**“Software Engineering”**

**Course**

**a.a. 2018-2019**

**Template version 1.0**

**Deliverable #2**

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

**Dashboard Monitoraggio Ambientale**

|  |  |
| --- | --- |
| **Date** | 23/12/2018 |
| **Deliverable** | 2 |
| **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.0 Detailed Scenarios

*<Describe in extreme details at least 3 scenarios. Use those scenarios to better understand what the system has to do in much detail. Use the information coming from this study to refine the functional requirements below>*

* Il gestore di distretto effettua il login, entrerà quindi nella schermata dedicata alla visualizzazione dello stato di tutti gli edifici appartenenti al suo ditretto;
* Il gestore di città effettua il login, entrerà quindi nella schermata dedicata alla visualizzazione dello stato di tutti I ditretti presenti nella sua città; ne compare uno dove sono presenti delle anomalie, la segnalazione delle anomalie avverrà tramite l’utilizzo di segnali colorati, può quindi entrare nel livello di distretto per poter capire quale è l’edificio che presenta anomalie;
* Il gestore di edificio effettua il login, entrerà quindi nella schermata dedicate alla visualizzazione delle aree in cui è suddiviso il suo edificio. Se in un’area è presente un’anomalia, questa verrà sengalata tramite l’utilizzo di segnali colorati. Può quindi entrare nel livello dedicato alla sua area per poter capire il tipo di anomalia rilevata dai sensori e dal sistema.

A.1 Functional Requirements

*<List, in bullet points, all the functional requirements your system shall implement>   
<Describe functional requirements and stakeholders through Use Case Diagrams>. <Then, prioritize them, and provide a table-based description of the most important requirements>*

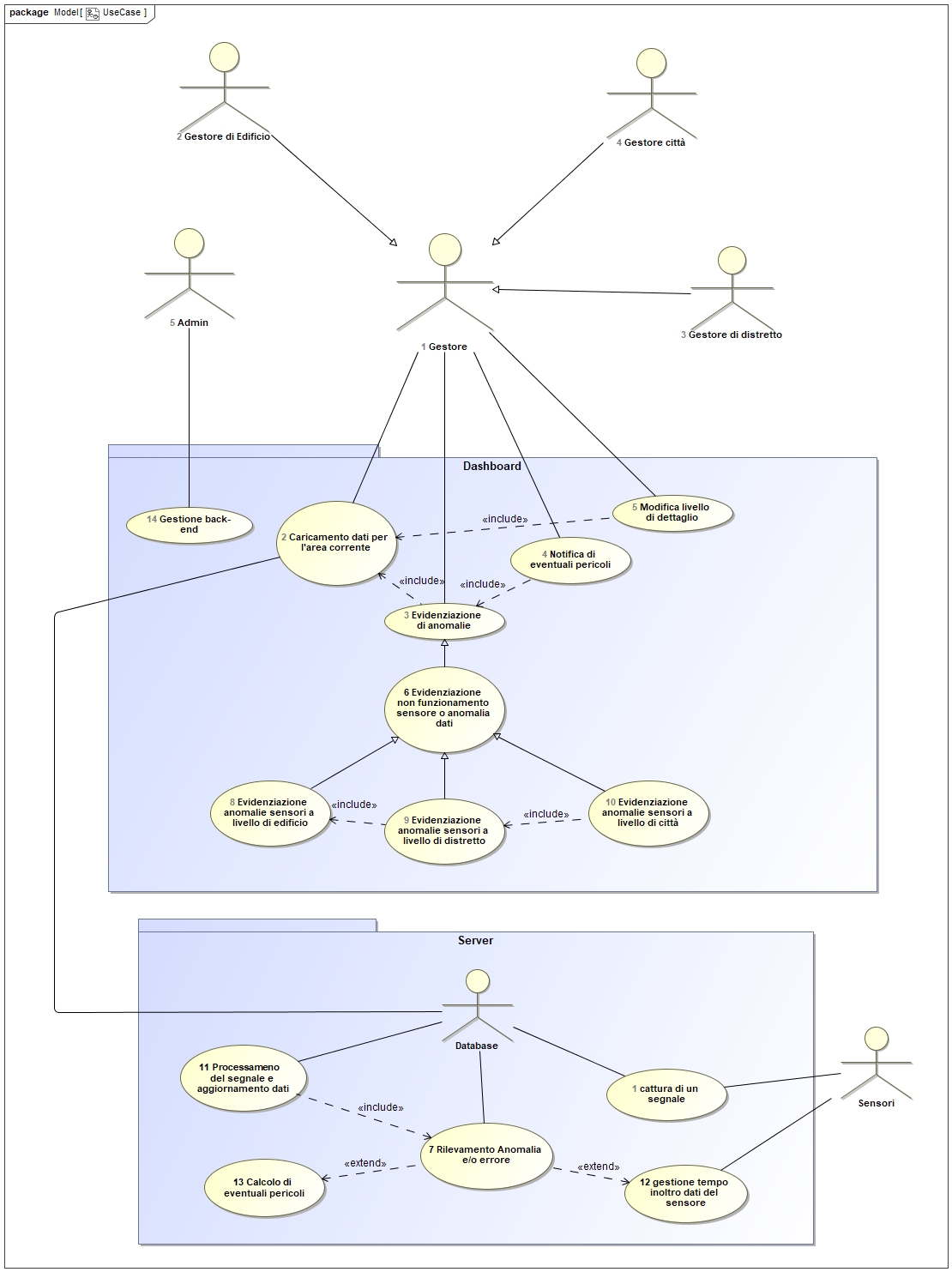
*<Provide a table-based description of the most important requirements, using the Alistair Cockburn Use Case Template>*

*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!!



**Figura 1.**  Use case diagram v2

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. Abbiamo 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.
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. L’aggiornamento dei dati in dashboard fa scattare il controllo sulle anomalie, mentre al termine del controllo sulle anomale , se ne sono state rilevate, verrà effettuato il controllo sui possibili pericoli; 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.

**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;

A1.1 GUI Requirements (da riempire a partire dalla Versione 2)

<*Please report here GUI-specific Requirements>*

La GUI deve:

* Essere intuitiva;
* Permettere che le anomalie debbano essere subito evidenziate effettuato l’accesso;
* Per evidenziare le anomalie si useranno I colori verde se non sono presenti anomalie, rosso se vi sono presenti anomalie o pericolic ed infine il giallo se vi sono anomalie di bassa importanza;
* Ogni gestore può scendere di un solo livello nel dettaglio, ad esempio il gestore di città può vedere al Massimo la dashboard del gestore di distretto, il gestore di edificio può scendere soltanto al livello area per vedere l’elenco dei sensori.

A1.2 Business Logic Requirements (da riempire a partire dalla Versione 2)

<*Please report here Business-logic specific Requirements>*

Il sistema dovrà:

* Calcolare la presenza di anomalie ad ogni singola ricezione di dato;
* In caso di presenza di anomalia modificare il campo alert di tutti I livelli di gestione;
* Calcolare la presenza di pericoli;

Questo garantirà la congruenza dei dati, aumenterà la velocità di gestione nel lato client, causerà un lavoro aggiuntivo nel lato server.

A1.3 DB Requirements (da riempire a partire dalla Versione 2)

<*Please report here DB-specific Requirements>*

|  |  |  |
| --- | --- | --- |
| USE CASE | #1: Caricamento 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 gestore sta visualizzando la 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 | Database  Gestore (generico) | |
| Trigger | Il timer d’aggiornamento dei dati scade | |
| DESCRIPTION | Step | Action | |
|  | 1 | Vengono richiesti i dati per l’area visualizzata al server. | |
|  | 2 | Se siamo a livello di edificio, controlliamo che i dati ricevuti da ogni sensore siano aggiornati:  Se si, il dato sostuisce il suo corrispondente nella dashboard  Altrimenti il dato non viene rimpiazzato ed inoltre se il dato ricevuto dal server risale ad oltre 3 minuti fa, segnala un’anomalia nel sensore responsabile (UC #6). | |
|  | 3 | La dashboard avvia il controllo sulle anomalie dei dati (UC #5). | |
| EXTENSIONS | Step | Branching Action | |
|  | 2a | Se siamo a livello di distretto o città:  A questo livello il sistema si limita a controllare nella risposta del server quali sono le sotto-aree che riportano anomalie nei dati (UC #8) | |

|  |  |
| --- | --- |
| RELATED INFORMATION | Caricamento dati per l'area corrente. |
| Priority: | Alta. |
| Frequency | Al più 50 volte al minuto. |
| Channels to actors | GUI |

**Figura 2**. Descrizione tabellare UseCase #1

|  |  |  |
| --- | --- | --- |
| USE CASE # | #2: Evidenziazione Anomalie | |
| Goal in Context | Riporta in dashboard informazioni visive sulle anomalie | |
| Scope & Level | Modulo anomalie del client - Primary Task | |
| Preconditions | Si sta visualizzando la dashboard | |
| Success End Condition | A livello di edificio:  I dati sopra (o sotto) la soglia vengono evidenziati  Le aree dell’edificio con multiple anomalie nei dati saranno marcate in modo molto evidente  Le aree dell’edificio in cui sono presenti sensori difettosi saranno contrassegnate più o meno marcatamente in base al numero di sensori rimanenti rispetto ogni parametro monitorato.  A livello di distretto e città:  Gli edifici e i distretti in cui sono presenti anomalie nei dati verranno evidenziati con colore che varia in base alla molteplicità delle anomalie rilevate nei livelli inferiori (con gradienti che vanno dal verde al rosso, in ordine crescente di anomalie). | |
| Failed End Condition | Le anomalie rilevate potrebbero non essere tutte evidenziate correttamente, fino alla prossima attivazione di questo UC. | |
| Primary,  Secondary Actors | Gestore (generico)  Database | |
| Trigger | Ricezione nuovi dati | |
| DESCRIPTION | Step | Action |
|  | 1 | Il modulo anomalie controlla nella risposta del server se sono presenti dati riguardanti anomalie |
|  | 2 | Se presenti, legge le informazioni e modifica i campi della dashboard per riflettere il tipo e la severità dell’anomalia |

|  |  |
| --- | --- |
| RELATED INFORMATION | Evidenziazione Anomalie |
| Priority: | Media priorità |
| Frequency | Fino a 50 / minuto |
| Channels to actors | GUI |

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

|  |  |  |
| --- | --- | --- |
| USE CASE # | #3: Notifica di eventuali pericoli | |
| Goal in Context | Dare un’allarme al gestore quando si riceve un broadcast di pericolo dal server che riguarda l’area di interesse | |
| Scope & Level | Modulo di notifica dei pericoli (Primary Task) | |
| Preconditions | Il gestore sta visualizzando i dati sulla sua dashboard | |
| 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 | La dashboard riceve una notifica di pericolo in broadcast dal server | |
| DESCRIPTION | Step | Action |
|  | 1 | La dashboad controlla le informazioni contenute nella notifica |
|  | 2 | Se la notifica riguarda la zona attualmente monitorata oppure una una sotto-zona, il sistema procede nell’identificazione del pericolo |
|  | 3 | Una volta appurato il tipo di pericolo e la zona o sotto-zona interessata, il sistema farà apparire in dashboard un’allarme che punta alla entry relativa alla zona in cui è stato rilevato il pericolo. |

|  |  |
| --- | --- |
| RELATED INFORMATION | Notifica di eventuali pericoli |
| Priority: | Alta |
| Frequency | Caso peggiore: 750 < n < 1500 al minuto |
| Channels to actors | GUI |

**Figura 4.** Descrizione tabellare UseCase #3.

|  |  |  |
| --- | --- | --- |
| USE CASE # | #4: 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 | Dashboard gestore – (primary task) | |
| 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, Admin (?)  Database | |
| Trigger | Il gestore selezione una voce dalla lista di aree che sono di sua competenza. | |
| 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. |
| EXTENSIONS | Step | Branching Action |
|  | 1a | Se il gestore è già sceso di un livello:  La dashboard non permetterà di fare una selezione |

|  |  |
| --- | --- |
| RELATED INFORMATION | Modifica livello di dettaglio. |
| Priority: | Medio |
| Frequency | Variabile (l’attivazione è interattiva, non automatica) |
| Channels to actors | Interattivo |

**Figura 5.** Descrizione tabellare UseCase #4.

|  |  |  |
| --- | --- | --- |
| USE CASE # | #5: Evidenziazione Anomalie Dati | |
| Goal in Context | Al gestore viene notificata, in base al livello di competenza, un’eventuale anomalia sui dati. | |
| Scope & Level | Modulo di Calcolo delle anomalie | |
| 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 | Database  Gestore (generico) | |
| Trigger | Viene completato il caricamento dei (nuovi) dati in dashboard | |
| DESCRIPTION | Step | Action |
|  | 1 | Nel caso di dati da singoli sensori, ovvero se il gestore di edificio abbia fatto lo zoom su una stanza:  Il Sistema controlla quanto i dati appena ricevuti si discostano dalla soglia massima. |
|  | 2 | 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 | Evidenziazione Anomalie Dati |
| Priority: | Alta |
| Performance | <the amount of time this use case should take> |
| Frequency | Almeno 50/minuto |
| Channels to actors | database |

**Figura 6.** Descrizione tabellare UseCase #5.

|  |  |  |
| --- | --- | --- |
| USE CASE # | #6: 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 | Modulo di calcolo delle anomalie | |
| 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 potrebbero non essere tutti evidenziati fino al prossimo avvio di questo caso d’uso | |
| Primary,  Secondary Actors | Database  Gestore di edificio, Gestore di distretto (dopo essere sceso di un livello) | |
| Trigger | Completato il caricamento dei dati in dashboard | |
| DESCRIPTION | Step | Action |
|  | 1 | Se si sta visualizzando la dashboard relativa ad una singola stanza o area dell’edificio:  Il Sistema recupera dalla risposta del server i dati relativi al funzionamento dei sensori presenti nella stanza attualmente monitorata nonché il timestamp dell’ultimo segnale ricevuto |
|  | 2 | Il sistema controlla quanti sensori per ogni tipo di dato monitorato sono ancora in funzione nella stanza; un sensore non è in funzione se e solo se l’ultimo segnale risale ad oltre 3 minuti fa OPPURE il sensore ha riportato di essere malfunzionante |
|  | 3 | Evidenzia il parametro (temperatura,…) in grigio se c’è almeno un sensore di backup, altrimenti evidenzia in nero. |
| EXTENSIONS | Step | Branching Action |
|  | 1a | Se si sta visualizzando la dashboard relativa ad un intero edificio:  Il Sistema recupera dalla risposta del server i dati relativi al funzionamento dei sensori presenti in tutte le stanze e aree dell’edificio attualmente monitorato nonché il timestamp dell’ultimo segnale ricevuto da ognuno di essi. |
|  | 2a | Il sistema controlla quanti sensori per ogni stanza sono in funzione |
|  | 3a | Evidenzia il box relativo ad una stanza in grigio se almeno un sensore è malfunzionante, evidenzia in nero se almeno il 50% dei sensori nella stanza sono malfunzionanti |

|  |  |
| --- | --- |
| RELATED INFORMATION | Evidenziazione anomalie nei sensori a livello di edificio |
| Priority: | Media priorità |
| Frequency | Almeno 50 / minuto |

**Figura 7.** Descrizione tabellare UseCase #6.

|  |  |  |
| --- | --- | --- |
| USE CASE # | #20: Cattura di un segnale | |
| Goal in Context | Ricevere un segnale in ingresso e passarlo al buffer | |
| Scope & Level | Ricettore segnali - Primary Task | |
| Preconditions | Il server è in funzione | |
| Success End Condition | Il segnale viene messe in coda nel buffer d’ingresso | |
| Failed End Condition | Il segnale viene scartato oppure la fonte viene incorrettamente bollata come estranea | |
| Primary,  Secondary Actors | Sensore  Server | |
| Trigger | Ricezione di un segnale mandato da un sensore | |
| DESCRIPTION | Step | Action |
|  | 1 | Il Server controlla l’intestazione del segnale, dove potrà riscontrare che sia effettivamente un segnale inviato da un sensore |
|  | 2 | Se risulta in regola, il payload del segnale viene aggiunto in coda nel buffer d’ingresso |
| EXTENSIONS | Step | Branching Action |
|  | 2a | Se il buffer d’ingresso è pieno:  Il segnale verrà scartato (ciò potrebbe indurre il sistema a pensare che sia un problema con il sensore da cui ha avuto origine) |

|  |  |
| --- | --- |
| RELATED INFORMATION | Cattura di un segnale |
| Priority: | Alta |
| Performance | <the amount of time this use case should take> |
| Frequency | Almeno 150.000 / minuto |

**Figura 8.** Descrizione tabellare UseCase #20

|  |  |  |
| --- | --- | --- |
| USE CASE # | #21: Processamento segnale e aggiornamento dati | |
| Goal in Context | Identificare il sensore da cui si è ricevuto il segnale e aggiornarne i dati. | |
| Scope & Level | Database (Primary Task) | |
| Preconditions | Il server è in funzione | |
| Success End Condition | Il sensore inviante viene correttamente identificato e I suoi dati aggiornati. | |
| Failed End Condition | Il segnale catturato contiene errori. | |
| Primary,  Secondary Actors | Server  Sensore | |
| Trigger | È presente almeno un payload nel buffer d’ingresso (UC #20) | |
| DESCRIPTION | Step | Action |
|  | 1 | Viene prelevato il payload di un segnale dal buffer d’ingresso |
|  | 2 | Vengono ricavati ID sensore, variabile ambientale e stato di funzionamento dal segnale. |
|  | 3 | Vengono aggiornati i dati del sensore in DB. |
| EXTENSIONS | Step | Branching Action |
|  | 1a | Formato dei dati ricavati non corretto o ID non presente nel database.  I dati vengono scartati |

**Figura 9.** Descrizione tabellare UseCase #21.

|  |  |  |
| --- | --- | --- |
| USE CASE # | #22: Rilevamento anomalia e/o errore. | |
| Goal in Context | In seguito all’aggiornamento dei dati (di un sensore), vengono controllati lo stato di funzionamento del sensore e la variabile ambientale misurata (confronto col massimale). | |
| Scope & Level | Rilevatore anomalia e/o errore ,Primary Task. | |
| Preconditions | Il server ha catturato e processato e il segnale inviato da un sensore, e ne ha aggiornato i dati. | |
| Success End Condition | In base all’esito del controllo, l’alert del sensore e dei livelli superiori (area, edificio e distretto) che lo contengono, vengono opportunamente modificati o lasciati invariati. | |
| Failed End Condition | Azione sbagliata sull’alert del sensore, che si trascina anche ai livelli superiori. | |
| Primary,  Secondary Actors | Server.  Sensore | |
| Trigger | Aggiornamento dati relativi ad un sensore. | |
| DESCRIPTION | Step | Action |
|  | 1 | Vengono aggiornati i dati relativi ad un sensore. |
|  | 2 | Vengono controllati la variabile ambientale misurata e lo stato di funzionamento. |
|  | 3 | Uno dei due valori o entrambi portano alla modifica dell’alert del sensore e ai livelli superiori. |
| EXTENSIONS | Step | Branching Action |
|  | 3a | I valori controllati non sono tali provocare anomalia o errore.  L’alert non viene modificato. |

|  |  |
| --- | --- |
| RELATED INFORMATION | #22: Rilevamento anomalia e/o errore. |
| Priority: | Alta |
| Frequency | Almeno 150.000/minuto (in media) |
| Channels to actors | Database |
| OPEN ISSUES | Range di valori per l’alert. |
| Superordinates | Processamento del segnale e aggiornamento dati. |

**Figura 10.** Descrizione tabellare UseCase #22.

|  |  |  |
| --- | --- | --- |
| USE CASE # | #23: Gestione tempo inoltro dati del sensore | |
| Goal in Context | Aumentare/Diminuire la frequenza con cui il sensore invia i dati | |
| Scope & Level | Modulo di calcolo delle anomalie | |
| Preconditions | Il server deve aver già ricevuto alcuni dati dal sensore, nei quali ha riscontrato delle anomalie. | |
| Success End Condition | La frequenza di inoltro del dato rilevato dal sensore viene aumentata quando si riscontrano anomalie e diminuita quando l’anomalia non viene più riscontrata. | |
| Failed End Condition | Non è possibile modificare la frequenza | |
| Primary,  Secondary Actors | Server  Sensore | |
| Trigger | Maggiore quantità di dati ricevuti | |
| DESCRIPTION | Step | Action |
|  | 1 | Il sistema ha riscontrato la presenza di anomalie nei dati ricevuti / il sistema riscontra che l’anomalia non è più presente |
|  | 2 | Il sistema imposta un arco di tempo minore per l’inoltro del dato oppure quando non più presente l’anomalia il sistema ripristina l’arco di tempo ad 1 minuto |

|  |  |
| --- | --- |
| RELATED INFORMATION | Gestione tempo inoltro dati del sensore |
| Priority: | Media priorità |
| Frequency | Al Massimo 150000/ minuto ( caso in cui tutti I sensori hanno un anomalia |
| Subordinates | <optional, depending on tools, links to sub.use cases> |

**Figura 11.** Descrizione tabellare UseCase #23.

|  |  |  |
| --- | --- | --- |
| USE CASE # | #24: Calcolo di eventuali Pericoli | |
| Goal in Context | Vengono riscontrate le presenze di eventuali pericoli | |
| Scope & Level | Modulo di calcolo dei pericoli | |
| Preconditions | Il server deve aver già ricevuto alcuni dati dal sensore, nei quali ha riscontrato delle anomalie  Nella stessa stanza sono state riscontrate diverse anomalie da diversi tipi di sensori | |
| Success End Condition | Viene riportato un allarme di tipo grave per presenza di pericoli | |
| Failed End Condition | Non è possibile calcolare il pericolo | |
| Primary,  Secondary Actors | Server | |
| Trigger | Maggiore quantità di dati ricevuti | |
| DESCRIPTION | Step | Action |
|  | 1 | Il sistema ha riscontrato la presenza di anomalie nei dati ricevuti da sensori di vario genere presenti nella stessa area |
|  | 2 | Il sistema, utilizzando le opportune regole di calcolo, determina quale sia il pericolo |
|  | 3 | Il sistema segnala il pericolo a tutti i livelli di gestione |

|  |  |
| --- | --- |
| RELATED INFORMATION | Calcolo di eventuali Pericoli |
| Priority: | Media priorità |
| Frequency | Nel Peggiore dei casi (pericoli in tutte le aree)  750<n <1500 |

**Figura 12.** Descrizione tabellare UseCase #24.

A.2 Non Functional Requirements

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

**-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)

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 posizioniamo 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, finora 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 una gestione dell’intervallo di tempo in cui il sensore invia il segnale 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 >*

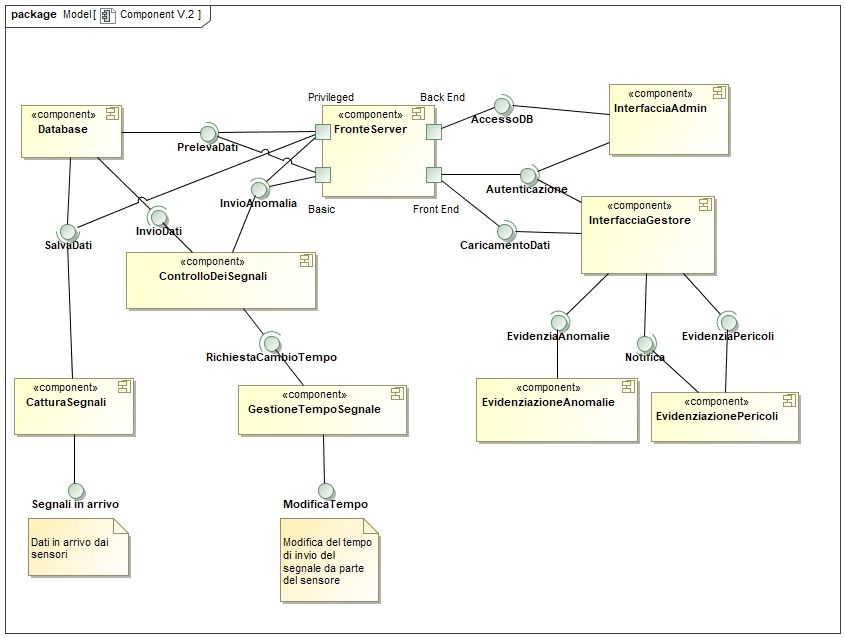
C.1The static view of the system: Component Diagram

*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 13.** Component Diagram rappresentante la decomposizione del nostro sistema.

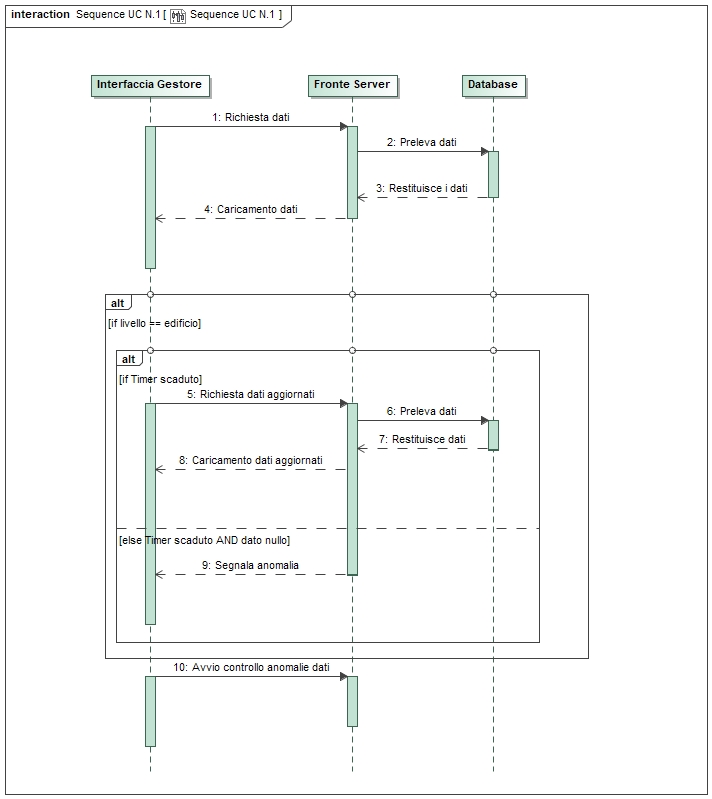


Come si può notare dai vari use case ed il component Diagram ci sono state molte modifiche dal primo deliverable. Le principali riguardano la gestione del dato direttamente utilizzando funzioni server, così da non fare i vari controlli sui dati in ogni singolo client, alleggerendo così il compito del client, evitanto inoltre il dover permettere la copia dei dati in ogni livello rischiando la Perdita di persistenza dei dati stessi, diminuendo quindi il quantitative di dati trasmessi e ricevuti dal server.

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

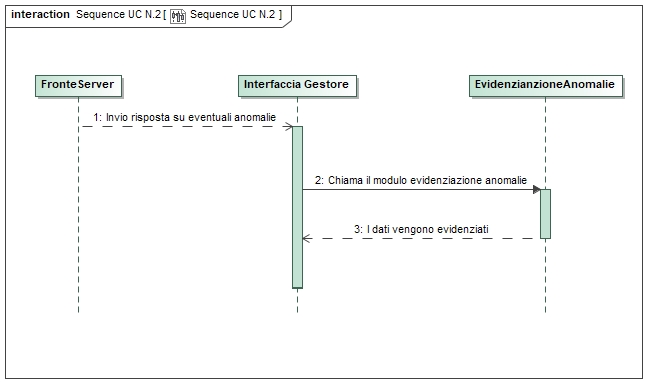
*BE SURE THAT THE STRUCTURE IS SYNCHRONIZED WITH THE DYNAMIC VIEW*

**Caricamento dati per l’area corrente**



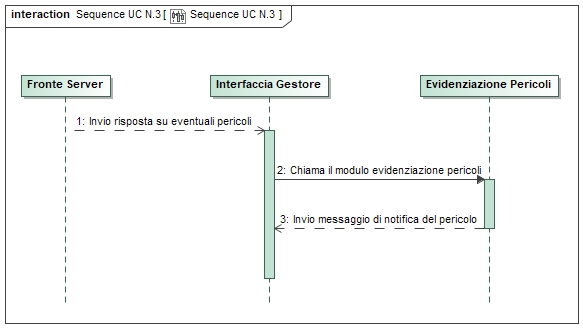
**Figura 14 –** Sequence Diagram per lo UseCase #1

**Evidenziazione di anomalie**



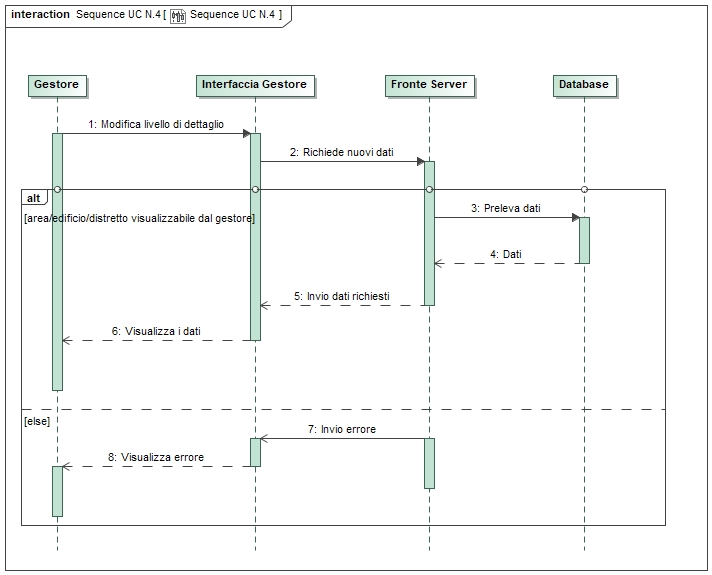
**Figura 15 –** Sequence Diagram per lo UseCase #2

**Notifica di eventuali pericoli**



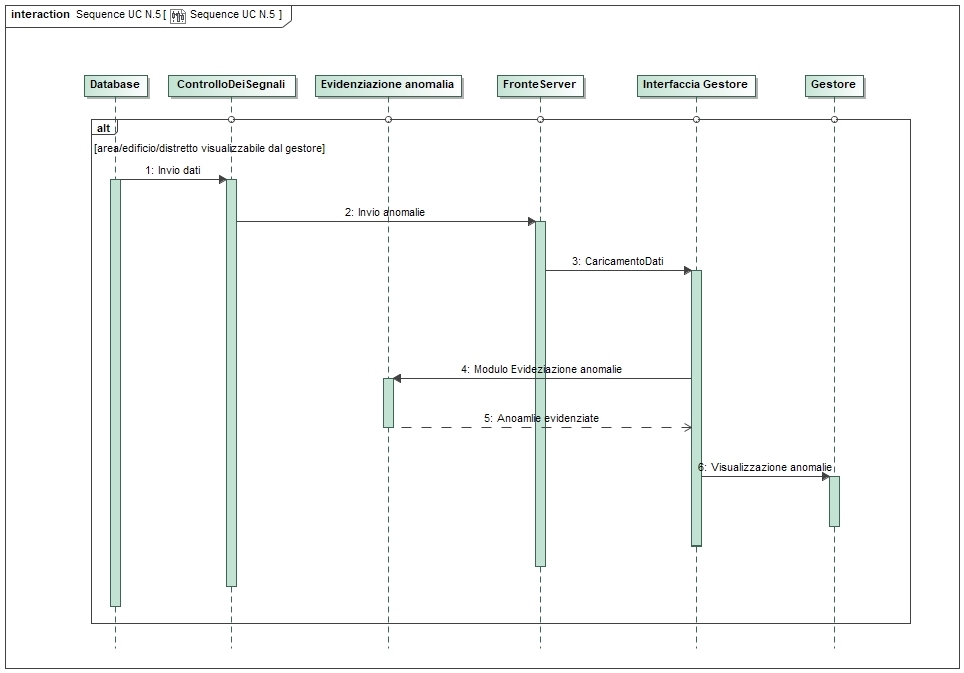
**Figura 16 –** Sequence Diagram per lo UseCase #3

**Modifica di livello di dettaglio**



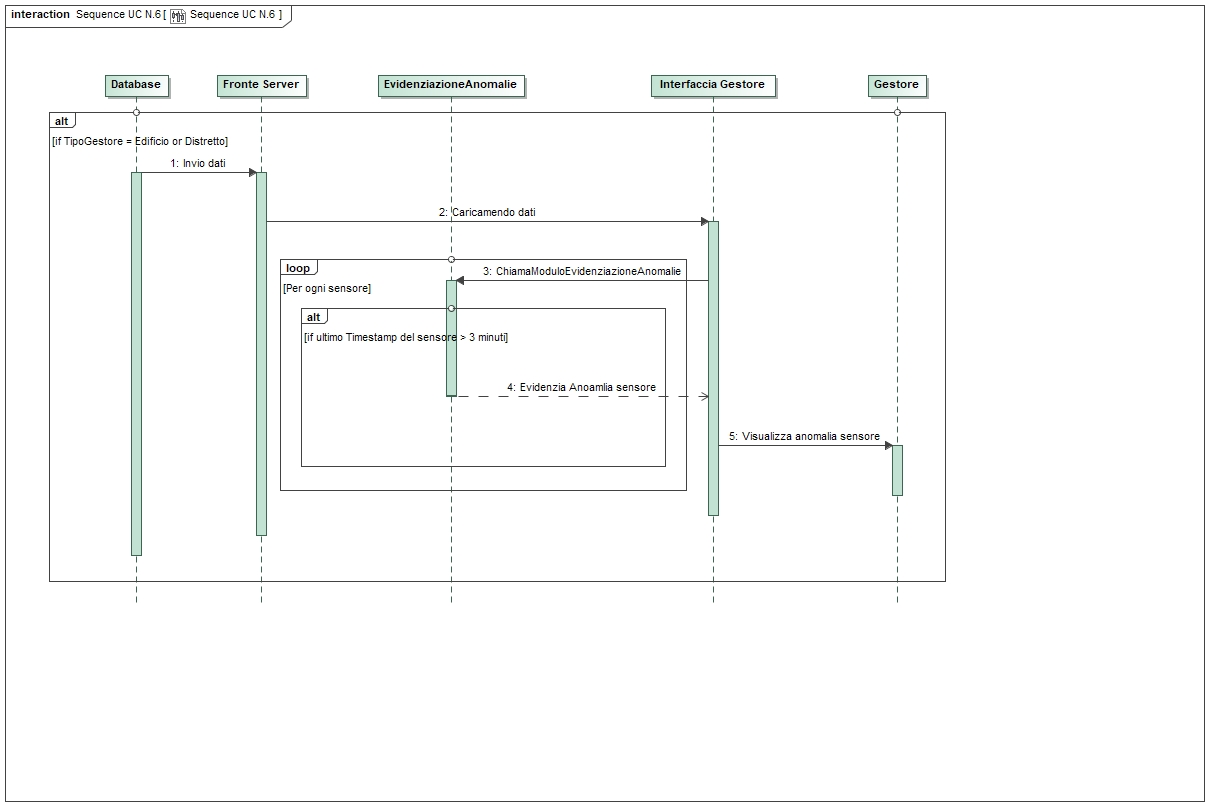
**Figura 17 –** Sequence Diagram per lo UseCase #4

**Evidenziazione anomalie dei dati**



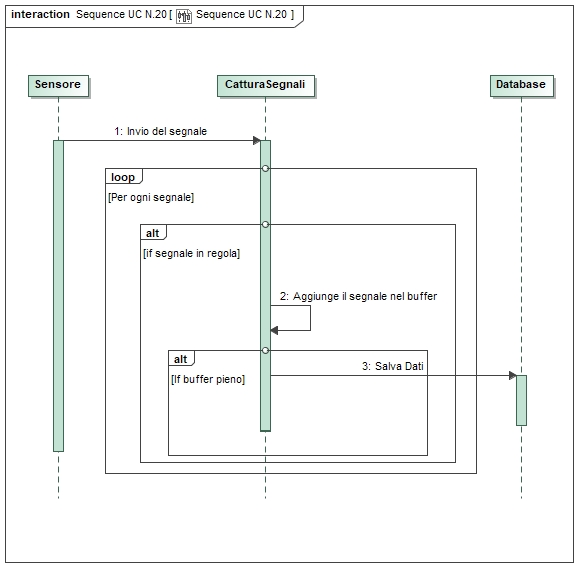
**Figura 18 –** Sequence Diagram per lo UseCase #5

**Evidenziazione anomalie sensori a livello di edificio**



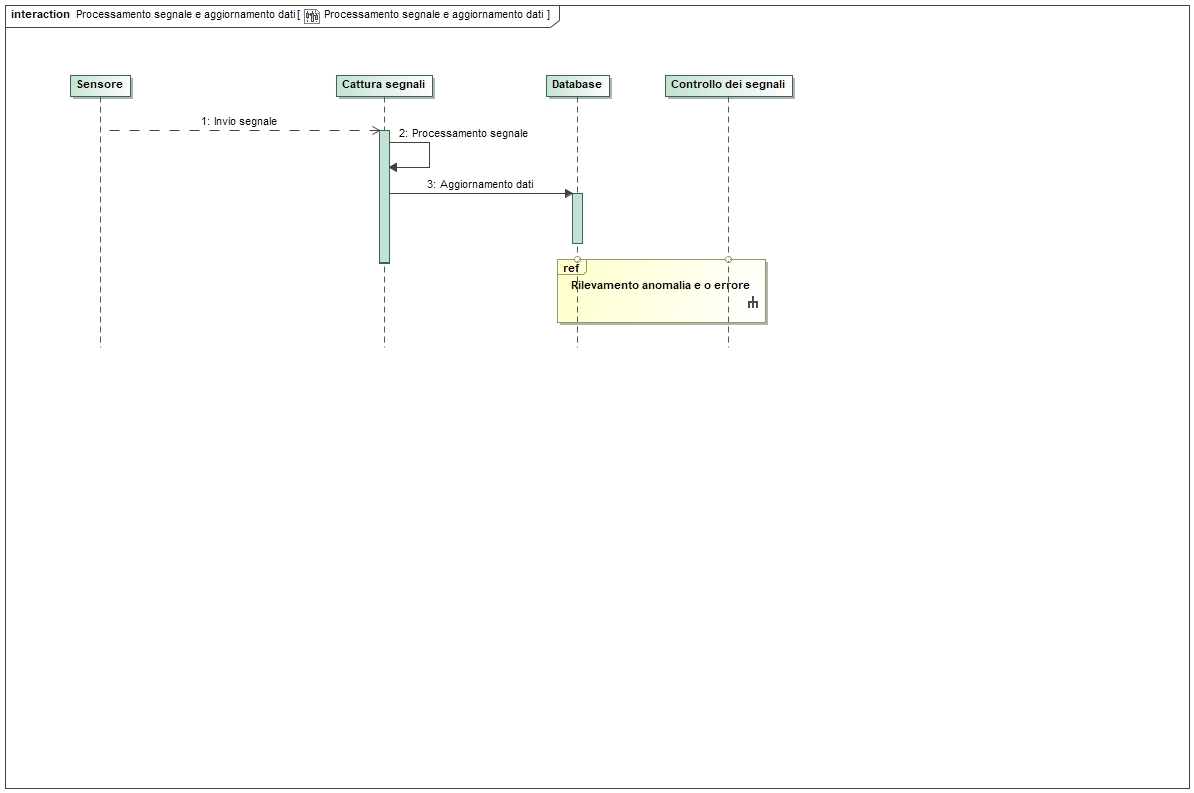
**Figura 19 –** Sequence Diagram per lo UseCase #6

**Cattura di un segnale**



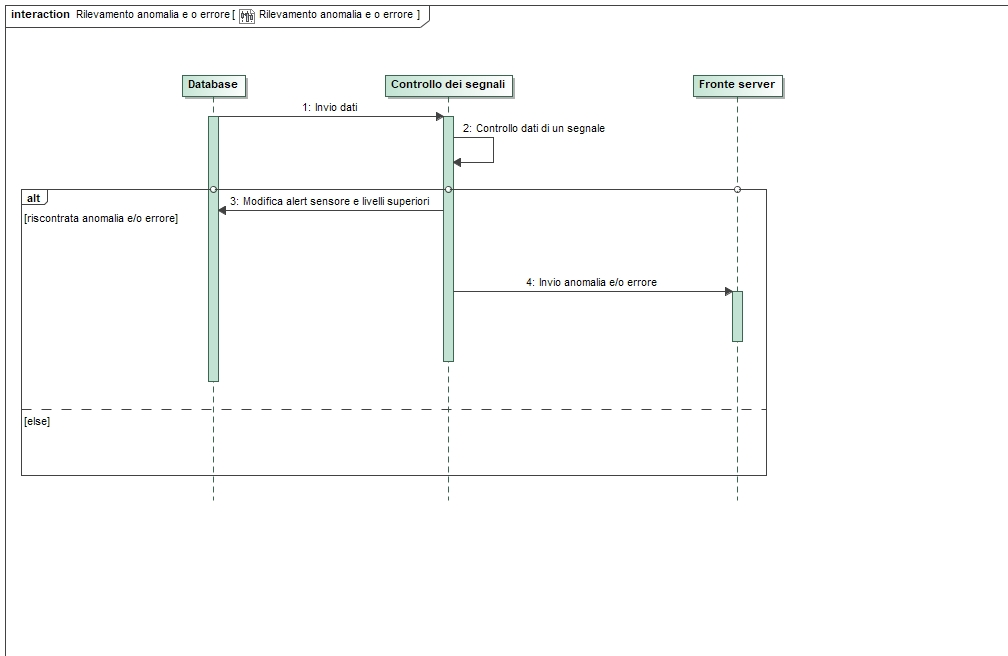
**Figura 20 –** Sequence Diagram per lo UseCase #20

**Processamento del segnale e aggiornamento dati**



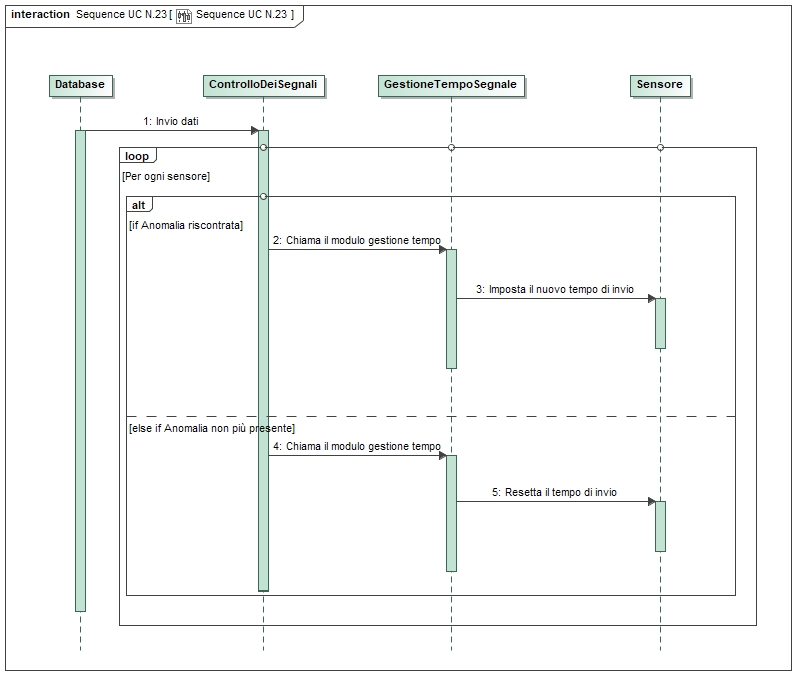
**Figura 21 –** Sequence Diagram per lo UseCase #21

**Rilevamento anomalia e/o errore**



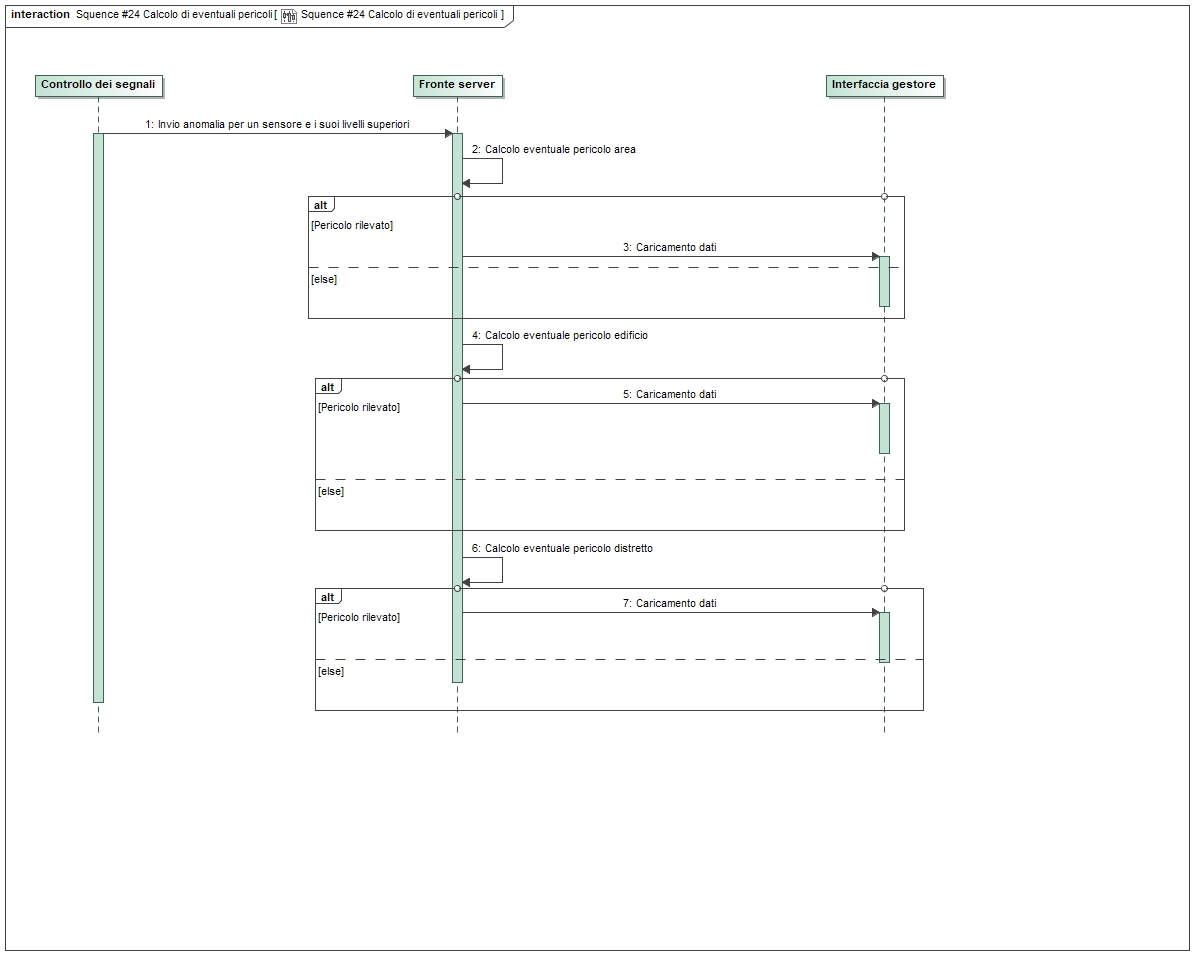
**Figura 22 –** Sequence Diagram per lo UseCase #22

**Gestione tempo inoltro dati del sensore**



**Figura 23 –** Sequence Diagram per lo UseCase #23

**Calcolo di eventuali pericoli**



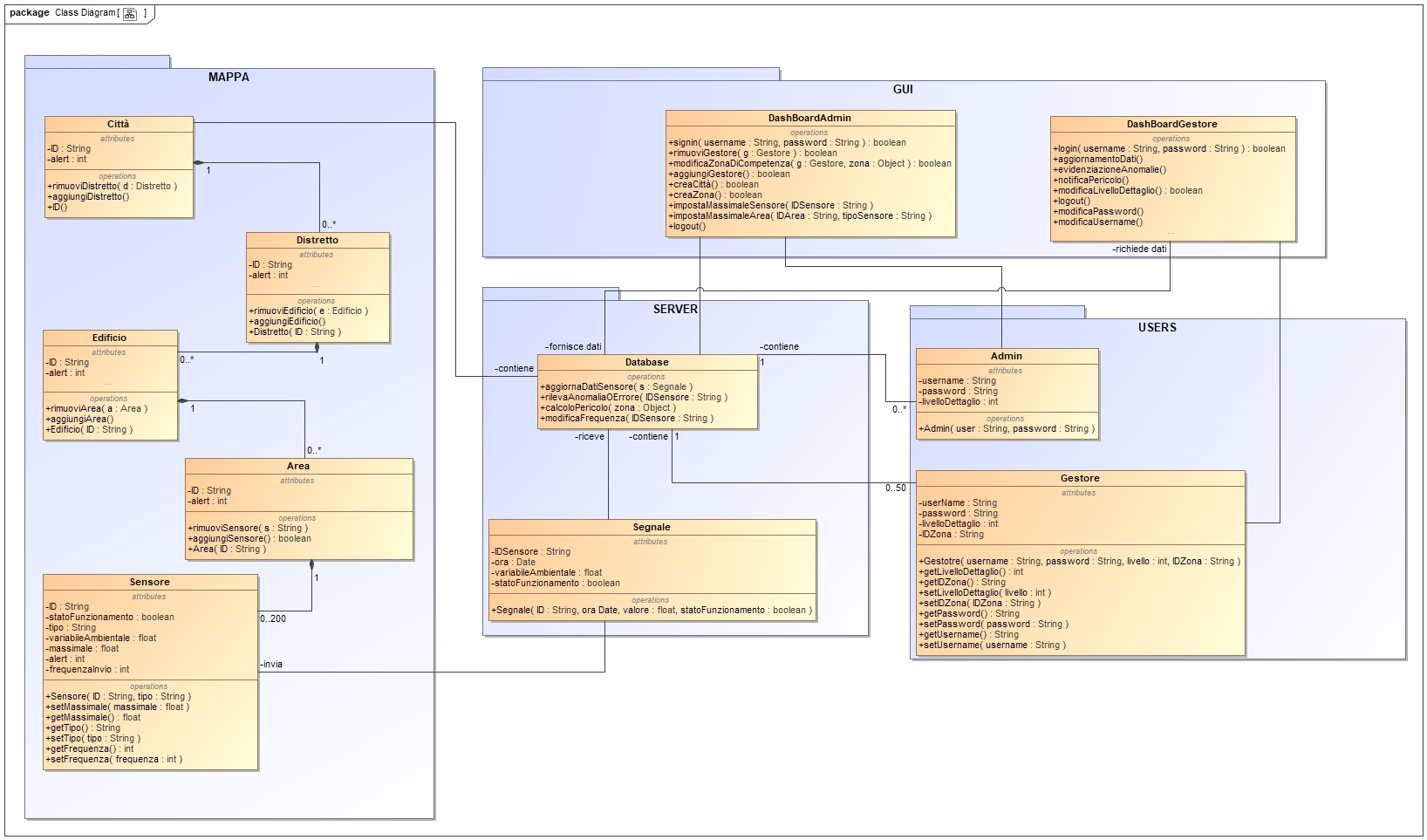
**Figura 24 –** Sequence Diagram per lo UseCase #24

Nota: non sono presenti I sequence diagram e le tabelle degli usecase 7 ed 8, questi sono una specificazione di quello che avviene nello usecase 5, quindi sono stati omessi per brevità

C. ER Design

<Report here the Entity Relationship Diagram of the system DB>

D. Class Diagram of the implemented System



**Figura 25 –** Immagine raffigurante il Class Diagram

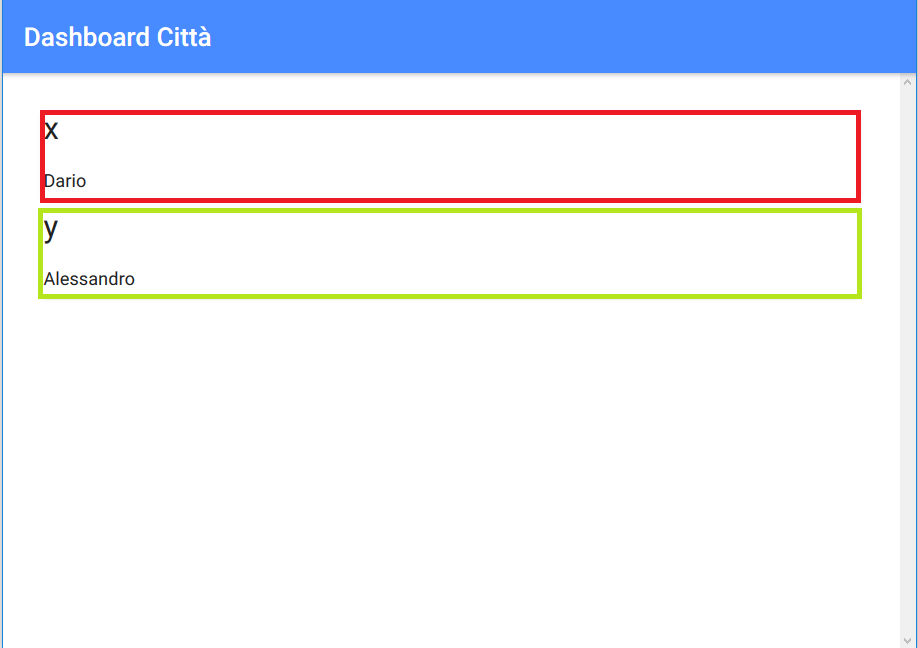
E. Design Decisions   
<Document here the **5** most important design decisions you had to take. You can use both a textual or a diagrammatic specification.>

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 ed un DB non relazionale**: si era valutato un’architettura con server locali e server centrale ma la complessità non era gestibile con le nostre conoscenze, inoltre per rendere scalabile la persistenza dei dati utilizzeremo un DB non relazionale
2. **Realizzeremo una webApp:** si renderà quindi possibile l’accesso da un qualunque dispositivo, verrà implementate utilizzando il framework Ionic.
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. **Il dato del singolo sensore non verrà visualizzato da tutti i livelli di gestione,** l’unico gestore che potrà visualizzare il dato sarà il gestore di edificio, quindi ai gestori dei livelli superiori (distretto e città) verrà semplicemente comunicato che nel livello immediatamente successivo vi è un errore/anomalia

F. Explain how the FRs and the NFRs are satisfied by design

*<Report in this section how the design you produced satisfies the FRs and the NFRs>*



**Figura 26 –** Immagine raffigurante la schermata della dashboard del gestore di città

Nella schermata soprariportata vi è il frutto principale delle modifiche apportate in questo deliverable, l’evidenziazione di anomalie nei livelli superiori con il semplice utilizzo di colorazioni differenti, il tutto riportando dal livello più basso a quello più alto la semplice presenza di una anomalia.

Secondo quanto descritto nel precedente deliverable, le anomalie si sarebbero dovute calcolare in ogni singola dashboard dopo l’aver effettuato la richiesta di tutti i dati presenti nella propria sezione di gestione della città, creando quindi molti problemi nella trasmissione dei dati, nella congruenza dei dati, nella continua necessità di aggiornare i dati lato client ed ogni volta calcolare anomalie e pericoli.

Secondo quanto descritto in questo deliverable ad ogni singola ricezione di un dato da parte del server vengono calcolate anomalie ed eventuali presenze di pericoli, così da dover aggiornare nei vari livelli di gestione un semplicissimo campo di allarme chiamato Alert.

In questo deliverable sono state apportate una serie di modifiche di quello che era stato descritto nel precedente documento. L’effettuare queste modifiche ha portato via molto tempo in questa fase, ci ha però permesso di migliorare la qualità del documento e soprattutto di quello che sarà il prodotto finale.

Sono scarne le sezioni riguardanti il prototipo perché si è voluta dare una maggiore importanza alla ridefinizione della gestione dei dati.

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.*

***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:21+ 9+ 24+ 18+ 4= 76 ore**
* **Tempo totale speso dal gruppo su Learning: 10+ 4+ 6+ 4= 24 ore**
* **Tempo totale speso dal gruppo su Doing: 11+ 9+ 20+ 12 = 52ore**

Appendix. Code   
*<Report in this section a* ***documented*** *version of the produced code. I do not need a copy&cut of your code, but rather an explanation of how your code satisfies the Functional and Non functional requirements.  
<Show some screenshots of the code behavior>*

*<please upload your executable code in the dropbox folder>*