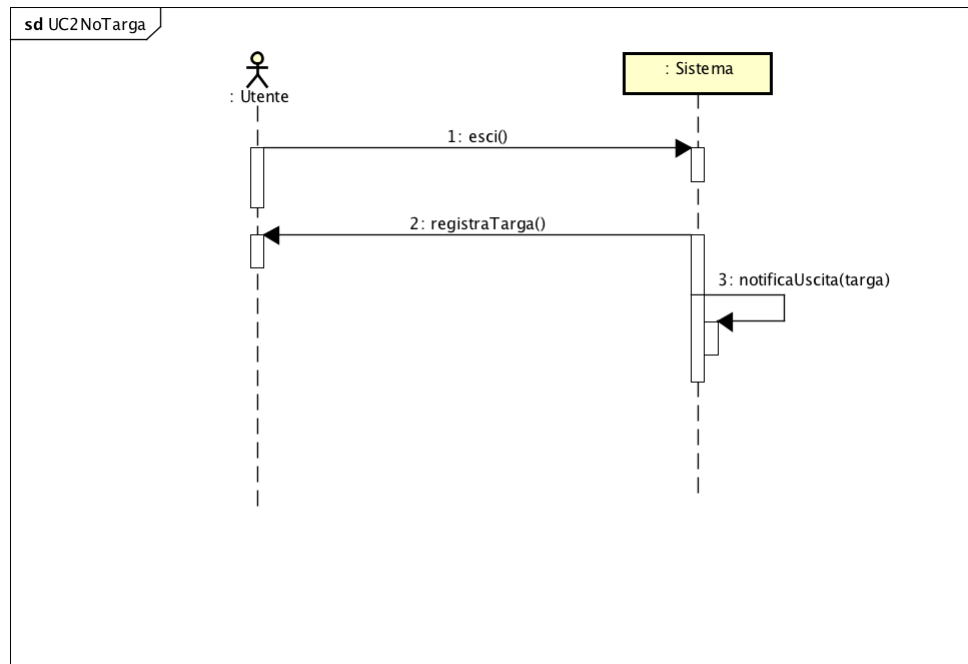


Da notare come si è voluto enfatizzare che la notifica del transito viene eseguita internamente al sistema.

## 6.2 UC2



Introduciamo anche qui il caso in cui l'utente è senza dispositivo.

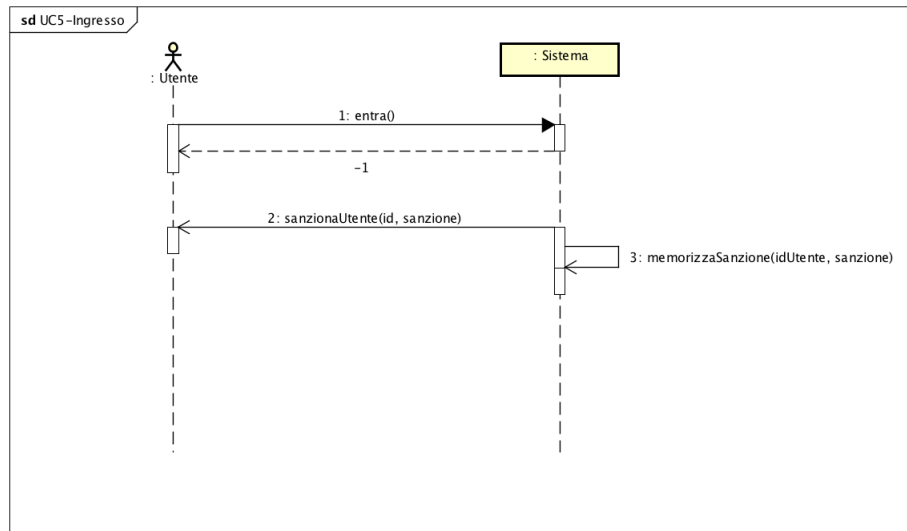


Anche qui enfatizziamo come la notifica d'uscita sia eseguita internamente.

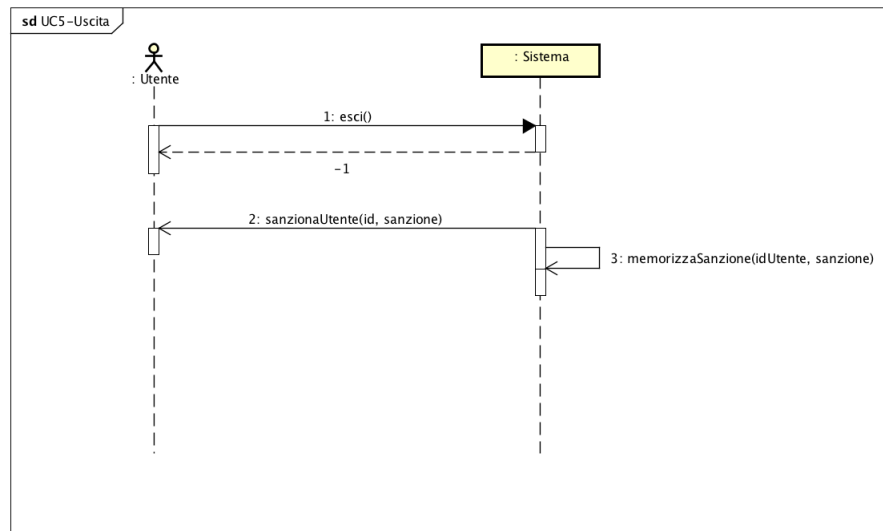
### 6.3 UC5 - Emetti Sanzione

Tale caso d'uso, a livello di interazioni, è collegato ai casi d'uso *UC1* ed *UC2*. Temporalmente esso avviene dopo la creazione dell'istanza d'ingresso/aggiornamento della stessa in uscita. Lo rappresenteremo dunque come un caso a se stante, riassumendo i casi d'uso *UC1* ed *UC2* con i messaggi **entra()** ed **esci()** rispettivamente.

Nel caso che l'utente stia commettendo un infrazione in ingresso, avremo il seguente diagramma:



Mentre nel caso d'uscita avremo:



L'emissione della sanzione consta di due azioni fondamentali: la notifica all'utente e la registrazione in una lista interna al sistema, in modo tale che si possa tenere traccia. Dettagli sull'implementazione vanno lasciati alla fase di progettazione, in questa fase vogliamo solo indicare quel che l'operazione fa, e nel prossimo paragrafo definiremo le operazioni più dettagliatamente.



## 7 Contratti delle Operazioni

### 7.1 CO5 - notificaIngressoTarga()

- **Operazione:** `notificaIngressoTarga(String : targa)`
- **Riferimenti:** Caso d'uso UC1
- **Pre-condizioni:** un utente senza dispositivo si trova in procinto di accedere nella zona a traffico limitato
- **Post-condizioni:** l'utente è entrato nella zona a traffico limitato, ed il suo transito irregolare è stato notificato

### 7.2 CO6 - notificaUscitaTarga()

- **Operazione:** `notificaUscitaTarga(String : targa)`
- **Riferimenti:** Caso d'uso UC2
- **Pre-condizioni:** un utente senza dispositivo si trova in procinto di uscire dalla zona a traffico limitato
- **Post-condizioni:** l'utente è uscito e l'uscita è stata notificata

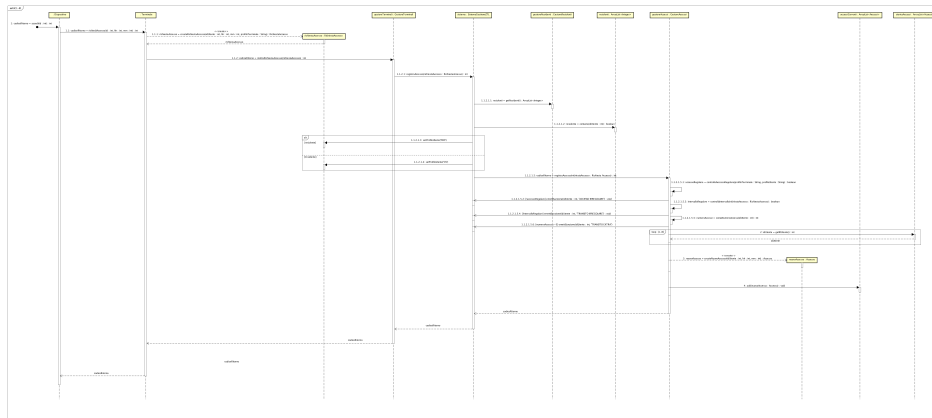
### 7.3 CO7 - emettiSanzione()

- **Operazione:** `emettiSanzione(int : idUtente, Sanzione : sanzione)`
- **Riferimenti:** Caso d'uso UC5
- **Pre-condizioni:** l'utente ha commesso un'infrazione in ingresso/uscita nella zona a traffico limitato
- **Post-condizioni:** l'utente è entrato/uscito, e la relativa sanzione è stata notificata e registrata

## 8 Diagrammi di Interazione

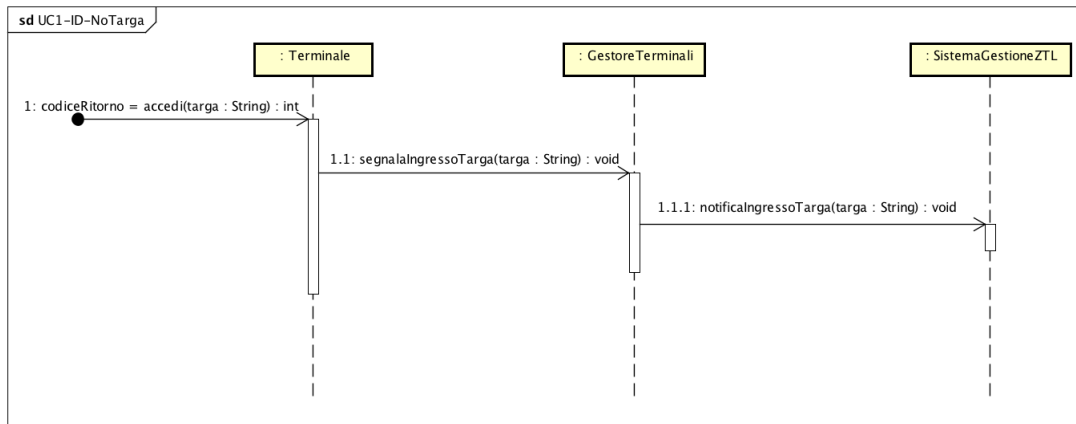
### 8.1 UC1 - Registra Ingresso

Se l'ingresso avviene regolarmente con dispositivo, vale il seguente diagramma di interazione. Vale sempre la regola secondo la quale la variabile `codiceRitorno` sia pari al codice ID dell'utente se l'accesso sta avvenendo in piena regolarità, -1 altrimenti. I controlli della regolarità del transito vengono fatti dal gestore degli accessi. Osserviamo come i gestori funzionano da intermediari, in quanto inoltrano le richieste dei singoli componenti verso il sistema centrale. Inoltre, si vuole notare l'applicazione del pattern *Command*. Si vuole delegare il ruolo dell'*Invoker* al terminale, ed il ruolo del *Receiver* al gestore dei terminali, che, come già detto, inoltrerà la richiesta al sistema centrale che medierà la stessa all'opportuno gestore. Viene passata l'istanza di richiesta stessa che il gestore degli accessi userà per inizializzare l'istanza d'accesso stesso.



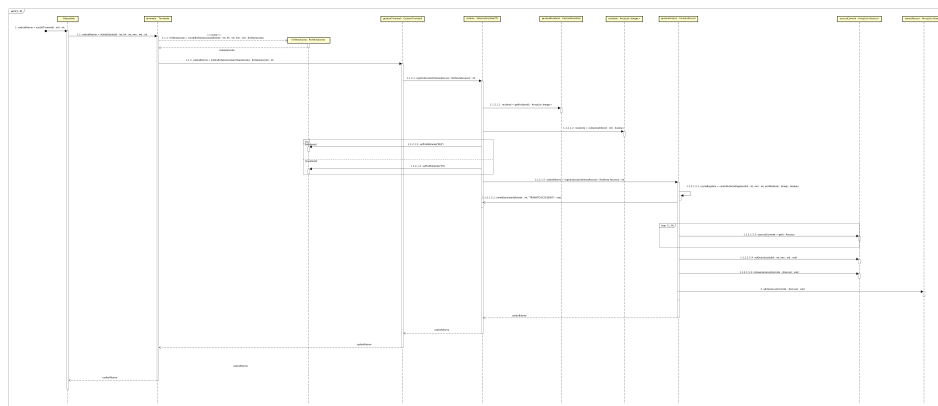
Per un maggior dettaglio grafico, fare riferimento alla figura UC1-ID.png od il file UC1-ID.asta.

E' di grande importanza mostrare le interazioni che avvengono all'ingresso dell'utente sprovvisto di dispositivo.



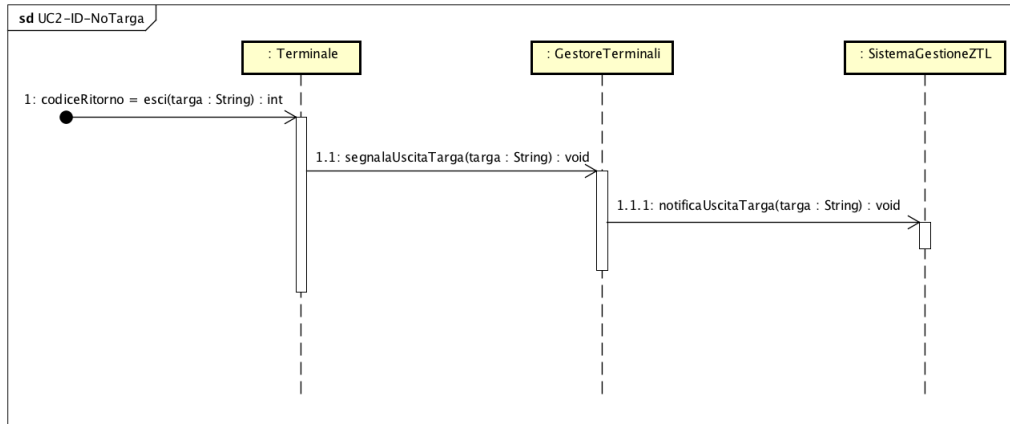
A differenza del caso precedente, qui il terminale segnala l'ingresso irregolare dell'utente senza dispositivo. Le conseguenze sono fuori dalla portata dell'applicazione, pertanto il nostro ruolo si limita a segnalare l'ingresso e l'uscita.

## 8.2 UC2 - Registra Uscita



Per un maggior dettaglio grafico, fare riferimento alla figura UC2-ID.png od il file UC2-ID.ast.

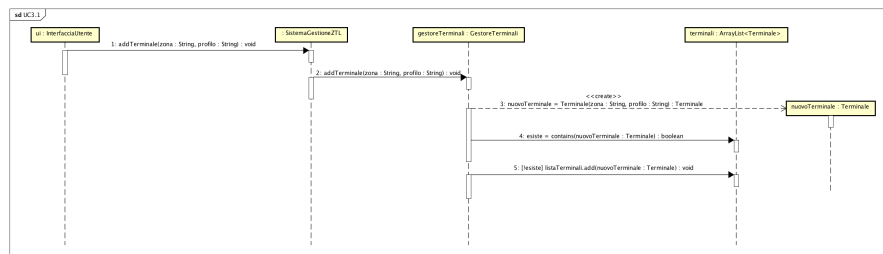
Similmente al caso d'uso precedente, mostriamo il diagramma d'interazione per lo scenario in cui l'utente è senza dispositivo.



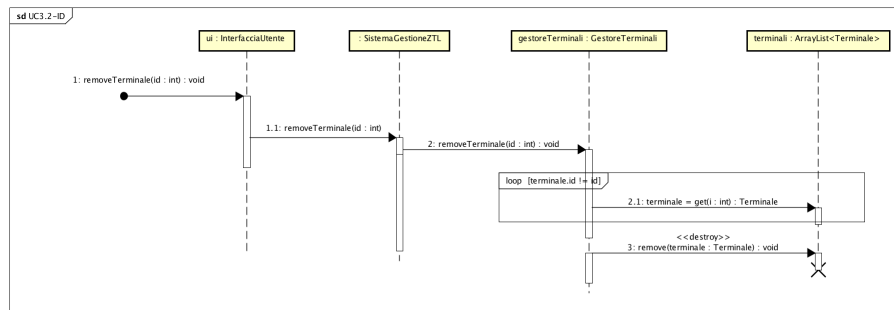
Si noti come, analogamente al caso precedente, in caso di infrazione, il gestore degli accessi chiama il sistema centrale. Il seguito a tale chiamata è illustrato nell'opportuno diagramma di interazione per il caso d'uso *UC5*.

### 8.3 UC3 - Gestisci Terminali

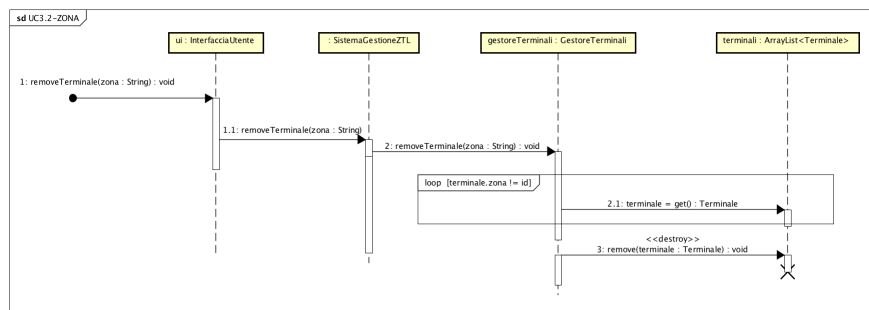
Il caso d'uso *UC3.1* presenta il seguente diagramma d'interazione



Se il terminale esiste già, verrà stampato a schermo un messaggio. Passiamo al caso d'uso *UC3.2*, quando utilizziamo la rimozione per codice id:

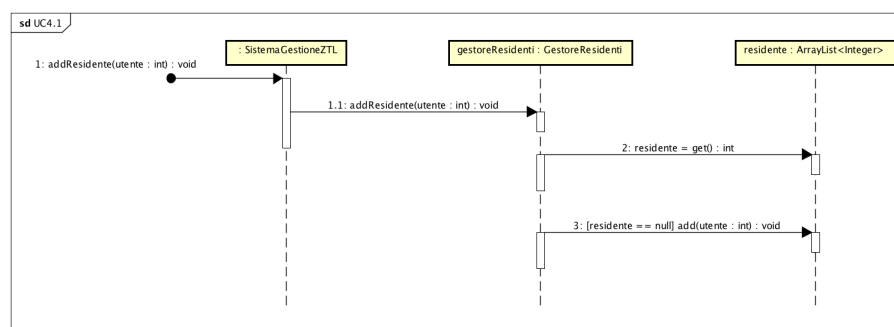


Quando eseguiamo la rimozione per zona, si avrà il seguente:

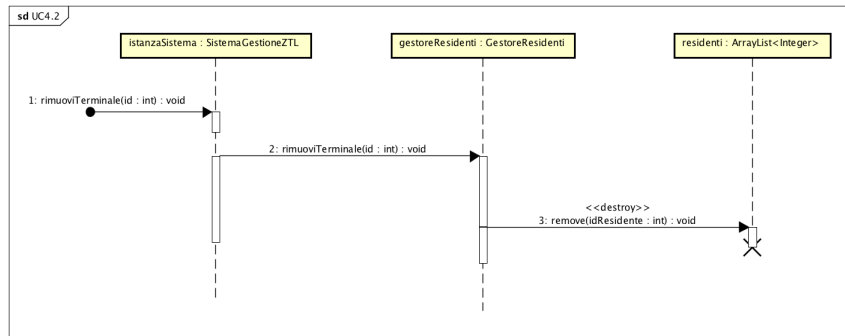


## 8.4 UC4 - Gestisci Residenti

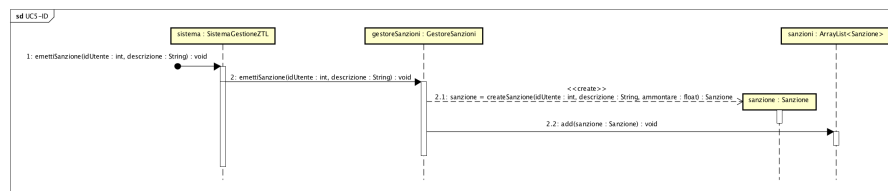
Per il caso d'uso *UC4.1* abbiamo il seguente diagramma di sequenza:



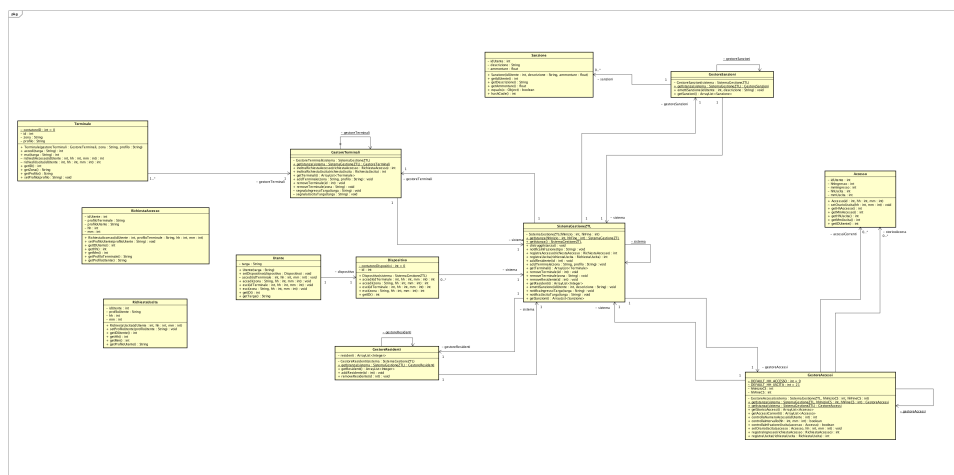
Mentre per il caso d'uso *UC4.2* abbiamo:



## 8.5 UC5 - Emetti Sanzione



## 9 Diagrammi delle Classi Progettuali



Consultare i file `DiagrammaClassi.asta` e/o `DiagrammaClassi.png` per un maggiore dettaglio grafico

## 10 Motivazioni progettuali

### 10.1 Riduzione dell'accoppiamento tramite *Controller*

Nonostante i sotto-componenti siano gestite dai componenti gestori principali, notiamo che esse rimangono accoppiate al sistema centrale. Pertanto le comunicazioni al sistema centrale passeranno dai gestori in modo tale che le componenti rimangano accoppiate solamente ai gestori, e non al sistema centrale.

### 10.2 Incapsulare richieste d'accesso/uscita

La richiesta viene incapsulata in una classe per consentire di astrarre ulteriormente l'applicazione e per permettere un'agevole creazione delle relative istanze.

### 10.3 Interfaccia Utente migliorata

Sebbene l'interfaccia sia a riga di comando, abbiamo aggiunto un'interfaccia funzionale, anch'essa componente che utilizzerà il sistema centrale come mediatore per le sue richieste. Questo ci permetterà di avere un sistema, dove grazie anche all'aggiunta della classe *Utente*, ci permetterà di simulare a pieno la zona a traffico limitato, ed il suo funzionamento a fronte delle varie interazioni. Le operazioni *CRUD* saranno eseguite tramite essa.

## 11 Casi di Test

In questa iterazione, verificheremo che le sanzioni ricevute siano quelle giuste e testeremo ancora che le funzioni che controllano gli intervalli siano corrette. In particolare, queste ultime verranno testate con opportuni test unitari, mentre le altre con test di integrazione, dato che coinvolgono diverse parti di sistema.

A differenza delle altre iterazioni, vogliamo incrementare il numero di test in modo da garantire un più ampio test coverage, per quanto riguarda i test unitari e di integrazione. La presenza di un ambiente di simulazione permetterà di eseguire altre simulazioni per verificare che il sistema funzioni in maniera soddisfacente.

## 11.1 Test Unitari della classe *GestoreAccessi*

Di questa classe, è di nostro interesse testare i metodi `controllaIntervallo()` e `controllaInfrazioneUscita()`. Ricordiamo che, nei test, il sistema ammette utenti carico scarico dalle 9 alle 21. I casi di test scelti sono i seguenti:

- L'orario 9:00 dovrà dare `true`
- L'orario 8:59 dovrà dare `false`
- L'orario 21:00 dovrà dare `true`
- L'orario 21:01 dovrà dare `false`

Scegliamo, analogamente, dei casi di test che garantiscano che il codice sia raggiunto in una buona percentuale:

- L'intervallo 9:00 - 10:00 dovrà dare `true`
- L'intervallo 9:00 - 10:01 dovrà dare `false`
- L'intervallo 9:00 - 9:59 dovrà dare `true`
- L'intervallo 21:00 - 22:00 dovrà dare `true`
- L'intervallo 21:00 - 22:01 dovrà dare `false`
- L'intervallo 21:00 - 21:59 dovrà dare `true`

## 11.2 Test d'integrazione di *GestoreSanzioni*

Per i seguenti test, vengono testate, per un utente ipotetico di ID 0, che le sanzioni emesse per lui in seguito a tutte le possibili cause, siano presenti nella lista. Per fare ciò si istanziano le sanzioni dalla classe *Sanzione* e viene controllata la lista per verificare che esistano le sanzioni istanziate, con i corretti valori.



### 11.3 Test d'integrazione di *Dispositivo*

I casi che andremo a verificare per `accedi()` sono:

- Utente CS entra da terminale CS alle 8:59, e darà -1
- Utente RES entra da terminale CS alle 8:59, e darà l'ID
- Utente CS entra da terminale CS alle 9:00, e darà l'ID
- Utente CS entra da terminale CS alle 21:00, e darà l'ID
- Utente CS entra da terminale CS alle 21:01, e darà -1
- Utente RES entra da terminale RES alle 2:59, e darà l'ID
- Utente CS entra da terminale RES alle 11:00, e darà -1

Mentre i casi che verificheremo per `esci()` sono:

- Utente CS (9:11 - 10:11), che darà l'ID
- Utente CS (21:00 - 22:01), che darà -1
- Utente CS (21:00 - 22:00), che darà l'ID
- Utente CS (11:00 - 17:00), che darà l'ID