**“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** | Deliverable n.1 |
| **Team (Name)** | G-SELL |

|  |  |  |
| --- | --- | --- |
| **Team Members** | | |
| **Name & Surname** | **Matriculation Number** | **E-mail address** |
| Enrico Monte | *246980* | *enrico.monte@student.univaq.it* |
| Giuseppe Gasbarro | *246886* | *giuseppe.gasbarro@student.univaq.it* |
| Lorenzo De Filippis | *230011* | *lorenzo.defilippis@student.univaq.it* |
| Loreto Cicerone | *244333* | *loreto.cicerone@student.univaq.it* |
| Stefano Florio | *246817* | *stefano.florio@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

**List of Challenging/Risky Requirements or Tasks**……………………………………………4

Challenging Task Table……………………………………………………………………………….4

Glossario…………………………………………………………………………………………………….5

**A. Requirements Collection**…………………………………………………………………………….6

A.1 Functional Requirements……………………………………………………………………..6

A.1.1 Use Case Diagrams…………………………………………………………………….........7

A1.2 Tabular description of the most relevant use cases…………………………...11

A.2 Non Functional Requirements……………………………………………………………..18

A.3 Excluded Requirements……………………………………………………………………….19

A.4 Assumptions………………………………………………………………………………………..19

A.5 Prioritization………………………………………………………………………………………..20

**B. Software Architecture**………………………………………………………………………………..21

C.1The static view of the system: Component Diagram……………………………..21

C.2 The dynamic view of the software architecture: Sequence Diagram...…23

**E. Design Decisions**…………………………………………………………………………………………28

**G. Effort Recording**…………………………………………………………………………………………28

PERT…………………………………………………………………………………………………………..28

Logging ……………………………………………………………………………………...................29

Summary statistics……………………………………………………………………………………..31

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** |
| Identificare un ruolo preciso per ogni tipo gestore. | 19/11/18 | 30/11/18 | Il GestoreEdificio ha il compito di avvertire le autorità in caso di allerta o pericolo, e di settare i valori per le stanze. Il GestoreZona invece deve controllare solo gli edifice della sua zona; allo stesso mod oil GestoreCittà monitora solo le zone della città. |
| Identificare un sistema di comunicazione tra i gestori | 26/11/18 | 30/11/18 | Creazione del Feedback System, illustrato nel component e nel system diagram. |
| Definire dei livelli di allerta distinti e fare in modo che il Sistema capsica una situazione di pericolo. | 23/11/18 | 26/11/18 | Creazione dei component CalcoloPriorità e CalcoloPericolo. |
|  |  |  |  |

**GLOSSARIO**

* SENSORE = Dispositivo con un codice univoco che misura determinate **variabili ambientali** (temperatura, umidità, pressione, luminosità, fumo) ed invia i valori rilevati al sistema.
* MAX VALUE = **Soglia** **massima** per una variabile ambientale rilevabile da un sensore, in una determinata aerea; al superamento del MAX VALUE viene attivato un segnale di allerta.
* MIN VALUE = **Soglia minima** per una variabile ambientale rilevabile da un sensore, in una determinata aerea; quando il valore scende al di sotto del MIN VALUE viene attivato un segnale di allerta.
* ANOMALIA = Un’anomalia si verifica quando una o più variabili ambientali rilevate da un sensore non rientrano nel **range** [MIN VALUE,MAX VALUE]
* GESTORE = Monitora una determinata area. Il gestore si divide in:
  + **GestoreCittà** (supervisione di città)
  + **GestoreZona** (supervisione di zone)
  + **GestoreEdificio** (supervisione di edifici, piani, stanze).
* AREA = Area geografica nella quale vi sono posizionati sensori.

Comprende: **Città, Zone, Edifici, Piani, Stanze**(corridoio, atrio, aula, parcheggio ecc.).

* STATO AREA = Valori medi delle variabili ambientali per l’area in questione e se quest’ultima presenta o meno anomalie o malfunzionamenti, con il relativo livello di allerta.

A. Requirements Collection

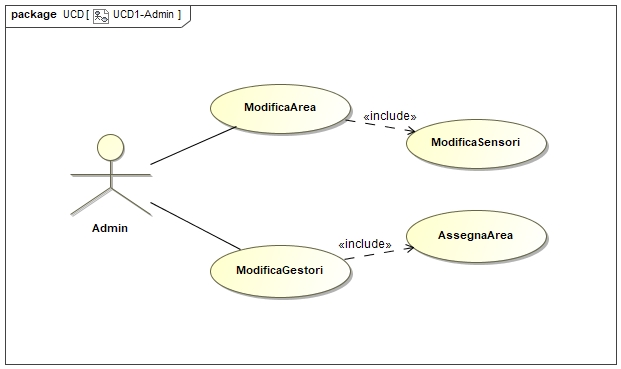
A.1 Functional Requirements

* I **sensori** devono essere in grado di inviare i dati rilevati al sistema, ad intervalli regolari.
* Il **gestore città** deve essere in grado di monitorare la città ad esso assegnata, visualizzando lo stato di tutte le zone.
* Il **gestore città**  può richiedere al gestore zona un feedback relativo ad un’anomalia/malfunzionamento in una zona.
* Il **gestore zona** deve essere in grado di monitorare la zona ad esso assegnata, visualizzando lo stato di tutti gli edifici.
* Il **gestore zona**  può richiedere al gestore edificio un feedback relativo ad un’anomalia/malfunzionamento in un edificio.
* Il **gestore edificio** deve essere in grado di monitorare l’edificio ad esso assegnato, visualizzando lo stato di tutti i piani e di tutte le stanze; per ogni stanza può visualizzare i singoli sensori, con le relative variabili ambientali rilevate.
* Il **gestore edificio** può modificare i MIN VALUE e MAX VALUE relativi ad una stanza.
* Il **gestore edificio** in caso di pericolo deve essere in grado di allertare le autorità competenti.
* La **dashboard** deve notificare al gestore anomalie e malfunzionamenti relativi ai sensori, fornendo dei livelli di allerta in base all’entità del problema.
* La **dashboard** deve notificare al gestore eventuali situazioni di pericolo.
* La **dashboard** deve fornire una visualizzazione delle aree diversa in base al tipo di gestore in servizio.
* L’**Admin** deve poter inserire o modificare un gestore, e assegnare determinate aree a determinati gestori.
* L’**Admin** deve poter inserire o modificare un sensore.

A1.1 Use Case Diagrams

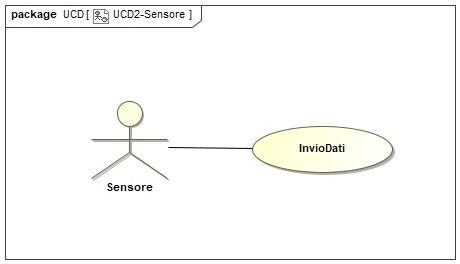
**Figure 1: UCD dell’Admin**

L’Admin si occupa dell’amministrazione di tutto il sistema.



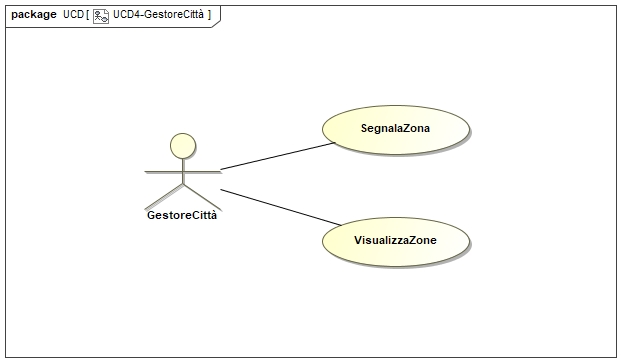
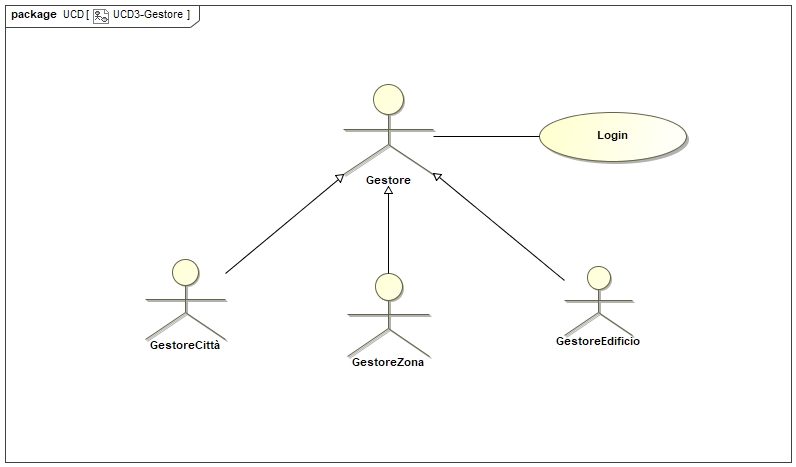
**Figure 2: UCD del Sensore**

I Sensori mandano periodicamente i dati prelevati al DB; non sono interrogabili dal sistema.



**Figure 3: UCD del Gestore**

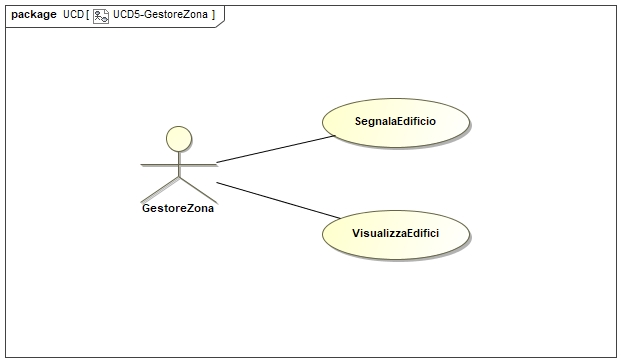
Specializzazioni del Gestore.



**Figure 4: UCD del GestoreCittà**

Il GestoreCittà ha una panoramica delle zone di tutta la città.

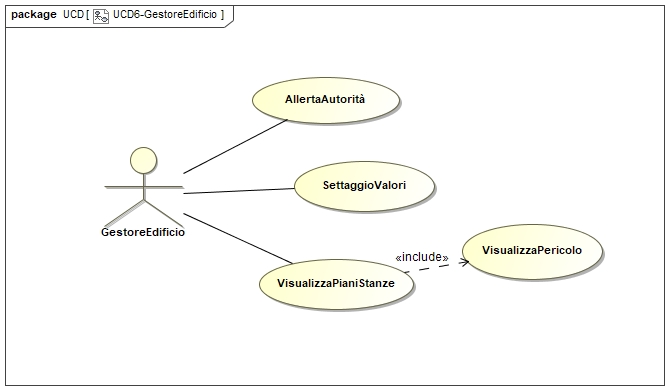
**Figure 5: UCD del GestoreZona**



Il GestoreZona controlla tutti gli edifici della sua zona.

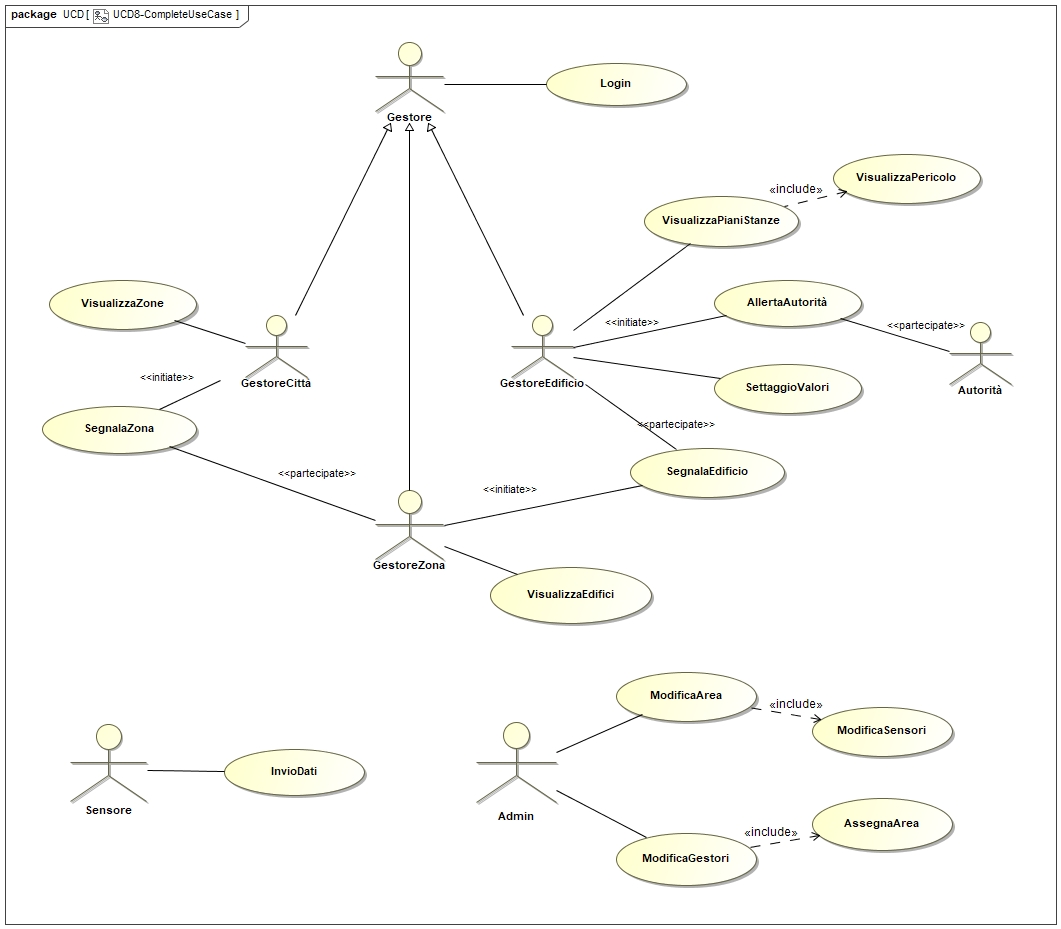
**Figure 6: UCD del GestoreEdificio**

Il GestoreEdificio è il più importante: in caso di allerte o pericoli è lui che avverte le autorità; inoltre è il responsabile per il settaggio dei valori minimali e massimali.



**Figure 7: UCD Completo**

Visione completa del sistema e dei suoi attori.



A1.2 Tabular description of the most relevant use cases

* UC1

|  |  |
| --- | --- |
| *Use case name* | **VisualizzaPianiStanze** |
| *Participating actors* | **GestoreEdificio** |
| *Flow of events* | 1. Il **GestoreEdificio** è appena loggato nel sistema. 2. Il **SISTEMA** mostra l’elenco dei piani dell’edificio con il loro stato.   3. Il **GestoreEdificio** può selezionare un piano.  4. Il **SISTEMA** mostra l’elenco delle stanze del piano con relativo stato.  5. Il **GestoreEdificio** seleziona la stanza che intende visualizzare, oppure torna all’elenco dei piani ( 2. ).  6. Il **SISTEMA** mostra l’elenco dei sensori relativi alla stanza selezionata con il relativo stato;  7. Il **GestoreEdificio** può usufruire di “SettaggioValori”, o torna all’elenco delle stanze ( 4. ).  8. Il **SISTEMA** chiama “SettaggioValori”. |
| *Entry condition* | * Il **GestoreEdificio** è loggato nel sistema. |
| *Exit conditions* | * Il **GestoreEdificio** ha potuto navigare tra edifici, piani e stanze, fino alla visualizzazione dei singoli sensori, controllando eventuali presenze di anomalie. |
| *Priority* | 5 (massima) |

* UC2

|  |  |
| --- | --- |
| *Use case name* | **VisualizzaEdifici** |
| *Participating actors* | **GestoreZona** |
| *Flow of events* | 1. Il **GestoreZona** è appena loggato nel sistema. 2. Il **SISTEMA** mostra l’elenco degli edifici associati alla zona di sua competenza; per ogni edificio è visibile lo stato |
| *Entry condition* | * Il **GestoreZona** è loggato nel sistema. |
| *Exit conditions* | * Il **GestoreZona** ha potuto vedere gli edifici associati alla sua zona di competenza, ognuno con il suo stato. |
| *Priority* | 3 |

* UC3

|  |  |
| --- | --- |
| *Use case name* | **VisualizzaZone** |
| *Participating actors* | **GestoreCittà** |
| *Flow of events* | 1. Il **GestoreCittà** è appena loggato nel sistema. 2. Il **SISTEMA** mostra l’elenco completo delle zone associate alla città; è visibile lo stato di ogni zona. |
| *Entry condition* | * Il **GestoreCittà** è loggato nel sistema. |
| *Exit conditions* | * Il **GestoreCittà** ha potuto vedere l’elenco delle zone della città con il relativo livello di priorità. |
| *Priority* | 3 |

* UC4

|  |  |
| --- | --- |
| *Use case name* | **SettaggioValori** |
| *Participating actors* | **GestoreEdificio** |
| *Flow of events* | 1. Il **GestoreEdificio** clicca sul bottone per il settaggio dei valori. 2. Il **SISTEMA** mostra un’interfaccia in cui è possibile modificare i valori minimali e massimali per ciascuna variabile ambientale.   3. Il **GestoreEdificio** può selezionare i campi delle variabili e immettere i nuovi valori.  4. Il **GestoreEdificio** clicca sul bottone per il salvataggio dei valori immessi.  5. Il **SISTEMA** salva i valori immessi dal **GestoreEdificio**. |
| *Entry condition* | * Il **GestoreEdificio** è loggato nel sistema. * Il **GestoreEdificio** sta monitorando una stanza. |
| *Exit conditions* | * Il **GestoreEdificio** ha modificato uno o più valori per la stanza, **OR** * Il **GestoreEdificio** non ha effettuato modifiche |
| *Priority* | 4 |

* UC5

|  |  |
| --- | --- |
| *Use case name* | **AllertaAutorità** |
| *Participating actors* | **GestoreEdificio, Autorità** |
| *Flow of events* | 1. Il **GestoreEdificio** durante il monitoraggio, rileva un pericolo e preme il bottone per inviare una segnalazione alle autorità. 2. Il **SISTEMA** mostra un’interfaccia contenente i numeri di telefono delle autorità competenti. 3. Il **GestoreEdificio** chiama le autorità necessarie. 4. L’**Autorità** chiamata si mobilita sul luogo. |
| *Entry condition* | * Il **GestoreEdificio** è loggato nel sistema. * Il **SISTEMA** ha rilevato un possibile pericolo e lo ha notificato al **GestoreEdificio** |
| *Exit conditions* | * L’**Autorità** ha preso in carico la risoluzione del problema. |
| *Priority* | 5 (massima) |

* UC6

|  |  |
| --- | --- |
| *Use case name* | **SegnalaZona** |
| *Participating actors* | **GestoreCittà, GestoreZona.** |
| *Flow of events* | 1. Il **GestoreCittà** rileva un’ anomalia o malfunzionamento in una zona. 2. Il **GestoreCittà** preme un bottone associato alla zona in questione per richiedere un feedback al **GestoreZona** responsabile. 3. Il **SISTEMA** invia una notifica al **GestoreZona**. 4. Il **GestoreZona** visualizza la notifica e preme il bottone su di essa per esprimere che ha preso in carico il problema. 5. Il **SISTEMA** notifica al **GestoreCittà** che il **GestoreZona** è consapevole del problema. 6. Il **GestoreCittà** visualizza la notifica. |
| *Entry condition* | * Il **GestoreCittà** è loggato nel sistema. * Il **GestoreZona** è loggato nel sistema. |
| *Exit conditions* | * Il **GestoreCittà** ha notificato al **GestoreZona** la presenza di un problema e si è assicurato che quet’ultimo lo sta risolvendo. |
| *Priority* | 4 |

* UC7

|  |  |
| --- | --- |
| *Use case name* | **SegnalaEdifico** |
| *Participating actors* | **GestoreZona, GestoreEdificio.** |
| *Flow of events* | 1. Il **GestoreZona** rileva un’ anomalia o malfunzionamento in un’edificio. 2. Il **GestoreZona** preme un bottone associato all’edificio in questione per richiedere un feedback al **GestoreEdificio** responsabile. 3. Il **SISTEMA** invia una notifica al **GestoreZona**. 4. Il **GestoreEdificio** visualizza la notifica e preme il bottone su di essa per esprimere che ha preso in carico il problema. 5. Il **SISTEMA** notifica al **GestoreZona** che il **GestoreEdificio** è consapevole del problema. 6. Il **GestoreZona** visualizza la notifica. |
| *Entry condition* | * Il **GestoreZona** è loggato nel sistema. * Il **GestoreEdificio** è loggato nel sistema. |
| *Exit conditions* | * Il **GestoreZona** ha notificato al **GestoreEdificio** la presenza di un problema e si è assicurato che quet’ultimo lo sta risolvendo. |
| *Priority* | 4 |

* UC8

|  |  |
| --- | --- |
| *Use case name* | **VisualizzaPericolo** |
| *Participating actors* | **GestoreEdificio** |
| *Flow of events* | 1. Il **SISTEMA** ha rilevato un’anomalia riconducibile ad un pericolo. 2. Il **SISTEMA** mostra immediatamente una notifica di pericolo ben visibile. 3. Il **GestoreEdificio** visualizza la notifica e clicca sul bottone su di essa. 4. Il **SISTEMA** mostra l’elenco delle stanze che hanno generato il pericolo. 5. Il **GestoreEdificio** a questo punto può usufruire di **AllertaAutorità** |
| *Entry condition* | * Il **GestoreEdificio** è loggato nel sistema. * Il **SISTEMA** ha rilevato un’anomalia e ha calcolato un pericolo |
| *Exit conditions* | * Il **GestoreEdificio** è stato prontamente avvisato di un pericolo con una notifica ben visibile. |
| *Priority* | 5 (massima) |

A.2 Non Functional Requirements

**Performance**

* Il sistema deve essere in grado di gestire fino a 150.000 sensori per città, quindi deve supportare la ricezione di almeno 150.000 messaggi al minuto.
* Il systema deve permettere l’accesso in parallelo di almeno 50 gestori (modifiche sincronizzate).
* La dashboard deve avere un tempo di refresh inferiore al “min t” per permettere ai gestori di avere una situazione aggiornata della propria area di competenza.
* Scalability**:** Il sistema deve essere in grado di supportare eventuali future espansioni che potrebbero aumentare i valori di Performance.

**Availability**

* Il sistema deve essere accessibile e disponibile 24/24h e 7/7gg; un mancato funzionamento dello stesso potrebbe causare gravi incidenti in quanto non si riuscirebbe a monitorare e localizzare un’anomalia o un malfunzionamento.

**Safety**

* Il sistema deve essere sempre in grado di rilevare un’anomalia o un malfunzionamento, e di calcolare la priorità in maniera precisa e puntuale, senza errori, in modo da notificare l’allerta al gestore. Il gestore edificio può risolvere autonomamente il problema (sostituire il sensore, accendere i climatizzatori ecc) oppure può contattare le autorità competenti, evitando di mettere a rischio la vita delle persone che occupano un’area.

Inoltre, come ulteriore sicurezza, tutti i diversi tipi di gestore vedono un’anomalia o un malfunzionamento nella propria area, potendo comunicare tra di loro per sapere se un problema è in risoluzione o meno.

**Security**

* Il sistema deve essere installato solo sulle macchine presenti nell’ufficio dei gestori (in modo da prevenire che persone esterne possano provare ad entrare nel sistema tramite altri pc e cambiare un valore dei sensori).
* L’accesso da parte di fonti sconosciute alle macchine è tenuto sicuro tramite un login. Se il login fallisce per 5 volte di seguito il sistema blocca l’accesso a quella macchina per un minuto mandando una notifica all’admin che verifica che il sistema sia protetto.

A.3 Excluded Requirements

* Il sistema non deve fornire un servizio per l’interrogazione dei sensori.

A.4 Assumptions

**ASSUNZIONI**

* Un gestore sarà responsabile di una sola area (città, zona, edificio).
* In una singola area vi possono essere da 10 a 200 sensori.
* I sensori di fumo inviano solo segnali di stato: 0 (assenza di fumo), 1 (presenza di fumo), ad intervalli di 1 minuto. Questi sensori vengono posti soltanto in stanze importanti (aule frequentate da molte persone, aule conferenze, aule che contengono dispositivi tecnologici ecc) per non eccedere con i costi e per avere un feedback immediato in caso di incendio.
* Intervalli di invio valori ambientati (t) :
* Temperatura: t = 60s
* Umidità: t = 90s
* Pressione: t = 120s
* Luminosità: t = 300s

Nel momento in cui viene rilevata un’anomalia, “t” si riduce in base alla priorità di quest’ultima:

* + Bassa priorità = t/2 ;
  + Media priorità = t/4 ;
  + Alta priorità = t/6 ;
* LIVELLI DI ALLERTA =
  + Verde: situazione normale
  + Giallo: bassa priorità
  + Arancione: media priorità
  + Rosso: alta priorità
  + Pericolo: situazione critica che può mettere in pericolo la vita delle persone

***A.5 Prioritization***

*Requisiti in ordine di priorità in una scala da 1 a 5:*

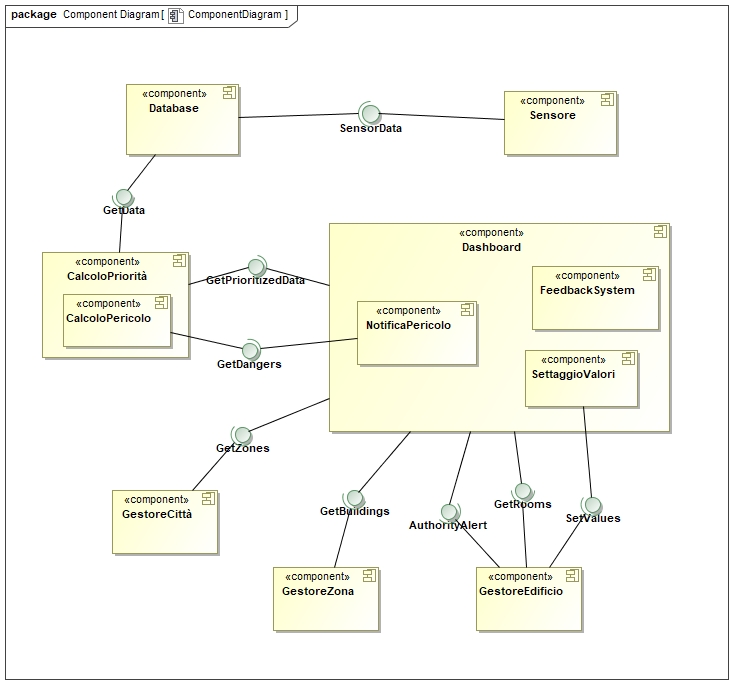
* I **sensori** devono essere in grado di inviare i dati rilevati al sistema, ad intervalli regolari. (5)
* La **dashboard** deve notificare al gestore anomalie e malfunzionamenti relativi ai sensori, fornendo dei livelli di allerta in base all’entità del problema.(5)
* Il **gestore edificio** deve essere in grado di monitorare l’edificio ad esso assegnato, visualizzando lo stato di tutti i piani e di tutte le stanze; per ogni stanza può visualizzare i singoli sensori, con le relative variabili ambientali rilevate. (5)
* La **dashboard** deve notificare al gestore eventuali situazioni di pericolo.(5)
* Il **gestore edificio** in caso di pericolo deve essere in grado di allertare le autorità competenti. (4)
* Il **gestore edificio** può modificare i MIN VALUE e MAX VALUE relativi ad una stanza. (4)
* Il **gestore città** deve essere in grado di monitorare la città ad esso assegnata, visualizzando lo stato di tutte le zone. (3)
* Il **gestore zona** deve essere in grado di monitorare la zona ad esso assegnata, visualizzando lo stato di tutti gli edifici. (3)
* La **dashboard** deve fornire una visualizzazione delle aree diversa in base al tipo di gestore in servizio. (3)
* L’**Admin** deve poter inserire o modificare un gestore, e assegnare determinate aree a determinati gestori. (3)
* L’**Admin** deve poter inserire o modificare un sensore. (3)
* Il **gestore città**  può richiedere al gestore zona un feedback relativo ad un’anomalia/malfunzionamento in una zona. (2)
* Il **gestore zona**  può richiedere al gestore edificio un feedback relativo ad un’anomalia/malfunzionamento in un edificio. (2)

B. Software Architecture

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

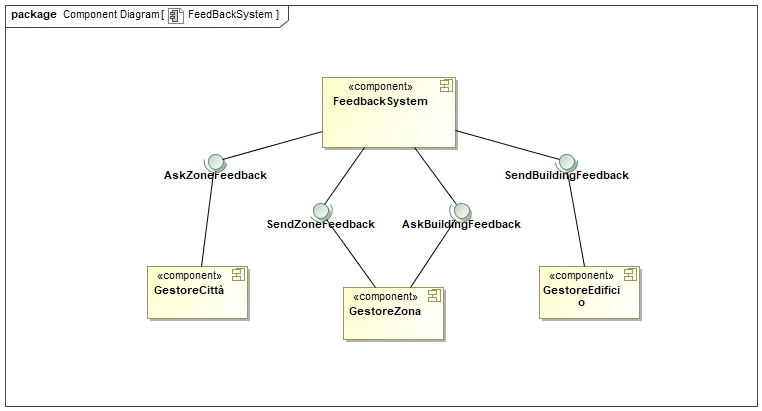
**Figure 8: CD Sistema Completo**

La parte fondamentale è il componente CalcoloPriorità, il quale, assieme a CalcoloPericolo, permettono di assegnare ad ogni area visibile nella Dashboard il giusto livello di allerta, analizzando i dati rilevati dai sensori. La Dashboard fornisce una View adatta al tipo di Gestore connesso.



**Figure 9: CD Feedback System**

Il Feedback System è un sistema di notifiche integrato nella Dashboard che consente l’interazione tra i vari Gestori; in questo modo un Gestore a un livello più alto può sollecitare un Gestore al livello inferiore affinché risolva un problema nell’area.

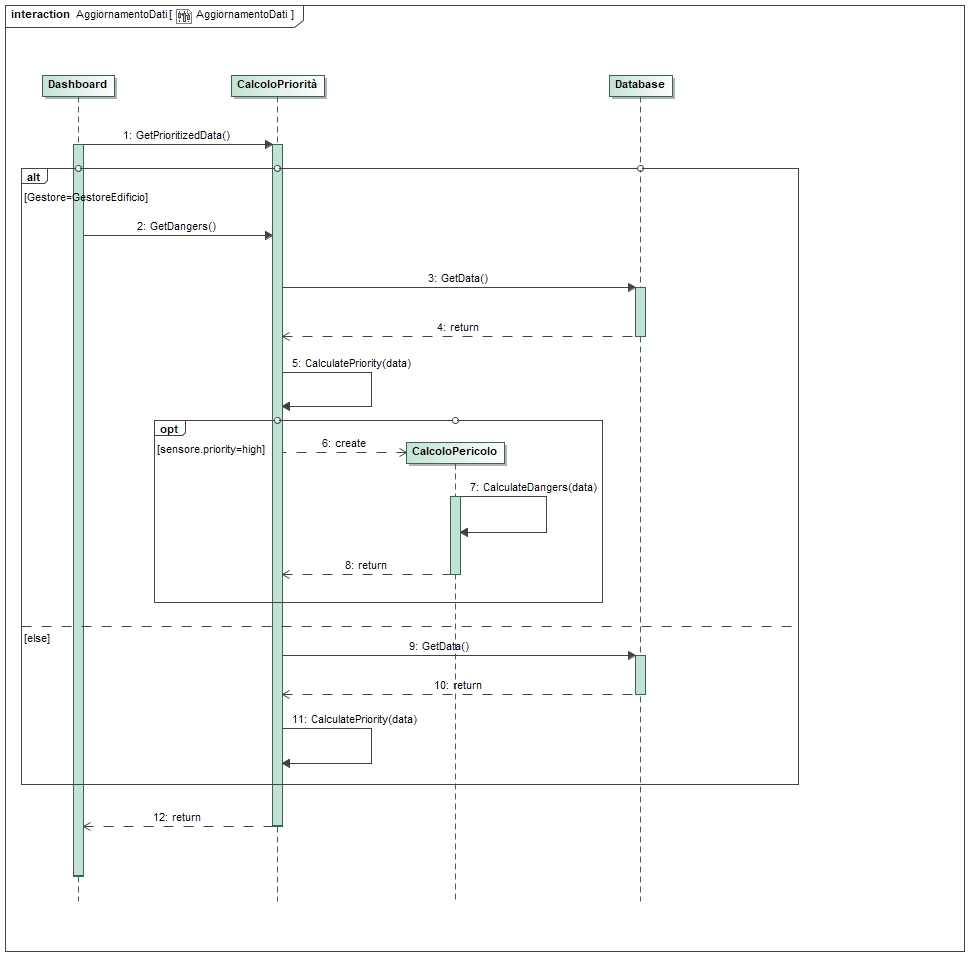


C.2 The dynamic view of the software architecture: Sequence Diagram

**Figura 10: SD Aggiornamento dati**

La Dashboard richiede un aggiornamento dei dati da visualizzare.

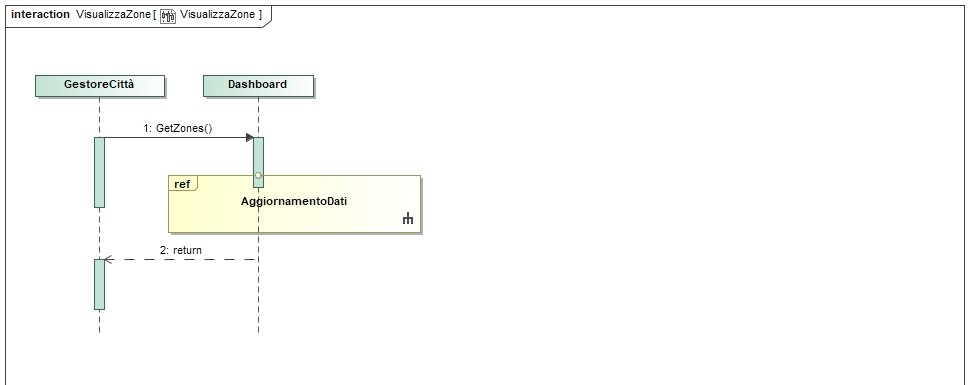
Solo per il GestoreEdificio verranno calcolati gli eventuali pericoli (calcolati solo se l’anomalia rilevata ha alta priorità).



**Figura 11: SD Visualizza Zone**

Il GestoreCittà, quando entra nel sistema, richiede implicitamente alla Dashboard la visualizzazione delle zone appartenenti alla città a lui assegnata.

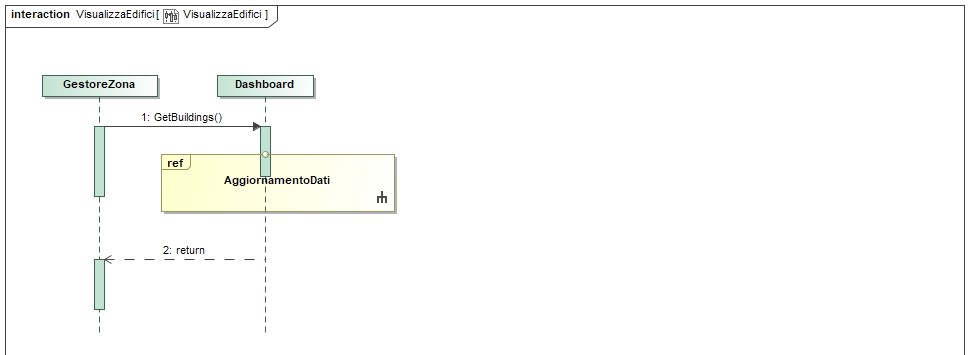
La Dashboard utilizza l’ ”AggiornamentoDati”.



**Figura 12: SD Visualizza Edifici**

Il GestoreZona, quando entra nel sistema, richiede implicitamente alla Dashboard la visualizzazione degli edifici appartenenti alla zona a lui assegnata.

La Dashboard utilizza l’ ”AggiornamentoDati”.

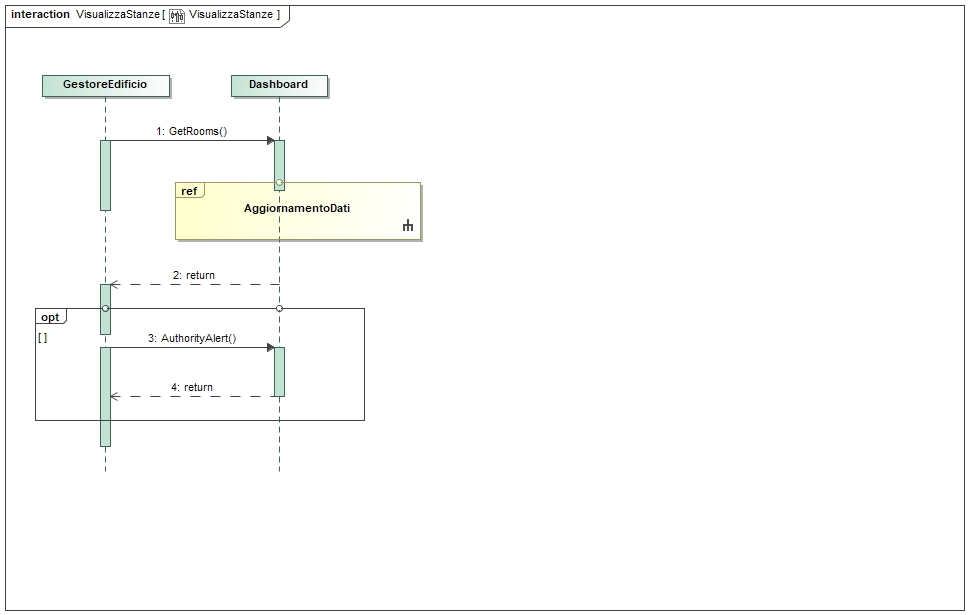


**Figura 13: SD Visualizza Stanze**

Il GestoreEdificio, quando entra nel sistema, richiede implicitamente alla Dashboard la visualizzazione delle stanze (raggruppate in piani) dell’edifcio a lui assegnato.

La Dashboard utilizza l’ ”AggiornamentoDati”.

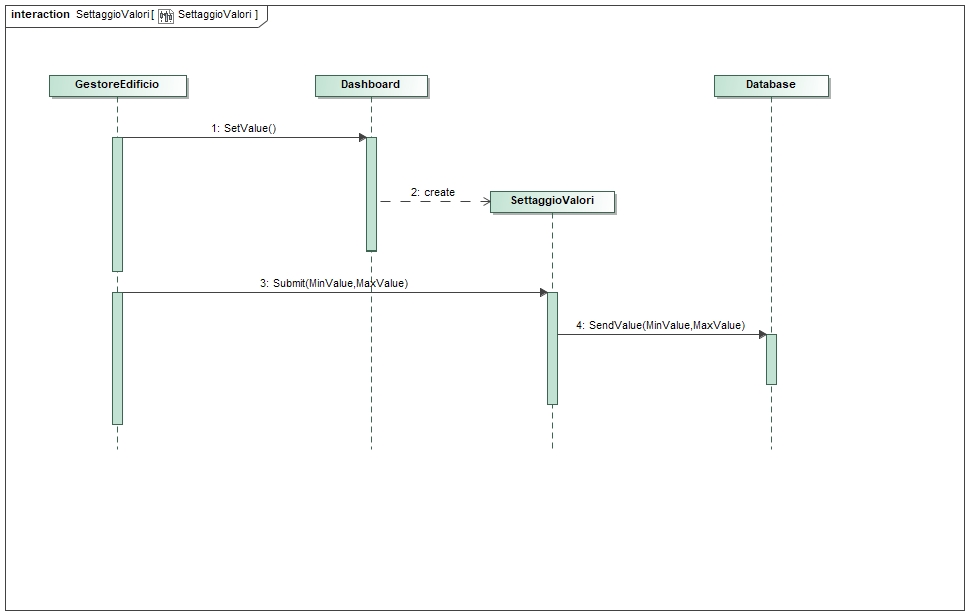
Il GestoreEdificio può contattare le autorità competenti richiedendo alla Dashboard i numeri di quest’ultime.



**Figura 14: SD Settaggio Valori**

Il GestoreEdificio può settare il MIN VALUE e il MAX VALUE per i sensori dell’edificio a lui assegnato.

Tali valori saranno poi inviati al Database tramite il component “SettaggioValori”.

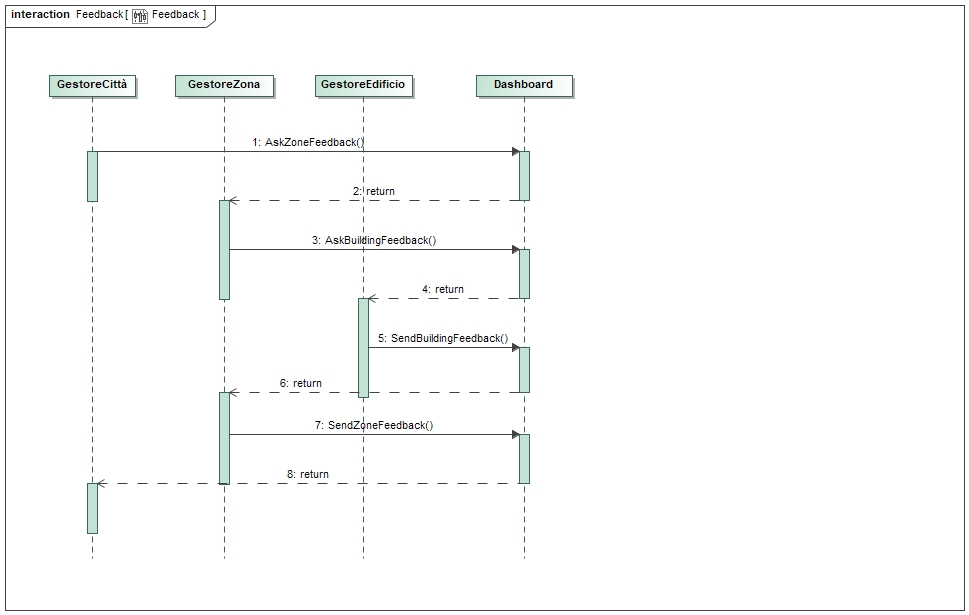


**Figura 15: SD Feedback**

Il GestoreCittà può richiedere un feedback al GestoreZona, il quale a sua volta richiede un feedback al GestoreEdificio.

Quando il GestoreEdificio risolve il problema (o i problemi) invia un feedback al GestoreZona.

Quando il GestoreZona risolve il problema (o i problemi) invia un feedback al GestoreCittà.



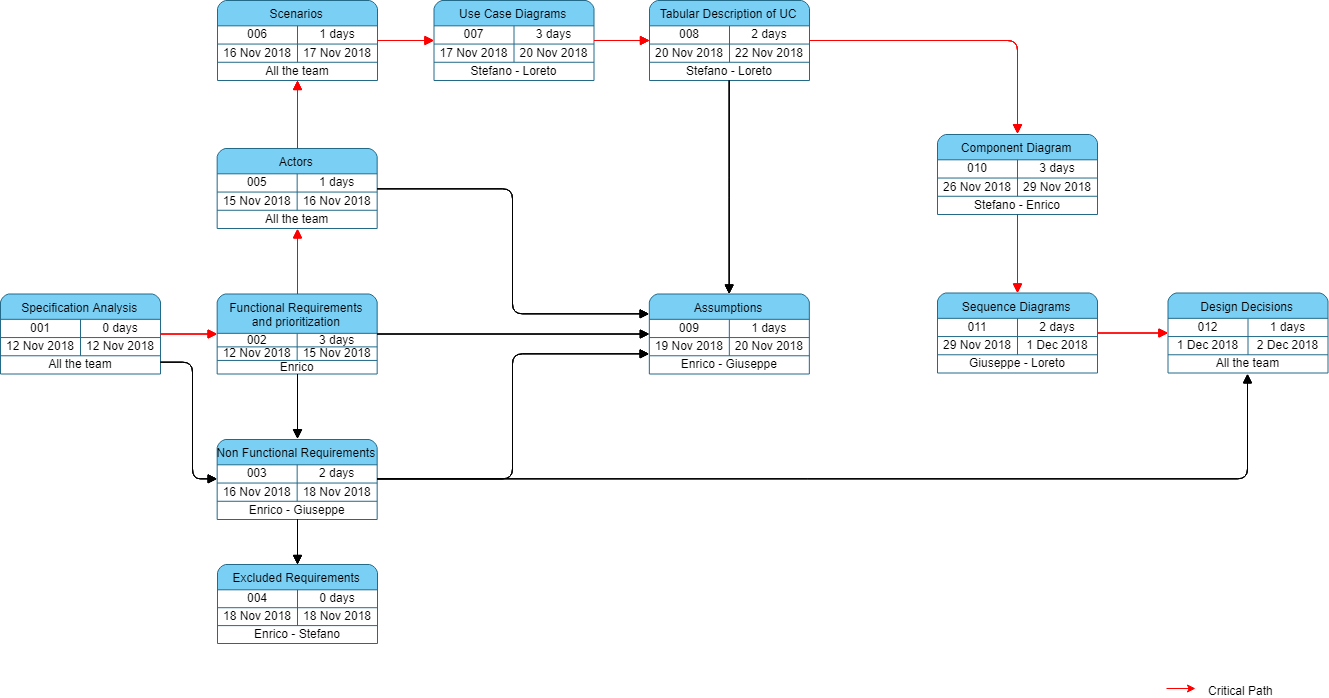
E. Design Decisions

* Per gestire la mole di dati rilevata dai sensori utilizzeremo un database NoSQL, più efficiente del canonico SQL.
* Essendo un sistema basato su interfacce, diverse a seconda del gestore attivo, utilizzeremo il pattern MVC.
* Utilizzeremo un linguaggio orientato agli oggetti per modellare al meglio le entità trovate durante lo studio.
* Per la comunicazione tra i gestori abbiamo deciso di utilizzare un sistema basato su feedback.
* La grafica della dashboard la implementeremo tramite un linguaggio web (HTML/CSS)

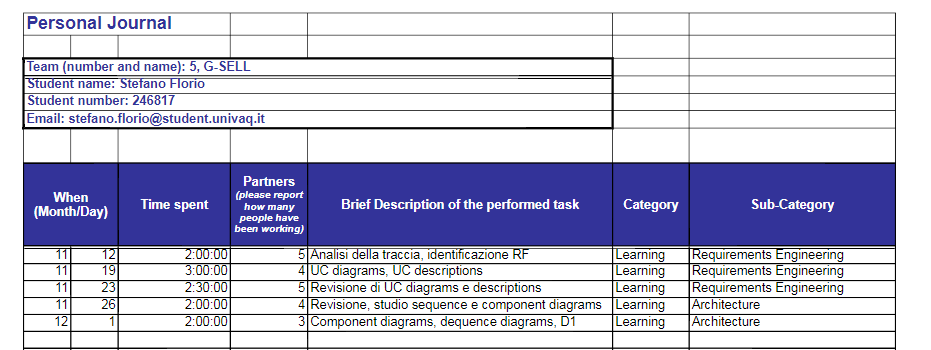
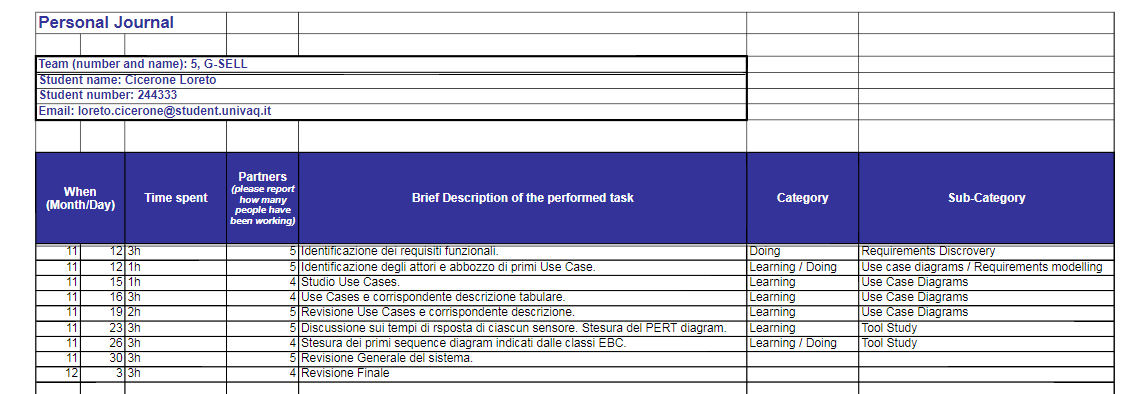
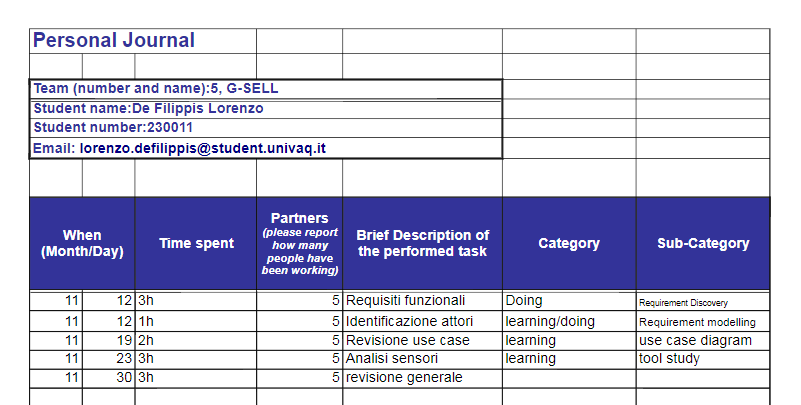
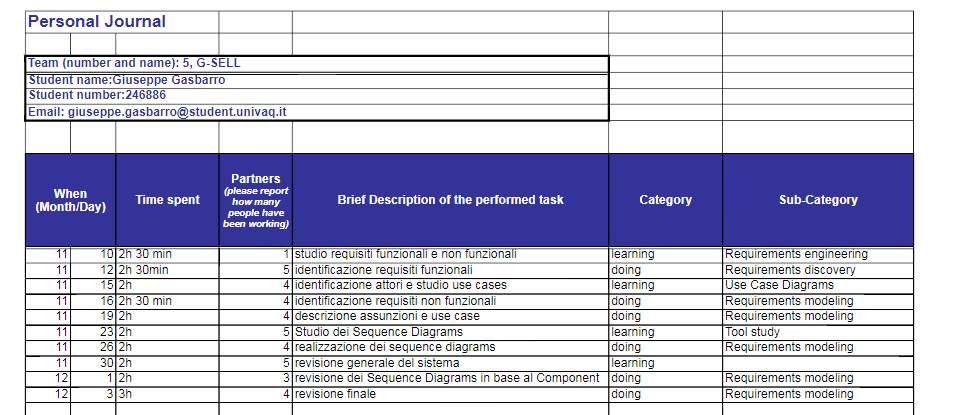
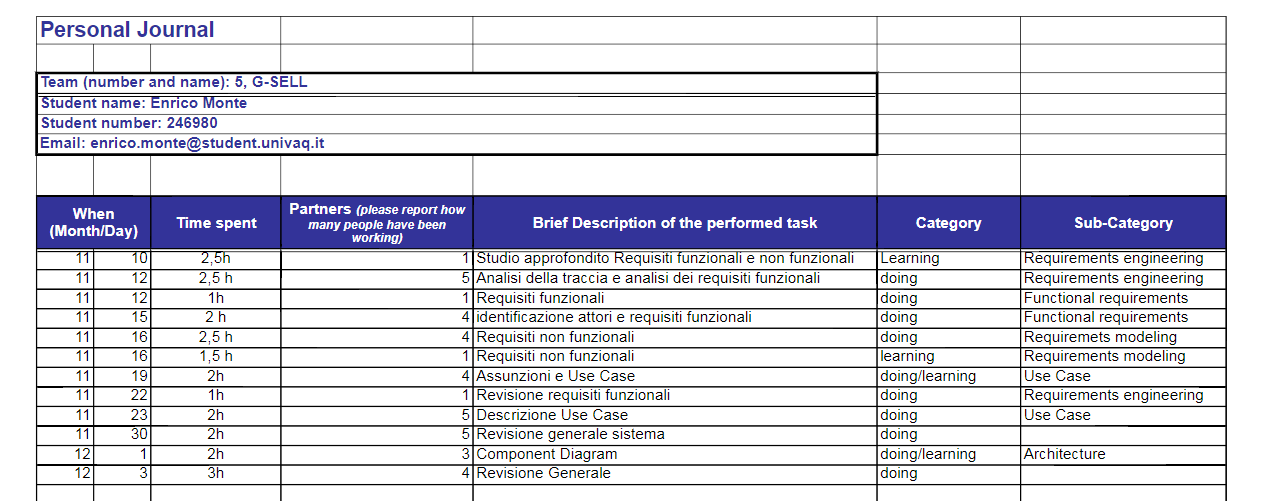
G. Effort Recording



***PERT***



***Logging***



***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.*

**Learning:7h**

**Doing:10h**

***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***

***Requisiti funzionali: 11h***

***Identificazione attori: 5h***

***Use Case: 15h***

***Requisiti non-funzionali: 6h 30min***

***Assunzioni: 2h***

***Component Diagrams: 6h***

***Sequence Diagrams: 8h***

***Design Decision: 3h***

***TIME SPENT FOR DOING: 51h***

***TIME SPENT FOR LEARNING: 33h 30min***