Sistema Gestione ZTL

Riccardo Maria Pesce

Anno Accademico 2019-2020

Abstract

Durante questa seconda iterazione, verranno inizialmente aggiornati/rifiniti i requisiti ed i casi d'uso, in modo da includere gli utenti carico-scarico e chiarificare alcuni punti definiti vagamente nelle fasi precedenti.

Contenuti

1 Resoconto Iterazione 1

1.1 Resoconto incontro

Durante l'ultimo incontro con i rappresentanti del comune per il quale stiamo realizzando il software, abbiamo avuto un feedback abbastanza positivo. Tuttavia, è stato stabilito che le API del dispositivo Telepass vengono sviluppate da noi.

1.2 Resoconto proof-of-concept

Il cliente ha potuto testare il sistema con delle classi opportune, e ha verificato che il sistema risponde come previsto.
Pertanto eseguiremo, oltre che un lavoro di incremento di funzionalità, anche un lavoro di rifinitura e refactoring.

2 Piano per la seconda iterazione

In questa seconda iterazioni, verranno rivisitati i modelli di analisi (dominio e sequenza) e progettazione (interazione e classi).

- Implementeremo lo scenario alternativo del caso d'uso UC1, ove a richiedere accesso è un utente carico-scarico
- Implementeremo lo scenario alternativo del caso d'uso UC2, dove, anche qui, a richiedere l'uscita è un utente carico-scarico
- Implementeremo la classe *Dispositivo* che accompagnerà la classe *Utente* utilizzata solo come Test Driver (questa volta farà parte del sistema), responsabile dell'accesso. Occorre precisare che la classe di test Utente verrà utilizzata, mentre la classe dispositivo sarà responsabile dell'autenticazione e sarà interna al sistema
- Implementeremo il caso d'uso UC3.1 *Aggiungi Terminale*
- Implementeremo il caso d'uso UC4.1 *Aggiungi Residente*

Per rendere l'iterazione breve e timeboxed, e quindi essere in linea con il prossimo incontro, imposteremo un intervallo orario per utenti carico-scarico predefinito, definito nel sistema centrale. Delegheremo ad ogni terminale, in

una futura iterazione, la scelta customizzata dell'intervallo di transito per utenti carico-scarico.

3 Requisiti Funzionali

Vogliamo riportare i requisiti, gia comunque accennati in fase di ideazione, e leggermente modificati per chiarificare alcune funzionalità. Tali saranno i requisiti implementati in questa iterazione.

- L'impiegato deve poter aggiungere terminali al sistema centrale. Tali terminali avranno un codice id univoco, una zona (ossia il valico nella quale operano) ed un profilo (residente, se permette accesso solo ai residenti, carico-scarico ad entrambe le tipologie di utenti)
- L'utente residente può accedere ed uscire senza nessun limite dalla zona a traffico limitato
- L'utente carico-scarico, quando accede registra un'istanza di *Accesso*, con orario d'ingresso. All'uscita l'istanza viene aggiornata con l'orario d'uscita. Sia all'ingresso che all'uscita viene controllato che il transito sia avvenuto regolarmente. In caso contrario, verrà stampata a video l'infrazione commessa
- L'utente dispone di un dispositivo elettronico per fruire dell'accesso (il dispositivo non è personale, ma, proprio come il telepass, purchè uno lo ha, può usarlo a proprio piacimento e nei limiti imposti dal tipo di utenza).

4 Casi d'uso

4.1 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 gia 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 notificato l'ID e l'infrazione *ingresso irregolare*
  - **Il transito sta avvenendo in un intervallo non consentito**
    1. Viene notificato l'ID e l'infrazione *intervallo non consentito*
- **Requisiti speciali:** Bassa latenza
- **Frequenza:** Ogni qualvolta un utente si presenta al varco

4.2 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, avvicinatosi 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 semplicemnte 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
- **Scenari alternativi**
  - **L'utente è rimasto per più del tempo consentito**
    1. Viene notificato l'ID e una multa di tipo *transito in eccesso*.
- **Requisiti speciali:** Bassa latenza
- **Frequenza:** Ogni qualvolta un utente si presenta al varco d'uscita

4.3 Caso d'uso UC3.1 (*Aggiungi Terminale*)

Attore primario: Impiegato

- *Scenari principali di successo*
  - **Aggiungi Terminale**
    1. L'impiegato immette il codice identificativo del terminale da aggiungere
    2. Il sistema, assicuratosi che tale codice non appartenga ad altro terminale installato, richiede all'impiegato il profilo da associare al nuovo terminale
    3. Si possono verificare le seguenti opzioni:
       * Il profilo richiesto è *carico-scarico*. In questo caso, se non esiste un terminale con tale profilo, la richiesta viene accettata. Altrimenti, verrà chiesto di modificare il profilo dell'attuale terminale *carico-scarico* prima di

procedere. Questo perchè deve essere solo uno il terminale autorizzato al riconoscimento di tale categoria di utenti

    * Il profilo richiesto è *residente*. In tal caso, dato che non esistono limitazioni circa tale categoria, il profilo viene impostato correttamente e nessun'altra azione è richiesta

- *Scenari alternativi*
  - **Aggiungi Terminale**
    * Se il numero inserito è gia presente, allora il sistema ritornerà un messaggio d'errore
    * Se si sta aggiungendo un terminale di profilo carico-scarico dove è gia presente un terminale di profilo residente, quest'ultimo andrà a sostituire il primo
    * Se invece si sta aggiungendo un terminale di profilo residente laddove è presente un terminale di profilo carico-scarico
      1. Si verifica che almeno un terminale, oltre quello da modificare, sia di profilo carico-scarico
      2. Se affermativo, si procede alla modifica, altrimenti viene tornato un messaggio d'errore

4.4 Caso d'uso UC4.1 (*Aggiungi Residente*)

Attore primario: Impiegato

- *Scenari principali di successo*
  - **Aggiungi Residente**
    1. L'impiegato registra il codice identificativo del telepass nella lista dei residenti
    2. Il sistema, assicuratosi che l'utente il codice non esista gia, lo registrerà con successo.
- *Scenari alternativi*
  - **Aggiungi Residente**
    1. Se l'utente gia esiste, la richiesta verrà semplicemnte ignorata con un messaggio d'errore.

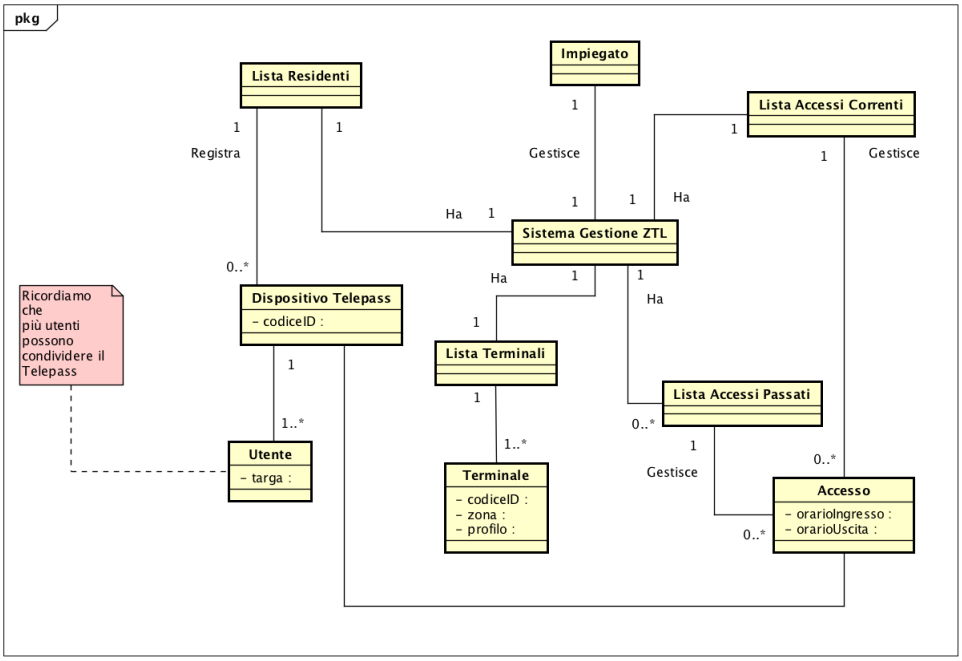
5 Regole di Business

In tale iterazione, si sono identificate le seguenti regole di business:

- **R3:** il telepass non è personale, e può essere condiviso
- **R4:** i terminali di profilo carico-scarico possono autenticare all'ingresso sia i profili residente che i profili carico-scarico
- **R5:** tutti gli utenti possono usufruire di qualsiasi terminale all'uscita

6 Modello di Dominio

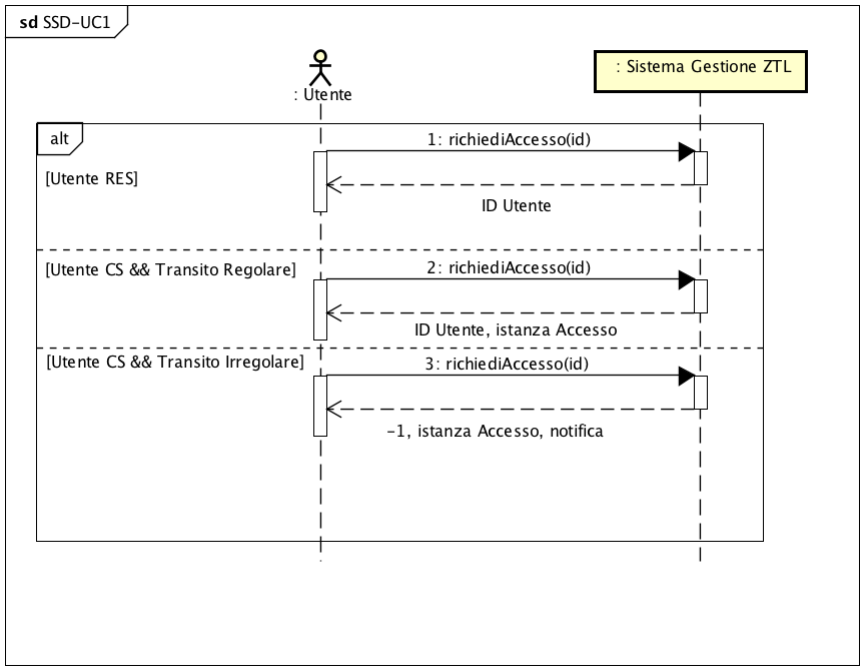
Il modello di dominio, per tale iterazione, sarà il seguente.



7 Diagrammi di Sequenza di Sistema

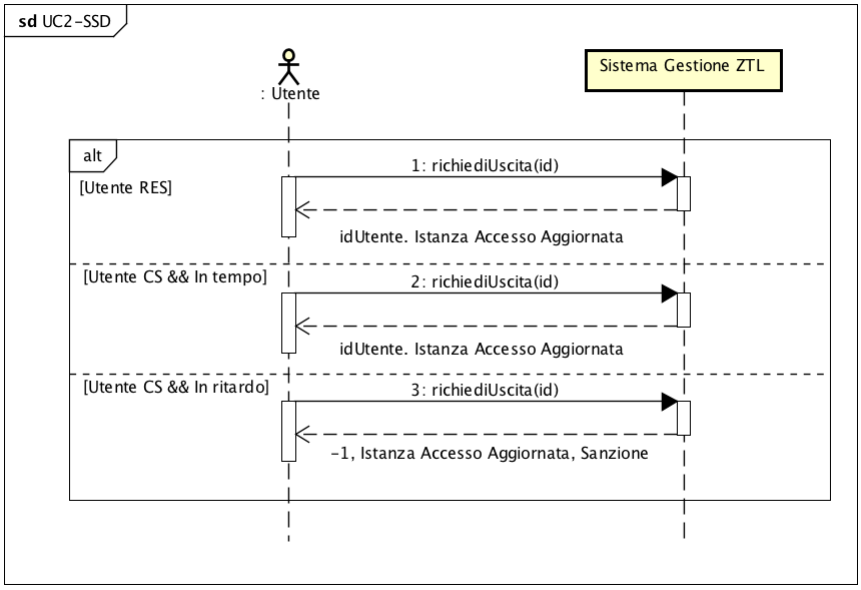
7.1 UC1

Per accomodare la presenza di utenti carico-scarico, il diagramma di sequenza è diviso in tre sezioni, che rappresentano i messaggi ritornati nel caso rispettivamente in cui l'utente sia residente, non residente con transito regolare e non residente con transito irregolare.

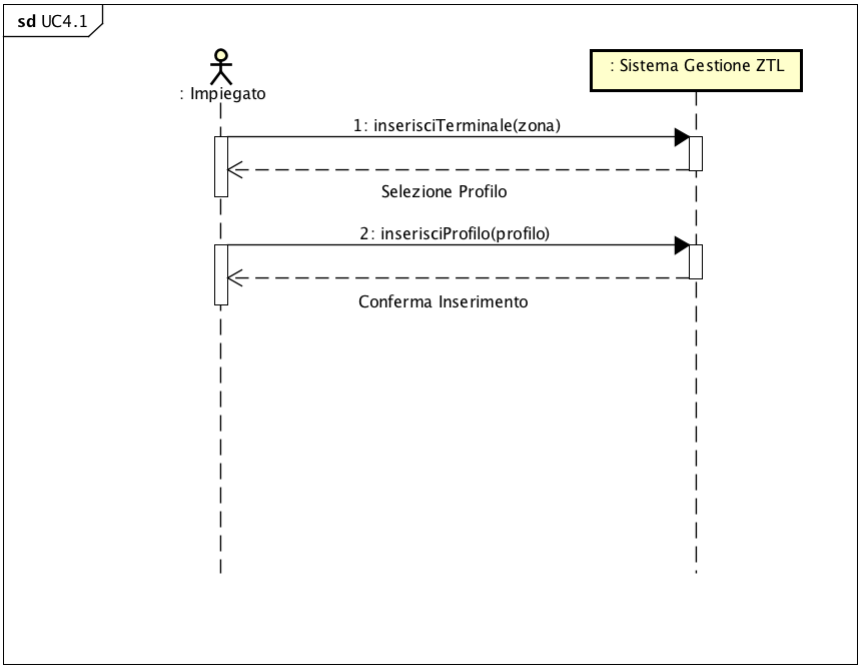


7.2 UC2

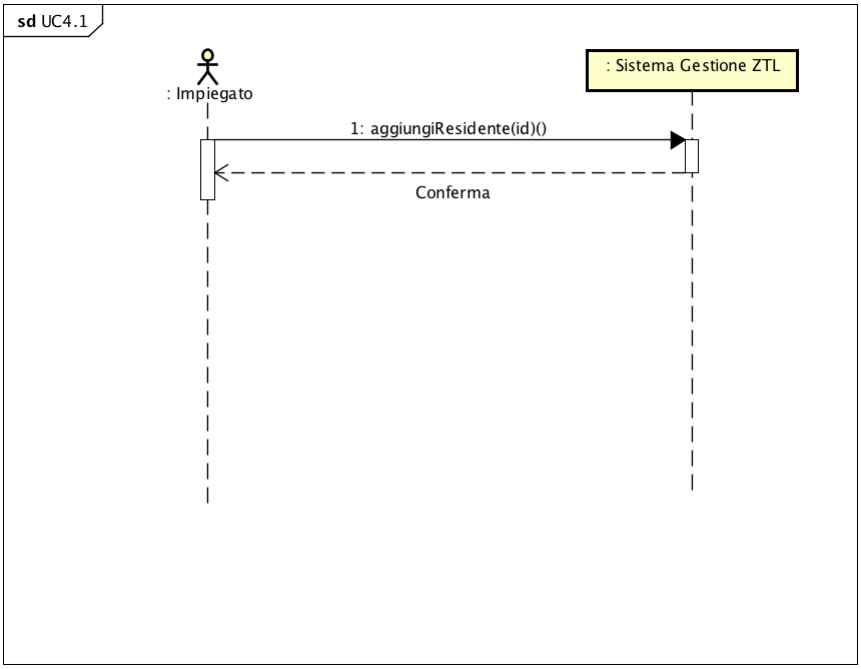
Come per il caso d'uso *UC1*, includiamo nel diagramma di sequenza il caso in cui l'utente carico-scarico esce senza aver commesso irregolarità ed il caso ove invece compie un'irregolarità.


sd UC2-SSD
: Utente
Sistema Gestione ZTL
alt
[Utente RES]
1: richiediUscita(id)
idUtente. Istanza Accesso Aggiornata
[Utente CS && In tempo]
2: richiediUscita(id)
idUtente. Istanza Accesso Aggiornata
[Utente CS && In ritardo]
3: richiediUscita(id)
-1, Istanza Accesso Aggiornata, Sanzione

7.3 UC3.1


sd UC4.1
: Impiegato
: Sistema Gestione ZTL
1: inserisciTerminale(zona)
Selezione Profilo
2: inserisciProfilo(profilo)
Conferma Inserimento

7.4 UC4.1



8 Contratti delle operazioni

Per accomodare gli utenti carico-scarico, modifichiamo i contratti precedenti e le relative operazioni, dato che adesso gli accessi per tale tipologia di utenti devono essere registrati

8.1 CO1

- **Operazione:** `richiediAccesso(idUtente : int, hh : int, mm : int)`
- **Riferimenti:** Caso d'uso UC1
- **Pre-condizioni:** un utente si trova in procinto di accedere
- **Post-condizioni:** l'utente è entrato nella zona a traffico limitato e, se carico-scarico, l'ingresso è stato registrato

8.2 CO2

- **Operazione:** `richiediUscita(idUtente : int, hh : int, mm : int)`
- **Riferimenti:** Caso d'uso UC1
- **Pre-condizioni:** un utente è procinto di uscire dalla zona a traffico limitato
- **Post-condizioni:** l'utente è uscito dalla zona a traffico limitato e, se carico scarico, l'istanza d'accesso è stata aggiornata e registrata nello storico degli accessi

8.3 CO3

- **Operazione:** `registraAccesso(idUtente : int, hh : int, mm : int)`
- **Riferimenti:** Caso d'uso UC1
- **Pre-condizioni:** un utente carico-scarico ha eseguito l'accesso
- **Post-condizioni:** l'utente è stato registrato nel sistema come utente in transito

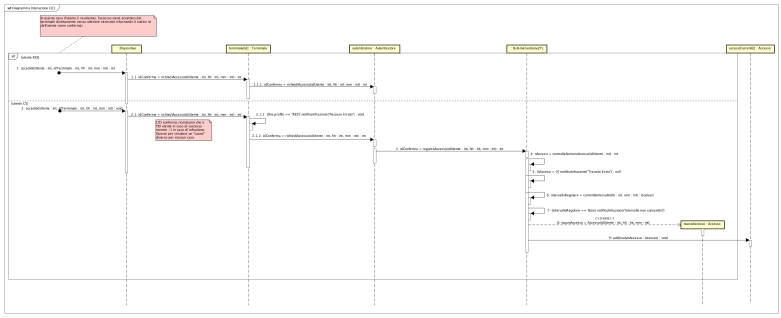
8.4 CO4

- **Operazione:** `registraUscita(idUtente : int, hh : int, mm : int)`
- **Riferimenti:** Caso d'uso UC2
- **Pre-condizioni:** un utente carico-scarico ha lasciato la zona a traffico limitato
- **Post-condizioni:** l'accesso viene modificato con l'orario di uscita e viene inserito in uno storico di accessi

9 Diagrammi di Interazione

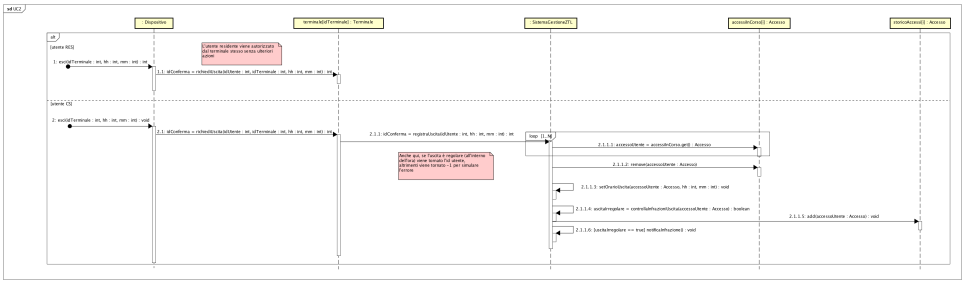
9.1 UC1

Riportiamo di seguito il diagramma per il Caso D'Uso *UC1* riportando le casistiche nel caso in cui l'utente sia residente o carico-scarico.



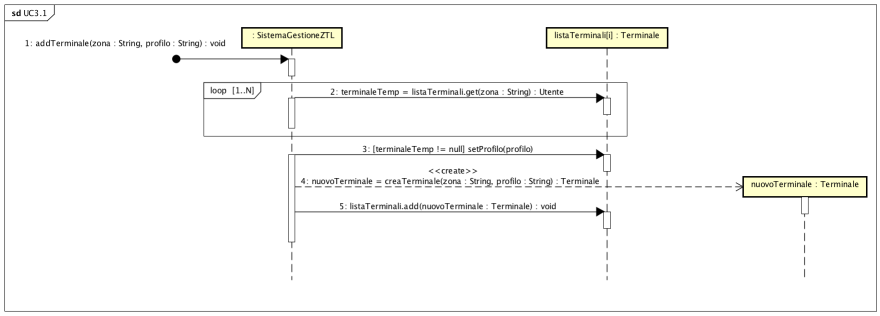
9.2 UC2

Illustriamo di seguito il diagramma per il Caso D'Uso *UC2* riportando anche qui le casistiche nel caso in cui l'utente sia residente o carico-scarico.



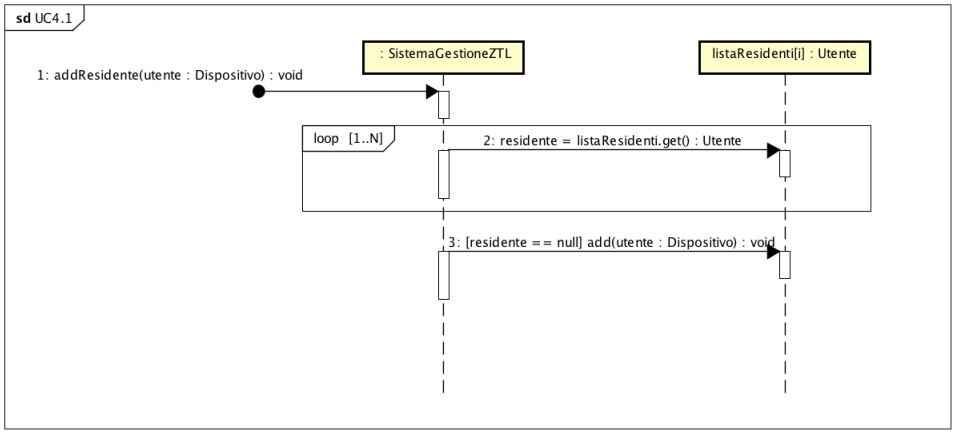
9.3 UC3.1

Illustriamo di seguito il diagramma per il Caso D'Uso *UC3.1 - Aggiungi Terminale*. Aggiungiamo i dati, verifichiamo l'esistenza ed a seconda, o aggiorneremo il profilo, oppure creiamo un nuovo terminale e lo aggiungiamo. modo per identificarlo alla ZTL.

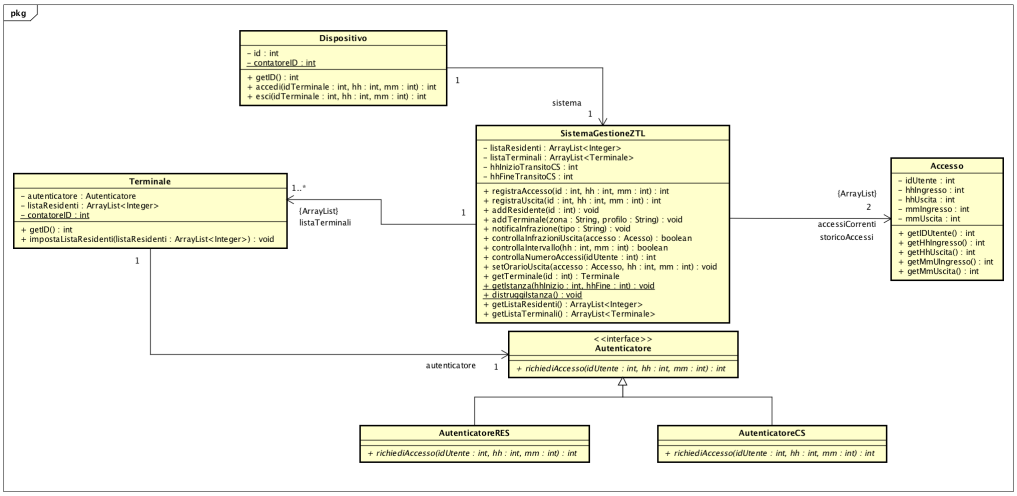


9.4 UC4.1

Illustriamo di seguito il diagramma per il Caso D'Uso *UC4.1 - Aggiungi Residente*. Notiamo che l'utente è di tipo *Dispotivo*, in quanto esso è l'unico modo per identificarlo alla ZTL.



10 Diagramma delle Classi Progettuali



11 Motivazioni progettuali

11.1 Meccanismo di sottoscrizione con pattern *Observer*

Quando un utente viene aggiunto alla lista dei residenti, questa deve essere aggiornata in tutti i terminali. Utilizziamo il pattern *Observer* per implementare un efficace meccanismo di sottoscrizione dove ad ogni utente aggiunto, viene inviata la "notifica" ai terminali.

11.2 Encapsulamento dell'algoritmo di riconoscimento con pattern *Strategy*

Il terminale, a seconda del profilo, implementa il pattern Strategy, in modo che a profili diversi corrispondono strategie diverse.

11.3 Distribuzione delle responsabilità

Utilizzando un insieme di pattern *GRASP*, abbiamo voluto delegare la creazione di Terminali e Residenti al sistema centrale dato che sembra il più

adeguato a tale mansione *(Creator)*. Stessa cosa per le istanze della classe *Accesso*, che vengono delegate al sistema centrale dato che le registra.

12 Casi di Test

12.1 Test unitari per `richiediAccesso`

Testiamo suddetto metodo, utilizzando le seguenti combinazioni:

- Utente Residente, Terminale Residente
- Utente Carico-Scarico, Terminale Residente
- Utente Carico-Scarico, Terminale Carico-Scarico
- Utente Residente, Terminale Carico-Scarico

I primi due verranno eseguiti per la classe *AutenticazioneRES*, e verrà enfatizzato nel test il fatto che i residenti hanno accesso illimitato (viene fatto accedere alle 2:00 di notte) mentre gli utenti carico scarico nemmeno nell'intervallo consentito (9:00-21:00) possono accedere.
Gli ultimi due invece afferiscono alla classe *AutenticazioneCS*, e testeremo che nel primo caso, l'ingresso alle 21:20 sia illecito e l'ingresso alle 11 lecito. Nel secondo caso, alle 2 di notte, mostreremo che l'ingresso sia lecito.
Per gli intervalli, testiamo un utente carico-scarico all'interno, all'esterno, e proprio al punto di uscita, e ci aspetteremo che in questo ultimo caso non succeda nulla.
Vogliamo anche testare un accesso multiplo di un utente carico-scarico in modo da verificare che esso venga notificato.

12.2 Test unitari per la classe `SistemaGestioneZTL`

In tale classe, vogliamo verificare che i metodi `controllaIntervallo` e `controllaInfrazioneUscita` siano esatti. In particolare, il primo deve ritornare `false` se l'ora data in input si trova al di fuori dell'intervallo mentre il secondo deve controllare, data l'istanza d'accesso, che l'utente sia uscito entro l'ora e non oltre lo scattare dell'ora: un'ora ed un minuto è da considerarsi uscita irregolare. In particolare:

- La zona a traffico limitato consentirà l'accesso dalle 9 alle 21.

- L'orario 8:30 dovrà restituire `false` alla chiamata a `controllaIntervallo`.
- L'orario 9:00 dovrà restituire `true` alla chiamata a `controllaIntervallo`.
- L'orario 21:00 dovrà restituire `true` alla chiamata a `controllaIntervallo`.
- Viene creato un accesso alle 9:30 che esce alle 10:31. Tale dovrà restituire `true` alla chiamata a `controllaInfrazioneUscita`. Lo steso viene fatto uscire alle 10:30, e dovrà restituire *false*.

12.3 Test Strutturali di Sistema

Come al solito, il cliente (comune) eseguirà un test attraverso la classe `TestDriver` usando la classe `Utente` che simula la vettura. In tal modo sarà lui stesso a rendersi conto dell'andazzo del progetto e darci un feedback.