**“Software Engineering”**

**Course**

**a.a. 2018-2019**

**Template version 1.0**

**Deliverable #1**

**Lecturer: Prof. Henry Muccini (henry.muccini@univaq.it)**

**Dashboard Monitoraggio Ambientale**

|  |  |
| --- | --- |
| **Date** | 03/12/2018 |
| **Deliverable** | 1 |
| **Team (Name)** | 404 not found |

|  |  |  |
| --- | --- | --- |
| **Team Members** | | |
| **Name & Surname** | **Matriculation Number** | **E-mail address** |
| D’Ercole Dario | *244615* | [dario.dercole@student.univaq.it](mailto:dario.dercole@student.univaq.it) |
| Carestia Alessandro | *244236* | [alessandro.carestia@student.univaq.it](mailto:alessandro.carestia@student.univaq.it) |
| De Flaviis Manuel | *231100* | [manuel.deflaviis@student.univaq.it](mailto:manuel.deflaviis@student.univaq.it) |
| Pennacchia Francesco | *247848* | [francesco.pennacchia@student.univaq.it](mailto:francesco.pennacchia@student.univaq.it) |
| Avallone Andrea | *179324* | [andrea.avallone@student.univaq.it](mailto:andrea.avallone@student.univaq.it) |

**Project Guidelines**[do not remove this page]*This page provides the Guidelines to be followed when preparing the report for the Software Engineering course. You have to submit the following information:*

* *This Report*
* *Diagrams (Use Case, Component Diagrams, Sequence Diagrams, Entity Relationships Diagrams)*
* *Effort Recording (Excel file)*

***Important:***

* ***document risky/difficult/complex/highly discussed*** *requirements*
* *document decisions taken by the team*
* ***iterate****: do not spend more than 1-2 full days for each iteration*
* ***prioritize*** *requirements, scenarios, users, etc. etc.*

Project Rules and Evaluation Criteria

***General information:***

* *This homework will cover the 80% of your final grade (20% will come from the oral examination).*
* *The complete and final version of this document shall be* ***not longer than 40 pages*** *(excluding this page and the Appendix).*
* *Groups composed of five students (preferably).*

*I expect the groups to submit their work through GitHub*

***Use the same file to document the various deliverable.   
Document in this file how Deliverable “i+1" improves over Deliverable “i".***

**Project evaluation:**

*Evaluation is not based on “quantity” but on “quality” where quality means:*

* *Completeness of delivered Diagrams*
* *(Semantic and syntactic) Correctness of the delivered Diagrams*
* *Quality of the design decisions taken*
* *Quality of the produced code*

Table of Contents of this deliverable

Sommario

**Digitare il titolo del capitolo (livello 1) 1**

Digitare il titolo del capitolo (livello 2) 2

Digitare il titolo del capitolo (livello 3) 3

**Digitare il titolo del capitolo (livello 1) 4**

Digitare il titolo del capitolo (livello 2) 5

Digitare il titolo del capitolo (livello 3) 6

List of Challenging/Risky Requirements or Tasks

*<In this section, you should describe using the table below the most challenging or discussed or risky design tasks, requirements, or activities related to this project. Please describe when the risk arised, when and how it has been solved.>*

PLEASE FILL IN THIS TABLE AT EACH DELIVERABLE

|  |  |  |  |
| --- | --- | --- | --- |
| **Challenging Task** | **Date the task is identified** | **Date the challenge is resolved** | **Explanation on how the challenge has been managed** |
| Formato memorizzazione del dato; | 14/11/2018 |  |  |
| Gestione comportamento dei sensori (in caso di dati anomali); | 20/11/2018 | 30/11/2018 | Prevediamo che il Sistema potrà fornire un feedback al sensore sul suo stato così da regolarne la frequenza di invio dei dati |
| Trovare tool adeguati alla gestione delle fasi di progettazione; | 20/11/2018 |  | Project Libre è scomodo, troveremo un altro tool per il secondo deliverable |
| visualizzazione del giusto livello di dettaglio da parte di un gestore | 30/11/2018 | 03/12/2018 | È giusto che un gestore veda i dati di tutto il sistema e quindi di ogni singolo sensore? Ha senso che veda solo se nel suo livello inferiore vi sono anomalie |
| elaborazione del dato in base al livello di gestione | 30/11/2018 | 03/12/2018 | Il dato lo dobbiamo aggregare per poterlo vedere anche nei livelli superiori oppure abbiamo bisogno di un semplice alert di eventuali anomalie? |
| Rischio tutti i sensori aumentano la loro frequenza di invio | 30/11/2018 |  |  |

A. Requirements Collection

*In this section, you should describe both the application* ***features/functional*** *requirements as well as the* ***non functional*** *ones. You shall also document* ***constraints*** *and* **rules***, if they apply.*

A.1 Functional Requirements

**REQUISITI FUNZIONALI:**

* Cattura del Segnale;
* Interpretazione dei segnali;
* Memorizzazione dei dati;
* Caricamento/Aggiornamento dati in dashboard;
* Modifica del livello di dettaglio;
* Evidenziazione anomalie sensore;
* Evidenziazione dati anomali;

*<Describe functional requirements and stakeholders through Use Case Diagrams>. <Then, prioritize them, and provide a table-based description of the most important requirements>*

*PLEASE COPY HERE: (i) ALWAYS the diagram to be discussed, (ii) the text explaining the DECISIONS taken when creating the diagram (that is, do not spend time in EXPLAINING the details written in the diagram, but provide a small synthesis and focus on the decisions taken to create the diagram).*

PRIORITIZE THEM

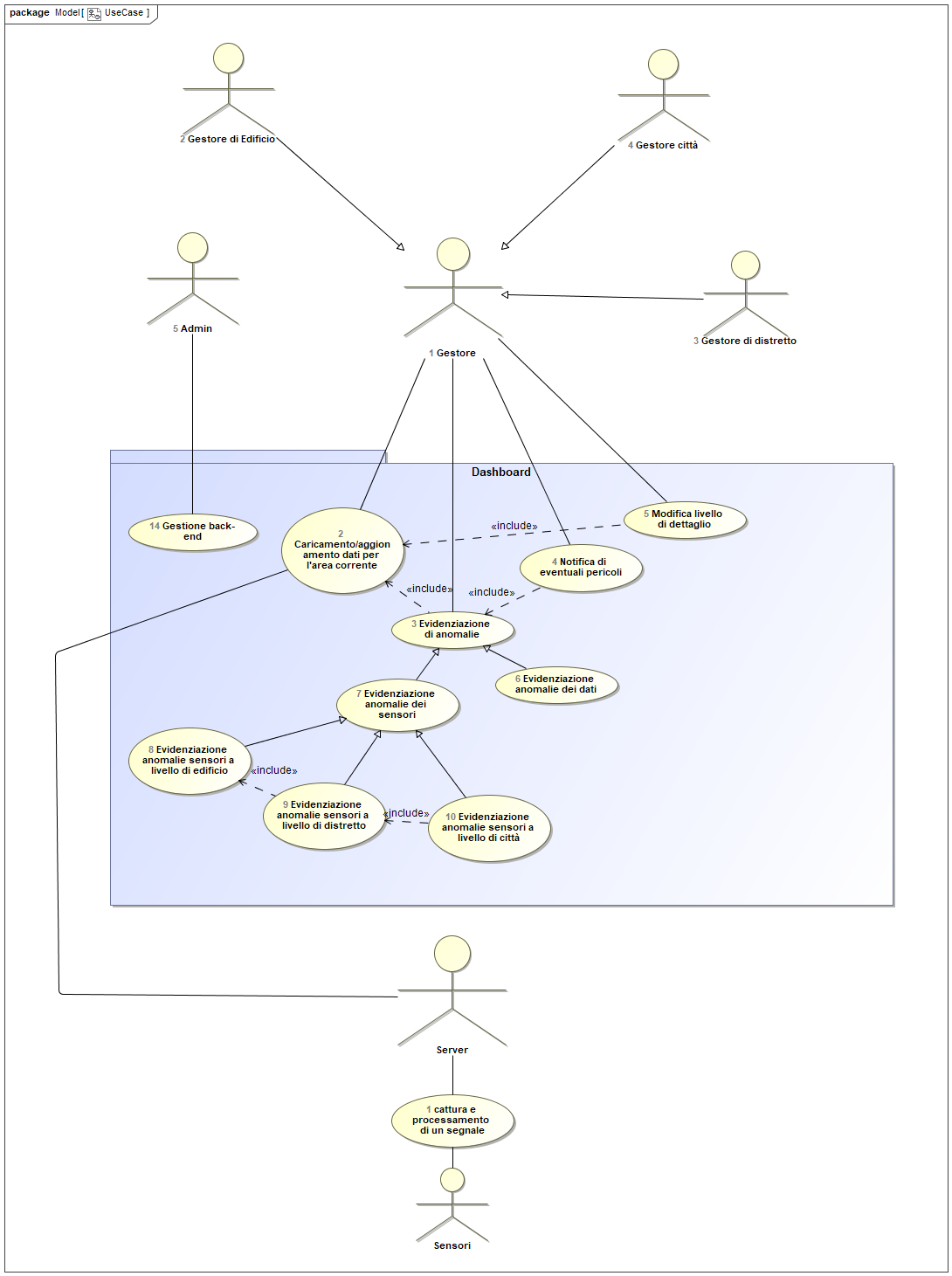
FOR EACH FIGURE (Use Case D, Class, Sequence, etc) add a number and a label to it (e.g., Figure 1: Sequence diagram of the xxx scenario)

. Please do not submit the model alone: add a textual description for each model!!

A1.1 Use Case Diagrams

Alcune note sul diagramma:

1. Questa prima versione comprende esclusivamente requisiti ad alto livello, i requisiti di sistema e i corrispondenti scenari/casi d’uso saranno introdotti nel secondo deliverable, ciò include le funzionalità lato server, qui riportato come semplice attore. (Motivazione: Ci sono stati grandi ritardi nella prima fase)
2. La gestione in back-end sarà gestita nel prossimo deliverable, quando daremo anche un modello per i dati.
3. Ho preferito separare nei 3 livelli di astrazione (edificio, distretto, città) il caso d’uso per le anomalie dei sensori dal momento che i dati (dati singoli vs dati aggregati) e i processi coinvolti (threeshold basato su sensori di backup rimanenti vs threeshold basato su rapporto sensori funzionanti / sensori totali) sono troppo eterogenei per riassumerli sotto un unico use-case. (Dario D’Ercole)
4. Le relazioni di inclusione tra i casi d’uso 10 -> 9 , 9 -> 8 saranno probabilmente eliminate in quanto i casi d’uso 9 e 10 utilizzeranno dati che sono stati prodotti dai casi d’uso 8 e 9 ma lo faranno in modo asincrono: i trigger per questi casi d’uso sono la ricezione di nuovi dati rispetto all’area monitorata, non la disponibilità di nuovi dati relativi alle anomalie sui sensori.
5. Le relazioni di inclusione tra casi d’uso 3 -> 2, 4 -> 3 indicano un trigger: L’aggiornamento dei dati in dashboard (#2) fanno scattare il controllo sulle anomalie (#3), mentre al termine del controllo sulle anomale (#3), se ne sono state rilevate, verrà effettuato il controllo sui possibili pericoli (#4); La relazione di inclusione tra casi d’uso 5 -> 2 è invece in senso stretto, dal momento che per focalizzarsi su una sotto-area, la dashboard farà richiesta al server dei dati relativi a quella sotto-area come 1-step routine.



**Figura 1 –** UseCase Diagram V1

A1.2 Tabular description of the most relevant use cases

|  |  |  |
| --- | --- | --- |
| USE CASE # | #1: Cattura e processamento di un segnale | |
| Goal in Context | Salvare nel database I dati portati da un segnale in arrivo | |
| Scope & Level | Sistema di persistenza - Primary Task | |
| Success End Condition | I dati ricevuti con un segnale sono stati correttamente memorizzati sul database | |
| Primary,  Secondary Actors | Sensore,  Server | |
| Trigger | Ricezione di un segnale mandato da un sensore | |
| DESCRIPTION | Step | Action |
|  | 1 | Il Sistema riceve un segnale da un sensore |
|  | 2 | Il sistema parsa il contenuto del segnale per estrarne i dati |
|  | 3 | Il Sistema costruisce una query per il database con i dati ricevuti |
|  | 4 | Viene eseguita la query al database |

|  |  |
| --- | --- |
| RELATED INFORMATION |  |
| Priority: | Highest priority |
| Frequency | Almeno 150.000 / minuto |
| OPEN ISSUES | Formato dei dati |

**Figura 2 –** Descrizione tabellare per lo UseCase #1

|  |  |  |
| --- | --- | --- |
| USE CASE #2 | Caricamento/aggionamento dati per l'area corrente. | |
| Goal in Context | Caricare i dati dei sensori dal server e aggiornare quelli già presenti nella dashboard. | |
| Scope & Level | Interfaccia Gestore - Primary task. | |
| Preconditions | Il dato nel server relativo ad un sensore deve essere successivo (nel tempo) a quello nella dashboard. | |
| Success End Condition | Il dato nel server relativo al sensore è succesivo (nel tempo) a quello nella dashboard, e viene aggiornato. | |
| Failed End Condition | Mancato aggiornamento dei dati. | |
| Primary,  Secondary Actors | Server.  Sensore. | |
| Trigger | Richiesta dei dati al server. | |
| DESCRIPTION | Step | Action |
|  | 1 | Vengono richiesti i dati dei sensori al server. |
|  | 2 | Per ogni sensore, viene comparato il dato nel server con quello nella dashboard. |
|  | 3 | Se il dato nel server è più recente, va a rimpiazzare quello della dashboard. |
| EXTENSIONS | Step | Branching Action |
|  | 3a | Se il dato nel server non è più recente:  Non si aggiorna il dato nella dashboard e richiama il rilevatore di anomalie per controllare se è il caso di segnalare un problema (UseCase #8) |

|  |  |
| --- | --- |
| RELATED INFORMATION |  |
| Priority: | Alta. |
| Frequency | Al più 50 volte al minuto. |
| Channels to actors | Sistema |
| Superordinates | Evidenziazione di anomalie. |

**Figura 3 –** Descrizione tabellare UseCase #2

|  |  |  |
| --- | --- | --- |
| USE CASE #4 | #4: Notifica di pericoli | |
| Goal in Context | Dare un’allarme al gestore in caso dei parametri fuori soglia sono indicatori di un certo pericolo | |
| Scope & Level | Modulo di calcolo dei pericoli (Primary Task) | |
| Preconditions | Il gestore sta visualizzando i dati di una qualsiasi area | |
| Success End Condition | Il gestore viene notificato di un possibile pericolo | |
| Failed End Condition | Il pericolo non viene notificato oppure viene notificato un allarme errato | |
| Primary,  Secondary Actors | Server  Gestore | |
| Trigger | Vengono riscontrate anomalie nell’area attualmente monitorata (UseCase 6) | |
| DESCRIPTION | Step | Action |
|  | 1 | Il sistema controlla dove sono localizzate e quali siano le anomalie riscontrate |
|  | 2 | Il sistema calcola quale sia lo stato di pericolo per ogni sottoarea anomala |
|  | 3 | Ogni sottoarea con uno stato di pericolo consistente visualizzerà un messaggio in cui viene spiegato quale sia il possibile pericolo |

|  |  |
| --- | --- |
| RELATED INFORMATION | #4: Notifica di pericoli |
| Priority: | Alta |
| Frequency | 50 / minuto |

**Figura 4 –** Descrizione tabellare UseCase #4

|  |  |  |
| --- | --- | --- |
| USE CASE # | #5 Modifica livello di dettaglio | |
| Goal in Context | Possibilità di accedere ad un livello più dettagliato di informazioni. In base alla tipologia di gestore. | |
| Scope & Level | Modulo di accesso dati. | |
| Preconditions | Il gestore sta visualizzando la sua Dashboard. | |
| Success End Condition | Vengono visualizzate le informazioni relative all’area selezionata. | |
| Failed End Condition | Viene visualizzato un messaggio di errore e il gestore continua a visualizzare i dati dell’area corrente. | |
| Primary,  Secondary Actors | Gestore | |
| Trigger | Aggiornamento dei dati. | |
| DESCRIPTION | Step | Action |
|  | 1 | Il gestore seleziona l’area di interesse (Città, Distretto, Edificio). |
|  | 2 | Il sistema recupera i dati relativi alle sotto-aree dell’area selezionata. |
|  | 3 | Il sistema visualizza i dati sulla Dashboard. |

|  |  |
| --- | --- |
| RELATED INFORMATION |  |
| Priority: | Medio |
| Frequency | Minimo 150000/Minuto |
| Channels to actors | Interattivo |

**Figura 5 –** Descrizione tabellare UseCase #5

|  |  |  |
| --- | --- | --- |
| USE CASE # | #6: Evidenziazione Anomalie Dati | |
| Goal in Context | Al gestore viene notificata, in base al livello di competenza, un’eventuale anomalia sui dati. | |
| Scope & Level | Rilevatore anomalie – Primary Task | |
| Preconditions | Il gestore sta visualizzando la sua dashboard. | |
| Success End Condition | Nella Dashboard vengono riportati tutti i dati e le relative anomalie | |
| Failed End Condition | Uno o più dati potrebbero non essere visualizzati oppure un/più anomalia/e non viene/vengono segnalata/e | |
| Primary,  Secondary Actors | Server  Gestore | |
| Trigger | Aggiornamento dei dati | |
| DESCRIPTION | Step | Action |
|  | 1 | Il Sistema controlla quanto i dati appena ricevuti si discostano dalla soglia massima. |
|  | 2 | Nel caso di dati da singoli sensori: evidenzia i dati normali in verde, gli anomali in arancio |
| EXTENSIONS | Step | Branching Action |
|  | 1a | Nel caso di dati aggregati, relative ad un’area più estesa (cioè un intero edificio o un distretto) :  Evidenzia in verde I dati normali, in arancio i dati delle aree che registrano un solo tipo di anomalia, in rosso i dati delle aree che registrano più tipi di anomalia. |

|  |  |
| --- | --- |
| RELATED INFORMATION |  |
| Priority: | Alta |
| Frequency | Almeno 50/minute |

**Figura 6 –** Descrizione tabellare UseCase #6

|  |  |  |
| --- | --- | --- |
| USE CASE # | #8: Evidenziazione anomalie nei sensori a livello di edificio | |
| Goal in Context | Evidenziare il/i dato/i proveniente/i da un sensore che presenta problemi | |
| Scope & Level | Rivelatore Anomalie – Primary task | |
| Preconditions | Il gestore sta visualizzando i dati relativi ad una certa area | |
| Success End Condition | I dati relativi a sensori che riportano problemi vengono evidenziati in modo distinto. | |
| Failed End Condition | I dati relativi a sensori che riportano problemi non saranno evidenziati fino al prossimo avvio di questo caso d’uso | |
| Primary,  Secondary Actors | Server  Gestore di edificio, Gestore di distretto | |
| Trigger | Aggiornamento dei dati sulla dashboard | |
| DESCRIPTION | Step | Action |
|  | 1 | Il Sistema recupera i dati relativi al funzionamento dei sensori presenti nella zona attualmente monitorata e il timestamp del loro ultimo segnale inviato |
|  | 2 | Il sistema controlla quanti sensori per ogni tipo di dato monitorato sono ancora in funzione nell’area |
|  | 3 | Evidenzia con bassa priorità se c’è almeno un sensore di backup, altrimenti evidenzia con alta priorità |

|  |  |
| --- | --- |
| RELATED INFORMATION |  |
| Priority: | Media priorità |
| Frequency | Almeno 150.000 / minuto |

**Figura 7 –** Descrizione tabellare UseCase #8

|  |  |  |
| --- | --- | --- |
| USE CASE # | #9: Evidenziazione anomalie nei sensori a livello di distretto | |
| Goal in Context | Evidenziare il/i dato/i proveniente/i da un gruppo di sensori in cui la maggior parte presenta problemi | |
| Scope & Level | Rilevatore anomalie – Primary Task | |
| Preconditions | Il gestore sta visualizzando i dati relativi ad un certo distretto | |
| Success End Condition | I dati relativi ad un gruppo di sensori in cui la maggior parte riporta problemi vengono evidenziati in modo distinto. | |
| Failed End Condition | I dati relativi ad un gruppo di sensori in cui la maggior parte riporta problemi non saranno evidenziati fino al prossimo avvio di questo caso d’uso | |
| Primary,  Secondary Actors | Server  Gestore di distretto, Gestore di città | |
| Trigger | Primo caricamento o aggiornamento automatico dei dati sulla dashboard | |
| DESCRIPTION | Step | Action |
|  | 1 | Il Sistema controlla il numero totale di sensori per ogni edificio disponibili per ogni parametro monitorato. |
|  | 2 | il sistema verifica per ogni edificio, e per ogni parametro monitorato, il relativo numero di sensori che presentano anomalie (UseCase #8) |
|  | 3 | In base al rapporto sensori con anomalie / sensori totali, l’informazione verrà riportata sulla dashboard con colore più o meno marcato. |

|  |  |
| --- | --- |
| RELATED INFORMATION |  |
| Priority: | Alta priorità |
| Frequency | Almeno 50 / minuto |

**Figura 8 –** Descrizione tabellare UseCase #9

|  |  |  |
| --- | --- | --- |
| USE CASE # | #10: Evidenziazione anomalie nei sensori a livello di città | |
| Goal in Context | Evidenziare il/i dato/i proveniente/i da un gruppo di sensori in cui la maggior parte presenta problemi | |
| Scope & Level | Rivelatore anomalie – Primary task | |
| Preconditions | Il gestore sta visualizzando i dati relativi ad una certa città | |
| Success End Condition | I dati relativi ad un gruppo di sensori in cui la maggior parte riporta problemi vengono evidenziati in modo distinto. | |
| Failed End Condition | I dati relativi ad un gruppo di sensori in cui la maggior parte riporta problemi non saranno evidenziati fino al prossimo avvio di questo caso d’uso | |
| Primary,  Secondary Actors | Server  Gestore di città | |
| Trigger | Primo caricamento o aggiornamento automatico dei dati sulla dashboard | |
| DESCRIPTION | Step | Action |
|  | 1 | Il Sistema controlla il numero totale di sensori per ogni distretto disponibili per ogni parametro monitorato. |
|  | 2 | il sistema verifica per ogni distretto, e per ogni parametro monitorato, il relativo numero di sensori che presentano anomalie (UseCase #9) |
|  | 3 | In base al rapporto sensori con anomalie / sensori totali, l’informazione verrà riportata sulla dashboard con colore più o meno marcato. |

|  |  |
| --- | --- |
| RELATED INFORMATION |  |
| Priority: | Alta priorità |
| Frequency | Almeno 50 / minuto |

**Figura 9 –** Descrizione tabellare UseCase #10

A.2 Non Functional Requirements

*<List and describe here the* ***most important*** *non functional requirements.>*

**-Performance:**

* il sistema deve supportare lo storage e il processing di almeno 150.000 segnali al minuto;
* il sistema deve riuscire a gestire l'accesso in parallelo di 50 gestori;

**-Scalabilità**: il sistema deve supportare future espansioni;

**-Affidabilità**: Il sistema deve essere disponibile H24 365/anno;

**-Usabilità:** I gestori devono riuscire di usare il sistema senza nessuna formazione specifica;

Note:

1. Il requisito di Affidabilità lo abbiamo assunto dal momento che trattandosi di un sistema di monitoraggio ambientale che deve notificare eventuali pericoli, dovrà essere minimizzato il tempo offline.
2. Nel caso peggiore, il requisito di performance sui segnali da processare può raggiungere un picco di 300.000 segnali al minuto (infatti supponiamo la frequenza con cui i sensori inviano dati raddoppi in caso di anomalia)

*BE CAREFUL NOT TO MAKE CONFUSION AMONG DIFFERENT NON FUNCTIONAL REQUIREMENTS. PLEASE FOCUS ON NO MORE THAN THE 3-4 MOST IMPORTANT NON FUNCTIONAL REQUIREMENTS- PLEASE JUSTIFY THEM!!!*

A.3 Excluded Requirements  
*<List and describe here the* ***most important******excluded*** *functional requirements.>*

* **Installazione dei sensori:** se ne occupa un’altra organizzazione, noi dobbiamo solamente assicurarci che i sensori di cui l’ID non sia ancora registrato in DB vengano aggiunti automaticamente alla ricezione del primo segnale a loro corrispondente.
* **Sostituzione / riparazione di sensori guasti o malfunzionanti:** ci è richiesto soltanto di notificare le anomalie sulle dashboard dei gestori.

A.4 Assumptions

*<Briefly document, in this section, the most relevant requirement assumptions/decisions you had to made during your project>*

* I sensori avvertiranno di essere guasti per qualsiasi tipo di problema che si presenti, anche se i dati continuano ad essere veritieri
* I 3 tipi di gestore saranno distinti al momento del login sul sistema
* Chi installa i sensori provvederà ad assegnarli gli ID strutturati nel modo seguente: <ID città><ID distretto><ID edificio><ID interno>
* I sensori non omnidirezionali (come ad esempio i sensori di luminosità) andranno installati tenendo conto degli standard previsti rispetto alla zona in cui appartengono (ad esempio i sensori di luminosità per ambienti esterni devono essere puntati in modo che la luce solare non causi “false” anomalie.)

***A.5 Prioritization***

<List here all the requirements, in prioritized order>

1. **Rilevazione anomalie**: ci sono ancora molti punti da chiarire sulle responsabilità dei sottosistemi e sull’aggregazione dei dati provenienti dai livelli di astrazione inferiori, specialmente per le anomalie dei sensori.
2. **Performance**: nel gruppo c’è tendenza ad ottimizzare, per questo lo piazzo al primo posto tra i requisiti non funzionali.
3. **Scalabilità:** è forse uno dei requisiti non funzionali più complicati per noi dal momento che richiede da parte nostra uno sforzo aggiuntivo per lo studio di tecnologie che rendano il sistema scalabile (come ad esempio l’utilizzo di un database non relazionale, fin’ora mai trattati da nessun membro del gruppo).
4. **Cattura e processamento di un segnale:** c’è una questione aperta su come i sensori possano sapere di essere in uno stato anomalo o meno così da aggiustare la loro frequenza di invio dei segnali, si è valutato di introdurre un feedback dal sistema nello UseCase relativo a questo requisito.
5. **Tutti gli altri requisiti:** stileremo una lista aggiornata al prossimo deliverable, che includerà anche gli eventuali requisiti di sistema

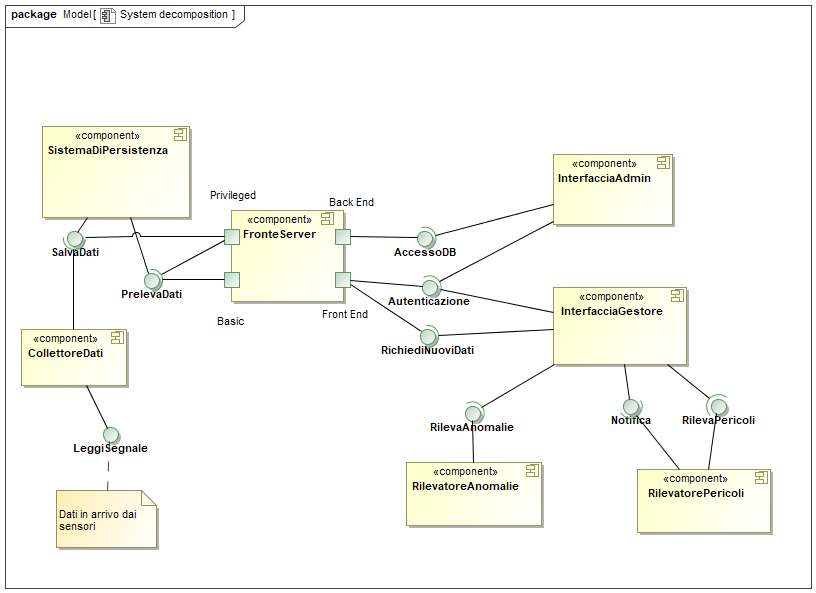
B. Software Architecture   
*<Report here both the static and the dynamic view of your system design, in terms of a Component Diagram, Class Diagrams and their related Sequence Diagrams >*

C.1The static view of the system: Component Diagram

*AVOID TO MAKE IT TOO COMPLEX AND FINE GRAINED. FOCUS MORE ON “HOW”*

*THE COMPONENTS SHALL INTERACT IN ORDER TO SATISFY THE REQUIREMENTS*

*ADD INTERFACES and their parameters*



**Figura 10 –** Component Diagram rappresentante la decomposizione del nostro sistema.

Note:

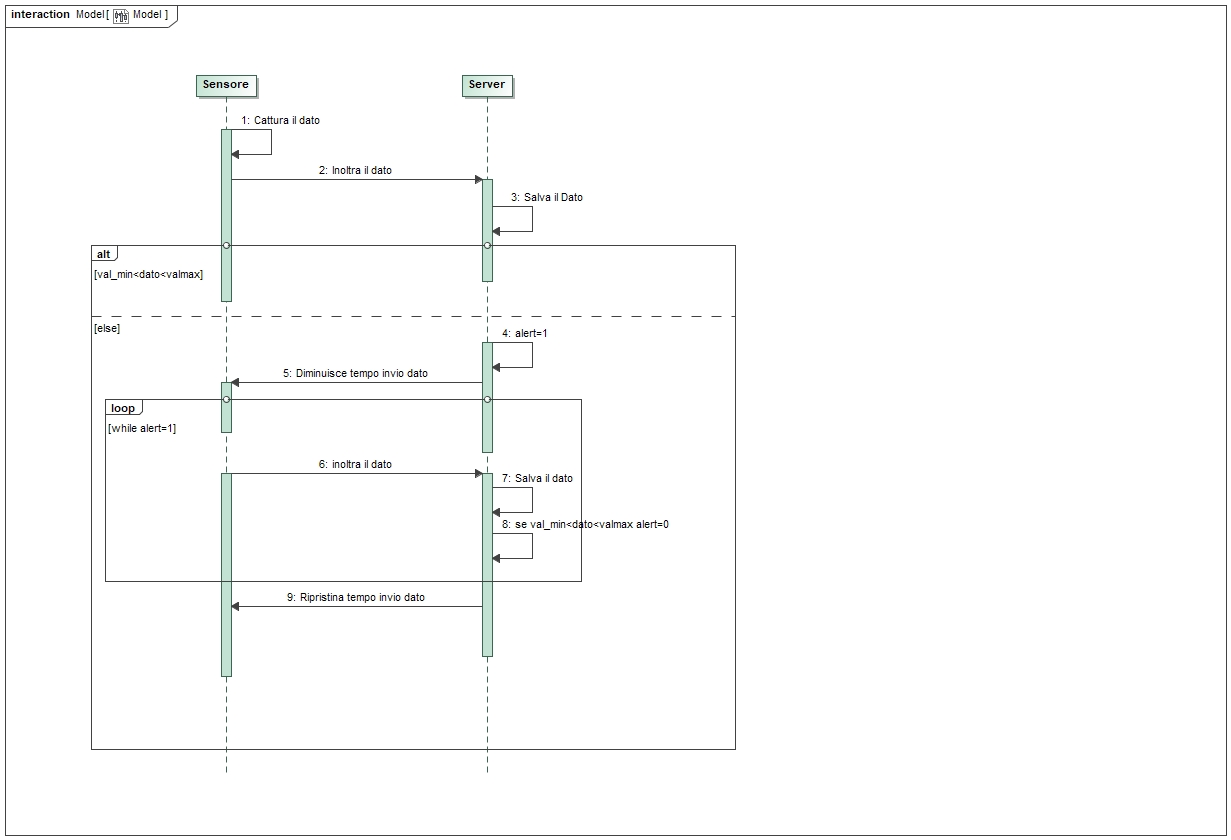
- L’autenticazione non è stata specificata come caso d’uso, ma potrebbe apparire nel secondo deliverable in quanto ci sono alcuni file che vorremo passare lato client contestualmente all’autenticazione (ad esempio un file per le soglie dei parametri monitorati).

- Al lato server verrà aggiunta un’interfaccia per fornire feedback ai sensori sul loro stato di anomalia così da regolarne la frequenza di invio dei loro segnali.

C.2 The dynamic view of the software architecture: Sequence Diagram

*BE SURE THAT THE STRUCTURE IS SYNCHRONIZED WITH THE DYNAMIC VIEW*

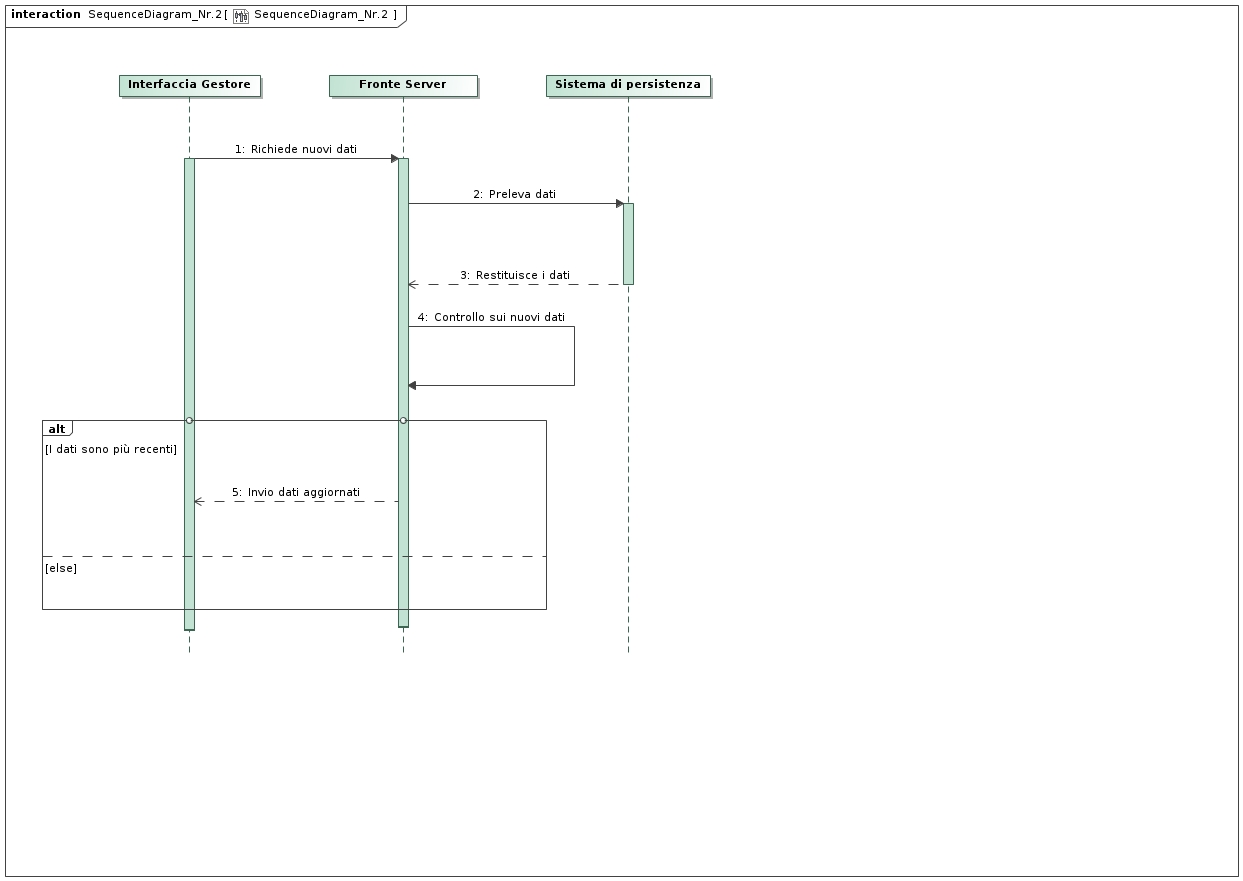
* **Cattura e processamento di un segnale:**



**Figura 11 –** Sequence Diagram per lo UseCase #1

Nota: Si presuppone che abbiamo un modo di mandare un feedback al sensore che sta inviando il segnale, così da notificarlo sul suo stato di anomalia

* **Caricamento / Aggiornamento dei dati in dashboard:**

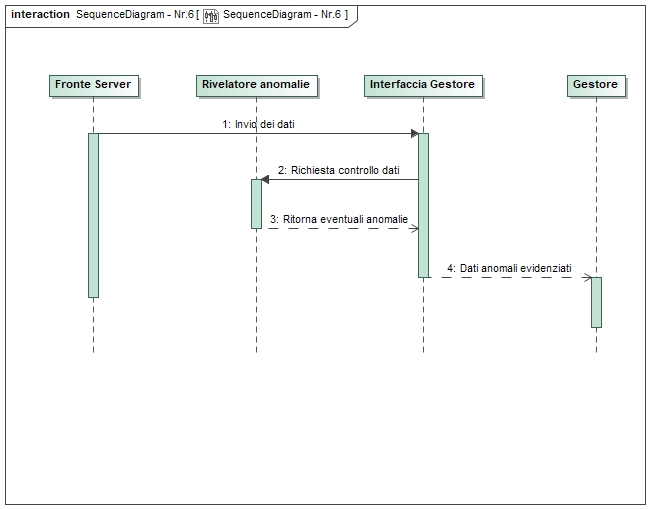


**Figura 12 –** Sequence Diagram per lo UseCase #2

Note:

- in questo diagramma si presuppone che il controllo sul timestamp viene effettuato lato server, prima di ritornare il dato al client.

- **Rilevazione anomalie sui dati:**

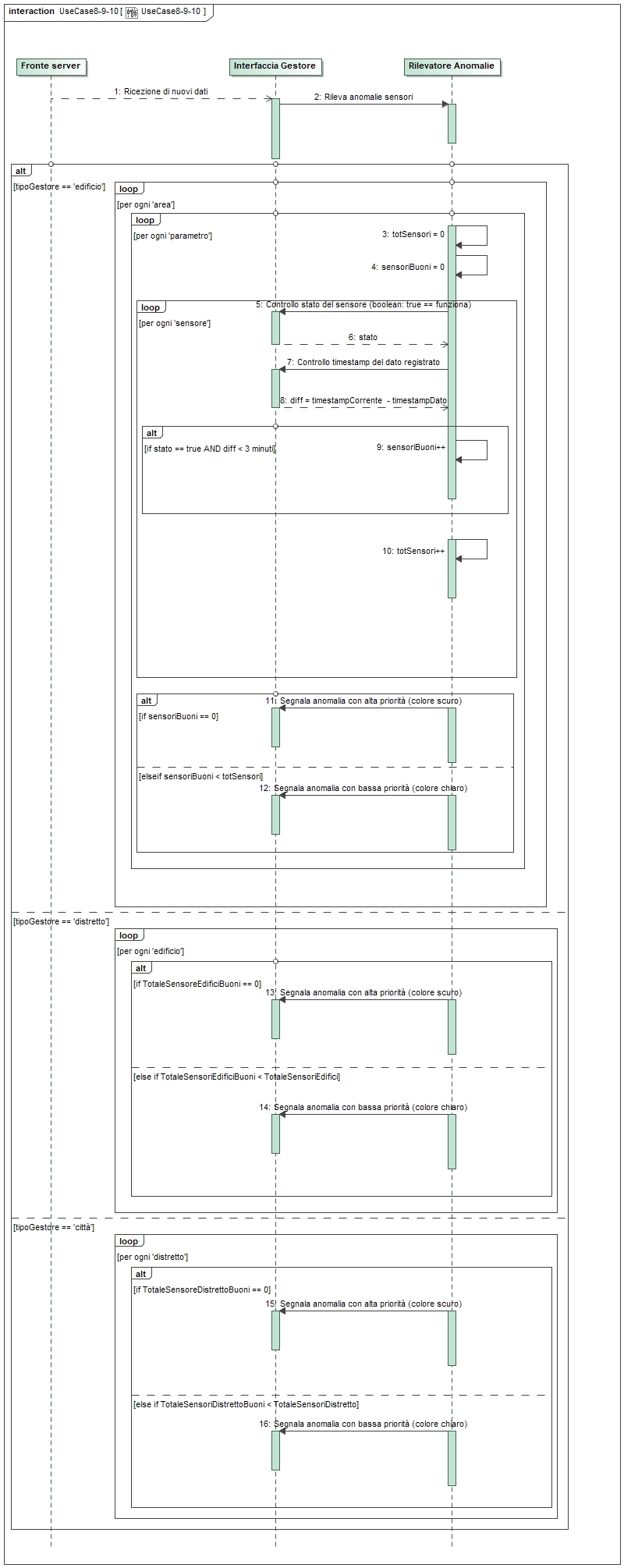


**Figura 13 –** Sequence Diagram per lo UseCase #6

Note:

- a differenza dello UseCase per il rilevamento delle anomalie nei sensori, qui non abbiamo distinto nei 3 livelli di astrazione (edificio, distretto, città), poiché le differenze sono simili a quelli riscontrabili tra gli UseCase 8, 9 e 10.

- **Rilevazione delle anomalie nei sensori:**



**Figura 14 –** Sequence Diagram per gli UseCases #8, #9 e #10

Note:

- gli scenari inclusi qui si distinguono in base al tipo di gestore che sta visualizzando la dashboard (flusso alternativo più esterno):

1) [tipoGestore == ‘edificio’] indica il flusso per lo UseCase #8

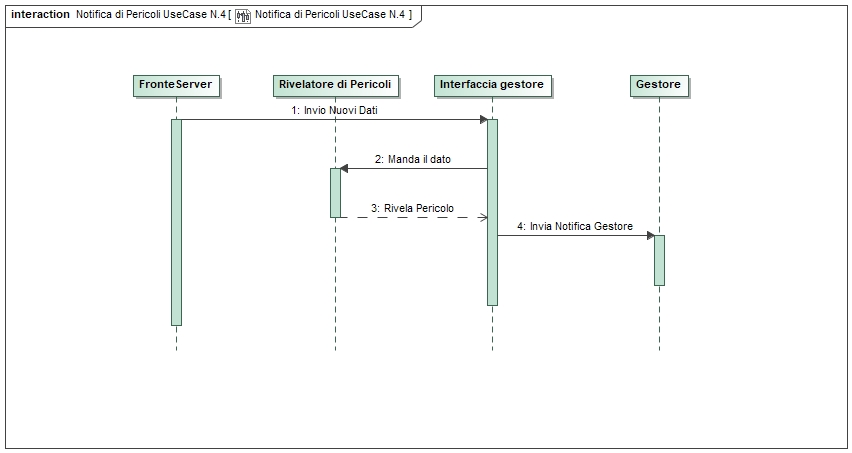
2) [tipoGestore == ‘distretto’] indica il flusso per lo UseCase #9

3) [tipoGestore == ‘città’] indica il flusso per lo UseCase #10.

- gli scenari relativi alle anomalie a livello di edificio sono stati riportati in modo più dettagliato per dare un’idea del procedimento che avviene ai livelli superiori, anche se in quel caso si opererà su dati aggregati anziché sui singoli dati.

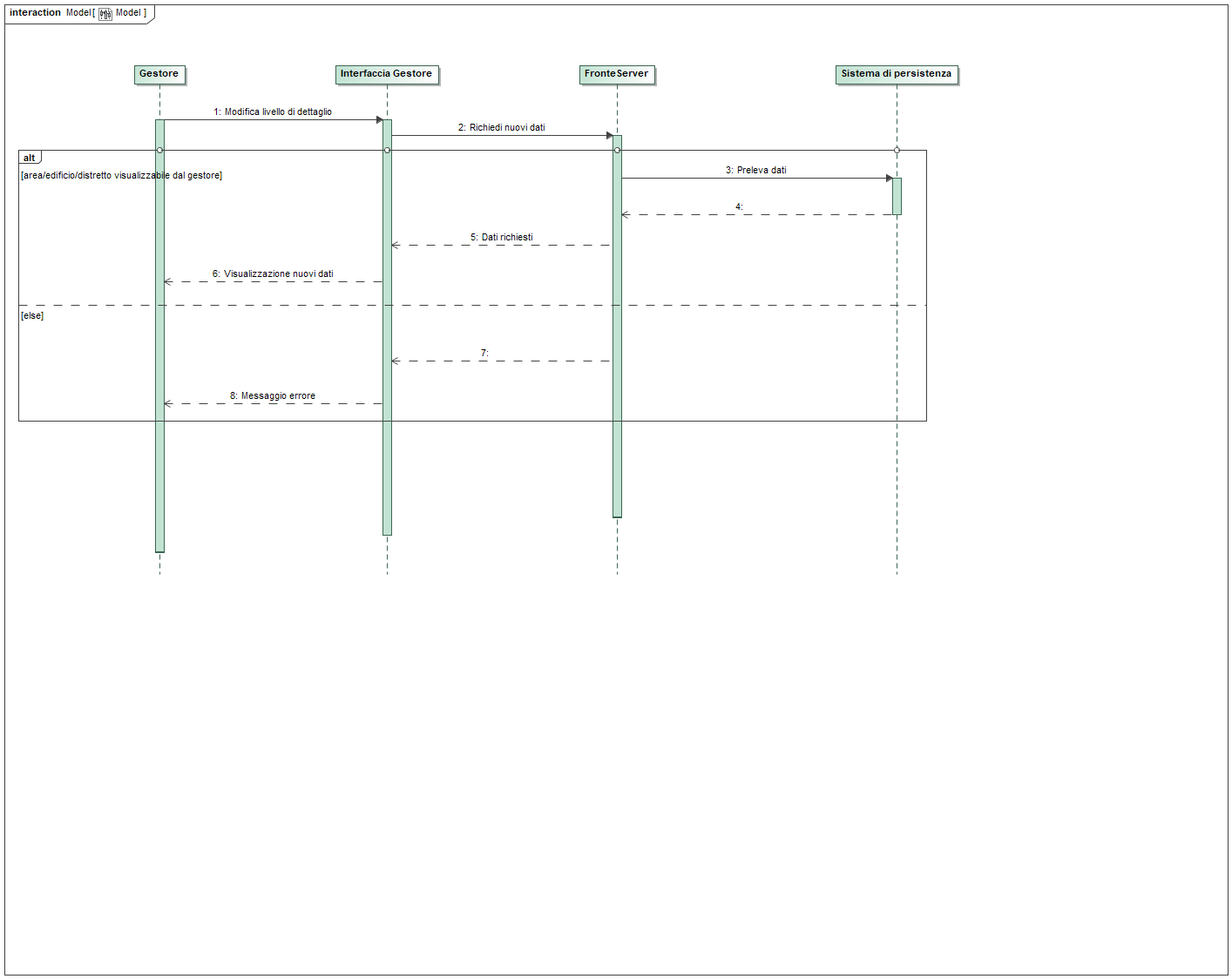
- Le guardie poste nei flussi alternativi per gli scenari a livello di distretto e città vanno modificati in quanto in quei casi abbiamo intenzione di riportare le informazioni sulle anomalie dei sensori basandoci su delle percentuali, mentre qui si considerano i casi “se nessun sensore sta funzionando” e “se almeno un sensore non funziona”; l’ideale sarebbero due guardie del tipo “se meno del x% dei sensori funziona” e “se meno del y% dei sensori funziona”, dove x < y.

- **Notifica di pericolo:**



**Figura 15 –** Sequence Diagram UseCase #4

- **Modifica livello di dettaglio:**



**Figura 16 –** Sequence Diagram UseCase #5

E. Design Decisions   
<Document here the **5** most important design decisions you had to take. You can use both a textual or a diagrammatic specification.>

THIS IS A VERY IMPORTANT PART. BE SURE TO DOCUMENT THE 5 MOST IMPORTANT DECISIONS (related to your requirements and design) YOU MADE

1. **Utilizzeremo un server centrale**: si era valutato un’architettura con server locali e server centrale ma la complessità non era gestibile con le nostre conoscenze
2. **Utilizzeremo un DB non relazionale:** per rendere scalabile la persistenza dei dati
3. **Utilizzeremo un protocollo ethernet per la ricezione dei segnali dai sensori**
4. **Forniremo feedback ai sensori sul loro stato di anomalia (regolazione della frequenza di invio)**
5. **L’informazione sui valori di soglia verrà mantenuta su un file lato client, ottenuto all’autenticazione dal database**

G. Effort Recording



***PERT****Make a PERT documenting the tasks and timing you expect to spend on the deliverable. Try to be as precise as possible. Check, after the deliverable deadline, if and how you satisfied (or not) the deadlines.*

**File: PERT.pdf nel repository**

***Logging*** *As you are working on the assignment, record what you are doing and how long you spent. As a rule of thumb, you should add a log entry every time you switch tasks. For example, if you do something for two hours straight, that can be one log entry. However, if you do two or three things in half an hour, you must have a log entry for each of them. You do not need to include time for logging, but should include the time spent answering the other parts of this question.*

*For this purpose, please use the* ***LogTemplate.xls*** *file.*

***Categorization*** *When logging the time spent on the project, please create different sub- categories.**Specifically, it is important to clearly distinguish between two main categories: the time spent for “****learning****” (the modeling languages, the tools, etc.) from the time needed for “****doing****” (creating the models, taking the decisions, …). Learning tasks are in fact costs to be paid only once, while doing costs are those that will be repeated through the project.*

*For each category, please define sub-categories. Examples follow. You may add other sub-categories you find useful.*

|  |  |
| --- | --- |
| ***Learning***   * ***Requirements Engineering*** * ***Non functional Requirements*** * ***Use Case Diagrams*** * ***Tool study*** | ***Doing:***   * ***Requirements discovery*** * ***Requirements Modeling (UC diagrams)*** |

***Summary Statistics****Based on the attributes defined above, calculate the summary statistics of the time spent for “learning”, the time spent for “doing”, and the total time.*

***Note: this Deliverable report shall document only the Summary Statistics for the different deliverables (D1, D2, and Final). Detailed information shall be reported in the Excel file.***

***COPY HERE (computed from the spreadsheet): i) the total number of hours spent by the group (that is, hours per task X number of people working on that task), ii) the time spent for LEARNING and for DOING***

* **Tempo totale speso dal gruppo: 12 + 28 + 19 + 7 + 24 = 90 ore**
* **Tempo totale speso dal gruppo su Learning: 7 + 14 + 8 + 3 + 11 = 43 ore**
* **Tempo totale speso dal gruppo su Doing: 5 + 14 + 11 + 4 + 13 = 47 ore**