

UNIVERSITÀ
DI PISA

ANNO ACCADEMICO 2020-2021

INGEGNERIA DEL SOFTWARE

PROGETTO UN GIORNO AL MUSEO

MATTEO CECCHERINI - 607086

FABIO MARCHETTI - 581806

GIORGIO MAZZEI - 607468

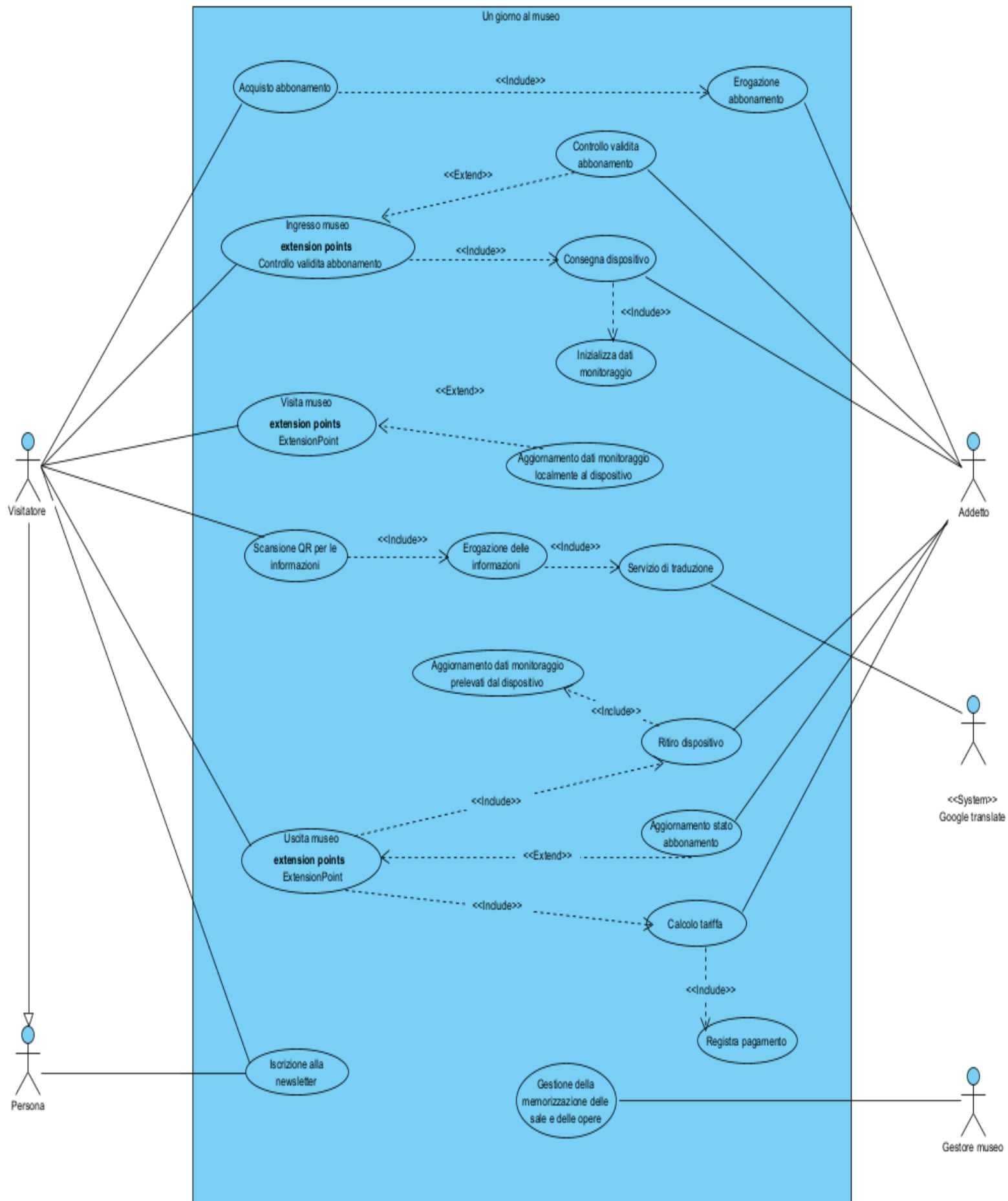
TOMMASO SOMIGLI - 598357



SOMMARIO

Esercizio 1	3
Narrativa caso d'uso "Ingresso museo"	4
Esercizio 2	7
Esercizio 3	8
Esercizio 4	9
Esercizio 5	12
Esercizio 6	13
Esercizio 7	14
Esercizio 8	15
Esercizio 9	16
Esercizio 9a	16
Variabili casi di test	16
Stub	16
Classi di equivalenza	17
Esercizio 9b	18
Esercizio 9c	19
Batteria di test	19
Esercizio 9d	19

Esercizio 1



Narrativa caso d'uso "Ingresso museo"

Caso d'uso : Ingresso museo

Breve descrizione	Procedure da effettuare una volta che il visitatore si presenta alla biglietteria con l'intenzione di entrare nel museo
Attori primari	Visitatore
Attori secondari	Addetto
Precondizioni	Il visitatore si presenta in biglietteria
Sequenza principale degli eventi	<ol style="list-style-type: none">Se si dichiara abbonato<ol style="list-style-type: none">extends Controlla validità abbonamento<ol style="list-style-type: none">Se è valido, torna all'inizioaltrimenti vai al punto 2.<i>Altrimenti</i><ol style="list-style-type: none">Il visitatore sceglie il tipo di tariffa che vuole pagare (bianca o verde)include Consegna dispositivo
Postcondizione	Al visitatore viene consegnato il dispositivo e può entrare nel museo
Sequenza alternativa degli eventi	L'abbonamento risulta scaduto o completato

Caso d'uso : Consegna dispositivo

Breve descrizione	Consegna dispositivo comprende l'inizializzazione e la consegna del dispositivo al cliente da parte dell'addetto.
Attori primari	Addetto
Attori secondari	Visitatore
Precondizioni	Il visitatore ha scelto la tariffa o si è identificato (se abbonato).
Sequenza principale degli eventi	<ol style="list-style-type: none">1. Inizializza i dati dispositivo in base alla tariffa/abbonamento2. include Inizializza dati monitoraggio3. consegna il dispositivo al visitatore
Postcondizione	Il visitatore ha il dispositivo ed entra nella prima sala

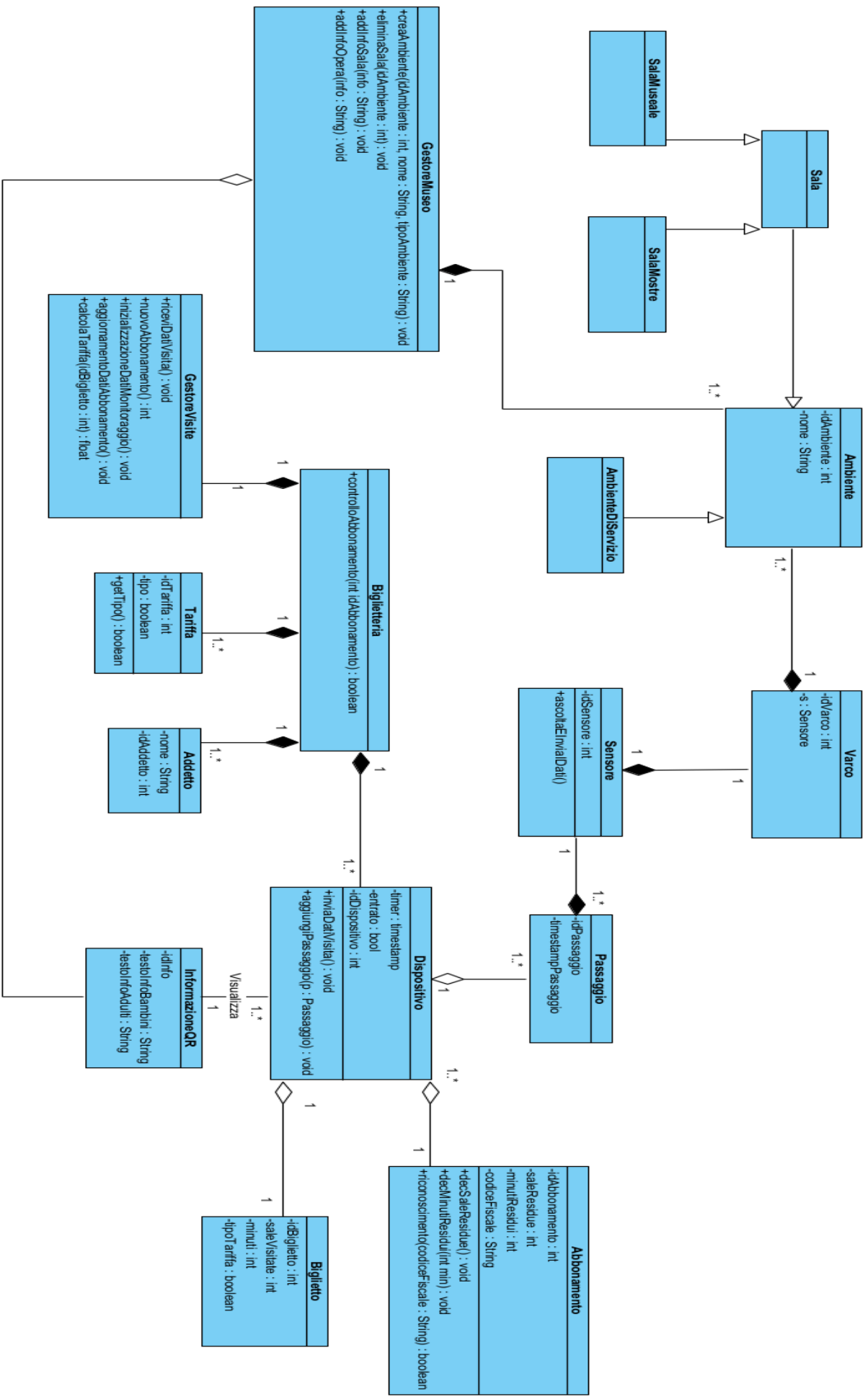
Caso d'uso : Controllo validità abbonamento

Breve descrizione	Serve a controllare che l'abbonamento specificato dal visitatore esista e sia in corso di validità.
Attori primari	Addetto
Attori secondari	Visitatore
Precondizioni	Il visitatore ha scelto la tariffa o si è identificato (se abbonato).
Sequenza principale degli eventi	L'addetto verifica la presenza del codice indicato dal visitatore all'interno del sistema e che sia in corso di validità (se abbonato)
Postcondizione	
Sequenza alternativa degli eventi	

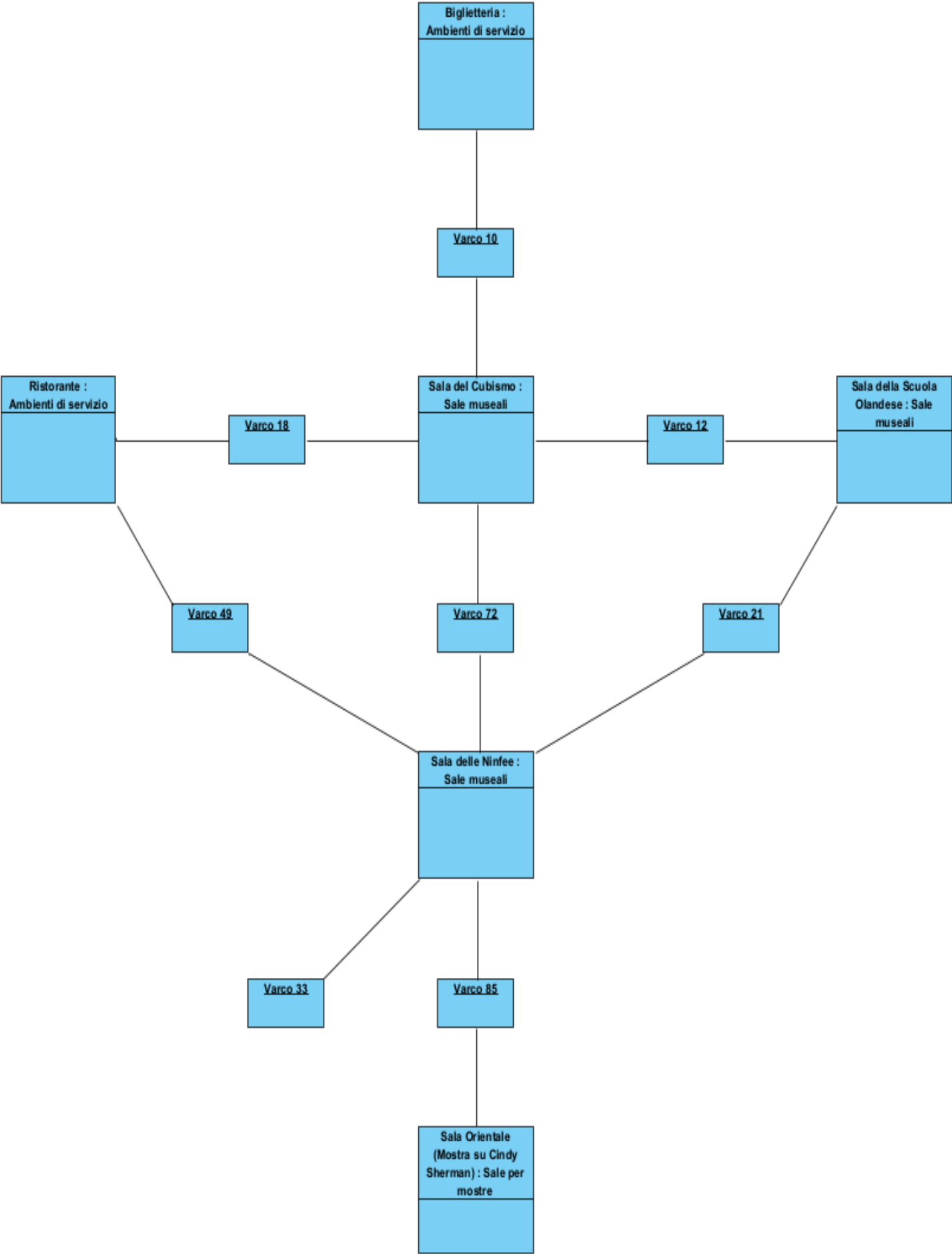
Caso d'uso : Inizializza dati monitoraggio

Breve descrizione	Inizializza dati monitoraggio permette di memorizzare varie informazioni relative ad abbonamenti e biglietti, tra cui anche i varchi attraversati e i minuti passati durante la visita.
Attori primari	Nessuno
Attori secondari	Addetto
Precondizioni	Il visitatore ha scelto la tariffa o si è identificato (se abbonato).
Sequenza principale degli eventi	<ul style="list-style-type: none">a. Se non è stato scelto un abbonamento, memorizza un nuovo biglietto all'interno del sistema avente valori di default iniziali, con indicazione del tipo di tariffa. Questi dati saranno aggiornati durante e alla fine della visita.b. Altrimenti, riduce di uno le sale visitabili dall'abbonamento indicato solo se la prima sala a cui si accede è effettivamente una sala.
Postcondizione	I dati sull'abbonamento/biglietto sono stati inizializzati

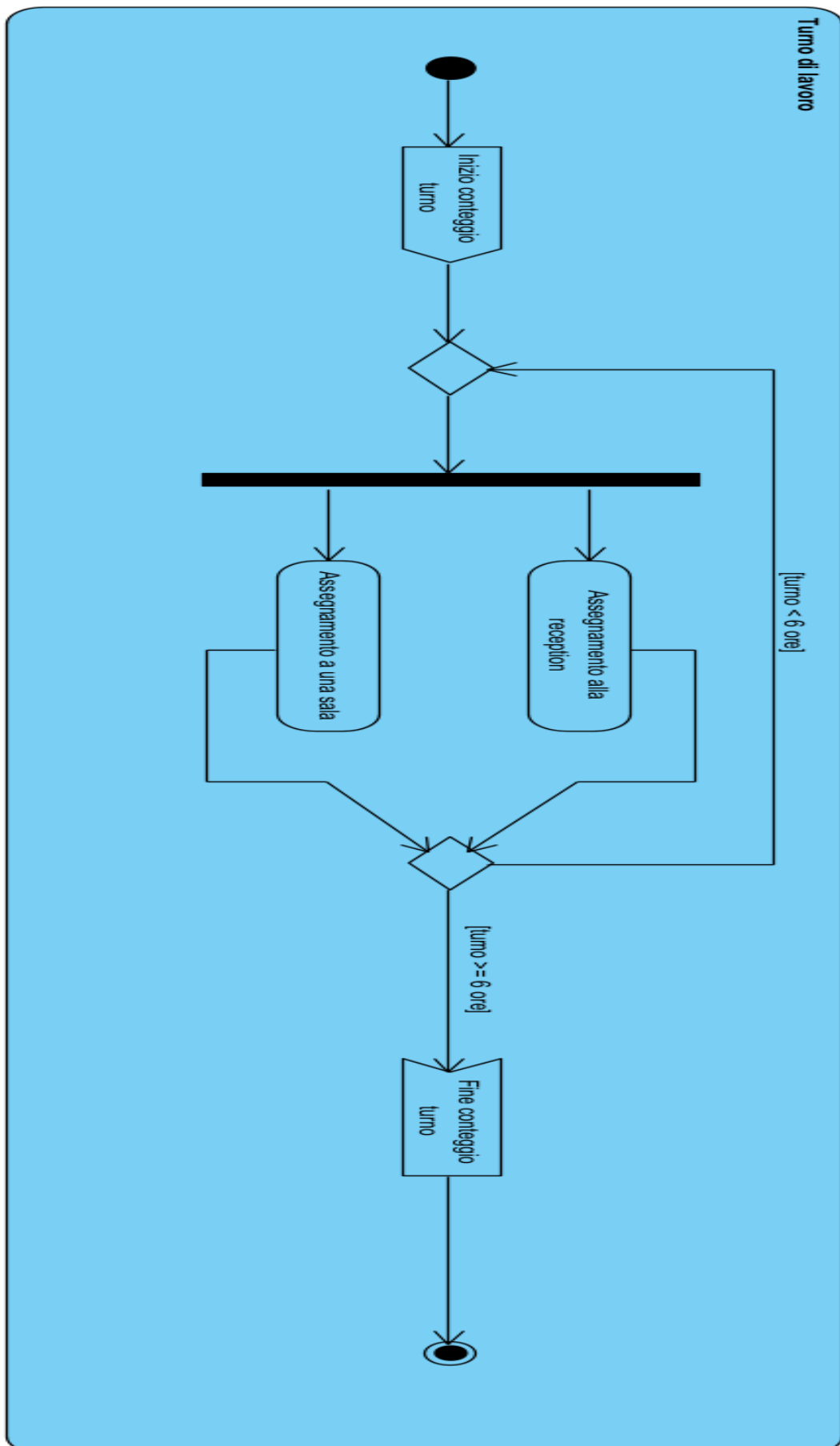
Esercizio 2



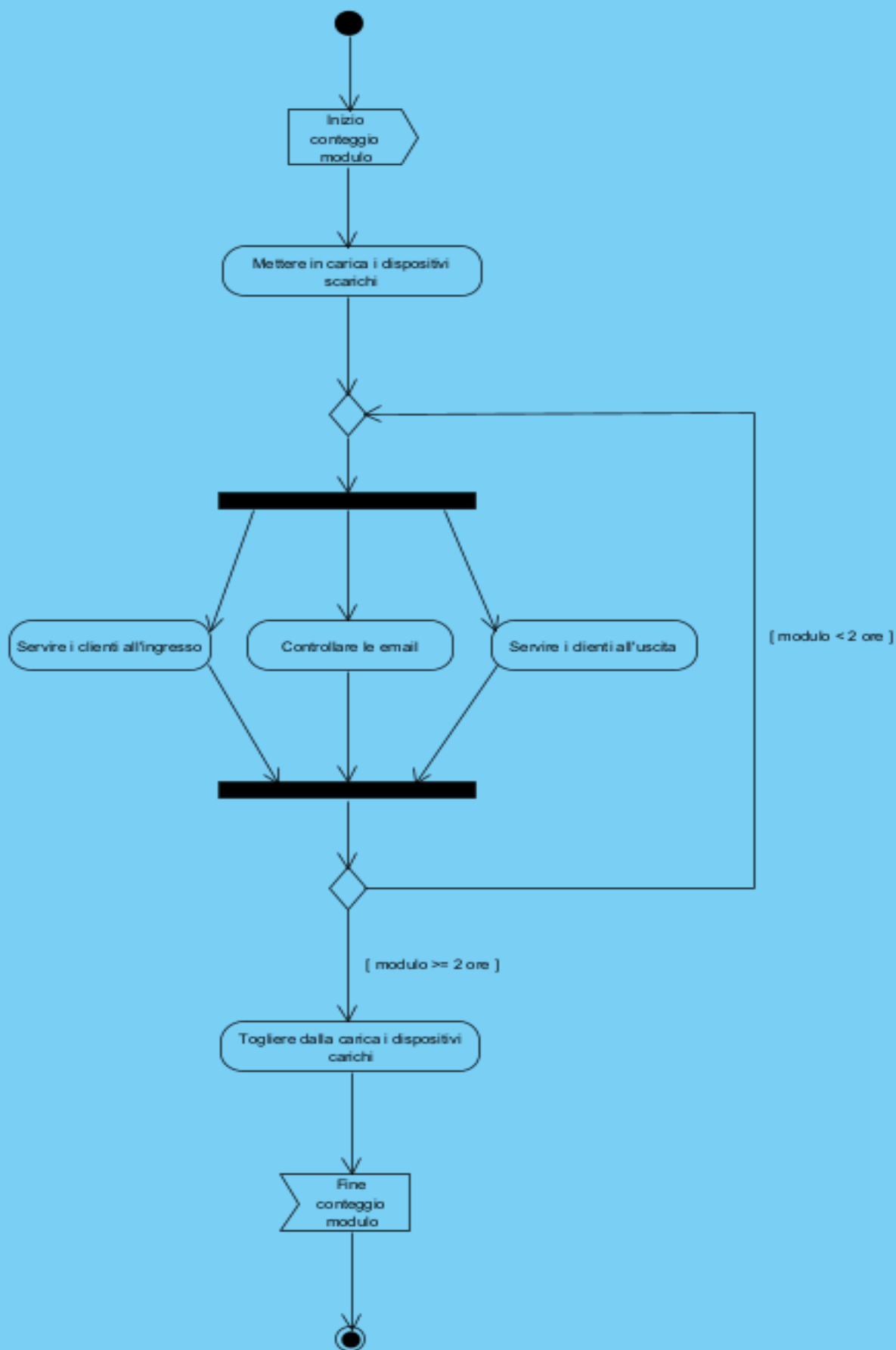
Esercizio 3



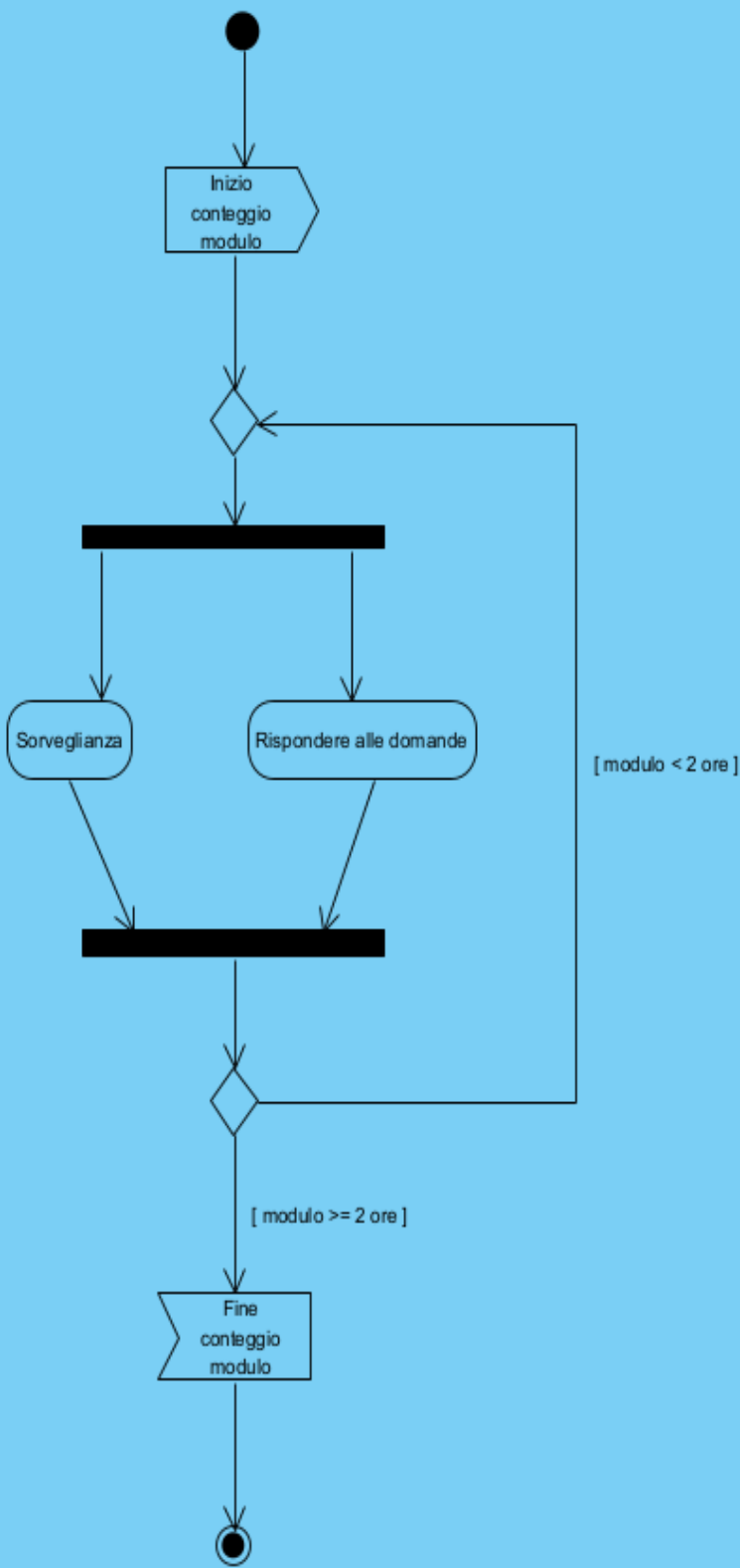
Esercizio 4



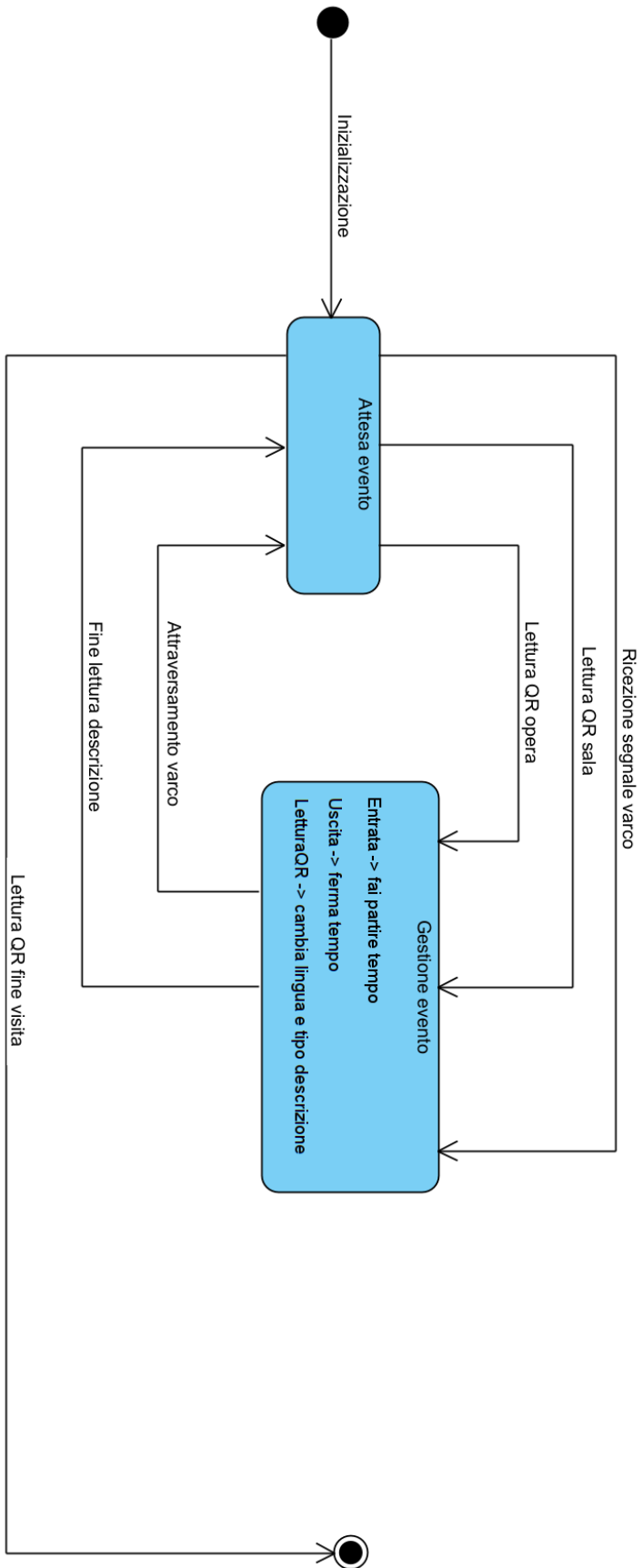
Assegnamento alla reception



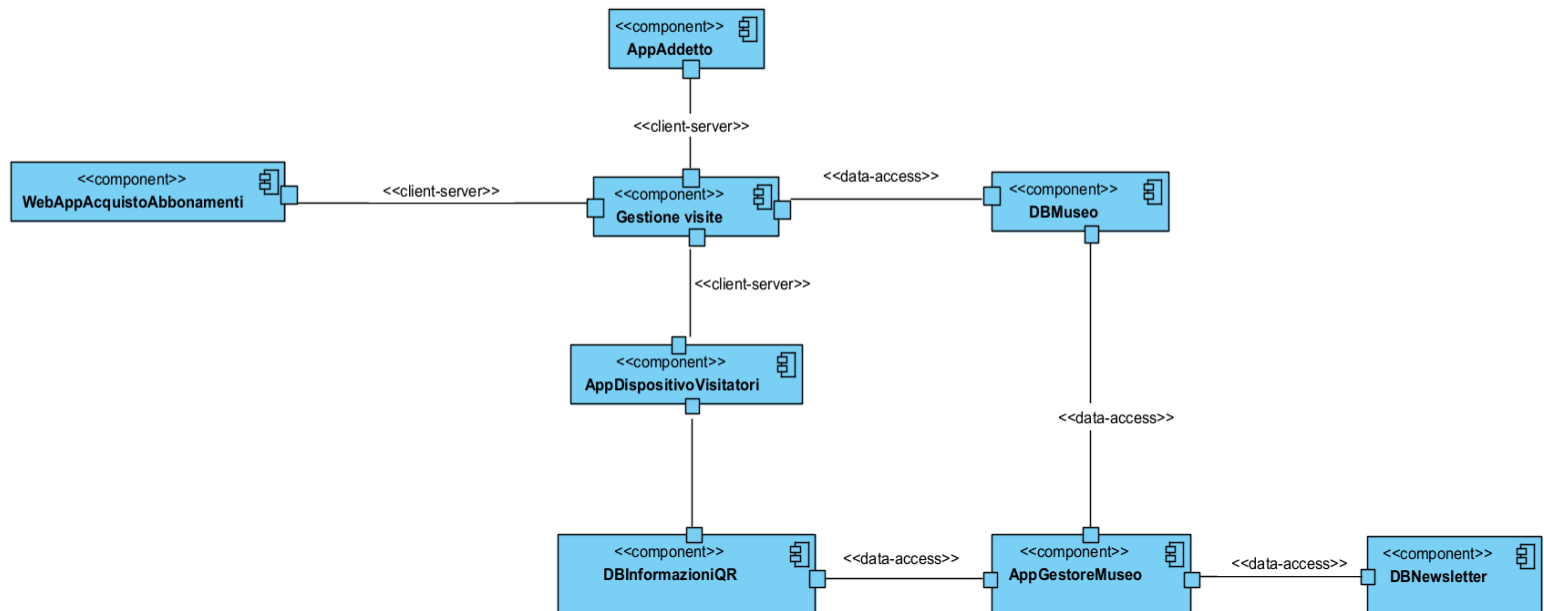
Assegnamento a una sala



Esercizio 5



Esercizio 6



Al momento della consegna del dispositivo, GestioneVisite inizializza il dispositivo sulla base della tariffa scelta (tariffa bianca, verde o abbonamento).

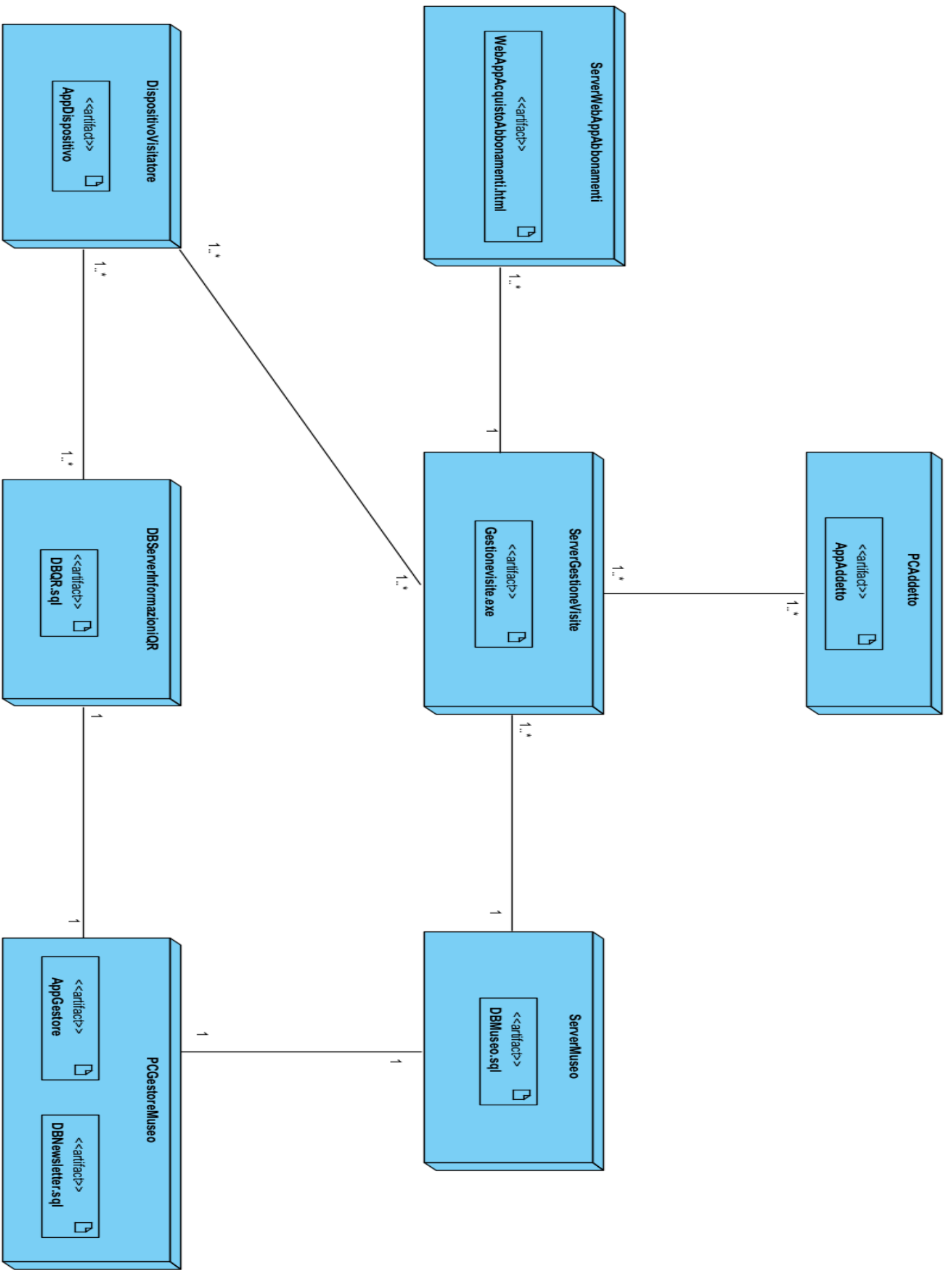
Durante la visita, il dispositivo aggiorna il suo stato interno ogni volta che il visitatore attraversa un varco, memorizzando un oggetto di tipo Passaggio.

Il dispositivo, alla fine della visita, attraverso una comunicazione client-server, invia al GestoreVisite un pacchetto contenente:

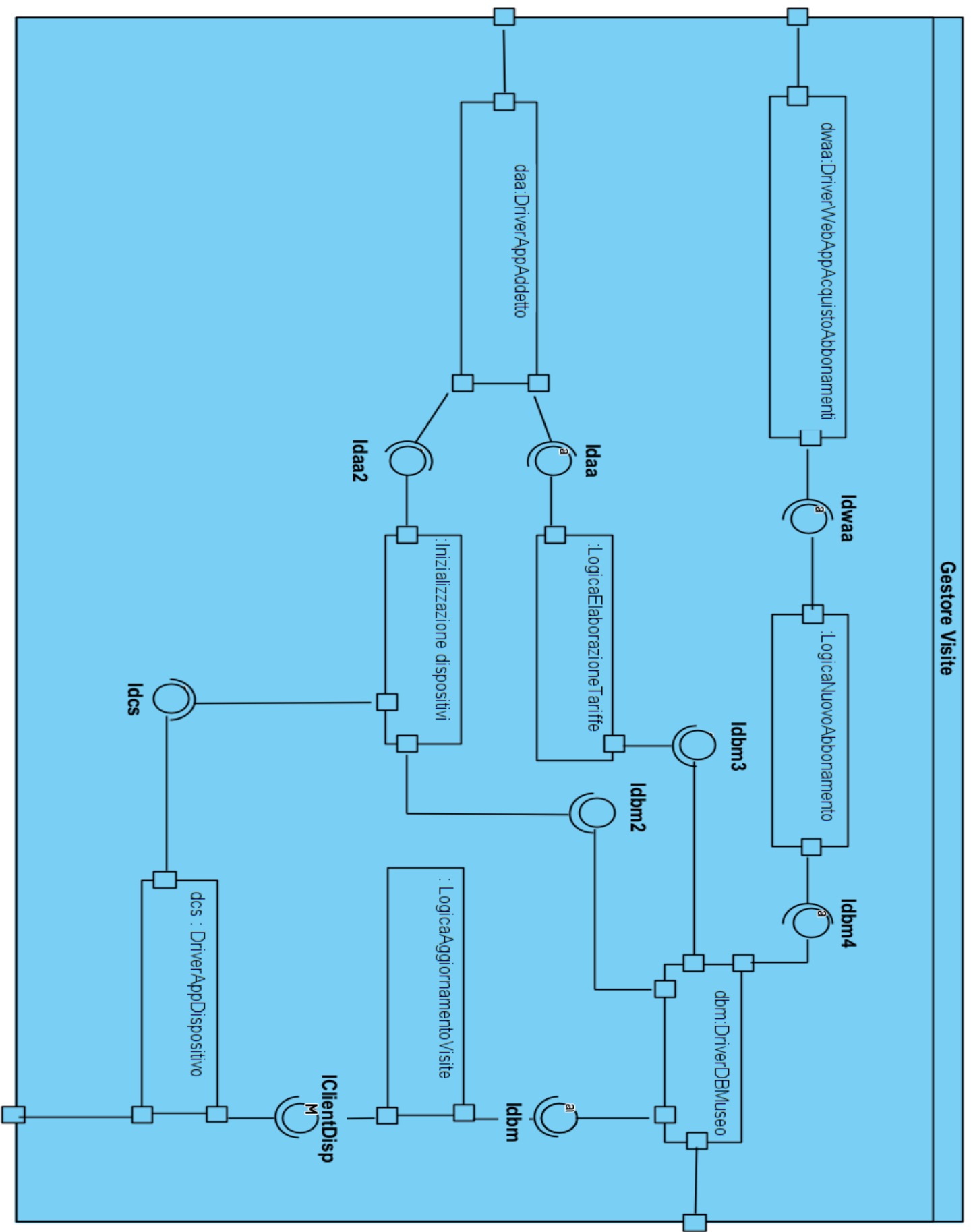
- codice biglietto/abbonamento
- array di tipo Passaggio, contenente le informazioni relative ai passaggi da un varco.

Il GestoreVisite provvederà, di conseguenza, ad aggiornare le informazioni nel database relative alla visita, oltre che a calcolare la tariffa, che sarà salvata sul db e successivamente inoltrata all'AppAddetto.

Esercizio 7



Esercizio 8



Esercizio 9

```
1  int calcolaTariffaBianca ([Passaggio] lp){
2      int tariffa = 0;
3      for (i = 1; i < lp.length; i++){
4          if (lp[i].orario - lp[i-1].orario >= 30){
5              if salaPermanente(lp[i-1].sala)
6                  tariffa += 3;
7              if salaTemporaneo(lp[i-1].sala)
8                  tariffa += 5;
9          }
10     }
11     return tariffa;
12 }
```

Esercizio 9a

Variabili casi di test

- `lp` → array di tipo `Passaggio`.
- `lp[i].orario` → orario in cui è avvenuto il passaggio dal sensore.
Vincoli: `orarioApertura <= lp[i].orario <= orarioChiusura`
- `lp[i].sala` → sala di arrivo
Vincoli: sala presente nel database
- `tariffa` → tariffa calcolata dalla funzione `calcolaTariffaBianca`

Stub

```
tipoSala (int IdA){
    if(( IdA%2 ) == 0 ){
        this.nome="Sala del Cubismo";
        this.tipo="Sale Museali"
    } else {
        this.nome="Sala Orientale";
        this.tipo="Sale Mostre"
    }
}
```

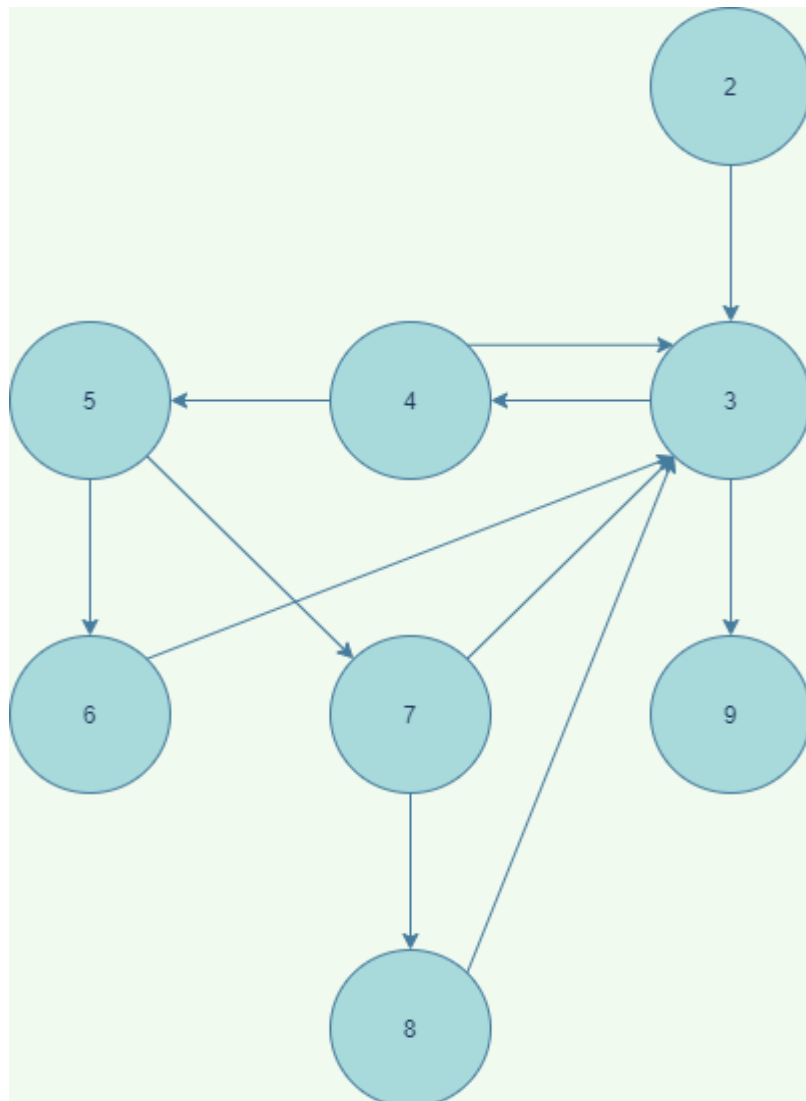

Classi di equivalenza

Intervallo di tempo	Tipo ambiente	Output atteso	Quantificazione output
$0 \leq \text{tempo} \leq 30$	Sale Museali	Tariffa Calcolata	Bool : Corretto Valore = 0
$0 \leq \text{tempo} \leq 30$	Sale Mostre	Tariffa Calcolata	Bool : Corretto Valore = 0
$0 \leq \text{tempo} \leq 30$	Ambiente di servizio	Tariffa Non Calcolata	Boll : Errato Valore : errore
$\text{tempo} > 30$	Sale Museali	Tariffa Calcolata	Bool : Corretto Valore = $3 * \text{tempo}$
$\text{tempo} > 30$	Sale Mostre	Tariffa Calcolata	Bool : Corretto Valore = $5 * \text{tempo}$
$\text{tempo} > 30$	Ambiente di servizio	Tariffa Non Calcolata	Boll : Errato Valore : errore
$\text{tempo} < 0$	Indifferente	Tariffa Non Calcolata	Boll : Errato Valore : errore

$\text{tempo} = \text{tempo ingresso} - \text{tempo uscita}$

Esercizio 9b

```
1  int calcolaTariffaBianca ([Passaggio] lp){
2      int tariffa = 0;
3      for (i = 1; i < lp.length; i++){
4          if (lp[i].orario - lp[i-1].orario >= 30){
5              if salaPermanente(lp[i-1].sala)
6                  tariffa += 3;
7              if salaTemporanea(lp[i-1].sala)
8                  tariffa += 5;
9          }
10     }
11     return tariffa;
12 }
```



Esercizio 9c

Batteria di test

```
lptest1 = {  
    lp[0].orario = 8*3600,           // Passaggio effettuato alle ore 8:00:00  
    lp[0].sala = 1,                 // Entra nella prima sala  
    lp[1].orario = 9*3600,           // Passaggio effettuato alle ore 9:00:00  
    lp[1].sala = 2                   // Entra in una sala museale  
}  
  
lptest2 = {  
    lp[0].orario = 8*3600,           // Passaggio effettuato alle ore 8:00:00  
    lp[0].sala = 1,                 // Entra nella prima sala  
    lp[1].orario = 9*3600,           // Passaggio effettuato alle ore 9:00:00  
    lp[1].sala = 3                   // Entra in una sala mostra  
}  
  
lptest3 = {  
    lp[0].orario = 8*3600,           // Passaggio effettuato alle ore 8:00:00  
    lp[0].sala = 1,                 // Entra nella prima sala  
    lp[1].orario = 8*3600 + 29,      // Passaggio effettuato alle ore 8:00:29  
    lp[1].sala = 2                   // Entra nella seconda sala  
}  
  
lptest4 = {  
    lp[0].orario = 10*3600,          // Passaggio effettuato alle ore 10:00:00  
    lp[0].sala = 1,                 // Entra nella prima sala  
    lp[1].orario = 8*3600,           // Passaggio effettuato alle ore 8:00:00  
    lp[1].sala = 2                   // Entra nella seconda sala  
}
```

Batteria di test = { < (lp_{test}1), 3 >, < (lp_{test}2), 5 >, < (lp_{test}3), 0 >, < (lp_{test}4), ERRORE > }

Esercizio 9d

Se queste due funzioni fornissero un risultato errato:

- Overflow → si verifica un overflow nel caso in cui uno dei parametri passati allo stub abbiano una dimensione maggiore di 2^{32} .
- Id negativo → il sistema è progettato per generare id aree positivi, per cui non sarebbe in grado di accedere ad id che non ha generato.