"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	Deliverable 2	
Team (Name)	Handy Sparke	

Team Members					
Name & Surname	Matriculation	E-mail address			
	Number				
Di Berardino Sara	247976	sara.diberardino@student.univaq.it			
Rampa Valentina	246816	valentina.rampa@student.univaq.it			
Covelli Alexandra	248488	alexandra.covelli@student.univaq.it			
Cimoroni Marco	251369	marco.cimoroni@student.univaq.it			
Fegatilli Francesco	236845	francesco.fegatilli@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

Α.	Requirements Collection	5
	A.1 Functional Requirements	5
	A1.1 Use Case Diagrams	
	A1.2 Tabular description of the most relevant use cases	
	A.2 Non Functional Requirements	10
	A.3 Excluded Requirements	11
	A.4 Assumptions	12
	A.5 Prioritization	
В.	Software Architecture	15
	C.1 The static view of the system: Component Diagram	
	C.2 The dynamic view of the software architecture: Sequence Diagram	
Ε.	Design Decisions	18
	Effort Recoding	

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

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

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
Decisioni sul Database	16/11/2018	18/12/2018	La decisione presa è quella di utilizzare un solo DBMS che ci permetta di gestire database distribuiti per quanti sono gli edifici in ogni zona. Garantendo una facile gestione del sistema avendo un unico Management System
Accesso alla dashboard	16/11/2018	18/12/2018	Discussione sull'accesso alle differenti dashboard. Sistema aperto o chiuso?!? Si è deciso che un sistema chiuso porterebbe alle seguenti considerazioni: -sistema strettamente privato al gestore, fine a se stesso per il lavoro da svolgeremaggiore stabilità della rete. (no sovraccarico, rete libera).
Component Diagram	26/11/2018	22/12/2018	Discussione sulla struttura del diagramma da rappresentare. Si è deciso di procedere più nel dettaglio dopo la correzione del primo Deliverable.
Sequence Diagram	26/11/2018	22/12/2018	Discussione sul livello di dettaglio da rappresentare nel diagramma. Arrivando alla conclusione di voler descrivere le due sequenze fondamentali del nostro sistema.

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

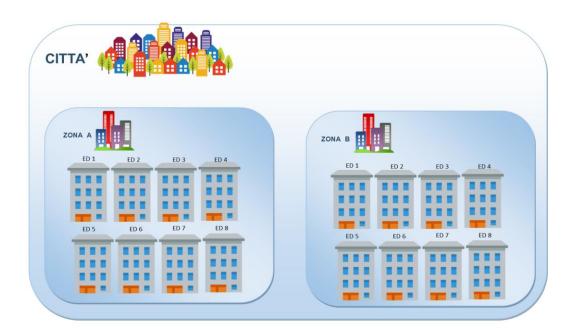
Il dominio della nostra applicazione raffigura un ambito cittadino compreso di un insieme di zone residenziali.

Ogni complesso residenziale ha al suo interno degli edifici divisi per piani e per ognuno di essi si hanno delle aree monitorate da tre diverse tipologie di sensore (temperatura, umidità e pressione).

Il numero complessivo di sensori per ogni area può variare da i 10 ai 200, ma non tutti i tipi di sensori sono presenti, ad esempio un'area può avere solo sensori di temperatura mentre un'altra può avere tutte e tre le tipologie.

I gestori (edificio/zona/citta) possono controllare lo stato dell'area a cui hanno accesso per monitorare la situazione e valutare eventuali anomalie e/o allarmi che verranno poi risolte dalla figura del manutentore.

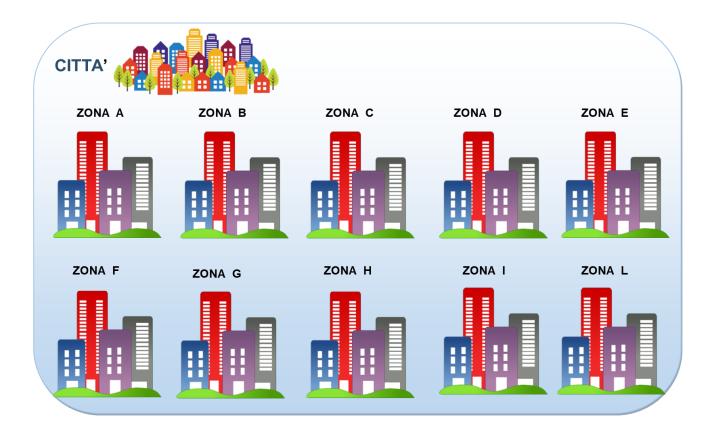
Schema di massima della macro struttura:



Dettaglio città:

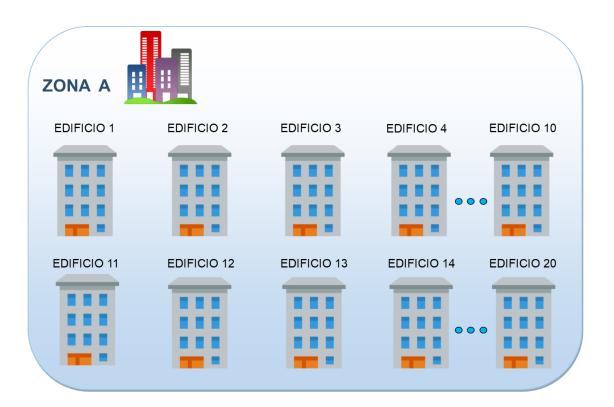
Il gestore di città, figura a più alto livello del sistema, ha una visione generale dell'ambiente cittadino preso in considerazione, che gli permette di controllare a livello macro la situazione delle zone (nel nostro caso specifico saranno 10) con la possibilità di entrare nel dettaglio fino agli edifici che le compongono.

Le immagini delle zone mostrate sull'interfaccia saranno affiancate da bottoni di colore verde/arancione/rosso necessari per identificare le informazioni sulla situazione in real-time dei sensori.



<u>Dettaglio zona:</u>

Il gestore di zona, figura intermedia responsabile di un'unica zona, ha una visione generale che gli permette di controllare la condizione dei singoli edifici all'interno della zona stessa (nel nostro caso specifico saranno 10) con la possibilità di entrare nel dettaglio sugli edifici. Le immagini degli edifici mostrati sull'interfaccia saranno affiancate da bottoni di colore verde/arancione/rosso necessari per identificare le informazioni sulla situazione in real-time dei sensori.



SE course – Deliverables 2018-2019

• <u>Dettaglio edificio:</u>

Il gestore di edificio, figura a più basso livello del sistema, ha una visione dell'edificio di cui è responsabile che gli permette di controllare a livello micro la condizione dei piani e dei sensori all'interno della struttura.

I dati puntuali di ogni sensore sono visibili solo a livello edificio. Lo storico dei singoli sensori che rilevano allarmi e/o malfunzionamenti di rottura vengono salvati su la tabella eventi per essere poi gestiti.

Stato	Data - Ora	Tipo Sensore	ID Sensore	Valore	Stanza	Edificio	Zona
	01 dic 18 - 00:00:00	SU	1000001	60	S1	E1	Z1
	01 dic 18 - 00:00:00	SP	2000001	1028	S1	E1	Z1
	01 dic 18 - 00:00:00	ST	3000001	20	S1	E1	Z1
	01 dic 18 - 00:01:00	SU	1000002	60	S1	E1	Z1
	01 dic 18 - 00:01:00	SP	2000002	1020	S1	E1	Z1
	01 dic 18 - 00:01:00	ST	3000002	18	S1	E1	Z1
	01 dic 18 - 00:02:00	SU	1000003	110	S1	E1	Z1
	01 dic 18 - 00:02:00	SP	2000003	1028	S1	E1	Z1
	01 dic 18 - 00:02:00	ST	3000003	21	S1	E1	Z1
	01 dic 18 - 00:03:00	SU	1000004	60	S2	E1	Z1
	01 dic 18 - 00:03:00	SP	2000004	1028	S2	E1	Z1
	01 dic 18 - 00:03:00	ST	3000004	20	S2	E1	Z1
	01 dic 18 - 00:04:00	SU	1000005	60	S2	E1	Z1
	01 dic 18 - 00:04:00	SP	2000005	1021	S2	E1	Z1
	01 dic 18 - 00:04:00	ST	3000005	20	S3	E1	Z1
	01 dic 18 - 00:05:00	SU	1000006	60	S3	E2	Z1
	01 dic 18 - 00:05:00	SP	2000006	1028	S3	E2	Z1
	01 dic 18 - 00:05:00	ST	3000006	21	S3	E2	Z1
	01 dic 18 - 00:06:00	SU	1000007	60	S3	E2	Z1
	01 dic 18 - 00:06:00	SP	2000007	1028	S3	E2	Z1
	01 dic 18 - 00:06:00	ST	3000007	26	S3	E2	Z1
	01 dic 18 - 00:07:00	SU	1000008	90	S3	E2	Z1
	01 dic 18 - 00:07:00	SP	2000008	1035	S3	E2	Z1
	01 dic 18 - 00:07:00	ST	3000008	21	S3	E2	Z1
	01 dic 18 - 00:08:00	SU	1000009	60	S3	E2	Z1
	01 dic 18 - 00:08:00	SP	2000009	1028	S3	E2	Z1
	01 dic 18 - 00:08:00	ST	3000009	21	S3	E2	Z1
	01 dic 18 - 00:09:00	SU	1000010	60	S3	E2	Z1
	01 dic 18 - 00:09:00	SP	2000010	1019	S3	E2	Z1
	01 dic 18 - 00:09:00	ST	3000010	NA	S3	E2	Z1

Dettagli sui sensori:

Il numero di sensori per ogni edificio è 1500, abbiamo 100 edifici per un totale di 150k sensori.

Gli edifici sono divisi in zone considerando tra i 10 ed i 20 edifici a zona abbiamo 5/10 zone per città.

Ogni edificio è di 5 piani, 300 sensori a piano, per ogni piano abbiamo dalle 3 alle 5 aree monitorate.

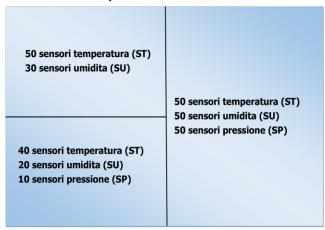
Leggenda:

ST = Sensori Temperatura.

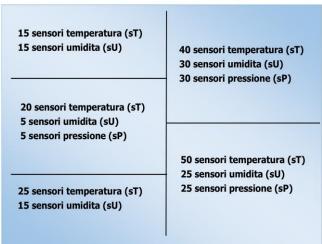
SU = Sensori Umidità.

SP = Sensori Pressione.

Schema di un piano a tre stanze con 300 sensori



Schema con piano con 5 stanze e 300 sensori



A.1 Functional Requirements

functional requirements your system shall implement> in bullet points, all the <List. <Describe functional requirements and stakeholders through Use Case Diagrams>. <Then, prioritize</p> 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>

* **REQUISITI FUNZIONALI:**

1. Lettura del dato:

La lettura del dato è un operazione del sistema che consiste nell'acquisizione delle informazioni dal database, esso viene analizzato e successivamente inviato alla dashboard.

2. Gestione malfunzionamenti:

La gestione dei malfunzionamenti viene operata dal sistema guando si ha un evento anomalo quale il dato fuori soglia (gestione allarme) o il dato non inviato (o rottura) dal sensore (gestione errore).

Questi eventi vengono gestiti tramite due procedimenti differenti per livelli di priorità e per livelli di urgenza.

Entrambe le situazioni vengono prese in carico per la risoluzione dalla figura del manutentore.

3. Gestione utenti:

La gestione degli utenti comprende:

- > il ruolo del super-amministratore che può effettuare modifica, cancellazione e inserimento degli utenti (gestori) quali Edificio, Zona e Città.
- > ogni utente una volta validato dal Sistema sarà indirizzato alla dashboard contenente i permessi relativi al proprio livello di astrazione.

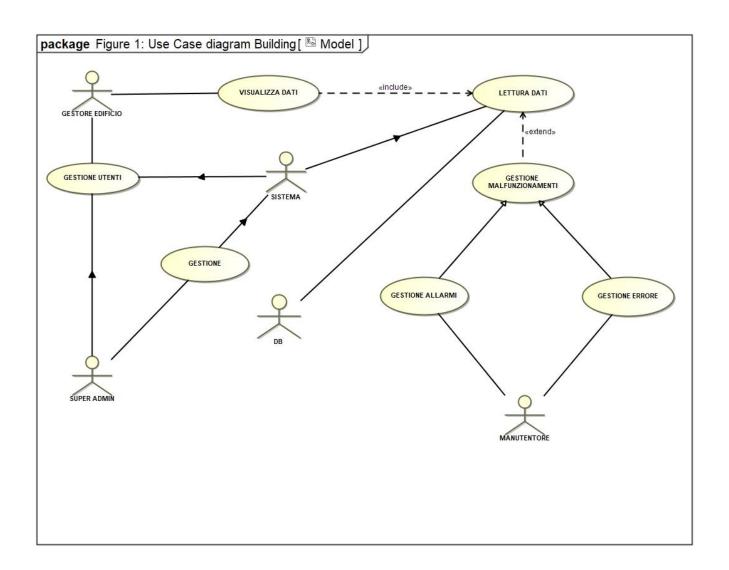
Use case:

> Figura 1. Use Case Diagram Building.

L'Attore individuato è il gestore di edificio, ottenuta la validazione da parte del sistema e quindi i suoi relativi permessi per l'area da amministrare, visualizzerà la dashboard che mostrerà le informazioni derivate dalla lettura dei dati inviati dai sensori che sono salvati nel Database.

Nel caso di malfunzionamenti, l'incarico di risolvere il problema viene assegnato al manutentore.

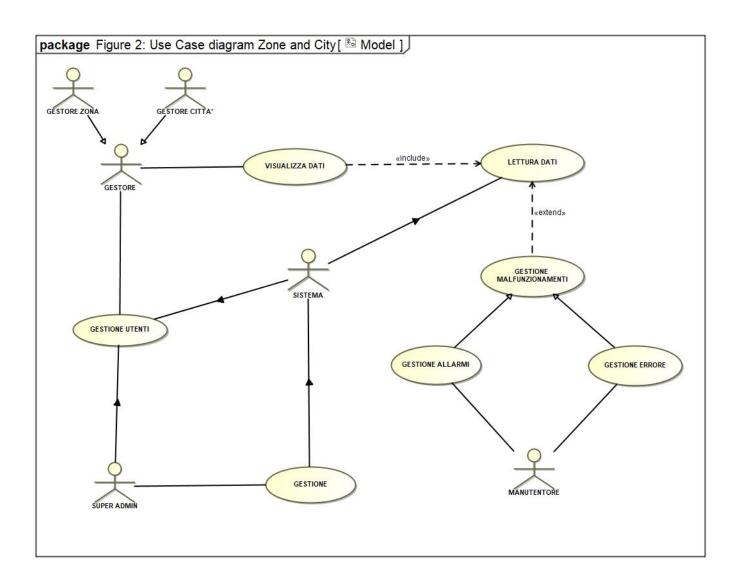
Inoltre abbiamo individuato un super amministratore che gestisce l'intero sistema e gli utenti.



> Figura 2. Use Case Diagram Zone and City

Si ha una generalizzazione dei due gestori, Zona e Città, una volta validati dal Sistema possono visualizzare le informazioni relative alla loro area competente.

Per visionare i dati di ogni sensore bisogna accedere più in profondità fino al livello di dettaglio del gestore di Edificio.



* Alistair Cockburn Use Case Template:

USE CASE #1	Visualizzazione Dati			
Goal in Context	Visualizzare i dati con il giusto livello di dettaglio le informazioni relative al gestore specifico.			
Scope & Level	Mostrare i valori riportati dai s	ensori.		
Preconditions	Il gestore deve essere stato preventivamente riconosciuto dal sistema. È disponibile un'interfaccia del sistema comprendente la zona che deve monitorare.			
Success End Condition	Il Sistema ha saputo gestire i vari segnali avvisando il gestore.			
Failed End Condition	Il sistema non è stato in grado di far visualizzare i segnali ai gestori.			
Primary	Gestore (Edificio, Zona, Città)			
Trigger	Attraverso il Login gli attori ac	cedono alla visualizzazione dei dati.		
DESCRIPTION	Step	Action		
	1	Gestore esegue il login		
	2	Il Sistema riconosce il ruolo del gestore		
	3	Il Sistema segnala al gestore lo stato dei segnali relativo alla sua area.		

RELATED INFORMATION	Visualizzazione Dati
Priority:	Alta.
Performance	40% del tempo totale dello sviluppo.
Frequency	Sempre.
Channels to actors	Interattivo.
OPEN ISSUES	Collegamenti contemporanei di più gestori.
Due Date	20/01/′19

USE CASE #2	Gestione Malfunzionamenti		
Goal in Context	Selezionare un'area e mostrare i valori che sono fuori soglia e segnalare i sensori non funzionanti.		
Scope & Level	Gestio	one degli allarmi e gestione degli errori.	
Preconditions	Sister	na permette al gestore di loggarsi e mostra i dati.	
Success End Condition	Il Sist	ema è stato in grado di riconoscere il segnale e di gestirlo.	
Failed End Condition	Il Sist	ema non è stato in grado di gestire il tipo di segnale.	
Primary, Secondary Actors	Manutentore Gestori (edificio, zona, città)		
Trigger	Attraverso la lettura dei segnali, il Sistema gestisce eventuali malfunzionamenti		
DESCRIPTION	Step	Action.	
	1 Gestore entra nel sistema.		
	2 Sistema: mostra i dati relativi alla zona associata al gestore.		
	3 Il Sistema gestisce i malfunzionamenti		

RELATED INFORMATION	Gestione Malfunzionamenti			
Priority:	Alta.			
Performance	40% del tempo totale dello sviluppo.			
Frequency	Quando necessario.			
Channels to actors	Interactive, Database.			
OPEN ISSUES	Riconoscimento allarme.			
Due Date	20/01/′19			

SE course – Deliverables 2018-2019

USE CASE #	Gestione Utenti		
Goal in Context	Gestione degli utenti ai vari livelli di astrazione.		
Scope & Level	Monitorare dettagliatamente le informazioni relative al livello dei gestori.		
Preconditions	Si ha u	ın Sistema in grado di gestire gli utenti.	
Success End Condition	Ha successo quando il Sistema è in grado di riconoscere i gestori che gli accedono.		
Failed End Condition	Quando il Sistema non gestisce in maniera corretta gli accessi.		
Primary,	Sistema controllo accessi		
Secondary Actors	Gestori (Edificio, zona, città)		
Trigger	Quando il super Admin accede alla modifica/cancellazione/inserimento degli utenti. Login utenti.		
DESCRIPTION	Step Azione		
	1	Super admin/gestore accede al Sistema.	
	2	Sistema fornisce permessi.	
	3	Acquisiti i permessi ogni utente agisce in base al proprio livello di accesso.	

RELATED INFORMATION	Gestione Utenti		
Priority:	Media.		
Performance	20% del tempo totale dello sviluppo.		
Frequency	Una volta per sessione.		
Channels to actors	Statico.		
OPEN ISSUES	In attesa che il Sistema permetta la gestione		
Due Date	20/01/′19		

USE CASE #4	Lettura Dei Dati			
Goal in Context		Gestire i dati in arrivo dal database e successivamente inviarli alla dashboard.		
Scope & Level	Lettura de o errore.	Lettura del dato per la gestione di possibili eventi di allarme o errore.		
Preconditions	Invio dei	dati dal DB, connessione con la dashboard.		
Success End Condition	Il Sistema	ha saputo gestire i dati inviatigli.		
Failed End Condition	Mancata connessione al db o errata lettura del dato.			
Primary,	Il gestore che effettua il login.			
Secondary Actors	Il Sistema che gestisce la gestione del dato.			
Trigger	Login utente e richiesta delle informazioni dell'utente loggato al sistema			
DESCRIPTION	Step Action			
	1	Login		
	2	Lettura del dato		
	3	Gestione dati in allarme		

RELATED INFORMATION	Lettura Dati
Priority:	Alta
Performance	40% del temp totale dello sviluppo
Frequency	Sempre
Channels to actors	Interattivo
OPEN ISSUES	Collegamenti contemporanei di più getsori
Due Date	20/01/'19

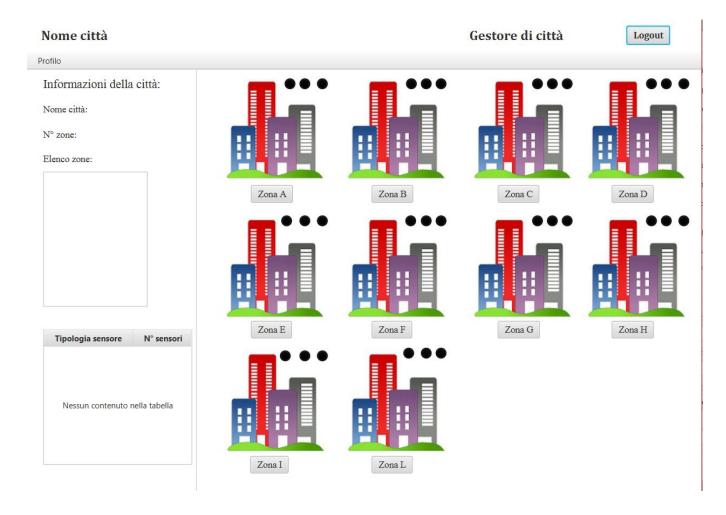
A1.1 GUI Requirements (da riempire a partire dalla Versione 2)

Interfaccia Gestore Città:

L'interfaccia del gestore di città mostra il proprio profilo e tutte le informazioni riguardanti la sua area da amministrare comprendente di zone.

Attraverso i tre bottoncini neri monitora lo stato in real-time delle zone, che si accendereanno di colore verde, arancione e rosso in base alla situazione.

Per vedere più in dettaglio ciò che compone la zona potrà cliccare sopra il suo nome.



Interfaccia Gestore Zona:

L'interfaccia del gestore di zona mostra il proprio profilo e tutte le informazioni riguardanti la sua area da amministrare comprendente di edifici.

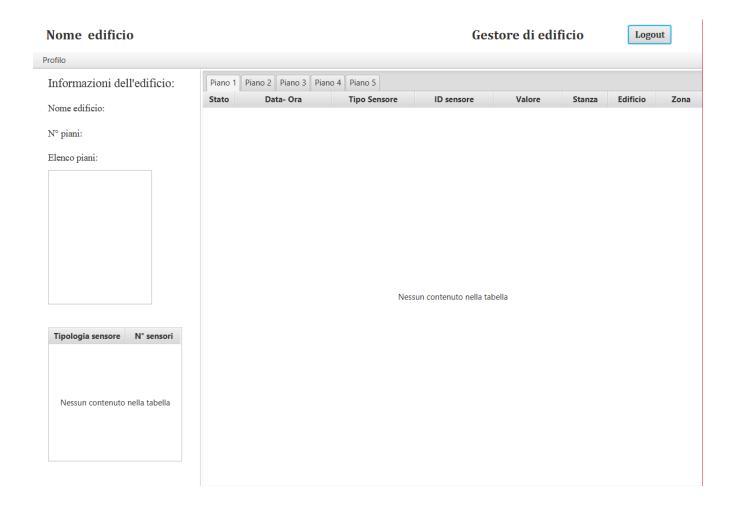
Per vedere più in dettaglio ciò che compone un edificio in sé potrà cliccare sopra il suo nome.



Interfaccia Gestore Edificio.

L'interfaccia del gestore d'edificio mostra il proprio profilo e tutte le informazioni riguardanti la sua area.

Per vedere più in dettaglio i piani dell'edificio potrà cliccare sopra il suo nome.



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

< Riportare qui i requisiti specifici della business logic>

Nella Business Logic, in base al ruolo che ricopre un determinto gestore si potranno svolgere le sequenti funzionalità:

- Leggere ogni 1 minuti le variabili ambientali monitorate dai sensori nell'area pertinente al gestore attraverso l'ID dello stesso.
- In caso di errori, attraverso l' ID del sensore, i gestori potranno individuare quale di esso è in errore (es: rottura di un sensore) e in base alla priorità assegnatagli agire di conseguenza (es: inviando il manutentore).
- In caso di malfunzionamenti, invece, i gestori attraverso l' ID del sensore potranno individuare quale di esso sta segnalando malfunzionamenti (es: un sensore di una determinata area avrà superato il range di soglia precedentemente decisa in base alla tipologia dell'area monitorata).

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

<Pleae report here DB-specific Requirements>

La decisione presa è quella di utilizzare un database distribuito che viene controllato da un Database management System (DBMS) dove gli archivi dati verranno memorizzati su più nodi. In altri termini il database in senso fisico verrà dislocato su più punti connessi tra loro sotto forma di sistema distribuito.

Formalmente si ha uno schema logico globale comune a tutti i nodi ed un insieme di schemi logici locali che dipendono funzionalmente da quello globale.

La frammentazione sarà di tipo orizzontale che possiede le seguenti proprietà:

- lo schema resta uguale in ogni frammento;
- ogni frammento contiene un sottoinsieme dei record di R;
- normalmente un frammento è definito da una selezione.

Nel nostro caso di studio abbiamo stabilito che i nodi si riferiscono agli edifici, quindi ognuno di essi avrà il suo archivio personale.

In questo modo la tabella dati, che memorizza i dati che arrivano dai sensori, riceverà tra 90.000 e i 115.000 segnali l'ora, inoltre consideriamo una percentuale di allarmi massima del 25% con un invio di informazioni ogni 30 secondi.

La tabella dati avrà uno storico delle ultime 2 ore.

A.2 Non Functional Requirements

La nostra attenzione va sui **Product requirements**

- Usability
- Efficency
 - Performance
 - Space
- Dependablity
 - Avalilability
 - Reliability
 - Safety
 - Security
 - Maintainability
 - Survivability
 - Fault tolerance
 - Repairability

Usability: nel progetto viene richiesta una semplicità d'uso e visibilità in considerazione del tipo di utenti che andranno ad usare il sistema.

Il sistema sarà accessibile da diversi gestori, i quali dovranno avere una user experience diversa, in base al tipo di gestore: di edificio, di zona o di città.

Esempio di specifiche della GUI: l'interfaccia grafica avrà dei menù con al massimo due livelli per semplificare l'usabilità e la navigazione.

Le informazioni avranno solo tre tipi di classificazione e verranno evidenziate per colore facilmente distinguibili (rosso \rightarrow allarme, arancione \rightarrow warning, verde \rightarrow OK).

Efficency: Da specifiche di progetto sono richieste soluzioni che garantiscono buone performance, ma con particolare attenzione ai costi implementativi. Si utilizzeranno delle soluzioni Hw / Sw che garantiscono tempi buoni di risposta su più accessi in parallelo (vanno garantiti almeno 50 accessi), ponendo particolare attenzione alla infrastruttura di rete e soprattutto alla struttura dei DB e relative viste/query, per rendere il più efficiente possibile la storicizzazione, la lettura, la scrittura e la sincronizzazione dei dati presenti.

Maintainability: come da richiesta di progetto, il sistema deve essere scalabile, quindi la struttura Hw / Sw è pensata per essere facilmente replicata su richieste di espansioni future garantendo una facile adattabilità a richieste di nuovi requisiti.

Fault tolerance: Essendo un sistema distribuito, verranno valutate delle specifiche ridondanze per garantire una adequata fault tolerance sulla base delle priorità fornite dal cliente su alcuni specifici ambienti / aree.

Repairability: Non ci sono richieste specifiche, verrà prevista una policy di backup (da decidere se su tutte le strutture o solo su le più grandi) in accordo con il cliente, facendo sempre riferimento ai costi.

A.3 Excluded Requirements

Tra i requisiti non funzionali escludiamo tutti i requisiti relativi alle seguenti macro categorie:

- Organizational requirements
- External requirements

I primi (**Organizational requirements**) che sono relativi a policy e procedure aziendali non vengono considerati in quanto non abbiamo informazioni in merito, né ci sono specifiche nel progetto che possono essere considerate.

Lo stesso per i secondi (External requirements), che sono al di fuori del progetto in esame (Interoperability requirements, Legislative requirements, etc),

(Es "External requirements": non ci atteniamo ai vincoli di legge imposti sulla temperatura di un edificio).

Sono stati esclusi inoltre i seguenti NFR:

- Safety: N/A, per assunzione non è un sistema che gestisce allarmi specifici relativi alla sicurezza delle persone (non è un sistema antincendio)
- Security: N/A il tipo di dati trattati non implicano delle misure di sicurezza particolari.

A.4 Assumption

Assumiamo che i sensori da monitorare siano già posizionati nelle varie "aree" prese in considerazione dal nostro sistema, questo implica che conosciamo la collocazione di ognuno di essi.

Tutti i sensori sono collegati tra di loro attraverso dei cavi (no wireless).

In caso di normale funzionamento, essi inviano ogni minuto le informazioni ambientali che misurano (es. temperatura, pressione, umidità) al sistema.

Se dopo 10 minuti il sistema non riceve segnali da un determinato sensore viene generato un evento di errore (eccetto per quelle aree che hanno priorità maggiore dove l'allarme viene inviato al primo fallimento di ricezione del segnale).

Le aree monitorate, che corrispondono a zone specifiche di un dato edificio (ad esempio aule, computer room, celle frigorifere, corridoi, ecc) devono rispettare dei parametri che ci vengono forniti espressamente dal cliente con relative priorità. Se i valori recepiti dai sensori non sono tali, verrà inviato un segnale di allarme (di qualsiasi tipologia), che resta attivo finché il problema verrà preso in carico dal manutentore.

Il normale monitoraggio dell'area riprenderà nel momento in cui la problematica sarà risolta.

Definire un generico evento come qualsiasi variazione subita dai sensori rilevante ai fini della gestione. Una suddivisione potrebbe essere:

- Evento di allarme (deve essere gestito dal manutentore)
- Evento di avviso (warning arancione se il problema rientra da solo viene solo segnalato).
- Evento di errore (deve essere gestito dal manutentore)

> <u>Gestione allarmi</u>

Durante un malfunzionamento del sensore, inteso come recezione sbagliata delle informazioni ambientali, descriviamo due livelli di priorità:

- Arancione: Se l'area monitorata presenta almeno un valore fuori soglia tale informazione verrà riportata in arancione.
- Rosso: Se l'area monitorata presenta più valori fuori soglia tali informazioni verranno riportate in rosso (in caso di area con priorità alta il warning sarà sempre rosso).

Gestione errori

In caso di rottura di un sensore, descriviamo due livelli di urgenza:

- Priorità bassa: L'area monitorata dispone di più sensori dello stesso tipo, nel caso in cui vi fosse la rottura di uno di essi, la priorità è settata su 'bassa' poiché tramite i sensori ancora funzionanti presenti possiamo comunque recepire le informazioni ambientali che riquardano l'area stessa.
- Priorità alta: L'area monitorata dispone di un unico sensore per cui, in caso di rottura, la priorità è settata su 'alta' poiché non possiamo più recepire le informazioni ambientali che la riguardano.

A.5 Prioritization

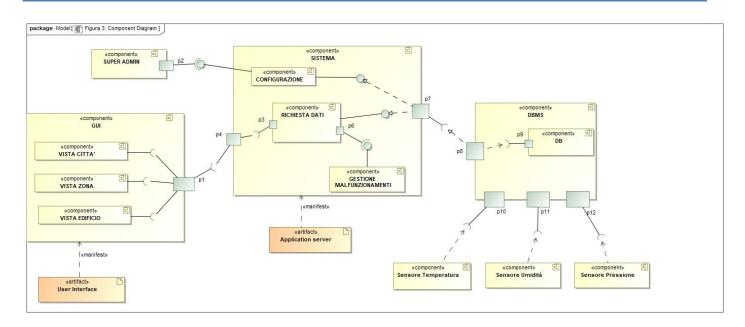
<List here all the requirements, in prioritized order>

Ordine	Requisito	Priorità	Motivazione
1	Lettura del dato	Alta	Funzione principale del sistema, permette la visualizzazione delle variabili ambientali.
2	Gestione malfunzionamenti	Alta	Funzione indispensabile del sistema; in quanto: in caso di malfunzionamento ed in base alla priorità dello stesso si potrebbero verificare conseguenze pericolose.
3	Gestione Utenti	Bassa	Permette di fornire i permessi agli utenti.

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



Descrizione Component diagram:

Nel nostro Component Diagram abbiamo individuato i seguenti macro-componenti:

- Grafica User Interface (GUI): E' composta da una schermata di dashbord principale che sarà diversa a seconda del ruolo ricoperto da chi effettua il login, nel nostro caso I gestori di edificio, zona e città.
- Super Admin è l'amministratore dei gestori del nostro Sistema.
- Sistema:

Che consiste in tre sotto-componenti:

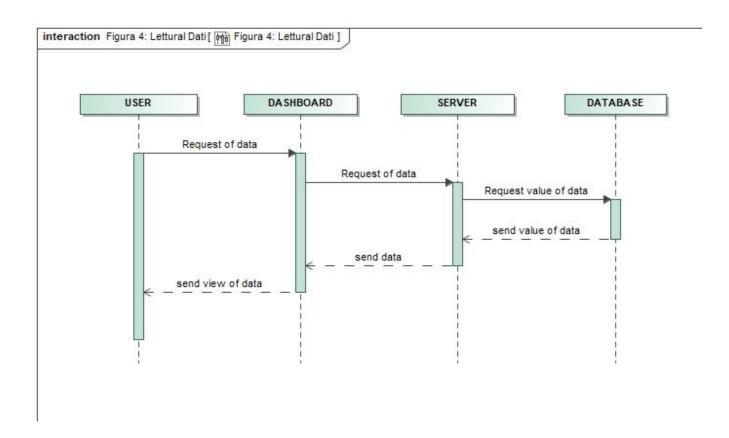
- o Configurazione: Attraverso di essa il super admin può creare, modificare e cancellare gli utenti e tutti gli altri oggetti del Sistema.
- Richiesta dati: Il sistema richiede i dati al DBMS.
- o Gestione malfunzionamenti: Attraverso di esso andremo a gestire i valori fuori soglia e i valori in errore.

- DBMS: E' il nostro gestore del DB distribuito per i vari edifici
 - o DB: E' l'archivio dei dati.
- Sensori Temperatura.
- Sensori Umidità.
- Sensori Pressione.
- Dalla dashboard principale può mostrare i tre livelli di astrazione dipendenti dall'utente che si college.

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

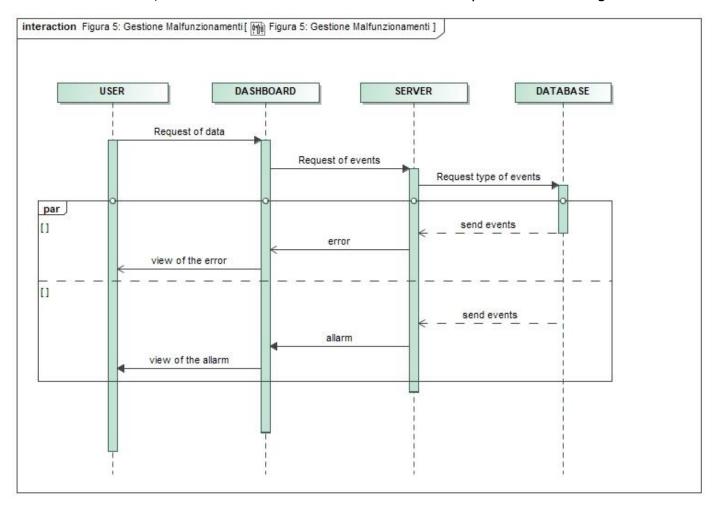
❖ Figura 4: LETTURA DATI

- L' User attraverso la dashboard richiede di visualizzare i dati.
- La Dashboard chiede al server le informazioni riguardo i segnali per la loro lettura che verranno forniti a seconda dei permessi dell'utente logato.
- Il server richiede al Database il valore dei segnali, una volta ricevuti li invia alla dashboard che li mostra all'User.



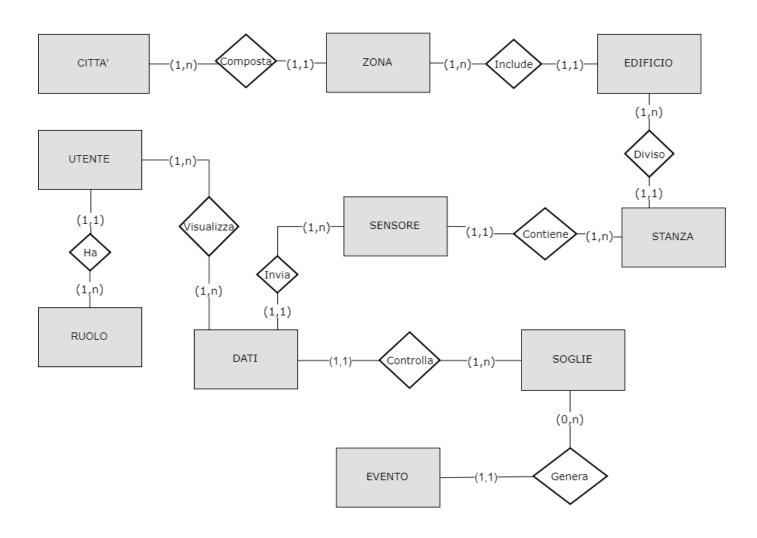
❖ Figura 5: GESTIONE MALFUNZIONAMENTI

- L' User attraverso la dashboard richiede di visualizzare i dati dell'edificio che ha malfunzionamenti.
- La Dashboard chiede al server le informazioni riguardo gli eventi.
- Il server richiede al DB il tipo di evento, se quest'ultimo è:
 - o Un errore, la Dashboard mostra all'utente i sensori in errore.
 - o Un allarme, la dashboard mostra all' utente le aree con parametri fuori soglia.

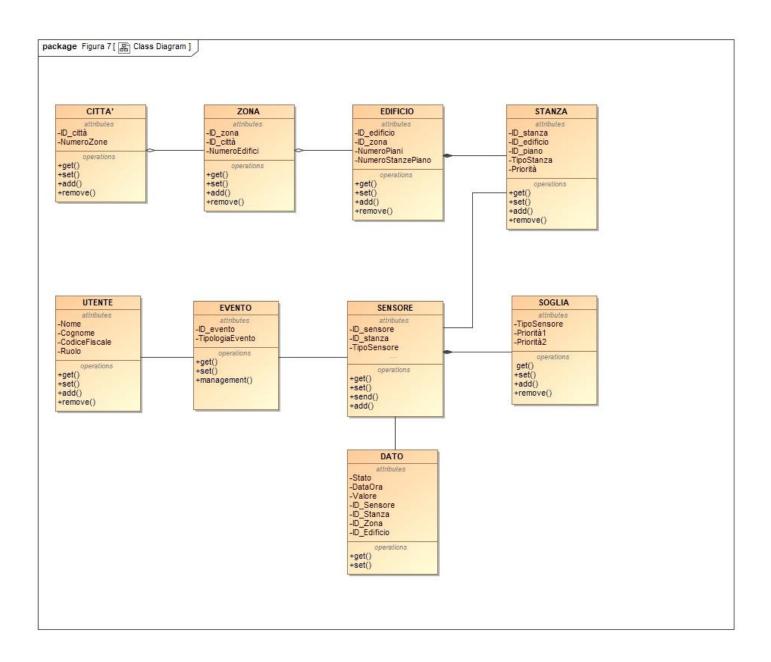


C. ER Design

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



D. Class Diagram of the implemented System



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.>

Per evitare una complessità eccessiva con il rischio di non centrare l'obiettivo del progetto e poter definire al meglio le Design Decisions, abbiamo contestualizzato il nostro progetto.

Il contesto da noi immaginato è il seguente: una serie di aree residenziale al cui interno possiamo avere da cinque a dieci edifici. Ogni edificio avrà 1500 sensori distribuiti nei vari piani, in ognuno dei quali, i diversi ambienti possono contenere, come da specifiche, dai 10 ai 200 sensori.

Dato per assunto che tutti i sensori sono cablati e collegati, i segnali verranno inviati da ogni edificio su DB in cloud opportunamente dimensionati.

Negli edifici con particolari esigenze dettate dal cliente, verrà previsto un db server locale di backup, che in caso di problemi di connessione verso il cloud, memorizzerà i dati dei sensori che altrimenti andrebbero persi.

I dati degli edifici, memorizzati in Real time, verranno analizzati dalla dashboard di più basso livello che avrà anche le funzionalità di gestione degli eventi anomali (errore/allarmi).

I dati aggregati degli edifici saranno visualizzati dalle dashboard di zona e di città rispettivamente.

L'invio dei dati di tutti i sensori avviene ogni minuto.

Sensori/edificio	1500
Edifici	100
Aree min	10
Aree Max	20
Segnali per edificio	
Numero segnali contemporanei Max	1500/tipo di sensori
Numero segnali ora min	1.500*12=18.000
Numero segnali ora MAX (stima 20% MAX eventi allarme)	300*60+1200*12=32400

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>

- Partendo dai requisiti funzionali descriviamo:
 - Lettura del dato, esso viene soddisfatto attraverso l'implementazione prendendo i dati dal db. Riguarda la parte dinamica del sistema infatti bisogna fare riferimento al sequenze diagram (figura 4).
 - Gestione dei malfunzionamenti, viene soddisfatta attraverso l'implementazione facendo opportuni controlli e generando eventi. Riguarda la parte dinamica del sistema infatti bisogna far riferimento al sequenze diagram (figura 5).
 - Gestione degli utenti, viene soddisfatta attraverso l'implementazione. Il gestore effettua il login con il quale ottiene del permessi per amministrare la proprio area.
- Passando ai requisiti non funzionali abbiamo:
 - Usability, viene soddisfatta attraverso le interfacce grafiche descritte con disegni e colori in base sia alle varie situazioni che si presentano sia al livello di gestione in cui ci si trova.
 - Fault Tolerance, viene soddisfatta attraverso l'inserimento di un db per ogni edificio in modo tale che in caso di guasti esso entra in una fase di urgenza invece che tutto il sistema.
 - Maintainability, viene soddisfatta attraverso aggiunte sia a livello architetturale, come per esempio un ulteriore db per un nuovo edificio, sia a livello implementativo, (software) inserendo i dati nel db; per esempio fare un insert di una nuova zona oppure inserire una nuova tipologia di sensore.

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 find useful. vou

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.

SE course – Deliverables 2018-2019

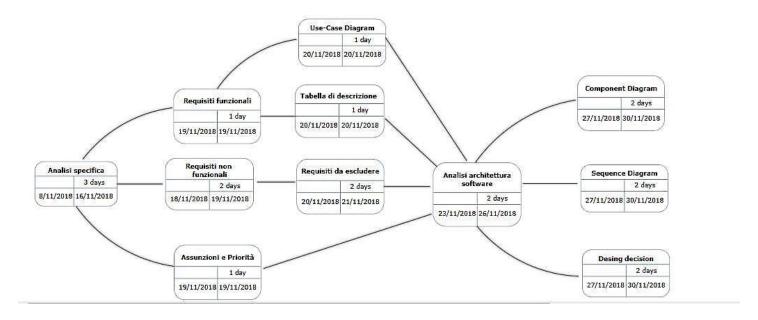
Log Template Deliverable 1:

	nen h/Day)	Time spent	Partners (please report how many people have been working)	Brief Description of the performed task	Category	Sub-Category
11	08-16	25:00:00 *	5	Analisi specifica, ragionamento sul sistema da realizzare in modo generale e elenco delle decisioni da	Learning	/
11	19	01:30:00	1	Ricerca dei requisiti funzionali	Learning	Doing
11	17-19	06:00:00 *	2	Analisi dei requisiti non funzionali	Learning	Doing
11	19	01:30:00	1	Assunzioni e priorità	Learning	Doing
11	20	02:00:00	1	Use case diagram e tabelle di descrizione	Learning	Doing
11	20	02:00:00 *	2	Requisiti esclusi	Learning	Doing
11	23-26	16:50:00 *	5	Analisi architettura	Learning	/
11	27-30	01:30:00	1	Component diagram	Learning	Doing
11	27-30	03:00:00	1	Sequence diagram	Learning	Doing
11	27-30	01:30:00	1	Design decisions	Learning	Doing
11	30	01:00:00	1	Pert	Learning	Doing
11	30	30:00:00 *	5	Revisione e confusione documentazione		Doing
ore	otali	90:70:00		Media a persona 18h circa		

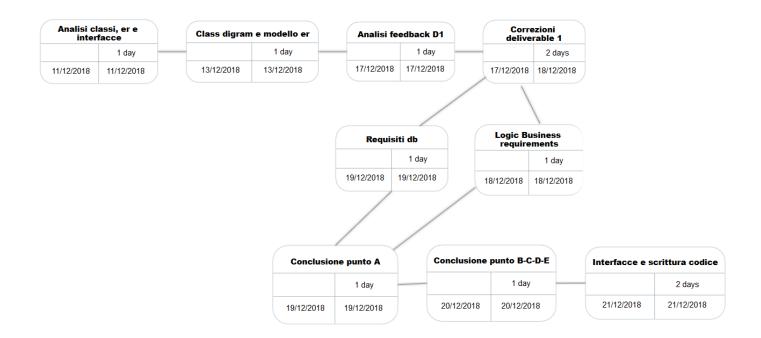
Log Template Deliverable 2:

Wh (Mont	ien h/Day)	Time spent	Partners (please Iepost how mang people have been	Brief Description of the performed task	Category	Sub-Category
12	11	12:30:00 *	5	Analisi delle possibili classi, del modello ER e dell'interfacce	Learning	/
12	13	10:00:00 *	5	Bozza class diagram e modello ER	Learning	Doing
12	17	07:30:00 *	3	Analisi dei feedback della Deliverable 1	Learning	/
12	17	04:00:00 *	2	Correzioni deliverable 1	Learning	Doing
12	18	27:30:00 *	5	Analisi deliverable 2 e correzione deliverable 1	Learning	Doing
12	18	03:00:00	1	Logic Business requirements	Learning	/
12	19	00:30:00	1	Requisiti del db	/	Doing
12	19	10:00:00 *	5	Conclusione punto A. Requirements collection con tutte le parti contenute in esso	Learning	Doing
12	19	07:30:00 *	3	Conclusione punto A. Requirements collection con tutte le parti contenute in esso	Learning	Doing
12	20	12:30:00 *	5	Conclusione punto B-C-D-E con tutte le sue parti, bozza interfacce	Learning	Doing
12	20	12:00:00*	4	Conclusione punto B-C-D-E con tutte le sue parti, bozza interfacce	Learning	Doing
12	21	05:00:00	1	Interfacce grafiche, implementazione classi e collegamento tra le varie interfacce, creazione db e codice di connession	Learning	Doing
12	22	04:00:00	1	Scrittura codice, punto F deliverable 2	Learning	Doing
12	22	06:00:00*	3	Revisione Documento con inserimento ultime immagini	Learning	Doing
12	23	00:30:00	1	Pert	/	Doing
12	23	03:00:00*	3	Revisione completa del documento	Learning	Doing
ore totali		124:10:00	Ī	Media a persona 24h circa		

Pert Deliverable 1:



Pert Deliverable 2:



Appendix. Code

<Report in this section a documented version of the produced code. I do not need a copy&cut of</p> 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>